SilverWare
Command your fleet of Internet of Things!

Internet of Things: The complete solution

Miroslav Jaroš
www.silverspoon.io
www.silverware.io
A Short word on IoT

IoT Architecture

Bulldog Library
  ○ GPIO, I²C, SPI, ...
  ○ BeagleBoneBlack, CubieBoard, Raspberry Pi

Silverspoon
  ○ Apache Camel

SilverWare
  ○ Microservices
A Short word on IoT

In 2020 50 billion devices connected to the Internet [Cisco].

- cars, wearables, sensors, actuators, appliances, ...

The Internet of Things will generate a staggering 400 zettabytes (ZB) of data a year by 2018 [Cisco].

Collected data are processed to build a better world

- smart devices, smart cities, ...
IoT Architecture

- Ingestion
- Analytics
- Gateways
- Configuration
- Control Messages

- Sensors
- Actuators
- Dashboards
- Workflow
- Integration

www.silverspoon.io
www.silverware.io
Java GPIO library

- Gather sensor data, control actuators, ...
- Supports interfaces like I2C, SPI, PWM, ...
- Write once, run elsewhere: BeagleBoneBlack, CubieBoard, Raspberry Pi

```java
//Detect the board we are running on
Board board = Platform.createBoard();

//Set up a digital output
DigitalOutput output = board.getPin("P1_11").as(DigitalOutput.class);

// Blink the LED
output.high();
```
Apache Camel

Apache Camel is an open source Java framework that focuses on making integration easier and more accessible to developers.

- concrete implementations of Enterprise Integration Patterns (EIPs)
- connectivity to a great variety of transports and APIs
- easy to use Domain Specific Languages (DSLs) to wire EIPs and transports together
- Embeddable in your JVM-based application

Components: MQTT, JMS, File, HTTP, AMQP, SMTP, POP3, HL7, HipChat

http://camel.apache.org/components.html
Portable IoT Platform

- wraps Bulldog into an Apache Camel component
- easily integration with external systems and applications using EIP
- provides Maven Archetypes for rapid development

```xml
<camelContext trace="false" xmlns="http://camel.apache.org/schema/spring">
  <restConfiguration bindingMode="auto" component="jetty" port="8080" />
  <rest path="/rest">
    <post uri="/led">
      <to uri="bulldog://gpio?pin=P1_11" />
    </post>
  </rest>
</camelContext>
```
Brief introduction to Microservices

Monoliths and Microservices:

A monolithic application puts all its functionality into a single process...

... and scales by replicating the monolith on multiple servers

A microservices architecture puts each element of functionality into a separate service...

... and scales by distributing these services across servers, replicating as needed.

"Do one thing, and do it right."

Image taken from http://martinfowler.com/articles/microservices.html
Minimalistic and easy to use Microservices implementation.

SilverWare builds on top of existing technologies and frameworks you are familiar with (REST, CDI (Weld), Camel, Vert.x, NoSQL, ActiveMQ, Amazon SQS, Hystrix...).

```java
@Microservice
class HelloMicroservice implements HelloService {
    public String greet(String whom) {
        return "Hello " + whom;
    }
}
```
An award-winning PaaS.

OpenShift Enterprise 3 is built around a core of application containers powered by Docker, with orchestration and management provided by Kubernetes, on a foundation of Red Hat Enterprise Linux.

OpenShift provides a wide range of languages and services, deployed as containers (Node.js, Python, Ruby, JBoss EAP, …)

https://openshift.redhat.com
Thank you for your attention!

mjaros@silverware.io

https://github.com/px3

https://github.com/px3/silverspoon-demos

https://github.com/px3/SilverWare-Demos

https://twitter.com/silverwareio

https://dzone.com/articles/doing-iot-with-bulldog-library