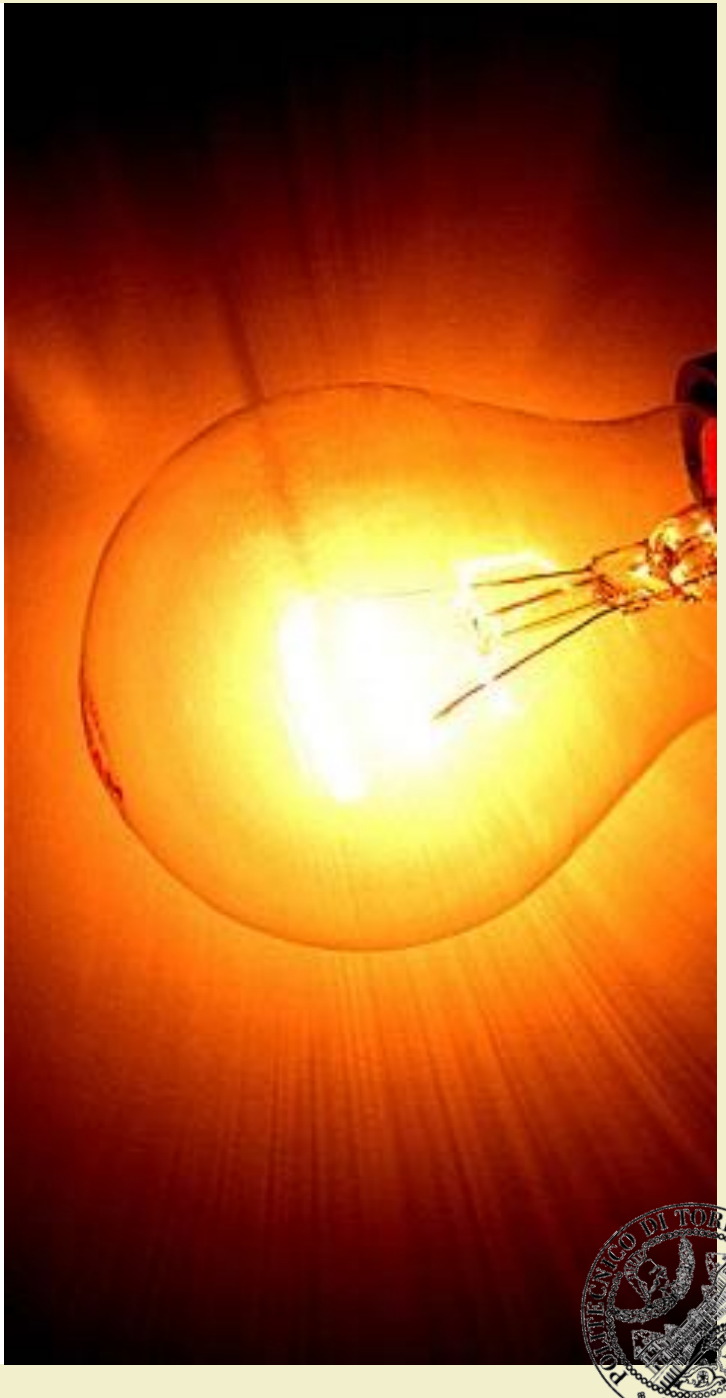
Python

Interfacing (smart) devices

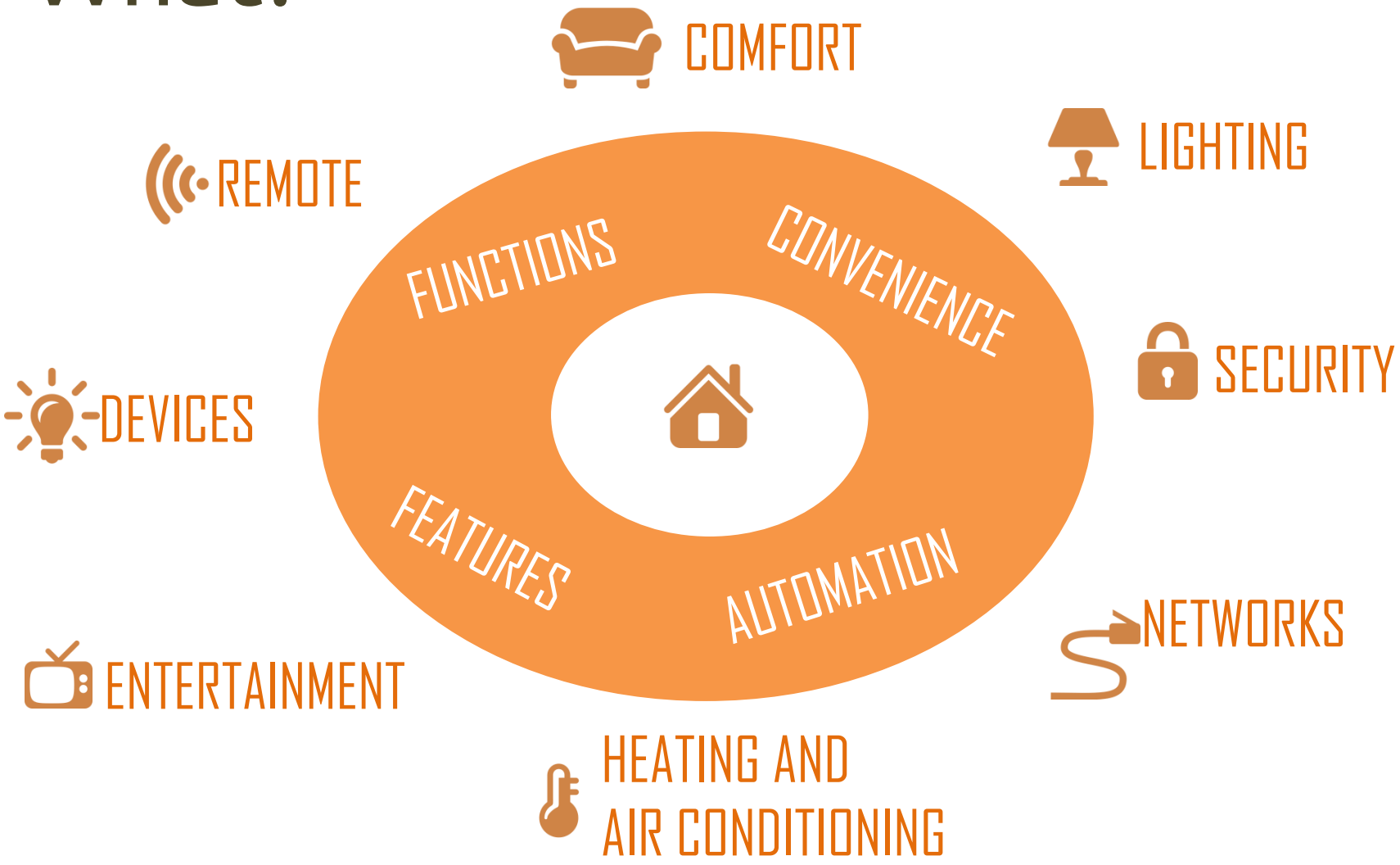
How to control and integrate smart objects and devices using different network technologies



Goals

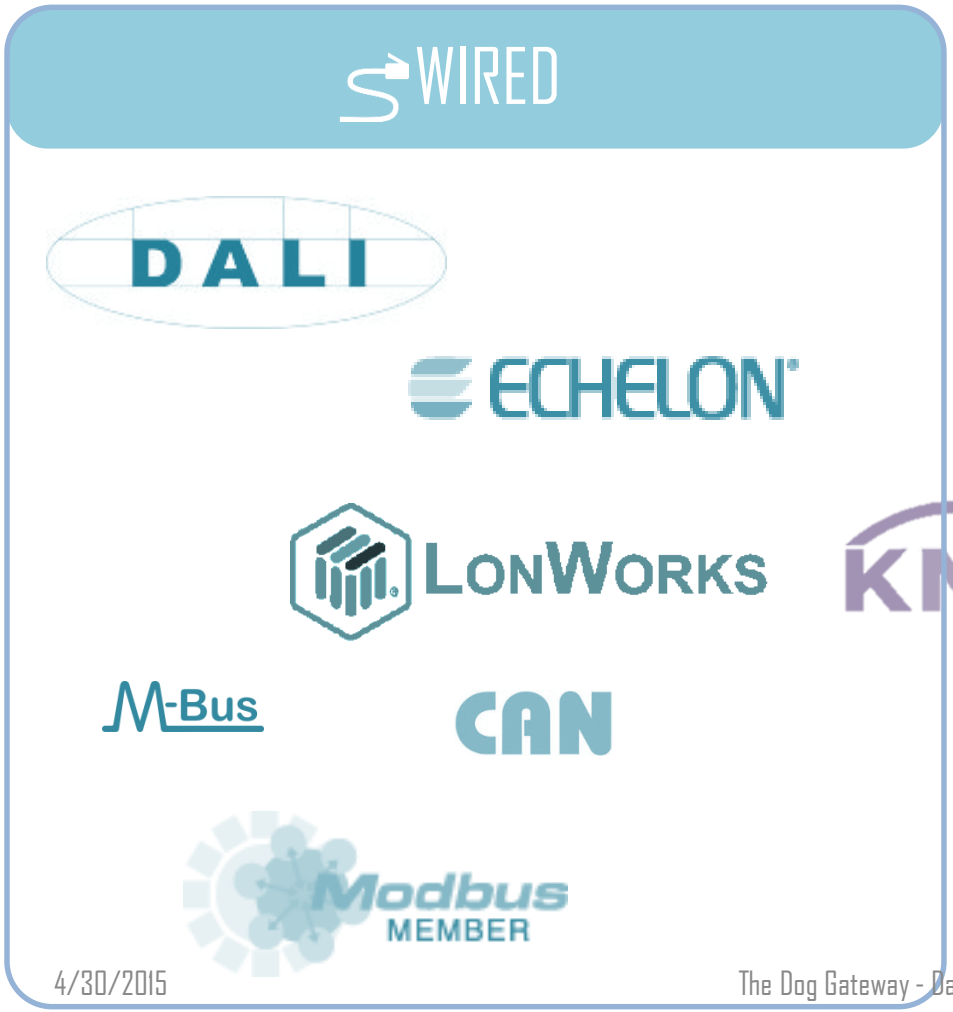
- Understand REST API specifications
  - Use case: the Dog gateway
- Understand device interoperation
- Develop a Python script to:
  - Turn on for 10s all lamps connected to one Dog gateway instance, without caring of the underlying technology
- Develop a simple interoperation script using Python and Dog

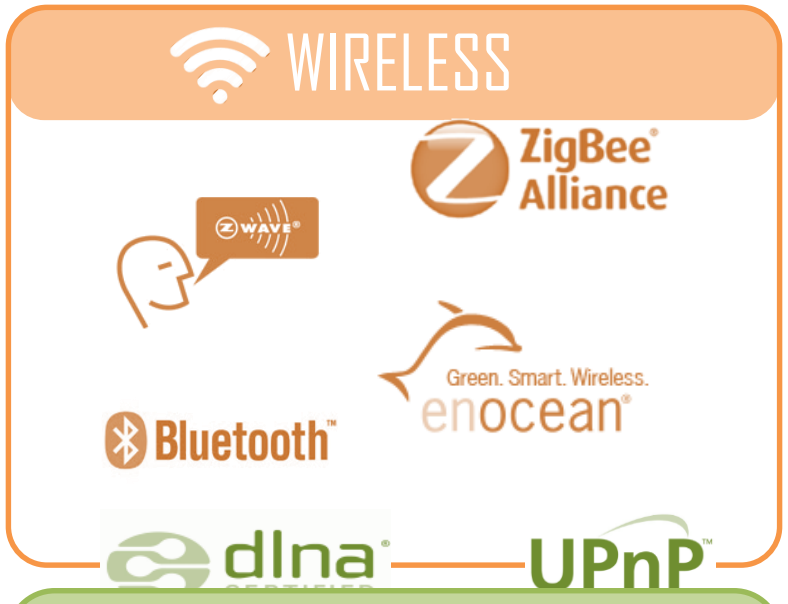
What?

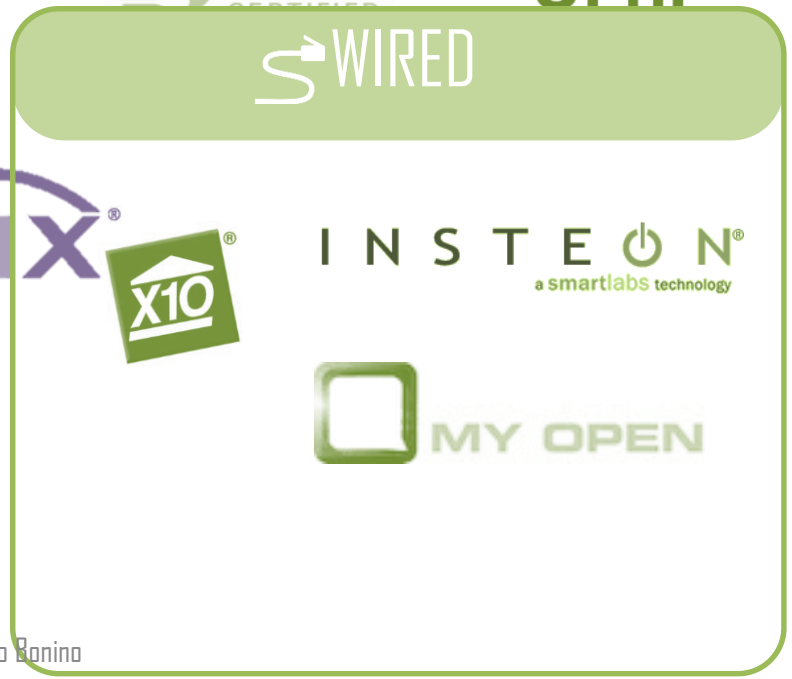

COMFORT
LIGHTING
REMOTE
FUNCTIONS
CONVENIENCE
SECURITY
DEVICES
FEATURES
AUTOMATION
NETWORKS
ENTERTAINMENT
HEATING AND
AIR CONDITIONING

However

MOST DIFFUSED NETWORKS







Abstraction / Information hiding

SEGREGATION OF THE "PARTS" THAT ARE MOST LIKELY TO CHANGE, THUS PROTECTING OTHER COMPONENTS FROM EXTENSIVE MODIFICATION IF THE "PARTS" ARE CHANGED

APPLICATION PROGRAMMING INTERFACE

SINGLE
SLOWLY EVOLVING

INFORMATION HIDING

NETWORK

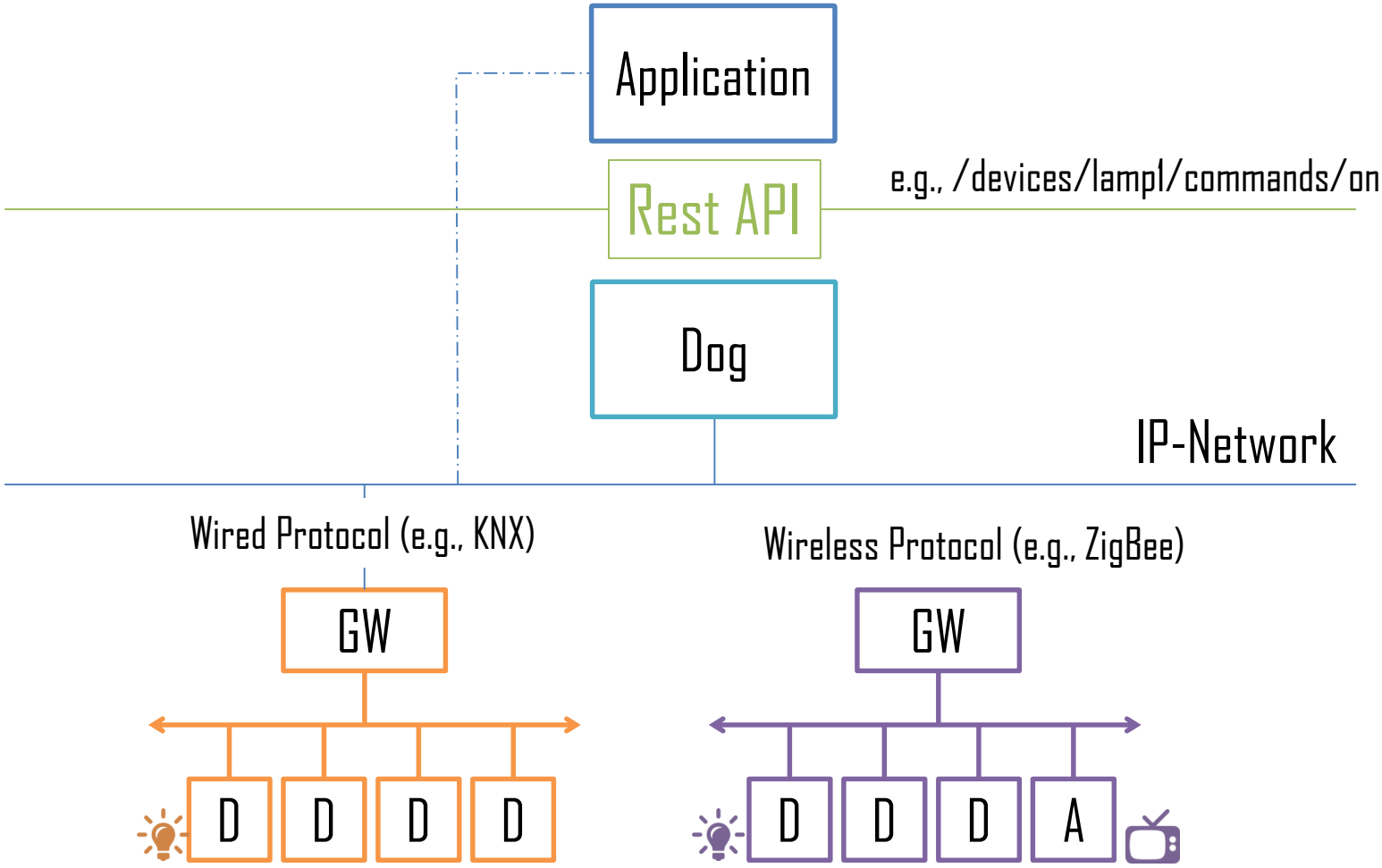
CHANGES IN PARADIGMS
CHANGES IN TIME
CHANGES IN FEATURES

Dog

THE GATEWAY

Architecture, technology and APIs

General Architecture



Dog RESTful API

- Quickly evolving / constantly updated
- Technology independent (based on DogOnt)
- Transfers XML / JSON
- 3 main APIs
  - Devices
  - Environment
  - Rules

Device API

- Allows to manage connected devices:
  - query the gateway about installed devices, their location, functionalities and configurations;
  - require execution of commands to existing devices;
  - monitor device statuses and measures in real-time;
  - add, modify or update the set of devices controlled through the gateway;

Device API - Query

Resource: /devices

GET

http://the.dog.address/devices

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dhc:dogHomeConfiguration>
<dhc:controllables>
<dhc:device domoticSystem="ELITE" id="oven1" class="ElectricalOven">
        <dhc:description>A ElectricalOven instance named oven1</dhc:description>
        <dhc:isIn>kitchen</dhc:isIn>
        <dhc:pluggedIn>MainsPowerOutlet_p12_kitchen</dhc:pluggedIn>
        <dhc:controlFunctionality class="OnOffFunctionality">
            <dhc:commands>
                <dhc:command name="on" class="OnCommand"/>
                <dhc:command name="off" class="OffCommand"/>
            </dhc:commands>
```

Device API – Execute Commands

Resource: /devices/{device-id}/commands/{command-name}

PUT

http://the.dog.address/devices/lamp1/commands/on

On

Device API – Status

Resource: /devices/status
Resource: /devices/{device-id}/status

GET http://the.dog.address/devices/lamp1/status

```
{
   "id" : "lamp1",
   "description" : " The lamp over the closet near to the livingroom armchair",
   "active" : true,
   "status":[
   {
      "OnOffstate" : "on"
   }
   ]
}
```

Device API - Update

A lamp

PUT

http://the.dog.address/devices/lamp1

```
{
    "description" : "The Lamp
near to the armchair
"
}
```

The Lamp near to the armchair

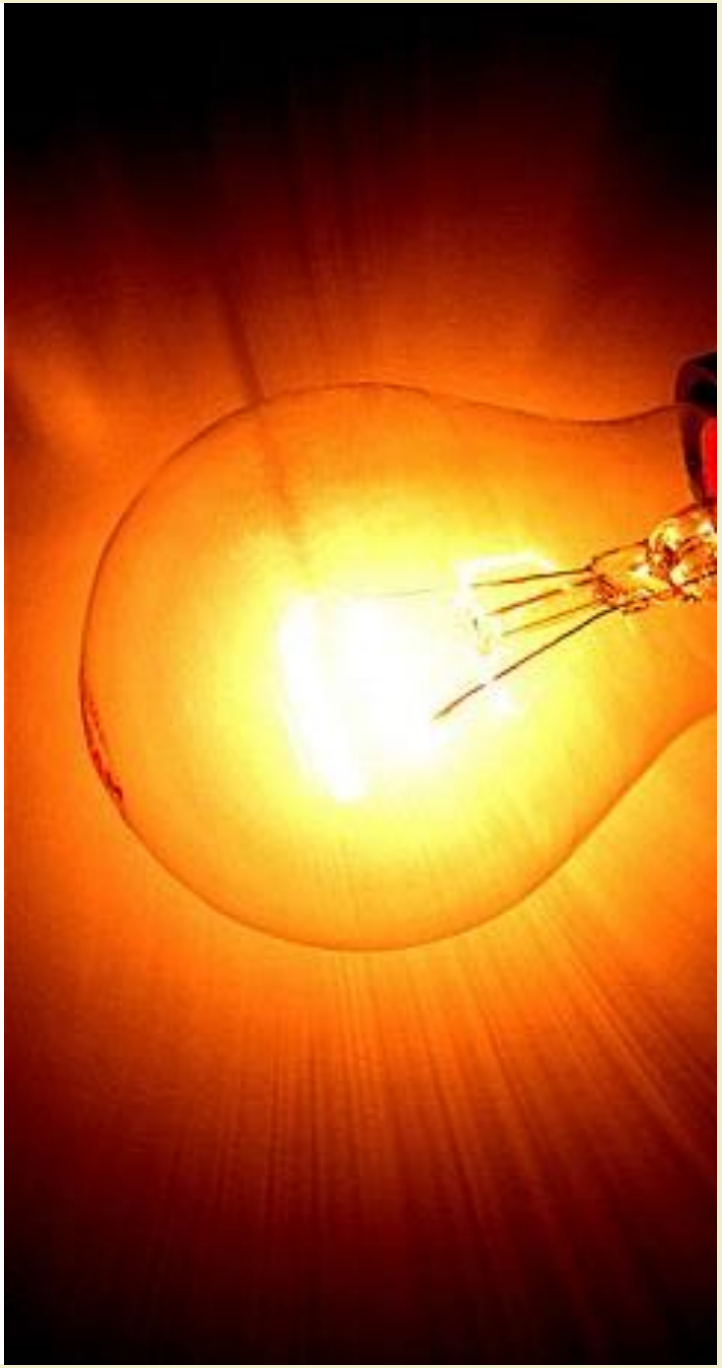
Result

- Checkout on GitHub
  - https://github.com/AmI-2015/python-device-integration.git

Questions?

01PRD Ambient Intelligence: Technology and Design

Dario Bonino
bonino@ismb.it



POLITECNICO DI TORINO
e-Lite
SOME RIGHTS RESERVED
BY NC SA