



# SMART HOMES FOR ALL

An embedded middleware platform  
for pervasive and immersive  
environments for-all

**Leonardo Querzoni**

Sapienza Università di Roma



SAPIENZA  
UNIVERSITÀ DI ROMA

## Outline

Overview of the project:

- Fact sheet
- Goals
- Possible scenarios
- Technical challenges
- Architecture
- Current status

Open research problems:

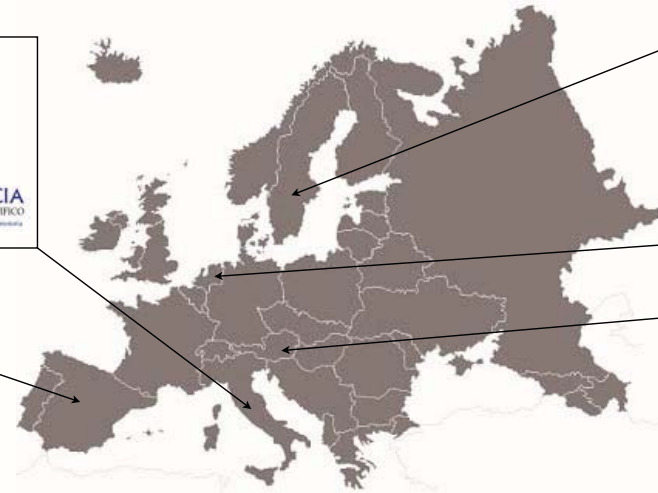
- Failure detection
- Concurrency management



# Fact sheet



- STREP - FP7 224332
- 1/9/2008 - 31/8/2011
- Coordination: Prof. Roberto Baldoni
- <http://www.sm4all-project.eu/>
- Partners:



# SM4All goals

Home automation today:

- many incompatible solutions.
- limited choices
- complex setup
- predefined functionalities



# SM4All goals



A vision on the future:

- widespread interoperability
- a lot of devices
- complex services
- simple installation and automatic configuration
- cheap systems

# SM4All goals



The project objective:

*provide a new software infrastructure for complex service provisioning in a home automation scenario where the user is at the center of the stage*

# Possible usage scenarios



- Bob comes back at home and decides to watch a movie.
- The system tries to fulfill this goal:
  - dims the lights in the living room;
  - turns on the TV set;
  - shows to Bob's the list of titles available in the media server;
  - allows incoming calls only from Bob's girlfriend number.



# Possible usage scenarios



- Chuck is a disabled person and Alice is his nurse.
- Chuck wants to have a relaxing bath.
- He instructs the system through his BCI unit.
- The system prepares the bath following his preferences (room temperature, water temperature, etc.).
- When the environment is ready the system alerts Alice (currently in the kitchen) that Chuck must be moved from the bedroom to the bathroom.

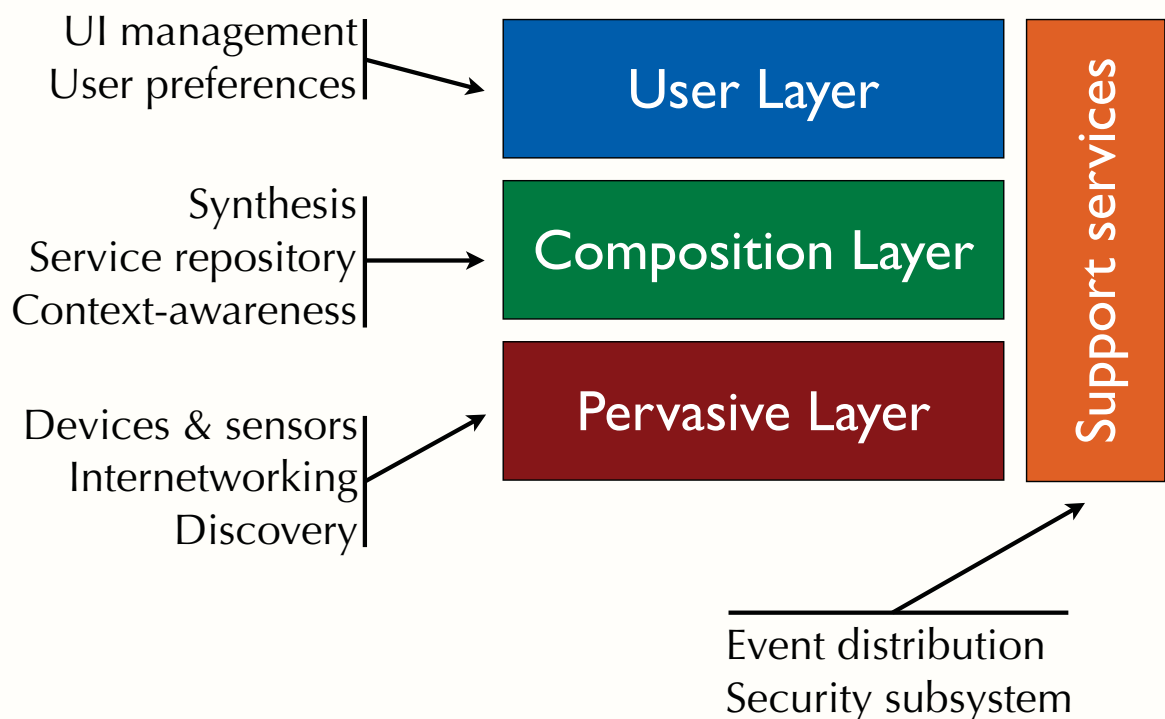


# Technological challenges

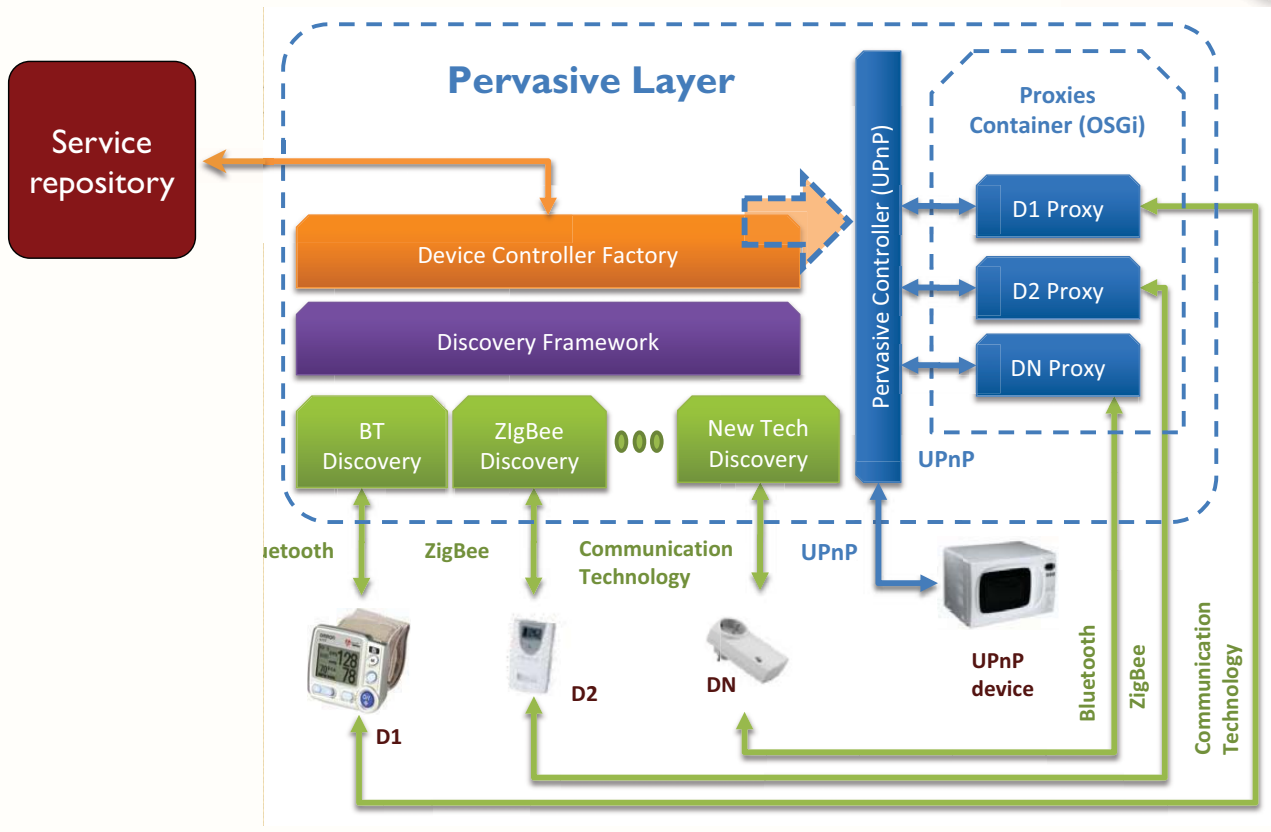


- Devices using different technologies
  - interoperability through a SOA approach
- Support many devices and multiple users
  - scalability, reliability and concurrency issues
- Offer complex functionalities
  - semantic-based service composition and orchestration
- Allow interaction with users with different abilities
  - multiple UI: web, touchscreen, BCI, etc.

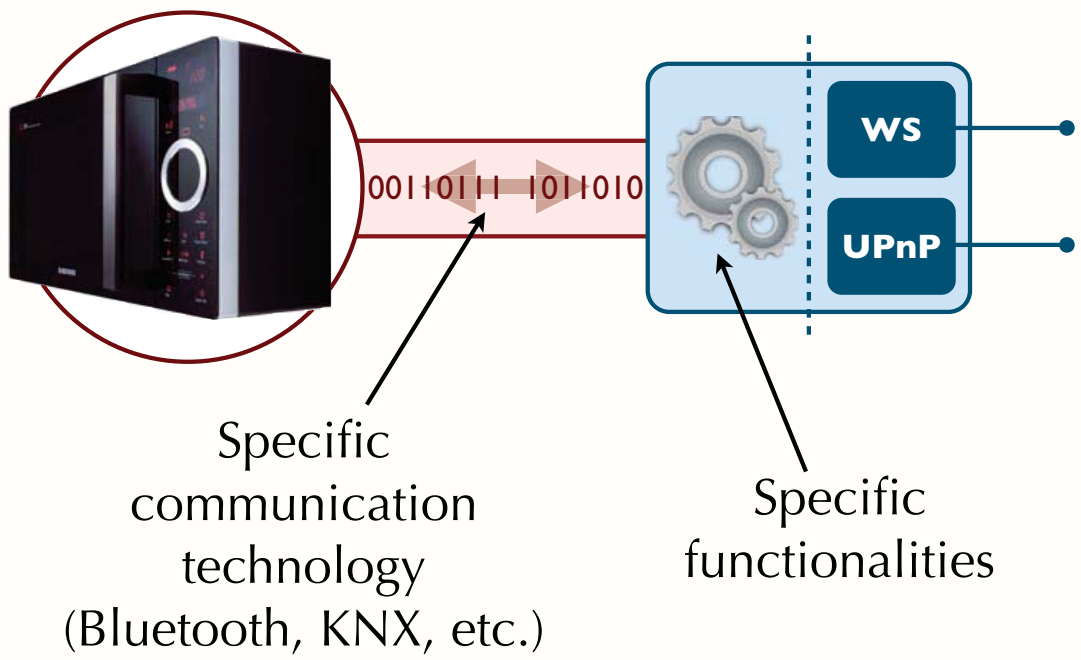
# Architecture



# Pervasive layer

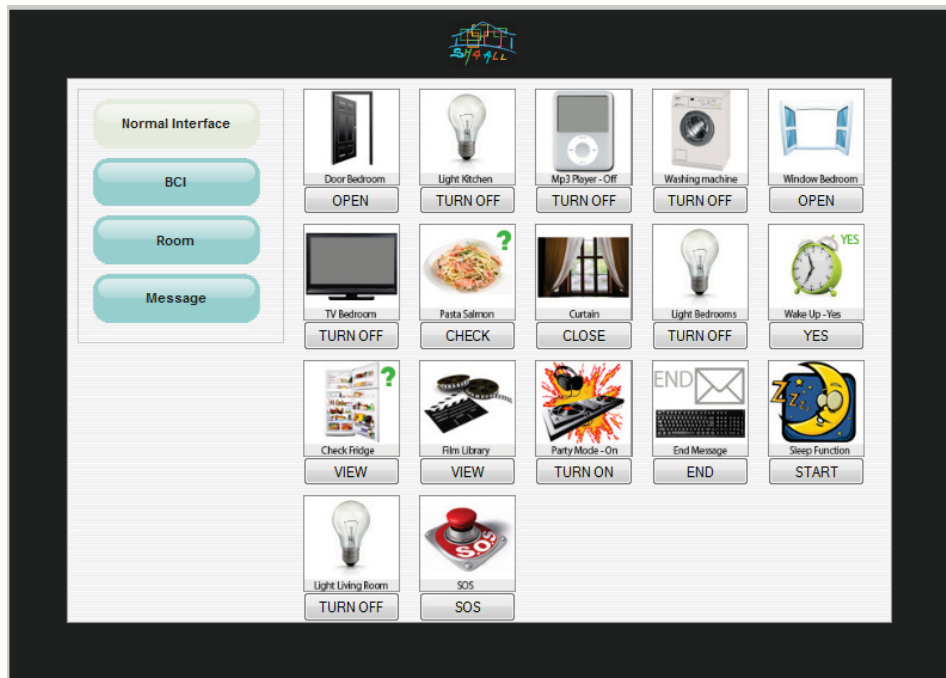


# Pervasive layer



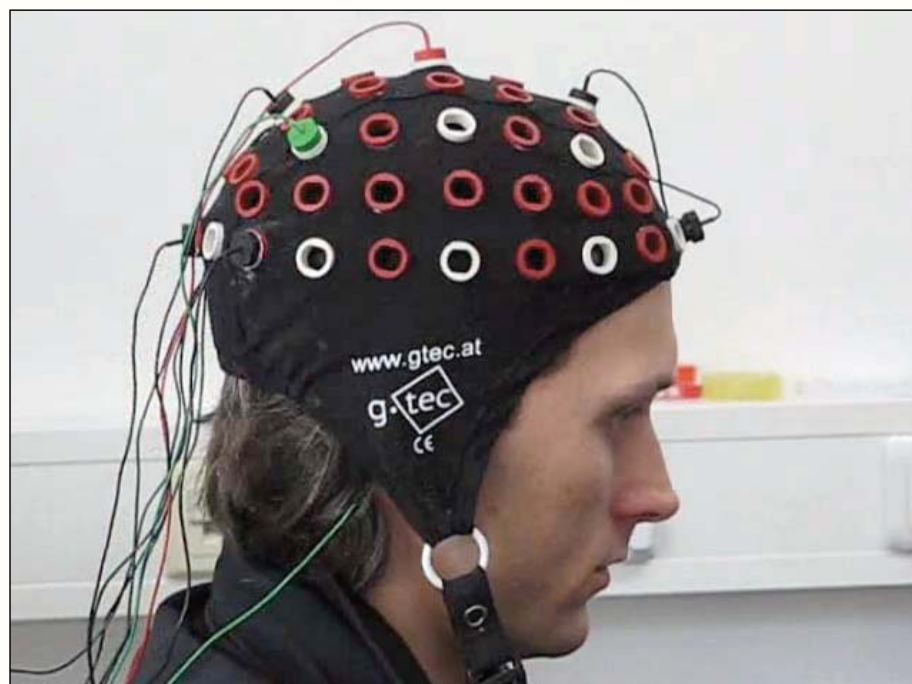
# User layer

- Standard web interface

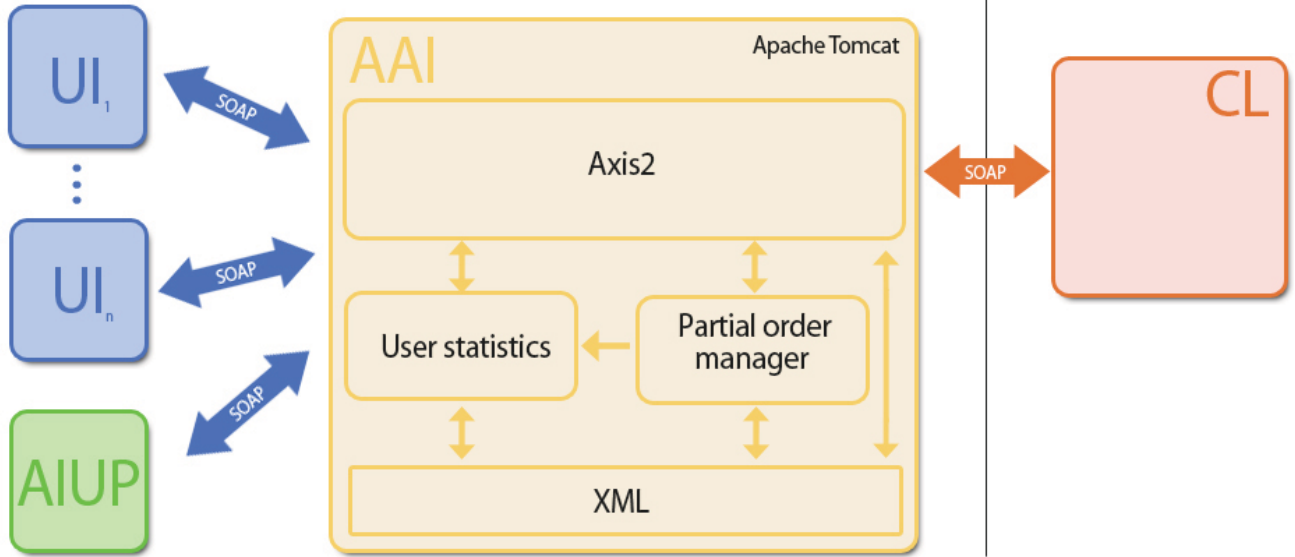


# User layer

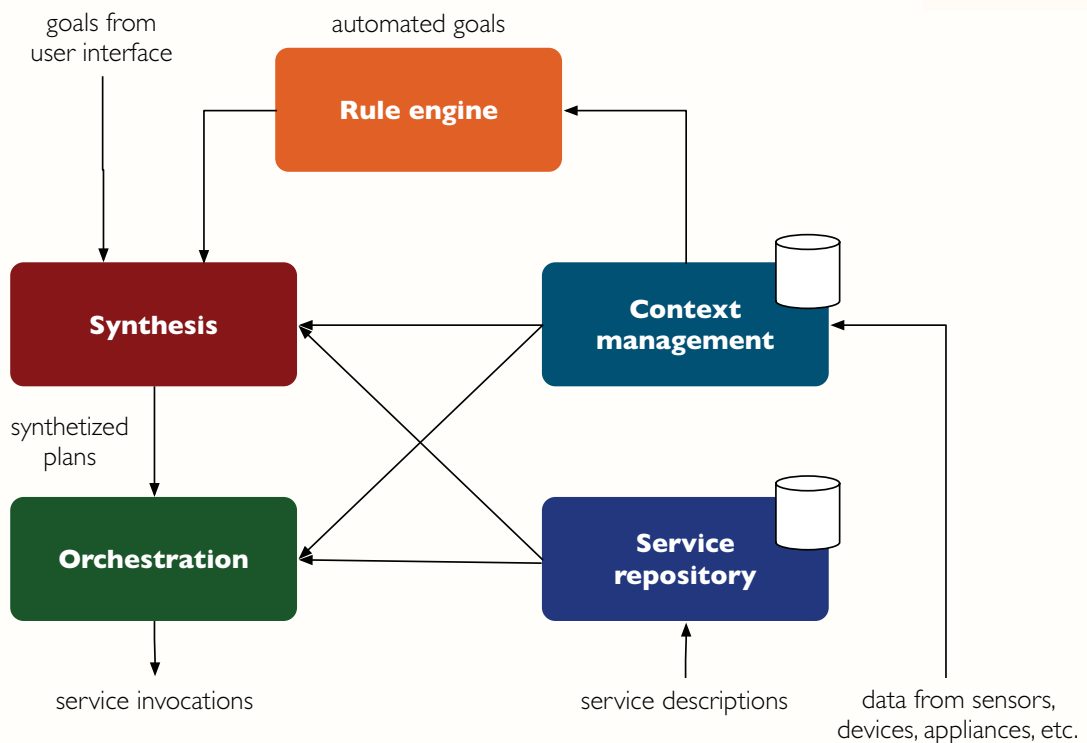
- Brain Computer Interface (BCI)



# User layer



# Composition layer

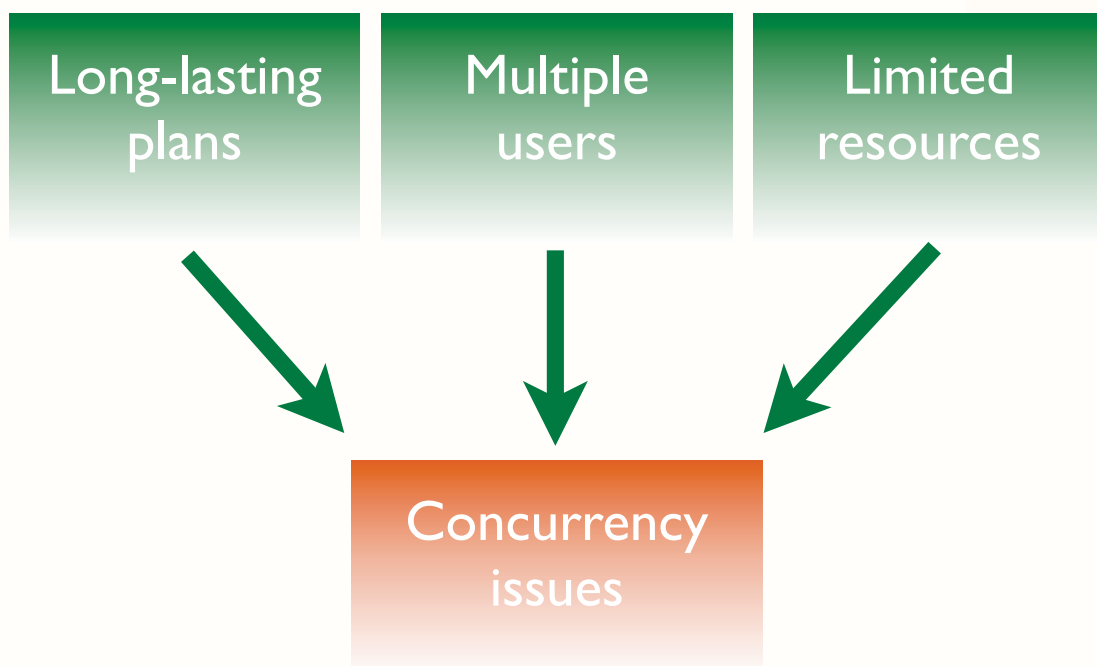


# Current status



- Software prototype
- Ongoing implementation and testing of the various software components
- Planned integration camps
- Simulation-based eval. on real disabled users
- Showcase next summer in “Casa Agevole” FSL
- Research challenges:
  - Concurrency management
  - Failure detection

# Concurrency management



Deadlocks, starvation, etc.

# Concurrency management



- Well-known problem in many different areas:

- Databases



- Distributed systems

- The dining philosophers



- The driving philosophers



# Concurrency management



Peculiar characteristics of our scenario:

- Different resources have different usage patterns
- Exogenous factors impact resource availability/status

What we look for:

- An adequate formalization of concurrency in the specific scenario.
- New algorithms able to adapt with respect to continuously changing conditions.

# Failure detection



A fault on a device can lead to possibly severe errors in the system.

- Inconsistency between the real environmental status and its representation within the system.

Devices have different self-diagnosis capabilities:

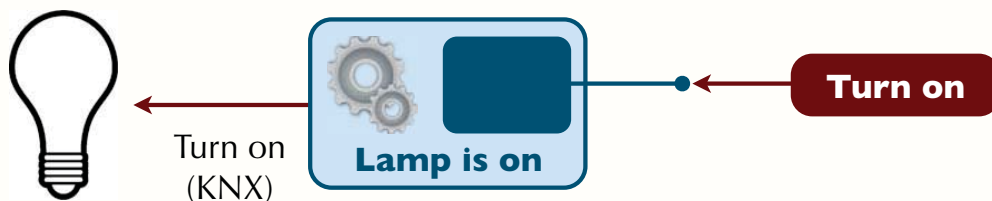
- Modern TV set → full self diagnostics
- Old-style lamp with automated KNX switch → no diagnostics

# Failure detection



Goal: "Turn on main lamp in the bedroom"

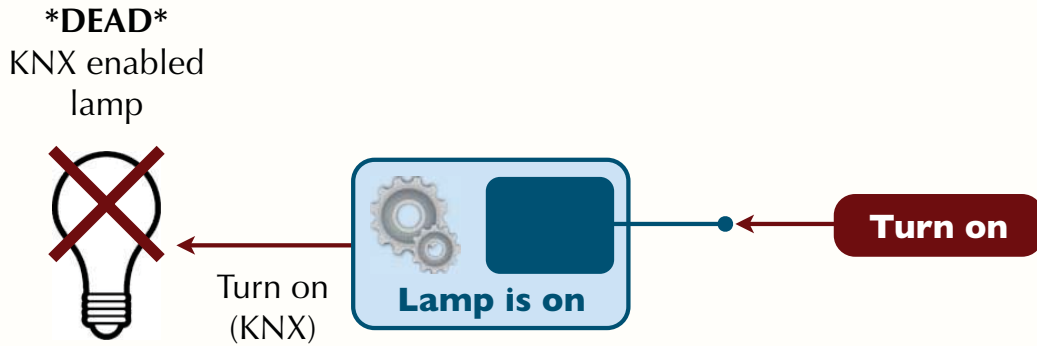
KNX enabled  
lamp



# Failure detection



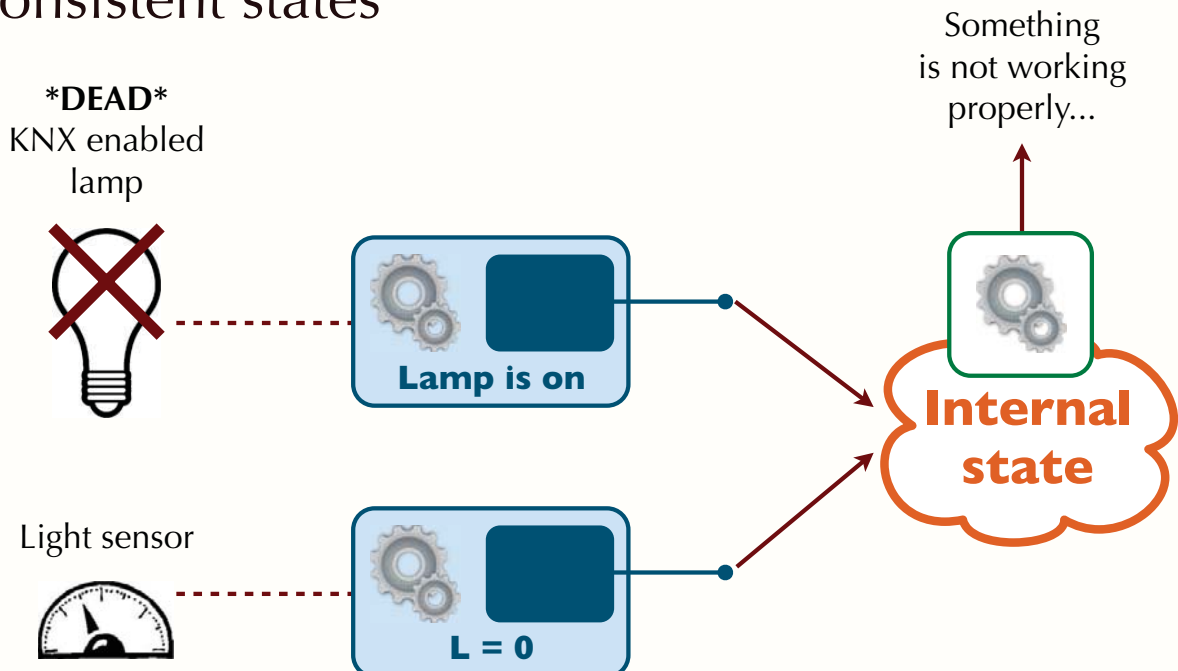
Goal: "Turn on main lamp in the bedroom"



# Failure detection



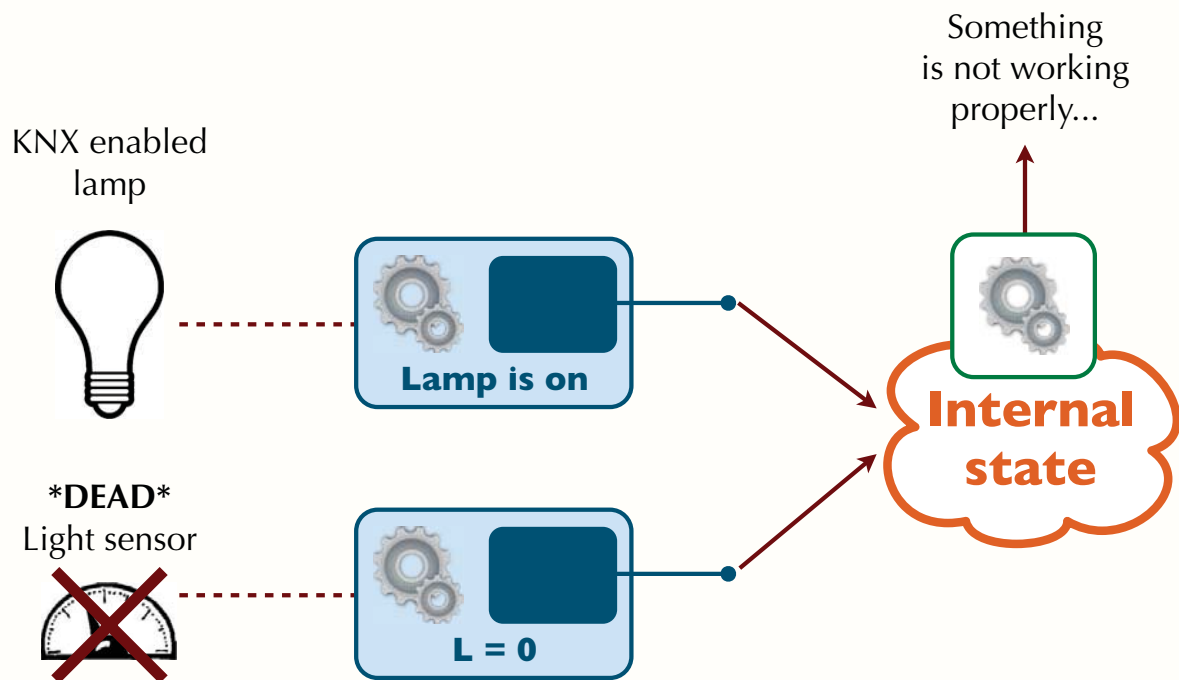
Exploit multiple device to discover inconsistent states



# Failure detection



How to break symmetry ???



# Failure detection



What we look for:

- A characterization of faults for our scenario.
- A characterization of inconsistent states.
- A theory that defines which faults can be detected in which situation.
- Techniques to track down the source of the problem.