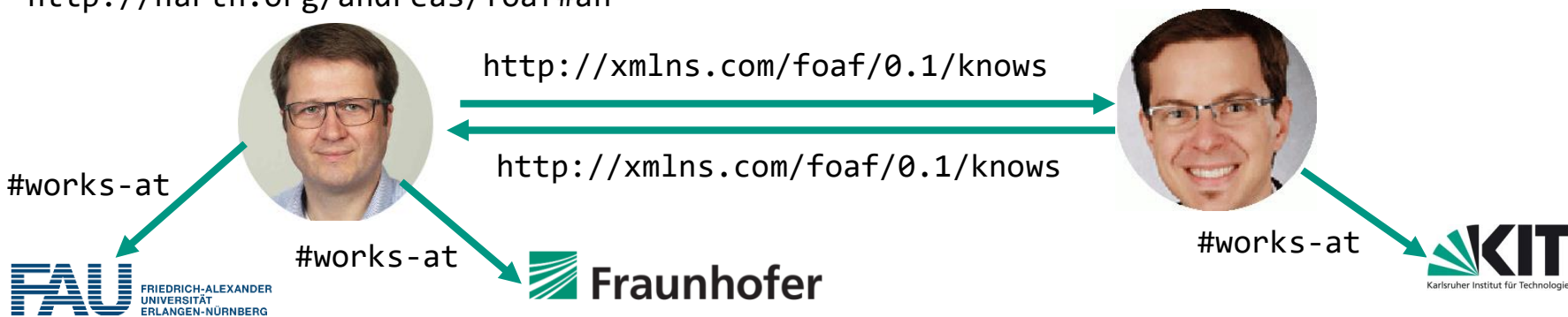


# Tutorial on Distributed Knowledge Graphs for the Web of Things, Part VIII: Workflows

Tobias Käfer (KIT) and Andreas Harth (FAU)

Tutorial @ 10<sup>th</sup> International Conference on the Internet of Things (IoT), 2020

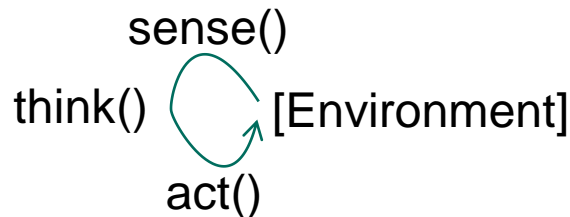
<http://harth.org/andreas/foaf#ah>



# How Far Away Are We From AI Agents on the Web of Things?

- Cognitive loop:
  - while true:
    - sense()
    - think()
    - act()
- Read-Write Linked Data gives us:
  - sense() and act() to interact with distributed sources
  - Knowledge Graphs to describe data

Russell / Norvig's Agent Layer Cake [1]	Ingredients
Agents with goals	Capability descriptions
<b>Agents with internal state</b>	<b>State Maintenance</b>
Simple reflex agents	Execution semantics
(Describe Perception)	Data model
(Perception/action means)	Interaction



[1] Russell and Norvig: "Artificial Intelligence – A Modern Approach". Prentice Hall (1995).

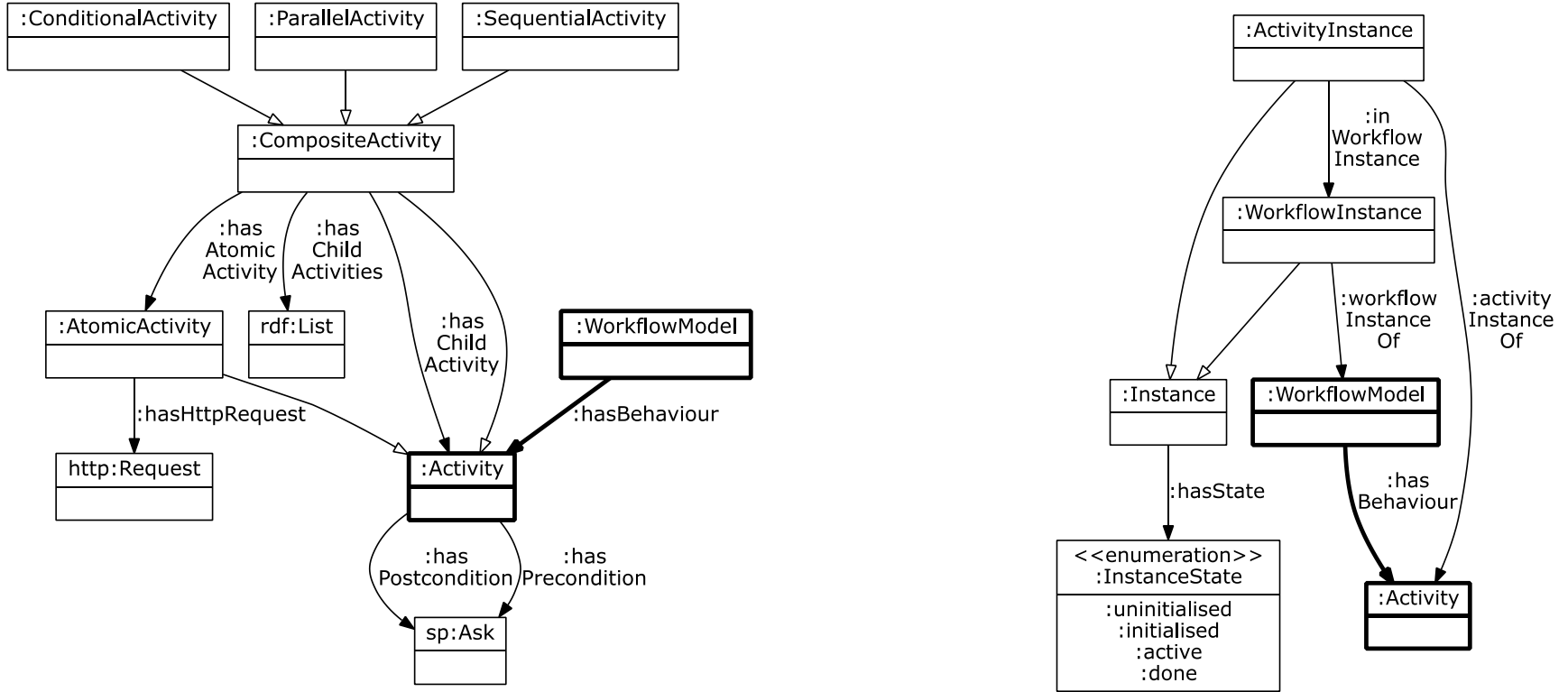
# i-VISION and Workflows in Linked Data

“SELECT the push-buttons in the Virtual Reality that are involved in the upcoming steps of the currently running take-off workflow and highlight them”



<http://www.ivision-project.eu/>

# The WiLD Ontology – our Workflow Language for Workflow Models and Instances



# WiLD Operational Semantics – Example Rule: Activate the First Activity of a Sequence



Excerpt

$wild: WorkflowInstance(? wfi) \wedge wild: hasState(? wfi, wild: active)$   
 $\wedge wild: workflowInstanceOf(? wfi, ? wfm)$

$\wedge wild: WorkflowModel(? wfm) \wedge wild: containsActivity(? wfm, ? parentact)$   
 $\wedge wild: SequentialActivity(? parentact) \wedge wild: hasChildActivities(? parentact, ? act1, ? act2, \dots)$

$\wedge wild: instanceof(? parentacti, ? parentact) \wedge wild: hasState(? parentacti, wild: active)$   
 $\wedge wild: instanceof(? act1i, ? act1) \wedge wild: hasState(? act1i, wild: initialised)$

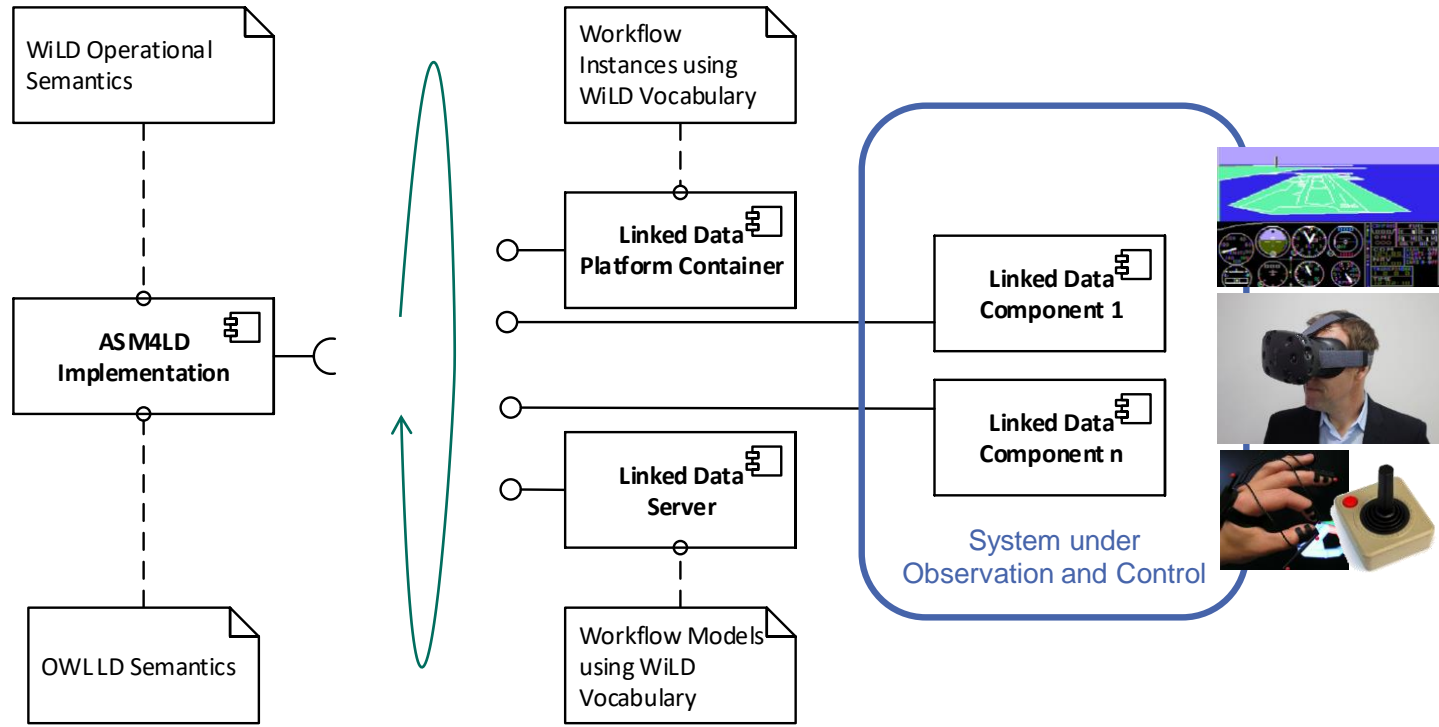
$\wedge wild: inWorkflowInstance(? parentacti, ? wfi) \wedge wild: inWorkflowInstance(? act1i, ? wfi)$   
 $\wedge wild: hasHttpRequest(? act1, ? req)$

cf. <http://purl.org/wild/semantics>



- Setze **?act1i** auf **aktiv** mittels HTTP PUT
- Führe HTTP request **?req** aus

# Systemarchitektur mit Workflowmanagement-System, Systemzustand



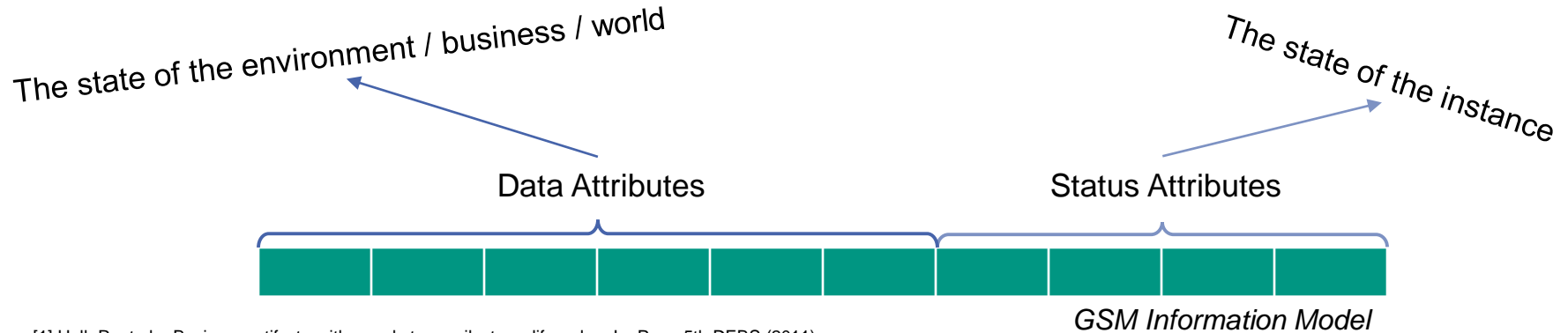
# i-VISION Video

# Guard-Stage-Milestone [1] (Syntax, roughly)

- Stage: some form of activity
- Guard\*: has to be passed before stage can execute
- Milestone\*: objective significant during execution time



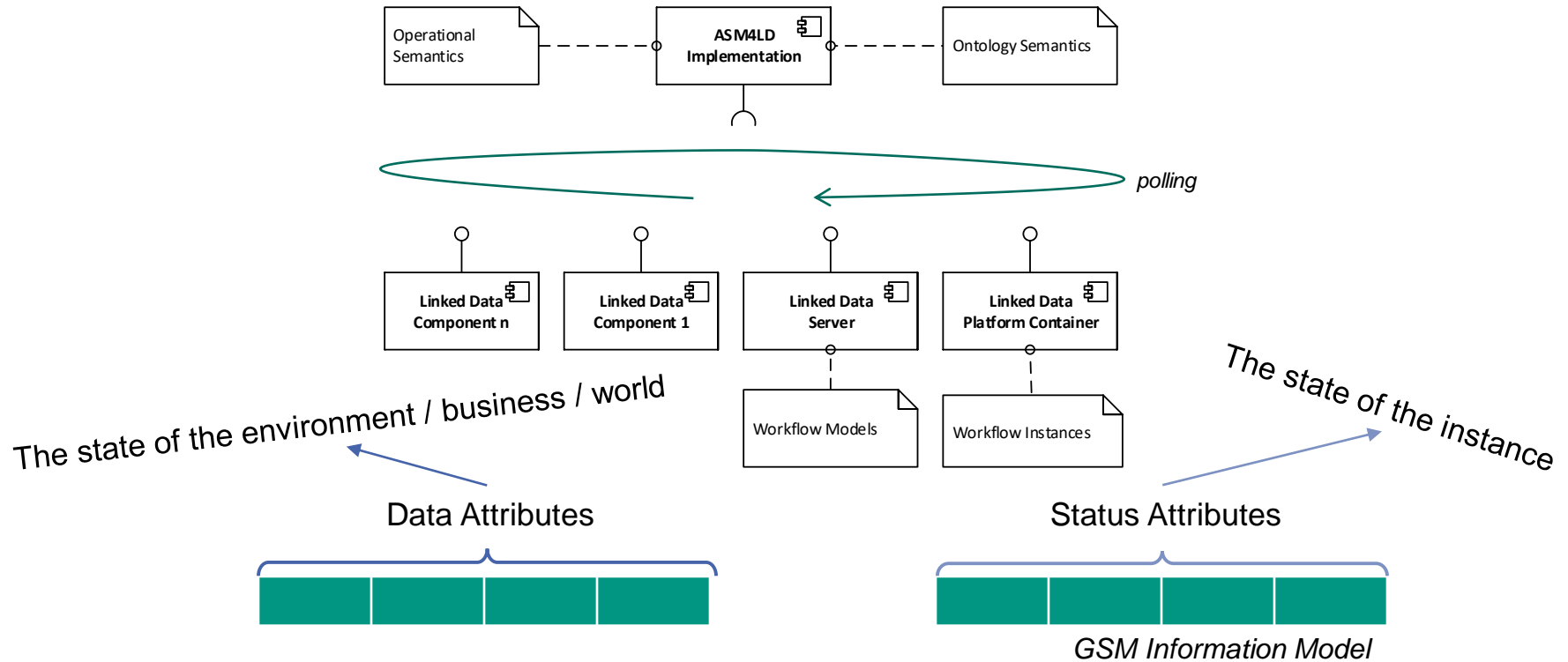
\* Comes with conditions (“sentries”) that range over the information model



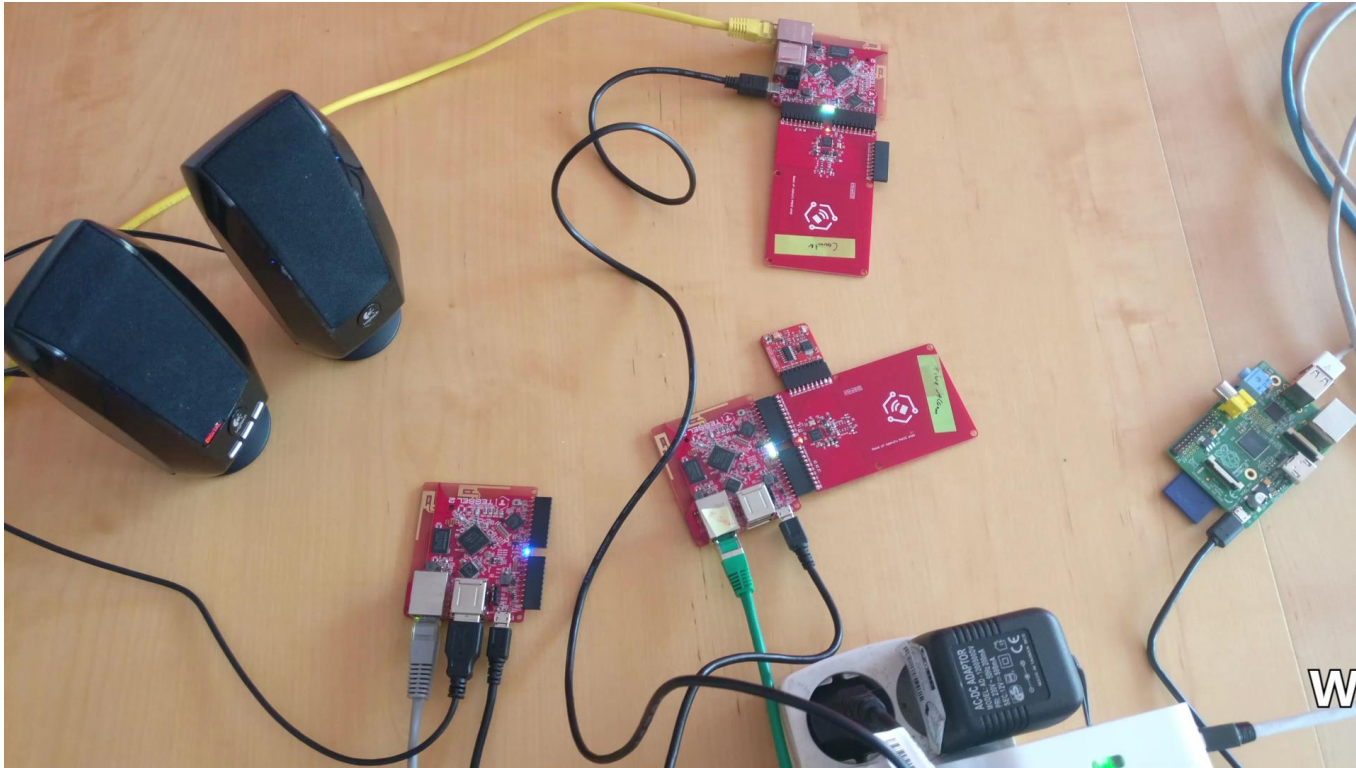
[1] Hull, R, et al. : Business artifacts with guard-stage-milestone lifecycles. In: Proc. 5th DEBS (2011)



# Architecture



# Building Automation Demo @SEMANTiCS19



<http://people.aifb.kit.edu/co1683/2019/gsm/semantics-demo/GSM.mp4>