# Make IoT Child‘s play

Gamifying IoT with Vorto & Kura

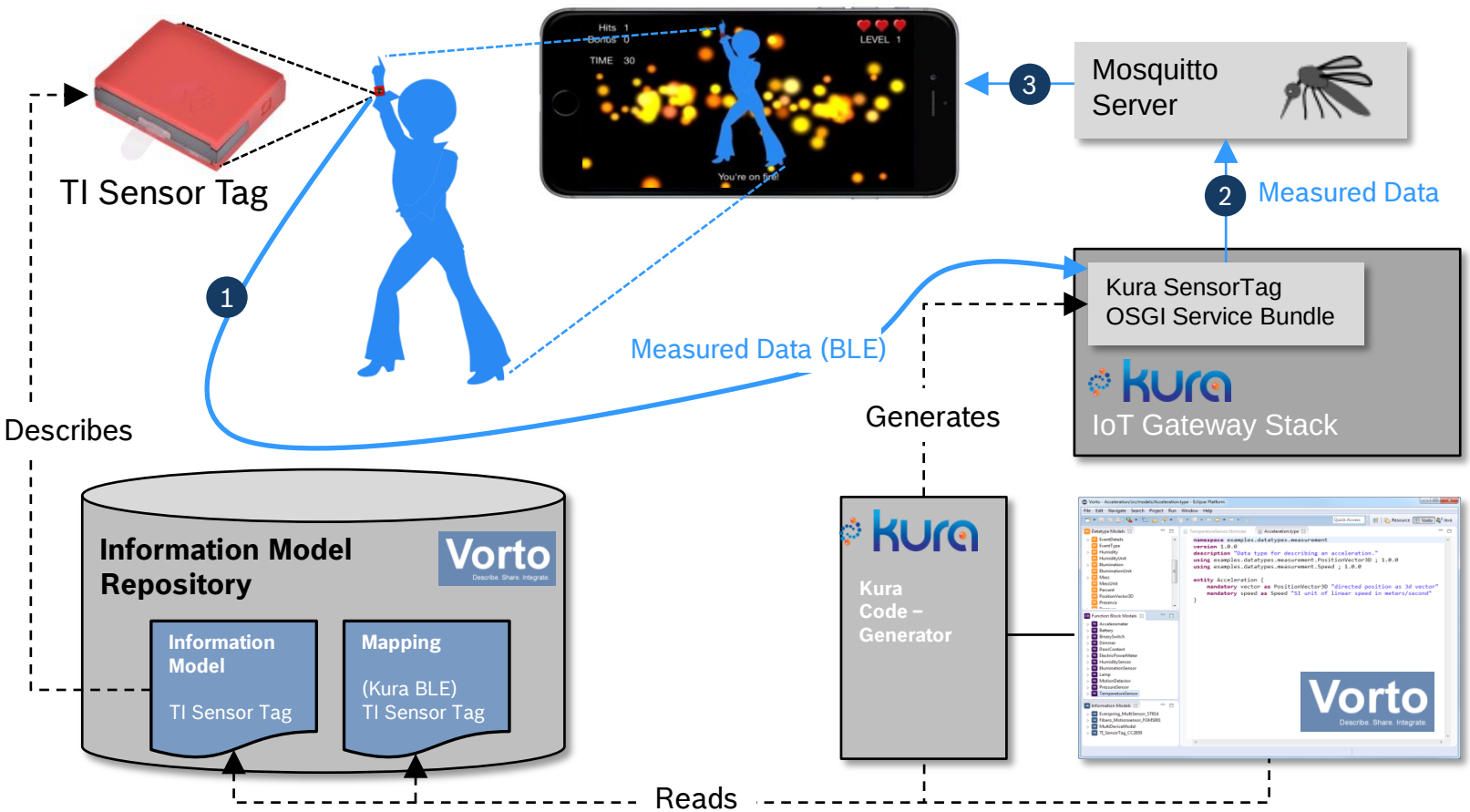
Alexander Edelmann, Vorto Committer

Luca Dazi, Kura Committer

kura

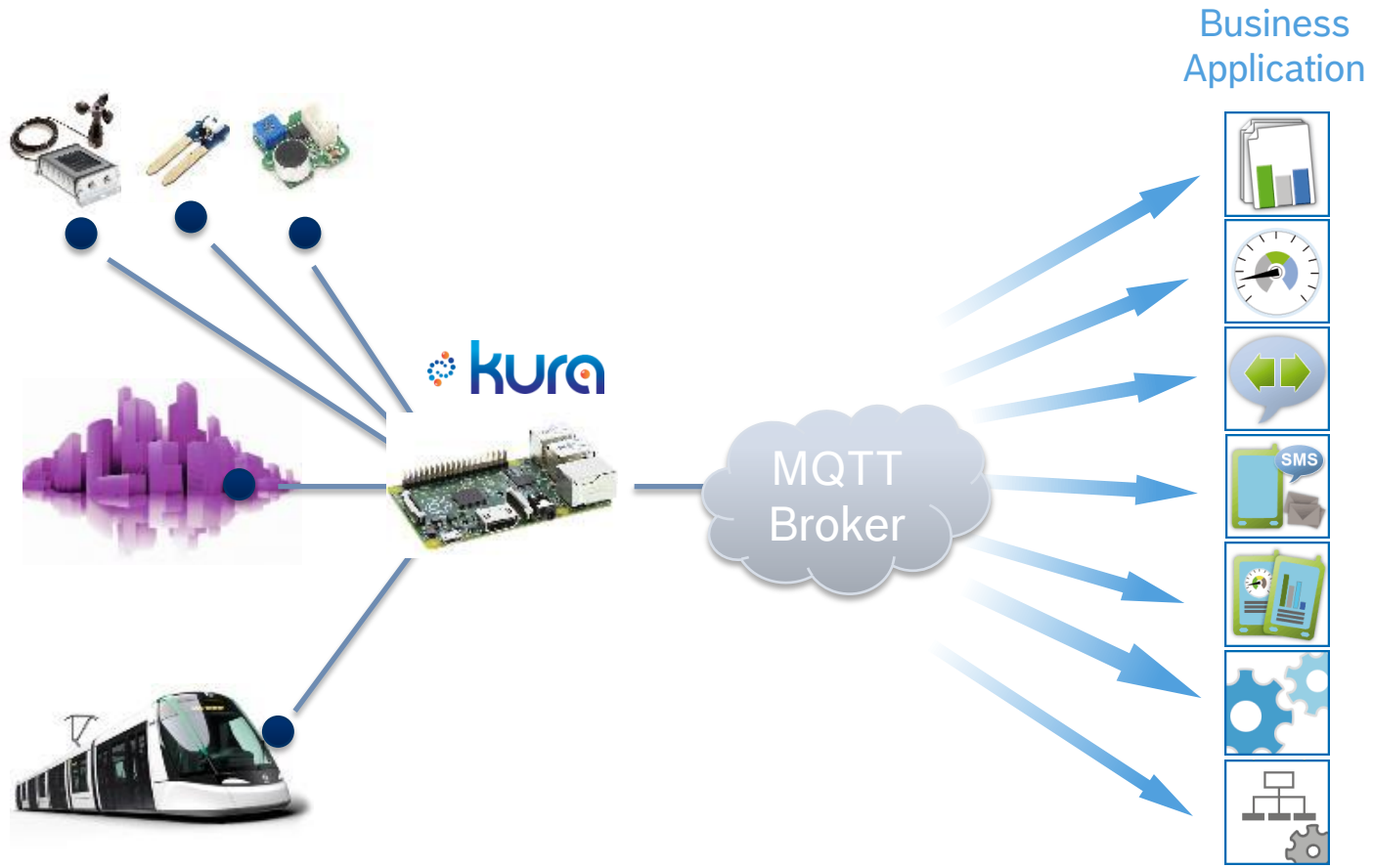
Vorto
Describe. Share. Integrate.

# IoT Fever Game with Vorto & Kura

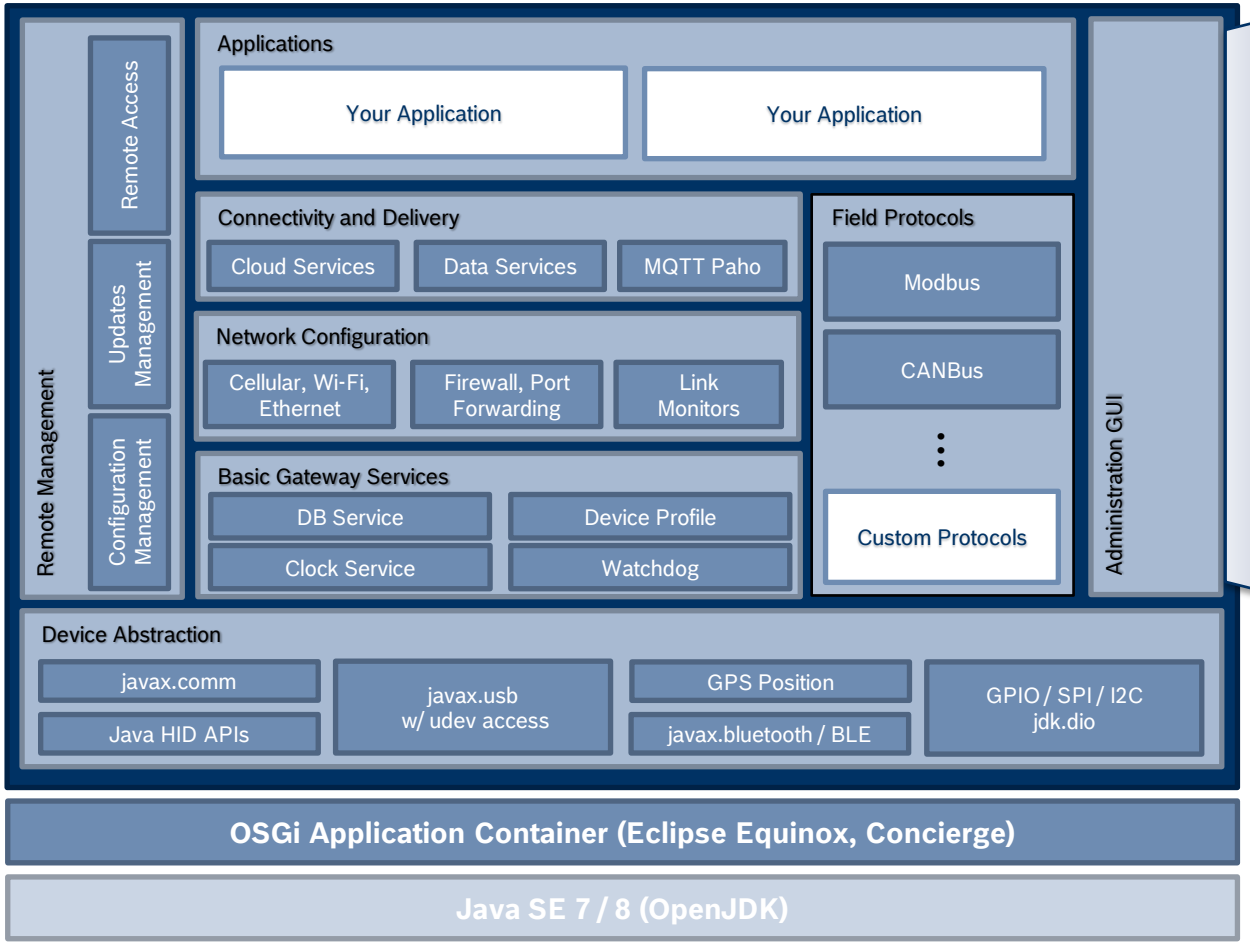


# IoT Gateways

## Revolution: Towards Real-time Actionable Data



# Eclipse Open IoT Stack for Java



# Using Services

Acquire the relevant Declarative Service

```
<reference name="GPIOService"
           policy="static"
           bind="setGPIOService"
           unbind="unsetGPIOService"
           cardinality="1..1"
           interface="org.eclipse.kura.gpio.GPIOService"/>
```

Define Service members

```
public void setGPIOService(GPIOService gpioService) {
    m_GPIOService = gpioService;
}

public void unsetGPIOService(GPIOService gpioService) {
    m_GPIOService = null;
}
```

Define Service hooks methods

```
protected void activate(ComponentContext componentContext,
        Map<String, Object> properties) {
    s_logger.info("Activating Shoot A Pi...");

    try {
        m_CloudClient = m_CloudService.newCloudClient(APP_ID);
        m_CloudClient.addCloudClientListener(this);
    } catch (KuraException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    doUpdate();
}
```

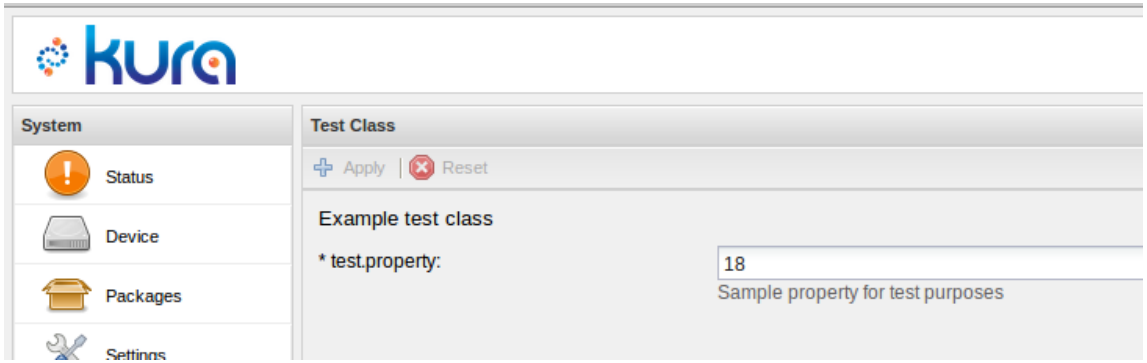
Use the Service

```
// Publish the message
try {
    int messageId = m_CloudClient.publish(topic, payload, qos, retain);
    s_logger.info("Published to {} message: {} with ID: {}", new Object[] { topic, payload
} catch (Exception e) {
    s_logger.error("Cannot publish topic: " + topic, e);
}
```

# OSGi Metatype Definition

Define a metatype for Application-specific configurable data...

```
 name="Test Class"
 description="Example test class">

<!--
<Icon resource="" size="32"/>
-->

<AD id="test.property"
    name="test.property"
    type="Integer"
    cardinality="0"
    required="true"
    default="18"
    description="Sample property for test purposes"/>
```




And manage those data with the WebUI (Or remotely...)

**Information model –** Abstract representation of functionalities and properties of the TI SensorTag

Functionalities

- Accelerometer Sensor
- Temperature Sensor
- Humidity Sensor
- Pressure Sensor
- etc.

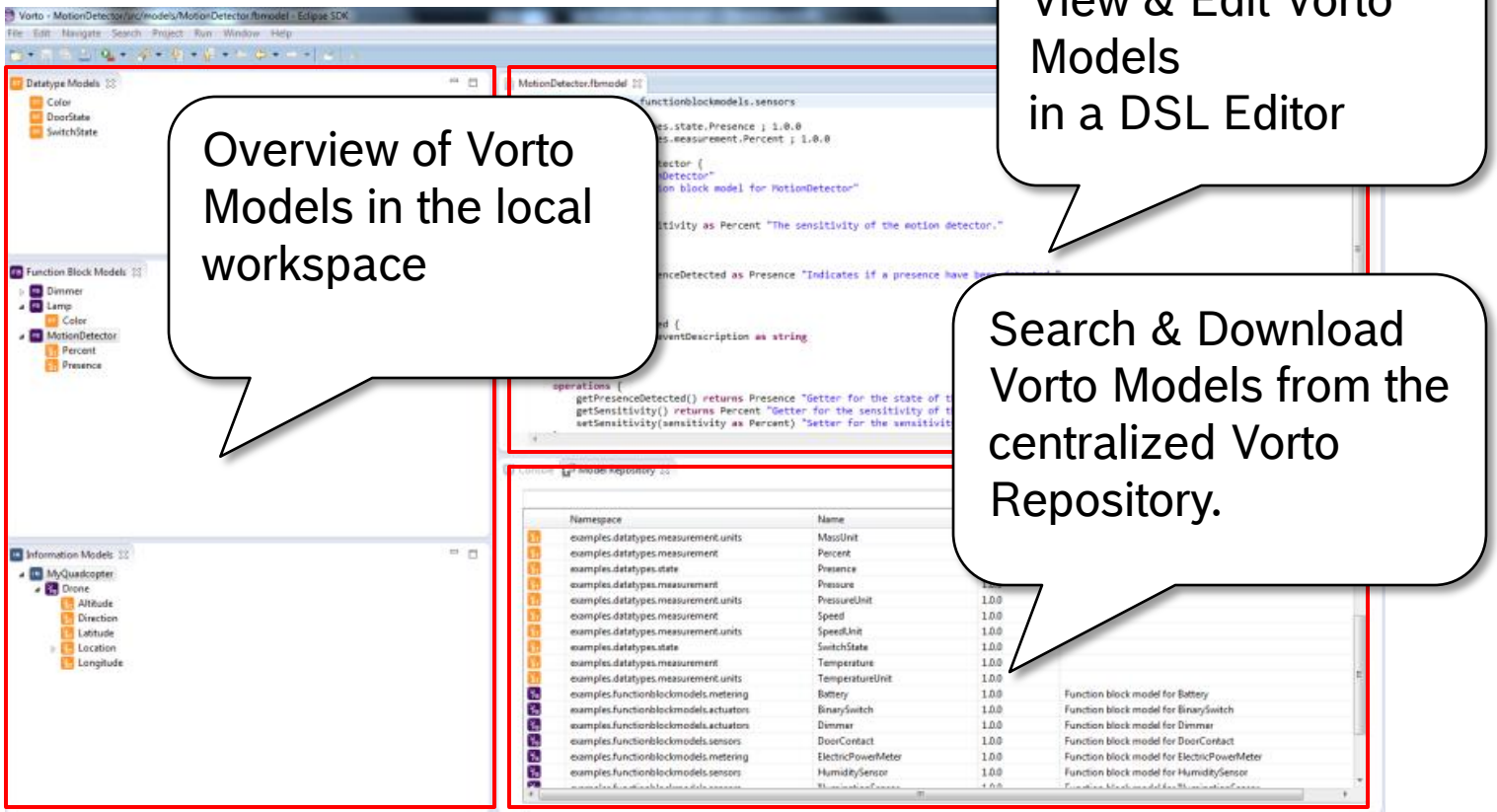
Mapping

**TI Sensor Tag Information Model**

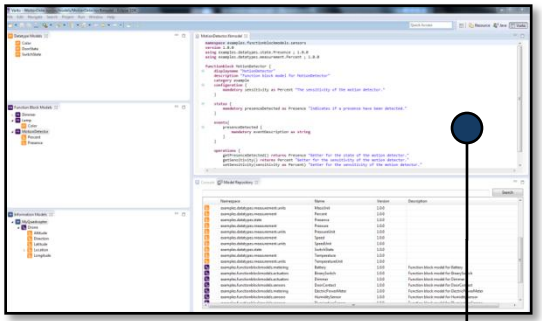
- Functionblock: Accelerometer Sensor
- Functionblock: Temperature Sensor
- Functionblock: Humidity Sensor
- Functionblock: Pressure Sensor
- ...
- Device specific information

Information Models are created using the **IoT Toolset**.

**The IoT Tool Set –** realized as Eclipse plugins and provides a textual DSL editor



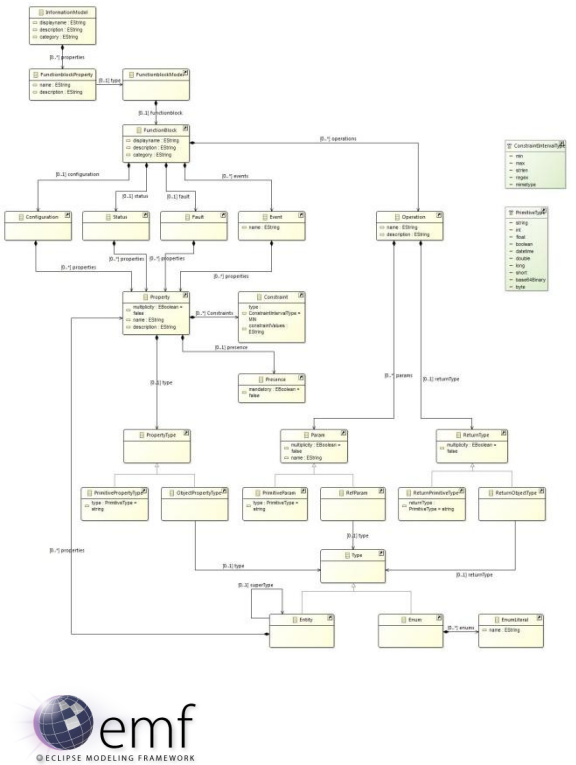
# **The Vorto DSL:** An easy way to describe a device



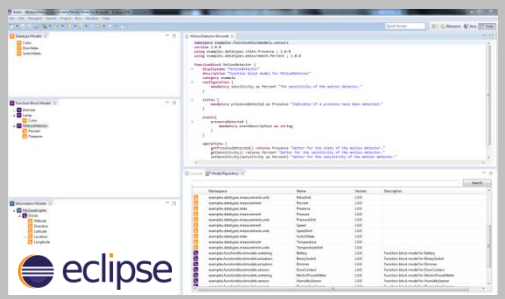
Complies to

```
namespace examples.informationmodels.sensors
version 1.0.0
displayname "TI SensorTag CC2650"
description "Information model for the TI SensorTag CC2650."
category demo
using examples.functionblockmodels.sensors.Accelerometer ; 1.0.0
using examples.functionblockmodels.sensors.TemperatureSensor ; 1.0.0
using examples.functionblockmodels.sensors.HumiditySensor ; 1.0.0
using examples.functionblockmodels.sensors.PressureSensor ; 1.0.0
using examples.functionblockmodels.Microphone ; 1.0.0
using examples.functionblockmodels.Magnetometer ; 1.0.0
using examples.functionblockmodels.sensors.MagneticSensor ; 1.0.0
using examples.functionblockmodels.Light ; 1.0.0
using examples.functionblockmodels.Gyroscope ; 1.0.0

infomodel TI_SensorTag_CC2650 {

    functionblocks {
        accelerometer as Accelerometer "Function block representing the accelerometer of the device."
        temperaturesensor as TemperatureSensor "Function block representing the temperature sensor of the device."
        humiditysensor as HumiditySensor "Function block representing the humidity sensor of the device."
        pressuresensor as PressureSensor "Function block representing the pressure sensor of the device."
        microphone as Microphone "Function block representing the digital microphone of the device."
        magnetometer as Magnetometer "Function block representing the magnetometer of the device."
        magneticsensor as MagneticSensor "Function block representing the magnetic sensor of the device."
        light as Light "Function block representing the LED light of the device."
        gyroscope as Gyroscope "Function block representing the gyroscope of the device."
    }
}
```

## Vorto Meta Model



# **Vorto Model Repository –** Manage & Share Information Models in a centralized repository.




Vorto IoT Tool Set

REST

*Search*
*Download*
*Upload*

REST

3rd Party Tool / App

**Vorto Repository** Upload Model Learn more

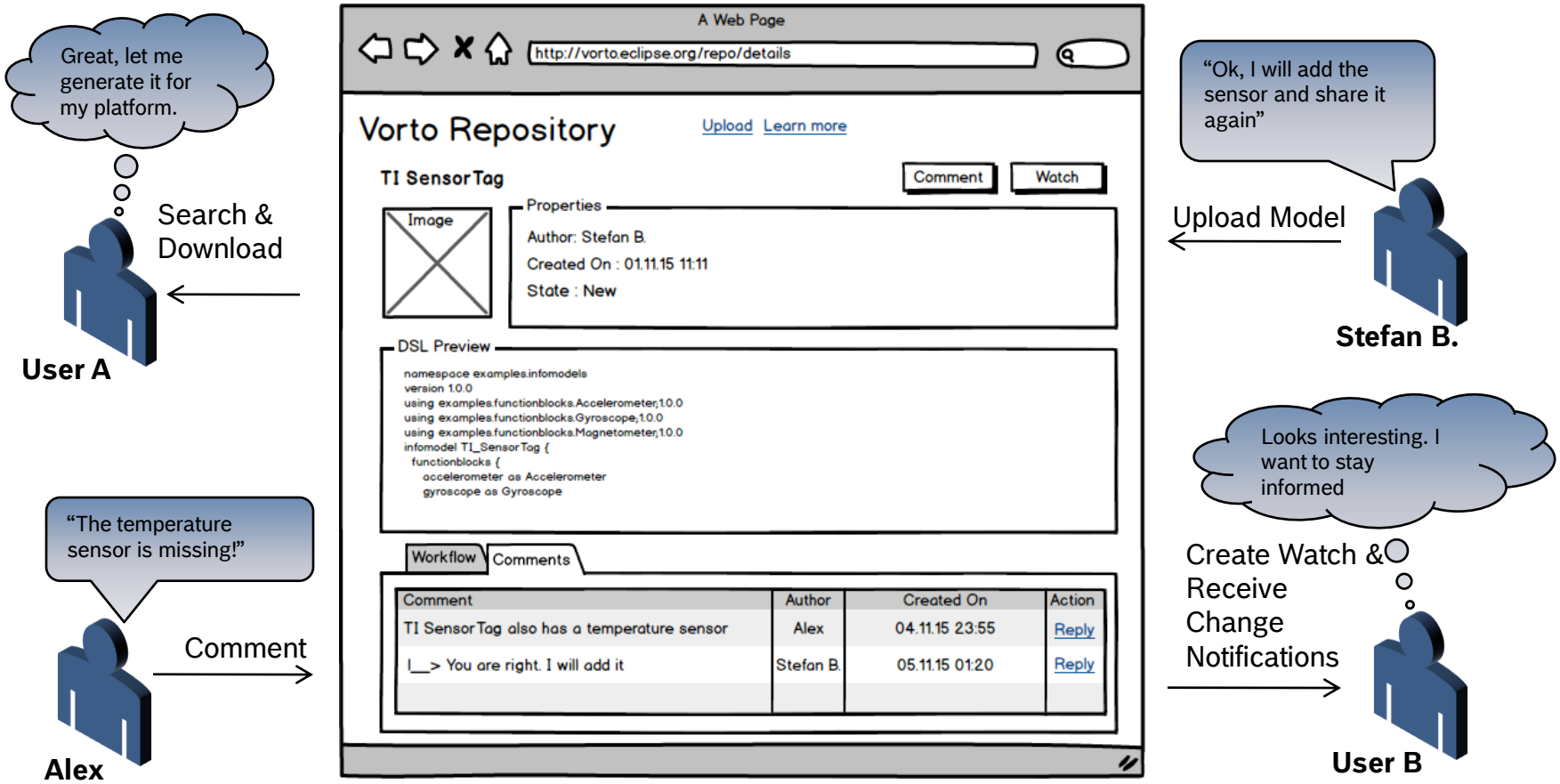
All | Information Model | Function Block | Data Type

* Search Clear

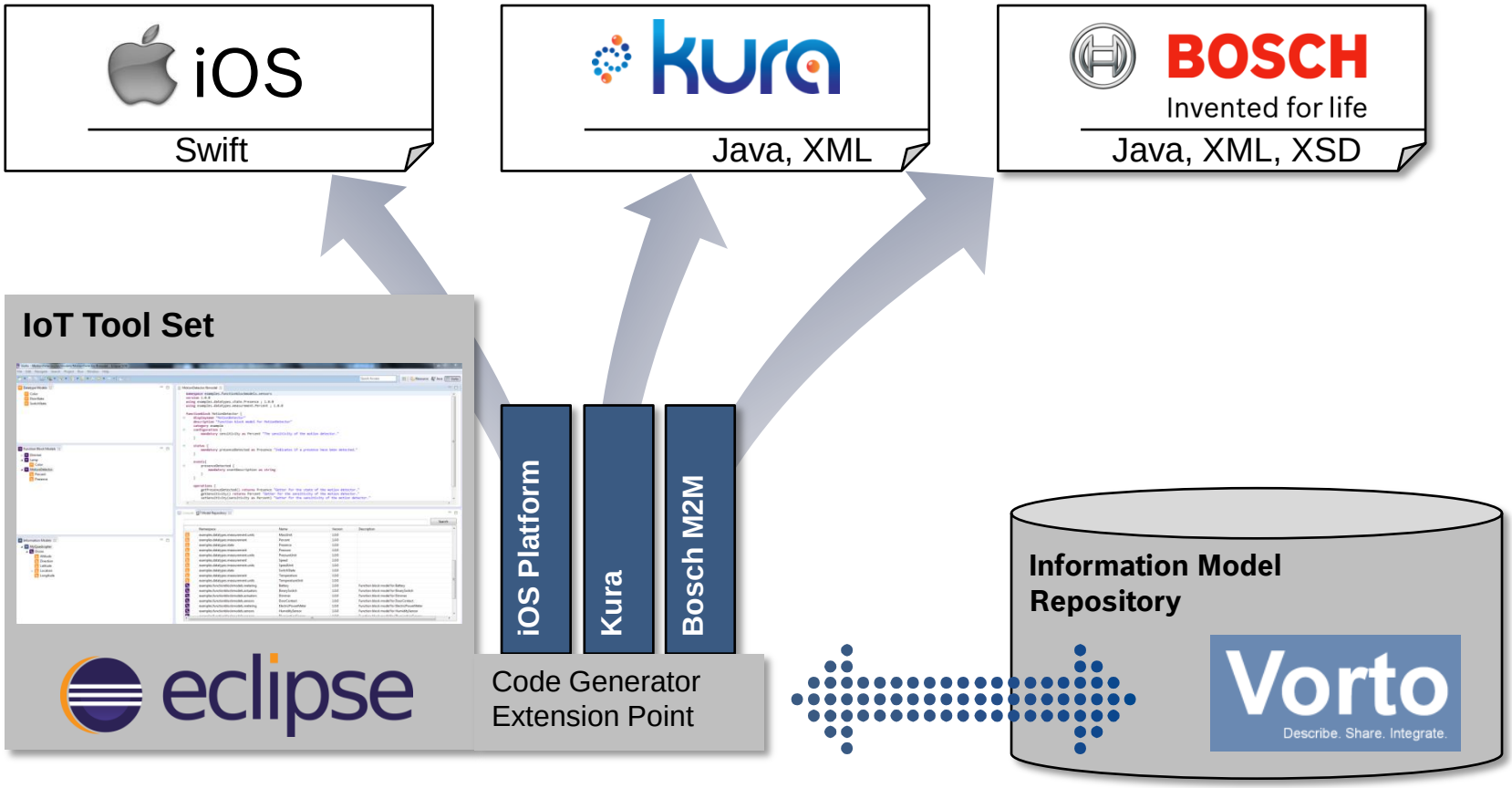
| Type | Namespace | Name | Version | Description | Action |
|---|---|---|---|---|---|
| FB | examples.functionblockmodels.sensors | DoorContact | 1.0.0 | Function block model for DoorContact | Details |
| FB | examples.functionblockmodels.actuators | Dimmer | 1.0.0 | Function block model for Dimmer | Details |
| FB | examples.functionblockmodels.actuators | BinarySwitch | 1.0.0 | Function block model for BinarySwitch | Details |
| FB | examples.functionblockmodels.metering | Battery | 1.0.0 | Function block model for Battery | Details |
| DT | examples.datatypes.measurement | Duration | 1.0.0 | | Details |
| DT | examples.datatypes.measurement | Percent | 1.0.0 | | Details |
| DT | examples.datatypes.state | DoorState | 1.0.0 | | Details |
| DT | examples.datatypes.state | SwitchState | 1.0.0 | | Details |
| DT | examples.datatypes.measurement.units | DurationUnit | 1.0.0 | | Details |
| DT | examples.datatypes.measurement.units | ElectricPowerUnit | 1.0.0 | | Details |

vorto.eclipse.org/repo

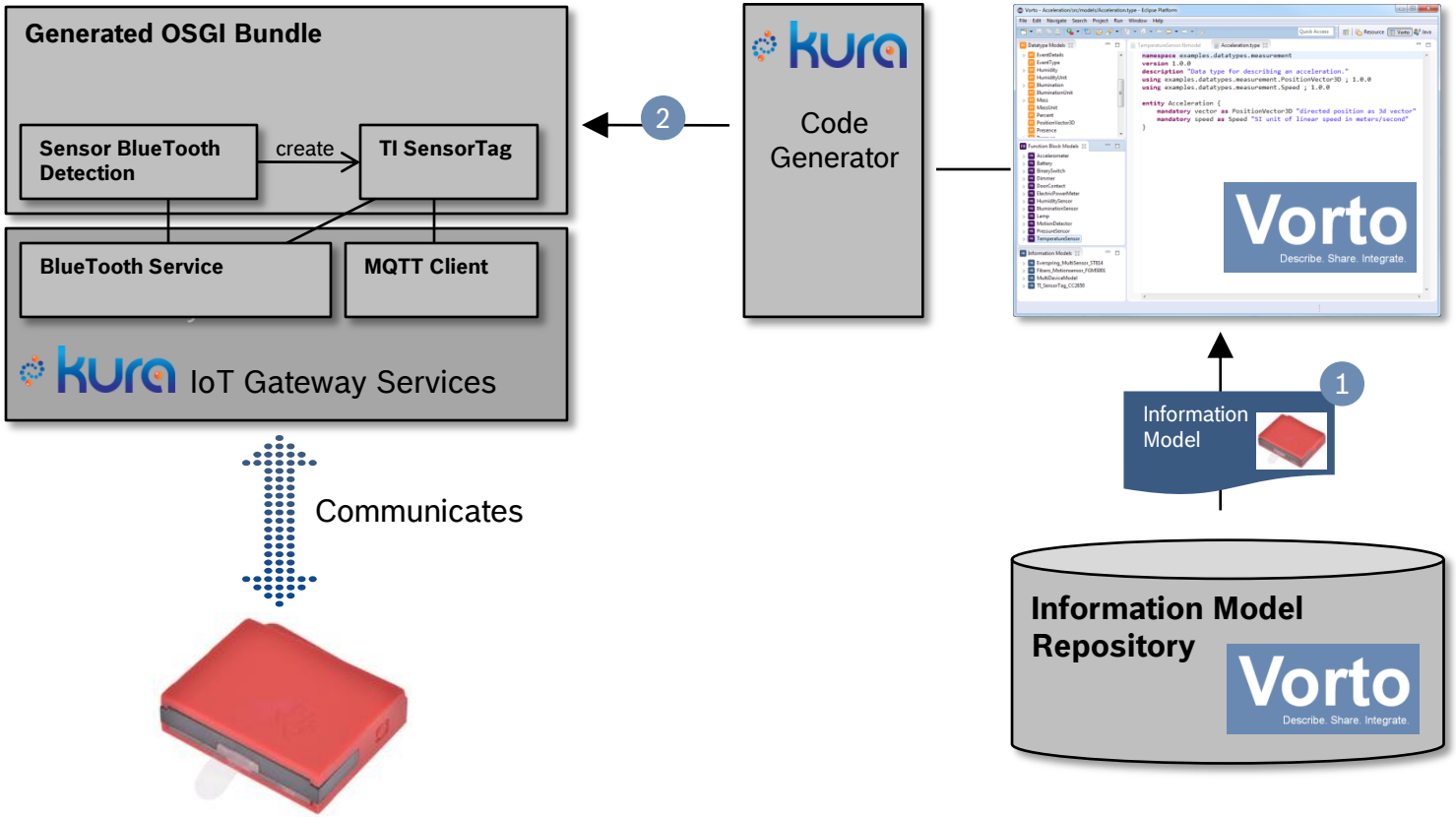
# A single platform for different stakeholders



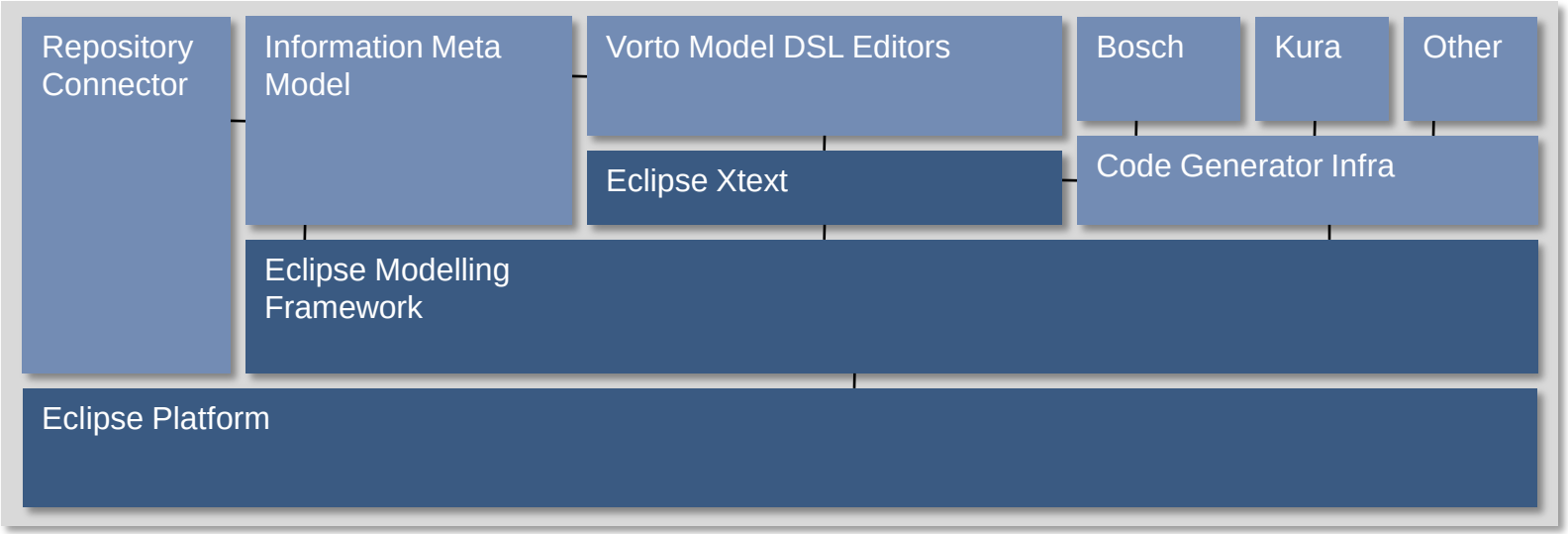
**Information Models** can be transformed in various representations. Example Code Generators are:



# **Code Generation:** Kura TI Sensor Tag OSGI Bundle



# Vorto Toolset Architecture



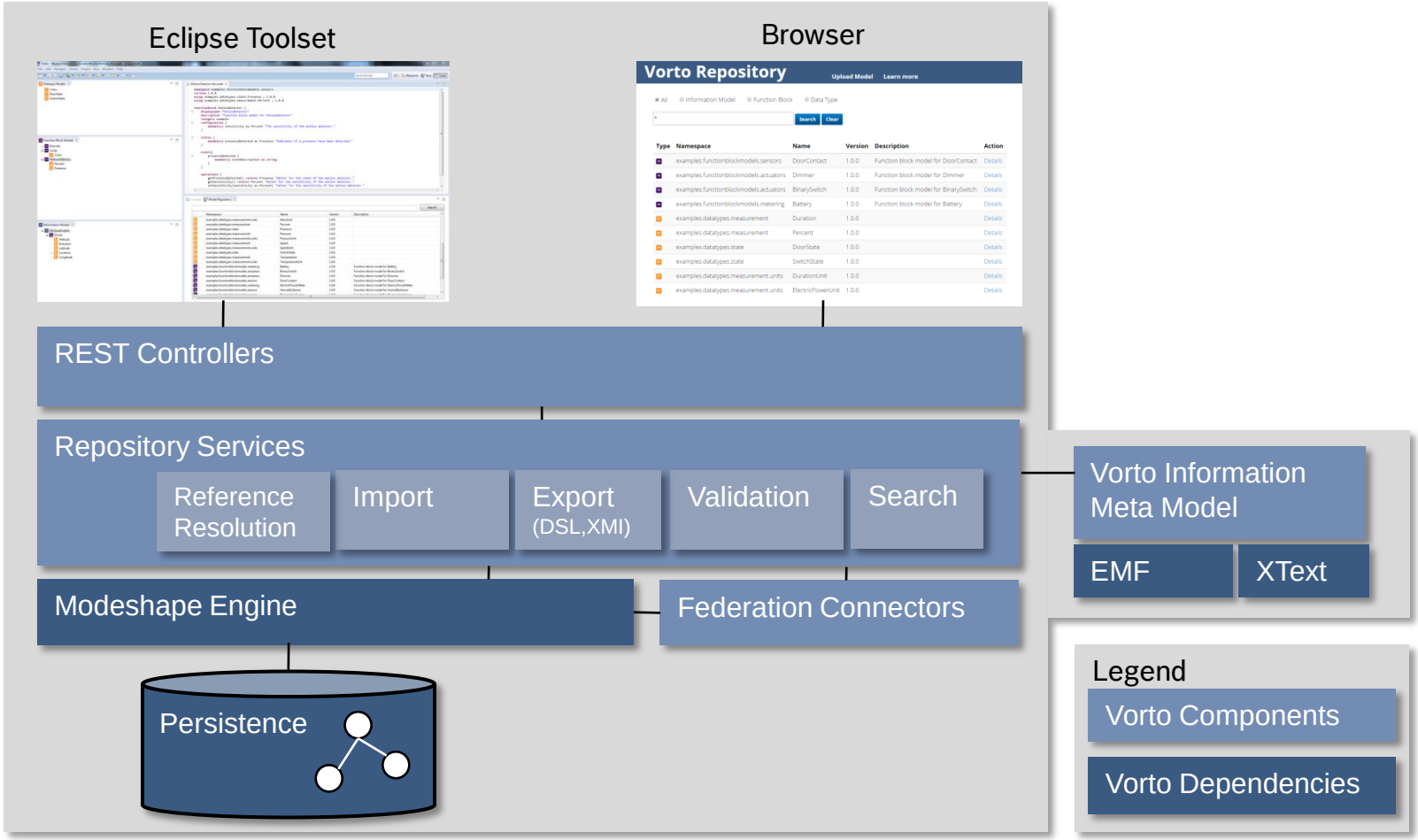

Source Code available under
https://github.com/eclipse/vorto

Legend

Vorto Components

Vorto Dependencies

# Vorto Repository Architecture



# Putting it all together in a demo…