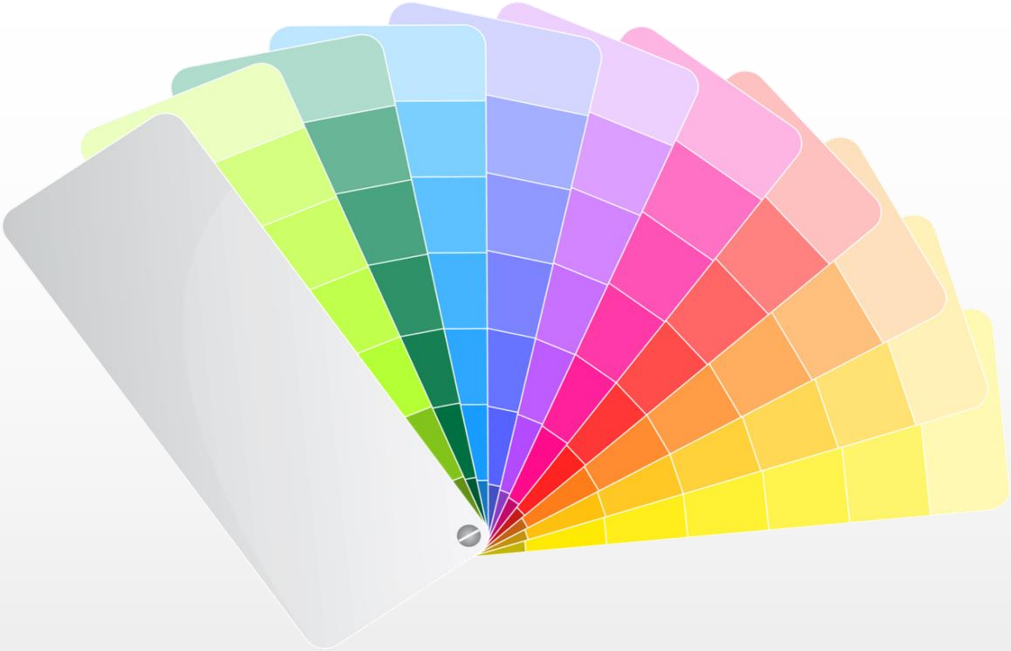


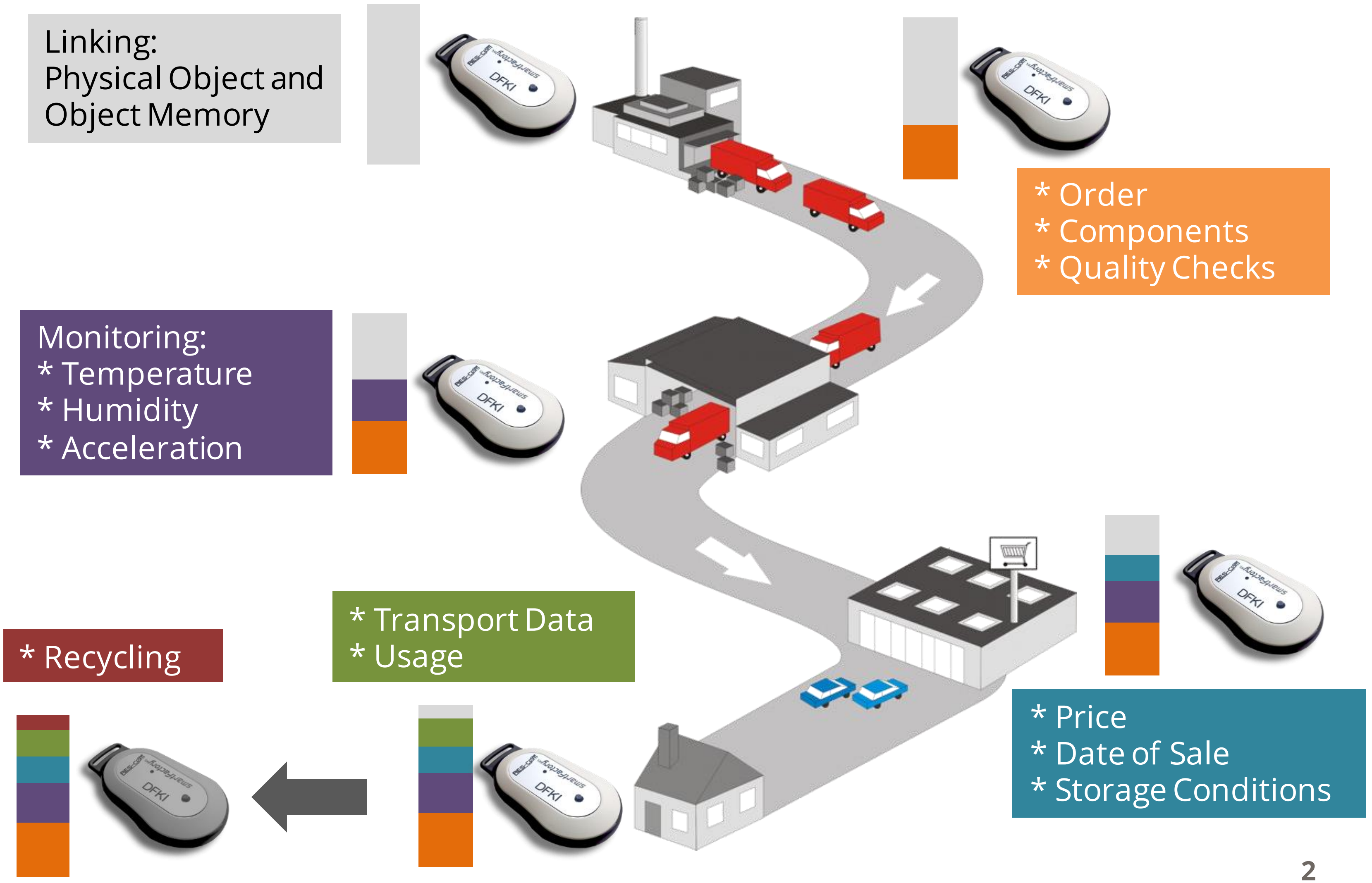
Digital Object Memories for the Web of Things

Dr.-Ing. Jens Haupert

German Research Center for Artificial Intelligence (DFKI)

Web of Things Workshop, June 25th 2013

Sample Scenario



Linking:
Physical Object and
Object Memory

* Order
* Components
* Quality Checks

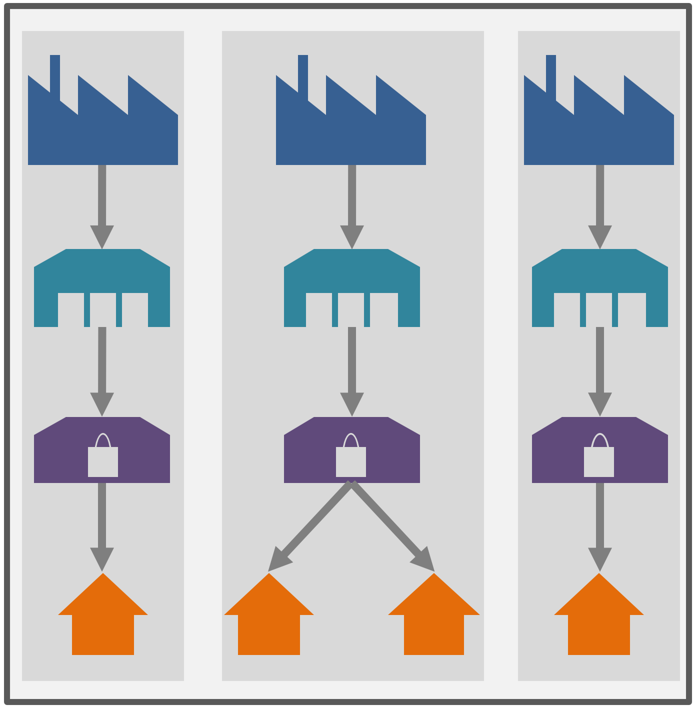
Monitoring:
* Temperature
* Humidity
* Acceleration

* Recycling

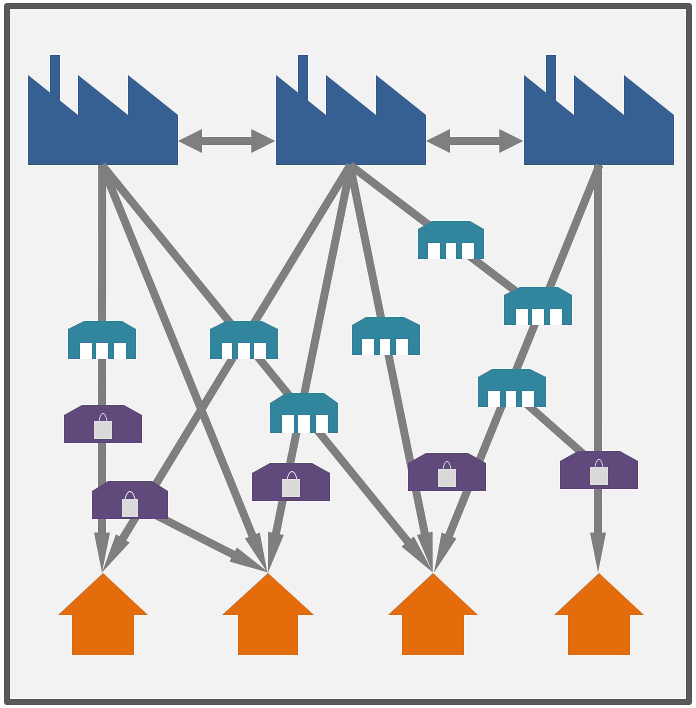
* Transport Data
* Usage

* Price
* Date of Sale
* Storage Conditions

Open-Loop Scenarios



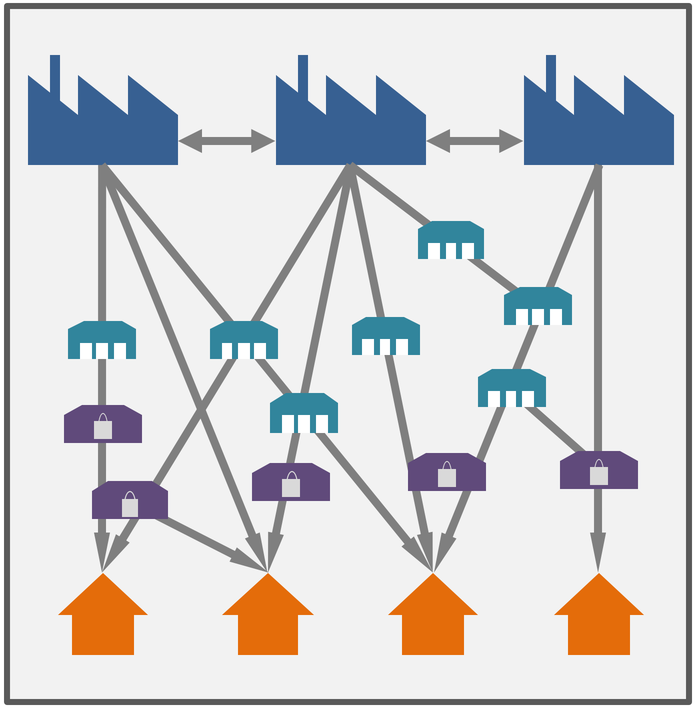
Closed-Loop



Open-Loop

Open-Loop Scenarios (2)

- A large set of stakeholders
- Ad-hoc lifecycle chain
- No a priory definitions
- Heterogeneous datasets

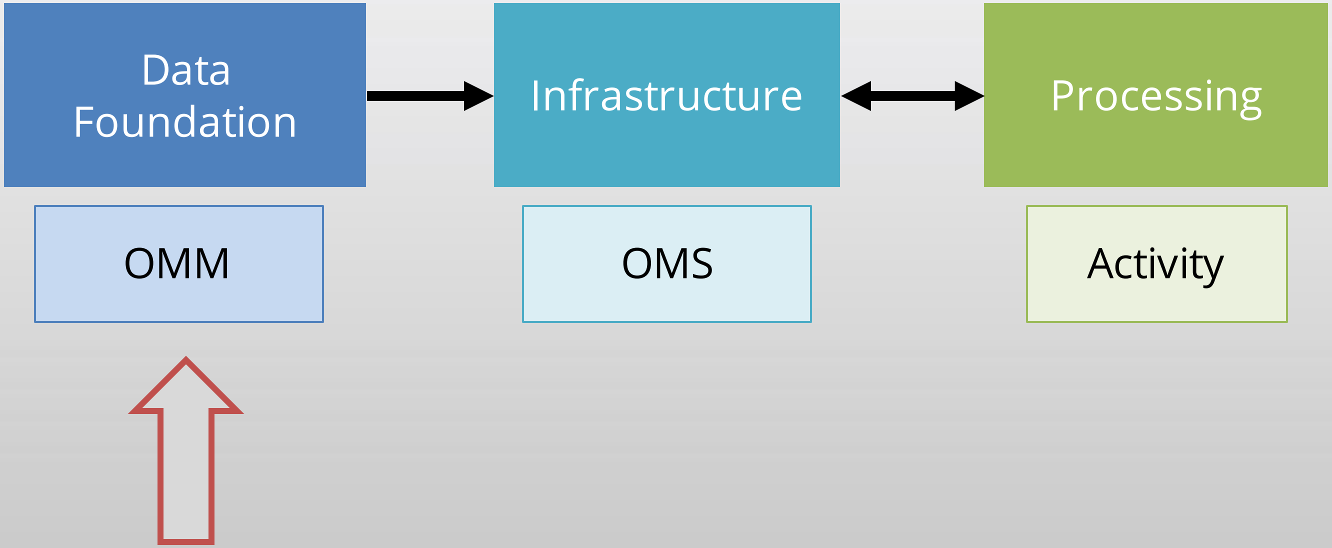


Open-Loop

New Requirements

- Life Cycle Chain
  - Object-related information
  - Open-loop distribution
- Maintenance
  - Specific configurations / Lot size one
  - Retrieving spare parts
- After Sales
  - (Remote) condition-based monitoring
- Improved Recycling
  - Provide information about components and elements
  - Recycling feedback to manufacturer

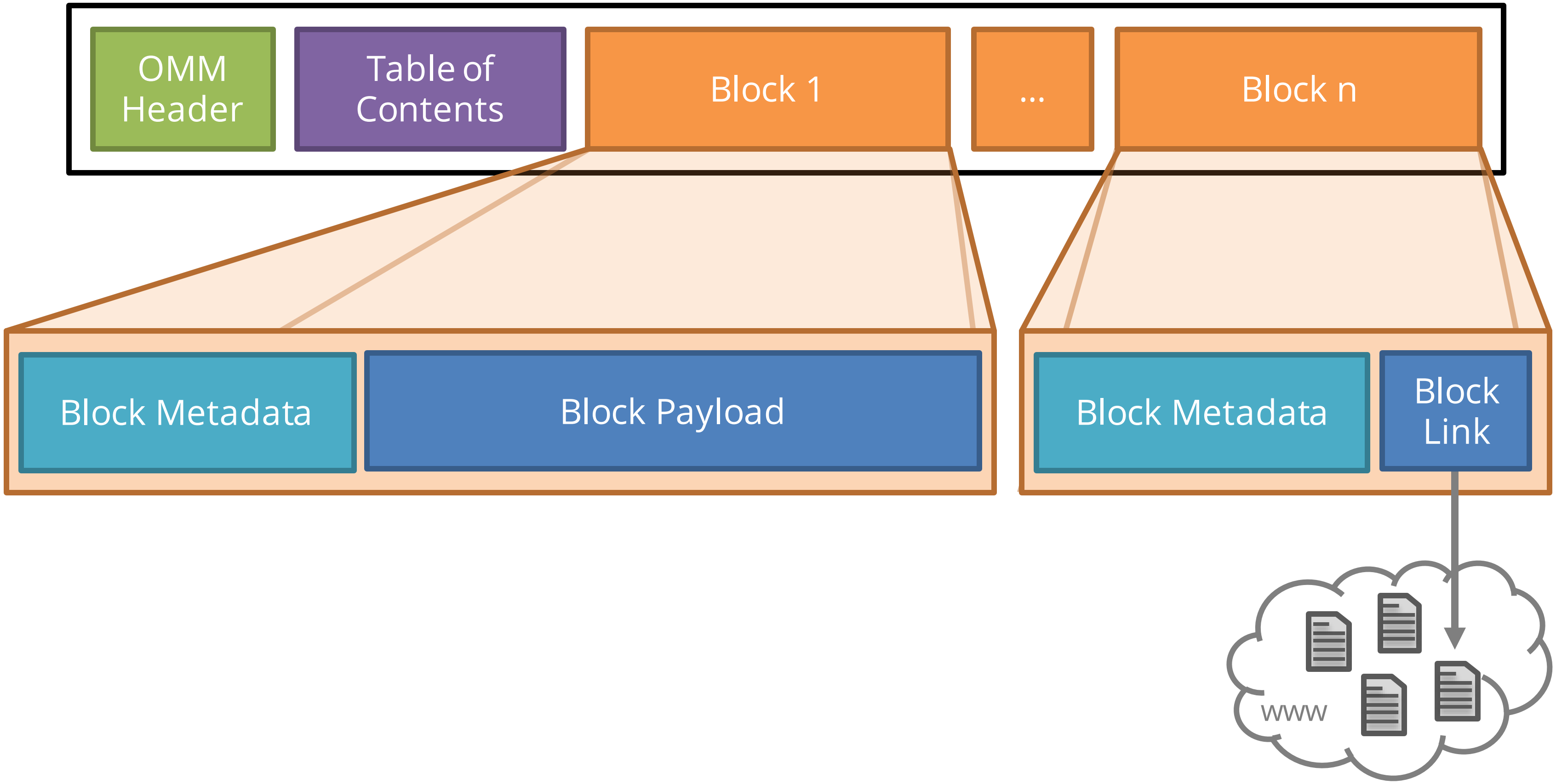
Data Foundation



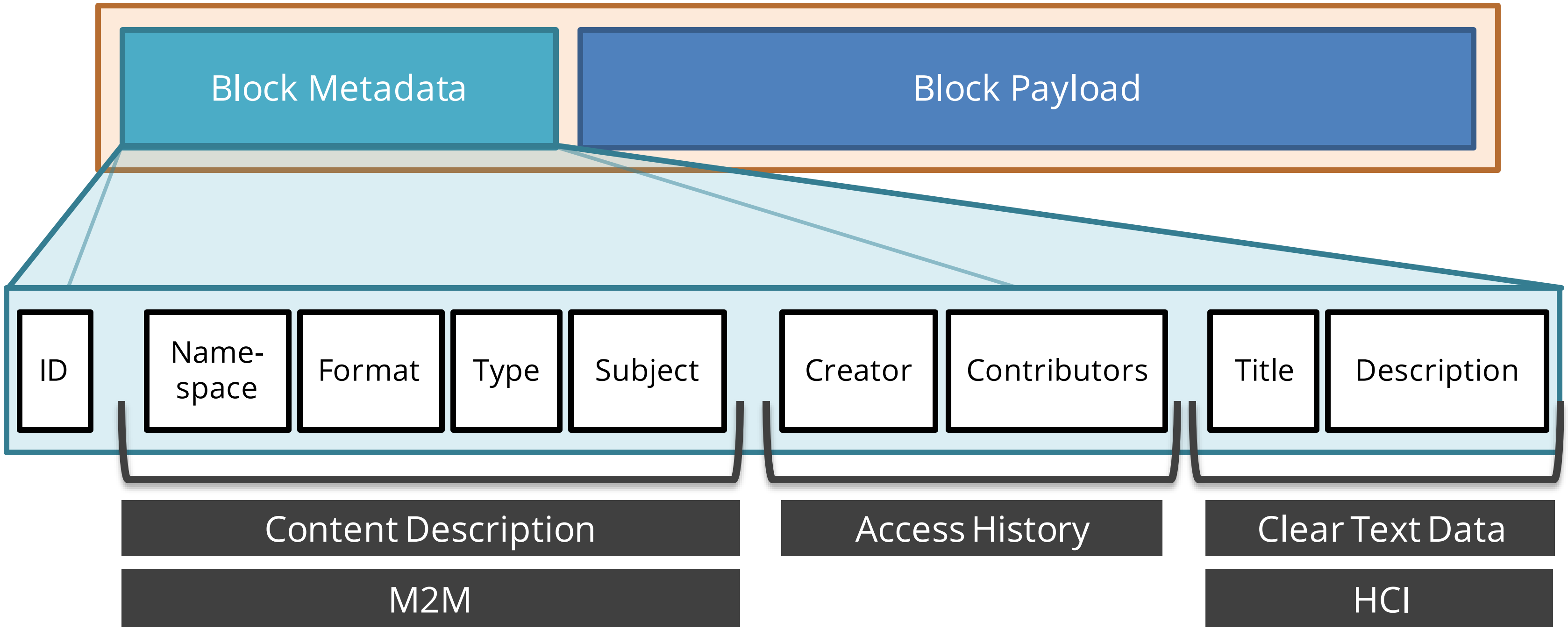
Object Memory Model

- *Structure model* to partition heterogeneous object memory data
- Group associated data to *blocks*
- Each block consists of actual *payload* and additional *meta data*
- Payload can be out-sources to *external media*
- Abstract model with XML- and embedded HTML5-representations
- W3C® Incubator Group (OMM-XG)

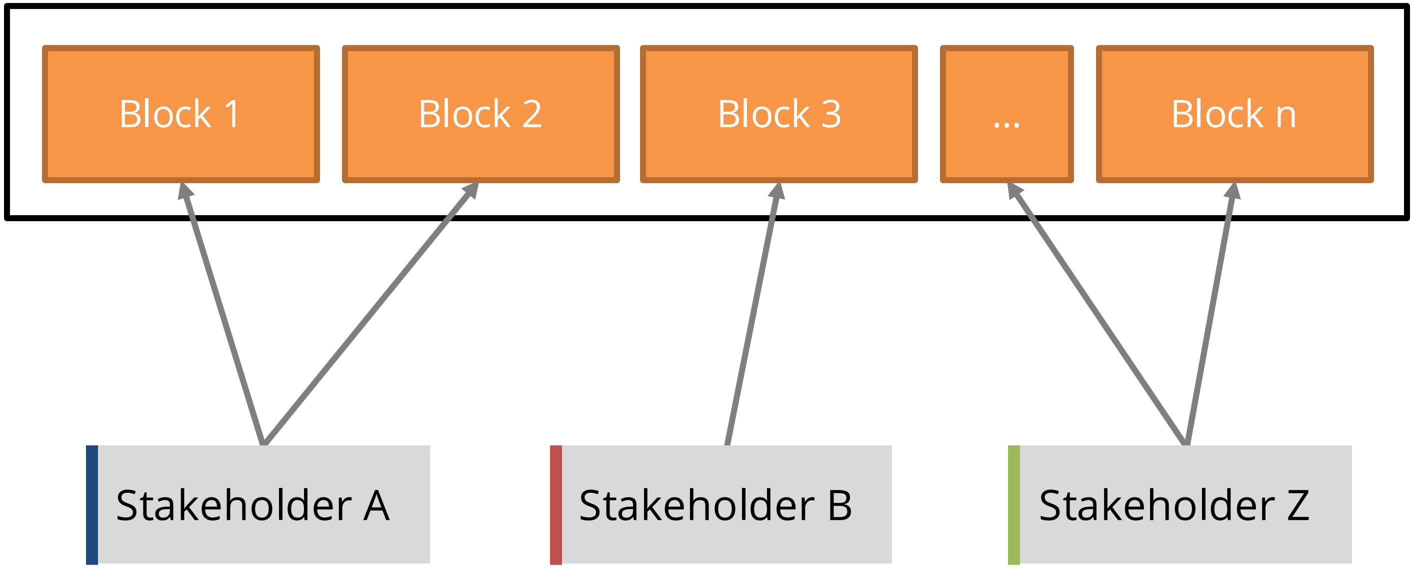
Object Memory Model (2)



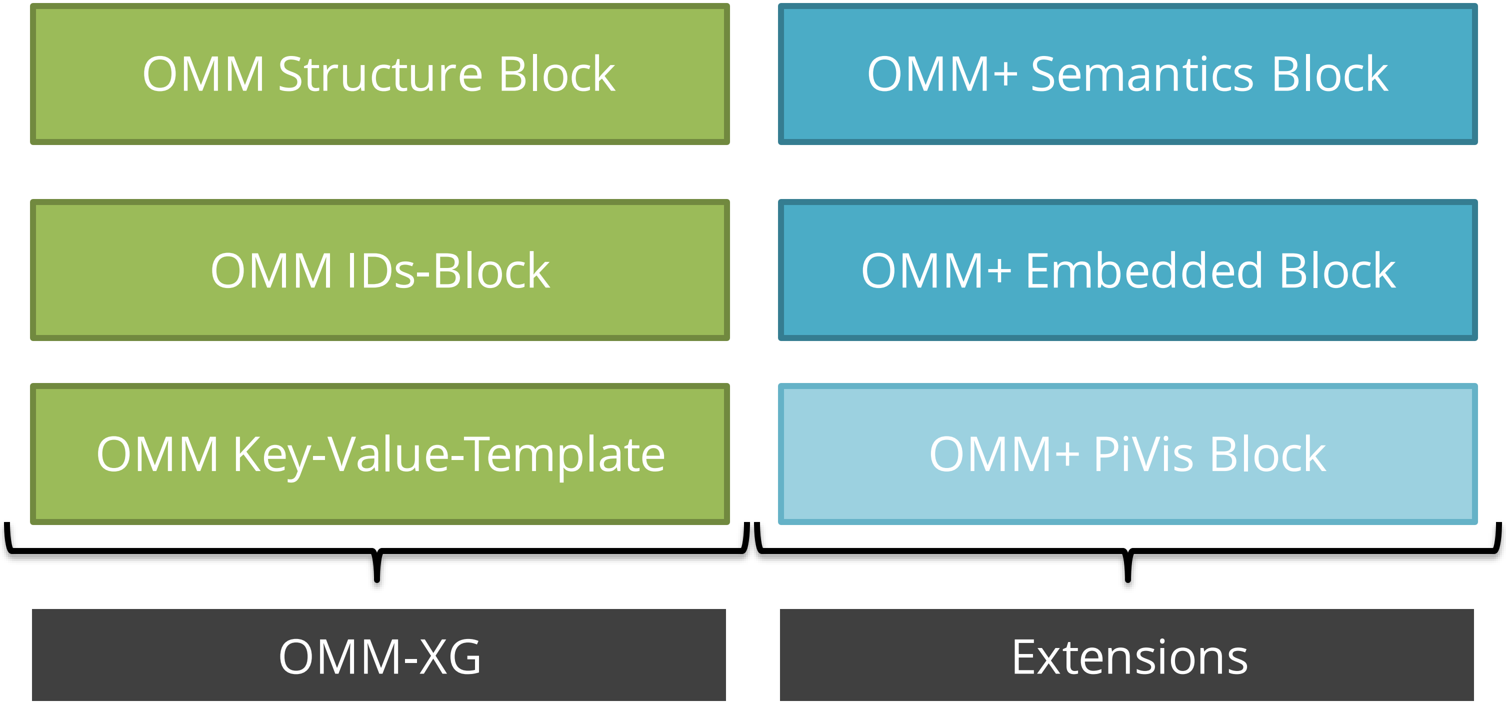
Object Memory Model (3)



Object Memory Model (4)



Defined Blocks



OMM Structure Block

OMM IDs-Block

OMM Key-Value-Template

OMM+ Semantics Block

OMM+ Embedded Block

OMM+ PiVis Block

OMM-XG

Extensions

Sample Memory (1)

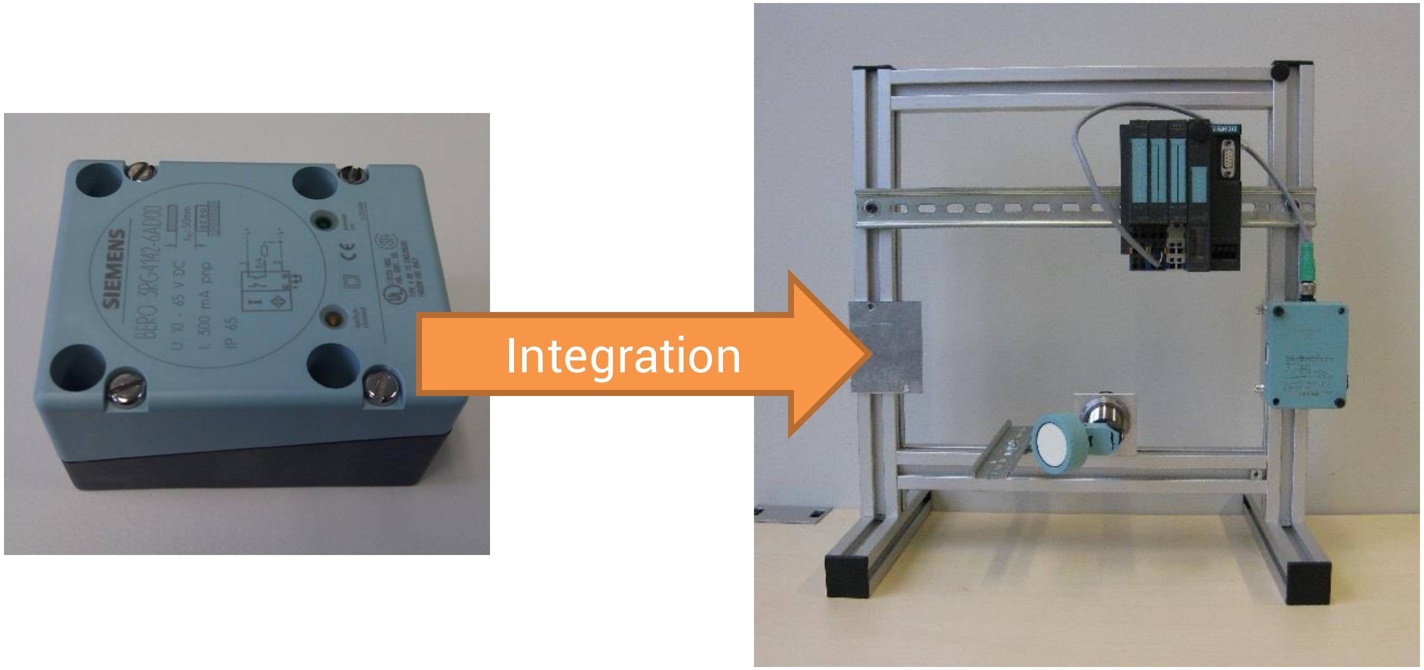
OMM-Block with sensor readings

ID	block_123
Format	application/xml Schema: http://mycomp.com/sensor.xsd
Creator	123456789@DUNS, 16.01.2013 16:15
Title	“SensorReading (temp/humi/shock)”
Type	http://purl.org/dc/dcmitype/Dataset
Subject	“sensorReadings” http://mycomp.com/o.owl#Temperature http://mycomp.com/o.owl#Humidity http://mycomp.com/o.owl#Shock

```
<?xml version="1.0">
<log>
 <reading>
  <value>-6</value>
  <topic>#Temp</topic>
  <unit>#Celsius<unit>
 </reading>
 <reading>
  <value>80</value>
  <topic>#Humid</topic>
  <unit>#Percent<unit>
 </reading>
 <reading>
  <value>1.6</value>
  <topic>#Shock</topic>
  <unit>#G<unit>
 </reading>
</log>
```

Sample Memory (2)

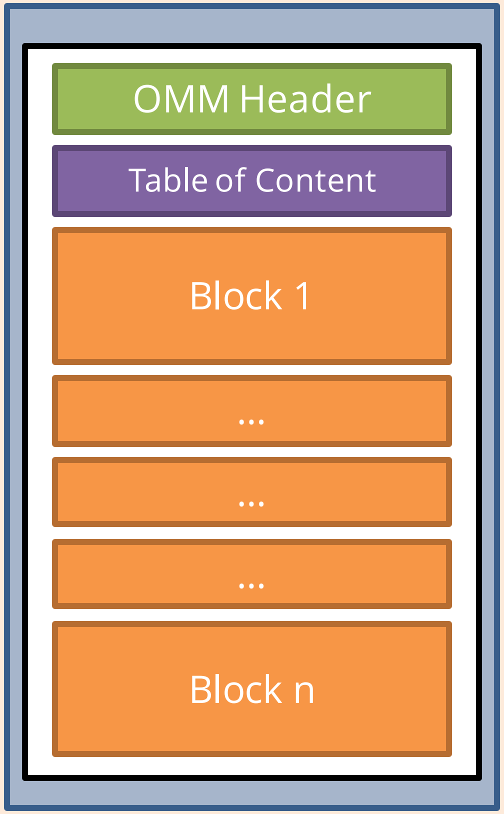
OMM-block with embedded memory



Sample Memory (3)

OMM-Block with embedded memory

ID	block_456
Namespace	urn:ommplus:blocks:embeddedBlock
Format	application/xml
Creator	123456789@DUNS, 16.01.2013 16:15
Title	“Memory: Sensor Optobero 124”
Type	http://purl.org/dc/dcmitype/PhysicalObject
Subject	„primaryID.opto124“ „myLoggingInfo“ http://ont.org/o.owl#Maintenance



OMM Header

Table of Content

Block 1

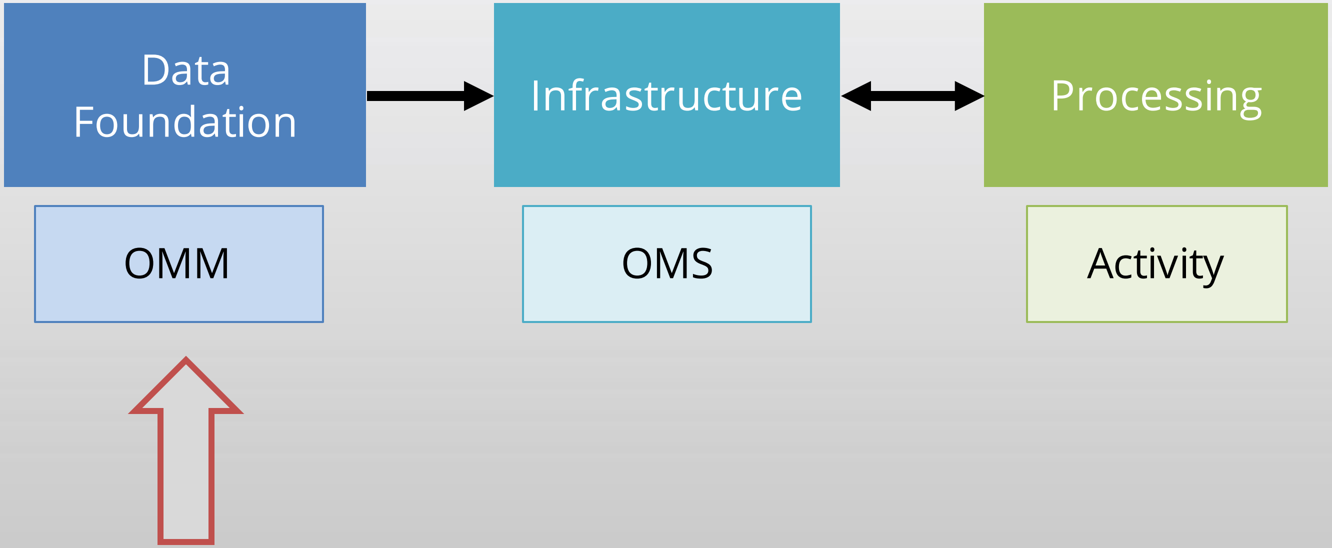
…

…

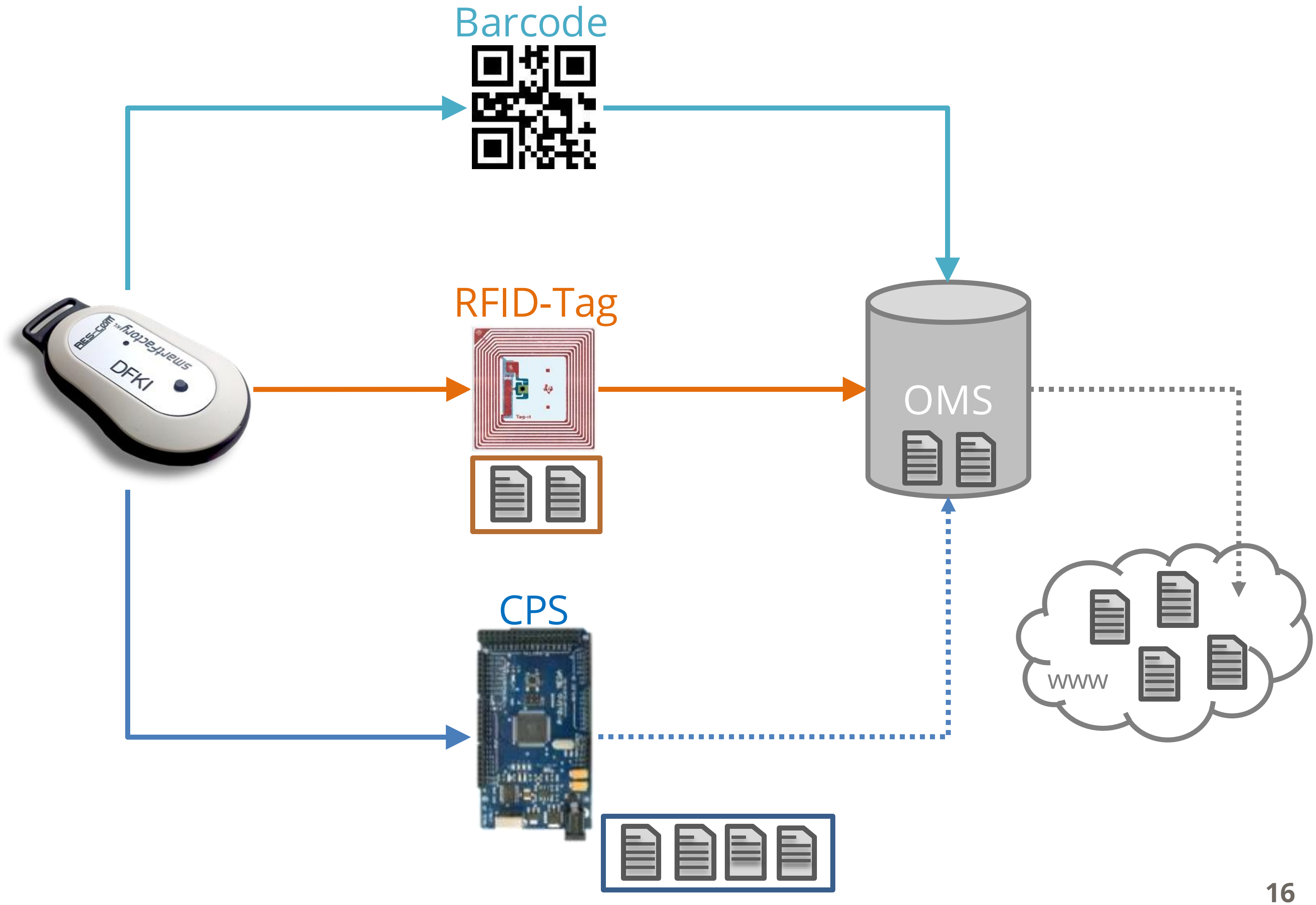
…

Block n

Infrastructure

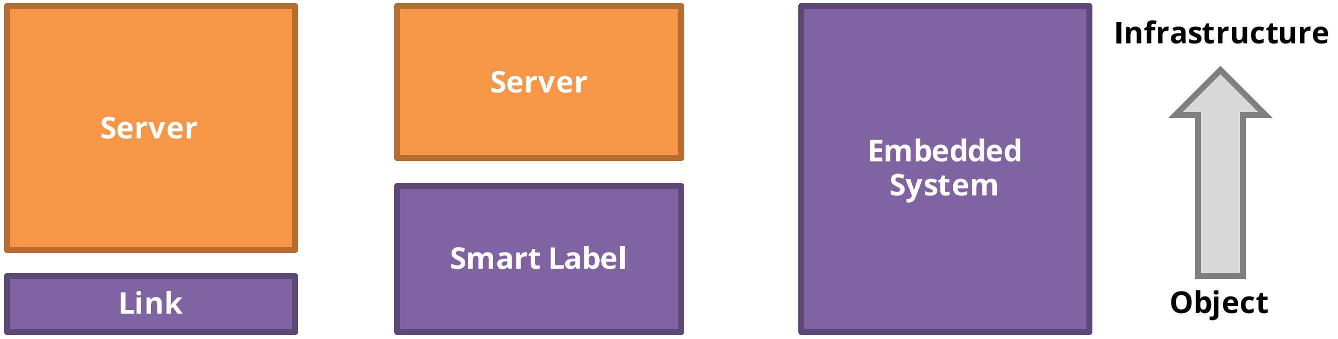


Memory Access



Object Memory Server

- Modular Architecture
  - Complete feature set with dedicated server
  - Partial functionality with embedded systems
  - Minimal core with simple systems



Server

Link

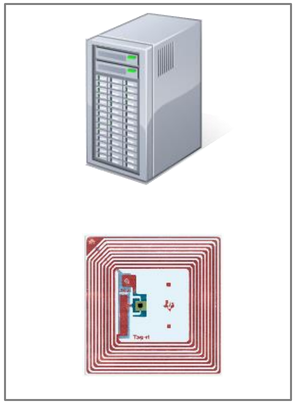
Server

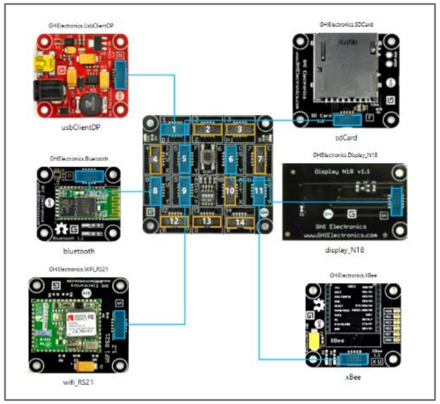
Smart Label

Embedded System

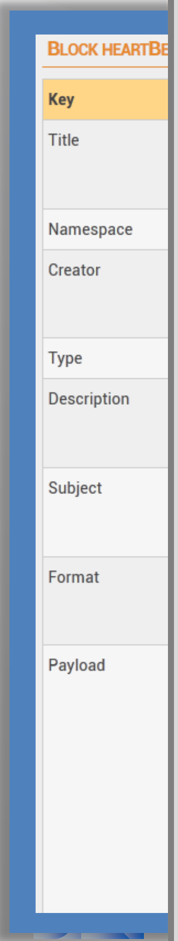
Infrastructure

Object





Object Memory Server (2)



BLOCK HEARTBE

Key

Title

Namespace

Creator

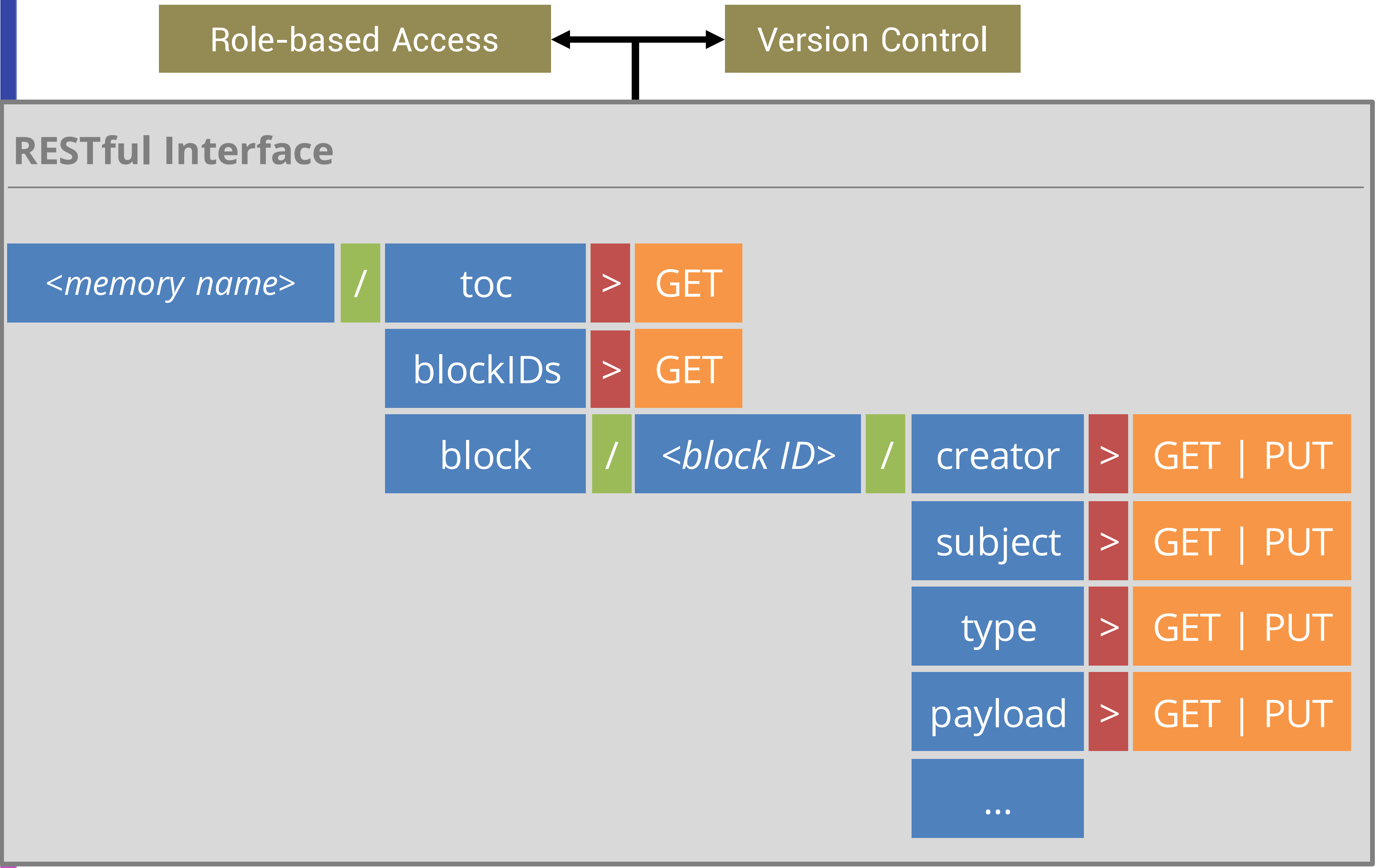
Type

Description

Subject

Format

Payload



Role-based Access

Version Control

RESTful Interface

<memory name> / toc > GET

blockIDs > GET

block / *<block ID>* / creator > GET | PUT

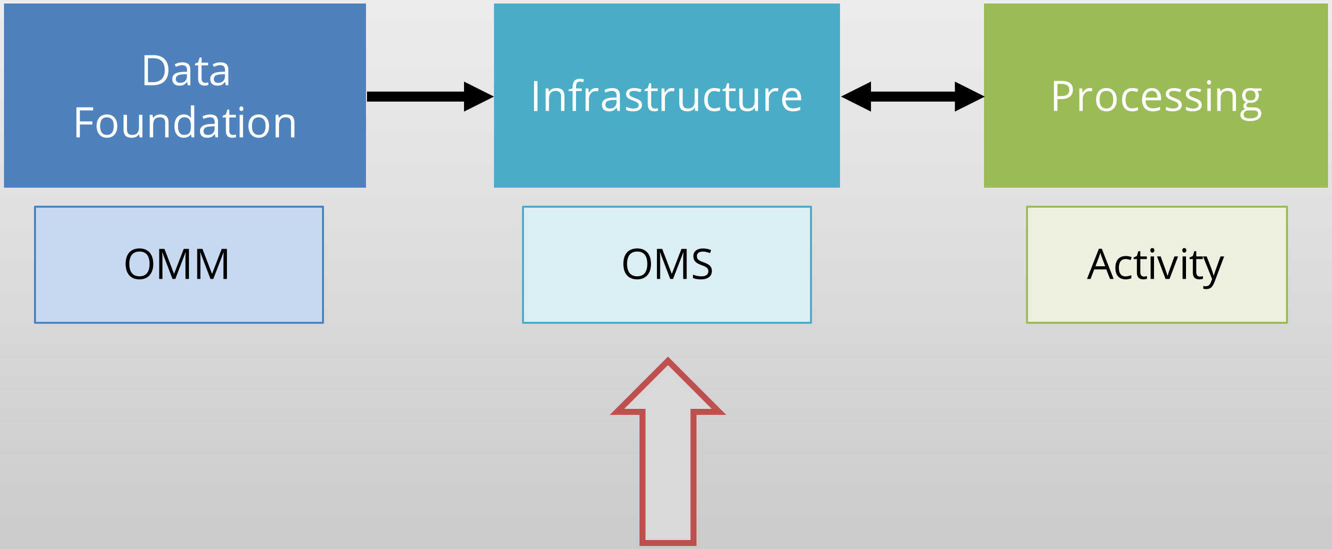
subject > GET | PUT

type > GET | PUT

payload > GET | PUT

...

Processing



Data Foundation

Infrastructure

Processing

OMM

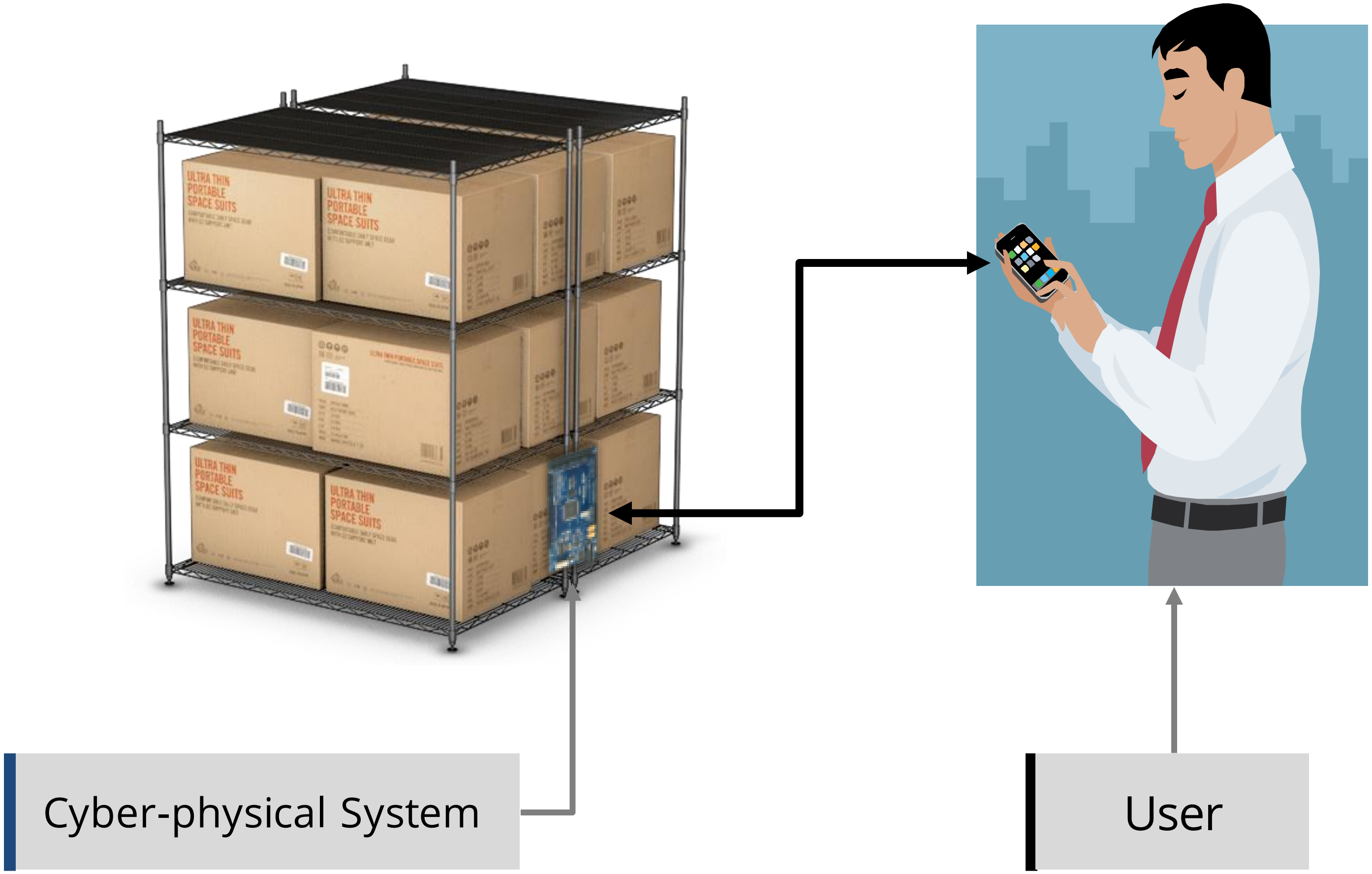
OMS

Activity

Activity

- Previous Architecture:
  - Passive data storage only
- Add processing logic to object memories
- Execute small, local operations (snippets) within object memory

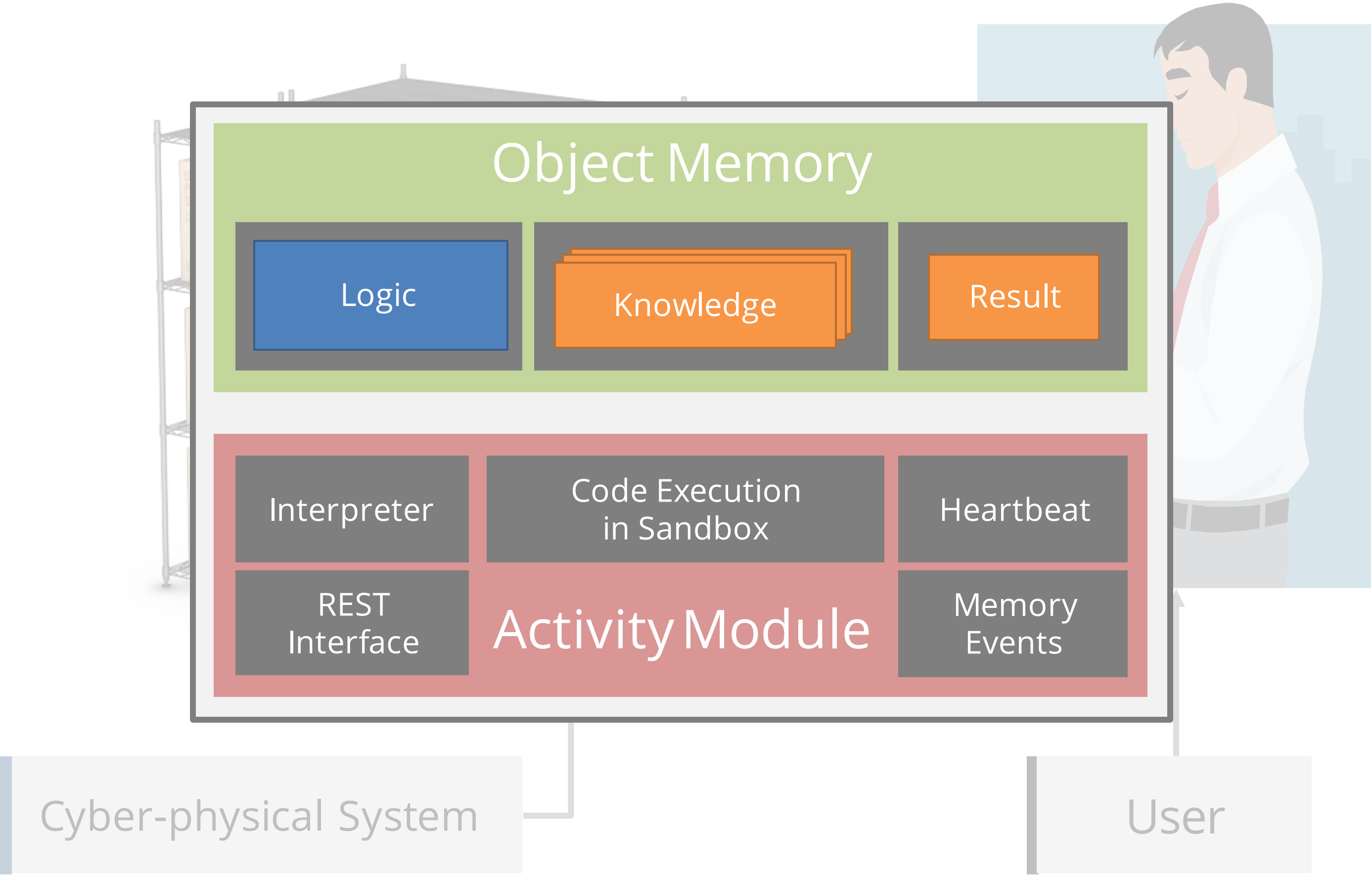
Activity (2)



Cyber-physical System

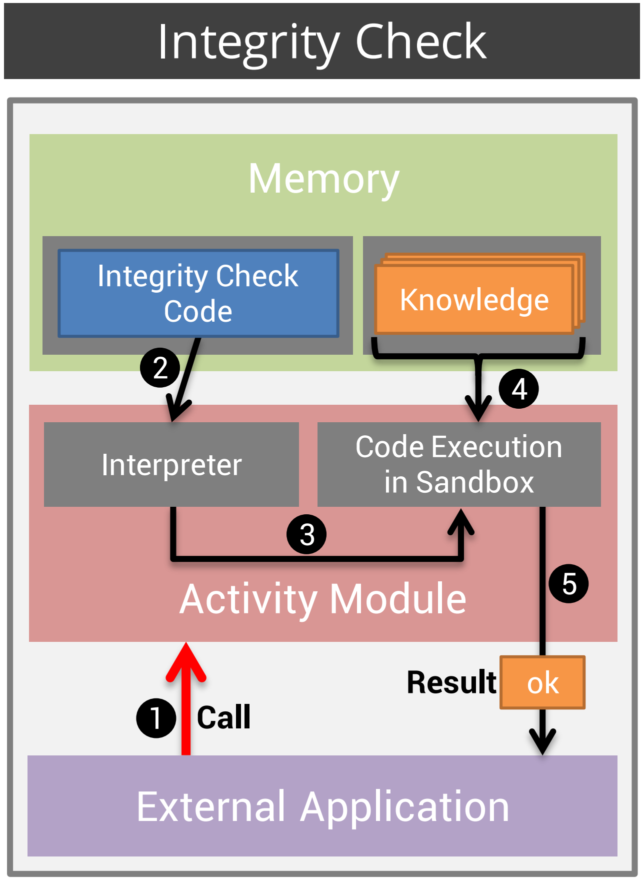
User

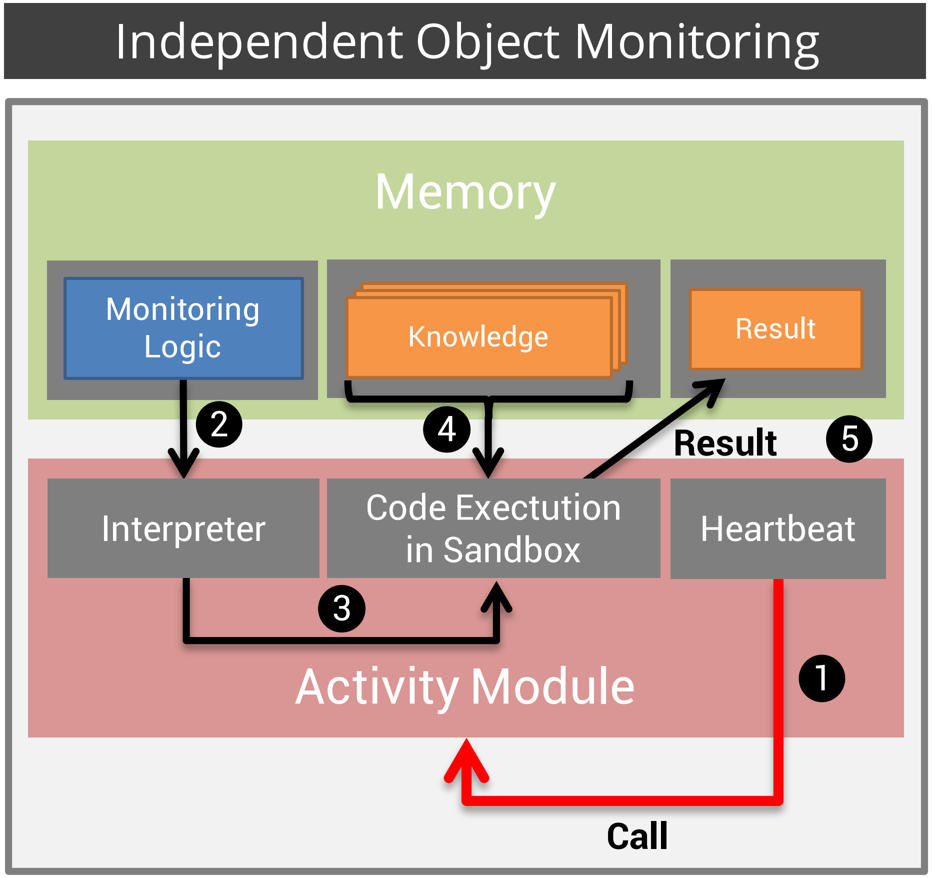
Activity (2)



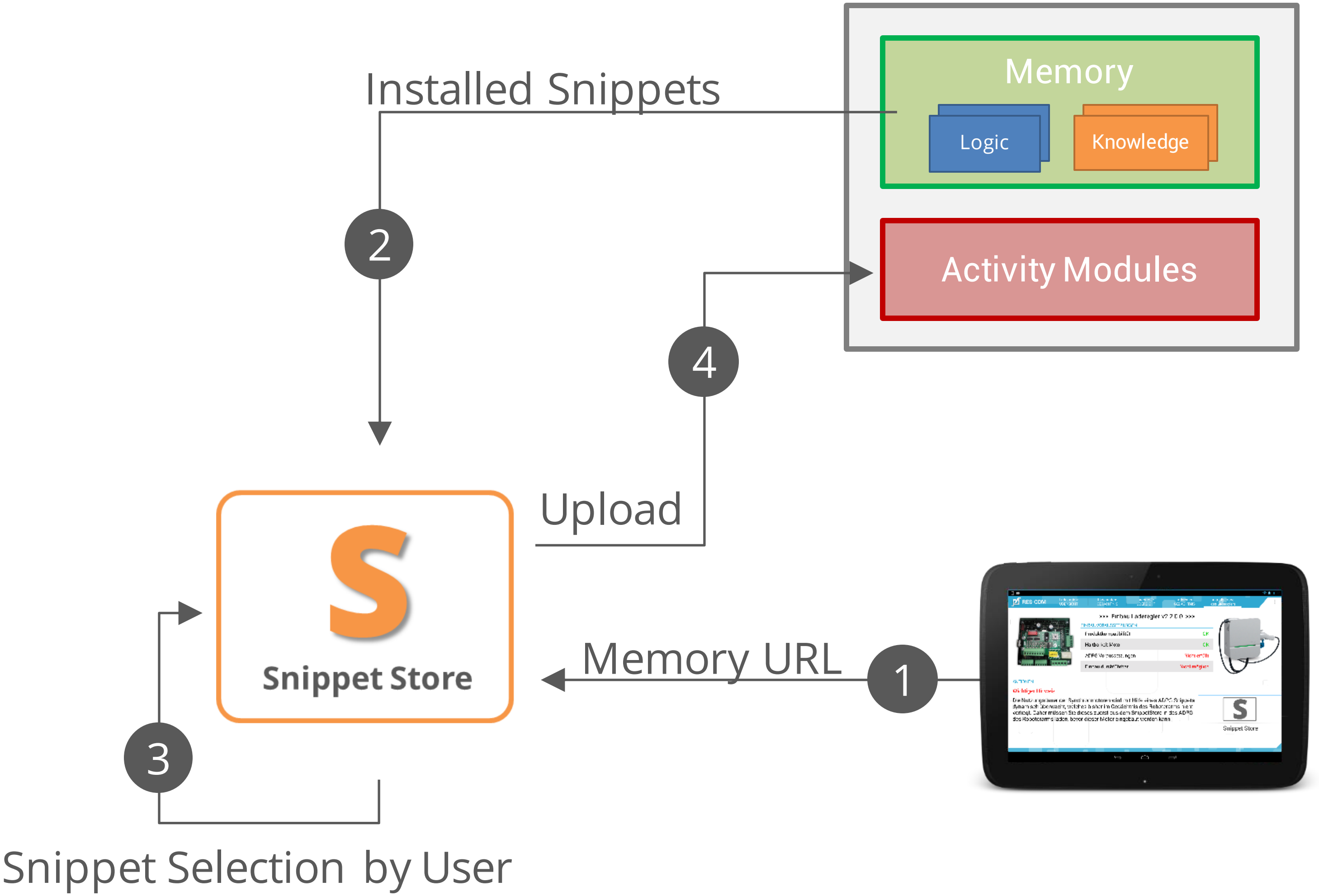
Object Memory
Logic
Knowledge
Result
Interpreter
Code Execution in Sandbox
Heartbeat
REST Interface
Activity Module
Memory Events
Cyber-physical System
User

Activity (3)


Integrity Check
Memory
Integrity Check Code
Knowledge
2
4
Interpreter
Code Execution in Sandbox
3
5
Activity Module
Result
ok
1
Call
External Application


Independent Object Monitoring
Memory
Monitoring Logic
Knowledge
Result
2
4
Result
5
Interpreter
Code Exectution in Sandbox
Heartbeat
3
Activity Module
1
Call

Activity (4)



Installed Snippets

Memory

Logic

Knowledge

Activity Modules

2

4

Upload

Snippet Store

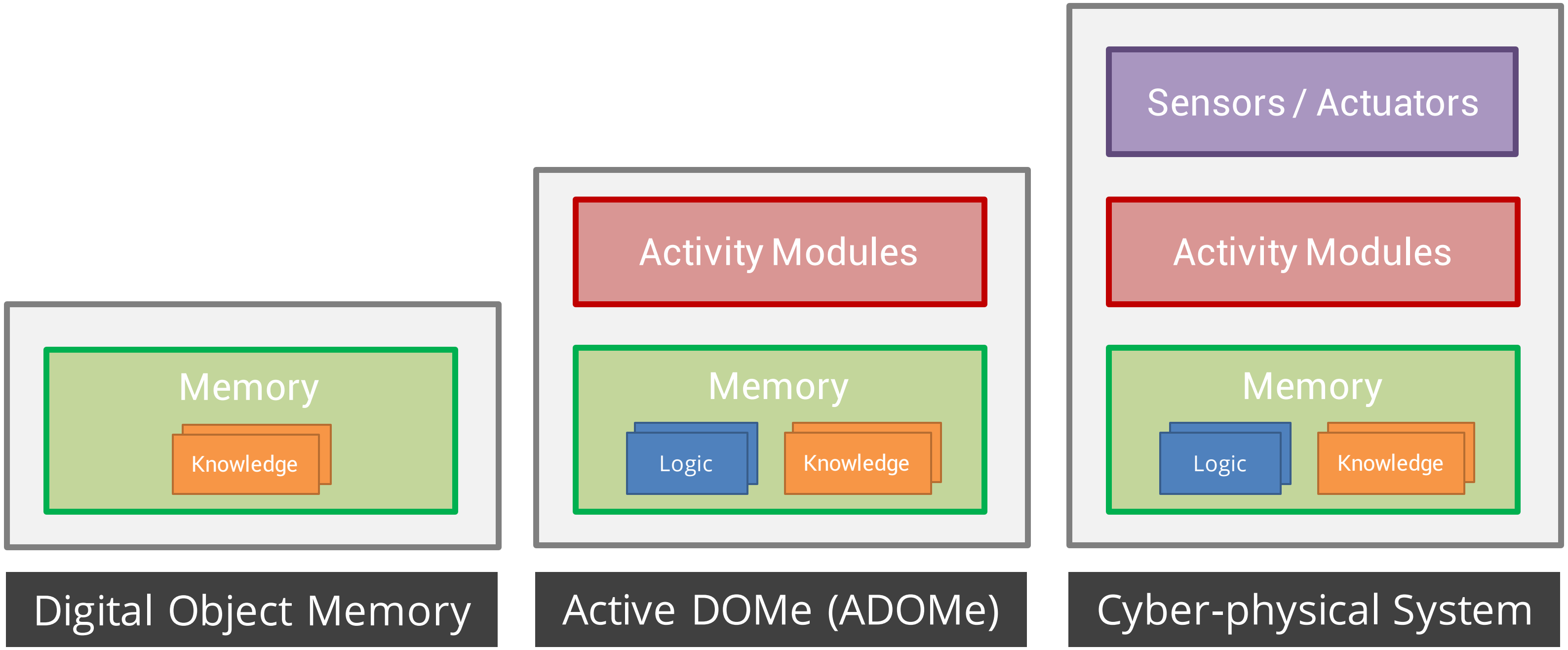
Memory URL

1

3

Snippet Selection by User

ADOMe vs. CPS



Memory

Knowledge

Digital Object Memory

Activity Modules

Memory

Logic

Knowledge

Active DOMe (ADOMe)

Sensors / Actuators

Activity Modules

Memory

Logic

Knowledge

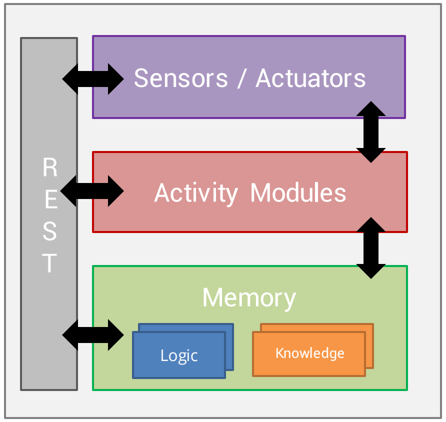
Cyber-physical System

OMM on Embedded Platf.

Raspberry Pi

Object Memory Server (OMS)

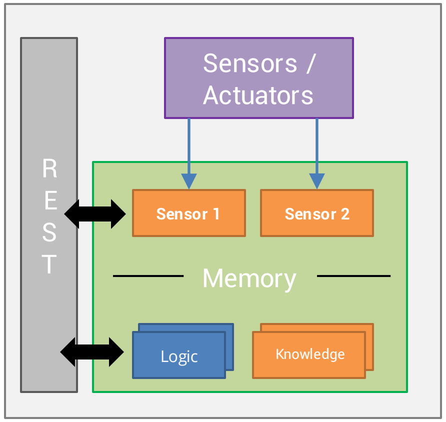
- Java-Version w/o VCS und ACL
- RESTful interface
- Activity Module and Snippets
- OMM XML representation



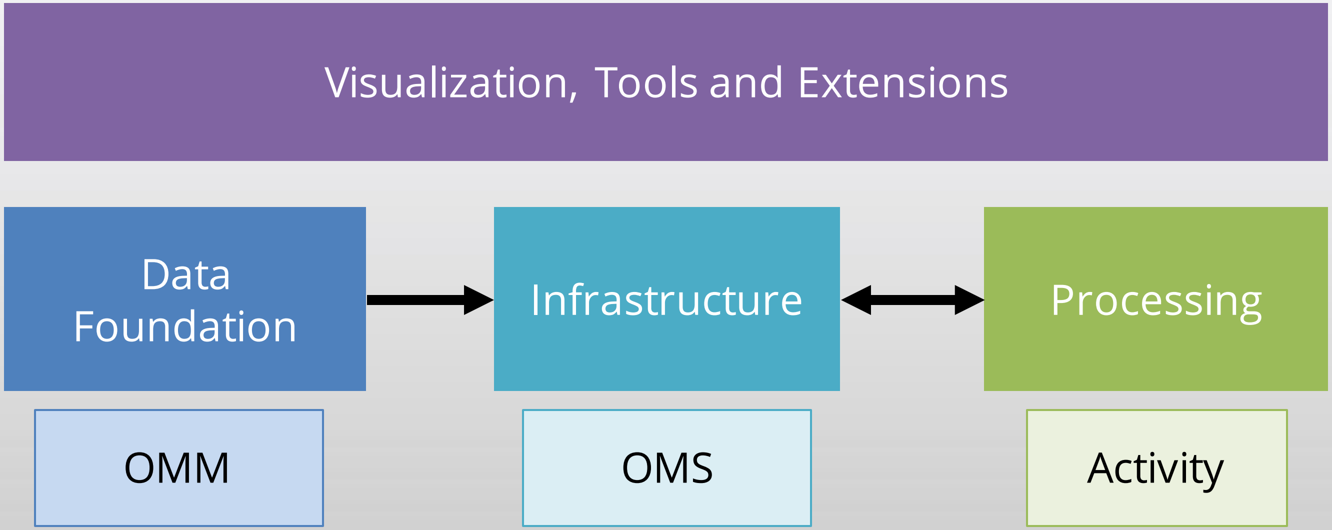
Gadgeteer

Object Memory Server (OMS)

- C# compact implementation
- RESTful interface
- Transparent sensor integration
- OMM Binary representation



Visualization, Tools and Extensions



Visualization, Tools and Extensions

Data Foundation

Infrastructure

Processing

OMM

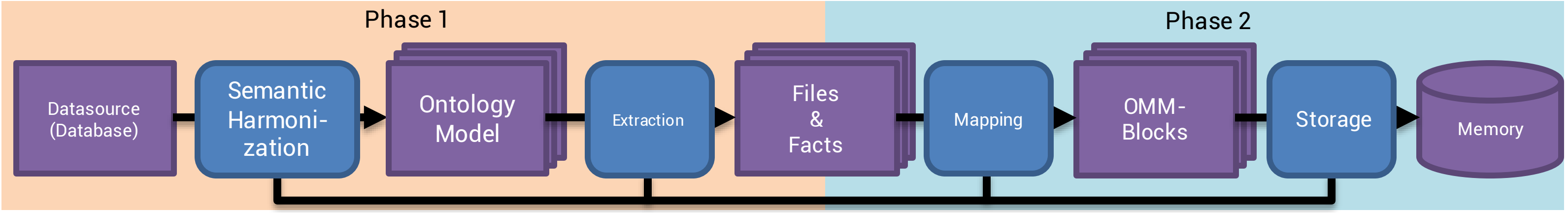
OMS

Activity

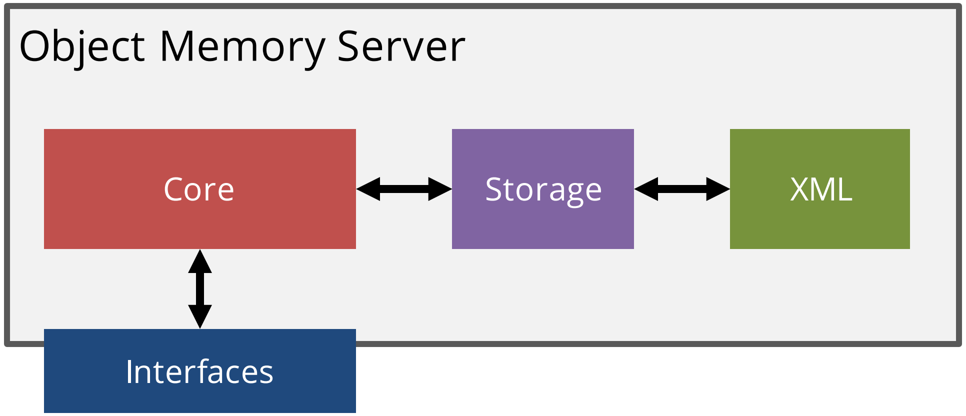
Database Migration

- *Today*: Data stored in databases
- *The day after tomorrow*: Digital object memories

- Migration:
  - Semantic Harmonization via Ontology Model
  - Facts and File-extraction
  - Automatic Memory generation



Data Representation



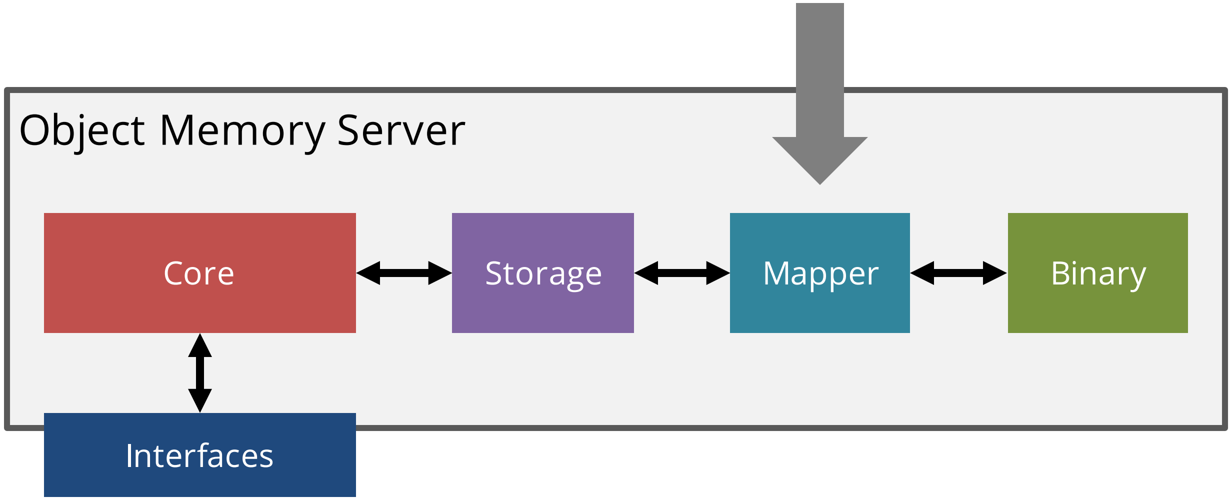
Pros:

- Standardized / Well defined
- Widely used
- Human readable

Cons:

- Verbosity and complexity
- Slow processing
- Bad cost-value ratio

Data Representation (2)

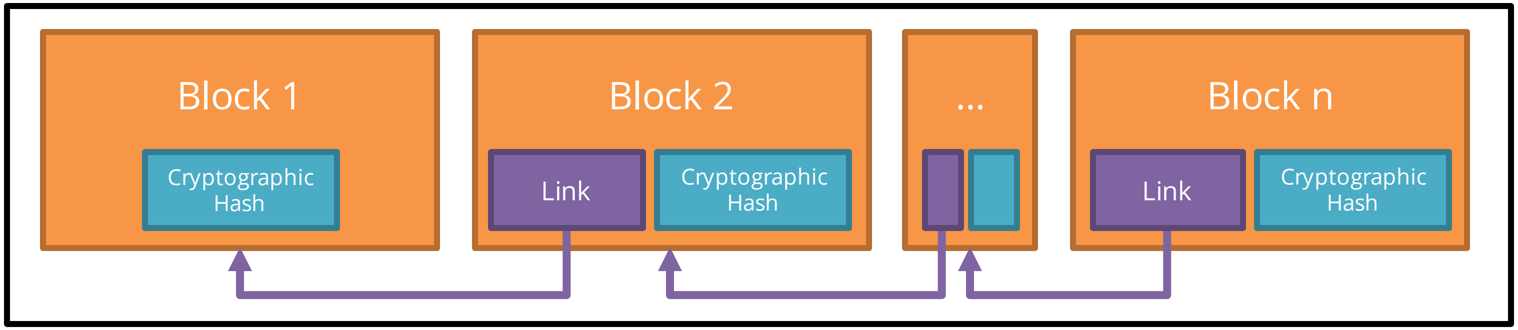


- Dynamic mapper for binary representation
  - Lossless conversion for low storage space consumption
  - Lossy conversion for minimum space consumption
    - Partial restore of lost data possible via mapping table

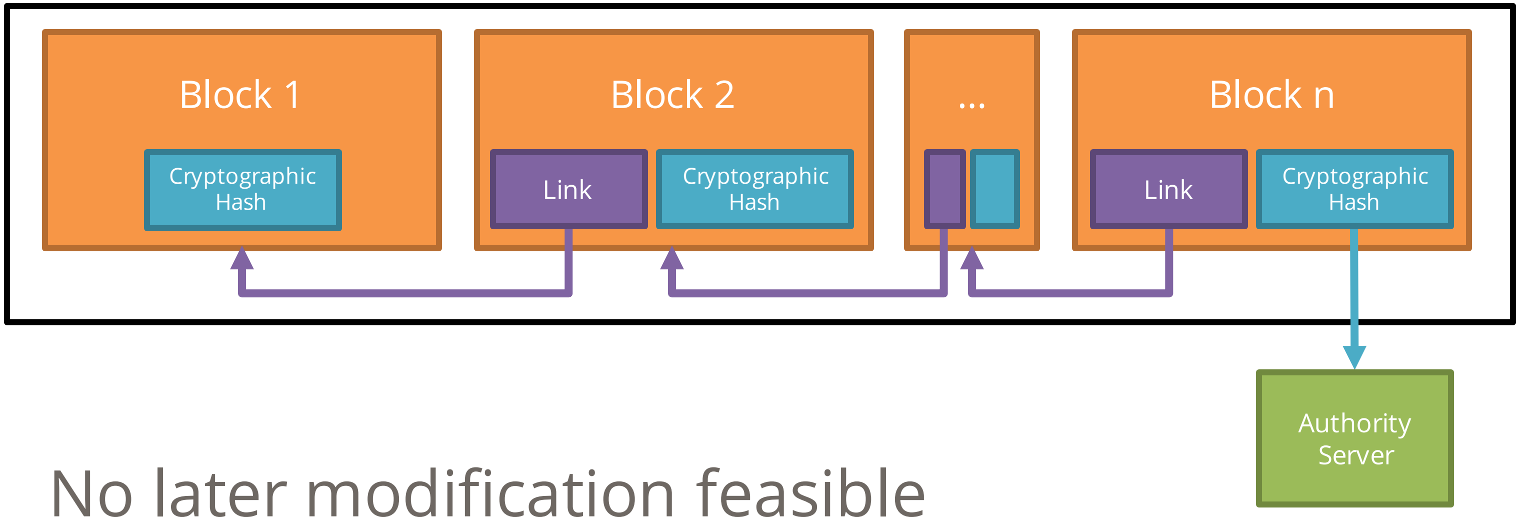
Secure Provenance

- Warranty deeds and legal grounds
  - Secure and verifiable data necessary

➢ Memory with secure provenance



Secure Provenance



- No later modification feasible
  - No alteration of existing blocks
  - No insertion of new blocks
  - No deletion of existing blocks (with Authority Server)

Results

- Definition of a ***meta model*** to structure heterogeneous object memory data and positioning in W3C OMM-XG
- ***Reference implementation*** of a OMM-based memory infrastructure for open-loop scenarios
- ***Hardware abstraction*** allows usage independent of object properties and capabilities
- Plug-In-based, dynamic ***visualization*** for heterogeneous data
- Additional ***toolset*** to create new applications and migrate existing systems

Discussion

- DOMe-related Data by Trumpf?
- Where could digital object memories be used?
  - Manufactured products?
  - Machinery?
- New potentials and business models?

[redacted]

Discussion

- Which types of memories are suitable?
  - Embedded/on-object
  - Server-based/off-object
- What data could enrich the product life-cycle?
- Where could autonomous activity be used?
- How could a Snippet Store look like?

[redacted]

Thank you very much for your attention.

http://www.dfki.de/omm-tools/

Benefits

- Open data storage
  - Communicate with open-loop partners
  - Standardized formats and proprietary extensions
- Self-explaining objects
  - Object-related information attached to physical object
  - Replace paper documentation
  - Define monitoring conditions
  - Maintenance data
- Maintenance data attached to product
  - Retrospective
  - Spare parts

Benefits

- Object memory data storage location
  - On-object for high value products
  - Off-object for low cost products
  - Arbitrary hybrid solutions possible
  - Communication via standardized web techniques

- Versatile applications for object memories
  - Raw materials
  - Machinery
  - Goods
  - Hierarchical solutions

Benefits

- Object memory as storage space for snippets
- Secure *sandbox* execution
- Update Snippets via AppStore
  - Subsequent parameter adjustment
- Object-related logic (lot size one)
  - Integrity Check
- Autonomous/independent monitoring
  - Independent Object Monitoring