DataFinder

FrOSCon 2010

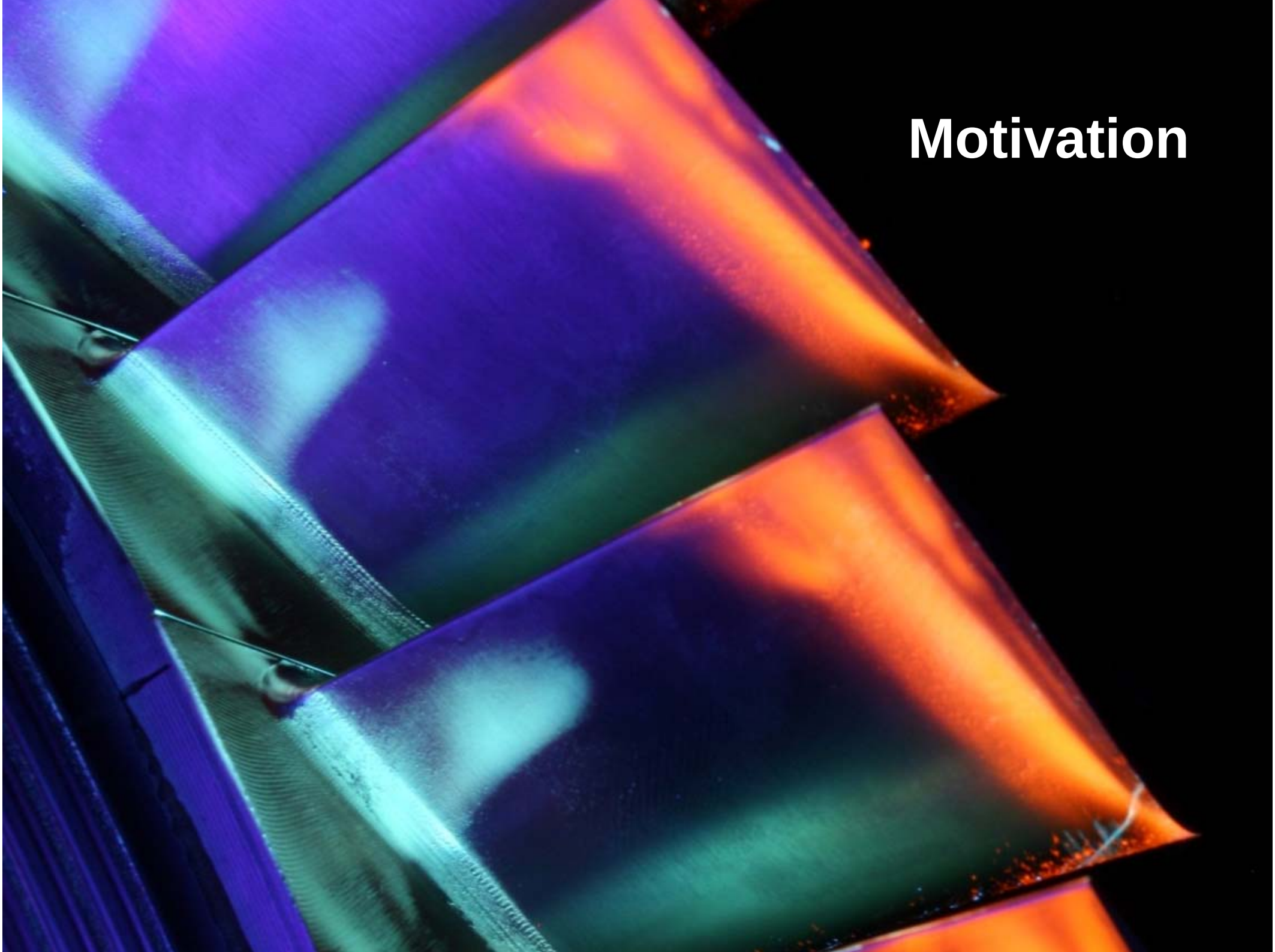
Miriam Ney <*Miriam.Ney@dlr.de*>

Deutsches Zentrum für Luft- und Raumfahrt e.V., Berlin/Köln/Braunschweig

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

Überblick

- Motivation
- Konzepte
- Prototyp und erste Version
- Aktuelle Version (DataFinder 2.0)
- Open Source Verfügbarkeit
- Anwendungsbeispiele



Motivation

Motivation

Hintergrund: Data Management Problem

Fehlende organisatorische Strukturen

- Keine zentrale Richtlinie zur Datenlagerung
- Jeder Mitarbeiter verwaltet die Daten einzeln
  - → Forscher verwenden 30% ihrer Zeit mit Datensuche
  - → Probleme mit Daten von Zeitmitarbeitern

Zunahme von zu verwaltenden Daten

- Wachstum der erzeugten Daten aus Simulationen und Experimenten
- Gesetzliche Richtlinien erfordern die Langzeitverfügbarkeit von Daten (bis zu 50 Jahre!)

Situation ist ähnlich für jedes DLR Institut,

viele Forschungslabore und die Industrie

Motivation

Suche nach Lösungen für das Datenverwaltungsproblem

Definition eines „Standard Problems“ zur Evaluation

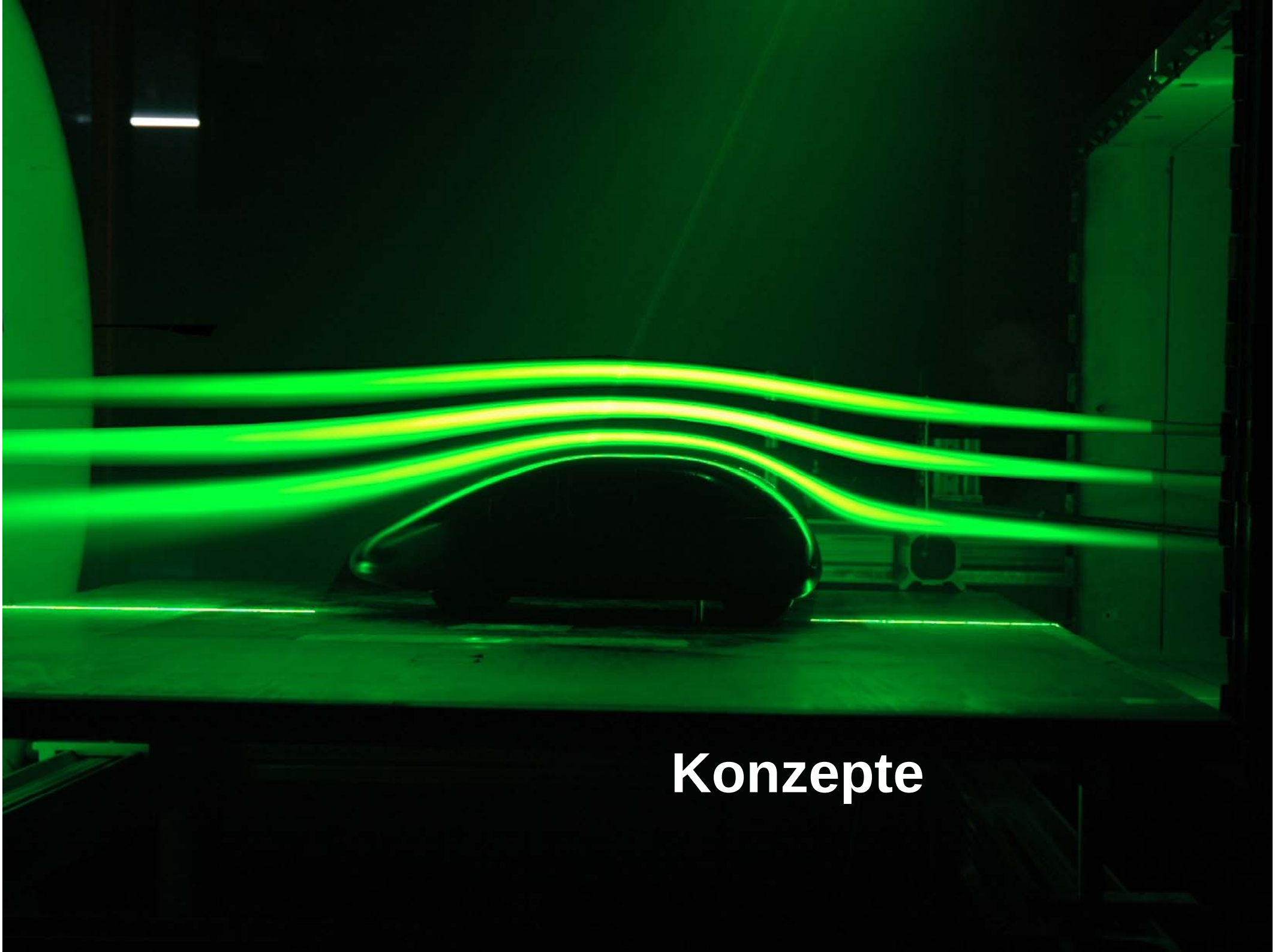
- Aerodynamische Simulation von Helikopter

Bewertung von kommerziellen Product-Data-Management-Systemen (PDM)

- Kostenintensiv
- Überflüssige Funktionen
- Selbst definierte oder unverständliche Skriptsprachen

Ziele bei der Entwicklung des DataFinder

- **Leichtgewichtige** Datenverwaltungsanwendung für existierende Server-Umgebungen
- **Gerade genügend Funktionalität** für unsere Probleme


Konzepte

Konzepte

Ansätze zum Verwalten von großen Datensets

- **Datenstrukturierung:** Meta- Informationen und Datenmodelle
- **Flexible Nutzung von Speichermedien:** Data Stores
- **Infrastrukturaufbau:** Server-Client-Struktur
- **Umgebungsintegrierung:** Erweiterungen durch Skripte
- **Programmiersprache:** Python

Nützliche Software zum effizienten Verwalten von wissenschaftlichen und technischen Daten

Konzepte
Python in Forschung und Industrie

Beobachtung:

- Wissenschaftler und Ingenieure wollen keine Software entwickeln sondern Probleme lösen
- Code schreiben: so einfach wie möglich!

Warum ist Python perfekt?

- Einfach zu lernen und zu benutzen

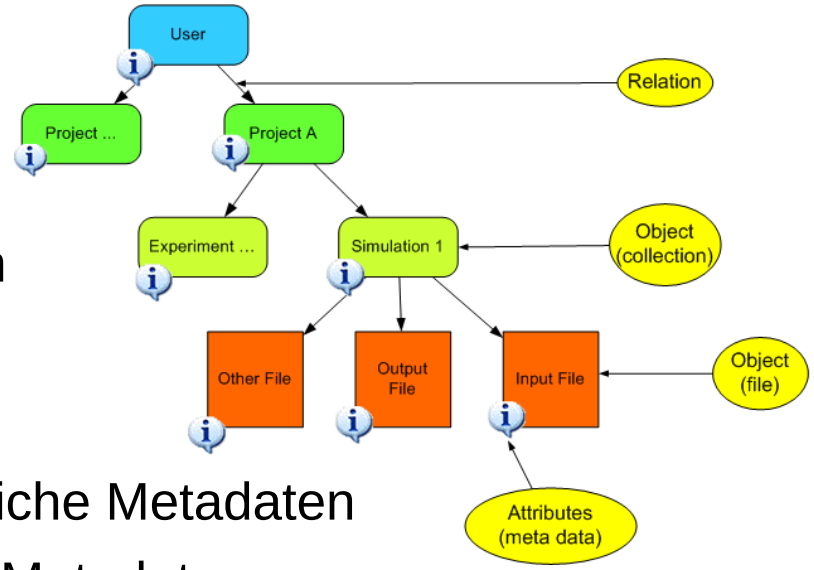
  (= *steile Lernkurve*)
- Ermöglicht schnelle Entwicklungen

  (= *kurze Entwicklungszeiten*)
- ***Inhärente sehr gute Wartbarkeit***

"Ich will Flugzeuge entwerfen und nicht Software!"

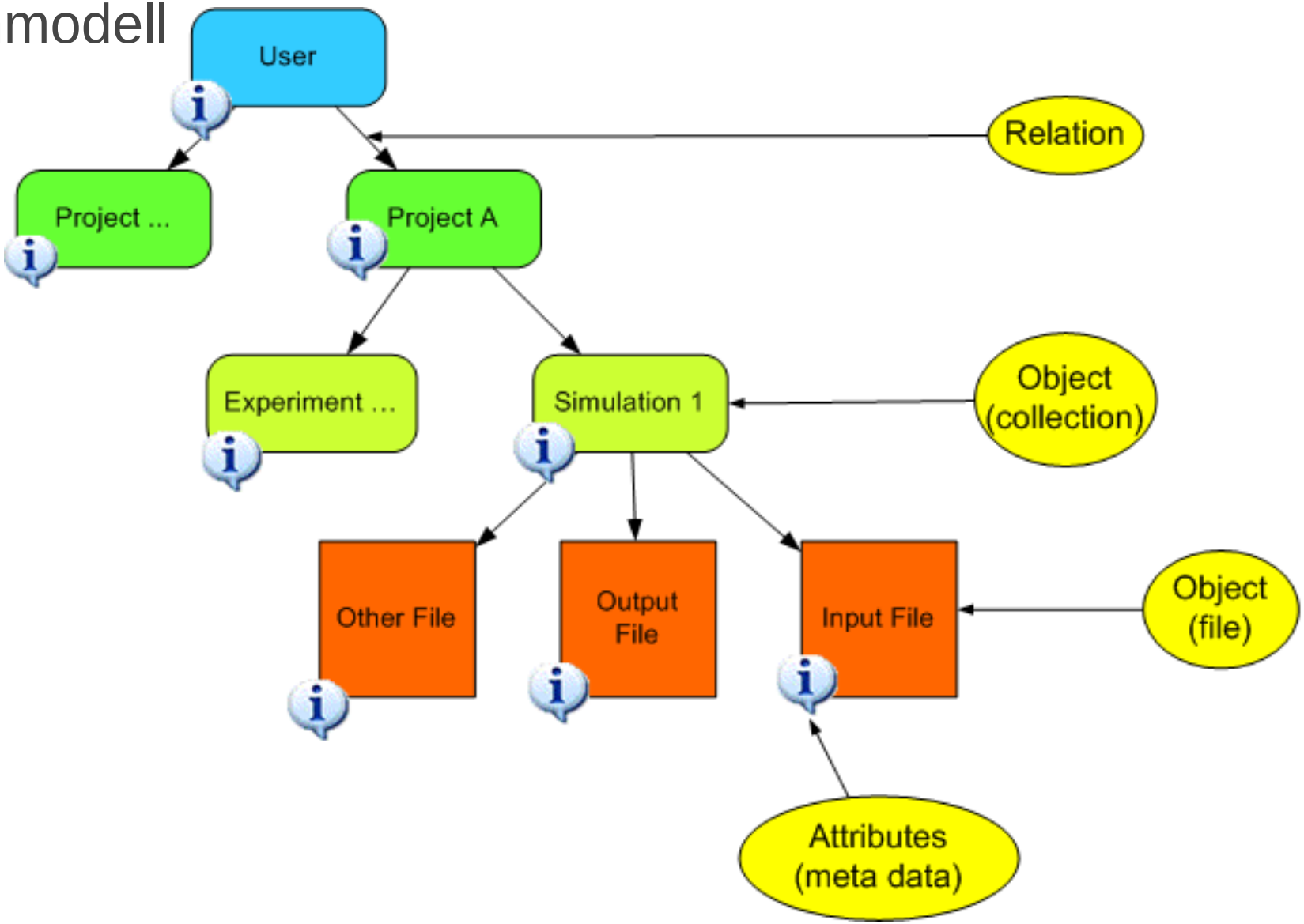
Konzepte
Datenmodell: Datenstruktur und Metadaten

- Datenmodell: Definition der Datenstruktur und der Metadaten
- Speicherung als XML
- Anwender kann in Metadaten suchen
- Verschiedene Level von Metadaten
  - Administrator definiert erforderliche Metadaten
  - Anwender definiert zusätzliche Metadaten
- Verschiedene Datentypen



Konzepte

Datenmodell



Konzepte
Auswirkungen auf den Anwender

DataFinder schränkt die Rechte des Anwenders ein!

- Einhalten von „gutem Benehmen“

Anwender muss sich an organisatorische Standards halten

- Speicherung von Daten in einer bestimmten Hierarchie auf einem Server
- Erforderliche Metadaten müssen vor dem Upload gesetzt werden
- Nutzer hat eingeschränkte Rechte innerhalb einer Hierarchie

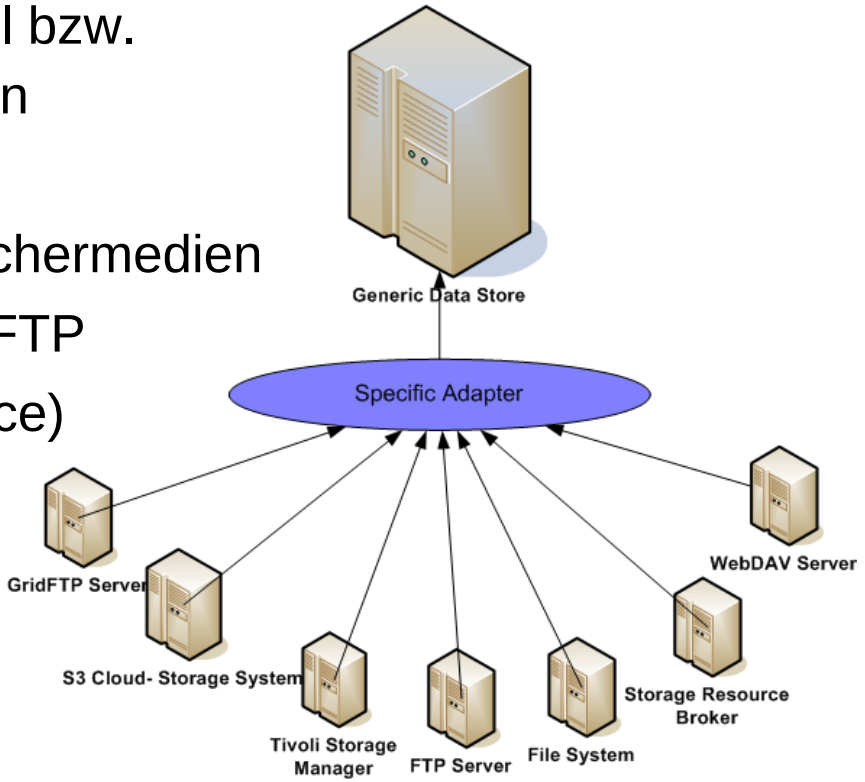
“Mist! Ich bin ein großartiger Wissenschaftler, ich will die Freiheit meine Daten selbst zu organisieren… “

Konzepte
Data Stores

- Trennung der logischen Sicht des Nutzers von der tatsächlichen Server-Struktur

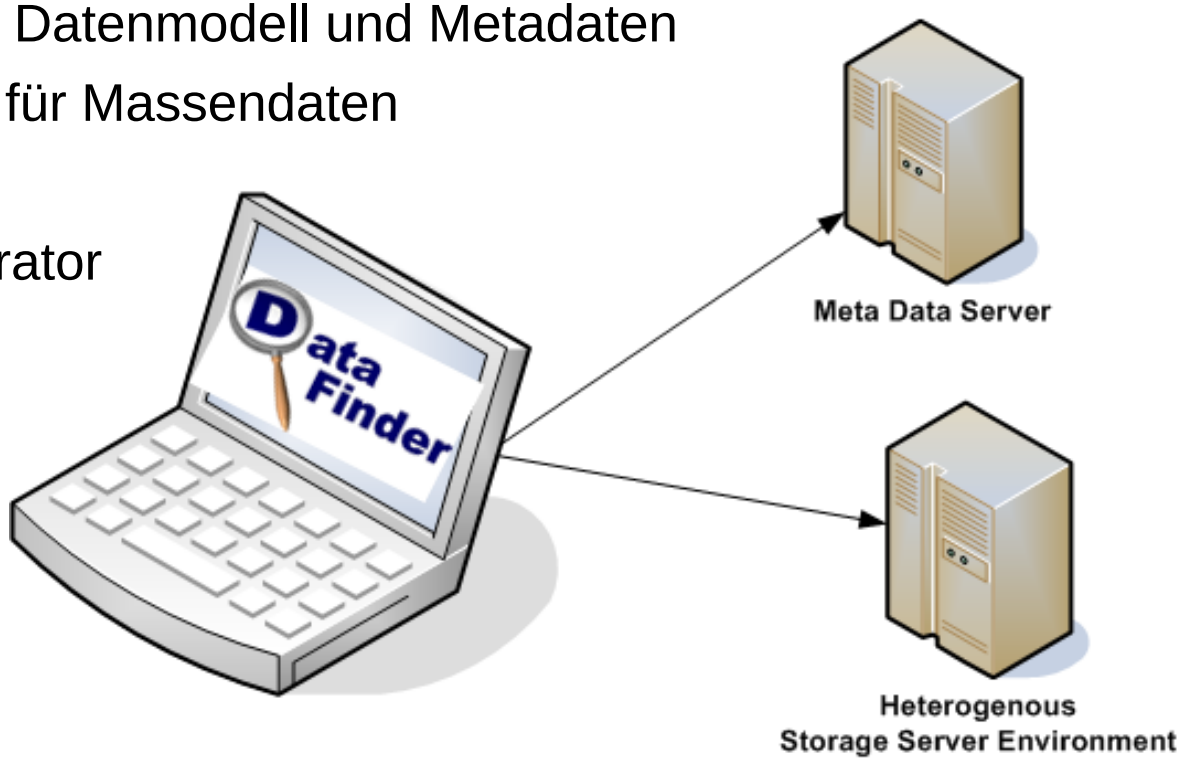
- Getrennte Speicherung von Datenmodell bzw. Metadaten und den tatsächlichen Dateien

- Flexibler Gebrauch von (verteilten) Speichermedien
  - Dateisystem, WebDAV, FTP, GridFTP
  - Amazon S3 (Simple Storage Service)
  - Tivoli Storage Manager (TSM)
  - Storage Resource Broker (SRB)



Konzepte
Verteiltes System

- **Client-Server-Lösung**
- Basierend auf **offenen und stabilen Standards**
- Server:
  - **WebDAV-Server** für Datenmodell und Metadaten
  - **Data Store** Konzept für Massendaten
- Client:
  - Nutzer und Administrator



Konzepte

Python-Skripte zur Erweiterung und Automatisierung

Motivation: Integrierung des DataFinder in die Umgebung

- Nutzer, Infrastruktur, Software, …

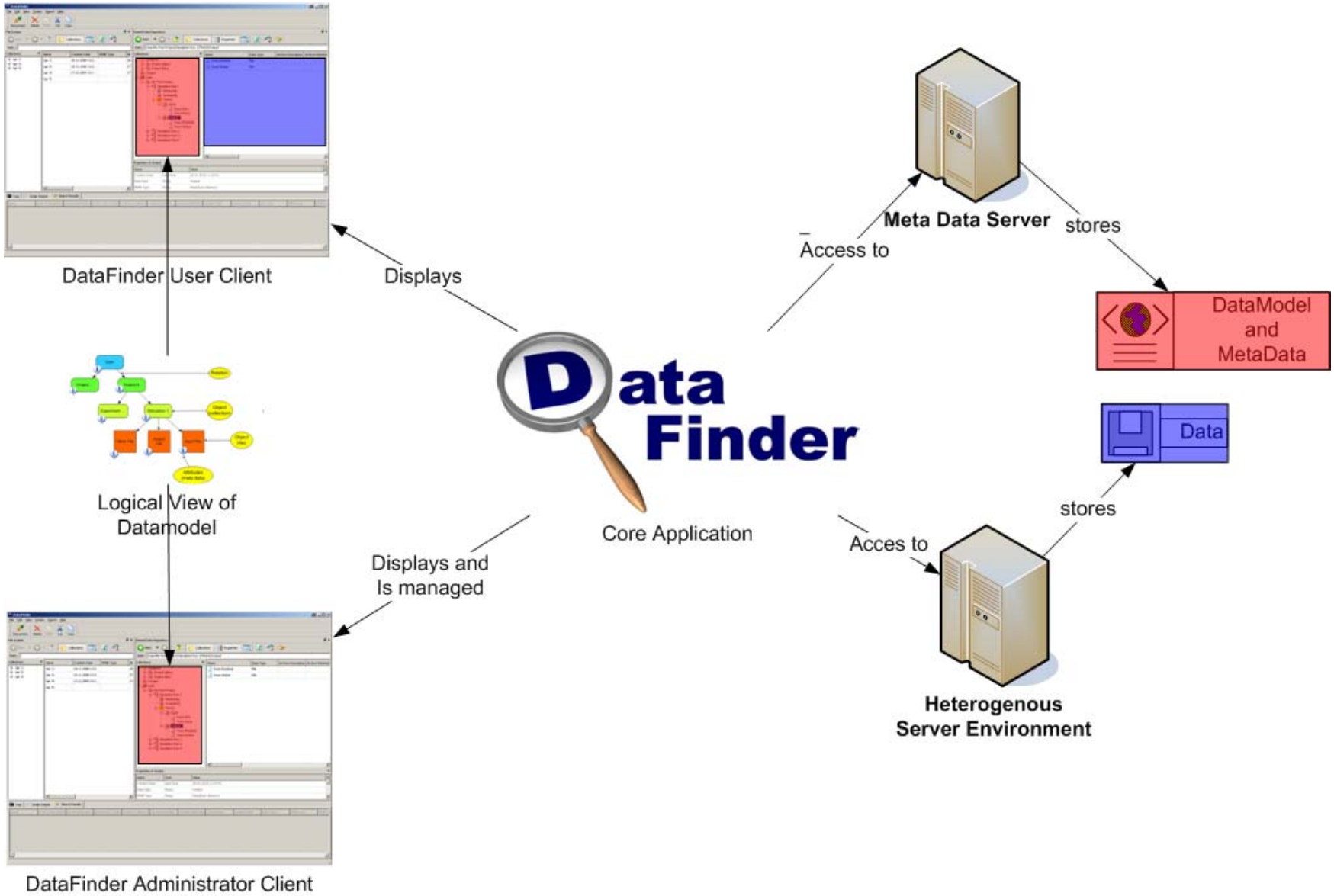
Typische Erweiterungen

- Aktionen für einzelne Ressourcen (wie Dateien oder Ordner)
- Zusätzliche Benutzeroberflächen

Typische Automatisierungen und Anpassungen

- Migration und Import von Daten
- Starten von externen Anwendungen
- Auslesen von Metadaten aus Ergebnissen
- Automatisierung von sich wiederholenden Aufgaben

Konzepte: Aggregation



Konzepte

Prozess zur Einführung des DataFinder

Anforderungsanalyse

- Analyse der Daten, der Arbeitsumgebung und der typischen Arbeitsschritte eines Anwenders mit den Daten

Konfiguration

- Definition und Konfiguration des Datenmodells
- Konfiguration von verteilten Speichermedien (Data Stores)

Anpassung

- Schreiben von funktionalen Erweiterungen mit Python-Skripten

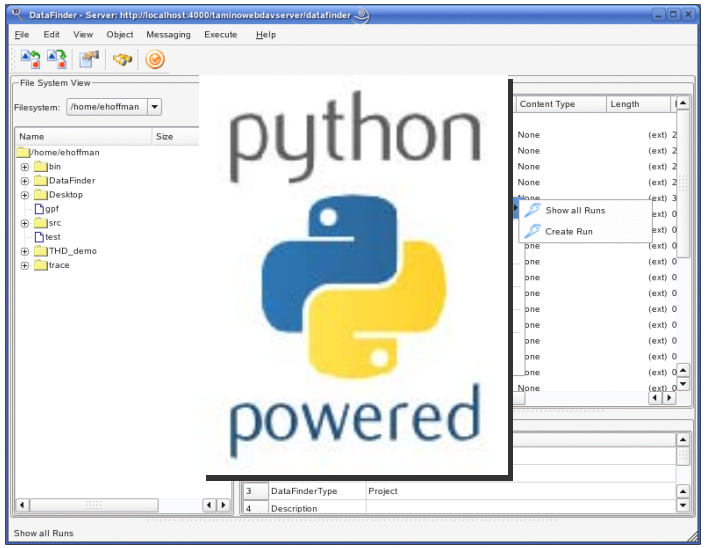
Prototyp und erste Version

Prototyp

Vom Java-Prototyp zum Python-Produkt…

- Entwicklung des Prototypen in Java
- Nutzung von existierendem Code
- Rückschläge: Java Probleme auf wichtigen Plattformen (SGI IRIX)
- Vorteil: Integrierter Jython-Interpreter

- Nutzer: "*Die Java GUI ist scheixxx, aber das Python-Scripting ist toll. Wir wollen eine reine Python-Lösung!*"
- Entwicklung des DataFinder von Grund auf neu mit Python



Erste Version
Realisierung

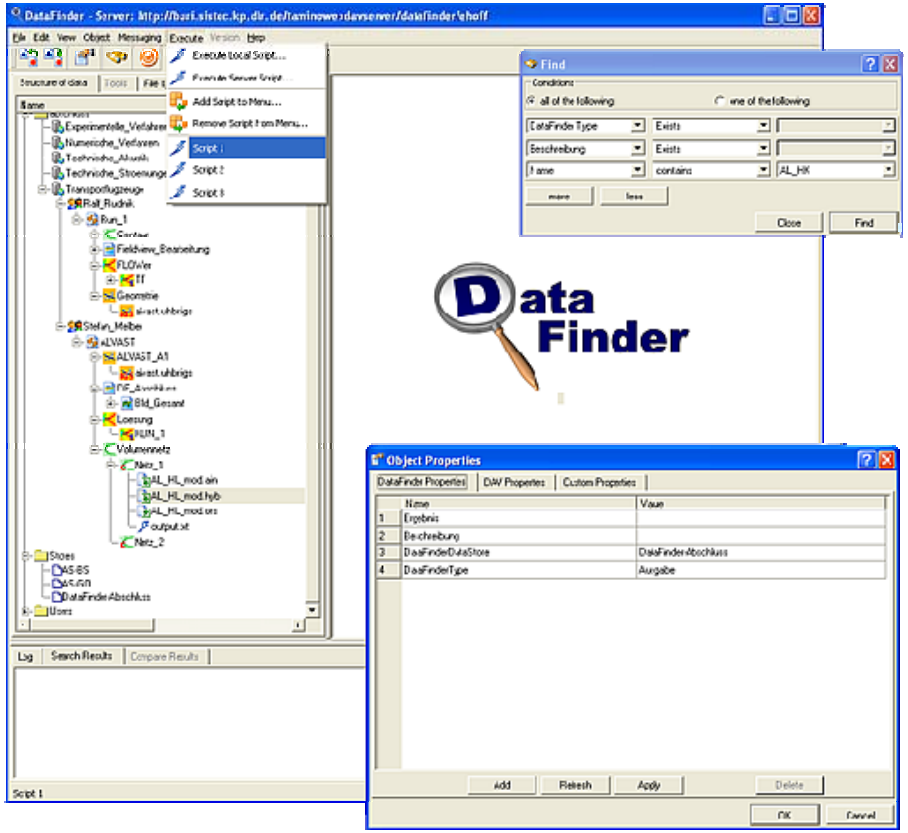
- Komponenten
  - Python 2.5.6
  - Meta Data Server: Tamino XML DB
  - Unterstützte „Data Stores“:
    - FTP
    - WebDAV
    - File system
  - GUI: Qt 3 + PyQt

- Code-Struktur:
  - Kaum vorhanden

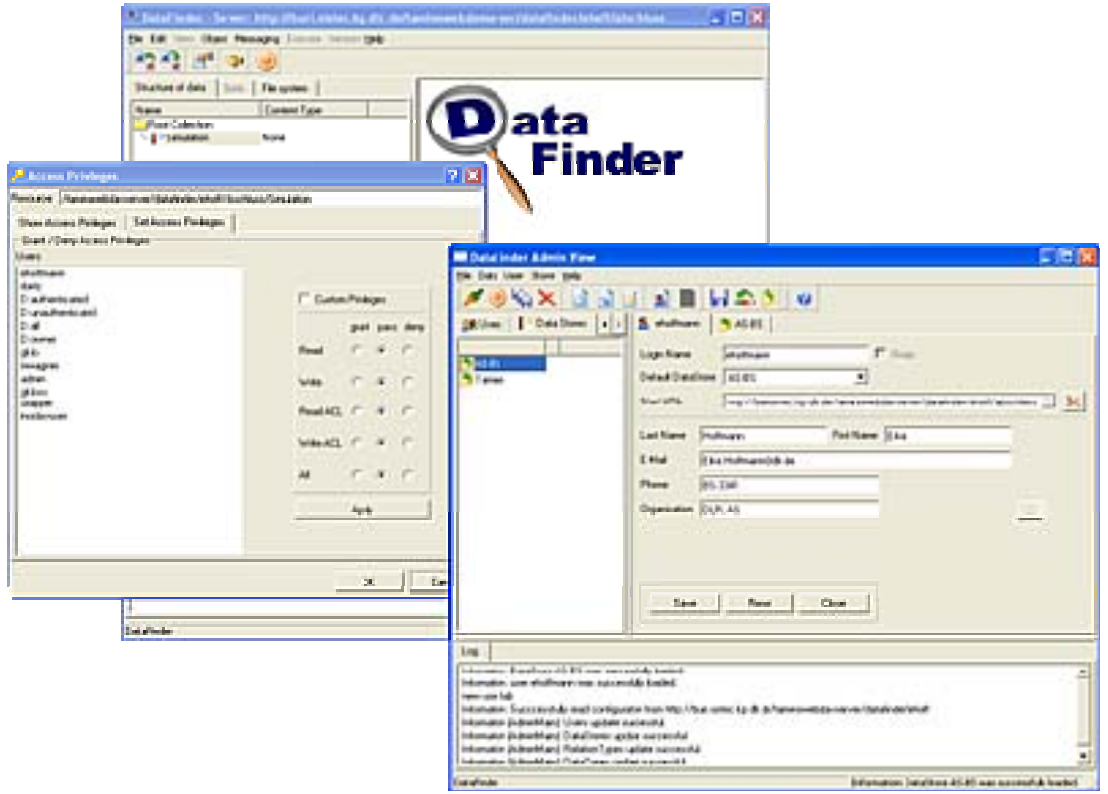
Erste Version

Benutzeroberfläche des DataFinder 1.x

Benutzer Client



Administrator Client




Aktuelle Version
(DataFinder 2.0)

Aktuelle Version (DataFinder 2.0)
Realisierung

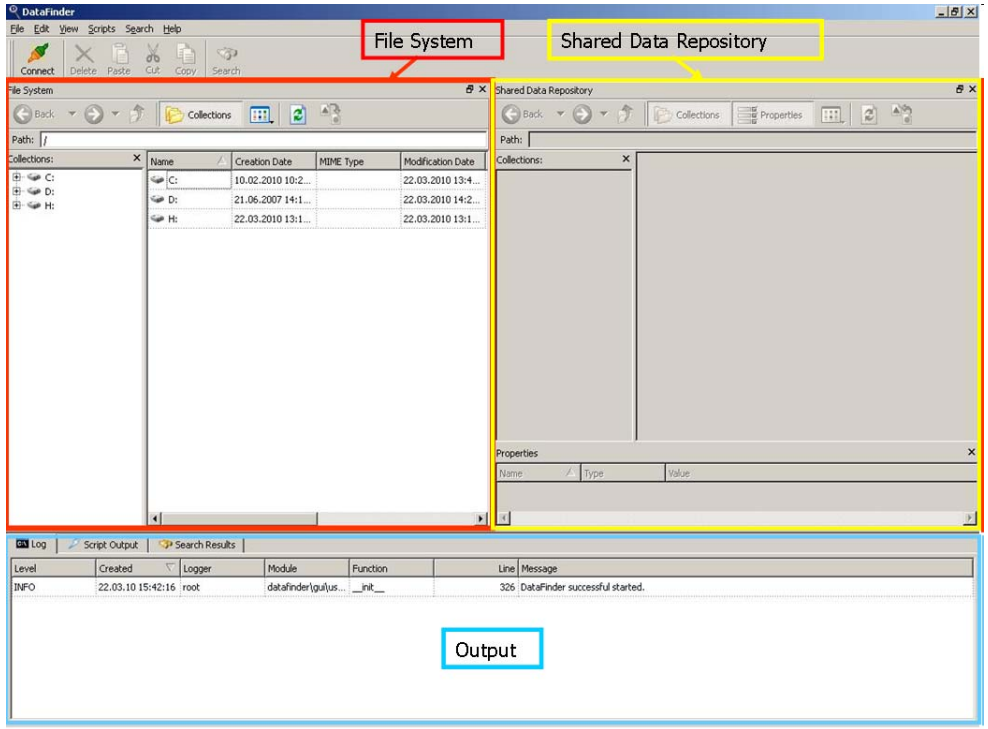
- Komponenten
  - Python 2.6
  - Meta Data Server: Limestone (basierend auf Catacomb)
  - Unterstützte Data Stores:
    - FTP, WebDAV, FS, Amazon, S3, …
  - GUI: Qt 4 + PyQt

- Code-Struktur
  - Schichtenmodell
  - Abstraktion von Konzepten
  - Separate Skript API

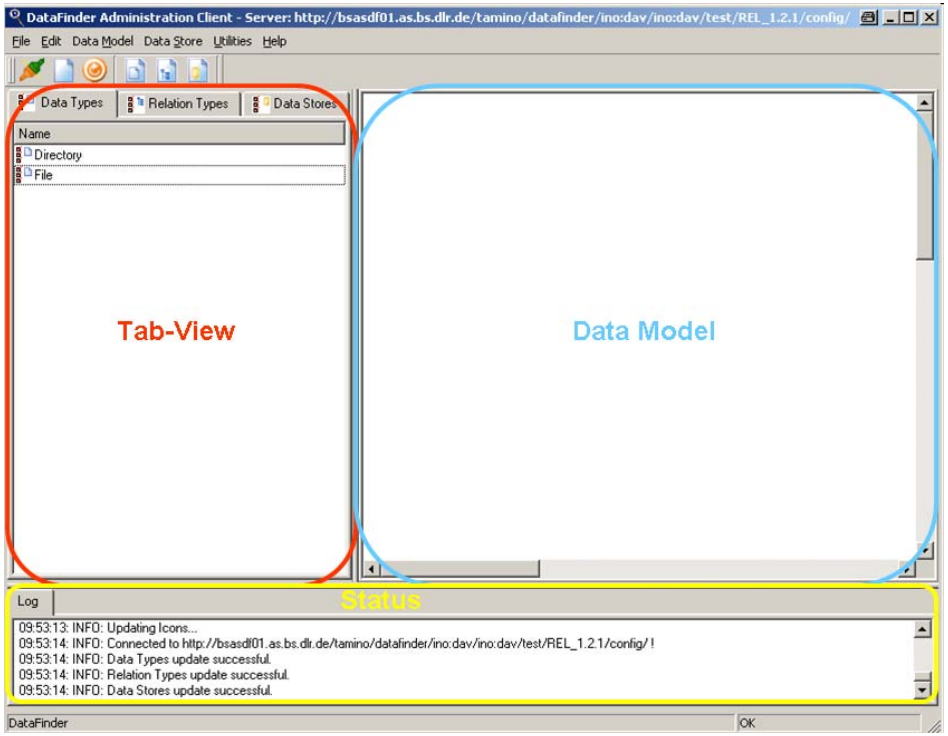
Aktuelle Version (DataFinder 2.0)

Benutzeroberfläche des DataFinder 2.x

Nutzer Client



Administrator Client



Aktuelle Version (DataFinder 2.0)

Skript Beispiel: Erstellen einer Datei

```
# Creating a file "/text.txt" using data store "Data Store".
from datafinder.gui.user import script_api as gui_api
from datafinder.script_api.repository import setWorkingRepository
from datafinder.script_api.item.item_support import createLeaf


# Get representation of the current managed repository
mr = gui_api.managedRepositoryDescription()
# Get currently selected collection in DataFinder Server-View
if not mr is None:
setWorkingRepository(mr)
    def _createLeaf():
        properties = dict()
        properties["____dataformat____"] = "TEXT"
        properties["____datastorename____"] = "Data Store"
        …
       createLeaf("/test.txt", properties)
script_api.performWithProgressDialog(_createLeaf)
```


Live Demo

Open-Source-Verfügbarkeit

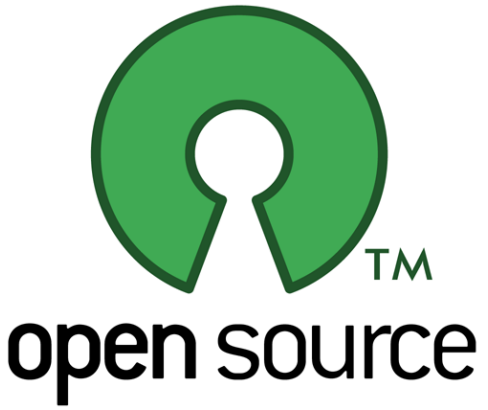
Open-Source-Verfügbarkeit
Hintergrund

- Entwicklung mit Open-Source-Komponenten
  - Python
  - Qt
  - …
- Werkzeug soll allen Interessenten frei zur Verfügung stehen
  - Keine kommerziellen Interessen des DLR
- Einflussnahme von externen Nutzern auf die Entwicklung

Projekt-Hosting und Weiterentwicklung auf „Launchpad“

Open-Source-Verfügbarkeit
Orte

- Aktuelles stabiles release: DataFinder 2.0
- Simplified BSD License
- Websites
  - **Launchpad (Code)**
  - **Sourceforge (Binaries)**
  - **Freshmeat (Ankündigungen)**



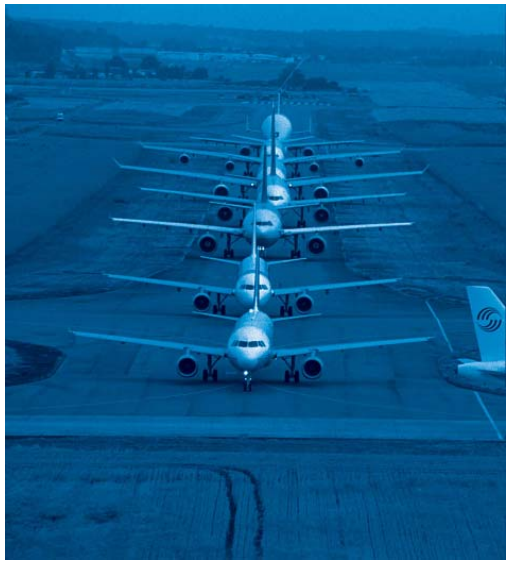
Anwendungsbeispiele

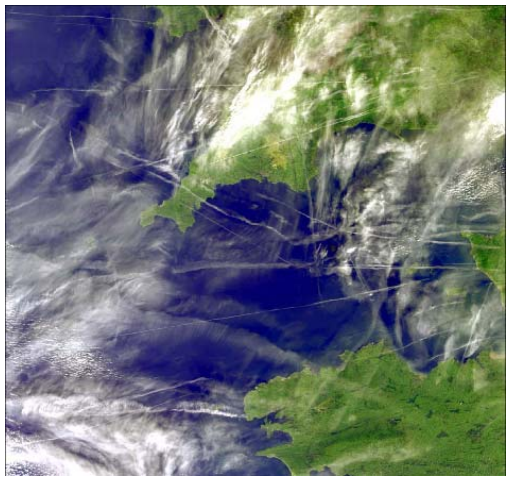
Anwendungsbeispiele
Im Allgemeinen

- Anpassen an aktuellere Bibliotheks-Versionen
  - PyQt
- Unterstützung zusätzlicher Python-Interpretern
  - Jython
  - Python 3.x
- Innerhalb des DLR
  - DataFinder-Web-Portal für Luftverkehrs-Projekte
    - Web-Framework Liferay
  - Provenance-Integration (Projekt AeroGrid)
- Nutzergruppen und Entwickler außerhalb des DLR
  - Max Planck Gesellschaft
  - BeSTGrid (Neuseeland)

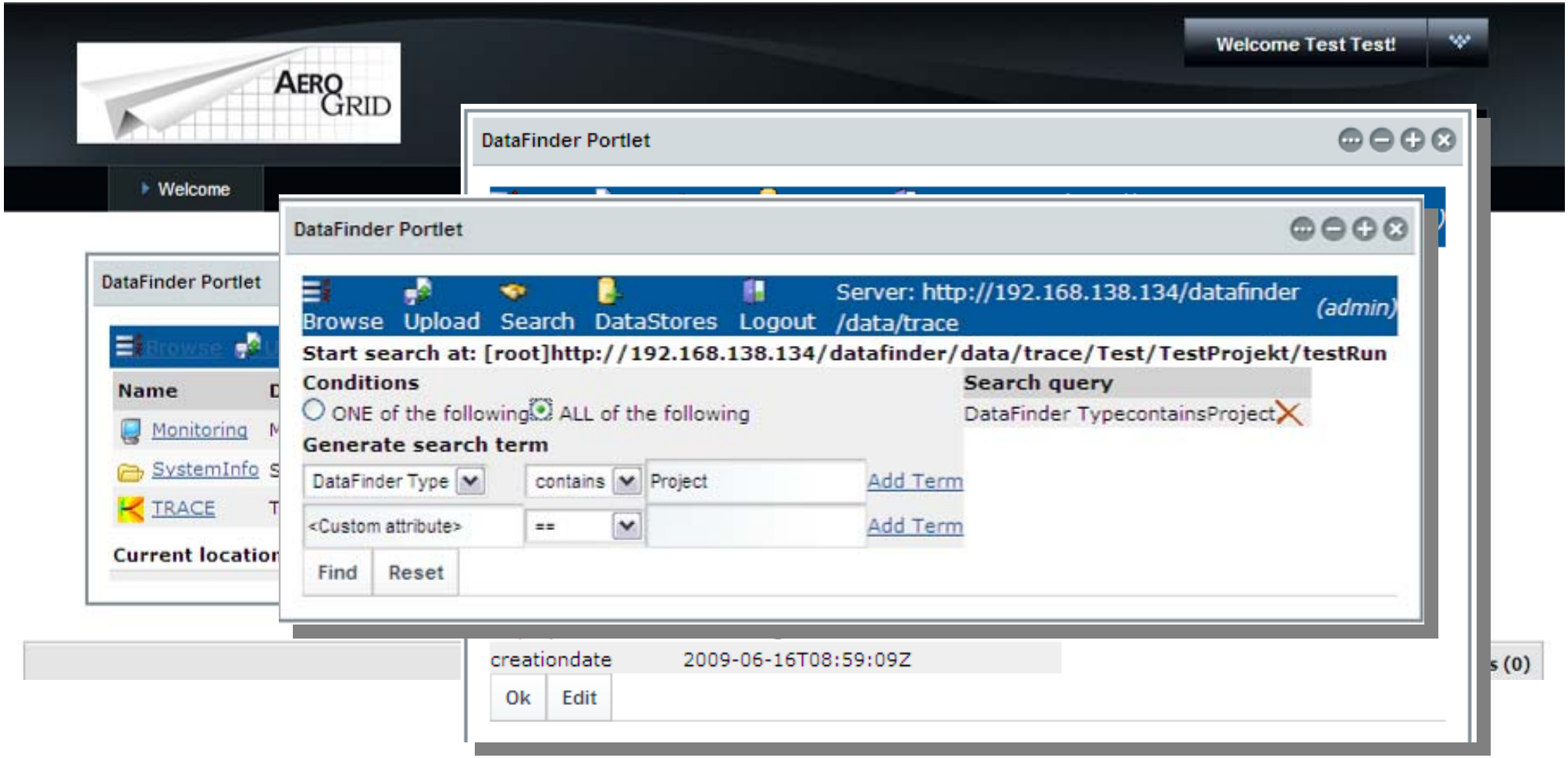
Anwendungsbeispiele
Datenbank zur Flugverkehrsbeobachtung

- Flugverkehrsbeobachtung ist wichtig für die Forschung
  - Vorhersage von Flugverkehr
  - Neue Verkehrsmanagementansätze
- Anwendung des DataFinder
  - Datenbank für Verkehrsdaten und Berichte
  - Projektorientierte Sicht auf Daten





Anwendungsbeispiele
Web Portal



Information

Links

DataFinder Webseite
http://www.dlr.de/datafinder

DataFinder Projektseiten
- **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 | Sl

Werde DataFinder-Fan bei Facebook!

Was machst du gerade?

Anhängen:

DataFinder + andere | DataFinder | Nur Andere | Einstellungen

Seite bearbeiten
Mit Werbenanzeige 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

DataFinder DataFinder-Vortrag auf der FrOSCon 2010

froscon2010: DataFinder
programm.froscon.org
Der DataFinder ist eine in Python entwickelte Open Source Software zur Datenverwaltung. 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

Facebook-Seiten helfen dir dabei, neue Künstler, Unternehmen und Marken zu entdecken. Du kannst dich zudem mit denen vernetzen, die du bereits magst.

Weitere Werbeanzeigen


Fragen?
Kontakt:
Miriam Ney
DLR Simulations- und
Softwaretechnik, Berlin
Email: Miriam.Ney@dlr.de