

'Dit hus' introduction

/ Jeppe Brønsted,

Arne Skou

and

Per Printz Madsen,

Rune Torbensen

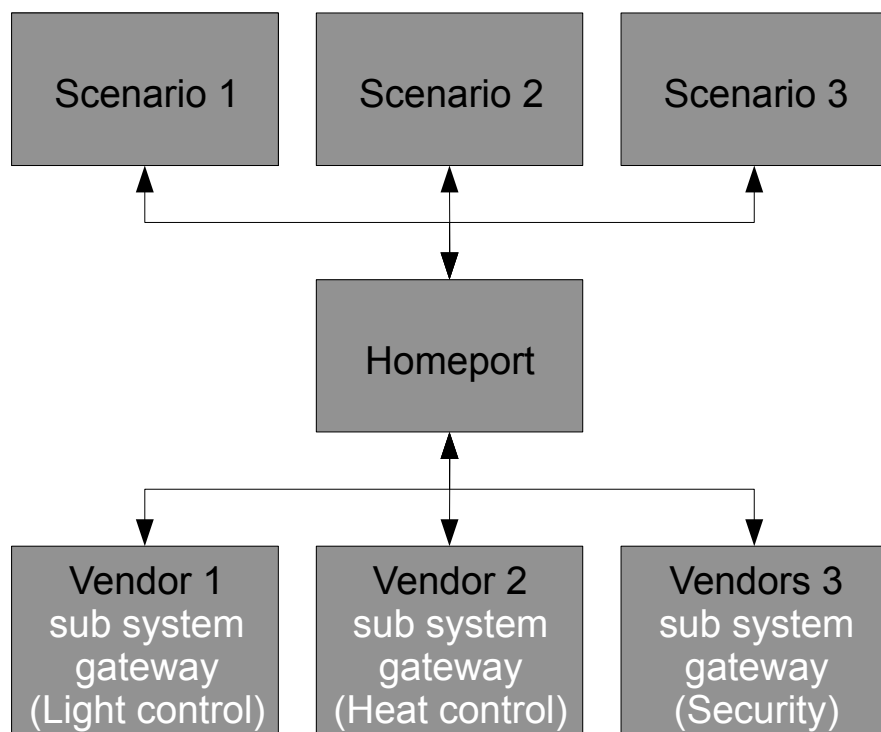
Project facts

- Period: April 2009 (August 2008) – October 2011
- Budget: 10.6 mill.kr.
- Funding: 8 mill. kr. (2.6 mill. kr. for companies)
- Knowledge institution partners: AAU (CISS/CMI), AU/Alexandra Institute, CSI, IHA, HIH (AU)
- Initial private partners: Develco, Seluxit, Servodan

Project goals

- Implementation of prototype for co-existence of different home control devices
- Validation of a number of scenarios and interoperabilities
- Collaboration with industrial partners
- Investigate the business potential of a homeport
- Knowledge dissemination of project results

Home port idea



- Home port
 - A SOA architecture
 - A protocol
 - A rule definition language
 - Configuration conventions
- Gateways
 - Connects vendor sub systems
 - Subsystems provide resources for the home port
- Scenarios
 - Abstract away from specific technologies
 - Interpreted/executed by home port
 - Exploit home port resources

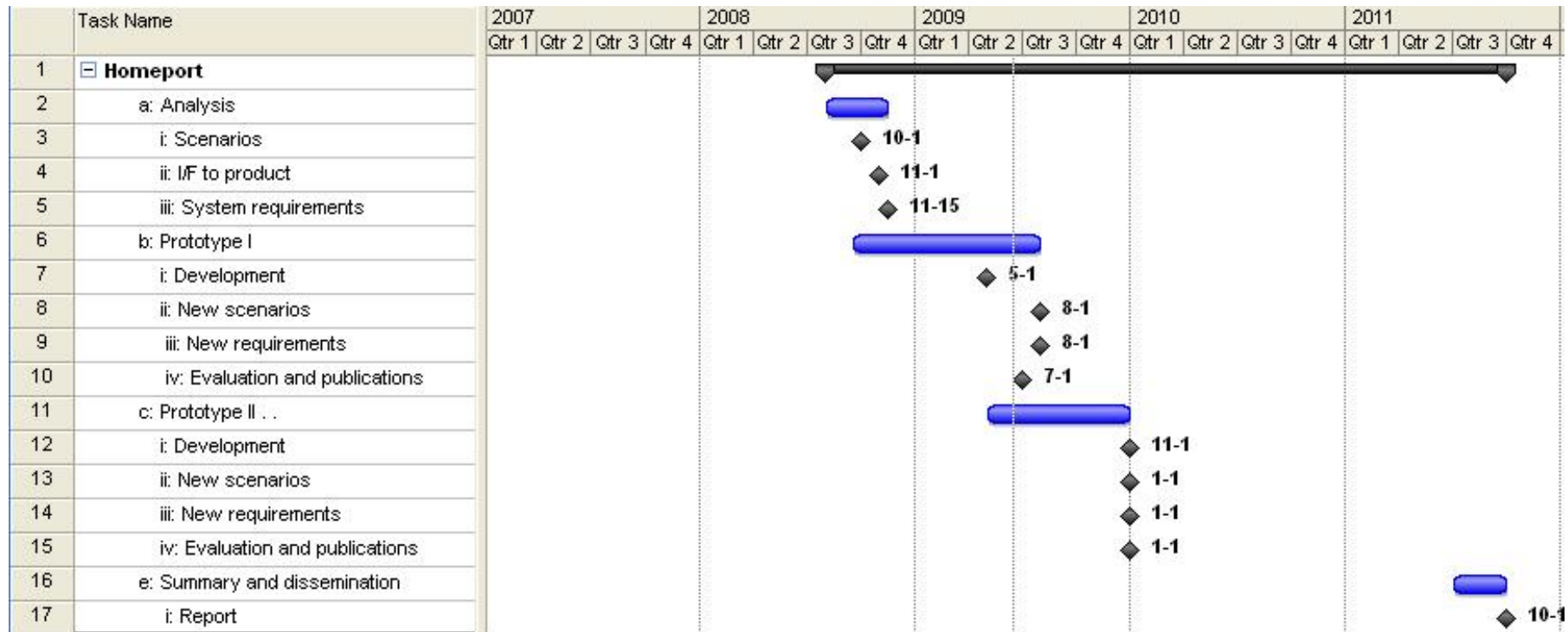
Project scope

- Assumption: Gateways spend minimal energy resources.
 - Focus is on interoperability
 - Gateway energy consumption is initially ignored
- We extend the functionality through scenario implementations
- Approach
 - The industrial partners define the scenarios
 - The industrial partners provide products/hardware/functions
 - The university partners contribute with methods/protocols and prototypes of homeport interoperability

Project phases

- Analysis:
 - Selection of initial scenarios and system requirements
 - Definition of interface towards subsystems
- Prototype I
 - Development
 - New scenarios and system requirements
 - Evaluation and publications
- Prototype N . . .
 - Development
 - New scenarios and system requirements
 - Evaluation and publications
- Summary and dissemination

Plans and milestones



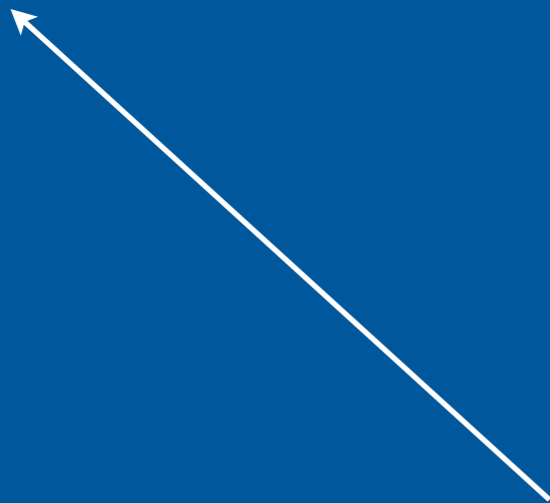
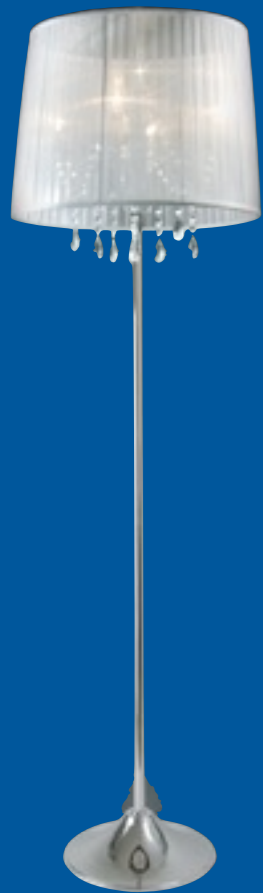
Collaboration examples

- Develco:
 - I/F between homeport and ZigBee standard controlling selected devices
 - Runs for autumn 2009
- Seluxit:
 - I/F between homeport and Z-Wave standard
 - Rule language
 - Transfer protocols
 - Configuration (internal/external)
 - Runs for the next year
 - Possibly I/F to 'minbolig'.
- Servodan:
 - Communication with Servodan controller
 - Runs for autumn 2009



OVERORDNET ARKITEKTUR

I Dit Hus





AGENDA

- › Motivation
- › Home automation
- › Arkitektur
 - › Krav
 - › Layer View
 - › Deployment View
 - › Brug af webteknologi
 - › Composite service - example
 - › Web UI demo
- › Åbne spørgsmål



MOTIVATION: BÆREDYGTIGT ENERGIFORBRUG I HJEMMET



- › Ide:
 - Sameksistens af kommunikationsteknologier til styring og overvågning af energi mv.
- › Optimere energiforbrug gennem integration af home automation teknologier
 - › Bruge mindre - f.eks. sluk-alt knap
 - › Bruge på det rigtige tidspunkt - f.eks. vaske om natten
 - › Opbevare energi - f.eks. tænde for varmepumpe om natten
- › Optimeringspotentialer
 - › Suboptimalt energiforbrug - f.eks. lad lyset stå tændt
 - › Afhjælpes af automatisering - f.eks ved at forbinde alarmsystem med lysstyring
 - › Ødsel adfærd - f.eks. langt brusebad
 - › Afhjælpes af ændret adfærd - f.eks. ved at visualisere forbrug
- › **Forudsætning:** integration af hjemmets apparater

"If you cannot measure it,
you cannot improve it."

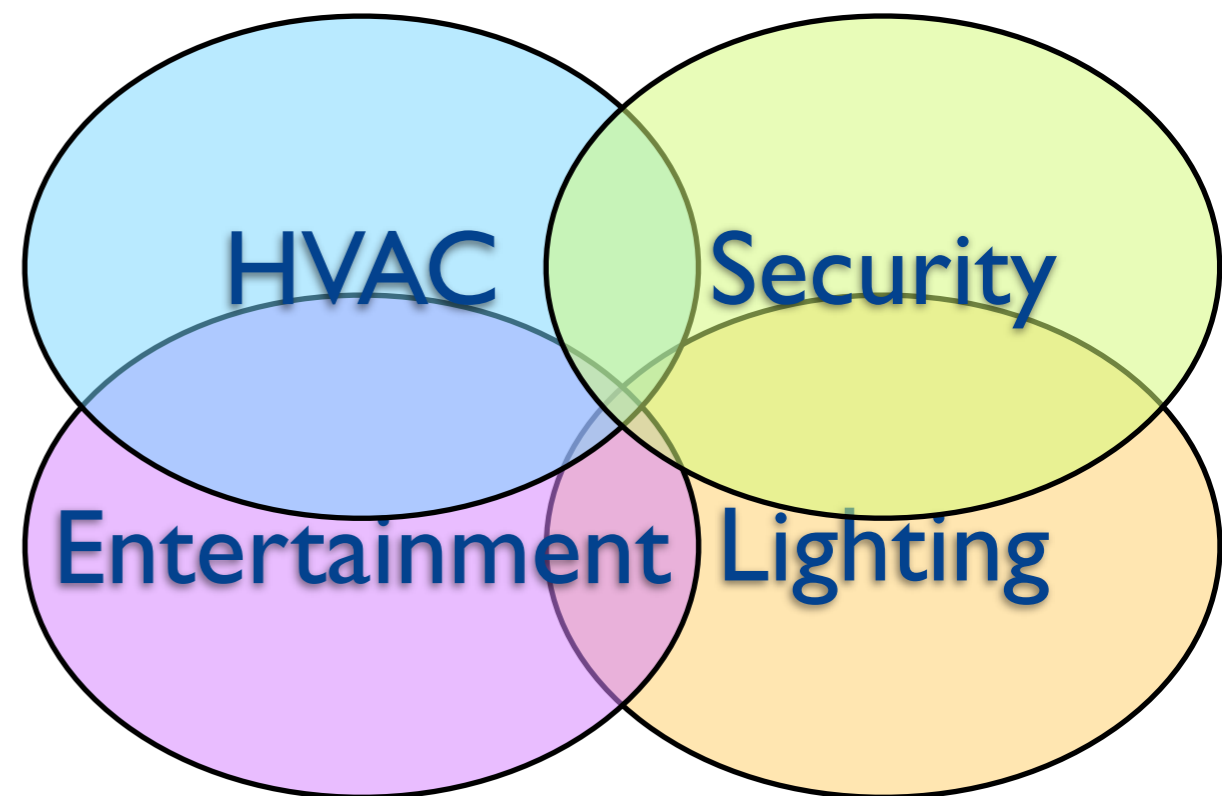
- Lord Kelvin





HOME AUTOMATION

- › Indeholder traditionelt flere subdomæner
 - › Med forskellige krav
- › Indenfor hvert domæne findes forskellige standarder/producenter
- › Det kan ikke antages at alt kontrolleres af den samme entitet
 - › (ville ellers gøre det let)
- › Tilgange
 - › Standardisering (som i telekom)
 - › Sameksistens
 - › (kommunikation via homeport)



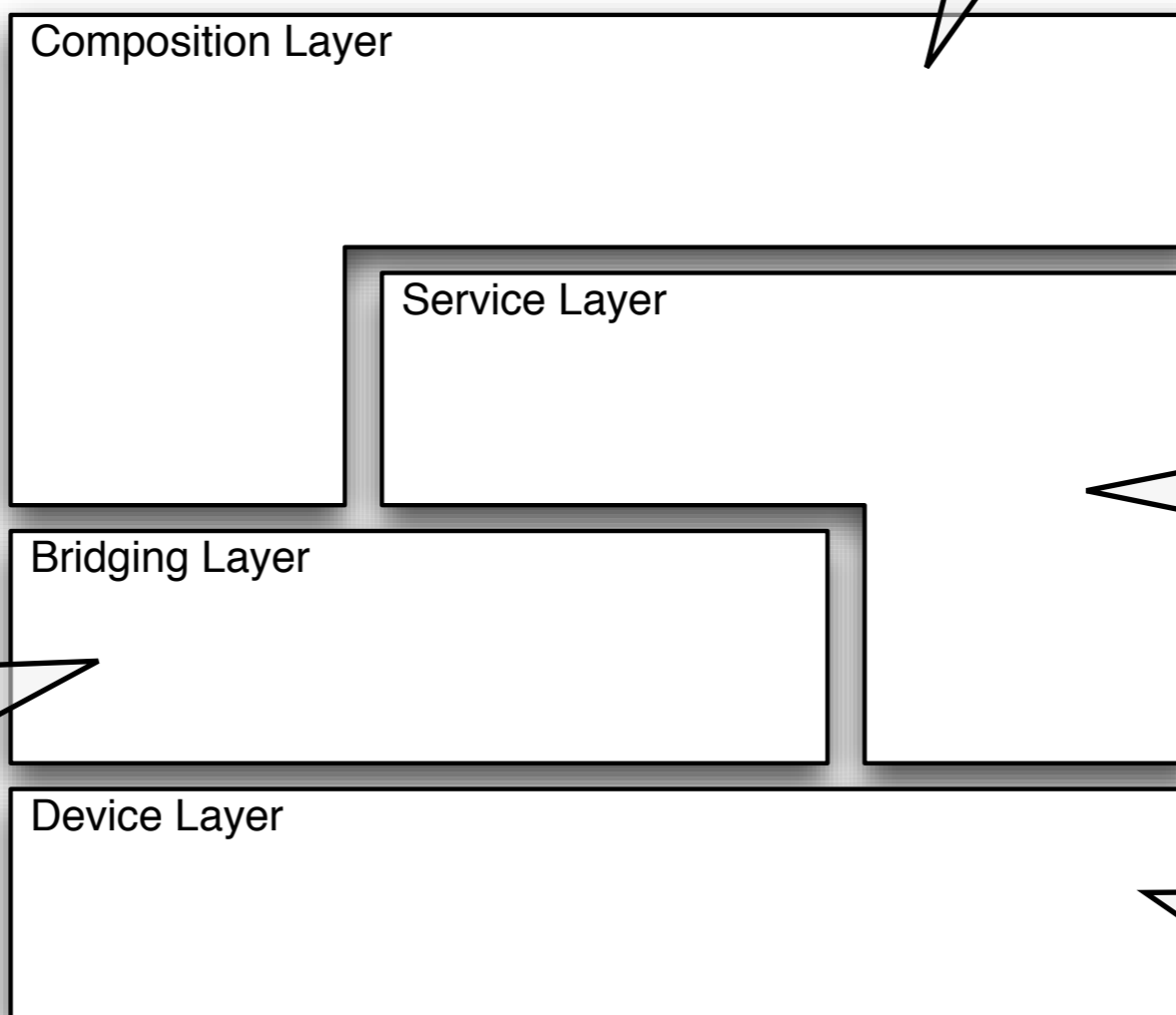


KRAV TIL ARKITEKTUR

- › Funktionalitet
 - › Apparater skal kunne kommunikere på en fornuftig måde
- › Forretningsmæssige krav
 - › Producenter skal selv bestemme hvad de vil åbne op for
 - › Eksisterende end-devices skal kunne fungere uden at skulle ændres
- › Krav til kvaliteter (arkitekturegenskaber)
 - › Modificerbarhed
 - › Det skal være let at tilføje nye subsystemer og end-devices efter deployment - uden at påvirke kørende system
 - › Brugbarhed
 - › Systemet skal være anvendeligt af forskellige kategorier af brugere
 - › Skalerbarhed
 - › 100er af samtidige enheder
 - › Skal kunne implementeres på resourcebegrænsede systemer



Service composites og
controllere sammensætter
funktionalitet.



Services præsenterer
funktionalitet gennem
REST/HTTP interface

Bridgelag videresender
kommandoer til og fra
enheder

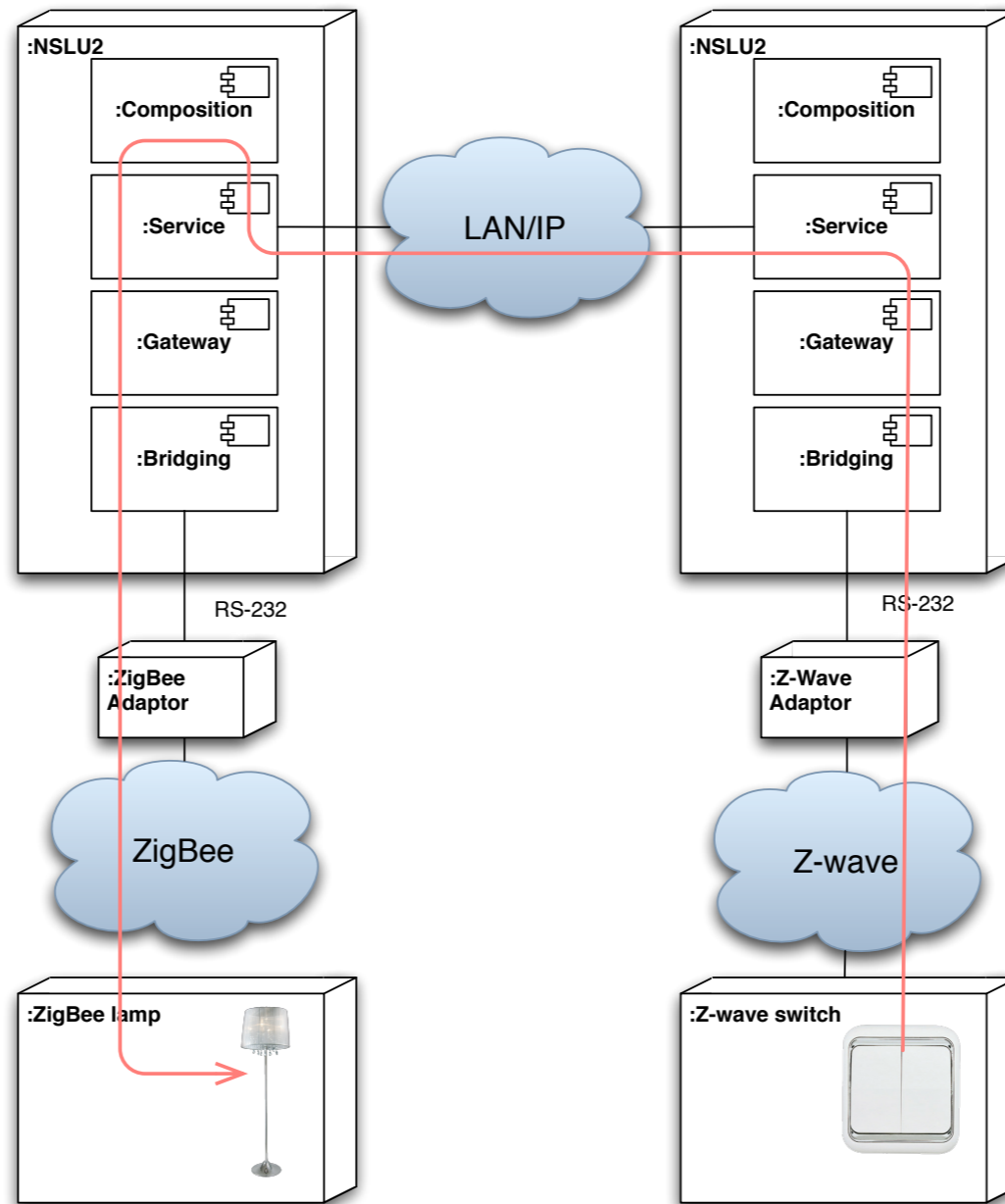
End-devices forbundet til
subsystemer.
E.g. Z-wave enheder

LAYER VIEW



BRUG AF WEBTEKNOLOGI

- › **Motivation**
- › Lav barrier-of-entry:
 - › De fleste plaforme understøtter HTTP, URLs og HTML/XML/json
- › Device, programmeringssprog, og service uafhængighed
- › Fast interface - data ændres.
 - › HTTP metoder med semantik: GET, PUT, POST, DELETE
 - › Klienter og servere behøver ikke nødvendigvis forstå alt for at kunne fungere fornuftigt
- › Udtrykskraft tilstrækkelig
 - › (demonstreret på www)
- › **Implementation**
- › Url hierarchy
 - › services
 - › composites
 - › subscriptions - push and pull (streaming)
 - › gui
- › Accept-header determines content-type of reply
 - › XML, plain, HTML, json
- › Browser can be used for inspection
- › XML schemas for basic types

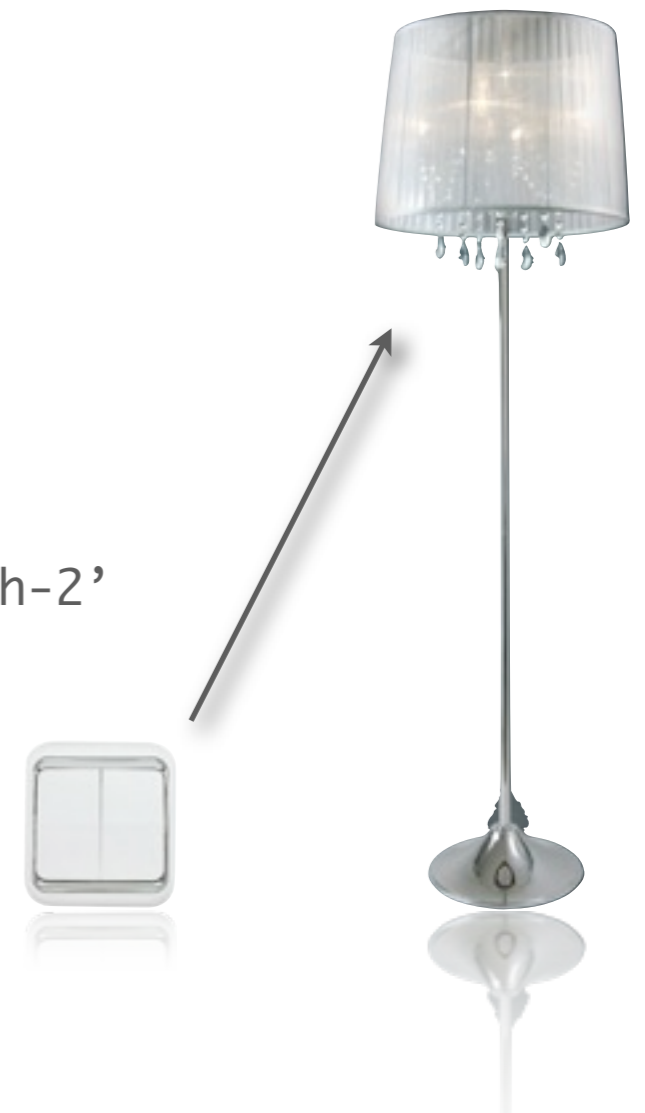


DEPLOYMENT VIEW



COMPOSITE SERVICE - EXAMPLE

- › Simple switch-lamp composite
- › Composite subscribes to switch events
- › Use HTTP-streaming
 - › GET subscriptions/subscribe/services/switch
- › When event is received the lamp state is toggled
 - › PUT services/lamp 'on'
- › Composite can be reconfigured
 - › PUT services/switch-lamp-composite/lamp 'services/lamp-2'
 - › PUT services/switch-lamp-composite/switch 'services/switch-2'





HVORDAN SER SYSTEMET UD FOR FORSKELLIGE KATEGORIER AF BRUGERE?

- › Slutbruger (brug)
 - › Oplever systemet gennem hjemmets apparater (lamper, kontakter, vægpaneler osv.)
- › Slutbruger (konfigurering/installation)
 - › Tilgår systemet gennem brugergrænseflade (PC, web, mobil, el. lign.)
- › End-device udvikler
 - › Behøver ikke vide at homeport eksisterer
- › Controller/composition udvikler
 - › Tilgår services via HTTP/REST. (eksempel næste slide) Bruger service discovery mekanismer
- › Gatewayudvikler
 - › Er ansvarlig for at tilbyde subsystem device funktionalitet gennem HTTP/REST interface (skal være subsystemsekspert)
 - › Baseret på standard biblioteker



WEB UI DEMO

> <http://www.cs.au.dk/~jrb/dithus/mockup/gui2.html>



ÅBNE SPØRGSMÅL

- › Skal devices beriges med ekstra intelligens?
 - › Nogle apparater understøtter ikke tovejs kommunikation
 - › Lampe der kun understøtter tænd/sluk og ikke status requests. sluk-alt?
 - › Kontakt der kun sender kommandoer - ingen status
 - › Logik til at sikre korrekt brug
 - › Undgå slid på mekanik / lyspærer
 - › Dekorering med information om energiforbrug (30 watt pære)
 - › Caching af værdier - både læsning og skrivning
 - › Service discovery
 - › Få fat i alle lamper
- › Hvor i infrastrukture skal hvilken logik ligge?
 - › Bridge, gateway, controller/composite



TAK FOR OPMÆRKSOMHEDEN

Spørgsmål?

