# The M3 Architecture for Smart Spaces

## Overview of Semantic Information Broker Implementations

F. Viola[1] A. D'Elia[1] D. Korzun[2]
I. Galov[2] A. Kashevnik[3],[4] S. Balandin[5]

[1]University of Bologna, Bologna, Italy

[2]Petrozavodsk State University (PetrSU), Petrozavodsk, Russia

[3]SPIIRAS, St. Petersburg, Russia

4 ITMO University, St. Petersburg, Russia

5 FRUCT Oy, Helsinki, Finland

19th FRUCT, 7-11 November 2016

# Outline

# Outline

# Smart-M3

Smart-M3 is an open source **interoperability platform** implementing the M3 paradigm (**multi-device**, **multi-vendor**, **multi-domain**).

Originally framed in:

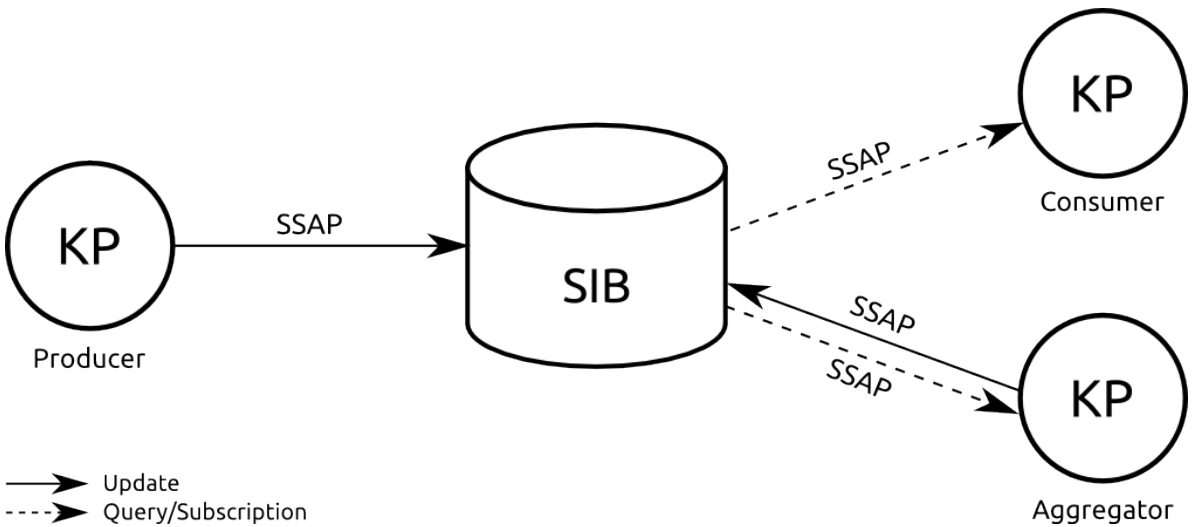
- **ARTEMIS** JU European funded SOFIA (Smart Objects For Intelligent Applications) project;
- Finnish nationally funded **DIEM** (Device Interoperability Ecosystem) program.

Formerly developed by Nokia, it is now maintained by **FRUCT Association**, **SOFIA Community** and the **ARCES department** of the University of Bologna.

# Smart-M3

The M3 architecture relies on three main components:

- **SIB** (*Semantic Information Broker*)
- **KPs** (*Knowledge Processors*)
- **SSAP** (*Smart Space Access Protocol*)



# Smart-M3

The SIB is a SPARQL endpoint built on top of an RDF triple-store. This publish-subscribe broker allows to:

- update the knowledge base (using SPARQL, RDF/XML or a triple-pattern based formalism, i.e. RDF-M3)
- query the knowledge base (through SPARQL or RDF-M3)
- subscribe to user-defined subgraphs (through SPARQL or RDF-M3)

# Outline

# The Piglet-based SIB

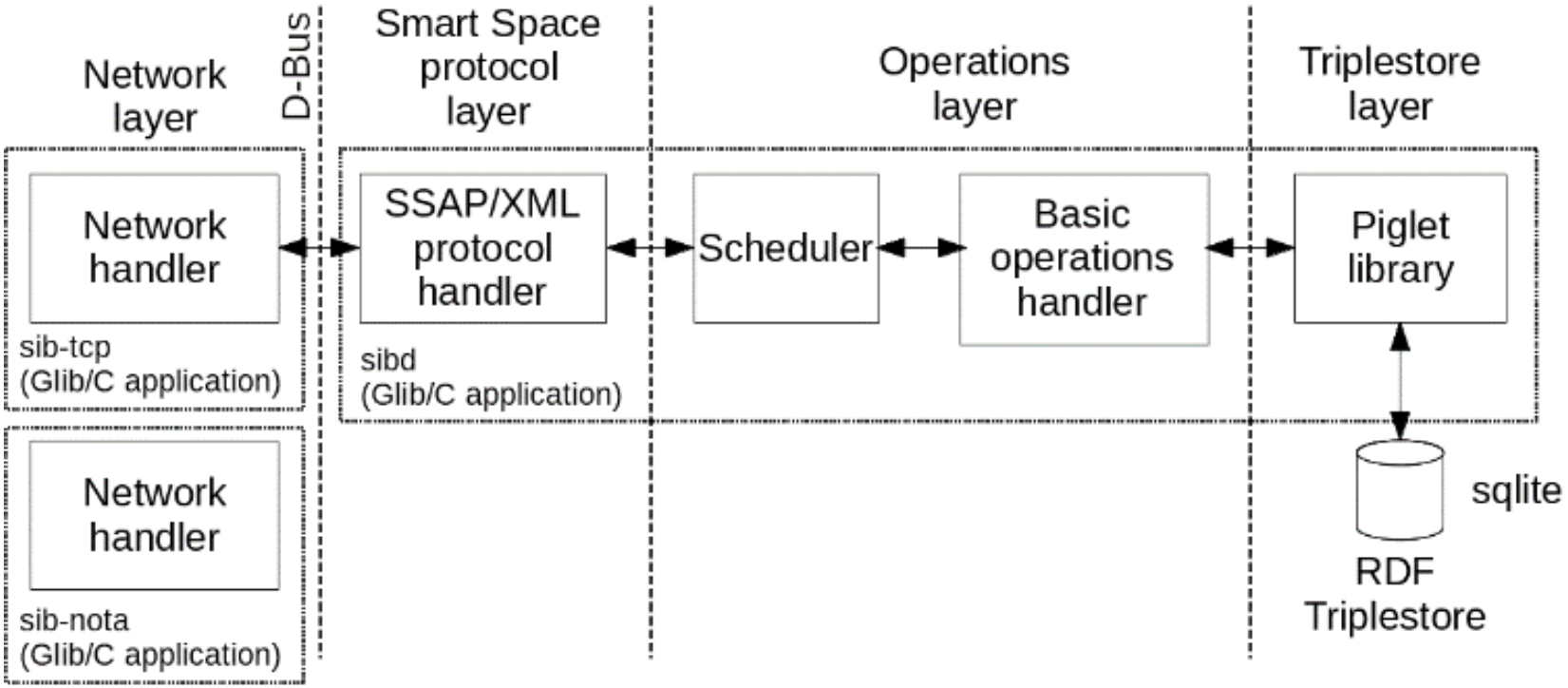
**Where** Developed by Nokia Research Center

**When** Project started in 2009

**Why** To formalize the concept of space-based information sharing

**What** It is the first official implementation of a Smart-M3 Semantic Information Broker

# Piglet-based SIB Architecture



# RedSIB

**Where** Developed by the University of Bologna

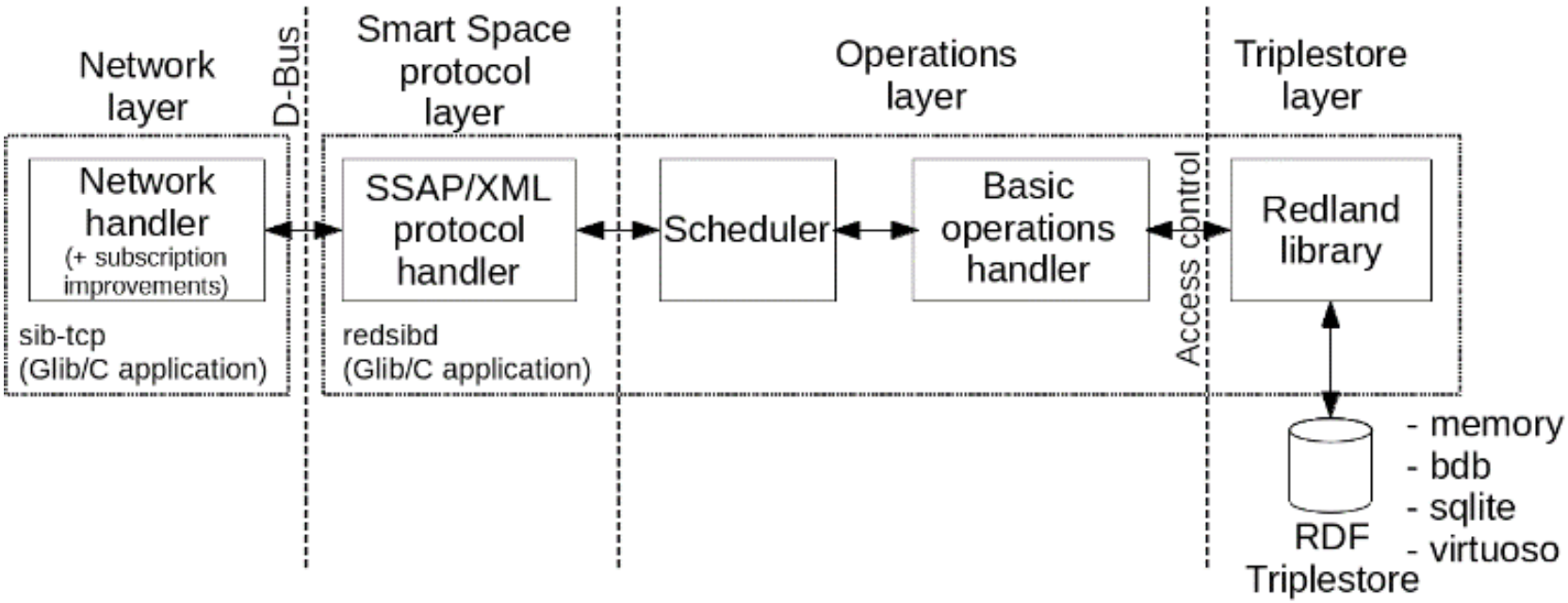
**When** Project started in 2012

**Why** It is an evolution of the old Piglet-based SIB born to avoid some criticalities and improve performance

**What** It provides:

- support for Virtuoso triple store and for volatile storage based on BDB
- prototype of access control mechanism
- optimized subscription engine

# RedSIB Architecture



# The OSGi SIB

**Where** Developed by the University of Bologna and Eurotech

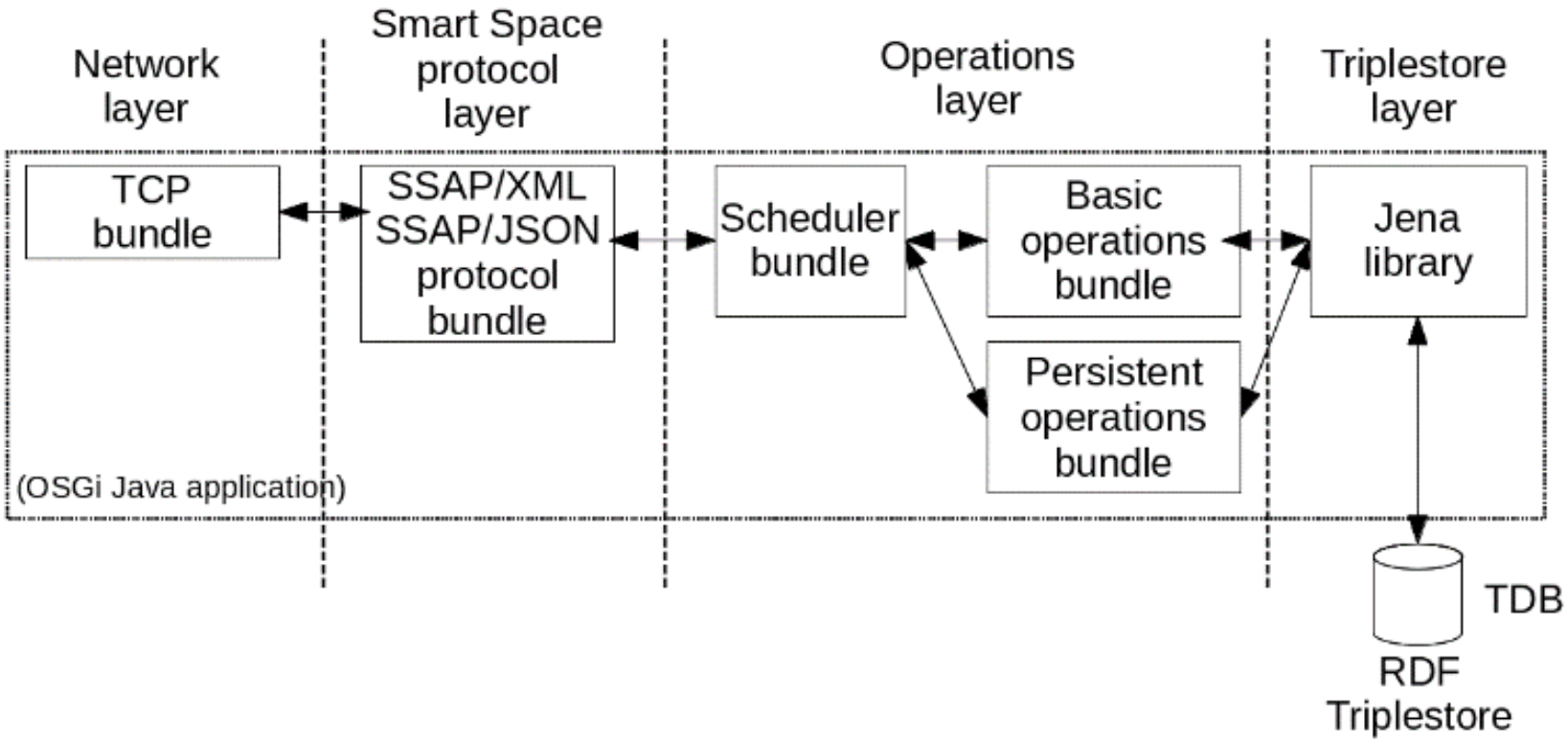
**When** Project started in 2010

**Why** To provide a portable SIB for industrial domains

**What** A modular architecture based on OSGi bundles that:

- provides reliable query engine
- introduces the new Persistent Update (PU) primitive
- opened the way to the development of an Android SIB

# The OSGi SIB Architecture



# pySIB

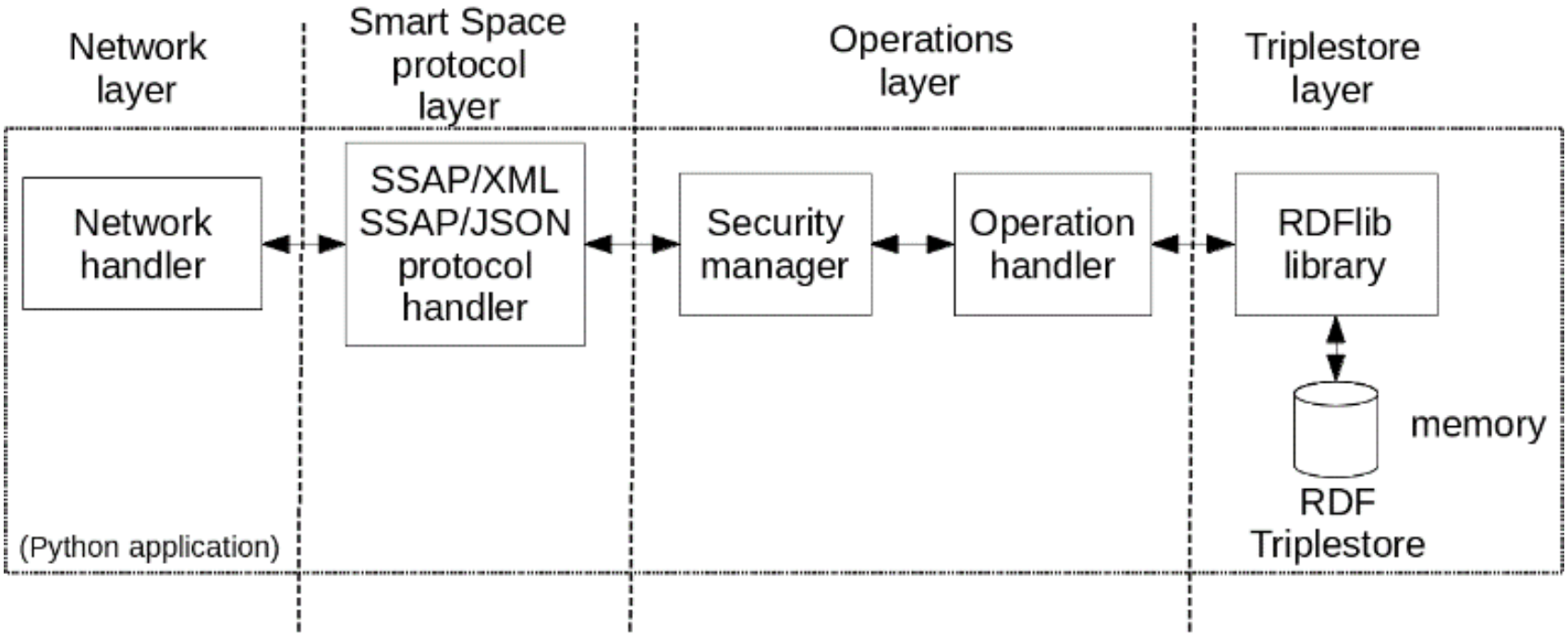
**Where** Developed by the University of Bologna

**When** Project started in late 2015

**Why** To provide a portable and lightweight SIB, with a modular and easy structure suitable for didactics

**What** Lightweight Python implementation supporting an experimental JSON-encoded SSAP

# pySIB Architecture



# CuteSIB

**Where** Developed by the Petrozavodsk State University (PetrSU)

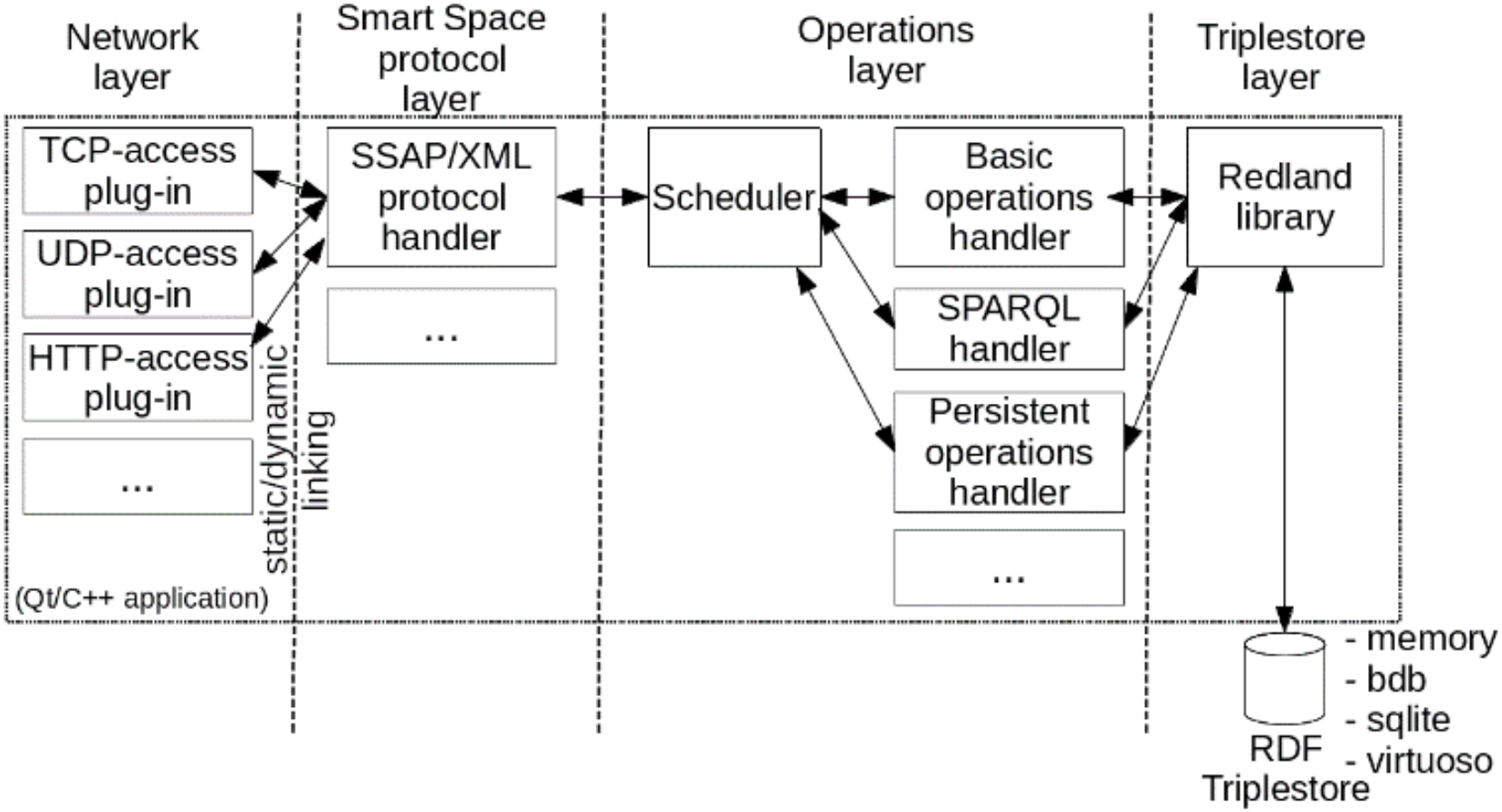
**When** Project started in 2015

**Why** To provide a renewed C implementation of the SIB

**What** A fork of RedSIB that:

- is based on QT framework
- replaces the D-BUS
- has a modular structure
- is portable and extensible

# CuteSIB Architecture



# RIBS

**Where** Developed at the VTT Technical Research Center of Finland

**When** During the SOFIA project (2008-2012)

**Why** To provide a secure architecture oriented at low-resources devices

**What** RIBS was a SIB addressing two of the main problems of IoT: providing security and supporting low-resources nodes. Due to a not open source license it failed to build a community of users and developers, so the project was suspended.

# The ADK SIB

**Where** Developed by Indra & Tecnalia

**When** During the SOFIA Project (2008-2012)

**Why** It was designed to have a powerful suite for ontology based code generation and model based application development

**What** ADK was developed using the OSGi framework and provided a SIB integrated in the Eclipse Development Environment.

# Outline

# Evaluation

The following performance tests have been performed on the available SIBs:

# Evaluation

The following performance tests have been performed on the available SIBs:

- The update mechanism
  - i.e. the time needed to put data into the SIB

# Evaluation

The following performance tests have been performed on the available SIBs:

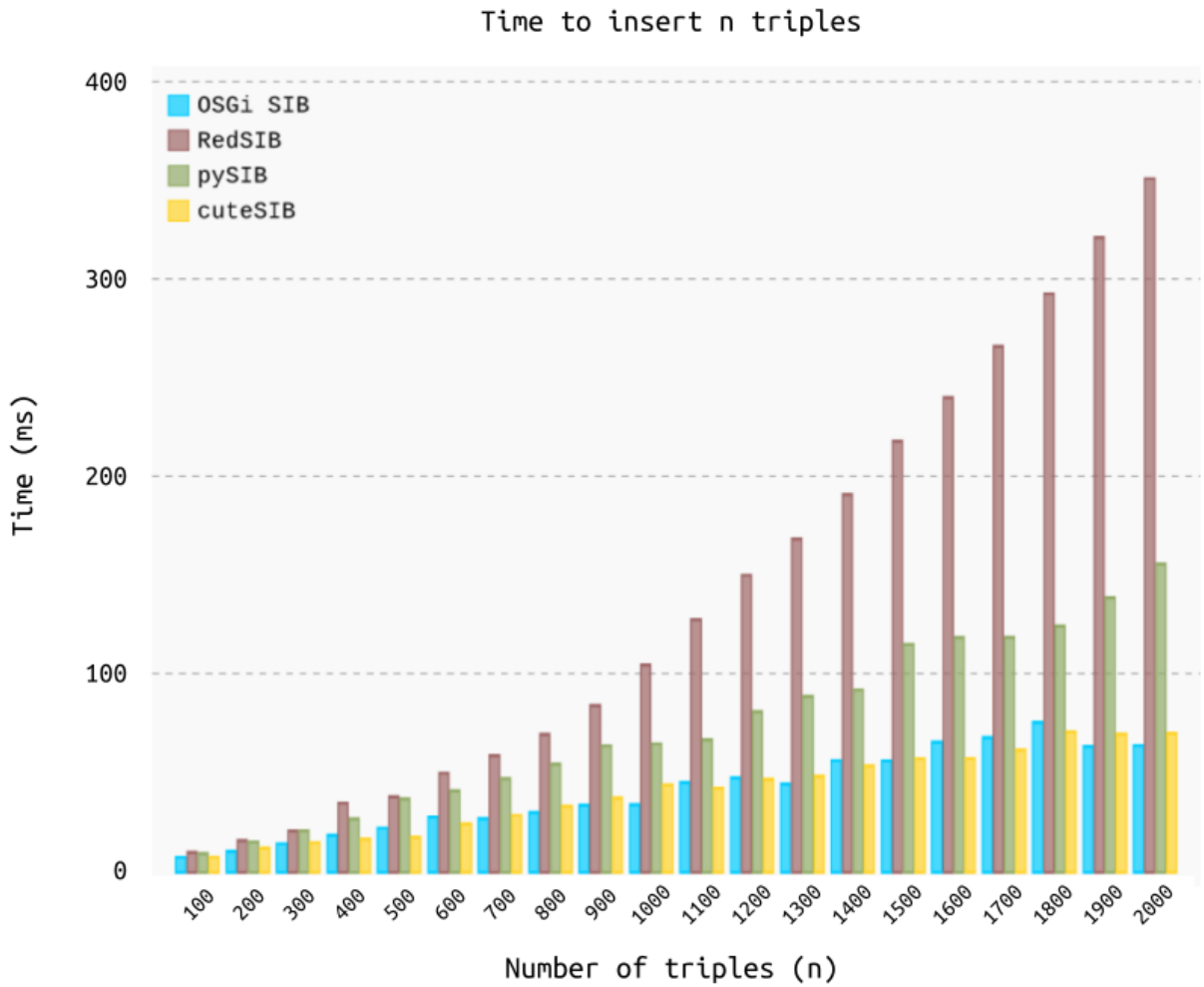
- The update mechanism
  - i.e. the time needed to put data into the SIB
- The query mechanism
  - i.e. the time needed to retrieve data from a SIB

# Evaluation

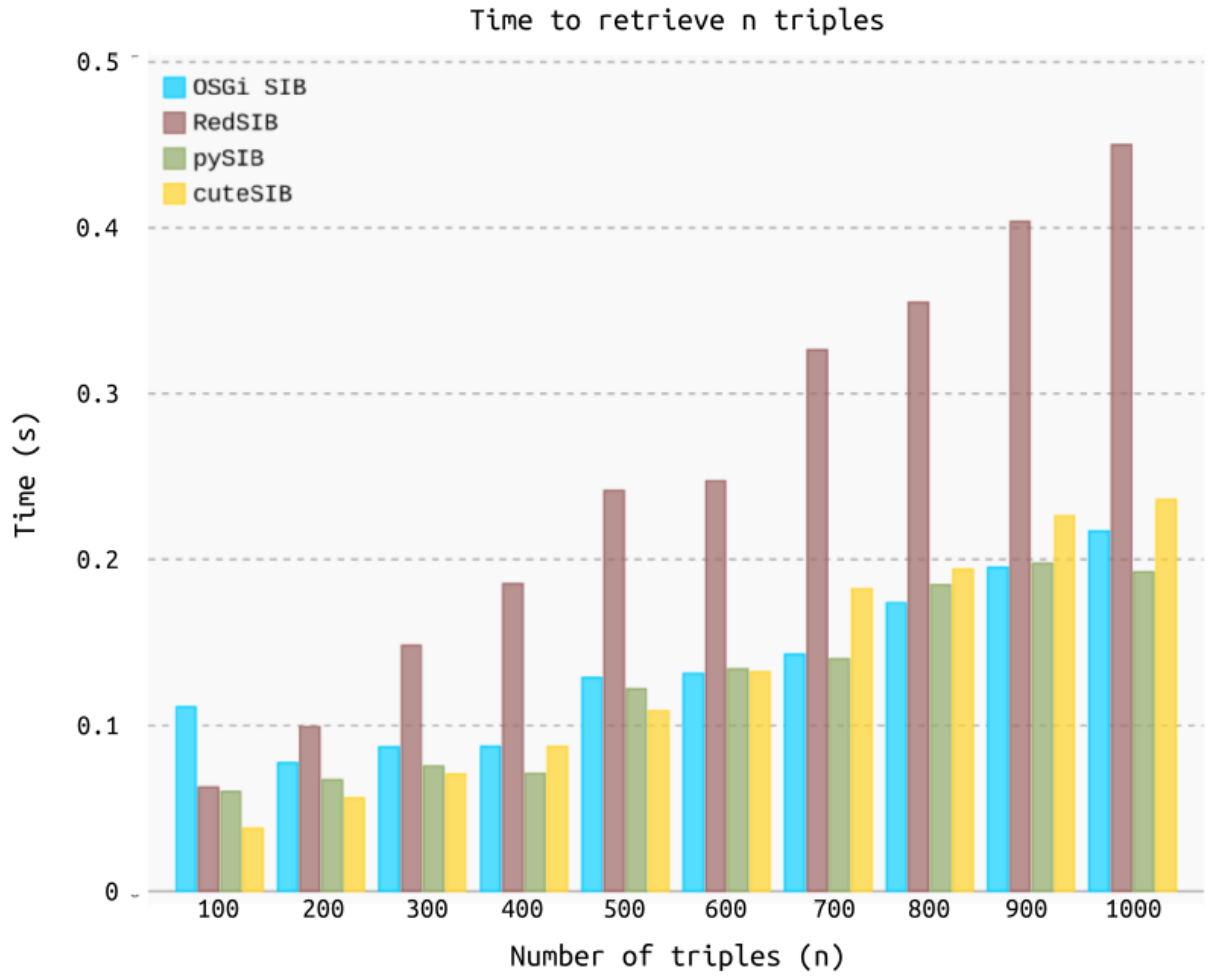
The following performance tests have been performed on the available SIBs:

- The update mechanism
  - i.e. the time needed to put data into the SIB
- The query mechanism
  - i.e. the time needed to retrieve data from a SIB
- The subscription mechanism
  - i.e. the time needed to receive a notification

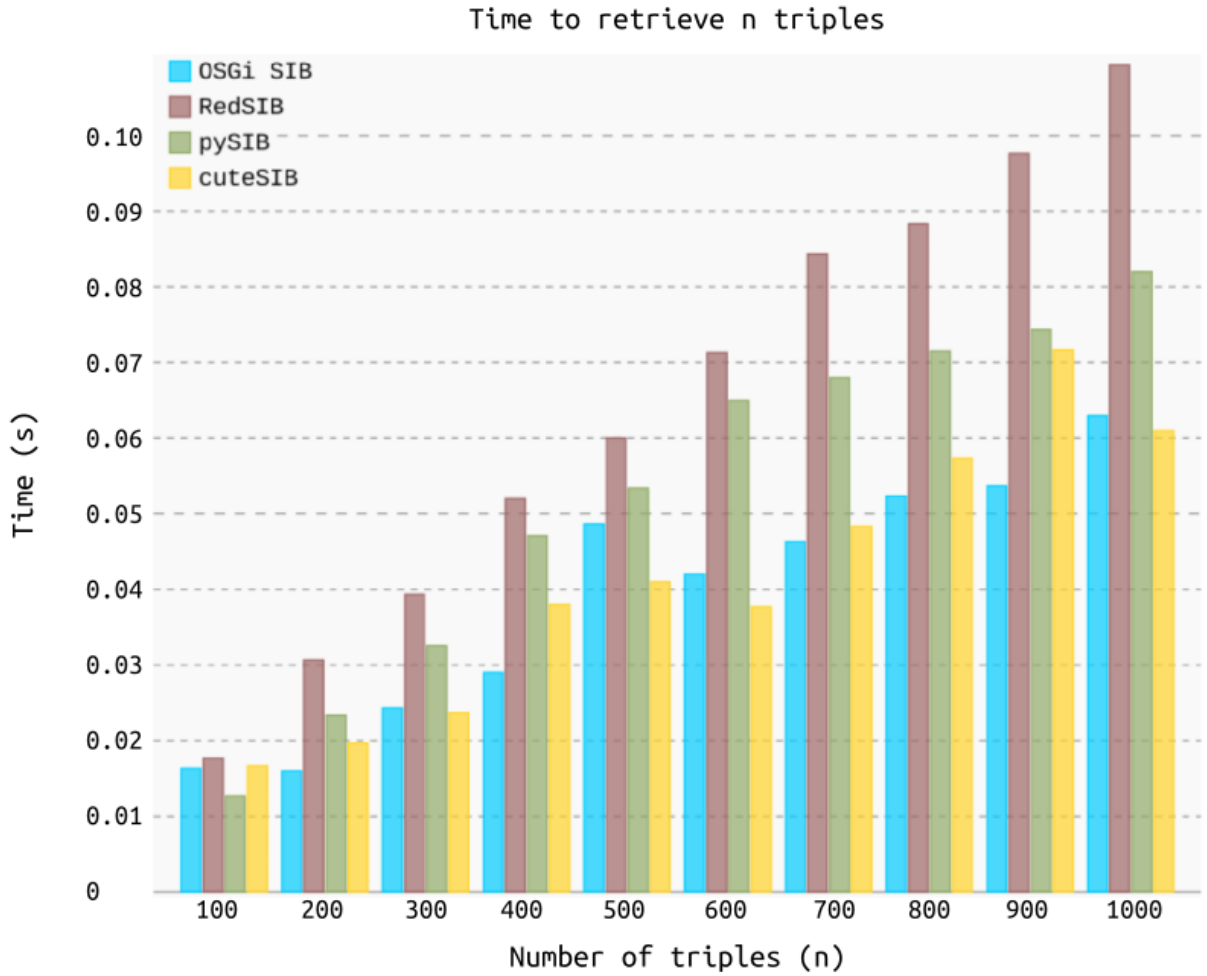
# Evaluation of the Update mechanism



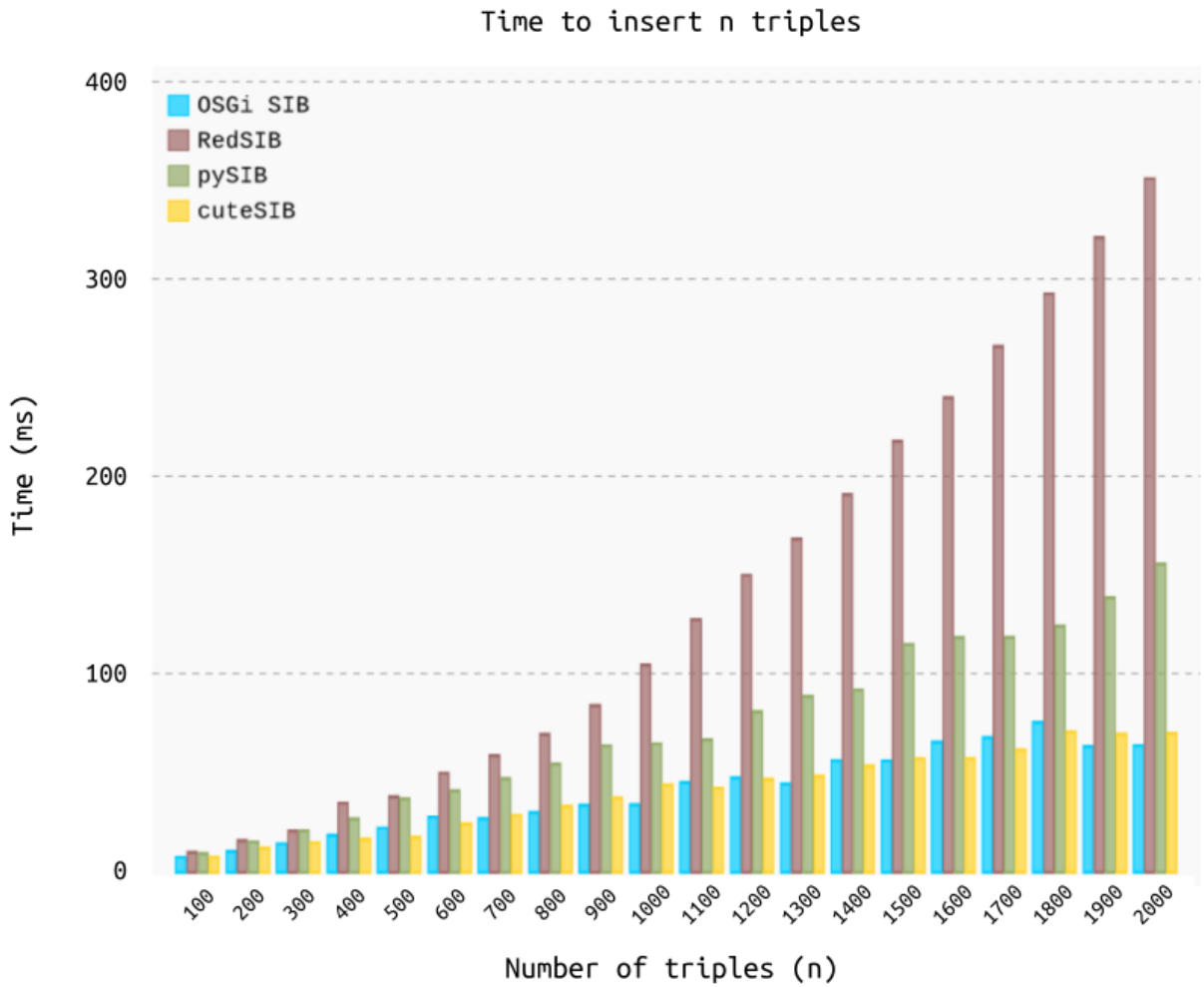
# Evaluation of the Query engine (with RDF-M3)



# Evaluation of the Query engine (with SPARQL)



# Evaluation of the Subscription engine



# Outline

# Summarizing...

| SIB | Developer | Language | Active |
|---|---|---|---|
| **ADK SIB** | | Java | No |
| **CuteSIB** | PetrSU | C | Yes |
| **OSGi SIB** | ARCES | Java | Yes |
| **Piglet-based SIB** | NOKIA | C | No |
| **pySIB** | ARCES | Python | Yes |
| **RedSIB** | ARCES | C | Yes |
| **RIBS** | VTT | C | No |

# Outline

# Conclusion

Smart-M3 providesa promising technology for smart spaces. We reviewed the Smart-M3 platform by focusing on the main SIB implementations.

The evaluation of the currently available SIBs highlights good peformance of CuteSIB and pySIB. On the other hand a weak point is represented by the SPARQL engine that looks less reliable than the one provided by the OSGi SIB.