FIWARE
Connecting the dots

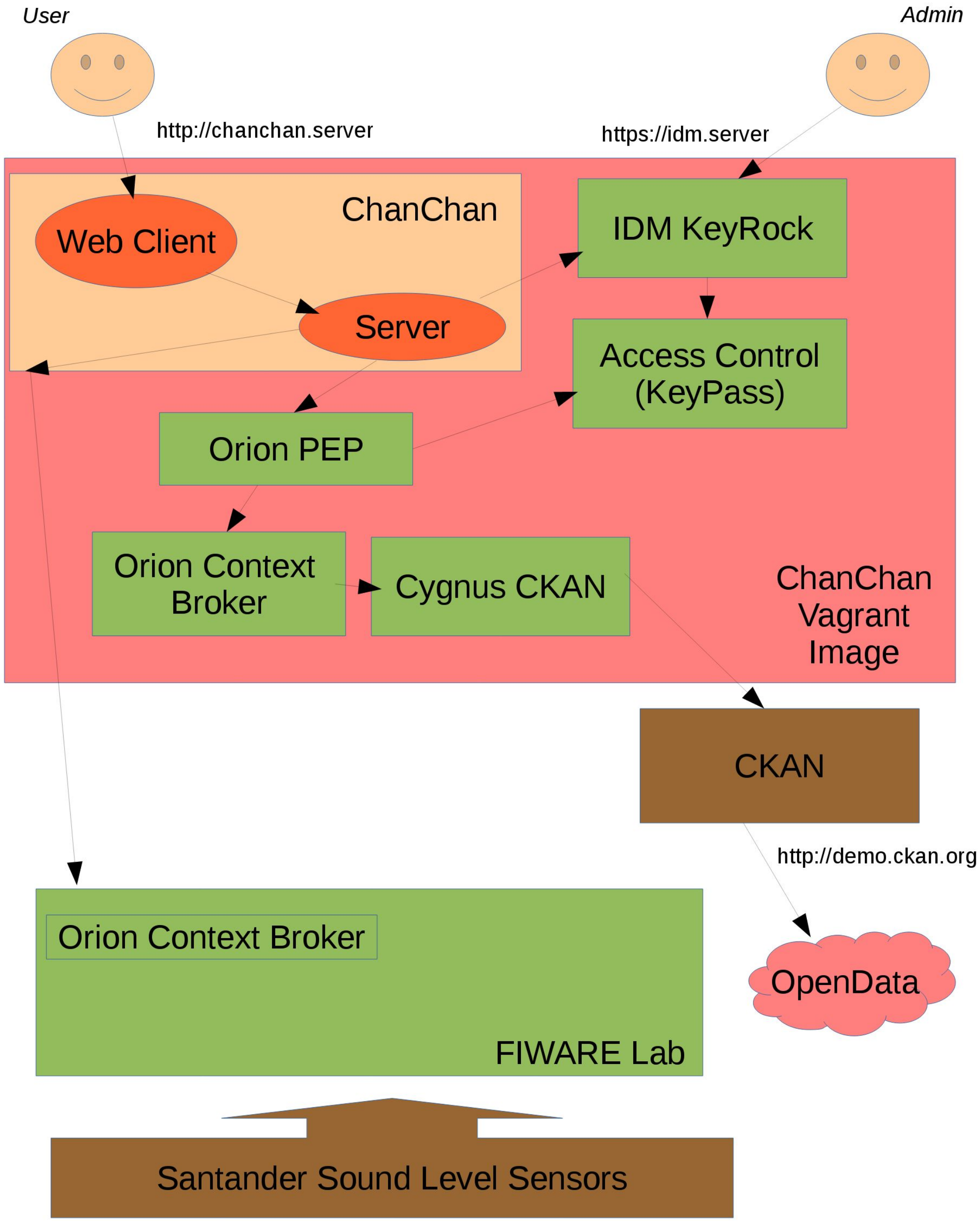
Bitergia Team: fiware-testing@bitergia.com
https://github.com/Bitergia/fiware-devguide-app

Summary

- Introduction
- Docker and docker compose
- FIWARE tour guide
- fiware-devguide-app
  - Architecture
  - Orion Context Broker
  - IoT Agents (IDAS)
  - Cygnus
  - Security GEs
  - REST API

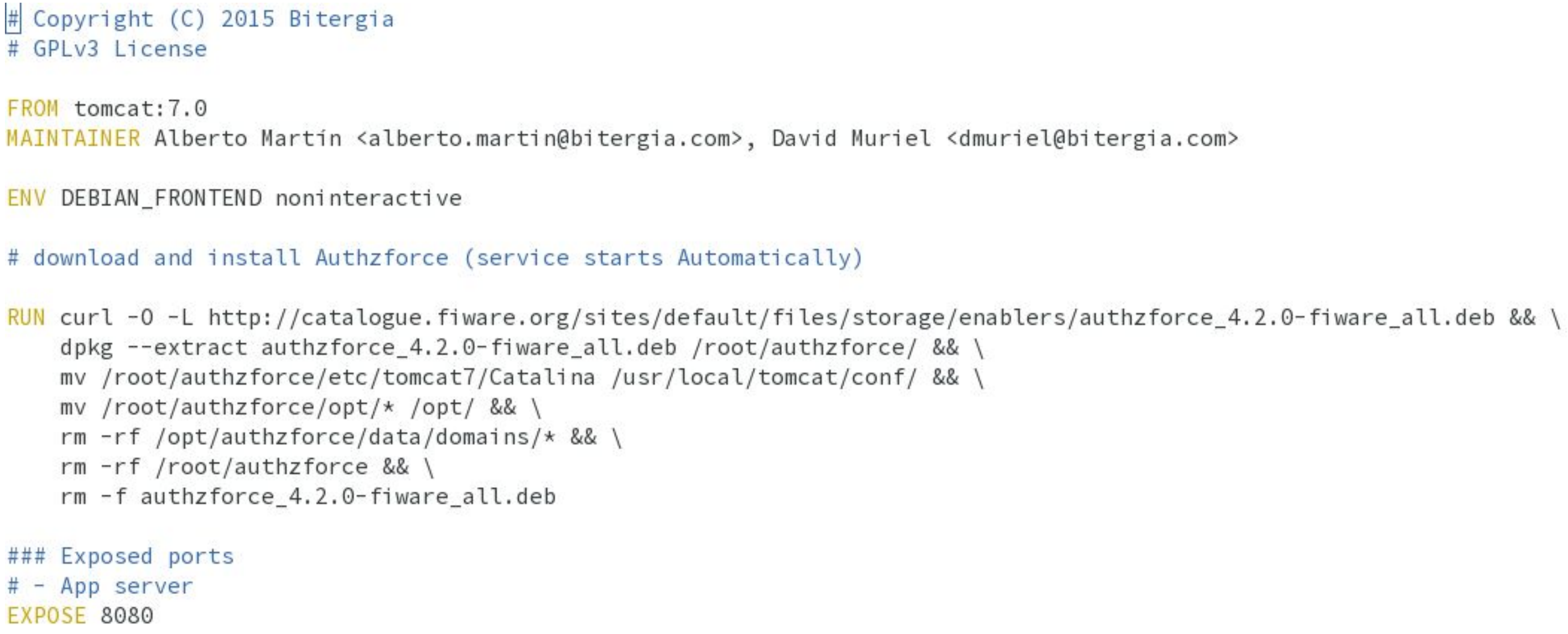
Introduction

- First iteration connecting GEs: fiware-chanchan
- Vagrant image
  - Security GEs
  - Orion Context Broker
  - Santander Sound Level sensors
  - Published data in CKAN
  - REST API


User
Admin
http://chanchan.server
https://idm.server
ChanChan
Web Client
Server
IDM KeyRock
Access Control (KeyPass)
Orion PEP
Orion Context Broker
Cygnus CKAN
ChanChan Vagrant Image
CKAN
http://demo.ckan.org
OpenData
Orion Context Broker
FIWARE Lab
Santander Sound Level Sensors

Docker

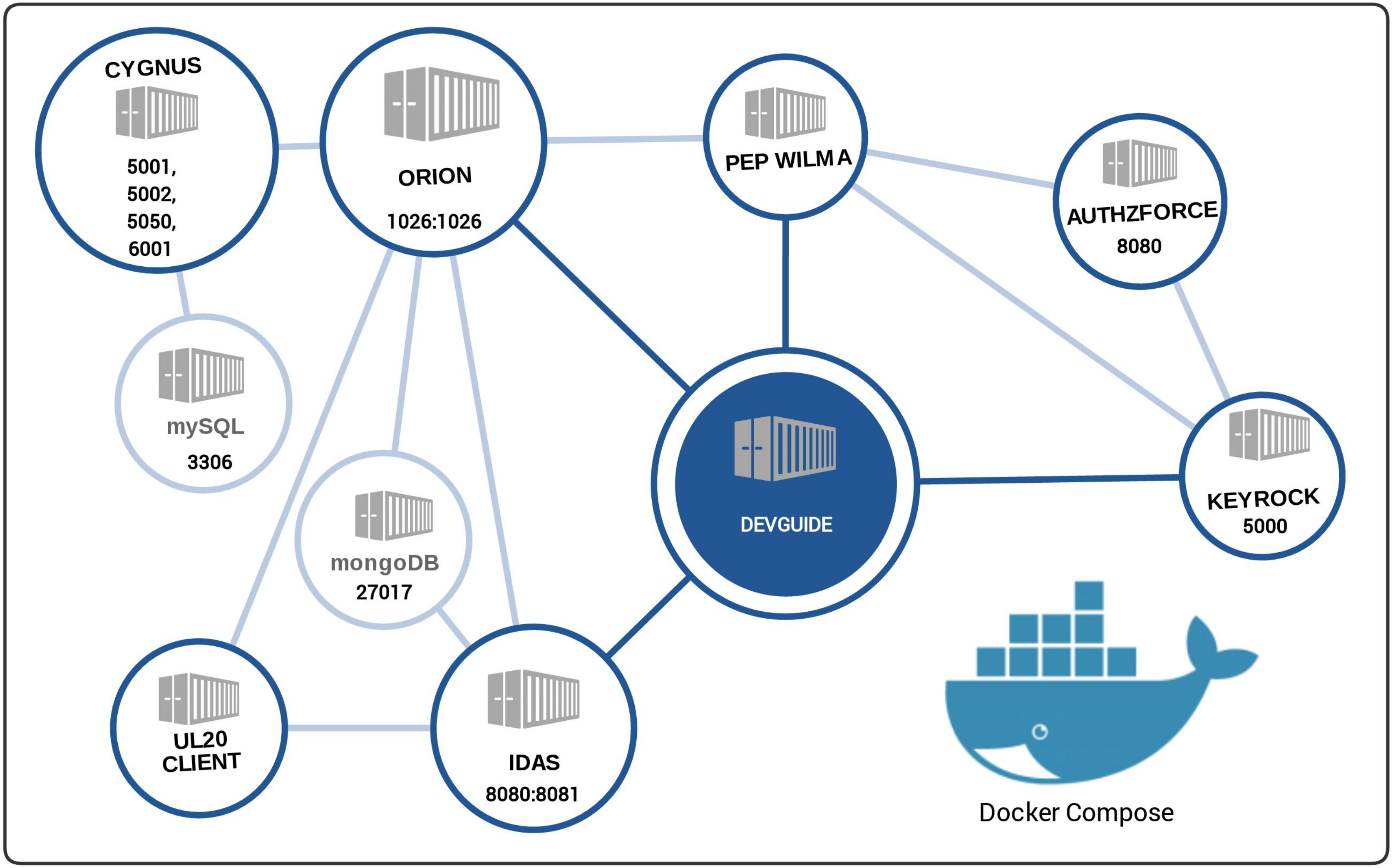
- Why Docker?
  - Allow to start tenths of containers in seconds
  - Development environment
  - GEs isolation
  - Easier configuration and integration



```
# Copyright (C) 2015 Bitergia
# GPLv3 License

FROM tomcat:7.0
MAINTAINER Alberto Martín <alberto.martin@bitergia.com>, David Muriel <dmuriel@bitergia.com>

ENV DEBIAN_FRONTEND noninteractive

# download and install Authzforce (service starts Automatically)

RUN curl -O -L http://catalogue.fiware.org/sites/default/files/storage/enablers/authzforce_4.2.0-fiware_all.deb && \
    dpkg --extract authzforce_4.2.0-fiware_all.deb /root/authzforce/ && \
    mv /root/authzforce/etc/tomcat7/Catalina /usr/local/tomcat/conf/ && \
    mv /root/authzforce/opt/* /opt/ && \
    rm -rf /opt/authzforce/data/domains/* && \
    rm -rf /root/authzforce && \
    rm -f authzforce_4.2.0-fiware_all.deb

### Exposed ports
# - App server
EXPOSE 8080
```

Docker-compose

- Link containers by writing a simple .yml file
- Security: expose just the ports needed
- Environment variables for configuration
- Volumes (host and containers) for data persistance and development

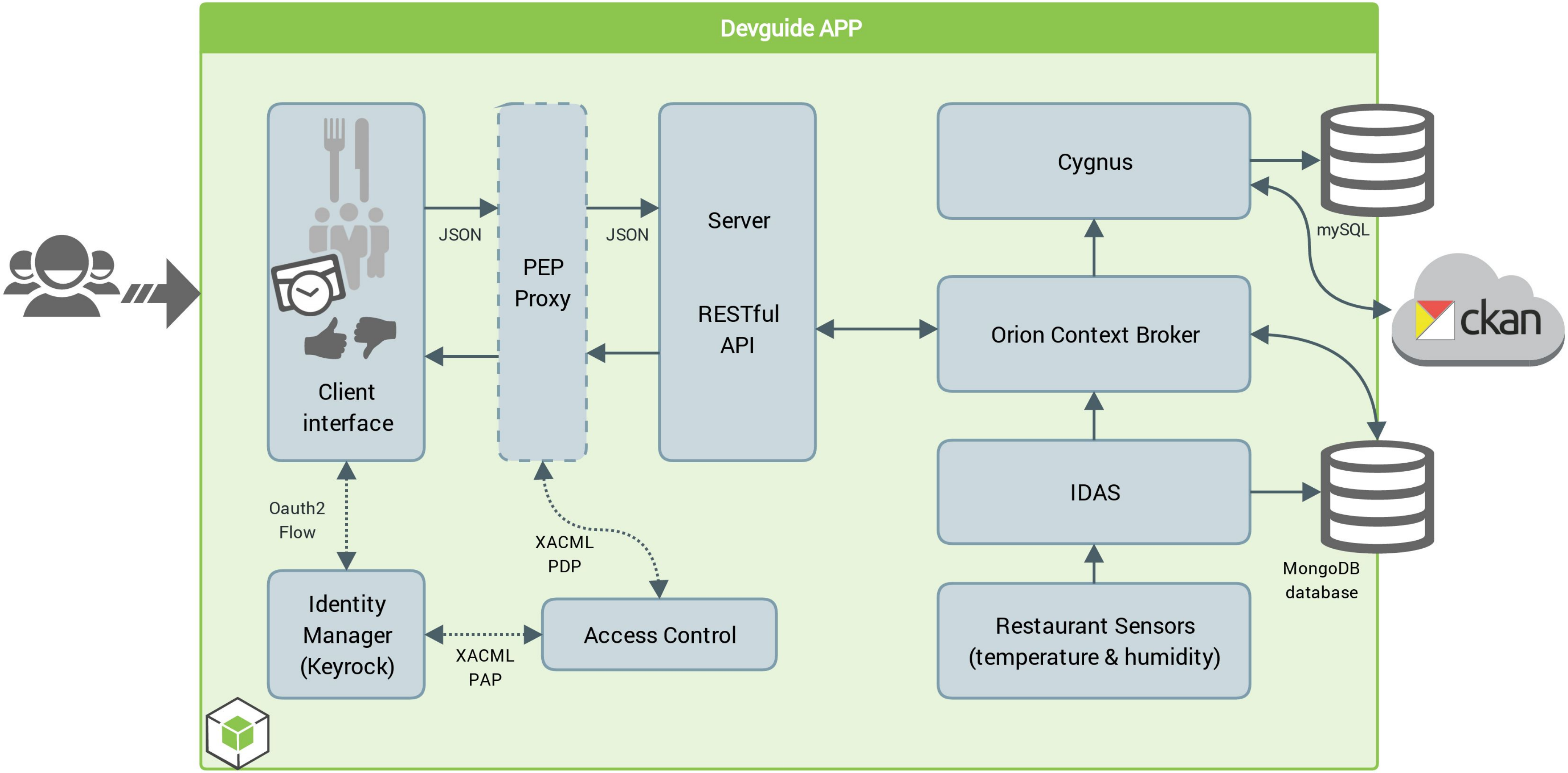

CYGNUS
5001,
5002,
5050,
6001
ORION
1026:1026
PEP WILMA
AUTHZFORCE
8080
mySQL
3306
mongoDB
27017
DEVGUIDE
KEYROCK
5000
UL20
CLIENT
IDAS
8080:8081
Docker Compose

Tour Guide

- Provides different examples of what you can create using Generic Enablers
  - Context Aware applications
  - Connect to the Internet of Things
  - Publish context information as Open data
  - Handling authorization and access control to APIs
  - More...

fiware-devguide-app

- Example application to show how to implement the different scenarios of the Tour Guide
- The application allows to manage large Restaurant chains, reviews and reservations
- Based on docker containers
- Using the last GEs versions
  - New Keyrock (Openstack based)
  - New Access Control (now open-source)
  - Orion operability using NGSIv2
  - Implemented IoT agents (temperature and humidity sensors)
  - Cygnus now also uses Hadoop sink


Devguide APP
Client interface
JSON
PEP Proxy
JSON
Server
RESTful API
Cygnus
mySQL
ckan
Orion Context Broker
IDAS
MongoDB database
Restaurant Sensors (temperature & humidity)
Oauth2 Flow
XACML PDP
Identity Manager (Keyrock)
XACML PAP
Access Control

Orion Context Broker

- Restaurant data pre-loaded image
- 600+ restaurants added using Euskadi OpenData
- Added Geo-location using Google Maps API
- AggregateRating context based on reviews
- Multi-tenancy based on Franchises
- Using subscriptions and notifications to keep restaurant contexts updated with the last sensors data (humidity and temperature)

IoT Agents

- Sensors simulated for each restaurant and Room
- Temperature and humidity levels at kitchen and dining room
- Updated restaurants contexts with every new value through notifications

Cygnus

- Sink for MySQL for data persistance
- Sink for CKAN
- Now also sink for Hadoop (at Cosmos instance)

Security GEs

- Based in three different GEs:
  - Keyrock:
    - users, roles and permissions management
    - Oauth2 authentication
  - PEP Proxy
    - Protect resources
  - Access Control
    - Access policy based on XACML

REST API

- Protected endpoint /api/orion
- Authenticated requests
- Actions over Restaurants, reviews and reservations
- Adapted output to Schema.org