Symbiosis of smart objects across IoT environments

688156 - symbIoTe - H2020-ICT-2015

1st Open Call – Technical Details

The symbIoTe Consortium

Intracom SA Telecom Solutions, ICOM, Greece
Sveučiliste u Zagrebu Fakultet elektrotehnike i računarstva, UNIZG-FER, Croatia
AIT Austrian Institute of Technology GmbH, AIT, Austria
Nextworks Srl, NXW, Italy
Consorzio Nazionale Interuniversitario per le Telecomunicazioni, CNIT, Italy
ATOS Spain SA, ATOS, Spain
University of Vienna, Faculty of Computer Science, UNIVIE, Austria
Unidata S.p.A., UNIDATA, Italy
Sensing & Control System S.L., S&C, Spain
Fraunhofer IOSB, IOSB, Germany
Ubiwhere, Lda, UW, Portugal
VIPnet, d.o.o, VIP, Croatia
Instytut Chemii Bioorganicznej Polskiej Akademii Nauk, PSNC, Poland
NA.VI.GO. SCARL, NAVIGO, Italy

For more information on this document or the symbIoTe project, please contact:
Sergios Soursos, INTRACOM TELECOM, souse@intracom-telecom.com

symbIoTe: Technical Details

symbIoTe addresses a challenging objective to create an interoperable Internet of Things (IoT) ecosystem that will allow for the collaboration of vertical IoT platforms towards the creation of cross-domain applications. Thus, it designs an *interoperable mediation framework* to enable the discovery and sharing of connected devices across existing and future IoT platforms for rapid development of cross-platform IoT applications. symbIoTe allows for *flexible interoperability mechanisms* which can be achieved by introducing an incremental deployment of symbIoTe functionality across the platform's space, which will in effect influence the level of platform collaboration and cooperation with other platforms within a symbIoTe-enabled IoT ecosystem. Syntactic and semantic interoperability represent the essential interoperability mechanisms in the future symbIoTe-enabled ecosystem, while organizational interoperability has different flavors within symbIoTe (platform federations, dynamic Smart Spaces and roaming IoT devices) to enable platform providers to choose an adequate interoperability model for their business needs.

The symbIoTe Architecture

The symbIoTe architecture is built around a layered IoT stack connecting various devices (sensors, actuators and IoT gateways) within Smart Spaces with the Cloud. Smart Spaces share the available local resources (connectivity, computing and storage), while platform services running in the Cloud will enable IoT Platform Federations (associations between two platforms) and open up the Interworking Interface[1] to third parties. The architecture comprises four layered domains, 1) Application Domain, 2) Cloud Domain, 3) Smart Space Domain and 4) Device Domain, as depicted in Figure 1. Hereafter we list the main functional objectives for each of these domains:

Application Domain (APP): enables platforms to register IoT devices which they want to advertise and make accessible via symbIoTe to third parties, while symbIoTe provides the means for discovery of IoT devices across platforms by its Core Services. We also envision domain-specific back-end services (enablers) to be placed in APP: They utilize the infrastructure provided by the underlying platforms to offer value-added services, e.g. data analytics on top of sensor data acquired from different platforms, which can ease the process of cross-platform and domain-specific application development (specifically for mobile and web applications).

Cloud Domain (CLD): provides a uniform and authenticated access to virtualized IoT devices exposed by platforms to third parties through an open API (Interworking Interface). In addition, it builds services for IoT Platform Federations enabling close platform collaboration, in accordance with platform-specific business rules.

[1] Interworking Interface is a symbIoTe defined interface which opens up platform resources as IoT Services in the Cloud Domain.

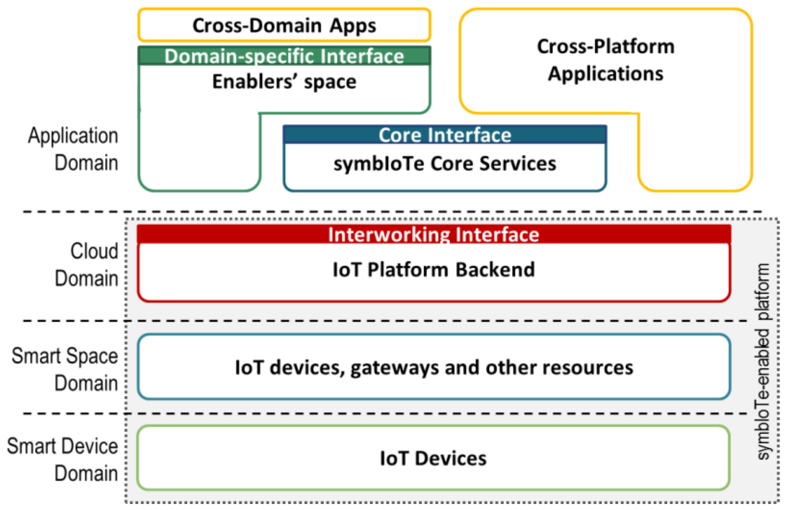

Figure 1. The symbIoTe high-level architecture

Smart Space Domain (SSP): provides services for discovery and registration of new IoT devices in dynamic local smart spaces, dynamic configuration of devices in accordance with predefined policies in those environments, and well- documented interfaces for devices available in smart spaces.

Smart Device Domain (SDEV): relates to smart devices and their roaming capabilities. We assume that devices have the capabilities to blend with a surrounding smart space while they are on the move. In other words, smart devices can interact with devices in a visited smart space, which are managed by a visited platform, in accordance with predefined access policies.

Interoperability Aspects

symbIoTe allows for flexible interoperability mechanisms which can be achieved by introducing an incremental deployment of symbIoTe functionality across the listed architectural domains (APP, CLD, SSP and SDEV). This approach will enable platform providers to choose an appropriate level of integration of symbIoTe-specific services within their platforms, which will in effect influence the level of platform collaboration and cooperation with other platforms within a symbIoTe-enabled ecosystem. For example, a platform may only choose to expose its Interworking Interface and selected IoT services to third parties in order to advertise them by using the symbIoTe Core Services, or it may opt for a closer collaboration with another platform by forming a platform federation. Platform federations require additional symbIoTe components to be included and integrated within a platform space in CLD.

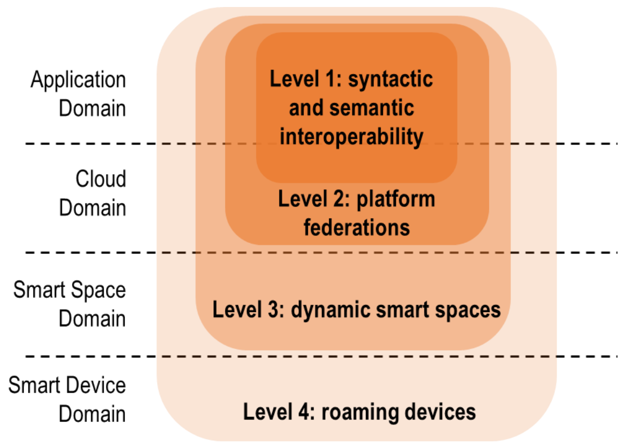
Application Domain
Cloud Domain
Smart Space Domain
Smart Device Domain
Level 1: syntactic and semantic interoperability
Level 2: platform federations
Level 3: dynamic smart spaces
Level 4: roaming devices

Figure 2. symbIoTe Compliance Levels

We define four different Compliance Levels (CLs) for IoT platforms, as depicted in Figure 2. They reflect different interoperability modes, which an IoT platform can support. Different interoperability modes affect the functionality which needs to be supported by symbIoTe-enabled platforms, and require specific symbIoTe components to be integrated within different domains.

Level 1 (L1) Compliant Platform: This is a "lightweight" symbIoTe CL since a platform opens up only its Interworking Interface to third parties to advertise and offer its virtualized IoT devices through the symbIoTe Core Services. It enables the syntactic and semantic interoperability of IoT platforms in a symbIoTe ecosystem, and affects only APP and CLD.

Level 2 (L2) Compliant Platform: This level assumes that platforms federate, which requires additional functionality to be included in CLD, for example for sharing/bartering of devices. The functionality provided at this level enables the so-called organizational interoperability.

Level 3 (L3) Compliant Platform: This CL assumes that platforms integrate symbIoTe components within their smart spaces to simplify the integration and dynamic reconfiguration of IoT devices within local spaces.

Level 4 (L4) Compliant Platform: This level offers support for device roaming and can enable the interaction of smart objects with visited smart spaces. A prerequisite for this level is that a platform is already Level 1, 2 & 3 compliant, so that smart spaces can discover new visiting devices and integrate them (e.g., grant access to certain local resources) in accordance with SLAs between platforms. Note that those platforms should thus be in a federation (Level 2), while smart spaces need the functionality for dynamic reconfiguration (Level 3).

L1 compliance can be directly mapped to *semantic and syntactic interoperability*, as identified in the IERC Whitepaper on Interoperability. L2, L3 and L4 platforms can clearly be categorized as systems supporting *organizational interoperability*. symbIoTe proposes here an original approach with finer granularity of organizational interoperability by placing specific interoperability concepts in the CLD for L2, in the SSP for L3 as well as in both SSP and SDEV for L4 compliance. In particular, L2 platforms form platform federations, L3 platforms support dynamic and reconfigurable smart spaces, while L4 platforms support roaming of smart devices which can use services in visited smart spaces. To achieve L2 compliance, a platform should first adhere to L1 compliance, while an L4 platform requires a full symbIoTe framework (i.e., an L4 platforms is also L1, L2 and L3 compliant).

Technical requirements for the 1st Open Call

Since **symbIoTe's 1st Open Call is seeking for platforms which will become Level 1 Compliant**, hereafter we focus on explaining the services placed in two domains, APP and CLD, which are needed to accomplish this compliance level. IoT platforms which want to become part of the symbIoTe ecosystem need to integrate the symbIoTe Interworking Interface with its existing components running in CLD. This enables semantic interoperability and open access to IoT services which a platform chooses to register and make discoverable via the symbIoTe Core Services. The access to those devices stays under the control of a platform provider.

APP is designed to offer a unified view on different platforms to a new generation of cross-platform IoT applications. This is achieved by the symbIoTe Core Services that enable the discovery of IoT devices across platforms. Note that the Core Services store and manage only IoT resource descriptions (i.e. resource metadata), while the access to those resources (e.g., sensor data and actuation primitives) is provided by the underlying platforms. Thus, the symbIoTe Core Services are in close interaction and collaboration with the services provided within the Cloud Domain which offer the actual access to virtualized IoT resources. In addition to the search functionality, the Core Services implement symbIoTe specific authentication and authorization methods providing the means for secure access to underlying platform-specific resources.

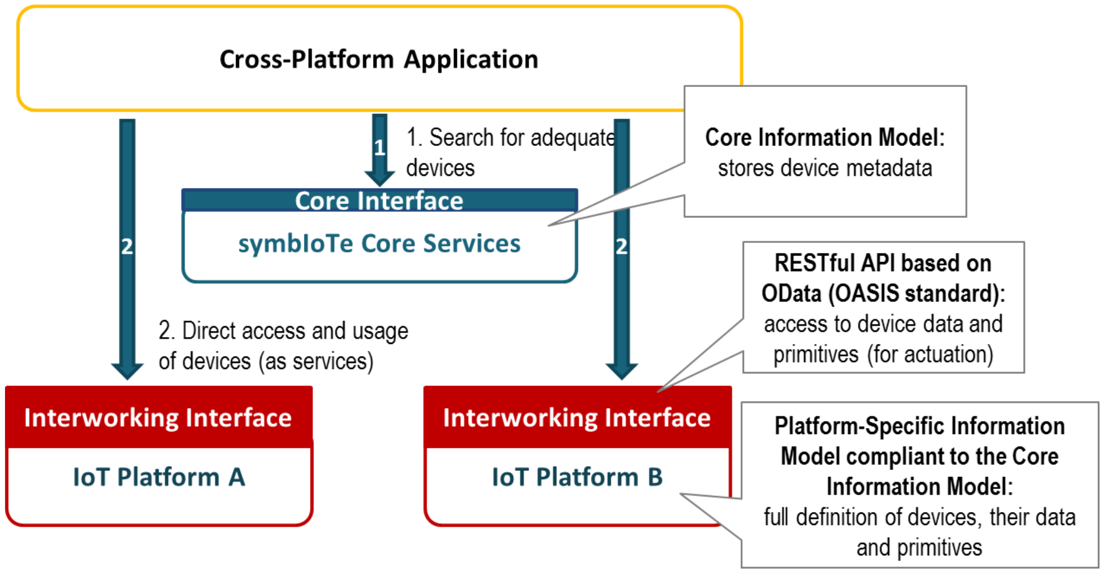
Cross-Platform Application
1
1. Search for adequate devices
Core Interface
symbIoTe Core Services
Core Information Model: stores device metadata
2
2
2. Direct access and usage of devices (as services)
RESTful API based on OData (OASIS standard): access to device data and primitives (for actuation)
Interworking Interface
IoT Platform A
Interworking Interface
IoT Platform B
Platform-Specific Information Model compliant to the Core Information Model: full definition of devices, their data and primitives

Figure 3. Illustrating Level 1 Compliance

Figure 3 shows the benefits of L1 compliance by an example depicting two platforms A and B using the symbIoTe Core Services. When an application searches for devices and identifies adequate ones, the application accesses the devices offered by the two platforms through the Interworking Interface. In other words, cross-platform applications i) use the symbIoTe Core Services to find adequate devices across platforms and ii) access, integrate and use those devices through a uniform and open interface. We are currently supporting a part of the OData standard for pull-based access to platform devices and will define a push-style mechanism for continuous delivery of sensor data to applications.

symbIoTe Information Model. To reach the goal of semantic interoperability, i.e., “the ability of computer systems to exchange data with unambiguous, shared meaning”, symbIoTe defines three types of Information Models related to the description of resources which platforms want to expose through symbIoTe:

- The **Core Information Model** (**CIM**) defines all information that symbIoTe needs to understand on an abstract level, e.g. a class *Sensor* that is related to the class *Location* via a relation/property *hasLocation*.
- **Platform-Specific Information Models** (**PIM**s) are platform-specific and contain all classes and their relations which a platform wants to expose through symbIoTe, e.g., a custom property of a *Sensor* called *hasColor*. A PIM is an extension of the CIM and must comply to it, e.g., by adding new properties to classes of the CIM.
- The **Best Practice Information Model** (**BIM**) is a special form of a PIM that is predefined by and shipped with symbIoTe. Its purpose is to provide a default and simplified way to use symbIoTe whenever a platform does not require having a custom PIM.

The design of the CIM and BIM is currently work in progress; they will be available in Q1 2017. To give at least an impression of these models, hereafter we present the current version of the CIM.

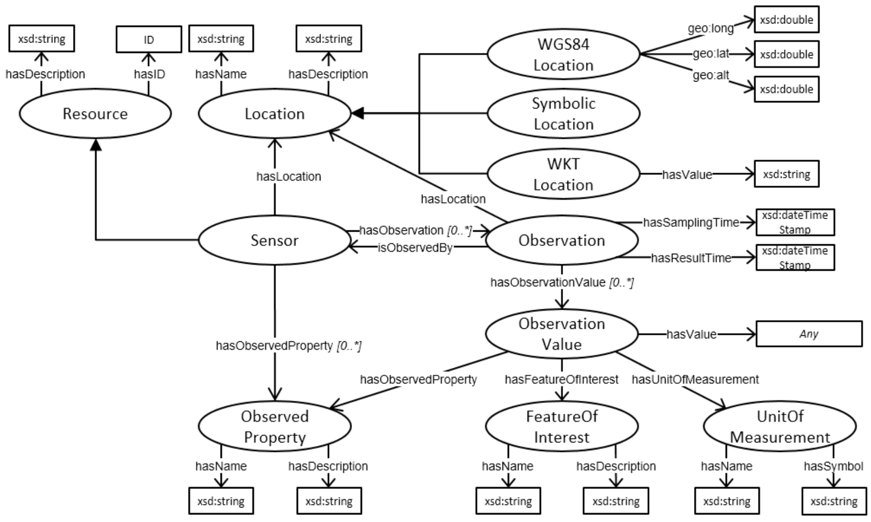

Figure 4. The current version of the Core Information Model (CIM)

When exposing IoT devices to symbIoTe Core Services, devices need to be semantically described. This can be done using one of the following two approaches:

- Using the BIM: As the BIM is a complete information model and known to symbIoTe you will be able to register devices via a simple REST-based call with JSON payload.
- Using a custom PIM: If the BIM does not fit your needs, you can describe the exposed resources of your platform using a custom PIM which must be compliant to the CIM. This must be expressed as an ontology using the Resource Description Format (RDF)[2]. The symbIoTe team can provide support for this task.

[2] https://www.w3.org/RDF/

symbIoTe components for L1 compliance. symbIoTe Core Services are composed of APP components which enable the interaction between third-party applications and platforms. The main foreseen services are the following:

1. Administration Service - provides a web-based GUI as well as an API for platform administrators. The provided functionality will include: creation and management of the platform and its properties, generation of platform credentials for authentication.
2. Registry - repository for storing all platform-related metadata. It provides an API for registering and updating of platform's devices/services which are described using the symbIoTe Information Model. During the registration process, symbIoTe/specific unique IDs are generated for each device/service.
3. Search Engine enables applications to find relevant registered devices/services within the Registry.
4. Resource Monitor tracks the availability of registered devices in order to ensure their availability.
5. Resource Access Monitor that tracks information about device popularity.
6. Core Authentication and Authorization Manager authenticates third-party users and applications (i.e., users and applications that are not associated with any IoT platform) and provides attributes required to access symbIoTe Core Services.

The Core Services are designed and implemented based on the microservices architecture (using Spring Cloud), having in mind the scalability and distributed characteristics of the architecture. The listed services will be offered and managed by the symbIoTe consortium for testing purposes. The ones which are of particular interest to platforms are the Administration Service, Registry and Resource Monitor, since they are accepting requests from symbIoTe-enabled platforms.

The following CLD components will facilitate the integration of a new platform with symbIoTe. All interfaces mentioned here are part of the Interworking Interface, which exposes platform devices as IoT services. The symbIoTe consortium will provide interface definitions for all four components (REST/JSON) together with a supporting Java library implementing non-platform-specific parts of those components.

1. Registration Handler - handles platform-side registration of devices with the Registry by using the symbIoTe Information Model. An interface description will be provided which needs to be implemented by the platform owner. We will also provide a blueprint implementation in Java which will require a platform-specific plugin to be implemented by a selected applicant.
2. Resource Access Proxy - enables access to devices/services (i.e. resources) on the platform side which are registered within the Registry. An interface description will be provided which needs to be implemented by the platform owner so as to forward requests to platform-specific actions and return results (resource data) conforming to symbIoTe Information Model. It is also responsible for the access control mechanism, based on the Attribute-based access control (ABAC) criteria. Specifically, it defines the access policy, controls the list of attributes sent by the application during the resource access process, and allows/denies the access to the resources.

3. Platform Authentication and Authorization Manager (AAM) offers authentication and authorization mechanisms on the platform side. Authorization is based on the ABAC mechanisms which assign ‘attributes’ to all client applications and entities in the system. Accordingly, the access to resources is controlled through the Access Control Policies. An access policy, defining a specific combination of attributes needed to grant access to a resource, is assigned to each resource by the producer of that resource. Therefore, a client application may be granted access to a resource only if it possesses a set of attributes that match the predefined access policy. In symbIoTe this policy can contain at the same time attributes assigned to users and objects as well as to particular environmental conditions connected to a request (i.e., specific time constraints). Note that AAM only releases attributes. The management of the policy is, instead, handled by the Resource Access Proxy. The symbIoTe consortium will provide a Java implementation of the platform AAM together with a library for handling security requests on components on the platform side.
4. Monitoring component monitors the status of registered devices/services and reports the status periodically to the Resource Monitor within the Core Services.

The aforementioned four components will need to be implemented by the applicants and integrated with existing platform-specific components. The symbIoTe team will provide interface definitions and component implementations in Java that require further extensions to be integrated with platform-specific components. We acknowledge the fact that all extensions written in other languages require more resources.

Summary. There are two major requirements which L1 platforms need to fulfill: 1)the existing internal information model of the platform must be mapped either to the BIM or to a custom PIM (if needed) and 2) a platform must integrate the symbIoTe Interworking Interface to open up access to its IoT devices exposed through symbIoTe Core Services (syntactic interoperability).

To make a platform Level 1 compliant, a platform provider needs to do the following:

- Analyze the symbIoTe solution for Level 1 compliance
- Expose the description IoT devices accessible through symbIoTe either using the symbIoTe BIM or a custom PIM (in the form of an ontology).
- Integrate the Interworking Interface required for Level 1 compliance within the CLD domain with their platforms
- Provide feedback and comments which can improve and simplify the process of creating L1 platforms

What will symbIoTe offer to platform providers to extend their platforms to become L1 compliant?

- A documented design and prototype implementation of the required symbIoTe Core Services
- A documented Interworking Interface and corresponding components in Java for CLD
- An example procedure for mapping of an existing information model to the symbIoTe information model

- A documented example explaining the process for creating a L1 platform based on the example of the open-source OpenIoT platform