

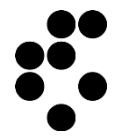
The web of things

CAROLINA FORTUNA AND MARKO GROBELNIK

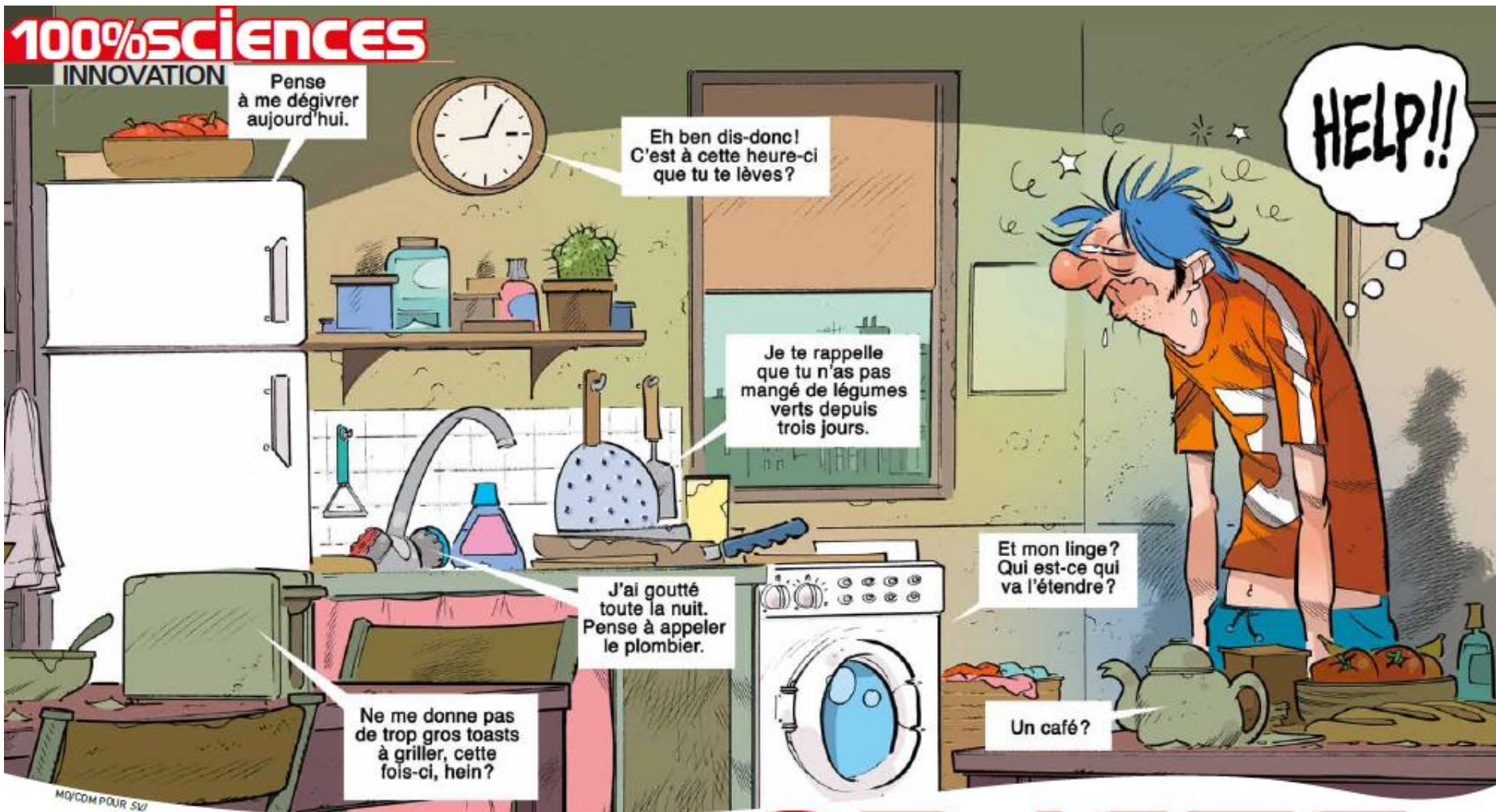
CAROLINA.FORTUNA@IJS.SI MARKO.GROBELNIK@IJS.SI

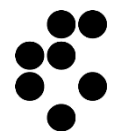
JOŽEF STEFAN INSTITUTE, LJUBLJANA, SLOVENIA

HTTP://SENSORLAB.IJS.SI



The WoT explained to children





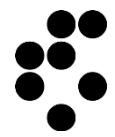
A case for the web of things

Facts:

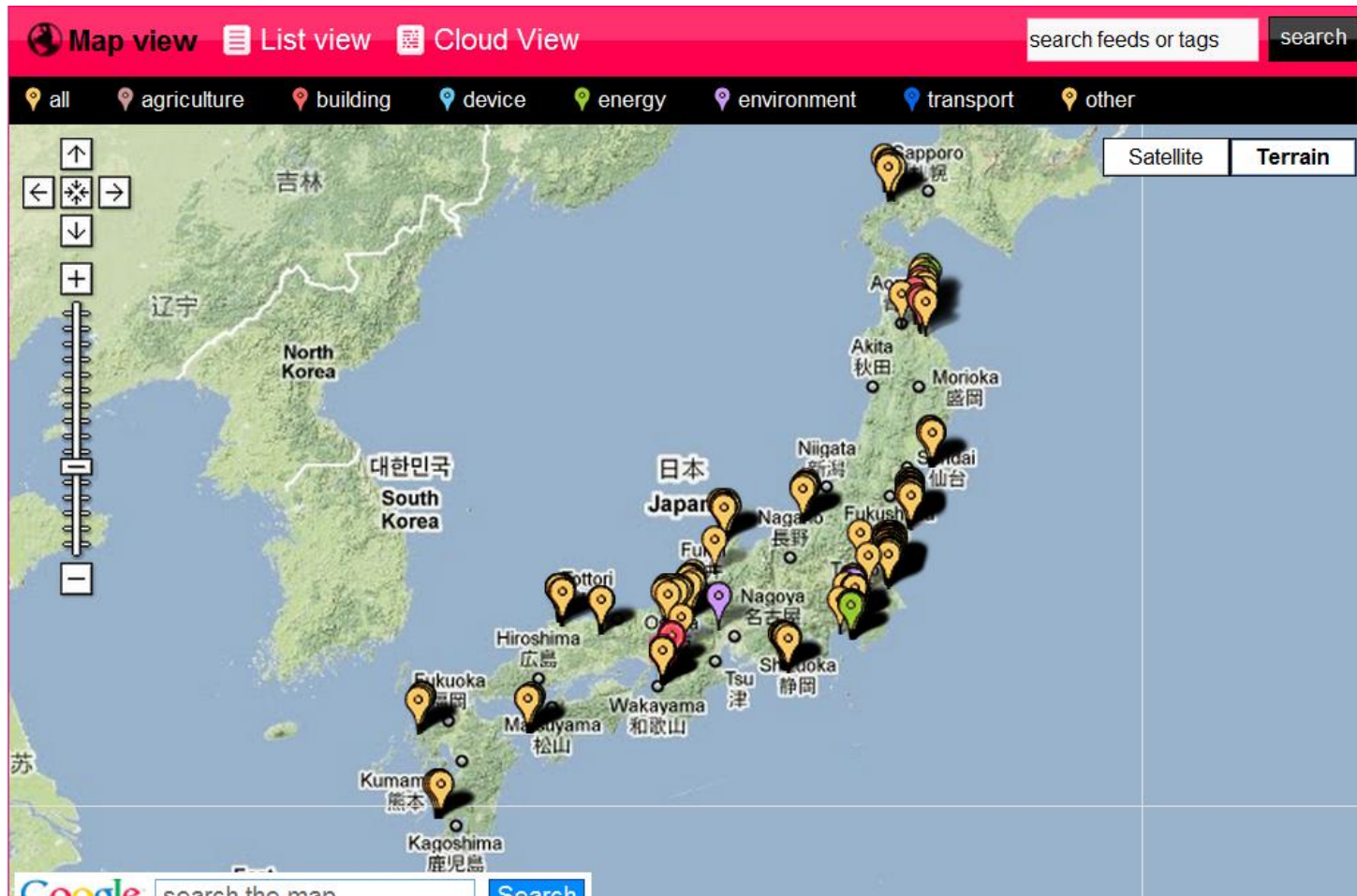
- March 11, 2011: Tōhoku earthquake and tsunami in Japan
- Nuclear reactors were affected: explosions and radioactive pollution
- Confusing information about the **levels of radioactivity** from authorities
- **Radiation level maps** based on Geiger counter data started to appear



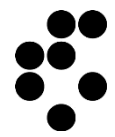
http://en.wikipedia.org/wiki/2011_Tōhoku_earthquake_and_tsunami



Radiation level map



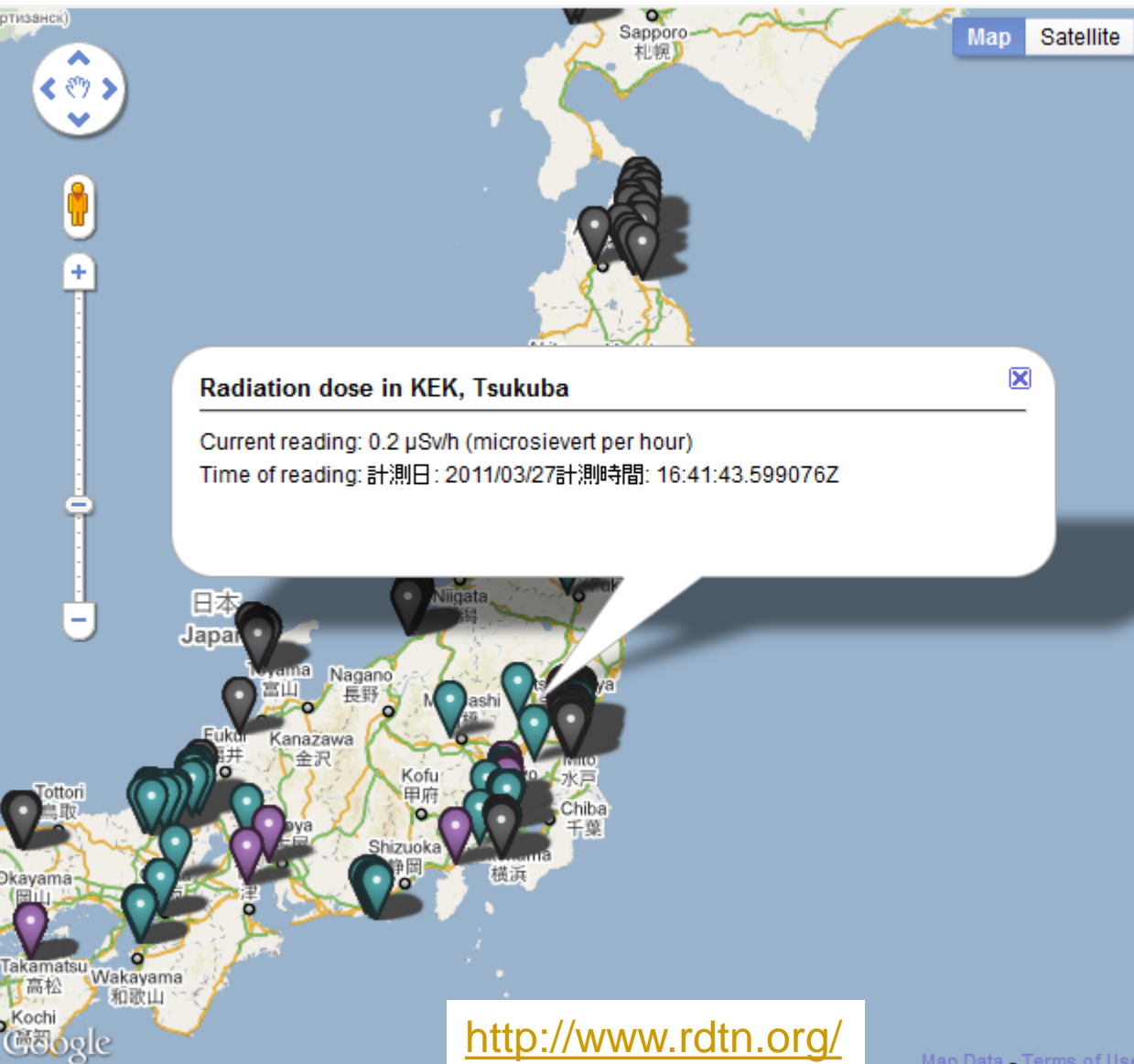
<http://blog.pachube.com/2011/03/real-time-radiation-monitoring-in-japan.html>



Radiation level map



Radiation level map



Readings

We too have been watching events unfold in Japan. We have created this site in an effort to display the reliable data readings as they become available. Although we are careful to evaluate new data sources, we welcome new reliable data from those on the ground in this crisis.

To submit readings

1) Purchase a radiation detection device.

INTERNATIONAL MEDCOM

AMAZON

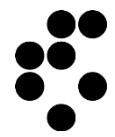
LAB SAFETY SUPPLY

COLE-PARMER

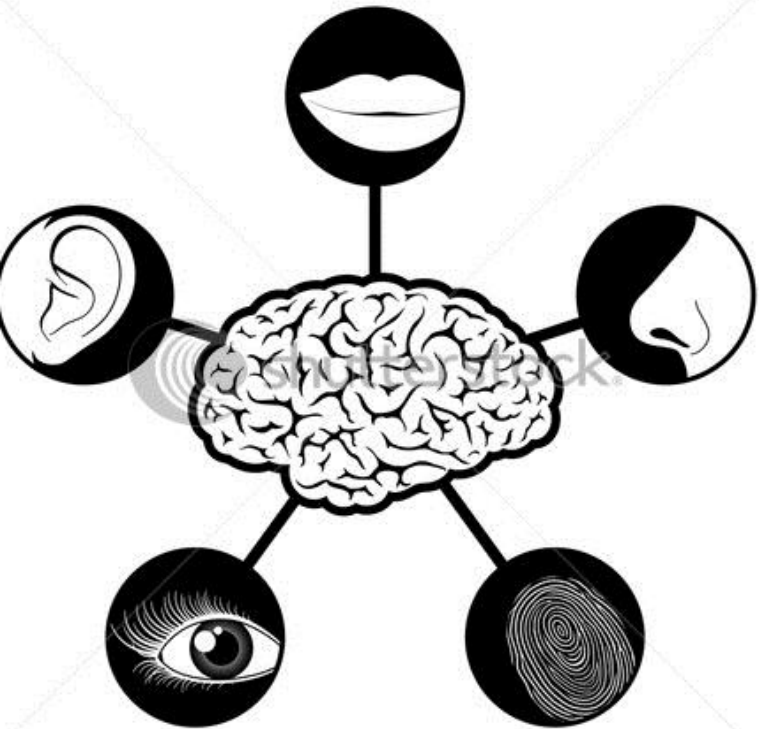
2) Take readings in your area.

3) Post readings to RDTN.

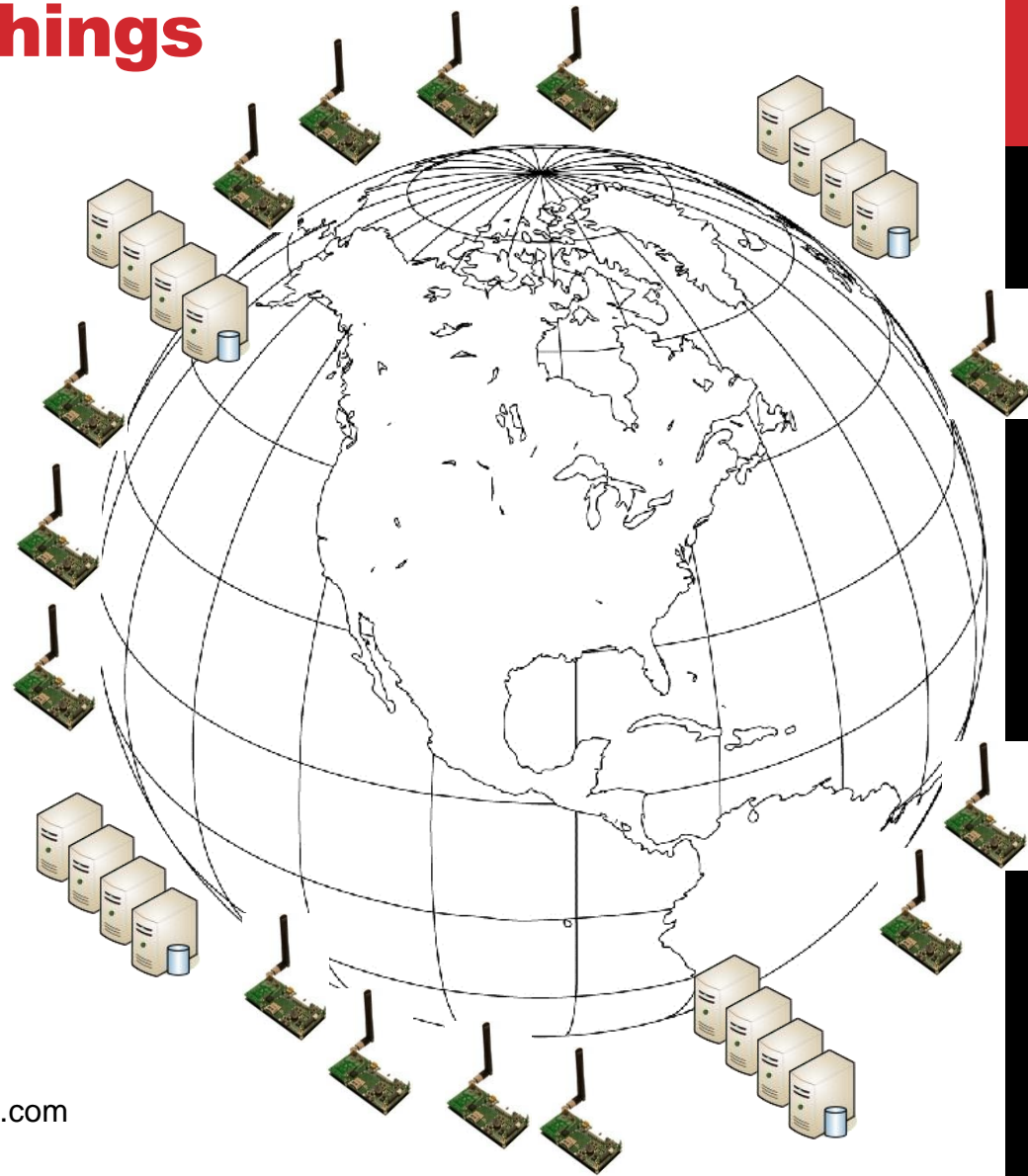
SUBMIT A READING >



Analogy humans - things

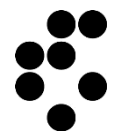


www.shutterstock.com · 59068843



Pictures: <http://www.shutterstock.com>, freusandworldmaps.com

Fortuna et al., Towards Building a Global Oracle: a Physical Mashup Using Artificial Intelligence Technology, International Workshop on the Web of Things, 2012.



Outline

Part I. Motivation & background

Part II. Technology and tools for exploiting the WoT

Part III. Demos, tools & research directions



Part I. Motivation & background outline

Web Of Things

- What is it? What problems can it solve?

Architectural considerations

- How it looks like? What are its components?

The “Things”

- What are the ingredients?

The “Glue”

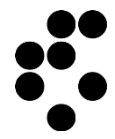
- How do things stick together?

Applications and services

- What can be built on top of it?

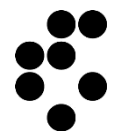
Quick start recipes

- How does the “Hello World!” look like?



How Web-of-things fits on the map?

	Description	Technologies
Web 1.0	Static HTML pages (web as we first learned it)	HTML, HTTP
Web 1.5	Dynamic HTML content (web as we know it)	Client side (JavaScript, DHTML, Flash, ...), server side (CGI, PHP, Perl, ASP/.NET, JSP, ...)
Web 2.0	Participatory information sharing, interoperability, user-centered design, and collaboration on the World Wide Web (web of people)	weblogs, social bookmarking, social tagging, wikis, podcasts, RSS feeds, many-to-many publishing, web services, ... URI, XML, RDF, OWL, SparQL, ...
Web 3.0	...definitions vary a lot – from Full Semantic Web to AI (web as we would need it)	http://en.wikipedia.org/wiki/Web_3.0#Web_3.0
Web of Things	Everyday devices and objects are connected by fully integrating them to the Web. (web as we would like it)	Well-accepted and understood standards and blueprints (such as URI, HTTP, REST, Atom, etc.) http://en.wikipedia.org/wiki/Web_of_Things



Web of Things vs Internet of Things: what is the difference?

Internet = Interconnected networks

- They are interconnected via IP (Internet Protocol)
- There are IP addresses in the internet, no domain names such as wikipedia.org
- Started around 1950 in an effort to make two computers talk to each other

Web = Linked documents and resources

- Uses HTTP
- The web needs the Internet underneath to function
- Started around 1980 in an effort to help people share data over the Internet



Transition towards machine generated information

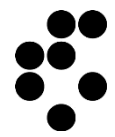
Past:

“manual input of information by 500 million or a billion users”¹

Future:

“new information can be created automatically without human data entry... the next generation of sensor networks can monitor our environment and deliver relevant information – automatically.”¹

¹[Pete Hartwell, How a Physically Aware Internet Will Change the World, Mashable, October 13, 2010.](#)



Web of things use cases

Motivated by an increased interest in automatic management of large systems

- Commercial use cases¹ (non-exhaustive list):
 - Power grids
 - Transport systems
 - Water distribution
 - Logistics
 - Industrial automation
 - Health
 - Environmental intelligence
- Academic
 - Distributed sensing infrastructure

Alternative solutions

Ethical issues and abuse¹

¹[Ludwig Siegele, A special report on smart systems, The Economist, Nov. 4 2010.](#)



Commercial use case: Power grids¹

“If the power grid in America alone were just 5% more efficient, it would save greenhouse emissions equivalent to 53m cars (IBM).“

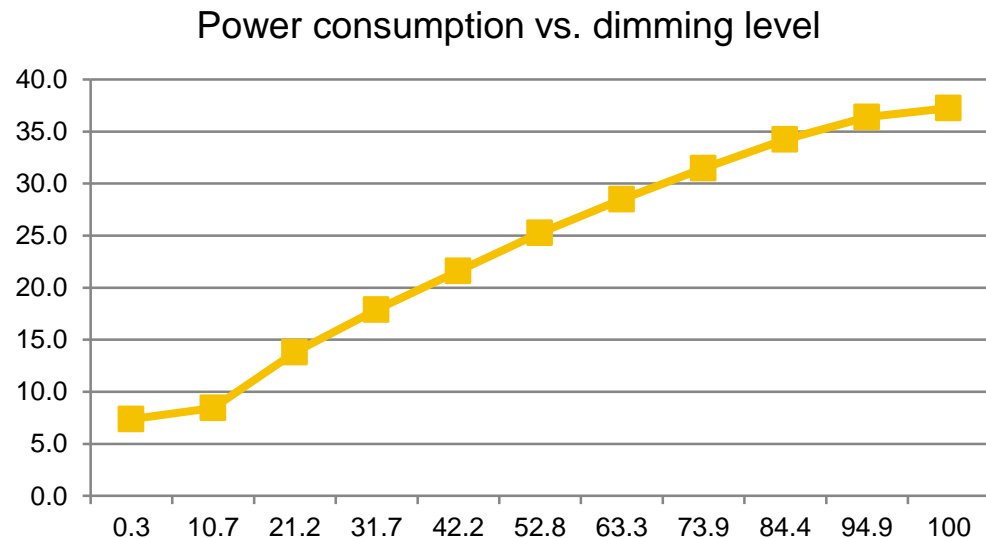
Solutions:

- demand pricing – 10-15% peak hour demand cut
 - Energy consumption monitoring with smart meters encourage shifting consumption to off-peak hours through personalized price plans
- demand response – extra 10-15% cut
 - Save energy by sensing and actuation: smart meters + actuators turn off air-conditioning systems when demand for electricity is high

¹[Ludwig Siegele, A special report on smart systems, The Economist, Nov. 4 2010.](#)

Public lighting control: greener lights + control

Dimming level [%]	Power consumption [W]
0,3	7,4
10,7	8,4
21,2	13,8
31,7	17,9
42,2	21,6
52,8	25,2
63,3	28,5
73,9	31,5
84,4	34,2
94,9	36,4
100	37,2



SGA LSL 30 main characteristics

- the number of LED: 30*1w
- consumption: 35W (at full power)
- colour of the light: 4200K
- light current: 2700lm
- life-expectancy: min. 60.000h
- IP66
- NET mass: 4,8k



Public lighting control: lights dimming

Lights dimmed to 75% luminosity between 23:00 and 5:00 with smooth 15-minute linear transitions.

$$P(\text{not dimmed}) = 37.2W$$

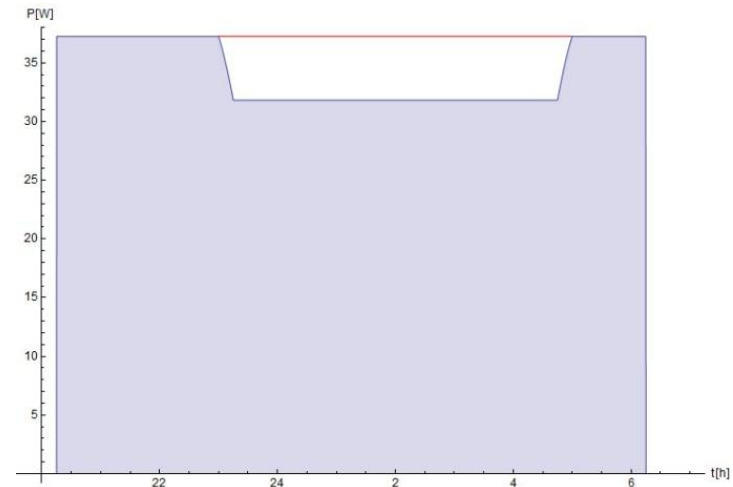
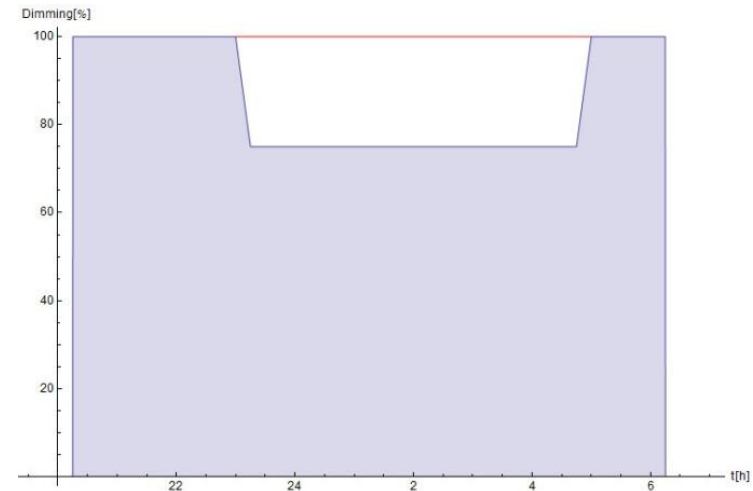
$$P(\text{dimmed}) = 31.8W$$

Electricity consumption per night:

$$A_e(\text{no dimming}) = 0.372kWh$$

$$A_e(\text{dimming}) = 0.341kWh$$

Reduced by **~ 8,3%**.



Dimming and power in time.
Red line represents light poles with no dimming.



Commercial use case: Transport systems¹

“In 2007 its congested roads cost the country 4.2 billion working hours and 10.6 billion litres of wasted petrol (Texas Transportation Institute)”¹

Solutions:

- Charging for city centers and busy roads
 - London, Stockholm, Singapore, etc.
- Green wave
 - Adjustment of traffic lights to suit the flow of vehicles
- Automatic parking guidance
 - Singapore is developing a parking-guidance system (cars looking for somewhere to park are now a big cause of congestion).
- Real-time dynamic pricing
 - Singapore

¹[Ludwig Siegele, A special report on smart systems, The Economist, Nov. 4 2010.](#)



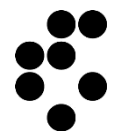
Commercial use case: Water distribution¹

Utilities around the world lose between 25% and 50% of treated water to leaks (Lux Research).

Solutions:

- Renew infrastructure
 - London, UK, Thames Water was losing daily nearly 900m litres of treated water and had to fix 240 leaks due to aging infrastructure¹.
- Install sensors for monitoring the pipe system
 - Automatically detect leaks fast (instead of customers calling and reporting leaks). London, Singapore, etc.
- Automate the management and maintenance process
 - Automatic scheduling of work crews and automatic alerts (i.e. text messages to affected customers)

¹[Ludwig Siegele, A special report on smart systems, The Economist, Nov. 4 2010.](#)

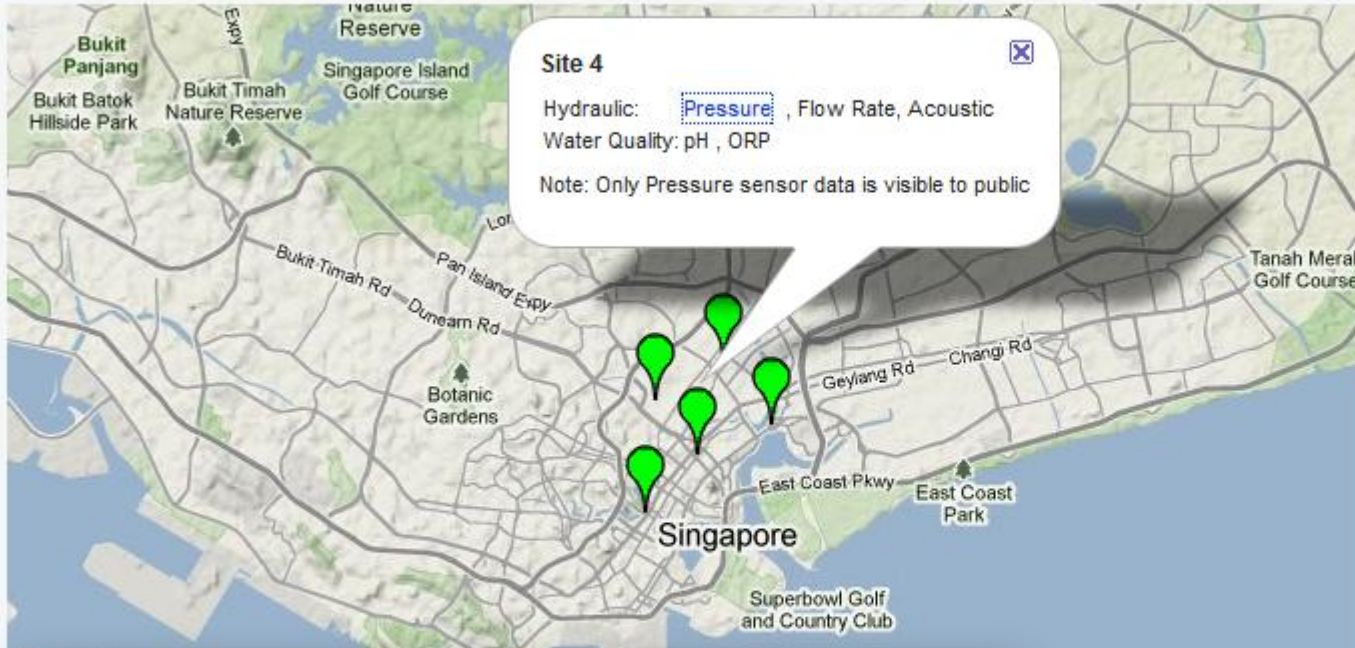


Water distribution

WaterWiSe in Singapore

- develop generic wireless sensor network capabilities to enable real time monitoring of a water distribution network.
- three main applications:
 - On-line monitoring of hydraulic parameters within a large urban water distribution system.
 - Integrated monitoring of hydraulic and water quality parameters.
 - Remote detection of leaks and prediction of pipe burst events.

You are free to play with a limited data sets from a few deployment sites you see on the interactive map. We are streaming a lot more data for many more hydraulic, water quality and derived parameters from many other locations that are not available through this public portal.



Selected stations:

Pressure_Site 4

Remove

Time:

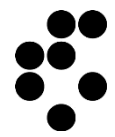
- Last 24 Hours (Archived)
- Last 7 Days
- Last 30 Days
- Time as below:

Start Date	H	M	S
2011-03-26	12	58	16
End Date	H	M	S
2011-03-27	13	58	16

Assume Time Zone as:

- Data Time Zone
- My Time Zone





Commercial use case: Logistics

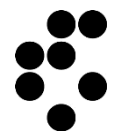
Cargo loss due to theft or damage is significant, estimates that the global financial impact of cargo loss exceeds \$50 billion annually (The National Cargo Security Council)¹. The cost is eventually passed to the customers.

Solutions:

- Automatic track and trace
 - Tag and trace their wares all along the supply chain (RFIDs and sensors) - and consumers to check where they come from (i.e. FoodLogiQ, SenseAware)²
- Event detection and mitigation
 - Detect events that affect the cargo (i.e. delay, inappropriate transport conditions) and minimize damage (i.e. re-route)

¹ [Tom Hayes, The Full Cost of Cargo Losses](#)

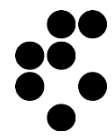
² [Ludwig Siegele, A special report on smart systems, The Economist, Nov. 4 2010.](#)



Logistics

- SenseAware
 - temperature readings
 - shipment's exact location
 - shipment is opened or if the contents have been exposed to light
 - real-time alerts and analytics between trusted parties regarding the above vital signs of a shipment





Supply chain mash-up

epcisWEB adapter

Product Description

Product Image

Product Video

Stock History

Stock Info

Map

Product Buzz

Feed

Event Statistics

EPCIS Browser

Event Finder

Location

urn:br:maxhavelaar:natal

urn:br:maxhavelaar:palm

urn:ch:frey:buchs:factory

urn:ch:migros:basel:ware

urn:ch:migros:stgallen:wa

urn:ch:migros:zurich:distr

Reader

urn:ch:frey:buchs:convey

urn:ch:frey:buchs:factory

Time

Electronic Product Cod

urn:epc:id:sgtin:61800.82

urn:epc:id:sgtin:61800.82

urn:epc:id:sgtin:61800.820712.2003

urn:epc:id:sgtin:61800.820712.2004

urn:epc:id:sgtin:61800.820712.2005

urn:epc:id:sgtin:61800.352613.1001

urn:epc:id:sgtin:61800.352613.1002

The screenshot shows the EPCIS Mashup Dashboard with several active panels:

- Twitter Buzz:** A list of tweets related to Lindt Chocolate, such as "White Chocolate Lindt Bats are my favourite!" and "I'm celebrating as we speak - Hotel...".
- Product Description:** Details for Lindt Chocolate, including a Wikipedia link and a "Praline" article snippet.
- Event Finder:** A search window showing a list of EPCIS events with columns for Location, Reader, and Time. The location "urn:ch:frey:buchs:factory" is highlighted.
- Calendar:** A monthly calendar for December 2010, showing dates from 1st to 25th.
- Map:** A Google Map showing the location of Lindt Chocolate in SAP Future Retail Center.

Lindt Chocolate

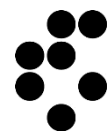
Lindt Chocolate

Lindt Chocolate

Chocolate Frey

Chocolate Frey

<http://epcmashup.appspot.com/>



Commercial use case: Industrial automation

The integration gap between the production and business processes comes at a high cost, especially in multi-site enterprises.

Solutions:

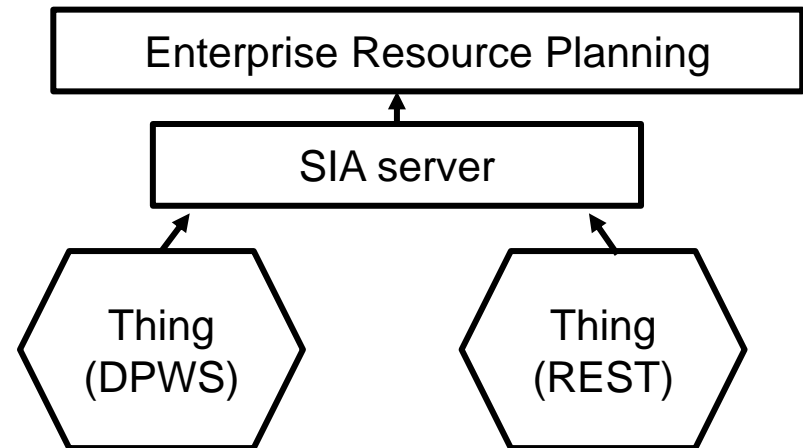
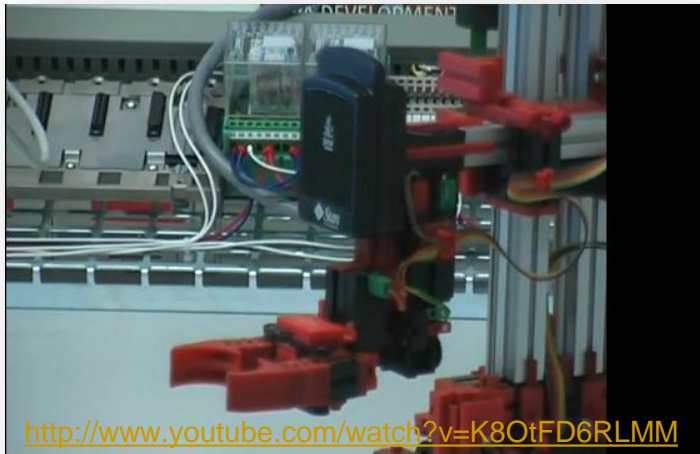
- Automatic monitoring of the production process
 - Monitor the devices on the production floor (i.e. robotic arm overheating)¹
- Automatic event detection and notification
 - Process the measurements, detect anomalies and notify the business process (i.e. production at site interrupted, relocate)
- Productivity comparison
 - Machines equipped with sensors allow productivity comparison based on sensed data (i.e. Heidelberger Druckmaschinen)²
- Dynamic production optimization
 - 5% increase in paper production by automatically adjusting the shape and intensity of the flames that heat the kilns for the lime used to coat paper²

¹SOCRADES project, <http://www.socrades.eu/>

²Ludwig Siegele, A special report on smart systems, The Economist, Nov. 4 2010.

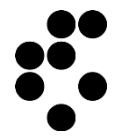
Process integration

- SunSpot on Robotic ARM, exposing measurements as Web service
- SunSpot GW connected to Windows machine, then to the Enterprise Network or Internet
- Failure, production interruption alarm – moving to alternative production site



¹SOCRADES project, <http://www.socrades.eu>

²D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio, Interacting with the SOA-based Internet of Things: Discovery, Query, Selection and On-Demand Provisioning of Web Services, IEEE Transactions on Services Computing, Vol. 3, July-Sept 2010.



Discover things

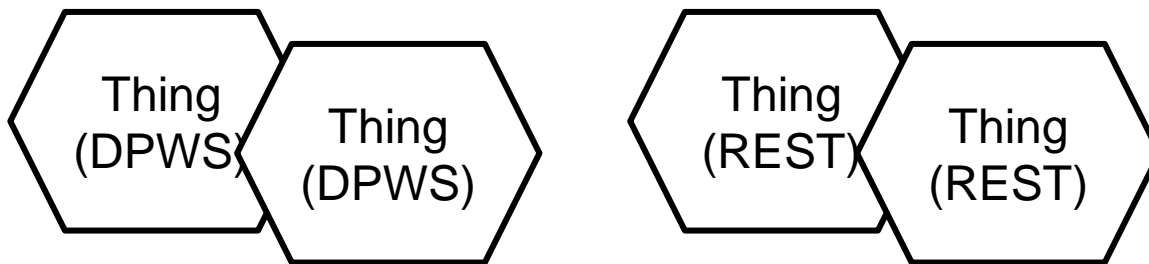
Automatic context data collection

Device Profile for Web Services (DPWS)

- Subset of Web Service standards (WSDL and SOAP)
- Successor of Universal Plug and Play (UPnP)

Representational State Transfer (REST)

- Lightweight, suitable for less complex services

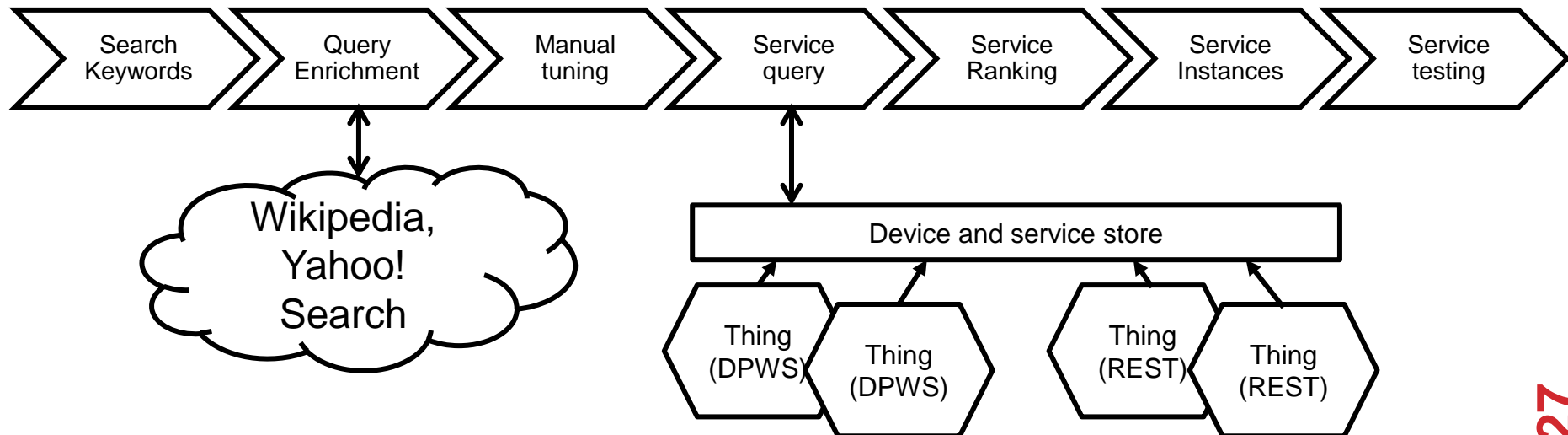


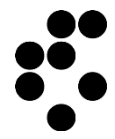
¹SOCRADES project, <http://www.socrades.eu>

²D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio, Interacting with the SOA-based Internet of Things: Discovery, Query, Selection and On-Demand Provisioning of Web Services, IEEE Transactions on Services Computing. Vol. 3. July-Sept 2010.

Query embedded services

- Insert search keywords, perform query enrichment (augmentation)
 - Tested 2 strategies: Wikipedia and Yahoo! Search
- Manually tune the augmented query by adding/deleting keywords
- Search services in the store and rank them according to some criteria (i.e. QoS)



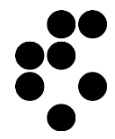


Commercial use case: Health

In health care, sensors and data links offer possibilities for monitoring a patient's behavior and symptoms in real time and at relatively low cost.¹

Solutions:

- Patient monitoring
 - When suffering from chronic illnesses can be outfitted with sensors to continuously monitor their conditions as they go about their daily activities.
 - Asthma, diabetes, heart-failure
- Extended healthcare for elders
 - Needs to extend from hospital to home care to ensure cost efficient provisioning and improve quality of living (ambient assisted living).
 - Fall detection, emergency call, user localization, hazard monitoring (toxic gases, water, fire)
- Fitness monitoring for personalized fitness scenario



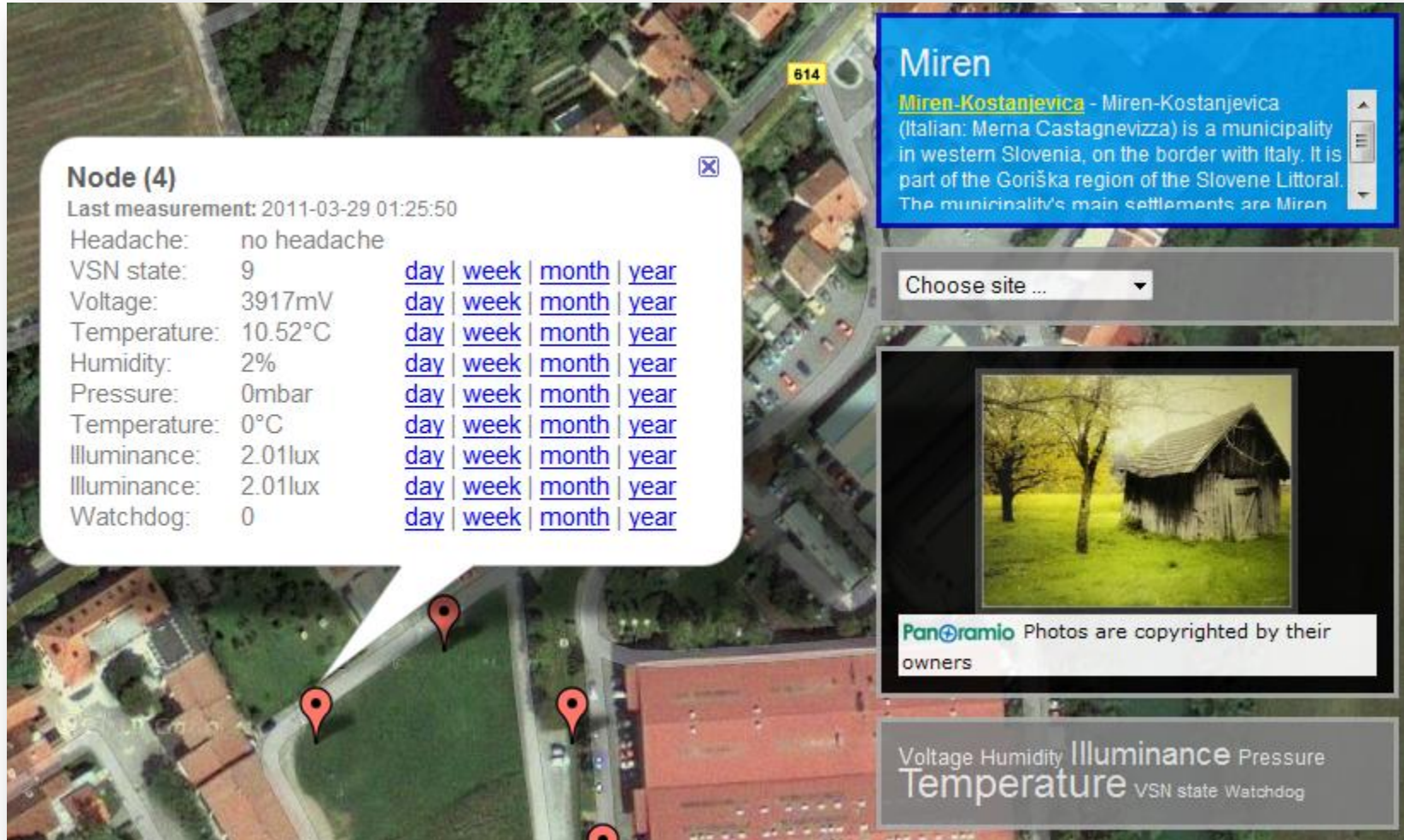
Commercial use case: Environmental intelligence

Data from large number of sensors deployed in infrastructure (such as roads) or over other area of interest (such as agriculture fields) can give decision makers a real-time awareness on the observed phenomena and events.

Solutions:

- Remote monitoring of cultures, soil moisture, insect infestations or disease infections
- Irrigation and pesticide spraying in precision agriculture
- Livestock monitoring for maximizing production (meat, milk, eggs) and achieve higher reproduction rates

Videk – AI mash-up



Node (4) ✕


Last measurement: 2011-03-29 01:25:50

Headache:	no headache	
VSN state:	9	day week month year
Voltage:	3917mV	day week month year
Temperature:	10.52°C	day week month year
Humidity:	2%	day week month year
Pressure:	0mbar	day week month year
Temperature:	0°C	day week month year
Illuminance:	2.01lux	day week month year
Illuminance:	2.01lux	day week month year
Watchdog:	0	day week month year

Miren

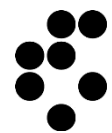
Miren-Kostanjevica - Miren-Kostanjevica (Italian: Merna Castagnevizza) is a municipality in western Slovenia, on the border with Italy. It is part of the Goriška region of the Slovene Littoral. The municipality's main settlements are Miren

Choose site ...



Panoramio Photos are copyrighted by their owners

Voltage Humidity Illuminance Pressure
Temperature VSN state Watchdog



Environmental monitoring



Main Page

Main Page

Data Access Portal

Store and access fieldsite information ⓘ

SwissEx Users' Area

Quick access to your project's data ⓘ

Demonstration Datasets

Open datasets demonstrating the SwissEx platform ⓘ



Fieldsite database: the semantic wiki provides a user interface to a metadata/data database

The Swiss Experiment Platform

*(generally known as Swiss Experiment or SwissEx)
A platform to enable real-time environmental experiments through wireless sensor networks and a common, modern, generic cyber-infrastructure. This infrastructure will be used to enable interdisciplinary environmental research, allowing scientists to work efficiently and collaboratively to find the key mechanisms in the triggering of natural hazards and to efficiently distribute the information to increase public awareness.*

Latest News

See the news page for more news

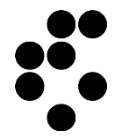
WSL/SLF Jubilee celebrations.

SwissEx and Hydrosys will be presented at the WSL/SLF Jubilee celebrations on Sunday 20th March, Bündawiese, Davos Dorf. Posted by Ndawes in **WSL/SLF**

Coastal flood prediction



<http://webgis1.geodata.soton.ac.uk/flood.html>



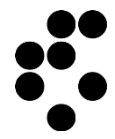
Academic: Distributed sensing infrastructure

Scientists defines their hypothesis, collect the necessary data and then try to validate the hypothesis. Manually collected data is generally expensive to get¹ while access to large datasets is generally restricted by the owners.

Solution:

- Deploy sensors in small and medium size testbeds
 - On a riverbed, volcano, mountain, etc.
- Build an open data publishing and sharing platform which can federate the testbeds
- Share your data with others so that also others share it with you

¹Matt Welsh, Sensor Network for the Sciences, Communications of the ACM, November 2010, Vol. 53, No. 11.



Use Cases: Alternative solutions

Several of the previously mentioned use cases can be solved by other approaches, crowdsourcing being one of the most obvious.

Roadify, Waze are using real time traffic information reported by participants in traffic may solve traffic congestion problems

- Human “sensor” reporting and consuming via handheld terminals
- Costs and benefits will determine the best solution.



Use Cases: Ethical issues and abuse¹

- For every technology created for a noble purpose, less noble applications can be found and vice-versa.
- Smart systems may be used for privacy invading applications, for restricting the liberty of people, for creating chaos, misinformation, false alarms, etc.

¹[Ludwig Siegele, A special report on smart systems, The Economist, Nov. 4 2010.](#)



Part I. Motivation & background outline



Web Of Things

- What is it? What problems can it solve?

Architectural considerations

- How it looks like? What are its components?

The “Things”

- What are the ingredients?

The “Glue”

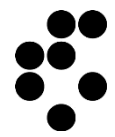
- How do things stick together?

Applications and services

- What can be built on top of it?

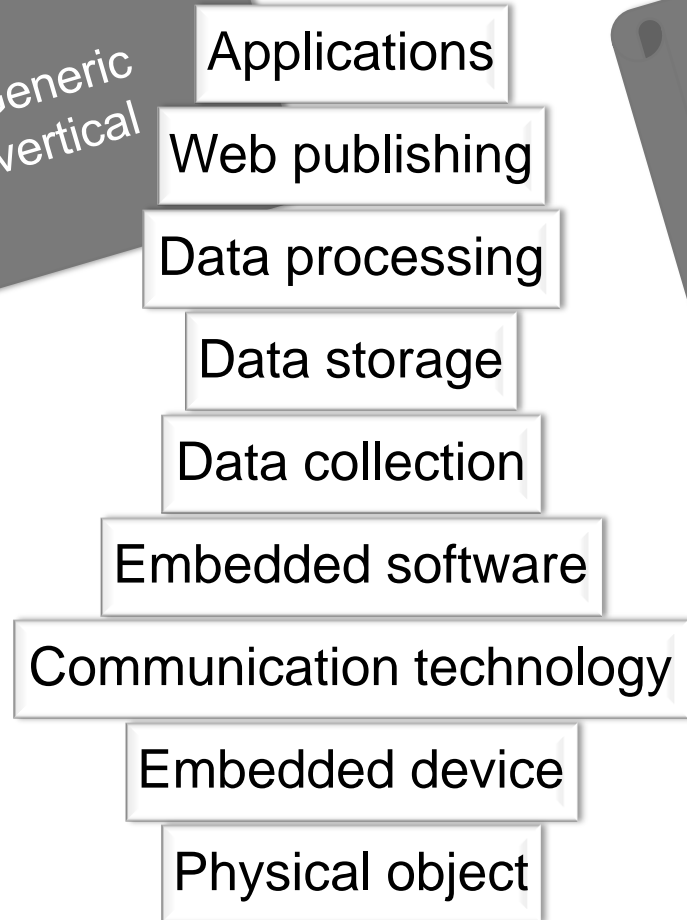
Quick start recipes

- How does the “Hello World!” look like?

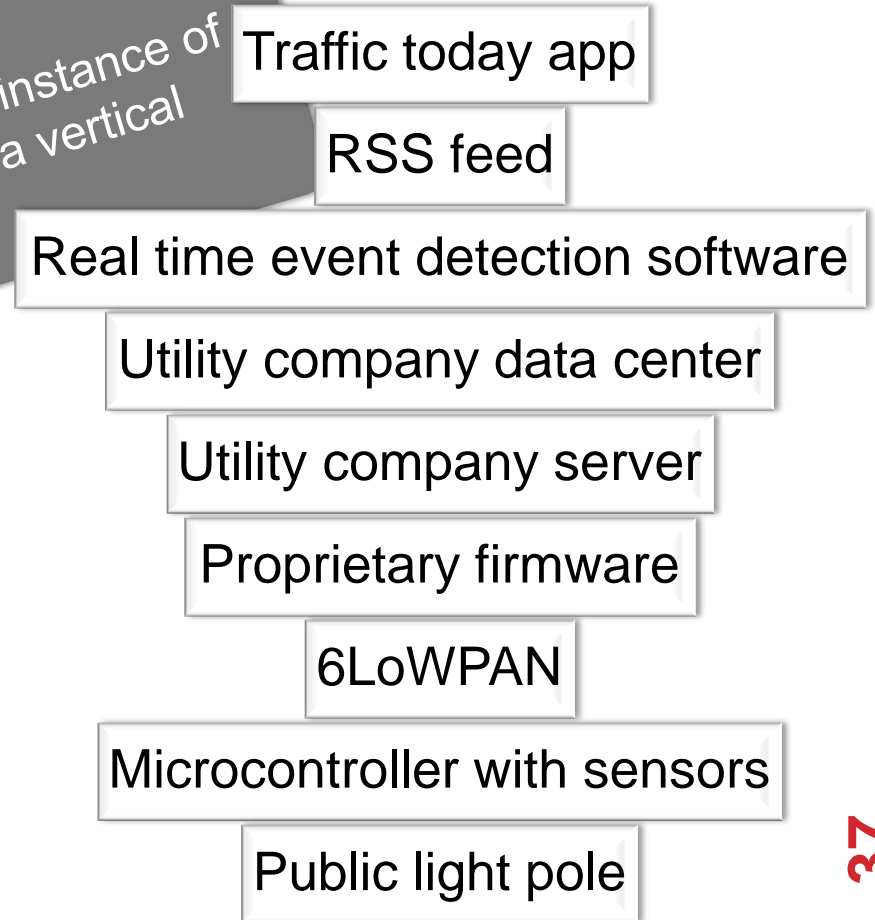


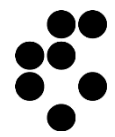
Architectural considerations

Generic vertical

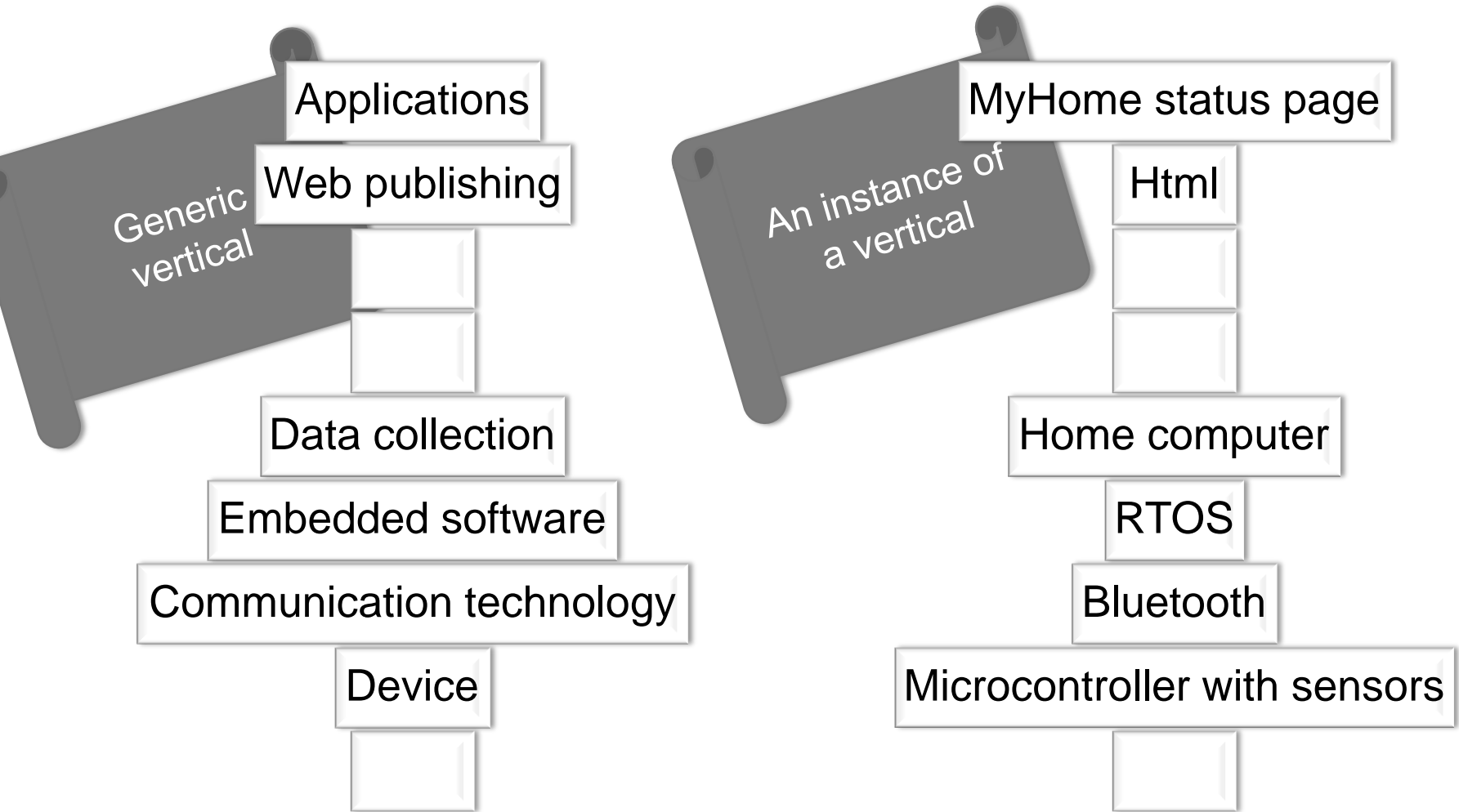


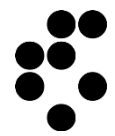
An instance of a vertical



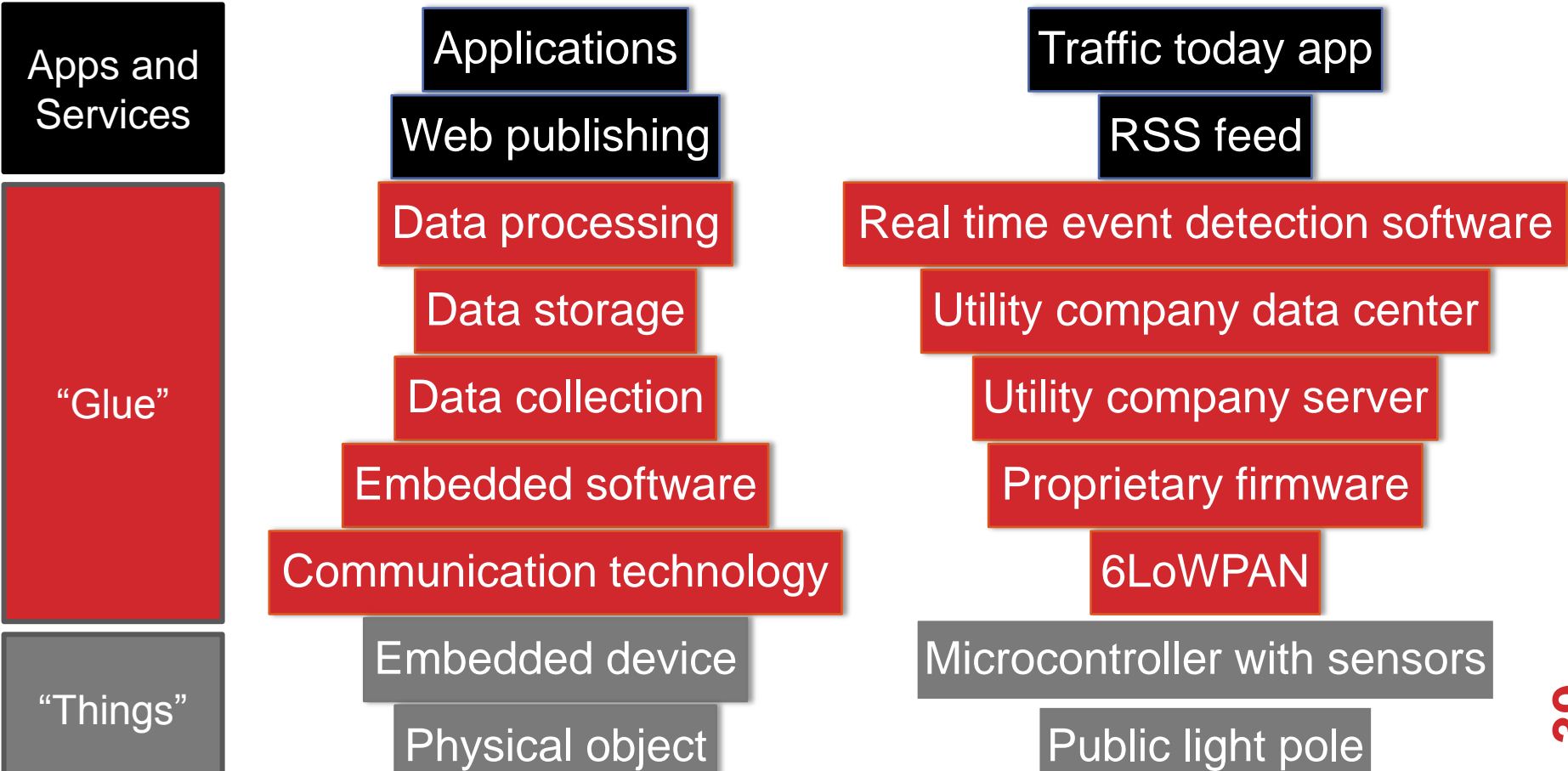


Architectural considerations





Main Components of a vertical





Part I. Motivation & background outline



Web Of Things

- What is it? What problems can it solve?



Architectural considerations

- How it looks like? What are its components?

The “Things”

- What are the ingredients?

The “Glue”

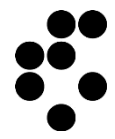
- How do things stick together?

Applications and services

- What can be built on top of it?

Quick start recipes

- How does the “Hello World!” look like?



The “things”

“Things”

Embedded device

Physical object

Microcontroller with sensors

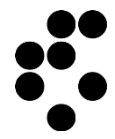
Public light pole

= embedded device + physical object (smart public light pole)

= sensor node (SunSpot, MicaZ, Sensinode, VESNA, WASPMote, etc)

= mobile phone

= a set of sensor nodes and/or embedded device + physical things which are abstracted as one “thing” (large water tank + set of sensor nodes monitoring water level, temperature and purity)



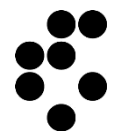
Definitions of components related to things

physical object

- An object built for fulfilling other tasks than computing
 - Coffee mug, show, light pole, washing machine, electric oven, fruit press, water tank

sensor

- a material or passive device which changes its (conductive) properties according to a physical stimulus
 - Thermo couple (temp->voltage), photo resistor (light->resistance variations), etc.



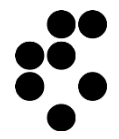
Definitions of components related to things

embedded system

- A simple or complex system built into a physical device to perform dedicated functions and enhance the functionality through computation. It features actuators and/or sensors.
 - Microprocessor, microcontroller, DSP, FPGA or PLC based system built into a variety devices, including washing machines, electric ovens, industrial robots etc.

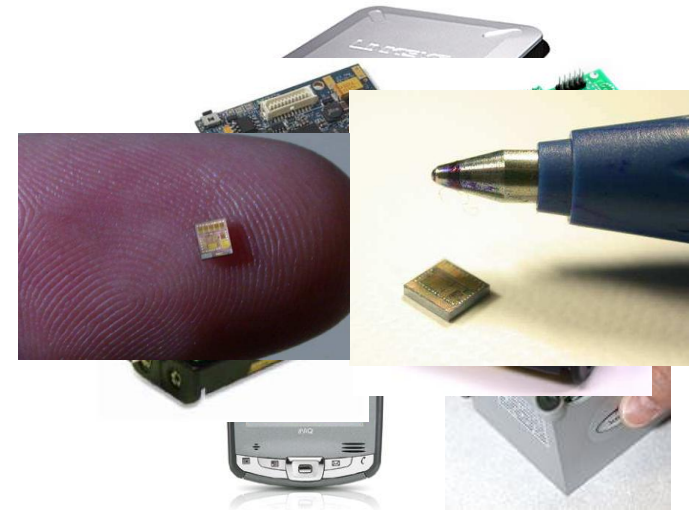
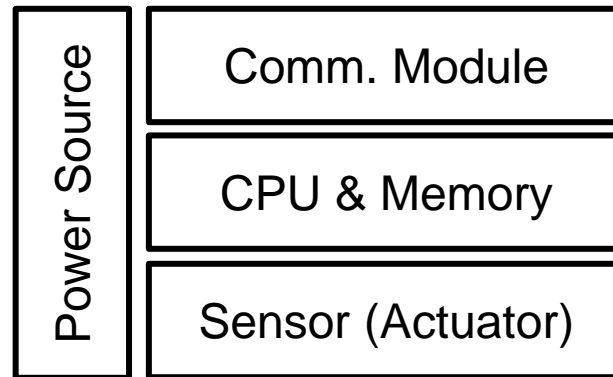
sensor node

- A computing and communicating device equipped with sensors and possibly actuators whose functionality revolves around measuring, reporting and possibly actuating. It can be standalone or embedded into physical objects.
 - Typically a device composed of microcontroller, power supply, communication interface and sensors/actuators.



Sensor nodes and their structure

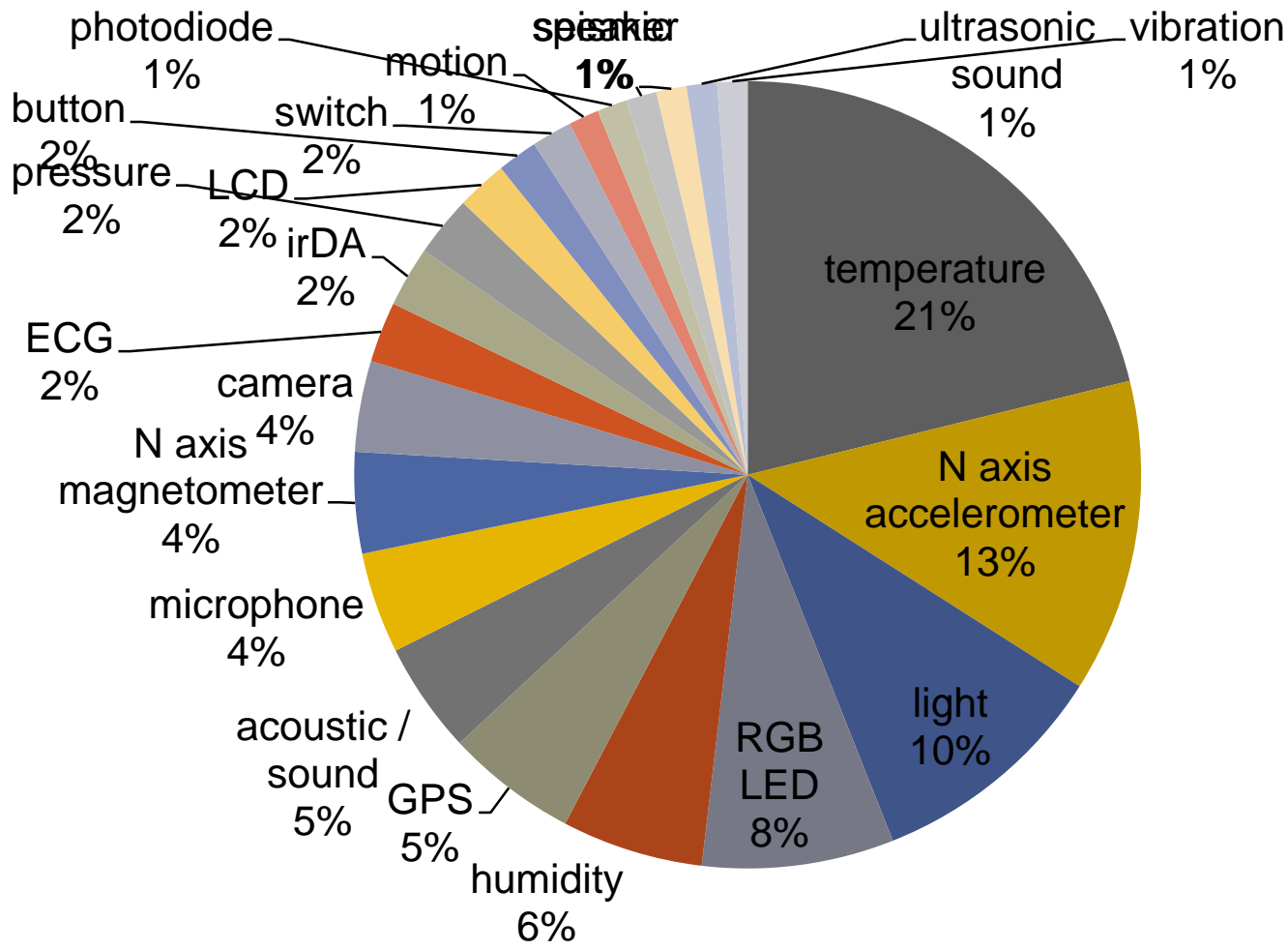
= Sensors + Microcontroller + Communication Module
+ Power Source

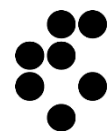


Classification:

- adapted/augmented general-purpose computers
- embedded sensor modules
- system on chip (SoC) solutions

Common types of sensors found in the literature





Existing solutions for sensor nodes

Solutions developed in **research community or by groups of enthusiasts.**

- Combine HW components from different produces (for radio, it seems that TI chips are used in vast majority of ,products').
- open-source experimental software such as Contiki OS, TinyOS (& NesC), Nano-RK, FreakZ stack (except for Arduino/Libelium where OEM radio is used whilst crowdsourcing is happening on the level of easy microcontroller programming.
- open source development tools are usually used.

Commercial solutions from particular producers (TI, Atmel, Microchip,...)

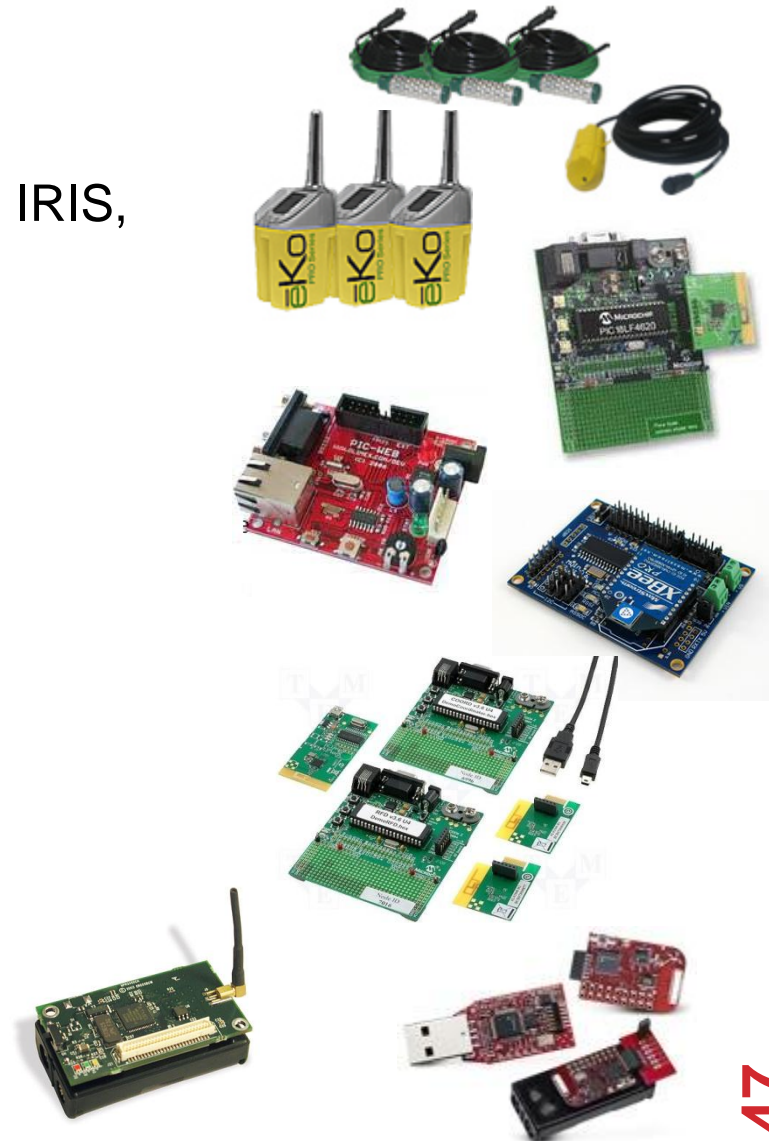
- composed of components sold by produces themselves.
- development kits can usually be used with proprietary integrated development environments and allow compiling of certified stacks (most often Zigbee).

Modules assembled by **companies trying to sell software solutions**

- Sun is in this case promoting the use of Java for sensor networks
- Sensinode is selling one of the 6LoWPAN ports.

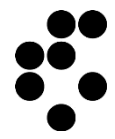
Examples of the three categories of solutions

- FreakLabs Chibi
- Memsic (ex. Crossbow) MICAz/ MICA2, IRIS, TelosB, eKo kit
- CMU FireFly
- GINA
- Arduino/Libellium (XBee)
- TI eZ430-RF2500
- Microchip PICDEM Z
- Atmel RZ600
- Ember InSight
- Jennic JN5148
- SunSPOT
- Sensinode
- NanoSensor



Developer friendly hardware solutions for WoT compared

Solution	Pro	Con	Cost
Arduino	big community, open-source hw, documentation	Computational power	25-90 €
Nanode	built-in web connectivity , open-source hw, Arduino compatible	Computational power, documentation	~ 35 €
openPICUS	built-in web connectivity , good documentation, support	Computational power, development windows oriented	~ 70 €
Netduino	open-source hw, documentation, arduino compatible, .NET programming	Computational power, development windows oriented	25-90 €
libelium	documentation, solid, radio boards, sensors boards, over the air programming, libelium support.	Computational power	~ 150 €

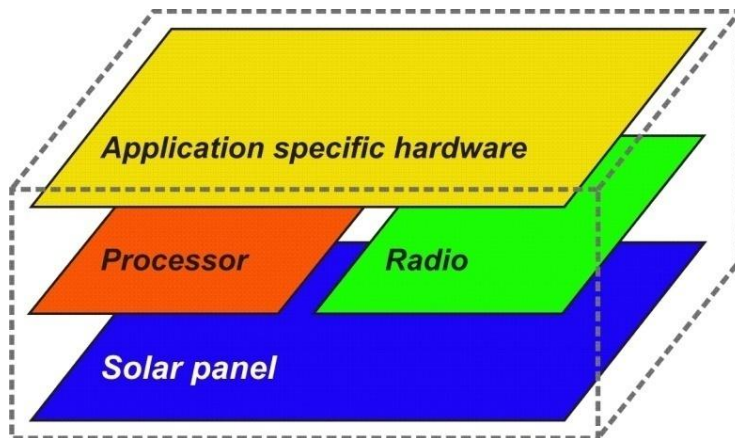


VESNA Sensor Node

Built at JSI, used for some of the demos presented in Part 3 of the tutorial.

Modular platform for wireless sensor networks (SNCore + SNRadio + SNApplication + SNPower = VESNA sensor node)

- High **processing power** and low **energy consumption**
- **Sensor node & gateway** (multi-tier / IP) capability
- **Battery, solar or external** power supply
- **Re-configurable radio**



VESNA Sensor Node

SN-Core

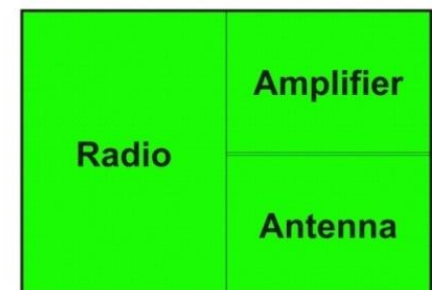
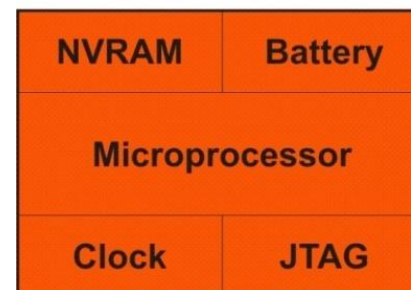
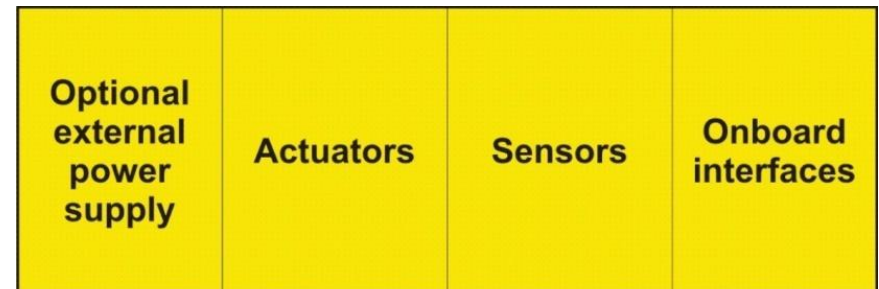
- Analog and digital sensor/actuator interfaces
- Possibility to use operating system (real-time, event-driven)
- Multiple expansion options
- Open C/C++ code libraries
- Onboard memory

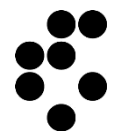
SN-Radio

- 300-900 MHz, 2.4 GHz radio interface (all ISM bands)
- ZigBee, 6LoWPAN and other IEEE 802.15.4 based solutions

SN-Expansion

- Bluetooth, Wi-Fi, Ethernet, GSM/GPRS
- Sensors/actuators
- PoE

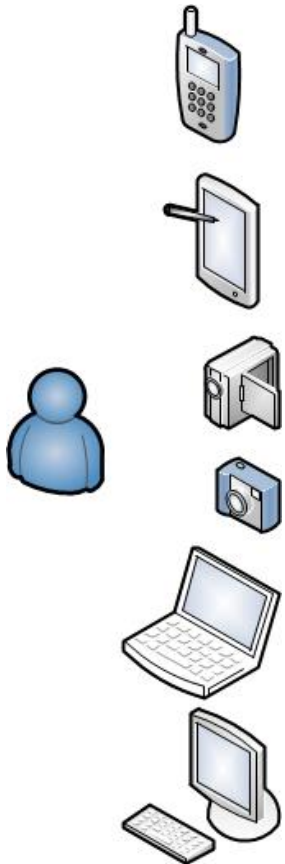




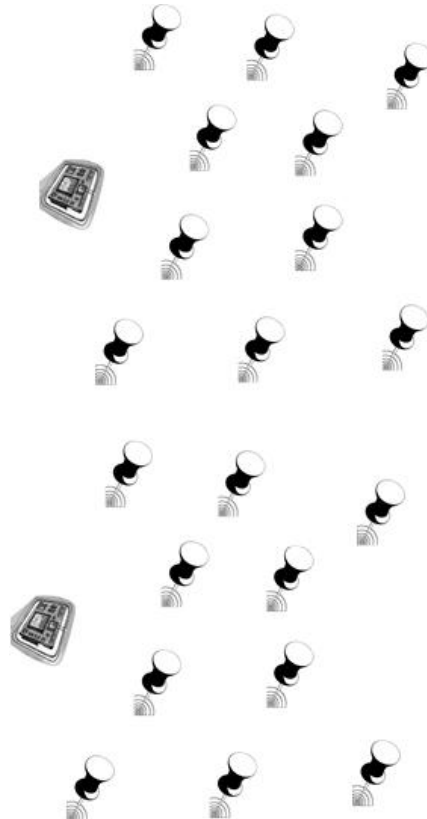
Why are sensor nodes different than other computing devices?

- Most SNs are **application specific**.
- Asymmetric, highly directional **information flow** (data fusion).
- **Energy** is highly constrained.
- Networks of SNs may have huge amount of **nodes**.
- Application **run-time** is extremely long.

Sensor nodes vs computing devices



Personal devices



Sensor nodes

Diminishing maintenance costs:

- Integrating sensors into personal computing devices such as phones/laptops
- Efficient remote configuration and management
- Disposable



Part I. Motivation & background outline

✓ Web Of Things

- What is it? What problems can it solve?

✓ Architectural considerations

- How it looks like? What are its components?

✓ The “Things”

- What are the ingredients?

The “Glue”

- How do things stick together?

Applications and services

- What can be built on top of it?

Quick start recipes

- How does the “Hello World!” look like?

The “glue”

The communication

- The communication medium
- The network

Node centric programming

- operating system
- virtual machine

System level programming (macro-programming)

- distributed/centralized storage and retrieval
- content management

“Glue”

Data processing

Data storage

Data collection

Embedded software

Communication technology

Real time event detection software

Utility company data center

Utility company server

Proprietary firmware

6LoWPAN

Communication medium

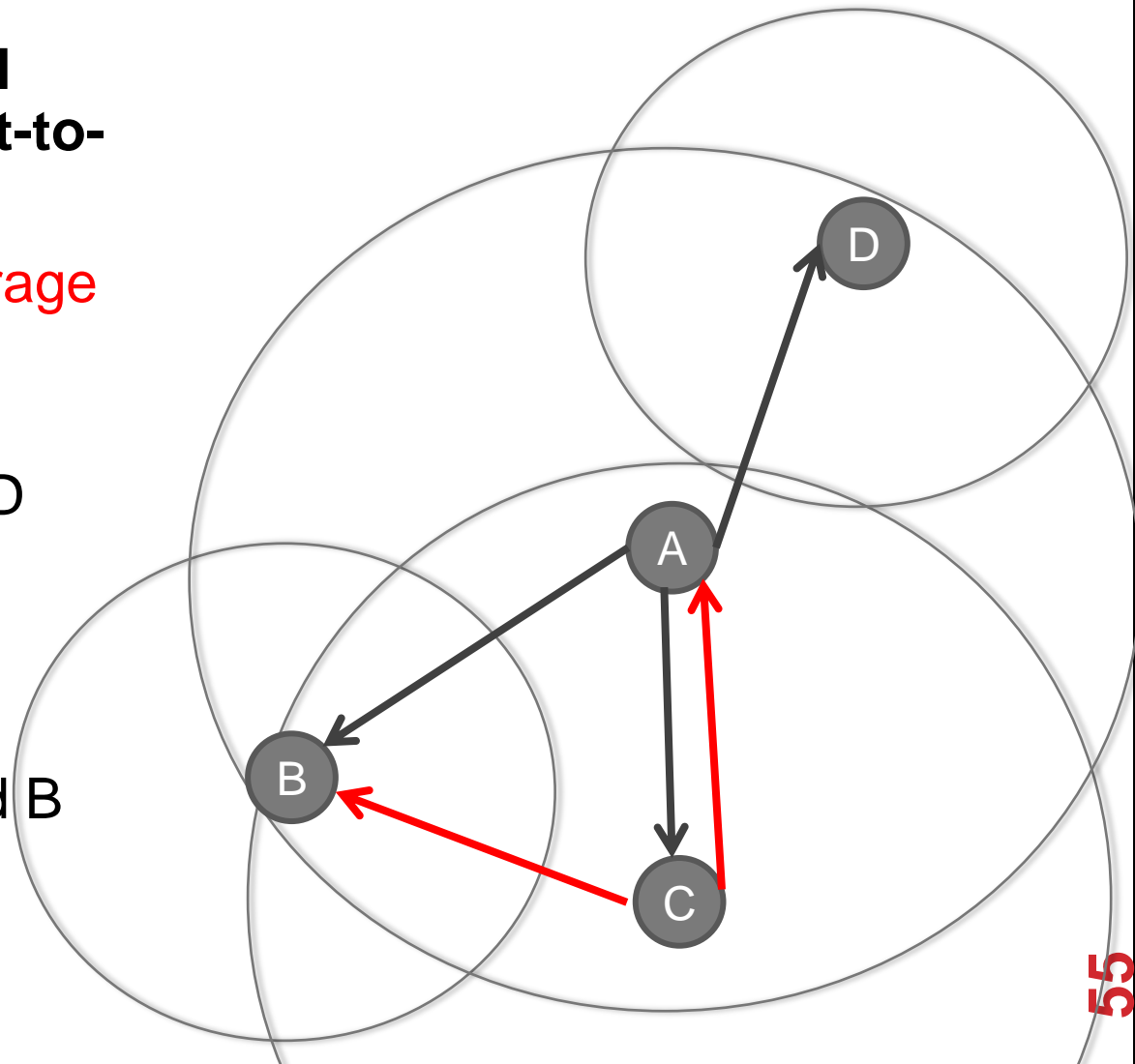
Wireless and/or Wired point-to-point or point-to- multipoint

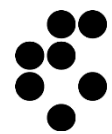
B, C and D in the **coverage range** of A

- When A sends a message, B, C and D receive it

A, B in the range of C

- When C sends a message, only A and B receive it

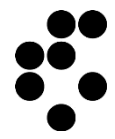




Communication medium: open wireless standards

Wireless

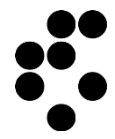
- Mostly performed in **unlicensed bands according to open standards**
 - Standard: IEEE 802.15.4 - Low Rate WPAN
 - 868/915 MHz bands with transfer rates of 20 and 40 kbit/s, 2450 MHz band with a rate of 250 kbit/s
 - Technology: **ZigBee, WirelessHART**
 - Standard: ISO/IEC 18000-7 (standard for active RFID)
 - 433 MHz unlicensed spectrum with transfer rates of 200 kbit/s
 - Technology: **Dash7**
 - Standard: IEEE 802.15.1 – High Rate WPAN
 - 2.40 GHz bands with transfer rates of 1-24 Mbit/s
 - Technology: **Bluetooth** (BT 3.0 Low Energy Mode)
 - Standard: IEEE 802.11x – WLAN
 - 2.4, 3.6 and 5 GHz with transfer rates 15-150 Mbit/s
 - Technology: **Wi-Fi**



Communication medium: licensed bands

Wireless

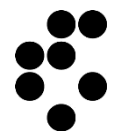
- Sometimes in **licensed bands**
 - Standard: 3GPP – WMAN, WWAN cellular communication
 - 950 MHz, 1.8 and 2.1 GHz bands with data rate ranging from 20 Kbit/s to 7.2 Mbit/s, depending on the release
 - Technology: **GPRS, HSPA**



Communication medium: proprietary standards

Wireless

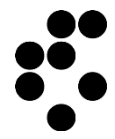
- Sometimes according to **proprietary standards and protocols**
 - **Z-Wave** – for home automation
 - 900 MHz band (partly overlaps with 900 MHz cellular) with data rates of 9.6 Kbit/s or 40 Kbit/s
 - **ANT** – for sportsmen and outdoor activity monitoring, owned by Garmin
 - 2.4 GHz and 1 Mbit/s data rates
 - **Wavenis** – for M2M periodic low data rate communication
 - 868 MHz, 915 MHz, 433 MHz with data rates from 4.8 Kbits/s to 100 Kbits/s
 - most Wavenis applications communicate at 19.2 kbits/s.
 - **MiWi, SimpliciTI, Digi xxx, ...**



Communication medium: standards for wired communication

Wired

- Standard: IEEE 1901 - Power Line Communications (PLC) standard used for transmitting data on a conductor also used for electric power transmission
 - Frequencies and data rates vary, >100 MHz, data rates of up to 500 Mbit/s
 - Technology: **HomePlug**
- Standard: ITU G.hn – PLC for home grids
 - 100-200 MHz with data rate up to 1 Mbit/s
 - Technology: **HomePNA**
- Standard: IEEE 802.3 – High speed LAN
 - 10 Mbit/s – 100 Gbit/s
 - Technology: **Ethernet**

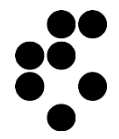


Communication technology: implementation aspects

Implementation of the technologies

- Traditionally HW
- Mostly HW + some SW
- Trend towards HW + mostly software

Communication Standard	Protocol Stack Implementation
IEEE 802.15.4	“Implementation of IEEE 802.15.4 protocol stack for Linux”
ZigBee	Z-Stack, Open-ZB, FreakZ, Microchip Stack
IEEE 802.11	smxWiFi
WirelessHART	“WirelessHART- Implementation and Evaluation on Wireless Sensors”, “WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control”
ISA100.11a	NISA100.11a
Bluetooth	TinyBT, Axis OpenBT, BlueZ, Affix

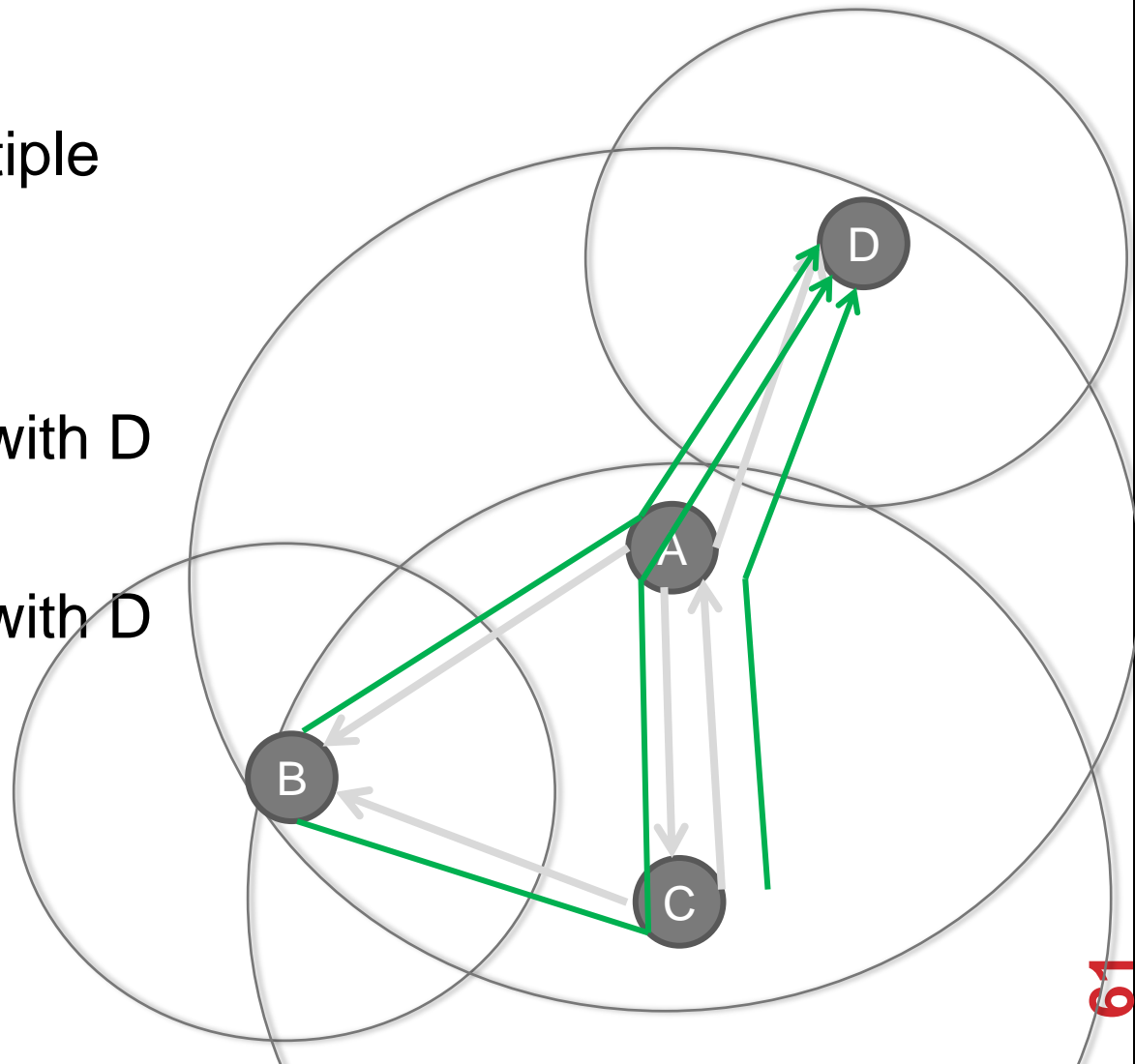


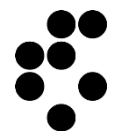
The network

The connections are **logical** (typically multiple physical hops).

C can communicate with D via A

C can communicate with D via A or via C and A

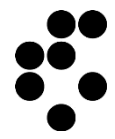




The network: abstraction

The network (or OSI Layer 3 abstraction) provides an abstraction of the physical world.

- Devices which are not physically “connected” via the communication medium can “talk” to each other
- At the network layer, only the devices and the links between them can be seen, the communication medium is hidden
- Communication protocol
 - defines the functions that have to be implemented and services that have to be provided by the parties involved in the information exchange.
 - In computer and sensor networks, protocols are organized as a stack and the number of layers in the stack is standard specific.



The network: standards

Global network level standard: IPv4 (towards IPv6)

Also a version for low power devices exists: 6LoWPAN.

It is unlikely that all things will eventually use a version of IP

- We foresee island of things implementing some kind of network layer protocol
 - Centralized: a central sink node collects all the data coming from the “things” of the network
 - Decentralized: Data aggregation is performed locally at each “thing” using only the measurements coming from neighbouring “things”
 - Hierarchical: Nodes are divided in hierarchical levels. Data move from the lower levels (sensor nodes) to the higher ones (sink nodes)
- Android@home is a good example of this
- The islands will be connected at higher levels of abstraction

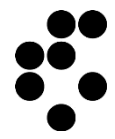


The network: implementation aspects

Implementation of the technologies

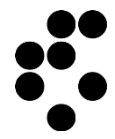
- SW
 - On the microcontroller on the communication interface (system on chip (SoC) and CPU+ OEM radio devices)
 - On the device's CPU (Microcontroller + PHY/MAC Radio devices)
- Stand alone protocol stack vs compatible/integrated with the OS

Communication Standard	Protocol Stack Implementation
ZigBee	Z-Stack, Open-ZB, FreakZ, Microchip Stack
6LoWPAN	NanoStack2.0, Mantus, μ IPv6, BLIP



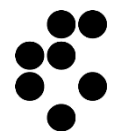
Node centric abstractions

- **Operating System (OS)**
 - abstracts task synchronization and memory management among others from the programmer
- **Virtual Machine (VM)**
 - another level of abstraction which further hides hardware specific issues from the programmer, for instance abstracting while loops with listeners



Embedded Operating system

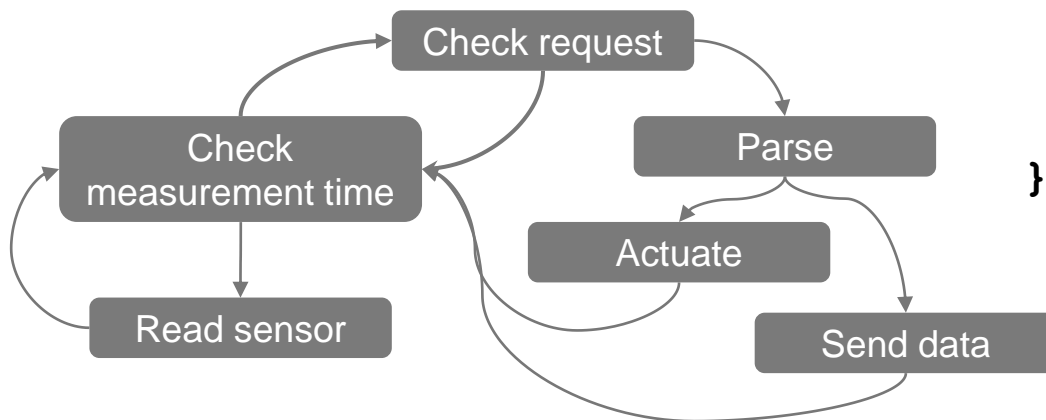
- **OS running on devices with restricted functionality**
 - In the case of sensor nodes, these devices typically also have limited processing capability
- **Restricted to narrow applications**
 - industrial controllers, robots, networking gear, gaming consoles, metering, sensor nodes...
- **Architecture and purpose of embedded OS changes as the hardware capabilities change (i.e. mobile phones)**



Embedded Operating system

Abstracts the hardware configuration, task synchronization and memory management

- Example: web service with one sensor and one actuator
 - Data from the sensor can be requested
 - Actuator can be commanded
- Without an OS, all the program states have to be manually prepared



```
while(1) {  
    if(got_request()) {  
        parse_request();  
        if(got_data_request())  
            send_data();  
        if(got_command())  
            actuate();  
    }  
    if(time_for_measurement())  
        read_sensor();  
}
```

Embedded Operating system

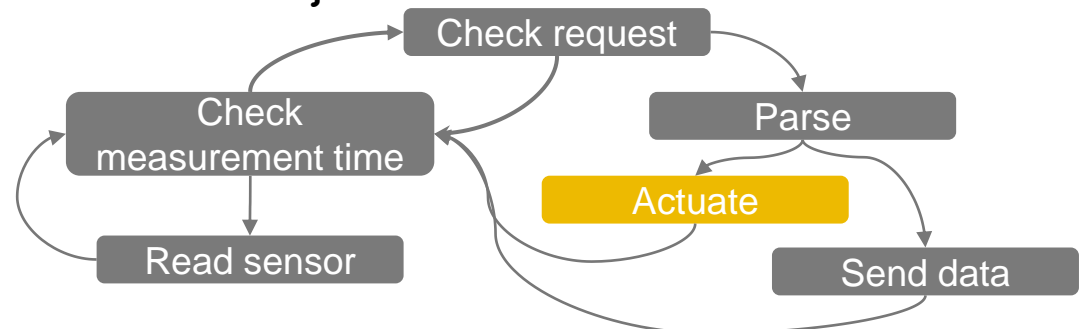
Abstracts the hardware configuration, task synchronization and memory management

- What happens if the actuate function takes too much time?
 - The system won't respond to requests

Example code:

```
void actuate() {
    actuate_start();
    delay(1000);
    send_command1();
    delay(1000);
    send_command2();
    delay(1000);
    send_command3();
}
```

```
while(1) {
    if(got_request()) {
        parse_request();
        if(got_data_request())
            send_data();
        if(got_command())
            actuate();
    }
    if(time_for_measurement())
        read_sensor();
}
```

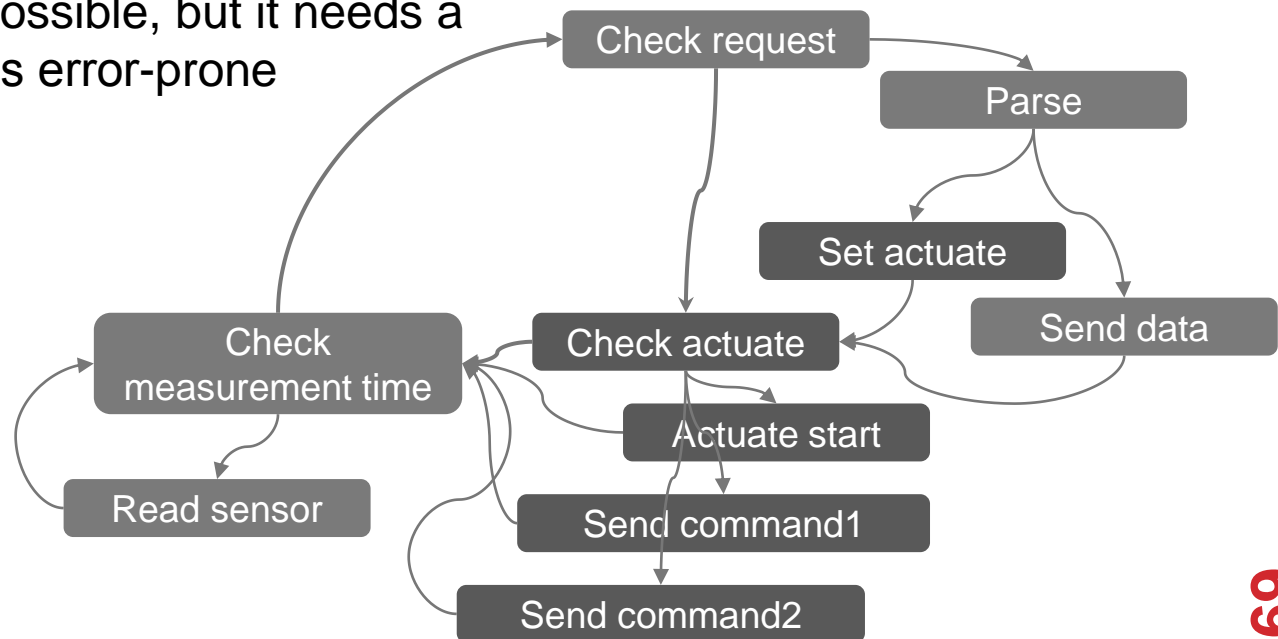


Embedded Operating system

Abstracts the hardware configuration, task synchronization and memory management

- What happens if the actuate function takes too much time?
 - The system won't respond to requests
 - Workaround is possible, but it needs a lot of effort and it's error-prone

```
void actuate() {
  actuate_start();
  delay(1000);
  send_command1();
  delay(1000);
  send_command2();
  delay(1000);
  send_command3();
}
```



Embedded Operating system

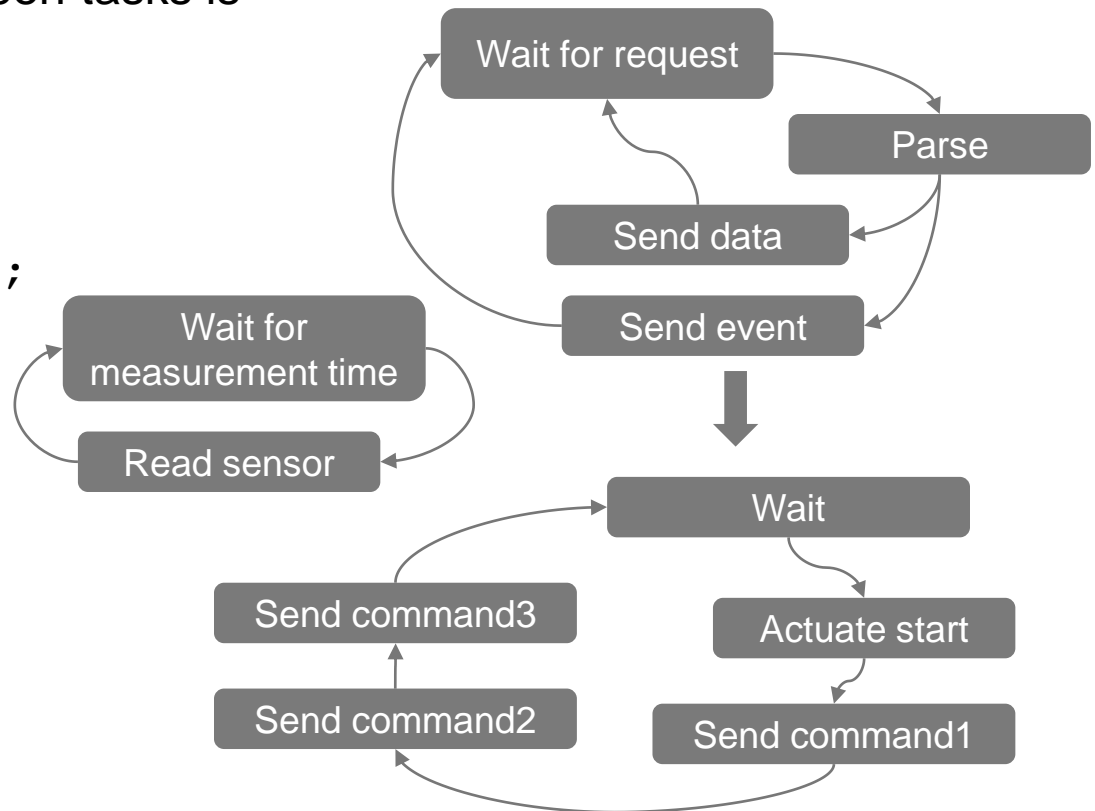
Abstracts the hardware configuration, task synchronization and memory management

- With an OS, switching between tasks is simple

```

PROCESS_THREAD(actuate) {
  while(1) {
    PROCESS_WAIT_EVENT();
    actuate_init();
    PROCESS_WAIT(1000);
    send_command1();
    PROCESS_WAIT(1000);
    send_command2();
    PROCESS_WAIT(1000);
    send_command3();
  }
}

```



Embedded OS Classification by scheduling model

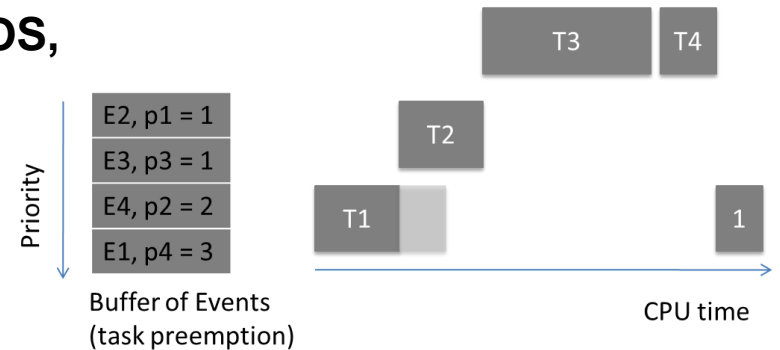
Event driven model (Contiki, TinyOS, SOS)

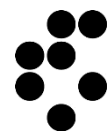
- No locking - only one event running at a time
- One stack – reused for every event handler
- Requires less memory
- Synchronous vs. asynchronous events



Thread driven model (FreeRTOS, eCOS, Nut/OS, eCOS)

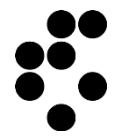
- Each thread has its own stack
- Thread stacks allocated at creation time (Unused stack space wastes memory)
- Locking mechanisms - to prevent modifying shared resources





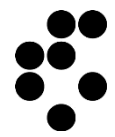
Embedded OS classification by system image

- **Monolithic (TinyOS, FreeRTOS, eCOS, uC/OS-II, Nut/OS)**
 - One system image : (kernel) + modules + application compiled together
 - Efficient execution environment (optimization at compilation)
 - High energy costs for updating
- **Modular (Contiki, SOS)**
 - Static image: (kernel) + loadable component images
 - Lower execution efficiency (no global optimization at compilation time)
 - Updates are less expensive (smaller size) - energy and time



Embedded OS comparison

Name	Sched.	Mem. Mgmt.	Kernel	Image/Re (programming)	Foot print	Protocol stack	VM	Dev. status/ reliability	Doc and supp
eCOS	Thread, preempt	Multiple stacks, static	Yes	Monolithic, no	variable	lwIP, FreeBSD TCP/IP	(yes)	Yes/(yes)	yes
uC/OS-II	Thread, preempt	Multiple stacks, static	Yes	(Monolithic, no)	variable	uC/TCP-IP	(no)	Yes/(yes)	limited
FreeRTOS	Thread, preempt	(Multiple stacks, static)	Yes	(Monolithic, no)	variable	lwIP	(no)	Yes/(yes)	yes
Nut/OS	Thread, preempt	Multiple stack, Dynamic	Yes	Monolithic, (no)	variable	BTNut, Nut/Net (TCP/IP)	(no)	Yes/(yes)	limited
TinyOS	Event, (thread)	Single stack, Static	No	Monolithic, wireless	variable	CC100, CC2420, TinyBt, serial	yes	Yes/yes	yes
SOS	Event	Single, dynamic	Yes	Modular,	variable	message	(no)	No/no	limited
Contiki	Event, thread		Yes	Modular, wireless	variable		yes	Yes/(no)	yes

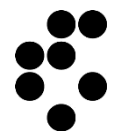


Virtual machine

A virtual machine is a software implementation of a machine and provides a level of abstraction over the physical machine.

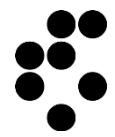
VM for embedded systems

- Replace the operating system
- Add extra functionality to the operating system (memory management)
- Provide a friendlier application development environment



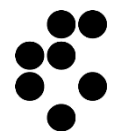
Classification of virtual machines

- **System VM**
 - virtualize hardware resources and can run directly on hardware.
 - In embedded systems they implement functions of the OS and completely replace it
 - Squawk, .NET Micro
- **Application (process) VM**
 - typically run on top of an OS as an application and support a single process.
 - Mate, Darjeeling, VM*, SwissQM, CVM, DVM



Virtual machine comparison

Name	OS	ASVM	Platform	Memory	Multi thread	Supported PL
Mate	TinyOS	no	Rene2 and Mica	1KB RAM 16KB ROM 7.5KB ROM, 600B RAM	yes	TinyScript
.NET Micro	With/without OS TinyOS	No ?	Imote2	300KB RAM 512MB of flash memory	yes	C#
Darjeeling	TinyOS, Contiki, FOS	No (?)	Tnode,Tmote Sky, Fleck3/Fleck3B	2K RAM	Yes	Java subset
Squawk	With (Solaris, Windows, MAC OS X, linux systems) /without	Yes ?	SunSpots Java Card 3.0	Core 80KB RAM Libraries: 270 Kb flash	yes	Java mostly
VM*	OS*	yes	Mica, ongoing work: Telos, XYZ, Stargate and handheld devices	6kb code 200 bytes data (depends on the app req)	no	Java
SwissQM	TinyOS	Yes	Mica2 and tmote sky	33kb flash and 3kb SRAM(on Mica2)	Yes	Subset of Java (37 instructions + 22 specific)
CVM	Contiki			8 RAM 1344 ROM		



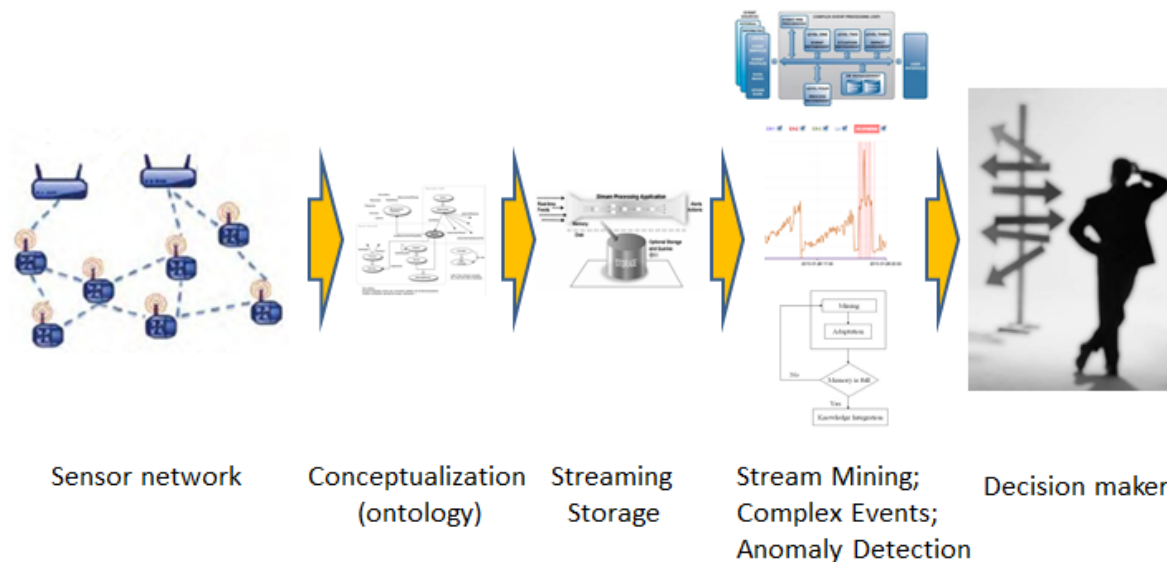
Are virtual machines necessary?

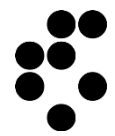
- **Trade-off between the resources needed and the services they provide**
- **Advantage**
 - Reduce the distribution energy costs for software updates
 - VM code smaller than native machine code
 - Simpler reprogramming process
- **Disadvantage**
 - Additional overhead
 - Increased time and memory requirements for execution
 - Increased energy spent in interpreting the code

What happens with the data?

Macro-programming abstractions

- Data, once generated, serve decision makers to understand the observed environment
- To bring the data to the decision maker, we need several macro programming abstractions
 - ...this includes technologies like: conceptualization, storage, stream mining, complex event detection,





Macro-programming abstractions

- **Semantic streams**
 - K. Whitehouse, F. Zhao, J. Liu, *Semantic Streams: a Framework for Composable Semantic Interpretation of Sensor Data*, 2005.
- **TinyDB**
 - A Declarative Database for Sensor Networks, <http://telegraph.cs.berkeley.edu/tinydb/>
- **Logical neighbourhoods**
 - L. Mottola, *Programming Wireless Sensor Networks: From Physical to Logical Neighborhoods*, PhD Thesis, 2009.



Part I. Motivation & background outline

✓ Web Of Things

- What is it? What problems can it solve?

✓ Architectural considerations

- How it looks like? What are its components?

✓ The “Things”

- What are the ingredients?

✓ The “Glue”

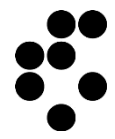
- How do things stick together?

Applications and services

- What can be built on top of it?

Quick start recipes

- How does the “Hello World!” look like?



Apps and services



Combine data, presentation or functionality from several sources (mash-up) to create new services.

Things generate only part of the data sources

Selected demos shown throughout the presentation.



Part I. Motivation & background outline

✓ Web Of Things

- What is it? What problems can it solve?

✓ Architectural considerations

- How it looks like? What are its components?

✓ The “Things”

- What are the ingredients?

✓ The “Glue”

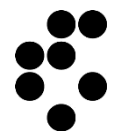
- How do things stick together?

✓ Applications and services

- What can be built on top of it?

Quick start recipes

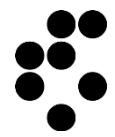
- How does the “Hello World!” look like?



Programming the “things”

Complex and time consuming process

- tool chain: IDE, compiler, debugger
- microcontroller is programmed and executes the code
- radio chip is not programmed, but controlled by microcontroller, usually via SPI which sets/reads registers
- compiled code is loaded to the microcontroller using bootloader or JTAG
- protocol stack may be precompiled and available through API or available as library
- operating system (not needed for simple tasks)
- virtual machine (optional)



Decision process

Before starting, the following questions should be answered:

What is the scope or application?

- Monitoring measurements?

What is the scenario?

- A thing with embedded web service?
- A set of things connected through a gateway?

What programming language?

- Options: C, nesC, Java, C#

What is the publishing infrastructure?

- None, custom, third party.

Embedded web service

What is the scope?

- Expose measurements as embedded web service.

What is the scenario?

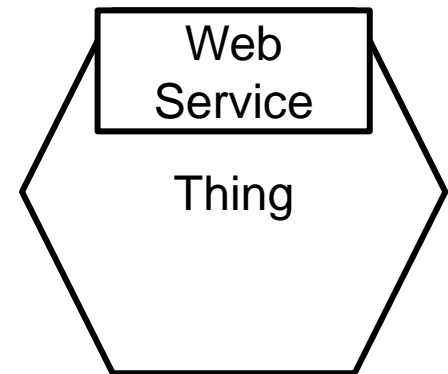
- SunSpot with embedded web service.

What programming language?

- Java.

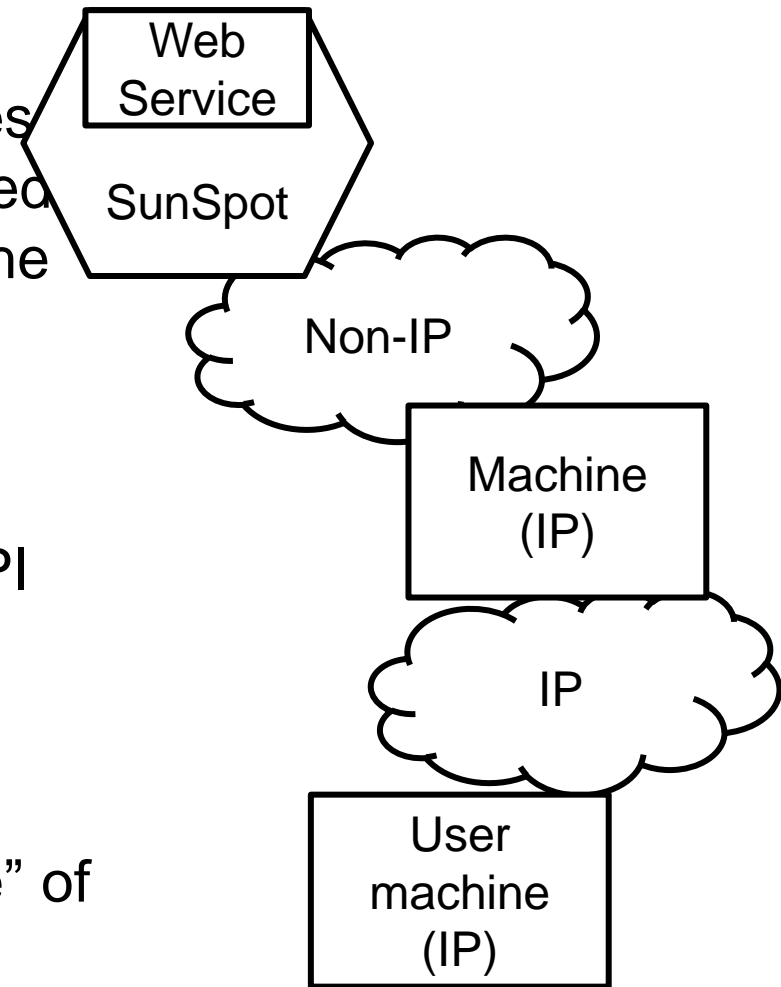
What is the publishing infrastructure?

- None.



Embedded web service

- the easiest implementation assumes connecting the thing to an IP enabled machine through which access to the embedded web service can be provided from the internet
- Direct integration via HTTP/TCP/IP possible starting from SunSPOT API V6.0
- Request:
<http://.../spot1/sensors/temperature>
 requests the resource “temperature” of the resource “sensor” of “spot1”¹



GET <http://IP:Port/service>

¹D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio, **Interacting with the SOA-based Internet of Things: Discovery, Query, Selection and On-Demand Provisioning of Web Services**, IEEE Transactions on Services Computing, Vol. 3, July-Sept 2010.

Publish measurements

What is the scope?

- Publish measurements on the web.

What is the scenario?

- SunSpot.
- Pachube account, registered feed, API key.

What programming language?

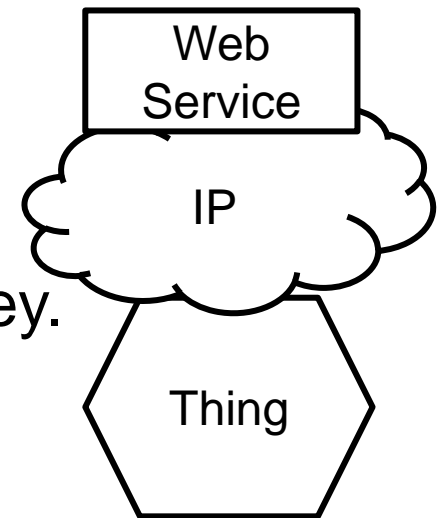
- Java.

What is the publishing infrastructure?

- Pachube.

SunSpot, <http://www.sunspotworld.com/>

Pachube, <http://www.pachube.com/>



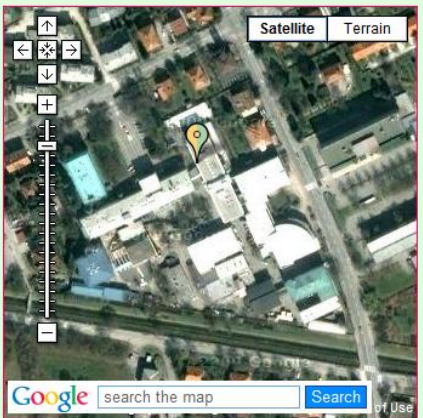
Sunspot + Pachube

output - use a feed



SunSPOT 2

<http://www.pachube.com/api/feeds/6894.xml>
<http://www.pachube.com/api/feeds/6894.csv>
<http://www.pachube.com/api/feeds/6894.json>
 Data updated: Tue May 04 14:11:45 UTC 2010, currently: frozen.
 Published by: carolinmap
 Website: <http://www.pachube.com/feeds/6894>

Share / Save



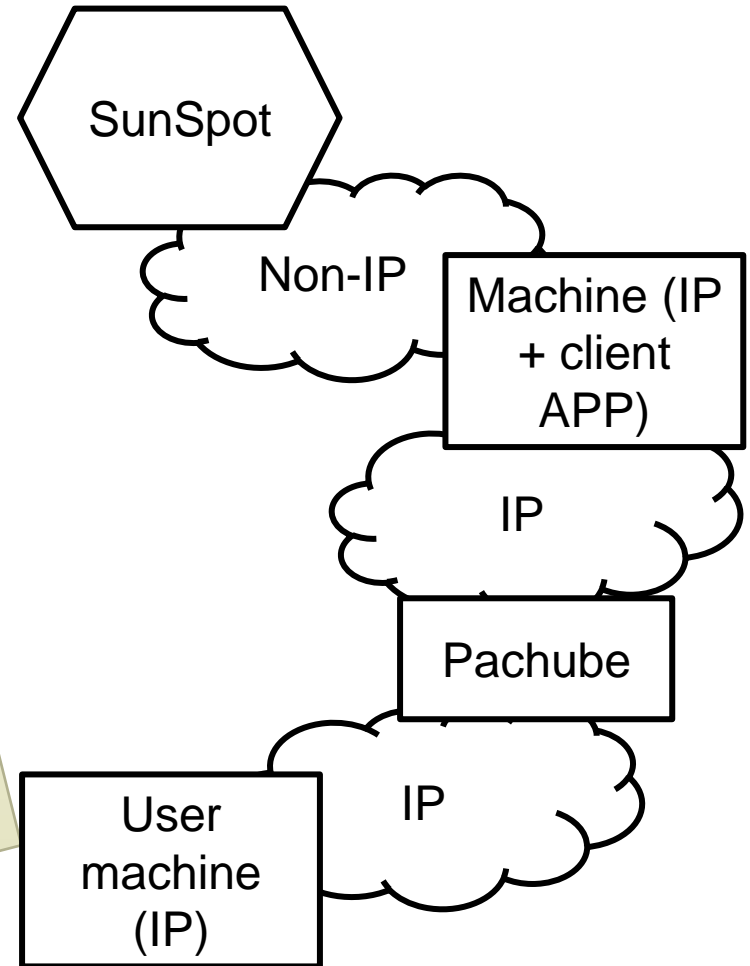
Location name: WSN SensorLab.
Domain: physical, Exposure: indoor

ID	Tags	Value	Units	24 hr History
0	Temperature	25.0	Celsius (°C)	 <small>3 months 4 days 24 hours last hour</small> <small>embed, history, triggers, etc</small>
1	Light	33	Percent (%)	 <small>3 months 4 days 24 hours last hour</small> <small>embed, history, triggers, etc</small>

Display

SunSpot, <http://www.sunspotworld.com/>

Pachube, <http://www.pachube.com/>



Publish measurements

What is the scope?

- Expose measurements as web service.

What is the scenario?

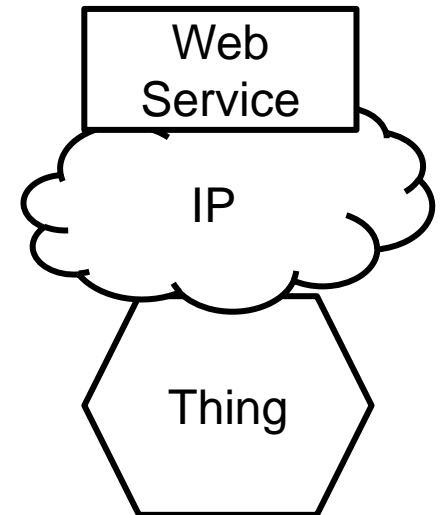
- Arduino Ethernet shield with board and sensors.
- Pachube account, registered feed, API key.

What programming language?

- C.

What is the publishing infrastructure?

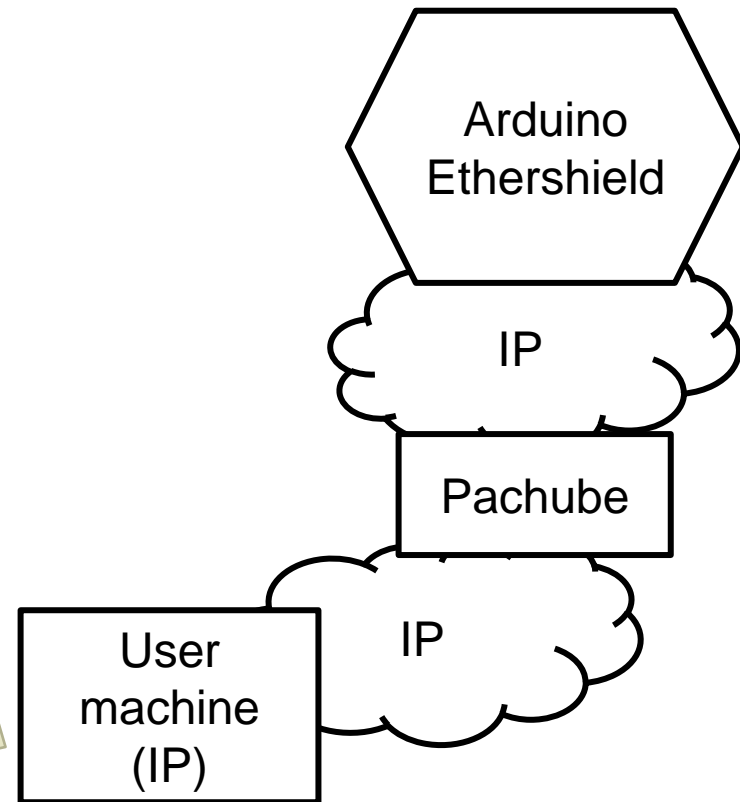
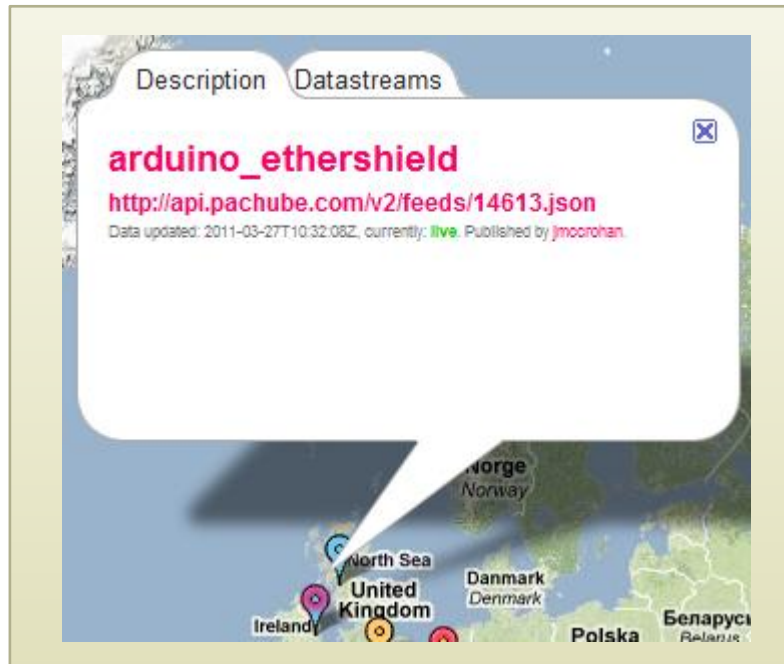
- Pachube.



Arduino, <http://www.arduino.cc/> Pachube, <http://www.pachube.com/>

Pachube Client on Arduino, <http://arduino.cc/en/Tutorial/PachubeClient>

Arduino + Pachube



Arduino, <http://www.arduino.cc/> Pachube, <http://www.pachube.com/>

Pachube Client on Arduino. <http://arduino.cc/en/Tutorial/PachubeClient>

Publish measurements

What is the scope?

- Publish measurements on the web.

What is the scenario?

- SunSpot (and/or Arduino).
- GSN installation and adequate wrapper.

What programming language?

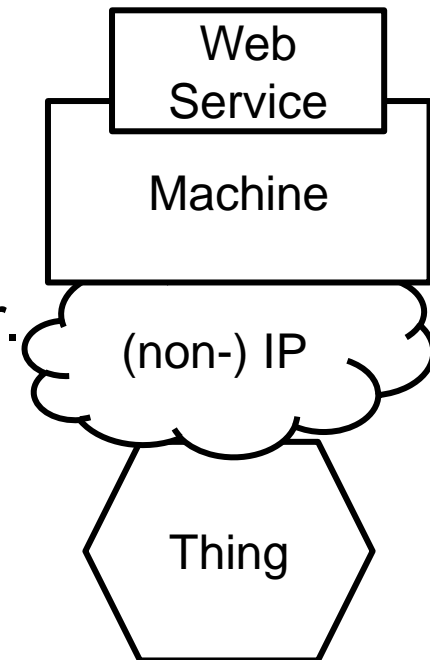
- Java (and/or C).

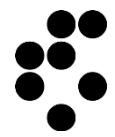
What is the publishing infrastructure?

- GSN.

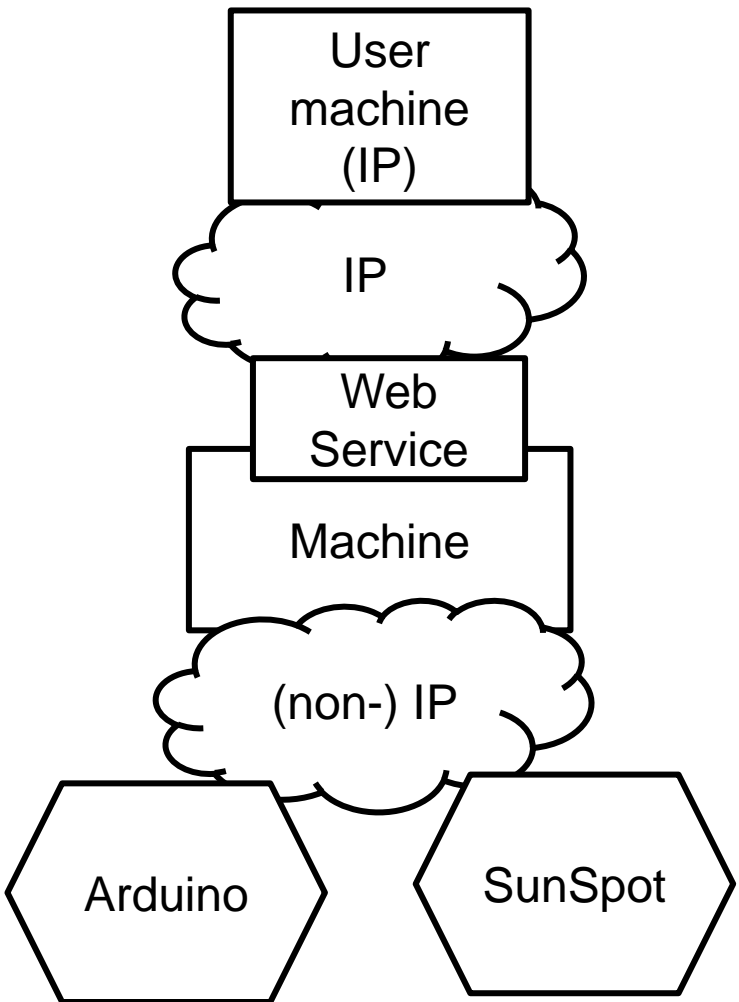
SunSpot, <http://www.sunspotworld.com/>


Global Sensor Network, <http://apps.sourceforge.net/trac/gsn/>





Thing + GSN



 **VSN Server :: GSN**

HOME DATA MAP FULLMAP

Welcome to Global Sensor Networks. The first ten sensors are displayed by default, but you can easily close them with the *close all* button. By clicking on a virtual sensors on the left sidebar, it will bring it to the top of the list.

Auto-refresh every : refresh close all

Real-Time	Addressing	Structure	Description	Download
miren_404780fd_2114 24/03/2011 18:45:34 +0100 ✕				
vsn_state 9				
battery_voltage 3491.0				
temperature_sht11 14.35				
temperature_scp1000 0.0				
humidity 2.63				
pressure 0.0				
luminance_1 0.36				
luminance_2 0.36				
farm_40668fe9 24/03/2011 18:20:52 +0100 ✕				
battery_voltage 3.86				
temperature_sht11 16.07				
temperature_scp1000 16.43				
humidity 62.73				
pressure 1018.94				

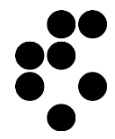
Description
VSN Public Lights T

Author
SensorLab, Jozef S (matevz(DOT)vucni

Virtual sensors

- + Group: farm
- + Group: kostar
- + Group: miren



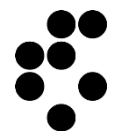


How to start?

Do you want to work with “things” that are under your direct control?

Things:

- Easy to use with a wide community support are Arduino and SunSpot.
- Crossbow, Libellium, Sensinode, etc. are also possible solutions but may require more effort.
- For very specific solutions you may need to go for hardware design.

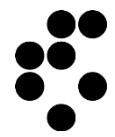


How to start?

Do you want to just publish the data?

Solutions:

- Pachube is straightforward but functionality is limited. There are various types of fee based accounts which offer additional functionality.
- GSN is free and can be customized, the code is open source.
- Nimbits promises to connect people, sensors and devices.
- Sensorpedia, Sensor.Network and other solutions are still in early stages or discontinued.



Summary

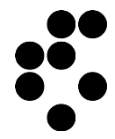
The concept of Web of Things was discussed

A list of relevant use cases and application areas

- Power grids
- Transport systems
- Water distribution
- Logistics
- Industrial automation
- Health
- Environmental intelligence

Architectural considerations and possible components of vertical systems were discussed and the components classified in:

- The “Things”
- The “Glue”
- Apps & Services



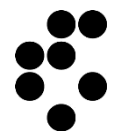
Summary

A decision process for how to start setting up a Web of Things system

- Programming things is time consuming

Quick start scenarios were presented in increasing order of complexity

- Things with embedded web services
- Things to an existing data management infrastructure
- Things to self deployed data management infrastructure

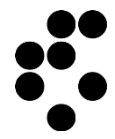


Outline

Part I. Motivation & background

Part II. Technology and tools for exploiting the WoT

Part III. Demos, Tools & Research directions



Part II. Technology and tools for WoT data

Information infrastructure for “Web of Things”

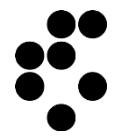
Conceptualization of sensors domain

Stream Data Processing

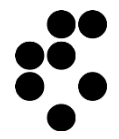
Stream Mining

Complex Event Processing

Anomaly Detection

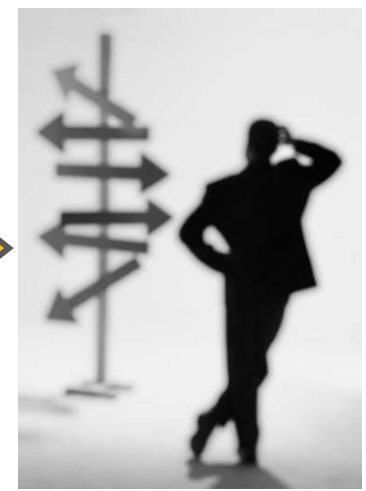
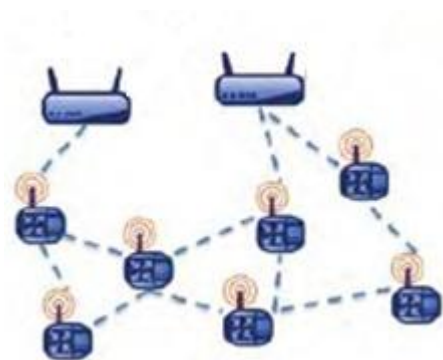


INFORMATION INFRASTRUCTURE FOR “WEB OF THINGS”



Why we need WoT?

...the key objective is to make decision maker more efficient by understanding observed environment



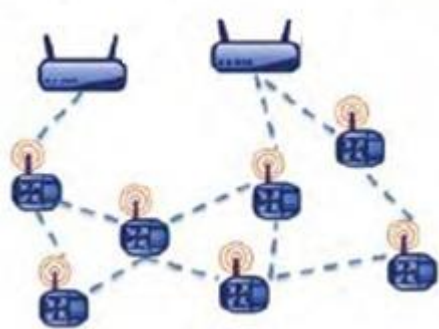
Sensor network

Decision maker

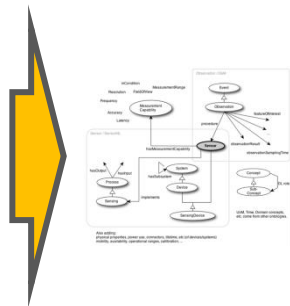
Why we need WoT?

...the key objective is to make decision maker more efficient by understanding observed environment

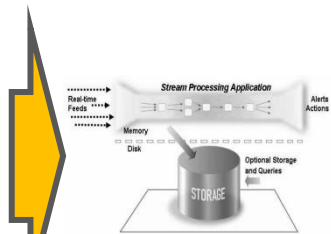
To achieve this, we need to introduce several information layers between sensor setup and decision maker:



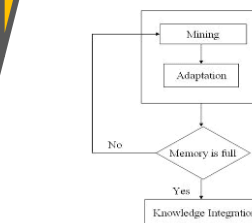
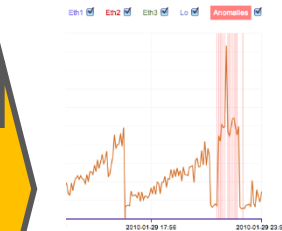
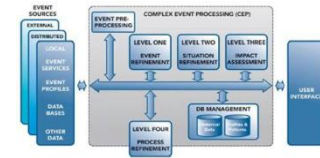
Sensor network



Conceptualization
(ontology)



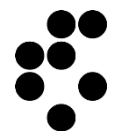
Streaming
Storage



Stream Mining;
Complex Events;
Anomaly Detection



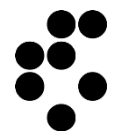
Decision maker



Outline of this part of the talk

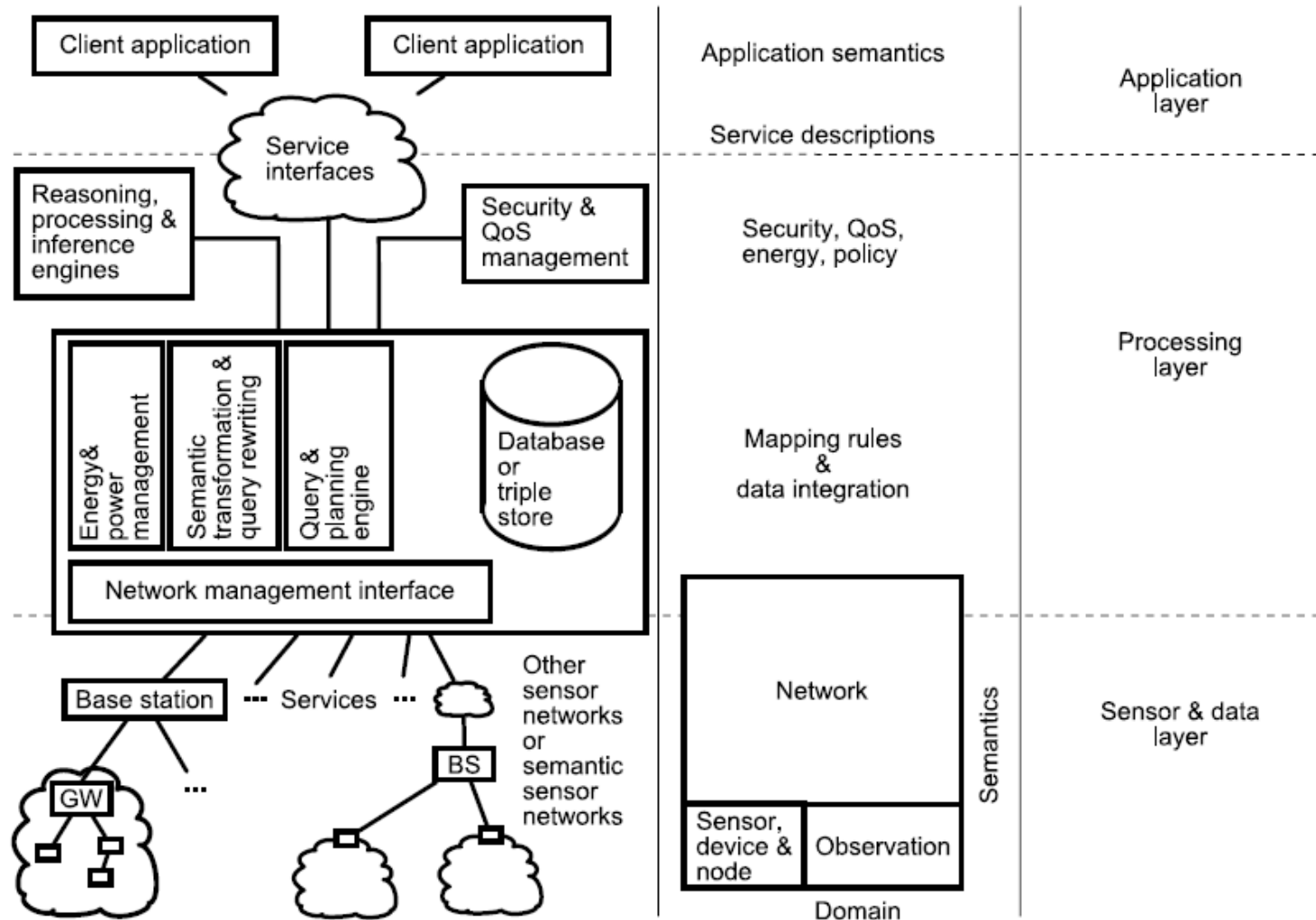
In this part we will review approaches on

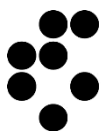
- How to conceptualize sensor domain?
- How to process streaming data?
 - How to detect complex events?
 - How to perform mining on streaming data?
 - How to detect anomalies?



CONCEPTUALIZATION OF SENSOR DOMAIN

Semantic Sensor Network architecture





Sensor ontologies

Several ontologies are covering sensor domain

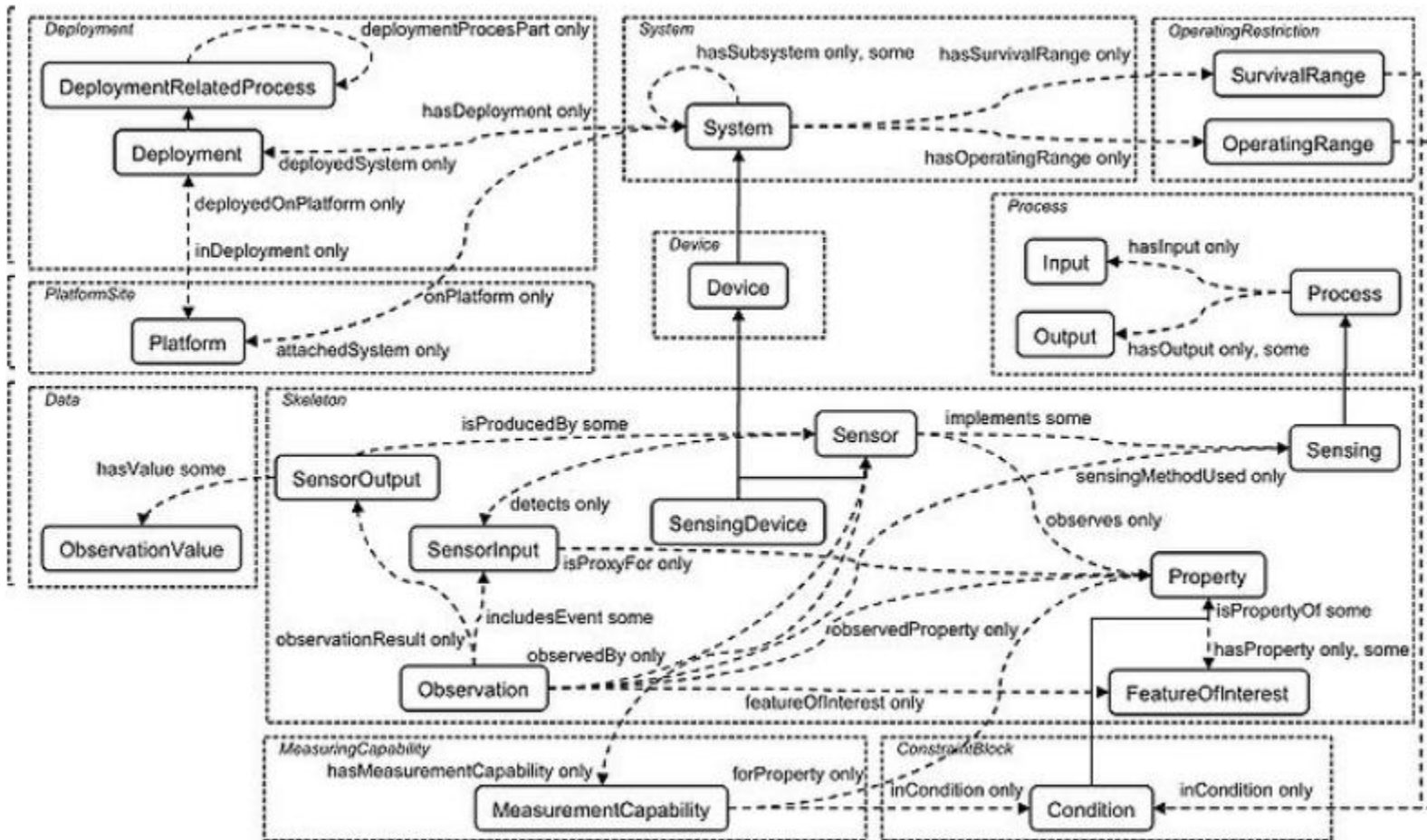
- ...most of them only parts

W3C Semantic Sensor Network (SSN)

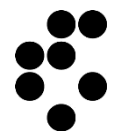
Ontology (next slide) is an attempt to cover complete domain

ontology	base concepts	sensor						physical			observation				domain							
		sensor hierarchy	identity & manufacturing	contacting & software deployment	configuration	history	components	action & process	location	power supply	platform	dimension, weight, etc.	operating conditions	data/observation	accuracy	frequency	response model	field of view/sensing	units of measurement	feature/quality	sampled medium	time
MMI	sensor (system) & process		✓		✓	✓		✓	✓	✓			✓	✓	✓				✓	✓		
CSIRO	sensor & process	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		
OOSTethys	component, system & process						✓	✓		✓			✓						✓	✓		
CESN	sensor	✓			✓			✓					✓						✓		✓	
SWAMO	agent, process & sensor			✓			✓	✓		✓			✓				✓	✓	✓	✓		✓
Kim	sensor					✓		✓					✓	✓	✓		✓		✓			
OntoSensor	component & sensor	✓			✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓		✓
Eid	sensor		✓			✓		✓	✓				✓	✓	✓	✓	✓					
Matheus	system & sensor	✓		✓			✓	✓					✓		✓							
Avancha	sensor			✓				✓	✓				✓	✓	✓	✓	✓	✓	✓	✓		✓
ISTAR																						

W3C Semantic Sensor Network (SSN) ontology structure



http://www.w3.org/2005/Incubator/ssn/wiki/Main_Page



So, how does a value look like?

Having all the semantic infrastructure in place, how an observed value is encoded in SSN?

```
<owl:Thing rdf:about="http://purl.oclc.org/NET/ssnx/product/smart-  
knife#WiTilt30MeasurementRange_3MinValue">  
  <rdf:type rdf:resource="http://purl.oclc.org/NET/ssnx/product/smart-knife#AccelerationValue"/>  
  <smart-knife:hasQuantityValue rdf:datatype="http://www.w3.org  
/2001/XMLSchema#float">-4</smart-knife:hasQuantityValue>  
</owl:Thing>
```

The observed value

Sensors, knowledge modeling and transliteration

Individual : [VicNode1](#) 1st CURE

on the term

isa : [ElectronicDevice](#)
isa : [Computer](#)
connectedTo : [virtualsensor3VicNode1](#) [TSL2561VicNode1](#) [sht11VicNode1](#)
 (connectedTo [scp1000VicNode1](#) [VicNode1](#))
 (connectedTo [virtualsensor2VicNode1](#) [VicNode1](#))
 (connectedTo [virtualsensor1VicNode1](#) [VicNode1](#))
 (deviceUsed [Testing](#) [VicNode1](#))
hasDevices : [virtualsensor3VicNode1](#) [TSL2561VicNode1](#) [scp1000VicNode1](#) [sht11VicNode1](#)
[virtualsensor2VicNode1](#) [virtualsensor1VicNode1](#)
latitude : [@/Degree-UnitOfAngularMeasure](#) 46.042873
longitude : [@/Degree-UnitOfAngularMeasure](#) 14.487469
nameString : "VicNode1"
objectFoundInLocation : [IndoorMounting](#) [Vic](#)
physicalParts : [virtualsensor3VicNode1](#) [TSL2561VicNode1](#) [scp1000VicNode1](#) [sht11VicNode1](#)
[virtualsensor2VicNode1](#) [virtualsensor1VicNode1](#)
(queryHasVeryHighPertinenceForThing [GetLinkToMap](#) [VicNode1](#))
supportedBy : [VicBuilding1](#)

Knowledge base



Transliteration

Device Identification Protocol



Sensors

Mash-up

Vič
Trnovo Bridge - Trnovo Bridge is a bridge crossing the river Gradaščica in Ljubljana, the capital of Slovenia. It connects the neighborhoods of Krakovo and Trnovo. A bridge had stood on the site since the late 17th

Choose site ...

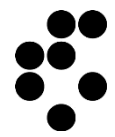
Panoramio Photos are copyrighted by their owners

Select by feature: **Air temperature**
 Voltage Camera Humidity Illuminance Pressure
 Voltage (solar) VSN state Watchdog Water level Water temperature

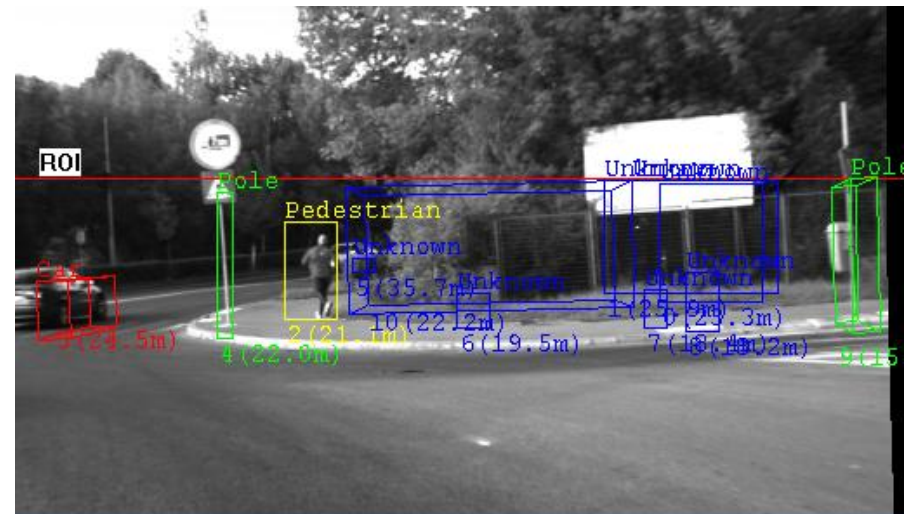
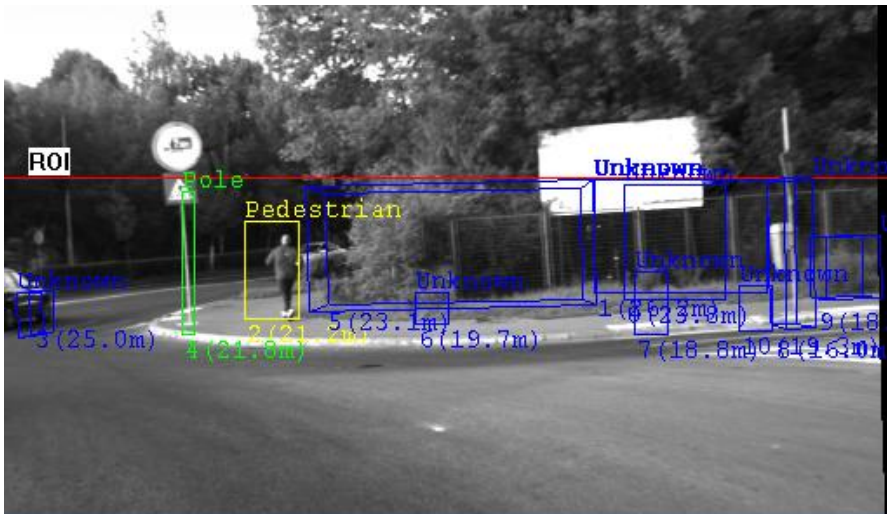
[02:03:04] Lights in Miren (6) are on.

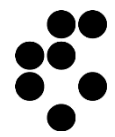
monitoring power consumption included the use of ReberNode3.

powered by Cyc



Traffic Scene Object Detection





Traffic Scene Understanding

QUESTION: Image depicts Person?

ANSWER: PEDESTRIAN2A000282 is a person.

QUESTION: Image depicts UtilityPole?

ANSWER: UNCLASSIFIED0A000282 is a utility pole. POLE4A000282 is a utility pole. UNCLASSIFIED6A000282 is a utility pole. UNCLASSIFIED7A000282 is a utility pole. UNCLASSIFIED8A000282 is a utility pole. UNCLASSIFIED9A000282 is a utility pole. UNCLASSIFIED10A000282 is a utility pole.

QUESTION: Image depicts Automobile?

ANSWER: UNCLASSIFIED3A000282 is a car.

QUESTION: Image depicts Person?

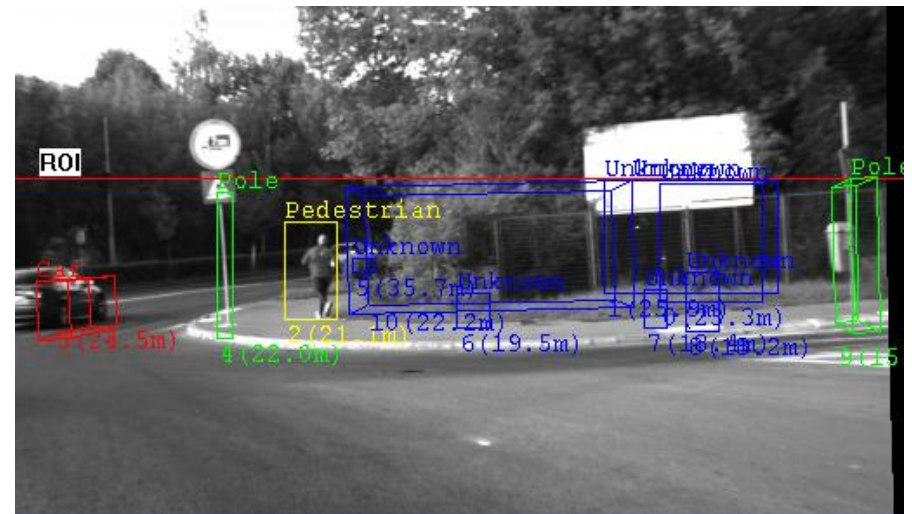
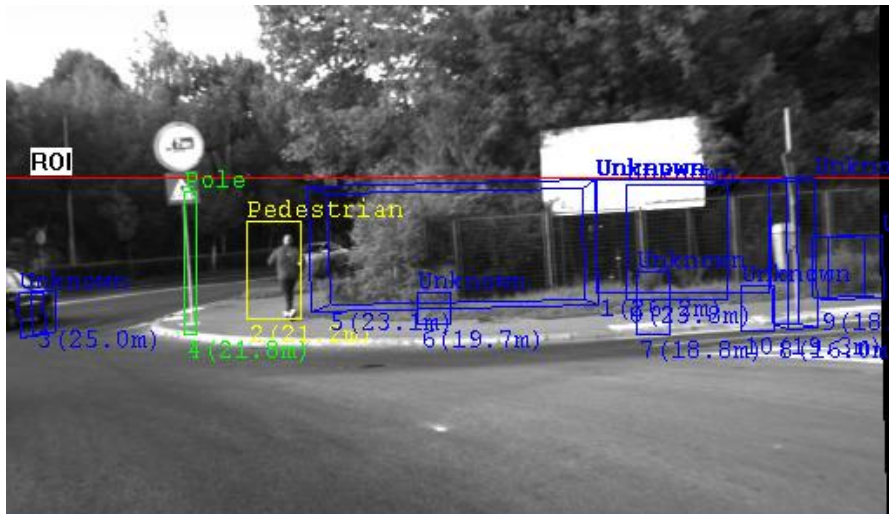
ANSWER: PEDESTRIAN2A000283 is a person.

QUESTION: Image depicts UtilityPole?

ANSWER: UNCLASSIFIED0A000283 is a utility pole. POLE4A000283 is a utility pole. UNCLASSIFIED6A000283 is a utility pole. UNCLASSIFIED7A000283 is a utility pole. UNCLASSIFIED8A000283 is a utility pole. POLE9A000283 is a utility pole.

QUESTION: Image depicts Automobile?

ANSWER: UNCLASSIFIED3A000282 is a car.



Traffic Scene Understanding

QUESTION: Image depicts ObjectWithUse?

ANSWER:

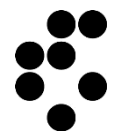
UNCLASSIFIED0A000283 is a utility pole, every utility pole is a post, every post is a shaft, every shaft is a rod, every rod is an implement, every implement is a device, every device is an object with uses.

CAR3A000283 is a car, every car is a device that is not a weapon, every device that is not a weapon is a device, every device is an object with uses.

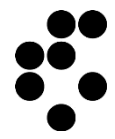
POLE4A000283 is a utility pole, every utility pole is a post, every post is a shaft, every shaft is a rod, every rod is an implement, every implement is a device, every device is an object with uses.



Brehar et al, Spatio-temporal reasoning for traffic scene understanding. 2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing.



STREAM DATA PROCESSING

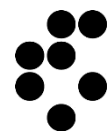


Stream data processing

Applications that require real-time processing of high-volume data streams are pushing the limits of traditional data processing infrastructures

In the following slides we present the requirements of that system...

- ...based on the paper “**The 8 Requirements of Real-Time Stream Processing**” by Stonebraker, Çetintemel, Zdonik; ACM SIGMOD Record Volume 34 Issue 4



Eight rules for stream processing (1/2)

Rule 1: Keep the Data Moving

- *Processing messages “in-stream”, without requirements to store them; ideally the system should also use an active (i.e., non-polling)*

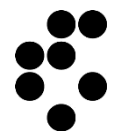
Rule 2: Query using SQL on Streams

- *High-level SQL like language with built-in extensible stream oriented primitives and operators*

Rule 3: Handle Stream Imperfections

- *Dealing with stream “imperfections”, including missing and out-of-order data, which are commonly present in real-world data streams*

Rule 4: Generate Predictable Outcomes



Eight rules for stream processing (2/2)

Rule 5: Integrate Stored and Streaming Data

- Combining stored with *live streaming data*

Rule 6: Guarantee Data Safety and Availability

- *Integrity of the data maintained at all times, despite failures*

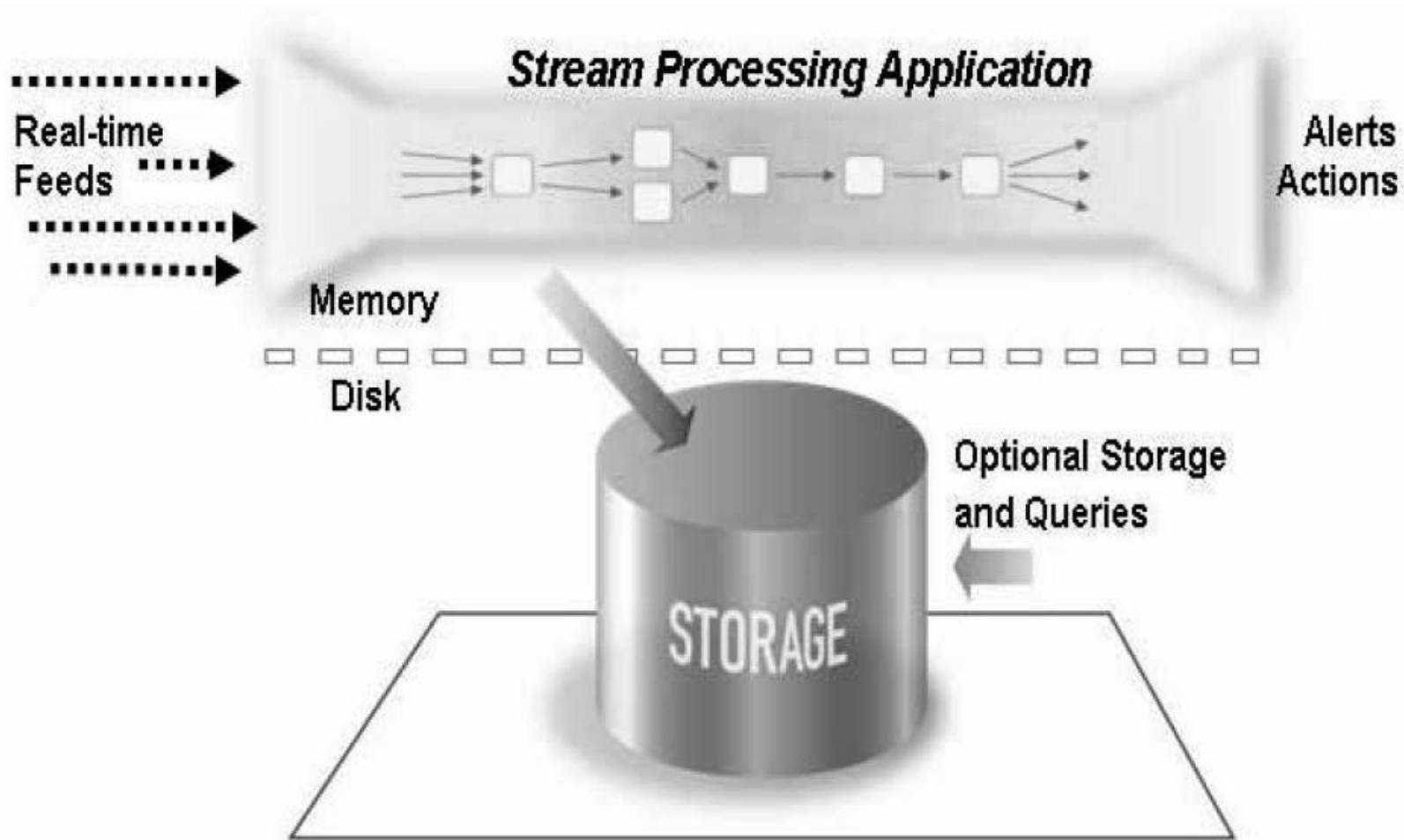
Rule 7: Partition and Scale Applications Automatically

- *Distribute its processing across multiple processors and machines to achieve incremental scalability*

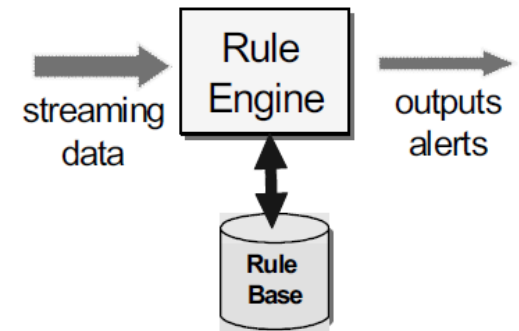
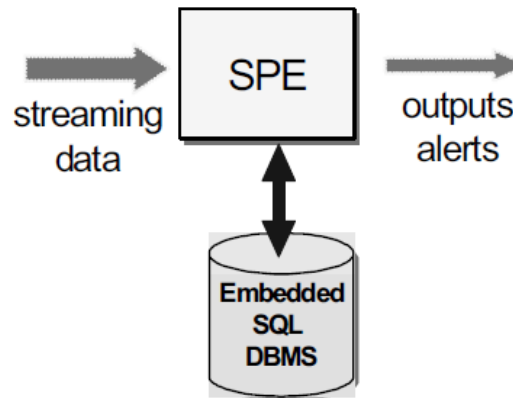
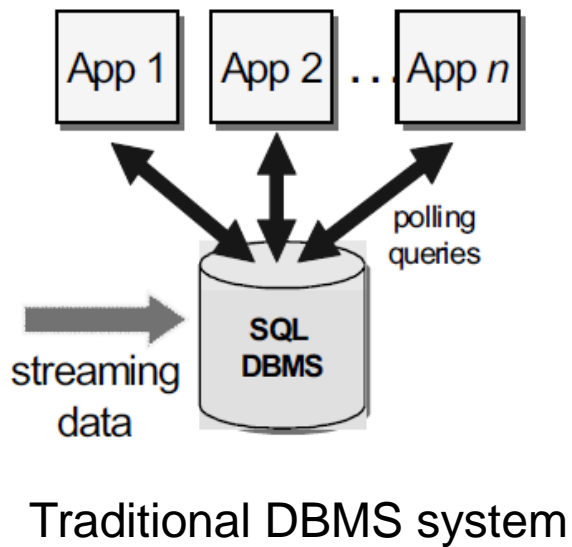
Rule 8: Process and Respond Instantaneously

- *Minimal-overhead execution engine to deliver real-time response*

“Straight-through” processing of messages with optional storage

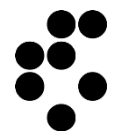


Basic architectures for stream processing databases



The capabilities of various systems software

	DBMS	Rule engine	SPE
Keep the data moving	No	Yes	Yes
SQL on streams	No	No	Yes
Handle stream imperfections	Difficult	Possible	Possible
Predictable outcome	Difficult	Possible	Possible
High availability	Possible	Possible	Possible
Stored and streamed data	No	No	Yes
Distribution and scalability	Possible	Possible	Possible
Instantaneous response	Possible	Possible	Possible

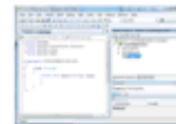


COMPLEX EVENT PROCESSING

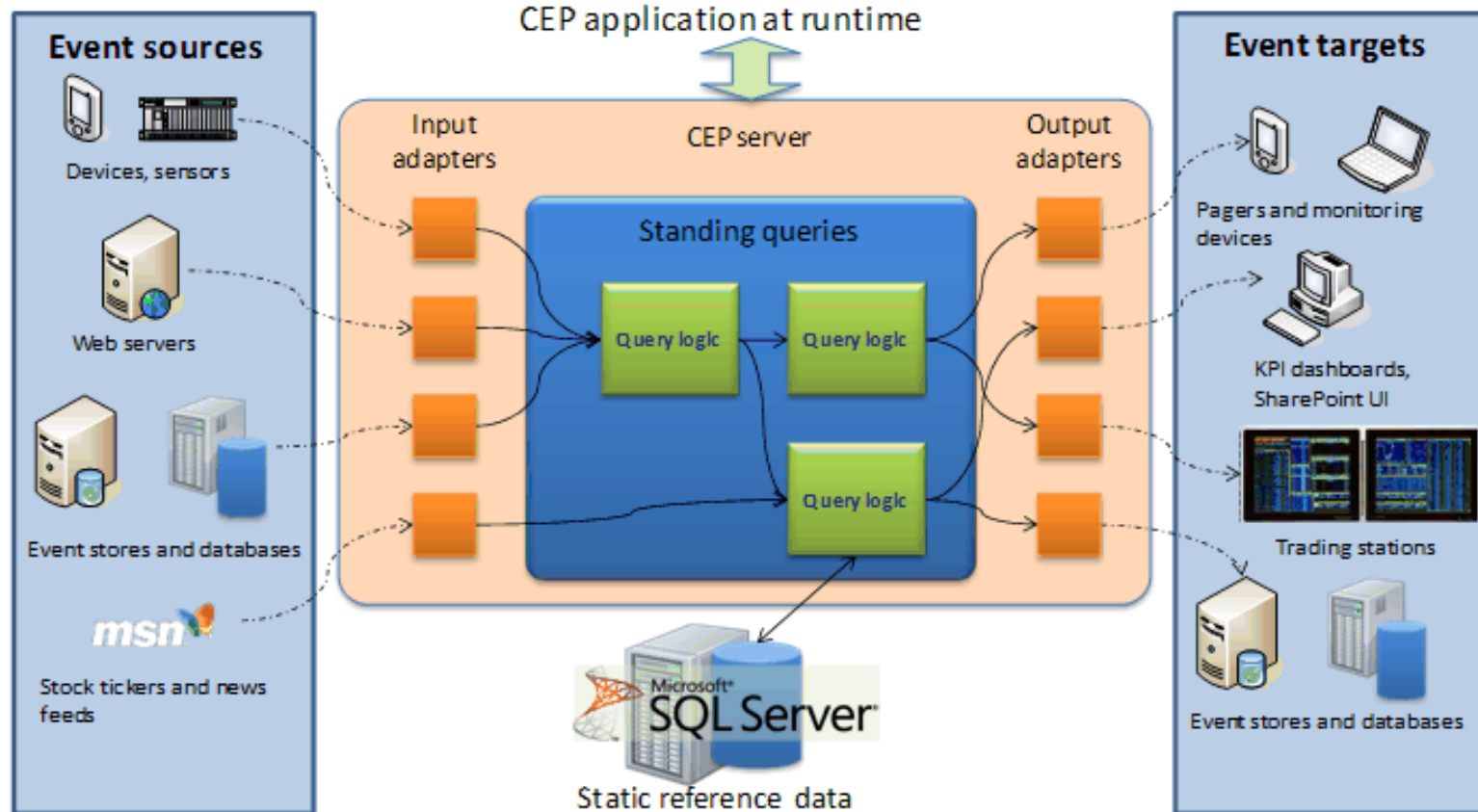
Microsoft *StreamInsight* Architecture

StreamInsight Platform

CEP application development



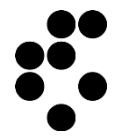
.NET
C#
LINQ





Complex Event Processing Application Development

1. Defining event sources and event targets (sinks)
2. Creating an input adapter to read the events from the source into the CEP server
3. Creating an output adapter to consume the processed events for submission to the event targets
4. Creating the query logic required to meet your business objectives
 1. binding the query to the adapters at runtime, and
 2. to instantiate the query in the CEP server



Examples of Queries in Microsoft StreamInsight (1/2)

Filtering of events

- from e in inputStream where e.value < 10 select e;

Calculations to introduce additional event properties

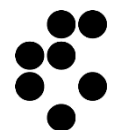
- from e in InputStream select new MeterWattage {wattage=(double)e.Consumption / 10};

Grouping events

- from v in inputStream group v by v.i % 4 into eachGroup from window in eachGroup.Snapshot() select new { avgNumber = window.Avg(e => e.number) };

Aggregation

- from w in inputStream.Snapshot() select new { sum = w.Sum(e => e.i), avg = w.Avg(e => e.f), count = w.Count() };



Examples of Queries in Microsoft StreamInsight (2/2)

Identifying top N candidates

- `(from window in inputStream.Snapshot() from e in window orderby e.f ascending, e.i descending select e).Take(5);`

Matching events from different streams

- `from e1 in stream1 join e2 in stream2 on e1.i equals e2.i select new { e1.i, e1.j, e2.j };`

Combining events from different streams in one

- `stream1.Union(stream2);`

User defined functions

- `from e in stream where e.value < MyFunctions.valThreshold(e.Id) select e;`

Event Models in Microsoft StreamInsight

Interval model

- Event has predefined duration

Event Kind	Start Time	End Time	Payload (Power Consumption)
INSERT	2009-07-15 09:13:33.317	2009-07-15 09:14:09.270	100
INSERT	2009-07-15 09:14:09.270	2009-07-15 09:14:22.253	200
INSERT	2009-07-15 09:14:22.255	2009-07-15 09:15:04.987	100

Point model

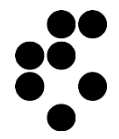
- Event is occurrence in a point in time

Event Kind	Start Time	End Time	Payload (Consumption)
INSERT	2009-07-15 09:13:33.317	2009-07-15 09:13:33.317	100
INSERT	2009-07-15 09:14:09.270	2009-07-15 09:14:09.270	200
INSERT	2009-07-15 09:14:22.255	2009-07-15 09:14:22.255	100

Edge model

- Only start time known upon arrival to server; end-time is updated later

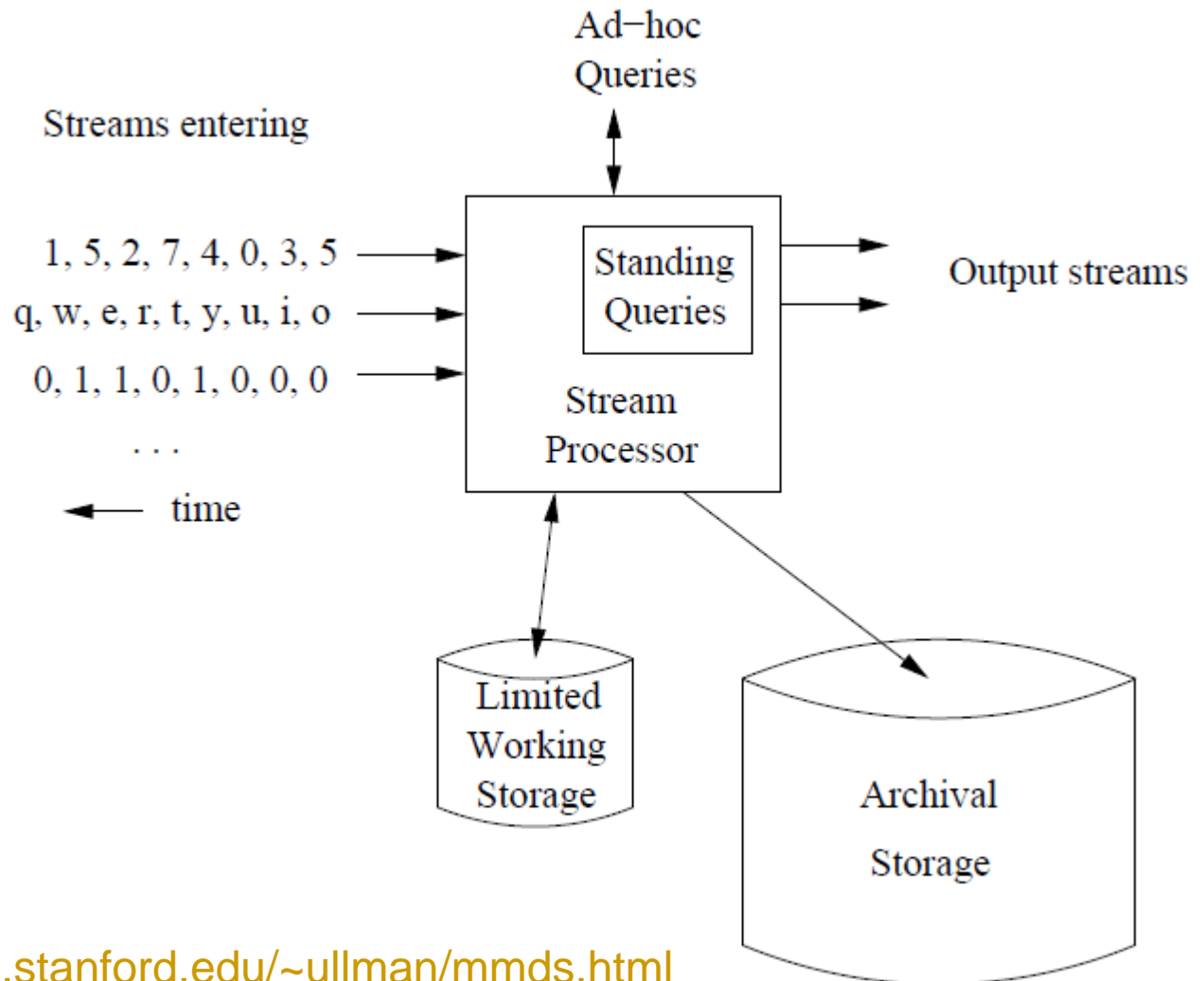
Event Kind	Edge Type	Start Time	End Time	Payload
INSERT	Start	t0	∞	a
INSERT	End	t0	t1	a
INSERT	Start	t1	∞	b
INSERT	End	t1	t3	b
INSERT	Start	t3	∞	c

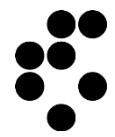


STREAM MINING

Typical stream mining architecture

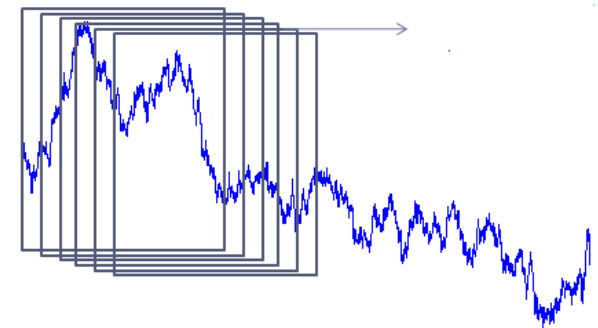
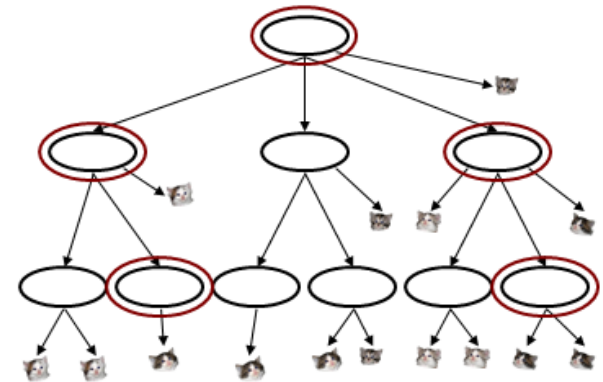
- Streams can include different types of data
- We can prepare system ahead of time for “Standing Queries”
- We can prepare only for certain class of “Ad-hoc Queries”





How we mine streams?

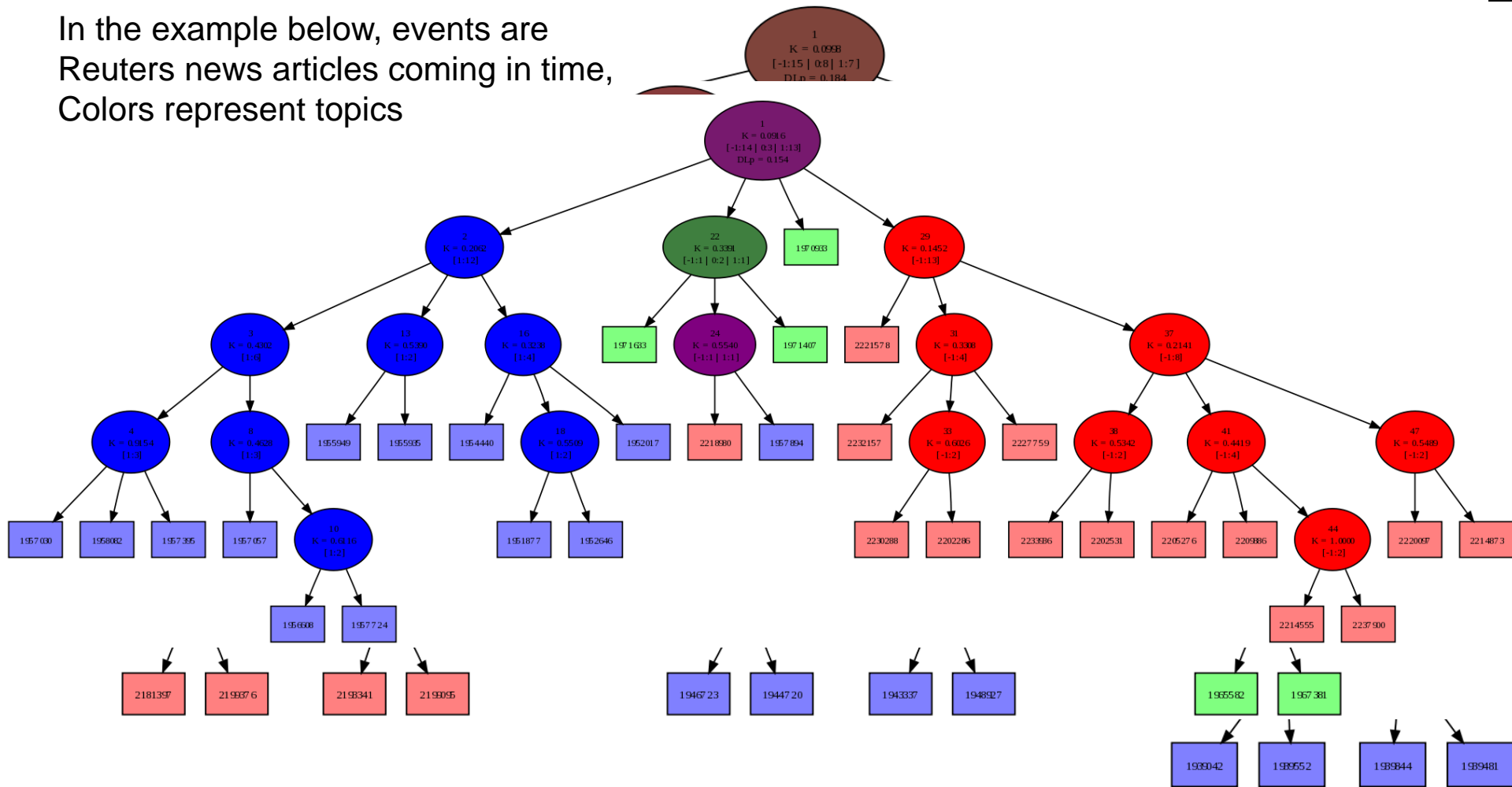
- Maintain **summaries of the streams**, sufficient to answer the expected queries about the data
 - ...summaries can be in various forms: clusters (flat or hierarchic, statistical aggregates, ...)
- Maintain a **sliding window** of the most recently arrived data
 - ...operations on a sliding window mimic more traditional database/mining operations



Example: Stream summarization by incremental hierarchical clustering

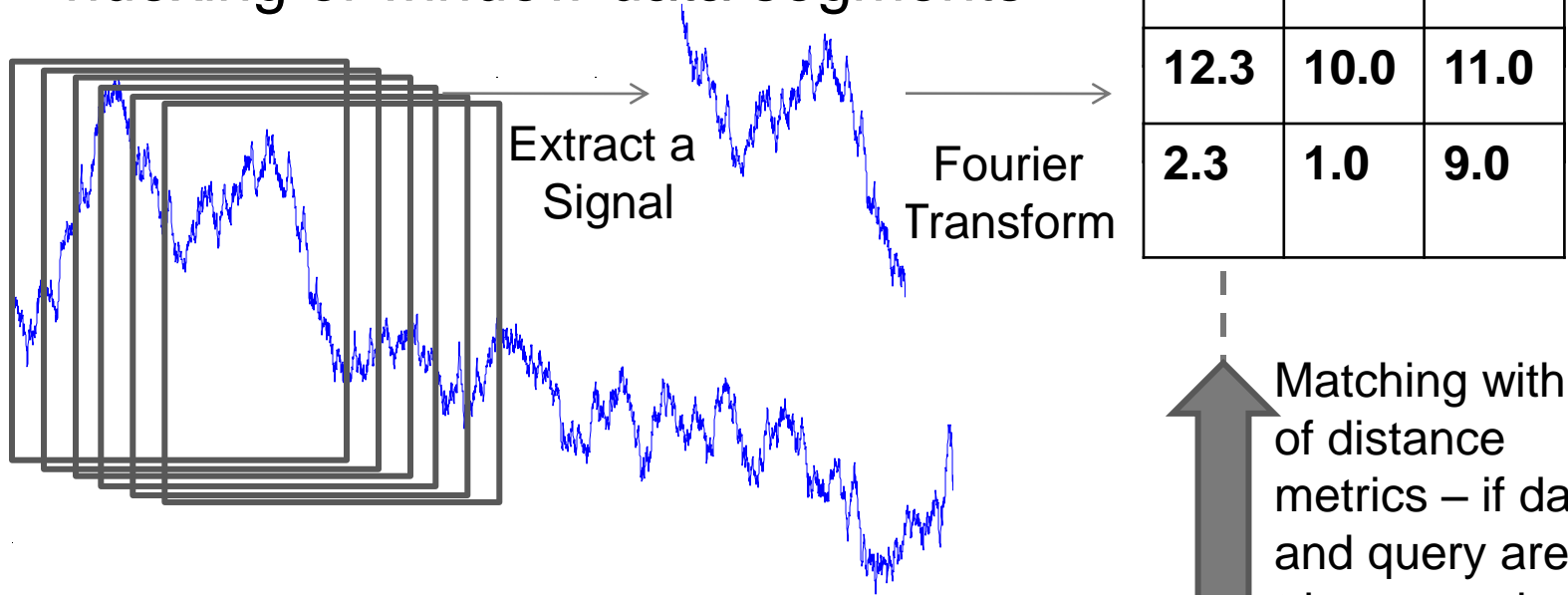
The goal is to maintain summary of data from stream in a form of a taxonomy of prototype clusters – each new events updates the taxonomy

In the example below, events are Reuters news articles coming in time, Colors represent topics

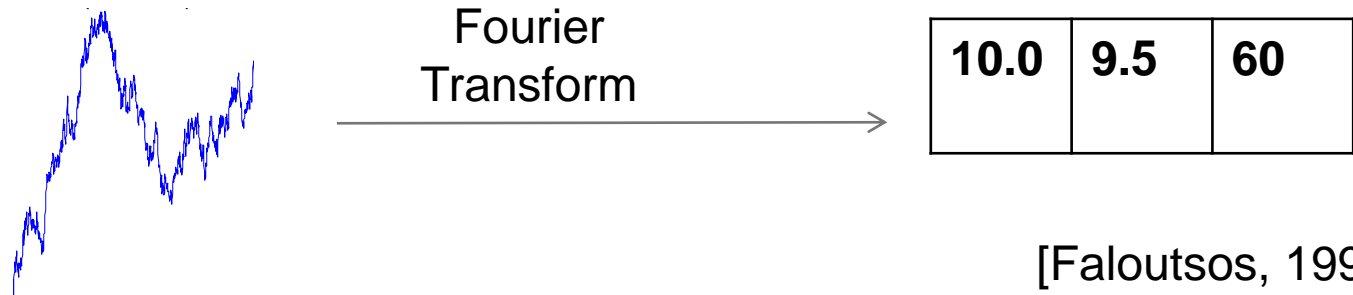


Example: Stream processing on sliding window

- Indexing of window data segments

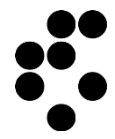


- Query/Template Pattern Preprocessing



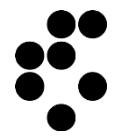
Matching with one of distance metrics – if data and query are similar trigger an alert

[Faloutsos, 1994]



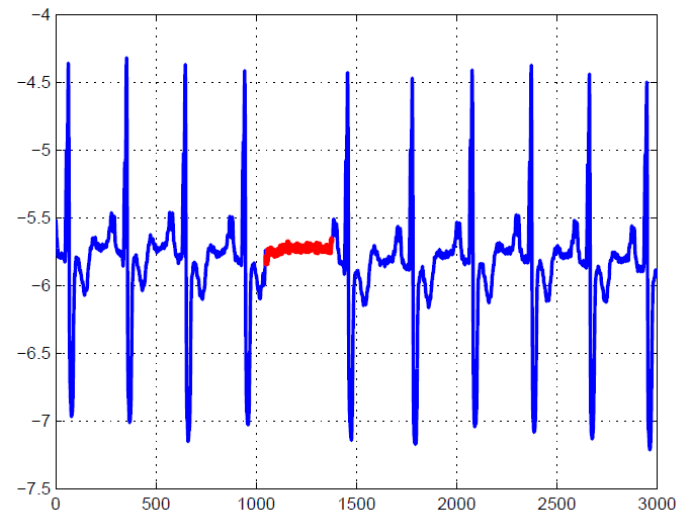
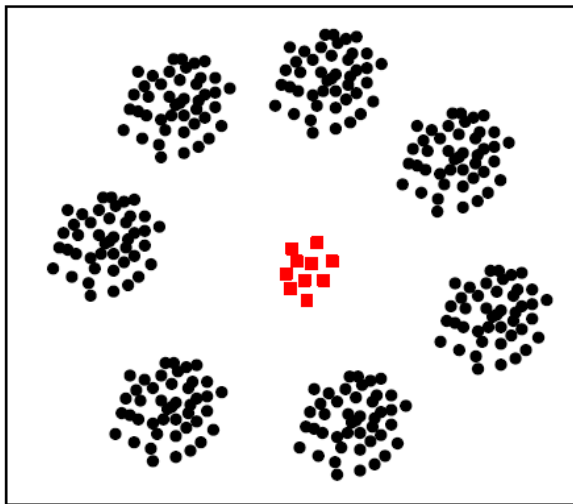
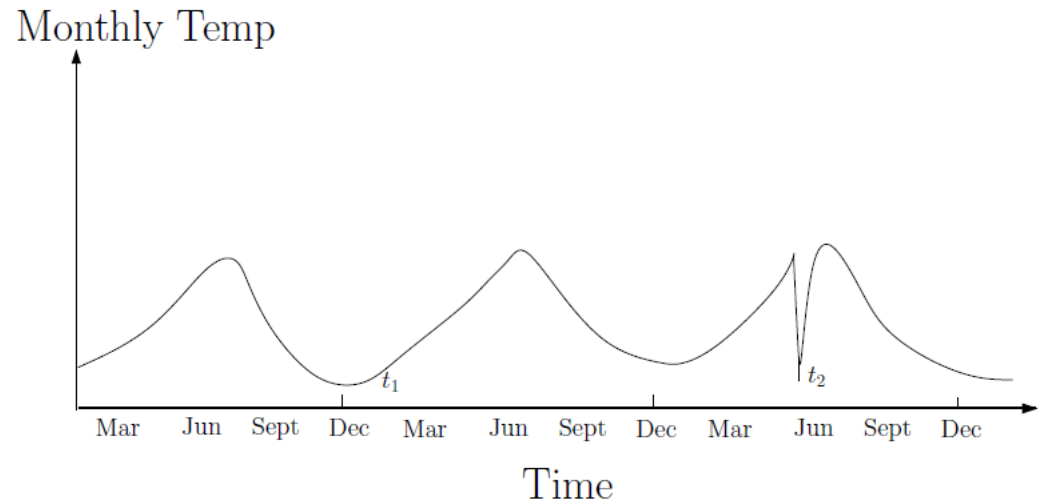
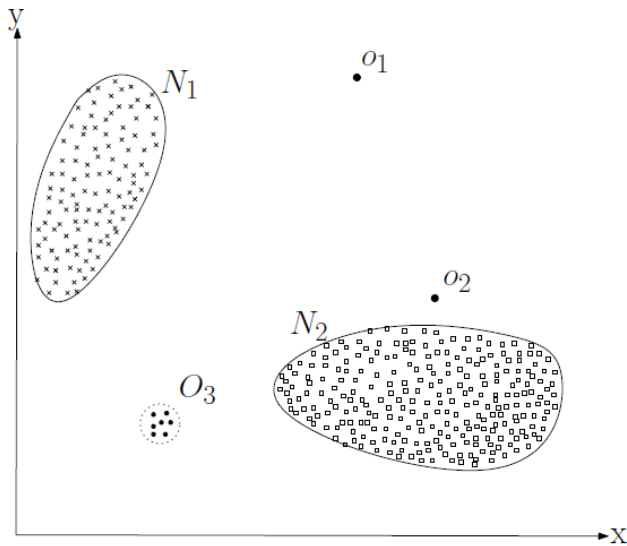
Data reduction stream mining tasks

- **Sampling**
 - ...challenge is to obtain representative data sample (i.e., enabling to perform correctly required operations on data)
- **Filtering**
 - ...simple filters are easy to implement (e.g. simple conditions like “ $x < 10$ ”)
 - ...filtering by a membership of a set which doesn't fit in the main memory requires more sophisticated algorithms (e.g. Bloom filtering)
 - (example of set membership: list of spam URLs)



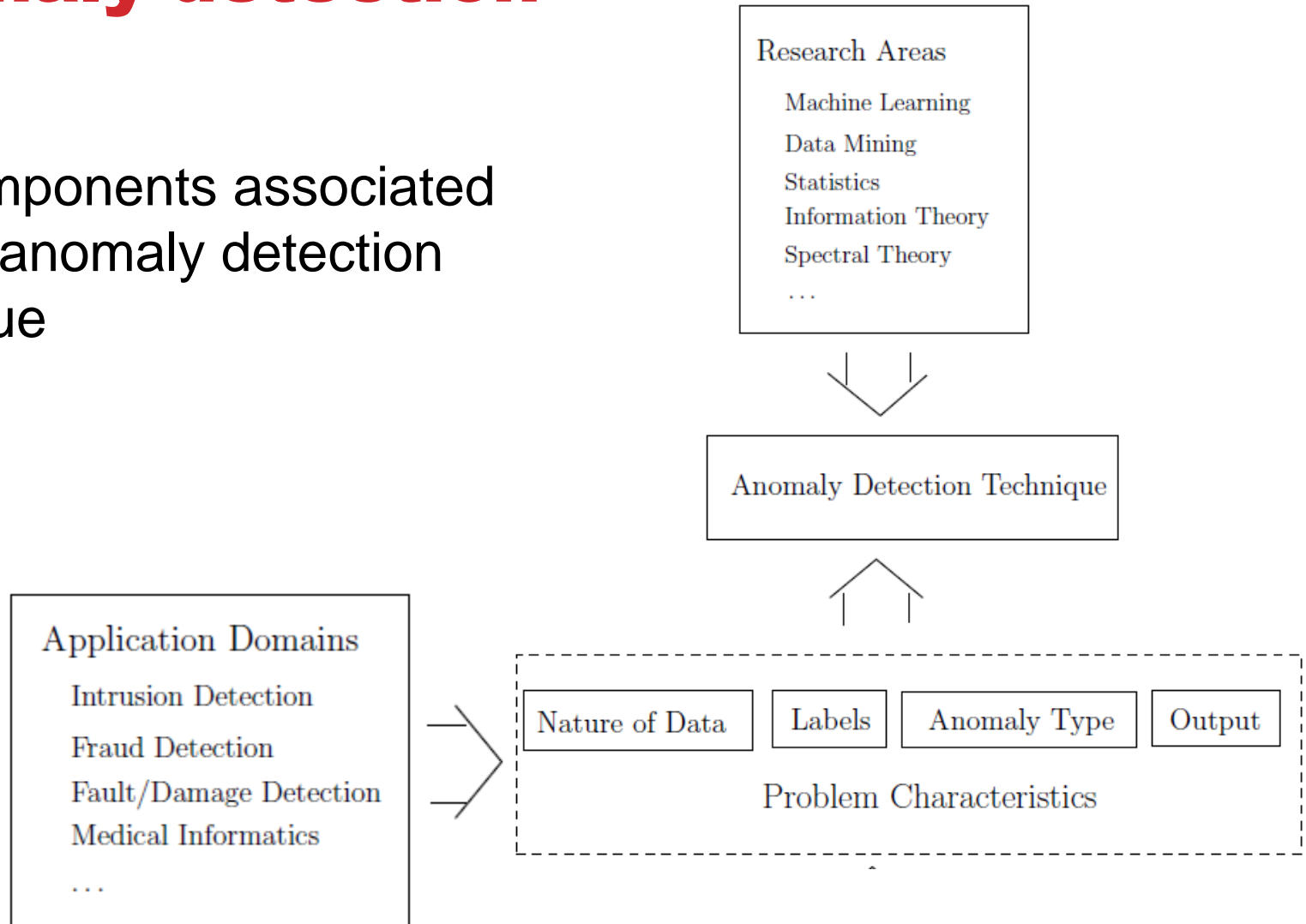
ANOMALY DETECTION

What are anomalies?



Anomaly detection

Key components associated with an anomaly detection technique



Techniques to detect anomalies

Classification based

- A pre-trained classifier can distinguish between normal and anomalous classes

Clustering based

- Normal data instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters

Nearest neighbor approaches

- *Normal data instances occur in dense neighborhoods, while anomalies occur far from their closest neighbors*

Statistical approaches

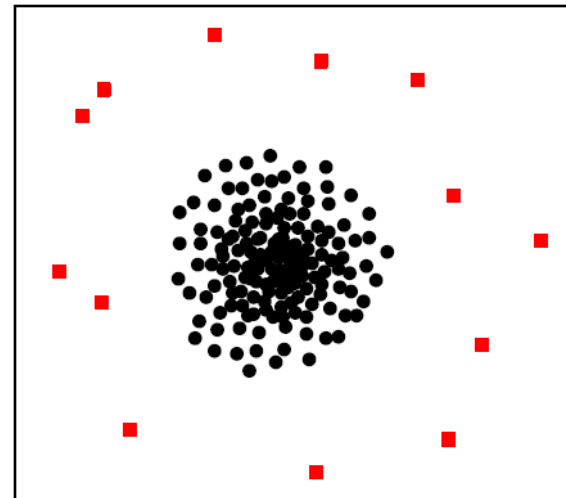
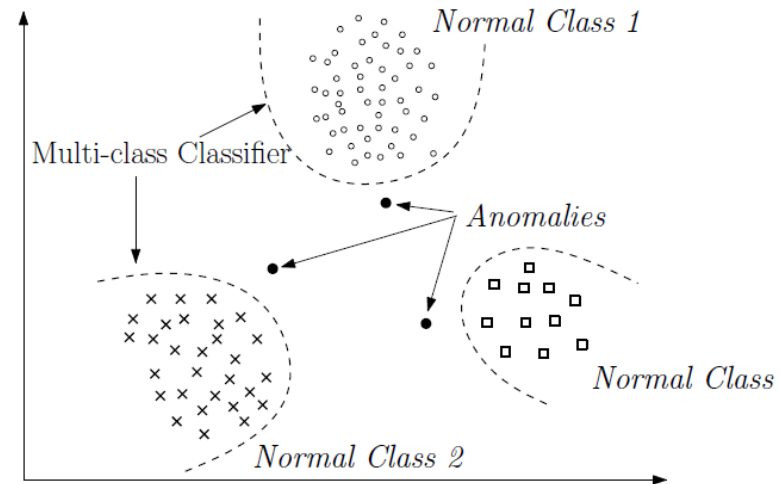
- *Normal data instances occur in high probability regions of a stochastic model, while anomalies occur in the low probability regions*

Information theoretic approaches

- *Anomalies in data induce irregularities in the information content of the data set*

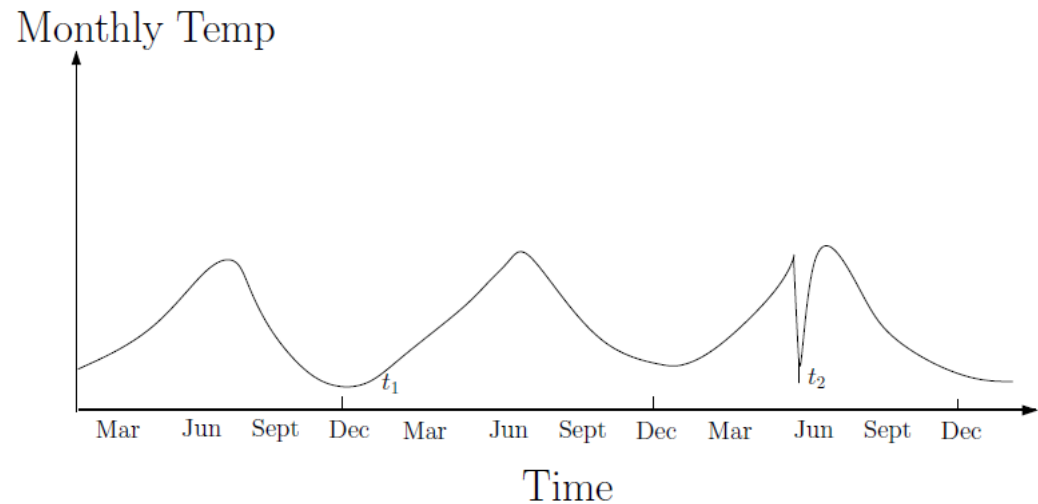
Spectral methods

- *Normal instances appear in a lower dimensional subspace, anomalies in the rest (noise)*



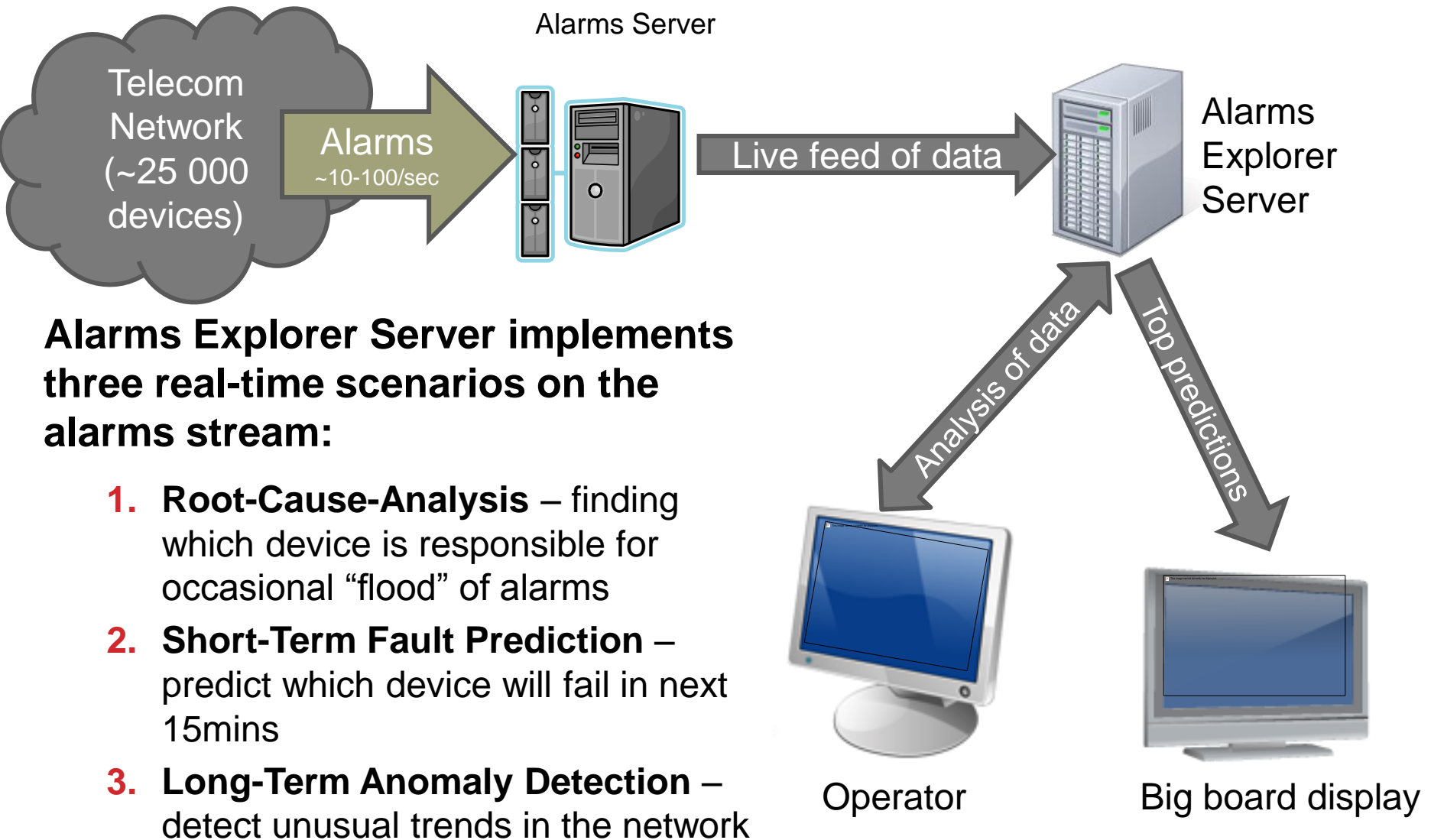
The key to a successful anomaly detection is proper feature engineering!

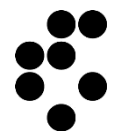
Anomalies are detectable if data instances are represented in an informative feature space



Contextual anomaly t_2 in a temperature time series. Note that the temperature at time t_1 is same as that at time t_2 but occurs in a different context and hence is not considered as an anomaly.

Application: Telecommunication Network Monitoring



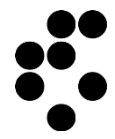


Outline

Part I. Motivation & background

Part II. Technology and tools for exploiting the WoT

Part III. Demos, Tools & Research directions



Outline

Part III. Demos, Tools & Research directions

Use cases

- What systems and prototypes exist?

Open problems

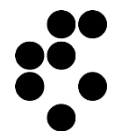
- Are there unsolved problems?

Summary

- What was this tutorial about?

List of sources for further studies

- Where to start digging?

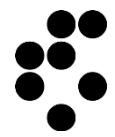


Use cases

Environmental intelligence

Others

- Intelligent buildings
- Smart cities
- Smart infrastructures
- ...



Environmental intelligence

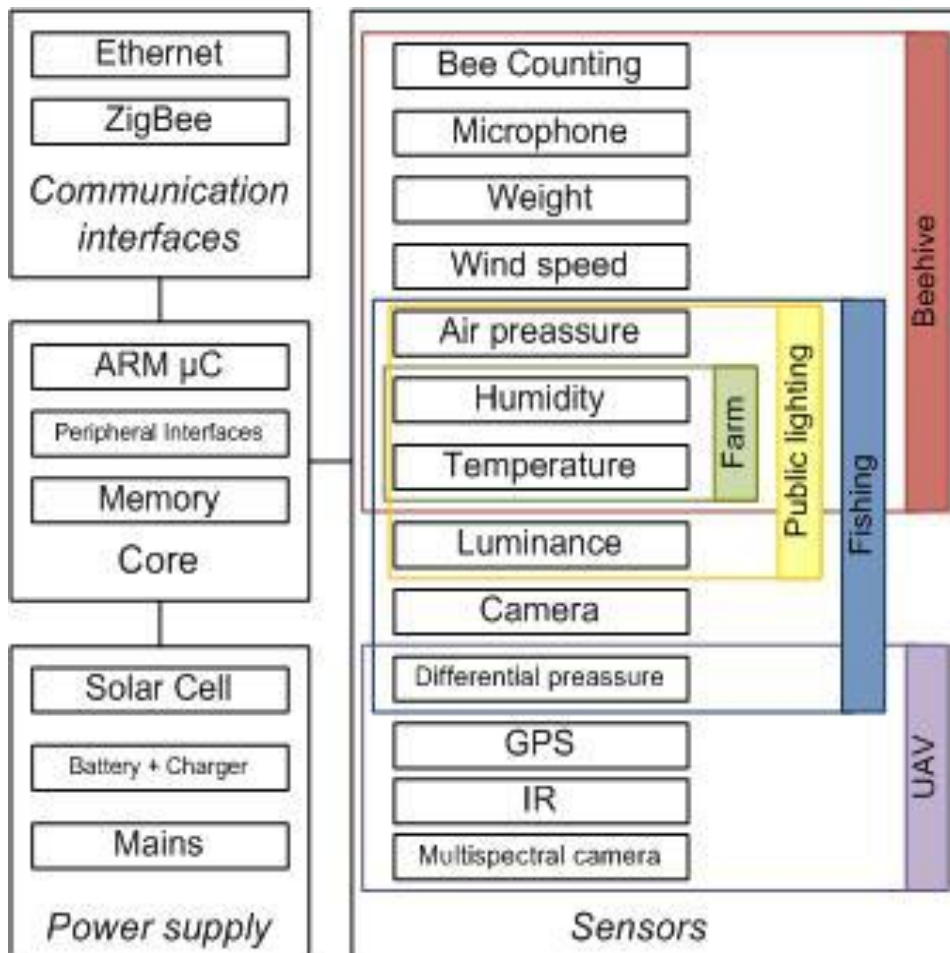
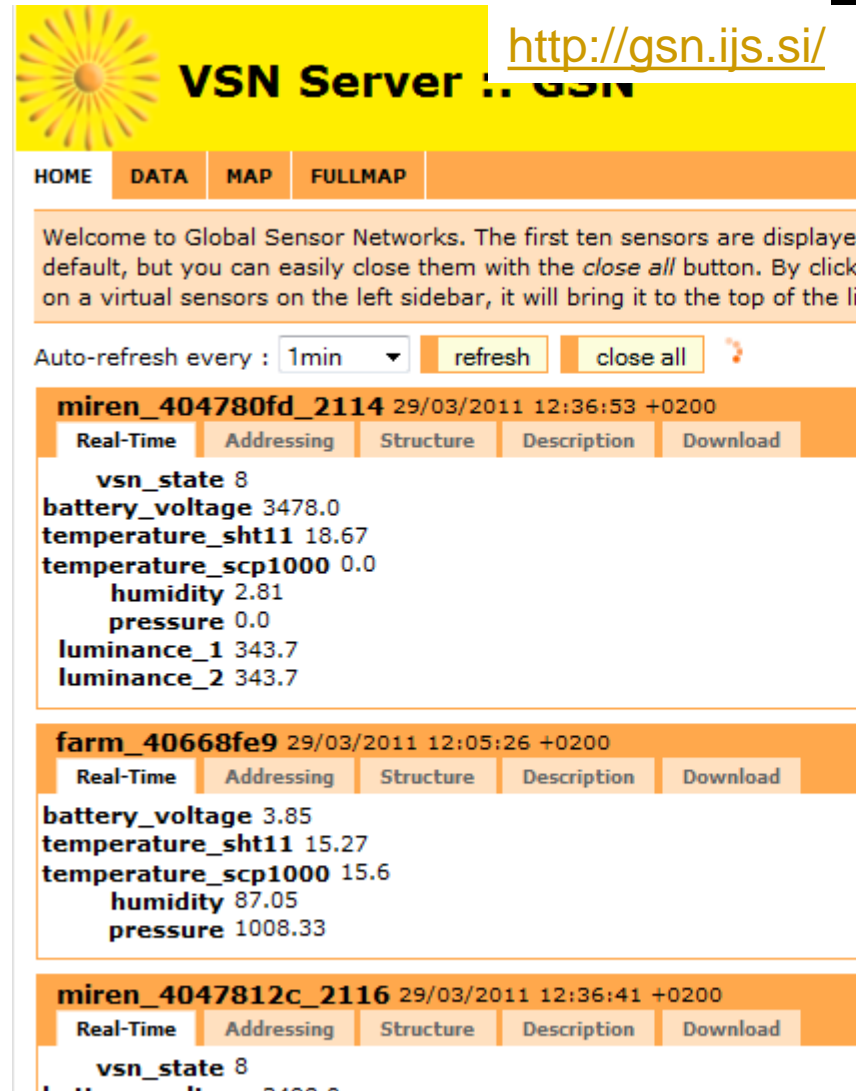
Data from large number of sensors deployed in infrastructure (such as roads) or over other area of interest (such as agriculture fields) can give decision makers a real-time awareness on the observed phenomena and events.

Solutions:

- Remote monitoring of cultures, soil moisture, insect infestations or disease infections
- Irrigation and pesticide spraying in precision agriculture
- Livestock monitoring for maximizing production (meat, milk, eggs) and achieve higher reproduction rates

Our quick start

VESNA Sensor Node

<http://gsn.ijs.si/>

VSN Server

HOME DATA MAP FULLMAP

Welcome to Global Sensor Networks. The first ten sensors are displayed by default, but you can easily close them with the *close all* button. By clicking on a virtual sensors on the left sidebar, it will bring it to the top of the list.

Auto-refresh every : refresh close all

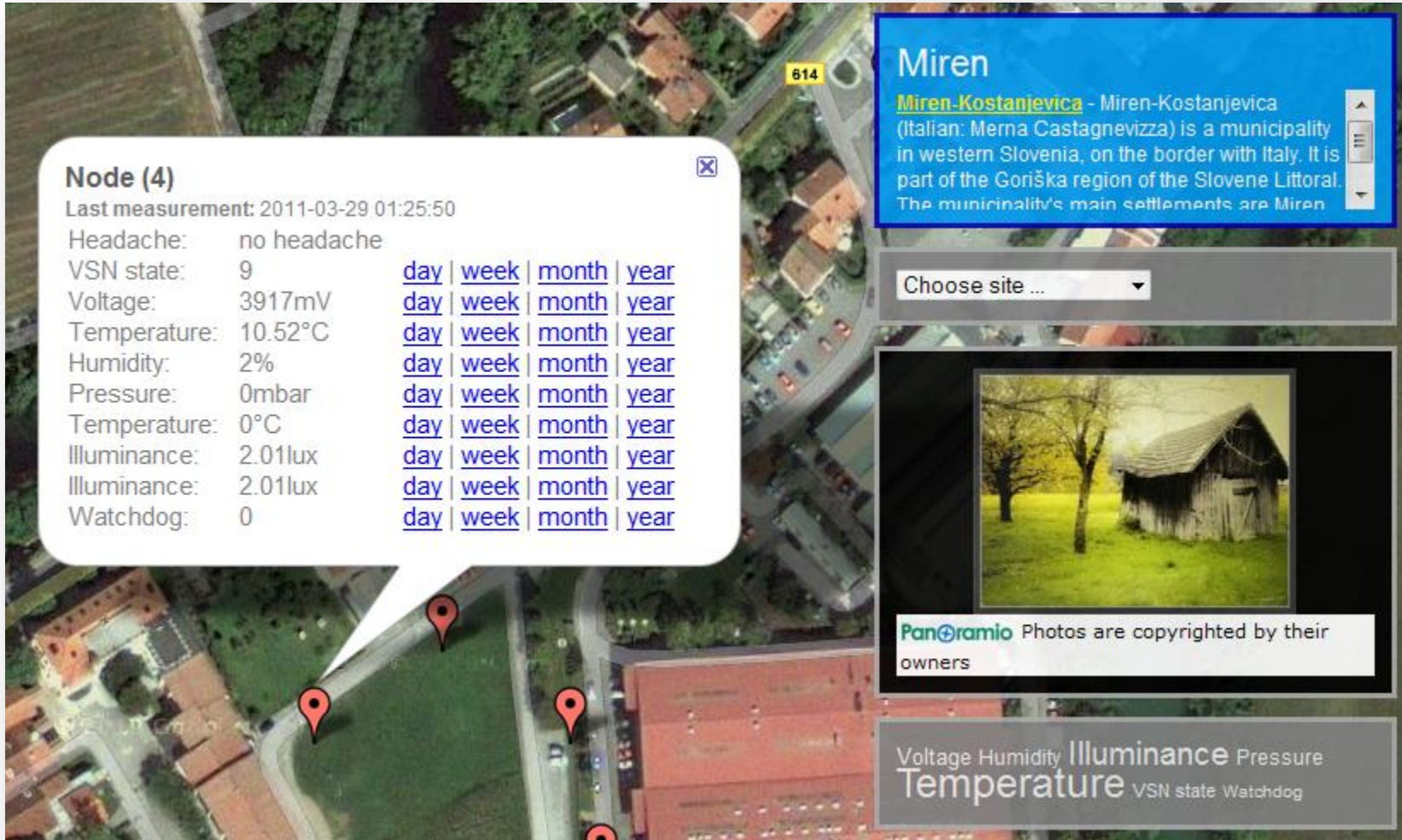
Real-Time	Addressing	Structure	Description	Download
miren_404780fd_2114 29/03/2011 12:36:53 +0200				
vsn_state 8				
battery_voltage 3478.0				
temperature_sht11 18.67				
temperature_scp1000 0.0				
humidity 2.81				
pressure 0.0				
luminance_1 343.7				
luminance_2 343.7				
farm_40668fe9 29/03/2011 12:05:26 +0200				
battery_voltage 3.85				
temperature_sht11 15.27				
temperature_scp1000 15.6				
humidity 87.05				
pressure 1008.33				
miren_4047812c_2116 29/03/2011 12:36:41 +0200				
vsn_state 8				
battery_voltage 3499.0				

Environmental monitoring and lights control testbed

- Location: Slovenia, Europe (project started August 2010)
- The “things”: public light poles + VESNA sensor nodes
- Sensors: temperature, humidity, pressure, illuminance, etc.
- Actuator: dim the intensity of the light (pulse width modulation)



Videk: mash-up for environmental intelligence



Node (4) ✕


Last measurement: 2011-03-29 01:25:50

Headache:	no headache	
VSN state:	9	day week month year
Voltage:	3917mV	day week month year
Temperature:	10.52°C	day week month year
Humidity:	2%	day week month year
Pressure:	0mbar	day week month year
Temperature:	0°C	day week month year
Illuminance:	2.01lux	day week month year
Illuminance:	2.01lux	day week month year
Watchdog:	0	day week month year

Miren

Miren-Kostanjevica - Miren-Kostanjevica (Italian: Merna Castagnevizza) is a municipality in western Slovenia, on the border with Italy. It is part of the Goriška region of the Slovene Littoral. The municipality's main settlements are Miren

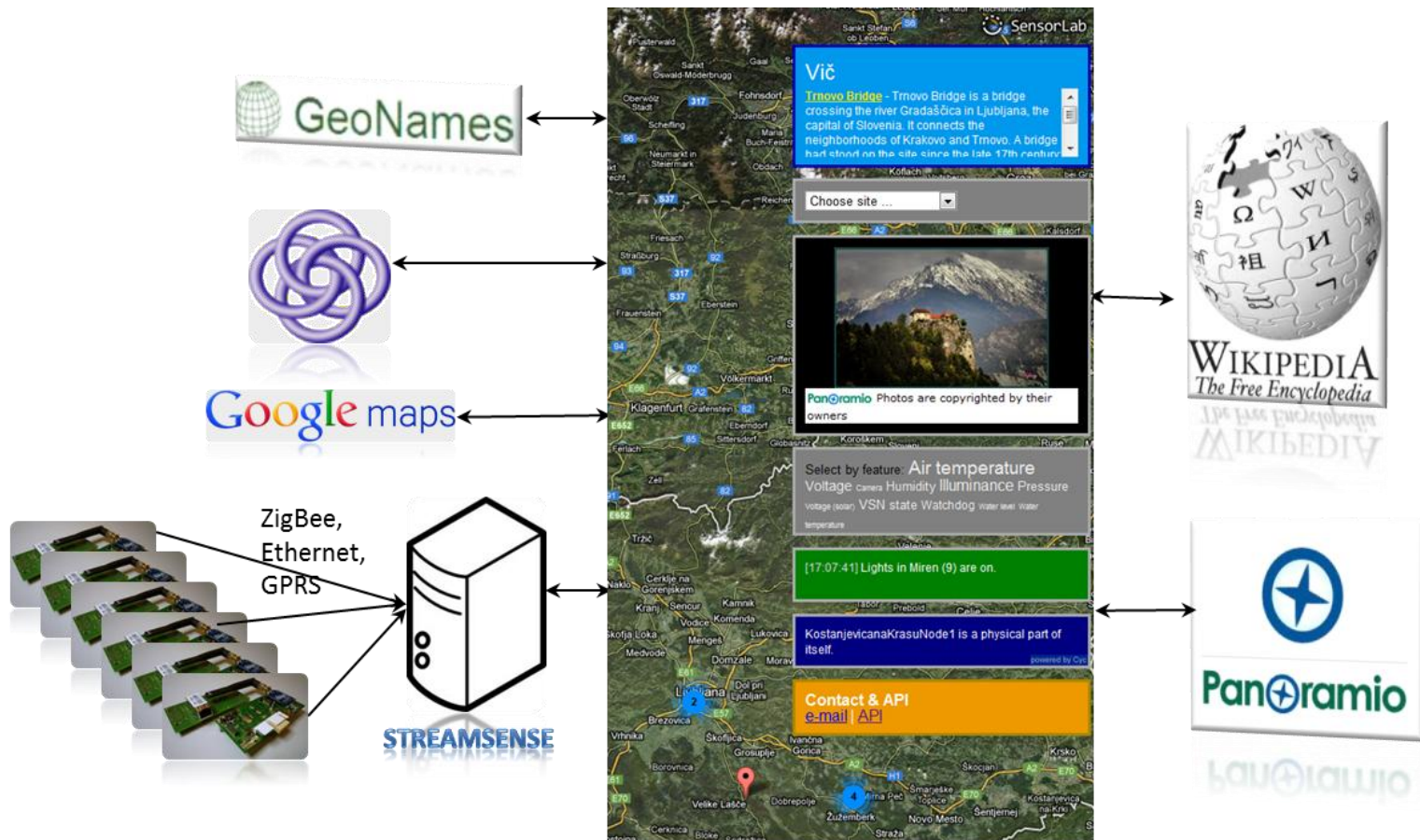
Choose site ...



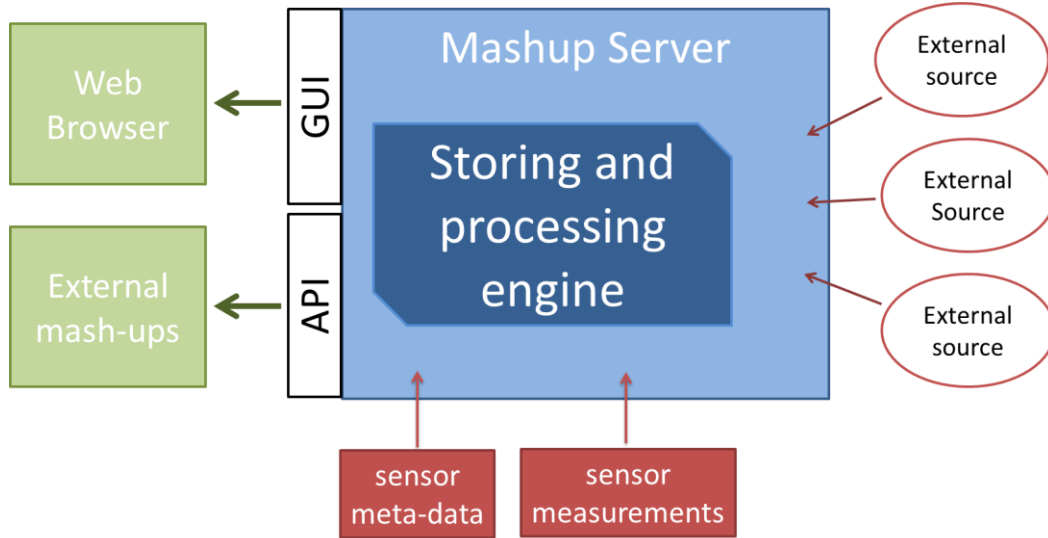
Panoramio Photos are copyrighted by their owners

Voltage Humidity Illuminance Pressure
Temperature VSN state Watchdog

Videk: mashed sources of data



Technology behind Videk



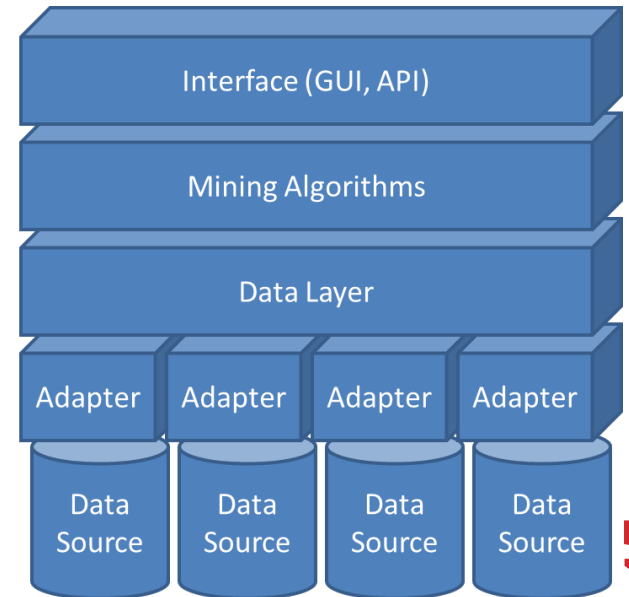
User interface and web service API to the lower levels

Wrappers around mining algorithms from GLib

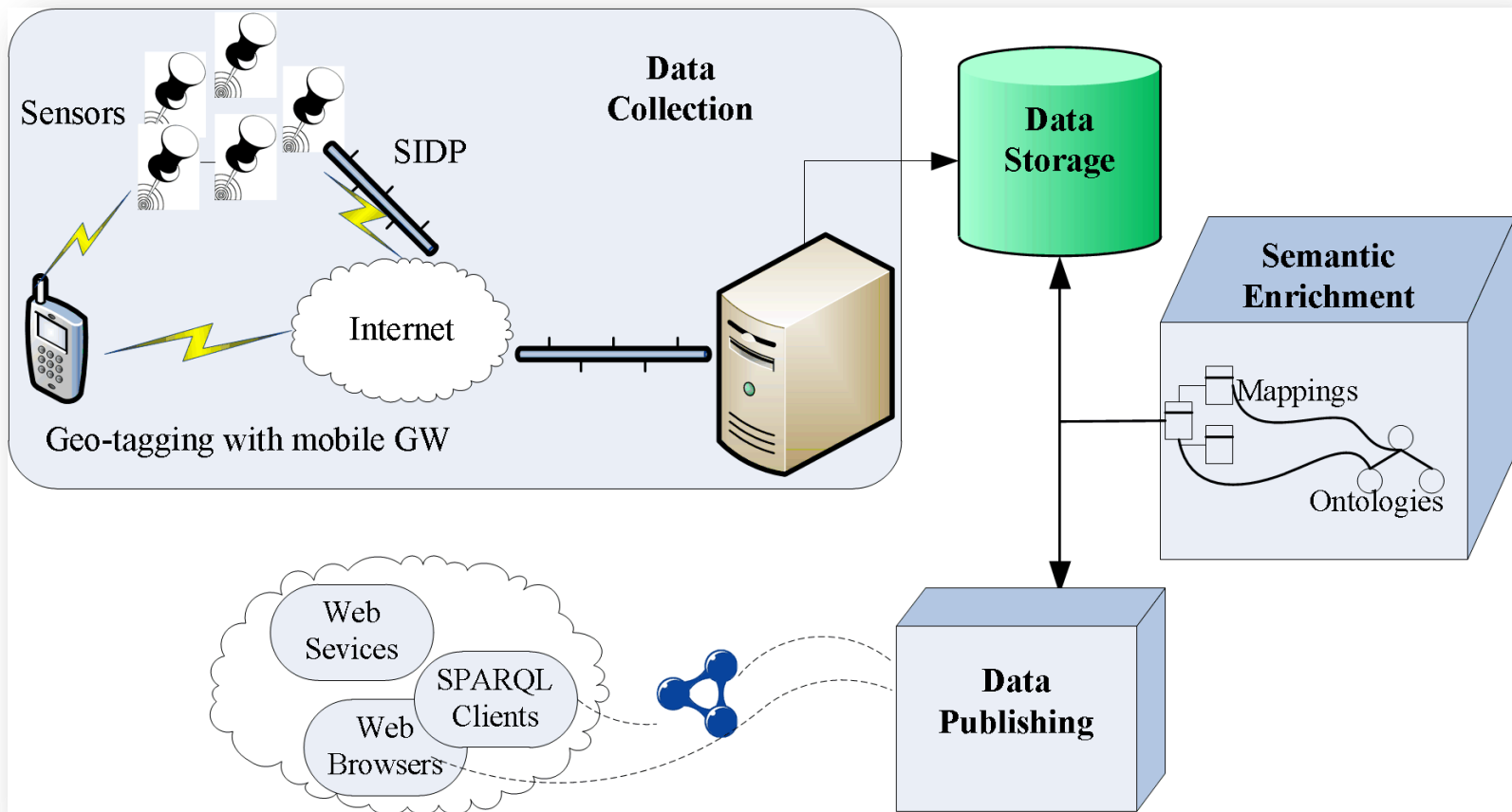
Efficient access and sampling from the data sources

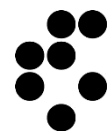
Data Source Adapters

Multiple data inputs of structured and non-structured data



Environmental intelligence: SemSense system architecture





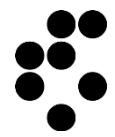
Environmental intelligence: SemSense implementation details

Scenario

- architecture for collecting real world data from a physical system of sensors and publishing it on the Web

Implementation:

- VESNA Sensor Nodes platform are the “things”
- Self-Identification Protocol
 - Custom protocol for collecting meta-data and data
- MySQL database for storage of data and meta-data
- Meta-data semantic enrichment component
 - RDF representation
 - Semantic Sensor Network (SSN) ontology, Basic GeoWGS84 Vocabulary, GeoNames and FOAF as vocabulary
 - Linking to Linked Opened Data Cloud
 - D2R for mapping the database schema



Environmental intelligence: browse the semantic representation

JSI SensorLab
Running at <http://sensorlab.ijs.si:2020/>

Home | [Archive-1Day-Sampling](#) | [DeviceType](#) | [Observation](#) | [ObservationValue](#) | [Platform](#) | [Property](#) | [SensingDevice](#) | [SensorOutput](#) | [System](#)

This is a database published with D2R Server. It can be accessed using

1. your plain old web browser
2. Semantic Web browsers
3. SPARQL clients.

1. HTML View

You can use the navigation links at the top of this page to explore the database.

2. RDF View

You can also explore this database with **Semantic Web browsers** like [Tabulator](#) or [Disco](#). To start browsing, open this entry point URL in your Semantic Web browser:

<http://sensorlab.ijs.si:2020/all>

3. SPARQL Endpoint Browse at: <http://sensors.ijs.si:2020/>

Sensor Search Example

Search results

A Galveston Pleasure Pier
 Station information for Galveston Pleasure Pier (8771510). Observed data: WaterLevel, WaterLevelPredictions, Winds, AirTemperature, WaterTemperature, BarometricPressure.

- sensor-WaterLevel - WaterLevel instrument for station 8771510
- sensor-WaterLevelPredictions - WaterLevelPredictions instrument for station 8771510
- sensor-Winds - Winds instrument for station 8771510
- sensor-Winds - Winds instrument for station 8771510
- sensor-Winds - Winds instrument for station 8771510
- sensor-AirTemperature - AirTemperature instrument for station 8771510
- sensor-WaterTemperature - WaterTemperature instrument for station 8771510
- sensor-BarometricPressure - BarometricPressure instrument for station 8771510

urn:x-noaa:def:station:NOAA.NOS.CO-OPS::8771510

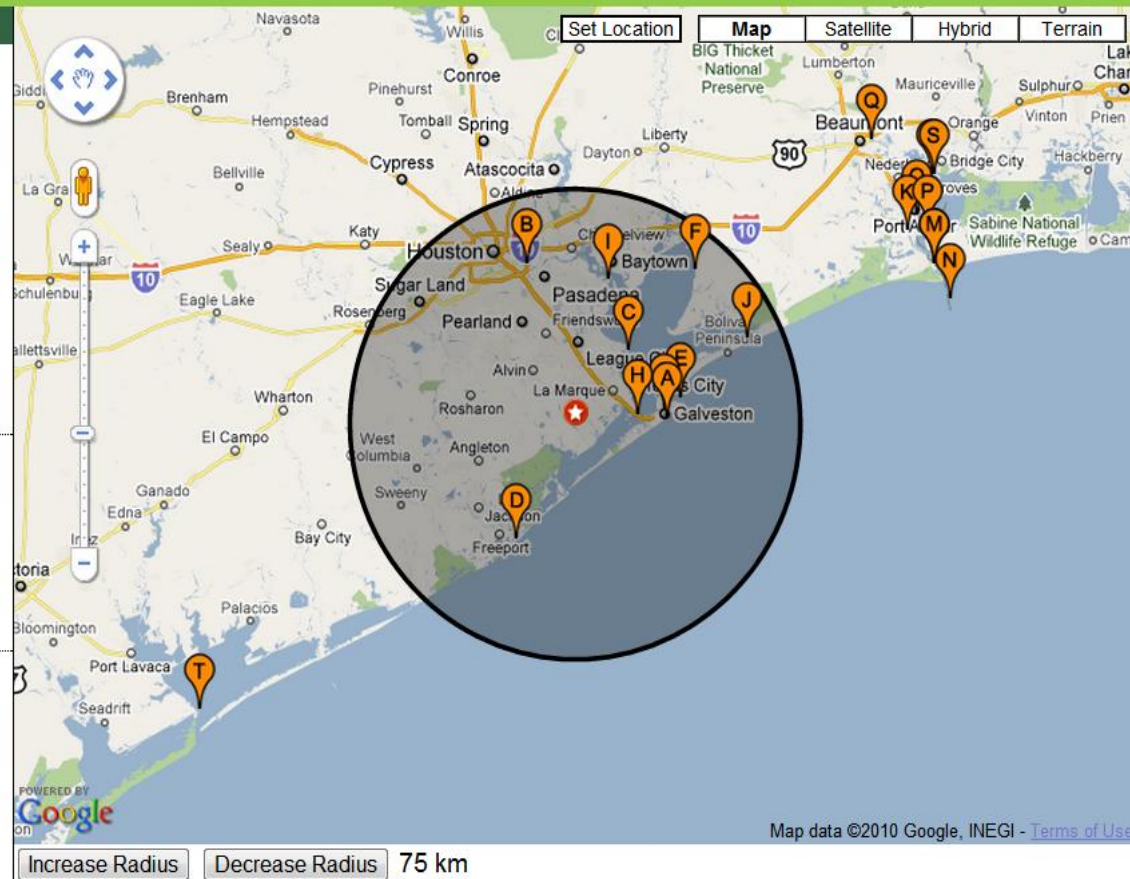
B Manchester
 Station information for Manchester (8770777). Observed data: WaterLevel, WaterLevelPredictions, WaterTemperature.

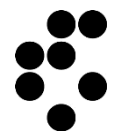
- sensor-WaterLevel - WaterLevel instrument for station 8770777
- sensor-WaterLevelPredictions - WaterLevelPredictions instrument for station 8770777
- sensor-WaterTemperature - WaterTemperature instrument for station 8770777

urn:x-noaa:def:station:NOAA.NOS.CO-OPS::8770777

C Eagle Point
 Station information for Eagle Point (8771013). Observed data: WaterLevel, WaterLevelPredictions, Winds, AirTemperature, WaterTemperature, BarometricPressure, Conductivity, Salinity.

- sensor-WaterLevel - WaterLevel instrument for station 8771013
- sensor-WaterLevelPredictions - WaterLevelPredictions instrument for station 8771013
- sensor-Winds - Winds instrument for station 8771013
- sensor-Winds - Winds instrument for station 8771013





Sensor search and ranking

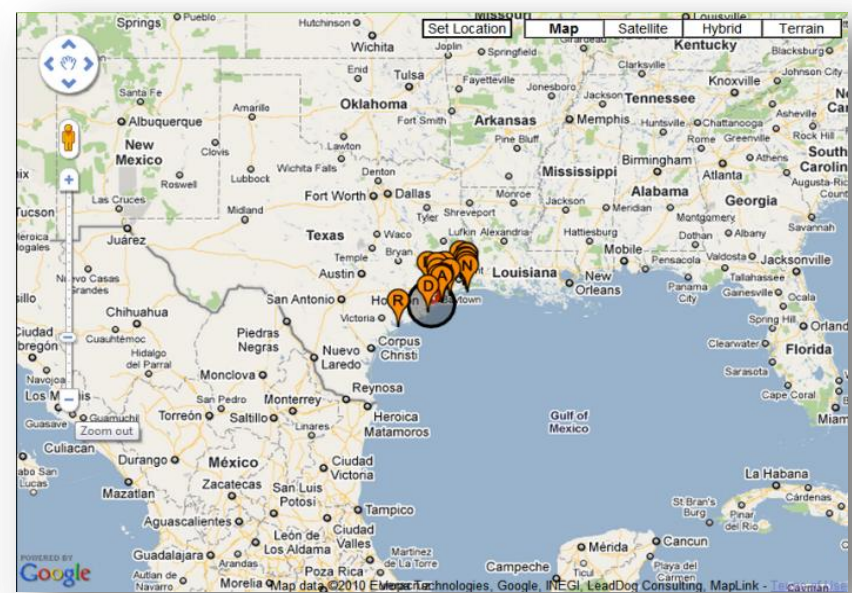
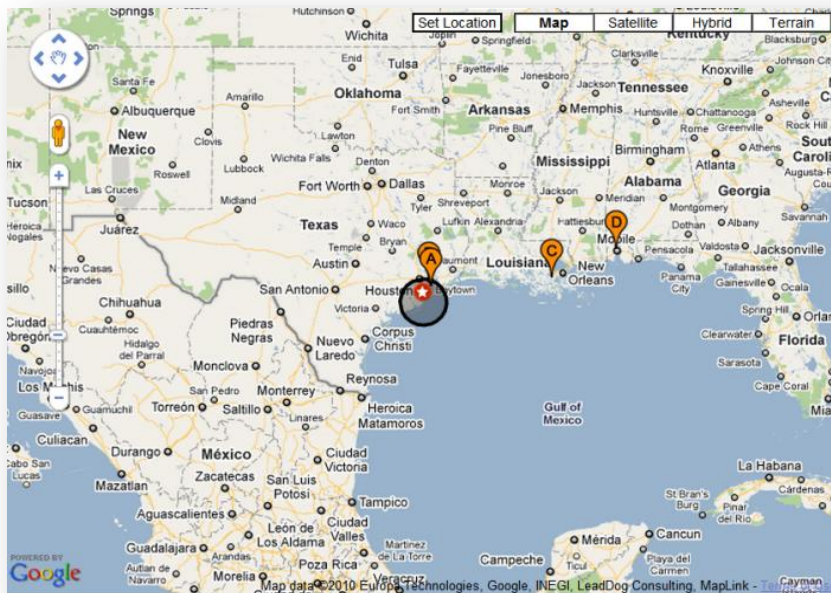
The goal of the search

- retrieve and rank a list of sensors based on the user's request
- Input:
 - keyword query
 - geographic location (given by latitude and longitude coordinates)
 - distance (interpreted as a radius around the location)
- Output:
 - list of ranked sensors

Sensor Search Example

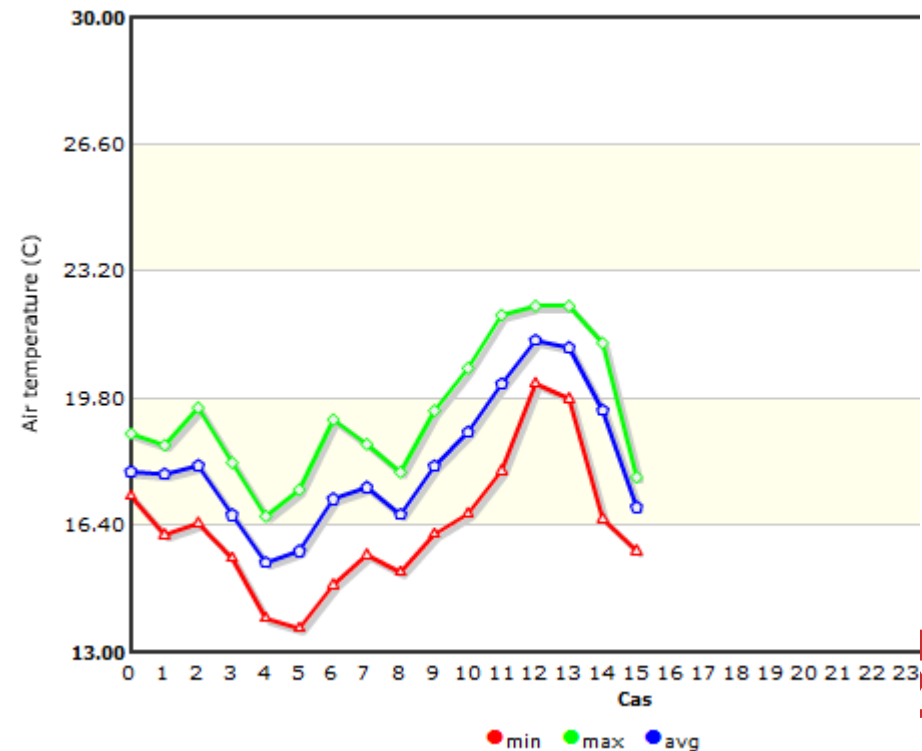
Performing the proposed ranking results in obtaining more platforms closer to the area of interest

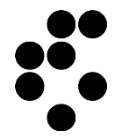
- we consider relevant also sensors located on the same platform or those that are in the same deployment



Hyperthermia detection in stables

- The goal is to measure temperature and humidity inside and outside stables in order to detect the danger of hyperthermia at cows and issue an early warning.





Hyperthermia detection in stables



Node (18)
Last measurement: 2011-05-16 10:08:21

Headache:	no headache	
Voltage:	3.79mV	day week month year
Air temperature:	16.62°C	day week month year
Humidity:	76.32%	day week month year
Pressure:	1012.45mbar	day week month year
Air temperature:	16.87°C	day week month year

Reber
Žužemberk - Žužemberk, is a town and a municipality in the Dinaric Alps of Slovenia, located south-east of the Slovenian capital of Ljubljana. As of 2002 the municipality had a total population of 4579. Žužemberk lies in the

Choose site ...

Panoramio Photos are copyrighted by their owners

Select by feature: **Air temperature**
Voltage camera Humidity Illuminance Pressure
Voltage (solar) VSN state Watchdog Water level Water
temperature

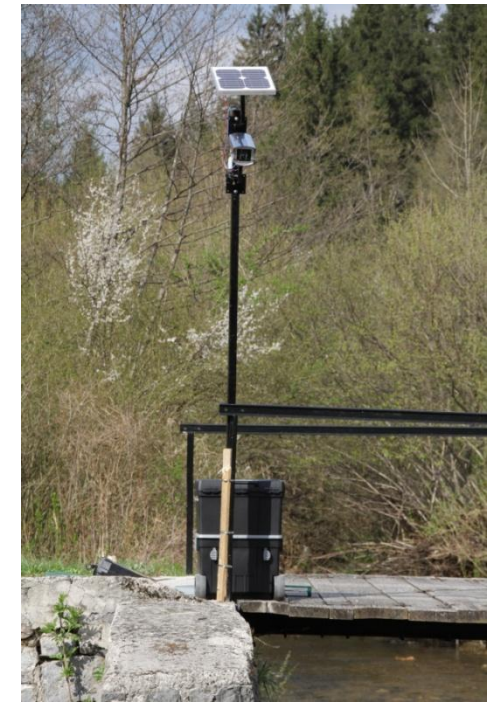
<http://sensorlab.ijs.si/sl/demos.html>

Remote observation of sport-fishing conditions

By using WSN technology information on water level, picture of a fishing spot and water and outside temperature can be provided



<http://sensorlab.ijs.si/sl/demos.html>



Remote observation of sport-fishing conditions

Node (20)
 Last measurement: 2011-05-16 10:39:27

Headache:	no headache	
Voltage:	4138mV	day week month year
Camera:	1	
Water level:	0.58m	day week month year
Water temperature:	9°C	day week month year
Air temperature:	9°C	day week month year
Pressure:	942mbar	day week month year
Air temperature:	21.12°C	day week month year
Humidity:	23.22%	day week month year



Vič

Trnovo Bridge - Trnovo Bridge is a bridge crossing the river Gradaščica in Ljubljana, the capital of Slovenia. It connects the neighborhoods of Krakovo and Trnovo. A bridge had stood on the site since the late 17th century.

Choose site ...

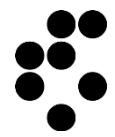


Panoramio Photos are copyrighted by their owners

Select by feature: **Air temperature**
 Voltage camera Humidity Illuminance Pressure
 Voltage (solar) VSN state Watchdog Water level

News
 26.02.11: For developers

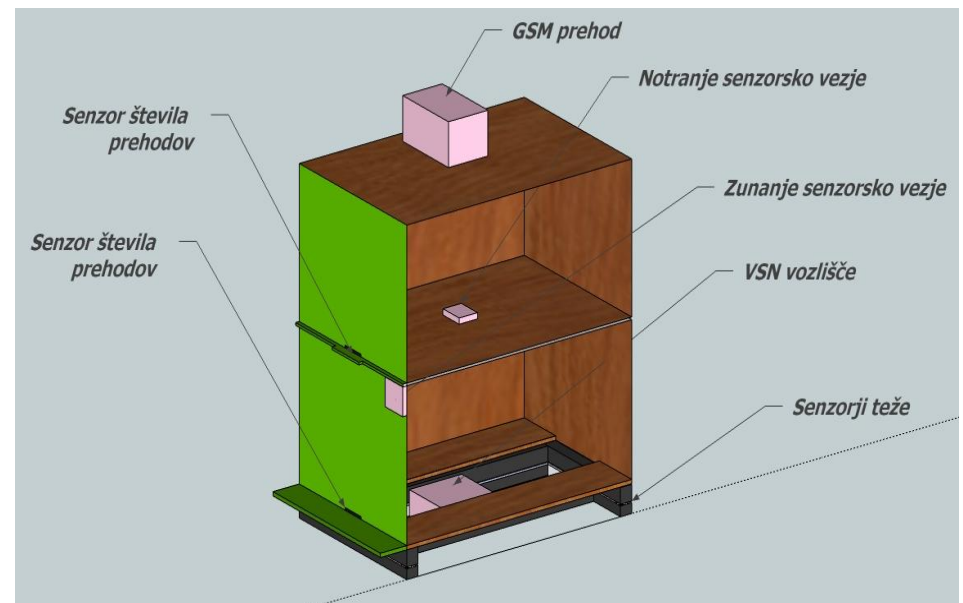
<http://sensorlab.ijs.si/sl/demos.html>



Beehive local climate conditions

The purpose of this testbed is to monitor climate conditions inside (temperature and humidity) and outside (temperature, humidity, air pressure, wind direction and speed) of the beehives.

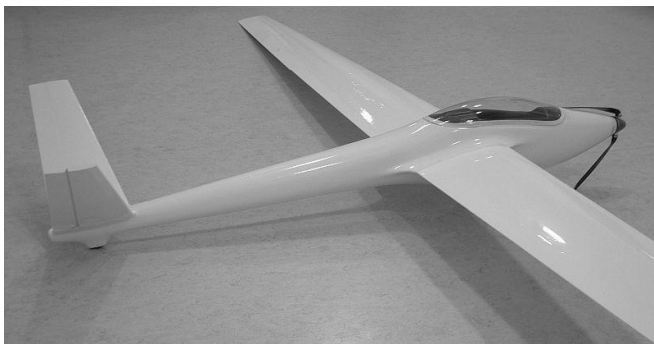
Through bee counting sensor presence of pesticides in the vicinity can be detected. For the test purposes also sound monitoring is possible

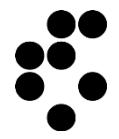


Multispectral imaging and data harvesting over Unmanned Aerial Vehicle (UAV)

Data harvesting over large areas where deployed WSNs have no Internet connection can be a very time consuming and expensive task.

Our solution uses Unmanned Aerial Vehicles (UAV) equipped with a gateway sensor node. In addition, UAV is used to collect multispectral images with a Tetracam ADC camera.



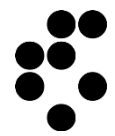


Use cases

Environmental intelligence

Others

- Intelligent buildings
- Smart cities
- Smart infrastructures
- ...



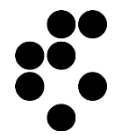
Intelligent building

Berkley: Motescope*

- Soda Hall, the Computer Science building
- Permanent testbeds for research, development and testing
- 78 Mica2DOT nodes



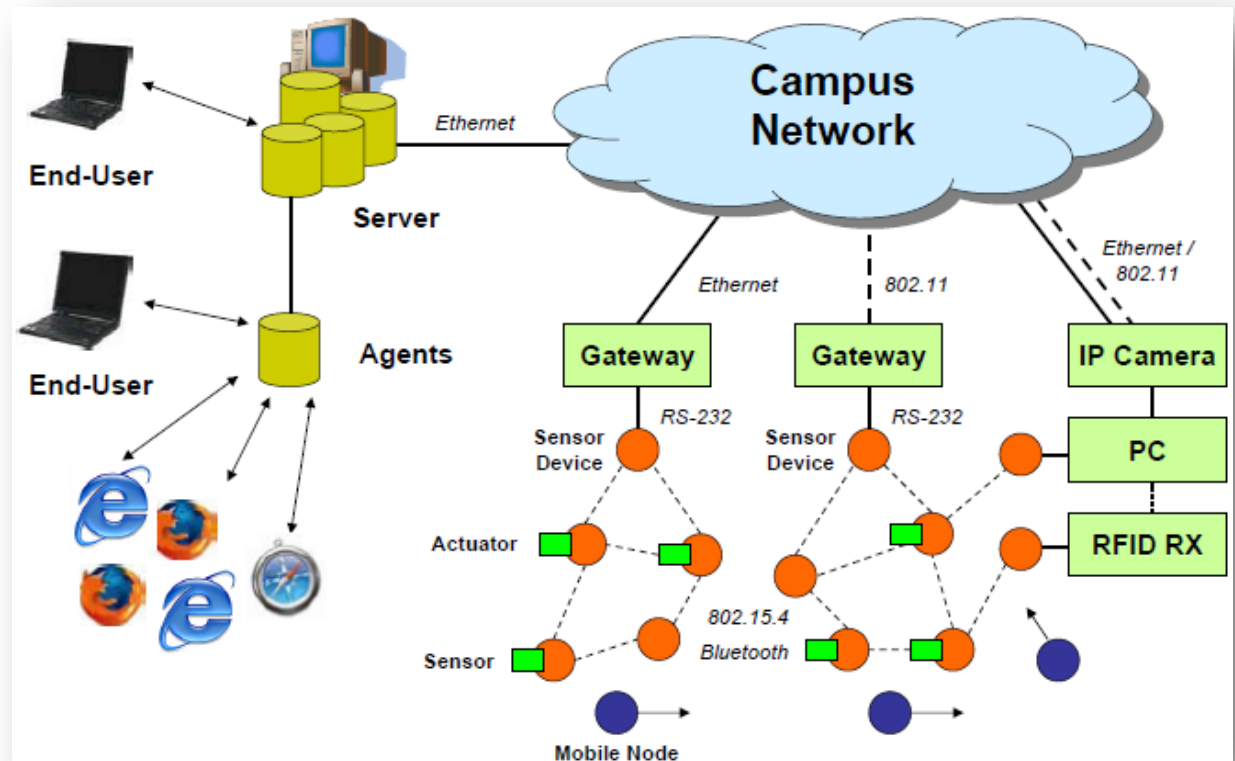
*According to web site visited on Oct 2010.



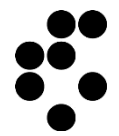
University campus

CMU: [SensorAndrew](#)*

- campus-wide testbed
- Firefly nodes
- Unknown scale



* According to web site on Oct 2010 and tech report from 2008.



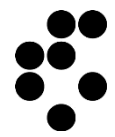
Smart city

MIT: [Senseable City Lab](#)*

- Sensor nodes built into the wheels of bikes
- Unknown number



*Neil Savage, Cycling through Data, Communications of the ACM, Sept 2010.



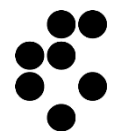
Smart infrastructure

Harward, BBN: CitySense*

- 100 wireless sensors deployed across a city
- Sensor nodes are embedded PC, 802.11a/b/g interface, and various sensors for monitoring weather conditions and air pollutants
- open testbed



* According to web site visited on Oct 2010, last modified in 2008.



Federation of Sensor deployments

Pachube*

- 3700 sensor nodes, over 9400 data streams (April 2010)
- Sensor data and meta-data
- Open to upload/download

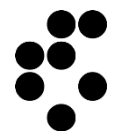
Sensorpedia*

- Similar to Pachube, limited testing Beta

Global Sensor Network*

- Framework for federated testbeds
- Used in the Swiss Experiment

* According to web site visited on Oct 2010.



Outline

Part III. Demos, Tools & Research directions

Use cases

- What systems and prototypes exist?

Open problems

- Are there unsolved problems?

Summary

- What was this tutorial about?

List of sources for further studies

- Where to start digging?



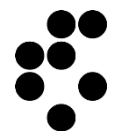
Current state and open problems with respect to Sensor Nodes

WSN is

- Well developed field with many degrees of freedom
- Complex, large-scale, resource constrained systems
- Focus is on intra network communications

Efficient management and maintenance of the “things”

- Remote reconfiguration of parameters
- Remote software updates
- Real implementations solving real problems, particularly large scale (see next slide)



Myths & lessons regarding Sensor Networks

Myth #1: Nodes are deployed randomly.

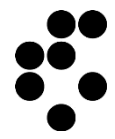
Myth #2: Sensor nodes are cheap and tiny.

Myth #3: The network is dense.

Lesson #1: It's all about the data.

Lesson #2: Computer scientists and domain scientists need common ground.

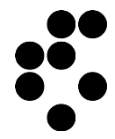
Lesson #3: Don't forget about the base station!



Challenges with respect to conceptualization

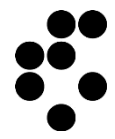
WoT covers a long pipeline of technologies from sensors to high level services

- ...current ontologies are covering just parts of the space and are yet to be interlinked
- ...ideally, sensor network domain should be linked to general common-sense ontologies and further to domain specific service ontologies



Challenges with respect to analytics & CEP

- Traditional mining and analytic techniques are not ready for the scale and complexity coming from large sensor setups
- ...in particular:
 - connection to background knowledge (ontologies) for enrichment of sensor data for expressive feature representations needed for analytic techniques
 - "complex events" are in the context of WoT much more complex compared to traditional "complex events" research
 - real-time response on complex events appearing in WoT setups



Outline

Part III. Demos, Tools & Research directions

✓ Use cases

- What systems and prototypes exist?

✓ Open problems

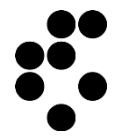
- Are there unsolved problems?

Summary

- What was this tutorial about?

List of sources for further studies

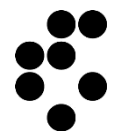
- Where to start digging?



Summary

The tutorial had 3 parts:

1. Motivation & background
 - Problems that the Web of Things can solve
 - Components and complexity of the system, from “Things” to Apps and Services
 - Quick start recipes
2. Technology and tools for exploiting the WoT
 - Semantic aspects
 - Analytic aspects
 - Services
3. Demos, Tools & Research directions
 - Overview of existing setups and tools used for their implementation
 - Research directions



Outline

Part III. Demos, Tools & Research directions

✓ Use cases

- What systems and prototypes exist?

✓ Open problems

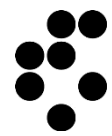
- Are there unsolved problems?

✓ Summary

- What was this tutorial about?

List of sources for further studies

- Where to start digging?



Relevant Conferences

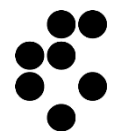
- **WWW** - International World Wide Web Conferences
- **ICML** – International Conference of Machine Learning
- **NIPS** – Neural Information Processing Systems
- **KDD** – ACM Knowledge Discovery in Databases
- **ICWS** - IEEE International Conference on Web Services
- **ISWC** – International Semantic Web Conference

- **IPSN** – Information Processing in Sensor Networks
- **Percom** - IEEE Pervasive Computing and Communication
- **SenSys** - ACM Conference on Embedded Networked Sensor Systems
- **MobiSys** - International Conference on Mobile Systems, Applications, and Services
- **INSS** – International Conference on Networked Sensing Systems
- **DCOSS** - International Conference on Distributed Computing in Sensor Systems

- **iThings** - IEEE International Conference on Internet of Things

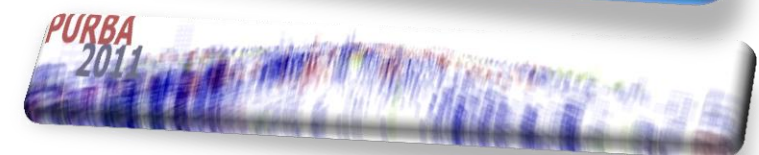
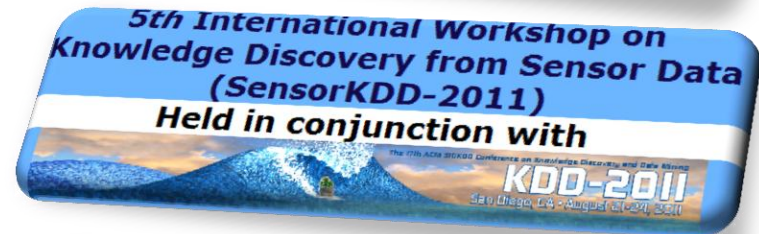
Apps and
Services

“Glue”

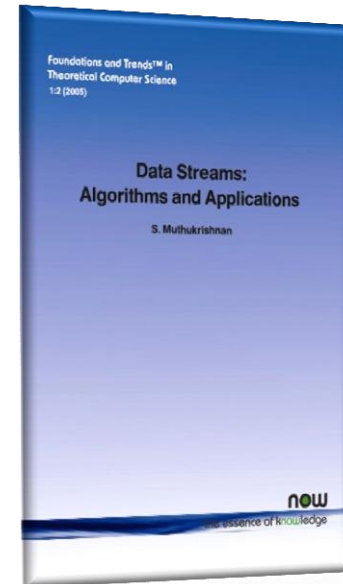
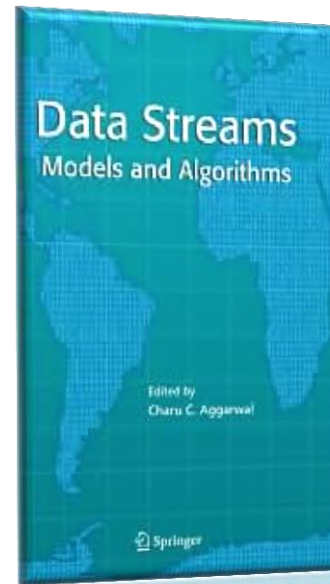
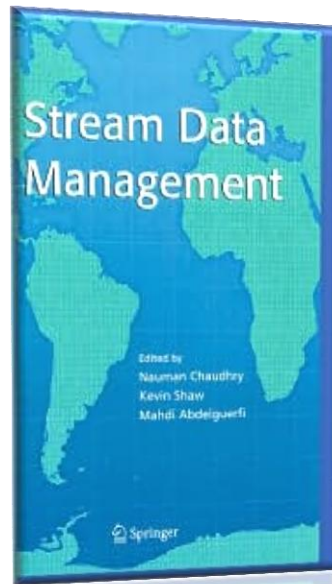
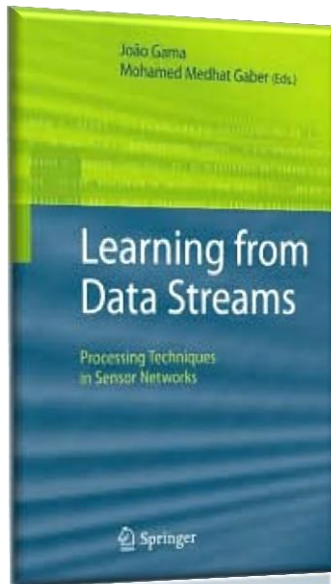


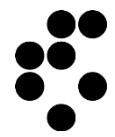
Relevant Workshops

- **WebOfThings** - International Workshop on the Web of Things
- **SensorKDD** - International Workshop on Knowledge Discovery from Sensor Data
- **PURBA** - Workshop on Pervasive Urban Applications
- **Urban-IOT** – the Urban Internet of Things Workshop
- **Web Enabled Objects** - International Workshop on Web-Enabled Objects
-

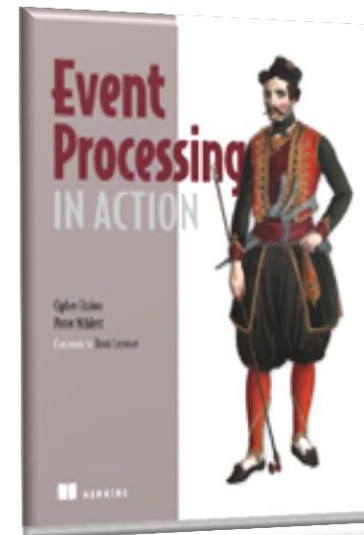
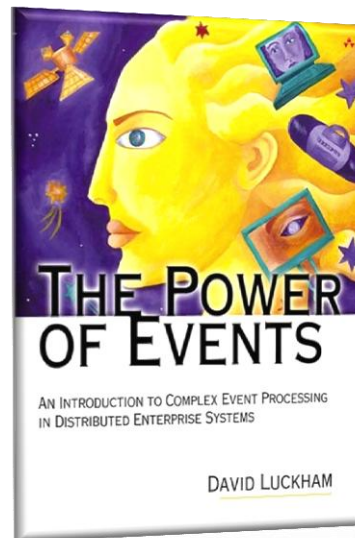


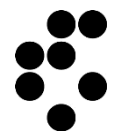
Books on data streams



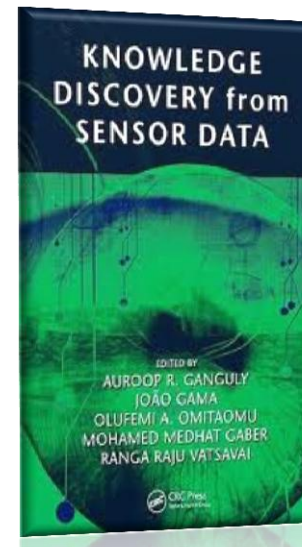
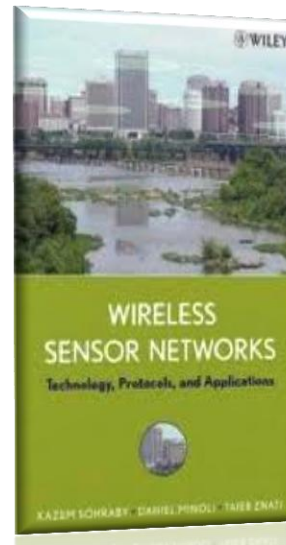
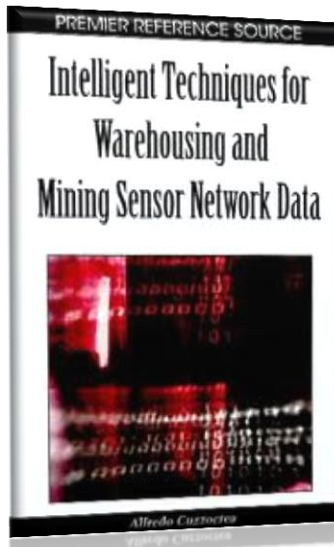


Books on event processing





Books on sensor networks



Relevant blogs

- Web of Things Blog
- Wireless Sensor Network Blog
- The Internet of Things
- Dust Networks – In the News
- ReadWriteWeb



Related Wikipedia Links

Data Stream Mining:

http://en.wikipedia.org/wiki/Data_stream_mining

Complex Event Processing:

http://en.wikipedia.org/wiki/Complex_Event_Processing

Real Time Computing:

http://en.wikipedia.org/wiki/Real-time_computing

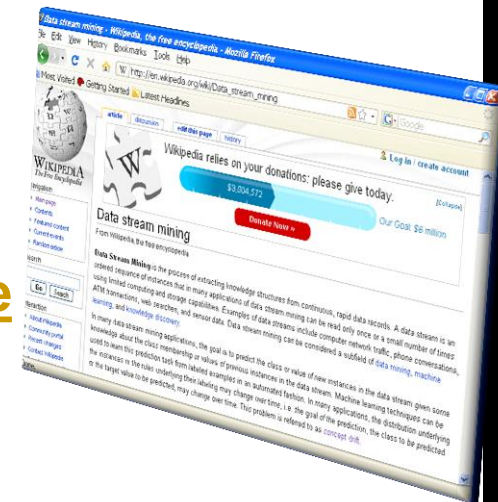
http://en.wikipedia.org/wiki/Real-time_computing

Online Algorithms:

http://en.wikipedia.org/wiki/Online_algorithms

Worst Case Analysis:

http://en.wikipedia.org/wiki/Worst-case_execution_time



Related Wikipedia Links

Web of Things:

http://en.wikipedia.org/wiki/Web_of_Things

Internet of Things:

http://en.wikipedia.org/wiki/Internet_of_Things

Wireless Sensor Networks:

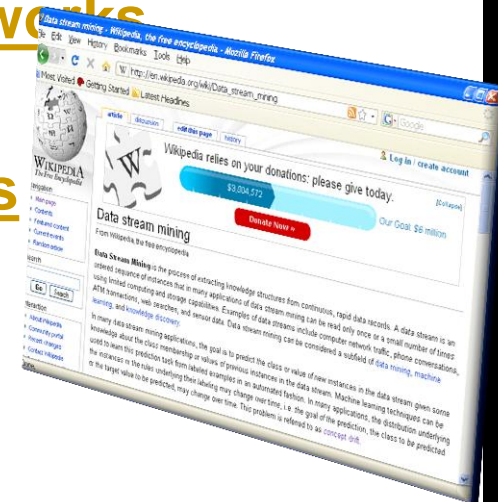
http://en.wikipedia.org/wiki/Wireless_Sensor_Networks

Major Appliance:

http://en.wikipedia.org/wiki/Household_appliances

RFID – Radio Frequency Identification:

<http://en.wikipedia.org/wiki/RFID>



Video Tutorials

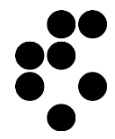
State of the Art in Data Stream Mining: Joao Gama, University of Porto

- http://videolectures.net/ecml07_gama_sad/

Data stream management and mining: Georges Hebrail, Ecole Normale Superieure

- http://videolectures.net/mmdss07_hebrail/

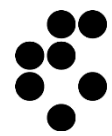




Outline

Part III. Demos, Tools & Research directions

- ✓ **Use cases**
 - What systems and prototypes exist?
- ✓ **Open problems**
 - Are there unsolved problems?
- ✓ **Summary**
 - What was this tutorial about?
- ✓ **List of sources for further studies**
 - Where to start digging?



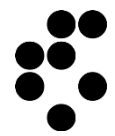
Thank you!

CAROLINA.FORTUNA@IJS.SI

MARKO.GROBELNIK@IJS.SI

[HTTP://SENSORLAB.IJS.SI](http://SENSORLAB.IJS.SI)





Help us improve the tutorial!

Send comments and relevant info to

carolina.fortuna@ijs.si

Acknowledgements

We would like to thank Miha Smolnikar, Kemal Alic, Zoltan Padrah, Klemen Kenda and Alexandra Moraru for contributing some slides, and the SensorLab team for their support.



envision
environmental services infrastructure with ontologies