



# Importing a new application into the Virtual Imaging Platform: a short demo

Sorina Pop, Axel Bonnet, Frederic Cervenansky



# Demo Outline

- Short VIP overview
- Creating a docker image
- Boutiques application descriptor
- Importing the application into VIP
- Wrap-up

### Web portal

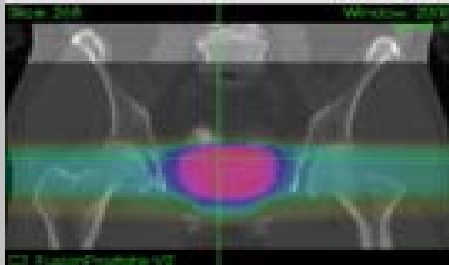
### Application as a service

File transfer to/from grid



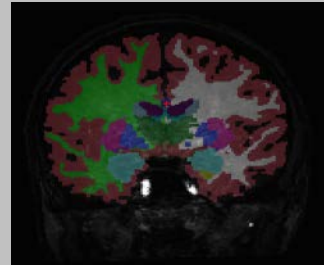
### Scientific applications

#### Cancer therapy simulation



Prostate radiotherapy plan simulated with GATE (L. Grevillot and D. Sarrut)

#### Neuro-image analysis



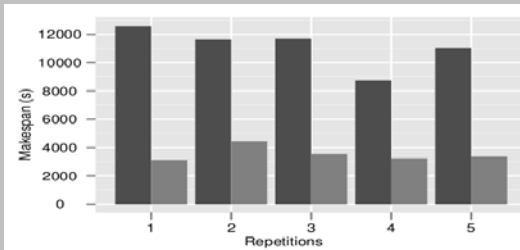
Brain tissue segmentation with Freesurfer

#### Image simulation



Echocardiography simulated with FIELD-II (O. Bernard *et al*)

#### Modeling and optimization of distributed computing systems



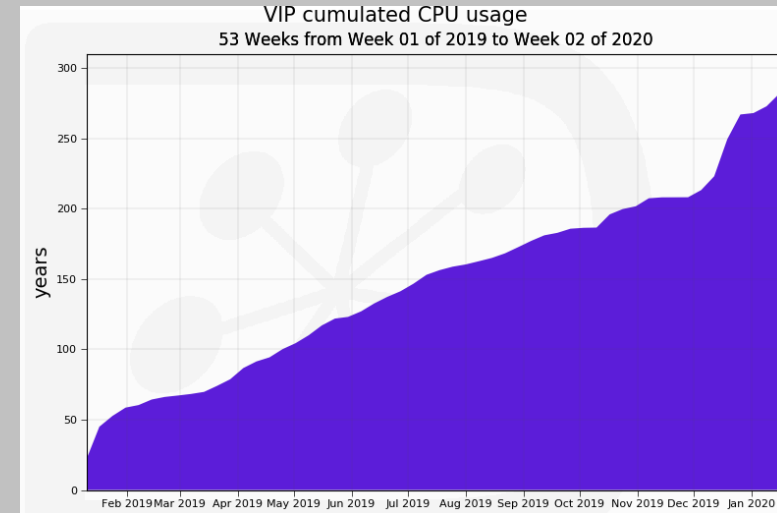
Acceleration yielded by non-clairvoyant task replication (R. Ferreira da Silva *et al*)

### Infrastructure

Supported by EGI Infrastructure

Uses biomed VO (~65 sites in Europe and beyond)

VIP consumes ~23 CPU years every month



France-Grilles



### Users

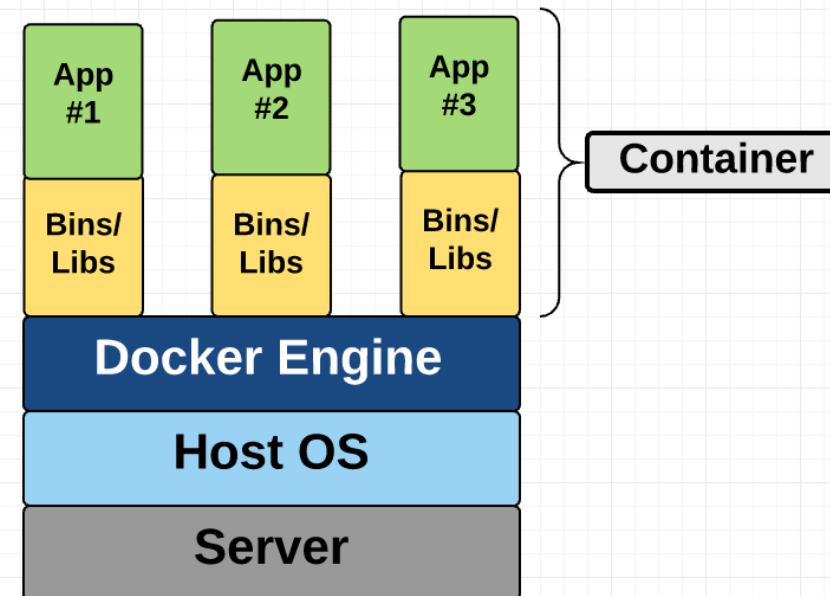
1220 registered users in November 2020

61 publications since 2011



# Containers

- A container = an entire runtime environment
  - An application + all its dependencies, libraries and other binaries, and configuration files needed to run it, bundled into one package
  - Differences in OS distributions and infrastructures are abstracted away
- Popular container technologies
  - Docker and Singularity
- Hands-on
  - Build and push a container to Docker-hub



# Boutiques



- Describe, publish, integrate and execute applications **across platforms**
  - facilitate application porting
  - import and exchange of applications
  - Linux containers to facilitate application installation and sharing
- <https://github.com/boutiques>
  - <https://github.com/boutiques/boutiques/tree/master/schema>

## Findable

1. Globally persistent records
2. Described with rich metadata
3. Searchable

We leverage **Zenodo [2]** to create DOIs for Boutiques descriptors which can be accessed via the Zenodo API.

## Interoperable

1. Formalized and shared metadata standard
2. Metadata standards adopted are FAIR
3. Linking between objects where appropriate

**CARMIN [3]** and **Boutiques [4]** standards are used to describe and launch tools, either locally or through a RESTful API.

## Accessible

1. Easily retrievable
2. Universal access
3. Persistent metadata beyond data lifetime

The retrievable tool descriptions contain **immutable** human- and machine-readable instructions for testing and launching each tool.

## Re-Usable

1. Multiple accurate and relevant attributes
2. Clearly licensed
3. Meets minimum domain standards

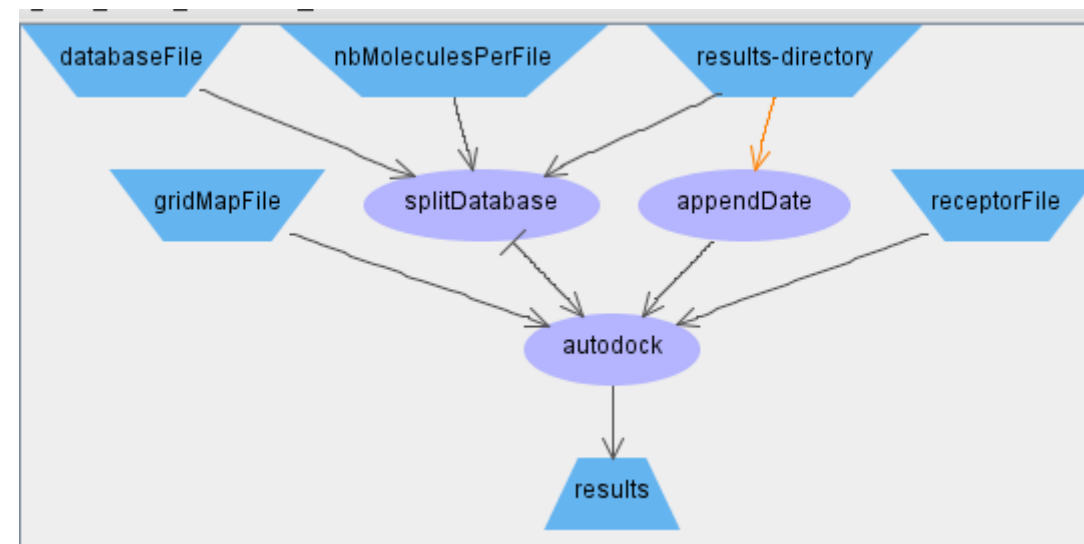
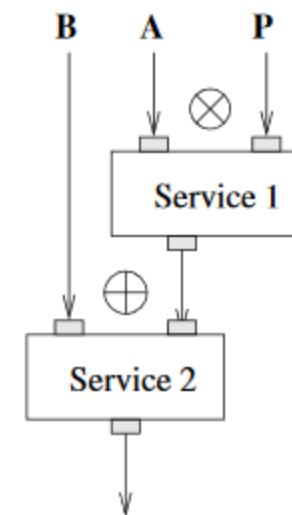
**Docker [5]** and **Singularity [6]** virtualization enable re-runability across platforms and enclosed testing. Simulation and querying allow runtime evaluation.

FAIR tools. Credits: Gregory Kiar and Tristan Glatard



# What about more complex workflows ?

- Describe individual applications with Boutiques
- Write the corresponding workflow
  - Gwendia language for the Moteur Workflow Engine
- Moteur Workflow Engine
  - Data oriented
  - Allows to easily express parallelization
  - Dot and cross product iteration strategies
  - <https://hal.archives-ouvertes.fr/hal-00691832/document>



VIP autodock workflow

# Wrap-up

- VIP offers scientific applications as a service
  - No need for installation on the users' side
- Integration of a new standalone application
  - Boutiques application descriptor
  - Build a container of your application
- Integrating a pipeline
  - Same as above +
  - Workflow to be written with help from VIP team

**THANK YOU FOR YOUR ATTENTION!**

