Introduction to Data Science

MODULE I – PART I

DATA PROVENANCE

Prof Sergio Serra e Jorge Zavaleta

Provenance (in science) is not new…. It was originated in Arts!

Where did this data(set) come from?

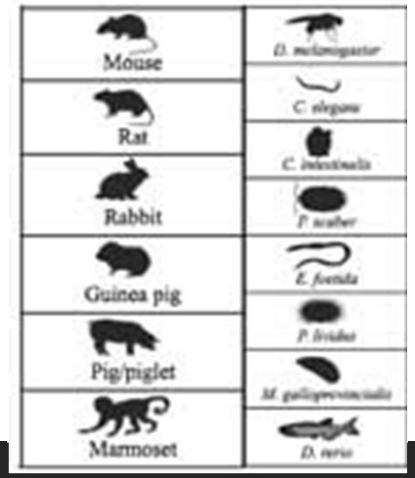
How did I get this particular result?

What mappings produced it?

How much should I trust (believe) it?

…

Types of Experiments



IN VIVO VS IN VITRO

Have you ever read any medical studies? If you have, then you have probably seen that some of them are "in vitro" and some others are "in vivo". These two terms sound very similar and don't look English at all, so it's not surprising that they are often confused. However, there is a very big difference between them.

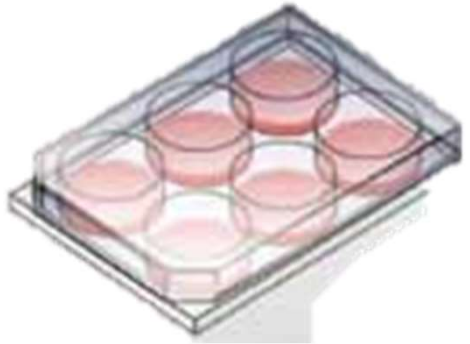
DEFINITION

IN VIVO describes a medical experiment or a test that is performed on a living organism, e.g. a human being or a laboratory animal.

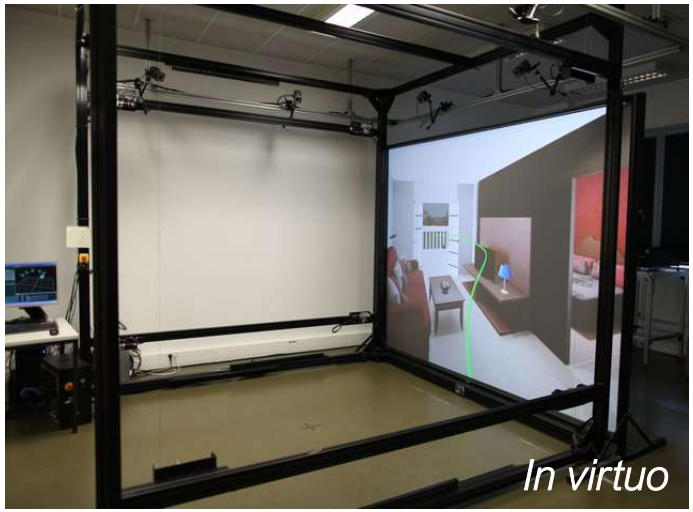
DEFINITION

IN VITRO is a medical experiment or a study that is performed only in a laboratory dish or a test tube.

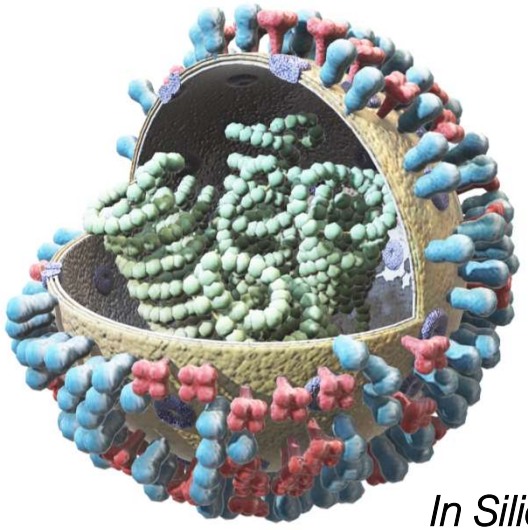
Fonte: https://7esl.com/in-vivo-vs-in-vitro/



Types of Experiments



In virtuo



In Silico

- Involve the **interaction among participants** and a **computerized model of reality**.
- **Behavior of the environment** with which subjects interact is described as a model and represented by a **computer program**

- Both the **subjects and real world being described as computer models**.
- **The environment is fully composed by numeric models** to which no **human interaction** is allowed.

Travassos and Barros "Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering." WSESE 2003

Desmeulles et al. The virtual reality applied to biology understanding: The in virtuo experimentation. Expert Systems with Applications. Volume 30, Issue 1, January 2006, Pages 82-92

Travassos and Barros "Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering." WSESE 2003

What Provenance is?

Many Definitions...

The **history or pedigree** of a work of art, rare book, etc...

Provenance (Dictionary Definitions)

1. The Merriam-Webster online diction – Origin , Source
2. Oxford English Dictionary – The place of origin or earliest known history of something; origin, derivation.

Provenance (Literature Definitions)

1. Provenance refers to the **source of Information** such as entities and processes involved in producing or delivering an **artifact**. (Yolanda, Gil)
2. Provenance is a description of how things came to be, and how they came to be in the state they are in today. **Statements about provenance can themselves be considered to have provenance**. (Myers, Jim)

What Provenance is?

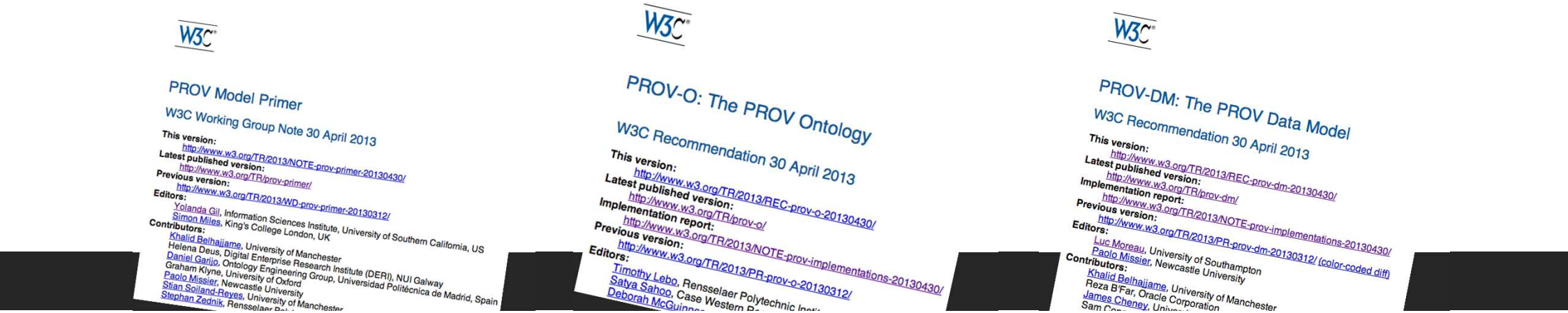
Other Provenance Definitions

1. On the web, provenance would include information about the **creation and publication of web resources** as well as **information about access of those resources**, and activities related to their **discussion, linking, and reuse**.
2. In scientific research, provenance may include the **set of physical and computational processes** applied to a sample that would **allow repetition of an experiment as well as descriptive information** about a sample, the experimental protocol that would allow **reproduction of the work**
    1. E-science is computationally intensive science carried out in highly distributed network environments, uses immense datasets that require HPC.
3. In business, provenance may include **information about financial and legal processes** (e.g. in contracts) as well as the electronic (e.g. online ordering) and physical (e.g. shipping) processes that have occurred.
4. In database, data provenance, a kind of metadata, sometimes called “lineage" or “pedigree" is the **description of the origins of a piece of data and the process** by which it arrived in a database.

What Provenance is?

Provenance Working Definition

1. Provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource. **Provenance provides a critical foundation for assessing authenticity, enabling trust, and allowing reproducibility**. **Provenance assertions** are a form of contextual metadata and can themselves become important records with their own provenance. (W3C)

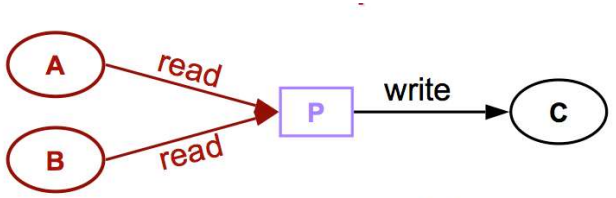


Why provenance is important?

The need of provenance for data integration and reuse

- Data comes from various diverse data sources
- Varying Data Quality
- Reuse
- Audit Trail
- Reproducibility
- Trust / Attribution
- Debugging

"If you are a scientist, or any kind of scholar, you would like to have **confidence** in the **accuracy** and **timeliness** of the data that you are working with. Research requires **tight controls on the quality of data** because mistakes can harm people's health, etc.."



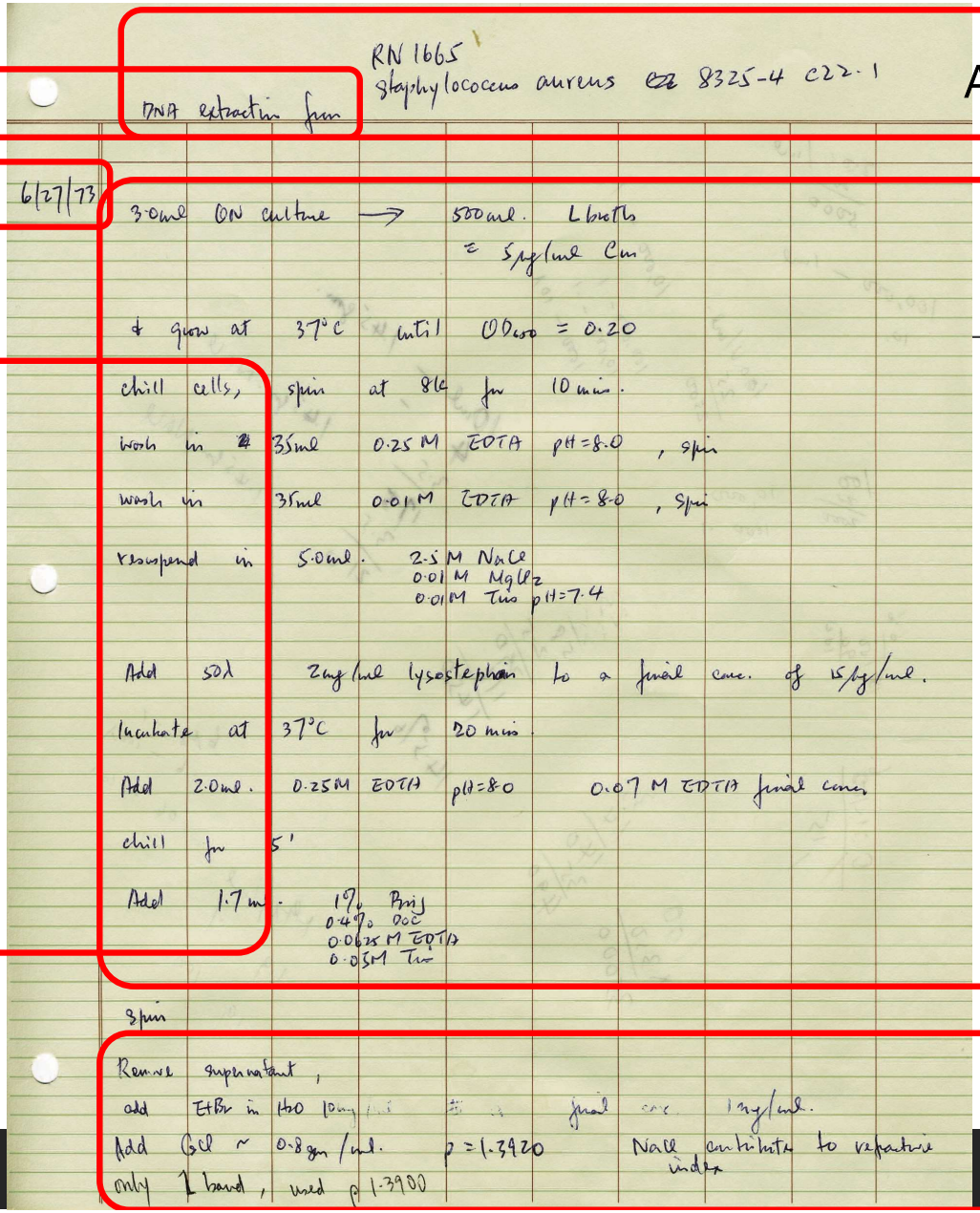
Provenance of C

- Input Files A, B
- Application P
- Command line Args
- Environment
- Processor type, OS, etc

What

When

How



Annotations

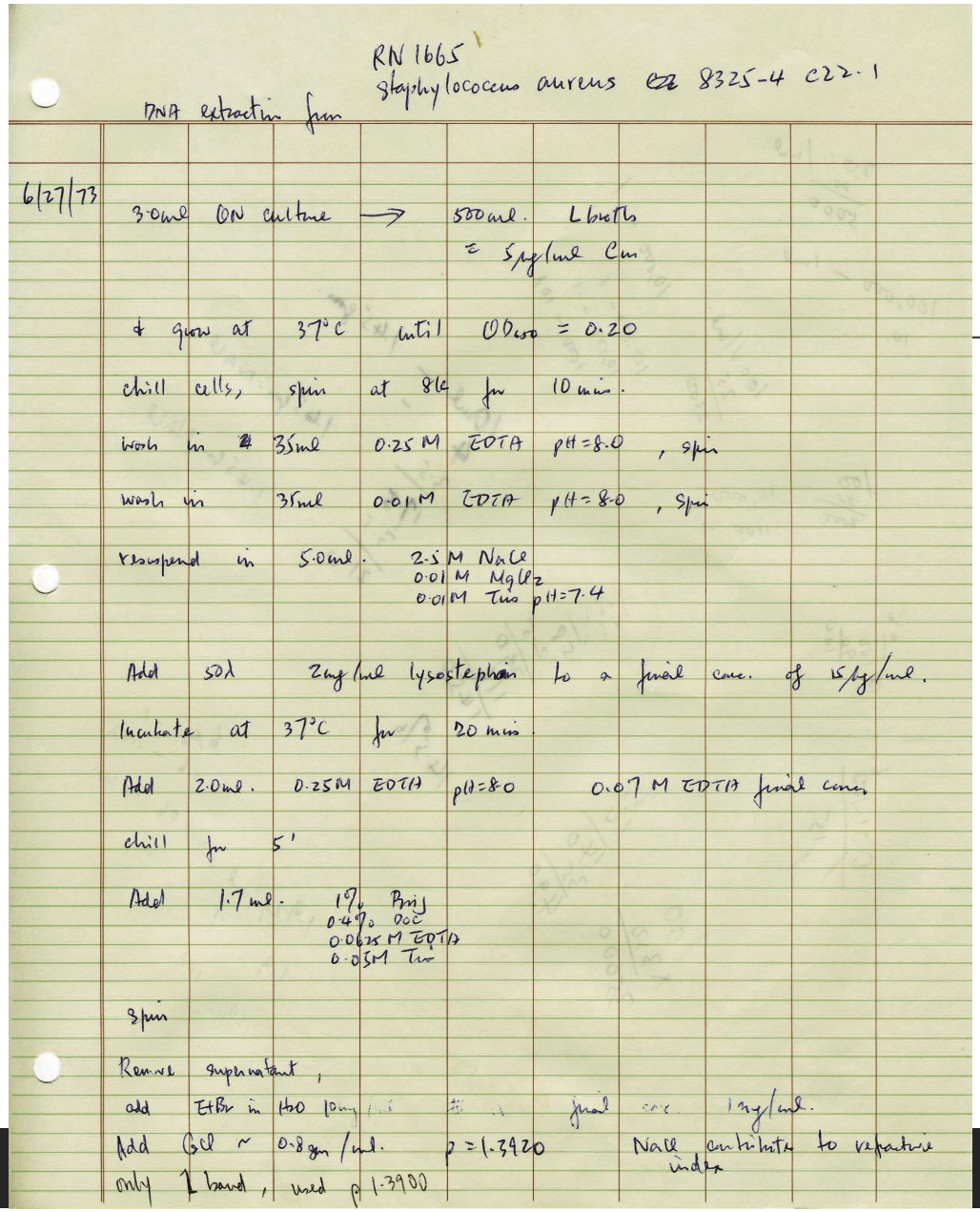
Data

Data

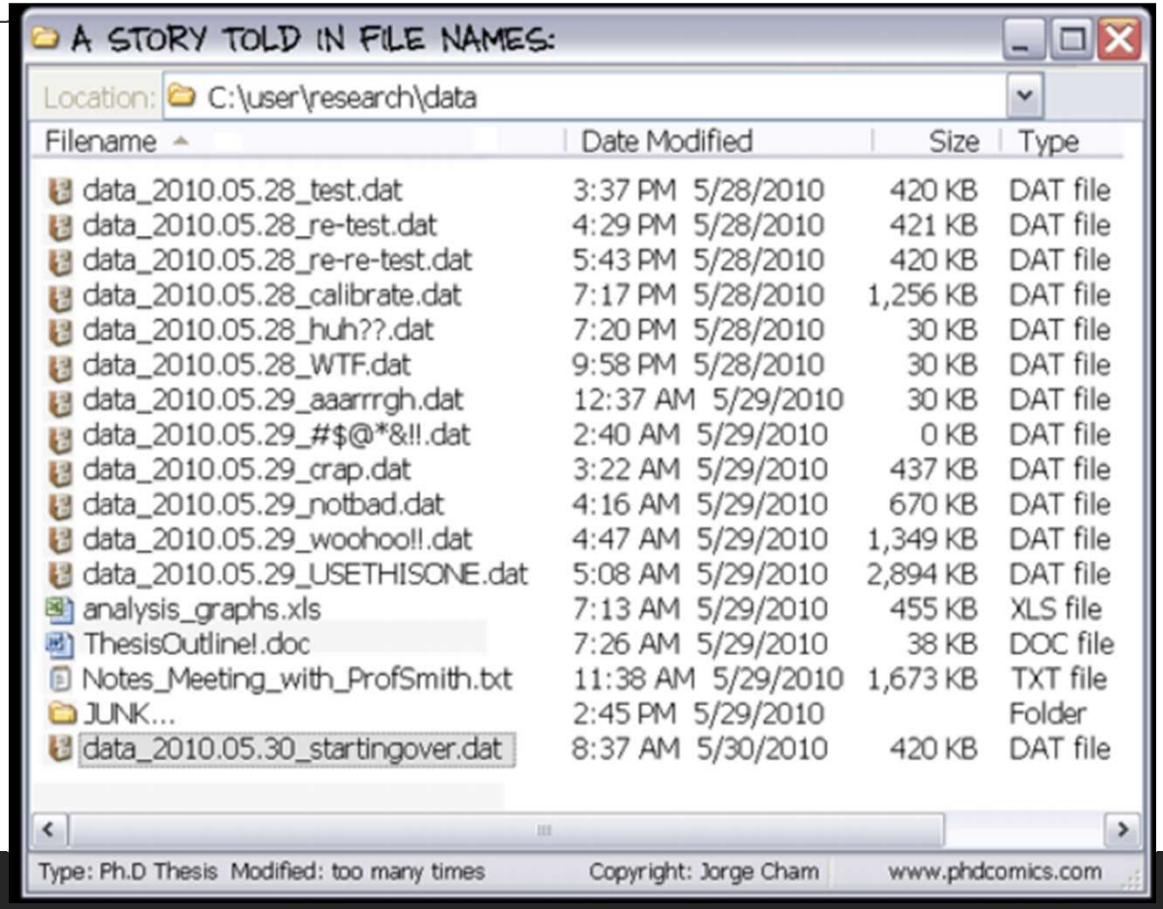
Who

A page of Annie Chang's notes on the *S. aureus* experiment.

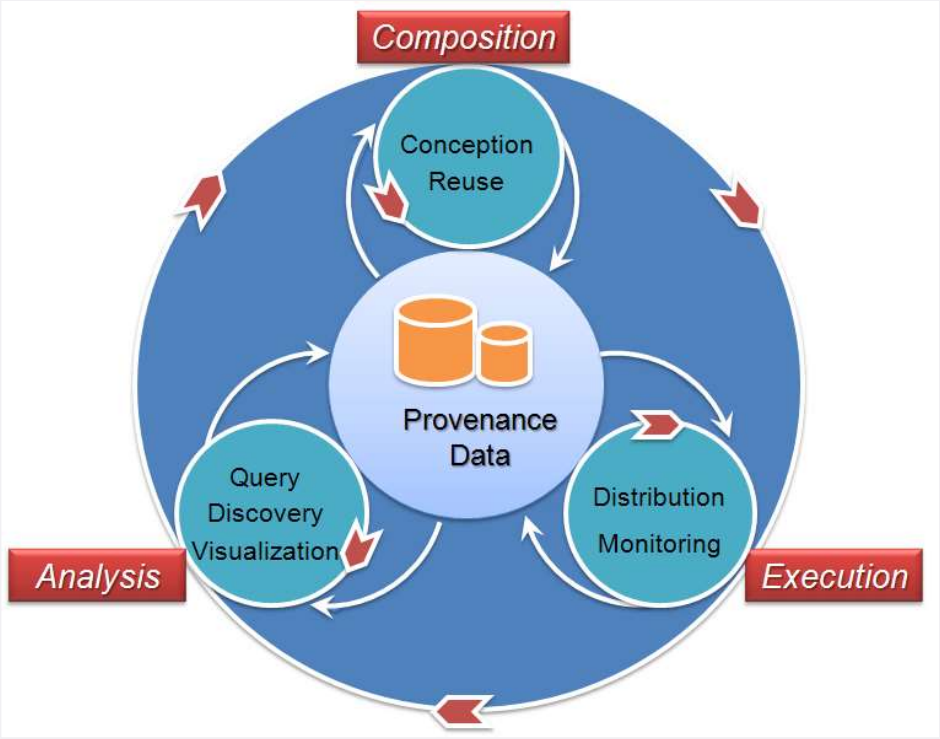
https://medium.com/lsf-magazine/the-invention-of-recombinant-dna-technology-e040a8a1fa22



Not Scalable!
Not Searchable!
Metadata is on file names!



Scientific Experiment Lifecycle



1- Composition

- Elicit requirements to build/reuse workflows as a software
- Define hypothesis; Select partners, resources, datasets
  - No concern about infra-structure!

2 - Execution

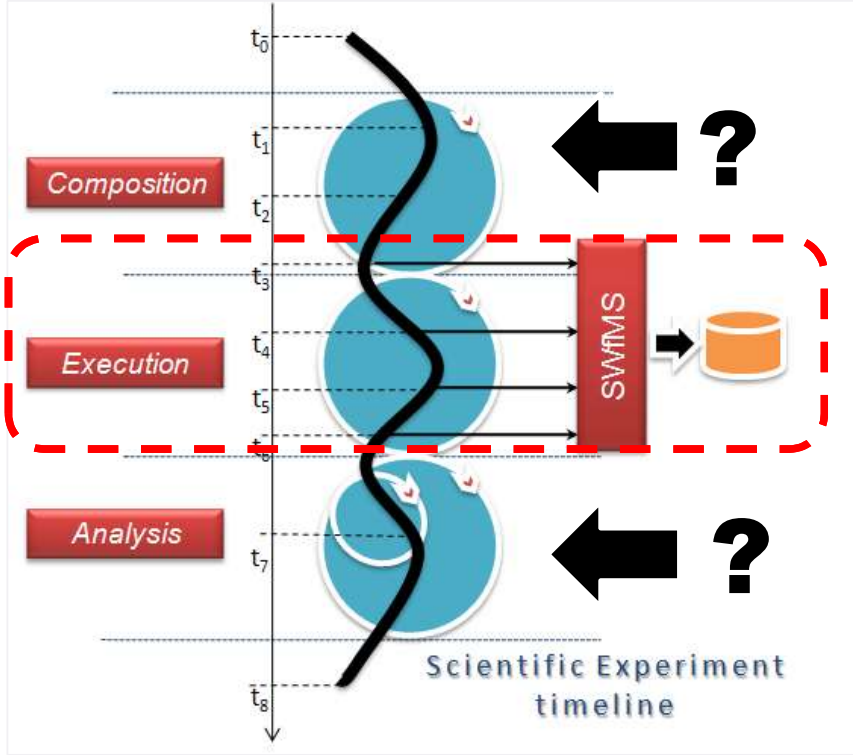
- Run workflows instances (*real parameters, datasets, environment*)
- Monitor local/distributed environments

3 - Analysis

- Analyze workflows´ results
- Visualize or query provenance
  - Insights about the experiment!

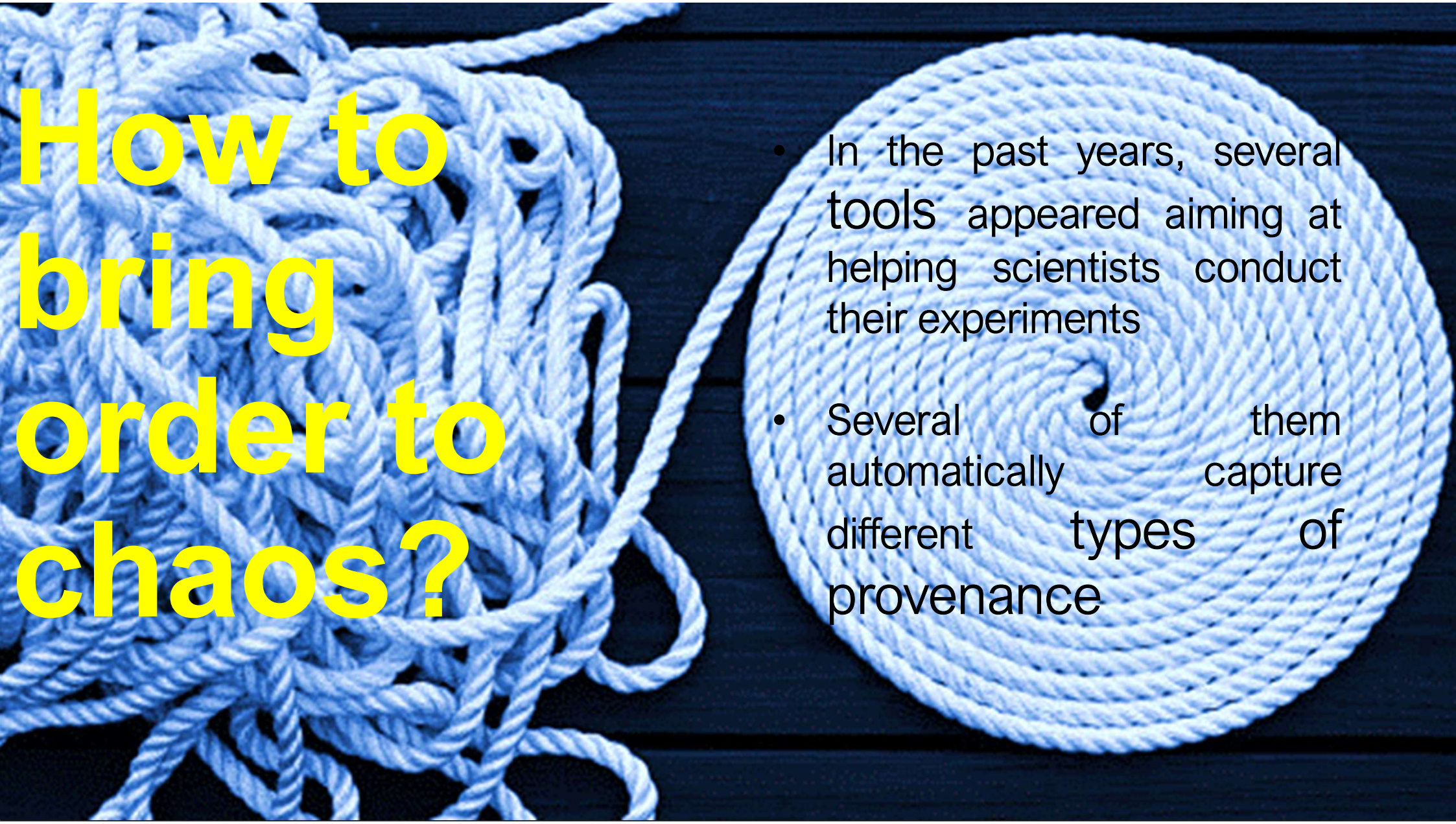
Mattoso, M., et al.. (2010) "Towards Supporting the Life Cycle of Large Scale Scientific Experiments". International Journal of Business Process Integration and Management, v. 5, p. 79-92.

Scientific Experiments Lifecycle

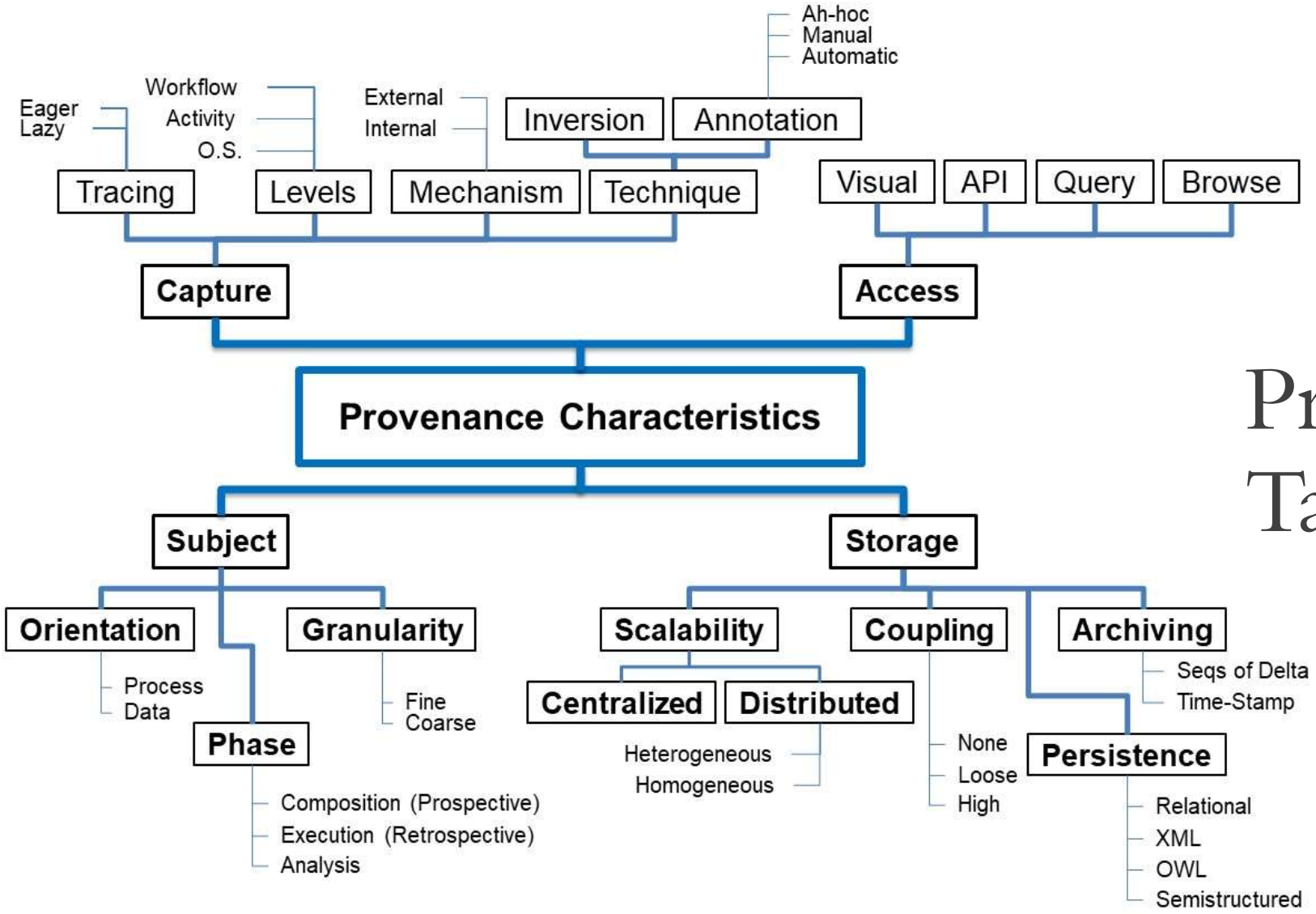


Provenance data of all phases of the experiment must be collected and stored

Each stage generates different kinds of provenance

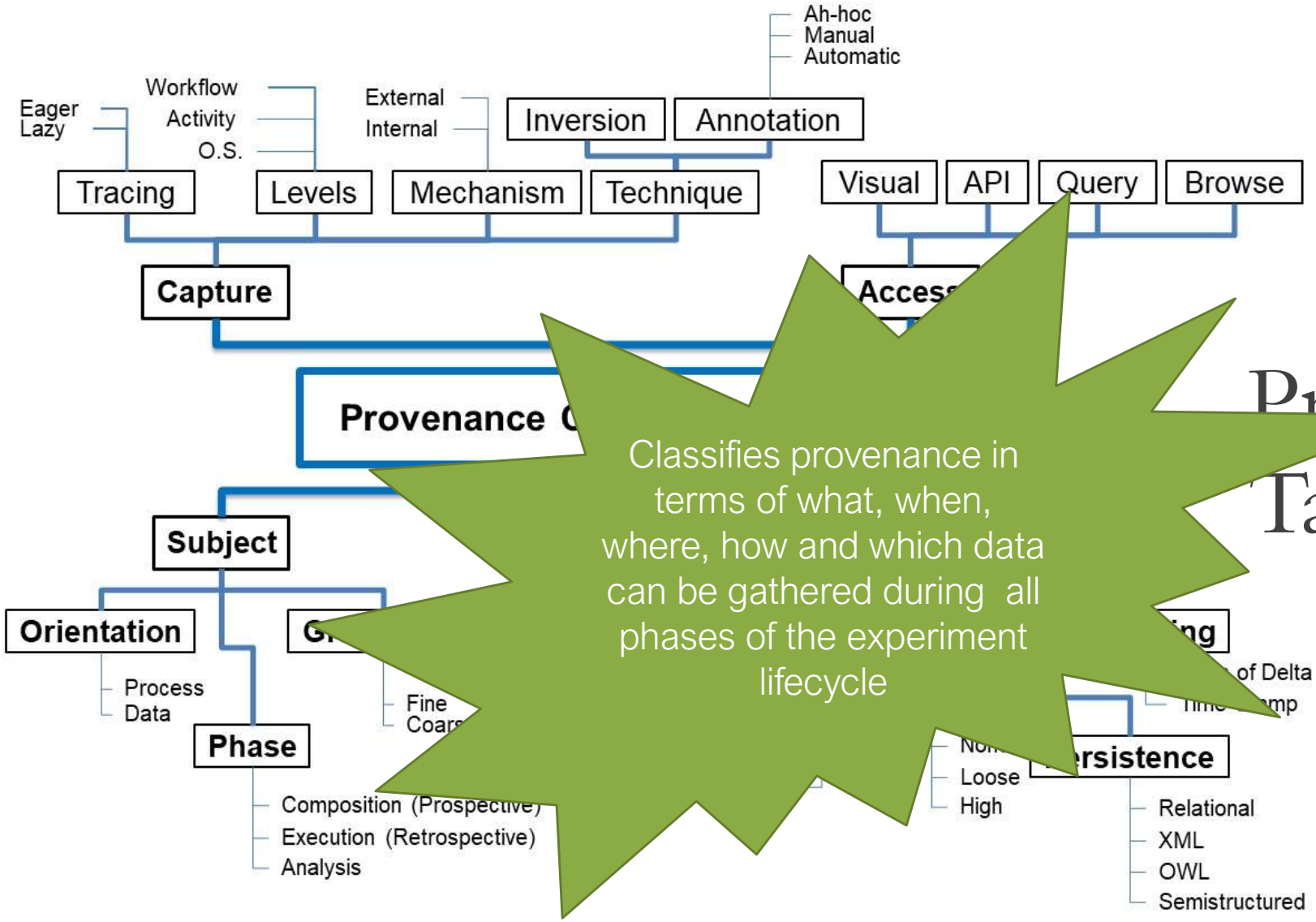

How to bring order to chaos?
• In the past years, several tools appeared aiming at helping scientists conduct their experiments
• Several of them automatically capture different types of provenance

Provenance Taxonomy



Cruz et al., Towards a Taxonomy of Provenance in Scientific Workflow Management Systems - June 2009 DOI: 10.1109/SERVICES-I.2009.18

Provenance Taxonomy

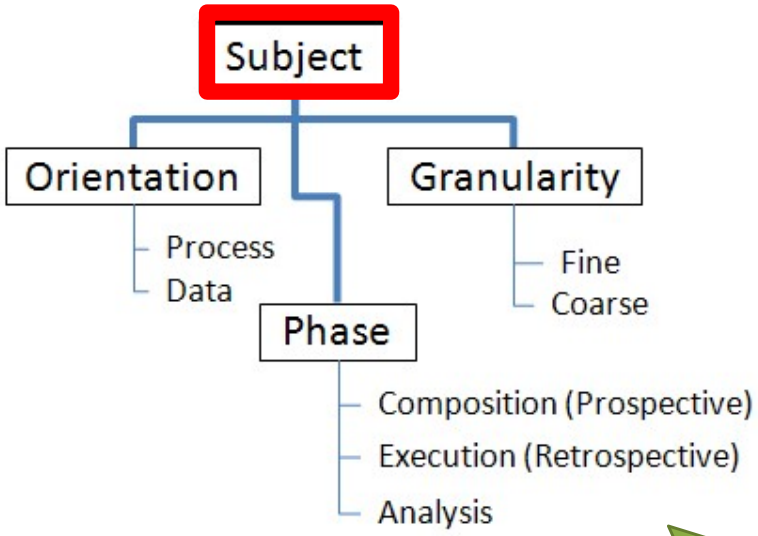


Classifies provenance in terms of what, when, where, how and which data can be gathered during all phases of the experiment lifecycle

* Need a FAIR + scripts update

Pimentel et al. (2019). A Survey on Collecting, Managing, and Analyzing Provenance from Scripts. ACM Comput. Surv., Vol. 1, No. 1, Article 1. Publication date: January 2019

Provenance Taxonomy

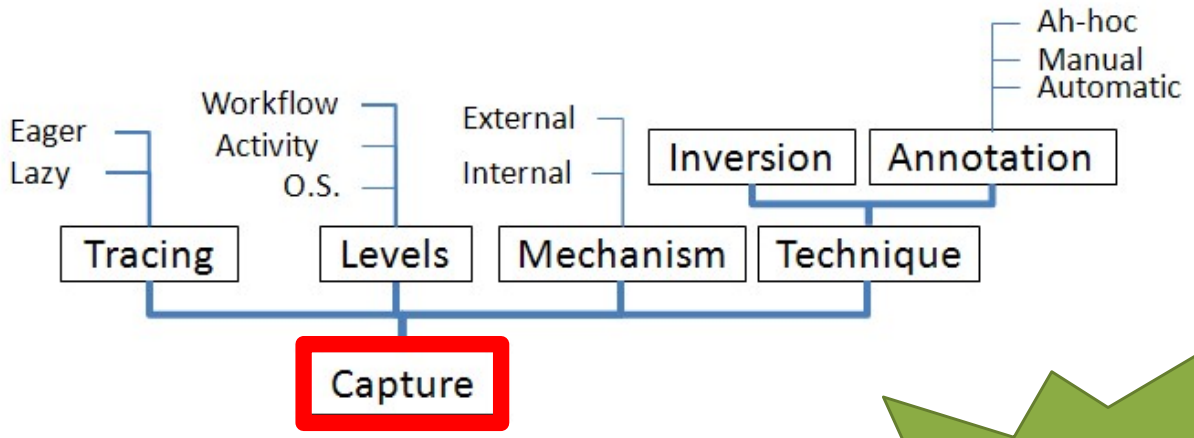


Orthogonal to all phases

Provenance data can be represented in terms of different subjects and distinct levels of details.

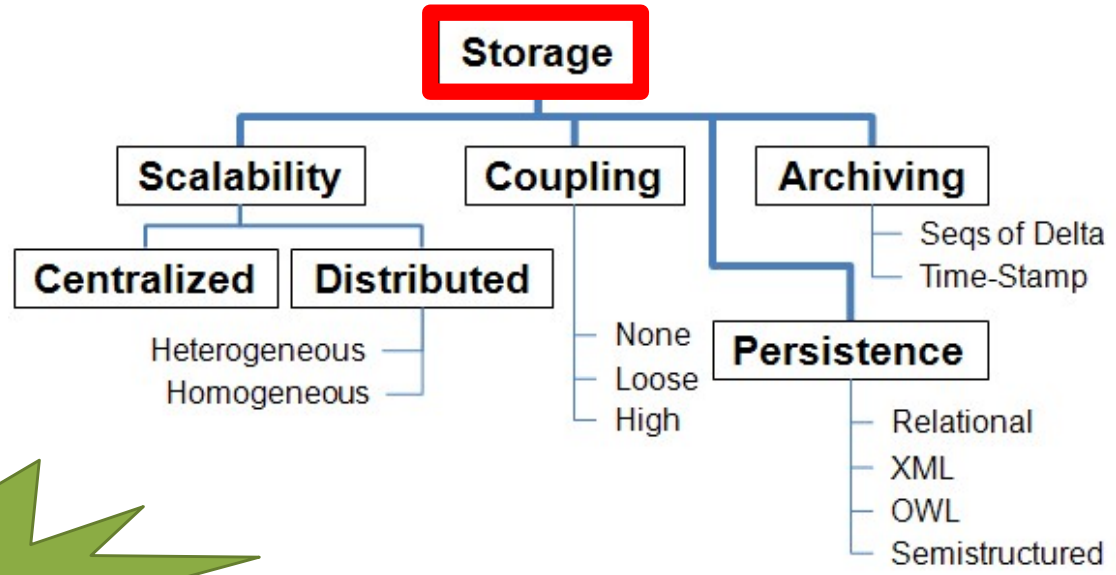
Provenance Taxonomy

Classifies how provenance data can be captured on the existing provenance systems.



Composition & Execution

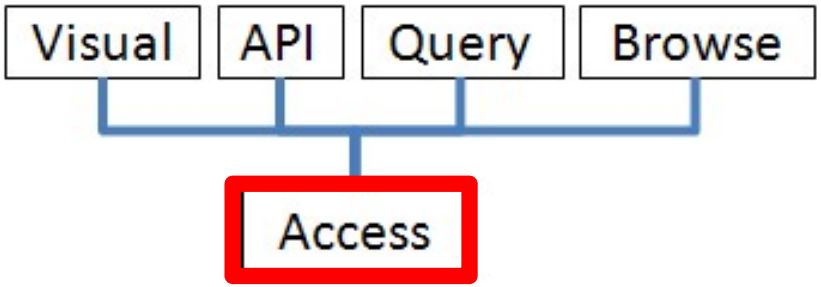
Provenance Taxonomy



Execution

Includes the approaches that provenance systems use to register provenance data.

Provenance Taxonomy



Analysis

- Describes how scientists can access provenance data repositories.

Provenance Uses (1)

Attribution - provenance as the sources or entities that were used to create a new result

- Responsibility - knowing who endorses a particular piece of information or result
- Origin - recorded vs reconstructed, verified vs non-verified, asserted vs inferred

Process - provenance as the process that yielded an artifact

- Reproducibility (e.g. workflows, mashups, text extraction)
- Data Access (e.g. access time, accessed server, party responsible for accessed server)

Evolution and versioning -

- Republishing (e.g. re-tweeting, re-blogging, re-publishing)
- Updates (e.g. a document with content from various sources and that changes over time)

Justification for decisions – Includes argumentation, hypotheses, why-not questions

Entailment - given the results to a particular query, what tuples led to those results

Provenance Uses (2)

Dissemination control – Track policies specified by creator for when/how an artifact can be used

- Access Control - incorporate access control policies to access provenance information
- Licensing - stating what rights the object creators and users have based on provenance
- Law enforcement (e.g. enforcing privacy policies on the use of personal information)

Understanding - End user consumption of provenance

- abstraction, multiple levels of description, summary
- presentation, visualization

Accountability - the ability to check the provenance of an object with respect to some expectation

- Verification - of a set of requirements
- Compliance - with a set of policies

Trust - making trust judgments based on provenance

- Information quality - choosing among competing evidence from diverse sources (e.g. linked data use cases)
- Incorporating reputation and reliability ratings with attribution information

Types of Provenance

Prospective Provenance

- captures a computational **task's specification** (whether it's a **script** or a **workflow**) and corresponds to the **steps (or recipe)** that must be **followed to generate** a data product or class of data products

Retrospective Provenance

- captures the **steps executed** as well as **information about the environment** used to **derive a specific data product** – it's a detailed log of a computational task's execution and the **data (input/output)** involved in the **execution**

Provenance Granularity (Coarse Grained/Fine Grained)

The usefulness of provenance and the **cost of collecting** and **storing provenance** in a certain domain is linked to the **granularity** at which it is collected.

Range from provenance on **attributes and tuples in a database to provenance for collections of files**, generated by an ensemble experiment run.

Workflow Provenance

- Workflow Provenance is coarse-grained
- Refers to records of history of the derivation of the final output of workflow
- Perform typically for complex processing tasks

Data Provenance

- Data provenance is Fine-grained
- Derivation of a piece of data i.e. results of transformations
- Description of the origin of a piece of data and process by which it arrives in a database

Provenance Granularity (Coarse Grained/Fine Grained)

Workflow Provenance

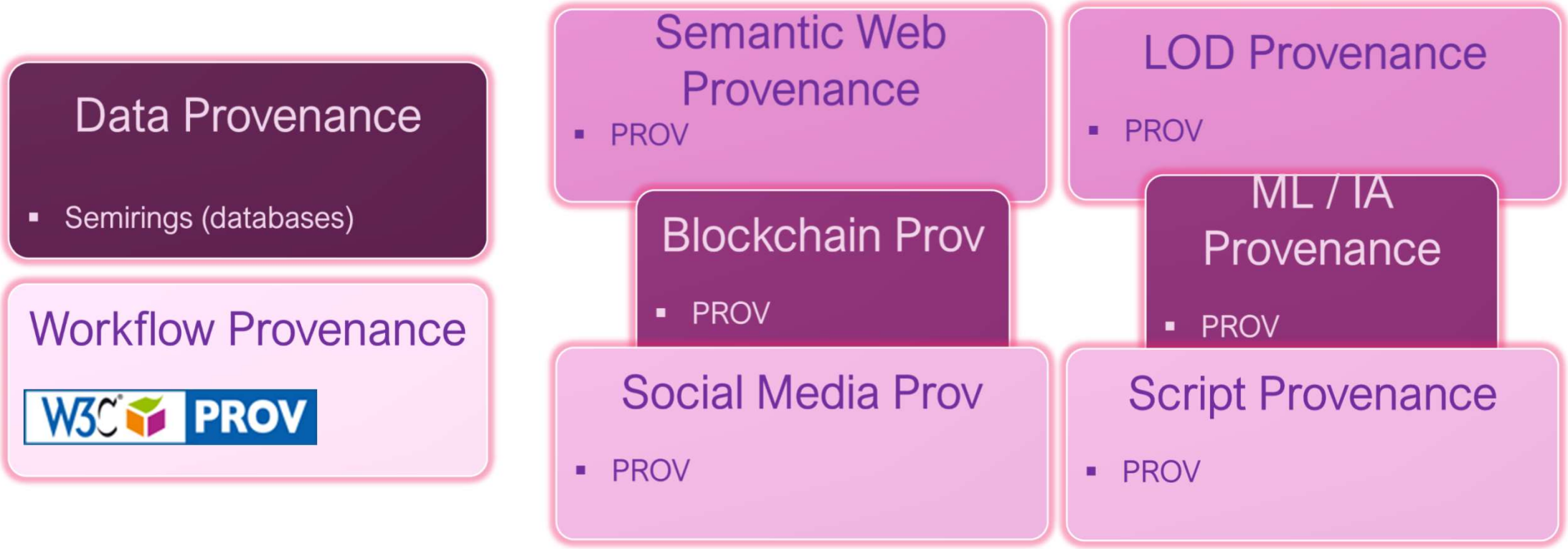
- When Information raw observations that is referred to as “coarse-grain” or “workflow” provenance.
- The widespread use of **workflow tools for processing scientific data facilitates for capturing provenance information.**
- The workflow process describes all the steps involved in producing a given data set and, hence captures it provenance information.

Data Provenance*

- Fine-grain provenance is also known as **where, how and why-Provenance**.
  - For example: A query execution copy data elements from some source to some target database
- **Where-provenance** identifies these source elements where the data in the target is copied from.
- **Why-provenance** provides justification for the data elements appearing in the output
- **How-provenance** describes some parts of the input influenced certain parts of the output.

* Green, T. Provenance semirings. Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systemsJune 2007 Pages 31–40https://doi.org/10.1145/1265530.1265535

Provenance Granularity →Provenance Models



* Green, T. Provenance semirings. Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systemsJune 2007 Pages 31–40https://doi.org/10.1145/1265530.1265535

Provenance as Annotations on Data

Source relations

R

A	B
1	2
1	4

S

B	C
2	3
3	2
4	3

View $V_1 = R \bowtie S \cup S \bowtie S$

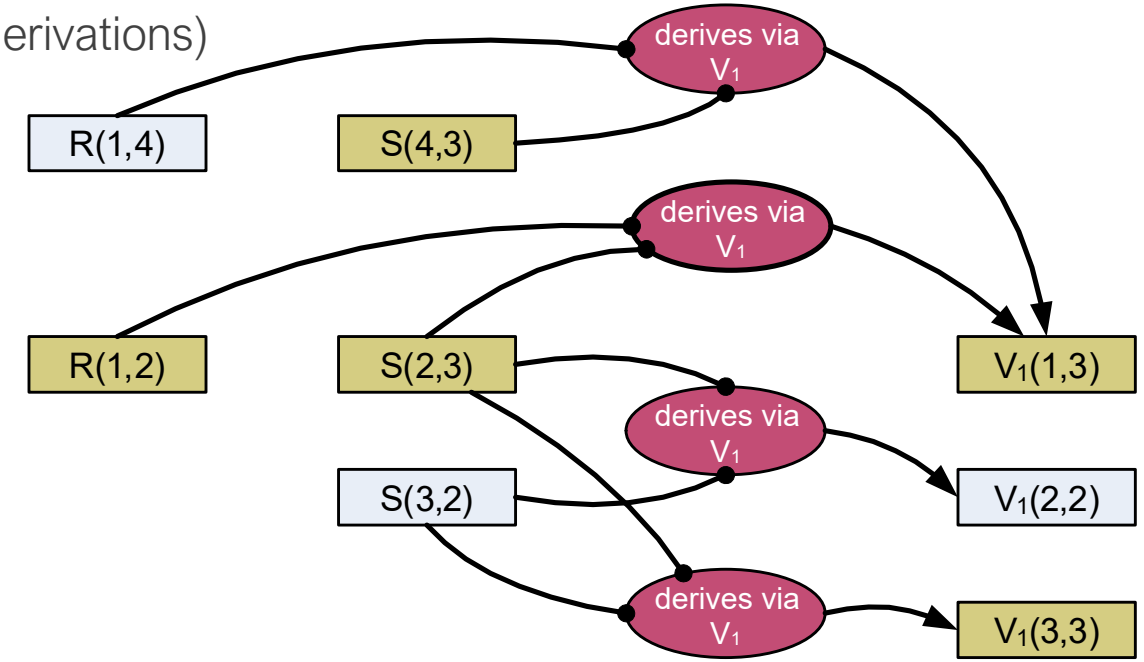
A	C	*directly derivable by => provenance annotation*
1	3	$R(1,2) \bowtie S(2,3) \cup R(1,4) \bowtie S(4,3)$
2	2	$S(2,3) \bowtie \rho_{B \to A,\ C \to B}\ S(3,2)$
3	3	$S(3,2) \bowtie \rho_{B \to A,\ C \to B}\ S(2,3)$

Annotate each derivation with an “explanation” in terms of relational algebra and the tuple operands

* Green, T. Provenance semirings. Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systemsJune 2007 Pages 31–40https://doi.org/10.1145/1265530.1265535

Provenance as a Graph of Relationships

- Bipartite graph: tuple nodes connected via "derivation nodes"
  - Encodes a hypergraph (hyperedges = derivations)
- Makes *direct derivation relationships* more explicit



* Green, T. Provenance semirings. Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systemsJune 2007 Pages 31–40https://doi.org/10.1145/1265530.1265535

Making the two interchangeable

We can make these **equivalent by introducing provenance tokens** (equiv. node IDs) for each tuple

Derived tuples' annotations = expressions over tokens

R

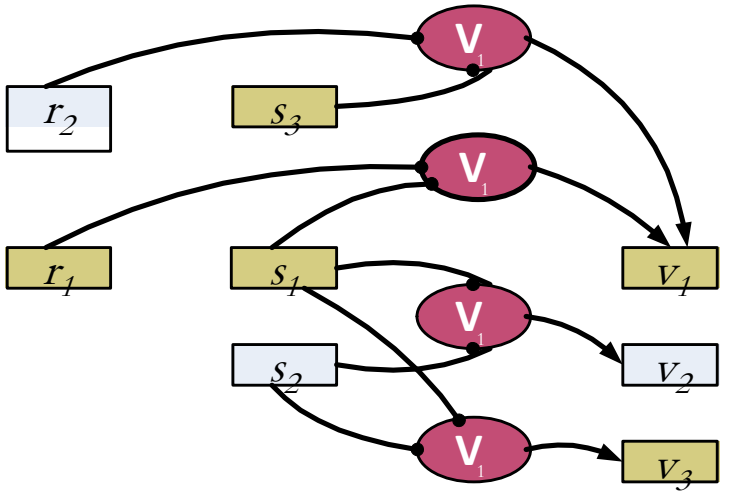
A	B	*ann*
1	2	r_1
1	4	r_2

S

B	C	*ann*
2	3	s_1
3	2	s_2
4	3	s_3

V_1

A	C	*ann*
1	3	$v_1 = r_1 \bowtie s_1 \cup r_2 \bowtie s_3$
2	2	$v_2 = s_1 \bowtie s_2$
3	3	$v_3 = s_2 \bowtie s_1$



* Green, T. Provenance semirings. Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systemsJune 2007 Pages 31–40https://doi.org/10.1145/1265530.1265535

The Provenance Semiring Model

To represent **data provenance**, use:

- A set of *provenance tokens* or tuple IDs, K
- Abstract operators representing combination of tuples
  - **Abstract sum operator, ⊕, for union or projection**
    - has identity element 0 (a ⊕ 0 ≡ 0 ⊕ a ≡ 0)
  - **Abstract product operator, ⊗, for join**
    - has identity element 1 (a ⊗ 1 ≡ 1 ⊗ a ≡ 1)
    - also (a ⊗ 0 ≡ 0 ⊗ a ≡ 0)

This is formally a **commutative semiring**

R

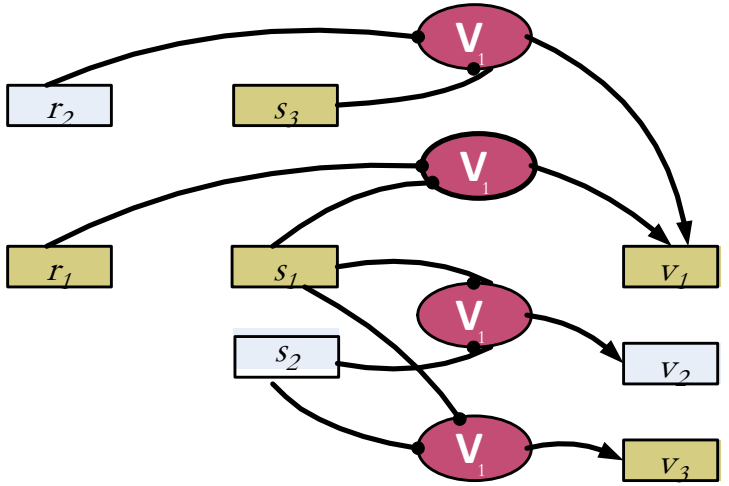
A	B	*ann*
1	2	r_1
1	4	r_2

S

B	C	*ann*
2	3	s_1
3	2	s_2
4	3	s_3

V_1

A	C	*ann*
1	3	$v_1 = r_1 \otimes s_1 \oplus r_2 \otimes s_3$
2	2	$v_2 = s_1 \otimes s_2$
3	3	$v_3 = s_2 \otimes s_1$



* Green, T. Provenance semirings. Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systemsJune 2007 Pages 31–40https://doi.org/10.1145/1265530.1265535

Tokens for Mappings

Sometimes we would like to assign a token to the actual *mapping* or *rule* used – so we can assign it a value

R

A	B	*ann*
1	2	r_1
1	4	r_2

S

B	C	*ann*
2	3	s_1
3	2	s_2
4	3	s_3

View V_1 (in Datalog):

$V_1(x,z)$:- R(x,y), S(y,z) ← *Call this* m_1

$V_1(x,x)$:- S(x,y), S(y,x) ← *Call this* m_2

V_1

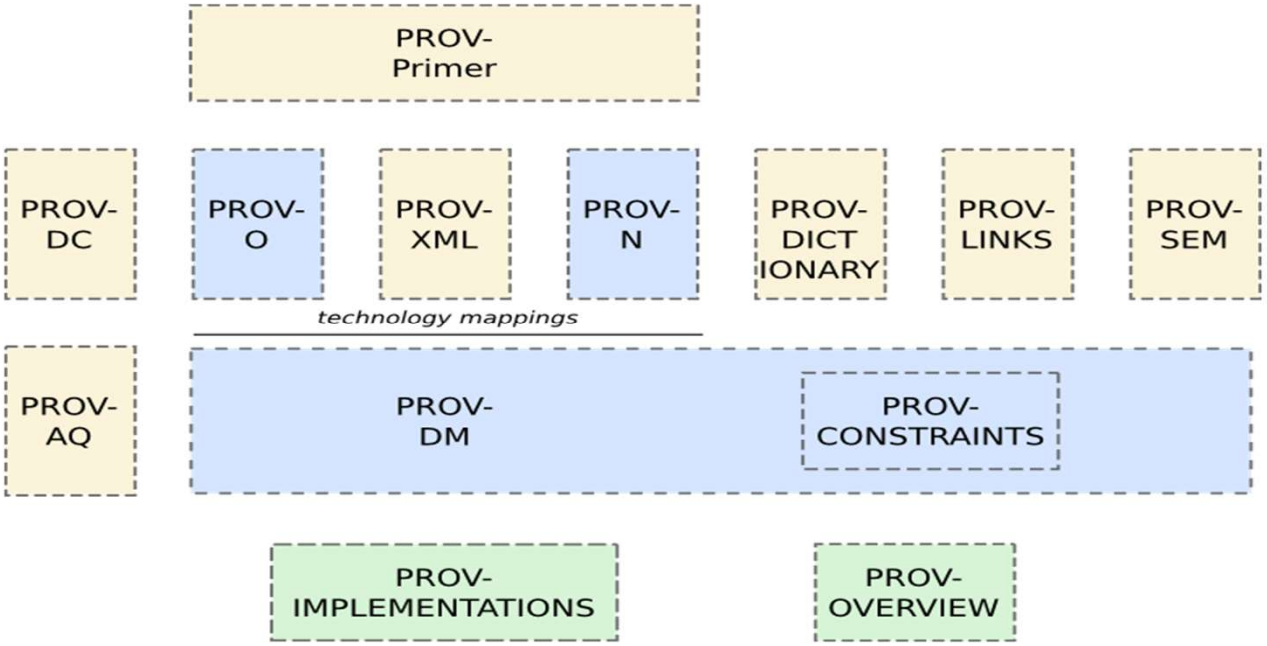
A	C	*ann*
1	3	$v_1 = m_1 \otimes [r_1 \otimes s_1] \oplus m_2 \otimes [r_2 \otimes s_3]$
2	2	$v_2 = m_2 \otimes [s_1 \otimes s_2]$
3	3	$v_3 = m_2 \otimes [s_2 \otimes s_1]$

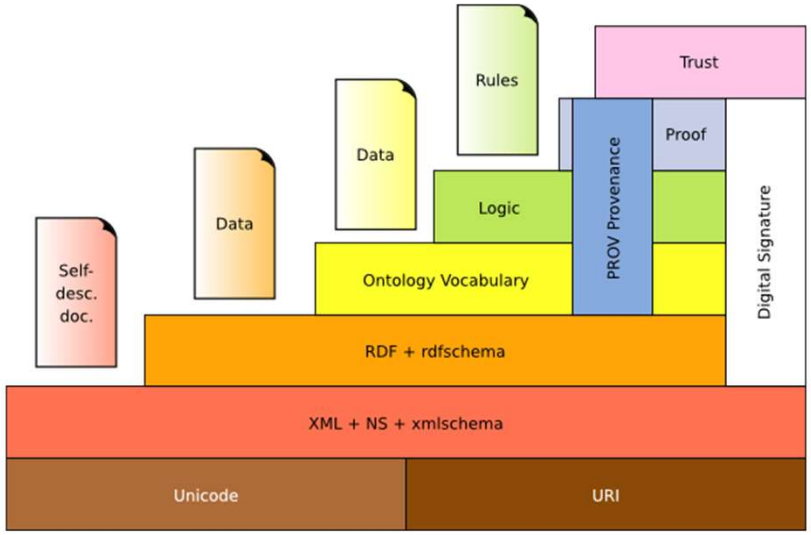
* Green, T. Provenance semirings. Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systemsJune 2007 Pages 31–40https://doi.org/10.1145/1265530.1265535

The PROV Model

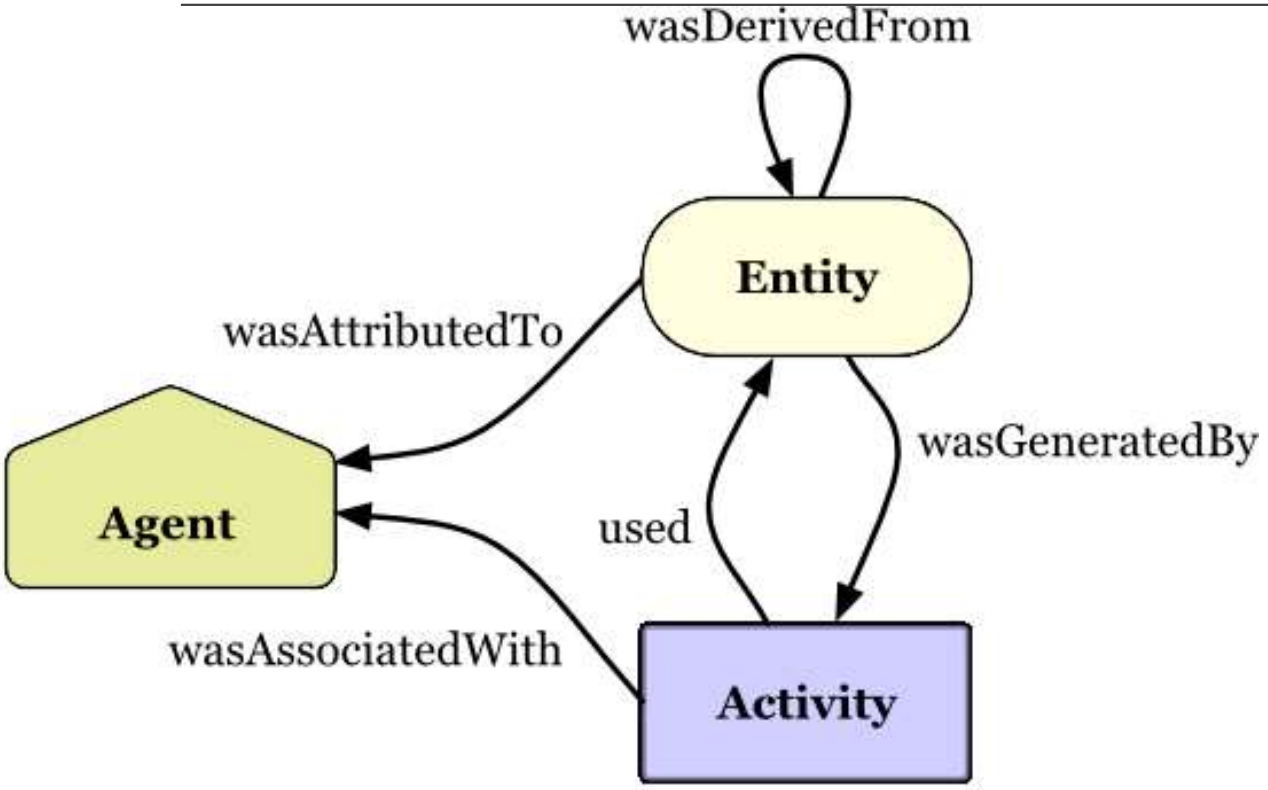
- **Different perspectives on provenance. Provenance records are metadata.**
- **PROV** was developed by the **W3C provenance incubator group**
- Perspective 1 - ***agent-centered provenance*** - what people or organizations were involved in generating or manipulating the information.
  - For example, in the provenance of a picture in a news article we might capture the photographer who took it, the person that edited it, and the newspaper that published it.
- Perspective 2 - ***object-centered provenance*****,** trace the origins of portions of a document to other documents.
  - For example, is having a web page that was assembled from content from a news article, quotes of interviews with experts, and a chart that plots data from a government agency.
- Perspective 3 - ***process-centered provenance*****,** capturing the actions and steps taken to generate the information in question.
  - For example, a chart may have been generated by invoking a service to retrieve data from a database, then extracting certain statistics from the data using some statistics package, and finally processing these results with a graphing tool.

PROV Family of Specifications





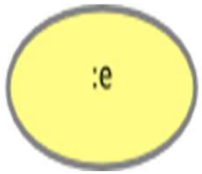
The PROV Model



- PROV is data model for provenance interchange on the Web

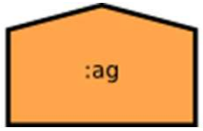
- PROV is a specification to express provenance records, which contain descriptions of the **entities** and **activities** involved in **producing** and **delivering** or otherwise **influencing** a given object

Three Core Classes + *n* Causality Relations



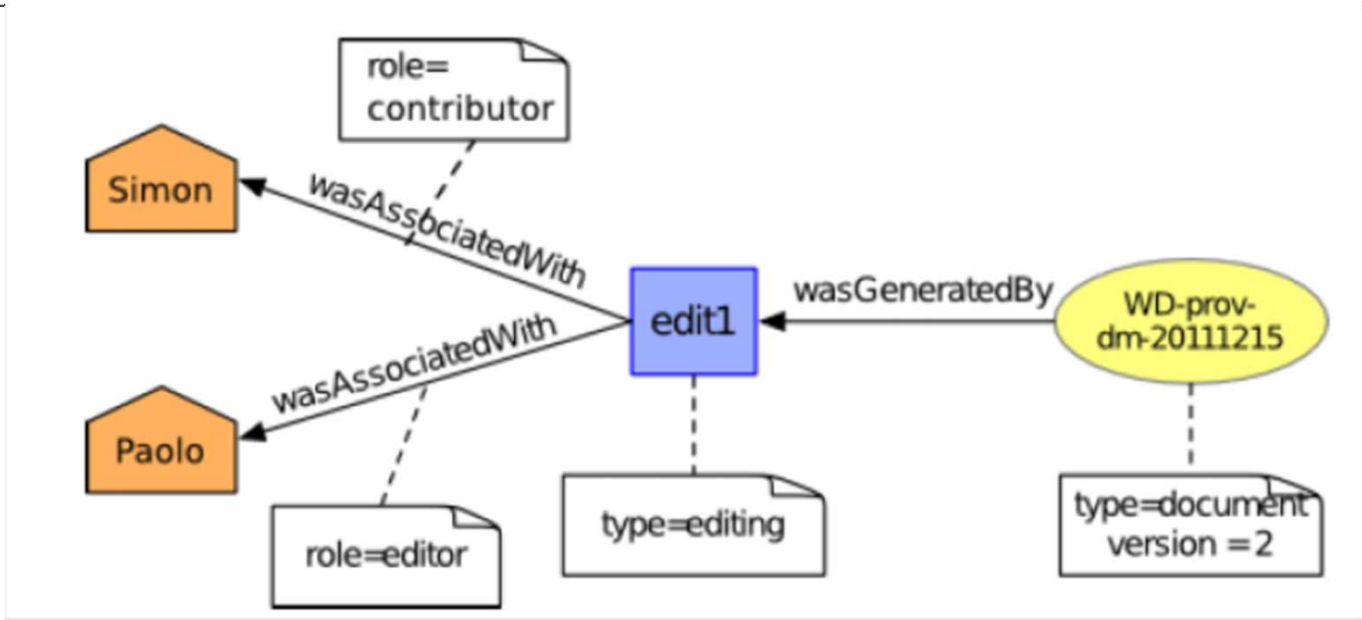
An entity is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary.

An activity is something that occurs over a period of time and acts upon or with enti- ties; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities.



An agent is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.

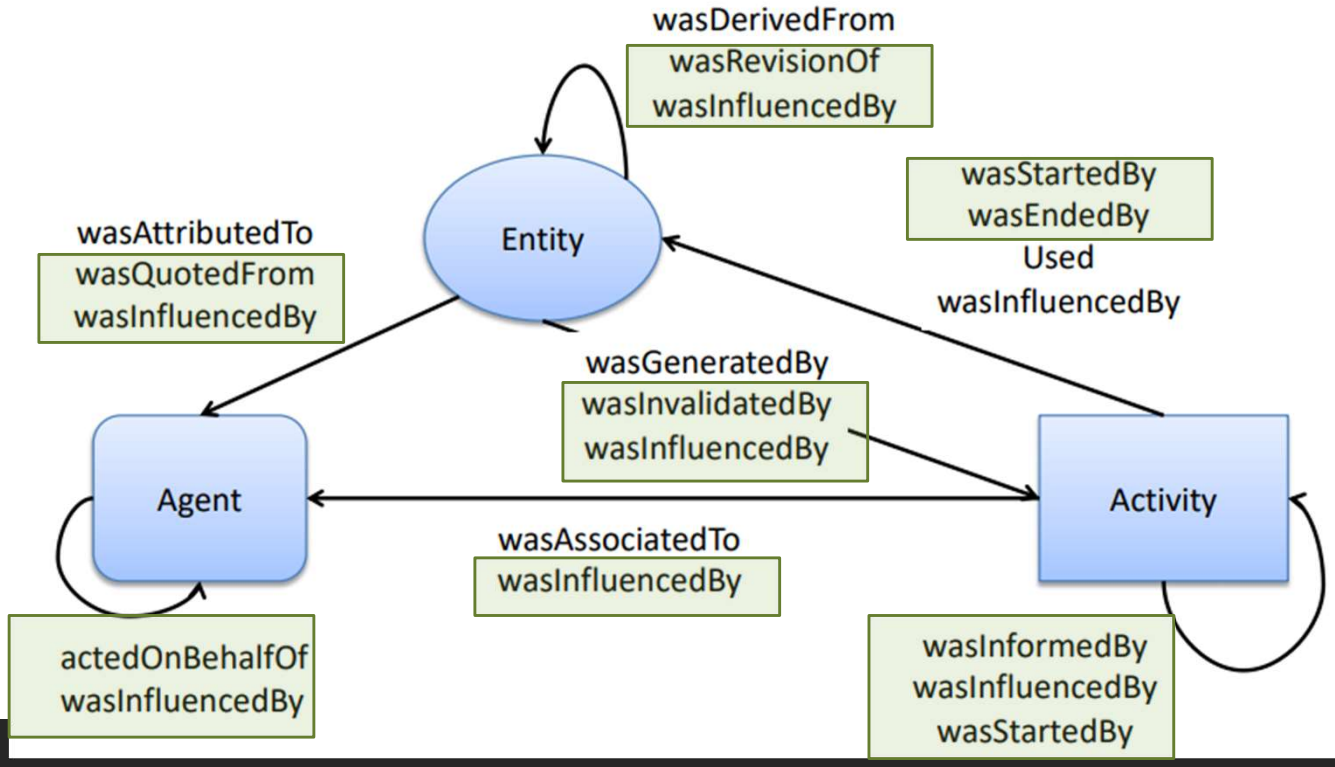
Example: Provenance graph



Provenance in PROV can be represented textually as well
The representation resembles Datalog facts , XML fragments or Turtle → RDF

PROV Model is extensible

- New properties can be added to agents, processes, entities and relationships as needed



PROV Model –*Example*

- An online newspaper publishes an article with a chart about crime statistics based on data (GovData) provided by a government portal. The article includes a chart based on the data, with data values composed (aggregated) by geographical regions.
- A blogger, Betty, looking at the article, spots what she thinks to be an error in the chart.
- Betty retrieves a record of the provenance of the article, describing how it was created.
- Betty finds the following descriptions of entities in the provenance…

Full Example → https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/

PROV Model – Entities (1)

```
exn:article      a prov:Entity ;
                 dcterms:title "Crime rises in cities" .
exg:dataset1     a prov:Entity .
exc:regionList   a prov:Entity .
exc:composition1 a prov:Entity .
exc:chart1       a prov:Entity .
```

Turtle

```
entity(exn:article, [dcterms:title="Crime rises in cities"])
entity(exg:dataset1)
entity(exc:regionList)
entity(exc:composition1)
entity(exc:chart1)
```

PROV-N

```
<prov:document>
  ...
  <prov:entity prov:id="exn:article">
    <dct:title>Crime rises in cities</dct:title>
  </prov:entity>
  <prov:entity prov:id="exg:dataset1"/>
  <prov:entity prov:id="exc:regionList1"/>
  <prov:entity prov:id="exc:composition1"/>
  <prov:entity prov:id="exc:chart1"/>
</prov:document>
```

XML

The samples use the namespace prefixes **prov** (denoting terms from the **PROV ontology),** and prefixes **exc**, **exn, exb, exg**, denoting terms specific to the example.

The example how **PROV** can be used in **combination with other languages/ontologies/vocabularies**, such as **FOAF and Dublin Core** (namespace prefix **foaf** and **dcterms** respectively).

PROV Model – Entities (2)

```
exn:article      a prov:Entity ;
                 dcterms:title "Crime rises in cities" .
exg:dataset1     a prov:Entity .
exc:regionList   a prov:Entity .
exc:composition1 a prov:Entity .
exc:chart1       a prov:Entity .
```

Turtle

Statements, refer to the article (exn:article), original data set (exg:dataset1), a list of regions (exc:regionList), data aggregated by region (exc:composition1), and a chart (exc:chart1), and state that each is an entity. Any entity may have attributes, such as the title of the article, expressed using dcterms:title above.

Different namespace prefixes were used → for the article it corresponds to the newspaper that published it (exn), for the dataset it is the government namespace (exg). The dcterms:title namespace is taken from the Dublin Core vocabulary.

PROV Model – Activity

The provenance describes that there was an activity (exc:compile1) denoting the compilation of the chart from the data set.

The provenance also includes reference to the more specific steps involved in this compilation, which are first composing the data by region (exc:compose1) and then generating the chart graphic (exc:illustrate1).

```
exc:compile1 a prov:Activity .
exc:compose1    a prov:Activity .
exc:illustrate1 a prov:Activity .
```

PROV Model – Usage & Generation

Descriptions state that the composition activity (exc:compose1) used the original data set, that it used the list of regions, and that the composed data was generated by this activity.

Similarly, the chart graphic creation activity (exc:illustrate1) used the composed data, and the chart was generated by this activity.

```
exc:compose1      prov:used           exg:dataset1 ;
                  prov:used           exc:regionList1 .
exc:composition1  prov:wasGeneratedBy exc:compose1 .
exc:illustrate1 prov:used           exc:composition1 .
exc:chart1      prov:wasGeneratedBy exc:illustrate1 .
```

PROV Model: Agents & Responsibility (1)

Digging deeper…

- Betty wants to know who compiled the chart. Betty sees that **Derek was involved in both the composition and chart creation activities**:

```
exc:compose1    prov:wasAssociatedWith exc:derek .
exc:illustrate1 prov:wasAssociatedWith exc:derek .
```

- The record for Derek provides the following description that **Derek is an agent**, specifically **a person**, **followed by non-PROV (FOAF) information giving attributes of Derek**.

```
exc:derek a prov:Agent ;
          a prov:Person ;
          foaf:givenName "Derek"^^xsd:string ;
          foaf:mbox      <mailto:derek@example.org> .
```

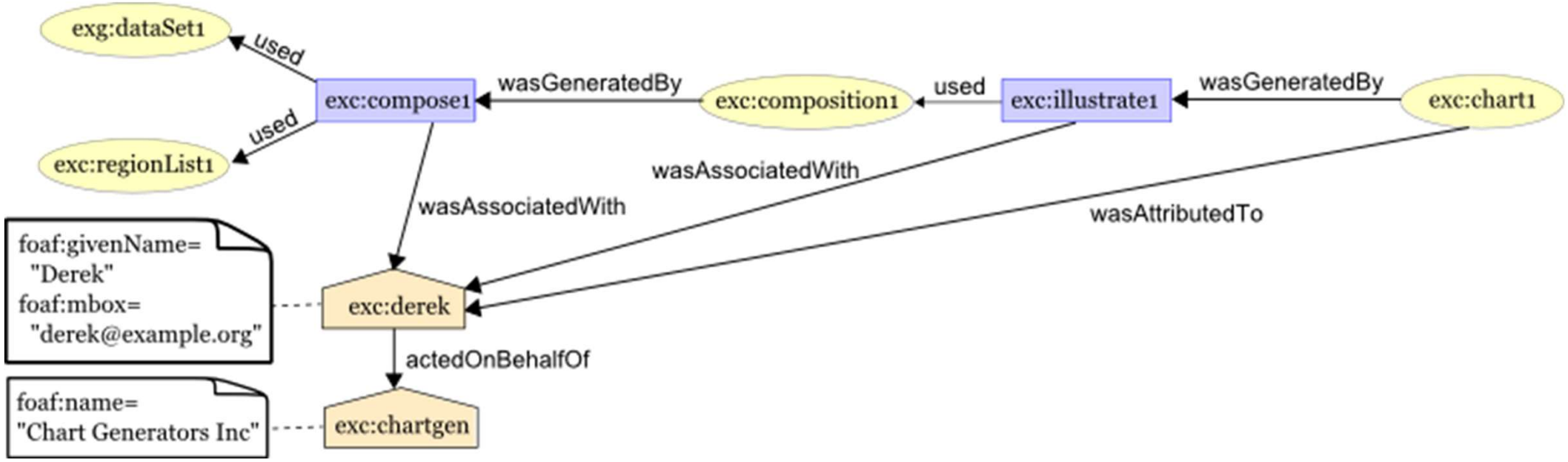
- **Derek works as part of an organization** (Chart Generators Inc), the provenance declares that he **acts on their behalf**. Note, **the organization is an agent**. The namespace prefix used by the organization is **exc**.

```
exc:derek prov:actedOnBehalfOf exc:chartgen .
exc:chartgen a prov:Agent ;
             a prov:Organization ;
             foaf:name "Chart Generators Inc" .
```

- It is possible to express the more specific statements!…Derek worked on the organization's behalf for a particular activity, rather than in general, so may have **acted on behalf of other organizations for other activities**.

```
exc:chart1 prov:wasAttributedTo exc:derek .
```

PROV Model: Agents & Responsibility (1)



PROV Model: Roles (1)

For Betty understand where the error lies, she needs more detailed information on how entities have been used in and generated by activities.

- She determined that **exc:compose1** used entities **exc:regionList1** and **exg:dataset1**, but she does not know what function these entities played in the processing.
- She knows that **exc:derek** was associated with the activities,
- She does not know if Derek was the analyst responsible for determining how the data should be composed.
- The above is described as **ROLES IN THE PROVENANCE**. The composition activity involved entities in four roles:
  - the data to be composed (**exc:dataToCompose**),
  - the regions to aggregate by (**exc:regionsToAggregateBy**),
  - the resulting composed data (**exc:composedData**),
  - the analyst doing the composition (**exc:analyst**).

```
exc:dataToCompose        a prov:Role .
exc:regionsToAggregateBy a prov:Role .
exc:composedData         a prov:Role .
exc:analyst              a prov:Role .
```

- The **usage of the data** set by the **compose activity** is expressed as…

```
exc:compose1 prov:used exg:dataset1 .
```

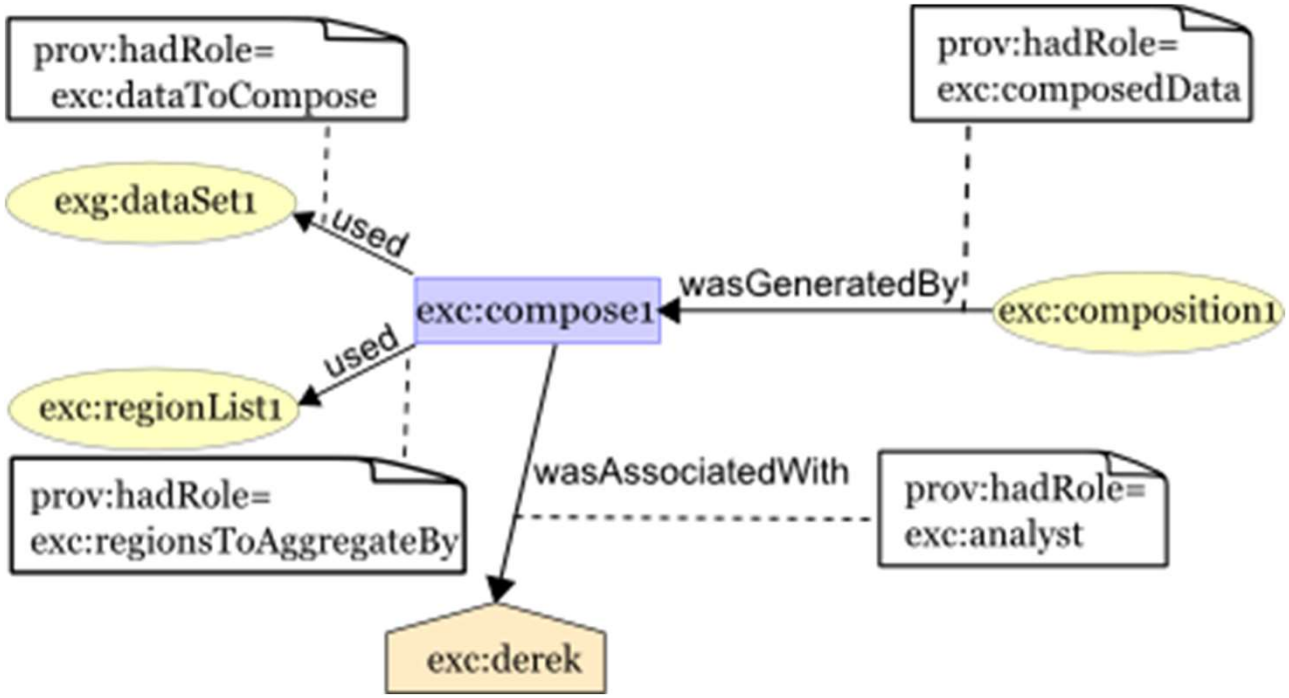
PROV Model: Roles (2)

The provenance can contain more details of exactly how these entities and agents were involved in the activity.

- PROV-O refers to **qualified usage, qualified generation, etc.**, which are descriptions consisting of several statements about how usage, generation, etc. took place.
  - For example, the qualified involvement is the role an entity played in an activity.
- The descriptions state that the composition activity (**exc:compose1**) included the usage of the government data set (**exg:dataset1**) in the role of the data to be composed (**exc:dataToCompose**).

```
exc:compose1 prov:qualifiedUsage [
             a prov:Usage ;
             prov:entity  exg:dataset1 ;
             prov:hadRole exc:dataToCompose
   ] .
```

PROV Model: Roles (3)



PROV Model: Derivation & Revision

After looking the detail of the compilation activity, there appears to be nothing wrong… **Betty concludes the error is in the government dataset!**

- She looks at the dataset **exg:dataset1**, it is missing data from one of the zipcodes in the area.
- She contacts the government, a new version of GovData is created, declared to be the next revision of the data.
- The provenance of this new dataset, **exg:dataset2**, states that it is a revision of the old data set, **exg:dataset1**.

```
exg:dataset2 a prov:Entity ;
             prov:wasRevisionOf exg:dataset1 .
```

Derek notices that there is a new dataset available and creates a new chart based on the revised data, using another compilation activity.

Betty checks the article again, wants to know if it is based on the old or new GovData.

- She sees a new description stating that the new chart is derived from the new dataset (the same relation could be expressed between the old chart and old dataset).

```
exc:chart2 a prov:Entity ;
           prov:wasDerivedFrom exg:dataset2 .
```

PROV Model: Time

The government agency is concerned to know how long the incorrect chart was in circulation before the corrected chart was created.

The agency wish to compare the times at which the original and the corrected charts were generated.

- The picture below shows that the second chart was generated a month after the first.

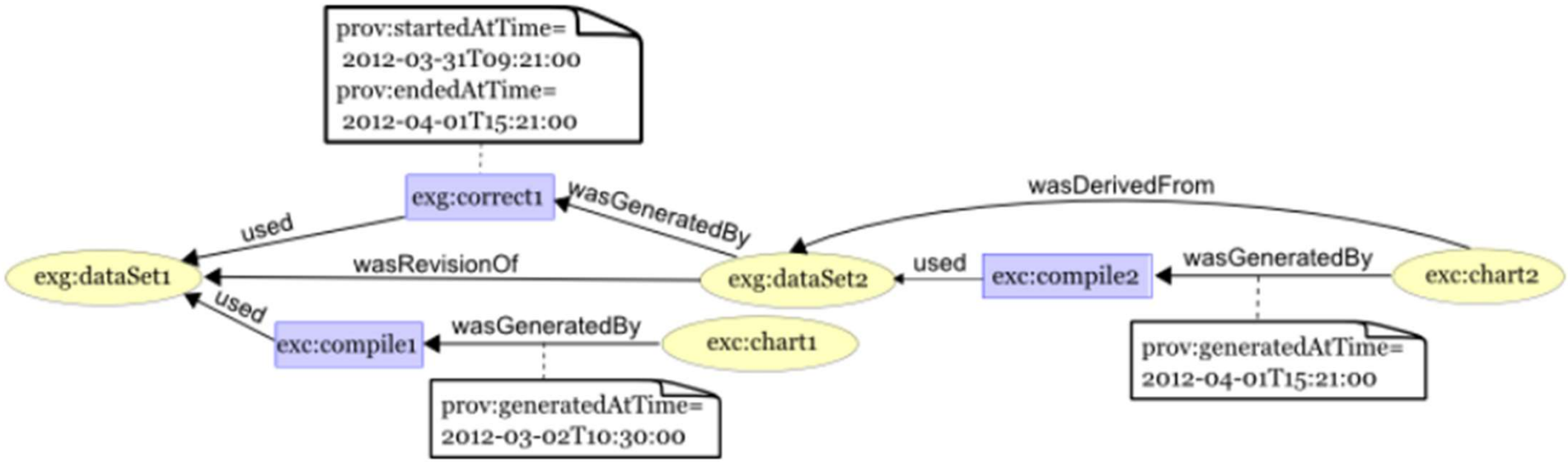
```
exc:chart1 prov:generatedAtTime "2012-03-02T10:30:00"^^xsd:dateTime .
exc:chart2 prov:generatedAtTime "2012-04-01T15:21:00"^^xsd:dateTime .
```

The agency wishes to know how long the corrections took once the error was discovered. They wish to know the start and end times of the correction activity (exg:correct1).

- These details are expressed as follows, showing that the corrections took a little over a day.

```
exg:correct1 prov:startedAtTime "2012-03-31T09:21:00"^^xsd:dateTime ;
             prov:endedAtTime   "2012-04-01T15:21:00"^^xsd:dateTime .
```

PROV Model: Time



PROV Model: Alternate Entities & Specialization

Before noticing anything wrong with the data…

- Betty had already posted a blog entry about the article. The blog entry had its own published provenance. It contains some text copied from the article, and the provenance states that this text (exb:quoteInBlogEntry-20130326) is quoted from the article (namespace prefix (exb) used for the blog.).

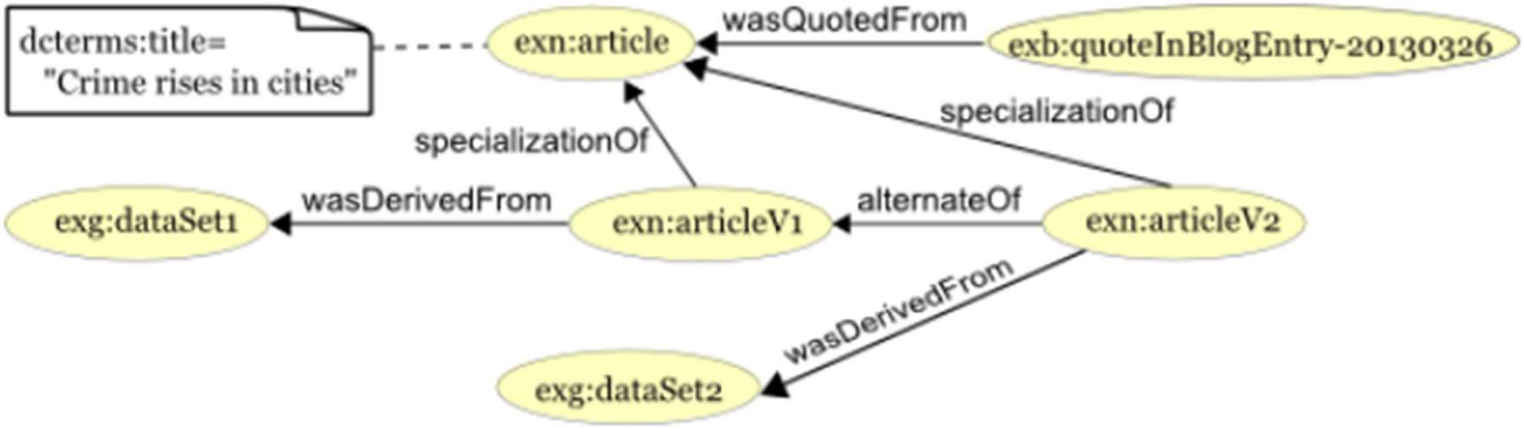
```
exb:quoteInBlogEntry-20130326 a prov:Entity ;
                              prov:value "Smaller cities have more crime than larger ones" ;
                              prov:wasQuotedFrom exn:article .
```

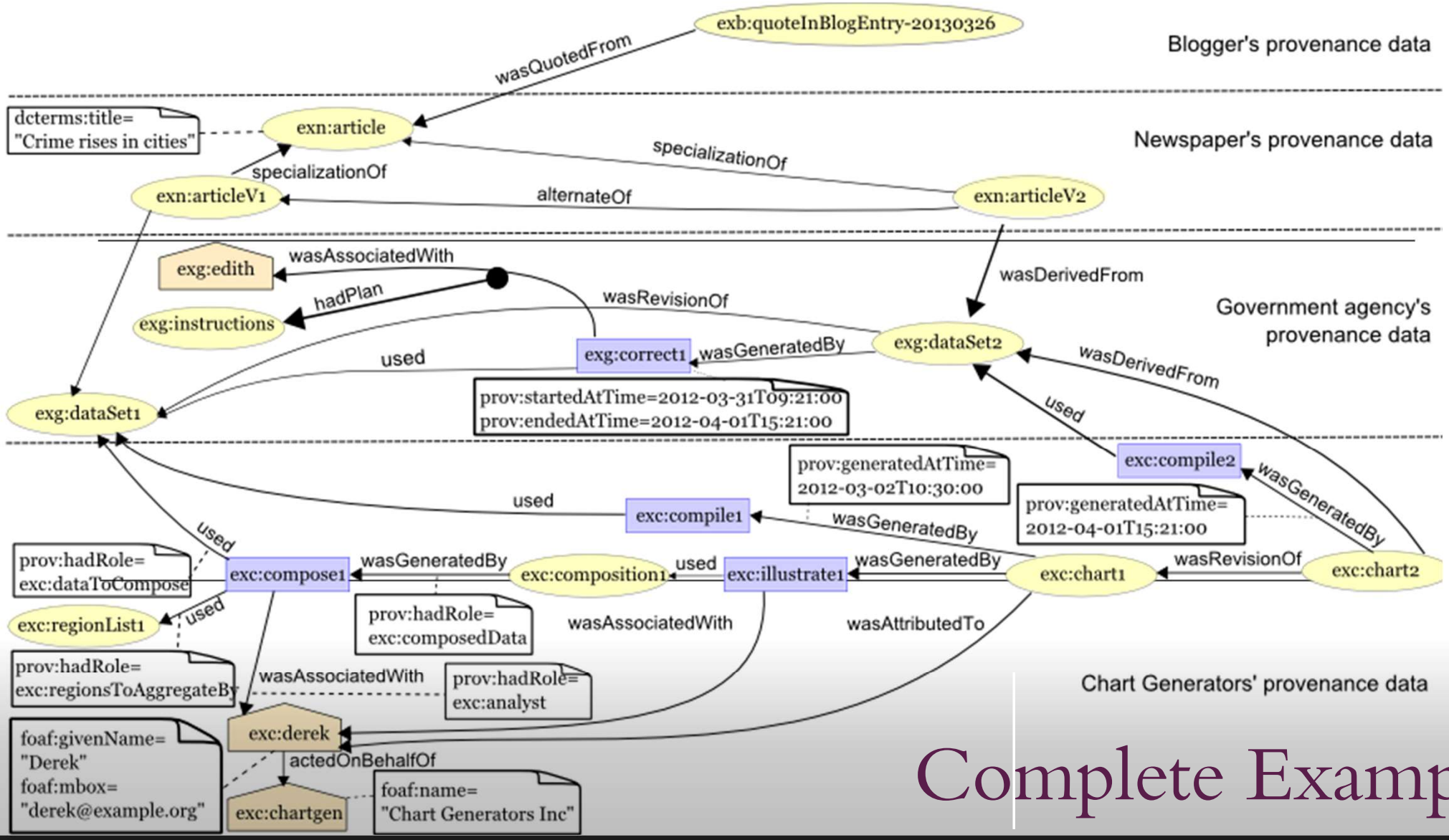
- The newspaper, from past experience, anticipated that there could be revisions to the article, created identifiers for both the article in general (exn:article) as a URI that got redirected to the first version of the article (exn:articleV1), allowing both to be referred to as entities in provenance data.

```
exn:articleV2 prov:specializationOf exn:article .
exn:articleV2 prov:alternateOf      exn:articleV1
```

PROV Model: Alternate Entities & Specialization

Note that above we could have also stated that exn:articleV2 was a revision of exn:articleV1, as we did between exc:chart2 and exc:chart1, which would describe concretely how the alternate entities are related.





Complete Example

Full Turtle code here → https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/primer-turtle-examples.ttl

Is the Data Provenance important in the Business World?

Data provenance, also known as data lineage, is a form of metadata that captures the history of data, detailing its origins, transformations, and journey through various processes. **It is a critical component of data management that ensures the integrity and reliability of data within an organization.**

Data provenance is essential for maintaining the **quality and trustworthiness of data**, which is the foundation for making informed decisions in business and technology environments.

- Data provenance provides a comprehensive history of data, including its source, how it has been altered, and by whom.
- It is **vital for regulatory compliance**, as it helps organizations demonstrate the accuracy and legitimacy of their data.
- Provenance metadata can help in identifying and **correcting errors, thus improving overall data quality**.
- It **enhances transparency and accountability** in data handling, which is crucial for cybersecurity.
- Implementing data provenance can be **challenging due to the need for cross-system tracking and standardization**.

[Data provenance] helps ensure that the data being used to make decisions is accurate and reliable.

Source: https://www.secoda.co/blog/importance-of-data-provenance

How does data provenance contribute to data governance?

Data provenance is a cornerstone of data governance, providing the necessary context to enforce policies, standards, and practices that govern the use of data within an organization.

- By documenting the lineage of data, organizations can ensure that their data governance frameworks are effective and that the data they rely on is accurate and trustworthy.
- Provenance ensures that data governance policies are supported by a traceable record of data changes and usage.
- It aids in establishing the pedigree of information, which is crucial for data stewardship and quality management.
- Data provenance supports audit trails, which are necessary for compliance with laws and regulations.

What role does data provenance play in regulatory compliance?

Data provenance plays a pivotal role in regulatory compliance by providing a verifiable trail of data's origins and modifications, which is often a requirement in legal and financial contexts.

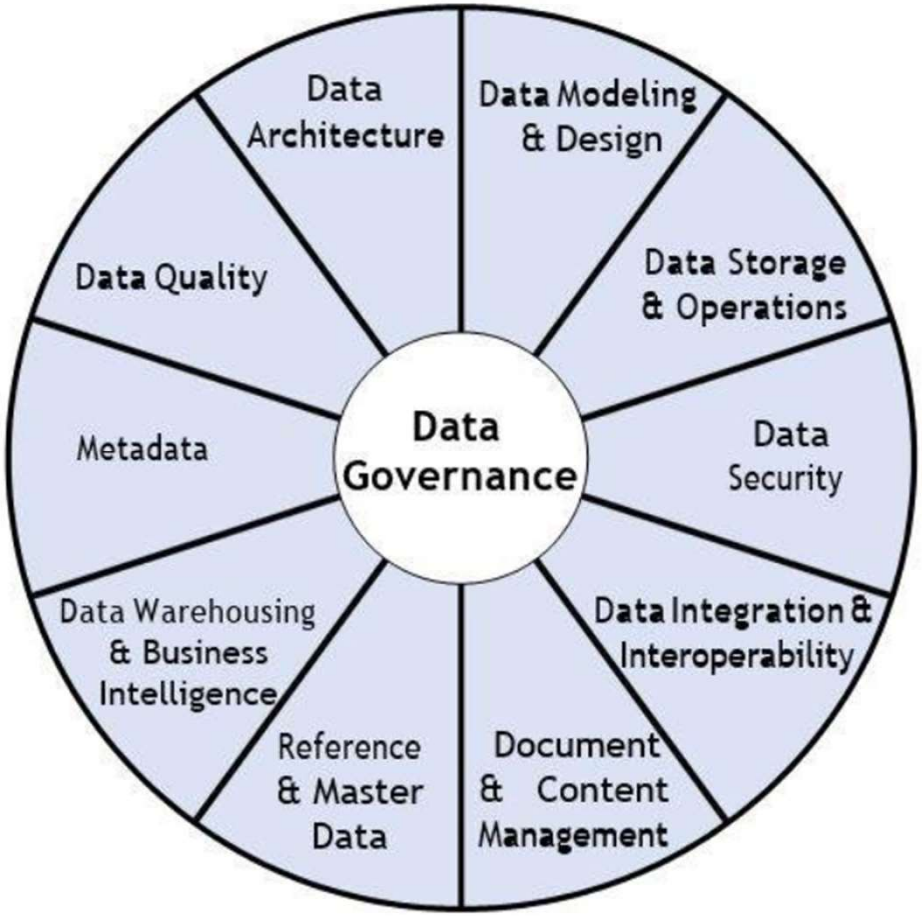
- Organizations can use provenance to demonstrate that their data handling practices meet industry standards and legal requirements, thereby avoiding penalties and maintaining their reputation.
- Provenance metadata is key for demonstrating compliance with data protection laws like GDPR and HIPAA (LGPD).
- Audit trails created by data provenance can be used to verify the integrity of financial records and transactions.
- Regulatory bodies often require detailed data lineage to ensure transparency and accountability.

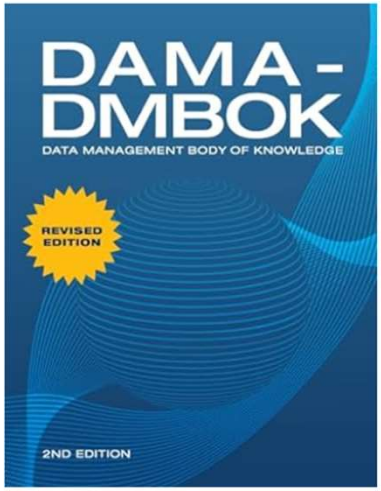
Can data provenance improve cybersecurity?

Yes, data provenance can significantly enhance cybersecurity by maintaining immutable logs that track data access and changes, which can be used to detect unauthorized or malicious activities.

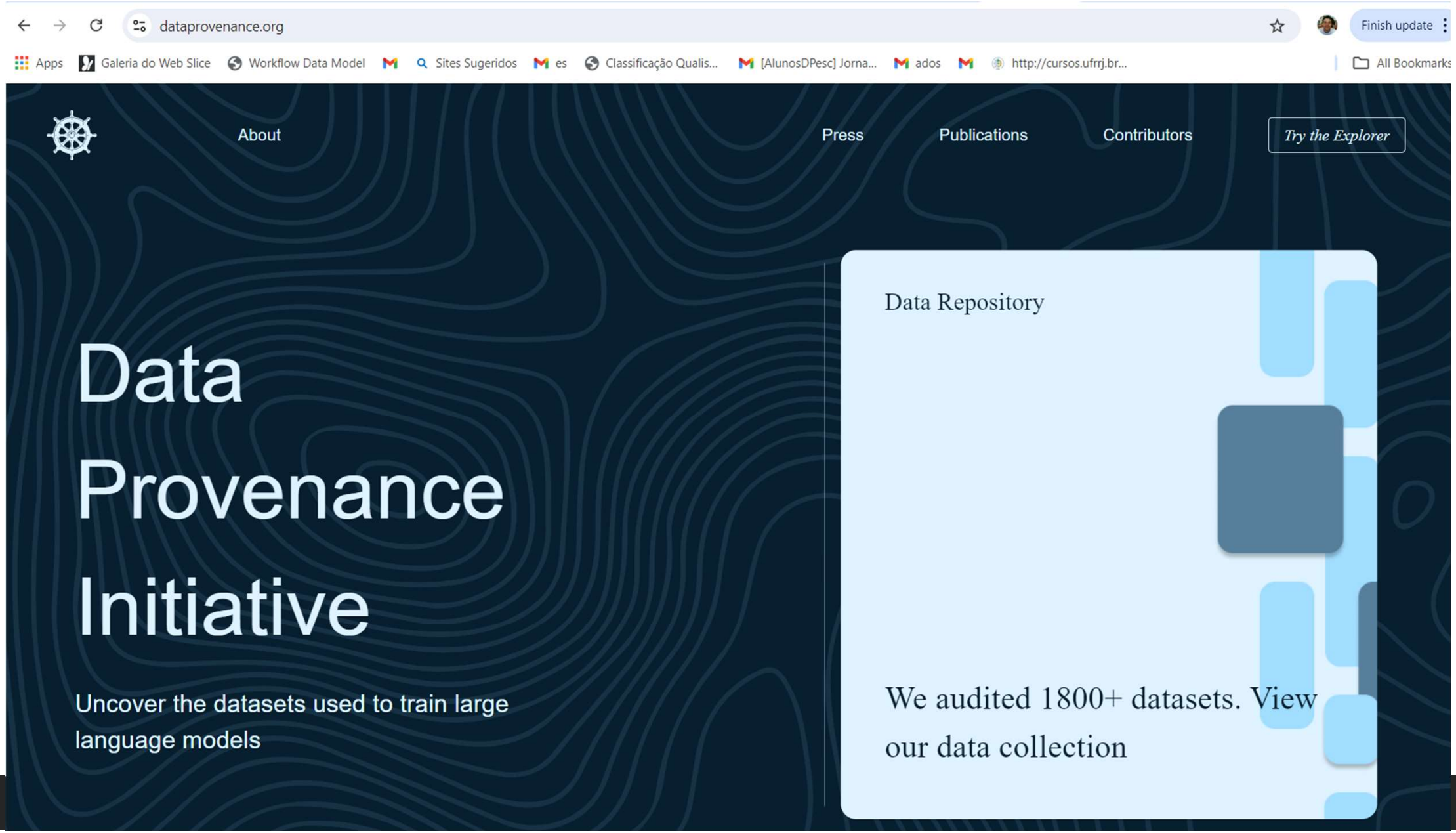
- By having a clear record of data movements and transformations, organizations can quickly identify and respond to security incidents, thereby protecting sensitive information.
- Immutable logs help in spotting unauthorized changes or access, which are indicators of potential security breaches.
- Data provenance allows for the reconstruction of events in the case of a cyber attack, aiding in forensic analysis.
- It ensures that data handling practices are transparent, making it easier to enforce security policies.

Data Provenance x Data Governance






Data
Provenance


dataprovenance.org
Finish update
Apps
Galeria do Web Slice
Workflow Data Model
Sites Sugeridos
es
Classificação Qualis...
[AlunosDPesc] Jorna...
ados
http://cursos.ufrrj.br...
All Bookmarks
About
Press
Publications
Contributors
Try the Explorer
Data Provenance Initiative
Uncover the datasets used to train large language models
Data Repository
We audited 1800+ datasets. View our data collection

Introduction

Latest Release build passing coverage 90% Code Health wheel yes python 3.6 | 3.7 | 3.8 | 3.9 license MIT

A library for W3C Provenance Data Model supporting PROV-O (RDF), PROV-XML, PROV-JSON import/export

- Free software: MIT license
- Documentation: http://prov.readthedocs.io/.
- Python 3 only.

https://github.com/trungdong/prov

Features

- An implementation of the W3C PROV Data Model in Python.
- In-memory classes for PROV assertions, which can then be output as PROV-N
- Serialization and deserialization support: PROV-O (RDF), PROV-XML and PROV-JSON.
- Exporting PROV documents into various graphical formats (e.g. PDF, PNG, SVG).
- Convert a PROV document to a Networkx MultiDiGraph and back.

Uses

See a short tutorial for using this package.

This package is used extensively by ProvStore, a free online repository

Tutorial do livro:
https://github.com/trungdong/prov

jupyter nbviewer

JUPYTER FAQ

notebooks / PROV Tutorial.ipynb

Support JSON, XML and RDF

PROV Python Library - A Short Tutorial

The PROV Python library is an implementation of the Provenance Data Model by the World Wide Web Consortium. This tutorial shows how to use the library to:

- create provenance statements in Python;
- export the provenance to PROV-N, PROV-JSON, and graphical representations like PNG, SVG, PDF; and
- store and retrieve provenance on ProvStore.

Installation

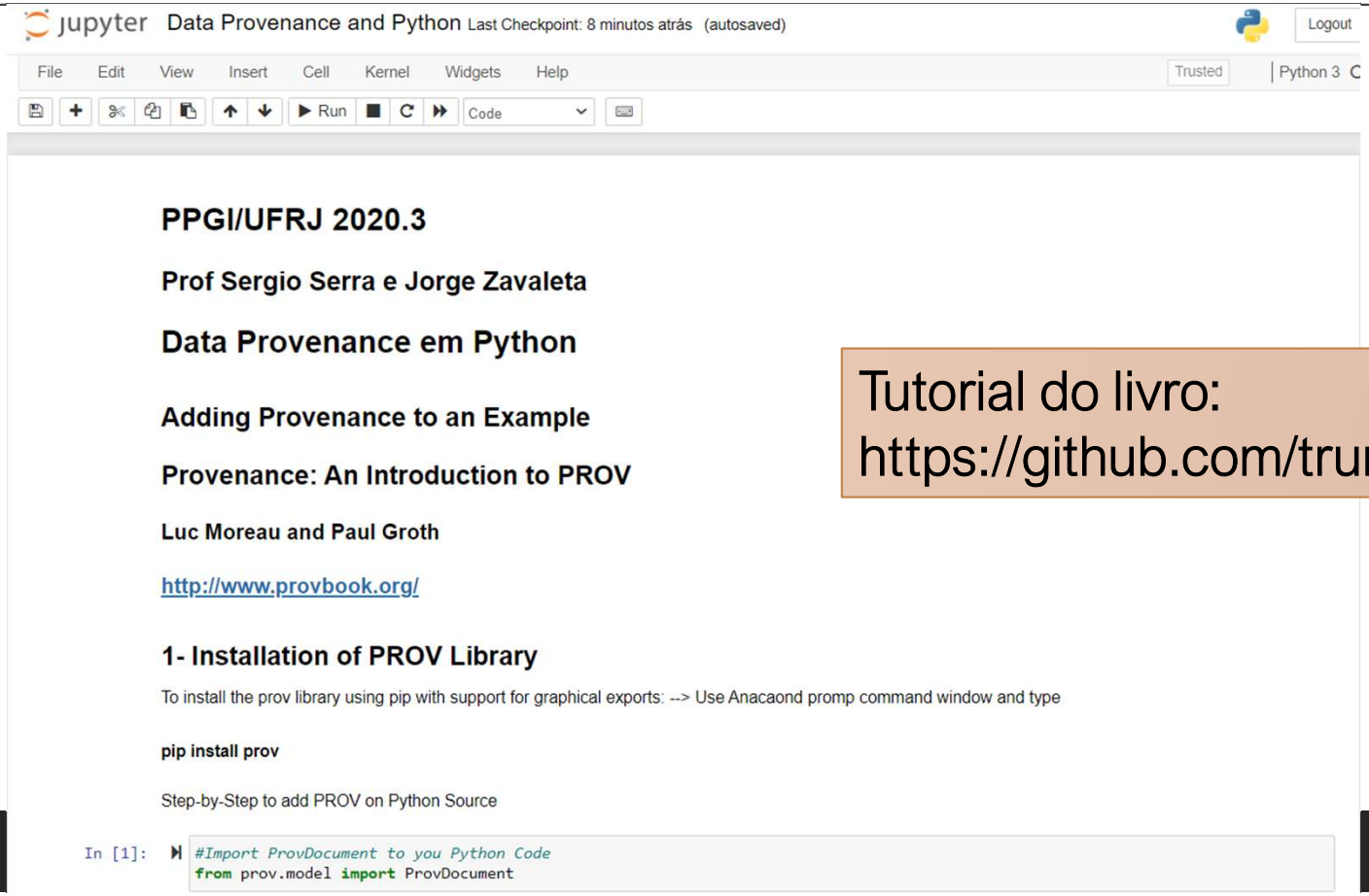
To install the prov library using pip with support for graphical exports:

```
pip install prov[dot]
```

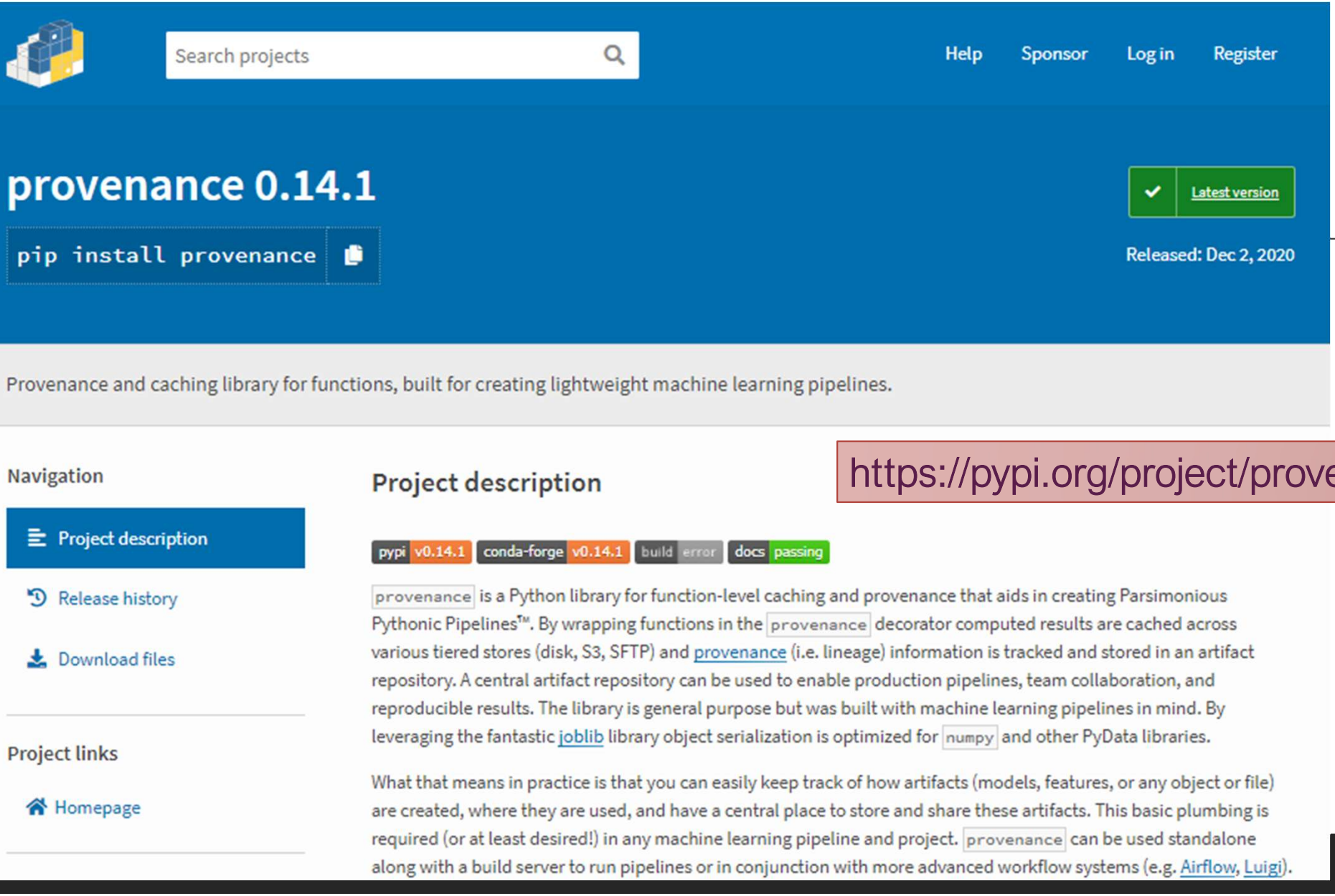
Note: We recommend using virtualenv (and the excellent companion virtualenvwrapper) to avoid package version conflicts.

No Jupyter

jupyter Data Provenance and Python



Tutorial do livro:
https://github.com/trungdong/prov


Search projects
Help
Sponsor
Log in
Register
provenance 0.14.1
pip install provenance
Latest version
Released: Dec 2, 2020
Provenance and caching library for functions, built for creating lightweight machine learning pipelines.
Navigation
Project description
Release history
Download files
Project links
Homepage
Project description
https://pypi.org/project/provenance/
pypi v0.14.1
conda-forge v0.14.1
build error
docs passing
provenance is a Python library for function-level caching and provenance that aids in creating Parsimonious Pythonic Pipelines™. By wrapping functions in the provenance decorator computed results are cached across various tiered stores (disk, S3, SFTP) and provenance (i.e. lineage) information is tracked and stored in an artifact repository. A central artifact repository can be used to enable production pipelines, team collaboration, and reproducible results. The library is general purpose but was built with machine learning pipelines in mind. By leveraging the fantastic joblib library object serialization is optimized for numpy and other PyData libraries.
What that means in practice is that you can easily keep track of how artifacts (models, features, or any object or file) are created, where they are used, and have a central place to store and share these artifacts. This basic plumbing is required (or at least desired!) in any machine learning pipeline and project. provenance can be used standalone along with a build server to run pipelines or in conjunction with more advanced workflow systems (e.g. Airflow, Luigi).

Must read ...

Provenance and the Different Flavors of Computational Reproducibility

Juliana Freire
New York University
juliana.freire@nyu.edu

Fernando Chirigati
New York University
fchirigati@nyu.edu

Abstract

While reproducibility has been a requirement in natural sciences for centuries, computational experiments have not followed the same standard. Often, there is insufficient information to reproduce computational results described in publications, and in the recent past, this has led to many retractions. Although scientists are aware of the numerous benefits of reproducibility, the perceived amount of work to make results reproducible is a significant disincentive. Fortunately, much of the information needed to reproduce an experiment can be obtained by systematically capturing its provenance. In this paper, we give an overview of different types of provenance and how they can be used to support reproducibility. We also describe a representative set of provenance tools and approaches that make it easy to create reproducible experiments.

1 Introduction

The need to reproduce experiments to verify and extend them is not new in science. Revisiting and reusing past results – or as Newton once said, "standing on the shoulders of giants" – is the standard paradigm of all sciences. Unfortunately, achieving reproducibility has proved elusive for computational experiments, which, due to the explosion in the volume of available data and widely accessible computing infrastructure, have become an integral component of science in many different domains.

Scientific papers published in conferences and journals present a large number of tables, plots, and beautiful pictures that summarize the obtained results, but that loosely describe the steps taken to derive them [16,38]. Not only can the methods and the implementation be complex, but their configuration may require setting myriad parameters. Consequently, reproducing the results from scratch is both time-consuming and error-prone at best, and sometimes impossible.

Reproducibility of computational experiments across platforms and time brings a range of benefits to science. First, reproducibility enables reviewers to test the outcomes presented in papers. This is specially important given the growing concern that many spurious research findings are published in respected venues [5,12,29], which is reflected in the increasing number of paper retractions [44, 58]. Second, it allows new methods to be objectively compared against methods presented in reproducible publications. Third, researchers are able to build on top of previous work directly. Last but not least, recent studies indicate that reproducibility increases impact, visibility, and research quality [3, 7, 26, 36, 50, 59] and helps defeat self-deception [46].

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

FREIRE, J.; CHIRIGATI, F. Provenance and the Different Flavors of Computational Reproducibility. IEEE Data Engineering Bulletin. V. 41:15-26, 2018.

frontiers in Neuroinformatics

PROTOCOLS
published: 04 January 2018
doi: 10.3389/fninf.2017.00069

Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions

Fabien C. Y. Benureau[1,2,3]* *and Nicolas P. Rougier*[1,2,3]

[1] INRIA Bordeaux Sud-Ouest, Talence, France, [2] Institut des Maladies Neurodégénératives, Université de Bordeaux, Centre National de la Recherche Scientifique UMR 5293, Bordeaux, France, [3] LaBRI, Université de Bordeaux, Bordeaux INP, Centre National de la Recherche Scientifique UMR 5800, Talence, France

Scientific code is different from production software. Scientific code, by producing results that are then analyzed and interpreted, participates in the elaboration of scientific conclusions. This imposes specific constraints on the code that are often overlooked in practice. We articulate, with a small example, five characteristics that a scientific code in computational science should possess: re-runnable, repeatable, reproducible, reusable, and replicable. The code should be executable (re-runnable) and produce the same result more than once (repeatable); it should allow an investigator to reobtain the published results (reproducible) while being easy to use, understand and modify (reusable), and it should act as an available reference for any ambiguity in the algorithmic descriptions of the article (replicable).

Keywords: replicability, reproducibility of results, reproducible science, reproducible research, computational science, software development, best practices

OPEN ACCESS

Edited by:
Sharon Crook,
Arizona State University, United States

Reviewed by:
Thomas E. Nichols,
Independent Researcher, Oxford, United Kingdom
Paul Pavlidis,
University of British Columbia, Canada
Thomas Marston Morse,
Department of NeuroScience, Yale School of Medicine, Yale University, United States

***Correspondence:**
Fabien C. Y. Benureau
fabien@benureau.com

***Received:** 28 August 2017*
***Accepted:** 17 November 2017*
***Published:** 04 January 2018*

Citation:
Benureau FCY and Rougier NP (2018) Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Front. Neuroinform. 11:69. doi: 10.3389/fninf.2017.00069

INTRODUCTION (R^0)

Replicability[1] is a cornerstone of science. If an experimental result cannot be re-obtained by an independent party, it merely becomes, at best, an observation that may inspire future research (Mesirov, 2010; Open Science Collaboration, 2015). Replication issues have received increased attention in recent years, with a particular focus on medicine and psychology (Iqbal et al., 2016). One could think that computational research would mostly be shielded from such issues, since a computer program describes precisely what it does and is easily disseminated to other researchers without alteration.

But precisely because it is easy to believe that if a program runs once and gives the expected results it will do so forever, crucial steps to transform working code into meaningful scientific contributions are rarely undertaken (Schwab et al., 2000; Sandve et al., 2013; Collberg and Proebsting, 2016). Computational research is plagued by replication problems, in part, because it seems impervious to them. Contrary to production software who provides a service geared toward a practical outcome, the motivation behind scientific code is to test a hypothesis. While in some instance production software and scientific code are indistinguishable, the reasons why they were created are different, and, therefore, so are the criteria to evaluate their success. A program

[1] *Reproducibility* and *replicability* are employed differently by different authors and in different domains (see for instance the report from the U.S. National Academies of Sciences, 2016). Here, we place ourselves in the context of computational works, where data is produced by a program. In this paper, we call a result *reproducible* if one can take the original source code, re-execute it and reobtain the original result. Conversely, a result is *replicable* if one can create a code that matches the algorithmic descriptions given in the published article and reobtain the original result.

BENUREAU, F., ROUGIER, N. *Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions.* Frontiers in Neuroinformatics. V.11, article 69, 2018.

References

Benureau, F., Rougier, (2018) N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in Neuroinformatics. V.11, article 69.

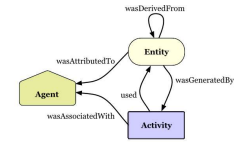
Cruz et al., (2009) Towards a Taxonomy of Provenance in Scientific Workflow Management Systems - June DOI: 10.1109/SERVICES-I.2009.18

Freire, J.; Chirigati, F. (2018). Provenance and the Different Flavors of Computational Reproducibility. IEEE Data Engineering Bulletin. V. 41:15-26

Mattoso, M., et al.. (2010) Towards Supporting the Life Cycle of Large-Scale Scientific Experiments. International Journal of Business Process Integration and Management, v. 5, p. 79-92.

Moreau L. and Groth P. (2013). Provenance: An Introduction to PROV. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool.

Pimentel et al. (2019). A Survey on Collecting, Managing, and Analyzing Provenance from Scripts. ACM Comput. Surv., Vol. 1, No. 1, Article 1. Publication date: January 2019



DATA FLOW

ARTICLE FLOW

AUTHOR MANAGED FUNCTIONS

Generate and prepare data files

Write experimental article

Author includes data citation with article submission

Deposit dataset to repository

Submit manuscript to publisher

Article reviewers access data on repository

Data checked and curated

Article and data reviewed

REPOSITORY MANAGED FUNCTIONS

PUBLISHER MANAGED FUNCTIONS

Findable Accessible Interoperable Reusable

Discoverable links via DOIs

Published dataset

Published article

Fonte: https://data.research.cornell.edu/content/preparing-fair-data-reuse-and-reproducibility

Extra : Turtle

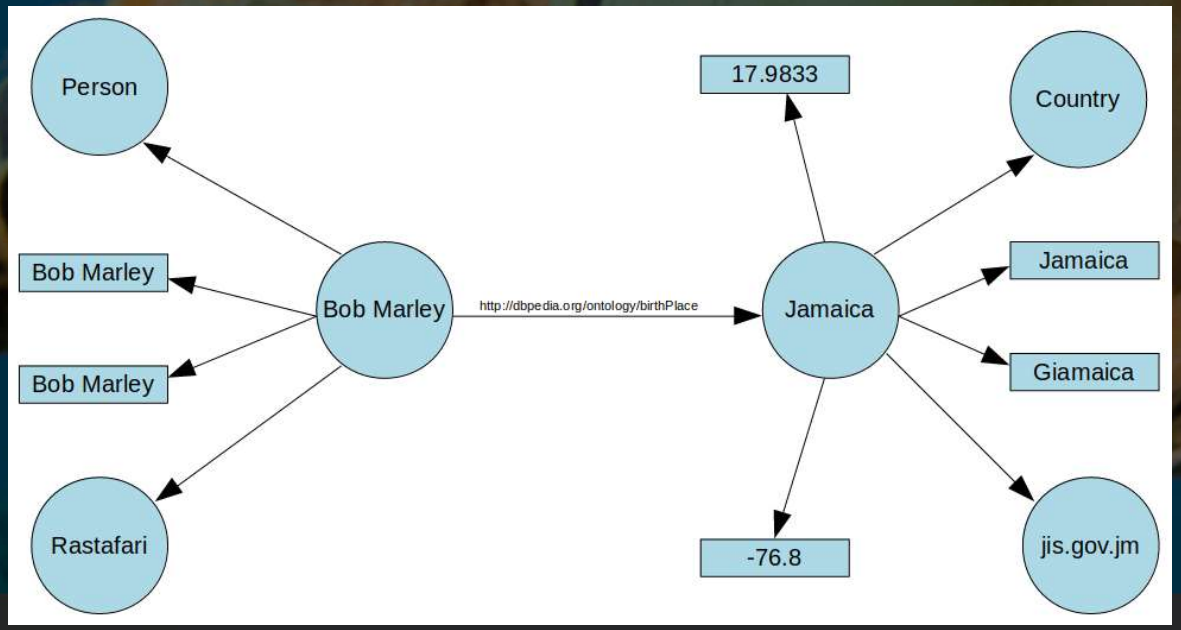
RDF is commonly stored in one of four formats in RDF libraries and **triplestores**.

- N-Triples (.nt),
- **Turtle (.ttl),**
- JSON-LD (.json)
- RDF/XML (.rdf).

Advantages

- Reading (**as a human**) RDF in Turtle format is much easier as you can define prefixes at the beginning of the .ttl file, shortening each triple.
- Another feature of turtle is that multiple triples with the same subject are grouped into blocks

Extra : Turtle




```
@prefix dbr:    <http://dbpedia.org/resource/> .
@prefix dbo:    <http://dbpedia.org/ontology/> .
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .
@prefix geo:    <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix xsd:    <http://www.w3.org/2001/XMLSchema#> .
@prefix schema: <http://schema.org/> .

dbr:Bob_Marley
   a foaf:Person ;
   rdfs:label "Bob Marley"@en ;
   rdfs:label "Bob Marley"@fr ;
   rdfs:seeAlso dbr:Rastafari ;
   dbo:birthPlace dbr:Jamaica .

dbr:Jamaica
   a schema:Country ;
   rdfs:label "Jamaica"@en ;
   rdfs:label "Giamaica"@it ;
   geo:lat "17.9833"^^xsd:float ;
   geo:long "-76.8"^^xsd:float ;
   foaf:homepage <http://jis.gov.jm/> .
```