Towards a new paradigm for managing computing continuum applications

1st Víctor Casamayor Pujol
Distributed Systems Group
TU Wien, Vienna, Austria
v.casamayor@dsg.tuwien.ac.at

2nd Philipp Raith
Distributed Systems Group
TU Wien, Vienna, Austria
p.raith@dsg.tuwien.ac.at

3rd Schahram Dustdar
Distributed Systems Group
TU Wien, Vienna, Austria
dustdar@dsg.tuwien.ac.at

2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI) | 978-1-6654-1621-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/COGMI52975.2021.00032

***Abstract*—"Computing continuum" systems are expected to sculpt our future society through a manifold of applications, most of them leveraging artificial intelligence systems. However, their inherent complexity, brought by their dependency on their underlying infrastructure, requires to propose new methodologies for managing them. The methodologies that are now in place are founded on the first Internet systems and they are not able to cope with the complexity of these systems.**
In this article we first develop the idea of the complexity on these "computing continuum" systems that have Edge Intelligence applications running on them. Then we present our vision on how to represent and manage these emerging systems. Based on a use case, we highlight the system's inherent complexity and we finally sketch how our vision would work on the showcased system.

***Keywords*-Distributed Systems, Computing continuum, Edge Intelligence, Markov Blanket**

I. Introduction

A new type of Internet computing systems are recently emerging, these are known to require concurrent execution on multiple computing tiers. These systems are referred as "computing continuum" systems and they are executed simultaneously on the Internet of Things (IoT), Edge, Fog and Cloud tiers [1], [2].

These systems enable applications that will sculpt our future societies as they allow the development of applications related with all the "smart-systems", such as Smart-cities or Smart-industry, and also healthcare applications to control future possible pandemics or they will enable the use of autonomous vehicles.

"Computing continuum" systems take the best from each tier and avoid their limitations by developing an ad-hoc architecture [3]. Simply put, they can use the Cloud infrastructure to perform heavy computations that do not have real-time constraints, the Edge or Fog infrastructures for those services were response latency needs to be at a minimum, and use the IoT devices directly for AI inference to keep privacy as a cornerstone.

In this regard, Edge Intelligence (EI) is becoming a key technology to be accommodated within the "computing continuum" domain. Nevertheless, EI has hard requirements in terms of computation, latency and privacy, which increases the level of complexity for "computing continuum" systems.

Currently, the architectures and management methodologies for these new systems are built based on the first Internet systems, founded on client-server architectures. The appearance of Cloud computing has provided new features for these management methodologies [4], but still within the rigidity of the first systems. Now, the scale of these systems, their variety of components, their strict functional requirements and their mixture of topologies make "computing continuum" systems completely different from their germinal systems.

Whereas the first Internet systems could be completely specified by the application software. The emerging "computing continuum" systems do not allow a perfect specification from the application level. Furthermore, their characteristics are completely dependent on their underlying infrastructure, as "computing continuum" systems are seamlessly blending application and infrastructure. This implies that the responsibility of the application rely in both the application software and the underlying infrastructure software. Hence, this new actor requires a higher level of responsibility, as it is of utmost importance for their definition and success.

Simply put, due to the characteristics of their underlying infrastructure, these systems behave similarly as complex systems. In this regard, we focus our approach on two complementary methodologies to address their definition and management. First we use the Markov Blanket [5] concept to specify the causality relations between the components of the system, forcing that the state of the system is the central node of this representation. And second, we use ideas from the Free Energy Principle [6], which comes from neuroscience, to provide adaptive means to the system leveraging the Markov Blanket representation.

This article presents our vision for dealing with these new systems. We aim at breaking with previous methodologies and present a new way to define and manage the computing continuum systems that is fostered by their current characteristics.

The rest of the paper is organized as follows, section II provides our arguments to develop a new management methodology for "computing continuum" systems. Then, section III develops our vision for the proposed methodology. Section IV describes an application to

978-1-6654-1621-4/21/$31.00 ©2021 IEEE
DOI 10.1109/CogMI52975.2021.00032

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.

provide an example of the characteristics of the systems that are being targeted, then, section V describes the infrastructure needed to develop such a system and ends with a short example about what we would expect from our methodology. We finish the article in section VI with the conclusions and future steps foreseen to develop our vision.

II. Background

This section develops the ground for the need of new methodologies for managing "computing continuum" systems. To do so, we first present the shared characteristics observed on all these systems, and then, we also develop the main requirement for EI applications.

A. Computing continuum

There are several common characteristics of these systems, which stem from their underlying infrastructure, that require to be taken into account in order to properly understand and tackle their needs.

- **They are geographically distributed in large environments**. It can range from a city street to an entire country. Actually, they can even be internationally distributed, for instance considering production and distribution chains or applications to control pandemic outbursts.
- **They encompass multitude of components**. Only to develop a system to control street traffic, there can be several cameras in each street intersection, other sensors on the traffic lights, on the pedestrians crossings, around the cycling way, etc. But this are only sensors, it also requires computing units, gateways and many other components.
- **Their components are largely varied in characteristics and capacities**. Just on the edge, there can be found IoT sensors, edge gateways, cloudlets or single board computers (SBCs). Actually, components on the cloud tier can be considered homogeneous, but as a system is developed towards the Edge or IoT tiers the heterogeneity of components is manifold.
- **There are many dependent interconnections between their components**. These applications can be seen as ecosystems of components gathering, distributing and analysing data to create knowledge for users or other parts of the same application. This creates a large set of dependencies among components, which requires to be handled with precaution not to develop a fragile system.
- **They are influenced by the environment (open systems)**. Their performance can be affected by external or environmental effects. One can not assume that the system will continuously and indefinitely operate as designed without any intervention. The system can be affected by network issues, some components can become unavailable due to many reasons, it can suffer security threats, it can encounter corrupted data or it can suffer from unbalanced requests due to its geographical distribution.

The first idea drawn from the previous set of characteristics is that "computing continuum" systems mainly depend on the infrastructure where they are deployed. Hence, it is required that the underlying infrastructure software takes the responsibility of the system's performance. In this regard, our vision emphasizes the role of the infrastructure over the application from an overall system perspective.
The second idea worth highlighting is that the "computing continuum" systems behave similarly to any complex systems as they share many characteristics. Actually, *a system is complex if its behavior crucially depends on the details of the system* as said in [7]. In this regard, the new approach has to embrace the idea that the system is complex and has to take advantage of the technologies and methodologies develop for this type of systems [8].

B. Edge Intelligence

In general, EI makes reference to both AI on Edge and AI for Edge [9], where the first references using AI applications at the Edge tier and the second to use AI to enable the Edge tier. In this regard, we are focusing on the needs to have AI on the Edge, for the remaining of the article we are using this interpretation of EI.
EI applications are characterized by an important interaction between the different computing tiers [9]. In this regard, EI proposes to solve various problems derived from cloud centered architectures, such as latency issues in model inference, the processing of sensitive data and saving bandwidth by preprocessing the raw data at the origin.
Therefore, EI systems have hard constraints on the following characteristics:

- **Latency**. Interactive applications that require a smooth mixed reality for the user experience are very latency sensitive.
- **Performance**. Video-based analysis applications have high computational requirements.
- **Privacy**. Inference on sensitive data requires clear and trustworthy data pipelines to ensure high privacy standards.
- **Context awareness**. Geographic awareness for mobility applications is needed or energy awareness for optimizing performance on remote deployments.

From the previous set of requirements that this type of systems face, we can, again, draw two main conclusions. First, they require adaptive capabilities on the underlying infrastructure. Given their distribution, complexity and scale the only way to ensure that they are compliant with the application requirements, if an external perturbation affects the system, is to allow the system to autonomously adapt to

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.

its own underlying infrastructure.
Second, the management mechanisms for these systems need to be proactive, this means that they have to adapt before the harm from the perturbation is done. Again, the complexity of these systems can make not feasible to autonomously fix an issue once it has spread as this type of systems can suffer from cascade failures.

III. Vision

This section describes our high-level vision to define and manage the emerging systems of the "computing continuum". We are concerned that specifying this type of systems with static architectures, as it has been done since the first client/server Internet architectures, is not enough for dealing with them. Therefore, we propose a managing methodology that goes beyond any specific architecture of the system, and develops tools that can autonomously control "computing continuum" systems.

A. System representation

The complexity of these systems requires high level abstractions to describe the state of the application. Their representation has to encode its complexity but not to expose it. Otherwise, taking management decisions within such complexity becomes not feasible. At this point, it can be argued that deep learning models could take that complex input in order to select the best adaptive mechanism, however, due to explainability issues that would not be suited for many systems.
Furthermore, the system state representation has to be high-level and abstracted, so that it can be used in different application domains from healthcare, to retail or autonomous vehicles.
Our vision uses similar state variables that the ones proposed in [4] for cloud systems, these are *Resources*, *Quality* and *Cost*. They provide a high-level representation and a clear understanding of the system state which, additionally, can be related to the underlying infrastructure used. Nevertheless, they require to be further developed given the heterogeneity inside "computing continuum" systems, as already identified in [10].
Now, it is required to set the high-level system variables in a framework that allows retrieving them from practical observations. In other words, how can we know the values for *Resources*, *Quality* and *Cost* directly from system observation?
This can be achieved by leveraging the Markov Blanket [5] concept. In general terms, it consist of inferring the value of a random variable, the system state variables, from only those variables that provide meaningful information about it. This means, that if we relate a set of meaningful system metrics with the application state variables, their observation will be enough to infer the overall system state.
This set of metrics, as well as their precise relation with the system state variables, needs to be specified through the application requirements given that depending on the application domain some relations or metrics can be more relevant for the overall system. Hence, from one side this set of metrics relate to the system state variables, and from the other side, these relate with the actual computing-continuum resources. In other words, the underlying infrastructure of the system is aggregated and filtered through a set of specific metrics, determined by the application requirements, which are then related with high level system state variables that provide a comprehensive view of the overall system state.
The Markov Blanket is usually represented as a directed acyclic graph (DAG). Therefore, it can encode causality relations between the nodes of the graph. This provides mechanisms to obtain the causes of the system state change, which is useful to take actions with respect to that. Furthermore, encoding causality also provides a temporal dimension to the system description, allowing to develop concepts such as system evolution.
Another benefit, obtained from using a Markov Blanket to represent the system, is that it can be set as a causality filter in order to determine the scope of the problem. In this regard, we set the Markov Blanket centered over the system state variables, so that the specific metrics are its Markov Blanket, but the resources from the underlying infrastructure lay outside, which provides a more manageable scope for the methodology.
Additionally, due to the system's scale and complexity the representation has to be able to allow nested systems' representation, in which the same methodological analysis can be used at different scales. In this regard, the concept of the Markov Blanket, as well as, its DAG representation allows to focus on the entire application, or to have lens and observe a smaller part.
To sum up, we have presented how to represent and relate the system state of a "computing continuum" system with a set of metrics without losing contact with the underlying infrastructure. This way our management methodology encodes causality relations between them, allowing to precisely identify issues on the system. Nevertheless, we haven't yet defined means for the system to adapt to this disturbances or perturbations.

B. System adaptation

The perturbations that any "computing continuum" application can suffer mostly come from the dynamic behavior of the characteristics of its underlying infrastructure, which derive from its intrinsically complex nature. Hence, leveraging the causality filter provided by the Markov Blanket allows to analyze the underlying infrastructure of the system as if it is the environment. Therefore, this perspective allows us to address a "computing continuum" system as an entity that requires being adapted to its environment, which is dynamic and causes perturbations on the system's performance.

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.

Leveraging the temporal dimension of the system state provided by the causality relations, it is possible to define the concept of system equilibrium, which provides a new approach for these systems to decide when an adaptive mechanism is required. In this regard, if the system's equilibrium is disturbed then the system requires to perform an action to recover its equilibrium. It is worth mentioning that we tie an equilibrium state with a specific configuration of its underlying infrastructure, therefore another system equilibrium will have another infrastructure configuration, which can lead to a different operation mode for the application. Nevertheless, the requirements for the application are embedded in the system relations, therefore a different operation mode does not mean that the requirements are not fulfilled.

Hence, we can now complete the DAG representation of the system's Markov Blanket with the children nodes of the high-level representation of the system's state, as nodes that represent the state of the possible actions that the system can perform over the underlying infrastructure. It is important to remark that doing this comes at a cost, given the Markov Blanket definition, these new action nodes that are children of the system state are also required to infer its values. Simply put, they define the configuration of the underlying infrastructure with respect to the application, and this is needed to understand the system's state.

This action nodes can also be understood as valves, that due to a change of pressure upstream, given by the values of their parent nodes, they are induced to change their state. In that case, it would motivate a different path for the gas or liquid that they are routing. For the sake of completeness, these action states can have other parents, besides the system state variables. These can be both, a node from the set of metrics or the underlying infrastructure, as seen in Figure 1.

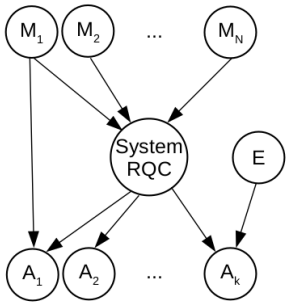

Figure 1. Complete DAG of a generic "computing continuum" system representation. M represents the set of specific metrics, R, Q, C are the three high-level system state variables, A are the action states and finally E represents an environment state that can directly affect an A.

It is important to remark that leaving the underlying infrastructure as a part of the environment is not contradictory with the idea that we also want to emphasize here: the underlying infrastructure requires higher level of responsibility on the system performance. However, it is more practical to provide these mechanisms to the infrastructure through the system actions rather than involving the entire underlying infrastructure on the system's definition. Additionally, this lets the door open to provide means to different systems for sharing components of the underlying infrastructure. Figure 2 provides a schema for the representation described.

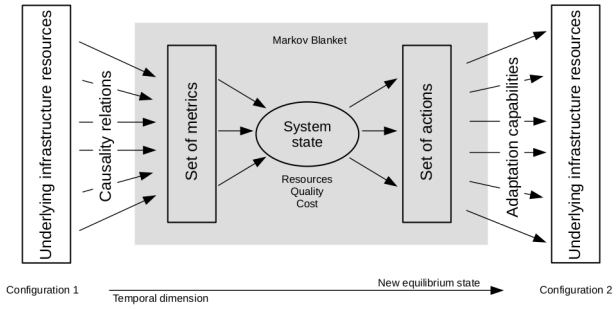

Figure 2. The schema shows a "computing continuum" system represented through a Markov Blanket showing its relation with the environment, or its underlying infrastructure. It also can be seen the temporal dimension of the system that relates consecutive configurations of the underlying infrastructure with the system's evolution and its equilibrium states.

Interestingly, there is work in neuroscience that has been inspirational to develop our vision for the managing methodology of "computing continuum" systems. This develops a model of the brain using a Markov Blanket and derives the Free Energy Principle (FEP) [6] in order to mimic the behavior of the brain, which consists on minimizing the free energy of the system, not the free energy from thermodynamics, but the difference between an expected observation and the obtained one. In this regard, it is not yet clear if the FEP can help on defining the best action for our system to take, as there are unresolved issues, such as defining generative models for the expected observations, as precised in [11], but it is a promising field to explore.

C. System intelligence

At this point our vision has presented a new representation for "computing continuum" systems that provides mechanisms for them to adapt to perturbations coming from the environment, which has been defined as their underlying infrastructure. However, as exposed in Section II these are complex systems, hence, it is not realistic to assume that for each situation the precise tool to trigger is known; even if, it is specified by the free energy or a metric that, in general, grasps the behavior of the system. Therefore, it is required to provide also the system with tools for learning about the environment, itself and its actions.

In this regard, our vision places the system's Markov Blanket in the same framework that any agent for reinforcement

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.

learning is placed, as can be seen from Figure 3. Therefore, by exploiting this well-known paradigm, it will be possible to provide knowledge to the system on three axes. The

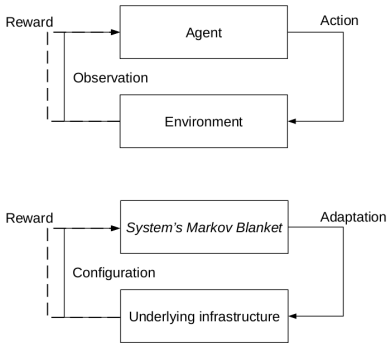

Figure 3. Reinforcement learning schema for computing continuum applications.

first axis will develop the system's knowledge about its environment, to better specify the relations between the underlying infrastructure and the system metrics. Also, it will develop knowledge on the general behavior of its infrastructure, so that it can predict or foresee when some infrastructure resources might appear or vanish. The second axis will focuses on the system itself, so that the relations between the metrics and the system state can be further specified, or the system state can be better linked to the requirements specified by the application, improving the overall performance. It will also learn the relation with the action states and, eventually, define new possible adaptation mechanisms. The third axis will deal with the relation between the actions of the system and its environment, simply put, it will focus on model the effect of an action on the environment.

IV. Use case

This section describes a use case that crosses domain boundaries and is settled in Food Computing, Smart City as well as Smart Health. We aim to showcase the different devices, requirements and data flows between cloud and edge to illustrate the complexity on "computing continuum" systems. The use case revolves around Smart Retail and illustrates a scenarios in which we explain how customers can interact with the store, and additionally how the store can react to the customers' actions. The former showcases different applications that can help the customer navigating throughout the store and recommend products. It is meant to depict typical situations while visiting a store and how different applications can act together to improve the overall experience. Also, it emphasizes the store's abilities to identify and react to accidents that happen daily in supermarkets around the world.

Shopping trip. It starts when the customer enters the store, upon which the customer's smartphone recognizes

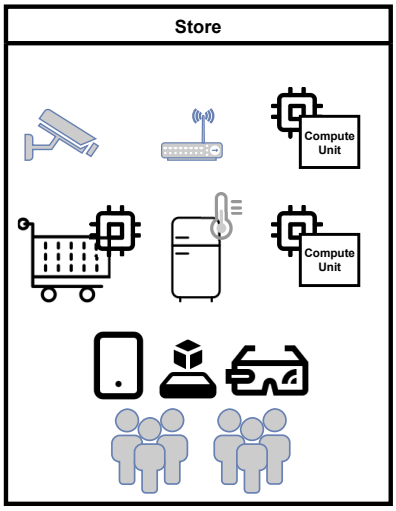

Figure 4. Different compute elements and actors in store

the specific store and connects to the local access point in order to retrieve the store's map. At the same time, the store recognizes the user due to the user's active membership and can record the shopping trip for later analysis in a centralized and large cloud of the company, which ingests anonymized data, and also to provide personalized recommendations for the customer computed on a local edge cluster to preserve data privacy.

During the shopping trip, users can get support from Augmented Reality (AR) applications. These could be integrated into smart wearables (i.e., Smart Glasses), smartphone or the shopping cart. Smart Glasses offer the user to get unobtrusive recommendations during their stay. Customers can profit from several recommendations that all can stem from different sources. For example, the store might offer a discount on soon-to-be-perished products in order to save food wastage. Personalized pricing can be offered by the customer's loyalty or previous shopping trips. Activity recognition and awareness of medical conditions or dietary preferences further widen the spectrum of possible inputs. However, in this field privacy requirements have to follow the highest standards. Health based recommendations can be classified into short term or long term. That the customer may purchase protein-rich products after training is short term, while recommending healthy food due to chronic diseases or allergies are seen as long term.

To engage and increase customer trust, augmented reality (AR) applications can inform customers of these recommendations that are safely processed either on edge nodes in the store or even on-device. Shopping carts also offer possible interactions, by tracking the items that are put into them and displaying recommendations on small mounted displays. These screens can act as personalised surfaces [12] and offer contextual information based on the customer. Personalized

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.

spaces can also be found on shelves or fridges and detect users in near proximity. These actions (advertisements) can be categorized as hyper targeting and can have a negative impact on the customer [13]. In contrast to highly personalized recommendations, that require coordinated process to satisfy privacy concerns, there exist other data sources that can efficiently predict the user's desire. Seasonal patterns of vegetable growth can be learned for regions and societal preferences. Additionally observations and learning based on restricted areas (i.e., district) can lead to fine-grained demand forecasts. In the same manner, business' can adapt marketing strategies in order to show advertisements based on environmental factors. After selecting different items from the shop, the user can checkout via a self-checkout and are aided by an avatar [14].

We can further expand the scenario showing what possible sequence of events are triggered in case an article falls from a shelve onto the floor and causes spillage or, in case of a jar, creates a dangerous environment for customers. We assume that cameras are installed throughout the shop to monitor the entire floor. Real time video analytics pose a big problem due to large bandwidth needs, low latency requirements and privacy concerns [15]. Therefore, edge nodes in the store can process the video stream and send anonymized pre-processed data back to the cloud, where big data jobs can extract patterns from the video and perform analysis [16]. Analytics at the edge also allows the shop to identify customers that are in near proximity of the accident and can in this case warn them [17]. The store can upon detection automatically act on these events. For example, the light can be changed in this part of the shop and a cleaning brigade is sent. Due to the emergence of vacuum cleaning robots in households, we believe that this cleaning brigade can be supported by machines. Through the advances in AI and the combination of compute power at the edge, robots can coordinate themselves and use sophisticated AI models in real time [18], [19].

This scenario highlighted how users can interact with the store (i.e., AR) but also how the store can interact with customers. It depicts the typical shopping trip of many people but we also want to showcase technologies that can react to more unusual events that require the store to act and prevent possible injuries.

V. Solution

This section first describes the required system to develop the use case depicted in Section IV. Then, it proposes a managing scenario for the use case, in order to provide an idea of how the proposed methodology would work on that example. The latter will help deriving the first required steps to pursue this research.

A. System

Based on our use case, we explain the different layers and components that are necessary to implement our vision of a Smart Store. The description is split into two parts. First, we introduce the compute continuum, its layers and their advantages as well as disadvantages. Second, based on the bottom layer, the sensors and users, common components are explained and put into context of the use case.

1) Compute continuum: We divide the continuum into four layers that vary in network latency, performance and privacy guarantees, see Figure 5. The cloud has been established over the last years as unlimited pool of compute resources. Platforms offer nowadays a variety of compute units (e.g., CPU & GPU) and make them suitable to train in a centralized manner AI models [20]. Though we examine several disadvantages that hinder the implementation of applications that require low latency and with high privacy requirements. Our use case showcases applications that fit

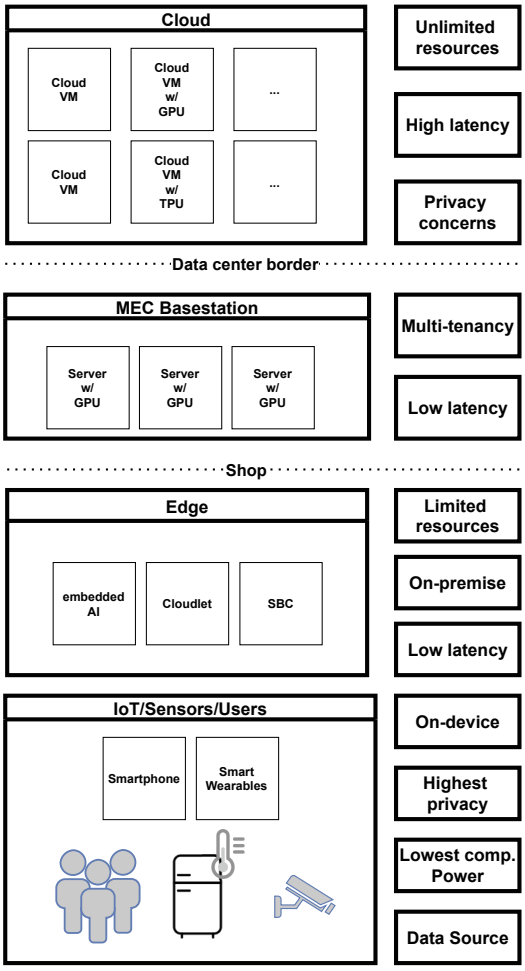

Figure 5. Compute continuum over four layers. Each layer is characterized through high level advantages as well as disadvantages.

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.

into the cloud paradigm to perform intensive computational task efficiently, but also highlights applications that require low latency and guaranteed privacy oriented computation. The next layer is based on the Mobile Edge Computing paradigm, which considers deployment of compute resources in mobile network basestations (i.e, 5G) [21]. These basestations can act as internet backhauls for stores and offer computational resources. Due to the low latency and high bandwidth, computational offloading to these stations offers several advantages for our use case. For example, base stations can be equipped with moderately powerful compute units, including modern hardware accelerators, and therefore offering low latency inference required for video analytics. A caveat is that resources are not unlimited, as in the cloud, and smart resource management is necessary in order to fulfill all requests. Further, the basestations are operated by telcom providers and therefore privacy guarantees may not be fully transparent, especially when considering that many other customers can make use of the hardware, allowing malicious attempts to possibly intercept traffic [22], [23]. We title the next layer *Edge* and summarize in this layer all dedicated compute units that are directly at the edge. In our example, compute units located in-store fall into this category. The store owners have full governance over this hardware and therefore can fulfill privacy concerns that are required for several applications from our use case (i.e., user entry registration, spillage detection). We consider the resources even more limited, but offer through local Wi-Fi the lowest latency and highest privacy guarantees. The last layer compounds the Internet of Things (i.e., temperature sensors) and user devices (i.e., smartphone). This layer has the highest trade-off in terms of performance versus privacy as well as network latency. On-device computation can be used to train models in a Federated Learning setup on highly sensitive data without exposing sensitive data [24].

The proposed layer describe the compute continuum and include all requirements that we encounter in our use case. A distributed computation model, that varies in layer, is required to fulfill all needs, concerning network latency, performance and privacy guarantees.

2) Store components: After describing all layers and their characteristics, we want to focus on the last layer. This is where the data is generated and selected computational tasks are executed on. In the first step a set of components is described that are deployed in the store, while afterwards the focus is on user specific equipment. Further, this section is dedicated to show the rich diversity of sensors and highlights the compute continuum's ability to consume the heterogeneous data.

Accordingly to our use case, we start at the entrance of the shop, where cameras are placed to possibly recognize loyal customers. Recognition can also be done via NFC or card readers. The entrance also triggers the smartphone to download the local map and metadata to show location specific details and guidelines. The corridors contain various sensors (i.e., cameras and light) to observe the state in order to trigger events. For example, a camera can detect spillage, which results in the dispatching of a cleaning unit. Shelves register the items and can be read from customers via RFID to visualize additional information (i.e., nutritional). Fridges monitor their state via temperature, door and power sensors and can report anomalies (i.e., temperature drop) to the system. During the shopping trip, users can use shopping baskets to carry products around the store. They also support customers by showing information based on the products in the basket. We envision that baskets are equipped with RFID readers, Wi-Fi connection and a screen. They connect to the store and offer visual guidance to users. Further, Bluetooth can be used to connect smartphones and buying baskets for a seamless and safe integration of personalized applications. At the checkout customers can choose between low and high social presence [14] alternatives: a RFID-based automated paying system or a robot that acts as an employer and represents an avatar. The latter option requires microphone, speakers and a screen to communicate with the customer. A diverse set of sensors is required in the store in order to realize our proposed use case. Additionally, we identify in the next step a set of components that users provide (see Figure 6). Whereas we split it into two sections: sensors and smartphone. Sensors represents distinct components, while the smartphone itself offers various components necessary for our use case. While Smartphones are already ubiquitous, Smart Belt, Smart Glasses and other smart wearables [25] have yet to arrive in the general public. Though, they allow the development of new pervasive applications that can support people in their everyday lives. Especially Virtual and Augmented Reality, enabled through Smart Glasses and similar, can guide and help customers during their stay. The Smartphone represents the source for private and sensitive data, which requires on-device computation and thus must not leave the form (at least not prior to anonymization). Activity recognition and private data (i.e., dietary preferences) can improve recommendations.

B. Illustrative example

The previous subsection presents a typical solution based on the "computing continuum" paradigm for the use case presented in section IV. Now our intention is provide a flavor on how this can be translated to our vision to manage "computing continuum" systems.

Let's suppose that we have been able to represent the described system using the representation developed. Hence, the system state is represented with *Resources*, *Quality* and *Cost* and it is the center of a Markov Blanket, which encode the causality relations between its components. The analysis of the state shows that the overall *Cost* of the system is increasing, endangering the system's equilibrium. Then, by leveraging the causality relations on the representation, we

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.

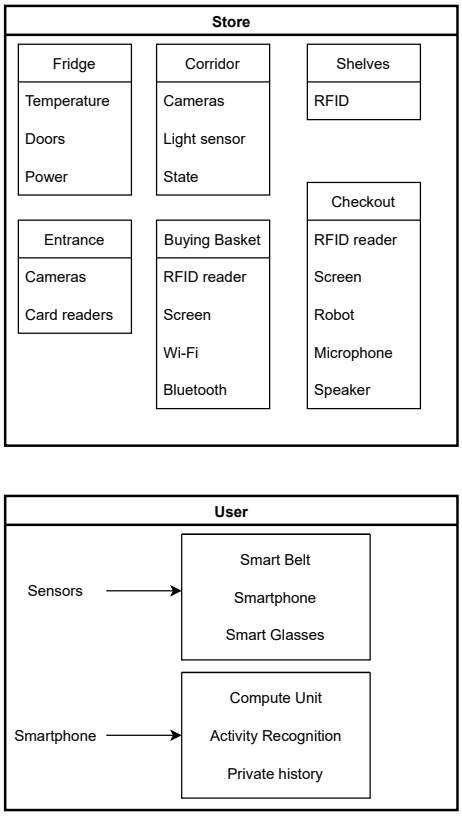

Figure 6. Components encountered in the use case. The top collection represents distinct sensors in the store, while the bottom one shows personal user equipment.

are able to localize the issue affecting the system's *Cost*.
Now, instead on dealing with the entire system, we can focus on the system's part that is risking the equilibrium of the system due to the increase on *Cost* by using the nested capacity of the Markov Blanket. This part is in charge of an application running in the cloud that performs AI-based inference for the store's recommendation system. This process is monitored through several metrics, one of them is cost-effectiveness [26], [27] which provides an idea of how resources and cost are linked, which can be easily mapped to the central state of this system.
In this regard, the value of the metric is deviating from its specified value given by the application requirements due to the large amount of customers visiting the shop, and the consequent increase of requests for the recommendation system. Hence, *Cost* is increasing risking the system's equilibrium. Therefore, an action state is activated triggered by this equilibrium change, and it modifies the way data from customers are gathered, lowering its granularity. Therefore, inference is performed less frequently, which decreases the application's accuracy but not below its requirements, and more importantly, it is able to return *Cost* to an acceptable value recovering the system's equilibrium.
It is obvious that this is only a possible solution that the framework could provide. Similarly, another action state could have been triggered moving this coarser inference from the Cloud to an Edge Cloudlet providing a better impact on cost but limiting the availability of low latency resources. The complexity of this second solution shows that a learning framework is required on top of the system in order to develop these solutions for the system.

VI. Conclusion

This article presents our vision for the required methodology to manage "computing continuum" systems. We have shown that the complexity inherent to their underlying infrastructure, as well as, the hard requirements of developing AI system on the Edge make obsolete the previous managing techniques for distributed Internet systems.
Then, we explain our vision for the required methodology, that takes advantage of the Markov Blanket concept in order to create a framework to represent the system and to develop its adaptive mechanisms through a scheme similar to the one used for reinforcement learning.
Through a smart retail use case we showcase the complexity of "computing continuum" systems by explaining its functional features and developing a typical implementation. Finally, we are providing a glance on how we would our methodology to manage these systems.
The fulfilment of this vision still require many steps, but we take this endeavor convinced that it is the needed path to provide society with these "computing continuum" applications that will sculpt our future. In this regard, future work will be centered on developing the representation of these systems by leveraging learning techniques to develop directed acyclic graphs from data and specific constraints.

References

[1] P. Beckman, J. Dongarra, N. Ferrier, G. Fox, T. Moore, D. Reed, and M. Beck, "Harnessing the computing continuum for programming our world," in *Fog Computing*, A. Zomaya, A. Abbas, and S. Khan, Eds. John Wiley & Sons, Ltd, apr 2020, ch. 7, pp. 215–230. [Online]. Available: \\\url{https://onlinelibrary.wiley.com/doi/full/10.1002/9781119551713.ch7}

[2] D. Balouek-Thomert, E. G. Renart, A. R. Zamani, A. Simonet, and M. Parashar, "Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows," *International Journal of High Performance Computing Applications*, vol. 33, no. 6, pp. 1159–1174, nov 2019. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1094342019877383

[3] L. Baresi, D. F. Mendonça, M. Garriga, S. Guinea, and G. Quattrocchi, "A unified model for the mobile-edge-cloud continuum," *ACM Transactions on Internet Technology*, vol. 19, no. 2, apr 2019. [Online]. Available: https://doi.org/10.1145/3226644

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.

[4] S. Dustdar, Y. Guo, B. Satzger, and H. L. Truong, "Principles of elastic processes," *IEEE Internet Computing*, vol. 15, no. 5, pp. 66–71, sep 2011.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[6] K. Friston, J. Kilner, and L. Harrison, "A free energy principle for the brain," *Journal of Physiology Paris*, vol. 100, no. 1-3, pp. 70–87, jul 2006.

[7] G. Parisi, "Complex systems: a physicist's viewpoint," *Physica A: Statistical Mechanics and its Applications*, vol. 263, no. 1-4, pp. 557–564, feb 1999.

[8] J. M. Ottino, "Engineering complex systems," *Nature 2004 427:6973*, vol. 427, no. 6973, pp. 399–399, jan 2004. [Online]. Available: https://www.nature.com/articles/427399a

[9] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, 2020.

[10] H. L. Truong, S. Dustdar, and F. Leymann, "Towards the Realization of Multi-dimensional Elasticity for Distributed Cloud Systems," *Procedia Computer Science*, vol. 97, pp. 14–23, jan 2016.

[11] V. Raja, D. Valluri, E. Baggs, A. Chemero, and M. L. Anderson, "The Markov blanket trick: On the scope of the free energy principle and active inference," *Physics of Life Reviews*, sep 2021.

[12] S. Shahzadi, M. Iqbal, and N. R. Chaudhry, "6g vision: Toward future collaborative cognitive communication (3c) systems," *IEEE Communications Standards Magazine*, vol. 5, no. 2, pp. 60–67, 2021.

[13] E. T. Bradlow, M. Gangwar, P. Kopalle, and S. Voleti, "The role of big data and predictive analytics in retailing," *Journal of Retailing*, vol. 93, no. 1, pp. 79–95, 2017.

[14] D. Grewal, S. M. Noble, A. L. Roggeveen, and J. Nordfalt, "The future of in-store technology," *Journal of the Academy of Marketing Science*, vol. 48, no. 1, pp. 96–113, 2020.

[15] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *computer*, vol. 50, no. 10, pp. 58–67, 2017.

[16] M. Popa, L. Rothkrantz, Z. Yang, P. Wiggers, R. Braspenning, and C. Shan, "Analysis of shopping behavior based on surveillance system," in *2010 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2010, pp. 2512–2519.

[17] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *2018 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2018, pp. 1–6.

[18] V. K. Sarker, J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "Offloading slam for indoor mobile robots with edge-fog-cloud computing," in *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*. IEEE, 2019, pp. 1–6.

[19] A. K. Tanwani, N. Mor, J. Kubiatowicz, J. E. Gonzalez, and K. Goldberg, "A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4559–4566.

[20] D. Crankshaw, G.-E. Sela, X. Mo, C. Zumar, I. Stoica, J. Gonzalez, and A. Tumanov, "Inferline: latency-aware provisioning and scaling for prediction serving pipelines," in *Proceedings of the 11th ACM Symposium on Cloud Computing*, 2020, pp. 477–491.

[21] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.

[22] F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the ai-driven internet of things (iot)," *Information Systems*, p. 101840, 2021.

[23] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on multi-access edge computing security and privacy," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1078–1124, 2021.

[24] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[25] M. Chen, Y. Jiang, N. Guizani, J. Zhou, G. Tao, J. Yin, and K. Hwang, "Living with i-fabric: Smart living powered by intelligent fabric and deep analytics," *IEEE Network*, vol. 34, no. 5, pp. 156–163, 2020.

[26] S. Nastic, A. Morichetta, T. Pusztai, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, "SLOC: Service level objectives for next generation cloud computing," *IEEE Internet Computing*, vol. 24, no. 3, pp. 39–50, may 2020.

[27] T. Pusztai, S. Nastic, A. Morichetta, V. Casamayor Pujol, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, "A Novel Middleware for Efficiently Implementing Complex Cloud-Native SLOs," in *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, 2021.

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on May 16,2022 at 06:54:01 UTC from IEEE Xplore. Restrictions apply.