# The Politics of Mobile Code

The Politics of Mobile Code
L. Mosemghvdlishvili. The Netherlands, 2021

Cover and chapter design by: Lela Mosemghvdlishvili and Fred Feij
Printed by: Print Service Ede / Proefschriftprinten.nl

ISBN 978-90-831295-1-8

Proefschriftenprinten.nl
Kelvinstraat 27
6716 BV EDE

# The Politics of Mobile Code
# Exploring Possibilities and Limitations of the Programmability of Smartphones

Een verkenning van mogelijkheden en beperkingen voor gebruikers bij het vormgeven van mobiele technologieën

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
B y command of the
rector magnificus

Prof.dr. F.A. van der Duijn Schouten
and in accordance with the decision of the Doctorate Board

The public defence shall be held on 25 February 2021
at 13:00 hrs
by
Lela Mosemghvdlishvili
born in Tbilisi, Georgia

**Doctoral Committee:**

**Promotors:** prof.dr. J. Jansz

prof.dr. P. Arora

**Other members:** prof.dr. J.F.T.M. van Dijck
prof.dr. T. Poell
dr. J.H. Pridmore

dr. M. Slot
dr. M.T. Schäfer

Dedication

*This thesis is lovingly dedicated to the memory of my father Zakharia Mosemghvdlishvili.*

# Abstract

This dissertation provides a qualitative inquiry into the programmability of one of the world's most pervasive devices, smartphones, and explores the conditions, possibilities and limitations for expert users to engage in shaping the affordances of these devices. How do hobbyist developers and digital right activists interpret the meaning of the programmability of smartphones? And how do their practices connect to, depend on, and contest the structural limitations imposed by more powerful actors in the smartphone ecology? This dissertation advances the argument that, due to software's intrinsic prescriptive and performative qualities, opening up the programmability of pervasive technologies has the potential to increase the range of relevant social groups who are involved in the negotiation process of smartphone development. It explores and evaluates how the process of programmability is both enabled and constrained in the context of mobile devices. Three qualitative, empirical studies investigate different aspects of how developers write apps for competing platforms; how digital rights activists politicize software in the context of mobile devices; and how free software activists create Free/libre and open-source software (FLOSS) apps for mobiles. The dissertation's findings point to a major shift in the way programmability is enabled. Three characteristics of the new model are the atomization of apps as single commodities; the gatekeeping role of app stores; and structural reliance on data surveillance. In the context of smartphones, both proprietary software and open-source programmability are embedded in data surveillance, which is increasingly contested by both digital rights' and free/libre and open-source software (FLOSS) activists. Furthermore, Google's hegemony in the realm of open source for mobile devices is resisted by digital rights activists, who create tools and software for smartphones to bypass data surveillance. All in all the discourse of free software has shifted away from the developers' four freedoms and access to source code toward the criticism of dependence on centralized servers and data surveillance.

# Preface

In this preface, I intend to lay bare what academic work (executed according to the conventions of contemporary research methodologies, written in academic prose, negotiated within academic structures, and performed by me) has meant for me as a human being: as an individual with my own unique experiences; family and interpersonal relationships; personal desires, hopes and fears. This manuscript is merely the top of a heavy iceberg, a travelogue of the lengthy, and at times thorny, journey of being socialized into academia and the process of coming to identify as a part of it.

When I first accepted a paid position as a researcher and lecturer at Erasmus University Rotterdam, I was profoundly excited. After having worked for two years in a corporate market research consultancy, going back to academia was a significant opportunity for me. Even though my new full-time salary was just half of what I earned at the consultancy, I was still eager to make the transition, as working in academia meant the possibility *to be free* in conducting my own research. Nowadays, I see the same enthusiasm among my students, which I have come to recognize as the beauty and resilience intrinsic in our desire to explore and uncover, to learn and teach others about the world(s) we inhabit.

Ultimately, this project turned out to entail much more than simply earning a degree. Along the way, it led me, paradoxically, to both lose myself and become myself. Through this process of change, I discovered a whole new world; experienced a profound shift in my worldview; became a vehement critic of anything related to capital; dove into radical philosophy; and experienced an increasing rupture between my academic work, worldviews and the structures of academia. Conversely, I also suffered the deterioration of my mental wellbeing, burnout, two years of depression, and a slow recovery.

Little did I know when I was a child, staying with my cousins in the countryside and burning thick tomes written by a man named Karl Marx in order to make a fire, that I would end up reading the very same works in my PhD years, and that they would open a swirling gateway into the world of critical theory, poststructuralism, Marxism, and neo-Marxism: in short, criticism of everything that ever existed for me. Back then in Georgia, during the collapse of the Soviet Union, even the air we breathed was imbued with antisocialist and anti-communist sentiments and tensed with the resurgence of nationalist patriotism. Like other children, I too had internalized these antisocialist and anti-communist views and understood well that there were symbols and figures that were "bad," and that among these were red flags and the voluminous books of Marx, Engels and Lenin.

Twenty-five years later, I would be sitting in my apartment in Amsterdam reading Marx's *Capital*, making handwritten notes in the margins, finding it utterly fascinating and feeling as if a veil had been lifted off my common-sense perceptions of the world. When I reached Chapter 6: "The Sale and Purchase of Labour-Power," I experienced a sort of intellectual epiphany, realizing that I now understood the hidden structure of social organization and power that Marx sought to reveal. This was during the second year of the PhD and my newfound fascination with *Capital* compelled me to explore Marxist analysis even further, in turn exposing me to more contemporary Marxist critiques of intellectual property, copyright, structurally biased academic publishing, and data commodification, among others.

The profound influence Marx's labor theory had on my thinking was also a sort of curse. I became virtually incapable of seeing other aspects of social reality as everything became filtered through the lens of the class struggle. As I advanced in my PhD career, published my first articles and attended academic conferences, my discontent only grew. There seemed to be a vast chasm between the type of social criticism articulated in intellectual debates and the way my colleagues and I actually lived in, acted through and navigated the broader structures of academia. To give just one example, when I received my author's copy of the book in which I published my first invited chapter, instead of happiness I found myself filled with rage. Just one week of (online!) access to the chapter that I wrote and contributed for free, and that reviewers also reviewed for free, would cost $25 US dollars for anyone without access to institutional subscriptions. I felt like a fraud, criticizing intellectual property in relation to software yet contributing to its enforcement by publishing in closed academic journals. Moreover, when I received an email and was asked to go back to the Blackboard system and double-check if I had (accidentally) shared any copyrighted material, and to delete it before an inspection would result in a fine for my university, I felt even worse.

The inability to reconcile my increased engagement with the critique of political economy and the reality of the structural conditions in academia resulted in a severe loss of motivation to continue working on the PhD. Experiencing subsequent delays as the end date of my appointment quickly approached only served to heighten my stress levels. One day, I was working in the Amsterdam Public Library (OBA), together with my close friend and colleague. When she noticed tears rolling down my face while I was writing, she demanded that I stop and admit that I needed help. That day, I reported sick at work and began a leave of absence. It was one of the most difficult yet most important decisions in my life. It compelled me to take a break, acknowledge my own limitations and, in the end, become a more resilient yet humble person.

After a two year hiatus, I came back to academia, started teaching again and, eventually, resumed my research and completed this manuscript. In the wake of this experience, I not only became capable of seeing through these overwhelming structural forces, but also developed the strength to not be swept up by them and instead to focus on the potential instances of creative resistance. Throughout the process, I received the invaluable and unwavering support of my supervisor, Prof. Jansz, who never doubted my ability to complete this project. Lastly, in a twist of irony, I have also benefited (from a productive standpoint) from the lockdown during the COVID-19 pandemic, and will always remember those quiet, productive days without any interruption that it afforded me.

# Acknowledgements

I owe a profound debt of gratitude to Professor Jeroen Jansz, who supervised this project from the very start, providing me with the unwavering support and steady guidance without which this project would never have been realized. I want to thank him for his impeccable academic integrity and the professionalism with which he approached incredibly difficult times in my life, and for never discouraging me from pursuing new theoretical and methodological paths (even diving into poststructuralist discourse theory!). I will long remember the lengthy discussions we had about the theoretical and methodological detours that I would occasionally make. Those moments were not merely feedback sessions, but also great learning opportunities. My sincere gratitude also goes to Professor Payal Arora, who stepped into the project as my supervisor in a much needed and most critical time, contributing her expertise and inspiration to the final phase of the writing. My heartfelt appreciation goes to two of my most beloved peers and colleagues: Joyce Neys and Johannes von Engelhardt. Not only have I shared my office with them, but we have also developed a deep and sincere friendship. Furthermore, this project would not have been accomplished without the goodwill, open disclosure and shared insights of the participants. My sincerest thanks to each and every one of them, as well as to the (anonymous) reviewers and conference discussants who offered insightful comments and feedback to help strengthen this project along the way. Special appreciation goes to my dear students, who helped me learn and interrogate my own views as much as I helped them to do so. I would also like to acknowledge the special contribution made by Dafni Mangalousi, whose master's thesis and subsequent collaborations with me have been very fruitful to my development as a scholar.Last but not least, I will always be grateful to my mother, sister and daughter, and close friends for their love and steadfast support throughout my life.

# Table of Contents

Abstract ........................................................................................ 5

Preface ........................................................................................... 6

Acknowledgements ....................................................................... 9

List of Figures ............................................................................. 15

List of Tables ............................................................................... 16

List of Abbreviations .................................................................. 17

Introduction .................................................................................. 19

Situating programmability ........................................................... 22

Focus of the study ....................................................................... 26

Theoretical positioning ................................................................ 27

Methodological overview ............................................................ 29

Structure of the manuscript ......................................................... 32

Chapter 2 A Conceptual Model to Study Programmability ......................... 35

Revising a constructivist view on technology .......................................... 36

Historical context ......................................................................... 36

The social construction of technology ................................................ 39

Limitations of the social construction of technology ........................... 42

Critical constructivism ................................................................. 45

Formal and substantive bias ....................................................... 46

Design code(s) ............................................................................. 49

Democratic rationalization .......................................................... 50

A glossary and visualization of the analytical tools ............................ 51

Chapter 3 Demarcating Smartphones ........................................................ 55

Intellectual property and smartphones .................................................. 57

Circuits of labor ........................................................................................ 61

Net Neutrality ............................................................................................ 66

Conclusion ................................................................................................. 69

Chapter 4 Negotiability of Technology and Its Limitations ........................ 71

Introduction ............................................................................................... 72

Theoretical framework .............................................................................. 74

Conceptualization of user-developer ................................................... 76

Research questions ............................................................................... 77

Methodology .............................................................................................. 78

Participants ............................................................................................ 78

Analysis ..................................................................................................... 81

Power dynamics within the smartphone industry ................................ 81

Competing platforms ............................................................................ 83

The review process and control over distribution ................................ 88

Developers' reflections on the future of app development .................. 89

Discussion ................................................................................................. 91

Conclusion ................................................................................................. 94

Chapter 5 Free Your 'Most Open' Android ................................................ 97

Introduction ............................................................................................... 98

A primer into FLOSS licenses ........ 101

Discourse-theoretical framework ........ 104

Political discourse theory ........ 104

Analytical tools ........ 105

Exemplary texts and analysis ........ 106

Analysis ........ 109

Signifying Android ........ 109

Privacy and control ........ 113

Subject positions ........ 114

Social imaginaries on mobile technology ........ 116

Discussion ........ 116

Conclusion ........ 117

Chapter 6 Libre Software for Mobiles ........ 119

Introduction ........ 120

Literature review ........ 123

Making of FLOSS (1982-1998) ........ 123

Fragmentation and the birth of the dual regime (since 1998) ........ 125

Political and socio-economic relevance of FLOSS ........ 128

Research questions ........ 133

Methodology ........ 134

Thematic analysis ........ 135

Reflexive bracketing ........ 137

Introducing participants ........ 137
Analysis ........ 138
Becoming a FLOSS activist ........ 138
Situating FLOSS for mobiles ........ 143
Licenses: a pinnacle of disagreement ........ 148
Android - its dominance and discontent ........ 150
Libre software for smartphones ........ 154
Conclusion ........ 157
Conclusion ........ 163
Bibliography ........ 173
Appendix 1: Portfolio ........ 208
Graduate courses ........ 208
Conference presentations ........ 209
Review work ........ 210
Publications ........ 211
Other ........ 211

# List of Figures

Figure 1 Conceptual model to study programmability ................................. 54

Figure 2 Conceptualization of an app user-developer .................................. 76

Figure 3 Divergent radial: Perceptions of FLOSS movement.................... 147

# List of Tables

Table 1  Overview of research sub-questions and corresponding chapters.. 31

Table 2 Key concepts used in the thesis with primary sources and critical revisions ........................................................................................... 52

Table 3  Demographics of interview participants......................................... 80

Table 4 Comparative Table of app stores' conditions in 2012 ..................... 84

Table 5 Exemplary texts included in the signifier-level analysis............... 107

Table 6 Division of roles in relation to the code production of Android ... 115

Table 7 Participants' background and involvement in FLOSS projects..... 138

# List of Abbreviations

3GPP - 3rd Generation Partnership Project

3GPP2 - 3rd Generation Partnership Project 2

AOSP – Amsterdam Open Source Project

CDA – Critical Discourse Analysis

CTT – Critical Theory of Technology

DRC – Democratic Republic of Congo

DRM - Digital Rights Management

EFF – Electronic Frontier Foundation

FCC – Federal Commission of Communication

FLOSS - Free/Libre and Open-Source Software

FSF – Free Software Foundation

FSFE - Free Software Foundation Europe

GPS - The Global Positioning System

GSM - Global System for Mobile Communications

ISPs – Internet Service Providers

ICT - Information and Communication Technology

ITU - International Telecommunication Union

## List of Abbreviations

OS – Operating System

PDA – Personal Digital Assistant

PDT – Political Discourse Analysis

SACOM - Students and Scholars against Corporate Misbehavior

SCOT – Social Construction of Technology

SDK – Software Development Kit

SOMO - Stichting Onderzoek Multinationale Ondernemingen

SSK - Sociology of Scientific Knowledge

SST – Social Shaping of Technology

# Introduction

The paradox of our times is the incommensurable distance the average person feels between herself and the processes that produce most items in her immediate surroundings, whether it be technological gadgets, clothing, food or pieces of furniture. Yet, it is fair to state that each of us spends a considerable portion of our conscious life making a living in order to afford those items, which “fill up” our being, give us a sense of who we are and enable us to communicate, mediate and express ourselves. Despite the tremendous dependency on technology in almost every aspect of our lives, the average person remains far removed from the decision-making processes in technological development (Feenberg, 2002). In fact, they have little opportunity to engage with the design and development of the very technology that structures their everyday life, other than by exercising the so-called *veto power of a consumer* in order to either embrace or reject it (Cockburn, 1993). In this thesis, I will be critically examining the social shaping of one of the most pervasive devices in our contemporary lives, the smartphone, and attempting to understand the politics of its making.

A mere look around on any form of public transport immediately confronts us with the undeniable omnipresence of these devices. The individual immersed in their smartphone, whether on the street, at a bus stop, on a lunch break, in the lecture hall or even at purportedly “social” gatherings, has become the new “normal.” The International Telecommunication Union’s (ITU) time-series data by geographic region (2005-2019) sheds light on the global dynamics of the smartphone’s adoption. Comparing indicators such as household ownership of a computer and access to the internet, it reveals an interesting tipping point. Since 2018, more households around the globe have access to the mobile internet than own a computer (ITU, 2019). What this data points to is the significance of the smartphone as an access technology through which millions of new users become connected to the global information network. When evaluating this data, however, it is important to bear in mind that it pertains specifically to the 52% of the world’s population that is connected to the internet (ibid.).

Though fascinating to consider, the aim of this thesis is not to engage directly with the broader social and/or existential implications of the pervasiveness of smartphones (e.g. on what our 24/7 connectivity means for our perception of time and space, habits, routines, and relationships), but rather to direct attention to *how* these devices are shaped: to the politics of their making. In the title of the manuscript, I use the word "politics*,*" which is understood in this thesis as micropolitics, or the social relations among actors involved in negotiating the development of smartphones. In other words, *I am concerned with who is able to participate in shaping such pervasive and personal technologies and how such participation occurs.*

The social relevance of studying the conditions and practices of *programmability*[1] is primarily influenced by the increased role of software as a means to relegate agency and inscribe complex processes into contemporary technologies (Berry, 2016). Code is "modelling the future" (Franco "Bifo" Berardi, 2013, p. ix), in the sense that future modes of action are inscribed through code. The prescriptive and performative qualities of software make the study of the programmability of smartphones extremely relevant, especially when bearing in mind that writing software is a human activity, deeply embedded in the broader social norms of a particular society, as well as echoing the values and practices of their makers/authors.

Smartphones represent a special case for several reasons. First, these devices represent a convergence of hitherto different industries - computing, internet, and telecom - into one global, mobile sector. Given their differences in terms of history, degree of concentration, culture and professional norms, this convergence has caused the reshuffling of power centers in the ICT sector into

[1] In this thesis, programmability is understood as an aspect and intrinsic quality of digital technologies that is made possible through computational code: by writing a coherent program, that is, software. Throughout the text, I will regularly use the terms programmability and code/software interchangeably. Only when I refer specifically to computational (programming) languages or software (as a product) will I more carefully distinguish between them and use them separately.

a new constellation. Second, smartphones are programmable cell phones. In this context, programmability is defined as the capability to run a complete operating system (OS) as well as to install application software, also known as apps (Raento, Oulasvirta, & Eagle, 2009). Programmability effectively transforms a cell phone into a multipurpose digital device. Third, smartphones are not merely ubiquitous but have also become deeply and inextricably embedded in our daily life and instrumental to how individuals access and participate in the global information society.

In this thesis, I develop the argument that the programmability of a smartphone is a decisive factor in shaping potential affordances of the device because it is precisely through the programs (software, apps) that different functionalities are created. What an end user can do with a smartphone, to what social use she can put this converged device, is mediated by software. In light of this, I propose that programmability represents a *potentiality* for opening the shaping of smartphone affordances to wider social groups. Through writing software, new affordances and possible social uses can be encoded in devices and made available to billions of users. This, in turn, can lead to reducing the distance between design and development on one the hand and the use of technologies on the other: the paradox of which I spoke in the opening to this introduction.

Chapter 1

# Situating programmability

"Code's colonization of the political makes it
a battleground for democracy." (Chun, 2006)

"Proprietary code is language in debt."
(Franco "Bifo" Berardi, 2013, p. xii)

The premise of this dissertation is to examine programmability with respect to smartphones and evaluate it in terms of its potential for democratizing access to the design and development of these technologies (by enabling participation in structuring affordances through programmability). Before delving further into the research design of this thesis, I will situate how programmability has been conceptualized within (new) media and technology studies.

To begin, the increased significance of and reliance on programmability is easy to demonstrate because not only media technologies, but also a vital part of societal infrastructure at large has become code/software-enabled and mediated. Indeed, whether with respect to consumer goods, media, or personal technologies, software now mediates a vast array of social activities to which such hardware/devices can be put. And although societies have long relied on bureaucratic processes such as sorting, sending, receiving, and processing information to enable societal institutions to function, the introduction of software/code into these processes, coupled with the pervasiveness of algorithms, has marked a significant shift. Berry (2016) conceptualizes this shift as the *relegation of agency* of inscription systems from material carriers (like paper) onto software, with a major difference between them being that software is mutable and performative. Thus, this shift marks a transition that enables the delegation of complex and highly sophisticated mental processes to machines and computational systems. This is a crucially important dimension because it instills a "greater degree of agency" into technical devices (ibid., p. 2). Earlier, Lessig (1999) proposed examining the role of code as being comparable to that of the law. In this context, Feenberg (2010) writes that the implications of this relegation are greater than the implications Weber saw for the increased bureaucratization in modern societies.

Within the field of social studies of technology, Williams (1997) has characterized software as a "critical layer" in modern multimedia technologies due to the unique function it serves in linking machines/devices/hardware to their potential social uses. Moreover, a number of theorists also consider the *prescriptive* function of software to be vital to its understanding and proper conceptualization.

In conceptualizing the prescriptive function of software, Fuller (2006), Manovich (2002, 2013), and Glazier (2006) emphasize its resemblance to language and the means of production. Manovich (2002) asserts that software is the very language of new media. He argues that software is like a medium itself because it enables and constitutes new media forms, influencing how content is created, organized, and consumed. In his later work, he expands this argument and emphasizes that software is not only the backbone of media but that it also permeates all areas of contemporary society. Computational code resembles a language, writes Glazier (2006), as it commonly serves as a means to create something new (a program, or a novel function for hardware). Code is also transmutable, meaning that it is intrinsically open to modification (Hughes & Lang, 2006). [2] On the whole, programmability has been acknowledged as inherent to digital media, as its "intrinsic" modality (McPherson, 2006). Even in circumstances when code/software is not immediately visible (e.g., when looking at the interface on the screen), it is still active through computation, creating the representation of the interface in real time.

The second quality that has been emphasized in conceptualizing code is its *performativity* (Mackenzie, 2005), meaning that it not only serves as prescriptive instructions but that it also enacts these activities (when code is run/executed). Drawing on cultural studies, most notably the works of Judith Butler and Jacques Derrida, Mackenzie (ibid) has introduced the concept of

[2] If it is not foreclosed through techno-legal constraints such as a restive (proprietary) license that criminalizes modification, or the Digital Rights Management (DRM) instruments that put limitations on usage (regional lockout, limited installations or through product keys).

performativity in order to analyze Linux[3] (the FLOSS operating system) and demonstrate that complex software is not static but rather a performative cultural object. The performativity of such objects implies not merely the execution of code but the active enacting of practices that enable such performativity. In other words, Linux draws force from existing inscriptions and sets of rules, which are distinct from the rules under which proprietary programs are created.

Given the qualities of code/software that we have reviewed thus far, the reader might easily get the impression that code/software embodies increasingly autonomous, even self-perpetuating (performing), lines of script. This view would be misleading, however, because both prescription and performativity rely on human activities. Code/software production is essentially a social activity (Berry, 2008) and has been regarded by many researchers as one of the first examples of user participation (Bruns, 2007; Schäfer, 2008).

User participation through collaborative code production has, in turn, foreshowed the emergence of the *peer-to-peer model of production* (Benkler, 2006; Kostakis & Bauwens, 2014), a decisively important development which is increasingly woven into the fabric of contemporary society. A prime example of collective code production was Linux, which has been extensively covered in the academic literature. Linux is widely understood to represent and embody a critique of the corporate organization of labor (specifically concerning code production), as well as the existing ownership of computer code (Himanen, 2001; Moody, 2001).

That participation has been  made possible through access to source code. In short, a computer program can be closed, fixed, or open to modification and adaptation (that is to say, its source code is released  publicly for users to modify). Access to the source code[4] has been seen as empowering users,

[3] Linux is a FLOSS operating system collectively produced by thousands of dispersed volunteers.

[4] The reasons why source code must be open have been argued by many authors based on different grounds, which will be discussed more in-depth in Chapters 5 and 6. For now, it will suffice to note that there has been an ideological and discursive struggle

equipping them with the agency to modify prescriptive functions of performative cultural objects. Without such access to the source code, a program has a "fixed" function. Thus, what a user is capable of doing is "downstreamed" as opposed to allowing for bottom-up modifications. In this context, "downstreamed" means that functionalities are encoded in a closed manner. There is no possibility to alter and change them (other than hacking and violating the copyright of proprietary software).

The implications of access to the source code and free and open-source software development has been theorized by several scholars. Notably, Jesiek (2003) has proposed open-source development as a potential source for the democratization of technological development: as representing "one more degree of freedom in the proactive shaping and modification of technologies, both in terms of design and use" (ibid., p. 1). Similarly, Berry and Moss (2006) have argued for the democratizing potential of open (non-proprietary) software for governments, through which values such as transparency and inclusion could be extended into the technological domain.

A note of caution: Although I stress the critical role of code/software in potentially democratizing access to the design and development of technologies, I also heed warnings against the temptation to fall into code determinism. Chun (2006, 2008), among others, has warned of the pitfalls of falling into *software fetishism* and against interpreting and understanding software outside of its material and social contexts.

over articulating the reasons why source code should be open. This can be split into two ideological camps. On the one hand, there is the  Free Software Movement (with Richard Stallman as the founder), which sees free code in terms of the developers' four freedoms rather than merely in terms of price. On the other hand, there is the Open Source Initiative (OSI), which argues that access to code is not a question of basic human rights, but of artisanship and craft, and that it is also a more efficient way to produce better quality.

## Focus of the study

The central issue addressed in this dissertation is the politics of mobile code, understood as the possibilities and structural limitations of writing software for smartphones. I probe into the implications of programmability and examine the extent to which an increase in the range of actors involved in shaping the affordances of smartphones can be considered to be a step in the direction of democratizing technological design. Put another way, I will explore practices of programmability of smartphones, and evaluate their consequences in light of *democratic rationalization*. Democratic rationalization (Feenberg, 2002, 2010, 2017) imbues my inquiry with a normative emphasis. It points toward the need to democratize technological design and to orient the design and development of the artifacts that structure our everyday lives towards human-centered and environmentally conscious goals. Democratic rationalization is an oxymoron coined by Feenberg, through which he aims to evoke Weber's work on rationality (as understood in terms of control and domination), but combine it with the contrasting term democratic. Through this combination, he aims to emphasize the possibility of salvaging rationality from technocracy and subjecting it to democratic values and processes (2013, p. 22). Democratic rationalization stands in stark contrast to technocratic rationality. It promotes subjecting the process of negotiability of technological development to public involvement by inserting democratic values and processes into the process of technological design.

To recap, this dissertation aims to explore the social shaping of the programmability of smartphones and to consider its consequences for the democratization of structuring the affordances of this pervasive technological artifact. Due to the inherent qualities of code, writing software can become a means to structurally widen the deliberation in the initial phase of developing technologies. Various scholars have supported linking collaborative software production through decentralized networks with the potentiality for the democratization of technological decision making (Berry & Moss, 2006; Fortunati, 2006; Jesiek, 2003).

## Theoretical positioning

In this thesis, software is positioned as a critically important layer in mediating the functionalities and defining the affordances of contemporary mobile technologies. However, this does not mean that software is considered to be a determining force. On the contrary, software (just like other technologies) is seen as a product of social activities, molded under the conditions of specific social relations and in accordance with the values of the actors involved. The view on technology articulated in this thesis is embedded in the social shaping of technology perspective and emphasizes *negotiability* as a starting point (Williams & Edge, 1996; Williams, Stewart, & Slack, 2005). It also embraces a normative commitment to the democratization of technological decision making by integrating insights from the critical theory of technology[5] (Feenberg, 1999, 2002, 2010, 2017).

At its core, *technology* in this work is defined as a totality of artifacts, including the material aspect of hardware as well as the immaterial performativity of software, together with the social relations and knowledge necessary to produce and sustain their functioning. Throughout this manuscript, I set forth a definition based on a critical constructivist view of technology, emphasizing the *co-shaping* of the social and the technical. I distance myself from interpreting technology as either *neutral tools* or as a *self-propelling force* with an inherent essence that drives human progress (or, on the contrary, deprives us of our "humanness"). Co-shaping means, on the one hand, that technology is contingent and shaped through social forces, but, on the other hand, that it also shapes our living conditions and influences social relations, thus organizing and mediating our lives in specific ways.

To uphold such a definition, I draw on two notions: the *negotiability* and the *ambivalent neutrality* of technology. The first notion stems from the tradition of the social shaping of technology (MacKenzie & Wajcman, 1985, 1999; Williams, 1997; Williams & Edge, 1996); the second from the critical theory

[5] Andrew Feenberg also uses the term *critical constructivism* to refer to his work (Feenberg, 2020, p. 27).

of technology (Feenberg, 1991, 1999, 2002, 2017). *Negotiability* of technology opposes the idea of the inherent logic of technology advanced by technological determinism. Negotiability means there is not one and only one, but rather a number of technological solutions available for any social problem. Which of these solutions becomes adopted, standardized and integrated into the fabric of social life is not an autonomous process but rather is socially shaped and influenced by social groups and forces. By grounding the definition of technology in the notion of negotiability, I explicitly position this study in the tradition of the social shaping of technology and thereby adopt a particular stance with respect to the relation between the social and the technical.

The second notion of *ambivalent neutrality* is a combined concept that I have coined in order to highlight a nuanced position that rejects not only the essentialist claim that technology has an inherent essence, but at the same time, also refutation of the idea that technologies are capable of carrying ingrained values and biases. My inspiration for coining this combined concept rests on Andrew Feenberg's decade-long work in the field of the critical theory of technology (Feenberg, 2002). In a nutshell, Feenberg proposes that "in certain cases, *neutrality and bias* are not opposite but merely different aspects of a single concrete object" (ibid., pp. 80–88). What he articulates here, in my understanding, is a rather uncommon view that "neutrality and bias" can co-exist in one technological object [6] or in the whole technological configuration/system[7]. How neutrality and bias can co-exist will be further

[6] To illustrate, Feenberg (2002) uses the example of an assembly line, which, on the one hand appears to be neutral, as the machine is composed of relatively neutral elements and is blind to any social distinctions, such as *who* operates/uses it. However, it is still *biased* in its way of reconfiguring labor in small repetitive steps, which can be performed without much training and can be easily replaced by another worker. Hence, the bias is not in its working, per se, but in making, selecting and introducing this particular configuration in the real world.

[7] Similarly, in Chapter 5, where I will be discussing Google Play (the app store), I argue that it appears to be neutral in its treatment of all apps because, unlike Apple, it has no review process. However, this does not mean that that system has no biases built into its design. For example, if a developer wants to place her own app on Google

elucidated in the theoretical chapter while reviewing the notions of *formal bias* and *technical code*. However, for the present purpose of the introduction and the positioning of my study, it will suffice to state that, in my view, adopting a definition of technology that rests on these concepts (*negotiability* and *ambivalent neutrality*) prevents reducing technological artifacts to a priori neutral or a priori value-laden tools, and opens up space to inspect the values embedded in their design. In other words, acknowledging both the contingency of technological development (through negotiability) as well as the potential for ambivalent neutrality (in which bias and neutrality can co-exist) is the point of departure taken for this manuscript to study the politics of software development for smartphones.

## Methodological overview

In this thesis, I aim to explore, describe, and evaluate a particular social phenomenon: that is, the shaping of the programmability of smartphones. Through empirical studies, I aim to answer the following guiding research question: "*What are the conditions (possibilities and limitations) of writing software for smartphones?",* and to evaluate this software writing process in light of democratic rationalization.

In order to explain how the research object (which is not a reified "thing" but rather a complex social process) is demarcated, I will draw on Blaikie's notions "logic of inquiry" (2000; see also: Blaikie & Priest, 2019), which categorizes research strategies in the social sciences into four main types: inductive, deductive, abductive and retroductive. Each logic of inquiry is associated with specific philosophical traditions and incumbent research paradigms. According to this typology, the current PhD thesis is carried out using abductive and retroductive logics of inquiry. Abductive logic, as Blaikie explains, refers to "the process used to generate social scientific accounts from social actors' accounts; for deriving technical concepts and theories from lay

Play, she must participate in the Google cloud services as well as partake in datafication and consumer surveillance.

concepts and interpretations of social life" (2000. p.114). It best fits with the interpretivist framework and draws on data sources such as interviews. Through such a perspective, the social word is experienced and interpreted by research participants. Their subjective meanings become intersubjective, because members of a particular group share interpretations and common understandings. In this thesis, the use of abductive logic in designing my research is visible through my choice of methodology, data sources and forms of data gathering (see: Table 1 for an overview).

Alongside an abductive logic of inquiry, this thesis also employs a retroductive research strategy. In Blaikie's (ibid.) view, retroductive logic shares with deductive logic the aim to *explain*, but *how* the explanation is achieved is different. Namely, the retroductive logic of inquiry aims to establish and (re-)construct the underlying mechanisms that produce certain regularities in observable phenomena. Conceptual (analytical) models are constructed for this purpose. These models are then used to analyze and understand the phenomenon. In this thesis, I develop an interdisciplinary conceptual model (see: Chapter 2, Figure 1), synthesizing analytical tools from the social shaping of technology, critical constructivism, and software studies into an integrated framework. The conceptual model is introduced and further explained in the coming chapter and serves as the analytical model through which the shaping of programmability is comprehended, understood, and evaluated.

This PhD project is an article-based manuscript, meaning that the guiding research question is divided into sub-questions and explored in separate studies. For an overview (see: Table 1), I present the sub-questions per chapter and indicate the sources and forms of data used. Further methodological details regarding of how the respective studies were designed and carried out are explained in the corresponding chapters.

The **guiding research question** asks: "*What are the conditions (possibilities and limitations) of writing software for smartphones?*" and evaluates this software writing process in the light of democratic rationalization.

**Table 1** Overview of research sub-questions and corresponding chapters

| *Chapter* | *Research sub-questions* | *Sources of data* |
|---|---|---|
| Chapter 1 | n/a | n/a |
| Chapter 2 | n/a | n/a |
| Chapter 3 | 1. Which relevant social groups are involved in the shaping of smartphones?<br>2. What are the points of tension and disagreement between the relevant social groups involved?<br>*3.* Which (competing) design codes can be identified in the process of the co-shaping of smartphones? | documents, industry literature, company and press materials |
| Chapter 4 | 4. How do expert users (developers) position themselves in relation to other relevant social groups?<br>5. How is code production (programmability) made possible for mobile platforms?<br>6. What are the structural limitations that expert users encounter in this process? | interviews, documents |
| Chapter 5 | 7. How is Android discursively constructed and contested?<br>8. How does Google articulate what an open-source platform is?<br>9. How do digital rights activists contest Google's articulation and practices of open source? | documents, press material by Google, and FSFE |
| *Chapter 6* | 10. How do FLOSS activists construct the meaning of free software in the context of mobile devices?<br>11. How do FLOSS activists evaluate the dominant Android platform from a free software perspective?<br>12. Which initiatives do they consider to be the most important for bringing free software to mobile devices, and why? | interviews |

## Structure of the manuscript

This thesis is organized into six chapters. Thus far in this introduction, the overall aim of the research and the guiding research question have been introduced, the social and scientific relevance of the study has been explained, and a brief overview of the theoretical and methodological positioning has been provided. The content and objectives of the subsequent chapters are as follows:

**Chapter 2 A conceptual model to study programmability** elucidates the theoretical framework of the dissertation, through which the programmability of the smartphone has been analyzed in the subsequent empirical studies. I first conceptualize code/software and review its intrinsic qualities (being transmutable, prescriptive and performative). Following this, I embed programmability in the tradition of the social construction of technology (SCOT) and review its core concepts: *negotiability, interpretative flexibility, relevant social groups,* and *closure*. These analytical tools are used throughout the empirical studies in the thesis. Following a review of the strengths of these analytical tools, I delve deeper into criticism of SCOT. Finally, I propose a way to reconcile the shortcomings of SCOT by integrating insights from the critical theory of technology. More specifically, I borrow analytical tools that refer to *formal vs. substantive bias*, *design code*, and *democratic rationalization*. The chapter serves as the theoretical backbone, analytical toolbox and rationale of the thesis.

**Chapter 3 Demarcating smartphones** serves to provide relevant background information for the thesis and to demarcate the domain of the shaping of smartphones. I apply analytical tools, namely the concepts: *relevant social groups, negotiation*, and *design code*, to review the most important changes as well as the most "troublesome" (Goggin, 2010) points of tension between relevant social groups. First, I reconstruct the events that led to the creation of app stores and argue that the decision by Apple to simplify the development of software for iPhones (as a response to jailbreaking) was a decisive step, which other OS providers soon replicated to such a degree that app stores became the core organizing principle of the smartphone ecosystem. Second, I turn to examine how the global value chain is constructed with respect to smartphones and deconstruct the complex geography of smartphone making. Questions of both material and immaterial labor are addressed. Third, I dive into patent usage and litigation in the case of smartphones, which were notorious in the first phase of smartphones' introduction to the market. Last but not least, I review changes that have resulted from mobile network carriers' interest in redefining net-neutrality. By considering the following four aspects: app stores, labor, hardware patentability and network neutrality, I illustrate tensions between the relevant social groups involved, which differ in their understanding (i.e, interpretation of the interpretative flexibility) of these technologies.

**Chapter 4 Negotiability of technology and its limitations: Practices of app development** presents the first empirical inquiry: namely, an exploration into how expert users (hobbyist developers) write software (apps) for smartphones. Through 20 semi-structured interviews with developers from 12 different countries, the chapter dives into how developers create apps for smartphones and thereby negotiate the development process of this technological artifact. Based on insights from the participants, the technical and structural limitations that these individuals encounter in the process of app development are identified and contextualized. The findings of the study point to the emergence of app stores as a very important factor in the process of negotiability of smartphone programmability. The major OS providers are tied to their respective distribution channels and, by defining the terms and conditions of how an app can be distributed, they take on the role of gatekeepers. OS providers also dictate the programming language in which an app needs to be written. For hobbyist developers, this is a hindrance. In the absence of interoperability, they encounter the need to rewrite the same app in various programming languages to suit different platform requirements. Differences between competing platforms are identified, such as accessibility, costs, barriers to entry, and, most importantly, the degree of control that their respective app stores exercise over submitted content. Google's Android is generally considered by developers to be more accessible (easier to start developing and cheaper, as it does not require payment from developers), but not without limitations, as the following chapter will illustrate.

**Chapter 5 Free your "most open" Android: A comparative discourse analysis** takes the findings of the first study as its starting point and dives further into the limitations of the so-called "most open" Android. By conducting a comparative discourse analysis, I deconstruct how Google (as the main developer and owner of Android) communicates and articulates what an open platform and open source development is. I juxtapose this with the discursive practices and communicative acts produced by digital rights activists within the Free Software Foundation Europe (FSFE). Methodologically, I use Laclau and Mouffe's discourse theory and its key analytical tools (key signifiers and chains of signification) to identify, compare and deconstruct how these two organizations articulate the meaning of Android. I find Google's definition of open source practices strictly *conditional* and argue that Google uses Android compatibility not only rhetorically, but also techno-legally to exercise control over the Android platform. On the other hand, free software activists in the "Free Your Android!" campaign expand into questions of surveillance and attempt to politicize access to code in the mobile domain.

**Chapter 6 Libre software for mobiles: A qualitative inquiry into the meanings and practices of writing FLOSS apps for smartphones** is the last empirical study in the thesis. It shifts attention to the lesser-known domain of Libre apps and custom modifications of Android operating system. Through in-depth interviews with seven key individuals engaged in mobile FLOSS projects (lead developers of the major initiatives), it provides an exploration of how activists give meaning to code/software in the context of smartphones, evaluate the dominant platform, and create alternative software for mobiles. The Free/libre software community in the mobile context is a small but a vibrant scene, actively relying on volunteering, political campaigning, and alternative networks of organizing.

The **conclusion** is the final chapter of the manuscript. Here, I first contextualize and discuss the most important findings of the three empirical studies. Following this, I weave together insights gained through the inquires and provide an answer to the guiding research question of the thesis. The chapter also includes a discussion of the limitations of this manuscript and implications for future research.

# Chapter 2 A Conceptual Model to Study Programmability

In the introduction, I explicated the inherent qualities of code: its resemblance to language in its structure (Glazier, 2006; Manovich, 2002) and to a means of production in its working. Furthermore, I adopted a conceptualization of software/code as a critical layer of contemporary technology due to its function of defining affordances, a range of potential uses, as well as processes of relegating agency. Aside from its increasingly important role, I have argued in the introduction that software is *prescriptive* and *performative*. These qualities make software not only an especially interesting case, but also a critical one to explore considering access to and participation in the decision-making processes through which the affordances of smartphones are defined.

The conceptual model presented in this chapter expounds the premise that, due to its unique, inherent qualities (being transmutable, prescriptive, and performative), as well as its critical role in structuring the affordances of mobile technologies, opening the development and operation of code/software to wider public deliberation has democratizing potential. In order to develop and ground the analytical model, I first revisit and discuss the social construction of technology (SCOT), and review the key concepts of *negotiability, interpretative flexibility, relevant social groups,* and *closure*. These analytical tools are used throughout the empirical studies in the thesis. Following a review of the strengths of these analytical tools, I then delve deeper into the critical literature on SCOT, examining calls for its improvement, expansion and revision.

After highlighting the limitations of SCOT, I then propose a way to reconcile its shortcomings by integrating insights from the critical theory of Technology. This is done by introducing the concepts: *formal* vs. *substantive bias*, *design code* and *democratic rationalization.* By constructing this integrated conceptual model, I aim to expand SCOT without being forced to reject its useful tools. Further, I maintain that such an integration allows for the reconciliation of its shortcomings. In what follows, I will review the concepts that are noted in the model at (greater) length, paying special attention to their (a) historical origin and context of emergence, (b) criticism, and (c) applicability to the empirical studies of the thesis. At the end of the review, I will present a glossary (Table 2) and visualization of the key analytical tools in the form of a concept map (Figure 1).

## Revising a constructivist view on technology

> "Technologies constitute a site on which various social forces converge." (Mackay & Gillespie, 1992, p. 709)

### *Historical context*

The emergence of the social constructivist approach known as the social shaping of technology (SST)[8] represented a shift in focus from studying the effects/impacts of technology towards investigating *the process(es) of making* these technologies (Bijker & Law, 1992; Lievrouw, 2006; MacKenzie & Wajcman, 1999; Wajcman, 2002; Williams & Edge, 1996). In the 1980s, empirical studies examining the practices of scientists and engineers led to the perspective that technologies that had previously been perceived to be neutral and free from social values were, in fact, deeply embedded within social relations. These social influences, in turn, were intertwined with dynamics of class, gender, the state and economic interests, among others.

The term "social shaping of technology" was introduced by MacKenzie and Wajcman in a reader of the same title (1985), in which a range of studies were brought together to make a convincing case against technological

[8] To specify, the same broad direction within the studies of science and technology has been referred to by others as *constructivism* (Söderberg, 2011, p. 28), *social constructivism* (Winner, 1993, p. 364) or as the *social shaping of technology* (Lievrouw, 2006; Williams & Edge, 1996). These commentators acknowledge the differences among individual schools but treat them as belonging to the same broad approach.

determinism. In their introductory essay, the editors wrote that the volume aimed to persuade social scientists to pay at least equal time to the study of the effects of society on technology, as opposed to only investigating the effects of technology on society. Contributors to the volume univocally rejected treating technology as an independent factor. They provided empirical evidence that demonstrated how various social forces shape the design of diverse technologies.[9] A set of potential factors, including previous technology itself, economic interests and forces, the state and gender were put forward as the most influential in this process (ibid.).

Among the aforementioned factors, the role of economic forces was highlighted as the most powerful: “the economic shaping of technology is, in fact, the social shaping of technology” (MacKenzie & Wajcman, 1985, p. 15). The editors named the influence of private interests (e.g. through investments) as examples of how economic interest was shaping technological developments. Next to these examples, the interest of factory owners (as well as owners of other enterprises) to control workers and to reduce labour costs were named as more subtle ways in which economic forces were shaping technological artifacts. In the second edition of the aforementioned reader (MacKenzie & Wajcman, 1999), the editors collapsed the differences between the economic and the social by asserting that *the economic is also social*. “Furthermore, even if sure calculation of costs and profits—and even optimisation—were possible, the economic shaping of technology would still be its social shaping. Economic calculation and economic ‘laws’ are, after all, specific to particular forms of society, not universal, as Karl Marx famously argued.” (ibid.). By reformulating the argument in this way and affirming that the economic is social, the authors aimed to acknowledge the importance of

[9] In the volume (MacKenzie & Wajcman, 1985) included study by Thomas P. Hughes (1985) revealed how economic interests were pivotal in the design of the light bulb; Langdon Winner’s (1985) contribution proposed to pay attention to the inherent political nature and qualities of technologies (focusing on the example of the Moses bridge); and Cynthia Cockburn’s (1985) study showed how gender relations and patriarchal values were reinforced through the design and production of domestic technologies.

economic forces, yet at the same time attempted to move away from the universal and essentialist claims (as in orthodox Marxism) of economic determinism. For critics of the social shaping of technology, this reformulation indicated moving away from interrogating economic interests (Elzinga, 2004).

After the nineties, SST had become the dominant perspective in new media studies. As Lievrouw and Livingstone (2006) suggest, "it had displaced the technologically-deterministic discourse common in communication research at the time in new media studies" (p. 4). Just to name few studies that used this approach to study new media technologies; development of ugmented Reality (Liao, 2018), study of blogging (Siles, 2012), emergence of location-based media technologies (Fast, Ljungberg, & Braunerhielm, 2019), exploration of internet country domains (Hrynyshyn, 2008).

Williams (1997) wrote one of the first empirical works to explore software through the lens of the social shaping of technology. He examined what he called "emerging digital multimedia devices," which may be considered precursors of smartphones and are thus relevant to this thesis. Williams argued that three features were becoming characteristic of emerging ICT technologies: first, they were becoming *increasingly configurational* systems; second, the configurational aspect was resulting in the emergence of industry standards for enabling interoperability; and third, the proliferation of industry standards was leading to the creation of "locked-in" technologies.

Over the past forty years, SST has formed a rich intellectual tradition with different theoretical and methodological schools, without any clear orthodoxy. Williams and Edge (1996) suggest that it is best to see SST as an umbrella term under which various theories, each with their own methodological and conceptual tools, meet (for reviews see: Howcroft, Mitev, & Wilson, 2004; Williams & Edge, 1996). The three major schools recognized as part of SST are the social construction of technology (Pinch & Bijker, 1984), actor-network theory (Akrich & Latour, 1992; Callon, 1990; Latour, 1987) and the systems approach (Hughes, 2012).

In this thesis, SCOT is adopted and its key concepts are integrated into an analytical framework that enables engagement in the social shaping of smartphones. The reason for choosing SCOT is twofold: first, as will later be explained in greater detail, SCOT rests on the notion of *interpretative flexibility*, which is at the core of developing the central argument of this thesis. Second, its tools are well-suited to both analytically dissect and empirically explore the complex process of social shaping.[10]

### *The social construction of technology*

The social construction of technology has become a major school in the SST approach. It was initially developed by transferring and applying conceptual tools and methodological approaches from the sociology of scientific knowledge (SSK) to the study of technologies. SSK focused on the process of development of scientific "truths" by identifying critical points of contingency when ambiguities arose. Through historical studies, it demonstrated that scientific theories do not emerge simply because they are *true* or *better* but rather because they are socially constructed and collectively supported in their particular social and historical context. This embeddedness of truth in the social circumstances of its acceptance was explored through the principle of symmetry. The principle of symmetry was developed by David Bloor (1973 as quoted in Bijker, 1993, p. 118) who maintained that the sociologist, when examining scientific knowledge, should evaluate true and false claims "symmetrically," i.e. through the same conceptual apparatus. The acceptance of any claim then had to be considered not through its content, but rather through the circumstances of it acceptance.

[10] In this thesis, actor-network theory is not adopted as a guiding theory. The reason why preference is given to SCOT, along with  concepts integrated from the critical theory of technology, is due to ontological considerations. This thesis is embedded in interpretivist studies, which is evidenced by the choice of methodologies for the empirical studies (see chapters 4 and 6). ANT is considered  less suitable for studies with an interpretivist framework (Cordella & Shaikh, 2003).

Pitch and Bijker (1984) argued that it was not the inherent qualities and performance of a particular technology (whether one solution or the other was inherently better) that was most important, but rather the process of its acceptance by relevant social groups. In other words, the winning technological design did not need to be understood inherently true or better, but rather as accepted by the relevant groups in their socio-historical context(s).

The core premise and primary contribution of SCOT was to look at the process of technological development as a *contingent* process, paying special attention to how the design of technologies was socially shaped and negotiated among relevant social groups.[11] The negotiability of technology is made possible due to the fact that technologies exhibit *interpretative flexibility*. This term was first introduced by Pinch and Bijker (1984) in their study of the development of the bicycle (see also: Bijker, 1997). Interpretative flexibility means that the same technological artifact can carry different meanings for distinct social groups (called relevant social groups). Hence, the problems and the solutions associated with a particular artifact may present themselves differently for these diverse groups. Exploring and demonstrating the interpretative flexibility of a technological artifact involves moving beyond that artifact's *identity*, its *working* and other intrinsic properties and paying attention to how these qualities are ascribed to the artifact by social groups who, in their turn, have different understandings and can construct a different meanings of the same technological artifact. In Bijker's words: "without recognising the interpretative flexibility of technology, one is bound to accept a technological determinist view" (Bijker, 2001, p. 27).

[11] In the late works of Wiebe E. Bijker, one of the founders of SCOT, a move more towards mutual shaping of technology and the call to 'politicize' the process is seen (Bijker, 2001). In my view, Bijker's version of SCOT came close to the position developed by Feenberg (1999, 2004, 2010) where he stressed the mutual shaping and the need for democratic rationalization.

*Relevant social groups* is the second most important concept. It denotes any organized or non-organized group(s) of individuals who share the same patterns/sets of meanings associated with a technological artifact. Relevant social groups can be identified as institutions, organizations or networks of individuals. Identifying relevant social groups is the task of the researcher, who needs to follow the actors involved in the negotiation process of a specific technological artifact (Bijker, 1997; Bijker & Law, 1992; for critical response see: Russell, 1986; Humphreys, 2005). Most critiques aimed at improving and expanding SCOT have focused specifically on the notion of relevant social groups and, as will be elaborated on in the coming section, have called for introducing the factor of power asymmetries. In this thesis, the concept of relevant social groups is adopted in order to (help) differentiate software (app) developers as a group involved in shaping smartphones.

The interpretative flexibility of technology is not infinite or unlimited because the negotiation process is accompanied by stabilization and closure. Stabilization increases as interpretative flexibility decreases. Hence, as more members of the relevant group (or more relevant groups) come to accept and share collective meanings about the artifact, those meanings become more homogeneous and a higher degree of stabilization is achieved. Stabilization leads to closure**.**

The *closure* is a concept borrowed from the sociology of scientific knowledge (SSK) to articulate how, following a scientific controversy, a consensus emerges in an academic community by redefining and "closing" the controversy. Following such closure, new research does not revisit the controversy but rather draws on the new set of propositions. Similarly, in the case of technological negotiation, closure is achieved when a consensus among the relevant social group(s) is reached. However, once it is achieved it is "almost irreversible – almost, but not completely" (Bijker, 1993, p. 122). Building on Pinch and Bijker's work, Humphreys (2005) further developed the concept and suggested using *temporary closure* as a better descriptive term. The concept of closure has been critically reevaluated by Mackay and Gillespie (1992), who argue that, in its original conception, the concept accounted neither for the persuasive power of marketing, nor for end-users' creative appropriation of them. Thus, they stress the need to expand the SST approach to include “questions of both ideologies in technology design and marketing, and the subjective social appropriation of technologies into the debate” (p.709). In this thesis, I embrace the critique by Mackay and Gillespie (1992) and argue that, by introducing the notion of *design code*, the formal biases in the design of an artifact can be explored and explicated.

In addition to the possibility of closure, the social shaping of technology also acknowledges that some choices may be irreversibly foreclosed, even before relevant groups engage in negotiating the possible outcomes (Williams & Edge, 1996). Though these irreversibly foreclosed choices are less explored in the scholarship, I will argue that they are nonetheless extremely relevant to gaining a thorough understanding of alternative and potential future paths.

### *Limitations of the social construction of technology*

> “[…] the strength of SCOT is also its weakness, as highlighting the complexity of technology design can lead very easily to cloaking the simple and obvious.”
> (Prell, 2009, p. 42)

In the earliest critical response, Russel (1986) argued that SCOT, on the one hand, made a very important contribution in disproving linear models of technological development and going against 'black box' treatment of technology. Black box treatment in this context describes the absence of insight into the process of design and development and perception of technology as a mere outcome of application of scientific knowledge to engineering. On the other hand, the approach also had shortcomings in need of improvement. First, Russel criticized its relativism. [12] Though he acknowledged that it was useful as a methodological-analytical strategy, “relativism as a method might have slipped into relativism as a position on conflicting viewpoints. Such substantive relativism would collapse into political neutrality with respect to different social interests” (pp. 332-333). As a solution, he proposed grounding SCOT not in a priori relativism, but rather

[12] See also Christians (2016) concerning the criticism of relativism for the study of ICT technologies.

by starting "from a political commitment to demonstrating the possibility of alternative technologies for alternative goals, and to opening up the process of technological development to sections of society denied access to it" (p.333). In this thesis, this shortcoming will be addressed by including an explicitly normative dimension by introducing the notion of democratic rationalization. In addition to noting the problematic absence of any political commitment, Russel also argued that the SCOT approach had an "inadequate conception of social groups". In his view, relevant social groups had to be located within a broader structural context. In failing to locate and contextualize relevant social groups with respect to their own economic, political and ideological constraints and influences, SCOT was missing an important explanatory mechanism about the connection between relevant groups and their social interests. He called for SCOT to expand its analytical tools based on the already existing scholarship on social analysis in the Marxist tradition (p. 343).

Later criticism developed by Winner (1993) largely agreed with the arguments voiced by Russel, primarily the dissatisfaction with the analytical conception of *relevant social groups*. For Winner, the key question was how relevant social groups were identified and, more importantly, who was suppressed and deliberately excluded from them. Winner problematized “disregard for the social consequences of technical choice” (p.368) and stressed the need for researchers to engage with the broader consequences of choices and not merely treat those as winning solutions. The “[…] apparent disdain of anything resembling an evaluative stance” (p.373) resulted in what Winner figurately called opening up the black box but finding it empty and hollow.

Thus, Winner (1993) expressed concerns quite similar to those voiced by Russel. They both applauded SCOT’s research aims of critically examining how particular design choices were constructed/negotiated, but they also both criticized the lack of analytical tools to adequately explore reasons for the disagreements between relevant social groups during this process. It is in this context that Hård (1993) offered an alternative to what he called the consensus-oriented social constructivist perspective and challenged the idea that negotiation was at the root of the development of any technology. He argued that conflict, rather than consensus among relevant social groups, offered the better explanatory frame.

In the wake of further developments in the criticism of SCOT as well as attempts to expand it, Klein and Kleinman (2002) problematize the absence of any explanatory mechanism for how closure is achieved (i.e. how consensus emerges) and how one group's interpretation wins over the others. They contend that what is missing from the analysis are concepts that enable discussion of the differential/unequal *capacity of power* of the relevant social groups. They further suggest that a group's power capacity should be understood within broader organizational and structural terms. In other words, they advocate bringing to the fore the structural limitations in the analysis, and exploring how power asymmetries among the relevant actors may influence the negotiation process (e.g. by pushing for closure). They write that "an adequate understanding of the limits of interpretative flexibility, stabilization, and closure requires attention to power asymmetries" (p. 35).

The previously discussed critical inquiries were all strictly theoretical. In contrast, Prell's (2009) argument to *rethink* SCOT is grounded in the context of her empirical studies of the development of a particular technology - the Connected Kids project.[13] Through her work, she too sheds light on SCOT's shortcomings. She maintains that SCOT provides a very useful "analytical vocabulary for making sense of the interactions that surround and give rise to a particular artifact", but that its analytical tools may also "lead to cloaking the strongest influences in a technology's design" (2009, p. 36). Her criticism is primarily directed at the concept of technological frames in accounting for the structural constraints within relevant social groups. She maintains that by only looking at the technological frames of the relevant social groups, the

[13] Connected Kids was a service and website for youngsters in New York, built through collaboration with local government, youth-service organizations, the Rensselaer Polytechnic Institute (RPI), the University at Albany and the State University of New York (SUNY).

analyst may possibly fail to recognize even more influential forces at work, namely investment (money), knowledge and access to resources.

In summary, criticisms of SCOT have primarily concerned its limitations and inadequacies with respect to exploring asymmetries of power between relevant social groups, as well as to capturing the interests of structurally marginalized or excluded groups in this process. With respect to solutions for how to improve the SCOT, Russel (1986) suggested embedding it in social analysis primarily in the Marxist tradition. Winner (1993) called for a cross-disciplinary discussion of how to orient the technology towards democratic and environmental principles. Klein and Kleinman (2002) proposed situating SCOT's analytical tools within structural terms from organizational sociology and political economy. In what follows, I expand on the work of the authors and propose addressing SCOT's shortcomings by integrating insights from critical constructivism (a.k.a. the critical theory of technology).[14]

## Critical constructivism

> "…technology is not merely the servant of some
> predefined social purpose; it is an environment
> within which a way of life is elaborated"
> (Feenberg 2010, p. 15)

Within the context of the humanities, Andrew Feenberg (1999, 2002, 2010, 2017) developed a critical theory of technology by revisiting the Frankfurt School, most notably the works of Max Horkheimer and Herbert Marcuse (Feenberg, 2005b) as well as the second generation theorist, Jürgen Habermas (Feenberg, 1996). Feenberg's work aims to provide an explanatory framework that is critical but that avoids essentialism. In other words, to develop a critical theory without totalizing determination. As I will argue in the coming review,

[14] A comparable theoretical transition is proposed by Söderberg (2011, pp. 29–40) in his study of hacker culture and practices.

his insights into the politics of the design of technology provide a promising vista/horizon for the expansion of SCOT.

What makes Andrew Feenberg’s contribution so significant, in my view, are the implications of his analysis. It provides an alternative to both social and technological determinism and enables us to maintain a critical perspective without falling into essentialist critique. Feenberg’s critical constructivism is a form of critical theory, embedded in and developed upon revisiting the Frankfurt school, most notably its insights on rationality, which he reformulates. The Frankfurt School became famous by developing a critique of rationality that asserted that the modern form of rationality was inevitability leading to domination. ‘Progress’ and technology were a concretized form of this very (modern) rationality, which was doomed to lead to total control instead of the liberation of humanity. The seminal work in developing this critique of rationality in relation to technological progress was Marcuse's One Dimensional Man (1964/2013), in which he argued that modern rationality promised freedom but delivers total domination. Feenberg (1995, 2005), while largely agreeing with Marcuse’s view, admits that Marcuse failed to elaborate a proposal on how to overcome the limitations (or domination) of modern rationality. Feenberg does not support withdrawal from modern society, or romanticizing pre-industrial life, but rather stresses the need to bring the discussion about rationality and technological progress back into the academic debate and recover human agency in ‘rationality.’ That is to say that, through his work, Feenberg calls for the “reformulation of rationality, the role of technology in society, critique, and progress” (Wolff, 2019, p. 174). This is achieved by deconstructing the technical rationality as such, revealing its contingency and introducing democratic rationality as a potential, and much-needed, substitute for technical rationality.

### *Formal and substantive bias*

One of the most important contributions of the critical theory of technology which can also complement the social shaping of technology perspective, is a nuanced elucidation of how values can be ingrained in technological design: an idea that pushes back on the notion of “neutral” technologies. Like scholars in the SST tradition, Feenberg too acknowledges the interpretative flexibility of technology and starts with the possibility of the negotiability of technology (Feenberg, 1999, 2010). At the same time, however, he argues that despite negotiability among relevant social groups, the very procedures of these *negotiation processes* of technological decision making are not neutral, but rather incorporate the biases of the broader socio-economic system.

In the introduction to this thesis, I presented Feenberg's argument that bias and neutrality can coexist in one artifact (or technological system). Feenberg explains how this is possible by introducing two concepts (from Weber's work): *substantive* and *formal bias* (1996, 2002). *Substantive bias* relates to value judgments that are unfair, encountered in the context of lived experience and that discriminate based on prejudicial criteria (e.g. racial, ethnic, gender or other types of prejudice). *Formal bias*, on the other hand, is a value judgment or bias that has been translated into a rational process or solution, which does not appear biased anymore. Feenberg offers two examples. The first is tools for right-handed people. There is no substantive prejudice against people who are left-handed, yet right-handed tools embody the implicit bias inherent in very neutral-looking artifacts (e.g. scissors). A second example is a culturally-biased test that, based on its composition (e.g. the types of questions posed and the forms of knowledge necessary to answer them) favors one group over another. Though such tests may be fairly administered (e.g. individuals, or an organization or even an entire institution, administering the test may not have the same culturally-biased assumptions and can thus proceed in a fair manner, without any prejudice and intention to discriminate against one ethnic group over the other), the bias is already built into the test itself. Feenberg (1996) calls such an inequity a formal bias, in contrast to the more palpable substantive bias that we commonly perceive in the lifeworld. Formal bias is a consequence of the formal properties of the biased activity, not of substantive value choices.

Feenberg uses formal bias to refer to values incorporated in artifacts as the "materialization of interest and ideologies in technical disciplines and design," which show that "technically rational objects are actually value laden" (2017, p. 56-57). However, once established, such artifacts and the process of their working seem obvious, inevitable, often as simply a technical necessity. It is comparatively easy to demonstrate substantive bias. Formal bias, however, is much more subtle and difficult to oppose.[15] Because formal bias is not plainly visible in the context of a social system, it comes across as fair treatment within a given range of standards. However, the bias is already implicit within the very standards through which the efficiency or working of a particular solution are judged.

Feenberg further demonstrates that formal bias can be identified at the micro-level of a single technological artifact, or on a wider, macro-level scale. One example Feenberg offers for demonstrating the formal bias of a broader system (namely, economic organization) is Marx's critique of capitalism. The pinnacle of Marx's critique of capitalism was to demonstrate how equal exchange (governed by the rules of the rational market) was leading to accumulation and inequality. Workers were receiving wages which were purportedly equivalent to their worth on the labour market. In that sense, there was no substantive bias against the workers, yet the system clearly benefited one group over the other, leading to capital accumulation and thus to inevitable cyclical crises in capitalism. In much the same way, when discussing the manufacturing of smartphones in the context of the global labour market (chapter 3), wages given to workers in sweatshops are purported to be in accordance with the value of their labour in that specific market. Thus, the allocation of wages in that context seems 'fair'. What this means is that formal bias will only be made problematic (and visible) when we discuss it within a broader social context and consider alternatives outside the range of allowed possibilities. In other words, when one looks at the working of an assembly line it appears as a rational, neutral and very efficient technology. However, when we examine it within the context of the deskilling of labour,

[15] Substantive and formal bias can be used to capture prejudice beyond technological choices. An analogy with what ignited the BlackLivesMatter movement comes to my mind: blatantly prejudiced acts in daily lived experience led to the protests and were readily identified as bias. On the other hand, much deeper and complex institutionalized forms of formal racial bias, which are far more difficult to reveal and combat, require a whole new approach: one that must necessarily critique our most basic and foundational notions of justice.

for example, we are better able to recognize and evaluate its formal bias. In the same way, I will be arguing later in this thesis that *datafication is a formal bias* which, while treating each nod (user) fairly and without discrimination in the network, simultaneously embeds *bias* and contributes to advancing the particular interests of some at the expense of those of others.

### *Design code(s)*

The second analytical tool that is useful in exploring the politics of the technological choices and sedimented values embodied/inherent in design is the notion of *technical* or *design code*[16], which aims to capture such formal biases of technologies. As Feenberg defines it: ”a technical code is a rule under which technologies are realized in the social context”, and it works at the intersection of “ideology and technique” (2010, p. 68-69). More precisely, “a technical code is a realization of interest or ideology in a technically coherent solution to a problem” (2010, p. 21).

Technical codes are usually invisible because, like culture itself, they appear self-evident. Nonetheless, they serve as “a criterion to select between alternative feasible technical designs in terms of a social goal and realize that goal in design” (p. 68). “Feasible” in this quote refers to technically working solutions (technology simply should work), and social goals are not universal criteria but rather widely-held values (e.g. safety, health). For instance, in a classic example of negotiating the interpretative flexibility of bicycles, Pinch and Bijker (1984) demonstrate that the artifact started as two different devices: as a sportsman’s racing device and as a general transportation device. The design corresponding to the first device had high front wheels (speed was of great value) and was rejected under the rationale of the technical code of ‘safety.’ Accordingly, Feenberg argues that the “safety” design won, and that the entire, subsequent history of bicycles (whether for racing or transportation purposes) stems from that line of technical development (1999, p. 78-80).

[16] Feenberg used the term “technical code” in earlier versions of his theory up to 2015. In more recent works, he has substituted technical code with “design code.” Both terms are used interchangeably.

Whereas the development of the bicycle is “reassuringly innocent” Feenberg argues that many technical solutions have more serious political implications and profound effects on the distribution of power and wealth, such as assembly line (1999, p.80).

Crucially, technical codes are always biased to some extent with respect to the values of the dominant actors and reflect the unequal distribution of social power. For example, how efficiency is measured through profit-maximization in a capitalist society can be seen as an illustration of formally biased in measurements. However, subordinated groups may yet challenge the technical code and thereby influence the development of technologies. It is important to note that Feenberg maintains that technical development is not definitely towards any particular path. On the contrary, it forms  a branching pattern. The final determination is *not* inscribed in the technology itself, but rather emerges through meaning-making. Feenberg further differentiates between the *function* and *meaning* of technology. The function of technology denotes the prescribed “affordances” (Hutchby, 2001) without reference to their actual social use. Affordances are “functional and relational aspects which frame, while not determining, the possibilities for agentic action in relation to an object” (ibid., p.444). These often deviate from the actual social use. The full meaning of technology is realized through the specific context of its usage.

### *Democratic rationalization*

The normative dimension in critical constructivism implies strengthening technical codes that are in accordance with the public’s interest(s). This is what Feenberg calls *democratic rationalization*, signifying it as means of instilling public agency into technocratic decision-making processes. Feenberg argues in favor of pursuing an openly normative approach due to the fact that, in capitalist society, the developmental path of technologies was dictated for generations by the pursuit of efficiency and deeply ideological practices restricting opportunities to participate in decision-making (Feenberg, 2010). To give one more illustration, the operational rationality of capitalism presupposes the separation of workers from the means of production. This highly subjective, material and historical aspect of private capitalist enterprise became a formal, structural element and the basis of what was considered to be “rational” in economic activities themselves.

Likewise, the concept of efficiency became an unquestioned ideological justification for certain types of strategic decisions. However, the problem lies in how efficiency is conceived. From an environmentalist perspective, efficiency would entail quite different measures than under capitalism. It is through such justifications that the assembly line became a progressive form of technology that has increased the profits of company owners. The fact that this happened at the expense of de-skilling workers is of little importance since the primary measure of efficiency does not account for worker participation. Consequently, the interests of particular groups (the owners of the means of production) ultimately solidify into the very structure (hence, become inscribed as formal bias) under which choices about which technologies to develop are legitimated.

*Democratic rationalization* aims to contribute to the goal of subjecting the development of technology to democratic processes. This suggests an alternative form of rationalization based on assuming responsibility for human and natural contexts and the values of equality and participation, and not merely on economic efficiency (with a narrow understanding of maximizing profits). In the context of new information and communication technologies, the agenda for the democratization of technologies focuses primarily on the struggle over the structure, access and ownership of communicative networks and practices.

### *A glossary and visualization of the analytical tools*

In Table 2, I provide a glossary of the major  key concepts that were reviewed in this chapter. Each concept is given with a definition and original sources. Major critical revisions, and empirical applications are also listed.

**Table 2** Key concepts used in the thesis with primary sources and critical revisions

| Concept | Definition | Primary source | Criticism | Application |
|---|---|---|---|---|
| **Interpretative flexibility** | range of different meanings that relevant social groups construct about the same technological artifact | (Pinch & Bijker, 1984) | (Orlikowski, 1992; Doherty, Coombs, & Loan-Clarke, 2006) | (Ahmed, Qureshi, & Khan, 2015) |
| **Relevant social groups** | organized or non-organized networks of individuals who share the same understanding of the technological artifact in question | (Pinch & Bijker, 1984) | (Humphrey, 2005; Van Baalen, van Fenema, & Loebbecke, 2016; M. Wilson & Howcroft, 2005) | (Pense & Cutcliffe, 2007; Lassinantt, Ståhlbröst, & Runardotter, 2019; Prell, 2009) |
| **Closure** | when a consensus about the meaning of a technological artifact emerges and stabilization is reached (*can lead to irreversibility - a point where alternative possibilities are foreclosed) | (Pinch & Bijker, 1984) | (Boyd & Schweber, 2018; Richardson et al., 2006; Khoo, 2005) | (Boyd & Schweber, 2018; Siles, 2012) |
| **Technical code (a.k.a. Design code)** | are rules and criteria to select between alternative feasible technological designs and represent an intersection of ideology and design | (Feenberg, 1991, 2002, 2010, 2017) |  | (Flanagin, Flanagin, & Flanagin, 2010; Hamilton & Feenberg, 2005) |
| **Formal bias** | is when rationally designed, coherent and apparently properly functioning technological device or system has | (Feenberg, 2010) | (Kirkpatrick, 2013, 2020) |  |

| | | | | |
|---|---|---|---|---|
| | discriminatory consequences | | | |
| **Democratic rationalization** | aims to subject technological control and design to democratic processes | (Feenberg, 2002, 2004, 2005a) | | (Jesiek, 2003) |

In Figure 1 visualization of this conceptual model is presented. Programmability is placed as the starting point of my analysis. The programmability then is linked to code/software's inherent qualities and is placed within negotiation process. Relevant social groups are denoted, and the potential difference in their power dynamics are acknowledged. The closure and its potential connections to the power dynamics of involved actors are also indicated.

To summarize, throughout this chapter, I elaborated the theoretical framework and rationality of this dissertation. I chose the social construction of technology, which provides useful concepts to explore how different relevant social groups shape a new technological device. I reviewed and pointed to the limitations of SCOT and proposed its expansion by integrating conceptual tools from Feenberg's critical constructivism.

In the following chapters I apply the analytical tools elaborated in the conceptual model and use notions such as relevant social groups, closure, formal bias and technical code to provide insights into how the demands of relevant social groups are negotiated and inscribed into the technical configurations of smartphones.

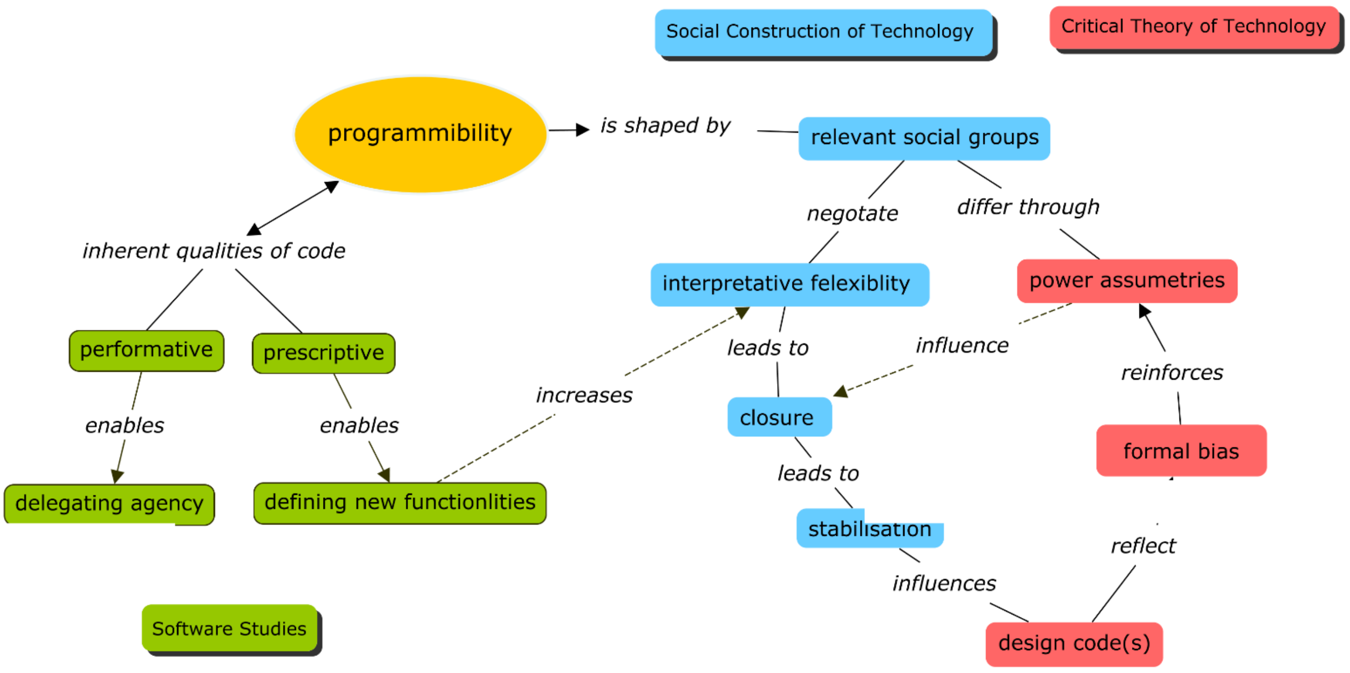


**Figure 1** Conceptual model to study programmability

# Chapter 3 Demarcating Smartphones

## The politics of code and networks[17]

The first converged devices called *smartphones*, with a computer, cellular phone and internet capabilities combined in one artifact, were sold in 2003. Somewhat earlier, in 1999, a comparable system called iMode, was introduced in Japan and was widely accepted (Daliot-Bul, 2007). In the West, however, the release of the iPhone in 2007 is commonly considered to be the decisive moment in popularizing smartphones (Goggin, 2009; West & Mace, 2010). Goggin (2009) described how Apple marketed the iPhone as a "breakthrough" with cell phones and a paradigm shift in mobile design. Similarly, Burgess (2012) argued that the iPhone came to symbolize a distinctive "moment" in the history of cultural technologies, one that rearticulated the values of personal ICT technologies and continuous connectivity. Campbell and La Pastina revealed how the cultural context of the introduction and marketing of the iPhone was steeped in contemporary American popular and Christian culture (2010). Through an analysis of the blogosphere and mainstream media coverage of the first iPhone, the authors illustrate how religious imagery and pop culture icons were evoked in both the media and popular discourses. In particular, the authors (Campbell & La Pastina, 2010) traced the origin of the term "Jesus Phone", which came to be used by bloggers in reference to  the iPhone in early 2007.  It spread not only in the blogosphere, but was also reiterated by mainstream media in the US.

Interestingly, the first iPhone came only with pre-installed apps. Within just two months of its release, George Hotz, 17-year-old student from New Jersey, unlocked the iPhone. His primary goal was to bypass the limitation of being tied to AT&T. He documented the process and shared his method publicly by

[17] Parts from this chapter were printed in Mosemghvdlishvili, L. (2015). Mobile internet: The politics of code and networks. In A. Bechmann & S. Lomborg (Eds.), *The Ubiquitous Internet: User and Industry Perspectives*. New York: Routledge. doi: 10.4324/9781315856667

uploading the video on YouTube. Hotz's method involved opening the iPhone, replacing the SIM card, soldering it and erasing the original firmware to connect to an alternative mobile network (Scelsi, 2007, p. 19).

Apple responded by releasing an update to undo such attempts at unlocking the phone. Following the software update, devices which were unlocked became non-functional and many native apps were rendered inoperable. The move swiftly transformed these devices into "iBricks" and led to the famous criticism by Zittrain (2008), who feared that tethered instead of generative devices would fundamentally alter the future of the internet. Hotz was compromising the exclusive deals between Apple and mobile network carriers, as it was only available for people in specific countries with specific subscriptions.[18] Following Hotz, 27-year-old computer science doctoral student Jay Freeman attracted widespread attention for jailbreaking an iPhone and making an alternative app store called Cydia. The alternative store soon contained more apps than were available in the preinstalled version on the iPhone. Soon after, some 1.7 million users had also jailbroken their iPhones (Magaudda, 2010).

Arguably due to increased instances of jailbreaking and the availability of more functions (via unauthorized apps) for the iPhone, Apple began allowing developers to write software for iPhones, under the condition that it be exclusively distributed through its app store. Soon other OS providers replicated the same model of opening their platforms to third party developers while maintaining distribution through their own channels. In 2008, the first app store on the iPhone listed less than 500 apps (Wade Morris & Elkins, 2015). By 2019, the same store listed up to 2.2M iOS apps. Google Play listed somewhat more: 2.8M Android apps (Han et al., 2020).

[18] Apple could not pursue legal charges against George Hotz. The circumvention of copyright protection was punishable under the Digital Millennium Copyright Act, but in 2006 the Copyright Office had created a three year exception which allowed users to unlock their phones (meaning connect it to alternative carrier) for personal purposes only  (Scelsi, 2007, p. 9).

At this juncture, it is necessary to shift our focus for the remainder of this chapter from exploring apps for smartphones toward a critical examination of some of the most relevant aspects of the manufacturing of these devices. A mere glance at branded phone shops such as Verizon, Sprint and AT&T in the developed countries, or second-hand repair stalls in busy markets in developing countries, presents us with an overwhelming range of options with respect to brands, models, and (non-branded) modifications. [19] This overwhelming abundance of choice is reflected in the proliferation of both high-end and low-end (cheaper) smartphones. Despite the abundance of competing models, when smartphones are taken apart and their parts are tracked, or cross-licensing is examined, for example, a very different, and much more homogenous, picture emerges. Competing firms exploit the same geography and circuits of global manufacturing and are often characterized by complex inter-firm relationships.

## Intellectual property and smartphones

If a single state of affairs were to be chosen as most characteristic of the first phase of smartphone development, it would be the tremendous increase in patent lawsuits among key companies. Since 2011, several expensive legal disputes over patent infringements between giants like Apple and Samsung (Edwards, 2013; Parish, 2011), Google and Oracle, and Microsoft and Motorola became the subject of intense media interest. Subsequently, terms like “patent wars” and “smartphone wars” appeared not only in the blogosphere and Wikipedia, but in academic work as well (Paik & Zhu, 2013, 2016). The number of court cases rose so rapidly that even the most modest assessments of the scale of patent litigation between corporations seemed bewildering. The following estimate captures the magnitude of patent turmoil when it comes to smartphone design: since 2010, $ 20 billion has been spent

[19] The appearing abundance is closely connected with short life span (and usage period) of mobile devices, producing e-waste at alarming rates (Huang & Truong, 2008; Paiano, Lagioia, & Cataldo, 2013).

on patent litigation alone in the mobile phone industry (Cohan, as quoted in Paik & Zoe, 2013).

There are 250,000 active patents, with respect to smartphones alone. This means that, in the US, one out of every six active patents applies to these devices (O'Connor, 2012). This is a very important detail that helps reveal that smartphones are not simply new technologies; on the contrary, they rely on existing technologies. One explanation for this dramatic increase in patent litigation is that smartphones are *cumulative* innovations (Shapiro, 2000), meaning that they are built on previous discoveries. Shapiro (ibid.) problematized the use of patents in the context of complex ICT technologies. As he explained, new product innovations in this domain would inevitably infringe on existing patents because they were built on existing, previously-patented innovations. This would lead to patents impeding, rather than promoting, innovation. effectively turning them into a strategic business tool (such as cross-licensing). Cross-licensing refers to a rather counterintuitive practice whereby rival companies rely on each other's patents. For example, when an HTC smartphone which runs Android (Google's OS) is sold, its competitor Apple collects between $6 and $8 for each device. Similarly, Microsoft earns as much as $5 for each LG, Acer and Samsung phone sold (Koetsier, 2012).

Another case where the negative effects of intellectual property (namely patents) intersect with the shaping of smartphones is with respect to network connectivity. It is possible to connect a mobile device to the internet via various access technologies. Among these, the two most predominant are so-called mobile broadband (3G, 4G) and wireless broadband (Wi-Fi). Although both Wi-Fi and 3G enable internet access, they differ in terms of service, industry, architectural origins, and even philosophies. In a comparison of Wi-Fi and 3G technologies as two alternate development paths for accessing the mobile web, Lehr and McKnight (2003, p. 353) point out that "3G offers a vertically integrated, top-down, service-provider approach to delivering wireless internet access; while Wi-Fi offers (at least potentially) an end-user-centric, decentralized approach to service provisioning".

Moreover, the deployment costs of 3G (and 4G) were considerably higher than for Wi-Fi (or WiMAX, a successor technology to Wi-Fi). Nevertheless, 3G was still the favored option for mobile network providers due to its top-down and centralized architecture. However, in order to use 3G, both mobile network providers as well as mobile device manufacturers need to comply with a set of essential standards, including the Third Generation Partnership Projects (3GPP and 3GPP2).

These standards are set[20] within the industry by private companies which collectively hold a substantial share of the market (Lemley, 2002, 2007). In the case of 3G, the corporate members of these partnerships have identified up to 8,000 patents that they have declared to be essential for 3G standards, 90% of which are owned by just 13 companies. The following four firms account for 30% of these patents: Qualcomm, Ericsson, Nokia and Motorola. Despite being declared as essential, a study by Goodman and Myers (2005) revealed that only 21% were actually vital for 3G technologies.

One salient outcome resulting from this is that companies that manufacture equipment or offer services for third-generation cellular systems pay royalties for 79% of the patented technologies, which are excessively broadly defined as being essential by patent owners  (Goodman & Myers, 2005). Even though the companies that own patents for standards are required to license them on fair, reasonable and non-discriminatory terms, in reality royalties “tend to be higher than the benchmark level” defined by law. In the case of 3G phones, this amounts to an increase of 30% of the total price of each phone (Lemley & Shapiro, 2007, p. 27).

[20] There is also the possibility of *de facto* standardization, which is when certain products (e.g. Microsoft’s operating system Windows) become widely accepted on the market; or standardization through government intervention, which sets the appropriate standard and compels all participants in the market to comply (e.g. the Federal Communications Commission; Lemley, 2002).

The relationship between standards and patents has received considerable attention in policy and innovation studies and has raised important questions regarding the implications of standards increasingly being patented (Bekkers, Verspagen, & Smits, 2002; Bekkers & West, 2009). The interests of patent holders, in contrast to the public interest, are problematized in research that has explored the GSM standard-setting procedure. Citing the example of Motorola (Bekkers et al., 2002), the study's authors illustrate how patent ownership enabled the company to define the terms of the standardization process. In light of this, they argue that "at the level of the public interest in standard-setting, the increasingly sharp negotiations about essential IPRs are not necessarily a positive development". Lemley (2002, p. 1990), who focused specifically on the process of standard-setting in the ICT sector, also argued that when standards are owned by a just few of the largest companies, rather than promoting competition, patent owners instead act "as a cartel with the power to reduce output by excluding certain kinds of products" (p. 1900). The problem of excessive patent use has also been criticized by others as a source of profit increasingly outweighing social benefits (Bessen & Maskin, 2009; Bessen & Meurer, 2008; I. M. Cockburn & MacGarvie, 2009; Coriat & Orsi, 2002).

In summary, the first period of the smartphone industry was accompanied by the increased use of patents, which are a decisive factor in the distribution of profits. This had profound implications because how the chain of value is created and, more importantly, how value is divided is principally organized through intellectual property. This points to an argument advanced by Feenberg (2017), who asserts the need to discuss technological decision making and the politics of value together with patterns of ownership (market structures) and administration. What is also evident from the excessive use of patents is how the shaping of smartphones is deeply embedded in and mediated by the structures and enforcement of intellectual property rights. The greater implications of intellectual property in relation to software will be discussed further in this thesis (see Chapter 6).

## Circuits of labor

In the context of mobile devices and digital technologies in general, much attention has been directed toward exploring and explaining new forms of labor. Such new labor forms include *immaterial labour* (Lazzarato, 1996; Terranova, 2006), *audience commodification as a source of value* (Fuchs, 2012; Kang & McAllister, 2011; Nixon, 2014) and the *precariat of creative industries* (Kücklich, 2005; Lund, 2014). These new forms of labor, which have come into being as a result of digital (mobile) technologies, are increasingly recognized as being characteristic of a new, post-Fordist stage of capitalism known as informational capitalism.

Despite increased attention to these forms of immaterial labor in the current scholarly literature, I would like to shift focus instead towards the manufacturing labor necessary to produce smartphones. Namely, when we look at the global value chain of a smartphone we see that the least costs are spent on labor for manufacturing.

Scholars from organizational studies (Ali-Yrkkö, Rouvinen, Seppälä, & Ylä-Anttila, 2011; Dedrick, Kraemer, & Linden, 2011; Delautre, 2017; Kraemer, Linden, & Dedrick, 2011) have attempted to 'tear apart' the global supply chain of smartphones in order to analyze which actors capture the most value. These works reveal a picture that seems very counterintuitive yet is nonetheless normalized. Manufacturing, assembly and initial labour costs are located in the East Asian countries.[21] However, most value is captured through brand owner firms. For example, Apple, through its high-wage functions,

[21] by 2020, there is an observable trend to move production from the East Asian countries such as China and Taiwan to Vietnam, and India. For example, Samsung, the world's largest manufacturer of smartphones, completely closed its plants in China and announced the decision to move to Vietnam and India. Changing regulations, as well as the increase of Chinese manufacturers, were cited as reasons for the move (Park, 2019). Likewise, Apple Inc. has announced that it will be producing new models of the iPhone SE 2020 in India in a strategic move to escape increased tensions between Washington and Beijing (Singh, 2020).

design and marketing, as well as through its ownership of patents, captures 58.5% of the value of every iPhone sold. In contrast, producers of input materials receive just 21.9% of the value and total labor costs amount to less than 6% (for the full distribution see: Kraemer et al., 2011). An earlier study of the Nokia N95 showed a comparable geographic dispersion (Ali-Yrkkö et al., 2011) and percentage of profit captured by the Finnish conglomerate.

If we consider these facts from the perspective of a Marxist analysis of capital accumulation and critically examine the source(s) of this capital accumulation, then it becomes clear that, despite the increase of immaterial labour (for example, profit generated through consumer surveillance or rent through patents), low cost material labor still function as a significant and indispensable part of informational capitalism. Examining the costs of material labour may seem an insignificant (due to the fact that it is the smallest portion in the value chain), but I contend that it is vital to acknowledge and include in the analysis.

Along these lines, Ursula Huws (2014) argues that the existence of a separate sphere of non-manual labor is not evidence of some "immaterial" realm of economic activity, but simply "an expression of the growing complexity of the division of labour, [where] the fragmentation of activities into separate tasks, both 'mental' and 'manual', increasingly capable of being dispersed geographically and contractually to different workers who may be barely aware of one another's existence" (p. 157). Fundamentally, all digital labor, as well as usage of mobile devices relies on the physical labour necessary to manufacture and transport these devices and dispose of the e-waste (Smith, Sonnenfeld, & Pellow, 2006). Works exploring such material aspects of creating mobile technologies and other ICT tools stress the critical need to investigate aspects such as mining minerals, manufacturing and assembling devices, as well as dismantling, and disposing of the e-waste as much an important and integral part as design, software development and digital labour.

Since 2010, reports of severe conditions in the "circuits of labour" (Qiu et al., 2014) across the value chain of manufacturing smartphones have been resurfacing in the media, human rights organizations' reports and scholarly works. Disturbing images emerge when mobile devices are traced back and attention is shifted to these critical aspects of the global value chain. (Brophy & De Peuter, 2014; Harvey & Krohn-Hansen, 2018; Qiu, 2018; Qiu, Gregg, & Crawford, 2014; Sandoval, 2016; Sandoval & Bjurling, 2013).

The most extreme cases relate to mining the minerals necessary for the production of mobile devices and other electronic goods. Mining for the basic elements needed to create these products has been traced primarily to African countries, mainly the Democratic Republic of Congo and Zambia (ACIDH, 2011). Practices such as child labor, forced prostitution, extremely unsafe working conditions, and low payment are among the major concerns. The Free the Slaves report (2013) explored the situation in the three most massive mines in the Democratic Republic of Congo, which is the largest producer of cobalt (one of the minerals used in producing mobile phones). The field report was based on interviews with 742 mine workers. Twenty-three percent of the workers interviewed were under 18 years old. Some of them started (and, in some cases, were forced) to work in the mines at as young as 11 years old (ibid, p. 15).

A comparable report by Amnesty International (2016) documented cases where children as young as nine years old were working in the mines, with severe consequences to their health. SwedWatch report estimated that over 50,000 children under the age of 18 were working in the mining sector (ACIDH, 2011, p. 9). As of 2019, a lawsuit on behalf of the parents of 14 diseased children as a result of working in the mines in the Democratic Republic of Congo was filed and is awaiting a hearing in Washington, D.C. Apple, Google, Dell, Microsoft and Tesla are all called as defendants in the lawsuit (Kelly, 2019).

In addition to child labor, reports studying the conditions of people living and working at the literal bedrock of the smartphone industry also document various forms of slavery (Amnesty International, 2016; Free the Slaves, 2013). These include debt bondage, sexual slavery and forced prostitution. Debt bondage, in which a worker must borrow money to purchase tools needed for mining, as well as to cover subsistence costs such as food, is reported to be the most widespread form. Due to the high interest rates charged (at times more than 30%) and low daily wages, many have little chance to pay the debt off. Adult male workers earn up to $2 for a full day of work; children earn even less: approximately $0,87 (ibid.). This form of debt is *inheritable*, meaning that when a worker dies, family members are forced to pay the debt and, in most cases, end up in similar conditions.

Amnesty International (2016) was the first international organization to trace and document the link between these extreme mining conditions and the manufacturing of electronic devices by the largest conglomerates such as Apple, HP and Samsung. This connection was also highlighted in a report published by the European initiative makeITfair, which conducted a five year investigation into the global supply chain of leading European mobile network carriers and suppliers. Their report concluded that mining under the above-described conditions was an integral part of the mobile phone companies supply chain (Huijstee & Haan, 2009) and that, even though firms admitted to being aware of these conditions, they reported that they had little demand from consumers for "fair" and "green" technologies. (Germanwatch, 2012). In more recent years, however, we have seen the emergence of new products and services that are rooted in the demands of environmentalists. A study conducted by Carberry, Bharati, Levy, and Chaudhury (2019) on the views and perceptions of corporate managers found that social movement demands represent an important input (as resource and pressure alike) for corporate social responsibility innovations. This provides a good illustration of Feenberg's (2010, 2017) discussion of how the demands of social groups become translated into new technical codes of an artifact. In the future, we may (and hopefully will) see more environmentally-friendly, green technologies become as necessary to technical specifications as safety is today.

It is not only mining for minerals that has been subject to severe criticism; but the processes of manufacturing smartphones as well. The most notable example was of Hon Hai Precision (traded under the name Foxconn), which is the biggest employer in the world involved in manufacturing and assembling branded smartphones. The company garnered international media attention due to the increasing number of workers' suicides at its facilities. In 2010, seventeen individuals between 17-24 years old attempted to commit suicide at the Foxconn factory. Only two survived: 18-year-old Rao Shu-qin and 17-year-old Tian Yu. As a result of the suicide attempt, Tian Yu was paralyzed below the waist.

Students and Scholars against Corporate Misbehavior (SACOM) has been documenting these cases over the past 10 years, resulting in a global campaign blending online and offline activism challenging digital capitalism (Pun, Tse, & Ng, 2019). The collective initially brought together 60 students and 20 scholars from various universities. SACOM became instrumental in collecting evidence from workers and making public the cases of extremely long working hours (60-70 hours a week), abusive control by managers and low wages (Fuchs, 2014; Qiu et al., 2014).

Conditions in the Foxconn factories in China did not represent isolated cases, but rather offered an apt illustration of general labor practices in China, as well as in Vietnam and the Philippines. For example, Sandoval and Bjurling (2013) reviewed independent reports demonstrating ample cases of comparable practices at Multec (a contractor of Sony, Motorola, and Phillips), Flextronics (a contractor of Microsoft) and Celestia (a contractor of IBM). They summarize their findings by stating that insufficient wages and hazardous working conditions, as well as the impossibility for workers to create unions and engage in collective bargaining, are some of the most worrisome structural problems.

In addition to critical investigations into the severe conditions in the mining of minerals and labor violations in smartphone manufacturing, the Centre for Research on Multinational Corporations' (SOMO) investigation into the mobile supply chain has also problematized the issue of e-waste. The short lifespan of smartphones and the push for ever-newer versions exacerbates the problem of e-waste. Not only precious metals but also hazardous parts are disposed of in developing countries (Huijstee & Haan, 2009; see also: Sandoval, 2016; Chen, 2016).

The unequal, global division of labor is not only characteristic of the mobile industry, but is equally visible in other sectors, including the fashion and garment industries, for example. Nonetheless, being aware of how information and communication technologies and networks are specifically embedded in and made possible through material inputs, such as physical and manual labor, is vital because it has significant repercussions for how value and profits are conceptualized with respect to smartphones. Unfortunately, a more nuanced discussion of what constitutes value in contemporary capitalism would take us far outside the scope of this chapter. For the moment, it is sufficient to note that a dynamic debate about what constitutes the source of value in contemporary capitalism has been discussed among contemporary critical political economists (Fuchs, 2010; O'Neil, 2015; Pasquinelli, 2009).[22] Rather than delving further into the details of this debate, I will now shift the discussion to another important site of struggle which has accompanied the shaping of technology and, more specifically, a clash of hitherto different organizational cultures and business models between relevant groups representing internet and telecom firms. That is the issue of net neutrality.

## Net Neutrality

The transformation of a mobile phone into *an access device for the internet* brought about the convergence of two hitherto very different industries: computing and telecommunications services. These industries were formerly considered to be distinct (Goggin, 2010; Zuckerman, 2010) because of their disparate structures, concentration of ownership, as well as organization of labor.

The government-owned telecommunication sector was traditionally centralized and monopolistic, but a massive deregulation process in late 80ies resulted in the establishment of regional and global network carriers.

[22] The disagreement can best seen across lines of two intellectual schools: Italian Operaismo (Antonio Negri, Michael Hardt, Franco (Bifo) Berardi) and Orthodox Marxists (Christian Fuchs. Mathieu O'Neil). The main thesis of Operaismo is emergence of post-Fordist, cognitive capitalism, where firms such as Google generate surplus value by extracting interactions, free content and free labour. See for more depth discussion on *rent* (Pasquinelli, 2009; Vercellone, 2008).

Ironically, deregulation, which was justified on the grounds of creating more competition and allowing new entrants to join the telecommunications sector, did not lead to a fundamentally different market structure (Wilson, 1992). On the contrary, the historically dominant carriers grew into new organizations providing mobile services as part of their integrated businesses (Goggin, 2010; Winseck, 1997). To illustrate, in the US, the offspring of the American Telephone and Telegraph Company, Verizon Wireless and AT&T Mobility, together captured 70% of the market share and 90% of mobile subscribers were tied to only four corporations (Statista, 2014).

Increased tensions between the telecommunications and internet industries were brought to light in the conflict surrounding *net neutrality*. Net neutrality is based on a principle that the internet is a decentralized and non-discriminatory network (Cooper & Brown, 2015; Wu, 2003). Non-discriminatory means no one, single authority has the power to discriminate in terms of content and information flows (Barratt & Shade, 2007; Meinrath & Pickard, 2006).

However, due to the emergence of and significant investments by mobile carries in 2G and 3G technologies, the attempt was made to change legislation concerning net neutrality. In the US, on January 14, 2014, the Court of Appeals for the D.C. Circuit ruled in favor of Verizon, the largest mobile network in the US. It struck down the Federal Communication Commission's (FCC) Open Internet Order, which prohibited Internet Service Providers (ISPs) from discriminating against content. The decision meant that network providers were able to engage in business deals with content providers and charge consumers differently for accessing various content (e.g. blogs and YouTube), and to discriminate in terms of content by artificially slowing down traffic (Levy, 2014). The court's decision became a source of controversy. The granting of gatekeeping powers to network owners was vigorously contested by digital rights organizations, namely the Save the Internet coalition. Within two weeks of the court's decision, more than a million users had signed a petition urging the FCC to intervene. Following much outcry, the FCC, with the support of (former) President Obama, classified broadband internet as a public utility, thus guaranteeing net neutrality. However, in 2017, after President Trump appointed Ajit Pai as the chairmen of FCC, the order was repealed. The new proposal published by the FCC on the organization's forum received more than 20 million comments. Despite the appeal to retain the 2015 Open Internet Order, the FCC ruled against its earlier decision and the new regulation went into effect in June 2018 (Collins, 2018).

The situation in the US is not exceptional; the same struggle is evident in Europe, where the EU Commission has been promoting a new proposal to reform the EU telecom market since September 2013. The interests of leading European telecom companies were strongly represented in the proposed reforms (Mangalousi & Mosemghvdlishvili, 2014). Activist groups were struck by the similarities between the reform proposal and the recommendations issued by the Economics and Technologies for Inter-Carrier Services consortium. This consortium was led by Alcatel-Lucent and was comprised of the dominant telecom operators such as BT, Orange and Deutsche Telekom. During the negotiation stage, to which the so-called CEO-roundtable representatives of the interested large private enterprises were invited, small ISPs, as well as advocacy groups, were excluded (Mangalousi & Mosemghvdlishvili, 2014). Further, member states differ in terms of national regulations. In the context of the Netherlands, for example, following a vigorous campaign led by the digital rights activist organization *Bits of Freedom,* the country became the second after Chile to pass the net neutrality regulation (Higgins, 2012). Comparable decisions in countries such as France, Belgium and Italy are still pending.

These struggles over net neutrality are still taking place at the regulatory level and help illustrate the complex interplay in which tensions between relevant social groups, who have a different understanding of internet, as well as interests, structure the terrain of (legal) possibility. In this case, the competing interests and differing interpretations of technologies by mobile network providers versus internet companies helps reveal their different incentives in promoting legislative changes. These regulatory mechanisms have a direct effect for end-users,[23] the operations of organizations and the direction of

[23] A comparable case took place in India when, in 2014, mobile service provider Airtel announced that it would charge consumers more for making calls through WhatsApp

technological development. Furthermore, the example of the US also illustrates the contingency of this process and its dependence on the current political climate.

## Conclusion

In this chapter, I have looked at evident points of tension between relevant groups involved in the shaping of smartphones and identified some of the most troublesome developments that were documented in both the academic literature and international organizations' reports. These cases span multiple dimensions, including intellectual property, labor, and access to the internet. Whereas the debate over net neutrality is manifestly political, the case of patents appears to be more subtle. As seen in the example of patents in 3G technologies, the private interests of patent holders are often neutralized in new standards. In this context, intellectual property directly contributes to the *design codes* that embody interests of powerful actors.

The technical codes of capitalism depend on the reification of intellectual property, thus discussing the current terrain of possibilities in which technologies developed is not possible without considering the peculiarities of treating forms of knowledge and information as property. What is currently legally safeguarded as intellectual property (particularly in the forms of patents) was considered here to instead be a range of privileges, more akin to welfare and the common good than property. Concrete legal and institutional changes led to the establishment of the "new IP regime" (see e.g. Bessen & Meurer, 2008; Coriat & Orsi, 2002; Coriat, 2000), when patentability was expanded to include new forms of information and knowledge (research on the human genome, software, and the so-called business models). However, unlike material property (which is naturally scarce), forms of information and knowledge are not scarce, but are *made scarce* to enable their commodification (May, 2006; Kleiner, 2010; Bessen & Mayer, 2008). A

or Skype. This was followed by vocal user discontent, which eventually lead to rolling back the announcement.

salient outcome of this in relation to the development of technology is that patents are not only used to extract rent (Vercellone, 2008), but also grant power to private patent holders to channel the development of technology in accordance with maximizing profits and securing market share. Aside from the importance of intellectual property regulations, geographic disparities in how the minerals necessary to produce the devices are mined, as well as labor conditions in the global capitalist network, require our attention. All in all, these developments point toward the complex character of the global conditions within which the design and development of smartphones takes place.

# Chapter 4 Negotiability of Technology and Its Limitations

## Practices of app development[24]

In this chapter I present an empirical exploration of how programmability, namely writing apps is possible for different mobile platforms. Since 2008, when popular OS (operating system) providers allowed independent developers to write application software (apps), the number of apps has spurred and led many commentators in popular discourse to quickly embrace a new app revolution. Despite the hype, there is a dearth of empirical studies exploring the politics and practices of how software is written for smartphones from the perspective of developers. From the standpoint of the social shaping of technology, which emphasizes that the development of any technological artifact is negotiated amid relevant groups, I explore how expert users, in particular, independent developers are negotiating the development of smartphones by creating apps. Based on semi-structured interviews with 20 developers from 12 different countries, I identify how this newly-emerged group gives meaning to the programmability of smartphones and what structural and technical constraints they encounter in the process of writing apps for different platforms.

[24] Earlier version of this chapter was published in Mosemghvdlishvili, L., & Jansz, J. (2013). *Negotiability of technology and its limitations: The politics of app development*. Information, Communication & Society, 16(10), 1596–1618. doi: 10.1080/1369118X.2012.735252

Chapter 4

# Introduction

”We need to greatly remodel our understanding of what the moving media platforms that apps – or, really, software-based cultures for mobiles – represent, and might become if we can make them a transformational force.”
(Goggin 2011, p. 157)

”It’s a world of imagination; then, through coding, we convert it into apps.” (14-year-old participant from India)

Smartphones have been described as the Swiss army knife of the digital age because they unite the functionalities of a cellular phone, a personal computer, an audio player, a digital camera, a GPS receiver, and a PDA in one converged device. Browsing the internet and playing simple games was possible on earlier (featured) mobile phones, but still, what made the smartphone a distinct technological artifact was its programmability (Raento et al. 2009). Programmability transformed cell phones into pocket-size computers, with smartphones running a complete operating system (OS) that enables the installation of application software, which is now commonly referred to as an app. Writing software for mobile phones was possible earlier, but was a very restricted and often “frustrating experience” for developers (Goggin 2009, p. 237). As Zuckerman (2010) explains, this was due to the tight control of transnational mobile network carriers, which meant that software developers needed to sign agreements with carriers to offer their applications to users.

However, since 2008, when the powerful and popular OS providers for smartphones (e.g. Apple’s iOS, Google’s Android) allowed independent developers to create and distribute application software, the number of apps has increased rapidly. Indeed, in 2011, Distimo, an app store analytics company, counted up to 386,000 apps for the iPhone and 295,000 for Android-run smartphones, most of which were produced by individual developers. The magnitude of these figures led commentators in mainstream discourse to quickly applaud the mobile app revolution. In the world of academia, some started to ponder whether a new “arena of innovation” was emerging

(Sawhney 2009). Such an arena is conceptualized by Sawhney and Lee (2005) as a creative environment, where new configurational potentialities of communication technology are identified by users outside established networks, and spawn new uses of the technology, as well as enhance it with more user-oriented applications.

Despite the rapid growth of apps and the exaggerated claims in popular discourse, Goggin (2011) stressed that there is a serious lack of empirical research addressing the nature and politics of app development, the terms under which apps were developed, and by whom and how. This study aims to contribute to providing insight into the field by conducting a detailed empirical exploration of the current state of app development for smartphones.

The leading question in this study *concerns how developers are appropriating smartphones by creating apps and thereby negotiating the development of this technological artifact.* The answer to this question will be presented in the following order. First, I elaborate on the Social Shaping of Technology as the leading theoretical perspective, with particular focus on its central concept of negotiability. Then, I conceptualize developers in the light of user-participation and present the research questions. Subsequently, the research design and methodology are explained, and this is followed by a presentation of the results. Finally, in conclusion, the major findings are discussed and linked to recommendations for future research.

Chapter 4

## Theoretical framework

By exploring the activities of developers within the complex system of the smartphone industry, two ideas - the *negotiability of technology* and *user participation* are connected, where I argue that opening up the programmability of smartphones for users can be regarded as widening the range of actors who are involved in shaping this technology, which in turn, is stressed by Feenberg (2010) as one of the potentialities in democratizing the technological decision-making process.

Negotiability is a central concept within the research tradition of the social shaping of technology  (SST). It expresses the idea that technology is a social product, patterned by the conditions of its creation and use (Williams, 1997). Negotiability points to the existence of a surplus of workable solutions or choices for any given technology, both in the process of design, as well as in its utilization. These choices are negotiated amid the range of players (or relevant social groups) who may have a different relationship with the technology, as well as a different understanding of it. In other words, there is a range of choices that make various technological routes possible, potentially leading to different social and technological outcomes, whether for society at large or a particular social group. The success or failure of different solutions highlights the scope and extent of relevant groups and forces that shape technologies. The characterization and conceptualization of available choices, and their relation to large-scale social and economic structures, is debated among proponents of the SST and varies among intellectual schools subsumed under this broad approach (Williams & Edge 1996).

Yet, negotiability is not infinite (Orlikowski, 1992), and can be limited at least by two conditions.[25] First, power inequality imposes a limitation: the different groups involved in the development of technology may not be equal in terms of their power or access to resources (Klein & Kleinman 2002). Second, a

[25] Historical background, review of the SST approachn and the limitations of the negotiability thesis are addressed in greater detail in Chapter 2.

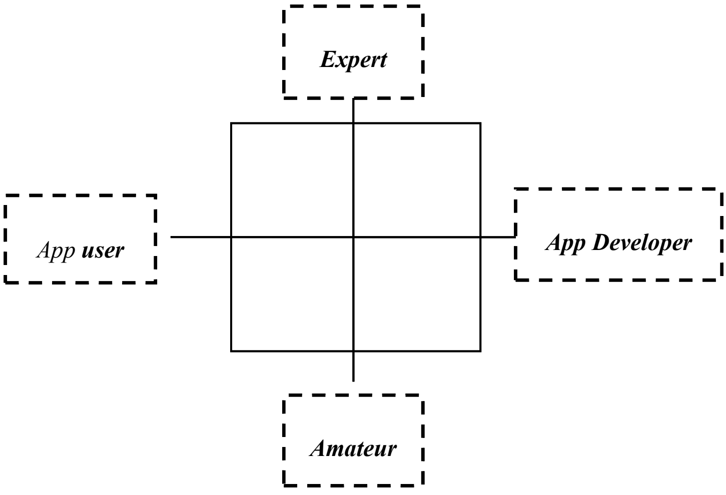
range of solutions may be narrowed down due to irreversibly foreclosed choices. For example, earlier technological choices may influence subsequent developments, while certain solutions may become entrenched in the technical or social infrastructure of existing technologies, thus producing locked-in or black-boxed solutions (Williams 1997). In this way, some technical artifacts may become standardized and stabilized and made available to a user "as commodities, with well-established attributes" (Williams 1997, p. 18). As a result, users may have little opportunity to influence the design and development processes, although they can still develop a different understanding and uses of an artifact. Smartphones are configurational technologies that cater to many different user requirements and needs. Nevertheless, it is software that forms an interface between a machine and the range of social activities to which a device can be applied (Williams 1997). Due to the programmability of smartphones, software has become a critical layer of this technological artifact and makes studying app development an important area for research.

Pursuing the argument further, software has a fundamental characteristic that Huges and Lang (2006) call transmutability, meaning that it can be easily altered and modified. Moreover, software is a means of production that resembles a language in its structure (Glazier, 2006). As a consequence, it is not only open to modification but also serves as a means of creating something new. Admittedly, software development by users has been regarded as the earliest mainstream form of user participation, with an example being the Linux kernel, where users provided software with functionalities beyond what was offered by proprietary packages (Bruns 2006). Certainly, free and open software development by users has long been seen as a way to increase public participation in technical design (Feenberg 2002, 2010). In the present case of smartphone apps, I connect the negotiability of technology with user-participation and argue that due to the cardinal qualities of software, app development by users can be seen as widening the range of players who are involved in the development process of smartphones.

### *Conceptualization of user-developer*

The emergence of what Jenkins (2006) calls a "participatory culture" has challenged the traditional distinction between producers and users as amateurs and professionals. To be able to tackle the diversity of roles, I developed a model (adapted from Budhathoki, Bruce, & Nedovic-Budic, 2008) (see Figure 2). The horizontal axis presents the continuum from an app user (anyone who uses smartphone/apps) to a professional app developer (a person who is developing apps as a part of her work). Whereas developing an app does not necessarily require an individual to be an employed professional, the task certainly needs basic programming skills. The vertical axis, meanwhile, highlights the level of expertise, which ranges from being an amateur to being an expert. Respectively, amateur users without programming skills will fall into the lower quadrant, expert users are in the upper left quadrant, and professional developers can be placed in the upper right quadrant. The lower right part of the figure is empty, since being a professional developer without any programming skills is, by definition, inconceivable.

**Figure 2** Conceptualization of an app user-developer



App development can also be regarded as an instance of what Schäfer (2011) calls the “extrinsic participation” of users who, by creating and releasing software, offer a novel use of a device that is not necessarily prescribed by its manufacturer. Other instances of extrinsic participation have been researched in various contexts, for example, a case of game modders, who publish their (re)creation of game content online for other players to enjoy (Jansz & Theodorsen, 2009).

To summarize the theoretical framework employed in this study explores the negotiability of technology, but with a critical understanding of the possible power asymmetries within the smartphone industry and the possibility of foreclosed choices. I argue that due to the inherent qualities of software, it is particularly ripe for exploration in the light of the social shaping of technology, and we also emphasize that due to the programmability of smartphones, a new group has emerged in the hitherto centralized and concentrated mobile industry (Goggin, 2010): app developers. Thus, in this study, I inquire into how this group is interpreting the affordances of this technological device and negotiating its development.

### *Research questions*

The question guiding this research concerns how developers create apps for smartphones and thereby negotiate the development process of this technological artifact. The main question is divided into three sub-questions: First, I inquire: (RQ.1*) which relevant groups are involved in the development of smartphones*? Then, bearing in mind the possible power asymmetries between relevant groups, the second question asks: (RQ.2) *how do developers position themselves in relation to other groups within the industry*? Finally, taking into account the concept of irreversibly foreclosed choices, the third question explores the possible limitations of available choices by asking: (RQ.3*) what technical and structural limitations do these individuals encounter in the process of app development*?

Chapter 4

# Methodology

Since app development is an emerging practice, there is little research documenting how apps are written for smartphone devices. Consequently, and to impart first-hand experiences and knowledge about the issue, I decided to conduct semi-structured interviews with developers. This enabled to explore the practices and politics of app creation from the perspective of these individuals, who were conceptualized as expert users. To recruit respondents, I placed an invitation in (a) the discussion page of the LinkedIn group, Innovative APP Users & Developers for iPhone, iPad, Blackberry, Android & Smartphone Mobile Devices (which had 3,619 members by November 2011); and (b) the Google Group's discussion forum for Android Developers (56,554 members by November 2011).

The interviews were conducted using Skype and lasted between 50 and 90 minutes each. The format of the interview was mixed, combining voice communication and text messaging. Initially, a video call was held with a respondent to explain the purpose of the study and obtain informed consent. Following this, the interview was resumed in the synchronous text messaging format. Initiating the conversation by video allowed us to establish a certain level of rapport. It also provided a practical solution to the problem that some people may assume a fake identity in an online interview. The interviews were guided by a topic list that was constructed in advance to cover the issues relevant to the research questions. After conducting eight interviews, the list was adjusted to include two additional topics that emerged from the answers of the participants (distribution of apps through online stores, and web-based apps as an alternative platform for app development). The interview transcriptions were analyzed systematically using qualitative interpretation and the ATLAS.ti software.

### *Participants*

Twenty developers from 12 different countries were interviewed (five India, three UK, three Vietnam, and one each from South Africa, Israel, Pakistan, Sri Lanka, Germany, the Netherlands, the Philippines, the USA, and Sweden).

The average age of the respondents was 29. The youngest was a 14 year[26] old developer from India, who, despite his young age, had released 11 apps (for free) on the Google Marketplace. The oldest participant was a 44-year-old developer, who was creating his first app for a smartphone (see Table 4).

Overall, the participants were highly educated, with the majority (12) having a Bachelor's degree. Three developers said they had obtained a Master's degree and two had a PhD. The remaining three participants did not receive schooling beyond secondary education. The group I interviewed reflected the common observation of women being under-represented in the ICT sector. The absolute majority of 19 developers were male. Eight participants identified themselves as independent (or freelance) developers, the second largest group (n=5) was comprised of professional (employed) developers who worked on personal projects in their leisure time, four participants had part-time jobs and devoted an equal amount of time to work-related and personal projects, while the remaining three participants were head of their own start-up company concerned with developing apps.

[26] Due to ethical considerations, the interview was terminated and resumed after receiving parental consent.

**Table 3** Demographics of interview participants

| | **Country** | **Age** | **Education** | **Employment status** |
|---|---|---|---|---|
| 1 | India | 23 | High School Diploma | Part-time employed |
| 2 | South Africa | 23 | B.A. Degree | Founder of a start-up |
| 3 | India | 14 | Secondary School | Freelancer |
| 4 | Sweden | 32 | B.A. Degree | Founder of a start-up |
| 5 | Sri Lanka | 25 | B.A. Degree | Employed as a developer |
| 6 | India | 24 | B.A. Degree | Part-time game developer |
| 7 | India | 31 | Master's Degree | Freelancer |
| 8 | Vietnam | 31 | B.A. Degree | Employed as a developer |
| 9 | Vietnam | 30 | B.A. Degree | Part-time employed |
| 10 | Philippines | 44 | B.A. Degree | Freelancer |
| 11 | USA | 21 | PhD candidate | Freelancer (for educational projects) |
| 12 | UK | 40 | PhD | Freelancer |
| 13 | UK | 37 | Secondary Education | Part-time employed |
| 14 | UK | 27 | B.A. Degree | Freelancer |
| 15 | India | 31 | B.A. Degree | Employed as a developer |
| 16 | Israel | 31 | Master's Degree | Employed as a developer |
| 17 | Pakistan | 31 | B.A. Degree | Employed as a developer |
| 18 | Netherlands | 36 | B.A. Degree | Founder of a start-up |
| 19 | Vietnam | 28 | B.A. Degree | Freelancer |
| 20 | Germany | 29 | Master's Degree | Freelancer |

## Analysis

### *Power dynamics within the smartphone industry*

In order to explore how developers negotiate the development of smartphones with other groups, the participants were first asked to identify the key actors within the smartphone industry. The majority distinguished four main groups; OS platform providers (1) and device manufacturers (2) were named as the most important players in the industry, followed by mobile network carriers (3) and, finally, app developers (4).

Interestingly, the developers regarded themselves as a group of people who give meaning to a device: *"developers influence how people use their smartphones and what place in their daily life it takes. We provide the content and what the user can do with the phone"* (female, 31, Israel). As one participant figuratively put it: "*without developers a smartphone is an expensive brick*", expressing the idea that apps are what makes a smartphone a multipurpose device and provide various options of how this technological artifact can be appropriated.

Overall, the participants acknowledged that it is hard to "*gauge*" developers' power, and when asked to position developers (as a group) in relation to the other players within the smartphone industry, opposing opinions were heard, ranging from the claim that: *"developers are the most important part of it* [smartphone industry]" (31, Pakistan), to the statement that app developers are the least influential group: "*developers are pretty much powerless - they are driven by the popularity of the OSs and devices*" (25, Sri Lanka). Nevertheless, a core group of nine participants agreed that developers have collective power:

> *Individually they have no power at all, collectively they have all the power, for example, I personally have no power to alter the course of events in the Android space or iOS or Blackberry or anything else [...] but in aggregate, the 100k or more developers in Android space and the 400k or whatever developers in the iPhone space basically can lock out the competitive hardware's options.* (44, Philippines)

The collective power was seen mainly in terms of influencing other players within the smartphone industry:

> *I think that app developers can shape the industry by building apps which require different resources, and then manufacturers will put these in their devices. For example, if many app developers see the need for some component in the system, this may be seen as more common.* (21, US)

Three participants related the "app movement" to the work done by developers and claimed that developing apps for smartphones became massively popular after Apple allowed third-party developers to work for their platform, which was, in turn, a result of jailbreaking[27] the first iPhones. As a 25-year-old developer from Sri Lanka put it when asked to give an example of the collective power of developers:

> *At that time* [when the iPhone was released in 2007]*, as I can remember they* [Apple] *haven't had an idea of* [an] *app store concept, and a 14-year-old kid*[28] *hacked the iPhone and after that, the jailbreaking system came. So jailbroken phones got many new applications, but the main device which was developed by Apple had only standard applications. So after that, they* [Apple] *tried to stop the jailbreaking and introduced SDK and came up with the concept of an app store.* (25, Sri Lanka)

It is important to note that while recognizing the existence of collective power, the participants largely agreed that, individually, they are very dependent on OS providers. As one of the interviewees framed it, a developer's individual power is "abused" by OS providers: "*It's an interesting thing on the whole, though, because developers' individual power is so small they're often abused*

[27] Jailbreaking – is a term that denotes removing restrictions of an operating system to installing modification. A related term is unlocking, that means removing restrictions to allow a smartphone connect to a different mobile network carriers.

[28] The respondent probably refers to George Hotz who was 17 years old when he jailbroke the first iPhone in 2007 (see Chapter 3).

*by the platform providers.* [Question: Can you explain how?] *Well, a variety of ways really, draconian usage agreements, shoddy developer utilities or information flows, capricious rules*" (44, Philippines).

In order to explore in more detail how developers were both limited and enabled by OS providers, it was necessary to differentiate between various platforms.

### *Competing platforms*

Generally, the developers I interviewed work with two or more OS platforms, meaning that they rewrite the same app to make it compatible with the different OSs. The developers acknowledged that having multiple platforms hinders the productive development of apps since it requires expertise in different programming languages. At the time of conducting this research there were five major OS platforms: Apple's iOS, Google's Android, RIM's BlackBerry OS, Microsoft's Windows Phone, and Symbian.

The available platforms varied in their: (a) accessibility (open to individuals from certain countries); (b) the type of software licensing (proprietary or open-source); (c) the type of programming language required for development; and (d) control over distribution (terms and conditions on how a developer can distribute an app through an online marketplace). Table 3 sets out the differences between the most commonly-used OS platforms: iOS, Android, Windows Phone, BlackBerry OS, and Symbian. Next to the information provided by the participants, I completed the data by examining the terms and conditions of the OS providers.

**Table 4** Comparative table of app stores' conditions in 2012

| **OS** | **Company** | **License** | **Programing langage** | **Distribution channel** | **Fees** | **Review process** | **Share kept by the app store** | **Country limitations** |
|---|---|---|---|---|---|---|---|---|
| iOS | Apple | Proprietary | Objective-C | exclusively own app store | Annual fee $99 | Yes | 30% | n/a |
| Android | Google | Open source Apache 2.0 | Java | Google Play (or alternative app store) | One-time $25 | No | 0 | 29 [29] |
| Windows pho. | Microsoft | Proprietary | Silverlight and XNA [30] | Windows Phone Marketplace | Annual fee $99 | Yes | 30% [31] | 37 [32] |

[29] Only developers from the following 29 countries could sell apps through Google marketplace (others could only submit free apps Argentina, Australia, Austria, Belgium, Brazil, Canada, Denmark, Finland, France, Germany, Hong Kong, Ireland, Israel, Italy, Japan, Mexico, Netherlands, New Zealand, Norway, Portugal, Russia, Singapore, Spain, South Korea, Sweden, Switzerland, Taiwan, UK, US.

[30] To test an app, Visual Studio 2010 is required, which only runs on Windows Vista SP2; earlier versions of Windows are not supported.

[31] The Windows Marketplace pays a developer 70% of all application. The developer is responsible for paying all of the taxes on the payments she receives.

[32] List of countries from where developers can submit a paid app through the Windows marketplace are Austria, Belgium, Brazil, Canada, Chile, China, Colombia, Czech Republic, Denmark, Finland, France, Germany, Greece, Hong Kong, Hungary, India, Ireland, Israel, Italy, Japan, Korea, Luxembourg, Mexico, Netherlands, New Zealand,

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Blackberry OS | RIM | Proprietary | Java | BlackBerry app world | Free | Yes | 20% | n/a |
| Symbian | Accenture/Nokia | Open source Eclipse Public Licens) | C++ | Ovi Store[33] | 1$ | Yes | 30% | n/a |

When asked to name their favorite OS for developing, the participants were split between the two most popular; Google's Android was favored by a majority of 11 developers and Apple's iOS by seven interviewees. Two participants preferred RIM's Blackberry OS and Microsoft's Windows Phone.

The primary reason for preferring Android was its open-source and availability. As a participant from India (33, male) explained:

> *Android is java based. Java is a syntax that has not changed in like 15 years. Its SDK is open source.* […] *They have very neat and well thought out documentation. The reference guide helps you get things known quickly. As compared to iPhone, which is very restrictive in allowing application developers to do a lot of things the Android platform is very open about. For example, in Android, I can override an app provided by Google like Contacts. And write my own, using the same SDK. But Apple will just not allow this. Apple allows its own apps to have special features and permissions that it does not allow any other app to have.*

Norway, Poland, Portugal, Russia, Singapore, South Africa, Spain, Sweden, Switzerland, Taiwan, United Kingdom, United States.

[33] Ovi Store and the Windows Phone Marketplace merged by the end of 2012.

Another reason for choosing Android was its cost-effectiveness: "*You do not need to buy an iPhone and have a Mac to start developing for Android, you can use any other hardware*" (Vietnam, 31, male).

It was generally recognized that for novice users, who are just starting to develop their very first apps, Google's Android is the easiest to access. Six of seven participants with no previous experience of developing were creating apps using Android. This may account for the fact that there are far more freelancers for the Android platform than for Apple's iOS. The youngest developer within the sample also favored Android: "*I started with HTML five years ago and started moving forward. I like developing for Android. The android-developers list has very helpful, if occasionally sarcastic, people on it. In short it is my hobby*".

The second most popular system was Apple's iOS proprietary software. Initially, when the iPhone was first released in June 2007, it was closed to developers and had preinstalled apps, all produced within boundaries of the company. After a year, Apple changed its strategy from closed to semi-closed, and released the Software Development Kit, thereby allowing third-party developers to also write software for the iPhone, although it retained control over which programming language had to be used. Apple also introduced an in-company review of apps before publication and retained the exclusive destination to distribute them through its app store.

In contrast to Google, which relies on an open-source code (Java, which many developers were already familiar with), apps for the iPhone have to be created in a programming language known as Objective C and C++. Apple has been taking serious measures to stream freelancers in using exclusive tools that are applicable only for iOS. As one participant recalled in April 2010, there was a heated controversy between Apple and developers when the company made changes to its policy, banning all apps that were written in anything other than the language that it approved. The participants in this study generally agreed that Apple's iOS is more restricted for developers in comparison to Google's Android: "*Apple can be very strict sometimes with what they allow developers to do*" (36, Netherlands).

Among the seven developers who favored Apple's iOS as a primary platform to work with, two did not necessarily see the company's restrictions as being negative for developers:

> *The iOS itself is pretty solid and stable, and it's got a great phone design. […] Sure, there are some things Apple does in their own apps that 'normal' developers can't do, but I've never been in a position where this has hampered me in any way.* (31, UK)

The reason to choose Apple's iOS was primarily driven by the popularity of the iPhone among consumers and the prospect of selling apps. As a developer from India explained:

> *Apple is very famous in the US and UK, and that is where the "paying" market is. Android is famous everywhere other than the US/UK since it's a cost effective phone. So the developers get a raw deal when they develop for Android. Also, since it's easy to start [with] Android, as compared to [the] iPhone, there are many more developers. So Android developers go with the strategy of "free" first and then "paid" upgrade. iPhone go with the strategy of "pay" and refund if you don't like it. [An] iPhone developer can make more money than an Android developer since android apps get buried in that mountain of apps.* (31, India)

A Dutch developer who has his own start-up explained further*:*

> *I think that there is a clear difference between iOS and Android. On Android it is very difficult to make money... our sales is [sic] barely 10% on Android compared to iOS, and that is still considered to be good. Regarding paid apps, a majority of developers choose to develop for iOS for this reason, I believe.*

### *The review process and control over distribution*

The label "app store" (although Apple applied a trademark to the term) is generally used to refer to online distribution platforms, where smartphone owners can download an app either for free or for a particular sum of money. All OS providers have these distribution platforms, which are primary channels through which developers reach users. In all cases, developers have to pay an annual or one-time fee to create an account and submit an app to an online marketplace (see Table 3).

The major OS providers offer their own distribution channel for apps, i.e. their 'app store.' However, the terms and conditions according to which a developer can share for free or sell his or her app vary. The most restrictive in this regard is Apple, which retains an exclusive right over iPhone apps distributed through its store. Unless a user jail-breaks an iPhone, software cannot be installed from other channels. Google's Android is the most open in this regard. As a developer from Israel explained:

> *For iOS it [distribution of apps] is pretty much limited to app store. [N]o option for distribution through mobile companies and not much alternative markets. [T]he alternative market is available only to jail broken phones, so this way a developer that wants to reach the biggest market has only one place to go which is completely controlled by [A]pple, meaning that you not only have to share your profits with them but first have to be approved. [W]ith android there are more possibilities: of course [G]oogle's app market where you can publish your app in a day, you can work with mobile companies or any alternative market. [B]asically there are more options and the developer has more control over revenues and the app itself as he doesn't have to obey to some strict rules.* (female, 31, Israel)

Along with distribution, the major OS providers (except Google) retain the right to review an app for both technical compatibility and content before it is published online (see Table 3). So, developers have to submit their creations for a review process, which varies per company and can take from five days to two weeks. Decisions are generally communicated to developers under a non-disclosure agreement, meaning that they cannot reveal why an app was rejected. The participants in this study agree that the guidelines for the review process are vague, especially when it comes to content restrictions.

Along with content restriction, the review process is a hindrance when a developer wants to update an app, since it then has to again be approved by an app store. As one of the participants explained:

> *The main problem here is with the review process. Because it takes time for an application to get reviewed, it's not possible to do quick bug fixes or make fast changes to an application. This can lead to situations where bugs in an application that cause bad reviews are live for long periods of time.* (31, UK)

Google has fewer restrictions in terms of the review process. Unlike Apple's centralized procedure, it relies on 'the wisdom of the crowds', i.e. app users' feedback is trusted as a way to detect inappropriate content or malicious apps. The waiting time to publish an app on Google's marketplace amounts to a couple of hours. However, the main problem identified by developers is the confusion around how Google constructs the list of promoted apps on its website, since there is no explanation available.

Developers can submit their apps to online marketplaces to be distributed for free or to be downloaded for paid. However, there are country limitations when it comes to distributing paid apps. As seen in Table 3, for instance, Google lists only 29 countries whose citizens are allowed to sell apps. India is not on the list and the five Indian developers within the sample highlighted this as a major concern.[34]

### *Developers' reflections on the future of app development*

The respondents were asked to compare app development for smartphones with developing work for PCs and the internet. Those who had previous

[34] By 2015 the list of countries were developers could also submit paid apps to Google Play was – 74. By 2019 up to 50 countries are still excluded.

developing experience agreed that smartphones are more restrictive. As one participant explained, it will have an impact, but mostly for the so-called 'power' users: "*I agree that they are more closed. I do think that it probably hinders innovation to some extent, because there are things you might want to be able to do that you just can't without special systems' support that isn't available"* (21, USA). Google's Android was named in this regard as the least restrictive platform, due to its source code being available. In addition, the participants expressed the view that that app development in the case of smartphones "*is* [in comparison to early PCs] *definitely more commercially oriented*" (female, 31, Israel).

When asked to reflect on the future of the app industry, most participants agreed that apps will become crucial for various businesses as well as for the daily lives of their users. They also expect the developers sector to become more concentrated, meaning that the number of individual developers will decline. As one respondent put it: "*no more single man bands*" will be possible. A 37 year old developer from the UK commented:

> *I think we're still on an uphill at the moment; I've thought for the last few years that this growth in the market wouldn't be sustainable for even this long. I think we'll see that apps become much more of a commodity where even the smallest of businesses have apps to promote themselves. This will likely happen through easier and fast production lines in app development houses.* (37, UK)

When inquired further into the possible reasons of concentration of app market and the decline of number of (individual) freelance developers, there was a shared sentiment among participants that the free market rules influence this process;

> *I think concentration will continue. I think that, left to itself, any market will always tend toward maximum efficiency, and the only way to gain efficiency is by consolidation and reduction of duplicated efforts, that's one and next to this I think value [of information] will be reduced by these corporate owners by insisting on monetizing what is essentially free.* (44, Philippines)

## Discussion

App development is becoming an important platform for mobiles, which in contrast to earlier featured mobile phones can be considered as increasing the range of actors, who are involved in negotiating the development of this technological artifact. Since the early criticism of app development, when Zittrain (2008) compared an iPhone to a closed 'brick', positive changes have occurred; and currently, individual developers are allowed to contribute to app development. This research has shown that a diverse group (in terms of geographic dispersion and position in the industry) is engaged in creating apps for Smartphones, including seasoned developers who switched from working on PCs to Smartphones, as well as a 14-year-old teenager who creates apps out of interest.

The group largely sees its role as giving meaning to the device; by creating apps for various purposes, developers provide new ways for smartphones to be used and appropriated. This resonates with the view of software being a link between a machine and the range of social activities to which a technical artifact can be applied.

The negotiation of the development of smartphones occurs in a complex system among relevant groups. My first research question aimed to identify the key groups involved in this process. In the interviews, the participants named four main groups: OS providers, manufacturers, mobile networks (carriers) and app developers. I asked about (RQ2) the extent to which the groups varied in terms of their power and influence. Generally, OS providers and manufacturers were considered to be the most powerful when it comes to negotiating the future development process of this technological artifact. This differs from the observation made by Goggin (2010), who indicated that carriers were "decisively" the most important group in the global mobile media industry, and were in the position to influence the direction of the development process (of smartphones) through pricing policies, marketing campaigns and strategic alliances with device manufacturers. The developers in this study, however, perceived a shift of power within the industry from mobile carriers to the OS providers. A conceivable explanation is that for creating apps developers are strongly dependent on the platform that is run by the OS providers. Furthermore, I found there was a tendency in the opinion of my participants to advocate the existence of collective power. In principle, developers could influence other groups while negotiating the development of smartphones when they joined forces.

The main limitations (RQ3) that developers encountered in this process of negotiation of the development of Smartphones were related to the terms and conditions under which the OS providers allowed them to create and distribute apps. As seen in the comparative table, there is a degree of difference between the OS platforms with regard to their openness and accessibility or the required programming language. Generally speaking, there is a tendency to release a compulsory SDK and allow third-party developers to create apps, while the OS platform itself remains closed under a proprietary software license (for example, Apple's iOS or Microsoft's mobile phone). In other words, developers can write software for smartphones, but under strictly defined terms and conditions. Furthermore, the main proprietary platforms retain the right to review and approve an app, which undeniably strengthens the role of the OS providers. By controlling for both technical compatibility and content, this group takes on the important role of gatekeeper. In a commentary on the politics of app development, Goggin reaches a similar conclusion, arguing that although apps have become an important platform "it is still firmly in the gift of Apple" (2011, p. 154).

The strategic decision by Google to have an open-source platform for app development is certainly positive when it comes to making software development for smartphones accessible for users. However, it can equally be seen as an example of what Sawhney (2009, p. 113) (2009, p. 113) calls a corporate effort to "harness open-source energy for their own benefit", implications of which should be further explored. Lerner and Tirole (2005) explain in what situations it can be advantageous for a commercial company to release propitiatory software under an open-source license; when a company expects to boost its profit on a complementary segment or is lagging behind the leader in that segment of the market. In fact, the first Android enabled Smartphone was released only a year after the iPhone was introduced. Google, by releasing source code for Android, on one hand quickly attracted a large number of developers and on the other hand, made possible to boost profits on its complementary services (e.g. Google Play, formerly known as Google Market).

As such non-proprietary (or copyleft, meaning that software is not protected by copyright and can be freely used and modified) software has been discussed as a way to realize the democratic potential of communication technologies (Berry & Moss, 2006), leading to user innovation (Von Hippel, 2006), as well as a better, that is, a more efficient methodology for software development (Raymond, 2001). In other words, it has been presented as a political (a basic human right and moral) or an engineering issue. Despite these differences in emphasis, which themselves have implications for the discourse, Smith and Smythe (2009) highlight the centrality of three concepts in both approaches: (1) freedom, (2) open, and (3) collaborative. They argue that open source and free software development has a different production process and organizational form, which is collaborative and based on a different sense of reward, one that is intrinsic, not extrinsic.

In the case of Google's Android, the interviews revealed one big difference between contemporary app development for smartphones and the previous open-source/free software initiatives in the software industry. Whereas open source/free software (e.g. Linux) is almost always a product of people working together, collaboration within a community is largely absent in the app industry. The most common picture painted is that of independent developers mainly working alone (as freelancers) or in small groups (in start-ups). One possible explanation for the absence of collaborative work can be the type of license that protects Android. All parts of Android except Linux Kernel (which is protected under the free/libre software General Public License) operate under the Apache License. The Apache license is open source but not copyleft free/libre software license. The major difference is that users who develop products based on the Apache license are not required to distribute source code (B. Smith, 2011; Stallman, 2011). In other words, software written under the Apache license can become copyrighted as well as be made proprietary. This is essentially forbidden in case of copyleft licenses, which aim to guarantee that created software does not get enclosed in a proprietary model.

Above all, for the negotiability of smartphone development, the emergence of app stores is a very important factor. The major OS providers are tied to their respective distribution channels and, by defining the terms and conditions of how an app can be distributed, they require: a developer to register and pay registration costs. App stores also retain the right to review (except Google[35]) an app; and set country-specific limitations on distribution.

These practices can arguably be seen as narrowing down the possible choices/solutions for app developers. Then, app development remains on the threshold of creating a piece of a single commodity, which gets integrated into the platform (only) through an online app store. Respectively, apps created by individuals and small groups are only brought together in app stores, which represent a marketplace rather than a generative platform for user-innovation.

## Conclusion

This study contains a timely empirical exploration of the opinions and practices of the developers of smartphone apps. This group has recently emerged and has, thus far, received scarce attention from researchers. The study was theoretically embedded in the social shaping of technology perspective on ICT development. I emphasized the negotiability of technology, with a critical understanding of its limitations, and explored how app developers were negotiating the development of smartphones through app creation. In pursuing this, I inquired which main groups were involved in the development process and how app developers positioned themselves in relation to other players. As well as this, I explored the limitations of

[35] There is no official content related review process, and developers have to themselves ensure that their app is compatible with Android. However, in May 2013, Google removed advertising blocking apps from the Google Play store, one of which was Adblock Plus.

negotiability by documenting which structural and technical obstacles developers encountered in the process of app creation.

The negotiability of smartphones is a complex process involving (but not limited to) OS providers, device manufacturers, carriers and app developers. From the perspective of app developers in this process, they are primarily dependent on OS providers, who are in the position to define the terms and conditions of how software can be written. However, in the negotiation process, developers recognize that by writing software they give meaning to the device and also have the collective power to influence other groups (e.g. OS providers). As such, and in comparison with previous cell phones, a wider group of players is now involved in the development of smartphones. Certainly, there is an evident shift of power from carriers to OS providers, who have emerged as a strategically important group in the smartphone industry, which, as well as overseeing app development, takes on the role of gatekeeper (through a review process).

App development has become a new platform for expert users and enables them to engage in the negotiation of technological design. Nevertheless, the structural limitations they encounter in this process cannot be overlooked. In this context, the emergence of app stores as a primary medium between developers and end-users, as well as the use of open source but not libre/free software licenses is crucial. From the perspective of the Social Shaping of Technology, this can be seen as an example of irreversibly foreclosing choices.

This exploratory study was limited by its scope. Using LinkedIn and Google Groups, I succeeded in interviewing a diverse group of app developers. The number of participants was small, but their geographical diversity and age range were assets. Without claiming to have presented a representative overview of this emerging sector, still, I believe the study reached diverse voices. Future research with a larger sample and addressing developers on platforms beyond LinkedIn and Google will advance our understanding of how technological development is negotiated by app developers and with what limitations. In order to fully understand these limitations, it is important to scrutinize the terms and conditions under which developers are enabled to write software, as well as the license types under which their works can be distributed (shared). In addition to this, the recent spur of the so-called patent wars that have flooded the smartphone industry invites for close inspection and evaluation. For future research, it will also be vital to expanding empirical exploration from expert to amateur users, who comprise the overall majority of the population and inquire to what extent this group is involved in the negotiation process of how smartphones develop and are used. From the theoretical perspective, it is critical to conceptualize the link between user participation and the notion of the negotiability of technology and explore its implications for public involvement in technological design.

# Chapter 5 Free Your 'Most Open' Android

## A comparative discourse analysis on Android[36]

Through this chapter, I convey a comparative discourse analysis of how Google Inc. and the Free Software Foundation Europe (FSFE) discursively construct and contest Android. Methodologically, I use Political Discourse Theory to engage in the textual analysis, identify and compare key signifiers and nodal points across the exemplary texts from the two actors, and interpret their meaning vis à vis contextual insights about the political economy of Android's production. Albeit being marketed as 'the first truly open platform' for mobiles, through the textual analysis, I find Google's definition of open source practices strictly *conditional*. I argue that Google's usage of compatibility rhetorically as well as techno-legally justifies the conglomerate's control over the platform. By contrast, the discursive moment by free software activists, through a campaign "Free Your Android!" deconstructs the discourse on open source and attempts to politicize the access to code in the mobile domain. This is done by extending from the well-known developer's four freedoms onto users' privacy due to the personal character of mobile devices. Such articulation of free software in relation to the privacy of user data is a new development and arguably has the potential to contribute to widening support to the movement outside a dedicated group of followers.

[36] The chapter was published in Mosemghvdlishvili, L., & Jansz, J. (2018). Free your 'most open' Android: A comparative discourse analysis on Android. *Critical Discourse Studies*, 17(1), 56–71. doi: 10.1080/17405904.2018.1554536

Chapter 5

# Introduction

Through this chapter, I aim to contribute to the critique of open source, explore and explain what is understood of current open source practices in the context of smartphones, using the example of the dominant operating system (OS) Android. In what follows, I build a case to comparatively explore how texts produced by Google Inc. and the Free Software Foundation Europe (FSFE) discursively construct and position Android. By comparatively studying discourses that appear in sharp contrast, I aim to problematize the practices of open source development for mobiles and propose modest recommendations to the digital rights activists on how to emancipate discourse on digital rights to encompass the broader issues at stake.

Whereas there is burgeoning literature exploring meaning, practices, and implications of free and open-source software development vis à vis proprietary (Chopra & Dexter, 2008; Kelty, 2013; Sullivan, 2011; Wolf, Miller, & Grodzinsky, 2009), my focus lies on code production and distribution for mobile devices (smartphones and tablets). I suggest such exploration is critical because smartphones, just like computers, are programmable devices. This means the affordances of this technology (how it can be used and with what ramifications) is prescribed through software.

While I recognize the "politics of artifacts" (Winner, 1985), meaning that values of the broader socio-economic system are inscribed in (any) technological design, I put particular emphasis on the politics of code/software, because of its prescriptive nature. To elucidate; code is *performative (*meaning it runs on hardware and executes certain actions, simulates what a user sees and engages with) but, more importantly, code is *prescriptive* because through software the range of possible uses of hardware is defined (Fuller, 2006). Without diminishing the way end users may appropriate a device, I stress that the range of affordances is encoded through software that depending on the techno-legal status (proprietary, open-source, or free/libre) carry different power arrangements with respect to access and ownership. Without further speculation I uphold, that code/software has a critical functioning in much of current infrastructure and areas of social life.

Not merely due to our reliance on code-mediated technologies, but inherent qualities of code (as means of delegating agency and prescribing affordances of technologies), I argue that software production needs to be closely studied and examined with respect to concepts such as power, ownership, and access.

To sketch a broad picture, nowadays, similar to the PC domain, we can talk about several locked-in mobile platforms, which not surprisingly are structured around Operating Systems (OS), often written in different programming languages, owned by competing corporations on the market (Mosemghvdlishvili & Jansz, 2013) and made not interoperable (meaning, an app - written for one platform cannot run on another, without being rewritten in the respective language).

Concerning distribution, Google's Android retains the largest market position. From 2012-2017, between 59% and 85,9% of the globally sold mobile devices came with Android preinstalled as the main OS (Statista, 2018). For an advertisement and search company, that was a new actor in the hitherto concentrated telecom market, entering the mobile domain without any previous expertise or assets, and gaining such dominance was impressive but not surprising. Through the political-economic analysis of Android Spreeuwenberg and Poell (2012) proposed that such dominance was achieved by Google strategically adopting only "certain open source practices" and negating others. The conglomerate (later renamed into Alphabet Inc.) bought the OS from a start-up for an undisclosed sum already in 2005, and in two years released to the market along with the members of the Open Handset Alliance (OHA)[37] as "the first truly open-source and free platform for mobiles" (Google Inc., 2007).

Despite being marketed as 'open,' the company's open-source practices were contested by Free/Libre and Open Source Software (FLOSS) activists and tech

[37] The OHA was formed by Google's initiative and comprised 34 members; among them are: the handset manufacturers, HTC, Motorola, Samsung Electronics, and LG; large mobile carriers such as T-Mobile; and content providers. Currently, the consortium includes 84 companies and provides a segment of the mobile industry that is grouped around the free (as in no costs) Android operating system.

commentators (Amadeo, 2013; Carmody, 2011). A year after Android's introduction, the Free Software Foundation's European branch started a campaign "Free Your Android!", producing texts as well as instructions on how to liberate Android devices by installing free software on Android-powered devices.

To explore subtleties of open source practices for mobiles, I delignated the two organizations and comparatively studied texts about Android published by Google Inc. and the Free Software Foundation Europe (FSFE). These two organizations stand in sharp contrast in how they articulate the meaning of Android in particular, and software/code more broadly. This was chosen deliberately. By juxtaposing appearing in opposition discourses, I aim to produce an insightful critique on how open source and open platform are understood in the mobile domain.

With this in mind, I explore through discourse analysis how Android is signified by the two organizations using analytical tools drawn from political discourse theory (Glynos, Howarth, Norval, & Speed, 2009; Howarth, 2010; Howarth & Stavrakakis, 2000; Howarth & Torfing, 2004; Torfing, 1999). Before delving into the methodology, I will first briefly review the differences between open-source, free/libre, and proprietary software, and how these are formed by assigning different software licenses.

### *A primer into FLOSS licenses*

In a nutshell, free and open-source software (FLOSS) is software that is distributed with source code (human-readable instructions on how the program is written and what it does) and potentially means that code literate individuals can see what the program effectively *is,* and not only how it performs or appears. The difference between free software and open source is often vague. Technically, both require the source code to be accessible for modification. However, the terms and conditions relating to how these modifications can be further distributed draw a line between *copyright* and *copyleft* licenses and lies at the core of understanding the subtle differences between what came to be named as free/libre software on the one hand and open-source software on the other, while both are opposed to proprietary software.

To trace back the current situation, where there are various forms of software development and distribution we need to make a short detour into the construction of intellectual property concerning code. To explain; initially, software was not copyrightable and supplied together with hardware. However, in 1980, Congress of the United States included "computer program" in the list of copyrighted goods (under Title 17 of the United States Code, which outlines copyright law) and effectively enabled companies to start selling software (Lemley, 1995). As a result, over the past 25 years, most software was pushed to the market under proprietary licenses (e.g. Microsoft's Windows, or Mac OS X) and the software industry became one of the largest in terms of accumulated capital.

The FLOSS movement was a reaction to the commodification of software and enclosure of source code through intellectual property; however, there came to be a difference between free/libre and open-source software. Following the change in copyright law, in 1985, Richard Stallman founded the Free Software Foundation, an organization that became a flagship and vocal actor for the movement. The term Free Software was defined as a set of principles that guaranteed to: (1) use a program for any purpose, without restrictions, such as date, purpose, or geographic area, (2) study workings of a program and adapt to own needs, without placing any legal or technical restrictions to access and modify the source code, (3) improve the program, and (4) release it back to the public, so that the whole community benefits, known as the reciprocity principle (see also: Wolf et al., 2009; Sullivan 2011; Chopra & Dexter, 2010).

What came to be referred to as f*our freedoms* was legally protected by activists into a copyleft license, namely the General Public License (GPL) and its later versions. The term *open source,* as such, was coined later in 1998 by Eric Raymond, shortly after Netscape Communications Corporation announced that it was releasing the source code of its browser (Mozilla) freely on the internet, which many perceived as the untapped business potential of free software. In the original essay "Goodbye `free software'; hello, `open source`" Raymond (1998) argued that there were two problems with *free software;* first, it was a "confusing" and "very ambiguous" term, and second, it was making "a lot of corporate types nervous." Therefore, to "make serious gains in the mainstream business world," a "new and better label" was necessary (ibid.).

However, open-source was not merely a new label, its definition outlined in 10 criteria by the Open Source Initiative (OSI, an organization founded by Richard Raymond and Bruce Perens), and most importantly, it shifted from the above mentioned *four freedoms/rights of users* to the *rights of the producer* to freely distribute (sell or give freely away) such software. Respectively, in the past years, myriads of open source, and permissive licenses were also developed, which require the source code to be accessible but do not restrict how this can be distributed.

To recapture, without introducing copyleft, it is not possible to differentiate within FLOSS software. Copyleft is an antipode of copyright: as it "uses copyright law, but flips it over to serve the opposite of its usual purpose" (Kleiner, 2010). In other words, copyleft claims ownership legally but practically renounces it by giving everyone the right to use, modify, and distribute code, but with a responsibility to share-alike. This is called the *reciprocity principle,* a term coined by the FSF to denote such obligation that if one modifies free software, the derivative works must be released under the same terms (so that others will also benefit). Hence, by guaranteeing reciprocity, the free software activists try to preserve the common pool of resources and prevent the fruits of the labor of a community from becoming enclosed by permissive or proprietary licenses.

In such a line of reasoning, we can differentiate between proprietary and free/libre and open-source (FLOSS) software based on whether access to source code is given. And within the FLOSS software, further distinguish whether the reciprocity principle is treasured in a license or not, between *free software* (e.g., GNU General Public License - GPL) from *non-free/permissive* licenses (e.g., Apache 2).

Why such scrutiny of licenses is needed is determined by their ramifications because alongside Kelty (2013), I maintain that software produced under such different licenses, albeit both being open source, benefits different actors. In other words, the permissive licenses do not require adaptations to be redistributed back under the same terms; this enables making the adaptation of software proprietary again, hence what was developed by the community for free or released in the public domain can be enclosed by a private party and commodified. With respect to licenses, Android is an interesting and peculiar case; the OS was built around the Linux kernel, which is protected by a copyleft license (GNU GPL version 2). However, most of the remaining code that makes up the OS was released by Google under a permissive license (Apache 2).

With this in mind, I focus on Android, which is open source and commonly perceived to be a free OS for mobiles. Using the analytical tools of political discourse theory, I examine comparatively texts produced by two organizations, the conglomerate Google and a non-profit advocacy group, the FSFE.

Chapter 5

# Discourse-theoretical framework

## *Political discourse theory*

Analytically, in this study, discourse is understood as a temporarily established totality of meaning, where each sign is in a certain relationship to other signs. It represents a particular view of reality, often masking the ethico-political subjectivity of an articulatory practice. There are various ways to conduct a discourse analysis (for a review see: Glynos et al., 2009), but I draw on the analytical tools of Political Discourse Theory (Howarth, 2000).

Political Discourse Theory was initially developed by Ernesto Laclau and Chantal Mouffe in 1985 when the authors conducted a genealogical analysis of Gramsci's concept of hegemony (2001). Their way of analysis came to be referred to as the discourse theory.[38] Over past years, it has been applied across different disciplines: *policy studies* (Howarth & Torfing, 2004; Rear & Jones, 2013), *media studies* (Carpentier & De Cleen, 2007; Carpentier & Spinoy, 2008), and *technology studies* (Berry, 2004; Dahlberg, 2014; Mangalousi, 2013).

This theory is primarily a social theory that stresses the radical contingency of the social and maintains that all social phenomena are discursively constituted. This does not imply that the existence of material reality is denied; rather, the existence of meaning formation about it outside discursive practices is rejected. Consequently, within the rationale of the theory, all natural, social, and physical objects or phenomena are treated analytically as discursive, constructed within a discourse, and subject to discourse analysis.[39] This

[38] Alternatively terms such as Poststructuralist discourse theory and the Essex School of Discourse have been used as well.

[39] Laclau and Mouffe (2001, [1985]) clarify that by arguing “that every object is always constituted as an object of discourse”, the existence of the objects external to thought is not denied, but rather the very assertion that ”that they could constitute themselves as objects outside any discursive condition of emergence” (p. 108). For example, an earthquake is an event that certainly exists in the sense that is occurs at a

theoretical abstraction is a necessary precondition for a discourse analyst to engage in the study of texts.

The reason why I chose the PDT is that it provides analytical tools to dissect the formation of *any* discourse by looking at how each signifier is related to another and what holds these relationships together (nodal points). Unlike another dominant approach, the Critical Discourse Analysis (CDA), in this study, I do not trace linguistic elements and modes but dissect texts into signifiers and examine how they relate to each other.

### *Analytical tools*

The primary concept that enables differences between discourses to be analyzed is through identifying *nodal points.* Nodal points according to Laclau and Mouffe (2001), are key signifiers, sort of privileged signs that redefine the meaning of other signifiers by becoming center-points of a discourse. The fixation of nodal points in a certain relationship to other signifiers organizes discourse and is known as articulation. Discourse theory maintains that any articulation is inherently a political act, as it establishes a particular relationship between signifiers. Laclau and Mouffe further develop this, maintaining that establishing a relationship between different signifiers is possible through the logic of equivalence or difference. The former articulates a certain sameness between different signifiers, whereas the latter dismantles existing differences among signifiers and mitigates them. These practices are at the core of signifying the identity of any object or subject. Exploring the key signifiers (and nodal points) and the logic of signification (equivalence or difference) through which they are related to each other enables a researcher to study the formation of different discourses. In this study, I adopt the same analytical strategy to understand *how Android is articulated in relation to other signifiers and whether this process is different between texts produced by Google and the FSFE.*

certain time and place, but whether this event is constructed in terms of a 'natural phenomenon' or 'expression of the will of God', is discursively constituted.

Political discourse theory, in my view, is unique because it allows examination of the signification process on various levels, from phenomena to 'identities.' Yet, the word *identities* is avoided in this tradition. This is because PDT departs from the orthodox Marxist understanding of identities as determined by a pre-existing structure (e.g., class as determined by an individual's relation to the mode of production: capitalist vs. working class); instead, it sees identity formation as a contingent process where individuals (and groups) identify with discursively formed subject positions. Hence in PDT, we speak of subject positions instead of identities. *A subject position* captures an individual's (or group's) position within a discursive structure. Discourses also include *social imaginaries*, which are defined as visions on how a particular aspect of social life or society as a whole should be structured (Dobbernack, 2010). Analytically, I will use these concepts to explore what subject positions are formed in the examined texts and whether views (social imaginaries) about mobile code differ between Google and free software activists.

### *Exemplary texts and analysis*

The discourse analysis at hand is a contextual interpretation of the textual elements (chains of significations, key signifiers) that were identified in exemplary texts. The material comes from a textual corpus that was collected between 2012 and 2015. Namely, all press releases from Google's official blog that mentioned mobile or Android, and all pages of the Android's official website were retrieved and stored. Likewise, all main pages of the campaign "Free Your Android!" published by the FSFE, as well as email newsletters of the campaign used by volunteers to discuss practical matters, were retrieved. Due to the absence of explicit informed consent, texts retrieved from the internal email newsletters were not used as exemplary; however, contextual information acquired by reading them closely was actualized in the analysis process.

The textual material selected to capture the discourse associated with Google Inc. included the first press release published together with the OHA members. It was one of the most lengthy press releases and widely shared in (online) media. The material further included webpages of the Android Open

Source Project and a blog post on the Android's official blog by Andy Rubin, former SVP of Mobile and Digital Content, at Google Inc. The exemplary texts from the Free Software Foundation comprised the webpage of the campaign "Free Your Android!" and an article - written for The Guardian by the president of the organization Richard Stallman "Is Android really free software?". The article was discussed by FSFE volunteers and used as input for structuring the text for the aforementioned campaign. In Table 5, the exemplary texts are described, and their source, word count, and retrieval date are noted. These discursive moments were chosen because they addressed one of the three themes that were identified for this inquiry: signification of Android, possible subject positions in relation to Android, and social imaginaries regarding mobile technologies.

**Table 5** Exemplary texts included in the signifier-level analysis

| # | Author | Type of document | Word count | In-text citation | Source |
|---|---|---|---|---|---|
| 1 | Google Inc. | Press release | 3560 | (Google Inc, "Industry Leaders Announce Open Platform for Mobile Devices Group Pledges to Unleash Innovation for Mobile Users Worldwide", 2007) | Industry Leaders Announce Open Platform for Mobile Devices Group Pledges to Unleash Innovation for Mobile Users Worldwide. (n.d.). Retrieved September 9, 2015, from https://googlepress.blogspot.com/2007/11/industry-leaders-announce-open-platform.html |
| 2 | Google Inc. | Webpage | 567 | ("Philosophy and Goals \| Android Open Source Project", 2012)[40] | Philosophy and Goals \| Android Open Source Project. Retrieved September 11, 2015, from http://source.android.com/about/philosophy.html |

[40] As of 14 November 2017, the URL is not accessible. The page has been renamed into "Governance Philosophy" and assigned a new address https://source.android.com/setup/#governance-philosophy

| | | | | | |
|---|---|---|---|---|---|
| 3 | Google Inc. | Webpage | 554 | ("Peoples and Roles of Android Open Source Project", 2012)[41] | Peoples and Roles of Android Open Source Project. (n.d.). Retrieved July 4, 2013, from http://source.android.com/source/roles.html |
| 4 | FSFE | Campaign Website | 1055 | (FSFE, "Free Your Android!", 2012) | Free Your Android! - FSFE. (n.d.). Retrieved September 10, 2015, from http://fsfe.org/campaigns/android/android.html |
| 5 | Rubin, A. (SVP Google Inc.) | Blog Post | 1083 | (Robin, 2012) | Rubin, A. (2012, September 14). The Benefits & Importance of Compatibility. Retrieved August 23, 2015, from http://officialandroid.blogspot.nl/2012/09/the-benefits-importance-of-compatibility.html |
| 6 | Stallman, R. | Article | 1522 | (Stallman, 2011) | Stallman, R. (2011, September 19). Is Android really free software? Retrieved November 1, 2015, from https://www.theguardian.com/technology/2011/sep/19/android-free-software-stallman |

During the process of analysis, the texts were explored on the sentence and word level; (a) key signifiers and chains of meaning (equivalence or difference) were identified in each exemplary text, (b) nodal points were identified within each text and compared across texts, (c) identified key signifiers and nodal points were then compared across producers (Google and FSFE). While interpreting the meaning of the signifiers and the ramifications of particular articulations, I drew on the contextual knowledge that was gathered by reading the whole textual corpus as well as relevant academic literature on the political economy of Android (Fuchs, 2011a; Spreeuwenberg

[41] As of 28 March, 2018, the URL is not accessible. The page has been renamed into "Project Roles" and assigned a new address https://source.android.com/setup/start/roles

& Poell, 2012). In the coming section, I first present the textual analysis; the key signifiers and relations between them, followed by discussing what such discursive formations entail.

## Analysis

### *Signifying Android*

### *Google: Android is open but compatible*

The key signifiers, as identified across the exemplary texts that define Android through their chains of equivalence are: *open (-source, -platform, -ecosystem); software product/stack;* and *compatible.* Android is primarily presented as open, whether it is a platform or (in later texts) an ecosystem. It is referred to as "intentionally open", "first truly open", "with [a] new level of openness", and "pragmatically open" (Google Inc, "Industry Leaders Announce Open Platform for Mobile Devices Group Pledges to Unleash Innovation for Mobile Users Worldwide", 2007).

Open itself is empty of meaning, and Google draws on two chains of equivalence to anchor its definition: (a) open is a platform protected by an open-source license; and (b) open means freedom to collaborate (enabling freedom of use and customization). By open source license, Google refers to Apache 2, which is applied to a significant part of the OS but does not fully cover all of its parts. To clarify, the architecture of the OS is comprised of different layers of code: the Android Open Source Project (AOSP) is covered by Apache 2; the Linux kernel is protected by General Public License (GPL); and a whole range of Google's services come as proprietary apps (e.g., Gmail, Google Maps, and Google Play). This aspect is important to note as it completely ignores the existence of proprietary elements in what is marketed as an open-source platform.

The signifier *open* is pivotal in constructing the meaning of Android, also because it is put forward as a solution to a problem that justifies why Android was created in the first place.

> *We [Google] created Android in response to our own experiences launching mobile apps. We wanted to make sure that there would always be an open platform available for carriers, OEMs [Original equipment manufacturers], and developers to use to make their innovative ideas a reality. We wanted to make sure that there was no central point of failure, where one industry player could restrict or control the innovations of any other. The solution we chose was an open and open-source platform.* ("Philosophy and Goals | Android Open Source Project", 2012)

As a result, a strong chain of equivalence is constructed between being an open platform (protected by an open-source license) and enabling collaboration; collaboration itself is set as a condition for innovation. Google as a company is positioned as an enabler, curator, and literally the "shepherd" of the open platform (Robin, 2012). On the other hand, Google emphasizes that an open platform is necessary to prevent one player's control over the market. Notwithstanding this, when juxtaposed with the market reality (that is, the distribution of operating systems), Google has a dominant position in the oligopolistic market of mobile Operating Systems.

With respect to the constructed chains of difference (what Android is not, or is different from) in Google's texts, I observed a difference between talking about open source and free software, which is not only marginalized (excluded from the universality of the open platform), but also antagonized. This stands out in Google's explanation of its choice of license: "Android is intentionally and explicitly an open-source, *as opposed to free software*, effort: a group of organizations with shared needs has pooled resources to collaborate on a single implementation of a shared product" (emphasis added, "Philosophy and Goals | Android Open Source Project", 2012).

In other words, Google presents Android as an open-source platform, including the efforts of the developers' community and industry players. At the same time, it excludes 'free software'; through the logic of difference, free software is relegated to the margins. This can be seen in the above-listed quote were Android is positioned explicitly "as opposed" to free software. In discourse theory's terms, such differentiation creates a new (antagonistic) polarity in which *open source ≠ free software*. This polarity conflicts with the technical understanding of these concepts in which open source and free software are more similar than different, as they are both based on access to the source code, as well as on community-driven development.

The second key signifier defining Android is a *software product* (alike labels were “holistic software product”, “integrated software stack”, “single product”, and “shared software product”). Google puts the most effort into discursive moments to fix the meaning of Android as being both an open platform (guaranteed with an open-source license), free to any modification and usage, while at the same time presenting it as a “holistic product” (Robin, 2012). Treating multi-layered, vast lines of code as one single product discursively dismantles differences between parts of Android that are legally licensed under distinct agreements and positions the whole OS as one commodity.

There emerges a logical inconsistency between having an open platform that is available for modification and one single product. The way it is discursively achieved is by problematizing the freedom to use and customize the OS, as leading to fragmentation (of the OS) and negative user experiences as a probable consequence. The constructed danger then is resolved through the third signifier, *compatibility*, which redefines what it means to be open, and serves as a nodal point.

### *Compatibility*

In Google’s usage, compatibility is not only a discursive construct but also a legal-technical configuration. It consists of the Android “Compatibility Definition Document” (CDD), which lists what it means to be Android compatible, and a downloadable program, the “Compatibility Test Suite” (CTS), where developers can test the compatibility of their apps.

> *Devices that are ‘Android compatible’ may participate in the Android ecosystem, including Google Play; devices that don't meet the compatibility requirements exist outside that ecosystem. In other words, the Android Compatibility Program is how we separate*

> *'Android-compatible devices' from devices that merely run derivatives of the source code* (Robin, 2012).

Compatibility becomes a condition that allows the use of Android's source code; i.e. anyone can download, modify, and use the source code. In fact, this is confined to the AOSP, which excludes Google's apps. An inevitable consequence is that, without agreeing on compatibility and signing the compatibility document, the layer that is normally associated with the basic functionalities of a smartphone is excluded.

> *Anyone can (and will!) use the Android source code for any purpose, and we welcome all such uses. However, in order to take part in the shared ecosystem of applications that we are building around Android, device builders must participate in the Compatibility Program* ("Philosophy and Goals | Android Open Source Project", 2012).

***FSFE: Android "almost free"***

In the texts produced by FSFE, Android is primarily deconstructed through its negative relationship to free software. The signifiers that emerge as key are: *free (vs. non-free software), privacy, and control.*

*Free* is the primary signifier used by the FSFE to deconstruct Android as is illustrated by statements like "it is almost free", "a mostly free operating system mainly developed by Google" (FSFE, "Free Your Android!", 2012). Such articulation stresses the idea that Android is only partially free. What is free, as explained in the campaign, is only a part of the OS, known as the AOSP, which releases the source code of Android after each major update has been completed. Developers can then download the source code, use and modify it under the Apache 2 license. Despite this, it is stressed that it is not possible to run devices on free software because all Android devices come with preinstalled proprietary drivers (the so-called firmware, which is a small segment of software that enables phone manufacturers to start the OS) and Google's proprietary apps.

The FSFE campaign evokes discourse on free software and refers to the established definition of the four freedoms as formulated by Stallman. In FSFE's texts, the four freedoms are enacted, and all non-free software is articulated as endangering democracy. This claim is based on two arguments, notably that non-free software is dangerous for democracy because it violates the four freedoms, and second because it threatens users' privacy. The first argument is consistent with the early discourse on free software (Berry, 2004). What is new, though, is relating free software to privacy.

### *Privacy and control*

The threat to privacy is stressed as being more dangerous and pervasive because smartphones are personal devices, carried around by a user in almost all social settings.

> *Most users do not have full control over the personal data on their device. Convenient solutions for synchronization and data backup trick more and more people into storing all their data on centralized servers run by some profit-driven corporation. These are usually based in the US and are required to hand your data over to the US government on mere request. Whoever has personal information about us is able to manipulate us. Therefore, non-free devices are a threat to democracy and to our society* (FSFE, "Free Your Android!", 2012).

This quote explicates the claim that proprietary devices and software are designed to hand data to private corporations. Even though the exemplary text was produced by the European branch of free software foundation, we see a reference to the US government. Here, the authors probably hint at the NSA mass surveillance programs in the USA that were exposed by Edward Snowden in the same time period as the campaign text was written.

What I found particularly notable in the way the meaning of privacy was constructed in the campaign texts was the absence of any reference or an attempt to establish a chain of equivalence with the appropriation of user data for-profit interests. Needless to remind, in this context, that Google's business model is primarily driven by advertisement fees (see Fuchs, 2012), which is made possible through harnessing data from its (free of charge) services like Gmail, Google Play, Google Analytics. Nonetheless, a critique of such appropriation of user data by private corporations was not present in activists' texts. Unlike Google's use of the word, where *control* is understood as a necessary obligation for Google to prevent fragmentation of the platform into many incompatible parts, the digital rights activists perceive control as a user's right to "truly' own the devices they have purchased.

### *Subject positions*

After pointing out the differences in how Android is presented by the two organizations, I now proceed to discuss the *subject positions* that are formed within the documents I have analyzed. Both organizations use the omnipresent 'we' as a rhetorical tool, but investigating who is considered to be part of the 'we' shows that Google and the FSFE have different communities in mind.

#### *Google: everyone is a contributor*

While promoting Android as a truly open OS where everyone can use the code, but also maintaining strong control over the platform, Google had to reconcile openness with control. Android's official site, next to a page with the title "Philosophy of Android" (where the choice of license is explained), is a webpage "People and Roles" (solely dedicated to possible roles that one can take in relation to writing code for Android). Table 6 presents the roles listed by Google, which are assessed in terms of the four freedoms (to use, to study, to modify, to integrate/publish back changes in the OS), and indicates the organizational boundaries of Google. What is striking in the way Google defines subject positions is the fact that, while everyone can use Android and contribute code to it, reviewing and accepting code is a role exclusively reserved for Google employees ("Peoples and Roles of Android Open Source Project", 2012).

**Table 6** Division of roles in relation to the code production of Android

| **Signifier** | **Use** | **Modify** | **Contribute changes** | **Integrate/publish changes** | **Employed by Google** |
|---|---|---|---|---|---|
| **Contributor** | Yes | Yes | Yes | No | n/a |
| **Developer** | Yes | Yes | 'missing' | 'missing' | n/a |
| **Verifier** | Yes | Yes | Yes | No | Yes |
| **Approver** | yes | Yes | Yes | No | Yes |
| **Project lead** | yes | Yes | Yes | Yes | Yes |

This is a crucial distinction because on the one hand the c*ontributor* emerges as a new subject position and dislocates and mitigates existing differences between various groups (e.g., between the handset manufacturers and independent developers) who use and write code for Android.

### *FSFE: we the community*

Much like Google, the FSFE also uses the 'we' identifier when referring to a 'community'. Instead of a *consumer* or an *end-user,* the FSFE refers to the same individuals as *users* and *citizens*. The activists present the community as inclusive, as they aim to include a wide range of users. In the campaign texts, it is explicated that it is not anymore necessary to be a developer or code-literate to be part of the community. "*Even though you may not have the skills to directly exercise all of your freedom, you will benefit from a vibrant community that can do it together.*" (FSFE, "Free Your Android!", 2012).

In terms of antagonisms between groups, the FSFE antagonizes users' interests against corporate interests (referred to as: "companies such as Apple", "profit-driven firms"). This is in stark contrast with Google's articulation, where the relationship between consumers and industry is neutralized to a degree that there are no conflicts of interest and the only threat is the monopolization of the market and/or fragmentation of the platform, which according to Google would result in bad user experience.

### *Social imaginaries on mobile technology*

In terms of social imaginaries on mobile technology, both discourses are embedded in the understanding of technology as something utilitarian, in particular as ‘tools’. However, there is a difference between how Google and free software activists give meaning to the same technological artifact: smartphones. While Google emphasizes the entertainment aspect of smartphones, naming them as a new “consumer gadget”, which is tied to fun and the communicative needs of users, free software activists emphasize the computer-like functionality and the private character of these devices.

The FSFE volunteers deliberately avoid the word smartphones (due to its commercial connotations) and refer to devices as small computers to stress the relevance of having free software for mobiles. Through Google’s texts, technological artifacts appear as neutral end products, which are a response to harmonized market relations (industry responding to consumers’ needs). Meanwhile, in the FSFE texts, the same relationship is problematized because the Foundation advocates that control must be relegated from centralized corporate actors to individual citizens.

## Discussion

The analysis shows that Android is only *conditionally open*. This is because Google does not involve developers directly in the modification of the OS, which is one of the key features of open source development, and secondly, in order to write an app for Android, an individual developer or enterprise must agree to the requirements of the Android Compatibility Program. When developers do not agree, their access to the functionalities of an Android smartphone is very limited, which makes it almost impossible to build attractive apps (e.g. without push notifications, integrated location). In the political discourse theory terminology, one could argue that Google’s use of *open-source* resembles a hegemonic intervention (Mouffe, 2008), where a particular understanding is anchored through other means rather than only discursive. In this case, such means are the techno-legal obligations to agree to Android’s compatibility program.

In addition to restricting access through Google's compatibility program, releasing the AOSP under a permissive license is arguably negative for the free software community, because instead of "freeing information" such licenses may provide more shades of ownership and producer-control (Kleiner, 2010, p. 35). Likewise, one may argue that by adopting an open-source but permissive license instead of a free software license, Google is able to benefit from the contributions of the open-source community, without the obligation to reciprocity.

On the other hand, the discourse on free software in the mobile context bares the same rationalization as in the context of personal computers (Berry, 2004). The definition is anchored in the developers' four freedoms and formulated within a human rights framework. However, I also saw an attempt to connect free software to the protection of the privacy of users, due to the personal character of mobile devices and alleged surveillance practices by corporate and governmental actors. This can be seen as a response to expanding the discourse on free software into an "emancipatory struggle" (Sullivan, 2011).

Nonetheless, this fails to provide a solid rationale for the call for the democratization of technology and increasing social control of the production of code. I suggest that stressing the prescriptive function of code (comparable to the law) might enable digital rights activists to increase the relevance of the call. So, reconceptualizing code as inherently political, where the political aspect is defining the meaning (or affordances) of the technologies, can potentially increase the significance of free software for the wider public. I argue that this can be further developed to provide a more rational and emancipatory view on code, which escapes both the cunning de-politicization of the development of technologies by Silicon Valley companies and avoids claims on the rights and duties of neoliberal subjects.

## Conclusion

This chapter focused on the discursive construction and contestation of Android, a particular (yet dominant) operating system for mobile phones, between two organizations: Google Inc. and the Free Software Foundation

Europe (FSFE). Using analytical tools from political discourse theory (key signifiers, nodal points, chains of signification), I dissected texts produced by these organizations to understand how these actors construct the meaning of Android vis à vis ownership and control of (mobile) code. Texts produced within the organizational boundaries of Google and the FSFE show evident antagonism and are embedded in opposite discourses. Google evokes and draws on the discourse on open source and marginalizes free software as an effort outside a new community (of "contributors" to Android). The FSFE's texts, on the other hand, are explicit in their opposition to open source and contain strong deontological claims. Yet what unites them is that both are rooted in certain individualistic, liberal understandings of social relations. On the one side, the justificatory horizon is captured by consumer rights (and needs), and on the other by the rights of users (developers). Through this discourse analysis I explicated that Google promotes Android as an open platform, but through its use of compatibility redefines the meaning of such. Free software activists, on the other hand, expand the discourse on privacy and problematize the absence of free software as a danger to democracy, yet do not address the commodification of user data by platform owners like Google.

In closing, if we take into account the market position of Android in relation to other mobile operating systems, we can argue that, in contrast to the OS market for PCs, we see a dominance of open source in mobiles. However, the open-source development of Android is under the tight corporate control of Google. If we rephrase this in metaphorical terms, Android is still a cathedral rather than a bazaar,[42] however, everyone can contribute to it, either by writing code actively or by using Google's 'free' services.

[42] The metaphors of "bazaar" versus "cathedral" were used by Raymond (2001) to denote different methodologies for producing software: open source versus closed/proprietary.

# Chapter 6 Libre Software for Mobiles

## A qualitative inquiry into the meanings and practices of writing FLOSS apps for smartphones

In this chapter, I present a qualitative inquiry into the state of Free/Libre and Open-Source Software (FLOSS) for mobiles from the perspective of digital rights activists and lead developers of free software projects for mobiles. FLOSS has attracted considerable attention in academia; however, in the context of mobile computing, it is lesser-known. This is because free and open-source software in the mobile domain came to be hegemonized by Google's Android. In the context of the Google/Apple duopoly, Google's Android is commonly considered to be free and open source. However, closer inspection of Google's practices reveals limitations to this claim (Mosemghvdlishvili & Jansz, 2018).

Despite its potential, free and libre software for mobiles is currently confined to a smaller community of hackers and digital rights activists. A potential reason for its relative invisibility is the fact that free software is not searchable by any official app store. Thus, the vast majority of smartphone users do not get exposed to it. This lack of visibility means that digital rights activists must rely on campaigning, alternative networks (such as mailing lists, wikis), and organizing physical meetings to advocate alternative apps. It is within this context that the present study aims to explore and understand how FLOSS activists make sense of free software and engage in its development in the context of mobiles.

Chapter 6

# Introduction

Due to its peer-to-peer (P2P) model, voluntary participation, and egalitarian structure, Free/Libre and Open Source Software (FLOSS) has attracted much scholarly interest across the social sciences, from new media studies to political economy, legal studies to social movement studies. In various works, FLOSS has been theorized as a better and more ethical way of developing computer programs due to the transparency of the development process and public accessibility of its instructions (Chopra & Dexter, 2008; Powell, 2012; Stallman, 1994, 2002); as a more efficient methodology capable of producing better quality software than more traditional, closed corporate R&D (Raymond, 1999); as a critique of the capitalist mode of production and the existing intellectual property regime (Kelty, 2008; Rigi, 2013; Söderberg, 2015); and as an example of a political movement for social justice (Calderaro, 2011; Sullivan, 2011).

Through the introduction of smartphones (programmable phones), FLOSS was extended into the mobile domain as well. Before 2007, writing software for mobile devices was mainly in the hands of device manufacturers and mobile network providers. It was hardly accessible to hobbyist programmers. In the short period between 2007-2008, competing mobile platform providers (iOS, Android, Windows Phone, etc.) began allowing third-party developers to write software (i.e., apps) for their devices (Mosemghvdlishvili & Jansz, 2013).

In the context of mobiles, a platform is an integrated architecture of standards primarily centered around an operating system that allows the development of complementary technologies and is controlled by one or several firms (Mian, Teixeira, & Koskivaara, 2011; Teece, 2012, 2018). Platforms as discursive, organizational, and competing structures were already formed in the personal computing and gaming industry (Teece, 1986; West, 2003). Following the penetration of Web 2.0 across many areas; however, the term platform was increasingly used to describe online content hosting and service providing intermediaries (e.g., YouTube, Myspace, Facebook). As Gillespie (2010) wrote, because of its connotations of representing an "open, neutral, and progressive support for activity" (p.352), the platform became a deceptive metaphor because it actively concealed the limitations of the platforms by implying their technical neutrality. Recently, Poell, Nieborg and Van Dijck (2019) suggested that *platformisation* can be seen as a reorganization of socio-cultural practices around new (digital) spaces that rely on the systematic collection, algorithmic processing, and monetization of data. This process is not confined to the mobile domain. On the contrary, it permeates different economic sectors and areas of life.

In the context of mobiles, since 2012, Android has maintained the dominant position among mobile platforms, holding up to 70% of the global market share. Android became "a de facto standard" (Ippolita, 2013), comparable in its market dominance to Microsoft's Windows during the PC era. Google's Android may resemble Windows in terms of its dominant position on the market, but it differs in terms of how it is provided to other firms. The Windows operating system is a classic example of a proprietary program, where copies of the software are sold to other firms or consumers. In contrast, Android is open source released under Apache 2[43] license, and given without monetary payment *for free*[44] to a range of handset providers. Proprietary versus open-source was initially considered "the two extremes" of how a platform could be organized (West, 2003, p. 1259).

Platform politics has become a pressing topic within new media studies. As Poell and colleagues have stressed (Poell et al., 2019), the relationship between platform operators and end-users is extremely volatile and "structured by fundamentally unequal power relations" (p.6). An example of the asymmetry of power, according to the authors (ibid.), is offered by the app

[43] Apache 2 is a permissive free and open source license, meaning it does not have a reciprocity clause and it can be combined with proprietary licenses. For a wide range of actors such as hand-set manufacturers, this is more acceptable than the GPL license with its strong copyleft clause.

[44] One of the reasons why free software activists came to increasingly use the word Libre (that stands for liberty) is to avoid conflation with the word free (as in free from payment).

stores. They point out that, whether with iOS or Android, app stores serve as key aggregators of datafication, capturing ubiquitous user data about users. Smartphone users do not need to download software from distributed locations, as was commonly done on PCs, but rather go through respective app stores. Concentration of user data have been extensively problematized as an unfair power relationship leading to economic (consumer) surveillance and audience commodification (Fuchs, 2011b, 2012; Kang & McAllister, 2011).

Despite the validity of the above criticism, there is still a significant difference between Apple's iOS and Android: namely, it is not possible to install an alternative *app store* on iOS, or to circumvent the approval procedure by Apple. The only possibility is to jailbreak the iPhone, which will cancel out the user warranty. On Android, in contrast, a user can install an alternative app and use it instead of Google Play. This seemingly small difference is vital for free software activists. As will be explained later in this chapter, one of the most significant projects in mobilizing free software in one repository is F-Droid, which serves as an alternative app store for proponents of FLOSS software. Thus, because Google (unlike Apple) also provides part of the OS itself as open source, FLOSS activists can modify the operating system. Consequently, there are more custom modifications of Android OS than there are for iOS.

The remainder of the chapter unfolds in the following manner: First, in the literature review, I will contextualize FLOSS[45] and review critical arguments

[45] Among studies represented in this literature review, differing terms are adopted by various authors; for example, scholars from organizational studies (Kogut, 2001; Von Hippel, 2008; Von Krogh & Von Hippel, 2006) exclusively use the term *open source*, whereas researchers from new media studies and sociology (E. G. Coleman, 2009; Lin, 2005; Sullivan, 2011) use *F/OSS* or *Libre software* (Dalle & Jullien, 2003; Robles, Amor, Gonzalez-Barahona, & Herraiz, 2005). Libre, in this context, is borrowed from the French "libre" which refers to freedom. By using a word that only has the meaning of freedom, as opposed to "free" in English that can also denote without charges, some authors emphasize the copyleft and freedom-based character of such software. Occasionally, an acronym FLOSS (as in Free, Libre and Open

advanced in the scholarly debate concerning its political and socio-economic implications. Details about the study design, criteria of the purposeful sampling, guiding research questions and the process of interviewing and analysis will be explained in the methodology. Following this, the thematic analysis will be presented, leading to discussion and conclusions.

## Literature review

This literature review embeds the current study in the scholarly debate about the nature and implications of FLOSS and contributes to the existing multidisciplinary body of work. In the following pages, the relevant historical context is first outlined, and the most decisive moments in the existence of FLOSS, from its emergence to its institutionalization, are concisely presented. Respective periods are introduced in order to demarcate the most notable events and changes over the 40-year existence of FLOSS. For this purpose, insights from multidisciplinary academic literature are incorporated, notably from *media and digital anthropology* (Berry, 2008; Chopra & Dexter, 2008; E. G. Coleman, 2012; Kelty, 2008; Söderberg, 2015), *organizational and innovation studies* (Elliott & Scacchi, 2008; Von Hippel, 2006), and *political economy* (Bauwens, 2005; Kostakis & Bauwens, 2014; Kostakis & Stavroulakis, 2013). Following this overview of how FLOSS came to develop, spread, and become institutionalized, critical theoretical tensions concerning two main topics, the (a) political nature and (b) socio-economic implications of FLOSS, are reviewed.

### *Making of FLOSS (1982-1998)*

The principles of FLOSS are commonly traced back to the early days of computing when sharing code was not only a common practice, but the default mode of operation (Elliott & Kraemer, 2008; Kelty, 2008; Söderberg, 2015;

Source Software) is also used (Joode et al., 2006; Lin, 2005). In this chapter, I use FLOSS, as it reflects both the Free/libre as well as Open Source aspects.

Tuomi, 2002). An event that preceded the legal construction of FLOSS was the extension of copyright to computer code. This change meant that the traditional ways of accessing and freely exchanging computer code were restricted. The "commodification of software" (Coleman, 2012, pp. 64) sowed discontent in the hacker community, which viewed computer code and software as collective resources (Kelty, 2008; Levy, 1984; Söderberg, 2015).

Some commentators (Dafermos & Söderberg, 2009; Lessig, 2004; Schoonmaker, 2007) have contextualized the creation of free software as a reaction to the enclosure of computer code within the intellectual property regime. Leading technology firms such as IBM and Microsoft were instrumental in lobbying to pass legal amendments, which would enable the copyrighting of software (Kostakis & Bauwens, 2014), just like a piece of literature or any other artistic work. Instrumental in creating an alternative that came to be known as free software was work by Richard Stallman within the GNU project[46] (see for detailed accounts: Coleman, 2012; Kelty, 2008).

The development of the General Public License (GPL) was crucial in establishing FLOSS as a legal, technological assemblage. What made the GPL software license "a clever legal hack" (E. G. Coleman, 2004) was using copyright itself to ensure that copyright would not enclose the software. By using the very same legal system that he was opposing, Stallman inverted the aims of the copyright. Instead of reserving all rights, he waved the four rights to use, to study, to modify, and to share to end-users. Such use of copyright came to be called *copyleft*. Thus, in the GPL license, the copyright was used to ensure that users were able to access, use and change the software.

Alongside the alternative use of copyright, another essential feature of GPL was the so-called "viral clause," meaning that derivative works made from software protected by GPL were automatically licensed under the same license

[46] The GNU was a recursive acronym and stood for GNU's Not Unix! This meant that GNU was a Unix-like system, but that it did not contain Unix code and was free software.

(Ross, 2013). This is also known as the *reciprocity principle* and is the distinguishing characteristic of copyleft licenses. Through copyleft, free software activists secured “legal autonomy for software production” within the rapidly changing intellectual property regime, which was adopted almost univocally on a global scale (E. G. Coleman, 2012).

The first period in the making of FLOSS paralleled broader socio-economic processes through which property rights were extended to information and forms of knowledge. Boyle (2003) suggested that the ownership of information was becoming a vital source of power comparable to ownership of land, material resources and means of production. He argued that, by turning hitherto non-copyrightable and non-patentable information and knowledge into property, a “second enclosure” was taking place. Some political economists saw such legislative changes as an “urgent concern for ordinary citizens,” leading to information feudalism (Drahos & Braithwaite, 2002).[47]

### *Fragmentation and the birth of the dual regime (since 1998)*

In 1998 another initiative called the Open Source Initiative was created within FLOSS, which emphasized that free software was not a political matter or social movement, but simply a better methodology to develop software (Raymond, 1999). The emergence of open source as discourse as well as associated licenses was the result of the “efforts of a group of people within the free software community to make the concept of free software seem less threatening to business” (Ross, 2013, p. 203). It must be stressed that technically free software is by definition open source; there is no other

[47] In a detailed case study, the authors documented how a new intellectual property regime was detrimental for patients in South Africa, where, due to new legislation, it was not possible to manufacture drugs to treat HIV/AIDS. They feared that such a regime was leading to an era of information feudalism comparable with medieval feudalism: not through ownership of lands, but of patents on intellectual property, and called for more democratic property rights.

possibility. On the other hand, it is possible to have open-source software but limit the usage of the code or allow its enclosure.

The prime differences between free and open source software were in the types of discourses they were embedded in and which values were stressed as important (Berry, 2004; Berry & Moss, 2006). The following quotes from two prominent representatives of these respective movements help illustrate their emphasis on different values.

> *Software design and implementation should be a joyous art, and a kind of a high-level play. If this attitude seems preposterous or vaguely embarrassing to you, stop and think; ask yourself what you've forgotten. Why do you design software instead of doing something else to make money or pass the time? You must have thought software was worthy of your passions once... To do the Unix philosophy right, you need to have (or recover) that attitude. You need to care. You need to play. You need to be willing to explore.* (Raymond as quoted in Himanen, 2004, p. 423)

The second quote, by Stallman puts emphasis on different values, stressing its social relevance:

> *The free software movement is a campaign for computer users' freedom; we say that a nonfree program is an injustice to its users. The open source camp declines to see the issue as a matter of justice to the users, and bases its arguments on practical benefits only.* (Stallman, 2016)

These two movements, led by two competing organizations (the Free Software Foundation and the Open Source Initiative), came to endorse different sets of licenses. The former supported those compatible with GPL and copyleft licenses, and the Open Source Initiative recommended both permissive as well as copyleft licenses.[48]

Open source became associated with a pragmatic and apolitical approach: as an efficient methodology of software development framed as a creative process and an individual act of play and art. In contrast, free software (also referred to as *libre* software) came to be associated with protecting freedom of speech by equating access to source code with the individual right to freedom of expression. However, it was also often marginalized as radical or socialist, which commentators have justifiably questioned due to the fact that the driving ideology of free software was never embedded in socialism or communism, but rather in a liberal understanding of individual rights (see Berry, 2004; Coleman, 2012).

The creation of open source, along with its associated, less restrictive licenses, led to the "the birth of mixed regime" (Kostakis & Bauwens, 2014), leading to the integration of FLOSS in proprietary infrastructure and corporate services. Even companies that were vocally against free software two decades earlier (e.g., IBM) came to embrace open-source development and began incorporating it into their processes (Coleman & Hill, 2004; Joode et al., 2006; Schoonmaker, 2007). The process, as Gehring put in words, was a rather "ironic" development (2006).

Over time, FLOSS gradually became more widely accepted and was produced by tens of thousands of individuals across the globe who worked together in a peer-to-peer (P2P) manner. FLOSS projects (e.g., Linux) resembled a "fractal organization," comprised of a community composed of smaller subcommunities (Tuomi, 1999).[49] Empirical studies consistently reported that

[48] Over the past decade, the number of licenses has mushroomed. According to the latest count, by 2019 there were up to 200 different licenses available.

[49] In absolute terms, the largest number of contributors were from the USA, but in per capita terms European countries were leading. An analysis of credit files (Tuomi, 2002) showed that, in 1994, among a total of 407 contributors, 167 developers came from EU countries, with Finland, the Netherlands and Denmark being at the top of the list. (pp.170-171).

FLOSS software was written by loosely coordinated efforts of individuals who engaged in active communication and deliberation in small subgroups (Croeser, 2012; Crowston & Howison, 2005; Krishnamurthy, 2005).

### *Political and socio-economic relevance of FLOSS*

#### *FLOSS as a political movement*

Questions about the political nature, governing ideology and greater implications of the FLOSS movement have been explored in different academic fields, including new media studies, digital anthropology, sociology, legal studies and managerial studies. A recurring question across these fields is whether the free software community can be considered a social movement. If yes, what ideologies are shared by its members, and to what extent is it possible to extrapolate coherent political and social justice claims from its discourses?

Scholars writing about these questions tend to advocate one of two positions. Some maintain that FLOSS is a social justice movement in itself, while others reject the existence of a coherent driving ideology and describe the community as heterogeneous and apolitical, despite their actions having substantial, unintended political consequences.

Among proponents of FLOSS as a social movement, Sullivan (2011, p. 232) writes that FLOSS is “expanding its initial constituencies of software hackers and users to tackle broader social justice aims.” Issues such as digital rights, consumer sovereignty (as in owning and managing one’s own device and not being tracked), and information commons (open data initiatives) have become part of the FLOSS movement. Lievrouw (2011) also discusses FLOSS as a social movement, but more as "an alternative/activist new media genre" (p.100). For her, such alternative activism amounts to writing software and programs to escape censorship and surveillance, to improve encryption, and protect the privacy of users. Similarly, Elliot and Scacchi (2008) frame FLOSS as a new genre of computerization movements.[50] Berry (2004), in his study of discourses associate with free software and open source, concludes that these are movements that differ based on ideological grounds, with both aiming to win the discursive struggle.

Views expressed by the aforementioned authors seem plausible when FLOSS is discussed as one coordinated movement. However, empirical studies of how FLOSS communities actually work present a more complex picture. Based on ethnographic research, Coleman (2013) writes that, during her fieldwork with FLOSS developers and activists, she consistently witnessed "a reluctance to signify free software beyond a narrow politics of software freedom" (p. 159). She describes an active yet very politically heterogeneous group and argued that the central feature of the FLOSS community is its "political agnosticism" (p.160). She contends that "FLOSS has been able to escape the various ideological polarizations (such as liberal versus conservative) so common in our current political climate" (p. 22). She further argues that the strength of FLOSS is in its (a) transparency and (b) in articulating criticism of intellectual property law and mainstream economic theory through its techno-legal design.[51] "Its most profound political effect," she maintains, "has been to devitalize the hegemonic status of intellectual property law and catalyze a series of transformations in the arena of intellectual property law" (p. 160).

The transparency of FLOSS is primarily supported by embedding it in freedom of speech, where access to source code and deliberate discussion about projects are tenets and characteristics of FLOSS development (E. G. Coleman, 2009; E. G. Coleman & Golub, 2008). For example, case studies conducted by De Paoli and Teli (2006), who explore controversies regarding

[50] Computerization movement (CM) refers to a movement where computing is linked with advancing/improving certain social problems. For more on this see: (Kling, 1991; Kling & Iacono, 1988).

[51] Meaning copyleft and other free software licenses, practice of collaborative development, and visibility of code, among others.

FLOSS licenses and how these are deliberated among members, further demonstrate that the FLOSS community is politically heterogeneous, yet characterized by the transparency of negotiation of the decision-making process (De Paoli, Teli, & D'Andrea, 2008; Teli, 2005; Teli & De Paoli, 2006).

How is it possible to make sense of this "apolitical politics" (Söderberg, 2017, p. 970) of the free software movement? In this study, I find arguments developed by DeLanda (2001) and Kelty (2008, 2013), who both stress the material and configuration aspect of the movement, to be especially helpful. In his essay on the philosophy of the movement, DeLanda (2001) maintains that the strength of FLOSS is not in its political ideology, per se (which, in a fragmented form, does exist), but rather in its *configuration* and *design*. Thus, what mattered most about the movement for him is not the "intentional action" of its leading figures but rather the "unintended collective consequences" of the movement itself. These consequences are enabled by engrained values in the design of the software development model and the development process itself, meaning its main license (GPL) and the design of its production model (P2P).

Similarly, Kelty (2008) maintains in his ethnographic study of free software that the ideological aspect is "just one component of Free Software and, indeed, the one that has come last, after the other practices were figured out and made legible, replicable, and modifiable" (p.302). What he means by "practices" is how free software is constantly changing, emerging, and being contested and negotiated in a public manner. According to Kelty, the real power of free software is in its ability to translate "these values into material objects" (Kelty, 2013).

Thus, DeLanda (2001) and Kelty (2008, 2013) both emphasize that the political aspect of FLOSS is not as directly visible as in partisan politics, but rather is embedded in its design and practices; in the radical deliberation and involvement in the decision making process; and the resistance to and legal autonomy from intellectual property. This leads us to discuss the second major implication of FLOSS, which concerns the socio-economic organization of labor and re-orientation between knowledge and property.

### *FLOSS as a mode of production*

The way FLOSS's development process is organized, outside wage labor, and mainly through voluntary involvement, made it puzzling. Inquiries primarily from the perspective of economics, business, and innovation studies (for reviews see: Aksulu & Wade, 2010; Von Krogh & Von Hippel, 2006) aim to understand what drives free software. Why are hackers and hobbyists willing to spend time and skills to create a shared product (Hars & Ou, 2002; K. R. Lakhani & von Hippel, 2003; K. Lakhani & Wolf, 2003)? Through surveying contributors to FLOSS projects, researchers propose a number of motivations. *Altruism* and *individual rationality* (e.g., an individual seeking to improve their own software or build a reputation) are suggested as two main motivations (Joode et al., 2006) that explain such behavior. However, the peer-to-peer (P2P) principle of organization extended to areas other than merely writing code/software (e.g., Wikipedia, Indymedia tools). Thus, the growth of the FLOSS model outside hackerspace and participation in these projects by various actors intensified the need to discuss FLOSS in the context of labor and economic consequences. Following the year 2000, a "consensus definition" (Kreiss, Finn, & Turner, 2011) emerged among a range of scholars that *a new mode of production* was evolving (Bauwens, 2005; Benkler, 2006; Benkler & Nissenbaum, 2006), one that was having a profound impact on modern societies. While these commentators mostly agree about the rapidly changing nature of work, collaboration, and organization, they remain divided on the question of *how the new mode relates to the existing capitalist mode of production.*

For example, Benkler (2006) theorizes that the emergence of this new mode of production (which he calls the third mode of production) is developing alongside the other two (industry-based and market-based) modes of production. It is characterized by its non-monetary and decentralized nature and is caused by the reduction in the price of ICT technologies and connectivity of the internet. "The quintessential instance" of such a new mode of peer production Benkler refers to free and open-source software development. Notably, in his work, the third mode of production is neither a threat nor an alternative to existing modes, but rather a parallel development with profound consequences.

Bauwens (2005), too, theorizes P2P production as a *"third mode of production"* but with a significant difference from Benkler's perspective. Most notably, the relationship between the new modes of collaborative production and the existing capitalist market structures are contested. Bauwens stresses that P2P depends on the capitalist market, a market that in turn, is increasingly becoming dependent on P2P distributed networks. He also notes that, on an individual level, members of P2P communities cannot make a living from peer production. Even though they may derive meaning from such work, they still rely on other forms of market and industry-based relations for income and subsistence. A second major factor contributing to Bauwens's more critical perspective is the increased integration of P2P infrastructure (collaborative practices) within existing modes of production.

The diverging positions of Bauwens and Benkler are illustrative of a broader academic debate concerning the role of labor in FLOSS projects, in which collaborative co-creation outside of monetary rewards can be considered either as a form of empowerment and meaningful activity or as exploitation and the provision of free labor (for criticism of free labor see: Terranova, 2000, 2004, 2006).

Unfortunately, a more in-depth discussion of the critiques of FLOSS as free labor exceeds the scope of this literature review. It may be pointed out, however, that some of the solutions voiced in the critical political economy include proposals to strengthen commons-based production by institutionalizing it (Bauwens & Kostakis, 2014; Bauwens, Kostakis, & Pazaitis, 2019; Kostakis & Bauwens, 2014) and preventing commercialization by strengthening copyleft (De Filippi & Vieira, 2014; Kleiner, 2010; Vieira & De Filippi, 2014).

In addition to the aforementioned critiques, there is also a notable body of criticism about the social consequences of FLOSS rooted in the work of feminist scholars. FLOSS governance and decision-making processes have been contested by feminist scholars for being incapable of producing a fair system due to deep gender inequalities (Ford & Wajcman, 2017). The consequences of traditional gender role divisions are vividly visible in the light of the drastic absence of women in FLOSS communities (Padala et al., 2020). In recent years, another notable development complementing the critical feminist work is the emergence of generative proposals. Bardzell (2010; see also: Bardzell & Bardzell, 2016), among others, has argued for imbuing the design of technologies with emancipatory feminist interaction design qualities such as pluralism, participation, ecology, embodiment, and self-disclosure.

## Research questions

Against the backdrop of this previous academic work, the present study now directs attention to FLOSS practices, specifically in the mobile domain. Through in-depth interviews, I aim to provide insights into and a deeper understanding of the state of free software for smartphones. The three main questions guiding this study are*:*

> *(1) How do FLOSS activists construct the meaning of free software in the context of mobiles?;*
>
> *(2) How do FLOSS activists evaluate the dominant platform Android from a free software perspective?; and,*
>
> *(3) Which initiatives do they consider the most important for bringing free software to mobiles, and why?*

In the following section, the methodology of the study will be explained, including how the interviews were conducted and analyzed.

Chapter 6

# Methodology

This study aims to contribute to the empirical exploration of the perceptions and experiences of being a free software activist in the context of smartphones. The sampling strategy was purposeful and aimed at identifying individuals who were leading developers and vested advocates of FLOSS, meaning they were involved in the movement for a long time. The participants involved had to have been committed to FLOSS for more than ten years and to have initiated a new project, performed the role of lead developer, or have served as a maintainer of FLOSS projects.

In order to identify and recruit participants, I observed and followed the email newsletter group "Free Your Android!" during the years 2012-2017. The group was comprised of up to 200 subscribers and was active in initiating, discussing, and sharing information on FLOSS projects for mobiles (mainly on Android-run devices). The newsletter was a prime channel to inform other activists about new ideas for FLOSS projects, to invite collaborations, ask for translations[52] of materials, or plan an event or a campaign.

Through long-term participant observation, I identified five key individuals who were the most active in the internal communications of the group. I sought contact with these individuals and discussed their involvement in the FLOSS community in greater detail. For those interactions, the majority of participants required the use of encrypted email communication. Two individuals refused to use anything other than free software for voice communications as well. Following initial inquiries into the activities of the potential respondents, I selected the three individuals whose projects were directly connected to mobile devices (as opposed to FLOSS for PCs) and invited them for an interview. The remaining four participants were recruited through the snowballing method. Between 2017-2018, seven in-depth interviews were conducted and recorded (through phone conversations). The

[52] By 2019, Free Your Android! campaign was available in 8 languages: English, Dutch, German, Russian, Turkish, Albanian, Italian and Greek.

interviews lasted between 50 to 90 minutes each. After transcription of the interviews, a follow-up written communication proceeded in order to clarify aspects that remained unclear to me. Participants were also sent a copy of the verbatim transcript for their reference. By choosing phone conversation as a medium for communication and in-depth interviewing as a research method, rich data was collected. In its transcribed form, the material comprised 32,246 words.

### *Thematic analysis*

The verbatim transcribed interviews were coded and served as materials for subsequent thematic analysis. According to Boyatzis (1998), thematic analysis is a process for encoding qualitative information that enables researchers and scholars to understand and interpret observations and to access a wide variety of phenomenological details. In various disciplines across the social sciences, thematic analysis has frequently been used without overtly naming it (Braun & Clarke, 2006). Over the past decade, however, interest in the method has increased dramatically (see for review Braun & Clarke, 2019). A theme is a pattern in the qualitative information that describes and organizes the observations and interprets aspects of the phenomenon. It is "a thread of underlying meaning implicitly discovered at the interpretative level" based on the subjective understandings of participants (Vaismoradi, Jones, Turunen, & Snelgrove, 2016, p. 101). A theme needs to represent some level of patterned response to the research questions (Braun & Clarke, 2006).

In thematic analysis, the process moves across phases from familiarizing oneself with the material to identifying meaning units and coding, categorizing and searching for themes. In this study, the four-stage process proposed by Vaismoradi and colleagues (2016) was adopted. The first phase of analysis entailed: *Initialization* - reading and familiarizing myself with the transcripts, highlighting meaning units, coding, and writing reflective notes. During the Initialization stage, transcripts were first coded through in-vivo coding. In-vivo coding is verbatim coding, aimed at capturing the voices of participants in their own words (Saldaña, 2015) by using words and phrases as uttered by participants. It is an open-ended coding method that enables the researcher to capture rich data for analytical purposes. During the process of coding, the researcher actively produces reflective memos that provide analytical comments on the coding process. The second phase of *Construction* is comprised of classifying codes, labeling categories, and organizing them. During this process, 19 different categories were created that captured thematically organized in-vivo codes (such as the meaning of free software, disagreement about licenses, the experience of working with AOSP, description of FLOSS project, etc.). Together with retrieved segments of coded materials, analytical memos also served as an input at this stage of analysis. The *Rectification* stage required iterative immersion and distancing from the material by constructing themes from the categories and relating themes to established knowledge. *Finalization* included developing the storyline and deciding on the order of presenting the material. The following themes were constructed during the analysis based on what the participants brought forward during the interviews: Demarcating FLOSS; Reasons for committing to FLOSS activism; Disagreements about licenses; Android: its possibilities and limitations; and FLOSS projects for smartphones.

In the coming pages, the thematic analysis will be presented, enriched, and illustrated by quotes from the transcripts. The presentation of themes unfolds in the following manner. I first introduce participants and describe their stories of becoming an activist. Following this, the phenomena under investigation, i.e., free software in the mobile context, will be demarcated as explained by participants, paying attention to commonalities and disagreements in their perceptions. The two remaining themes, "Android and its discontents" and "FLOSS for smartphones," cover how activists experience working with the Android platform as well as what they describe as the most important FLOSS initiatives for smartphones. The conclusion will address core insights gained from the empirical inquiry and how these relate to the previous scholarly work. As a final note before presenting the thematic analysis, I would like to pause and provide reflexive bracketing of my own dispositions regarding the topic under investigation.

### *Reflexive bracketing*

Qualitative research increasingly acknowledges the need for reflexivity and self-disclosure of the researcher. Agger (1991) challenged the idea that in qualitative research, any interpretation by a researcher can be understood without reference to her social position and the context of the research. The bracketing of a researcher's prior assumptions and social position vis à vis research phenomena is increasingly expected as a part of methodological rigor and to meet ethical standards (Ahern, 1999; Darawsheh, 2014; Guillemin & Gillam, 2004; Mauthner & Doucet, 2003). After years of reading academic work on the social shaping, political economy, and politics of contemporary mobile technologies, I developed a view of free and open-source software as an ethically better way of creating knowledge than the proprietary model. I have not participated in the community and have not developed code or contributed to FLOSS projects. For the purposes of this study, I have acknowledged my preconceived views on the ethical aspect of FLOSS. Therefore, while working on the interview analysis, I wrote reflective notes to examine my perceptions and to bracket these out from the respondents' stances. To the greatest degree possible, I consciously removed my personal understanding of the phenomena and approaches from the analysis and endeavored to reconstruct for the reader the participants' expressed perceptions about free software and their experiences of being an activist.

### *Introducing participants*

This study takes the perceptions and views of the activists as its starting point. It is through the understanding of the participants' experiences of writing and advocating FLOSS projects for mobiles that the phenomenon is described. Before describing the themes that emerged through the analysis, the participants will first be introduced. Table 7 presents an overview of the participants' educational background, occupational sector, country of residence, and years of involvement in the FLOSS movement. What was shared among all participants, as seen in the years spent within the movement, was a vested interest in and long-term involvement with FLOSS projects.

**Table 7** Participants' background and involvement in FLOSS projects

| | Country of residence | Education background | Domain of current professional activity | Involved in FLOSS |
|---|---|---|---|---|
| M1 | Germany | Philosophy and computer science | Non-profit advocacy | Since 1995 |
| M2 | France | Engineering | Employed in the computer industry | Since 1985 |
| M3 | USA | Computer science | Entrepreneur | since 2000 |
| M4 | USA | Mobile communications & engineering | Entrepreneur | since 2000 |
| M5 | Germany | Political science | Non-profit advocacy | Since 1990 |
| M6 | Denmark | Theoretical Physics | Employed in the computer industry | since 2000 |
| M7 | Albania | ICT engineering | Non-profit foundation | Since 2005 |

# Analysis

### *Becoming a FLOSS activist*

By collecting stories and memories of the first use of FLOSS software, connecting with other developers and getting involved in FLOSS projects, I aimed to reconstruct the paths that led the participants to identify as FLOSS activists.

> *I was using it* [free software]*, and it was good, but then I got more and more interested, digging into it, looking into morals behind it […] And I think, maybe it's the ideology part that drove me from that on, to say yes, this is indeed way to, to achieve better society, a better way to organize our society, if you share knowledge, if you use free software, if we put users freedom first, and then second, also you search for*

> *programs and see community behind it., you see that there are people behind; people who think, people who make decisions, based on something they discuss in the open; they have communities, and that was something different than having a product, a software product that you can buy, and this was not related to the people behind, and that was then the magic also. The magic of seeing people work for the cause, doing it (M5).*

Getting interested in the political questions concerning how software was developed emerged as the most prevalent theme in entering the FLOSS community. Topics such as privacy, as well as the dependence of social structures on software, were highlighted. The intellectual legacy of Richard Stallman was repeatedly referenced throughout the stories of activists, which is not surprising given that the theoretical foundation for justification and rationale was laid out in Stallman's work.

One participant recalled that even though he got exposed to free software while studying at university in the early 90'ss, he was not initially interested. Due to the increasing dominance of software in structuring social life, however, he eventually changed his view:

> I*n early 2000, I could see we were building a complete infrastructure in society on software, and all of a sudden, Stallman's ideas struck me as very important. At around the same time, I started reading Stallman's work again, and switched to Ubuntu and other Linux systems on my personal computers, and it got into my head, I want to do something too, I want to help building a world where everybody owns and uses free software and I want to stop using any proprietary software myself to the extent that I can*" (M6).

The sentiment that the omnipresence and integration of information and communication technologies across all areas increases the concern and need for free software was voiced by other participants too.

In addition to the political question, what also emerged in the life stories of two participants was the experience of early computing, when sharing source code was the default practice and established way they were socialized into computing and programming. *"As a kid, when I was exposed to programming, it was very much like, 'here this is source code,' so that was just natural to me. [...] that's what computing was, that was sharing source code*" (M3). Similarly, another participant recalled the context of programming software for music:

> *The culture for this specific scene* [music]*, was entirely free software culture, we are talking something like 1993 like you would share music score, which you would send via the program that you used to make music. So my first real experience of getting deep into programming was in the world where [...] default for the most part was all free software, without anyone ever talking open source or free software.* (M4)

A common theme in the stories told by the participants was that personal dissatisfaction with the changes in the computer industry (how software was managed) and how it affected their lives seemed to contribute to a desire to become an activist. For example, the following story shared by one of the participants recalls how work-related experiences were a cause of intense distress and led to a conscious move to become actively involved in FLOSS advocacy:

> *You know, I did work with a technology corporation [name anonymized], they acquired a company I started, which means they owned all of the intellectual property I developed, and all of the patents and I was really disappointed and depressed when they - sort of the company - made decisions that ended my work and they owned it and I had no rights or access to it, couldn't negotiate to open-source it, and it was many years of my life just disappeared, and that was really a wakeup call for me some 15-16 years ago.* (M3)

The story of losing ownership of their work is notable because it points to the volatile and precarious relationship software developers (and workers at large) have while working under wage labor agreements or contracts with large corporations. It also points out how programming is a part of human expression and how the authors take pride in their work, the loss of which can become a source of sorrow and psychological distress.

Among the participants, two individuals worked with tech companies and maintained their involvement in FLOSS advocacy outside their work time and obligations. Three participants were a part of different non-profit organizations and two other participants were entrepreneurs, creating free apps for circumventing surveillance.

### *Gender differences*

Not surprisingly, there was no balance among the participants with respect to gender. Only one participant identified herself as female.[53] Even though an examination of gender differences was not the focus of the inquiry, the life stories told by the male and the female participants pointed to different, gendered experiences. The following three memories shared by the female participant shed some light on these disparate experiences. No male participant told a story about how he had to struggle with his parents for being interested in computers or pursuing a tech career. On the contrary, one male participant recalled how his mother brought him to a social café for learning programming as something "fun for a boy to do." Another participant recalled being "lucky" that his father was also interested in computers, which allowed him to tinker with his first computer and learn coding with his father's help. For the sole female participant, however, the family situation was rather different:

[53] Gender balance was not included as a criterion in the sampling process. The main criterion, as explained in the Methodology, was years of experience being involved in FLOSS projects specifically for mobile devices.

> *All the fights I had with my mom, in my high school years it was because of a computer. I even remember that the only times when I would lie to them was about computers. It was always a problem. My father regretted buying us* [me and my sister] *computers*. (M7)

The second instance of gender difference was made visible when the female participant recalled her education (as an IT engineer), as well as her participation in the national Olympics in mathematics, where she was in the minority:

> *You don't really feel comfortable going to places like this* [being the only girl], *because you have all eyes on you. I think some women after being in this gender unequal spaces for the first time, say yeah, I am not going there anymore. But I didn't care, of course, I felt very uncomfortable, but I didn't leave*. (M7)

Being in the minority was not terribly frustrating, but what did elicit strong emotional feelings in the female participant was being treated differently because of her gender. An instance she recalled during her university years illustrates this uneasiness:

> *For example, one professor, he was like oh let's let girls answer this question. Like he was making a question and hands were raising, and he says: let's let the girls answer it, because they are women, or whatever, let's give them opportunity to be vocal [...] but my problem in general is when it affects my intelligence and intellect, like I would never accept a job just because they are accepting me because I am a woman.* (M7)

The last instance of how a female participant experienced the FLOSS community differently than male participants relates to her involvement in editing Wikipedia articles. She recalls that, while using her real username (where her gender was visible), she attracted many more comments and discussions than when she used a neutral username:

> *When I used to edit online articles for Wikipedia, for example, then you couldn't tell whether it was a female or a male, I set a username*

> *in that way. And during this period I never got attacked, attacked in a way when people are trying to redo your content, because somehow they think yours is not good enough, or they remove it, because they do not agree with your references or stuff like that, but after I got my personal account, I had a short bio in my profile, I was actually blocked three times, I don't think they did it because I was a woman like you are a woman so I am blocking you, but I mean they were more up for arguing, while when I was with the other account, there were like, ok, yeah we are going to check that again in the future, and whatever information was there.* (M7)

The life stories shared by the female participant illustrate particular instances of gender inequality in the FLOSS community. More importantly, however, these narratives point to the different socialization processes, upbringing, and education that might help explain the underrepresentation of women in FLOSS communities. Moving further, I will now introduce the core themes that emerged through the thematic analysis, starting with demarcating FLOSS in the context of mobiles.

### *Situating FLOSS for mobiles*

To understand FLOSS in the context of mobiles, I inquired into how participants constructed the meaning of and explained the importance of free software. In defining free software, Richard Stallman's legacy was frequently evoked. Most participants mentioned him and were aware of his work. Thus, the four freedoms to study, to use, to improve and to share software were used as a foundation on which all other justifications were built.

What helped enrich the formal definition of free software, however, was to hear how participants explained the meaning of free software. The first aspect that was shared by most participants was the question of control, as in *being in control* as opposed to being controlled:

> *Free software means you control your own software, you do not have to use something that was programmed by another party with conflicting interests and you can also use it the way you want yourself,*

> *and not be influenced by what other people do with some applications. It is important also for privacy and for security because when the software is not free it is basically a black box, it is very difficult to know what it does, what bugs are new, and when the software is free it is very easy to audit code and know what exactly the application is doing*. (M2)

Access to source code has always been a precondition for the transparency of the working of software. Moreover, the assumption among the free software community is that it is easier to reveal shortcomings when there is a community that has access to the software's source code. Thus, by using free software even those users who do not code can still benefit.

The second aspect that made free software meaningful for the participants on an individual level was related to software being a *medium that enabled expression* and the creation of something new:

> *Maybe inherent part of it, is that, software is human expression. Software is, you know, there is poetry in software. More, it is, this weird mix of something of a creative output like writing prose, but that functions. You cannot separate those two, it is a functional piece of technology and it is also actively a human expression. Not 100 % the same, but there is always a concept that is kind of fundamental to humanity.* (M4)

In this way, software was situated as a form of human expression that is as important as literature or the free press for democratic societies. Being immersed in coding for hours (even after a full workday), blending it with making music, and expressing satisfaction with having produced good quality code showed how, for some participants, coding was more about expression than about an engineering solution.

While, for some participants, free software was first and foremost about individual expression, all of them touched upon the increased role of software in everyday life and its broader societal importance:

> *Software touches at this point almost every aspect of our lives, and that is increasing. I mean right now, we are talking because of the software, a lot of our communication, and work we do, entertainment, how music is created, how movies are made, how books are written.... It is different because it is impacting kind of everything [...] that is one thing, not necessarily inherent, but in our world the software is controlling everything.* (M4)

In this line of thought, the comparison between freedom of software and freedom of the press was particularly stressed by those activists who were involved in advocacy and promoting free software to non-profit and governmental sectors:

> *You need free press in a free society, so that people can be informed; even if they don't know how to write well themselves, are not journalist themselves, it is still important freedom, and free software I would say is kind of analogous because free software is important for society like I mentioned earlier; So that the society can control functionality of the society that is more and more used. Software is more and more doing like the functionalities of the society.* (M1)

### *Comparison between PCs and Smartphones*

Most participants of this study were already involved in developing FLOSS when smartphones were introduced to the market. Therefore, I inquired into potential differences they experienced while writing free software for PCs and smartphones. Evaluating the introduction of smartphones and its 10-year legacy from the perspective of free software highlighted two tendencies.

First, the pervasiveness of smartphones seems to have accentuated a particular trend in social development where control is largely relegated to software. Second, with the increased accessibility of smartphones, the magnitude of surveillance has also increased;

> *I would say the pervasiveness of smartphones has helped making the question of software freedom more pervasive, more important, but at the same time they* [smartphones] *have also weakened the position of free software because it's much more difficult to own free software on those devices.* (M6)

What the participant means by difficulty relates to the configuration of smartphones. Namely, the fact that a smartphone is composed of different parts and the software layer itself is also fragmented into the firmware (software - installed by handset makers), drivers, operating system and apps. Thus, one 'piece' of software does not fit all models. The proliferation of competing platforms and device models makes this even more complex. As a solution, the free software community maintains lists of smartphone models for which there is already free software available. This has led to establishing temporary networks of FLOSS developers working on writing free software for specific models.[54]

### *Perceptions of FLOSS as a movement*

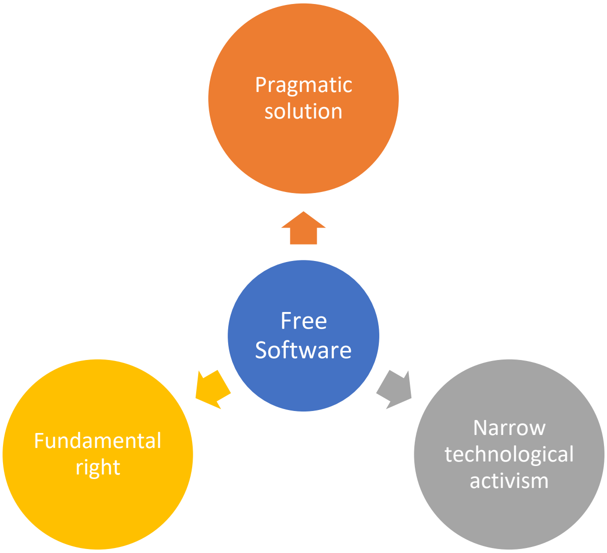
There was common agreement about the definition of free software, as well as a consensus about its necessity and the justifications for why it represents an essential aspect of democratic societies. However, more in-depth inquiry into the meaning and aims of free software as a social movement revealed far less commonality. This divergence in views about the purpose, merits, and boundaries of free software as a social movement is illustrated below in the form of a radial (Figure. 3).

The divergent radial shows the following three positions: (1) perception of the free software movement strictly within the context of the four freedoms and

[54] The most complete list is compiled  and regularly updated by Lineage OS Wiki, a custom modification of Android OS (Lineage OS Wiki, 2020)

placed within a broader justificatory framework of fundamental human rights; (2) as a pragmatic solution to oppose surveillance and provide an alternative way to create software; and (3) criticism of the FLOSS movement as "narrow technological activism." The first position seems to be closely connected to the institutional elements of free software, (i.e., the Free Software Foundation and its sister organizations in Europe and India). We can see this stance as an orthodox form of FLOSS, committed to enforcing, advocating, and spreading the software according to the values that are enshrined in the four freedoms.

**Figure 3** Divergent radial: Perceptions of FLOSS movement



The second position is more moderate; a middle way expressed by the participants who identified themselves as supporters and activists in the FLOSS movement, but who also distanced themselves somewhat from those "who are not willing to compromise" or "traditional digital rights free software folks." The narrative that emerged from this so-called pragmatic approach emphasized the need to bring free software to a wider population, including users who had little technical skill to tinker with code, but who were still concerned with privacy and surveillance.[55]

The third position was articulated by a participant who, despite being actively involved for more than ten years within the movement, felt frustrated by what he called the narrow focus on the technological aspect, in particular, on the software side of technological development. He explained that, in his experience, social movements in Europe were very fragmented and disconnected from each other. Even though many social movements were focused on progressive change (e.g., LGBT rights, environmental protection, movements for indigenous rights), there was little to no cooperation between these movements, which in his view weakened the efforts of activists.

The diverse views about the scope and aims of the FLOSS movement were also expanded on in response to the question concerning licenses and legal aspects. Namely, disagreements were expressed regarding which license was most beneficial to use for the free software movement.

### *Licenses: a pinnacle of disagreement*

As explained in the literature review, the legal construction of free software was made possible by creating the GPL license. GPL is a copyleft license, meaning that, on the one hand, it guarantees that users are able to copy, modify and share software, but also requires them to license the new work under the same conditions.

[55] An example of FLOSS apps in this context is  the so called Guardia Project, a collection of FLOSS apps (released not necessarily with the GPL, but also with other more permissive licenses), which aim to circumvent surveillance, or provide tools such as encryption and anonymous  browsing TOR to smartphone users. The participants who advocated such a stance articulated that it was more important for them to bring certain functionalities (e.g. privacy to activists) and that they therefore had to ‘compromise’ by using permissive licenses.  The apps were highly downloaded outside Europe and the United States, in countries as Iran, Russia, and Cuba, for example.

> *Copyleft means turning copyright on its head; It's using copyright to protect freedoms of free software by saying, I give you this work with all these freedoms attached, but if you give it to other people you also need to give them those freedoms, that I gave you. So that nobody in the middle is able to strip those freedoms away. But that is just one part of free software.* (M1)

What the respondent means by "*just one part of free software"* is that there are many other licenses that guarantee that source code is open and accessible, but do not require the reciprocal licensing. For example, Apache 2, the license used for the large part of the AOSP by Google, does not have such a clause. Thus, it is not a strong copyleft license, but is instead considered a permissive. In principle, through the Apache2 license, "those freedoms" can be stripped away. That is not possible in the case of GPL, as anything created on the basis of such code must be released under the same freedoms.

The reciprocity clause (also labeled as strong copyleft) was considered essential by some participants, but others framed it as an extra feature of free software.

> *It's the copyleft principle which is helping you protect these freedoms, but there are certain instances, where I think, it won't even be helpful. For example, if you are writing a library, software library, in many times, you don't want it to be under strong copyleft license because it makes very different for people who have software under different license to use that. So there you want to use something less protecting to enable widespread usage of your license.* (M1)

The choice to adopt permissive licenses is framed as a "necessary compromise" by one participant, who explains that, for him, it is more critical that the apps he develops (which circumvent surveillance and allow anonymous browsing of the web) reach more individuals. Therefore, he is willing to use the type of license that is suitable to be mixed with proprietary ones and that can be adopted by companies too.

> *I think, it is also important to strategically make compromises to move people to that [free software] direction. I am not saying that it*

> *can be done only this way, and this is the right way to do it, either do the right thing or nothing at all. So a lot of the other licenses, might be false choices if I am trying to get companies piece of software, then I would use the Apache license, and I have the feeling, that is moves the world to the free software so that it is a worthwhile compromise. And I am always happy that there are the people who are not willing to compromise and will point out and say hey this is a compromise.* (M3)

In this quote, the participant is explicitly referring to individuals such as Richard Stallman, as well as organizations such as the Free Software Foundation, which emphasizes the need to have as much code as possible under strong copyleft in order to prevent it from becoming enclosed into proprietary elements at some later point. This relates to criticism of the choice of license for the free and open source part of Android as well. Participants acknowledge that Google could have chosen a different license than Apache 2, but it was most suitable to be used in combination with all sorts of proprietary parts. This feature was attractive for handset manufacturers, who could then combine AOSP with their own firmware as well.

***Android - its dominance and discontent***

The relationship activists have with Android can be described as both a blessing and a curse. Having the dominant platform open-source, which both opens access to millions of users and makes it possible to "fork" (replicate), is truly cherished by developers. It makes their work possible. Android was a "huge opening" (M5). For individual developers, the accessibility of Android offers the potential to reach out to many users and bring their own code to more devices.

In early 2000, handset manufacturers were developing software for own devices. There were proliferation operating systems, each geared to particular brand of devices. Writing a program that could have been used on all mobile devices was impossible. "*It was a nightmare*" – recalled one of the participants of the pre-Android period:

> *The ability to write one app that you could run on all those devices, that was a dream, that we thought would never gonna happen. So benefit for small developers to build applications and reach huge amounts of portable computing devices, for most of the planet, it's amazing, Android's impact there is so beneficial.* (M3)

The dark side of Android's dominance, on the other hand, is its data surveillance and potentially negative impact on possible alternatives. As one respondent explained:

> *On the one hand it [Android] gave the possibility of freedom to all of us, but the difficulty is that you need to be a technician to achieve this. And on the other side, it may have hindered other developments, which maybe would have been even more free and available without all the data aggregation.* (M2)

***The development behind closed doors***

During the interviews, I probed more in-depth in order to understand what the experience of FLOSS activists was like when working with AOSP code, and to comprehend the limitations that compelled them to work on alternatives to Android. In the following quote, one participant explained what the limitations of Android are from a developers perspective:

> *So technically it is open source, it is technically free software, but there are two things, one is that it is not developed like free software, it is developed like proprietary software, and that means it is very difficult to get your changes in Android. [...] with the Android, almost all of their work process is entirely within Google, and you cannot probably have insight in what they do. The way so the most of free software is used and watched is in each increment of development, publicly. In Android, they just announce a new version every 6 months, a year, and that is when you see all of the updates come out. So the term people use for this is, called throw it over the wall, so they are inside the walls working and every here and there they throw out a big chunk of what they have worked on over for the rest of us to see. I mean this is how Windows is developed; this is*

> *how IOS or MacOS is developed this closed way. The only good point is that we occasionally get to see the source code, which is a very important difference.* (M4)

Participants in this study unanimously saw the way Google developed Android behind closed doors as a problematic and *"frustrating experience."* This observation is vital due to the fact that it directly affects the work of developers. They are confronted with many changes all at once and face uncertainty with respect to the direction the next version of Android will take. In addition to this, the closed development process makes Android only conditionally open because the most important aspect of free and open-source development – the transparency of the process - is missing. Transparency of the process makes software "*entirely auditable*" (i.e., inspectable), which means that anyone can inspect and study the working of the program. The way Google develops Android forecloses the transparency of the process.

***Impossibility to run a phone only on AOSP***

The second reason for discontent among activists was the fact that, over the past eigh years, the free parts of Android (i.e., what was included and released as the AOSP) considerably "shrunk". This means that with each new version, fewer parts were included in the AOSP. This has direct implications for developers because, over time, they had access to fewer and fewer parts of the operating system.

> *The problem is, that the main developer of this AOSP is Google, which is more and more losing interest in this system. So, more and more you see that this AOSP doesn't work anymore as it should because they are neglecting it, and they are pulling more and more functional parts out of it, and developing it as non-free software. So we are losing more and more parts. For free software, it is very unfortunate.* (M1)

When Google discontinues the free and open-source version of a specific functionality, it then typically offers it as a part of a proprietary package (as an app) or as part of the Google cloud service. For example, in order to use push notifications or location, a developer needs to agree to use Google's cloud services. All participants in this study were strongly critical of the prospect of being 'tied to' Google's cloud because this means that their apps are also participating in data aggregation. It is not surprising that most of them work to find alternative solutions and create software that helps users circumvent centralized surveillance by Google or other large companies.

In practice, when Google encloses a functionality, FLOSS activists often respond by initiating a new project. For example, in response to withdrawing push notification from the AOSP, FLOSS activists developed Tutanota, an app offered through F-Droid that provides an alternative to Google's push notifications and, in the long run, aims to provide an open-source alternative for Gmail. What the example of push notification illustrates is a close relationship between changes in the accessibility of Android and responses from the FLOSS community.

Considering that Google has been withdrawing a number of functionalities from the free and open-source part of Android, I asked the participants whether it was possible to run an Android device only with the free and open-source (AOSP) part. They were skeptical of such a possibility. "*Nearly impossible*" (M2), one answered, especially for the end-users.

Severe limitations to running a device only on the free and open-source part of Android mean that marketing claims that the platform is free and open source are only partially correct. It may be accurate enough for handset providers because they benefit from the whole OS, but for the users interested in FLOSS, as well as FLOSS activists, this is a misleading claim.

### *A loophole to freedom*

Despite the closed development process and the shrinking of the AOSP, Android is still recognized as "fundamentally different" from other proprietary mobile platforms because of the possibility to install an app from a location other than Google Play. This means that a user can install software which does not come directly from the Google Play Store. This is not possible on Windows Phone or iOS.

From the perspective of FLOSS developers, although Google allows the installation of alternative app stores, it also employs various strategies to make this process complicated and channel developers to still opt for the Google Play Store.

> *On Android, theoretically you can install your own applications, control is users. In practice however, everything is made so that you use Google Play store. So in practice it is not as free as that, but still an important point is that you can still install your own applications. (*M2)

However, as one participant in this study explains, one of the most successful and important FLOSS initiatives for mobiles, the repository F-Droid, fills precisely this gap. By being installed as an alternative to Google Play, developers can reach smartphone users through F-Droid.

Having the option to install an alternative distribution channel on Android run phones offers a loophole through which FLOSS apps are delivered to end users. This is already a positive development in the eyes of FLOSS activists because it enables them to offer apps that circumvent surveillance or allow encrypted communication to Android users through F-Droid.

### *Libre software for smartphones*

#### *F-Droid and Custom Roms*

As discussed in the previous section, participants in this study considered F-Droid to be one of the most important developments for bringing FLOSS apps to a wide range of users. The project is non-profit and fully volunteer-run. It was initiated by Ciaran Gultnieks in 2010. F-Droid gradually grew into a hub for FLOSS apps[56].

[56] When visiting the wiki of the repository on July 19, 2015, I recorded that that there were 1, 576 apps on F-Droid. When inspecting the number on August 1, 2020, F-

The main premise of F-Droid is twofold. On the one hand, it is a repository of free apps (apps with open source code, without advertisements and trackers), and individual developers can submit their FLOSS apps to be included. Each app is then inspected by F-Droid volunteers to make sure it is FLOSS software and does not contain *antifeatures*. Containing antifeatures means that an app itself is FLOSS, but it also depends on non-free software. For example, the app will not be useful if it depends on Google Maps (which is a proprietary app). Another antifeature is tracking users without their knowledge, e.g., sending reports in the background without the permission or knowledge of the user.

On the other hand, F-Droid also functions like an app store itself, which has the form of an app and can be downloaded on any Android-run smartphone. Once downloaded, a user can search for FLOSS apps through F-Droid. Installing F-Droid does not require rooting.[57] It can be used together with official app stores. Furthermore, F-Droid itself can be copied (because it is free and open-source software), and other groups can potentially make their own app stores. This possibility was used by developers in Cuba who, as one of the participants, has witnessed, created an independent repository of apps on a local wi-fi network. In the absence of infrastructure, F-Droid was adapted to local needs and had some two thousand apps available.

Droid included a total 2,951 apps. This number reflected inspected apps, meaning F-Droid volunteers have inspected the app.

[57] Rooting is a term comparable to 'jailbreaking' and means gaining access to the complete operating system of a smartphone. Rotting a phone allows one to override the restrictions placed on the phone, e.g. by changing firmware placed by handset manufacturers, or deleting so called 'bloatware' – software which comes installed on the phone and that it is not possible to uninstall. Rooting requires technical knowledge. Even though there are detailed instructions available on the dedicated Wikis, it is highly unlikely that a user without any prior knowledge in programming would attempt it. The benefits of rooting as explained by participants of the study were longer battery life, no dependency on surveillance, fewer background location checks, more storage, and simply full control over the device.

It should be noted that, outside of F-Droid, it is not possible to search specifically for FLOSS apps in any other app store. Thus, without knowing the name of a specific app in advance, a user, if she wants to discover FLOSS software for mobiles, cannot search through conventional app stores. Therefore F-Droid makes disperse FLOSS apps accessible in one repository.

Although F-Droid has been instrumental in bringing a variety of FLOSS apps to users, installing F-Droid alone is not sufficient to run a smartphone on fully free software. As participants explained, this is because any Android-run phone comes with a preinstalled operating system, firmware (a small piece of software installed by a handset manufacturer to enable the proper functioning of hardware), various drivers, and preinstalled apps such as Gmail, Google Play, etc. Among this set of elements, only part of the Android OS is free and open source. The rest - drivers, firmware, as well as all Google apps - are proprietary.

Given this context, what the participants saw as the *real* development for having free software for mobile devices is the proliferation of Android OS modifications, called Android custom ROMs.

> *Custom ROMs, those are forks from Android that use the core body of the Android system, that is the free software, and forget the rest. So if you want to have free software on your mobile, currently the only way to achieve this is, a custom ROM and F-Droid. Both together can help you get free software running on a mobile phone.* (M5)

Installing a *custom ROMs* requires rooting a phone. Rooting is a term comparable to ‘jailbreaking’ that means gaining access to the complete operating system of a smartphone. Rooting a phone allows one to override the restrictions placed on the phone, e.g. by changing firmware or deleting so called ‘bloatware’ – software which comes preinstalled on the phone and that it is not possible to uninstall.

Rooting requires technical knowledge. Even though there are detailed instructions available on the dedicated Wikis, it is highly unlikely that a user without any prior knowledge in programming would attempt it. The benefits of rooting as explained by participants of the study were longer battery life, no dependency on surveillance, fewer background location checks, more storage, and simply full control over the device. In many jurisdictions, rooting invalidates the warranty on the device. A solution often proposed is to reuse second-hand smartphones for these devices. Among free software activists, there are often campaigns to donate old phones for rooting purposes.

As a result of these developments, participants pointed out, a vibrant community emerged around Android custom ROMs where new modifications were regularly initiated by individual developers and documentation about the process was shared. An important exchange platform is the XTA forum for developers, where information about new custom ROMs and experiences in making and using them are actively shared.

Some examples of such modified versions of Android include CyanogenMod and its successor fork, Lineage OS; Firefox OS; and Puur OS. CyanogenMod was one of the first custom ROMs, which attracted many independent developers. Even though the project disintegrated, some participants of this study were still using it to have “free smartphones.” Custom ROMs become alternatives to Google’s official Android. They enable a community of FLOSS activists to make new modifications to the OS and use a smartphone without surveillance and without relying on centralized servers. As one of the participants joked, if Google does indeed take a course that is not acceptable to developers, then it is always possible to fork Android.

## Conclusion

Due to the hegemonization of open source development by Google, FLOSS software for mobiles is less visible not only in scholarly inquiries but also to average end-users. There is considerable criticism directed at Google’s selective open-source practices and the consequences of the *perceived* Apple/Google duopoly in the mobile domain. On the one hand, Google’s adoption of open source is theorized to explain the conglomerate's rapid success in gaining dominance in the mobile industry (in which it was a newcomer). On the other hand, Google’s Android is often perceived to be free, open-source, and geek-friendly. In light of this, I aimed to direct attention in this study to a lesser-known phenomenon by exploring mobile FLOSS. Through in-depth interviews with activists who have developed and advocated FLOSS specifically for smartphones, this study aimed to gain deeper insight into the state of FLOSS code in the context of mobiles. I will now proceed by first responding to the three guiding research questions and, following this, will provide concluding remarks about the limitations and implications of this study.

The first research question inquired into *how FLOSS activists construct the meaning of free software in the context of mobiles*. The legacy of Richard Stallman and his definition of free software was reflected in virtually all of the respondents' replies. The four freedoms he outlined seemed naturalized to such a degree that they served as cornerstones, or base pillars, from which some deviated while others argued to maintain its ‘purity’. The meaning of FLOSS was primarily constructed through the understanding that free software allows an individual to be in control of her device and data. It was experienced as personally meaningful work for participants, and they understood to also be important for other individuals and society at large. It gave them a sense of working towards something important for others, “a sort of a higher goal,” as one of the participants put it. In spite of this, the sentiment that many end-users “do not care” about privacy or controlling their own devices was also widely shared. It appeared as both a source of frustration and inspiration for advocating FLOSS. In this context, activists saw the pervasiveness of cloud computing and remote control of smartphones as problematic but also as presenting a possibility to increase awareness for the need for free software.

The second research question was aimed at understanding *how FLOSS activists perceived the role of Android for the movement.* Although the dominant platform for mobile devices is Android, which is released under an open-source license and commonly perceived to be free software, participants of this study disagreed with this perspective. Their discontent was primarily concerned with the way Google developed and maintained Android. Despite being marketed as a free and open-source platform, the development process was completely lacking transparency, and Google (as the main developer of the OS) was releasing the source code of every successive version all at once.

Developers found this process frustrating, like “dumping down” large amounts of code without offering any insight into what processes took place during the six months in which it was being developed behind closed doors. The major problem with this approach to developing Android was the loss of transparency and enclosure of the decision-making process in the hands of the main developer (Google). The process of transparent development and the possibility to audit code at any given moment are the two main premises for ensuring the quality of FLOSS code.

In addition to that, another reason for concern was Google’s practice of withholding a growing number of functionalities from the free and open-source part of the Android platform. Participants in the study stressed that, over the past eight years, more and more features were moved from the AOSP into the proprietary layer of Android. This meant that for developers to have access to such functionalities (e.g., location, notification), they had to participate in Google’s cloud services. This was contested and deemed unacceptable due to making their apps both subject to, as well as participants in, Google’s data surveillance.

Last but not least, the study aimed to identify *which initiatives FLOSS activists consider to be the most important in bringing free software to mobiles and why.* One possibility for bringing free software to mobiles was presented by the non-exclusivity of Android. As it allowed the installation of apps from an alternative source, free software activists exploited this opportunity to bring F-Droid to users. F-Droid is a repository of FLOSS apps. It is unique because it allows searches for free apps (meaning apps that are released with source code and also do not track users, or depend on centralized corporate servers), which no other app store allows. Apps in F-Droid are checked by volunteers to determine whether they contain any trackers, to which servers they are connected (by sending user data), and if they depend on any proprietary systems. F-Droid is used by FLOSS activists and supporters as an alternative to Google Play. This has implications on many levels. Primarily, F-Droid is an alternative to well-curated app stores that are embedded in platforms and instrumental in the *datafication* process (Poell et al., 2019). Given Google’s business model, which does not depend on selling software, but rather on gathering and selling user data (Fuchs, 2011, Kang & McAllister, 2011), the implications of *F-Droid* for end-users are potentially far-reaching.

Beyond free apps, making forks of the AOSP, and creating new custom modifications of the operating system (such as CyanogenMod, Lineage OS, and Replicant), the participants also viewed custom ROMs, a new vibrant area with great potential, as decisive for FLOSS in the mobile context. Albeit more contested, this area is equally important for scholars to explore. Participants in this study saw custom ROMs as a "real innovation led by users." Forking is not only intrinsic to free and open source development but also represents a particular value inscribed in the design of FLOSS licenses. Forking is essential because it allows members of a community to benefit from shared code, even in cases when a project is going in a direction that some developers may not agree with (e.g., something that happened with CyanogenMod when the lead developer decided to incorporate it and take a more commercial path). In such situations, they can fork it, and the project will develop in two branches. This observation suggests that when code is free and open-source, it cannot be easily enclosed. Nonetheless, broader economic interests are at play in constraining user modifications to the Android OS (primarily by voiding warranties on the handsets), to which activists respond by rooting second-hand devices.

Reflecting on the limitations and implication of the study's findings vis-à-vis other literature, I would like to address a few points. First, the participants of the study were all from Europe and the United States, which can be seen as one limitation of the study to substantiate arguments beyond these geographic areas.

When I juxtapose the insights gained from the participants of this study with the insights from other scholars who have explored processes of software modification for smartphones in the Global South, it becomes evident that the reasons for modifying software for smartphones may be very different. For example, jailbreaking iPhones in Vietnam (Nguyen, 2016) not as a way to resist proprietary or closed architecture, but as a way to gain access to a fashionable device amidst the absence of formal infrastructure. Likewise, practices of repairing and unlocking second-hand phones in grey markets

(Chandra, Ahmed, & Pal, 2017) cannot be articulated through the same rationale as FLOSS in the context of the Global North, but rather requires cultural sensitivity to interpreting localized practices (Arora, 2019b).

Do such considerations devalue the importance of FLOSS software projects such as F-Droid, Lineage OS, or Replicant produced mainly by a homogeneous social group in the Global North? Not necessarily. More likely, the absence of diversity points to the impossibility of decoupling the shaping of technological design from the social norms and inequalities prevalent in society. The case of gender is an important example, demonstrating that the call for generative proposals from prominent feminist scholars is applicable not only to specific areas but rather to all studies of technological designs (Shaw, 2014), even those which seem to be 'genderless' and neutral.

If, in the Global North, unlocking an iPhone represents a critique of closed architecture and unrooting an Android smartphone is a form of protest to circumvent Google's datafication process, the same practices might carry different meanings in other socio-economic contexts. This observation does not mean there is no interconnectedness, however. On the contrary, as the example of using F-Droid in Cuba shows, technologies 'travel' and are creatively appropriated to respond to local needs. In my view, such translation illustrates the strength of FLOSS's characteristics: structural openness of code and reciprocity, beyond its discursive justification.

Last but not least, the same acts that were considered grounds for user innovation some 30 years ago are now restricted and rearticulated as criminal activity. The shift in the image from hacker as a hero to a hacker as a criminal has been normalized, yet is only a part of the picture. Today, there is an increased tendency to curate 'hacking' to enclose and manage potential *disruptions* within a walled garden.

# Conclusion

Through qualitative inquiry into the practices, discourses, and perceptions of independent software developers and digital rights activists, I aimed to explore and understand how the programmability of smartphones was shaped. Employing the social construction of technology (SCOT) and critical theory of technology (CTT) as analytical tools, I investigated the relationships between relevant social groups and emerging structural possibilities and constraints in accessing and participating in the programmability of smartphones. I proposed that due to the role of code/software in structuring affordances of devices, as well as its inherent qualities of being prescriptive and performative, opening up the programmability of smartphones to expert users' participation could contribute to widening the range of actors involved in the negotiation process of this technological artifact. In turn, I argued that this, in turn, could be potentially beneficial for the democratization of design and development of these pervasive technologies, allowing more diverse groups to gain access and engage in the initial phase, creating potential new functions and affordances for these pervasive devices.

The guiding research question contained both exploratory and explanatory parts, seeking first to identify the conditions of possibility and limitations for writing software for smartphones and, second to evaluate the process in light of democratic rationalization. Democratic rationalization (Feenberg, 2002, 2007) reflects a normative call to subject the process of the design and development of new technologies to democratic principles, processes, and values.

Methodologically, I carried out three empirical studies using qualitative data gathering and analysis techniques, including discourse analysis and thematic analysis. Through this inquiry, I aimed to understand the conditions of writing software (apps and OS) from the perspective of individual, independent developers. The empirical studies were interconnected, with each successive study building on the most important findings of the previous one. In this way, the first study (Chapter 4) explored a diverse group of independent developers and identified the differences between the existing platforms from the perspective of developers. The findings showed how a newly emergent group of mobile operating system providers exercised control in shaping the conditions of programmability through app stores. The most accessible of these platforms for developers was Google's Android because it was also released as open source. This meant that developers could not only write an app for this operating system but tinker with the operating system itself as well. The easy accessibility of Android was accompanied by increased criticism of Google's selective adoption of open-source practices. This led me to delve deeper into understanding Android from the perspective of developers and to investigate how open and free Android really was. A comparative analysis (Chapter 5) of Google's official communications regarding Android versus the "Free Your Android!" campaign by European digital rights activists showed a serious discrepancy between the public marketing of Android as the "first free and truly open mobile platform" and the real practices through which Android was developed and constituted. The last study (Chapter 6) focused on a small but active community of FLOSS (Free/Libre and Open Source Software) activists in Europe and North America who were challenging Android's dominance by creating alternative distributions, and promoting FLOSS apps for mobiles. By exploring their views and perceptions, as well as their efforts to create libre software, I documented why and how digital rights activists were contesting the dominance of Android.

Drawing on the findings of these empirical studies, I argue that three particular developments – the commodification of apps, the conditionality of open source, and the absence of collaboration – constrain the democratizing potential of the programmability of smartphones. Rather than facilitating collaborative, user-led innovation, the programmability of smartphones has been configured and shaped into a highly-structured digital marketplace, with non-interoperable mobile platforms. The structural limitations placed on the programmability of mobile devices are more restrictive than they were in the case of PCs. This was especially so in the earliest period in which smartphones became available. Most decisively, participating in programmability is directly connected with surveillance, datafication, and enforcement of intellectual property rights for digital/immaterial goods and resources.

To borrow the language of SCOT, the programmability of smartphones has reached a temporary closure, meaning a specific understanding has stabilized, and a consensus has emerged. Thus, software for mobiles is primarily understood as a single app, written for a specific operating system, and distributed through a specific channel (i.e., an app store). *App - Commodity – Store - User* is a new model. I found the same model being replicated across all competing mobile platforms. Even though there were differences, such as the degree of control exercised by OS providers and the accessibility and costs involved in distributing an app through an app store, the model of an *app as a single commodity mediated through app stores* has become standardized.

The role of app stores emerged as decisive for independent developers. On the one hand, developers gain access to thousands, if not millions, of users through centralized app stores. Participants referred to this as a "huge opening" that was "unimaginable" during the cell phone era. On the other hand, participation is highly structured and integrated into the broader datafication processes of mobile platforms. Moreover, through acting as mediators, app stores have emerged as gatekeepers. By reviewing, ordering, curating and distributing software to users, app stores act as intermediaries with increased power to influence not only the potential functionalities/affordances of smartphones, but also content. The process is carried out as a technical procedure to make sure that an app is compatible with an operating system. Developers in this study expressed uncertainty about the exact justification for rejections and shared that the decisions are communicated under the terms of non-disclosure agreements, making it difficult to publicly discuss banned apps. I consider the way app stores are configured as an example of formal bias in the shaping of programmability. In this context, formal bias means that particular values that benefit powerful actor(s) more than others are sedimented in the configuration of technological devices.

The review process pushes the implications of an app store beyond the mere marketplace. In the case of smartphones, app stores act as regulatory gatekeepers. This is very different from the role filled by hosting servers for websites or ISP's for network access. In the current configuration of how the programmability of mobile devices is made accessible, the app stores are not neutral channels, but active mediators in the selection of affordances and potential uses of these devices. In carrying out this role, they resemble the functioning of a media organization more than that of a technological organization. This leads me to propose that future research should reconceptualize and investigate the ordering of the power of technologies in a manner similar to the role of news media for politics or the formation of public opinion.

All major stores review apps for content, except Google's Android. Google has only blocked apps in special cases, and it is commonly accepted that content on Google Play is not reviewed. More importantly, unlike Microsoft's Windows Phone or Apple's iOS, Android allows the installation of an alternative app store. This means that on an iPhone, a user cannot (legally) access any programs distributed outside the official Apple OS environment. Doing so amounts to hacking the iPhone. On Android run smartphones, it is technically possible to install software outside of Google Play. This difference is important for developers and especially for digital rights activists because it serves as a sort of "loop to freedom" enabling access to Android's wide userbase.

Furthermore, Google, unlike other major OS providers, makes Android's source code publicly available (releasing it under a permissive, yet open-source license: Apache 2). Characterizing Android as open source requires some qualification because what is publicly shared is not the whole Android OS but rather the Android Open Source Project (AOSP), which excludes many of the smartphone's core functionalities (such as location, notifications). While criticism of Android's selective adoption of open-source practices has been documented in previous research (Spreeuwenberg & Poell, 2012), my inquiry into the perspectives of developers further demonstrated that using only the AOSP part of Android is insufficient for creating an app. Participants also stressed that Google employs various techniques to co-opt developers into using its own proprietary parts and cloud infrastructure, thereby including their creations (apps) in the centralized datafication structure. The limitations of the AOSP are exacerbated by the way Google manages the open part. Over the past 10 years, the company has been removing functionalities from the AOSP and adding them into the closed, proprietary part of Android. This presents a serious limitation because, even though Google markets Android as an open and free mobile platform, it is only partially open source, and the accessible part keeps "shrinking."

Another shortcoming of the way Google manages Android is the foreclosure of accessibility to directly participate in developing the operating system. This means that Android's development process takes place behind closed doors and that the source code is only shared after work on a new version has been completed. As participants consistently explained, the way Android is written exclusively within the boundaries established by Google is more similar to a proprietary model of software development than to an open-source model. This means that, given the process of its development, Android resembles Window's Microsoft and Apple's iOS much more than is commonly perceived. This differs significantly from how Linux (a popular FLOSS operating system for PCs) was developed and shows how free/libre and open-source development has been rearticulated in the context of mobile devices.

Considering the pervasive dominance of Android, I argue that, in the case of smartphones, there is hegemonization of open source by Google. By choosing the metaphor of hegemonization, I aim to stress that even though Google's Android is marketed as a free and open-source platform, it rules out the potential for collaborative development and forecloses the possibility of having Android as a shared product. The company does indeed give the program to handset providers without cost and occasionally releases the source code to the public, but its process of labor organization still follows the same model as Microsoft or Apple.

In the case of free and open-source development, many scholars have identified deliberative discussions and interactions among contributors  as the most important and distinct factors contributing to the democratic potential of FLOSS (Kelty, 2008, 2013). In contrast , however, discussions and interactions regarding Android's development remain closed and code is only occasionally "thrown over the wall," as one participant put it. Curatorship of

Android by Google still resembles a walled garden, with the exception that visitors are occasionally invited in to see the source code.

The hegemony of Android as an open-source platform for mobile devices reflects a broader shift in which we see the expansion of commodification into a new domain, namely user interactions and data. Apple and Google both use the same model of app-commodity-store but differ in terms of how they commodify the app stores and what their source of profit is. Apple keeps the source code of its OS protected under copyright law. Any tinkering equals jailbreaking. Thus, the only means to participate in the Apple platform is to use its exclusive distribution channel, to agree to be reviewed for content, and to pay Apple 30% of the sale price for each app sold. Apple creates (artificial) foreclosure by enclosing iOS software in a proprietary manner through copyright. The broader structural mechanisms that allow Apple to capture the most profit from a single iPhone (as illustrated in the review of its global value chain in Chapter 3) are cheap manufacturing and ownership of patents and copyright on software. Google, on the other hand, does not rely on making its immaterial products (software) scarce. On the contrary, it allows its usage, reusage, modification, and even forking (i.e. copying), but only under strictly established boundaries. There is a pragmatic explanation for Google's strategy related to the expansion of commodification from software to user data. Google is not concerned with keeping its Android operating system scarce because its very usage constitutes its main assets: the "surveillance assets" (Zuboff, 2015). More traffic equals more data, which is the source of profit for an advertising company. Thus, as participants explained, even though Google releases Android as open source, it is "almost impossible" to use it without participating in Google's centralized services. Android's *conditional openness* is a double-edged sword. On the one hand, it hegemonizes the understanding of open source. On the other hand, it serves as a loop to freedom. This leads us to the last and most significant finding of this inquiry: namely, how the current dominance of app stores is contested locally by digital rights activists as well as entrepreneurs who exploit the problem of surveillance to create new products.

The increase in surveillance has prompted digital rights activists to expand their discourse on the criticism of surveillance. This means that, in the context of mobile devices, free software activists not only contest black-boxed technologies (such as proprietary software, DRM), but also increasingly direct their efforts at bypassing data surveillance and dependence on the centralized (cloud) servers of large corporations. Libre/free software programs for mobiles are primarily presented to the wider public as a means to bypass datafication and surveillance.

For digital rights activists and free software proponents, FLOSS for mobile devices is increasingly understood as software that not only has open source code, but that is also shielded from participating in the datafication process. My inquiry looked specifically into the perceptions of European and North American digital rights activists. Therefore, this shift in discourse should be understood in its  local socio-political context. For these activists, relying on notions such as individual property rights is culturally "native" and represents a discursively accessible repertoire. In the Global South, however, writing FLOSS software or practices of jailbreaking and unlocking may not be accompanied by the same liberal claims on access to information or individual rights of developers to tinker with technology. My criticism about the limits of this discourse does not imply that these practices do not matter, but merely calls for sensitivity in understanding the limitations of a "particular normative" view of privacy in the Global North, which is largely uncontested (Arora, 2019a).

To restate, free software in Europe and the United States serves primarily as a critique of the structural embeddedness of programmability in the surveillance economy. This represents a marked shift from 1980s, when the accessibility to source code and criticism of the copyright system formed the backbone of the FLOSS movement. In my understanding, this shift of emphasis and expansion of discourse is connected to the fundamental change in configuring practices and pathways of writing and distributing software within the smartphone industry. The very design code through which programmability is encoded in these devices is mediated through commodification and embedding it in the datafication process. The structure seems universal as it is utilized by all competing platforms. Digital rights activists have responded to this change by shifting FLOSS discourse from the developers' four freedoms to questions of data ownership.

## Conclusion

While applying the concept of democratic rationalization (Feenberg, 2017) and looking at the conditions and practices under which the programmability of smartphones is being configured, it is difficult to give a straightforward evaluation of its consequences. Yes, the number of apps has skyrocketed and there are more possibilities to write software for smartphones, but the very tools and structures that are needed to create and share these apps make individuals dependent on conditional surveillance. Despite the omnipresence of centralized data servers and, at times, the perceived impossibility of escaping them, concrete projects and marginal communities that challenge and resist the global commodification of information do exist. Without romanticizing the work of small groups of FLOSS activists, these instances of disobedience, hacking and creative appropriation still matter because they produce needed tools which are later used and translated globally. Initiatives such as making custom ROMs (OS modifications) for Android, writing libre/free alternatives of firmware and drives, and maintaining F-Droid - a repository of FLOSS apps for smartphones – represent marginal but important efforts through which dominant forms of programmability (that are tied to datafication and proprietary systems) are locally contested. These volunteer-run projects demonstrate that there are limitations to the total closure of the interpretative flexibility of smartphones and show that there are different interpretations of how the programmability and functionality of this technological artifact are understood.

As Feenberg maintains, social groups challenge dominant design codes, subverting them through bottom-up initiatives. Through this inquiry into the work of FLOSS activists in Europe and North America, I found examples of creative resistance through hacking and modification of the Android OS and by making custom modifications to it which bypass Google's datafication process. FLOSS apps are still not easily accessible for end users. Despite numerous attempts to include "average' users" (e.g., by organizing Free Your Android meetings in various European cities), opting out of the dominant smartphone milieu and running a phone solely on free and open-source software is not simple. It requires commitment and access to the community of FLOSS activists. In this context, it is important to highlight that there is also a large degree of homogeneity (in terms of gender, social class, and educational background) in the FLOSS community in Europe. Privacy versus surveillance (enabled through datafication) have emerged as design codes for contemporary (mobile) media. These design codes reflect the power interests of involved actors who benefit from the commodification of audience data and interactions (Fuchs, 2011b). The intellectual debate about the nature of this relationship is still open and ongoing. However, platformisation (Poell et al., 2019) as a broader process/trend is becoming normalized. Feenberg's critique is particularly important here as we witness how datafication acts as a formal bias in the design of contemporary information systems that benefit powerful actors. The findings of the thesis resonate with broader social developments, namely the increased reliance on surveillance and integration of datafication into the fabric of our daily lives. These trends are reflected in the emergence of new fields of inquiry such as privacy studies and surveillance studies.

Ultimately, what I came to see clearly after working on this thesis is that, despite the fact that centralized surveillance structures appear overwhelming due to their considerable power, they are still contingent structures that are constantly challenged through local practices and discourses. In my view, employing a global framework that remains sensitive to local discourses and practices is the most fruitful way to approach the question of how digital surveillance infrastructure is resisted. By global framework, I mean the importance of understanding the role of intellectual property rights, labor (especially those who are marginalized), and consequences for the environment. By localized knowledge and movements, I mean the discourses and practices of embedded communities which resist hegemonic understandings and configurations. In the end, this means that the design and development of smartphones cannot be discussed or examined without juxtaposing it with value creation (profit extraction) in the digital economy.

So, *if Google is the new Microsoft*, *then where is Linux?* This rhetorical question captures the most decisive shift in the "universe of possibilities" within which the programmability of smartphones has been configured and is different than in the context of PCs. This is a shift through which open source development curated under corporate control has been normalized. Mobile platforms, whether propriety or open source, are equally involved in the commodification of apps and the datafication process. It is within this context that we have witnessed an expansion of struggle and resistance, from the access to code to access and ownership of data. This change and relocation of centers of struggles is made possible through the naturalization of surveillance.

# Bibliography

ACIDH. (2011). *Unheard Voices, Mining Activities in the Katanga Province and the Impact on Local Communities*. Retrieved from https://www.somo.nl/wp-content/uploads/2011/12/Unheard-Voices.pdf

Agger, B. (1991). Critical Theory, Poststructuralism, Postmodernism: Their Sociological Relevance. *Annual Review of Sociology*, *17*(1), 105–131. doi: 10.1146/annurev.soc.17.1.105

Ahern, K. J. (1999). Ten tips for reflexive bracketing. *Qualitative Health Research*, *9*(3), 407–411. doi: 10.1177/104973239900900309

Ahmed, H., Qureshi, O. M., & Khan, A. A. (2015). Reviving a ghost in the history of technology: The social construction of the recumbent bicycle. *Social Studies of Science*, *45*(1), 130–136. doi: 10.1177/0306312714560640

Akrich, M., & Latour, B. (1992). A summary of a convenient vocabulary for the semiotics of human and nonhuman assemblies. In J. Law & W. E. Bijker (Eds.), *Shaping Technology / Building Society: Studies in Sociotechnical Change* (pp. 259–264). Cambridge: The MIT Press.

Aksulu, A., & Wade, M. (2010). A comprehensive review and synthesis of open source research. *Journal of the Association for Information Systems*, *11*(11), 576–656. doi: 10.17705/1jais.00245

Ali-Yrkkö, J., Rouvinen, P., Seppälä, T., & Ylä-Anttila, P. (2011). Who captures value in global supply chains? Case Nokia N95 smartphone. *Journal of Industry, Competition and Trade*, *11*(3), 263–278. doi: 10.1007/s10842-011-0107-4

Amadeo, R. (2013). Google's iron grip on Android: Controlling open source by any means necessary. Retrieved October 21, 2013, from Ars

Technica website: http://arstechnica.com/gadgets/2013/10/googles-iron-grip-on-android-controlling-open-source-by-any-means-necessary/

Amnesty International. (2016). *"This is what We Die For": Human Rights Abuses in the Democratic Republic of the Congo Power the Global Trade in Cobalt*. Retrieved from https://www.amnesty.org/download/Documents/AFR6231832016ENGLISH.PDF

Arora, P. (2019a). General Data Protection Regulation—A global standard? Privacy futures, digital activism, and surveillance cultures in the Global South. *Surveillance & Society*, *17*(5), 717–725. doi: 10.24908/ss.v17i5.13307

Arora, P. (2019b). *The next billion users: Digital life beyond the West*. Cambridge: Harvard University Press.

Bardzell, S. (2010). Feminist HCI. *CHI '10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1301–1910. New York, New York, USA: ACM Press. doi: 10.1145/1753326.1753521

Bardzell, S., & Bardzell, J. (2016). Feminist design in computing. In *The Wiley Blackwell Encyclopedia of Gender and Sexuality Studies* (pp. 1–7). Singapore: John Wiley & Sons, Ltd. doi: 10.1002/9781118663219.wbegss470

Barratt, N., & Shade, L. R. (2007). Net Neutrality: Telecom policy and the public interest. *Canadian Journal of Communication*, *32*(2). doi: 10.22230/cjc.2007v32n2a1921

Bauwens, M. (2005). The political economy of oeer production. *CTHEORY*, (1000 Days of Theory). Retrieved from https://journals.uvic.ca/index.php/ctheory/article/view/14464/5306

Bauwens, M., & Kostakis, V. (2014). From the Communism of Capital to Capital for the Commons: Towards an Open Co-operativism. *TripleC: Communication, Capitalism & Critique. Open Access Journal for a Global Sustainable Information Society*, *12*(1), 356–361. doi: 10.31269/triplec.v12i1.561

Bauwens, M., Kostakis, V., & Pazaitis, A. (2019). *Peer to peer: The commons manifesto*. London: University of Westminster Press. Retrieved from https://www.jstor.org/stable/j.ctvfc53qf

Bekkers, R., Verspagen, B., & Smits, J. (2002). Intellectual property rights and standardization: the case of GSM. *Telecommunications Policy*, *26*(3–4), 171–188. doi: 10.1016/S0308-5961(02)00007-1

Bekkers, R., & West, J. (2009). The limits to IPR standardization policies as evidenced by strategic patenting in UMTS. *Telecommunications Policy*, *33*(1–2), 80–97. doi: 10.1016/j.telpol.2008.11.003

Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. London: Yale University Press.

Benkler, Y., & Nissenbaum, H. (2006). Commons-based Peer Production and Virtue. *Journal of Political Philosophy*, *14*(4), 394–419. doi: 10.1111/j.1467-9760.2006.00235.x

Berry, D. M. (2004). The contestation of code. *Critical Discourse Studies*, *1*(1), 65–89. doi: 10.1080/17405900410001674524

Berry, D. M. (2008). *Copy, rip, burn: The politics of copyleft and open source*. London: Pluto Press.

Berry, D. M. (2016). *The philosophy of software: Code and mediation in the digital age*. London: Springer.

Berry, D. M., & Moss, G. (2006). Free and open-source software: Opening and democratising e-government's black box. *Information Polity*, *11*(1), 21–34. doi: 10.3233/IP-2006-0084

Bessen, J., & Maskin, E. (2009). Sequential innovation, patents, and imitation. *The RAND Journal of Economics*, *40*(4), 611–635. doi: 10.1111/j.1756-2171.2009.00081.x

Bessen, J., & Meurer, M. J. (2008). *Patent failure: How judges, bureaucrats, and lawyers put innovators at risk*. Princeton: Princeton University Press.

Bijker, W. E. (1993). Do Not Despair: There Is Life after Constructivism. *Science, Technology & Human Values*, *18*(1), 113–138. doi: 10.1177/016224399301800107

Bijker, W. E. (1997). *Of bicycles, bakelites, and bulbs: Toward a theory of sociotechnical change*. Cambridge: The MIT Press.

Bijker, W. E. (2001). Understanding technological culture through a constructivist view of science, technology, and society. In S. H. Cutcliffe & C. Mitcham (Eds.), *Visions of STS: Counterpoints in Science, Technology, and Society Studies* (pp. 19–34). Albany: State University of New York Press.

Bijker, W. E., & Law, J. (1992). *Shaping technology/building society: Studies in sociotechnical change*. Cambridge: The MIT Press.

Blaikie, N. (2000). *Designing social research*. Cambridge: Polity Press.

Blaikie, N., & Priest, J. (2019). *Designing social research: The logic of anticipation*. Cambridge: John Wiley & Sons.

Boyatzis, R. E. (1998). *Transforming qualitative information: Thematic analysis and code development*. London: Sage Publications.

Boyd, P., & Schweber, L. (2018). Unintended consequences: institutional artefacts, closure mechanisms and the performance gap. *Building Research & Information*, *46*(1), 10–22. doi: 10.1080/09613218.2017.1331096

Boyle, J. (2003). The Second Enclosure Movement and the Construction of the Public Domain. *SSRN Electronic Journal*, *66*(1/2), 33–74. doi: 10.2139/ssrn.470983

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. doi: 10.1191/1478088706qp063oa

Braun, V., & Clarke, V. (2019). Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, *11*(4), 589–597. doi: 10.1080/2159676X.2019.1628806

Brophy, E., & De Peuter, G. (2014). Labors of mobility: Communicative capitalism and the smartphone cybertariat. In A. Herman, J. Hadlaw, & T. Swiss (Eds.), *Theories of the mobile internet: Materialities and imaginaries* (pp. 72–96). New York: Routledge.

Bruns, A. (2007). Produsage. *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition - C&C '07*, 99–106. doi: 10.1145/1254960.1254975

Budhathoki, N. R., Bruce, B., & Nedovic-Budic, Z. (2008). Reconceptualizing the role of the user of spatial data infrastructure. *GeoJournal*, *72*(3–4), 149–160. doi: 10.1007/s10708-008-9189-x

Burgess, J. (2012). The iPhone moment, the Apple brand and the creative consumer: From'hackability and usability' to cultural generativity. In L. Hjorth, J. Burgess, & I. Richardson (Eds.), *Studying mobile media: Cultural technologies, mobile communication, and the iPhone* (pp. 28–

42). New York: Routledge.

Calderaro, A. (2011). New political struggles in the Network Society: The case of Free and Open Source Software (FOSS) movement. *SSRN Electronic Journal*, 1–25. doi: 10.2139/ssrn.2954445

Callon, M. (1990). Techno-economic networks and irreversibility. *The Sociological Review*, *38*(1), 132–161. doi: 10.1111/j.1467-954X.1990.tb03351.x

Campbell, H. A., & La Pastina, A. C. (2010). How the iPhone became divine: New media, religion and the intertextual circulation of meaning. *New Media & Society*, *12*(7), 1191–1207. doi: 10.1177/1461444810362204

Carberry, E. J., Bharati, P., Levy, D. L., & Chaudhury, A. (2019). Social movements as catalysts for Corporate Social Innovation: Environmental activism and the adoption of green information systems. *Business & Society*, *58*(5), 1083–1127. doi: 10.1177/0007650317701674

Carmody, T. (2011). There is no such thing as Android, only Android-compatible. Retrieved September 11, 2015, from WIRED website: http://www.wired.com/2011/12/only-android-compatible/

Carpentier, N., & De Cleen, B. (2007). Bringing Discourse Theory into Media Studies: The applicability of Discourse Theoretical Analysis (DTA) for the Study of media practises and discourses. *Journal of Language and Politics*, *6*(2), 265–293. doi: 10.1075/jlp.6.2.08car

Carpentier, N., & Spinoy, E. (Eds.). (2008). *Discourse theory and cultural analysis: Media, arts and literature*. London: Hampton Press.

Chandra, P., Ahmed, S. I., & Pal, J. (2017). Market Practices and the Bazaar. *Proceedings of the 2017 CHI Conference on Human Factors in*

*Computing Systems*, 4741–4752. New York: ACM. doi: 10.1145/3025453.3025970

Chen, S. (2016). The Materialist Circuits and the Quest for Environmental Justice in ICT's Global Expansion. *TripleC: Communication, Capitalism & Critique*, *14*(1), 121–131. doi: 10.31269/triplec.v14i1.695

Chopra, S., & Dexter, S. D. (2008). *Decoding Liberation*. New York: Routledge. doi: 10.4324/9780203942147

Christians, C. G. (2016). Social justice and Internet technology. *New Media & Society*, *18*(11), 2760–2773. doi: 10.1177/1461444815604130

Chun, W. H. K. (2006). *Control and Freedom: Power and paranoia in the age of fiber optics*. Cambridge: The MIT Press.

Chun, W. H. K. (2008). On "sourcery," or code as fetish. *Configurations*, *16*(3), 299–324. doi: 10.1353/con.0.0064

Cockburn, C. (1985). Caught in the wheels: The high cost of being a female cog in the male machinery of engineering. In D. MacKenzie & J. Wajcman (Eds.), *The social shaping of technology: How the refrigerator got its hum* (pp. 55–65). Philadelphia: Open University Press.

Cockburn, C. (1993). Feminism/constructivism in technology studies: Notes on genealogy and recent developments. *Conference on European Theoretical Perspectives on New Technology: Feminism, Constructivism and Utility,*. CRICT, Brunel University, 16th–17th September.

Cockburn, I. M., & MacGarvie, M. J. (2009). Patents, thickets and the financing of early-stage firms: Evidence from the software industry. *Journal of Economics & Management Strategy*, *18*(3), 729–773. doi:

10.1111/j.1530-9134.2009.00228.x

Coleman, B., & Hill, M. (2004). How Free Became Open and Everything Else under the Sun. *M/C Journal*, *7*(3). doi: 10.5204/mcj.2352

Coleman, E. G. (2004). The political agnosticism of free and open source software and the inadvertent politics of contrast. *Anthropological Quarterly*, *77*(3), 507–519. Retrieved from http://www.jstor.org/stable/3318232

Coleman, E. G. (2009). Code is speech: Legal tinkering, expertise, and protest among free and open source software developers. *Cultural Anthropology*, *24*(3), 420–454. doi: 10.1111/j.1548-1360.2009.01036.x

Coleman, E. G. (2012). *Coding freedom: The ethics and aesthetics of hacking*. Princeton: Princeton University Press.

Coleman, E. G., & Golub, A. (2008). Hacker practice: Moral genres and the cultural articulation of liberalism. *Anthropological Theory*, *8*(3), 255–277. doi: 10.1177/1463499608093814

Collins, K. (2018, June 11). Net Neutrality Has Officially Been Repealed. Here’s How That Could Affect You. *The New York Times*. Retrieved from https://nyti.ms/2HAyfId

Cooper, A., & Brown, I. (2015). Net Neutrality: Discrimination, competition, and innovation in the UK and US. *ACM Transactions on Internet Technology*, *15*(1), 1–21. doi: 10.1145/2700055

Cordella, A., & Shaikh, M. (2003). Actor Network Theory and After: What’s New for IS Resarch? *ECIS 2003*, 40. Retrieved from http://aisel.aisnet.org/ecis2003/40

Coriat, B., & Orsi, F. (2002). Establishing a new intellectual property rights regime in the United States: Origins, content and problems. *Research*

*Policy*, *31*, 1491–1507. doi: 10.1016/S0048-7333(02)00078-1

Croeser, S. (2012). Contested technologies: The emergence of the digital liberties movement. *First Monday*, *17*(8). doi: 10.5210/fm.v17i8.4162

Crowston, K., & Howison, J. (2005). The social structure of free and open source software development. *First Monday*, *10*(2). doi: 10.5210/fm.v0i0.1478

Dafermos, G., & Söderberg, J. (2009). The hacker movement as a continuation of labour struggle. *Capital & Class*, *33*(1), 53–73. doi: 10.1177/030981680909700104

Dahlberg, L. (2014). Capitalism as Discoursive System. *Critical Discourse Studies*, *11*(3), 257–271. doi: 10.1080/17405904.2014.915384

Daliot-Bul, M. (2007). Japan’s mobile technoculture: the production of a cellular playscape and its cultural implications. *Media, Culture & Society*, *29*(6), 954–971. doi: 10.1177/0163443707081710

Dalle, J.-M., & Jullien, N. (2003). ‘Libre’ software: turning fads into institutions? *Research Policy*, *32*(1), 1–11. doi: 10.1016/S0048-7333(02)00003-3

Darawsheh, W. (2014). Reflexivity in research: Promoting rigour, reliability and validity in qualitative research. *International Journal of Therapy and Rehabilitation*, *21*(12), 560–568. doi: 10.12968/ijtr.2014.21.12.560

De Filippi, P., & Vieira, M. S. (2014). The commodification of information commons: The case of Cloud Computing. *The Columbia Science and Technology Law Review*, *16*(1), 102–143. Retrieved from https://ssrn.com/abstract=2488127

De Paoli, S., Teli, M., & D’Andrea, V. (2008). Free and open source licenses in community life: Two empirical cases. *First Monday*, *13*(10). doi:

10.5210/fm.v13i10.2064

Dedrick, J., Kraemer, K. L., & Linden, G. (2011). The distribution of value in the mobile phone supply chain. *Telecommunications Policy*, *35*(6), 505–521. doi: 10.1016/j.telpol.2011.04.006

DeLanda, M. (2001). *Open-Source: A movement in Search of a Philosophy*. Retrieved from https://www.cddc.vt.edu/host/delanda/pages/opensource.htm

Delautre, G. (2017). The distribution of value added among firms and countries: The case of the ICT manufacturing sector. In *International Labour Office, Research Department Working Paper*. Geneva. Retrieved from http://ilo.org/wcmsp5/groups/public/---dgreports/---inst/documents/publication/wcms_544190.pdf

Dobbernack, J. (2010). 'Things fall apart': Social imaginaries and the politics of cohesion. *Critical Policy Studies*, *4*(2), 146–163. doi: 10.1080/19460171.2010.490637

Doherty, N. F., Coombs, C. R., & Loan-Clarke, J. (2006). A re-conceptualization of the interpretive flexibility of information technologies: Redressing the balance between the social and the technical. *European Journal of Information Systems*, *15*(6), 569–582. doi: 10.1057/palgrave.ejis.300065

Drahos, P., & Braithwaite, J. (2002). *Information feudalism: Who owns the knowledge economy?* London: Earthscan.

Edwards, G. (2013). Smartphone Patent Wars: Timeline of the Apple v. Samsung Legal Battle. Retrieved February 11, 2014, from Quandary Peak Research website: https://quandarypeak.com/2013/09/smartphone-patent-wars-the-apple-vs-samsung-legal-battle/

Elliott, M. S., & Kraemer, K. L. (2008). *Computerization movements and technology diffusion: From mainframes to ubiquitous computing*. Modford: Information Today, Inc.

Elliott, M. S., & Scacchi, W. (2008). Mobilization of software developers: The free software movement. *Information Technology & People*, *21*(1), 4–33. doi: 10.1108/09593840810860315

Elzinga, A. (2004). Making Science and Technology Studies Relevant for Technology Policy - Gains and Losses? *Social Studies of Science*, *34*(6), 949–956. Retrieved from http://www.jstor.org/stable/4144352

Fast, K., Ljungberg, E., & Braunerhielm, L. (2019). On the social construction of geomedia technologies. *Communication and the Public*, *4*(2), 89–99. doi: 10.1177/2057047319853049

Feenberg, A. (1991). *Critical theory of technology*. New York: Oxford University Press.

Feenberg, A. (1996). Marcuse or Habermas: Two critiques of technology. *Inquiry*, *39*(1), 45–70. doi: 10.1080/00201749608602407

Feenberg, A. (1999). *Questioning Technology*. London: Routledge.

Feenberg, A. (2002). *Transforming technology: A critical theory revisited*. New York: Oxford University Press.

Feenberg, A. (2004). Democratic rationalization: Technology, power, and freedom. *Readings in the Philosophy of TechnologyReadings in the Philosophy of Technology*. Retrieved from http://www.sfu.ca/~andrewf/demrat.pdf

Feenberg, A. (2005a). Critical Theory of Technology: An Overview. *Tailoring Biotechnologies*, *1*(1), 47–64. Retrieved from http://www.sfu.ca/~andrewf/ctt.htm

Feenberg, A. (2005b). *Heidegger and Marcuse: The catastrophe and redemption of history*. New York: Routledge.

Feenberg, A. (2010). *Between reason and experience: Essays in technology and modernity*. Cambridge: The MIT Press.

Feenberg, A. (2013). The Mediation is the Message. *Techné: Research in Philosophy and Technology*, *17*(1), 7–24. doi: 10.5840/techne20131712

Feenberg, A. (2017). *Technosystem*. Cambridge: Harvard University Press.

Feenberg, A. (2020). Critical Constructivism, Postphenomenology and the Politics of Technology. *Techné: Research in Philosophy and Technology*, *24*(1), 27–40. doi: 10.5840/techne2020210116

Flanagin, A. J., Flanagin, C., & Flanagin, J. (2010). Technical code and the social construction of the internet. *New Media & Society*, *12*(2), 179–196. doi: 10.1177/1461444809341391

Ford, H., & Wajcman, J. (2017). 'Anyone can edit', not everyone does: Wikipedia's infrastructure and the gender gap. *Social Studies of Science*, *47*(4), 511–527. doi: 10.1177/0306312717692172

Fortunati, L. (2006). User design and the democratization of the mobile phone. *First Monday*. Retrieved from https://doi.org/10.5210/fm.v0i0.1615

Franco "Bifo" Berardi. (2013). Forward. In G. Cox & A. McLean (Eds.), *Speaking Code: Codins as aesthetic and politicla expression* (pp. ix–xiii). doi: 10.7551/mitpress/8193.001.0001

Free the Slaves. (2013). *Investigative Report: Congo's Mining Slaves*. Retrieved from https://www.freetheslaves.net/wp-content/uploads/2015/03/Congos-Mining-Slaves-web-130622.pdf

Fuchs, C. (2010). Labor in Informational Capitalism and on the Internet. *The Information Society*, *26*(3), 179–196. doi: 10.1080/01972241003712215

Fuchs, C. (2011a). A Contribution to the Critique of the Political Economy of Google. *Fast Capitalism*, *8*(1), 31–50. doi: 10.32855/fcapital.201101.006

Fuchs, C. (2011b). Web 2.0, prosumption, and surveillance. *Surveillance & Society*, *8*(3), 288–309. Retrieved from fuchs.uti.at/477/

Fuchs, C. (2012). Google Capitalism. *TripleC: Communication, Capitalism & Critique*, *10*(1), 42–48. doi: 10.31269/triplec.v10i1.304

Fuchs, C. (2014). Theorising and analysing digital labour: From global value chains to modes of production. *The Political Economy of Communication*, *1*(2). Retrieved from http://www.polecom.org/index.php/polecom/article/view/19

Fuller, M. (2006). *Software studies: A lexicon*. Cambridge: The MIT Press.

Gehring, R. A. (2006). The institutionalization of Open Source. *Poiesis Und Praxis*, *4*(1), 54–73. doi: 10.1007/s10202-005-0012-1

Germanwatch. (2012). *NO FAIR PHONES YET*. Retrieved from https://www.somo.nl/no-fair-phones-yet/

Gillespie, T. (2010). The politics of 'platforms.' *New Media & Society*, *12*(3), 347–364. doi: 10.1177/1461444809342738

Glazier, L. P. (2006). Io Sono At Swoons. In T. Swiss & A. Morris (Eds.), *New Media Poetics: Contexts/Technotexts/Theories* (pp. 211–2017). Cambridge: The MIT Press.

Glynos, J., Howarth, D., Norval, A., & Speed, E. (2009). Discourse

Analysis: Varieties and Methods. In *ESRC National Centre for Research Methods*. Retrieved from http://eprints.ncrm.ac.uk/796/

Goggin, G. (2009). Adapting the mobile phone: The iPhone and its consumption. *Continuum: Journal of Media & Cultural Studies*, *23*(2), 231–244. doi: 10.1080/10304310802710546

Goggin, G. (2010). *Global mobile media*. London: Routledge.

Goggin, G. (2011). Ubiquitous Apps: The politics of openness in global mobile cultures. *Digital Creativity*, *22*(3), 148–159. doi: 10.1080/14626268.2011.603733

Goodman, D. J., & Myers, R. A. (2005). 3G Cellular Standards and Patents. *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, *1*, 415–420. Maui, HI: IEEE. doi: 10.1109/WIRLES.2005.1549445

Guillemin, M., & Gillam, L. (2004). Ethics, reflexivity, and “ethically important moments” in research. *Qualitative Inquiry*, *10*(2), 261–280. doi: 10.1177/1077800403262360

Hamilton, E., & Feenberg, A. (2005). The Technical Codes of Online Education. *E-Learning*, *2*(2), 104. doi: 10.2304/elea.2005.2.2.1

Han, C., Reyes, I., Feal, Á., Reardon, J., Wijesekera, P., Vallina-Rodriguez, N., … Egelman, S. (2020). The Price is (Not) Right: Comparing Privacy in Free and Paid Apps. *Proceedings on Privacy Enhancing Technologies*, *2020*(3), 222–242. doi: 10.2478/popets-2020-0050

Hård, M. (1993). Beyond harmony and consensus: A social conflict approach to technology. *Science, Technology, & Human Values*, *18*(4), 408–432. doi: 10.1177/016224399301800402

Hars, A., & Ou, S. (2002). Working for free? Motivations for participating in open-source projects. *International Journal of Electronic Commerce*, *6*(3), 25–39. doi: 10.1109/HICSS.2001.927045

Harvey, P., & Krohn-Hansen, C. (2018). Introduction. Dislocating labour: anthropological reconfigurations. *Journal of the Royal Anthropological Institute*, *24*(S1), 10–28. doi: 10.1111/1467-9655.12796

Higgins, P. (2012). The Netherlands Passes Net Neutrality Legislation. Retrieved June 1, 2020, from https://www.eff.org/deeplinks/2012/05/netherlands-passes-net-neutrality-legislation

Himanen, P. (2001). *The Hacker Ethic and the Spirit of the Information Age*. New York: Random House Trade Paperbacks.

Himanen, P. (2004). The hacker ethic as the culture of the information age. In M. Castells (Ed.), *The Network Society: A Cross-cultural Perspective* (pp. 420–432). Cheltenham: Edward Elgar Publishing.

Howarth, D. (2010). Power, discourse, and policy: articulating a hegemony approach to critical policy studies. *Critical Policy Studies*, *3*(3–4), 309–335. doi: 10.1080/19460171003619725

Howarth, D., & Stavrakakis, Y. (2000). Introducing Discourse Theory and Political Analysis. In D. J. Howarth, A. J. Norval, & Y. Stavrakakis (Eds.), *Discourse Theory and Political Analysis: Identities, Hegemonies and Social Change* (pp. 1–37). Manchester: Manchester University Press.

Howarth, D., & Torfing, J. (Eds.). (2004). *Discourse theory in European politics: Identity, policy and governance*. London: Palgrave Macmillan. doi: 10.1057/9780230523364

Howcroft, D., Mitev, N., & Wilson, M. (2004). What we may learn from the social shaping of technology approach. In J. Mingers & L. Willcocks

(Eds.), *Social theory and philosophy for information systems* (pp. 329–371). Chichester: John Wiley & Sons Chichester.

Hrynyshyn, D. (2008). Globalization, nationality and commodification: The politics of the social construction of the internet. *New Media & Society*, *10*(5), 751–770. doi: 10.1177/1461444808094355

Huang, E. M., & Truong, K. N. (2008). Breaking the disposable technology paradigm. *Proceeding of the Twenty-Sixth Annual CHI Conference on Human Factors in Computing Systems - CHI '08*, 323–332. New Yor: ACM Press. doi: 10.1145/1357054.1357110

Hughes, J., & Lang, K. (2006). Transmutability: Digital decontextualization, manipulation, and recontextualization as a new source of value in the production and consumption of culture products. *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, *8*, 168–178. IEEE. doi: 10.1109/HICSS.2006.511

Hughes, T. P. (1985). Edison and electric light. In D. MacKenzie & J. Wajcman (Eds.), *The social shaping of technology*. Philadelphia: Open University Press.

Hughes, T. P. (2012). The evolution of large technological systems. In T. P. Hughes, W. E. Bijker, & P. Trevor (Eds.), *The social construction of technological systems: New directions in the sociology and history of technology* (pp. 45–76). Cambridge: The MIT Press.

Huijstee, M. van, & Haan, E. de. (2009). *Fair Phones in the Netherlands: Its your call*. Retrieved from https://www.somo.nl/fair-phones-in-the-netherlands-its-your-call/

Humphreys, L. (2005). Reframing social groups, closure, and stabilization in the social construction of technology. *Social Epistemology*, *19*(2–3), 231–253. doi: 10.1080/02691720500145449

Hutchby, I. (2001). Technologies, texts and affordances. *Sociology*, *35*(2), 441–456. doi: 10.1177/S0038038501000219

Huws, U. (2014). *Labor in the global digital economy: The cybertariat comes of age*. New York: Monthly Review Press.

Ippolita. (2013). *The Dark Side of Google*. Amsterdam: Institute of Network Cultures.

ITU. (2019). Time series of ICT data for the world (2005-2019) Households with a computer and with Internet access. Retrieved March 2, 2020, from https://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2019/ITU_Key_2005-2019_ICT_data_with LDCs_28Oct2019_Final.xls

Jansz, J., & Theodorsen, J. (2009). Modifying video games on Web 2.0: an exploration of motives for publishing creative game content. *Proceedings of the Annual Meeting of the International Communication Association*. Chicago: International Communication Association.

Jenkins, H. (2006). *Convergence Culture: where new and old media collide*. New York: New York University Press.

Jesiek, B. (2003). Democratizing software: Open source, the hacker ethic, and beyond. *First Monday*, *8*(10). doi: 10.5210/fm.v8i10.1082

Joode, R. V. W., Lin, Y., & David, S. (2006). Rethinking free, libre and open source software. *Knowledge, Technology & Policy*, *18*(4), 5–16. doi: 10.1007/s12130-006-1001-y

Kang, H., & McAllister, M. P. (2011). Selling you and your clicks: Examining the audience commodification of google. *TripleC*, *9*(2), 141–153. doi: 10.31269/vol9iss2pp141-153

Kelly, A. (2019). Apple and Google named in US lawsuit over Congolese child cobalt mining deaths. Retrieved March 1, 2020, from The Guardian website: https://www.theguardian.com/global-development/2019/dec/16/apple-and-google-named-in-us-lawsuit-over-congolese-child-cobalt-mining-deaths

Kelty, C. M. (2008). *Two Bits: The cultural significance of free software*. Durham: Duke University Press. Retrieved from http://read.dukeupress.edu/books/book/1136/Two-BitsThe-Cultural-Significance-of-Free-Software

Kelty, C. M. (2013). There is no free software. *Journal of Peer Production*, *3*(July). Retrieved from http://peerproduction.net/issues/issue-3-free-software-epistemics/debate/there-is-no-free-software/

Khoo, M. (2005). Technologies aren't what they used to be: Problematising closure and relevant social groups. *Social Epistemology*, *19*(2–3), 283–285. doi: 10.1080/02691720500145506

Kirkpatrick, G. (2013). Formal Bias and Normative Critique of Technology Design. *Techné: Research in Philosophy and Technology*, *17*(1), 25–46. doi: 10.5840/techne20131713

Kirkpatrick, G. (2020). The theory of bias and the ethics of technology design. In G. Kirkpatrick (Ed.), *Technical politics: Andrew Feenberg's critical theory of technology*. Manchester: Manchester University Press. doi: 10.7765/9781526105349.00006

Klein, H. K., & Kleinman, D. L. (2002). The social construction of technology: Structural considerations. *Science, Technology, & Human Values*, *27*(1), 28–52. doi: 10.1177/016224390202700102

Kleiner, D. (2010). *The Telekommunist Manifesto*. Amsterdam: Institute of Network Cultures.

Kling, R. (1991). Computerization and social transformations. *Science, Technology, & Human Values*, *16*(3), 342–367. doi: 10.1177/016224399101600304

Kling, R., & Iacono, S. (1988). The mobilization of support for computerization: The role of computerization movements. *Social Problems*, *35*(3), 226–243. Retrieved from https://www.jstor.org/stable/800620

Koetsier, J. (2012, May). Apple and Microsoft could make 600% more from Android than Google in 2013. Retrieved May 16, 2013, from Venturebeat website: http://venturebeat.com/2012/11/13/in-2013-apple-and-microsoft-could-make-600-more-from-android-than-google/

Kogut, B. (2001). Open-source software dDevelopment and distributed innovation. *Oxford Review of Economic Policy*, *17*(2), 248–264. doi: 10.1093/oxrep/17.2.248

Kostakis, V., & Bauwens, M. (2014). *Network society and future scenarios for a collaborative economy*. London: Palgrave Macmillan. doi: 10.1057/9781137406897

Kostakis, V., & Stavroulakis, S. (2013). The Parody of the Commons. *TripleC: Communication, Capitalism & Critique*, *11*(2), 412–424. doi: 10.31269/triplec.v11i2.484

Kraemer, K. L., Linden, G., & Dedrick, J. (2011). *Capturing value in global networks: Apple's iPad and iPhone*. Retrieved from https://economiadeservicos.com/wp-content/uploads/2017/04/value_ipad_iphone.pdf

Kreiss, D., Finn, M., & Turner, F. (2011). The limits of peer production: Some reminders from Max Weber for the network society. *New Media & Society*, *13*(2), 243–259. doi: 10.1177/1461444810370951

Krishnamurthy, S. (2005). Cave or community? An empirical examination of 100 mature open source projects. *First Monday*. doi: 10.5210/fm.v0i0.1477

Kücklich, J. (2005). Precarious playbour: Modders and the digital games industry. *The Fibreculture Journal*, Vol. 5. Retrieved from http://five.fibreculturejournal.org/fcj-025-precarious-playbour-modders-and-the-digital-games-industry/

Laclau, E., & Mouffe, C. (2001). *Hegemony and socialist strategy: Towards a radical democratic politics* (2nd ed.). London: Verso.

Lakhani, K. R., & von Hippel, E. (2003). How open source software works: “free” user-to-user assistance. *Research Policy*, *32*(6), 923–943. doi: 10.1016/S0048-7333(02)00095-1

Lakhani, K., & Wolf, R. G. (2003). Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. *SSRN Electronic Journal*. doi: 10.2139/ssrn.443040

Lassinantti, J., Ståhlbröst, A., & Runardotter, M. (2019). Relevant social groups for open data use and engagement. *Government Information Quarterly*, *36*(1), 98–111. doi: 10.1016/j.giq.2018.11.001

Latour, B. (1987). *Science in action: How to follow scientists and engineers through society*. Cambridge: Harvard University Press.

Lazzarato, M. (1996). Immaterial labor. In P. Virno & M. Hardt (Eds.), *Radical thought in Italy: A potential politics* (pp. 133–149). Minneapolis: University of Minnesota Press.

Lehr, W., & McKnight, L. W. (2003). Wireless Internet access: 3G vs. WiFi? *Telecommunications Policy*, *27*(5–6), 351–370. doi: 10.1016/S0308-5961(03)00004-1

Lemley, M. A. (1995). Convergence in the law of software copyright. *High Techology Law Journal*, *10*(1), 1–34. Retrieved from https://www.jstor.org/stable/24122470

Lemley, M. A. (2002). Intellectual Property Rights and Standard-Setting Organizations. *California Law Review*, *90*(6), 1889–1980. doi: 10.2307/3481437

Lemley, M. A. (2007). Ten things to do about patent holdup of standards (and one not to). *Boston College Law Review*, *48*(1), 149–168.

Lemley, M. A., & Shapiro, C. (2007). Patent Hold-Up and Royalty Stacking. *UC Berkeley: Competition Policy Center*. Retrieved from https://escholarship.org/uc/item/8638s257

Lerner, J., & Tirole, J. (2005). Economic perspectives on open. In J. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software* (pp. 47–78). Cambridge: The MIT Press.

Lessig, L. (1999). *Code and Other Laws of Cyberspace Basic Books*. New York: Basic Books.

Lessig, L. (2004). *Free culture*. New York: The Penguin Press.

Levy, J. (2014). Verizon Escalates Assault on Net Neutrality. Retrieved February 15, 2014, from Free Press website: http://www.savetheinternet.com/blog/2014/02/05/verizon-escalates-assault-net-neutrality

Levy, S. (1984). *Hackers: Heroes of the computer revolution*. Garden City: NY: Anchor Press/Doubleday.

Liao, T. (2018). Mobile versus headworn augmented reality: How visions of the future shape, contest, and stabilize an emerging technology. *New*

*Media & Society*, *20*(2), 796–814. doi: 10.1177/1461444816672019

Lievrouw, L. A. (2006). New Media Design and Development: Diffusion of Innovations v Social Shaping of Technology. In *Handbook of new media: Social shaping and social consequences of ICTs* (pp. 246–265). London: Sage Publications. doi: 10.4135/9781446211304.n14

Lievrouw, L. A., & Livingstone, S. (2006). Introduction to the updated student edition. In *Handbook of new media: Social shaping and social consequences of ICTs* (pp. 1–14). London: Sage Publications. doi: 10.4135/9781446211304.n1

Lin, Y. (2005). The future of sociology of FLOSS. *First Monday*, *2*(Special Issue: Open Source). doi: 10.5210/fm.v0i0.1467

Lineage OS Wiki. (2020). Devices. Retrieved July 21, 2020, from https://wiki.lineageos.org/devices/

Lund, A. (2014). Playing, gaming, working and labouring: Framing the concepts and relations. *TripleC: Communication, Capitalism & Critique*, *12*(2), 735–801. doi: 10.31269/triplec.v12i2.536

Mackay, H., & Gillespie, G. (1992). Extending the social shaping of technology approach: Ideology and appropriation. *Social Studies of Science*, *22*(4), 685–716. doi: 10.1177/030631292022004006

Mackenzie, A. (2005). The performativity of code. *Theory, Culture & Society*, *22*(1), 71–92. doi: 10.1177/0263276405048436

MacKenzie, D., & Wajcman, J. (1985). Introductory essay: The social shaping of technology. In D. MacKenzie & J. Wajcman (Eds.), *The social shaping of technology: How the refrigerator got its hum* (1st ed., pp. 2–25). Philadelphia: Open University Press.

MacKenzie, D., & Wajcman, J. (1999). Introductory Essay: The social shaping of technology. In D. MacKenzie & J. Wajcman (Eds.), *The social shaping of technology* (2nd ed.). Buckingham: Open university press.

Magaudda, P. (2010). Hacking practices and their relevance for consumer studies: The example of the 'jailbreaking'of the iPhone. *Consumers, Commodities and Consumption*, *12*(1), 11–12. Retrieved from https://csrn.camden.rutgers.edu/newsletters/12-1/magaudda.htm

Mangalousi, D. (2013). *Debating Over the Unitary Patent : Implications on Software Development*. [Master Thesis] Erasmus University Rotterdam.

Mangalousi, D., & Mosemghvdlishvili, L. (2014). The debate on establishing the Digital Single Market in Europe and its implications for Net Neutrality. *International Association for Media and Communication Research Annual Conference*. Hyderabad: IAMCR.

Manovich, L. (2002). *The language of new media*. Cambridge: The MIT Press.

Manovich, L. (2013). *Software takes command*. New York: Bloomsbury.

Marcuse, H. (2013). *One-dimensional Man: Studies in the ideology of advanced industrial society*. London: Routledge. (Original work published 1964).

Mauthner, N. S., & Doucet, A. (2003). Reflexive accounts and accounts of reflexivity in qualitative data analysis. *Sociology*, *37*(3), 413–431. doi: 10.1177/00380385030373002

McPherson, T. (2006). Reload: Liveness, mobility, and the web. In W. H. K. Chun & T. Keenan (Eds.), *New media, old media: A history and theory reader* (1st ed., pp. 199–208). New York: Routledge.

Meinrath, S., & Pickard, V. (2006). *The new network neutrality: Criteria for Internet freedom*. Retrieved from https://ssrn.com/abstract=2119750

Mian, S. Q., Teixeira, J., & Koskivaara, E. (2011). Open-source software implications in the competitive mobile platforms market. *Conference on E-Business, e-Services and e-Society*, 110–128. doi: 10.1007/978-3-642-27260-8_9

Moody, G. (2001). *Rebel code: Linux and the open source revolution* (1st ed.). Cambridge: Perseus Publishing.

Mosemghvdlishvili, L., & Jansz, J. (2013). Negotiability of technology and its limitations: The politics of app development. *Information, Communication & Society*, *16*(10), 1596–1618. doi: 10.1080/1369118X.2012.735252

Mosemghvdlishvili, L., & Jansz, J. (2018). Free your 'most open' Android: A comparative discourse analysis on Android. *Critical Discourse Studies*, *17*(1), 56–71. doi: 10.1080/17405904.2018.1554536

Nguyen, L. U. (2016). Infrastructural action in Vietnam: Inverting the techno-politics of hacking in the global South. *New Media & Society*, *18*(4), 637–652. doi: 10.1177/1461444816629475

Nixon, B. (2014). Toward a Political Economy of 'Audience Labour' in the Digital Era. *TripleC: Communication, Capitalism & Critique*, *12*(2), 713–734. doi: 10.31269/triplec.v12i2.535

O'Connor, D. (2012). One in six active us patents pertain to the smartphone. Retrieved February 3, 2014, from Disruptive Competition Project website: http://www.project-disco.org/intellectual-property/one-in-six-active-u-s-patents-pertain-to-the-smartphone/

O'Neil, M. (2015). Labour out of Control: The political economy of capitalist and ethical organizations. *Organization Studies*, *36*(12), 1627–1647. doi: 10.1177/0170840615585339

Orlikowski, W. J. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organization Science*, *3*(3), 398–427. Retrieved from http://www.jstor.org/stable/2635280

Padala, S. H., Mendez, C. J., Dias, L. F., Steinmacher, I., Steine Hanson, Z., Hilderbrand, C., … Sarma, A. (2020). How Gender-biased Tools Shape Newcomer Experiences in OSS Projects. *IEEE Transactions on Software Engineering*. doi: 10.1109/TSE.2020.2984173

Paiano, A., Lagioia, G., & Cataldo, A. (2013). A critical analysis of the sustainability of mobile phone use. *Resources, Conservation and Recycling*, *73*, 162–171. doi: 10.1016/j.resconrec.2013.02.008

Paik, Y., & Zhu, F. (2013). The impact of patent wars on firm strategy: Evidence from the global smartphone market. *35th DRUID Celebration Conference 2013*, 1–45. Citeseer. doi: 10.1007/s13398-014-0173-7.2

Paik, Y., & Zhu, F. (2016). The impact of patent wars on firm strategy: Evidence from the global smartphone industry. *Organization Science*, *27*(6), 1397–1416. doi: 10.1287/orsc.2016.1092

Parish, J. (2011). Apple vs. Samsung: the complete lawsuit timeline. *The Verge*. Retrieved from http://www.theverge.com/apple/2011/11/2/2533472/apple-vs-samsung

Park, J. (2019, October 19). Samsung ends mobile phone production in China. *Reuters*. Retrieved from https://www.reuters.com/article/us-samsung-elec-china/samsung-ends-mobile-phone-production-in-china-idUSKBN1WH0LR

Pasquinelli, M. (2009). Google's PageRank algorithm: A diagram of cognitive capitalism and the rentier of the common intellect. In K.

Becker & F. Stalder (Eds.), *Deep search: The politics of search beyond Google* (pp. 152–163). London: Transaction Publishers.

Pense, C. M., & Cutcliffe, S. H. (2007). Risky talk: Framing the analysis of the social implications of nanotechnology. *Bulletin of Science, Technology & Society*, *27*(5), 349–366. doi: 10.1177/0270467607306592

Pinch, T. J., & Bijker, W. E. (1984). The social construction of facts and artefacts: Or how the sociology of science and the sociology of technology might benefit each other. *Social Studies of Science*, *14*(3), 399–441. doi: 10.1177/030631284014003004

Poell, T., Nieborg, D., & van Dijck, J. J. (2019). Platformisation. *Internet Policy Review*, *8*(4). doi: 10.14763/2019.4.1425

Powell, A. (2012). Democratizing production through open source knowledge: from open software to open hardware. *Media, Culture and Society*, *34*(6), 691–708. doi: 10.1177/0163443712449497

Pozzebon, M., Titah, R., & Pinsonneault, A. (2006). Combining social shaping of technology and communicative action theory for understanding rhetorical closure in IT. *Information Technology & People*, *19*(3), 244–271. doi: 10.1108/09593840610689840

Prell, C. (2009). Rethinking the social construction of technology through ‘following the actors’: A reappraisal of technological frames. *Sociological Research Online*, *14*(2), 36–47. doi: 10.5153/sro.1913

Pun, N., Tse, T., & Ng, K. (2019). Challenging digital capitalism: SACOM’s campaigns against Apple and Foxconn as monopoly capital. *Information, Communication & Society*, *22*(9), 1253–1268. doi: 10.1080/1369118X.2017.1414290

Qiu, J. L. (2018). China’s digital working class and circuits of labor.

*Communication and the Public*, *3*(1), 5–18. doi: 10.1177/2057047318755529

Qiu, J. L., Gregg, M., & Crawford, K. (2014). Circuits of labour: A labour theory of the iPhone era. *TripleC: Communication, Capitalism & Critique*, *12*(2), 564–581. doi: 10.31269/triplec.v12i2.540

Raento, M., Oulasvirta, A., & Eagle, N. (2009). Smartphones: An emerging tool for social scientists. *Sociological Methods & Research*, *37*(3), 426–454. doi: 10.1177/0049124108330005

Raymond, E. (1998). Goodbye, “free software”; hello, “open source.” Retrieved from http://www.catb.org/esr/open-source.html

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, *12*(3), 23–49. doi: 10.1007/s12130-999-1026-0

Raymond, E. (2001). *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA: OReilly Media Inc.

Rear, D., & Jones, A. (2013). Discursive struggle and contested signifiers in the arenas of education policy and work skills in Japan. *Critical Policy Studies*, *7*(4), 375–394. doi: 10.1080/19460171.2013.843469

Rigi, J. (2013). Peer production and Marxian communism: Contours of a new emerging mode of production. *Capital & Class*, *37*(3), 397–416. doi: 10.1177/0309816813503979

Robles, G., Amor, J. J., Gonzalez-Barahona, J. M., & Herraiz, I. (2005). Evolution and growth in large libre software projects. *Eighth International Workshop on Principles of Software Evolution (IWPSE’05)*, 165–174. doi: 10.1109/IWPSE.2005.17

Ross, D. (2013). The place of free and open source software in the social apparatus of accumulation. *Science & Society*, *77*(2), 202–226. doi: 10.1521/siso.2013.77.2.202

Russell, S. (1986). The social construction of artefacts: a response to Pinch and Bijker. *Social Studies of Science*, *16*(2), 331–346. Retrieved from http://www.jstor.org/stable/285210

Saldaña, J. (2015). *The coding manual for qualitative researchers*. London: Sage Publications.

Sandoval, M. (2016). Foxconned labour as the dark side of the information age: Working conditions at Apple's contract manufacturers in China. In C. Fuchs & V. Mosco (Eds.), *Marx in the age of digital capitalism* (pp. 350–395). Leiden: Brill.

Sandoval, M., & Bjurling, K. (2013). Challenging labour-eorking conditions in the electronics industry. In S. Garwood, S. Croeser, & C. Yakinthou (Eds.), *Lessons for Social Change in the Global Economy: Voices from the Field* (pp. 99–124). London: Lexington Books.

Sawhney, H. (2009). Innovations at the edge: The impact of mobile technologies on the character of the Internet. In G. Goggin & L. Hjorth (Eds.), *Mobile Technologies: From telecommunications to media* (pp. 105–117). New York: Routledge.

Sawhney, H., & Lee, S. (2005). Arenas of innovation: understanding new configurational potentialities of communication technologies. *Media, Culture & Society*, *27*(3), 391–414. doi: 10.1177/0163443705051750

Scelsi, C. (2007). The iPhone - Hacking and Cracking and Copyright. *Entertainment and Sports Lawyer*, *25*(3), 18–22.

Schäfer, M. T. (2008). *Bastard Culture! User participation and the extension of the cultural industries*. Utrecht: Utrecht University.

Schoonmaker, S. (2007). Globalization from below: Free software and alternatives to neoliberalism. *Development and Change*, *38*(6), 999–1020. doi: 10.1111/j.1467-7660.2007.00462.x

Shapiro, C. (2000). Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard-Setting. *Innovation Policy and the Economy*, *1*, 119–150. Retrieved from http://www.jstor.org/stable/25056143

Shaw, A. (2014). The internet is full of jerks, because the world is full of jerks: What feminist theory teaches us about the internet. *Communication and Critical/Cultural Studies*, *11*(3), 273–277. doi: 10.1080/14791420.2014.926245

Siles, I. (2012). The rise of blogging: Articulation as a dynamic of technological stabilization. *New Media & Society*, *14*(5), 781–797. doi: 10.1177/1461444811425222

Singh, M. (2020, July 24). Apple begins assembling iPhone 11 in India. *Techcrunch*. Retrieved from https://techcrunch.com/2020/07/24/apple-begins-assembling-iphone-11-in-india/

Smith, B. (2011). Statement on OpenOffice.org's move to Apache. Retrieved July 11, 2012, from Free Software Foundation website: http://www.fsf.org/news/openoffice-apache-libreoffice

Smith, P. J., & Smythe, E. (2009). Open spaces, open sources: The World Social Forum and international communication rights in a digital world. *Information, Communication & Society*, *12*(6), 793–816. doi: 10.1080/13691180802158524

Smith, T., Sonnenfeld, D. A., & Pellow, D. N. (Eds.). (2006). *Challenging the chip: Labor rights and environmental justice in the global electronics industry*. Philadelphia: Temple University Press.

Söderberg, J. (2011). *Free software to open hardware: Critical theory on the* *frontiers of hacking*. Goteborg: University of Goteborg.

Söderberg, J. (2015). *Hacking capitalism: The free and open source software movement* (2nd ed.). New York: Routledge.

Söderberg, J. (2017). Inquiring hacking as politics: A new departure in Hacker Studies? *Science, Technology, & Human Values*, *42*(5), 969–980. doi: 10.1177/0162243916688094

Spreeuwenberg, K., & Poell, T. (2012). Android and the political economy of the mobile Internet: A renewal of open source critique. *First Monday*, *17*(7). doi: 10.5210/fm.v17i7.4050

Stallman, R. (1994). Why software should not have owners. doi: 10.1016/b978-075067932-9/50041-5

Stallman, R. (2002). Free software, free society: Selected essays of Richard M. Stallman. In J. Gay (Ed.), *Computer Networks* (1. ed). Boston: GNU press. doi: 10.1029/2004TC001693

Stallman, R. (2011). Is Android really free software? Retrieved July 15, 2012, from The Guardian website: http://www.guardian.co.uk/technology/2011/sep/19/android-free-software-stallman

Stallman, R. (2016). FLOSS and FOSS. Retrieved March 12, 2017, from GNU Operating System website: https://www.gnu.org/philosophy/floss-and-foss.html

Statista. (2014). Market share: wireless telecommunication carriers in the U.S. 2011. Retrieved February 15, 2014, from http://www.statista.com/statistics/219720/market-share-of-wireless-carriers-in-the-us-by-subscriptions

Statista. (2018). Global mobile OS market share 2009-2017. Retrieved

August 28, 2018, from https://www.statista.com/statistics/216420/global-market-share-forecast-of-smartphone-operating-systems/

Sullivan, J. L. (2011). Free, open source software advocacy as a social justice movement: The expansion of F/OSS movement discourse in the 21st century. *Journal of Information Technology & Politics*, *8*(3), 223–239. doi: 10.1080/19331681.2011.592080

Teece, D. J. (1986). Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy*, *15*(6), 67–88. doi: 10.1016/0048-7333(86)90027-2

Teece, D. J. (2012). Next-generation competition: New concepts for understanding how innovation shapes competition and policy in the digital economy. *Journal of Law, Economics & Policy*, *9*(1), 97–118.

Teece, D. J. (2018). Profiting from innovation in the digital economy: Enabling technologies, standards, and licensing models in the wireless world. *Research Policy*, *47*(8), 1367–1387. doi: 10.1016/j.respol.2017.01.015

Teli, M. (2005). Looking at free and open source software: A study about F/OSS developers' culture. In M. Scotto & G. Succi (Eds.), *OSS 2005 - Proceedings of the 1st International Conference on Open Source Systems* (pp. 324–325). Geneva, 11-15 July.

Teli, M., & De Paoli, S. (2006). Boundaries and Diversity in Free and Open Source Software. *OSS 2006 - Proceedings of the 2nd International Conference on Open Source Systems*. Retrieved from https://core.ac.uk/display/100236504

Terranova, T. (2000). Free Labour: Producing Culture for the Digital Economy. *Social Text*, *18*(2), 33–58. Retrieved from https://www.muse.jhu.edu/article/31873.

Terranova, T. (2004). *Network culture: Politics in the information age*. London: Pluto Press.

Terranova, T. (2006). Of sense and sensibility: Immaterial labour in open systems. In J. Krysa (Ed.), *Curating Immateriality: The work of the curator in the age of network systems* (pp. 27–36). New York: Autonomedia.

Torfing, J. (1999). *New theories of discourse: Laclau, Mouffe, and Žižek*. Oxford: Blackwell Publishers.

Tuomi, I. (1999). Data is more than knowledge: Implications of the reversed knowledge hierarchy for knowledge management and organizational memory. *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32.* IEEE. doi: 10.1109/HICSS.1999.77279

Tuomi, I. (2002). *Networks of innovation: Change and meaning in the age of the internet*. Oxford: Oxford University Press.

Vaismoradi, M., Jones, J., Turunen, H., & Snelgrove, S. (2016). Theme development in qualitative content analysis and thematic analysis. *Forum: Qualitative Social Research*, *20*(34). Retrieved from http://dx.doi.org/10.17169/fqs-20.3.3376

van Baalen, P. J., van Fenema, P. C., & Löbbecke, C. (2016). Extending the Social Construction of Technology (SCOT) Framework to the Digital World. *ICIS 2016 Proceedings*.

Vercellone, C. (2008). The new articulation of wages, rent and profit in cognitive capitalism. *The Art of Rent*, pp. 1–9. Retrieved from http://halshs.archives-ouvertes.fr/halshs-00265584/

Vieira, M. S., & De Filippi, P. (2014). Between Copyleft and Copyfarleft: Advanced reciprocity for the commons. *Journal of Peer Production*, *4*(Value and currency). Retrieved from http://peerproduction.net/issues/issue-4-value-and-currency/invited-comments/between-copyleft-and-copyfarleft-advance-reciprocity-for-the-commons/

Von Hippel, E. (2006). *Democratizing innovation*. Cambridge: The MIT Press.

Von Hippel, E. (2008). Democratizing Innovation: The Evolving Phenomenon of User Innovation. *International Journal of Innovation Science*, *1*(1), 29–40.

Von Krogh, G., & Von Hippel, E. (2006). The promise of research on open source software. *Management Science*, *52*(7), 975–983. doi: 10.1287/mnsc.1060.0560

Wade Morris, J., & Elkins, E. (2015). FCJ-181 There's a History for That: Apps and Mundane Software as Commodity. *The Fibreculture Journal*, (25), 63–88. doi: 10.15307/fcj.25.181.2015

Wajcman, J. (2002). Addressing technological change: The challenge to social theory. *Current Sociology*, *50*(3), 347–363. doi: 10.1177/0011392102050003004

West, J. (2003). How open is open enough? *Research Policy*, *32*(7), 1259–1285. doi: 10.1016/S0048-7333(03)00052-0

West, J., & Mace, M. (2010). Browsing as the killer app: Explaining the rapid success of Apple's iPhone. *Telecommunications Policy*, *34*(5–6), 270–286. doi: 10.1016/j.telpol.2009.12.002

Williams, R. (1997). The social shaping of information and communications technologies. In H. Kubicek, W. H. Dutton, & R. Williams (Eds.), *The*

*Social Shaping of Infonnation Superhighways: European and American Roads to the Information Society* (pp. 299–338). New York: St. Martin's Press.

Williams, R., & Edge, D. (1996). The social shaping of technology. *Research Policy*, *25*(6), 865–899. doi: 10.1016/0048-7333(96)00885-2

Williams, R., Stewart, J., & Slack, R. (Eds.). (2005). *Social Learning in Technological Innovation: Experimenting with Information and Communication Technologies*. Cheltenham: Edward Elgar Publishing.

Wilson, K. (1992). Deregulating telecommunications and the problem of natural monopoly: A critique of economics in telecommunications policy. *Media, Culture & Society*, *14*(3), 343–368. doi: 10.1177/016344392014003002

Wilson, M., & Howcroft, D. (2005). Power, politics and persuasion in IS evaluation: a focus on 'relevant social groups.' *The Journal of Strategic Information Systems*, *14*(1), 17–43. doi: 10.1016/j.jsis.2004.11.007

Winner, L. (1985). Do artifacts have politics? In D. MacKenzie & J. Wajcman (Eds.), *The social shaping of technology: How the refrigerator got its hum* (pp. 26–37). Philadelphia: Open University Press.

Winner, L. (1993). Upon opening the black box and finding it empty: Social constructivism and the philosophy of technology. *Science, Technology, & Human Values*, *18*(3), 362–378. doi: 10.1177/016224399301800306

Winseck, D. (1997). Canadian Telecommunications: A History and Political Economy of Media Reconvergence. *Canadian Journal of Communication*, *22*(2). doi: 10.22230/cjc.1997v22n2a995

Wolf, M. J., Miller, K. W., & Grodzinsky, F. S. (2009). On the meaning of free software. *Ethics and Information Technology*, *11*(4), 279–286. doi: 10.1007/s10676-009-9207-9

Wolff, R. (2019). Towards a Critical Theory of the Technosystem. *Jus Cogens: A Critical Journal of Philosophy of Law and Politics*, *1*, 173–185. doi: 10.1007/s42439-019-00011-z

Wu, T. (2003). Network neutrality, broadband discrimination. *Journal of Telecommunications and High Technology Law*, *2*, 143–177. doi: 10.2139/ssrn.388863

Zittrain, J. L. (2008). *The Future of the Internet and How to Stop it*. New Haven: Yale University Press.

Zuboff, S. (2015). Big other: Surveillance Capitalism and the Prospects of an Information Civilization. *Journal of Information Technology*, *30*(1), 75–89. doi: 10.1057/jit.2015.5

Zuckerman, E. (2010). Decentralizing the mobile phone: A second ICT4D revolution? *Information Technologies & International Development*, *6*(SE), 99–103.

# Appendix 1: Portfolio

Summary of activities and outputs during PhD trajectory

## Graduate courses

September – October 2014, *Big Data Analysis and Data Visualization* (2.5 ECTS), offered by the Erasmus Graduate School (EGS3H) at the Erasmus University Rotterdam

October – November 2014, *Philosophy of the Social Sciences and the Humanities* (2.5 ECTS), offered by the Erasmus Graduate School (EGS3H) at the Erasmus University Rotterdam

January 30-31, 2014, *RMeS Winterschool and Graduate Symposium*, at the Radboud University, Nijmegen (2 ECTS)

June 10-13, 2013 *RMeS/ERMeCC Summer School, 10-13 June 2013: Audiences and users* (2 ECTS)

January 31 - February 1, 2013. *RMeS Winterschool and Graduate Seminar* at Maastricht University (2 ECTS)

March 29-31, 2012, *Researching Mobile and Locative Media: Methods and Ethics,* PhD course organized at the Aarhus, University, Denmark (2.5 ECTS)

April – July 2012, *Academic Writing in English for PhD-students* organized by Risbo at Erasmus University Rotterdam.

## Conference presentations

January 2011: ETMAAL 2011, University of Twente, Paper (co-authored J. Jansz) Preaching, negotiating, and negating Christianity on YouTube - *presented.*

May 2011: ICA Conference in Boston, USA (attended through virtual conferencing) Poster Preaching, negotiating and negating Christianity on YouTube - *presented.*

February 2012: ETMAAL 2012, University of Leuven, Poster (co-authored J. Jansz): Negotiating interpretative flexibility of Smartphones: interviews with freelance Apps developers - *presented.*

March 8-10, 2012 Unlike Us: Understanding Social Media Monopolies and their Alternatives, Institute of Network Cultures, Amsterdam *(attended)*

May 2-4, 2012 4th ICTs and Society-Conference 2012, Uppsala, Sweden *(attended)*

July 14-29, 2012 IAMCR 2012 Durban, South Africa, Paper (co-authored J. Jansz) Analytical Model to analyze Software Development - *presented*

February 7-8, 2013 ETMAAL 2013, Rotterdam, Netherlands

(a) Paper (co-authored J. Jansz) Free your “most open” Android: Critical analysis of discourses on Android - *presented*

(b) Poster (first author S. Hoffman) Perceiving spaces through digital augmentation: an exploratory study of Augmented Reality apps - *presented*

May 28-30, 2013 ICT and Work, University of Paris (Sorbonne), Paris, France, Paper (co-authored with J. Jansz) What is free? Open-source software development by Google - *presented*

June 16-21, 2013, ICA 2013, London, the UK

(a) Paper (co-authored with J. Jansz) Understanding the backbone of technology: a conceptual model to research Code - *presented*

(b) Paper (first author S. Hoffman) Perceiving spaces through digital augmentation: an exploratory study of Augmented Reality apps - *presented*

July 15-19, 2014, IAMCR 2014, Hyderabad, India

(a) Paper (co-authored with J. Jansz) How free and open is Google's Android? An analysis into differences in open-source code development for mobiles - *presented*

(b) Paper (co-authored with D. Mangalousi), The debate on establishing the Digital Single Market in Europe and its implications for Net Neutrality - *presented*

## Review work

Reviewer Etmaal 2013
Reviewer ICA 2012, 2013
Reviewer IAMCR 2014
Reviewer New Media and Society
Reviewer Current Sociology
Reviewer for a special issue of Scan: Journal of Media Arts Culture
Discussant for the sessions: Mobile communication; Technologies, Policy, and Usage at IAMCR, 2014

## Publications

Mosemghvdlishvili, L., & Jansz, J. (2013). Negotiability of technology and its limitations: The politics of app development. *Information, Communication & Society*, *16*(10), 1596–1618. doi: 10.1080/1369118X.2012.735252

Mosemghvdlishvili, L. (2015). Mobile internet: The politics of code and networks. In A. Bechmann & S. Lomborg (Eds.), *The Ubiquitous Internet: User and Industry Perspectives*. New York: Routledge. doi: 10.4324/9781315856667

Mosemghvdlishvili, L., & Jansz, J. (2018). Free your 'most open' Android: A comparative discourse analysis on Android. Critical Discourse Studies, 17(1), 56–71. doi: 10.1080/17405904.2018.1554536

### Publications not included in PhD

Mosemghvdlishvili, L., & Jansz, J. (2013). Framing and praising Allah on YouTube: Exploring user-created videos about Islam and the motivations for producing them. *New Media & Society*, *15*(4), 482–500. doi: 10.1177/1461444812457326

Hoffman, S. & Mosemghvdlishvili, L. (2014). Perceiving spaces through digital augmentation: An exploratory study of navigational augmented reality apps*, Mobile Media & Communication, 2(3), pp. 265-280, doi: 10.1177/2050157914530700*

## Other

24-25 March 2015 initiator and co-organizer of ERMeCC & NeFCA Symposium: *Surveillance, Big Data and Digital Rights* at the Erasmus University Rotterdam