Collaboration Mechanisms for IoT Platform Federations Fostering Organizational Interoperability

Ivana Podnar Žarko*, Joaquin Iranzo Yuste†, Christoph Ruggenthaler‡, Jose Antonio Sanchez Murillo†, João Garcia§, Pavle Skočir*, and Sergios Soursos¶
*University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia, ivana.podnar@fer.hr, pavle.skocir@fer.hr
†ATOS Spain SA, Spain, jose.sanchezm@atos.net, joaquin.iranzo@atos.net
‡AIT Austrian Institute of Technology GmbH, Austria, christoph.ruggenthaler@ait.ac.at
§Ubiwhere, Portugal, jmgarcia@ubiwhere.com
¶Intracom SA Telecom Solutions, Greece, souse@intracom-telecom.com

***Abstract*—The Internet of Things (IoT) is faced with a plethora of platforms which are built and operated as siloed solutions. However, the lack of interoperability and collaboration between platforms will negatively influence their adaptability to support future business cases targeting large-scale and cross-domain IoT deployments. While syntactic and semantic interoperability, as a prerequisite for platform cooperation, has been widely addressed by standardization and research actions, organizational interoperability enabling direct and business-driven interactions between IoT platform owners has so far not been in focus. In this paper we present the interoperability approach pursued by the project symbIoTe that implements a flexible interoperability framework, and put forward its solution for organizational interoperability offering decentralized and secure interworking between IoT platforms—*IoT-platform federations*. Three main collaboration mechanisms are introduced and elaborated: management of Service Level Agreements (SLA) between federated platforms, multi-level trust and reputation management, and a bartering mechanism for fair sharing of federated resources. The aforementioned mechanisms are put in place to foster decentralized platform collaboration which has higher potential to be adopted by platform owners compared to centralized approaches.**

I. Introduction

The need for cross-domain IoT applications dealing with multiple aspects of everyday life is becoming more apparent nowadays. Vertically isolated IoT platforms need to be extended to cover other domains in which, however, the companies may not have the required expertise. Strategic partnerships are expected to be the only viable option since companies need to enrich their existing solutions to tackle larger and more complex projects. Organizational interoperability and collaborative IoT solutions are thus becoming a key ingredient of a future IoT ecosystem targeting large-scale IoT deployments.

Organizational interoperability in the IoT context is defined in the ETSI Whitepaper [1] as "the ability of organizations to effectively communicate and transfer (meaningful) data (information) even though they may be using a variety of different information systems over widely different infrastructures." One way to realize organizational interoperability is to allow the formation of IoT platform federations between multiple partnering institutions in locations or domains originally out of their reach. We define IoT platform federations as associations between two or more platforms which are willing to share access to their IoT resources in order to facilitate their fair interaction and collaboration. For example, platforms can be enabled to perform collaborative sensing/actuation tasks to complement each other's infrastructure, and to interact directly in a decentralized way without exposing their business relationship to a centralized authority. Reasons for such a collaboration can vary: e.g., similar IoT platforms that operate in different locations can federate to offer seamlessly to their clients IoT services in other locations, or collocated platforms can benefit from each other by forming partnerships to offer cross-domain solutions.

A platform federation enables applications to use resources managed and operated by other federated platforms as if they were offered by a single platform. This removes the burden of interacting with multiple platforms and different interfaces from an application or a service, while platforms increase the portfolio of offered resources. For example, if Platform B offers to share data produced by its static temperature sensors within a federation formed by platforms A and B, this means that Platform A can use and offer temperature readings produced by those sensors as if Platform A was managing the devices. Platform A offers in turn its temperature sensors located in another location to Platform B, or Platform A offers humidity sensors to complement Platform B's offerings. Therefore, an application or service registered within a single platform can use all shared sensors as if they were managed by a single platform.

Since successful federations need to be orchestrated by a set of mechanisms which ensure fair usage of shared resources, resource usage and platform's offerings must be continuously monitored and validated (in a distributed manner) to avoid cases of free-riding or malicious/untrustful activities. Hence, we propose and implement a mechanism for defining and monitoring Service Level Agreements, a solution for multi-level trust and reputation management as well as a bartering solution to guide the sharing process among federated platforms. The aforementioned mechanisms are offered as open source components to be integrated with existing platforms extending thus their original set of features.

In this paper we present the technical details related to novel aspects of organizational interoperability introduced by the symbIoTe project[1], and put them in relation to the symbIoTe architecture which is built around a hierarchical IoT stack, as reported in [2]. symbIoTe enables platforms to federate and to act as prosumers: A platform acts both as a provider and consumer of resources primarily with a goal to extend resource offerings to its applications. Agreements on both the quality of shared resources and the bartering scenarios are made at the platform level. Such an approach allows for a decentralized federation management, where platforms share resources in a controlled, secure and trustful way with minimal intervention of a centralized authority.

The paper is structured as follows: Section II presents an overview of relevant projects and standardization efforts. Section III outlines the symbIoTe architecture for organizational interoperability, while Section IV provides further details on collaboration mechanisms for decentralized and trustful interactions within platform federations. Section V concludes the paper and outlines future research challenges.

II. Related work

IoT interoperability is in the focus of a number of project initiatives and standardization working groups that address this challenging topic by using the following strategies [3]: publishing standards, reference architectures and frameworks; defining protocols and media-type standards; and using abstract interface definitions and semantic technologies. This section presents an overview of IoT-EPI[2] projects and standardization activities that combine the aforementioned approaches to achieve interoperability in the evolving IoT ecosystem.

IoT-EPI was formed to build a vibrant and sustainable IoT ecosystem in Europe, maximizing the opportunities for platform deployments, interoperability and information sharing. The total of six projects, Inter-IoT, bIoTope, BIG IoT, AGILE, and VICINITY, including symbIoTe, have the objective to achieve interoperability between IoT platforms on different architectural levels. The general approach proposed by IoT-EPI projects is to define APIs that IoT platforms need to use and implement to become interoperable, and to make use of the developed interoperability-relevant services.

AGILE focuses on *syntactic interoperability* at both software and hardware levels [4] and develops a gateway that supports various wireless and wired IoT networking technologies for fast prototyping of IoT solutions. The other projects mainly focus on *semantic interoperability* and deal with some aspects of organizational interoperability. Similar to AGILE, VICINITY also considers lower levels of the IoT stack for interoperability [5]. It develops software for an interoperability gateway and targets semantic interoperability by specifying an ontology implemented using the W3C Web Ontology Language (OWL). It promotes usage of existing ontologies or standard information models, and selects a set of available ontologies to describe the exposed data which is offered to independent service operators who have opportunities to create new IoT services.

InterIoT proposes to achieve semantic interoperability by applying semantic data processing techniques for mapping between supported ontologies [6], [7]. A selected core ontology is used within the system (e.g., the W3C Semantic Sensor Network Ontology, SSN), while other ontologies, either domain-dependent or domain-independent, are mapped to the core ontology. InterIoT also implements direct and near real-time data translation between the supported ontologies. bIoTope uses the Open Messaging Interface (O-MI) and Open Data Format (O-DF) standards as the core APIs for the creation of an open, shared IoT space that integrates proprietary IoT systems implementing those APIs [8]. O-MI adds meta-information to a message payload relating to a specific vocabulary, so that platform owners can describe their resources according to their domain-specific vocabularies. In addition, platform owners can define business-related terms which specify how resources are to be used, e.g., how their usage will be charged. BIG IoT focuses on developing a common marketplace where platforms can register their resources, typically sensors, as continuous data sources, while developers consume the published data to create IoT services and applications [9]. One of the key features within the BIG IoT marketplace is vocabulary management which exposes semantic descriptions of platforms' data sources. Common information models and vocabularies must be used to enable data sharing between different platforms, applications, and services. Business interactions, such as charging and access control, are also offered by the common marketplace services.

One of the standardization organizations that promotes IoT reference architectures is oneM2M[3], a partnership of different standardization organizations, including the European Telecommunications Standards Institute (ETSI), and IoT companies. oneM2M defines standardized platform interfaces and aims to provide an interworking framework across different sectors [10]. It defines a minimal Base Ontology for oneM2M-compliant platforms, and proposes that other organizations map their ontologies to the Base Ontology. The architecture specifies an interworking proxy which is responsible for the full syntactic and semantic interworking, including the mapping of other data models and protocols to the ones specified by oneM2M. oneM2M addresses organizational interoperability by specifying communication between different IoT platforms within the infrastructure domain between IoT servers. The focus is on resource sharing in the form of mutual registration, resource announcement, and subscriptions to information about resources offered by different platforms [11]. However, features for the management of platform federations and collaboration mechanisms for fair and trustful interactions are not defined. Other organizations, such as AIOTI [12] and ITU-T [13], specify high level architectures of M2M/IoT system and service capabilities with a focus on syntactic and semantic interoperability.

Compared to other interoperability aspects, semantic interoperability is still in the spotlight of both ongoing projects

[1] https://www.symbiote-h2020.eu/

[2] IoT European Platforms Initiative, http://iot-epi.eu/

[3] http://www.onem2m.org/

and standardization initiatives. An IoT ecosystem comprising different platforms that natively use various information models is still state-of-the-art, and easy to use solutions are more than needed to facilitate the development of cross-platform and cross-domain IoT services and applications. While business relationships are considered by BIG IoT and bIoTope proposing centralized solutions for charging data usage, organizational interoperability facilitating business interactions in a decentralized and peer-to-peer manner is, to the best of our knowledge, not considered, with the exception of oneM2M. symbIoTe tackles this issue by proposing secure, fair and trustworthy interactions between platforms without a centralized mediator, so that IoT platform owners can engage in direct partnering relationships by use of *symbIoTe's platform federations*.

III. Platform Federations in symbIoTe

symbIoTe introduces the support for IoT platform federations by implementing interoperability-related features as microservices that are deployed and operated within an IoT platform's space, and that complement existing platform features managing platform's own resources[4]. Own resources shared within a platform federation become *federated resources* to enrich and broaden the number and diversity of resources offered to third parties by all federated platforms. By joining one or more federations, a symbIoTe-enabled platform can significantly expand its resource offerings beyond own capabilities, since its applications and registered users have access to all federated IoT resources. We assume that platforms choose to share selected resources freely within a federation, while access to a selected subset of such resources may be further managed by the symbIoTe bartering mechanism. Platforms exchange resource metadata describing federated resources to facilitate search within each platform's space, and associate Quality of Service (QoS) levels to shared resources. Service Level Agreement (SLA) management is in place to guarantee that the access to federated resources meets specified quality levels (e.g., availability and performance), and to rule out the possibility of platforms sharing less valuable or even malfunctioning devices.

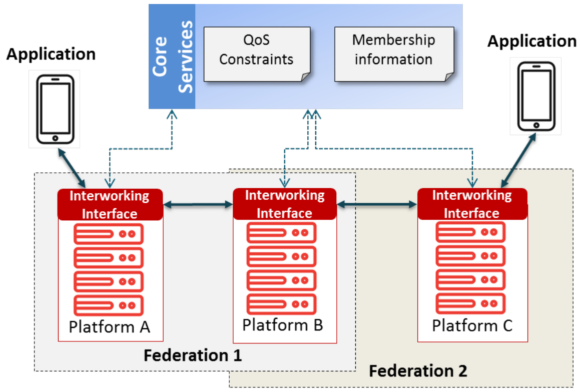

Fig. 1. Platform federations in symbIoTe

Fig. 1 depicts a high-level overview of platforms participating in federations, and stresses a decentralized nature of their interaction which frees this solution from a central resource repository, i.e., each platform maintains a local registry storing the metadata specifying federated resources. The owner of a federated resource is responsible to maintain its up-to-date descriptions and propagate any changes directly to other federated platforms in a publish/subscribe style. Only the federation membership and general QoS constraints are managed in a centralized manner by the symbIoTe Core Services to facilitate federation creation, while all other interactions happen either directly between federated platforms or between IoT applications and platforms. Note that federated platforms may either agree to use a common information model when describing federated resources, a Best-practice Information Model (BIM), or have to use a service for semantic mapping and alignment of different information models used by platforms within a federation. Of course, the first approach is simpler, but requires each platform to perform an of-line mapping of its internal information model to the BIM. The second approach complicates the management of metadata descriptions across platform registries: it requires the mapping between Platform-specific Information Models (PIMs) and online translation to resource metadata from one PIM to another. Further details on symbIoTe aspects relevant to semantic interoperability are provided in [14].

Our solution relies on symbIoTe-specific Attribute-based Access Control (ABAC) mechanism [15] to ensure that resources are accessible only to users of IoT applications and platforms who present the right set of attributes to platforms managing the federated resources. This means that a user must be registered and recognized by, at least, one of the platforms inside a federation to be granted access to a federated resource managed by another platform. The fine-grained access control implementation is based on the following three pillars: 1) certificates for authentication of both applications and users, 2) JSON Web Tokens (JWT) for authorization, and 3) a challenge-response protocol to verify the authenticity of two sides in every interaction.

In addition to the basic mechanism of resource sharing, we propose two additional collaboration mechanisms to foster fairness within federations: a multi-level trust and reputation management mechanism to monitor and award/penalize platform behavior, and a bartering mechanism to facilitate fair usage of selected federated resources. Access to a bartered resource is guaranteed as long as other federated platforms provide access to some of their resources in a fair way.

Fig. 2 shows components facilitating platform federations in symbIoTe. Interactions between platforms as well as access from applications to a federation occur through the Interworking Interface which exposes RESTful interfaces of the microservices within a platform's space. The following components are introduced on the platform side: *Federation Manager* manages all required federation information at the platform level by passing the information from the Core to the platforms and among federated platforms as required. *Platform Registry* enables the registration and update of federated IoT resources, i.e. their metadata, by all platforms

[4] The symbIoTe software is published as open source and available at https://github.com/symbiote-h2020. Microservices relevant to platform federations are located within the SymbioteCloud repository.

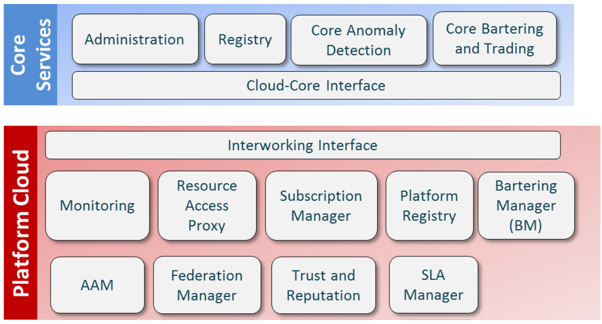

Fig. 2. symbIoTe components for platform federations

that are members of a federation, and utilized the services offered by *Subscription Manager* that forwards notifications (e.g., federation or resource updates) in a publish/subscribe manner to all platforms in a federation. *SLA Manager* hosts and monitors SLAs signed by the platforms when they join a federation and depends on the *Monitoring* component which gathers metrics about resources managed by the platform to produce aggregate metrics relevant to SLAs that will guarantee a defined level of QoS. *Authentication and Authorization Manager* (AAM) provides tokens and certificates for secure interactions between symbIoTe components, and authenticates and authorized symbIoTe-enabled applications/users. *Resource Access Proxy* (RAP) enables symbIoTe-compliant access to resources within an IoT platform in collaboration with AAM, and grants access with coupons to bartered resources. A platform needs to implement a RAP plugin to integrate its native resources with RAP services that adopt the OData (Open Data Protocol)[5] as the protocol for accessing IoT resources [2]. *Trust and Reputation* supports a platform owner or application in taking informed decisions about the selection of a platform and its federated resources for interaction by calculating trust and reputation metrics at the level of resources and platforms. *Bartering Manager* manages the bartering process within federation platforms as far as it happens in a decentralized way.

Centralized features are implemented by the following components: *Administration* offers a GUI-based administration tool for federation management, mainly membership and initial platform coordination. *Core Bartering and Trading* performs the validation of resource bartering interactions in a centralized way[6]. *Core Anomaly Detector* detects 0-day attacks (targeting vulnerabilities that are unknown at the deployment time) and other types of security violations by using a signatureless approach. *Registry* stores information about federations, such as membership and general QoS constraints.

Note that the interaction of platforms with the Core Services is minimal. Only two components, the Federation Manager and Bartering Manager are invoking centralized services using the Cloud-Core interface.

IV. Collaboration mechanisms

Service Level Agreement. The symbIoTe framework uses the mechanisms needed to maintain the specified QoS levels within federations. Both the shared and bartered resources must comply with a series of QoS parameters, mainly availability and load, to guarantee a predefined quality level to applications using them, and also to assist in the calculation of resource and platform trust scores. When a platform joins a federation, an SLA is signed based on QoS parameters and their constraints. The SLA Manager (SLAM) obtains the necessary metric data from the Monitoring component, which is responsible for gathering periodic reports relevant to the defined QoS parameters and metrics, checking and assessing that the agreements are respected. If at least one of the parameters is not respected, the SLAM generates a corresponding violation and notifies the federated platforms and other relevant components, such as the Trust and Reputation. A SLAM runs within each federated platform's space. SLAM is composed of several sub-components, as shown in Fig. 3.

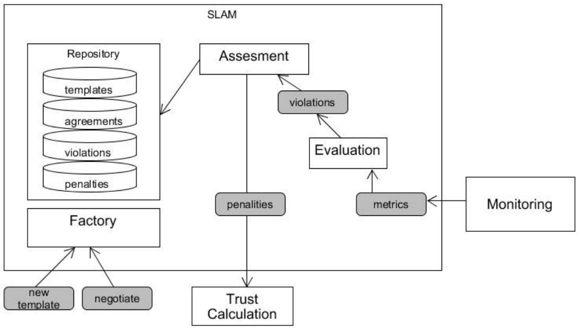

Fig. 3. Component diagram of the SLA Manager

SLA Factory is the main entry point which provides an API to external entities and takes care of the following steps: negotiation, interaction with the SLA Repository to generate and retrieve SLA templates and agreements, and activation of the SLA enforcement once an agreement is "signed". *SLA Repository* is responsible for storage of SLA templates, SLA agreements and events related to SLA violation actions. *SLA Assessment* manages SLA rules to determine the way to proceed when an SLA is activated. It detects SLA violations and generates notifications according to defined rules. It also interacts with the Trust and Reputation component. *SLA Evaluation* provides access to the monitoring information related to the agreed QoS aspects to determine whether SLAs are being fulfilled or not.

The SLAM is involved in the complete federation life cycle, both during federation creation and management. When a new platform joins a federation, a join message is generated by the Administration component and is sent to the Federation Manager passing the federation QoS constraints. The Federation Manager passes the constraints to the SLAM that creates an SLA. It does so by adding a facade to the SLA Factory that transforms these constraints to an SLA Template. Next, the template is saved in the SLA Repository and with this template the facade signs an agreement that is stored in the repository.

[5] OData is an ISO/IEC approved, OASIS standard that defines a set of best practices for building and consuming RESTful APIs. It is information model agnostic.

[6] We envision that this feature may be implemented by a distributed ledger offering thus a completely decentralized bartering solution.

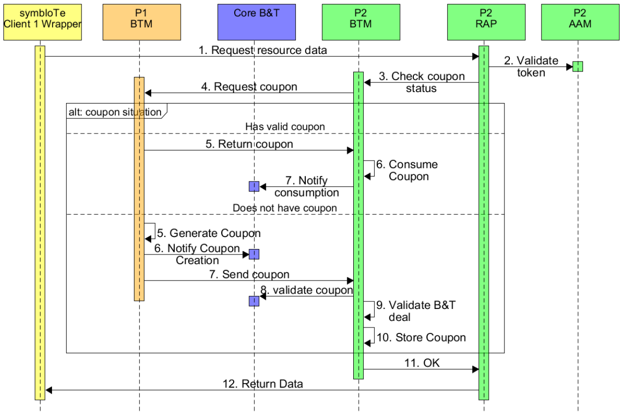

Fig. 4. Bartering process

The created SLA is returned to the Federation Manager which forwards it to the Monitoring component to notify it about the metrics that need to be monitored. The SLA Assessment component checks the active SLAs periodically and asks the SLA Evaluation to evaluate them based on the information received from the Monitoring component. When a violation is detected, the SLAM is informed by the SLA Assessment about the details of the event, and SLAM passes this information further to the Federation Manager.

The SLAM implementation is based on the WS-Agreement standard [16] which is widely used to negotiate SLAs in distributed scenarios. It provides the definition and mechanisms to automate the SLAs set-up, for monitoring and enforcement. This standard defines a format based on XML and specifies the following content relevant to our implementation: Context, Service Description Terms, Service References, Service Properties, and Guarantee Terms.

Bartering. The basic economic concept of bartering refers to a market situation when two or more market participants exchange their respective goods or services directly for other goods or services, without monetary implications. While the concept itself is a rather old one, it has been repeatedly criticized for its alleged inefficiency, for instance with respect to difficulties in matching suitable partners, issues with determining a common value metrics, and problems arising from the fact that certain goods may be indivisible and hence impossible to precisely match in terms of their value. However, the main justification for employing a bartering mechanism originates from the fact that it allows two parties to achieve a joint win-win situation without the need of resorting to an explicit exchange of money. In the context of IoT platform federations, most of the aforementioned problems disappear by definition: matching suitable partners is relatively easy, as all participating platforms are *prosumers*, while they also choose to enter into federations with partnering platforms under specified terms. Hereby, a service typically consists of allowing or making use of access to IoT resources, e.g., sensors and their corresponding data, which circumvents the problem of indivisibility: we can easily define small units of service and thus provide a mutually acceptable metrical unit for comparing the value of an offer or a request.

The symbIoTe bartering model is based on the concept of *coupons* that grant holders access to certain resources on defined platforms. A coupon contains the following information:

- *Issuer (platformId)* specifies who issued the coupon;
- *Beneficiary (platformId) (optional)* defines the beneficiary of the coupon. This is an optional field and, if left empty, the coupon can be passed around through several platforms.
- *Federation Identifier (federationId)* specifies the federation this coupon belongs to;
- *Resource Type*: the type of resources being bartered;
- *Expiration*: the expiry date of the coupon;
- *Single Use*: a Boolean indicating whether the coupon can be used only once or several times.

Fig. 4 presents the flow of actions when a platform wants to access another federated platform's resource under a bartering scenario. The basic idea is that a platform (P1), wanting to access another platform's (P2) resources, must provide a coupon that grants such access. If P1 does not have such a coupon, it will generate its own coupon and offer it in exchange for access to the desired resource in P2. P2 can later use the received coupon to access federated resources managed by P1, or possibly other federated resources offered under the bartering mechanism within the same federation. By keeping track of the coupons that are generated and used, the federation can identify platforms that are not contributing resources (i.e., platforms generating a lot of coupons that are unused by other platforms) and can take appropriate actions. The Core Bartering and Trading component is needed to keep track of all coupons and relevant events (creation, usage, consumption). This allows to monitor the bartering process within a given federation, to detect malicious and unfair platforms and to identify expired coupons. The bartering statistics can periodically be provided to the Trust and Reputation component. Note that platforms are free to define the rules for bartering their own resources. This means that platforms can specify, for example, if they are only willing to barter certain types of resources, or with platforms above a given trust level.

Trust and Reputation Management. To enable a reliable environment for distributed SLA enforcement and bartering transactions, additional trust management mechanisms are applied within the symbIoTe ecosystem. Similarly to the proposed definitions in [17] and [18], we distinguish between trust and reputation to reflect ecosystem reliability. Trust and reputation management is implemented as a multi-level approach addressing resource-related trust metrics, but also supporting a distributed reputation mechanism at the platform level, as depicted in Fig. 5.

Depending on the level and intended purpose, different metrics and parameters are taken into account to determine adequate resource trust and platform reputation values: *Resource Trust* is calculated per resource by the platform and is shared between all federated platforms. The value indicates the expected behavior and trustworthiness of the actual resource based on resource specific aspects, such as dependability or data stability. *Platform Reputation* reflects an internal, subjective factor calculated by each platform for each federated

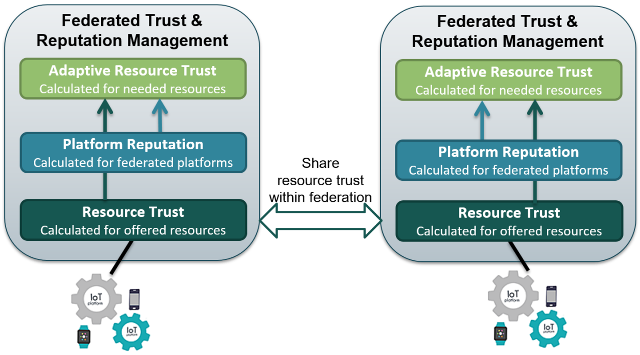
Federated Trust & Reputation Management
Adaptive Resource Trust
Calculated for needed resources
Platform Reputation
Calculated for federated platforms
Resource Trust
Calculated for offered resources
Share resource trust within federation
Federated Trust & Reputation Management
Adaptive Resource Trust
Calculated for needed resources
Platform Reputation
Calculated for federated platforms
Resource Trust
Calculated for offered resources
IoT
IoT
IoT
IoT

Fig. 5. Multi-level trust & reputation management approach

platform. The algorithm takes different internal data sources (e.g., bartering transactions or anomaly detection) and historic events into account and thus enhances the reliability perspective with collected long-lasting experience and interaction information at the platform level. The combination of Resource Trust and Platform Reputation enables the calculation of individual *Adaptive Resource Trust* ratings, which provide a more realistic view on offered resources within federations, and allow platforms and their applications to make informed decision in consuming offered resources. Thus, the proposed multi-level approach reduces the negative influence and impact on the overall trust establishment caused by misbehaving third parties or bad-mouthing effects.

V. Conclusion

The paper proposes collaboration mechanisms for orchestrating IoT platform federations facilitating organizational interoperability, which has so far received little attention in literature. The symbIoTe project proposes and implements an original solution for decentralized management of IoT platform federations so as to facilitate direct and secure collaboration between federation members. The solution is in line with a growing demand for strategic partnerships to be supported by interoperability solutions where data and information is exchanged directly in a controlled, meaningful and trustful manner.

The implemented collaboration mechanisms enable inter-organizational communication and fair collaboration connecting heterogeneous, siloed IoT platforms. The introduced IoT federation layer enables the exchange of resources among the platforms within a federation. Specifically, the proposed solution enables access to resources, both sensors and actuators, managed by different platforms in a homogeneous and unified way: For example, an application can search for federated resources as if they were managed by a single platform as well as access sensor data or trigger actuation on federated resources using the same credentials provided for its own original platform. SLA management ensures that QoS policies associated to resources are respected, while bartering mechanism and the needed trust and reputation management are in place to keep federation operations in harmony.

The decentralized federation approach enriched with SLA and trust management as well as the bartering features creates potential for strengthening existing business models in traditional industries and lays the ground for innovative business opportunities and future large-scale IoT deployments. Future work will be directed to monitor and evaluate the proposed mechanisms in practice while orchestrating collaborations between symbIoTe-enabled IoT platform federations.

Acknowledgment

This work is supported by the H2020 symbIoTe project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688156. The authors would like to cordially thank the entire symbIoTe consortium for their valuable comments and discussions.

References

[1] H. van der Veer and A. Wiles, "Achieving Technical Interoperability - the ETSI Approach," ETSI Whitepaper No. 3, 2008.

[2] I. Podnar Zarko et al., "Towards an IoT framework for semantic and organizational interoperability," in *Proc. 2017 Global Internet of Things Summit (GIoTS)*, June 2017, pp. 1–6.

[3] V. M. Tayur and R. Suchithra, "Review of interoperability approaches in application layer of internet of things," in *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, Feb 2017, pp. 322–326.

[4] "AGILE IoT Project," Project datasheet, 2016. [Online]. Available: http://agile-iot.eu/about/

[5] Y. Guan et al., "An open virtual neighbourhood network to connect IoT infrastructures and smart objects - Vicinity: IoT enables interoperability as a service," in *Proc. 2017 Global Internet of Things Summit (GIoTS)*, 2017.

[6] Inter-IoT Consortium, "Requirements and business analysis," Horizon 2020 project deliverable, 2016. [Online]. Available: http://www.inter-iot-project.eu/wp-content/uploads/2016/02/D2.3_INTER-IoT_Requirements-and-business_v1.2.pdf

[7] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective," *Journal of Network and Computer Applications*, vol. 81, pp. 111–124, 2017.

[8] bIoTope Consortium, "bIoTope SoS Reference Platform Specification," Horizon 2020 project deliverable, 2017. [Online]. Available: http://www.biotope-project.eu/results

[9] A. Bröring et al., "Enabling IoT Ecosystems through Platform Interoperability," *IEEE Software*, vol. 34, no. 1, pp. 54–61, 2017.

[10] oneM2M, "M2M Functional Architecture," Technical Specification, 2016. [Online]. Available: http://www.onem2m.org/images/files/deliverables/Release2/TS-0001-%20Functional_Architecture-V2_10_0.pdf

[11] J. Kim et al., "Standard-based IoT platforms interworking: implementation, experiences, and lessons learned," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 48–54, July 2016.

[12] AIOTI WG03 - IoT Standardisation, "High Level Architecture (HLA)," Technical specification, 2017. [Online]. Available: https://aioti.eu/wp-content/uploads/2017/06/AIOTI-HLA-R3-June-2017.pdf

[13] V. Gazis, "A survey of standards for machine-to-machine and the internet of things," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 482–511, Firstquarter 2017.

[14] M. Jacoby, A. Antonic, K. Kreiner, R. Lapacz, and J. Pielorz, "Semantic interoperability as key to iot platform federation," in *LNCS 10218: Interoperability and Open-Source Solutions for the Internet of Things*, 2017, pp. 3–19.

[15] S. Sciancalepore et al., "Attribute-based access control scheme in federated iot platforms," in *LNCS 10218: Interoperability and Open-Source Solutions for the Internet of Things*, 2017, pp. 123–138.

[16] A. Andrieux et al., "Web Services Agreement Specification (WS-Agreement)," Open Grid Forum, 2007. [Online]. Available: https://www.ogf.org/documents/GFD.107.pdf

[17] Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Trust management system design for the internet of things: A context-aware and multi-service approach," *Computers & Security*, vol. 39, pp. 351 – 365, 2013.

[18] T. Eder, D. Nachtmann, and D. Schreckling, "Trust and reputation in the Internet of Things," *Universität Passau, Tech. Rep.*, 2013.