Dziedzinowo zorientowane usługi i zasoby infrastruktury PL-Grid dla wspomagania Polskiej Nauki w Europejskiej Przestrzeni Badawczej

DataNet – Flexible Metadata Overlay over File Resources

Daniel Harężlak[1], Marek Kasztelnik[1], Maciej Pawlik[1], Bartosz Wilk[1], Marian Bubak[1,2]

[1]ACC Cyfronet AGH,
[2]AGH University of Science and Technology, Institute of Computer Science AGH

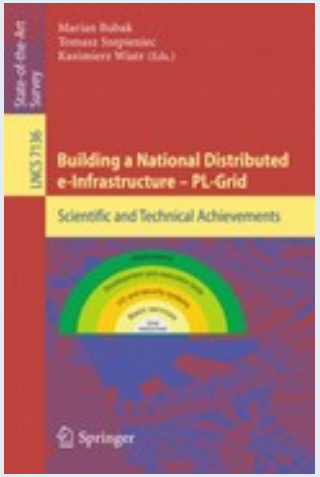
Workshop on Cloud Services for File Synchronisation and Sharing
November 17-18, 2014, CERN

Presentation Plan

- PL-Grid Computing Infrastructure
- Motivation behind DataNet
- Metadata Management Requirements
- Architecture Description
- Deployment
- Conclusions

DataNet – PLGrid Landscape

- PL-Grid Programme
  - PL-Grid
    - Computing power: ca. 230 Tflops
    - Storage: ca. 3600 Tbytes
    - Basic infracructure services
  - PL-Grid PLUS
    - Additional 500 Tflops and 4,4 Pbytes
    - Support for domain Grids



DataNet – Rationale and Objectives

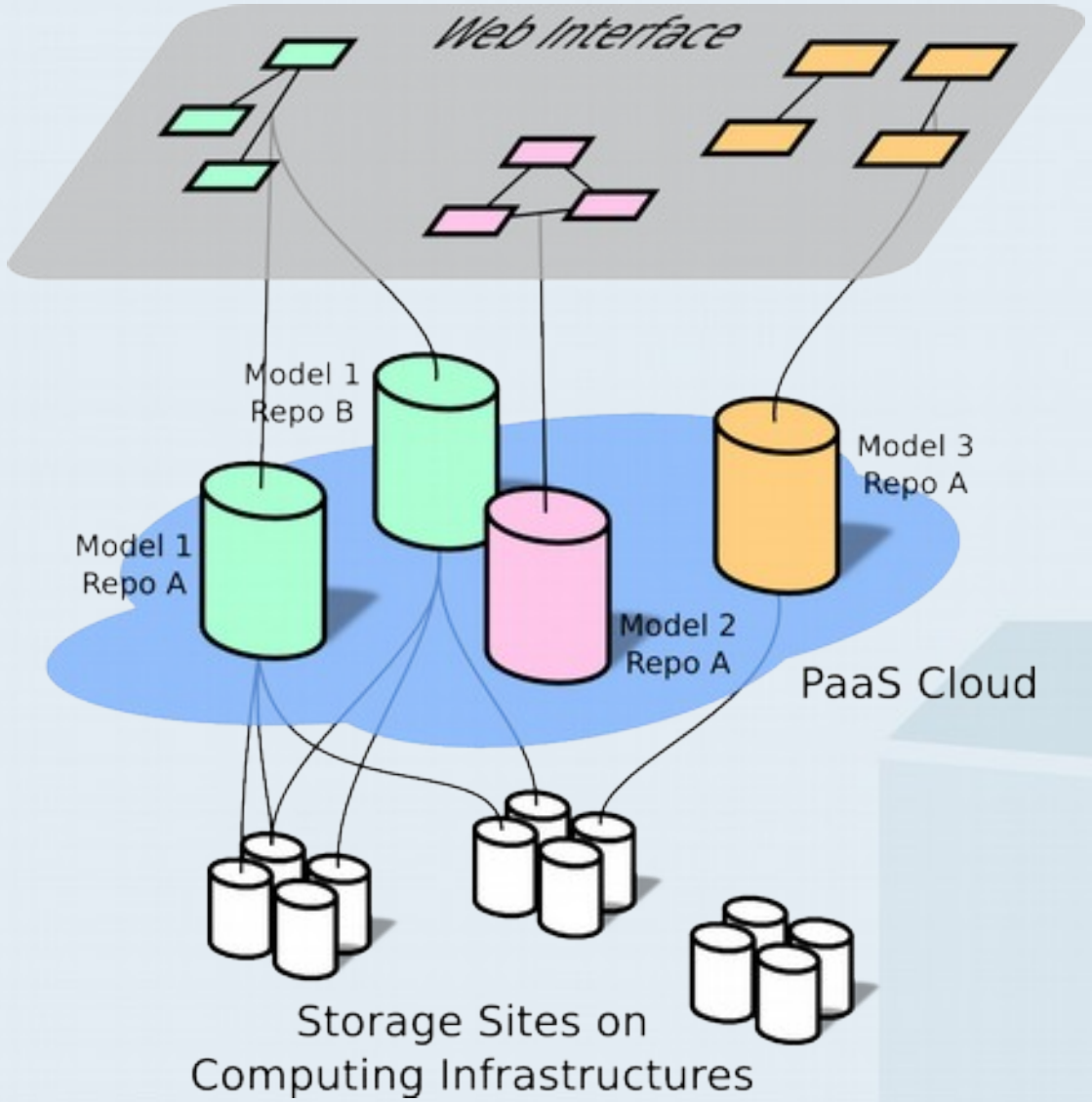
- Rationale
  - Data management as a common requirement in computational sciences
  - Workflow and scripting engines provide only a little support
  - Each application is different and requires a dedicated metadata/data model
- Objectives
  - Provide means for **ad-hoc metadata model creation** and deployment of corresponding storage facilities
  - Create a research space for **metadata model exchange and discovery** with associated data repositories with access restrictions in place
  - Support **different types of storage sites** and **data transfer protocols**
  - Support the exploratory paradigm by making the models evolve together with data

DataNet – PLGrid Requirements

- PLGrid infrastructure – supporting different e-Science domains
  - Various applications coming from different scientific communities
  - Common computational resources
- Deployment of model data as repositories
  - Robust enablement of a dedicated interface
  - Access control capabilities
  - Exploitation of available storage infrastructure
- Universal availability of the repository
  - Platform independent
  - Facilitated by existing standards

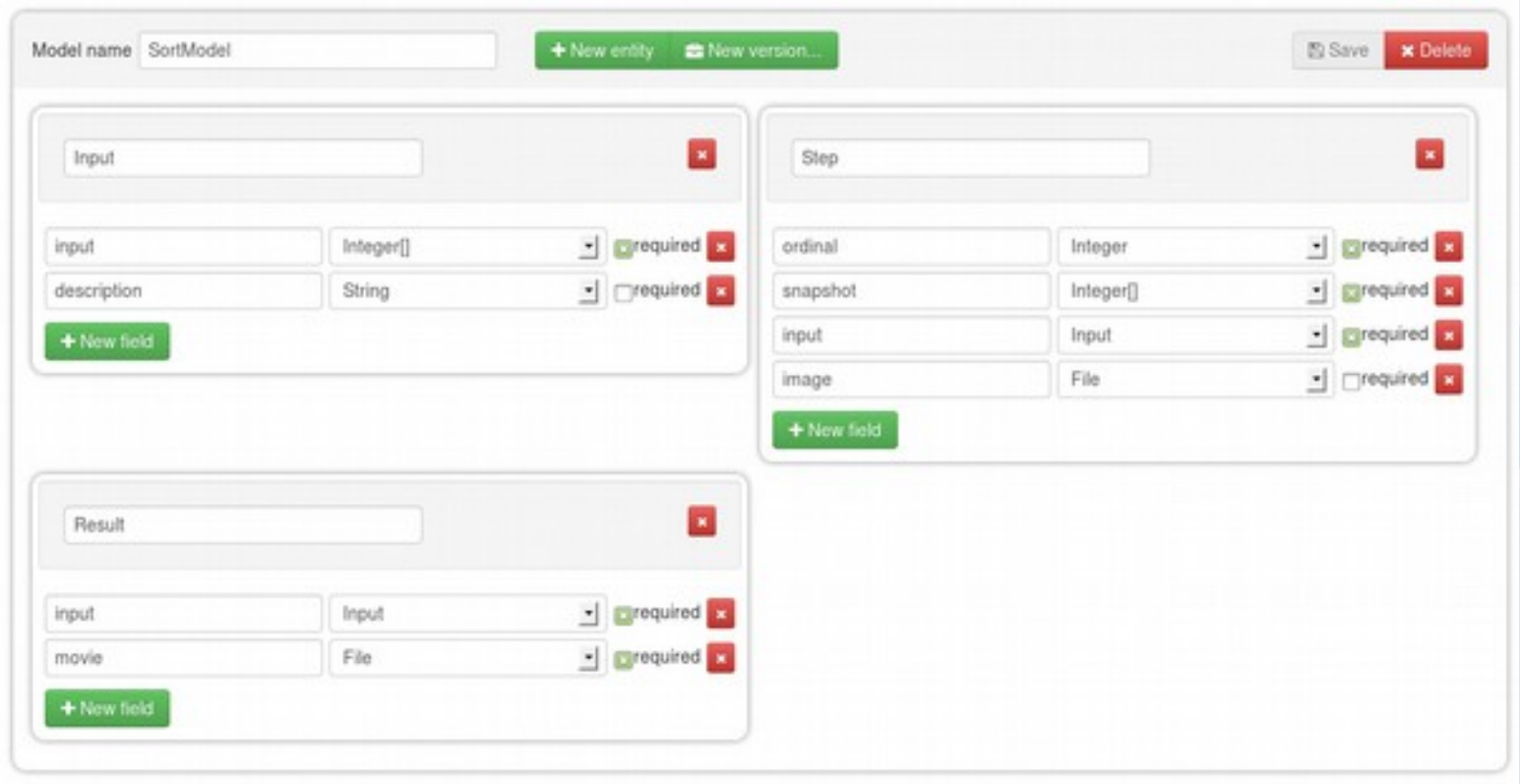
DataNet – Architecture

- **Web Interface** is used by users to create, extend and discover metadata models
- Model repositories are deployed in the **PaaS Cloud** layer for scalable and reliable access from computing nodes through REST interfaces
- Data items from **Storage Sites** are linked from the model repositories



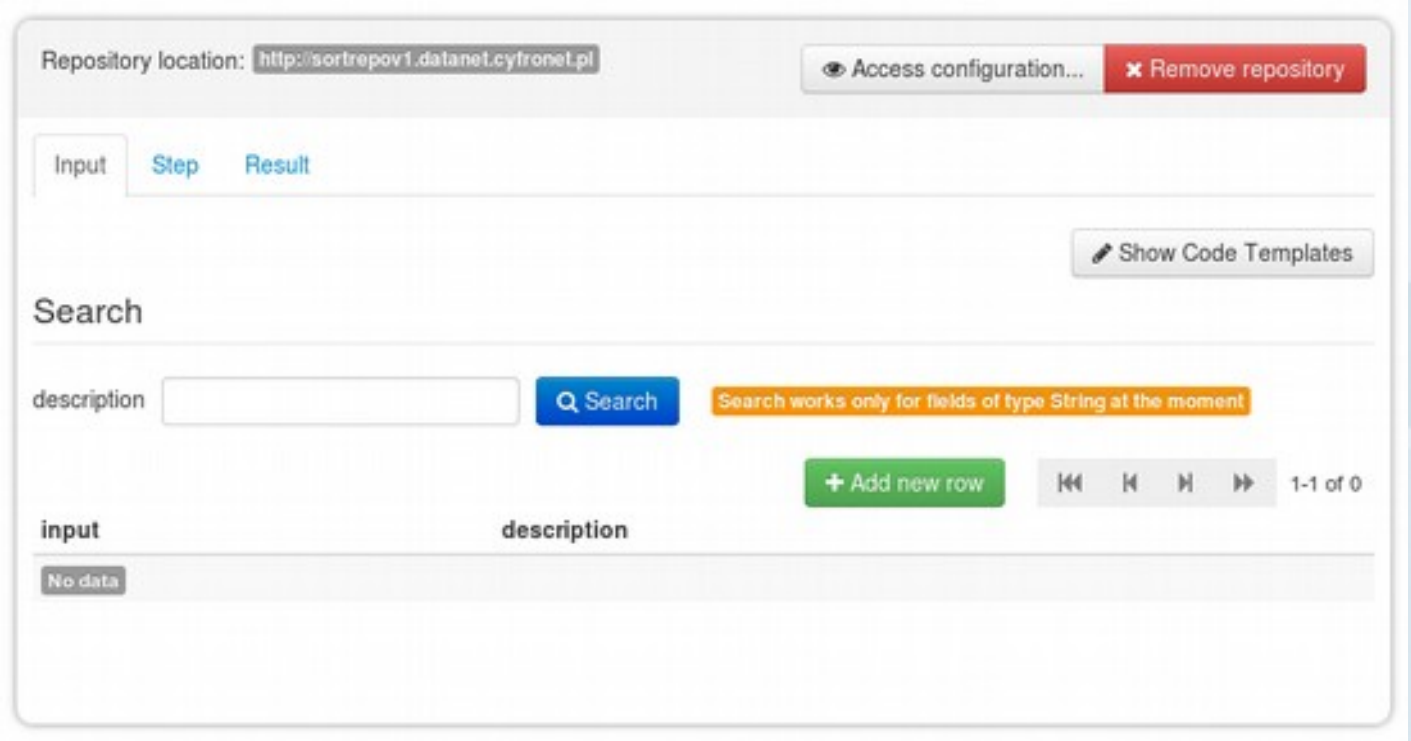
DataNet – Data Model

- Set of entities with fields
  - Simple types
  - Array types
  - File type
  - Relations



DataNet – Repository

- Repositories are accessed through REST
  - Data view through a web application
  - Configurable Access control
    - Public
    - Private (within a group of users)



DataNet – Repository Access

- Data sent over with JSON or FORM
  - REST methods
    - POST – submit new data
    - PUT – modify data
    - DELETE – remove data
    - GET- retrieve data
    - Queries with URL

```
require 'rest-client'
require 'json'

datanet=RestClient::Resource.new('http://a:a@repo.datanet.cyfronet.pl')
datanet.get
def get_user(first_name, last_name)
  {first_name: first_name, last_name: last_name}.to_json
end
get_user "marek", "kasztelnik"
datanet['user'].post get_user("Marek", "Kasztelnik")
datanet["user"].get
10.times {datanet['user'].post get_user("Marek", "Kasztelnik")}
datanet["user"].get
datanet["user/519dbfed2fbb0c79f400000b"].delete
datanet["user"].get
```

```
import requests as req
import json

headers = {'content-type': 'application/json'}
resp = req.post('http://test5.datanet.cyfronet.pl/Hello',
        data = json.dumps({'name': 'hello1'}), auth = ('', ''), headers = headers)
```

DataNet – PLGrid Deployment

- PLGrid Users
  - Access with regular PLGrid account
  - REST interface
  - Web Application for simple use cases
- Domain applications
  - User proxy delegation retrieved from PLGrid OpenID provider
  - REST interface
- DataNet as a Service already in place
  - Deployment procedure followed

DataNet – Service summary

Status	Production
Number of users	5 – 10 (30 – 40 planned)
Default and Maximum Quota	5GB (Home), 100GB (Storage)
Linux/Mac/Win user ratio	3/0,5/2
Desktop/Mobile/Web access ratio	Web only
Technology	Cloud Foundry, Java, Ruby
Target communities	Researchers in various fields
Integration in your current environment	Using the same storage elements
Risk factors	Network reliability (upload interruptions)
Most important functionality	REST access
Missing functionality (if any)	None so far

DataNet – User Feedback

- Easy integration in any programming language
- Possibility to view data via a web application
- Eesy extension to other metadata engines

- Not and end-user software

DataNet – DONEs and TODOs

- DONEs
  - Custom CloudFoundry environment was setup as a PaaS platform to ensure quick deployments of required application and storage services
  - Schema for metadata model creation was elaborated and was evaluated for NoSQL storage service MongoDB
  - Storage site access libraries were implemented and tested
  - Deployment of a web-based tool to create, discover and manage metadata models
- TODOs
  - Support various types of metadata storage services to fulfil different application requirements (if required)
  - Prototype a utility for data migration between model versions

Thank You

- Acknowledgements
  - This research has been partially supported by the European Regional Development Fund program no. POIG.02.03.00-00-096/10 as part of the PL-Grid PLUS project

- Contact us and help make DataNet better

- Visit http://dice.cyfronet.pl for more information

- See DataNet in action at https://datanet.cyfronet.pl (PLGrid account required)