Provenance state-of-the-art

Vyacheslav Tykhonov
Senior Information Scientist
Research and Innovation department (DANS)

CLARIAH provenance workshop,
03.09.2018
Provenance introduction

• Why provenance information is important for reproducibility
  
  (Reproducible Science)

• Why do we need to keep any change in provenance

• What standards we have for provenance

• How to keep provenance information up-to-date for data and
Provenance and reproducibility in scientific workflows

- Where was dataset found?
- How it was produced, what is workflow?
- What versions of software and OS were used?
- Are all facts considered?
- Is it up-to-date?
- Where to find all versions of software and data?
Current provenance standards

PROV (provenance vocabulary) is a W3C standard introduced in 2013.

- PROV-DM, data model for provenance
- PROV-N: The Provenance Notation
- PROV-O: The PROV Ontology
- PROV-JSON: a JSON Representation for the PROV data model
- GDPRov: ontology enrichment for describing the provenance of data and consent lifecycles using GDPR terms
PROV-O terms
PROV-O example from PARTHENOS project

```xml
<prov:document>
  <prov:activity prov:id="d4s:STANBOL_WRAPPER_ID_600f1ed4-9b15-406a-a496-47debdc6da5">
    <prov:startTime>17/07/2017 16:23:17</prov:startTime>
    <prov:endTime>17/07/2017 16:23:20</prov:endTime>
    <prov:type xsd:QName>d4s:computation</prov:type>
    <prov:softwareAgent prov:id="d4s:dataminer.d4science.org"/>
    <prov:person prov:id="d4s:vyacheslav.tykhonov"/>
    <prov:entity prov:id="d4s:operator_name">
      <prov:value xsd:string>STANBOL_WRAPPER</prov:value>
    </prov:entity>
    <prov:entity prov:id="d4s:operator_id">
      <prov:value xsd:string>org.gcube.dataanalysis.wps.statisticalmanager.synchserver.mappedclasses.transducerers.STANBOL_WRAPPER</prov:value>
    </prov:entity>
    <prov:entity prov:id="d4s:operator_description">
      <prov:value xsd:string>This process sends an input text file to the Apache Stanbol REST API and returns a semantically enriched output text file in json-ld format</prov:value>
    </prov:entity>
    <prov:entity prov:id="d4s:VRE">
      <prov:value xsd:string>/d4science.research-infrastructures.eu/D4Research/NERLiX</prov:value>
    </prov:entity>
    <prov:entity prov:id="d4s:status">
      <prov:value xsd:string>100</prov:value>
    </prov:entity>
    <prov:entity prov:id="d4s:input_1STANBOL_WRAPPER_ID_600f1ed4-9b15-406a-a496-47debdc6da5.csv">
      <prov:value xsd:string>http://data.d4science.org/eTNiaHN3U2t1VXpxWERZdytjSj1Q1h2cVhKtmNSOHZHbWJQNStlS0N6Yz0</prov:value>
    </prov:entity>
  </prov:activity>
</prov:document>
```
Provenance information in Software Agents

All KNAW institutes involved in CLARIAH using Docker to distribute their software. Docker is the most popular (at the moment) tool that performs operating-system-level virtualization.

Some useful features:

- Docker contains some provenance information in yaml specification
- Docker images can be distributed with OS and installed software components
- The state of running Docker container can saved as snapshot, archived in some repository and transmitted for further collaboration with other groups of developers or researchers

Docker orchestration tools allow to build workflow pipelines keeping some provenance
Example: Docker Compose specification (Timbuctoo)

```yaml
version: '2.1'

services:
  elasticsearch:
    image: huygensing/elasticsearch:5.6.5
    ports:
      - 9200:9200
  indexer:
    image: huygensing/timbuctoo-elasticsearch-indexer
    environment:
      indexelasticsearch_host: "http://elasticsearch:9200"
      indexer_port: "80"
      indexertimbuctoo_graphql_endpoint: "http://timbuctoo/v5/graphql"
      indexertimbuctoo_login_endpoint: "http://timbuctoo/v2.1/authenticate"
      timbuctoo_user: ""
      timbuctoo_password: ""
    ports:
      - 3000:80
  timbuctoo-ui:
    image: huygensing/timbuctoo-ui:develop
    environment:
      REACT_APP_API_URL: "http://localhost:8080/v5/graphql"
    ports:
      - 3006:80
  timbuctoo:
    image: huygensing/timbuctoo
    command: "$@< app/bin/timbuctoo server /app/example_config.yaml"
    environment:
      timbuctoo_elasticsearch_host: http://elasticsearch
      timbuctoo_elasticsearch_port: 9200
      base_url: http://localhost:8080
      JAVA_OPTS: $JAVA_OPTS
      timbuctoo_dataPath: "$@< /app/example_config.yaml
      timbuctoo_dataPath: "$@< /app/example_config.yaml
    volumes:
      - .:/timbuctoo-instancev4/temp_for_debugrun/
      - .:/timbuctoo-instancev4/example_config.yaml:/app/example_config.yaml
      - ./bazel-bin:/app/bazel-bin
      - ./timbuctoo-instancev4/target/appassembler/bin:/app/bin
      - ./timbuctoo-instancev4/target/appassembler/repo:/app/repo
    ports:
      - 8080:80
      - 8081:81
      - ${DEBUG_PORT:-5005}:5005
```
Provenance of Docker images

Main challenge in Docker orchestration: how to track provenance information in distributed images? Can you trust Docker Hub?

Docker Content Trust provides strong cryptographic guarantees over what code and what versions of software are being run in your infrastructure.

It integrates The Update Framework (TUF) into Docker using Notary, an tool that provides trust over any content.

- Image Provenance is critical for Production
- Distributed content should be digitally signed
Provenance of data

• Data repository should be able to archive all revisions of dataset containing provenance information

• In the ideal situation every change in data should have provenance record, for reproducibility purposes

• Archived datasets with provenance information can get “second life” in the new research keeping all references between original and forked datasets

• In the combination with SoftwareAgent provenance, it can form an
Provenance repositories

• Apache NiFi was designed to automate the flow of data between software systems
• Dataverse has specific API to keep provenance information about Software Agents and workflows up-to-date (PROV-JSON)
• ProvStore is a web service that allows you to store, browse and manage your provenance documents

The most intriguing is to transmit provenance information from repository to LOD!
Linked Data provenance challenges

Provenance representation:

• Annotation method (provenance is pre-computed and stored as metadata, can be archived as well)

• Inversion method (provenance is computed when needed)

Provenance storage approaches:

• Stored within the same storage system

• Distributed storage in different locations, VoID (Vocabulary of Interlinked Datasets)
Questions?