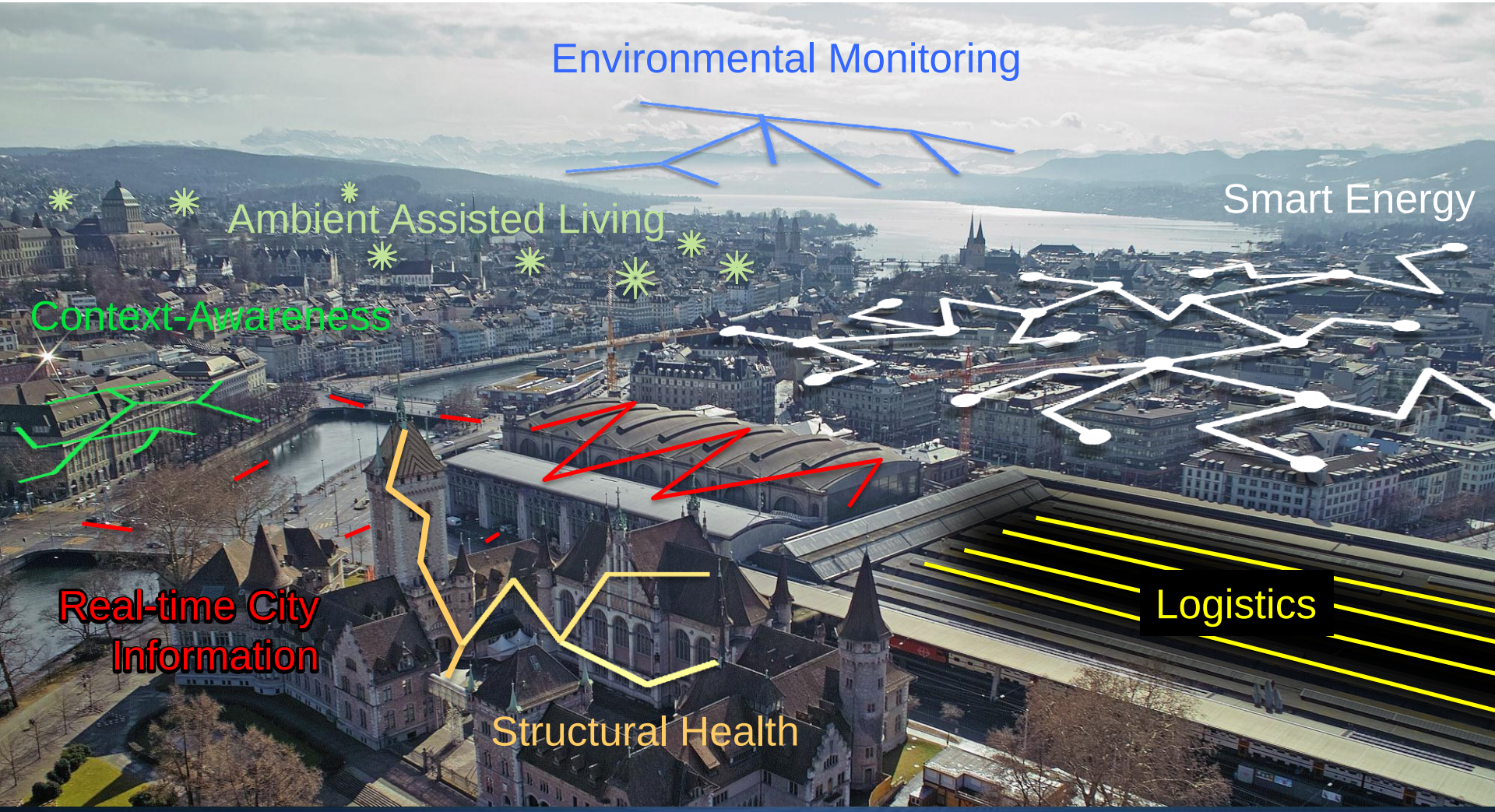
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Actinium: A RESTful Runtime Container for Scriptable Internet of Things Applications

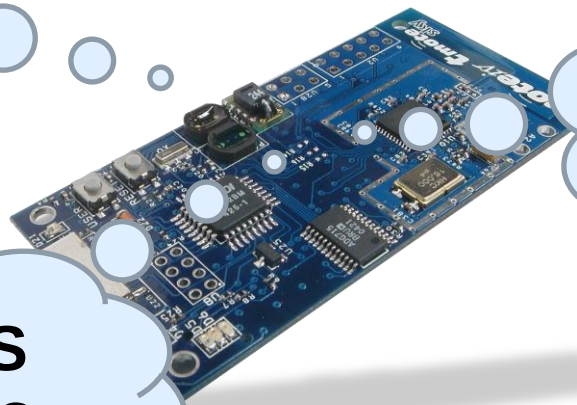
Matthias Kovatsch, Martin Lanter, and Simon Duquennoy
kovatsch@inf.ethz.ch

Wireless Sensor Networks for the IoT



Problem: Application Development

Class 1 devices
~100kB ROM
~10kB RAM



Macro-programming?
Islands!

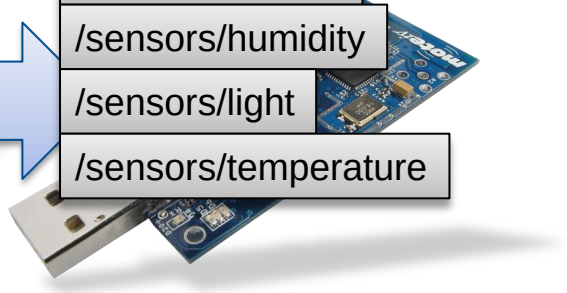
Embedded OS programming?
Experts!

Web of Things for Class 1 Devices

End-to-end RESTful interaction

Web server / REST API

- /actuators/leds
- /debug/channel
- /debug/neighbors
- /debug/observers
- /debug/routes
- /sensors/battery
- /sensors/button
- /sensors/humidity
- /sensors/light
- /sensors/temperature



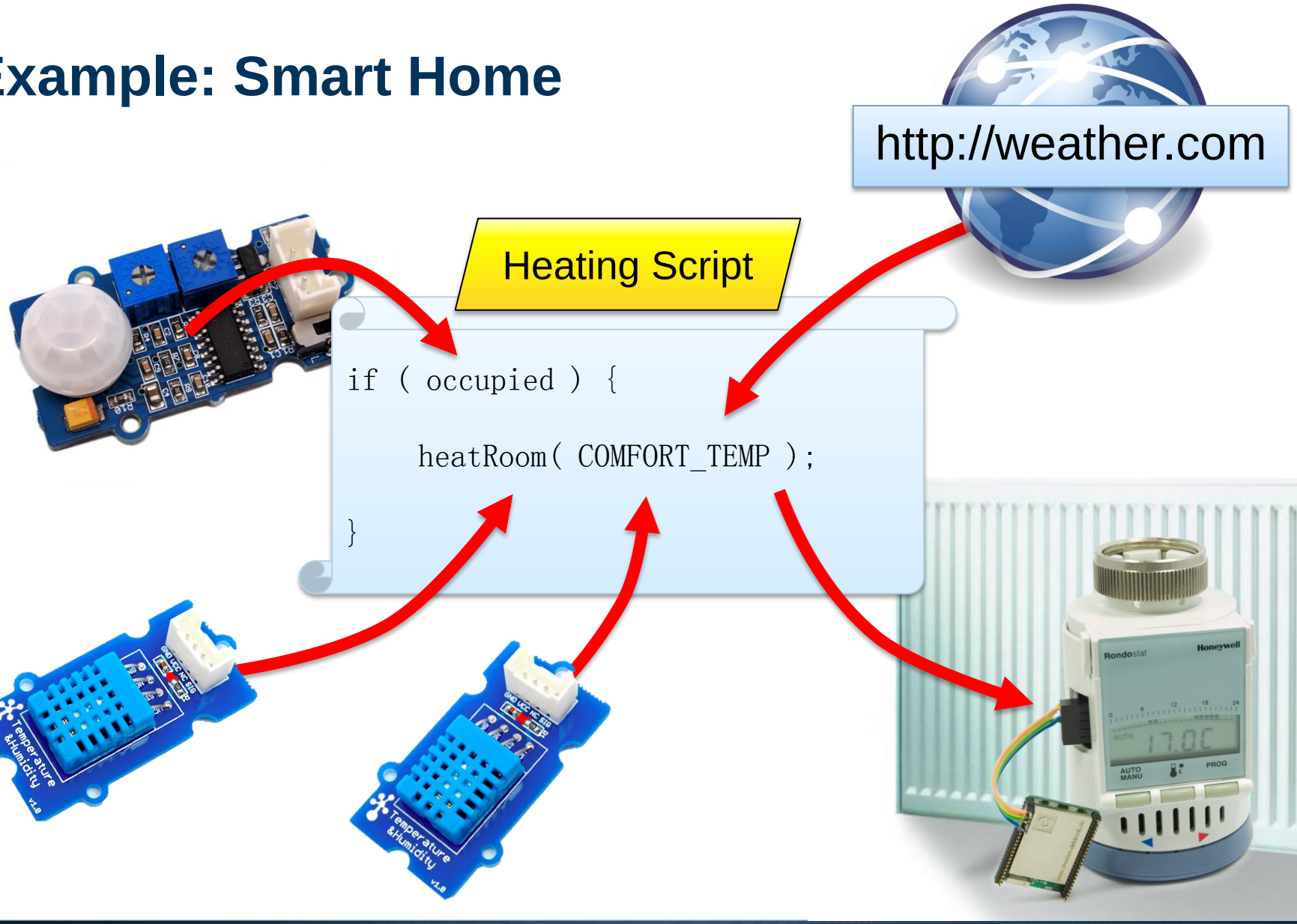
Web scripting

6LoWPAN

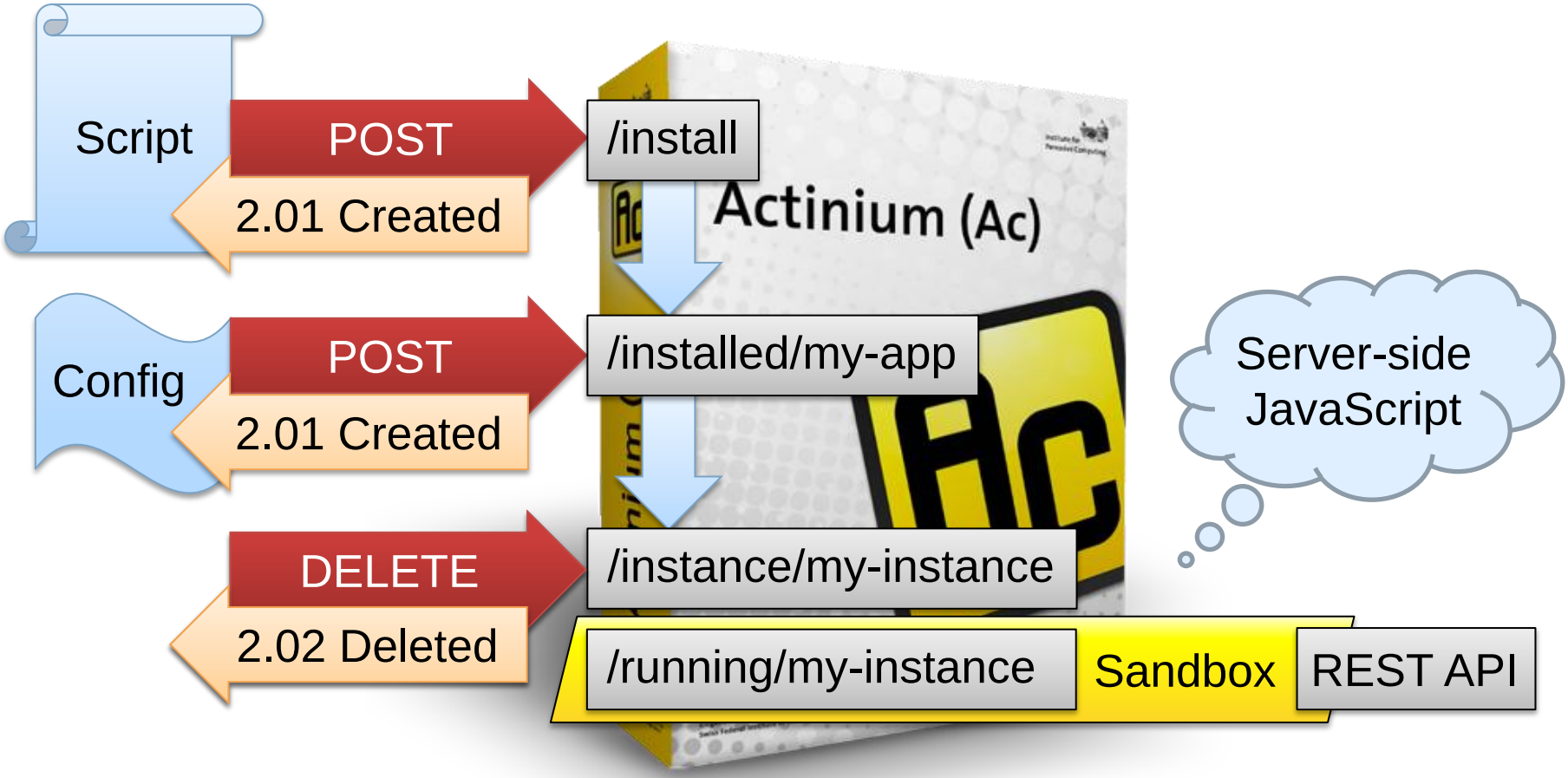
CoAP

IETF I-D: draft-ietf-core-coap-12

Example: Smart Home



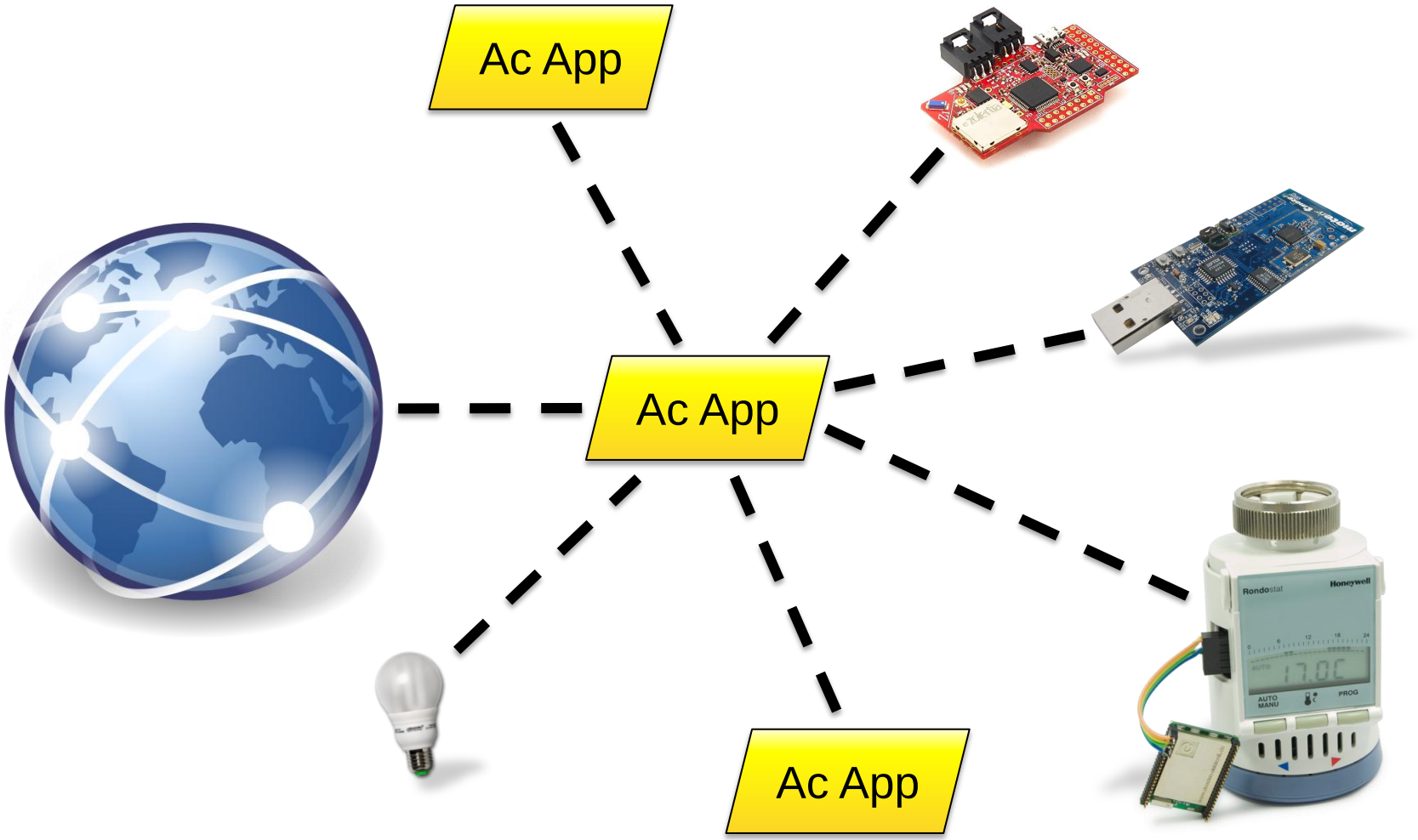
Actinium RESTful Runtime Container



Actinium Apps Are REST Resources

```
// handler for GET requests to "/"
app.root.onget = function(request) {
// returns CoAP's "2.05 Content" with payload
        request.respond(2.05, "Hello world");
    };

// sub-resource "/config"
var sub = new AppResource("config");

sub.onput = function(request) {
        variable = request.payloadText;
        request.respond(2.04); // "2.04 Changed"
    };
app.root.add(sub);
```

Mashups



Mashups with Classic Web Services

```
// Usual AJAX
var xhr = new XMLHttpRequest();

// GET weather conditions from Web service
xhr.open("GET", "http://api.wunderground.com/…", false);

xhr.send();

// Parse the retrieved JSON data and mash it up
var condition = JSON.parse(xhr.responseText);
```

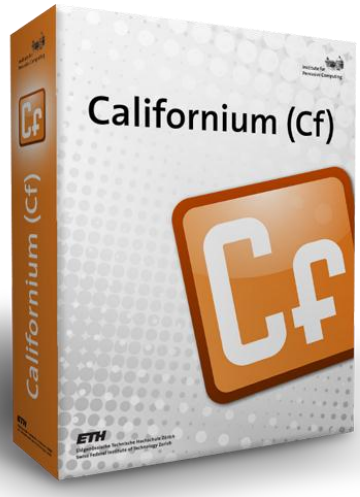
Mashups with Devices and other Apps

```
var req = new CoapRequest();
// Request the temperature sensor reading via CoAP
req.open("GET", "coap://mote1.example.com/sensors/temp",
              false /*synchronous*/);

// Set Accept header to application/json
req.setRequestHeader("Accept", "application/json");

req.send(); // blocking

// Use IoT data just like Web data
var roomTemperature = JSON.parse(req.responseText);
```

CoapRequest with CoAP-specific Features

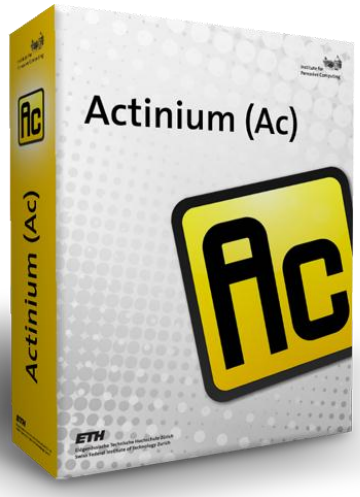
```
req.open("GET", "coap://app-server.example.com/
    running/occ-room1/occupancy", true /*asynchronous*/);

// Register for CoAP Observe push notifications
req.setRequestHeader("Observe", "0");

// define the callback for push notifications
req.onprogess = function() {
        switchLights(this.responseText=="true");
    };
// define the callback for normal/final response
req.onload = function() {
        handleNonObservable(this);
    };

req.send(); // non-blocking
```

Implementation: https://github.com/mkovatsc



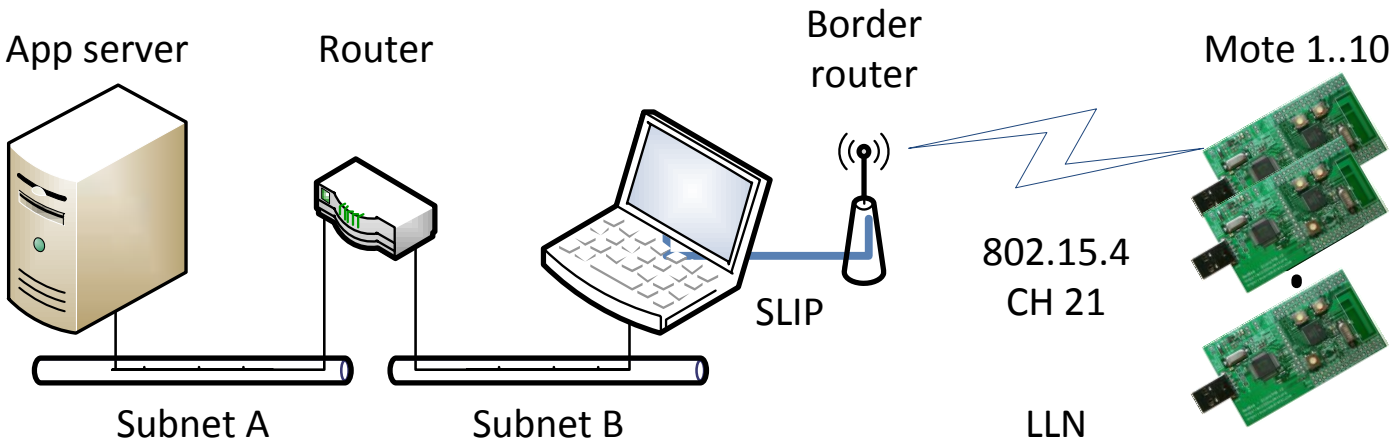
+

=



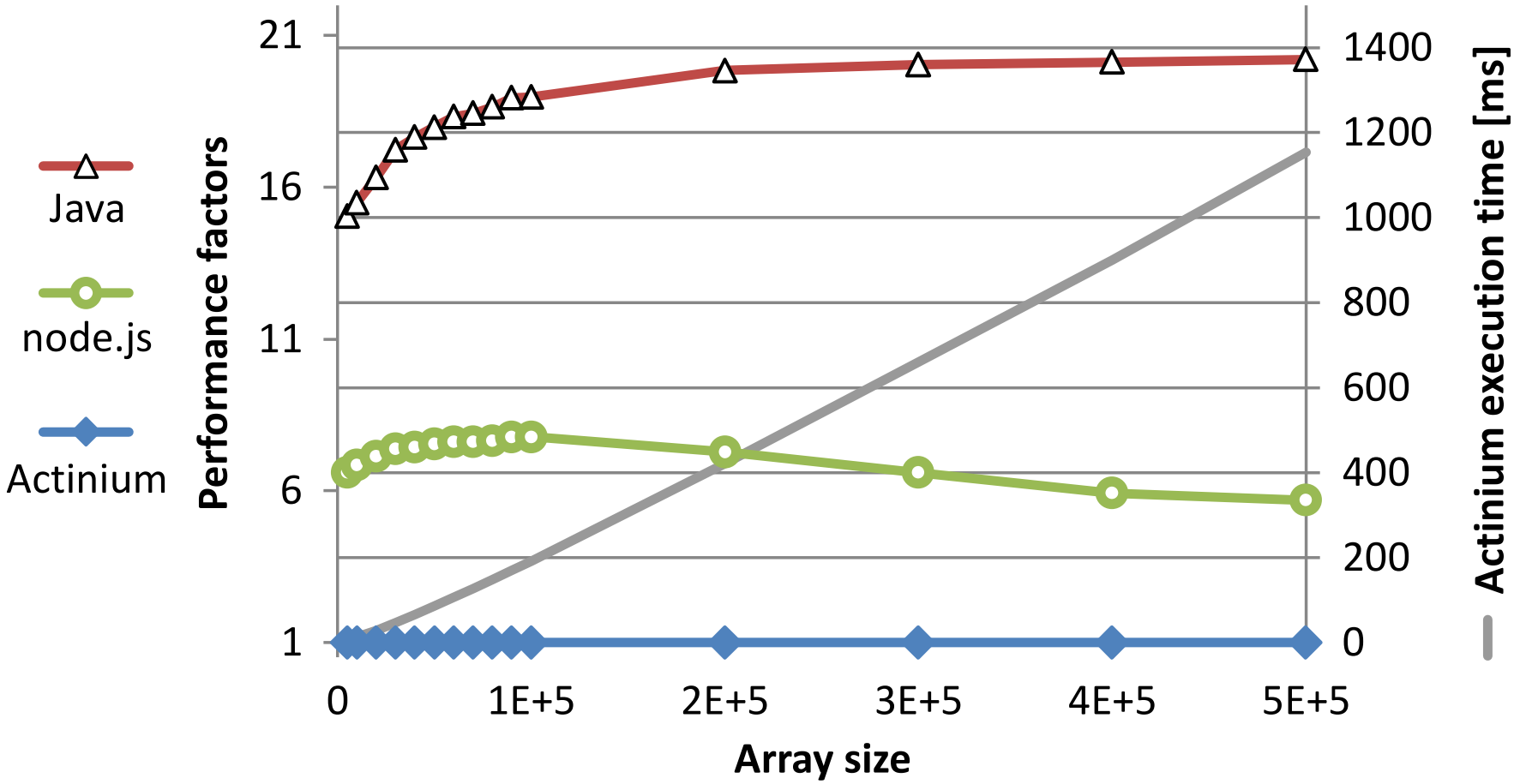
Californium CoAP framework	Mozilla Rhino JavaScript Engine	Actinium (Ac) App-server

Ac Round-Trip Time Overhead

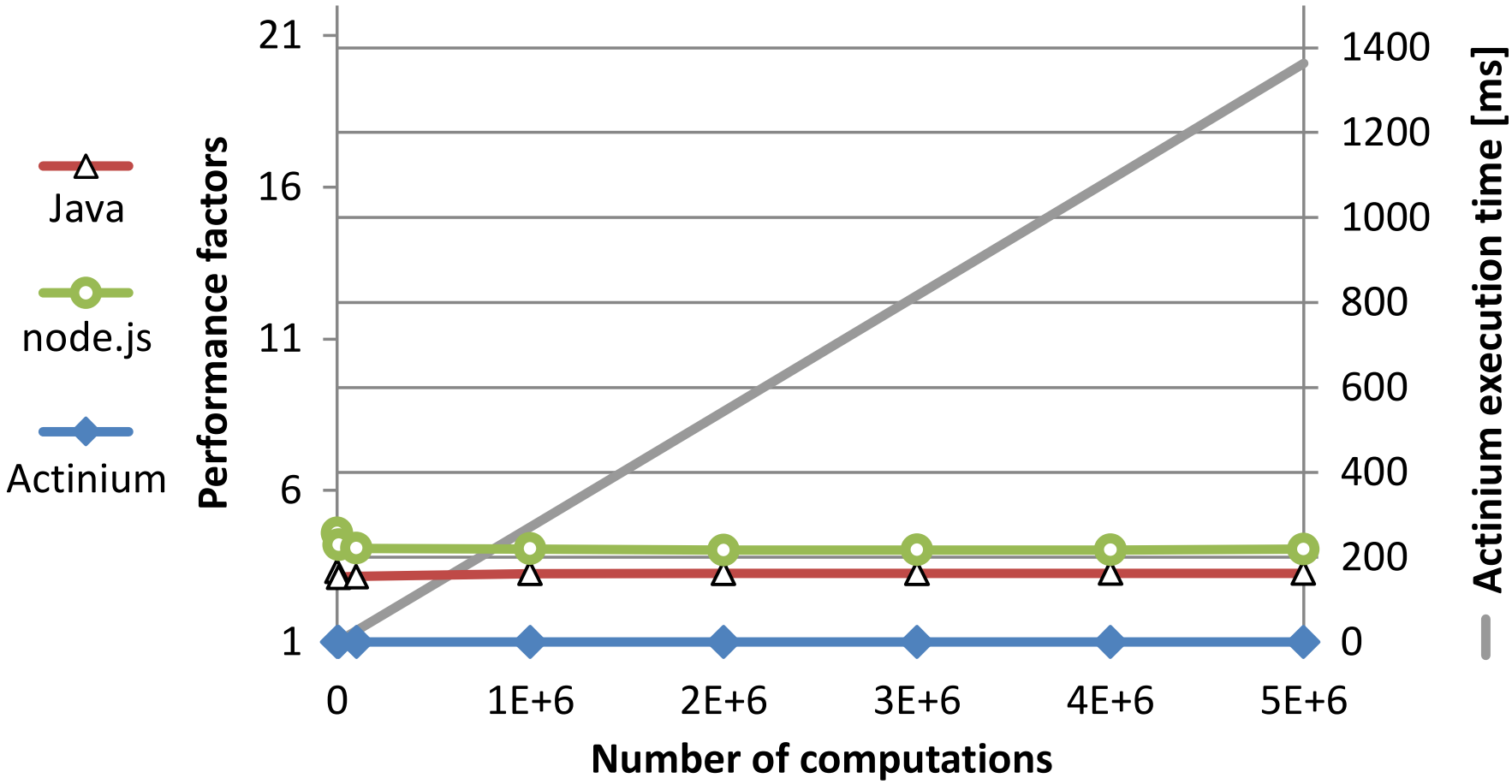


1000 Requests	Minimum	Maximum	Average	Overhead
Ping	32ms	77ms	46ms	
CoAP (Cf)	34ms	78ms	48ms	+2ms
Actinium	33ms	97ms	49ms	+1ms

Ac Performance: Quicksort (Memory-intensive)



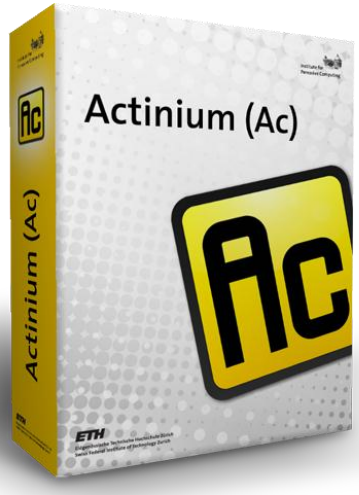
Ac Performance: Newton (Arithmetic-intensive)



Conclusion

- Applications for Class 1 devices can easily be scripted
  - Web-like IoT application development
  - Seamless Web integration
- Performance governed by low-power networking
  - Scripting well-suited for application logic
  - Native implementations for memory-intensive tasks
- Security considerations (in the paper)

- Get Actinium from GitHub
  - ***https://github.com/mkovatsc***



THANK YOU

Questions?

Publicly Available Resources

- Open Source Software under 3-Clause BSD on GitHub:
  - ***https://github.com/mkovatsc***
- Maven repository:
  - ***http://maven.thingml.org/***
- Copper (Cu) Firefox add-on:
  - ***https://addons.mozilla.org/en-US/firefox/addon/copper-270430/***
- CoAP test server:
  - ***http://vs0.inf.ethz.ch/*** for detailed information

The CoAP Universe

- IETF working group «Constrained RESTful Environments»

- Constrained Application Protocol (CoAP)
  - RESTful Web services for mote-class devices
  - Transparent mapping to HTTP
- Observing Resources in CoAP
- Group Communication for CoAP
- Blockwise Transfers in CoAP
- CoRE Link Format
- CoRE Resource Directory
- …