Towards a Smart Society through Personal Assistants Employing Executable Choreographies

Lenuta Alboaie *adria@info.uaic.ro*
Faculty of Computer Science of the University "Al. I. Cuza"
Iasi, Romania

Abstract

With the increased use of Internet, governments and large companies store and share massive amounts of personal data in such a way that leaves no space for transparency. Large organizations and institutions are known to be ineffective in data safeguarding, and since these data are extremely valuable, criminal organizations or foreign governments are often effective in their theft. The analysis of executable choreographies and their implementation in the real systems led us to the conclusion that it is possible to increase data privacy by using a different kind of automation made possible by the personal assistant of the future. A possible approach may be employing software systems integrated on a large scale, while the data control may be made by data owners. As it is very laborious to control this access manually, we argue in this paper that the same may be achieved via personal digital assistants working for the data owners. Step by step, these assistants can become the real representatives of the people and the institutions that have legal access to private data management.

Keywords: smart society, executable choreographies, personal assistants

1. Introduction: Integration and Executable Choreographies

Our experimental and practical concerns regarding the integration between various cloud systems have led us to the observation that there is a tight link between two efforts: building personal assistants and achieving smart systems aligned to the IoT (Internet of Things) trend. By "smart systems", we understand the complex integrated systems including mobile applications, software systems for smart cities, smart communities and other various applications with IoT flavour.

Smart systems integrate technology, organizations and people in order to accomplish complex processes that are controlled by computer systems. From a technical standpoint, the integration perspective is very important for smart systems. For a large number of integration points, integration is achieved through classical ESB (Enterprise Service Bus) - type systems [8], MOM (Message-Oriented Middleware) systems [10], systems based on EIP (Enterprise Integration Patterns) [13] or through the orchestration of services through custom code or languages used to model business processes [16].

All these methods tend to be sufficient to integrate the components belonging to one organization. On the other hand, the integration among multiple organizations should be addressed using choreographies as any centralized solution is risky in terms of security and private data protection. Composition of systems using orchestration tends to create centralized systems.

Although many companies perceive choreographies as a mechanism to describe in a more formal way the contracts among several organizations [34], the academic research proposed the concept of executable choreographies [11], [18], [21], [28]. They suggest transforming the descriptions of the choreographies in code that is executed inside each organization participating in the choreography. As such, choreography is not only a formal description of a contract among organizations but it is also a description of a workflow in an executable way. The same description (choreography) gets to run in several organizations and therefore any need to translate the choreography into other programming languages disappears.

A classification from [28] shows three types of executable choreography that are necessary in ensuring integration and data protection.

Verified Choreographies are executable choreographies accompanied by automated methods of verification in the usage of private data. Using choreography for integration leads to a logic separation between the code that runs in the processing nodes and the code that actually makes the integration. Usage of private data can be observed only by checking the integration code (choreography) and this reduces the effort and the complexity of the verification instruments.

Encrypted choreographies are based on encryption key control, mechanisms of identification and authentication enabling safe choreographies from the perspective of data protection between two or several organizations. The verifiable choreographies aim only to pinpoint the location (organizations) and the type of transferred private data. By contrast, the encrypted choreographies are making available a set of instruments and the self-implementation of choreographies minimizing the risk of sharing private information.

The implementation of encrypted choreographies is based on data storage systems employing specific encryption techniques aiming at achieving practical implementation of partial homomorphic encryption [3], [14], [24], [31]. Furthermore, the implementation of encrypted choreographies enables various methods of encrypted data storage belonging to independent organizations. This method aims to develop encryption protocols anonymize and divide data before storage through choreographies. As such, the risk of discretionary copying of data by an administrator or an attacker controlling one of the nodes (participating organization) is minimized. The encrypted choreographies are also employing communication safety encryption protocols [30], as well as privacy policy modeling systems [1], [4], [15], [17], [19].

Serverless Choreographies are encrypted choreographies adapted to run platforms on public cloud that provides full automation of deployment and monitoring. As there is no need for human intervention, we can increase the possibility of running cloud applications without the access to private data by people with physical or administrative access to servers in the cloud. Basically, the serverless choreographies aim at enabling enterprise applications or mobile apps to use cloud resources almost at the same level of risk as if they were using a private server to which only the user has access to.

We may notice that serverless choreographies have Encrypted Choreographies properties. Also, Encrypted Choreographies have Verified Choreographies properties.

Platforms that allow the execution of choreographies are still in their beginnings, but there is a potential for significant evolution of architectures based on web services from the perspective of security and personal data protection.

2. Interpretation of privacy by design from a technical perspective

The latest trend in approaching privacy revolves around the Privacy By Design principles (PbD)[20] and their legal interpretation (e.g. GDPR - EU General; Data Protection Regulation)[12].

The Privacy by Design principles is the foundation of the modern thinking about privacy issues. Privacy by Design principles are regarded as relatively vague as they do not intrinsically hold practical instructions on how they should be implemented. This issue is raising a certain level of technical difficulty [26].

This paper will further-on employ the technical term of PbD in order to unify the industry dedicated concepts of “Privacy by Design” and “Privacy by Default”.

Privacy by Design (sometimes denominating data safety embedded in the designing phase) implies certain regulations to be incorporated in the software development methodologies when processing private data. Privacy by Default (sometimes denominating data safety enabled by default settings) means that whenever using a product, the consumer must acquire it with parameters set up for maximum protection.

Subsequently, we summarize the seven PbD principles as they are found in the scientific literature. For a detailed approach it is recommended [5], [22], [25]. Our approach here is to present these principles with the perspective of a software architect willing to implement them in real systems and not of a lawyer:

- **Proactive not reactive; Preventative not remedial**
  The first PbD principle states that private data protection is to be performed proactive and not reactive. Obviously, the remedy of data theft or detecting private data copying does not help stopping the alleged harm in illegally using the data. This principle covers both technical prevention means and organizational approaches (policies, standards, safety and privacy oriented organization culture).
- **Privacy as the default setting**
  The modern systems tend to be highly customizable by users and administrators. The second principle states that the default settings for a new user should enable only the system behaviors that protect private data, and not those enabling data leakage. Let be given a social network and its settings. Even if there is a system setting enabling or disabling the visibility of the phone number or e-mail address, the default setting according to this principle should be the one rendering the private data invisible. Because of commercial use aspects, many current internet services are not abiding by this principle. A motive for this potentially harmful behavior consists in the benefits achievable by user behavioral learning and user private data gathering.
  The applied principle might mean the implementation of mechanism enabling the following: specifying the purpose for data gathering, limiting the data gathering to only the specified purpose, gathered or shared Data Minimization, limiting the data storage time, limiting the use of private data to only the specified purpose.
- **Privacy embedded into design**
  The third PbD principle stated that private date safety must be approached and embedded in the initial phases of design, proactively and not reactively. Ideally, private data protection and safety mechanisms should be formally verifiable since the system design phase. However, because of increased complexity and lack of suitable methods, the practice is scarcely applied as we speak. Our research endeavor in the field of verifiable choreographies may be perceived as added value to the industry.
- **Full functionality – positive-sum, not zero-sum**
  The fourth PbD principle aims to not prioritize private interest in detriment of social and group interests. As citizens, we cherish the benefits of communication between various organizations and social players. The progress and material wealth is based on capitalizing on trust and private data. In order to promote socially healthy commercial use and good and services exchange, we need robust systems able to employ private data access according to law. This principle is one of the least understood in the academic community – as it is sometimes the case in industry – because of the approach of solving social issues by technological means. This principle is in need of a broader understanding and of acceptance in the systems architecture of the seemingly contradictory forces that are defining the concept of privacy. Chiefly, this principle rejects the idea of data protection hindering the commercial use of data.
- **Visibility and transparency – keep it open**
  As a result of the previous principle, it is obvious that there cannot be magical boxes perfectly complying with applicable laws and rights. Because of human interference, the software systems will always be vulnerable to their users. However, the employment of audit mechanisms or detailed log-in mechanisms when concerning private data processing and access may significantly diminish the associated risks. This principle states that the visibility and transparency of the system operating or user related processes should be maximized by design. Any improper implementation of this principle may transform transparency in a data safety and protection related problem source.
  Essentially, this principle may be achieved by verifying the following aspects:
  - Assigning responsibility: any private data access should be logged in, and a subsequent audit should be able to verify the legality of the access;

  - Transparency: private data management policies and practices must be acknowledgeable by legitimate concerned parties;
  - Compliance: the organizations that are processing private data are to comply with accurately defined regulations, standards and procedures on data usage.
- **End-to-end security – full lifecycle protection**

  This principle explicitly states that all aspects concerned in using a system may contribute to the compliance or violation of regulations and policies concerning private data. Data protection must be a perpetual concern starting with the assessment, implementation, maintenance and methodology of software systems' design and operating procedures.

  This principle may be intuitively summed up by understanding that security aspects, as well as standards implementation and good practices are required in order to achieve systems able to provide data protection.
- **Respect for user privacy – keep it user-centric**

  In spite of conflict between private interest and group interest, the seventh principle states that when in doubt, the user's private interest should be prioritized.

  Essentially, this principle may be achieved by verifying the following aspects:
  - Acquiring consent: the private data gathering and processing should be performed only after acquiring the user's consent;
  - Accuracy: the private data should be accurate, complete, updated in order not to inflict personal damages;
  - Access: the individual should be able to access his/her own data, and to be able to request these to be deleted;
  - Compliance: the organizations that are processing private data are to comply with accurately defined regulations, standards and procedures on data usage.

Beside the organizational aspects addressed by PbD, a sum of these principles approachable by the implementation parties (software architects and programmers), and not law practices specialists, should cover the following aspects:

- obtaining valid consent
- preserve quality of data
- data minimization (obtain only the required data)
- reaction to breaches
- the right to be forgotten

Given the flexible character of these principles, one should be aware that if the privacy issues are to be left only to the concern of technical specialists and efficiency and profit oriented business decision-makers, their interpretation would be much more relaxed than the optimal social interest requires. This is why in the subsequent section we propose a software implementable principle using executable choreographies, creating a trend for an applied technical approach of these principles.

3. Data self-sovereignty principle and choreographies

In this section, we propose the concept of data self-sovereignty as a PbD implementation principle. We make note of several similar approaches that will facilitate the understanding of the concept by comparison. There are papers introducing the concept of data self-sovereignty [32] as establishing the nation-state where the cloud storage service providers are storing the data physically in order to ensure they are meeting their contractual geographic obligations. By contrast, we consider data sovereignty to be the ability of the user to have full control over his data and the entities to which it is shared or revoked.

To achieve data sovereignty [6] proposes to store encrypted data in cloud federations. We envision a similar approach by using executable choreographies on a federated service bus as presented in [29]

Analyzing risks related to the protection of personal data, we propose the introduction of the data self-sovereignty principle (DSSP). In an ideal world, private data access is granted only to authorized people or legal organizations. In current practices, data is copied very easily and the data owner lose track of those copies. The main idea of DSSP is that any private data is stored and handled in ways that preserve ownership information and any access that the data is directly accessible only with the consent of a user or a legally authorized entity. In this article, we are not intending to present algorithms and technical details, instead we will focus in justifying how DSSP is related with the principles recognized by the community as staying as the foundation of the modern privacy thinking PbD. These principles are translated in actual laws, the most representative these days being the GDPR (**General Data Protection Regulation)**. However all the principles and laws require a separate translation in operational procedures that can be translated into real code. This problem of enforcing PbD in code is recognized as being a difficult one [2]. Any software architectures that follow DSSP could be candidates on implementing PbD in code.

In our DSSP proposal, all the private data records should be imagined as stored long time in a ‘safe box’ for private data. It is possible to copy data for processing in other nodes or for short term caching purposes (in memory) but it should not be stored long term on persistent storage medias. The purpose of using this restriction is to ensure that any access to private data is made only from the ‘safe box’ and each access can be recorded.

In our daily applications, once the data is obtained, it can be stored without notifying the owner, so it is difficult to apply the DSSP principle without legal support. These restrictions cloud also lead to performance issues. While the DSSP principle seems almost obvious, the current technical reality is that our data is copied very easily and the ownership for data is in many cases unclear because of the technical constraints.

The relation between DSSP and the above aspects is presented in the Table 1.

Table 1. DSSP implemented with Executable Choreographies can address technical aspects required by GDPR

Technical aspect	DSSP implemented with Choreographies
obtaining valid consent	By Design, DSSP assumes that each sharing of a private data should be with user’s consent
preserve quality of data	If we keep only a copy of the data, the quality of data is increased because all the systems referencing data from the “safe box” will be updated without any manual intervention.
data minimization	This is partially solved but at least, the increased visibility of places in which data got shared allows insights and a better sovereignty
the right to be forgotten	If the data is under user’s control he can directly delete access to any organization holding a copy
reaction to breaches	Offering data minimization by design, DSSP reduces the risks associated with breaches

This article proposes executable choreographies as a direction in solving the technical difficulties of implementing Privacy by Design, as well as implementing DSSP. The link between Privacy by Design principles and choreographies is presented below (Table 2) in the context of executable choreographies.

Table 2. Executables Choreographies and suggestions for HOW TO implement the Privacy by Design

Privacy by Design Principles	Solutions using Choreographies
1. Proactive not reactive; Preventative not remedial	Any software system based on executable choreographies is formally verifiable. A verification method is to count the number of breaches of the DSSP principle. Other verification mechanisms can accurately outline all the actors that could have access to private data. These verifications prevent privacy breaches by providing an early warning mechanism if privacy issues are mistreated in the planning phase or during implementation.
2. Privacy as the default setting 3. Privacy embedded into design	Using Verified Choreographies, privacy and security concerns are embedded into design. This claim can be accepted because by their nature choreographies extract all the integration layer from the code and the verification effort is reduced.
4. Full functionality – positive-sum, not zero-sum	Executable Choreographies are an executable programming model that does not impede the overall business goals.
5.Visibility and transparency – keep it open	Verifiable choreographies underlying the implementation of all other executable choreographies model in a transparent way how private data is transmitted between classic organizations. Other technologies that are promote the usage of web services / APIs do not allow an overview of how the various organizations or systems communicate with each other.
6. End-to-end security – full lifecycle protection	Encrypted choreographies ensure by design end-to-end security by encrypting communication and by formal code verification of the architecture.
7. Respect for user privacy – keep it user-centric	Executable choreographies can be verified against the number of breaches of the DSSP principle. According to the DSSP principle, any access to private data requires explicit acceptance from the user. This principle leads to software architectures that allow users to be in control of their private data. Therefore, by using executable choreographies, it is possible to formally measure if a software system prioritizes the user`s interests over commercial interests.

4. Towards resolving fallacies within regulations

In [2] there are a few fallacies related with the EU regulation about data privacy. We will discuss them through our approach based on executable choreographies.

The first fallacy in [2] mentions “Too much focus on informational self-determination”. The personal assistants operating on DSSP compliant system would be able to address the “mythology of consent” issue. This is a real problem residing in the human time and attention-span limitations. If the data storage systems hold information on data type and access policies, while the assistant knows the data owner’s preferences, one will be able to imagine intelligent systems able to inference automated decisions or to present intelligible information to the user, without employing dozens of text pages written in legal jargon requiring his or her consent.

The second fallacy in [2] mentions “Too much faith in controller actions”.
Translating principles and regulations into code and into systems design diminishes the risk of arbitrary interpretation. Furthermore, a technical approach as is the one we propose falls into the “ante regulation” category, rather than “ex-post regulation”. An increased focus on accountability and oversight aiming at increased accountability run a risk of leading to more paper rather than more data protection [2].

The third fallacy in [2] mentions “Regulating everything in one statutory law”.
The main concern raised by this fallacy is that many times “Law in the books” does not always become, nor does it always resemble, law in action [2]. The employment and regulation of technical methods such as those proposed by DSSP enable a more direct and applicable implementation, reducing the cases of formal law enforcement. The verifiable choreographies enable the effective assessment of principle breaches and thus it is possible to evaluate more objectively the concept of “data minimization”.
These fallacies can be addressed by the approach proposed in the next chapter.

5. Personal assistants – a potential solution

In recent years, many large companies have begun to accelerate their efforts to build personal assistants. An important indicator to determine the viability of this direction was the special success of the conversational interfaces in Asia through WeChat platform [33]. WeChat has managed to build a user experience different from social networks and chat applications, allowing the user to take advantage of the application almost all the time but being able to solve effectively everyday problems without switching to other applications. Basically, instead of using five or ten applications and websites, a WeChat user uses only one application that integrates all necessary functions. Without using highly advanced artificial intelligence methods, WeChat appears as a prime exponent of a successful personal assistant software category.

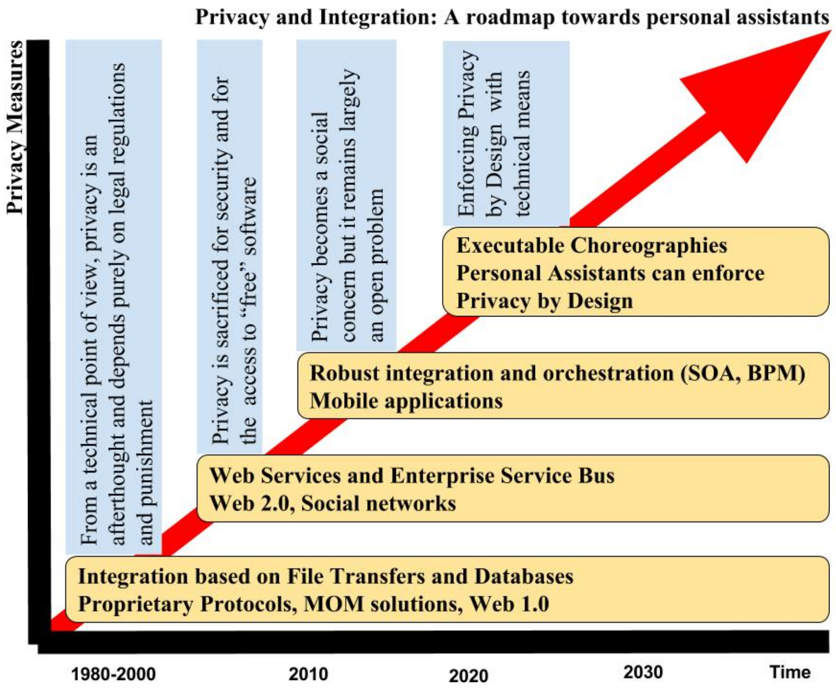
Google, Facebook, Samsung, Microsoft, Apple and other companies have started working on applications [9] that include both artificial intelligence, natural language processing techniques, as well as a style of interaction like WeChat.
Obviously, personal assistants and all applications of this kind, bring up deep issues of privacy. At the moment, “free” business software models still rely on convenience and lack of understanding of personal and social risks due to large-scale collection of personal data.
Major companies seek to comply with certain rules and standards, such as differential privacy [7]. States and international organizations start to gradually introduce principles and standards, the most notable being Privacy By Design. Collecting information in parallel with the absence of technical constraints on how companies can use the data intentionally or unintentionally begins to be perceived as a risk. On the one hand, there are risks for companies because users could refuse to adopt privacy challenged technologies. On the other hand, we have risks regarding the whole society, the most obvious being represented by the potential that some companies can influence society in illegal and immoral manners.

From our discussion point of view, giving up to the standard communication promoted by web technologies and moving towards a model of communication verifiable as the one proposed by executable choreographies, we have the opportunity to have a formal verification of how the data are used by the personal assistants.
Commercial exploitation of private data has come to create the impression that people are exploited commercially in ways that do not adequately compensate for the risks they take. A more transparent model that allows fair and equitable use of personal data is needed. Considering all these aspects, the article proposes that the DSSP principle applied through choreographies can lead to software architecture in which private data’s storage places are under the strict control of the user's personal assistant.

Being a software system, a personal assistant may be able to authorize or refuse access to private data in real time but also to take into consideration in an intelligent manner all the user’s preferences, desires and commercial interests. Therefore, legally authorized access to private data can be performed in a controlled manner and under a stricter social audit.

In Figure 1 we summaries our discussion and we propose a roadmap towards personal assistants through privacy and integration perspective.

Fig 1: An evolution of enforcing privacy from the integration perspective


Privacy and Integration: A roadmap towards personal assistants
Privacy Measures
From a technical point of view, privacy is an afterthought and depends purely on legal regulations and punishment
Privacy is sacrificed for security and for the access to "free" software
Privacy becomes a social concern but it remains largely an open problem
Enforcing Privacy by Design with technical means
Executable Choreographies Personal Assistants can enforce Privacy by Design
Robust integration and orchestration (SOA, BPM) Mobile applications
Web Services and Enterprise Service Bus Web 2.0, Social networks
Integration based on File Transfers and Databases Proprietary Protocols, MOM solutions, Web 1.0
1980-2000
2010
2020
2030
Time

6. Conclusions

This article aims at presenting the executable choreographies role in solving problems related to privacy. In the PrivateSky [23] research project we develop a platform that allows the execution of all three types of choreographies mentioned in this paper. As a way of validating this system we have proposed a personal assistant that uses these executable choreographies. As part of the efforts to implement the PrivateSky platform, a formal specification for creating a system based on DSSP principle is presented in [27].

Our analyses led us to the idea that personal assistants used in the future could have the important responsibility of moderating the 'safe box' containing personal data. In this article we have argued that personal assistants could be very useful to increase the level of personal data protection by automating data granting access, while still keeping the data owners in control. Without changing the ways of storing and sharing private data and without an artificial intelligence capable to work on our behalf, all the regulations are very hard to be respected.

Personal data is valuable economically, therefore only a personal assistant with infinite patience and fast reaction speed could allow accurate exploitation of personal data for various purposes. By using executable choreographies together with pragmatic approximation of DSSP principle, personal assistants could be transformed from a threat to privacy into a crucial ally for each of us.

Acknowledgements

This work is partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P_40_371/13/01.09.2016)

References

1. Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M.: "Enterprise privacy authorization language (EPAL 1.2)". Submission to W3C (2003)
2. B.J. Koops: The trouble with European data protection law, International Data Privacy Law, doi:10.1093/idpl/ipu023, Publisher: IEEE (2014)
3. C. Gentry. "Fully homomorphic encryption using ideal lattices." In Proceedings of the 41st Annual ACM Symposium on Theory of Computing, Bethesda, MD (2009)
4. Cavoukian, A., and Jutla, D.: Privacy Policies Are Not Enough: We Need Software Transparency (2014)
5. Cavoukian, A., Shapiro, S. and Cronk, R. J.: Privacy Engineering: Proactively Embedding Privacy, by Design (2014)
6. Christian Esposito, Aniello Castiglione, Kim-Kwang Raymond Choo: Encryption-Based Solution for Data Sovereignty in Federated Clouds, Published in: IEEE Cloud Computing (Volume: 3, Issue: 1, Jan.-Feb. 2016), Page(s): 12 - 17; INSPEC Accession Number: 15806080, DOI: 10.1109/MCC.2016.18 (2016)
7. C. Dwork: Differential privacy, Encyclopedia of Cryptography and Security, Springer US, pp. 338-340 (2011)
8. D. Chappell: Enterprise service bus, O'Reilly Media, Inc., ISBN 0-596-00675-6, (2004)
9. D. Olanoff, J. Constine: Facebook Is Adding A Personal Assistant Called "M" To Your Messenger App, https://techcrunch.com (2015)
10. E. Curry: Message-oriented middleware, Middleware for communications, ISBN 978-0-470-86206-3 (2004)
11. F. Akkawi, D.P. Fletcher, T. Cottenier, D.P. Duncavage, R.L. Alena, T. Elrad: An executable choreography framework for dynamic service-oriented architectures, IEEE Aerospace Conference (2006)
12. General Data Protection Regulation (GDPR): https://gdpr-info.eu/
13. G. Hohpe, B. Woolf: Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley Professional (2004)
14. Guy Zyskind Oz Nathan Alex 'Sandy' Pentland. Enigma: Decentralized Computation Platform with Guaranteed Privacy (2015)
15. Jean Yang, Kuat Yessenov, and Armando Solar-Lezama: A Language for Automatically Enforcing Privacy Policies,. POPL '12 Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages, Pages 85-96 (2012)
16. Ko, Ryan KL.: A computer scientist's introductory guide to business process management (BPM). Crossroads, 15(4), ACM Press (2009)
17. Kolter, J.P.: User-centric Privacy: A Usable and Provider-independent Privacy Infrastructure, Josef Eul Verlag GmbH (2010)
18. L., Alboaie, S. Alboaie, and Panu, A.: Swarm Communication - A Messaging Pattern Proposal for Dynamic Scalability in Cloud, 15th IEEE International Conference on High Performance Computing and Communications (HPCC 2013). IEEE, pp. 1930 – 1937 (2013)
19. Moritz Y. Becker, M. Y., Malkis, A. and Bussard, L.: A practical generic privacy language, Proceedings of the 6th international conference on Information systems security (ICISS'10). Springer-Verlag, pp. 125-139 (2010)
20. P. Hustinx: Privacy by design: delivering the promises, Identity in the Information Society, Volume 3, Issue 2, pp 253–255 (2010)
21. P. Besana P, A. Barker, An Executable Calculus for Service Choreography, OTM Confederated International Conferences, On the Move to Meaningful Internet Systems, pp. 373-380, Springer Berlin Heidelberg, (2009)
22. Privacy by Design, The 7 Foundational Principles: www.iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf (2011)
23. PrivateSky (P_40_371/13/01.09.2016): https://github.com/PrivateSky/swarmcore

24. Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich: CryptDB: Protecting Confidentiality with Encrypted Query Processing. In Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP), Portugal (2011)
25. Ross McKean, Olswang LLP, EU Data Protection Reform - privacy-by-design http://www.olswang.com (2014)
26. S. Gürses, C. Troncoso, C. Diaz: Engineering privacy by design, Computers, Privacy & Data Protection, https://www.esat.kuleuven.be/cosic/publications/article-1542.pdf (2011)
27. Sînică Alboaie, Doina Cosovan, "Private Data System enabling Self-Sovereign Storage managed by Executable Choregraphies", DAIS 2017 - 17th IFIP International Conference on Distributed Applications and Interoperable Systems
28. Sinica Alboaie, Lenuta Alboaie, Andrei Panu, Levels of Privacy for e-Health systems in the cloud era, 24thInternational Conference on Information Systems Development Harbin, China, August 25-27 (2015)
29. Sinica Alboaie, Lenuta Alboaie, and Mircea-Florin Vaida: Web service transformations in a federated Enterprise Service Bus based on executable choreographies, Proceedings of the Conference on Mathematical Foundations of Informatics MFOI 2016, Chisinau, Republic of Moldova (2016)
30. Tor: Hidden Service Protocol, www.torproject.org/docs/hidden-services.html.en
31. Úlfar Erlingsson, Vasyl Pihur, Aleksandra Korolova: RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response, CCS '14 Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Pages 1054-1067 (2014)
32. Zachary N. J. Peterson, Mark Gondree, and Robert Beverly: A position paper on data sovereignty: the importance of geolocating data in the cloud. In Proceedings of the 3rd USENIX conference on Hot topics in cloud computing (HotCloud'11). USENIX Association, Berkeley, CA, USA, 9-9 (2011)
33. WeChat's world, http://www.economist.com/, The Economist, 2016
34. WSCDL Specification- https://www.w3.org/TR/ws-cdl-10/