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

- Environmental Monitoring
- Ambient Assisted Living
- Smart Energy
- Context-Awareness
- Real-time City Information
- Structural Health
- Logistics
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

Actinium (Ac)

Web scripting

6LoWPAN

CoAP

I-D: draft-ietf-core-coap-12
Example: Smart Home

```
if ( occupied ) {
    heatRoom( COMFORT_TEMP );
}
```

http://weather.com
Actinium RESTful Runtime Container

- **Script** POST /install "2.01 Created"
- **Config** POST /installed/my-app "2.01 Created"
- **DELETE** /instance/my-instance "2.02 Deleted"

Server-side JavaScript
Actinium Apps Are REST Resources

```javascript
// 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

```javascript
// 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

```javascript
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

```javascript
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.onprogress = 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 (Cf) + Mozilla Rhino JavaScript Engine = Actinium (Ac) App-server
## Ac Round-Trip Time Overhead

### 1000 Requests

<table>
<thead>
<tr>
<th></th>
<th>Minimum</th>
<th>Maximum</th>
<th>Average</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ping</strong></td>
<td>32ms</td>
<td>77ms</td>
<td>46ms</td>
<td></td>
</tr>
<tr>
<td><strong>CoAP (Cf)</strong></td>
<td>34ms</td>
<td>78ms</td>
<td>48ms</td>
<td>+2ms</td>
</tr>
<tr>
<td><strong>Actinium</strong></td>
<td>33ms</td>
<td>97ms</td>
<td>49ms</td>
<td>+1ms</td>
</tr>
</tbody>
</table>
Ac Performance: Quicksort (Memory-intensive)

![Graph showing performance factors for Ac, Java, and node.js over array size.](image)

- **Ac** shows a stable performance with slight fluctuations.
- **Java** has a steep increase in execution time.
- **node.js** exhibits a moderate increase in execution time.

Performance metrics:
- Array size
- Execution time [ms]
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](https://github.com/mkovatsc)

- Maven repository:
  - [http://maven.thingml.org/](http://maven.thingml.org/)

- Copper (Cu) Firefox add-on:

- CoAP test server:
  - [http://vs0.inf.ethz.ch/](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
- ...