MIDDLEWARE IMPLEMENTATIONS

Sakthi Ramanathan Sivaraman
Topics to be discussed

- Application Level Middleware
  - Implementation in IFTTT, ROS frameworks
- Middleware for data & network management
  - Implementation in Oracle Fusion Middleware
- Device Level Middleware
  - Implementation in openHAB (eclipse smarthome) & FPGAs
IoT Challenges solved by Middleware

*Interoperation*: basic connectivity between physical devices to higher layers

*Context detection*: Context data retrieval from sensor device data for decision making

*Device discovery and management*: Neighbor devices detection in the network

*Security and privacy*: Basic necessity in industries like health, finance and critical infrastructure

*Managing Large Data volume*: Middleware is required to efficiently find, fetch and transfer this data
API Management – Application Level middleware

- Fundamental enabler of performance & scalability for the Internet of Things
- APIs allow for faster integration, which allows to take advantage of the mass proliferation of “intelligent” devices
- Leverage the relationships between connected device to automate tasks
- APIs help companies to realize true value for the IoT by focusing on the value-added services on data derived from IoT, by combining this data with their legacy systems and business assets
IFTTT (If This Then That)

- Acts as an **intelligence** layer for the internet of things.
- Automates tasks via control and connection between devices or web applications, leveraging the context dependence between them.
- Makes the things “Smarter”, by connecting them seamlessly at the application level.
- Can build custom recipes made from “triggers” (this in IFTTT) and “actions” (that).
- UI & Programming for the **future**.

[Images of various if-then scenarios]

- Every day at __ : __ turn the lights on
- When your Nest is set to away, your lights will turn off
- Receive an emergency call if smoke is detected
- At sunset, turn on your lights
- Turn your lights on when you arrive home

[Link to recipes page: https://ifttt.com/recipes]
ROS (Robot Operating System)

- A collection of software libraries and tools for complex robot software development (developed by Willow Garage)
- Provides hardware abstraction, device drivers, libraries, visualizers, message-passing, packet management, and more
- Completely open source and reusable software
- ROS-Industrial is an open-source project that extends the advanced capabilities of ROS to manufacturing

http://wiki.ros.org/
ROS (Robot Operating System)

- Communication between multiple machines to a single master & each other using a stateless HTTP based protocol
- "Node" – an executable that uses ROS to communicate with other nodes
- "Topics" – named buses over which nodes exchange messages (TCPROS)
- "Messages" – A simple datastructure comprising typed fields (along with header info)

Communication of messages over topics happens between publisher (talker) & subscriber (listener) nodes.
iRobot Create example

- **cmvision node**: I will receive images on topic "image" and publish blobs on topic "blobs".
- **control node**: I will receive blobs on topic "blobs" and publish velocities on topic "cmd_vel".
- **camera node**: I will publish images on topic "image".
- **create node**: I will receive velocities on topic "cmd_vel".

![Diagram of iRobot Create setup with nodes and connections](image-url)
Middleware for Data & Network Management

1. Equipment sends temperature data at regular intervals to gateways
2. Gateway becomes smart with local analytics to analyze temperature data and identify fluctuations.
3. Send breach alerts and temperature readings in real-time.
4. Secure Gateway and Identity Management to secure services and manage access.
5. Event processing identifies sustained threshold breaches and raises critical alerts.
6. Event-driven process gets triggered to resolve the issue.
7. Technician
   - Analyzes equipment service history using BI Analytics on his mobile app
   - Performs remote troubleshooting
   - Orders part replacement
   - Based on order cost, process update ERP

Middleware for Data & Network Management

- **Real-time Analysis**
  - Large amounts of data, alerts
  - Information in different formats
  - Various array of devices, sensors and other machines

- **Integration**
  - Smooth orchestration and manageability of disparate systems
  - Embedding intelligence by way of real-time data gathering from gateways and devices through business processes

- **Security**
  - Access control and data authentication with privacy of users (or devices)
  - Resilient to attacks & with high degree of reliability

- **Monitoring**
  - User interaction
  - Critical in fast decision making
1. Real-time Analysis

- **Complex Event Stream Processing**: (with Oracle event processing)
  - Processes data stream across many disparate event sources or event sinks
  - Ability to process 1M events per second

- **Scalability with Oracle Coherence & Oracle Exalogic**:
  - handle storage of data in memory
  - very high throughput & low latency on a single Oracle Exalogic compute node

- **Continuous Query Language (CQL)**: (for continuous query execution)
  - SQL + constructs that support streaming data
  - Local decision and filtering done by Oracle Event Processing for Java Embedded

CQL query to detect temperature alerts and filter data looks like this:

```sql
SELECT sensorID, status, temperature from heatInputChannel[now]
WHERE temperature < temp_low or temperature > temp_high
```
Oracle Fusion Middleware

2. Integration

- **Oracle BPM**: (Business Process Management)
  - integrates devices, applications and human intervention
  - coordinates responses to events received from IoT

- **Oracle SOA**: (uses Shared Services infrastructure)
  - for scalability of IoT/enterprise applications.
  - to correlate and make sense of various event patterns

- **Oracle Service Bus**:
  - intermediary layer between devices and backend apps.
  - integrates new devices & services (true plug & play)
  - provides a layer of abstraction to virtualize services
  - supports HTTP/REST, HTTP/SOAP, WS-I & JMS
Oracle Fusion Middleware


- **Oracle API Gateway**:  
  - a lightweight interface between OAM & client services  
  - secures & manages APIs mediating traffic to enterprise systems

- **Oracle Access Manager (OAM)**:  
  - authenticates and authorizes a user, mobile device or service.  
  - after authentication, it returns an access token back to the app

4. Monitoring

- **Oracle DF Mobile**:  
  - Helps build mobile apps to visualize sensor information and control gateways/actuators remotely  
  - Integrates with Oracle IAM solution to ensure authorized and authenticated access to data & insights from devices
Oracle Fusion Middleware

SFPark Architecture

Oracle Service Bus

Operation Data Store

Meter Enforcement Summary

Data Warehouse

Oracle Business Intelligence (OBIEE) for Pricing Analytics

Parking Enforcement Handheld (Enforcement Webpage)

Web Services for Parking Availability

Text Messaging Service

SFMTA Variable Message Signs

SFMTA Website

Financial Information

Garage Pricing & Inventory Data

On-street Pricing & Inventory

Parking Meters

Garage Real-time Occupancy Data

Other Sensor Vendors
Intranet of Things – Device Level Middleware

To get the full control of our devices and our data. (and not give the control to cloud services)

Especially beneficial for smart home applications.
openHAB (eclipse smarthome)

Events: Commands & Status updates

“item” abstraction: allows for seamless scalability

Modeling level

Item Registry: Keeps track of latest status

Persistence services: Takes events and stores in database.

Physical

Bindings: Translators to the real world.

- openHAB supports a long list of devices, like KNX, Homematic, Philips Hue and many more.

http://www.openhab.org/
openHAB (eclipse smarthome)

- openHAB Runtime
  - a set of OSGi framework (a pure Java solution) and so needs a JVM to run
  - highly modular architecture, allowing plug-play functionality during runtime without stopping the service

- openHAB Designer
  - an Eclipse RCP application (IDE) for configuring the openHAB Runtime
FPGAs for IoT

- *Key enabler:* Programmable aspect of an FPGA
- real-time visibility into operations and control
- hardware based security and isolation mechanisms that offer robust protection against various forms of attack
- can act as a local IoT concentrator, serving as a hub for RF links, managing link interfaces and exchanging data with these.
Why FPGAs?

- Enable hardware acceleration of complex control algorithms with optimized DSP technology
- Hardware update & platform re-configurability possible even after shipping
- Very fast setup using existing FPGA IP cores like Xillybus (DMA over PCIe)
- Shorter time to market compared to ASIC solution through re-programmability and reduced risk
- Do not go through the same time consuming physical design, design-rule closure, tape-out, and fabrication processes that ASICs do
THANK YOU