model-Based fRamework for dependable sensing
and Actuation in INtelligent decentralized IoT systems

D5.9 - Guidelines for privacy compliance and control in IoT services models

Deliverable ID	D5.9
Deliverable Title	Guidelines for privacy compliance and control in IoT services models
Work Package	WP5
Dissemination Level	PUBLIC
Version	1.0
Date	31/03/2021
Status	Final
Lead Editor	LINKS
Main Contributors	Enrico Ferrera (LINKS), Jure Rosso (LINKS), Michele Ligios (LINKS), Davide Conzon (LINKS)

Published by the BRAIN-IoT Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780089.

Document History

Version	Date	Author(s)	Description
0.1	17/01/2021	Enrico Ferrera (LINKS)	First Draft with TOC
0.2	05/02/2021	Enrico Ferrera (LINKS)	Section 3 added
0.3	15/02/2021	Jure Rosso (LINKS), Michele Ligios (LINKS)	Inclusion of Privacy Control System specifications
0.4	19/02/2021	Davide Conzon (LINKS)	Document ready to be reviewed
1.0	31/03/2021	Enrico Ferrera (LINKS)	Document ready to be submitted

Review History

Version	Review Date	Reviewer	Summary of Comments
0.4	26/02/2021	Vincent Bonneau (IDATE)	Approved with minor comments.
0.4	29/03/2021	Ricardo Vazquez (EMALCSA)	Approved with minor comments.

Table of Contents

1 Introduction

The deliverable D5.9, titled "*Guidelines for privacy compliance and control in IoT services models*" is the third outcome of the Task 5.3 "*Privacy awareness and control*".
The first two outcomes of the task have been the followings:

- D5.3 - Initial enablers for Privacy awareness and control
- D5.7 - Final enablers for Privacy awareness and control

D5.3 focused on an analysis of the General Data Protection Regulation (GDPR)'s precepts and defined a methodology based on Privacy Impact Assessment (PIA) to evaluate the risks for personal data misuse in an IoT application like the ones considered in BRAIN-IoT project (see D2.6 for a description of the use-cases). According to GDPR, risks assessment based on PIA must always be done when data containing information related to individuals are intended to be handled. This procedure helps to understand whether a specific application or service do satisfy the requirements to be allowed for the treatment of such personal data.

As described in D5.7, the planning of the activities related to Task 5.3 evolved a lot between the end of the second year of the project and during the third year, following the feedback received by the Reviewers and the External Stakeholder Group (ESG) members, as well as the hints provided by the H2020 Large Scale Pilot (LSP) MONICA Project's members. In fact, the collaboration between BRAIN-IoT and MONICA led to the identification of a set of use-cases which brought Brain-IoT Consortium few doubts about how to remain compliant with GDPR while still providing a reasonably good level of the quality of service.
The identified use-case, described in D2.6 and D5.7, differs slightly from a usual case where a data owner establishes a relationship with a service provider, who handles her/his personal data to deliver the required service. In fact, the identified use-case consists in a service provider who delivers a service, which is composed by micro services implemented and managed by different realms and administrative domains.
Such case is covered by GDPR in the sense that the internal micro services would be considered like third party services that the macro service is supposed to indicate in the PIA along with all the details about how the personal data will be used, treated, and managed. Basically, such cases would be handled composing the PIA of the single micro services in one big all-encompassing PIA of the macro service.
This methodology is fully GDPR-compliant and aligned with legal frameworks but, from a technological point of view, it is completely unscalable and very difficult to manage both from the side of the service provider and from the data owner's side. In fact, whenever the macro service is supposed to be composed by a large number of micro services, the PIA gets difficult to be performed and also it may easily be error prone. Moreover, in a context where the micro services can frequently vary, e.g., when the provided macro service implements a location-based user experience and the micro-services can churn according to the position of the end user, the macro service if forced to contact the end-user/data owner very frequently to inform her/him about the new service conditions and data management information. Such situations would be very complex to handle and also annoying for a smooth experience and sufficient quality of the service provided.

Apart from the use-case defined in conjunction between BRAIN-IoT and MONICA, wider context and scenarios have been identified as possibly affected by the issues mentioned above. More specifically, these issues could be relevant for:

- Smart City scenarios where data provided by citizens can be required and exploited by a multitude of services belonging to the City;
- Scenarios where IoT systems would like to access data source, which provide data streams in an open or monetized way.

Smart cities provide services that may use data provided by citizens to implement public utility services, free or commercial services, marketing and entertainment services, integrating and federating sub-services (i.e., micro services) provided by public or private entities belonging to the city itself.

Data can come from the most disparate sources, public or private. Data can be generated by citizens' personal IoT devices, and citizens also may want to provide such data in an open or monetized way. The lowest common denominator which includes such scenario and privacy issues, is the right of the data owners to grant the access to her/his data according to her/his own rules.

1.1 Scope

The goal of D5.9 is to address a general conceptual model as a guideline for future research in the domain of data access control for the purpose of privacy protection or for granting that data owners' requirements are respected in the context of a Marketplace of Data, where open or monetized data are made available to requiring services.

Section 2 describes the reference conceptual model for authorized data exploitation by data consumer services. Section 3 addresses the BRAIN-IoT way to model access control policies, along with some suggestions for improvement. Finally, Section 4 reports the documented specifications of the BRAIN-IoT Privacy Control System for enforcing data access control policies in scenarios like the proposed ones.

1.2 Related documents

ID	Title	Reference	Version	Date
[RD.1]	Initial Threat modelling and Security assessment of target scenarios, solutions	D5.3	1.0	31/03/2019
[RD.2]	Final enablers for Privacy awareness and control	D5.7	1.0	30/11/2020
[RD.3]	Final Visions, Scenarios, Use Cases and Innovations	D2.6	1.0	31/08/2020

2 Conceptual model for authorized data exploitation

Smart cities deploy and administrate IoT-enabled platforms to provide services and applications, which may be implemented as an interoperable federation of a multitude of heterogeneous data sources as well as existing micro services used as a sort of building blocks.

Figure 1 represents the conceptual model proposed as part of the control framework for data privacy compliance and enforcement. It represents the relations between main data access control concepts, actors: service providers as well as data owners.

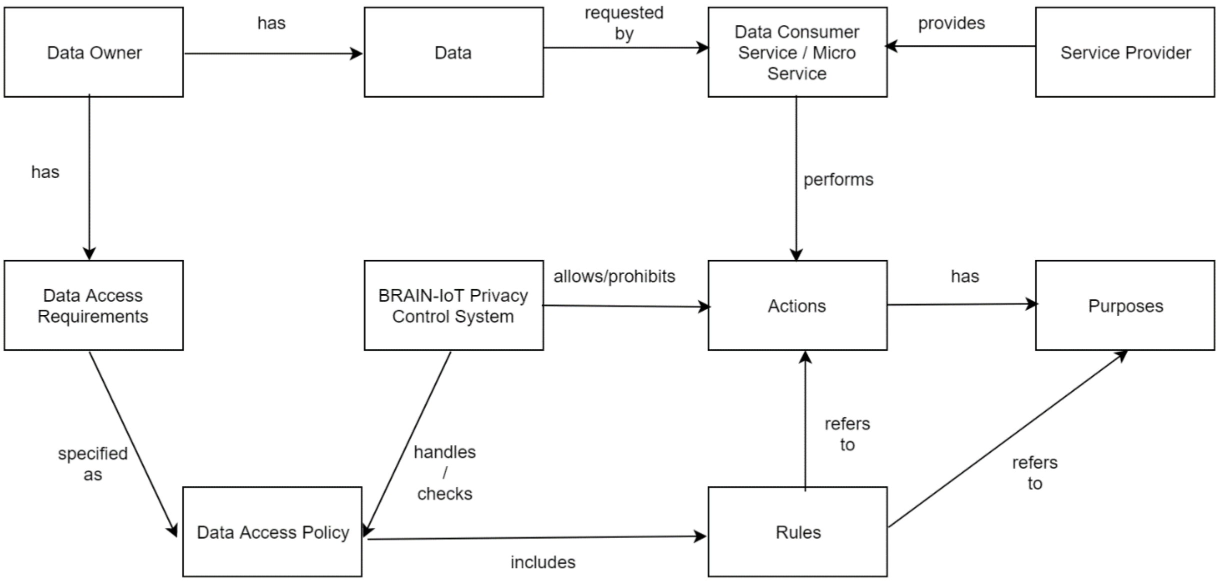

Figure 1 - Conceptual model

The *Service Provider* is the person/company/public administration, which uses means, facilities, and algorithms to perform *Actions* on the Data Owners' data with specific *Purposes* for providing a *Data Consumer Service* or *Micro Service*, implementing specific functionalities. In the context of a Smart City platform, many parties, implementing dedicated Micro Services and possibly belonging to different realms and administrative domains, can take part in the offered services having access to the data for manipulating it.

The *Data Owner* is a person who produces and owns data. It represents a person whose personal data is going to be stored and/or processed by a Data Consumer Service or by micro services, which have been federated to compose a larger and more complex service. The Data Owners have the rights stipulated by the GDPR, which oblige the Data Consumer Service to inform transparently and in details how the Data are used, processed, managed. However, in some cases, the Data Owner could also be interested to provide her/his own data in an open market, with the ambition to support e.g., researchers, public services the community, with open data streams, or to monetize such data, selling them to anyone who might be interested in it. In these scenarios, the main interest of the Data Owner is to provide an access point to their data source and guarantee that the rights to access such data are respected under the terms established by the Data Owner herself/himself.

The focus of GDPR is principally personal *Data*, which means any information relating to an identified or identifiable person. The authors believe that any kind of Data stored and/or processed by a Data Consumer Service needs to be protected in line with the data protection preferences of the relevant Data Owner. So, in BRAIN-IoT, the partners extend the notion of sensitive Data to include any confidential business Data. In fact, the approach used by the *BRAIN-IoT Privacy Control System* intends to be applied to *allow* or *prohibit* unauthorized access to any kind of sensitive Data (personal or not), according to Data Owners' *Data Access Requirements*. Data Access Requirements are specified under the form of *Data Access Policies*.

In the following, further elements of the conceptional model are explained. A *Data Owner* provides *Data* to a *Data Consumer Service* that intends to process the *Data* performing specific *Actions* on it. The different *Actions* define what is going to be done with the *Data* by the *Data Consumer Servi*ce. The *Data Owner* has *Data Access Requirements* that satisfy their intentions in terms of preserving and protecting her/his own *Data*. *Data Owners* can specify their *Data Access Requirements* as *Data Access Policies* and attach them to the *Data* in such a way that the such policies will be enforced by the *BRAIN-IoT Privacy Control System* under the defined requirements every time that *Data Consumer Service* requires it. In other words, a *Data Access Policy* is generated from the *Data Own*ers' input that specifies the *Data Access Requirements* of a *Data Owner* in a machine-readable way. Such a policy contains *Rules* that define obligations to the *Actions* and *Purposes* of the *Data Consumer Service* against the *Data*. These obligations can for example prohibit the processing of *Data* for marketing purposes.
The *Rule* is a core element of the proposed conceptual model. It refers how the *Data Owner* desires her/his *Data* will be used and under which circumstances the *Data* can be accessed and used by *Data Consumer Services*. The *Rules* are related to *Data* by adopting the approach of sticking them to the *Data* under the form of *Data Access Policies*. The *Rules* refer directly to *Actions* and *Purposes* that are allowed for the *Data* treatment by the *Data Consumer Services*. In other words, by this association to the *Data*, the *Rules* defines the set of *Actions* and *Purposes* that are allowed to be applied on the *Data*.

In the next chapter, the simple way to categorize *Actions* and *Purposes* adopted by the BRAIN-IoT approach is reported, along with a description of how the *Data Access Policies* are shaped according to the *Rules* derived by the *Data Owners' Data Access Requirements*.

3 Data Access Policies: Rules for authorized Actions and Purposes

The enforcement of the Data Access control strategies is regulated by the Data Access Policies.
Policies are indications for establishing whether certain actions or purposes are permitted with the Data Owners' Data. In IoT domain, people often use to thinking about data as products of some devices which does not deserve or require any specific usage regulation for using it. That's an error. As for software releases, Data also requires licenses which specify certain rules that must be respected if a third-party would like to obtain permission for using it. In BRAIN-IoT, Data Access Policies are meant to be a generalization of the licence concept, combined with privacy requirements: the Data Access Policies specify which actions and purposes are permitted with Data Owners' Data.

Data Access Policies include the Rules describing the Data Owners' Data Access Requirements. Rules can be specified as follows:

- Disclosure: specifies whether the Data of the Data Owner is allowed to be disclosed to audience.
- Storage: specifies whether and how long the Data of Data Owner is allowed be stored at maximum.
- Purpose: specifies to which purpose the Data Owners' Data is allowed to be used and processed.
- Attribution: specifies whether the Data Consumer Service is required to cite Data Owners whenever they use their Data.
- Commercial: specifies whether Data are allowed to be used by Data Consumer Services for commercial purposes.
- Modification: specifies whether Data can be manipulated (e.g., aggregation, processing) or it should be used only as delivered by the data source.
- Original Data Redistribution: specifies whether Data can be redistributed by the service.
- Derivative Data Redistribution: specifies whether manipulated data (e.g., aggregation, processing, anonymization) are allowed to be redistributed as a new Data stream. A specific case of Derivative Data Redistribution rule is when derivative data are allowed to be redistributed only if the same policies are applied to the new Data stream.
- Anonymization: specifies whether Data are allowed to be used only after Data anonymization.

From the Data Consumer Service side, the Service Provider is also obliged to complement the description of the policies related to the service with some more information required by the GDPR, such as how can the Data Owner access, delete, visualize, download her/his data.

Every Data Consumer Service has a purpose for requiring and exploiting Data Owners' Data that is related to the scope of the service provisioning. Along with the Actions, the Purposes constitute the set of the elements in accordance with which Rules can be defined, depending on specific Data Owners' Data Access Requirements. By indicating whether specific actions and purposes are allowed, it is possible for the BRAIN-IoT Privacy Control System to decide whether the Data Consumer Service can obtain the Data or not. The BRAIN-IoT Privacy Control System manages both the sides of the coin: Data Owners' Data Access Policies and the policy declared by the Service Providers for their Data Consumer Services. These two policies are compared by the Policy Decision Point of the BRAIN-IoT Privacy Control System in order to establish whether the two policies are compliant and consequently allow or disallow the reception of the Data by the Data Consumer Service.
In BRAIN-IoT, the purposes of the services and micro services have been categorized in a set of few classes:

- Advertisement
- Profiling
- Recommendation
- Safety
- Entertainment
- Public Utility

Because of the constraints in terms of available resources, BRAIN-IoT focused its activities on the implementation of an infrastructure capable of handling the data accessibility in accordance with policies determined at runtime by users, rather than the modelling of the Policies. For this reason, the selected approach has been to model such policies as a simple JSON objects, represented as JSON Web Token (JWT), where authorized Actions and Purposes are included, following directives indicated via a Policy Dashboard to establish whether they are permitted or not. The list of possible Purposes can be extended whenever new classes are requested and identified.
Different policy modelling approaches can be defined while keeping the proposed conceptual model still valid. For instance, a definition of an ontology representing Actions and Policy is a viable and interesting approach, in conjunction with the adoption of JSON-LD, a method of using JSON to express linked data, including Resource Description Framework (RDF) and Web Ontology Language (OWL) data.

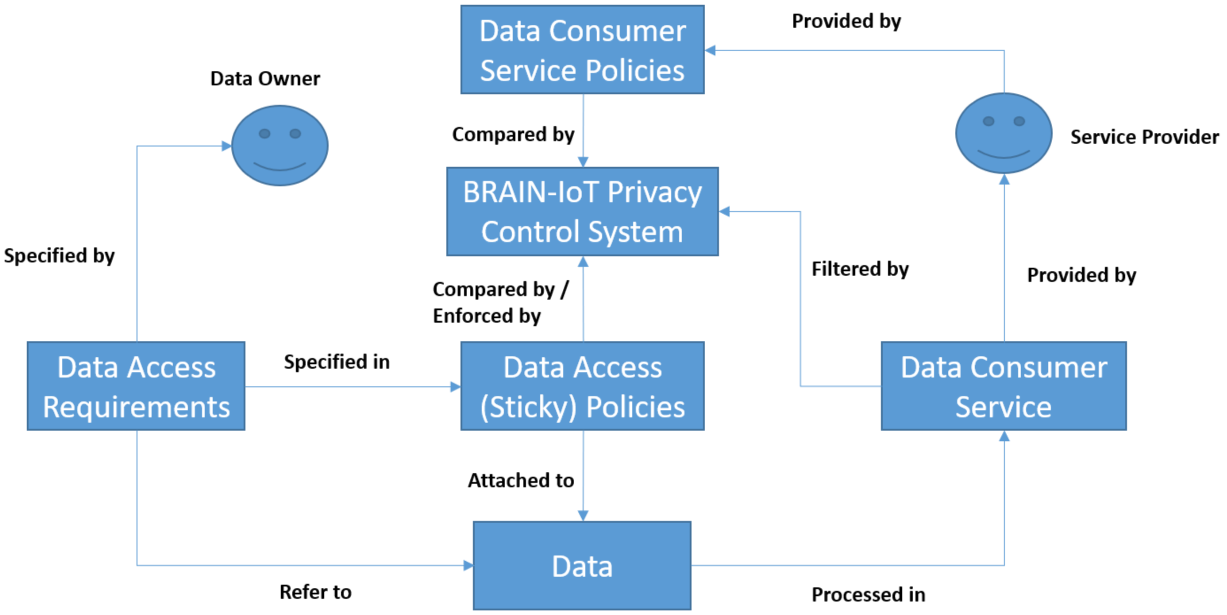

Figure 2 - Data Access Policy handling overview

Figure 2 shows the relations between the components involved in the Privacy Control approach defined in BRAIN-IoT. The Data Access Requirements are specified by the Data Owners and translated into Data Access Sticky Policies. The Data Access Sticky Policies are attached to the Data that is going to be processed in the Data Consumer Services provided by the Service Providers. Data Consumer Service Policies are in a first step provided by the Service Providers and afterwards compared with Data Access Policies by the BRAIN-IoT Privacy Control System, which is also responsible for enforcing the Data Access Policies according to the result of the match.

4 BRAIN-IoT Privacy Control System Specifications

The BRAIN-IoT Privacy Control System is one possible implementation of the general conceptual model described in Section 2 and Section 3. This section reports the specifications of the BRAIN-IoT Privacy Control System for enforcing data access control policies in scenarios like the ones addressed so far.
For the readers convenience, Figure 3 reminds the high-level architecture of the Privacy Control System.

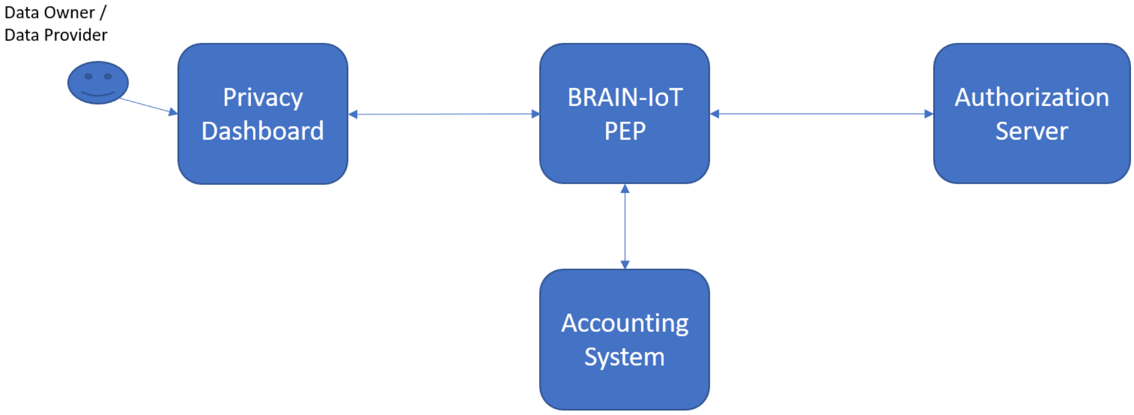

Figure 3 - Privacy Control System Architecture

For details about the internal workings of the component, the reader can look at D5.7 – "Final enablers for Privacy awareness and control".

4.1 Methods

4.1.1 Device

`GET /api/v1/device/{device_id}`

Get policies of a device (**getDevicePolicyMappingApiV1DeviceDeviceIdGet**)

> This endpoint provides a way to get the list of roles/policies (signed) by providing device_id (MAC). Should be used by the component that wants to know the policies to be associated to the message before forwarding it.

Path parameters

device_id (required)

Path Parameter —

Return type

DevicePolicyMapStrSigned

Example data

Content-Type: application/json

```
{
  "device_id" : "device_a",
  "signature" :
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXZpY2VfaWQiOiJkZXZpY2VfYSIsInBvbGljeV9saXN0IjpbInB
vbGljeV9hIl19.TbYWnLC0BWb3mw1ITxK6HcaOG2JB32tsOsJzh3RQgAYUFjXRsoZzDxABgp1ZR7Itplz5IWXdWNO9vUd
VoySGQOeM4lPgg8AY_8qLFB9CIL_Gj-
hB3f42bdJc5HiQZojW6336zsaQ7sK8X6QHqDVaP4rmFKPrS9ENLJCVfZ_3axw",
  "policy_list" : [ "policy_a", "policy_b" ]
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Successful Response DevicePolicyMapStrSigned

422

Validation Error HTTPValidationError

4.1.2 Policy

`GET /api/v1/policy`

Get available and user's devices policies (**getPoliciesApiV1PolicyGet**)

> This endpoint provides a way to get a list of the available policies and the ones already set on the user's devices. Available policies are all the ones set on the brain-iot domain to be associated to a specific device by the user.

Return type

UserDevicePoliciesAvailability

Example data

Content-Type: application/json

```
{
  "available_policy" : [ "policy_a", "policy_b", "policy_d", "policy_c" ],
  "device_policy_list" : [ {
    "device_id" : "device_a",
    "policy_list" : [ "policy_a", "policy_b" ]
  }, {
    "device_id" : "device_a",
    "policy_list" : [ "policy_a", "policy_b" ]
  } ]
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Successful Response UserDevicePoliciesAvailability

`POST /api/v1/policy`

Set user's devices policies (**setUserPoliciesApiV1PolicyPost**)

> This endpoint provides a way to set the selected policies on the user's devices as attribute of the user. The user will inherit the union of the policies of his devices.

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

body UserDevicePolicyMapStr **(required)**

Body Parameter —

Example data

Content-Type: application/json

```
""
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Successful Response

422

Validation Error HTTPValidationError

4.1.3 Service

`POST /api/v1/service/filter`

Get filtered list of allowed services (**filterServiceListApiV1ServiceFilterPost**)

> This endpoint provides a way to filter a list of services based on a list of policies signed by the brain-pep. This API will receive both a list of service_ids and a token containing the list of signed policies associated with the incoming message to be forwarded to the services.

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

body ServiceListStrPolicyToken **(required)**

Body Parameter —

Return type

ServiceList

Example data

Content-Type: application/json

```
{
  "service_list" : [ {
    "service_name" : "Service2",
    "scope_list" : [ "policy_a", "policy_b" ]
  }, {
    "service_name" : "Service4",
    "scope_list" : [ "policy_a" ]
  } ]
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Successful Response ServiceList

422

Validation Error HTTPValidationError

`GET /api/v1/service`

Get services and their policies (**getResourceServiceScopesApiV1ServiceGet**)

> This endpoint provides a way to retrieve the list of scopes/policies associated to a Service x. (Should be used as debug endpoint to know the policies associated to the registered services)

Return type

ScopedServiceList

Example data

Content-Type: application/json

```
{
  "service_scoped_list" : [ {
    "service_name" : "Service1",
    "scope_list" : [ "policy_b" ]
  }, {
    "service_name" : "Service2",
    "scope_list" : [ "policy_a", "policy_b" ]
  }, {
    "service_name" : "Service3",
    "scope_list" : [ "policy_c" ]
  } ]
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Successful Response ScopedServiceList

4.2 Models

4.2.1 `DevicePolicyMapStr` - DevicePolicyMapStr

DevicePolicyMap Model

device_id

String The device id

example: device_a

policy_list

array[String] The list of policies associated to the device

example: ["policy_a","policy_b"]

4.2.2 `DevicePolicyMapStrSigned` - DevicePolicyMapStrSigned

DevicePolicyMap Model

device_id

String The device id

example: device_a

policy_list

array[String] The list of policies associated to the device

example: ["policy_a","policy_b"]

signature

String The jwt token containing the list of policies signed by the authorization service

example:
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXZpY2VfaWQiOiJkZXZpY2VfYSIsInBvbGljeV9saXN0IjpbInBvbGljeV9hIl19.TbYWnLC0BWb3mw1ITxK6HcaOG2JB32tsOsJzh3RQgAYUFjXRsoZzDxABgp1ZR7Itplz5IWXdWNO9vUdVoySGQOeM4lPgg8AY_8qLFB9CIL_Gj-hB3f42bdJc5HiQZojW6336zsaQ7sK8X6QHqDVaP4rmFKPrS9ENLJCVfZ_3axw

4.2.3 HTTPValidationError - HTTPValidationError

detail (optional)

array[ValidationError]

- **ScopedService - ScopedService**

Model that use orjson to serialize and deserialize

service_name

String The name of the service

example: Service1

scope_list

array[String] The name of the service

example: ["policy_b","policy_c"]

4.2.4 ScopedServiceList - ScopedServiceList

Model that uses orjson to serialize and deserialize

service_scoped_list

array[ScopedService] List of services with associated scopes/policies

example:
[{"service_name":"Service1","scope_list":["policy_b"]},{"service_name":"Service2","scope_list":["policy_a","policy_b"]},{"service_name":"Service3","scope_list":["policy_c"]}]

4.2.5 ServiceList - ServiceList

Model that uses orjson to serialize and deserialize

service_list

array[ScopedService] List of services with associated scopes/policies

example:
[{"service_name":"Service2","scope_list":["policy_a","policy_b"]},{"service_name":"Service4","scope_list":["policy

_a"]}]

4.2.6 `ServiceListStrPolicyToken` - ServiceListStrPolicyToken

Model that uses orjson to serialize and deserialize

service_list

array[String] The list of services to be filtered

example: ["Service1","Service2","Service3","Service4"]

policy_sig_jwt

String The jwt token containing the list of policies signed by the authorization service.

*example:
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXZpY2VfaWQiOiJkZXZpY2VfYSIsInBvbGljeV9saXN0IjpbInBvbGljeV9hIl19.TbYWnLC0BWb3mw1ITxK6HcaOG2JB32tsOsJzh3RQgAYUFjXRsoZzDxABgp1ZR7Itplz5IWXdWNO9vUdVoySGQOeM4lPgg8AY_8qLFB9CIL_Gj-hB3f42bdJc5HiQZojW6336zsaQ7sK8X6QHqDVaP4rmFKPrS9ENLJCVfZ_3axw*

4.2.7 `UserDevicePoliciesAvailability` - UserDevicePoliciesAvailability

Model that uses orjson to serialize and deserialize

available_policy

array[String] The list of available policies on the system

example: ["policy_a","policy_b","policy_d","policy_c"]

device_policy_list

array[DevicePolicyMapStr] The list device-policies mapping

4.2.8 `UserDevicePolicyMapStr` - UserDevicePolicyMapStr

UserDevicePolicyMapStr Model

user_id

String User id of the devices owner

example: user_a

device_policy_list

array[DevicePolicyMapStr]

4.2.9 `ValidationError` - ValidationError

loc

array[String]

msg

String

type

String

5 Conclusions

This deliverable "D5.9 – Guidelines for privacy compliance and control in IoT services models" contains:

- a general conceptual model as a guideline for future research in the domain of data access control.
- the BRAIN-IoT approach to model access control policies, along with some research directions for future research activities.
- the BRAIN-IoT specifications for adopting the proposed approach for data access control.

The general conceptual model intends to be a guideline for future research in data access control for the purpose of privacy protection or for granting that data owners' requirements are respected in the context of a Marketplace of Data, where open or monetized data are made available to requiring services. Along with the Interoperability Layer and the Communication and Management Layer (see D2.7), the Privacy Control System may constitute an enabler for a Marketplace of Data (see D4.7) in the context of the Smart Cities.

The different focus of the of project and the constraints in terms of available resources prevent BRAIN-IoT for investing more on the realization of such type of Marketplace but, also according to the outcomes of the investigation performed in Task 4.3 – "WoT-based Marketplace building and business dynamics", it came out a possible business interest which the BRAIN-IoT Consortium considers worth to be further pursued in future research projects. Along with AI-based multi-agent and semantic reasoning mechanisms, such Data Marketplace can constitute a valuable semi-automated platform for negotiating data procurement by service providers, also adopting a Data Monetization paradigm.

Acronyms

Acronym	Explanation
AI	Artificial Intelligence
ESG	External Stakeholders Group
GDPR	General Data Protection Regulation
H2020	Horizon 2020
JWT	JSON Web Token
LSP	Large Scale Pilot
PIA	Privacy Impact Assessment

List of figures