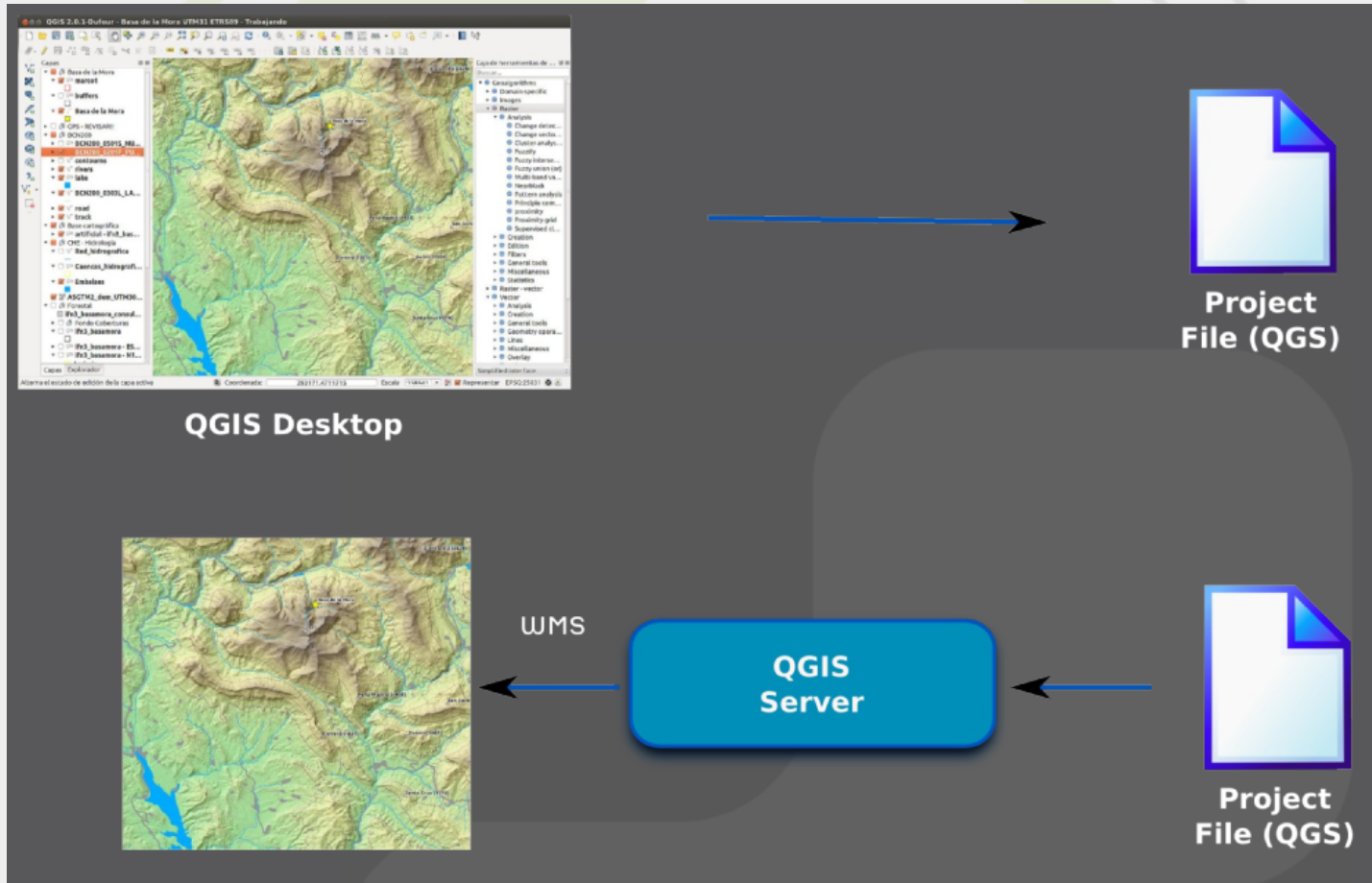


FOSS4G FR 2018

Implémenter et déployer des services OGC avec QGIS 3.0

David Marteau - 3Liz

Qgis server



Qgis server

Historique :

- Qgis server Annoncé en mai 2007 - Marco Hugentobler (SourcePole)
- Début du développement de Qgis 3.0
- Décembre 2016: Code sprint Qgis Server à Lyon (QEP #74)
- Février 2018: Sortie de Qgis 3.0

QGIS 3.0

- Refonte de l'api Qgis (C++, python)
- Refonte de Qgis server (intégrable, extensible)

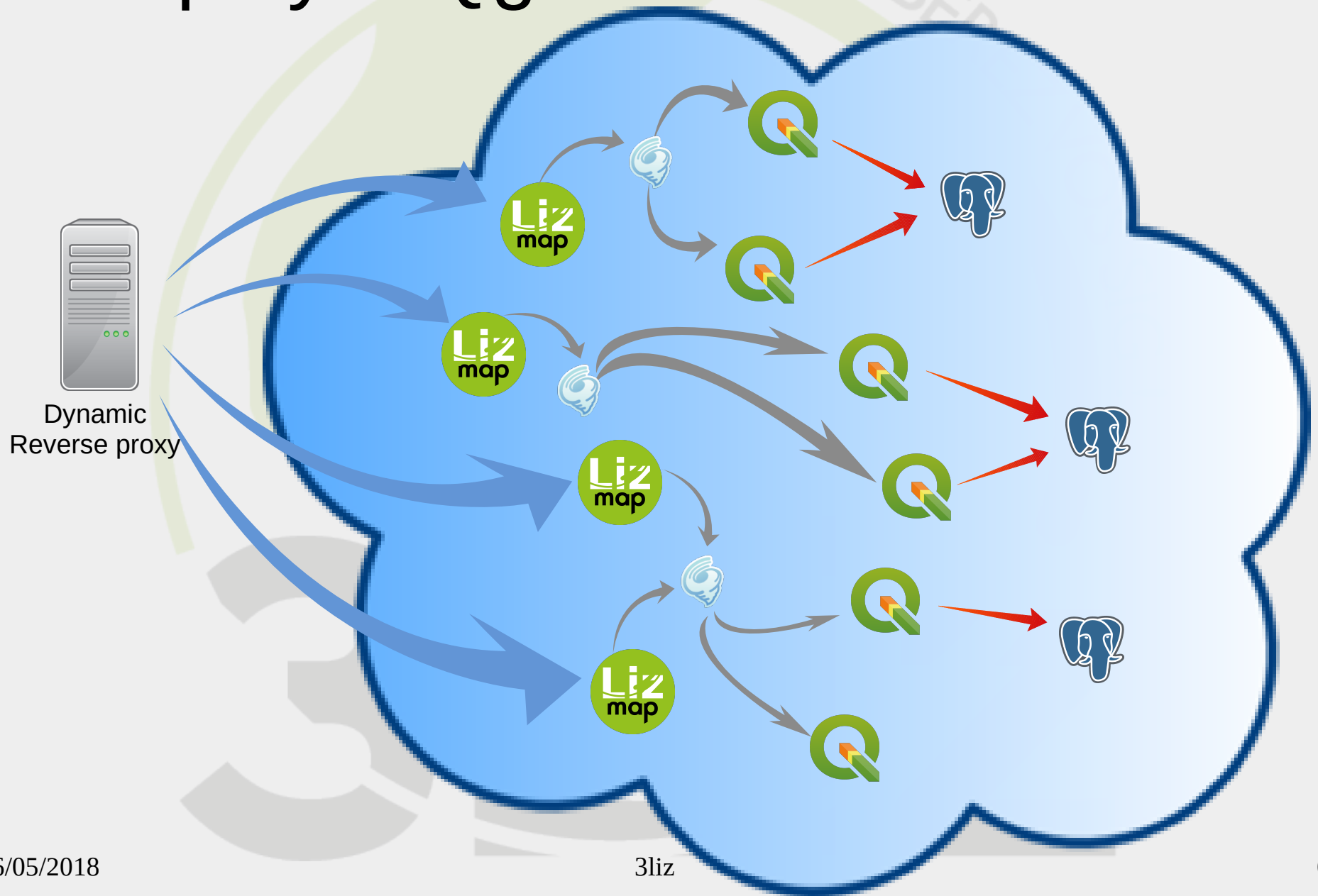


Régis Haubourg – Lyon 2016

3LIZ

- Hébergement de services SIG en ligne avec LizMap et Qgis server
- Problèmes à résoudre :
 - Scalabilité
 - Architecture distribuée
 - Déploiement Zeroconf
 - Monitoring
 - Sécurité

Déployer Qgis server en cluster



Déployer Qgis server en cluster

HTTP
Asynchrone



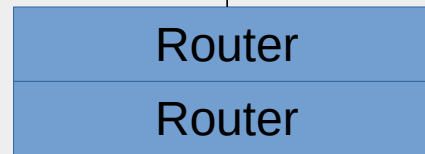
HTTP
Asynchrone



HTTP
Asynchrone



OMQ



OMQ



+

Intégrer Qgis server avec 0MQ

- Qgis 3 server api : Interface uniforme pour les requêtes et les réponses.

```
class Qgs0MQRequest(QgsServerRequest) :
```

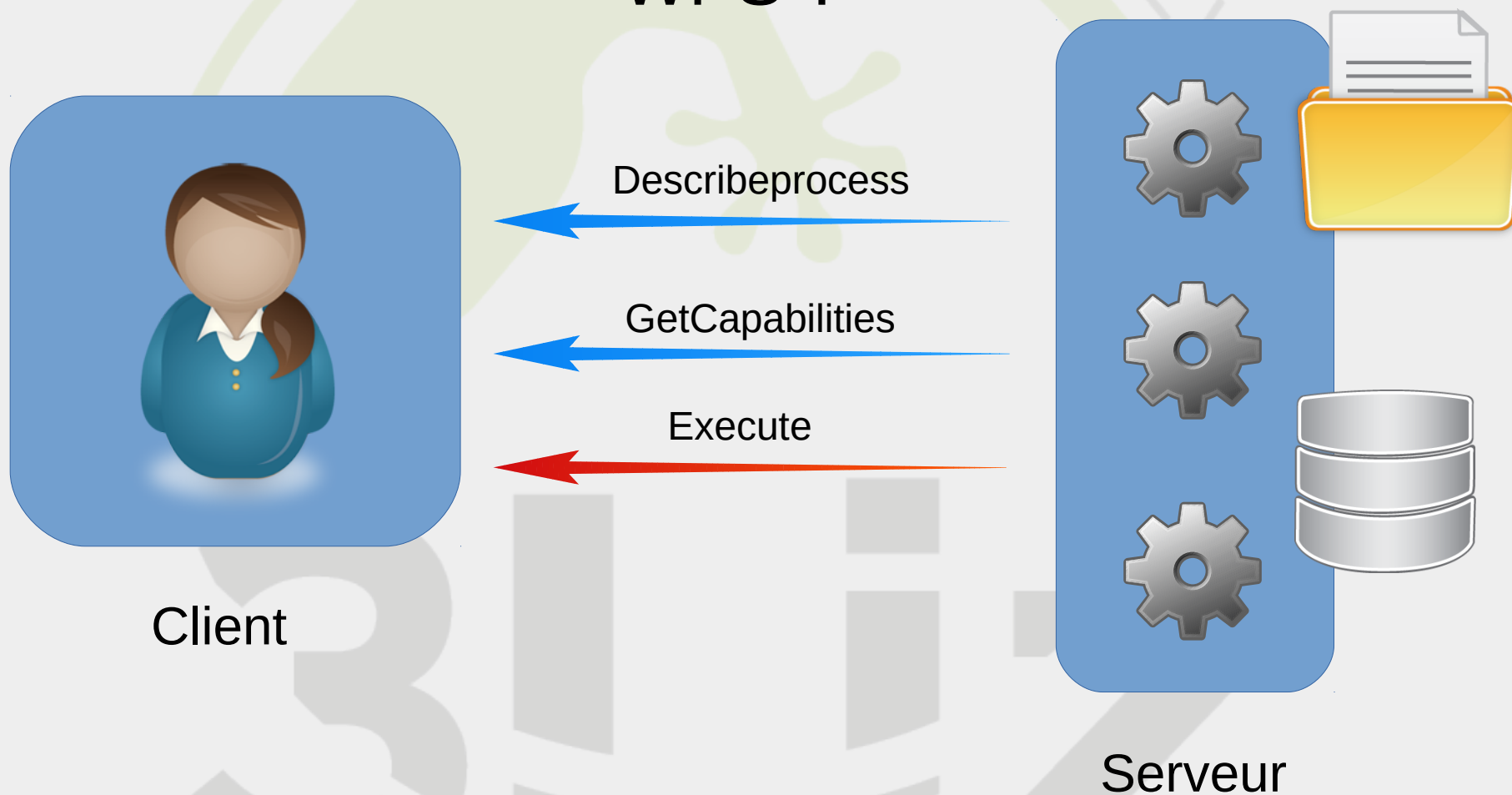
```
class Qgs0MQResponse(QgsServerResponse) :
```

```
qgis_server.handleRequest(request, response, project=project)
```

- Gestion du cache
- Gestion des logs (hook QgsMessageLog)
- Gestion des buffers/streaming

Services OGC

WPS ?

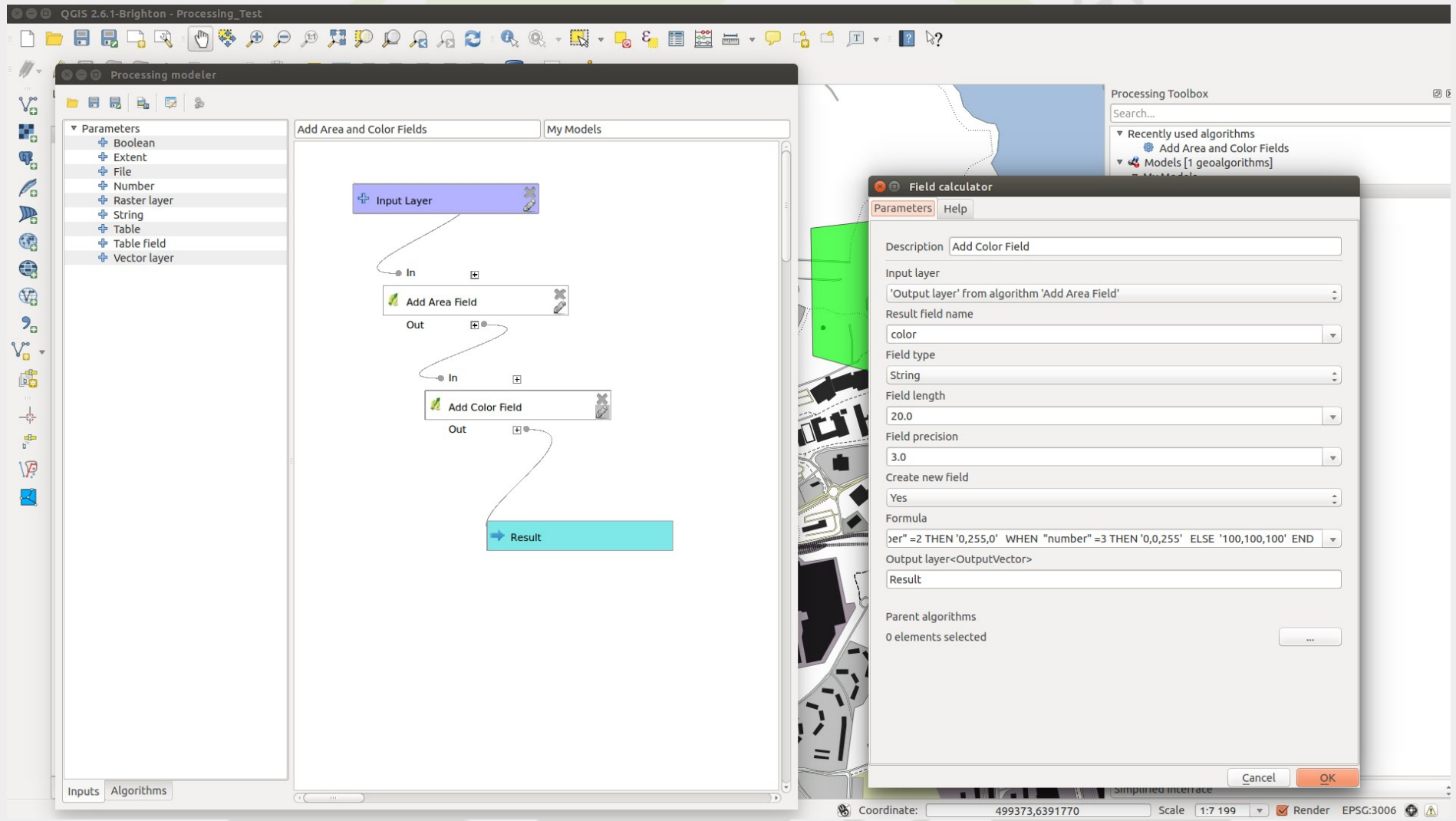


Services OGC

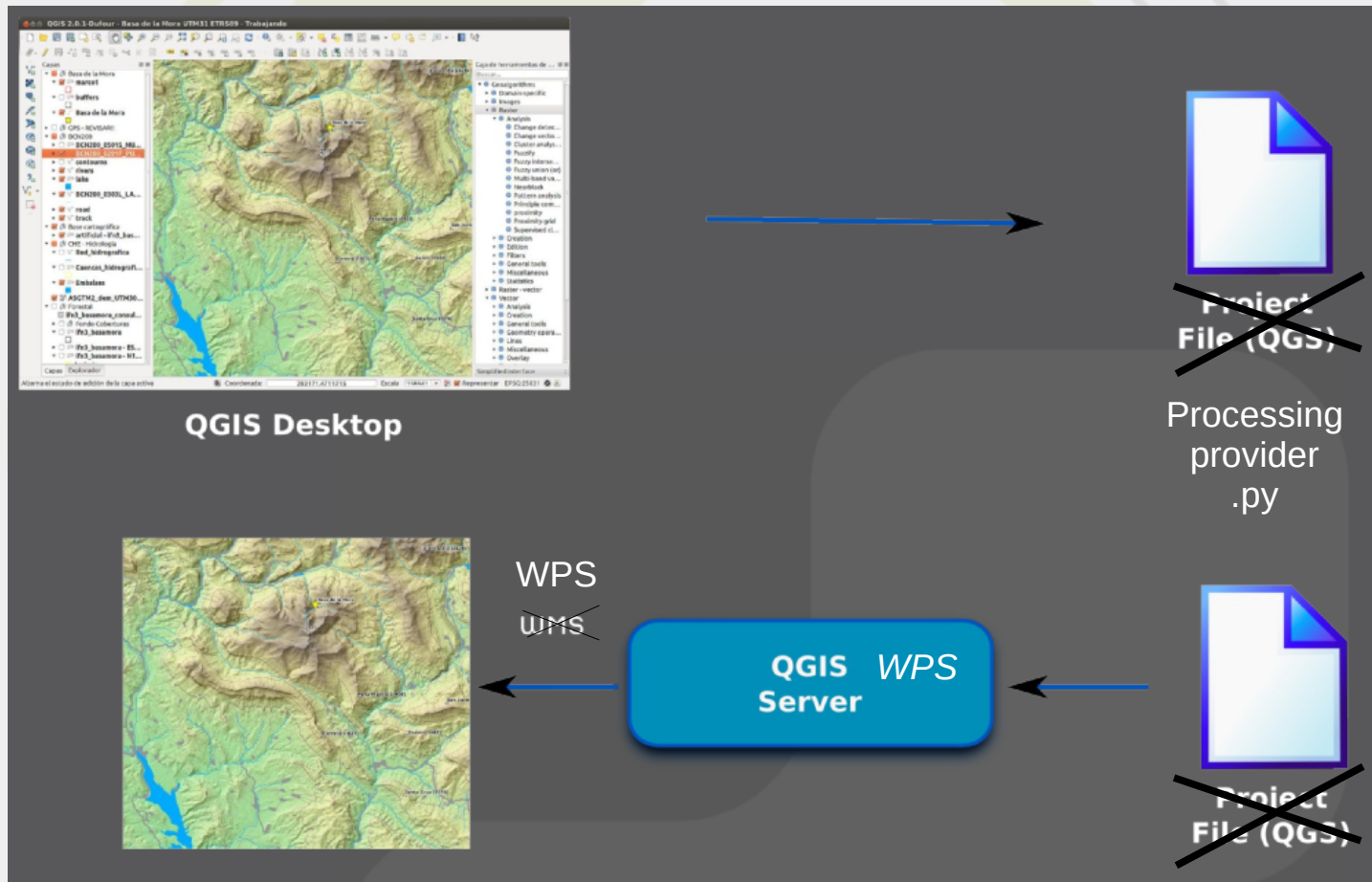
WPS ?

- Qgis server : WMS, WFS, WCS
- wps4server : wps basé sur processing
 - Qgis 2.8
 - Pas asynchrone
 - Processus unique
 - Plugin

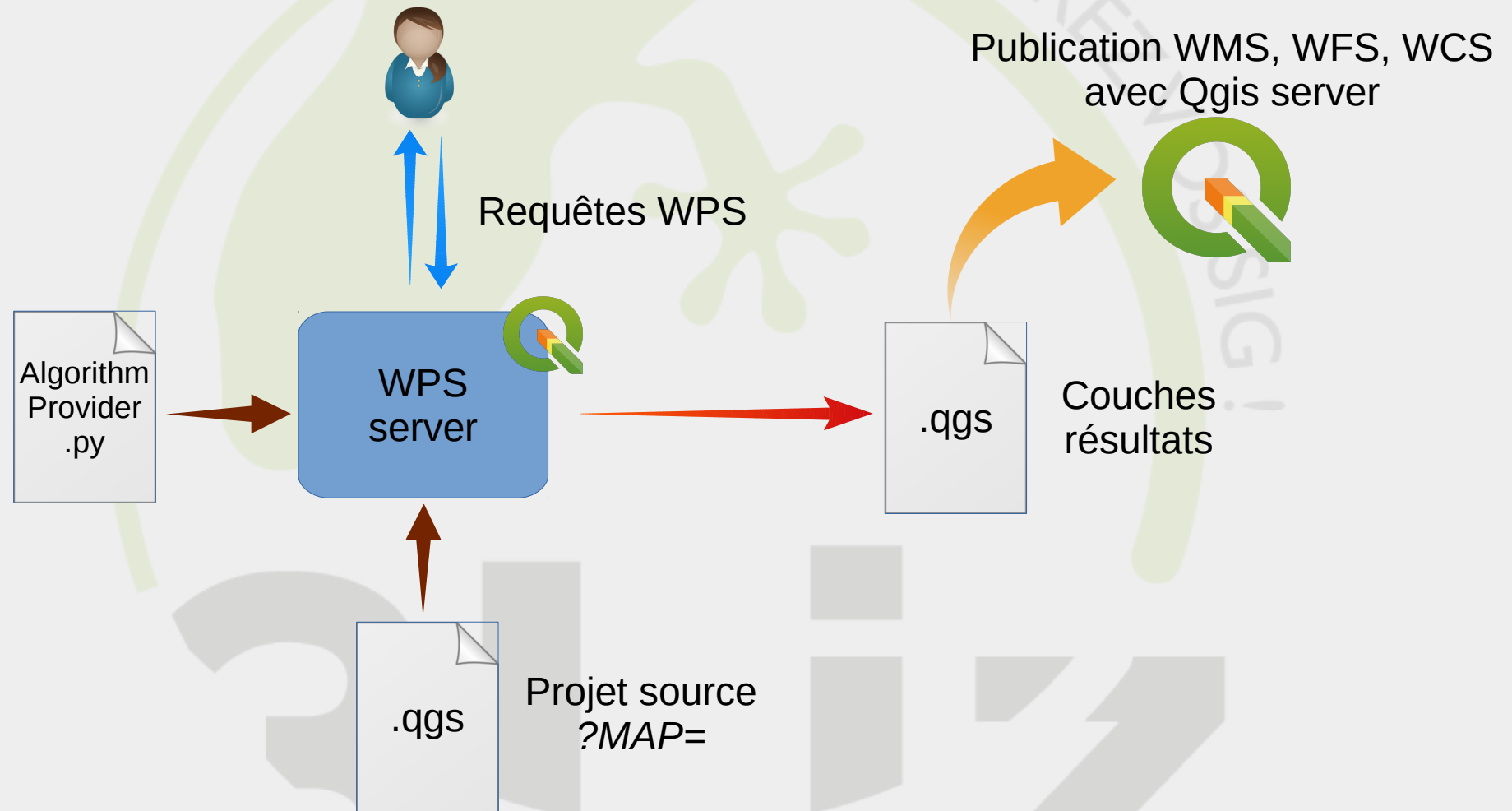
QGIS Processing



WPS+PROCESSING



WPS+PROCESSING



WPS+PROCESSING

- Contrainte : écriture uniforme des traitements processing pour le desktop **ET** le serveur
- API Qgis 3 :
 - ***QgsProcessingContext***: définition d'un Projet "Source" et d'un projet "cible"
 - ***QgsProcessingFeedback*** : encapsule la mise à jour du status WPS.
- WPS
 - Couches résultats : « complex output » associé à une adresse WMS.

WPS : Implémentation

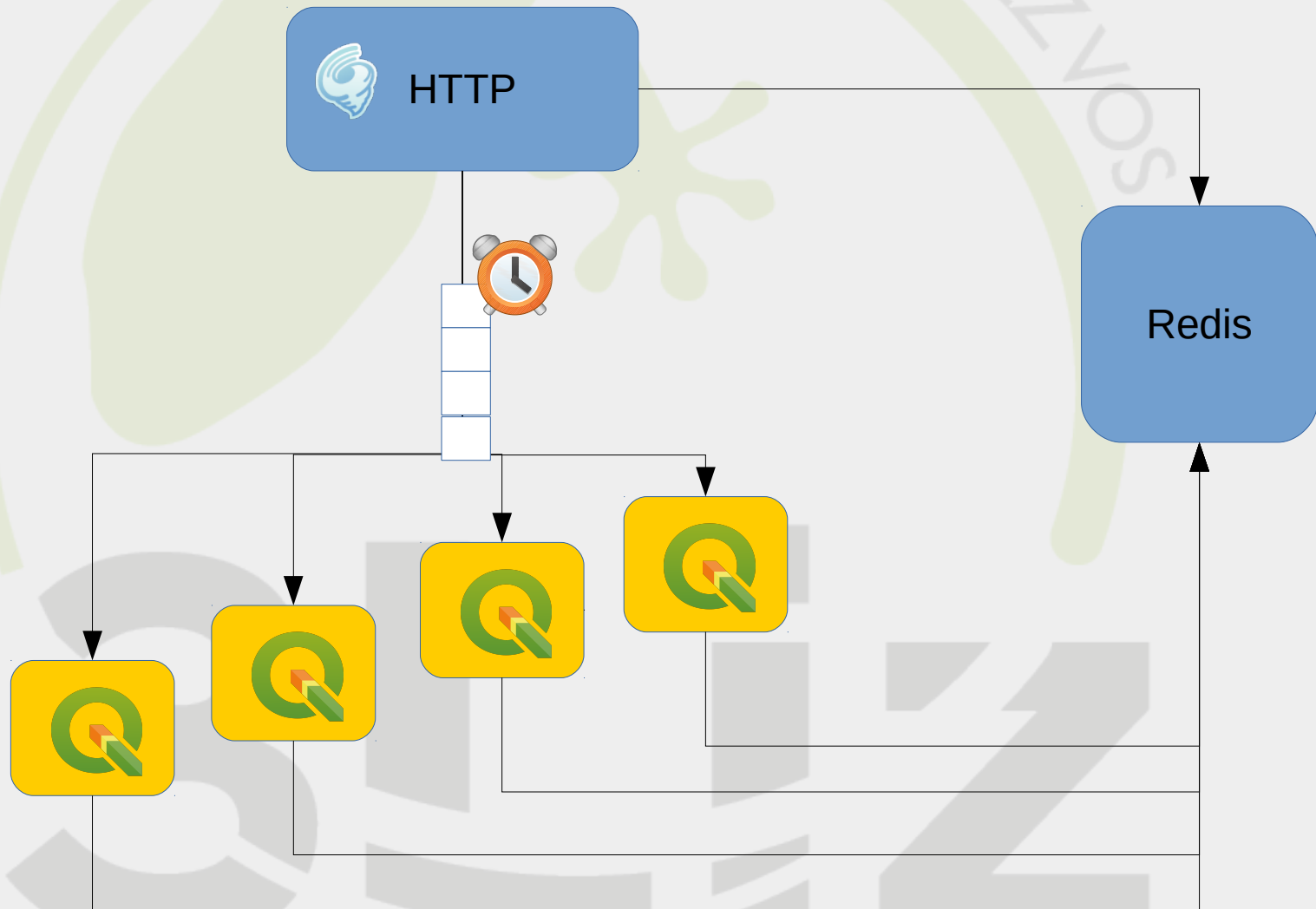
- Financé par l'ONFI (Projet PETRA)
- Fork de PyWPS (Jachym Cepick, OSGeo Foundation)



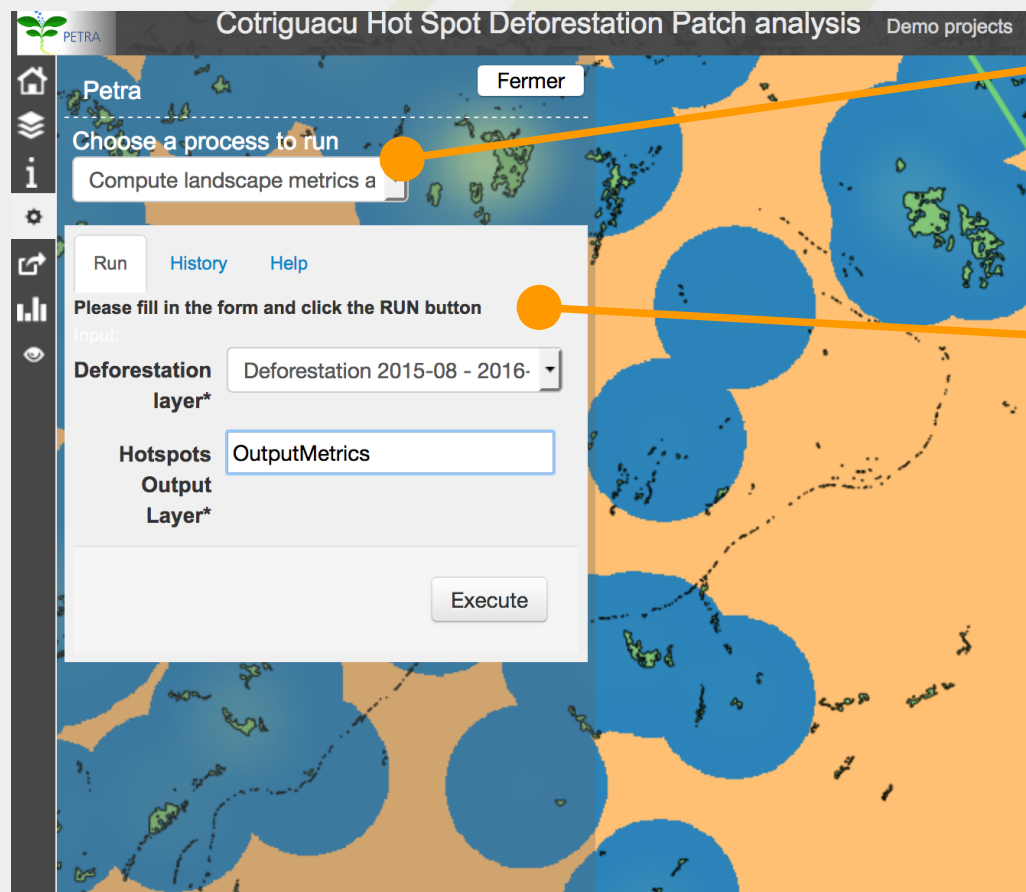
WPS : Implémentation

- Différences avec PyWPS
 - Serveur HTTP **asynchrone** (Tornado, Python 3 asyncio)
 - Gestion asynchrones des requêtes (même avec ``storeExecuteResponse=true``)
 - Logging uniforme - utilisation du module 'logging'
 - Status API
 - Backend REDIS
 - Extension: TIMEOUT et EXPIRE
 - Sélection des features avec des expressions Qgis.

WPS : Implémentation



Lizmap: un client WPS



Sélection de l'algorithme

Paramètres

Lizmap: un client WPS

Petra Cotriguacu Hot Spot Deforestation Patch analysis Demo projects

Choose a process to run
Compute landscape metrics

Run History Help

Start	End	Status	Actions
11/05/2018 à 17:23:03	11/05/2018 à 17:23:24	✓	↑ ↓ ×

11/05/2018 à 17:23:03

Name	Type	Value
Deforestation layer	raster	Deforestation_2016_2017
Hotspots Output Layer	rasterDestination	OutputMetrics

WPS status web api

WPS Status

Processes_(raw)

Waiting:0

Running:0

Done:5

Error:2

✓	lzmtest:testcopylayer	started: 11/05/2018 à 15:39:58	finished: 11/05/2018 à 15:39:59	🗑
✓	lzmtest:testcopylayer	started: 11/05/2018 à 15:39:58	finished: 11/05/2018 à 15:39:59	🗑
✓	lzmtest:testcopylayer	started: 11/05/2018 à 15:39:58	finished: 11/05/2018 à 15:39:58	🗑
⚠	lzmtest:testlongprocess	started: 11/05/2018 à 15:39:59	finished: 11/05/2018 à 15:40:02	🗑
✓	lzmtest:testcopylayer	started: 11/05/2018 à 15:39:59	finished: 11/05/2018 à 15:39:59	🗑
✓	lzmtest:testcopylayer	started: 11/05/2018 à 15:40:02	finished: 11/05/2018 à 15:40:03	🗑
⚠	lzmtest:testraiseerror	started: 11/05/2018 à 15:40:03	finished: 11/05/2018 à 15:40:03	🗑

WPS status web api

WPS Status

lzmtest:testcopylayer

Refresh

cc4562a4-5520-11e8-af0a-0242ac130002 [\(raw\)](#)

[Status](#) [Inputs](#) [Resources](#) [Log](#)

Name	Type	Size
OUTPUT.cpg	file	5.0B
OUTPUT.dbf	file	10K
OUTPUT.prj	file	143B
OUTPUT.qpj	file	257B
OUTPUT.shp	file	48K
OUTPUT.shx	file	132B
processing.log	file	926B
testcopylayer.qgd	file	0.0B
testcopylayer.qgs	file	38K

WPS Status web api

WPS Status

lzmtest:testcopylayer

[Refresh](#)

cc4562a4-5520-11e8-af0a-0242ac130002 [\(raw\)](#)

[Status](#)[Inputs](#)[Resources](#)[Log](#)







































```
2018-05-11 13:39:58,932 [53] INFO Starting task lzmtest:testcopylayer:cc4562a4
2018-05-11 13:39:58,980 [53] DEBUG Updated project cache key=france_parts path=/projects/france_parts.
2018-05-11 13:39:59,076 [53] INFO lzmtest:testcopylayer:cc4562a4 Results: {'OUTPUT': './OUTPUT.shp'}
2018-05-11 13:39:59,112 [53] DEBUG Getting style for lzmtest:testcopylayer: OUTPUT </processing/qml/ha
2018-05-11 13:39:59,112 [53] DEBUG Adding style to layer france_parts_2 (outputName OUTPUT)
2018-05-11 13:39:59,118 [53] DEBUG Adding Map layer './OUTPUT.shp' (outputName OUTPUT) to Qgs Project
2018-05-11 13:39:59,120 [53] INFO Task finished lzmtest:testcopylayer:cc4562a4-5520-11e8-af0a-0242ac1
2018-05-11 13:39:59,120 [53] DEBUG Writing Results to /srv/data/cc4562a4-5520-11e8-af0a-0242ac130002
2018-05-11 13:39:59,140 [53] INFO lzmtest:testcopylayer:cc4562a4 memory: start=142.242Mb end=150.875M
```

Déploiement avec Docker+Rancher

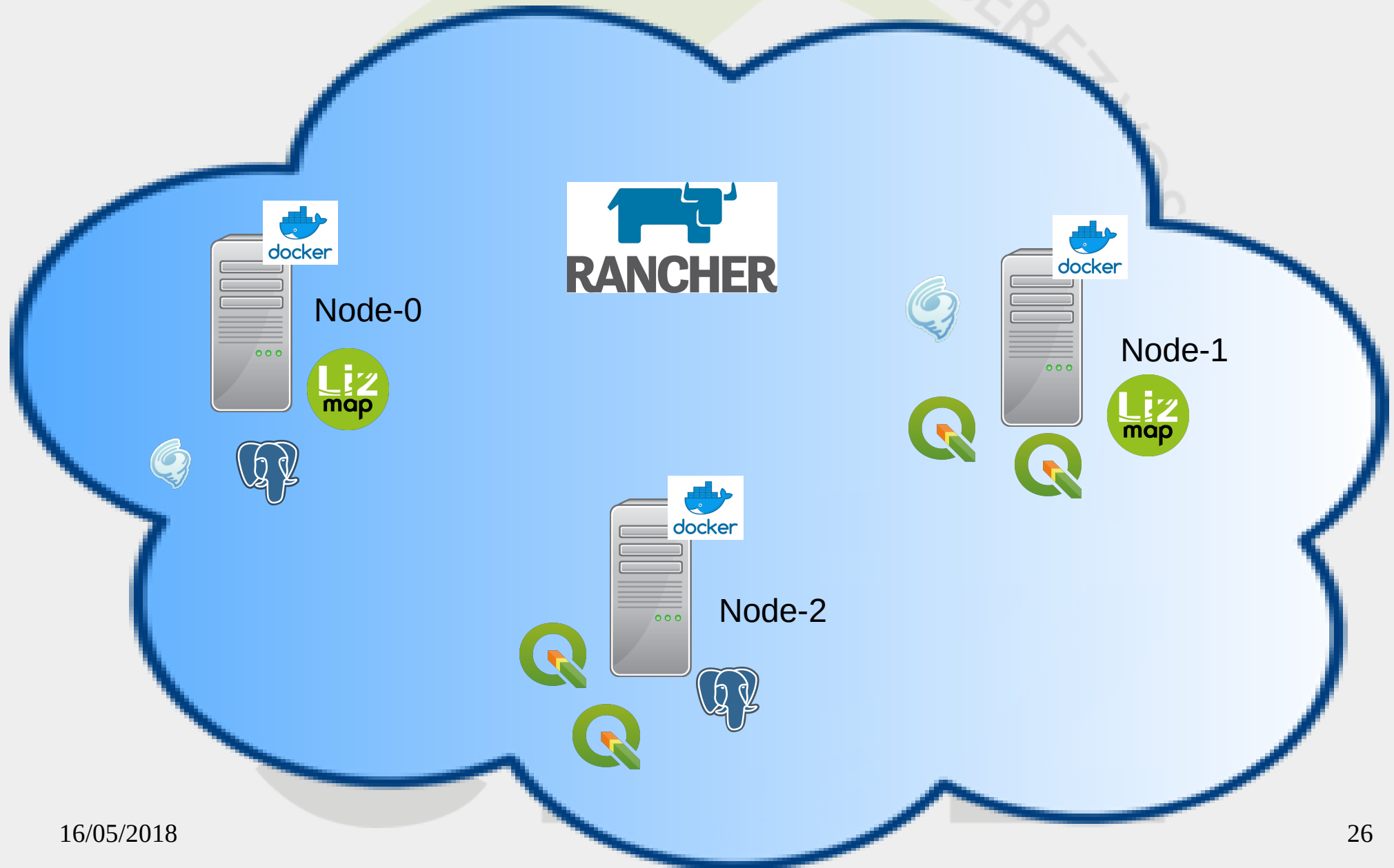


Docker + orchestrateur Rancher : organisation sous forme de piles (stacks) logicielles déployés sur un cluster de machines.

Déploiement avec Docker

 ONFI STACKS CATALOG INFRASTRUCTURE ADMIN API 									
User Stacks		Add Stack	Add from Catalog	Sort By: State Name					
 clusac-petra				Upgrade available	Add Service 	4 Services	4 Containers	 	
 Active	lizmap 	Image: registry.snap.lizlan:5000/lizmap-web-client:3.2				Service	1 Container	 	
 Active	map 	Image: registry.snap.lizlan:5000/qgis3-map-service:1.0				Service	1 Container	 	
 Active	redis 	Image: registry.snap.lizlan:5000/redis:4				Service	1 Container	 	
 Active	wps 	Image: registry.snap.lizlan:5000/qgis3-wps-service:1.0				Service	1 Container	 	
 clusac-petra-geodata				Up to date	Add Service 	1 Service	1 Container	 	
 Active	geodata 	Image: registry.snap.lizlan:5000/qgis3-geodata-service:1.0				Service	1 Container	 	
 clusac-petra-postgis				Up to date	Add Service 	1 Services	2 Containers	 	
 Active	qgisdb + 1 Sidekick 	Image: registry.snap.lizlan:5000/lzmcloud/postgis:9.6-2.4-0				Service	2 Containers	 	

Orchestration avec Rancher



Conclusion

- Satisfaction d'utiliser complètement le modèle editorial de Qgis en tant qu'outil de production de service OGC:
 - Configuration des sources de données
 - Symbologie
 - Prototypage et traitements analytique avec Processing
- Mise à l'épreuve de la nouvelle API python qgis 3 et qgis server.
 - Refonte de Processing apporte des solutions élégantes une utilisation dans des environnements et contraintes fonctionnelles variés.
 - Refonte de Qgis server atteint son objectif en permettant de s'interfacer relativement simplement avec différents protocoles.

Merci

- <https://github.com/3liz/py-qgis-wps>
- <https://github.com/3liz/py-qgis-server>
- <https://lizmap.com>
- <http://pywps.org>