

Porting Your Application to VIP

Pascal Girard¹, Sorina Pop¹, Axel Bonnet¹, Frédéric Cervenansky¹, Tristan Glatard²

¹CREATIS; CNRS (UMR 5220); INSERM (U1206); INSA Lyon; Université de Lyon, France,

²Concordia University, Dept of Computer Science and Software Engineering, Montreal.

Application import principles in VIP

- Provide to VIP the application (script, binary (executable), etc.)
 - And execution context:
 - librairies
 - called scripts
 - OS
 - => **Docker technology**
- Explain to VIP how to launch the application
 - Description of:
 - application
 - inputs
 - outputs
 - => **JSON descriptor (Boutiques specifications)**

2 steps before import in VIP

- Bundle the **implementation** (executable) of your application

Docker container
<http://www.docker.com>

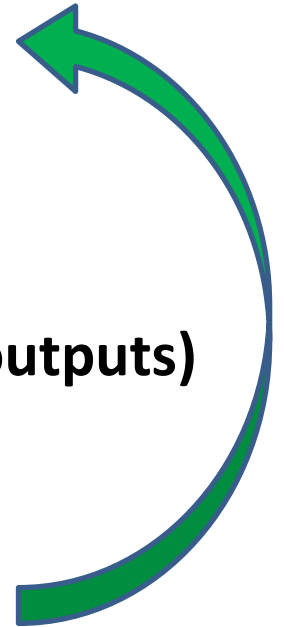


- Describe the **interface** of your application (inputs / outputs)

JSON descriptor
<http://boutiques.github.io>



references



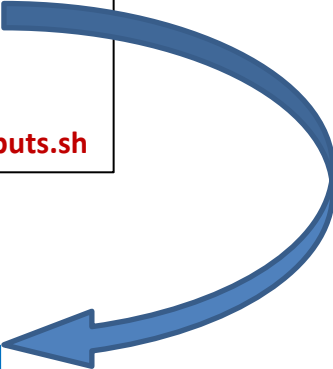
Docker technology

- Technology to bundle your **application** and all its **dependencies**
- On docker site :
 - “Everything required to make a software run is packaged into isolated containers”
 - “Unlike VMs, containers do not bundle a full OS”
 - Efficient, lightweight, self-contained systems
 - and guarantees that software will always run the same, regardless of where it’s deployed
- TODO : build a docker image
 - Definitions : a **docker container** = a running **docker image**

How to build a docker image ?

- My application & its dependencies

```
-rw-rw-r-- 1 pgirard pgirard 245 Jun 22 15:10 Dockerfile
-rw-rw-r-- 1 pgirard pgirard 31 Jun 22 15:09 final_file.txt
-rw-rw-r-- 1 pgirard pgirard 39 Jun 22 15:09 initial_file.txt
-rw-rw-r-- 1 pgirard pgirard 362 Jun 22 15:09 script_optional_inputs.sh
```



- Dockerfile** = text file

```
FROM centos:latest

RUN yum -y update

COPY ./script_optional_inputs.sh script_optional_inputs.sh
COPY ./initial_file.txt initial_file.txt
COPY ./final_file.txt final_file.txt

RUN chmod 777 script_optional_inputs.sh

WORKDIR /

ENV PATH=$PATH:/" data-bbox="308 532 684 875"/>
```

How to build a docker image ?

- Docker build

```
[pgirard@newk docker_demo]$ docker build -t pascalgirard/optional2803:v2 .
Sending build context to Docker daemon 5.12 kB
Step 1/8 : FROM centos:latest
latest: Pulling from library/centos
d5e46245fe40: Pull complete
Status: Downloaded newer image for centos:latest
---> 3bee3060bfc8
Step 2/8 : RUN yum -y update
...
Step 8/8 : ENV PATH $PATH:/
---> Running in 441daabc55b7
---> e14c5ec802eb
Removing intermediate container 441daabc55b7
Successfully built e14c5ec802eb
[pgirard@newk docker_demo]$

[pgirard@newk docker_demo]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
pascalgirard/optional2803	v2	e14c5ec802eb	2 minutes ago	318 MB

```
docker run -t -i pascalgirard/optional2803:v2 /bin/bash
```

```
docker push pascalgirard/optional2803:v2
```

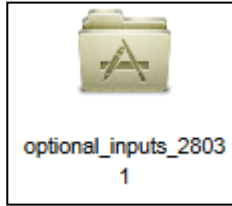
JSON descriptor (Boutiques specifications)

- Boutiques
 - = specifications to describe your **application command line**
 - executable + inputs + outputs
- Relies on a JSON descriptor
 - = text file
 - contains **keys** and associated **values**
 - **JSON descriptor describes command line**
 - **Tells VIP how to launch the application**

```
-rwxrwxrwx 1 pgirard pgirard 362 Jun 22 15:09 script_optional_inputs.sh
```

- **script_optional_inputs.sh** initial_file.txt final_file.txt -v ch1 ch2

JSON example



optional_inputs_2803 1

Documentation and Terms of Use

Execution Name*

Results directory*
Directory where the results will be stored.

List

Initial text file*
File containing text. The file content will be copied in the file name final_file.txt

List

Copy text flag (-v)
The flag is present on the command line at the launch of application, it will be mentioned in final_file.txt file

Copy line below
Copy the line text in final_file.txt file

List

Add text
Copy the line text in final_file.txt file

List

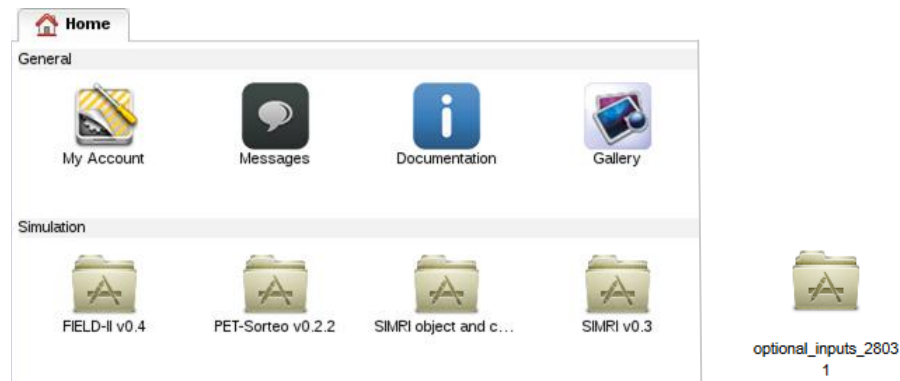
```

"name": "optional_inputs_2803",
"tool-version": "1",
"description": "Allow to test optional inputs and checkbox",
"command-line": "script_optional_inputs.sh [INITIAL_FILE] [FINAL_FILE] [FLAG] [OPTIONAL_INP_1]",
"docker-image": "pascalgirard/optional2803:0.2",
"docker-index": "http://index.docker.io",
"inputs": [
  {
    "id": "initial_file",
    "name": "Initial text file",
    "type": "File",
    "description": "File containing text. The file content will be copied in the file name .....",
    "command-line-key": "[INITIAL_FILE]",
    "optional": false,
  },
  {
    "id": "flag_for_copy",
    "name": "Copy text flag",
    "type": "Flag",
    "description": "The flag is present on the command line at the launch of application, .....",
    "command-line-key": "[FLAG]",
    "optional": true,
    "command-line-flag": "-v",
  },
  {
    "id": "optional_input_1",
    "name": "Copy line below",
    "type": "String",
    "description": "Copy the line text in final_file.txt file",
    "command-line-key": "[OPTIONAL_INP_1]",
    "optional": true,
  },
]
"output-files": [
  {
    "id": "final_file",
    "name": "Final file",
    "type": "File",
    "description": "File which will receive text from initial_file.txt AND if a flag is put in the .....",
    "command-line-key": "[FINAL_FILE]",
    "path-template": "final_file.txt",
    "optional": false,
  }
]

```


My application : use

- 2 ways to use my application
 - **VIP IHM**



- **API** : see Axel's presentation

Lessons learnt and perspectives

- **Robust** application import mechanism (Docker image + JSON)
- Challenge MICCAI 2016 : **VIP, the execution platform**
 - 3 months : 23 applications imported for 15 challengers
 - Mass import of pipelines
 - (Docker) containers : a must for pipeline integration at a large-scale
- Perspectives
 - Improve VIP to support schema V4 of Boutiques