

# Workflows in MediGRID und BauVOGrid

---

Andreas Hoheisel



**Fraunhofer** Institut  
Rechnerarchitektur  
und Softwaretechnik



# Übersicht

---

- Einführung:  
Definition „Grid-Workflow“,  
Grid Workflow Execution Service
- Abstrakte Modellierung von Workflows mit High-Level-Petrinetzen
- Abbildung auf ausführbare Workflows
- Fehlertolerante Ausführung von Workflows
- Anwendungen in BauVOGrid und MediGRID
- Demonstration



# Grid-Workflows

---

## = Automatisierung von IT-Prozessabläufen in einer Grid-Umgebung

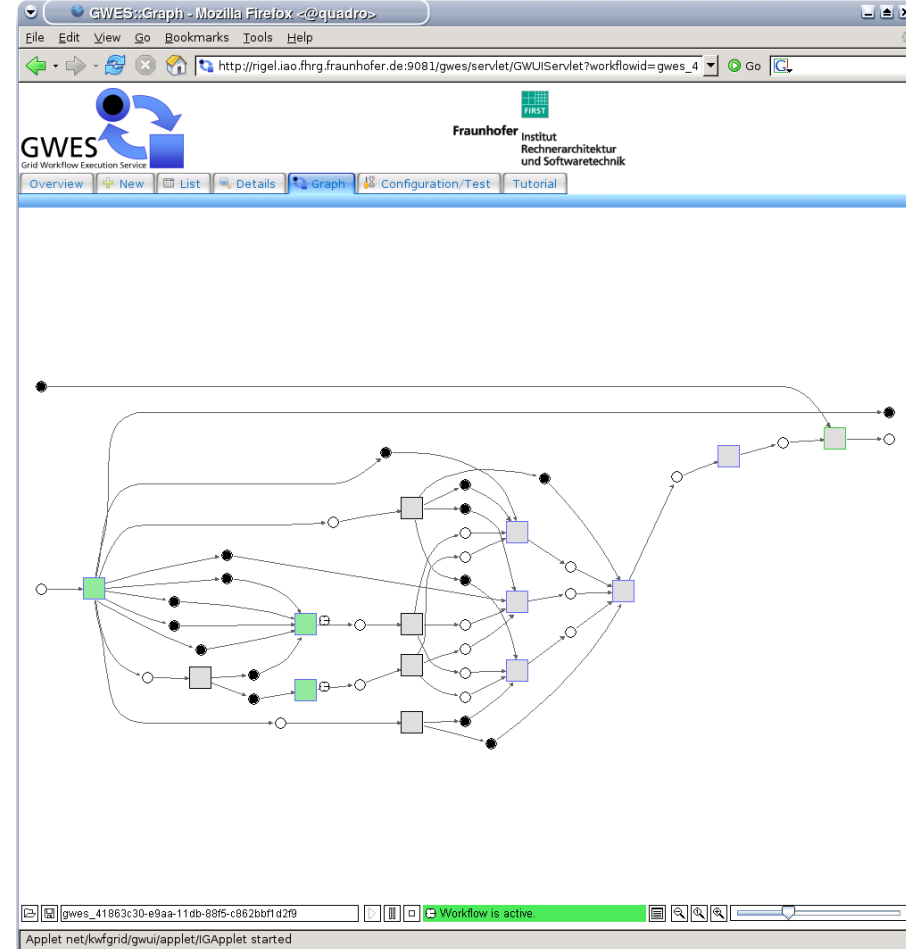
### State-of-the-Art

- Bislang keine einheitliche Grid-Middleware
- Kein einheitlicher und akzeptierter Standard für die formale Beschreibung von Workflows von Grid-Anwendungen
- Keine ausreichende Berücksichtigung der Besonderheiten im Grid
- Choreographie, Orchestrierung und Ausführung von Grid-Anwendungen meist sequentielle Prozessschritte ohne Interaktion
- Die meisten Ansätze sind nicht für den Endanwender mit geringen Informatikkenntnissen geeignet (nicht intuitiv, keine abstrakte Modellierung)



# Grid Workflow Execution Service (GWES)

- Automatisierung von IT-Prozessen
- Interaktion mit dem Nutzer
- Ressourcenabstrahierung mit automatischer und dynamischer Abbildung auf jeweils geeignete Hard- und Softwareressourcen
- Workflow-Aktivitäten: GT4-Jobs + Webservice-Methodenaufrufe + Transfer von Daten + ...
- Fehlertoleranz, Persistenz
- Anwendungsbereiche: Medizinforschung, Verkehrsmanagement, Flutvorhersage, Risikoanalyse, Warenwirtschaft, Bauindustrie, ...



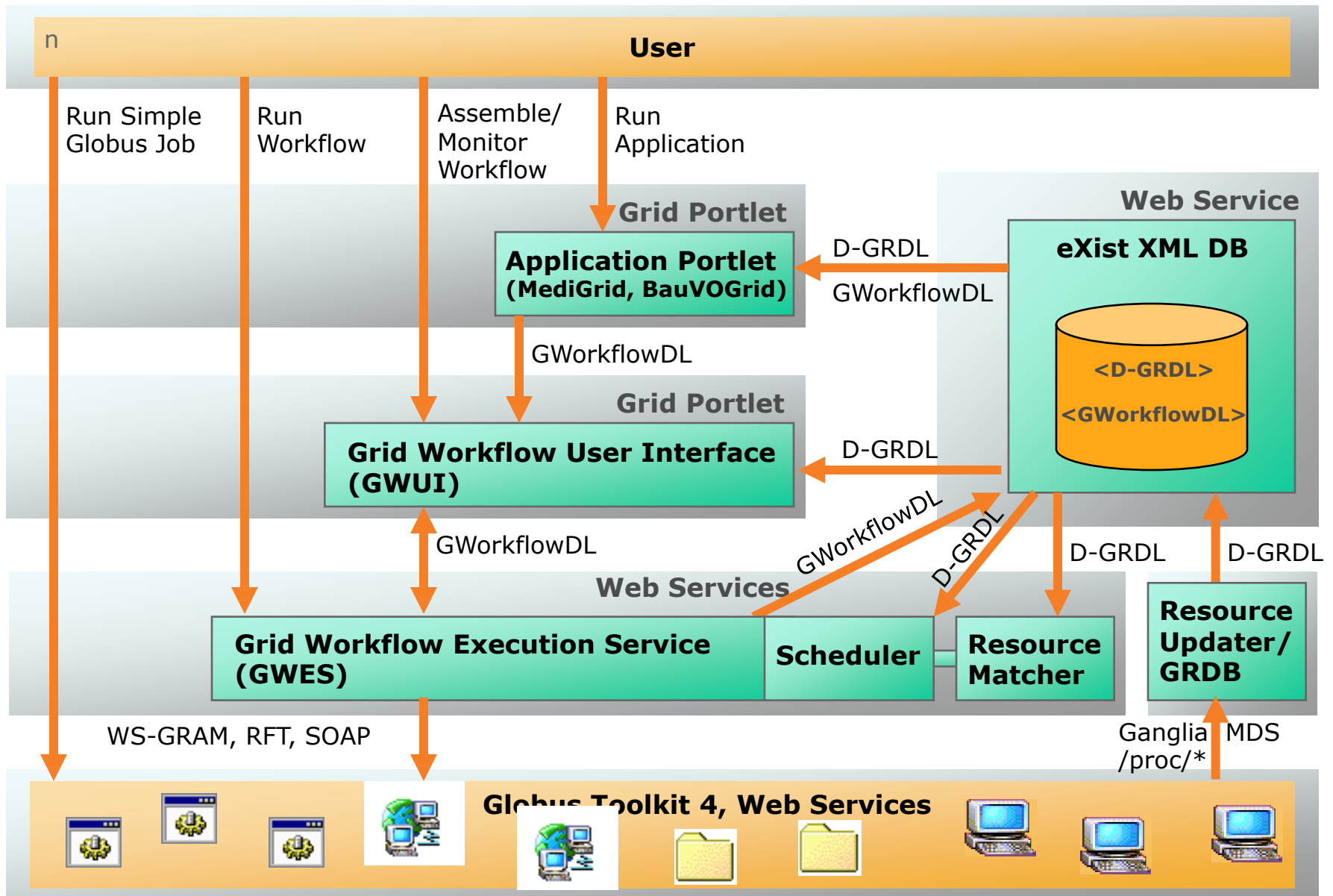
# Geschichte des „Grid Workflow Execution Service“

---

- **Fraunhofer Resource Grid** **Seit 2001**: Erster Prototyp einer Petri-Netz-basierten Workflow-Engine auf Basis von Globus Toolkit 2.4
- **K-Wf Grid (EU)** **2004–2007**: Automatische Erstellung und Ausführung von Workflows, Redesign und Portierung des GWES auf Web-Services + Globus Toolkit 4
- **CoreGrid (EU)** **2004–2008**: Standardisierung der GWorkflowDL, Interoperabilität mit anderen Systemen (z.B. LCG, gLite)
- **Instant-Grid (BMBF)** **2005–2007**: Betriebssystem + Grid-Middleware + Workflow-Management + Anwendungen = Knoppix-CD
- **Enterprise Grids (Fraunhofer)** **2005–2008**: Anpassung für industriellen Einsatz
- **MediGrid (D-Grid)** **2005–2008**: Workflows für die medizinische Forschung, Anpassung an D-Grid, Meta-Scheduling
- **BauVOGrid (D-Grid)** **2007–2010**: Workflows für die Bauindustrie (z.B. Mängelerfassung)



# Architektur Workflow-Management-System



---

# Choreographie:

Abstrakte Modellierung von lose gekoppelten Grid-Anwendungen mittels „High-Level Petrinetzen“



# Grid-Workflow-Beschreibungssprachen

---

## Inhärentes Modell

Workflow wird verteilt innerhalb der Softwarekomponenten definiert  
(z.B. MPI, CORBA, Cactus)

## Externes Modell

Workflow wird oberhalb der Softwarekomponenten definiert  
Globale Sicht des Workflows  
(z.B. skript- oder graphbasiert)

## Skriptsprachen

GridAnt, JPython (XCAT)

## Kombiniert

WSFL, XLANG, BPEL4WS, UNICORE, GSFL

## Netze/Graphen

DAG-basiert: Condor DAGman, Symphony  
**Petrinetzbasiert: GWorkflowDL**

# Warum Petrinetze?

---

## BPEL

Sehr komplexe Semantik  
Keine geeigneten Abstraktionsmechanismen  
Für Validierung nicht formal genug  
Kein „stabiler“ Standard

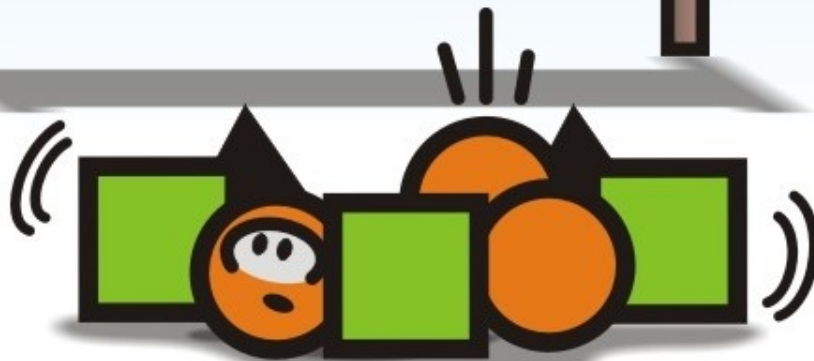
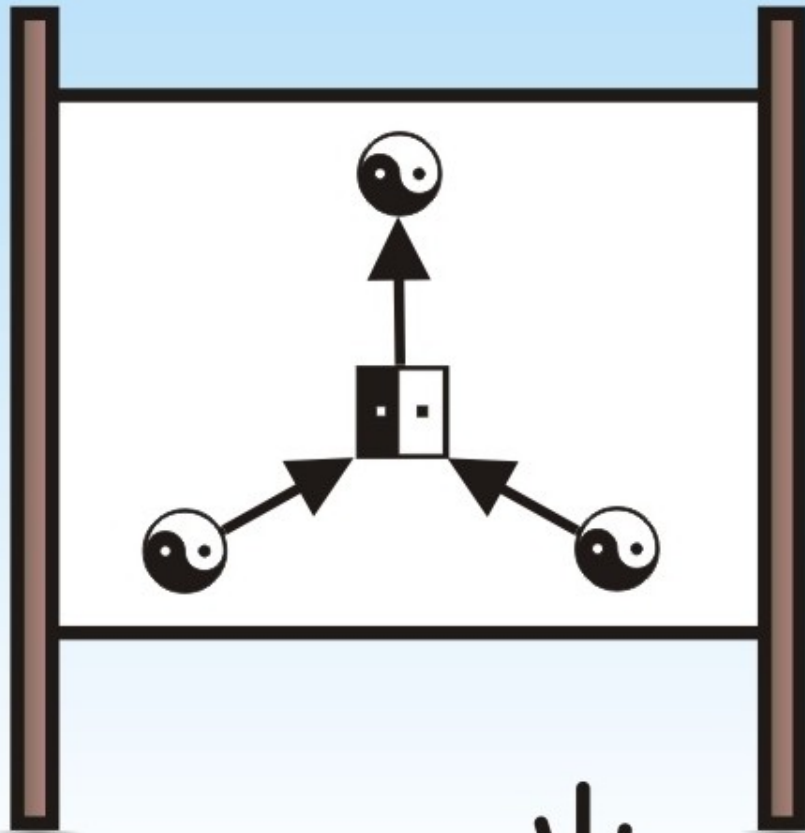
## DAG-basiert

Directed Acyclic Graph  
Zustand und Schleifen werden nicht explizit durch  
Graphen modelliert

## Petrinetzbasiert


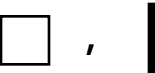
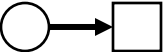
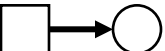

Zustand und Aktionen werden explizit modelliert  
Umfangreiche Theorie verfügbar  
Einfach  
Ausdrucksstark  
Intuitive Visualisierung möglich

Konvertierung von anderen Workflow-Sprachen  
möglich (z.B. BPEL2PN, ARIS-EPKs, ...)

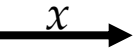
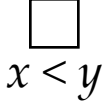


# Konzept – High-Level Petrinetze

---

-  **Stellen**                      Repräsentieren Platzhalter für Daten oder Marken
-  **Transitionen**                      Repräsentieren (abstrakte) Operationen (z.B. Web-Service-Methodenaufruf oder Ausführung eines Programms)
-  **Kanten von Stellen nach Transitionen (Flussrelation)**
-  **Kanten von Transitionen nach Stellen (Flussrelation)**
-  **Kapazität**                      Maximale Anzahl von Marken auf einer Stelle

## Verknüpfung mit realen Daten

- **Unterscheidbare Marken** Beinhalten reale Daten (Parameter, Seiteneffekte) oder stellen einen Zustand dar (z.B. done, failed)
-  **Kantenanschrift**                      Repräsentiert Variablennamen der Operation (z.B. SOAP message parts) oder XPath-Ausdrücke
-  **Bedingung**                      Aktivierte Transitionen schalten nur, wenn alle Bedingungen wahr sind (XPath)

# Terminologie – High-Level Petrinetze

---

## aktiviert (*enabled*)

Eine Transition heißt „aktiviert“ (*enabled*) wenn:

- mindestens eine Marke auf allen Eingabestellen vorhanden ist *und*
- keine der Ausgabestellen ihre Kapazität erreicht hat

## schalten (*occur, fire*)

Aktivierte Transitionen können „schalten“ (*fire*) indem sie eine Marke von jeder Eingabestelle nehmen (z.B. Eingabedaten, Vorbedingungen) und eine neue Marke auf jede Ausgabestelle legen (z.B. Ausgabedaten, Seiteneffekte).

## Verfeinerung (*refinement*)

Ein Teil eines Petrinetzes kann durch ein Sub-Petrinetz ersetzt werden

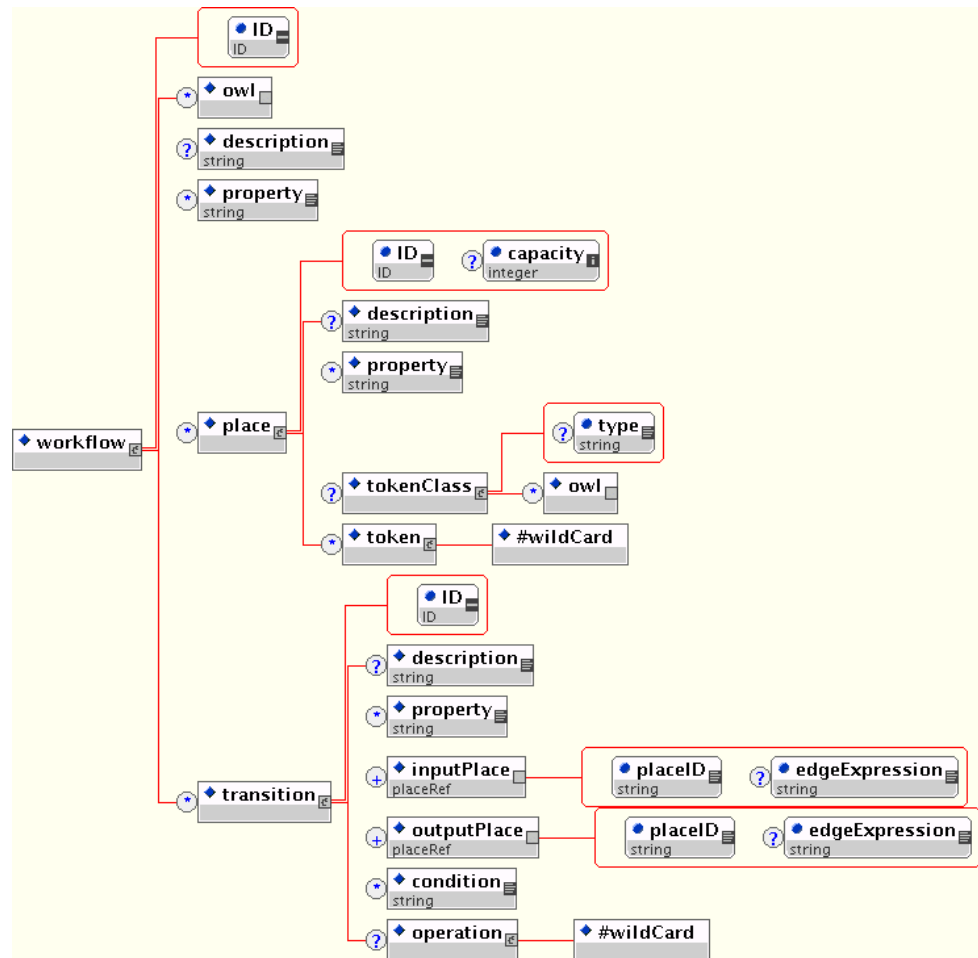
# Grid Workflow Description Language (GWorkflowDL)

## XML

Basiert auf Petrinetzen

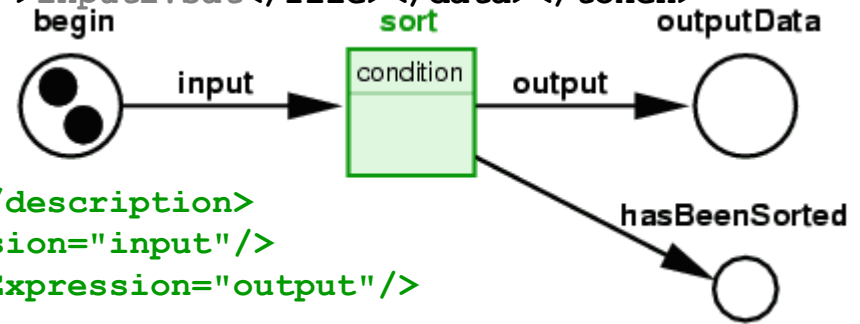
Sprache unabhängig von der Grid-Infrastruktur

Unterstützt verschiedene Abstraktionsstufen in einer Sprache



# GWorkflowDL – Beispiel

```
<workflow xmlns="http://www.gridworkflow.org/gworkflowdl"
  xmlns:pe="http://www.gridworkflow.org/gworkflowdl/programclassexecution"
  ID="sortWorkflow">
  <place ID="begin">
    <token><data><file xsd:type="xsi:anyURI">input1.dat</file></data></token>
    <token><data><file xsd:type="xsi:anyURI">input2.sat</file></data></token>
  </place>
  <place ID="outputData"/>
  <place ID="hasBeenSorted"/>
  <transition ID="sort">
    <description>sorts strings or numbers</description>
    <inputPlace placeID="begin" edgeExpression="input"/>
    <outputPlace placeID="outputData" edgeExpression="output"/>
    <outputPlace placeID="hasBeenSorted"/>
    <condition>string-length($input/token)>0</condition>
    <operation>
      <pe:programClassExecution softwareClass="sort.xml">
        <pe:programExecution software="~/sort.sh" hardware="node15" quality="0.9"/>
        <pe:programExecution software="~/sort.sh" hardware="node20" quality="0.3"/>
      </pe:programmClassExecution>
    </operation>
  </transition>
</workflow>
```



---

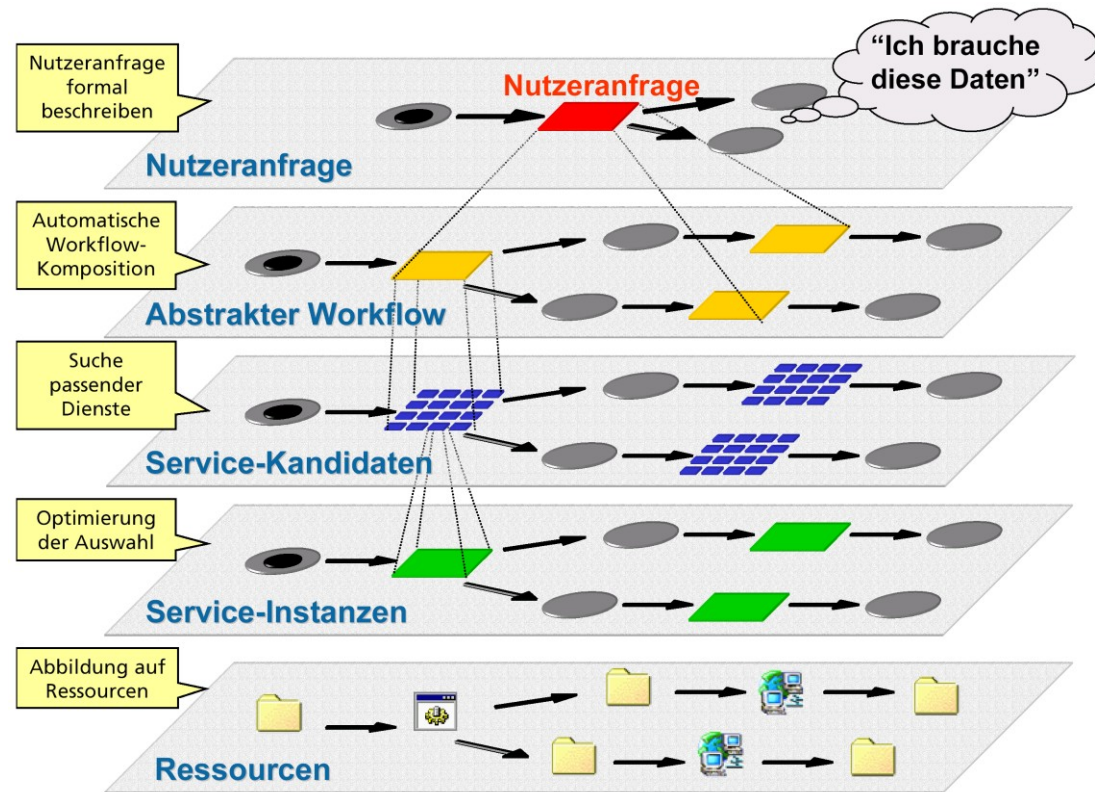
# Orchestrierung:

Abbildung auf konkrete  
(ausführbare) Workflows



# Ressourcenabbildung und Scheduling

- Im Grid können Ressourcen (Software, Hardware, Dienste, Daten) jederzeit wegfallen bzw. fehlschlagen oder auch neu hinzukommen
- Daher sollten möglichst abstrakte Prozessabläufe definiert werden, die dann erst zur Laufzeit auf die aktuell verfügbaren Ressourcen abgebildet werden



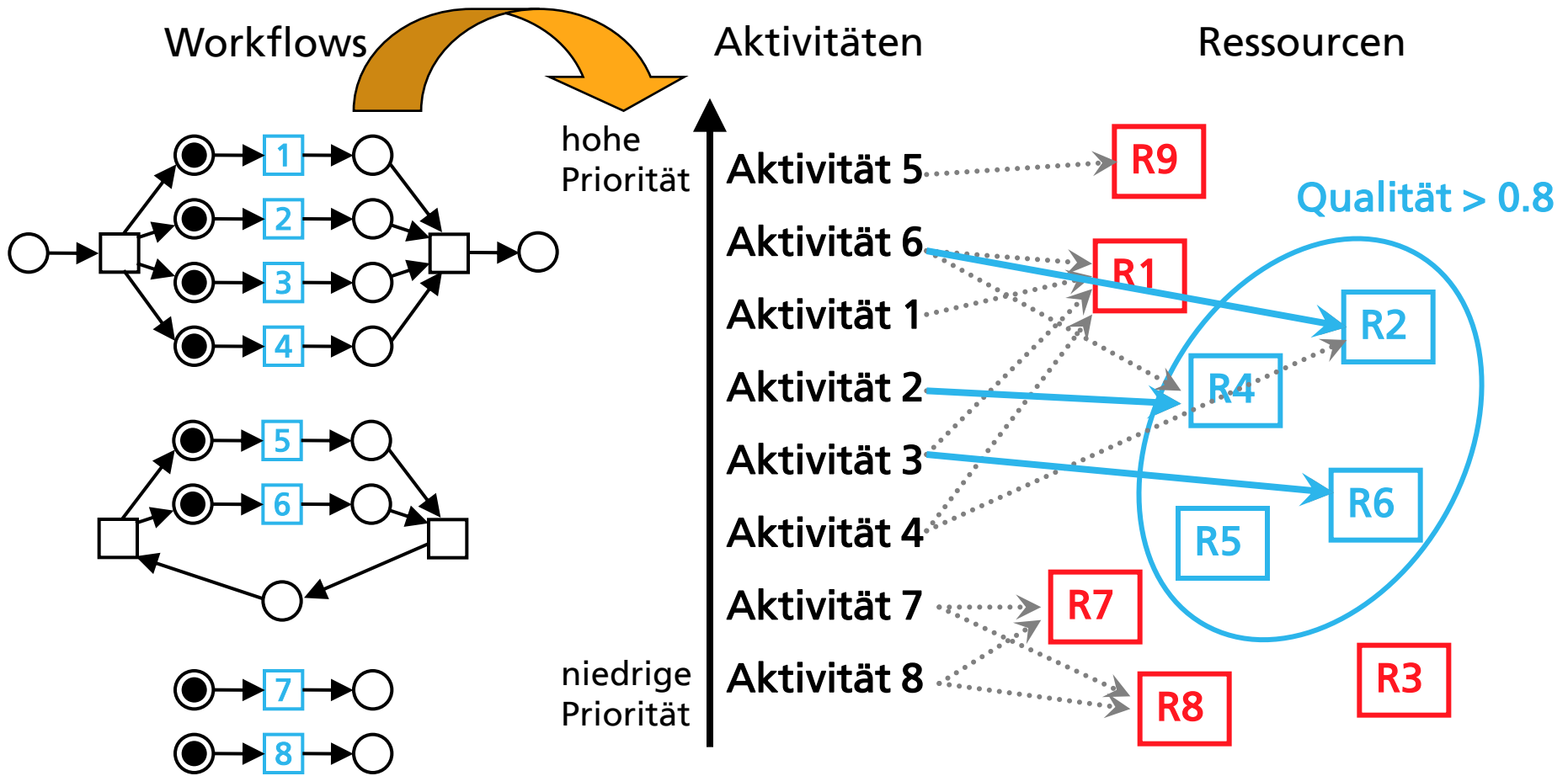
# Workflow-Scheduler (bzw. Resource-Selector)

---

- Ziel: Einfache und robuste Optimierung der Ressourcenauswahl
  - Maximierung der Auslastung verfügbarer Rechenknoten (unabhängig Ihrer Leistungsfähigkeit)
  - Optimierung der Effizienz (= Durchsatz) für das gesamte Grid
  - „Time-Sharing“ erlaubt (bei Fork)
  - In Arbeit: Optimierung der Ausführungsdauer von einzelnen Aktivitäten oder Workflows
  - In Arbeit: „Advanced Reservation“
- Konzept: Alle passenden Ressourcen mit einer „Qualität“ größer als ein Schwellwert werden zufallsverteilt den Workflow-Aktivitäten der Warteschlange zugeordnet, beginnend bei hoher Priorität



# Beispiel

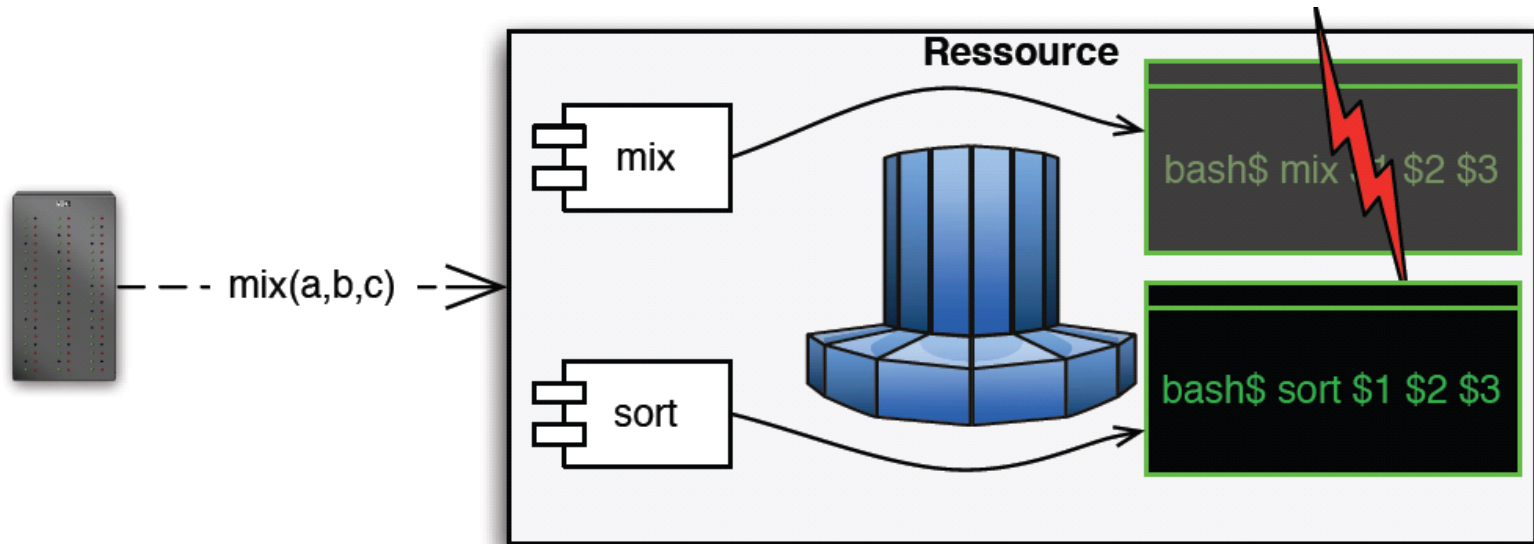


---

# Fehlertolerante Workflows

In Zusammenarbeit mit der Uni Potsdam  
Bettina Schnor, Matthias Schulz

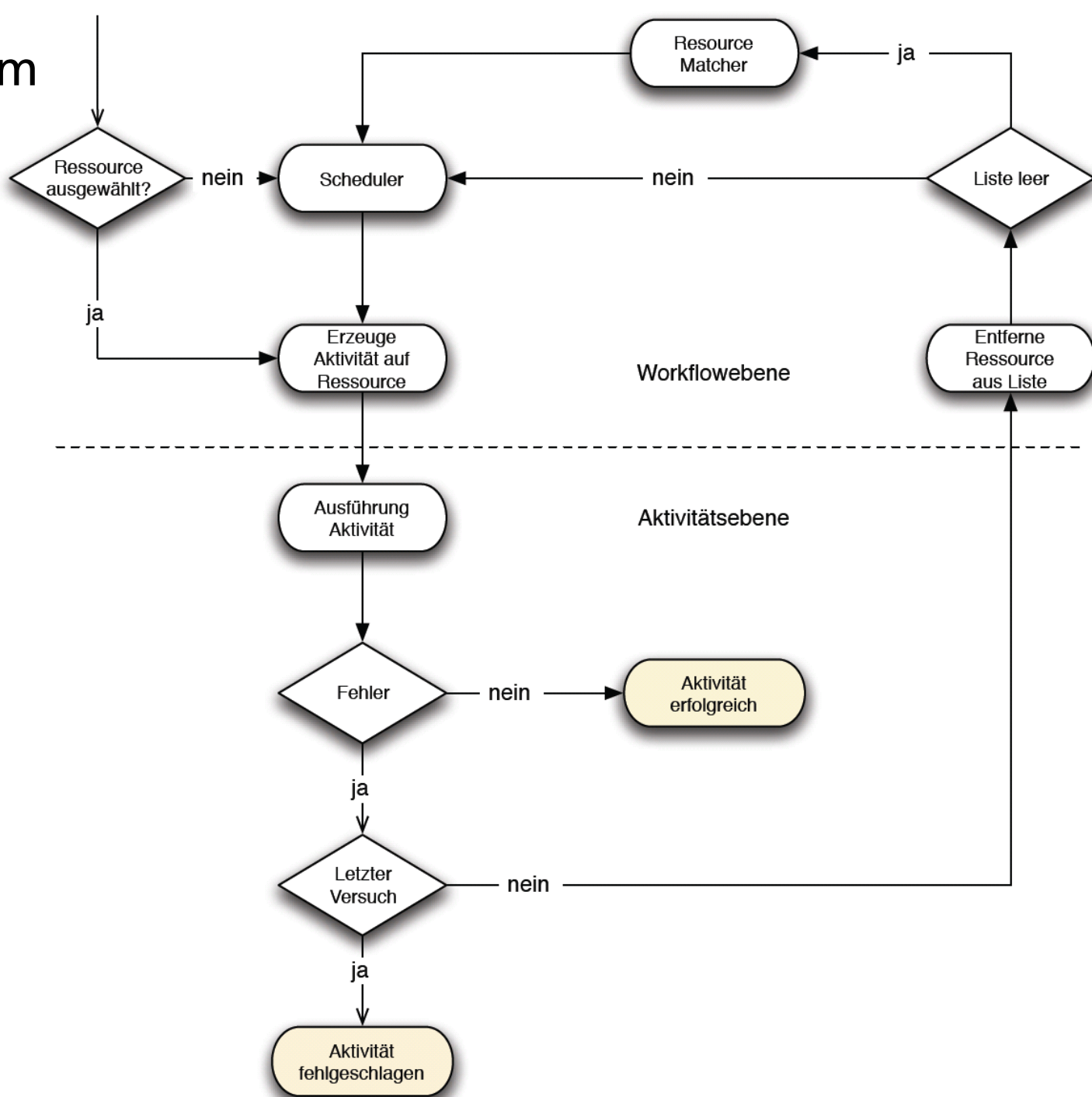
# Beispiel für Interaktionsfehler im Grid



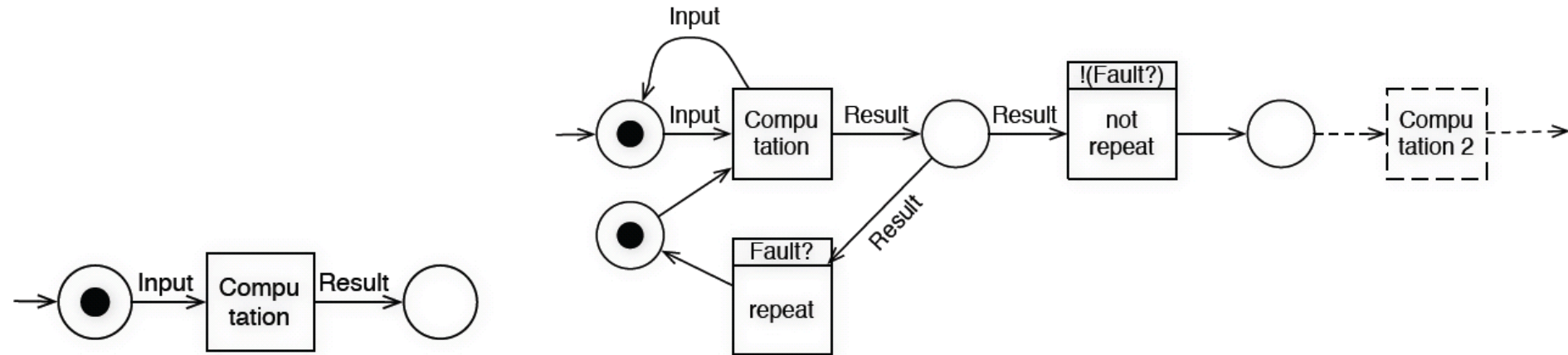
(Quelle: Schulz, 2007)

# Ablaufdiagramm implizite Fehlertoleranz

(Quelle: Schulz, 2007)



# Beispiel für explizit modellierte Fehlertoleranz

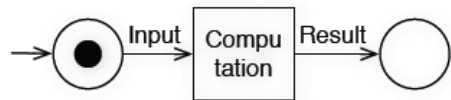
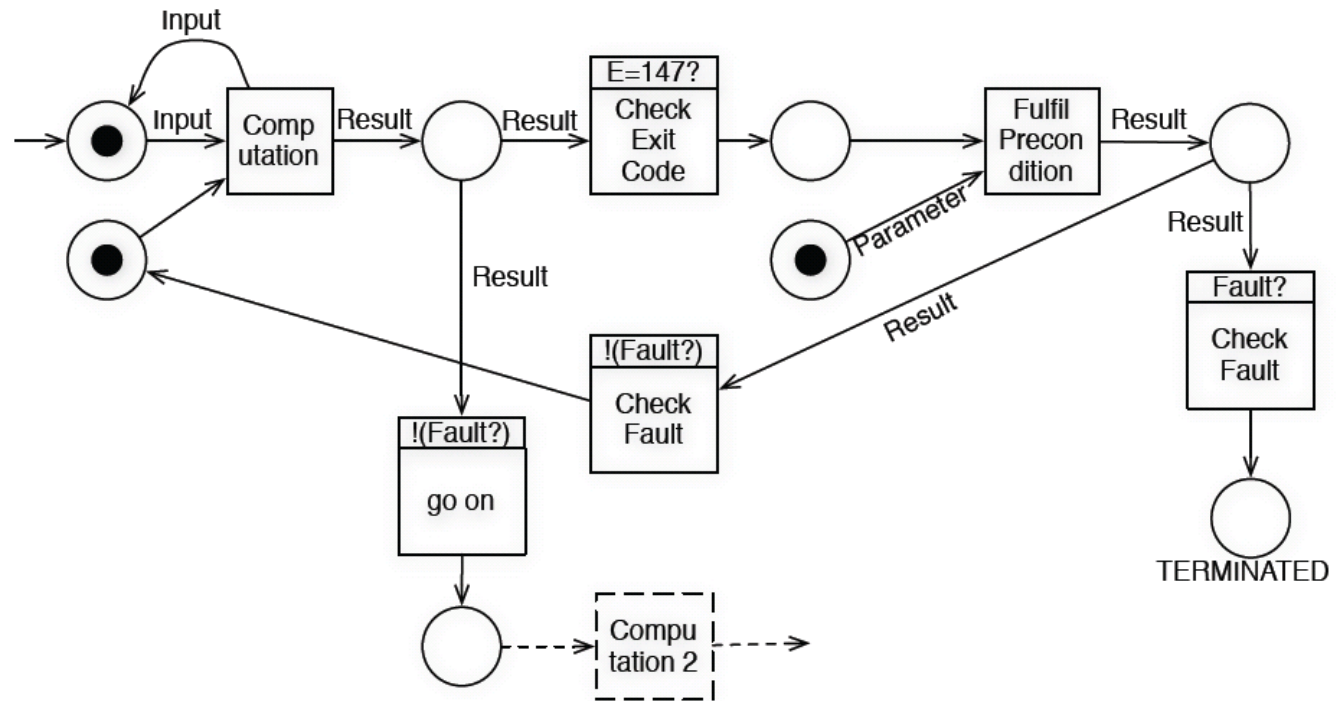


Workflow ohne  
Fehlertoleranz

Workflow mit Fehlertoleranz:  
Wiederholung einer fehlgeschlagenen Aktivität

(Quelle: Schulz, 2007)

# Beispiel für explizit modellierte Fehlertoleranz

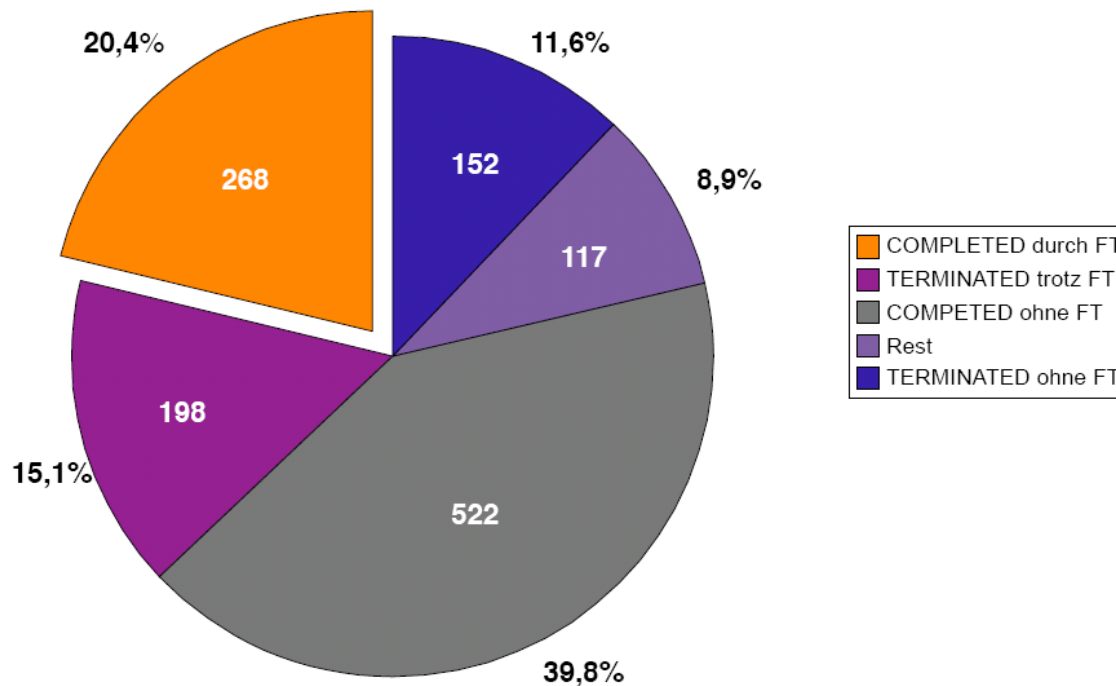


Workflow ohne Fehlertoleranz

Workflow mit Fehlertoleranz: Starten eines Lizenzservers falls Fehler „147“ auftritt und Wiederholung der Aktivität

(Quelle: Schulz, 2007)

# Statistik: Zustände von MediGRID-Workflows



Verteilung der Zustände von MediGRID-Workflows für Oktober 2007: 20% der Workflows konnte wegen der Fehlertoleranz trotz Fehlschlagens einer Aktivität erfolgreich abgeschlossen werden (Quelle: Schulz, 2007)

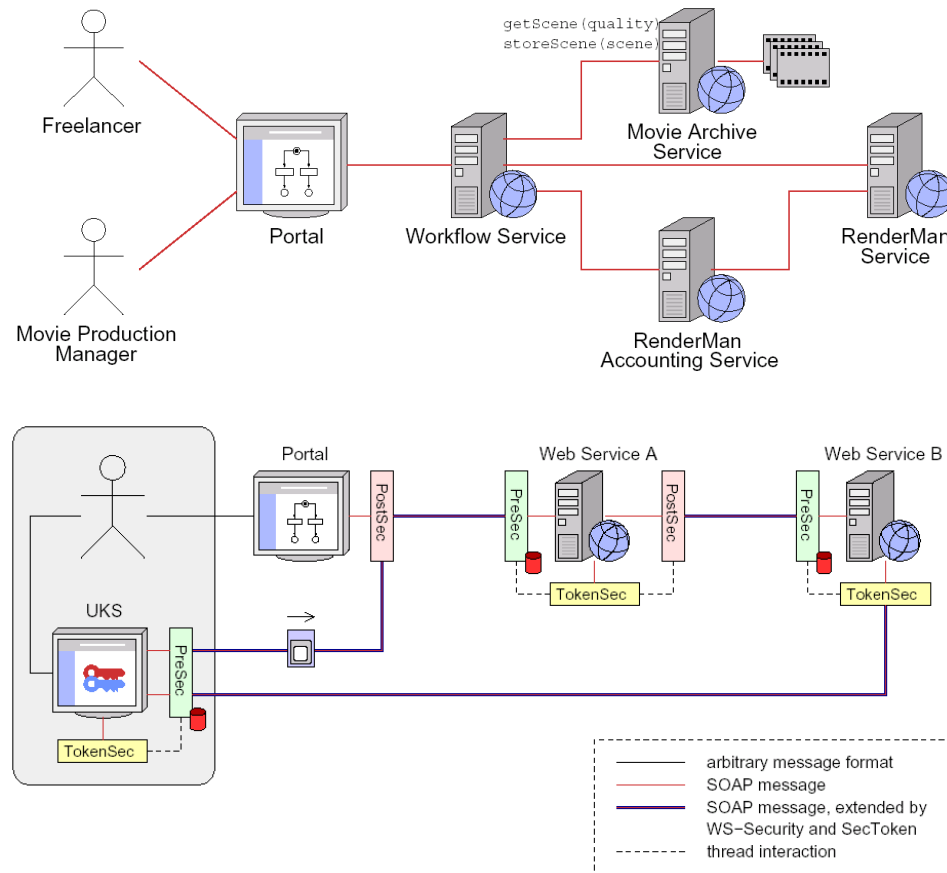
---

# Sicherheit für Grid-Workflows

# Sicherheitsinfrastruktur für Grid-Workflows: „Yagsi“

- Abgestufte Sicherheit für SOAs
  - existierende Web-Service-Architekturen
  - Grid-Services
- Unterstützung Virtueller Organisationen
- Feingranulare und rollenbasierte Autorisierung
- Eingeschränkte Delegation von Rechten unter Kontrolle des Nutzers

In Zusammenarbeit mit der  
Uni-Potsdam  
(Bettina Schnor, Stephan Müller)



---

# Anwendungen

# BauVOGrid – Ein Grid für die Baubranche

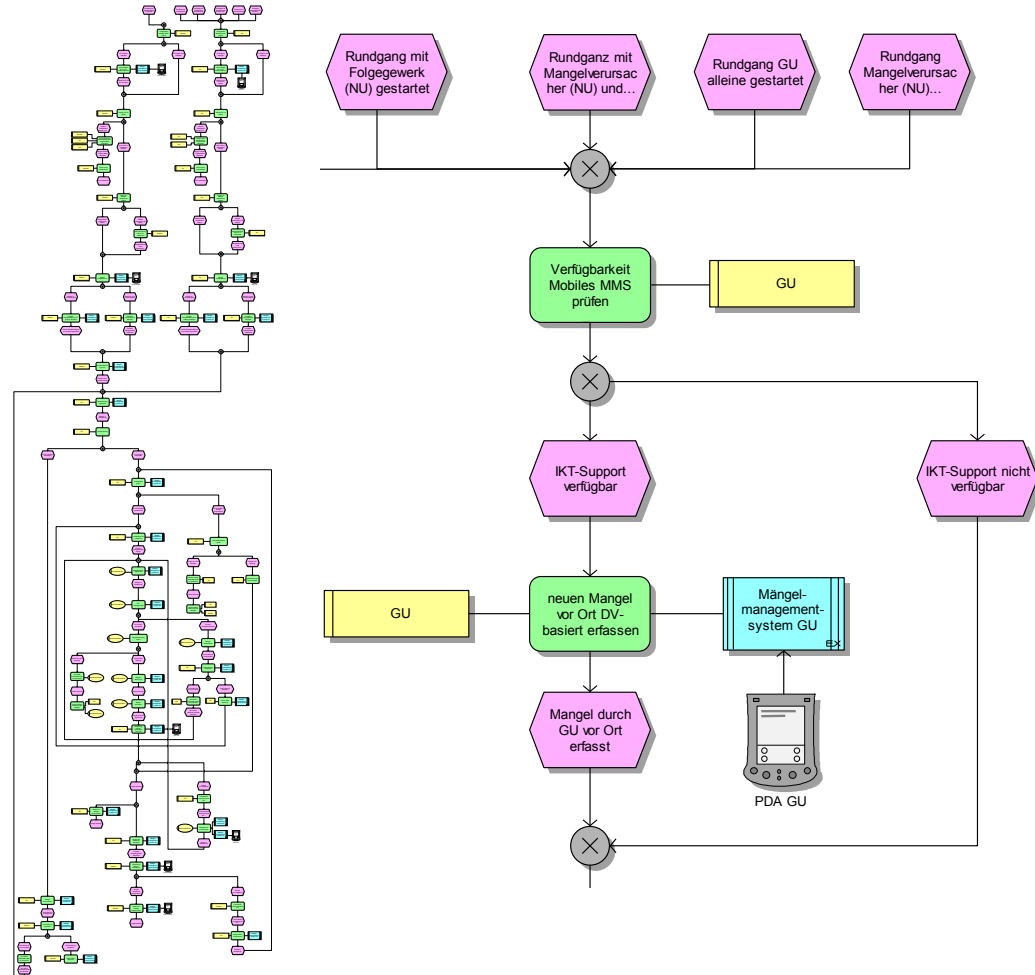
Verbesserung der Struktur, Funktionsweise und Operabilität virtueller Organisationen im Bauwesen

- Abbilden von Zuständigkeits- und Autorisierungsstrukturen
- Integration von Informationen aus unterschiedlichen Quellen
- Bereitstellen von Informationen im Büro als auch mobil
- Ermöglicht Ad-hoc Änderungen von Projektabläufen (z.B. ausgelöst durch Havarien)
- Mobile Erfassung von Prozessen und Prozess- sowie Produktionsständen auf der Baustelle



# BauVOGrid – Kopplung von ARIS und GWES

- ARIS: Methode und Werkzeug vom Marktführer IDS Scheer zur Modellierung und Simulation von Geschäftsprozessen
- ARIS verwendet Ereignisprozessketten (EPK) – ähnlich Petrinetzen
- Ein Ziel in BauVOGrid: Abbildung von EPKs auf GWorkflowDL zur Ausführung von Workflows im Grid
- ARIS als Workflow-Editor, enge Integration mit EPKs



# MediGRID Anwendungscluster

## Bioinformatik

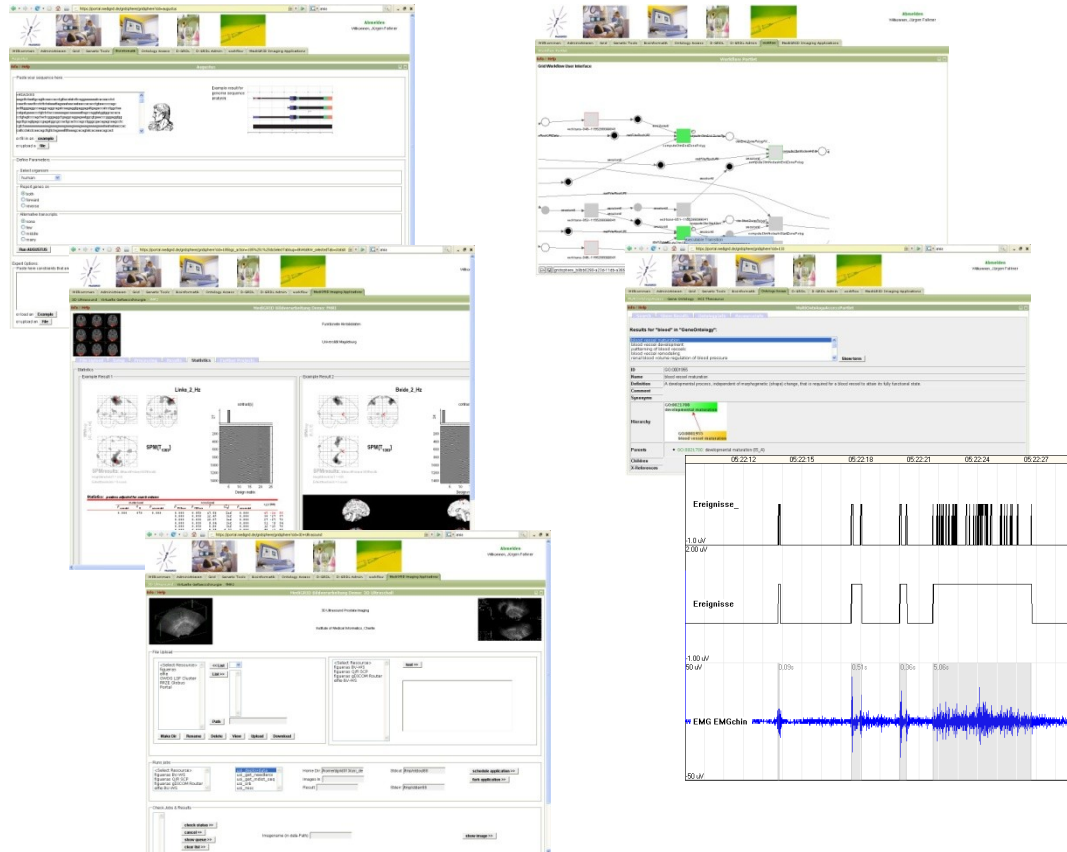
- Genomsequenzanalyse
- SNPSelection
- *RNAi*
- *Sequorr*

## Medizinische Bildverarbeitung 3D Ultraschall Prostatabiopsie

- Virtuelle Gefäßchirurgie
- Analyse funktioneller Hirnbilddaten

## Klinische Forschung

- Analyse klinischer Studien in der Neurologie



---

# Demonstration



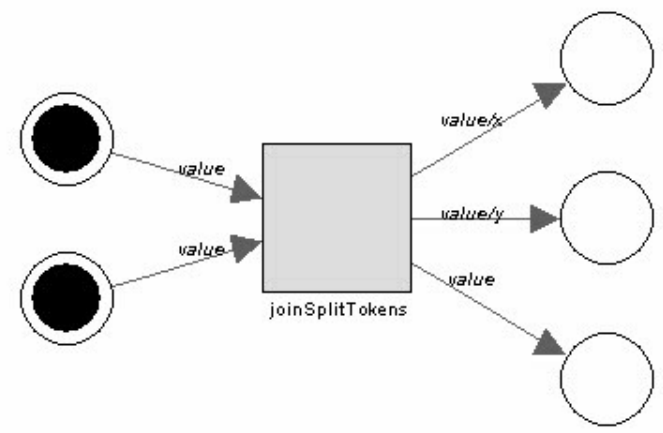
- Overview
- New
- List
- Details
- Graph
- Configuration/Test
- Tutorial

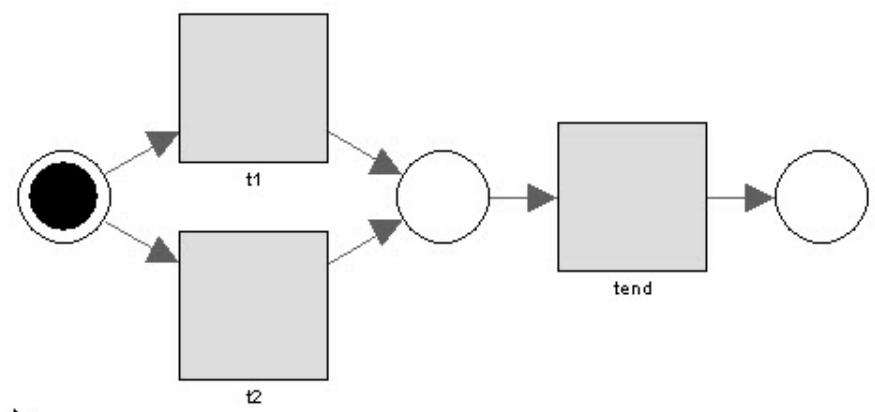
## Workflow List

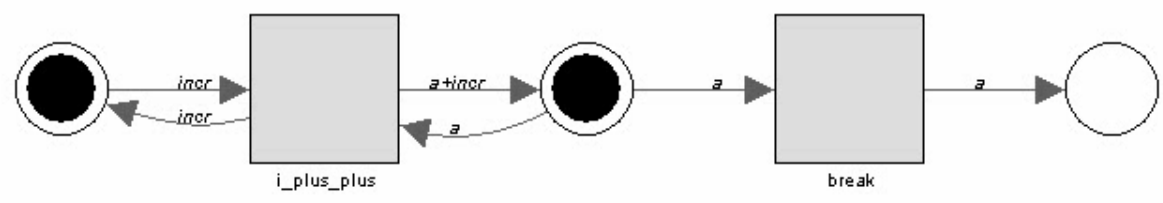
Refresh List

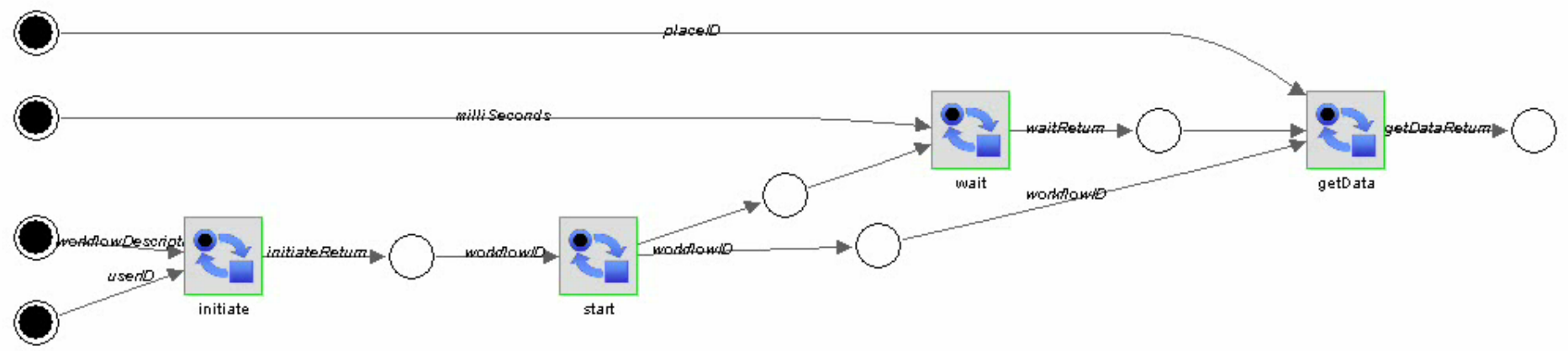
Currently the following workflow IDs are handled by the Grid Workflow Execution Service or stored in the workflow database:

	Actions	User	Description	Status	Begin	End	Level
<input type="checkbox"/>		medigrd	Test workflow	NEW INITIATED	Mar 02 18:28:06	--	MEMORY
<input type="checkbox"/>		medigrd	<a href="#">little workflow example that contains a conflict</a>	NEW INITIATED	Mar 02 18:21:22	--	MEMORY
<input type="checkbox"/>		medigrd	<a href="#">loop that counts from 0 to 100</a>	NEW INITIATED	Mar 02 18:21:14	--	MEMORY
<input type="checkbox"/>		medigrd	<a href="#">First join two tokens and then split t...</a>	NEW INITIATED	Mar 02 18:21:03	--	MEMORY
<input type="checkbox"/>		medigrd	Test workflow	COMPLETED	Mar 02 18:20:41	Mar 02 18:24:32	MEMORY
<input type="checkbox"/>		nagios	<a href="#">This workflow automatically deploys and execute...</a>	COMPLETED	Mar 02 17:27:36	Mar 02 17:32:07	MEMORY
<input type="checkbox"/>		nagios	<a href="#">Make 10x20s load on MediGrid</a>	COMPLETED	Mar 02 17:27:19	Mar 02 17:32:21	MEMORY
<input type="checkbox"/>		nagios	<a href="#">little workflow example that transfers one file...</a>	TERMINATED	Mar 02 16:40:26	Mar 02 16:41:36	MEMORY
<input type="checkbox"/>		nagios	<a href="#">AUGUSTUS MediGRID Workflow</a>	COMPLETED	Mar 02 14:12:00	Mar 02 14:20:52	MEMORY
<input type="checkbox"/>		nagios	<a href="#">MediGrid BV workflow</a>	TERMINATED	Mar 02 11:34:35	Mar 02 11:55:18	MEMORY
<input type="checkbox"/>		gwes	Test workflow	COMPLETED	Mar 01 20:05:40	Mar 01 20:05:41	MEMORY
<input type="checkbox"/>		gwes	Test workflow	COMPLETED	Mar 01 20:02:32	Mar 01 20:02:32	DATABASE
<input type="checkbox"/>		gwes	Test workflow	COMPLETED	Mar 01 20:02:26	Mar 01 20:02:26	DATABASE
<input type="checkbox"/>		gwes	Test workflow	COMPLETED	Mar 01 19:53:48	Mar 01 19:53:48	DATABASE
<input type="checkbox"/>		nagios	<a href="#">This workflow automatically deploys and execute...</a>	COMPLETED	Mar 01 17:27:36	Mar 01 17:36:07	DATABASE
<input type="checkbox"/>		nagios	<a href="#">Make 10x20s load on MediGrid</a>	COMPLETED	Mar 01 17:27:19	Mar 01 17:34:13	DATABASE
<input type="checkbox"/>		nagios	<a href="#">little workflow example that transfers one file...</a>	TERMINATED	Mar 01 16:40:26	Mar 01 16:41:26	DATABASE
<input type="checkbox"/>		nagios	<a href="#">AUGUSTUS MediGRID Workflow</a>	COMPLETED	Mar 01 14:12:00	Mar 01 14:16:01	DATABASE
<input type="checkbox"/>		nagios	<a href="#">MediGrid BV workflow</a>	TERMINATED	Mar 01 11:34:34	Mar 01 11:55:06	DATABASE
<input type="checkbox"/>		nagios	<a href="#">This workflow automatically deploys and execute...</a>	COMPLETED	Feb 29 17:27:36	Feb 29 17:29:27	DATABASE
<input type="checkbox"/>		nagios	<a href="#">Make 10x20s load on MediGrid</a>	COMPLETED	Feb 29 17:27:19	Feb 29 17:36:31	DATABASE











Logout  
Welcome, Andreas  
Hoheisel

### GWES Overview



Welcome to the Grid Workflow Execution Service. Here you can:

- [Configure and test](#) the Grid Workflow Execution Service
- [Learn](#) more about the workflow formalism by starting some example workflows
- [Upload](#) a new workflow
- [List](#) workflows
- [Directly access](#) the workflow history database (The default login for the XML database is "admin" without password)
- [Directly access](#) the resource description database (The default login for the XML database is "admin" without password)
- [Read](#) the GWES User Manual
- [Read](#) the GWES Developer Manual

Further information about the Grid Workflow Execution Service is available at:

- [GWES software development site](#)
- [Grid activities of Fraunhofer FIRST](#)
- [The Fraunhofer Resource Grid](#)
- [K-Wf Grid](#)
- [Instant-Grid](#)
- [MediGRID](#)
- [Enterprise Grids](#)





Logout  
Welcome, Andreas  
Hoheisel

- Welcome
- Administration
- Grid Workflows
- Monitoring
- D-GRDL
- D-GRDL Admin
- Genetic Tools
- MediGrid Bildverarbeitung
- Ontology Access
- Bioinformatics

Augustus 2 DiAlign

Info / Help

Augustus

Input

Upload (mutiple) FASTA file

 Browse...

Organism



Aedes aegypti

Options

Report genes on

- both strands
- forward strand only
- reverse strand only

Alternative transcripts

- none
- few
- medium
- many

example input

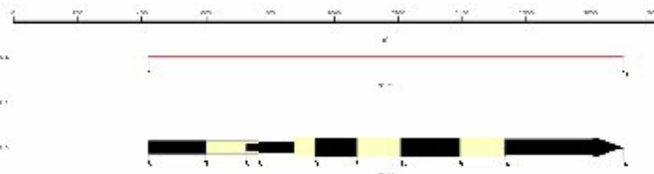
use this example

>HS04636

```
gagctcacattaactatttacagggtaacgtctaggaccagtattatgaggagaattta
ccttcccgcctctcttccaagaaacaaggagggggtgaaggtaacggagaacagattt
ctctgtgaaagcaacttagctacaaagataaattacagctatgtacactgaaggtagc
tatttcattccacaaaataagagtttttaaaaagctatgtatgtgtctgcatatag
agcagatatacagcctattaagcctgctactaaaacataaacatgctagcctttcta
...
```

>HS08198

```
agcgggcggcggtcgtggggggggtgcaggcgaggctcaacgaacgtggtctgaccgt
cggcgctccctgttgcggggccctgagcaagtggctcatgaaccccgtagcgttggcca
tggagataagaccactgggtgatggtttaaggaagataacgtgtaaaagggttaaggactg
tcggtggaatcagggtgcaggagaaatggataaacagccagaggtcaactcggacttt
gtacataggacatggtgccaggccctgcagggaagtgcagatcgaagctaggctcacgag
...
```



check options and start job

Expert options

Allowed gene structure

- predict any number of (possibly partial) genes
- predict only complete genes
- predict only complete genes - at least one

Overlapping genes

ignore conflicts with other strand (allow overlapping genes)

Untranslated regions



# D-Grid: Mögliche Synergien bzgl. Workflows

---

- **Workflow-Editor** z.B. gemeinsamer Portal-basierter Workflow-Editor mit Online-Überprüfung (Typsicherheit, Deadlock, Lebendigkeit, ...)
  - **Ausführen von Aktivitäten:** Gemeinsame Middlewarekomponente zur Ausführung von gekoppelten Aktivitäten auf D-Grid-Ressourcen (Meta-Scheduling)
  - **Interoperabilität** innerhalb von D-Grid herstellen und Interoperabilität mit externen Workflow-Systemen untersuchen (z.B. Triana, Taverna, Karajan)
- ? **Workflow-Beschreibung:** Vereinheitlichen der Workflow-Beschreibung wegen unterschiedlichen Anforderungen und politischen bzw. zeitlichen Rahmenbedingungen in D-Grid nicht sinnvoll?

# Vielen Dank für Ihre Aufmerksamkeit!

---

Kontakt:

[andreas.hoheisel@first.fraunhofer.de](mailto:andreas.hoheisel@first.fraunhofer.de)

<http://www.andreas-hoheisel.de/>

Download Grid Workflow Execution Service:

<http://www.gridworkflow.org/gwes/>

