OCCloware : Year 1 Milestone
Docker Studio, Studio Factory, pluggable XaaS runtime & Linked Data use case end-to-end POC

Marc Dutoo, Open Wide / Smile

Cloud Expo Europe, London
Open Cloud Developer Park - April, 13rd 2016
Overview

Speaker

- Marc Dutoo, Head of R&D Dept. at Open Wide, a Smile group company
  - OCCIware coordinator, Data / Cloud expert

Schedule

- 6' OCCI(ware) introduction
- 7' OCCIware Year 1 Outputs
- 7' LDaaS end-to-end proof of concept with erocci
OCCLware Factsheet

• 72 man year, 5,6m€ budget, sponsored by French ministry of Industry over 2015-2018
• 3 academics, 5 companies, 2 associations
• To lower Cloud Computing adoption costs and break up barriers between its various implementations, layers, domains
  - Especially Data Center, deployment, Big Data, Linked Data
• By bringing to OGF's Open Cloud Computing Interface (OCCI) the power of formal languages and model driven engineering (MDE)
Cloud Computing – the problem

- Too many technologies
- Lack of overarching standard

Domains:
- Application
- Platform
- Infrastructure

Layers:
- Laptops
- Monitoring
- Content
- Collaboration
- Communication
- Finance
- Desktops
- Object Storage
- Identity
- Runtime
- Queue
- Database
- Compute
- Block Storage
- Network
- Tablets

Partitioning - Lock-in - Partitioning - Lock-in - Partitioning
Everything is **Resource** or **Link**, be it at ...
OCCLware Target Outputs

• A formal, model-driven platform to manage any Cloud resource
  - Formal model of OCCI, on MIT's Alloy – Inria & TSP
  - OCCLware Studio, on Eclipse EMF & Sirius – Obeo
  - OCCLware@Runtime & console, using Models@Runtime and erocci
    - ActiveEon & Scalair
  - Deploy@OCCLware - ActiveEon & UJF

• 4 Use cases
  - Data Center as a Service / IaaS, on Scalair infrastructure
  - Big Data / HPC, on ActiveEon ProActive HPC platform
  - Linked Open Data, on Ozwillo app store's Datacore – Pôle Numérique & Open Wide
  - Deployment interoperability, on Linagora & ActiveEon's

• Open Source (OW2, Eclipse) and standardization (OGF) with help from a 10-strong international Scientific Orientation Committee towards an international standard
OCCLware Year 1 Main Outputs

• OCCLware Studio
  - **OCCLware Studio Factory**: produce visually customizable diagram editors for any Cloud configuration business domain modeled in OCCI using the OCCI Extension Studio
  - such as the flagship **Docker Studio**
  - **Models@Runtime**: e.g. deploy your Docker diagram by a mere click in the Docker Studio and see the result there

• OCCLware runtime
  - erocci, a scalable generic **OCCI bus** written in erlang
  - that federates multiple Cloud runtimes ("**backends**") using e.g. **Java** through DBus or **python**,
  - such as the Roboconf **PaaS server** and the ActiveEon Cloud Automation **multi-iaas connector**
Big Picture – Studio to Runtime to all providers

- Studio
- XML-based Extension
- OCCI
- OCCI Requests
- ozwillo
- Roboconf
- ProActive Cloud Automation
- Scalair Cloud Manager
- Cloud Manager
OCCIware generic Studio

• Generate doc, runtime configuration, scripts...
Docker Studio

• From red to green: let your models go live!
OCCIware Studio Factory

... your own Studio!
Runtime Architecture

HTTP
CoAP
XMPP
...

Non-functional

authn
listeners API
rendering/parsing

model checking
OCCI Runtime Kernel

authz / ACL
backend APIs
HTTP/OCCI API
native API (Java...)

deployment
backend repository management
fault-tolerance

DB (SQL, NoSQL...)

OpenStack / EC2 / ...

Docker

Web Services...

erocci

Technology providers & adopters
Supported OCCI Extensions

- OCCI Core
  - OCCI Infrastructure
  - OCCI Platform
    - ProActive Cloud Automation
    - Public Clouds
    - OCCIware Runtime
    - Roboconf
    - BDaaS
    - LDaaS
  - OCCI Compute Templates
  - Hypervisors
  - Docker
- OCCI Monitoring
- OCCI SLA
- Simulation
OCCI POC: Linked Open Data

• Linked Open Data ? That's Open Data sets that can be cross-queried because they have been reconciled together
• Enter Ozwillo Datacore - it holds data that is shared between applications of the Ozwillo app store : geographical elements, organizations, reusable app business data...
• Datacore OCCI(ware) use cases = Datacore « configuration as a Service » use cases : letting app developers configure...
  - its own data models, their rights and governance policy,
  - its own usage profiles : data collaboration app, high-write for IoT sensor status notifications, analytics for aggregating those
  - define custom indexes, up to deploy dedicated data Caches as a Service…
• => demo of OCCIware Studio & Runtime's first features
  - demo 1 : try to Dockerize current Ozwillo production architecture
  - demo 2 : governance of Linked Data projects (and their models)
Linked Data scenarii overview

LDNode
name: main
source: -

Current Ozwillo Infrastructure
- VM
- VM

OCCI Infrastructure
- CPU
- Storage
- Network
- VM

LDNode
name: myNode
source: main

Mongo primary
Mongo synchro
Mongo secondary
Datacore synchro
Dedicated Mongo

LD OCCI API
Datacore API

LDProject
status: published
LDUseProfile
robust: false
LDUseProfile
storage: secondary

Project governance
Energy consumption probes
Deep data analysis

My Ozwillo App
Custom optimized business requests
D1 – Docker-specific Studio
D2 - Studio : OCCI extension

For governance of Linked Data projects & models
...to doc & runtime configuration

*.occie extension file → Cloud Designer → Generate textile doc, erocci runtime XML conf...
Studio: OCCI configuration
Governance action example:
in order to publish a stable version of the org_0 project, let's **freeze all its models**:

```
*.occic configuration file > Cloud Designer > Generate curl script:

```
Runtime setup (erocci with DBus)

- **Configuration**
  - OCCI extension configuration: Core, **LDProject**
  - erocci backend: **DBus**, with the erocci-dbus-java bridge deployed along, which talks to the regular **REST API** of the Ozwillo Datacore

```bash
vi sys.config
...
{erocci_backend_dbus,
 [{schemas, [
  {path, "/tmp/occi.xml"},
  {path, "/tmp/org.occiware.ozwillo.data.models.xml"}]]}
},
...
```

- **Let's start it using docker:**

```bash
sudo docker run --name="erocci_linked-data" -v `pwd`/sys.config:/tmp/sys.config -v `pwd`/occi.xml:/tmp/occi.xml -v `pwd`/org.occiware.ozwillo.data.models.xml:/tmp/org.occiware.ozwillo.data.models.xml -P -t -i erocci/erocci
```
Initially, models are unfrozen

As shown in Linked Data server backoffice (API Playground):
Freeze models – ask erocci by a POST (done in curl)

```
$ curl -v -H 'content-type: application/json' -X POST http://localhost:8080/collections/project/ -d @occiware_geo_0_published.json
Host: localhost:8080
User-Agent: curl/7.35.0
Accept: */*
content-type: application/json
content-length: 135

> POST /collections/project/ HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost:8080
> Accept: */*
> content-type: application/json
> Content-Length: 135

> upload completely sent off: 135 out of 135 bytes
< HTTP/1.1 201 Created
< connection: keep-alive
< date: Wed, 23 Mar 2016 12:01:38 GMT
< content-length: 0
< Server erocci OCCII/1.1 is not blacklisted
< server: erocci OCCII/1.1
< content-type: text/plain
< vary: accept
< location: http://localhost:8080/collections/project/2229d339-1889-3da8-9115-5ae853e77c5
```
Freeze models – erocci DBus to Java bridge calls Linked Data server REST API

BackendDBusService [Java Application] /usr/lib/jvm/jdk1.8.0_45/bin/java (23 mars 2016 13:02:05)

ID: 2
Address: http://localhost:8180/dc/type/dcnp:Project_0/geo_0
Http-Method: PUT
Content-Type: application/json
Headers: {Content-Type=[application/json], Accept=[application/json], X-Datacore-Project=[oas...]
Payload: {
    "@id": "http://data.ozwillo.com/dc/type/dcnp:Project_0/geo_0", "o:version": 38, "@type"...}

mars 23, 2016 1:07:59 PM org.apache.cxf.interceptor.LoggingInInterceptor
INFOS: Inbound Message

ID: 2
Response-Code: 200
Encoding: ISO-8859-1
Content-Type: application/json
Headers: {content-type=[application/json], Date=[Wed, 23 Mar 2016 12:07:59 GMT, Wed, 23 Mar 2...}
Payload: {
    "@id": "http://data.ozwillo.com/dc/type/dcnp:Project_0/geo_0", "o:version": 39, "@type"...}
At the end, models are frozen

As shown in Linked Data server backoffice (API Playground):

```
Datacore (http://localhost:8180):
/dc/type/dcmp:Project_0/geo_0

{
  "@id": "http://data.ozwillo.com/dc/type/dcmp:Project_0/geo_0",
  "o:version": 42,
  "dcmp:frozenModelNames": [
    *
  ],
}
What now in OCCIware?

• In the works:
  - Studio: more generators & connectors (ex. get current runtime state and update the diagram accordingly), backend generation, integrating simulator, decision-making tool...
  - Runtime: complete erocci-dbus-java bridge, use it to integrate all backends behind erocci
  - Deployment, monitoring, OCCI administration console
  - use case platforms development / deployment / setup...
  - and especially extending this Linked Data as a Service implementation to support all scenarii, including IoT ones!
Any questions?

Thanks for your attention!

Contact: http://www.occiware.org - marc.dutoo at openwide.fr

Source: https://github.com/occiware

Partners:

Sponsors:

DGE (PIA) & System@tic, SCS, Images & Réseaux, PICOM, Minalogic clusters