HomePort 1.0

"The software that everybody needs, but no one will pay to develop."

Concepts and Prototype: Jeppe Brøndsted and Rune Torbensen

Implementation 1.0: Jesper B. Rosenkilde, Thibaut Le Guilly, [Régis Louge], and Petur Olsen

Management: Anders P. Ravn and Arne Skou

BSD license

- Get it now!
  - https://github.com/home-port/HomePort
- Try it out:
  - https://github.com/home-port/HomePort
- Not working or need feature?
  - Register an issue:
    - https://github.com/home-port/HomePort/issues
  - Contact us:
    - homeport-support@cs.aau.dk
- Want to contribute?
  - Send a pull request on GitHub

Basic Idea

"Internet of Things ?!"

HomePort

Basic Idea

"Internet of Things ?!"

HomePort

History of Homeport

- Started as a PhD-project in 2008 with Jeppe Rørbæk Brønsted and Rune Torbensen
- Adopted by the research project Dit Hus in April 2009 with Develco, Seluxit, Servodan, CISS and Alexandra Instituttet (funding ends July 2012).
- Marts 2011 Jesper (me) hired
- September 2011 Regis and Thibaut hired
- Encourage (funding)
- February 2012, Petur Joins the team

Servicecentric

Event: On motion

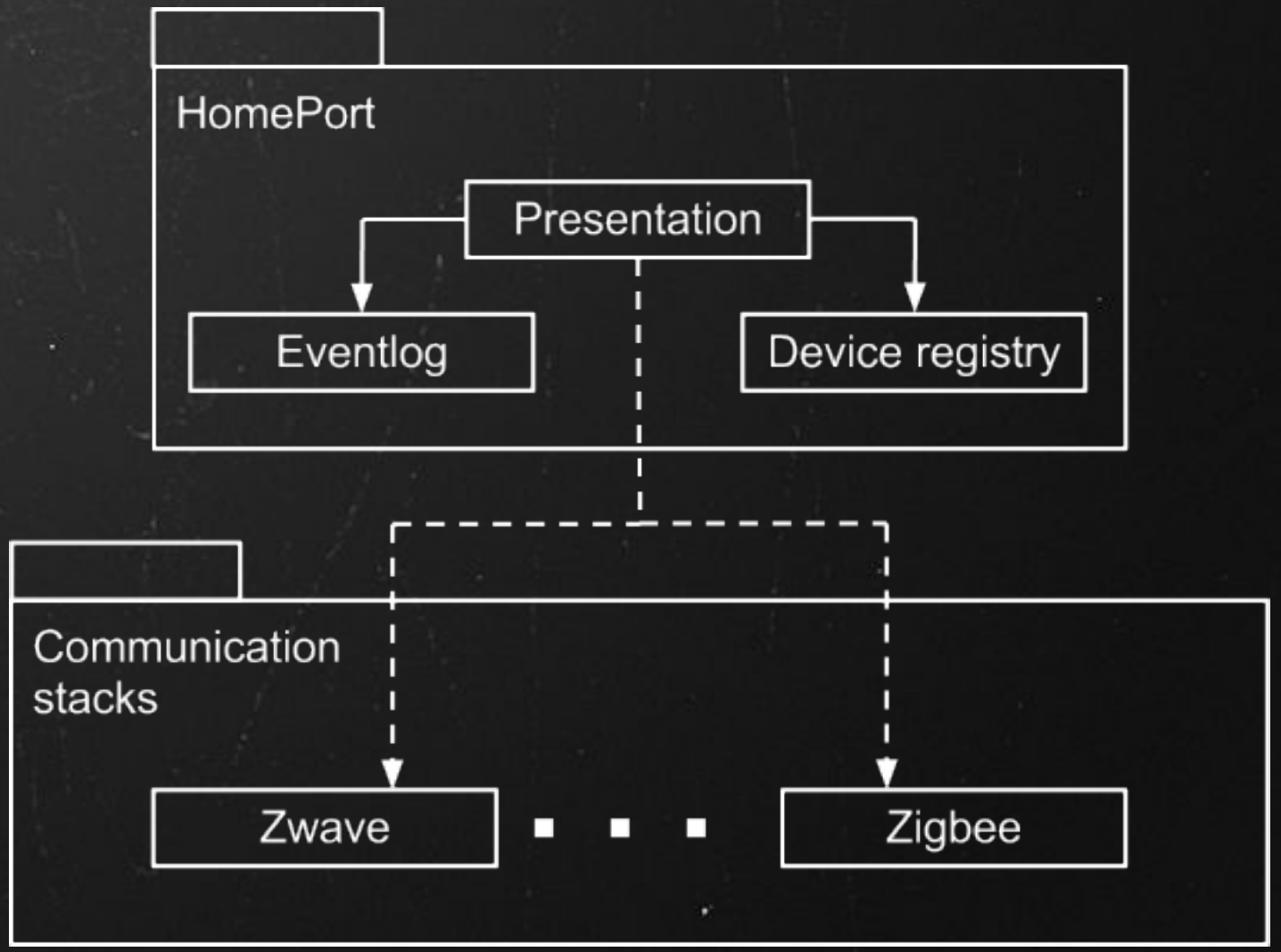
Event: Door/window open

Output: Temperature
Humidity

Input: Turn power on/off
Output: Power state
Event: On power state change

Input: Set temperature

Architecture



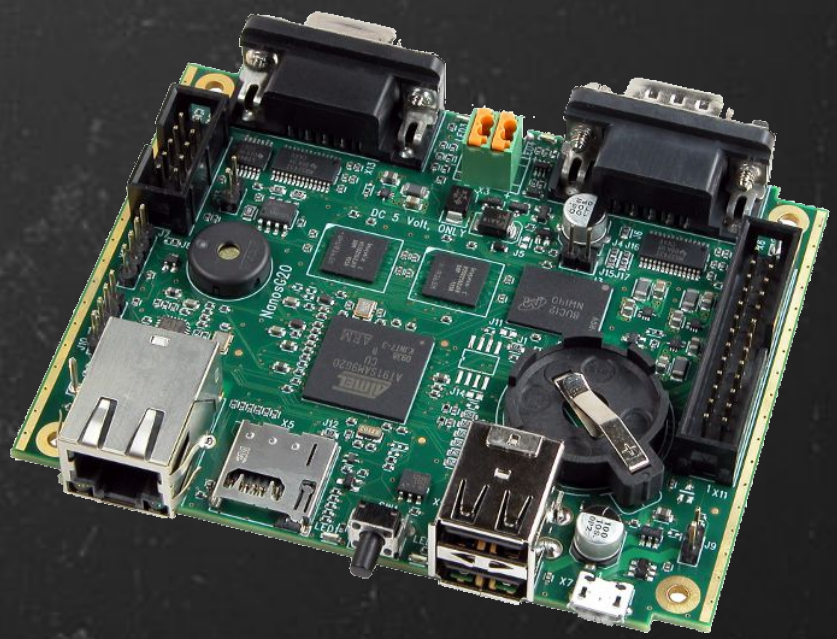
Technology

- Embedded Linux
- HTTP (REST)
  - HTTP verbs are used extensively
  - EventSource used to push events
  - RESTfull approach ensures idempotence
  - Easy access from applications
- ZeroConf and mDNS (Avahi)
  - Auto configure network
  - Distribute device directory
- SSL
  - Standard way to add security
- XML

Embedded Linux

- *Getting very easy to work with*
- *Emdebian*
  - Cross toolchains
    - cross compiling (still a bitch ;)
  - Multistrap
- Nice hardware

NanosG20



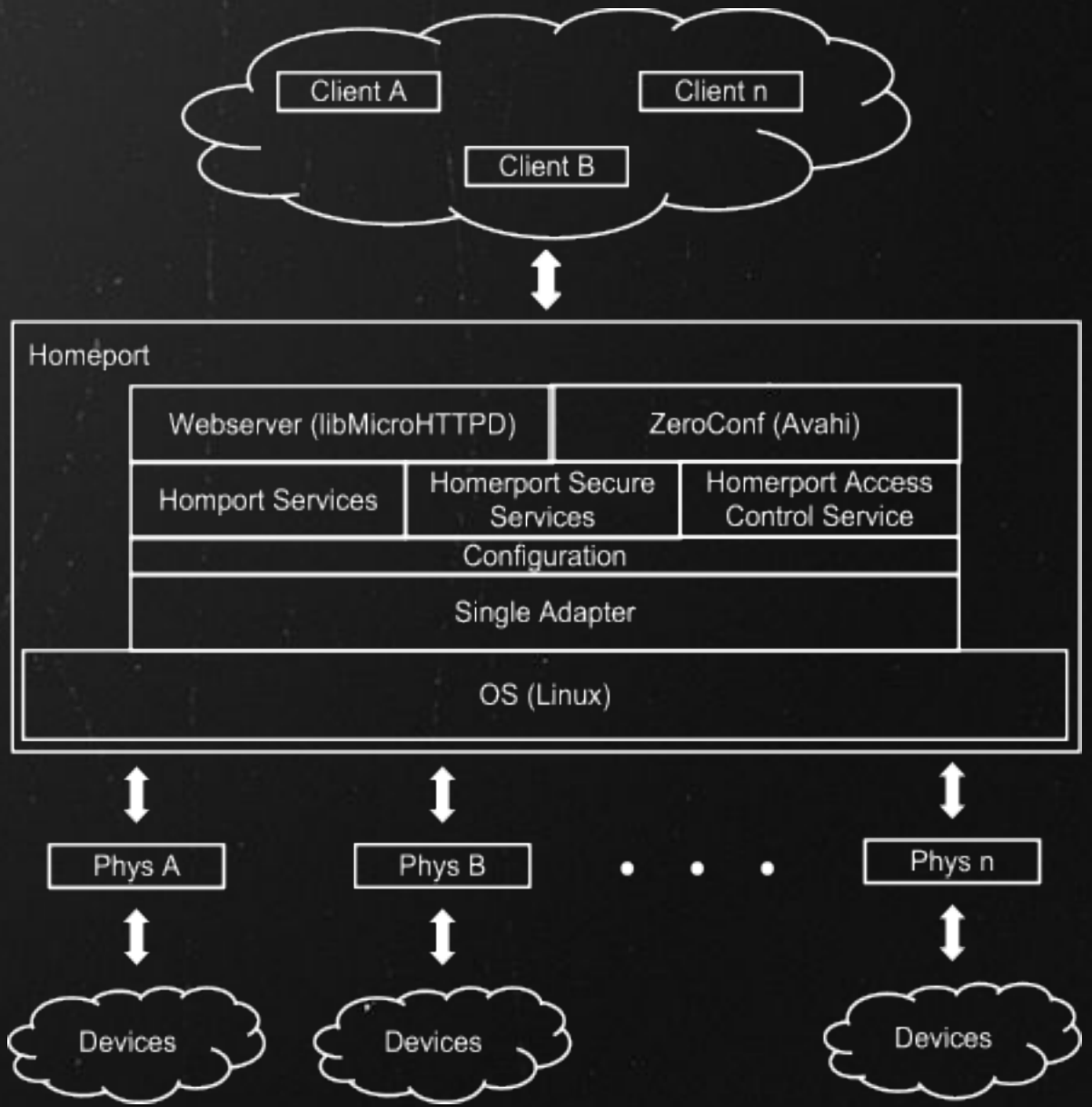
- Linux Computer with an Atmel® ARM AT91SAM9G20 microcontroller.
- Energy consumption < 1W at full system load.
- 400 MHz, 256 MB Flash and 64 MB SDRAM.
- Ethernet, MicroSD, 2 x USB host, RS232 and RS485.
- Cost-efficient and long-term available.
- OS Debian Linux.

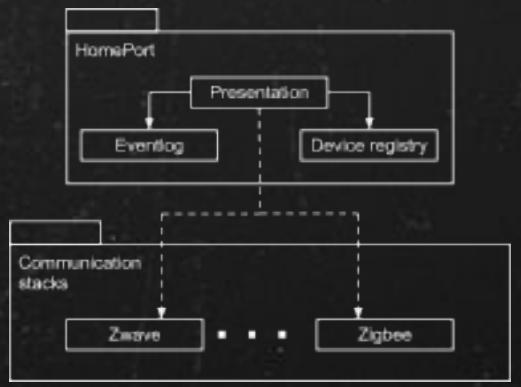
- Pris: 99€.

Service and device description

```
<?xml version="1.0" encoding="UTF-8"?>
<device desc="Lamp" id="0" location="Living room" port="0"
type="lamp">
  <service desc="On/off service" id="0" type="onoff" unit=""
value_url="http://127.0.0.1/devices/0/lamp/0/">
  </service>
</device>
```

Implementation





Representational state transfer (REST)

- Layered system
  - A client realize whether it is connected directly to the end server, or to an intermediary.
    - Can be used to enforce security policies, help scalabilty, etc.
- Code on demand (optional)
  - Transfer code to client
- Uniform interface
  - Identification of resources (URIs and metadata)
  - Manipulation of resources through these representations
  - Self-descriptive messages

Representational state transfer

Resource	GET	PUT	POST	DELETE
/resources/	**List** the URIs and perhaps other details of the collection's members.	**Replace** the entire collection with another collection.	**Create** a new entry in the collection. The new entry's URL is assigned automatically and is usually returned by the operation.	**Delete** the entire collection.
/resources/item17/	**Retrieve** a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	**Replace** the addressed member of the collection, or if it doesn't exist,**create** it.	Treat the addressed member as a collection in its own right and **create** a new entry in it.	**Delete** the addressed member of the collection.

Application interface

http://127.0.0.1/devices/

list of devices

Application interface

http://127.0.0.1/devices/0

list of device services

Application interface

http://127.0.0.1/devices/0/lamp

description of service type

Application interface

http://127.0.0.1/devices/0/lamp/0/

return state of service identified by URL

Adapter interface

```
int HPD_start( unsigned int option, char *hostname, ... );

int HPD_stop();

int HPD_register_service( Service *service_to_register );

int HPD_unregister_service( Service *service_to_unregister );

int HPD_register_device_services( Device *device_to_register );

int HPD_unregister_device_services( Device *device_to_unregister );

int HPD_send_event_of_value_change ( Service *service_changed,
        char *updated_value );
```

Past projects

- iAbis
  - Rest home
- Conlan
  - access control system
- Danfos
  - Thermostat
- Zensehome
  - Power line communication
- Seluxit
  - Zwave

Current projects

- Develco
  - Zigbee and metering
- Zensehome and Nabto
  - Internet and STUN
- Eglu
  - Demo house

Future projects

- Encourage
  - energy optimisation
- Danish defence
  - energy optimisation

Selling Open Source

- BSD vs. GPL
- Binary blobs
- Who has the responsibility for maintenance
- Standardisation
- Open source group vs. real company

HomePort 2.0

- Multiple sandboxed adapters
- Better event interface
- Programmable rules
  - Triggers
  - Simulink to HomePort
- Binary HomePort
  - Tiny version of HomePort
  - One device, eight services
  - Target pic18
- Switch from XML to JSON
- Homogeneous network

BSD license

- Get it now!
  - https://github.com/home-port/HomePort
- Try it out:
  - https://github.com/home-port/HomePort
- Not working or need feature?
  - Register an issue:
    - https://github.com/home-port/HomePort/issues
  - Contact us:
    - homeport-support@cs.aau.dk
- Want to contribute?
  - Send a pull request on GitHub

Questions