Provenance: concepts, architecture and envisioned tools

Professor Luc Moreau
L.Moreau@ecs.soton.ac.uk
University of Southampton
www.gridprovenance.org

Provenance Team

- University of Southampton
  - Luc Moreau, Victor Tan, Paul Groth, Simon Miles, Luc Moreau
- IBM UK (Project Coordinator)
  - John Ibbotson, Neil Hardman, Alexis Biller
- University of Wales, Cardiff
  - Omer Rana, Arnaud Contes, Vikas Deora
- Universitad Politecnica de Catalunya (UPC)
  - Steven Willmott, Javier Vazquez
- SZTAKI
  - Laszlo Varga, Arpad Andics
- German Aerospace
  - Andreas Schreiber, Guy Kloss, Frank Danneman

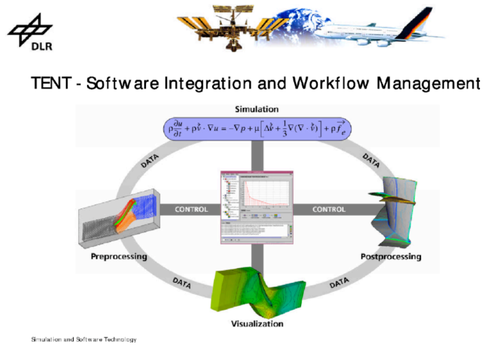
Overview

- Context
- Provenance Concepts & Definitions
- Architectural Design
- Provenance tools
- Conclusions

Context:
Importance of Past Processes

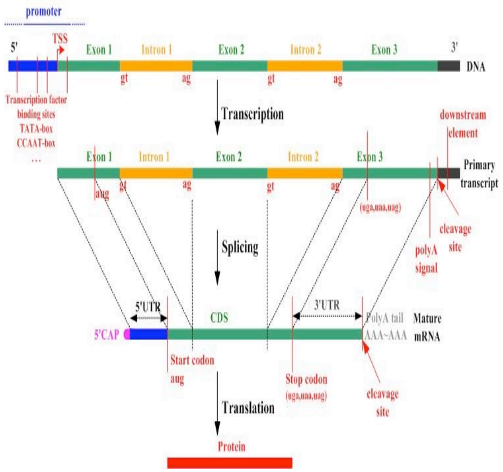
Context (1)

Aerospace engineering: maintain a historical record of design processes, up to 99 years.

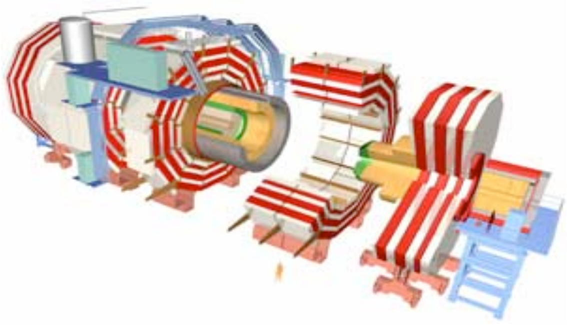

DLR
TENT - Software Integration and Workflow Management
Simulation
DATA
CONTROL
CONTROL
DATA
Preprocessing
Postprocessing
DATA
DATA
Visualization
Simulation and Software Technology

Organ transplant management: tracking of previous decisions, crucial to maximise the efficiency in matching and recovery rate of patients

Context (2)



Bioinformatics: verification and auditing of "experiments" (e.g. for drug approval)



High Energy Physics: tracking, analysing, verifying data sets in the ATLAS Experiment of the Large Hadron Collider (CERN)

Concepts & Definitions

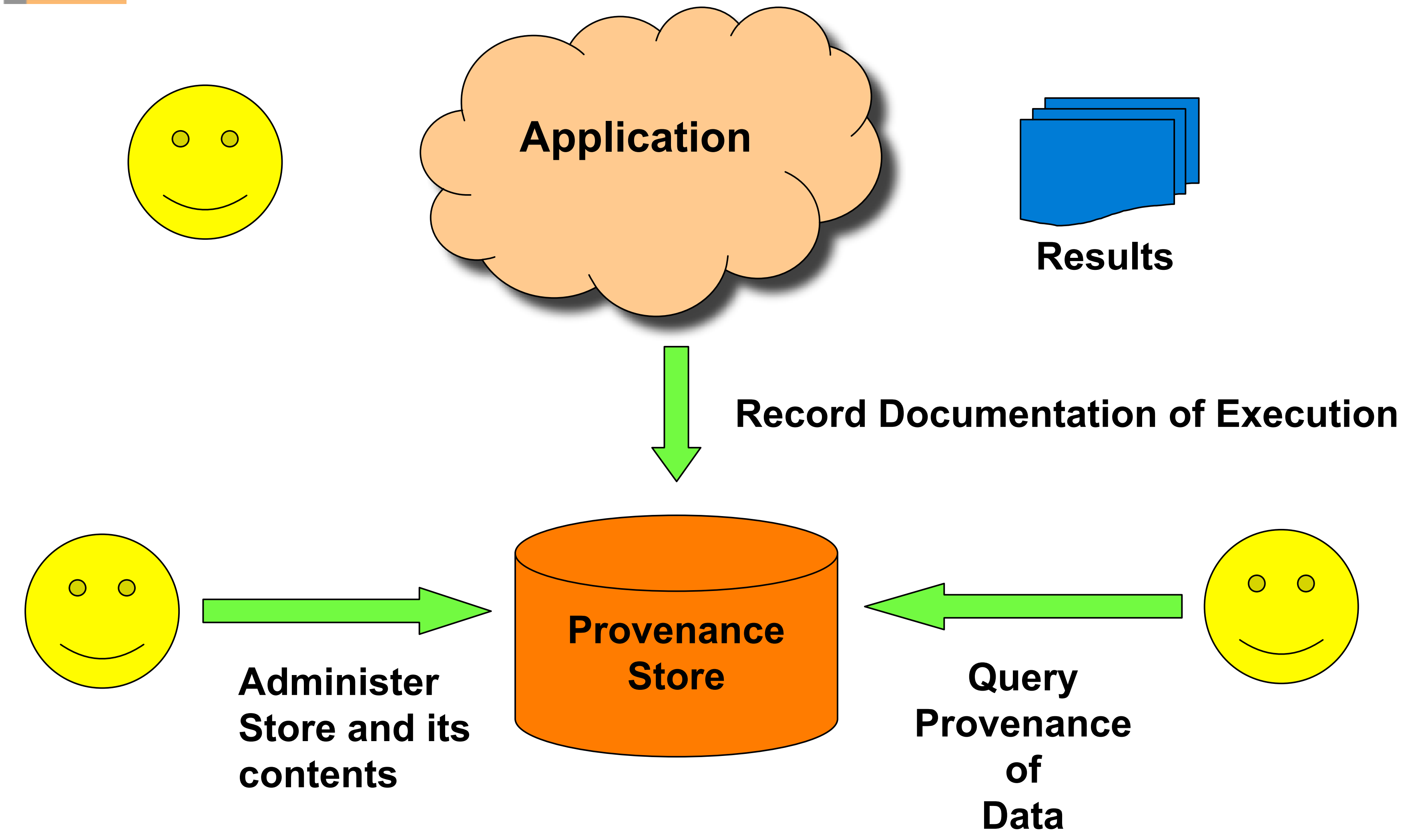
Provenance:
dictionary definition

- Oxford English Dictionary:
  - the fact of coming from some particular source or quarter; origin, derivation
  - the history or pedigree of a work of art, manuscript, rare book, etc.; concretely, a record of the ultimate derivation and passage of an item through its various owners.
- Concept vs representation

Provenance Definition

- Our definition of provenance in the context of applications for which process matters to end users:
  - The provenance of a piece of data is the process that led to that piece of data
- Our aim is to conceive a computer-based *representation* of provenance that allows us to perform useful analysis and reasoning to support our use cases

Core Interfaces to
Provenance “Lifecycle”
Provenance Store



Application

Results

Record Documentation of Execution

Provenance Store

Administer Store and its contents

Query Provenance of Data

Nature of Documentation

- We represent the provenance of some data by *documenting* the process that led to the data:
  - documentation can be complete or partial;
  - it can be accurate or inaccurate;
  - it can present conflicting or consensual views of the actors involved;
  - it can provide operational details of execution or it can be abstract.

p-assertion

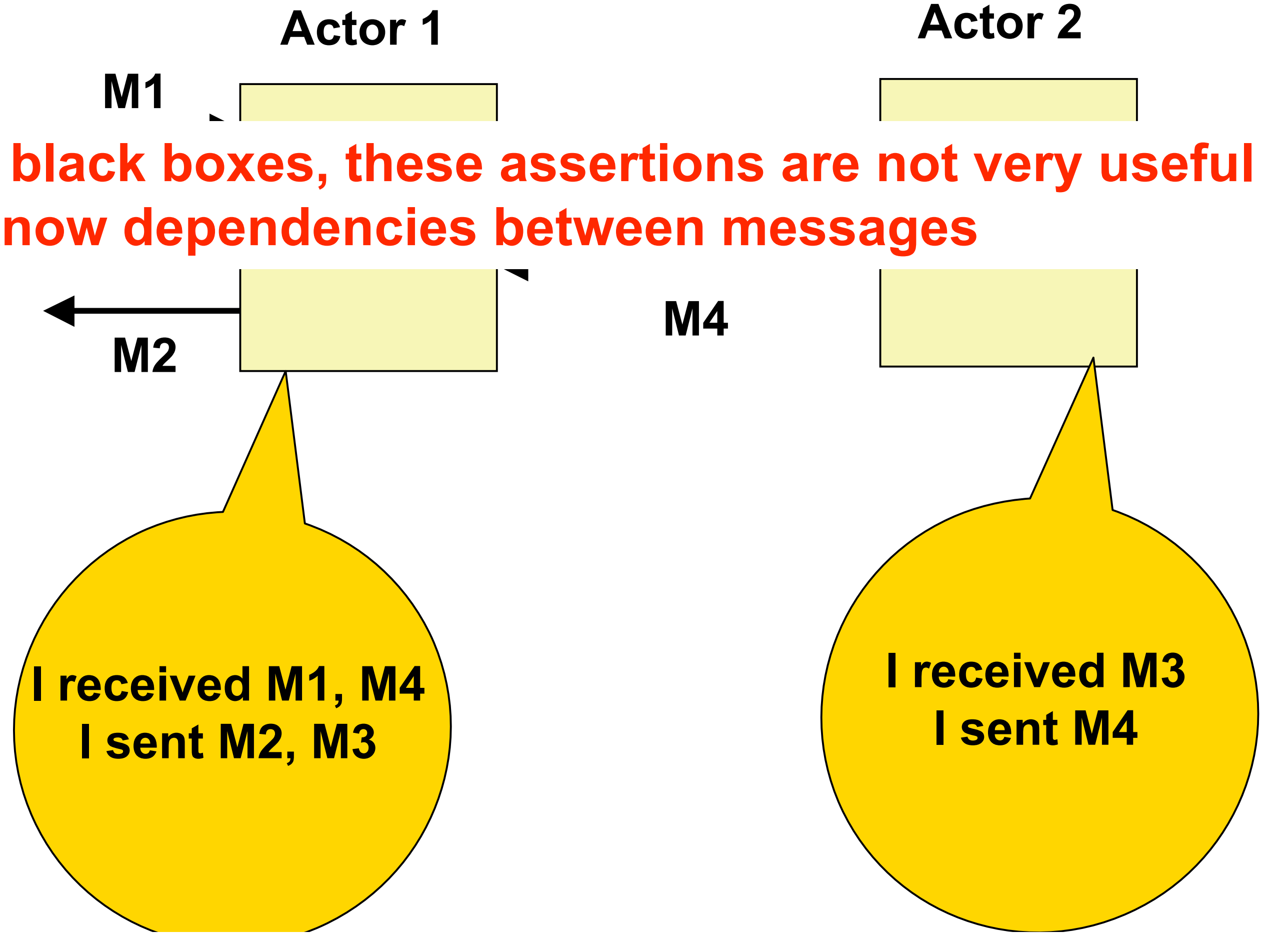
- A given element of process documentation will be referred to as a p-assertion
  - p-assertion: is an assertion that is made by an actor and pertains to a process.

Service Oriented Architecture

- Broad definition of service as component that takes some inputs and produces some outputs.
- Services are brought together to solve a given problem typically via a workflow definition that specifies their composition.
- Interactions with services take place with messages that are constructed according to services interface specification.
- The term actor denotes either a client or a service in a SOA.
- A process is defined as execution of a workflow

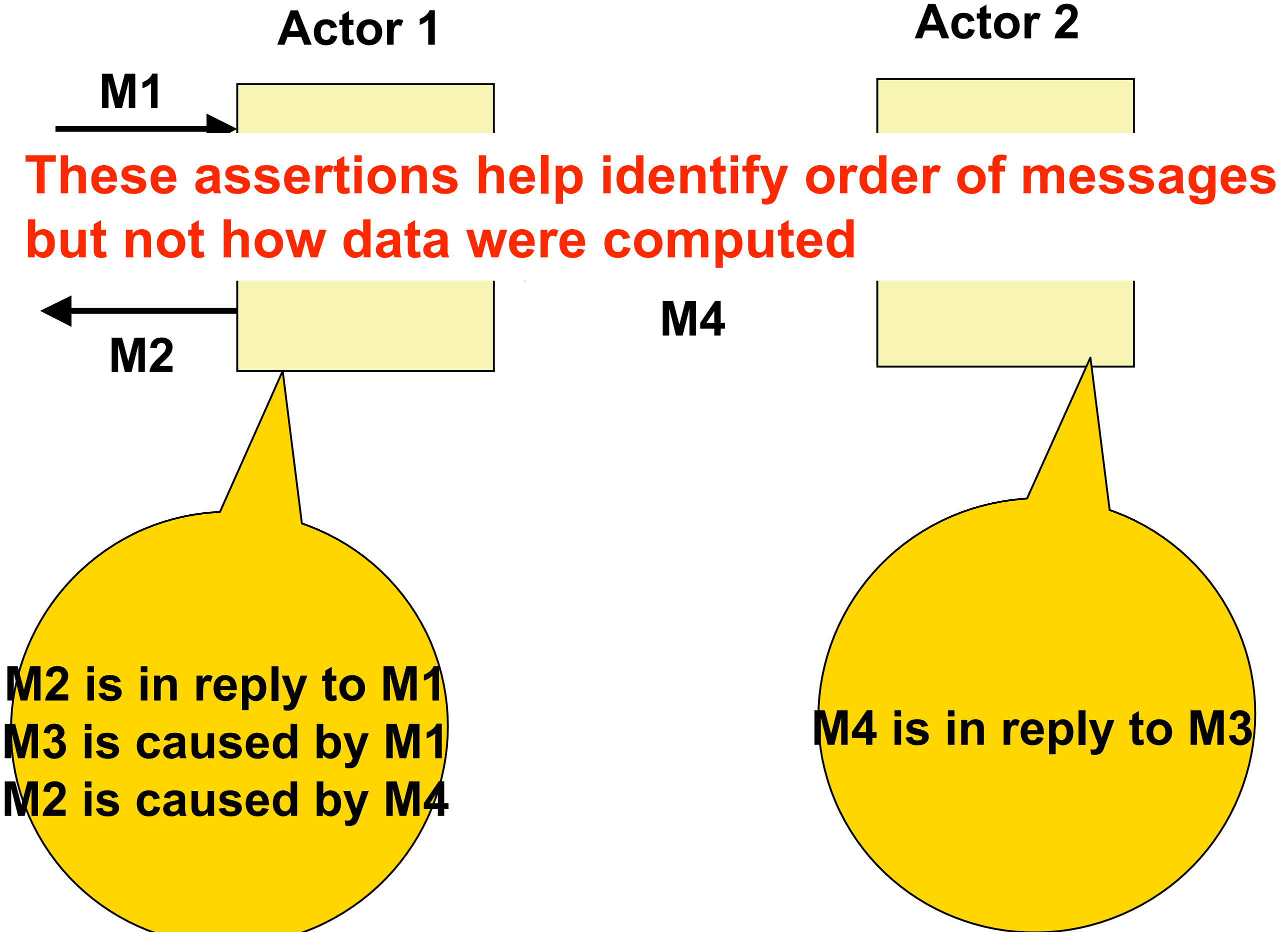
Process Documentation (1)

From these p-assertions, we can derive that M3 was sent by Actor 1 and received by Actor 2 (and likewise for M4)



If actors are black boxes, these assertions are not very useful because we do not know dependencies between messages

Process Documentation (2)

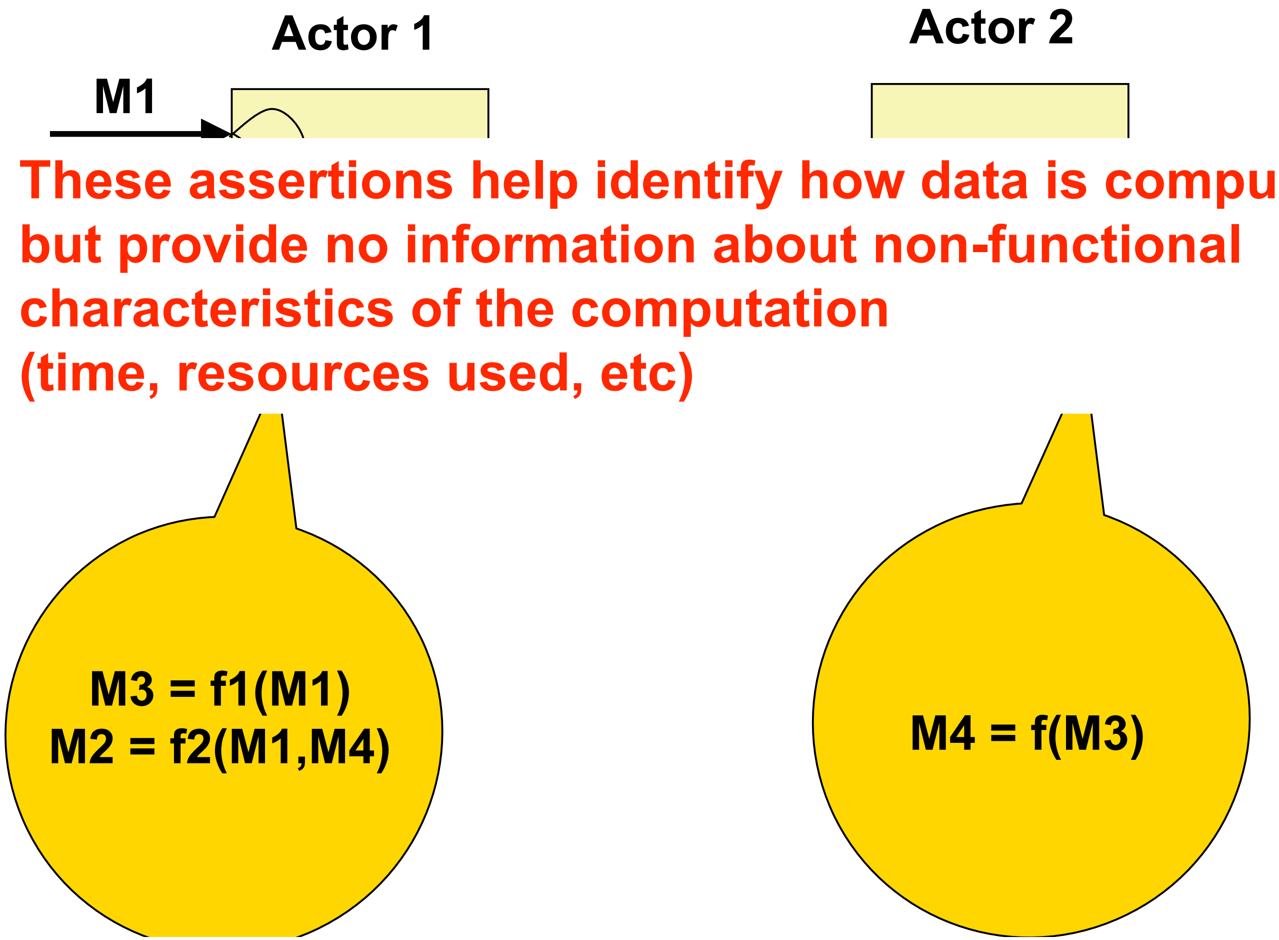


These assertions help identify order of messages, but not how data were computed

M2 is in reply to M1
M3 is caused by M1
M2 is caused by M4

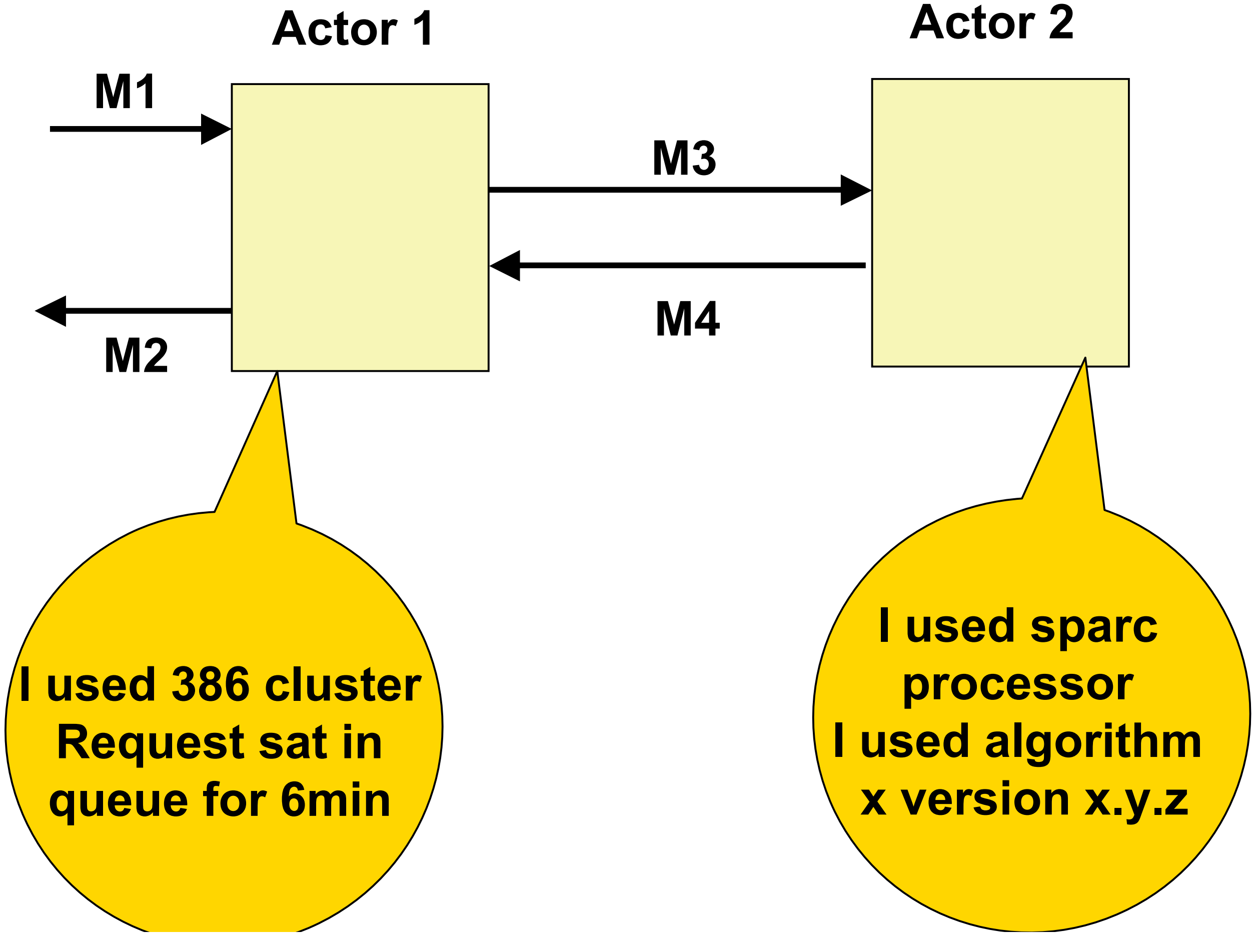
M4 is in reply to M3

Process Documentation (3)



These assertions help identify how data is computed, but provide no information about non-functional characteristics of the computation
(time, resources used, etc)

Process Documentation (4)


Actor 1
Actor 2
M1
M2
M3
M4
I used 386 cluster Request sat in queue for 6min
I used sparc processor I used algorithm x version x.y.z

Types of p-assertions (1)

- Interaction p-assertion: is an assertion of the contents of a message by an actor that has sent or received that message

I received M1, M4
I sent M2, M3

Types of p-assertions (2)

- Relationship p-assertion: is an assertion, made by an actor, that describes how the actor obtained output data or the whole message sent in an interaction by applying some function to input data or messages from other interactions.

M2 is in reply to M1
M3 is caused by M1
M2 is caused by M4

M3 = f1(M1)
M2 = f2(M1,M4)

Types of p-assertions (3)

- Actor state p-assertion: assertion made by an actor about its internal state in the context of a specific interaction

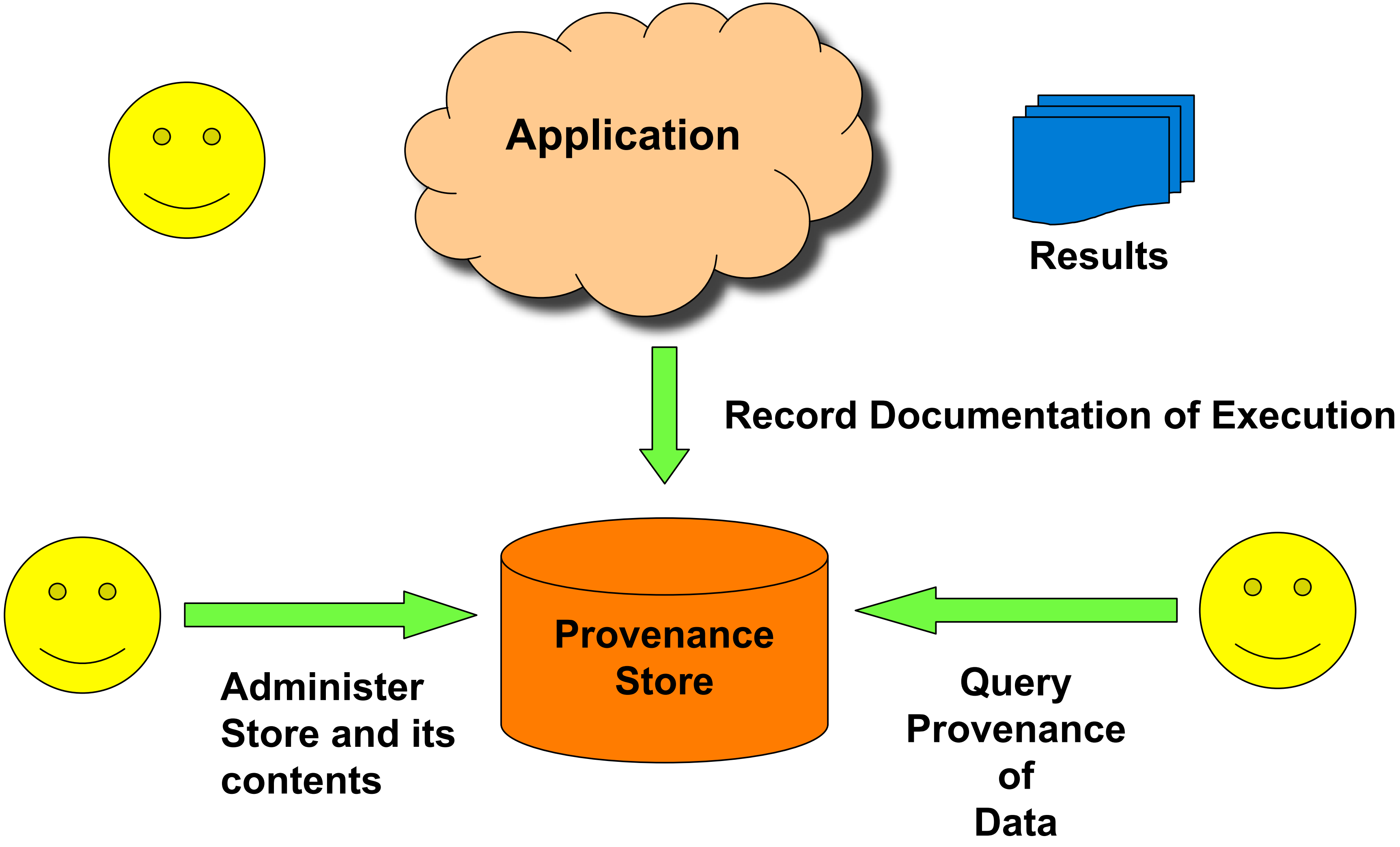
I used sparc processor
I used algorithm x version x.y.z

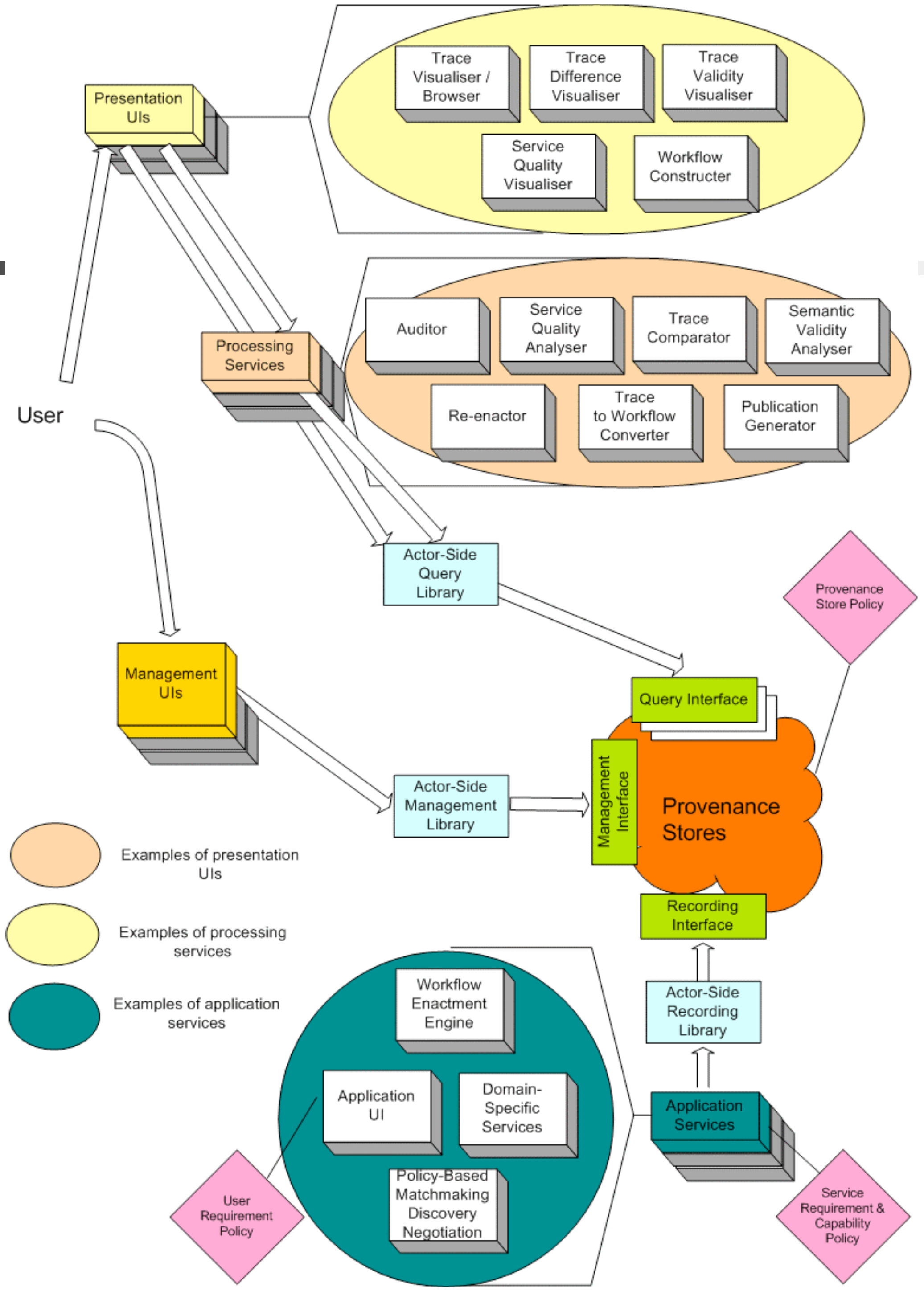
Data flow

- Interaction p-assertions allow us to specify a flow of data between actors
- Relationship p-assertions allow us to characterise the flow of data “inside” an actor
- Overall data flow (internal + external) constitutes a DAG, which characterises the process that led to a result

Architectural Design

Interfaces to Provenance Store


Application
Results
Record Documentation of Execution
Provenance Store
Administer Store and its contents
Query Provenance of Data



Presentation UIs

Trace Visualiser / Browser

Trace Difference Visualiser

Trace Validity Visualiser

Service Quality Visualiser

Workflow Constructer

Processing Services

Auditor

Service Quality Analyser

Trace Comparator

Semantic Validity Analyser

Re-enactor

Trace to Workflow Converter

Publication Generator

User

Actor-Side Query Library

Provenance Store Policy

Management UIs

Query Interface

Management Interface

Provenance Stores

Actor-Side Management Library

Recording Interface

Actor-Side Recording Library

Application Services

Examples of presentation UIs

Examples of processing services

Examples of application services

Workflow Enactment Engine

Application UI

Domain-Specific Services

Policy-Based Matchmaking Discovery Negotiation

User Requirement Policy

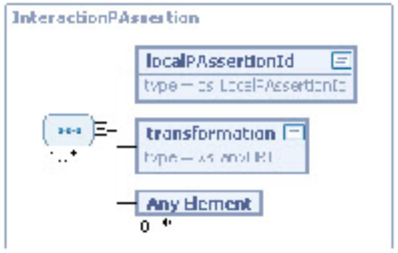
Service Requirement & Capability Policy

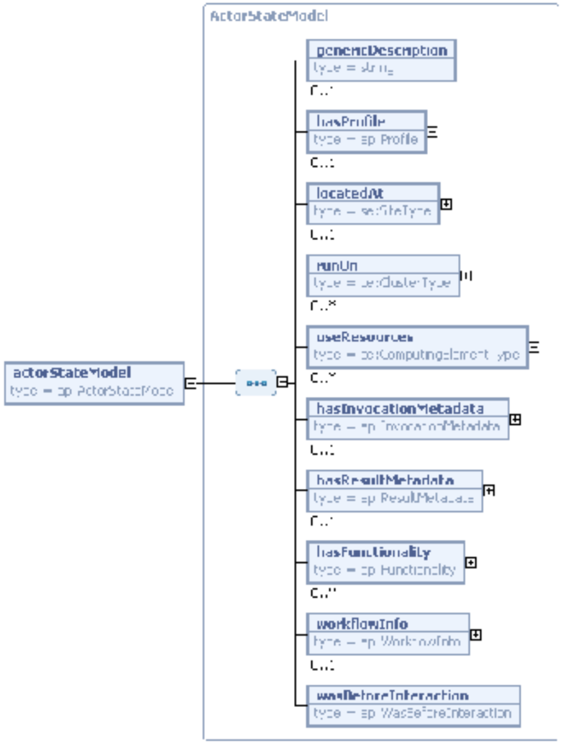
Provenance Tools

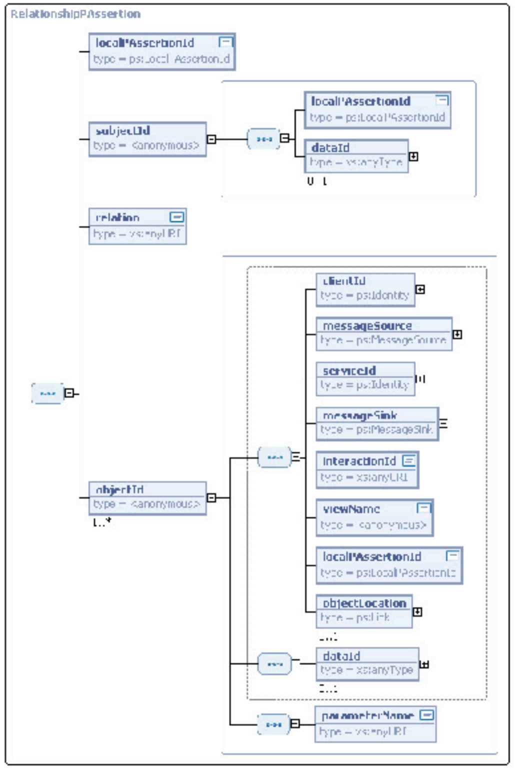
Provenance Tools

- Five core deliverables
  - Data model and schema
  - Provenance store
  - Client side libraries
  - Generic Provenance tools
  - Methodology

Provenance Modelling



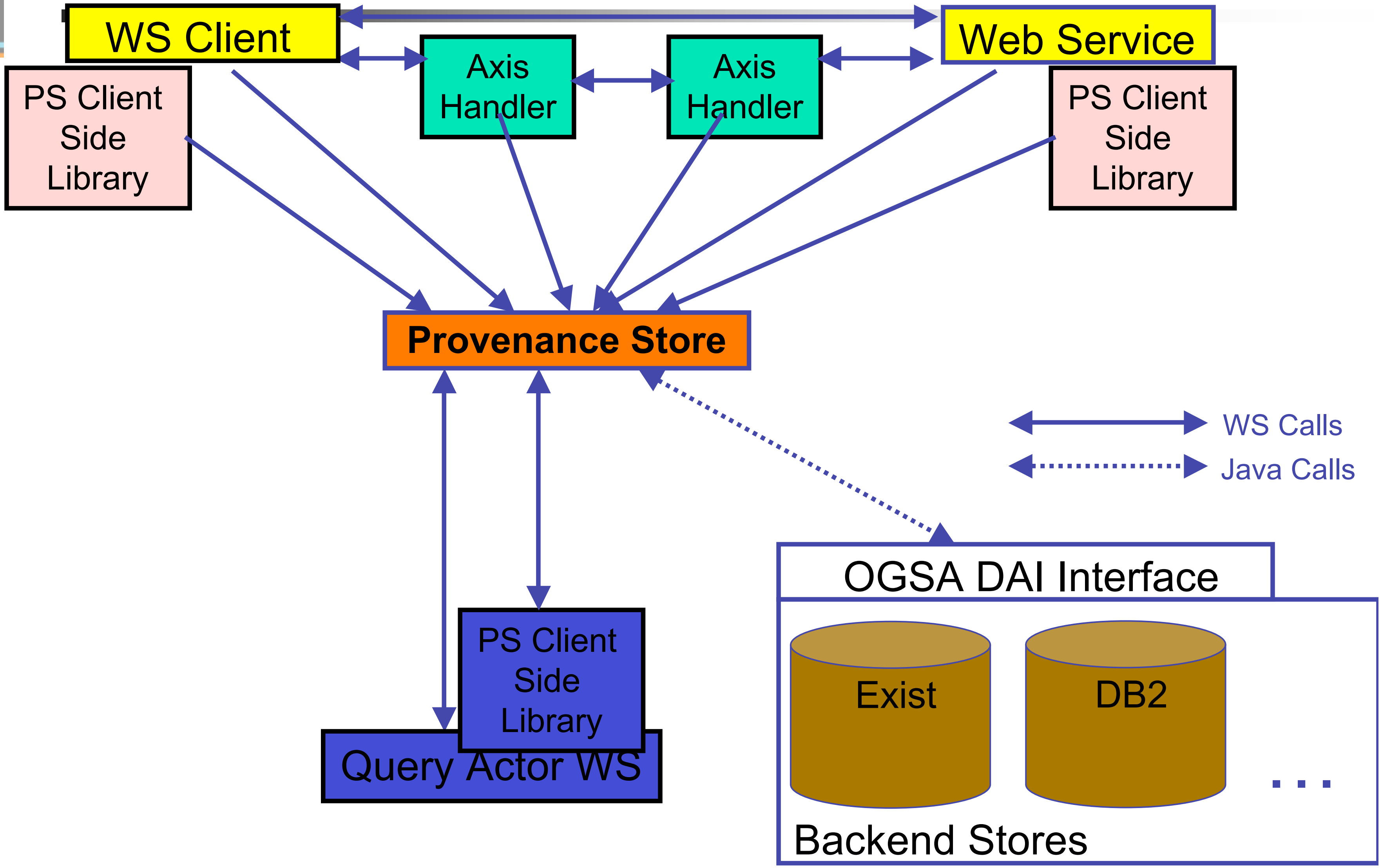




Provenance Store Reference Implementation

- Implementation of recording, querying and managing interface
- Provenance store implemented as a Web Service
- Client side libraries for using Provenance Store
- Axis Handler for automatically recording communication between Axis-based Web Services

Implementation Diagram



Implementation Details

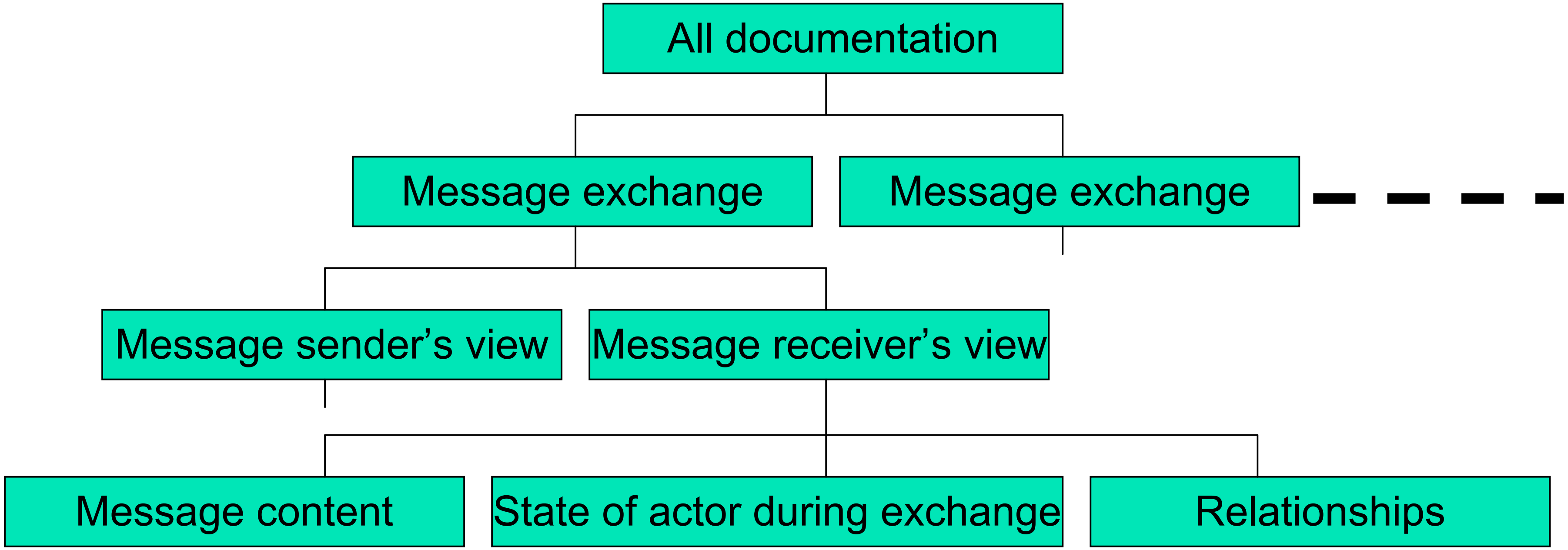
- Currently functional prototype is a pure Web Services solution (based on Tomcat/AXIS)
- Security will be based on WS-Security
- WSRF offers a number of interesting opportunities, and we are considering mapping the (technology-neutral) architecture on to a WSRF-oriented stack.

Query Interface

- Purpose
  - Obtain the provenance of some specific data
  - Allow for "navigation" of the documentation of execution
- Abstract interface
  - Allows us to view the provenance store *as if* containing XML data structures
  - Independent of technology used for running application and internal store representation
  - Seamless navigation of application dependent and application independent provenance representation

Structure of Documentation

- The documentation of processes recorded by actors can be categorised into a hierarchy



XML Query Languages

- Two existing query languages provide ways of navigating hierarchical data: XPath and XQuery
- For instance, we can use XPath to refer to:
  - The message exchange with ID 345
  - The client’s view of that exchange
  - The body of the message exchanged

```
// messageExchange [id=“345”]
    / clientView / messageContent
```

Navigating Message Content

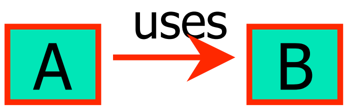
- If message content is in XML format, or can be mapped to it, then XPath and XQuery can be used to navigate into the message content
- For example, we can add application-specific navigation to the previous XPath:
  - The SOAP envelope that encloses the message
  - The body of the message within the envelope
  - The customer name within the body

```
// messageExchange [id="345"]
  / clientView / messageContent
    / soap:envelope / soap:body // customerName
```

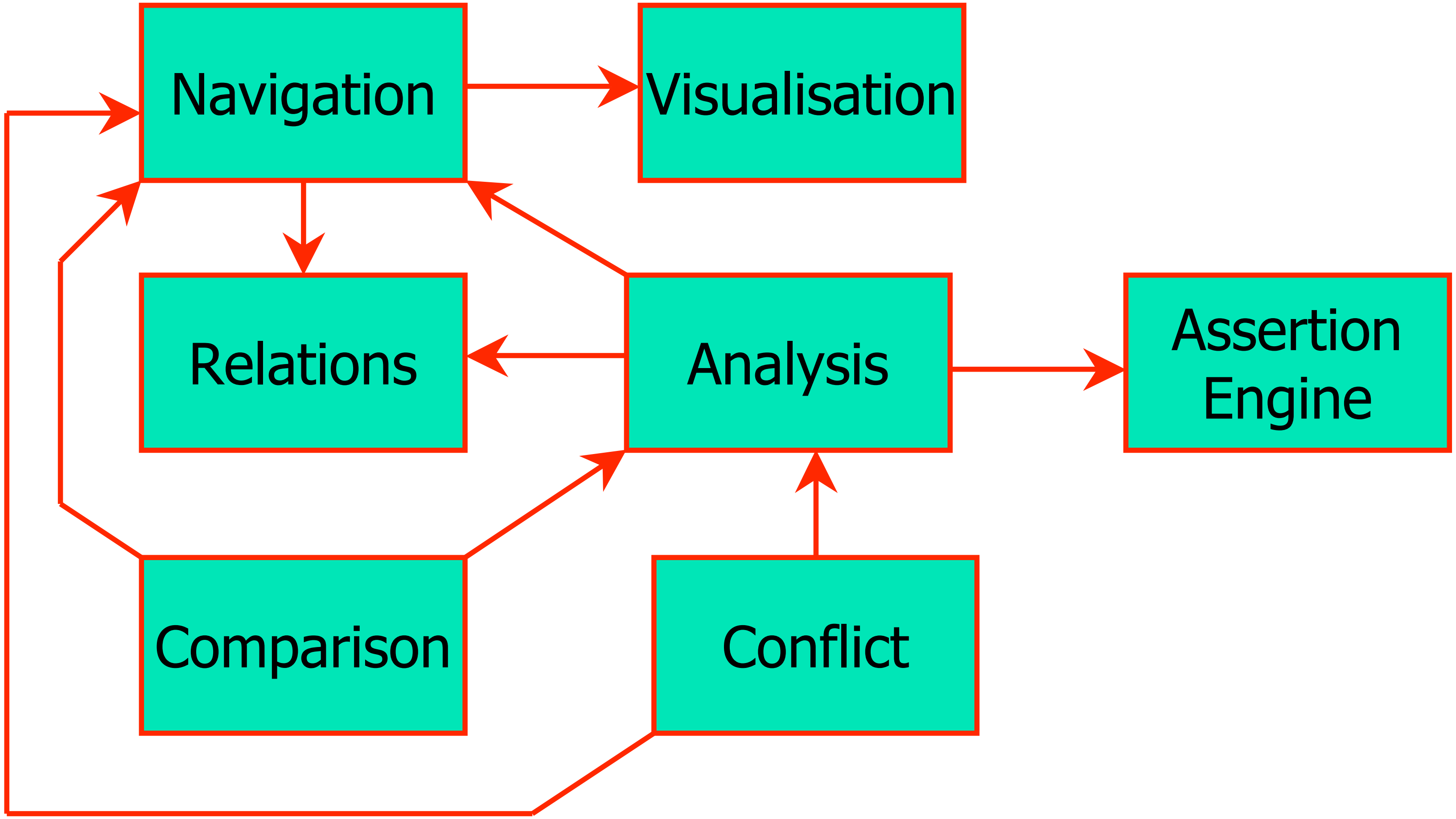
Other Query Requirements

- Execution Filtering: include/exclude all p-assertions that are marked as part of an execution by a single actor.
- Functionality Filtering: include/exclude p-assertions that have one of a given set of operation types.
- Process Filtering: include/exclude p-assertions that belong to a given (set of) process(es).

Generic Tools



A makes use of B



Generic Tools

- Analysis: constraint satisfaction over p-assertions and their content
- Comparison: comparison between assertions
- Conflict detection: detect conflicts between assertions
- Rule engine: verify that provenance of some data satisfy some constraints
- Visualisation: Implemented as a Portlet (using the eXo Portal Framework – JSR 168 compliant

Methodology

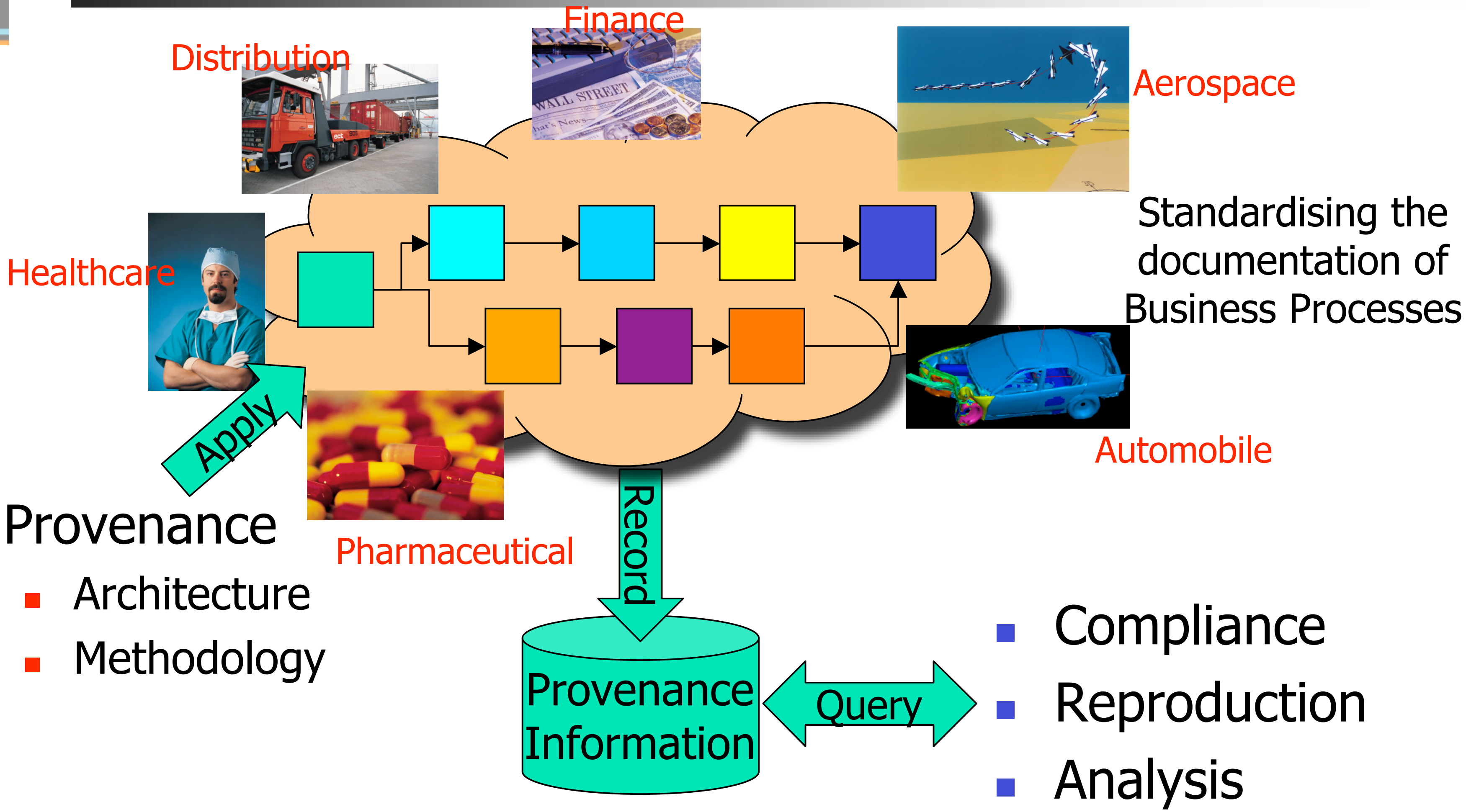
- How to design applications (whether legacy or new) so that they become provenance aware
- Sets of useful schema
- Guidelines on what to record

Key Deliverables

- NOW: First functional prototype
- NOW: Architecture (technology independent), first public version
- 04/06: Set of tools
- 04/06: Final Architecture
- 09/06: Web Service standardisation proposal
- 09/06: Full implementation, secure and scalable
- 09/06: Methodology: how to make your application provenance-aware

Conclusions

Applying Provenance



- Provenance
  - Architecture
  - Methodology

- Compliance
- Reproduction
- Analysis

Conclusions

- Mostly unexplored area that is crucial to develop trusted systems
- Definition of provenance
- Specification of provenance representation
- Architecture
- Tools
  - Data models
  - Provenance Store
  - Client side tools
  - Generic tools
  - Methodology

Conclusions

- Current work:
  - System and protocol designing, architecture specification, generic support for use cases
  - Pursue the deployment in concrete application and performance evaluation
  - Work towards a standardisation proposal
  - Methodology
- Software soon to be available
- Tell us about your use cases: we are keen to find new collaborations in this space!
- Download the architecture definition from www.gridprovenance.org