SitRS – A Situation Recognition Service based on Modeling and Executing Situation Templates

Pascal Hirmer[1], Matthias Wieland[1], Holger Schwarz[1], Bernhard Mitschang[1],
Uwe Breitenbücher[2], Frank Leymann[2]
[1]Institute of Parallel and Distributed Systems,
[2]Institute of Architecture of Application Systems
University of Stuttgart, Stuttgart, Germany

University of Stuttgart
Germany

DFG Deutsche Forschungsgemeinschaft

IAAS IPVS

Motivation – Integrate Internet of Things..

- Increasing interconnection of IT systems and physical objects
  - Smart watches, smart phones, smart production systems, smart home technology, …
- Huge amounts of sensors generating sensor data
- ➢ Uniform sensor data integration and sensor data processing needed
  - ➢ Enable the Internet of Things

© Pascal Hirmer

Motivation – Situation Recognition

- Situation: An occurring event in a SMART* environment (SMART factory, SMART home…)
- Examples: production machine damaged, server load critical, room occupied, room temperature increased…

- Situations recognized based on sensor data
  - But: huge amount of low-level, raw sensor data
    - → difficult to process
  - ➢ We need a means to extract high-level situations based on raw sensor data

© Pascal Hirmer

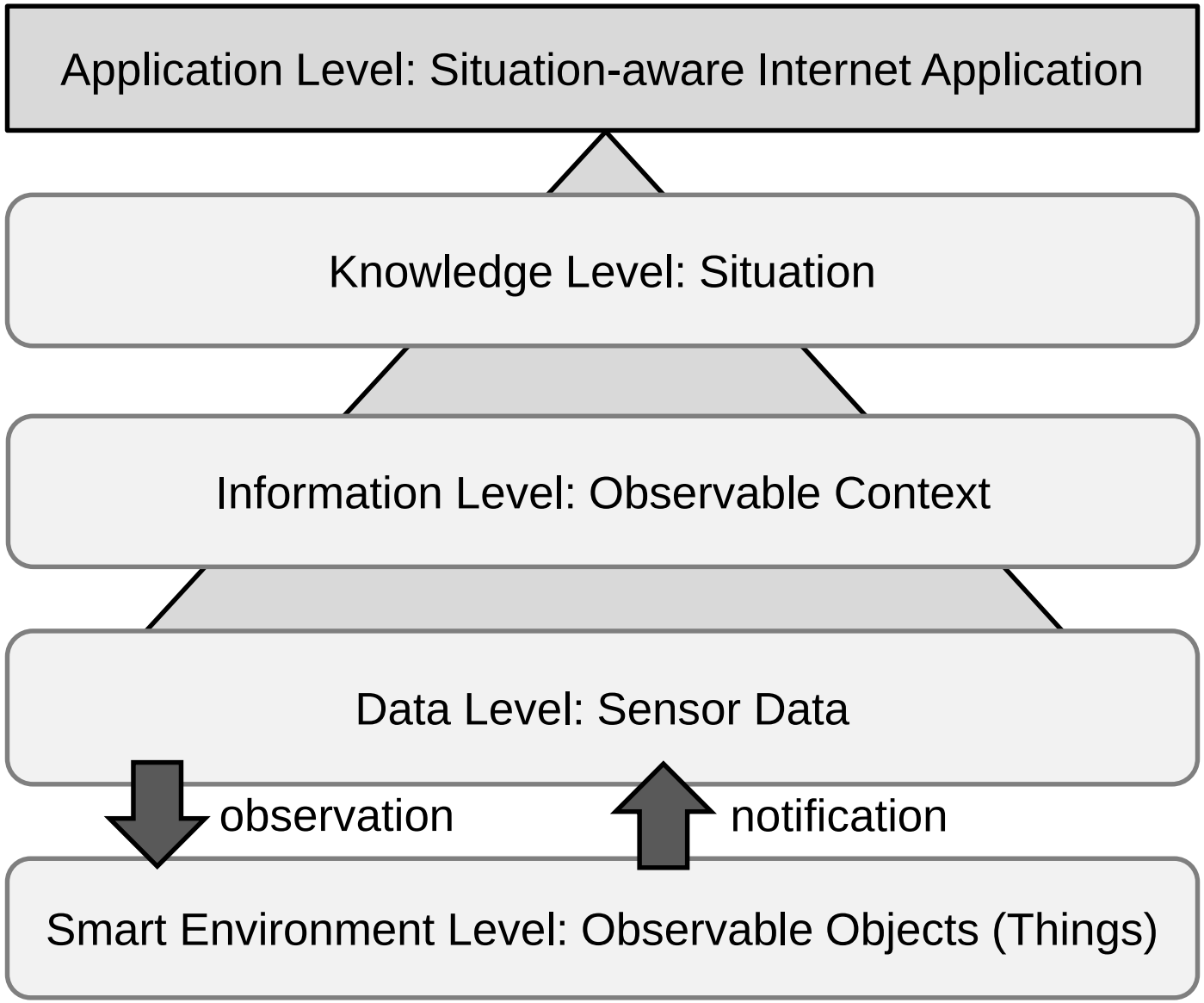
Paper Contribution and Agenda

- Concept and architecture for a situation recognition service – SitRS
  - Cloud-based service
  - Automated sensor binding
  - Process raw sensor data to recognize occurring situations

- Agenda
  (1) Situation modeling with Situation Templates
  (2) Situation recognition method
  (3) Situation model
  (4) SitRS architecture
  (5) Summary and outlook

© Pascal Hirmer

Situation Detection Pyramid – Levels



1. Register situation recognition (e.g., for server)
2. Adapt to recognized situations

Situation-Model,
e.g., state of server
“ready” or “critical”

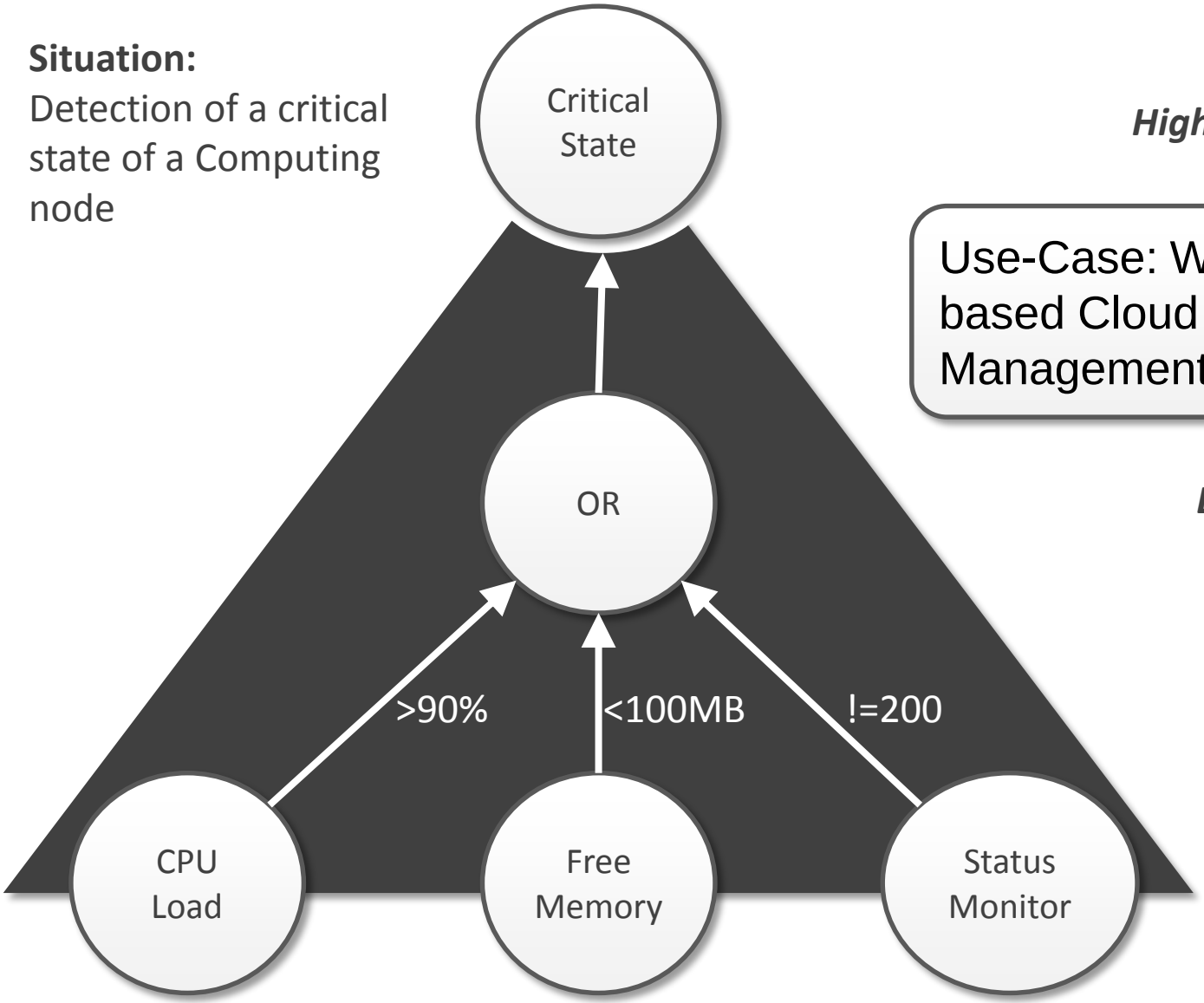
Context-Model,
e.g., Object.RamState

Low-level data types,
e.g., MB of RAM

© Pascal Hirmer

Situation Template modeled as Situation Aggregation Tree

Situation:
Detection of a critical state of a Computing node



High-Level Situation

Use-Case: Workflow-based Cloud Infrastructure Management

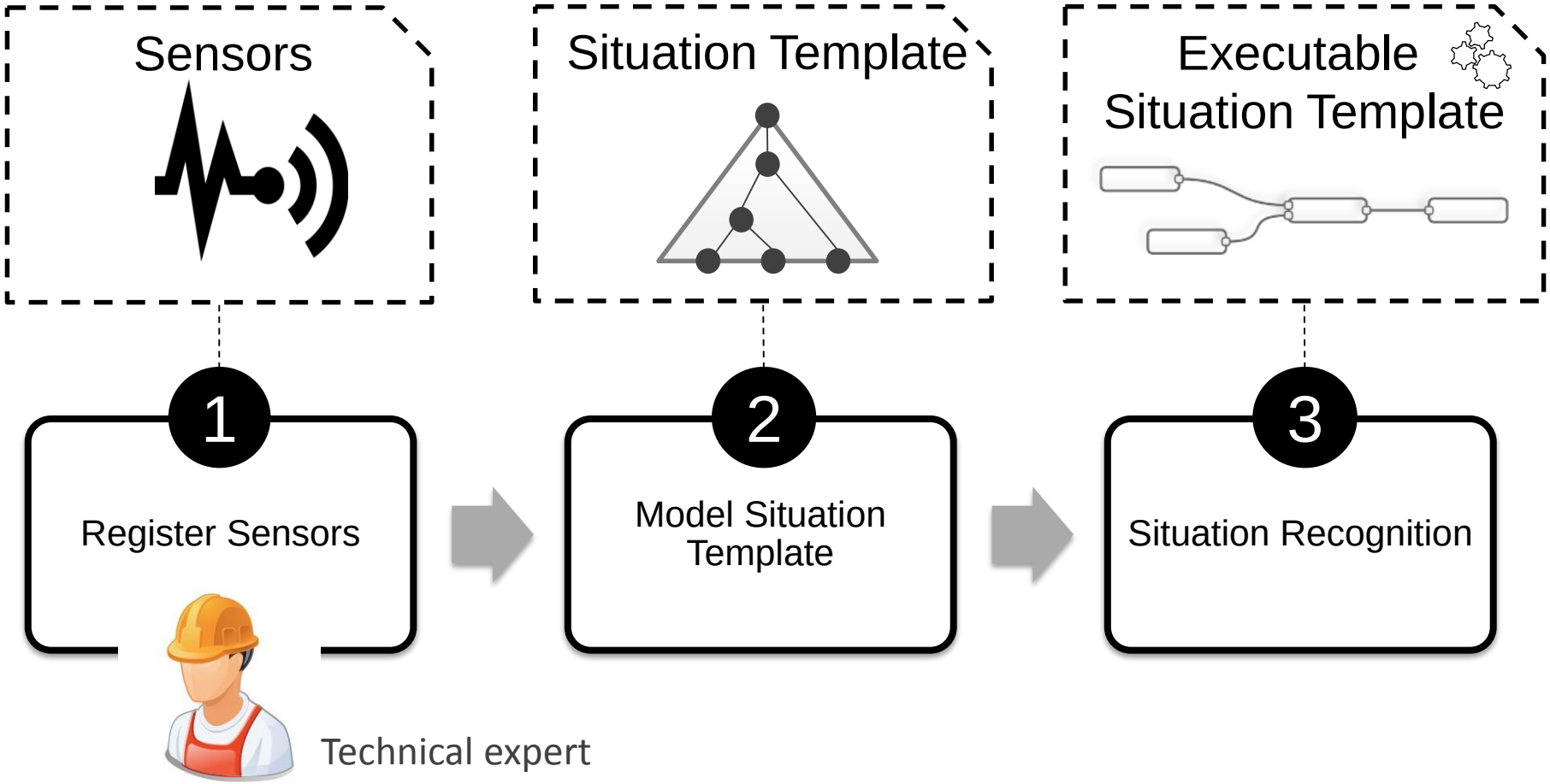
Logical Operations

Conditions

Context Nodes

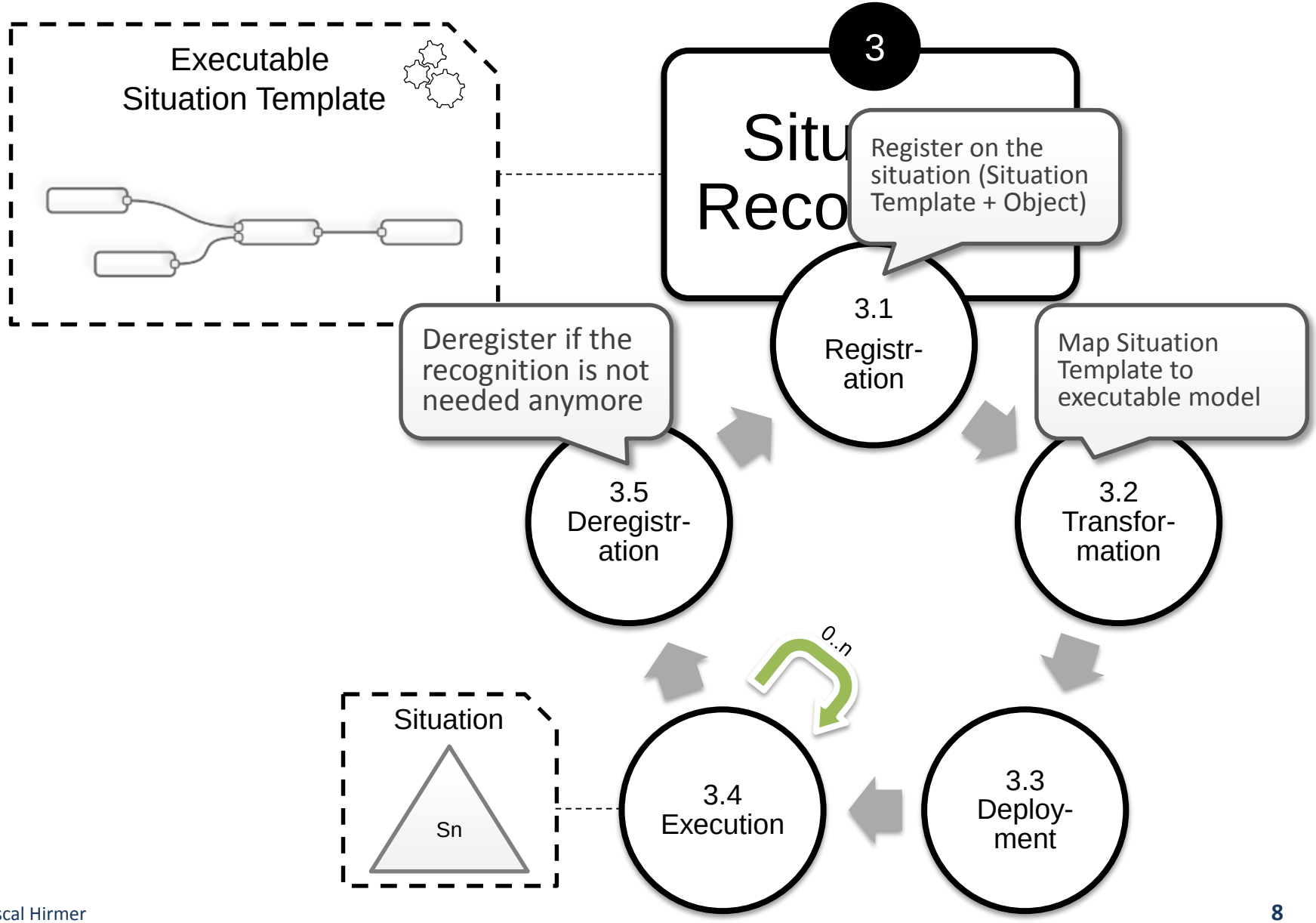
© Pascal Hirmer

SitRS – Situation Recognition Method



© Pascal Hirmer

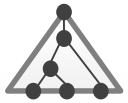
Step 3 – Detailed View



© Pascal Hirmer

Result of Situation Detection – Situation Object

Machine failure

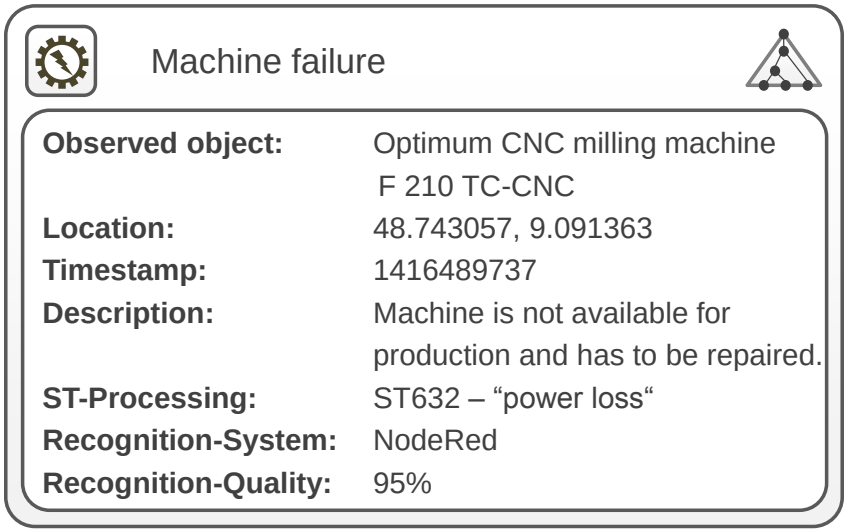


Observed object:	Optimum CNC milling machine F 210 TC-CNC
Location:	48.743057, 9.091363
Timestamp:	1416489737
Description:	Machine is not available for production and has to be repaired.
ST-Processing:	ST632 – “power loss“
Recognition-System:	NodeRed
Recognition-Quality:	95%

© Pascal Hirmer

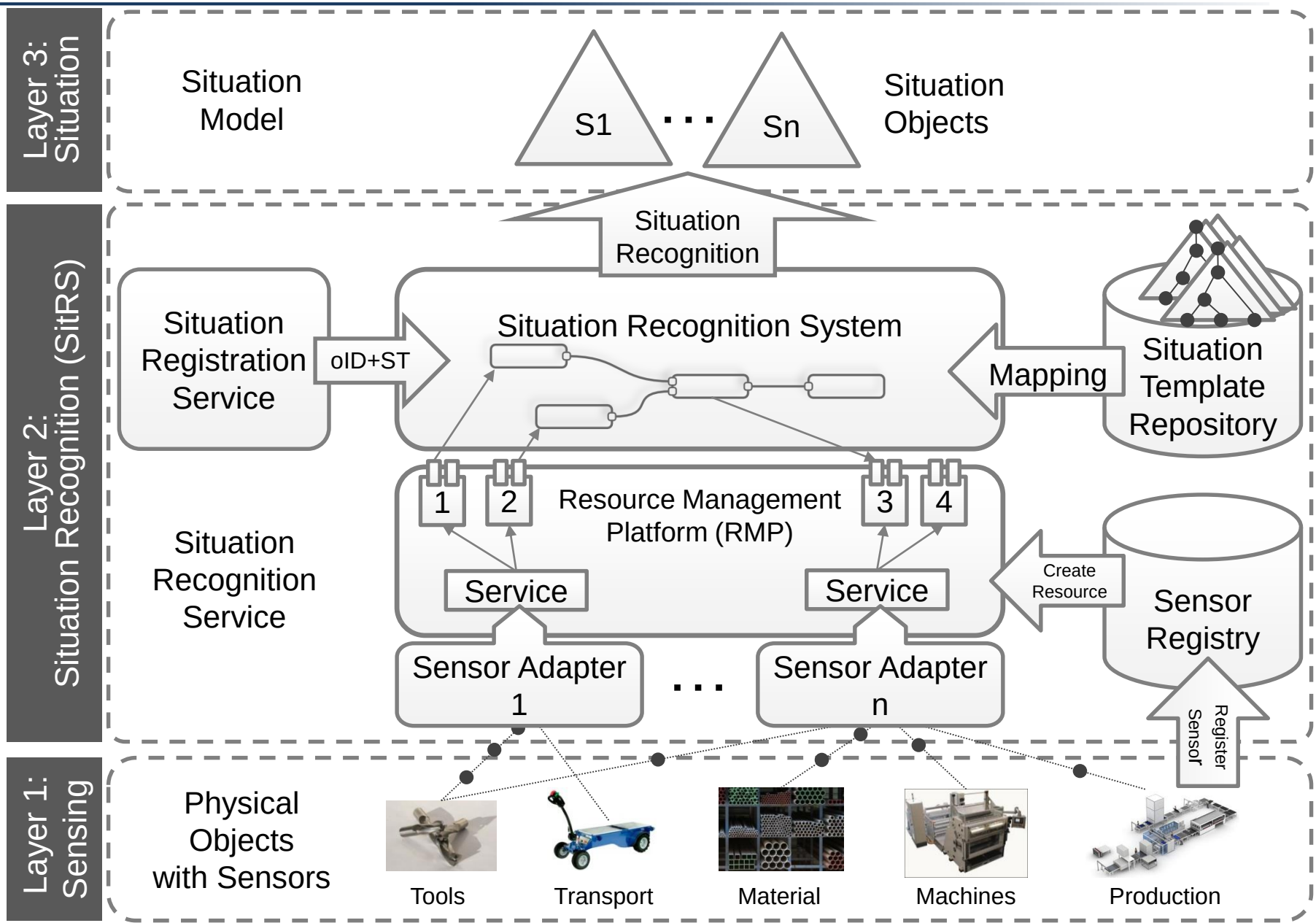
Structure of a Situation Object defined by Situation Model

- **Name**: Name of the situation.
- **Icon**: Figure of the situation as a sketch.
- **Observed Object**: The ID of the object in the context model that is observed and hence described by the situation
- **Location**: Longitude and Latitude in GPS Coordinates.
- **Observation-Timestamp**: Timestamp in UNIX time when the situation was detected.
- **Description**: Textual description of the situation.
- **ST-Processing**: Link to the situation template that was used
- **Recognition-System**: System that was used to implement and execute the situation recognition.
- **Quality**: A measurement of the overall quality of the conducted situation recognition process



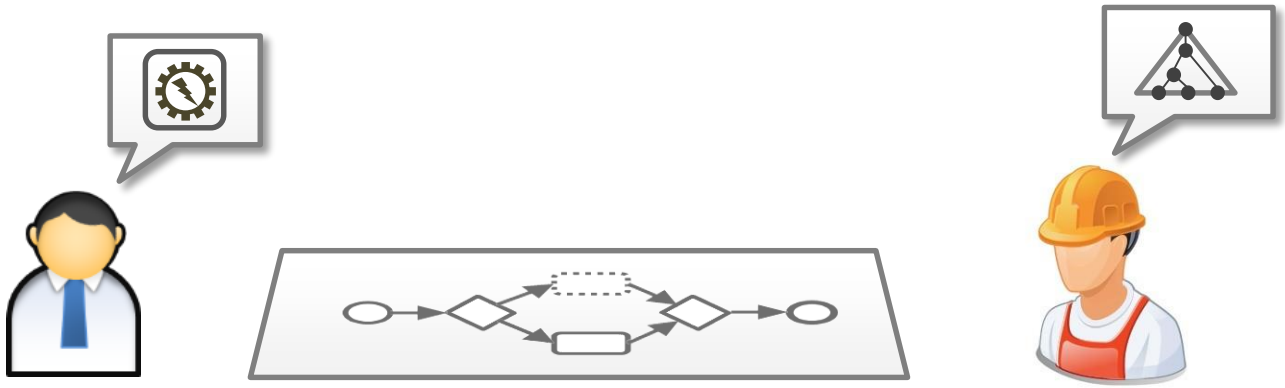
© Pascal Hirmer

SitRS – Architecture



Use case – Situational Control Flow Modeling (I)

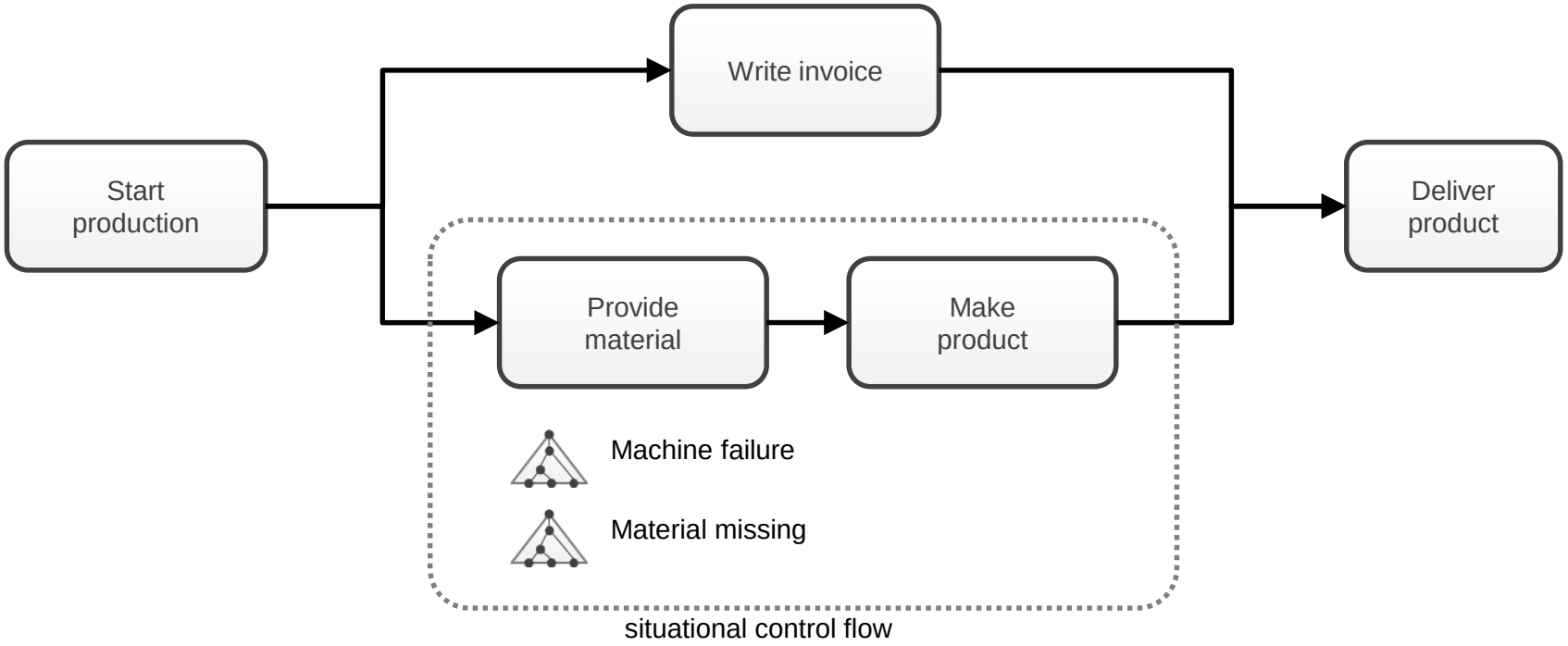
- Analysts and Planners can model their process as standard workflow
- Domain experts can model their knowledge as Situation Template for situation recognition
  - „Whenever event 5 happens always situation X occurs “
- Together situational exceptions can be modeled
  - Annotate situation to standard workflow
  - Handling different situations with situational workflows



© Pascal Hirmer

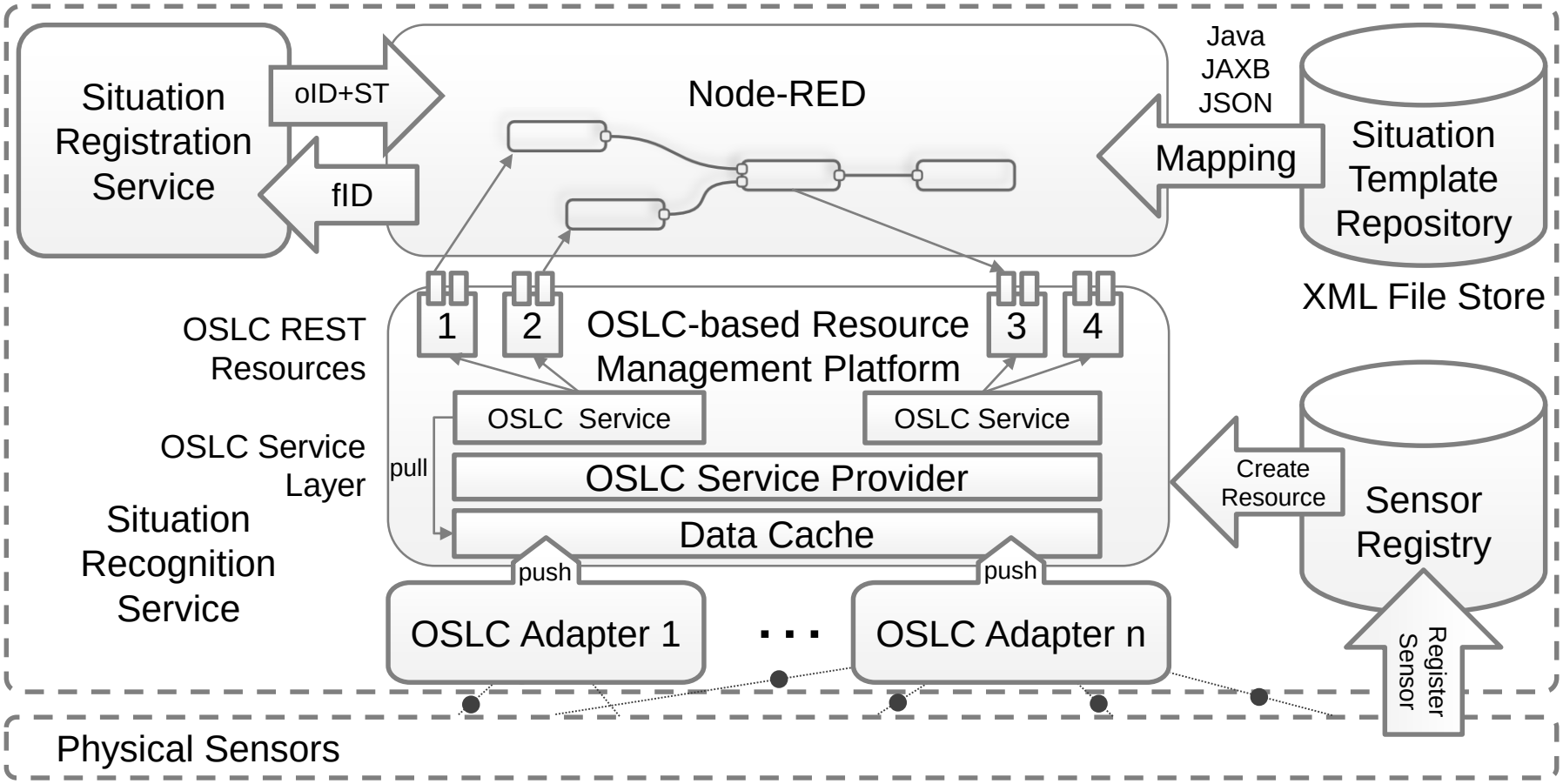
Use case – Situational Control Flow Modeling (II)

- Result: Situation-aware Workflow



© Pascal Hirmer

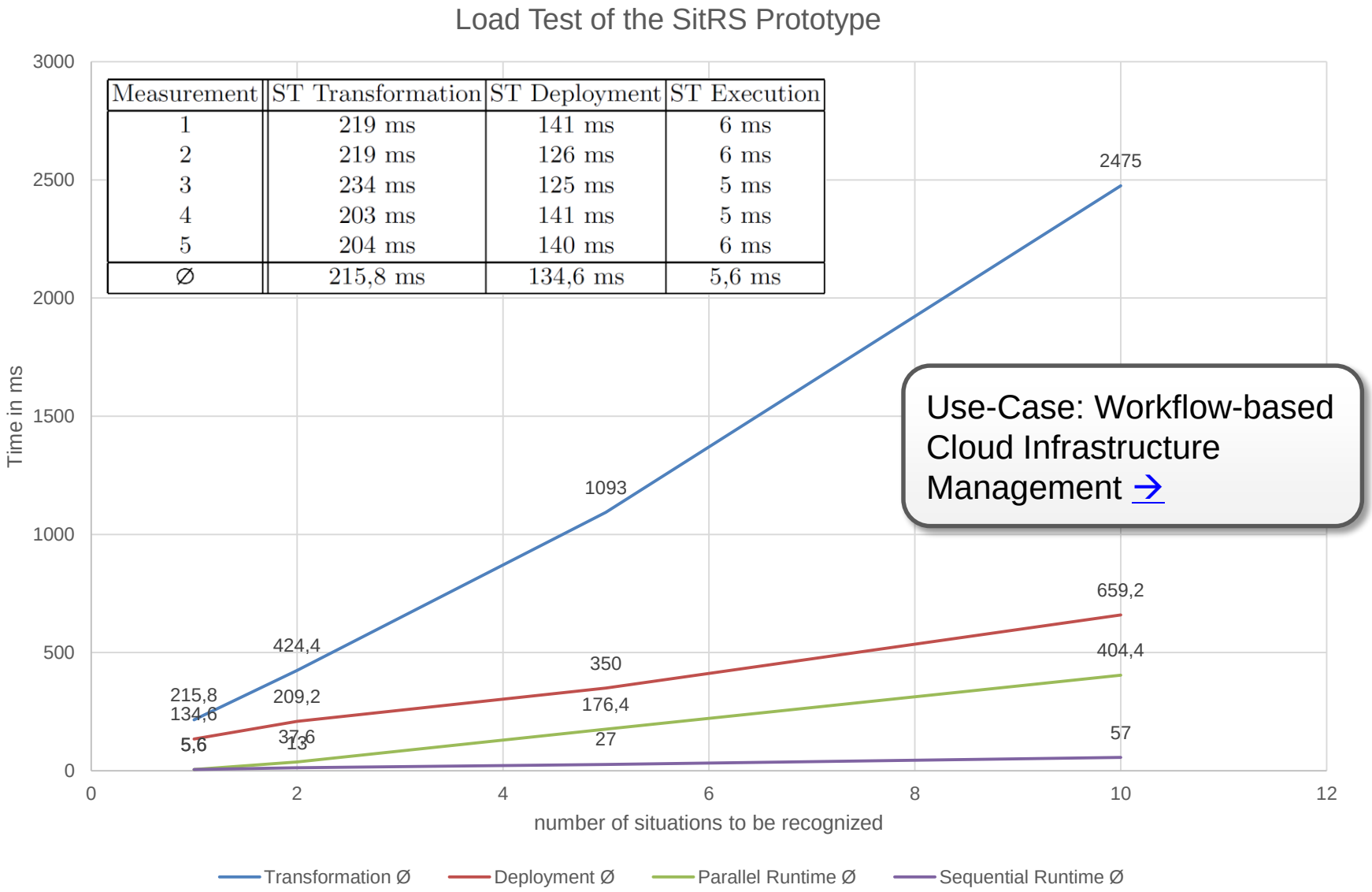
Prototypical Implementation



© Pascal Hirmer

Summary – Prototype & Measurements

Measurement	ST Transformation	ST Deployment	ST Execution
1	219 ms	141 ms	6 ms
2	219 ms	126 ms	6 ms
3	234 ms	125 ms	5 ms
4	203 ms	141 ms	5 ms
5	204 ms	140 ms	6 ms
Ø	215,8 ms	134,6 ms	5,6 ms



Use-Case: Workflow-based Cloud Infrastructure Management →

© Pascal Hirmer

Summary and Outlook

SitRS is a general purpose, situation recognition service that can be used in different use-cases

- Summary
  - Defined **Situation Templates** to model situations based on the levels of the **Situation Detection Pyramid**
  - Defined a **Situation Model** for defining **Situation Objects** characterizing the state of the environment
  - Showed a way to **integrate different processing technologies**
    - Data streaming, Complex Event Processing, Internet of Things technologies
  - **Initial goal: Recognize situations based on raw sensor data achieved**

- Planned future work
  - Integrate other event and data stream processing technologies
  - Ontology based sensor registration and integration
  - Automatic situation recognition based on historic data to learn situations
  - Formalization of the SitRS approach

© Pascal Hirmer

SitRS – A Situation Recognition Service based on Modeling and Executing Situation Templates

Pascal Hirmer[1], Matthias Wieland[1], Holger Schwarz[1], Bernhard Mitschang[1],
Uwe Breitenbücher[2], Frank Leymann[2]
[1]Institute of Parallel and Distributed Systems,
[2]Institute of Architecture of Application Systems
University of Stuttgart, Stuttgart, Germany

University of Stuttgart
Germany

DFG Deutsche Forschungsgemeinschaft

IAAS IPVS

Backup – Use Case: SMART Factory

- SMART factories are an important part within the “Industrie 4.0” movement
  - Highly interconnected machines and robots work together to manufacture a product
- SitRS for SMART factories
  - Goal: recognize error situations, material shortage etc. & react automatically
  - Connect (machine) sensors to SitRS
  - Monitor the SMART factory using situation recognition
  - React on occurring situations accordingly

© Pascal Hirmer