KURA an OSGi-based Application Framework
for Multi-Services Gateways

Introduction & Technical Overview

Pierre Pitiot

Grenoble 19 février 2014
Multi-Service Gateway Approach
ESF / Increasing Value / Minimizing TCO
The M2M Integration Platform
M2M Distributed Systems

Business Applications

M2M Integration Platform

Multi-Service Gateway

Integrate
+ Act
+ Connect
+ Collect & Integrate

M2M Protocol

Standard Interfaces

@ Standard Interfaces

Integrate

Collect & Integrate

EUROTECH
The M2M Integration Platform

The Basic Functionality

- Simple integration into business applications
- REST APIs
- MQTT/HTTP Bridge
- Real-Time APIs
- Google Charts Visualizations
- Data Exports

- Real-Time Analytics
- Advanced Data Patterns
- Event processing
- Alerts & Actions

- Historical Data
- Aggregated Queries
- Redundant Elastic Storage

- Open Transport & Payload
- Device Initiated
- Bandwidth Optimized
- Publish & Subscribe
- Secure & Dedicated

- Java Application Framework
- Modular Protocol Support
- Device Management
- Hardware Abstraction
- Driver Hardware Support
The M2M Integration Platform
Implementing a full “M2M Software Stack”

- Business Applications
- M2M Integration Platform
- Multi-Service Gateway

API's & Integrations
Data Management Layer
Data Store Layer
Communication / Broker
M2M Protocol
M2M Platform Client
Application Software Framework
Operating System

Integrate
Act
Store
Connect
Collect & Integrate
Connect

Communicate

• Open Standard Protocol with Open Standard Payload Format
• Device-initiated connection
  – No need to open ports into your network
  – Based upon TCP/IP
  – Zero-Configuration

• Persistent Socket
  – Extremely Low latency, high real-time
  – Light-weight asset tracking through built-in Device Connection monitoring

• Bandwidth and Retries
  – 2-bytes overhead per package
  – QoS levels
  – Data Agnostic
  – Recommended EDC Payload with efficient serialization
Connect

Manage
• Publish and subscribe
  – Decoupling the data producers from application consumers
  – Application does not need to know detailed device topology
  – Hierarchical Namespace
  – One-to-many message distribution
• Secure
  – SSL with Trusted Authority Certificates
  – Authenticated access
  – Two firewalls
• Dedicated
  – One dedicated Broker instance per tenant
• Device Application Development (ESF)
  – Modular, 100% Java, OSGi-based Application framework
  – Flexible Data Publishing Policies
  – Local Data Persistence to survive network drops
  – Simple C++/Java/... clients
Store

- Storage is realized using a combination of SQL for device/user management, dataplan management and NoSQL databases for device data
- Flexible:
  - Aggregates data along the hierarchical namespace
  - Allows queries by time ranges, topic, device, metrics
  - Can store binary data
- Secure and Available
  - guarantees tenant isolation at the storage level
  - Highly available with built-in failover mechanism
  - Database redundant and replicated across data centers
  - NoSQL database is optimized for large volume inserts
Act
Real-time Analytics

- Dedicated Complex Event Processing Engine
- Declarative rules coded SQL-like language
- Can be created/updated at any point in time
- Complex Statistical Operations available (min, max, avg, stddev, vwap...)
- Continuous queries applicable to single data points, moving data windows (by time or numbers) on a single device or on an aggregate of the devices
- Parametrizable Event Triggering
  - Alerts: SMS, Email, Twitter
  - Application Integration: MQTT Messages towards devices, REST Calls towards IT Applications
The M2M Integration Platform
Implementing a full “M2M Software Stack”
Integrate

- Easy to use REST APIs for application integration
- Standard WADL resource file for API documentation
- Bidirectional MQTT to HTTP bridge to control devices through the REST APIs
- Asynchronous REST/Comet APIs for HTTP-based subscriptions on real-time streams of data
- XML and JSON payload formats
- Java and C++ ready-to-use libraries
- Rich set of publicly available examples and documentation
- Native support for Google Chart Tools to easily visualize acquired data into rich, live dashboards
ESF Functional Blocks

Device Abstraction

Ready to use OSGi bundles exposing Java APIs for device access and with the native code dependencies pre-compiled.
ESF Functional Blocks
Gateway Basic Services

Configurable OSGi Services available to the applications to interact with the basic gateway functionality.
ESF Functional Blocks
Network Management

Configurable OSGi Services to access the current network configuration and administer it. It interacts with the Linux system to configure network interfaces including WiFi access points and PPP connections.
Configurable OSGi Services which greatly simplify the development of telemetry M2M applications interacting with a remote Cloud server.
ESF Functional Blocks
Operation & Management

- Remote Configuration Management
- Remote Software Updates
- Remote System Command
- Remote Log Retrieval
- Remote VPN Access

Applications
Connectivity and Delivery
Network Configuration
Gateway Basic Services
Device Abstraction
OSGi Application Container
Java VM
ESF Functional Blocks
Field Protocols

- ModBUS
- CANBus
- Vending
- Fitness
- People Counter
- Remote Serial Ports
- Custom Protocol
ESF Functional Blocks
Administration GUI

- Administration GUI
- Local SW Update
- Device Configuration
- Dynamic Service Configuration
- Java VM
- OSGi Application Container
- Gateway Basic Services
- Network Configuration
- Connectivity and Delivery
- Field Protocols
- Applications
## ESF Developer’s Experience

**Designed from ground-up for developers**

<table>
<thead>
<tr>
<th>Emulate on PC</th>
<th>Deploy on Target</th>
<th>Cloud Managed</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Laptop" /></td>
<td><img src="image2.png" alt="Laptop to Gateway" /></td>
<td><img src="image3.png" alt="Cloud" /></td>
</tr>
<tr>
<td>Start developing your M2M application in the comfort of your PC.</td>
<td>When you are ready, deploy your application on the gateway.</td>
<td>Provision your application to field devices from the Cloud.</td>
</tr>
<tr>
<td>• Full Eclipse Integration</td>
<td>• One-click Deployment</td>
<td>Manage your application configuration and lifecycle from a Cloud infrastructure. No more field visits!</td>
</tr>
<tr>
<td>• Target Platform Definition</td>
<td>• Eclipse Plugin</td>
<td>• Web-based Console</td>
</tr>
<tr>
<td>• Emulated Services</td>
<td>• Remote Debugging</td>
<td>• REST API Integration</td>
</tr>
<tr>
<td>• Run/Debug from Eclipse</td>
<td></td>
<td>• Smart Alerts</td>
</tr>
<tr>
<td>• Support Mac/Linux Hosts</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Thank You!