Building Automation Systems from Internet of Things

Professor Jerker Delsing
EISLAB
Luleå University of Technology
Heathrow terminal 5

5 million connected points!!
What about London railway then?

X.XXX.XXX number of bearings

- Connected bearings will support
  - Bearing condition monitoring
  - Railway wagon condition monitoring
IoT Product Segments

Conveyor (Tier2) Components and Parts (Tier3)

- Drive Heads
- LTU & Winches
- Belt Structure
- Belting
- Pulleys
- Feeder Breakers
- Components (a.u. idlers, motors, etc.)

Suppliers of these Products are:

- Potential partners, and;
- Future Service Providers

One customer, KGHM, one component

- 120 km conveyers
- 720,000 idler bearings
The automation challenge

- Annual growths more than 10% and over 500 billion connected devices are expected worldwide by 2025. - Cisco 2013

- Massive automation systems not possible with current technologies

- Not enough many engineers on the globe to do the job with current technology
ISA-95 systems in to the cloud?
Collaborative automation

Benefits to the production industry - Spire

- Better optimization and coordination of single processes or process chains and of complete plants and sites,
- Significantly improved resource efficiency.
- Better coordinated control loops in one process step and improved collaboration of control systems of different processes along a process chain give higher process yields which results in better material efficiency, waste reduction, less energy use and reduction of pollution.
- Improved product quality through better process control and smart quality control
- Higher utilization of equipment
- New collaborative solutions with integrated information management offer new possibilities for supply chain management including price-based coordination or optimised market mechanisms
- Safer operation of plants due to improved control and shut-down procedures.
- Possibilities to integrate multiple processes.
- Shorter delivery times and lower production cost.
Arrowhead
Process and energy system automation

4 years project
68M€
79 partners
Coordinated by an ARTEMIS CoIE

www.arrowhead.eu - jerker.delsing@ltu.se
Collaborative automation

To be demonstrated in real world applications
ISA-95 systems in to the cloud?
Arrowhead approaches

- TCP/IP everywhere, middleware nowhere.
- Internet of Things - IoT
- System of systems - SoS

The Integrating approach
- Service Oriented Architectures - SOA
The global cloud approach

A Survey of Commercial Frameworks for the Internet of Things. Hasan Derhamy, Jens Eliasson, Jerker Delsing, and Peter Priller, SOCNE workshop at ETFA 2015, Luxemburg
Collaborative automation in the cloud

- Automation is local - requirements on:
  - Real time
  - Security and safety
  - Continuous engineering

- Local clouds are beneficial to:
  - Latency - real time
  - Security - supporting safety
  - Less engineering dependencies

- Inter cloud actions are necessary and possibly secure!
Automation using SOA
Demonstrated in e.g.
Socrates and IMC-AESOP projects
Classical automation system characteristics

- Centralised controllers, DCS, SCADA, PLC,
- Pull based - time slotted streaming of all data
- Hard real time
- Design time bindings

Seams to have an upper bound of $X \times 10^5$ I/O’s
Cloud based automation systems

- Choice of centralised or distributed control and data to information computations
- Push on event or pull
- Late binding - runtime binding
- Hard real time?
IoT - properties

- Things comes and goes
- May have limited bandwidth
- May have limited energy supply

- Interoperable services at the device connected to the Internet
- Integration of IoT systems have to be dynamic
  - Based on demand and availability
Expectations on IoT automation

- Integrate any IoT device
- Real time
- Energy consumption
- Engineering
- Trust
  - Secure
  - Safe
- Privacy
  - Migration into/from legacy systems
Cloud integration of any IoT device

- Communication HW
- Existing commercial technology

- SOA
- But which SOA technology
Service Oriented Protocols - A Challenge

<table>
<thead>
<tr>
<th>Application</th>
<th>Semantics</th>
<th>Compression/EXI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot A Service def</td>
<td></td>
<td>CoAP</td>
</tr>
<tr>
<td>Pilot B Service def</td>
<td></td>
<td>DDS</td>
</tr>
<tr>
<td>Pilot C Service def</td>
<td></td>
<td>XMPP</td>
</tr>
<tr>
<td>Pilot D Service def</td>
<td></td>
<td>DPWS</td>
</tr>
<tr>
<td>Pilot E Service def</td>
<td></td>
<td>uPnP</td>
</tr>
<tr>
<td>Pilot A XML def</td>
<td></td>
<td>MQTT</td>
</tr>
<tr>
<td>Pilot B JSON def</td>
<td></td>
<td>OPC-UA</td>
</tr>
<tr>
<td>Pilot C XML def</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pilot D JSON def</td>
<td>HTTP 1.1</td>
<td></td>
</tr>
<tr>
<td>Pilot E XML def</td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPv4/IPv6/IP multicast</td>
<td></td>
</tr>
</tbody>
</table>
One Service Oriented Protocols - Works!

Application

<table>
<thead>
<tr>
<th>Pilot A Service def</th>
<th>Pilot B Service def</th>
<th>Pilot C Service def</th>
<th>Pilot D Service def</th>
<th>Pilot E Service def</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot A XML def</td>
<td>Pilot B JSON def</td>
<td>Pilot C XML def</td>
<td>Pilot D JSON def</td>
<td>Pilot E XML def</td>
</tr>
</tbody>
</table>

Semantics

Compression/EXI

<table>
<thead>
<tr>
<th>CoAP</th>
<th>DDS</th>
<th>XMPP</th>
<th>DPWS</th>
<th>uPnP</th>
<th>MQTT</th>
<th>OPC-UA</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>UDP</th>
<th>HTTP 1.1</th>
<th>TCP</th>
</tr>
</thead>
</table>

IPv4/IPv6/IP multicast
What about service protocol interoperability

Is it possible to make machine assisted translation like

- CoAP -> XMPP
- CoAP -> MQTT
- CoAP -> OPC-UA
- OPC-UA -> CoAP
- OPC-UA -> MQTT
- Necessary semantics translation
- Necessary data structure translations
- Service integrity over protocols, data structures, semantics etc.

Hasan Derhamy, Pal Varga, Jens Eliasson, Jerker Delsing and Pablo Punal Pereira
Translation Error Handling for Multi-Protocol SOA Systems, ETFA 2015, Luxembourg
Hard real time IoT cloud

- Hard real time dependent on underlaying communication capabilities
  - Local hard real time cloud to prescribe communication technology
    - e.g. Industrial ethernet, TTTech, time slotted 802.15.4

- SOA overhead eats bandwidth
  - Use compression
  - EXI

EXIP: A Framework for Embedded Web Development
In : ACM Transactions on the Web. 8, 4, 29 p.23
Real time local cloud automation & inter cloud automation
IoT energy consumption

- IoT devices may be battery powered
- Event orientation
  - Reduces cost of communication
  - No streaming of IoT data to cloud
- IoT data/info. to consumer on configured event
  - Distributed data -> information computation
  - Subscription to distributed information based on events
  - Enabling receiver tailored information
Engineering of IoT automation systems
System of systems, SoS, approach

- Information provided as a configurable services
- Orchestration of services
  - Supported by complex event processing
- Choreography
- To be supported by the technology architecture
How to build local cloud?
SOA - Abstracting IoT data to services

Services are produced
Services are consumed

Service producer

Application service

Service Consumer

IoT System A

IoT System B

Exchange information
How to build local cloud?
Fundamental conceptual overview

[Diagram of Arrowhead Framework core and application services]

- Core Systems and Services
  - Authorisation System
  - Service Registry
  - Orchestration System

- Application Systems and Services
  - Service providing system
  - Service consuming system

Included in the Arrowhead Framework

www.arrowhead.eu
Core Functionalities - System-of-Systems in a local cloud

ARROWHEAD FRAMEWORK COMPLIANT NETWORK

Application system

Core systems

Application system

Application system

Application system

Application system
Automation orchestration
local & inter cloud
Three mandatory local cloud services

- Service registry system
- Authorisation system
- Orchestration system
Service Registry

- supports a service registry functionality based on DNS and DNS-SD.

- all Systems within the network shall publish its producing service within the Service Registry by using the Service Discovery service

The Service Registry system consist of all active producing services within the network.
Authorisation System

• Authorisation Management service provides the possibility to manage the access rules for specific resources.
• Authorisation Control service provides the possibility to control the access for an external service to a specific resource.
• Service Discovery service uses the Service Discovery to publish the Authorisation Systems producing services within the Service Registry System.

The Authorisation System consists of access rules to system resources (i.e. services).
Orchestration System

• Orchestration Management service provides the possibility to manage the connection rules for specific services.
• Orchestration Store service provides the possibility to fetch configuration for a system.
• Service Discovery supports the publishing of the Orchestration Systems producing services within the Service Registry System.

The Orchestration System provides the functionality of manage connection rules (i.e. orchestration of the system of system composition).
Startup Application System B and establish connection
Automation support services
Arrowhead core systems

- Factory description system
- Deployment system
- Configuration system
- Event handler system
- Historian system
- Meta service registry system
- User registry system
- Quality of Service system
Factory description system

The purpose of the Plant description system is to provide a way to find Arrowhead devices and systems through browsable structures based on the physical systems the Arrowhead devices are connected to.

The first specification of this system is intended as a basic interface to present hierarchies and basic information about each object. It is intended to allow a user to find objects, physical or Arrowhead systems, based on either their physical location or based on their place in a functional context.
Configure system

As the devices running Arrowhead compliant systems are loosely coupled and provided by different suppliers the engineering is expected to move to open or independent engineering platforms rather than those provided by hardware manufacturers. The Configuration system allows the configuration of systems from different system suppliers through a uniform service interface. The Configuration system is designed so that the configuration possibilities are not limited by the service interface but allows all configurations that the configurable system is set to allow.
Deployment System

The purpose of the Deployment system is to automatically join pre-assigned new devices to a specific Arrowhead Framework enabled cloud and save installation/engineering time.

The idea is to allow an administrator of the local cloud to set conditions under which a factory issued identification key can be used to authenticate certain systems to allow distribution of more specific keys which then allows a system to connect to the Arrowhead framework without any detailed administration of the specific system.
Event Handler

The Event Handler system searches and connects to published services of the type EventLog in the ServiceRegistry.

If a system have registered, by use of the EventNotification service, to listen on some specific type of event or system that log events, it will be notified of the specific event when it arrives at the EventLog service interface.
Historian
The Historian is used for storing large amounts of sensor data, as well as distributing messages from resource constrained devices to a large number of clients. The built-in support for Arrowhead Events enables the Historian service to log events and act as an intermediated event cache for device to device or service to service interaction. Thus the Historian behaves like a network cash for data from resource constrained devices.
Meta Service Registry

The Meta-Service system stores additional information about a service for offline/later access.

This system is a support system for the service registry for store additional information such as constraint information, up-time, or other specific information that can be valuable for the usage.
Arrowhead Meta Service registry

The Arrowhead MSR is primarily designed to work with resource-constrained and battery powered wireless devices, and contains metadata about services and devices, such as:

- Battery level, renewable energy sources
- Signal strength, network topology, current access point
- Bandwidth requirements and low-latency real-time communication using QoS
- Uptime, no reboots,
- Software and hardware revision, manufacturer
- etc.
User / System Registry system

The User-System Registry system holds unique system identities for deployed systems within the Arrowhead network.

«System»
UserSystem Repository

«CP» DNS-SD
ServiceDiscovery

«CP» REST_WS-TLS-XML
UserSystemDiscovery
Quality of Service

The Quality of Service (QoS) approach takes care of handling requests from Service Consumers in order to guarantee the reservation of the network and/or computational resources and to give delivery guarantees to the communications with Service Producers.
Migration into/from legacy systems

Migration of SCADA/DCS Systems to the SOA Cloud
Experiments made
Socardes and IMEC-AESOP projects

- Boliden 2011
  - Control over wireless link

- Hydraulic control at damm in Tampere 2013
  - PLC in a global cloud

- LKAB 2013
  - SCADA in a local cloud
Necessary technology for large automation systems in the cloud

Robust communication, wired or wireless
IoT sensors, actuators, PLC:s, etc.
DCS and SCADA functionality’
MES and ERP functionality

Cloud integration technology
Engineering tools for cloud automation systems
Test tools and simulators for debugging
Migration of cloud automation into legacy production system
Suitable security
Engineering tools for cloud automation systems
Development support, documentation.

SoSD: System-of-Systems Description
SoSDD: System of Systems Design Description
SysD: System Description
SysDD: System Design Description
SD: Service Description
IDD: Interface Design Description
CP: Communication Profile
SP: Semantic Profile
Development tools
System Management tool
Service security
IoT and cloud security

- Security at service level
- Certificates
- Tickets
- Data encryption
- IPsec
- TLS
- System security validation methodologies

4. Authentication Process

![Authentication Process Diagram]

4.1 Authentication Method

On the authentication process the server must recognize the user as a valid user and communicate that to the CoAP-NAS. This process needs to be flexible and compatible with other standards and with this goal the propose framework creates a public login CoAP service on the CoAP-NAS. This login service must receive a PUT request with one of the following contents as a payload:

- User name and password as plain text. This option is only recommended during testing, debugging and development phases.
- User name and password hash. This is easy to implement and could be authenticated directly on the CoAP server (without RADIUS).
- A RADIUS packet (future work).

The possibility to run RADIUS protocol over CoAP (see section 2.4) gives to the framework a flexible authentication method usable with a standard RADIUS server.

Test tools for cloud automation.
Automation engineering time

- Automation is a service based on products
- Simplicity of automation service engineering is market key
- Arrowhead Framework reduces engineering time
  - From 5-6 days -> 6-8 hours (Abelko)
Can we build Arrowhead automation systems today?

Robust communication
IoT sensors, actuators, PLC:s, etc.
DCS and SCADA functionality
MES and ERP functionality
Cloud integration technology
Engineering tools cloud automation
Test tools and simulators
Migration to cloud automation
Suitable security

➡Products on the market
➡Some products on the market
➡First products on the market
➡Demonstrated in industrial env.
➡Some products on the market
➡Demonstrated in industrial env.
➡First products on the market
➡Demonstrated in industrial env.
➡First products on the market
Lift micro grids

- Renewable - PV at building roof
- Recovery from lift operation
- Grid supply
- Use of 3 shared services: energy tariffs, prediction, energy planning
- Energy savings up to 65%
Water distribution grid

- Use of prediction service enables flexibility in energy demand
- Energy savings 15%
Load balancing - Luleå Sweden

- Adaptive control curve service
- Load balancing of individual building peak energy demands service
- Multi site optimisation service
- Interacting with load balancing and the adaptive control curve
- Stena (housing company) claims 5% savings in energy usage.
Arrowhead Framework

- Public by fall 2015
- Documentation
- Cookbook
- Support wiki
- Core system code
- Tools - Open source and commercial
- Sample automation services - code
Critical platform technologies

Security - scalable and flexible security solutions
Latency - how provide "clouds" with latency "guarantees"
Dynamics/Continuous - engineering, configuration and deployment
Scalability - for massive numbers of resource constrained IoT and CPS devices
Critical system properties

- Trust in cloud automation systems
- Real life - at scale - demonstrators enables
  - Standards,
  - Society and political acceptance
Conclusions

- Very large scale IoT system of systems is desired
- Critical automation trust requires
  - Latency control and Security
  - Scalability
  - Ease of continuous engineering
- Solutions enabling dynamic automation systems:
  - Design and Engineering
  - Deployment, Operation and Maintenance
Arrowhead.eu
an
Artemis and ProcessIT.EU project

jerker.delsing@ltu.se