“Integrating scientific laboratories into the cloud”

Data Infrastructure Track

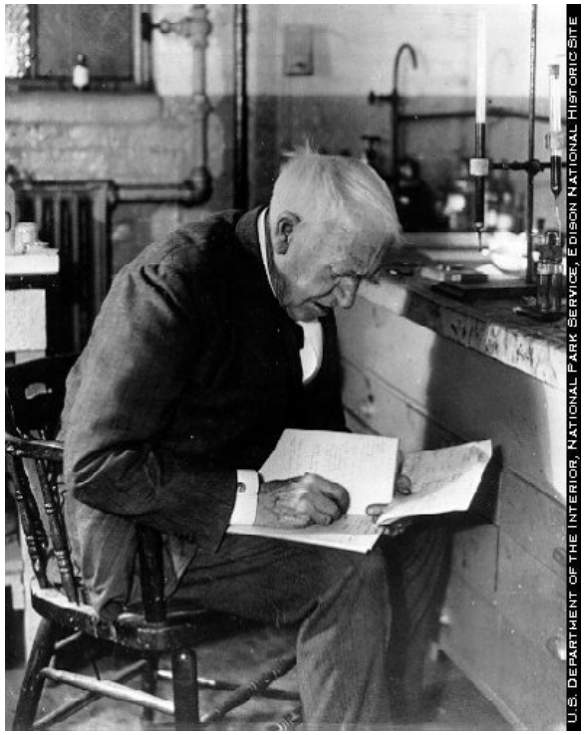
Miriam Ney < miriam.ney@dlr.de >
German Aerospace Center
http://www.dlr.de/sc

Overview

- Scientific Work In The Past
  - Paper based notebooks: unstructured notes
- Scientific Work At The Moment
  - Data management system: DataFinder
- Scientific Work In The Future
  - Enhanced Data Management: DataFinder using Grid and Cloud
- Scientific Work Summarized
  - Conclusion of the talk

Scientific Work In The Past

Scientific Work In The Past
Documentation by hand



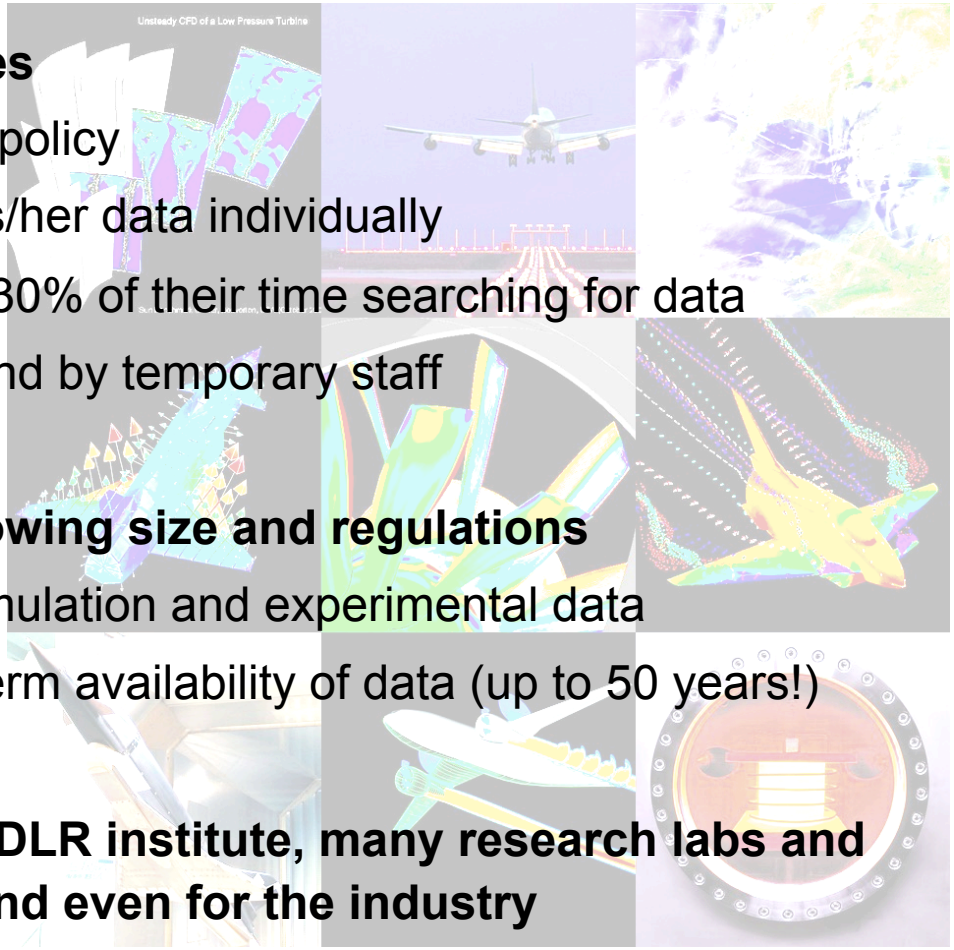
The principles of Good Laboratory Practice (GLP) have been developed

to promote the quality and validity of test data used for determining the

safety of chemicals and chemicals products.

OECD Principles on Good Laboratory Practice
(as revised in 1997)

Scientific Work In The Past
Problems with data management



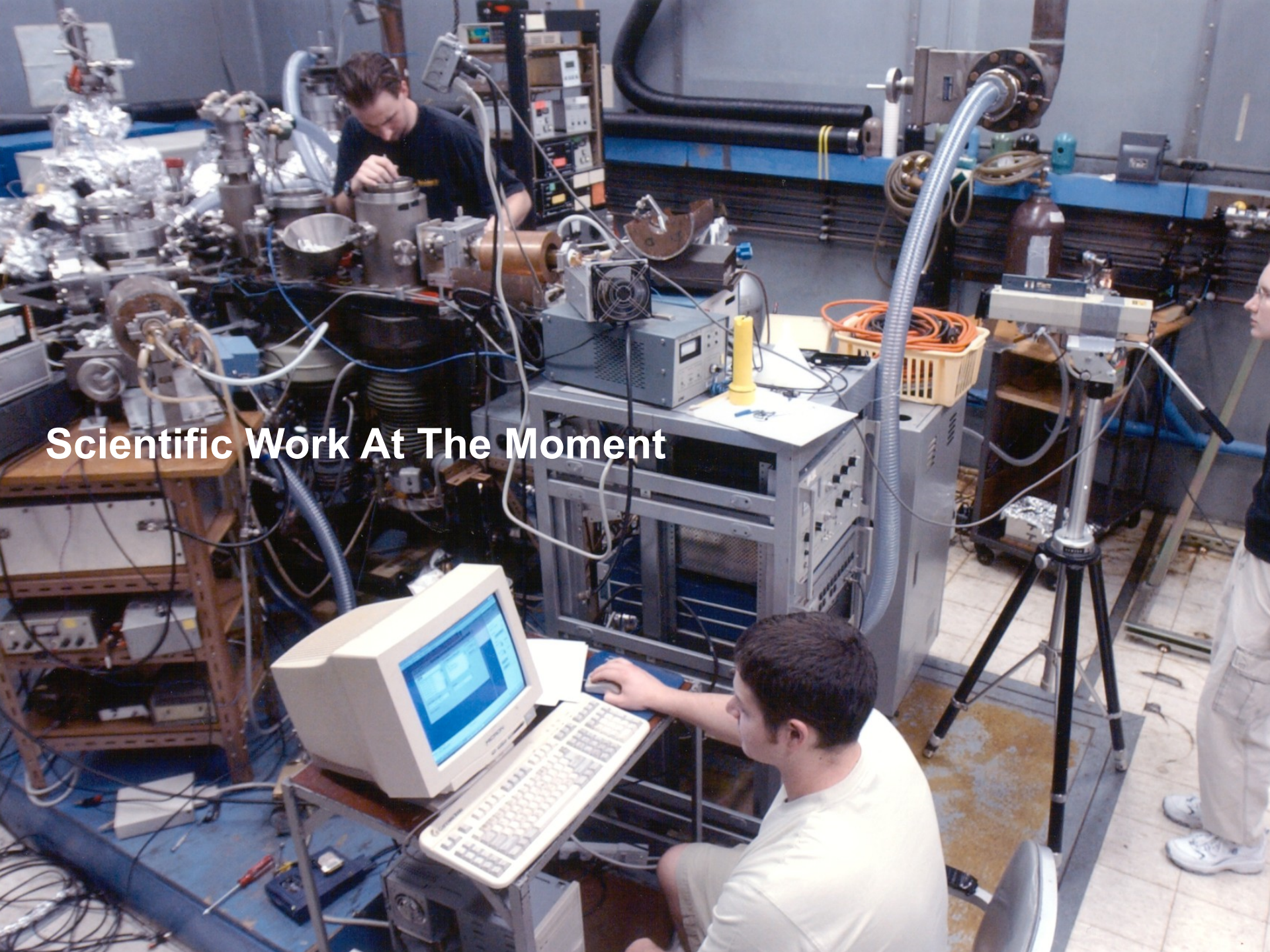
Absent organizational structures

- No central data management policy
- Every employee organizes his/her data individually
  - → Researchers spend about 30% of their time searching for data
  - → Problem with data left behind by temporary staff

Increase of data because of growing size and regulations

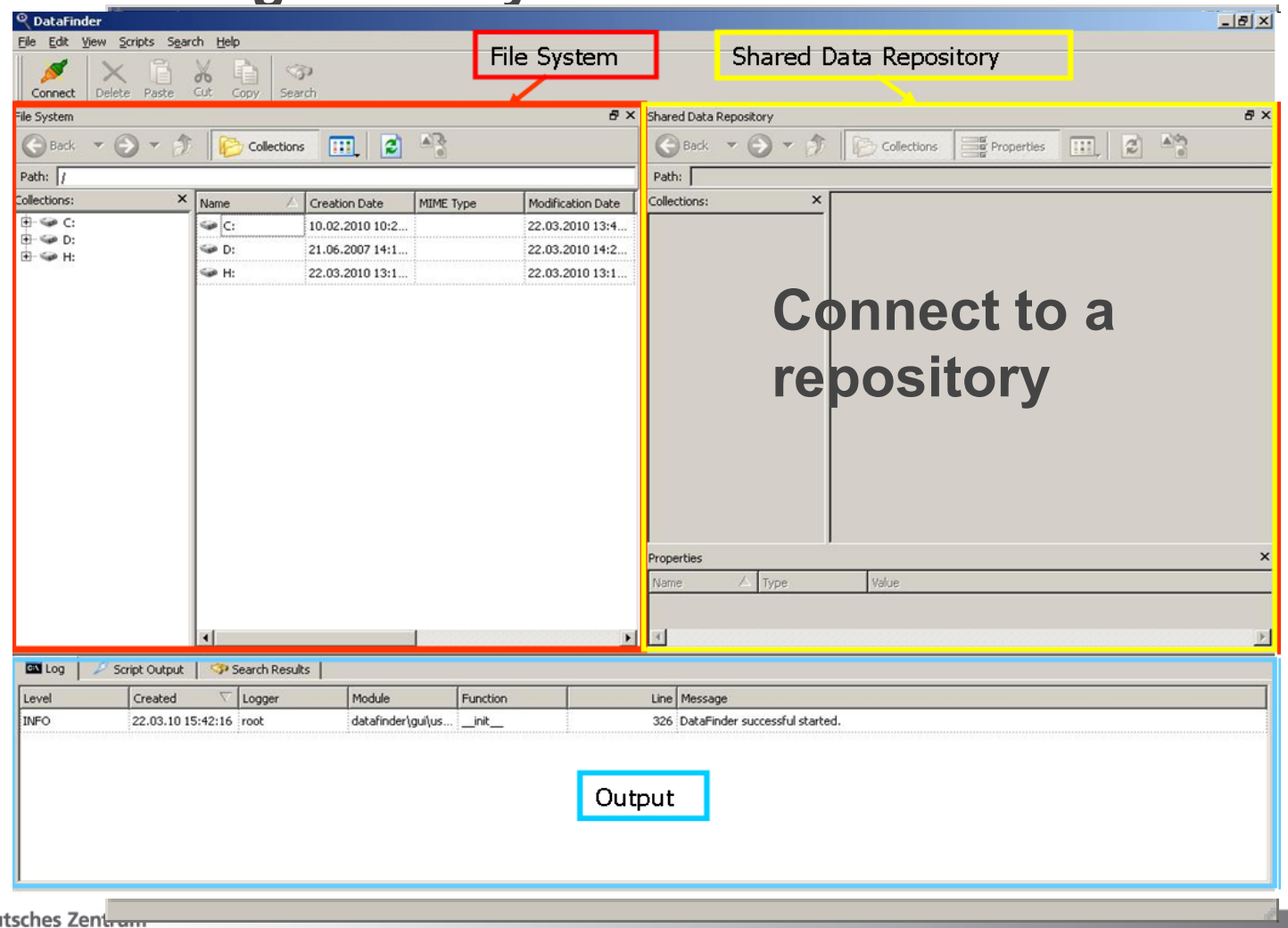
- Rapidly growing volume of simulation and experimental data
- Legal requirements for long-term availability of data (up to 50 years!)

Situation is similar for every DLR institute, many research labs and agencies and even for the industry


Scientific Work At The Moment

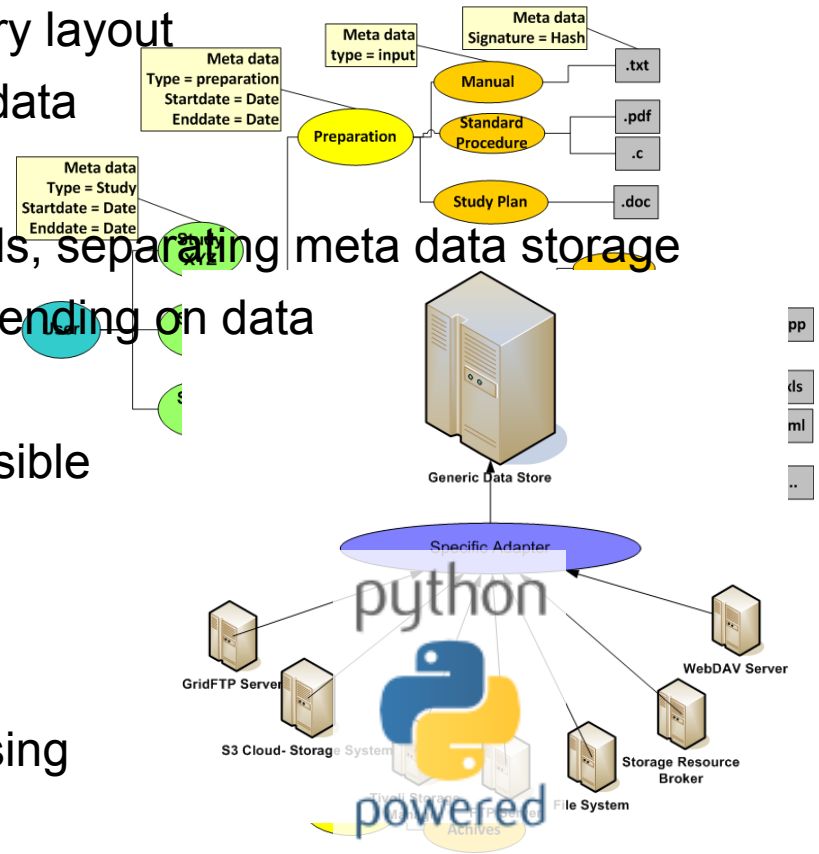
Scientific Work At The Moment

Data management system: DataFinder



Scientific Work At The Moment
Features of the DataFinder

- Structuring of data with a data model
  - Restricting the user to a directory layout
  - Forcing the user to enter meta data
- Using heterogenous storage backends, separating meta data storage
  - Best fitting storage solution depending on data
  - Existing resources can be kept
  - Offline storage location are possible
- Extension by scripts
  - Adjusting to your environment
  - Automation of workflow processing



Scientific Work At The Moment

Demo of storing data in the cloud

DataFinder

File Edit View Scripts Search Help

Connect Delete Paste Cut Copy Search

File System

Back Collections

Path: /

Collections:

C:
D:

Name	Creation Date	MIME Type	Modification Date	Owner
C:	30.09.2009 14:0...		19.03.2011 15:2...	
D:	30.09.2009 14:0...		19.03.2011 15:4...	

Shared Data Repository

Back Collections Properties

Path:

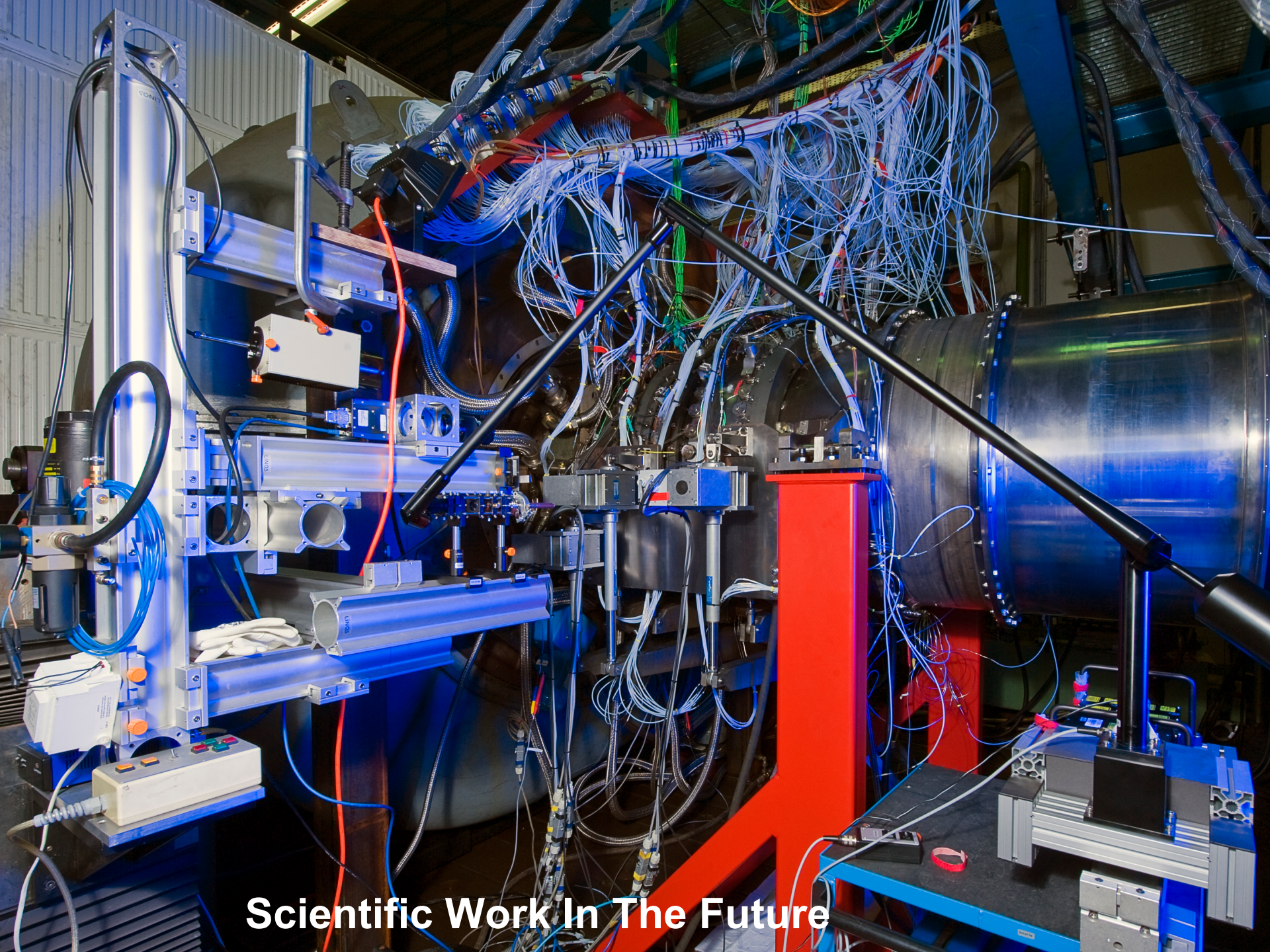
Collections:

Properties

Name	Type	Value

Log Script Output Search Results

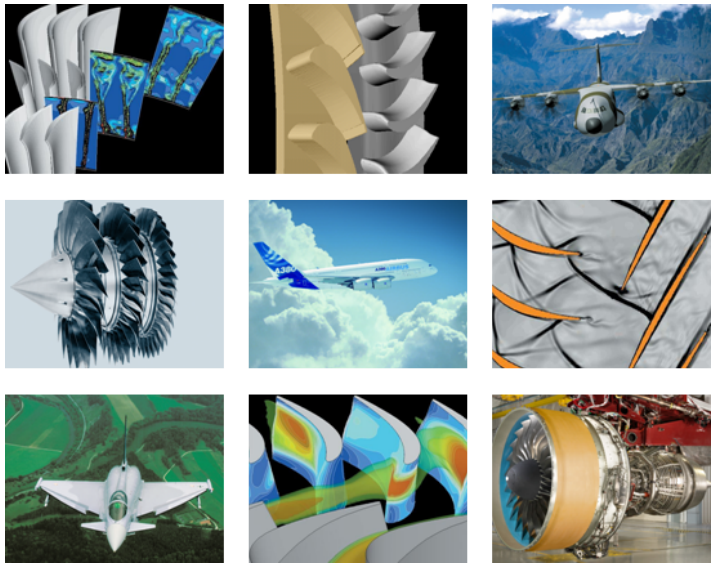
Level	Created	Logger	Module	Function	Line	Message
INFO	19.03.11 16:04:14	root	datafinder.gui.us...	__init__	382	DataFinder successful started.



Scientific Work In The Future

Example 1: Accessing Grid Infrastructure Fluid Dynamics Simulation

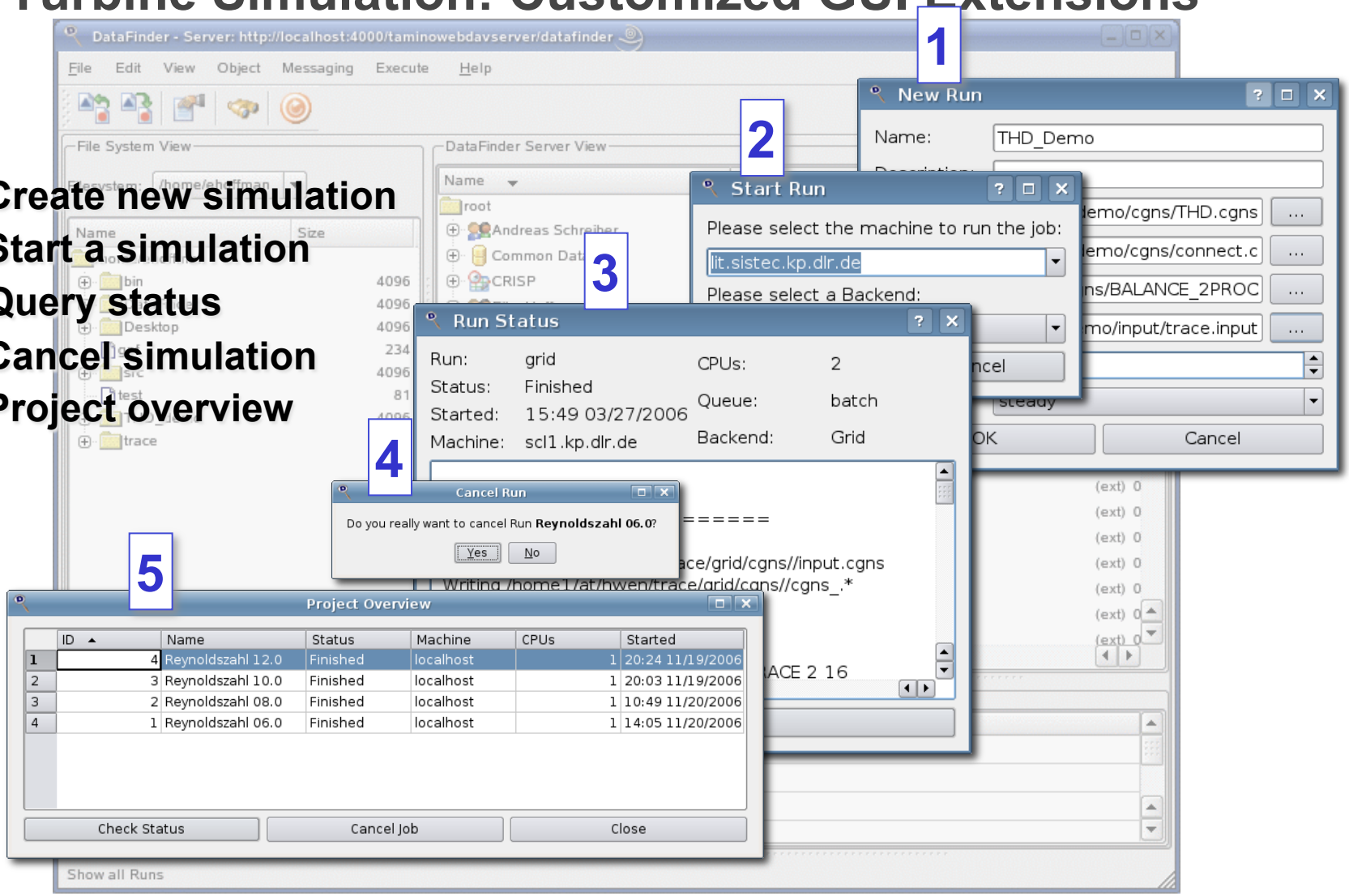
- Design of new turbine engines
- High-resolution simulation of flow
  - Computational Fluid Dynamics (CFD)
  - Use of high-performance computing resources (Cluster / Grid)
  - Huge amounts of data (>100 GByte)
- DataFinder used for
  - Management of results
  - Automation of simulation runs in Grid
  - Starting pre-/post processing



AERO GRID

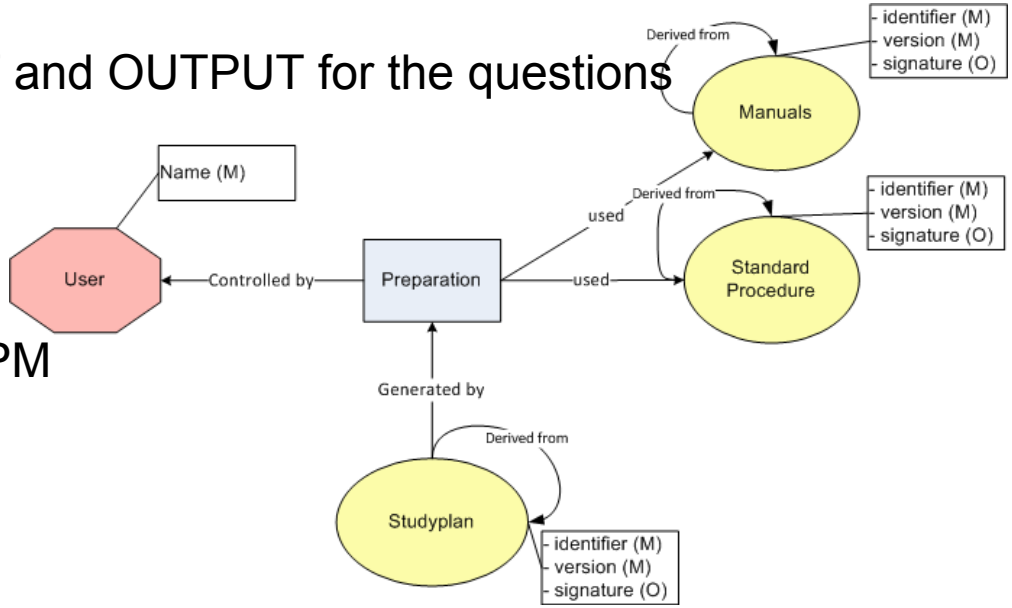
Turbine Simulation: Customized GUI Extensions

1. Create new simulation
2. Start a simulation
3. Query status
4. Cancel simulation
5. Project overview



Example 2: Provenance Integration
Developing a provenance model with PrIMe

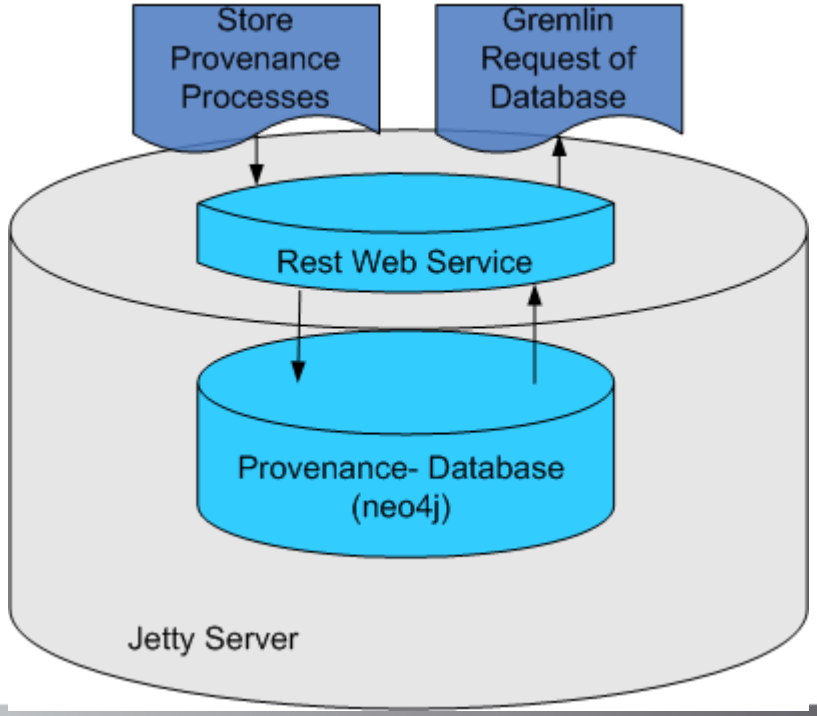
- *„The Provenance of a piece of information is the history of its creation“*
- Collecting QUESTIONS that should be answered by the system
  - Which item is the logical predecessor of item X?
- Identifying ACTORS, INPUT and OUTPUT for the questions
- Extracting PROCESSES
- Modeling Processes with OPM



Provenance Aware Application and Provenance Storing System

- DataFinder
  - Script watching on import event of document (output)
  - Extracting Process and Input
  - Sending the information to the „Noblivious“ service

- „Noblivious“ Provenance- Service
  - Web Server
  - Graph Database neo4j
  - Rest Interface
    - Storing processes
    - Querying the database



Scientific Work Summarized

Scientific Work Summarized
Conclusion

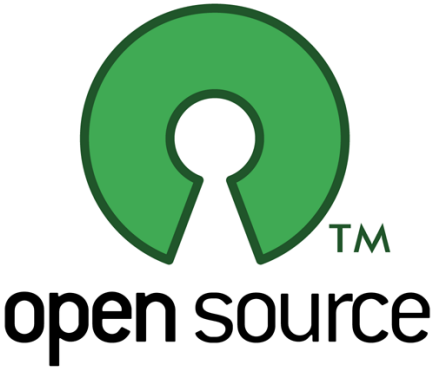
- DataFinder **is used** in several scientific laboratories at DLR and other German and international research institutes
- **Storing** of data in grid and clouds is possible and with DataFinder the scientist does not have to worry about configuration
- **Execution** of jobs on grids (and clouds) can be integrated/ started
- DataFinder can support collecting **provenance information** of data
- **`http://launchpad.net/datafinder`**

Questions?

Contact:
Miriam Ney
DLR Simulations- und Softwaretechnik, Berlin
Email: Miriam.Ney@dlr.de

Availability

- DataFinder core available as Open Source
  - Current stable release: DataFinder 2.1
  - Simplified BSD License
  - Open Source
    - **Launchpad (Code)**
    - **Sourceforge (Binaries)**
    - **Freshmeat (Announcement)**



Links

DataFinder Website

`http://www.dlr.de/datafinder`

DataFinder Projectpages

- `http://launchpad.net/datafinder`
- `https://launchpad.net/~datafinder-team`

- `http://sourceforge.net/projects/datafinder`

DataFinder Wiki

- `http://wiki.sistec.dlr.de/DataFinderOpenSource`

facebook Suche Startseite Profil Konto

DataFinder

Pinnwand | Info | Fotos | Diskussionen | Felder | SlideShare | +

Data Finder
Organize your data

Was machst du gerade?

Anhängen: Teilen

DataFinder + andere | DataFinder | Nur Andere | Einstellungen

DataFinder DataFinder-Vortrag auf der FrOSCon 2010

froscon2010: DataFinder
programm.froscon.org
Der DataFinder ist eine in Python entwickelte Open
Veröffentlicht unter der Simplified BSD Lizenz, ermöglicht sie es einfach, große Datenmengen,
wie sie häufig bei wissenschaftlichen Simulationen und Versuchen anfallen, zu verwalten. Dabei hilft die konsequente A...

08. Juli um 14:04 · Kommentieren · Gefällt mir · Teilen · Bewerben

Miriam Ney gefällt das.

XEmacs Slartibartfast Wird der Vortrag auch per Video aufgezeichnet & zur Verfuegung gestellt? Gibt es dazu auch eine Veroeffentlichung oder ein Paper, welches man an andere weitergeben kann, oder verlinken kann?
vor einigen Sekunden · Gefällt mir · Löschen · Melden

Schreibe einen Kommentar ...

DataFinder The face behind DataFinder :) (german)

audimax.de Masterstudium, Berufseinstieg, Studium, Karriere: Komplexe Software sucht Entwicklungshel
www.audimax.de
Du bist Student oder Absolvent? audimax.de ist deine Informationsplattform zu den Themen Studium, Berufseinstieg, Karriere und Masterstudium. Mit Studienhilfe, Stellenanzeigen, Gewinnspielen, Tipps und Tricks fürs Auslandssemester und vielem mehr.

25. Mai um 22:11 · Kommentieren · Gefällt mir · Teilen · Bewerben

DataFinder Last Friday:
DataFinder will be developed further using Launchpad... (Kind of fits for a Software developed by a space agency :-).

So check out the project: http://launchpad.net/datafinder

Seite bearbeiten
Mit Werbeanzeige bewerben
Zu den Favoriten meiner Seite hinzufügen
Freunden vorschlagen

DataFinder is a data management client developed in Python that primarily targets the management of scientific technical data. The system is able to handle large amounts of data and can be easily integrated in existing working environments.

Informationen

Gegründet:
2002

Statistiken

Alle anzeigen

0 ★☆☆☆☆ Qualität der Beiträge

0 Interaktionen In dieser Woche

Statistiken sind nur für Administratoren der Seite sichtbar.

13 Freunden gefällt das

Werbeanzeige erstellen

Facebook-Seiten

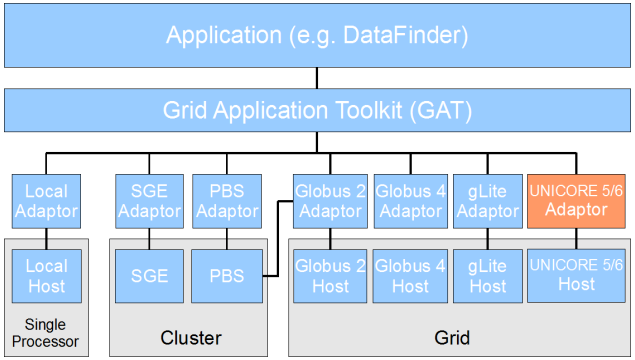
Facebook-Seiten helfen dir
Leuten mit denen verbunden
die du bereits magst.

Weitere Werbeanzeigen

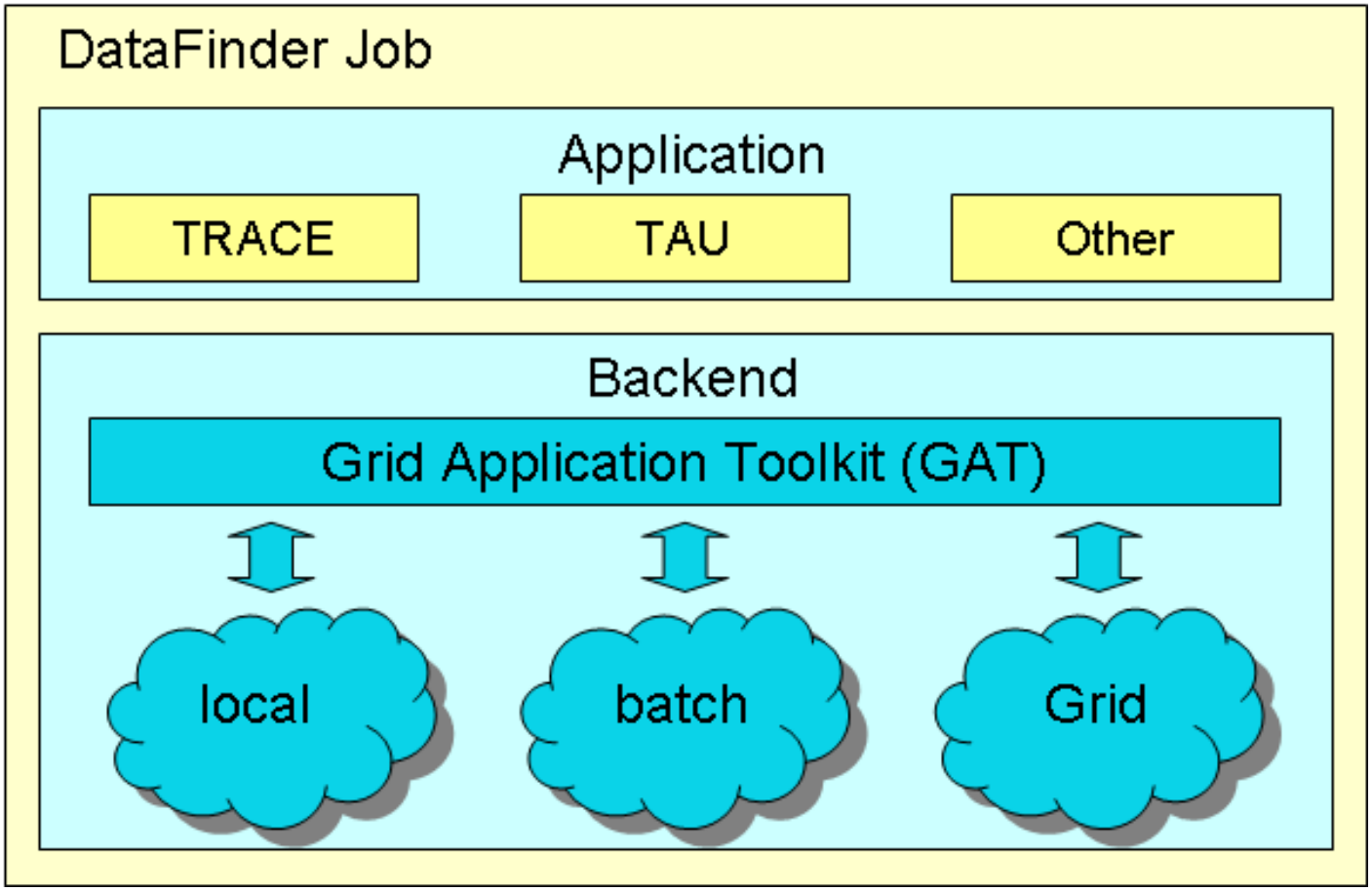
Become DataFinder Fan

Turbine Simulation
Job Management

- Usage of the abstraction layer JAVA-GAT for job submission
  - Provides a simple API to several grid applications
  - Based on **HiLA**
    - HiLA(*High-Level API)* supports the access to UNICORE 5 and UNICORE 6 via an easy and unique API.
    - It is not necessary to install components of UNICORE 5 or UNICORE 6 on the submitting (client) host.
- Current implementation allows performance of:
  - Local jobs,
  - Batch systems jobs,
  - UNICORE 5 / 6 jobs.



Turbine Simulation
Job Management Concept



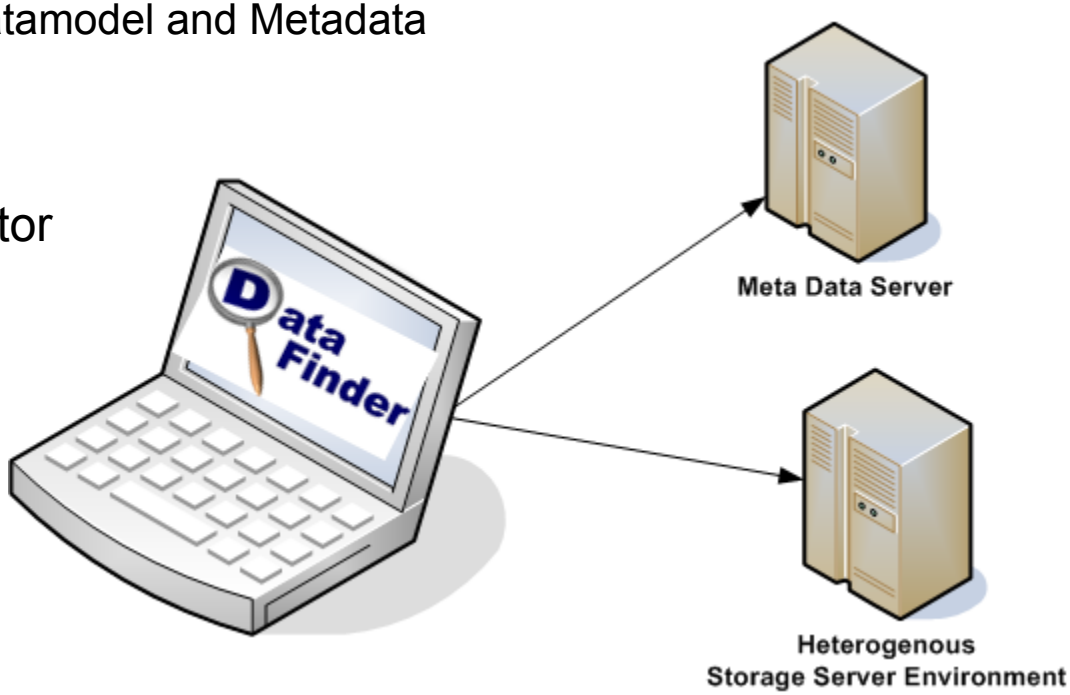
DataFinder: Concepts
Concepts for managing huge Datasets

- **Infrastructure:** Server Client Structure
- **Structuring Data:** Meta Information and Data Models
- **Flexible Resource Usage:** Data Stores
- **Environment Integration:** Extension with Scripts
- **Programming language:** Python

Suitable software for efficient management of scientific and technical data

DataFinder: Concepts
Distributed System

- **Client-Server solution**
- Based on **open and stable standards**
- Server:
  - **WebDAV server** for datamodel and Metadata
  - **Data Store** concept
- Client:
  - User and Administrator

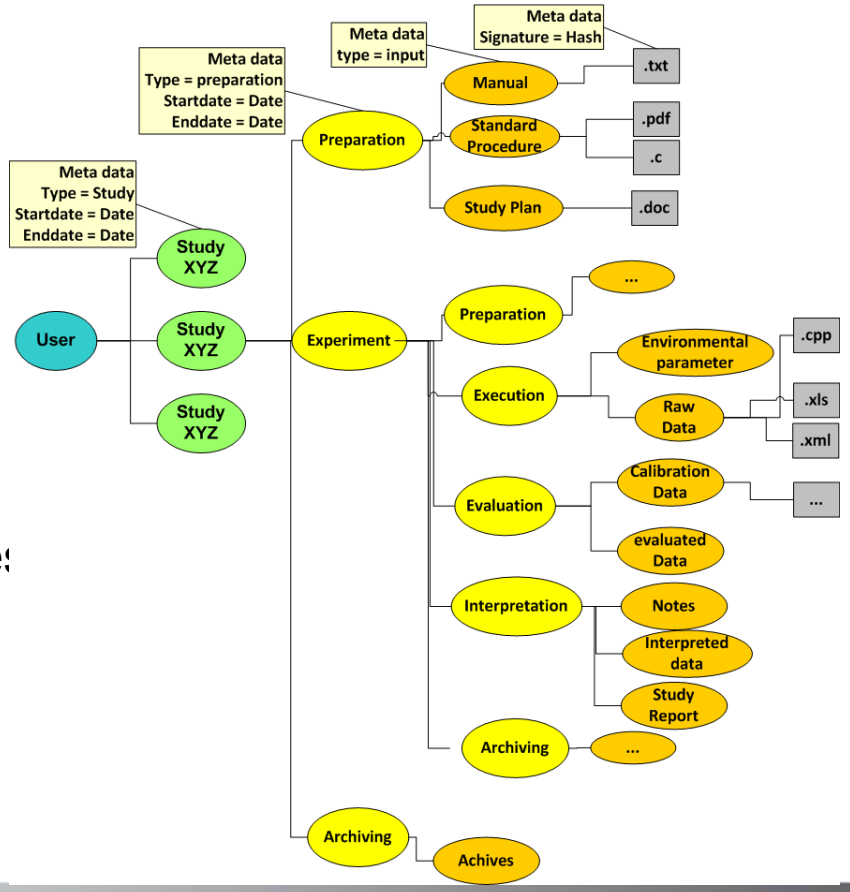


DataFinder: Concepts
Data Model and Meta Data

- Definition of data structuring and meta data (“data model”)
- Stored in XML format
- User can search in meta data

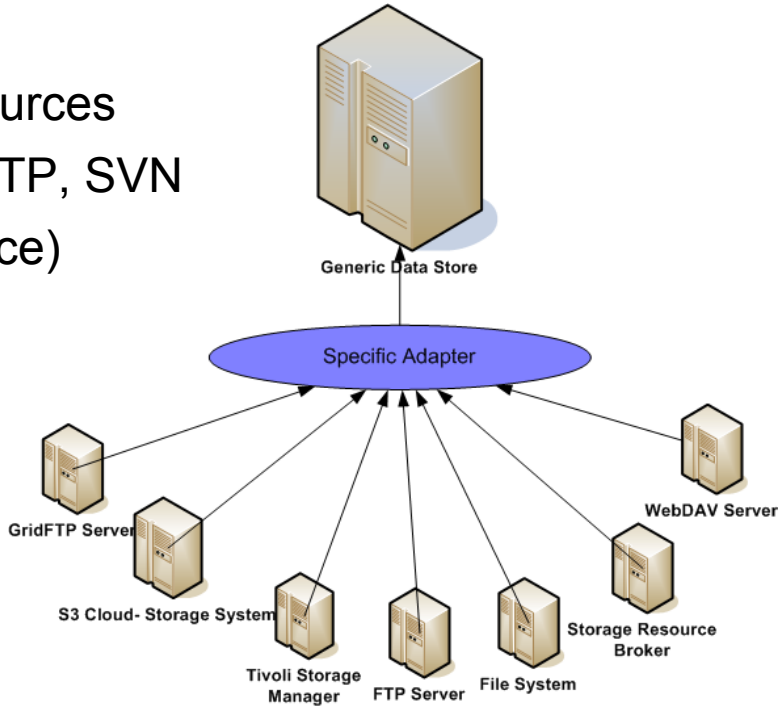
Variation:

- Different levels of meta data
  - Administrator: required attributes
  - User: additional ones
- Different types of meta data

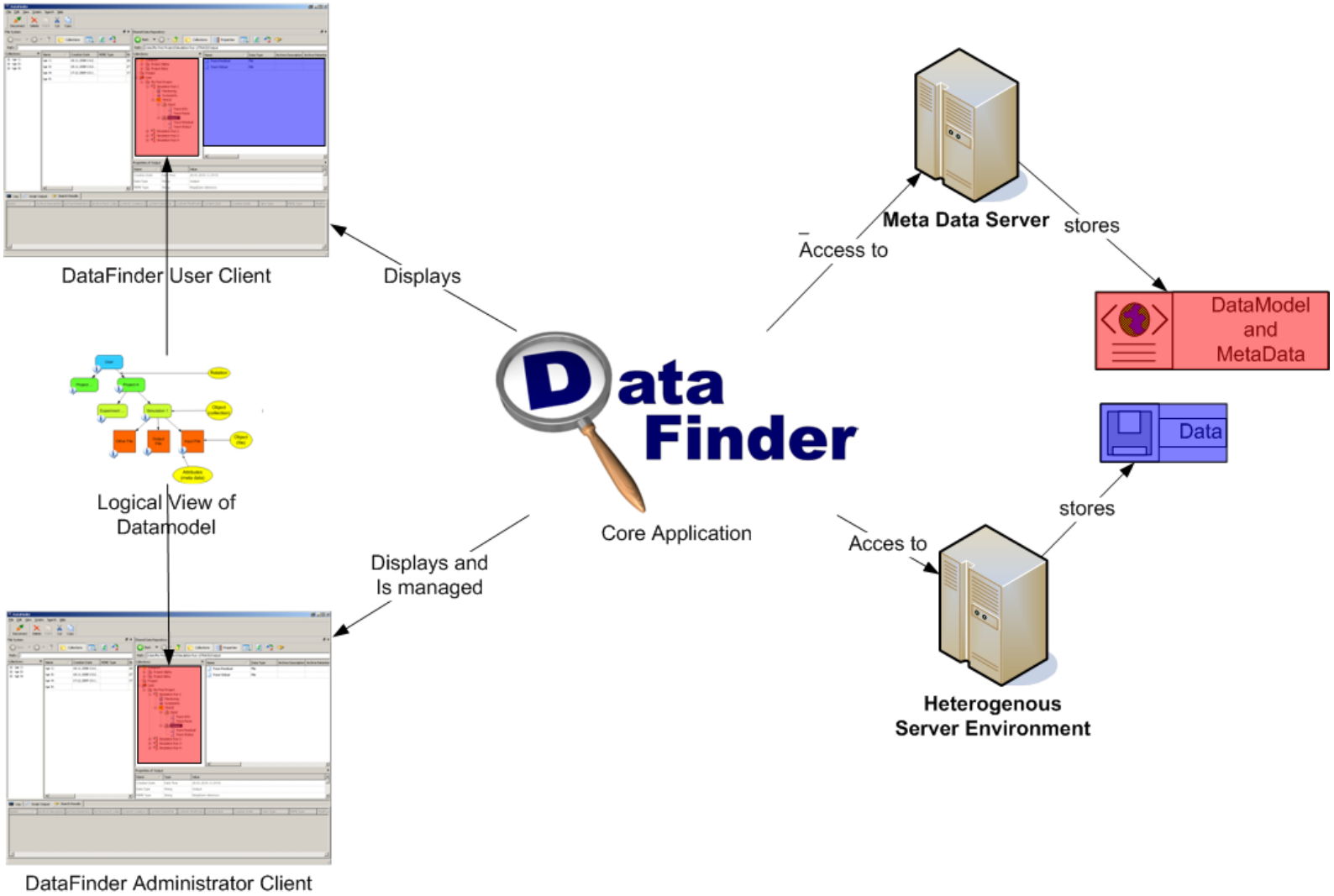


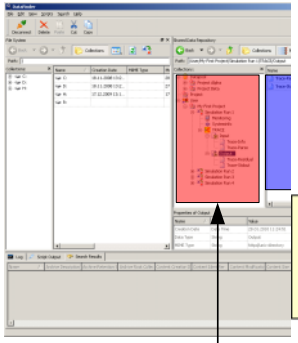
DataFinder: Concepts
Data Stores

- Abstracting the Users logical view of the server structure
- Separated storage of data structure / meta data and actual data files
- Flexible use of (distributed) storage resources
  - File system, WebDAV, FTP, GridFTP, SVN
  - Amazon S3 (Simple Storage Service)
  - Tivoli Storage Manager (TSM)
  - Storage Resource Broker (SRB)
- Complex search mechanism to find data



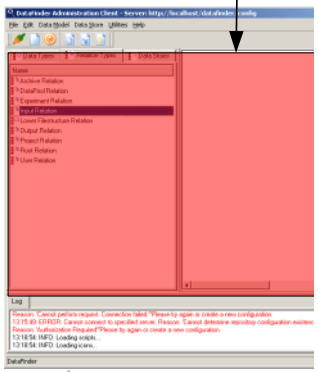
DataFinder Concepts: Aggregation



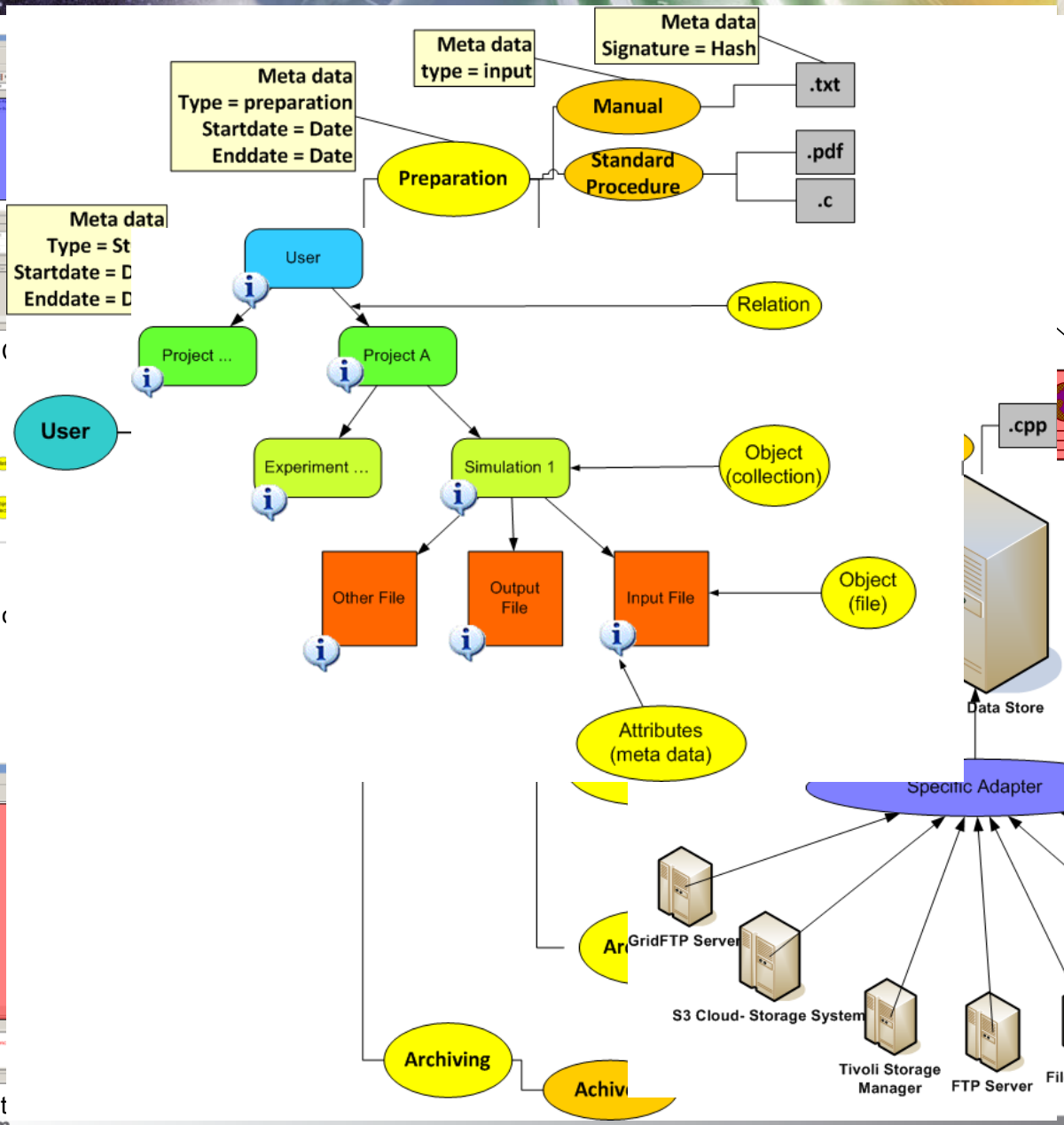


DataFinder User

Logical View of Datamodel



DataFinder Adminstrat



python powered

DataFinder: Concepts
Usage Workflow

Requirements Analysis

- Analyze data, working environment and users workflows

Configuration

- Define and configure data model
- Configure distributed storage resources (Data Stores)

Customization

- Write functional extensions with Python scripts

Rest Interfaces
General Provenance storing information:

- localhost:9999/rest/provenance?
  - **@param** process String having the information about the process and its general type e.g. "process"
  - **@param** input String symbolizing all inputs e.g. "InputType~input1@29;InputType~input2@30;InputType~input2@40"
  - **@param** ouput String symbolizing the output e.g. "OutputType~output21@29"
  - **@param** actor String symbolizing the actor e.g. "ActorType~actor1;ActorType~actor2"

- OutputType/ InputType = Type defined in the model e.g: Manual
- ~ @ ; → Delimiter
- Input1 = identifier of the object
- 29 : Version of the object (e.g. differing modification dates for one object)

Rest Interface
to query the Database

- localhost:9999/rest/gremlinquery
  - **@param** query gremlin query for a graphdatabase

- Gremlinquery
  - Getting all nodes in the database with the identifier „dataItem“ and returning the version of them

  - *$item := g:key($_g, 'IDENTIFIER', '' +* dataItem + *'')*
  - *$item/@version*