Software as a Device in Edge Computing

Radmila Juric
ALMAIS, Consultancy, UK
radjur3@gmail.com

Sang Suh
Texas A&M University, US
Sang/suh@tamuc.edu

Elisabetta Ronchieri
University of Bologna, Italy
elisabetta.ronchieri@unibo.it

Murat Tanik
University of Alabama at Birmingham, US
mtanik@uab.edu

Karoline Moholth McClenaghan
University of South-Eastern Norway
karolinem@usn.no

John Carbone
Forcepoint Global Governments and Critical Infrastructure, US
John.Carbone@forcepointgov.com

Abstract

The paper explores possibilities of defining a generic conceptual model for creating software as a device. This research was triggered by initiatives from the healthcare sector to start decoupling software, defined by device manufacturers, from their hardware and addresses problems in regulating medical devices and their accompanying software. The idea of creating situations where software plays the role of a device, might have long term benefits regarding (a) sustainability of mobile and wireless computing, particularly at the computing edge (b) proliferation of new gadgets/devices coupled with proprietary software which often deliver similar if not identical services and (c) rising heterogeneities of hardware and software which may be difficult to manage. For illustration this paper uses the example of Software as a Medical Device (SaMD). It gives a very clear picture of why it is urgent to change the way devices are manufactured and coupled with proprietary software when delivering services.

1. Introduction

Over the last two decades we have mastered the distribution of computing across the space between edges of computer networks and clouds (Angel et al., 2022). This was feasible due to advances in component-based software technologies and service-oriented software architectures, where software is envisaged as a *service*. In the early 2000s, Software as a Service (SaaS) was firmly established, because of the proliferation of web-services (Chou, 2008), (Chou and Chiang, 2013), and since then, we have introduced software as a service visionology (Badr and Caplat, 2010), defined service science (Larson, 2008) and service oriented software systems (Yan, et al., 2008), introduced software as a service architecture (Maheshwari, et al., 2020), and defined service-based software architectures (Berrio-Charry, et al., 2020).

In the times of post-cloud computing (Zenela, 2023), SaaS keeps a prominent role, triggered by the maturity of edge computing (Varghese et al., 2021), (Törngren et l., 2021), (Douch et al., 2022), attempts to create edge intelligence (Dustdar, 2021) and secure trustworthy and sustainable edge AI (Ding et al., 2021). The position of edge computing, at nodes of computer networks, very often supports the view that Edge-as-a-Service (EaaS) exists as a natural extension of SaaS (Davy et al.,2014), (Corcoran and Datta, 2016), (Ranjan, et l., 2021), (Escaleira et al., 2022), (Oikonomou and Rouskas, 2020). Therefore, it is not surprising that EaaS is a topic of interest in the world of modern software technologies. This is particularly true when applying EaaS in cyber physical systems, such as the internet of things, and utilizing pervasive computing within them (Gomes et al., 2016) (Zhang et al., 2022), (Escaleira et al., 2022), (Guan and Boukerche, 2022).

However, in the world of pervasive computing, where devices are its essential parts, services delivered by software are often perceived as *services delivered by devices*. It is easy to confuse them and difficult to distinguish between the two. SaaS does exist but, considering that we often compute with hand-held and mobile/wireless devices, it is more comfortable to assume that those devices are the ones which deliver services. We must not forget that without dedicated software on these devices, there would be no service delivered and thus we should keep in mind that services come from software which runs on these devices.

The last decade was characterised by a proliferation of new devices, gadgets and computational wearables which deliver services. This trend creates heterogeneities, not unknown in the history of computing. Heterogeneity of hardware, operating systems and software is inevitable. Consequently, software which runs on heterogeneous devices is often generated and owned by device manufacturers to address interoperability problems. In other words, similar or identical services are delivered using heterogeneous devices which either run different software or are equipped with proprietary software, which cannot be decoupled from physical devices.

Wellbeing wearables and watches, sensory equipped gadgets and medical devices in particular, use different

URI: https://hdl.handle.net/10125/107293
978-0-9981331-7-1

software for delivering almost identical services. The inability to decouple software from such devices creates
(a) a lack of transparency of software solutions which deliver services and
(b) dependability on hardware manufacturers which restrict our opportunity to understand and choose software we wish to run.

This has prompted the debate whether software at the computing edge can play the *role of a device*. We are interested in separating software from hardware when manufacturing devices and creating generic software we run on all of them. We can then look at possibilities of software playing the role of a device.

This paper aims to develop a conceptual model for creating software as a device suitable for edge computing. It is placed in the domain of creating Software as a Medical Device (SaMD), but it could be used in any other environment where our dependability on hardware manufacturers and their proprietary software, for delivering services, becomes a problem. We use the domain of medical devices because of numerous initiatives to define and create SaMD. The ideas behind SaMD initiatives clearly explain why we need software which plays the role of a device.

The paper is organized as follows. Section 2 introduces the domain of medical devices where the ongoing initiative of seeing software as a device has taken its momentum. Section 3 describes an environment for creating SaMD. Section 4 proposes a conceptual model and section 5 creates software architectures (SA) for SaMD. Section 5.1. defines SaMD from the software architectures and section 6 debates its implementation. Debates and related work are in section 7 and we conclude in section 8.

2. Software as a medical device (SaMD)

The FDA and International Medical Device Regulators Forum (IMDRF) initiated standards for defining what an SaMD is, to secure transparency when creating devices, which could run SaMD. They also define characteristics of SaMD (FDA, 2018), (IMDRF, 2015) (IMDRF, 2016). The most comprehensive source of information on SaMD is the US FDA. Their idea is to ensure the separation of software from hardware in defining SaMD. However, the FDA is a medicinal regulatory body and does not give any advice for creating software and thus it promotes the vision of running "*the same software upon devices from different manufacturers"*. There are commercial initiatives, as in (Promenade Software, 2022), in which software developers could tailor their products to fit any device or any software operating environments, when creating SaMD. However, this is still far from the FDA vision. Tailoring software to a particular device, after the device has been manufactured, is not the best practice for engineers and software developers.

Benefits, opportunities, and effectiveness of SaMD are numerous (Novelo et al., 2019) The FDA must oversee and review (as the body with enforcement discretion) broad categories of SaMDs, and it is convenient to assume that all SaMD might become Apps. This is a sharp contrast to (Hermon et al., 2021) where the authors argue about the definition and categories of SaMD, and question SaMD boundaries, which is not covered in the FDA vision. Interestingly, the European Union (EU) still sees SaMD as an integral part of medical devices (Granlund, 2020), (Owono, 2014) and focuses on strict control of compliance and conformity of devices in healthcare delivery. Their claim is not in line with the FDA. For the EU, software for medical devices must be developed with the same principle as devices on which they run. However, ***all-software medical devices,*** as part of a device manufacturing process, which meets EU regulatory requirements, do not promote decoupling of software from hardware in the EU initiatives.

Publications which connect SaMD and the computational AI algorithms exist (Benjamens et al., 2020), (Evans et al., 2020), (FDA, 2019). The deployment of AI is inevitable, despite being probably premature in this problem domain. We do not have a consensus on how to develop software which can become SaMD, therefore how could we talk about AI algorithms to become SaMD? The author of (Dafoe, 2018) talks about the future of Medical Device Regulation, Innovation and Protection, and (Ahmad et al., 2021) would regulate AI in Healthcare via responsible AI. There is an example of AI and SaMD in radiology (Pesapane, 2018). It focuses on medical imaging where current AI has substantial success.

To summarise, there are two aspects of SaMD important in its research and development.
(I) Considering that SaMD is software, it should be developed according to software engineering principles and based on conceptual and computational models.
(II) Defining abstractions from the environment where SaMD reside and conceptualizing them into models which create SaMD is essential. It goes without saying that SA and styles (Garland and Show, 1993), (Bass and Kazman, 2012), (Marquez et al., 2023), (Antonio et al., 2022), (Camara et al., 2020), (Juric et al., 2004) would be a natural outcome of this conceptualization.

3. The environment for creating SaMD

Figure 1 illustrates a cyber physical environment where SaMD may cohabit with devices of variable computational power and software applications for healthcare delivery in general, dependent on local

(edge) computing and clouds. These are divided into three vertical lines in Figure 1. The leftmost column in Figure 1 accommodates computing in proximity to patients with computational devices, and close to sensory generated data, as opposed to the right column with computations on the clouds, cloudlets, and fog. The middle column is a healthcare environment where healthcare is delivered by professionals (e.g. hospital).

It is important to note that the vertical lines in Figure 1 do not necessarily adhere to physical spaces. They are rather illustrations of different ***computational environments*** where data is generated, interpreted, processed, computed, analyzed, and used in healthcare delivery. Consequently, Figure 1 does not insist on a particular ***location*** for SaMD. i.e., at this stage we cannot say where the SaMD resides. Figure 1 just shows that in these vertical sections we see cyber physical spaces defined according to their purpose.

It would be reasonable to assume that the left column in Figure 1 is associated with edges of computer networks, containing devices in proximity to sources where data originate. It is also reasonable to assume that this column ***may contain SaMD***. However, it is too early to determine exactly where the SaMD will physically reside before we define its conceptual model(s) and debate its implementation. It would be appropriate to use the term "localized computing" for the left most column, because the data is

(a) generated locally, through sensory technology and patients' inputs,

(b) collected, interpreted and computed using local computational devices in proximity to patients,

(c) moved to a hospital or GP location (middle column in Figure 1) where the additional analysis and processing of data is available for healthcare professionals, in their clinical decision making.

The dependency on cloud computing in the right-most column of Figure 1 is unavoidable, considering that we might retrieve archives, store important data persistently and have cloud sources of data as an additional input in medical decision making. Updates or data retrievals from archived information or persistent data stores on clouds, enhance medical decision making and thus healthcare professionals should have access to cloud computing technologies.

The middle column represents an environment for running software applications relevant for clinical decision making. These software applications should be able to use sources of localized and cloud data, results of running local and cloud computations and perform its own computations whenever needed.

Figure 1 does not favor any column, as being central for identifying and running SaMD. It will depend on the purpose of running SaMD, which column would be the central environment for that SaMD computation.

As mentioned earlier, SaMD is very likely to be present in the left and never in the right most column, but mobile computational devices are not confined to any location and therefore Figure 1 is just an illustration of a potential operating environment in healthcare which delivers healthcare services.

From modern computing perspectives Figure 1 shows the feasibility of dispersing computing from the edges of computer networks (running local computing) to the clouds, and the possibility of opening doors for technologies involved in modern distributed computing, such as fog and cloudlets, plus microservices.

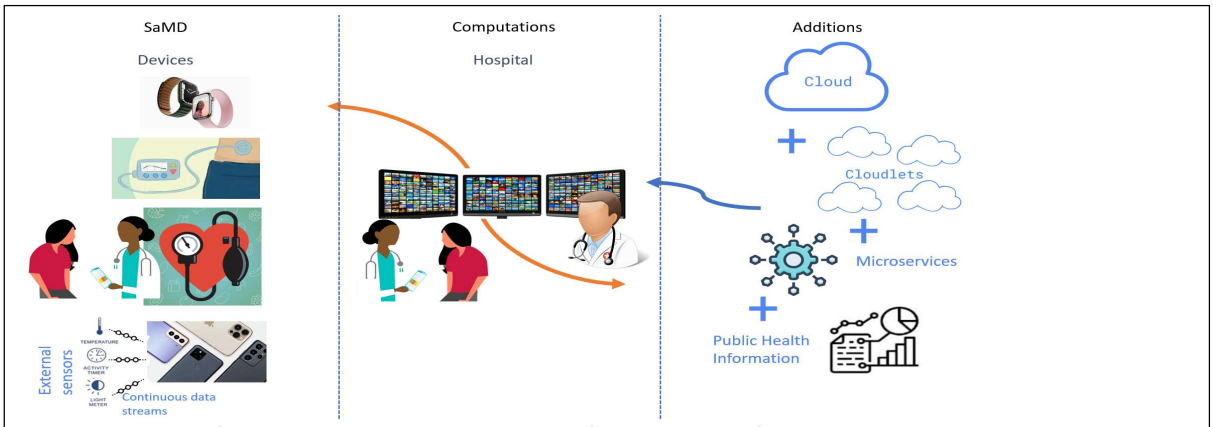

Figure 1. The Cyber Physical Environment with Computational Spaces

However, if this paper focuses on software as a device, using SaMD as an example, then it must follow software engineering practices for its development. They start with conceptualizations, based on software abstractions extracted from the problem domain. This includes outlining not only the functionality of SaMD, but also the deployment of these abstractions in real life, with software technologies. Therefore, the pathway towards a conceptual model for SaMD is to extract software concepts from Figure 1, create a conceptual model and implement it.

4. Towards Conceptual Model for SaMD

In cases of creating data intensive software applications, we narrow the process of finding their abstractions by identifying which data an SaMD would need, which computing will be performed as a part of SaMD functionality and where the results of computing will be stored.

Figure 2 is the first step towards the conceptualization of Figure 1. It keeps vertical columns from Figure 1 for one important reason. There is a need

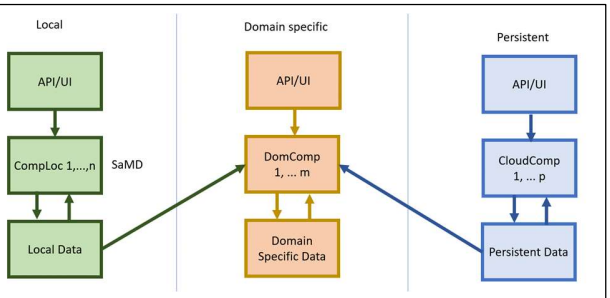

Fig. 2 Conceptual Model for Hospital Environment

to distinguish between localized computing, which is at the edge of computer networks and the clouds. It is also valuable to use the computational space between them (Juric et al., 2023). This would mean that we allow the allocation of "*where computing will take place*" for a later stage in software development, but keep our options of separating these computations in vertical columns open, at this stage.

The columns in Figure 2 are named according to the role data have in their computing. Obviously, clouds (blue) deal with persistent data and localized computing (green) use data generated locally. The domain specific computing (amber) should be able to access any type of data to deliver healthcare services and enable clinical decision making.

Figure 2 shows that it is possible to have local computing software components $ComLoc_{1,..,m}$ in separate physical locations to $DomCom_{1,..,n}$ and $CloudCom_{1,..,p}$ and vice versa - if required by the problem domain. However, the same computational software components could be executed together, within any of these operating environments, if they have access to the data they depend on.

Important note: Figure 2, shows only one instance of the conceptualization. The arrows which show connections between data and computing software components illustrate only what is happening in a hospital (amber). Patients are being remotely monitored through specific medical devices and clinical decisions are made per each patient, according to the

(i) data collected locally from either patients, sensors or medical devices,

(ii) persistent data available in clouds or servers (healthcare specific and medical knowledge)

(iii) data generated within hospital environments, relevant to the patient.

The bullets above (i)-(iii) resemble bullets (a)-(c) in Section 3.

There is one important outcome from Figure 2. Its conceptualization shows that we are using component-based software, i.e. we generate layered and component based software architectures (Bass and Kazman, 2012) and the architectural style which indicates that

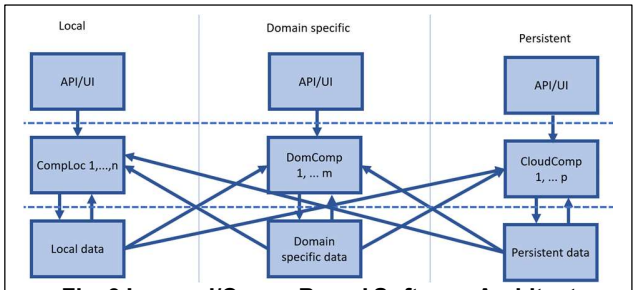

Fig. 3 Layered/Comp. Based Software Architecture

a) The middle layer contains software components responsible for computations and the bottom layer stores data repositories. There is no limit of the number of these components. The level of persistence of data repositories is dictated by the problem domain.

b) Data sharing is encouraged, which means that components from the bottom row of Figure 2 are reusable in any of the computations from the middle layer.

c) Computations defined in vertical columns in Figure 2 can carry on without any interference from or dependence on any other computation in the conceptual model. Therefore, computations defined in columns are self-sufficient if there is data available for their computing. Communications between computations are based on data sharing.

If we remind ourselves that SaMD is a cyber aspect of this cyberphysical space, then we have the freedom of deciding in which column of Figure 2 the SaMD would reside. It is natural to expect that SaMD resides very close to physical medical devices, i.e., in the left column of Figure 2, but modern software technologies allow plug-in and plug-out software components whenever required. The conceptualizations and abstractions create conceptual models which enable deciding which software technologies would be suitable for the deployment of models using the following steps: (I) Specifying roles SaMD would have within the environment, (II) Defining which characteristics as a set of software component SaMD may have and (III) Outlining steps in SaMD development and its

programming with software technologies suitable for the models' deployment.

There is no short cut to the steps above, if we really wish to achieve what has been promoted through the definition of SaMD, address the problems we outlined in the Introduction and exploit the semantic of these environments as illustrated in Figure 1.

5. Software Architecture for Identifying SaMD

Figure 3 summarizes all computations identified in Figure 1 and uses abstractions from Figure 2 to define a generic SA model which can house SaMD. In other words, if we wished to claim that we promote component based and layered software architectures, this means that Figure 3 is our final conceptual model which can be exploited according to the requirements of the problem domain.

The model from Figure 3 still does not specify where a particular SaMD is, but it opens doors to identifying the environment where it could be located and which computation and data it may use.

All three operating environments identified as columns in Figure 1 are shown in Figure 3. Specificities of computations and data sharing which exist among them are clear and it would not be difficult to deploy the architecture with software technologies and implement required software applications for each of these environments (Juric et al., 2021), (Juric and Kim, 2017), (Tarabi and Juric, 2018).

5.1. Defining SaMD

Layered and component-based SA from Figure 3 contains software components which can constitute an SaMD. The choice of software component which will be a part of an SaMD, will always depend on the problem domain. Therefore, as soon as we wish to identify SaMD from the SA in Figure 3, we stop being generic and embark on the specificity of SaMD and its implementation.

We have already mentioned that it is very likely that the SaMD would be identified in the left most column of Figure 1 because of the proximity of computations to the source of data generated by patients and sensors.

Figure4 shows our choice of defining SaMD (pink shading). It is not the only possibility for defining SaMD: many combinations of computations and data from Figure 3 can be a candidate for SaMD. According to Figure 3, SaMD software components take advantage of having access to available data repositories if the problem domain requirements give provision for it.

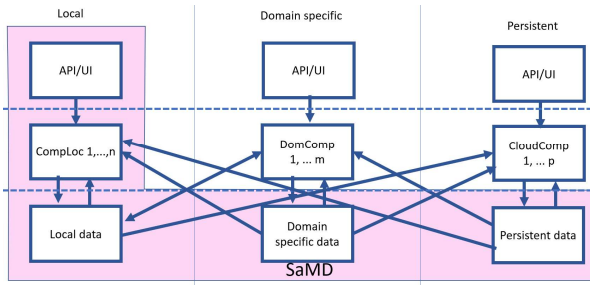

Fig. 4 Identifying SaMD in Software Architecture

SaMD will have its own API/GUI which allows communication with any software (including the software which manages physical devices) and help in the delivery of software service through this SaMD.

Computations for SaMD are likely to happen locally, at the computational edge, and in the proximity of e.g., physical devices. They in turn would allow for collecting, interpreting and storing sensor generated data. These physical devices might be wearable sensors, handheld devices, watches, diabetic pumps, to mention just a few. Being potentially installed on mobile devices, they may accompany patients to any location where the patient happens to be. Also, SaMD may use data available in hospitals and on clouds if the problem domain either requires or allows, according to the definition of the SA style.

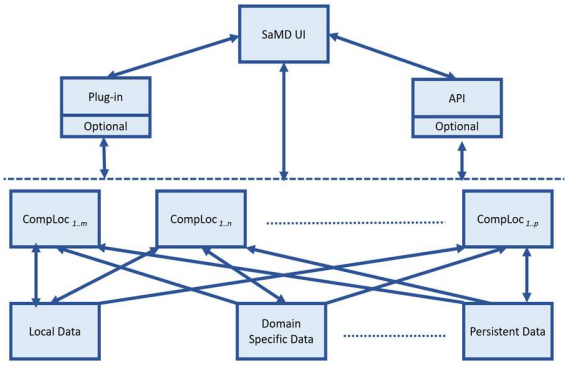

Figure 5 Towards implementation of SaMD

6. Towards the implementation of SaMD

Figure 5 is a software architectural model which could be deployed and thus enable the implementation of SaMD. Its purpose is to focus on the creation of SaMD and therefore when creating Figure 5 we focus only on the left part of Figures 2, 3 and 4. Software components in Figure 5 may become domain specific.

A set of computations $CompLoc_1$, $CompLoc_2$ ... $CompLoc_m$ deliver their functionalities "as a service", but they also include APIs, plug-ins, access control software or any other software which addresses heterogeneities of data and data structures in the problem domain. In an ideal world they could be in the format of tested and reusable software libraries. Heterogeneity is a desirable outcome of modern computing and ranges from platform, operating systems, and hardware to semantic data heterogeneities. In this case we must ensure that any type of operating systems or data structures in various repositories are not an obstacle to implementing SaMD.

Fig. 6: Free Lifestyle Libre 3 and Fitbit Watch

A set of $\{CompLoc_i\}$components can be either prepared in advance or created ad-hoc. It can

(a) give all possible solutions for fulfilling functionalities when delivering a service to a patient and

(b) be chosen according to circumstances in which we run SaMD.

It is important to note that each *CompLoc* may update data in *Local Data* repositories but can have read-only access to data available on clouds, cloudlets and hospital domain data.

In the implementation of SaMD from Figure 5 we had two options. The first option was to create SaMD from scratch, without any device in mind and use commercial sensors and user input to generate data. This is a rather trivial task. We can deploy the model from Figure 5 swiftly using the Android Development environment.

The second option is more interesting. We wanted to de-couple software from two devices of our choice and use it when creating an application from them, which can become SaMD, and thus deliver services of our choice on any device. We do not need to merge these two types of software, running on separate devices into one, because we can manage their functionalities separately through the conceptual model in Figure 5. They have their space in the middle layer within $\{ComLoc_{1,..,m}\}$.

6.1. Two Experiments

The choice of devices used in the experiments was impacted by our long-term interest in computations within wellbeing wearables and their dependencies on cloud computing (McClenaghan and Moholth, 2019a), (McClenaghan and Moholth, 2019b). We have examined numerous wearables for wellbeing (mostly watches) and 20+ constant glucose monitoring devices. There were no significant differences between them in terms of functionality computed by these devices. Figure 6 shows our choice: the FitBit Sense 2 watch, as a wrist wearable device, and constant glucose monitoring device Freestyle Libre 3. The contents of UIs from these two devices are indications of the simplicity of software functionalities running on the devices.

6.1.1. Deploying SaMD

In this experiment we created SaMD from scratch, as indicated in option 1, based on the model from Figure 5. Figure 7 is an implementation specific conceptual model, derived from Fig 5 which can be deployed in an Android operating environment.

We expected that the SaMD secures constant glucose monitoring plus controlling wellbeing /physiological functions, popular in wearable watches. Using commercial sensors and user inputs, plus enabling the sharing and analysis of collected data would be feasible without any hardware specifications for computations. There is no need to have special and separate device(s) for monitoring glucose levels, checking heartbeat, monitoring breathing and walking. The new SaMD from Figure 7 will deliver services, on a device of our choice, using a computer program of our choice (SaMD from Figure 7).

It is also feasible to add extra functionality in this SaMD which can co-relate glucose data with heartbeat, breathing and physical activity data. One of the important factors which may affect glucose levels in our body is physical exercise and this would be an ideal moment to relate and analyze all data together.

Figure 7 illustrates what could constitute $\{ComLoc_{1,..,m}\}$ from Figure 5. Managing sensor generated data and user input and generating outcome of wellness and glucose tracking give a context in which we run an SaMD. We could have enriched the context by consulting PHO and public data such as GPS. However, considering that an SaMD runs in mobile and wireless environments, the location where we run the SaMD is not essential in defining its context. It is user's intentions and body sensor generated data which are crucial in finding the semantic for running the SaMD.

We would like to think that performing analytics upon the collected data and archiving results of computations generated by SaMD would be optional to and personal decisions of users of SaMD. This is the moment where cloud computing could kick-in and in case of performing analytics, we may have the luxury of using the computational space between the edge and cloud computing (fog and cloudlets) if our chosen device has some hardware limitations. Finally, data sharing is essential and adds to the effectiveness of running SaMD software and its sustainability. Obviously, the level of persistence could be determined by a user who wishes to run the SaMD. This means that the amount of persistently stored data, reports and archives is also flexible and personal decisions by users.

Figure 7 is still very close to Figure 1 and biased in terms of defining that SaMD is a core of edge computing in this scenario. However, it is deployable using any Android development environment for many reasons (Bagheri et al.,2016) (Android Developer). The layering of software components and separation of concerns defined in Figure 5 and 7 fit Android Apps Architecture. Functionalities of software components at the edge of computer networks are not computationally demanding and accessing persistent data in public domains is quite simple. Sensory technologies are advancing rapidly and possibilities of wearing and choosing them are real (Phillip et al., 2022). This SaMD can run on any hand-held device with reasonable computational power and UI.

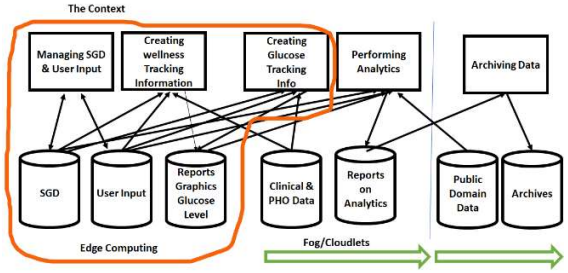

Fig. 7 Deploying SaMD in an Android environment.

6.1.2. Creating SaMD from Existing Software

In this experiment we wanted to re-use existing software found in FitBit and FreeLifestyle 3, create SaMD from both of them and use any device for running SaMD with either bult-in or wearable sensors.

However familiar one has been with these devices from Figure 6 and the software which runs on them, there were two expected problems when we wanted to re-use existing software available in FitBit and FreeLifestyle. First, their software is not available for inspection, customization and downloading to create SaMD. Second, it was impossible to determine where cloud computing kicks in when running these individual Apps. We could not easily judge the level of deployment of edge computing in the Apps connected to these two devices, which affects the way we design SaMD.

However, Figure 8 shows the possibility of reusing software from FitBit and FreeLifestyle 3, if we managed to decouple them from their devices. The only difference is the replacement of our own computational components with FitBit and FreeLifestyle 3 code (red boxes). If optional plug-ins are not available, we could create them or initiate their creation on forums. If the "red boxes" happened to be publicly available software libraries, then we could pick and choose them when personalizing SaMD through its context.

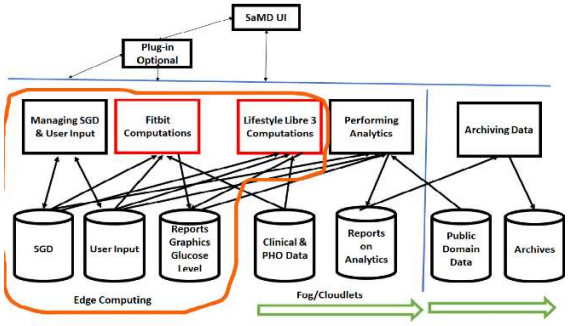

Figure 8 Deploying reusable software for SaMD

7. The Debate and Related Work

This paper is written by computer scientists, having software engineering principles in mind. If an SaMD is SOFTWARE then it should be developed according to the principles of software development. This statement has two implications.

We need software abstractions, conceptualizations, and SA, which can guarantee implementations. Therefore Figure 1,2,3,4 and 5, plus Figure 7 and 8 must be created in the process of software development and cannot be traded for anything else.

Creating conceptual models is a sine qua non in software development, which implies that we model abstractions (interfaces, data and computing) and leave current physical items (from the cyber physical space in Figure 1) outside the model. The FitBit physical watch and Free Lifestyle Libre 3 device – if they are part of a cyber physical space - cannot be a part of a conceptual model.

The second implication refers to the physical location(s) of each physical device and software components from Figure 5. They are all almost immaterial. Because of advances in mobile, wireless, and component-based software technologies, there is no need to specify locations of the software components in Figure 5.

Finally, the proposed model in Figure 5 and its deployment in Figures 7 and 8 are created with NO specific device in mind, and with no specification of the operating system. Figures 7 and 8 can be Android Apps because the model form Figure 5 is deployable in a context relevant for the implementation. The SaMD should run in any Windows, Android or iOS environments which could enable sensor data collection and manipulation.

There are so many options which would enable computations from Figure 5 and their implementation from Figures 7 and 8 to run on handheld devices. This can range from our mobile phones and laptops to medical devices, wearables, soft robots, cyborgs, to mention just a few.

Figures 7 and 8 are illustrations of options in the experiments. They contribute towards software reusability, conceptualization of SaMD and their personalization. However, the realization of these ideas might impact the industry of device manufacturing which has bult its existence through specific physical devices and proprietary software which runs on them. Will the ideas from this research bankrupt the industry of computational wearables?

7.1. Related work

It is very difficult to find publications which challenge this paper. There are no publications of EaaS which could generate software as device. In the domain of healthcare delivery there is also a shortage of publications. The main reason is the lack of communication between device manufacturers, software developers and healthcare professionals (in cases of developing SaMD) when creating software for edge computing. Business models of numerous companies which manufacture devices and equip them with software, suitable for the edge of computer networks, do not make any provision for decoupling software from hardware. They thrive with their proprietary software and do not see any reason for changing the situation.

However, the world of SaMD is rather active. These issues have been debated in one publication from 2009 (Ray et al., 2009) and the impact of the EU Medical Device Directive on Medical Device Software has been published (Owone, 2015). The University of California in San Francisco gives a very good starting point towards the SaMD development in (Varabelli, 2021). Their motto was that “developing software for medical devices is vastly different to developing other types of software”. They differentiate between software *in* medical devices and software *as a* medical device, insist that this distinction is important and require that the standards that govern and manage off-the-shelf software, which may become SaMD, are essential. This was in 2015.

There are a few publications which address a very relevant issue of creating and running proprietary software for managing chronic diseases and addressing wellbeing, using either special devices or web applications (McClenaghan and Moholth 2019 a,b), (Gunleiksrud Jensen et al., 2019), (Tarabi and Juric, 2018). They are all rather different and applied to different problem domains: from addressing diabetes by using diabetic pumps and managing possible reversibility of diabetes 2 to detecting anomalies of biophysical changes in our body which can help in wellbeing and detect possible health problems. All these examples have something in common: conceptualisation of software which runs on various devices, is used for creating SA. The SA is deployable using modern software technologies and as such creates a particular software application. However, these examples are all problem specific, i.e., their SA is not generic. By looking at their software architectural solutions, based on conceptualisation, readers may be convinced that this is the way of developing software which will not be dependent on design decisions of hardware manufacturers.

8. Conclusions

One of the outcomes of this work, favored by the authors, is to re-think the excessive manufacturing of devices which populate the edges of computer networks, used for delivering services to humans or environments where they happen to be. The proliferation of hardware coupled with relatively simple software, to deliver a service, is not sustainable anymore. At the time of advances in sensory technologies, existence of a data source continuum and appearance of continuous software engineering, enriched with intelligent edge-based service provisioning, there is no need to manufacture more hardware for delivering services through software. Decoupling software from devices is essential if we wish to empower edge computing and accept that software can play the role of a device.

The proposed conceptual model for creating software as a device in Figure 3 is simple, generic, and proved to be deployable using today’s software technologies. We do not need a special device to run it. By harvesting a selection of portable sensors, and using their data, utilizing data entries through human intervention, securing access to public or private persistent repository, and choosing existing or creating new computations, stored either in software libraries or public forums, we will enable the creation of the software as a device, ***without adding new hardware.*** It remains to be seen if these ideas can remove barriers

between device manufacturers and software developers involved in edge-based service provisioning or edge-as-a-service in general.

Not allowing de-coupling of software and hardware is not a way forward. We must wait and see if the convenience of beautifully designed computational wearables and their interfaces outweighs the user power in a) personalizing software as a device and b) taking control of accessing and storing data generated within the environments where software as a device operates.

References

Ahmad, M., Overman, S., Allen, C., Kumar, V., Teredesai, A., & Eckert, C. (2021). Software as a Medical Device: Regulating AI in Healthcare via Responsible AI. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Singapore, 4023–4024

Android Developer, Guide to App Architetcures available at https://developer.android.com/topic/architecture

Angel, N. A., Ravindran, D., Vincent, P. M., Srinivasan, K., & Hu, Y.-C. (2022). Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies. Sensors, 22(1). 196; https://doi.org/10.3390/s22010196

Antonino, PO, Capilla, R, Kazman, R., Kuhn, T., Schnicke, F., Treichel, T., Bachorek, A., Müller-Zhang, Z., Salamanca, V. (2022). Continuous Engineering for Industry 4.0 Architectures and Systems. Softw Pract Exper. 2022; 52(10). 2241–2262.

Badr, Y., Caplat, G. (2010) Software-as-a-Service and Versionology: Towards Innovative Service Differentiation, Proceedings of *24th IEEE International Conference on Advanced Information Networking and Applications,* Australia, 237-243.

Bagheri, H., Garcia, J., Sadeghi, A., Malek, S. Medvidovic, N (2016). Sftware architectural principles in contemporary mobile software: from conception to practice, Journal of Systems and Software, 119 (2016), 31-44.

Bass l.., C. P., & Kazman R. (2012). Software Architecture in Practice 3rd Edition. *Addison-Wesley Professional*.

Benjamens, S., Dhunnoo, P., & Meskó, B. (2020). The state of artificial intelligence-based FDA-approved medical devices and algorithms: an online database. NPJ Digital Medicine, 3(1). DOI:10.1038/s41746-020-00324-0

Berrio-Charry, E., Vergara-Vargas J., Umaña-Acosta, H. (2020). A Component-Based Evolution Model for Service-Based Software Architectures. *Proceedings of IEEE 11th International Conference on Software Engineering and Service Sc (ICSE.SS)*, China, 111-115.

Chou, S. (2008) "Web Services: Software-as-a-Service (SaaS), Communication, and Beyond. *Proceedings of 2008 IEEE Congress on Services Part II (services-2 2008),* China, 1-1.

Chou, S., & Chiang, C. (2013). Understanding the formation of software-as-a-service (SaaS) satisfaction from the perspective of service quality. Decis. Support Syst., *56*, 148-155.

Cámara, J., Schmerl B. and Garlan, D. (2020). Software Architecture and Task Plan Co-Adaptation for Mobile Service Robots, *Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Republic of Korea, 125-136.

Corcoran, P., Datta S. K. (2016). Mobile-Edge Computing and the Internet of Things for Consumers: Extending cloud computing and services to the edge of the network. IEEE Consumer Electronics Magazine, 5(4), Oct. 73-74.

Davy, S. Famaey, J., Serrat, J., Gorrichio, J.L., Miron, A., Dramitinos, M., Neves, P.M., Latre, S., Goshen, E. (2014) Challenges to support edge-as-a-service. IEEE Communications Magazine, 52(10). January. 132-139.

Ding, A.Y., Janssen, M., Crowcroft, J. (2021). Trustworthy and Sustainable Edge AI: A Research Agenda, *Proceedings of the* 3rd *IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, US. 164-172.

Douch, M. R., Abid, K., Zine-Dine, D., Bouzidi, Benhaddou, D. (2022). Edge Computing Technology Enablers: A Systematic Lecture Study. IEEE Access, 10, 69264-69302.

Dustdar, S. (2020). Edge Intelligence - The Co-evolution of Humans, IoT, and AI. In G. Wills, P. Kacsuk, & V. Chang (Eds.), *Proceedings of the 5th Int. Conference on Internet of Things, Big Data and Security, IoTBDS, p.* 5. SciTePress. http://hdl.handle.net/20.500.12708/58169

Escaleira, P., Mota, M., Gomes, D., Barraca J. P., Aguiar, R. L. (2022). Multi-access Edge Computing as a Service. *Proceedings of 18th International Conf. on Network and Service Management (CNSM)*, Greece,177-183.

Evans, B. J. a. P., Frank A. (2020). Product Liability Suits for FDA-Regulated AI/ML Software. The Future of Medical Device Regulation: Innovation and Protection (Cohen, I.G. Minssen, T., Nicholson M., Robertson C. D., Shachar C. eds.). *Cambridge University Press*.

FDA (2018) Software as a Medical Device (SaMD). FDA US Food and Drug Administration, available at https://www.fda.gov/medical-devices/digital-health-center-excellence/software-medical-device-samd

Garlan D. and Shaw M. (1993) An Introduction to Software Architecture January 1994 CMU-CS-94-166 School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213-3890 and in Advances in Software Engineering and Knowledge Engineering, Volume I, (V.Ambriola and G.Tortora, eds.) *World Scientific Publishing Company, New Jersey,* 1993.

Gomes, R. L., Bittencourt, L.F., Madeira, E. R. M., Cerqueira E. C., Gerla, M. (2016). Software-Defined Management of Edge as a Service Networks, IEEE Transactions on Network and Service Management, 13(2) 226-239.

Gunleiksrud Jensen, M., Juric,R., Moholth McClenaghan, K., Blagojevic Zagorac, G. (2019). Towards an Artificial Pancreas: Software Architectural Model and Implementation for Personalized Insulin Administration. *In Proceedings of the 52nd Hawaii International Conference on System Sciences, HICSS 52.* US.

Global Approach to Software as a Medical Device. (2022). FDA US Food and Drug Administration.

Granlund, T., Mikkonen, T., Stirbu, V. (2020). On Medical Device Software, CE Compliance and Conformity Assessment. *Proceedings of EEE Int. Conf. on Software Architecture Companion (ICSA-C)*, Brazil, 185-191.

Guan S. and A. Boukerche, "Intelligent Edge-Based Service Provisioning Using Smart Cloudlets, Fog and Mobile Edges. (2022). IEEE Network, 36(2), 139-145.

Hermon, R., Williams, P., & McCauley, V. (2021). Software as a Medical Device (SaMD): Useful or Useless Term? *Proceedings of Hawaii International Conference on System Sciences, HICSS 54,* US, https://doi.org/10.24251/HICSS.2021.451

IMDRF (2015) Software as a Medical Device (SaMD): Application of Quality Management System. (2015). International Medical Device Regulators Forum (IMDRF).

IMDRF (2016) Software as a Medical Device (SaMD): Clinical Evaluation. International Medical Device Regulators Forum (IMDRF). https://www.imdrf.org/sites/default/files/docs/imdrf/final/technical/imdrf-tech-170921-samd-n41-clinical-evaluation_1.pdf

Juric, R., Fuh, C. C., & Kim, I. (2021). Software Architectures and Efficient Data Sharing for Promoting Continuous Drug Re-purposing, *In Proceedings of the 54th Hawaii International Conference on System Sciences, HICSS 54.*

Juric, R., & Kim, I. (2017). Software Architectures for Smart Applications Which Merge Ontological Reasoning With Big Data Analytics. *Proceedings of the 2017 SDPS Conference,* University of Alabama, US. 75-80.

Juric, R., Kuljis, J., & Paul, R. (2004). A software architecture to support interoperability in multiple database systems. *Proceedings of the IASTED Conference on Software Engineering*, Austria, 71-77.

Juric, R., Ronchieri, E., Suh, S. (2023). Creating Intelligent Computational Edge through Semantic Mediations. *Proceedings of the 56th Hawaii International Conference on System Sciences, HICSS 56* US.

Larson, R. C. "Service science: At the intersection of management, social, and engineering sciences (2008) IBM Systems Journal,47(1), 41-51..

Maheshwari, R., Toshniwal, A. Dubey, A. (2020). Software As a Service Architecture and its Security Issues: A Review. *Poroceedigns of 4th International Conference on Inventive Systems and Control (ICISC)*, India, 766-770.

Márquez, G., Astudillo, H., Kazman, R. (2023) Architectural tactics in software architecture: A systematic mapping study, Journal of Systems and Software, 197.

Novelo, M., Weimer, N. G. (2019). Digital Health: Software as a Medical Device. Academic Entrepreneurship for Medical and Health Scientists, 1(3).

McClenaghan, K. M., & Moholth, O. C. (2019a). Computational Model for Wearable Hardware Commodities. *Proceedings of2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, (DASC/PiCom/CBDCom/CyberSciTech)*, Japan, 259-265.

McClenaghan, K. M., & Moholth, O. C. (2019b). Edge Computing for New Generation of Wearables, *Proceedins of the 2019 SDPS Conference of Society for Design and Process Science*, Taiwan, 79-86.

Oikonomou, E. Rouskas,A. (2020). Selection of Service Nodes in Edge Computing Environments. *Proceedngs of 7th Int. Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, France, 1-6.

Owono, F. G.(2015). Impact of EU Medical Device Directive on Medical Device Software. PhD distertation https://scholarworks.waldenu.edu/dissertations/353/

Pesapane, F., Volonté, C., Codari, M., Sardanelli, F. (2018). Artificial intelligence as a medical device in radiology: ethical and regulatory issues in Europe and the United States. Insights into Imaging, 9(5), 745-753.

Philip BJ, Abdelrazek M, Bonti A,Barnett S,Grundy J (2022) Data Collection Mechanisms in Health and Wellness Apps: Review and Analysis, JMIR Mhealth Uhealth 2022;10(3):e30468

Promenade Software (2022). Software as a medical device (SaMD). Promenade Software. https://www.promenadesoftware.com/mobile-apps-and-samd-for-medical-devices

Ranjan, F., Guim, M. Chincholkar, P. Ramchandran, R. Mishra R. Ranganath, S. (2021). Convergence of Edge Services & Edge Infrastructure. *Proceedings of IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Greece, 96-99.

Ray, A., Jetley, R., & Jones, P. (2009). Engineering high confidence medical device software. ACM SIGBED Review, 6.

Tarabi, M., & Juric, R. (2018). Software Architectures for Smart Applications in the Management of Chronic Diseases: A Study of Reversibility of Diabetes 2 *In Proceedings of the 51st Hawaii International Conference on System Sciences, HICSS 51.* US.

Törngren, M., Thompson, H., Herzog, E., Inam, R., Gross, J., Dán, G. (2021) Industrial Edge-based Cyber-Physical Systems - Application Needs and Concerns for Realization. *Proceedings of the IEEE/ACM Symposium on Edge Computing (SEC)*, US. 409-415.

Varabelli, 2019 Introduction to Software for Medical Devices in Early R&D Phases Talk at the University of California, San Francisco https://rosenmaninstitute.org/events/introduction-to-software-for-medical-devices-in-early-rd-phases/

Varghese B. (2021). Revisiting the Arguments for Edge Computing Research. *IEEE Internet Computing*, vol. 25, no. 5, pp. 36-42, 1 Sept.-Oct. 2021.

Yan, Y., Bode J., McIver, W. Jr. (2008) Between Service Science and Service-Oriented Software Systems, *Proceedings of IEEE Congress on Services Part II (services-2 2008)*, China. 189-195.

Zanella, M. (2023) Post-cloud Computing: Addressing Resource Management in the Resource Continuum. Special Topics in Information Technology, 105-115. https://doi.org/10.1007/978-3-031-15374-7_9.

Zhang, M. Cao, J., Sahni, Y., Chen,Q., Jiang S., Wu, T. (2022) "EaaS: A Service-Oriented Edge Computing Framework Towards Distributed Intelligence, *Proceedings of IEEE Int. Conf. on Service-Oriented System Engineering (SOSE)*, US. 165-175.