

Virtual Imaging Platform : pour une science ouverte et reproductible

Medical Imaging Research Laboratory

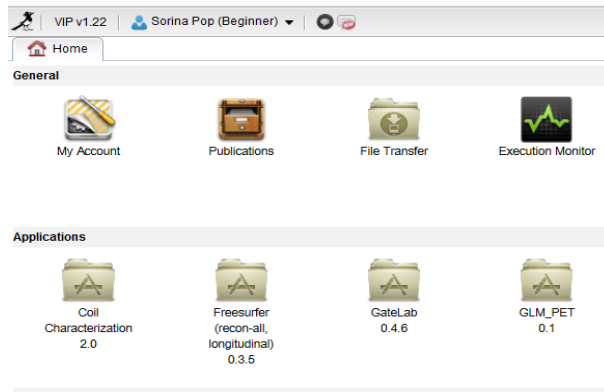
Sorina CAMARASU-POP
Frédéric CERVENANSKY

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

Journée ARAMIS « La reproductibilité en pratique : méthodes et outils »
Lyon 23/05/2019

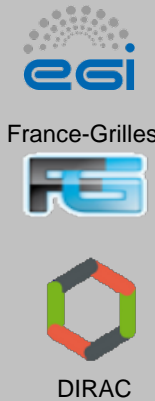
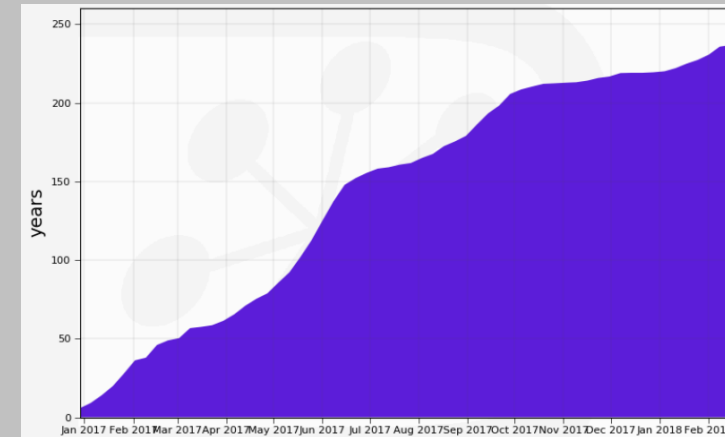
Outline

- The Virtual Imaging Platform
- Towards open and reproducible science
 - Boutiques and FAIR data analysis
 - Interoperability and CARMIN
- Organization of conference challenges
- Conclusions

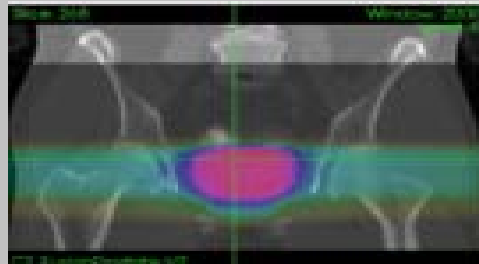


<https://vip.creatis.insa-lyon.fr>

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

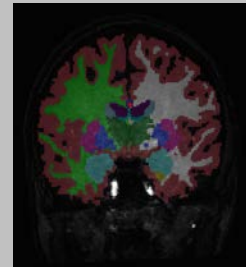


Cancer therapy simulation



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

Neuro-image analysis



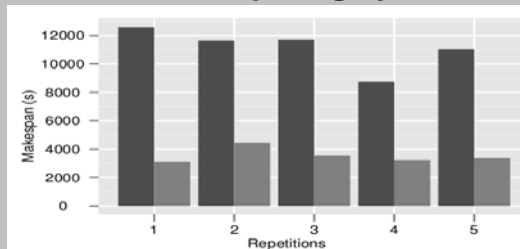
Brain tissue segmentation with Freesurfer

Image simulation



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

Modeling and optimization of distributed computing systems



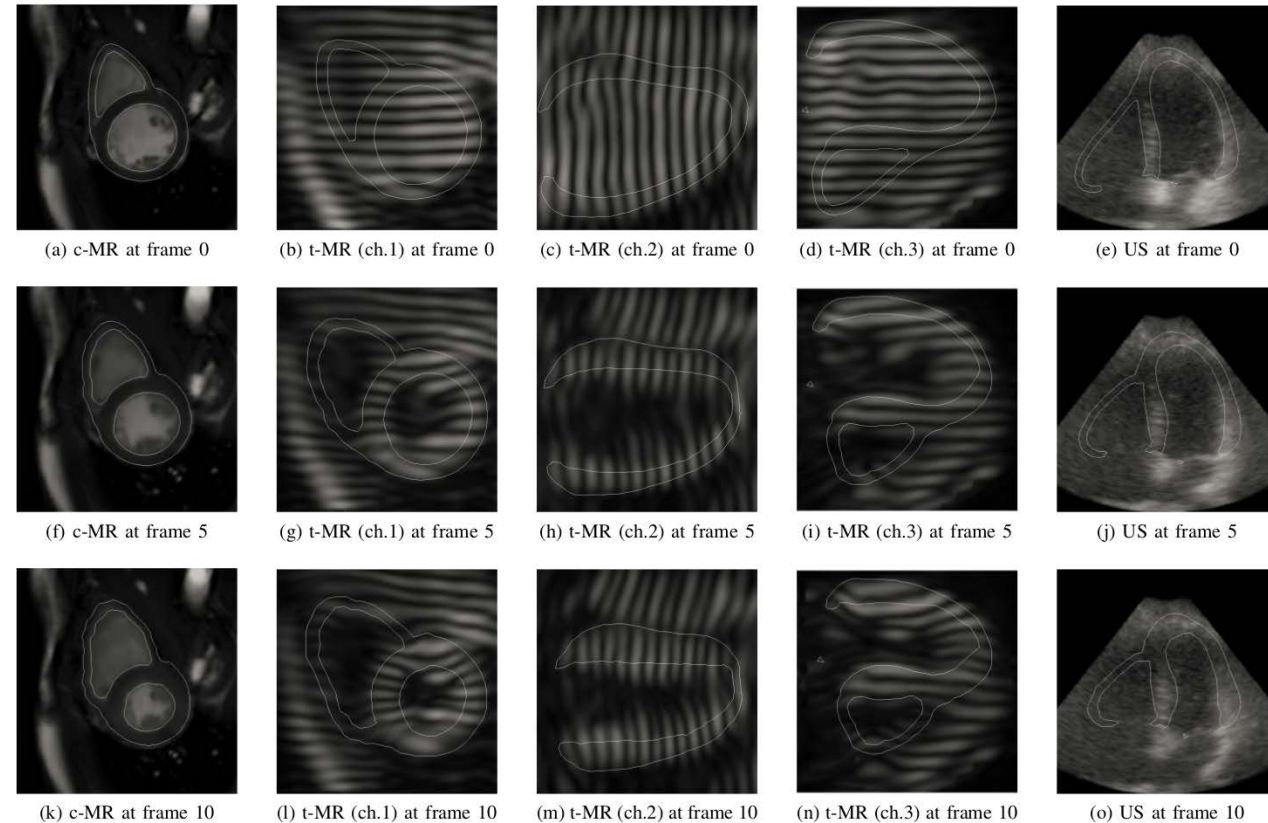
Acceleration yielded by non-clairvoyant task replication (R. Ferreira da Silva *et al*)

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

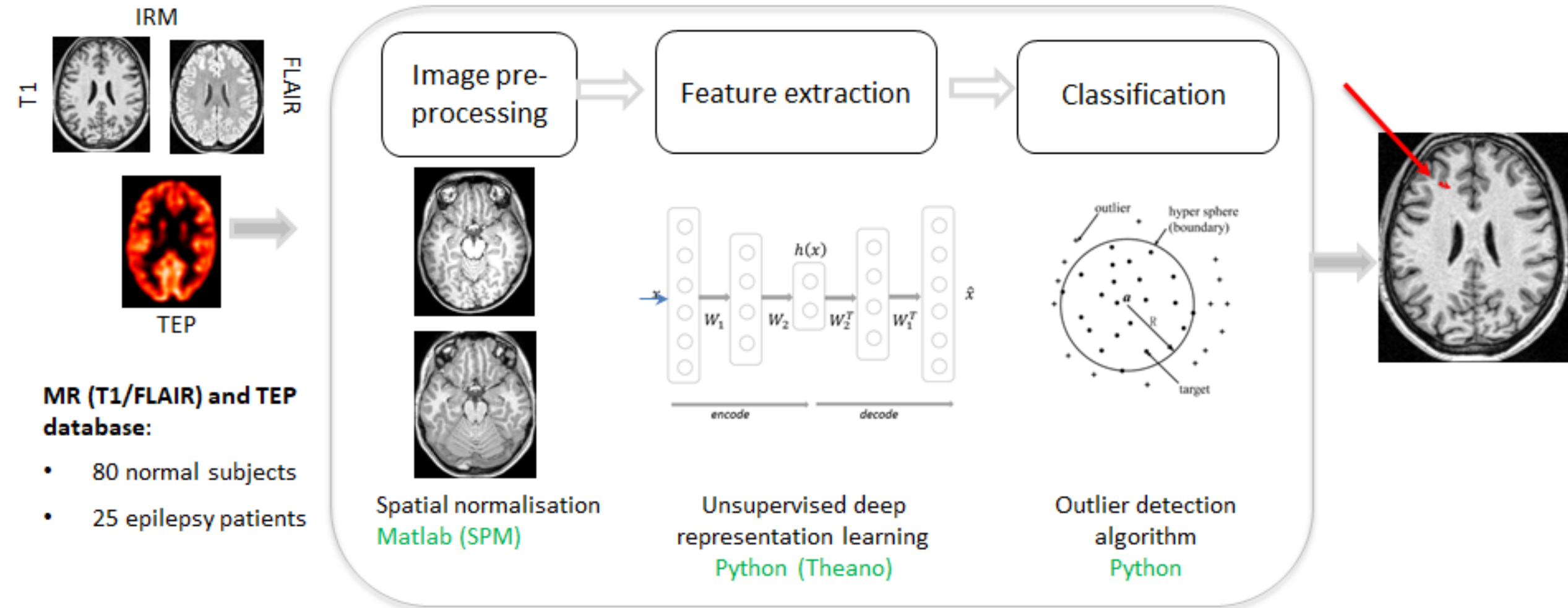


Example of a VIP use-case

- Generate virtual ultrasound and magnetic resonance images using VIP
 - Reference: Bernard et al. 2018. IEEE Transactions on Medical Imaging
 - EGI Research story
- MRI simulator integrated into VIP
- Catalogue of 18 virtual patients
 - 2700 image volumes
 - Benchmarked myocardial motion for validation purposes
- The generation of one full 3D sequence took 6 hours on VIP
 - 280 hours on a personal laptop



CAD for Epilepsy



VIP achievements and challenges

- Transparent access to distributed computing resources
 - VIP relies on the France Grilles DIRAC instance (<https://dirac.france-grilles.fr/DIRAC>)
 - Exploit parallelism through workflows
- Scientific applications as a service
 - No need for installation on the users' side
- Facilitate the sharing of applications and data
- GPU usage
- Foster open and reproducible science
 - Can I have my results reproduced (e.g., by reviewers)?
 - Can I share/exchange applications/data across platforms?
 - ...



Boutiques and FAIR data analysis

- Describe, publish, integrate and execute applications **across platforms**
 - Facilitate application porting
 - Import and exchange of applications
- <https://github.com/boutiques>

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

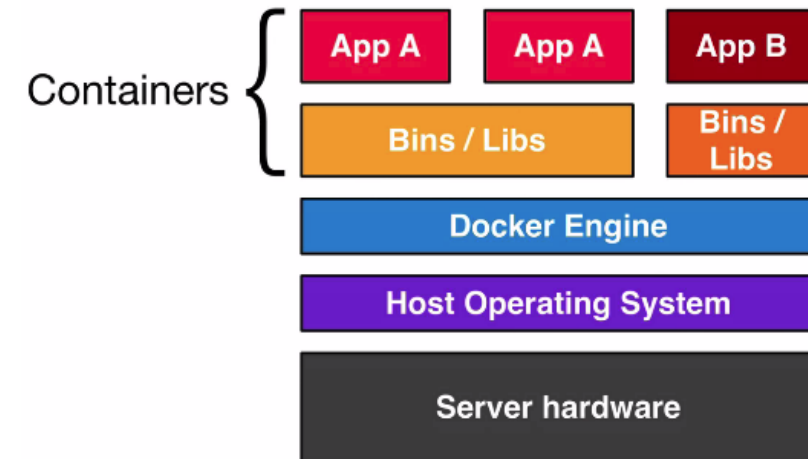
1. Multiple accurate and relevant attributes
2. Clearly licensed
3. Meets minimum domain standards

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

FAIR tools. Credit: Gregory Kiar and Tristan Glatard

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
 - **Facilitates application installation and sharing**
- Docker has become synonymous with container technology because its success, but
 - Container technology is not new
 - Other containers exist (**Singularity**)
- DockerHub
 - Image discovery and distribution
 - <https://hub.docker.com>



Credit: <https://stackoverflow.com>

Application publishing in VIP with Boutiques

- Share your tools in a packaged and fully described fashion
- Boutiques publishes descriptors to Zenodo (<https://zenodo.org>)

Application: GateCLforOpenDose + Add Version Refresh

Visible	Version	LFN
<input checked="" type="checkbox"/>	v0.1.0	/grid/biomed/creatis/vip/data/users/sorina_camara
<input checked="" type="checkbox"/>	v0.2.0	/grid/biomed/creatis/vip/data/groups/Opendose/Gi

Add/Edit Version

Application: GateCLforOpenDose

Version:

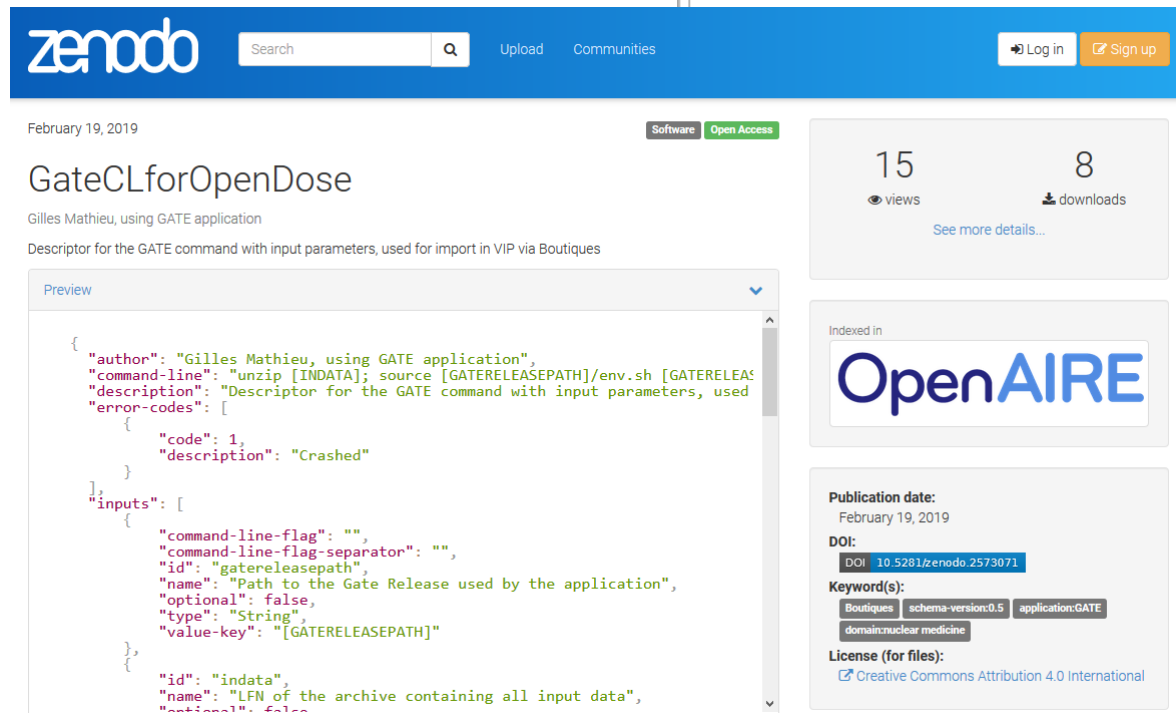
Gwendia LFN:

JSON LFN:

Visible

Publish Version

Status: Not published



zenodo Search Upload Communities Log in Sign up

February 19, 2019 Software Open Access

GateCLforOpenDose

Gilles Mathieu, using GATE application


Descriptor for the GATE command with input parameters, used for import in VIP via Boutiques

15 views

8 downloads

[See more details...](#)

Indexed in



Publication date: February 19, 2019

DOI: [10.5281/zenodo.2573071](https://doi.org/10.5281/zenodo.2573071)

Keyword(s): Boutiques schema-version:0.5 application:GATE domain:nuclear medicine

License (for files): [Creative Commons Attribution 4.0 International](#)

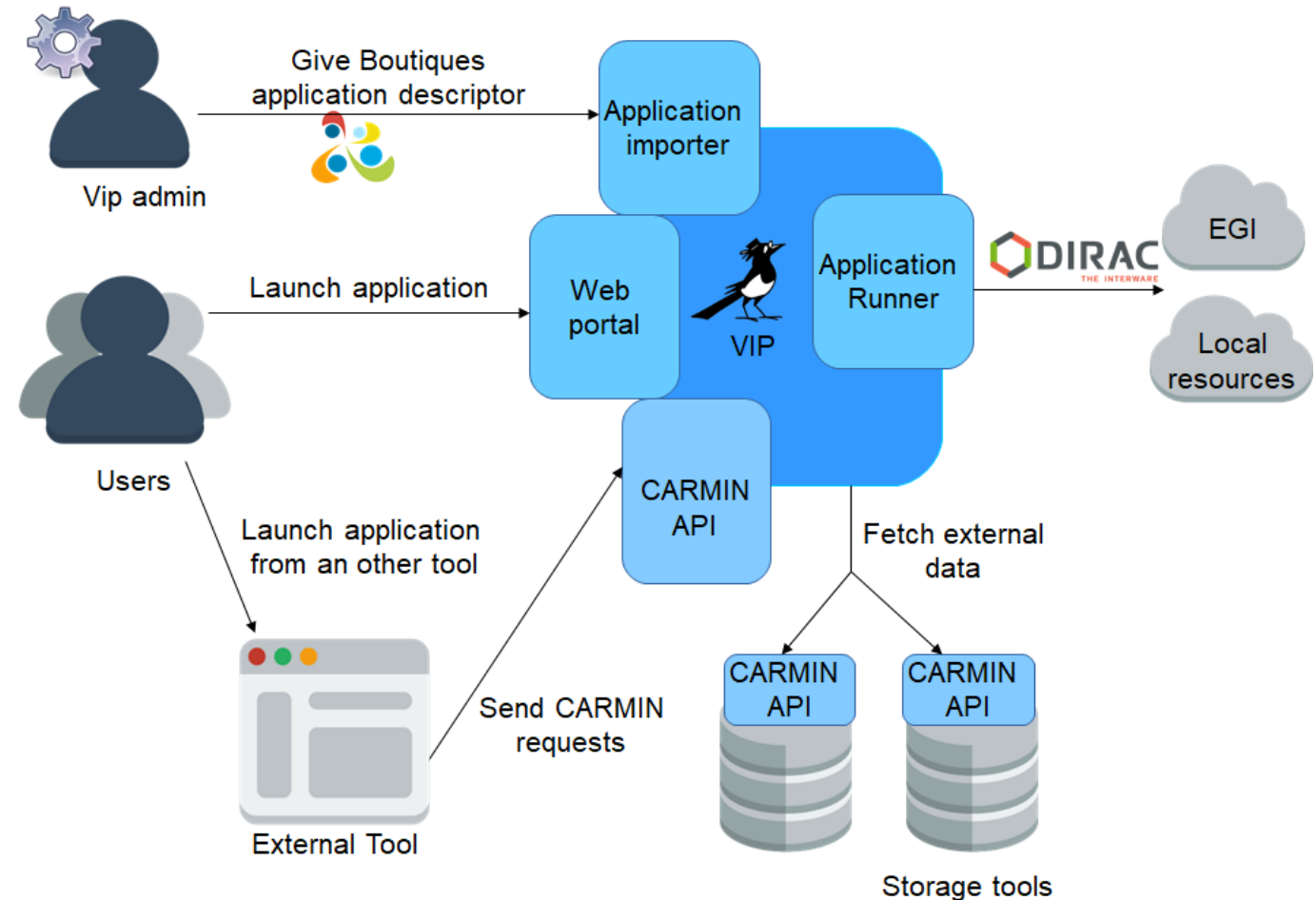
```

{
  "author": "Gilles Mathieu, using GATE application",
  "command-line": "unzip [INDATA]; source [GATERELEASEPATH]/env.sh [GATERELEASEPATH] [INDATA]",
  "description": "Descriptor for the GATE command with input parameters, used for import in VIP via Boutiques",
  "error-codes": [
    {
      "code": 1,
      "description": "Crashed"
    }
  ],
  "inputs": [
    {
      "command-line-flag": "",
      "command-line-flag-separator": "",
      "id": "gatereleasepath",
      "name": "Path to the Gate Release used by the application",
      "optional": false,
      "type": "String",
      "value-key": "[GATERELEASEPATH]"
    },
    {
      "id": "indata",
      "name": "LFN of the archive containing all input data",
      "optional": false
    }
  ]
}

```

Interoperability and CARMIN

- Common API for Research Medical Imaging Network
 - <https://github.com/CARMIN-org>
- Enables communication between services
- Interoperability among data and computing platforms



Outline

- The Virtual Imaging Platform
- Towards open and reproducible science
 - Boutiques and FAIR data analysis
 - Interoperability and CARMIN
- **Organization of conference challenges**
- Conclusions

REPRODUCIBILITY IN MEDICAL IMAGING PROCESSING:

PRIVATE AND SENSIBLE DATA.

SMALL DATASET (not least true).

COMPARISON NOT EASY.

CHALLENGE:

EVENTS DURING INTERNATIONAL CONFERENCE.

OPPORTUNITY FOR GROUPS WORKING WITH COMPETITIVE APPROACHES TO RANK THEIR METHODS.

SAME DATASET TO COMPARE.

SAME METRICS TO COMPARE.

DIFFERENTS PHASE (TRAINING, TESTING, RANKING).

COMPUTATION/RESEARCH/TESTING DURING A SHORT TIME.

LIVING AFTER CHALLENGE END.

CHALLENGES:

CETUS (2014): MICCAI

PICMUS (2016): IUS

ACDC(2017): MICCAI, 2 challenges, 10 teams, >50 submissions. **After:** 500 users, >250 submissions

SAF-VI (2018): IUS, 7 teams, 50 submissions

CAMUS (2019): Article, Submitted

ACDC CHALLENGE CONTEXT : data warehouse

2 challenges

Compare different segmentation methods on Heart (left ventricle, right ventricle and myocardium) and different cardiac pathologies (150 subjects).

Based on **GIRDER data warehouse** (free and open source web-data management platform).

Several instances deployed at Creatis (H2P, ultrasound-warehouse, ...) for different uses: simple warehouse, cohort, challenge, ...

Server: CherryPY, very flexible plugin system

Front-end: javascript + template pages (pug, stylesheets)

Database server: MongoDB (more flexible)

Plugins can be easily developed

example: DicomViewer (1 file/server, 4 files/client)

CHALLENGE PLUGIN: Metrics on **docker**, specific front-end for ranking and submission.

The screenshot shows the H2P web interface. On the left, there's a sidebar with navigation options: Collections, Users, Groups, and Admin console. The main area displays a DICOM image of a heart scan. To the right of the image is a metadata table with columns for Name and Value.

Name	Value
AcquisitionMatrix	0,236,148,0
AcquisitionNumber	1
AngioFlag	N
BitsAllocated	16
BitsStored	12
BodyPartExamined	HEART
CardiacNumberOfStrips	1
Columns	250
ContrastBolusAgent	GAOOLINUM
ContrastBolusAgentConcentration	0
ContrastBolusTotalDose	0
ContrastBolusVolume	0
dBf	0
DeviceSerialNumber	66655
EchoNumbers	1
EchoTime	1,859
FramePosition	1

Below the image, there's an 'Info' section showing file details: 1.958 MB (2095112 bytes), Created on November 17, 2017 at 15:54:46, Updated on November 17, 2017 at 15:54:46. A 'Metadata' section is also visible. At the bottom, there's a 'Files & links' section with a link to 'IM-0103-0010.dcm' and a '102.3 KB' size. Below that, there's a 'Download test dataset', 'Download ground truth data', and 'Submit your results' buttons. A 'LEADERBOARD' section is also present, showing a table of user performance metrics.

Rank	User	Mean DICE ED	Mean DICE ES	Mean Hausdorff ED	Mean Hausdorff ES	FF correlation	FF bias	FF standard deviation (std)	Volume ED correlation	Volume ED bias	Volume ED std
1	Fabien Dorset	0.958	0.931	7.394	5.005	0.991	0.178	3.028	0.997	2.558	5.726
2	Mahendra Khenedi	0.944	0.917	8.129	8.848	0.989	0.548	3.422	0.997	0.976	5.501
3	Yezheng Jiang	0.939	0.931	7.237	7.116	0.988	-0.330	3.281	0.993	-0.440	6.791
4	Christian Baumgartner	0.902	0.911	6.526	5.170	0.988	0.568	3.398	0.995	1.426	7.610
5	Marc-Alexandre Bui	0.897	0.900	7.483	10.747	0.988	-0.694	3.215	0.993	4.182	8.422
6	gomer Wobbenk	0.901	0.918	7.515	5.093	0.990	-0.094	3.421	0.993	3.046	6.632
7	devesh joshi	0.937	0.905	6.641	8.798	0.987	-1.736	3.573	0.997	3.649	6.381
8	Shobam Jain	0.935	0.885	8.212	10.929	0.971	1.734	5.426	0.997	3.964	6.671
9	Blati Gribas	0.948	0.888	8.898	11.348	0.975	-1.540	4.972	0.992	2.002	11.899
10	Xin Yeig	0.864	0.775	47.873	53.890	0.835	1.486	8.751	0.884	12.232	31.364

ACDC CHALLENGE RESULTS:

During challenge : 10 teams, > 50 submissions.

After: 500 users, > 250 submissions

Real-Time Ranking.

Each entry on leaderboard **associated** to an article.

Execution time between 5 min. and 30 min.

Issues on dataset.

Instability on early deployment

High maintenance.
Needed a strong support.
Metric code on github

1 articles with IEEE TMI

> 350 mails exchanged with challengers

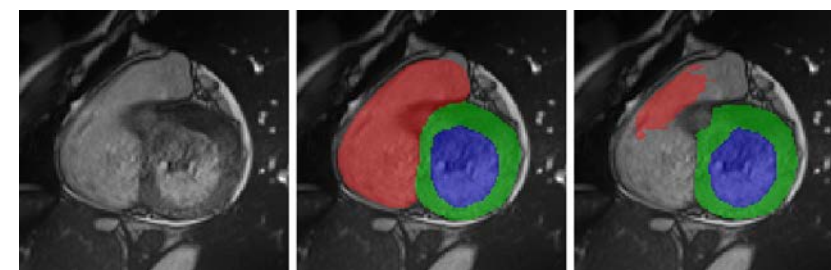
LEADERBOARD

Rank	User	Mean DICE ED	Mean DICE ES	Mean Hausdorff ED	Mean Hausdorff ES	EF correlation	EF bias	EF standard deviation(std)	Volume ED correlation	Volume ED bias	Volume ED std
LV											
1	Fabian Isensee	0.965	0.933	5.608	6.300	0.992	0.338	2.800	0.997	1.590	5.851
2	Clement Zotti	0.964	0.912	6.180	8.386	0.990	-0.476	3.114	0.997	3.746	5.146
3	Mahendra Khened	0.964	0.917	8.129	8.968	0.989	-0.548	3.422	0.997	0.576	5.501
4	Christian Baumgartner	0.963	0.911	6.526	9.170	0.988	0.568	3.398	0.995	1.436	7.610
5	Jelmer Wolterink	0.961	0.916	7.515	9.603	0.988	-0.494	3.421	0.993	3.046	8.692
6	Marc-Michel Rohé	0.957	0.900	7.483	10.747	0.989	-0.694	3.215	0.993	4.182	8.622
7	clement zotti	0.957	0.905	6.641	8.706	0.987	-1.186	3.573	0.997	9.640	6.381
8	Shubham jain	0.955	0.885	8.212	10.929	0.971	1.734	5.476	0.997	9.864	6.671
9	Ilias Griniias	0.948	0.848	8.898	12.934	0.970	-1.736	5.482	0.992	2.454	11.061
10	Xin Yang	0.864	0.775	47.873	53.050	0.926	1.496	8.731	0.894	12.232	31.964

Rank	User	Mean DICE ED	Mean DICE ES	Mean Hausdorff ED	Mean Hausdorff ES	EF correlation	EF bias	EF standard deviation (std)	Volume ED correlation	Volume ED bias	Volume ED std
RV											
1	Fabian Isensee	0.946	0.904	8.835	11.376	0.925	-2.966	5.088	0.991	2.136	8.965
2	Clement Zotti	0.934	0.885	11.052	12.650	0.869	-0.872	6.760	0.986	2.372	11.531
3	clement zotti	0.941	0.882	10.318	14.053	0.872	-2.228	6.847	0.991	-3.722	9.255
4	Mahendra Khened	0.935	0.879	13.994	13.930	0.858	-2.246	6.953	0.982	-2.896	12.650
5	Christian Baumgartner	0.932	0.883	12.670	14.691	0.851	1.218	7.314	0.977	-2.290	15.153
6	Jelmer Wolterink	0.928	0.872	11.879	13.399	0.852	-4.610	6.884	0.980	3.596	15.192
7	Marc-Michel Rohé	0.916	0.845	14.049	15.926	0.781	-0.662	9.896	0.983	7.340	13.363
8	Shubham jain	0.911	0.819	13.517	18.729	0.791	6.784	8.082	0.945	5.634	22.251
9	Ilias Griniias	0.887	0.767	19.041	24.249	0.756	-0.192	9.693	0.916	11.910	27.824
10	Xin Yang	0.789	0.770	30.285	31.089	0.576	8.832	23.198	0.789	47.322	41.915

Rank	User	Mean DICE ED	Mean DICE ES	Mean Hausdorff ED	Mean Hausdorff ES	Volume ES correlation	Volume ES bias	Volume ES standard deviation (std)	Mass ED correlation	Mass ED bias	Mass ED std
------	------	--------------	--------------	-------------------	-------------------	-----------------------	----------------	------------------------------------	---------------------	--------------	-------------

Ranking



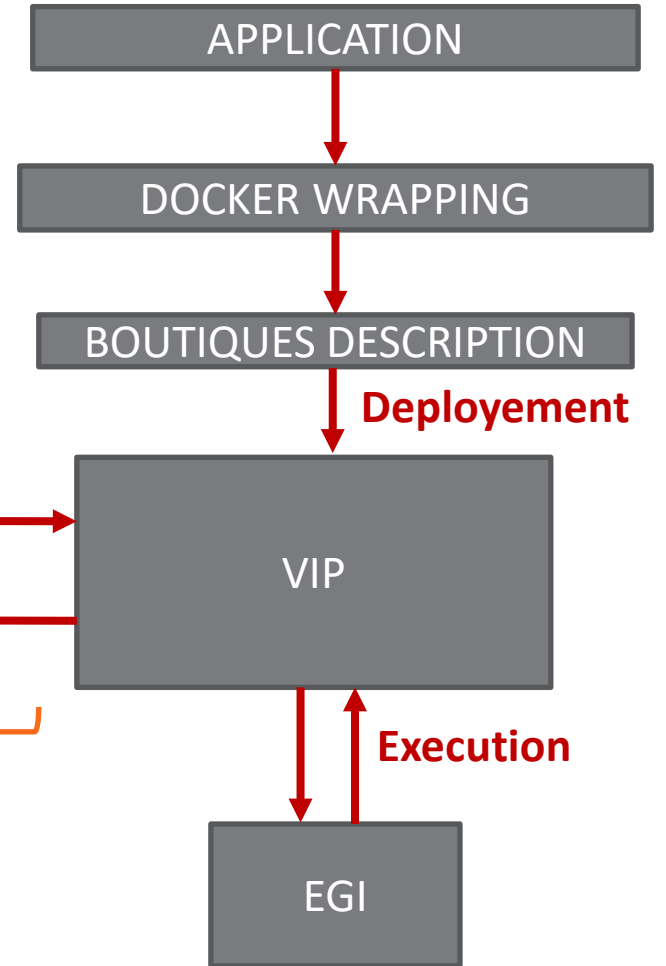
[Left] input image; [Middle] ground truth; [Right] prediction.

CHALLENGE FLI: CONTEXT



2 challenges

Compare different segmentation methods provided by challengers on PET IMAGES (100 images) and SEP images (53 subjects) during MICCAI 2016.



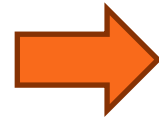
SHANOIR
(DATA WAREHOUSE)

	Number of challengers	Number of pipelines
MSSEG challenge	11	14
PETSEG challenge	4	9
Total	15	23

carmen API 0.1

CHALLENGE FLI: RESULTS

- During the challenge, the executions took place in production conditions (*i.e.* on a shared infrastructure) the resource usage was influenced by the charge of the platform at the moment of the execution.

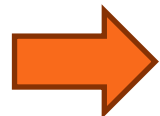


Benchmark :

- 1 Intel Xeon E5-2630L v4 processor (1.8GHz, 10 cores, 2 threads per core) and 64GB of RAM
- All MSSEG Pipelines
- Only 4 subjects
- Sequentially to avoid overlap

2 articles with Nature

> 650 mails exchanged between VIP team and challengers



- **Integration not too smooth** at the beginning for complex pipelines.
- Issues on MSSEG dataset: relaunch 3 three times all pipelines=> **time consuming**

Pipeline (method)	Elapsed time	Maximum resident set size
	Average (second)	(Mbytes)
	Average	Average
edgebasedmsseg_pipeline	232,00	3 469,39
pipeline_EOT	385,25	641,79
MSSeg_MIVG	620,75	4 409,77
sls-challenge2016	741,50	4 222,97
intensityNormalizedSegmentation.py	950,50	3 746,01
graphCutSegmentation.py	1 439,25	4 979,52
ms_run_t1_flair_only	1 617,00	10 133,26
Miccai_Urien_pipeline	2 158,00	6 743,54
ms_run	2 654,75	8 047,65
Plocus_MSseg	2 801,00	3 770,34
nabla-MS	4 669,00	22 510,86
deep_challenge	9 328,50	4 722,65
CRL2	12 646,25	11 452,65
muschellij2_msseg	15 852,25	15 745,71

DATA/RESULTS REPRODUCIBILITY

ACCESSIBILITY



Guidelines to share
Challenges
Identified and labelled
infrastructures

TRACABILITY



DOI
Identify pipelines/workflows

QUALITY



Open-source
Adapted tools/infrastructure

Conclusions

- Achievements

- Transparent access to distributed computing resources
- Scientific applications as a service
- Challenge organization

- Looking to the future

- Collaboration and tool “re-use”
- Support and sustainability
- Interoperability
- Open and reproducible science



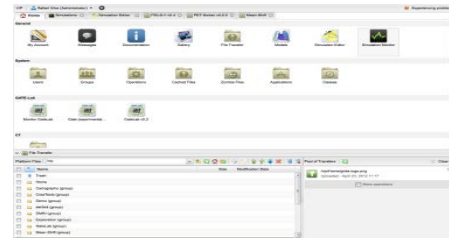
Thank you for you attention!
Questions?

— sorina.pop@creatis.insa-lyon.fr —
frederic.cervenansky@creatis.insa-lyon.fr



User

- 0. Login
- 1. Send input data



Web Portal

- 3. Launch workflow



Workflow Engine
(Moteur + GASW)

- 4. Generate and submit task



Storage Element

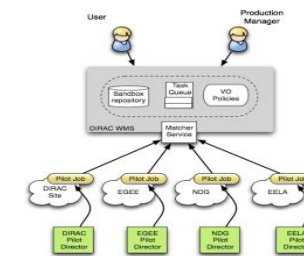
- 7. Get files
- 8. Execute
- 9. Upload results



Computing site

- 6. Get task

- 5. Submit pilot jobs

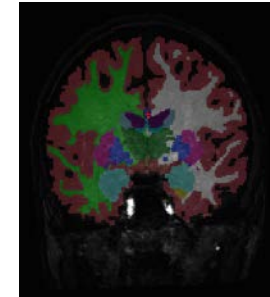


Pilot Manager
(DIRAC)

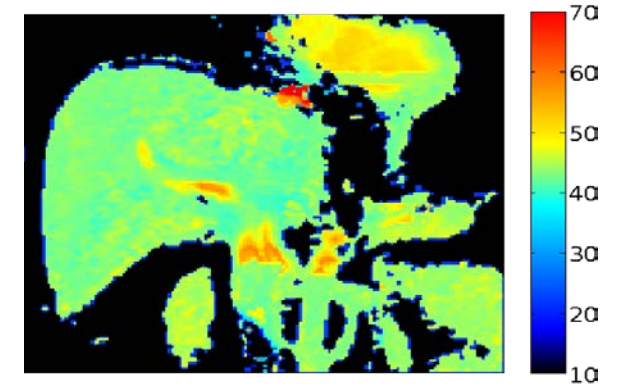


Biomed community

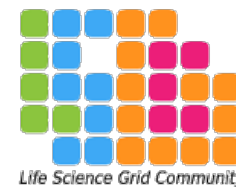
- Life Sciences sector with three main thematic groups
 - Medical image analysis
 - Bioinformatics
 - Drug discovery
- EGI's biomed VO
 - Operating since 2004
 - Approx 50 supporting sites
 - Heterogeneous resources and user profiles
 - 2.5 million jobs and 500 CPU years/year
 - Technical teams on shift for monitoring
- Open access
 - For non-commercial users
 - For life-science applications



Brain tissue segmentation
with Freesurfer



Hepatic perfusion index (%)
Credits: B. Leporq, O Beuf



Life Science Grid Community

<http://lsgc.org>



A Docker container for our application

- Prepare the Dockerfile
 - Use an existing nvidia image having cuda and cuDNN already installed (nvidia/cuda:7.5-cudnn5-devel-centos7)
 - Install and configure anaconda, theano and keras
 - Bring in code source with git clone (or “ADD” local files)
- Build the image
 - `docker build -t feature-extraction .`
- Use nvidia-docker
 - docker runtime enabling access to the GPU
- Start the container using the nvidia runtime
 - `docker run --runtime=nvidia -it feature-extraction`