

# **INTRODUCTION TO WEB PROCESSING SERVICES**

Benjamin Pross, Christoph Stasch


52°North GmbH

Geospatial Sensor Web Conference, 2018-09-03

# OVERVIEW

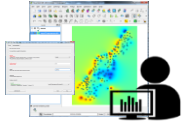

- Web-based Geoprocessing
  - Why and how?
  - OGC WPS
- Implementations & Details about the 52°North WPS
- Example applications

# MAIN FOCUS OF 52°NORTH




**Sensor Web  
Enablement**


## Geoprocessing



Desktop Apps



Web Apps



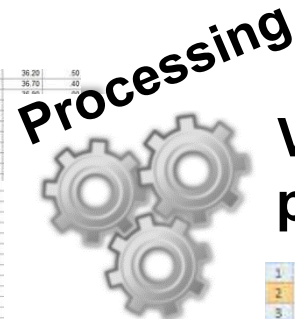
SDIs, SOA, Big Data

# **PROBLEM: WHY WEB-BASED GEOPROCESSING AND WPS?**

# GEOPROCESSING

Raw data

0.00	18.00	36.00	36.70	36.80	36.10	36.20	36.30	36.70	36.20	60	
1.00	22.00	37.10	37.10	37.10	36.70	36.80	36.60	37.10	36.70	40	
1.00	19.00	36.70	36.80	37.10	37.00	36.80	36.80	36.90			
1.00	19.00	36.80	36.90	36.90	35.80	35.80	36.00	36.80			
1.00	18.00	36.70	36.90	36.80	36.30	36.50	36.20	36.80			
1.00	18.00	36.60	36.80	36.70	36.60	36.50	36.60	36.70			
1.00	19.00	37.20	37.40	37.50	36.50	36.50	36.50	37.40			
1.00	19.00	37.00	37.00	37.00	36.90	36.80	36.90	37.00			
2.00	22.00	36.80	37.00	36.80	36.80	36.50	36.80	36.90			
1.00	21.00	37.00	37.00	36.90	36.90	36.20	36.30	37.00			
2.00	18.00	37.00	37.40	36.80	36.10	36.10	36.10	37.00			
1.00	1.00	18.00	36.60	36.70	36.80	36.10	36.20	36.30			
1.00	1.00	22.00	37.10	37.10	37.10	36.70	36.80	36.60			
1.00	1.00	19.00	36.70	36.80	37.10	37.00	36.80	36.80			
2.00	1.00	19.00	36.80	36.80	36.90	35.80	35.80	36.00			
1.00	1.00	18.00	36.70	36.90	36.80	36.30	36.50	36.20			
1.00	1.00	18.00	36.60	36.80	36.70	36.60	36.50	36.60			
1.00	1.00	19.00	37.20	37.40	37.50	36.50	36.50	36.50			
1.00	1.00	19.00	37.00	37.00	37.00	36.90	36.80	36.90			
2.00	1.00	22.00	36.80	37.00	36.80	36.80	36.50	36.80			
1.00	1.00	21.00	37.00	37.00	36.90	36.90	36.20	36.30			
2.00	1.00	18.00	37.20	37.40	37.40	36.40	36.70	36.30			
1.00	1.00	1.00	37.60	37.60	37.60	37.20	36.20	37.10			
2.00	1.00	24.00	35.50	35.90	35.90	36.40	36.60	36.70			
2.00	1.00	18.00	36.60	37.20	36.80	37.00	37.00	37.00			
1.00	1.00	20.00	36.50	36.50	36.50	36.30	36.50	36.20			
2.00	1.00	1.00	36.80	36.80	36.80	36.20	36.20	36.70	36.20	50	
1.00	1.00	20.00	37.10	36.80	36.80	36.80	36.70	36.90	36.80	10	
2.00	1.00	22.00	37.20	37.40	37.40	36.30	37.10	37.20	37.30	36.90	10
2.00	1.00	19.00	36.40	36.70	36.70	36.10	35.70	36.10	36.60	36.80	10
1.00	1.00	18.00	36.60	36.80	36.80	36.90	36.90	36.70	36.70	36.80	20
1.00	1.00	24.00	37.10	37.30	37.30	37.10	37.30	37.30	37.10	37.20	-10
2.00	1.00	20.00	36.60	36.70	36.70	36.40	36.70	36.80	36.70	36.80	10
2.00	1.00	23.00	36.90	37.00	37.00	36.50	36.70	36.70	36.90	36.80	10
2.00	1.00	17.00	37.00	37.10	37.10	36.80	36.80	36.10	37.00	36.90	10
1.00	1.00	22.00	36.80	36.90	36.90	36.30	36.50	36.30	36.90	36.10	80
1.00	1.00	21.00	37.20	37.30	37.30	36.80	36.80	36.90	37.20	36.90	40
2.00	1.00	19.00	36.80	36.80	36.80	36.50	36.50	36.80	36.80	36.50	30
1.00	1.00	20.00	36.90	37.00	37.00	36.90	36.90	36.90	36.90	36.90	90
2.00	1.00	18.00	37.00	37.10	37.10	36.80	36.80	36.90	37.00	36.90	20
2.00	1.00	18.00	36.90	37.00	37.00	36.00	36.20	36.90	36.10	80	



Value-added data products

1	Location	Tamworth, UK	
2	Date	May-09	
3			
4			
5	Date	Temperature	Rain
6	1	13.2	0.25
7	2	13.1	0
8	3	11	0
9	4	9	0.25
10	5	13.8	2
11	6	14.1	0
12	7	11.8	0.5
13	8	11.9	1
14	9	11.8	0
15	10	14.1	0
16	11	10.8	0
17	12	10	0
18	13	11.3	0.5
19	14	12	0.5
20	15	11	14.5
21	16	12.2	14

Temp

25

20

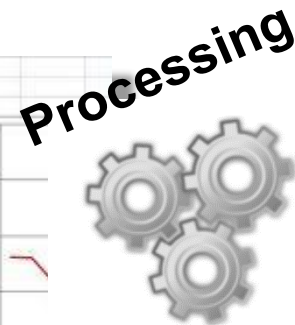
15

10

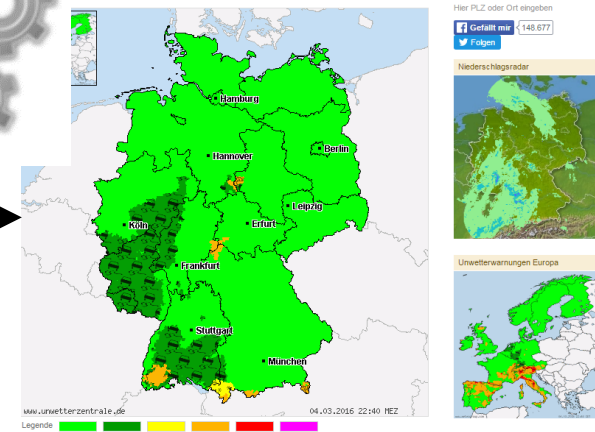
5

0

1 2 3 4 5

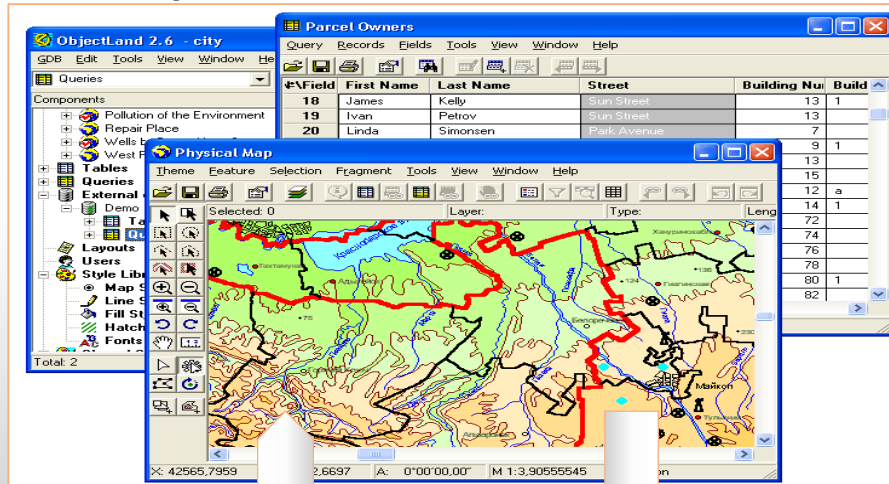


Information products



# GEOPROCESSING – EARLIER APPROACH

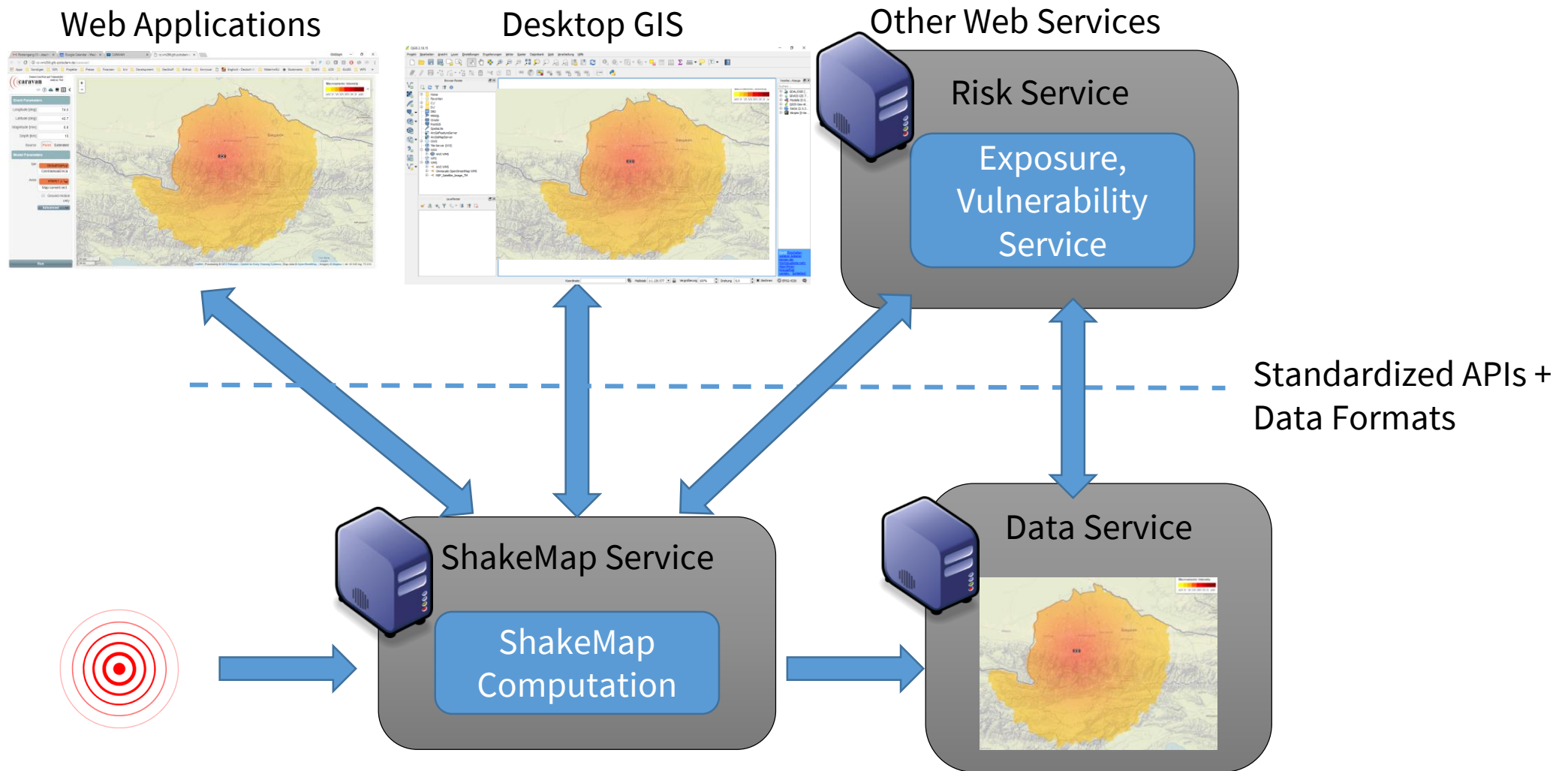
## Desktop GIS



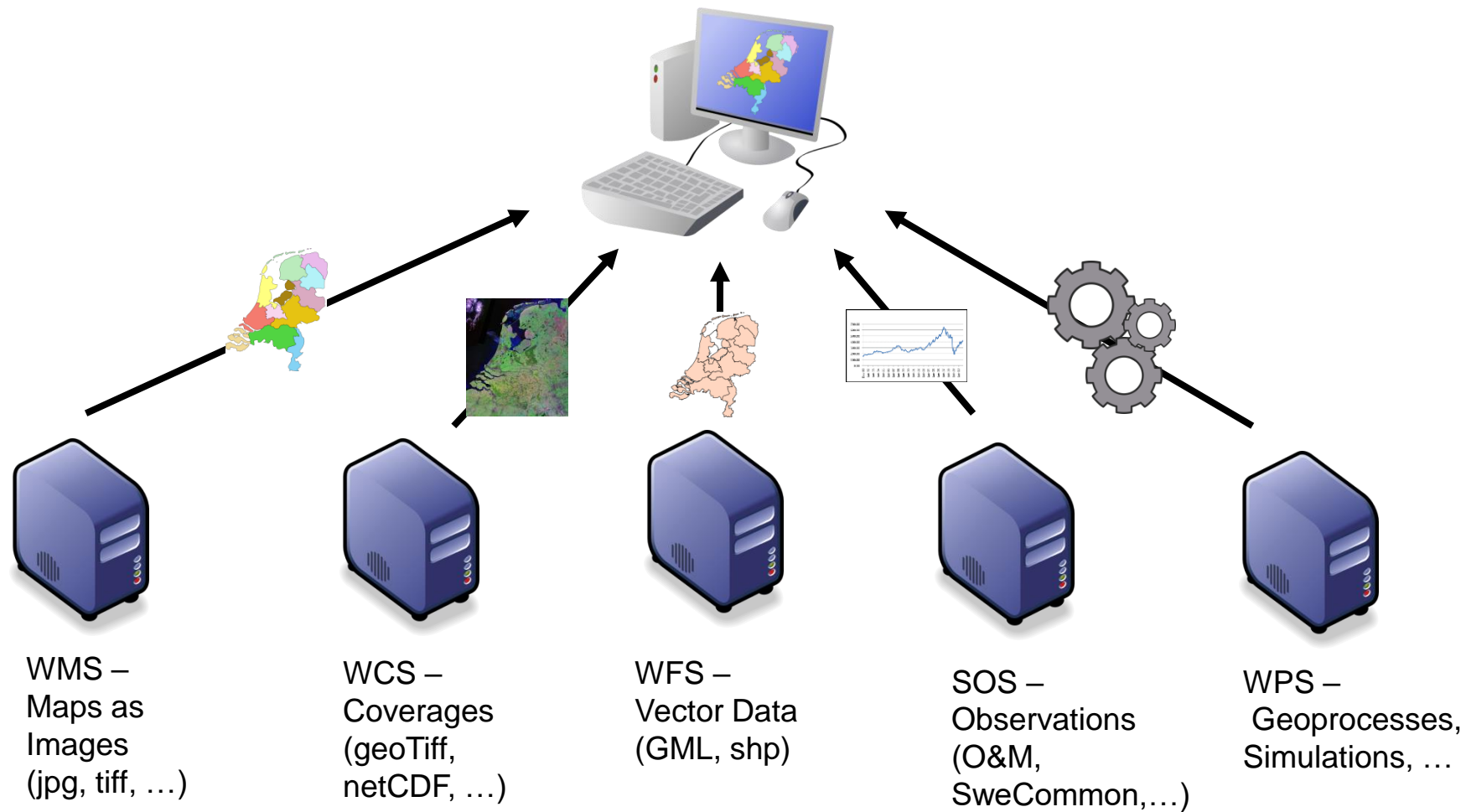
Input data

Output data

# WEB SERVICE - APPROACH

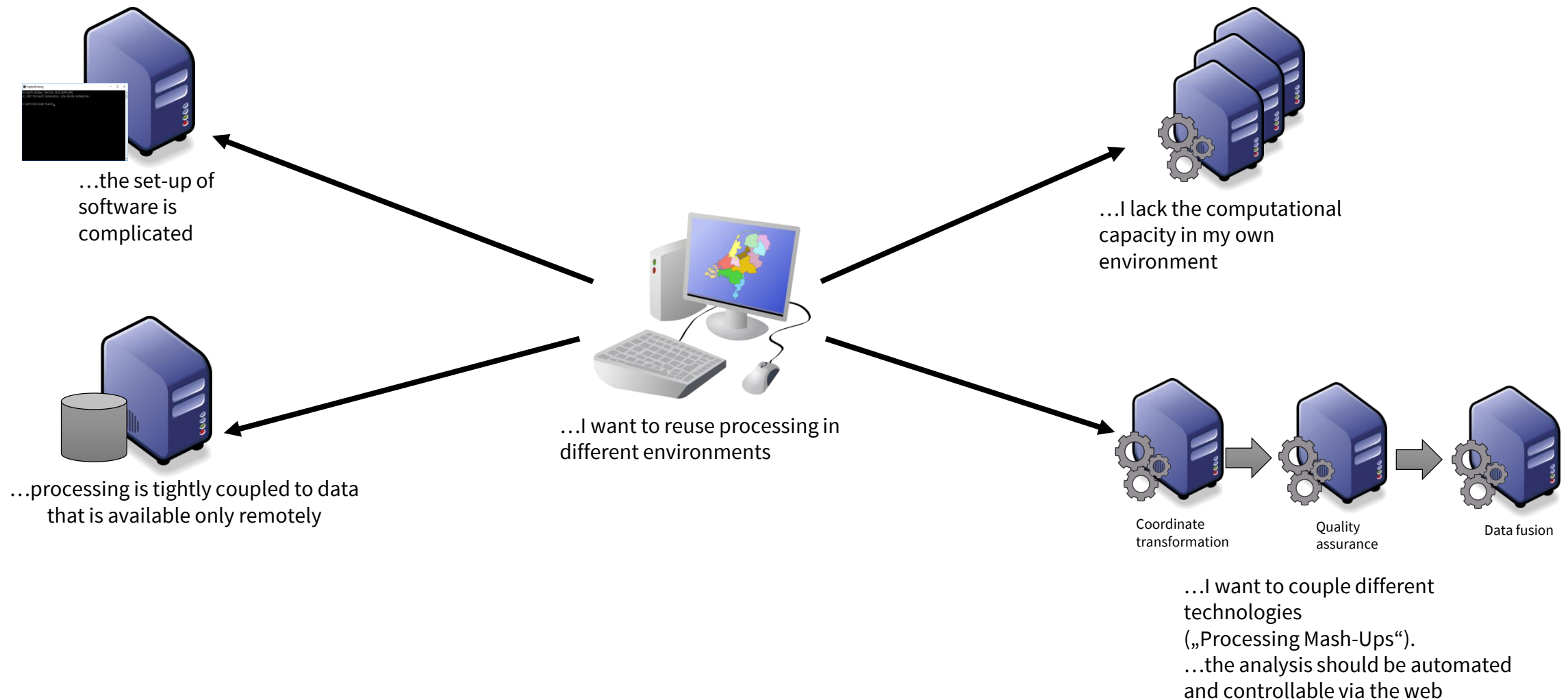


# OGC Web Processing Service (WPS)





# GEOPROCESSING IN THE WEB, BECAUSE...

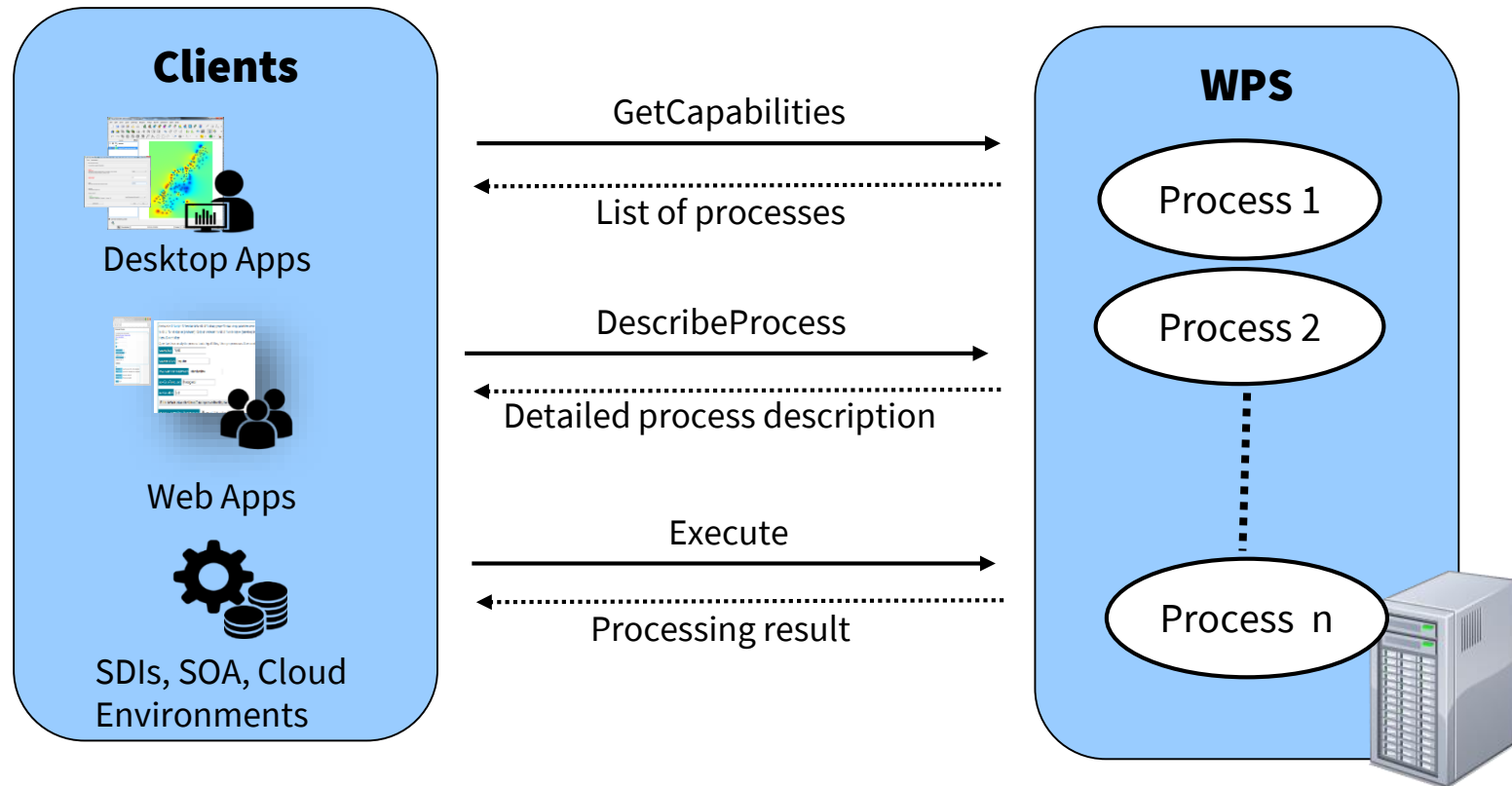


# **OVERVIEW: WHAT IS THE OGC WPS?**

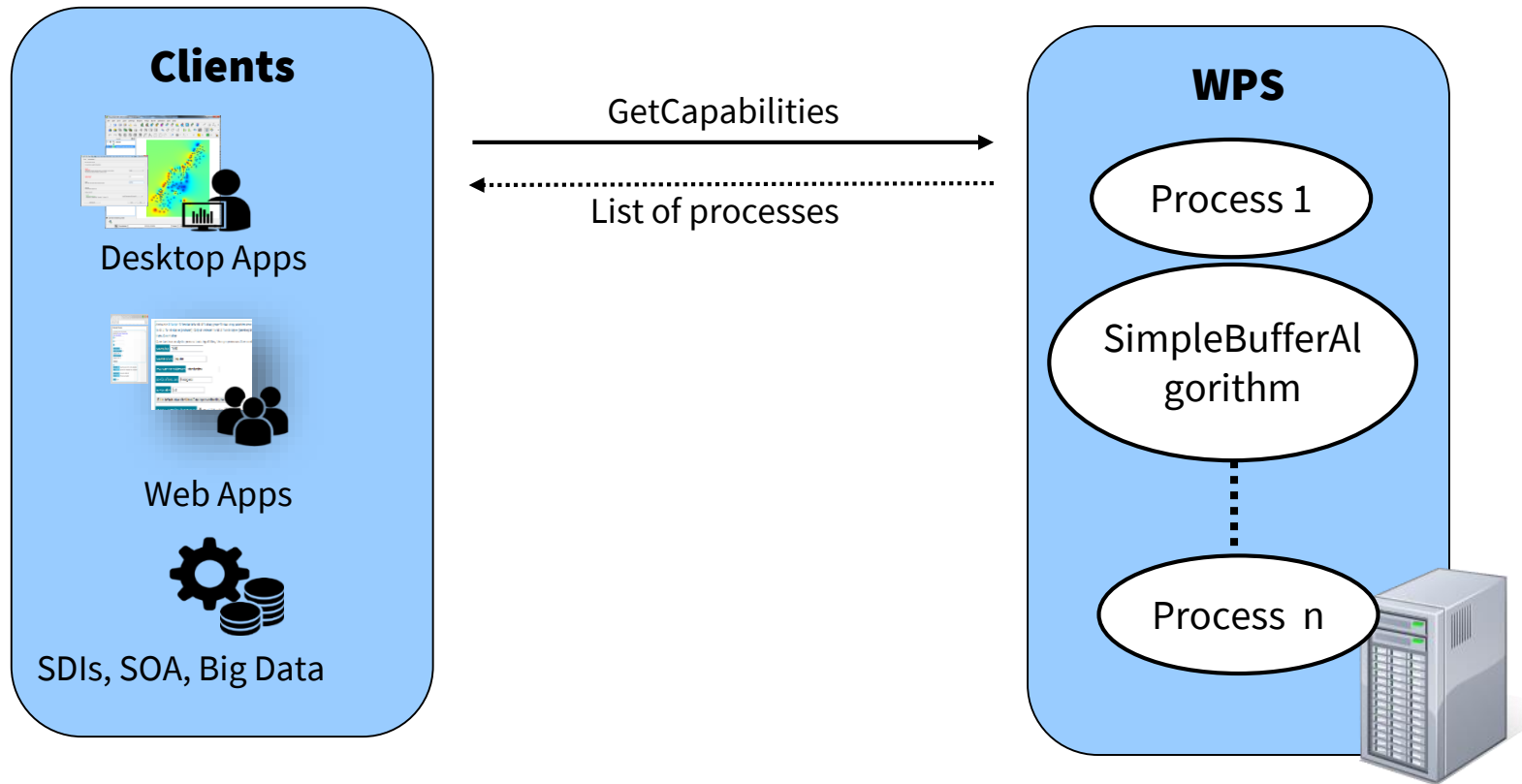
# OGC WEB PROCESSING SERVICE - OVERVIEW

- Official OGC Standard since 2007, Version 2.0.0 since 2015:
  - <http://www.opengeospatial.org/standards/wps>
- Standardized description of geoprocessing functionality („processes“)
  - Identifier
  - Textual description
  - Input and output parameters
- Predefined service operations for the description and execution of processes (synchronous, asynchronous)
  - DescribeProcess, Execute, GetResult
- Software:
  - 52°North WPS, PyWPS, Zoo WPS, ArcGIS Server, ERDAS Imagine, ...

# OGC Web Processing Service – BASIC OPERATIONS



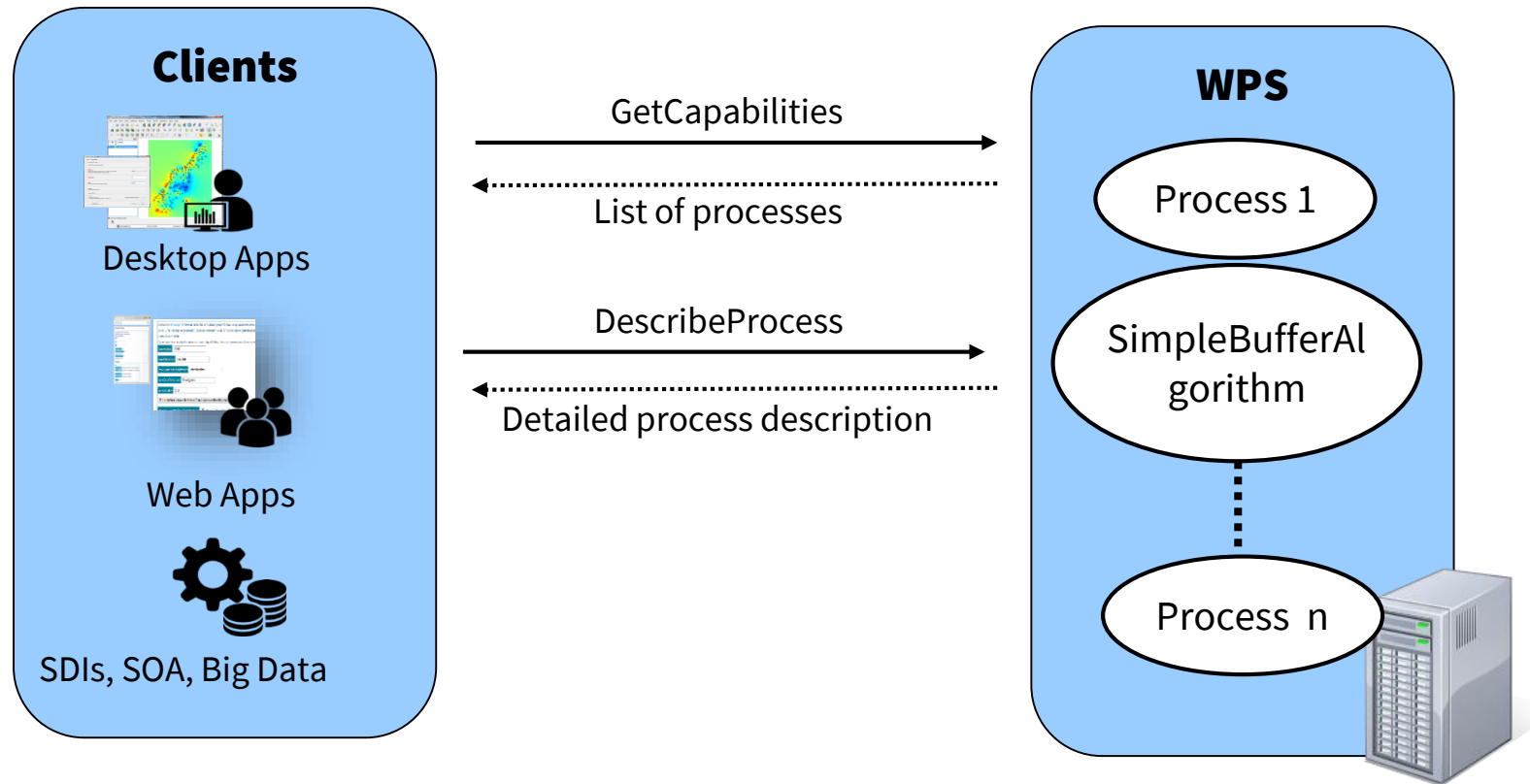
# WPS – WHICH PROCESSES ARE AVAILABLE?



# WPS – GETCAPABILITIES OPERATION

- Request via URL (HTTP GET with Key-Value-Pair encoding) or XML Request (HTTP POST)
- Returns service description of the WPS
- Basic information:
  - Endpoints
  - Technical request mechanisms
  - Information about the service provider/access constraints
- Short information about the offered processes

# WPS – WHAT DOES THE PROCESS DO? WHAT ARE THE INPUTS/OUTPUTS?

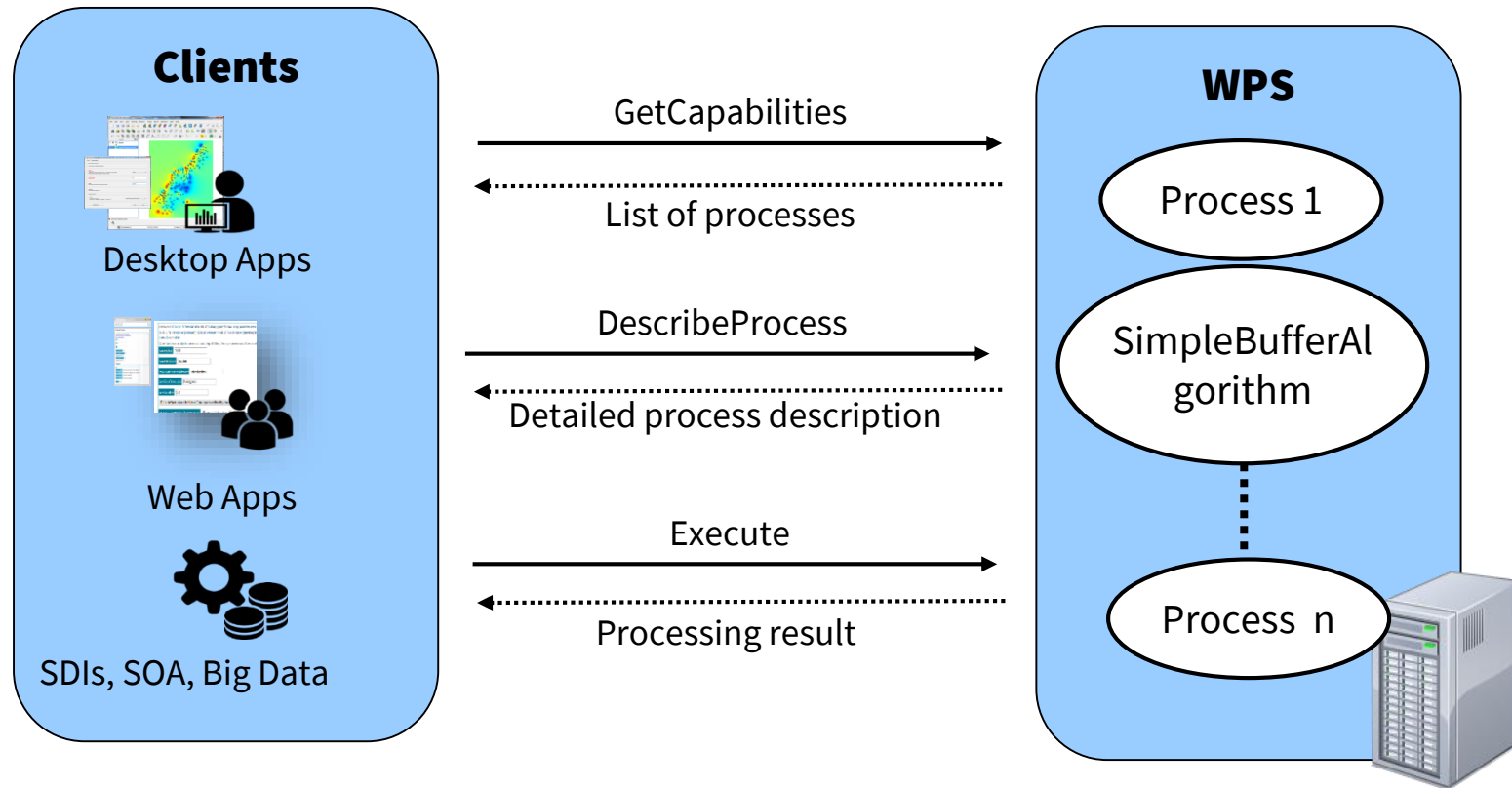


# WPS: DESCRIBEPROCESS OPERATION

- Request via URL (HTTP GET with Key-Value-Pair encoding) or XML Request (HTTP POST)
- Returns detailed description of a process based on the process id
- Defines the inputs and outputs
  - IDs
  - Default data formats
  - Further supported data formats



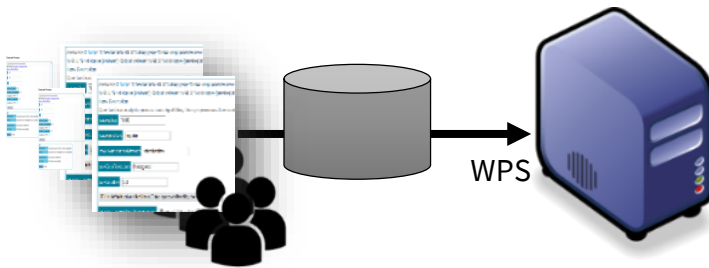
# WPS – HOW DO I EXECUTE A PROCESS?



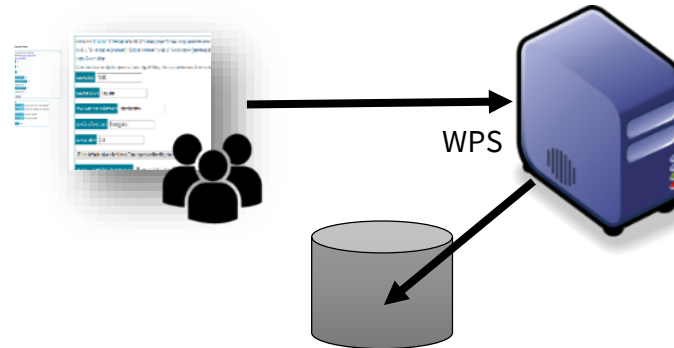
# WPS – EXECUTE OPERATION

- XML Request (HTTP POST)
- Execution of an offered process
- Request:
  - Must contain id and input parameters according to the process description
- Result
  - Can be returned directly or as reference to a web accessible resource
- Can be executed asynchronously for long running processes -> Client doesn't directly get the result, but can request the status of the execution

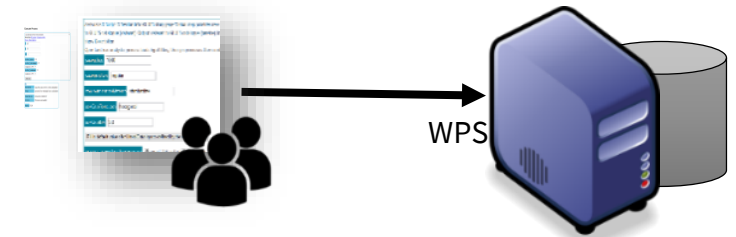
# WPS – HOW TO TRANSFER THE INPUT DATA



Option 1: Direct transfer



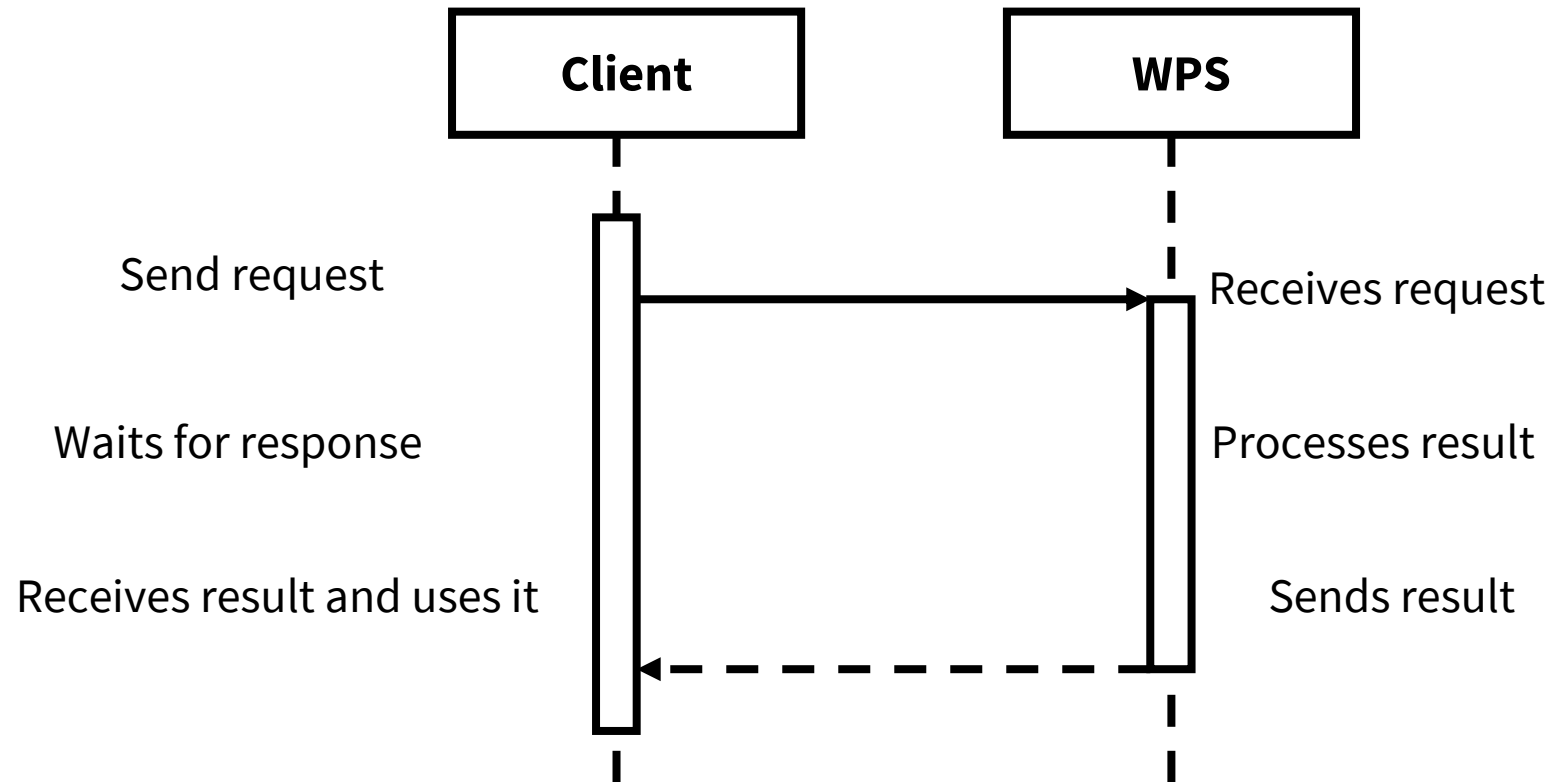
Option 2: Reference to web accessible resource



Option 3: Query on encapsulated data

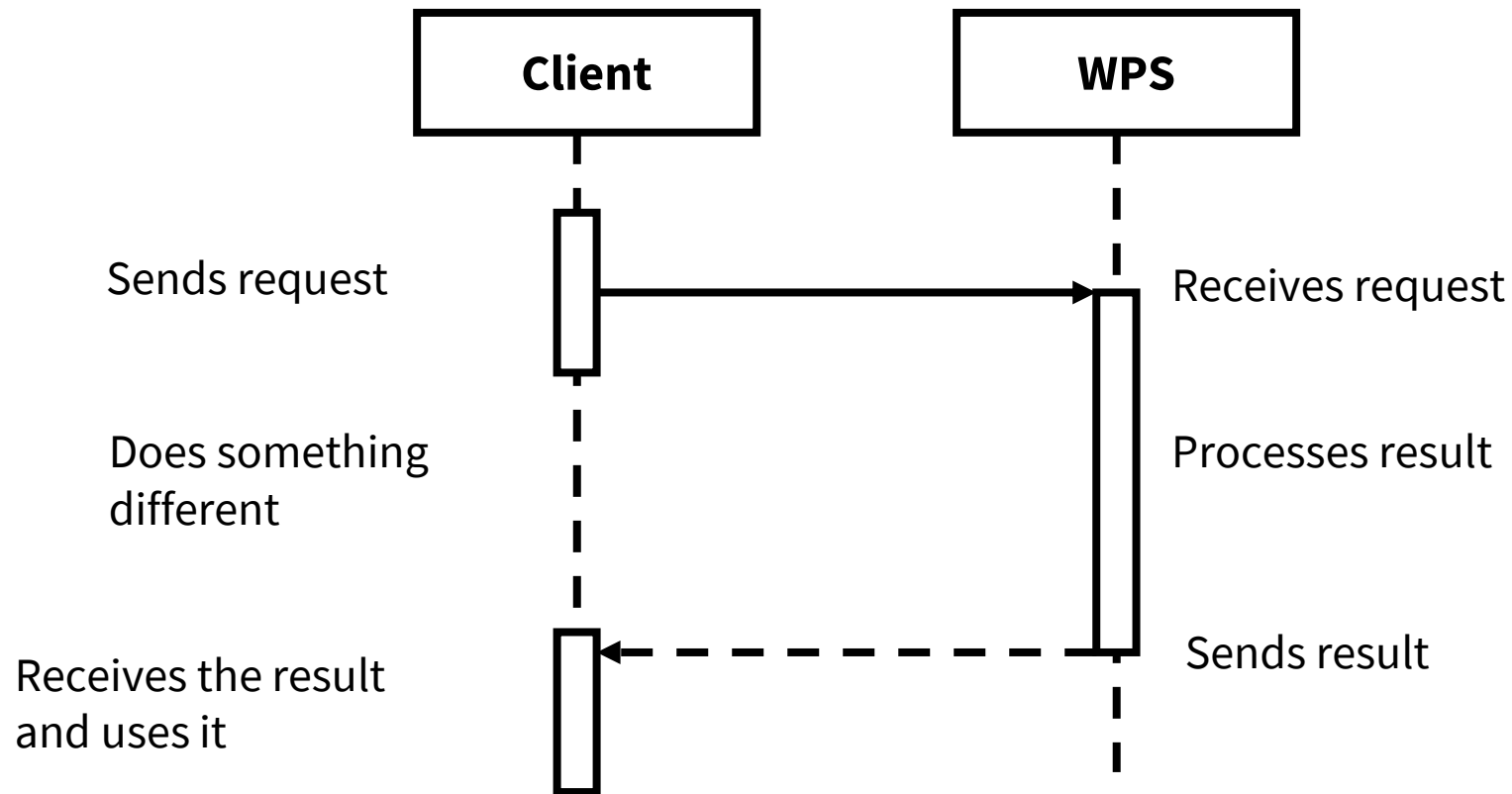
# **ASYNCHRONOUS EXECUTION: HOW DO I EXECUTE LONG RUNNING PROCESSES**

# WPS – SYNCHRONOUS EXECUTION

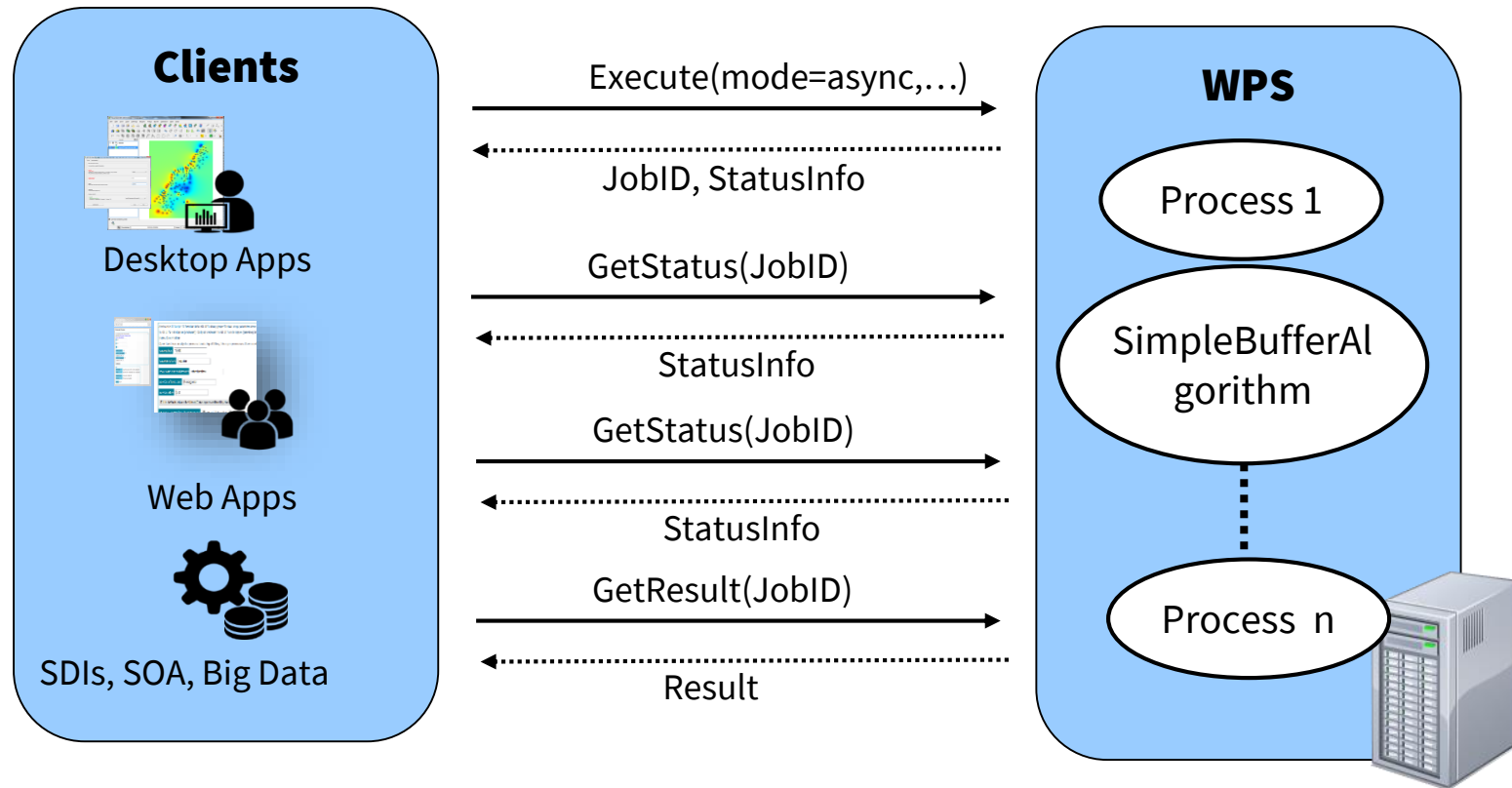


## WPS – ASYNCHRONOUS EXECUTION(PUSH-MODEL)

Sequence Diagram



# WPS SPEC 2.0 – ASYNCHRONOUS EXECUTION



# **EXAMPLE APPLICATIONS: WHERE ARE WPS USED?**



# USGS

Dataset Selection

Search


Algorithms

- Data Subsets
- Areal Statistics

### 4km Monthly Parameter-elevation Regressions on Independent Slopes Model Monthly Climate Data for the Continental United States.

**Abstract**


This dataset was created using the PRISM (Parameter-elevation Regressions on Independent Slopes Mode...



### Bias Corrected Constructed Analogs V2 Daily Climate Projections

**Abstract**


This archive contains projections of daily BCCA CMIP3 and CMIP5 projections of precipitation, daily ...



### Bias Corrected Spatially Downscaled Monthly CMIP5 Climate Projections

**Abstract**


This archive contains 234 projections of monthly BCSD CMIP5 projections of precipitation and monthly...



### Bias Corrected Spatially Downscaled Monthly Climate Predictions

**Abstract**


This archive contains fine spatial-resolution translations of 112 contemporary climate projections o...



### CIG Northern US Rockies and Pacific Northwest Statistical Downscaling

**Abstract**


The goal of this project was to (1) develop consistent historical and future downscaled climate and ...



### California Basin Characterization Model Downscaled Climate and Hydrology

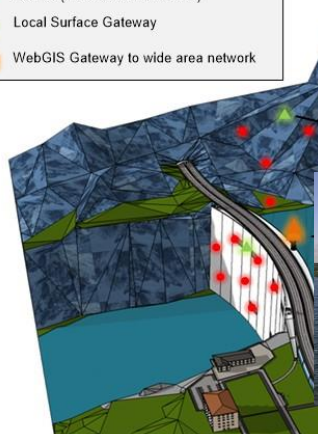
**Abstract**

The California Basin Characterization Model (CA-BCM 2014) dataset provides historical and projected ...



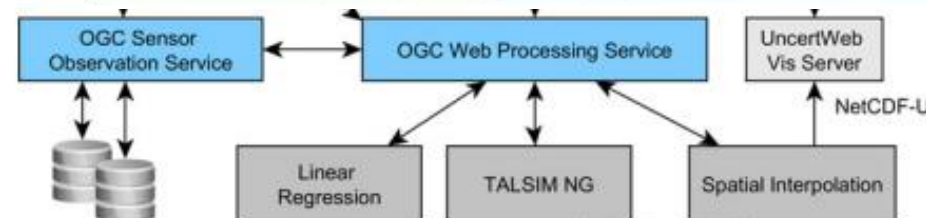
# EXAMPLE APPLICATION: TAMIS

- Legend
- Sensor (invasive/non-invasive)
  - ▲ Local Surface Gateway
  - ▲ WebGIS Gateway to wide area network



Timeseries modeling /  
Geoprocessing /  
Alerting

Open standards,  
Modelling Interface

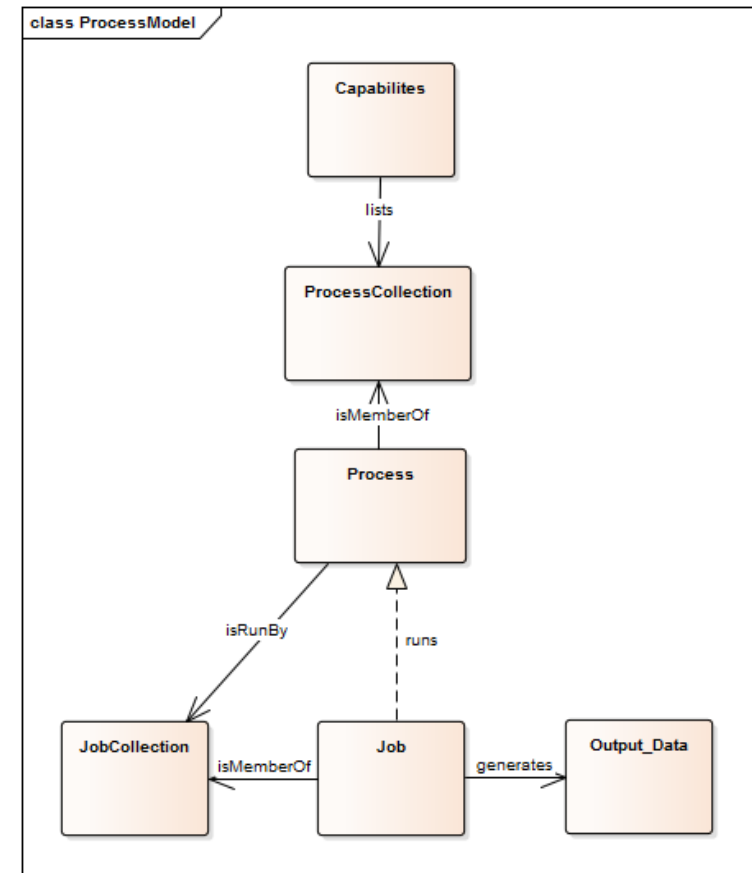


# **CURRENT WPS TOPICS**

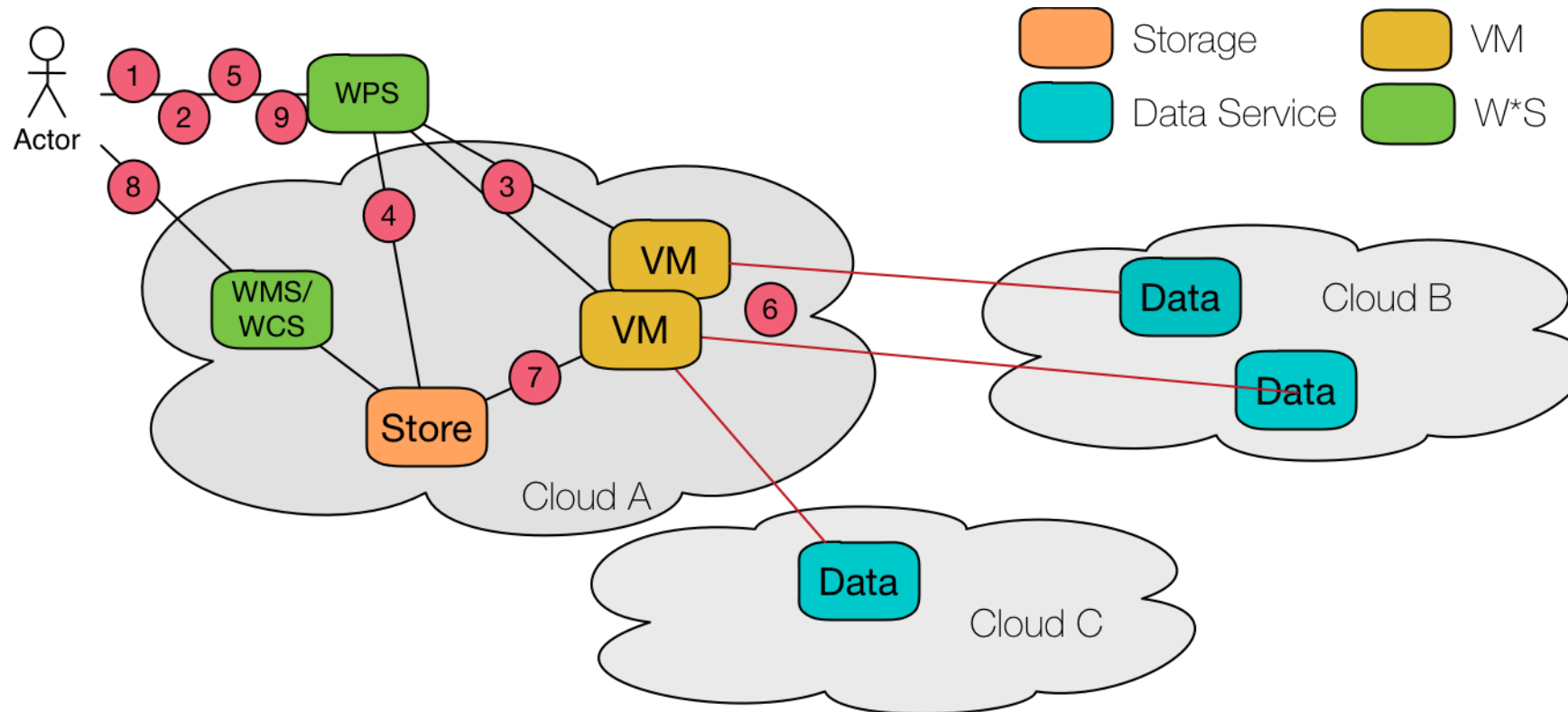
# REST API

- Currently in the process of standardization
- Based on a proposal in the OGC Testbed 12 - REST Architecture Engineering Report (OGC 16-035)
- Implementation as proxy for „normal“ WPSs available

REST proxy implementation: <https://github.com/52North/wps-proxy>  
Engineering Report: <http://docs.opengeospatial.org/per/16-035.html>

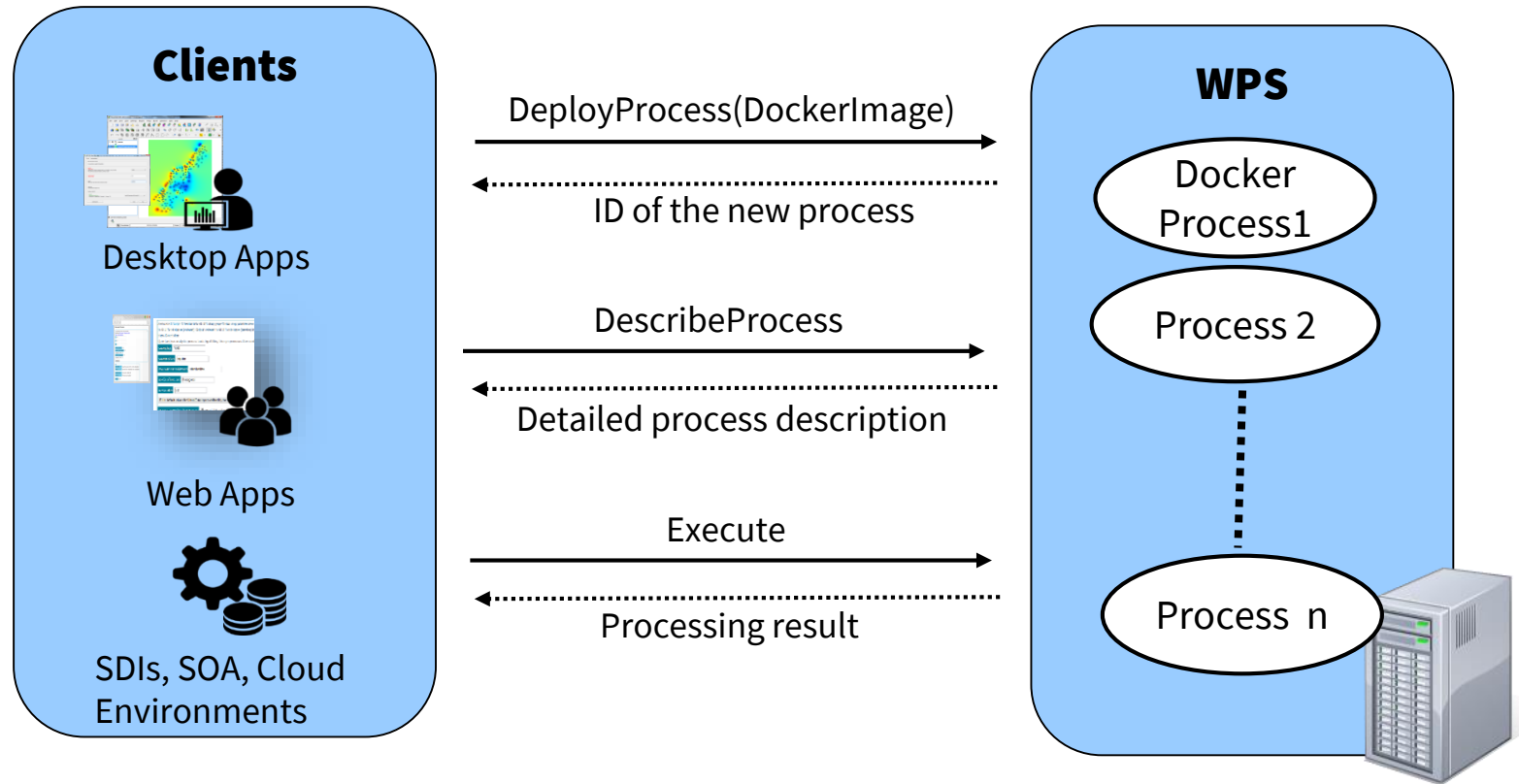


# INTERFACE FOR CLOUD-BASED GEPROCESSING



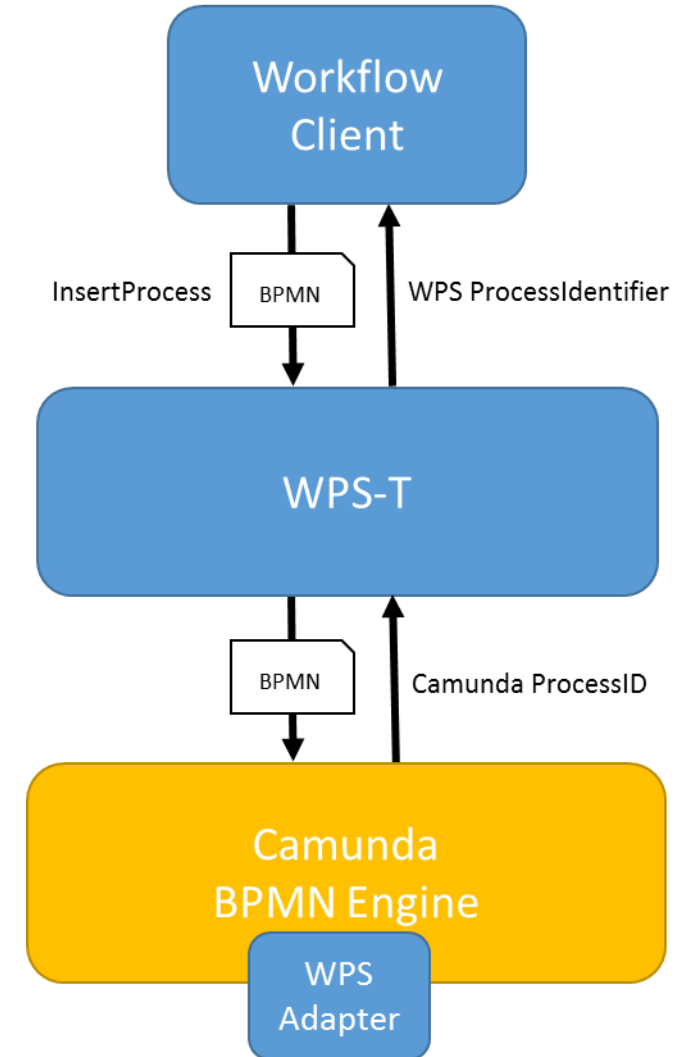
Quelle: <http://www.opengeospatial.org/node/2526#Cloud>

# HOSTED PROCESSING



# INTEROPERABLE GEOPROCESSING WORKFLOWS

- BPMN/Description Languages
- Provenance/Metadata (Discovery)
- Workflow Validation
- Uncertainty Propagation
- Security
- Granularity (Microservices)





# **SERVER IMPLEMENTATIONS: HOW CAN I PROVIDE GEOPROCESSING FUNCTIONALITY IN A WPS?**



# SERVER IMPLEMENTATIONS

- OGC lists 54 implementations
- Comercial:
  - ESRI, FME, Intergraph, Envitia, ERDAS, ...
- Open Source:
  - 52°North, pyWPS, ZOO, Geoserver, ...

# 52°NORTH WPS

- Version 3.3.x → Beta-Release of version 4.0.0 available
- Supports (all) features and operations of the WPS specification Version 1.0.0 and 2.0
- Support of:
  - GRASS 7, Sextante, R, Java/Python
- Java Process/R Script upload via Web UI
- Parser/Generators for common data formats:
  - SHP-Files, GML, GeoJSON, GeoTIFF, NetCDF, ...

# 52°NORTH WPS ADMIN APP

52 North

Home

Community

Resources ▾

Username...

Password...

☐ Remember Me

Login

CONFIGURATIONS

Server

Repositories

Generators

Parsers

SETTINGS

Users

Log

Service Identification

Service Provider

TESTING

Test Client

Backup & Restore

## 52° North WPS

### Documentation

- To learn more about the specification visit the [OGC website](#).
- To learn more about this implementation visit the [52°North Geoprocessing Community website](#).

### Administration

Please login to access administration pages.

### Examples

#### Requests

- [GetCapabilities request using HTTP GET](#)

#### Test Client

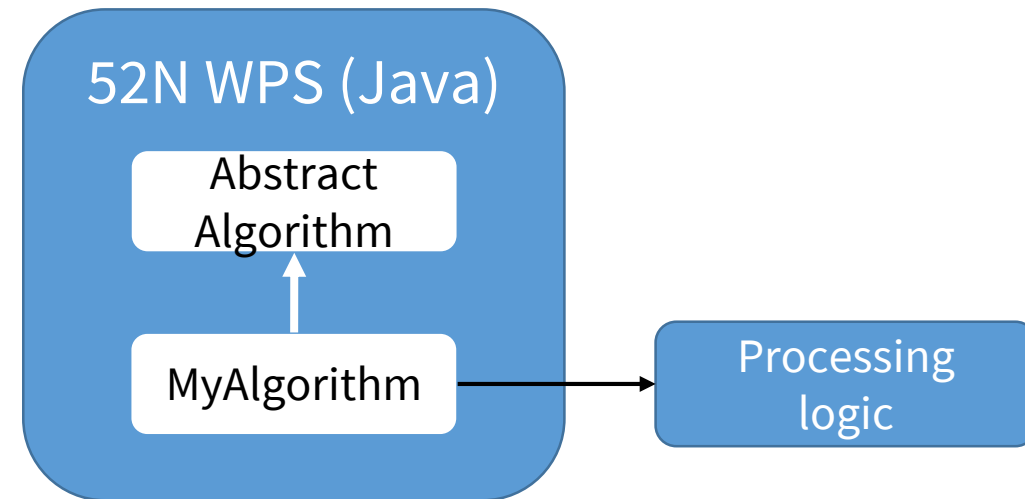
Open the test client of this WPS instance here: [52°North WPS test client](#).

# 52°NORTH WPS DEPLOYMENT PATTERNS

- Extending an existing WPS
- Deployment of annotated scripts (currently R)
- Custom Binaries with configuration files
- Databases
- Coupling with Geoserver

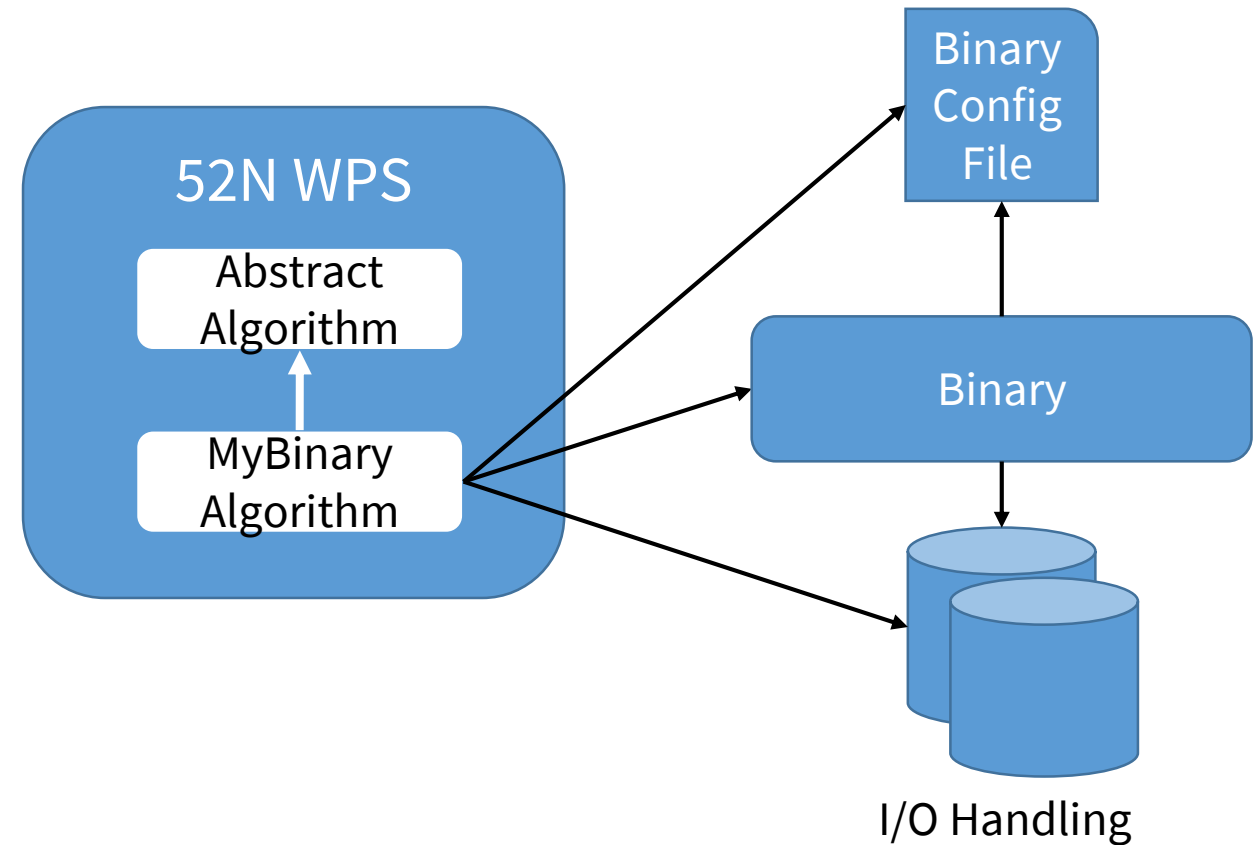
# WPS EXTENSION

- Direct extension of a existing WPS Server implementation
  - Reuse of I/O handlers
- Example: Implementation of an algorithm in Java as direct extension of the 52N WPS



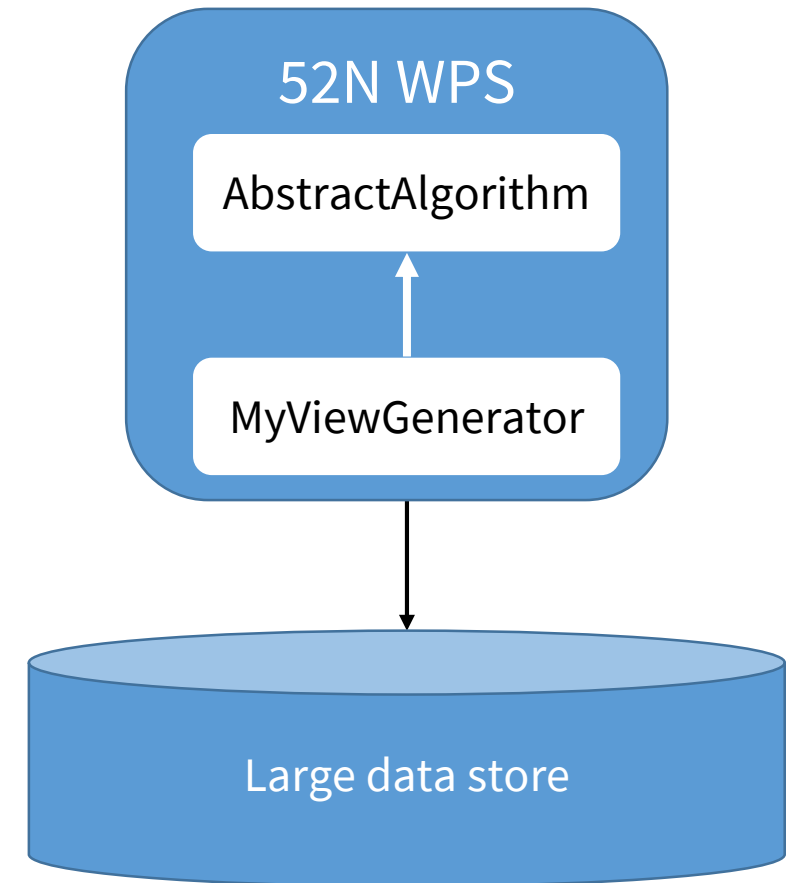
# EXECUTION OF CUSTOM BINARIES

- Execution of a program in binary code
- Configuration via config files
  - Parameters for controlling the process
  - Inputs/Outputs are stored locally and are referenced using config parameters



# WPS AS RICH DATA INTERFACE

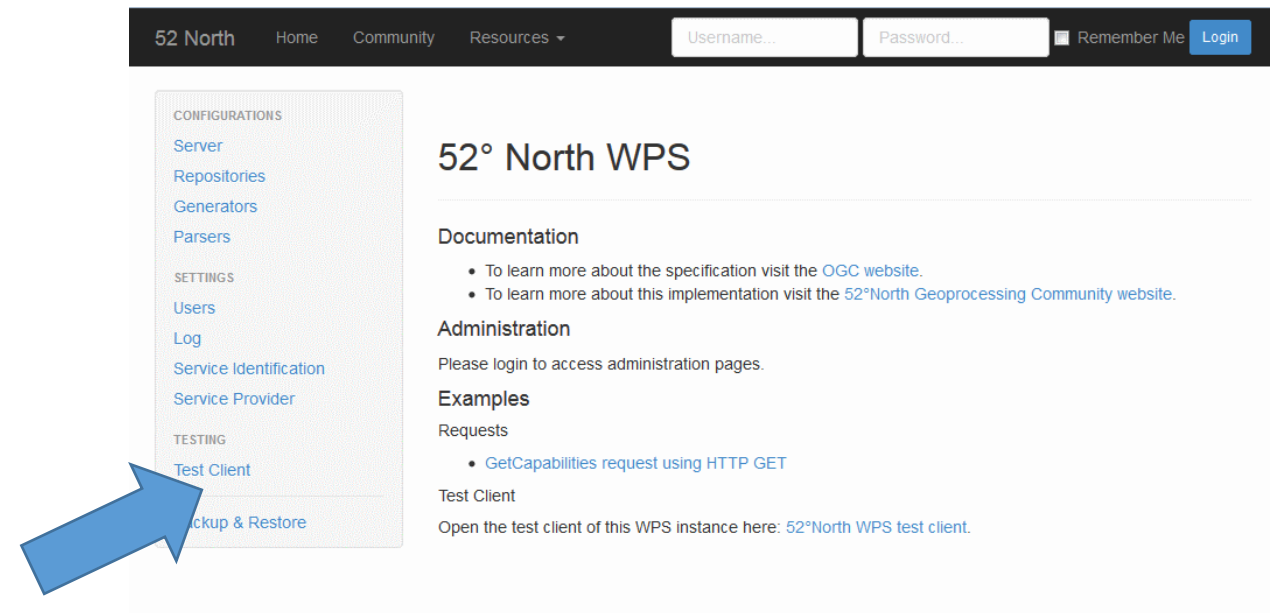
- WPS as interface for a large data store
  - Dynamic views can be generated using input parameters
- Inputs don't contain data for processing



# DEPLOYMENT OF R SCRIPTS

- Annotation in R scripts define inputs/outputs
- Upload of annotated R scripts allows deployment as WPS processes

```
# wps.des: id = Random, title = Random number generator,  
# abstract = Generates random numbers for uniform distribution;  
# wps.in: min, double, Minimum, All outcomes are larger than min, value = 0;  
# wps.in: max, double, Maximum, All outcomes are smaller than max, value = 1;  
# wps.in: n, integer, amount of random numbers, value = 100;  
# random number:  
x = runif(n, min=min, max=max)  
output = "outputfilename"  
write.table(x, output)  
# wps.out: output, text, Random number list, Textfile containing n random numbers in one column;
```





# ARCGIS SERVER WPS EXTENSION

ArcGIS Enterprise Portal Server Data Stores Cloud

Search Sign In English

## ArcGIS Server

Home Get Started Publish Services Use GIS Services Administer

Get Started / Introducing ArcGIS Server

### Introducing ArcGIS Server

- What is ArcGIS Server?
- Components of ArcGIS Server
- What's included with ArcGIS Server
- Compatibility of ArcGIS Server 10.6.1 with earlier versions
- What's new in ArcGIS Server 10.6.1 and 10.6
- Steps to get ArcGIS Server up and running
- Upgrade ArcGIS Server
- Tutorials
- Server licensing roles

## What is ArcGIS Server?

ArcGIS 10.6 (Windows) | Other versions

ArcGIS Server is software that makes your geographic information available to others in your organization and optionally anyone with an Internet connection. This is accomplished through web services, which allow a powerful server computer to receive and process requests for information sent by other devices. ArcGIS Server opens your GIS to tablets, smartphones, laptops, desktop workstations, and any other devices that can connect to web services.

To get started with ArcGIS Server, you'll need to prepare your hardware, software, and data, then you can set up GIS web services. Finally, you can use various types of applications to consume your services.

### Preparing hardware, software, and data

The hardware you use for your server is typically more powerful than your other desktop computers. ArcGIS Server requires a machine capable of running a 64-bit operating system. The ArcGIS Server architecture is scalable, meaning you can add multiple machines if extra processing power is needed.

#### In this topic

- Preparing hardware, software, and data
- Publishing GIS web services
- Using GIS web services
- Maintaining your server
- Summary

<http://enterprise.arcgis.com/de/server/latest/get-started/windows/tutorial-publishing-a-wps-service.htm>

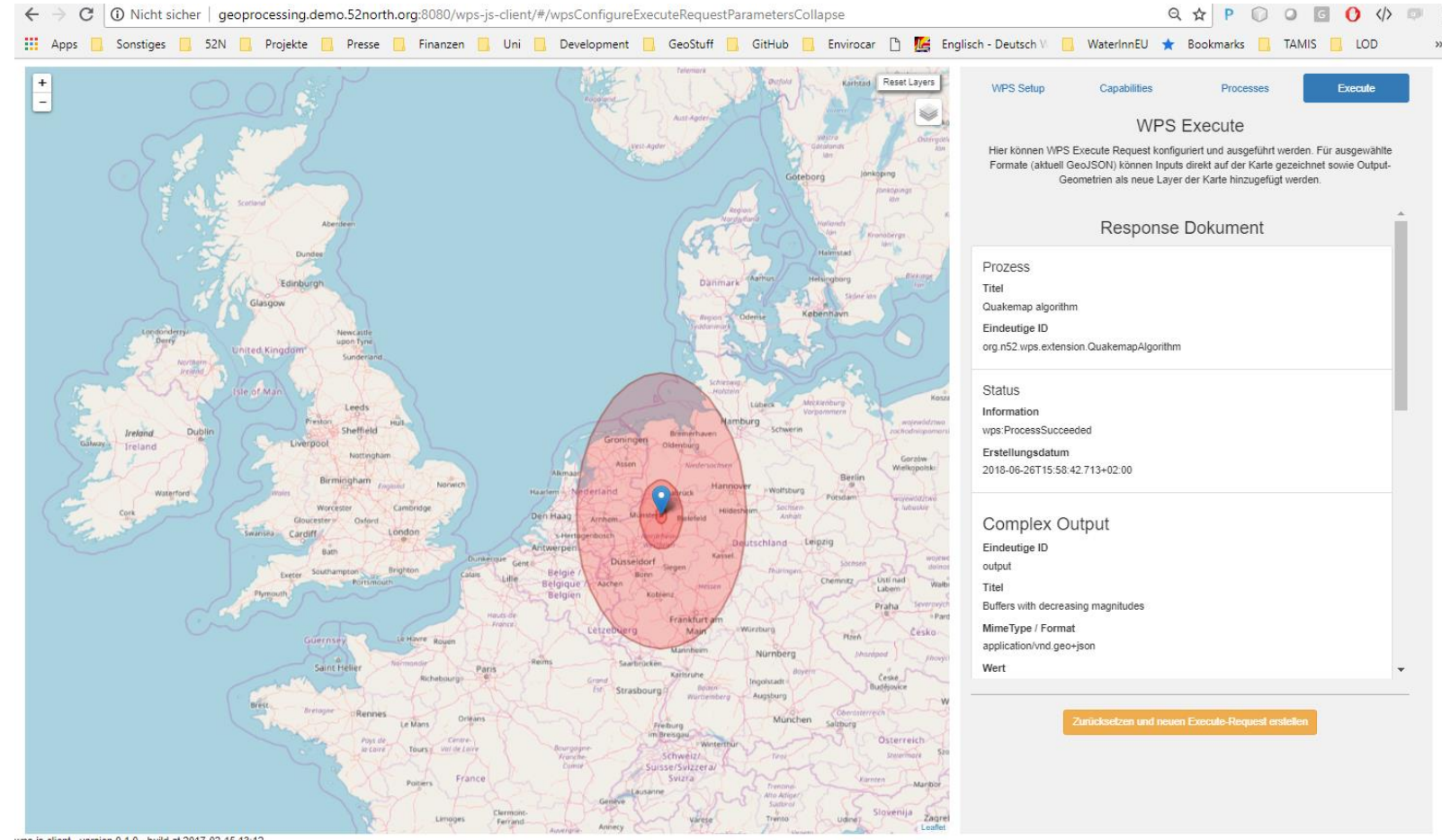
**CLIENTS:**  
**HOW CAN I USE GEOPROCESSING**  
**FUNCTIONALITY OF A WPS?**

# CLIENT IMPLEMENTATIONS

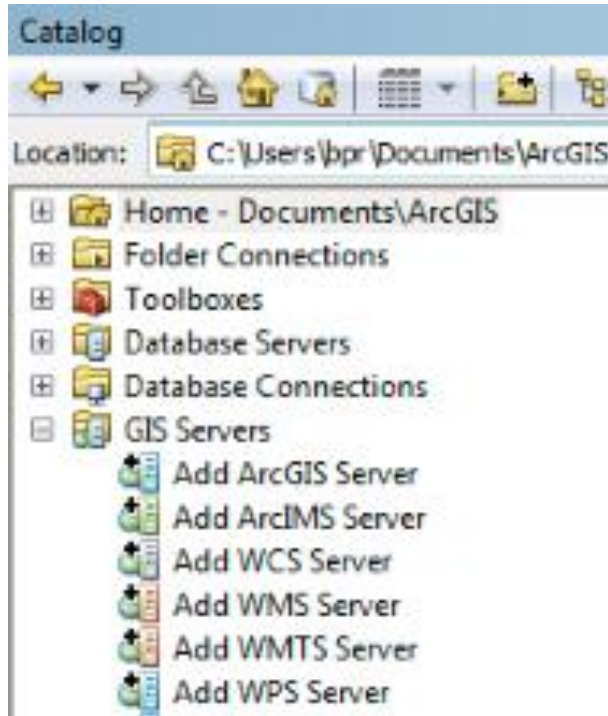
- 52°North ArcGIS WPS Client
  - ArcGIS Extension in collaboration with ESRI Inc.
  - Available as Open Source
- 52°North WPS-JS
  - JavaScript library for the creation of Web clients
- Quantum-GIS WPS – Plugin
- Custom clients for specific applications, e.g. USGS

# 52°NORTH WPS-JS-CLIENT

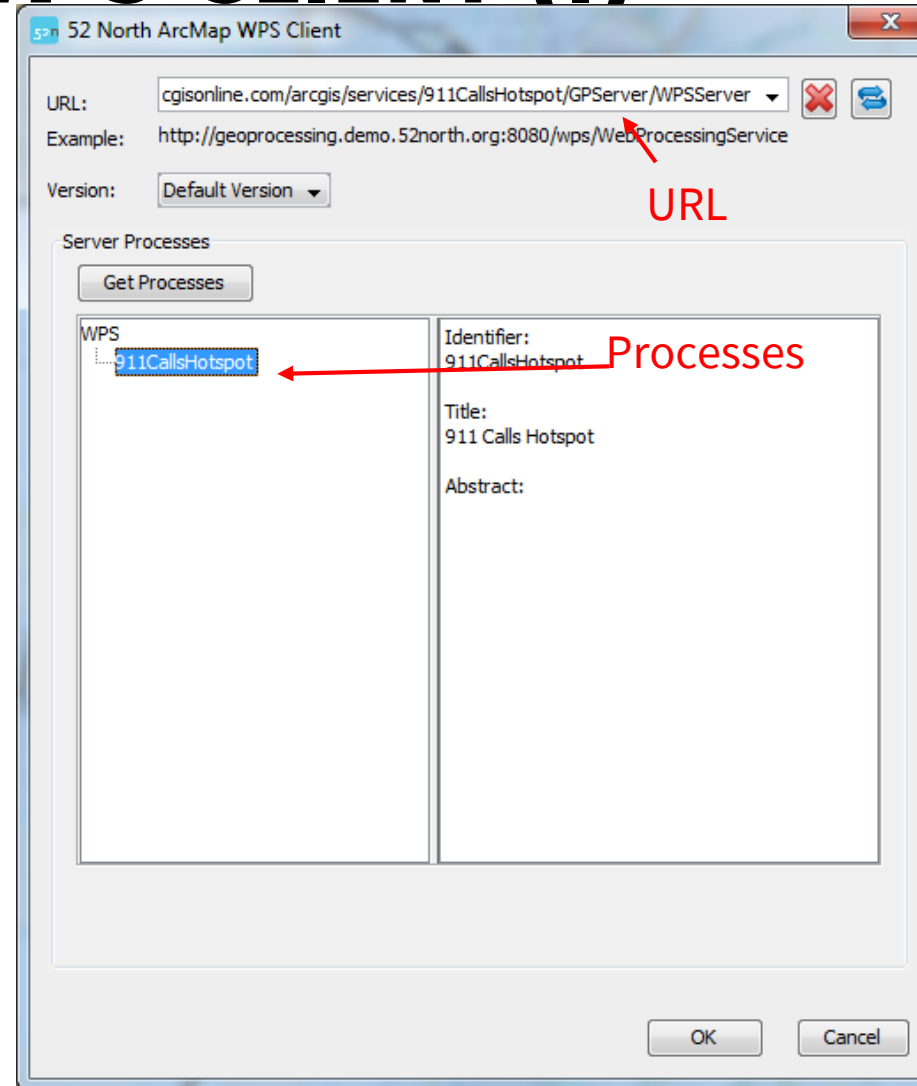
- wps-js:
  - JavaScript client library
- Wps-js-client:
  - JavaScript client
  - Based on the Angular Framework



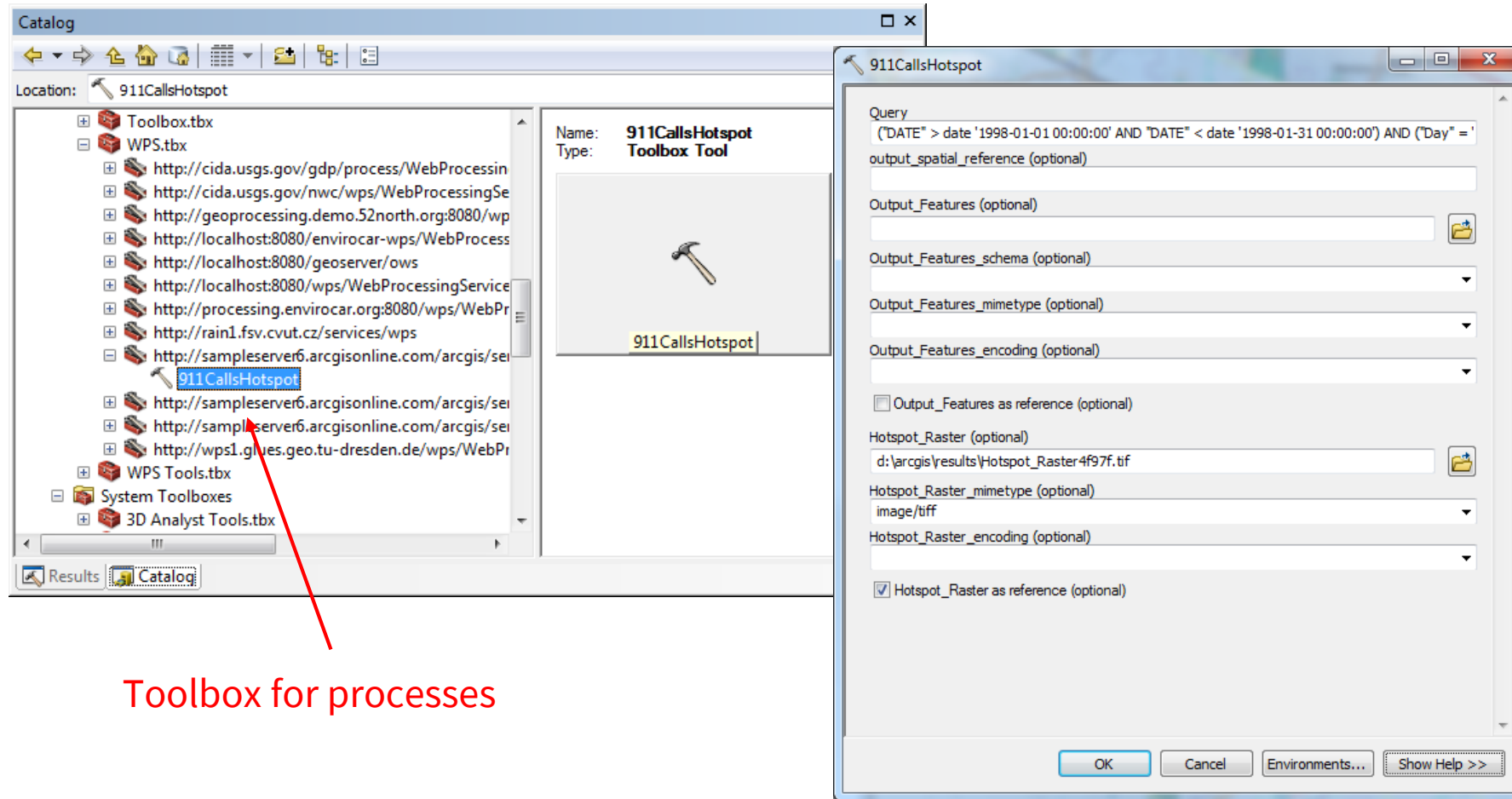
# 52°NORTH ArcGIS WPS CLIENT (I)



Adding of a WPS



# 52°NORTH ArcGIS WPS CLIENT (II)

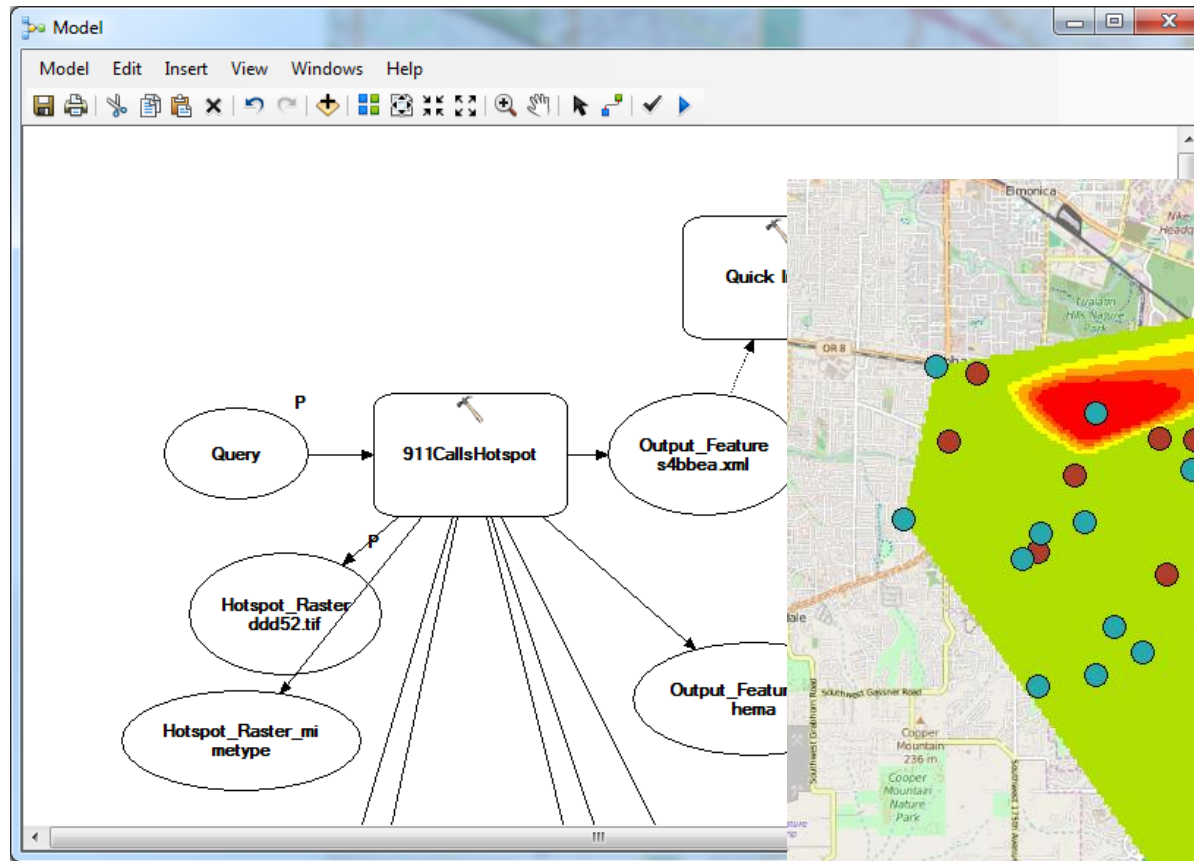


Toolbox for processes

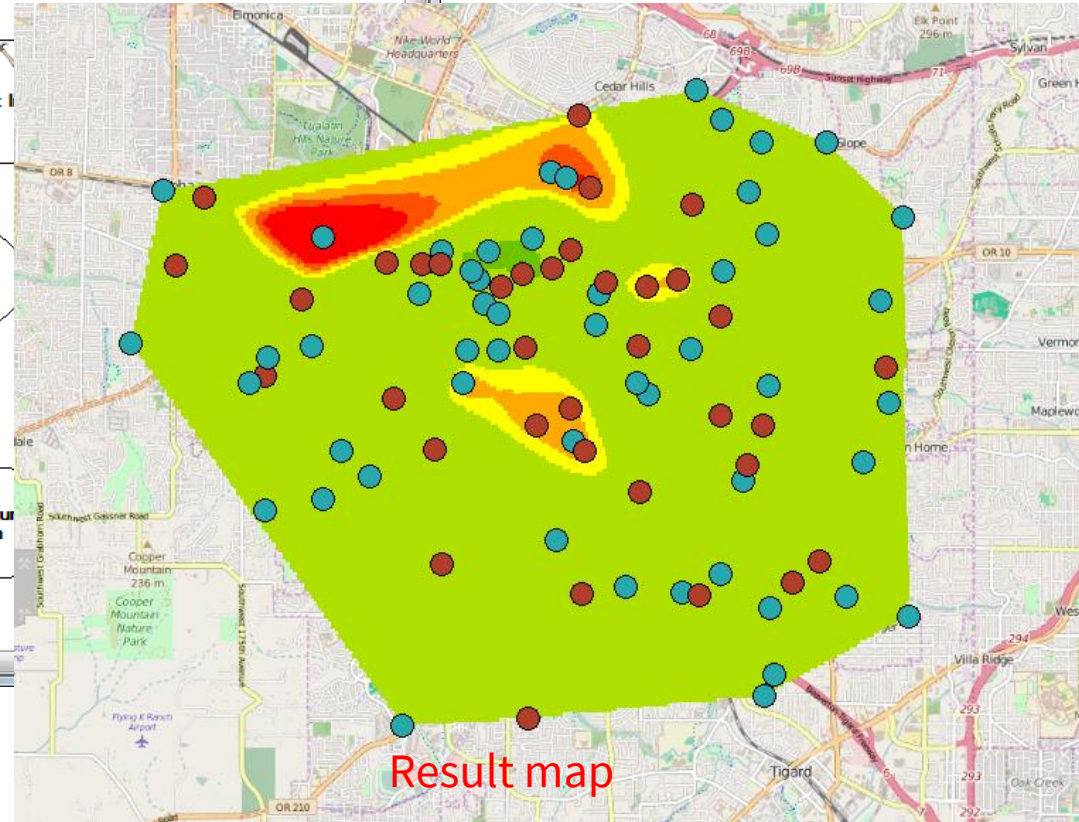
Input form for the execution



# 52°NORTH ArcGIS WPS CLIENT (III)



Integration in the Model Builder



Result map

# SUMMARY

- OGC WPS as standardized service for the access to geoprocessing functionality in the Web
  - Complementary to OGC – data services
  - Asynchronous execution
- REST Binding currently in standardization process:
  - Leaner and easier to use than the SOAP Binding
- 52N REST Proxy already usable with existing WPS servers
  - Currently restricted, e.g. the JSON Encoding is currently 52n-specific (not yet standardized)



# THANKS!



Benjamin Pross

WPS Expert

[b.pross@52north.org](mailto:b.pross@52north.org)



Christoph Stasch

Geodata processing

[c.stasch@52north.org](mailto:c.stasch@52north.org)

52N WPS Website: <https://52north.org/software/software-projects/wps/>

52N WPS Mailinglist: <https://list.52north.org/mailman/listinfo/geoprocessing>

# SHORT DEMO

# **ANNEX: WPS REST EXTENSION**

# MOTIVATION FOR A REST EXTENSION

- WPS 2.0 and other OGC specifications normally define a SOAP Binding as necessary interface
  - Exceptions : OGC Web Map Tiling Server and Sensor Things API
- REST APIs with JSON Encodings leaner for implementations of Web clients and easier to use
  - Focus on ressources (not on operations)
  - Usa of standard HTTP operations for CRUD of resources
- In OGC Testbed 12 REST APIs for different OGC services were tested and described in the Testbed 12 REST Architecture Engineering Report

## Testbed-12 REST Architecture Engineering Report

Publication Date: 2017-05-12

Approval Date: 2016-12-07

Posted Date: 2016-10-28

Reference number of this document: OGC 16-035

Reference URL for this document: <http://www.opengis.net/doc/PER/t12-A005-1>

Category: Public Engineering Report

Editors: Christoph Stasch, Simon Jirka

Title: Testbed-12 REST Architecture Engineering Report

### OGC Engineering Report

#### COPYRIGHT

Copyright © 2017 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

#### WARNING

This document is an OGC Public Engineering Report created as a deliverable of an initiative from the OGC Innovation Program (formerly OGC Interoperability Program). It is not an OGC standard and not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

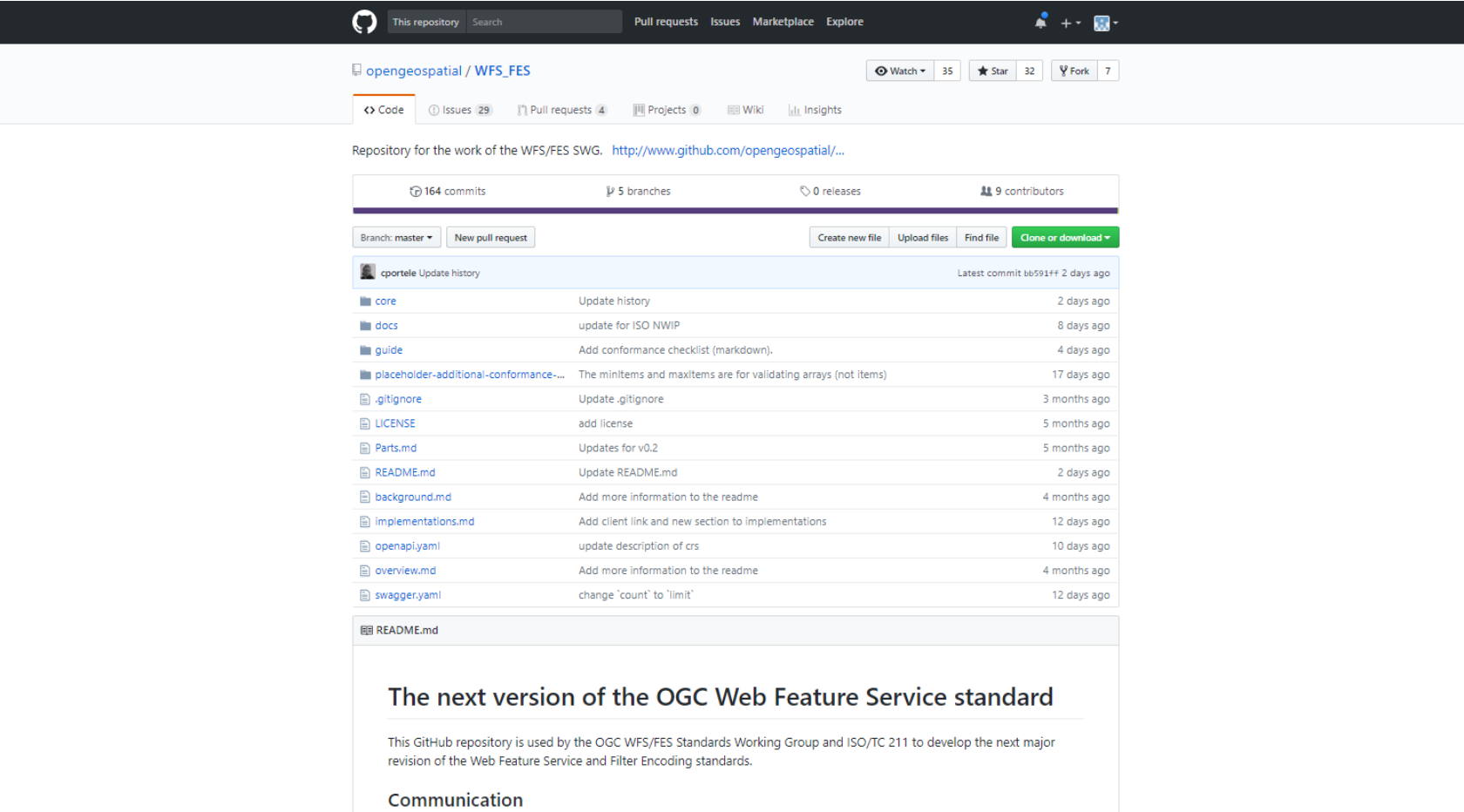
#### LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property; and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property, and to permit persons to whom the Intellectual Property

<http://docs.opengeospatial.org/per/16-035.html>

# EXAMPLE: WEB FEATURE SERVICE 3.0 (DRAFT)



Repository for the work of the WFS/FES SWG. <http://www.github.com/opengeospatial/...>

164 commits 5 branches 0 releases 9 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

Update history Latest commit bb591f7 2 days ago

core	Update history	2 days ago
docs	update for ISO NWIP	8 days ago
guide	Add conformance checklist (markdown).	4 days ago
placeholder-additional-conformance...	The minitems and maxitems are for validating arrays (not items)	17 days ago
.gitignore	Update .gitignore	3 months ago
LICENSE	add license	5 months ago
Parts.md	Updates for v0.2	5 months ago
README.md	Update README.md	2 days ago
background.md	Add more information to the readme	4 months ago
implementations.md	Add client link and new section to implementations	12 days ago
openapi.yaml	update description of crs	10 days ago
overview.md	Add more information to the readme	4 months ago
swagger.yaml	change 'count' to 'limit'	12 days ago

README.md

## The next version of the OGC Web Feature Service standard

This GitHub repository is used by the OGC WFS/FES Standards Working Group and ISO/TC 211 to develop the next major revision of the Web Feature Service and Filter Encoding standards.

### Communication

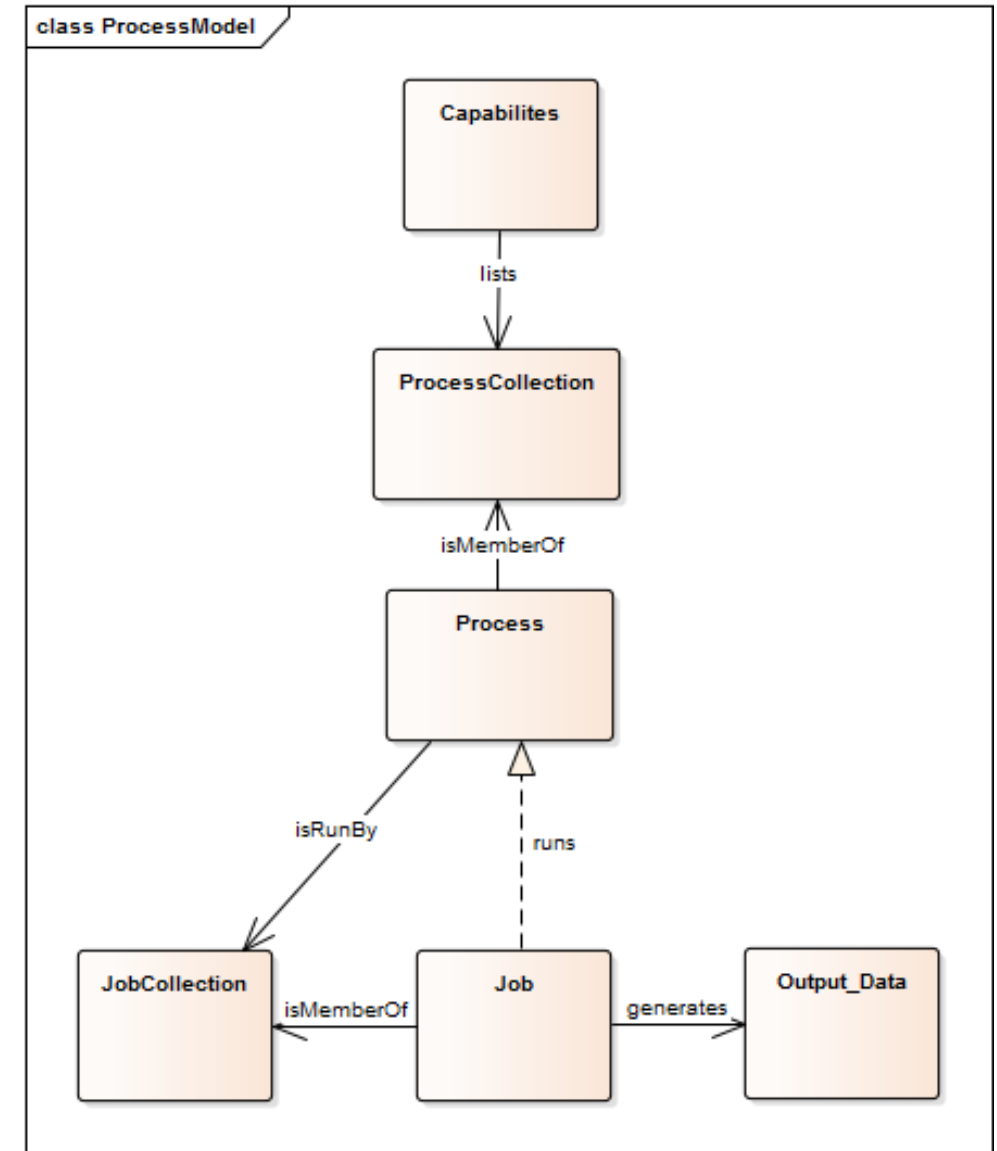
[https://github.com/opengeospatial/WFS\\_FES](https://github.com/opengeospatial/WFS_FES)

# EXISTING GEOPROCESSING REST APIs

- APIs for existing GIS software
  - ESRI's Geoservices REST API
  - FME Server
  - Hexagon Foundation API
  - ...
- APIs for particular geoprocessing applications
  - Geocoding (Bing Maps, Google, Yahoo, ...)
  - Routing (Google, HERE Maps, ...)
  - ...

# WPS RESSOURCE MODEL

- Based upon the general process model of WPS 2.0
- Hypermedia approach
  - Capabilities contains link to ProcessCollection
  - ProcessCollection contains link to single Process -ressources
  - ...



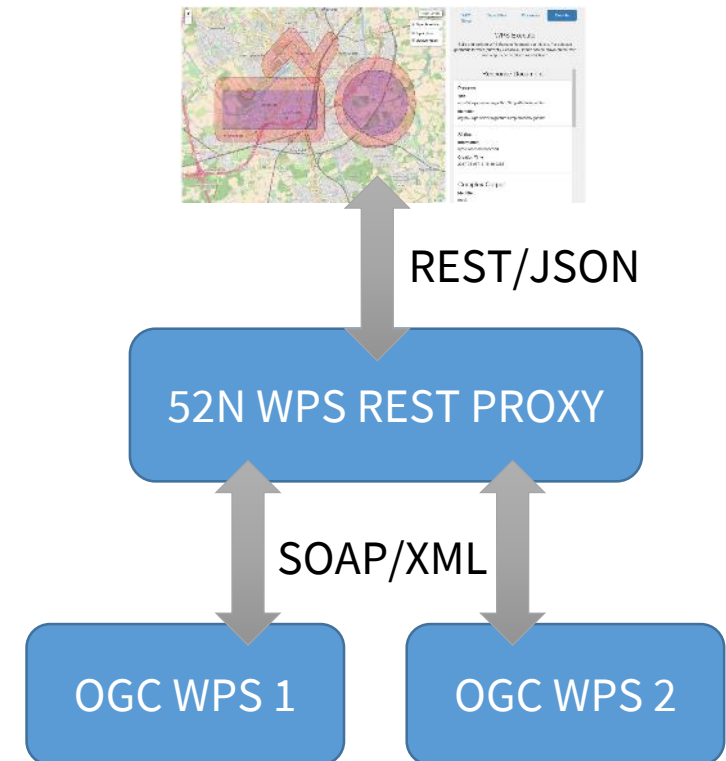
# ENDPOINTS FOR RESOURCE ACCESS

Ressource	Beschreibung	HTTP Operation	Endpunkt	Message Body
Capabilities	Request of the service description	HTTP GET	{WPSRestBasisURL}	-
ProcessCollection	Request of the list of processes	HTTP GET	{WPSRestBasisURL}/processes	-
Process	Request of a single process description	HTTP GET	{WPSRestBasisURL}/processes/{processID}	-
JobCollection	Request of the list of jobs (executions) of a process	HTTP GET	{WPSRestBasisURL}/processes/{processID}/jobs	-
Job	Execution of a process/creation of a new job	HTTP POST	{WPSRestBasisURL}/processes/{processID}/jobs	Execute Request in JSON
Job	Request of the status of a job	HTTP GET	{WPSRestBasisURL}/processes/{processID}/jobs/{jobID}	-
Outputs	Request of the results of a job	HTTP GET	{WPSRestBasisURL}/processes/{processID}/jobs/{jobID}/results	-



# 52N WPS-REST-PROXY

- Implements REST extension as proxy for existing WPS instances
  - Advantage: Can be used together with existing WPS 2.0 servers
  - Disadvantage: Increased communication effort
- GitHub Repo:
  - <https://github.com/52North/wps-proxy>
- Example instance for testing:
  - <http://geoprocessing.demo.52north.org:8080/wps-proxy>
- Tutorial:
  - <https://wiki.52north.org/Geoprocessing/WPSRESTProxy>



# REQUEST CAPABILITIES

HTTP GET <baseURL>

- No predefined pattern for endpoint-URL
- Returns Capabilities document containing the list of available processes

```
1  {
2    "Capabilities": {
3      "ServiceIdentification": {
4        "Title": "52°North WPS 4.0.0-SNAPSHOT",
5        "Abstract": "Service based on the 52°North implementation of WPS 1.0.0",
6        "ServiceType": "WPS",
7        "ServiceTypeVersion": ["1.0.0",
8          "2.0.0"],
9        "Fees": "NONE",
10       "AccessConstraints": "NONE"
11     },
12     "ServiceProvider": {
13       "ProviderName": "52North",
14       ...
15     }
16   },
17   "Contents": {
18     "ProcessSummaries": [{
19       "identifier": "testbed12.fo.DouglasPeuckerAlgorithm",
20       "title": "testbed12.fo.DouglasPeuckerAlgorithm",
21       "_processVersion": "1.0.0",
22       "_jobControlOptions": "sync-execute",
23       "_outputTransmission": "value",
24       "url": "http://geoprocessing.demo.52north.org:8080/wps-proxy/processes/
25         org.n52.wps.server.algorithm.DouglasPeuckerAlgorithm"
26     }, ...
27   ]
28 },
29 "_service": "WPS",
30 "_version": "2.0.0"
31 }
32 }
```

General service information

Process list

<http://geoprocessing.demo.52north.org:8080/wps-proxy>

# REQUEST PROCESS LIST

HTTP GET <baseURL>/processes

- Returns a list of short process summaries containing links to detailed process descriptions

```
1  {
2    "ProcessSummaries": [
3      {
4        "identifier": "testbed12.fo.DouglasPeuckerAlgorithm",
5        "title": "testbed12.fo.DouglasPeuckerAlgorithm",
6        "_processVersion": "1.0.0",
7        "_jobControlOptions": "sync-execute",
8        "_outputTransmission": "value",
9        "url": "http://geoprocessing.demo.52north.org:8080/wps-proxy/processes/
10       org.n52.wps.server.algorithm.JTSConvexHullAlgorithm"
11      },
12    ]
13  }
14
```

Link to detailed process description

<http://geoprocessing.demo.52north.org:8080/wps-proxy/processes>

# REQUEST PROCESS DESCRIPTION

HTTP GET <baseURL>/processes/<process-id>

- Returns the detailed description of a process including input and output parameters

<http://geoprocessing.demo.52north.org:8080/wps-proxy/processes/org.n52.wps.server.algorithm.JTSConvexHullAlgorithm>

```

1  {
2    "ProcessOffering": {
3      "Process": {
4        "Title": "org.n52.wps.server.algorithm.JTSConvexHullAlgorithm",
5        "Identifier": "org.n52.wps.server.algorithm.JTSConvexHullAlgorithm",
6        "Input": [
7          {
8            "Title": "data",
9            "Identifier": "data",
10           "ComplexData": {
11             "Format": [
12               {
13                 "_default": "true",
14                 "_mimeType": "application/vnd.geo+json"
15               }, ...
16             ]
17           },
18           "_minOccurs": "1",
19           "_maxOccurs": "1"
20         ]
21       },
22       "Output": [
23         {
24           "Title": "result",
25           "Identifier": "result",
26           "ComplexData": {
27             "Format": [
28               {
29                 "_default": "false",
30                 "_mimeType": "application/vnd.geo+json"
31               }, ...
32             ]
33           }
34         ]
35       ],
36       "_processVersion": "1.1.0",
37       "_jobControlOptions": "sync-execute async-execute",
38       "_outputTransmission": "value-reference",
39       "execute-url": "http://geoprocessing.demo.52north.org:8080/wps-proxy/
40         processes/org.n52.wps.server.algorithm.JTSConvexHullAlgorithm/jobs"
41     }
42   }

```

Input parameter

Output parameter

Execution-URL

# EXECUTION OF A PROCESS (I)

## HTTP POST

<baseUrl>/processes/<process-id>/jobs

- Parameter:
  - Execute Request in JSON (see to the right)
- Optional URL parameter:  
sync-execute=true|false (default false)
- A new job resource (process execution) is created

```
1 {
2   "Execute": {
3     "Identifier": "org.n52.wps.server.algorithm.JTSConvexHullAlgorithm",
4     "Input": [
5       {
6         "ComplexData": {
7           "_mimeType": "application/wkt",
8           "_text": "POLYGON((847666.55940505 6793166.084248,
9             ...,
10            847666.55940505 6793166.084248))"
11         },
12         "_id": "data"
13       }
14     ],
15     "output": [{
16       "_mimeType": "application/wkt",
17       "_id": "result",
18       "_transmission": "value"
19     }],
20     "_service": "WPS",
21     "_version": "2.0.0"
22   }
23 }
24
```

Process-ID

Input data

Desired outputs

<http://geoprocessing.demo.52north.org:8080/wps-proxy/processes/org.n52.wps.server.algorithm.JTSConvexHullAlgorithm/jobs>

# EXECUTION OF A PROCESS (II)

- *Asynchronous* execution (Examples to the right):

- HTTP 201 with link to job resource
- After execution is finished link to results

```
1 {  
2   "StatusInfo": {  
3     "JobID": "c731d14b-1de6-499c-9317-20224e056012",  
4     "Status": "Running",  
5     "Progress": 0  
6   }  
7 }
```

Running job

- Synchronous execution (Example on next slide):

- JSON result document (success|failure)

```
1 {  
2   "StatusInfo": {  
3     "JobID": "c731d14b-1de6-499c-9317-20224e056012",  
4     "Status": "Succeeded",  
5     "Output": "http://geoprocessing.demo.52north.org:8080/wps-proxy/processes/  
6               org.n52.wps.server.algorithm.JTSConvexHullAlgorithm/jobs/  
7               c731d14b-1de6-499c-9317-20224e056012/outputs"  
8   }  
9 }  
10
```

Finished job with link to results

# REQUEST PROCESSING RESULTS

## HTTP GET:

<baseUrl>/processes/<process-id>/  
jobs/<job-id>/outputs

- Returns JSON result document

```
1  {
2    "Result": {
3      "JobID": "2b21bc7a-18d3-4f6e-8046-680c4c69245d",
4      "Output": [
5        {
6          "ID": "result",
7          "ComplexData": {
8            "_mimeType": "application/wkt",
9            "_text": "POLYGON ((849873.67859648 6792439.9324794,
10             ...,
11             849873.67859648 6792439.9324794))"
12          }
13        }
14      ]
15    }
16  }
17
```

# WPS TESTCLIENT

Testclient verfügbar unter: [http://geoprocessing.demo.52north.org:8080/wps/test\\_client](http://geoprocessing.demo.52north.org:8080/wps/test_client)

The screenshot shows a web browser window with the URL `geoprocessing.demo.52north.org:8080/wps/test_client`. The browser's address bar and tabs are visible at the top. On the left side, there is a navigation menu with the following items: Server, Repositories, Generators, Parsers, Databases, SETTINGS, Users, Log, Service Identification, Service Provider, TESTING, Test Client (highlighted in blue), and Backup & Restore. The main content area is titled "Test Client" and "WPS test client". It contains a "Service URL" input field with the value `http://geoprocessing.demo.52north.org:8080/wps/WebProcessingService` and a "Request" dropdown menu set to "SimpleBuffer.xml". Below these fields is a large text area displaying an XML request. The XML is a WPS Execute request for the SimpleBufferAlgorithm. Below the text area are "Send" and "Clear" buttons. The bottom of the page features a dark footer with the following links: Communities, Get Involved, Affiliations, and Cooperation partners. In the bottom left corner, there is a blue square logo with the text "52n".

Test Client  
WPS test client

Service URL: `http://geoprocessing.demo.52north.org:8080/wps/WebProcessingService`

Request: SimpleBuffer.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>org.n52.wps.server.algorithm.SimpleBufferAlgorithm</ows:Identifier>
  <wps:DataInputs>
    <ows:Input>
      <ows:Identifier>data</ows:Identifier>
      <ows:Reference schema="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"
xlink:href="http://geoprocessing.demo.52north.org:8080/geoserver/wfs?
SERVICE=WFS&VERSION=1.0.0&REQUEST=GetFeature&TYPENAME=topp:tasmania_roads&SRS=EPSG:4326&OU
method="GET"/>
    </ows:Input>
    <ows:Input>
      <ows:Identifier>width</ows:Identifier>
      <ows:Data>

```

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ExecuteResponse xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ows="http://www.opengis.net/ows/1.1"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_response.xsd"
serviceInstance="http://geoprocessing.demo.52north.org:8080/wps/WebProcessingService?
REQUEST=GetCapabilities&SERVICE=WPS" xml:lang="en-US" service="WPS" version="1.0.0">
  <wps:Process wps:processVersion="1.1.0">
    <ows:Identifier>org.n52.wps.server.algorithm.SimpleBufferAlgorithm</ows:Identifier>
    <ows:Title>org.n52.wps.server.algorithm.SimpleBufferAlgorithm</ows:Title>
  </wps:Process>
  <wps:Status creationTime="2016-03-05T08:19:07.112Z">
    <wps:ProcessSucceeded>Process successful</wps:ProcessSucceeded>
  </wps:Status>
  <wps:ProcessOutputs>
    <ows:Output>
      <ows:Identifier>result</ows:Identifier>
      <ows:Title>result</ows:Title>
      <ows:Data>

```