FutureGateway

A new multi-infrastructure framework for customisable Science Gateways

Tomasz Zok [1,2], Marco Fargetta [3], Riccardo Bruno [3], Roberto Barbera [3], Marcin Plociennik [1], Michal Owsiak [1], Michal Urbaniak [1], Giovanni Aloisio [4], Sandro Fiore [4]

[1] Poznan Supercomputing and Networking Center
[2] Poznan University of Technology
[3] Istituto Nazionale di Fisica Nucleare
[4] Euro-Mediterranean Center on Climate Change

2017-06-19

Outline

Horizon 2020 Programme

Introduction

Horizon 2020
Programme

Motivation

- Research requires an ever-increasing amount of both computational power and storage space
- Distributed Computing Infrastructures (DCIs) is a solution, but ...
- ... it brings its own set of problems – users need to learn to operate DCIs and keep up with changes and novel technologies
- Science Gateway is an answer to that problem.

Horizon 2020
Programme

Design Principles

- Easier installation and maintenance
  - Ready to use installation scripts and Ansible role
  - Public open source project at GitHub
  - Configurable and customizable
- Flexible access to DCIs
  - Plugin modules using JSaga
  - PaaS access through TOSCA
- RESTful API
  - Well known standard for many programming languages
  - Supports both desktop and mobile applications
  - Hides complexity of back-end operations

Horizon 2020
Programme

FutureGateway

Horizon 2020
Programme

Main Entities

Application	Definition of activity to be done
Infrastructure	Environment where an application can run
Task	An instance of an application bound to specific infrastructure

Horizon 2020
Programme

Download
API Blueprint

Fork on GitHub
FutureGateway/FutureGateway-APIs

INTRODUCTION

REFERENCE

FutureGateway API Root

v1.0

- Task Collection
- Task
- Application Collection
- Application
- Infrastructure Collection
- Infrastructure
- Roles Collection

View a Tasks Details

Retrieves the details of the specified task.

Create a new task

This method will create a new task with a specific `ID` assigned by the user. If the `id` already exist the task is not modified but an error is returned to the user because the tasks are not modifiable but only some parameters which have specific APIs for their update.

Modify a task

This method will modify the task with a specific `ID` assigned by the user. Currently only the status can be modified and the only value accepted is `CANCELLED`. This has the effect to stop the task, free the associated resources and clean the temporary storage.

Delete a task

Application Collection

Switch to Console

v1.0 / Task / View a Tasks Details

`GET` https://private-anon-8404e373ae-csgfapis.apiary-mock.com/v1.0/tasks/task_id

Parameters

task_id ● ID of the Task Example: `1234`. String

Request

Auth

Mock Server | Raw | Try

HEADERS

```
Authorization:Bearer {access_token}
```

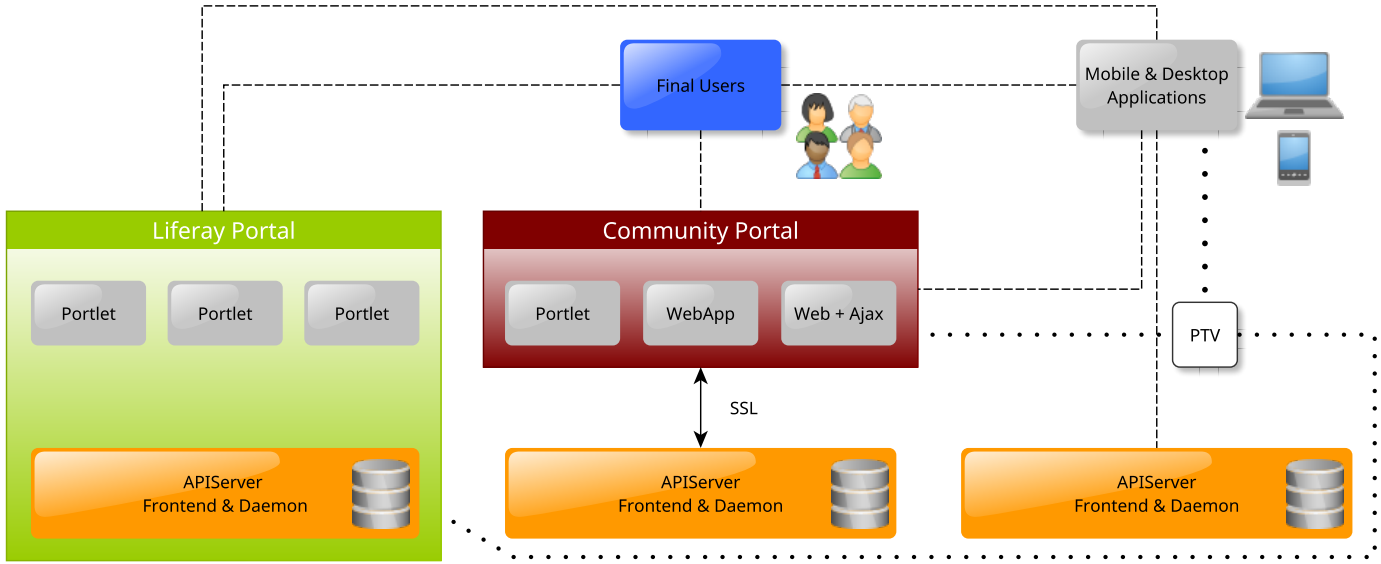
Response

200

HEADERS

```
Content-Type:application/vnd.indigo-datacloud.apiserver+json
```

Components



Access to FutureGateway is customizable:

- through a full portal experience or
- via separate interfaces to frontend and backend

INDIGO - DataCloud

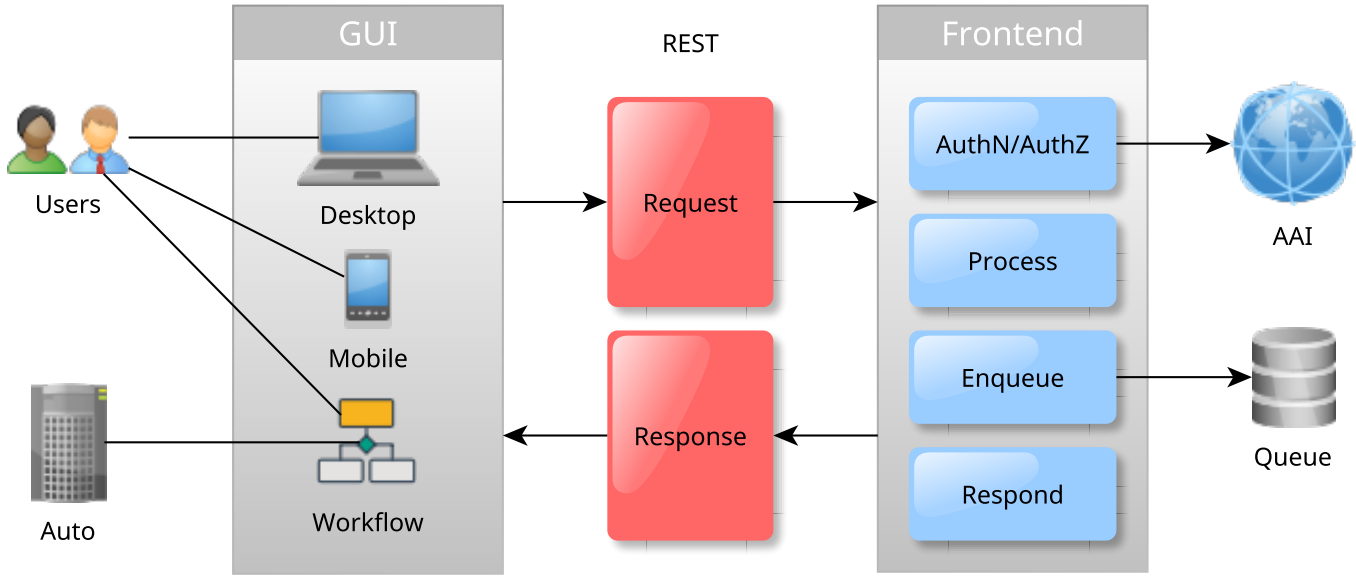
Horizon 2020 Programme

Components

Database	Maintains information about Applications, Infrastructures and Task, but also about the queue and user roles and groups.
APIServer Frontend	Fulfills RESTful API communication. Manages AuthN/AuthZ and definitions of Applications, Infrastructures and Tasks. Accepts and queues new Tasks.
APIServer Daemon	Polls for new Tasks. Interfaces concrete DCIs to perform actions consistently. Retrieves output. Extendable for new DCIs.

Horizon 2020 Programme

Frontend



- GUI sends a REST request
- Frontend checks with Authentication and Authorization Infrastructure (AAI)
- Actions to be done are queued
- Response is prepared

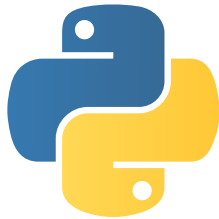
INDIGO - DataCloud

Horizon 2020 Programme

Frontend

- Available on GitHub: `https://github.com/indigo-dc/fgAPIServer`
- Written in Python using Flask microframework `http://flask.pocoo.org`
- Uses MySQL database
- Listens to REST calls compliant with documentation
- May run standalone or as a WSGI application (e.g. Apache)

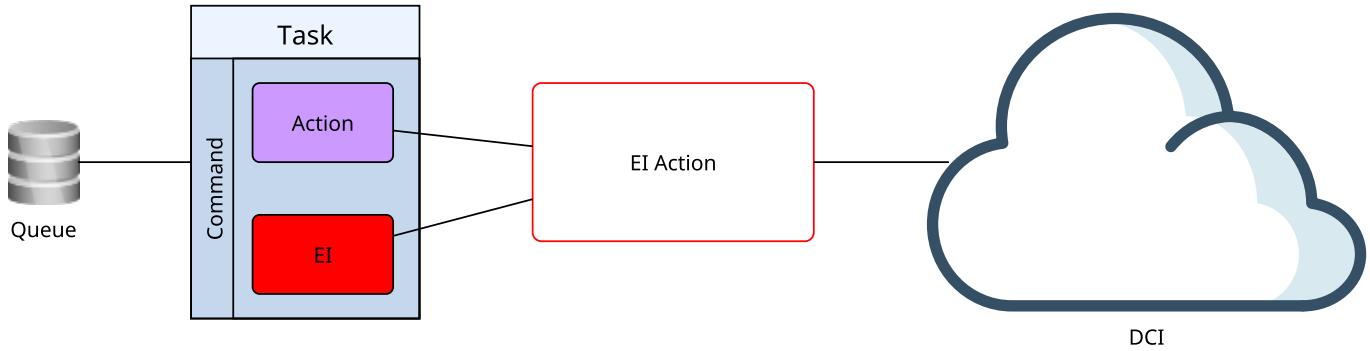
python



powered

Horizon 2020
Programme

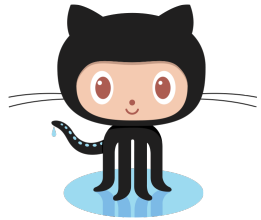
Backend

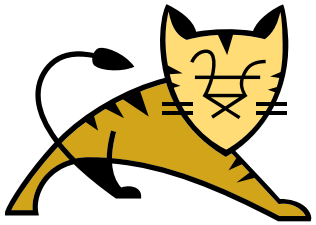


- Tasks are extracted from the queue
- Each command specifies Executor Interface (EI) and action
- Executor is dynamically instantiated to perform action on a DCI

Backend

- Available on GitHub: `https://github.com/indigo-dc/APIServerDaemon`
- Java web application running on top of Apache Tomcat
- Polls over the queue table in DB
- Interacts with DCIs
- Checks for consistency and resubmits failed tasks





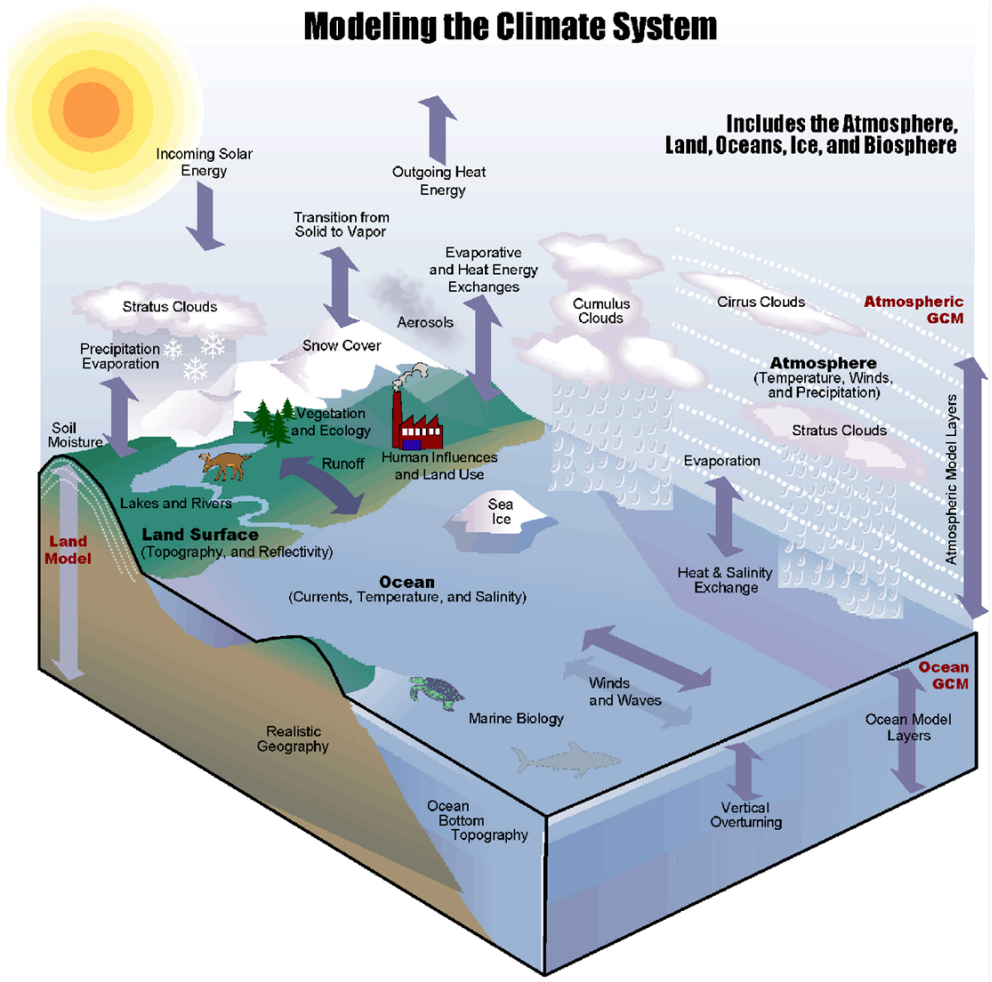
Horizon 2020 Programme

Climate Modeling

Horizon 2020
Programme

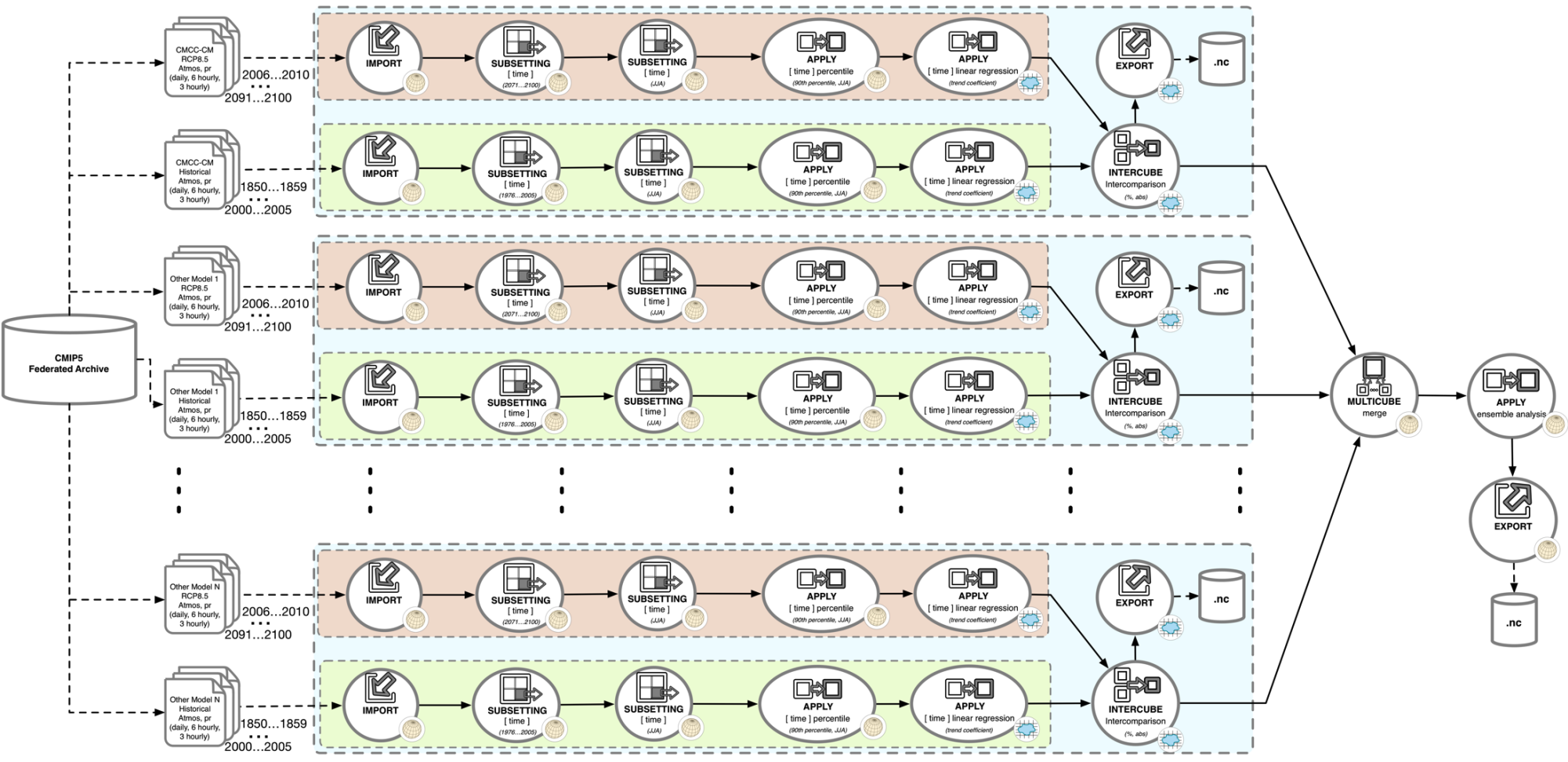
Introduction

- Complex processes
- Different time and spatial scales
- Largely interdisciplinary
- Inherently non-linear
- Huge computational resources and huge data volumes



INDIGO - DataCloud

Horizon 2020 Programme


CMIP5
Federated Archive
CMCC-CM
RCP8.5
Atmos, pr
(daily, 6 hourly, 3 hourly)
2006…2010
2091…2100
CMCC-CM
Historical
Atmos, pr
(daily, 6 hourly, 3 hourly)
1850…1859
2000…2005
Other Model 1
RCP8.5
Atmos, pr
(daily, 6 hourly, 3 hourly)
2006…2010
2091…2100
Other Model 1
Historical
Atmos, pr
(daily, 6 hourly, 3 hourly)
1850…1859
2000…2005
Other Model N
RCP8.5
Atmos, pr
(daily, 6 hourly, 3 hourly)
2006…2010
2091…2100
Other Model N
Historical
Atmos, pr
(daily, 6 hourly, 3 hourly)
1850…1859
2000…2005
IMPORT
SUBSETTING
[time]
(2071…2100)
SUBSETTING
[time]
(JJA)
APPLY
[time] percentile
(90th percentile, JJA)
APPLY
[time] linear regression
(trend coefficient)
EXPORT
.nc
SUBSETTING
[time]
(1976…2005)
INTERCUBE
Intercomparison
(%, abs)
MULTICUBE
merge
APPLY
ensemble analysis
EXPORT
.nc

INDIGO DataCloud

- Development of open source data and computing platform targeted at scientific communities
- Deployable on various software stacks provisioned over public, private or hybrid clouds

INDIGO DataCloud is a complete solution with compononents for:

- AuthN / AuthZ
- computing and storage resources management
- dynamic instantiation and configuration of VMs and containers
- ranking & selection of providers
- end-user interfaces

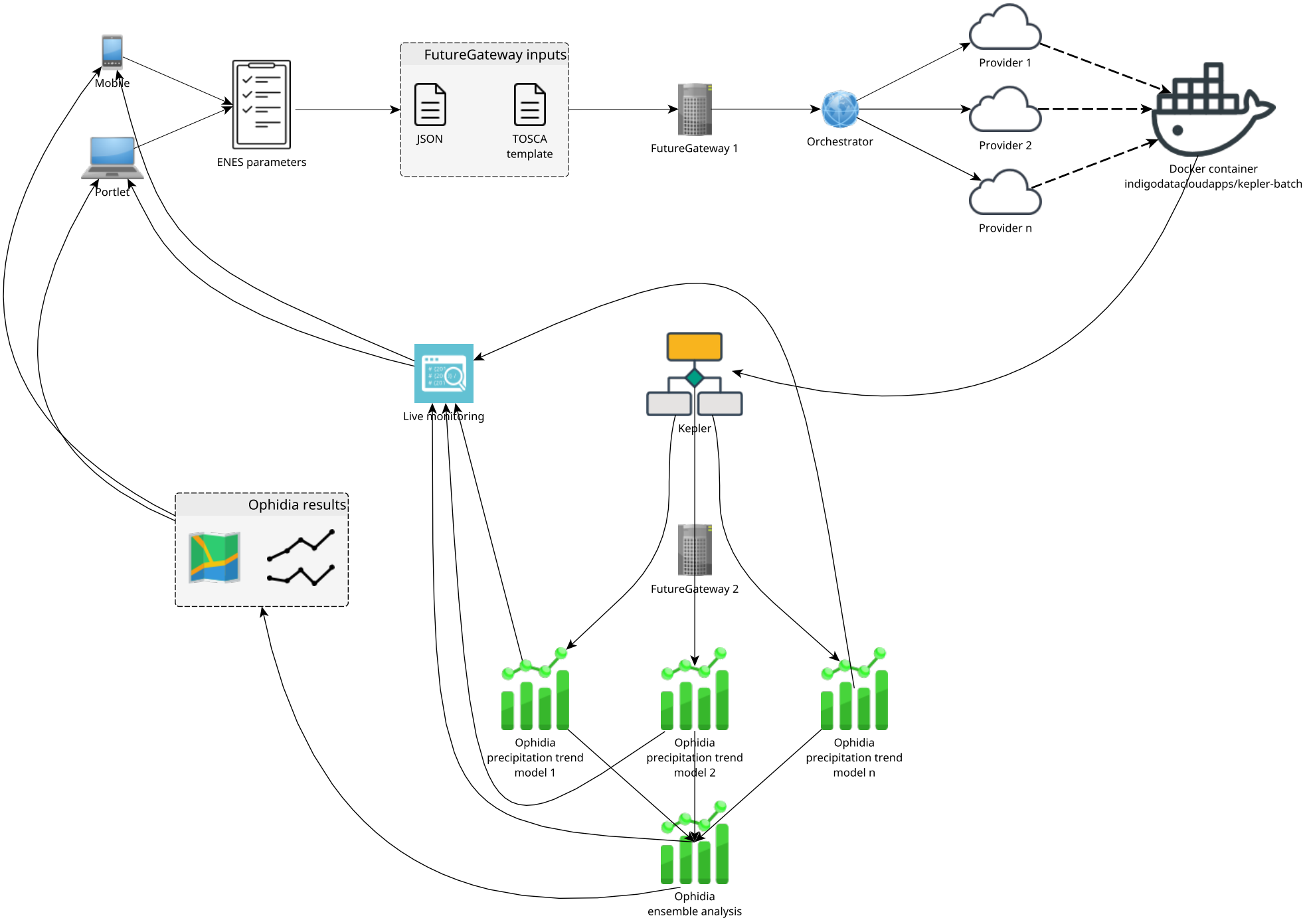
Horizon 2020
Programme

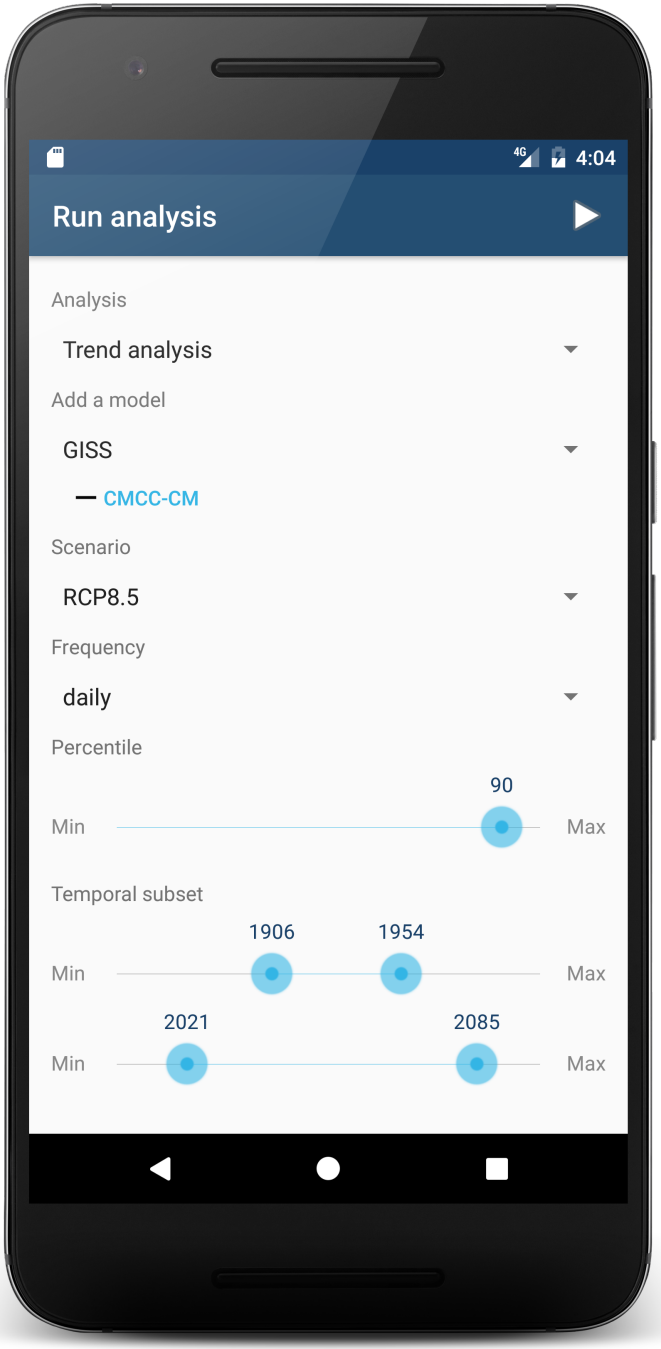
Solution Design

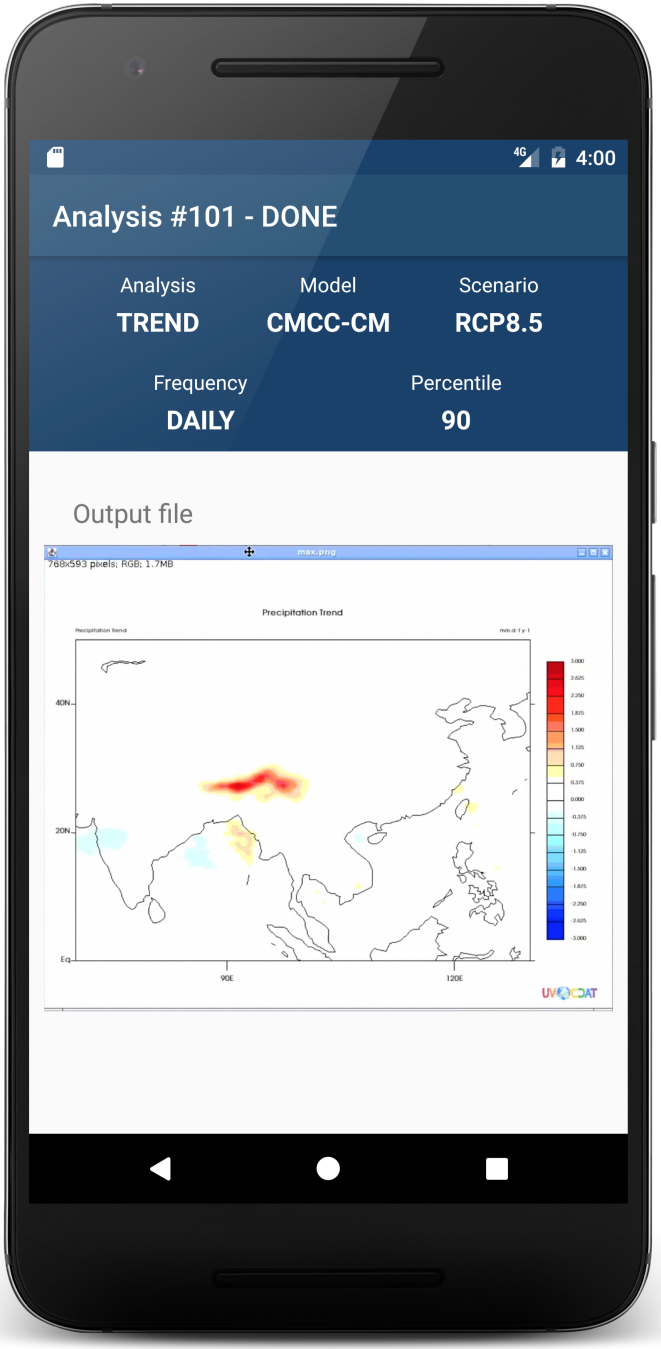
- FutureGateway in PSNC orchestrates whole process
- Computing nodes in CMCC, ORNL and LLNL
- Each node gets data to process
- Once finished, an ensemble analysis is done

INDIGO - DataCloud

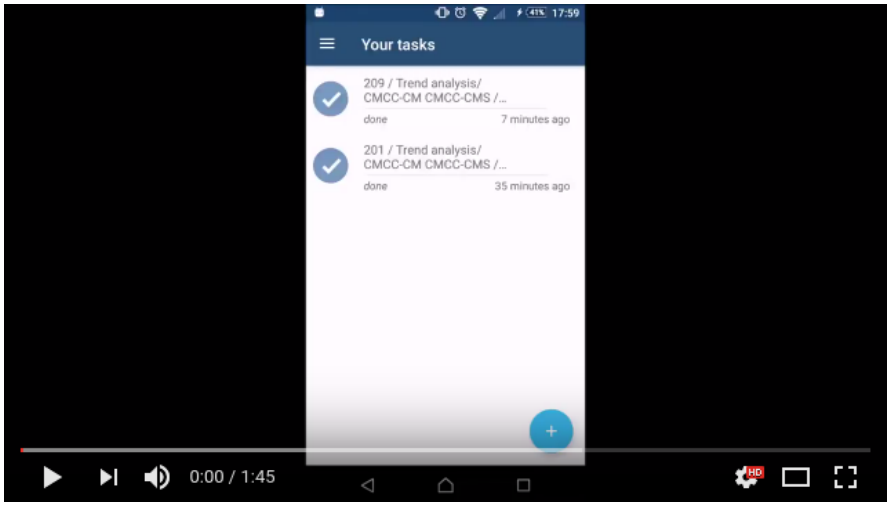
Horizon 2020 Programme

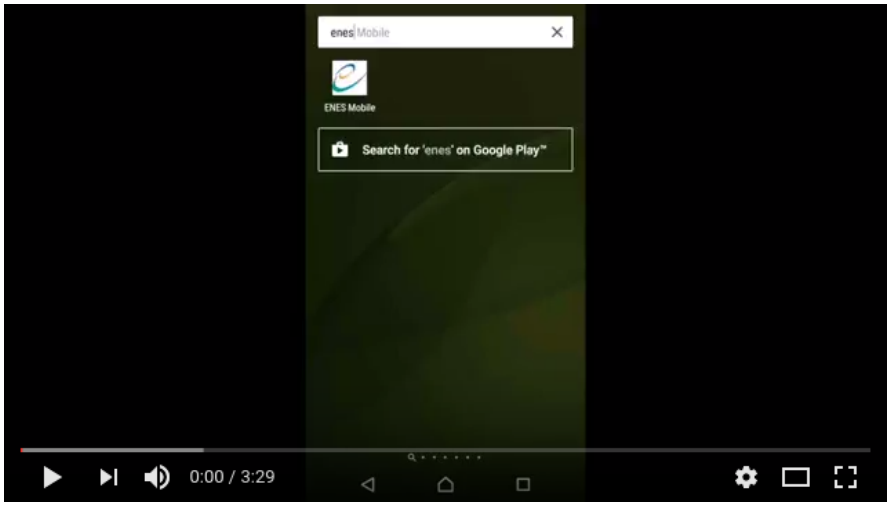

Mobile
Portlet
ENES parameters
FutureGateway inputs
JSON
TOSCA template
FutureGateway 1
Orchestrator
Provider 1
Provider 2
Provider n
Docker container
indigodatacloudapps/kepler-batch
Live monitoring
Kepler
Ophidia results
FutureGateway 2
Ophidia precipitation trend model 1
Ophidia precipitation trend model 2
Ophidia precipitation trend model n
Ophidia ensemble analysis


4:04
Run analysis
Analysis
Trend analysis
Add a model
GISS
CMCC-CM
Scenario
RCP8.5
Frequency
daily
Percentile
90
Min
Max
Temporal subset
1906
1954
Min
Max
2021
2085
Min
Max


4:00
Analysis #101 - DONE
Analysis
TREND
Model
CMCC-CM
Scenario
RCP8.5
Frequency
DAILY
Percentile
90
Output file
Precipitation Trend

Demo





INDIGO - DataCloud

Horizon 2020 Programme