Yang Xiang · Jingtao Sun
Giancarlo Fortino · Antonio Guerrieri
Jason J. Jung (Eds.)

LNCS 11226

Internet and Distributed Computing Systems

11th International Conference, IDCS 2018
Tokyo, Japan, October 11–13, 2018
Proceedings

Springer

Lecture Notes in Computer Science 11226

More information about this series at http://www.springer.com/series/7409

Yang Xiang · Jingtao Sun
Giancarlo Fortino · Antonio Guerrieri
Jason J. Jung (Eds.)

Internet and Distributed Computing Systems

11th International Conference, IDCS 2018
Tokyo, Japan, October 11–13, 2018
Proceedings

Springer

Editors
Yang Xiang
Swinburne University of Technology
Hawthorn, VIC, Australia

Jingtao Sun
National Institute of Informatics
Tokyo, Japan

Giancarlo Fortino
University of Calabria
Rende (CS), Italy

Antonio Guerrieri
ICAR-CNR
Rende (CS), Italy

Jason J. Jung
Chung-Ang University
Seoul, Republic of Korea

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-02737-7 ISBN 978-3-030-02738-4 (eBook)
https://doi.org/10.1007/978-3-030-02738-4

Library of Congress Control Number: 2018950190

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web, and HCI

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Following the previous ten successful editions of IDCS – IDCS 2008 in Khulna, Bangladesh, IDCS 2009 in Jeju Island, Korea, IDCS 2010 and IDCS 2011 in Melbourne, Australia, IDCS 2012 in Wu Yi Shan, China, IDCS 2013 in Hangzhou, China, IDCS 2014 in Calabria, Italy, IDCS 2015 in Windsor, UK, IDCS 2016 in Wuhan, China, IDCS 2017 in Fiji — IDCS 2018 was the 11th in the series to promote research in diverse fields related to the Internet and distributed computing systems.

Modern systems such as distributed systems, cloud computing, mobile computing, edge computing, fog computing, and cyber-physical systems have a tendency toward complexity, elasticity, dependability, and security especially when dealing with dynamic events or actions in their environments and/or Internet applications. We not only need to keep those systems running normally, but also need them to be self-adaptive to many changes. On the other hand, the development of the Internet is very rapid, and it has already entered the 5G era. The Internet as a society infrastructure and the widespread use of mobile edge, wireless wearable devices, or IoT sensors have laid the foundation for the emergence of innovative network applications and transportation and logistics. Under the influence of these most advanced technologies, human production and life are gradually changing. The academic and industrial worlds are constantly developing and innovating in areas such as mechanical learning, artificial intelligence, and media stream processing. These technologies enrich and improve not only the quality of life of modern people, but also the process of integration in many fields, the huge amount of data processing, and the integration of the digital world with the physical environment; they also contribute toward constructive development in biological, agricultural, and policy.

IDCS 2018 received papers on emerging models, paradigms, technologies, and novel applications related to cloud computing, distributed systems, Internet of Things, cyber-physical systems, wireless sensor networks, next-generation collaborative systems, extreme-scale networked systems, and self-adaptive systems.

The audience included researchers and industry practitioners who were interested in different aspects of the Internet and distributed systems, with a particular focus on practical experiences with the design and implementation of related technologies as well as their theoretical perspectives.

IDCS 2018 received a large number of submissions from which 23 regular papers were accepted after a careful review and selection process. This year's conference also featured four invited talks: (1) "Inter-Cloud Computing over Academic and Public Clouds" from Associate Professor Atsuko Takefusa, National Institute of Informatics and The Graduate University for Advanced Studies, Tokyo, Japan; (2) "Interconnecting the Edge with Software-Defined Overlay Virtual Private Networks" from Professor Renato J. Figueiredo, University of Florida, USA; (3) "Towards Opportunistic IoT Services: A Novel Paradigm for Engineering the Next-Generation IoT Systems" from Professor Giancarlo Fortino, University of Calabria, Italy; and (4) "Simultaneous

Scheduling of Routes for On-Demand Bus and Walking Passengers" from Associate Professor Naoki Shibata, Nara Institute of Science and Technology, Japan.

IDCS 2018 was held in the wonderful Hitotsubashi Hall, National Center of Sciences Building, in the center of Tokyo, Japan. The conference organization was supported by the National Institute of Informatics (Japan), Swinburne University of Technology (Australia), Western Sydney University (Australia), and the University of Calabria (Italy).

The successful organization of IDCS 2018 was possible thanks to the dedication and hard work of a number of individuals.

Specifically, we would like to thank our program chairs, Bahman Javadi (Western Sydney University, Australia), Giuseppe Di Fatta (University of Reading, UK), Lei Zhong (Toyota InfoTechnology Center, Japan), Sisi Duan (University of Maryland, Baltimore County, USA), and Markus Ullrich (University of Applied Sciences Zittau/Görlitz, Germany), our publicity and industry chairs, Antonio Guerrieri (ICAR-CNR, Italy), Mukaddim Pathan (Telstra, Australia), and Qiang Wang (Wuhan University of Technology, China), and our Web chair, Mingkang Chen (Central China Normal University, China), for their commendable work with the conference organization. We also express our gratitude to the general chair, Yang Xiang (Swinburne University of Technology, Australia), and the conference co-chairs, Jingtao Sun (National Institute of Informatics, Japan), Giancarlo Fortino (University of Calabria, Italy), and Jason J. Jung (Chung-Ang University, South Korea), for their supports of the conference.

October 2018

Yang Xiang
Jingtao Sun
Giancarlo Fortino
Antonio Guerrieri
Jason J. Jung

Organization

General Chair

Yang Xiang	Swinburne University of Technology, Australia

Co-chairs

Jingtao Sun	National Institute of Informatics, Japan
Giancarlo Fortino	University of Calabria, Italy
Jason J. Jung	Chung-Ang University, South Korea

Program Chairs

Bahman Javadi	Western Sydney University, Australia
Giuseppe Di Fatta	University of Reading, UK
Lei Zhong	Toyota InfoTechnology Center, Japan
Sisi Duan	University of Maryland, Baltimore County, USA
Markus Ullrich	University of Applied Sciences Zittau/Görlitz, Germany

PhD Workshop Chair

Kazushige Saga	National Institute of Informatics, Japan

Publicity and Industry Chairs

Antonio Guerrieri	ICAR-CNR, Italy
Mukaddim Pathan	Telstra, Australia
Qiang Wang	Wuhan University of Technology, China

Web Chair

Mingkang Chen	Central China Normal University, China

Steering Committee - IDCS Series

Jemal Abawajy	Deakin University, Australia
Rajkumar Buyya	University of Melbourne, Australia
Giancarlo Fortino	University of Calabria, Italy
Dimitrios Georgakopolous	RMIT University, Australia
Mukaddim Pathan	Telstra, Australia
Yang Xiang	Swinburne University, Australia

Program Committee

Abdelkarim Erradi	Qatar University, Qatar
Andrea Omicini	Università di Bologna, Italy
Andrea Vinci	ICAR-CNR, Italy
Antonio Guerrieri	ICAR-CNR, Italy
Antonio Liotta	University of Derby, UK
Bahman Javadi	Western Sydney University, Australia
Bin Guo	Institut Telecom SudParis, France
Carlo Mastroianni	ICAR-CNR, Italy
Claudio De Farias	PPGI-IM/NCE-UFRJ, Brazil
Claudio Savaglio	University of Calabria, Italy
Dimitrios Katsaros	University of Thessaly, Greece
George Pallis	University of Cyprus, Cyprus
Giancarlo Fortino	University of Calabria, Italy
Gianluca Aloi	University of Calabria, Italy
Giorgio Terracina	University of Calabria, Italy
Giuseppe Di Fatta	University of Reading, UK
Hu Xiaoya	Huazhong University of Science and Technology, China
Jason Jung	Chung-Ang University, South Korea
Jie Mei	Wuhan University of Technology, China
Lei Zhong	Toyota InfoTechnology Center, Japan
Marcin Paprzycki	IBS PAN and WSM, Poland
Markus Ullrich	University of Applied Sciences Zittau/Görlitz, Germany
Mengchu Zhou	New Jersey Institute of Technology, USA
Mukaddim Pathan	Telstra Corporation Limited, Australia
Norihiko Yoshida	Saitama University, Japan
Paolo Trunfio	University of Calabria, Italy
Pasquale Pace	University of Calabria, Italy
Raffaele Gravina	University of Calabria, Italy
Ragib Hasan	University of Alabama at Birmingham, USA
Riaz Ahmed Shaikh	King Abdul Aziz University, Saudi Arabia
Ruppa Thulasiram	University of Manitoba, Canada
Sergio Ochoa	University of Chile, Chile
Sisi Duan	University of Maryland Baltimore County, USA
Sun Jingtao	National Institute of Informatics, Japan
Valeria Loscri	Inria, France
Wenfeng Li	Wuhan University of Technology, China
Xinqing Yan	NCWU, China
Xiuwen Fu	Shanghai Maritime University, China
Yang Xiang	Swinburne University of Technology, Australia
Jingjing Cao	Wuhan Technology University, China
Bui Khac Hoai Nam	Chung-Ang University, South Korea
Zhengxue Cheng	Waseda University, Japan

Contents

Increasing Interoperability Between Heterogeneous Smart City Applications

Alexander Rech[1(✉)], Markus Pistauer[1], and Christian Steger[2]

[1] CISC Semiconductor GmbH, 9020 Klagenfurt, Austria
{a.rech,m.pistauer}@cisc.at
[2] Graz University of Technology, 8010 Graz, Austria
steger@tugraz.at

Abstract. Due to the increasing need for networked systems we can observe a rapid advance of IT-solutions in various sectors. However, most of the developed systems are custom-tailored solutions for specific problems and application areas, leaving us with a set of diverse frameworks. The resulting jungle of heterogeneous systems makes it difficult to find common interfaces for interconnecting the underlying businesses with each other, especially in regard to Smart City concepts. We envision a new paradigm shift towards "Smart City as a service" fueled by increased interoperability between different services with an additional emphasis on privacy-preserving data processing. This would contribute to a new level of connectivity between the environment, service providers, and people, facilitating our daily activities and enhancing the level of trust of the users. In order to achieve interoperability in the context of smart, connected cities, we propose the design of a generic, platform-independent novel architecture for interconnecting heterogeneous systems, their services, and user pools.

Keywords: Interoperability · Connected services · Data privacy
Smart City

1 Introduction

The need for stronger interrelationships between different services is constantly increasing. However, when interoperability between independent systems has to be established, we face several issues. One major problem is that systems tend to be tailored to a specific application area, so boundaries are often rigid and inflexible and only users of one specific application are addressed. Second, combining two or more systems can be a considerable challenge and, if integration is indeed attempted, integration methodologies can be very time- and cost-consuming which in turn results in longer time-to-market cycles. Third, when different parties have to share user-related datasets, special attention has to be paid to the correct use and processing of the data, especially in view of the increasingly stringent data protection laws.

Y. Xiang et al. (Eds.): IDCS 2018, LNCS 11226, pp. 64–74, 2018.
https://doi.org/10.1007/978-3-030-02738-4_6

A Smart City is a place where many different information systems come together. At first glance, some of these systems are already embedded into our daily lives. Smart services such as managing parking facilities, renting e-bikes or placing orders via mobile applications are already feasible in many cities. However, closely examined there is poor or almost no interaction between services of different kinds, as for the above mentioned, for instance. Therefore, in order to cope with the increasing amount of heterogeneous services in a Smart City, pervasive computing technologies should offer collaboration methodologies among businesses and their users to enable benefits for all participants involved. This, in further consequence, would contribute to a new level of connected services within cities, enabling businesses to collaborate more easily with each other. Due to this cooperation, more users could be addressed, resulting in additional advantages in terms of higher revenue. In contrast, users are provided with easier access to different, independent services. More specifically, users may save time and profit from additional offers, which in the end facilitates their daily activities.

We call into question the current situation where businesses are limited to a predefined scope regarding their services and users. We want to increase collaboration between independent entities through a federated solution, so that multiple parties do not have to completely adapt to each other's requirements in order to achieve mutual benefits. In further consequence, to accomplish a paradigm shift towards "Smart City as a service" we are working on a generic, platform-independent architecture that gives businesses and their applications access to a predefined set of services of other participating service providers. To avoid high integration costs, the functionality to enable this federated solution shall be offered as additional software layer to pre-existing systems. Additionally, since data needs to be shared across multiple parties, the privacy of the users shall be protected by abstracting their real identity.

This paper is organized as follows. Section 2 describes related work while Sect. 3 discusses design choices of our architecture and gives an overview on its components. Last but not least, Sect. 4 summarizes our ideas and offers suggestions regarding future work.

2 Related Work

Everyday life is becoming evermore connected to the digital world. Information services and applications run on an extensive set of different systems and are in constant interaction with people and their environment. Federated concepts are essential for increased system and business relationships. In order for these systems to communicate seamlessly with each other and to overcome the heterogeneity between them, interoperability is becoming increasingly important. Especially due to the increasing need for networked services from different businesses, interoperability is required ever more. According to IEEE the term interoperability is the "ability of two or more systems or components to exchange information and to use the information that has been exchanged" [1]. A more generic definition of interoperability is given by [2]. It is the "ability of things to interact for a specific purpose, once their differences have been overcome". This definition tries to expand the concept where a system is simply composed of single components, by saying that a system may also consist of multiple diverse autonomous subsystems. Only if cooperation between all systems is ensured and solutions are made to eliminate discrepancy between all components, can interoperability be achieved within the resulting system of systems [3].

In this context, a Smart City can be viewed as a larger system overarching diverse subsystems, which on their own act as autonomous components that should work together seamlessly. New concepts for information sharing among different services as well as for enhancing collaboration between Smart City services are being analyzed and elaborated [4,5].

The connection of different services is also a topic that is being elaborated by the Horizon 2020 STEVE project. The focus of the project is to implement and test a human-centric approach to electro-Mobility-as-a-Service (eMaaS), by connecting different e-vehicle solutions and "gamified" services for enhancing users' awareness, engagement and vehicle energy efficiency [6].

The idea of encapsulating an organization's functionality within an appropriate interface, advertising it as services, and trying to connect it with other similar services is not new [7]. In this sense, semantic web service-paradigms have been trying to enhance automated discovery, access, combination, and management of web services for years. Both academia and industry are researching ways to provide machine understandable representations of services, their properties, capabilities, and behavior [8,9].

Furthermore, interoperable networking is becoming increasingly important to concatenate different backends. There already exist a few approaches on how to combine heterogeneous cloud systems for sharing different datasets across geographically distributed resources and for better cooperation between different services [10–12].

Especially in the electronic marketplace area, the notion of interoperability is often closely associated with the term integration. From the perspective of service providers, pure integration methodologies may be advantageous: If businesses merge into clusters they often have a decreased management overhead with less setup and maintenance costs for instance. However, the downside is that such integration technologies and frameworks are often associated with high adaption costs, especially for businesses with rigid standards [13].

In our opinion, we do not only need new concepts for merging multiple systems and their services, which often proves to be difficult, costly or time consuming. More importantly, we need to emphasize on technologies that give multiple independent businesses the possibility and freedom to interconnect each others' services with negligible adaption costs for maximized mutual benefit. A collaboration of different service providers also means that each company may address more users. In this context, the more companies work together and exchange data, the more important data privacy protection rules become, especially in light of increasingly stringent data protection rules like the European General Data Protection Regulation (GDPR) [14]. Therefore, increased interoperability should not only offer advantages such as easier cooperation between multiple independent businesses, but also adequate data protection for end users.

3 Design of Architecture

This section is about the design of our proposed architecture. First, requirements are discussed and an overview on our architecture is given in 3.1. Afterwards, the architecture's business layer components and the core layer components will be described more detailed in Sects. 3.2 and 3.3.

3.1 Architectural Overview

Our proposed architectures addresses the following topics:

– *Interoperability.* Collaboration between different independent businesses and service providers shall be improved. Nowadays we do not have much cooperation between independent businesses, e.g. parking provider with a restaurant, or a museum with a car-rental service. No matter which product or which service a company offers, it would be beneficial if companies were able to utilize services from other businesses for increased collaboration between them as well as for addressing more users, and for offering more services to them. An example use case would be the following: A user authenticates himself at a parking gate using his smartphone for getting access to the shopping mall. As soon as he enters the garage, he receives a new service in form of a digital restaurant voucher, pushed onto his phone. In this example, two independent businesses, a garage provider and a restaurant, collaborate by sharing their users and services. The reverse case is also possible: a user getting a parking voucher as reward for eating in a restaurant.
– *Integration cost.* It is time consuming to define agreements between companies and to elaborate common interfaces. Therefore, the platform shall provide ways to facilitate this agreement process. Pre-defined interfaces shall be provided for faster integration and shorter time to market cycles.
– *Data Privacy.* When data, especially user-related data needs to be passed between different parties for enabling collaboration, privacy aspects become more important. In this context, only anonymized user data shall be passed between companies and the link between real and anonymized data shall be revocable at any time in order to be compliant with data protection rules.

With these requirements in mind we now present and discuss the structure of our architecture, which will be described more precisely in 3.2 and 3.3. Figure 1 provides a high-level overview. The overall architecture consists of at least one business layer as well as a core layer, both of which are further subdivided into several components.

Each business layer is managed by a company and contains service and user related datasets. Generally speaking, it consists of client and vendor applications as well as of a corresponding server unit that offers application specific functionality. While the client application is adapted to the user's needs, such as giving an overview on current services, the purpose of the vendor application is to provide and manage these services. Both applications operate in constant interaction with the business layer which provides them with application specific functionality and keeps the data synchronized across all devices.

Regarding the core layer's position in the proposed architecture, it is placed on top of a business layer. It can be reached by the components of the business layer via a RESTful interface. The idea behind the concept of the core layer is to provide a common trusted layer responsible for abstracting products and services from various business layers in order to establish an interrelationship between multiple heterogeneous businesses. Even if a company already utilizes its own business layer and corresponding applications, it should be feasible to communicate to the core layer in order to create new means for interacting with other service providers or rather other business layers. As interrelationships between different businesses involve a lot of time and effort for all participating businesses, we see the need for enhancing this integration and cooperation process in regard to speed and costs. Therefore, the entities of the business layer shall be extended by dedicated software libraries defining how to interact with the core and its subcomponents: the Tokenization and the Federation layer.

In order to interconnect different platforms and to anonymize datasets coming from different business applications, we introduce the Tokenization layer. Its task is to abstract application specific data of the business layer such as product and user data for bringing down all datasets to a common denominator, and increasing the level of privacy of the end users. After this abstraction process a mapping between the anonymized and the real data is created and stored at business side. This means that only the business layer which triggered the tokenization process of the data in the first place, may access the derived data from the core layer. Last but not least, there is an additional module called Federation layer. It is responsible for creating and managing services between different vendors. Furthermore, it is able to arrange trusted agreements between independent parties. This makes it possible to automate the issuing process of services as soon as predefined conditions are met.

3.2 Business Layer Components

The business layer provides users with features based on the respective application area of the system and the corresponding service provider. In summary, it consists of the three key components described below. Companies who already utilize their own business layer, may establish a link to the core via the RESTful interface.

Client Application. The mobile client app is used by customers of a specific vendor, shop or company. Together with application specific features it offers a user interface for acquiring services, e.g. parking app for booking parking tickets, restaurant app for ordering food. Regarding the redemption process of

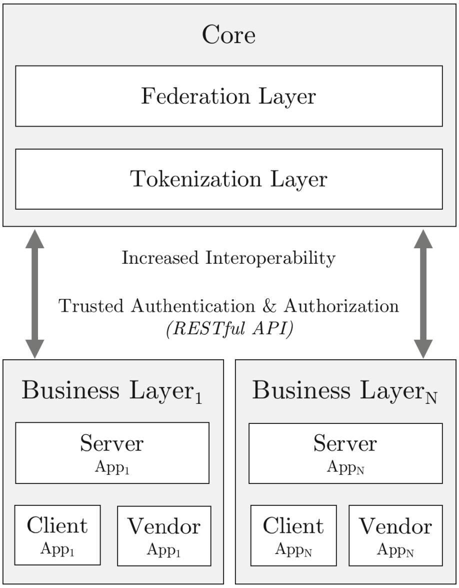

Fig. 1. Architectural overview

the acquired service we distinguish between two cases, depending on the specific application area and the service type in question. On one hand, the client application may redeem the service by directly communicating to the vendor application. In a parking use case, the drivers may redeem their parking voucher directly at the embedded vendor application of the parking gate. On the other hand, services can also be redeemed directly via an online interface without additional interaction of other devices (online food delivery).

Vendor Application. The vendor-side application functions as front-end for service providers to manage both their user pools as well as their services. Depending on the use case, the vendor application decides on how the services should be redeemed, e.g. online via server requests or by first communicating locally to a specific device. Just as with the client application, the vendor application works closely with the application server.

Application Server. The application server communicates with client and vendor devices and keeps them synchronized. On one hand, it is responsible for managing the users of the client application. On the other hand, it offers the possibility to manage shop- and service-relevant data on vendor side. If a business decides to cooperate with other service providers, interaction with the core layer, which is explained in the following, is required.

3.3 Core Layer Components

The goal of this layer is to seamlessly interconnect different services, thus contributing to a new level of interrelationships in the context of Smart City applications. The idea is to attach it on top of an existing system as an additional trusted ubiquitous layer that can be reached via a predefined set of REST calls. It is subdivided into two parts which will be described in the following. Figure 2 gives an overview on the most important components of the core layer, while Fig. 3 describes the high-level-interrelationship between the most important datasets of the different layers.

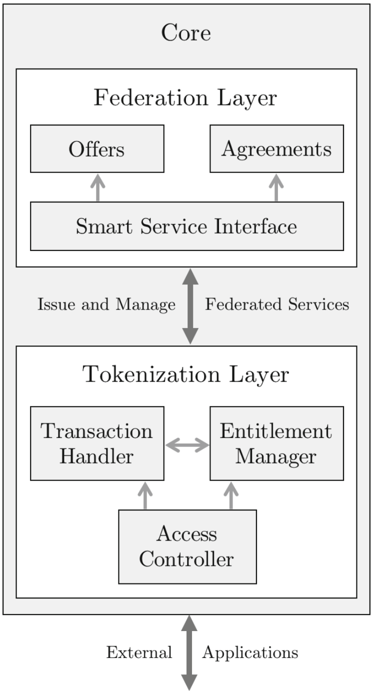

Fig. 2. Overview of core layer subdivided into Tokenization and Federation layer

Tokenization Layer. This layer provides the entry point for external applications. In the first place, the business layer of a system may use it for enabling cooperation with other applications. Since each company usually has its own implementation, this layer is mainly there to abstract the datasets coming from different businesses. This abstraction process converts the information received from the application layer into a mapping of several IDs that uniquely identify businesses, their services and users. Additionally, the Tokenization layer keeps log of transactions via the *Transaction Handler*. Each transaction refers to the participating users on client and vendor side and the acquired or redeemed services in question. According to the specified use case, the Tokenization layer may also communicate with the Federation layer to issue new services in form of entitlements or goods. The Tokenization layer distinguishes between two different token types:

- *AccessToken.* The user- and device related AccessTokens are managed by the *Access Controller*. They are issued in the course of a distributed Kerberos-based authentication procedure and signed by the core layer. Furthermore, they are responsible for providing user devices with authentication when communicating with the core layer and for countering eavesdropping and replay attacks.
  The real user data stored on the business layer is never forwarded to the core layer, but converted into anonymous IDs which are linked with the AccessTokens. Only the business in charge of the creation of the user account knows the user data, thus anonymizing the user's identity for all other participating businesses.
- *ServiceToken.* All business and service relevant data is transformed into a signed ServiceToken by the *Entitlement Manager*. Just as for the user data, also in case of service data, unique IDs are derived, identifying the service and the company. Each ServiceToken belongs to one AccessToken, authorizing a specific user to utilize a certain service. Furthermore, it can be determined which entity purchased or redeemed which services, and how many of them, via the *Transaction Handler*.

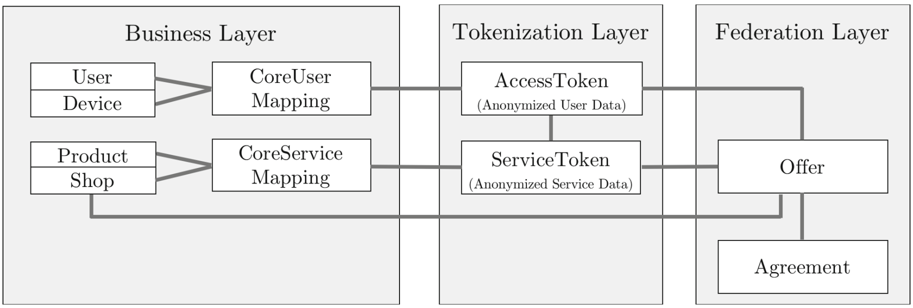

Fig. 3. Overview of interrelationship between datasets of business, Tokenization and Federation layers

Federation Layer. The Federation layer is an interface for interconnecting services from different businesses. It can be considered as federated service market place where businesses can place their products and services for special conditions or purchase services from other participants. Consequently, this cooperation enables companies to offer a wider range of services. Combined with gamification methodologies, goods and services of other companies can be forwarded as rewards to customers, enhancing both user satisfaction as well as revenue for the participating service providers. The Federation layer operates in interaction with the Tokenization layer and utilizes the tokenized data, including AccessTokens and ServiceTokens, to address services and users. In contrast to the actual user data which is only stored on business layer side, some product and company relevant data has to be forwarded from the Business layer to the Federation layer to be utilized by other interested businesses. In this sense, obligatory fields such as the name of the company and its address, have to be submitted together with optional key value pairs identifying the corresponding company's products and services. In summary, the Federation layer handles the services under discussion in two different ways:

– *Offer.* A service provider utilizing the Federated layer is authorized to publish specific services in this domain, stored as offers. Other providers are able to search for offers and acquire them. This layer functions as a market place where products and services can be found according to specific key words such as name, price, or category, for instance. Providers may acquire them and a new ServiceToken would be generated inside the Tokenization Layer. In further consequence, the link between the ServiceToken and the user's AccessToken will be established, enabling the user to utilize the service. In this sense, loyal customers can be rewarded with goods and services of other companies, e.g. a user who rented an e-bike gets an additional voucher for visiting a museum.
– *Agreement.* The idea behind agreements is that there should be the possibility to issue entitlements, aka ServiceTokens in an automated way. As soon as businesses subscribe to an offer, it becomes an agreement and is extended by specific datasets identifying the interested party as well as a condition that specifies under which circumstances the execution of the agreement should be triggered. A condition specifies how many goods or services of a company need to be acquired by a user before a reward is issued, or which specific service needs to be redeemed in order to trigger a reward, for instance. In any case, every time a customer acquires new services or redeems them, new transaction history entries will be generated inside the *Transaction Handler* of the Tokenization layer. Subsequently, it will be determined if the predefined condition is met through interaction between the Tokenization and the Federation layer. If so, the good or service specified inside the agreement block, will be acquired by the subscribed vendor and forwarded to the customer who triggered the current transaction. Furthermore, customers and vendors can keep track of the service-completion-state by utilizing the entries of the *Transaction Handler* as reference. An additional advantage for customers is that, even if they do not actively track the completion state, they can still receive rewards in an automated way.

4 Conclusion and Future Work

On account of poor integration methodologies, high integration costs, and rigid system interfaces, the widespread use of ubiquitous Smart City applications is still in its inception. In order to overcome these problems we are currently working on a generic software platform for increasing the cooperation between participating heterogeneous systems, service providers and user pools. In this paper we gave an overview on our proposed architecture, mainly subdivided into a business and a core layer. The core layer can be seen as additional layer that can be put on top of existing networked systems and enables them to exchange their services. Furthermore, in regard to user-data privacy, especially in view of strict data protection laws, we introduced a Tokenization Layer which task it is to reconcile user- and business-related datasets coming from arbitrary application servers. The user data itself is masqueraded and saved in a privacy-preserving way. In contrast, the Federation Layer provides means for service-based collaboration between independent companies.

The design of the architecture presented in this article is a starting point for us. Parts of the framework discussed in this paper will be evaluated within the European Union's Horizon 2020 project STEVE. Future work will concentrate on a research on semantic service oriented architectures to extend the interfaces between the business and core layer in a more dynamic way. Additionally, we are currently investigating on how to increase the level of trust between different entities. Blockchain in combination with smart contracts could therefore be suitable for our plans.

Acknowledgments. This work is supported by CISC Semiconductor and Graz University of Technology. A part of this work has been performed in the European Union's Horizon 2020 STEVE project under grant agreement No. 769944.

References

1. IEEE standard glossary of software engineering terminology. IEEE Std 610.12-1990, pp. 1–84 (1990)
2. Motta, R.C., De Oliveira, K.M., Travassos, G.H.: Rethinking interoperability in contemporary software systems. In: IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS), pp. 9–15 (2017)
3. Boardman, J., Sauser, B.: System of systems - the meaning of of. In: IEEE/SMC International Conference on System of Systems Engineering, pp. 6–11 (2006)
4. Cledou, G.: A virtual factory for smart city service integration. In: Proceedings of the 8th International Conference on Theory and Practice of Electronic Governance, pp. 536–539 (2014)
5. Antonić, A., Marjanović, M., Žarko, I.P.: Modeling aggregate input load of interoperable smart city services. In: Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems, pp. 34–43 (2017)
6. The European Commission: Smart-taylored L-category electric vehicle demonstration in hetherogeneous urban use-cases (2017). http://www.steve-project.eu
7. Cardoso, J., Sheth, A.: Introduction to semantic web services and web process composition. In: Cardoso, J., Sheth, A. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 1–13. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30581-1_1

8. Benatallah, B., Nezhad, H.R.M.: Interoperability in semantic web services. In: Cardoso, J., Sheth, A. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 22–25. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30581-1_3
9. Cardoso, J., Miller, J., Su, J., Pollock, J.: Academic and industrial research: do their approaches differ in adding semantics to web services? In: Cardoso, J., Sheth, A. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 14–21. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30581-1_2
10. Bavier, A., et al.: GENIcloud and transcloud: towards a standard interface for cloud federates. In: Proceedings of the 2012 Workshop on Cloud Services, Federation, and the 8th Open Cirrus Summit, pp. 13–18 (2012)
11. Zou, M., et al.: Collaborative marketplaces for eScience: a medical imaging use case. In: International Conference on Collaboration Technologies and Systems (CTS), pp. 500–506 (2014)
12. Akolkar, R., et al.: Towards cloud services marketplaces. In: 8th International Conference on Network and Service Management (CNSM) and Workshop on Systems Virtualiztion Management (SVM), pp. 179–183 (2012)
13. Guo, J.: Business interoperability on E-marketplace. In: Xu, L.D., Tjoa, A.M., Chaudhry, S.S. (eds.) Research and Practical Issues of Enterprise Information Systems II. ITIFIP, vol. 254, pp. 257–267. Springer, Boston, MA (2007). https://doi.org/10.1007/978-0-387-75902-9_26
14. European Parliament and European Council: Regulation 2016/679 on the protection of natural persons with regard to the processing of personal data. Official J. Eur. Union (L119) **1**, 1–88 (2016)