Towards a Zero-Configuration Wireless Sensor Network Architecture for Smart Buildings

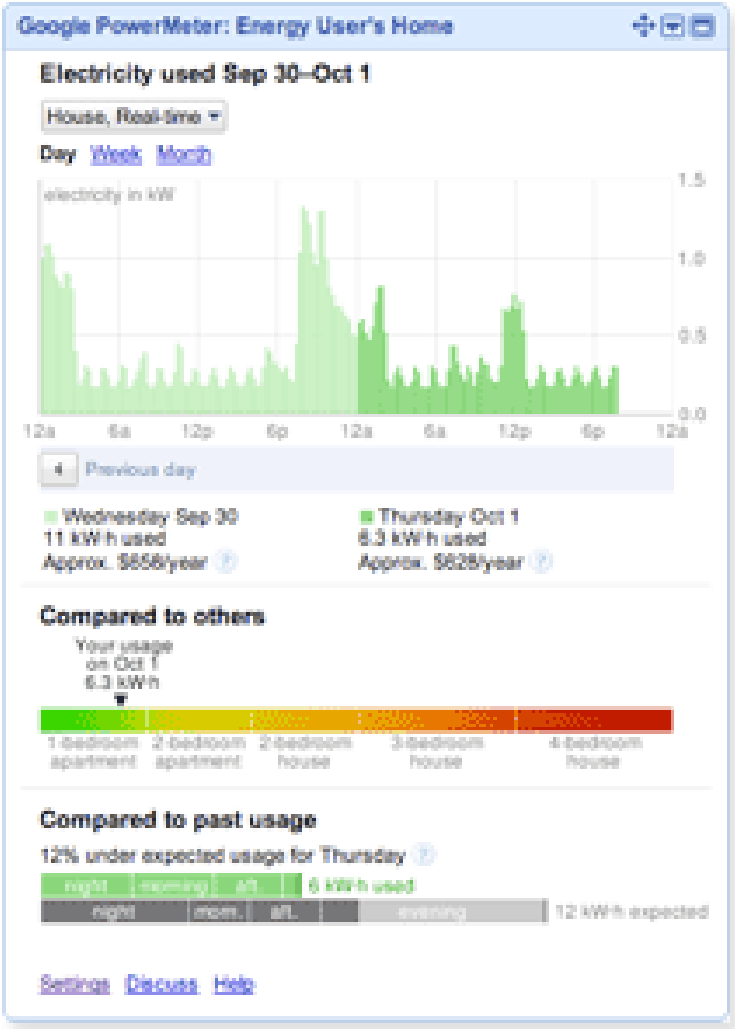
Lars Schor
Philipp Sommer
Roger Wattenhofer

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

Smart Buildings

- 40% of total energy use is for buildings
  - Make buildings greener and save money
- Building control systems get smarter
  - Learn where energy is used and how it can be saved
  - Increase comfort, health and safety of people
  - Integration with traditional IT systems



System Overview

- Sensor nodes
  - Report current energy usage, temperature, ...

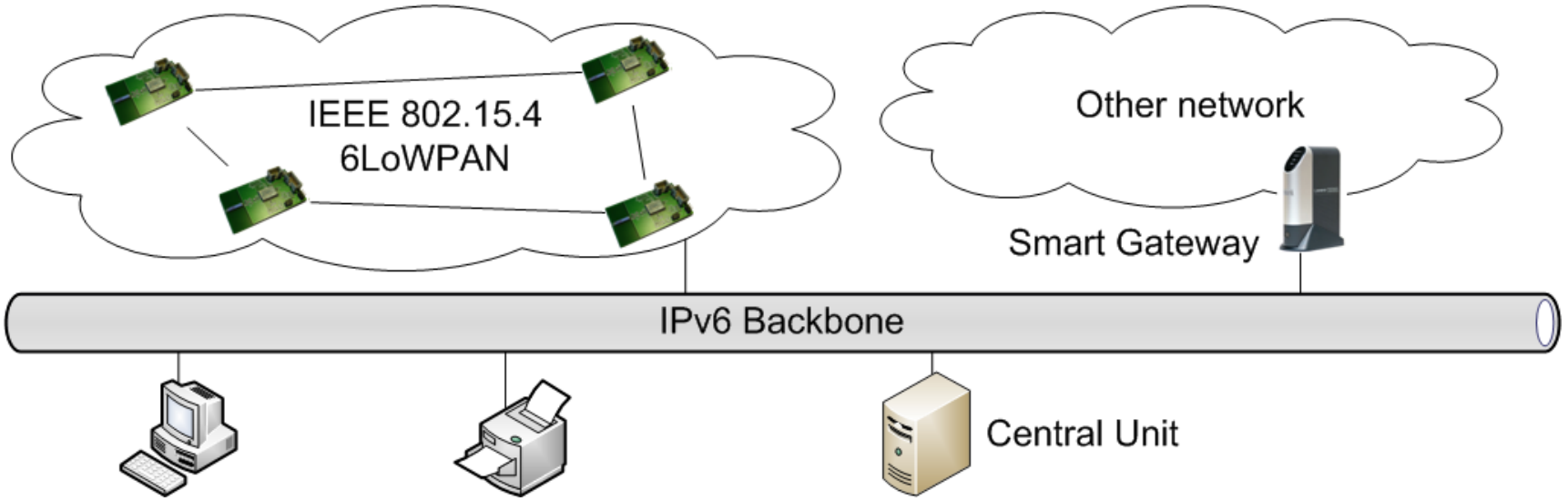
- Actuator nodes
  - Control small subsystems

- Central control unit
  - Coordinates different subsystems (actuators/sensors)

Integration of Sensor Nodes into IP-based Networks

Goals:

1. Integrate wireless sensor nodes into an existing IP-based infrastructure
2. Plug-and-Play system for building automation



Advantages of using IPv6 for Wireless Sensor Nodes

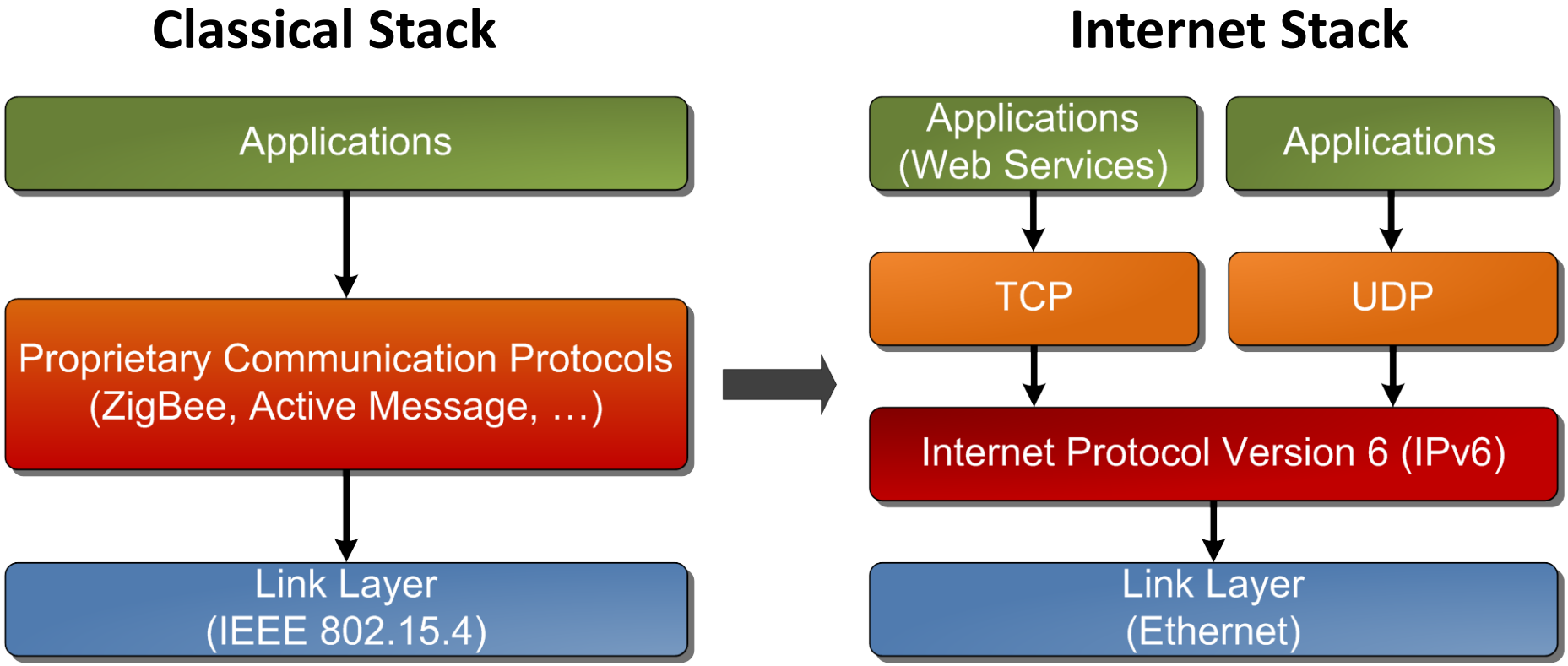
- Connectivity
  - Connect sensor with other devices in the Intranet/Internet
  - No application layer gateway required
- Use of standard networking tools/protocols in WSNs
  - ping, telnet, http, web services, ...
- Scalability
  - 128-bit address space

Target Platforms

- How can we connect devices which are fundamentally different?

	Sensor Node	Central Unit
CPU	8 MHz	2.4 GHz (4x)
RAM	8 kB	8 GB
Program Storage	128 kB	250 GB

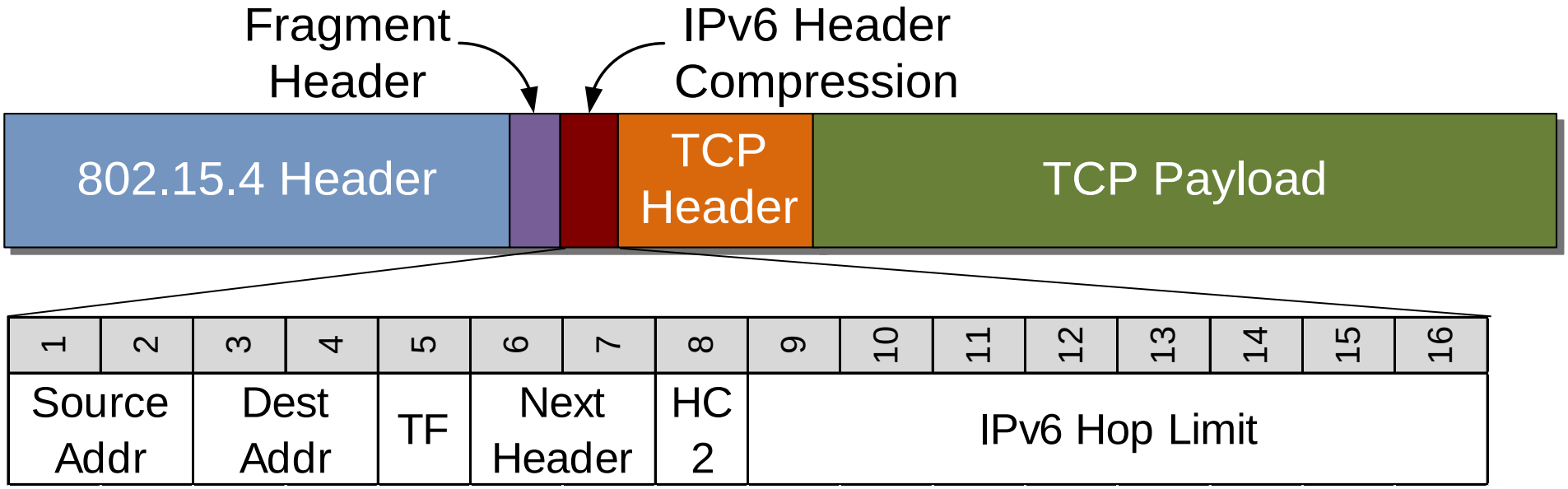
Communication Stack: WSN vs. IP



IPv6 over LoW Power wireless Area Networks (6LoWPAN)

- Problem:

- Solution:



How does…
… a device know about other devices?

Service Discovery

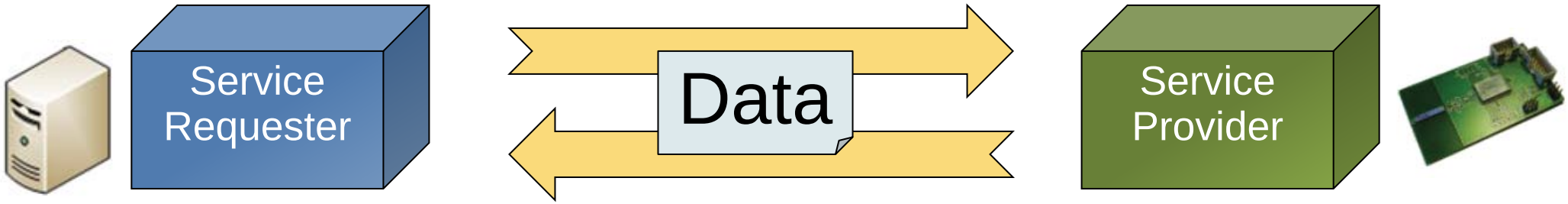
- Service announcements using multicast DNS packets
  - Implemented by Zeroconf/Bonjour/Rendez-vous
  - Service announcements using DNS SRV records (RFC 2782)
  - Implemented using UDP packets
  - Integrate new devices without additional configuration

Service: _rest._tcp.local
Hostname: sensor-1.local
Address: 2001:db8:0:8d3:0:8a2e:70:7344

How does…
… a device know which functionality another device provides

Web Services for Sensor Networks

- Web services enable machine-to-machine interaction



- Classic solution: SOAP and XML
  - Not optimal for memory-constrained sensor nodes
- Lightweight solution: REST and JSON

SOAP: Simple Object Access Protocol
XML: Extensible Markup Language
REST: Representational State Transfer
JSON: Java Script Object Notation

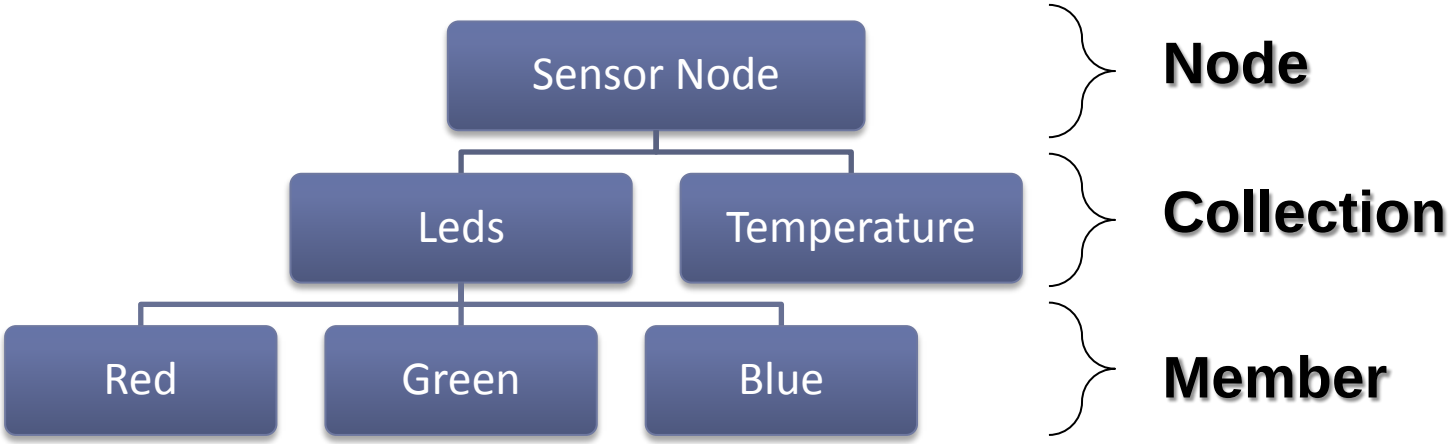
Representational State Transfer (REST)

- Functionality provided by a system is a set of resources
- Each resource can be identified using its distinct URI
  - Example: The red LED of a node can be accessed at "`/Leds/Red`"

- Web service based on REST:
  - Resources can be accessed using four basic operations provided by the HTTP protocol (GET, PUT, POST and DELETE)
- Not limited to sensor nodes!

REST: Resource Discovery and Access

- Example:



GET /Leds/Red HTTP/1.1

All Data of the Member

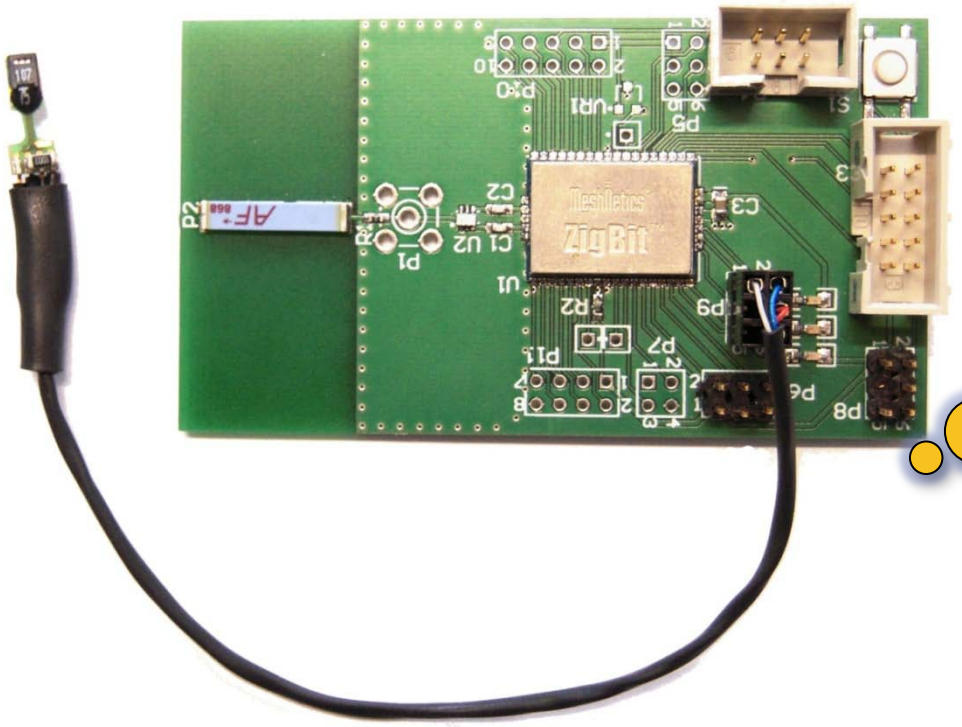
Response from the Sensor Node

- JavaScript Object Notation (JSON)

```
{
  "device": "temperature",   // name of the resource
  "method": [                // supported methods
    "G"                      // of the resource (GET)
  ],
  "param": [                 // array with all parameters
    {
      "n": "celcius",        // name
      "v": 26,               // value
      "t": "i",              // data type (integer)
      "u": 0                 // updatable
    }
  ]
}
```

Pixie Prototyping Platform

- Atmel ZigBit900 module
  - Cheap and flexible platform for prototyping sensor networks
  - Atmega1281 microcontroller + ATRF212 radio transceiver
  - Various IO pins (GPIO, SPI, I2C, UART)



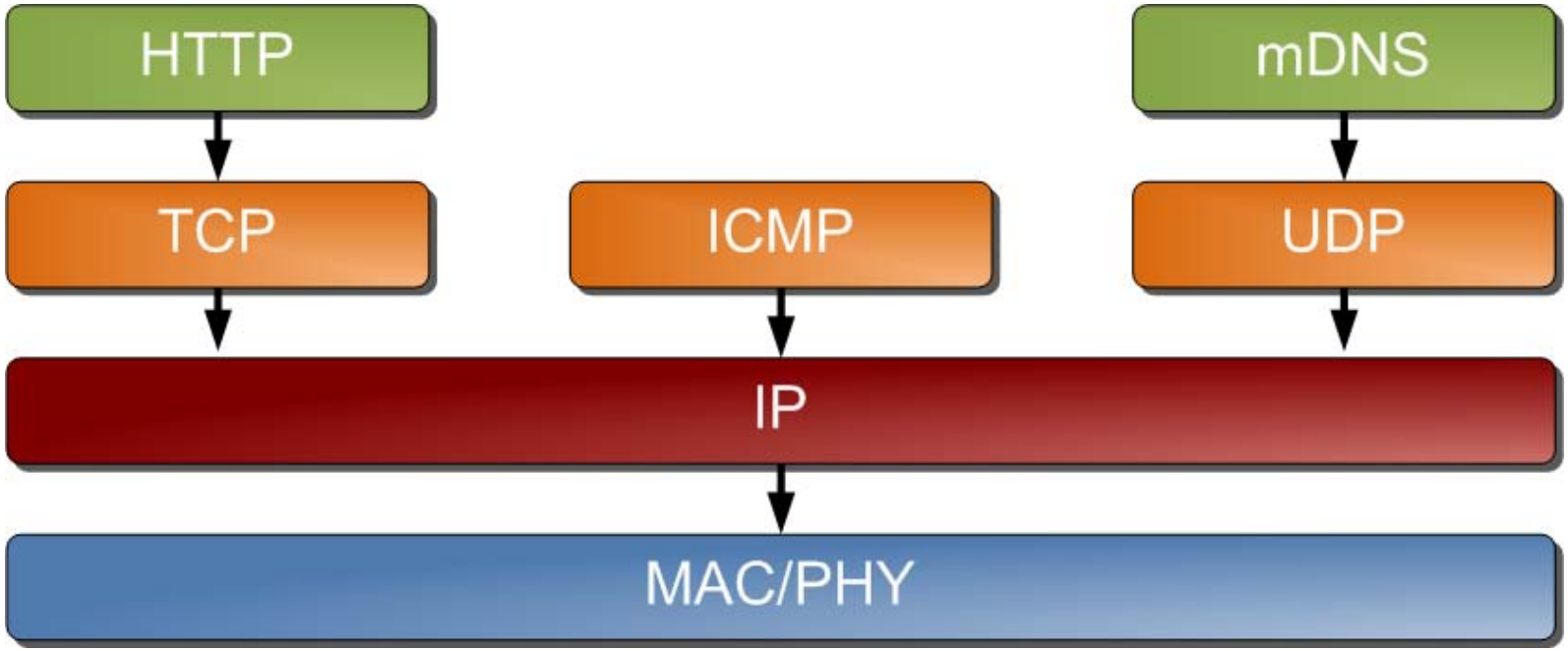
8 Kbytes RAM

128 Kbytes ROM

IEEE 802.15.4

TinyOS Implementation

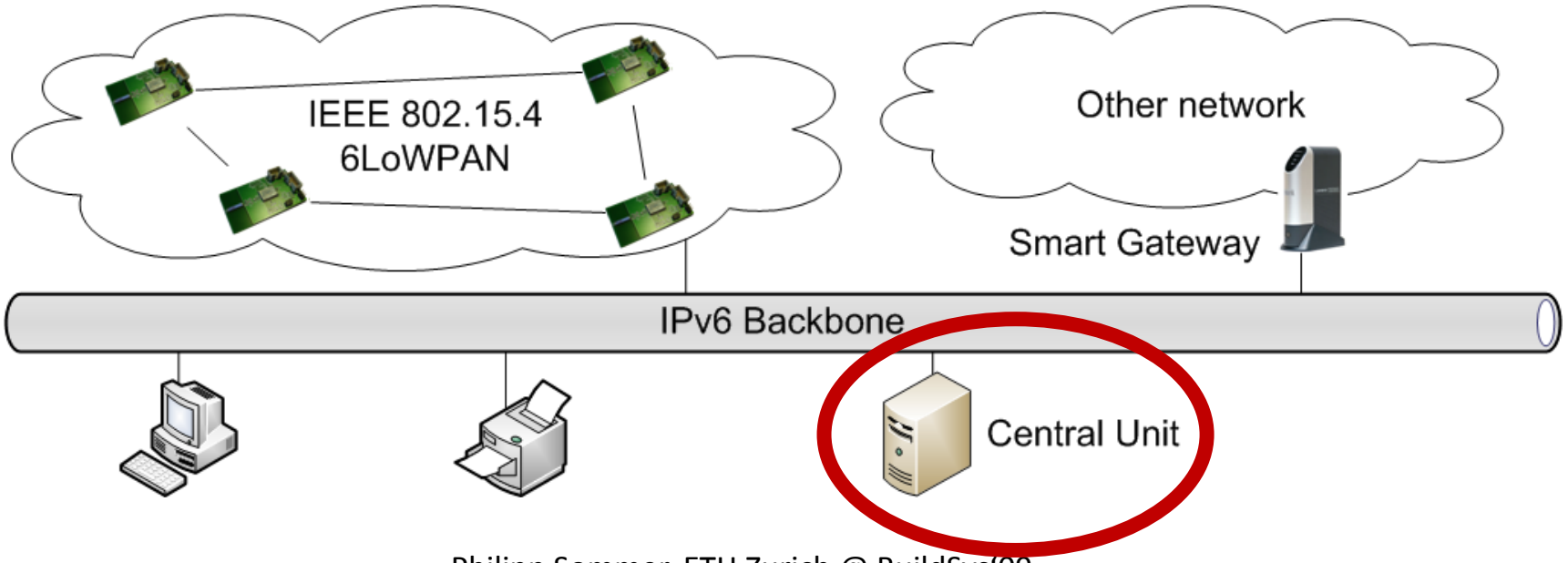
- We use **blip**: 6LoWPAN stack for TinyOS 2.1.1
- Implemented a small HTTP web server on each sensor node
- Service announcements using multicast DNS (mDNS) over UDP



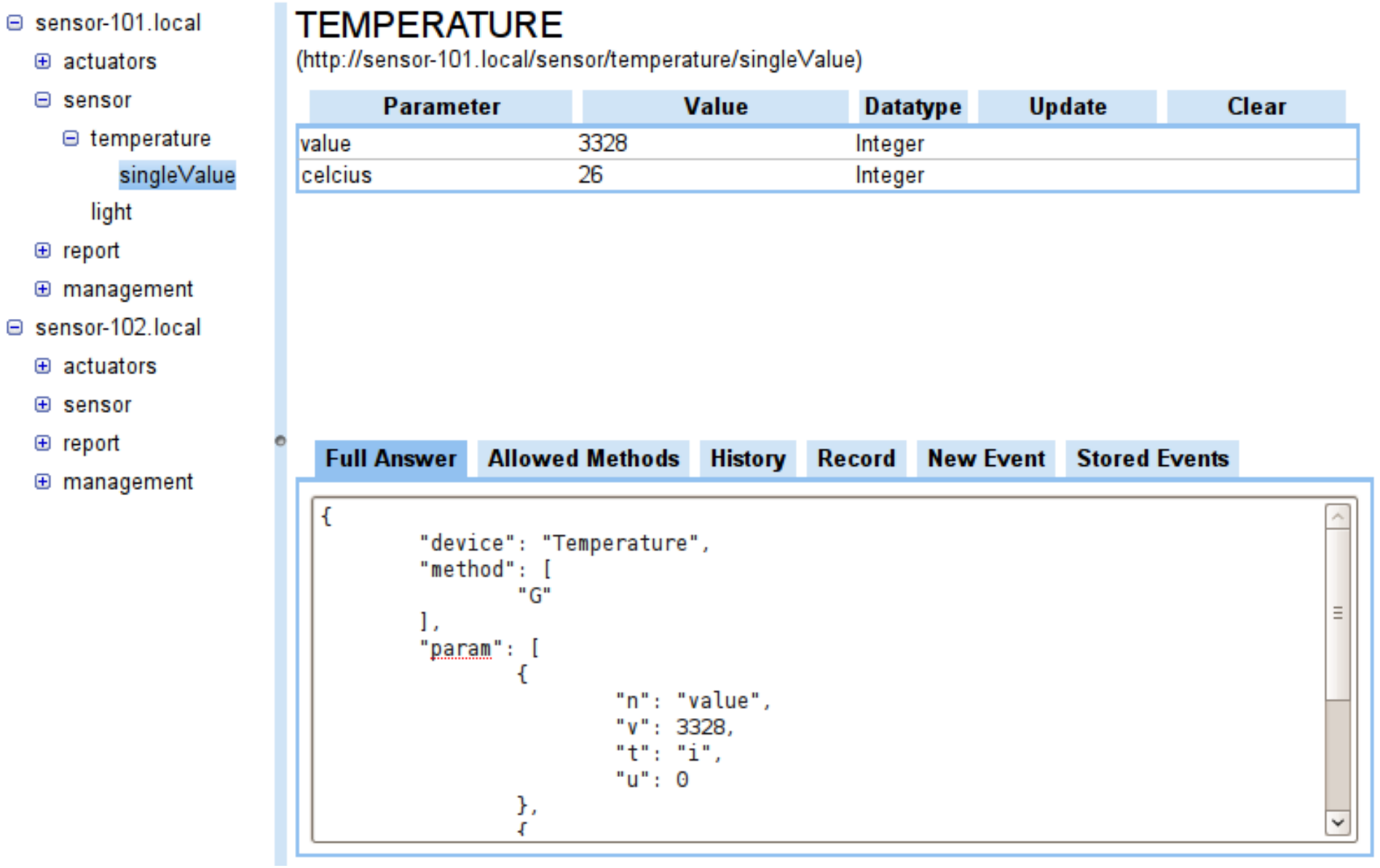
How does…
… this help to make buildings smarter

Integration into Smart Buildings

- Central unit
  - Get information from the sensors
  - Control the actuators
- Web application
  - Listen to service announcements from nodes
  - Rule-based control of actuators

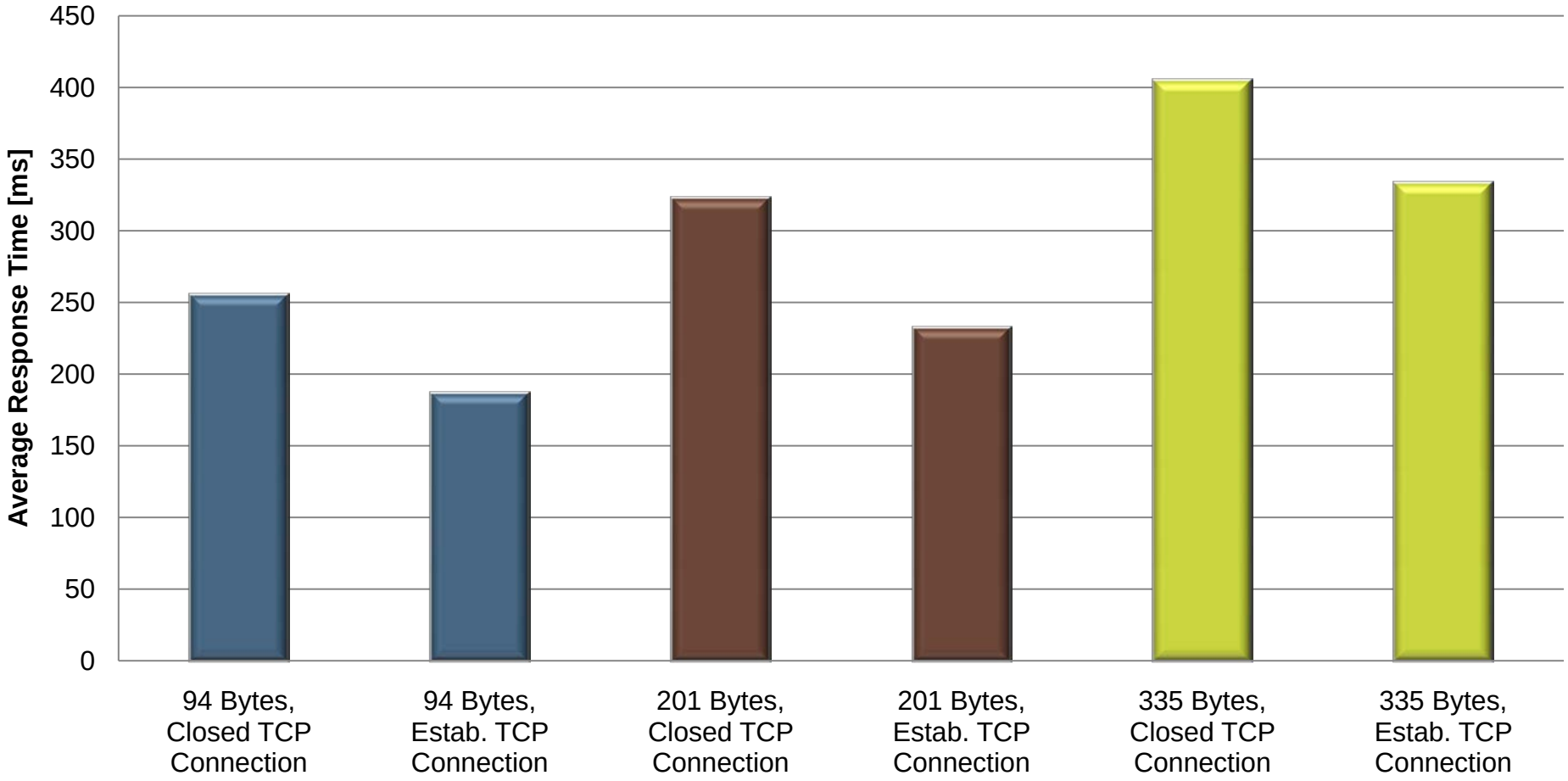


Web Application



Evaluation

- Response time for HTTP requests to the sensor node
  - Performance improvement by using persistent TCP connections



Summary

- Connected sensor nodes using standard Internet protocols
  - Lightweight web services based on REST/HTTP

- Zero-configuration approach
  - Service discovery using multicast DNS
  - Clients can fetch a list of services offered by a device

Service: „_rest._tcp.local“
Hostname: sensor-1.local
Address: 2001:db8:0:8d3:0:8a2e:70:7344

- And all this fits into less than 128kB ROM / 8kB RAM