Building the Internet of Things with Eclipse IoT*

Benjamin Cabé – Eclipse Foundation
@kartben

* and more!
The Number of IoT Developers 2014-2020

Source: VisionMobile estimates, 2014

Report: IoT: Breaking Free From Internet And Things | vmob.me/IoT
©VisionMobile | June 2014 | Licensed under CC BY ND
• 19 open-source projects*
• Lots of Java but also C, C++, Python, Go, .Net, …

→ IoT Standards
→ Services & Frameworks

* and counting!
End-to-end IoT?
End-to-end IoT?
End-to-end IoT?
End-to-end IoT?
What you will learn today

CONNECT
What you will learn today

CONNECT

MANAGE
What you will learn today

CONNECT  MANAGE  VISUALIZE
Connecting things to the IoT?

Network is often not reliable
Bandwidth == $$$
Different communication patterns
Connecting things to the IoT

- **CoAP**
  - « HTTP over UDP »
  - Expose your device as a resource to the Internet of Things

- **MQTT**
  - Publish/Subscribe model
  - TCP-based
CoAP: The web-of-things

- /walk
- /hand/left/raise
- /eye/picture

- /on
- /red
- /green
- /blue
- /mtbf

- /engine/status
- /position
- /fuel

- /buttons
- /buttons/1/push
- /bat-level

- /CO2
- /noise
- /lights/on
Eclipse Californium

- Focus on scalability and usability
- To be used in IoT cloud servers or M2M/IoT devices running Java
- Includes DTLS implementation (Scandium), HTTP/CoAP bridge, Plugtests, ...

http://eclipse.org/californium
1. Implement custom resources (extend CoapResource)
2. Add resources to the CoAP server
3. Start the server
import static org.eclipse.californium.core.coap.CoAP.ResponseCode.*;

public class MyResource extends CoapResource {
    @Override
    public void handleGET(CoapExchange exchange) {
        exchange.respond("hello world"); // reply with 2.05 payload (text/plain)
    }

    @Override
    public void handlePOST(CoapExchange exchange) {
        exchange.accept(); // make it a separate response

        if (exchange.getRequestOptions() ... ) {
            // do something specific to the request options
        }
        exchange.respond(CREATED); // reply with response code only (shortcut)
    }
}

MQTT: Publish & Subscribe

Pub KETTLE232/temp
Payload: 21°C

BROKER

Pub KETTLE232/temp
Payload: 21°C

Sub KETTLE232/*

Pub KETTLE232/temp
Payload: 21°C

Pub KETTLE232/temp
Payload: 21°C
MQTT in 5 keywords

Pub-Sub
Wildcards
Quality of Service
Last Will & Testament
Retained Messages
Eclipse Paho

- Open-source MQTT clients
- Pick your language!
  - Java
  - JavaScript
  - C/C++, Objective C
  - Go, Lua, Python, .NET, WinRT, ...

http://eclipse.org/paho
MqttClient c = new MqttClient("tcp://iot.eclipse.org:1883",
        MqttClient.generateClientId());

mqttClient.setCallback(new MqttCallback() {
    @Override
    public void messageArrived(String topic, MqttMessage message)
            throws Exception {
        // process received message
        // ...
    }
});

mqttClient.connect();
mqttClient.subscribe("mygateway/#");
Open source MQTT brokers

- **Eclipse Mosquitto**
  - C implementation
  - Pretty scalable (1000 clients == 3MB RAM)

- **But also…**
  - Moquette (Java, Based on Netty and LMAX disruptor)
  - VerneMQ (Erlang)
  - Mosca (Node.js)

Amazon just announced support for MQTT in their new AWS IOT cloud platform
Yup, lots of aspects to manage

- **Network**
  - PPP cellular connection, WiFi hotspot, Zigbee coordination, VPN, firewall …
  - offline/online mode

- **Applications**
  - Remote install, start, stop, configure, …
  - Sandboxing

- **Hardware**
Gateways to the rescue!
Eclipse Kura

Applications

App 1  App 2  . . . .  App n

Connectivity and Delivery

Network Management

Gateway Basic Services

Field Protocols

Device Abstraction

OSGi Application Container

Java VM

Linux

Hardware

Operation & Management

Administration GUI

Java VM

OSGi Application Container

Device Abstraction

Network Management

Gateway Basic Services

Connectivity and Delivery

Applications

App 1  App 2  . . . .  App n

Linux

Hardware

Administration GUI

Operation & Management
Installing Kura

cd ~
sudo apt-get update
wget https://s3.amazonaws.com/kura_downloads/raspbian/release/\1.3.0/kura_1.3.0_raspberry-pi-2_installer.deb
sudo dpkg -i kura_1.3.0_raspberry-pi-2_installer.deb
sudo apt-get install -f
sudo reboot
First steps with Kura

- Network management
  - Cellular Modem, WiFi
  - Firewall
  - NAT
- OSGi and system administration
- IoT server communication settings
First steps with Kura
First steps with Kura

### Device Information
- **Kura Version**: 0.2.0.2014072196322
- **Client ID**: E8:27:EB:82:CA:70
- **Display Name**: Raspberry-Pi
- **Uptime**: 2 days 19:39:42 hrs
- **Last WiFi Channel**: 11

### GPS Information
- **Latitude**: 0.0
- **Longitude**: 0.0
- **Altitude**: 0.0

### Hardware Information
- **Model Name**: Raspberry-Pi
- **Model ID**: Raspberry-Pi
- **Part Number**: Raspberry-Pi
- **Serial Number**: Raspberry-Pi

### Java Information
- **Java Virtual Machine**: Java HotSpot(TM) Client VM
- **Java Virtual Machine Version**: 24.0-b58
- **Java Runtime**: Java(TM) SE Runtime Environment 1.7.0_40-b43
First steps with Kura
First steps with Kura
First steps with Kura
First steps with Kura
First steps with Kura
First steps with Kura
Kura API

● OSGi services that you can re-use in your own components
  ○ ClockService
  ○ DataService, CloudService
  ○ CryptoService (AES, base64, SHA-1)
  ○ PositionService (geolocation)
  ○ … and many others

● And of course you can leverage a huge ecosystem of Java and OSGi libraries
CONNECT

MANAGE

VISUALIZE

✔

✔
End-user interaction

- JavaFX Charts
- Eclipse BIRT
- Smartphone app (e.g. Android)
  - https://www.eclipse.org/paho/clients/android
- MQTT + WebSockets = ♡
  - https://www.eclipse.org/paho/clients/js
MQTT + WebSockets

```javascript
var client = new Paho.MQTT.Client("ws://iot.eclipse.org/ws",
    "client-" + new Date().getTime());
client.onMessageArrived = function(message) {
    // my stuff
}
client.connect({
    onSuccess: function() {
        client.subscribe("myRootTopic/#");
    }
});
```
DEMO!
Data Analytics for IoT?
Apache Spark Streaming

- Stream processing in Java, Python & Scala
- Built-in connectors for Kafka, Twitter, ZeroMQ, Flume, Kinesis & ... MQTT!
- A nice programming model for consolidating time-series data
- Awesome combo when used with Spark MLlib!
DEMO!
Eclipse IoT is also...

**Industrial IoT**

- Open source implementations of IEC standards
- Eclipse SCADA, 4DIAC, Rise V2G, ...
Eclipse IoT is also...

Device Management

- **LWM2M** is an Open Mobile Alliance Standard
- Device Management on top of CoAP
- Eclipse [Leshan](#) and [Wakaama](#) are two implementations
Eclipse IoT is also...

Secured Service Discovery

- Eclipse Tiaki
- Leveraging DNS-SEC and DNS-SD for retrieving a device configuration parameter, or its public key for establishing secured communications
Eclipse IoT is also...

- Flexible Framework
- Based on Java and OSGi
- Huge number of “bindings”: KNX, Nest, Philips HUE, …
If you had to remember only 3 things...

#1

Kura is awesome!
Go download it now!

http://eclipse.org/kura
If you had to remember only 3 things...

#2

Build your own greenhouse & follow the tutorial

http://iot.eclipse.org/java/tutorial
If you had to remember only 3 things...

#3

Eclipse IoT is much more than Kura and Java!

http://iot.eclipse.org/
Get Involved!

- Open bugs / fix bugs
- Request new features
- Write articles, tutorials
- Participate on the mailing lists
- Propose your project!
One more thing...
One more thing...

Apply before November 23!

http://iot.eclipse.org/open-iot-challenge
Thank you! Questions?

benjamin@eclipse.org
@kartben
http://blog.benjamin-cabe.com

http://iot.eclipse.org