Monitoring environment-parameters for research towards energy-efficient buildings

Speaker's Intro

Speaker

Bart Voet

Day Job

Practice Lead Java Development at AXA

Evening and weekend

Family

Programming and hacking

Learning electronics

Skateboarding and snowboarding

...

Intro

What is the content of this thesis?

Stakeholders

Primary stakeholder

Division of Building Physics KUL
(Department Civil Engineering)

System capturing data from digital sensors

Scalable to different scenario's

Research facility

Educational environment

External research

Stakeholders

Stakeholder

Groep T

Evolution

Smart sensors

Digital interfaces

Sensor networking

New devices (and open source)

Rapsberry Pi, BeagleBoard, Cubieboard

AVR, Arduino, ...

...

Stakeholders

Stakeholder

Author (and other hobbyist)

Learning platform

Experienced Java Developer

Learning embedded development

Open source platform

Scalable to different devices

Focused on monitoring

Integrable in different scenario's

Requirements and scope

What is expected?

Startup requirements

Context

Department of Building Physics KUL

performing

research on energy **efficient buildings**

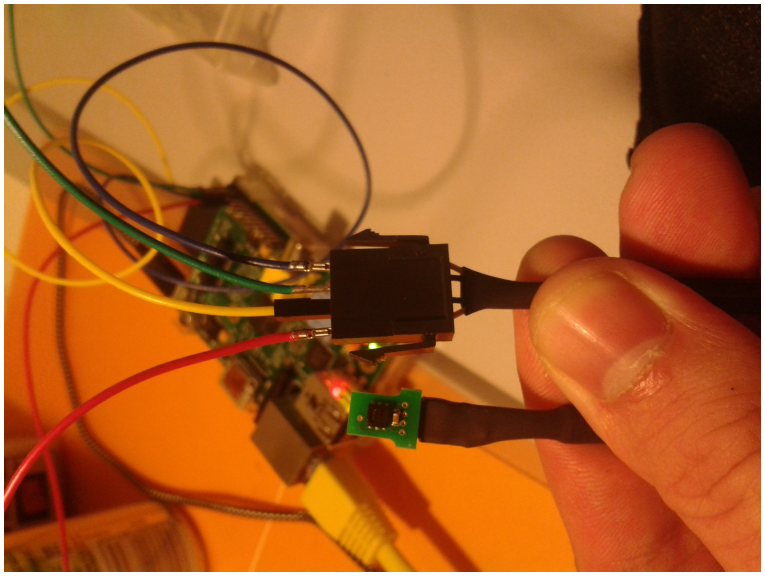
Startup requirements

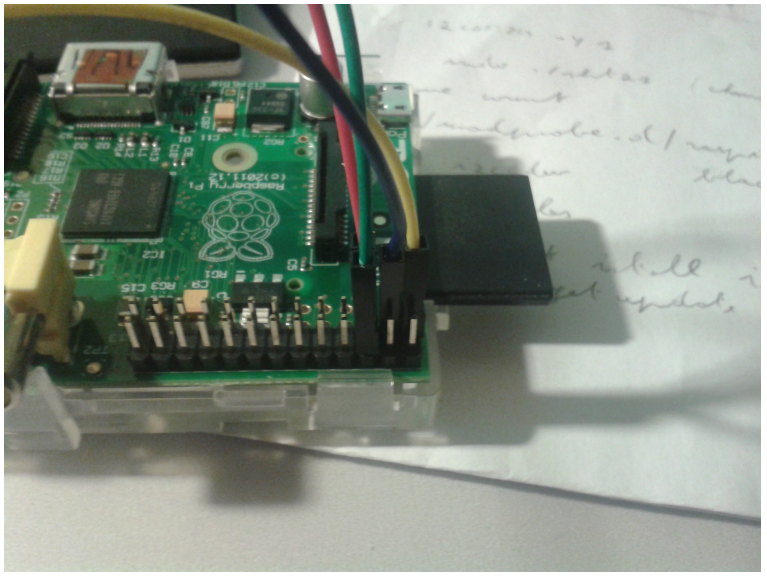
Demand

- System(s) for continuous measurement that is
  - Reliable
  - Inexpensive system(s)
  - Continuous measurement

Startup requirements

Use **Raspberry Pi** as a device for **sensing**





Startup requirements

Use Raspberry Pi as a device for **sensing** environment **parameters**

- Important for
  - Indoor climate
  - Energy consumption
- Example given
  - Relative humidity
  - Temperature
  - Differential pressure
- … and **other** measurements in the future

Startup requirements

More specifically, use Raspberry Pi as a device to

- **Control** and **configure** sensors
- **Collect** data
- **Store** sensor measurements
  - (for later evaluation and analysis)
- **Correlate** stored measurements
  - configuration
  - timing

Startup requirements

Using sensors (Sensirion)

- SHT21 (STS21-SHT25)
  - temperature
  - relative humidity

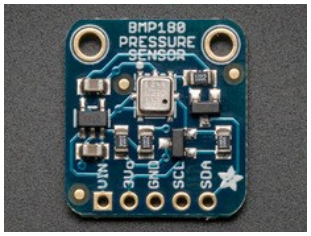
- SDP600 (later phase)
  - differential pressure

Startup requirements

(Not requested by stakeholder)

BMP180 (Bosh)

Barometric pressure



DHT11

Temperature

User profile

Taking into account profile of the users

- Students
- Researchers

Assuming only basic knowledge of

- Electronics
- Command line
- High level programming construct

(but not advanced)

Scope and focus

- Scope limited to **digital** (**smart**) sensors
  - Digital interface (i2c, spi or custom)
  - Integrated MCU performing
    - Calibration
    - Linearization
- No focus on **classic sensors**
  - Manual calibration
  - Precision resistors
  - …
  - (**although not excluded**)

Scope and focus

- Consequence
  - Focus on **system design** (not hardware design)
    - Integration
    - Extensibility
    - Ease of use
    - Reliability
    - Documentation
    - …

Scope and position

Consequence

Focus on a **pluggability** and **portability**

Different hardware

Different systems (OS or not)

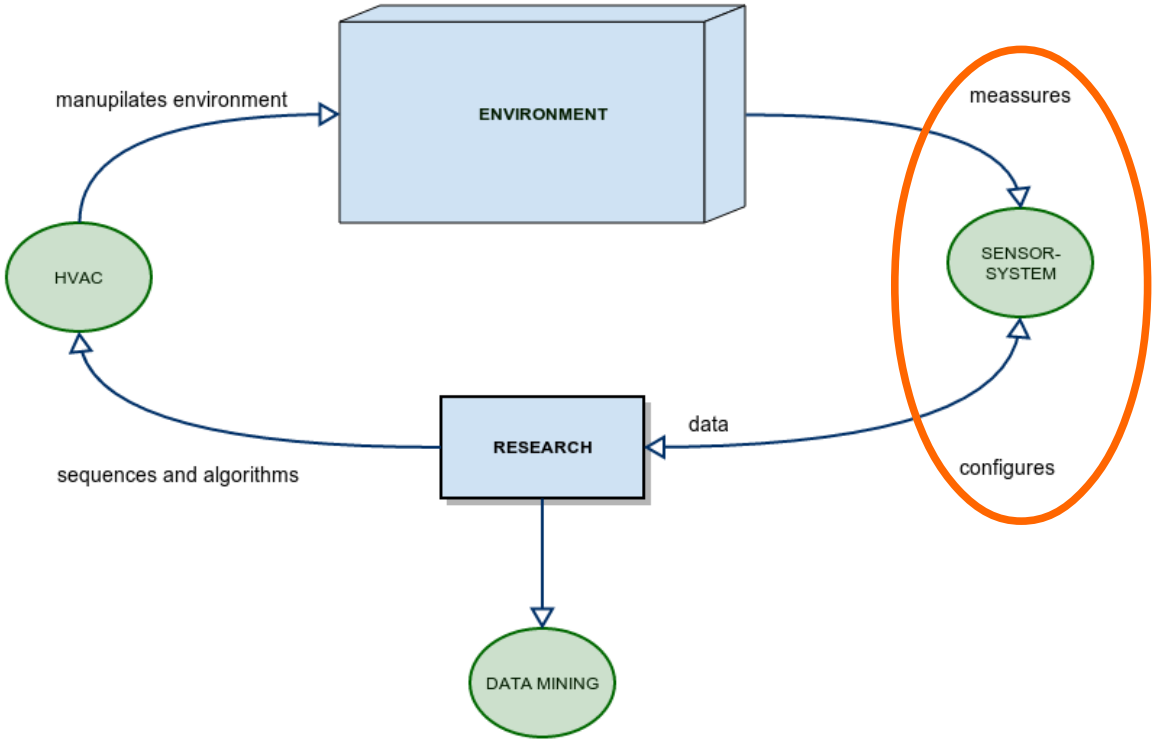
Different network-integration

...

Open for **extension**
Closed for **modification**

Environment

Research-process and its components



Scenario's

Primary scenario: Research-facility in Gent

- HVAC-infrastructure deployed
- Electricity and ethernet

- Deploy, install and configure sensors
- Aggregate data
- Long time

Scenario's

Scenario: Educational environment (students)

- Class-room environment
- Labo

- Experiment and learn
- Explore sensors

Scenario's

Scenario: Large buildings

- Mobile scenario
- No HVAC
- No assumptions on
  - Electricity
  - Network

- Similar to primary scenario
- More constrained environment

System Challenges

TOP Challenges

- **Multiple sensors**
  - **Concurrent** access
    - e.g. Sensirion-sensors having same i2c-adress (40)
  - **Large** area's
    - i2c and spi not developed for long distance (even if you lower the clock)

System Challenges

TOP Challenges

- **Reliability**, **durability** and **resilience**
  - Ability to recover from
    - Power interruption
    - Network incidents **...**
  - Alerting-capability
    - Sensor goes down
    - Errors coming from sensors
    - Processing device not working
    - Heating **...**

System Challenges

- TOP Challenges
  - **Usability**
    - Scalable to different scenario's
    - Users are no software- or hardware-engineers
    - Need an interface that's
      - Easy to integrate with other systems
      - Easy to integrate in personal computing (structured txt-files)

System Challenges

TOP Challenges

- **Extensibility**
  - Adding new sensors and configure
  - Adding new sensor-type without changing the system (open-closed-system)
- **Configurability**
  - Changing sensor-parameters
  - Changing scheduling

System Challenges

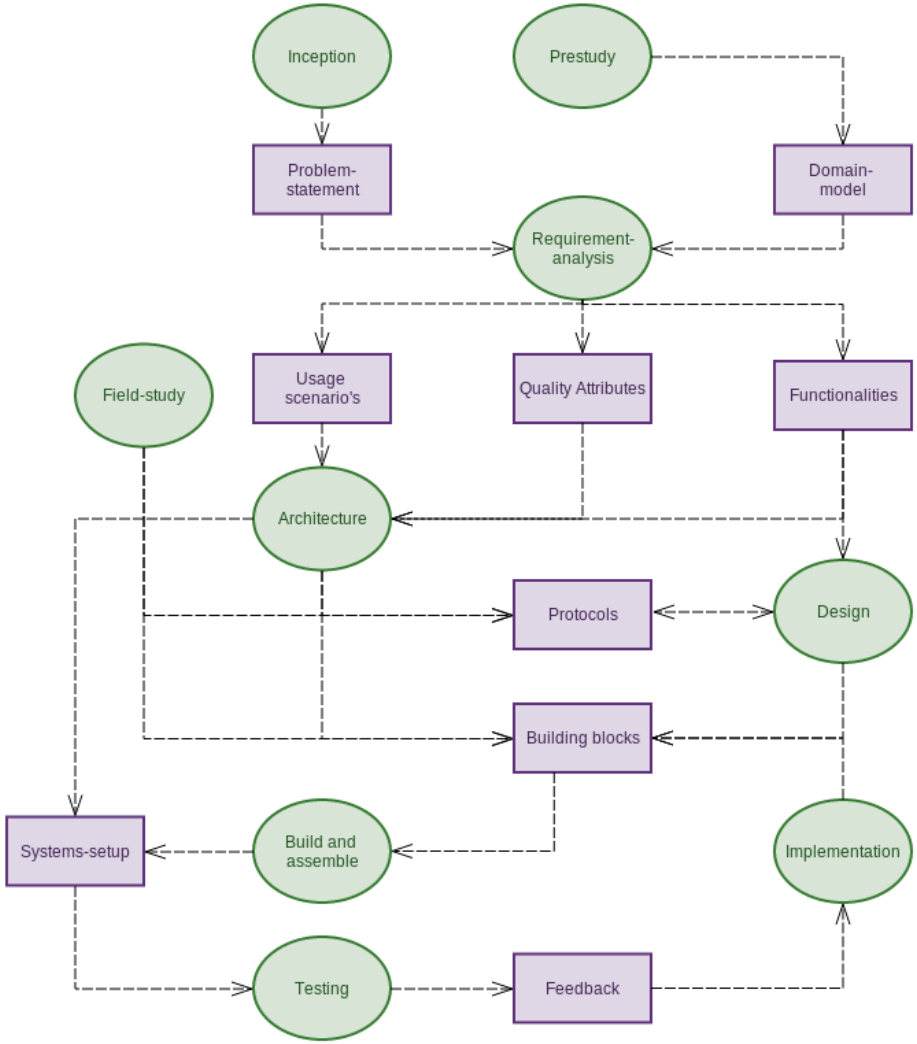
- TOP Challenges
  - **Correlation and timing**
    - Measurements need to be correlated to
      - Time
      - Configuration of sensors

Approach

Process of development?

Building blocks?

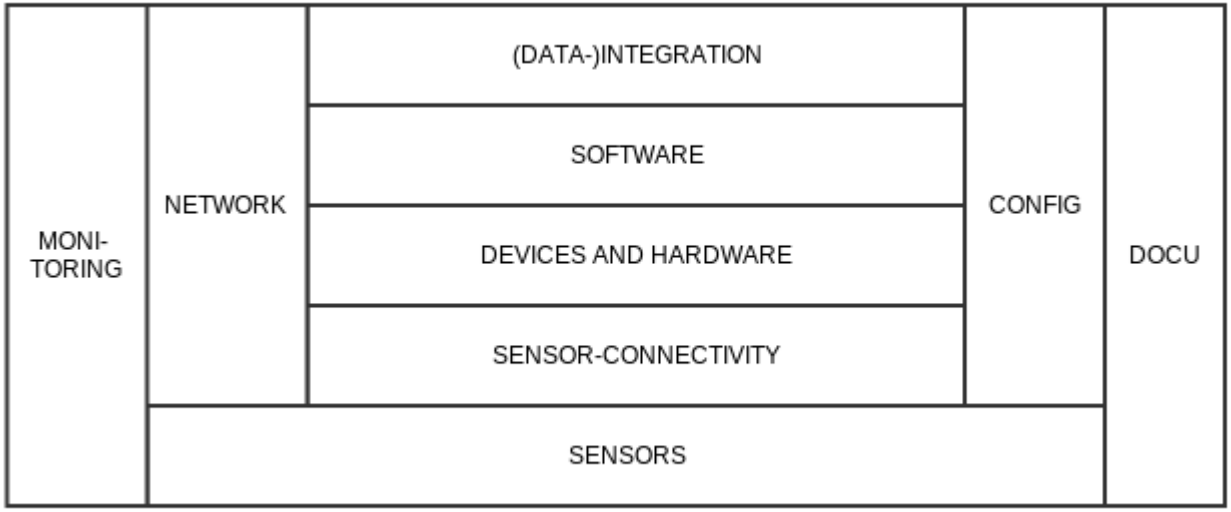
Approach


Inception
Prestudy
Problem-statement
Domain-model
Requirement-analysis
Field-study
Usage scenario's
Quality Attributes
Functionalities
Architecture
Protocols
Design
Building blocks
Systems-setup
Build and assemble
Implementation
Testing
Feedback

Approach

Category of building blocks (solutions)

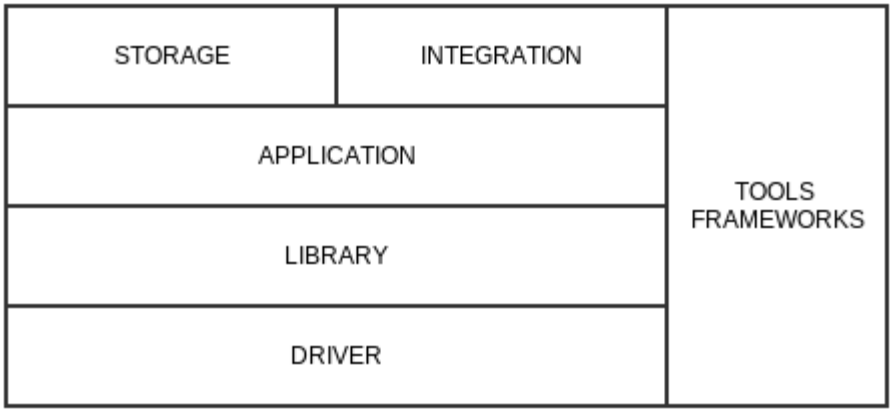
- Hardware, software, documentation
- Serve as annotation in thesis



Approach

Category of solutions

- Hardware, software, documentation
- Serve as annotation in thesis



Design and architecture

Concept?

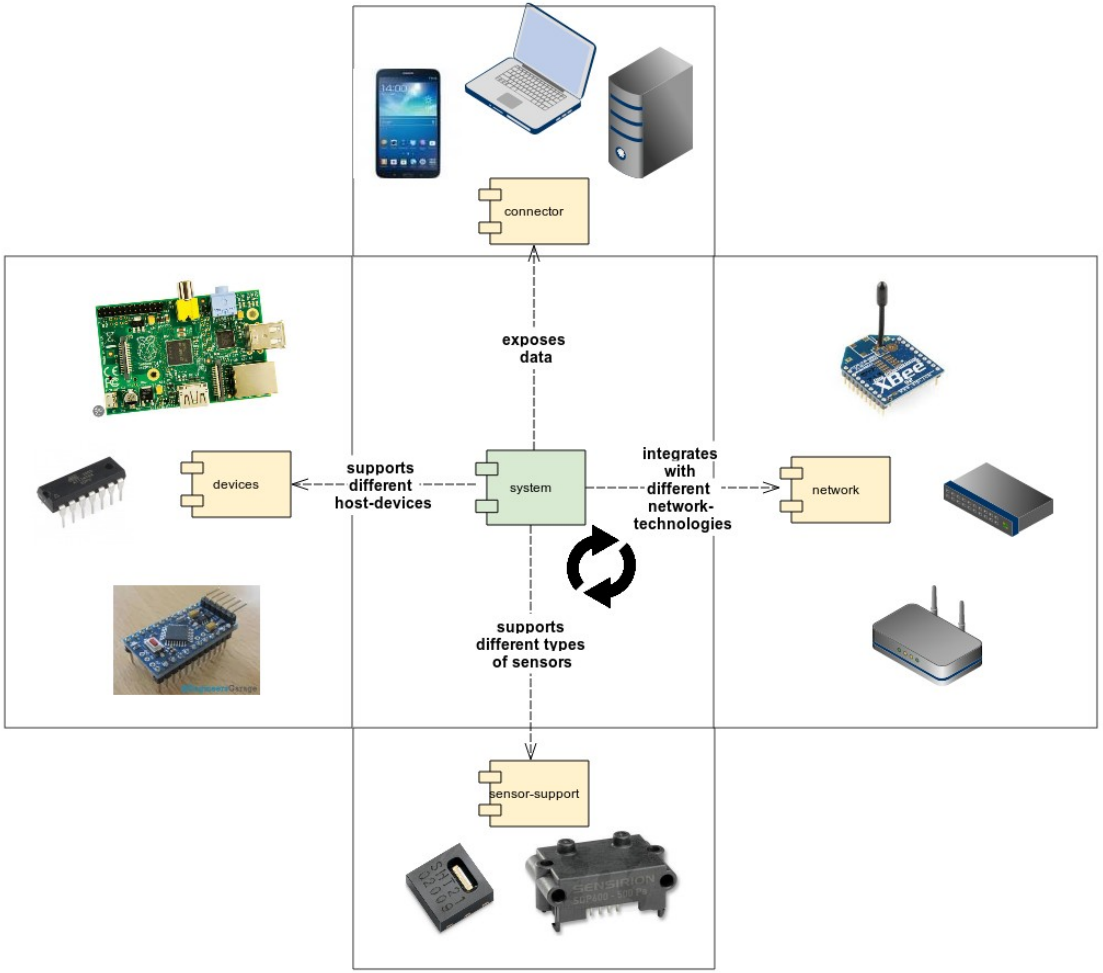
Principles?

Design?

Building blocks?

System Concept

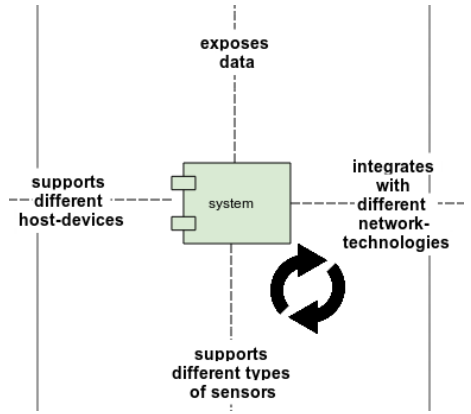
System-concept: **runtime and dependencies**



System Concept

System-concept **Runtime**

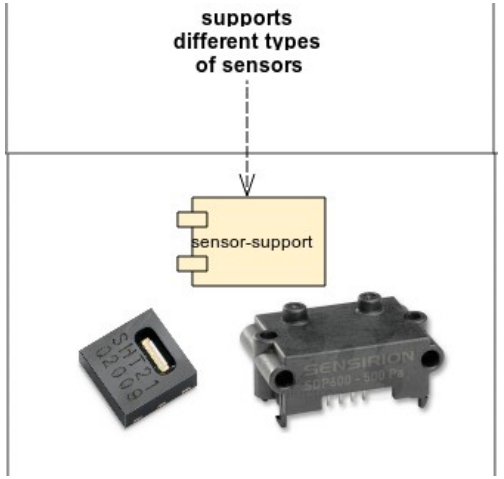
- Scheduling measurements
- Relying on system abstractions
  - Integration
  - Storage
  - Device-abstractions
  - More sensors
  - Of different types



System Concept

System-concept: **Support different sensors**

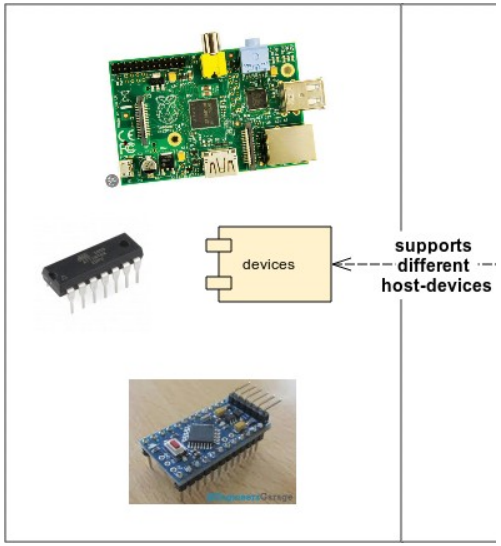
- System supports
  - Extracting data from multiple Sensors
  - Multiple types via Sensor-abstraction



System Concept

System-concept: **Support different devices**

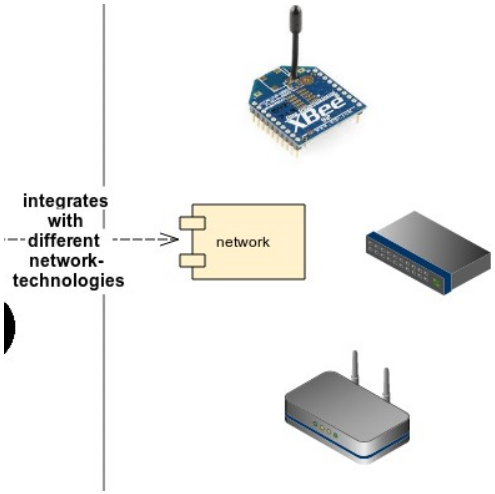
- Isolate system-dependencies
  - Scheduling
  - Digital interfaces
  - ...
- Support for
  - Low level (c-api)
  - High level (java)



System Concept

System-concept: **Network independence**

- Integration-capability isolated
  - Local integration
  - Zigbee
  - WIFI
  - MQTT
  - ...

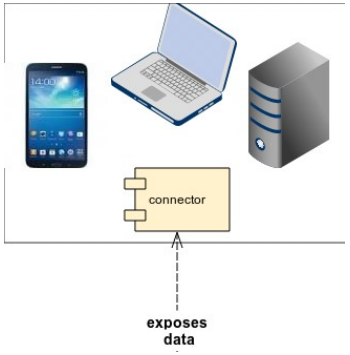


!! System provides abstraction and pluggability to adapt, not all implementations exists !!

System Concept

System-concept: **Data exposure**

- Connectors for clients
  - Open protocols to integrate with various kind of devices



- Current provided protocol
  - REST exposing
    - JSON
    - CSV (under construction)

Development principles

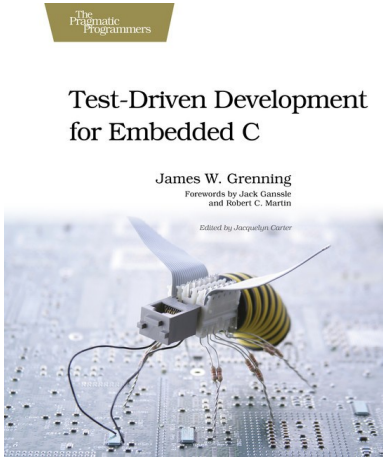
S Single Responsibility Principle

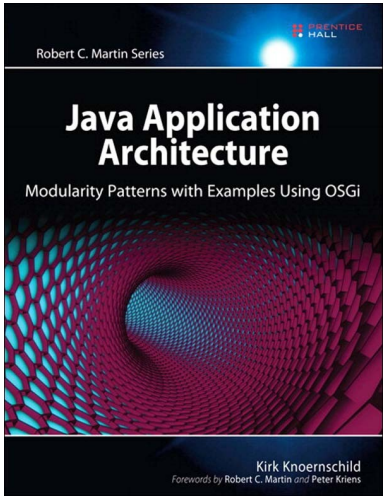
O Open Closed Principle

L Liskov Substitution Principle

I Interface Segregation Principle

D Dependency Inversion Principle

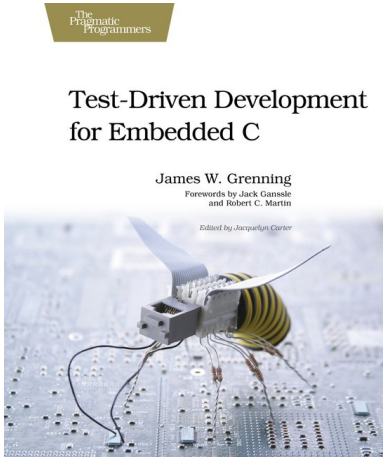


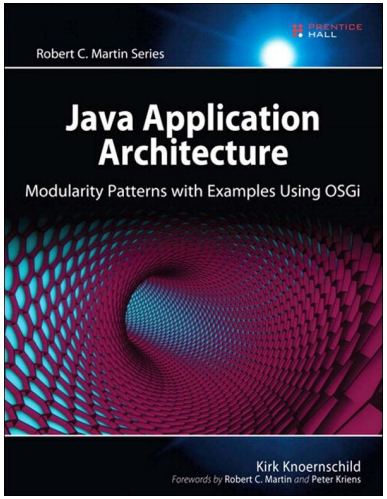


Development principles

SOLID

- Introduced by Robert C-Martin
- Principles for improving
  - Flexibility
  - Extensibility
  - Modularity
  - Testability
- Introduced in OO but applicable to all programming-paradigms





Development principles

Modularity

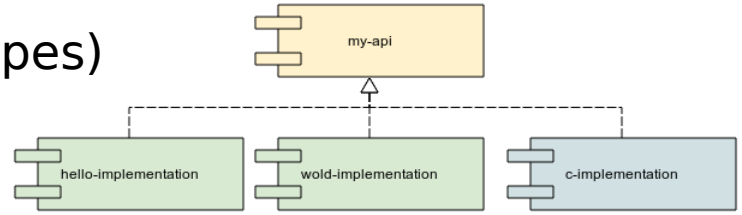
Package code and classes into reusable and composable package

Provide

api-components

(interfaces and types)

concrete implementations



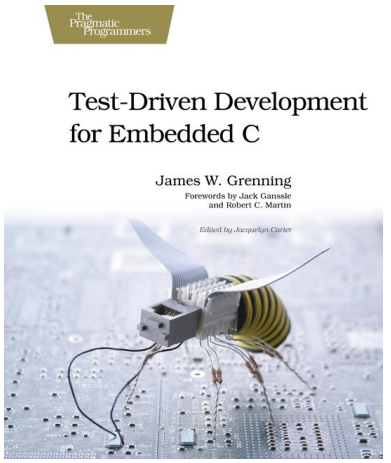
Code needs to be **SOLID**

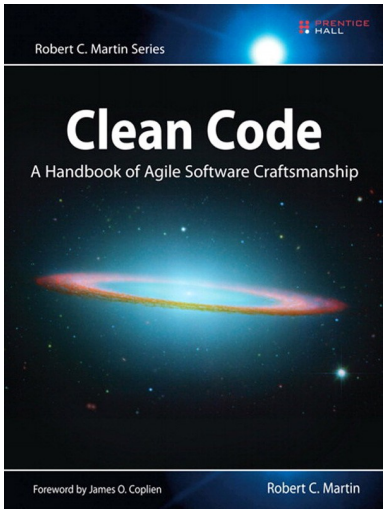
Development principles

Test Driven Development (**TDD**)

- Drive your code trough tests
- Just enough code
- Isolate dependencies

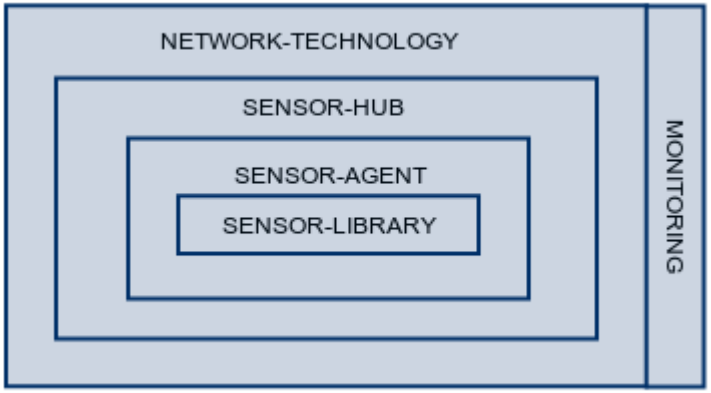
(Discovered/invented by Kent Beck)





System Architecture

Design-concept: **Layering**

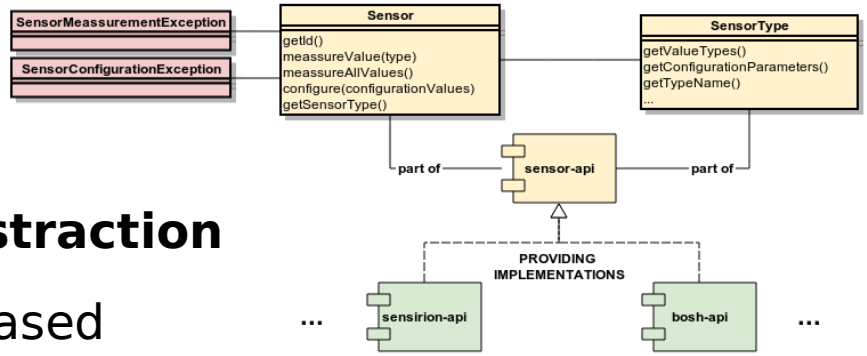


Different building blocks

- Built on top
- Inner layers can be used independently
- Segregation by interface

System Architecture

Layer: **Sensor-API**



Sensor- and SensorType-abstraction

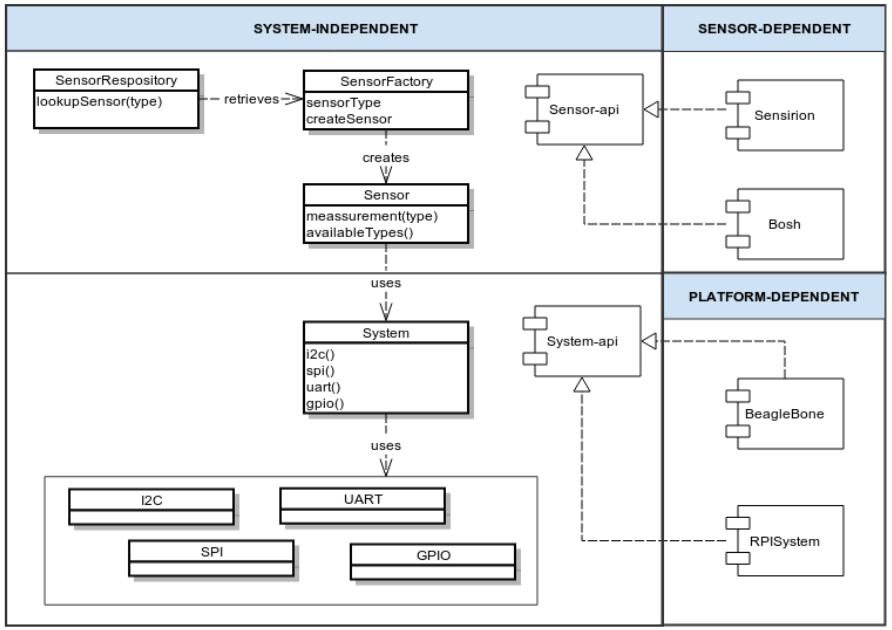
- Interfaces key-value pair based
- Modules containing concrete implementations
- Standardized exceptions

Goal

- Provide a repository for reuse (Github-project)
- Isolate the processing logic
- Provide an abstraction layer for Sensor-agent
- Translate the datasheet behind an abstraction

System Architecture

Layer: **Sensor-API**



System-abstraction of

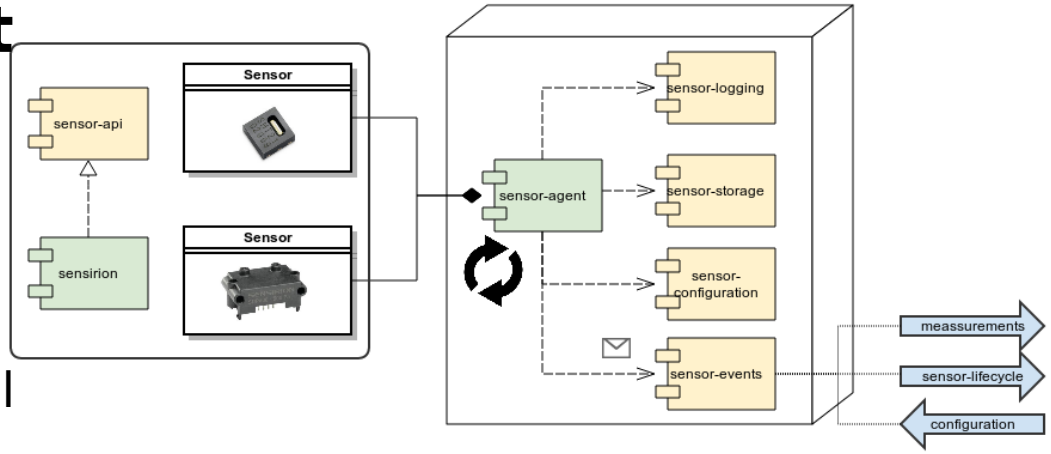
- Digital interfaces (i2c, spi, uart)
- Pin-access
- Timing

Goal

- Portability (vs scenario's)
- Choice of libraries (e.g. RPi can work JME or Pi4j)

System Architecture

Layer: **Sensor-Agent**



Runtime or application

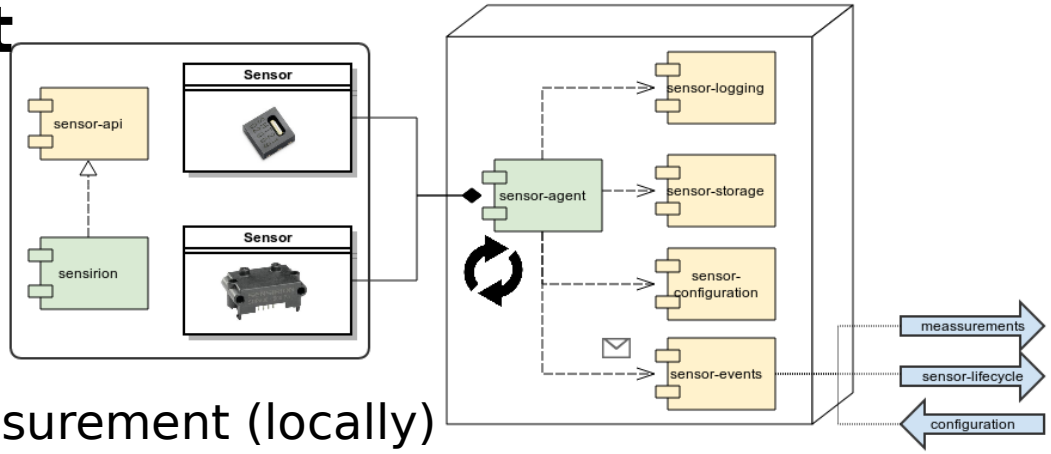
- Captures data at interval
- Manages sensors via sensor-api-abstraction
- Notifies and communicates via sensor-events

Goal

- Use the sensor-api without low-level coding
- Set up a measurement system based on configuration
- Plug-in architecture for cross-cutting-concerns

System Architecture

Layer: **Sensor-Agent**



Depends on abstractions

Logging

Storing the sensor-measurement (locally)

Storing the sensor-**configuration**
(might be another storage-medium than measurement)

Integrates with the outside-world via sensor-**events**

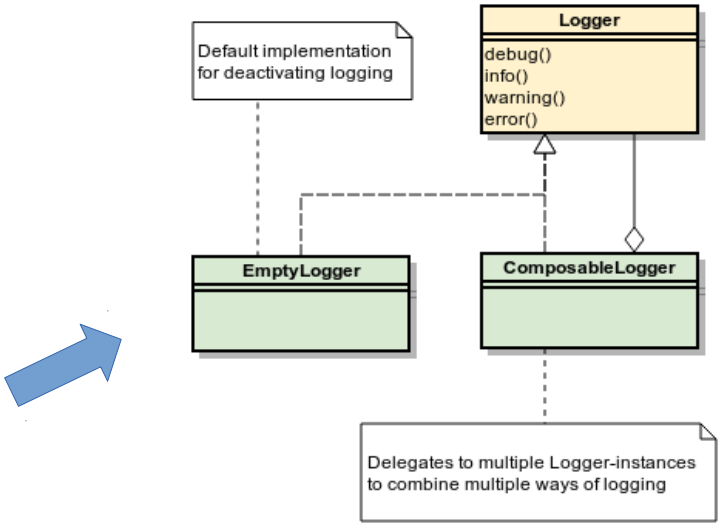
New measurement (out)
Sensor activated or reconfigured (out)
Instructions for reconfiguration (in)

System Architecture

Layer: **Sensor-Agent**

Abstractions

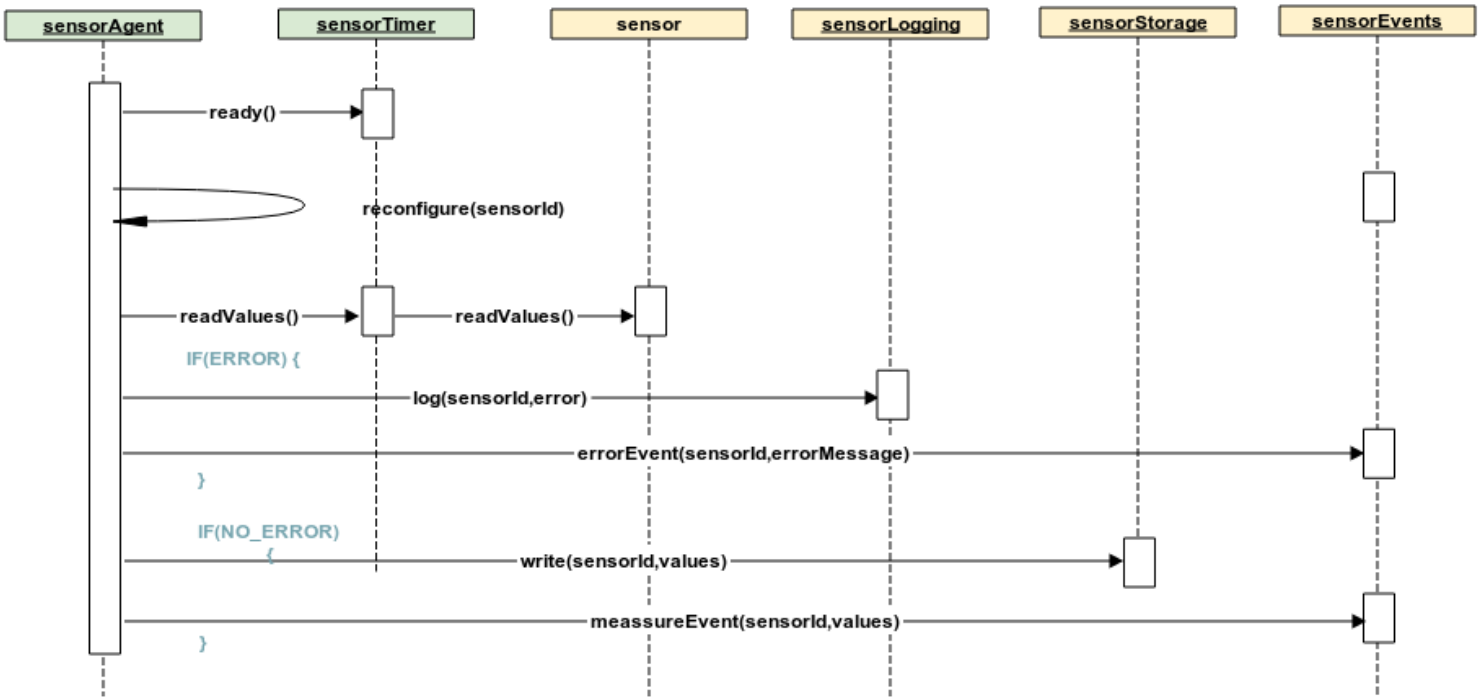
- are interchangeable
  - interface segregation
  - dependency injection
- can be combined (or composed)
- are deactivated by default
  - by default empty implementations



System Architecture

Layer: **Sensor-Agent**

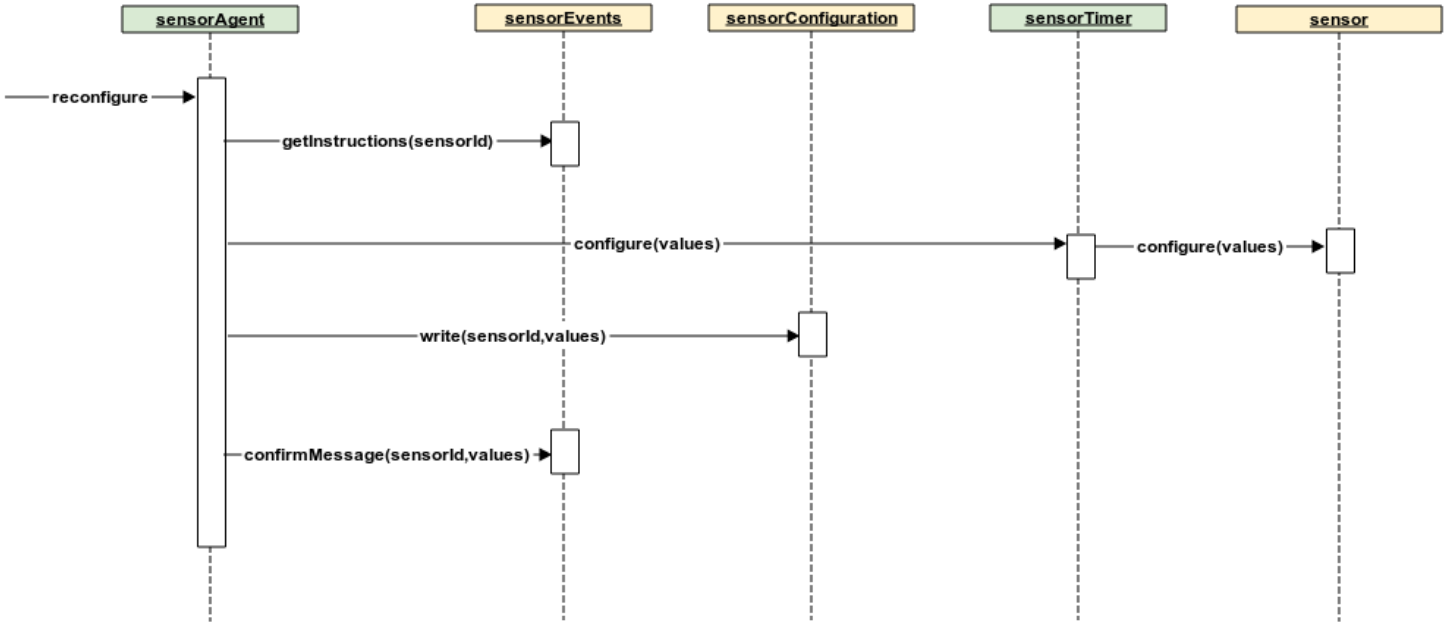
Runtime sequence



System Architecture

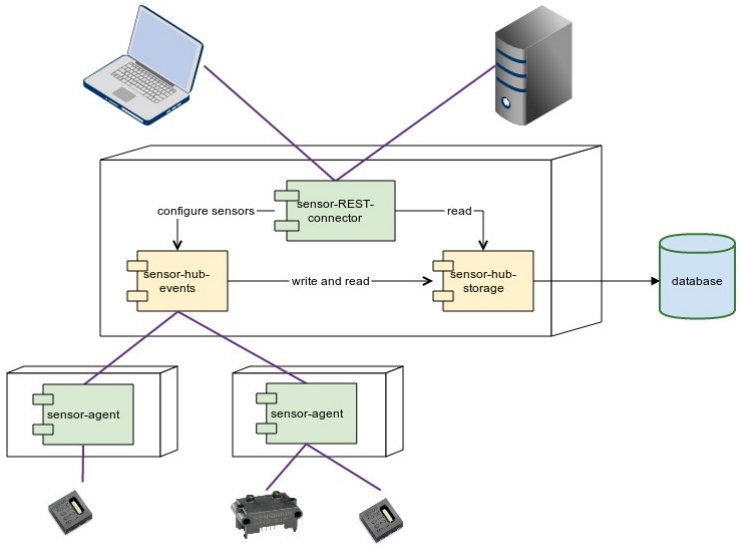
Layer: **Sensor-Agent**

Configuration sequence



System Architecture

Layer: **Sensor-Hub**



Runtime or (web-)application

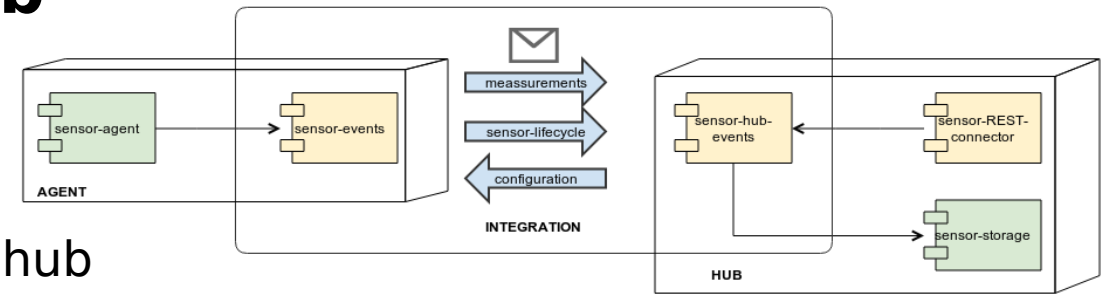
- Communicating with agents
- Centralizing data-storage
- Exposing data to users (and other devices/servers)

Goal

- Collecting and storing data from different sensors
- Enable user to query the data
- Interface for configuring remotely the sensors

System Architecture

Layer: **Sensor-Hub**



Link between agent and hub

- Measurements are pushed
- Instructions are forwarded to
- Events
  - Confirmation of configuration
  - Errors

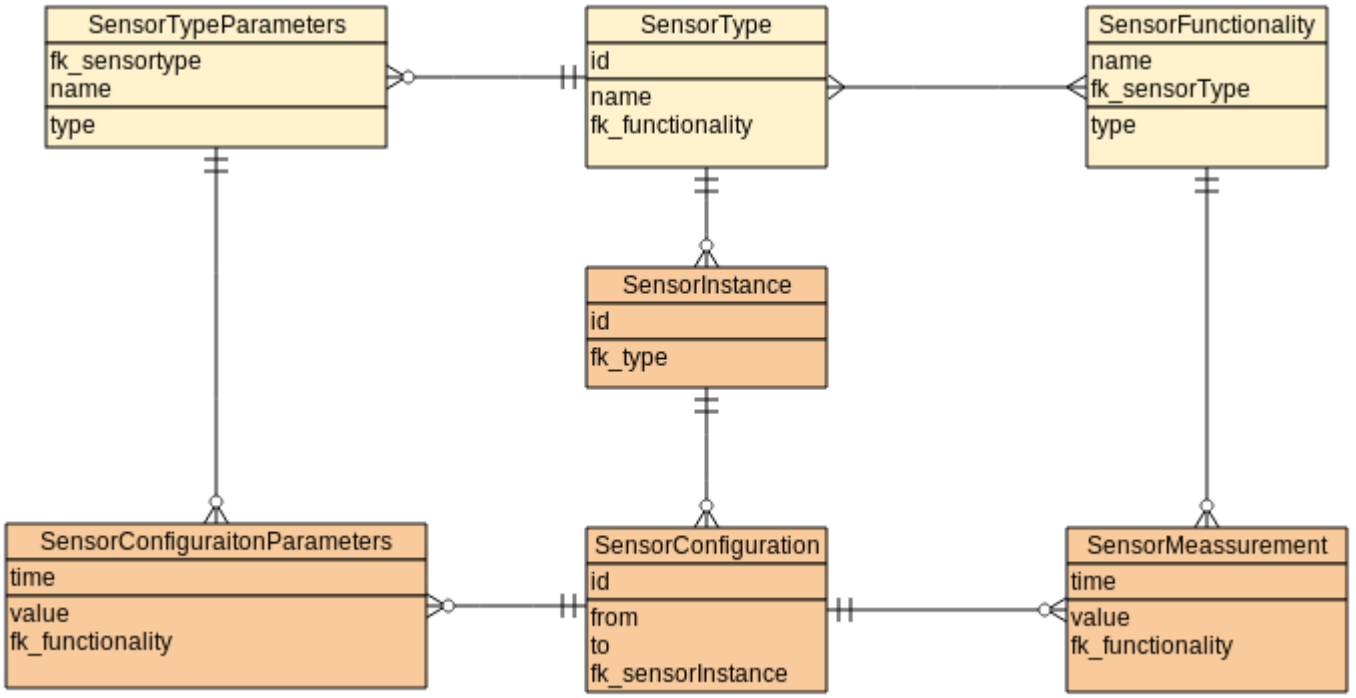
Sensor-hub-events and sensor-events

- Should integrate with same protocol
- Code is message based

System Architecture

Layer: **Sensor-Hub**

Datamodel



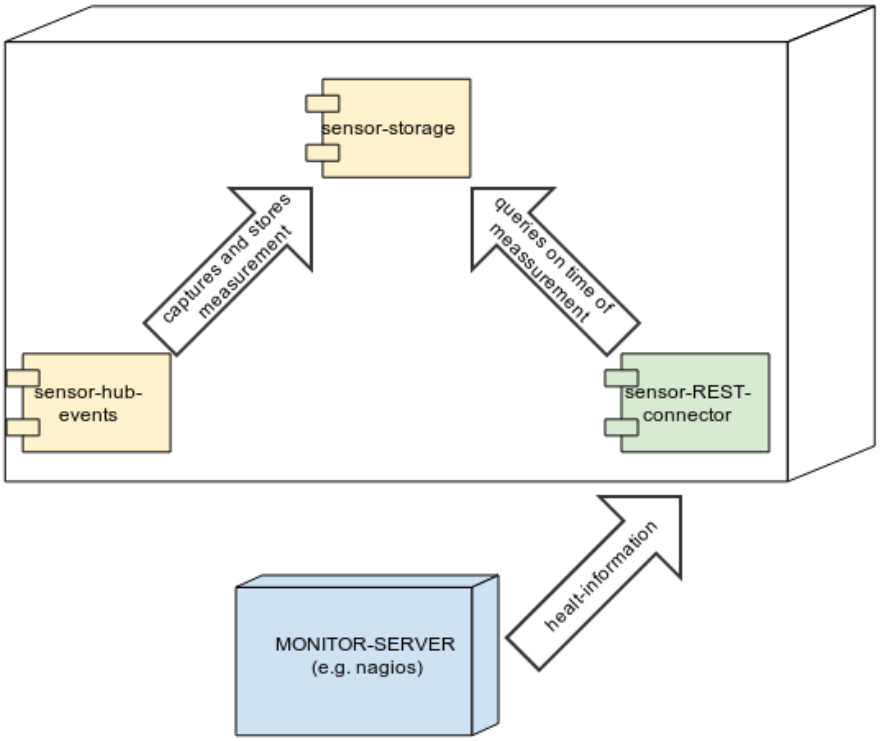
System Architecture

Layer: **Monitoring**

Monitor tool can query for

- Error events
- Deactivated agents or sensors (activity-monitoring)

REST-interface



In practice

Today?

Status?

Future?

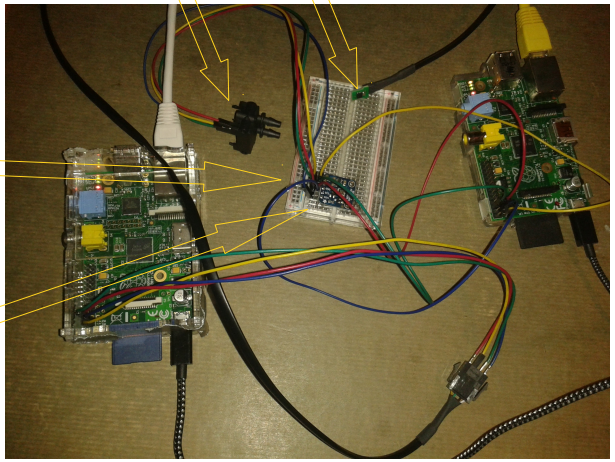
Today

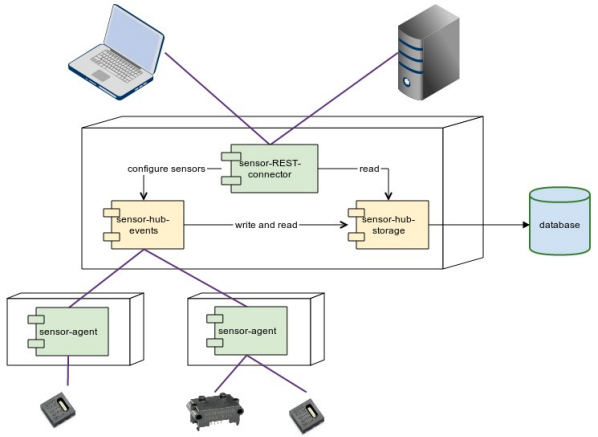
SDP600

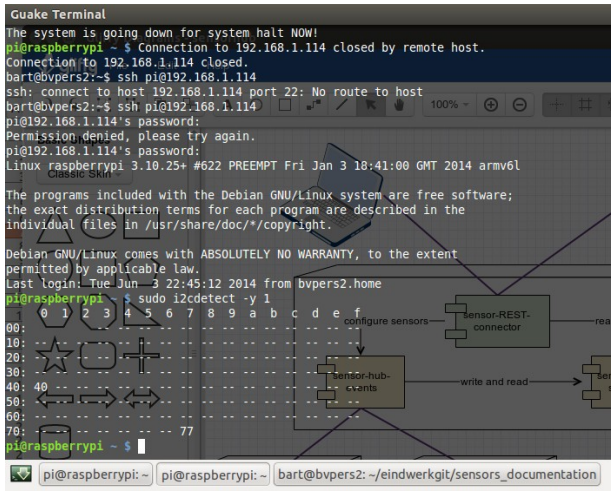
SHT21

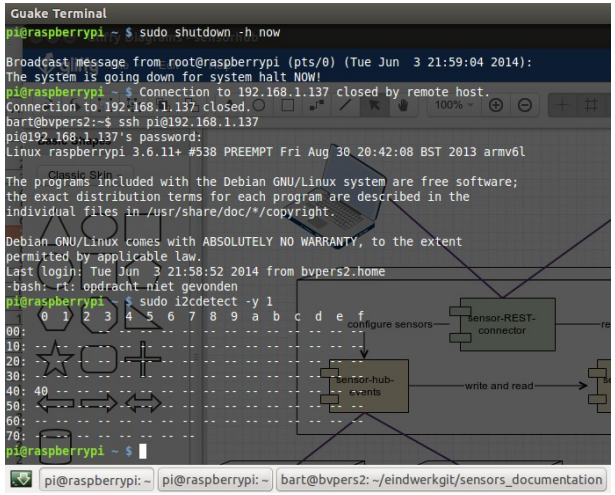
BMP180

Both BMP180 (77) and SDP600 (40) on same bus

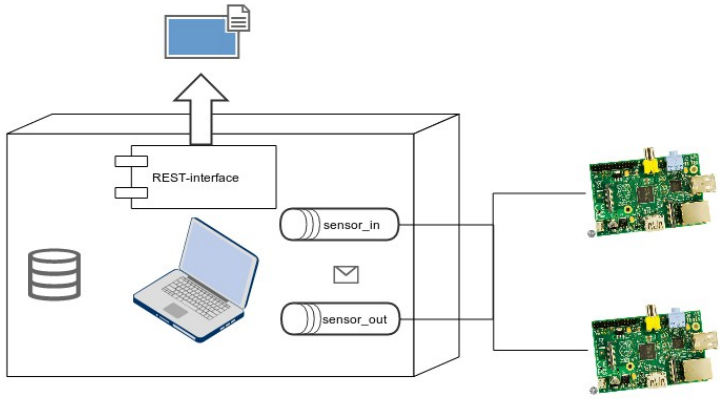


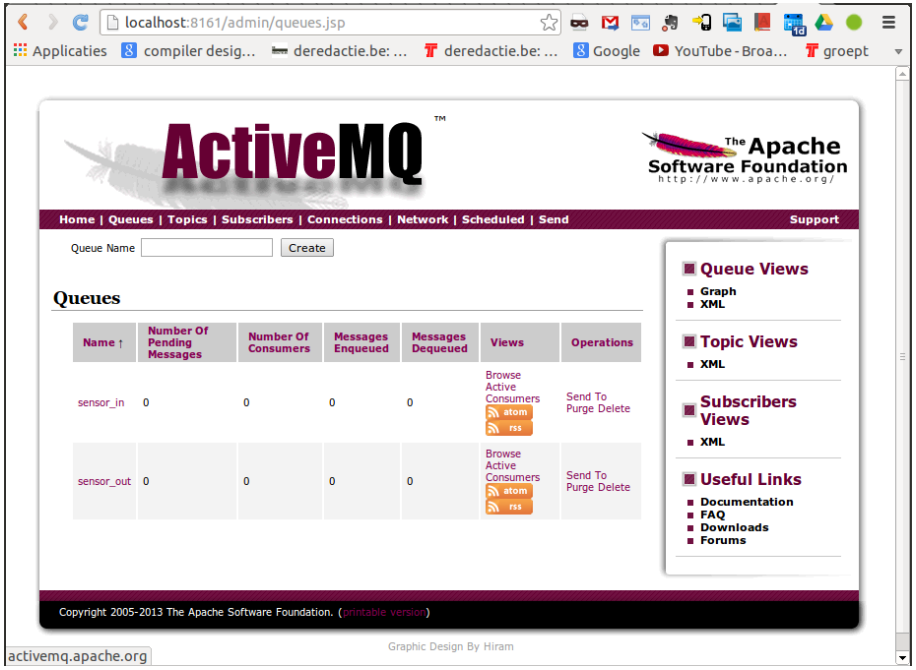






Today

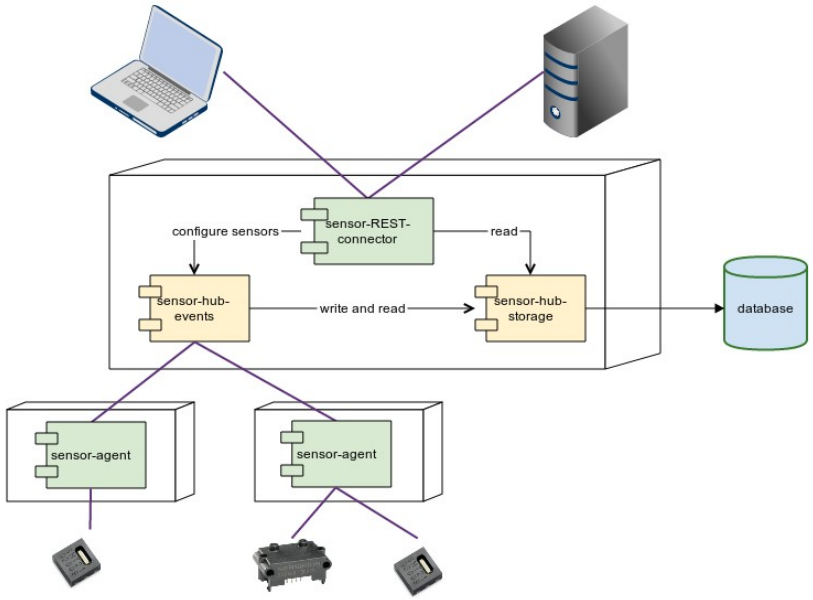

REST-interface
sensor_in
sensor_out


localhost:8161/admin/queues.jsp
Applicaties
compiler desig...
deredactie.be: ...
deredactie.be: ...
Google
YouTube - Broa...
groept
ActiveMQ
The Apache Software Foundation
http://www.apache.org/
Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send
Support
Queue Name
Create
Queues
Name
Number Of Pending Messages
Number Of Consumers
Messages Enqueued
Messages Dequeued
Views
Operations
sensor_in
sensor_out
Browse Active Consumers
Send To
Purge
Delete
Queue Views
Graph
XML
Topic Views
XML
Subscribers Views
XML
Useful Links
Documentation
FAQ
Downloads
Forums
Copyright 2005-2013 The Apache Software Foundation. (printable version)
Graphic Design By Hiram
activemq.apache.org

Recap of top challenges

Multiple sensors

Support for “heterogeneous sensor networks”



Recap of top challenges

- Reliability
  - Data Loss
    - Local and centralized storage
    - Scalability of sensor-hub and centralized storage
  - Support for different devices
    - For different non-functional requirements
  - Monitoring
    - Sensor-Hub can propagate errors
    - Sensor-Hub can non-activity of Sensor-agents

Recap of top challenges

- Correlation and timing
  - Data-model integrates configuration-id's
  - Timing is kept centralized
  - Few seconds difference can be tolerated
  - Sensor-Hub can be programmed to check differences (if required)

Recap of top challenges

Usability and Configurability

System requires no low-level programming to set up sensors

Sensor-configuration and -timing can be configured via hub

Recap of top challenges

- Extensibility
  - Sensor-implementation are abstracted behind generic interface
  - New sensors can be added by implementing an interface

Status

Java-implementations

- Sensor-API
  - SHT21 and SDP600 done
  - STS21, BMP118 ongoing
- Sensor-Agent (activemq)
  - READY TO BE TESTED !!!
- Sensor-HUB (Activemq and Tomcat)
  - READY TO BE TESTED !!!
  - DEPLOYMENT-refinements (configuration and discovery)

Status

C-implementations

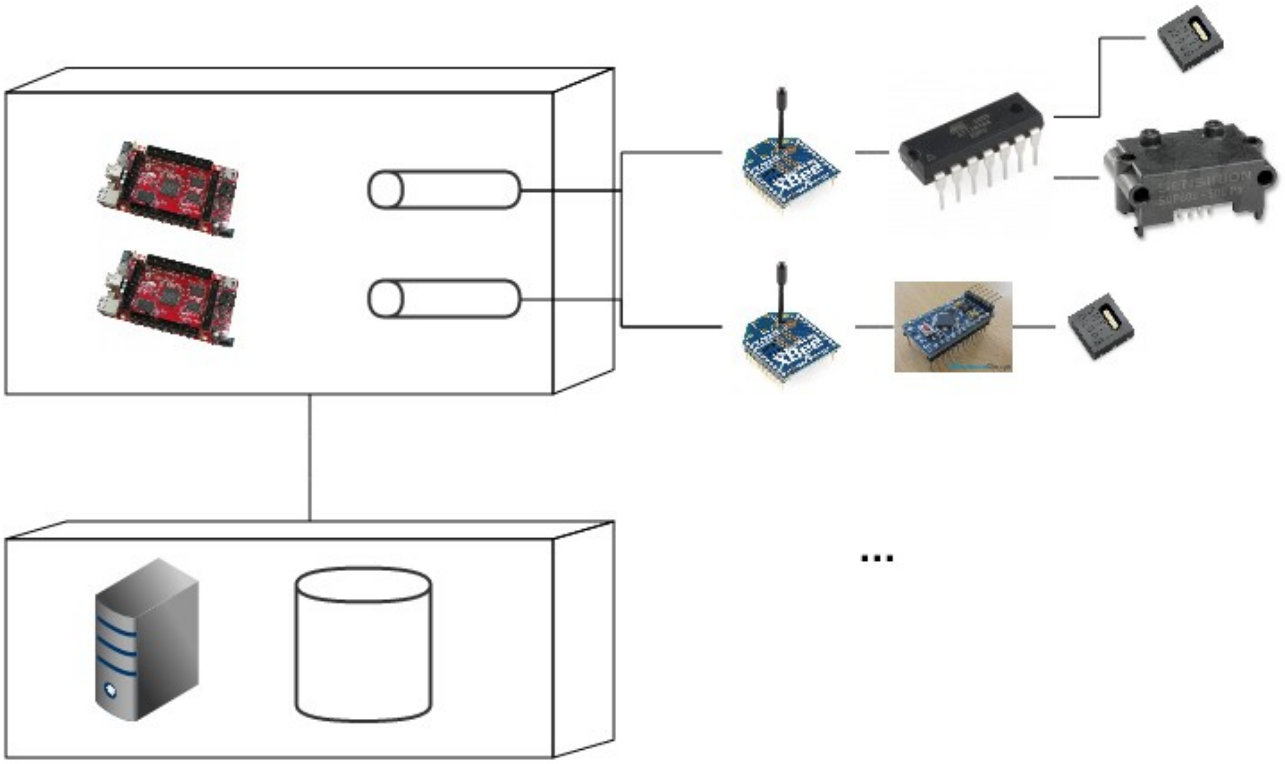
- Sensor-API
  - SHT21 done
  - SDP600, STS21, BMP118 ongoing
  - System-support limited to RPI
- Sensor-Agent
  - Raspberry Pi ongoing
  - AVR ongoing

Status

Documentation

- Deployment-guide
- Mini-training on Raspberry Pi
- Structured documentation on supported sensors ongoing

Future scenarios



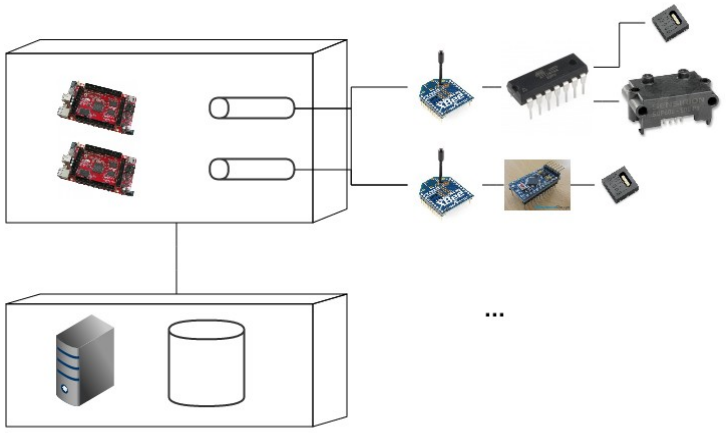
Future scenarios

Adding sensors

- Sensor-abstraction
- Adding new implementation to sensor-agent

Keep attention

- C and Java-implementation should remain in sync



Future scenarios

Elaborating/Finishing the C-agents

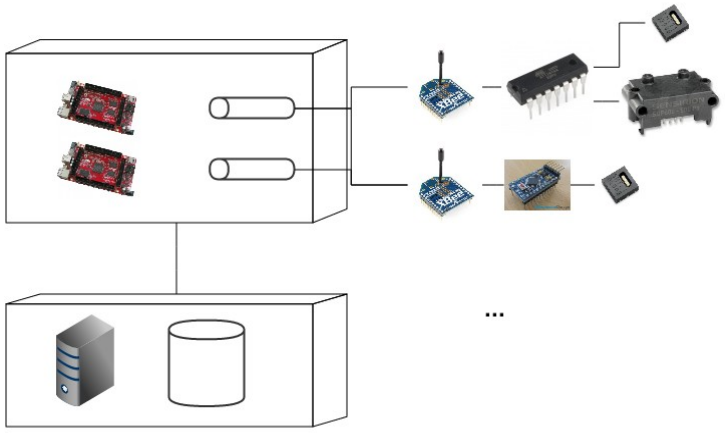
(currently only on Raspberry Pi)

Support for AVR

Allowing low-level devices

Larger scenario's (networks)

Constrained environments



Future scenarios

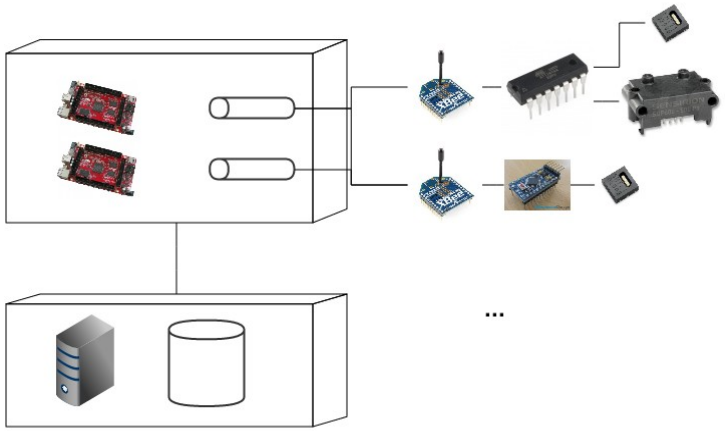
New integration-mechanisms

Data-Link, Network,Transport

- Meshing networks
- Zigbee - IEEE 802.15.4 – Bluetooth light – RF

Middleware

- MQTT (Message Queuing Telemetry Transport)
- MQTT-SN (MQTT for Sensor Networks)



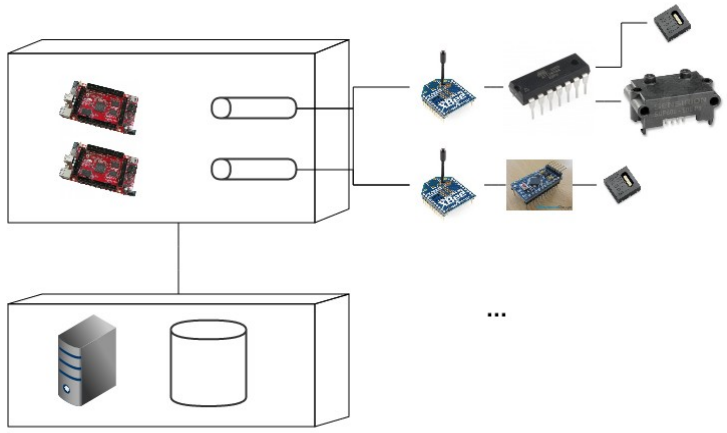
Future scenarios

Scaling

- Adding more sensor-hubs
  - (load balancing ActiveMq)
- Storage and backup on server

Monitoring

- Monitor rest-interface with Nagios



Future scenarios

SensorHub: Raspberry Pi best option?

	Raspberry Pi	BeagleBone Black	pcDuino	Olinuxino Micro	Cubietruck
Memory	4 / 32 GB DDR-1600	4 / 32 GB DDR-1600 ECC	2 / 16 GB DDR-1600	2 / 16 GB DDR-1600	2 / 16 GB DDR-1600
Clockspeed	700 MHz	1 GHz	1 GHz	1 GHz	1 GHz
CPU-core(s)	ARM-11	ARM-Cortex-A8	ARM-Cortex-A8	Dual ARM-Cortex-A7	Dual ARM-Cortex-A7
GPU-core	Videocore IV	SGX530	Mali-400	Dual Mali-400	Dual Mali-400
Hardware Codecs	H264, MPEG-4 AVC (MPEG-2, VC-1 optioneel)	-	MPEG-1/2/4 AVC, JPEG, H.263, H.264, AVS, VC1, WMV7/8, VP-6	MPEG-1/2/4 AVC, JPEG, H.263, H.264, AVS, VC1, WMV7/8, VP-6	MPEG-1/2/4 AVC, JPEG, H.263, H.264, AVS, VC1, WMV7/8, VP-6
RAM	512 MB	512 MB	1 GB	1 GB	2 GB
Flash	-	2 GB	2 GB	4 GB	2 GB





Future scenarios

SensorHub: Raspberry Pi best option?

	Raspberry Pi	BeagleBone Black	pcDuino	Olinuxino Micro	Cubietruck
Lithium-battery	-	-	-	Supported	Supported
Video	HDMI, Composer	Micro-HDMI	HDMI	HDMI, VGA (adapter)	HDMI, VGA
Connections	2 * USB 2.0	USB 2.0	2 * USB 2.0	3 * USB 2.0	3 * USB 2.0, Bluetooth
Audio	Audio-jack	-	-	Audio-jack, Micro	Audio-jack, SPDIF
Network	Fast Ethernet	Fast Ethernet	Fast Ethernet	Fast Ethernet	Gigabit, WIFI
Storage	SD	micro-SD	micro-SD	SD, micro-SD, SATA	SD, micro-SD, SATA
Usage	**Media Center** ++ PC + Home automation +	**Home automation** ++	PC +	Media Center + Router + NAS ++ **SERVER** ++ Home automation + **Mobile** ++	Media Center + Router ++ **NAS** +++ **SERVER** ++ Home automation + Mobile +
Price	35 €	45 €	60 €	65 €	95 €





Future scenarios

SensorHub: Raspberry Pi best option?

Depends on scenario

- **+** Accessible
- **+** Well documented and supported by community
- **-** Stability and industry compliance
- **-** Performance (strength is GPU not CPU)

Recommended for **educational scenario's**

Depending on storage-setup and power required

BeagleBone

Olinuxino

Cubieboard

Future scenarios

SensorHub: Raspberry Pi best option?

- Sensor-hub and -agent
  - Portable
  - Java is supported on most (all) high-level devices
  - C is supported on all low-level device**s**
  - System-dependencies isolated
- Recommended for **educational scenario's**

Depending on storage-setup and power required

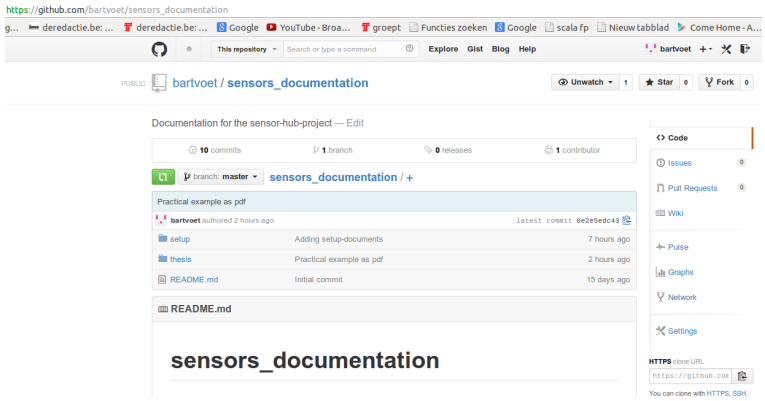
BeagleBone

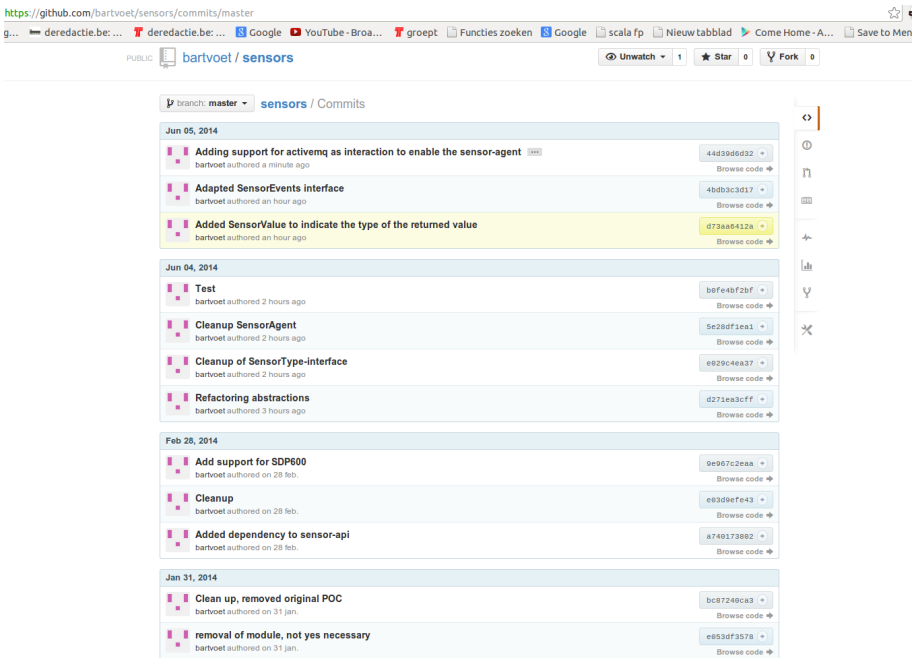
Olinuxino

Cubieboard

More information

Code and documentation on https://github.com/bartvoet/





Thank you for the attention!
Ready for Questions.