# Stream Reasoning and Linked Data: A Clash of Cultures

Andreas Harth
Karlsruhe Institute of Technology (KIT)
2nd RDF Stream Processing Workshop
ESWC 2017

# Outline

- Act I: Status quo and life-changing event
- Act II: We have achieved some alignment but still no dice
- Act III: An attempt at convergence

# ACT I

Status quo and life-changing event

# „Nothing Endures But Change“

- Current and future scenarios around the Internet of Things and the Web of Things require processing of (near) real-time data

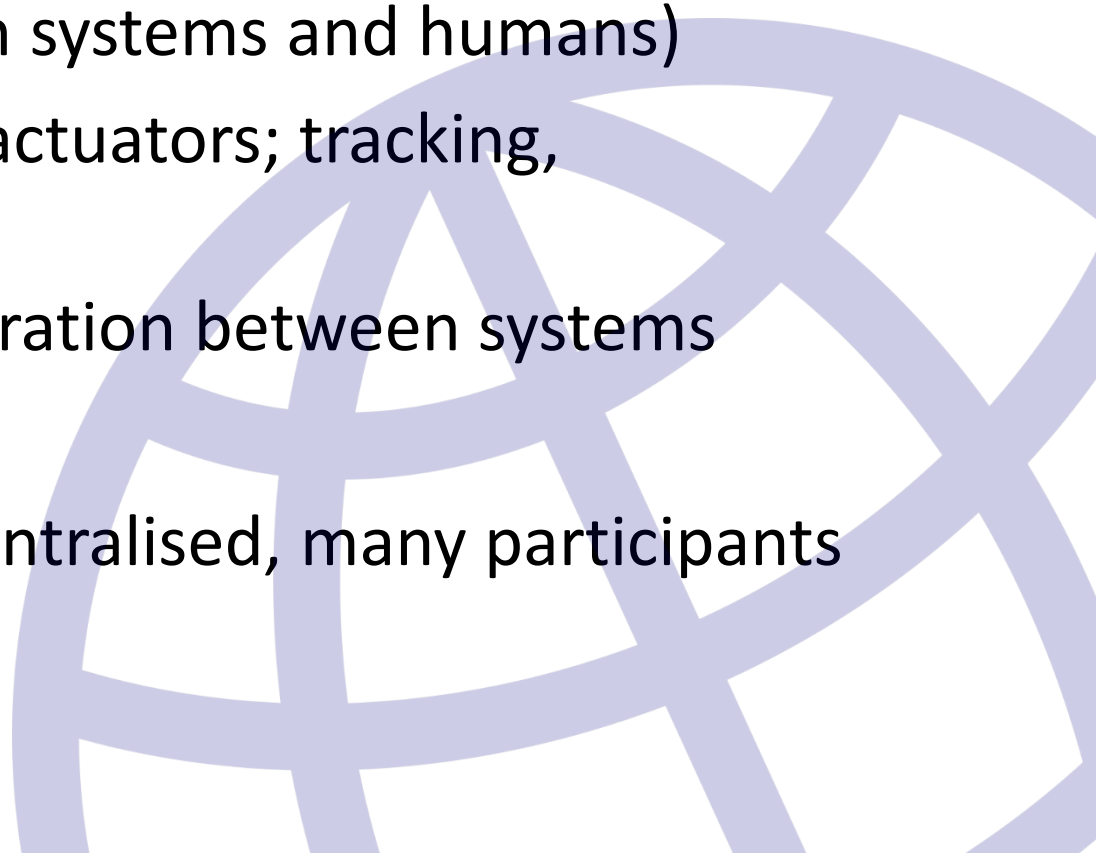
- Frillions of sensors in the near future

- We need systems for processing live data (e.g., read a temperature sensor or a website clickstream)
- The systems should also support actuators (e.g., set a heating/cooling system or send an email message)

# Requirements

- Process data in near-realtime (sub-second) for live query, control, virtual reality/augmented reality
- Handle dynamics and behaviour (of both systems and humans)
- Connect different subsystems (sensors, actuators; tracking, simulation, rendering…)
- Integration of data sources and interoperation between systems

- Web-scale: simple, elegant, robust, decentralised, many participants



# Which Data Model? Which Data Format?


{JSON}

protobuf
Protocol Buffers

AVRO™

Microsoft .net

RDF

Java
RMI

<xml />

CAP'N PROTO
cerealization protocol

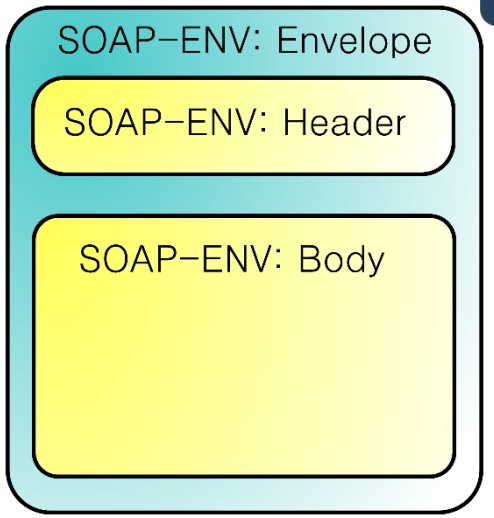
# Which Network Protocol?





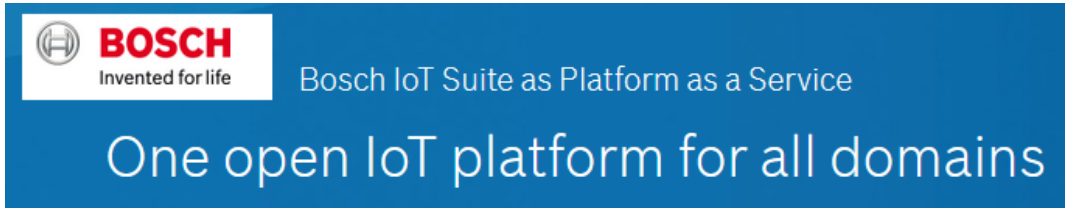








# Which IoT Platform?







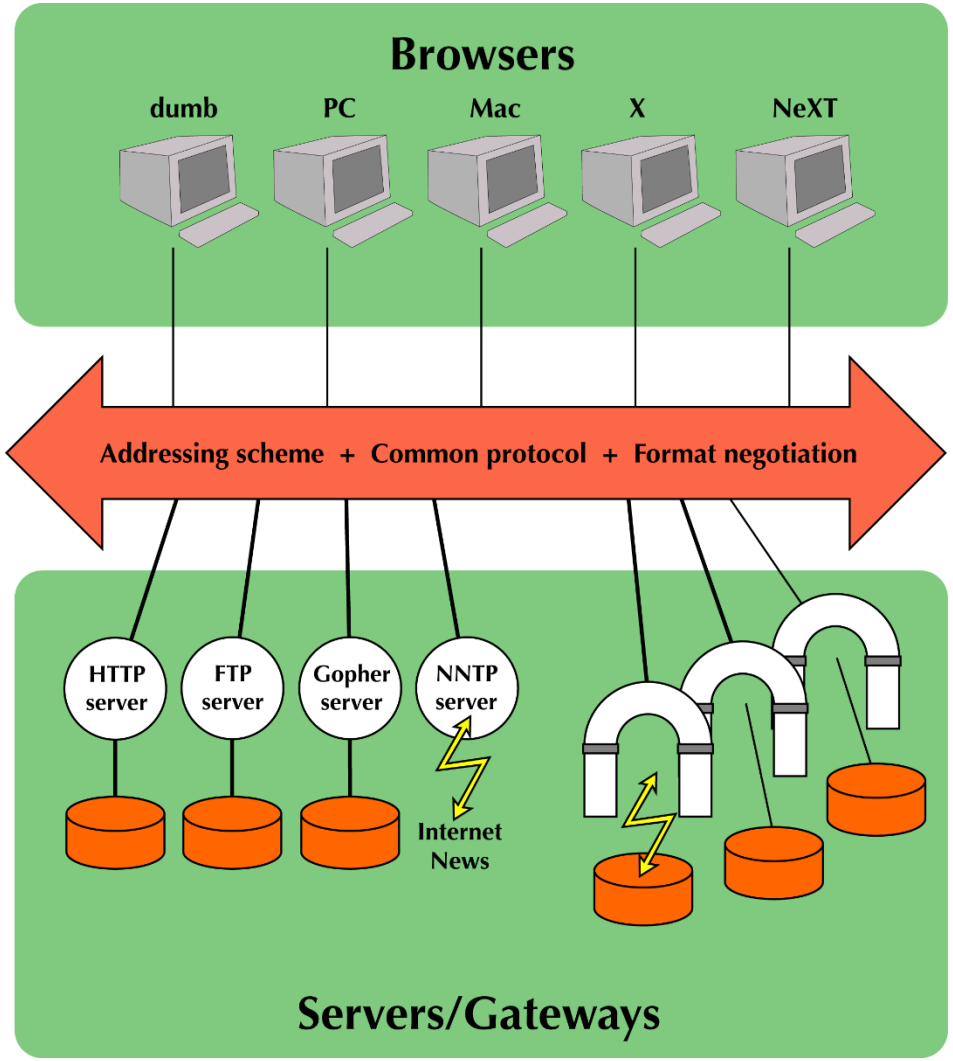





# Semantic Web


- Semantic: Why not agree on RDF as data model?
- Web: Why not agree on HTTP as network protocol?
- Web: Why not agree on a decentralised architecture?

# Web Architecture (1990)

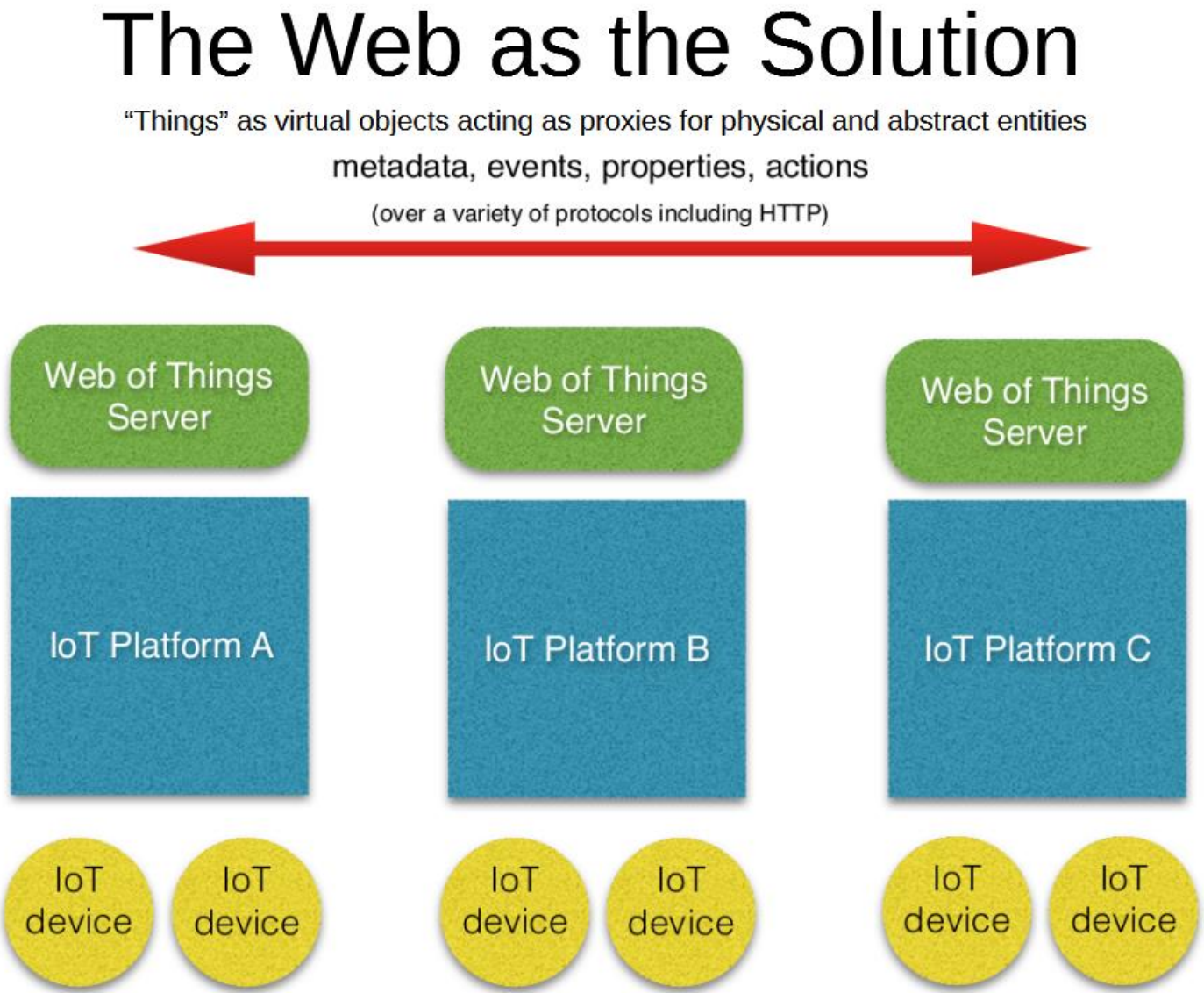



https://www.w3.org/People/Berners-Lee/1996/ppf.html

# Linked Data Architecture (2009)


https://www.w3.org/2009/Talks/0204-ted-tbl/#(7)

# Web of Things Architecture (2015)



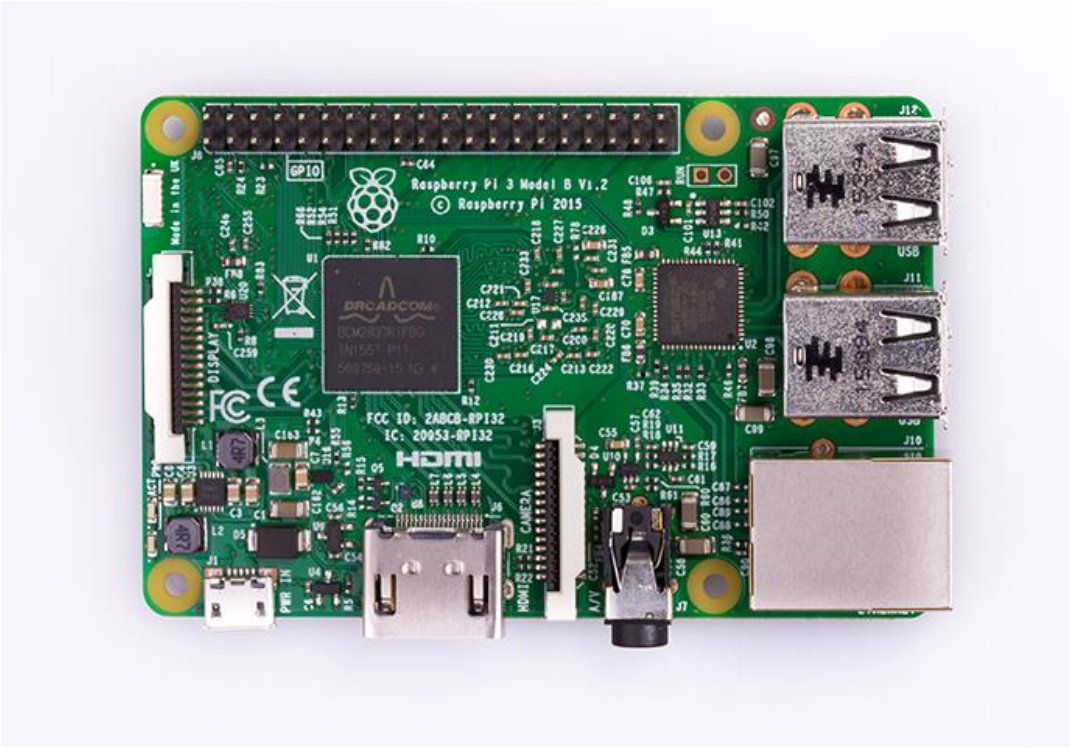

https://www.w3.org/2015/05/wot-framework.pdf

# ACT II

We have achieved some alignment but still no dice

# Linked Data Interface

- Sensor has URI (e.g., http://localhost/sensor/temp)
- GET on http://localhost/sensor/temp returns **current** reading

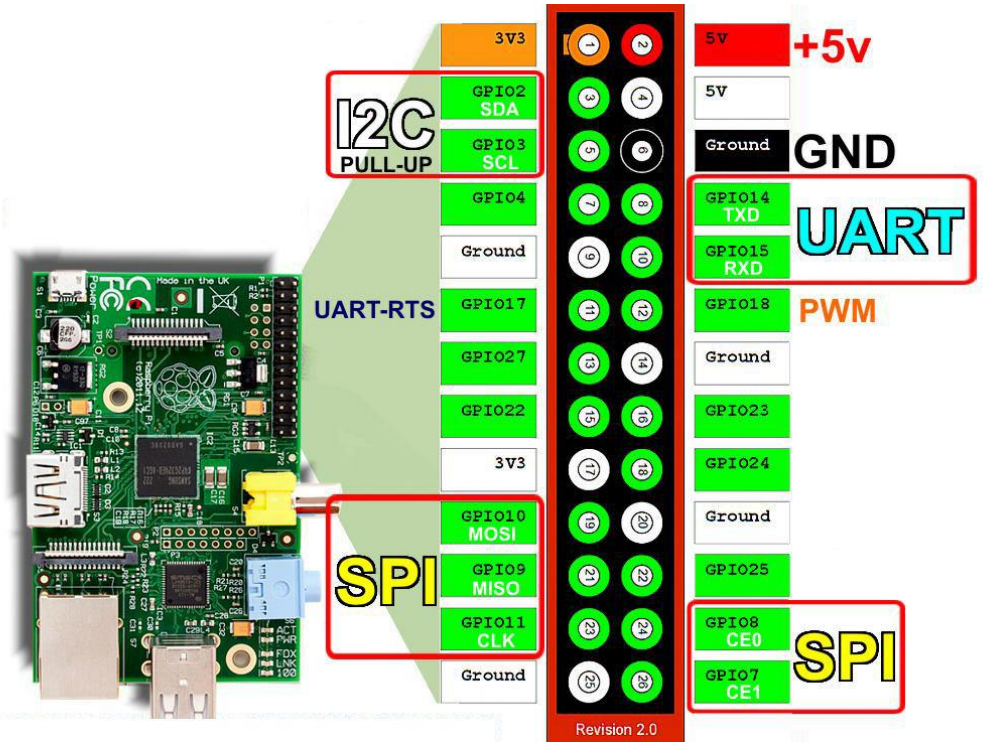


# SPARQL Query Current Temperature

```
SELECT ?temp
FROM <http://localhost/sensor/temp>
WHERE {
  ?x rdf:value ?temp .
}
```

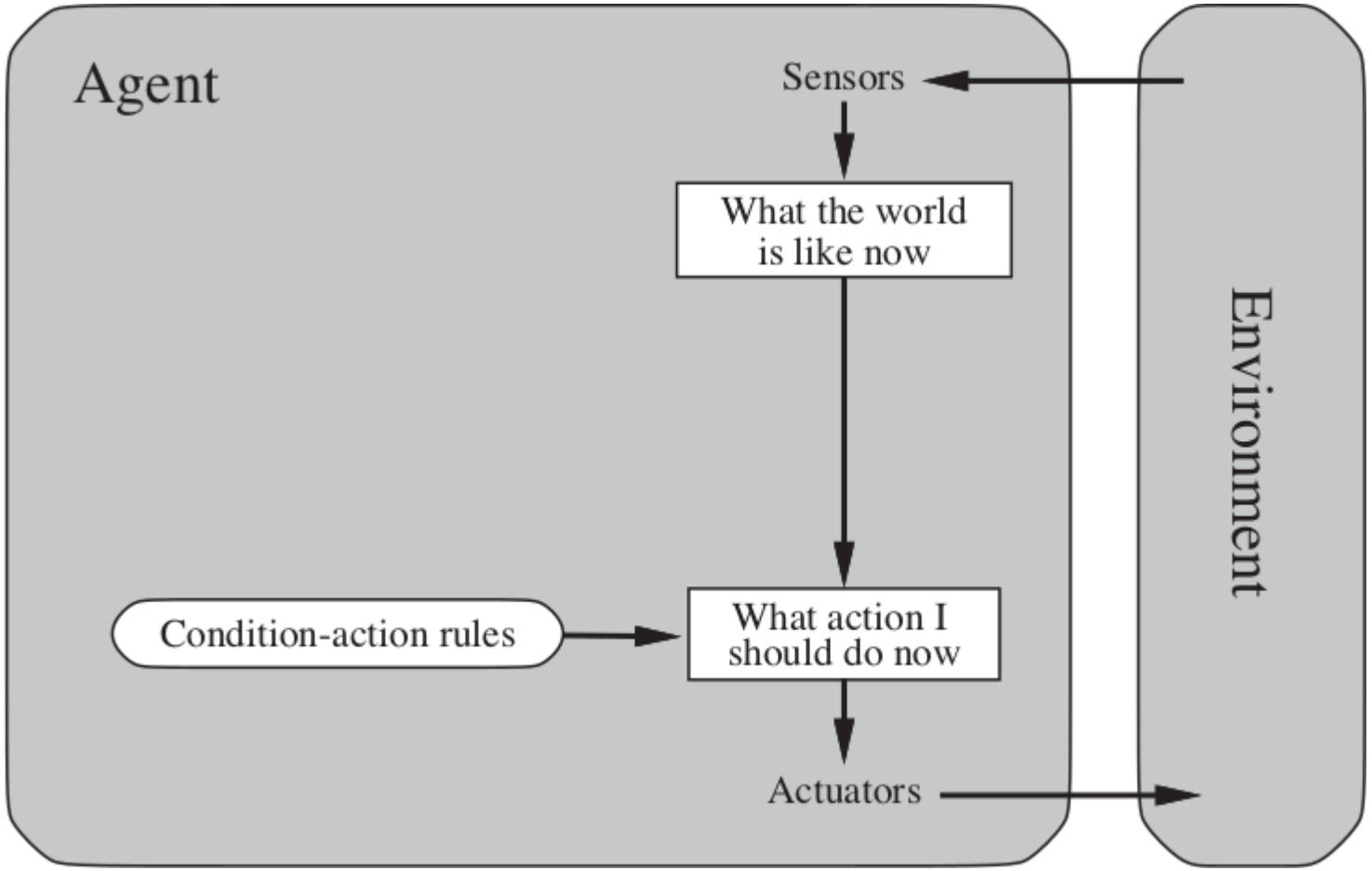
# Read-Write Linked Data Interface

- Actuator has URI (e.g., http://localhost/actuator/gpio1)
- PUT on http://localhost/actuator/gpio1 switches an electric consumer on or off (with the right RDF message body)



# Simple Reflex Agents

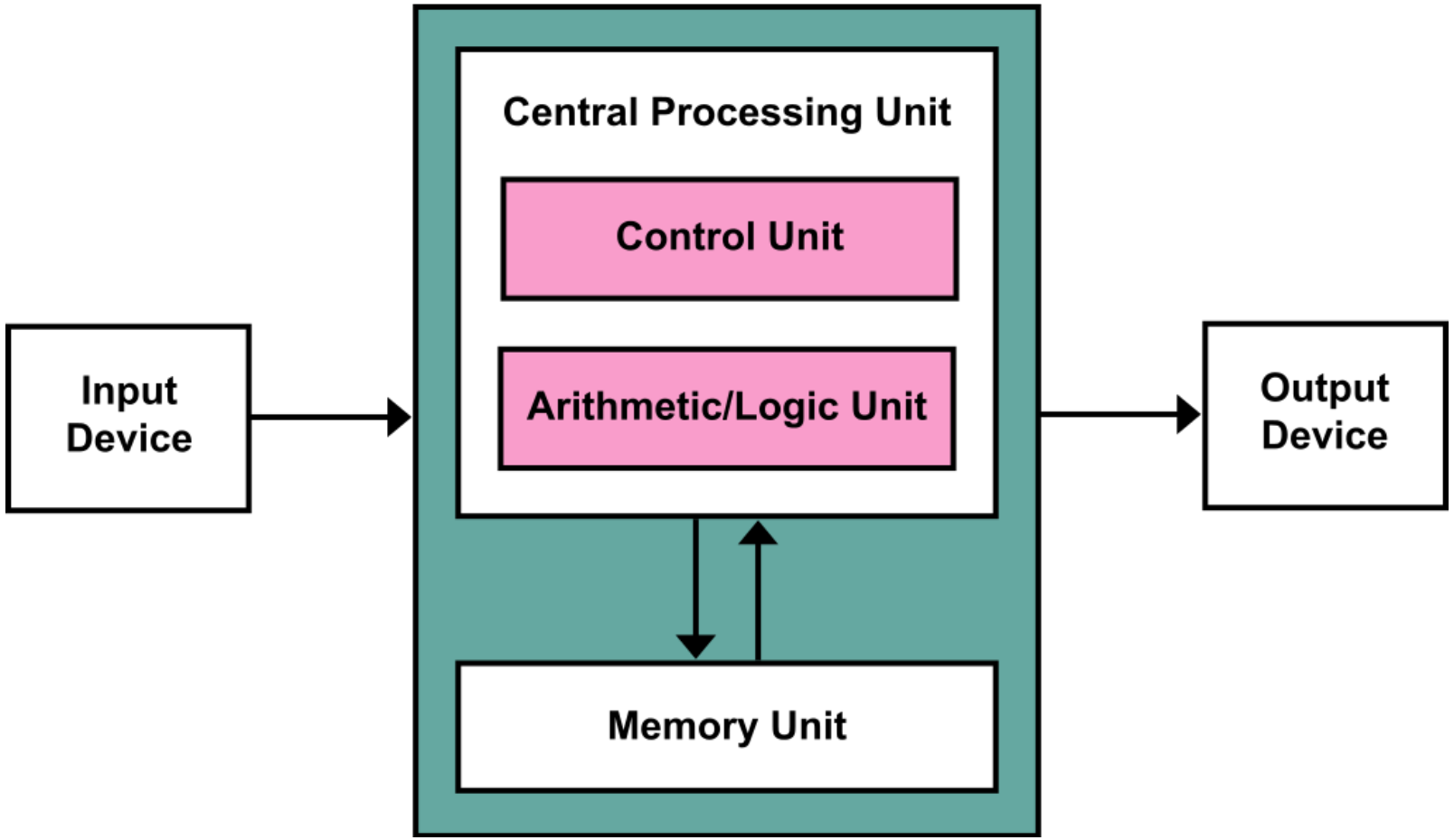
Or other cognitive architectures: SOAR, ACT-R…




Russel/Norvig. Artificial Intelligence – A Modern Approach, Third Edition, Pearson Education, 2010.

# Von-Neumann Architecture (1945)

von Neumann, John (1945), First Draft of a Report on the EDVAC.



# Simple Thermostat Reflex Agent

- Loop forever:
  - GET `http://localhost/sensor/temp`
  - Query `?x rdf:value ?temp .`
  - If ?temp > 25 °C:
    - PUT `http://localhost/actuator/heating`
    - With some RDF to set heating to OFF
  - If ?temp < 20 °C:
    - PUT `http://localhost/actuator/heating`
    - With some RDF to set heating to ON
  - Wait 1 minute

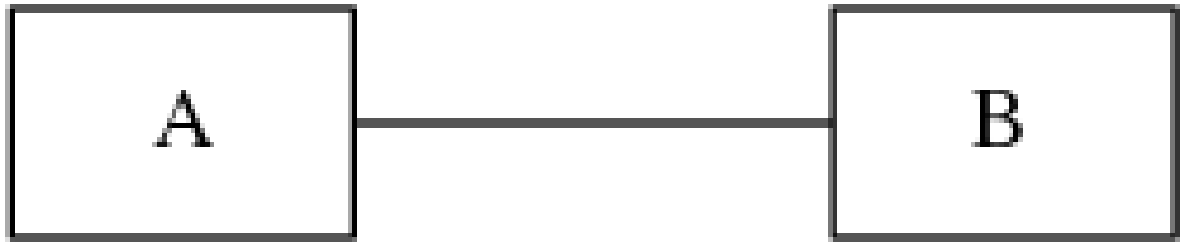
- The Reflex Agent is acting as user agent

# But I Want Streaming!

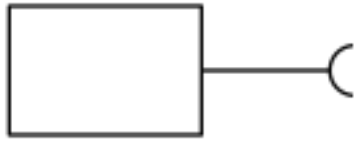
http://firehose.sandia.gov/images/firehose.jpg

# Uniform Network Interface

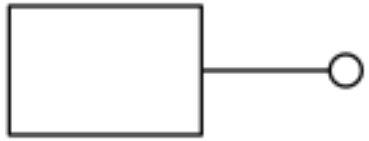
- Components A and B, data flows from A (source/producer) to B (sink, consumer)




- REST assumes request/response communication pattern between components with client connector and server connector
  - Clients emit requests, receive response



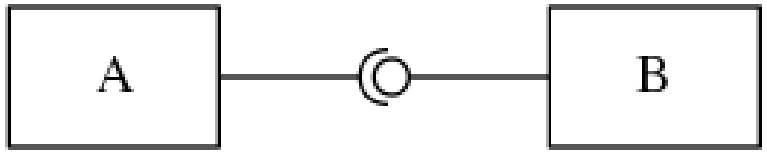
  - Servers answer to incoming requests with a response



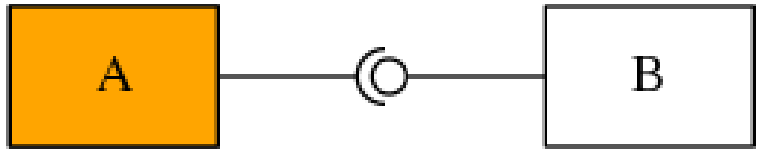
# Network Interface: Push vs. Pull

**Push**

- A is client, B is server
- A emits PUT request
- At A: B.put(value)

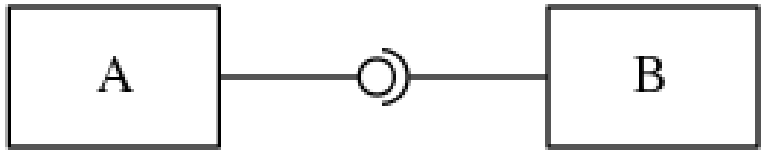



- Loop at A

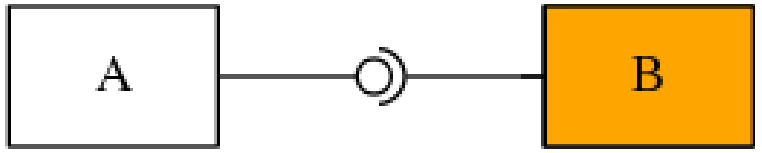



**Pull**

- A is server, B is client
- B emits GET request
- At B: value = A.get()




- Loop at B



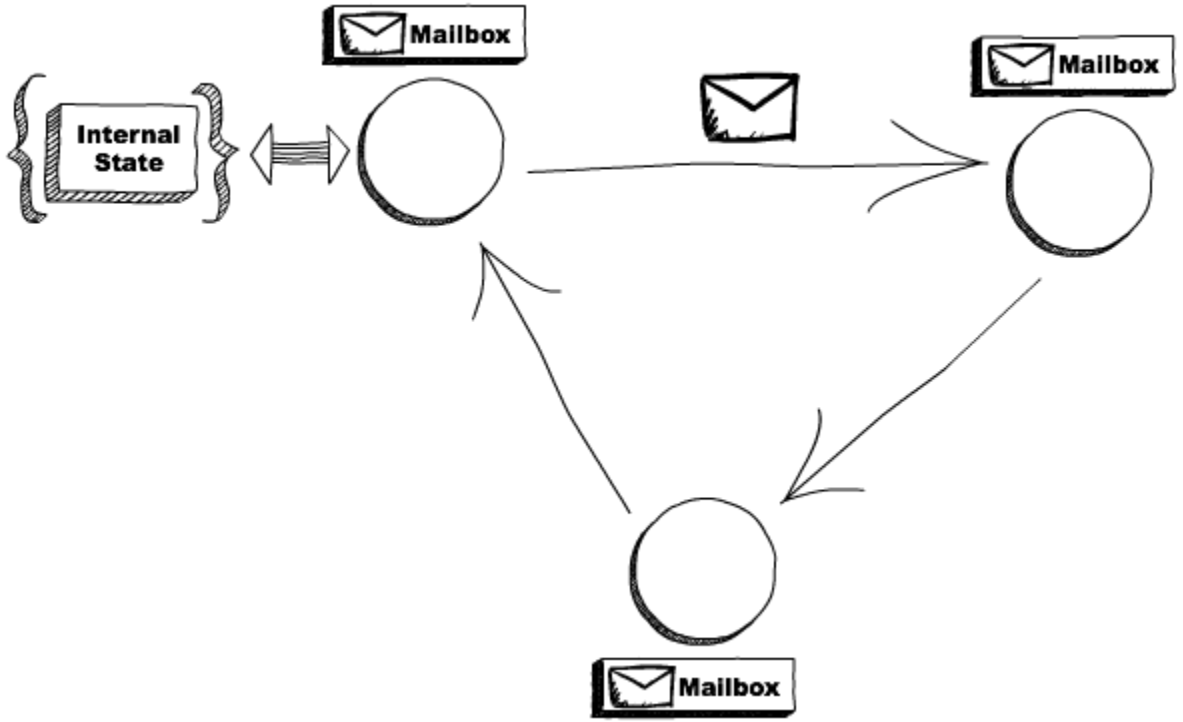
# Bi-directional Communication in HTTP

- Stream processor acts as server (which processes incoming HTTP requests)
- For publish/subscribe, each component is both client and server (a transducer)

# Actor Model (1973)

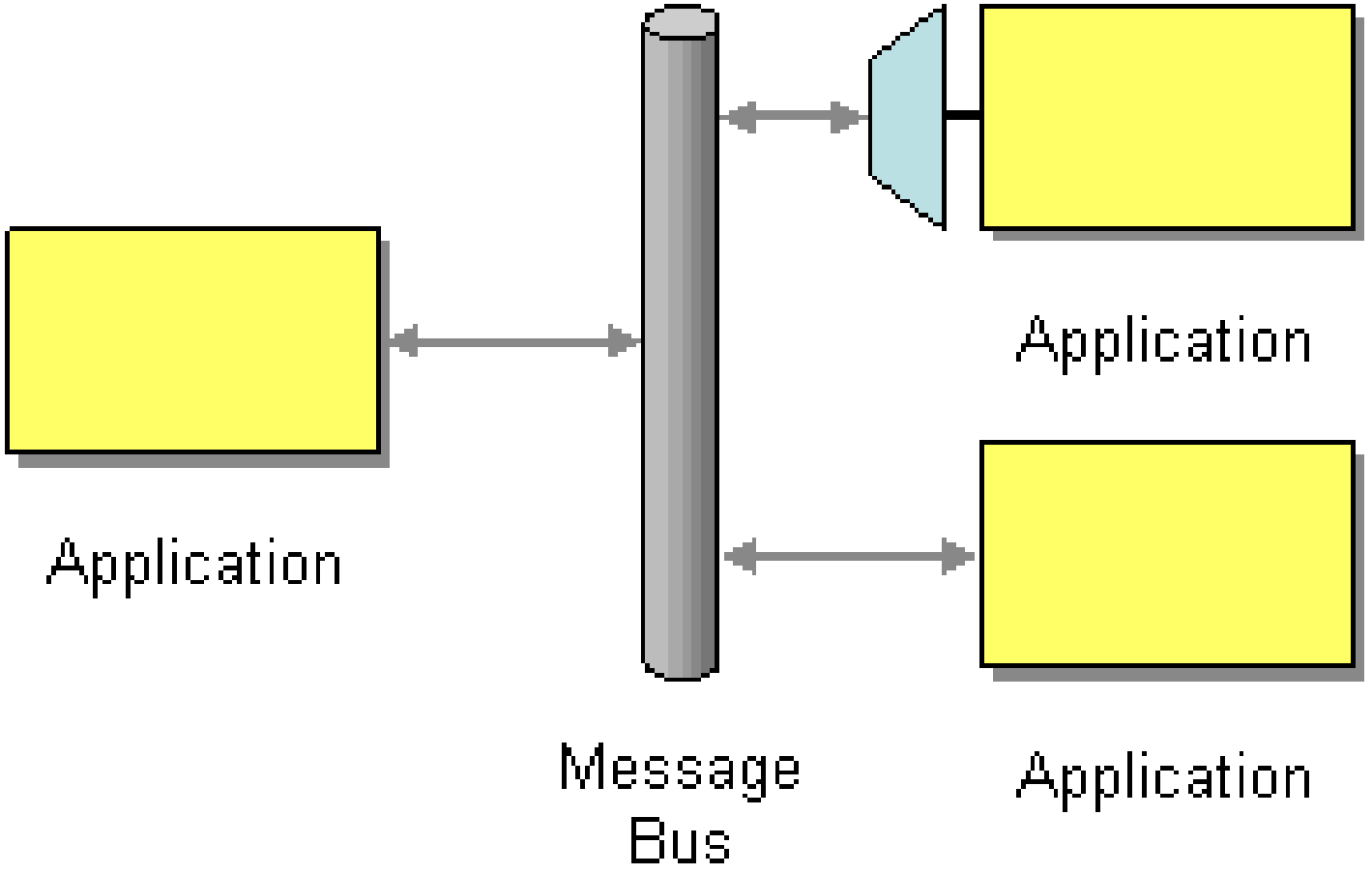
Carl Hewitt; Peter Bishop; Richard Steiger (1973). "A Universal Modular Actor Formalism for Artificial Intelligence". IJCAI.

“Actors communicate with each other by sending asynchronous messages. Those messages are stored in other actors' mailboxes until they're processed.”




http://www.brianstorti.com/the-actor-model/

# Message Bus (for Stream Processing)




Message bus systems centralise the actors‘ mailboxes

Examples are:

- ROS
- Kafka
- …

http://www.enterpriseintegrationpatterns.com/img/MessageBusSolution.gif

# Protocol/Architecture

**Linked Data**

- Distinction between client (user agent) and server
- User agents invoke requests, servers answer with responses
- Server is persistent
- No central message broker needed

**Stream Processing**

- No clear distinction between client and server
- All components can send messages and receive messages
- All components are persistent
- Some variants rely on central message broker

# Time

The New Media Reader, edited by Noah Wardrip-Fruin and Nick Montfort, book design by Michael Crumpton, 2003, http://www.newmediareader.com/,
50. Time Frames (from Understanding Comics), Scott McCloud, 1993

# Aspect: State vs. Event

- Event: Leonard Cohen is born on September 21, 1934

```
[] a :Event ; :type :birth ; :date "1934-09-21" .
```

- State: Leonard Cohen

```
:lc :birthDate "1934-09-21" .
```

- Event: Leonard Cohen dies on November 7, 2016

```
[] a :Event ; :type :death ; :date "2016-11-07" .
```

- State: Leonard Cohen

```
:lc :birthDate "1934-09-21" .
:lc :deathDate "2016-11-07" .
```

https://www.youtube.com/watch?v=YD6fvzGIBfQ

# Aspect of Data Representation

**Linked Data**

- Messages represent resource state
- Getting the combined current state of resources involves GETs on URIs

**Stream Processing**

- Messages represent events
- Some messages represent state as well
- Getting the combined current state of resources requires integration of events

# ACT III

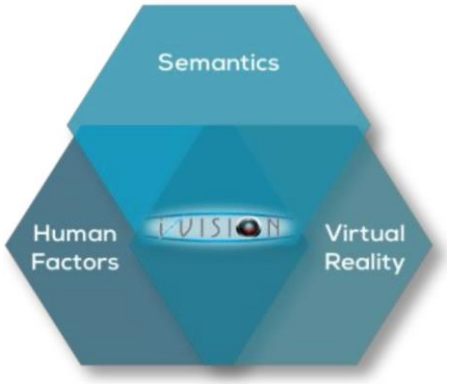
An attempt at convergence

# Selected Projects Using Reflex Agent Model

- ARVIDA: Applied Reference Architecture for Virtual Services and Applications
  - Break up monolithic industrial applications into distributed applications
  - Components based on RESTful interfaces
  - RDF, RDFS and OWL for knowledge representation (vocab.arvida.de)
- i-VISION: Immersive Semantics-based Virtual Environments for the Design and Validation of Human-centred Aircraft Cockpits
  - Human-Cockpit Operations Analysis
  - Semantic Virtual Cockpit
  - Virtual Cockpit Design Environment

Bundesministerium für Bildung und Forschung

ARVIDA




i-VISION

# Demo Setup


Kratzer vermeiden | 1. Loch in Fahrtrichtung
Dorn nicht bis Anschlag einstechen

# Demo Setup

# Linked Data Notifications

## W3C Recommendation 2 May 2017

**Editors**

Sarven Capadisli, University of Bonn, info@csarven.ca

Amy Guy, University of Edinburgh, amy@rhiaro.co.uk

## Abstract

Linked Data Notifications is a protocol that describes how servers (receivers) can have messages pushed to them by applications (senders), as well as how other applications (consumers) may retrieve those messages. Any resource can advertise a receiving endpoint (Inbox) for the messages. Messages are expressed in RDF, and can contain any data.

# Open Questions Stream Processing

- When to trigger query evaluation?
  - Every triple? (event-triggered)
  - Every incoming request? (event-triggered)
  - Every time slice? (time-triggered)
- When to trigger actions?
- How to integrate event and state messages?
- How to perform reasoning?
  - Over each message individually?
  - Over entire state at each time step?

# Conclusion

- In the Semantic Web community, building on HTTP and RDF seems reasonable (to me)
- Linked Data and Stream Reasoning systems have different characteristics, leading to differences in user agent/server roles
- In dynamic environments, the modelling of aspect warrants some attention, especially in conjunction with request/response and reasoning
- Reflex agents provide a simple model for operating on resource state
- Linked Data notifications could provide a model for the handling of events in stream processing and stream reasoning systems

# Linked Data Processing vs. Stream Processing

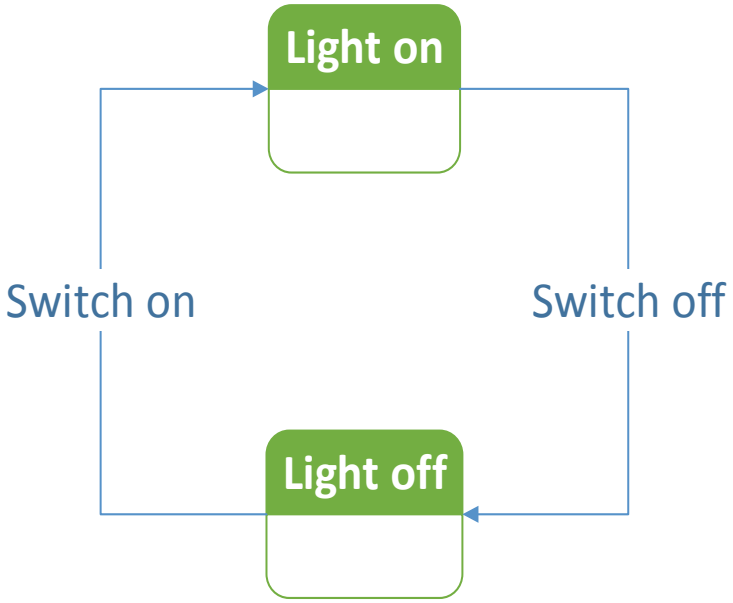
- What is the current temperature?

```
SELECT ?temp
FROM <http://localhost/sensor/temp>
WHERE {
  ?x rdf:value ?temp .
}
```

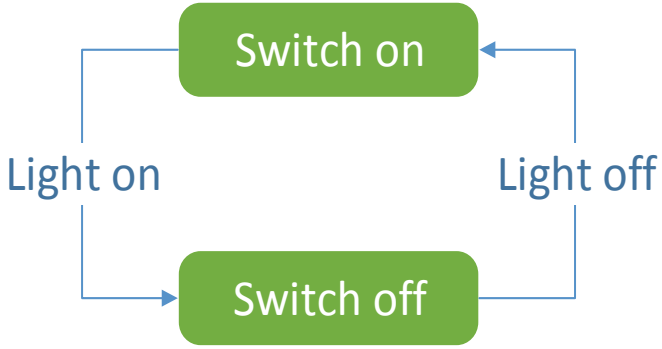
- Notify me when somebody accesses the web page foo.html!

```
tail -f access.log
| grep –e foo.html
| while read -r line; do
 mail -s „hit“ harth@kit.edu;
done
```

# UML State Diagrams vs. UML Activity Diagrams




Boxes represent states
Arrows represent events (transitions)




Boxes represent activities
Arrows represent states

# Further Reading


- Andreas Harth, Tobias Käfer, Felix Leif Keppmann, Dimitri Rubinstein, René Schubotz, Christian Vogelgesang. "Flexible industrielle VT-Anwendungen auf Basis von Webtechnologien". VDE Kongress 2016, Internet der Dinge, Nov 7-8, 2016, Mannheim, Germany.
- Tobias Käfer, Sebastian Bader, Lars Heling, Raphael Manke and Andreas Harth. "Exposing Internet of Things Devices on REST and Linked Data Interfaces". 2nd International Workshop on Interoperability & Open Source Solutions for the Internet of Things. Co-located with 6th International Conference on the Internet of Things (IoT 2016). Nov 7, 2016, Stuttgart, Germany.
- Felix Leif Keppmann, Maria Maleshkova, Andreas Harth. "Semantic Technologies for Realising Decentralised Applications for the Web of Things". 21st International Conference on Engineering of Complex Computer Systems, Nov 6-8, 2016, Dubai, UAE.
- Sarah Brauns, Tobias Käfer, Dirk Koriath, Andreas Harth. "Individualisiertes Gruppentraining mit Datenbrillen für die Produktion". GI-Jahrestagung 2016.
- Andreas Harth, Tobias Käfer. "Towards Specification and Execution of Linked Systems". 28. GI-Workshop Grundlagen von Datenbanken, May 24 - 27, 2016, Nörten-Hardenberg, Germany.