VIP : towards data platforms interoperability through CARMIN

Axel Bonnet ¹
Sorina Pop ¹
Frédéric Cervenansky ¹
Pascal Wassong ¹
Tristan Glatard ²

¹ CREATIS; CNRS (UMR 5220); INSERM (U1206); INSA Lyon; Université de Lyon, France
² Concordia University, Quebec, Canada
Overview

- VIP usage and limitations
- Need for data interoperability
- Issues to overcome
- Conclusion
The Virtual Imaging Platform

- Cancer therapy simulation
  - Prostate radiotherapy plan simulated with GATE (L. Grevillot and D. Sarrut)
- Neuro-image analysis
  - Brain tissue segmentation with Freesurfer

1000+ registered users in October 2019
55 publications identified since 2011

Supported by EGI Infrastructure
Uses biomed VO (~65 sites in Europe and beyond)
230 cumulated CPU years utilized by VIP applications in 1 year

VIP usage through the portal

1. Login
2.1 Send input data
3. Launch application

9.1 Download results
2.2 Transfer input files
9.2 Transfer output files

4. Launch workflow
5. Submit jobs

6. Get files
7. Execute
8. Upload results

Workflow Engine

VIP Web Portal

User

Storage Element

Computing site
VIP usage through the CARMIN API

1. Launch Application

2. Launch workflow
3. Submit jobs

4. Get files
5. Execute
6. Upload results
VIP usage through the CARMIN API

1. Launch Application

What about data?

2. Launch workflow
3. Submit jobs

4. Get files
5. Execute
6. Upload results
VIP usage through the CARMIN API

1. Launch Application
2. Launch workflow
3. Submit jobs
4. Get files
5. Execute
6. Upload results

What about data? -> Need data interoperability
Solution: access data where the user stores it

- User
  - Start application
  - Manage its data & Consult results
  - Direct transfers with the user storage solution

- External tool
- CARMIN Platform
- VIP
- Computing site

Solution: access data where the user stores it
Solution: access data where the user stores it

- Start application
- Manage its data & Consult results
- Direct transfers with the user storage solution

Is it that easy? 3 issues:
- How to communicate?
- How to authenticate?
- How to identify the input file?
1st issue: Communicate with the storage tool

- The storage supports CARMIN Data
1\textsuperscript{st} issue: Communicate with the storage tool

- The storage supports CARMIN Data

- Otherwise, a custom implementation is needed
2nd issue: Authenticate on the storage tool

- VIP needs to access the storage platform in the name of the user
- Credential delegation in 2 steps
  - an API key obtained from the storage tool is stored in VIP
  - The API key allows to get a temporary token for each execution
- Works for Girder (a data warehouse by Kitware)
- OpenID should work in theory (with refresh and access tokens)
3\textsuperscript{rd} issue: Identify the input file

- CARMIN extended to support external storage
  - External files as input and outputs
  - API keys configuration

```json
{
  "name": "JCAD Test",
  "pipelineIdentifier": "GrepTest/2.0",
  "inputValues": {
    "results-directory": "/vip/Home/out",
    "file": "brainPerfusionGirder:5d945bd3b848ce091e4849fc",
    "text": "mi"
  }
}
```
Conclusion

- Improved data interoperability in VIP
- Integrated into the CARMIN specification
- A step towards platforms networks