H2020-EUJ-02-2016
H2020 Grant Agreement Number 723076
NICT Management Number18302

Deliverable D5.1

Empowerment Components Definition

Version V1.0
March 30th, 2016

ABSTRACT

This deliverable resumes the initial work of task 5.1 with the objective of identifying the user empowerment components in terms of controlling the access to the information. Such aspects are vital in order to guarantee the security and access control in our platform. So, we firstly identify the functional and non-functional requirements associated to these aspects.

Personal data control is a remarkable point, which has also been dealt in this deliverable by introducing MyData Initiative as a common framework which allow individuals to easily control the way their personal data is published in the digital world, and on the other hand allowing companies to easily gather and process such information.

These requirements allowed us to go to the next step and select the most appropriate technologies for such user empowerment. Hence, we have identified a mechanism for defining flexible privacy policies, access control mechanism, as well as a dynamic consent management mechanism, as the one used by MyData to allow individuals to control their personal data dissemination through a single point access and a common framework. In addition, a different approach has been taken by TRON where they define the concept of a Personal Data Store in which access control mechanisms have been implemented.

Finally, the inclusion of these components in our CPaaS.io architecture is assessed placing them in the corresponding Functional Groups and Functional Components.

This work is licensed under the Creative Commons Attribution 4.0 International License.
To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

Disclaimer

This document has been produced in the context of the CPaaS.io project which is jointly funded by the European Commission (grant agreement n° 723076) and NICT from Japan (management number 18302). All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission and NICT have no liability in respect of this document, which is merely representing the view of the project consortium. This document is subject to change without notice.

Document Information

Editors	Antonio Skarmeta
Authors	Antonio Skarmeta, Juan A. Martínez, Agustín Cánovas, Fahim Khan
Reviewers	Stephan Haller, Toshihiko Yamakami
Delivery Type	R
Dissemination Level	Public
Contractual Delivery Date	March 31, 2017
Actual Delivery Date	March 31, 2017
Keywords	Access Control, Privacy, Authorization, Consent Management, Personal Data Store

Revision History

Rev.	Date	Description	Contributors
0.5	28.3.2017	Final Draft	Juan A. Martinez, Agustín Cánovas, Fahim Khan, Antonio F. Skarmeta
0.9	29.3.2017	Review	Stephan Haller, Toshihiko Yamakami

Table of Contents

1 Introduction 5
2 Requirements Analysis 5
2.1 Requirement Collection 5
2.2 Functional and Non-functional platform requirements 6
3 MyData Model and Citizen Control 9
4 Empowerment Component Technologies 10
4.1 Context-aware privacy policies 10
4.2 Capability-Based Access Control 12
4.3 Dynamic Consent Management 15
4.4 eTRON 19
5 Impact on architecture and platform view 22
5.1 Architecture impact 22
5.2 Platform components for users' empowerment 25
6 Conclusions 28
7 References 29

1 Introduction

This first WP5 deliverable resumes the initial work defined in task 5.1 whose purpose is that of identifying the user empowerment components in terms of controlling the access to the information required in the CPaaS.io architecture. To do so, firstly we have to specify the scope of our platform in terms of requirements (functional and non-functional) so as to provide users with such empowerment.

One remarkable aspect to be aware of is the concern of individuals about how their personal data is published, accessed and commercialized in the digital world. Such concern is considered by MyData Initiative which has also been taken into account in this work. MyData Initiative provides a valuable approach whose target is twofold: on the one hand, allowing the individuals to exercise their right to control the way their personal data is disseminated in the digital world, and in the other hand, providing a common framework for companies to find and process such information.

These two stages allowed us to define the empowerment component technologies required for our CPaaS.io in order to guarantee such control of information. We identify the need to establish context-oriented privacy policies for defining a variety of flexible policies, a capability-based access control mechanism which provide a rich representation of the resource to be access as well as their conditions for such access and the dynamic consent management which has been considered by MyData Initiative to allow individuals to control their personal data dissemination.

Data manipulation also poses specific requirements regarding authentication and access control which has been considered by a Personal Data Store proposed by TRON, whose access control mechanism is based on access control list.

Finally, the inclusion of these components in our CPaaS.io architecture is assessed placing them in the corresponding Functional Groups and Functional Components, as well as how they are integrated with the current components of our CPaas.io platform.

2 Requirements Analysis

While CPaaS.io D3.1 [1] already provided an exhaustive requirements analysis for the project, the main goal of this deliverable is to give an overview about how the main security and privacy requirements are addressed through specific components. Before the explanation about technologies and platform instantiation addressing such needs, below a summary of main requirements about users' empowerment is provided.

2.1 Requirement Collection

CPaaS.io requirements were mainly obtained by three sources: scenario, platform and business:

- **Scenario requirements:** they focus on the set of specific needs and expectations that must be addressed by the CPaaS.io platform, in order to realize a particular scenario (e.g. regarding properties, functionalities and interfaces). The main goal of such scenarios is to address both functional aspects (i.e. functionalities that are needed to implement the scenario story), and non-functional aspects, such as performance, deployment or scalability issues; those requirements should reflect end-users concerns as well.

- **Platform requirements:** they focus on the technical objectives that must be supported by the CPaaS.io platform in order to realize different use cases
- **Business requirements:** requirements that stem from business and/or exploitation perspectives;

In addition to these sources, CPaaS.io requirements have been mainly classified as functional requirements (i.e. related to the functionality that the system must provide), and non-functional requirements (i.e. related to the properties that should be provided by the system). Furthermore, this classification includes run-time and non-run-time categories. With these considerations, below we provide the list of CPaaS.io requirements related to security, privacy and trust, which have been used as the baseline to define the main functional and non-functional aspects of the platform.

2.2 Functional and Non-functional platform requirements

2.2.1 Functional requirements

Regarding functional requirements, CPaaS.io provides different categories that have been used to identify the main platform needs for users' empowerment:

- **Load/network resource balancing**. Different city actors should be provided with mechanisms to take into account these privacy concerns. Furthermore, decision making processes could be based on trustworthiness level of network nodes.
- **Discovery and Look-up**. Information should be granted according to proper access credentials and current context. Furthermore, semantics aspects must be considered to provide a more fine-grained discovery approach.
- **Controlled Access to Object (Virtual Entities), Data and Services**. Users must be empowered with approaches to govern how their devices' data can be accessed
- **Accounting**. While privacy aspects must be taken into account, users tracking must be enabled for eventual billing purposes.
- **Accountability**. The platform must provide some mechanisms in order to support non-repudiation and traceable anonymity.
- **Control Right and Privacy**. Data owners must be able to specify and manage their access policies, which must be additionally protected. Such access control preferences could be ignored in case of emergency or exceptional situations. Privacy aspects must be addressed through the possibility to use pseudonyms when accessing different services.
- **Data Queries**. Different metadata could be specified when searching and retrieving data, by including trust and reputation scores of the device resources in charge of generating such data.

Table 1 shows the specific *functional* requirements related to users' empowerments for security and privacy aspects are provided. It should be noted that this table represents a subset of the set of CPaaS.io requirements, which were provided in D3.1 [1].

ID #	Description	Priority	Category
FREQ.2 [REQ31]	The look-up service of CPaaS.io shall withhold or grant information depending on context such as application involved, requesting entity, and security permissions.	MUST	Discovery and Look-up
FREQ.5 [REQ38]	The IoT system shall enable the lookup of service descriptions of specified services for Virtual Entities with the VE identifier as key for the lookup.	MUST	Discovery and Look-up
FREQ.11	Data search and retrieval shall be achieved upon a collection of criteria that includes location (of the Physical entity concerned), characteristics of the underlying IoT Resource reputation and the quality of information.	MUST	Data Queries
FREQ.20 [REQ19]	User anonymity shall be provided in order to enforce privacy (e.g. at communication level)	MUST	Control Right and Privacy Accounting
FREQ.21 [REQ81]	Personal data in servers should be ciphered by a private key.	SHOULD	Security/Privacy & Trust
FREQ.22 [REQ84]	Secure storage of data should be ensured.	SHOULD	Security/Privacy & Trust
FREQ.23 [REQ34]	Data owners shall be able to set access-control rights/ policies (set up by data owners) to their data stored on resources.	MUST	Control Right and Privacy
FREQ.24 [REQ35]	Access-control rights/ policies (set up by data owners) should not be published publicly.	SHOULD	Control Right and Privacy
FREQ.25 [REQ63]	Communicated data shall remain confidential.	MUST	Control Right and Privacy
FREQ.26	Service providers shall be able to set access-control rights/ policies (set up by service owners) to their services	MUST	Control Right and Privacy
FREQ.30 [REQ116]	CPaaS.IO shall be able to evaluate access request depending on access control policies.	MUST	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.31 [REQ117]	CPaaS.IO shall provide an authorization policy to data, object and service for the different users or devices.	MUST	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.32 [REQ 59]	User or device as part of the platform deployment shall be registered before using any services provided by the platform.	MUST	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.33 [REQ60]	User or device shall be identified.	MUST	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.34 [REQ62]	User or device shall be authorized to use a service.	MUST	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.35 [REQ69]	Objects should authenticate a person or object that try to access its data.	SHOULD	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.36 [REQ77]	Authentication of user or service shall be carried out by devices before delivering data.	MUST	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.37 [REQ 82]	Authentication of user or service shall be carried out to access a service.	MUST	Controlled Access to Object (Virtual Entities), Data and Services

FREQ.38 [REQ123]	User's data shall be processed only with the user's consent.	MUST	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.39 [REQ133]	The IoT system should support context-aware access policies.	MUST	Controlled Access to Object (Virtual Entities), Data and Services
FREQ.50 [REQ90]	Services shall maintain a log of the operations done by users. This log shall not be modified by attackers.	MUST	Accountability
FREQ.70 [REQ131]	The IoT system shall ensure that processing of information takes place on trusted nodes.	MUST	Load/network resource balancing

Table 1. Functional security, privacy and trust CPaaS.io requirements for users' empowerment

2.2.2 Non-functional requirements

For non-functional requirements, and grouping run-time and non-run-time CPaaS.io platform needs, the classification related to users' empowerment requirements can be done as:

- **Trust**: they are related to reliability, as well as trust and reputation scores regarding platform services.
- **Security**: they are implied to fulfil confidentiality, integrity and availability (CIA triad) aspects, as the baseline to build a more trustworthy platform.
- **Privacy**: data minimization related to *Personally Identifiable Information* (PII) must be supported, as well as the use of privacy-preserving identity management schemes when accessing platform services.

Table 2 shows the specific requirements related to users' empowerments for security and privacy aspects are provided. It should be noted that this table represents a subset of the set of CPaaS.io requirements, which were provided in D3.1.

ID #	Description	Perspective	Priority	Category
NFREQ.9	Expose data and services to authorized users	Security	MUST	City Data, Platform
NFREQ.10	Ensure services are always accessible to entitled users	Security	MUST	City Data, Platform
NFREQ.11	Ensure data freshness	Security	MUST	Platform
NFREQ.12	Support access control mechanisms	Security	MUST	Platform
NFREQ.13	Have security mechanisms to protect data transmission	Security	SHOULD	Platform, Infrastructure
NFREQ.14	Make it difficult to spy on communicated messages	Security	SHOULD	Platform, Infrastructure
NFREQ.15	Be able to perform to detect threats at runtime	Security	SHOULD	Platform, Infrastructure
NFREQ.16	Provide trusted and secure communication and information management	Trust	SHOULD	Platform, Infrastructure
NFREQ.17	The platform infrastructure and services shall be trustable	Trust	SHOULD	Infrastructure
NFREQ.18	Allow users to use free services anonymously	Privacy	SHOULD	Societal Needs, Platform
NFREQ.19	Allow people to use free services anonymously	Privacy	SHOULD	Societal Needs, Platform

NFREQ.20	Allow users to control which data they are willing to provide and how their data should be used	Privacy	SHOULD	Societal Needs, Platform
NFREQ.21	Keep users access-control rights/ policies secured.	Privacy	SHOULD	Platform
NFREQ.22	Provide privacy protection for users interacting with the platform	Privacy	SHOULD	Societal Needs, Platform
NFREQ.23	Provide communication confidentiality	Privacy	SHOULD	Platform, Infrastructure

Table 2. Non-functional security, privacy and trust CPaaS.io requirements for users' empowerment

Both functional and non-functional requirements related to users' empowerment for security and privacy are used as the baseline for the specification of CPaaS.io platform components. Next sections provide an overview of specific technologies that are to be instantiated by different platform's components, in order to address such requirements.

3 MyData Model and Citizen Control

The use of personal data has become a world-wide mainstream business activity in the last years. Its value is such that according to The World Economic Forum, "Personal data is becoming a new economic asset class, a valuable resource for the 21st century that will touch all aspect of society" [2]. As a matter of fact, in a survey provided by Accenture where nearly 600 business around the world were responding, 79 percent of them affirmed to collect data directly from individuals (thanks to the customer account, for instance), as well as from connected devices and third-party data suppliers.

By contrast, such businesses are usually accompanied with a control loss by the individuals which have little or no knowledge over how data about them and their activities is created or used. In fact, nowadays, individuals grant legal consent to organizations and software applications for the collection and use of their personal data by accepting the terms of the service they use without understanding them or even without reading them due to their length and complexity.

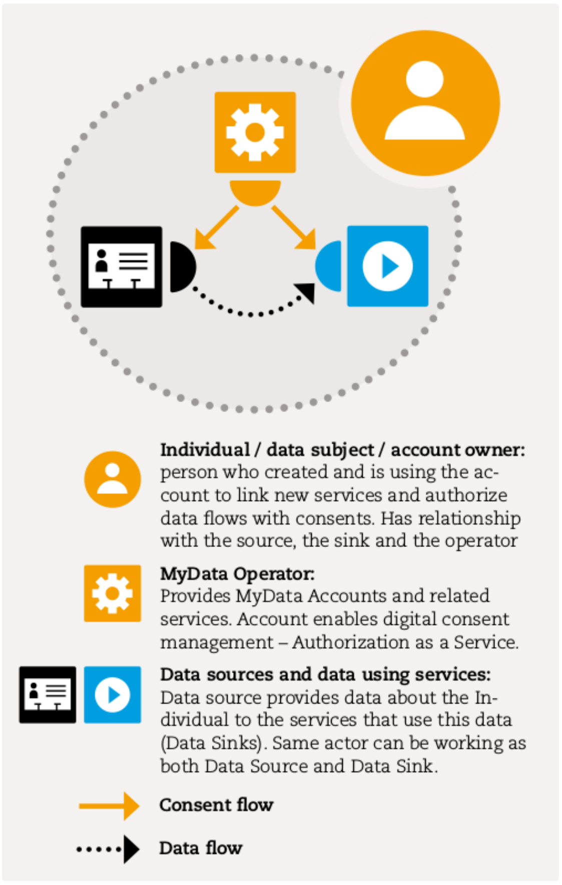
As in real life, in our digital life individuals should have legal rights and technical tools to manage personal data collected about them. This is an extension to the freedom of thought and expression that we all have as citizens.

On the other hand, the actual protection regulations prevent companies to create innovative services around personal data so they resort to ways to bypass them. In order to solve this situation MyData initiative has been proposed [3], encompassing a framework, principles and a model for a human-centric approach to empower individuals about their personal data management and processing.

The fundamental principles on which the MyData initiative is based are as follows:

1. **Human centric control and privacy**: Individuals are no longer passive targets, but empowered actors in the management of their personal lives both online and offline.
2. **Usable data**: It is essential that personal data is technically easy to access and use – it is accessible in machine readable open formats via secure, standardized APIs (Application Programming Interfaces).
3. **Open business environment**: A shared MyData infrastructure enables decentralized management of personal data, improves interoperability, makes it easier for companies to comply with tightening data protection regulations, and allows individuals to change service providers without proprietary data lock-ins.

MyData is a progressive approach to personal data management that combines digital human rights and industry need to have access to data. This approach benefits both sides: individuals and companies. For individuals, it provides easy-to-use and comprehensive tools for personal data management and transparency mechanisms that openly show how organizations use their data. For companies, it opens opportunities for new kinds of data-based businesses by facilitating the legal and technical access to pre-existing personal datasets when the individual is willing to give his/her consent. And in addition, also for the civil society, it creates the necessary structures, processes and policies for protecting the rights of individuals and fostering the use of personal data in the development of innovative services.


Individual / data subject / account owner: person who created and is using the account to link new services and authorize data flows with consents. Has relationship with the source, the sink and the operator

MyData Operator: Provides MyData Accounts and related services. Account enables digital consent management – Authorization as a Service.

Data sources and data using services: Data source provides data about the Individual to the services that use this data (Data Sinks). Same actor can be working as both Data Source and Data Sink.

Consent flow

Data flow

Figure 1. MyData architecture [3]

The architecture proposed by the MyData initiative is based on interoperable and standardized MyData accounts which serves as Personal Data Stores (PDS). The proposed model allows individuals to control their personal data from a single place in an easy way. Such accounts will be provided by organizations that act as MyData operators, giving also the possibility to individuals to host their own accounts.

The data flows from a data source to a service or application that uses the data. It is worth mentioning that the flow of consents or permissions is separate from the actual flow of data as described in Figure 1. Actually, the primary function of a MyData account is to enable consent management – the data itself is not necessarily streamed through the servers where the MyData account is hosted.

Finally, the representation of the consent management can be developed using the open consent meta-format (Kantara Initiative) which is also detailed below in Section 4.3.

4 Empowerment Component Technologies

In this section, we give some details and possible solutions to cope with the requirements listed in the previous sections.

4.1 Context-aware privacy policies

Given the pervasive, distributed and dynamic nature of IoT, context should be a first-class security component in order to drive the behavior of devices. This would allow smart objects to be enabled with context-aware security solutions, in order to make security decisions adaptive to the context in which transactions are performed. At the same time, context information should be managed by taking into account security and privacy considerations. In particular, current IoT devices (e.g. smartphones) can obtain context information from other entities of their surrounding environment, as well as to provide contextual data to other smart objects. These communications can be performed through lossy networks and constrained devices, which must be secured by suitable security mechanisms and protocols.

Additionally, trust and reputation mechanisms should be employed to assess the trustworthiness of data being provided by other entities in the environment. In this way, smart objects can discard information that comes from less reliable devices. Moreover, high-level context information can be reasoned and inferred by considering privacy concerns. Thus, a smartphone could be configured to provide information about a person's location with less granularity (e.g. giving the name of the city where he is, but not the GPS coordinates), or every long periods of time in order to avoid that daily habits of a person could be inferred by other entities.

In order to make security adaptive to context, CPaaS.io will provide a platform to enable users and smart objects to share information by maintaining different interacting entities to be uncoupled. In this sense, the resulting platform will be used for secure exchange of contextual data, so users and smart objects could adapt their security and privacy behaviour according to it. For example, this information could be used by an *Identity Management* component, which is intended to manage the identities of users and smart objects in an (optionally) privacy-preserving way. It is based on the use of anonymous credentials systems (e.g. Idemix[7]), enabling users and smart objects to select partial identities [8] (i.e. a subset of their identity attributes) to be used when accessing to platform services. In order to accomplish the envisioned functionality, this component is composed of another subcomponent. Specifically, a repository of privacy rules can be used to define privacy preferences of users for a proper selection of their partial identities based on their current context conditions (e.g. time, location). An example privacy policy could be "IF contextA=atWork AND contextB=workinghours, THEN partialIdentity=seniorResearcher", specifying a single attribute to be used for transactions carried out under those context conditions. While the antecedent of the rule specifies high-level context conditions, the consequent indicates which partial identity has to be deployed (among the ones already defined by the user). To this aim, there is a mapping between the partial identities defined in the rules and the identity credentials that represent such partial identities (e.g. Idemix proofs) that are managed by the credential manager subcomponent. The evaluation of these rules could be used, in turn, by high level graphical applications that can facilitate users to manage their partial identities under different contextual conditions. It should be pointed out that privacy policy of a user or smart object could select a different partial identity to the identity that is required by platform services-. In order to solve this conflicting situation, an identity negotiation process could be considered, in the case the service requires more identity attributes from the user than he wants to disclose in the current situation. In this regard, a comparison from the service's and user's privacy policy could be used in order to suggest the best partial identity to be adopted.

In addition to identity management, authorization functionality could be also based on contextual information. Specifically, as will be described in the next section, the CPaaS.io access control approach is based on a combination of access control models and techniques. Specifically, it makes use of access control policies (e.g. eXtensible Access Control Markup Language (XACML) 3.0)[9] , which are employed to generate lightweight authorization tokens based on the capabilities based approach [10][8], which will be explained in the next section. For a more fine-grained authorization mechanism in the IoT, contextual information is a key aspect to be considered when making access control decisions. In the same way that contextual information can be considered for partial identities selection, this component is expected to be deployed into services or devices, in order to drive the access control logic to protect resources being accessed. In this way, capability tokens could contain contextual conditions to be enforced by the service being accessed (e.g. related to current time or location). Therefore, when a user tries to get access to a resource being hosted in the service, the token could include restrictions related to the context (e.g. "IF distance<1meter") to be locally verified when the token is evaluated by the target device.

In addition to more established token-based access control approaches, there will be common situations in which information needs to be outsourced or shared through the use of a central data management platform. For these scenarios, an approach based on advanced cryptographic schemes, such as the Ciphertext Policy Attribute Based Encryption (CP-ABE) scheme [11], is key to guarantee security properties when this data needs to be shared with groups of user or services. In this case, the high-level context information could be used to select a specific CP-ABE policy to encrypt a certain piece of data. In particular, this component could contain a set of *Sharing Policies* specifying how the information should be disseminated according to contextual data. These policies are intended to be evaluated before information is disseminated by the smart objects. The result of the evaluation of these policies could be, in turn, a CP-ABE policy indicating the set of entities which will be enabled to decrypt the information to be shared. An example sharing policy could be "IF contextA=atPub AND data=myLocation, THEN CP-ABE policy=myfriends OR myfamily", specifying the location of a user is shared with friends or family members when he is at a pub. According to it, when a policy is successfully evaluated, the resulting CP-ABE policy is used to encrypt the information to be shared. In the case of two or more sharing policies are successfully evaluated, the most restrictive CP-ABE policy could be selected. After the information is encrypted and disseminated, this component of smart objects receiving such data will try to decrypt it with CP-ABE keys related to its identity attributes. It should be noted that the use of such approach could be integrated into end devices (e.g. smartphones) that will share their data with other users through the platform. At the same time, such approach could be included into the platform, in case other devices (e.g. sensors) are not able to deploy this mechanism.

4.2 Capability-Based Access Control

The inherent requirements and constraints of IoT environments, as well as the nature of the potential applications of these scenarios, have brought about a greater consensus among academia and industry to consider access control as one of the key aspects to be addressed for a full acceptance of all IoT stakeholders.

Due to heterogeneous nature of IoT devices and networks, most of recent access control proposals have been designed through centralized approaches in which a central entity or gateway is responsible for managing the corresponding authorization mechanisms, allowing or denying requests from external entities. Since this component is usually instantiated by unconstrained entities or back-end servers, standard access control technologies are directly applied. However, significant drawbacks arise when centralized approaches are considered on a real IoT deployment. On the one hand, the inclusion of a central entity for each access request clearly compromises end-to-end security properties, which are considered as an essential requirement on IoT, due to the sensitivity level of potential applications. On the other hand, the dynamic nature of IoT scenarios with a potential huge number of devices complicates the trust management with the central entity, affecting scalability. Moreover, access control decisions do not consider contextual conditions which are locally sensed by end devices.

These issues could be addressed by a decentralized approach, in which IoT devices (e.g. smartphones, sensors, actuators, etc.) are enabled with authorization logic without the need to delegate this task to a different entity when receiving an access request. In this case, end devices are enabled with the ability to obtain, process and transmit information to other entities in a protected way. However, in a fully distributed approach, the feasibility of the application of traditional access control models, such as RBAC or ABAC, has not been demonstrated so far. Indeed, as previously mentioned, such models require a mutual understanding of the meaning of roles and attributes, as well as complex access control policies, which makes challenging the application of them on IoT devices. Moreover, the impact of the potential applications of IoT in all aspects of our lives is shifting security aspects from an enterprise-centric vision to a more user-centric one. Therefore, usability is a key factor to be considered, since untrained users should be able to control how their devices and data are shared with other users and services.

As already mentioned, DCapBAC has been postulated as a feasible approach to be deployed on IoT scenarios [12] even in the presence on devices with tight resource constraints. Inspired by SPKI Certificate Theory [13] and ZBAC foundations [14], it is based on a lightweight and flexible design and that allows authorization functionality is embedded on IoT devices, providing the advantages of a distributed security approach for IoT in terms of scalability, interoperability and end-to-end security. The key element of this approach is the concept of capability, which was originally introduced by [15] as "token, ticket, or key that gives the possessor permission to access an entity or object in a computer system". This token is usually composed by a set of privileges which are granted to the entity holding the token. Additionally, the token must be tamper-proof and unequivocally identified in order to be considered in a real environment. Therefore, it is necessary to consider suitable cryptographic mechanisms to be used even on resource-constrained devices which enable an end-to-end secure access control mechanism. This concept is applied to IoT environments and extended by defining conditions which are locally verified on the constrained device. This feature enhances the flexibility of DCapBAC since any parameter which is read by the smart object could be used in the authorization process. DCapBAC will be part of the CPaaS access control system and integrated with Policy-based approach by using XACML, in order to infer the access control privileges to be embedded into the capability token.

4.2.1 Capability Token

The format of the capability token is based on JSON. Compared to more traditional formats such as XML, JSON is getting more attention from academia and industry in IoT scenarios, since it is able to provide a simple, lightweight, efficient, and expressive data representation, which is suitable to be used on constrained networks and devices. As shown below, this format follows a similar approach to *JSON Web Tokens* (JWTs)[16], but including the access rights that are granted to a specific entity.

Figure 2 shows a capability token example. Below, a brief description of each field is provided.

- Identifier (ID). This field is used to unequivocally identify a capability token. A random or pseudorandom technique will be employed by the issuer to ensure this identifier is unique.
- Issued-time (II). It identifies the time at which the token was issued as the number of seconds from 1970-01-01T0:0:0Z.
- Issuer (IS). The entity that issued the token and, therefore, the signer of it.
- Subject (SU). It makes reference to the subject to which the rights from the token are granted. A public key has been used to validate the legitimacy of the subject. Specifically, it is based on ECC, therefore, each half of the field represents a public key coordinate of the subject using *Base64*.
- Device (DE). It is a URI used to unequivocally identify the device to which the token applies.
- Signature (SI). It carries the digital signature of the token. As a signature in ECDSA is represented by two values, each half of the field represents one of these values using *Base64*.
- Access Rights (AR). This field represents the set of rights that the issuer has granted to the subject.
  - Action (AC). Its purpose is to identify a specific granted action. Its value could be any CoAP method (GET, POST, PUT, DELETE), although other actions could be also considered.
  - Resource (RE). It represents the resource in the device for which the action is granted.
  - Condition flag (F). Following the notation in [17], it states how the set of conditions in the next field should be combined. A value of 0 means AND, and a value of 1 means OR.
  - Conditions (CO). Set of conditions which have to be fulfilled locally on the device to grant the corresponding action.
    - Condition Type (T). The type of condition to be verified as stated by [17].
    - Condition value (V). It represents the value of the condition.
    - Condition Unit (U). It indicates the unit of measure that the value represents. Its value could be any of those stated by [18].
  - Not Before (NB). The time before which the token must not be accepted. Its value cannot be earlier than the II field and it implies the current time must be after or equal than NB.
  - Not After (NA). It represents the time after which the token must not be accepted.

```
{
      “id”: “eg3fq:fb5r23tra3”,
      “ii”: 1485172121,
      “is”: “issuer@odins.es”,
      “su”: “zNwS5FetB4rwzSKsWwSBAxm5wDa=JgLjHU8zSnmeSFQgSG9HhdsJrE8=”,
      “de”: “coap://sensortemp.floor1.computersciencefaculty.um.es”,
      “si”: “SbUudG4zuXswFBxDeHB87N6t9hR=PBQqCN3gpu7nSkuPzDk7kaR3dq1=”,
      “ar”: [
            {
                  “ac”: “GET”,
                  “re”: “temperature”
            }
      ],
      “nb”: 1485172121,
      “na”: 1485174121
}
```

Figure 2. Capability Token Example

4.2.2 DCapBAC scenario

In a typical DCapBAC scenario, an entity (*subject*) tries to access a resource of another entity (*target*). Usually, a third party (*issuer*) generates a token for the subject specifying which privileges it has. Thus, when the subject attempts to access a resource hosted in the target, it attaches the token which was generated by the issuer. Then, the target evaluates the token granting or denying access to the resource. Therefore, a subject which wishes to get access certain information from a target, requires sending the token together the request. Thus, the target device that receives such token can know the privileges (contained in the token) that the subject has and it can act as a Policy Enforcement Point (PEP). This simplifies the access control mechanism, and it is a relevant feature on IoT scenarios since complex access control policies are not required to be deployed on end devices.

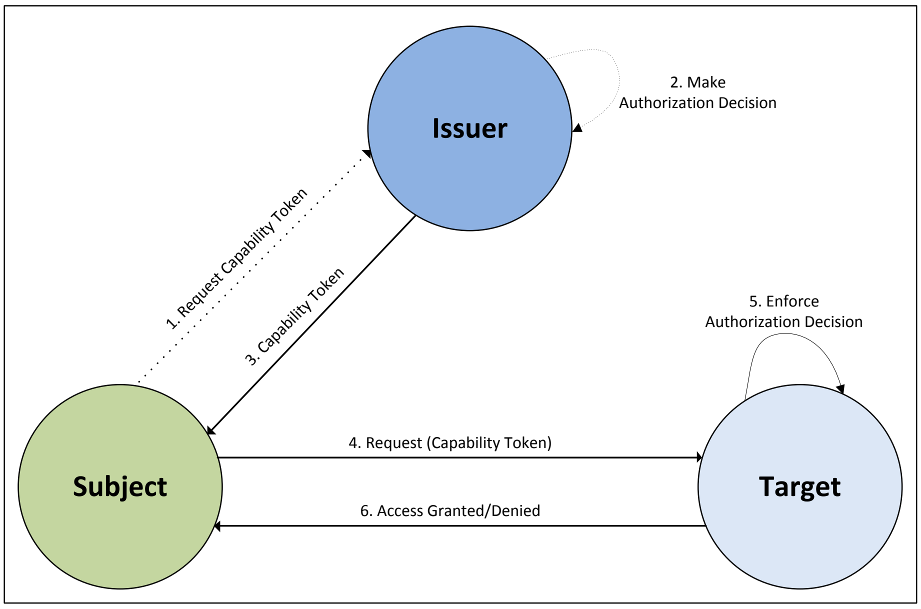

Issuer
2. Make Authorization Decision
1. Request Capability Token
3. Capability Token
5. Enforce Authorization Decision
4. Request (Capability Token)
6. Access Granted/Denied
Subject
Target

Figure 3. DCapBAC scenario overview

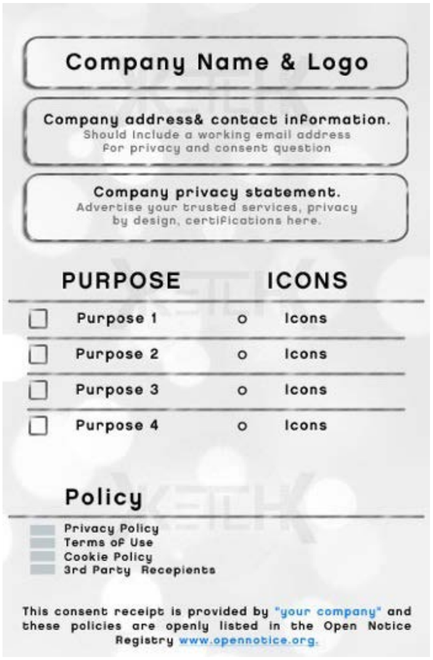
The basic operation of DCapBAC is shown in Figure 3. As initial step, the *Issuer* entity, which could be instantiated by the device owner or another entity in charge of the smart object, issues a capability token to the *Subject* to be able to access such device. Additionally, in order to avoid security breaches, such token is signed by the *Issuer*. In the SocIoTal access control system, this process is based on the use of XACML policies. Therefore, in the case of a “Permit” decision, a capability token is generated with that specific privilege. In addition, XACML Obligations can be used in order to embed contextual conditions to be locally verified by the target device. Once the *Subject* has received the capability token, it attempts to access the device data. For this purpose, a request is generated (e.g. by using CoAP), in which the token is attached. According to Figure 3, this request does not have to be read by any intermediate entity. When the *Target* receives the access request, the authorization process is carried out. First, the application checks the validity of the token (i.e. if it has expired) as well as the rights and conditions to be verified. Then, the *Issuer* signature is verified with the corresponding public key. Depending on the specific scenario, this key can be delivered to smart objects during the commissioning or manufacturing process, or it can be recovered from a predefined location. Finally, once the authorization process has been completed, the *Target* generates a response based on the authorization decision.

Additionally, this approach provides support for advanced features, such as *access delegation*. In this case, a subject S (acting as a *delegator)* with a capability token CT can generate another token CT′ for S′ (acting as a *delegated*), in which a subset of the privileges of CT are embedded. Consequently, CT′ can be used by S′ to get access to a resource in a target smart object. Furthermore, S can grant the right to S′ for additional delegations. This feature is valuable in order to address the dynamic and pervasive nature of IoT scenarios and everyday life. For example, elderly people can provide temporary privileges or delegation of them to home help personnel to get access to their homes in case of an emergency situation. In the case of delegation, it is necessary to sign each new capability token with the corresponding subset of privileges, in order to allow a full auditability of access and avoid security breaches.

4.3 Dynamic Consent Management

There is another aspect worth highlighting in this section. It is the need for providing a mechanism that allows citizens to give explicit consent on some actions like for instance revealing private information, or enrolling in a service. In this sense, the Consent & Information Sharing Work Group (CISWG), thanks to Kantara Initiative, (https://kantarainitiative.org/groups/ciswg/) is aimed at gathering information of the different use cases and scenarios that illustrate the exchange of such information so as to specify the policy and technology enablers that should be selected to enable this information to flow.

As a result of such actions they have developed the concept of ***Consent Receipt***, (Figure 4). “A record of a consent provided to an individual at the point in a person agrees to the sharing of personal information. Its purpose is to capture the privacy policy and its purpose for sharing personal information so it can be easily used by people to communicate and manage consent and sharing of personal information once it is provided”.



Company Name & Logo

Company address& contact information.
Should include a working email address
for privacy and consent question

Company privacy statement.
Advertise your trusted services, privacy
by design, certifications here.

	PURPOSE		ICONS
☐	Purpose 1	o	Icons
☐	Purpose 2	o	Icons
☐	Purpose 3	o	Icons
☐	Purpose 4	o	Icons

Policy

▨ Privacy Policy
▨ Terms of Use
▨ Cookie Policy
▨ 3rd Party Recepients

This consent receipt is provided by "your company" and these policies are openly listed in the Open Notice Registry www.opennotice.org.

Figure 4. Consent Receipt concept

This record enables tracking consents as regular receipts are used to track money. But as the latter does, it is needed a common and standard format for consent notice. This aspect is not neglected by the CISWG which, in order to encourage and promote the use of the Consent Receipt, has deployed a website (http://api.consentreceipt.org/doc/) where a server for receipt generation has been deployed and a complete API for its use is presented. The following subsection thoroughly describes it.

4.3.1.1 Consent Receipt API

The endpoint accepts HTTP POST request receiving as input a JSON format text. As output, it provides a signed JSON Web Token representing the Consent Receipt.

In the following table the top-level fields that should be added to the input JSON object are presented.

Field name	Data Type	Description	Example input	Required
Section 1: CR Header	---	**This is the first section of the receipt**	---	
`jurisdiction`	string. ISO two-letter country code if applicable, otherwise free text	This is the legal jurisdiction under which the processing of personal data occurs	`US`	✓
`iat`	number. Integer number of seconds since 1970-01-01 00:00:00 GMT	Timestamp of when the consent was issued	`1435367226`	✓
`moc`	Method of collection	Is used to describe how the consent was collected i.e. webform opt in, or implicit, verbal, etc	`web form`	✓
`iss`	string. HTTPS URL	This is the URI or Internet location of processing, i.e., one party-two party or three	`http://www.consentreceipt.org/`	✓
`jti`	string	Unique identifier for this consent receipt	`9ef6b81a414b2432ec6e3d384c5a36cea8aa0c30d3dd2b67364126ed80856f9c20654f032eef87ad981187da8c23c1186eefe1503714835c2e952bbb3f22729c`	✓
`sub`	string	Subject provided identifier, email address - or Claim, defined/namespaced	`example@example.com`	✓

Field name	Data Type	Description	Example input	Required
Section 2: Data Controller	---	**This section has the data controller, contact and privacy service information**	---	
`data_controller`	object	The identity and company of the data controller and any party nominated to be data controller on behalf of org The object contains information of the data controller in the following fields:	`{"on_behalf" : true, "contact": "Dave Controller", "company": "Data Controller Inc.", "address": "123 St., Place", "email": "dave@datacontroller.com", "phone": "00-123-341-2351"}`	✓

Field Name	Data Type	Description	Example Input
`on_behalf`	boolean.	acting on behalf of an organization?	`true`
`contact`	string.	person to contact	`Jon Doe`
`company`	string.	company name	`Data Controller Inc.`
`address`	string.	physical address	`123 Main St., Anywhere`
`email`	string. Email	contact email address	`jon@datacontroller.com`

		address		
		phone	string. Phone number	contact phone number	00-000-000-0000

Policy_uri	String. HTTP URL	The internet and immediately accessible privacy policy of the service referred to by the receipt	http://example.com/privacy	✓

Field name	Data Type	Description	Example input	Required
Section 3: Purpose Specification	---	**List Purpose**	---	
purpose	Array of string's arrays.	Explicit, Specific and Legitimate: interpreted here as: 'Naming the Service' and 'Stating the Active Purpose' see Appendix A these requirements	[Bob's store, delivery,]or [["CISWG Membership", "Join"]]	✓

Field name	Data Type	Description	Example input	Required
Section 4: Purpose Specification	---	**List 3rd Party Sharing Activities**	---	
sensitive	Array of string's arrays.	In many jurisdictions, there are additional notice and administrative requirements for the collection, storage and processing of what are called Sensitive Personal Information Categories. These are Sensitive in the business, legal, and technical sense, but not specifically in the personal context. This list of categories is required in some jurisdiction, but, the actual notice and purpose requirements are out the scope of the MVCR.	["health"]	✓

Field name	Data Type	Description	Example input	Required
Section 5: Information Sharing	---	**Sharing information with 3rd parties, what categories, with whom, and how information is shared**	---	
sharing	Object	This refers to the sharing of personal information collected about the individual, with another external party by the data controller (service provider). Should list categories of PII shared, from above list and under what purpose. Sharing is also a container for listing trust marks and trust protocols. The object contains information of the sharing in the following fields:	{party_name: "3rd Party Name or/3rd Party Category"}	✓

Field Name	Data Type	Description	Example Input
sharing	array of strings.	Data categories to share	None
party_name	string.	3rd party to share data	3rd Party Name or/3rd Party Category
purpose	string.	How information is shared	None

Field name	Data Type	Description	Example input	Required
Section 6: Optional Or In Review	---	---	---	

notice	String. HTTP URL	Link to the short notice enables usability and layered policy to provide enhanced transparency about data collection and information sharing practices	http://example.com/shortnotice	✓
scopes	String. Space separated string values	What you are allowed to do on the service (these can be tied to legal/business/technical layers)	read update	

Following these fields, a possible JSON object that could be sent to the server could be like the following one.

```
 {
 "jurisdiction" : "US",
 "iat": 1443282118,
 "moc": "web form",
 "iss": "http://www.consentreceipt.org/",
 "jti": "cba37edd4e223a44ea0197498663af81c0d68cdf7b5f13975096e34435339e51f86b6bf674f9725632b6f451b4a78c2fb09d3fcd38
c978f004fcf99e65bdceab",
 "sub" : "example@example.com" ,

 "data_controller" : {"on_behalf": true, "contact": "Dave Controller", "company": "Data Controller Inc.", "address"
: "123 St., Place", "email": "dave@datacontroller.com", "phone": "00-123-341-2351"},
 "policy_uri" : "http://example.com/privacy" ,

 "purpose" : [["Bob's Store", "delivery", "financial"]],

 "sensitive" : ["health"] ,

 "sharing" : {sharing:"financial",party_name: "demographic", purpose: "delivery"},

 "notice" : "http://example.com/shortnotice" ,
 "scopes" : "read update"
 }
```

The corresponding output is, as we already commented above a signed JWT.

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdXJpc2RpY3Rpb24iOiJVUyIsIm1vYyI6IndlYiBmb3JtIiwic3ViIjoiZXhhbXBsZ
UBleGFtcGxlLmNvbSIsIm5vdGljZSI6Imh0dHA6Ly9leGFtcGxlLmNvbS9zaG9ydG5vdGljZSIsInBvbGljeV91cmkiOiJodHRwOi8vZXh
hbXBsZS5jb20vcHJpdmFjeSIsImRhdGFfY29udHJvbGxlciI6eyJvbl9iZWhhbGYiOnRydWUsImNvbnRhY3QiOiJEYXZlIENvbnRyb2xsZ
XIiLCJjb21wYW55IjoiRGF0YSBDb250cm9sbGVyIEluYy4iLCJhZGRyZXNzIjoiMTIzIFN0LiwgUGxhY2UiLCJlbWFpbCI6ImRhdmVAZGF
0YWNvbnRyb2xsZXIuY29tIiwicGhvbmUiOiIwMC0xMjMtMzQxLTIzNTEifSwicHVycG9zZSI6W1siQm9iJ3MgU3RvcmUiLCJkZWxpdmVye
SIsImZpbmFuY2lhbCJdXSwic2Vuc2l0aXZlIjpbImhlYWx0aCJdLCJzaGFyaW5nIjp7InNoYXJpbmciOlsiZmluYW5jaWFsIl0sInBhcnR
5X25hbWUiOiJkZW1vZ3JhcGhpYyIsInB1cnBvc2UiOiJkZWxpdmVyeSJ9LCJzY29wZXMiOiJyZWFkIHVwZGF0ZSIsImp0aSI6ImNiYTM3Z
WRkNGUyMjNhNDRlYTAxOTc0OTg2NjNhZjgxYzBkNjhjZGY3YjVmMTM5NzUwOTZlMzQ0MzUzMzllNTFmODZiNmJmNjc0Zjk3MjU2MzJiNmY
0NTFiNGE3OGMyZmIwOWQzZmNkMzhjOTc4ZjAwNGZjZjk5ZTY1YmRjZWFiIiwiaWF0IjoxNDQzMjgyMTE4LCJpc3MiOiJodHRwOi8vd3d3L
mNvbnNlbnRyZWNlaXB0Lm9yZy8ifQ.LNY1NdOQg06iI003Mbi56_cnzd3VY7_hO6sn79z65OPXbEU06Budr8juV9HR_EHSCq9C5ungou02
b2r15Imp7beIkXJzoVZMdX-_nK-
BwaP4hu128TabCUkMAYq0Egk2IQVJV4tsrAjJMbC_l8rE8UDpWDPPNSoV40PCR12_vYeuvTn6Pe8LL9xwcPX0Gz57amqrp4bcs_MUaVfL6
L6QH7cPv3MZAnSWBrgGevcQh6m0X0b4jonasyr63falMl3AlCSzSZgwf33ZaPoH8Ioo6zMPEgTtw0EWnSVSBl8Tp06KAqdhFbZ0SPg6DSQ
oGcNS-vihJDDqmsV_gLv1RmFqQQ

The header portion of the JWT contains:

```
 {
 "alg": "RS256",
 "typ": "JWT"
 }
```

And its corresponding payload is the following:

```
 {
 data_controller: {
 address: "123 St., Place",
 company: "Data Controller Inc.",
 contact: "Dave Controller",
 email: "dave@datacontroller.com",
 on_behalf: true,
 phone: "00-123-341-2351",
 },
 iat: 1443282118,
 iss: "http://www.consentreceipt.org/",
 jti: "cba37edd4e223a44ea0197498663af81c0d68cdf7b5f13975096e34435339e51f86b6bf674f9725632b6f451b4a78c2fb09
d3fcd38c978f004fcf99e65bdceab",
 jurisdiction: "US",
 moc: "web form",
 notice: "http://example.com/shortnotice",
 policy_uri: "http://example.com/privacy",
 purpose:[["Bob's Store", "delivery", "financial"]],
 scopes: "read update",
 sensitive: ["health"],
 sharing: {
 party_name: "demographic",
 purpose: "delivery",
 sharing: ["financial"]
 },
 sub: "example@example.com"
 }
```

Such technology offers different advantages for both people and companies:

- For Individuals (Knowledge & Control): Consent management (before, during, and after)
- For Companies (Compliance & Trust): Makes it easy (and cheap) for companies to comply with new laws
- For Regulators (Oversight & Management): Provides Regulators with flexibilty to regulate and enforce regulations according to localised requirements
- For Everyone: Improving economic performance of policy solves many issues in identity management

4.4 eTRON

TRON Project has been aiming to realize the vision of IoT in which many computers in our surroundings cooperate together to offer better services to improve the quality of everyday living. To realize such an environment, we need the mechanism to let objects to communicate with each other to collaborate. But we also need a mechanism to learn and collect user contexts such as the language spoken, age, many physical characteristics, preferences, etc. Such knowledge must be reflected to create an optimized environment. For this purpose, TRON Project is now building a general-purpose PDS (Personal Data Store) using IoT and cloud computing technology. We call this PDS "Omotenashi Cloud", as depicted in Figure 1. *Omotenashi* is a Japanese word that encapsulates a multitude of hospitality philosophy and practices rooted in Japanese culture.

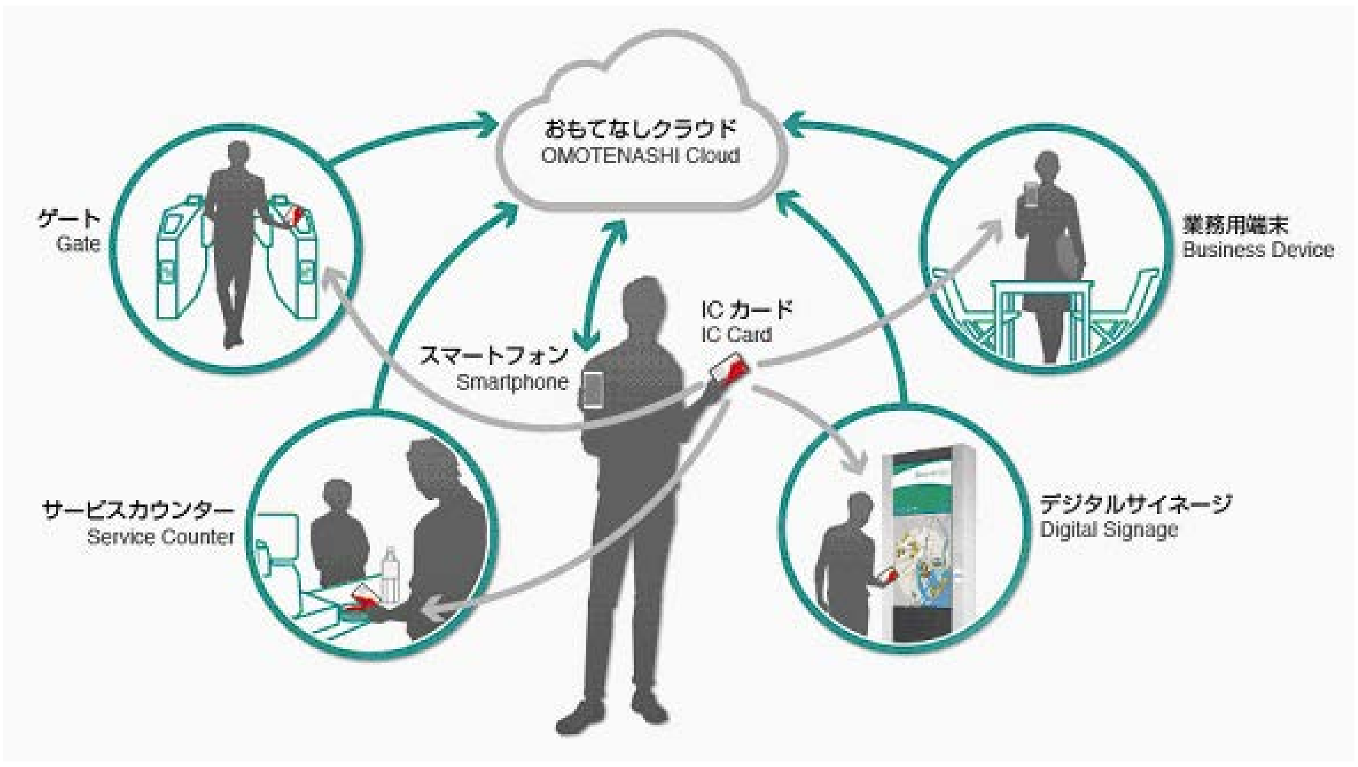


Figure 5. The interplay among Omotenashi Cloud, user devices and various services

In the short time span, this "Omotenashi cloud" can be used to advance the hospitality service to the increasing number of overseas tourists who are expected to peak during 2020 Tokyo Olympics and Paralympics. Customers (consumers) store their individual characteristics such as preferred spoken language, food taboos, passport information for sales tax-exemption in the cloud. From the application dashboard, users can select in advance what information to share with which vendors, and can thus control access to their information based context attributes and trust levels. Once the data is stored, such information is easily retrieved and passed on the spot to service providers such as stores and public services by using the smartphone apps or transportation IC cards as retrieval key.

The party who receives a service escrows the personal data in the PDS cloud, and lets it pass the data to the service vendors as the need arises. So, it is the customer (consumer) that controls the vendors. This concept, VRM (Vendor Relationship Management) is the reverse of the conventional idea of CRM (Customer Relationship Management). Omotenashi Cloud plays the important role of PDS in the scheme. This framework has been designed to be useful as legacy service platform in Japan even after 2020 when it will be utilized very much for hospitality service to foreign tourists. Small regional service vendors who did not have the resources to build conventional CRM can join Omotenashi Cloud service framework and can now provide better services suited for the new IoT age. Omotenashi Cloud will be useful for invigorating regional economy and is also expected to add to e-inclusion by accelerating the realization of Enableware and assistive technologies.

4.4.1 Architectural Overview

eTRON [4], an offshoot of the TRON project, is a basic architecture for "electronic entity" management. The term "electronic entity" or simply "entity", in eTRON parlance, represents special digital information that resembles real life objects like certificates or banknotes or keys. Certificates or banknotes are difficult to duplicate or alter as they use special papers, special ink, water marks or holograms. Keys are difficult to produce or reproduce as they have precisely cut edges made form hard materials. The "electronic entity", as envisioned in the eTRON architecture, should have such properties that make them difficult to produce, duplicate and alter. Creating an electronic entity is impossible with software alone, and hardware support is indispensable. The eTRON tamper-resistant hardware device can provide such support. An electronic entity produced by eTRON tamper-resistant hardware is composed of an indivisible unit of data. Such an electronic entity is difficult to counterfeit, produce a pirated copy, and illegally modify. Manipulation of electronic entities inside eTRON hardware – eTRON chip being the most common incarnation – is strictly governed by the entity transfer protocol (eTP) which features only a limited set of operations that can be exerted only after stringent mutual authentication and access control requirements have been satisfied.

4.4.2 Mutual Authentication and Access Control: User Empowerment in Data Manipulation

The eTRON mutual authentication [5] is performed by a challenge-response protocol using public key cryptography. After successful authentication, a session is created to share a secure key by Diffie-Hellman algorithm. By this session establishment an ephemeral key, valid only for the duration of the session, is shared between the interlocutors by Diffie-Hellman key sharing algorithm. In the eTRON architecture, this session is established in two phases; first a shared key is generated, and then the key is verified in order to thwart any possible man-in-the-middle attack.

In the eTRON architecture, mutual authentication has two modes: owner mode and issuer mode. By "owner" we mean owner of the chip or electronic entity and by "issuer" we mean issuer or creator of the entity (e.g., eTRON files or records). When an eTRON file is created, the eTRON ID of the issuer is attached to the file's file control block.

Issuer mode authentication is granted only when request to access an entity (file) comes from a party having the same eTRON ID as is written in the file control block of the entity. If the eTRON IDs match, the requesting party is given "issuer authority". Otherwise, "other authority" is given. On the other hand, in owner mode authentication, the requesting party proves himself as the owner of the chip by password or PIN, and upon successful authentication "owner authority" is granted.

eTRON's access control mechanism [5] is based on access control lists, where users can exert full control over their data by explicitly setting access privileges. The file access control list in eTRON is defined by setting or resetting different bits of a 16-bit value. "Issuer" of entity (file) has all rights by default, and rights for "owner" and "others" can be set for eight access privileges, namely updating record, reading record, changing mode of a file, acquiring status of a file, deleting file, transferring file, encrypting file, and decrypting file. The file access control list is defined by the issuer of the file during the file creation process.

In the current scope of the CPaaS.io project, the eTRON architecture is envisaged to be used on the Edge Side (as shown in the figure below) for secure maintenance of personal data.

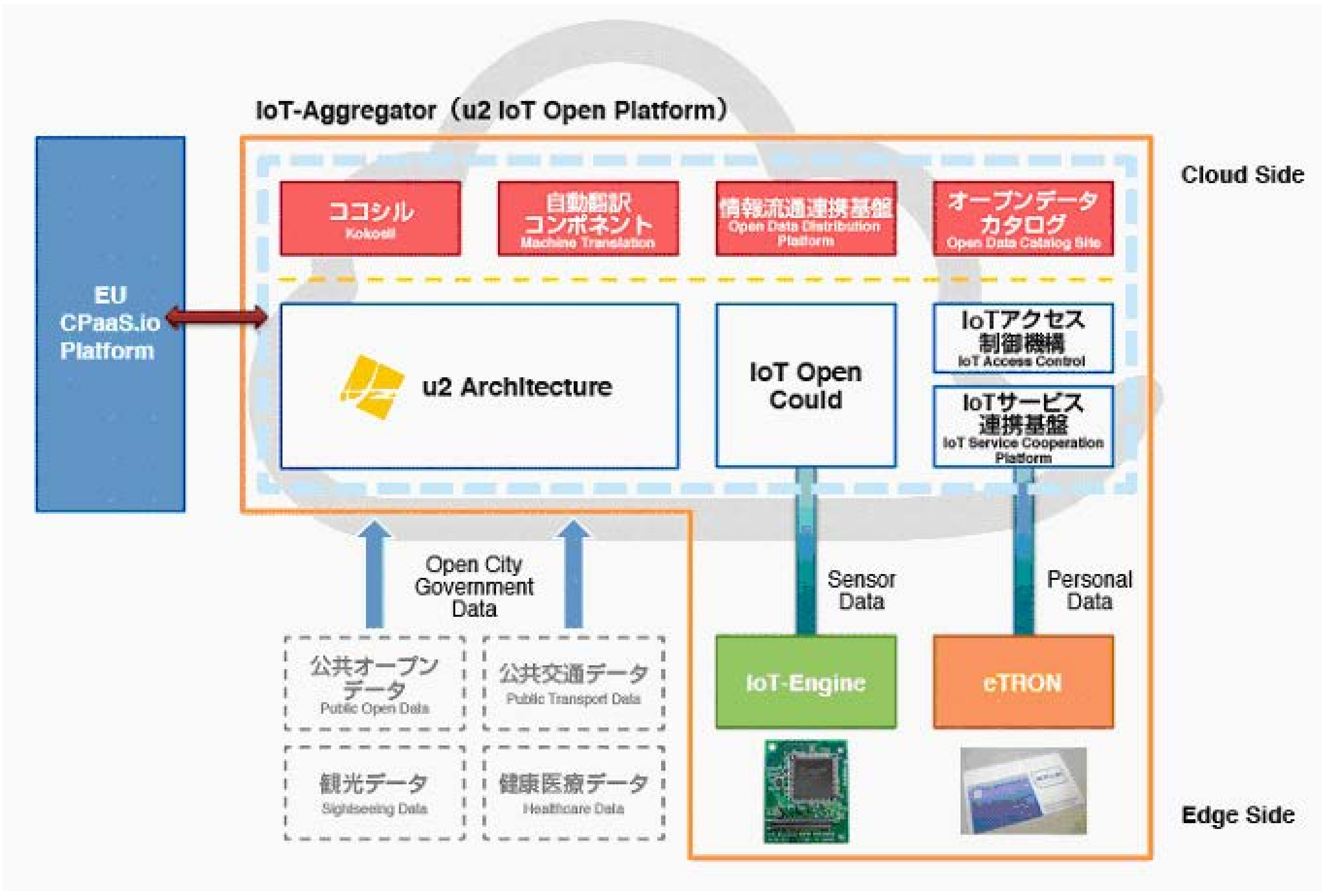

IoT-Aggregator（u2 IoT Open Platform）
Cloud Side
EU CPaaS.io Platform
ココシル Kokosil
自動翻訳コンポネント Machine Translation
情報流通連携基盤 Open Data Distribution Platform
オープンデータカタログ Open Data Catalog Site
u2 Architecture
IoT Open Could
IoTアクセス制御機構 IoT Access Control
IoTサービス連携基盤 IoT Service Cooperation Platform
Open City Government Data
Sensor Data
Personal Data
公共オープンデータ Public Open Data
公共交通データ Public Transport Data
観光データ Sightseeing Data
健康医療データ Healthcare Data
IoT-Engine
eTRON
Edge Side

Figure 6. Overall schematic of JP-side CPaaS.io platform comprising of Cloud Side and Edge Side

5 Impact on architecture and platform view

5.1 Architecture impact

In the below figure, Figure 7, you can find the functional architecture proposed in CPaaS-IO D3.2 [6] describing the functional groups and functional components involved in this architecture.

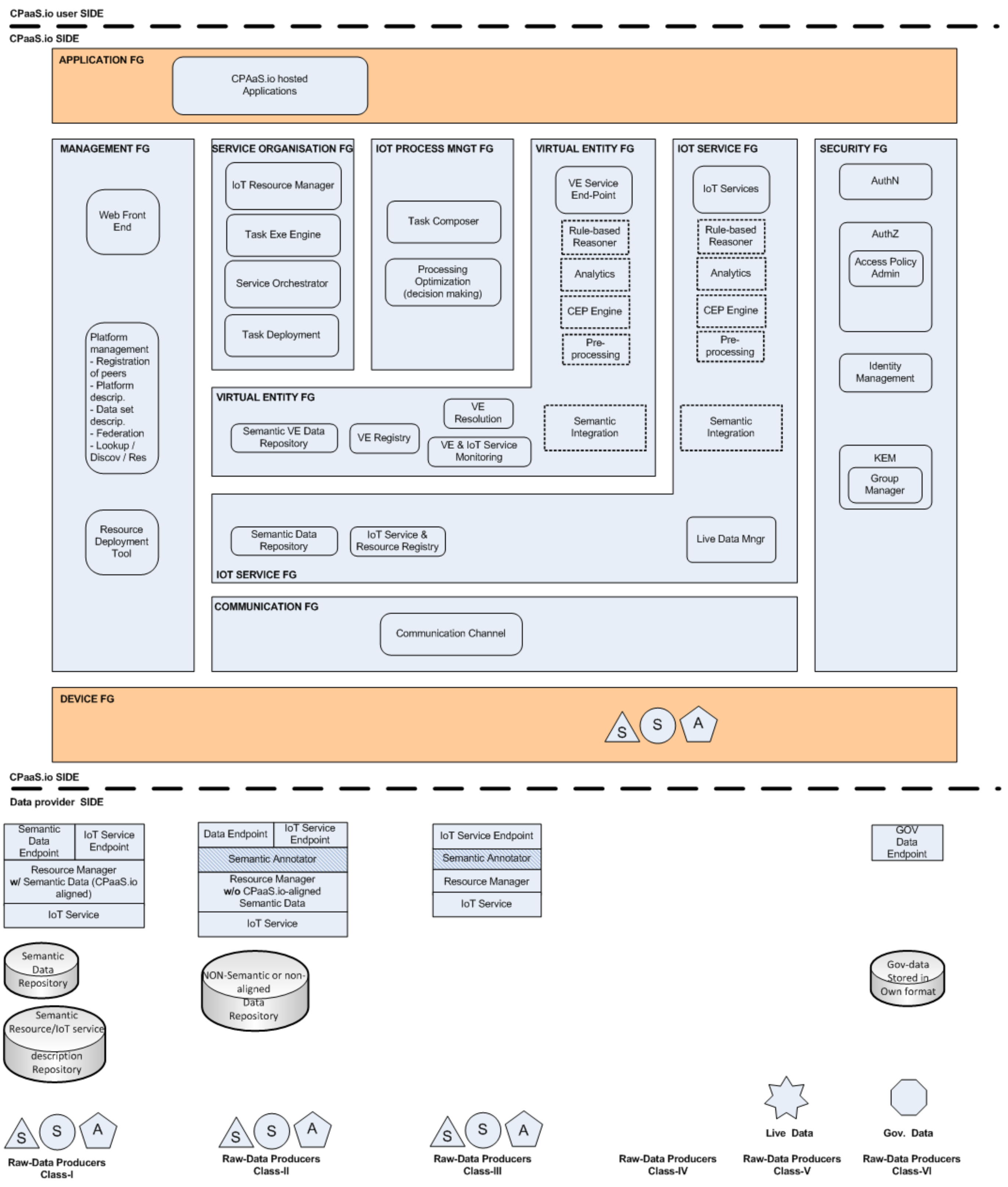


Figure 7. Coverage of Requirements [green-covered, dotted line–induced] vs. the functional decomposition

Since the present deliverable is focused on the empowerment components definition, let us first review the Security Functional Group definition and its components. The Security FG follows closely the recommendation for IoT-A native FCs. This FG comprises several components already identified in the IoT-A native Functional View:

- **AuthN**: This component is responsible for enforcing the authentication of registered CPaaS.io users. It also interacts with the User Management FC when the user registers in order to produce the user credentials. In addition, when a user requests access to a resource or service, such request is captured by a decision point as whether to grant or deny the request based upon whether the requester is authorized to do so. At this point, the AuthN FC can be contacted by the decision point to assert the authenticity of the requester.

- **AuthZ**: This component makes decisions about access control requests (intercepted at access decision points) based upon Access Control Policies (ACPs). Its scope embraces also the level of VEs/resources look ups.
    - Access Policy Administration: It is a sub-component of the Authz which provides a Policy Administration Point to the AuthZ FC through a GUI which allows the owner of the resource to create, update and delete access policies and attach them to its resources.
- **KEM**: It manages the exchange of security information between two parties. In particular, it ensures that the keys required to construct a secure and trusted communication channel is carried out in a secure manner.
- **Identity Management**: This component takes into account the management of the whole identity lifecycle and also the interoperability issues with components like Authentication, Authorization and KEM.

In addition to these components new ones should be added in order to cope with the technologies defined in the previous section.

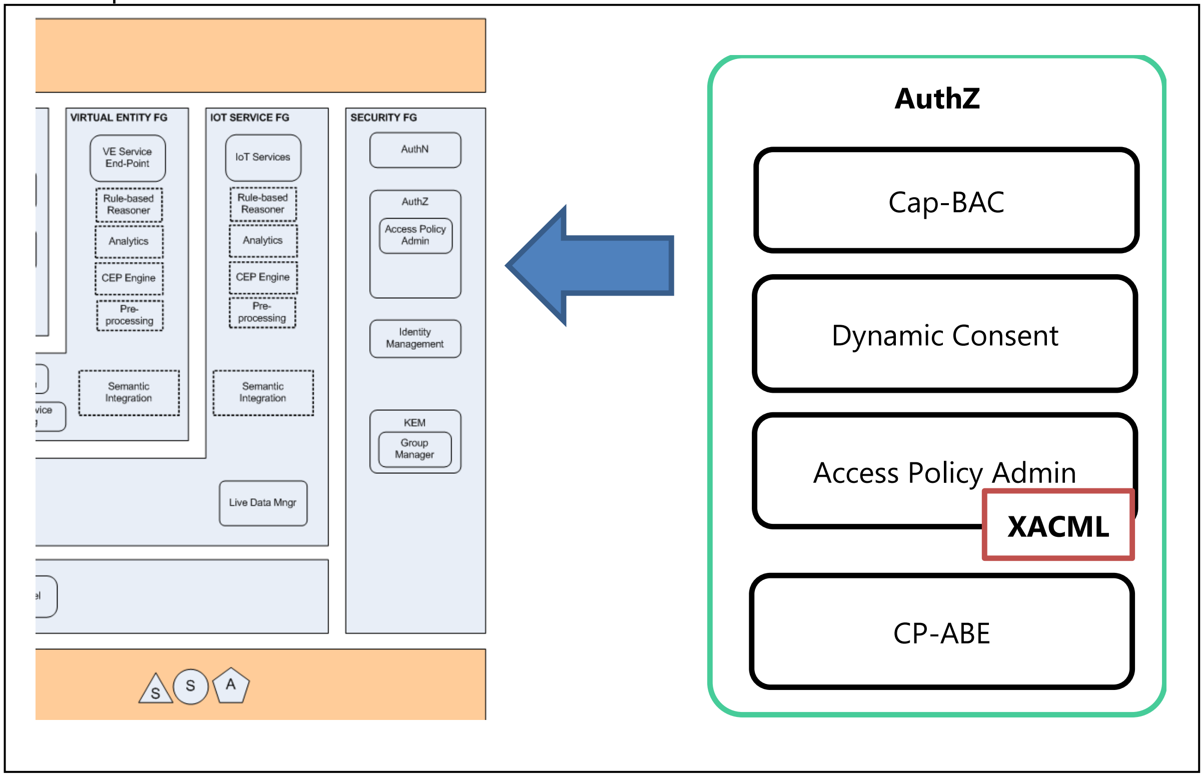


Figure 8. New FCs to be incorporated to the architecture

Figure 8 illustrates the new AuthZ FC embracing the technologies presented above in the previous section:

- CapBAC which will integrate the capability-based access control mechanisms allowing to dynamically grant access to specific resources and during a specific period of time over a resource.

- XACML which will enrich the Access Policy Admin by allowing to specify context-aware privacy policies.

- CP-ABE as a mechanism to securely distribute information over a shared media allowing only legitimate receivers with the appropriate attributes to decrypt the information.

- Dynamic Consent which will allow users to specify in a clear way their personal information they want to be published or commercialized.

5.2 Platform components for users' empowerment

Taking into account the technological approach, as well as the set of requirements regarding users' empowerment for security and privacy, Figure 9 gives an overview about different CPaaS.io components to cope with such requirements. It should be noted that some of these components are designed in the context of the European platform FI-WARE[20], and they are to be integrated with other CPaaS.io components. The main motivation for using FI-WARE is twofold: on the one hand, to take advantage of already mature security components that can be reused; on the other hand, to leverage the use of a well-known and widely used semantics model based on NGSI [19]. Furthermore, this overview is mainly focused on the use of the DCapBAC access control approach to empower users with the ability to govern how their devices (and associated data) can be accessed.

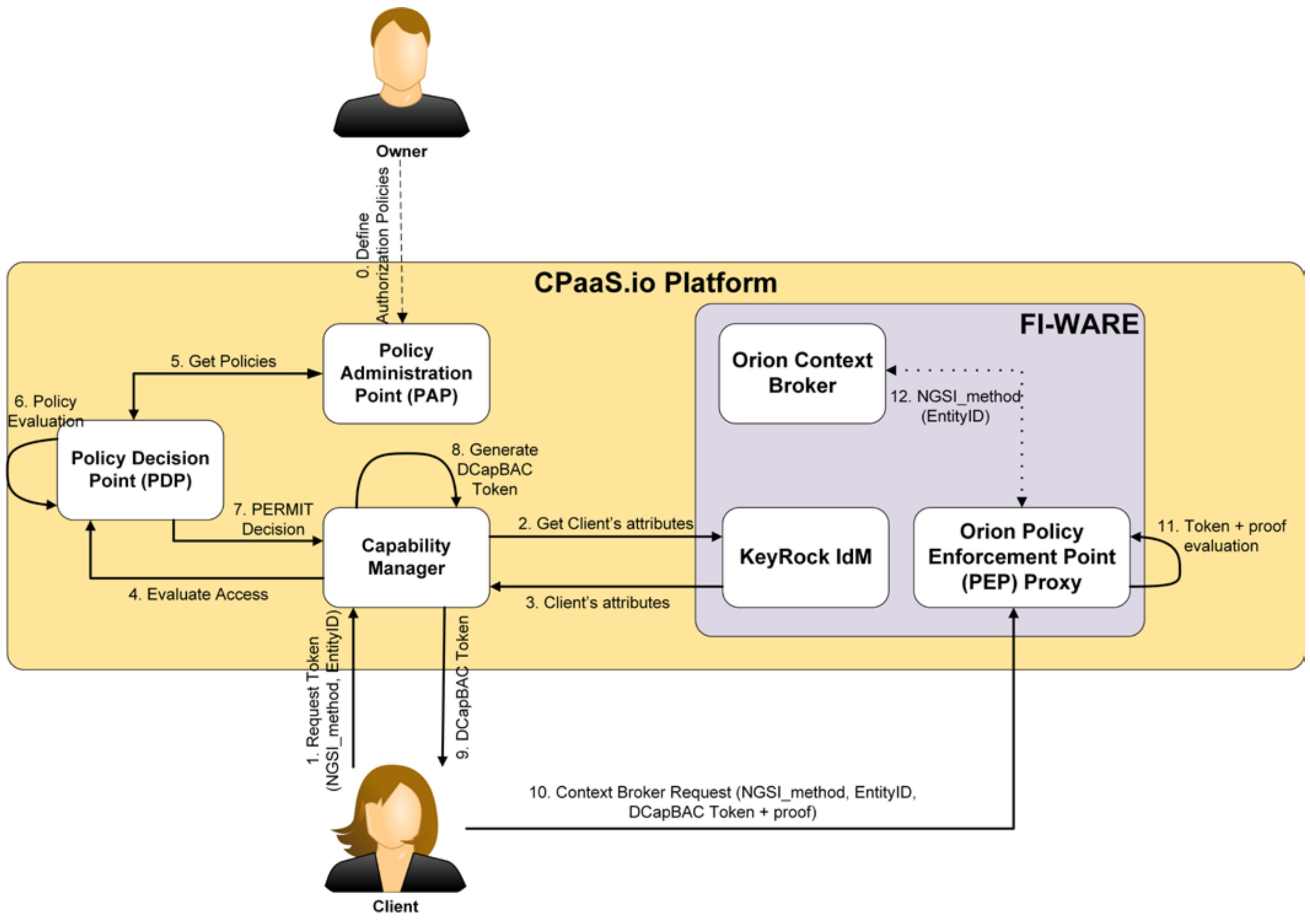

Owner
0. Define Authorization Policies
CPaaS.io Platform
FI-WARE
Policy Administration Point (PAP)
5. Get Policies
6. Policy Evaluation
Policy Decision Point (PDP)
7. PERMIT Decision
8. Generate DCapBAC Token
Capability Manager
4. Evaluate Access
2. Get Client's attributes
3. Client's attributes
Orion Context Broker
12. NGSI_method (EntityID)
KeyRock IdM
Orion Policy Enforcement Point (PEP) Proxy
11. Token + proof evaluation
1. Request Token (NGSI_method, EntityID)
9. DCapBAC Token
10. Context Broker Request (NGSI_method, EntityID, DCapBAC Token + proof)
Client

Figure 9. Platform components for users' empowerment

Before the description of the main required interactions, it should be noted that we have considered 2 main external entities to the platform:

- The *Owner* is the user or service in charge of defining access control policies, in order to guarantee only authorized clients (e.g. a user) will be able to get access to their devices' data.
- The *Client* represents a user, service or device that tries to perform a specific action (i.e. a NGSI

method) over an entity registered in the platform.

In addition to these external entities, in order to realize the DCapBAC's functionality, following components are defined:

- The *Policy Administration Point* (PAP) represents the point where users are empowered to define their access control policies, in order to govern the access to their devices' data.
- The *Policy Decision Point* (PDP) is responsible for evaluating access request against the set of access control policies that are defined in the PAP. Both PDP and PAP are based on the XACML standard.
- The *Capability Manager* is the component for generating DCapBAC tokens in case of receiving affirmative authorization decisions from the PDP.

As already mentioned, the CPaaS.io components related to the users' empowerment for DCapBAC are intended to be integrated with already defined mechanisms, which are provided by FI-WARE:

- The KeyRock IdM represents the central point of the platform where users, services or devices are registered, including their identifiers and attributes.
- The Orion Context Broker is intended to provide the functionality of a data broker, in order to allow different entities to remain decoupled, by following a publish/subscribe communication model.
- The Orion Policy Enforcement Point (PEP) Proxy is responsible for intercepting access requests to the *Orion Context Broker*, by evaluating the request content in order to determine if the access must be granted or denied.

Before describing the main interactions among these components, it should be noted there are different steps that are assumed to be made before the process. In particular, it is assumed that users' services and devices are also registered in the *KeyRock IdM* component. During this initial registration stage, identifiers and attributes of users, services, and devices are stored in such element. In this way, according to Figure 9, initially it is assumed the *Owner* has defined the set of access control policies through the *PAP*, to determine which entities are authorized to perform which actions over their devices in the platform (*step 0*). Then, when a *Client* tries to get access to perform a specific action over a specific device, it requests a DCapBAC token by querying the *Capability Manager* (step 1). This request includes the NGSI method (i.e. the action) and the device, which is represented as the *EntityID* registered in the *KeyRock IdM*. Upon receiving this request, the Capability Manager authenticates the Client, and additionally, it gets Client's identity attributes from the KeyRock IdM (steps 2, 3). With these attributes, as well as the NGSI method and EntityID specified within the request, the Capability Manager asks the *PDP* (step 4) to determine whether the requested credential must be generated. The PDP uses the policies defined by the *Owner* in the *PAP* (step 5), and evaluates them against the request (step 6). An example of XACML policy in which *OdinS* users are authorized to perform the *queryContext* action over the entity *CPaaSEntity01*, is shown in Figure 10.

```
<Policy "CPaaS_example">
        <Target>
                <Resource>
                        <AttributeValue DataType="String">CPaaSEntity01</AttributeValue>
                        <ResourceAttributeDesignator DataType="String" AttributeId="resource-id" />
                </Resource>
                <Action>
                        <AttributeValue DataType="String">queryContext</AttributeValue>
                        <ActionAttributeDesignator DataType="String" AttributeId=" action-id "
                </Action>
        </Target>
        <Rule "Permit">
                <Target>
                        <Subject>
                                <AttributeValue DataType="String">OdinS</AttributeValue>
                                <SubjectAttributeDesignator AttributeId="subject-id" DataType="String"/>
                        </Subject>
                </Target>
        </Rule>
        <Rule "Deny">
</Policy>
```

Figure 10. XACML policy example for CPaaS.io platform

In the case of an affirmative decision (step 7), the *Capability Manager* generates a token for the Client (step 8). Following the above example, Figure 11 shows an example for such action and entity. It should be noted that, following NGSI schema and the same DCapBAC token format, the semantics is slightly different. On the one hand, the field *de* represents the identifier of an entity that is registered in the plaform. On the other hand, the *ar* field makes reference to a NGSI method (*ac* field), and *re* represents specific attributes of that entity. Additionally, as already mentioned, this token includes the client's public key and time restrictions, delimiting the validity period for this credential. Furthermore, the *Capability Manager* signs the generated token.

```
{
      "id": "eg3fq:fb5r23tra3",
      "ii": 1485172121,
      "is": "capabilitymanager@cpaas.io",
      "su": "zNwS5FetB4rwzSKsWwSBAxm5wDa=JgLjHU8zSnmeSFQgSG9HhdsJrE8=",
      "de": "CPaaSEntity01",
      "si": "SbUudG4zuXswFBxDeHB87N6t9hR=PBQqCN3gpu7nSkuPzDk7kaR3dq1=",
      "ar": [
            {
                  "ac": "queryContext",
                  "re": "*"
            }
      ],
      "nb": 1485172121,
      "na": 1485174121
}
```

Figure 11. Capability Token Example for CPaaS.io platform

This way, once the client obtains the token (step 9), it tries to perform a specific action (i.e. NGSI method) over a specific device (i.e. an entity) by using the token, as well as a proof to demonstrate that it is effectively the entity associated with that token (*proof-of-possession* [21]) (step 10). This request is sent to the *Orion PEP Proxy*, which evaluates the content of the token (step 11) by verifying its validity, and checking if the requested operation (e.g. *queryContext* over *CPaaSEntity01)* is contained in the token. This interaction is to be performed with mutual authentication (e.g. with HTTPS) based on public key (that is, raw public key or certificate). Therefore, the *Orion PEP Proxy* can verify that the public key contained in the DCapBAC token is the same key that was used in the authentication during the TLS exchange. Furthermore, it verifies the token's signature by making use of the *Capability Manager*'s public key, which is assumed to be statically configured during an out-of-band process. Finally, in case the token is correctly evaluated, the action is performed over the *Orion Context Broker* (step 12).

6 Conclusions

According to Task 5.1, in this deliverable we have defined the main components for user empowerment. Such definition involves different subtasks which have been reflected in the different sections of this deliverable.

During a requirement analysis phase, we have collected the different functional and non-functional requirements associated to the users' empowerment. A worth-highlighting component to be taken into account is an authorization mechanism that will allow individuals to control how they want their personal data should be published and commercialized. Such concern is treated in this deliverable in Section 3 where MyData Initiative is explained as a valuable alternative for both individual and company sides allowing the former one to control how his personal information is published, as well as the latter to consume this information thanks to a single market where individuals have easily specified their agreement to the use of their personal information.

In section 4, we have detailed the different technologies relevant for the architecture where context-aware privacy policies, capability-based access control and dynamic consent management are presented.

Since this project proposed an international integration with Japan, eTRON has also been introduced in this deliverable presenting how both European and Japanese architecture are integrated.

Finally, the new architecture is presented in Section 5, where new components are integrated into the functional view of CPaaS architecture and example of their use are also presented.

The next version of the CPaaS.io architecture will be updated continuing Task 5.1 in which we also incorporate mechanisms for privacy preserving defined in Task 5.2. This way we will provide a global architectural vision of the approach provided in the project for the user empowerment.

7 References

[1] F. Carrez and A. Skarmerta. CPaaS.io Deliverable 3.1. Requirement Collection Analysys V1.0, Sept 2016.

[2] Cooper, T. and LaSalle, 2016. R. Guarding and growing personal data value. Accenture.

[3] Poikola, A. et al. MyData – A Nordic Model for human-centered personal data management and processing.

[4] Sakamura K, Koshizuka N (2001) The eTRON wide-area distributed-system architecture for e-commerce. IEEE Micro, 21(6): pp 7-12.

[5] Khan M.F.F., Sakamura K., Koshizuka N. (2017) Robust Enterprise Application Security with eTRON Architecture. In: Chang V., Ramachandran M., Walters R., Wills G. (eds) Enterprise Security. Lecture Notes in Computer Science, vol 10131. Springer, pp. 155-178.

[6] F. Carrez, A. Skarmeta, M-A. Zamora, A. Canovas, M. Strochbach, S. Haller, M. Fraefel, A. Gschwend, G. Solmaz, B. Cheng, M. Bauer, N. Koshizuka. Deliverable 3.2. CPaaS.io System Architecture v1. Jan 2017.

[7] Specification of the Identity Mixer Cryptographic Library. Version 2.3.40 IBM Research, Zurich January 30, 2013. Available http://www.zurich.ibm.com/idemix/

[8] Jose M. Such, Agustin Espinosa, Ana Garcia-Fornes, Vicent Botti, Partial identities as a foundation for trust and reputation, Engineering Applications of Artificial Intelligence, Volume 24, Issue 7, October 2011, Pages 1128-1136

[9] OASIS Standard. eXtensible Access Control Markup Language (XACML) Version 3.0. January 2013: http://docs.oasis-open.org/xacml/3.0

[10] J. L. Hernández-Ramos, A. Jara, L. Marín, A. Skarmeta. DCapBAC: embedding authorization logic into smart things through ECC optimizations. International Journal of Computer Mathematics, 93(2), 345-366. 2016

[11] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEE, 2007, pp. 321–334.

[12] S. Gusmeroli, S. Piccione, and D. Rotondi, A capability-based security approach to manage access control in the internet of things, Math. Comput. Model. 58(5–6) (2013), pp. 1189–1205.

[13] Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., & Ylonen, T. (1999). *SPKI certificate theory*. IETF RFC 2693, September.

[14] From ABAC to ZBAC: The Evolution of Access Control Models http://www.hpl.hp.com/techreports/2009/HPL-2009-30.pdf

[15] J. Dennis and E. Van Horn, *Programming semantics for multiprogrammed computations*, Commun. ACM 9(3) (1966), pp. 143–155.

[16] M. Jones, J. Bradley, and N.Sakimura, *JSON Web Token (JWT)*, OAuth Working Group, Internet Engineering Task Force (IETF), work in progress, draft-ietf-oauth-json-web-token-11, July 2013. Available at http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-11.

[17] S. Li, J. Hoebeke, F. Van den Abeele, and A. Jara, *Conditional observe in CoAP*, Constrained resources (CoRE) Working group, Internet Engineering Task Force (IETF),work in progress, draft-li-core-conditionalobserve-04, June 2013. Available at http://tools.ietf.org/html/draft-li-core-conditional-observe-04.

[18] C. Jennings, J. Arkko, and Z. Shelby, *Media types for sensor markup language (SENML)*, NetworkWorking group, Internet Engineering Task Force (IETF), Work in Progress, draft-jennings-senml-10, October 2012. Available at http://tools.ietf.org/html/draft-jennings-senml-10.

[19] NGSI Context Management. Open Mobile Alliance (OMA). May 2012.

[20] FI-WARE wiki. https://forge.fiware.org/plugins/mediawiki /wiki/fiware/index.php/Welcome_to_the_FI-WARE_Wiki

[21] P. Hunt, J. Richer, W. Mills, P. Mishra, and H. Tschofenig. OAuth 2.0 Proof-of-Possession (PoP) Security Architecture. IETF Internet Draft, draft-ietf-oauth-pop-architecture-07 (work in progress), 2016.