sensiNact

Website

Consortium

Objectives

Marketplace

Lab

Membership

Commercial Platform

Criteria for open platform in AHA and AAL domains which sensiNact platform complies with:

Open Source

Open Standards Based

Federatable

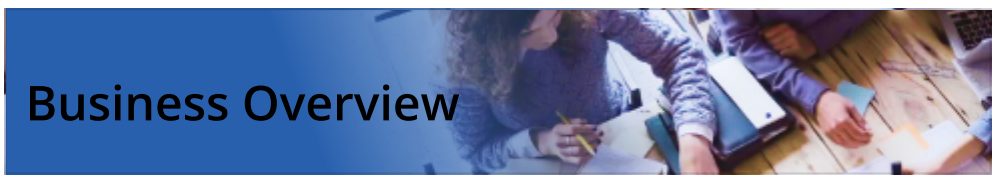
Shared Common Information Models

Vendor and Technology Neutral

Supports Open Data

Provides Open APIs

sensiNact: Open platform for smarter cities - appli...



Business Overview

The project lead of Sensinact is CEA-Leti: Laboratoire d'électronique des technologies de l'information, a French institute creating innovation with the goal to pass these on to the industry aiming to improve the quality of life worldwide. Parties which have expressed interest in collaborating with the project are Osaka University, Keio University, Engineering Ingegneria Informatica spa, ACUTUS Software, Inc., JRISS, Kyoto Sangyoo University and STMicroelectronics. Eclipse has been created to bring the community of developers and other smart city stakeholders together with the aim to tackle challenges such as, among others, economic crisis, security threats, natural disasters and ageing society with an open approach seeking to foster a smart city business ecosystem.

The core value proposition of the Sensinact project lies in the ambition to enable the collection, processing and redistribution of any data relevant to improving the quality of life of urban citizens and the programming of interfaces allowing different modes of access to this data (on-demand, periodic, historic, etc.). Finally it seeks to facilitate the application development and deployment to easily and rapidly build innovative applications on top of the platform.

Open Usage (adoptability)

Open Adaptation

Based on comprehensive communication strategy CEA has reached out to end users, developers and project partners transferring technology to the local industry through the various offices established all over France. To the platform ecosystem belong also numerous industrial and academic partners.

At the heart of Sensinact is its service-oriented approach in which IoT devices expose their functionalities in terms of services (temperature service, presence detection service, air quality monitoring service, alarm service, etc.). Each service then exposes one or several resources such as sensor data or actions. Building applications thus become a matter of composing sensing services with actuation services. Four elements build the core service offering of the platform.

With the lack of a de facto standard data model today in the IoT domain, Sensinact adopts a generic and extensible data model to facilitate building adapters for various protocols. Its core model is based on four types of resources: sensor data, action, state variables, and properties. Those resources are accessible by generic and easy to use Application Programming Interfaces (API) providing synchronous (on demand) and asynchronous (periodic or event based) access to data and actions of IoT devices, as well as access to historic data.

Learn more about SensiNact

In the Eclipse SensiNact *Physical layer*, emerging IoT devices, legacy systems, increasing number of social networks, mobile applications, open data repositories and web data are the potential exploitable data sources. SensiNact ships with southbound bridges for using a lot of common devices including Zigbee (motion sensors, force sensor, etc.), EnOcean (remote controls, windows opener detectors, etc.), CoAP (sliders, buttons, etc.). It also provides a bridge for retrieving context information using NGSi 9/10 protocol. Thanks to an OSGi based architecture, it is possible to add bridges on the fly, while the gateway is running, to allow communication with new kind of devices. The project uses third party software with various license types such as Apache v2.0 and EPL, JSON and MIT. sensiNact thus provides connectivity support to those data sources including today's IoT protocols and platforms such as LoRa, Zigbee, IEEE

802.15.4, Sigfox, enOcean, MQTT, XMPP, NGSI, HTTP, CoAP, etc. With its modular approach, connectivity support for new protocols can be rapidly developed and dynamically added to the platform, even at run-time. Those resources are accessible by generic and easy to use Application Programming Interfaces (API) providing synchronous (on demand) and asynchronous (periodic or event based) access to data/actions of IoT devices, as well as access to historic data.

Considering the *Service layer* and the *Application layer*, the Eclipse sensiNact platform aims at creating a common environment in which heterogeneous devices can exchange information and interact among each other in the IoT world. This environment is composed of two tools: the sensiNact Gateway, which refers to the *Service Layer* and aims at integrating devices and aggregating data from various sources and sensiNact Studio, which refers to the Application Layer, aiming at interacting with the sensiNact Gateway to visualize the devices and the data.

SensiNact Gateway interconnects IoT devices using different southbound IoT protocols such as Zigbee, EnOcean, LoRa, XBee, MQTT, XMPP, as well as platforms such as FIWARE and allows access to them with various northbound protocols such as HTTP REST, MQTT, XMPP, JSON RPC and CDMI. The gateway can also host applications and manage them using an application manager module.

The sensiNact Studio proposes an IDE (Integrated Development Environment) based on Eclipse to manage the existing devices, in addition to develop, deploy and manage IoT applications. It is a service composition tool which assists developers in building applications by binding the services via events and actions. With a Domain Specific Language, the developers can express the application logic in terms of ECA (Event-Condition-Action) rules, which is verified and validated by the tool before its deployment into the sensiNact platform. The application developers can then remotely monitor and manage applications (install, start, stop, uninstall, etc.). The tool also provides means to easily build support for new types of protocols and platforms and add it to the platform on-the-fly.

SensiNact adopts a generic and extensible data model to facilitate building adapters for various protocols. It is core model is based on 4 types of resources: sensor data, action, state variables, and properties. Those resources are accessible by generic and easy to use Application Programming Interfaces (API) providing synchronous (on demand) and asynchronous (periodic or event based) access to data/actions of IoT devices, as well as access to historic data.

A first level of security of the Sensinact platform is reached by the available security tools in the OSGi environment: ServicePermission and ConditionalPermissionAdmin. The data collected from the sensors connected to Sensinact

gateway use Application Programming Interfaces (API) providing synchronous (on demand) and asynchronous (periodic or event based) access to data/actions of IoT devices, as well as access to historic data. This is real-time information provided. This data can be visualized with sensiNact Studio which allows managing devices/services connected to the platform and rapidly creating applications and deploying them to the platform.

Semantic interoperability support is not present in sensiNact, but it is supposed to be implemented in future updates. SensiNact has been successfully used in close to real-life environments in various application domains such as smart city, smart home, smart shopping, smart health care within the context of collaborative projects.

Learn more about SensiNact

Contextual Overview

Sensinact is one of the open source projects established under the Eclipse Foundation. The Foundation is home to the Eclipse IDE, Jakarta EE, and over 350 open source projects, including runtimes, tools, and frameworks for a wide range of technology domains, such as the Internet of Things, automotive, geospatial, systems engineering, and many others.

The Eclipse Foundation was created in January 2004 as an independent not-for-profit organization supported by over 275 members who value the Foundation's governance model, open innovation processes, and community-building events. Members include industry leaders who have embraced open source as a key enabler for business strategy.

The Eclipse Project was originally created by IBM in November 2001 and supported by a consortium of software vendors. The Eclipse Project has been used by millions of developers.

The Eclipse community consists of individual developers and organizations spanning many industries. The Foundation employs a full-time professional staff to provide services to the community. The Eclipse Foundation is funded by annual dues from members and governed by a Board of Directors. Strategic Developers and Strategic Consumers hold seats on this Board, as do representatives elected by Add-in Providers and Open Source committers. Eclipse committers

are typically employed by organizations or are independent developers that volunteer their time to work on the Eclipse projects.

The Eclipse Foundation provides four key services to the Eclipse community: 1) IP Management, 2) Ecosystem Development, 3) Development Process, and 4) IT Infrastructure.

An important aspect of the Eclipse Foundation is the focus on enabling the use of open source technology in commercial software products and services. This is made possible by the fact that all Eclipse projects are licensed under the Eclipse Public License (EPL), a commercial friendly OSI approved license.

The Eclipse Foundation also undertakes a number of steps to attempt to ensure the pedigree of the intellectual property contained within Eclipse projects. The first step in the due diligence process is trying to ensure that all contributions are made by the rightful copyright holder and under the Eclipse Public License (EPL). All committers are required to sign a committer agreement that stipulates all of their contributions are their original work and are being contributed under the EPL. If a committer is sponsored to work on an Eclipse project by a Member organization, then that organization is asked to sign a Member Committer Agreement to ensure the intellectual property rights of the organization are contributed under the EPL.

The second step is that the source code related to all contributions which are developed outside of the Eclipse development process is processed through the Eclipse Foundation IP approval process. This process includes analysing selected code contributions to try to ascertain the provenance of the code, and license compatibility with the EPL. Contributions that contain code licensed under licenses not compatible with the EPL are intended to be screened out through this approval process and thus not added to an Eclipse project. The end result is a level of confidence that Eclipse open source projects release technology that can be safely distributed in commercial products.

Learn more about SensiNact