



Juha-Pekka Soininen | Sergey Balandin
Johan Lilius | Petri Liuha
Tullio Salmon Cinotti (Editors)

Proceedings of the Open International M3 Semantic Interoperability Workshop

Jointly organized by Open Local Data Application
activity at EIT ICT Labs Smart Spaces Action Line and
Open Innovations Association FRUCT

EIT ICT Labs Open Innovation House,
Espoo, 12th November, 2013

TURKU CENTRE *for* COMPUTER SCIENCE

TUUCS Lecture Notes
No 21, December 2013



Proceedings of the Open International M3 Semantic Interoperability Workshop

Jointly organized by Open Local Data Application activity at EIT (European Institute of Innovation and Technology) smart spaces thematic action line and Open Innovations Association FRUCT

EIT Open Innovation House, Espoo, 12th November, 2013

Editors:

Juha-Pekka Soininen

Sergey Balandin

Johan Lilius

Petri Liuha

Tullio Salmon Cinotti

TUCS Lecture Notes No. 21

December 2013

ISBN 978-952-12-2990-9

ISSN1797-8831

Preface

Semantic interoperability means that separate devices and systems publish their information for others using semantic format. It means that the actual data is linked and presented together with some specific element of the ontology model describing semantics of the data. When separate systems understand and have common ontology models, i.e. speak same language, they can understand correctly the meaning of the shared data. And this is naturally the basic requirement for interoperability, i.e. the creation of joint functionality. Semantic interoperability is nowadays widely acknowledged approach for creating complex and heterogeneous systems.

M3 concept was originally developed at Nokia Research Center. The concept extends the semantic RDF-graph (Resource Description Framework) storages with possibility to subscribe to changing data and with interfaces to various SOA (service Oriented Architecture) solutions. Especially the capability to create notifications based on subscriptions allows creating systems that are based mostly on shared information. The M3 concept and Smart-M3 open source solution were further developed in Artemis Sofia and Finnish national DIEM projects starting in 2009, but after these projects ended, the development has been more scattered and coordination has been missing.

The purpose of the first open international M3 semantic interoperability workshop was to give a possibility for students, researchers and organisations interested in semantic interoperability solutions based on Smart-M3 or M3 concept to meet, to present their results, and to discuss on how we should continue the development of the concept. The workshop focused on sharing the status of M3 related activities, on creation of a common understanding what has been done so far, and on discussing about M3 community.

The first Open International M3 Semantic Interoperability Workshops was organised as a joint effort of Open Local Data Applications activity at EIT Smart Spaces Thematic Action Line and Open Innovations Associations FRUCT. The workshop Call for Participation was advertised through known contacts by the programming committee and through EIT Smart Spaces Action Line and FRUCT mailing lists. The workshop contained 23 presentations from 13 organisations covering the areas of developments related to M3 concept itself, to M3 based infrastructures, to use cases based on M3, and to the ideas related to M3 community. All the presentations are included in this proceeding.

The participation well exceeded our expectations. We had more than 50 participants in the event. Also the M3 community that was thought to be scattered and small turned out to be much more coherent and active as can be seen from presentations. In many ways this workshop can be seen as a new launch of open M3 community and in the discussion section it was also decided that continuation of collaboration is needed.

December, 2013

Juha-Pekka Soininen, VTT Technical Research Centre of Finland
Johan Lilius, Åbo Akademi
Tullio Salmon Cinotti, University of Bologna
Sergey Balandin, FRUCT Oy
Petri Liuha, EIT ICT Labs

Contents

M3 Community	1
Smart Spaces Development: Open Innovations Association FRUCT Activities	3
M3 in Education and Research: Experiences and Lessons Learned	13
Smart M3 in Embedded Devices Communities.....	23
An OCPP Interoperable Services System Based on a SmartM3 Core.....	27
Resolution Infrastructure for M3 Based Systems	37
M3 Technology.....	47
A Scalable Distributed M3 Platform on a Low-Power Cluster.....	49
KSP: SPARQL-Like Knowledge Sharing for Resource Restricted Devices.....	59
A Portable Implementation of Semantic Information Broker in OSGi Technology.....	71
SMOOL – The Smart Space Modeling, Access and Deployment Framework.....	79
The Semantic Event Broker.....	87
Performance Evaluation of Operations in RedSIB with Substitution Mechanism	95
M3 Infrastructures	103
RoDaFlow: A Framework for Development of Dataflow Networks Agents in Smart-M3 with Substitution Method.....	105
Context-Based Access Control Model for Ridesharing Service	111
Reusable, Semantic, and Context-Aware Micro-Architecture	121
The SmartRoom Infrastructure: Service Runtime Reliability	127
M3 Based Knowledge Sharing Protocol Applied to Bluetooth LE Sensors and Actuators	133
Programming Android Client for Smart-M3 Applications.....	145
M3 Case Examples	151
M3 Interoperability for Remote Rehabilitation with Kinect	153
Smart-M3 and Geo2Tag Platforms Integration Agent	165
Recommendation System for Tourist Attraction Information Service.....	179
Development of Smart Room Services on Top of Smart-M3	189
Delivery of SmartRoom Services Using Mobile Clients	197
User Presence Detection in SmartRoom using Innorange Footfall Sensor	203
Discussion about the future of M3 and M3 Community.....	209

M3 Community

Smart Spaces Development:

Open Innovations Association FRUCT Activities

Dmitry Korzun, Sergey Balandin,
Alexey Kashevnik, Kirill Krinkin,
Ilya Paramonov

Open International M3 Semantic Interoperability Workshop
12.11.2013, EIT Open Innovation House, Espoo, Finland

FRUCT WG on Internet of Things and Smart Spaces

- Since 2008
- Research organizations and SMEs from Europe and Russia, fruct.org/smart/
- ☐Are You Smart (ruSMART)☐community and annual international conference, rusmart.e-werest.org
- Collaborative network with international R&D projects in the area of Ubiquitous Computing, IoT and Smart Spaces
- Leading developer team for Smart-M3 open source platform, sourceforge.net/projects/smart-m3/

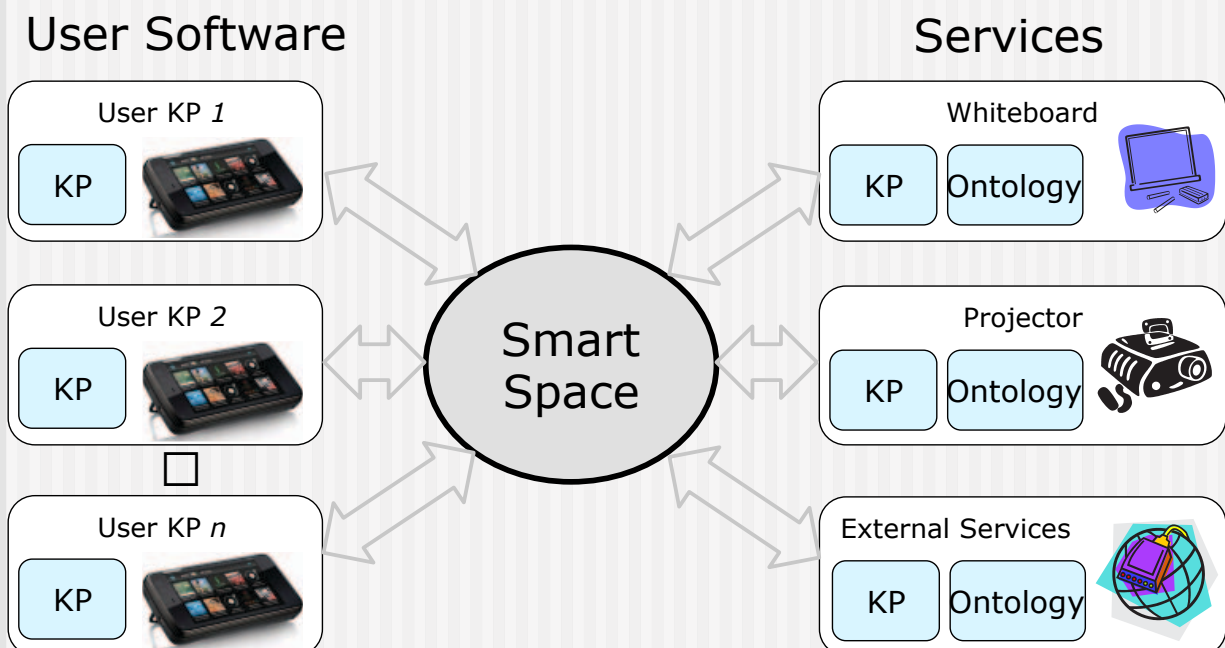
Services: Examples

- SmartConference
 - Assistance for conferencing activity
- SmartScribo
 - Mobile semantic multi-blogging
- SmartRoom
 - Service environment for collaborative activity
- Mobile Tourist Guide
 - Ridesharing: shared use of cars
 - Tourist Attraction Information Service (TAIS): information and recommendations

3

SmartConference

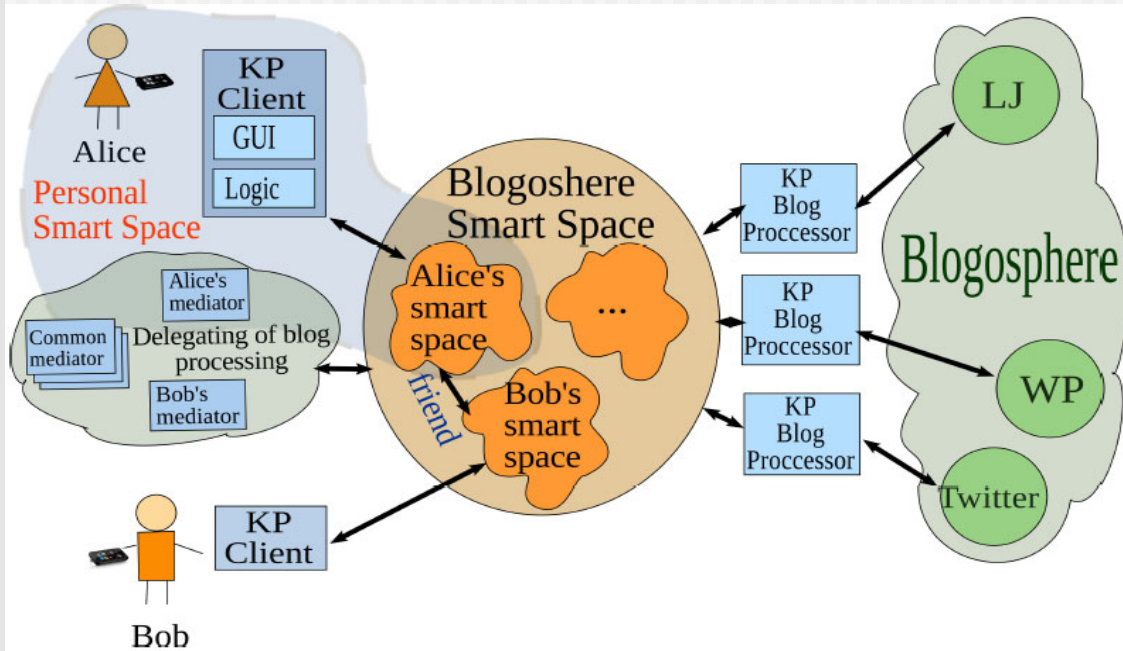
sourceforge.net/projects/smartconference



4

SmartScribo

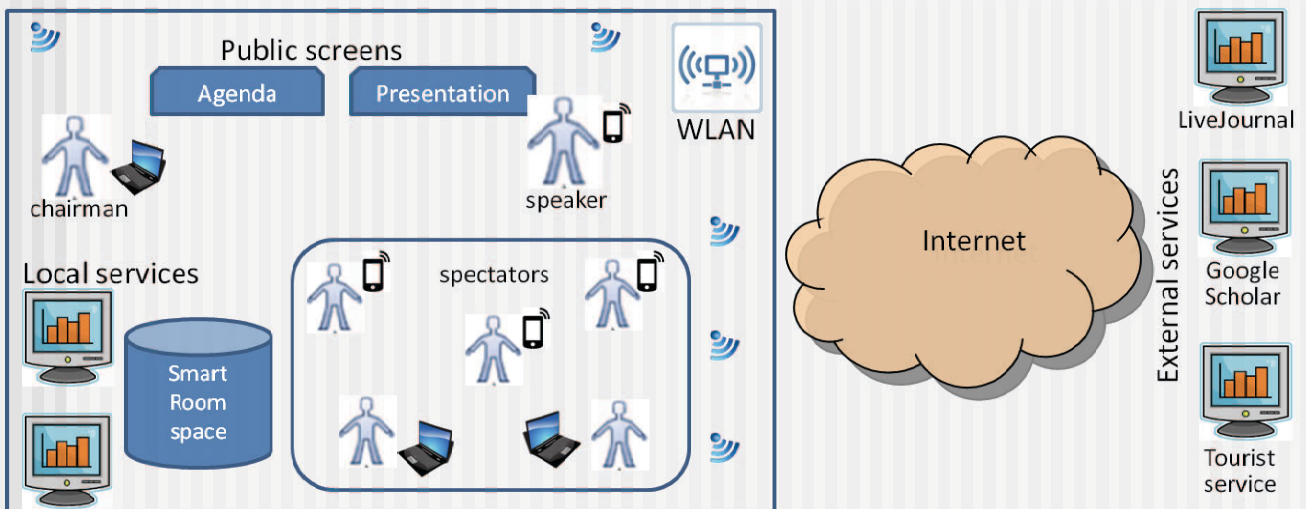
sourceforge.net/projects/smartscribo



5

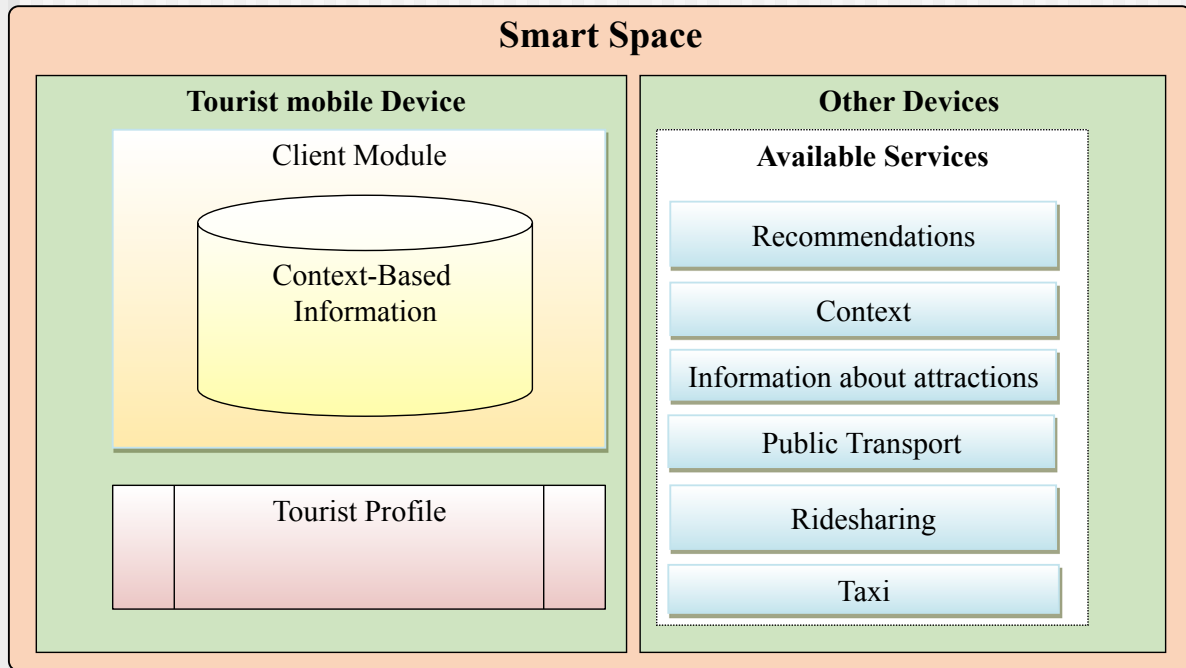
SmartRoom

sourceforge.net/projects/smartroom



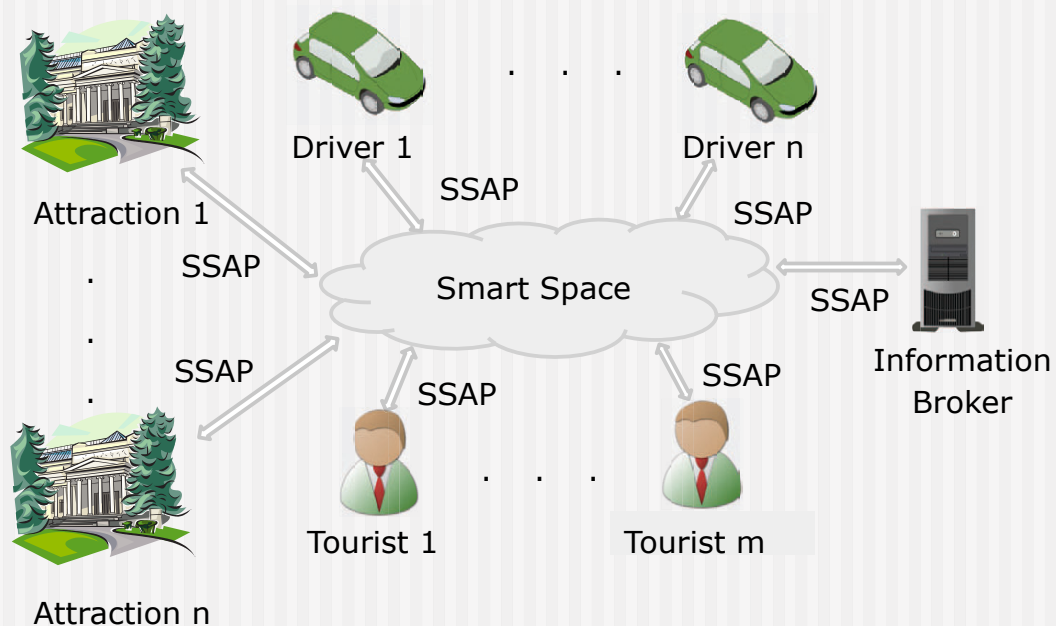
6

Mobile Tourist Guide



7

Ridesharing



8

Service Integration

- Pecha Kucha for SmartConference
 - Special client for automatic demo presentations show
- Blogging in SmartConference
 - Discussion service in conferencing
- Event Recording in SmartRoom
 - Summary report on the passed activity
- Ridesharing for Mobile Tourist Guide
 - Alternative transportation means for the tourist

9

Smart-M3 Platform

sourceforge.net/projects/smart-m3

- Maintenance and installation packages
- SDK
- Platform-embedded solutions
- Application-level solutions

10

Smart-M3 Maintenance (1/2)

Developers server: <http://smart-m3.atlassian.net>

Assigned to Me

T	Key	Summary	P ↓
+	SMART-13	deb packages for smart-m3 components	↑
+	SMART-14	Debian repository for end users	
⊞	SMART-16	Apply hip fixes from Ilya	
⊞	SMART-20	redsib hangs when running basic_test_a	
⊞	SMART-21	Upload python KP to github as own repo	
+	SMART-27	Repository with deb packages for smart-m3 platform	
⊞	SMART-28	Create and share Smart-M3 platform dev policy	
⊞	SMART-72	Redsibd couldn't get ses	
⊞	SMART-74	Smart-m3 platform is ava	
⊞	SMART-75	Add waiting loop and pre tcp	

Issue Statistics

STATISTICS: SMART-M3 (STATUS)

Open	20	26%
In Progress	3	4%
Resolved	34	45%
Closed	19	25%

Total Issues: 76

Road Map: Next 365 Days (Until 10/Nov/14)

Smart-m3 : current	15/Apr/13	9 of 9 issues resolved.
Smart-m3 : Smart-m3 Platform 0.1	31/Oct/13	19 of 27 issues resolved.
Smart-m3 : next	01/Nov/13	No issues.

Activity Stream

JIRA

Friday

Michael Bazhenov stopped progress on Smart-M3 roadmap on wiki
Friday at 11:49 AM

Thursday

Michael Bazhenov added Smart-M3 platform

Features	Date
Integrate RedSIB 0.9	05.11.201
Support for raptor2	05.11.201
Agent substitution mechanism	15.12.201
Support for Virtuoso db	15.12.201
Public package for C++/Qt KPI	15.01.201
Support for Python 2.7	15.01.201

Thursday at 9:05 PM

Michael Bazhenov started progress on Smart-M3 roadmap on wiki
Thursday at 8:39 PM

11

Smart-M3 Maintenance (2/2)

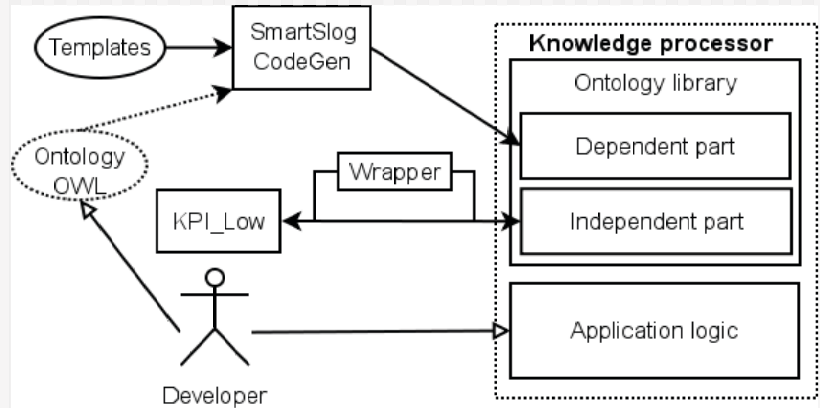
- Installation packages (deb repository)
<http://download.geo2tag.org/smart-m3-repo/>
- Short-term release plan

Feature set	Release	Date
- Red SIB 0.9 integration - Support for raptor2	0.1	05.11.2013
- Agent substitution mechanism - Virtuoso DB support - Unit test set	0.2	05.02.2014
- Smart-M3 security - Systematization of KPIs - Performance test suite	0.3	05.04.2014

12

SmartSlog: Ontology-driven SDK

- Code generation
- Low-capacity devices
- Session
- Subscription
- Multiple OWL ontologies

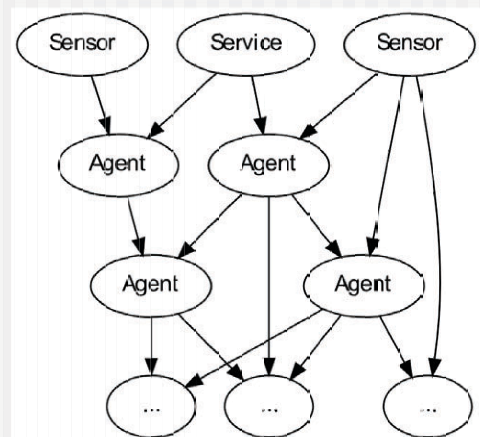


- ANSI C: Linux, Windows, Android, Raspberry Pi, Qt-based, Mac OS (C KPI)
- C#: Windows, Windows Phone (C# KPI, C KPI)
- <http://sourceforge.net/projects/smartslog/>

13

Agent substitution mechanism for dataflow network

- Uses restricted computational model: dataflow network
- When an agent falls down it is substituted by another agent
- Dependent services in the network are not disrupted
- Mechanism is implemented as SIB modification



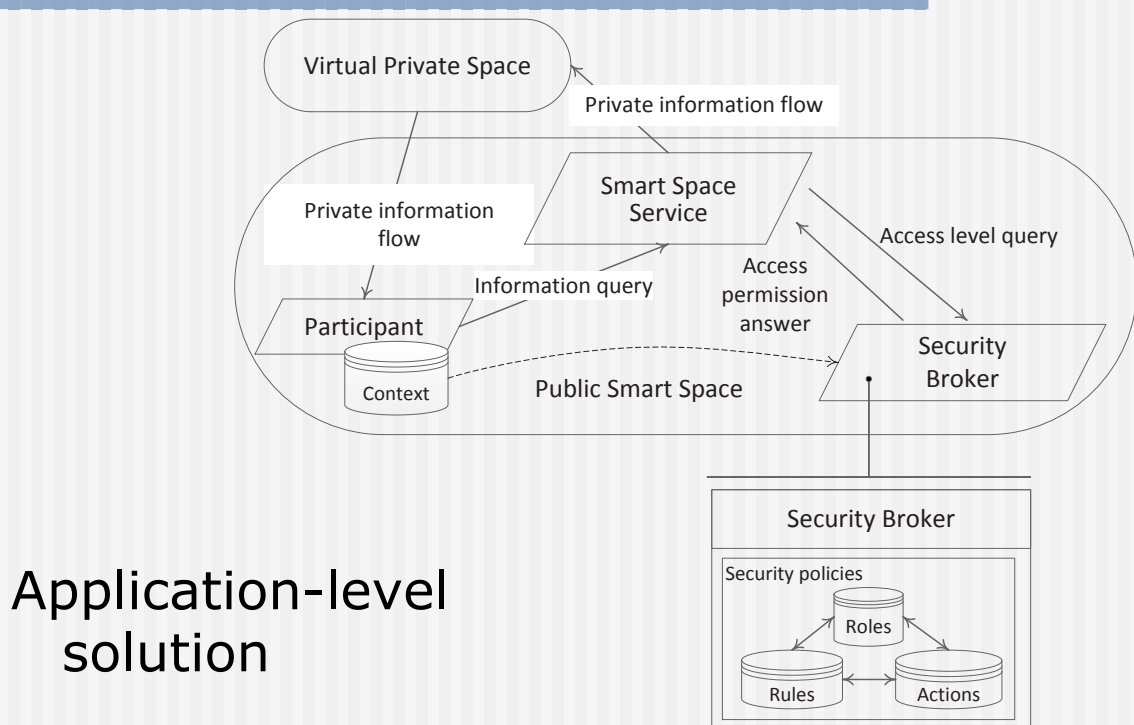
14

RoDaFlow Framework

- SDK for services based on dataflow network model
- Simplifies creation of dataflow network agents
- Developer describes only data processing procedure of the agent
- No knowledge of substitution implementation details expected from developer
- No boilerplate code for agent lifecycle needed

15

Access Control Model for Information Sharing



Application-level solution

16

Properties (1/2)

- Mobile participants
 - Personal end-user device is a primary access/control point
- Objects universe
 - devices, agents, services, compositions of them, etc.
- Interoperability
 - devices, information, services
- Dynamicity
 - presence-aware programming
- Localization
 - hub-like relation of locally and globally accessed knowledge

17

Properties (2/2)

- Ambient Intelligence in services
 - Adaptability, personalization, service composition, recommendation, proactive delivery
 - Delegation, mediation
- User collaboration
- Resilience in IoT-settings
- Security

18

Conclusion

- Pilot services for Smart-M3
 - New use cases and business models
 - Emerging market of smart spaces services
- Open source Smart-M3 platform and SDK
 - Leader developer team for the Smart-M3 platform
- Toward a methodology of Smart Spaces development
 - Experience, principles, methods

smart-info@fruct.org

19

Partners



20



M3 in education and research: experiences and lesson learned



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fabio Vergari and Luca Roffia

ARCES

ARCES

DEI

DISI

fvergari@arces.unibo.it

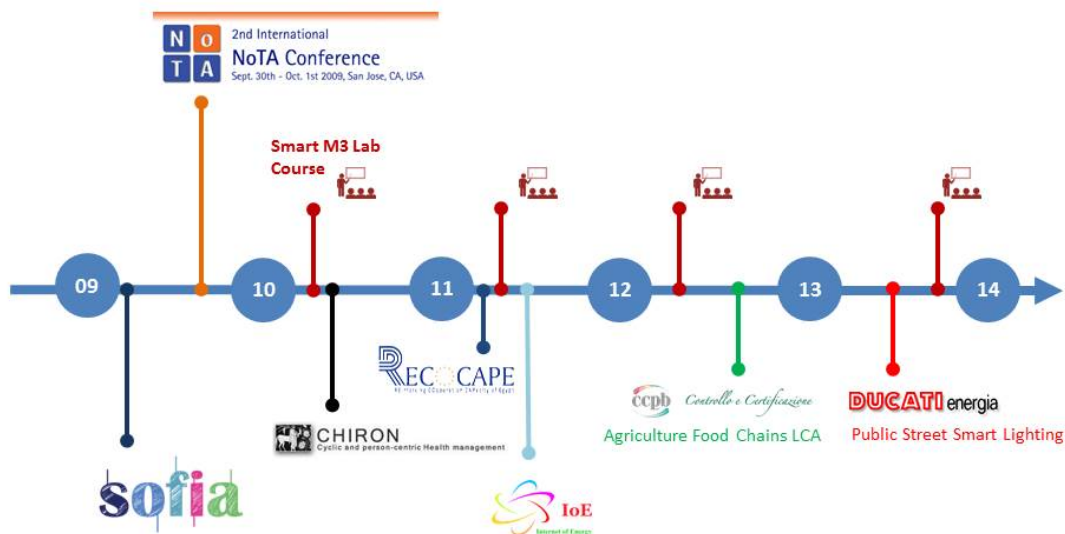
www.unibo.it

Innovation House – Otaniemi
12th November 2013

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



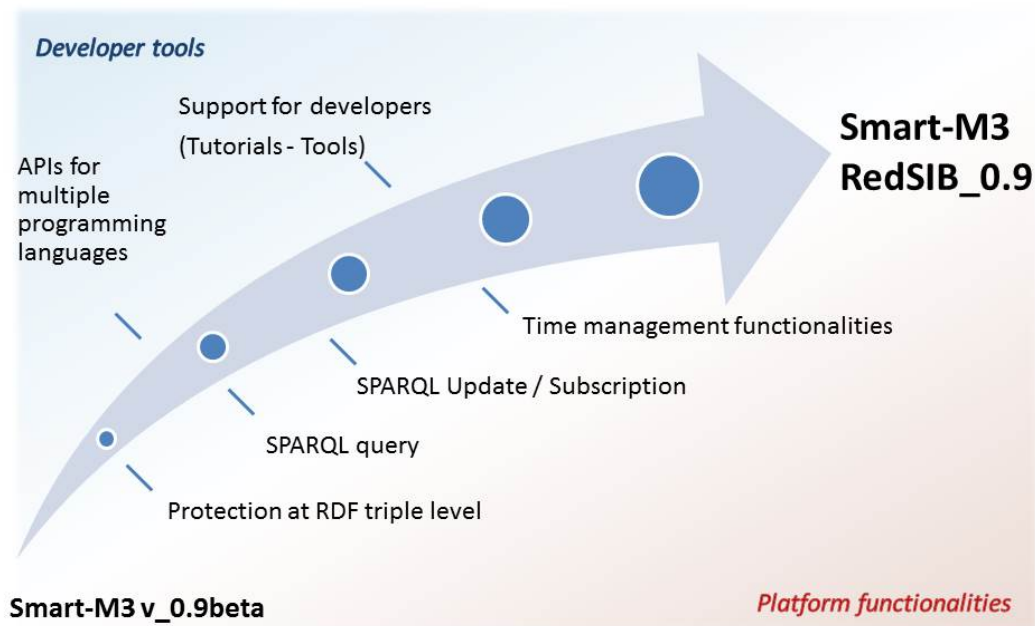
UNIBO & M3 timeline



ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



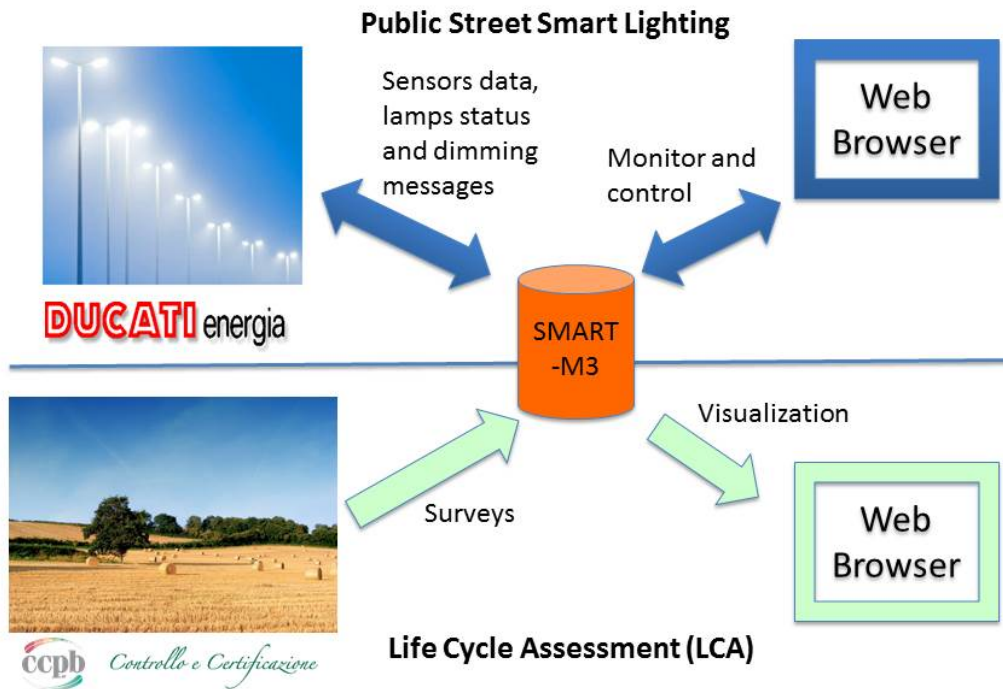
Smart-M3 evolution



UNIBO Contribution

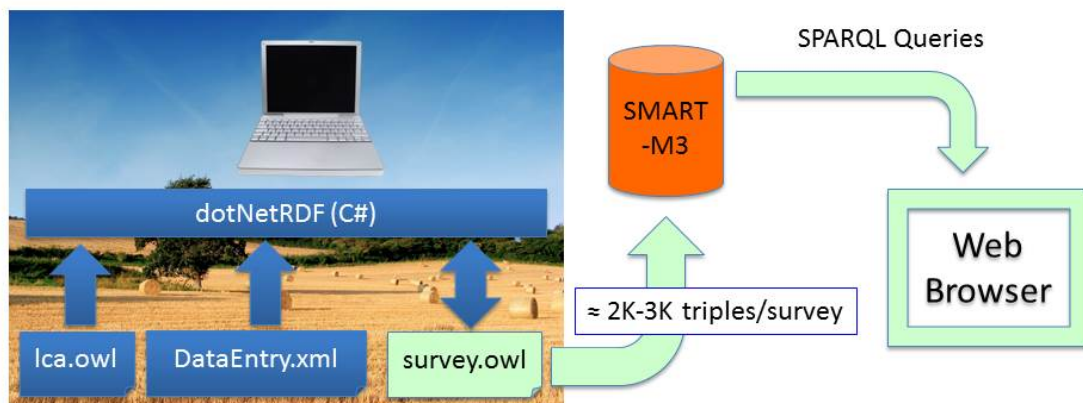
Platform	APIs	Domains	Education
<ul style="list-style-type: none">• Protection• SPARQL support query and subscription• Powered by RedLand or Virtuoso• OSGI SIB	<ul style="list-style-type: none">• Python• Java• C#• PHP• Javascript	<ul style="list-style-type: none">• Public street smart lighting• Agriculture food chains LCA• Home health monitoring• EV charging urban scenario	<ul style="list-style-type: none">• University courses and labs• RECOAPE international cooperation project

Many domains, one solution



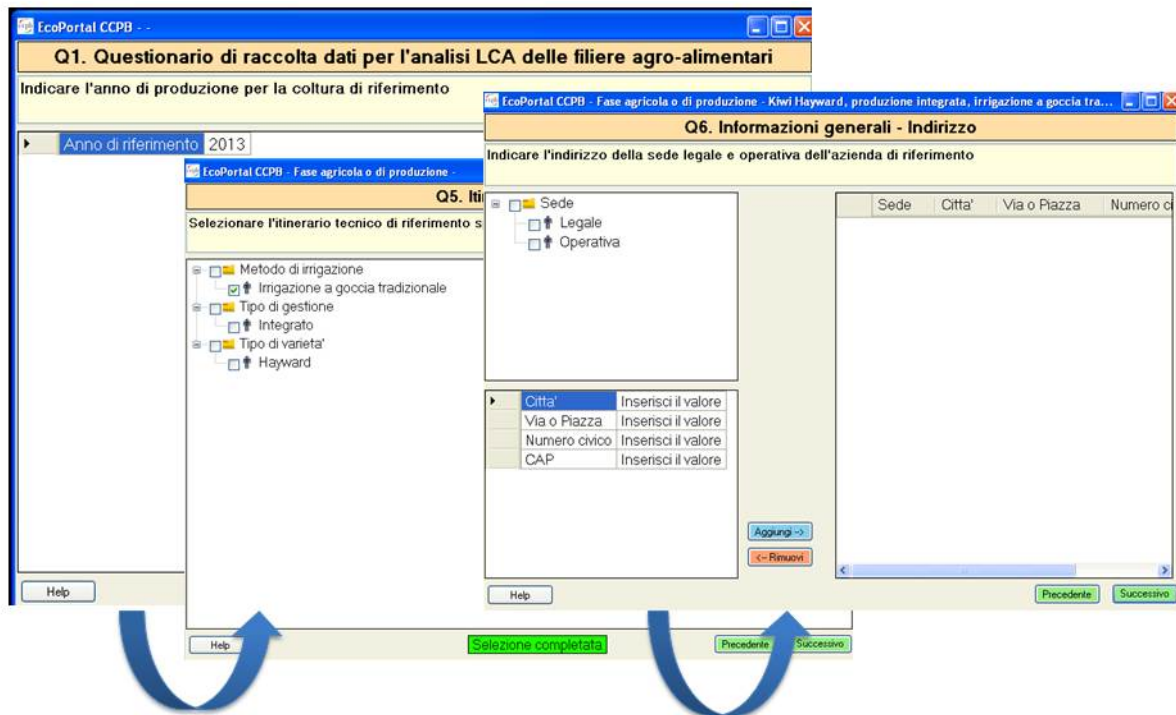
Life Cycle Assessment of agriculture food chains

- Ontology based Data entry GUI for on site operators
- Local caching (i.e. no Internet access) and re-configurable (i.e. ontology based)
- Showing collected surveys on a Web browser





Reconfigurable Data Entry Interface



ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



Survey results visualization using SPARQL queries from PHP

Scheda Questionario

Questionario	
Pero Abate, produzione integrata a bassa densita' - Anno 2011	
AZIENDA AGRICOLA GHELFI BRUNO	

Produzione Complessiva	
Durata economica della coltura (anni)	
Prodotto totale inviato alla fase successiva (q)	225
Resa media nell'ultimo triennio (q/ha)	350
Resa nell'anno di riferimento (q/ha)	210
SAU (ha)	11
Sesto d'impianto	3.5X1
Commento:	

SAU Altre Colture		
SAU (ha)	Varieta'	Tipo di coltura
0,4	CONFERENCE	Pero
3,4	VARIE	Melo
1,2	WILLIAM	Pero
5,5	ABATE	Pero

DEBUG: query time	
Info Questionario	0.06s
Produzione Complessiva	0.052s
SAU Altre Colture	0.237s
Consumi Energetici	0.64s
Produzione di Energia da Fonti Rinnovabili	0.04s
Fase Impianto	4.442s
Impianto Materiali	2.073s
Impianto Irrigazione	0.053s
Impianto Concimazioni	0.061s
Impianto Fitosanitari	0.044s
Operazioni Culturali	7.451s
Operazioni Materiali	0.06s
Operazioni Concimazioni	1.425s
Operazioni Fitosanitari	0.065s
Materiali in Uscita	0.056s
Trasporto Fase Succ.	0.04s
Totale	16.799s

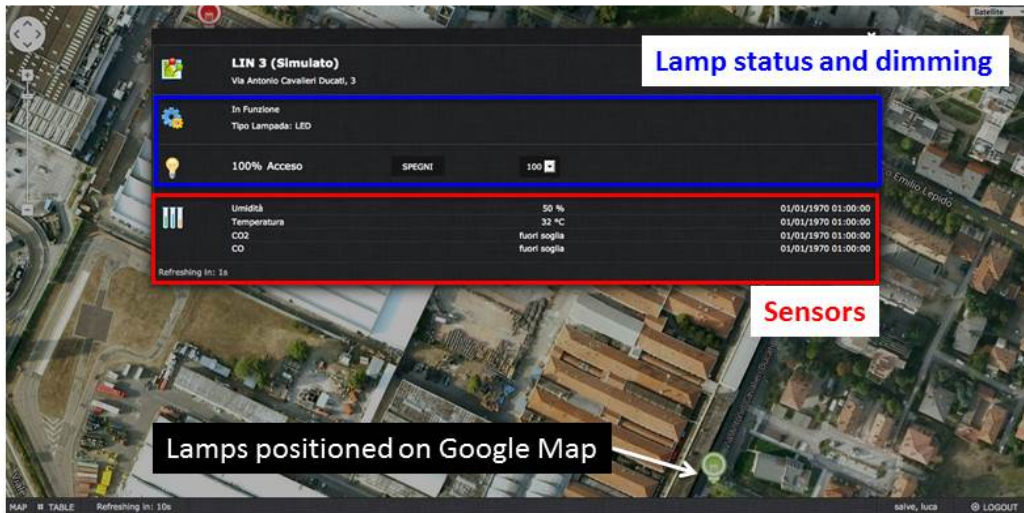
http://mml.arces.unibo.it/sib_ducati/cms/admin/it/questionari_async/modifica/Questionario_634952516621197509

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



Smart Lighting on Public Streets

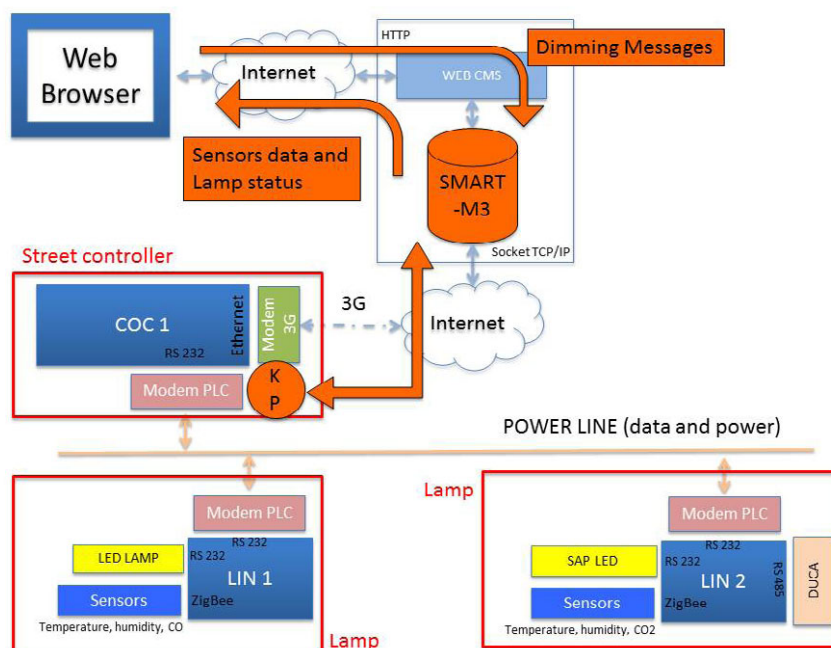
- Point-to-point lamp dimming
- Sensors (e.g. temperature, humidity, CO, CO2) on each lamp
- Power line communications



ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



Smart Lighting on Public Streets: Ecosystem Architecture



ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



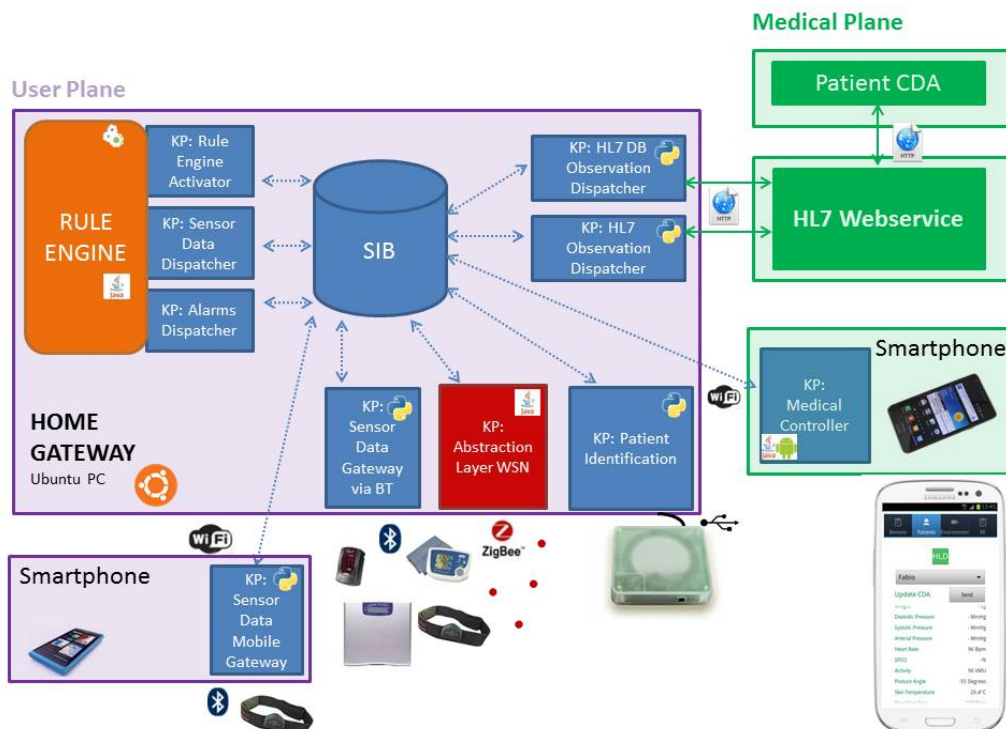
Health monitoring at home ARTEMIS Project CHIRON

- Patient at home
 - Local gateway to collect sensor data
 - Heterogeneous physiological and environmental sensors
- Local gateway services
 - Rule engine to detect anomalous situations
- Provide information to the medical world
 - Medical standards (HL7)
 - Standard solutions (Database, SMS services)

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



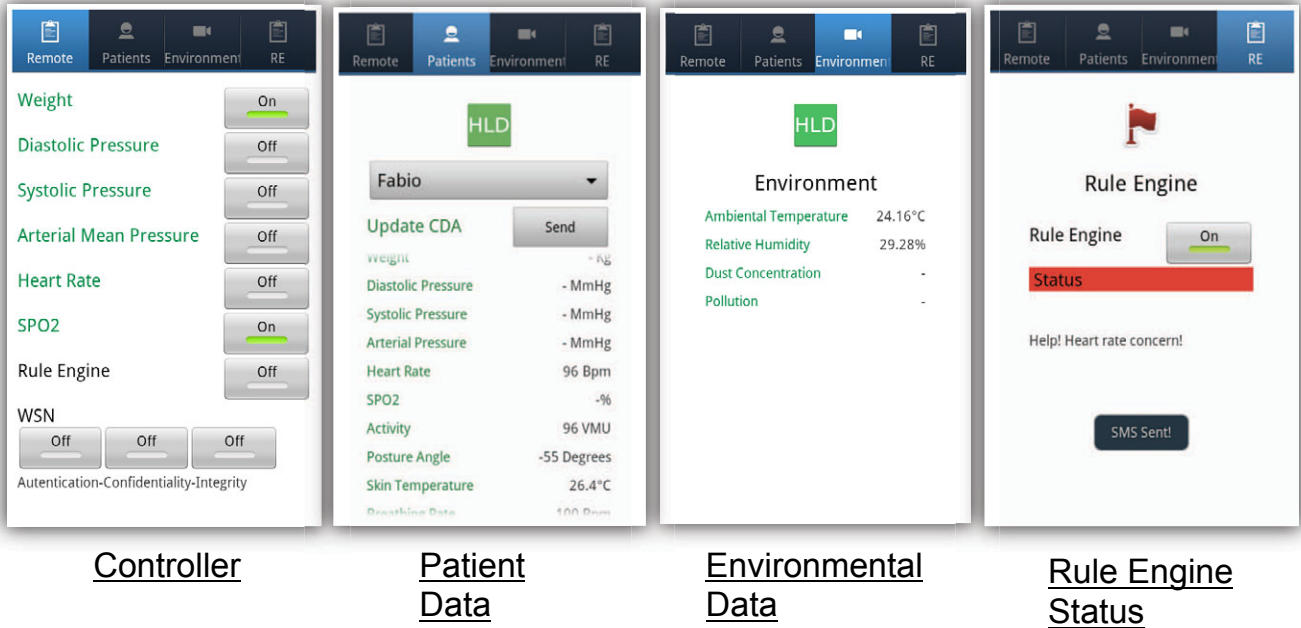
CHIRON ecosystem architecture for health monitoring at home



ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



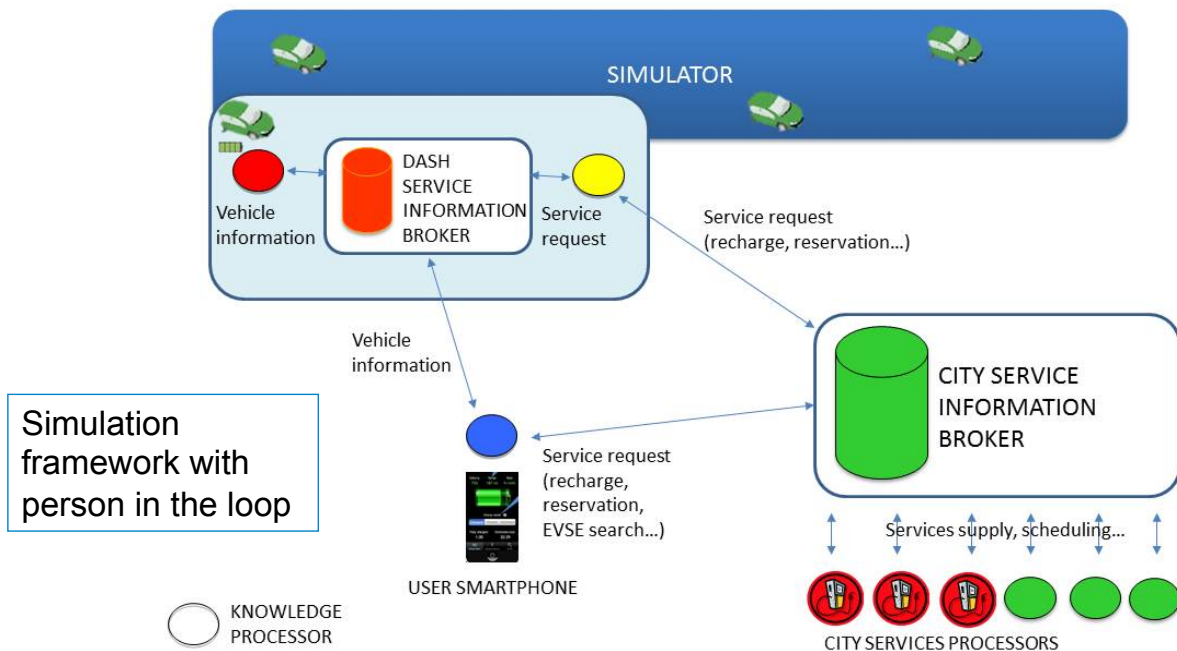
GUI for Health monitoring at home in CHIRON



ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



ARTEMIS Project Internet of Energy: Electric Vehicles Recharging in Urban Scenarios



ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



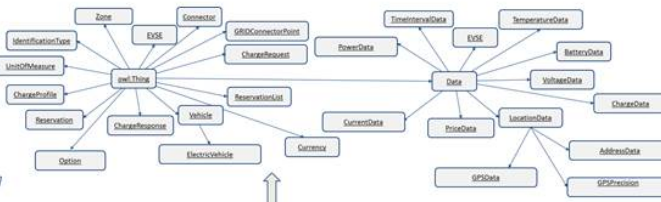
IoE: Ontology, models and GUIs for Electric Vehicles recharging in urban scenarios

- ▶ Information interoperability
 - ▶ SMART M3 middleware
 - ▶ Common ontology

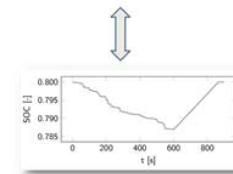
▶ Mobile Services

The mobile services interface includes several components:

- Current Values:** Displays real-time data such as State of Charge (SOC), Temperature, State of Health, Current Density, and Current Density Est.
- Reservation Observer:** A table showing reservation details for various vehicles.
- IoE Reservation Observer:** A larger table providing a comprehensive overview of reservations, including vehicle IDs, EVSE IDs, GCP names, start/end dates, and battery levels.



▶ Urban traffic simulation



▶ EV battery model

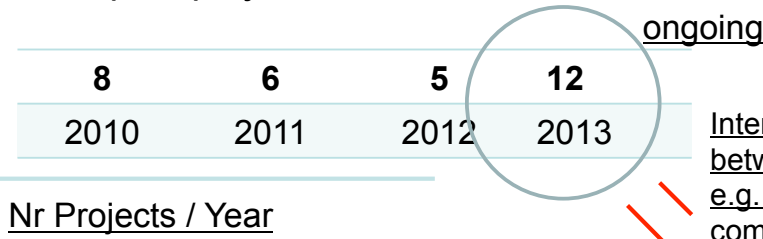
In co-operation with SIEMENS and AICIA (Seville)



Smart-M3 Lab Lab of Interoperability of Embedded Systems

- Starting from academic year 2009/10:
 - the lab provides a design style
 - the lab is a test bed for research results
- Topics: Semantic Web, smart space applications, Smart-M3 programming approach, design template
- Students are divided in teams of 3 to 5 persons. Each group proposes and develops a project

Projects spans from applications (E.g. Games, domotics, ...) to APIs (E.g. PHP and JavaScript)



Interaction between projects, e.g. games use a common ontology



Smart-M3 Lab Contribution

Student projects and feedback

Course Website

<http://didattica.arces.unibo.it/mod/resource/view.php?id=468>

- Course Description
- Tutorials
- Tools
- LabExercise and Projects

Tutorial

SMART-M3 v.0.9: A semantic event processing engine supporting information level interoperability in ambient intelligence

<http://amsacta.unibo.it/3877/>

Available under License [Creative Commons Attribution Non-commercial \(CC BY-NC 3.0\)](https://creativecommons.org/licenses/by-nc/3.0/).

Smart M3 in Embedded Devices Communities

Francesco Morandi

OVERVIEW

Smart M3 particularly fits to linux-based embedded devices (ARM).

- ✓ Linux based C-Sib is easy to compile and install as ARMHF packages. (It runs pretty well on Raspbian)
- ✓ Most of the projects involving these devices are close to the SOFIA scenarios (Smart Environments)

Mainly used in education and research, widely spread also in the industry

The largest community is Raspberry Pi (Arduino Galileo and Parallela have very interesting hardware and they are also growing up as communities).

Raspberry Pi and Intel Galileo hardware platforms

Raspberry Pi:

- ARM1176JZF-S (ARMv6k) 700 MHz.
- Memory:
 - 256 MB (Model A)
 - 512 MB (Model B rev 2)

Price: 35 Euro

Intel Galileo:

- 400 MHz 32-bit Intel Pentium instruction set architecture (ISA)
- 512 KBytes of on-die embedded SRAM
- ACPI compatible CPU sleep states supported
- 10/100 Ethernet connector
- Full PCI Express* mini-card slot, with PCIe 2.0 compliant features
- USB 2.0 Host connector

Price: Still to be defined

Parallela Board hardware platform

- Zynq-7000 Series Dual-core ARM A9 CPU (Z-7010 or Z-7020)
- 16 or 64-core Epiphany Multicore Accelerator
- 1GB RAM
- MicroSD Card
- 2x USB 2.0
- 10/100/1000 Ethernet
- HDMI port

➤ Ships with Ubuntu OS

Price: 99 \$

Conclusions

Opportunities:

- These devices have extremely large **communities** and can give SMART-M3 a huge visibility.
- Being open source, designs can be extended and more groups can contribute.

How to succeed:

By showing how Smart M3 solutions can improve the state of the art of the current Raspberry based projects.

What should we provide:

- Very clear programming model, to be perceived as “simple” from the developer.
- Ontology modeling guidelines
- **Reference Web Site** with tutorials



An OCPP interoperable services system based on a SmartM3 core

Á. Rodríguez-Serrano,
A. Torralba, E. Rodríguez-Valencia, J. Tarifa-Galisteo

*Electronic Engineering Group (GIE), Department of
Electronic Engineering, University of Seville*



2013 Open International M3 Semantic Interoperability Workshop

Helsinki, 12nd Nov 13

OUTLINE

- Introduction
- AICIA-GIE in IoE project
- Communication technologies
- Semantic Web
- Smart-M3 & OCPP
- Mobile Application
- Validation & Results
- Conclusions

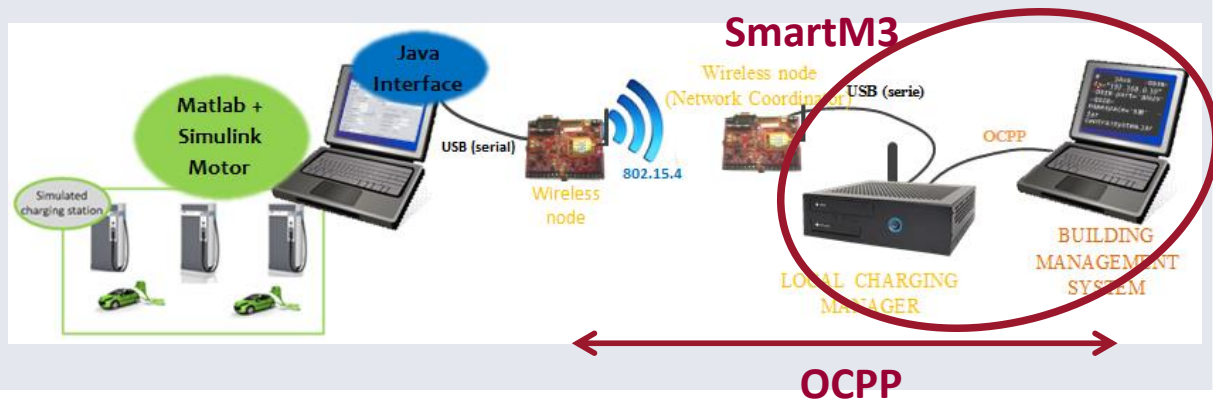


12/11/2013

2

Introduction

- **Electric Mobility** is becoming a reality.
 - » New **ICT services** and tools are needed.
 - » **Standardization** effort is crucial.
- **Semantic interoperability** and **Smart Environments** are pillars for the growing Semantic Web and IoT.



12/11/2013

3

Introduction

- **ARTEMIS IoE project:** Integration and interfacing between the power network represented by the **grid** and the data network represented by the **Internet** and focusing on transmission, substation and distribution control, metering, substation monitoring and diagnostics and location information systems into **seamless and coherent Internet of Energy**.



12/11/2013

4

AICIA-GIE in IoE project

➤ Our tasks

- » Developing a **standard communication network** applied to **electric vehicle system** → **OCPP**
- » Working towards an **open source platform** able to integrate different communication technologies based on **IP** → **SmartM3**
- » Focused on **wireless radio technologies** for smart metering and for the integration of sensors and actuators → **IEEE 802.15.4**
- » Contributing to the **user interface** through a smartphone, an application for remote monitoring of charging in progress → **Android app**

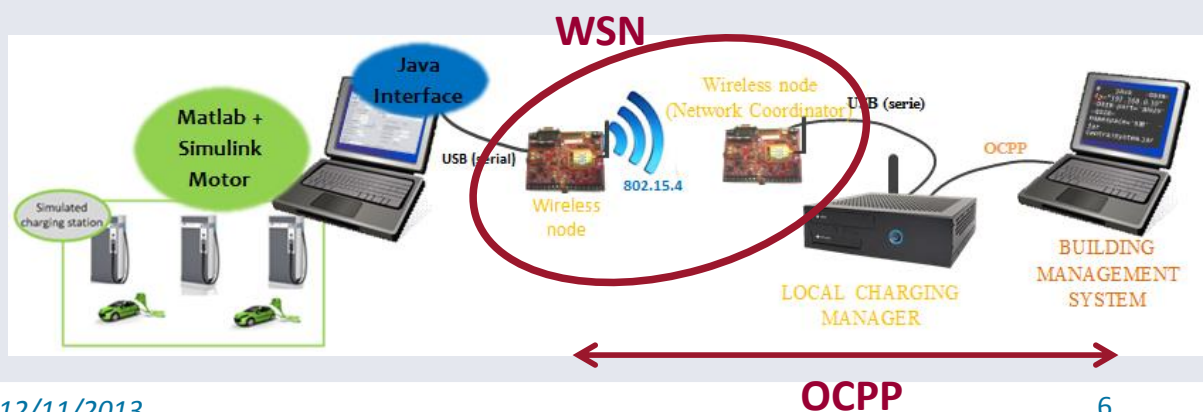
12/11/2013

5

Communication technologies

➤ Communications technologies:

- » WSNs: **IEEE 802.15.4** for Charging Stations (EVSE – Local Charging Manager).
- » **SSAP/IP** (OCPP ontology based, SmartM3) between Charging Stations and Central System.
- » **SOAP/IP** (pure OCPP) between Charging Stations and Central System for third parties.



12/11/2013

6

Semantic Web



- **SmartM3** is a Semantic Web open-source SW project → Smart Environments (born in ARTEMIS SOFIA project)
 - » Components: **Ontology** (data structure), **SIB** (data store) and **KP** (applications).



- System Smart-M3 based:
 - » Ontology models EV infrastructure entities. We share ontology with UNIBO.
 - » **Two SIBs**: EV data SIB & Charging Historical data SIB.
 - » **Three KPs**: GatewayIoE, CentralSystem and Phone.

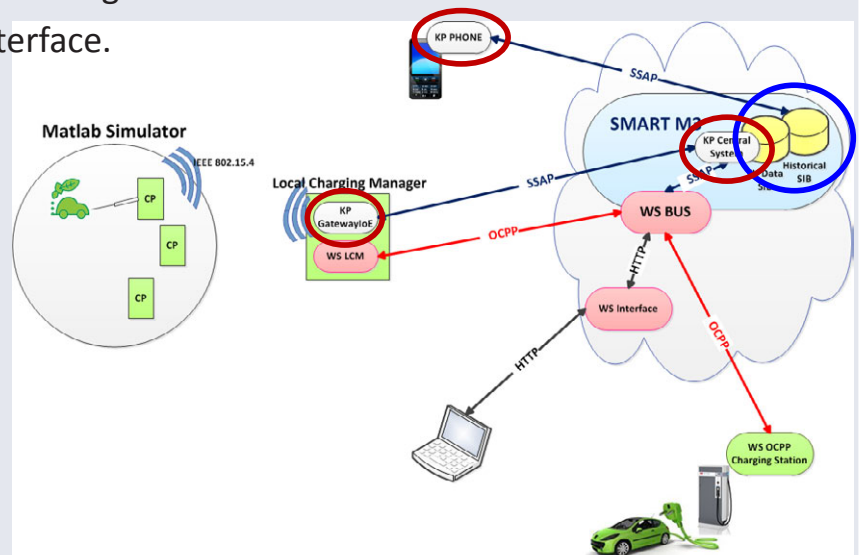
12/11/2013

7

Smart-M3 entities



- **Two SIBs**: EV data SIB & Charging Historical data SIB.
- **Three KPs**: GatewayIoE, CentralSystem and Phone.
 - » **Gateway IoE**: receives the data from the CP, processes and sends to the Central System.
 - » **CentralSystem**: manages the interaction with the SIBs.
 - » **Phone**: user interface.



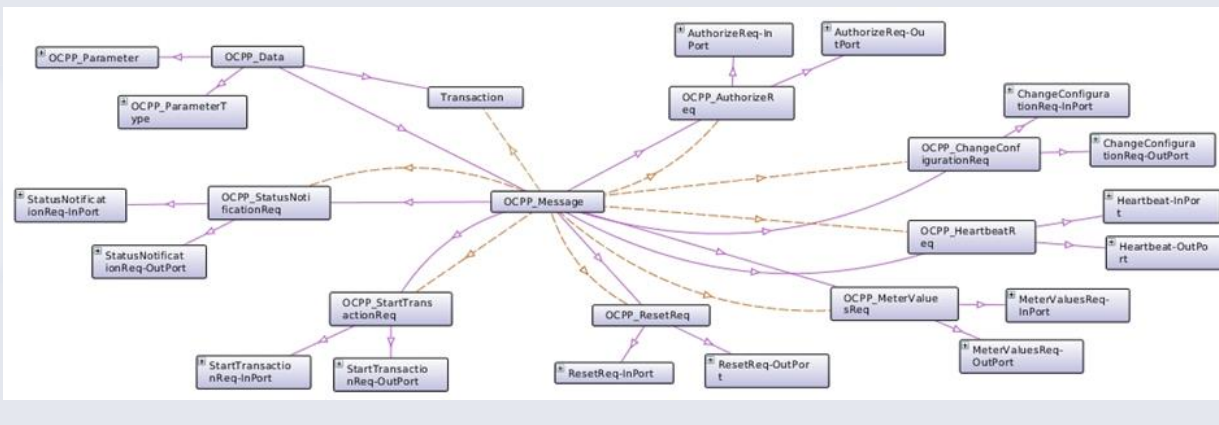
12/11/2013

8

Smart-M3 & OCPP

➤ OCPP interoperable:

- » Applying OCPP model data to the specification of the ontology.
- » Developing an interface to enable other “pure” OCPP systems to use the SmartM3 technology in an interoperable and transparent way.



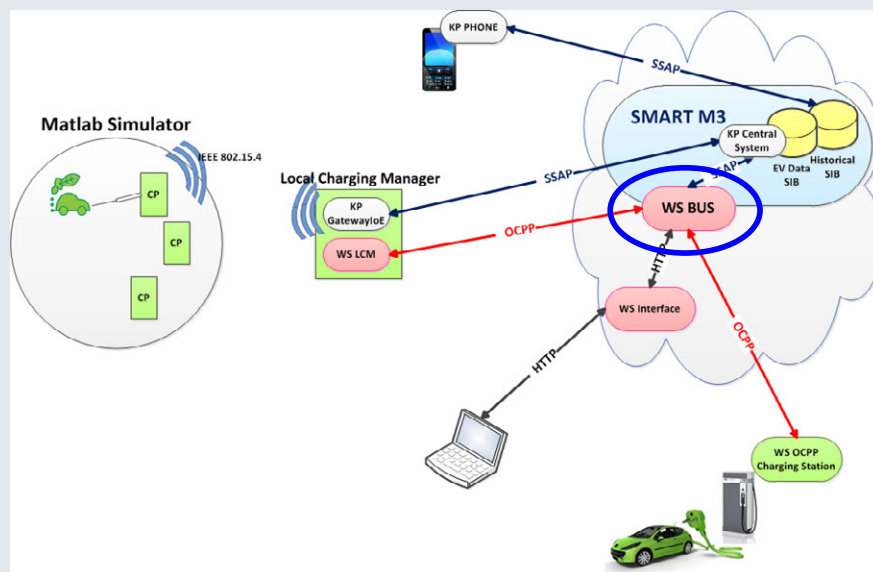
12/11/2013

9

Smart-M3 & OCPP

➤ OCPP interoperability using Web Services:

- » WS Interface: Implements a web-based interface.
- » WS LCM: OCPP at local level.
- » WS BUS: **Gateway OCPP/SmartM3 core.**

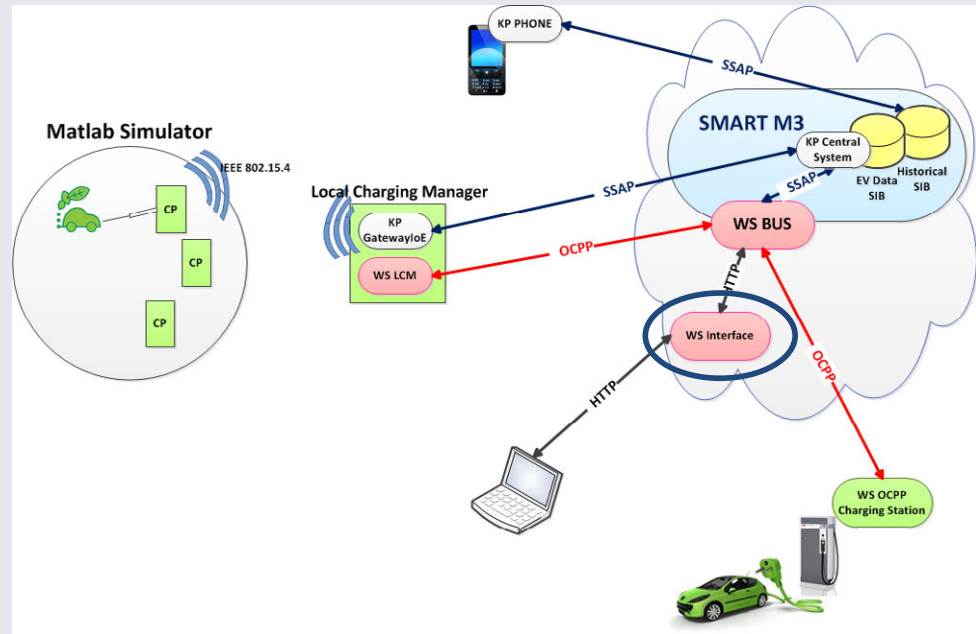


12/11/2013

10

Web Services functionality

- **WS Interface:** Implements a web-based interface. Based on OCPP and xml, display the information received from the SIB in a Web-based interface.

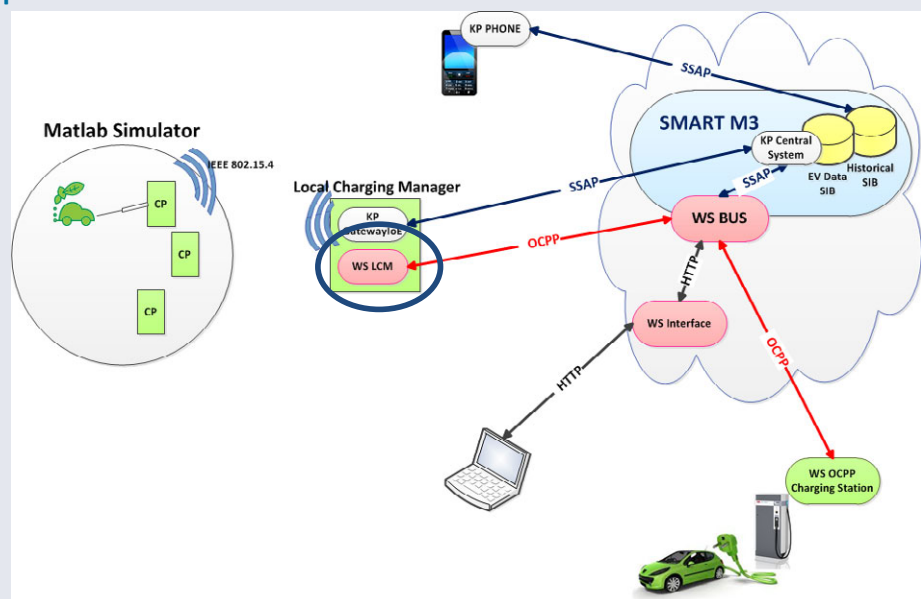


12/11/2013

11

Web Services functionality

- **WS LCM:** It allows the interaction with the SIB using OCPP. It receives the queries from the WS BUS and responds with the corresponding results when the SIB has performed the required operation.

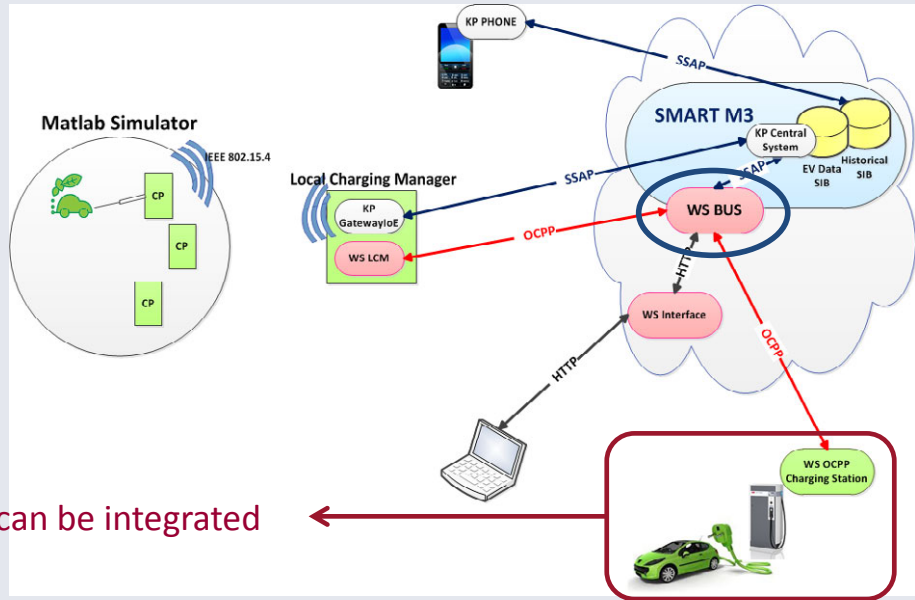


12/11/2013

12

Web Services functionality

- **WS Bus:** Core for OCPP interoperability, acting as a “virtual” gateway to the Smart-M3 Central System. Common entry point for third systems, receiving the queries coming from external web services.



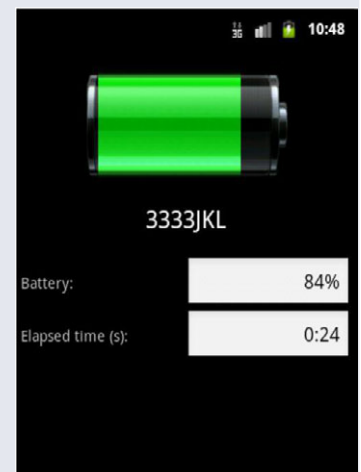
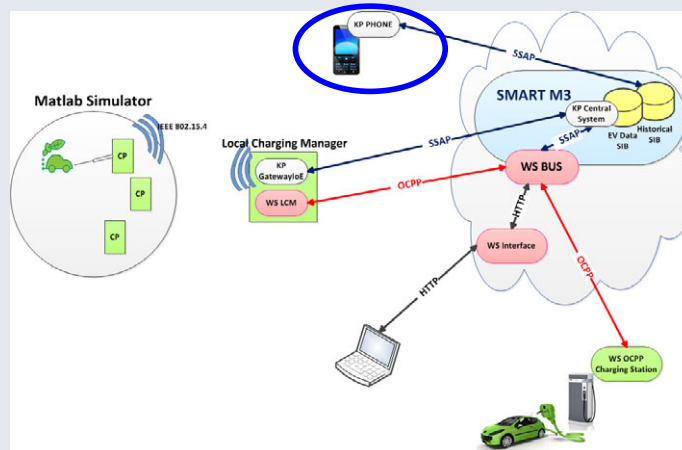
Third parties can be integrated

12/11/2013

13

Mobile Application

- A mobile application for Android® systems has been implemented to **monitor the EV charging process in the real time and to implement reservation service**, using subscriptions to the SIB. This application provides information about the user account, associated vehicles, previous recharges, etc.



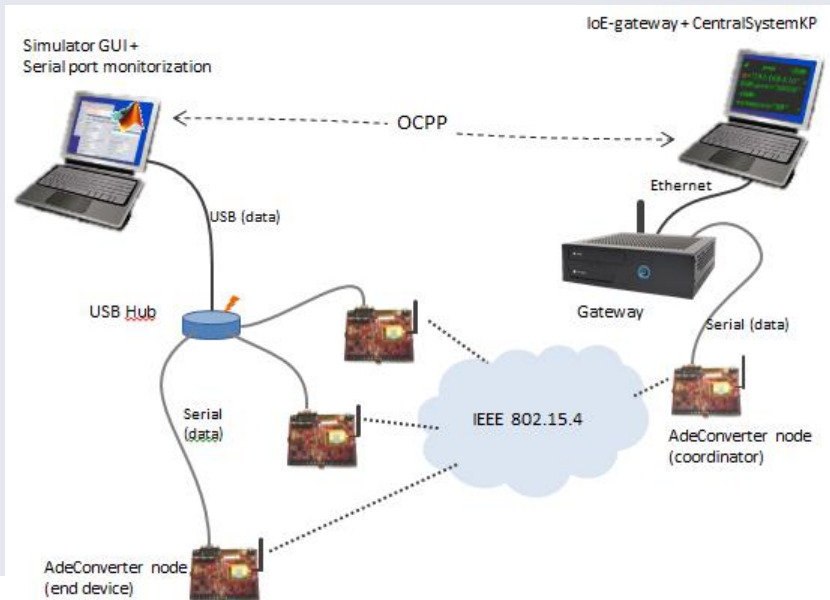
12/11/2013

14

Validation & Results

Tests:

- » Component tests → Functionality by unit.
- » System tests → Integration and interoperability.
- » Stress Tests → WLAN robustness.



12/11/2013

Conclusions

- WSNs provide wireless communication to EV Charging Stations (incoming wireless charging).
- Semantic web – SmartM3 provides:
 - » Interoperability with other services (Ontology specification).
 - » Scalable solution.
 - » Low-layers platform independence.
- OCPP provides:
 - » Interoperable and standard EV communication system.
 - » Independence among EVSE suppliers and Service Provider.
- Complete system, from user to Service Provider.
- Open source system that works.

12/11/2013

16



THANK YOU FOR YOUR ATTENTION

Ángeles Rodríguez Serrano

arodriguez@gie.us.es

Electronic Engineering Group (GIE), Department of
Electronic Engineering, University of Seville



12/11/2013

17

Resolution Infrastructure for M3 based Systems

M3 Semantic Interoperability Workshop

Jussi Kiljander

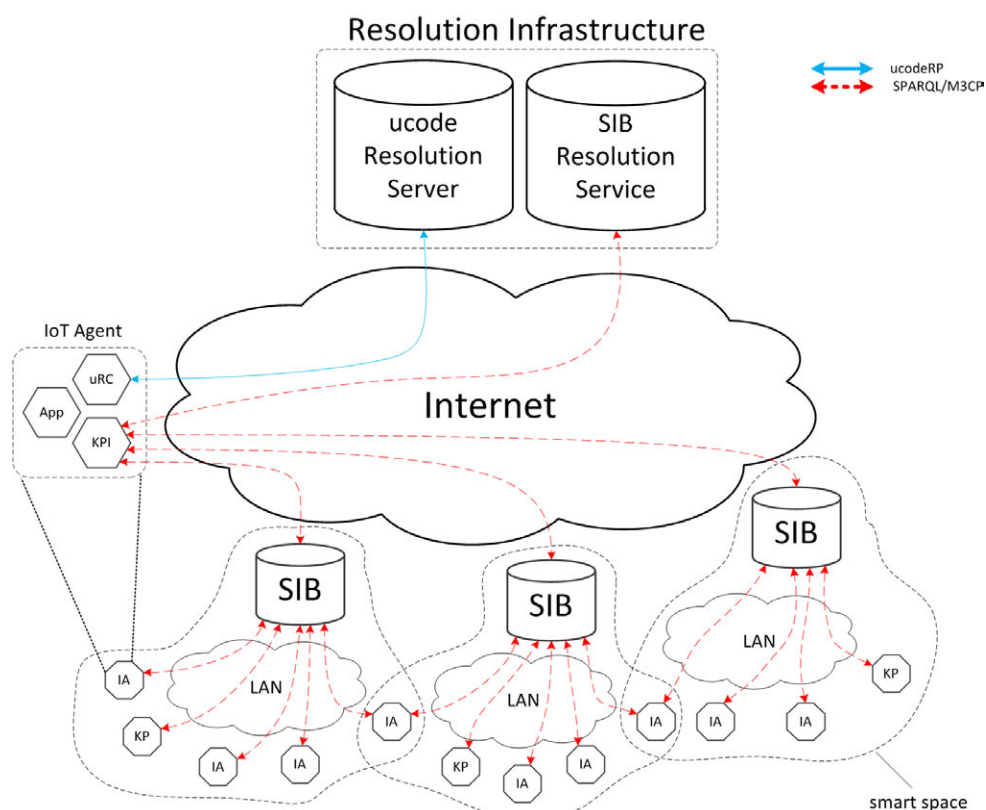
VTT Technical Research Centre of Finland

Introduction

- Internet of Things (IoT) can be seen as a global scale extension of ubiquitous and pervasive computing paradigms.
- When the idea of M3 based semantic interoperability is extended from local pervasive computing environments to the IoT there is a need for:
 - 1) Global identification methods for the physical objects
 - 2) Resolution infrastructure that provides methods to resolve the address of a SIB containing information about the physical objects of interest
- To this end, we combine M3 with uID Architecture and propose:
 - an identification scheme based on *ucodes*
 - a novel resolution infrastructure consisting of ucode Resolution Server and SIB Resolution Service

Ubiquitous Identification Architecture

- The uID architecture (from uID Center, Japan) is a ubiquitous computing solution that provides methods to uniquely identify and locate information about physical world objects.
- Central concepts in uID architecture are:
 - **ucode:** The *ucode* is 128 bit code without a meaning that can be stored into any kind of tag medium (e.g. RFID, BLE, NFC, QR code).
 - **ucode Information Server:** The *ucode Information Server* is a database that contains information about an object identified with a *ucode*.
 - **ucode Resolution Server:** The role of the *ucode Resolution Server* is to both manage *ucodes* and resolve *ucodes* to corresponding *Information Server* addresses. In practise this functionality is achieved with three operations: *issue*, *change entry*, and *resolve*.



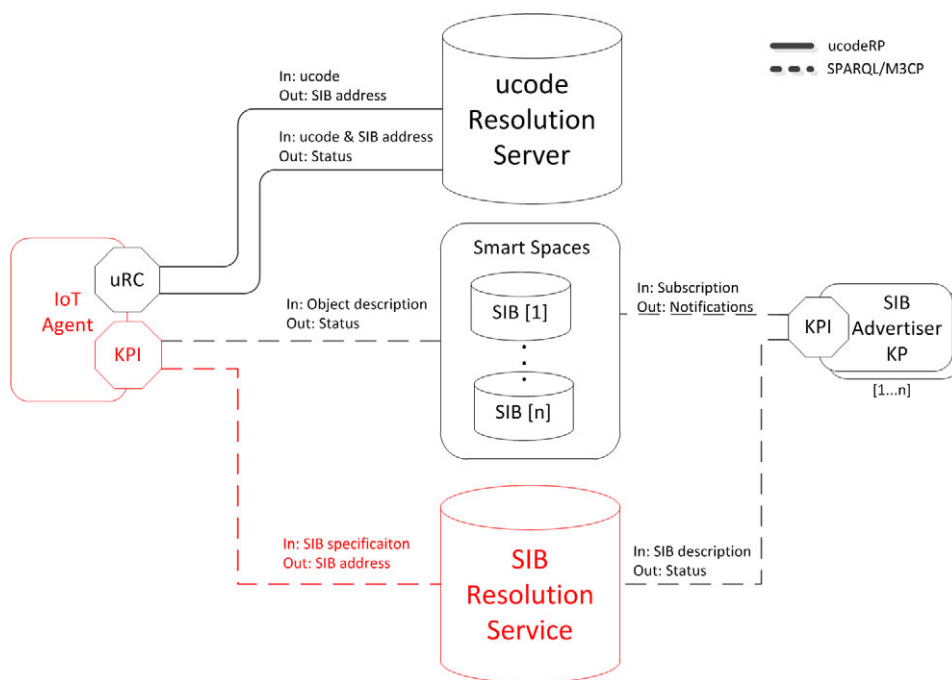
VE Identification

- The *ucode* is used as unique identifier for both the physical object and its virtual counterpart.
- The *ucode* based URN is used instead of HTTP URI recommended for Linked Data mainly for three reasons:
 - 1) M3 does not use HTTP protocol and implying that the resource is available via HTTP is a bad practice.
 - 2) The HTTP URI points always to a specific network location. This causes problems in typical IoT applications where data can be distributed into multiple heterogeneous information systems.
 - 3) The *ucode* has a fixed length and it is thus more predictable in term of memory and also faster to process.

SIB Resolution

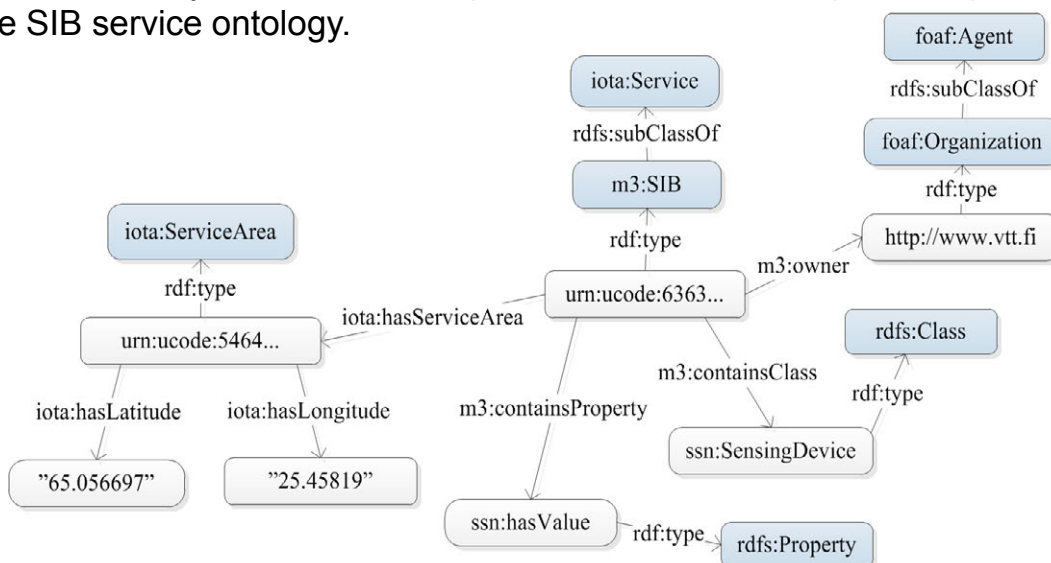
- Resolution Infrastructure is responsible for SIB resolution in the Internet of Things.
- Two type of resolution operations are provided:
 - **Lookup:** SIB address containing information about the physical object is resolved based on the object ID (i.e. *ucode*).
 - **Discovery:** SIB address is resolved based on SIB specification expressed as a SPARQL query.
- The Resolution Infrastructure consist of two components:
 - *ucode* Resolution Server
 - SIB Resolution Service

SIB discovery

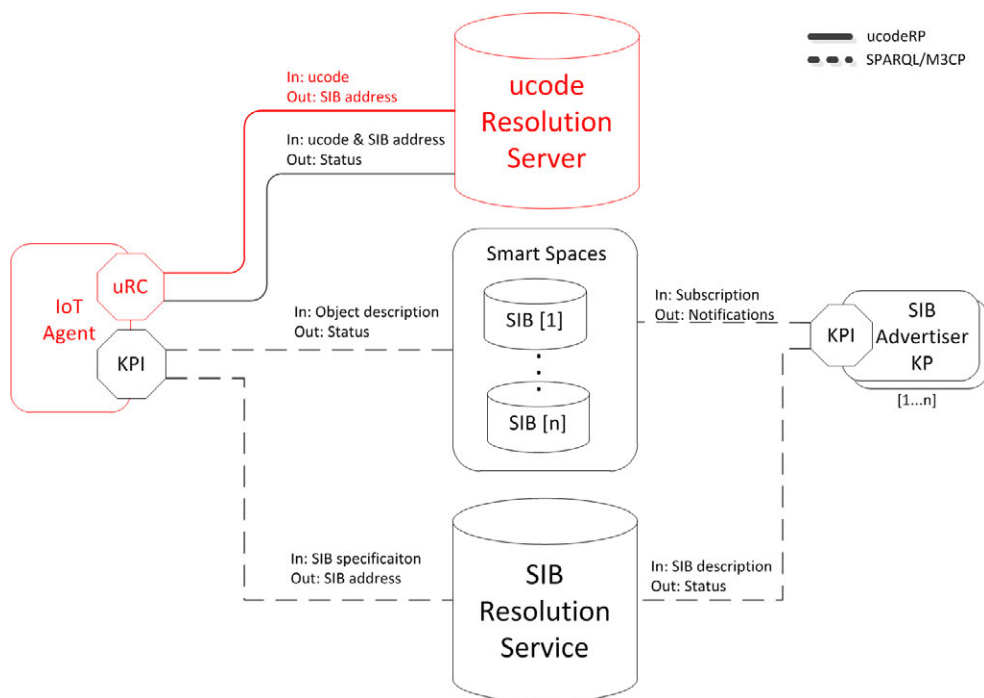


Ontology for SIB Service descriptions

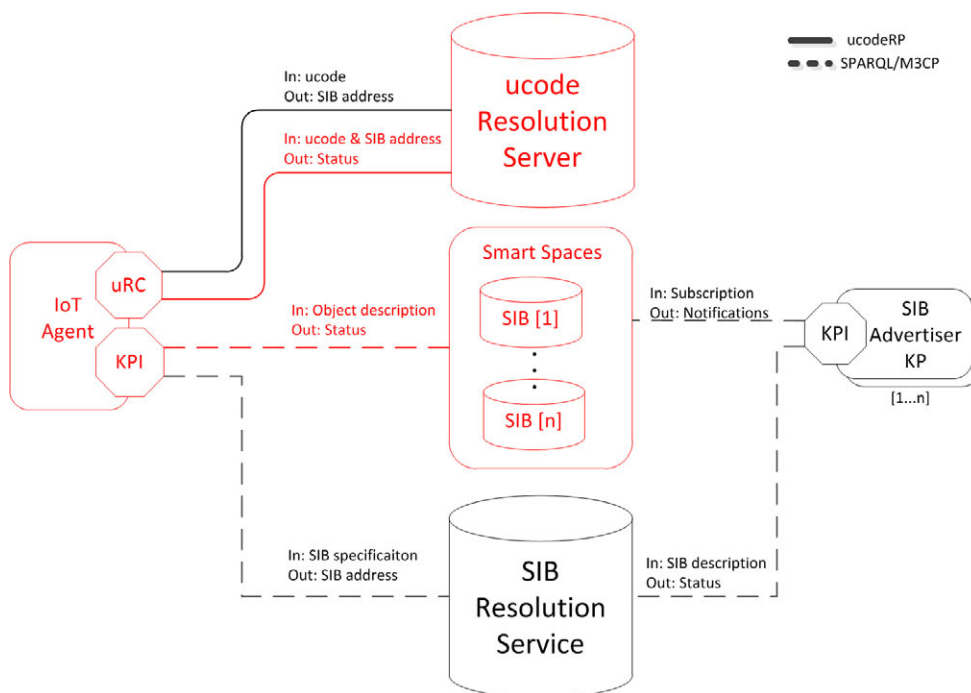
- The vocabulary for SIB service specifications and descriptions is presented in the SIB service ontology.



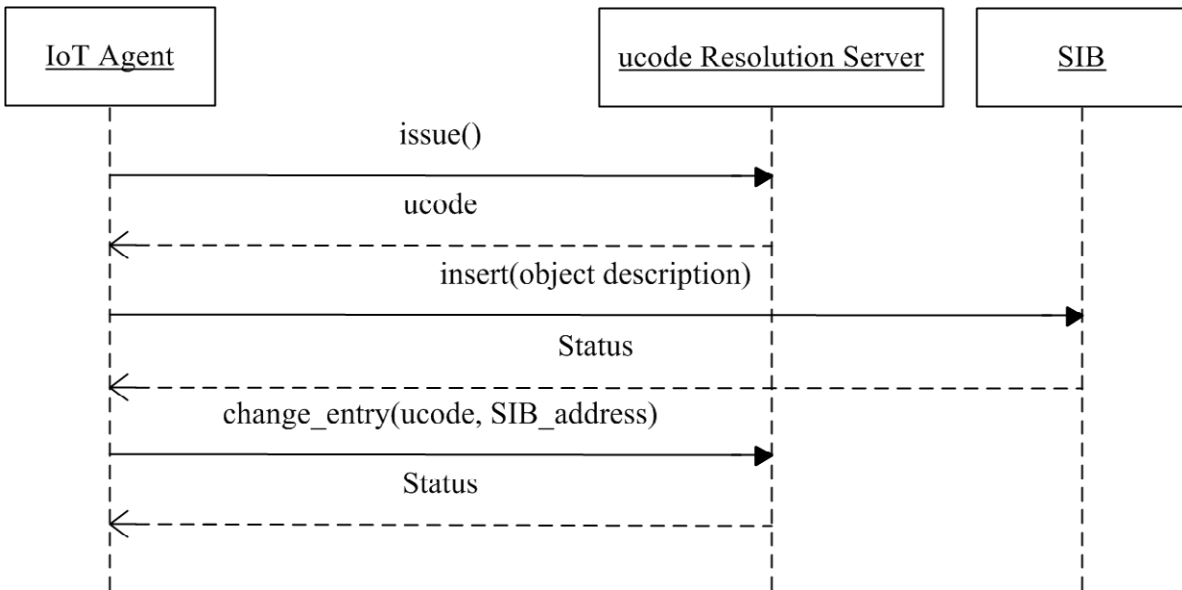
SIB Lookup



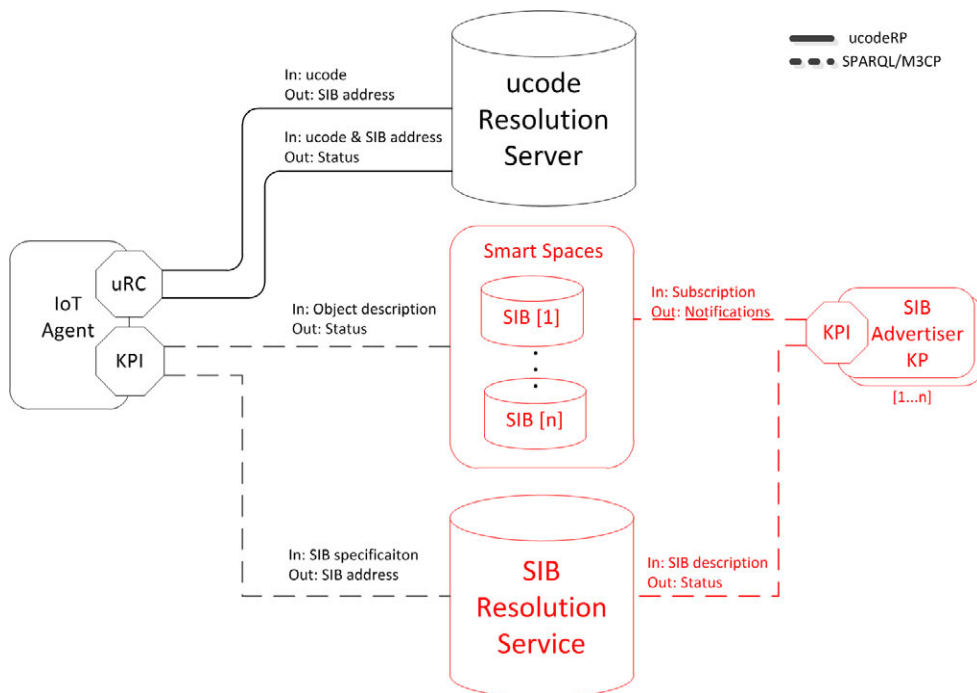
Creating and Publishing object descriptions



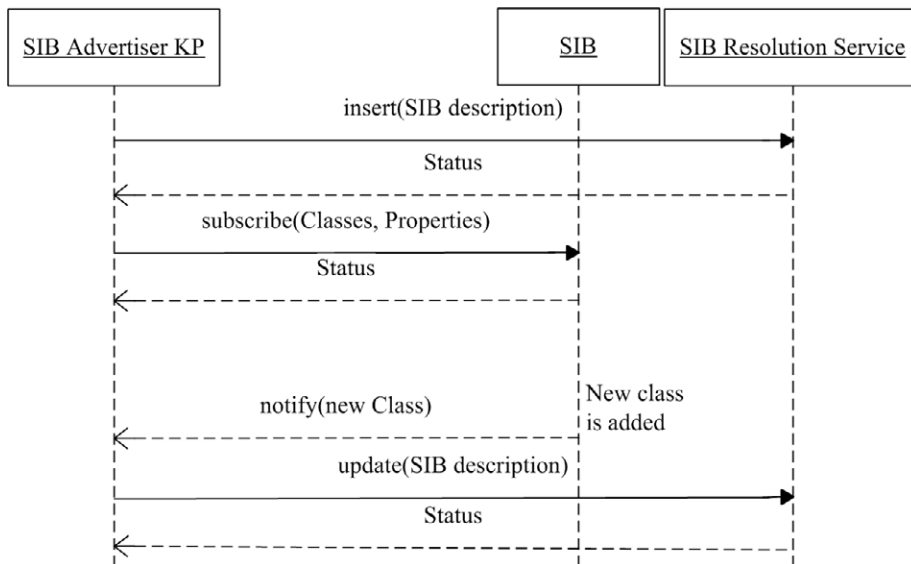
Creating and Publishing object descriptions



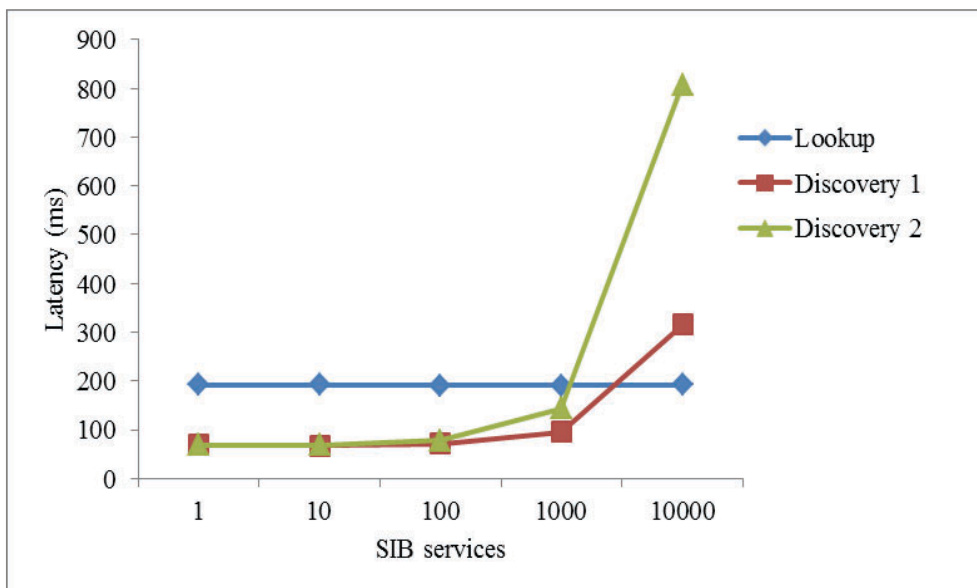
SIB service management



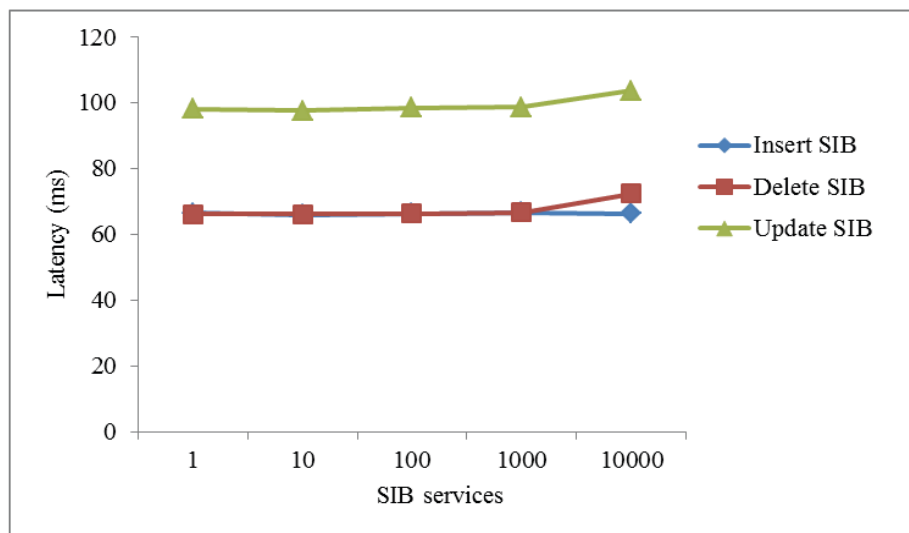
SIB service description management process



Evaluation: SIB resolution



Evaluation: SIB management



Evaluation: observations

- The discovery operation has significant increase in latency when the service count is increased from 1000 to 10000 services.
- Network latency constitutes majority of the latency for most operations:
 - In SIB *discovery* and management operations with 100 SIB services the network latency is around 95 %
 - In SIB *lookup* the network latency part is 78%
- If we ignore network latency the Resolution Infrastructure is able to process around 350 *discovery* and *management* operations per second (assuming maximum of 1000 SIB services).

Conclusions and Future Work

- The Resolution Infrastructure scales well up to 1000 SIB services and 100000 VEs (evenly distributed)
- Network latency is a big factor in the communication and could be optimized by utilizing more compact M3 communication protocols.
- In order to make the resolution infrastructure scalable for large scale IoT systems a distributed implementation of the *SIB Resolution Service* is needed in the future.



Thank You!

M3 Technology

A scalable distributed M3 platform on a low-power cluster

Anders Berg Petteri Karvinen Stefan Gronroos
Frank Wickstrom Shohreh Hosseinzadeh Natalia Diaz Rodriguez
Johan Lilius

Department of Information Technologies
Abo Akademi University, Turku, Finland

In order to facilitate the possibility for senior members of society to live at home, and lead an active life for as long as possible, it is often desirable to collect and process a large amount of sensor data. Sensor data includes vital signs such as heartbeat and blood pressure, as well as input from various in-home sensors to detect activities and events that might require further actions to be taken. In addition, high data-rate video streams might need to be stored and processed as well.

Most of this collected data could be stored and processed on-site without having to transport all data to a remote data center. In order to achieve this, we propose a compact data store consisting of one or several (depending on how much data storage and/or processing capacity is required) low-power computing nodes consisting of hardware similar to that found in mobile phones. This choice of hardware allows us to create a small form-factor, easily hidden device with modest cooling requirements. Currently our design is based on the Smart-M3 software, extended to connect to a distributed RDF store backend. We are also re-implementing the subscription handler of Smart-M3 using the RETE algorithm to facilitate improved handling of a large number of subscriptions with a large amount of stored RDF triples. Furthermore, we are developing security and privacy models for the platform.

Figure 1 illustrates the intended high-level architecture of the proposed platform. This architecture spreads the data storage engine out over several nodes if needed, while the middleware and RETE engine are running on a master node. In future iterations, we hope to also distribute the RETE engine and most of the middleware over the cluster nodes. Thus far, we have implemented the connection of Smart-M3 to the 4store distributed RDF store, and finished most of the integration of the CLIPS RETE engine with Smart-M3. The implementation is run on a four-node cluster of HardKernel ODROID-X (quad-core ARM Cortex-A9 CPU) nodes. We are evaluating the performance of the clustered approach as well as identifying any performance bottlenecks that appear as a result of the choice of a low-power architecture compared to traditional server hardware.

A RESTful client API will be available to query the data store, as well as to set up subscriptions for interesting data. To increase interoperability with the applications developed within the Health and Wellbeing action line of the EIT ICT Labs, we also plan to connect the proposed platform to the Personal Health Labs (PHL) data store, in order to enable the exchange of data between the two platforms.

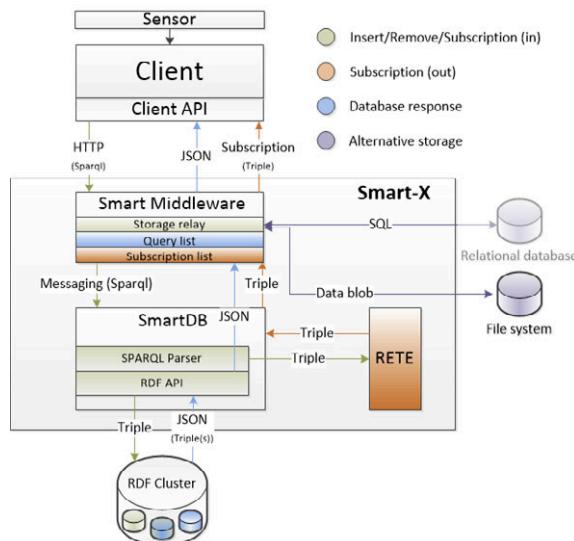


Figure 1: Architectural overview of the clustered data store.



Figure 2: The experimental cluster platform

A scalable distributed M3 platform on a low-power cluster

Anders Berg, Petteri Karvinen, Stefan Grönroos, Frank Wickström, Shohreh Hosseinzadeh, Natalia Díaz Rodríguez, Johan Lilius

Åbo Akademi University

November 2013

Introduction

Introduction



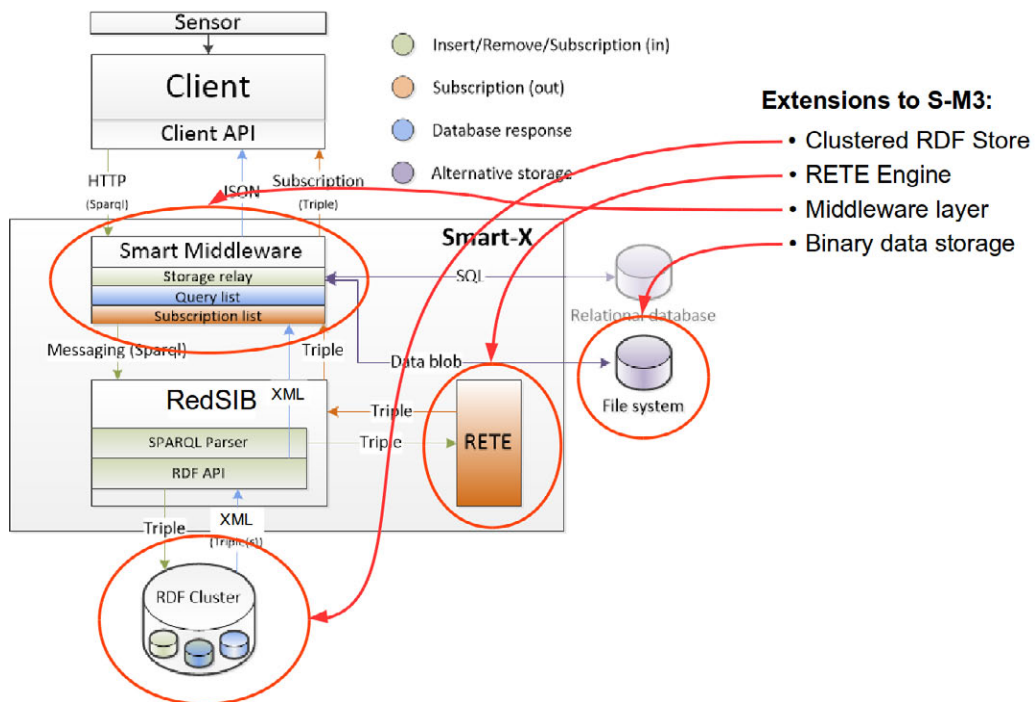
- Requirement: Small Smart-M3-based box
- In-home Smart-M3 box collects and processes
 - Sensor data (heartbeat, blood pressure, presence sensors, etc.)
 - Video streams (for example from wearable camera)
- Needs to process a large number of input data streams and handle a large number of subscriptions efficiently

The Prototype Box



- 4 ODRROID-X ARM development boards
- Cluster connected by 100 Mbps ethernet
- ODRROID-X Node:
 - Exynos 4 SoC
 - Quad-core ARM Cortex-A9 @ 1.6 GHz
 - 1 GB RAM
 - eMMC storage (64 GB)
- Power consumption (cluster):
 - Idle: 8 W
 - Full load: 24 W
- Prototype has a fan

Architecture overview



RETE algorithm using CLIPS

- The Publish/Subscribe paradigm one of the most useful features of the Smart-M3 platform
- The current solutions have problems regarding scalability
- We have looked in to the use of the RETE-algorithm ¹ as a scalable solution
- We have implemented RDF triple publish/subscribe functionality in RedSib using the CLIPS ² rule engine

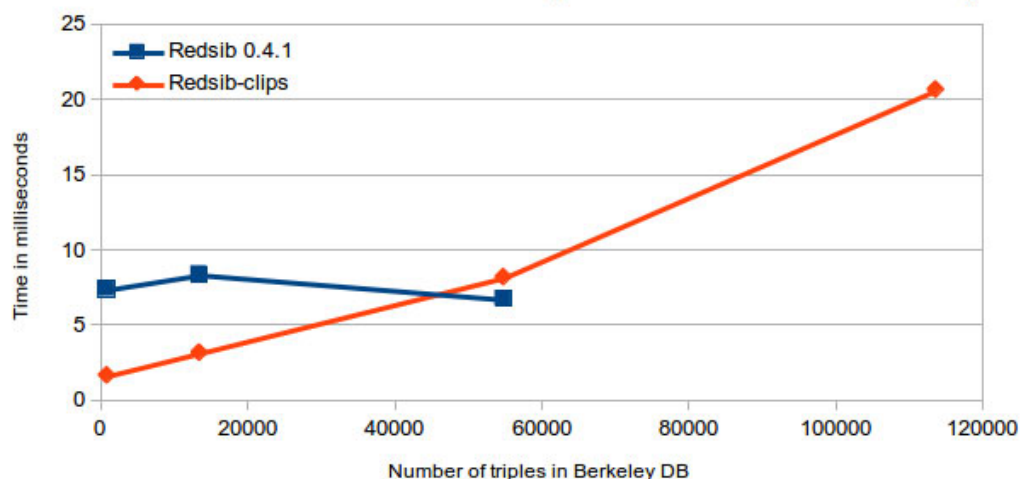
¹http://en.wikipedia.org/wiki/Rete_algorithm

²<http://en.wikipedia.org/wiki/CLIPS>

CLIPS implementation performance

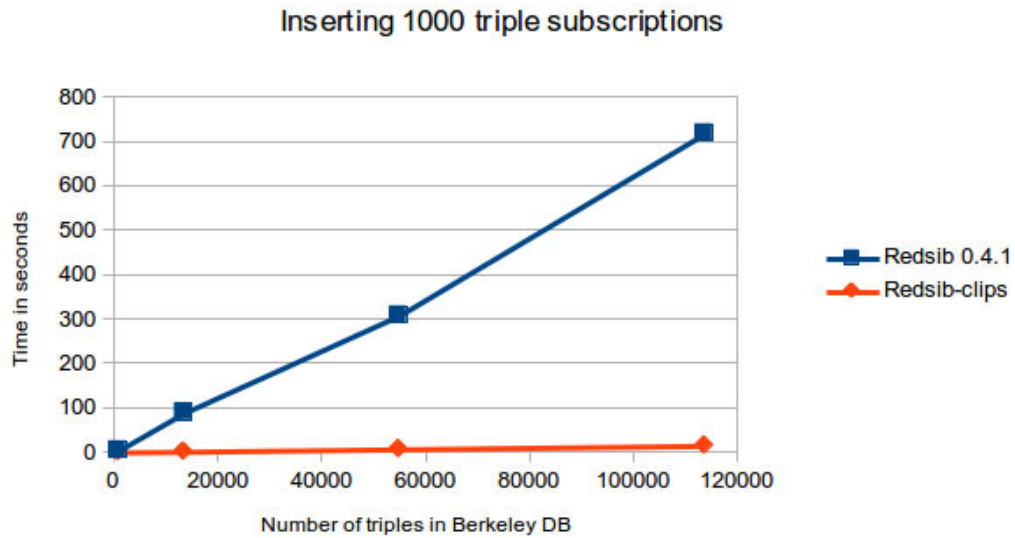
- Note: Very preliminary results
- Benchmarks performed on a desktop PC
 - Not yet tested on ARM platform, but relative performance vs RedSib 0.4 expected to be similar

Execution time before an activated subscription sends a notification message



CLIPS implementation performance

- Benchmarks performed on a desktop PC



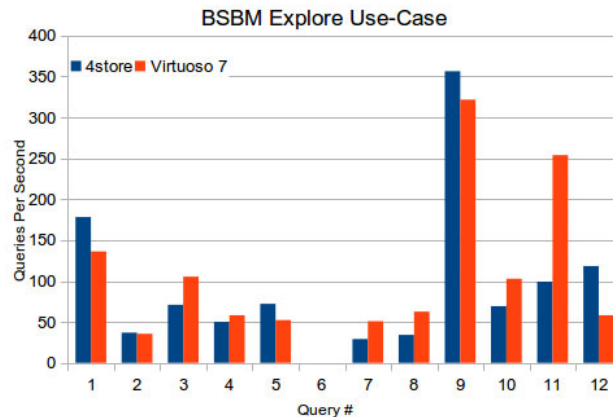
4store and Smart-M3

- We needed a distributed RDF-store to test our Smart-M3 box
- 4store is the only open-source, distributed, native RDF-store written in C
- We have written a librdf storage module for 4store, similar to the existing Virtuoso module



RDF-store Benchmarks

- Preliminary benchmarks using the BSBM³ benchmarking tool
- Virtuoso 7 vs 4store on a Laptop (queries only)
- Performance appears quite similar between the stores on average.



³<http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/>



RDF-store cluster benchmarks

- 1.6M triples: Performance similar for 1,2,4 nodes.
- Benchmark results at over 1.6M triples are still inconclusive.
- Some impressions:
 - Performance on one node seems to be dramatically reduced at 7M triples
 - 4 nodes at this number of triples still perform well.
 - Unstable results for one node already at 3.4M triples, while 2-4 nodes perform well.
 - Likely a consequence of the much larger amount of combined RAM in the whole cluster.
- DB also designed with 64-bit architectures in mind, while the ARM is 32-bit.





Context-aware Role-based Access Control

- Why?
 - Existing security model has only coarse-grained access control
 - We would like fine-grained control, down to the triple level
- We are planning an access control scheme using *access control ontologies* and *rules*
- Access control would be based on users' roles and context information
- Users may define their own rules for access to their data, which have higher priority than general rules defined by administrators.



Middleware

- We are developing a middleware between the Smart-M3 and users and sensors
- Modular design!
- The different parts of Smart Space will be accessible through a REST web framework
- The middleware manages Subscriptions in a **IF This Then That** ⁴ manner
 - IF **stepmeter value over 7000(This)** Then **post message on Twitter(That)**

⁴www.ifttt.com



Future Work

- Work still in progress on all parts of the platform.
- Further performance evaluation/benchmarking needs to be performed.



The End

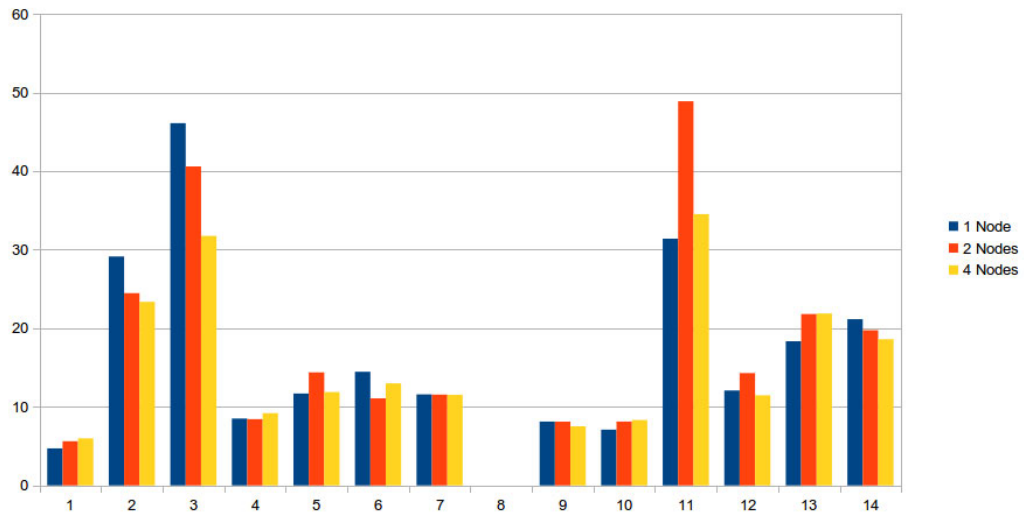
Thank you for listening!

Questions?



RDF-store cluster benchmarks

- 4-store cluster performance with 1.6M triples.
- No benefit from clustering at this number of triples.



KSP: SPARQL-like knowledge sharing for resource restricted devices

M3 Semantic Interoperability Workshop

Jussi Kiljander

VTT Technical Research Centre of Finland

Introduction

- Fundamental idea in M3 is to utilize semantic technologies to achieve interoperability in pervasive computing environments.
- Typical devices (e.g. sensors, actuators, etc.) and communication technologies (e.g. 6LowPAN, BLE, etc.) utilized in pervasive computing are resource restricted.
- On the other hand, semantic technologies use verbose syntaxes that are not suitable for resource restricted devices.
- Additionally, the M3 communication protocol (i.e. SSAP) is based on XML format that both requires a large amount of memory and is slow to process in low capacity computing platforms.

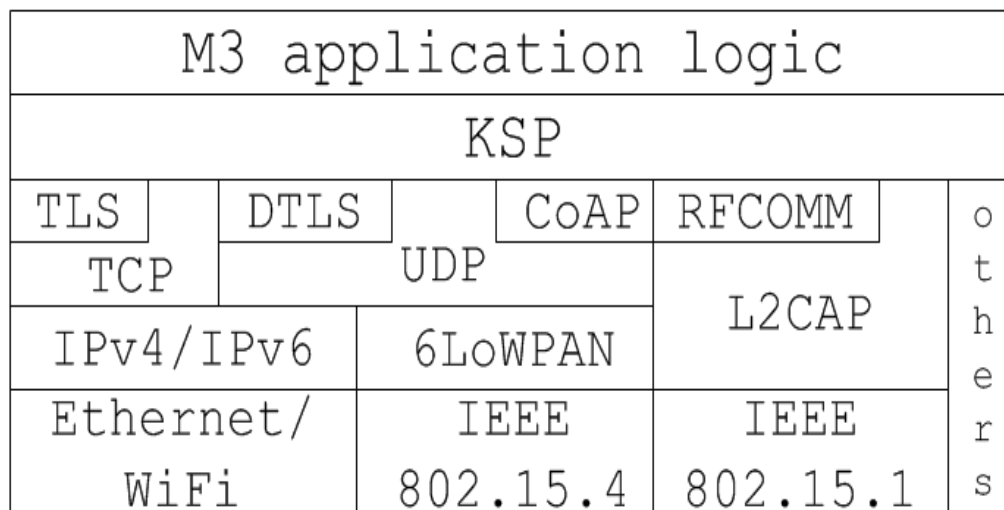
Introduction cont.

- In M3 systems this problem has been typically solved by utilizing gateways which transform proprietary format data from low capacity devices to semantic format.
- This approach **reduces interoperability** and **complicates the system** as for each new device a new gateway is needed (*i.e.* gateways are application/device specific).
- We propose a novel **Knowledge Sharing Protocol (KSP)** that enables semantic communication in low capacity devices and networks by providing:
 - **SPARQL-like mechanisms** for accessing and manipulating knowledge in smart spaces **in a compact binary format**
 - methods for **simplifying application logic** and **reducing traffic** in low capacity networks.

Differences between KSP and SSAP

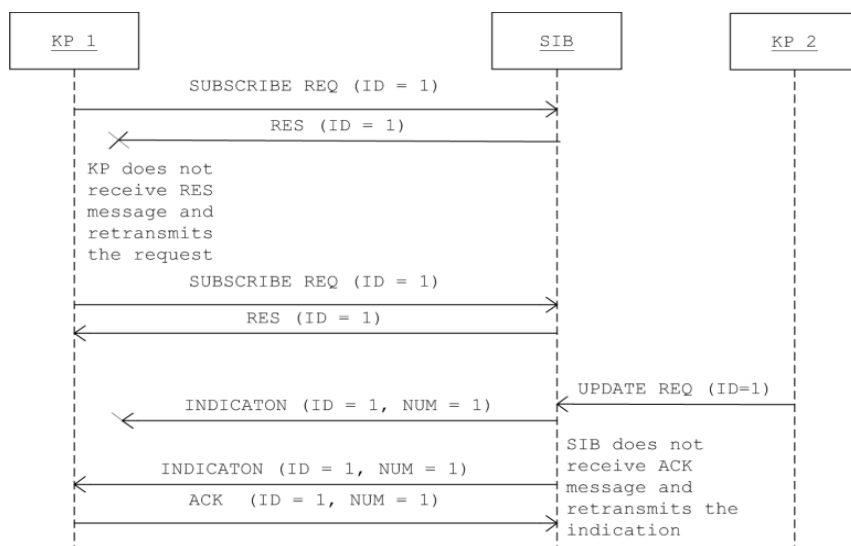
- 1) In KSP all operations are based on SPARQL 1.1.
- 2) Binary format for the messages instead of XML
- 3) No *join* and *leave* operations
- 4) In KSP it is possible to define the maximum size for SIB responses
- 5) KSP defines persistent format also for update operations

Knowledge Sharing Protocol stack



Messaging Model

- Message types: Non-confirmable (NON), and Confirmable (CON), response (RES), indication (IND), **acknowledgement (ACK)**.



Message format: Header

- Messages consist of three parts: *header*, *data*, and *options*.
- Fixed size *header* field contains parameters such as *version*, *transaction type*, *request type*, and *transaction identifier* that are common for all transactions.
- The structure of the *header* field depends on the message type (REQ, RES, IND, or ACK) and the transport technology (e.g. TCP, UDP, BLE, etc.).

Header format for TCP

Request:

0	8	40	47	48	63
Version	Length	Transaction type	Request type	Transaction ID	

Response/Indication:

0	8	40	47	48	63
Version	Length	Status code	Response type	Transaction ID	

Header format for UDP

Request:

0	8	15	16	31
Version	Transaction type	Request type	Transaction ID	

Ack:

0	8	16	31
Version		0x00	Transaction ID

Response:

0	8	15	16	31
Version	Status code	Response type	Transaction ID	

Indication:

0	8	15	16	32	48
Version	Status code	Response type	Transaction ID	Sequence number	

Transactions

Transaction type	Code
DELETE DATA	0x01
INSERT DATA	0x02
UPDATE DATA	0x03
DELETE	0x04
INSERT	0x05
UPDATE	0x06
SELECT	0x07
ASK	0x08
CONSTRUCT	0x09

Transaction type	Code
DELETE_PERSISTENT	0x0a
INSERT_PERSISTENT	0x0b
UPDATE_PERSISTENT	0x0c
SELECT_PERSISTENT	0x0d
ASK_PERSISTENT	0x0e
CONSTRUCT_PERSISTENT	0x0f
TERMINATE	0x10
RESET	0x11

Transaction type	Code
CREATE	0x12
DROP	0x13
COPY	0x14
MOVE	0x15

Message Format: Data Field

- In query and update messages the *header* field is followed by a *data* field which contains transaction specific information.
- In TERMINATE messages the *data* field is not needed.
- The structure of the *data* field depends on the operation type.
 - *E.g. SELECT, CONSTRUCT, ASK, DELETE, INSERT, etc.*

Data field format for query operations

SELECT REQ:

VC	Basic graph	OGC	[0-255]Optional graph
----	-------------	-----	-----------------------

SELECT RES/IND:

PC	[0-255]URI	VC	TRC	RC	[0-65535]Result
----	------------	----	-----	----	-----------------

[0-255] Variable

ASK REQ:

Basic graph	OGC	[0-255]Optional graph
-------------	-----	-----------------------

ASK RES/IND:

Ask result

CONSTRUCT REQ:

Construct graph	Basic graph	OGC	[0-255]Optional graph
-----------------	-------------	-----	-----------------------

CONSTRUCT RES/IND:

PC	[0-255]URI	TRC	RC	[0-65535]Triple
----	------------	-----	----	-----------------

Data field format for update operations

INSERT/DELETE DATA request:

Graph

UPDATE DATA request:

Delete graph	Insert graph
--------------	--------------

INSERT/DELETE request:

Graph	Basic graph	OGC	[0-255]Optional graph
-------	-------------	-----	-----------------------

UPDATE request:

Delete graph	Insert graph	Basic graph	OGC	[0-255]Optional graph
--------------	--------------	-------------	-----	-----------------------

CREATE/DROP request:

GC	[0-255]Graph name
----	-------------------

COPY/MOVE/ADD request:

Input graph name	Destination graph name
------------------	------------------------

Encoding Format for RDF graph

Graph field:

TC	[0-255]Triple
----	---------------

ST	PT	OT	Subject	Predicate	Object
----	----	----	---------	-----------	--------

- The *Graph* field consists of 8-bit triple count (*TC*) field and a zero or more (maximum 255) *Triple* fields.
- Each *Triple* field starts with 3-bit *ST*, 2-bit *PT*, and 3-bit *OT* fields, which specify the content of the following *Subject*, *Predicate*, and *Object* fields, respectively.

Type	Code
Empty	0x00
URI	0x01
Reserved Word	0x02
Variable	0x03
Literal	0x04

Field structure for RDF terms, variables and reserved words

URI field:

Prefix index	URI length	URI string
--------------	------------	------------

Literal field:

Literal type	Content
--------------	---------

Variable field:

Variable index

Reserved word field:

Code

Type	Value
xsd:string	0x00
xsd:interger	0x01
xsd:float	0x02
xsd:dateTime	0x03
xsd:Boolean	0x04

Reserved Words

Word	Value
rdf:type	0x00
rdfs:Class	0x01
rdfs:subClassOf	0x02
rdfs:property	0x03
rdfs:subPropertyOf	0x04
rdfs:range	0x05
rdfs:domain	0x06
owl:TransitiveProperty	0x07
owl:SameAs	0x08
xsd:string	0x09
xsd:interger	0x0a
xsd:float	0x0b
xsd:dateTime	0x0c
xsd:Boolean	0x0d

Message Format and Semantics: Options field

- One of the main advantages of XML based protocols is extendibility.
- In KSP options are a way to achieve a certain level of extendibility in a non-XML protocol.
- Options also enable to create more compact messages by leaving out parts that are not needed in the particular message.

Option	Code
PREFIX	0x00
DELETE GRAPH	0x01
INSERT GRAPH	0x02
QUERY GRAPH	0x03
FILTER	0x04
SOLUTION MODIFIER	0x05
BIND	0x06
MAX RESPONSE SIZE	0x07
CREDENTIALS	0x08

Options field: encoding

Prefix field:

PC	[0-255]	URI
----	---------	-----

Delete/Insert/Query graph field:

GC	[0-255]	Graph name
----	---------	------------

Filters field:

FC	[0-255]	Filter
----	---------	--------

TC	[0-255]	Token
----	---------	-------

Type	(Operand)
------	-----------

Binds field:

BC	[0-255]	Bind
----	---------	------

TC	[0-255]	Token	Variable index
----	---------	-------	----------------

Type	(Operand)
------	-----------

Credentials field:

KP ID	Password
-------	----------

Solution modifier field:

Order	Limit flag	Offset flag	Distinct
Unused	(Variable index)	(Limit)	(Offset)

Max response size field:

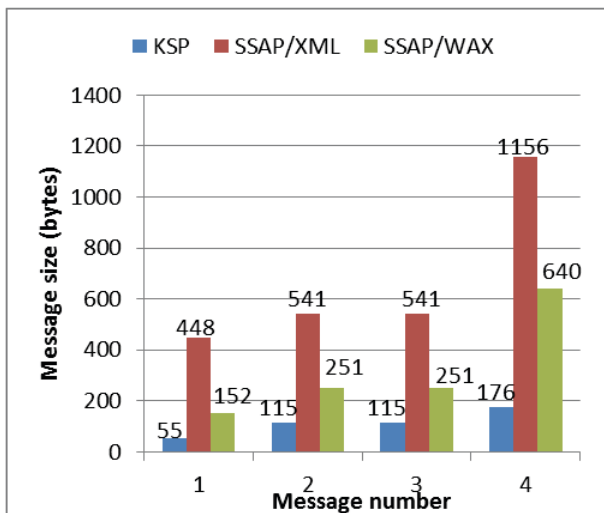
Size

Evaluation

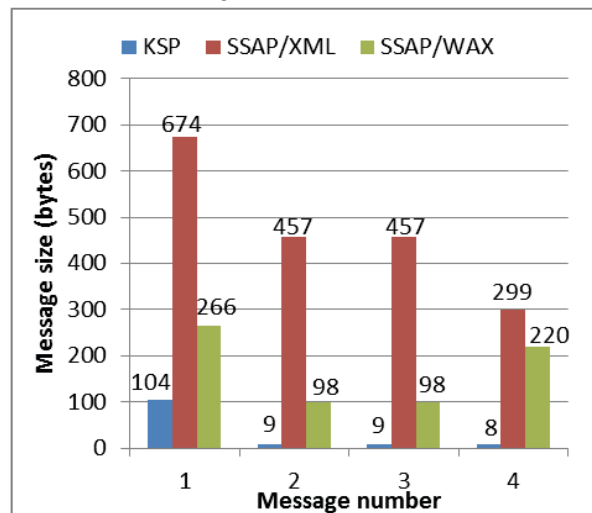
- Comparison of message sizes of KSP, SSAP/XML, and SSAP/WAX protocols in Smart Greenhouse demonstration.
- The KSP messages were on average 87.08% and 70.09% shorter than the SSAP/XML and SSAP/WAX messages respectively.
- We also demonstrated how a KP implementation can be significantly simplified with persistent update operations.

Evaluation

requests



responses



Drawbacks and Limitations

- Binary format limits both the amount and the size of entities such as *prefixes, graphs, triples* and *results*.
- KSP requires a good application programming interface (API) because the binary format is not suitable to be used by developers as such.
- Some of the SPARQL 1.1 functionalities are not supported by the KSP because they would have made the KSP too complicated.
 - E.g. the current version of the KSP does not support DESCRIBE queries, Property paths, Aggregates and Subqueries, and many SPARQL functions.

Conclusions and Ideas for Future work

- By providing more **compact messages** and operations that **simplify the application logic** the KSP is more suitable for low capacity devices and networks than the official SSAP.
- The KSP has certain **limitations** compared to standard SPARQL 1.1 that might restrict its use in certain situations.
- The KSP can be extended, but there is **also a need for M3 protocol that supports the official SPARQL 1.1** as such.
- In the future, we should decide **what do with the SSAP**.
 - replace with HTTP/SPARQL or CoAP/SPARQL?
 - design SSAP v2.0?



Thank You!



A portable implementation of Semantic Information Broker in OSGi technology



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Alfredo D'Elia

ARCES

DISI

adelia@arces.unibo.it

www.unibo.it

Innovation House – Otaniemi
12th November 2013

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



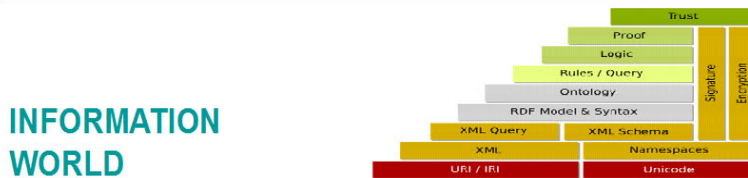
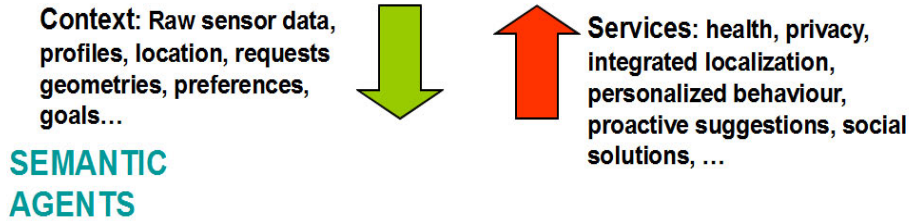
Overview

- Reference scenario
- Motivations
- Features
- Internal architecture
- Performance evaluation
- Conclusion

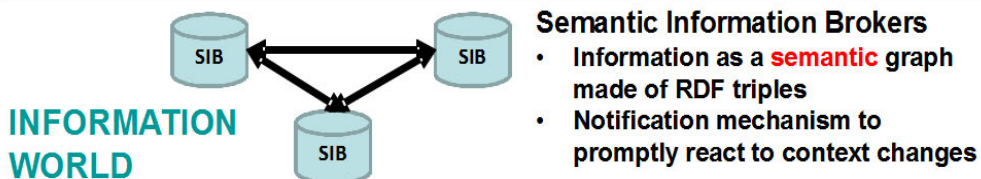
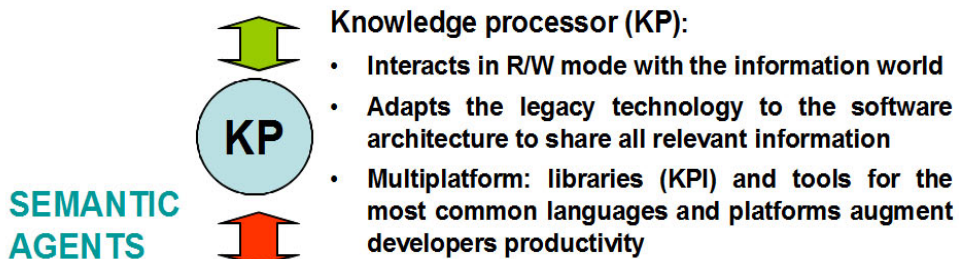
ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



Reference scenario



Reference scenario





Motivations

- Meet the requirements of evolving scenarios
- Portability
- Compatibility
- Flexibility
- Meet the technological expectations of dynamic communities (Semantic Web, Context awareness, etc.)

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



Combining M3 and OSGi

- Smart M3:
 - General purpose
 - Information interoperability support
 - Semantic software agents
- OSGi
 - Java
 - Portable
 - Modular
 - Maintainable
 - Reliable
- The OSGi SIB provides the advantages of both smart-M3 and OSGi in a single core component for semantic context aware applications.
- The OSGi SIB is more appropriate than a monolithic implementation as long as the scenario and the requirements are in rapid evolution.

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



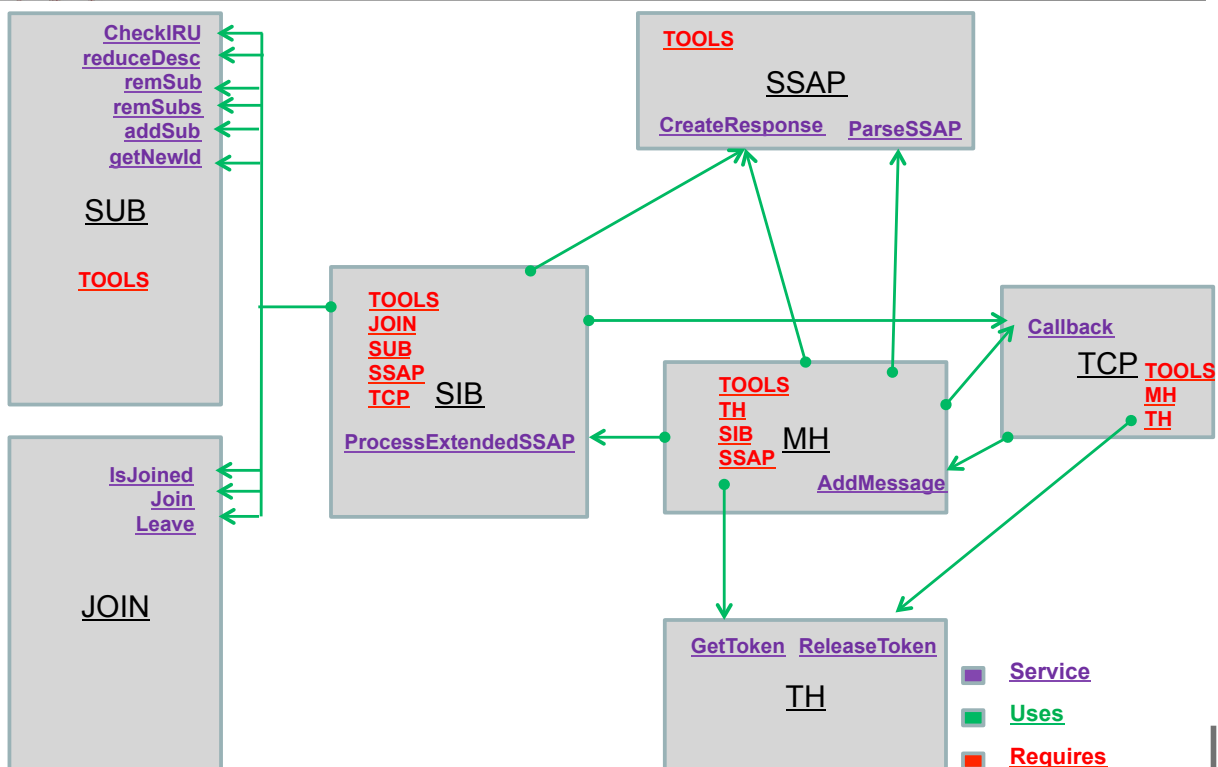
Features

- All basic SIB primitives
- Completely compatible with the current version of SSAP and existing libraries
- SPARQL query and update
- SPARQL subscription
- Persistent SPARQL UPDATE (similar to a rule) *new*
- DL reasoning at SROIQ level (using Pellet) *experimental prototype*

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



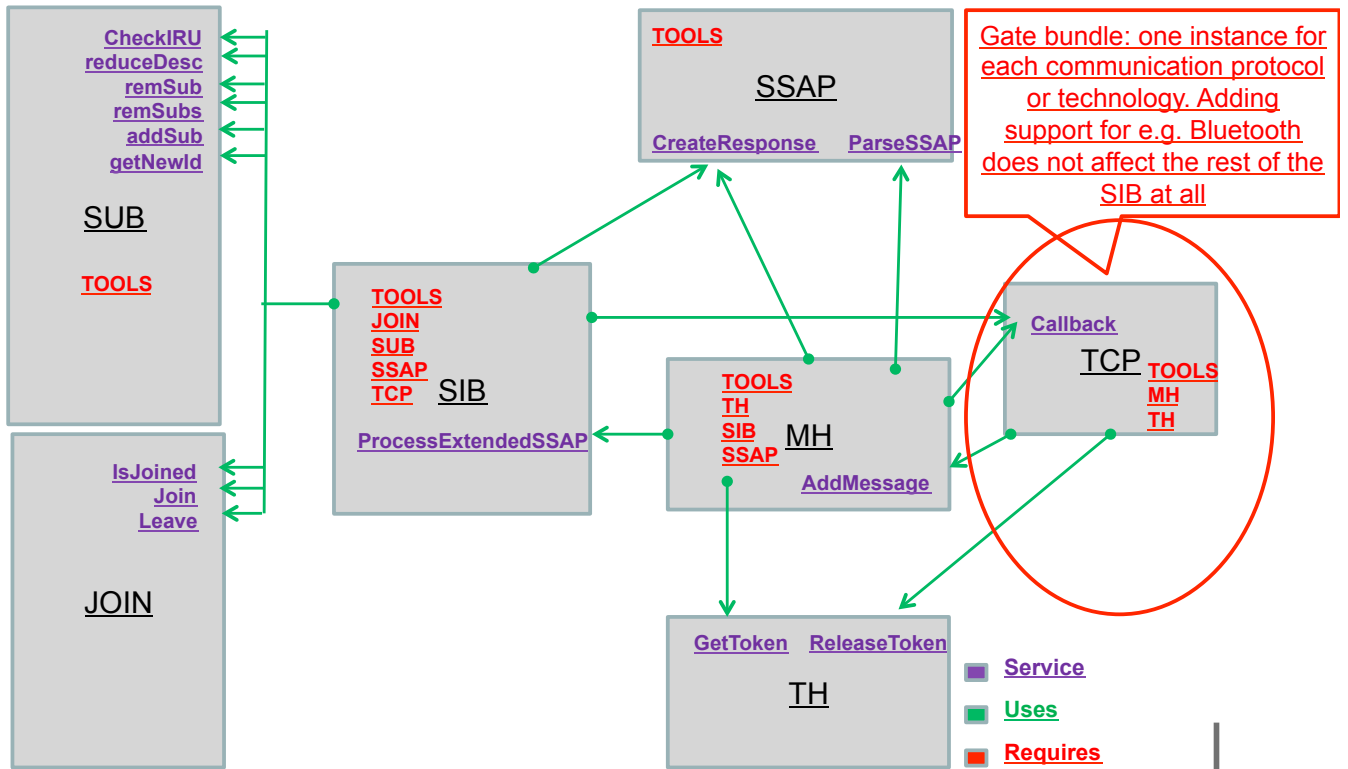
Internal architecture



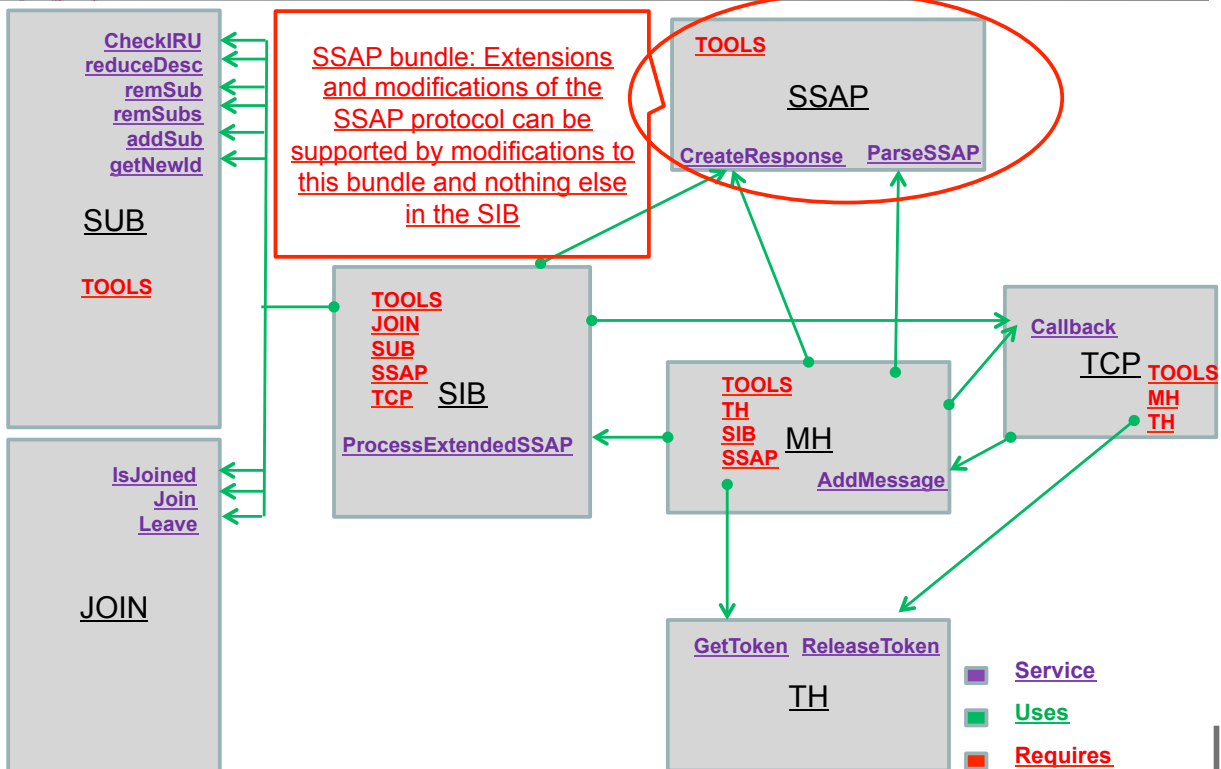
ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



Internal architecture

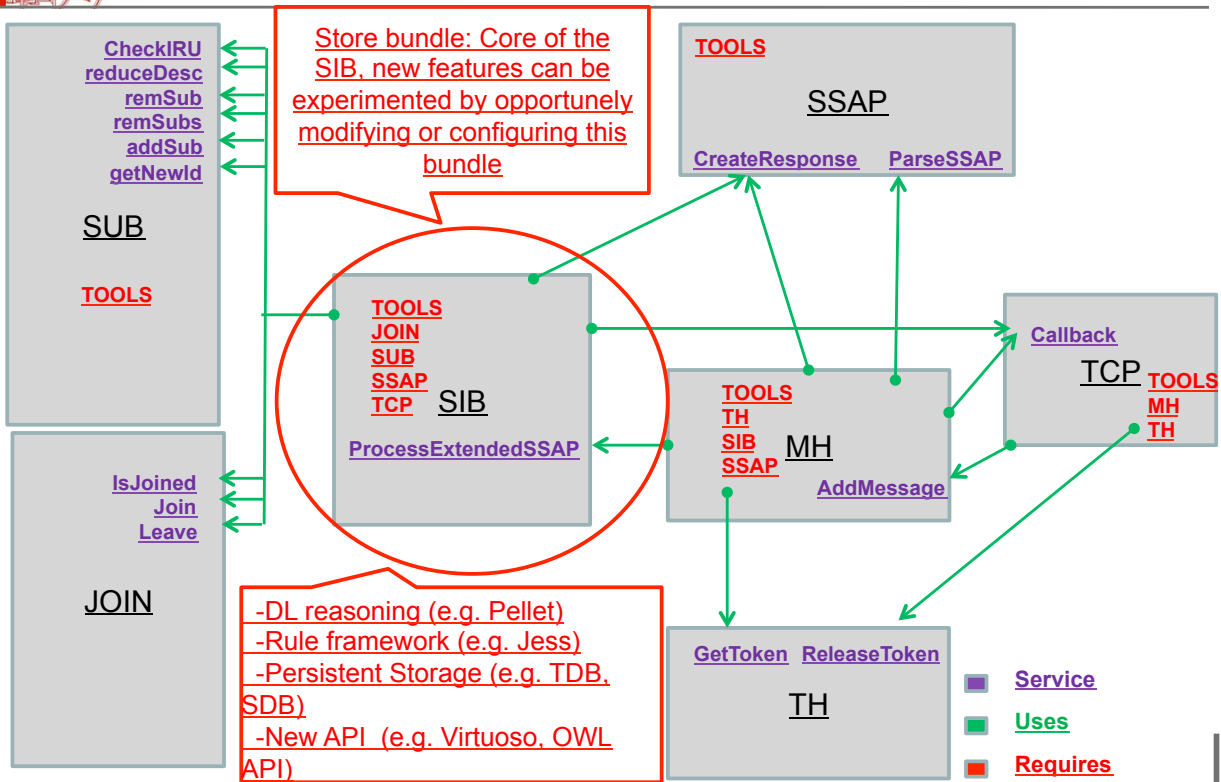


Internal architecture

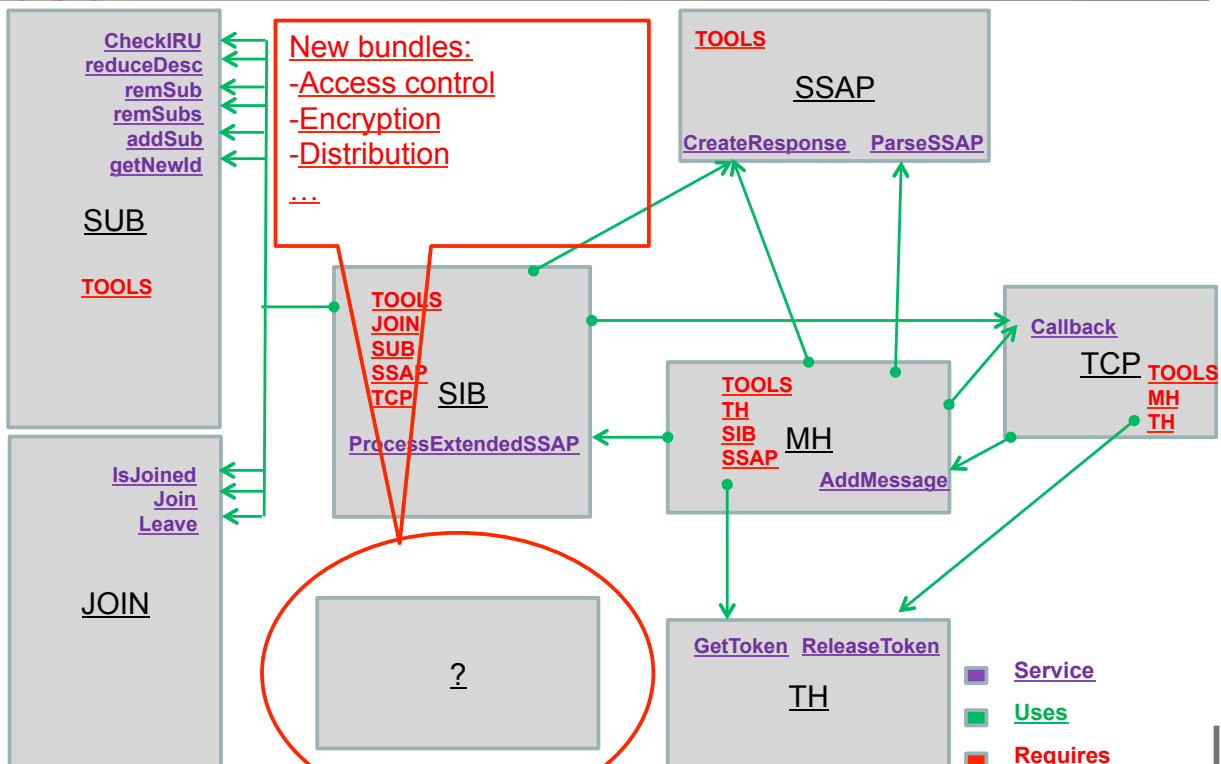




Internal architecture



Internal architecture





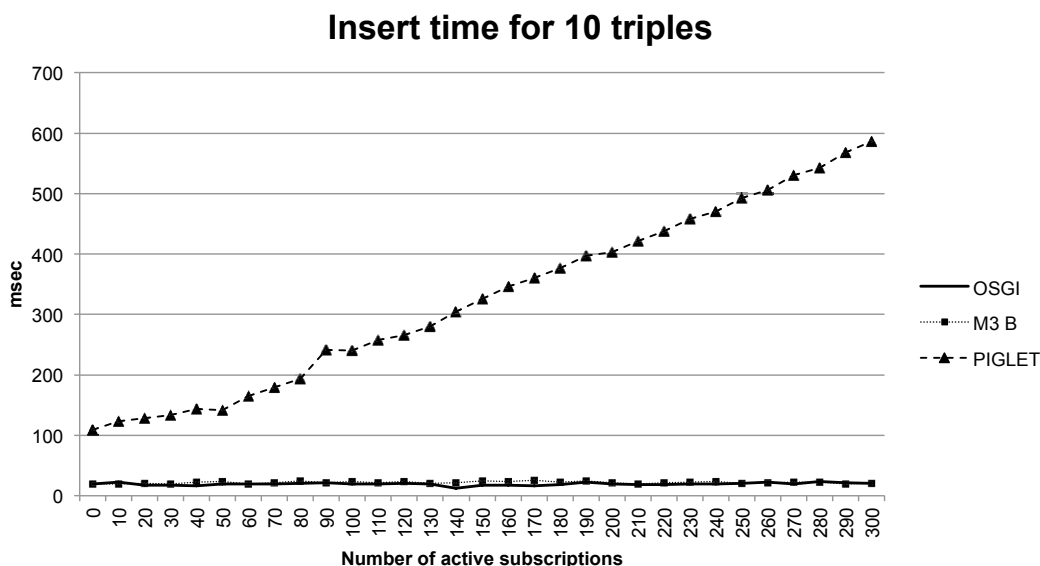
Performance evaluation

- Performance of primitives supported by the internal RDF store match in optimal conditions the performance of the store itself (i.e. Jena Framework for RDF graphs)
- Interesting topic is to study SIB specific functionalities (i.e. not originally supported by the internal store) and how they affect the other primitives

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



Performance with respect to subscription patterns



ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA



Conclusion

- New SIB implementation running on most popular operating systems
- New features implemented and tested in prototype versions
- Performance comparable to the reference implementation which is more optimized, but based on a monolithic approach
- Simple adaptation to possible future requirements among which:
 - new SSAP version
 - new primitives
 - security



SMOOL

The Smart Space Modeling, Access and Deployment framework

Adrian Noguero
(adrian.noguero@tecnalia.com)

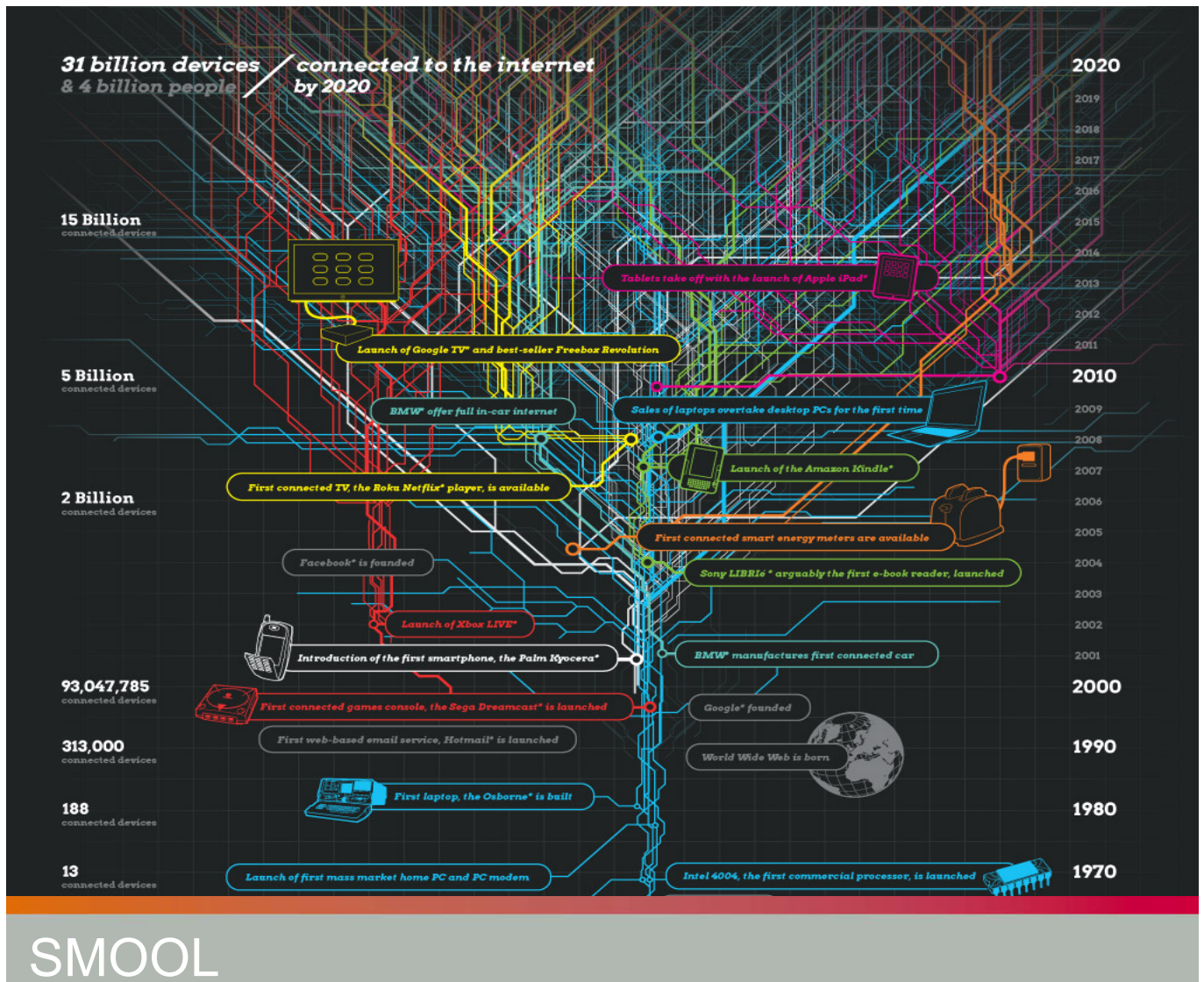


What's a SMART SPACE?

A smart space is *an environment* in which *devices interact* with each other to provide the users *added value services*.



SmartCityJaén
UNIVERSIDAD DE JAÉN



SMOOL

What are the main objectives of SMOOL?

- Manage **heterogeneity** among devices
- Provide a **Toolkit** to **design, validate and deploy** smart spaces
- **Flexibility** vs. ad-hoc solution
- Access through a **single application** to the smart space.

Infrastructure

- **SMOOL** is based on the well-known Smart M3 middleware. Branch of the Sofia Community.
 - Publish/Subscribe paradigm
 - Sematic Information Broker (SIB)
 - KP libraries (J2SE, Android, C++)
- Additional features developed:
 - **REST interface** for the SIB
 - **Persistency** manager
 - **Authorization** & user management

REST Interface

- The REST interface enables **read-only access** to (some) information stored in the SIB via HTTP
 - Responses are **JSON strings** with the triples
 - Queries are supported
- The main goal of the REST interface is to become a **gateway between hierarchically organized SIBs**
 - Local smart spaces can share data with smart spaces at a higher level

Persistency

- The SIB includes a **MySQL client** that connects to a database on start-up
 - Store persistent data
 - **Load and publish persistent data** to connecting KPs **on start-up**
- Persistency enables the **smart personality** concept in the smart space
 - The smart space stores the **preferences of the users** and shares them with the applications

Authorization and Users

- **SMOOL** includes **access control** with users, groups and permissions
 - Users and groups may define permissions to access produced data to other users/groups
 - Subscriptions may be linked to a user/group
 - “Guest” user may access limited data in the smart space
 - Communications on top of SSL
 - Users and permissions are only modified at design time

Development of KPs

- **SMOOL** has extended the SOFIA ADK:
 - A **single ontology** is used:
 - **Extensible**
 - Managed by the SIB (ensuring **backward compatibility**)
 - Improved KP generation wizard, including example application code:
 - Generation of **Model, SSAP and Connector layers**
 - **Direct compilation** to a shared library
 - **Simplified KP API:**
 - Easier discovery and connection control
 - Model layer helpers
 - **Unified API** for all supported languages



Modeling and Deployment Tool

- **SMOOL** provides an easy to use editor for non-expert users:
 - Model the devices connected to the smart space
 - Easily create smart applications
 - Define UI for smart-phone/tablet access
- The editor is currently under development, but...
 - ...let's [see how it'll look like!](#)



Conclusions

- **SMOOL** aims at becoming a **full-fledged framework** for smart spaces design and development.
- It is **based on the M3 middleware**, with some add-ons required for the implementation of the smart spaces concept:
 - REST Interface. Persistency and Authorization
- **Expert users** may create their applications with the **SMOOL SDK**, which extends the SOFIA ADK
- **Non-experts** will be able to create simple smart spaces without coding
- The **SMOOL Android application** will be released along with the smart spaces editor

THANKS

For more information visit:

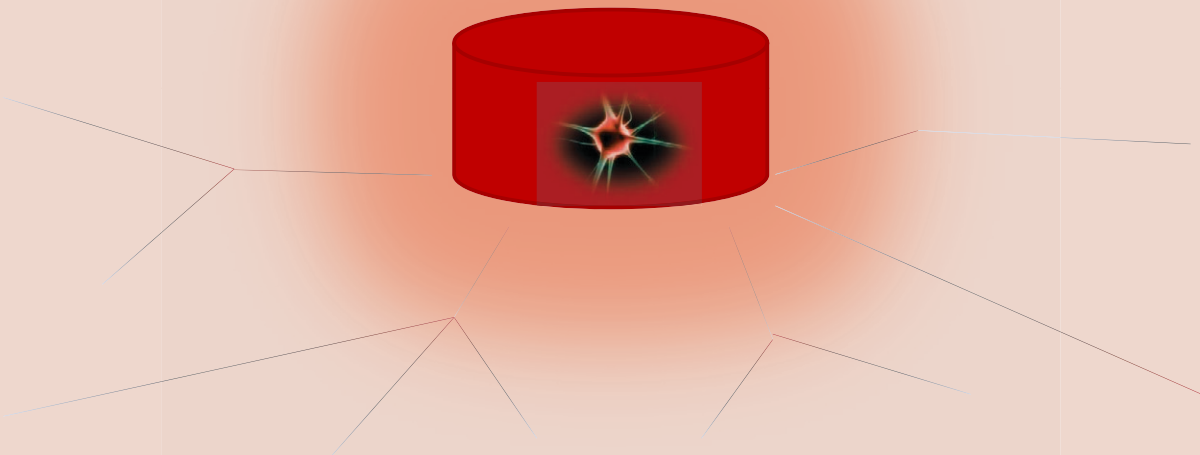
<http://code.google.com/p/smool/>

www.tecnalia.com

tecnalia  Inspiring
Business

The Semantic Event Broker

Francesco Morandi



1

Vision: what future for Smart-M3?

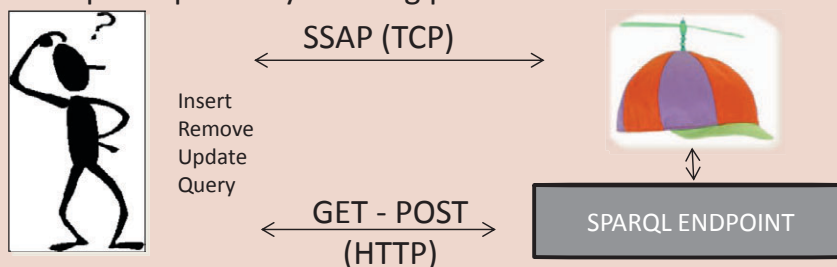
Is it possible to consider current Smart-M3 releases a «triple-store» or an «endpoint» alternative?

- Modern SPARQL Endpoints (like Jena, Sesame, Virtuoso...) have:
 - > Improved distributed database management
 - > Multithread architecture
 - > Big communities + Years of experience (in some cases almost 20 years)

... the only «smart» answer is no.

Smart M3 should be a **layer** over existing triple-stores or endpoints and provide:

- new **capabilities** for Smart Environments (or CPS) , e.g. new algorithms for adding new functionalities and for performance improvement
- a «**smart hat**» on top of a perfectly working platform



2

SOFIA project and our vision (1/2)

- The SIB is a dynamic real time context handler. We consider the Smart Environment and the SIB related by a bijective function between entities. So the SIB has to store a **“limited” amount of triples** coming from the physical world.
- The RDF content **changes continuously** with relatively **“small” graph updates** (an “update” is an insert, remove or insert-remove operation) because:
 - Sensors produce periodical updates of a limited number of parameters (e.g. temperature, pollution, hearth rate, weight, pressure...)
 - Aggregators usually publish little information at a time as result of some processing (maybe just some instructions to trigger actuators).
 - Actuators, in turn, usually react to notifications and may also update the SIB with some feedback.

3

SOFIA project and our vision (2/2)

- The SIB context information store is structured to retrieve real time data starting from the ontology; the SIB should not store massive historical information.
- Every history service should be implemented externally (KPs based) to avoid a negative impact on performance.
- The SIB should overcome **synchronization issues** between KPs that share multiple access to the same resources.
- One of the most powerful Smart-M3 specificity is its **publish-subscribe** mechanism: any KP can be automatically woken up (with no extra-overhead) whenever the sub-graph of interest changes; each sub-graph of interest could be the composition (based on filtering or logical operations) of sub-graphs.

4

SIB progress

- **SPARQL 1.1** (query and update) is now the **official** W3C language for RDF. All **other methods** for querying, subscribing and updating the SIB (RDF operations, other query methods...) can be considered **obsolete** or specific SPARQL sub-cases.
- **SPARQL applied to publish subscribe** can become extremely powerful, with a behavior similar to rule engines or **event processing languages** but with all **semantics related benefits**. An algorithm has been implemented to improve performance and exploit parallel execution.
- **A Time Management** mechanism is required to support synchronization and watchdog functionalities (the latter is quite cumbersome to implement in Smart-M3).

5

Towards the Semantic Event Broker: programming model (1/4)

The reference paradigm: Event Processing Languages.

E.g. TESLA* event based language (Running on powerful parallel HW, like CUDA):
Events are more “rich” due to the time integration

```
define Fire (Val)
from Smoke () and
each Temp(Val > 45) within 5 min from Smoke
where
Val = Temp.Val
```

Is it possible to achieve the same behavior with the added value of the semantic ?
(e.g. C-SPARQL, EP-SPARQL, ...)

It is extremely **difficult** to implement a similar rule in an ordinary SMART-M3 KP.

Implementing a KP with this type of behavior can be complex, as we need:

- > Threads with Sleeps , a counter thread with reset.
- > Synchronizations

**THE KP MODEL
SHOULD BE SIMPLER
AND BETTER DEFINED**

6

Towards the Semantic Event Broker: programming model (2/4)

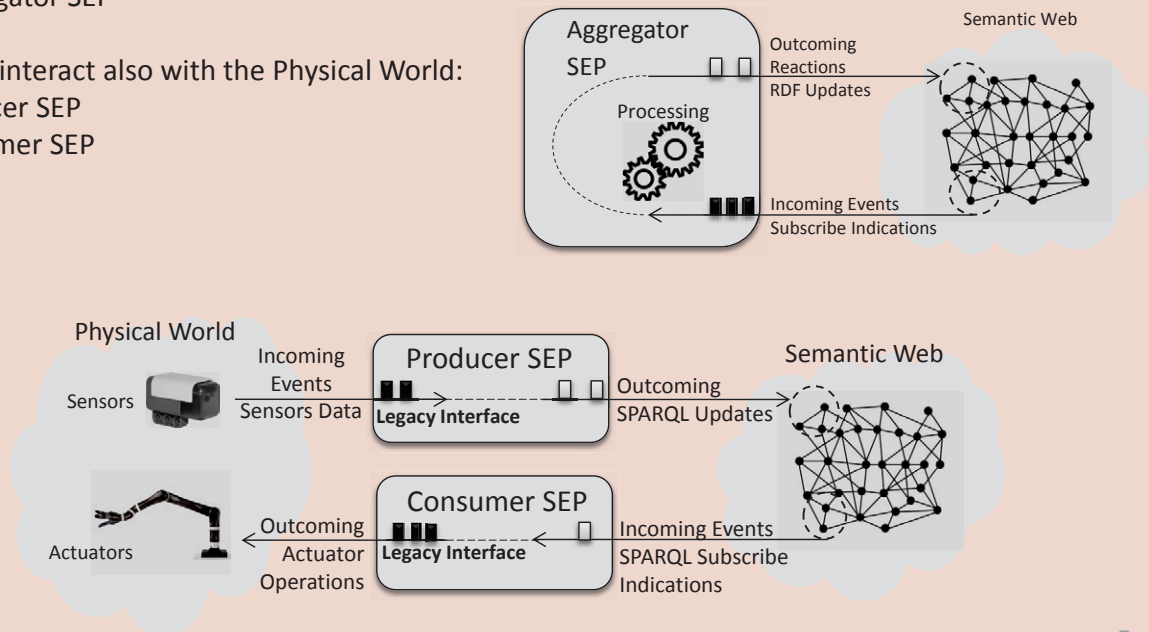
We can imagine to introduce some specialized types of KP, named Semantic Event Processors (SEP). SEPS are event-oriented and time-based; they are classified as follows.

A SEP can interact only with the semantic broker:

- ❖ Aggregator SEP

A SEP can interact also with the Physical World:

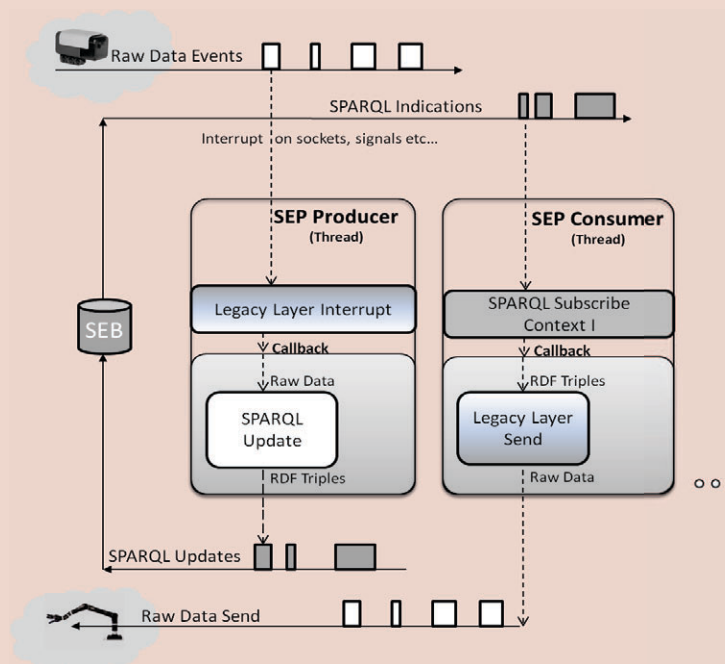
- ❖ Producer SEP
- ❖ Consumer SEP



7

Towards the Semantic Event Broker: programming model (3/4)

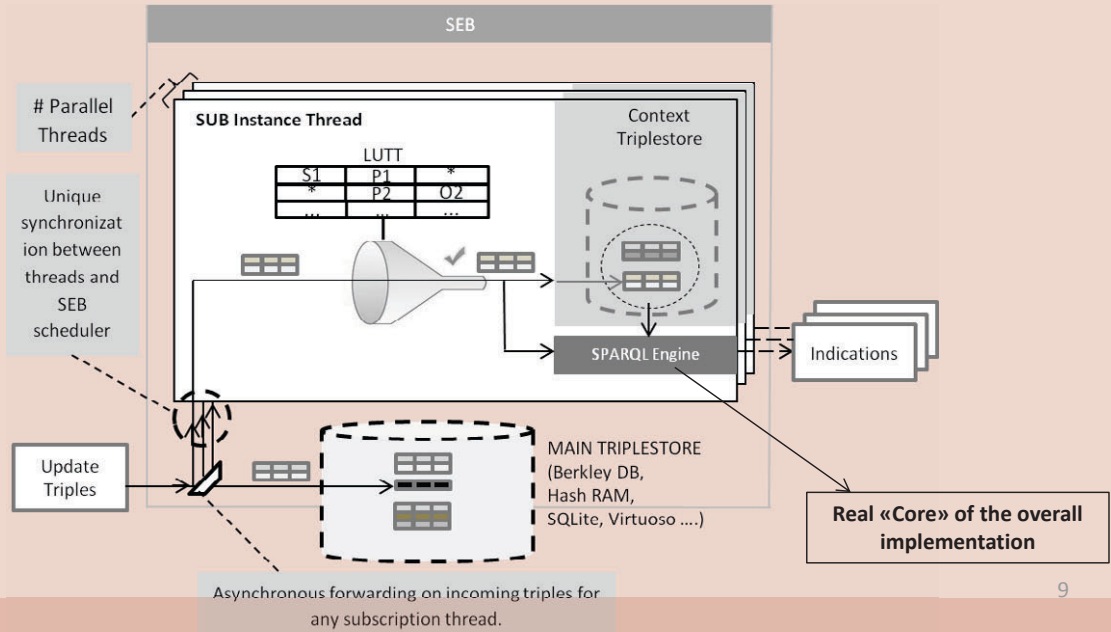
- Every Producer SEP is provided with a Legacy Layer (able to read data from the physical world) and performs SPARQL Updates on changes.
- Consumer KP holds a SPARQL Subscribe and sends data to the legacy layer at every notification.



8

Towards the Semantic Event Broker: The Internal Architecture

- Multithread for any active subscribe
- All the update operations are forwarded in to the threads through async queues
- Every subscription thread is provided by
 - o LUTT
 - o Internal Sub-Triplestore
 - o Sparql Accelerator * (or Sparql Engine)
 - o Dispatcher for Indications



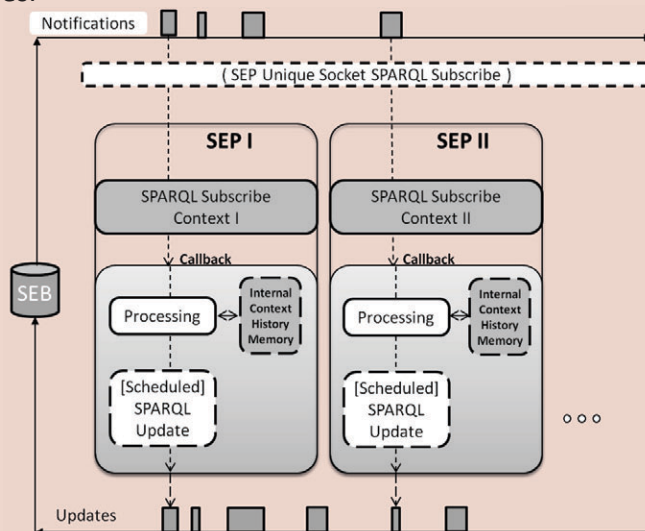
9

Towards the Semantic Event Broker: programming model (4/4)

Every aggregator is based on :

- Context Subscribe (Sparql Subscribe)
- Callback
 - o Processing (Combinatorial, Causal) [Optional]
 - o Sparql Update (Scheduled[Optional])

And on time functionalities.



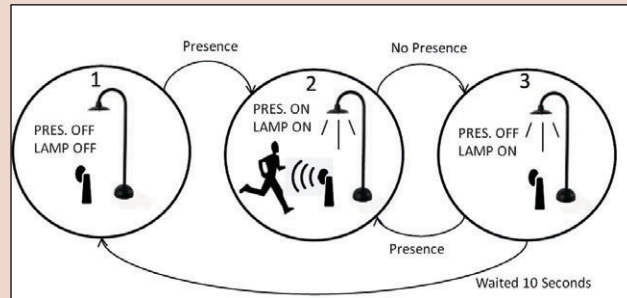
10

Towards the Semantic Event Broker: an example

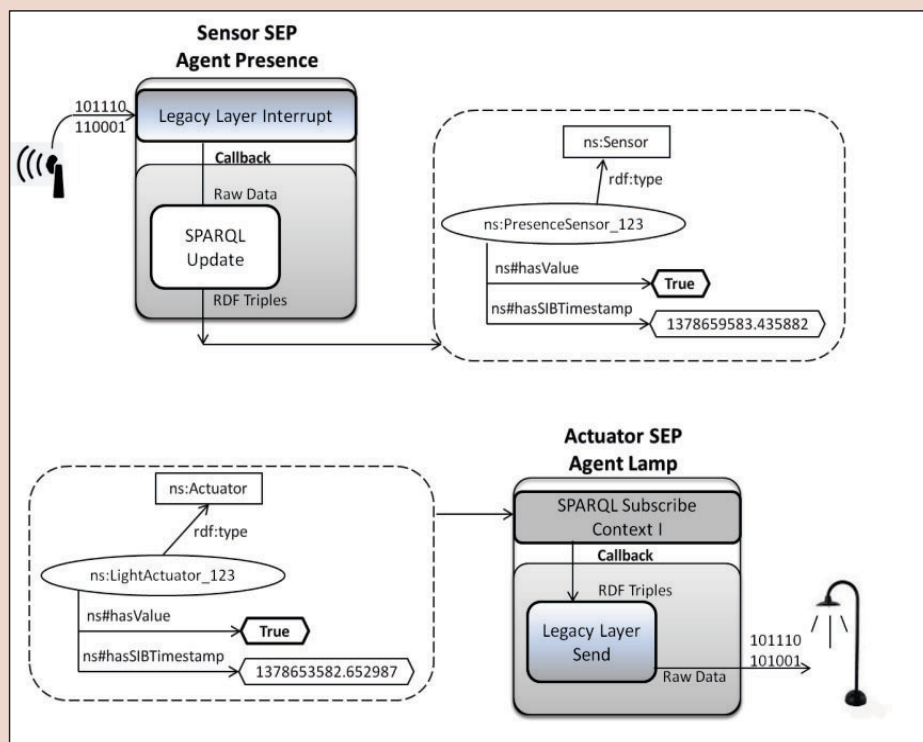
The scenario includes a presence sensor and a lamp.
 (the example is almost identical to the fire alarm of the events)

Behavior:

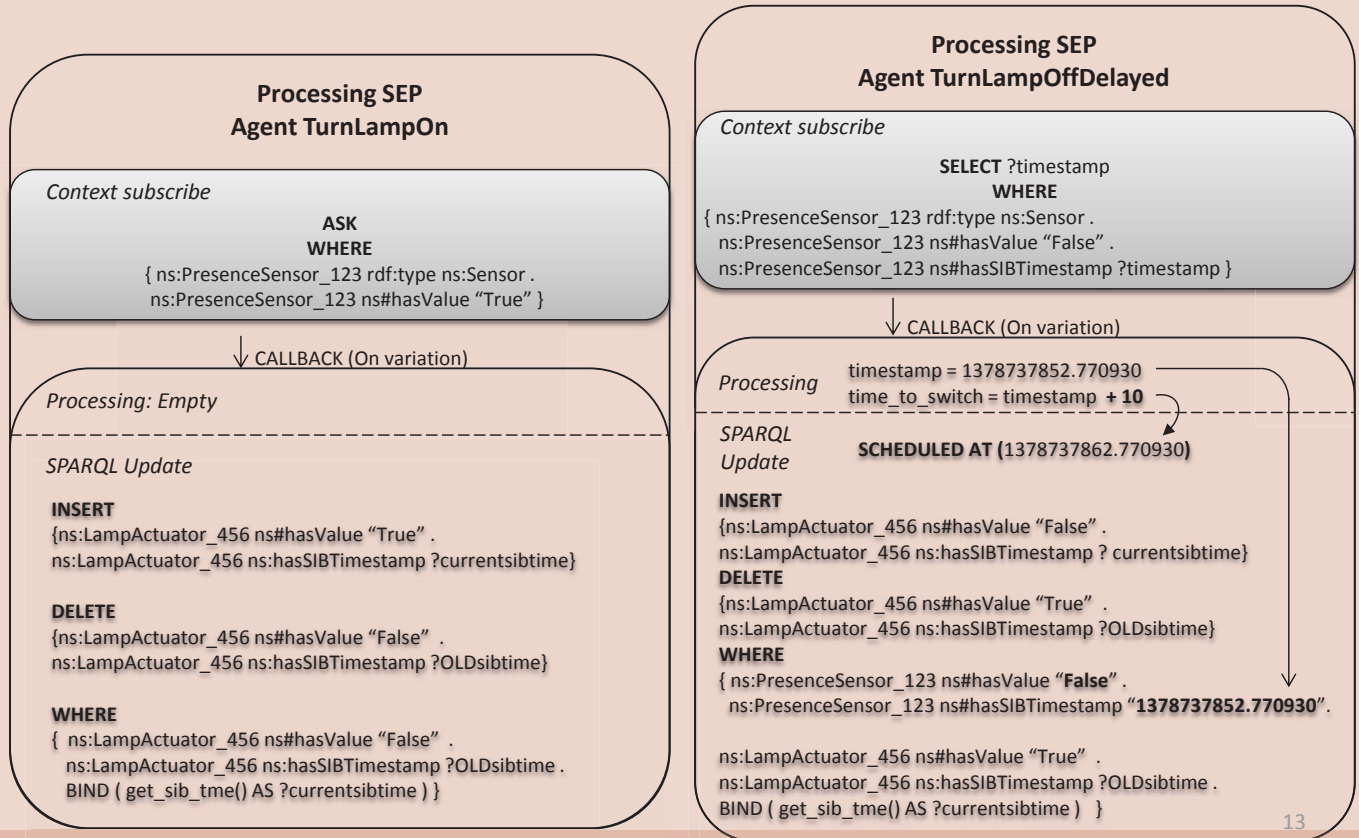
- If a presence is detected the lamp must be turned on.
- When the presence is no more detected for 10 seconds the lamp must be switched off.



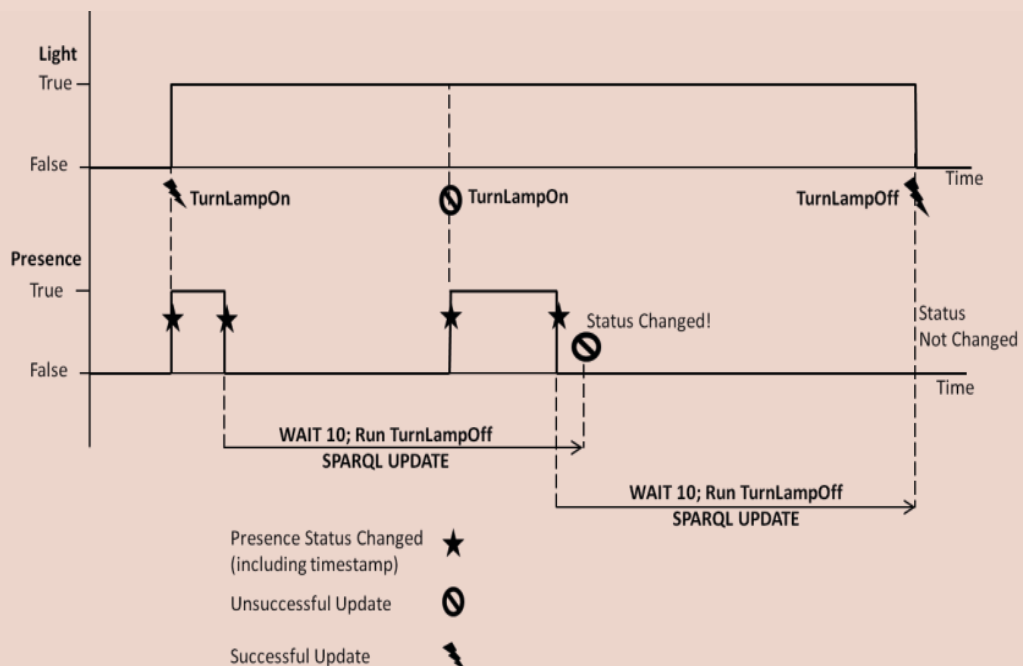
Towards the Semantic Event Broker producer and consumer



Towards the Semantic Event Broker aggregator



Towards the Semantic Event Broker event timing management



Conclusions

As simple triple-store, Smart-M3 can be considered inadequate compared to most SPARQL Endpoints.

If we agree to consider Smart M3 a *layer* on top of existing TS, the aim can be to converge to the semantic events domain thanks to:

- A KP design style based on two primitives (SPARQL SUBSCRIBE and SPARQL UPDATE) and time management functionalities (e.g. queries will become obsolete)
- A SIB core architecture with:
 - A fast algorithm for SPARQL subscriptions processing
 - Support for parallel processing
 - Time management support

Performance Evaluation of Operations in RedSIB with Substitution Mechanism

Andrey Vasilev

Yaroslavl FRUCT Laboratory, Yaroslavl State University



Andrey Vasilev

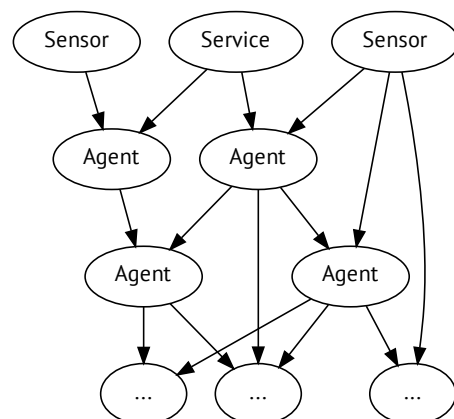
Performance Evaluation of Operations in RedSIB with Substitution M

Dataflow Network Model

- IoT is large set of interconnected elements
- The base are data coming from smart devices and sensors

Dataflow network

- Multi-step information refinement
- Well-suited for sensor data processing



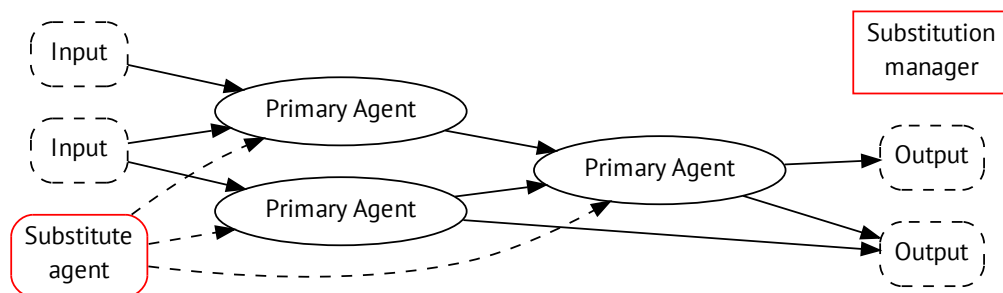
Andrey Vasilev

Performance Evaluation of Operations in RedSIB with Substitution M

Agent Substitution Mechanism

- Many units of IoT are mobile devices
 - have a limited power supply
 - use unreliable wireless channels
- Processing unit may unexpectedly disconnect
 - break existing data flows
 - loose accumulated context information

Agent substitution idea



Agent Substitution Mechanism Operations

- Keeping primary and substitute agents registry
 - Agent registration
 - Agent unregistration
- Agent substitution procedure
 - TCP/IP connection failure
 - Explicit substitution request
- Primary agent return
- Store subscription notifications during the substitution



Agent Substitution Implementation Aspects

Substitution mechanism is implemented as a SIB module

- Must detect disconnection as fast as possible
- Must efficiently manage agent context

Mechanism uses special configuration ontology

- Any KPI can be used to create dataflow agent
- Uses fixed number of internal subscriptions to track data modifications

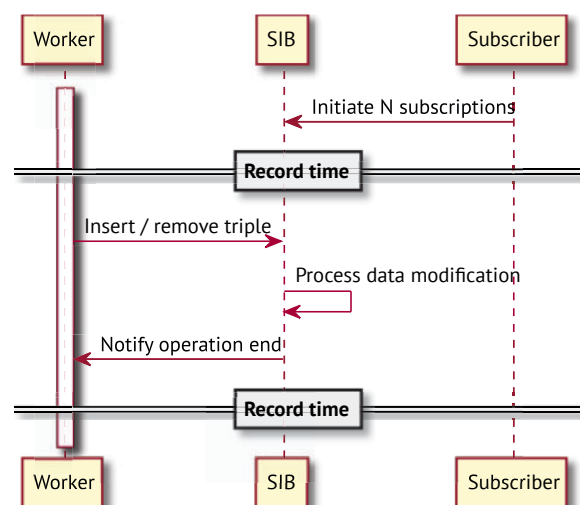


Core SIB Operations Testing

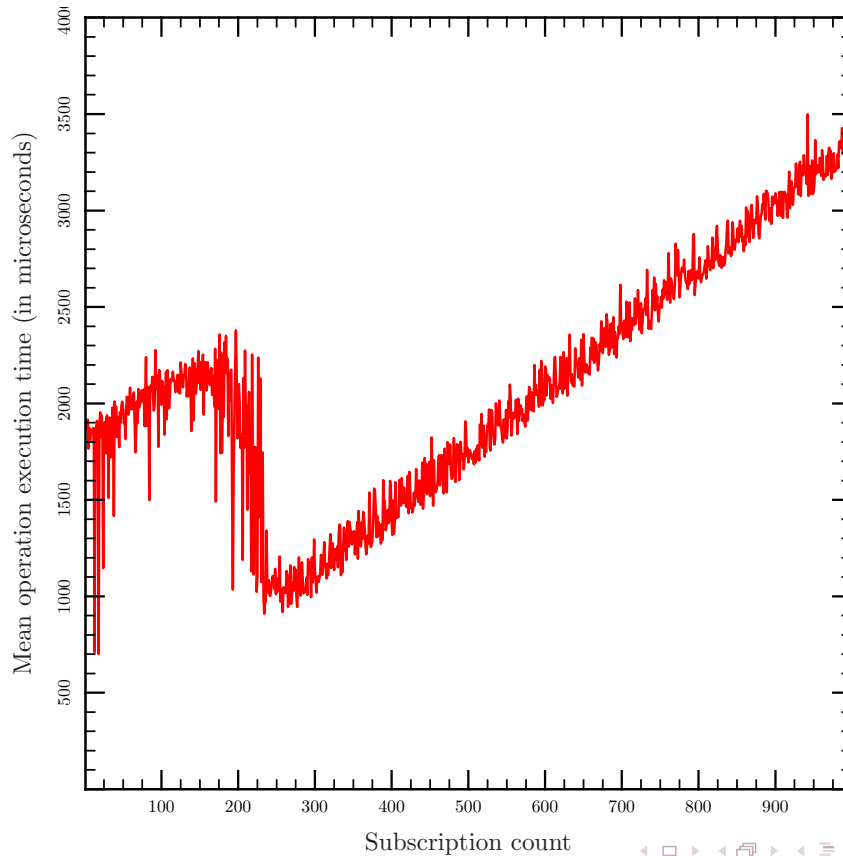
Experiment goal: estimate the mean time of insert and remove operations substitution

Perform 1000 following test cycles, where N is the current cycle number

- Setup N subscriptions to the smart space
- Insert and remove random 50 times (triples do not trigger subscriptions)



Core SIB Operations Testing Results



Andrey Vasilev

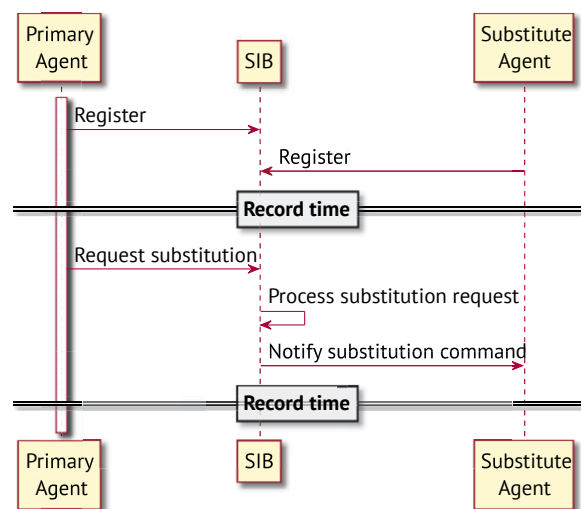
Performance Evaluation of Operations in RedSIB with Substitution M

Substitution Performance Testing

Experiment goal: estimate the mean time of substitution operation execution

Perform 35 following test cycles, where N is the current cycle number

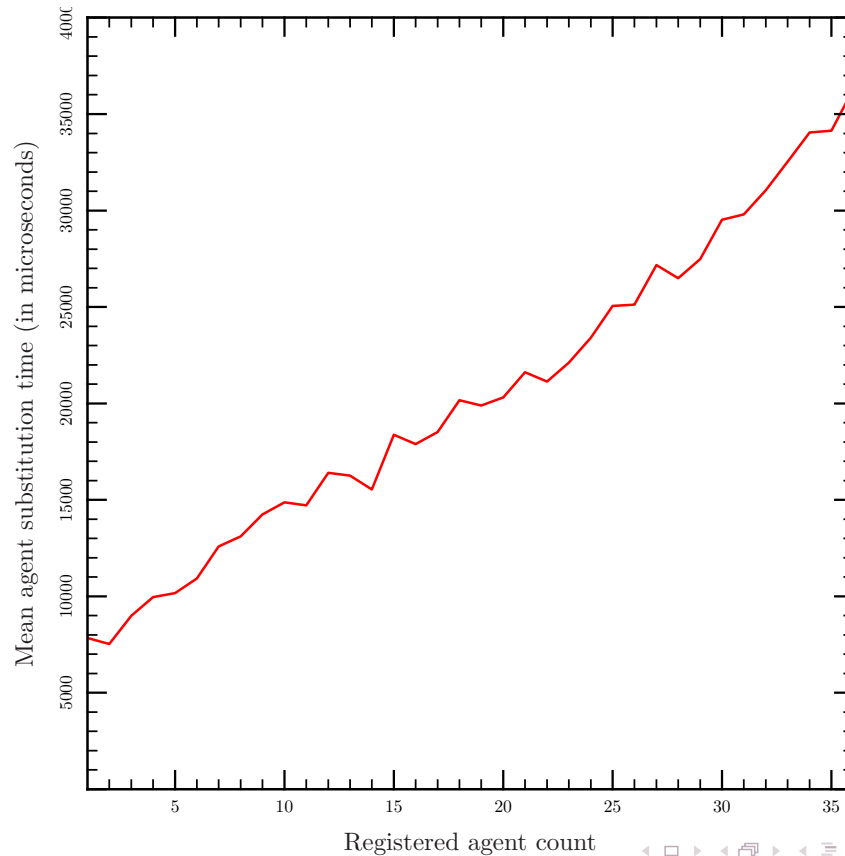
- Register N primary agents and N substitute agents
- Perform voluntary substitution request for primary agents



Andrey Vasilev

Performance Evaluation of Operations in RedSIB with Substitution M

Substitution Performance Testing Results



Andrey Vasilev

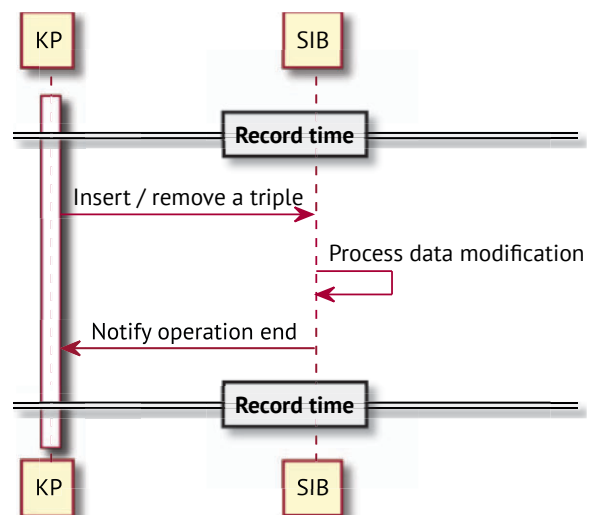
Performance Evaluation of Operations in RedSIB with Substitution M

Impact of Substitution on SIB Operations

Experiment goal: estimate the mean time of core operations execution during the substitution operation execution

Perform 900 following test cycles, where N is the current cycle number

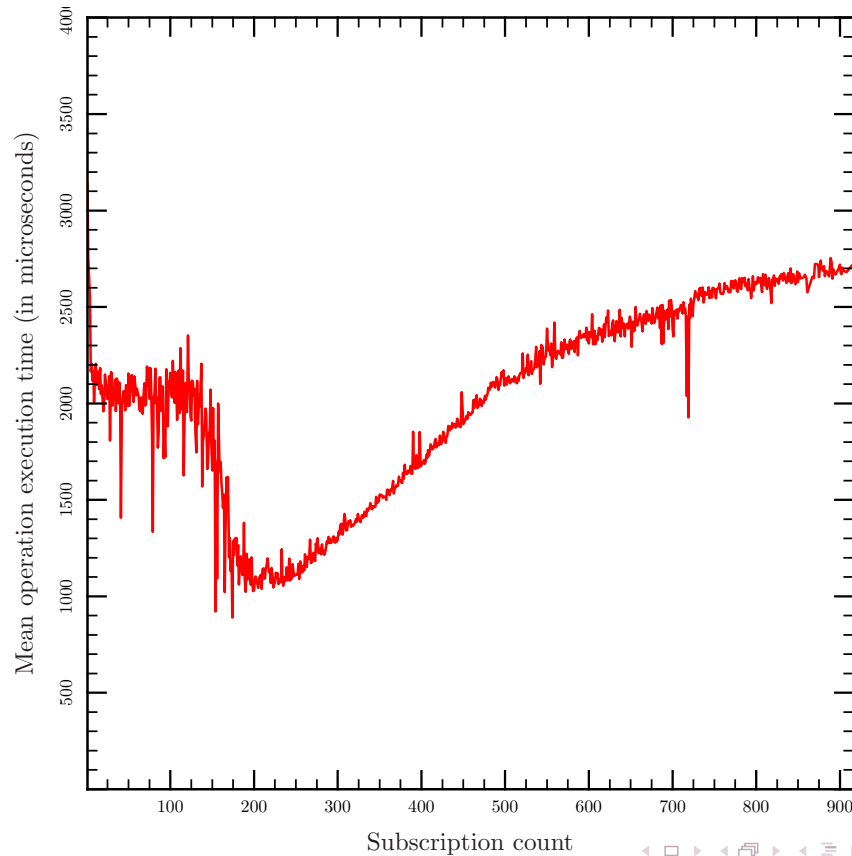
- Register 10 primary agents and 10 substitute agents
- Initiate N subscription to the smart space
- Simultaneously:
 - Insert and remove 50 random triples
 - Begin substitution for primary agents



Andrey Vasilev

Performance Evaluation of Operations in RedSIB with Substitution M

Impact of Substitution on SIB Operations



Andrey Vasilev

Performance Evaluation of Operations in RedSIB with Substitution M

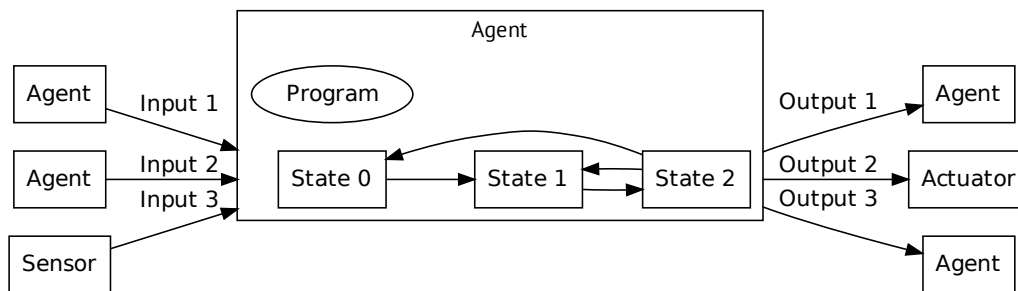
Conclusion and Future Work

- Substitution mechanism
 - Implemented as a SIB module
 - Uses special configuration ontolgy
 - Uses fixed number of subscriptions
- Performance considerations
 - Modifications did not impact on core operations
 - Substitution time is small, but should be improved
- Future Work
 - Migration to RedSIB 0.9

Andrey Vasilev

Performance Evaluation of Operations in RedSIB with Substitution M

Dataflow Agent Model

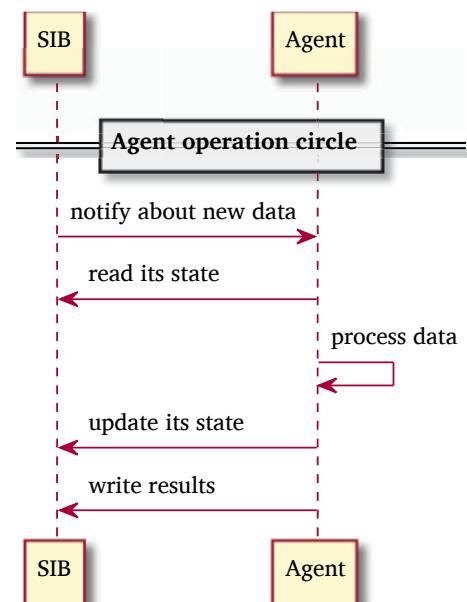


- Dataflow agents are stereotypical entities
- Dataflow agents are KPs conforming to certain rules
- Input channels are subscriptions
- Output channels are modified triples
- Internal state is a set of "private" triples

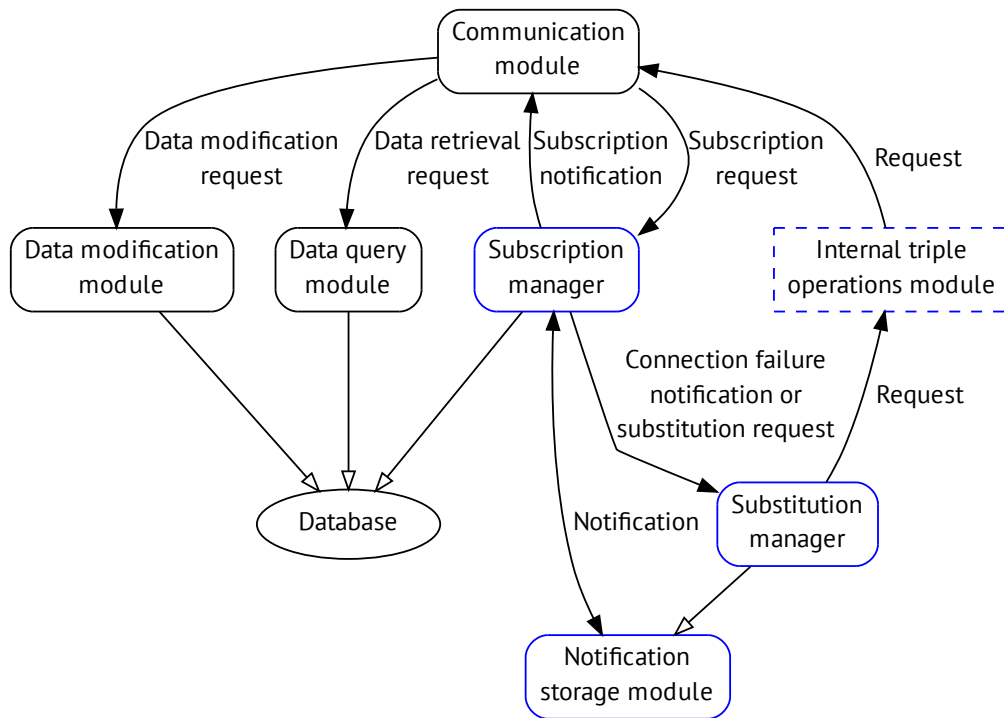


Dataflow Network Agent Implementation

- Dataflow nodes are KPs
- Context is stored inside SIB
- Inputs are subscriptions
- Outputs are triple modifications



SIB Modules Modification



M3 Infrastructures

RoDaFlow: A Framework for Development of Dataflow Network Agents in Smart-M3 with Substitution Method

Denis Laure
den.a.laure@gmail.com

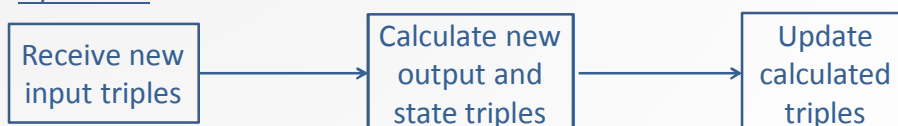
Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University 

Primary agent


Initialization

1. join the smart space
2. insert description triples
3. insert protection for description, output and state triples
4. calculate and inserts initial output and state triples
5. subscribe to input triples

Operation



Denis Laure

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University 

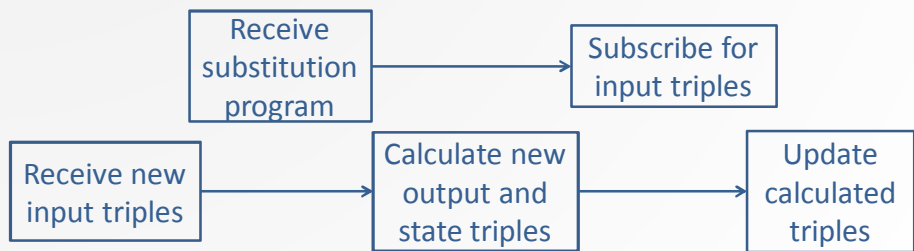
Substitute agent

3

Initialization

1. join the smart space
2. insert description triples
3. insert protection for description triples
4. subscribes to triple that indicates what primary agent is substituted by this one ('Substitutes' triple)

Operation



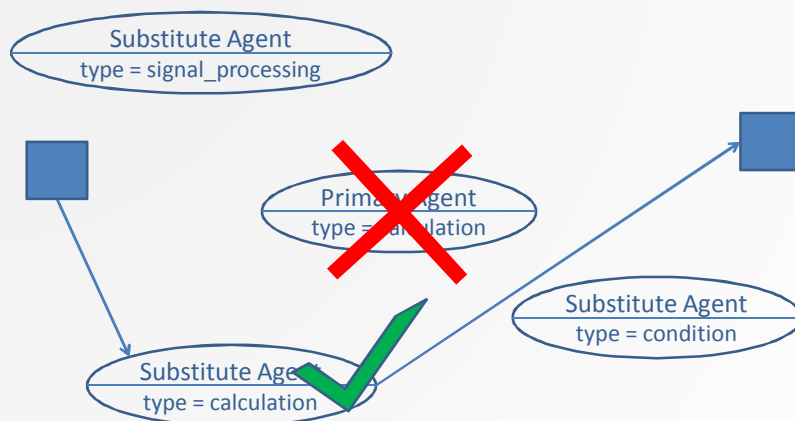
Denis Laure

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University



Choosing the Substitute Agent

4



Denis Laure

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University



Motivation

5

- The behavior and basic operations of the agents are always the same
- The agents differs only in their programs



Denis Laure

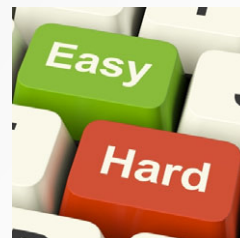
Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University



RoDaFlow Framework

6

- Allows to create dataflow network agents for Smart-M3 platform
- Allows to create agents by implementing only their programs
- Created agents support substitution mechanism
- Written in Java
- Uses Java KPI



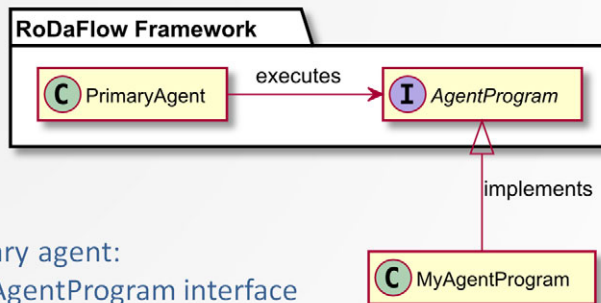
Denis Laure

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University



Implementing Primary Agent

7



To create primary agent:

- Implement AgentProgram interface
- Create instance of PrimaryAgent class
- Pass implemented agent program to created instance
- Call the joinSIB method on the PrimaryAgent instance

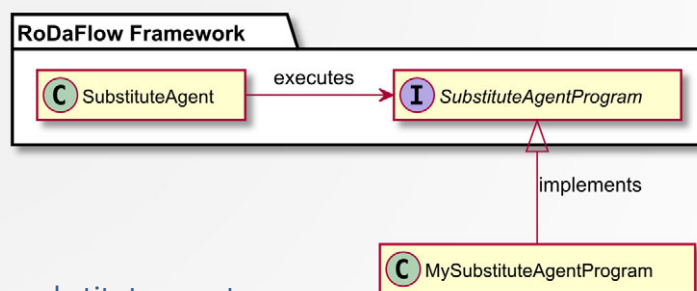
Denis Laure

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University



Implementing Substitute Agent

8



To create substitute agent:

- Implement SubstituteAgentProgram interface
- Create instance of SubstituteAgent class
- Pass implemented substitute agent program to created instance
- Call the joinSIB method on the SubstituteAgent instance

Denis Laure

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University



RoDaFlow Framework Benefits

- Saves the agents developer's time
- Simplifies the development of the agents
- Hence simplifies the development of dataflow network based systems
- Does not require from the developer any additional knowledge of substitution mechanism implementation
- Allows to create agents for:
 - Popular desktop platforms
 - Mobile devices
 - Oracle's Internet of Things platform



ORACLE

Denis Laure

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University 


Thank You!

Q&A



RoDaFlow framework homepage:
<http://yar.fruct.org/projects/rodaflow>

Denis Laure
den.a.laure@gmail.com

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University 

Context-Based Access Control for Ridesharing Service

Nikolay Teslya, Alexey Kashevnik, Michael Pashkin
St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
St.Petersburg, Russia
{teslya, alexey, michael}@iiias.spb.su

The ridesharing service provides a real-time fellow-travelers search. It makes mobile devices of drivers and potential passengers as well as computational devices interoperate in a common smart space, based on the Smart-M3 platform. The computational devices have a special software (e.g., ridesharing brokers) for complex computations, which cannot be run on the users' devices for several reasons, such as energy saving and computation capabilities. To operate the service needs information from users, such as their paths and preferences for searching matching paths. Most of this information cannot compromise the user's privacy, but there can be some information that has to be protected with access control for being used only by defined persons and services. For example, people do not often want to share their locations to others. At the current state of the Smart-M3 platform development, all of the information in the smart space is available for reading by any of its participant without any restrictions.

Nowadays there are three ongoing projects addressing information security for the Smart-M3 based smart space. The first project is based integrated mechanisms of triples protection in the Smart-M3. This protection only works for transactions, which modify the triples. It means that participants can protect triples to be inserted, updated or removed, according to a special template. However, all of the triples in smart space still can be queried or subscribed. The next project grants access permissions to triples like a file system grants permission to folders and files. It is achieved via mapping of all triples to a virtual file system and using the filesystem's possibilities of access control. Triples are mapped to files and the triples hierarchy mapped to the folder structure. Users are identified and authorized in the smart space through the host identity protocol (HIP). This approach provides rather flexible access control. At the same time, it is quite difficult to define user groups and to configure access permissions for all the files and folders since the project uses the discretionary access control model, which becomes very complicated in a dynamic system like a smart space with a lot of users and resources. The last project aims to provide a mechanism to grant access permissions based on the user's context. The following scheme of secure access to smart space resources has been proposed. The participant is identified by the system when registered in the smart space. The unique identifier is generated and saved in the access control broker. The public and private keys are generated using the RSA algorithm. A consumer service (the service of the participant) sends request to the public smart space to access some private information from service provider and subscribes to the corresponding response about the access granting. The smart space service provider accepts the request and calls a special access control broker service for the access permission. The access control broker confirms that this consumer is authenticated and applies the rules from the security policies to assign the role to the participant. The access permission is granted based on the role of the participant and then is sent to the smart space service, which requested it, through the virtual private space. Usage of the context for assigning roles makes the model more flexible and appropriate for services in the smart space.

The ridesharing service consists of two main parts: a client application and ridesharing broker. The interaction between these parts is achieved through the smart space, based on the Smart-M3 platform. The client application can be installed on user's mobile devices with Android operating system, which is one of the most popular mobile operating systems in world now. The main functions of the client application are:

- Collecting various information from the user and sending it to the smart space.
- Receiving information about joint trip from the smart space and presenting it to the user.

The ridesharing broker is a software, installed on computers with high computation capabilities. Its main function is receiving information from the smart space and finding matches between ridesharing service's users paths. Finding such matches is a very complicated task that needs a lot of computational power and not energy safe. Therefore, this task is performed on specialized computers.

General scheme of request process between client application and ridesharing broker using the context-based access control model is presented in Fig. 1. In this scheme, the smart space consists of: a participant, which requests the information; a service, which provides this information; and an access control broker, which provides access permissions to the participant based on its context. The information flow between the participant and the service becomes private due to the virtual private space, which has no intersections with the public space. The participant publishes the request to the public smart space for

accessing private information (in the RDF notation) and subscribes to the corresponding response about the access granting. The smart space service accepts this request and calls the access control broker for the access permission decision making.

The access control broker reads the participant's context and verifies its digital signature using the open key. If the signature is correct, the broker confirms that this user is authenticated and applies the rules from the security policies to assign the role to the participant. The access permission is granted based on the role of the participant and then is sent to the smart space service, which requested it. If the access to the resource is granted, the smart space service creates a virtual private space. The information requested by the participant is transferred to this private smart space. The connection information (space IP, space port and space name) is encrypted via the open participant's key and is sent to the public smart space. If the access is denied, the service sends the corresponding notification to the smart space participant.

Participant, who sends the information request, is notified via the Smart-M3 subscription mechanism. If the access is granted the participant decodes the encoded data with its private key and creates a connection to the specified virtual private smart space. When the requested information is transferred, the virtual private space is destroyed.

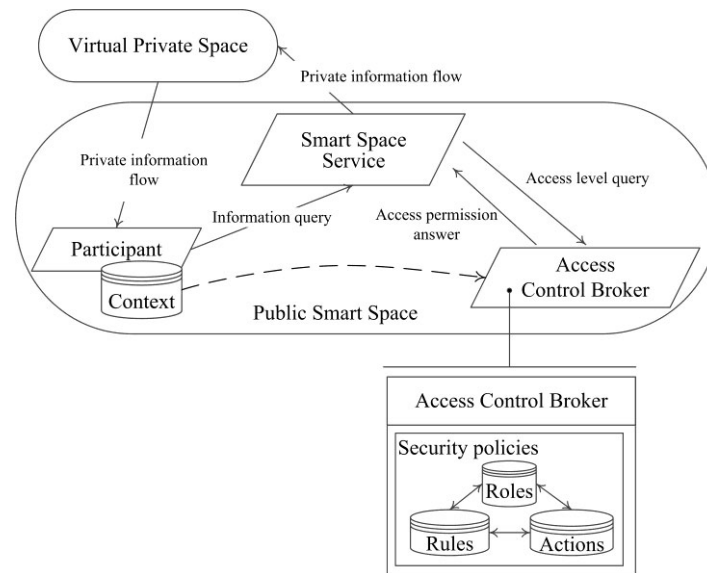


Fig. 1. General scheme of context-based access control model

User description in the ridesharing service includes private information about that user. First, there are no people who want to be tracked. Without the access control, every user can collect locations of other users and predict the future steps for different purposes. Also, there are no people who want to share their real names and social networks information for everyone. Law of any state also restricts collecting any information about any person. The access control broker for ridesharing service has been developed to prevent the collecting information about users of this service by criminals.

The access control broker for ridesharing service has been developed using Python programming language. As an example, situation of two users, trying to get access to the information about each other, is described. There are following rules for access permission granting: users should be friends or have mutual friends and both should be in the same region.

First, it is needed to configure the TrustValue rules in the access control broker. In Python, it looks as follows:

```
trustValue_rules = {'friendship':{'friend':0.9, 'not_friend':0.1}, 'is_a':
{'Driver':0, 'Passenger':1}, 'currentLocation':{'Russia:0.8, Finland:0.8, China:0.1,
North Korea:0.1}}
```

Then, the ranges for assigning roles and allowed actions for role actions need to be configured:

```
ranges_for_roles =
{'trustedUser': {'friendship':[0.8,1], 'is_a':[1], 'currentLocation':[0.7,1]},
'untrustedUser': {'friendship':[0,0.79], 'is_a':[0], 'currentLocation':[0,0.7]}}

role_actions = {'trustedUser': ['read_private_inf', 'read_only_public'],
'untrustedUser': ['read_only_public']}
```



Context-Based Access Control Model for Ridesharing Service

Nikolay Teslya, Alexey Kashevnik, Michael Pashkin

Laboratory of Computer Aided Integrated Systems
St.Petersburg Institute for Informatics and Automation of RAS (SPIIRAS)

Introduction



• Motivation

- Ridesharing service needs to have an access control to the user's private information:
 - There are no people who want to be tracked.
 - Without the access control, every user can collect locations of other users and predict the future steps for different purposes.
 - There are no people who want to share their real names and social networks information for everyone.
 - Law of any state also restricts collecting any information about any person.

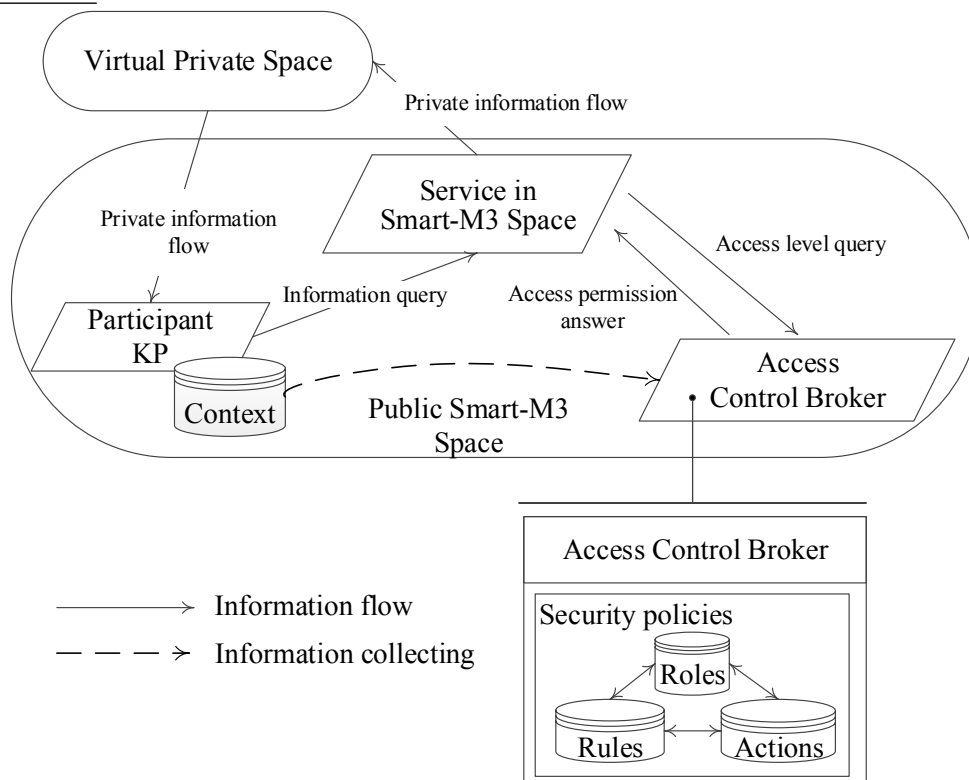


Table of contents

- Conceptual models of the context-based access control module and the ridesharing service
- Context formalization for the ridesharing service.
- Examples of rules configuration for access control module.

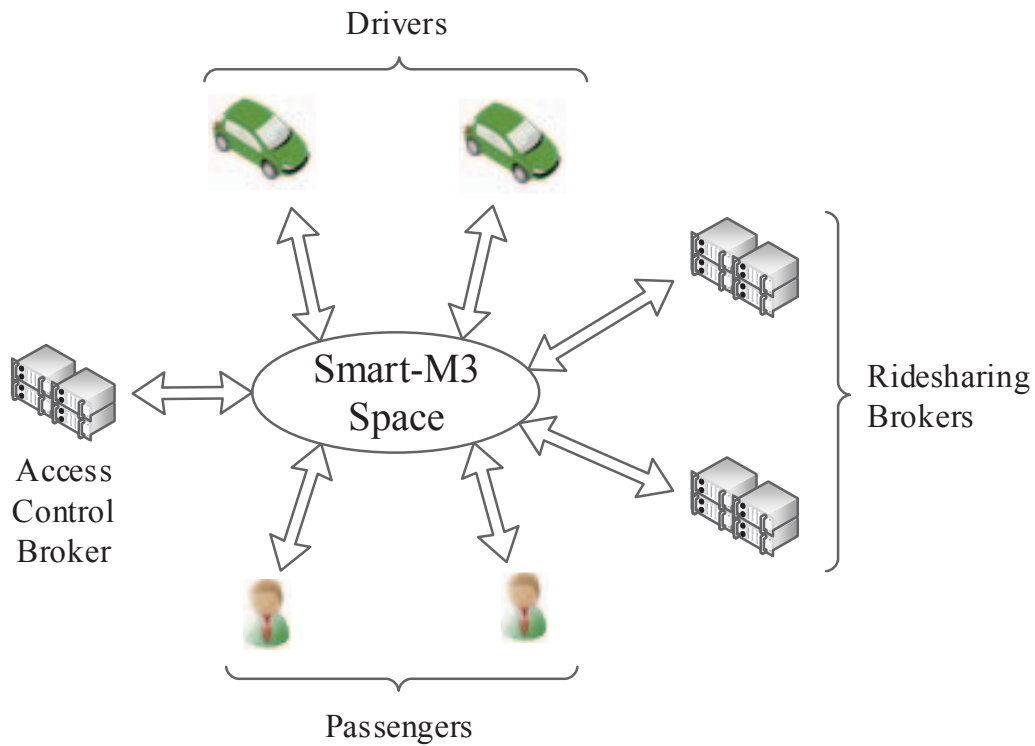
3

Conceptual Model of Smart Space Access Control Module



4

Conceptual Model of Ridesharing Service With Access Control

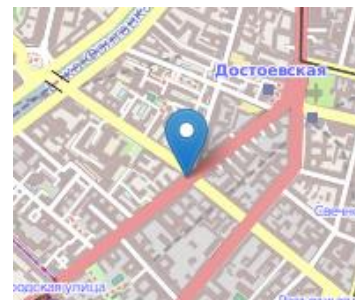


5

Context Formalization: Physical Part



- User Location. Latitude & longitude. Building routes and defining place to access information from



- Date and time. Access permission in specified time interval



6



Context Formalization: Virtual Part

- ID. Unique user's identifier in the service
- Different kind of authentication information:
 - Password;
 - Private key;
 - Certificate;
 - Device fingerprint (e.g. H. Bojinov, "fingerprinting" a smart phone through its accelerometer);
 - IMEI;
 - etc.

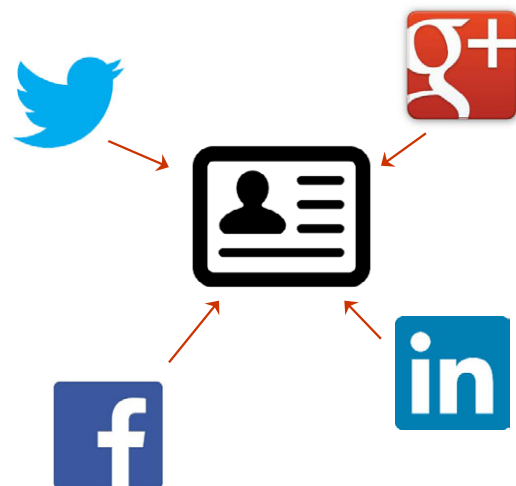


7



Context Formalization: Social Part

- Role in the ridesharing service
- Information from the different social networks:
 - Name, surname
 - Date of the birth
 - Birth place
 - Interests
 - etc.



8



Access Control Policy Rules

- Policy consists of 3 rule's types:
 - 1) *TrustValue* rules. Used to assign the numeric trust value to the context component.
 - 2) *Assign_role* rules. Used at the time of access request for role assignment.
 - 3) *Permissions* rules. These rules contain access control policies, which determine whether a participant with a certain role is allowed to access a particular resource type or not.

9



TrustValue Rules Examples

- $\text{TrustValue}(\text{isPassenger} == \text{true}) = 1;$
- $\text{TrustValue}(\text{isPassenger} == \text{false}) = 0;$
- $\text{TrustValue}('08:00' < \text{currentTime} < '17:00') = 0.8;$
- $\text{TrustValue}('08:00' > \text{currentTime} > '17:00') = 0.2;$
- $\text{TrustValue}(\text{currentLocation } 'in\ set' [\text{Russia, Finland}]) = 0.9;$
- $\text{TrustValue}(\text{currentLocation } 'in\ set' [\text{China, North Korea}]) = 0.1;$
- $\text{TrustValue}(\text{commonInterests } 'more\ than' 1/2) = 0.7;$
- $\text{TrustValue}(\text{commonInterests } 'less\ than' 1/4) = 0.3;$
- $\text{TrustValue}(\text{birthDateAccept } 'before' 1985) = 0.8;$
- $\text{TrustValue}(\text{birthDateAccept } 'after' 1985) = 0.2;$

10



Assign_role Rules (1/2)

- Five main roles: owner, trustedPassenger, untrustedPassenger, trustedDriver, untrustedDriver.
- Example:
 - A user has a trustedPassenger role only if:
 - the user is a passenger,
 - his/her current location is Russia or Finland,
 - the time is from 8 am to 5 pm,
 - his/her birth date is before 1985.

11



Assign_role Rules (2/2)

- According to the trust values presented before, the rules are set by the following way:
 - $\text{AssignRole}(\text{trustedPassenger}) = (\text{TrustValue}(\text{isPassenger}) = 1) \& (\text{TrustValue}(\text{currentLocation} \in (0.7, 1)) \& (\text{TrustValue}(\text{currentTime}) \in (0.6, 1)) \& (\text{TrustValue}(\text{birthDateAccept}) \in (0.7, 1))$.
 - $\text{AssignRole}(\text{untrustedPassenger}) = (\text{TrustValue}(\text{isPassenger}) = 0) \& (\text{TrustValue}(\text{currentLocation} \in (0.1, 0.5)) \& (\text{TrustValue}(\text{currentTime}) \in (0, 0.6)) \& (\text{TrustValue}(\text{birthDateAccept}) \in (0, 0.7))$.

12



Permissions Rules

- Determines whether a participant with a certain role is allowed to access a particular resource type or not.
- Examples:
 - `Permission(trustedPassenger) = "readCommon", "readPrivate";`
 - `Permission(untrustedPassenger) = "readCommon".`

13



Conclusion

- Access control in the ridesharing service allows users to share their private information only with persons they trust via using the easy configurable preferences.
- The ontology of the ridesharing service has been used for defining the context of the service users.
- The rules of the access control broker are based on the context of the service users.
- The access control broker for ridesharing service helps to prevent the collecting information about users of this service by criminals.

14

**Thank you for Attention
Questions are Welcome**



E-mail: teslya@iias.spb.su

Reusable, semantic, and context-aware micro-architecture

Approach to managing interoperability and dynamics in smart spaces

Susanna Pantsar-Syväniemi, VTT Technical Research Centre of Finland

Abstract

The amount of shared information has increased a great deal in ubiquitous systems, where the previously isolated devices and appliances have become part of the system and are producing or consuming the information. The ubiquitous system, or the smart environment, lacks an approach that supports scalability and enables semantic interoperability. It is challenging to provide a dynamic behavior at the run time without human intervention. A number of dedicated solutions have been developed for the ubiquitous environment because of its complexity. The dedicated solutions are usually non reusable.

An approach is needed that i) is reusable as such or partly, ii) provides the semantic interoperability, iii) enables dynamic and behavioral interoperability between the receiver and sender of the information at run time, and iv) is scalable by being modular, and decoupled.

This work proposes a novel approach to managing interoperability and dynamics in smart spaces. The approach includes a Context-Aware Micro-Architecture (CAMA), and a Context Ontology for Smart Spaces (CO4SS). This approach is independent of implementation languages and communication techniques. CAMA, as an architectural pattern, is usable without its semantic support, CO4SS. In the literature, it is the first approach that fulfills the requirements that are set for a context data distribution system.

The power in CAMA relies on the usage of the standard and web-based techniques, in the separation-of-concerns principle, and in the enhanced control loop, MAPE-K. The latter has four parts, Monitor; Analyze; Plan; Execute that share Knowledge. This work will combine the IOP and the MAPE-K loop in a single solution and will also extend the IOP with a context-awareness capability. Use cases implemented to validate the CAMA and the CO4SS has used IOP-based SIBs. The proposed solution, the micro-architecture for managing the context, requires an RDF-based database, but not necessary any of the IOP-based SIBs.

CAMA is highly dynamic, which is due to the run-time updatable rules. The creation of the rules is laborious, as they are written into text boxes of Message Sequence Charts. This will be improved when new tools are developed for the rule creation. Additional research is needed to validate the scalability of the approach with a "Big data". CO4SS can be widened with the domain-specific and quality ontologies. It supports the evolution management of the smart space: all smart spaces and their applications 'understand' the common language that is defined by it. CO4SS has the potential to be a de facto ontology for the context-aware, i.e., intelligent applications.

More information is available: <http://www.vtt.fi/inf/pdf/science/2013/S37.pdf>

Reusable, semantic, and context-aware micro-architecture. Approach to managing interoperability and dynamics in smart spaces

Open International M3 Semantic Interoperability Workshop, Helsinki, Finland, 12.11.2013

Susanna Pantsar-Syvaniemi, D.Sc. (Tech.)

susanna.pantsar-syvaniemi@vtt.fi, +358 40 505 6682

Information – produce, distribute, exploit



devices / things

applications

Internet / networks



Smart Space - Internet of Things



Definition and Management of Context Information

increasing amount of information

filtering the information

semantics

timely

context-awareness

relevant information



dynamics

openness

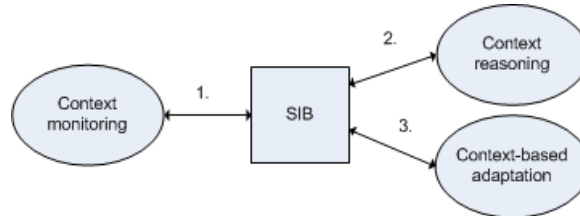
data security

real-time

reusability

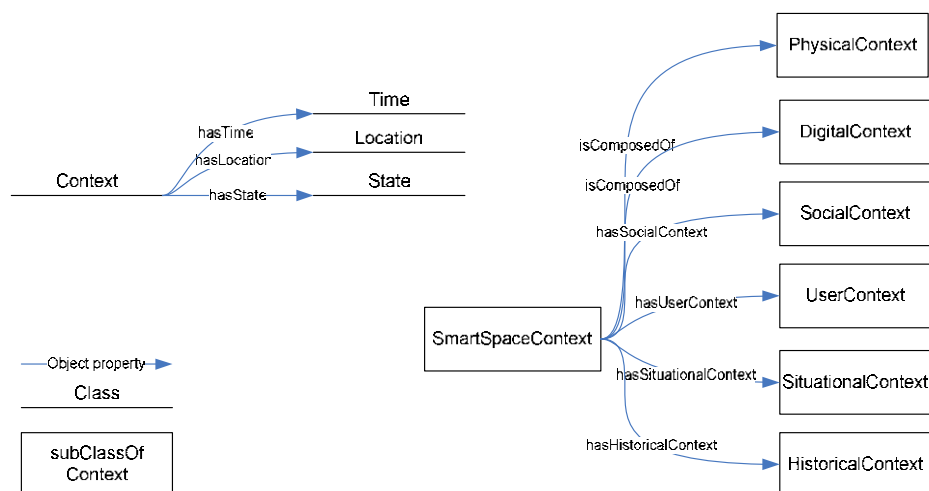
modularity

Context-Aware Micro-Architecture (CAMA)



- Scalable (modular, autonomous agents)
- Reusable – as such or partly
- Semantic interoperability
- Real-time context management
- Configurable at run time

Main class of Context Ontology (CO4SS) - Context

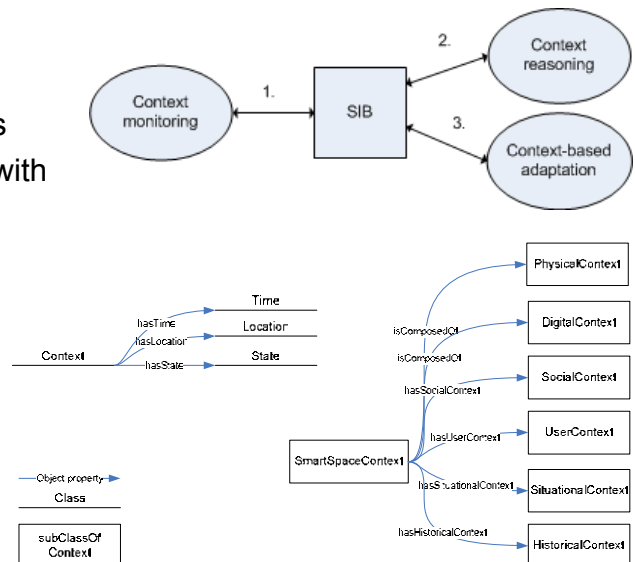


- Generic and expandable to different applications and to run-time quality management
- Concepts to recognize situations
- modular (consists of subsets), supports reusability

How to enhance

More testing

- with 'Big Data' database
- multiplying the amount of CAMA agents
- Stressing the context reasoning agent with more rules and with more nested rules



Thesis, video: Sofia VTT Smart Door

Thesis,

<http://www.vtt.fi/inf/pdf/science/2013/S37.pdf>

Sofia VTT Smart Door,

<http://www.youtube.com/watch?v=anRW0y2r1Q0>

The SmartRoom Infrastructure: Service Runtime Reliability

Ivan Galov, Dmitry Korzun

Petrozavodsk State University
Department of Computer Science



This project is supported by grant KA179 of Karelia ENPI — joint program of the European Union, Russian Federation and the Republic of Finland



14th FRUCT conference
November 12, 2013, Helsinki, Finland



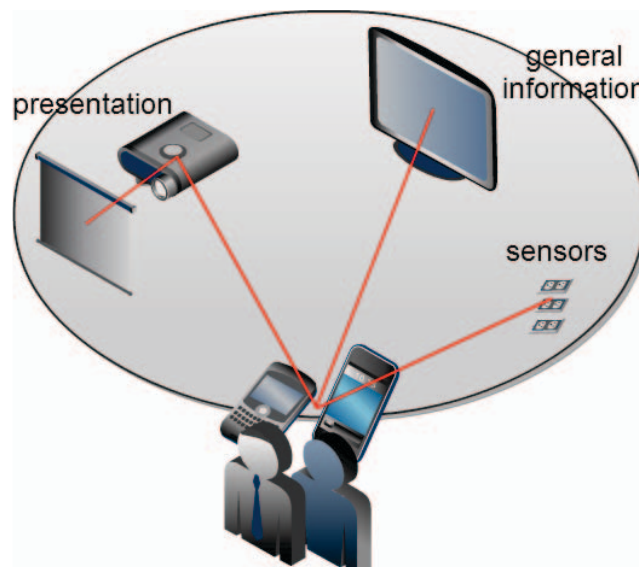
Ivan Galov

The SmartRoom Infrastructure

12.11.2013 1 / 12

Smart Room at PetrSU

- holding conferences, meetings, and lectures
- personalized interaction with room participant
- participating using mobile devices



Ivan Galov

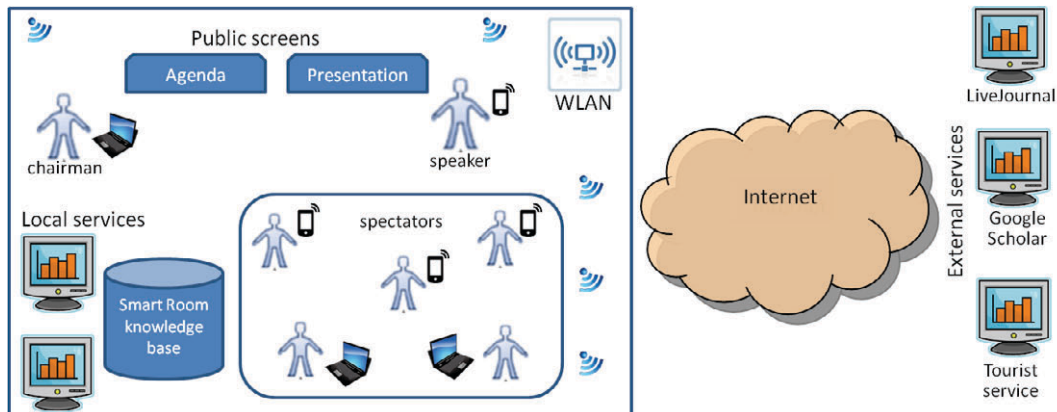
The SmartRoom Infrastructure

12.11.2013 2 / 12

SmartRoom services

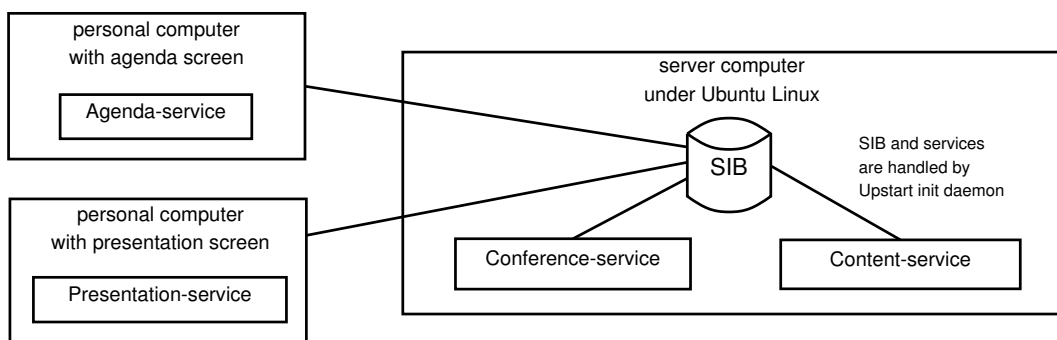
- conference services
- sensor services
- activity tracking services
- discussion services
- ...

SmartRoom is being deployed at PetrSU IT-park, 104 aud.



Navigation icons: back, forward, search, etc.

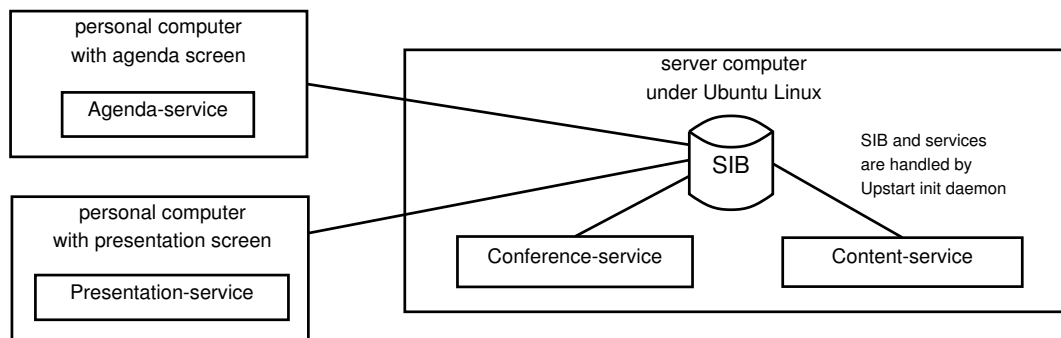
SmartRoom infrastructure



- SIB is deployed either locally or on a remote machine
- Conference-service: conference runtime management
- Presentation-service: slide show of current speaker
- Agenda-service: visual activity agenda
- Future: Meeting-service, Lecture-service, ...

Navigation icons: back, forward, search, etc.

Infrastructure KP deployment options



- 1 the same server computer with the SIB
- 2 dedicated computers to attach specific devices
- 3 additional server computers (non-SIB)

infrastructure also includes KPs for the system administration

Services reliability

- Infrastructure element failure: crash of Smart-M3 processes or SmartRoom services
- Subscription query: it is not guaranteed that the application receives every subscription notification

Possible solutions:

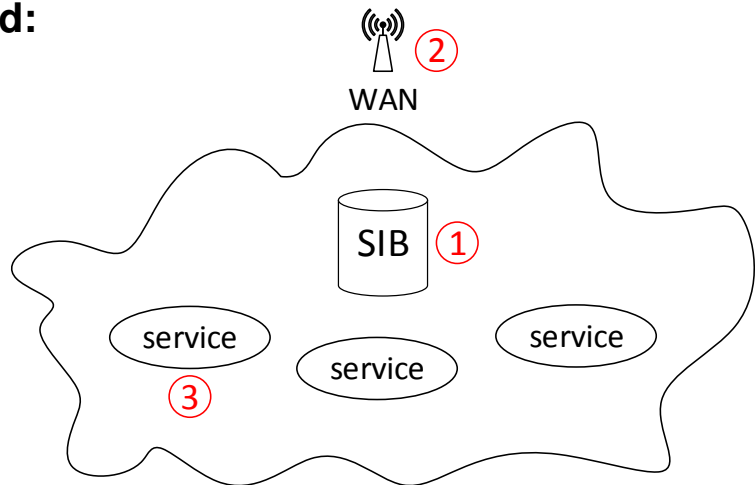
- Subscription application-level control: KP performs additional timer-based checks for the service subscription
- Restart/Reconnection: A service restores its runtime and connection with the smart space
- Infrastructure: online services and content management

Restart/Reconnection

- Restart: application is shut down and launched again
- Reconnection: application is still running but connection to SIB is re-established

Problems when it is needed:

- SIB failure
 - ▶ data are saved (Virtuoso)
 - ▶ data are lost
- WiFi access point failure
- particular service failure



Navigation icons: back, forward, search, etc.

Online Services

- main goal: increase reliability of service provision
- operate permanently and continuously on server computers
- provide the service in 24/7 mode
- Upstart event-based init daemon handles the services
 - ▶ automatic respawning: whatever happens with the service, it will be relaunched again
 - ▶ services chaining: one service is automatically launched when another service is started

redsibd → sib-tcp → Smart Room initialization →
conference-service → other services

Navigation icons: back, forward, search, etc.

Online Services example

redsibd	start on startup
sib-tcp	start on started redsibd stop on stopping redsibd
conference-service	start on started sib-tcp stop on stopping sib-tcp
...	...

```
start on started sib-tcp
stop on stopping sib-tcp
```

```
env SERVICE=/usr/local/bin/conference-service
```

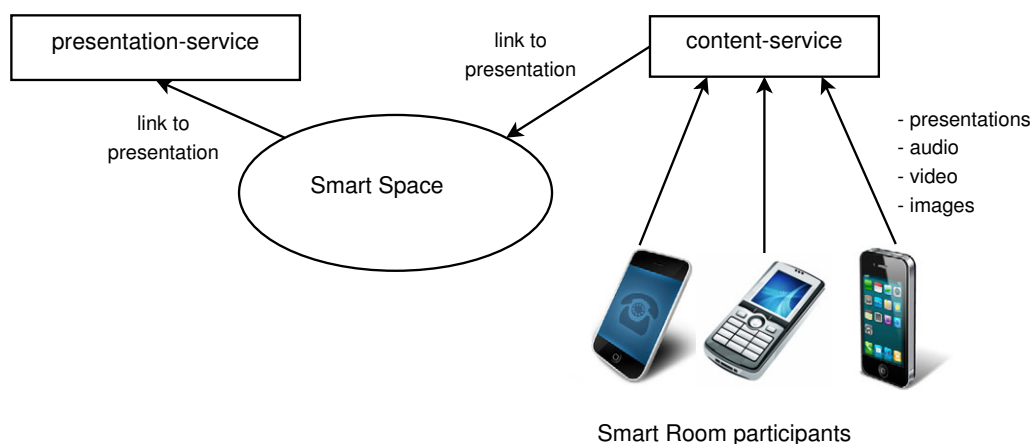
```
respawn
exec $SERVICE
```

```
post-start script
    python /usr/local/bin/register.py
end script
```

Navigation icons: back, forward, search, etc.

Content Management

- Smart space is a semantic hub: relating multi-source information without duplicating all the data in the smart space
- Content-service stores and shares information among others
- Every service or client can upload some data (video, audio, etc.)
- Content-service saves data and provides download links



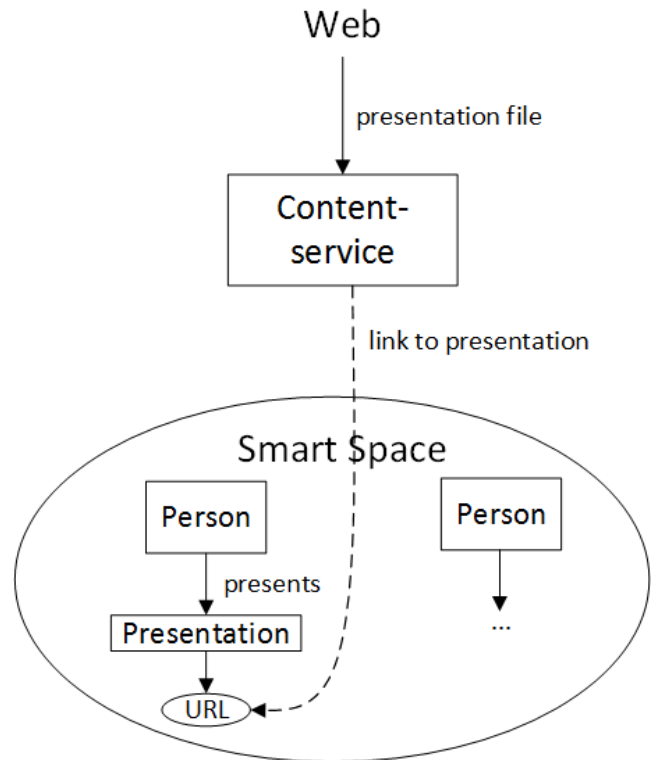
Navigation icons: back, forward, search, etc.

Content-service links

- direct links
 - ▶ global — on external resources
 - ▶ local — on resources stored by the service

Problem: dynamic changing of content-service IP for local links

- relative links — relative path to resources stored on content-service
Content-service IP is stored and updated in service space



Navigation icons: back, forward, search, etc.

Conclusion

- Mechanisms for service reliability
 - ▶ Service delivery: subscription control, reconnection
 - ▶ Infrastructure: online services, content management
- Basic online services controlled by Upstart init daemon: SIB processes, Conference-service
- Content-service is still in development stage
- Open source code:
<http://sourceforge.net/projects/smartroom>
- Welcome to our demo on 13.11.2013

Thank you for attention!

Navigation icons: back, forward, search, etc.

M3 Based Knowledge Sharing Protocol applied to Bluetooth LE sensors and actuators

Arto Ylisaukko-oja

Esa Viljamaa

Jussi Kiljander

VTT Technical Research Centre of Finland
Kaitoväylä 1
90570 Oulu
+358 20 722 111

firstname.lastname@vtt.fi

ABSTRACT

We describe the concept and the status of development of an M3 architecture based demonstration for low capacity sensors and actuators, using Bluetooth LE radios for communications. As a novel technology, Knowledge Sharing Protocol (KSP) is applied to all the semantic messages between the low capacity devices and the SIB (Semantic Information Broker). KSP uses a compact binary format [1]. This enables efficient implementation and low power consumption for the battery operated, resource constrained Bluetooth LE sensors and actuators.

Another key component is the M3Box, running a SIB implementation that can communicate using KSP compatible messages. This is the heart of the local M3 system. It is based mainly on Seco Qseven Quadmo747-E660 computer. The M3Box supports various radio technologies, including Bluetooth LE.

The purpose is to show that low capacity devices can be part of semantic interoperability, without undue adverse effects on requirements for power supply, processing power, memory size, communication capability and cost.

The demonstration is a home automation scenario, showing how the semantic architecture can enable re-use of different sensors or actuators for new applications that were not originally even designed as part of the system.

THE DEMONSTRATION CONCEPT

Figure 1 shows the starting point of the example home automation scenario: The system includes SIB (M3Box), a wireless Bluetooth LE motion detector sensor, and a wireless Bluetooth LE actuator, being capable of switching lighting on or off. In addition, there is a mobile phone which is used to update user presence information to the SIB. The motion detector updates movement data to SIB, and the lighting actuator reads both user presence and movement data from the SIB. This arrangement avoids switching the lights on because of a family pet movement, for example: the lights are switched on when there is movement *and* the mobile application has registered a person being present.

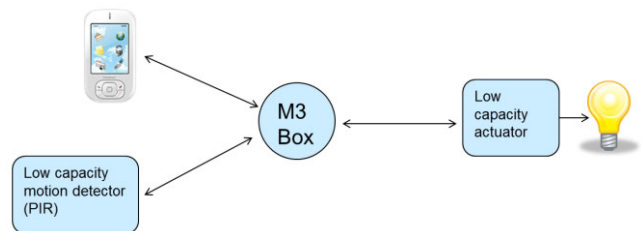


Figure 1. The demonstration in its simplest form: automatic light control based on movement detection and presence data.

Figure 2 shows how a burglar alarm feature has been added by a siren and optionally hall sensor(s) monitoring the doors and/or windows. The burglar alarm application can also utilize the user presence and motion detector data to launch the alarm.

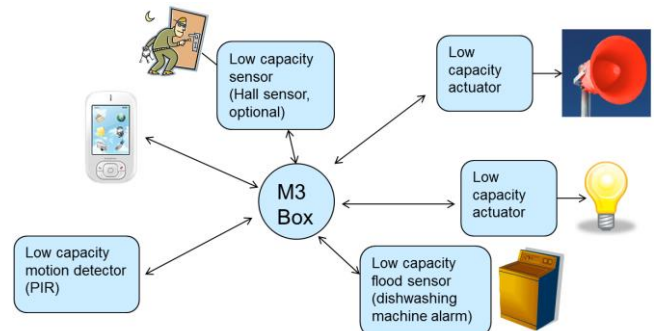


Figure 2. A siren is added for a burglar alarm application. Additional sensors such as hall sensors for doors can be added. Flood sensor is shown as a third application.

As another increment to the system, a flood sensor is added, making use of the siren (and possibly a mobile application) to notify the resident of the house so that damage can be avoided because of a water leakage.

REFERENCES

[1] Kiljander, J., Morandi, F., Soininen, J-P.: Knowledge Sharing Protocol for Smart Spaces. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 9, 2012.

M3 Based Knowledge Sharing Protocol Applied to Bluetooth LE Sensors and Actuators

12.11.2013

Arto Ylisaukko-oja

Esa Viljamaa

Jussi Kiljander

Goal and Method of the Research

- **Goal:** to demonstrate that low capacity devices can be part of semantic interoperability, without undue adverse effects on requirements for
 - Power supply
 - Processing power
 - Memory size
 - Communication capability
 - Cost
- **Method:**
 - Application of Knowledge Sharing Protocol (KSP) to M3 architecture, consisting of a local Semantic Information Broker (SIB) and Knowledge Processors (KP) that operate as sensors or actuators, utilizing Bluetooth Low Energy as a means for data transport.
 - Implementation in real HW and SW (some KPs in simulation only)
 - Appropriate measurements for figures such as required memory and power consumption

Key Technologies

Knowledge Sharing Protocol (KSP)

- A novel knowledge sharing protocol (KSP) for semantic technology empowered ubiquitous computing systems
- Designed to be used with M3 architecture
- Utilizes a compact, binary format that is designed to be suitable also for resource restricted devices and networks
- Can be used on top of various data transport layers
- **Refer to Jussi Kiljander's separate presentation about KSP**

M3 application logic					
KSP					
TLS	DTLS	CoAP	RFCOMM	o t h e r s	
TCP	UDP		L2CAP		
IPv4/IPv6	6LoWPAN				
Ethernet/ WiFi	IEEE 802.15.4	IEEE 802.15.1			

Bluetooth Low Energy

- Marketed as *Bluetooth Smart*
- Included to Bluetooth Specification 4.0
- Enables a new class of devices with very low power consumption, low cost and long operation times even from a small 3V coin cell battery
- Profiles have been defined for various applications such as health care, sports and fitness.
- Texas Instruments CC2540 "keyfob" development kit is used as a platform: 8051 microcontroller with 128 kB flash and 8 kB RAM memory.



M3Box

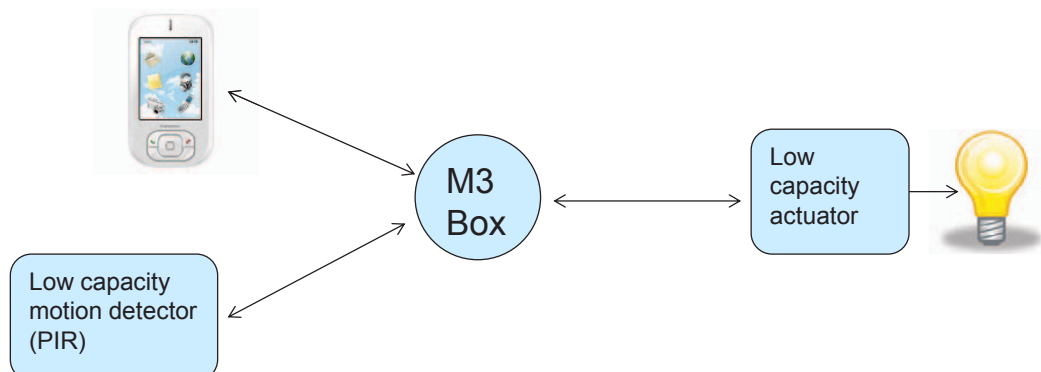
- A SIB platform by VTT
- In this case, M3Box runs a SIB implementation that can communicate using KSP compatible messages
- Based mainly on Seco Qseven Quadmo747-E660 single board computer with Linux operating system



Demonstration Scenario

Incremental Applications, step 1/3

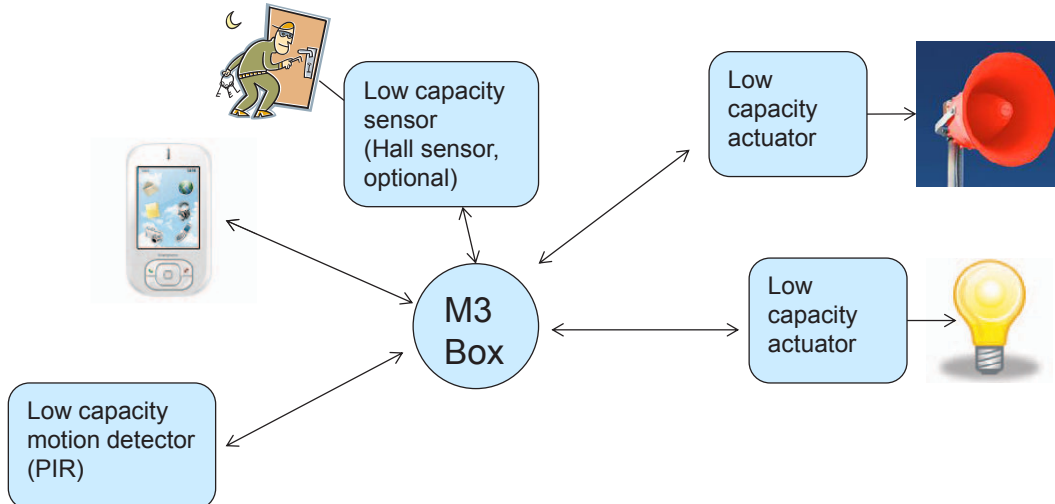
- To demonstrate the versatility of semantic technologies in smart spaces
- **Phase 1)** A basic application exists and consists of M3Box and low capacity sensors/actuators
 - Automatic lighting control based on user presence/location
 - User presence information based on mobile phone connection (e.g. BT LE)
 - User location based on motion sensor
 - Based on this data, the actuator will switch the room light on/off



Incremental Applications, step 2/3

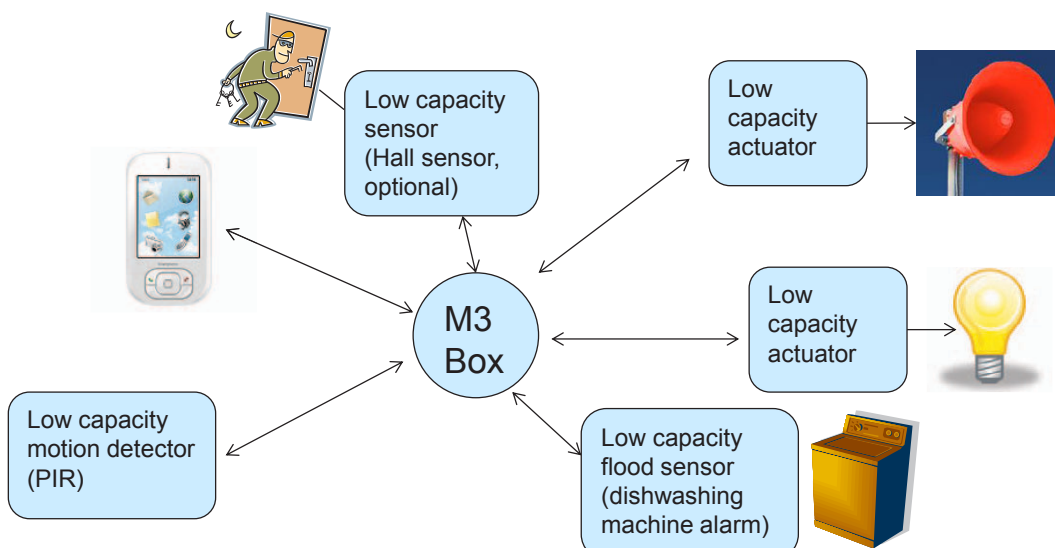
Phase 2) A siren and a door sensor (optional) are added to the system. The user wants to utilize also the existing motion sensor to implement an alarm system.

- Based on semantic technologies, this is basically possible. If motion is detected when no user is logged in, an alarm is triggered.

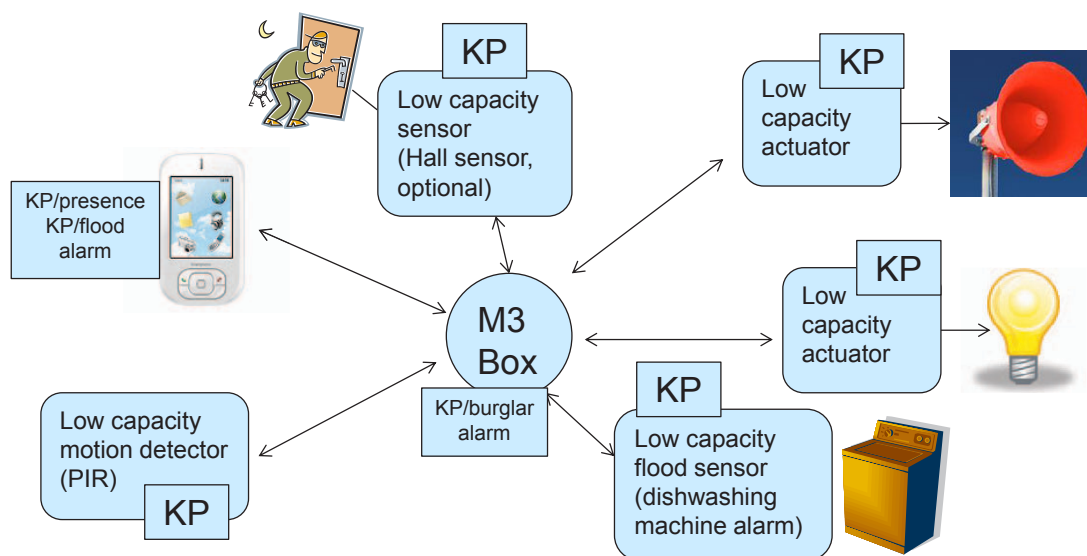


Incremental Applications, step 3/3

Phase 3) A flood sensor is added to the system. The user wants to monitor leaks under dishwashing machine to prevent damage caused by water. The siren that was purchased for burglar alarm application is re-used. In addition, smartphone application shows the reason for alarm.



Knowledge Processors (KP)



Implementation and Performance

Implementation

- M3Box HW and encapsulation have been implemented, SW development in progress
- SIB code with KSP support exists
- KSP has been implemented to the Bluetooth Low Energy motion sensor
 - A novel *KSP Bluetooth Low Energy Profile* has been implemented
- Work in progress with KSP enabled lighting control node (Bluetooth Low Energy)
- Some of the KPs will be simulated instead of physical implementation

Effect on Memory Consumption

Bluetooth Low Energy PIR sensor without KSP:

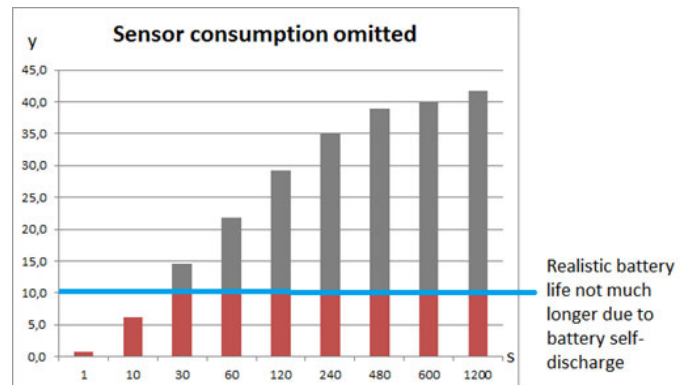
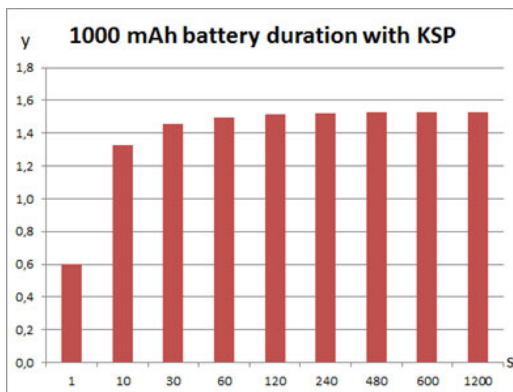
- Flash memory consumption 113 423 bytes
- RAM memory consumption 6289 bytes

Bluetooth Low Energy PIR sensor with KSP:

- Flash memory consumption 125 880 bytes (+11%)
- RAM memory consumption 6483 bytes (+3%)
 - Remark: may be increased up to ~5% in future enhanced version with larger writable characteristic for received KSP messages in the Bluetooth profile.

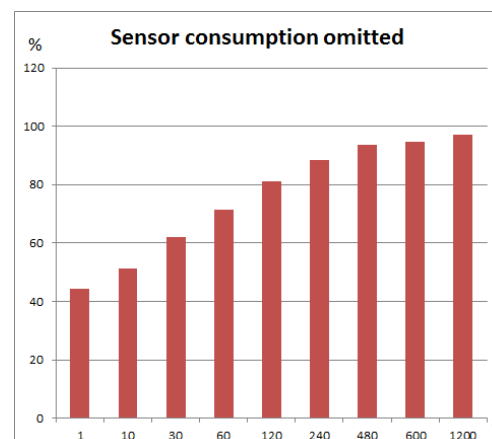
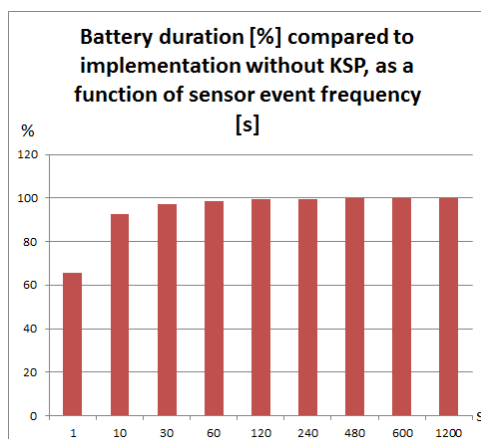
Power Consumption

- The figures show theoretical duration of the battery pack used (3 x 1,5V AAA batteries in series) as a function of sensor event frequency.
- Sensor event: “Motion detected” is reported to the BT master, followed later by “Motion not detected”.
- NON type KSP messaging: SIB does not need to acknowledge in KSP level.



Battery duration compared to implementation without KSP

- The figures show the percentage of battery duration compared to implementation without KSP, as a function of sensor event frequency.
- KSP data format increases power consumption, since a “motion detected” or “motion not detected” needs to be sent in five different BT LE messages (instead of only one in case of non-KSP implementation), each containing 20 bytes of data payload.



Conclusion

- The implemented PIR sensor demonstrates that KSP is feasible on top of Bluetooth Low Energy transport and a low capacity microcontroller.
- Increase with memory consumption was relatively modest and did not require switching to a chip version with higher memory resources; no additional cost in this case.
- There is a notable effect on power consumption and therefore, battery duration, but its significance depends on:
 - how data intensive the application is
 - Does the radio communication dominate power consumption, or something else such as sensing
- In a practical motion sensing scenario, the KSP's effect on power consumption or battery duration is negligible.

Programming Android Client for Smart-M3 Applications

Pavel Y. Kovyrshin, Dmitry G. Korzun

Petrozavodsk State University
Department of Computer Science



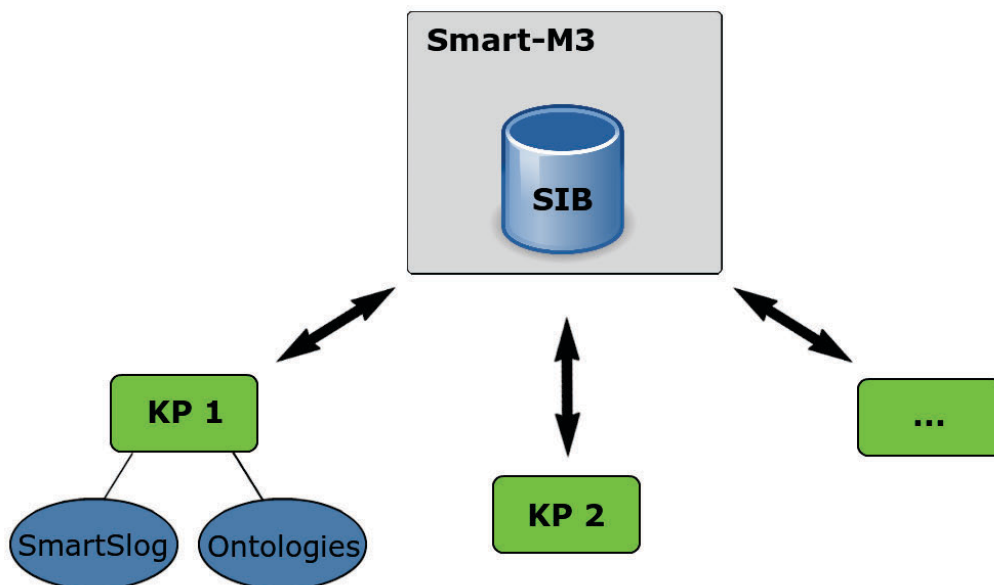
Grant KA179 "Complex development of regional cooperation in the field of open ICT innovations" of Karelia ENPI CBC programme 2007–2013 of the European Union, the Russian Federation and the Republic of Finland



14th FRUCT Conference
November 12, 2013, Helsinki, Finland



Smart-M3 Application



Implementation tasks



SmartSlog integration to android project

SmartSlog is a development kit for programming Smart-M3 Knowledge Processors (KP)

GUI for services representation

User interface constructed by android libraries

Smart Space interaction

Implementation of C based application logic

Development kits

Android SDK for Java based GUI design

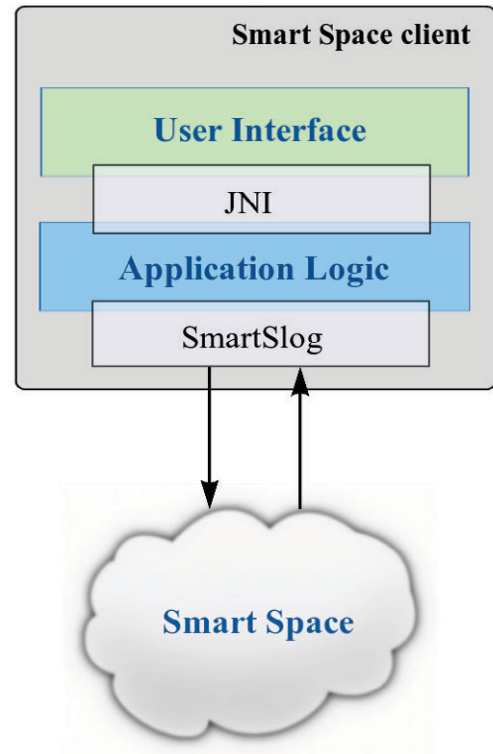
Android NDK for C based application logic implementation



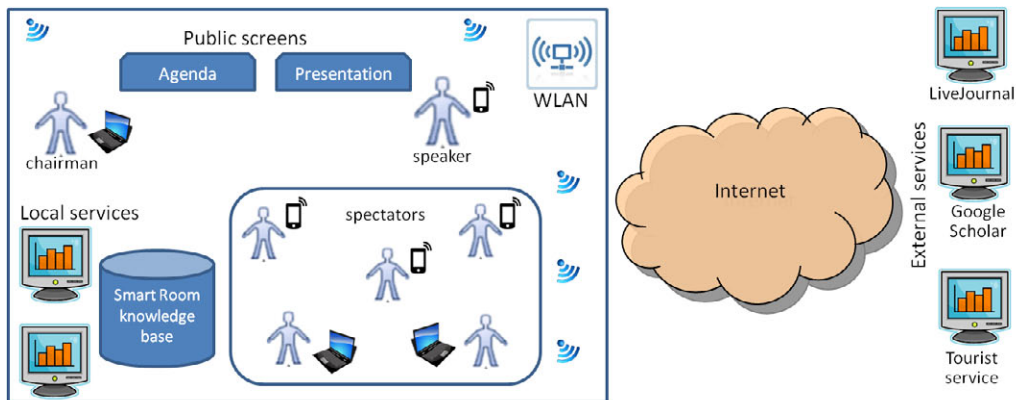
Java Native Interface (**JNI**) is a bridge between user interface and logic

Client architecture

- **User interface** module uses Android SDK and is Java based
- **JNI** is platform dependent
- **Application logic** is C written and supported by Android NDK
- **SmartSlog** supports:
 - ▶ high-level KP development
 - ▶ heterogeneous devices

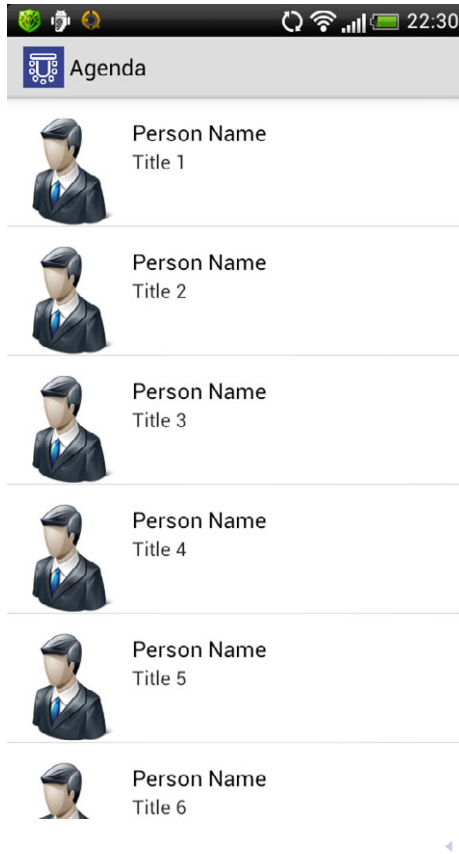


SmartRoom Service Environment

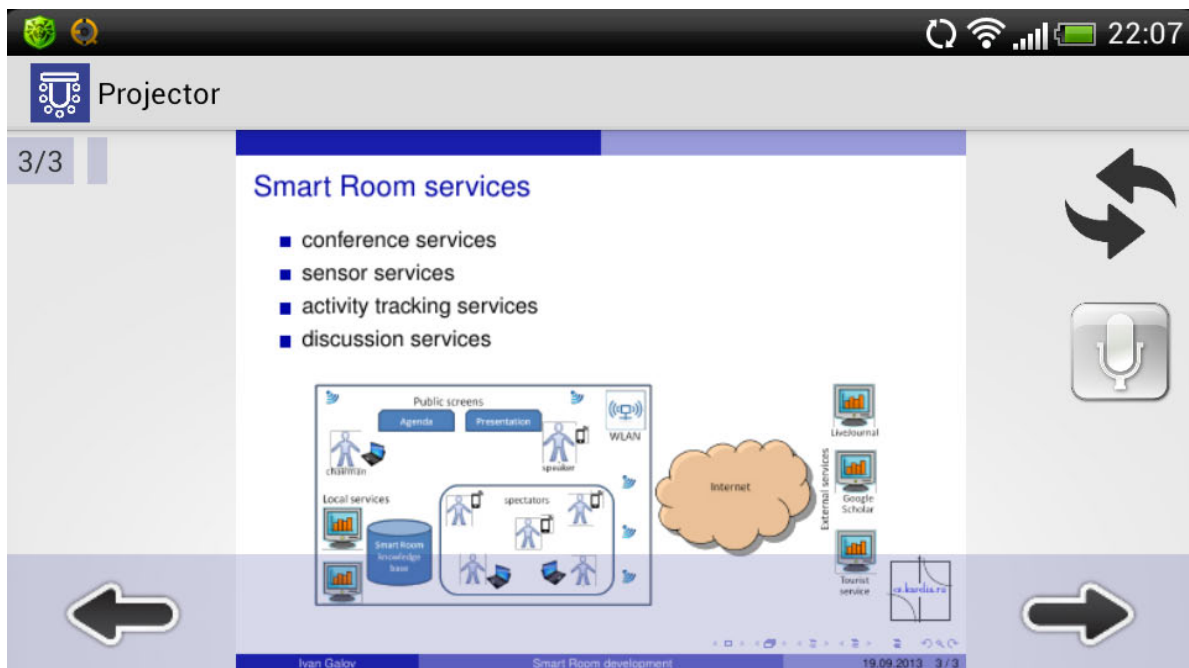


- Many services in SmartRoom
- Many personal clients to access the services
- Many mobile platforms to host the clients

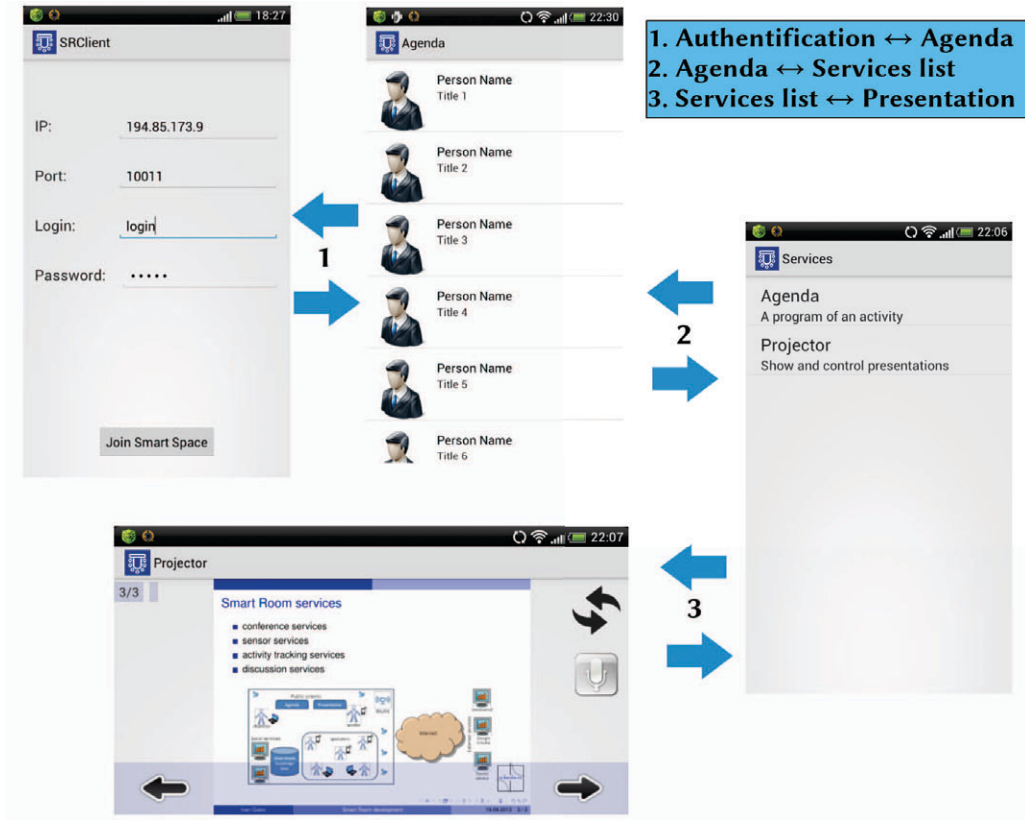
Agenda service



Presentation service

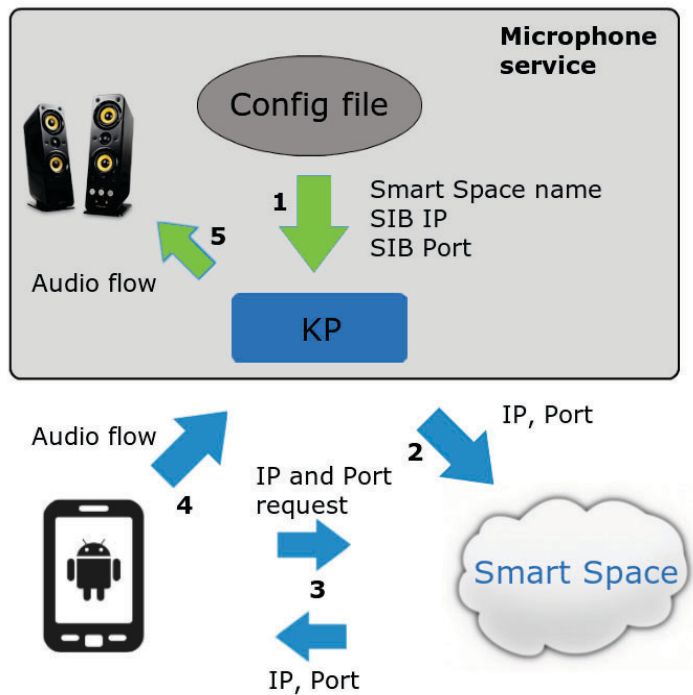


Client scenario



Microphone service

- 1 Initialization
- 2 Publishing address
- 3 Listening for connections
- 4 Receiving data
- 5 Playing audio flow



Conclusion



- Published on Google play

<https://play.google.com/store/apps/>

- Project sources on sourceforge

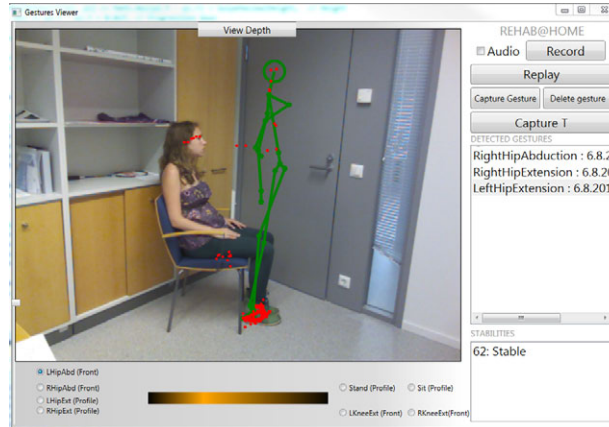
<http://sourceforge.net/projects/smartroom/>

- Project wiki

<http://oss.fruct.org/wiki/SmartRoom>

M3 Case Examples

M3 interoperability for remote rehabilitation with Kinect



Natalia Díaz Rodríguez¹, Stefan Gröroos¹, Franck Wickström¹, Petteri Karvinen¹, Anders Berg¹,
Shohreh Hosseinzadeh¹, Marion Karppi², Johan Liljus¹

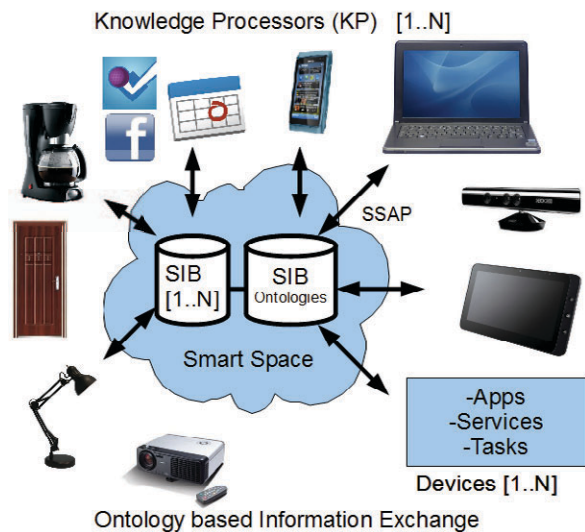
¹Turku Centre for Computer Science (TUUS), Åbo Akademi University, Turku, Finland

²Turku University of Applied Science, Well-being Services

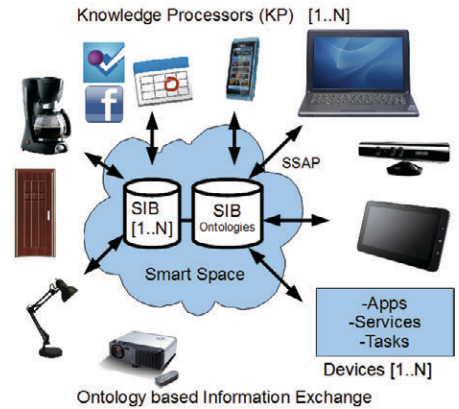
Open International M3 Semantic Interoperability Workshop 12.11.13

Introduction

- Context:
Ambient Intelligence and Smart Spaces
- A crucial task:
Human behaviour modelling and recognition
- Technology for detection of in home activities
- Elderly care



Introduction



- Context information:
 - stress, heart rate, sleep quality, mood, etc.
- Semantic Technologies and Ontologies:
 - Independent knowledge sharing minimizing redundancy.
- Use case: Rehabilitation @home
- Input: 3-D Depth sensor
 - body movement and interaction



Related Work: exercising at home

- Different exercise applications based on 3D depth cameras
 - Virtual Social Gyms
 - Physical therapy
 - Eyes-Free Yoga



[\[http://mobihealthnews.com/22351/slideshow-7-startups-using-microsoft-kinect-for-online-physical-therapy/\]](http://mobihealthnews.com/22351/slideshow-7-startups-using-microsoft-kinect-for-online-physical-therapy/)



Proposal: Rehab@Home Application

- Aim:
 - Recognize & Monitor simple orthopedic exercises
 - After **shoulder, hip or knee surgery**.
 - **Sit-Stand** exercise.
 - Give feedback
 - To the **patient**: on quality & frequency of the exercise. Real-time.
 - To the **physiotherapist**: remotely
- 3 Ontologies:
 - AHA platform with heterogeneous sensor data
 - Privacy and security ontology
 - Kinect Ontology¹

¹To be presented in UCAml'13: *Understanding Movement and Interaction: an Ontology for Kinect*.

5

Implementation

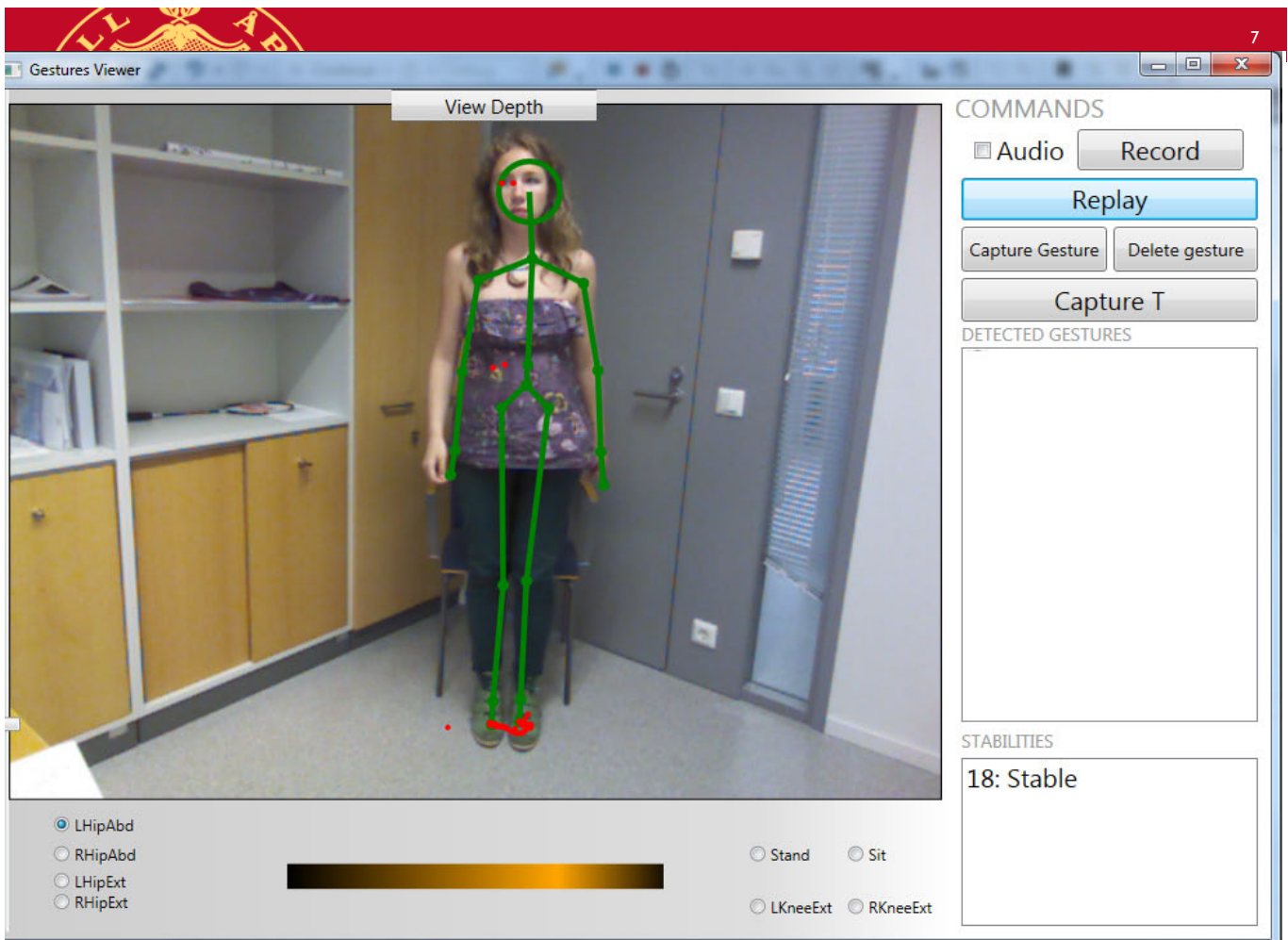
- Tracking & recognition: Kinect for Windows SDK C#.
- Postures & session info: input to M3 (visualized in *WebSIBExplorer*).
- Aim: run in ARM processor
 - Atm: Kinect drivers don't allow it => we use linux Atom board running M3.
 - Dual core Atom N2800 @ 1.86GHz 4 GB RAM SSD storage (Power consumption: 16W, ~12W in idle)
 - Future: Odroid cluster (~24W, 8W in idle).
 - Efficient cluster, low-power, persistent & distributed M3 architecture (for now RedSIB)
- Gathering sensor info, allows execution of semantic queries (SPARQL) and further knowledge reasoning (e.g. long term evolution of back posture).

6

User Interface

Allows feeding patterns from new users realizing exercises for the system to learn:

- **Record and Replay:** Records a session for training the system. Audio option activates and ends recording via voice (“Record”, “Stop”)
- **Stabilities:** Indicates the degree of stability of the skeleton tracked.
- **Capture and Delete Gesture:** Adds (and deletes) a template gesture to a gesture learning model.
- **Capture T:** adds a template posture to a posture learning model.
- **View Depth/View color:** Shows depth/color image
- **Exercises to be trained & recognized:**
 - In FRONT position with the camera:
 - **Left and Right Hip Abduction**
 - **Left and Right Knee Extension**
 - In PROFILE position with the camera:
 - **Left and Right Hip Extension**
 - **Sit and Stand**



Gestures Viewer

View Depth

REHAB@HOME

Audio

DETECTED GESTURES

RightHipAbduction : 6.8.2
 RightHipExtension : 6.8.20
 LeftHipExtension : 6.8.201

STABILITIES

62: Stable

LHipAbd (Front) RHipAbd (Front) LHipExt (Profile) RHipExt (Profile)

Stand (Profile) Sit (Profile)

LKneeExt (Front) RKneeExt(Front)

Gestures Viewer

View Depth

REHAB@HOME

Audio

DETECTED GESTURES

RightHipAbduction : 6.8.2
 RightHipExtension : 6.8.20
 LeftHipExtension : 6.8.201

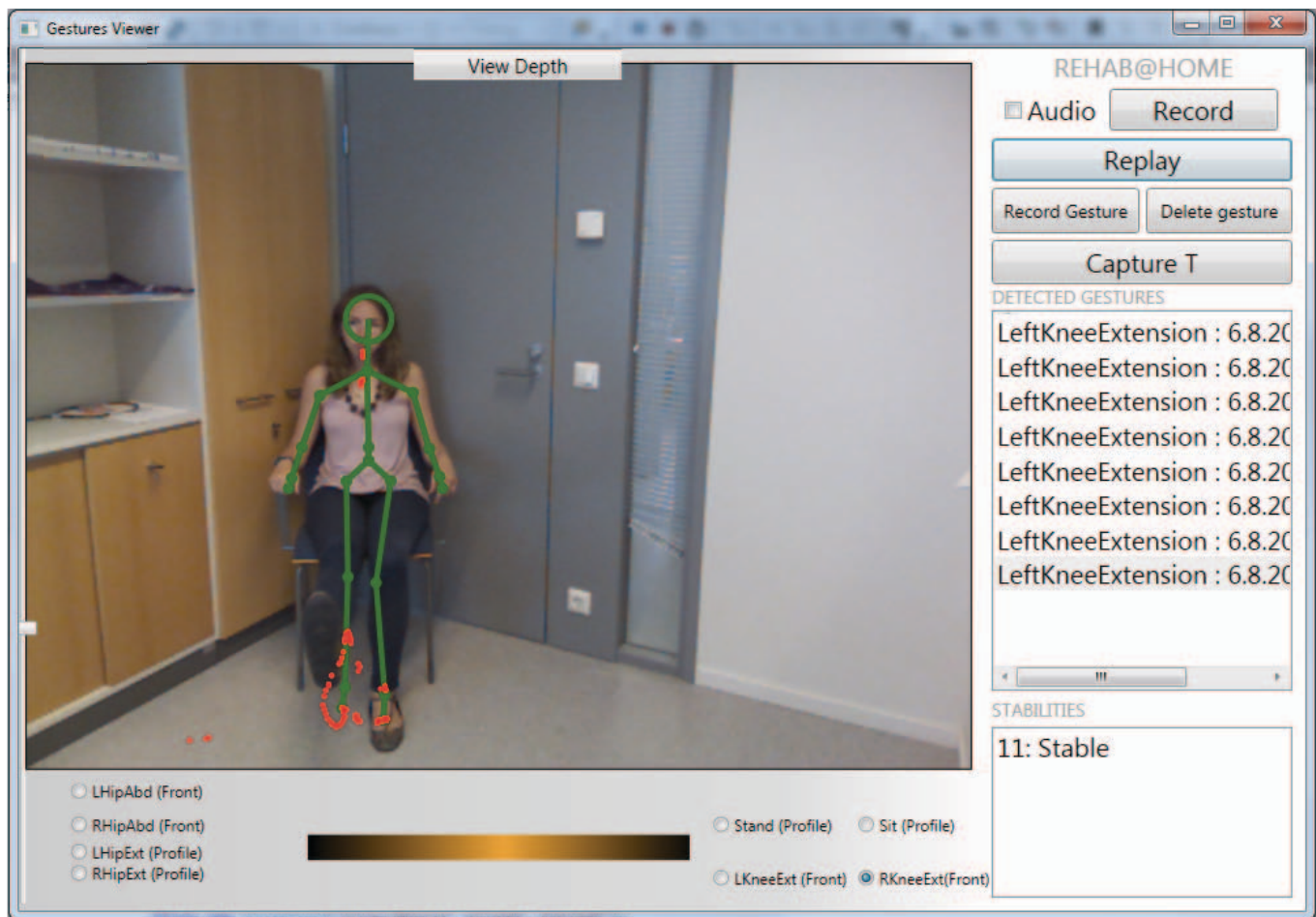
STABILITIES

62: Stable

LHipAbd (Front) RHipAbd (Front) LHipExt (Profile) RHipExt (Profile)

Stand (Profile) Sit (Profile)

LKneeExt (Front) RKneeExt(Front)

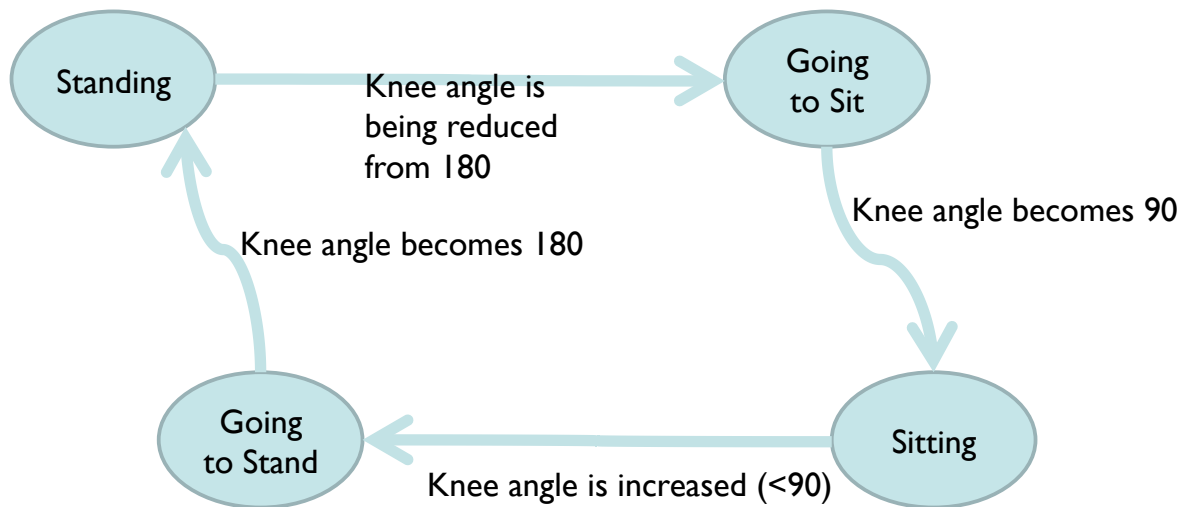


5 times Sit to Stand Test (*)

- Metaanalysis results demonstrated that individuals with times for 5 repetitions of this test exceeding the following can be considered to have worse than average performance (Bohannon, 2006)
 - 60–69 y/o 11.4 sec
 - 70–79 y/o 12.6 sec
 - 80–89 y/o 14.8 sec.
- E.g. 11.4 seconds => score of 100. This approach requires measuring exactly 5 times

(*) [\[http://web.missouri.edu/~proste/tool/5x-STS.rtf\]](http://web.missouri.edu/~proste/tool/5x-STS.rtf)

Sit-Stand Regular Automaton



13

Sit-Stand Test

- Demo:

<https://www.dropbox.com/s/tuyfu8688dqpw7h/Rehab%40Home.mov>

- Start session: touch your head ONCE until you hear the tin sound ONCE.
- Stop session: touch your head again ONCE until you hear the tin sound TWICE.

14

Challenges

- OpenNI C++ and Kinect for Windows SDK (C#): different skeletons
- Bone joint path-based learning machine: not accurate enough, ++ false positives.
- -> We add angle-based Learning Machine, for pose independent gesture recognition
 - Increments computational complexity and recognition speed.

15

Challenges

- Training: Recorded depth video of ~1 min: ~5GBs!!
- In general, hard to make it robust (+ sensors would be helpful)
- *Sitting mode*: ignores lower limbs. [Knee extension in standing mode: computationally more complex but not impossible].
- Voice commands (to start/stop recording) work sometimes.

16

Conclusions

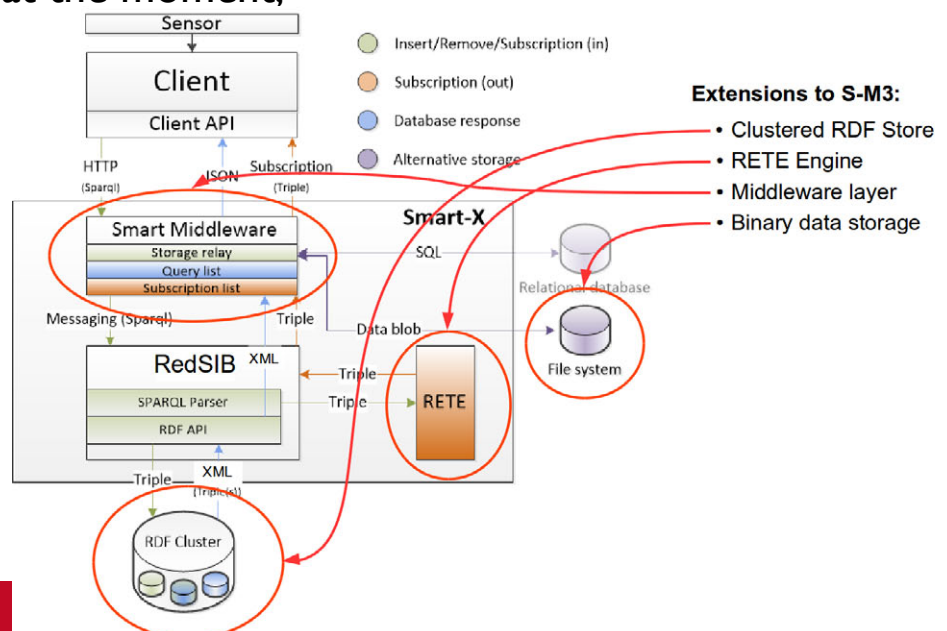
- Software allows
 - Recording new patterns from different users to train the system.
 - Recognize simple exercises
 - *Hip Extension*
 - *Hip Abduction*
 - *Sit-Stand*
 - *Knee Extension*
 - *Knee Abduction*
 - Insert data into M3 for further annotation, tracking and inference

Future Directions

- Integration into new M3 distributed architecture to provide low power distributed processing

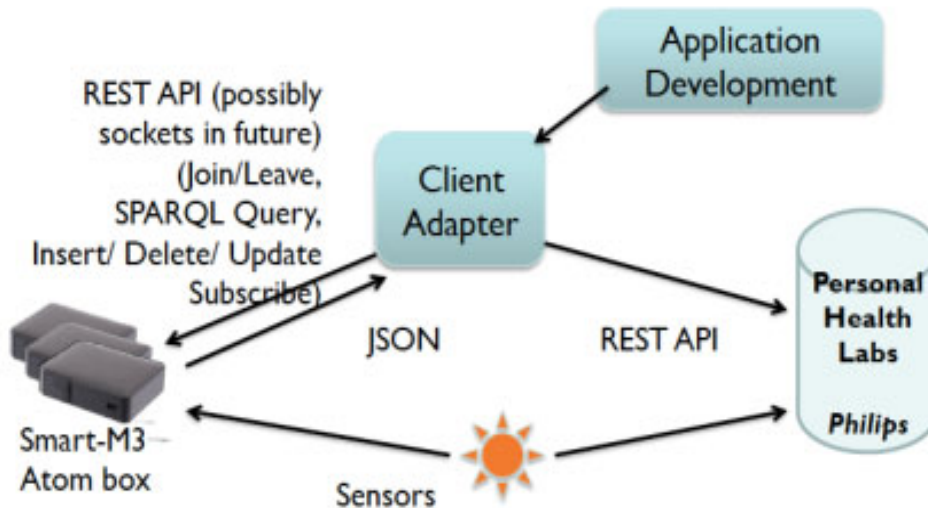
(Atom board at the moment,

Future aim:
ARM)



Future Directions

- Integration into Philips PHL (Personal Health Labs) platform



Future Directions

- Tackling feedback
- Person (face/height/width) recognition for work-out session logging
- Ontologies refinement (with Philips, Fraunhofer and INRIA Grenoble and DFKI)
- Scalability + performance experiments (reasoning, querying/ updating/ subscribing VS ontology size)
- Fuzzy rules to tackle imprecision, vagueness & uncertainty in knowledge representation
- Development of a *Gesture Definition Markup Language (GDML)* to specify exercises formally.

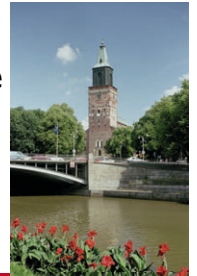
Thank you for your attention!

More info: Natalia D'az Rodr'guez

ndiaz@abo.fi

Embedded Systems Lab. Department of Information Technologies
Åbo Akademi University, Turku, Finland
TUCS (Turku Centre for Computer Science)

Department of Computer Science and Artificial Intelligence
University of Granada, Spain



Smart-M3 and Geo2Tag Platforms Integration Agent

Kirill Krinkin

St.-Petersburg State University of Aerospace
Instrumentation
Saint-Petersburg, Russia
kirill.krinkin@fruct.org

Kirill Yudenok

Saint-Petersburg Electrotechnical University
Saint-Petersburg, Russia
kirill.yudenok@gmail.com

Abstract—Smart Systems provide novel enabling functionalities and as such are currently a driving force behind product innovation. Smart Systems are therefore crucial for the competitiveness of companies and entire industry sectors. In many ways Smart Systems development will be decisive for solving the big challenges of mankind, such as an aging society, increasing energy demand and environmental problems, etc. Geo-tagging and smart spaces are two promising directions in modern mobile market. Geo-tagging allows to markup any kind of data by geographical coordinates and time. This is the basis for defining geographical context which can be used in different types of applications e.g. semantic information search, machine-to-machine (M2M) interactions. Smart spaces as the basis for seamless distributed communication field for software services provides semantic level for data processing. The paper is targeted to discuss common point of Smart-M3 and Geo2Tag platforms integration agent.

Keywords-component: *geo-tagging; geo-coding; Smart Spaces; Smart System; Smart-M3; Geo2Tag; LBS; GCSS;*

I. INTRODUCTION

Nowadays we have two most promising software trends – location based services and pervasive smart environments (smart spaces). Both of them will be a base for user- and machine- oriented proactive services. Smart spaces should provide continuous distributed semantic data and communication field for software services, which is being run on personal devices and autonomous computers and robots. Most desired feature of coming software is proactiveness and context awareness, i.e. services will be able to adapt to the user's needs and situations and be able to manage decisions and behaviors on behalf of the user [1]. One of the important part of context is location based data. This data is being used for two purposes: for clarifying semantic meaning of queries (when service retrieves the data from smart environment) and for limitation of space of search (usually there is no point to make global search). Geo-coding (or geo-tagging) is the technique of markup real or virtual object by adding geographical coordinates and time. If we consider software, we have only virtual (or digital) objects like media, events, documents etc. So far, smart spaces and geo-tagging systems are being developed mostly separately, there are only few works [2, 3, 4] where software design of smart spaces and geo-tagging integration is discussed.

This paper discussed the Smart System definition, based on the creation of an integrated platform as a part of the device for implementing the basic Smart System properties, Smart System use-cases, its architecture and criteria for the analysis of the constructed system.

II. SMART SYSTEM DEFINITION

Smart Systems defined, as miniature devices, which include the sensing functions ("feelings"), management and control. They are able to describe and analysis the situation, decision making on the basis of available data, intellectual or adaptive manner, thereby performing smart activities. In most cases "smartness" of the system may be associated with autonomous operation on the basis of the control theory, energy efficiency and networking opportunities [5, 6].

Smart System must have the following properties: autonomous, openness, context-awareness, self-organization, purposefulness, proactivity, availability, reliability, portability as well as optional properties, as diminutiveness, energy independence, network connectivity, cooperativeness and others.

Smart System is of two kinds:

- as a part of the space (interconnected devices communication);
- as a part of the device (integrated smart system).

In this paper we discuss the Integrated Smart Systems, i.e., the development of an integrated system for the processing incoming information.

At the moment current Smart System does not allow:

- perform a logical conclusion based on the knowledge (data) of the system;
- make independent decisions based on current statements and situations.

To address these drawbacks, it was decided the Smart System creation in the form of an integrated system within a device, which is responsible for the basic functions of Smart System, such as processing, storage and retrieval of data (knowledge), their management and control, analysis and logical conclusion, decision-making on based on the processed data and others.

In other words we need to design a system, which include components (technologies) to perform main and missing Smart System functions. Each system component is

responsible for the execution own functions and also provides an interface (API). The integrated system should have a common communication interfaces, protocols and programming interfaces for interaction with other Smart Systems.

As an initial system components (technologies) of the device were chosen Smart Spaces [7] Smart-M3 [8, 9] platform and the Internet of Things [10] LBS Geo2Tag [11, 12] platform, as one of the fastest growing platforms of these areas. Smart-M3 platform provides a common communication field for cooperation, as well as allows you to process and store the semantic information (knowledge). LBS platform Geo2Tag responsible for the provision of geospatial data from a variety sources.

Thus Smart System obtained by integrating the technologies in the device can be defined as a device with an integrated system for interaction and control of the acquired information with the expressed “smart” features.

To build a new Smart System based on selected technologies we must solve the following questions:

- integrated system architecture development;
- unification of general platforms levels (Smart-M3 and Geo2Tag);
- common protocols and communication interfaces between device;
- behavior model;

The main feature of the Smart System discussed in this paper is the ability to determine the location of an object in space and time.

III. GCSS HIGH-LEVEL DESIGN AND ARCHITECTURE

On Fig. 1. high-level layered design for GCSS is presented. There are four levels of system. Each level of the system is responsible for the functions and includes its own interface. The following are the layers of the system GCSS:

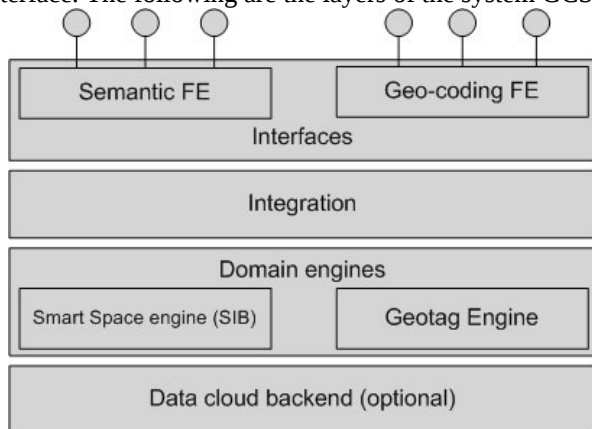


Figure 1. Layered design of GCSS

There are five base components (levels) that provide basic functioning contour of the system (life cycle):

- *data acquisition level* – presented by sensors and other receiving information interfaces from the outside world, a person, other systems;
- *data pre-processing level* – data storage and transformation of the primary form to a form suitable for analysis and decision-making;
- *decision-making level* – module responsible for information processing and making decisions to achieve the goals of the system, and support tasks related to self-diagnosis and self-organization;
- *command level* – responsible for making the transformation into control signals own functional components and external systems for the environmental impact implementation;
- *action level* – implementation of information and physical control of external systems, including the task of encoding and transmitting control signals to run-time systems and control command execution.

IV. GCSS REQUIREMENTS AND USE-CASES

The first main task of the Smart System integration platform - the Smart-M3 and Geo2Tag platforms integration and also it is expanding the smart space with new data – geo-data [13].

Smart System should implement main features from both types of platforms, which are:

- providing interfaces for semantic data and access (Smart-M3 API – Java, Python, Qt);
- distributed storage for semantic information;
- interfaces for association semantic objects with geo-tags; Two-way geo-data conversion mechanism;
- spacial and temporal filtration.

Also non-functional requirements should be taken in account:

- *Performance* – ability to work with big amount of semantic objects geo-tags like cloud based massive offline processing and local context indexing/caching.
- *Compatibility* – the GCSS should be accessible by legacy interfaces (i.e. SSAP or REST), which is required for seamless integration with existing systems.

The latest versions of Smart-M3 and Geo2Tag platforms fairly stable but have some defects such as those associated with security or performance of the Smart-M3 platform.

The functional requirements of the integration agent presented at list below:

- smart space Smart-M3 platform management:
 - o leave / join platform operations;
 - o query/insert/delete/update operations;
 - o subscribe/unsubscribe operations;
- Geo2Tag platform management:
 - o connection/disconnection;

- o obtain platform data;
- o search and filtration platform data;
- geo-tags conversion mechanism to space data (triples);
- algorithms for searching and filtering of space data, for example, by means of SparQL queries;
- ranking mechanism of space data (the algorithm of selection the latest actual objects, optional);

The first three requirements are fulfilled the main features of an agent to increase the space with new information, geo-data, that will be used to determine the location and search for objects in space, of which the first two are available by Smart-M3 and Geo2Tag platforms.

V. GCSS HIGH-LEVEL DESIGN AND ARCHITECTURE

Now the integration agent responsible for the platforms integration and fills the Smart-M3 space with geo-data by conversion mechanism. Next, the agent will combine the functionality of both platforms (Smart-M3 and Geo2Tag) and will become a sort of common platform within the device to control and manage data between all smart space devices [14].

Agent ontology we may create by using special Smart-M3 ontology generator – Smart Slog [15, 16].

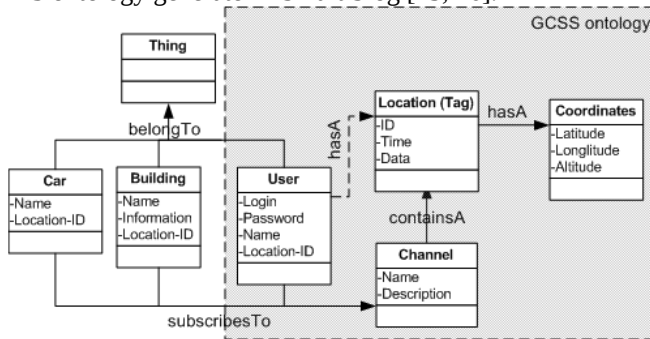


Figure 2. Overlay ontology used by GCSS

GCSS ontology consists of four classes – User class, Channels class, the Tag itself and its Coordinates. Class User is responsible for a user's of the Geo2Tag platform in space, the tags channel describes a set of tags for a given criterion, the Tag class describes itself data. It should be noted that users can subscribe to an unlimited number of channels, as well as a channel can contain unlimited number of tags. Class User can directly communicate with the tag through the property hasA. Coordinates are allocated in a separate class for more convenient their representations in an agent ontology. In more details GCSS ontology is described in [9].

Each Geo2Tag platform user, if it exists, will be associated with own user in the Smart-M3 space, if not, then will be created a new space user, and it will be automatically attached to the tag location and the channels to which it subscribes. Location may be attracted to any space object after adding new propertie (e.g., Location-ID)

in the object class of the space ontology. Class Tag property Data is mainly used for searching and filtering space objects, but it can be also used for its association with the object.

It should be noted that the user location or other space object (not static) can change location with time and in order to remain relevant data necessary to provide handling this situation. Smart-M3 platform provides publish-subscriptions mechanism by subscribing to specific triplets, the object will automatically receive new data after it changing. In our case, these data are the properties of the Coordinates class.

The agent will use the object model of the ontology representation, i.e., have clearly documented ontology classes names and their properties, as well as certain triplets (subscription). Thus, the space agent ontology will look like a list of properties that linked by a predicate. In the first version of the integration agent the space will be filled only with geo-data, which will be linked with their space objects (a person, object, etc.). In the future we plan to expand the space by the addition of the users and channels tags information.

Some details of the work with the agent ontology will be considered in the discussion of the tags conversion mechanism.

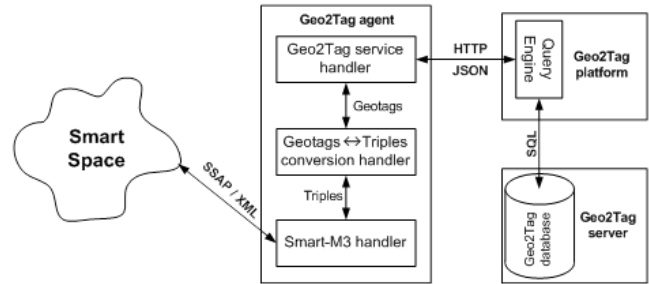


Figure 3. GCSS architecture

All Geo2Tag platform data are stored in a database on a dedicated server. Geo2Tag platform allow you to record and retrieve data using REST-specific queries in JSON format. There are also a variety number of clients to work with a Geo2Tag platform, mainly for mobile platforms.

Integration with the Smart-M3 platform will be implemented through a special mediator (agent). Its main task is to convert data from one platform format (Geo2Tag, JSON) to another format (Smart-M3, XML). As mentioned above, the Geo2Tag platform transmits data in JSON format, this is a text format, but in a more readable form for humans.

The agent consists of three main components:

- Geo2Tag service handler;
- Geotags – Triples conversion handler;
- Smart-M3 handler.

Geo2Tag service handler is responsible for obtaining geo-data, it connects to the server database and requests data using a special class LoadTagsQuery.

Geotags – Triples conversion handler is required to bring data to a convenient form for the triplets creation. Since the data are returned in JSON format, they need to be parsed and pulling the necessary data, namely, time, location and description of the geo-tag by saving them for later processing.

At the last stage is a connection to the space, then creates triplets according to the ontology which then placed to the Smart-M3 space. What the triplets are created and their number are discussed in "Geo-tags Conversion Mechanism".

After the smart system platform development within the device, each of its mechanism (algorithm) will be subject to thorough analysis by the following criteria: universality, performance, resources, the ability to integrate into embedded devices, memory size, the amount of transmitted traffic, response speed [17].

The main evaluation criteria's of the Smart System platform will serve – its performance, the ability to integrate into embedded devices, the amount of transmitted traffic and response speed. The analysis should show how the platform behaves in the real conditions and only then take steps to improve its operability.

CONCLUSION

In this paper we proposed description of the smart system based on common device platform, discussed smart system requirements and use-cases, high level design and architecture of smart-space and geo-coding middleware integration. This integration could be made by using special Smart-M3 Knowledge Processor, which monitors both spaces and translates data from one to another and vice versa.

At the current moment we develop integration agent with main use-cases analysis, and then start to think over system protocols and interfaces.

The next step in the development of smart systems device platform is the complete platform components integration, define common protocols and interfaces for communicating between all devices. There are still open questions for future development: overall system performance, effective object monitoring, temporal and spatial filtration, integration with media objects.

ACKNOWLEDGMENT

The authors would like to thank Finnish Russian University Cooperation in Telecommunication Program for provided equipment and support.

REFERENCES

- [1] J.Floch, M. Angermann, E. Jennings, M. Roddy.: Exploring Cooperating Smart Spaces for Efficient Collaboration in Disaster Management, Proceedings of the 9th International ISCRAM Conference – Vancouver, Canada (2012)
- [2] Nabian, N., Ratti, C., Biderman, A. and Grise, G.: MIT GEOblog: A platform for digital annotation of space for collective community based digital story telling. 3rd IEEE International Conference on Digital Ecosystems and Technologies, Piscataway, N.J.: IEEE: 353-358 (2009)
- [3] J. Rishede T. Man L. Yiu.: Effective Caching of Shortest Paths for Location-Based Services, SIGMOD '12, Scottsdale, Arizona, USA (2012)
- [4] K. Kolomvatsos, V. Papataxiarhis, V. Tsetsos.: Semantic Location Based Services for Smart Spaces", 2nd International Conference on Metadata and Semantics Research (MTSR), Corfu, Greece (2007)
- [5] Smart System (wikipedia) – http://en.wikipedia.org/wiki/Smart_system
- [6] EPoSS Strategic Research Agenda 2009 – <http://www.smart-systems-integration.org/public/documents/publications/>
- [7] D. J. Cook, S. K. Das, "How smart are our environments?" an updated book look at the state of the art. Pervasive and Mobile Computation 3(2), 53-73 (2007)
- [8] Smart-M3 (wikipedia) – <http://en.wikipedia.org/wiki/Smart-M3>
- [9] J. Honkola , H. Laine, R. Brown, O. Tyrkkö, "Smart-M3 Information Sharing Platform", 1st Workshop on Semantic Interoperability in Smart Spaces (2010)
- [10] Internet of Things (wikipedia) – http://en.wikipedia.org/wiki/Internet_of_Things
- [11] Geo2Tag – <http://www.geo2tag.org>
- [12] I. Bezyazychnyy, K. Krinkin, M. Zaslavskiy, S. Balandin, Y. Koucheravy.: Geo2Tag Implementation for MAEMO, 7th Conference of Open Innovations Framework Program FRUCT, Saint-Petersburg, Russia (2010)
- [13] Kirill Krinkin, Kirill Yudenok. Geo-coding in Smart Environments: Integration Principles of Smart-M3 and Geo2Tag. In Proceedings of the 13th International Conference, NEW2AN 2013 and 6th Conference, ruSMART 2013, St. Petersburg, Russia, August 28-30, 2013. Proceedings. Springer 2013 Lecture Notes in Computer Science, p. 107-116. ISBN 978-3-642-40315-6
- [14] D. Korzun, I. Galov, A. Kashevnik, N. Shilov, K. Krinkin, Y. Korolev. *Integration of Smart-M3 Applications: Blogging in Smart Conference*. Proc. 4th Conf. Smart Spaces (ruSMART 2011), Saint-Petersburg, Russia, 22-23 August 2011, pp.51-62.
- [15] D. Korzun, A. Lomov, P. Vanag, J. Honkola, S. Balandin. *Generating Modest High-Level Ontology Libraries for Smart-M3*. Proc. 4th Int'l Conf. Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2010). N. 103–109.
- [16] D. Korzun, A. Lomov, P. Vanag, J. Honkola, S. Balandin. *Multilingual ontology library generator for Smart-M3 information sharing platform*. International journal on Advances of Intelligent System 4 (3&4), 68-81, 2011.
- [17] M. Zaslavsky, K.Krinkin.: Geo2tag Performance Evaluation, Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Travel, Oulu, Finland (2012)



Smart-M3 and Geo2Tag Platforms Integration Agent

EIT Smart-M3 workshop

*Kirill Krinkin, FRUCT OSLL, ETU
Kirill Yudenok, FRUCT OSLL, ETU*

FRUCT 14, Helsinki, 12 November 2013

Agenda

- Smart System;
- Goal and Tasks;
- Geo-coding and smart spaces;
- Smart-M3 and Geo2Tag platforms;
- Geo-coded Smart Space (GCSS);
- GCSS layered architecture;
- GCSS high-level requirements;
- GCSS use-cases;
- GCSS agent ontology and architecture;
- Future steps.

Smart System

1/2

Smart System called intellectual integrated system, which has the following main features:

- system with a clear goal, which determines the directionality of the system;
- system receiving information from the outside world, a person, other systems (data acquisition);
- system responsible for processing information and making decisions to achieve the goals of the system.

Smart System properties:

- Autonomy;
- Openness;
- Context-awareness;

3

Smart System

2/2

- self-organization;
- purposefulness;
- proactivity;

and **optional** features:

- diminutiveness;
- energy independence;
- network relatedness;
- cooperativeness.

Two kinds of Smart System:

- as a part of the space (interconnected devices communication);
- as a part of the device (integrated smart system).

4

Goal and Tasks

Goal:

1. Smart Spaces and Geo-coding platforms integration problem;
2. Solve "*intellectual problem*" of the current Smart Systems, that does not allow:
 - perform a logical conclusion based on the knowledge;
 - make independent decisions based on current statements and situations.

Tasks:

1. Develop integration agent of Smart-M3 and Geo2Tag platforms;
2. Develop Smart System in form of an integrated system within a device;
3. Suggest common platform protocols and interfaces. **5**

Geo-coding and smart spaces

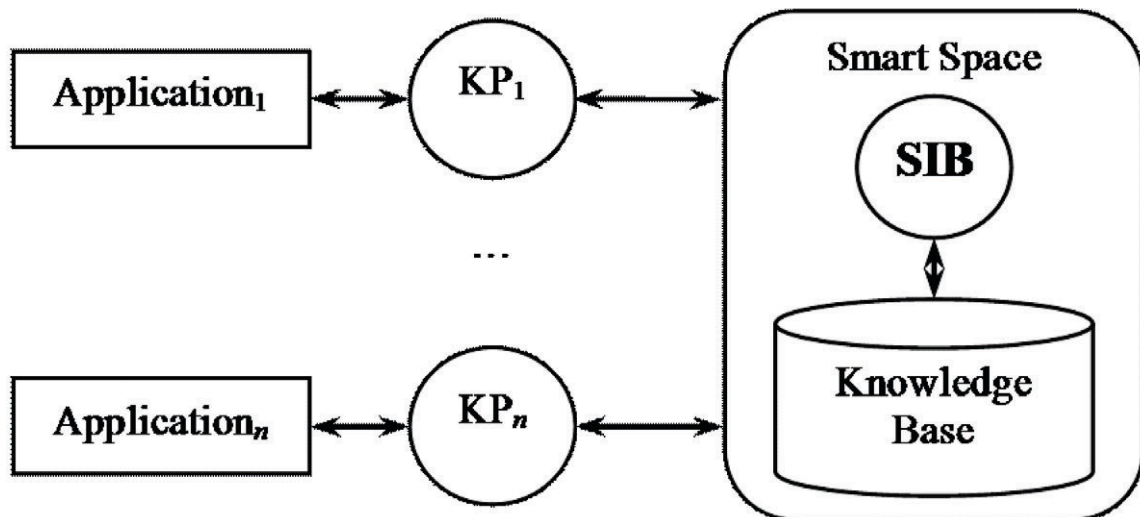
1. *Geo-tagging* allows to markup any kind of data by geographical coordinates and time (media, events, documents, etc.):
 - semantic information search;
 - M2M interactions;
2. *Smart Spaces* as the basis for seamless distributed communication field for software services provides semantic level for data processing:
 - continuous distributed semantic data;
 - communication field for software services;

As a geo-tagging and smart spaces areas platforms serve *Geo2Tag* and *Smart-M3* platforms accordingly.

6

Smart-M3 platform

Smart-M3 is an open source software platform that aims to provide Semantic Web information sharing infrastructure between software entities and various types of devices.



7

Geo2Tag platform

Geo2Tag platform is the centralized high performance geo-tagging database with dedicated server, which is provided REST API for access geo-tags.

- user management: registration, login, log-off, sessions;
- data retrieval about users and matching personal geographical spaces to the personal smart spaces;
- channel management: subscription/unsubscription;
- sending geographical data from smart-space to the geo-tagging system;
- getting data from geo-tagging system;
- spatial filtration;

8

Geo-coded Smart Space (GCSS)

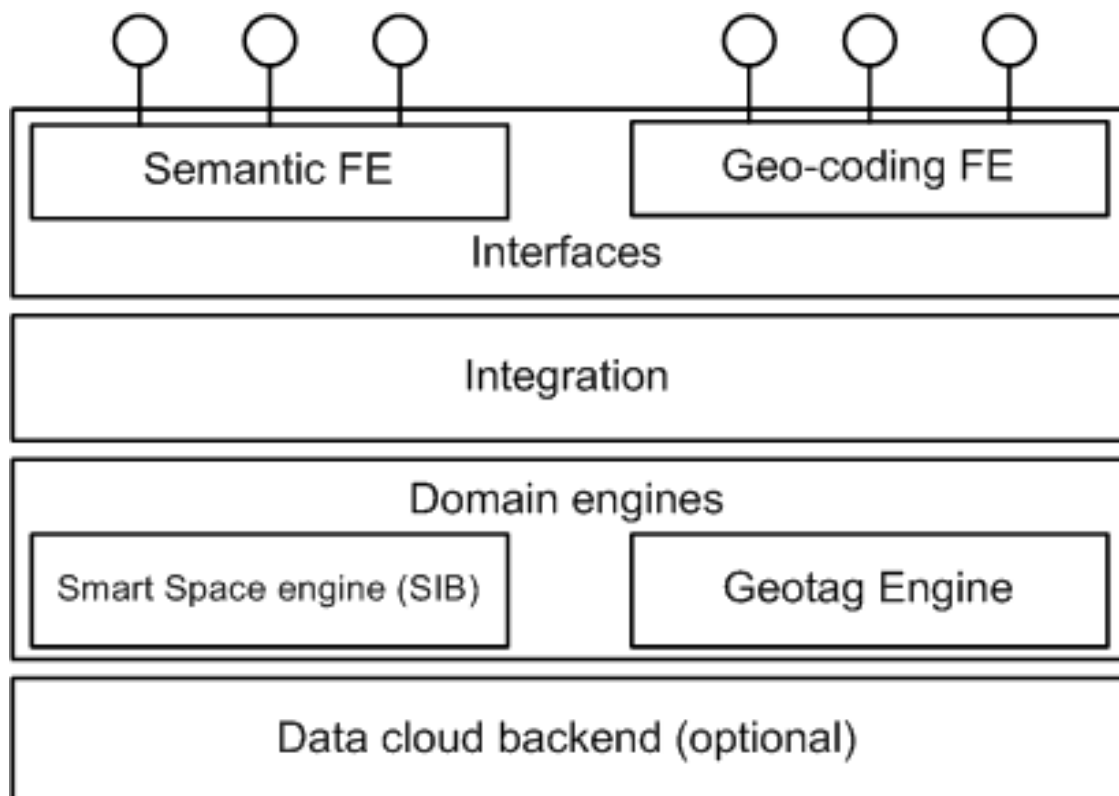
GCSS is an integration agent that looks like Smart-M3 agent (KP) by combining the work of two platforms *Smart-M3* and *Geo2Tag*.

- Main difference from previous works: use common platform for knowledge processing of Smart-M3 space;
- New property: location in space and time, that give the ability to search in a space (room, house).
- Agent is not an extension of the Smart-M3 platform, it is expanding the space with new data – geo-data;

9

GCSS layered architecture

1/2



10

- **Interfaces** – smart-spaces and geo-coding front-ends (FE) and their functionality;
- **Integration** – components for translating geo-tags from Geo2Tag format to Smart-Space format and vice versa;
- **Domain engines** – particular implementations of smart-space geo-coding middleware (Smart-M3 and Geo2tag);
- **Data cloud backend** – optional component, which providing services: offline data (pre-)processing, storage for BLOB objects, indexing, caching etc.

11

GCSS high-level requirements

- providing interfaces for semantic data and access;
- distributed storage for semantic information;
- interfaces for association semantic objects with geo-tags;
- spatial and temporal filtration.

Non-functional requirements:

- performance (cloud based massive offline processing and local context indexing/caching);
- compatibility (legacy interfaces: SSAP, REST);

12

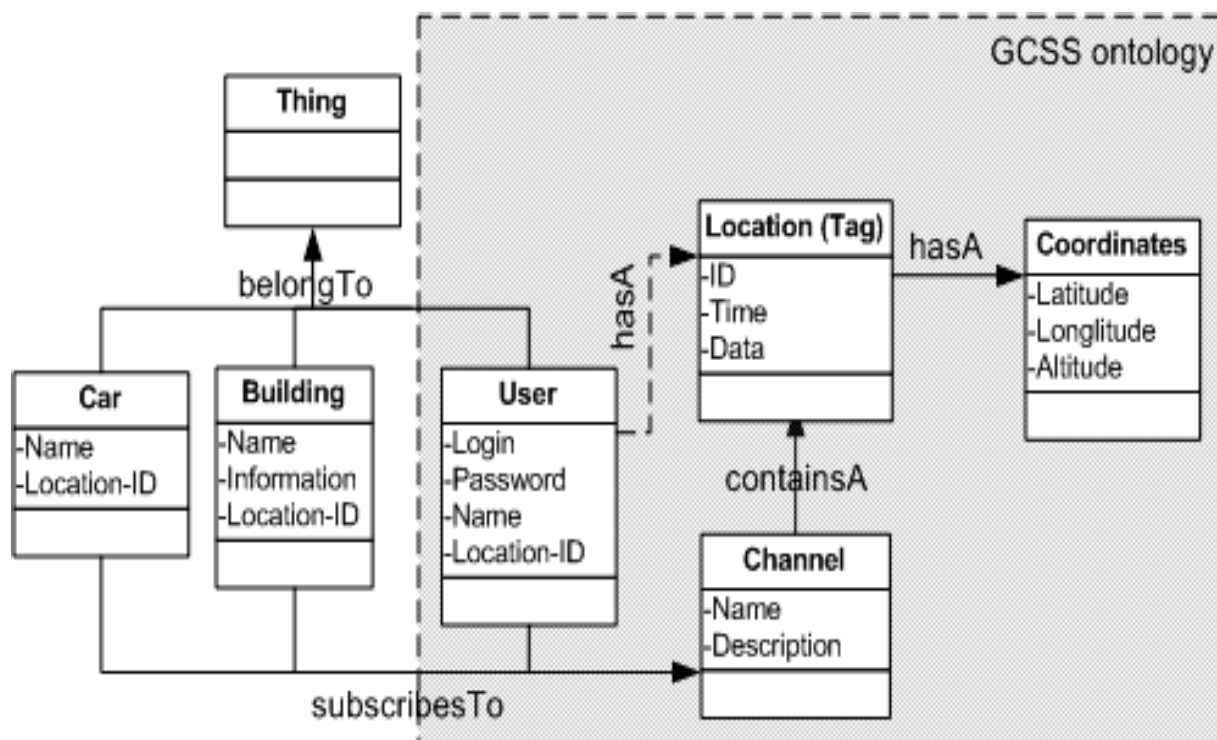
GCSS use-cases

1. smart space Smart-M3 platform management:
 - leave / join platform operations;
 - query / insert / delete / update operations;
 - subscribe / unsubscribe operations;
2. Geo2Tag platform management:
 - connection / disconnection;
 - obtain platform data;
 - search and filtration platform data;
3. geo-tags conversion mechanism to space data;
4. algorithms for searching and filtering of space data, for example, by means of SparQL queries;
5. ranking mechanism of space data (the algorithm of selection the latest objects by location, optional);

13

GCSS agent ontology

1/2



14

GCSS agent ontology

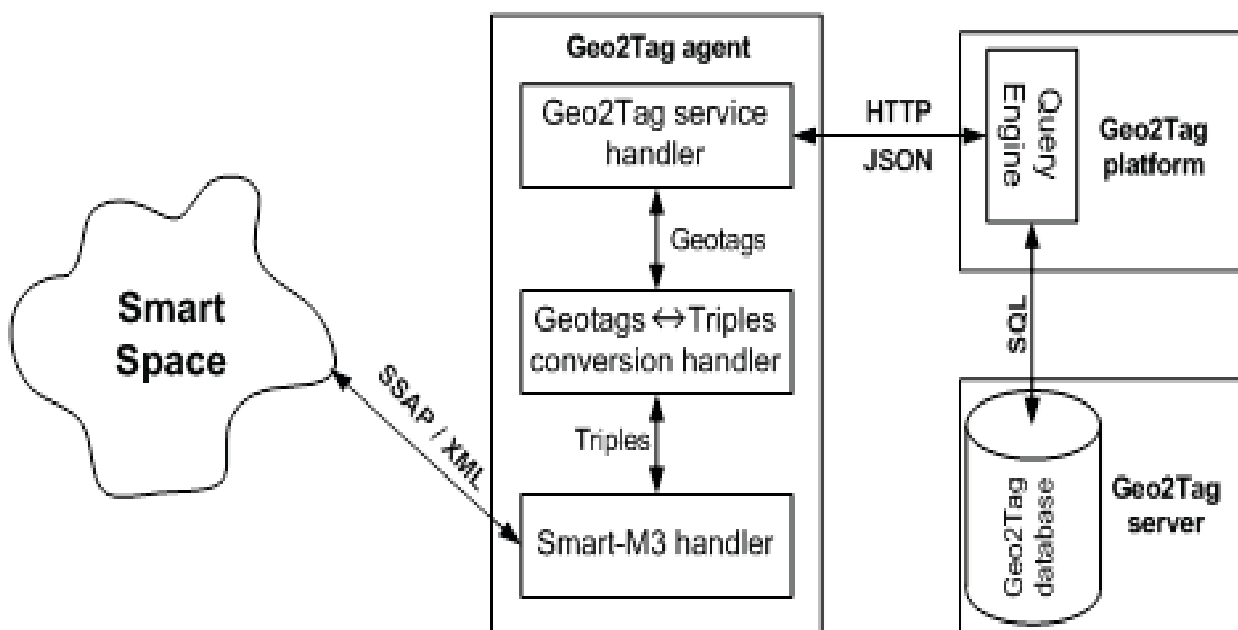
2/2

- Consist of 4 classes: User, Channel, Tag and Coordinates;
- Each Geo2Tag user associated with own user in space, if it is not exist, then created new Geo2Tag space user, that automatically attached with own location information;
- Class Tag property Data is mainly used for searching and filtering space objects, but it can be also used for its association with the an object;
- If space object location changed, we must only subscribe or Coordinates class triples to update it;
- GCSS agent use the object model of the ontology representation.

15

GCSS agent architecture

1/2



16

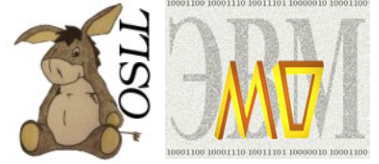
- All Geo2Tag data are stored in a database on a dedicated server, that queried in JSON format;
 - Integration will be implemented through a special mediator (agent);
 - Agent main task is conversion geo-data to space triples;
 - Agent main components: Geo2Tag service handler, Geotags – Triples conversion handler, Smart-M3 handler;
- 17**

Future steps

- complete geo-coding and smart spaces platforms integration;
- common protocol and interfaces for communication between all devices;

Open questions:

- overall system performance;
- effective object monitoring;
- temporal and spatial filtration;
- integration with media objects.



Q&A

Kirill Krinkin {kirill.krinkin@fruct.org}

Kirill Yudenok {kirill.yudenok@gmail.com}

FRUCT & ETU OSLL {<http://osll.fruct.org/>}

Recommendation System for Tourist Attraction Information Service

Alexander Smirnov, Alexey Kashevnik, Andrew Ponomarev, Nikolay Shilov, Maksim Schekotov, Nikolay Teslya
St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
St.Petersburg, Russia
{smir, alexey, ponomarev, nick, shekotov, teslya}@iias.spb.su

Recently, the tourist business has become more and more popular. More and more tourists prefer to use Internet services to book hotels, buy flights, and search attractions to see instead of booking complete tours. In this regard, information retrieval systems, which allow finding information about the tourist trip and provide the tourist interested information during the trip, are becoming more and more popular. The most valuable systems support online information searching in different Internet sources instead of using local information databases. However, such systems have to personify and implement the context-based filtering of information before providing it to the tourist. These systems have to recommend the tourist, which attractions are better to attend, and identify, which information about attraction is better to be shown to the tourist.

The presented recommendation system for tourist attraction information service allows recommending attractions, which are better to attend, based on the tourist preferences and the context information of the location area. The system allows the tourist to see detailed description of the interesting attraction acquired from Internet sources. Recommendation system chooses an Internet source that provides description of the interested attraction based on other tourists ratings.

The system is service-based and uses the smart space technology, which allows providing for information sharing between different services of the system.

The recommendation system described in the paper is developed as a distributed service-oriented application that allows tourists to get useful information using mobile client (Fig. 1).

The system accumulates tourist's interests and context-related information and searches for accessible at the moment and interesting for the tourist attractions based on this information.

Implementation of the recommendation system is based on the Android OS that is one of the most popular operating systems for mobile devices today. For interoperability support between mobile clients and different services the smart space technology is used, which allows providing for information sharing between different devices. For this purpose the Smart-M3 information sharing platform is used, which provides implementation of the smart space technology. The key idea of this platform is that the smart space is device, domain, and vendor independent. Smart-M3 assumes devices and software entities to be able to publish their embedded information for other devices and software entities through simple, shared information brokers. The platform has a decentralized architecture and allows seamless integration with other systems, services, and program modules.

Modern tendencies of information and communication technologies require development of stable and reliable infrastructures to extract and keep different kinds of information and knowledge from various members of the smart environment. The smart space assumes more than one device that uses common resources and services.

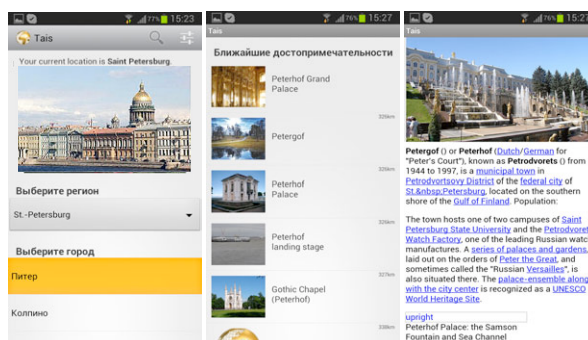


Fig. 1. Tourist attraction information service: screenshots

In accordance with the developed architecture, the recommendation system consists of knowledge processors and system information broker. Knowledge processors include a client module and different services. The client module is implemented

in the user mobile device and provides possibilities for interaction with the tourist. The client module acquires the tourist context and keeps his/her profile, that includes the tourist preferences for the system personification and usability.

Available services include: the recommendation service, the context service, the attraction information service, the public transport service, the ridesharing service, and the taxi service. The recommendation service implements information filtering that is provided for the tourist by other services. The context service acquires and provides information about current situation in the area (e.g. location, traffic situation, weather). The attraction information service searches for information around the tourist's location in different Internet resources (like Wikipedia, Wikivoyage, Wikitravel, Panoramio, Flickr). The public transport service finds ways to reach an attraction by public transport. The ridesharing service tries to find drivers, who move in the same direction and to determine possibilities of joint trips. The taxi service provides information about accessibility and prices for taxi for tourist trip to an attraction.

Service interaction scenario of recommendation system is presented in Fig. 2. The client module publishes information about the tourist context and preferences in the smart space. The context service reads this information and publishes information about the current situation in the area of tourist location. The attraction information service publishes the list of attractions that meets the tourist context and preferences and reaching possibility meets the current situation in the location area. The recommendation service reads information about attractions, tourist preferences, and context from the smart space, gets information from the rating service, and generates a list of most appropriate for the tourist at the moment attractions related to the his/her interests. Based on information from the transport service, the recommendation service analyzes which attractions and in which order can be proposed to the tourist for visiting. Then, the recommendation service publishes this information in the smart space and it becomes accessible for the client module, which presents it to the tourist.

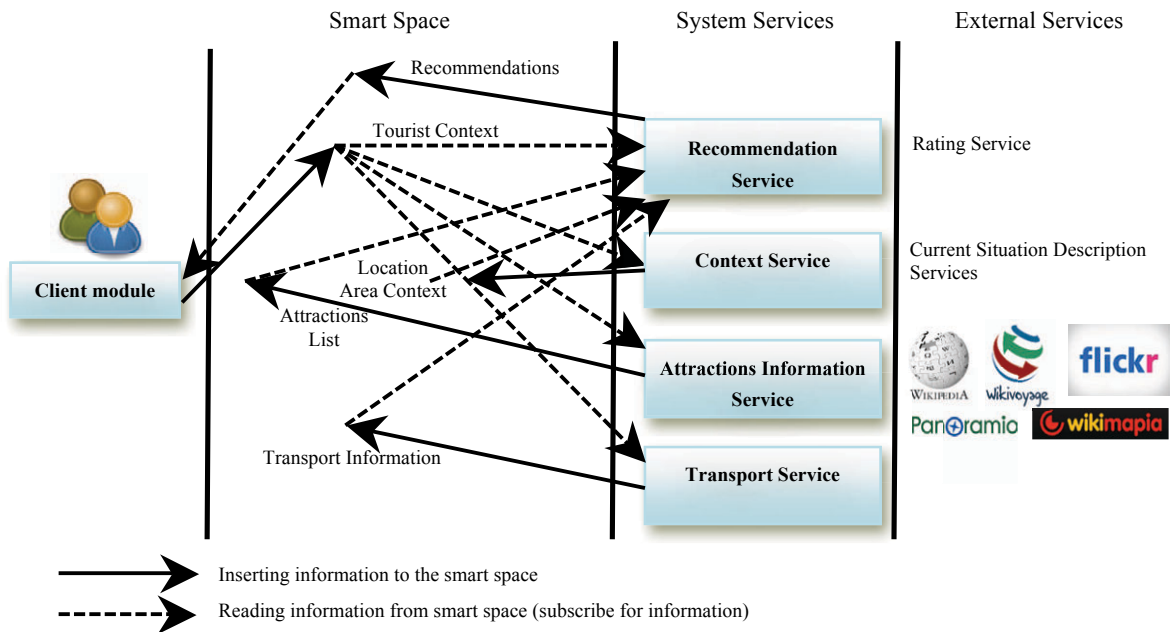


Fig. 2. Tourist attraction recommendation system services interaction based on smart space technology

The recommendation service reads information about attractions, tourist preferences, and context from the smart space, gets information from the rating service, and generates a list of most appropriate for the tourist at the moment attractions related to the his/her interests. Based on information from the transport service, the recommendation service analyzes which attractions and in which order can be proposed to the tourist for visiting. Then, the recommendation service publishes this information in the smart space and it becomes accessible for the client module, which presents it to the tourist.

The recommendation system allows the tourist to browse the attractions' descriptions for making a decision about visiting them. In this case, the client module publishes the information that the tourist is interested in an attraction. The attraction service gets this information, searches for descriptions of this attraction, and publishes links to information sources with these descriptions to the smart space. The recommendation service analyzes these information sources, user preferences, ratings made by other tourists for these sources and shares description (or descriptions) that are better for the tourist at the moment with the client module.

Recommendation System for Tourist Attraction Information Service

Alexander Smirnov, Alexey Kashevnik, Andrew Ponomarev,
Nikolay Shilov, Maksim Schekotov, Nikolay Teslya

St.Petersburg Institute for Informatics and Automation of the
Russian Academy of Sciences (SPIIRAS)

Table of Contents

- Motivation
- Tourist Attraction Information Service Overview
- Tourist Attraction Recommendation System
- Smart Space Communication
- Recommendation Scenarios
 - Attraction Recommendation
 - Attraction Images Filtering
 - Attraction Descriptions Filtering
- Conclusion

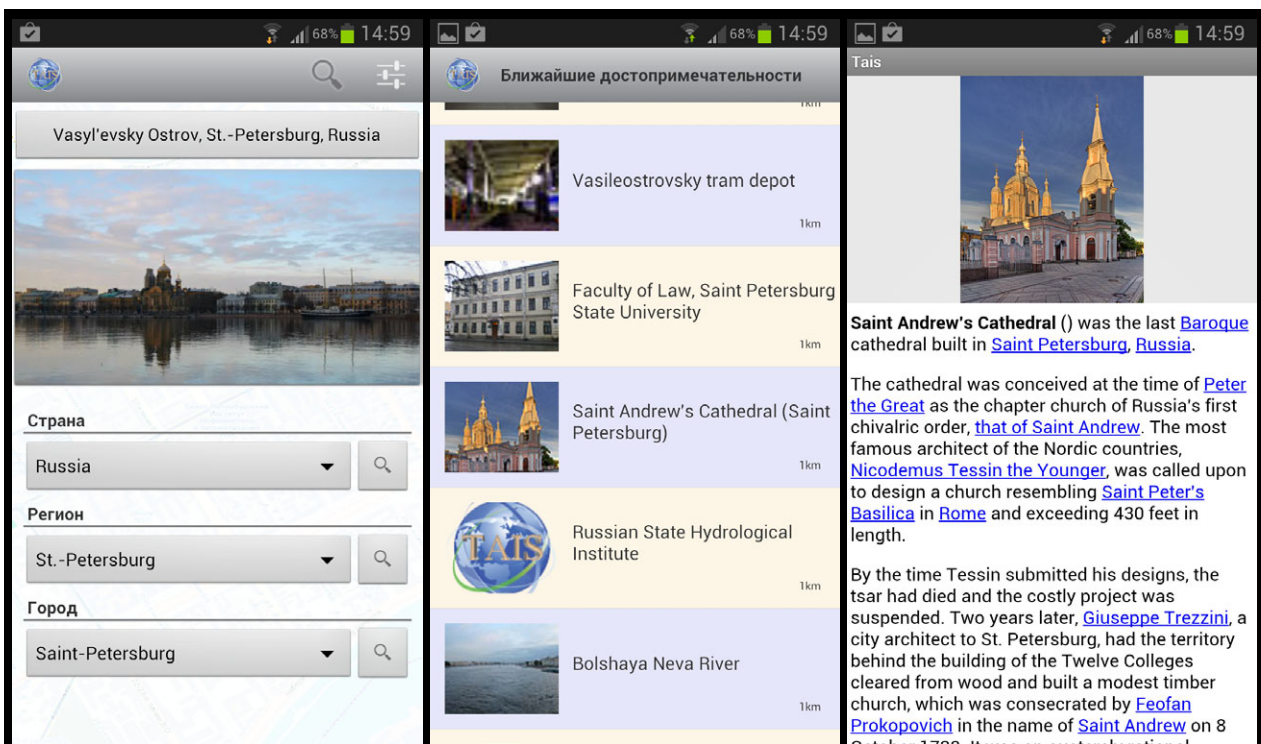
Motivation

- The tourist business has become more and more popular.
- More and more tourists prefer to use Internet services to book hotels, buy flights, and search attractions to see instead of booking complete tours.
- Tourist attraction information service
 - Finds attractions nearby the tourist
 - Provides descriptions of attractions acquired from different Internet Sources
 - Provides accessible in the Internet images of attractions



3

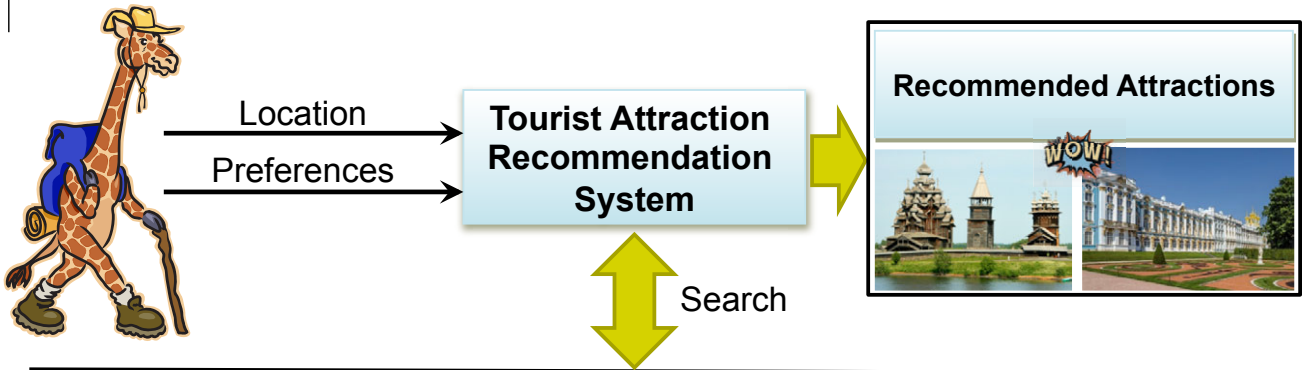
Tourist Attraction Information Service



4

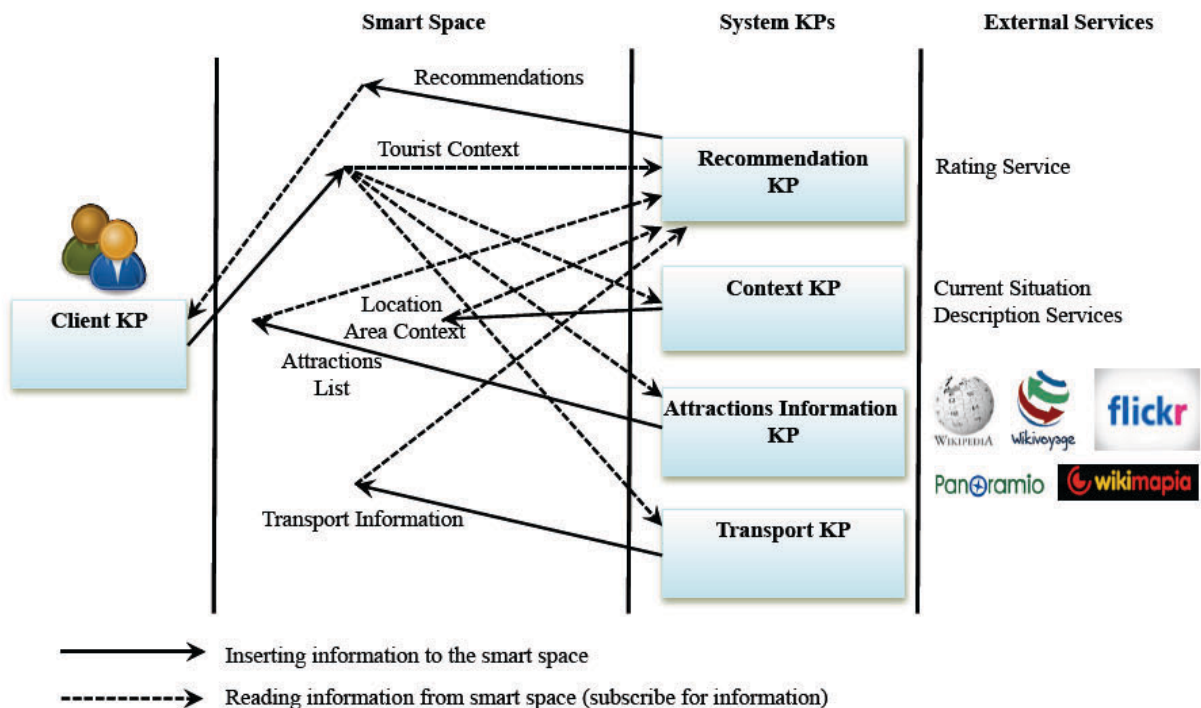
Tourist Attraction Recommendation System (TARS)

<https://play.google.com/store/apps/details?id=ru.nw.spiiras.tais>



5 5

TARS Services Interaction based on Smart Space Technology



6

Smart Space Communication

- Client KP

(alexey.kashevnik@gmail.com, is_a, tourist)
 (alexey.kashevnik@gmail.com, longitude, 60,12)
 (alexey.kashevnik@gmail.com, latitude, 30,24)
 (alexey.kashevnik@gmail.com, date_time, 2013-10-18 12:53)
 (alexey.kashevnik@gmail.com, language, English)
 (alexey.kashevnik@gmail.com, company, alone)
 (alexey.kashevnik@gmail.com, role, traveler)

- Context KP

(alexey.kashevnik@gmail.com, weather, Sunny)

- Attraction Information KP

(alexey.kashevnik@gmail.com, near_by, ListOfAttractionsXML)







- Recommendation KP

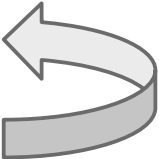
(alexey.kashevnik@gmail.com, "recommended_attractions", RecommendedAttractionsXML)

7

Attractions Recommendation

- Collaborative filtering systems make recommendations based on users interests similarity.

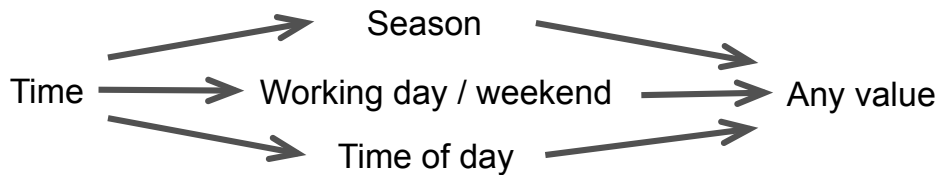
			
User 1 	2	3	2
User 2 	5	4	?
User 3 	4	5	5



8

Context for Attractions Recommendation

- Conditions in which a user rates an attraction (or gets recommendations) is called **context**.
 - Outside attractions will be rated better in sunny day then in rainy day
 - Zoological museum will be rated better if a user comes with children.
- The following context attributes have been identified for TARS:
 - Time
 - Weather
 - Company
- For taking context into account the **context generalization** method is used:



9

Example of Applying Context Generalization Method

Time: July 31, 2013 17:30
Company: With family
Weather: Sunny

5	4		3
4	5		2
	3	3	5
	2	5	4



Context Generalization

Time: Summer
Company: With family
Weather: Any value

5	4		3
4	5		2
	3	3	5
	2	5	4

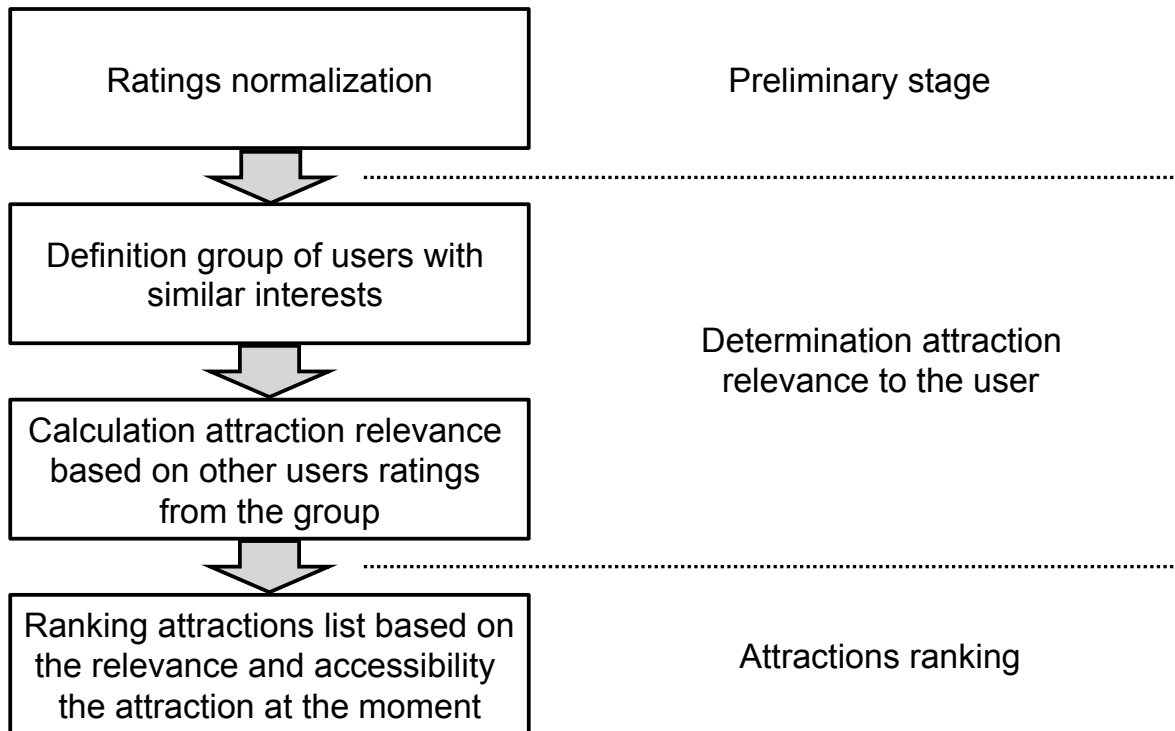


Time: Any value
Company: Any value
Weather: Any value

5	4		3
4	5		2
	3	3	5
	2	5	4

10

Algorithm for Attractions Recommendation



11

Attraction Images and Descriptions Filtering

- Internet services provide a big amount of images
- The system allows users to estimate every image (“like”, “dislike”)
- It is reasonable to show the user only the best images of the selected attraction
 - Show images that have the best score
 - Show new images (images without score)
 - Show small amount of images that have negative score (to exclude mistakes)
- Internet services provide several text blocks for an attraction
- It is needed to show the user the best text block and range other
 - User estimations («Like» / «Dislike»)
 - Text block characteristics: size, variety of vocabulary
 - Degree of similarity of a text block with the last shown

12

Conclusion

- The smart space based tourist attraction recommendation service can be overviewed during the DEMO section (Thursday, November 14 from 18.00 till 21.00, Seminar Hall A116).
- Recommendation Scenarios for TARS has been successfully implemented in special Recommendation KP.

13

Thank you for Attention
Questions are Welcome



St. Petersburg, Russia, E-mail: alexey@iias.spb.su

Development of Smart Room Services on Top of Smart-M3

Dmitry Korzun, Ivan Galov, Sergey Balandin

Petrozavodsk State University
Department of Computer Science



This project is supported by grant KA179 of Karelia ENPI — joint program of the European Union, Russian Federation and the Republic of Finland

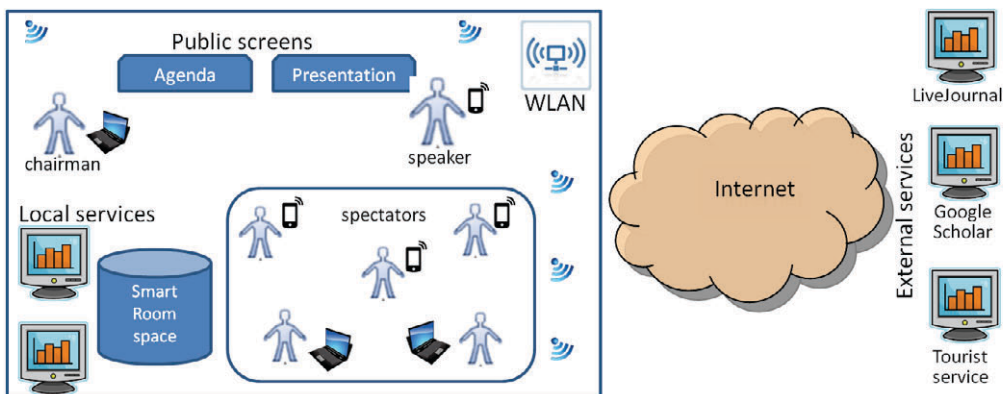


14th FRUCT conference
November 12, 2013, Helsinki, Finland



SmartRoom System

Collaborative work activity:
conferences, lectures, meetings, etc.

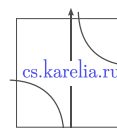


Evolution of SmartConference system (FRUCT, SPIIRAS)



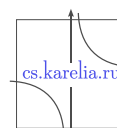
Functions to be made smart

- 1** Presentation
 - ▶ slide show
 - ▶ multiple sources for slides (e.g., video flows)
 - ▶ audio support (e.g., microphone)
 - ▶ interactive (e.g., online drawings)
- 2** Activity organization
 - ▶ agenda: automatic updates
 - ▶ chairman control and collaborative construction
 - ▶ background activity: online discussion
- 3** Local and remote participation
- 4** Content collection, analysis, summarization
- 5** Collaborative work (e.g., planning social events and tours)



Components

- Users: activity participants (end-users of services)
 - 1** chairman
 - 2** active speaker (in turn relay manner)
 - 3** spectators (inactive speakers, guests)
- Access/control: from clients on personal (mobile) devices
- Services: a distributed system deployed on surrounding devices, computers, Internet services
- Interoperability platform: Smart-M3

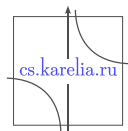


Development Method

Class: smart spaces service-oriented systems

Conceptual blocks:

- 1 digital environment
- 2 service set
- 3 computing infrastructure
- 4 ontological modeling
- 5 service construction and delivery



Digital (Service) Environment: Distribution

On the agent level

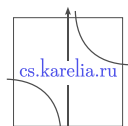
- Smart space is a named search extent of shared information
- Agents run on various computational devices
- Cooperative utilization of the smart space content

System

- Computing equipment is localized in a room
- WLAN provides connectivity
- External computing systems (corporate, global)

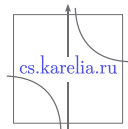
Extensions

- Resource-consuming processing can be delegated
- Existing Internet services can be used

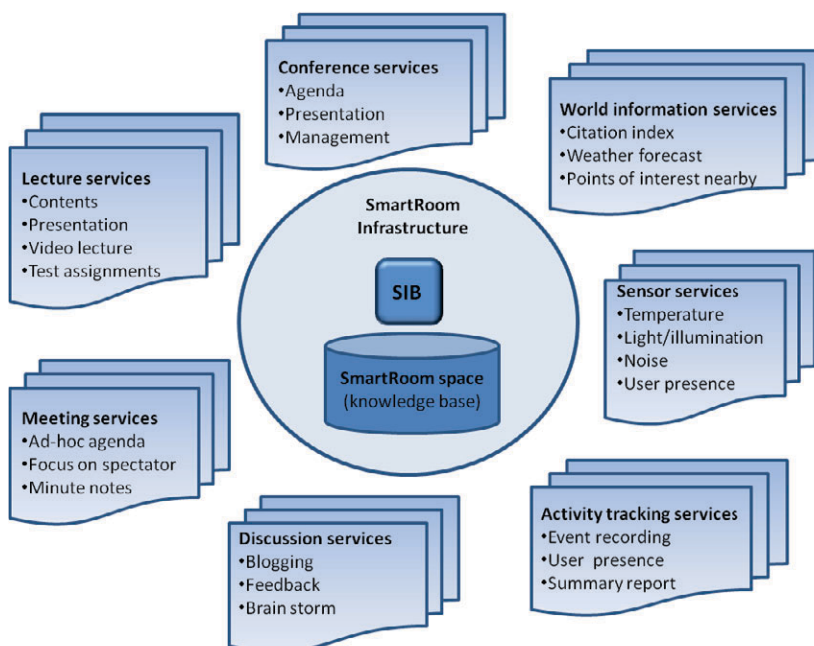


Digital (Service) Environment: Interfaces

- Core services: **Agenda** and **Presentation**
- Two big public screens are user interfaces for the services: **Agenda-screen** and **Presentation-screen**
- Information for all participants is visualized on public screens (composed from multiple services)
- End-user clients: personal screens
- SmartRoom UI elements: public screens and clients



Service Set



Activity type

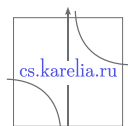
- Conference
- Meeting
- Lecture

Composition

- Presentation
- Agenda
- Extensions

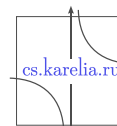
Informational

- Passive
- Control

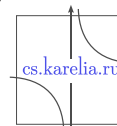
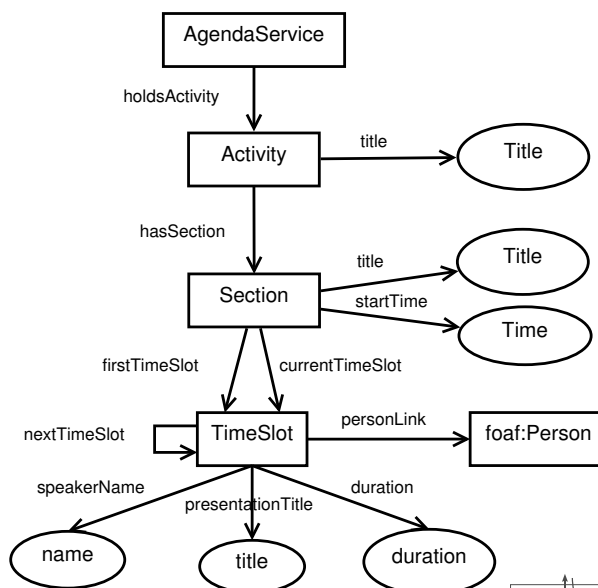
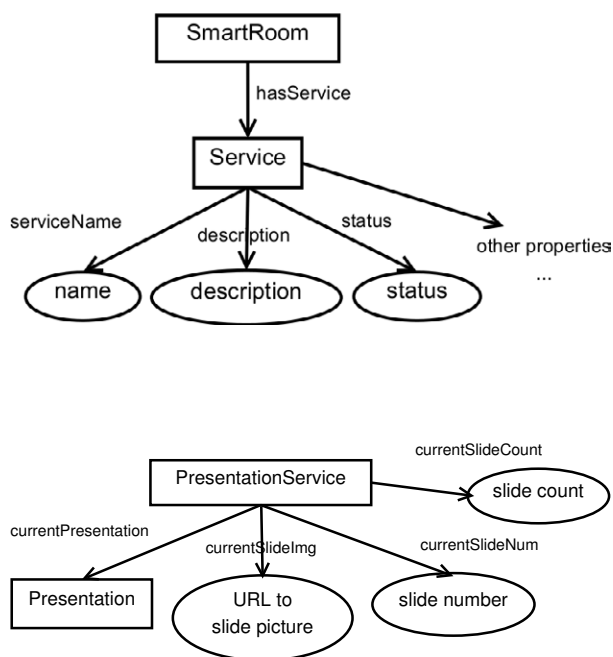


Computing Infrastructure

- interoperable information sharing platform
- Smart-M3 SIB: desktop computer or server
- Infrastructural KP: service-level knowledge processors
- Delegation of service processing to the infrastructure
- Deployment
 - ▶ Cluster near SIB: online 24/7 service mode, ease of installation and control
 - ▶ Device-specific: make an embedded or consumer electronics device a SmartRoom participant
 - ▶ Server: extensive or complex processing, data source mediation

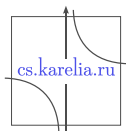


Ontological Modeling



Service Construction and Delivery

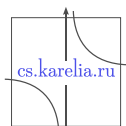
Property	Description
Explicit representation	Service is represented as an ontological instance in the smart space. A change in the representation activates the delivery
Compositional visualization	UI element uses multiple services and composes the most important information on a area-restricted screen
Personalization	End-user UI element uses personal information and context when visualizing the services
Collaboration	Service representation is constructed in a P2P manner by several KPs, including clients



Conclusion

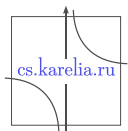
Reference smart space system?

- Multi-agent: IoT-like variety of devices and computing facilities
- Multi-service: core, extension, composition
- Digital services: visualization, personalization
- Development split
 - 1 digital environment
 - 2 service set
 - 3 computing infrastructure
 - 4 ontological modeling
 - 5 service construction and delivery



Open source code:

<http://sourceforge.net/projects/smartroom/>



Delivery of SmartRoom Services Using Mobile Clients

Andrey S. Vdovenko, Dmitry G. Korzun

Petrozavodsk State University
Department of Computer Science



This project is supported by grant KA179 of Karelia ENPI - joint program of the European Union, Russian Federation and the Republic of Finland



14th FRUCT conference

November 12, 2013, Helsinki, Finland

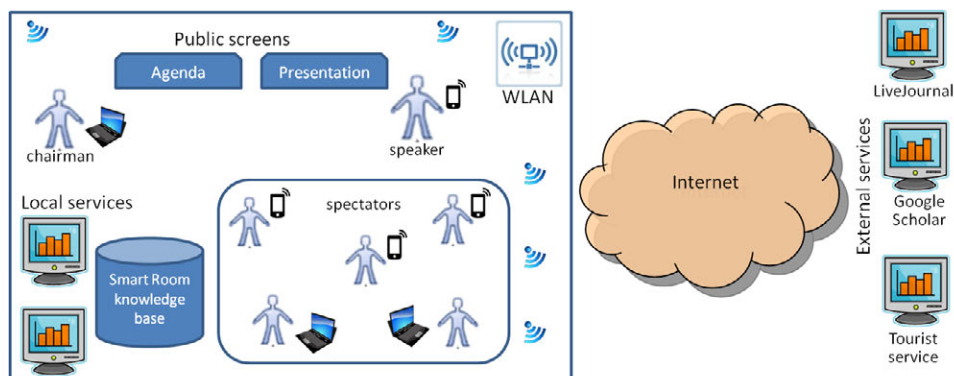


Dmitry Korzun

Delivery of SmartRoom Services Using Mobile Clients
SmartRoom

FRUCT14 1 / 10

SmartRoom system



- Many services (composition, personalization)
 - ↪ informational, control, collaborative work, ...
- Participation of many users
 - ↪ Many (mobile) clients running and accessing services
- Users come with own devices
 - ↪ Many mobile platforms, IoT-like device diversity



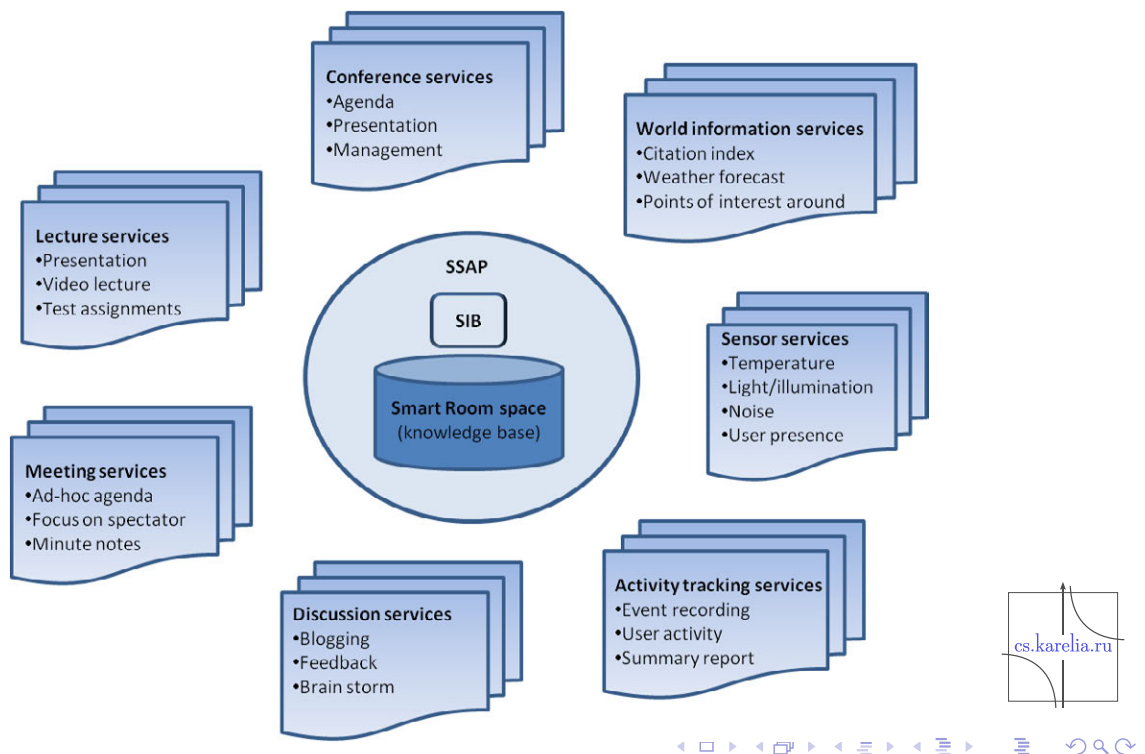
Dmitry Korzun

Delivery of SmartRoom Services Using Mobile Clients

FRUCT14 2 / 10

Multi-Service Property: Client

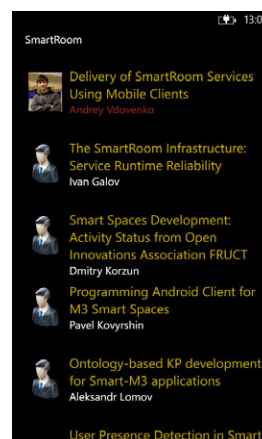
Service delivery: access interface on personal device



Two types of services (from client perspective)

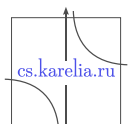
■ Off-the-shelf service

- ▶ thick client: local processing
- ▶ UI is customized
- ▶ platform-aware implementation
- ▶ low runtime flexibility

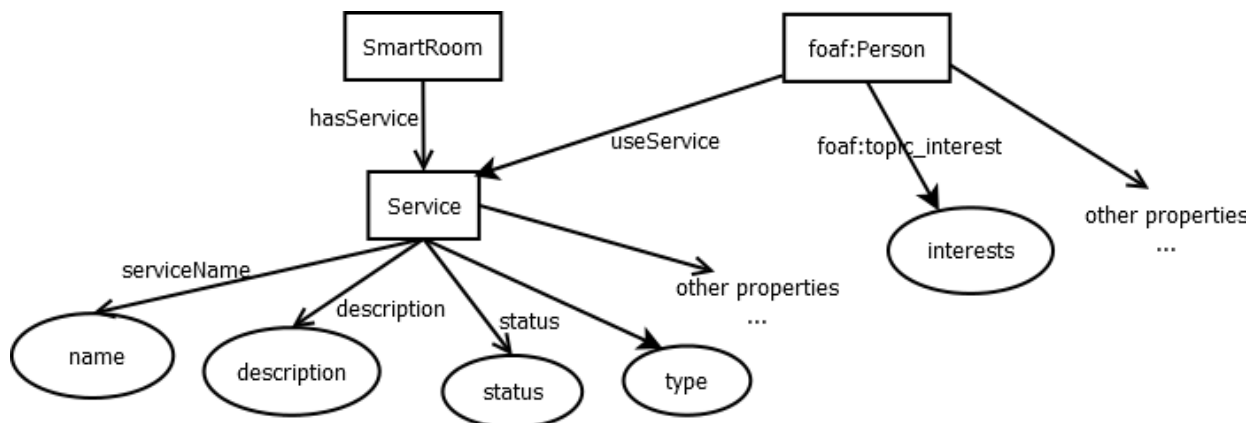


■ Ad-hoc service

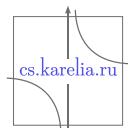
- ▶ runtime construction
- ▶ thin client: delegated processing
- ▶ lightweight UI
- ▶ flexibility for personalization



Service Ontology

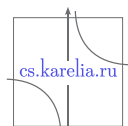
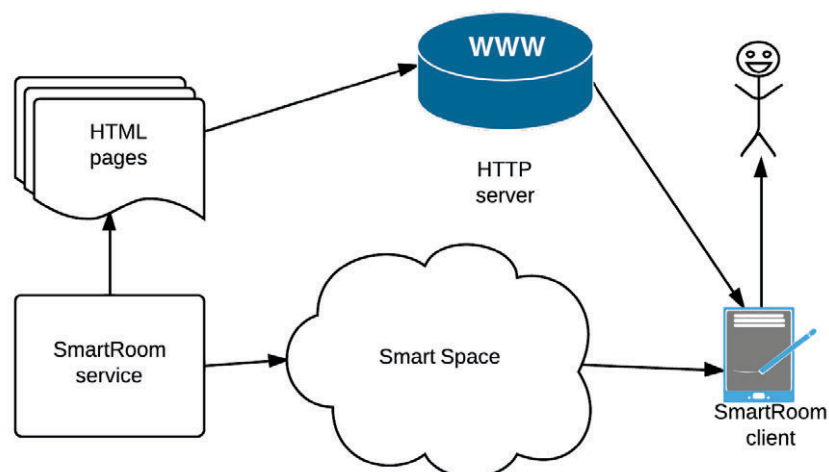


- Service **type**: for clients
- Service **description** and Person **interests**: semantic matching for personalization
- **useService**: runtime linking



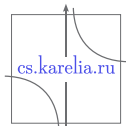
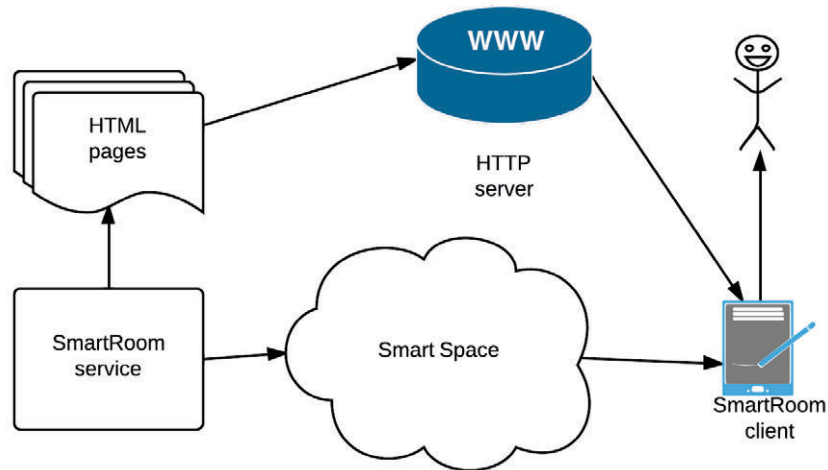
Ad-hoc Services: Architectural Vision (1/2)

- web application
- runtime construction is possible
- service has URL (shared in the smart space)



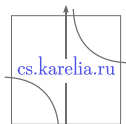
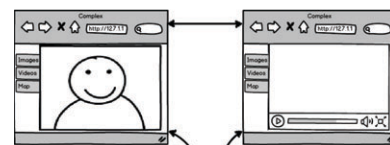
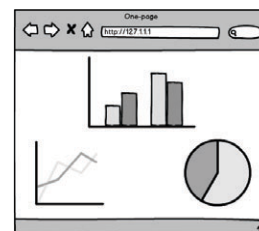
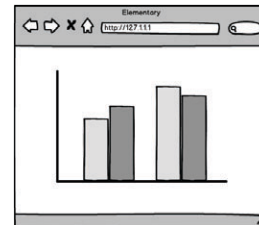
Ad-hoc Service: Architectural Vision (2/2)

- Web-application: HTML pages, JavaScript files, CSS styles
- Construction: client side vs. infrastructure



Ad-hoc Service: Composition on Client Side

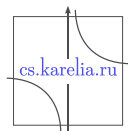
- Elementary
 - ▶ small piece of information
 - ▶ e.g., data from a sensor
- One-page
 - ▶ fits one web-page
 - ▶ structured visualization
 - ▶ e.g., activity report
- Complex
 - ▶ several web-pages
 - ▶ essential data processing on the client side
 - ▶ e.g., image search function



Personalization

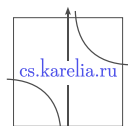
Relation of users and services

- Available services: this info shared in the smart space
- Personal information (personal space)
 - ▶ user registration
 - ▶ anonymous users
- known semantic matching methods
 - ▶ keyword matching with synonyms
- updates in personal space: context-awareness
- service composition on the client side



Conclusion

- Initial design solution
- Ad-hoc and off-the-shelf services: from client perspective
- Personalization in service delivering
- To appear in SmartRoom clients: Windows, Windows Phone, Android, Qt-based
- Open source code:
<http://sourceforge.net/projects/smartroom>



User Presence Detection in SmartRoom using Innorange Footfall Sensor

Sergey A. Marchenkov, Dmitry G. Korzun

Petrozavodsk State University
Department of Computer Science



This project is supported by grant KA179 of Karelia ENPI - joint program of the European Union, Russian Federation and the Republic of Finland



14th FRUCT conference
November 12, Helsinki, Finland



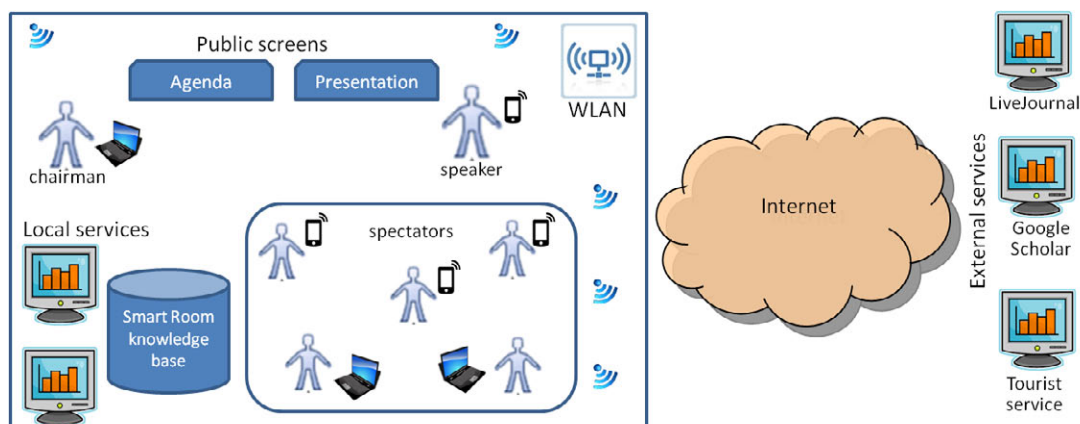
Dmitry Korzun

Presence Detection in SmartRoom

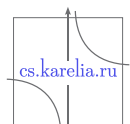
FRUCT14 1 / 11

User Presence: Scenarios

SmartRoom provides a set of digital services to many participants



- S_1 : User arrival to the room (spatial physical area)
- S_2 : User waits for beginning of the main activity
- S_3 : User joins and leaves during the main activity
- S_4 : Activity statistics



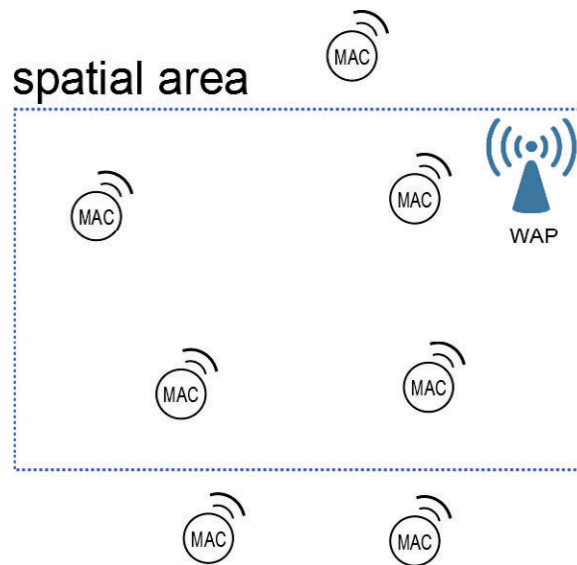
Dmitry Korzun

Presence Detection in SmartRoom

FRUCT14 2 / 11

Human participation

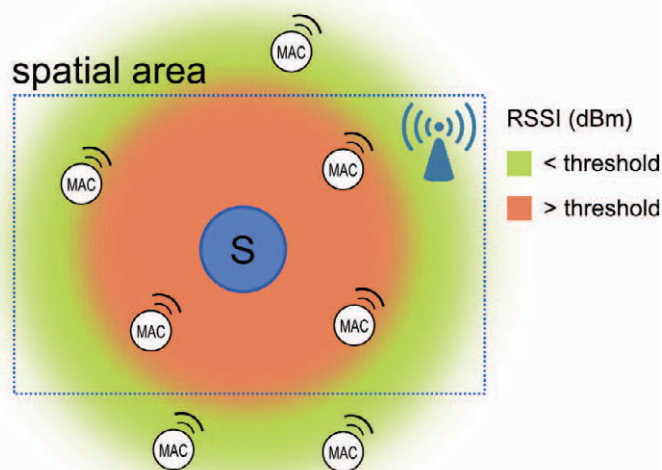
- End-users have personal computers and mobile devices
- User devices are connected to SmartRoom WLAN
- Mapping: MAC \leftrightarrow user



Navigation icons: back, forward, search, etc.

Passive Radio Detection using RSSI

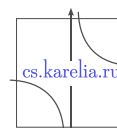
- Received Signal Strength Indication
- Estimate RSSI threshold (rough estimation)
- Low implementation cost
- Accuracy is satisfactory for typical instances of scenarios S_1-S_4



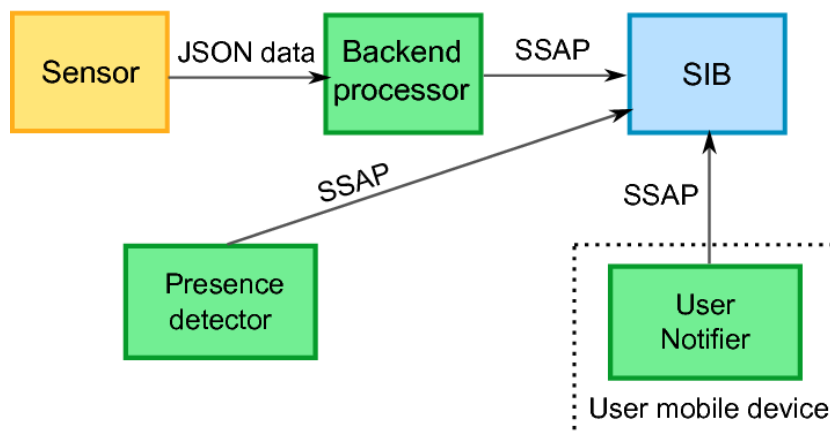
Navigation icons: back, forward, search, etc.

Innorange sensor: Footfall Technology

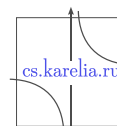
- Based on the methods of passive radio detection
- Detects presence of WLAN and BT capable devices
- Custom OpenWRT-based software
- Choosing Wi-Fi wireless channels



Architecture



- The sensor sends its measurements: MAC address, RSSI value and timestamp
- Backend processor:
 - ▶ HTTP endpoint to receive presence data from sensor
 - ▶ Presence information processing and store to SIB



Presence Detector KP

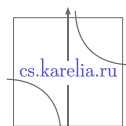
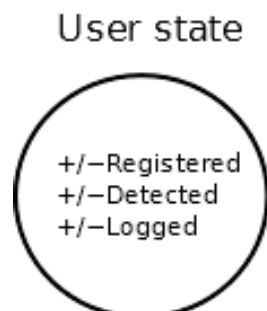
Name	Last seen	Degree of presence
Person 1	1 min	Full
Person 2	2 min	Full
Person 3	35 sec	Physical
Person 4	50 sec	Virtual
Person 5	1 min	Full

- The KP detects presence information change (enter and last seen timestamps)
- Determines the degree of a user's presence (e.g. absent or full)
- Publishes data in the SIB for other services and **User Notifier**

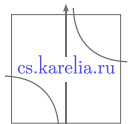
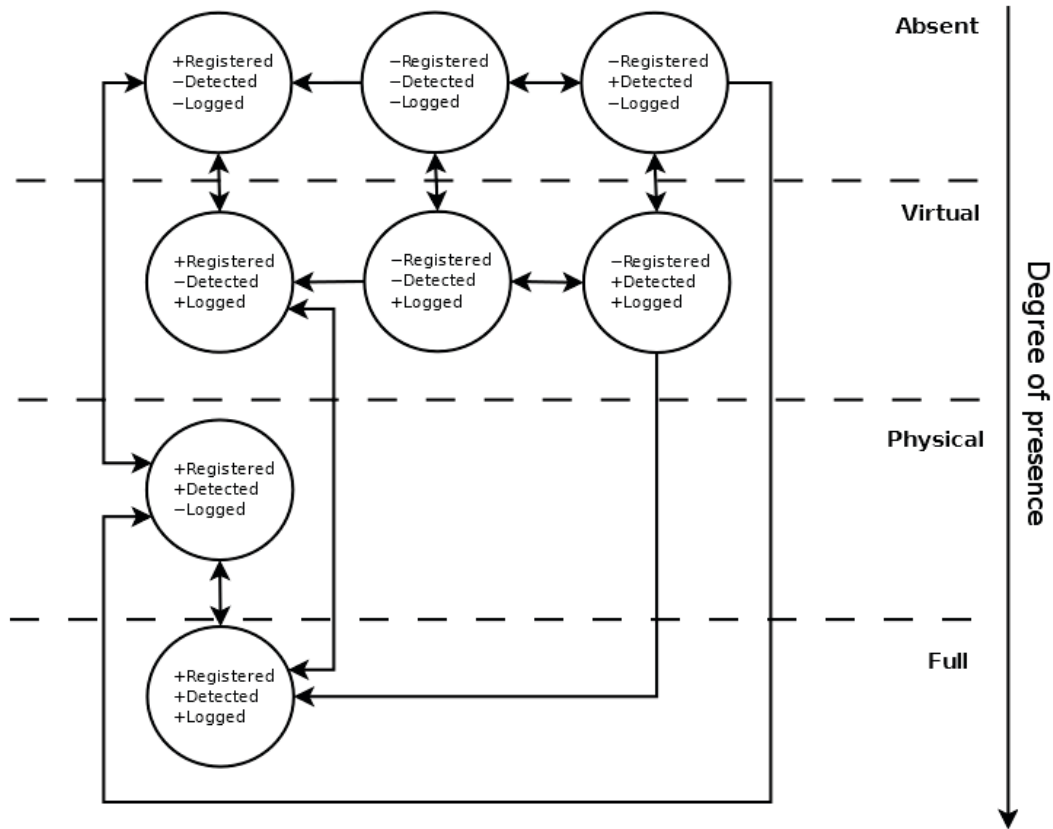


Degree of presence

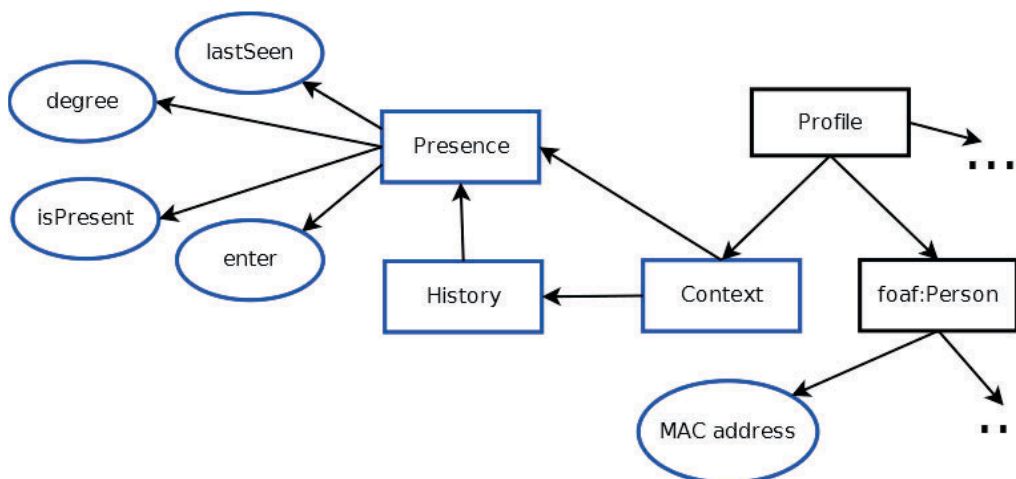
- User of the SmartRoom system has **3** basic triggers:
 - 1 Registered:** User registered in the system by registration service or client
 - 2 Logged:** User logged in the system with the client
 - 3 Detected:** User detected in the system by using Innorange sensor
- The user may be in different states at the same time
- Combinations of states determine differently the **degree of presence**



Degree of presence



Expansion of the SmartRoom Ontology



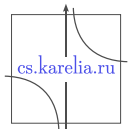
Triples example:

- $\langle \textit{Presence} - \textit{uuid}, \textit{enter}, 1362052175 \rangle$
- $\langle \textit{Presence} - \textit{uuid}, \textit{isPresent}, \textit{True} \rangle$
- $\langle \textit{Presence} - \textit{uuid}, \textit{lastSeen}, 1362052175 \rangle$
- $\langle \textit{Presence} - \textit{uuid}, \textit{degree}, \textit{Full} \rangle$



Integration of Footfall Sensor into SmartRoom

- Room with floor area of $13 \times 7 \text{ m}^2$
- Sensor is mounted the center of the room
- Personalized devices on different mobile platforms: Symbian, Android and Windows Phone
- RSSI threshold is -35
- Source code: <http://sourceforge.net/projects/smartroom/services/presence-service>



Discussion about the future of M3 and M3 community

Discussion about the future of M3 and M3 community

Juha-Pekka Soininen, Johan Lilius, Tullio Salmon Cinotti, Sergey Balandin, Petri Liuha

Juha-Pekka Soininen, VTT Technical Research Centre of Finland
Johan Lilius, Åbo Akademi
Tullio Salmon Cinotti, University of Bologna
Sergey Balandin, FRUCT Oy
Petri Liuha, EIT ICT Labs

The last session of this Workshop hosted by the Helsinki node of EIT-ICTLabs was a discussion about the future of M3 and M3 community. It started at 5 p.m. after a fruitful and very exciting day, with an audience of approximately 50 people coming from 5 countries: Russia, Finland, Spain, the Netherlands and Italy. Most of the participants, restless, had already attended 23 presentations, having provided smart feedback and comments that set the context for the final discussion which is here reported. The contribution from people that unfortunately could not personally join the WS but had sent opinions and feedback about M3 was also reported and discussed in the session.

The big picture.

M3 is reference concept architecture to connect the physical world with the information world, in order to make information in the physical world available for smart services [3].

M3 architecture was proposed by NRC in the first decade of this millennium, and its first C language implementation was released open source by NOKIA, under the name of SMART-M3, within the framework of ARTEMIS Project SOFIA. This happened at the second International NOTA conference in San Josè on October 1, 2009.

Since then M3 grew in terms of developments and localizations in several landscapes, including communities (FRUCT), European Technology Platforms (Artemis), the European Institute of Innovation and Technology (EIT ICT Labs), national research, International Cooperation (FP7-Capacities program, in cooperation with Egypt) and high education scenarios (University of Bologna).

Networking between teams of the M3 community has been mostly spontaneous so far with one noticeable exception: the FRUCT community has a stable cooperation plan in place, which is based on regular telcos, conferences and projects.

The WS demonstrated that the M3 community is active and consistent, well beyond expectations and it gained self-awareness at the WS.

The fundamental question that needs to be answered before discussing the future of the M3 community is:

Why investing on M3?

The WS clearly demonstrated that M3 not only matches the needs of many applications domains already approached by the community (see for example, the WS afternoon sessions presentations), but also it could become a perfect component of emerging classes of ecosystems like IoT, smart management of

physical space, large simulation frameworks with man in the loop and platforms to integrate different communication technologies (presentations of the first morning session).

Furthermore experimental applications in several domains were presented at the workshop, including, just to name some of them: context-aware support to the tourists, ridesharing, room and conference smart management with mobile services support, health and wellbeing (including remote rehabilitation), electric mobility, public lighting, context-based social media (Smart Scribo) and impact analysis of the food chain life cycle.

Meanwhile the M3 technology is progressively maturing.

In fact the intensive “marketing” activity above depicted is accompanied by an equally intensive and fascinating effort on M3 technology, which addresses performance, scalability, robustness, security, mapping on parallel execution vehicles, programming models and styles, protocols, low resource device support, separation of the processing infrastructure from the data infrastructure, moving computational load from the clients to the infrastructure, and tools. Most of the morning sessions presentations enlighten these developments which are currently in progress not only within research organizations but also in the industry. And in fact all WS contributors from Spain come from industrial research companies.

The consistency of the produced effort can be better understood by relating the WS results with Fig.1 which shows the M3 SWOT analysis made in November 2011 at the last SOFIA General Assembly just before the end of the project: while the *strengths* are still valid, most of the *opportunities* are being taken and most of the *threats* and *weaknesses* are being handled as specifically demonstrated by several of the WS presentations.

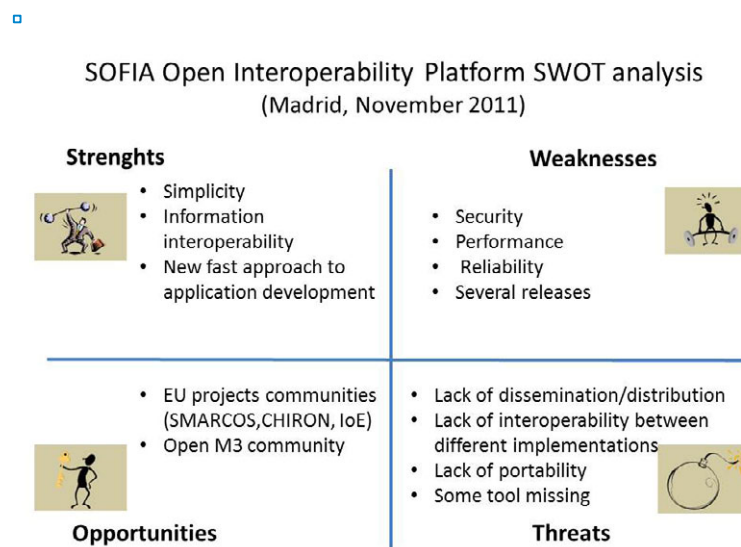


Fig. 1: M3 SWOT Analysis (2011)

No comments are needed about the strengths which are still valid.

The weaknesses are being handled mainly by VTT (security), by P. G. Demidov Yaroslavl State University (reliability and recovery procedures), by Åbo Akademi and University of Bologna (performance). On the other side having several implementations is not considered a weakness anymore, as there is no best implementation suitable for all possible scenarios. It is important to keep the number of releases as low as

possible and all releases should be compatible and motivated (e.g. by the target framework). Some competition between releases could also be fruitful along the M3 roadmap.

The threats too are being taken care of: dissemination material and manuals are becoming more and more available. A lot of material (including tools, student tests and tutorials originated by several sources) are available on the web-site of the Course named *Laboratory of Interoperability of Embedded Systems* offered by the University of Bologna. This course was set up also with the support of the “Open Local Data Applications” Activity of EIT ICT Labs and its web-site is deemed to be kept up-to-date as long as the course will be active.

The lack of portability of the reference implementation (which is linux based) is being overcome by a new OSGi based release (currently under test) which was presented at the WS. The lack of interoperability between releases requires at least some activity on the API. A significant effort is currently dedicated to standardize SPARQL as M3 query, update and subscribe language, so while the APIs are adapted to SPARQL they are also being revised to ensure M3 releases interoperability.

The “tools missing issue” has also been widely considered by several teams in the community. Here the most outstanding presentation was given by Adrian Noguero who presented Tecnia approach to a new superfast and hyper-friendly approach to mobile M3 applications development. But also other members of the community are working on useful tools, basically in all the countries represented at the WS.

Moving on to the “opportunities”, most of the participants are promoting M3 in several challenging contexts. On top of the actions summarized above there is a move to address the makers community (e.g. Raspberry community, the Arduino community and others) as well as an attempt to have M3 adopted by establishments that have not participated to the M3 developments. The most outstanding example was presented by AICIA in the first morning session.

Some new and more mature threat was actually raised by some researcher that could not participate to the WS. This relevant contribution is listed in the underneath table.

Tab. 1: “things to improve” as suggested by worldwide M3 users

□ **What are the most up-to-date needs/threats reported by M3 users which are not M3 architects? (2013)**

Threat generating needs	Actions to be taken (or already taken)
More detailed documentation to understand M3 evolution and the motivations for so many versions	<ul style="list-style-type: none"> • Agree on M3 releases and associated qualities • Roadmap for convergence and strong motivation for different implementations
<ul style="list-style-type: none"> • bug reporting facility needed • mis-behaviours claimed in multi-users applications with high rate of information interchange 	<ul style="list-style-type: none"> • Report bugs in repeatable form • Specify stability measurements (eg MTBF) in reference configurations • Define an M3 benchmark suit • Define standard recovery procedures
<ul style="list-style-type: none"> • More mature (less “prototype flavoured”) and neater offer • Ability to Handle non-English characters (e.g. arabic) 	
Comparison with other centralized publish/subscribe, content-based data distribution technologies	<ul style="list-style-type: none"> • E.g.:MQTT (IBM, Eurotech)
more organized ecosystem behind M3!	

These relevant feedbacks make it clear that a careful strategic plan is now required to materialize an M3 impact on the industry and on the society.

To approach this plan the discussion focused on the following topics:

1. A helicopter view of M3 technology status and its competitive potential
2. An overview of possible community reinforcing actions.

Helicopter view of M3 technology status

Fig. 2 is an abstract onion peeling view of an M3 infrastructure as perceived by the multi-language, multi-platform application developer. Fig. 3, in turn, is an attempt to sketch at a glance both genesis and history of the main M3 infrastructure implementations.

□

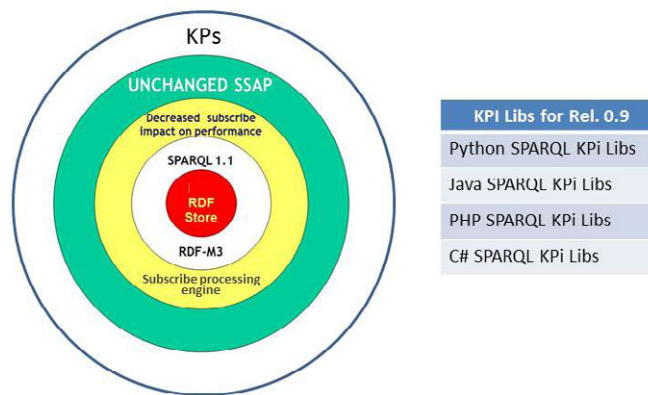


Fig. 2: Onion peeling view of M3 infrastructure architecture:
KPi Libraries may hide the SSAP protocol from the KPs.
Available KPi Libs are listed on the righthand side.

As shown by fig. 3, all M3 implementations start from the original Piglet based “Python SIB” born in NRC before SOFIA. From this ancestor four development lines started, three within SOFIA and one (the RIBS) within a national project.

□

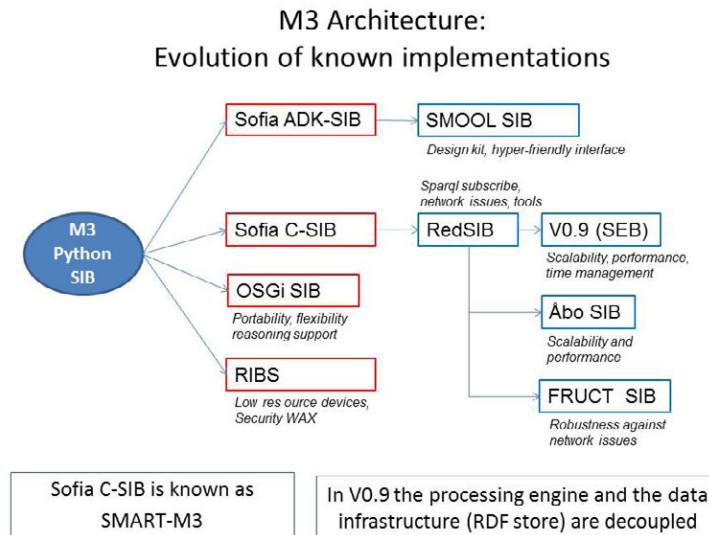


Fig. 3: M3 Implementations taxonomy and threads

The SOFIA-ADK SIB (a Java SIB) was created in Spain and all of the related developments were mostly focused on the development framework aiming to hide the ontology details and the low level triple based data model from the developer [20],[21]. An exciting, hyper-friendly and new design kit originating from SOFIA-ADK was presented at the WS (SMOOL).

SOFIA C-SIB, aka SMART-M3 was launched under this name (SMART-M3) as open source platform on October 1, 2009 at the Second International NOTA conference in San José (Fig. 4).

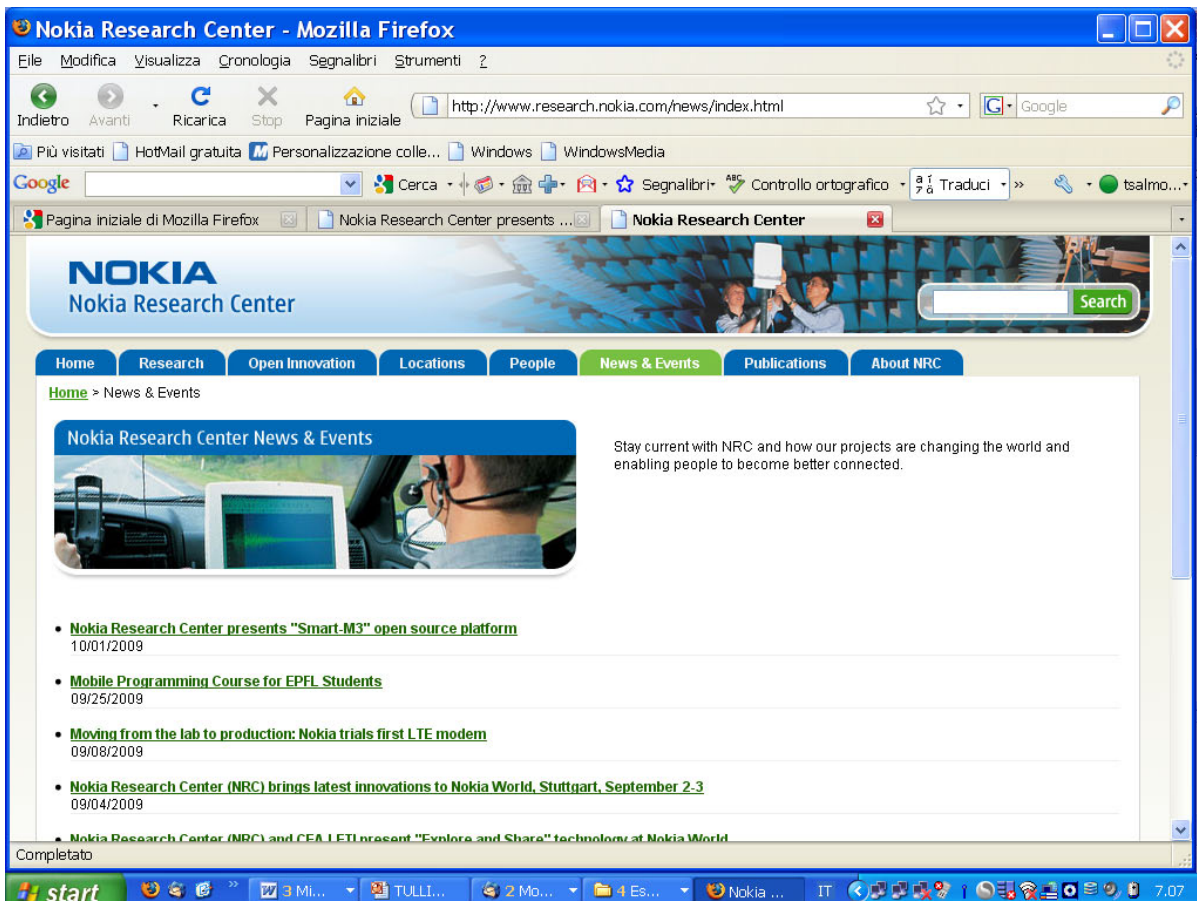


Fig. 4: "Smart-M3: Open Source Platform" - top NRC home page news on Oct 1 2009

From SOFIA-C SIB a consistent and incremental set of M3 Semantic Information Broker generations was developed in Italy. In November 2012 the REDSIB was presented at the FRUCT Conference [12] while Release 0.9 was first presented at the WS. With the REDSIB the glorious and extremely powerful Wilbur Query Language had to be - at least temporarily - abandoned and SPARQL was adopted. SPARQL in fact is the query language supported by the REDLAND RDF Store which replaced the no more supported Piglet. Powerful SPARQL Subscribes were also introduced. With Release 0.9 additional innovations came into play, including time management (delayed SPARQL UPDATE), scalability and sharper decoupling of the processing engine from the data store. Currently REDLAND and VIRTUOSO RDF stores are supported. Developments to enhance the REDSIB robustness and its performance level are currently undergoing in Finland and Russia, as depicted in fig. 3.

As the SOFIA C-SIB line is platform dependent (Linux), an attempt to create a portable M3 implementation compatible with the latest generations of SOFIA C-SIB is currently being made. This is the OSGi SIB which was presented at the Workshop. Given the OSGi framework reputation and diffusion in the industry, this implementation is a further attempt to introduce M3 technologies in the industrial market. On the other side the OSGi SIB is designed to facilitate the introduction of new features, new components and new core services. A noticeable feature already introduced in the OSGi SIB is the persistent SPARQL update, an important step to automate the interaction between KPs and SIB and reduce network traffic.

Last but not least, the RIBS is a very interesting M3 implementation created at VTT and focused on performance, security and low resource devices at the same time. As one of the performance bottlenecks in M3 ecosystems is the XML based Smart Space Access Protocol (i.e. the SIB-KP application level protocol), the RIBS enforces a different protocol with respect to all other M3 implementations.

This aspect brings into play a final consideration. As shown by Fig. 2, while a lot of effort was dedicated to all M3 infrastructure layers, the SSAP protocol was always left unchanged. This happened for very wise reasons: preserving cross-platform and backward compatibility and offering new developers a stable SIB access gate. Therefore, moving on from one generation to the next one, all Fig. 2 layers - but the SSAP - increased significantly their performance level. But in the meantime protocol researchers did not stand still. On the contrary they were very active in investigating and proposing new more efficient protocols: for example lightweight protocols for low power channels, and also more SPARQL-oriented protocols. The current status of this research was enlightened at the Workshop.

So replacing SSAP or putting next to it a more effective and performing protocol, is one of the directions where the M3 infrastructure is expected to evolve in the near future.

Additional M3 features, new ecosystems and new application domains will be discussed inside the M3 community which, as already pointed out, gained a lot of self-awareness at the Workshop and it is now getting ready to discuss an M3 technology development roadmap.

Overview of possible community reinforcing actions

The discussion and the workshop itself proved that M3 is a solid concept that has new users and developers, who think the idea is worth pushing forward. There have been earlier attempts to build the M3 community, but unfortunately they have not been very successful. The SOFIA M3 community website www.sofia-community.org was finalized in the very late phase of the project and never managed to attract developers and users. After SOFIA ended there was no more active organization to host and to put effort for it. Another major attempt, the Smart-M3 project in SourceForge, was established by Nokia, but the changes in Nokia's strategy did leave it in a floating state. However, the ownership of the Smart-M3 project has moved to University of Bologna, and the Smart-M3 SourceForge website, administered by former NRC researchers currently with FRUCT, is now an active place for distribution and development of the RedSIB implementation of Smart-M3. VTT also established a website www.open-m3.org as a place for information sharing related to M3. Neither this site managed to attract users due to poor marketing and lack of resources to update it. The FRUCT community has had lot of M3 activities, but they have been integrated with other FRUCT activities and there has not been a dedicated forum for M3.

It was also recognized that the partners developing M3 have not proper resources for running an organized community. Partners are University groups or research organization groups that do not have sustainable funding for this kind of activities. All operations are more or less project funding dependent which makes it very difficult to create organized communities. EIT ICTLabs and FRUCT Oy are more independent on project funding, but supporting this kind of a community is not purpose of EIT, and FRUCT is more focused in Finnish-Russian co-operation instead of more international operations that are required in this case.

Despite of the problems the participants of the workshop where quite unanimous about the need to reinforce the M3 community based on more light-weight and voluntary means. We agreed on four specific issues that need to be addressed in the near future:

1. **M3 community has to organize into something more stable.** It was agreed that in the future the M3 has to be standardized and there has to be some organization that looks after the modifications and diversifications of the approach. It was proposed the some kind of unofficial working group should be established.

2. Community needs a **forum for discussions and information sharing**. This workshop was a success and a strong desire to extend the workshop into a workshop series was expressed. It was also noted that especially in FRUCT there are set events that could be good forums for M3 related work.
3. **Dissemination of M3 related information** needs to improve. Currently the Smart-M3 code can be shared using SourceForge, but there are lot of other documentation and content such as specifications, user guides, procedures, design use cases, papers, ontologies, apps, libraries etc. that should also be easily available. A dedicated website for M3 and more public SIBs should be established.
4. **To increase the collaboration between project partners**. It was acknowledged that the best way to create and enforce the community is to work together. Possible specific actions that were considered where the creation of M3 AppStore for additional sharing of results and even for having a place to do business with them, the setting up of joint research and development projects with community members, and to support the researcher exchange between community members. The role of FRUCT and EIT ICTLabs was considered important is these activities.

The current program committee of the workshop volunteered to start discussions and actions related to these topics.

Conclusions

The audience concluded that the next M3 International Workshop needs to be planned soon and immediately two candidates location were envisioned: the next RuSmart and the next FRUCT conference. The proposal was made to invite W3C representatives and start a discussion about the M3 standardization roadmap.

Acknowledgements

The Committee is grateful to EIT ICTLabs for hosting the workshop and the final event in the beautiful Innovation House. Certainly the Innovation House “landscape” greatly contributed to the WS success.

References

This is an initial and still incomplete list of references providing background information on M3 technologies. It is partitioned into the following subsections: M3 on SourceForge, Historical Perspective, M3 in Education, M3 platforms and related developments, M3 tools and development platforms, .security and protocols. References to the applications that are already quite many and increase rapidly in several countries are not included.

Completing this list and creating an additional list covering M3 applications literature is one of the suggested community reinforcing actions.

M3 on SourceForge

1. SMART-M3 SourceForge website: <http://sourceforge.net/projects/smart-m3/>

Historical Perspective

2. O. Lassila, *Programming semantic web applications: a synthesis of knowledge representation and semi-structured data* [doctoral dissertation], Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Software Technology, 2007.

3. P. Liuha, J. Soininen, and R. Otaolea, "SOPIA: opening embedded information for smart applications," in Proceedings of the Embedded World Conference, Nuremberg, Germany, March 2010.
4. J. Honkola, H. Laine, R. Brown, O. Tyrkko: *Smart-M3 information sharing platform*. ISCC 2010: 1041-1046
5. D. G. Korzun, S. I. Balandin, V. Luukkala, P. Liuha, A. V. Gurtov, *Overview of Smart-M3 principles for application development*, Proc. Congress on Information Systems and Technologies (IS&IT'11), Conf. Artificial Intelligence and Systems (AIS'11) Vol. 4 Pg. 64-71
6. E. Ovaska, T. Salmon Cinotti, A. Toninelli. (2012). *The Design Principles and Practices of Interoperable Smart Spaces*. In Advanced Design Approaches to Emerging Software Systems: Principles, Methodology and Tools.
7. A. Toninelli, S. Pantsar-Syväniemi, P. Bellavista, E. Ovaska, *Supporting context awareness in smart environments: a scalable approach to information interoperability*, M-PAC '09 Proc. of the Int. Workshop on Middleware for Pervasive Mobile and Embedded Computing

M3 in Education

8. Frank Wickström, *Getting Started with Smart-M3 Using Python*. TUCS Technical Reports 1071, TUCS, 2013.
9. F. Morandi, F. Vergari, A. D'Elia, L. Roffia, T. Salmon Cinotti (2013). SMART-M3 v.0.9: A semantic event processing engine supporting information level interoperability in ambient intelligence. <http://amsacta.unibo.it/3877/>, 2013
10. L. Roffia, A. D'Elia, F. Vergari, D. Manzaroli, S. Bartolini, G. Zamagni, T. Salmon Cinotti, J. Honkola. (2010). *A Smart-M3 lab course: approach and design style to support student projects*. Invited paper, 8th Conference of Finnish-Russian University Cooperation in Telecommunications (FRUCT).Lappeenranta (Finlandia). 9-12 Nov 2010. (pp. 142 - 153).
11. Laboratory of Interoperability of embedded systems. Course material and tools. ARCES, University of Bologna, <http://didattica.arces.unibo.it/mod/resource/view.php?id=468>

M3 platforms and related developments

12. F. Morandi, L. Roffia, A. D'Elia, F. Vergari, T. Salmon Cinotti. (2012). *RedSib: a Smart-M3 Semantic Information Broker Implementation*, Proceedings of the 12th Conference of Open Innovations Association FRUCT, Eds Sergey Balandin and Andrei Ovchinnikov, Oulu, Finland, November 5-9, 2012, SUAI SAINT-PETERSBURG, (pp. 86-98) ISSN 2305-7254.
13. J. Suomalainen, P. Hyttinen, P. Tarvainen. 2010. *Secure Information Sharing between Heterogeneous Embedded Devices*. Proceedings of the Fourth European Conference on Software Architecture: Companion Volume. ACM Press, ss. 205-212. The First International Workshop on Measurability of Security in Software Architectures - MeSSa 2010, Copenhagen Denmark, 23 August 2010, <http://portal.acm.org/citation.cfm?id=1842793>
14. J-P. Soininen, J. Takalo-Mattila, E. Viljamaa, P. Hyttinen, J. Kiljander, M. Eteläperä, A. Ylisaukko-oja. 2013. *New smart systems by extending semantic web to low capacity devices*. In: Uniting the physical and virtual - ICT solutions for open smart spaces. Ailisto, Heikki & Belloni, Kaisa (Eds.). VTT Research Highlights 7. VTT, ss. 16 – 25, doi: <http://www.vtt.fi/inf/pdf/researchhighlights/2013/R7.pdf>
15. V. Luukkala, J. Honkola. Integration of an Answer Set Engine to Smart-M3. *NEW2AN 2010 - Smart Spaces and Next Generation Wired/Wireless Networking 2010 92-10, LNCS 6294*, Springer Berlin Heidelberg
16. I. Paramonov, A. Vasilev, D. Laure, I. Timofeev. *Agent Substitution Mechanism for Dataflow Networks: Case Study and Implementation in Smart-M3*. Internet of Things, Smart Spaces, and Next Generation Networking, LNCS, vol. 8121, Springer Berlin Heidelberg
17. P. Bellavista, V. Conti, C. Giannelli, J. Honkola: *The Smart-M3 Semantic Information Broker (SIB) Plug-In Extension: Implementation and Evaluation Experiences*. GreenCom 2012: 704-711
18. A. Ifredo D'Elia, J. Honkola, D. Manzaroli, T. Salmon Cinotti: *Access Control at Triple Level: Specification and Enforcement of a Simple RDF Model to Support Concurrent Applications in Smart Environments*. NEW2AN 2011: 63-74
19. D. Manzaroli, L. Roffia, T. S. Cinotti, P. Azzoni, E. Ovaska, V. Nannini, S. Mattarozzi, *Smart-M3 and OSGi: The interoperability platform*, 2010 IEEE Symposium on Computers and Communications (ISCC), pp.1053,1058, 22-25 June 2010

M3 tools and development platforms

20. J.F. Gómez-Pimpollo, R. Otaolea, "Smart Objects for Intelligent Applications - ADK," *Visual Languages and Human-Centric Computing (VL/HCC)*, 2010 IEEE Symposium on , vol., no., pp.267,268, 21-25 Sept. 2010

21. A. Katasonov, M. Palviainen, "Towards ontology-driven development of applications for smart environments," *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, , vol., no., pp.696,701, March 29 2010-April 2 2010, doi: 10.1109/PERCOMW.2010.5470523
22. D. Korzun, A. Lomov, P. Vanag, J. Honkola, S. Balandin, *Generating Modest High-Level Ontology Libraries for Smart-M3*, UBIComm 2010, The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies
23. D. G. Korzun, A. A. Lomov, P. I. Vanag, S. I. Balandin, J. Honkola. Multilingual Ontology Library Generator for Smart-M3 Information Sharing Platform. *International Journal On Advances in Intelligent Systems*, vol. 4, n. 3 and 4, 2011
24. A. Kaustell, M. M. Saleemi, T. Rosqvist, J. Jokiniemi, J. Lilius, I. Porres. *Framework for Smart Space Application Development*. Proc. of Int.l Workshop on Semantic Interoperability (IWSI 2011). SciTePress, pp. 3-12, January 2011, Rome, Italy.
25. M. M. Saleemi, N. Diaz Rodriguez, J. Lilius, I. Porres. *A Framework for Context Aware Applications for Smart Spaces*. Proc. of the 4th conference on Smart Spaces (ruSMART 2011). Springer-Verlag, pp. 14-25, August 2011, St. Petersburg, Russia.

Security and protocols

26. A. Evesti, E. Ovaska, "Ontology-based security adaptation at run-time", In Proc. SASO 2010, pp. 204 - 212
27. A. Evesti, R. Savola, E. Ovaska, J. Kuusijärvi, "The design, instantiation, and usage of information security measuring ontology", In proc. MOPAS 2011, pp. 1-9
28. A. Evesti, E. Ovaska, "Design time reliability predictions for supporting runtime security measuring and adaptation", In EMERGING 2011, pp. 94-99
29. A. Ylisaukko-oja, P. Hyttinen, J. Kiljander, J. Soininen and E. Viljamaa, "Semantic Interface for Resource Restricted Wireless Sensors," in *IC3K 2nd International Workshop on Semantic Sensor Web – SSW2011*, October, 2011, Paris, France.
30. J. Kiljander, F. Morandi, and J-P. Soininen, "Knowledge Sharing Protocol for Smart Spaces", in *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 3, No. 9, 2012.
31. A. Evesti, J. Suomalainen, E. Ovaska, "Architecture and knowledge-driven self-adaptive security in smart space", in *Computers*. MDPI. Vol. 2 (2013) No: 1, 34 - 66, doi-link: 10.3390/computers2010034

Turku Centre for Computer Science

TUCS Lecture Notes

1. **Ralph-Johan Back och Joakim von Wright**, Matematik med lite logik: Strukturerade härledningar I gymnasiematematiken
2. **Ralph-Johan Back och Joakim von Wright**, Matematik med lite logik: En kort kurs i talteori
3. **Ralph-Johan Back och Joakim von Wright**, Matematik med lite logik: Studentexamen i lång matematik, våren 2003
4. **Ralph-Johan Back ja Joakim von Wright**, Matematiikkaa logiikan avulla: Rakenteiset päättelyketjut lukiomatematiikassa
5. **Ralph-Johan Back ja Joakim von Wright**, Matematiikkaa logiikan avulla: Lyhyt lukuteorian kurssi
6. **Ralph-Johan Back ja Joakim von Wright**, Matematiikkaa logiikan avulla: Pitkän matematiikan ylioppilaskoe, kevät 2003
7. **Ralph-Johan Back och Joakim von Wright**, Matematik med lite logik: Introduktion till strukturerade härledningar
8. **Ralph-Johan Back och Joakim von Wright**, Matematik med lite logik: Logik för strukturerade härledningar
9. **Ralph-Johan Back och Joakim von Wright**, Matematik med lite logik: Strukturerade härledningar som allmänt bevisformat
10. **Ralph-Johan Back ja Joakim von Wright**, Matematiikkaa logiikan avulla: Johdatus rakenteisiin päättelyketjuihin
11. **Ralph-Johan Back ja Joakim von Wright**, Matematiikkaa logiikan avulla: Logiikka ja rakenteiset päättelyketjut
12. **Ralph-Johan Back ja Joakim von Wright**, Matematiikkaa logiikan avulla: Rakenteiset päättelyketjut yleisenä todistusmuotona
13. **Jarkko Kari (Editor)**, Proceedings of JAC 2010 – journées Automates Cellulaires
14. **Mike Stannet, Danuta Makowiec, Anna T. Lawniczak and Bruno N. Di Stefano**, Proceedings of the Satellite Workshops of UC 2011
15. **Timo Leino (Editor)**, Proceedings of the IRIS 2011 Conference
16. **Hongxiu Li (Editor)**, Studies on Inequalities in Information Society – Proceedings of the Conference, Well-Being in the Information Society. WIS 2012
17. **Vesa Halava, Juhani Karhumäki and Yuri Matiyasevich (Editors)**, RuFiDiM II, Proceedings of the Second Russian Finnish Symposium on Discrete Mathematics 2012
18. **Michael Butler, Stefan Hallerstede and Marina Waldén (Editors)**, Proceedings of the 4th Rodin User and Development Workshop
19. **Hongxiu Li and Jonna Järveläinen (Editors)**, Effective, Agile and Trusted eServices Co-Creation – Proceedings of the 15th International Conference on Electronic Commerce ICEC 2013
20. **Juhani Karhumäki, Markus Whiteland and Luca Zamboni (Editors)**, Local Proceedings of WORDS 2013
21. **Juha-Pekka Soininen, Sergey Balandin, Johan Lilius, Petri Liuha and Tullio Salmon Cinotti (Editors)**, Proceedings of the Open International M3 Semantic Interoperability Workshop

TURKU
CENTRE *for*
COMPUTER
SCIENCE

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www.tucs.fi



University of Turku

Faculty of Mathematics and Natural Sciences

- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics

- Institute of Information Systems Science



Åbo Akademi University

Division for Natural Sciences and Technology

- Department of Information Technologies

ISBN 978-952-12-2990-9

ISSN 1797-8831