Provenance für Workflows und Daten

Grid Workflow Workshop 2011 (04.03.2011, Köln)

Andreas Schreiber <*Andreas.Schreiber@dlr.de*>
Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Köln-Porz
`http://www.dlr.de/sc`

Überblick

- Definition „Provenance“
- Anwendungsgebiete
  - Wofür ist das wichtig?
  - Vorteile und Nutzen
- Provenance Model und Methodik
  - Open Provenance Model
  - PrIMe
- Implementierung
- Ausblick

Definition „Provenance“

Was ist „Provenance“?

Das sagen Wörterbücher…

Duden Fremdwörterbuch

Provenienz […*we*…; zu *lat.* provenire „hervorkommen, entstehen“] *die*; -, -en: Herkunft, Ursprung

Wiktionary (deutsch)

Provenienz

Bedeutungen:

[1] Herkunft, Ursprung

[a] die über die Vorbesitzer nachgewiesene Herkunft einer Ware.

[b] im Buchwesen den oder die Vorbesitzer eines Schriftstücks, meist private Sammlungen oder geistliche Institutionen, aus denen das Exemplar in den heutigen Besitz einer Bibliothek gelangt ist: Provenienz (Buch).

[c] im Archivwesen ***die Herkunft betreffend***

[d] in der Medizin wird ***Provenienz*** auch einfach im Sinne von ***Herkunft*** benutzt

Bestimmungen und Konformität
Hintergrundwissen

Wir leben in einer regulierten Welt:

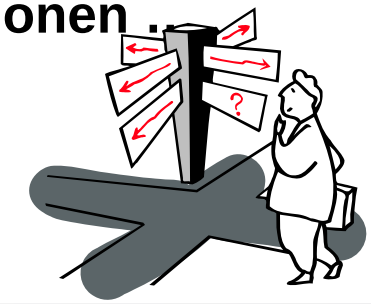
- Audits, Konformität und Regulierung: Teil der Geschäftssprache
- Geschäftliche „Spielregeln“: Produkthaftung, ISO 9000, Basel II, „Richtlinien zur Sicherung guter wissenschaftlicher Praxis”, …

Wie erzeugen und verwalten Organisationen Informationen?

- Akademischer „peer review” für Wissenschaft und Forschung
- Audit-Regeln für finanzielle Transaktionen
- Flugsicherungs-Regulierungen
- Bestimmungen zur Sicherheit von Patienten-Informationen
- Verfahren zur Zulassung pharmazeutischer Produkte

Für Konformität müssen Prozesse und gewonnene Informationen ..

- offen, transparent und auditierbar sein
- eine geprüfte Integrität haben



Provenance in der Informatik

Was wir darunter verstehen …

Ursprung und Authentizität von Ergebnissen

- Aufzeichnung von Prozessinformationen zur Laufzeit des Prozesses
- Mit dieser Dokumentation kann folgendes ermittelt werden:
  - Der Ursprung der Daten
  - Die Konformität des (Daten-) erzeugenden Prozesses

Das bezeichnen wir als „Provenienz”:

Die Provenienz einer Information ist die Geschichte ihrer Erzeugung

Provenance

Von der Anwendung zur Repräsentation

Beispiel: Bei komplexen Simulationen Aufzeichnung von

- Eingabedaten (Parameter),
- Programmausführungen,
- beteiligte Rechner oder
- erzeugte Dateien.

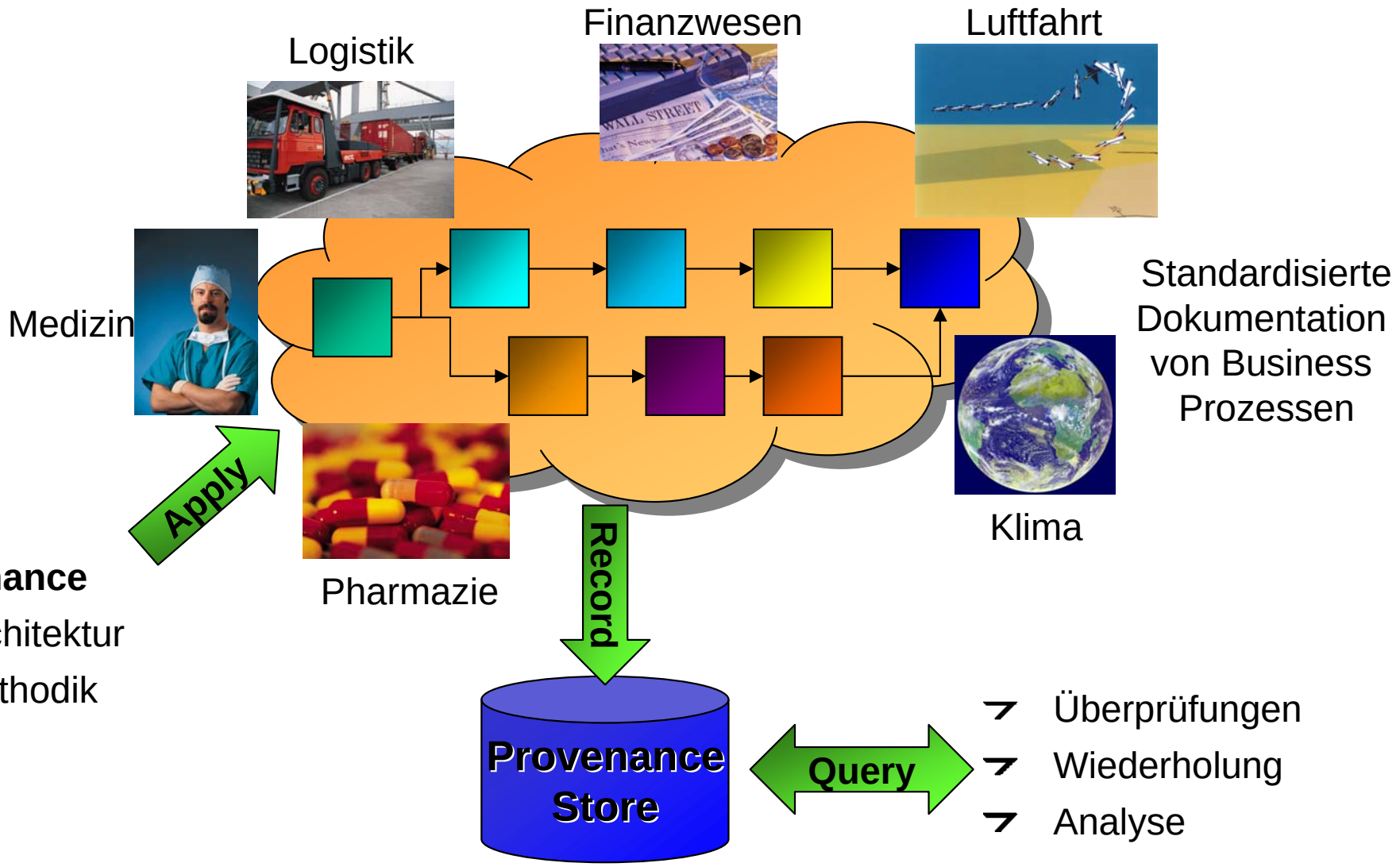
UND BEZIEHUNGEN UNTEREINANDER
(Unterschied zu „klassischem“ Logging!)

Dokumentation des Prozesses: „Provenance“-Informationen

dazu notwendig

Computergestützte **Repräsentation der Provenienz**, die uns erlaubt
- Sinnvolle Analysen durchzuführen
- Unsere Anwendungen zu belegen

Anwendung auf (verteilte) Prozesse



Anwendungsgebiete

Anwendungsbereiche

- Medizin
- Ingenieurwissenschaft
- Klimaforschung
- Finanzwirtschaft
- Bioinformatik
- Pharmazie
- Software-Entwicklung

Provenance in der Medizin

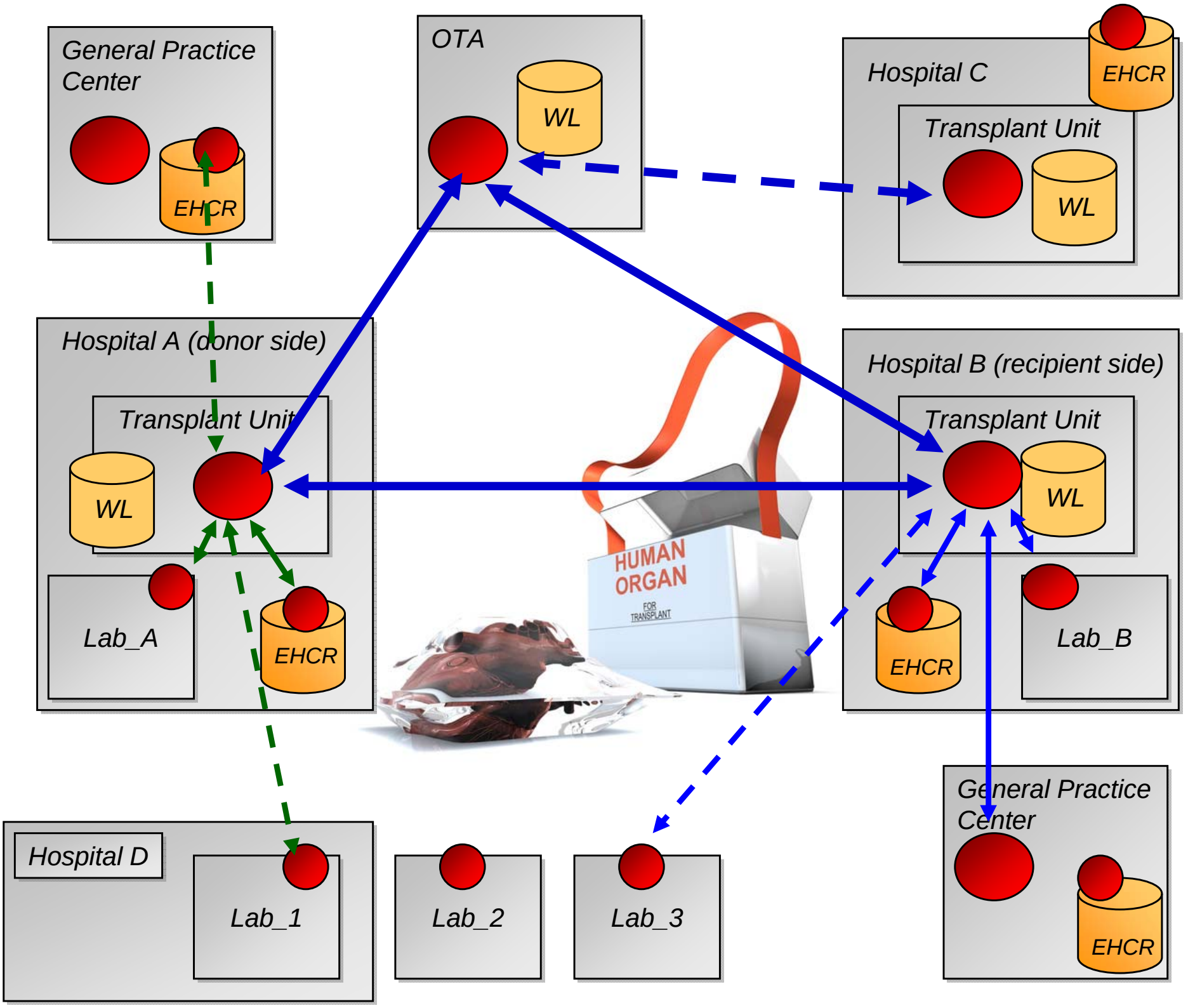
Nutzen von Provenance in der Medizin

- Einheitliche Sicht auf Daten, Workflows, Logbücher
- Überblick über alle früheren Behandlungen von Patienten
- Analyse der verteilten medizinischen Infrastruktur zu Erkennung von Problemen
- Durchführung von Audits (medizinisch / rechtlich)

Aufzeichnung von Provenance-Daten

- Ursprung medizinischer Entscheidungen und Workflows
- Die verfügbaren medizinischen Information in jedem Prozessschritt
- Der Ursprung dieser Informationen

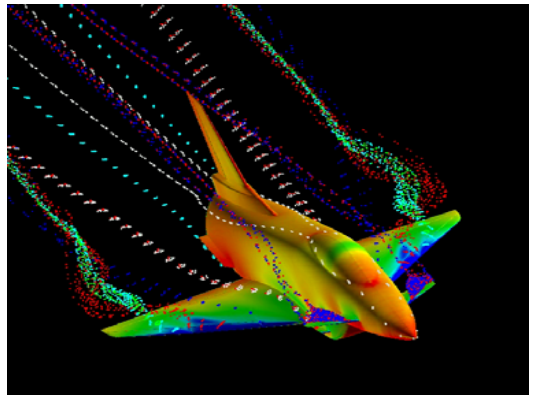
Beispiel: Organtransplantations-Management


General Practice Center
EHCR
OTA
WL
Hospital C
EHCR
Transplant Unit
WL
Hospital A (donor side)
Transplant Unit
WL
Lab_A
EHCR
HUMAN ORGAN
FOR TRANSPLANT
Hospital B (recipient side)
Transplant Unit
WL
EHCR
Lab_B
Hospital D
Lab_1
Lab_2
Lab_3
General Practice Center
EHCR

Provenance in den Ingenieurwissenschaften

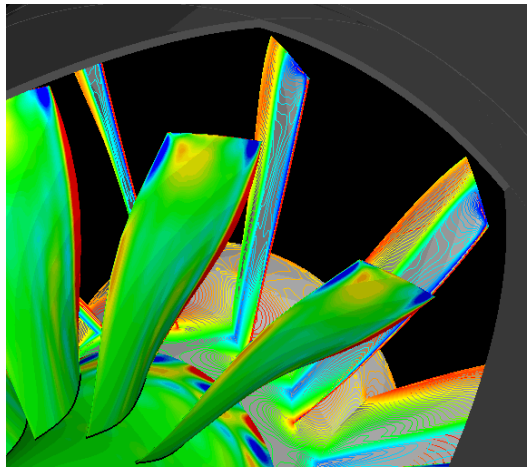
Nutzen von Provenance

- Detaillierte Nachvollziehbarkeit des Entstehungsweges eines Berechnungsergebnisses
- Klare Dokumentation von verteilten Berechnungs-Workflows
- Möglichkeit zum „Re-run“ von Simulationen
- Einfache Überprüfung auf Einhaltung von Regularien



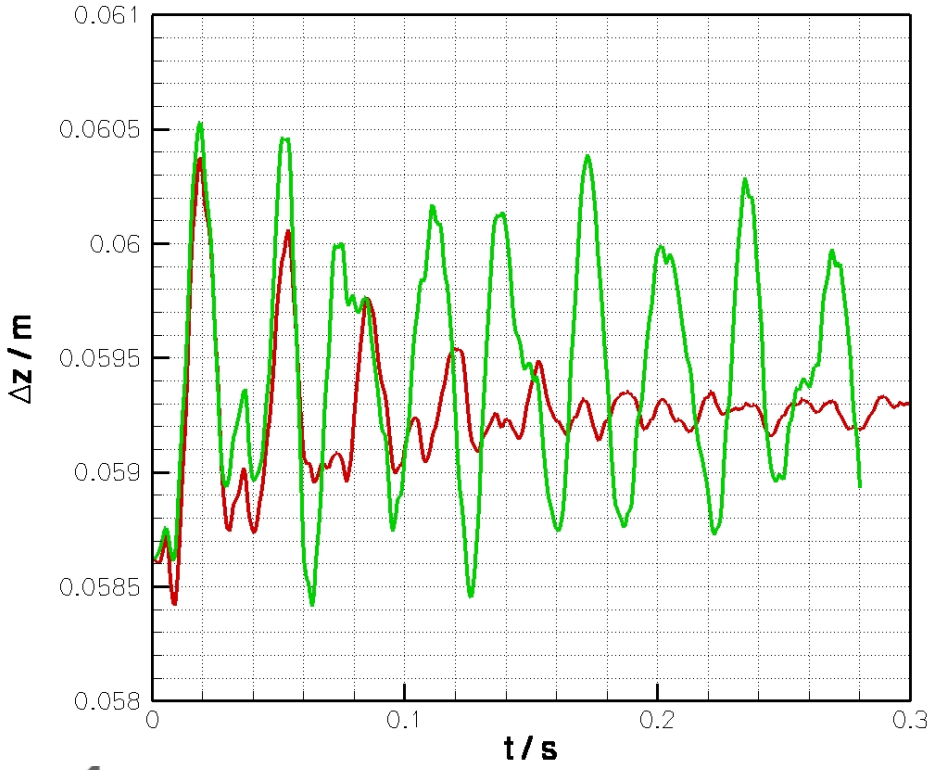
Aufzeichnung von Provenance-Daten

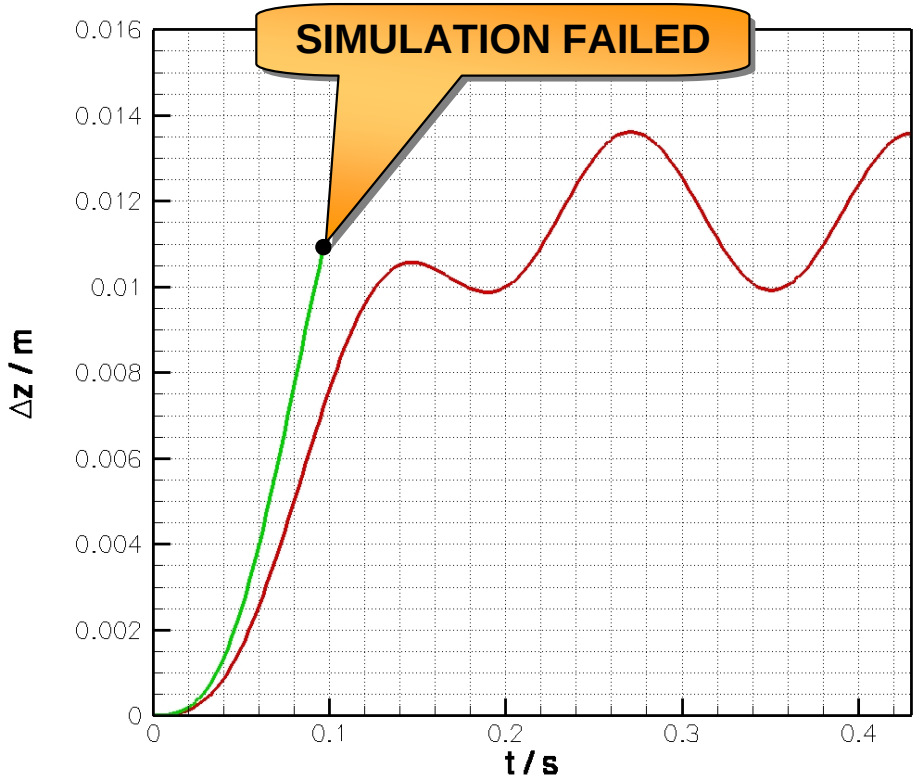
- Modell-Parameter
- Tool-Versionen
- Verwendete Hardware
- Verwendete Libraries & Compiler(-Parameter)



Frage: “Was ist passiert?”

- Lange Rechenzeiten:
  mehrere Tage auf großen HPC-Systemen <u>pro Konfiguration</u>





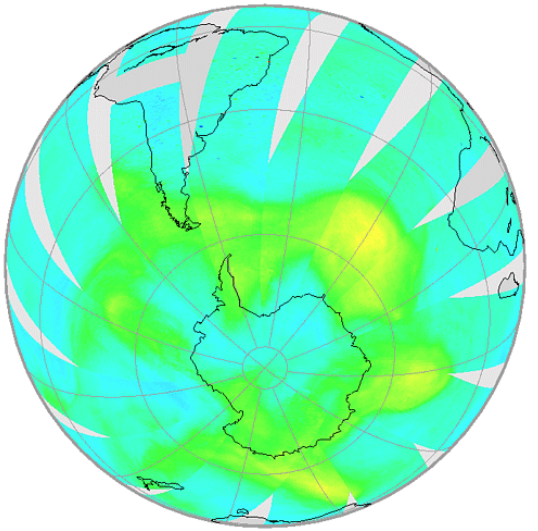
Provenance in den Ingenieurwissenschaften

Fragestellungen zur Nutzung der Informationen

- *In welcher Simulation wurde eine bestimmte Datei erzeugt?*
- *In welchen Simulationen wurde ein bestimmtes Modell berechnet?*
- *In welchen Simulationen wurde ein bestimmter Parameter verwendet?*
- *Welche Monitoring-Informationen wurden in einer Simulation mit Parameter == x aufgezeichnet?*
- *Welche Simulationen wurden mit einer bestimmten numerischen oder Modell-Konfiguration gerechnet?*
- *Haben bei vertraulichen/geheimen Rechnungen die Daten die Rechner der Firma nicht verlassen?*

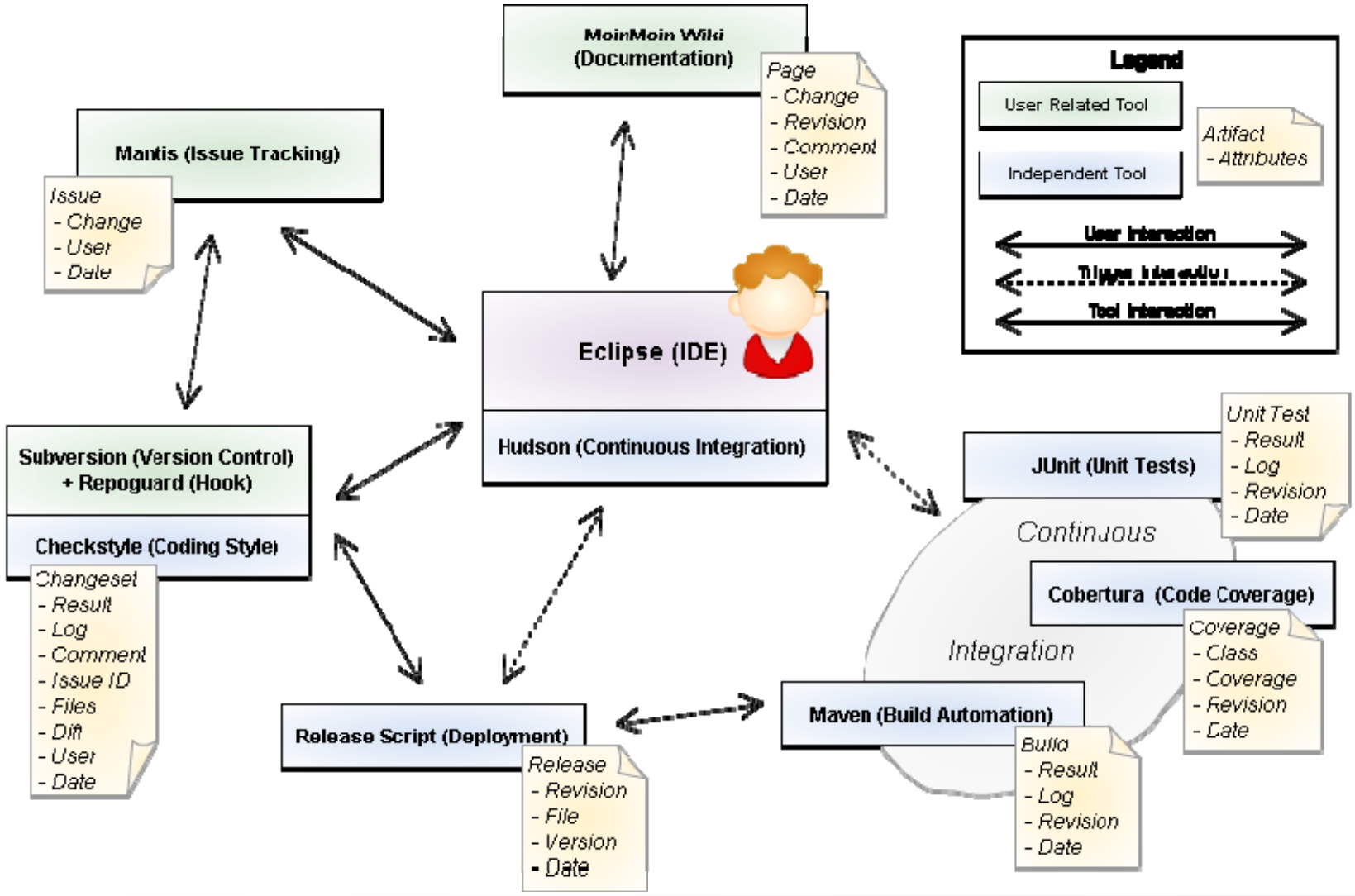
Provenance in der Klimaforschung

- Klimadaten werden in unterschiedlich(st)en Formaten ausgetauscht
- Suche und Benutzung der Daten erfordert standardisierte Metadaten (ISO 19115/19139)
- Sicherstellung der Datenqualität durch Provenance-Dokumentation („Lineage“):
  - Prozessierungsschritte
  - Datenquellen



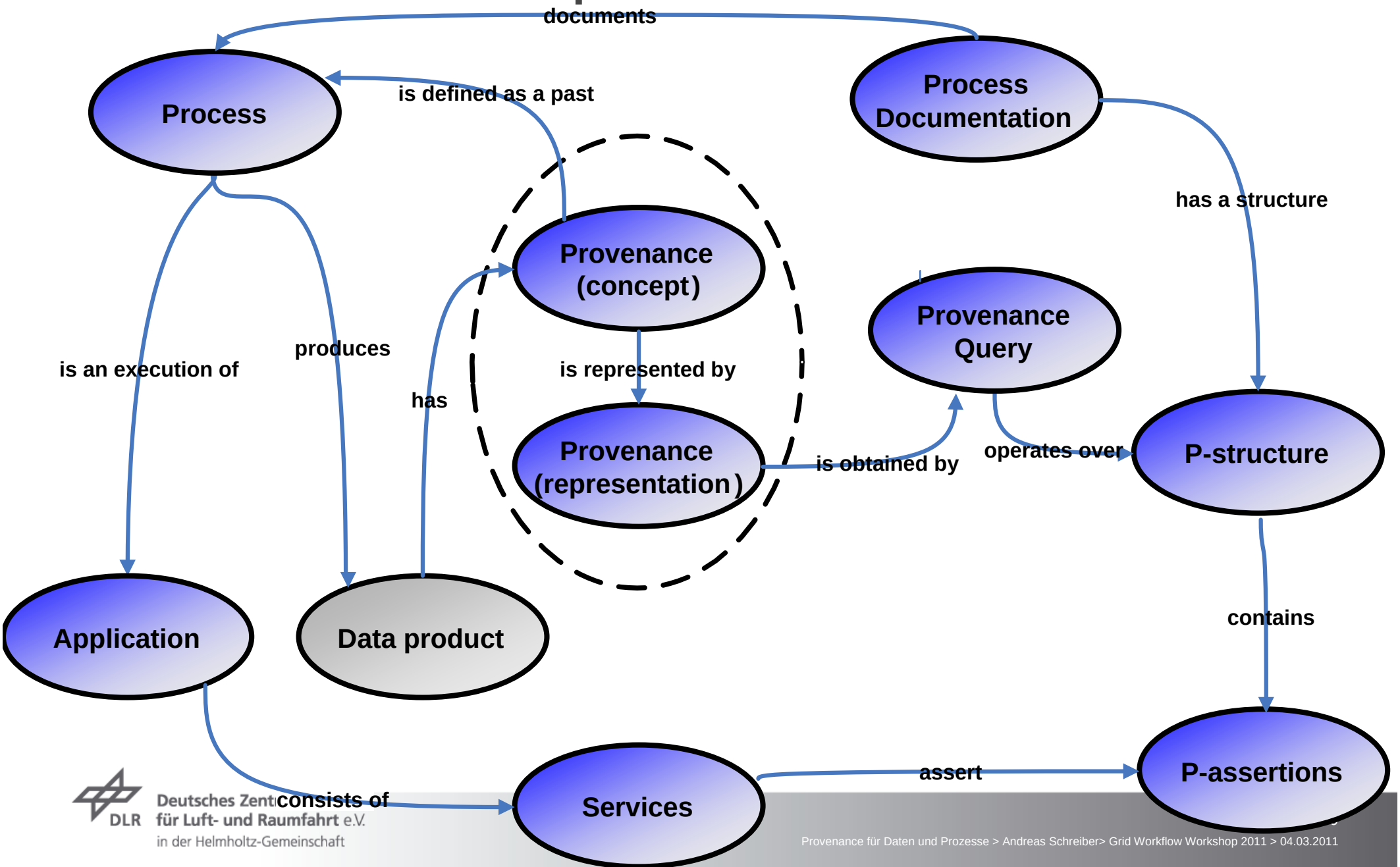
Provenance im Software Engineering

Nachvollziehbarkeit in komplexen SE-Prozessen

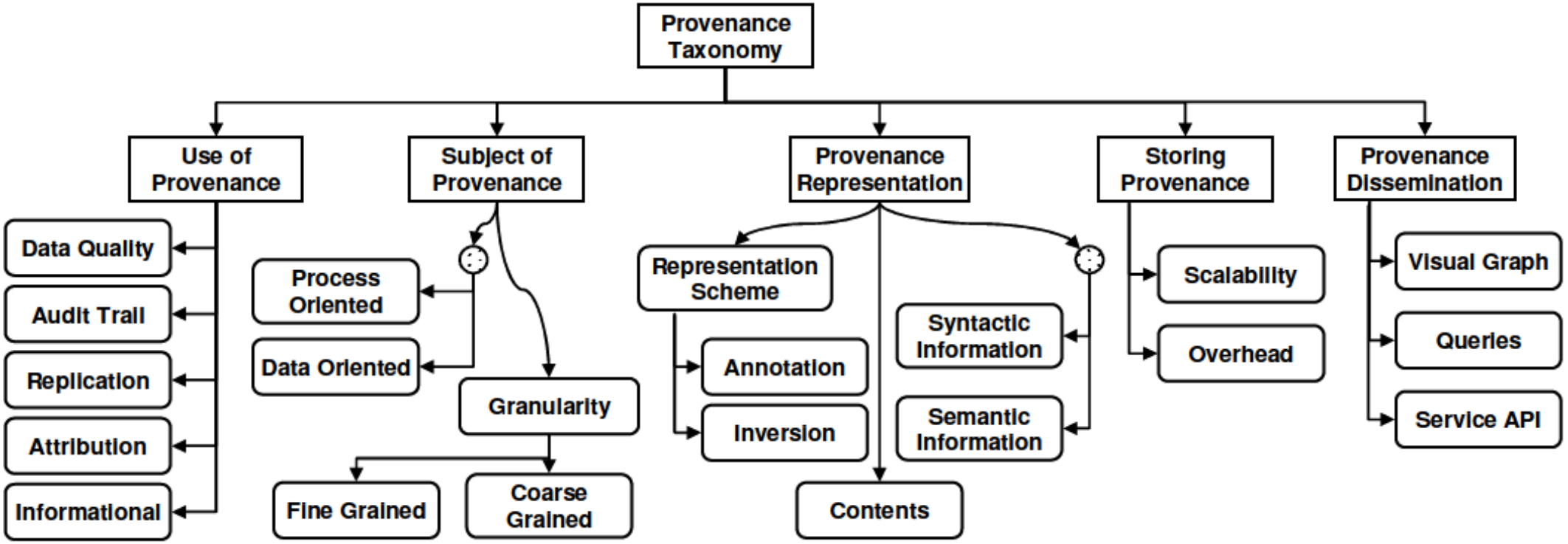


Provenance Model

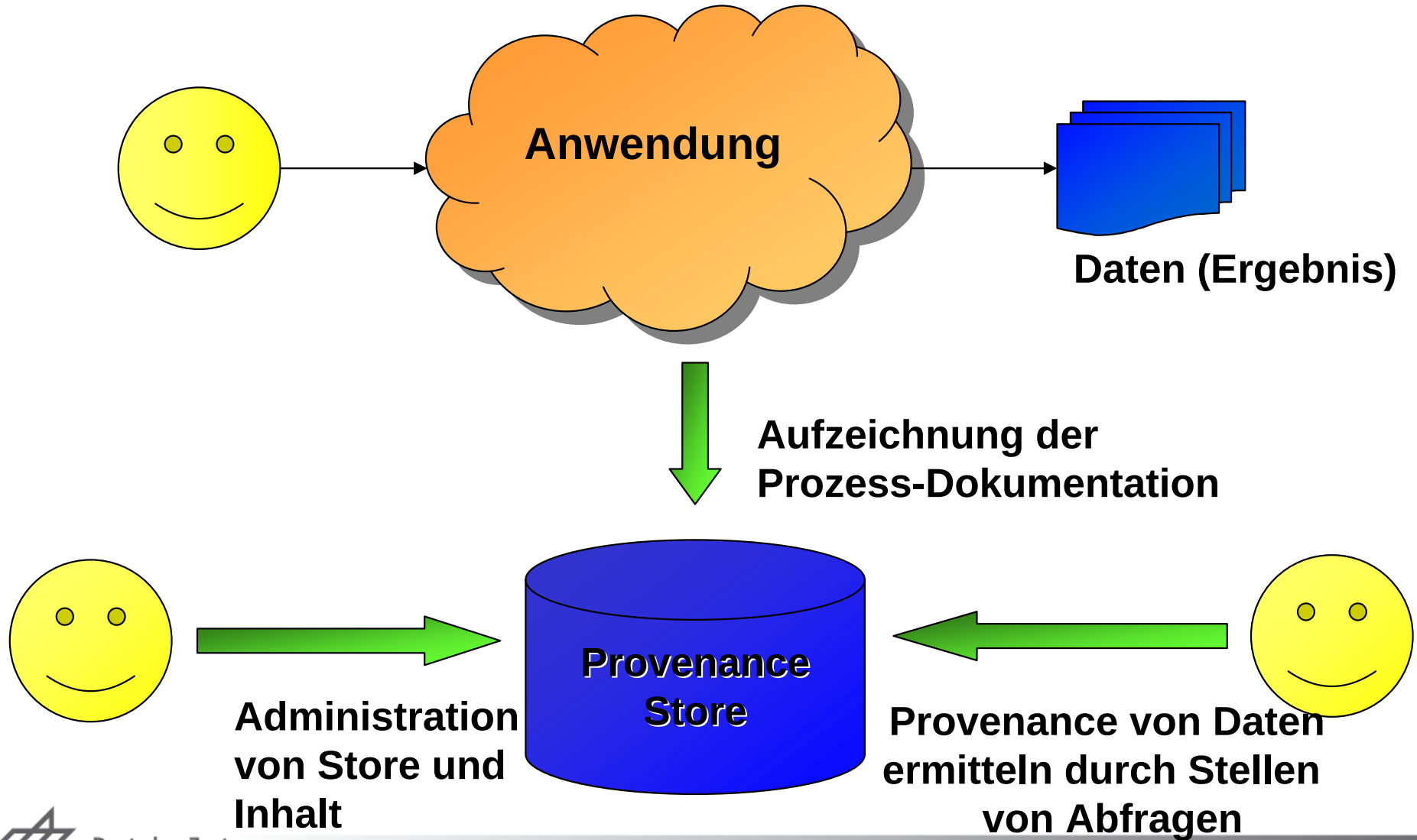
Provenance-Konzepte



Provenance Taxonomie



Provenance Life Cycle



Provenance-Beispiel

Die Provenance einer Flasche Wein umfasst beispielsweise

- die Trauben, aus denen er hergestellt wurde,
- den Ort, wo die Trauben gewachsen sind,
- der Prozess der Weinherstellung,
- die Art, wie der Wein gelagert wurde,
- die Beteiligten, zwischen denen der Wein transportiert wurde (z.B. erst vom Hersteller zum Großhändler, dann zum Händler) und
- das Auktionshaus, das den Wein versteigert hat.

Open Provenance Model (OPM)

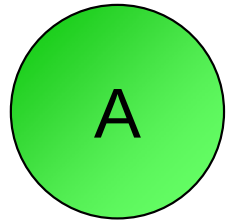
- Erlaubt, das Zustandekommen von Dingen zu beschreiben
- Ermöglicht eine **Prozessorientierte** und eine **Datenflußorientierte** Sicht
- Basiert auf der Vorstellung eines **annotierten Kausalitätsgraphen**

  (gerichteter azyklischer Graph, DAG)

Open Provenance Model
Nodes

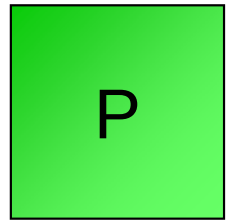
Artefakt *(Artifact)*

- Unveränderlicher Zustand
- Kann eine physikalische Verkörperung in Form eines physikalischen Objekts haben oder eine digitale Repräsentation in einem Computer sein



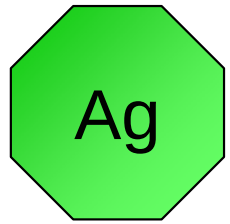
Prozess *(Process)*

- Eine Aktion oder Serie von Aktionen ausgeführt auf oder verursacht durch Artefakte
- Resultiert in neuen Artefakten



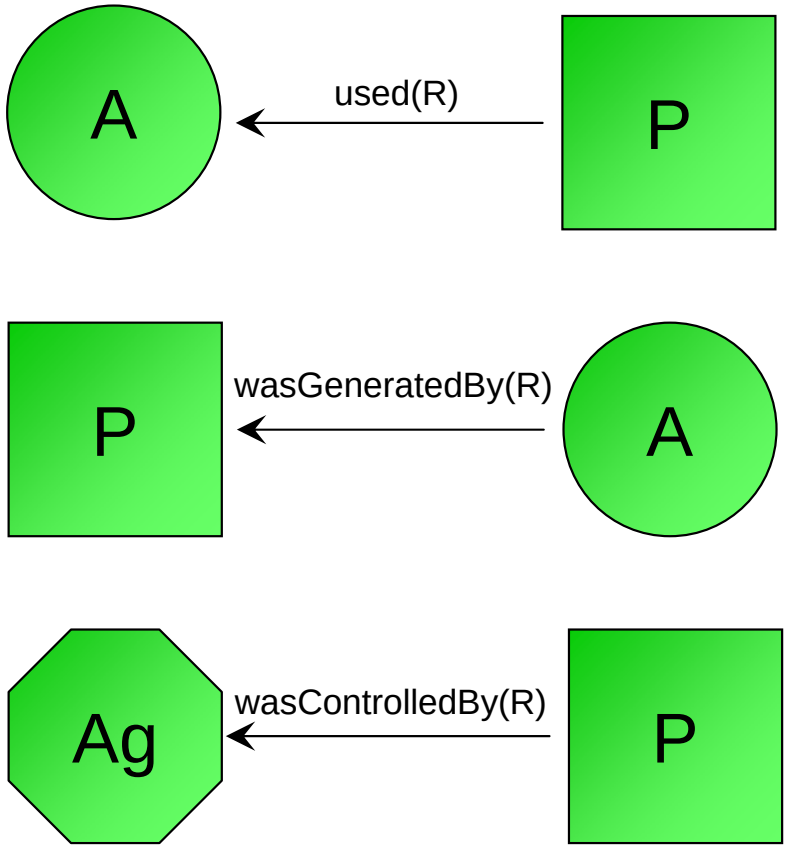
Agent

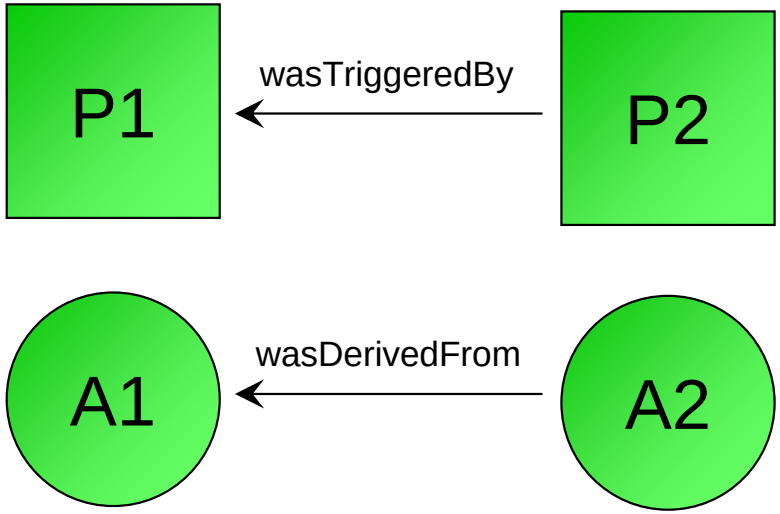
- Kontextabhängige Instanz, die als Katalysator für den Prozess wirkt
- Ermöglicht, erleichtert oder kontrolliert die Ausführung



Open Provenance Model
Edges

Edges können Rollen haben „(R)“ als textuelle Beschreibung.





Edges werden in der Vergangenheitsform bezeichnet, um zu verdeutlichen, dass es sich um einen vergangenen Prozessschritt handelt.

Open Provenance Model
Annotations

- Hinzufügen weiterer Informationen zum Graphen
- Annotiert werden können
  - der ganze Graph,
  - einzelne Nodes,
  - Edges und
  - Annotationen.

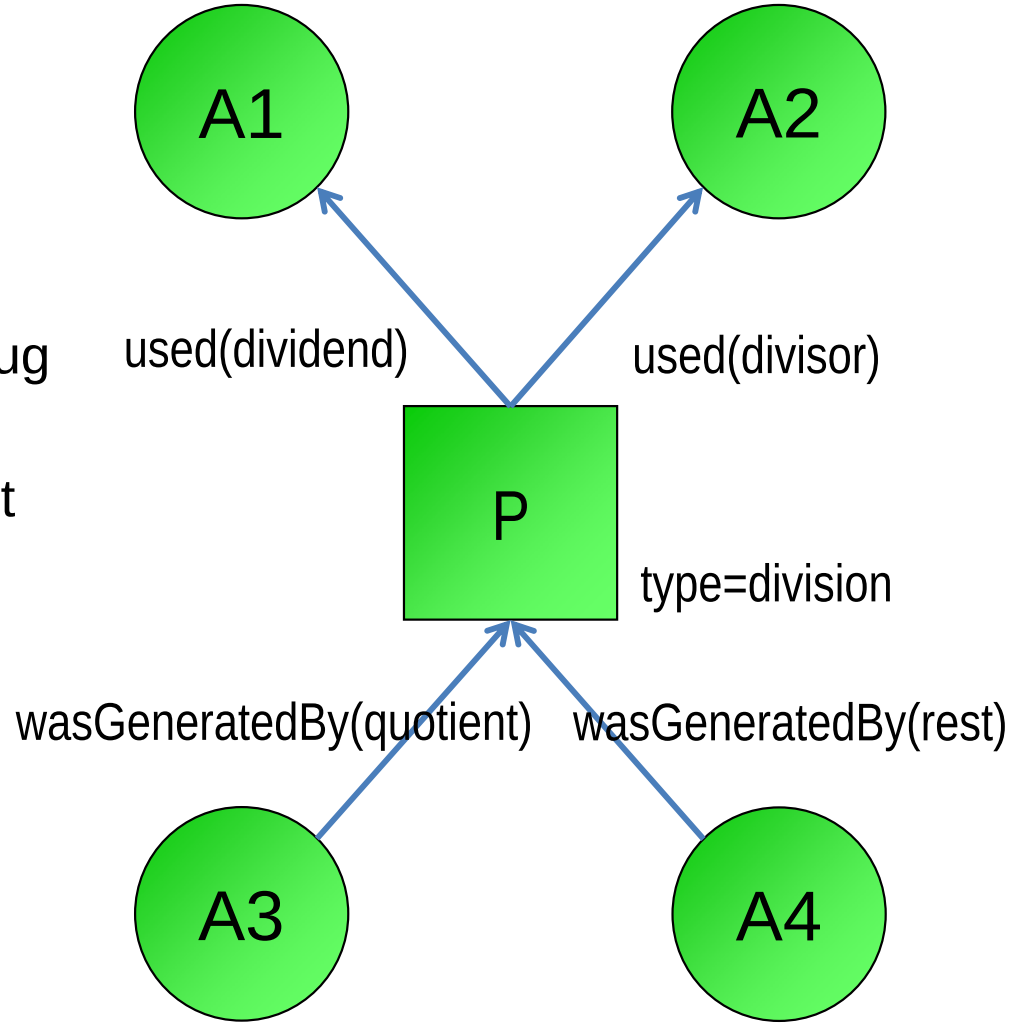
- Annotationen sind eine Liste von Key-Value-Paaren

Open Provenance Model

- Ein Prozess benutzt (“used”) Artefakte und generiert (“generated”) Artefakte
- Die Rollen der Edges bezeichnen die Funktion der Artefakte im Bezug auf den Prozess
- Edges und Nodes können typisiert sein

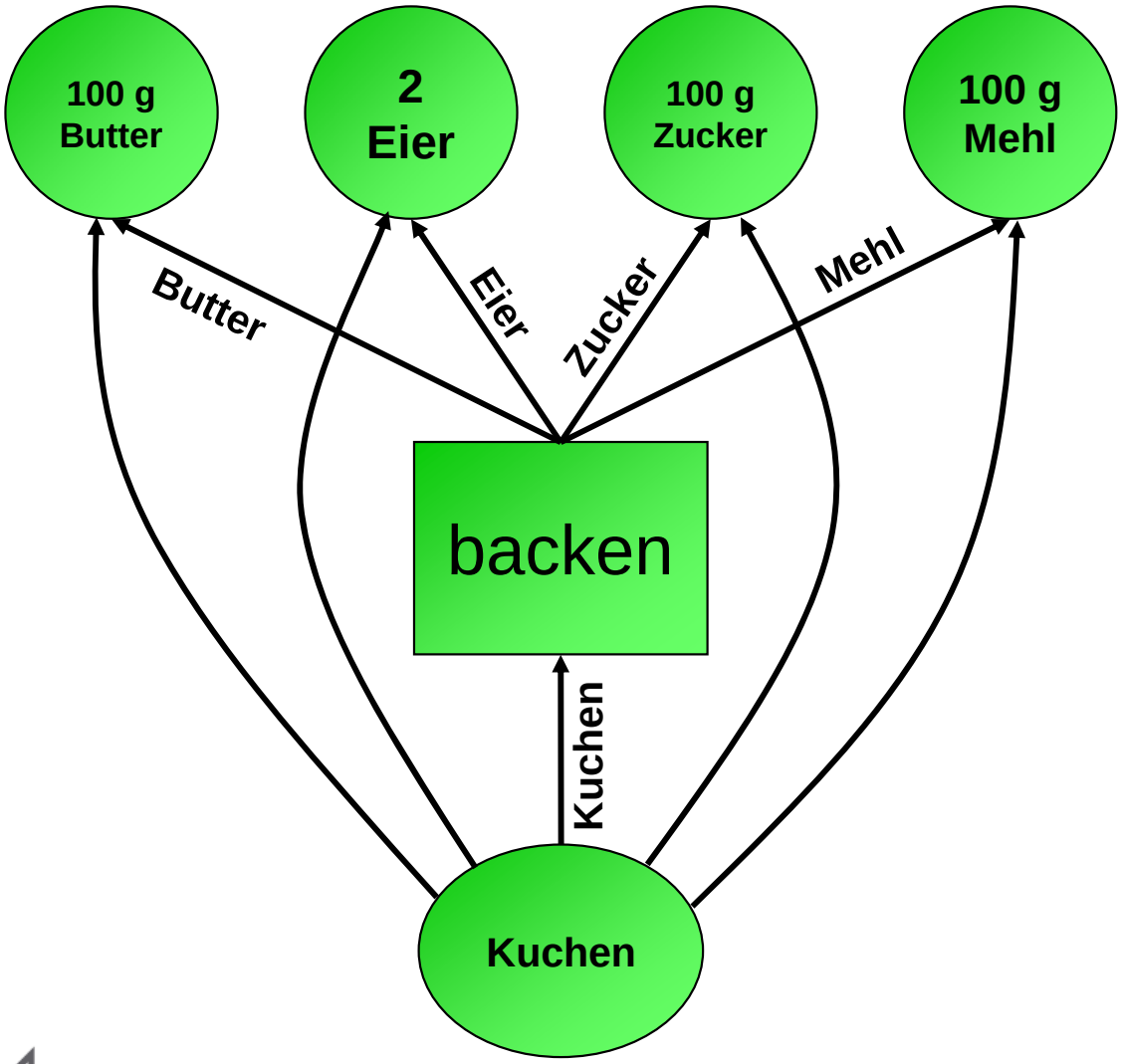
Kausalkette

- P wurde verursacht durch A1 und A2
- A3 und A4 wurden verursacht durch P



Open Provenance Model

Kuchen backen



Vorgehensweise

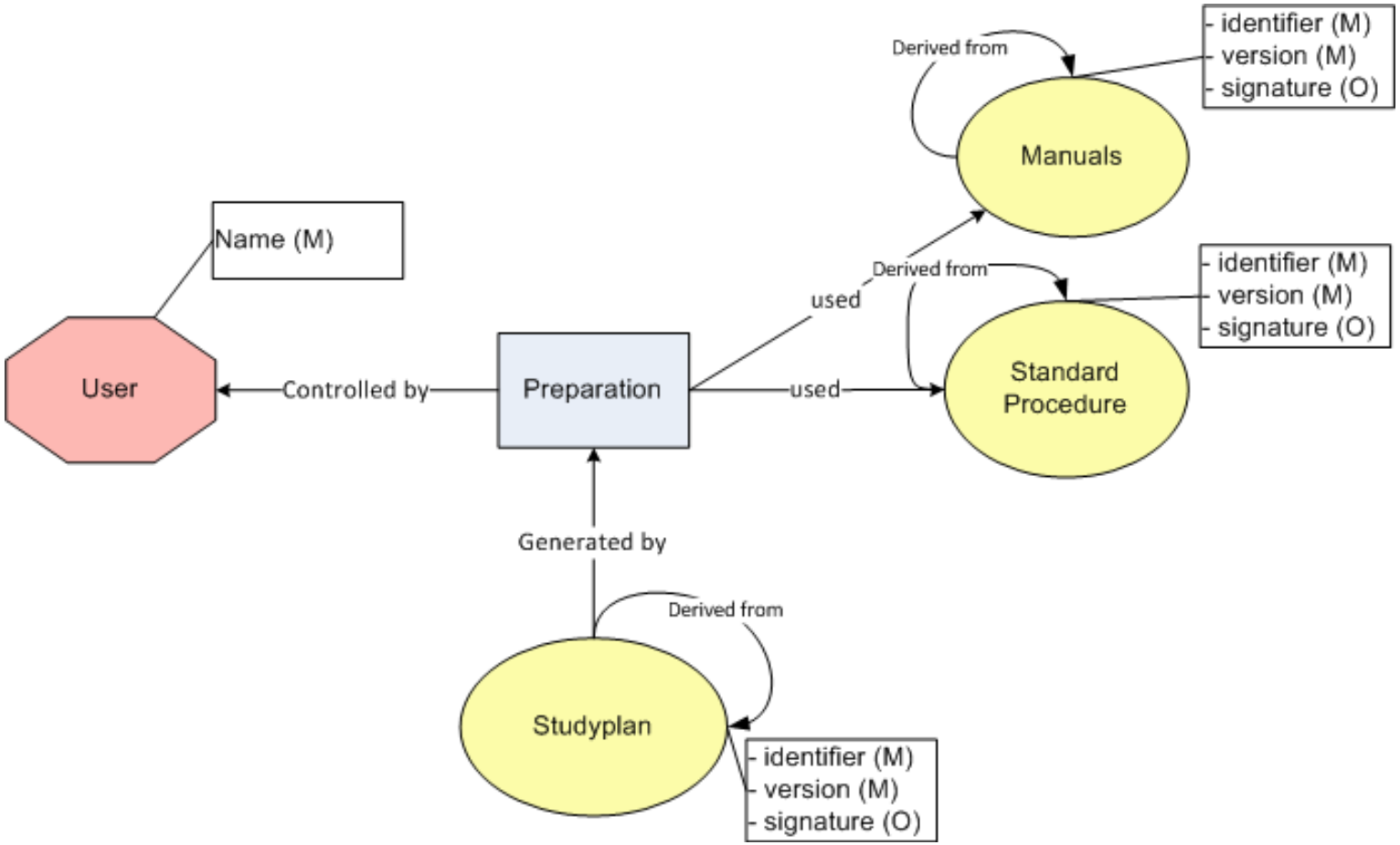
1. Sammeln von Fragen, welche beantwortet werden sollen
   - Wer ist verantwortlich für die Implementierung von X?
   - Welches Elememt ist der logische Vorgänger von Element X?

2. Identifikation der Aktoren, des Input und des Output für die Fragen.

3. Ermittlung der beteiligten Prozesse

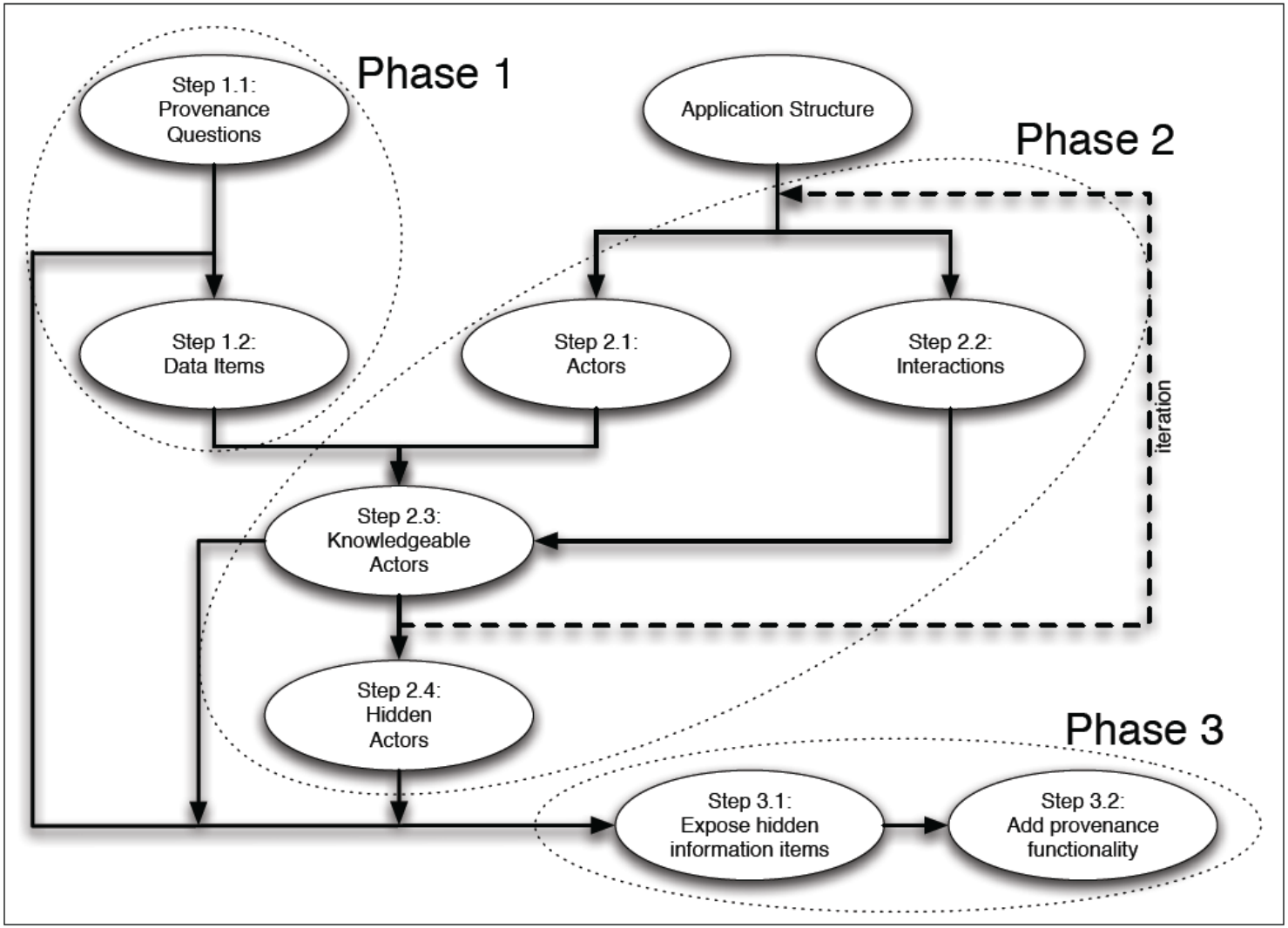
4. Entwicklung eine Provenance-Modells für die verschiedenen Prozesse

Beispielmodell
Laborbuch für Studien



Methodik „PrIMe“

Anwendungen „Provenance-Aware“ machen



Implementierungen

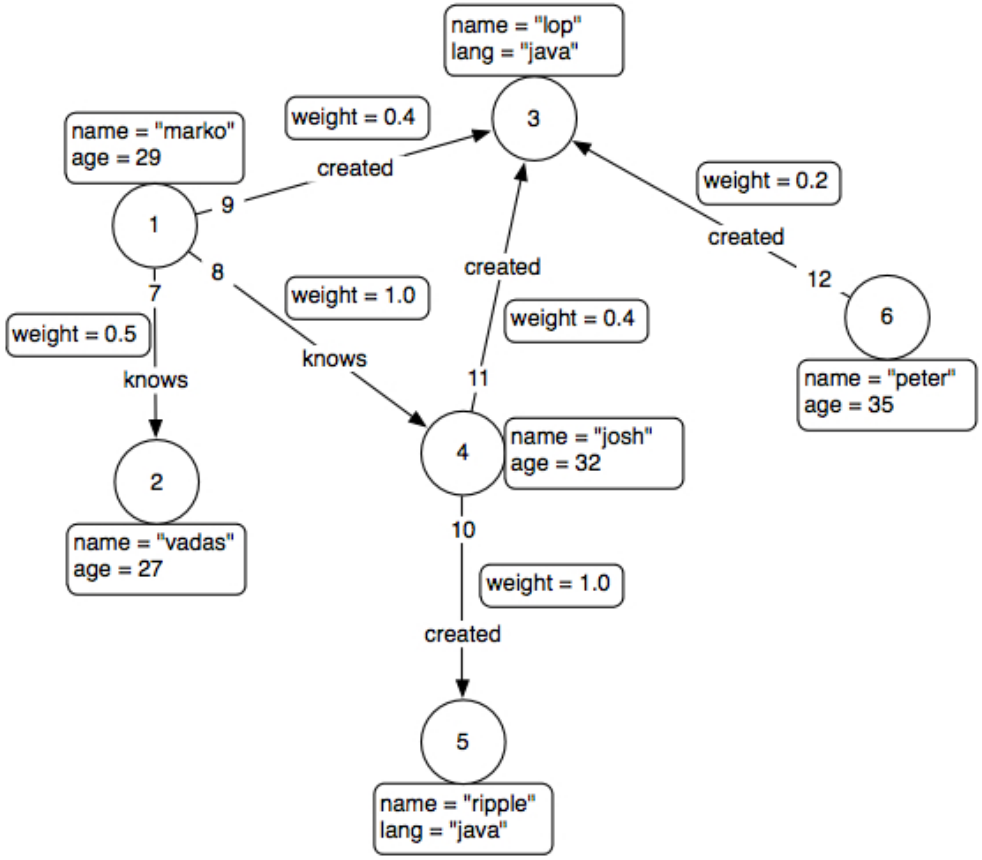
Implementierungsmöglichkeiten

Viele Technologien für Speicherung und Abfrage…

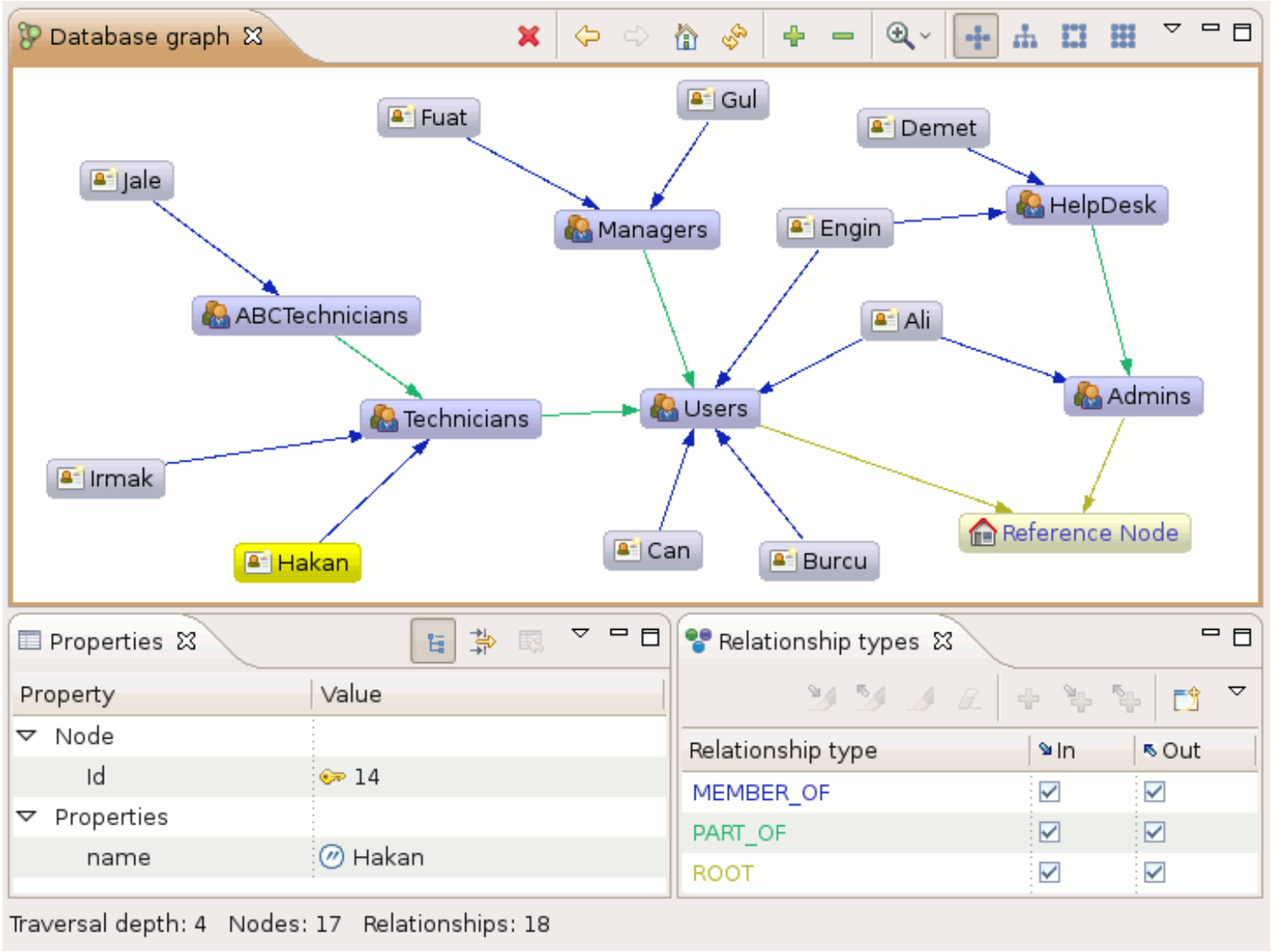
- **Relational**:
  - Relationale Datenbanken
  - Speicherung in Tabellen mit Zeilen/Spalten; Relationen zwischen Tabellen
  - Einfügen und Abfrage von Daten durch SQL
- **XML and XPath**:
  - Format für Datenaustausch in Form einer Metasprache für strukturieren hierarchischen Text
  - Abfrage durch XPath
- **RDF and SPARQL**:
  - Informationen sind als Graph strukturiert und nicht hierarchisch
  - Abfrage durch SPARQL
- **Semistructured**:
  - System von Objekten mit Attributen und Verbindungen zwischen ihnen, ohne formale zugrundeliegende Struktur
  - Kann durch sehr unertschiedliche Technologien realisiert werden, z.B. Objekte einer Programmiersprache und passende Abfragesprachen wie LINQ

Graph-Datenbanken

- Zum Speichern der Provenance Graphen bieten sich Graph-Datenbanken an
- Beispiel: **Neo4j** (http://neo4j.org/)
  - Open-Source
  - Implementiert in Java
  - Erlaubt die Speicherung in Form von *property graphs* (key-value-basiert, gerichtet und multi-relational)



Neo4j Eclipse Plug-In Neoclipse



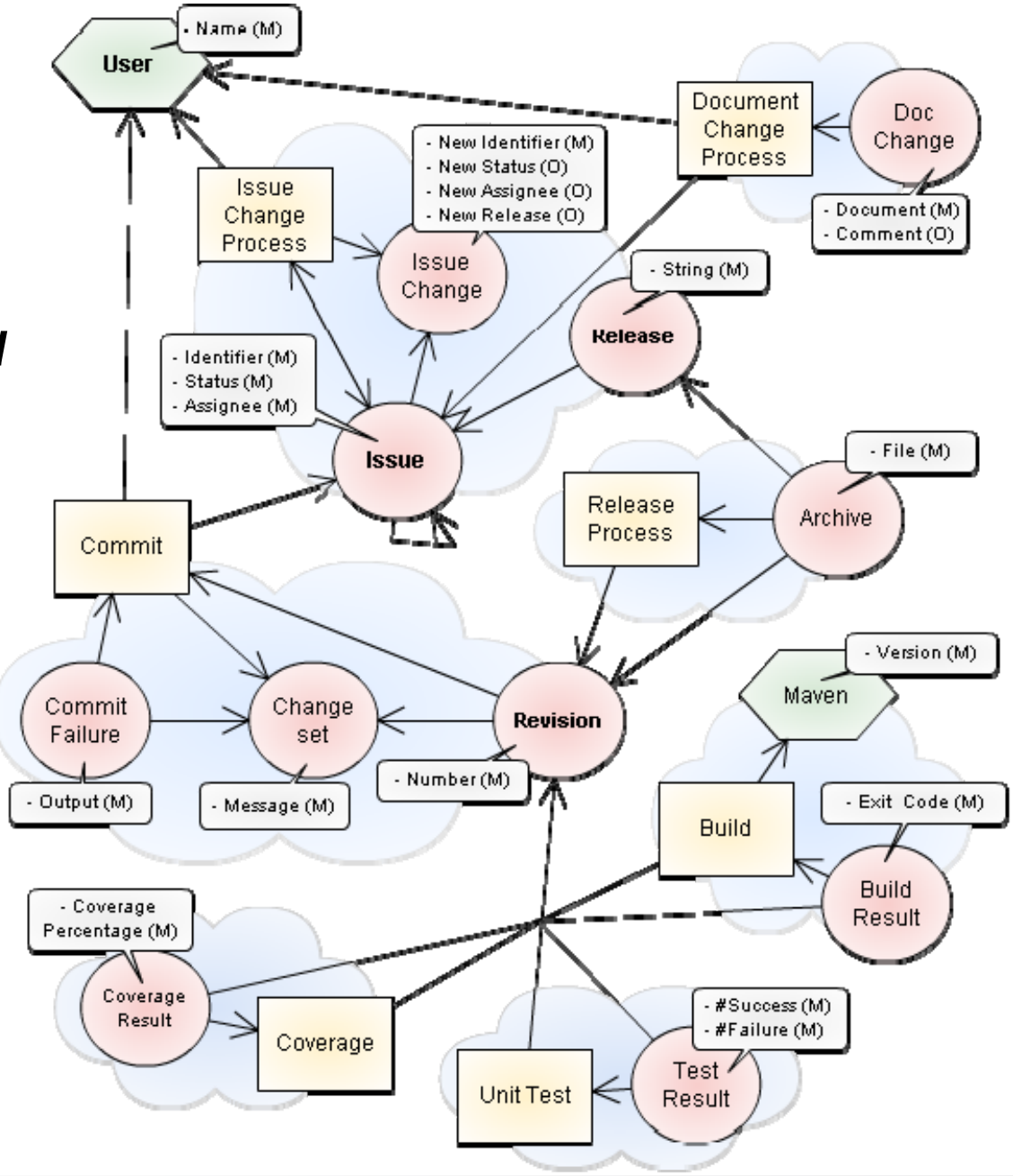
Abfrage und Analyse

- Graph-basierte Abfragesprechen
- Beispiel: **Gremlin** (https://github.com/tinkerpop/gremlin/wiki)
  - Unabhängig von der Graph-Datenbank
  - Abfrage über Kommandozeile oder Java API

Beispiel-Prozess
Software Engineering

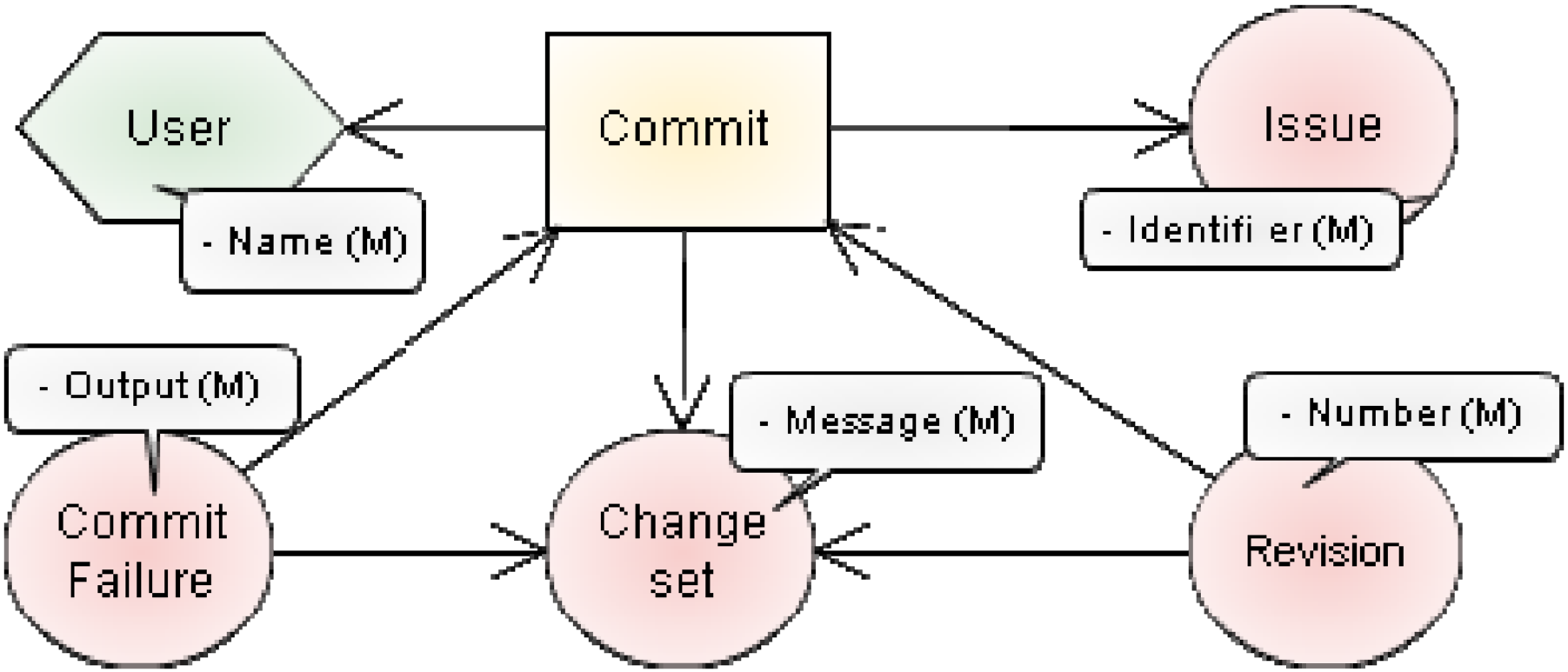
Frage:

- ***How many commits did developer X contribute to release Y?***

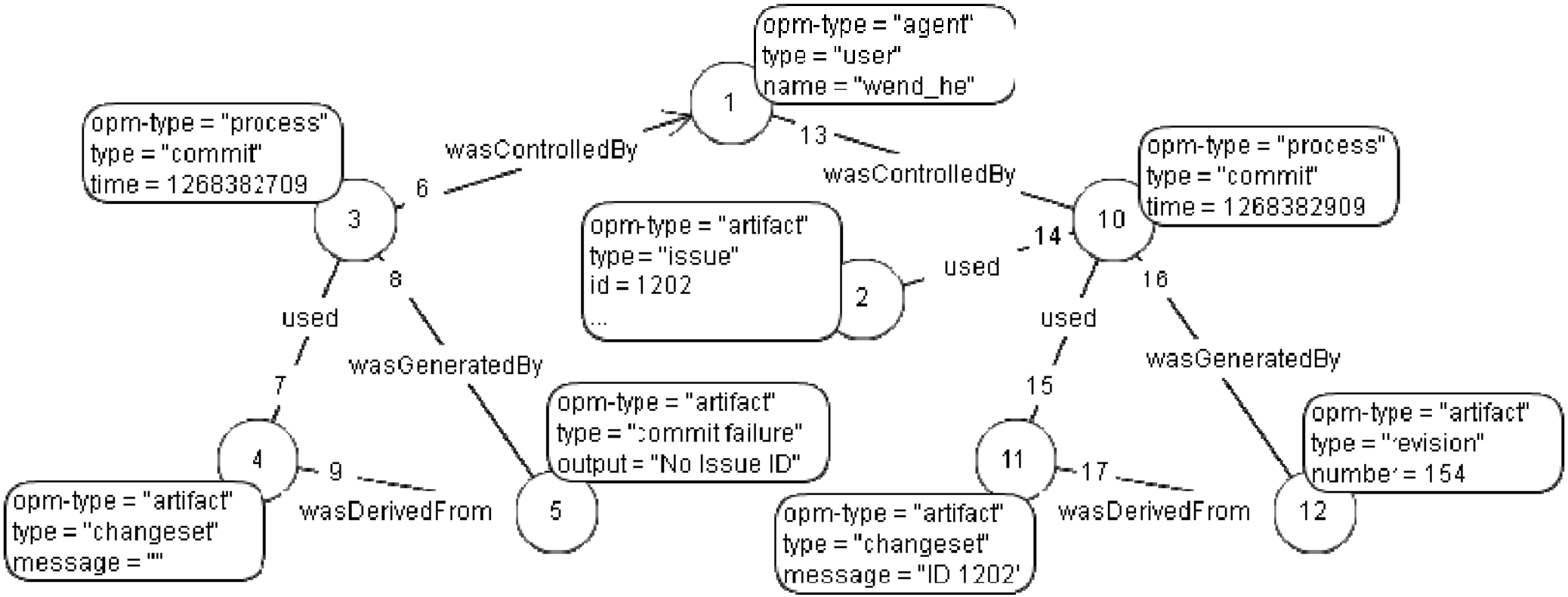


Teil-Prozess

„commit“



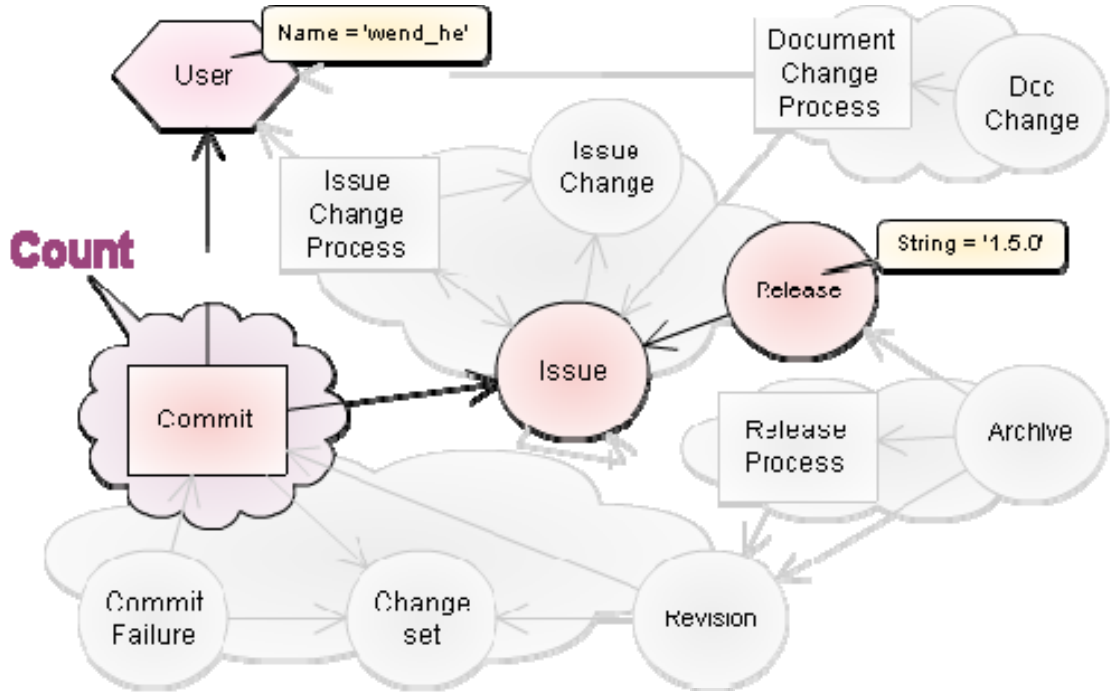
Prozess „commit“
Abbildung auf Neo4j

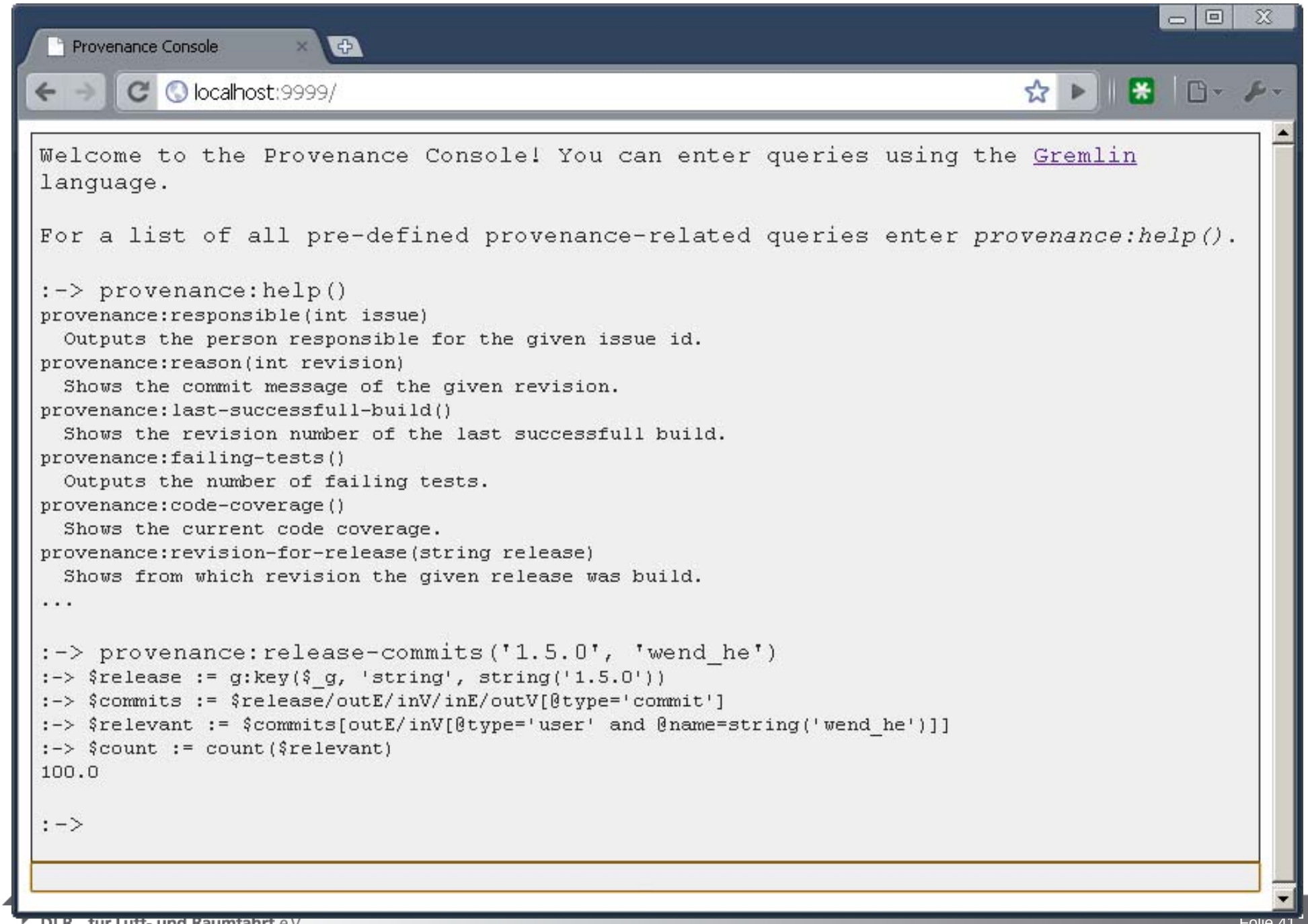


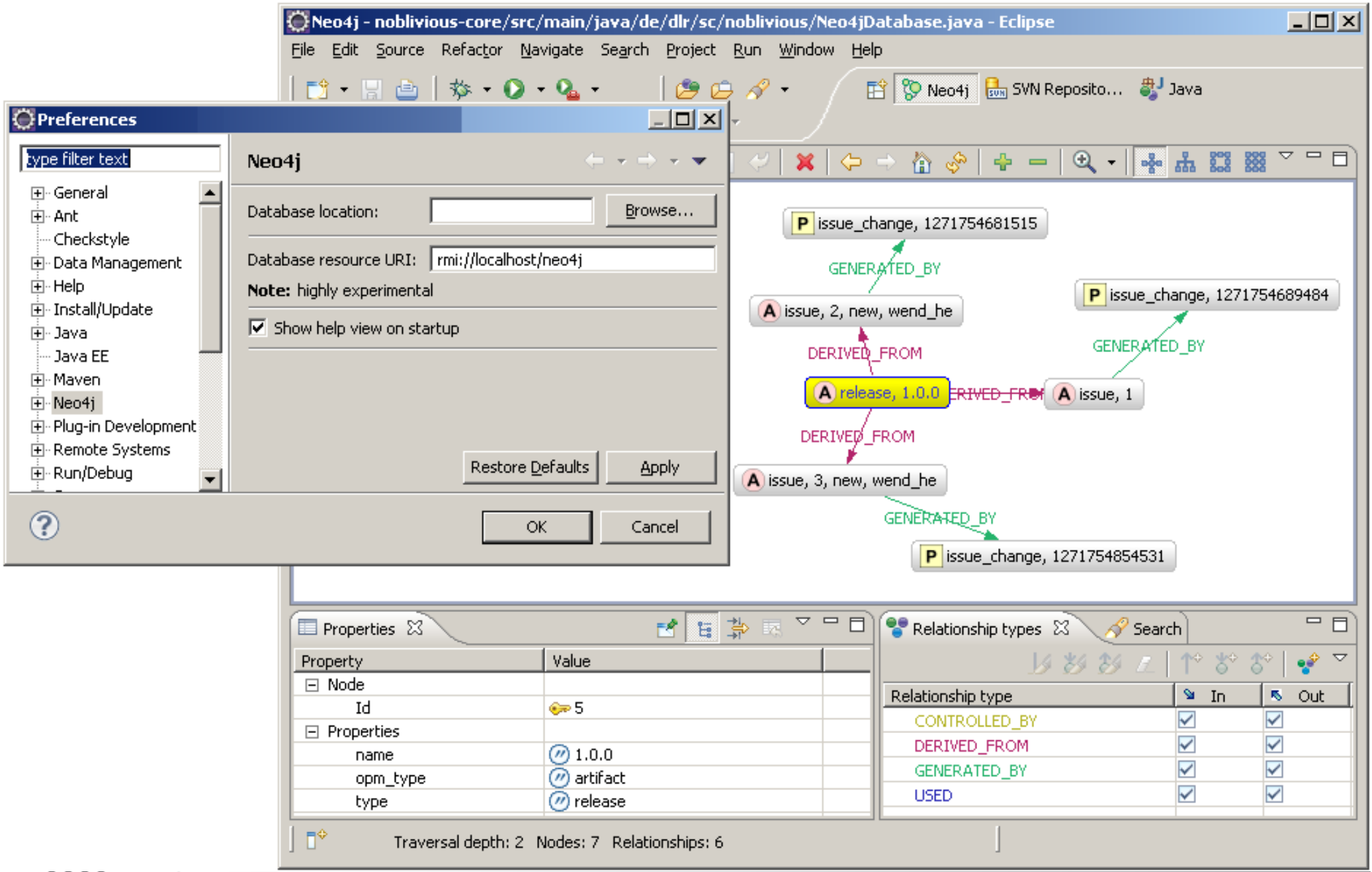
Beispiel-Abfrage und Visualisierung

How many commits did developer X contr. to release Y?

```
$release := g : key ($_g , 'string' , string($release))
$commits := $release /outE/inV/inE/outV [@type='commit']
$relevant := $commits[outE/inV[@type='user' and @name=string($developer)]]
$count := count($relevant)
```

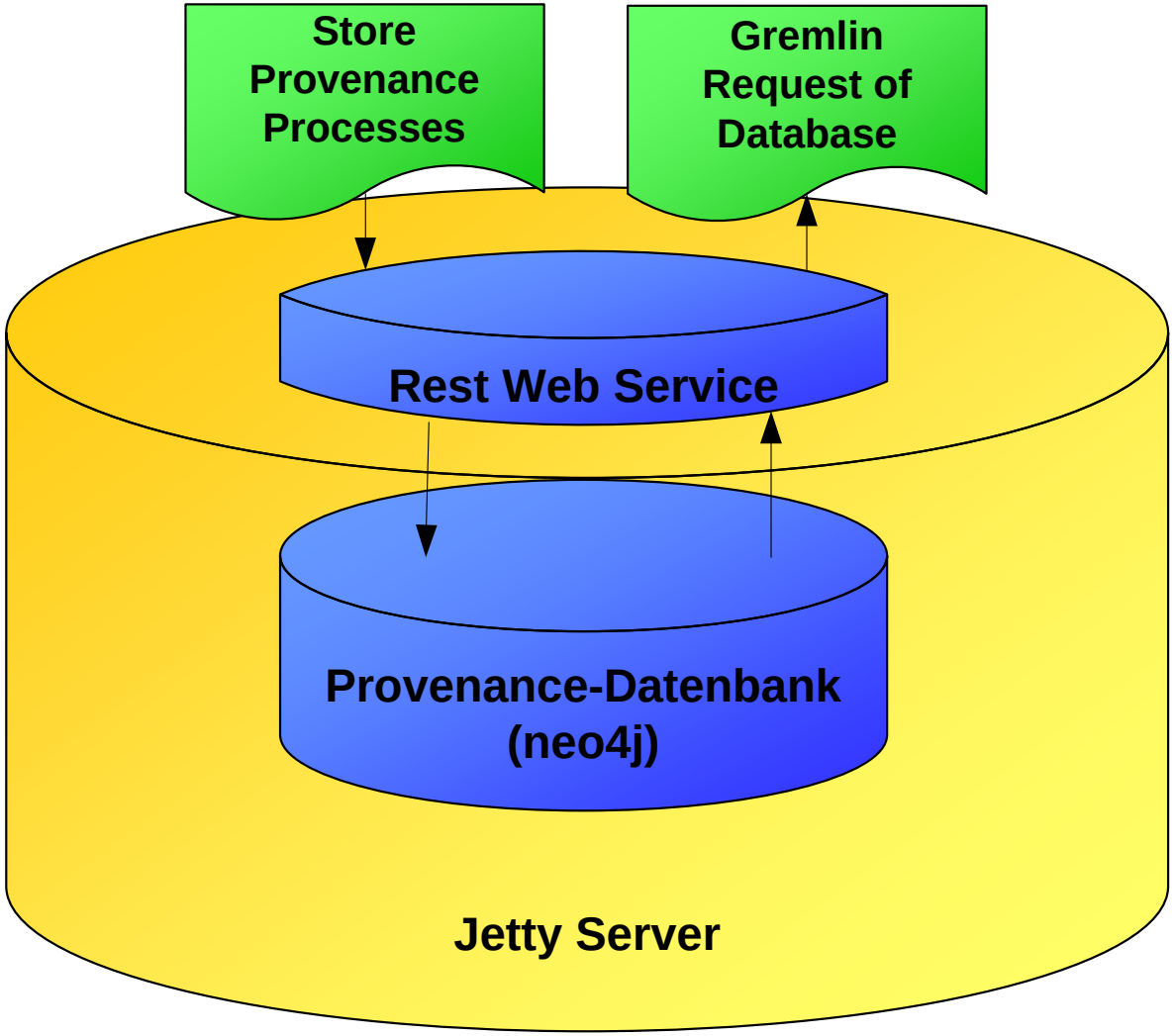



Provenance Console
localhost:9999/
Welcome to the Provenance Console! You can enter queries using the Gremlin language.
For a list of all pre-defined provenance-related queries enter provenance:help().
:-> provenance:help()
provenance:responsible(int issue)
Outputs the person responsible for the given issue id.
provenance:reason(int revision)
Shows the commit message of the given revision.
provenance:last-successfull-build()
Shows the revision number of the last successfull build.
provenance:failing-tests()
Outputs the number of failing tests.
provenance:code-coverage()
Shows the current code coverage.
provenance:revision-for-release(string release)
Shows from which revision the given release was build.
...
:-> provenance:release-commits('1.5.0', 'wend_he')
:-> $release := g:key($_g, 'string', string('1.5.0'))
:-> $commits := $release/outE/inV/inE/outV[@type='commit']
:-> $relevant := $commits[outE/inV[@type='user' and @name=string('wend_he')]]
:-> $count := count($relevant)
100.0
:->


Neo4j - noblivious-core/src/main/java/de/dlr/sc/noblivious/Neo4jDatabase.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Neo4j
SVN Reposito...
Java
Preferences
type filter text
General
Ant
Checkstyle
Data Management
Help
Install/Update
Java
Java EE
Maven
Neo4j
Plug-in Development
Remote Systems
Run/Debug
Neo4j
Database location:
Browse...
Database resource URI:
rmi://localhost/neo4j
Note: highly experimental
Show help view on startup
Restore Defaults
Apply
OK
Cancel
P issue_change, 1271754681515
GENERATED_BY
A issue, 2, new, wend_he
DERIVED_FROM
P issue_change, 1271754689484
GENERATED_BY
A release, 1.0.0
A issue, 1
DERIVED_FROM
A issue, 3, new, wend_he
GENERATED_BY
P issue_change, 1271754854531
Properties
Property
Value
Node
Id
5
Properties
name
1.0.0
opm_type
artifact
type
release
Relationship types
Search
Relationship type
In
Out
CONTROLLED_BY
DERIVED_FROM
GENERATED_BY
USED
Traversal depth: 2 Nodes: 7 Relationships: 6

Provenance-Service

REST-API zur einfachen Nutzung in Anwendungen



Ausblick

Weitere Themen

Visualisierung

- Standard-Visualisierung
- Produkt VisTrails

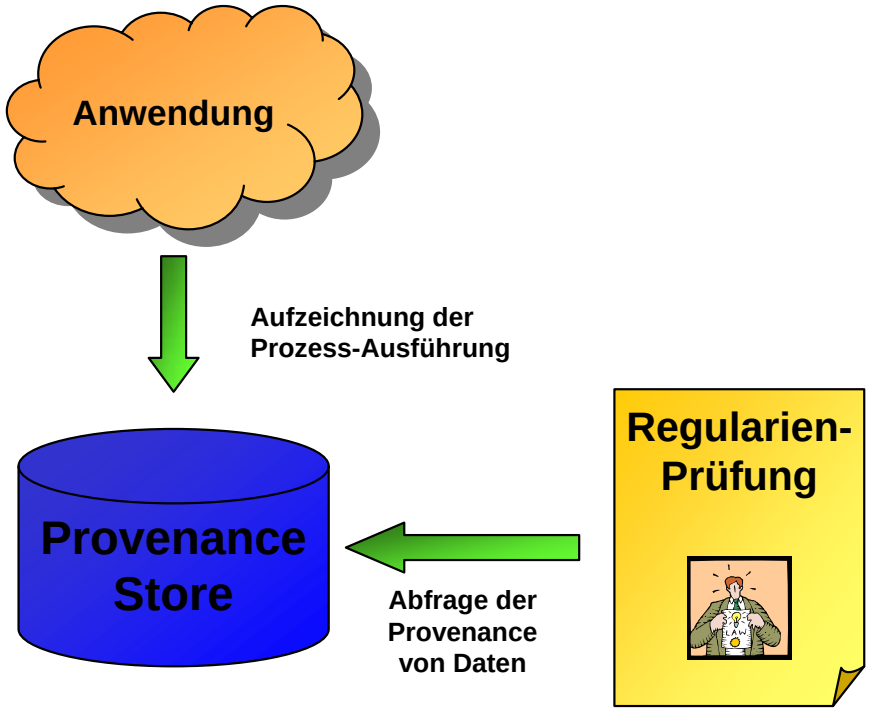
Security

- Zugriffskontrolle, Authentifizierung, Datenintegrität

Skalierbarkeit

- Provenance-Aufzeichnung muss genauso skalierbar sein wie die Anwendungen selber

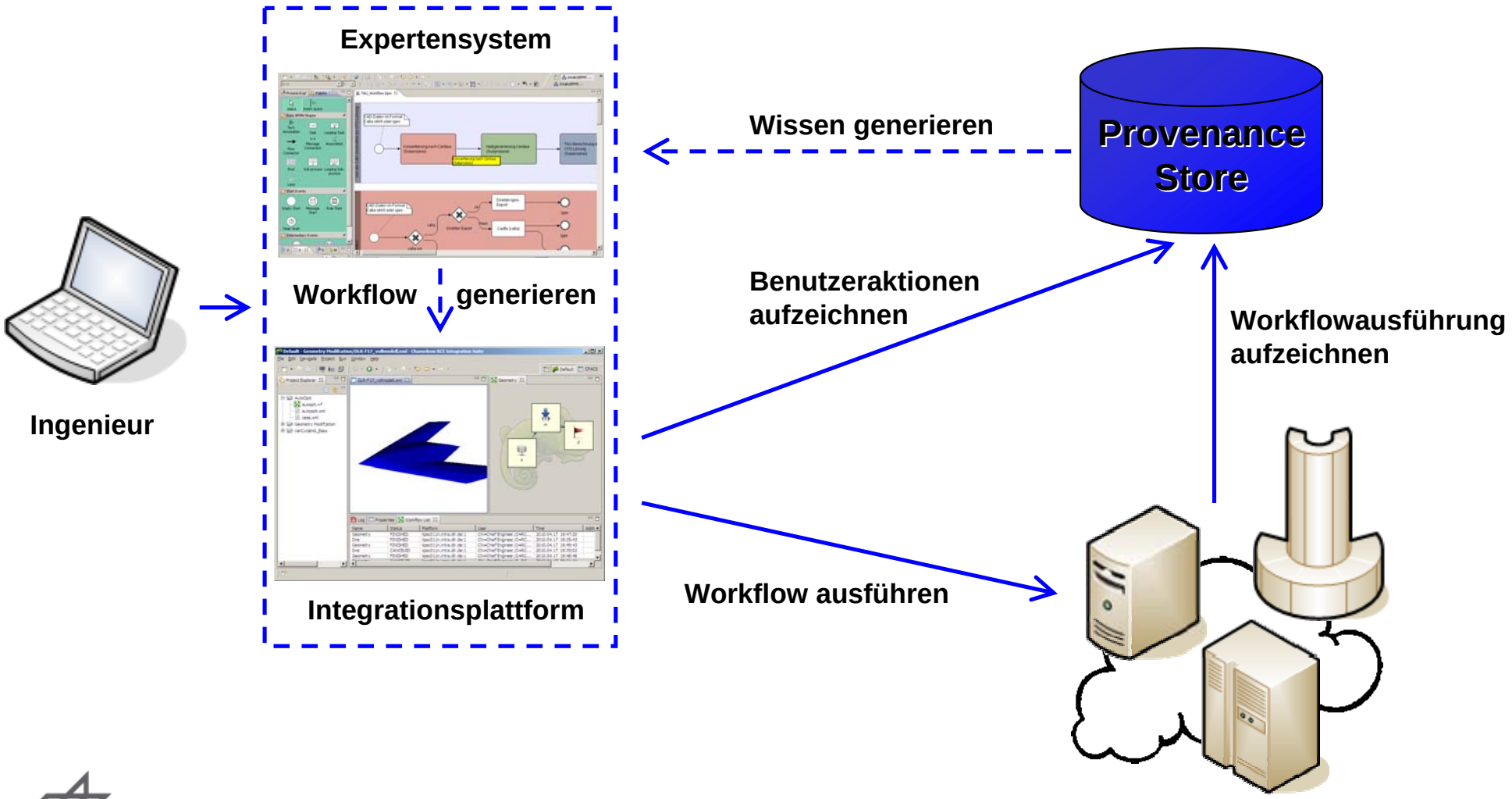
Compliance Oriented Architecture
Einhaltung von Regularien



- Erlaubt Applikations-übergreifende Überprüfung von Regularien
- Auch über mehrere Einrichtungen (z.B. bei Unteraufträgen, Outsourcing, Projekten)
- Geeignet für wiss. Peer-Review (e-Science) und Verifikation von Business Workflows

Einbettung in Arbeits-Umgebung

Integration mit weiteren Tools



Credits & Informationsquellen

Einzelne Slides bekommen, geklaut und inspiriert von

- Luc Moreau (University of Southampton) + Team
- Paul Groth (VU University Amsterdam)
- John Ibbotson (IBM UK)
- Guy K. Kloß (Auckland University of Technology)
- Miriam Ney (DLR, Berlin)
- Doreen Seider (DLR, Köln-Porz)
- Heinrich Wendel (Microsoft Deutschland)

Weitere Informationen & Software

- **`http://www.gridprovenance.org/`**
- **`http://openprovenance.org/`**
- **`http://www.ipaw.info`**


Fragen?