Service oriented Architecture results from Arrowhead and its usage in EMC2

Professor Jerker Delsing
Luleå University of Technology, Sweden

ARROWHEAD

Building Automation Systems from IoT

Arrowhead Framework: concepts and basic architecture

Professor Jerker Delsing
Luleå University of Technology, Sweden

ARROWHEAD

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

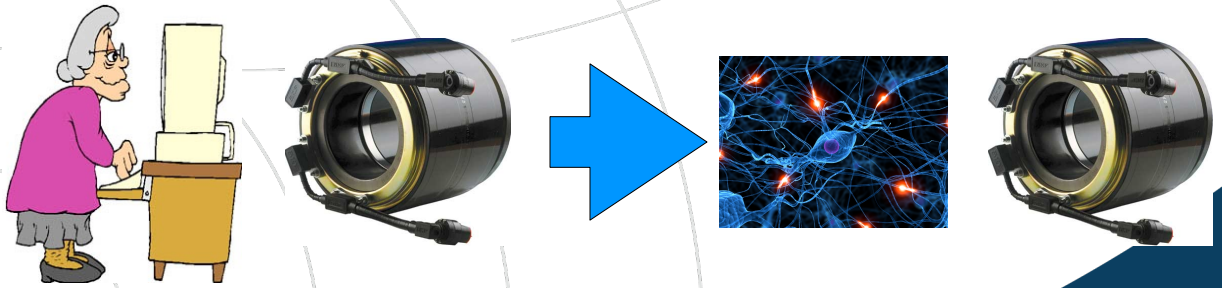
One customer, KGHM, one component

- **120 km conveyers**
- **720.000 idler bearings**

ARROWHEAD

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



ARROWHEAD

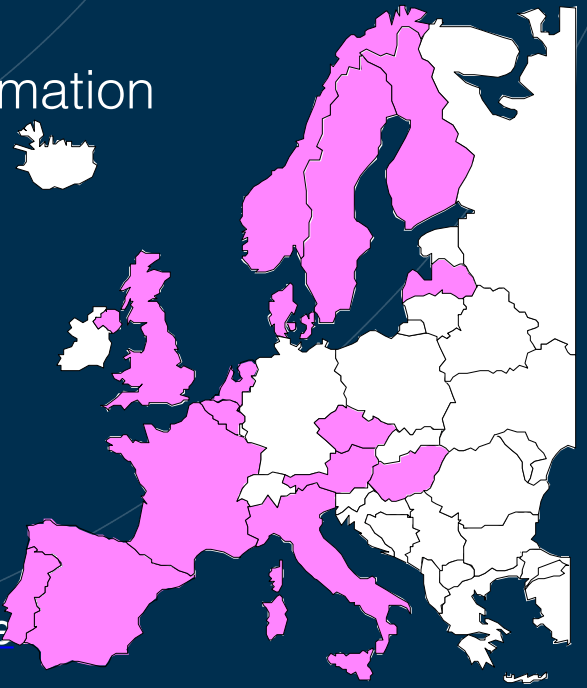
Arrowhead

Process and energy system automation

4 years project
68M€
78 partners
Coordinated by

processit.eu
Automation for process industries

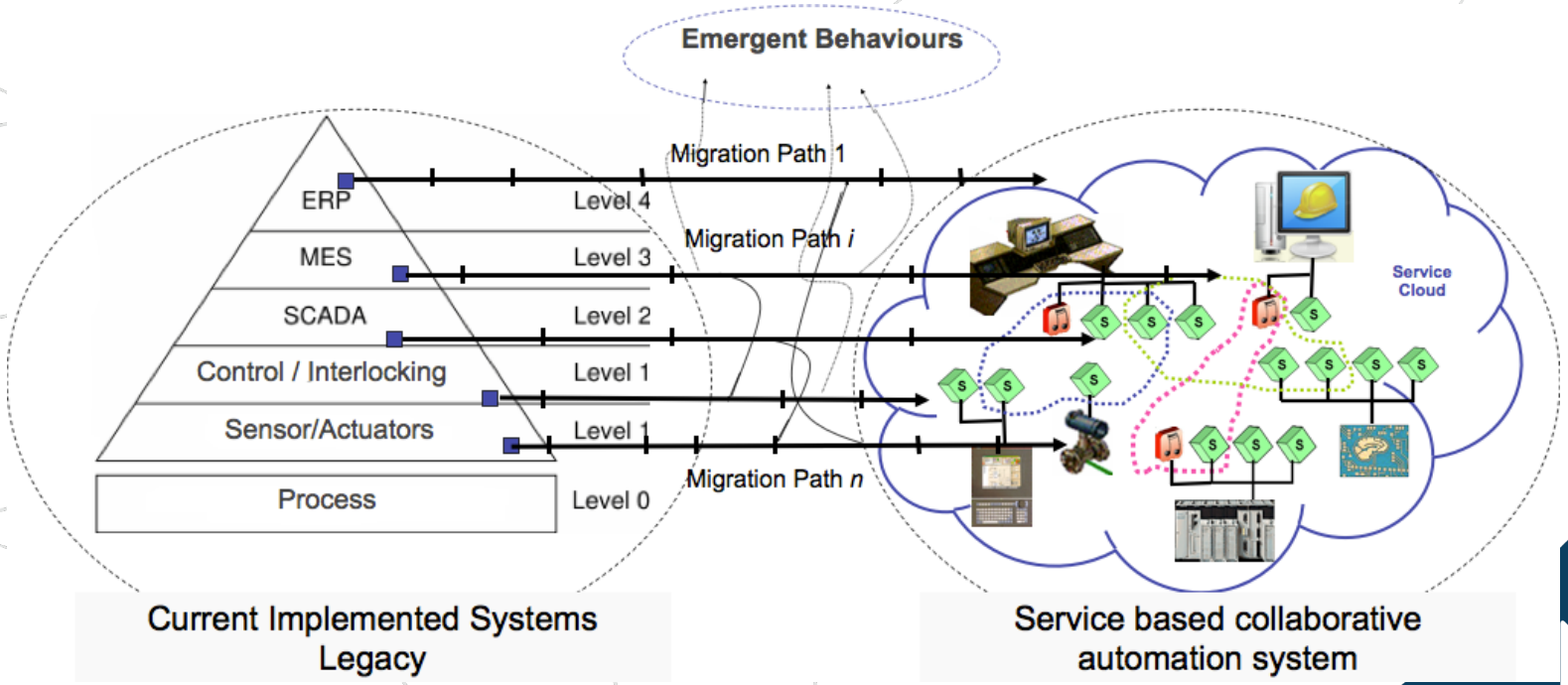
an ARTEMIS CoIE



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

ARROWHEAD

ISA-95 systems in to the cloud?



ARROWHEAD

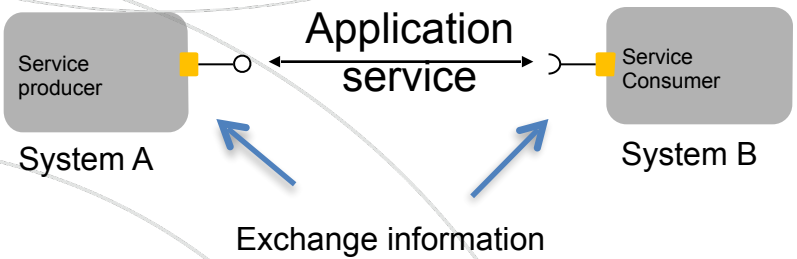
Arrowhead approaches

- **TCP/IP** everywhere, middleware nowhere.
  - Internet of Things - IoT
  - System of systems - SoS
- The Integrating approach
  - Service Oriented Architectures - SOA

ARROWHEAD

SOA

Services are produced
Services are consumed



ARROWHEAD

SOA Key properties

- Loosely coupled
  - Autonomy
  - Distributed
  - Owner is responsible and owns the information and decide whom to share to
- Late binding
  - Possible to use information anytime by connect to the correct resource at a given time
- Lookup
  - Publish and register for notify others about endpoint (how to reach me)
  - Discover others that I comply to (expected/wanted Service Type)

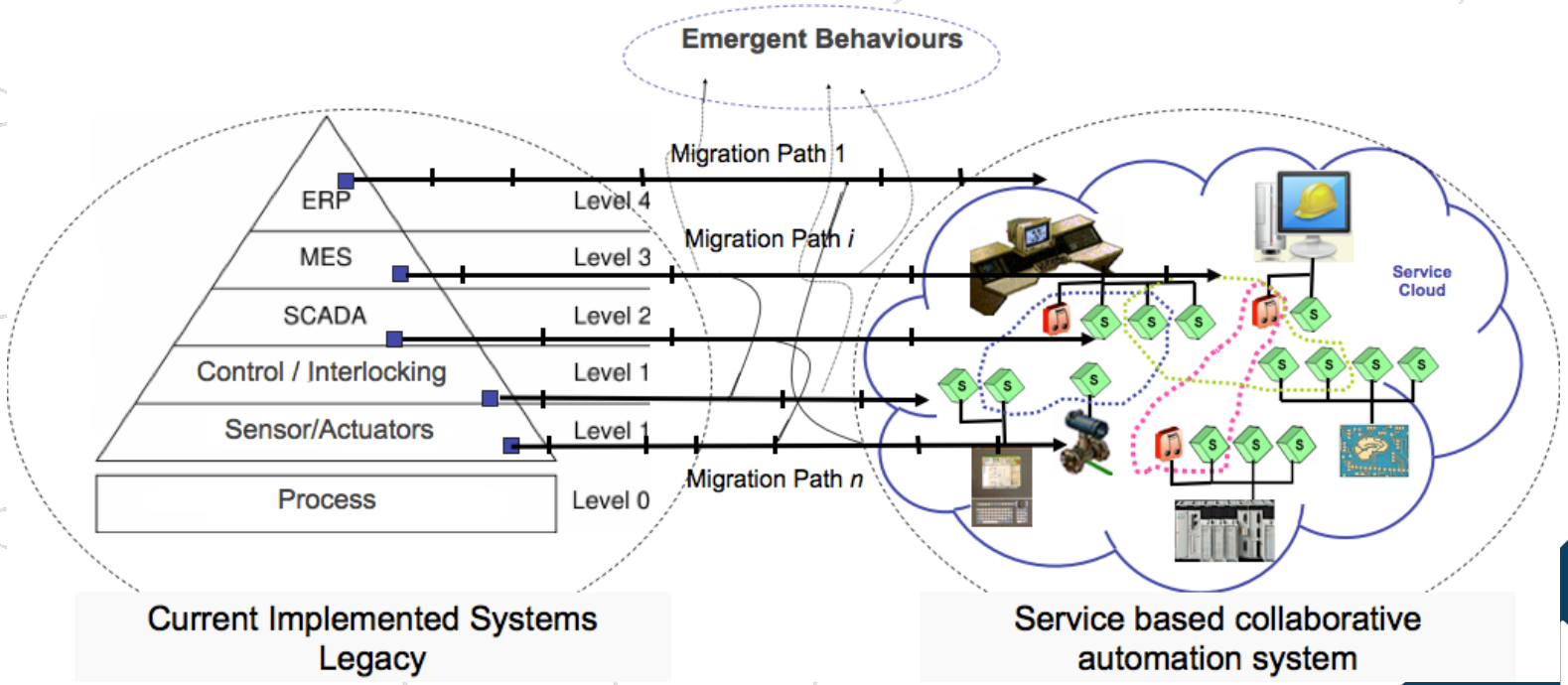
ARROWHEAD

Fundamental approaches in Arrowhead

- Information centric
- Information assurance at service exchange level
- Publish subscribe approach
- Push approach - Pull possible
- Minimal set of mandatory services in a System of Systems

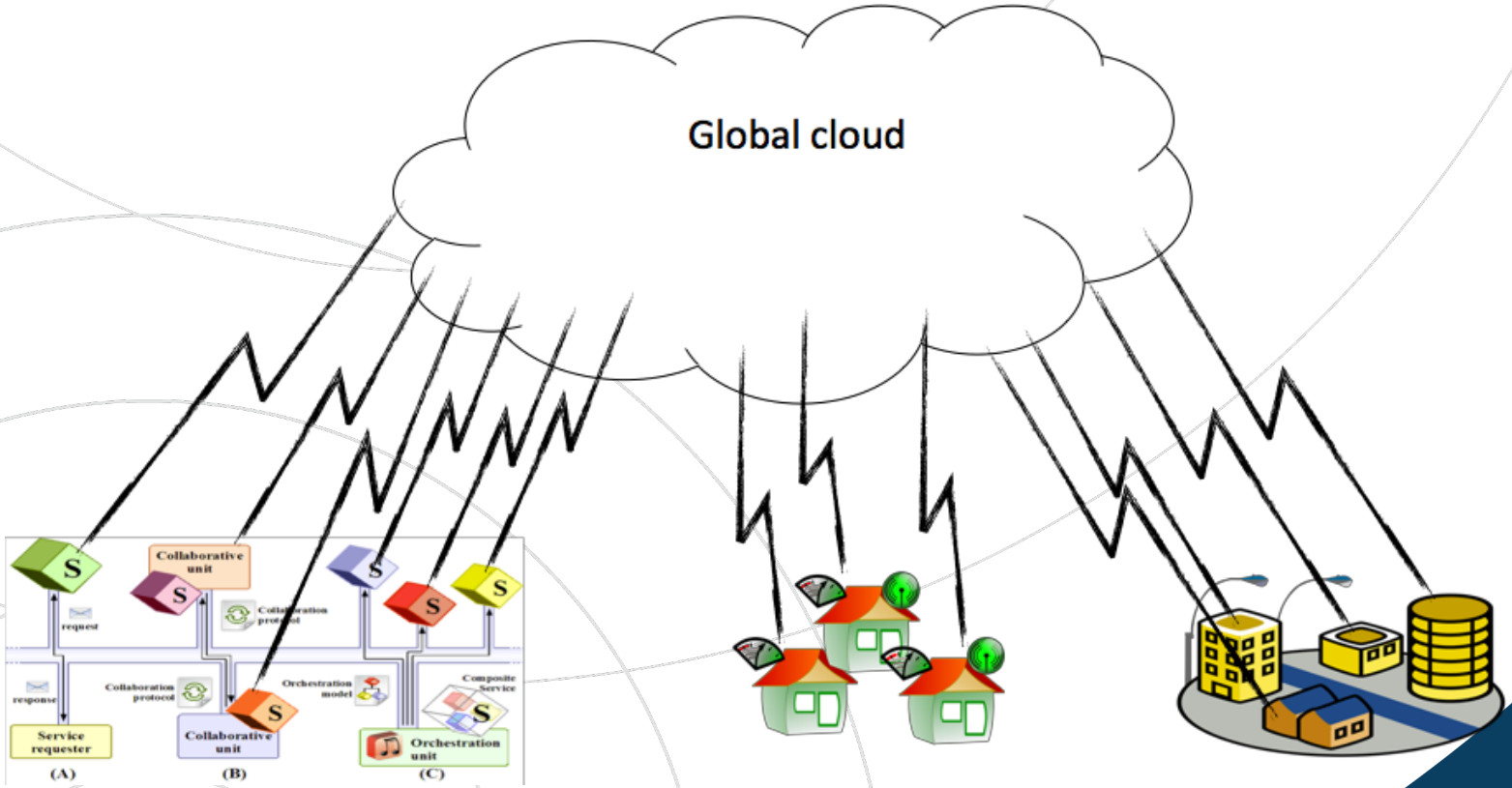
ARROWHEAD

ISA-95 systems in to the cloud?



ARROWHEAD

The global cloud approach

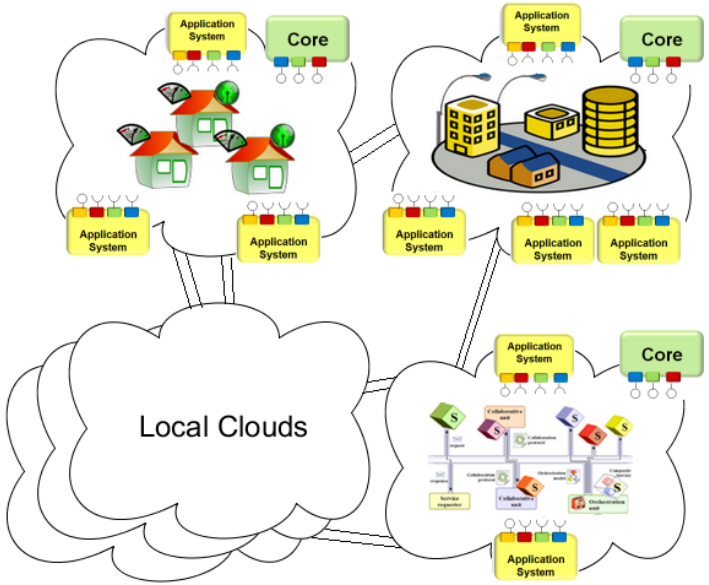


A Survey of Commercial Frameworks for the Internet of Things. Hasan Derhamy, Jens Eliasson, Jerker Delsing, and Peter Priller, SOCNE workshop at ETFA 2015, Luxemburg

ARROWHEAD

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!



ARROWHEAD

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 $X*10^5$ I/O's

ARROWHEAD

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?

ARROWHEAD

IoT - properties

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

- Integration of IoT systems have to be dynamic
  - Based on demand and availability

ARROWHEAD

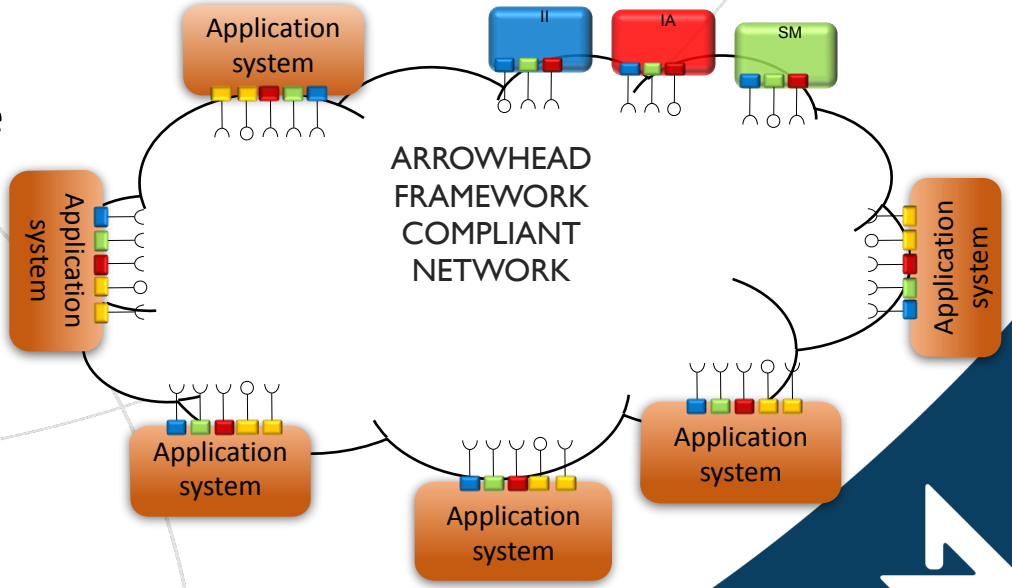
Cloud integration of any IoT device

- Communication HW
  - Existing commercial technology
- Providing support for local clouds
  - Arrowhead Framework: mandatory core systems
- Providing support for automation functionalities:
  - Arrowhead Framework: core systems
- Providing support for multiple SOA protocols
  - Protocol translation system

ARROWHEAD

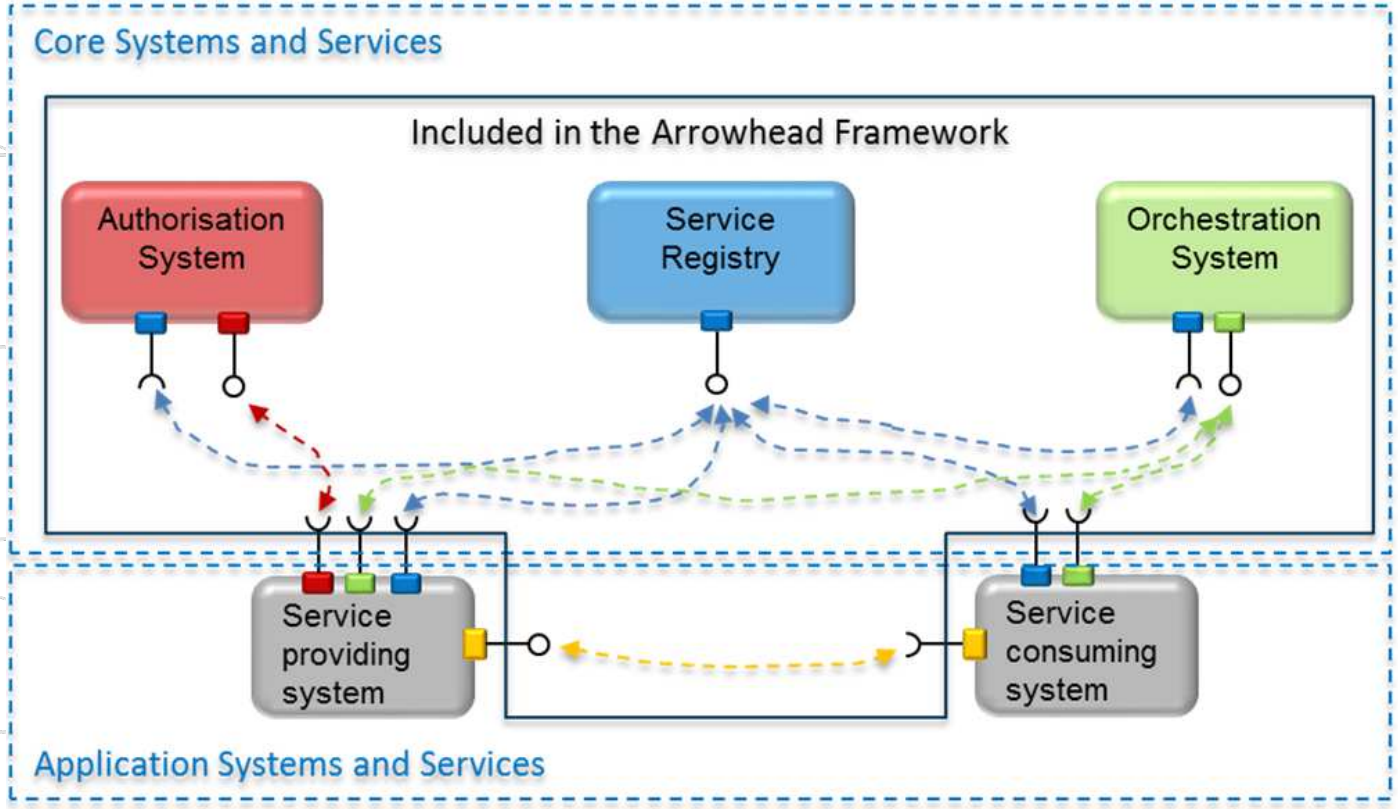
Arrowhead Framework - support for: System of systems in a local cloud

- Mandatory core systems:
  - Information infrastructure
  - System management
  - Information assurance



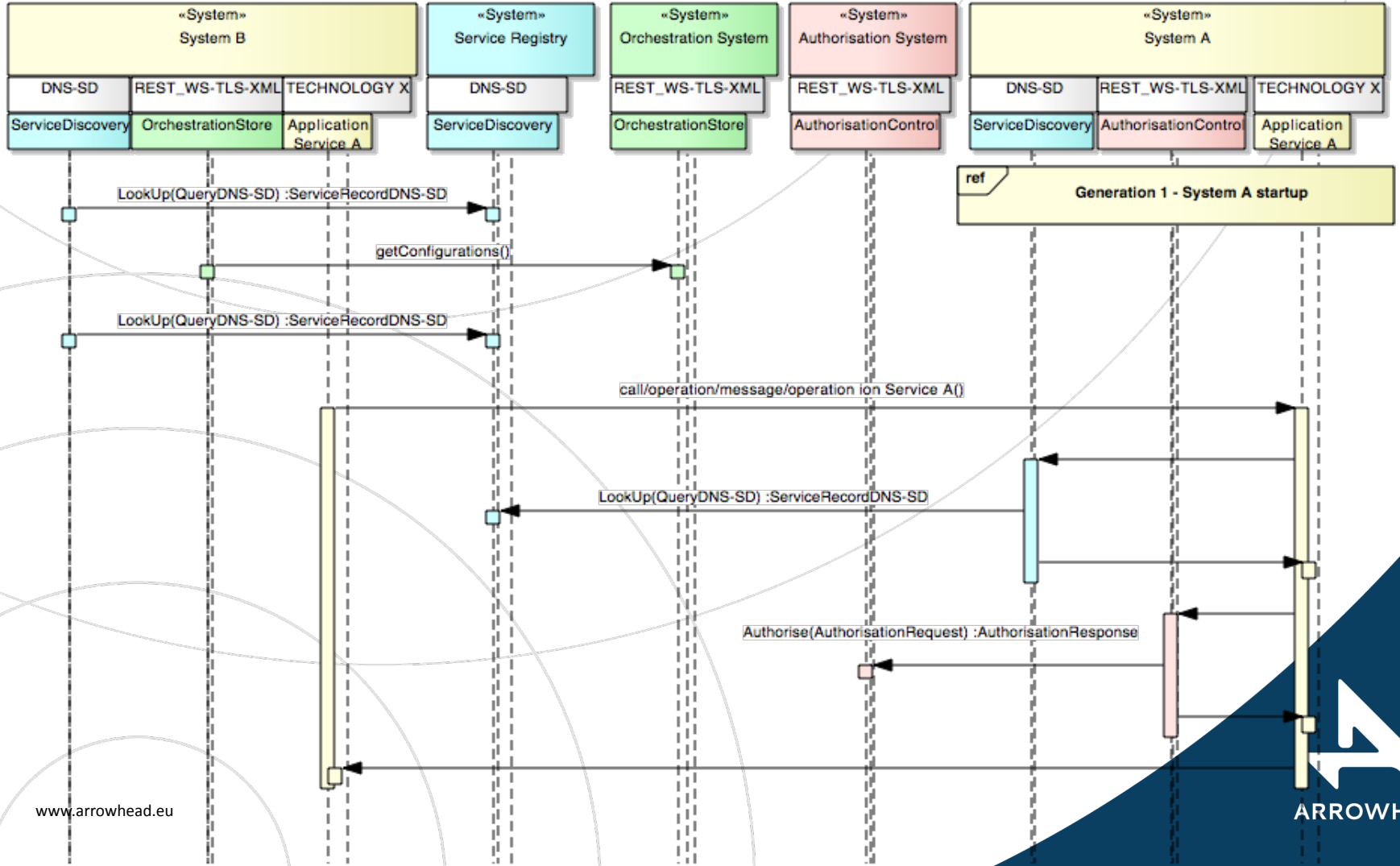
ARROWHEAD

How to build local cloud?
Fundamental conceptual overview



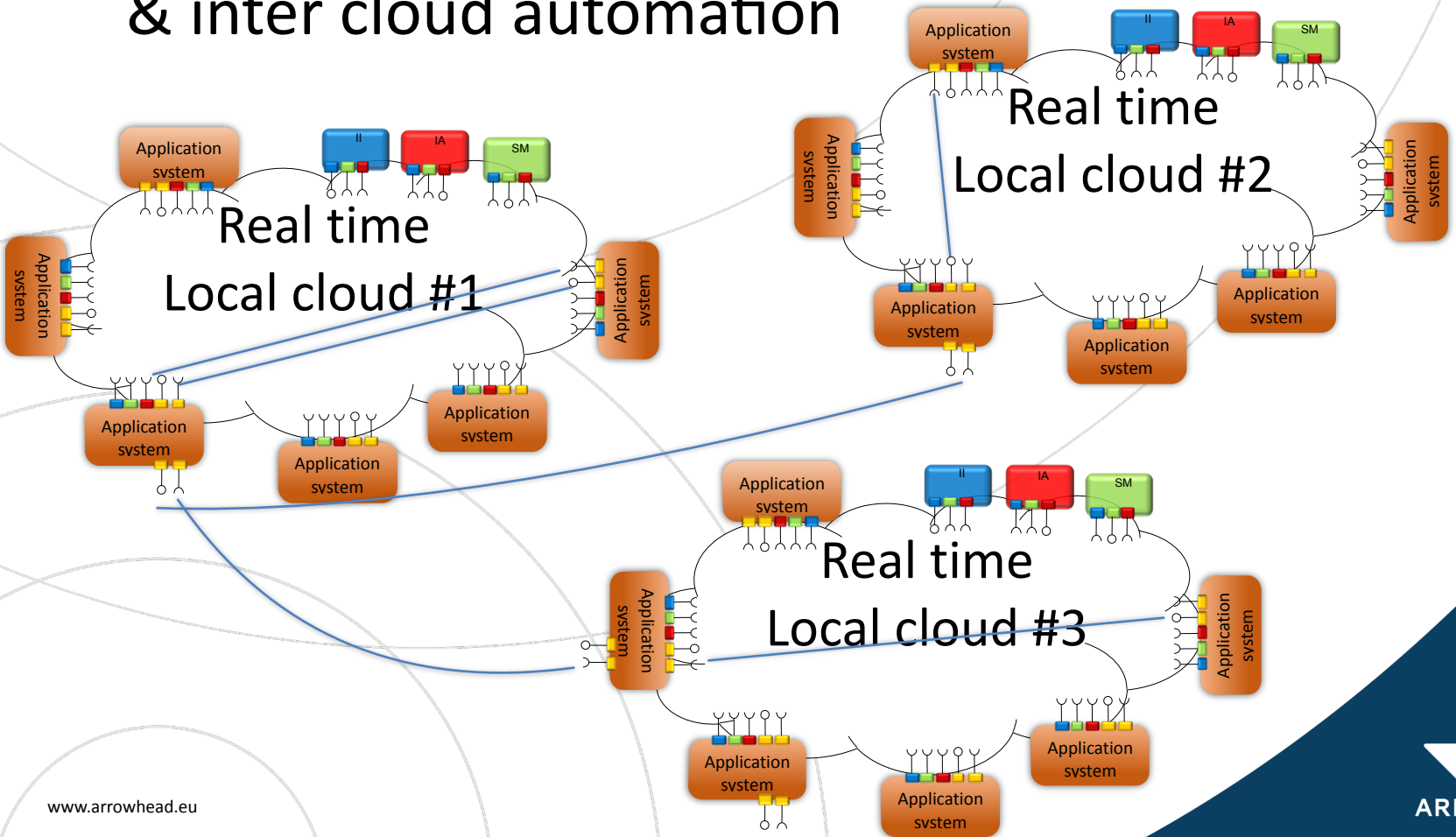
ARROWHEAD

Startup Application System B and establish connection



ARROWHEAD

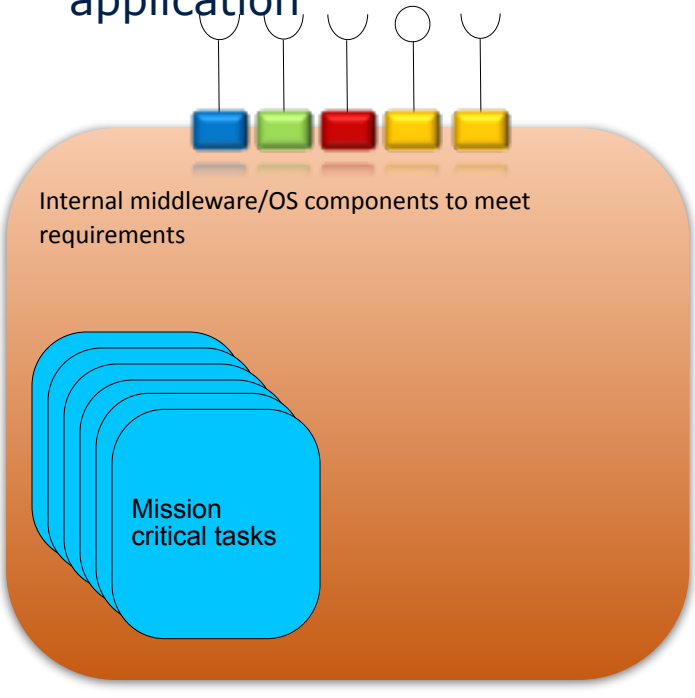
Real time local cloud automation & inter cloud automation

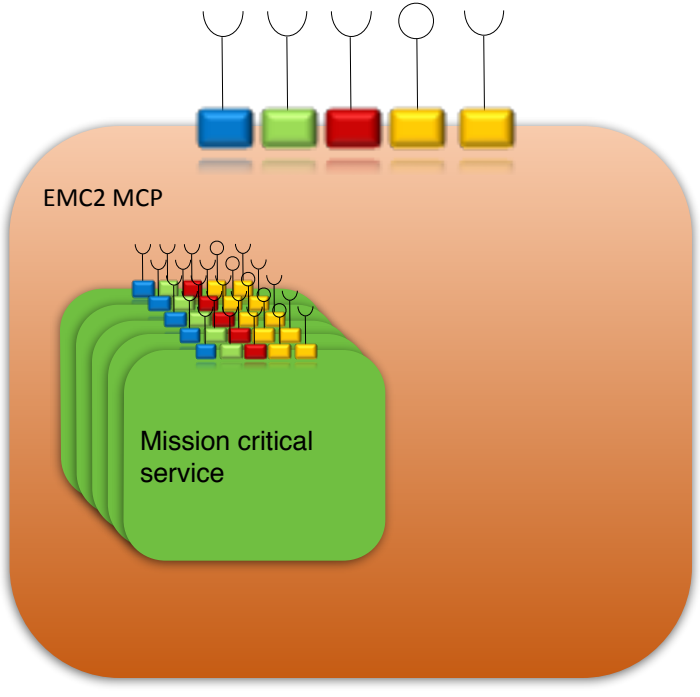


ARROWHEAD

EMC² architectural aspects

- Two possible principles
  - EMC2 service paradigm between any service executed on a single MCP, **a cloud on a chip**.
  - EMC2 services only as a shell to an MCP running multiple application





Hard real time local IoT clouds

- 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
Kyusakov, R., Punal, P., Eliasson, J. & Delsing, J. Oct 2014
In : ACM Transactions on the Web. 8, 4, 29 p.23

ARROWHEAD

Arrowhead core automation systems

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

ARROWHEAD

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

ARROWHEAD

Can we build Arrowhead automation systems today?

Robust communication	➡Products on the market
IoT sensors, actuators, PLC:s, etc.	➡Some products on the market
DCS and SCADA functionality	➡First products on the market
MES and ERP functionality	➡Demonstrated in industrial env.
Cloud integration technology	➡Some products on the market
Engineering tools cloud automation	➡Demonstrated in industrial env.
Test tools and simulators	➡First products on the market
Migration to cloud automation	➡Demonstrated in industrial env.
Suitable security	➡First products on the market

ARROWHEAD

Arrowhead Framework

- Public by fall 2015

- Wiki at forge.soa4d.org/projects/arrowhead-f
  - Documentation
  - Cookbook
  - Mandatory core systems: images and code
  - Tools
    - System management
    - Test tool
  - Sample simple service - code
  - Sample automation services - code

Selected/interested WP1 partners to be invited for pre-usage!

ARROWHEAD

The Arrowhead Framework map

Why & How

- Objectives
- *Requirements ????*
- Strategy

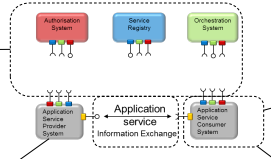
How to implement an Application system

- Java library API for Core services usage. (SW Download)
- Application system Example – Using the Core system via the Java library, and example Application services. (SW Download)
- Application system examples (separate "group") Some "easy examples /Pal"
- Design doc and code.
- Federico examples… (wrapping API)

Core Services & Systems

- Core Service specifications
- Core System specifications
- Core System prototypes (SW Download)
- On Line deployed clouds – access guide
- Local deployment guidance

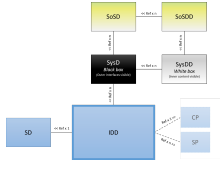
Technical Architecture



Application Services

- Service specifications

Technical documentation model



Do I comply?

- Compliance Criteria
- Test Tool (SW Download)
- Interoperability matrix

EMC^2 requirements

Information assurance

Authentication & Authorisation system - Arrowhead

Data encryption system - Arrowhead?

Security intrusion system (CEP)

Event handling system (CEP) Arrowhead

Certificate distribution system

Historian - Security logging

Historian system Arrowhead

Security vulnerability analysis system

System Management

"Plant" description system - Arrowhead

Deployment system - Arrowhead

Configuration system - Arrowhead

Orchestration system - Arrowhead

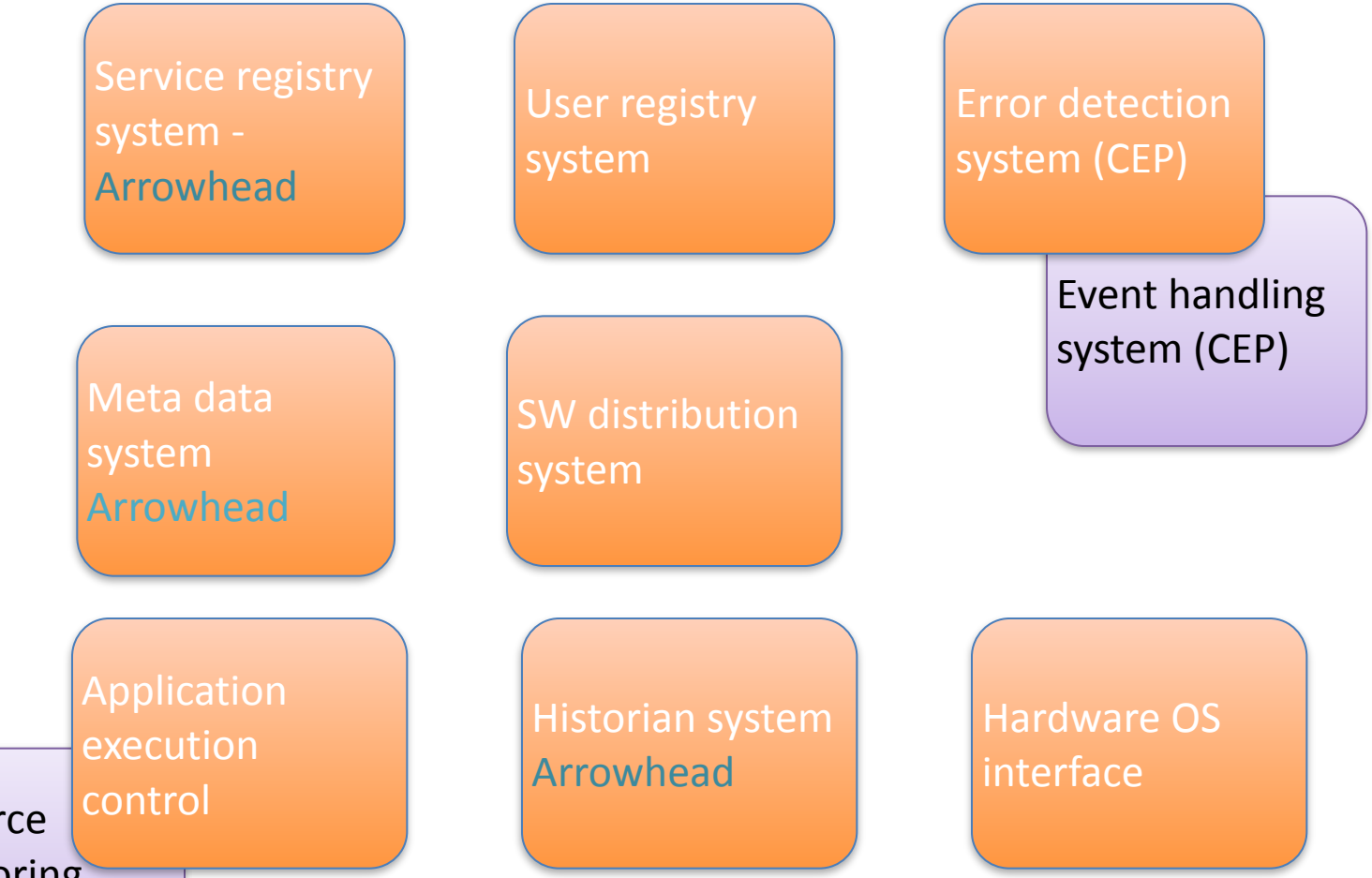
Resource monitoring system

Event handling system (CEP) - Arrowhead

Quality of Service system- Arrowhead

Historian system - Arrowhead

Information Infrastructure



Conclusions - Critical platform properties

- 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

Thanks for listening