

52° North WPS-G

A grid-enabled OGC Web Processing Service (WPS)

OGC-OGF Collaboration Workshop

The 22nd Open Grid Forum – OGF22

February 25-28, 2008

Bastian Baranski, Bastian.Baranski@uni-muenster.de

Institute for Geoinformatics at the University of Münster

Agenda



1. Gridification
2. 52° North WPS-G
3. Live Demo
4. Outlook

Definition

Adaptation of existing applications and services to grid environments and technologies.

Research Questions

General:

In what context can synergy effects between grid technologies and SDI emerge?

Special:

How to integrate OWS into grid infrastructure?

Level of Gridification

1. OWS use grid (only) as backend resource
2. OWS is a (full) service in the grid

Agenda



ifgi
Institut für Geoinformatik
Universität Münster

1. Gridification
- 2. 52° North WPS-G**
3. Live Demo
4. Outlook



Features (WPS)

- Full support of WPS version 0.4.0 and 1.0.0
- Java-based Open Source implementation
- Easy Maven deployment
- Pluggable framework for algorithms and XML data handling
- Full GML2 support for ComplexValues (i.e. FeatureCollections)
- Synchronous and asynchronous processing
- Support of raster processing (beta)
- Support of GRASS GIS (diploma thesis)
- Support of Workflows (beta)
- (...)

Features (WPS-G)

- Enables classic 52° North WPS with grid abilities
- Pluggable framework for different grid middlewares
- Full support of UNICORE middleware
- Distribute 52° North WPS algorithms in the grid

Planned Features (WPS-G)

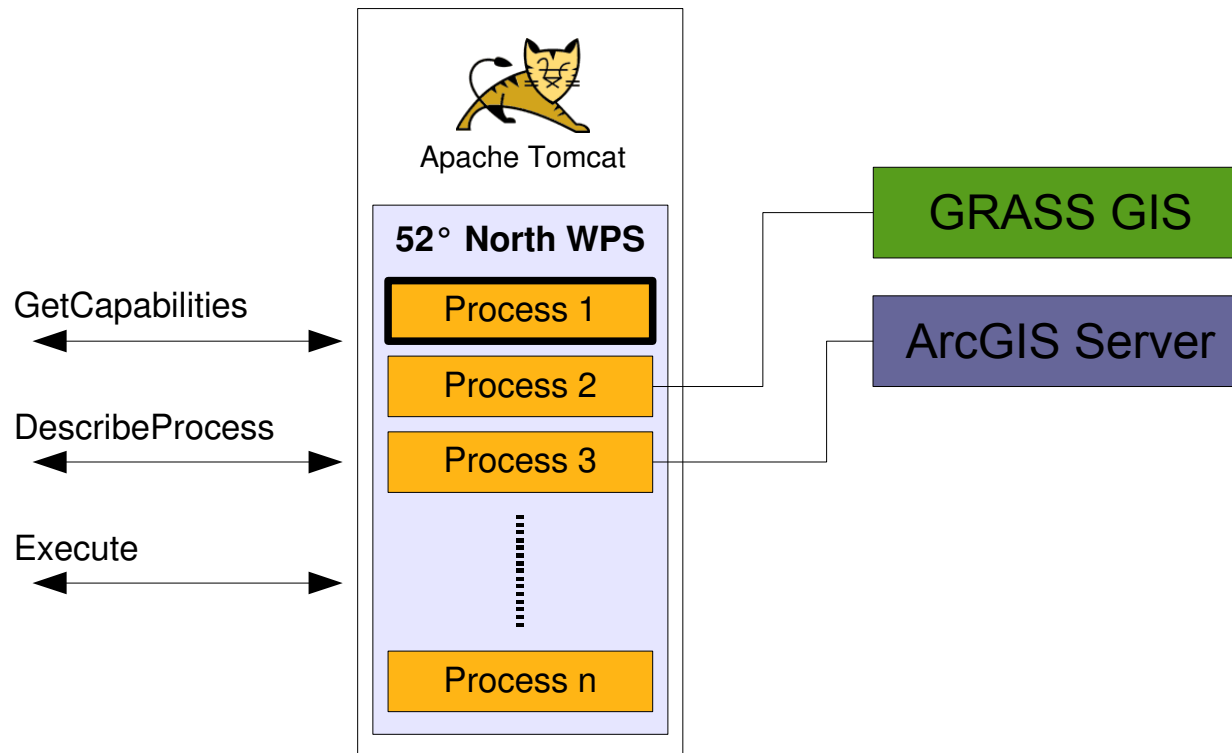
- Support of Desktop Grid technologies (e.g. BOINC, GridGain) (diploma thesis)
- Distribute GRASS processes in the grid (diploma thesis)

Download

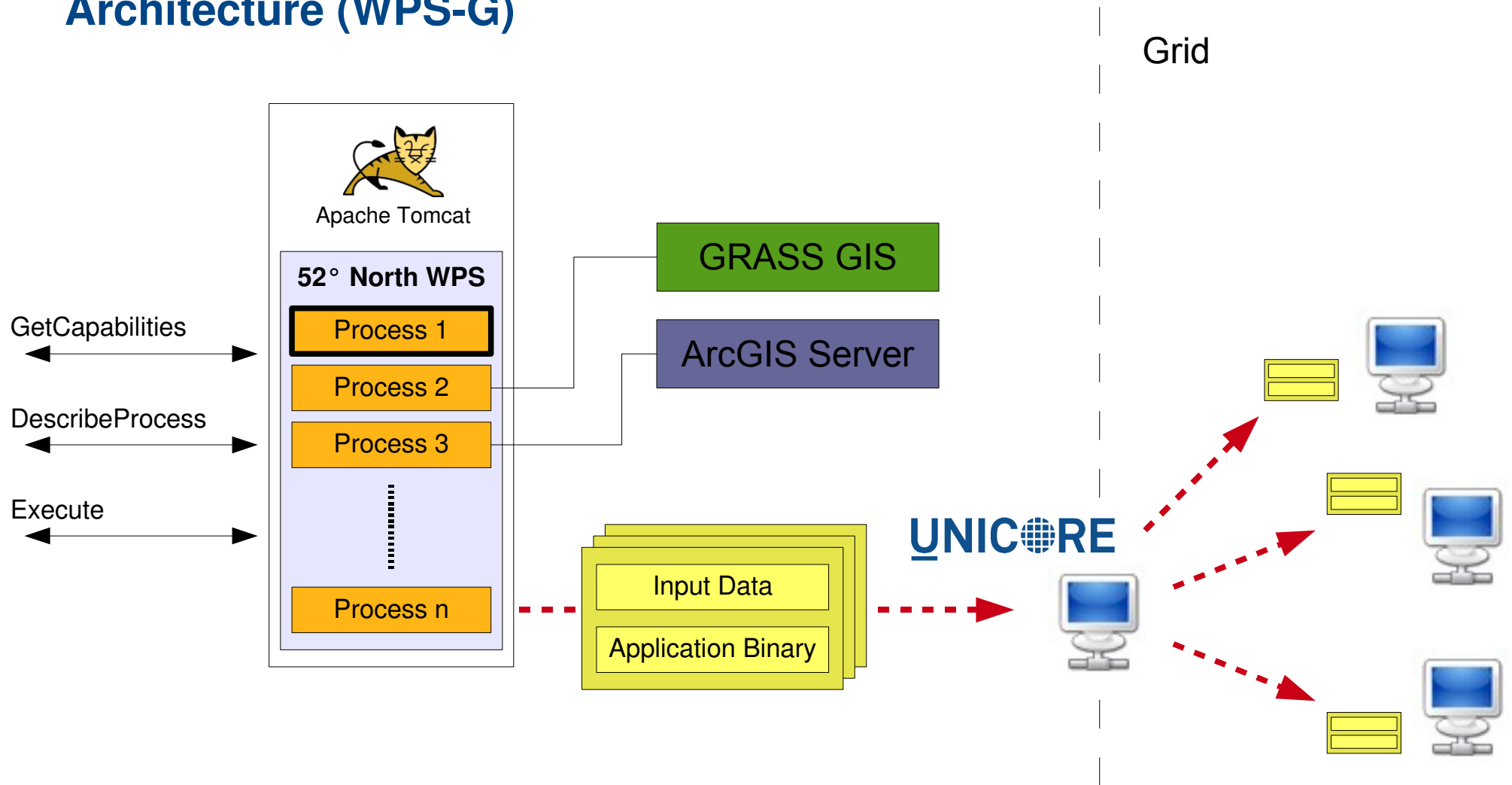
Server:

<http://core-52n.cvs.sourceforge.net/core-52n/WPS/?pathrev=WPS-G>

Architecture (WPS)



Architecture (WPS-G)



Algorithm Development

- Implement IGridAlgorithm Interface

```
package org.n52.wps.grid;
import java.util.List;
public interface IGridAlgorithm
{
    /* run algorithm */
    ProcessOutput run(ProcessInput pInput);

    /* split input data into n chunks */
    List<ProcessInput> split(ProcessInput pInput, int pCount);

    /* concatenate the results of distributed processes */
    ProcessOutput concatenate(List<ProcessOutput> pOutput);
}
```

Algorithm Development

- Create Process Description File

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessDescriptions version="1.0.0">
  <ProcessDescription wps:processVersion="2" statusSupported="true" storeSupported="true">
    <ows:Identifier>org.n52.wps.grid.algorithm.SimpleBufferAlgorithm</ows:Identifier>
    <ows:Title>Simple Buffer</ows:Title>
    <ows:Abstract>Create a buffer around a single polygon.</ows:Abstract>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>polygon</ows:Identifier>
        <ows:Title>Polygon to be buffered</ows:Title>
        <ows:Abstract>The Geometries to buffer</ows:Abstract>
        <ComplexData>
          <Default>
            <Format>
```

Algorithm Development

- Create Application Description File

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
  <app_name>org.n52.wps.grid.algorithm.SimpleBuffer</app_name>
  <version_num>1.0</version_num>
  <file_ref>
    <file_name>52n-wps-grid-1.0.0.jar</file_name>
  </file_ref>
  <file_ref>
    <file_name>gmlpacket-2.0-0.4.jar</file_name>
  </file_ref>
  (...)
</application>
```

Agenda



1. Gridification
2. 52° North WPS-G
- 3. Live Demo**
4. Outlook

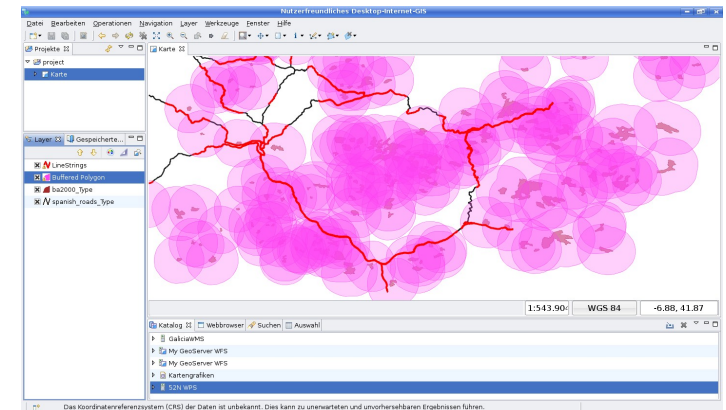
Demo Scenario

Generate a readable map which indicates the impact of recent fire threats to a transport infrastructure. *

* chosen location is the North-West of Spain

1. Import data of burnt areas (WFS)
2. Import data of spanish roads (WFS)
3. Create buffers around the burnt areas (WPS-G, local calculation)
4. Intersect buffers with road data (WPS-G, **distributed calculation in the grid** *)

* Using the public available Unicore 6 Testgrid at Research Centre Jülich. The Testgrid should only be used for testing purposes and currently there are 2 computation nodes available. For more information have a look at <http://www.unicore.eu/testgrid>.



Agenda



1. Gridification
2. 52° North WPS-G
3. Live Demo
- 4. Outlook**

52° North WPS-G

Outcomes

- Stable running proof-of-concept implementation of grid-enabled WPS
- Connection to (other) grid-middlewares could easily be done
- Grid Computing is a good choice to achieve high performance in geoprocessing
- Grid Computing is a good choice for outsourcing geoprocessing tasks

Research Questions

- How to integrate OWS as plain grid service into grid infrastructure?
- How to integrate complex SDI service chains into grid infrastructure?
- Moving code meets its limits very early.

Service Level Agreement (SLA)

A **Service Level Agreement (SLA)** is that part of a service contract between customers and service providers where the level of service (e.g. different Quality of Service) is formally defined.

Motivation for SLA in SDI

- Emergency situations (availability, delay, data quality, etc.)
- Outsourcing of geoprocessing tasks (monitoring, billing, etc.)

Architecture vision

- Attach SLA capabilities to OGC Web Services
- Include SLA into the GeoDRM architecture workflow

Grid Computing Technologies can be responsible to ensure these requirements!

Thank you!