| | |
|---|---|
| Project Acronym: | **VICINITY** |
| Project Full Title: | **Open virtual neighbourhood network to connect intelligent buildings and smart objects** |
| Grant Agreement: | **688467** |
| Project Duration: | **48 months (01/01/2016 - 31/12/2019)** |

# 2nd Open Call. Technical Details

**The VICINITY Consortium**
1. TU Kaiserslautern (Coordinator) (Germany)
2. ATOS SPAIN SA (Spain)
3. Centre for Research and Technology Hellas (Greece)
4. Aalborg University (Denmark)
5. GORENJE GOSPODINJSKI APARATI D.D. (Slovenia)
6. Hellenic Telecommunications Organization S.A. (Greece)
7. bAvenir s.r.o. (Slovakia)
8. Climate Associates Ltd (United Kingdom)
9. InterSoft A.S. (Slovakia)
10. Universidad Politécnica de Madrid (Spain)
11. Gnomon Informatics S.A. (Greece)
12. Tiny Mesh AS (Norway)
13. HAFENSTROM AS (Norway)
14. Enercoutim – Associação Empresarial de Energia Solar de Alcoutim (Portugal)
15. Municipality of Pylaia-Hortiatis (Greece)

**© Copyright 2018, the Members of the VICINITY Consortium**
*For more information on this document or the VICINITY project, please contact us in the following* link

European Platforms Initiative
Co-founded by the Horizon 2020 programme of the European Union

## Executive Summary

The VICINITY is built around the concept of connecting different IoT ecosystems through an open gateway API (providing interoperability as a service[1]) which enables interaction with IoT objects (devices and value-added services[2]) from other different ecosystems **as if they were their own**. The VICINITY interoperability services inter-connecting IoT ecosystems creates a common environment where value-added services utilizing different devices can be deployed **and can operate across different domains** (Figure 1).

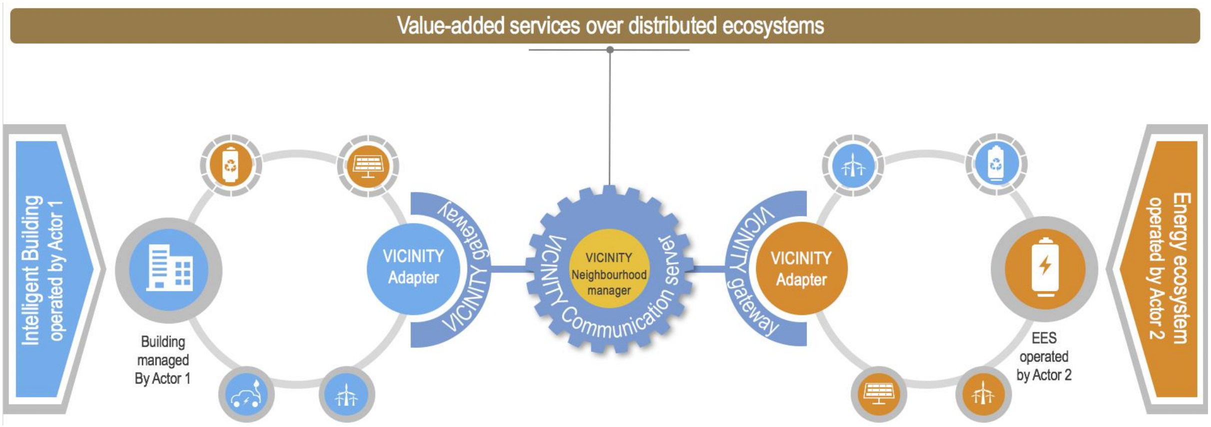


**Figure 1 VICINITY Concept**

In Figure 1, two separate ecosystems are presented: intelligent building and energy IoT ecosystems. Each of these ecosystems is integrated with the VICINITY by its VICINITY Adapter through the VICINITY Open Gateway API. Based on the setup of the virtual neighbourhood in VICINITY Neighbourhood Manager, VICINITY Adapters may access remote devices, for example a battery in an Energy ecosystem, and emulate it as a part of their ecosystem. Moreover, IoT objects shared by a VICINITY Adapter within a virtual neighbourhood may be accessed by value-added services to provide cross-domain services using common semantics based on VICINITY ontology (http://vicinity.iot.linkeddata.es/vicinity/ [3]).

[1] Interoperability as a service translates integrated IoT ecosystem information model into common VICINITY model based on semantic description of connected devices and services.

[2] Value-added service in context of VICINITY is defined as a piece of software that implements an algorithm (from a simple calculation/data processing to some advanced techniques such as clustering/big data analytics, data storage and auditing etc.). These services may provide the User interfaces to end-user in order to view notifications, statistical data, processed data etc. over collected data from the available integrated IoT infrastructure (IoT devices, sensors etc.).

[3] Note, that due implementation stage of the Project and evolution of the VICINITY ontology even after the life time of the project, VICINITY ontology is subject of change.

To make IoT infrastructure comply with VICINITY (such as building or energy ecosystems presented on previous figure) the following major requirements needs to be fulfilled:

1. map internal information model of devices and services exposed to VICINITY to the VICINITY common format;
2. integrate IoT infrastructure through the VICINITY Adapter to the VICINITY Open Gateway API to expose IoT objects.

An IoT ecosystem provider can receive the following benefits from being VICINITY compliant:

- access shared/exposed devices from other IoT ecosystems in different domains or based on different technologies;
- expose device capabilities towards cross-domain value-added services to extend the benefits of the connected devices for the end-user(s).

## Content

## 1. VICINITY Architecture

The objective of the VICINITY architecture is to facilitate interoperability between different IoT infrastructures' devices and to software-enable value-added services through a peer-to-peer (P2P) network of VICINITY Nodes. Each VICINITY Node provides access to IoT infrastructure and/or value-added service. Once the IoT infrastructure is integrated into the VICINITY neighbourhood through the VICINITY Node, devices connected to the infrastructure become accessible through the VICINITY Neighbourhood Manager in the VICINITY Cloud. In VICINITY Neighbourhood Manager IoT infrastructure owner can define sharing access rules of devices and service (i.e. has direct control over his or her devices). Based on these rules he or she creates social network of devices and service called "virtual neighbourhood".

The VICINITY Nodes create a controlled VICINITY Peer-to-peer (P2P) Network based on these rules defined by VICINITY Neighbourhood Manager (Figure 2 – yellow and blue arrows) in VICINITY Cloud. In VICINITY P2P Network, VICINITY Nodes communicate user data directly between each other (Figure 2 – red arrows). Moreover, the VICINITY P2P Network support VICINITY Node with security services (such as end-to-end encryption, data integrity, etc.) to ensure security and privacy of exchanged user data.

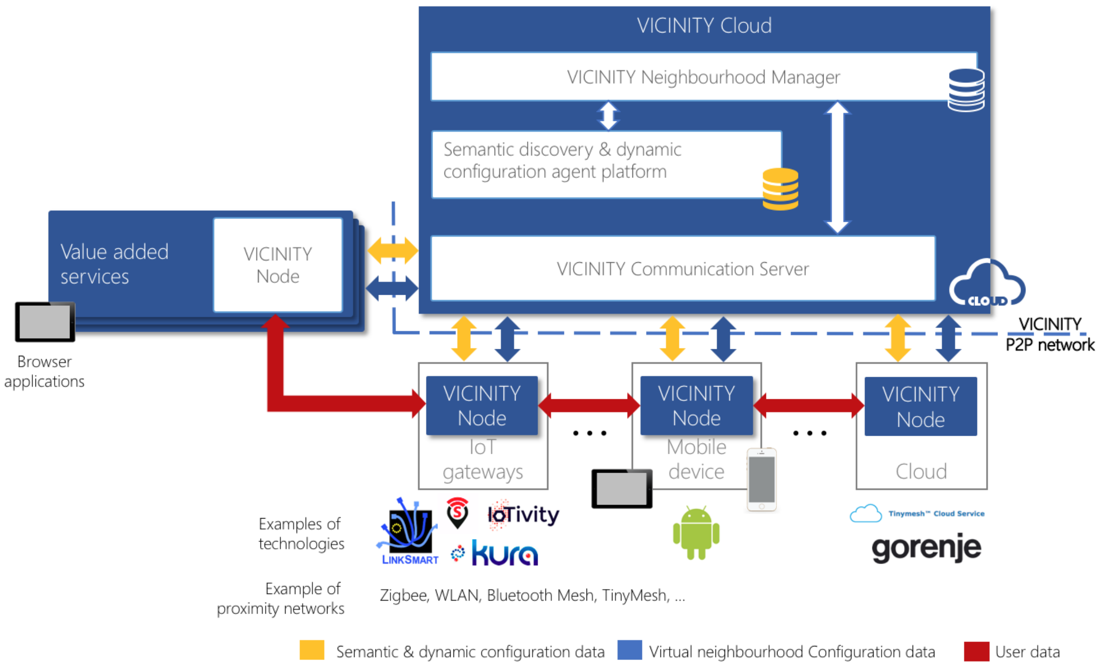


**Figure 2 High-level architecture of the VICINITY**

The VICINITY provides semantic interoperability features to facilitate exchange of user data between IoT devices and value-added services to overcome technology differences between each connected IoT ecosystem. Thus, communication with each device or service via VICINITY P2P Network is standardized regardless of the technology the device or service is connected to a VICINITY Adapter. This semantic interoperability approach is based on the work being done by the Web of Things (WoT) WG[4], where a proposal for describing, exposing and consuming *web things* by leveraging Semantic Web technologies is in development. Such web things are things that can be accessed through the Web, either physically or abstract.

One of the pillars of the W3C WoT is the Thing Description (TD), which aims to be a standard frame to support the description of *web things* semantically to make them interoperable. Thus, TDs are expected to cover the following aspects:

- Semantic meta-data, so to explicitly specify the semantics of a web thing;
- Thing's interaction resources: property, action and event;
- Security including concrete prerequisites to access things are stated;
- Communications, i.e., what kind of protocols and data exchange formats are supported, and which endpoints are exposed to give access to the existing interaction resources of a web thing.

Example of how the integrated IoT infrastructure will be accessed by other peers in VICINITY P2P Network is elaborated in following sections (1.4).

## 1.1. VICINITY Cloud

The VICINITY Cloud enables IoT infrastructure operators and Service providers to configure a virtual neighbourhood of connected devices and value-added services including the setup of sharing access rules between them through the user-friendly interface of VICINITY Neighbourhood Manager. Configuration of the virtual neighbourhood and sharing access rules are used by VICINITY Communication Server to setup communication channels between each VICINITY Node to control exchange of user data. IoT infrastructure operators and Service providers can search for devices and services in virtual neighbourhood based on semantic description of device properties, actions, events and service products & required inputs stored in semantic repository. Moreover, IoT operators, System integrators and Services providers can register the VICINITY Nodes (registration of the application API) to communication in peer-to-peer. The selected VICINITY Neighbourhood Manager functionalities are available through VICINITY Neighbourhood Manager API[5] as well. The API can be used by any value-added services to improve the user experience for its end-users.

## 1.2. VICINITY Node

A VICINITY Node is the set of software components which maintains the user data exchange between peers in the VICINITY P2P network based on configuration of the virtual neighbourhood and sharing rules received from VICINITY Communication Server. For that purpose, VICINITY Node consists of the following 3 main components:

[4] https://www.w3.org/WoT/IG/
[5] https://vicinityh2020.github.io/vicinity-neighbourhood-manager-api/

- VICINITY Gateway API and Communication Node;
- VICINITY Agent;
- VICINITY Adapter.

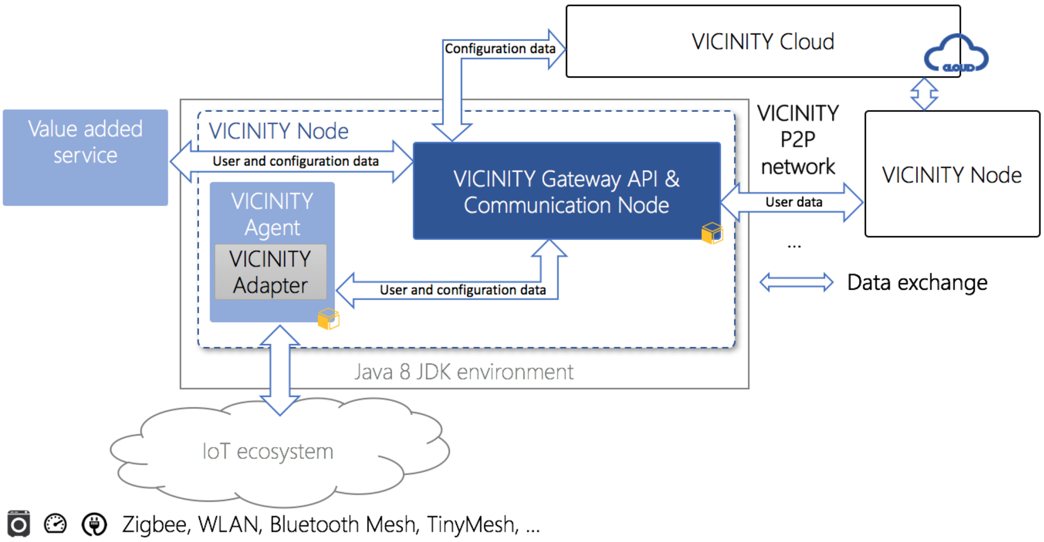


**Figure 3 Logical architecture of the VICINITY Node**

The *VICINITY Gateway API and Communication Node* service provides the following set of JSON HTTP REST[6] services in the VICINITY common format and data model:

- Registry service to register devices and value-added services using their simple semantic description as JSON document in VICINITY;
- Search service in virtual neighbourhood of connected IoT infrastructures and value-added services;
- Services to read/write properties, performing actions and process events from shared devices device/service from connected IoT infrastructure or value-added service;
- Expose properties, actions and events of device/service to VICINITY through simple REST API;
- Authentication services for the IoT infrastructure (authentication of the application) and devices and value-added services;
- Status check service.

The *VICINITY Adapter* is a component provided by IoT infrastructure owner or respective system integrator. The VICINITY Adapter provides simple API, which has to be implemented for every adopted infrastructure. The core responsibilities of the Adapter API are:

- to provide the description of IoT objects (devices, services) of infrastructure in common VICINITY format, which enables VICINITY to create internal models of used IoT objects in uniform way;
- to provide access to properties, actions and events of devices provided by infrastructure.

[6] https://vicinityh2020.github.io/vicinity-gateway-api/#/

The purpose of the VICINITY Adapter is to simplify the adoption of IoT infrastructure in as simple way as possible. The Adapter is used just to discover IoT objects in infrastructure and to communicate with IoT objects. Once the IoT infrastructure owner/system integrator provides the Adapter component implementing simple prescribed API, the IoT infrastructure can be easily adopted. The full VICINITY functionality is then managed by VICINITY Agent component.

The *VICINITY Agent* is the wrapper for the VICINITY Adapter. The VICINITY Adapter provides the full VICINITY specific functionality, as managing communication via P2P network, semantic discovery of IoT objects, semantic search of IoT objects, communication with IoT objects within the infrastructure, where each VICINITY specific interaction with IoT objects is translated in the VICINITY Adapter calls.
Simply speaking, the Adapter handles only very basic communication with IoT infrastructure by implementing simple API. Agent manages full VICINITY interactions with IoT infrastructure by translating this interaction into the Adapter API services.

## 1.3. VICINITY Value-Added Services / Microservices

Value-Added Services (VAS) could be defined as a piece of software that implements an algorithm (from a simple calculation/data processing to some advanced techniques such as clustering/big data analytics, data storage and auditing etc.). Moreover, there could be collaboration between the VAS including the exchange of data and outcomes from the algorithms already implemented by other VAS of the same use case. These services may also supply the User interfaces that each actor of the use case will be provided with in order to view notifications, statistical data, processed data etc. VAS collect data, in order to further process them, from the IoT infrastructure (IoT devices, sensors etc.) available in each Use Case.

VICINITY VAS aims to demonstrate the capabilities of the VICINITY Platform in the IoT world and to maximise the potential of adoption of the project's results. Thus, VICINITY will shape solutions regarding user preferences, real operational needs and the appropriate legal, security and privacy requirements augmenting VICINITY potential for wide adoption by user communities.

The four existing Pilot Cases cover various domains in Building, Energy, Transportation and Health sectors. Each of the Value-Added Services has been implemented in the scope of each Pilot Case and will add up to the primary Use Cases. Value-Added Services serve as a demonstration in order to highlight how VICINITY Platform and the underlying components, enable the realization of cross-domain applications, as a proof-of-concept on the potential brought by VICINITY. Benefits and value creation of these services can evolve by exploiting the unleashed volume of semantically enhanced information stemming from the emerging IoT ecosystems connected to the VICINITY interoperability platform.

## 1.4. How semantic interoperability works in VICINITY

VICINITY implements several services that conform the semantic interoperability, entailing that, IoT infrastructures integrated in VICINITY that are in the neighbourhood of a Gateway can be automatically discovered and accessed without any prior knowledge of them. In addition, the responses are expressed according to the VICINITY Ontology, therefore, the outputs are always expressed with the same model which eases the task of understanding and processing them.

In order to make a new IoT infrastructure interoperable in VICINITY no special effort is required. When a new IoT infrastructure is exposed by a VICINITY node the node itself submits a Thing Ecosystem Description (TED) that describes the set of IoT objects included in the infrastructure to Semantic Repository, i.e., a core component of VICINITY. These descriptions are later used by the Gateway API Services, another core component of VICINITY, to find IoT objects relevant to a query provided by a user or a value added service (VAS).

Semantic interoperability allows users or VASs to query the IoT objects allocated in their neighbourhood using the interoperability services through a Gateway API, as depicted in Figure 4. Gateway APIs implement a SPARQL endpoint that can receive a query (using the W3C standard SPARQL query language), such query is processed by the Gateway and finally the query results are returned with the aggregated data of IoT infrastructures that are relevant to answer the query and are allocated in the neighborhood.

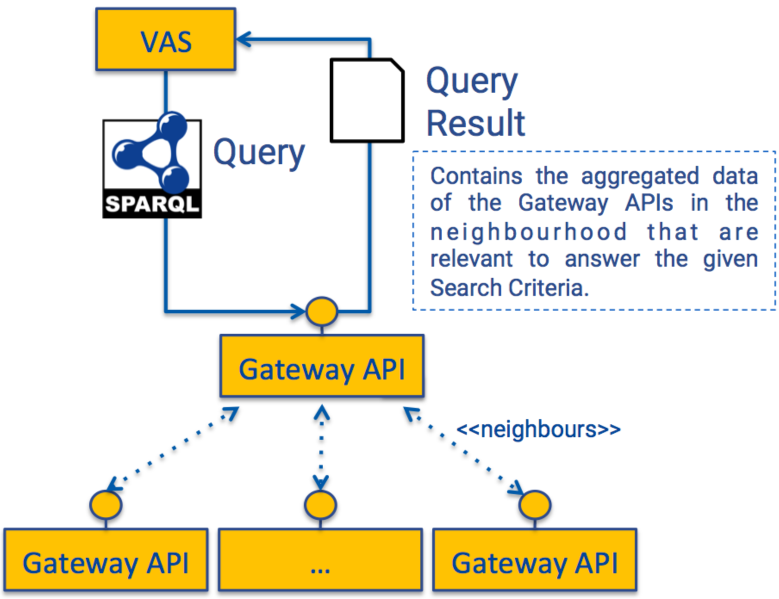


**Figure 4 Semantic interoperability solving a query**

To process a given SPARQL query the Gateway API relies on the Distributed Query Client as depicted in Figure 5. The Distributed Query Client forwards the query that was sent to the Gateway API to the core component Gateway API Services as shown in step 2 in figure 5. Then, the Gateway API Services process the query and interacts with the Semantic Repository, where previously all the IoT infrastructures nodes submitted their TED. As a result, the Gateway API Services provides the Distributed Query Client with two elements, on the one hand, it returns a TED containing the description of only the relevant IoT infrastructures required to answer the given query that are allocated in the neighborhood, on the other hand, it provides a Query Plan that specifies how to access the data of the relevant IoT infrastructures (step 3 in Figure 5).

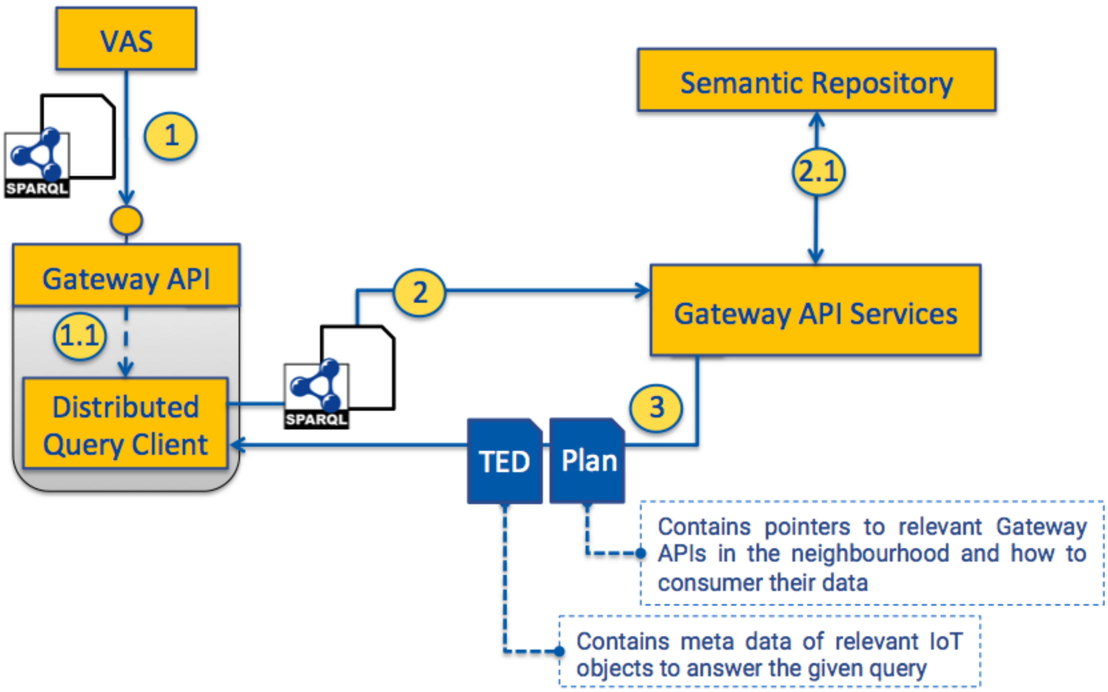


**Figure 5 Semantic interoperability discover relevant IoT objects in the neighbourhood**

Once the Distributed Query Client receives the TED and the Query Plan from the Gateway API Services it gathers the data from the Gateway APIs that are relevant to answer the given query (see step 4 from Figure 6). Finally, the Distributed Query Client aggregates all the data and lift it to the semantic data format RDF, check respectively step 5 and 6 from Figure 6.

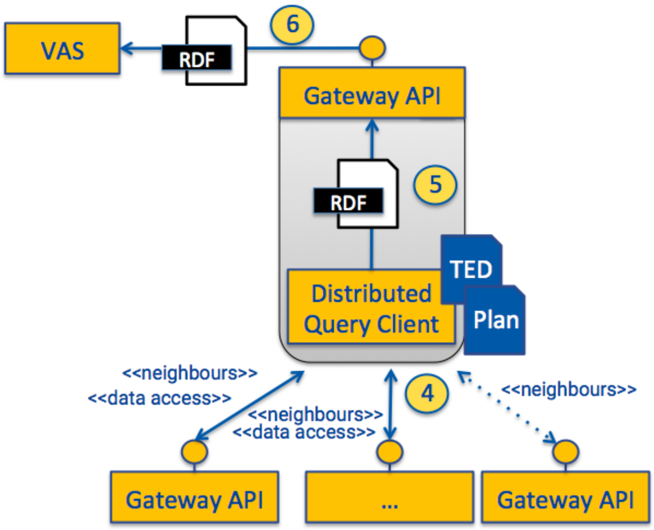


**Figure 6 Semantic interoperability accessing relevant IoT objects data**

In conclusion the Semantic Interoperability offers to users and VAS the opportunity to discover IoT infrastructures in their neighborhood with no prior knowledge of their existence (for instance a query may ask for all the sensors that are thermometers). In addition, the queries sent to the Gateway API Services may also access the real-time data of the sensors (for instance a query may ask for the average temperature of the thermometers). Furthermore, all the outputs follow the VICINITY Ontology, easing the task of processing and understanding the data.

# 2. VICINITY Pilots[7]

## 2.1. Oslo (Norway). Smart Building.

This use case for the VICINITY platform will be demonstrated at the M:6 co-working space in Moss/Oslo in Norway. M:6 provides open plan co-working space as well as traditional office spaces, their aim is to provide a unique environment where people from different companies can share knowledge and inspiration from each other. M:6 is operated and managed by CWI. They are eager to incorporate new technologies that can improve daily work and management of the building.

This VICINITY use-case will be centred around using IoT technologies to improve resource management, resource consumption and predictive operations in buildings. Using wireless door sensors, as well as wireless electricity- and water meters, the two VAS will inform and alarm the management team about typical

[7] For more related information regarding Value-Added Services and Use Cases pre Pilot Site refer to the deliverable https://vicinity2020.eu/vicinity/content/d51-value-added-services-definition-requirements-and-architectural-design

and non-typical situations. The information and alarms will enable them to target their cleaning efforts, shed electricity loads, discover water leaks and track their resource consumption in real time, thus saving time and money.

## 2.2. Tromsø (Norway) – Neighbourhood Smart Parking Assisted Living ecosystem

This pilot will take place at the “Teaterkvarteret 1. akt”. This is a newly constructed cluster of buildings near the central part of Tromsø, an arctic city located far north of the Polar Circle in Norway. The site includes three 6-store buildings with a total of 38 owner sections8, of which all are apartments. The 8 apartments are either owned or assigned for residents that are either elderly or disabled. The underground garage facility has 32 parking spaces, of which, 7 are allocated for larger vehicles, and 2 have electrical charger ports.

The use-case will have specific focus on managed healthcare apartments, and demonstrate how transport information and building data can be integrated with assisted living through agreements with vehicle space owners and other stakeholders like community parking operator.

In the demonstrated solution, shared and prioritized parking space, booking, traffic analysis, customized and messaging services based on authorization and access data will be managed according to conditional rulesets. The smart parking sensors will report proximity and temperature. The datasets will be used for big data analysis to identify usage profiles and generate traffic forecasts that will be available to building managers, traffic controls and virtual neighbourhoods. Booking, configuration and status of parking space will be handled through different devices and user platforms such as mobile apps.

## 2.3. Martim Longo (Portugal) - Neighbourhood GRID ecosystem

The Use Cases in the Martim Longo pilot include several buildings: the Solar Lab located on the Solar DEMO Platform, a cluster of the Municipality managed buildings - a secondary School, a Retirement home and a Sports Centre including a Swimming Pool, and a private home. This cluster of Municipal buildings and the private home form a neighbourhood both digitally and geographically. The cluster in combination with the Solar Lab would form a setting for use cases, which were first identified in D1.3 and further developed during 2017. The use cases also underwent stakeholder prioritisation and subsequent analysis and further development.

[8] http://teaterkvarteret.no/1.akt/vedtekter/

Three distinct categories of services/solutions that will form the base of Value Added Services enabled by VICINITY, address a separate stakeholder segment within the public or private sector. The public sector services are specifically focused on Municipal asset management (building use and resource consumption) and IoT related Services delivery for Municipalities that will be subsequently rolled out towards the wider public audiences they serve. The O&M IoT enabled services will be demonstrated at the Solar Platform DER RES site. Decarbonisation of the energy sector and the drive towards reducing operation and maintenance cost of distributed assets is at the core of this VAS.

The concept of IoT enabled equipment leveraging (use of resource for secondary use) within the neighbourhood at municipal level is being explored. The Solar Lab facility is a model building facility with a DER RES system (self-consumption) which will serve as a data input and source. It will demonstrate consumption flexibility and serve as a model for a community grid towards the Municipal Buildings Cluster.

Infrastructures participating in the pilot´s use cases are located at Solar Lab facility of Solar Demonstration Platform of Martim Longo, Municipal Buildings Cluster (Sports Centre - Swimming Pool, Retirement home, and School) and in the private home within the Municipality.

The three categories of VAS are expected to give rise to further developments in innovation and additional solutions rollouts through so-called Labs approach beyond the project duration.

## 2.4. Pilea-Hortiatis (Greece) – eHealth & Assisted Living

The pilot case will be demonstrated in the municipality of Pilea-Hortiatis of Northern Greece, with the participation of a number of targeted people, who will be identified by the municipality health care services. The municipality has 70.110 inhabitants in an area of 167.800 km2 and is a pioneer in e-health services in Greece. It is a member of the "National Inter-municipal Network Healthy City - Health Promotion (NINHC-HP)", which is certificated by the World Health Organization (WHO) as a National Range Cities Network and it is also offering a program for elderly and middle-aged people called "e-Help at Home", in order to allow Pilea-Hortiatis citizens to live an independent and healthy life. The program offers monitoring of the weight and blood pressure of citizens registered to the program, as well as two notification services in case of emergency (panic/fall button, positioning of an Alzheimer patient).

In order to further improve the municipality's existing activities towards assisted living and preventive medicine and motivate more citizens to participate, VICINITY pilot case will take advantage of the existing facilities and extend them with more monitoring devices and six new VAS. Eligible users are patients with chronic diseases such as respiratory, heart diseases, hypertension, diabetes, third aged people or any citizen on the grounds of prevention, as people with high cholesterol or high blood pressure.

# 3. Technical requirements for 2nd open call

This section defines the following technical requirements which needs to be fulfilled by applicant to make Value-Added Services compliant to VICINITY and is the part of the technical delivery of the proposal:

1. to specify the business logic and algorithm behind the Value-Added Service or Microservice they will implement
2. to specify service description for each supported Value-Added Service which will be implemented
3. to specify device description for each supported device which will be exposed and access through VICINITY and being use as demonstrator of the integration – **only if they will integrate new devices to infrastructure;**
4. to implement and integrate the VICINITY Adapter of the integrated Value-Added Service which will translate ecosystem communication into common VICINITY communication format;
5. connect real Value-Added Service and devices (**only if they will integrate new devices to infrastructure**) into VICINITY**;**
6. to demonstrate usage of Value-Added Services or Microservices.

## 3.1. Specify Value-Added Services for existing pilots or Microservices

The applicant shall:

- **provide description of Value-Added Services or Microservices** to be exposed to VICINITY through the VICINITY Adapter.

A description of the selected implementation algorithm and the business logic behind the service should be provided along with functionalities delivered e.g. endpoints exposed and the hosting of the service.

## 3.2. Specify service description templates

The applicant shall:

- **provide service description in VICINITY common format of each testing service** to be exposed to VICINITY through the VICINITY Adapter[9].

[9] https://github.com/vicinityh2020/vicinity-agent/blob/master/docs/TD.md

A service description is defined as a simple JSON document including at a least identification of the service and a list of specifications of the interaction patterns with the devices such as: properties, actions and events. The specification of the interaction patterns defines the REST service, where the VICINITY Gateway API will interact with the VICINITY Adapter to reach exposed specific service.

### 3.3. Specify device description templates for each supported device type

The applicant shall:

- **provide description in VICINITY common format of each testing device** (if new ones integrated) to be exposed to VICINITY through the VICINITY Gateway API.

A device description is defined as a simple JSON document including at a least identification of the device and a list of specifications of the interaction patterns with the devices such as: properties, actions and events. The specification of the interaction patterns defines the REST service, where the VICINITY Gateway API will interact with the VICINITY Adapter to reach exposed specific device.

### 3.4. Implement and integrate the VICINITY Adapter for the Value-Added Service

The applicant shall:

- **implement the VICINITY Adapter** which will translate the IoT infrastructure information model and its interaction with the VICINITY Common format provided by VICINITY Gateway API. It will include the service description (see section 3.2);
- **integrate VICINITY Gateway API services** to extend the required by interaction nature between IoT ecosystem and value-added services. However, note that, registry, authentication services and status check service are mandatory. For example, for remote access of shared devices through VICINITY, the services to access devices are required. However, on the other hand, in this example integrate exposing service is not required.

Note that, currently the VICINITY Gateway API supports only REST API communication over the HTTP.

### 3.5. Connect Value-Added Service and real devices into VICINITY

The applicant shall:

- **connect the Value-Added Service to VICINITY** through VICINITY Adapter;
- **connect real devices to IoT infrastructure (if applicable)**. If local IoT gateway is used it shall run on standard HW platforms able running Java 8 virtual machine (such as: Raspberry PI, BananaPRO, Cubieboard, PINE64+, Intel Edison, etc.). If integration with cloud service is selected, then VICINITY Gateway API shall run in JAVA 8 complaint environment. The applicant shall provide environment where VICINITY Node is running;

- **provide support and cooperation** of the connected devices, Value-Added Services, the VICINITY Adapters throughout the project.

The VICINITY currently supports the VICINITY Adapters or the VICINITY Adapters examples for Kura IoT platform, SiteWhere, IoTivity and LinkSmart and OpenHAB (work in progress).
Examples of VICINITY Adapters and VAS can be found under https://github.com/vicinityh2020. Two indicative examples are presented below:

**VICINITY Adapter for Individual Statistics VAS (MPH Pilot Site)**

The VAS is triggered by event, each time a new measurement is taken from blood pressure monitor or weight scale, or in case the panic button is pressed. It is responsible for calculating the frequency in which measurements are taken from medical devices and check on any abnormal values. Source code can be found at: https://github.com/vicinityh2020/vicinity-VAS-IndividualStatistics_with_adapter

**VICINITY Adapter for TINYMESH Devices**

Implementation of VICINITY Adapters with the corresponding endpoints to read information from TINYMESH devices. Source code can be found at: https://github.com/vicinityh2020/vicinity-adapter-tinymesh

## 3.6. Demonstrate usage of Value-Added Services or Microservices

The applicant shall **demonstrate proposed service.**

# 4. ANNEX

## 4.1. Documents and Deliverables

| Related Deliverables and Documents | Link to the Document | Description |
|---|---|---|
| D5.1 - Value-added services definition requirements and architectural design | https://vicinity2020.eu/vicinity/content/d51-value-added-services-definition-requirements-and-architectural-design | Goal of this deliverable (D5.1 - Value-Added Services definition, requirements and architectural design) is to express the definition of "Value Added Services" in the project, their role and purpose as well as their requirements, taking into account VICINITY Platform requirements, specification and architecture, as well as identified barriers (WP1). This deliverable also describes how VICINITY enables the creation of services that target the IoT domain and cover a wide range of markets. |
| D5.2 - Value-added services implementation framework | N/A | This deliverable (D5.2 – Value-Added Services implementation framework) is aiming at making a first version of VICINITY Value-Added Services available. |

| | | |
|---|---|---|
| D4.1 - Set of open sample VICINITY gateway adapters | https://vicinity2020.eu/vicinity/content/d41-set-open-sample-vicinity-gateway-adapters | VICINITY adapters act as a gateway between the outer world and the VICINITY infrastructure, thus providing its main objective: Interoperability. An adapter consists of partners-specific infrastructure implementation on one side and a standard VICINITY interface on the other side.<br><br>The project has implemented a vast set of different adapters to many other platforms. In the deliverable, we give an overview the available adapters that are provided by the partners of the VICINITY consortium. |
| D9.3 - Data Management Plan, second version | https://vicinity2020.eu/vicinity/content/d93-data-management-plan-second-version | This is the second version of the project Data Management Plan (DMP). It contains preliminary information about the data the project will generate, whether and how it will be exploited or made accessible for verification and re-use, and how it will be curated and preserved. The purpose of the Data Management Plan (is to provide an analysis of the main elements of the data management policy that will be used by the consortium with regard to all the datasets that will be generated by the project. The DMP is not a fixed document, but will evolve during the lifespan of the project |

| | | |
|---|---|---|
| D7.1 - Pilot area installation methodology and planning | N/A | In this deliverable, a common methodology and plan for each demonstration pilot site is set out. The common methodology for each pilot describes: the key benefit per use case, the infrastructure, the stakeholders, the installation requirements and the hardware and software that will be used. The methodology includes three main phases; Pre Installation, Installation and Post Installation which is described and detailed for each pilot site. The specific plans show the conditions at the pilot sites and provides details of the installation of the infrastructure needed for its use case. The Value Added Services, for the pilot sites, are defined in D.5.1 and are implemented in software according to the requirements of the pilot site. |

## 4.2. Links

| Name | Description |
|---|---|
| VICINITY Website | https://vicinity2020.eu/vicinity/ |
| Catalogue - Platform | https://vicinity2020.eu/vicinity/content/platform |
| Catalogue - Pilot | https://vicinity2020.eu/vicinity/content/pilot |
| Catalogue - Adapters | https://vicinity2020.eu/vicinity/content/adapters |
| Community | https://vicinity2020.eu/vicinity/content/vicinity-community |
| Get Started Documentation | https://vicinity-get-started.readthedocs.io/en/latest/# |
| VICINITY Github | https://vicinityh2020.github.io/ |
| Github Documentation | https://github.com/vicinityh2020 |
| VICINITY Ontology | http://vicinity.iot.linkeddata.es/vicinity/ |
| VICINITY Adapter API | https://app.swaggerhub.com/apis/intersoft.sk/vicinity-adapter/1.0.0 |