

The logo consists of the lowercase letters 'i3' in a white, sans-serif font, positioned on a solid blue rectangular background.

mainz

Institut für raumbezogene  
Informations- und Messtechnik  
Fachhochschule Mainz

A 3D point cloud visualization of a mountain peak, rendered in grayscale. The points are densely packed to form the mountain's silhouette and topography.

# Flexibles System zur geostatistischen Visualisierung

A 3D point cloud visualization of a crumpled piece of paper, rendered in grayscale. The points are densely packed to form the complex, irregular shape of the crumpled paper.

Nikolai Bock

- **Projekt 3P-GM**
- **Herausforderungen**
- **System**
  - **Architektur**
  - **Komponenten**
  - **WPS-Schnittstelle**
- **Aktueller Stand**
- **Fazit**

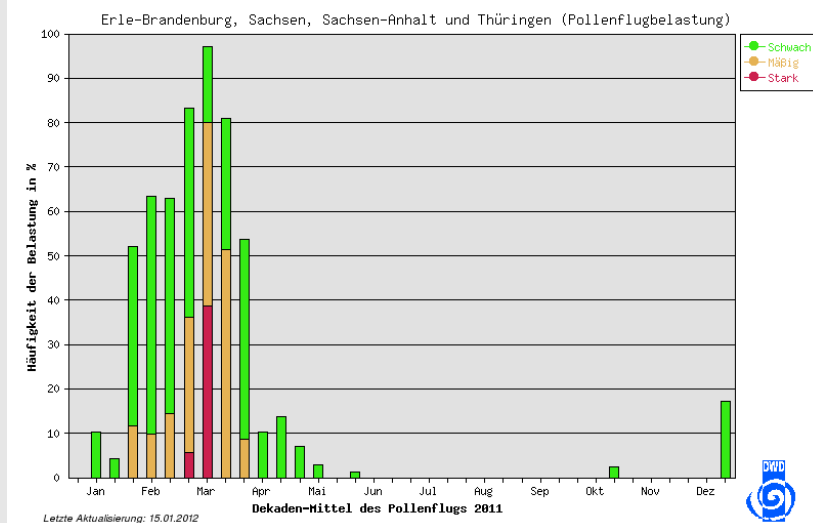


# 3P-GM

**Personalized Pollen Profiling  
and Geospatial Mapping**

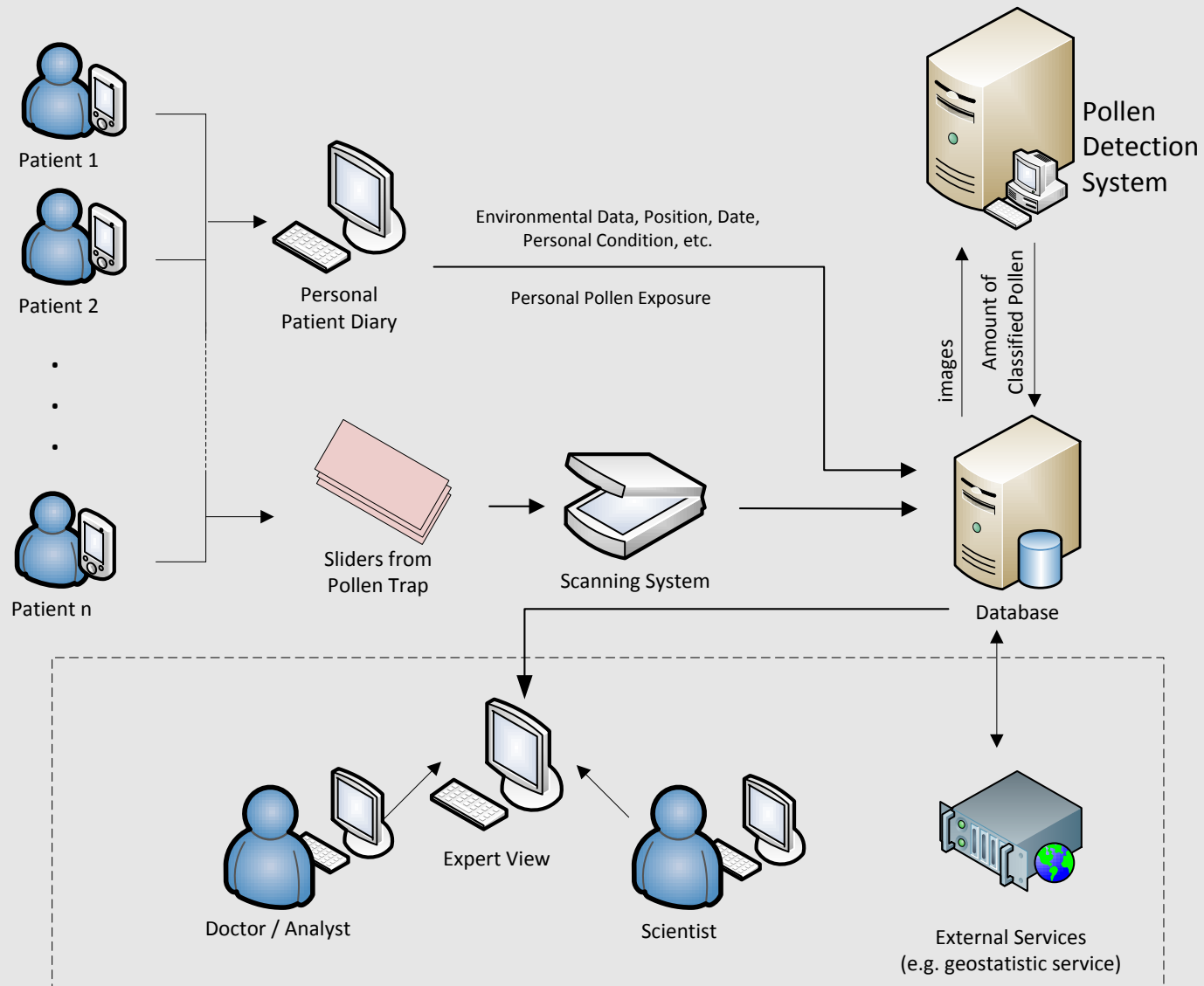
- **41 % aller Kinder und Jugendlichen reagieren in Deutschland auf Allergene**
- **Zahl bei Erwachsenen steigt ebenfalls**
- **Eine detailliertes kleinräumiges Wissen über die tägliche persönliche Pollenbelastung ist daher sehr interessant**

# Das passiert bisher



- **ca. 45 Pollenfallen**
- **Berechnung auf 28 Gebiete**
- **manuelle Auswertung der Fallen**

# Projektskizze



- **Mobile Pollenfalle**
- **Bildgestützte Algorithmen zur Segmentierung und Klassifizierung von Pollen**
- **Verwaltung von Metadaten und Bildinformationen der Pollenfallen**
- **Mobile WebApp als Tagebuch**
- **Experten-Tool zur manuellen Nachbearbeitung und Verbesserung der Algorithmen**
- **Tool für den Arzt zur Auswertung der persönlichen Belastung inkl. der Geovisualisierung**

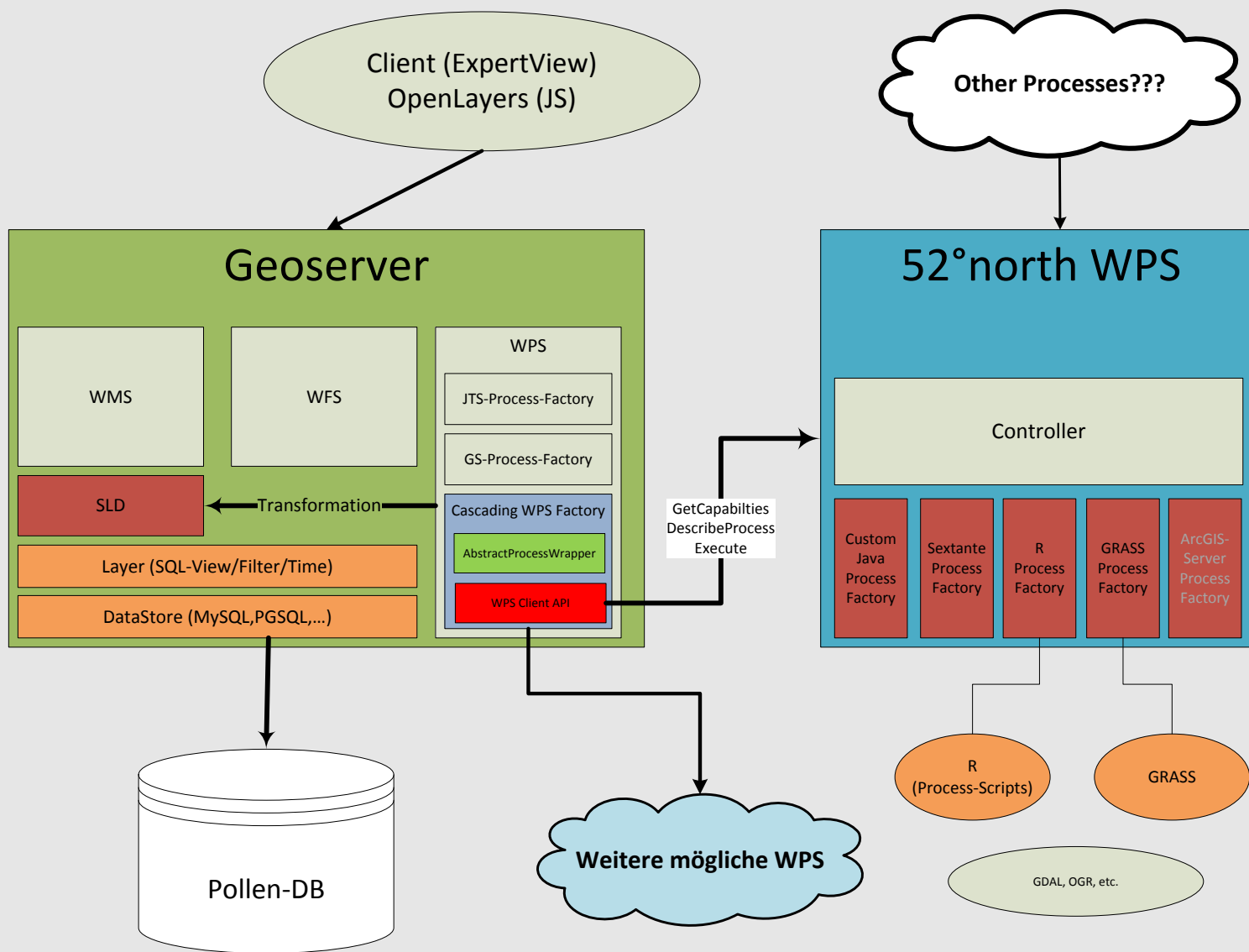
- **Versuch Ausweitung von Mapping**
- **Art der Darstellung schwer vorher zu definieren**
- **Somit möglichst breite Bandbreite an Algorithmen zur Verfügung stellen**
- **Verbindung der Stärken unterschiedlicher Systeme**





# System

Derzeitige Umsetzung



# Komponenten



GeoServer

ubuntu®



OpenLayers

maven



MySQL™



GeoTools

- **Upgrade von GeoTools Version 8**
  - **GML – Parser Veränderung zwischen 2.7 und 8 → keine Interoperabilität mit Geoserver**
- **Konfiguriert mit R-Connector (Rserve)**

VIEW Name  
PollenNutzerbelastung

SQL Statement

```
SELECT
m.ID,user.Name, user.Surname, m.dateTime,
m.gps_position as loc, m.humidity,
m.temperature, m.pressure,
COUNT(part.ID) as
pollenanzahl,pollfam.pollenFamily as
PollenFamily, pollgen.pollenGenus as
PollenGenus,pollspec.pollenSpecies as
PollenSpecies
FROM
tbl_Measurement as m
JOIN
tbl_StripeMeasurements as sm
ON
m.tbl_StripeMeasurements_ID=sm.ID
JOIN
```

SQL VIEW Parameter

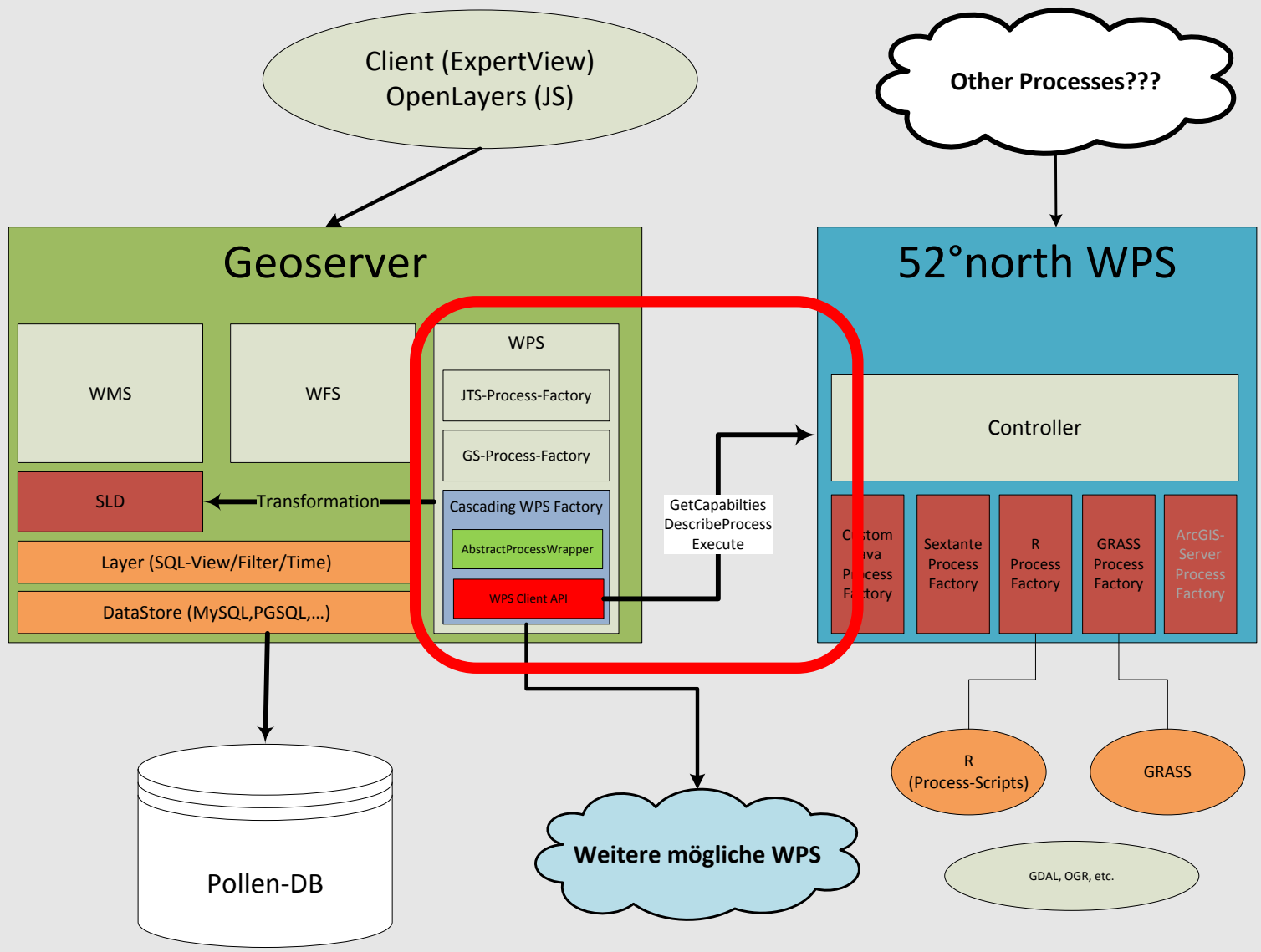
Schlage Parameter vor    Parameter hinzufügen    Ausgewählte löschen

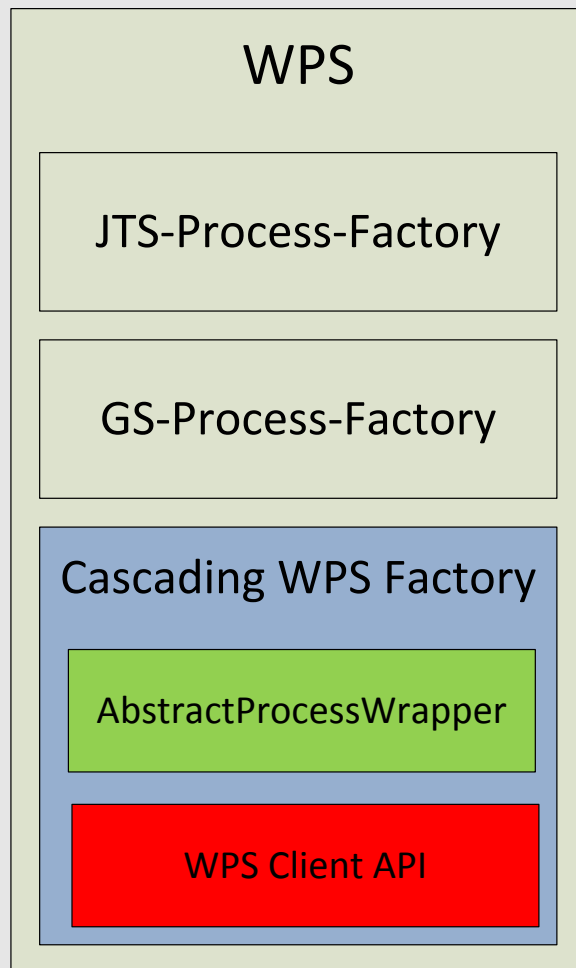
<input type="checkbox"/> Name	Standardwert	Reguläre Ausdruck-Validierung
<input type="checkbox"/> user		^[d]+\$
<input type="checkbox"/> pollenfamily		^[d]+\$

```
14       </Transformation>
15       <ogc:Function name="gs: BarnesSurface">
16        <ogc:Function name="parameter">
17         <ogc:Literal>data</ogc:Literal>
18        </ogc:Function>
19        <ogc:Function name="parameter">
20         <ogc:Literal>valueAttr</ogc:Literal>
21         <ogc:Literal>pollenanzahl</ogc:Literal>
22        </ogc:Function>
23        <ogc:Function name="parameter">
24         <ogc:Literal>scale</ogc:Literal>
25         <ogc:Literal>2500.0</ogc:Literal>
26        </ogc:Function>
27        <ogc:Function name="parameter">
28         <ogc:Literal>pixelsPerCell</ogc:Literal>
29         <ogc:Literal>8</ogc:Literal>
30        </ogc:Function>
31        <ogc:Function name="parameter">
32         <ogc:Literal>outputBBOX</ogc:Literal>
33         <ogc:Function name="env">
34         <ogc:Literal>wms_bbox</ogc:Literal>
35         </ogc:Function>
36        </ogc:Function>
37        <ogc:Function name="parameter">
38         <ogc:Literal>outputWidth</ogc:Literal>
39         <ogc:Function name="env">
40         <ogc:Literal>wms_width</ogc:Literal>
41         </ogc:Function>
42        </ogc:Function>
43        <ogc:Function name="parameter">
44         <ogc:Literal>outputHeight</ogc:Literal>
45         <ogc:Function name="env">
46         <ogc:Literal>wms_height</ogc:Literal>
47         </ogc:Function>
48        </ogc:Function>
49        </ogc:Function>
50       </Transformation>
```

- WPS - Plugin
- MySQL-Plugin
- Definition von unterschiedlichen SQL-Views an PollenDB
- Definition SLD – Styles
  - Transformation aus WPS

# WPS-Schnittstelle





- **Erweiterung des Geoserver mit weiterer Process-Factory**
- **Konfiguration mehrerer Remote WPS möglich**
- **Jeder Remote WPS hat eigenen Namespace, so dass Process-Identifizierer immer eindeutig sind**
- **Einfache Installation**

- **Definition der Factory-Class**
- **Angabe der Parameter:**
  - **Titel des WPS Prozess Dienstes**
  - **Namespace der Prozesse**
  - **URL des Remote-WPS**
  - **Caching der Prozess – Informationen**
  - **Service aktiv**

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">
3 <beans>
4 <bean id="TestCascadingWPSFactory" class="de.i3mainz.geoserver.cascadingwps.CascadingWPSFactory">
5   <constructor-arg index="0">
6     value="Geostatistik 52north WPS Processes" />
7   <constructor-arg index="1" value="52n" />
8   <constructor-arg index="2">
9     value="http://192.168.220.129:8080/wps/WebProcessingService?Request=GetCapabilities&Service=WPS" />
10  <constructor-arg index="3" value="true" />
11  <constructor-arg index="4" value="true" />
12 </bean>
13 <!-- <bean id="TestCascadingWPSFactoryGeoserver" class="de.i3mainz.geoserver.cascadingwps.CascadingWPSFactory"> -->
14 <!-- <constructor-arg index="0" -->
15 <!-- value="Geostatistik Geoserver WPS Processes" /> -->
16 <!-- <constructor-arg index="1" value="GeoServ" /> -->
17 <!-- <constructor-arg index="2" -->
18 <!-- value="http://192.168.220.129:8080/geoserver/ows?Request=GetCapabilities&Service=WPS"
19 /> -->
20 <!-- <constructor-arg index="3" value="true" /> -->
21 <!-- <constructor-arg index="4" value="true" /> -->
22 <!-- </bean> -->
23 </beans>
```



# Factory Class

- **Implementiert ProcessFactory aus GeoTools Framework**
- **Abstrakte Klasse sorgt für die Hauptfunktionalität**
- **Endgültige Factory kann so Client-abhängige Funktionalität beinhalten**
- **Funktionalität für:**
  - **GetCapabilities**
  - **DescribeProcess**
  - **Execute**

```
AbstractCascadingWPSFactory 288
├── F LOGGER : Logger
│   ├── title : InternationalString
│   ├── namespace : String
│   ├── remotewpsurl : URL
│   ├── descriptioncache : boolean
│   └── available : boolean
├── C AbstractCascadingWPSFactory(String, String, URL, boolean, boolean)
├── F isAvailable() : boolean
├── getImplementationHints() : Map<Key, ?>
├── F getTitle() : InternationalString
├── F getNames() : Set<Name>
├── F getTitle(Name) : InternationalString
├── F getDescription(Name) : InternationalString
├── F getVersion(Name) : String
├── F supportsProgress(Name) : boolean
├── F getParameterInfo(Name) : Map<String, Parameter<?>>
├── F getResultInfo(Name, Map<String, Object>) : Map<String, Parameter<?>>
├── create(Name) : Process
├── A createRemoteProcessesCapabilities() : void
├── A fillNamesList() : List<String>
├── A getTitleInternal(Name) : String
├── A getDescriptionInternal(Name) : String
├── A getVersionInternal(Name) : String
├── A supportsProgressInternal(Name) : boolean
├── A getParameterInfoInternal(Name) : Map<String, Parameter<?>>
├── A getResultInfoInternal(Name, Map<String, Object>) : Map<String, Parameter<?>>
├── A getProcessObject(Name) : AbstractCascadingProcess
└── AbstractCascadingProcess
```

- **GeoTools WPS Process Client**
  - **String als Response**
  - **GML als Parameter (ComplexDataType)**
  - **Rasterimages als Parameter/Response**
  - **PPIO aus WPS Server**
- **52north WPS Client API**
  - **Zusätzliche Abhängigkeiten**
  - **Raster**

## Server

- **GetCapabilities**
- **DescribeProcess**
- **Execute mit LiteralData**
- **Einbinden von Geoserver WPS in Styles**

### Offen:

- **Verarbeitung von komplexen Datentypen (GML, Rasterimages)**

## Mapping Client

- **OpenLayers Map**
- **Ansicht pro Patient / Pollenart**
- **Erweiterter LayerSwitcher**

### Offen:

- **TimeSlider / Timeline**
  - **Openlayers TimeSlider**
  - **GeoTempCo**

- **Flexibles System soll vielfältige Möglichkeit bei Nutzung von geostatistischen Visualisierung bieten**
- **Nutzung unterschiedlicher Frameworks**
- **Geoserver bietet Definition von Prozessen in der Style-Definition**
- **WPS Client sehr rudimentär entwickelt**
  - **Entwicklungsaufwand höher als eingeschätzt**
- **Test der Visualisierung benötigt zunächst Daten aus einer Test-Studie**

Vielen Dank für die Aufmerksamkeit

**Nikolai Bock**

**i3mainz – Institut für  
Raumbezogene Informations-  
und Messtechnik**

**[nikolai.bock@fh-mainz.de](mailto:nikolai.bock@fh-mainz.de)**