A complete IOT backend infrastructure in FIWARE

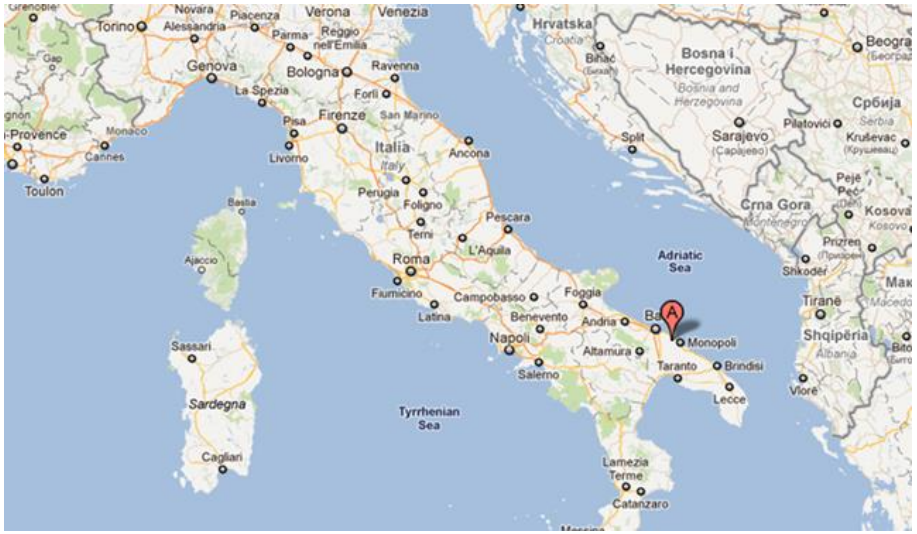
This talk will present the Tera solution about virtualization of services to provide a complete IOT backend solution in FIWARE Docker Container Service.

Original Solutions for Energy Efficiency

About us

Tera:

- ✓ Young knowledge-intensive SME,
- ✓ Launched in 2007 by 20-years working experienced persons;
- ✓ Fields of activity
  - energy efficiency;
  - renewable energy;
  - environmental monitoring;
  - predictive maintenance
  - ICT;
  - mechatronic.



CONFINDUSTRIA
Bari e Barletta-Andria-Trani

distretto PRODUTTIVO della MECCANICA pugliese

DESPUGLIA
DISTRETTO EDILIZIA SOSTENIBILE

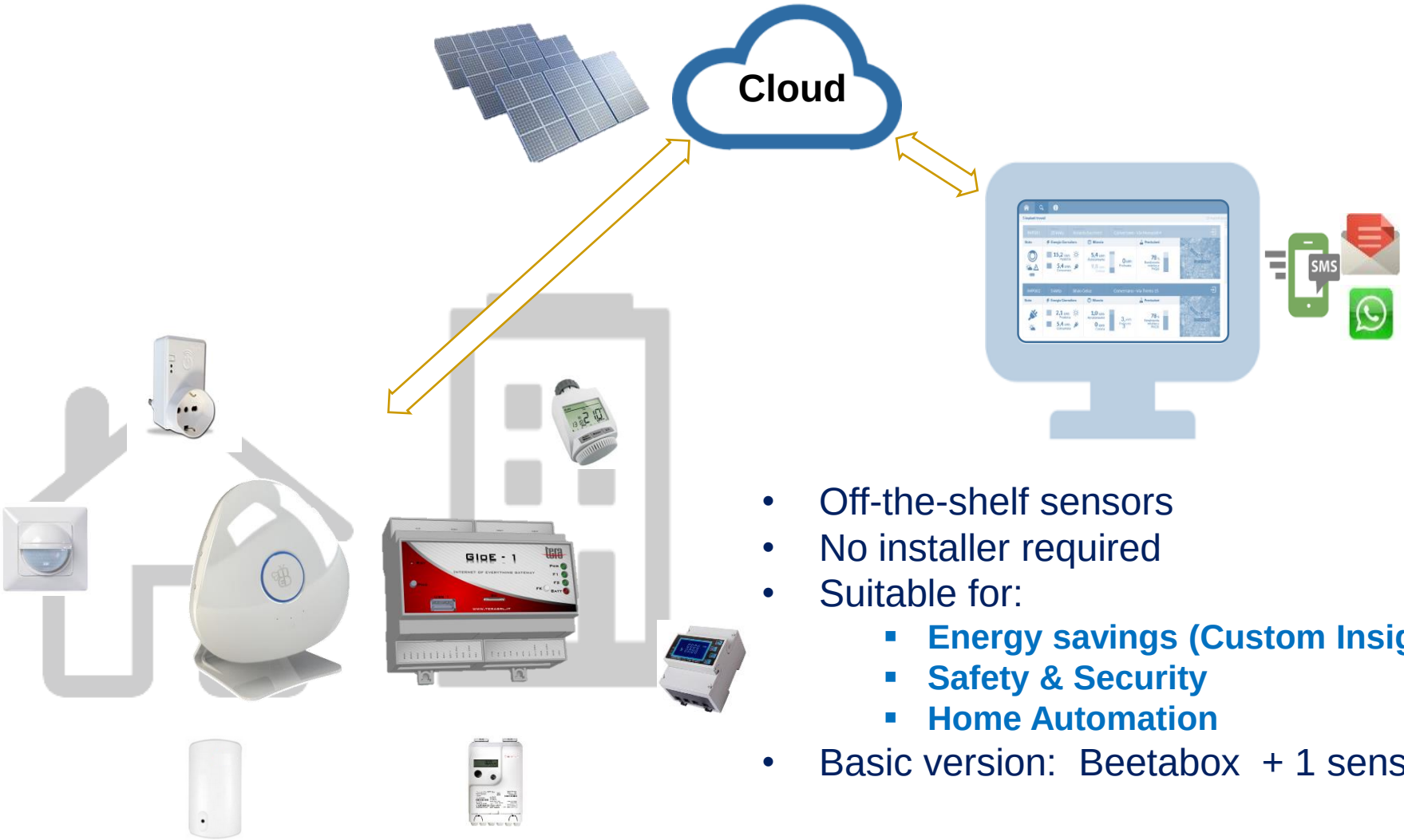
ALLCERT
Certification N. 03166
ISO9001:2008

DISTRETTO PRODUTTIVO PUGLIESE
LA NUOVA ENERGIA

Energy Efficiency
Financial Institutions Group

tera

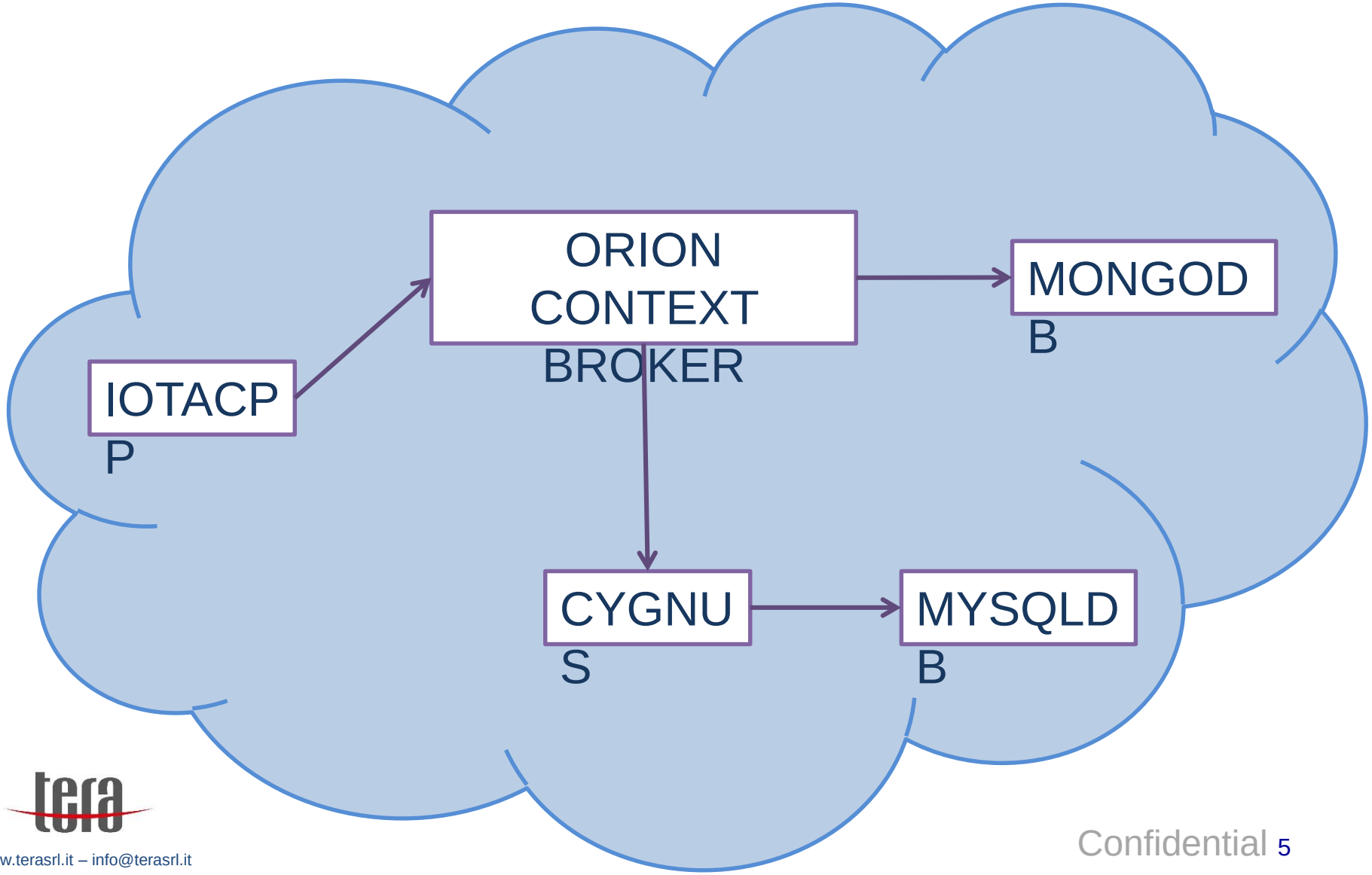
Tera solutions



- Off-the-shelf sensors
- No installer required
- Suitable for:
  - **Energy savings (Custom Insights)**
  - **Safety & Security**
  - **Home Automation**
- Basic version: Beetabox + 1 sensor

Confidential

Cloud solution



FDCS

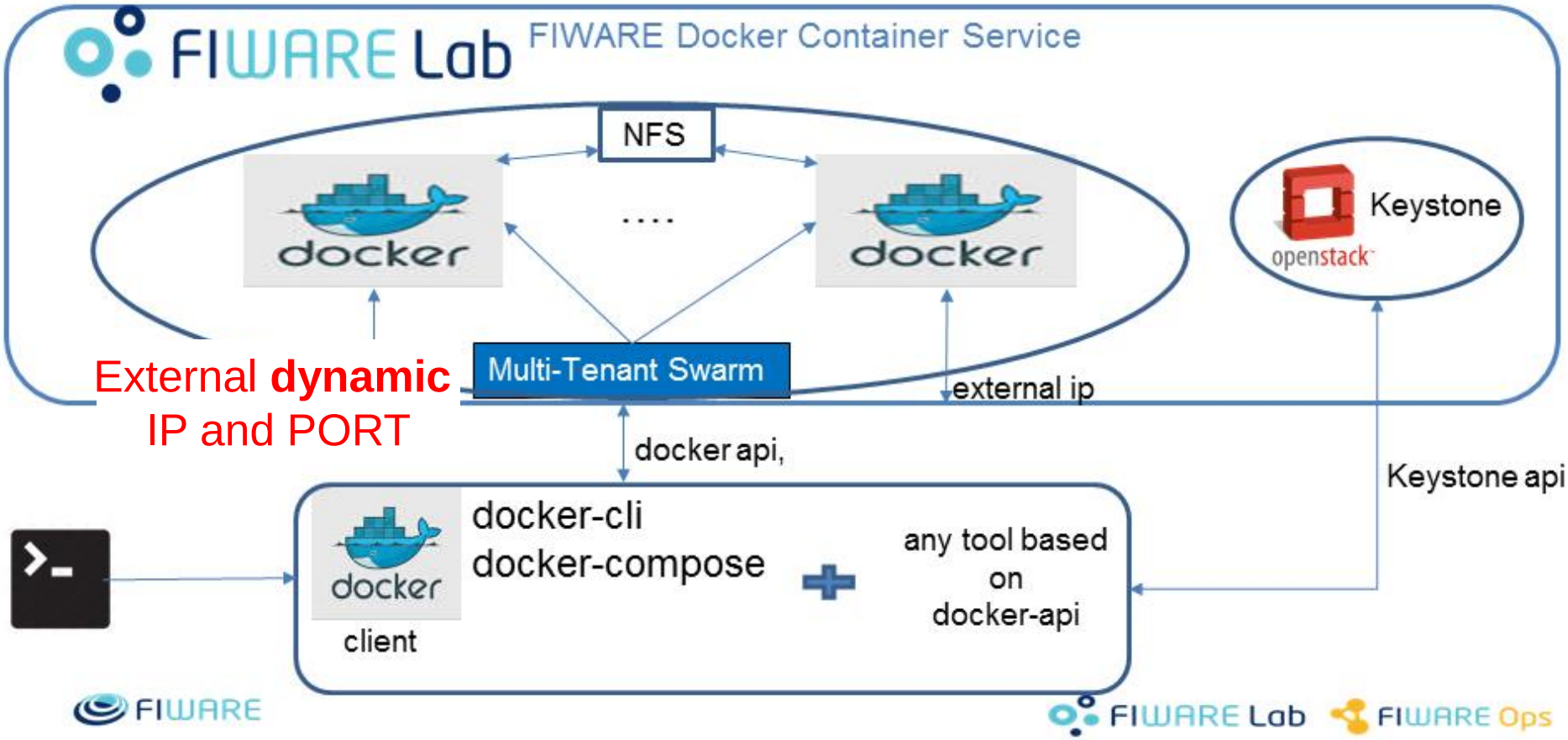
FIWARE Docker Container Service

Simple Docker hosting on FIWARE

Remotely Managed by Docker Client

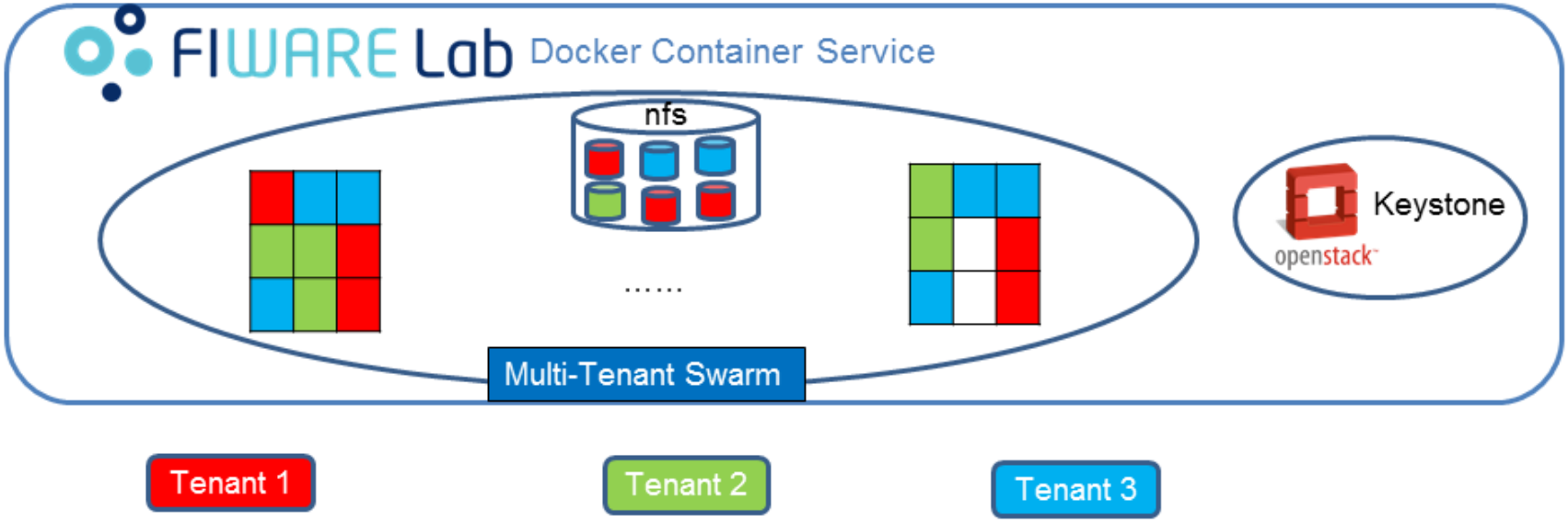
FDCS

FIWARE Docker Container Service



FDCS

FIWARE Docker Container Service



What we need

- Connect the differents modules each others
  - Know ip addresses and ports before building the image in some cases

Problem

Cygnus configuration file needs some parameters:

```
cygnus-ngsi.sinks.mysql-sink.type = com.telefonica.iot.cygnus.sinks.NGSIMySQLSink
cygnus-ngsi.sinks.mysql-sink.channel = mysql-channel
cygnus-ngsi.sinks.mysql-sink.enable_grouping = false
cygnus-ngsi.sinks.mysql-sink.enable_lowercase = false

cygnus-ngsi.sinks.mysql-sink.mysql_host = ??

cygnus-ngsi.sinks.mysql-sink.mysql_port = ??

cygnus-ngsi.sinks.mysql-sink.mysql_username = root
cygnus-ngsi.sinks.mysql-sink.mysql_password = admin
cygnus-ngsi.sinks.mysql-sink.attr_persistence = column
cygnus-ngsi.sinks.mysql-sink.data_model = dm-by-entity
cygnus-ngsi.sinks.mysql-sink.batch_size = 10
cygnus-ngsi.sinks.mysql-sink.batch_timeout = 5
cygnus-ngsi.sinks.mysql-sink.batch_ttl = 0
```

Solution

2 DIFFERENTS COMPOSE INTERCONNECTED

MYSQL
IOTACPP
ORION
MONGODB

DOCKER-COMPOSE A

```
mysql:
    image: <mysql-docker-image>
    ports:
      - "3306"
    expose:
      - "3306“
    volumes:
      - mysqldata:/var/lib/mysql
      - mysqlconf:/etc/mysql
mongodb:
    image: <mongodb-docker-image>
    ports:
      - "27017"
    expose:
      - "27017"
    volumes:
      - mongodata:/data/db
    command: --smallfiles
orion:
    image: <orion-docker-image>
    links:
      - mongodb
    ports:
      - "1026"
    expose:
      - "1026"
iotacpp:
    image: lprunella/iotagent-lino
    links:
        - mongodb
        - orion
    ports:
        ……..
```

Solution

2 DIFFERENTS COMPOSE INTERCONNECTED

MYSQL
IOTACPP
ORION
MONGODB

- cygnus-ngsi.sinks.mysql-sink.type = com.telefonica.iot.cygnus……
- cygnus-ngsi.sinks.mysql-sink.channel = mysql-channel
- cygnus-ngsi.sinks.mysql-sink.enable_grouping = false
- cygnus-ngsi.sinks.mysql-sink.enable_lowercase = false

cygnus-ngsi.sinks.mysql-sink.mysql_host = External Port
cygnus-ngsi.sinks.mysql-sink.mysql_port = External IP

- cygnus-ngsi.sinks.mysql-sink.mysql_username = root
- cygnus-ngsi.sinks.mysql-sink.mysql_password = admin
- cygnus-ngsi.sinks.mysql-sink.attr_persistence = column
- cygnus-ngsi.sinks.mysql-sink.data_model = dm-by-entity
- cygnus-ngsi.sinks.mysql-sink.batch_size = 10
- cygnus-ngsi.sinks.mysql-sink.batch_timeout = 5
- cygnus-ngsi.sinks.mysql-sink.batch_ttl = 0

Problem

There is no accessible NFS to store conf file and call it inside the Cygnus container

- cygnus-ngsi.sinks.mysql-sink.type = com.telefonica.iot.cygnus……
- cygnus-ngsi.sinks.mysql-sink.channel = mysql-channel
- cygnus-ngsi.sinks.mysql-sink.enable_grouping = false
- cygnus-ngsi.sinks.mysql-sink.enable_lowercase = false

cygnus-ngsi.sinks.mysql-sink.mysql_host = External Port
cygnus-ngsi.sinks.mysql-sink.mysql_port = External IP

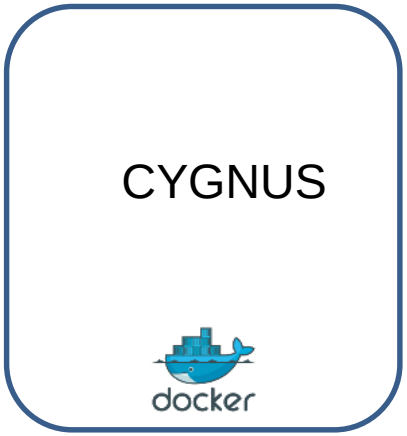
- cygnus-ngsi.sinks.mysql-sink.mysql_username = root
- cygnus-ngsi.sinks.mysql-sink.mysql_password = admin
- cygnus-ngsi.sinks.mysql-sink.attr_persistence = column
- cygnus-ngsi.sinks.mysql-sink.data_model = dm-by-entity
- cygnus-ngsi.sinks.mysql-sink.batch_size = 10
- cygnus-ngsi.sinks.mysql-sink.batch_timeout = 5
- cygnus-ngsi.sinks.mysql-sink.batch_ttl = 0

Solution

RUN WITH ENIVIRONMENT VARIABLES

DOCKER-COMPOSE B

```
cygnus:
    image: lprunella/cygnus-ngsi-fdcs3
    ports:
      - "5050"
      - "90"
    expose:
      - "5050"
      - "90"
    environment:
          - LOG_LEVEL=DEBUG
          -
CYGNUS_MYSQL_HOST=External Port
          -
CYGNUS_MYSQL_PORT=External IP
          - CYGNUS_MYSQL_PASS=
          - CYGNUS_MYSQL_USER=root
          -
CYGNUS_LOG_APPENDER=LOGFILE
```



Solution

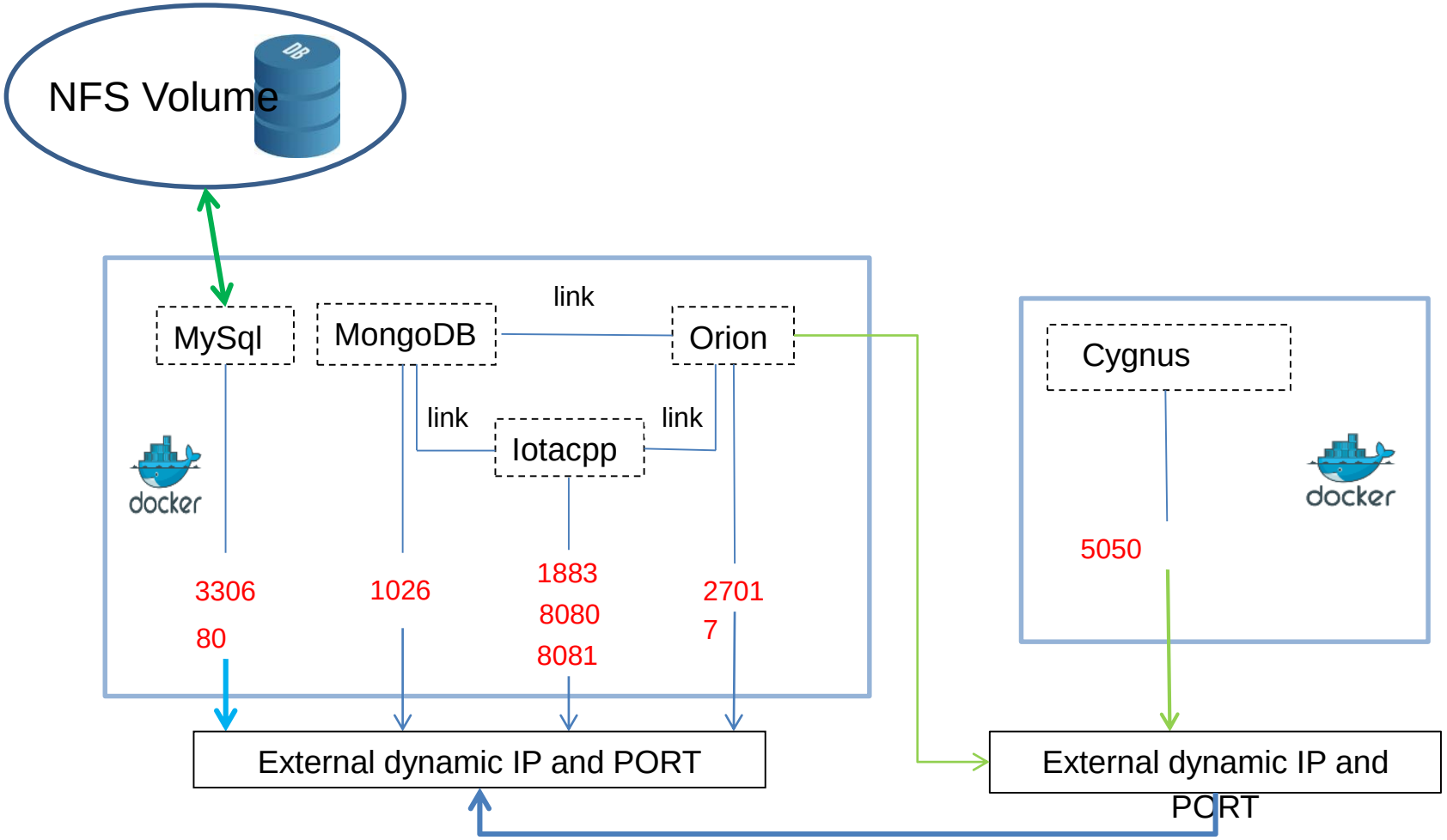
LAST STEP

Create subscription

```
(curl <orion-url>:<orion-port>/v1/subscribeContext -s -S --header 'Content-type: application/json' --header 'Accept: application/json' --header 'Fiware-Service: Library' --header 'Fiware-ServicePath: /catalog' -d @- | python -mjson.tool) <<EOF
{
    "entities": [
        {
            "type": "Book",
            "isPattern": "false",
            "id": "Book1"
        }
    ],
    "attributes": [
    ],
    "reference": "<cygnus-url>:<cygnus-port>/notify",
    "duration": "P1M",
    "notifyConditions": [
        {
            "type": "ONCHANGE",
            "condValues": [
                "title",
                "pages",
                "price"
            ]
        }
    ],
    "throttling": "PT5S"
}
EOF
```

OUR GOAL

Storicizing information with Cygnus on FDCS volume



SOLUTION STEPS

1. START DOCKER COMPOSE A

2. RETRIEVE MYSQL IP AND PORT

3. UPDATE DOCKER-COMPOSE FILE
WITH MYSQL IP AND PORT

4. START DOCKER COMPOSE B
configuration file
with fake msql_ho[illegible]. fake mysql_port

5. CYGNUS SUBSCRIPTION

info@terasrl.it
nicola.prunella@terasrl.it
antonio.lobefaro@terasrl.it

www.terasrl.it

https://it.linkedin.com/in/TERAsrl

https://twitter.com/terasrl