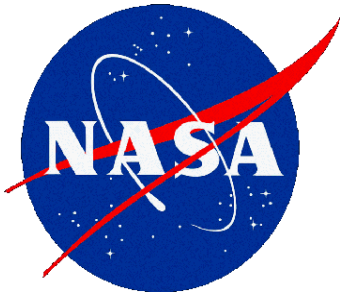


Provenance Challenges for Earth Science Dataset Publication

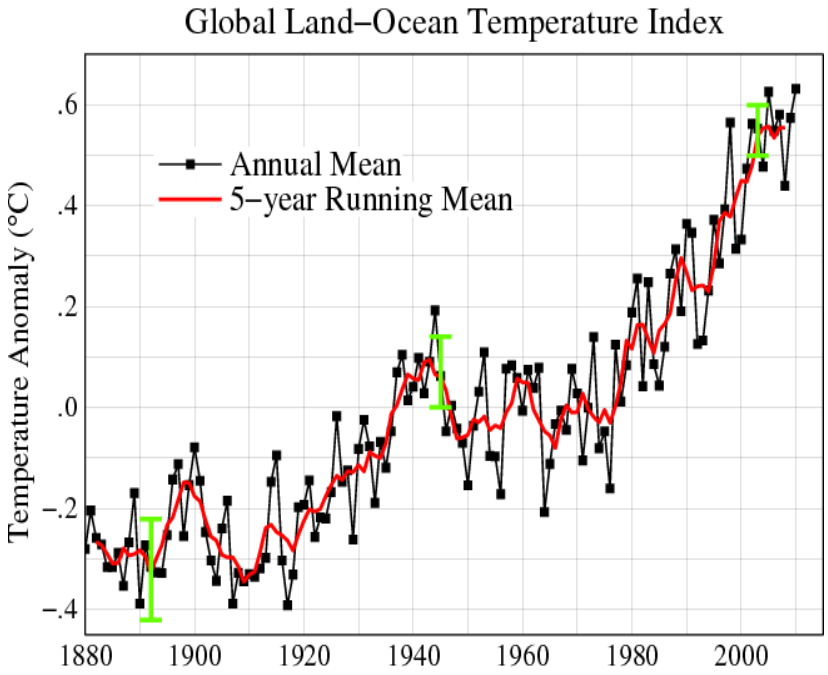
Curt Tilmes

Curt.Tilmes@nasa.gov

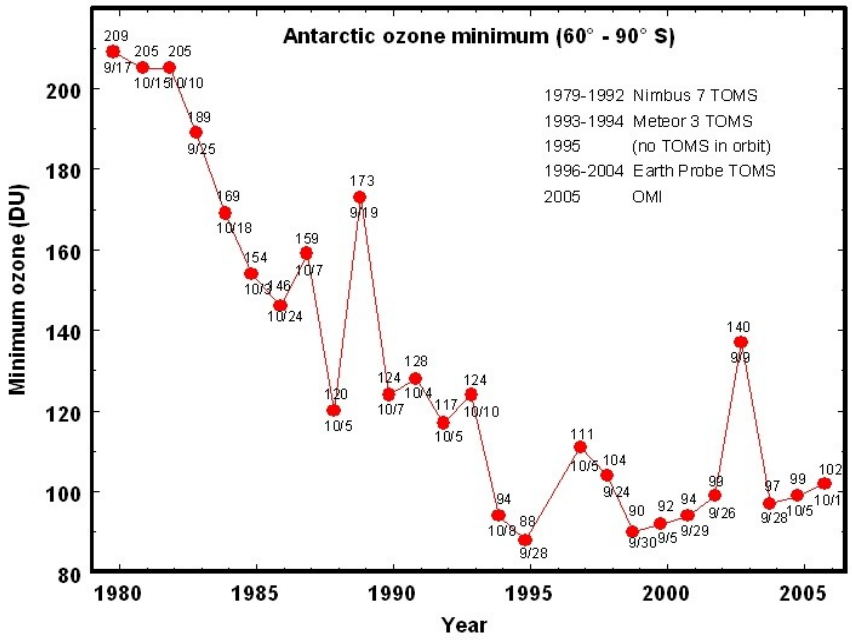
OGK 2011
2011-11-04

- "An inherent principle of publication is that others should be able to *replicate* and build upon the authors' published claims. Therefore, a condition of publication in a Nature journal is that authors are required to make materials, data and associated protocols available in a publicly accessible database [...] or, where one does not exist, to readers promptly on request."
  - *(Guide to Publication Policies of the Nature Journals, 2007)*
- Science must be reproducible
  - *(or it isn't science...)*
- Traditionally, one could read a scientific paper, construct an identical experiment and confirm results
  - *(well, most of the time...)*
- *Reproducibility* yields *Credibility*

- Some modern scientific research is the result of lengthy computer analysis of a **very large** amount of data, building on the contributions of hundreds (thousands?) of individuals

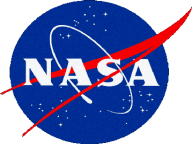


http://data.giss.nasa.gov/gistemp/graphs/



http://jwocky.gsfc.nasa.gov/eptoms/dataqual/ozone_v8.html

When scientific research is published, it should *reference* all data used in that research to a sufficient extent for *others* to *reproduce* that research and confirm the conclusions.

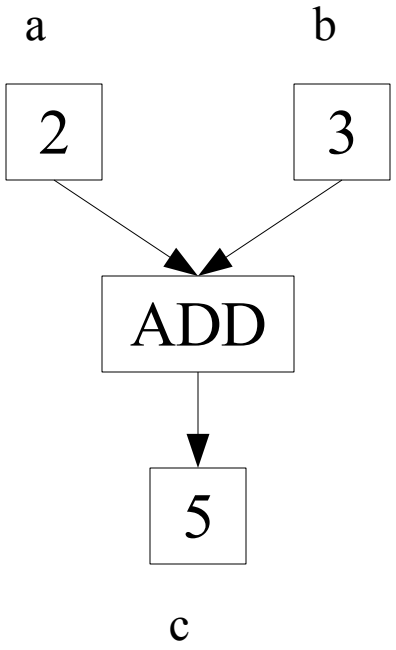


Provenance and Context Artifacts

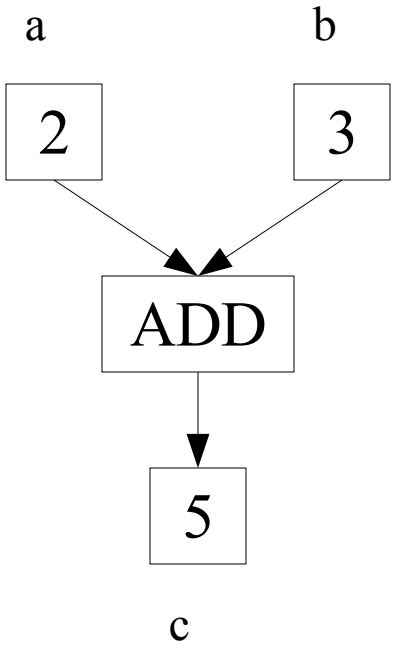
- All of the "artifacts" involved or related to the scientific result:
  - Data
  - Algorithms, Processes, Configuration Tables, Runtime Parameters ("Workflow Provenance")
  - Documentation (ATBDs, Design Docs, Commented Source)
  - Sensors/Instruments/Instrument platforms
  - People/Organizations (reputation)
  - Published scientific papers (add to credibility and understanding)
  - Computer systems, Hardware, OS, Libraries, Software
  - Abstract things like "a data transformation event," "Software Build Event" or "a validation experiment"
  - An ephemeral execution of a web service
  - Versions from all of the above: Rigorous Configuration Management.
  - Specific relationships between all the artifacts.
- Things that increase *understanding* and enable *reproducibility*.

- Basic configuration management works well for software.
- Any time the software is changed, we tag a snapshot with a revision number (v. 1.2.3) through our CM tools. – We can go back and check out that version of the software, compare versions, etc.
- Data versioning is more complicated. The direct predecessors and the software that produced a given granule could have the same version, but due to changes 'up-stream' in the workflow, the data are different.
- We frequently perform large-scale reprocessing with improved algorithms and discard older data – even if they are the basis of published research. (!)

- What aspects of the provenance are “essential” for reproducibility?
- Some things are definitely “essential”
  - Workflow artifacts – inputs, runtime parameters
- Some things are definitely “non-essential”
  - Name of processing host, who ran the process, date of processing
  - These are useful for auditing and increase credibility of provenance.
- Some things aren't so clear
  - Heinrich Hertz testing Maxwell's Equations – didn't report the size of the room he worked in – turned out to be “essential”
  - Compiler Flags? Library Versions? OS architecture?
- Can we differentiate “creation” provenance from “acquisition” provenance?

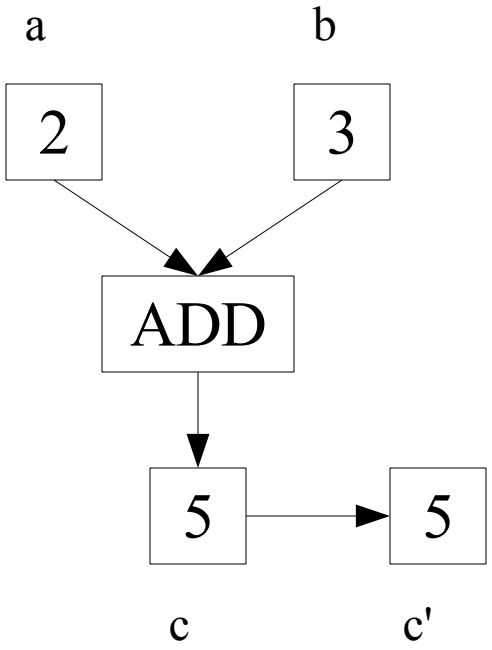


Granule “c” was created by applying process ADD to input granules “a” and “b”



Granule “c” was created by applying process ADD to input granules “a” and “b”

Joe performed this operation on Feb 2, 2011

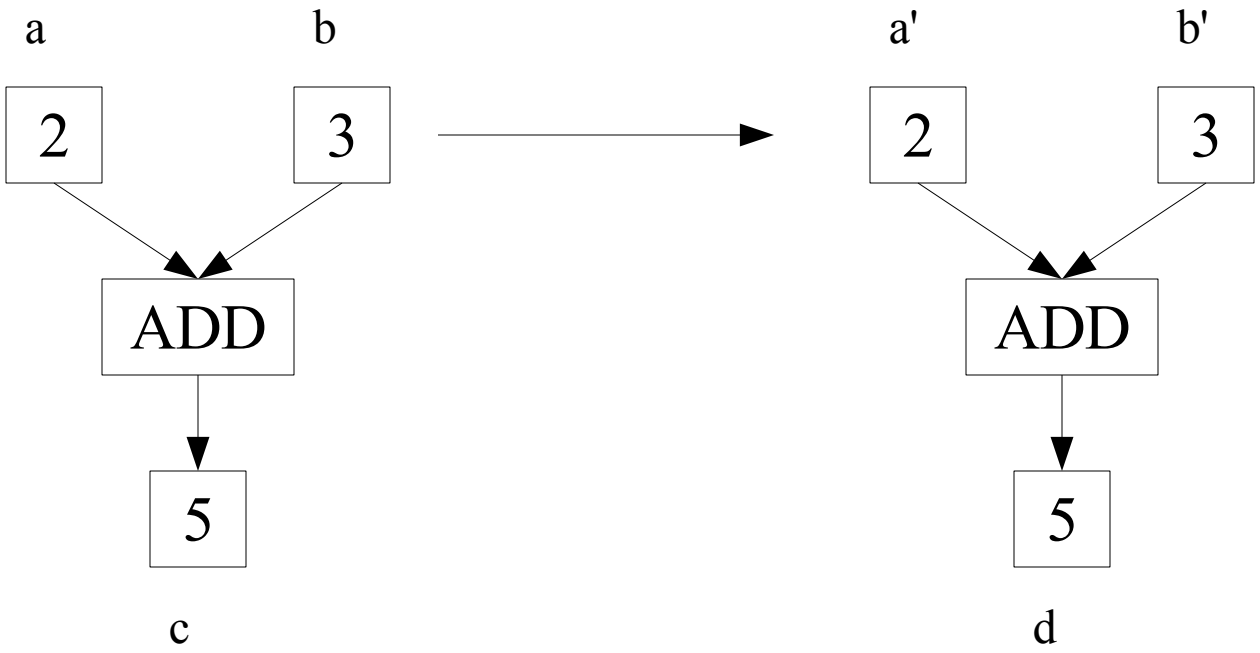


Granule “c” was created by applying process ADD to input granules “a” and “b”

Joe performed this operation on Feb 2, 2011

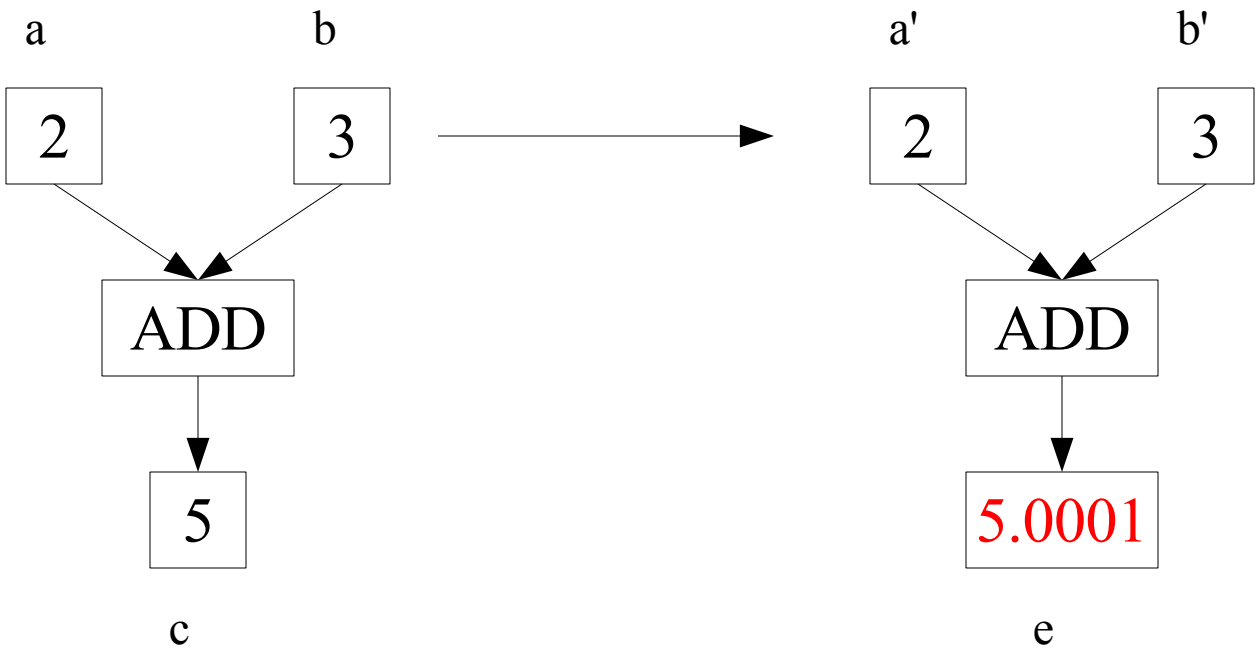
Fred downloaded granule c from Joe's archive on Feb 5, 2011

c and c' are 'identical' granules.



Sue downloaded a and b, and re-ran process ADD on them producing granule d

d has equal content to c



Sue downloaded a and b, and re-ran process ADD on them producing granule e

Her environment has slight differences, so the content is slightly off...

e does not have equal content to c. It may be 'equivalent'.

- For two granules of data to be *Perfectly Identical*, they must not only have identical contents, but also identical identifiers and identical creation provenance. This is only meaningful if you really are talking about the same granule, or two 'copies' of the same granule.

- Two granules have *Scientifically Equivalent Content* if the use of those granules in a scientific analysis will lead to the same results or conclusions.

- This definition allows 'slight' differences in the content – as long as they are close enough not to affect any analysis in a scientifically meaningful way.

- Proving perfect Scientific Equivalence in the general case is very difficult (impossible?), or at the least, very manual.

- *Scientifically Reproducible* refers to a process which is capable of reproducing granules that are *Scientifically Equivalent* to the original granules. *Scientific Reproducibility* is the extent to which a process is *Scientifically Reproducible*.

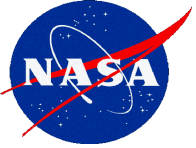
- Some processes are chaotic in that very slight differences in processing are compounded possibly producing drastically different results. We can apply sensitivity analyses to assess this characteristic and help determine if the process is suitably reproducible.

- If a process is unable to reliably reproduce data granules that are *scientifically equivalent*, we would claim that the process is not *reproducible*.

- There are two primary approaches for mechanically approximating this equivalence in a useful way:

  - Content Equivalence – Can I show that the contents of two granules are sufficiently equivalent?

  - Provenance Equivalence – Can I show that two granules were *created* in *essentially* the same way?



Provenance Equivalence

- We propose a *Provenance Equivalence Identifier* (PEI), created with a digital signature from a canonical serialization of the *essential* provenance of the granule.

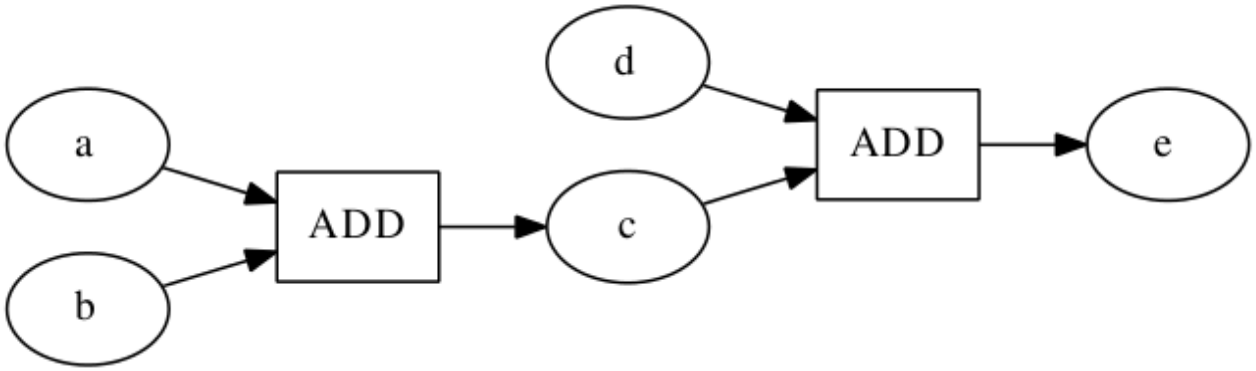
- Each granule sharing a PEI is made in a sufficiently similar manner (they share all *essential provenance* elements) that they are *scientifically equivalent*.

- Some granules come from 'outside' our processing system's scope. If they already have a PEI assigned to them --- great --- if not, we need to 'prime the pump'.
- Calculate a digital signature / hash of the content of the granule, and use that as the PEI.
- Independent systems that get the same granule will produce the same PEI for that granule.

- The PEI for each subsequent data granule is a hash of a canonical serialization of the essential provenance for that granule.

- For our demonstration implementation, and the examples here, we simplify to three things:
  - Runtime Parameters – these can change the manner of execution of the APP, environment variables, command line arguments, APP identifier, APP version
  - Input Granules – the PEIs of all other input files to the process. The order must be the same.
  - Output Granule Distinguisher – If there are more than one output file, we use a serial number to guarantee a distinct PEI.

- Simple workflow adding some numbers.



- a,b,d are leaf granules:

$$\mathbf{PEI}(a) = \texttt{401b30e3b8b5d629635a5c613cdb7919}$$

$$\mathbf{PEI}(b) = \texttt{009520053b00386d1173f3988c55d192}$$

$$\mathbf{PEI}(d) = \texttt{e29311f6f1bf1af907f9ef9f44b8328b}$$

- Construct a Provenance Equivalence File (PEF) to calculate the PEI of c:

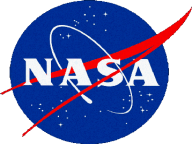
```
APP: ADD
APPVersion: 1.0
Inputs:
  - 401b30e3b8b5d629635a5c613cdb7919
  - 009520053b00386d1173f3988c55d192
Output: 1
```

$$\mathbf{PEI}(c) = \texttt{a84c0efc1873b527e6d25f380da7bcf1}$$

- Construct a PEF and calculate the PEI of e:

```
APP: ADD
APPVersion: 1.0
Inputs:
  - a84c0efc1873b527e6d25f380da7bcf1
  - e29311f6f1bf1af907f9ef9f44b8328b
Output: 1
```

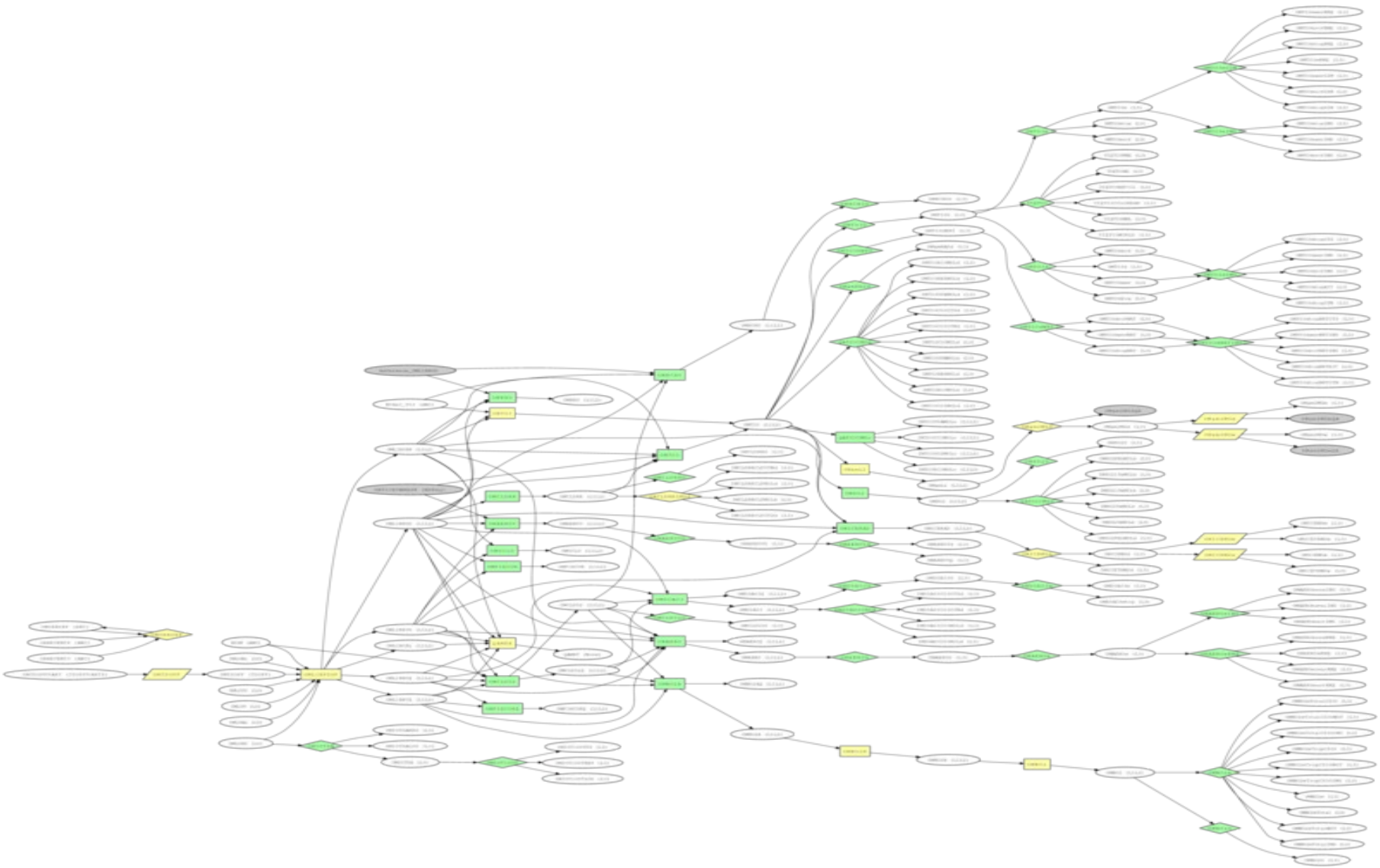
$$\mathrm{PEI}(e) = \texttt{cbedcb426502400ecf4f40a7dd7de89f}$$



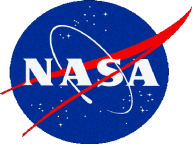
Provenance Equivalence Identifiers

- IF a process is reproducible, we can determine the essential provenance for the process.
- IF we repeat a reproducible process with identical essential provenance, we will get a scientifically equivalent granule.
- The PEI can be used as a proxy for the essential provenance graph that led to the creation of that data granule.
- Two granules with the same PEI will be scientifically equivalent to one another, even if their content varies slightly.

OMI Data Flow



- We can follow the provenance equivalence through multiple layers of production.
- Indexing the database on the PEI allows the system to locate equivalent granules.
- When portions of the data are removed, we can determine use the metadata and provenance database to determine the “essential provenance” using equivalence of predecessor files rather than requiring the exact files.
- The system can use “process on demand” to remake previous data, and assert its equivalence to the original.



Thank You!