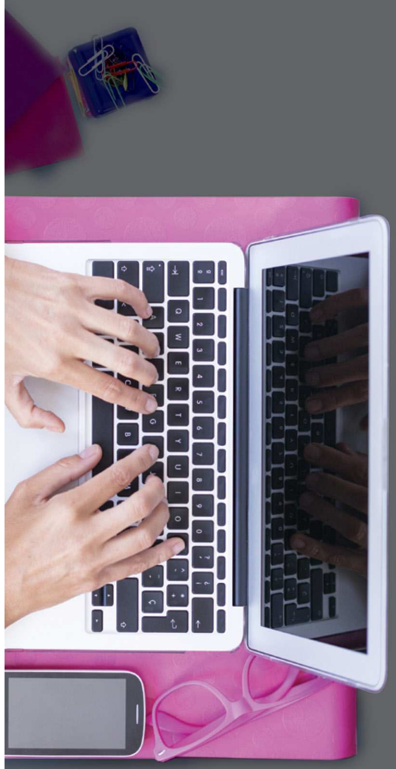


{open}:hack

tmforum | connecting digital ecosystems

TM Forum {open}:hack, Nice France, May 14-16.

# Vancouver Airwatch

## (in collaboration with Solent Airwatch)

Crowd sourced accurate air quality measurements for smart ports and cities using low cost IoT devices

SOLENT AIR WATCH

VANCOUVER AIR WATCH

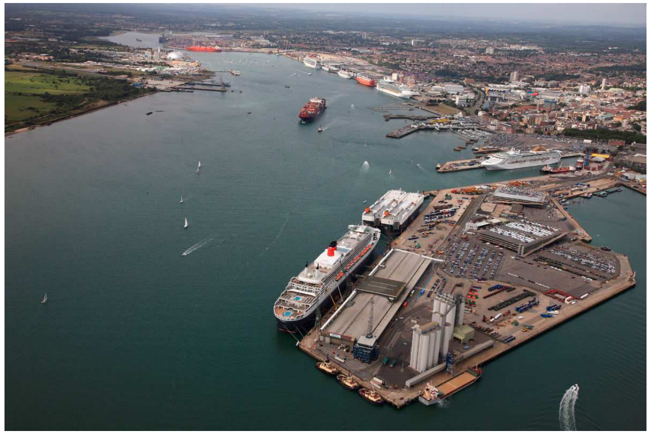
openNMS®

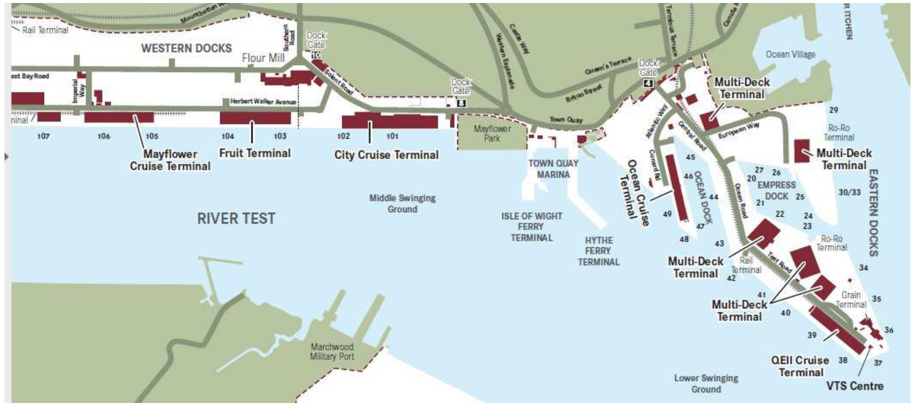
# The Challenge

- 1. Demonstrate a Smart (sea) Port Proof of Concept
  - Product Catalogue to enhance the economic potential of a major port
  - Demonstrate a Generalised Product Catalogue IoT service lifecycle management solution
- 2. Address specific problem of Air Pollution Reduction
  - Cities hosting major ports (or airports) encounter significant air quality problems
  - Accurate legally enforceable pollution measurements expensive
  - Can we used neural networks to increase the accuracy of cheap crowd sourced smoke detectors
- 3. Demonstrate feasibility of crowd sourced IoT sensor devices
  - Generalised Crowd Sourced IoT management solution which can be used in ports and cities
  - Apply a generalised catalogue portal to lifecycle of crowd sourced air quality sensors
- 4. Build on Previous work
  - Port-O-Matic - Simulated Drone based IoT sensors for ports (TMForum Nice 2017)
  - Solent Airwatch ‘Sniffy’ crowd sourced air quality detectors

# Southampton’s (Smart ) Port

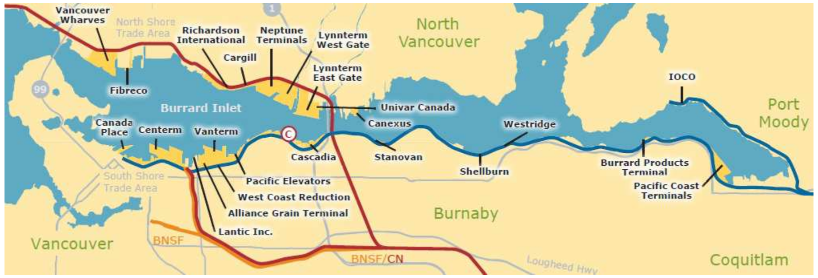
- Previous work : port-o-matic
  - Addressing the 5G and Smart city challenge
- The UK’s number one cruise port, which welcomes 1.7m passengers
- Each ship up to 6000 passenger and crew
- Ship turn around  1-2 days
- Increasing problem of air pollution due to generators running while ships in port
- Ships need dock side services including
  - Water, waste, electricity, communications
- Smart Ports
  - What is a Smart Port? no waste of space, time, money and natural resources
  - an integration of various infrastructures, both physical and IT. That includes different network technologies like radio, LAN, WAN and WLAN, RFID and positioning technologies.”
  - https://www.porttechnology.org/news/what_is_a_smart_port





# (Smart) Port of Vancouver

- https://www.portvancouver.com/
- 27 major marine cargo terminals
- Port activities annually sustain:
  - $24.2 billion in economic output
  - $11.9 billion in gross domestic product (GDP)
  - $7 billion in wages
  - 115,300 jobs in Canada
  - 96,200 jobs in British Columbia
  - $1.4 billion per year in tax revenues
- Known problems with pollution
  - Ship Shore power saves 2656 tonnes $CO^2$



# Solent Airwatch

- http://www.solentairwatch.co.uk/
- Community air monitoring project
- Not for profit charity run by Southampton volunteers
- With University of Southampton and Solent University
- Sniffy – version 1
  - https://github.com/SolentAirWatch/sniffy
  - Raspberry Pi Zero
  - Low cost laser Smoke detector
  - Ruggedized  container

SOLENT AIR WATCH

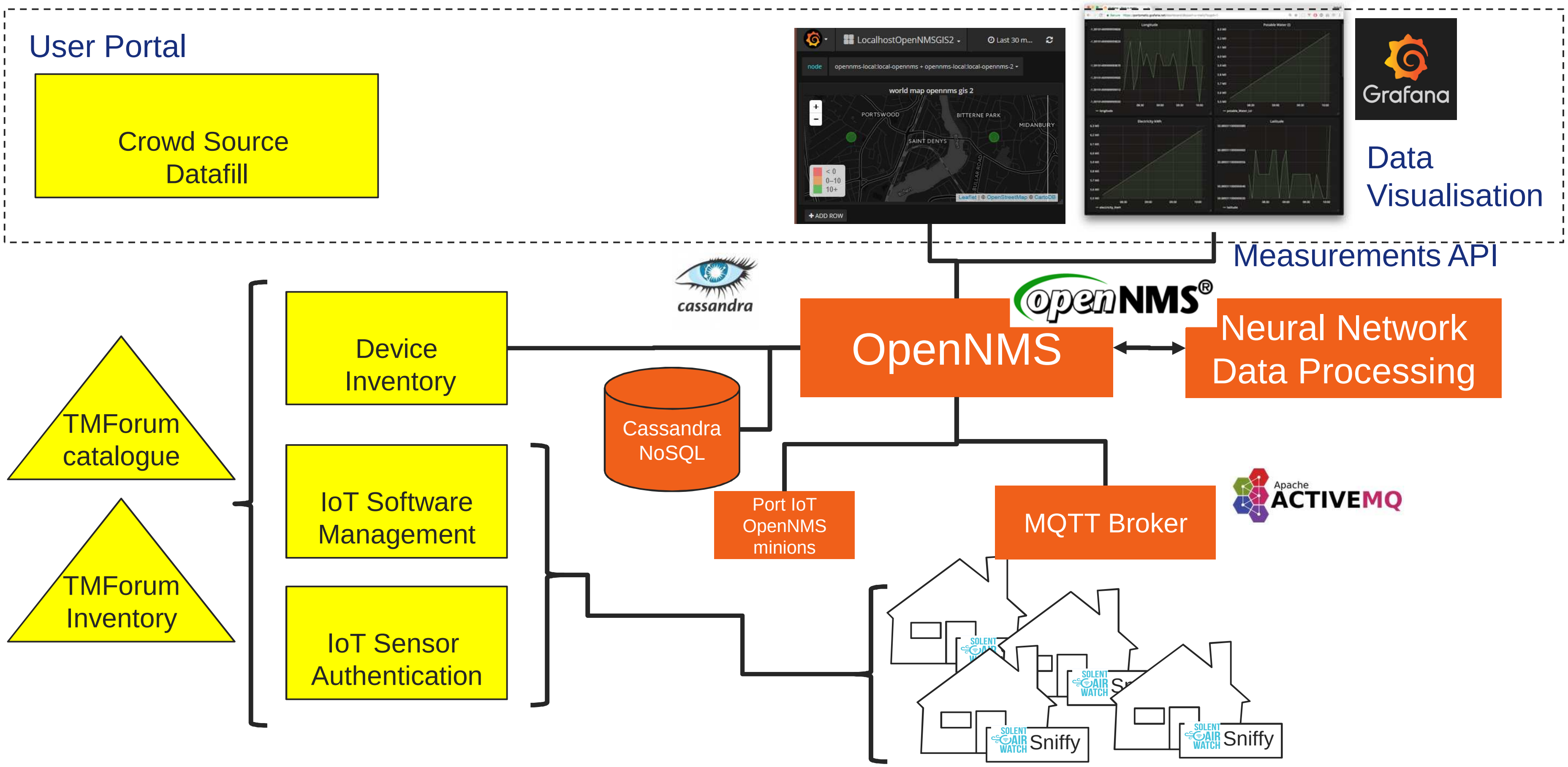
# The Solent Airwatch Challenge

- The problem
  - Currently impossible to accurately monitor city wide pollution
  - Legal enforcement of air quality requires calibrated sensors
  - Commercial calibrated air quality sensors very expensive
  - Cheap air quality sensors have wide variation in sensitivity and cant be used for enforcement
- The solution
  - Community sourced low cost air sensors
  - Gamification – web site to show your sensor data and city wide view
  - Engage schools, maker space groups , private citizens and city council
- The Differentiator - Data Science
  - Data Science makes low cost sensors viable for accurate measurements
  - Peer reviewed experiment plan to validate the science behind data processing
  - Big data processing of city wide sensor data
  - Neural Network based calibration against commercial sensors

# Crowd Sourced IoT Platform challenge

- The problem
  - Lots of attention on platforms for IoT devices themselves – Apache Karaf, Eclipse Kura, OpenHab, NodeRED etc.
  - Limited progress on common management services across a diversity of IoT devices and services
- Deployment and Maintenance of IoT software
  - Cost of installation and change - maximise reuse of IoT devices for diverse applications
  - Secure deployment – managing software and authenticating running instances
  - Service Assurance – managing issues with remote devices
  - Citizen science – managing the lifecycle of devices built by citizens but connected to our experiment

# Architecture



User Portal

Crowd Source Datafill

Data Visualisation

Measurements API

TMForum catalogue

TMForum Inventory

Device Inventory

IoT Software Management

IoT Sensor Authentication

Cassandra NoSQL

OpenNMS

Neural Network Data Processing

Port IoT OpenNMS minions

MQTT Broker

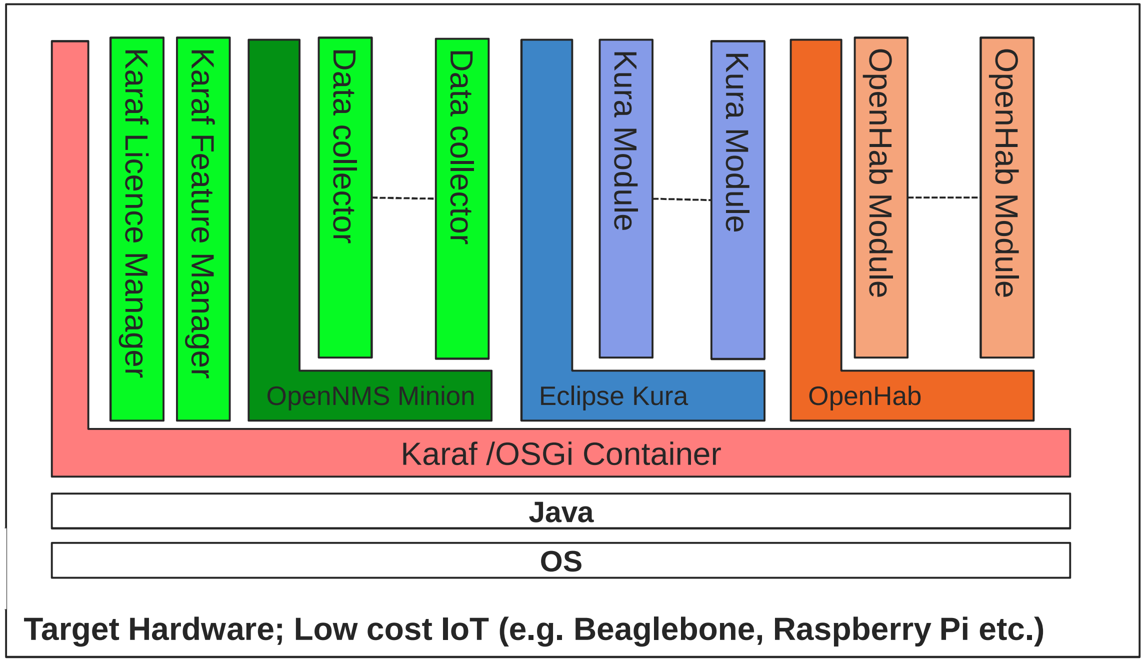
Sniffy

Sniffy

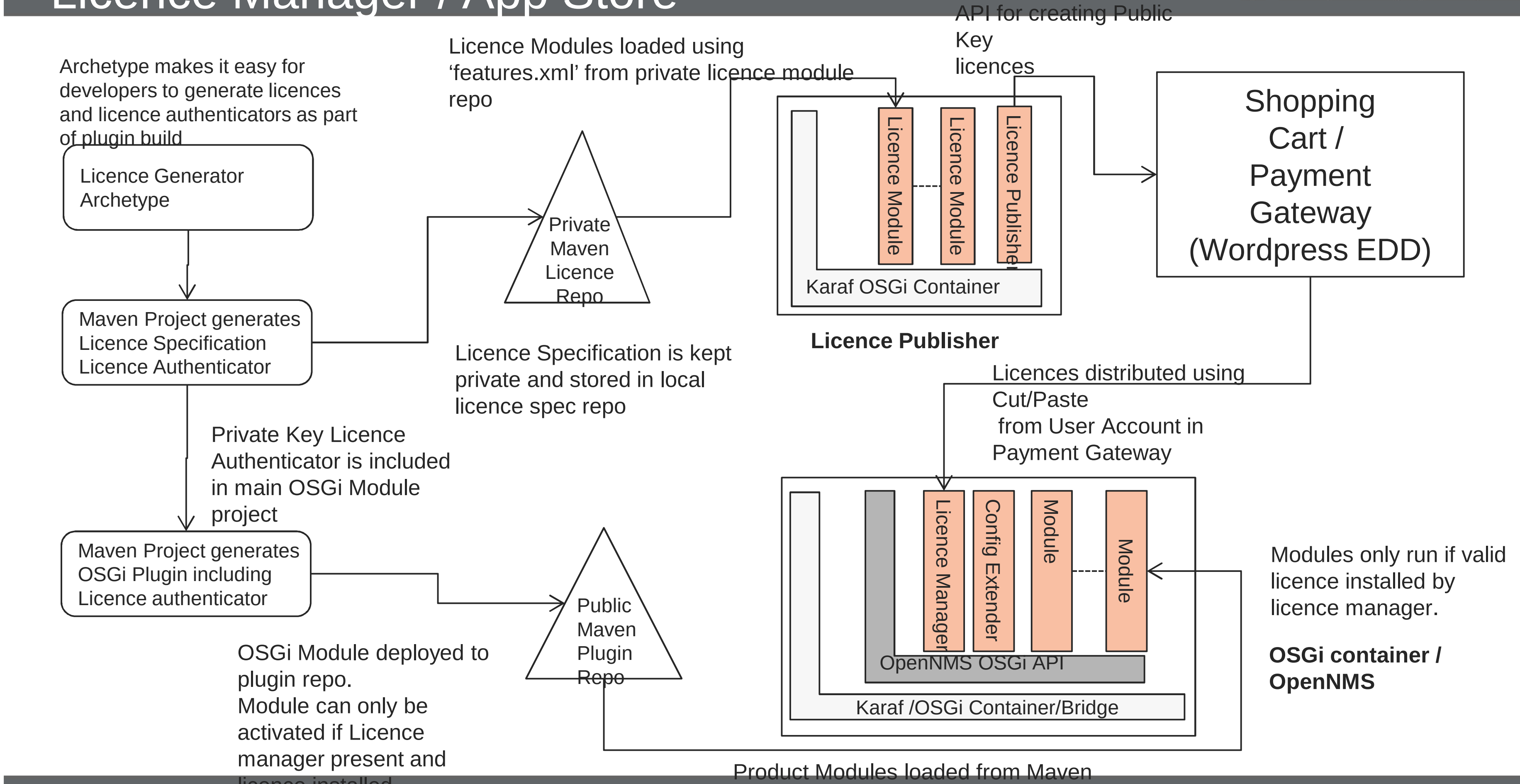
# Java IoT Ecosystem

- Java IoT Ecosystem
  - Apache karaf
  - Eclipse Kura
  - OpenHab

- OpenNMS Karaf Management
  - Minions – remote data collectors
  - Karaf Feature manager
  - Karaf  Licence manager
  - Karaf Plugin Manager
  - JMX Data collection



# Licence Manager / App Store



Archetype makes it easy for developers to generate licences and licence authenticators as part of plugin build

Licence Generator Archetype

Maven Project generates Licence Specification Licence Authenticator

Private Key Licence Authenticator is included in main OSGi Module project

Maven Project generates OSGi Plugin including Licence authenticator

OSGi Module deployed to plugin repo.
Module can only be activated if Licence manager present and licence installed

Licence Modules loaded using 'features.xml' from private licence module repo

Private Maven Licence Repo

Licence Specification is kept private and stored in local licence spec repo

Public Maven Plugin Repo

API for creating Public Key licences

Licence Module

Licence Module

Licence Publisher

Karaf OSGi Container

**Licence Publisher**

Shopping Cart / Payment Gateway (Wordpress EDD)

Licences distributed using Cut/Paste from User Account in Payment Gateway

Licence Manager

Config Extender

Module

Module

OpenNMS OSGi API

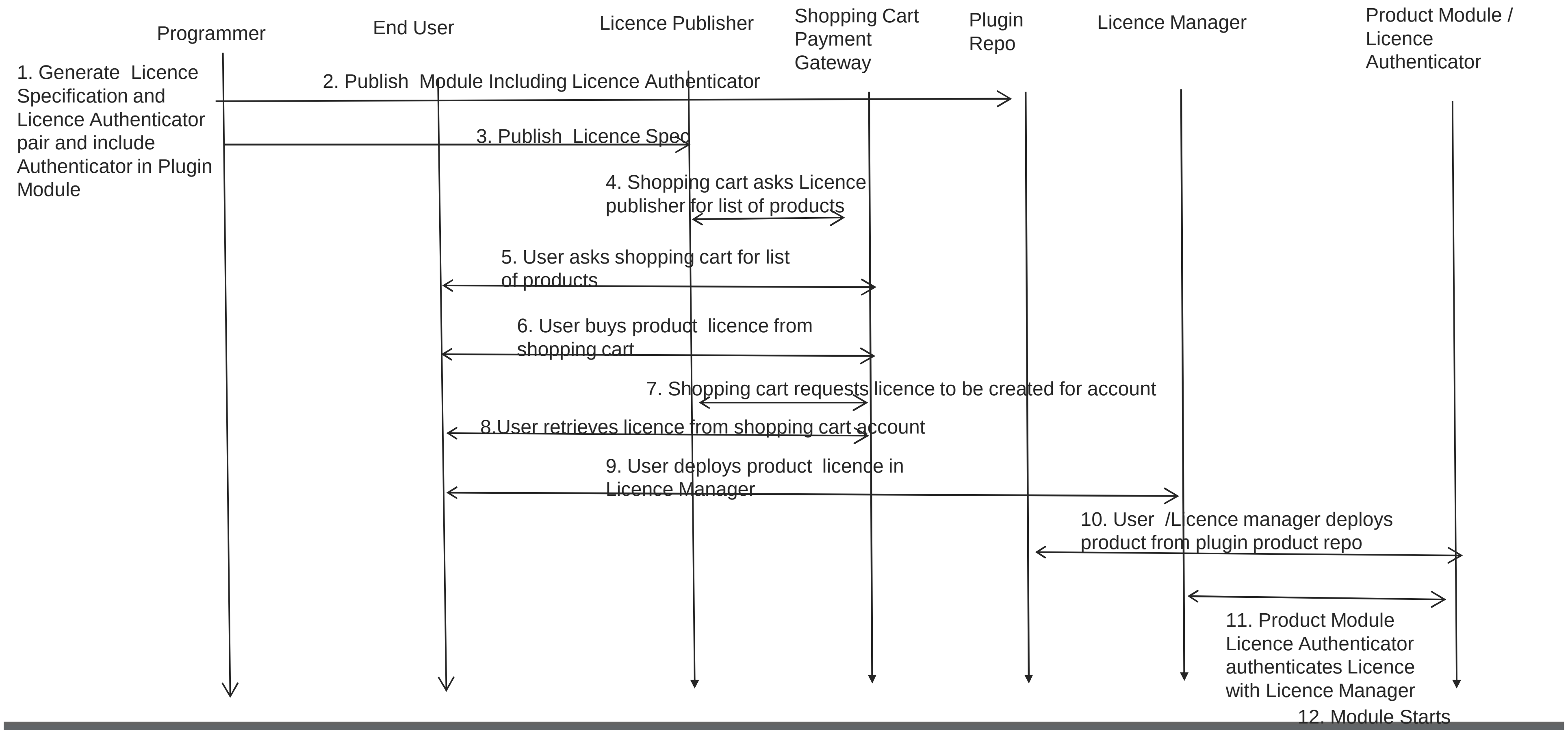
Karaf /OSGi Container/Bridge

Modules only run if valid licence installed by licence manager.

**OSGi container / OpenNMS**

Product Modules loaded from Maven plugin repo using features.xml

# Workflow for Activating Module



Programmer

End User

Licence Publisher

Shopping Cart Payment Gateway

Plugin Repo

Licence Manager

Product Module / Licence Authenticator

1. Generate Licence Specification and Licence Authenticator pair and include Authenticator in Plugin Module

2. Publish Module Including Licence Authenticator

3. Publish Licence Spec

4. Shopping cart asks Licence publisher for list of products

5. User asks shopping cart for list of products

6. User buys product licence from shopping cart

7. Shopping cart requests licence to be created for account

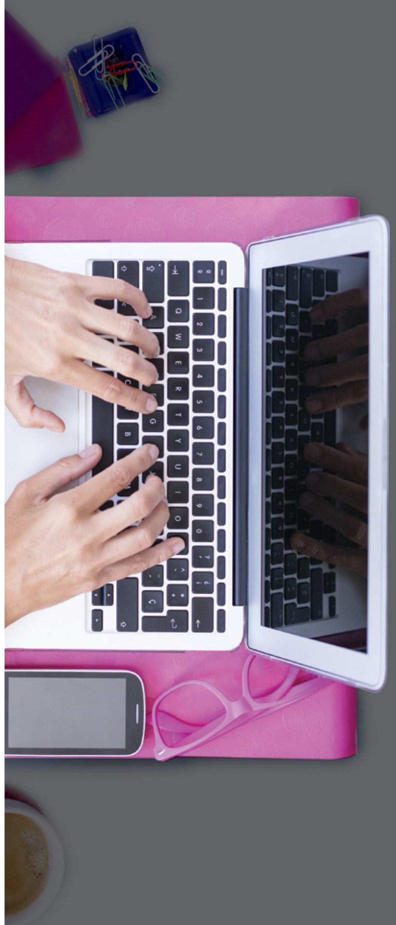
8.User retrieves licence from shopping cart account

9. User deploys product licence in Licence Manager

10. User /Licence manager deploys product from plugin product repo

11. Product Module Licence Authenticator authenticates Licence with Licence Manager

12. Module Starts



{open}:hack

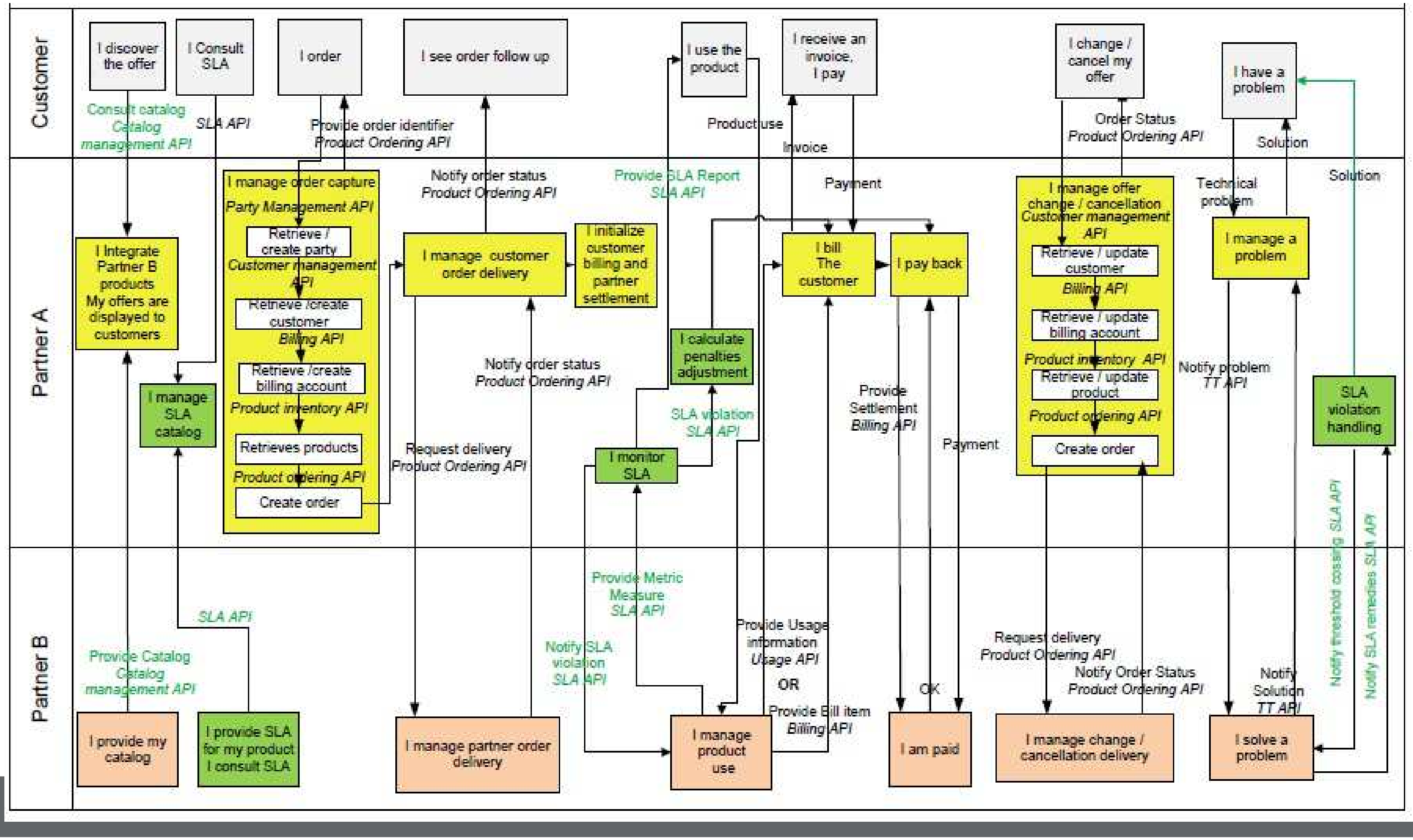
tmforum | connecting digital ecosystems

TM Forum {open}:hack, Nice France, May 14-16.

# backup

# Typical Complex TMF API usage

{open}:hack
tmforum | connecting digital ecosystems



Customer

I discover the offer

I Consult SLA

I order

I see order follow up

I use the product

I receive an invoice, I pay

I change / cancel my offer

I have a problem

Consult catalog
Catalog management API

SLA API

Provide order identifier
Product Ordering API

Product use

Invoice

Order Status
Product Ordering API

Solution

Partner A

I Integrate Partner B products
My offers are displayed to customers

I manage SLA catalog

I manage order capture
Party Management API
Retrieve / create party
Customer management API
Retrieve /create customer
Billing API
Retrieve /create billing account
Product inventory API
Retrieves products
Product ordering API
Create order

Notify order status
Product Ordering API

I manage customer order delivery

I initialize customer billing and partner settlement

Provide SLA Report
SLA API

Payment

I bill The customer

I pay back

I calculate penalties adjustment

SLA violation
SLA API

I monitor SLA

Notify order status
Product Ordering API

Request delivery
Product Ordering API

Provide Settlement
Billing API

Payment

I manage offer change / cancellation
Customer management API
Retrieve / update customer
Billing API
Retrieve / update billing account
Product inventory API
Retrieve / update product
Product ordering API
Create order

Technical problem

I manage a problem

Notify problem
TT API

Solution

SLA violation handling

Partner B

SLA API

Provide Catalog
Catalog management API

I provide my catalog

I provide SLA for my product
I consult SLA

I manage partner order delivery

Provide Metric Measure
SLA API

Notify SLA violation
SLA API

I manage product use

Provide Usage information
Usage API

OR

Provide Bill item
Billing API

OK

I am paid

Request delivery
Product Ordering API

Notify Order Status
Product Ordering API

I manage change / cancellation delivery

Notify Solution
TT API

I solve a problem

Notify threshold cossing SLA API

Notify SLA remedies SLA API

# TM Forum API's

| | |
|---|---|
| Activation and Configuration API | The REST API for Activation and Configuration allows the user to retrieve, create, update, delete services and retrieve the monitor resource used to monitor the execution of asynchronous requests on specific resource. Although all the examples given in the API specification are relative to Services, the same API can be used to Activate and Configure Services or Resources. |
| Address API | The Address API is one of the Pre-Ordering Management APIs. The Address API provides a standardized client interface to an Address management system .It allows to look for worldwide addresses. It can also be used to validate address data, to be sure that it corresponds to a real address. |
| Agreement API | The Agreement API provides standardized mechanism for managing agreements, especially in the context on partnerships between partners. |
| Appointment API | The Appointment API is one of the Pre-Ordering Management APIs. The appointment API provides a standardized mechanism to book an appointment with all the necessary appointment characteristics. First, the API consists in searching free slots based on parameters, as for example a party. Then, the appointment is created. The appointment has characteristics such as nature of appointment, place of appointment... |
| Billing Management API | Provides standardized mechanisms for billing account, bill item and settlement note advice management either in B2B or B2B2C contexts. |
| Customer Management API | Provides a standardized mechanism for customer and customer account management, such as creation, update, retrieval, deletion and notification of events. |
| Onboarding API | The Onboarding API provides standardized mechanisms for managing an onboarding process. The API allows the retrieval, creation, update and deletion of partnership type and its owned sub- resources.The intention for partner onboarding in the Digital Ecosystem is to have a lightweight approach similar to an end-user signing-on to terms and conditions for downloadable applications. The onboarding of the “Party”, the role can be Partner, Supplier, Developer, etc. The onboarding of the “Services” could be product offerings. |
| Party Management API | Provides a standardized mechanism for party management such as creation, update, retrieval, deletion and notification of events. A Party can be an individual or an organization that has any kind of relation with the enterprise. |
| Performance Management API | Provides a standardized mechanism for performance management such as the creation, partial or full update and retrieval of resources involved in performance management (Measurement Production Job, Measurement Collection Job, and Ad hoc Collection). It also allows notification of events related to performance. |
| Privacy API | The Privacy management API provides standardized mechanism for privacy profile types, privacy profiles and privacy agreements such as creation, update, retrieval, deletion and notification of events.. |
| Product Catalog Management API | Provides a standardized solution for rapidly adding partners’ products to an existing Catalog. It brings the capability for Service Providers to directly feed partners systems with the technical description of the products they propose to them. |
| Product Inventory Management API | Provides standardized mechanism for product inventory management such as creation, partial or full update and retrieval of the representation of a product in the inventory. It also allows the notification of events related to product lifecycle. |
| Product Ordering API | Provides a standardized mechanism for placing a product order with all of the necessary order parameters. The API consists of a simple set of operations that interact with CRM/Order negotiation systems in a consistent manner. A product order is created based on a product offering that is defined in a catalog. The product offering identifies the product or set of products that are available to a customer, and includes characteristics such as pricing, product options and market. |
| Quote API | The Quote API is one of the Pre-Ordering Management APIs. The customer Quote API provides a standardized mechanism for placing a customer quote with all of the necessary quote parameters. |
| Service Qualification API | Service Qualification API is one of the Pre-Ordering Management APIs. Service Qualification API goal is to provide service availability at Customer location. |
| SLA Management API | Provides a standardized interface for Service Level Agreement (SLA) life-cycle Management (SLA Negotiation, SLA configuration SLA Activation/enforcement, SLA Operations, SLA violation / consequence handling, SLA reporting) between a Customer and a Service Provider which provides offers (product with attached SLA in its catalogue) the customer can discover, browse, trigger and order. |

# TM Forum API’s

| | |
|---|---|
| Trouble Ticket API | Provides a standardized client interface to Trouble Ticket Management Systems for creating, tracking and managing trouble tickets among partners as a result of an issue or problem identified by a customer or another system. Examples of Trouble Ticket API clients include CRM applications, network management or fault management systems, or other trouble ticket management systems (e.g. B2B). |
| Usage Management API | Provides standardized mechanism for usage management such as creation, update, retrieval, import and export of a collection of usages. The API manages both rated and non-rated usage. For example, it allows a service provider to 1) retrieve usage generated by a partner service platform in order to rate it and 2) to provide rated usage to a partner for consumption follow up purposes. |
| Service Problem Management API (SPM) | The SPM API is used to manage service problems. Service problems are generated based on the information declared by a partner or the event information notified from infrastructure providers.The event information includes alarm information, performance anomaly information, trouble ticket information, SLA violation, maintenance information and prediction information. |
| Service Ordering API | The REST API for Service Order Management provides a standardized mechanism for placing a service order with all of the necessary order parameters. It allows users to create, update & retrieve Service Orders and manages related notifications. |
| Service Catalog API | The Service Catalog Management API allows the management of the entire lifecycle of the service catalog elements. |
| Service Test Management API | The Service Test API provides a standardized mechanism for placing a service test with all of the necessary test parameters. The API consists of a simple set of operations that interact with CRM/Service Management systems in a consistent manner. A service test is a procedure intended to check the quality, performance, or reliability of a service. |
| Change Management API | The Change Management API provides the standard integration capabilities between external applications and Change Management Application. The API consists of a simple set of operations that interact with Change Request in a consistent manner. A Change Request is an IT service management discipline. |
| Service Inventory API | The intent of this API is to provide a consistent/standardized mechanism to query and manipulate the Service inventory. |
| Loyalty Management API | The Loyalty API supports the management of loyalty program specifications, loyalty program members, their associated products and loyalty accounts with loyalty balances. The scope of the API also covers the management of loyalty rules and under what conditions the associated loyalty actions must be applied. |
| Service Quality Management API | Through this API, any Enterprise is able to access a Service Quality Management application and extract Service Level Specifications and associated Service Level Objectives (SLO) and their thresholds. |
| NFV Entity Provisioning API | REST API for NFV Entity Provisioning i.e. provisioning and lifecycle management of Network Services composed from Physical and Virtual Network Functions |
| Resource Ordering API | The REST API for Resource Order Management. It includes the model definition as well as all available operations. Possible actions are creating, updating and retrieving Resource Orders (including filtering). A Resource Order API provides a standard mechanism for placing a Resource Order with all necessary order parameters. |
| Resource Catalog API | The Resource Catalog Management API REST specification allows the management of the entire lifecycle of the Resource Catalog elements, the consultation of resource catalog elements during several processes such as ordering process, campaign management, sales management. |
| Prepay Balance Management API | REST API for Balance Management. It includes the model definition as well as all available operations for prepay balance management. Prepaid subscribers pay fees before using services. Therefore, the subscribers must have sufficient balances. Operators can provide multiple recharge channels for subscribers. Subscribers can pass credit between different subscriptions, therefore transferring balance from one account to another. |

# TM Forum OPEN HACK

{open}:hack
tmforum | connecting digital ecosystems

open:api

open:platform

open:data

+ open:innovators

= open:hack

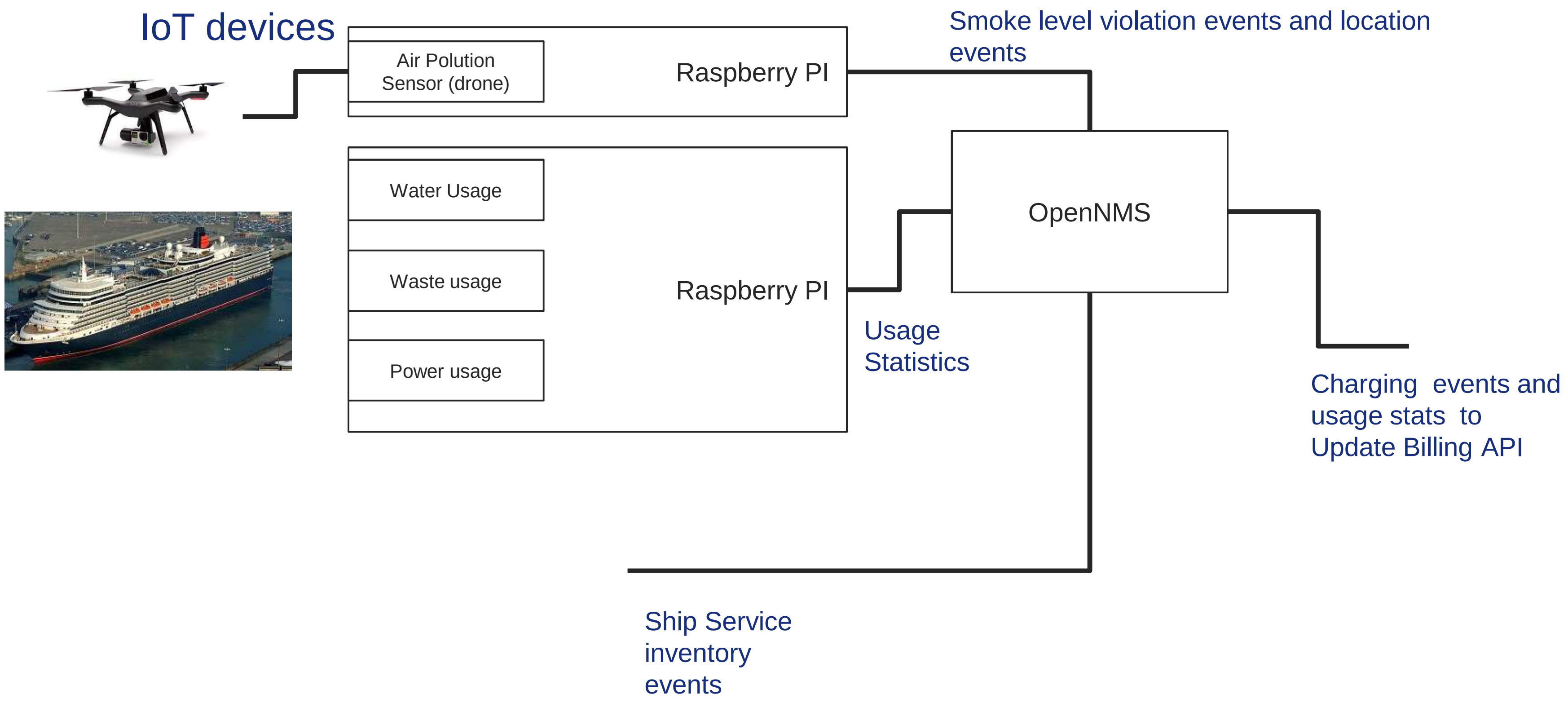
- DRAFT

- Requirements
  - Common IoT platform used across applications
  - Multiple independent applications/services  possible on platform
  - Independent developers can write and test applications
  - Applications authenticated to run on specific IoT devices using certificates/licences
  - Common Management platform controls community of ‘known’ IoT devices
    - Control software delivery and upgrade
    - Monitor applications and remote devices
    - Manage authentication of devices
- Technology platform
  - Java / OSGi
  - All code Open Source
  - See: https://github.com/gallenc/tmforumhack2017/tree/master/TMForumNice2017Hack

- Configuration

  - Security
    - Each Pi can run  multipe services
    - Each service needs autehnticated to run on Pi
    - Remote managemetn of service

# Smart Port API Scenario - utilities



IoT devices

Air Polution Sensor (drone)

Raspberry PI

Smoke level violation events and location events

Water Usage

Waste usage

Power usage

Raspberry PI

OpenNMS

Usage Statistics

Charging events and usage stats to Update Billing API

Ship Service inventory events

# API’s to order service

## *Use of  API’s*

- Catalog API
  - List of products available from which we can choose
- Service Qualification API
  - Service Qualification API goal is to show service availability at Customer location. We will use this to see if the desired services are  available at a given birth
- Appointment API
  - Booking a time slot in the birth for a given service
- Address API
  - Location of birth
- Order API  - not clear if Quote API or other
  - - We will need one other api to place an order

# API’s to bill or report problems

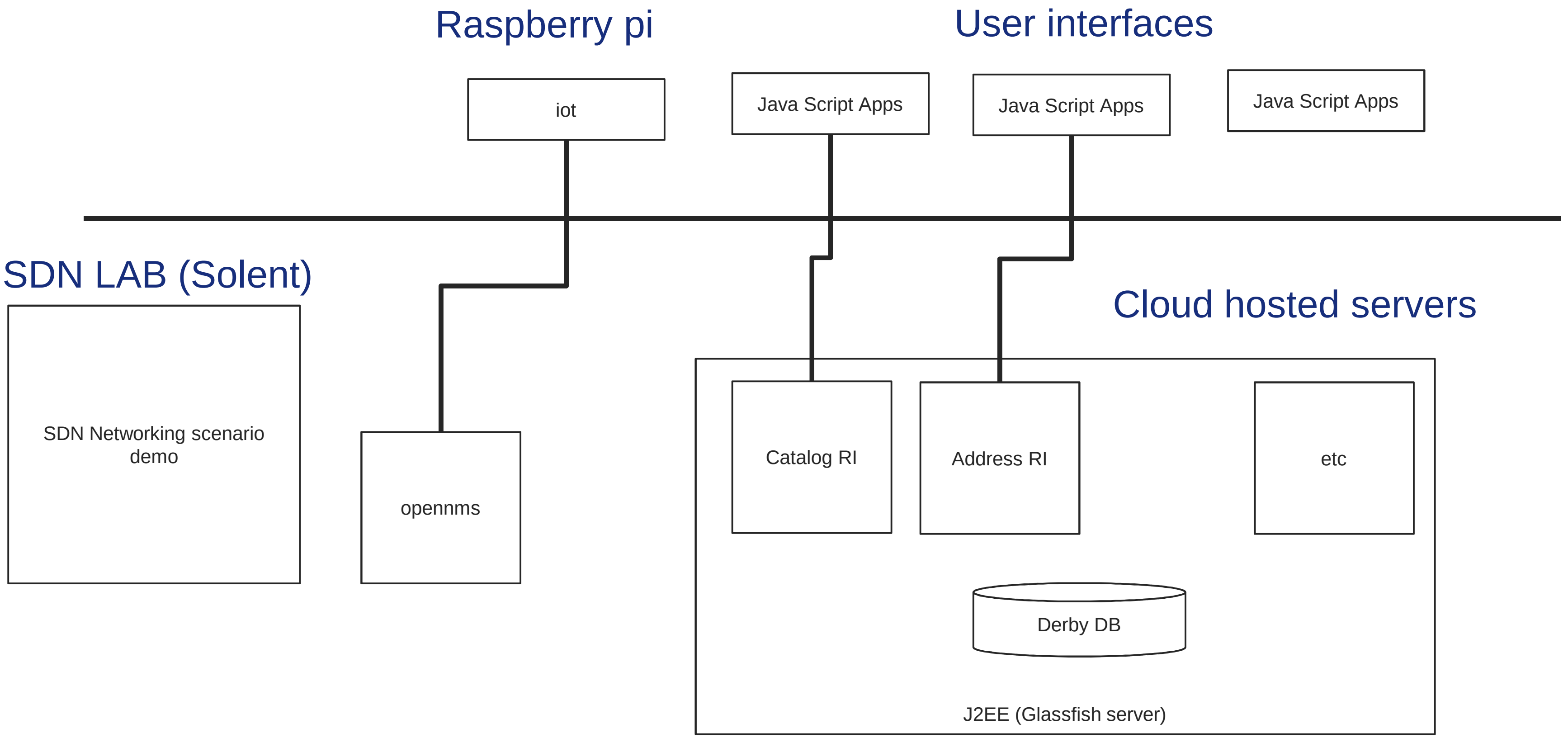
## *Use of API’s*

- Service Problem Management API / TT api
  - ❑ Show when drone detects excessive smoke from ship
- Billing API
  - ❑ Produce bill based upon usage at end of term in birth

# TMF Resources

- **Resources**
  - **https://github.com/tmforum/   Has both example code, RI's Example apps and specification documents**
  - **Specs: https://projects.tmforum.org/wiki/display/API/Open+API+Table**
- **Reference implementations**
  - **Product catalogue api RI**
    - **https://github.com/tmforum/DSPRODUCTCATALOG2**
  - **Product catalogue app – in JavaScript**
    - **https://github.com/tmforum/DSM-Product-Catalog-API-App**
    - **This runs against an RI instance at** http://tmforum-test.apigee.net/v1/tm-forum-product-category-api/
- **Product Order API ( note this is ant build and requires netbeans – haveny yet been able to build)**
  - Product order RI
    - https://github.com/tmforum/DSPRODUCTORDERING
  - Product order app – in javascript
    - https://github.com/tmforum/DSM-Product-Ordering-API-App
    - Runs against RI instance at  http://tmforum-test.apigee.net/v1/tm-forum-product-order-api

# Architecture



# VANCOUVER

AIR
WATCH