Industrial Agents and Distributed Agent-based Learning

Stefan Bosse

University of Bremen

Department of Mathematics & Computer Science,

ISIS Scientific Centre for Sensorial Materials

Bremen, Germany

Overview

A. From passive Sensors to Smart Sensor Nodes

B. The interdisciplinary approach: The design of Sensorial Materials

C. Smart Sensor and Mobile Networks: Metrics, Features, Capabilities, Challenges

D. The Internet-of-Things and Sensor Networks

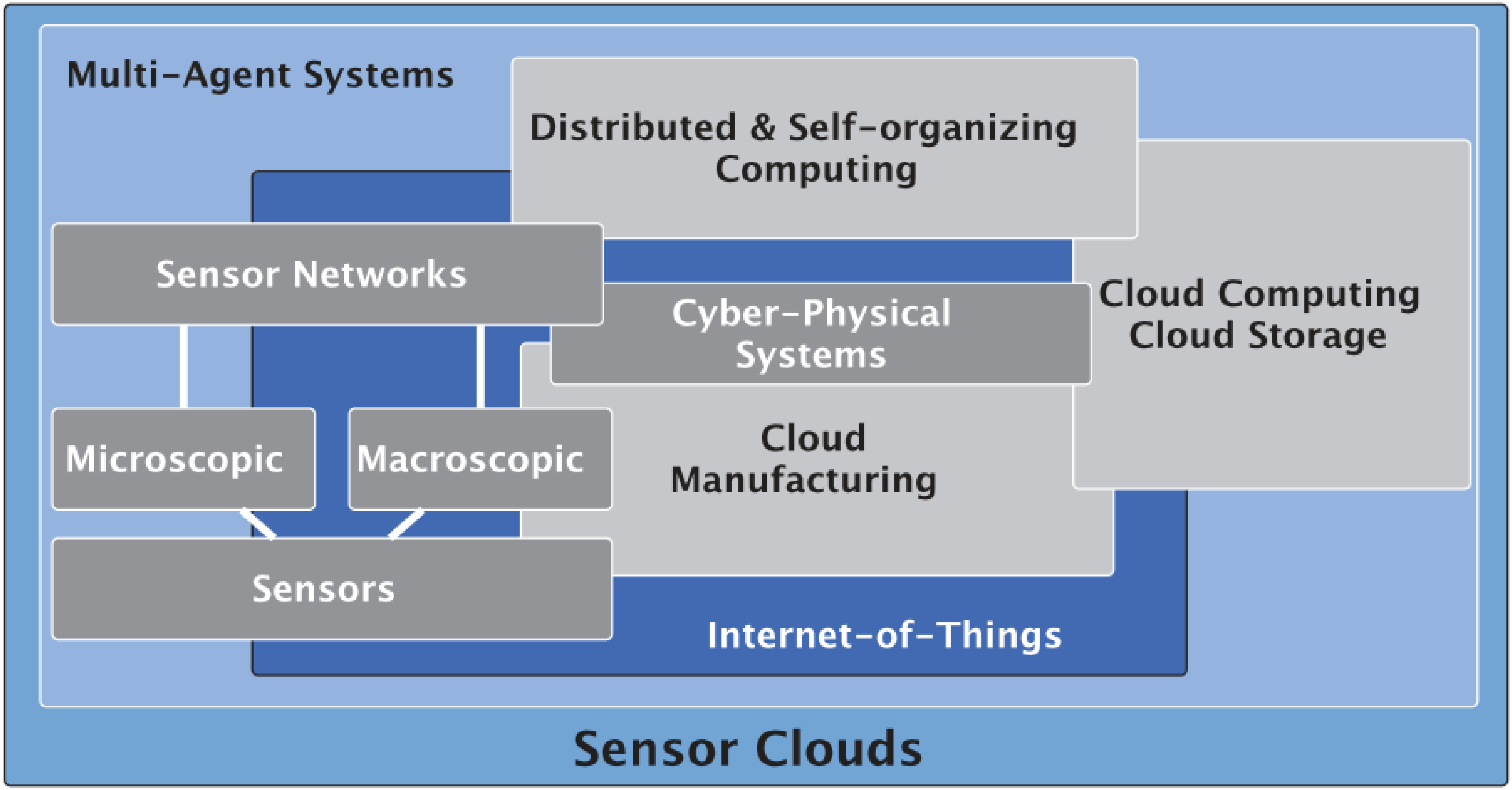
E. Distributed Computing, Big Data Processing, and Clouds
From the "Internet of Things" to the "Sensor Internet"

F. Architectures and Models for Distributed Computing:
The Mobile Agent, Multi-Agent and Self-organizing Systems

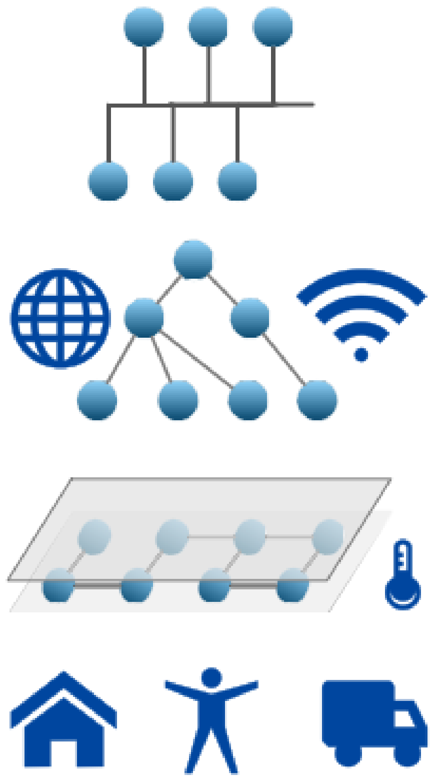
G. Use Cases: Structural Load Monitoring - Distributed Earthquake Monitoring with Ubiquitous Devices and Agents - Cloud Manufacturing

Overview (cont.)

The Future of Ubiquitous and Pervasive Computing

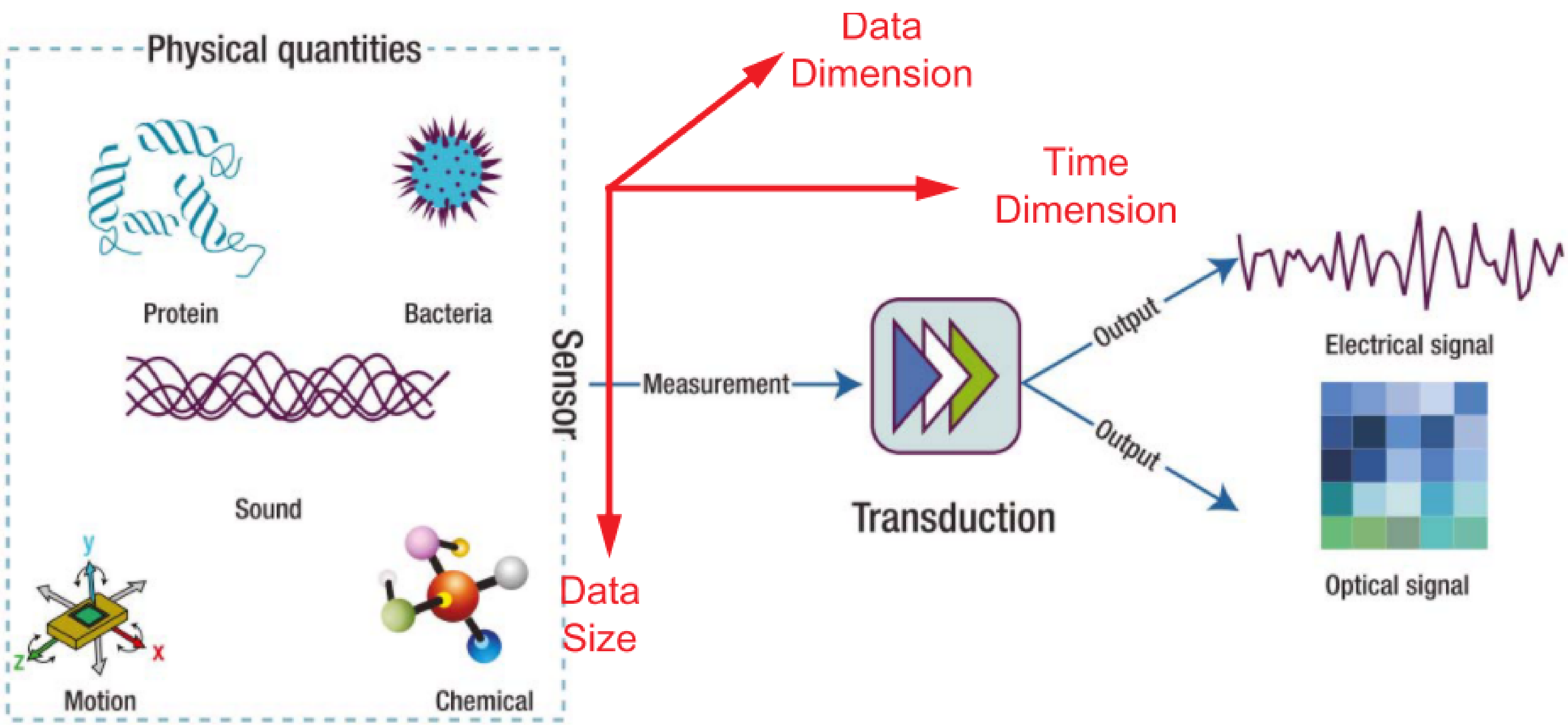


A. From passive Sensors to Smart Sensor Nodes



A. The Sensing Process

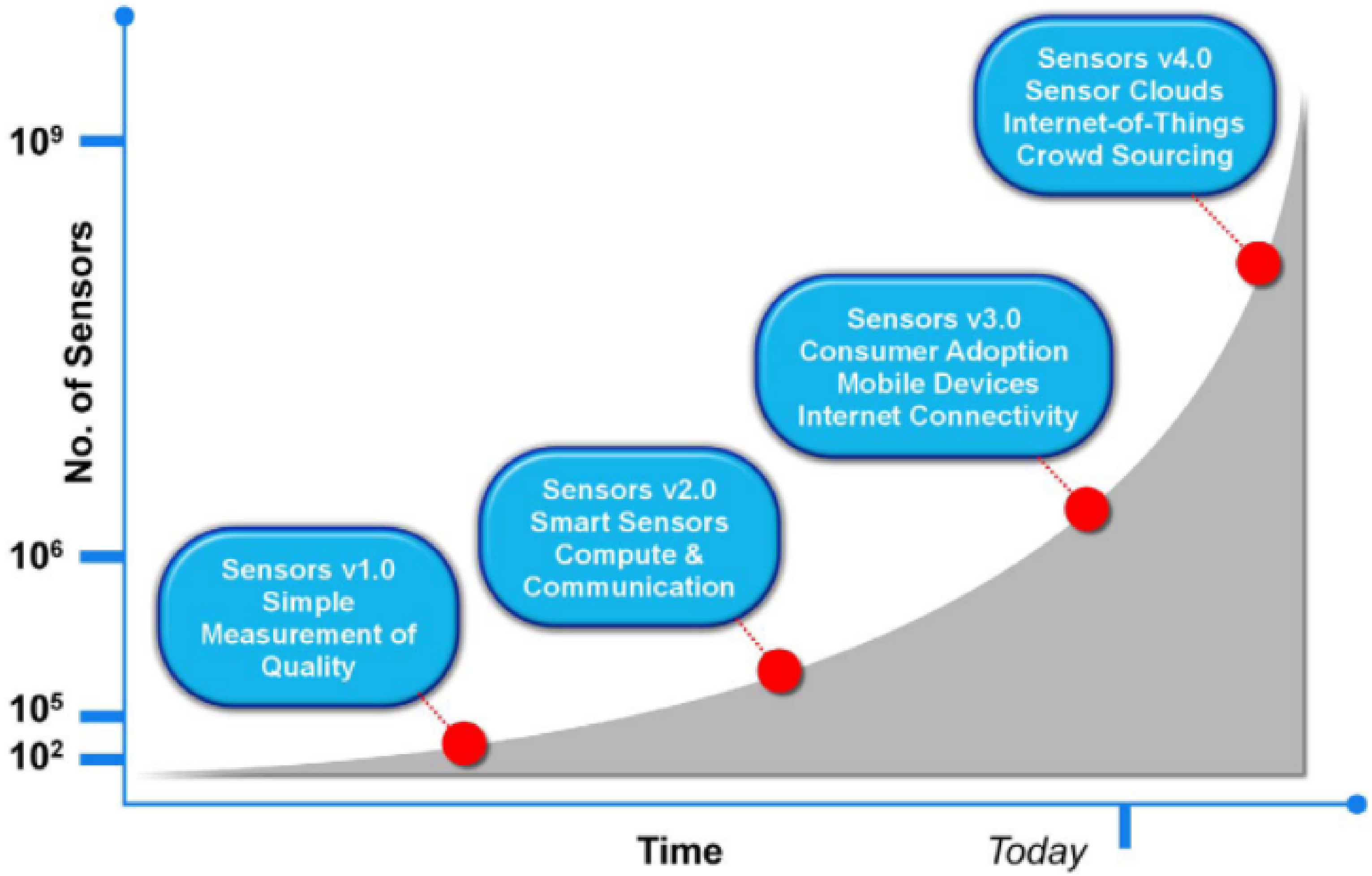
» Broad diversity of Sensors and Sensor Information for Processing [kind, size, dimension]



[McGrath et al., ST, 2014]

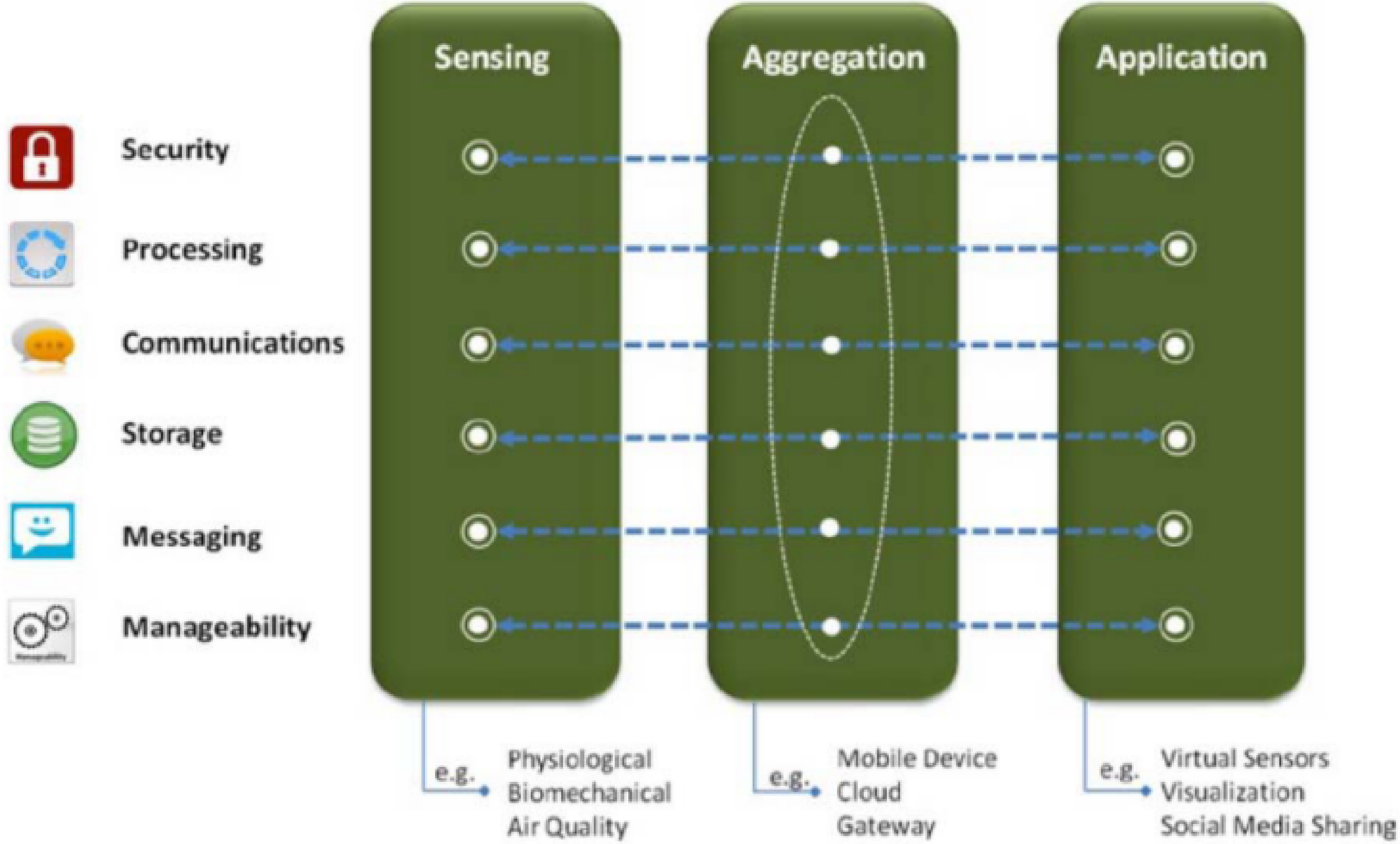
A. Sensor Integration - Progress in Density and Operation

- In the past decades: From passive to Smart Sensors with ICT
- From single sensor to sensor net-works
- A significant increase of sensor node density!



A. Sensor Data Processing: Functional Flow

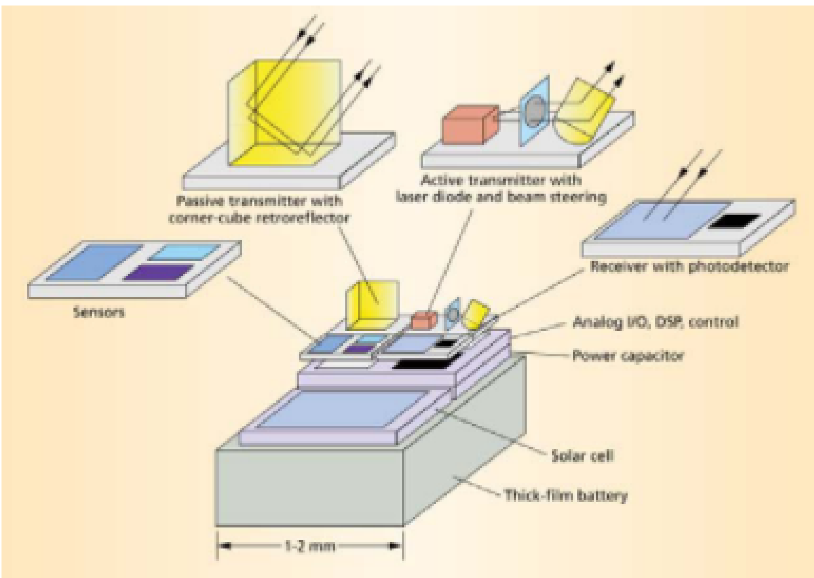
» Functional representation of software components in a Sensor Network



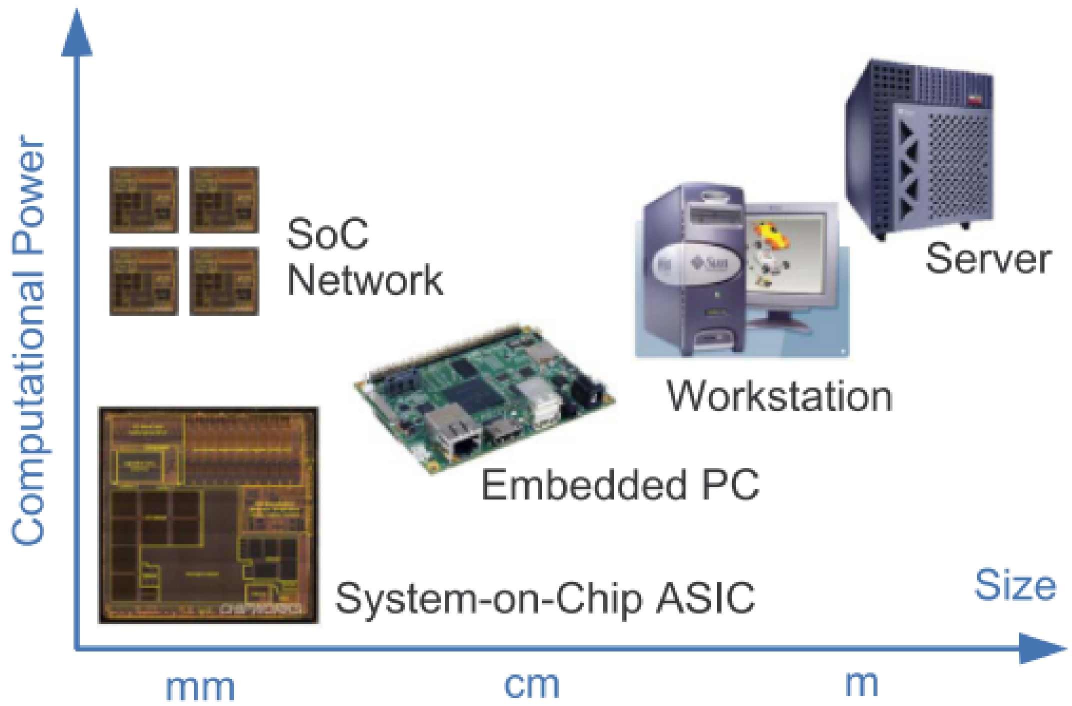
[McGrath et al., ST, 2014]

A. Computational Power and Complexity

- Traditionally software is executed on computers with high computational power and memory capacity.
- Approximation: (Computing Power $\times$ Storage Cap.) $\sim$ Size



Smart Dust Sensor Mote

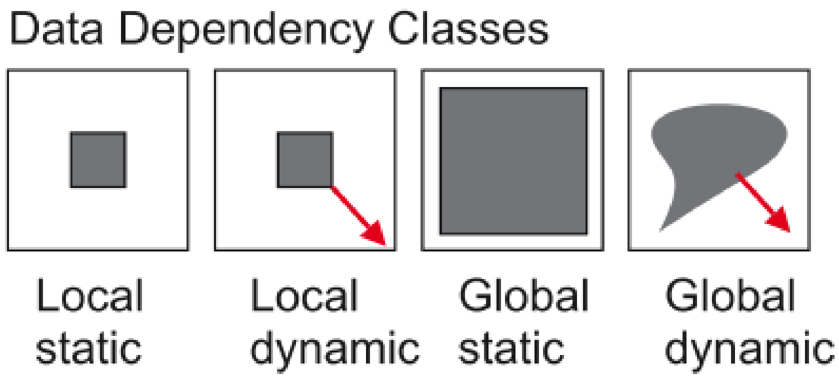


[Left: Warneke et al., Smart dust: Communicating with a cubic-millimetre computer, Computer, 2001]

A. Algorithmic Scaling

» Common algorithms used in Sensor Networks and sensing applications are characterized by their

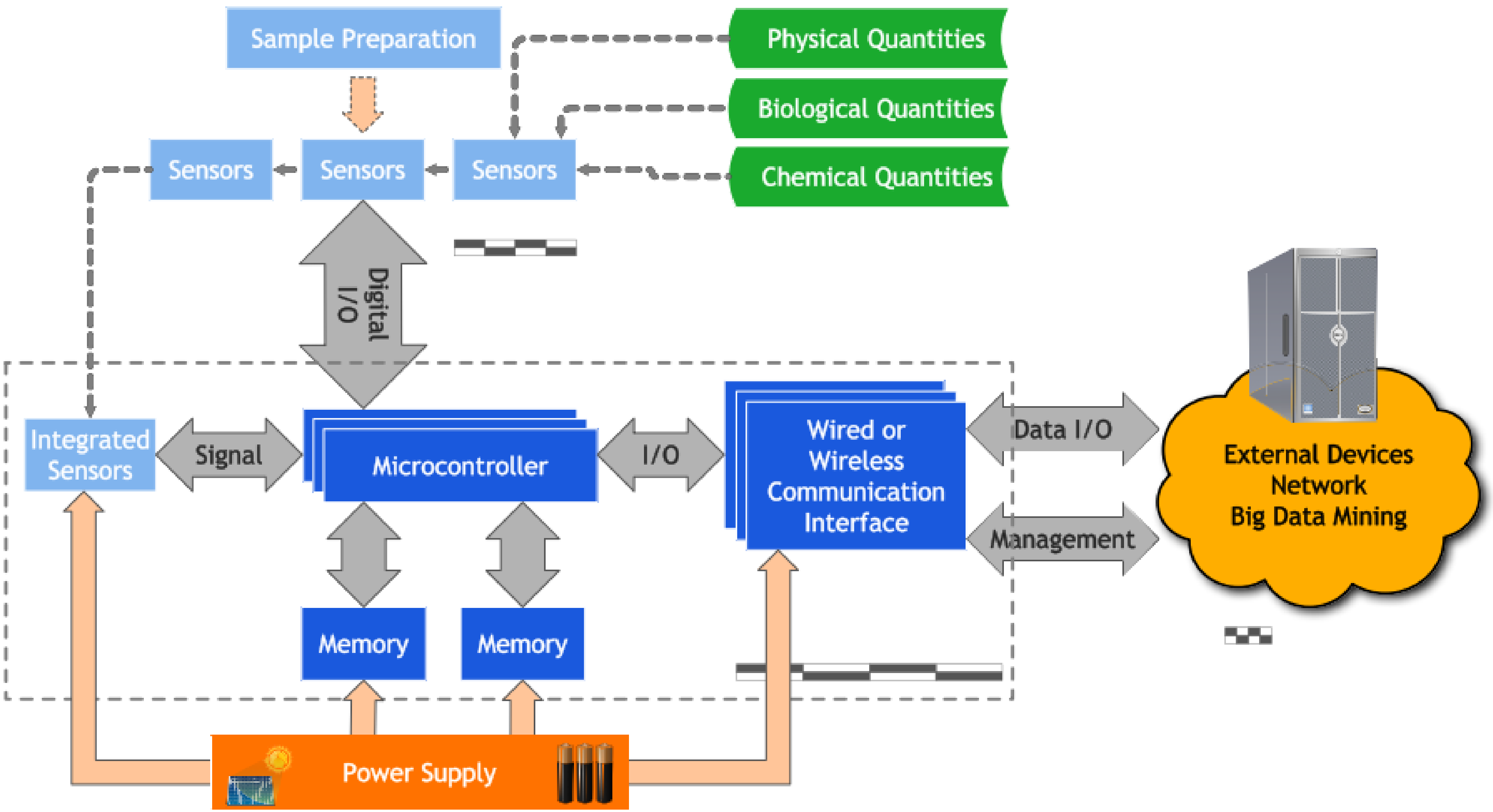
  » high data dependency,
  » high algorithmic complexity, and
  » restricted distribution capabilities.



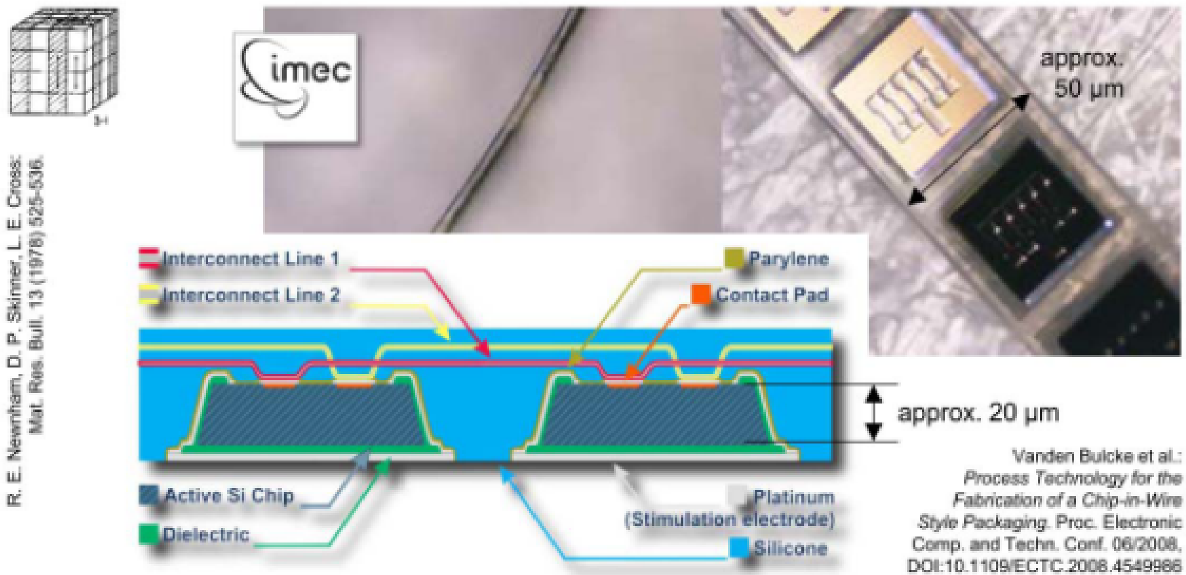
Integration of computing in technical structures or devices equipped with embedded systems requires:

Down-scaling of algorithms and methodologies towards **distributed processing networks** with **low-resource** platforms.

A. Smart Sensor Node

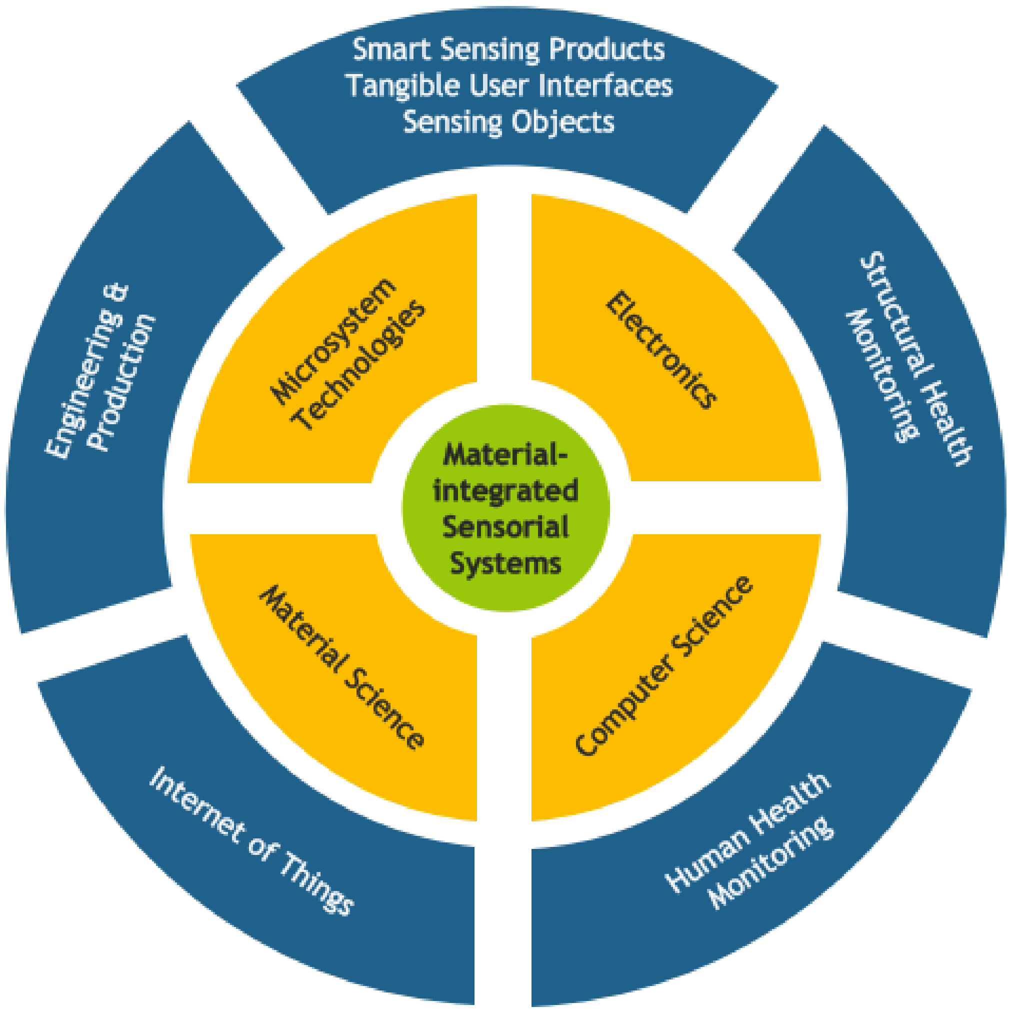


B. The interdisciplinary Approach: The Design of Sensorial Materials



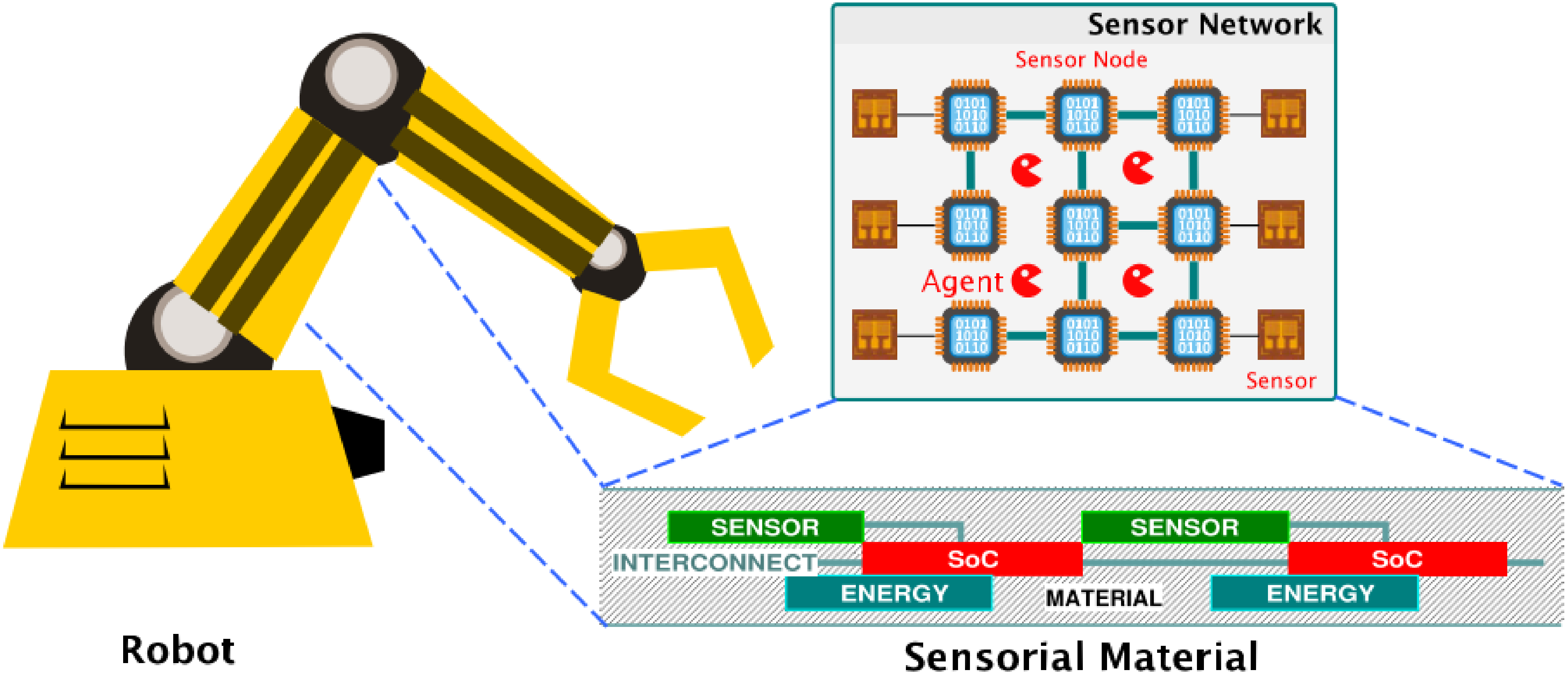
B. Material-integrated Sensorial Systems

- Material-integrated Sensing
- Material-integrated Sensor Networks
- Interdisciplinary Design Approach
- Broad diversity of Applications
- Improvement of Quality-of-Service
- Improvement and Extension of Capabilities
- Improvement of Robustness



B. Sensorial Materials

» Sensorial Materials are material-integrated Sensor Networks embedded in technical structures

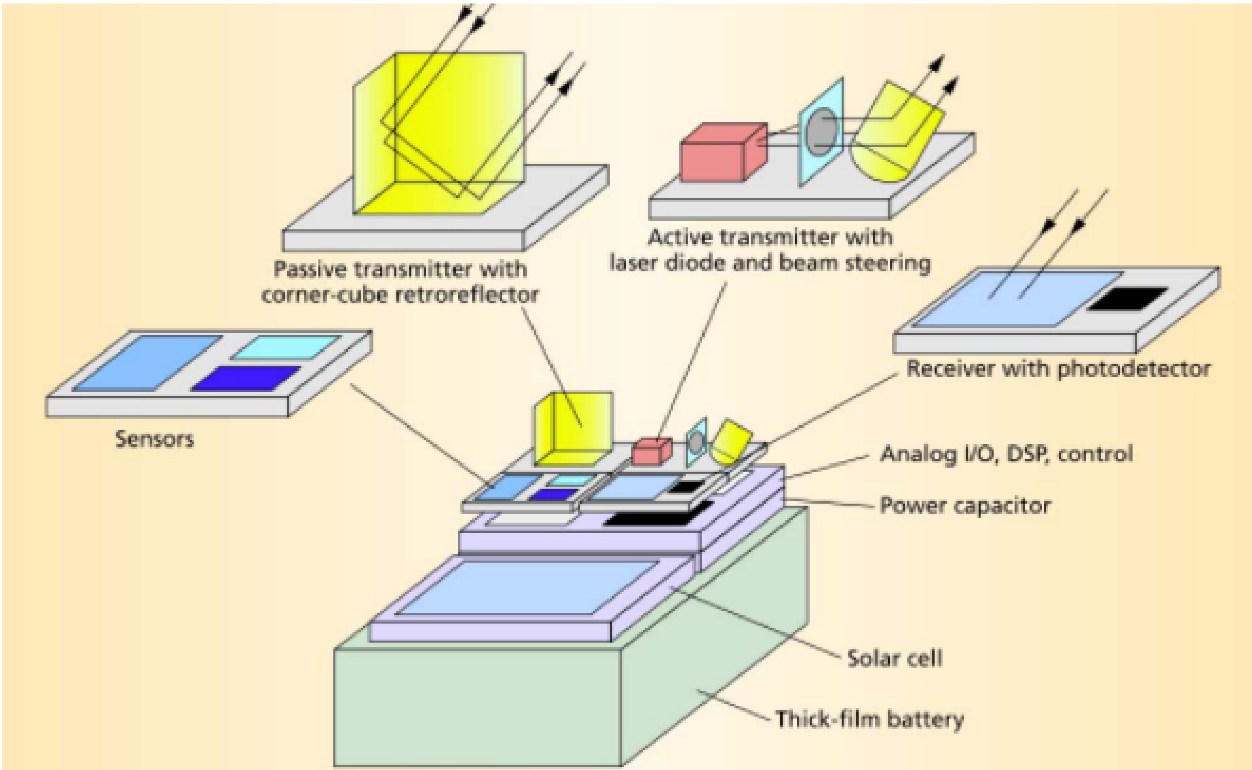


[Bosse, DOI:10.1109/JSEN.2014.2301938, 2014]

B. Sensorial Materials: Smart Dust

Autonomous Sensing, Computing, and Communication System Unit

» New Microsystem and Microfabrication technologies enable the integration of Sensors, AD Conversion, Digital Signal Processing, Energy Harvesting, and Communication in the mm^3 scale!

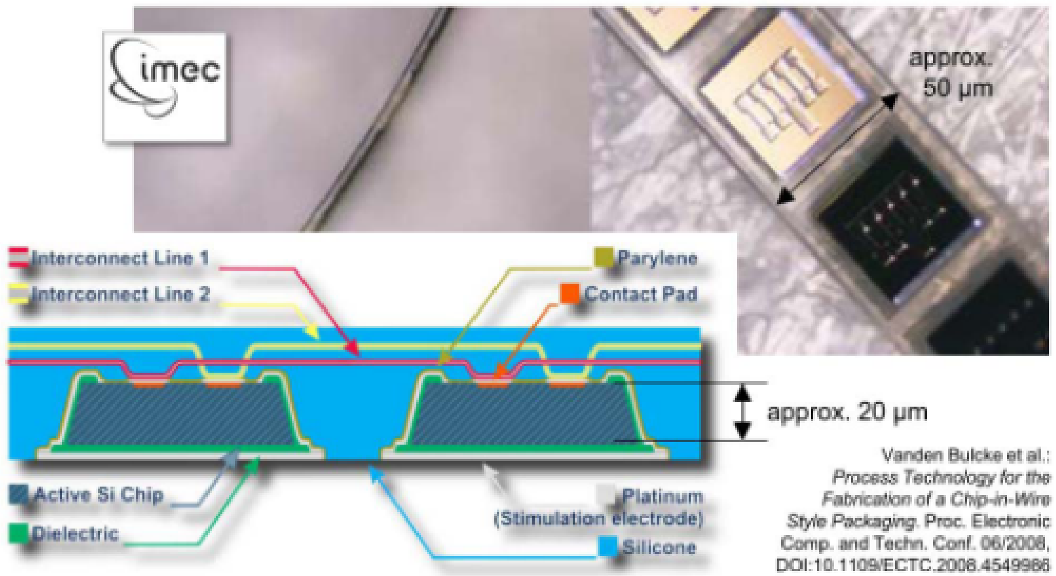


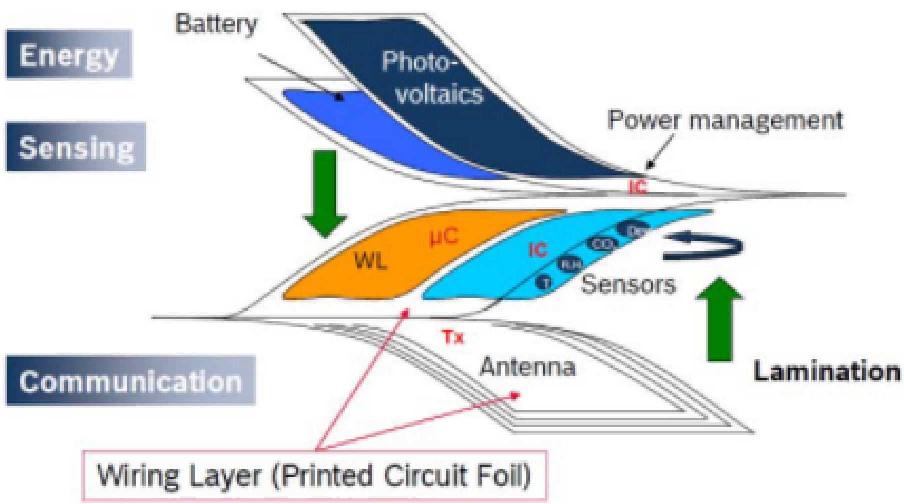
[Warneke et al., Smart dust: Communicating with a cubic-millimetre computer, Computer, 2001]

B. Sensorial Materials: Enabling Technologies

» Chip-on-Wire and Foil-to-Foil processes are promising enabling technologies for Smart Sensorial Materials

» One- or two dimensional network topologies

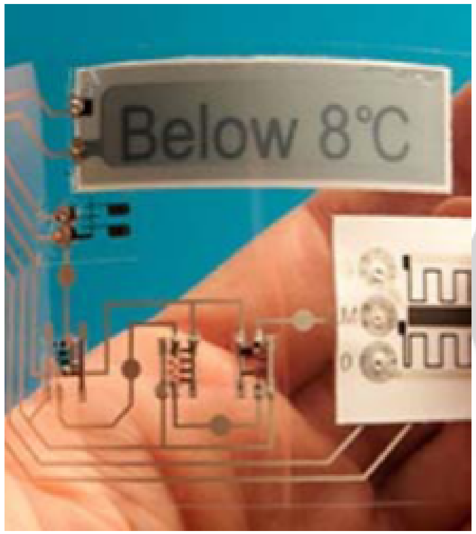


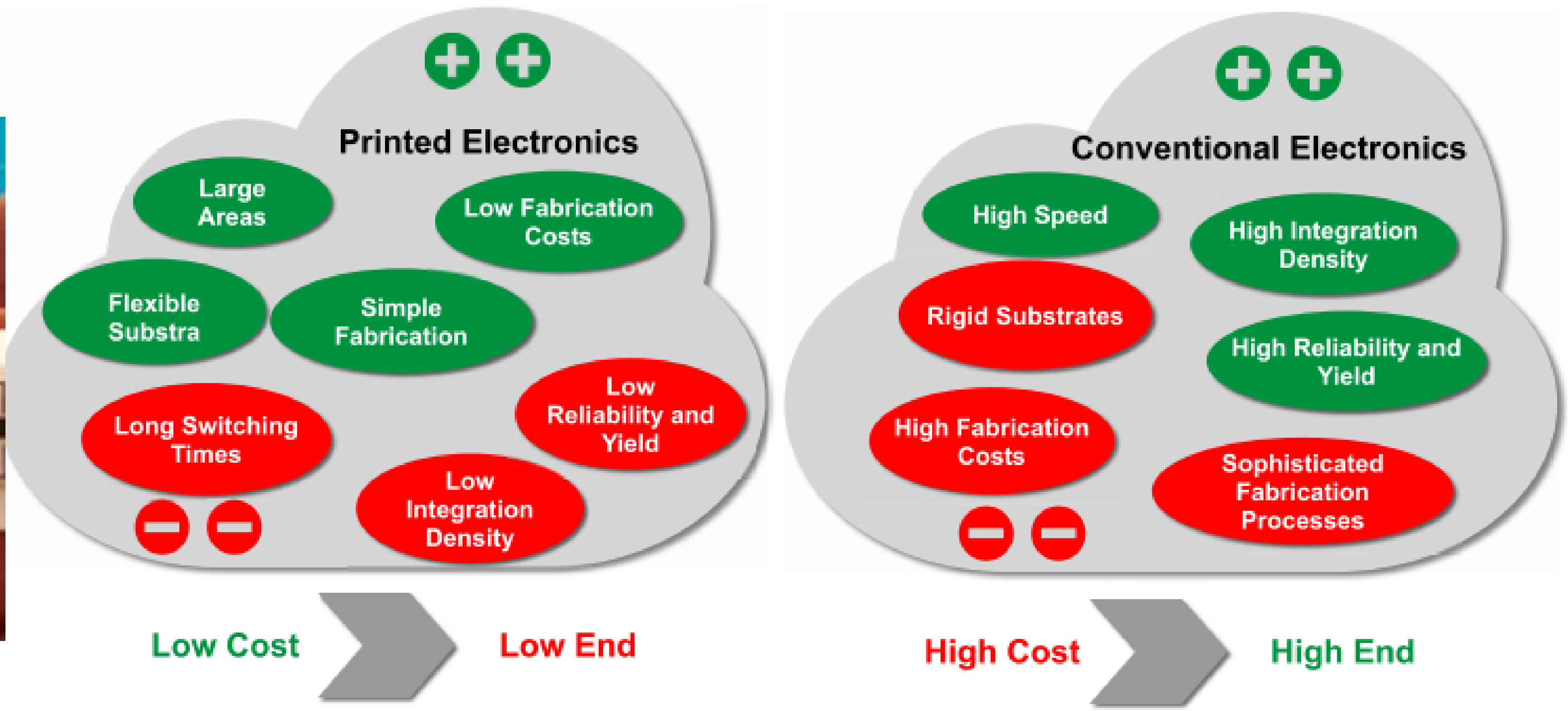


[Left: IMEC, Right: www.project-interflex.eu]

B. Sensorial Materials: Enabling Technologies (cont.)

- Print technologies can be used to print electronics, sensors, energy storage, displays, and interconnect structures on free form shapes!
- But comparison of printed and conventional semiconductor processes shows currently still significant differences





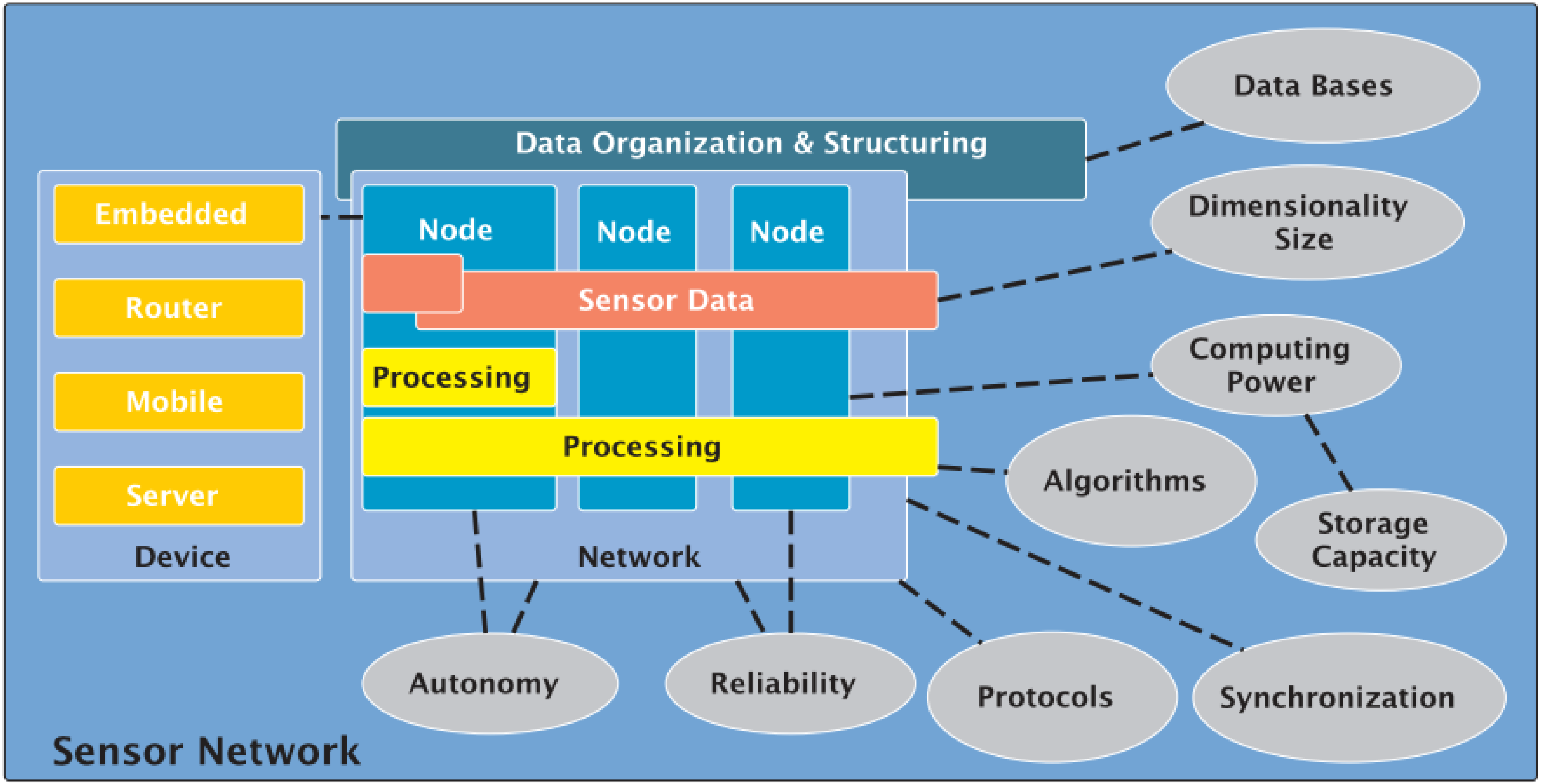
[Left: http://storify.com/GotJenn/printed-electronics-1]

C. Smart Sensor and Mobile Networks

C. Sensor Networks: Terminology and Features

Nodes, Devices, Processing, Data, Communication, Messaging, Storage

» Sensor Networks are composed of interacting autonomous nodes



C. Sensor Networks and Integration: Challenges

» Power Supply and Power Sources

» Communication Technologies

» Dynamic Network Structures

» Autonomous Node and Network Behaviour- Self-Organization

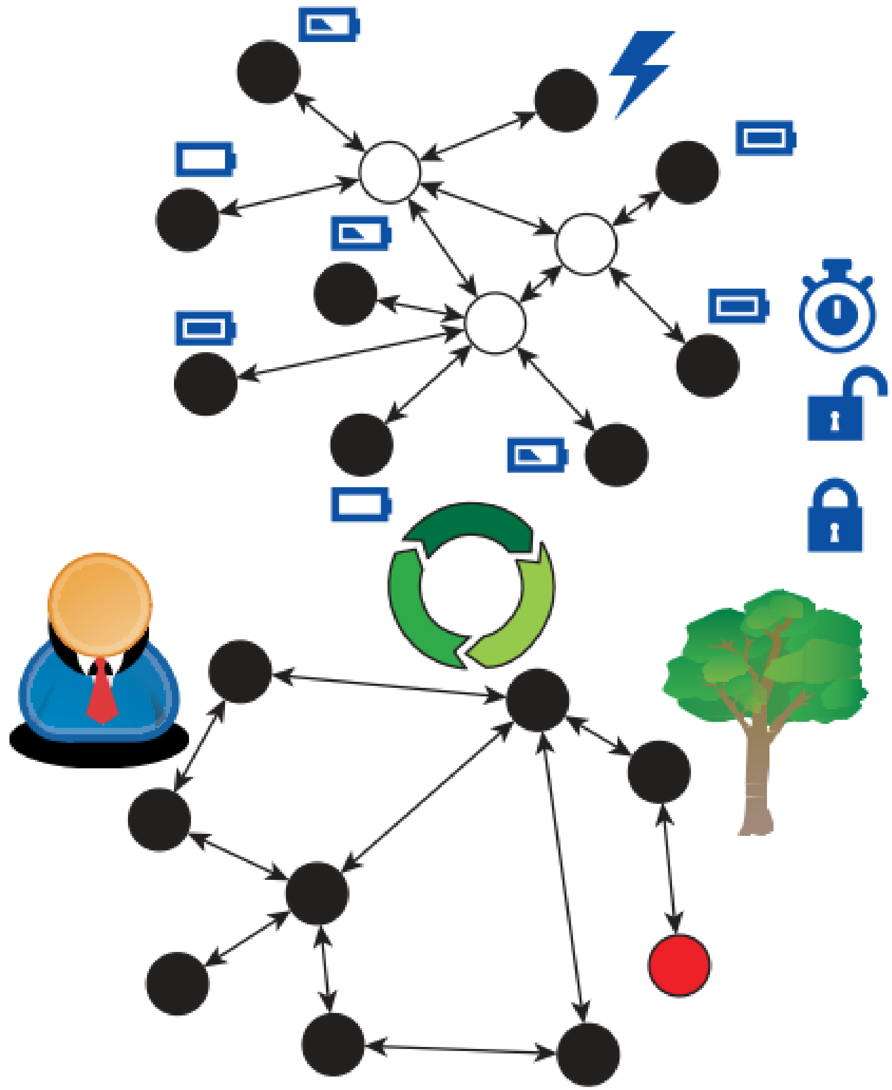
» Real-time capabilities

» Reliability: Dealing with Failure/Loss {Sensor, Node, Link, Msg., Network}

» Security, Privacy, Data Ownership

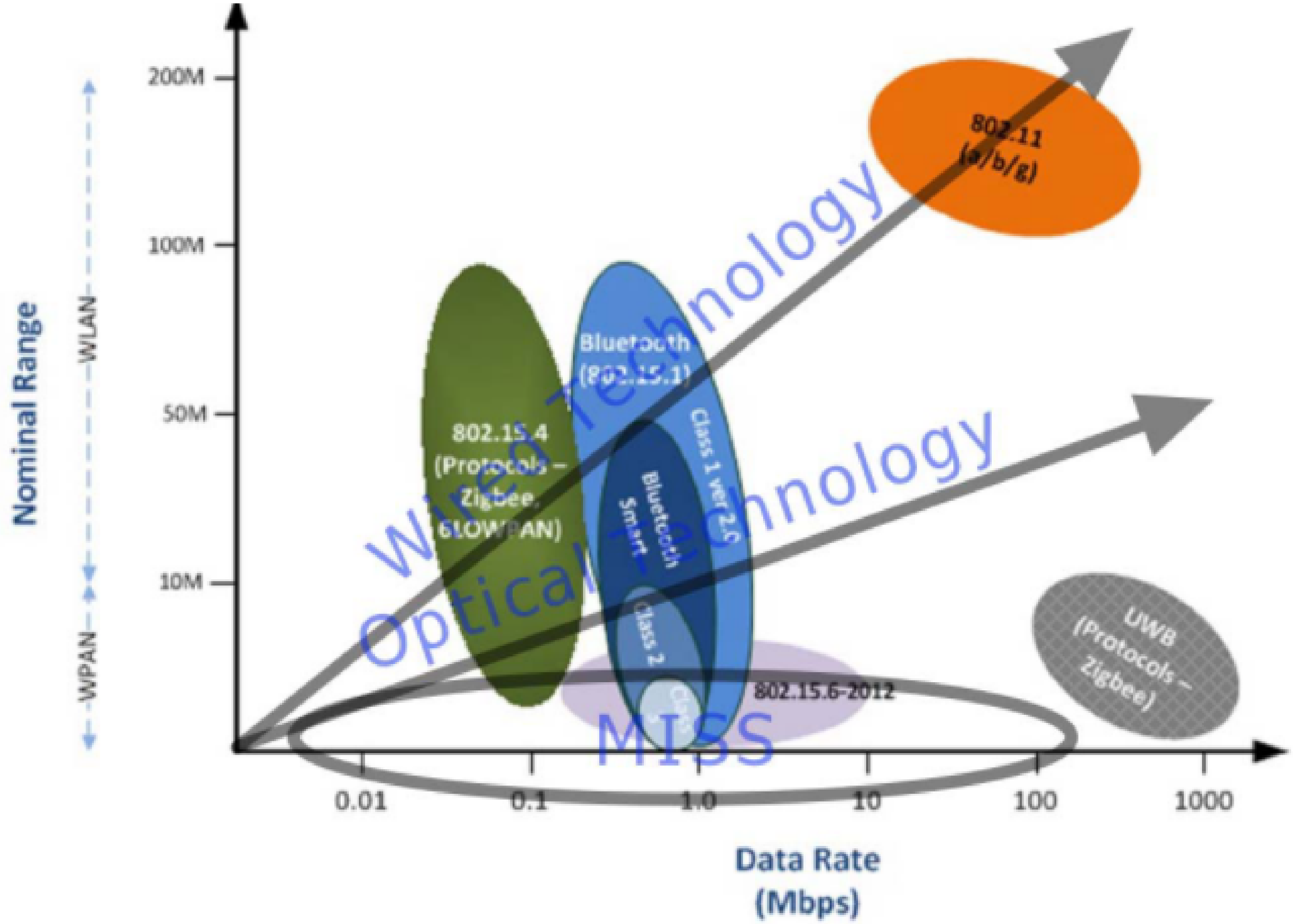
» Durability - Long term operation

» Environmental and Bio-Human Compatibility!



C. Sensor Networks and Communication Technologies

- » Scale offered by Wireless Communication Technologies: Meter Range
- » Scale required by Material-integrated Systems: Millimetre Range



[McGrath et al., ST, 2014]

C. Mobile Networks

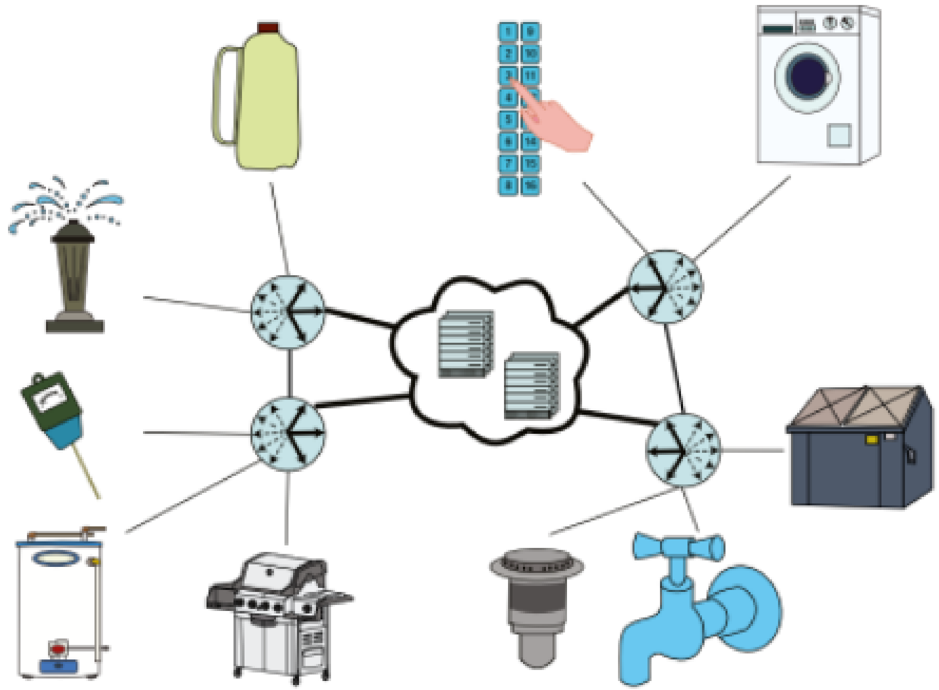
» Sensor Clouds, Pervasive and Ubiquitous Computing pose a shift from static network to dynamic mobile computing

» Key Features:
  - » Adaptability
  - » Ad-hoc Connectivity
  - » Hardware Abstraction Layer (HAL)

Constraints of Mobile Computing Environments:

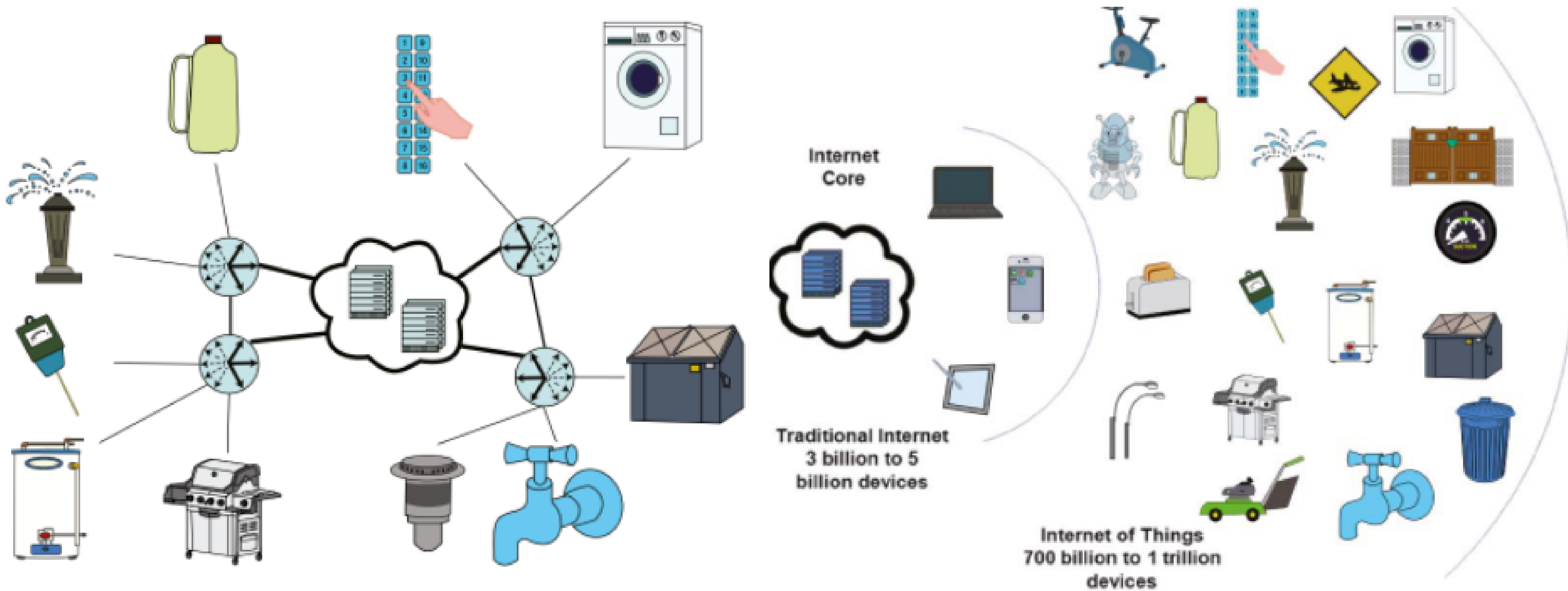
Resources	Mobile devices are more resource-poor than their static counterparts
Reliability	Mobile devices are less secure and reliable
Connectivity	Mobile connectivity can be highly variable (ad-hoc, bandwidth, latency, reliability)
Identification	Mobile devices support ad-hoc connectivity with random identities

D. The Internet of Things and Sensor Networks



D. The Internet-of-Things

- The Internet-of-Things connects Sensor/Actor Networks to the Internet
- Distributed Service, Computing, and Storage Architecture ➠ Clouds
- Collection of Things: Consumer Goods, Industrial Devices, Machines, Sensors, Sensorial Materials, Smart Dust, ..

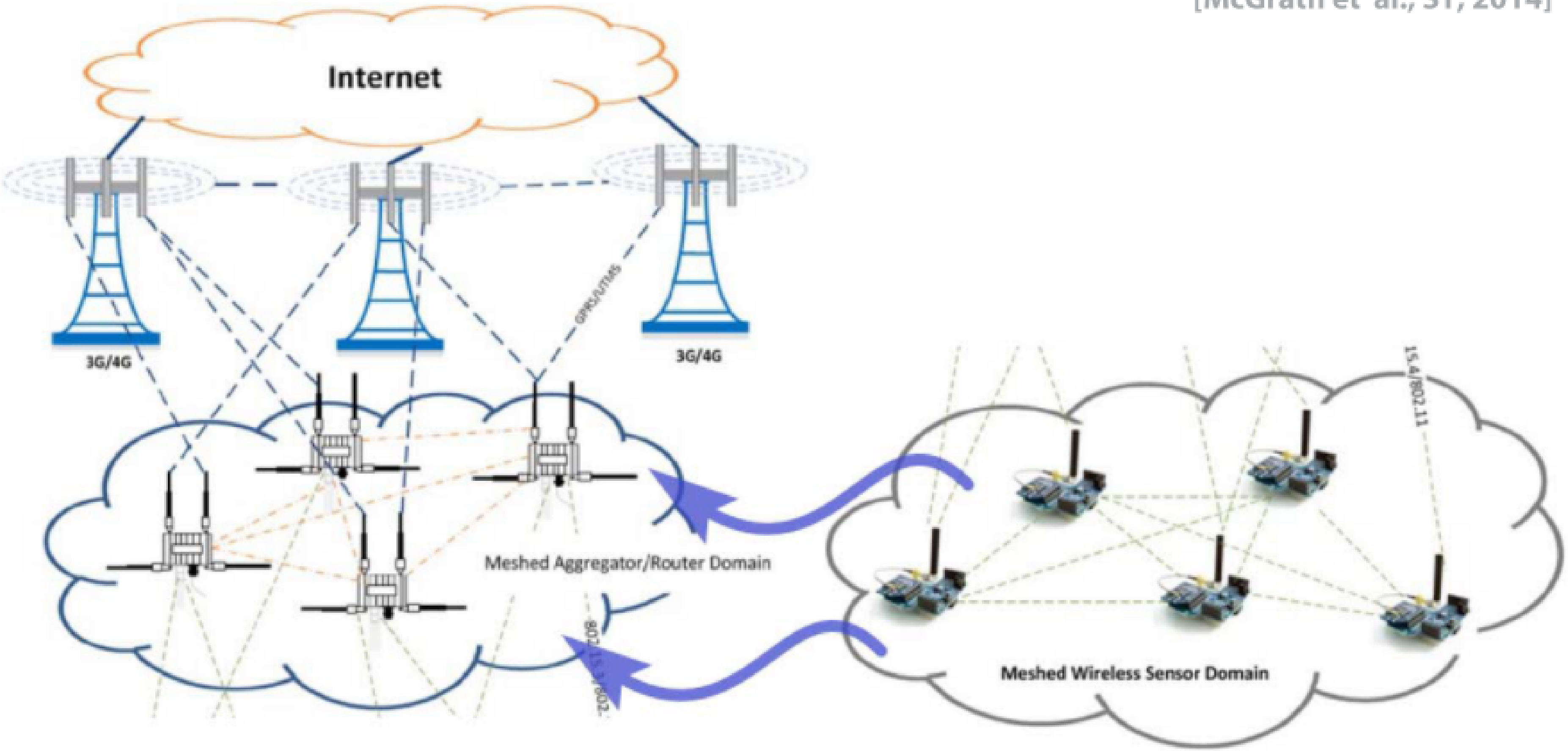


[DaCosta RtIoT, 2013]

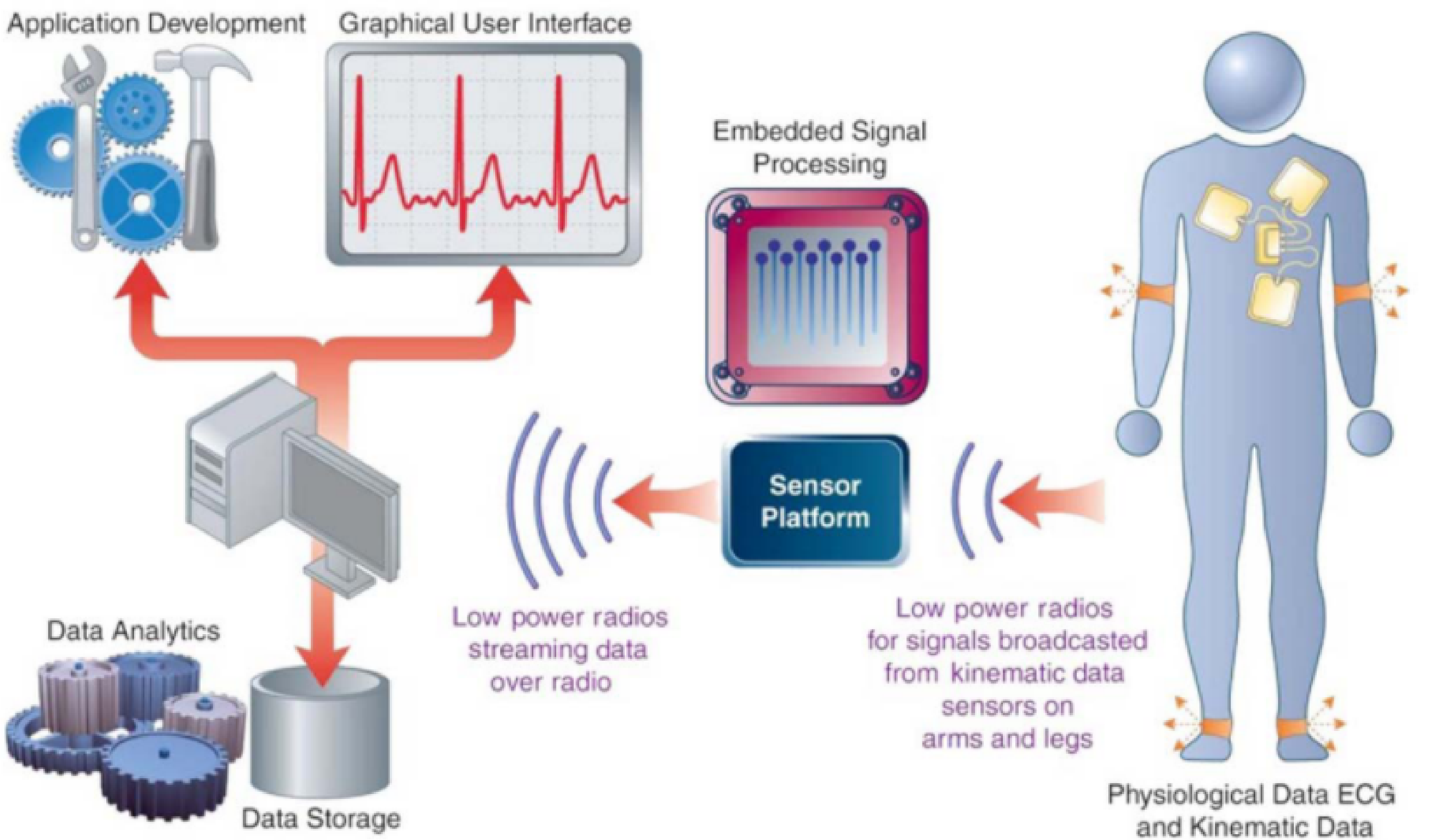
D. Wide Area Wireless Sensor Networks

» Large Scale Monitoring of bridges and buildings, Load/Health Monitoring of large structures (Wings), Earthquake Monitoring

[McGrath et al., ST, 2014]

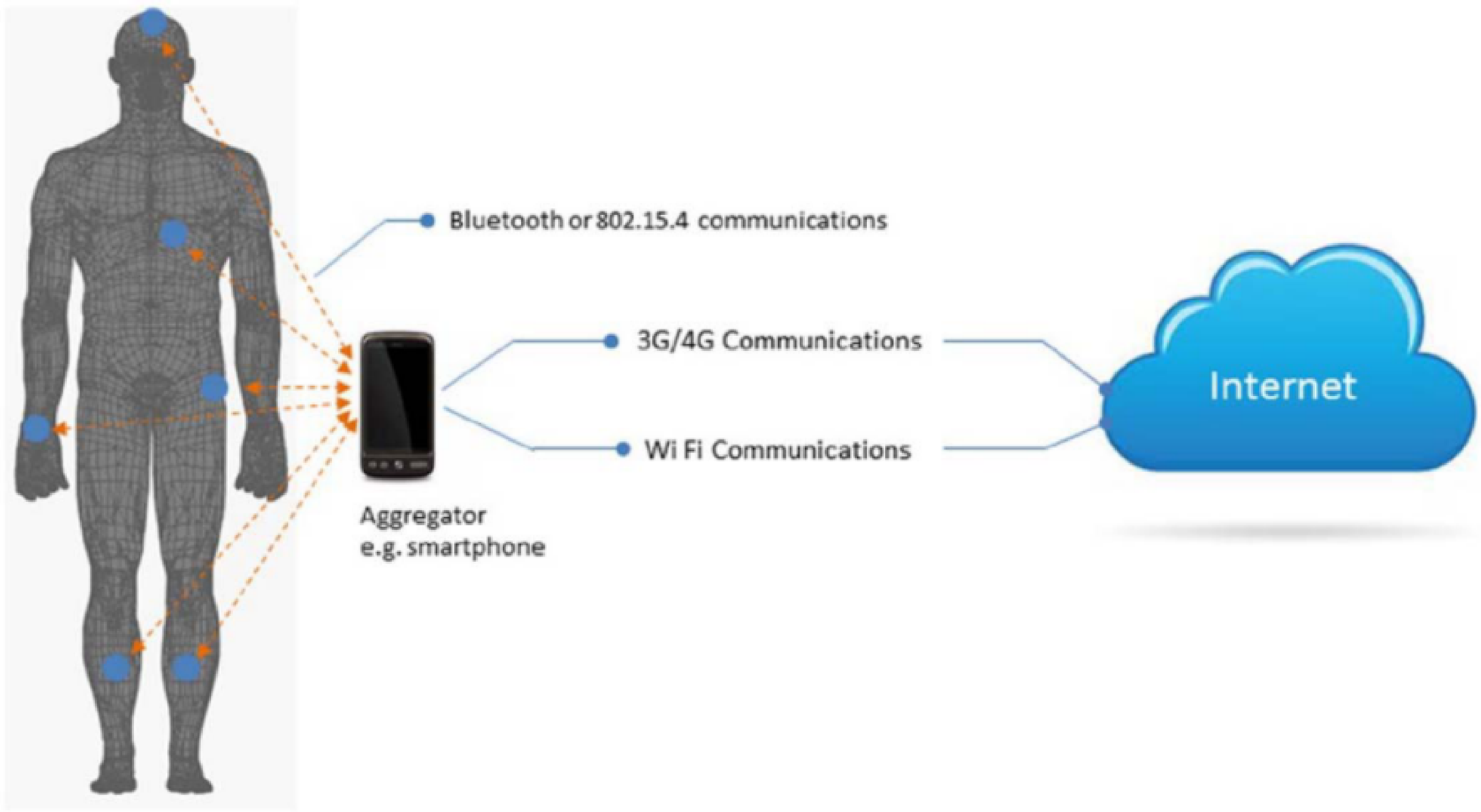


D. Smart Health Application-Human Body Sensor Networks



[McGrath et al., ST, 2014]

D. Wireless Body Area Network Application



[McGrath et al., ST, 2014]

E. Distributed Computing, Big Data, and Clouds

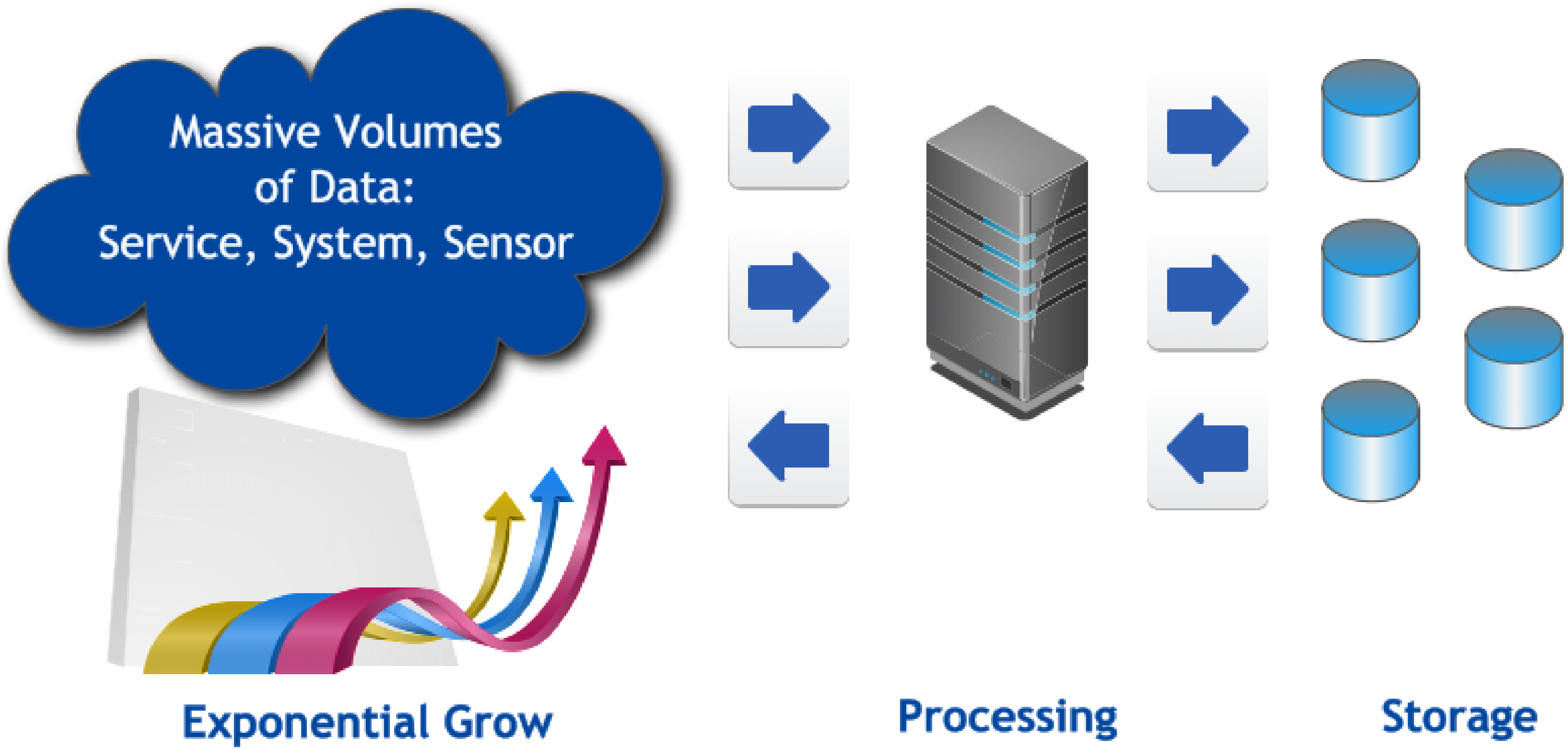
E. Real-time Big Data

- Big (Sensor) Data:
  - Large amount of data which is stored distributed
  - Data Mining is used to extract condensed information from data
  - Database model is used for structuring, storing, and retrieving data
- Streaming Big (Sensor) Data
  - Input: Sequential stream of data
  - Output: Processing of input stream results in an output stream giving (real-time) answers on input data
  - Stream-based Processing = continuously analyse and process massive data volumes
- Real-time Big (Sensor) Data Processing
  - React in real-time = in a bounded time interval to each and every new record

E. Big Data Processing - Traditional

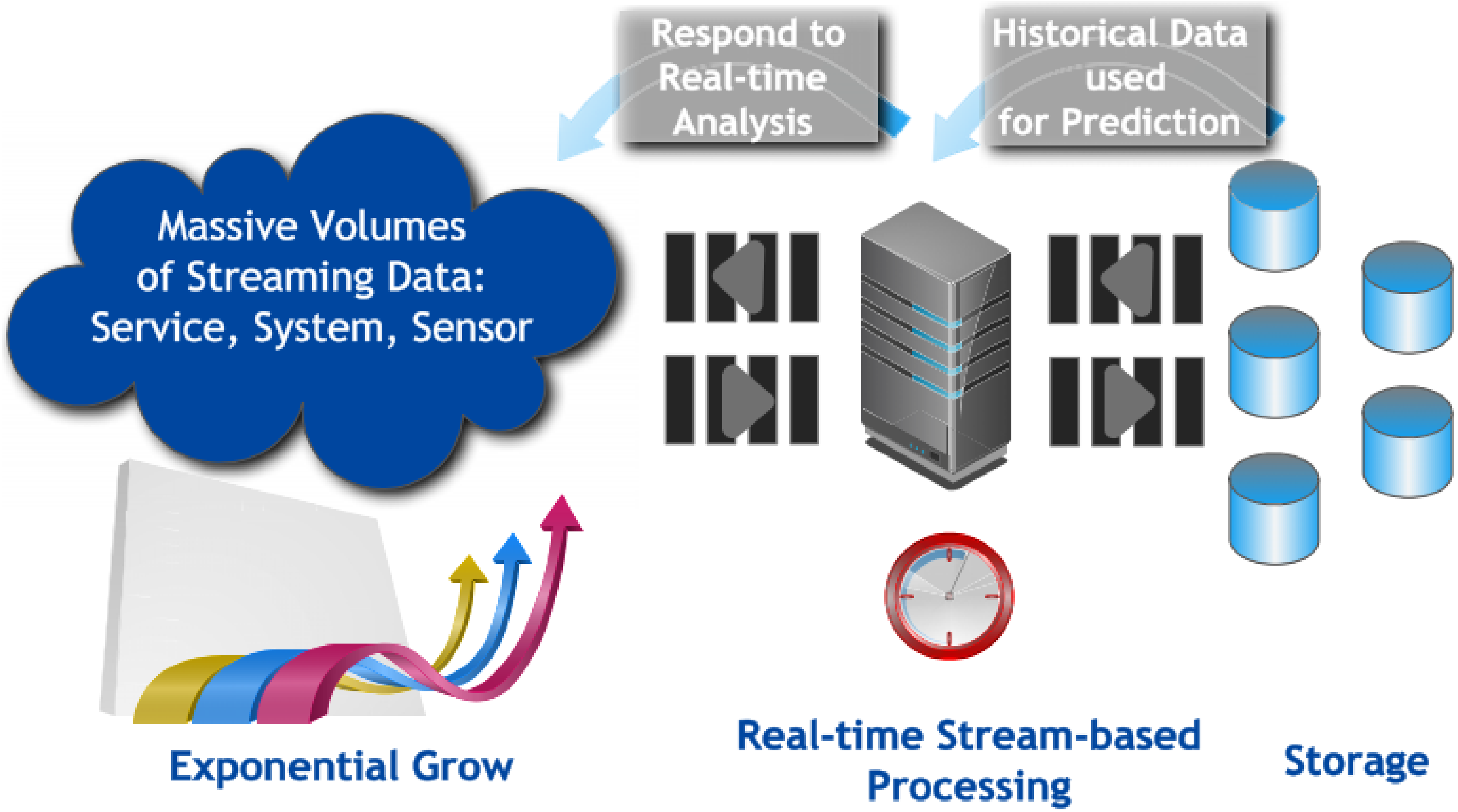
Traditional Process and Store architectures

- » Have poorly integrated operational platforms;
- » Perform semi-distributed Processing with Service-orientated Archit. (SOA).



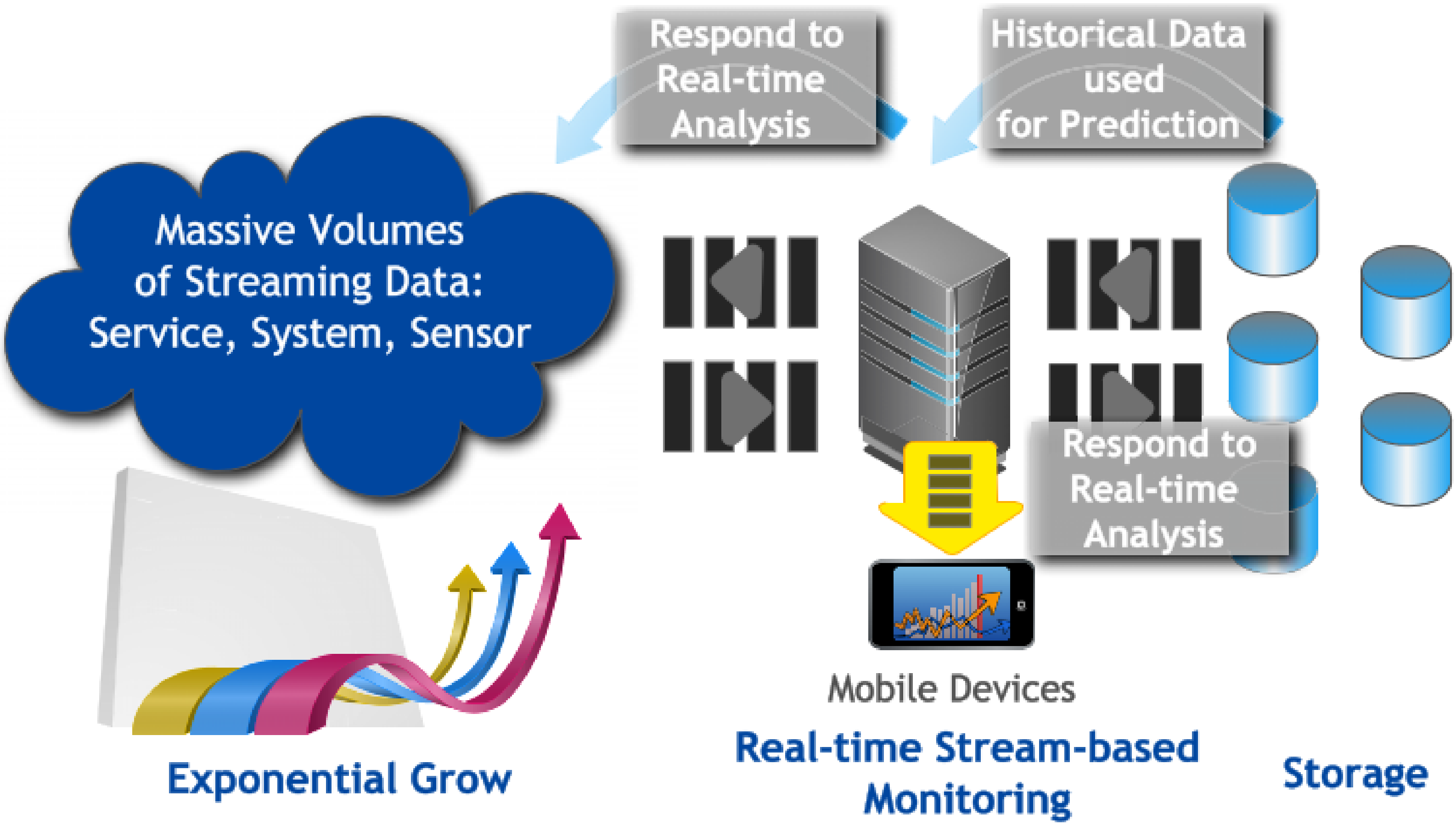
E. Big Data Processing - Real-time Data Streaming

Streaming of massive volumes of data in real-time with real-time Analysis



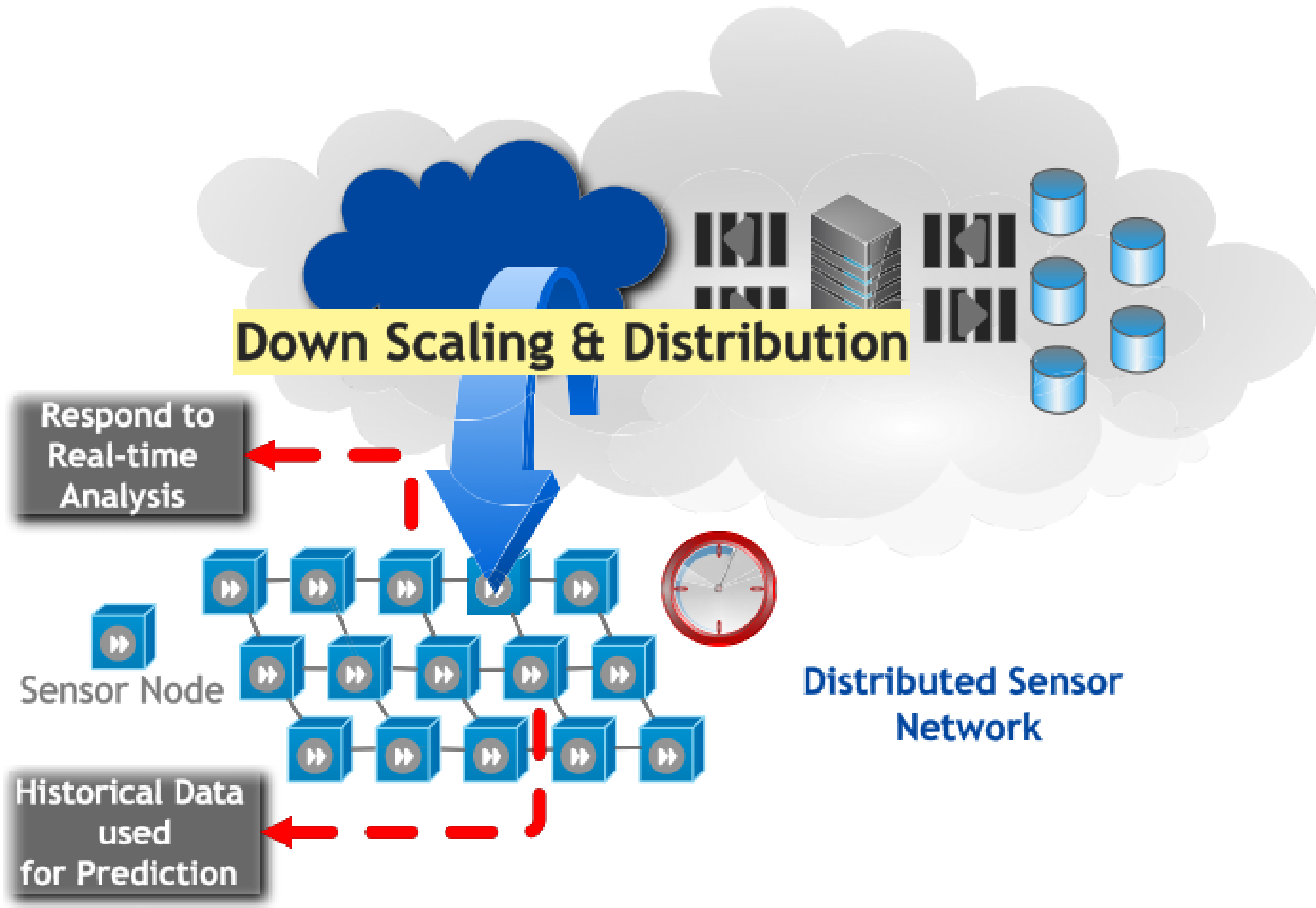
E. Big Data Processing - Monitoring

Monitoring: Real-time alerts and visibility with continuously updated streaming results



E. Integration of Sensor Networks in Big Data Processing?

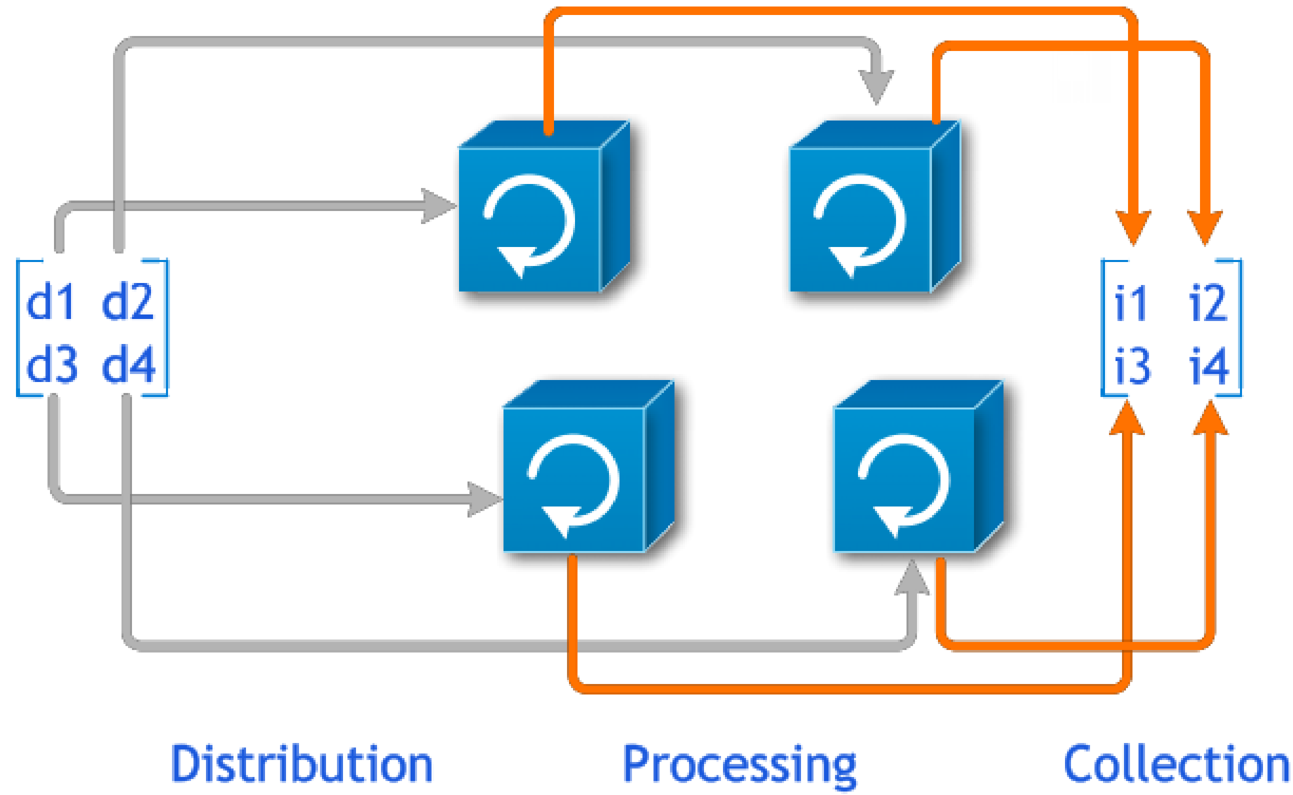
Required: Down-scaling of algorithms and massive distributed computing

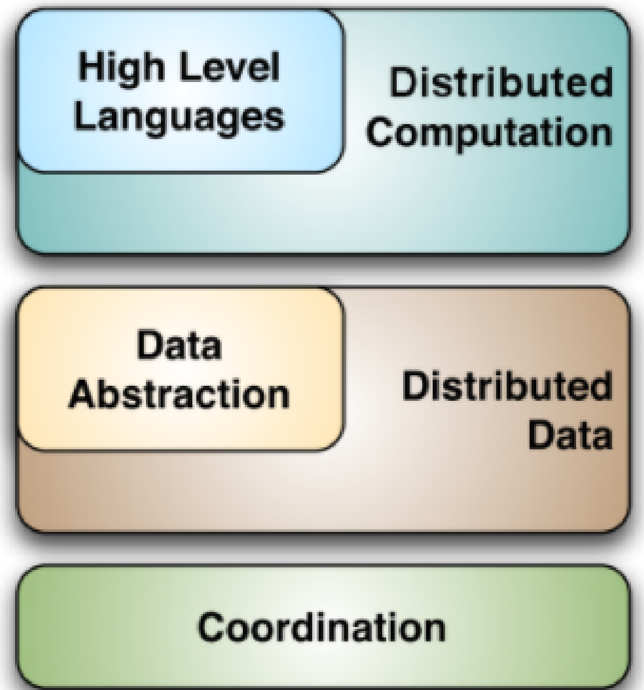


E. Distributed Computing in Computer Networks

» Distributed computing in computer networks requires:

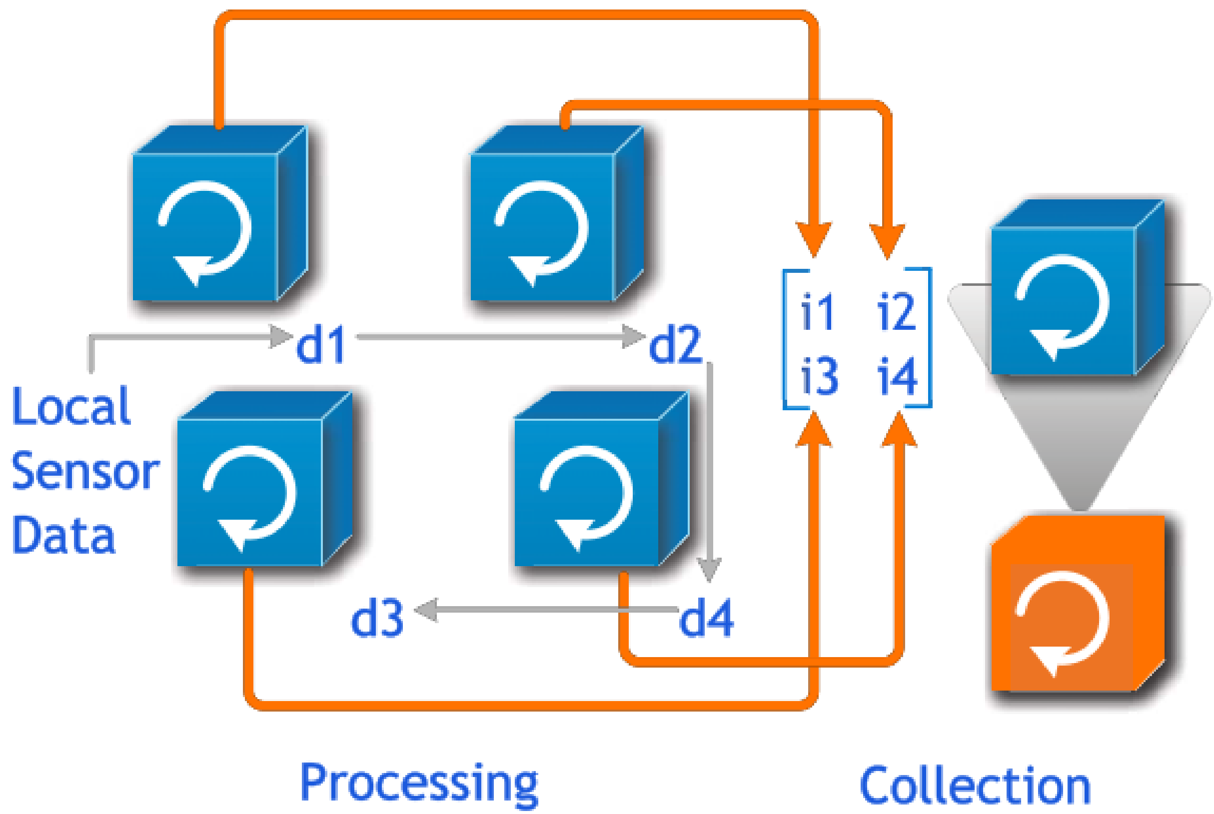
Input Data Distribution → Synchronization → Data Exchange → Processing → Output Data (Information) Collection

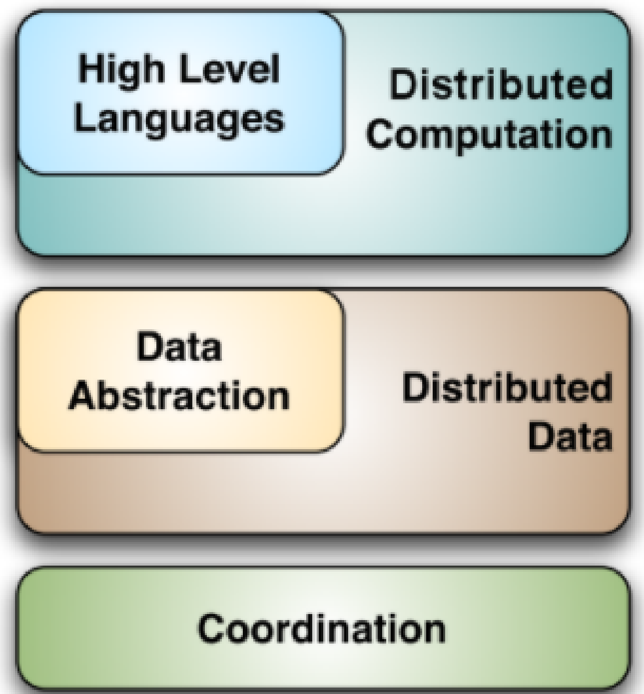




E. Distributed Computing in Sensor Networks

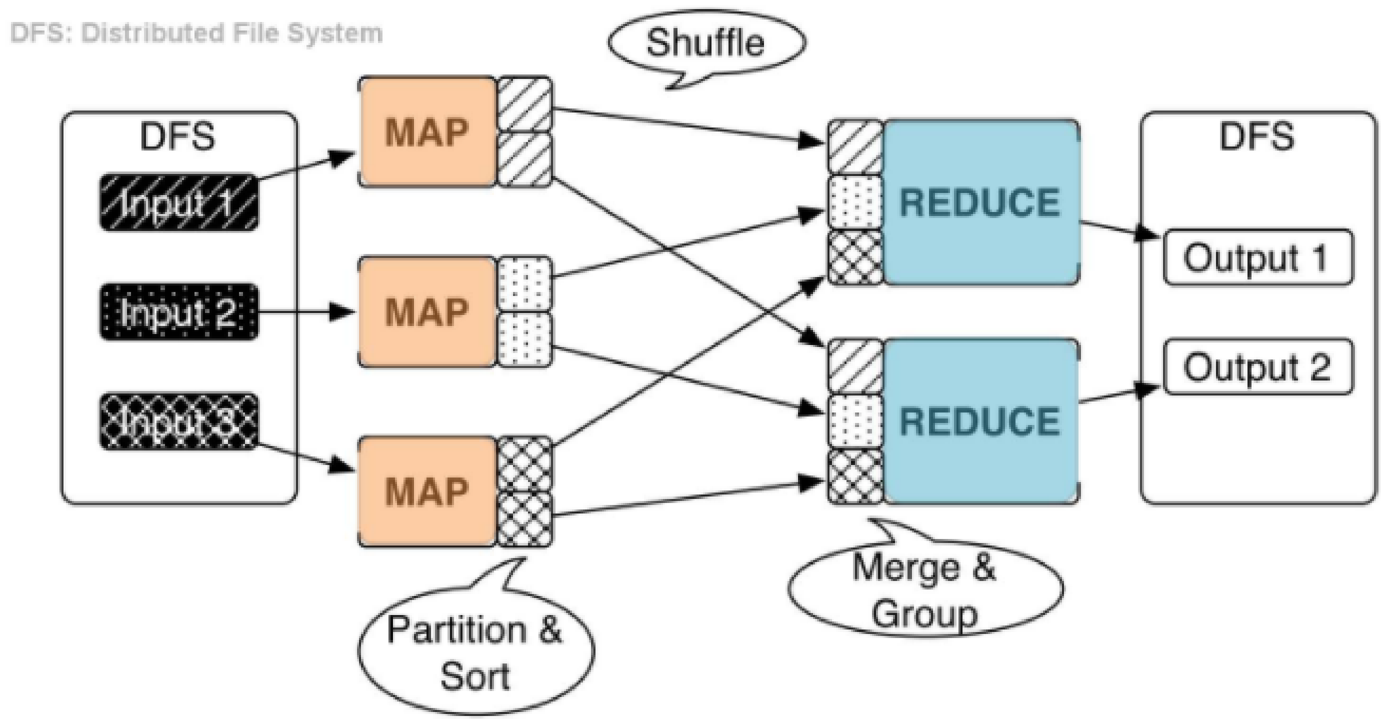
- Sensor Networks generate the input data locally at node level
  - Commonly negotiation between data sources and sinks required
  - Output data must be distributed in the networks based on request





E. Distributed Computing: Divide & Conquer

- Large (distributed) problems can be solved by Divide & Conquer
  - Divide & Conquer can be implemented with Self-organizing systems using basic cells
  - Map & Reduce are common algorithms in Big Data Computing & Analysis

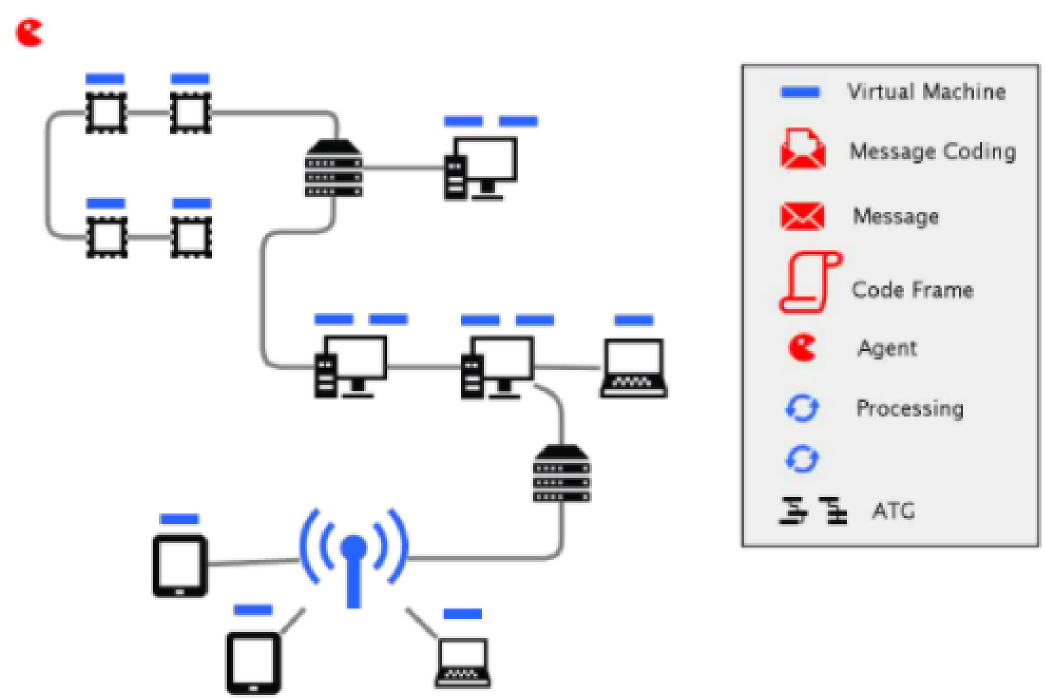


[B. Morales, BDW, 2012]

E. Distributed Computing: Mobile Code

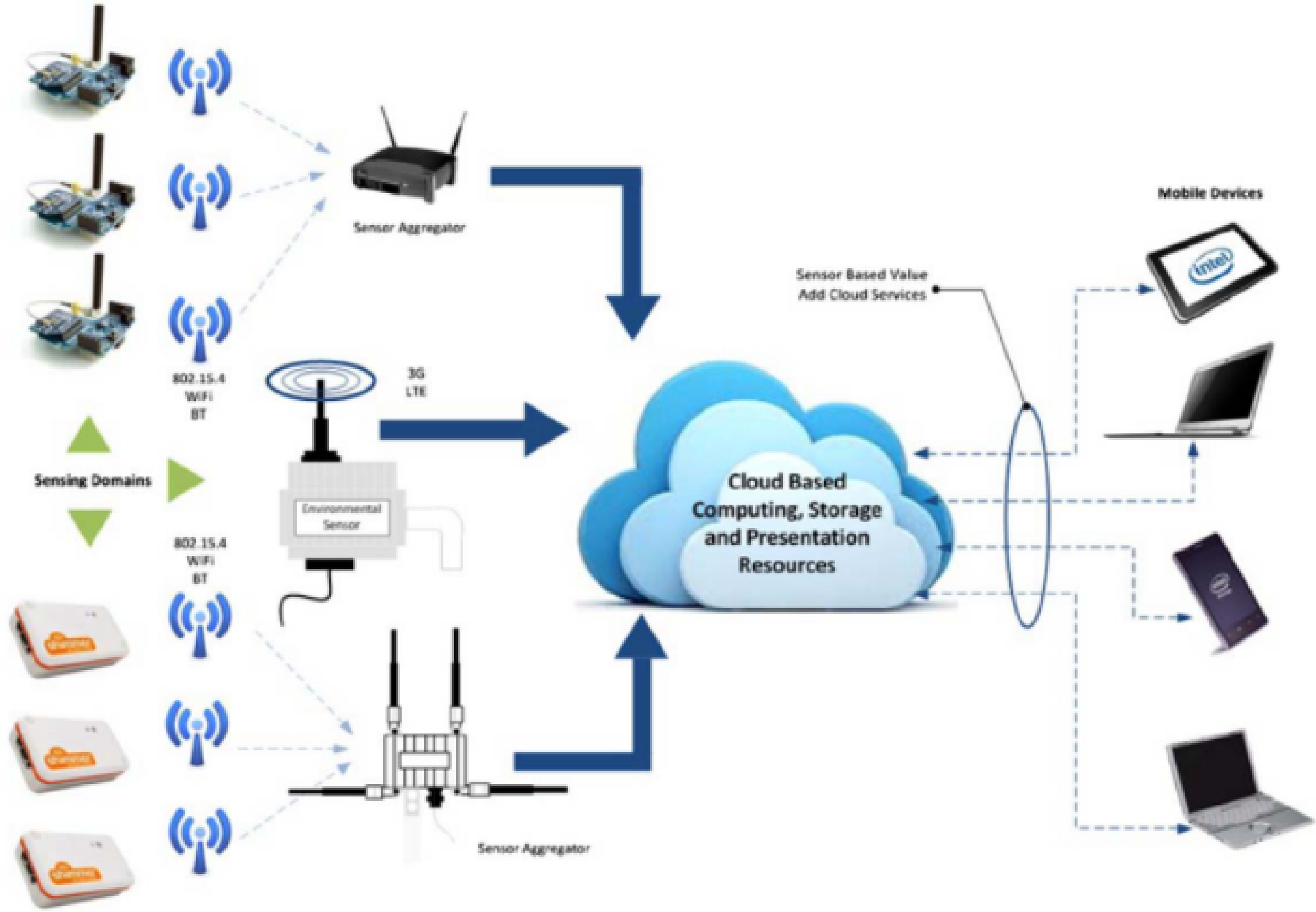
» Traditional distributed computing uses located processes and mobile data (message passing)

» Required paradigm shift: From mobile data to mobile code (processes)



E. Distributed Sensor Cloud Computing

- Big Data Computing
- Data Cloud
- Sensor Cloud
- Internet-of-Things
- Sensor Netw.
- MISS



[McGrath et al., ST, 2014]

F. Multi-Agent Systems and Mobile Agents

The Solution?

F. Data Processing with Mobile Agents - The Solution?

» **What are they?**

» Agents are a programming and communication model for distributed & parallel computing.

» An agent is an autonomous data processing unit, which is

  » Situated in its environment;

  » *Traditionally* a program executed on a computer system.

» An agent interact with its environment:

  » Hardware agents (robots) can change the physical world

  » Software agents (programs) can change the virtual world // Data

» A multi-agent system is a collection of loosely coupled autonomous agents migrating through the network and interacting with each others.

F. Data Processing with Mobile Agents (cont.)

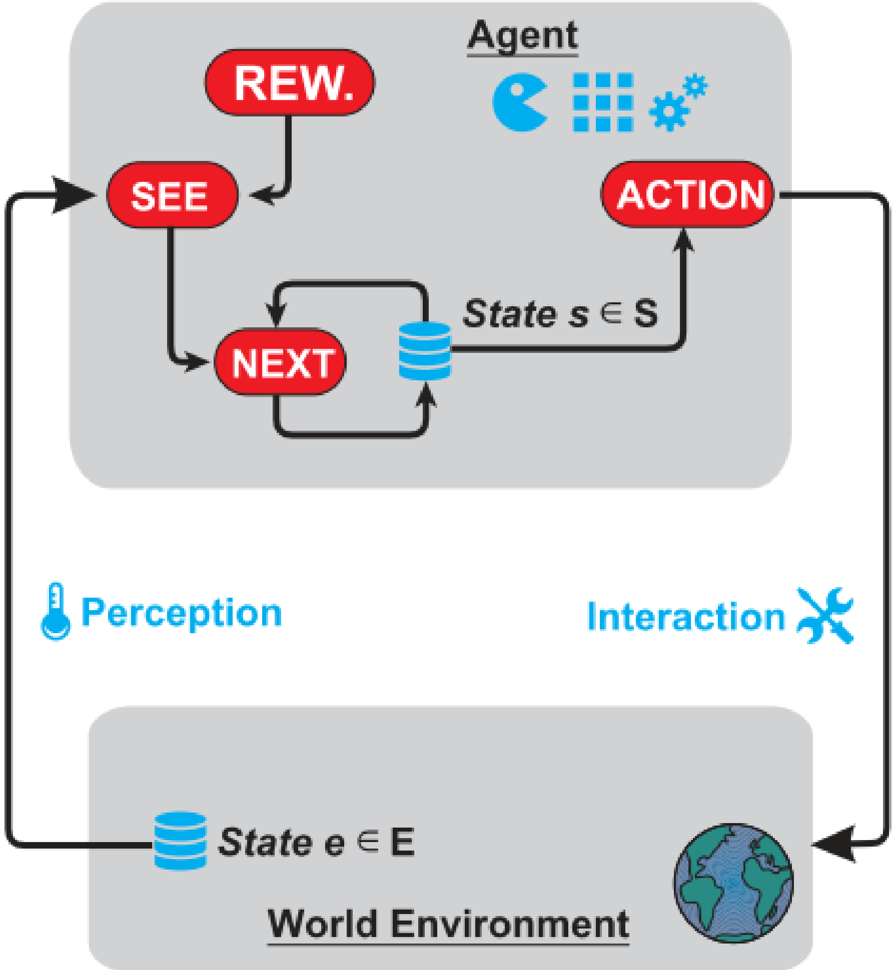
What can they do?

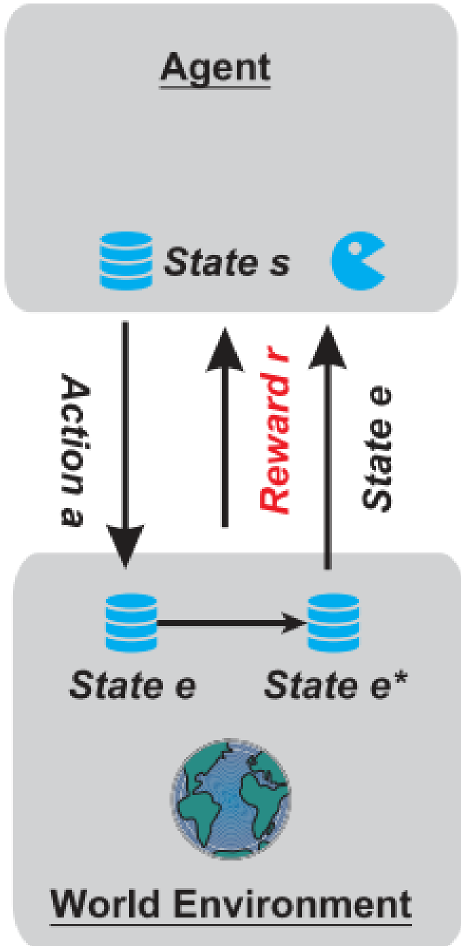
- Multi-agent Systems (MAS) can be used to implement
  - **autonomous**,
  - **reliable**, and
  - **adaptive** data processing in distributed networks.
- Agents can be basic cells for Self-organizing and Holonic Systems
- Agents can learn and can modify/adapt their behaviour
- Agents are independent of a specific host and network architecture!
- Agents can cross network barriers in strong heterogeneous networks

  Sensor Networks ↔ Ambient Networks ↔ Internet-of-Things ↔ Internet ↔ Clouds

F. The Agent Model: Reactive Behaviour

Perception ⇒ Processing ⇒ State Change ⇒ Decision ⇒ Action





F. Agent Processing

» *Traditionally* agents are software programs processed on generic computers by using a virtual machine executing agent activities.

  » Issues: *high computational demands, complex programming interfaces*.

» Solution: Agent-on-Chip and Virtual Machines

I. The agent behaviour and activities are *statically* implemented in hardware

II. The agent behaviour and activities are *dynamically* implemented with low-level VM code and the VM implemented in hardware.

  » Featuring: *Low resource platform* by using high-level SoC synthesis

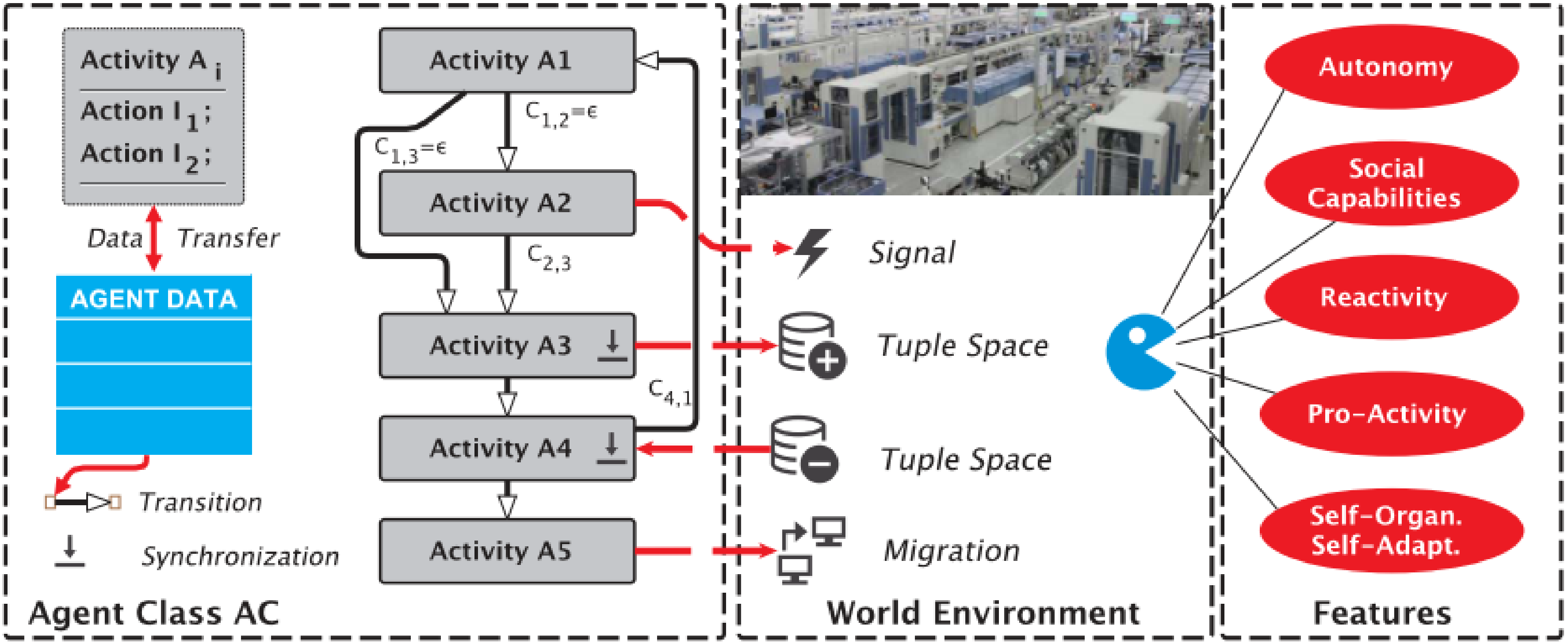
» Required: Agent Mobility

I. Migration of program code between nodes - code carries agent behaviour and state (control+data)

II. Transferring the agent state only - platform implements behaviour

Dynamic Activity-Transition Graphs

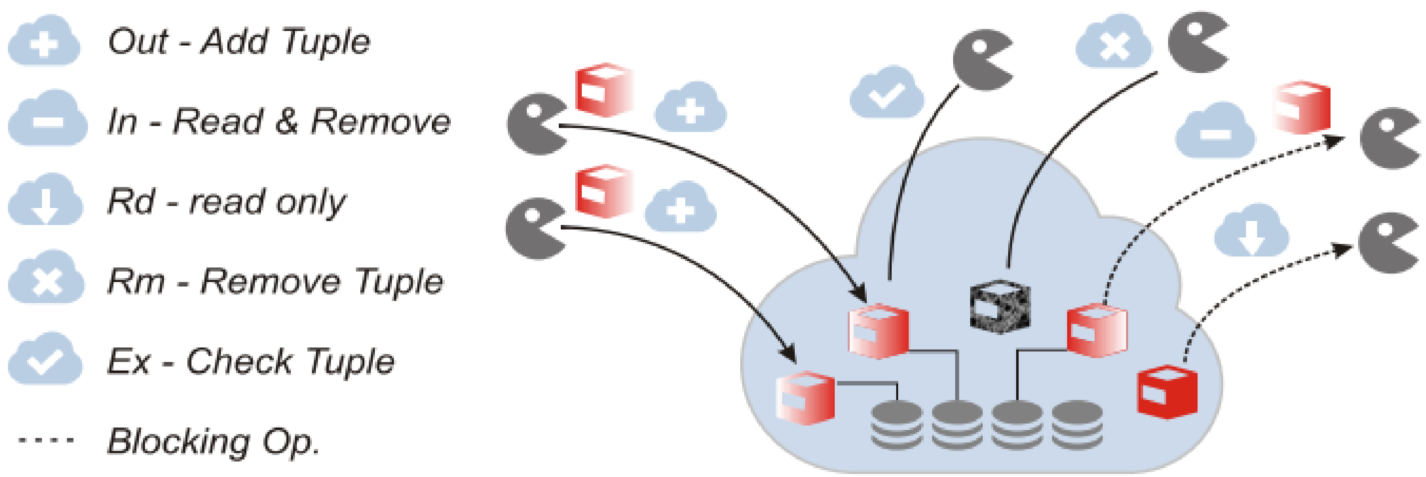
» Unified reactive-action programming model // Finite-State-Machine!
» The agent behaviour is defined by an Activity-Transition Graph (ATG).
» Programming model defines autonomy, interaction, and computing.



[S. Bosse, Industrial Agents and Distributed Agent-based Learning, ECSA 2016]

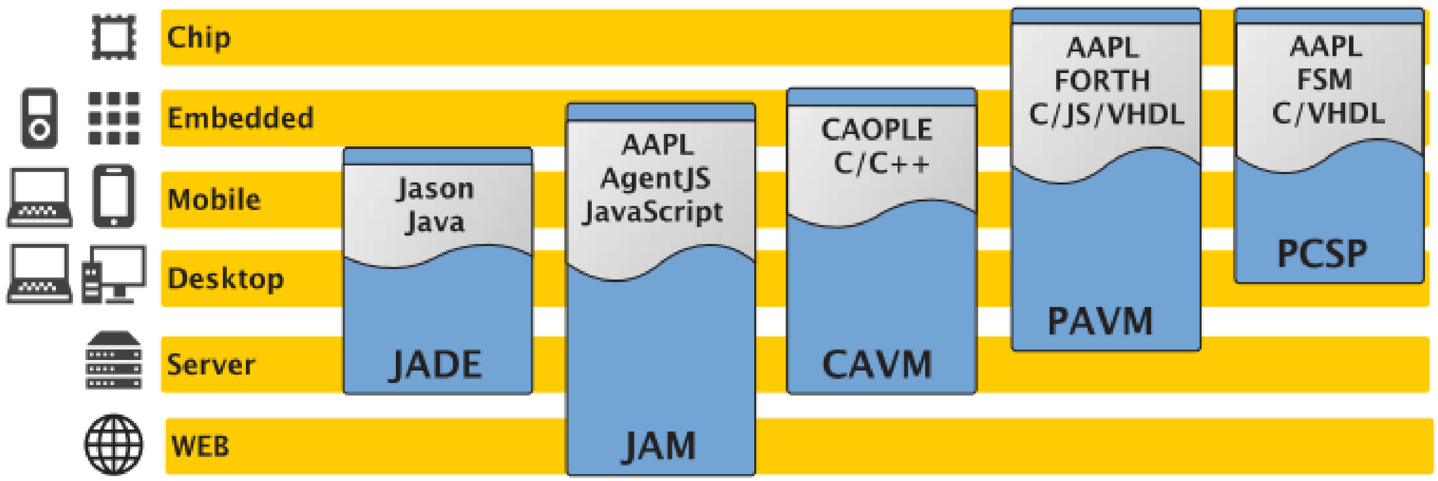
F. Agent Interaction

- Common agent communication languages and protocols: FIPA-ACL
  - Standard; but more complex semantic structures and protocols
- Tuple Spaces = Databases of n-ary Tuples
  - More generic and simpler to use with basic set of operations `{in,out,..}`
  - Virtualization of Storage / Resources
  - Synchronization and Pattern Matching
- Signals: Simple messages with single, multi- or broadcast behaviour



F. Agent Processing Platforms

- .. Enabling Technology for MAS deployment in Sensor Clouds ..
- Common: JAVA based frameworks, e.g., *JADE* [*Bellifemine et al.*], with Programming Language *Jason/AgentSpeak* [*Bordini et al.*]
  - Not very suitable for mobile and embedded host devices
  - Insufficient scaling capabilities, high resource requirements
- Support for different Host Platforms: *Embedded*, *Mobile*, *Chip*, *Server*
- Support for different Implementations: *Hardware*, *Software*, *WEB*, *Simu*.

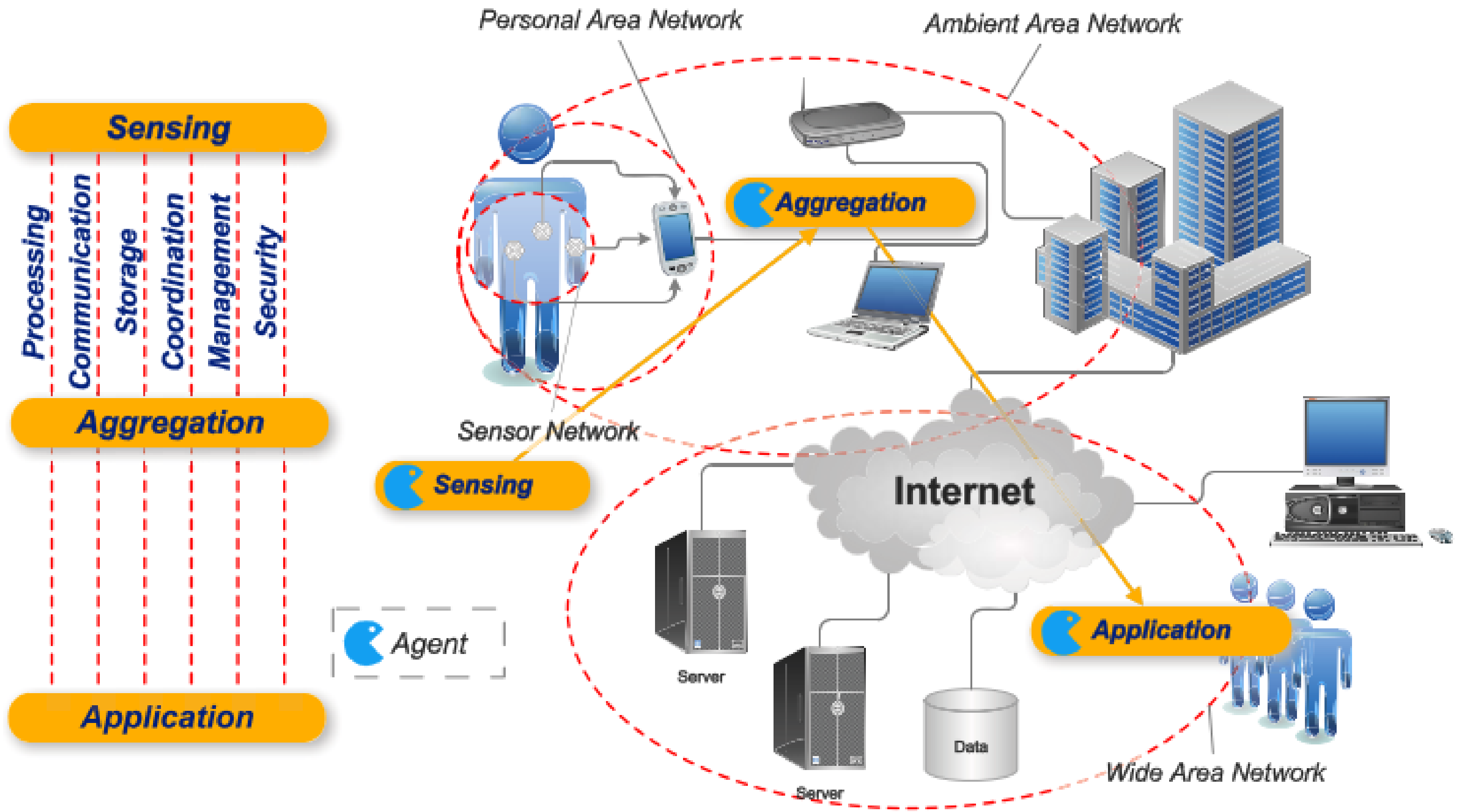


JADE: Bellifemine, 2001
JAM: Bosse, 2015
CAVM: Zhou, 2008
PAVM: Bosse, 2014
PCSP: Bosse, 2013

AAPL: ATG-based Agent Programming Language, Bosse, 2012

F. Operational Layers of Agents

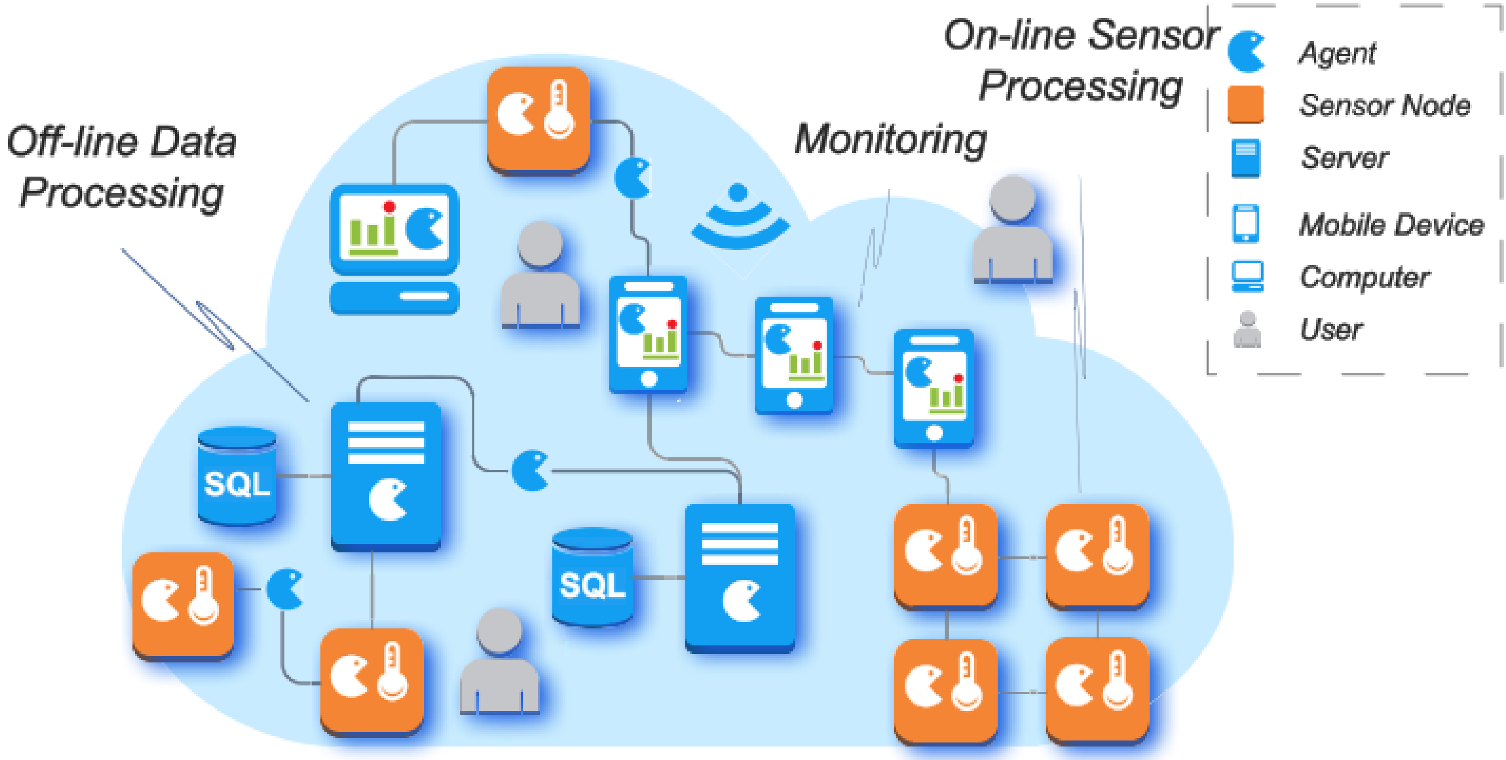
Agents: Sensing ➠ Aggregation ➠ Application
Large Scale & Hierarchical Networks



[Stefan Bosse, A Unified Distributed Computing Framework with Mobile Multi- Agent Systems and Virtual Machines for Large-Scale Applications: From the Internet-of-Things to Sensor Clouds, FEDCSIS Conference 2015]

F. Sensor Clouds and the Internet with Agents

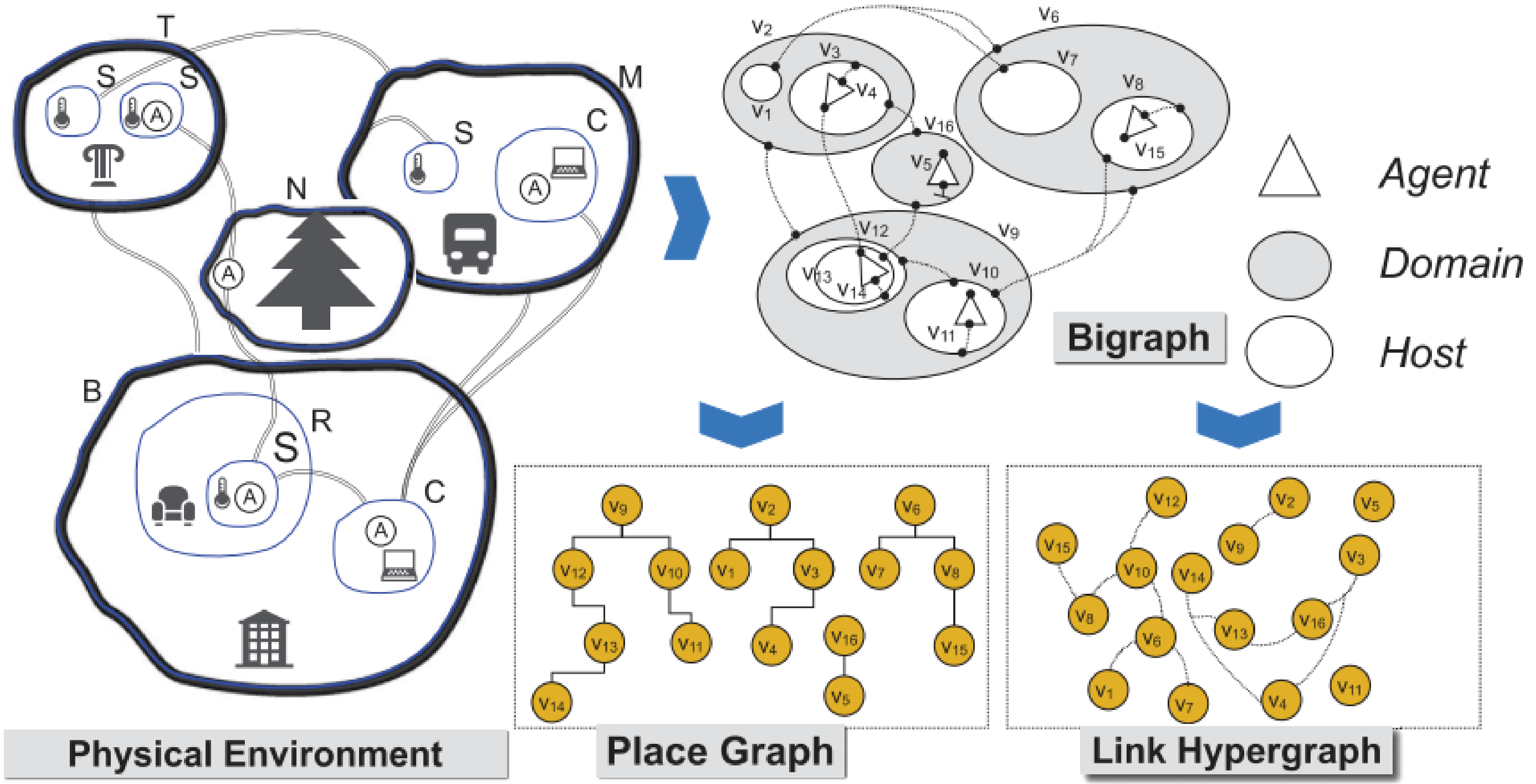
» Deployment of agents can overcome interface and network barriers
  » Closing the gap arising between different platforms and environments
  » Enables integration of sensor networks in WWW applications & Internet



[S. Bosse, From the Internet-of-Things to Sensor Clouds - Unified Distributed Computing in Heterogeneous Environments with Smart and Mobile Multi-Agent Systems, Smart Systems Integration Conference, 2015, Copenhagen]

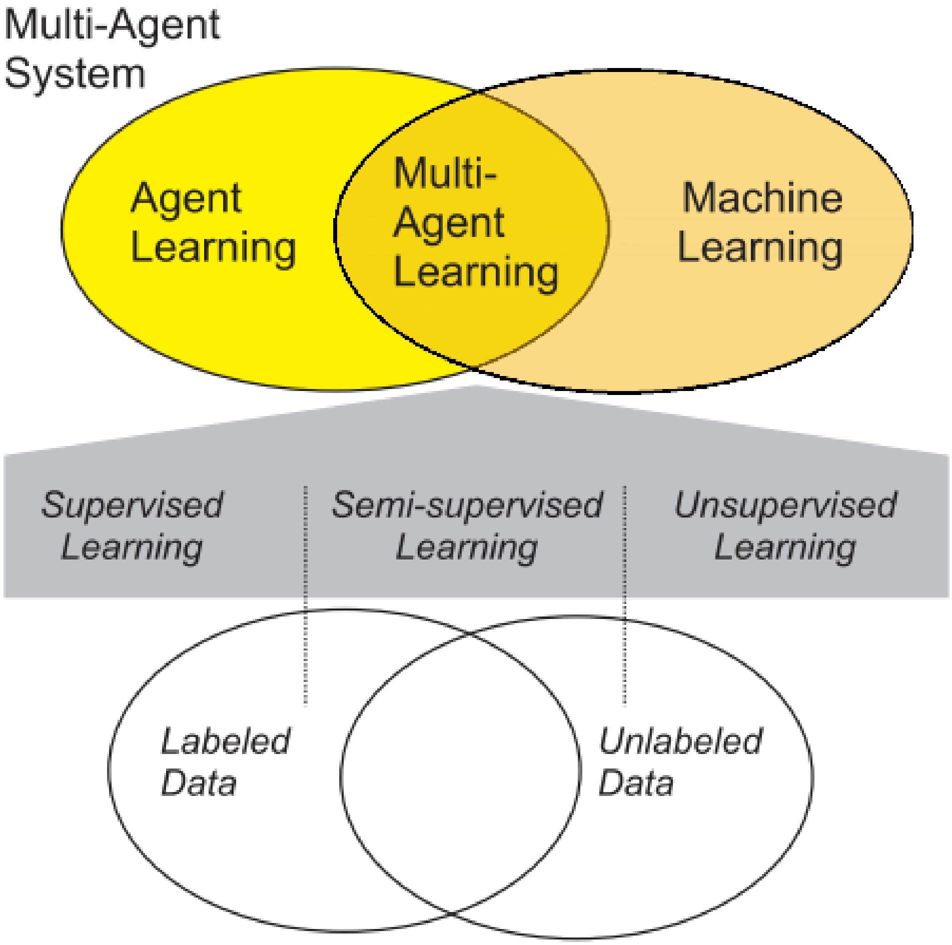
F. Bigraphs and Ontologies; Formal Methods

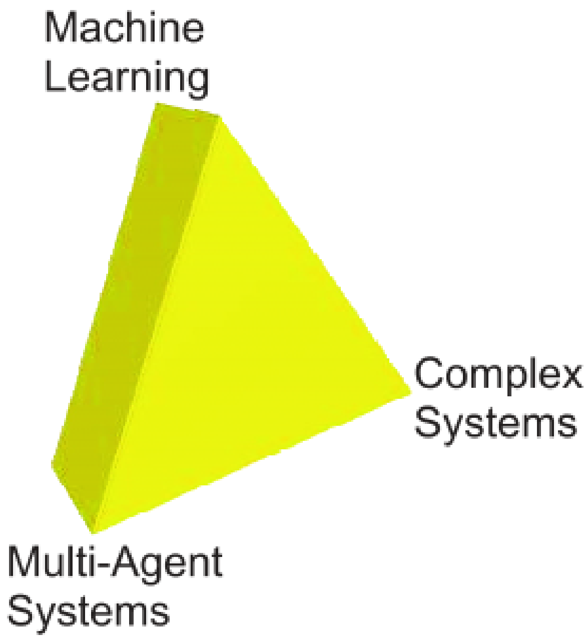
Organization and clustering of nodes in domains, hosts, and agents using the Bigraph Model *[Milner, 2009]*



F. Learning with Agents

Agents and Multi-agent Systems are well suited for Distributed and Regional Learning [Supervised, semi-supervised, and unsupervised]





F. Distributed Learning with Agents

- Regional Classification Learning: Sensor Data space is partitioned in regions with individual learners deployed in the regions.
- Fusion of multiple local learned models and classification ➠ Global Model
  - *K*/*k*: Learner Function, *M*/*m*: Learned Classification Models, *D*/*d*: Labeled Data, *S*/*s*: Sensor Data, *l*: Classification Labels / Symbols

$$K: f(D^l) \rightarrow M$$
$$M: f(S) \rightarrow l$$
$$D^l: (S^1, l_1), (S^2, l_2), \cdots$$
$$S: \begin{pmatrix} s_{1,1} & \cdots & s_{n,1} \\ \vdots & \ddots & \vdots \\ s_{1,m} & \cdots & s_{n,m} \end{pmatrix}$$

$$\xrightarrow{Distribution}$$

$$k_{i,j}: f(d^l_{i,j}) \rightarrow m_{i,j}$$
$$m_{i,j}: f(S_{i,j}) \rightarrow l_{i,j}$$
$$d^l_{i,j}: (S^1_{i,j}, l_1), (S^2_{i,j}, l_1), \cdots$$
$$S_{i,j}: \begin{pmatrix} s_{i-u,j-v} & \cdots & s_{i+u,j-v} \\ \vdots & \ddots & \vdots \\ s_{i+u,j-v} & \cdots & s_{i+u,j+v} \end{pmatrix}$$

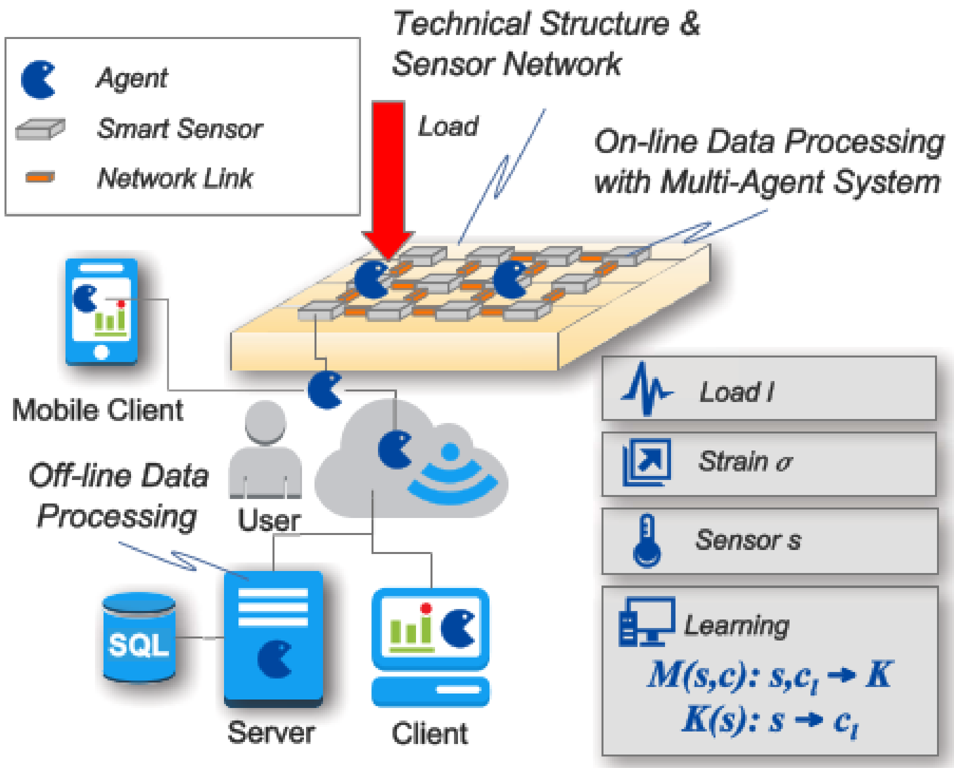
G. Use Cases

The Deployment of Agents in Technical and Environmental Systems

G. Use Case I: Structural Load Monitoring

Concept

» Load Monitoring and Load Classification of a mechanical structure
» Distributed Sensor Network embedded in the mechanical structure
» Mobile agents are deployed for on-line and off-line computation

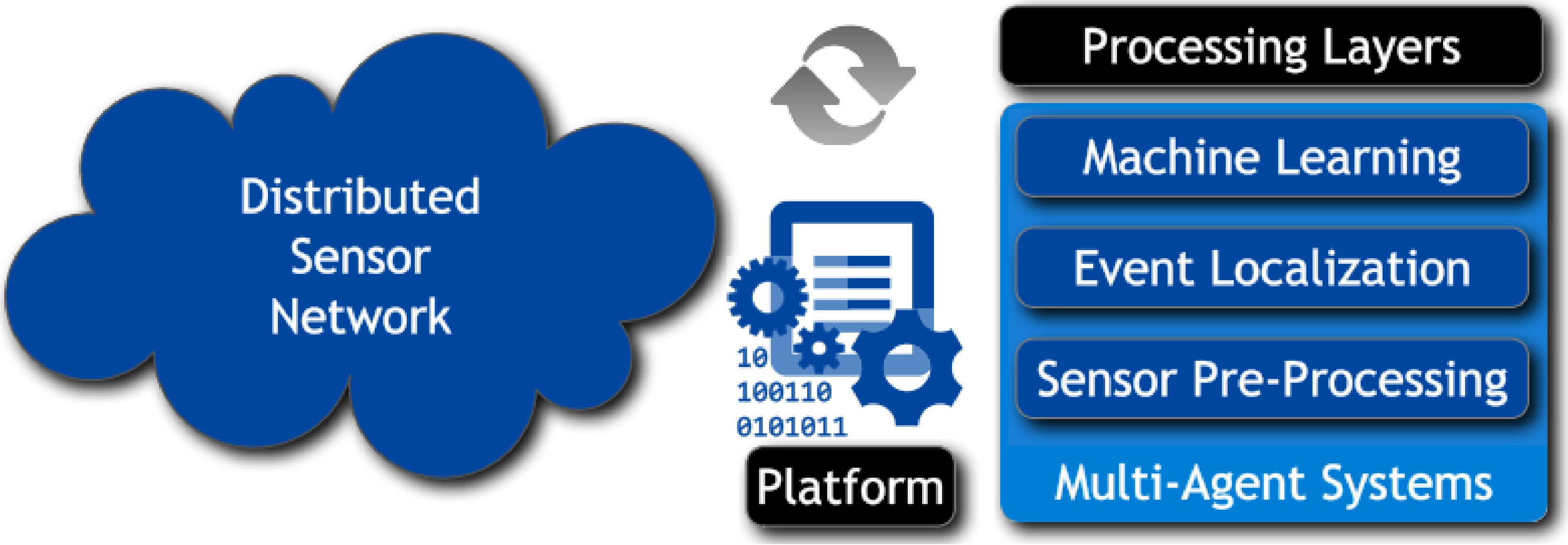


[S. Bosse, Structural Monitoring with Distributed-Regional and Event-based NN-Decision Tree Learning using Mobile Multi-Agent Systems and common JavaScript platforms, SysInt Conference 2016, in Procedia Technology]

G. Use Case I: Structural Load Monitoring (cont.)

Event-based Sensor Processing & Distributed Machine Learning with Agents

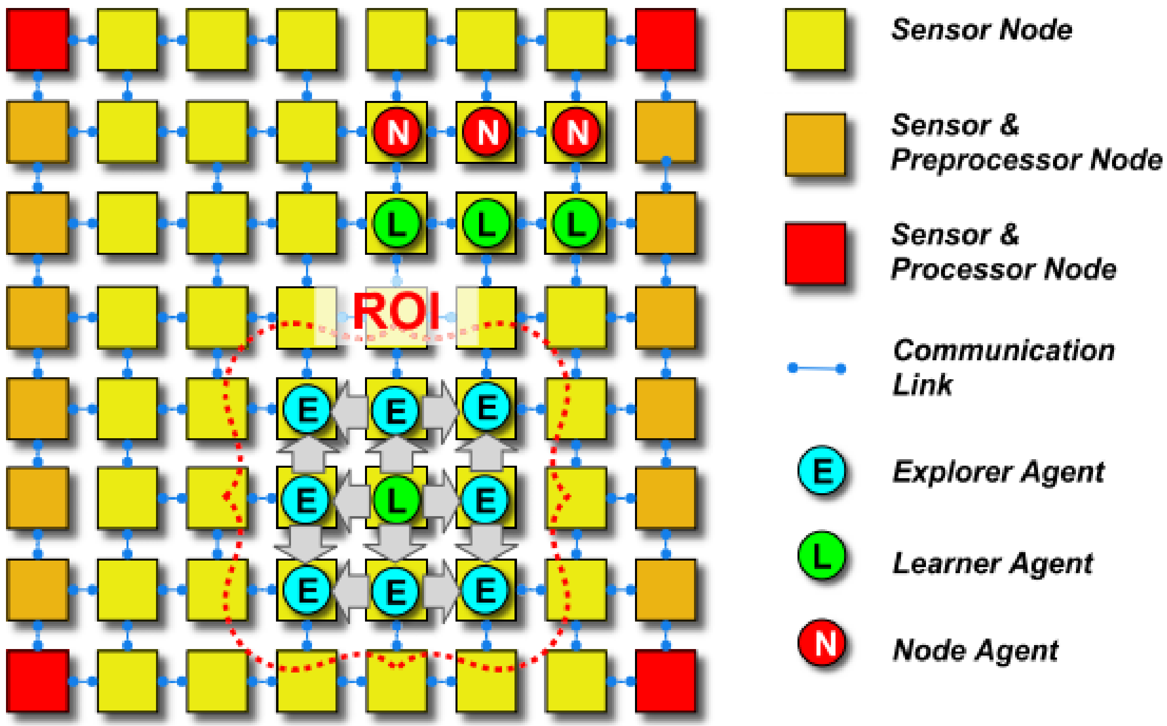
- Sensor-to-Information mapping requires stacked processing levels:
  - Event-based data processing (instead of continuously stream-based)
  - Distributed Machine Learning (DML) of different Load Situations
  - Multi-agent systems used for sensor pre-processing, event localization, learning, computation, and data/information distribution.



G. Use Case I: Structural Load Monitoring (cont.)

Sensor Processing & Machine Learning with different Agent Behaviours[1]

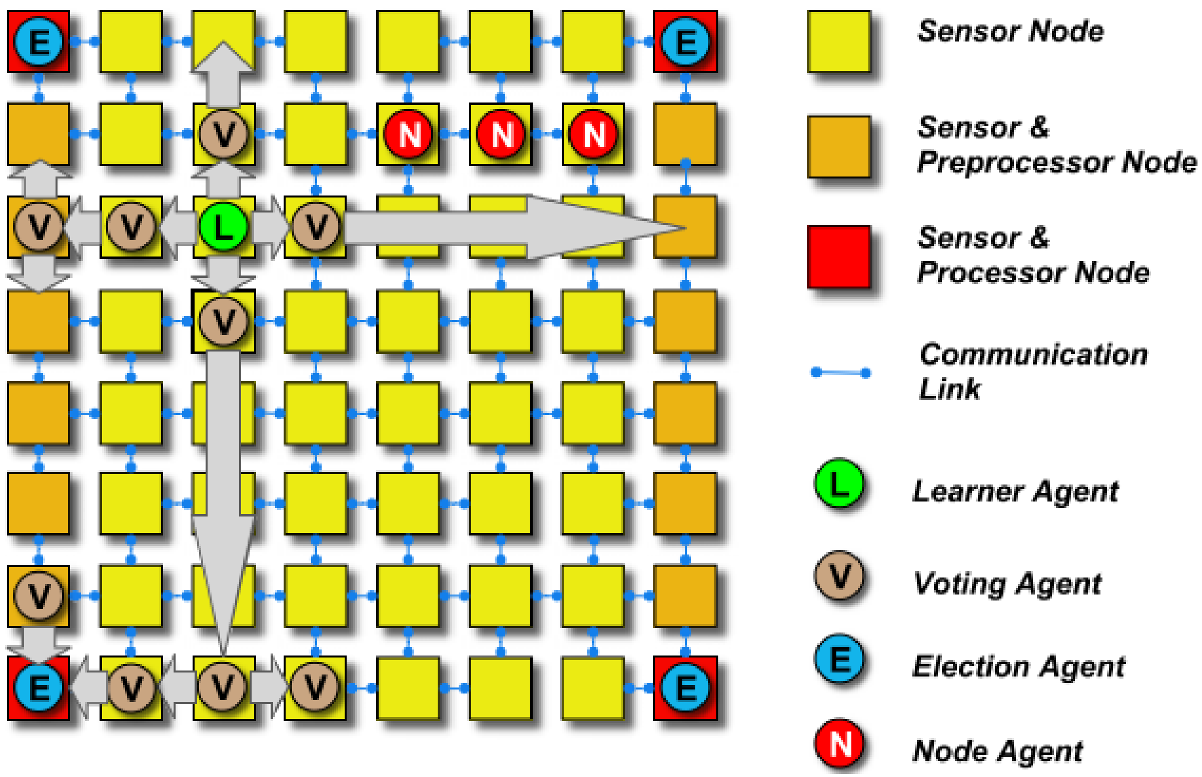
- Node Agent: Local sensor acquisition and event-based Sensor Processing
- Learner Agent: Load Classification with local ML in a Region-of-Interest (ROI)
- Explorer Agent: Self-organizing Divide&Conquer system for Sensor Aggregation



1. [Bosse, DOI 10.1016/j.protcy.2016.08.063, 2016]

G. Use Case I: Structural Load Monitoring (cont.)

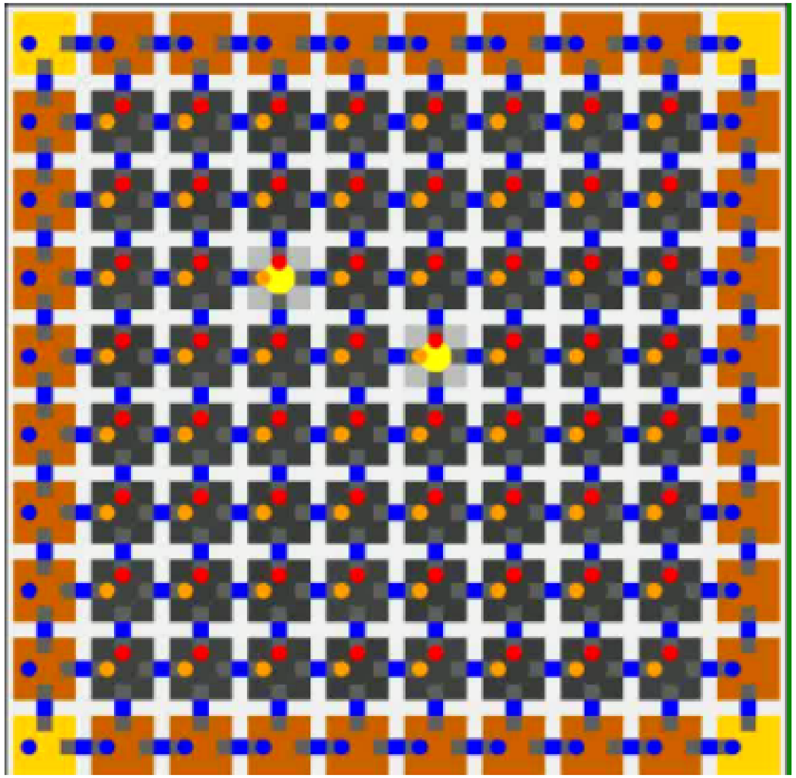
- Local Learner agents perform sensor aggregation in the ROI in *Learning* & *Classification mode* using Explorer agents - mode switch by Notification agents -
- *Classification mode*: Voter agents distribute classification votes to Election agents (network corners)
- *Classification mode*: Global voting with majority decision by Election agents

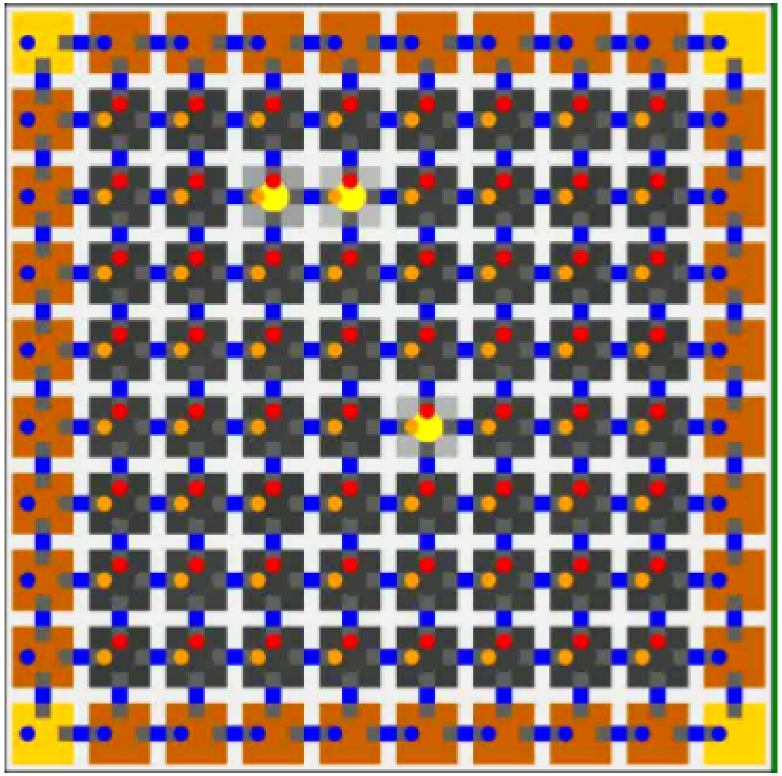


G. Use Case I: Structural Load Monitoring (cont.)

Ex.: Event-based Sensor Distribution with Explorer Agents and D&C

- Explorer agents try to find clusters of stimulated sensors in a ROI
- If a correlated cluster was detected, Distribution agents are send out[1]

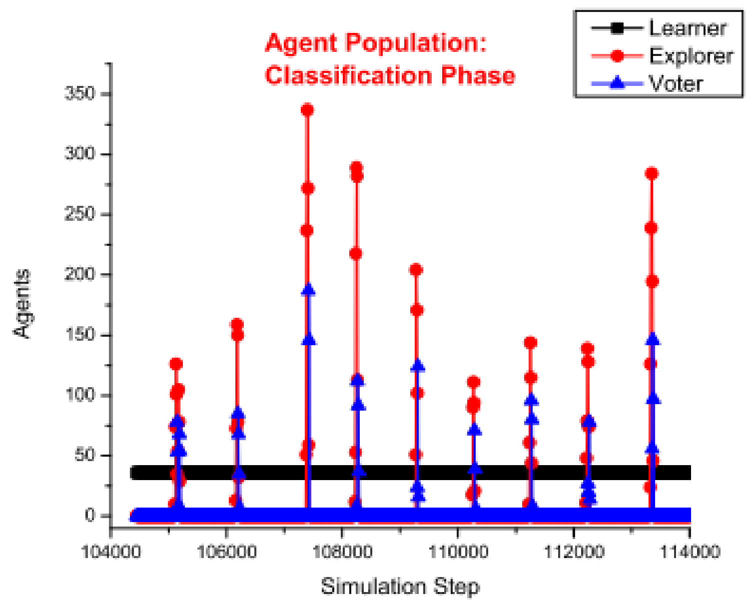


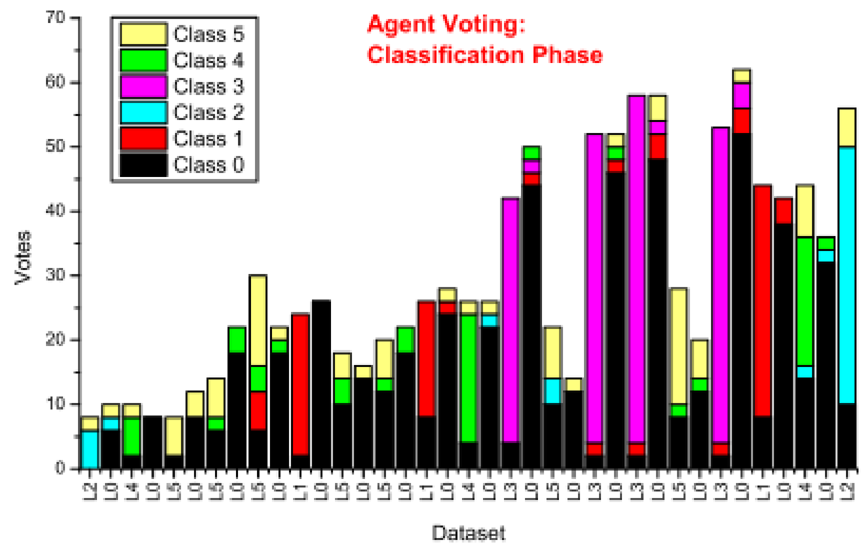


1. [Bosse, Lechleiter; DOI:10.1016/j.mechatronics.2015.08.005; 2016]

G. Use Case I: Structural Load Monitoring (cont.)

- Simulation & Results
  - MAS Simulation in SEJAM (Simulation Environment on top of a JAM VM)
  - Sensor Data from FEM Simulation - Different Load Situations (L0,L1,..)
  - (Left) Event-based Agent Population (Right) Classification Results (Ex.)





[S. Bosse, Structural Monitoring with Distributed-Regional and Event-based NN-Decision Tree Learning using Mobile Multi-Agent Systems and common JavaScript platforms, SysInt Conference 2016, in Procedia Technology]

G. Use Case II: Distributed Earthquake Monitoring

» Same approach as used in Structural Monitoring, Micro- → Macro-Scale

» Tasks performed with mobile agents:

  » Distributed learning and prediction of earthquake events

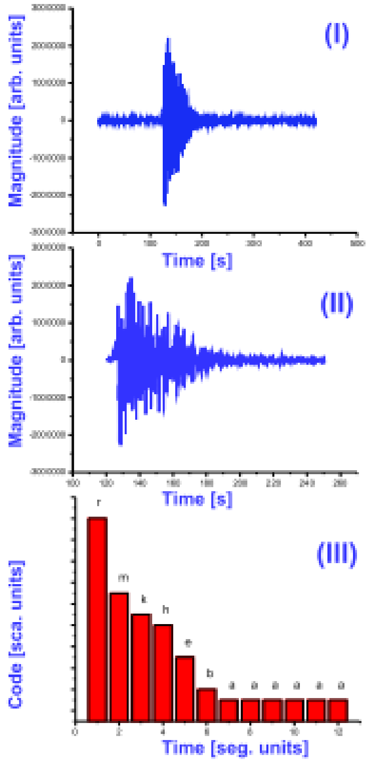
  » Sensor pre-processing and event-based sensor distribution

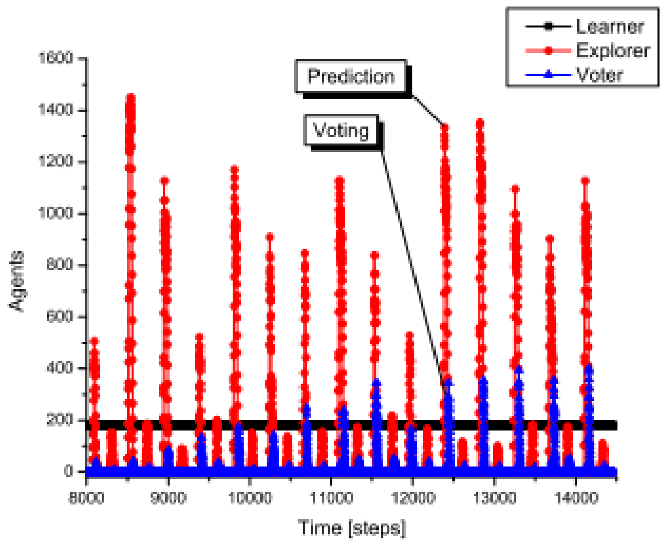
[Left: Google Maps, s: Seismic Station of the CI Network] [Middle: Seismic Stations+Mobile Devices = Sensor Cloud]
[S. Bosse, Distributed Machine Learning with Self-organizing Mobile Agents for Earthquake Monitoring, SASO/DSS Conference 2016]

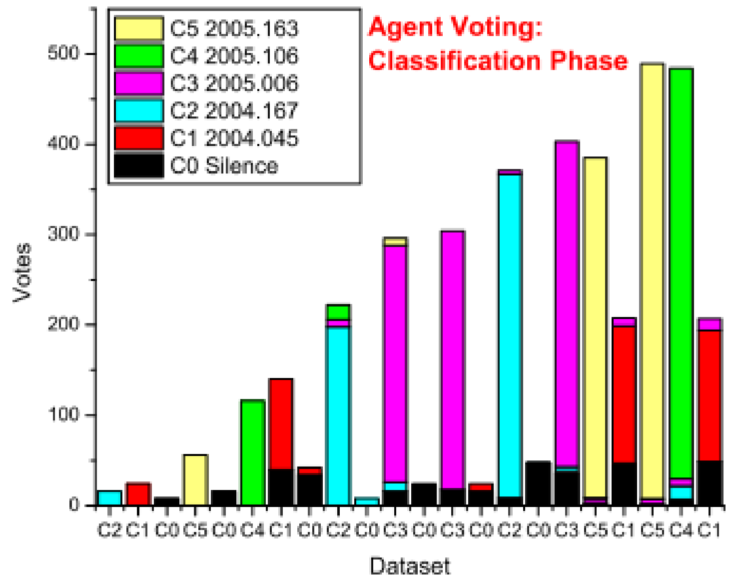
G. Use Case II: Distributed Earthquake Monitoring (cont.)

» Simulation & Results

  » Real Seismic Data from Earthquake events were used in Simulation
  » (Left) Data Reduction (Middle) Agent Population (Right) Class. Results (Ex.)







[S. Bosse, Distributed Machine Learning with Self-organizing Mobile Agents for Earthquake Monitoring, SASO/DSS Conference 2016]

H. Conclusions

- The Internet-of-Things including devices equipped with Smart Materials will grow exponentially in the next decades.
  - Result: Strong heterogeneous network environments with trillions of nodes
- Computing with mobile agents can unite different network and host platforms
- Multi-agent and Self-organizing Systems can scale up and support Divide-and-Conquer algorithms.
- Agent platforms basing on virtual machines and mobile code can occupy different host platforms on a wide scale.
- Regional Learning with global fusion is an efficient and suitable distributed approach with excellent scalability.

Industrial Agents and Distributed Agent-based Learning

Stefan Bosse

University of Bremen

Department of Mathematics & Computer Science,

ISIS Scientific Centre for Sensorial Materials

Bremen, Germany