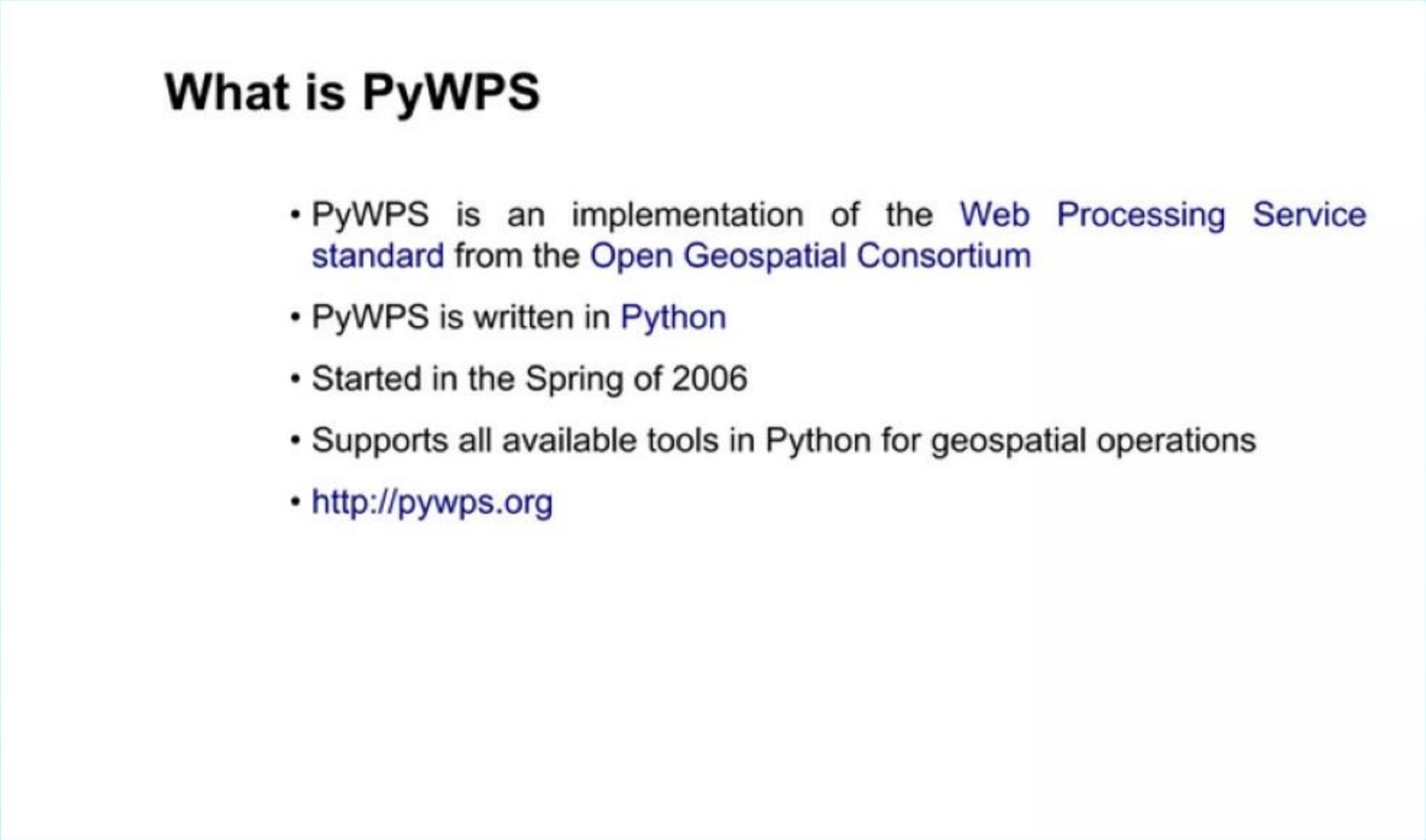
WPS Python Implementation

Lecturer: Mohsen Yousefzadeh

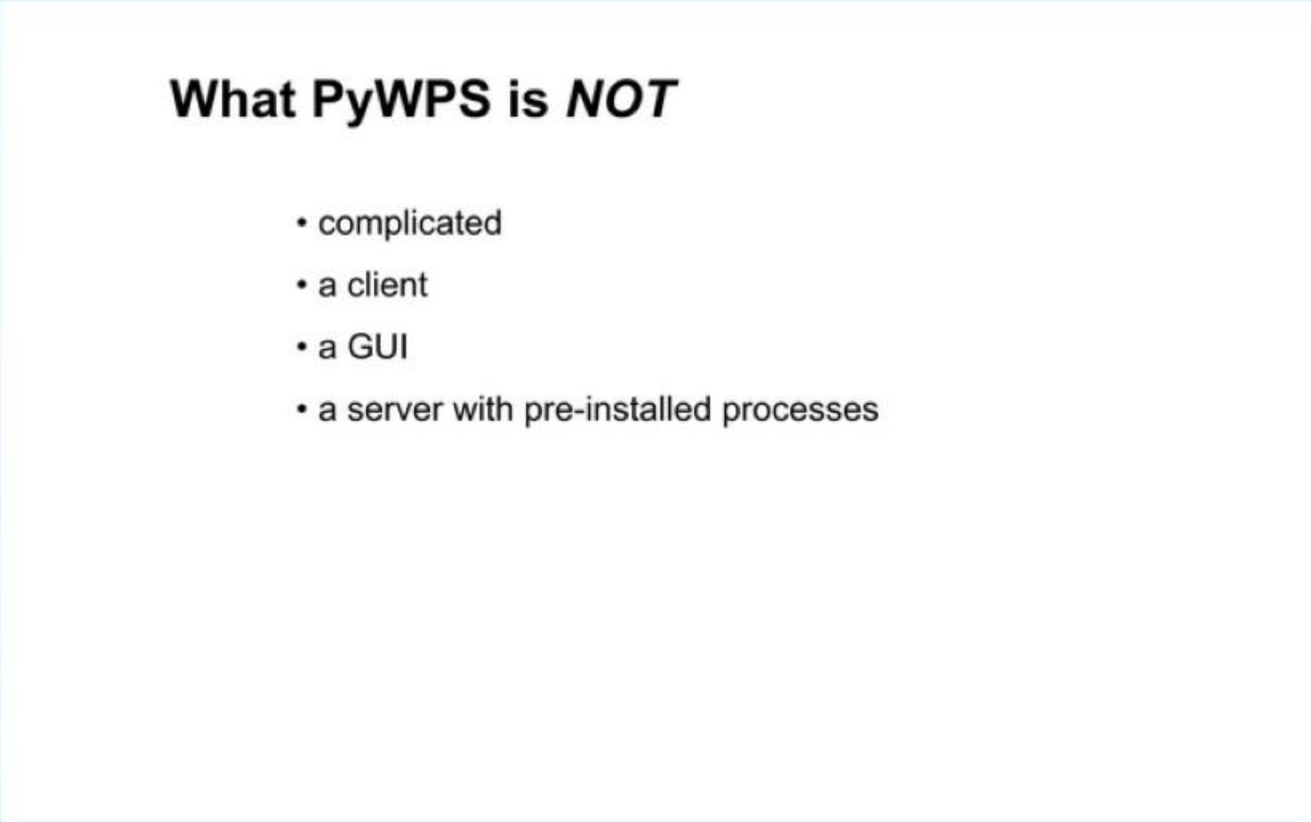
NCC – GIS Division

OCT - 2025



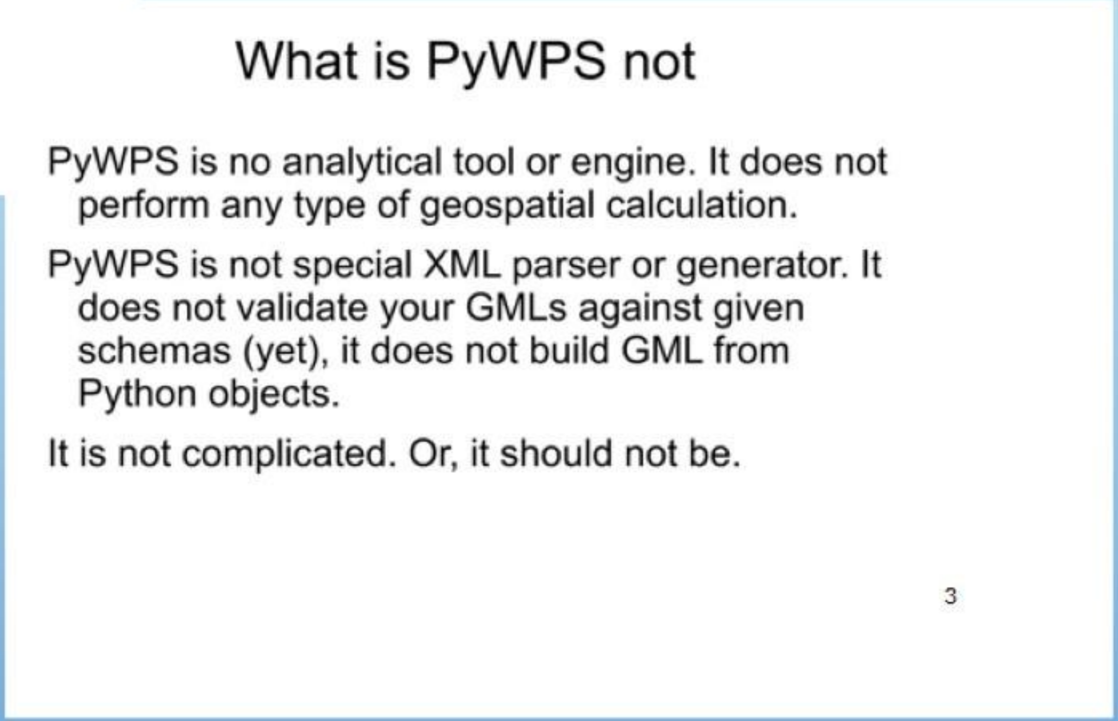
What is PyWPS

- PyWPS is an implementation of the Web Processing Service standard from the Open Geospatial Consortium
- PyWPS is written in Python
- Started in the Spring of 2006
- Supports all available tools in Python for geospatial operations
- http://pywps.org



What PyWPS is *NOT*

- complicated
- a client
- a GUI
- a server with pre-installed processes

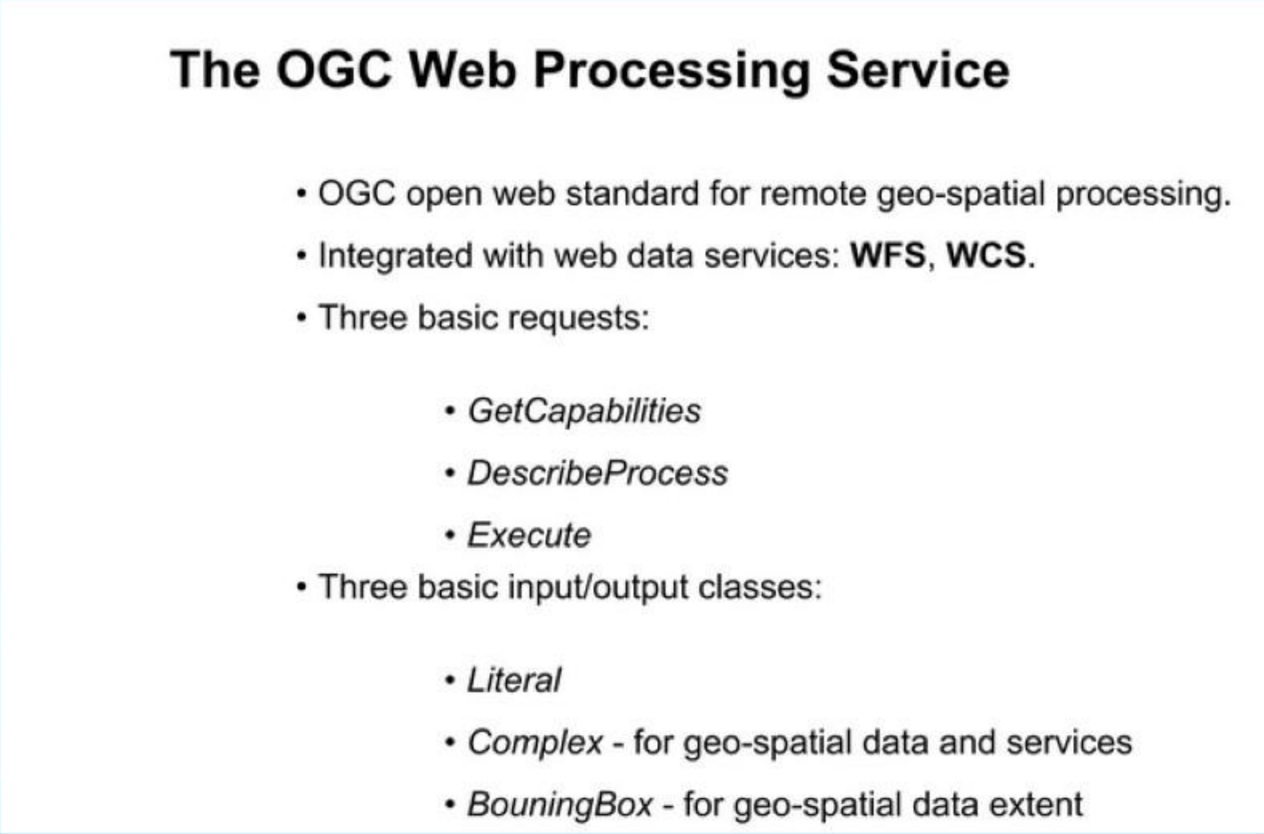


What is PyWPS not

PyWPS is no analytical tool or engine. It does not perform any type of geospatial calculation.

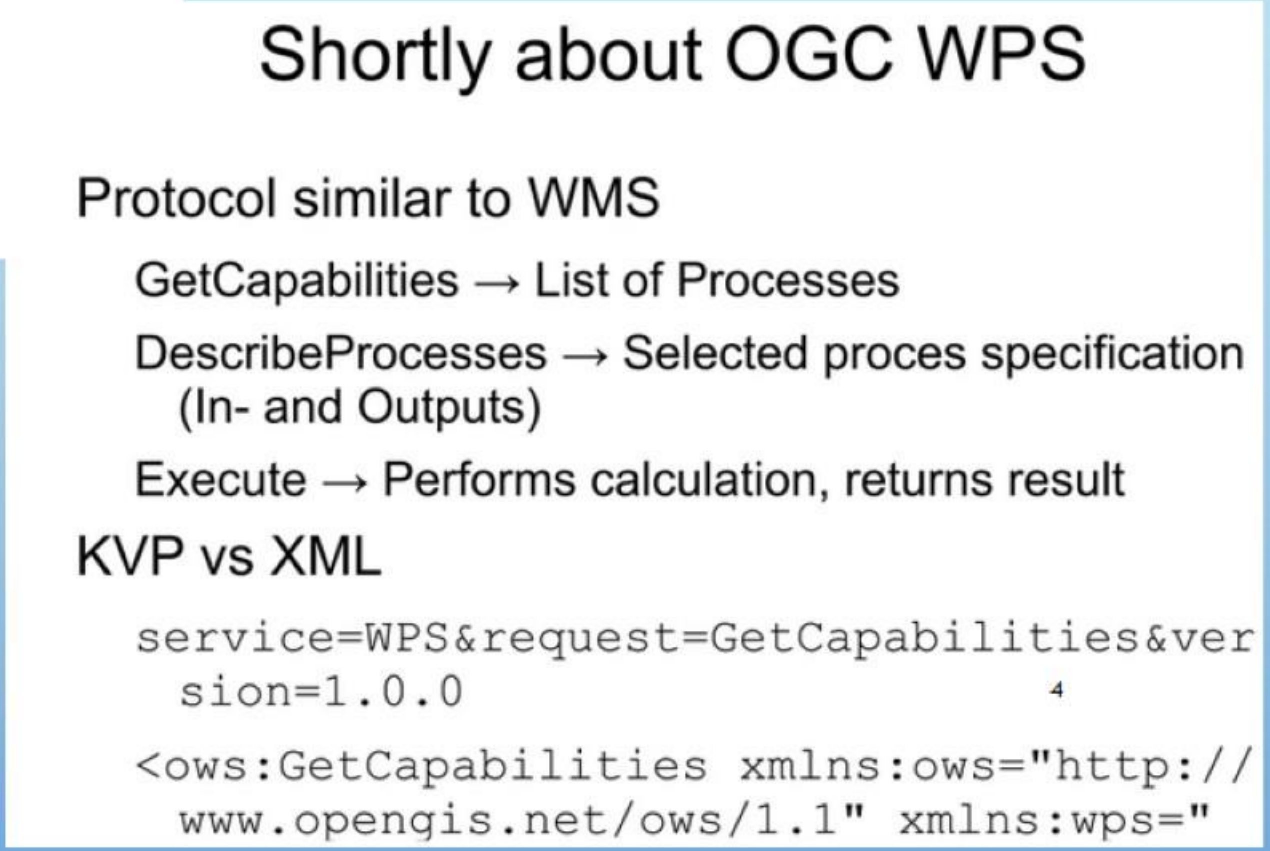
PyWPS is not special XML parser or generator. It does not validate your GMLs against given schemas (yet), it does not build GML from Python objects.

It is not complicated. Or, it should not be.



The OGC Web Processing Service

- OGC open web standard for remote geo-spatial processing.
- Integrated with web data services: **WFS**, **WCS**.
- Three basic requests:
  - *GetCapabilities*
  - *DescribeProcess*
  - *Execute*
- Three basic input/output classes:
  - *Literal*
  - *Complex* - for geo-spatial data and services
  - *BouningBox* - for geo-spatial data extent



Shortly about OGC WPS

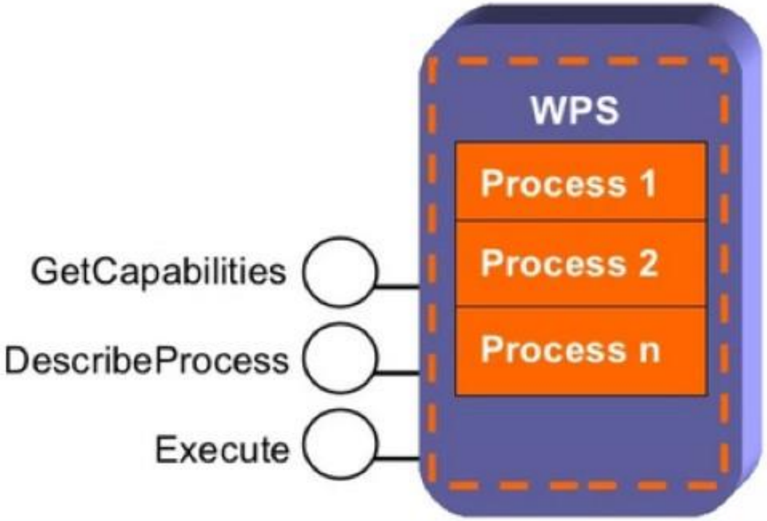
- Protocol similar to WMS
  - GetCapabilities → List of Processes
  - DescribeProcesses → Selected proces specification (In- and Outputs)
  - Execute → Performs calculation, returns result
- KVP vs XML

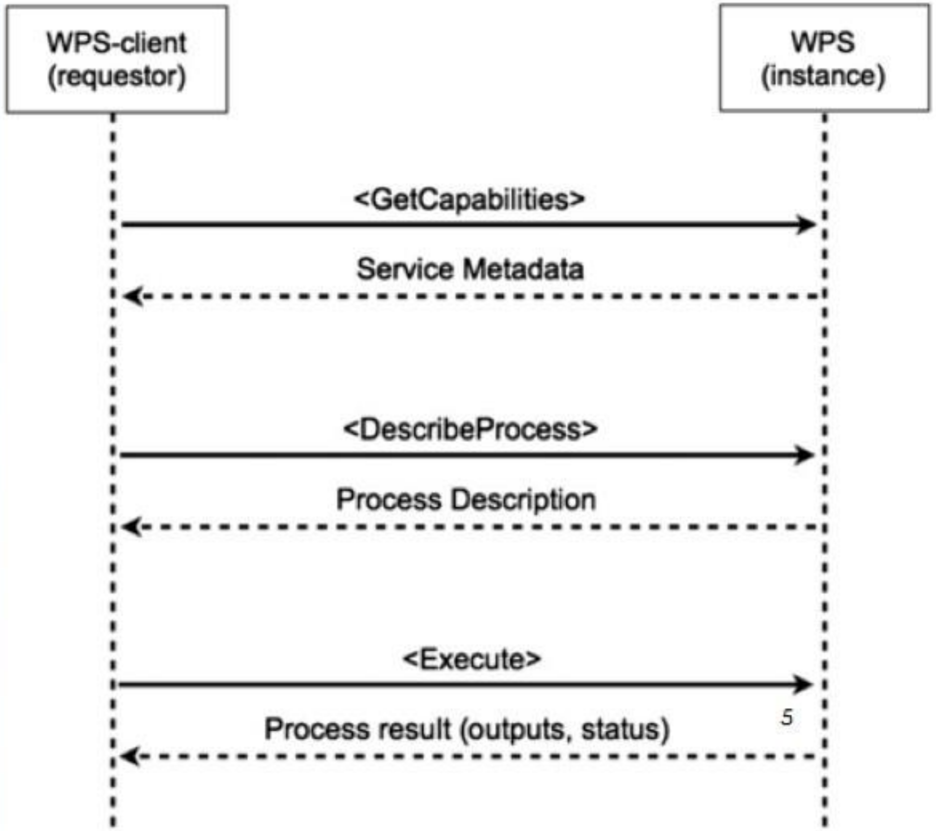
```
service=WPS&request=GetCapabilities&version=1.0.0
```

```
<ows:GetCapabilities xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:wps="
```

The OGC Web Processing Service

Data → Information





LiteralData

LiteralData can be any character string, float,date etc normally described as Primitive datatype in the W3C XML

WPS standard also allows the use of UOM (Unit of Measures), default values and AllowedValues.

Bounding Box Data

It is used to describe some sort of bounding box area .

The input description must state:

the default coordinate reference system (CRS) used

other CRS supported

```
&bboxInput=71.63,41.75,-70.78,42.90,urn:ogc:def:crs:EPSG:6.6:4326,2
```

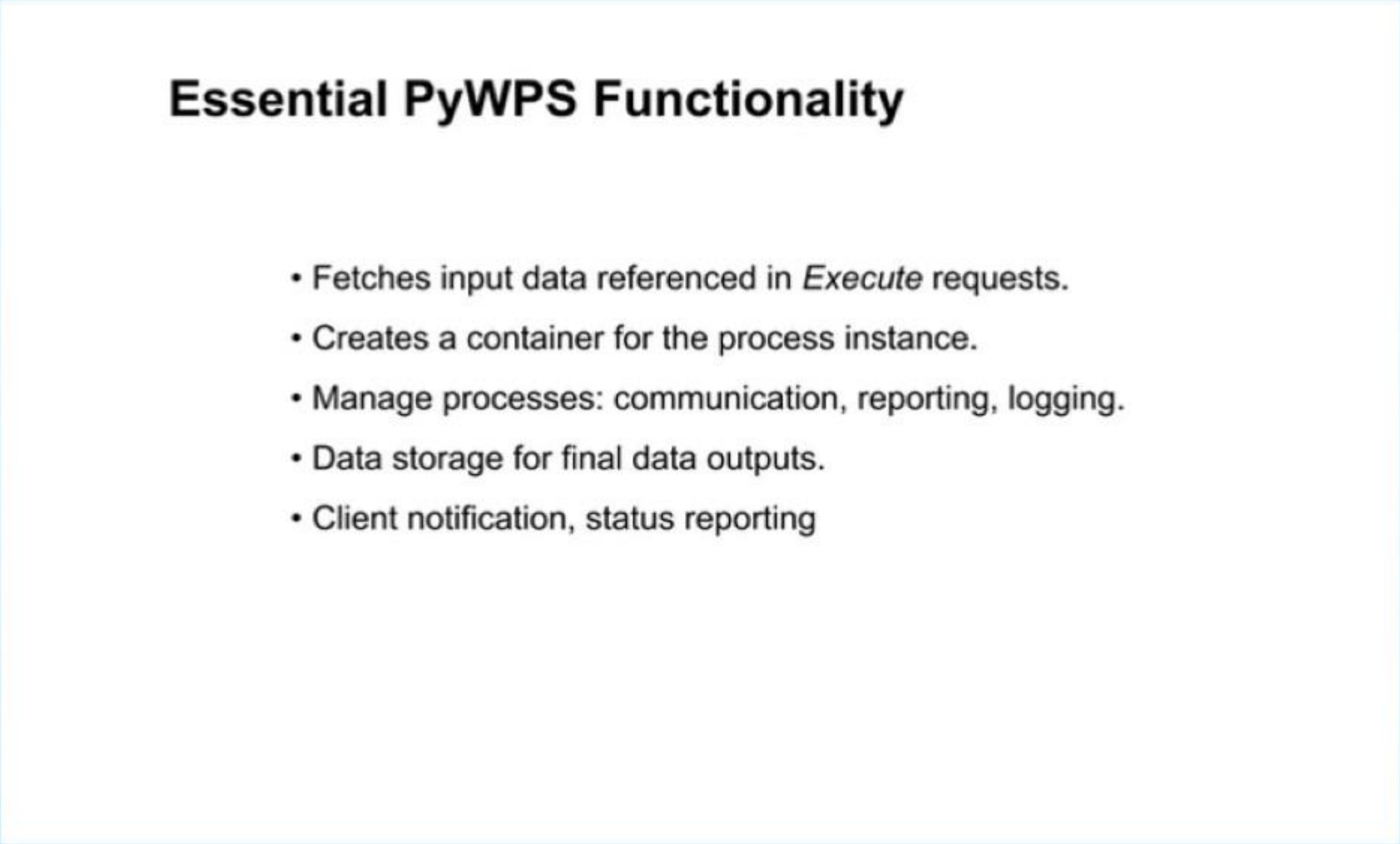
ComplexData

Complex Data data type is used for pasting complex - Vector- Raster- or other data to the server or obtain it as result of the process.

There are two ways, how this complex data are handled:

Either you **send them directly as part of the request** to the server or you obtain them as part of the XML response from the server.

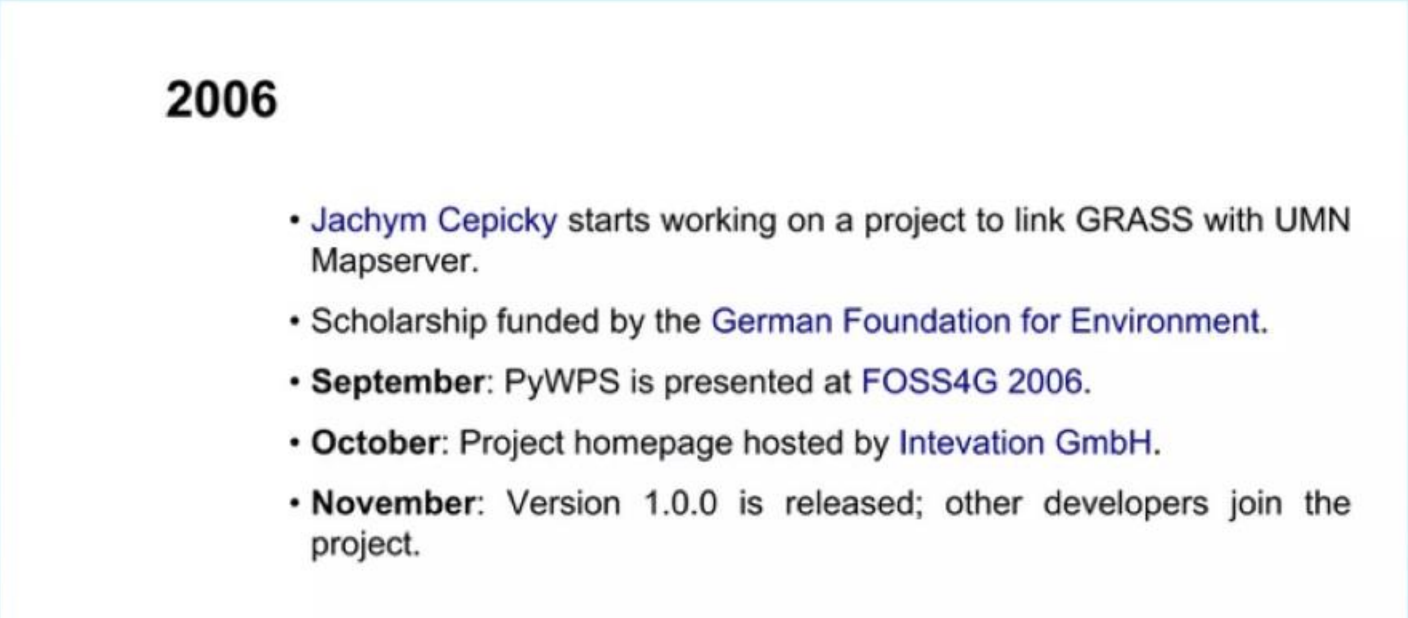
Or you send or obtain just **reference to the data** – URL to the file or service, where the data can be downloaded.



Essential PyWPS Functionality

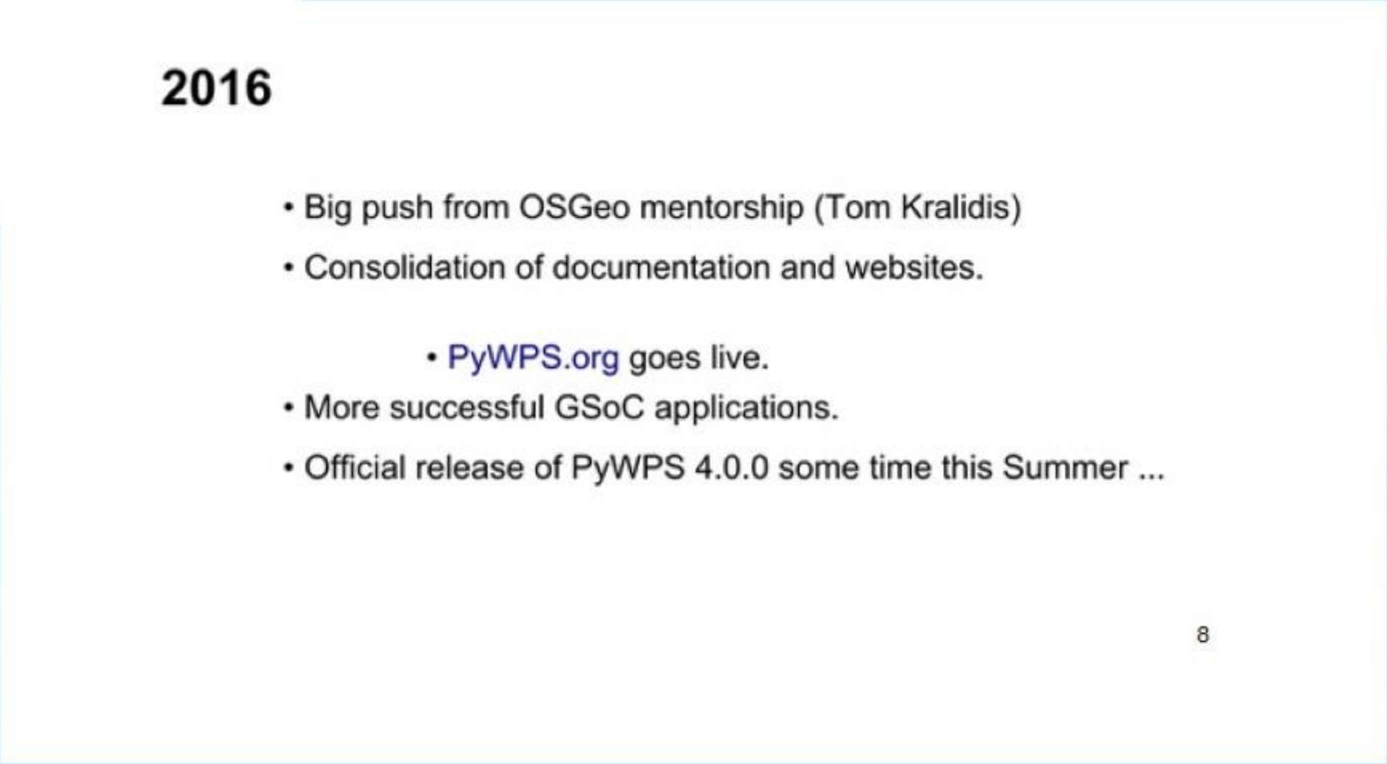
- Fetches input data referenced in *Execute* requests.
- Creates a container for the process instance.
- Manage processes: communication, reporting, logging.
- Data storage for final data outputs.
- Client notification, status reporting

دستورالعمل پیاده‌سازی تحلیل‌های مکانی تحت وب WPS در بستر زیرساخت داده‌های مکانی



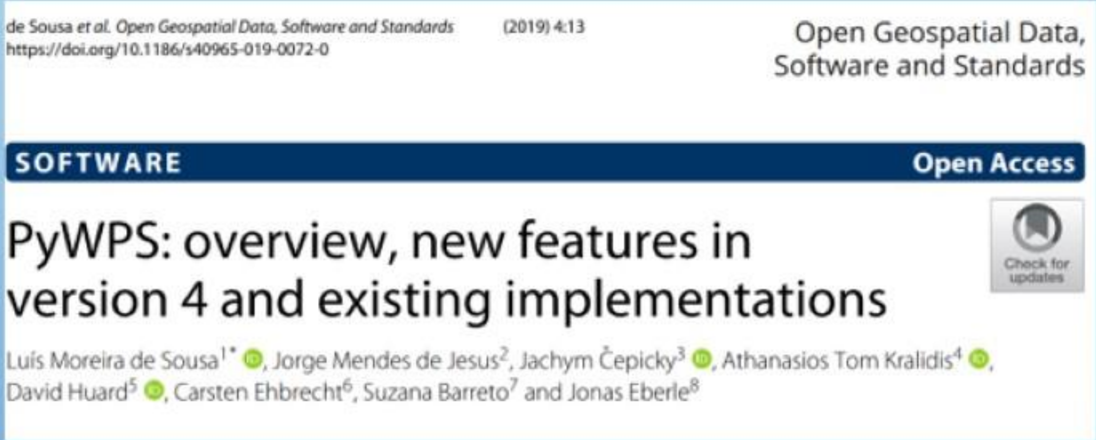
2006

- Jachym Cepicky starts working on a project to link GRASS with UMN Mapserver.
- Scholarship funded by the German Foundation for Environment.
- **September**: PyWPS is presented at FOSS4G 2006.
- **October**: Project homepage hosted by Intevation GmbH.
- **November**: Version 1.0.0 is released; other developers join the project.



2016

- Big push from OSGeo mentorship (Tom Kralidis)
- Consolidation of documentation and websites.
  - PyWPS.org goes live.
- More successful GSoC applications.
- Official release of PyWPS 4.0.0 some time this Summer ...



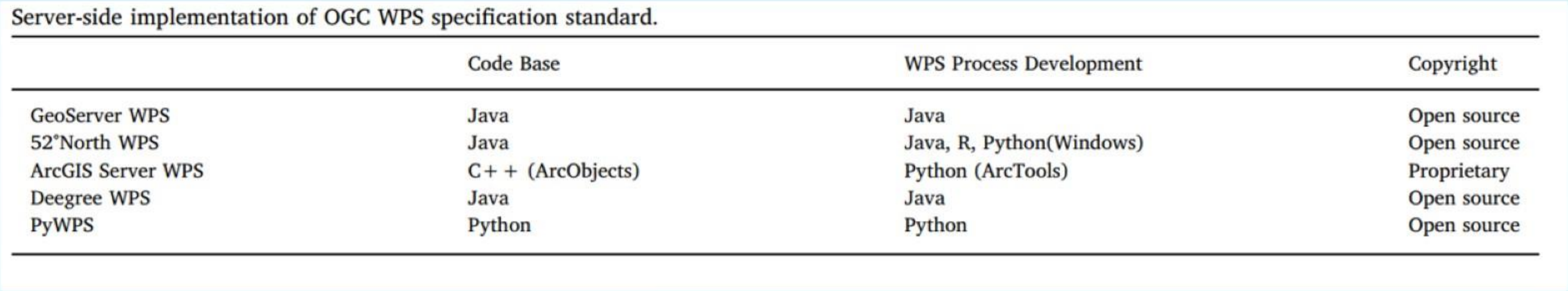
de Sousa *et al. Open Geospatial Data, Software and Standards* (2019) 4:13
https://doi.org/10.1186/s40965-019-0072-0

Open Geospatial Data,
Software and Standards

SOFTWARE **Open Access**

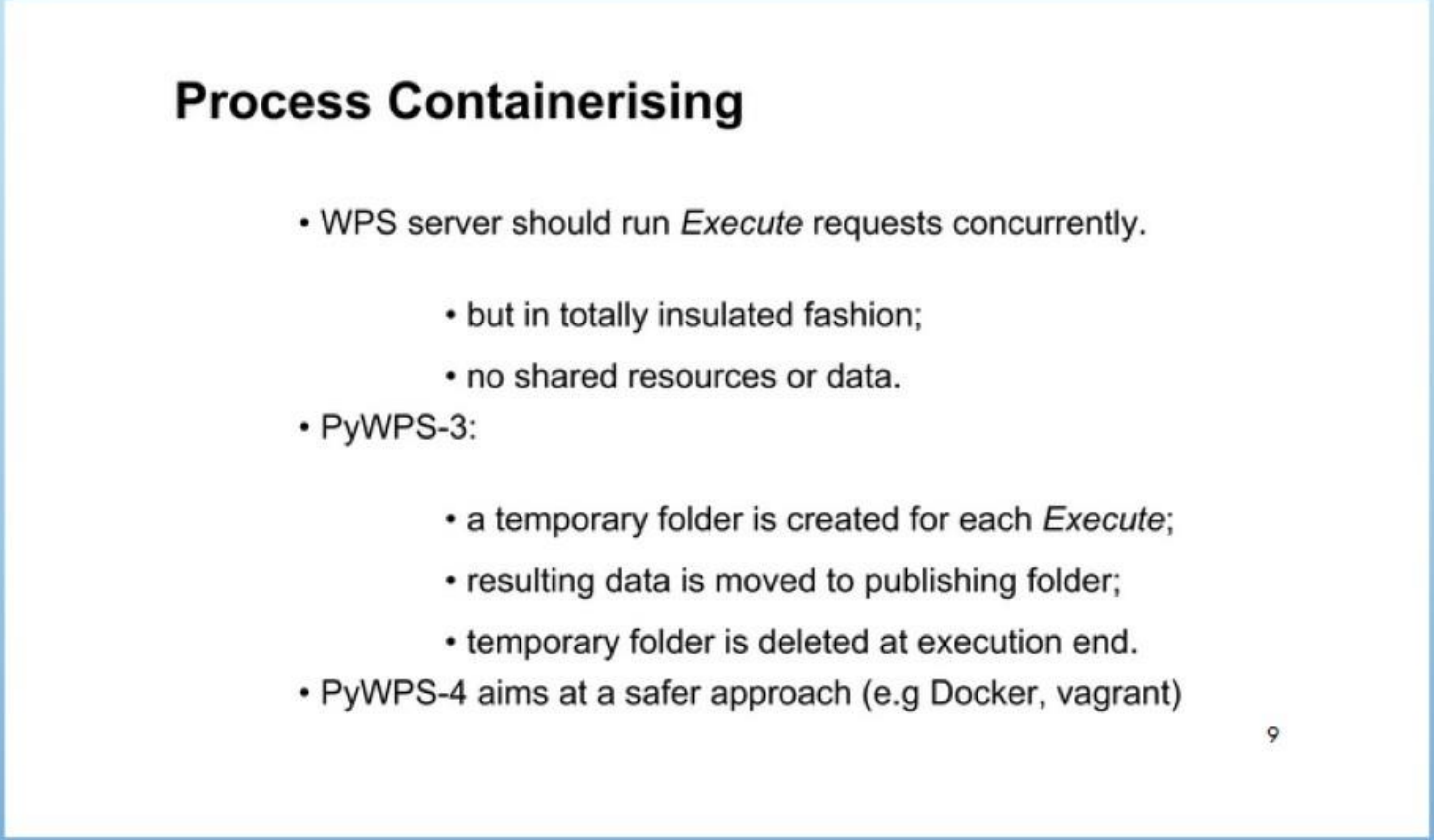
PyWPS: overview, new features in version 4 and existing implementations

Luís Moreira de Sousa[1*], Jorge Mendes de Jesus[2], Jachym Čepicky[3], Athanasios Tom Kralidis[4], David Huard[5], Carsten Ehbrecht[6], Suzana Barreto[7] and Jonas Eberle[8]



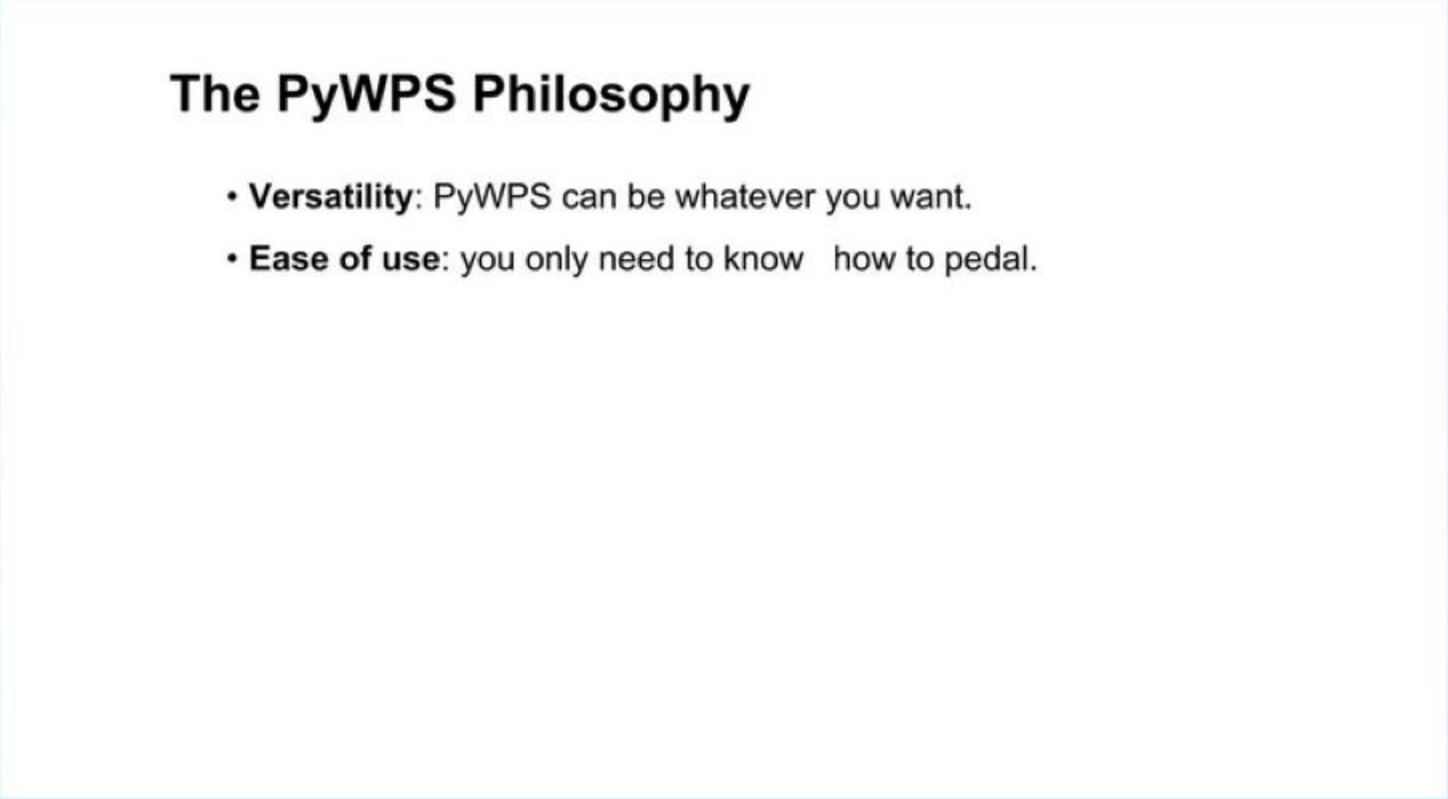
Server-side implementation of OGC WPS specification standard.

	Code Base	WPS Process Development	Copyright
GeoServer WPS	Java	Java	Open source
52°North WPS	Java	Java, R, Python(Windows)	Open source
ArcGIS Server WPS	C+ + (ArcObjects)	Python (ArcTools)	Proprietary
Deegree WPS	Java	Java	Open source
PyWPS	Python	Python	Open source



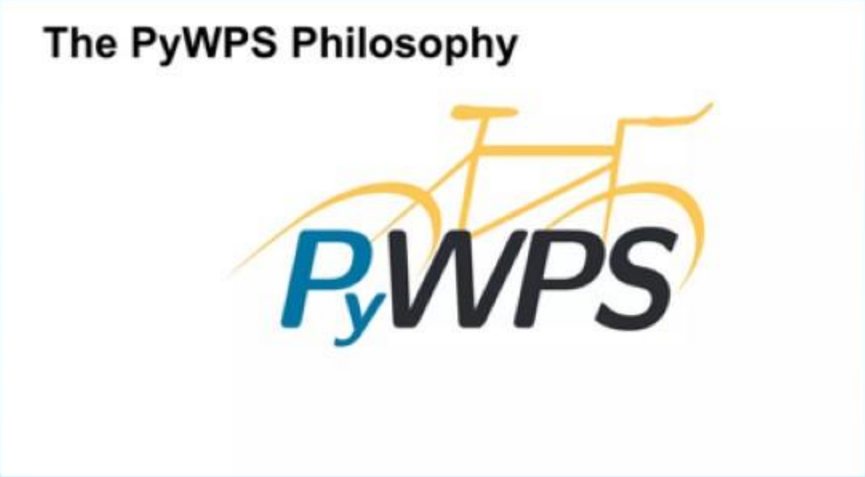
Process Containerising

- WPS server should run *Execute* requests concurrently.
  - but in totally insulated fashion;
  - no shared resources or data.
- PyWPS-3:
  - a temporary folder is created for each *Execute*;
  - resulting data is moved to publishing folder;
  - temporary folder is deleted at execution end.
- PyWPS-4 aims at a safer approach (e.g Docker, vagrant)



The PyWPS Philosophy

- **Versatility**: PyWPS can be whatever you want.
- **Ease of use**: you only need to know how to pedal.



The PyWPS Philosophy

Bikes

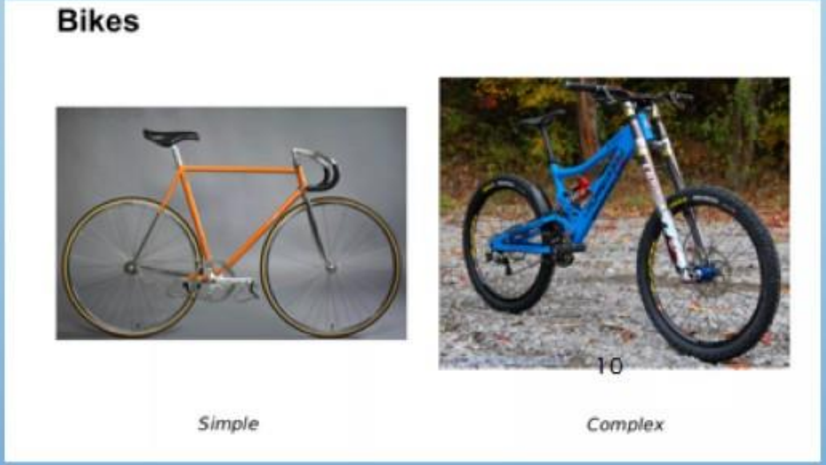
Large

Small

Bikes

Cheap

Expensive



Bikes

Simple

Complex

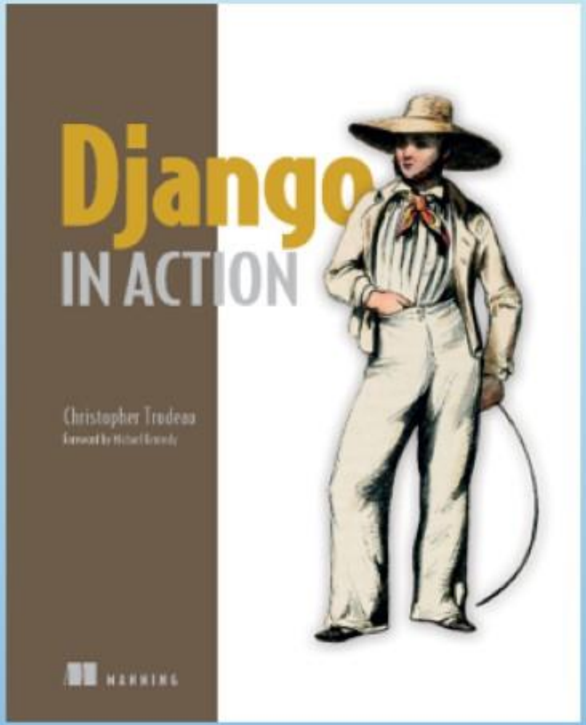
Examples In The Real World (Spatial & non-Spatial)

1. SayHello
2. Buffer
3. Intersect
4. DWithin
5. Grid (GRIDSDI)

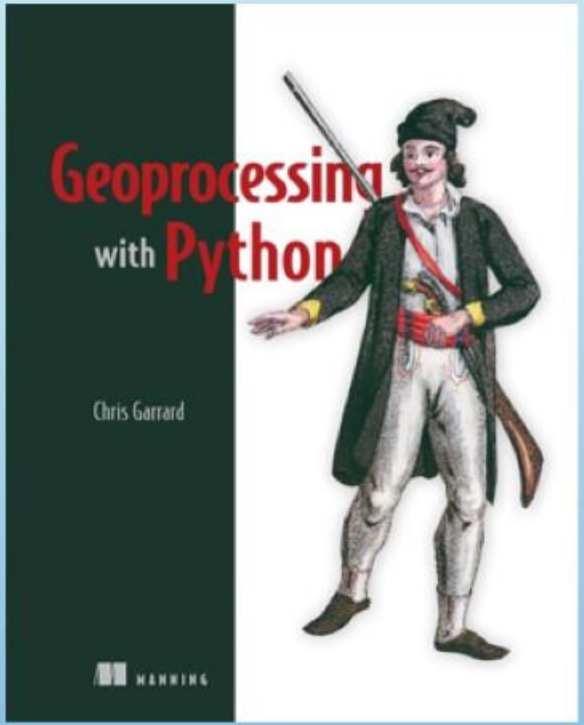
References

1. PyWPS: overview, new features in version 4 and existing implementations (Springer)
2. https://www.slideshare.net/slideshow/pywps-64463325/64463325#3
3. WPS mediation: An approach to process geospatial data on different computing backends (Elsevier)
4. https://www.slideshare.net/slideshow/pywps/8059604
5. Simplifying the deployment of OGC web processing services (WPS) for environmental modelling–Introducing Tethys WPS Server (Elsevier)

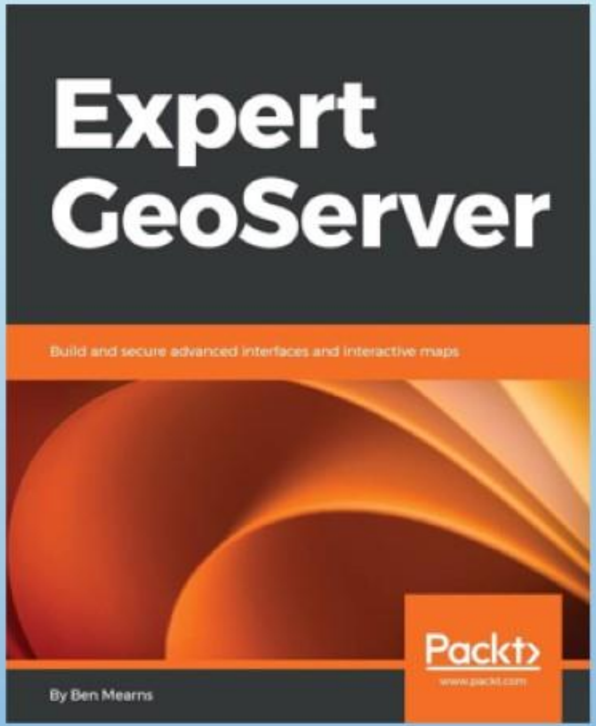
For more information



2024



2016



2018

Thank you!
Q&A

Contant info:

Gmail: *mail.myousefzadeh@gmail.com*

Phone: *021-63184015*

Bale id: *@geocommits*