Towards a computer-interpretable actionable formal model to encode data governance rules

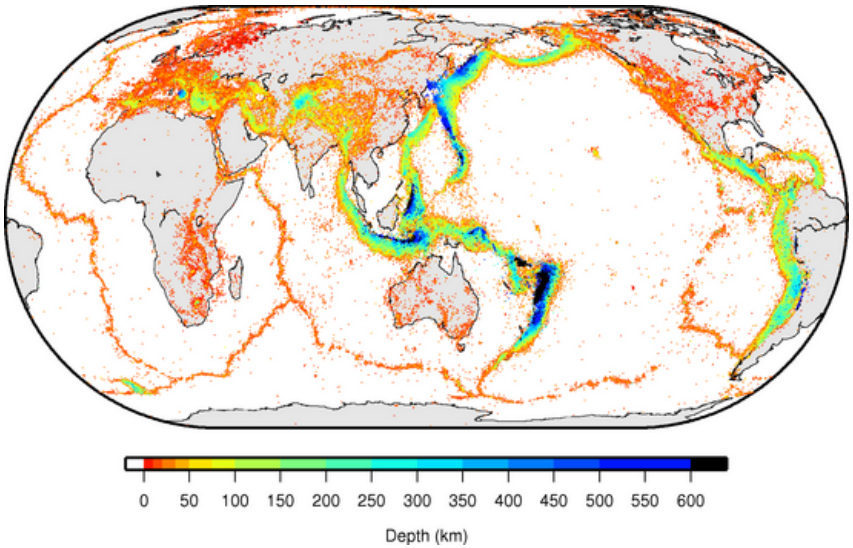
Rui Zhao & Malcolm Atkinson

Centre for Intelligent Systems and their Applications
School of Informatics
University of Edinburgh

24 September 2019

Outline

Data-sharing collaboration



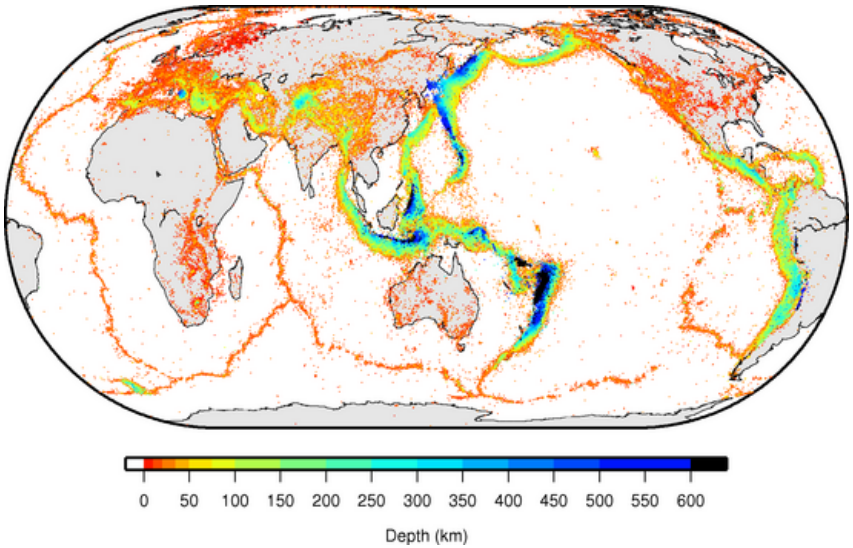
International Seismological Centre

IRIS

FDSN

Orfeus

Data-sharing collaboration

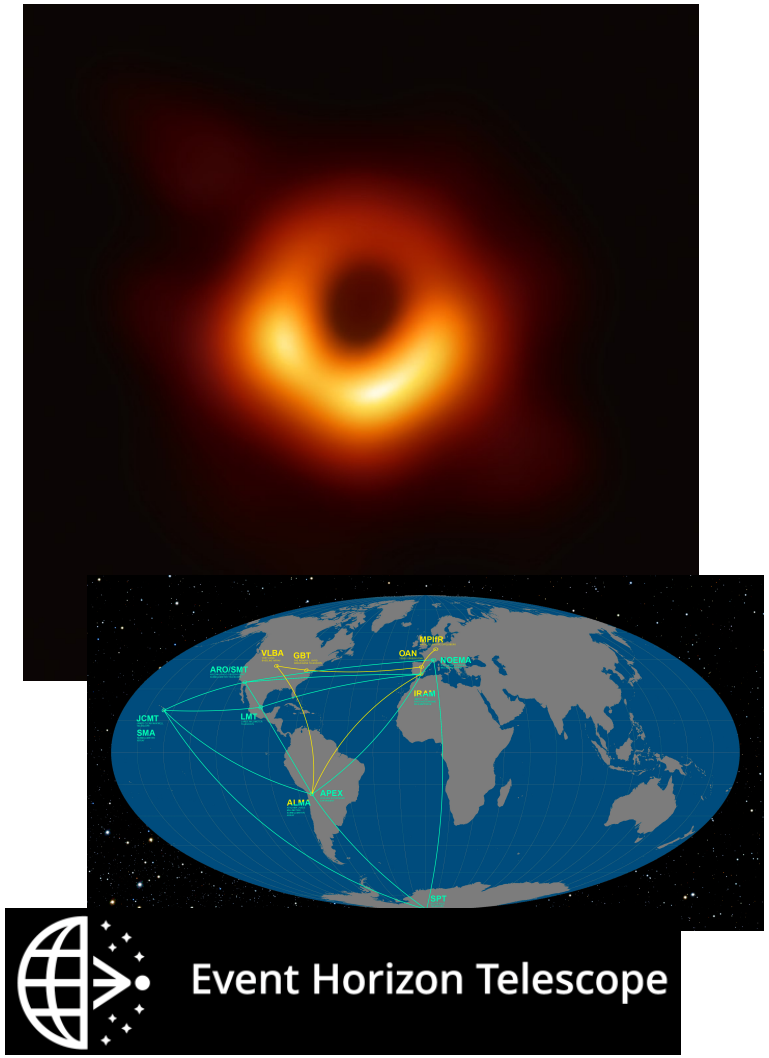


International Seismological Centre

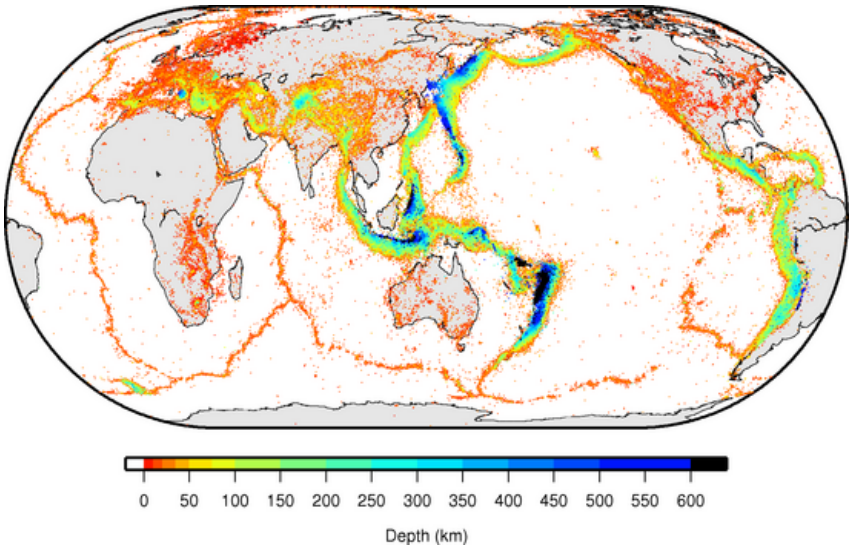
IRIS

FDSN

Orfeus



Data-sharing collaboration

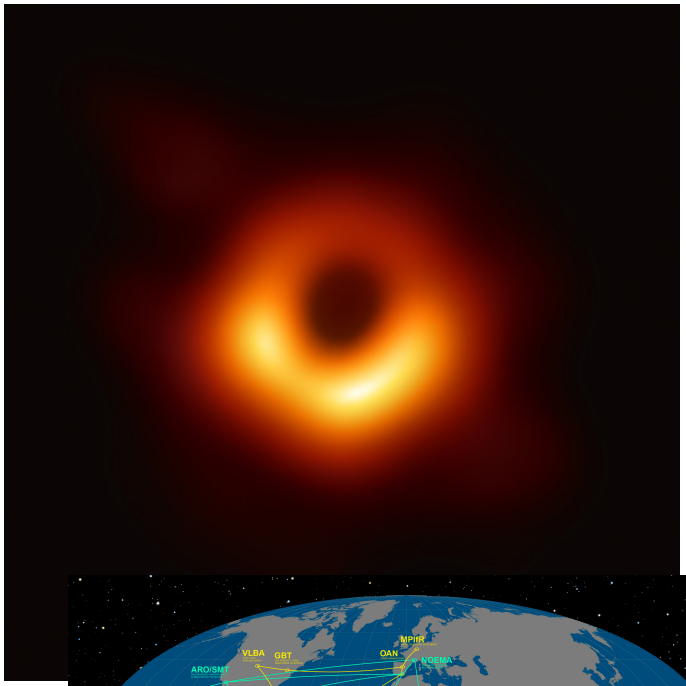


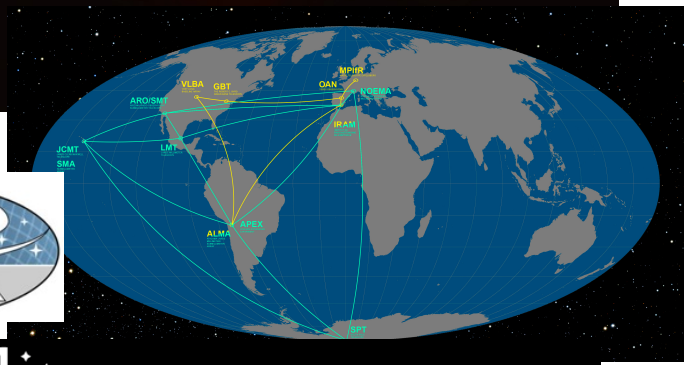
International Seismological Centre

IRIS

FDSN

Orfeus





IVOA

Event Horizon Telescope

Current practice

Data-sharing and more:

- Data
- Method
- Code
- (Re-)Execution

Current practice

Data-sharing and more:

- Data
- Method
- Code
- (Re-)Execution

Different organizational forms:

- Centralized
  - Single-institutional
  - Cross-institutional
- Federated

Current practice

Data-sharing and more:

- Data
- Method
- Code
- (Re-)Execution

Different organizational forms:

- Centralized
  - Single-institutional
  - Cross-institutional
- Federated

Observed problems

- Shredded
  - Hard to trace back
  - Hard to receive update
  - Hard to reuse
- "Random" data use
  - Unintended data leaks
  - Violating T&C from the data providers

Current solution... of *shredded*

- Workflow + standardized description language (e.g. CWL)
- Provenance record
- Research portal / scientific gateway

Current solution... of *"random" data use*

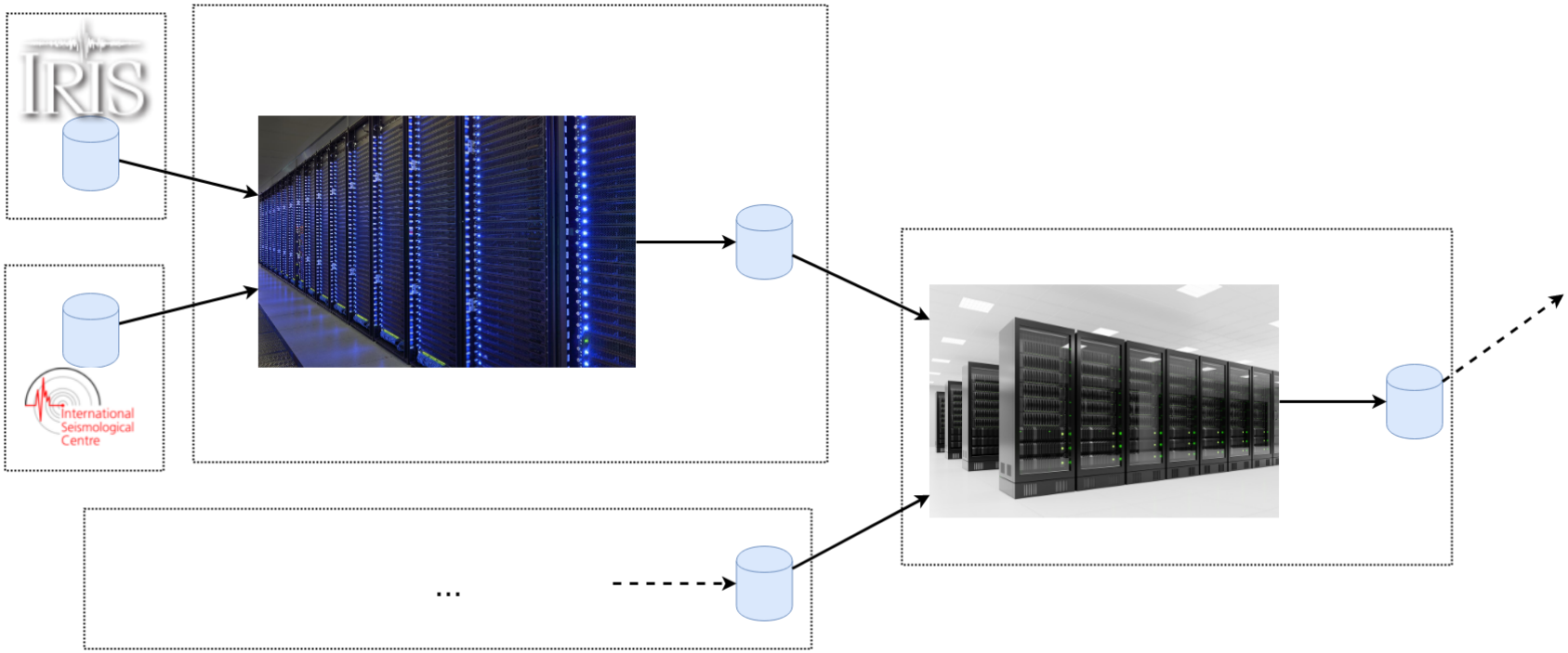
- Sensitive data
  - Restricted environment
  - Extra training
- Non-sensitive data
  - Very basic or no policy (open data)

Current ~~solution~~... of *"random" data use*

- Sensitive data
  - Restricted environment
  - Extra training
  - Long rigorous approval procedure
- Many stages & data in between middle ground
  - No systematic support available
- Non-sensitive data
  - Very basic or no policy (open data)
  - Hidden rules

$\Rightarrow$ Polarized, wasted effort and accidents

Re-think the big picture



Proposed strategy

- Encode data-use rules
- Explicit
  - for humans
  - for computers
- Precise
- Computer-aided compliance

Formal model for data policies (i.e. governance rules)

+

Supporting framework

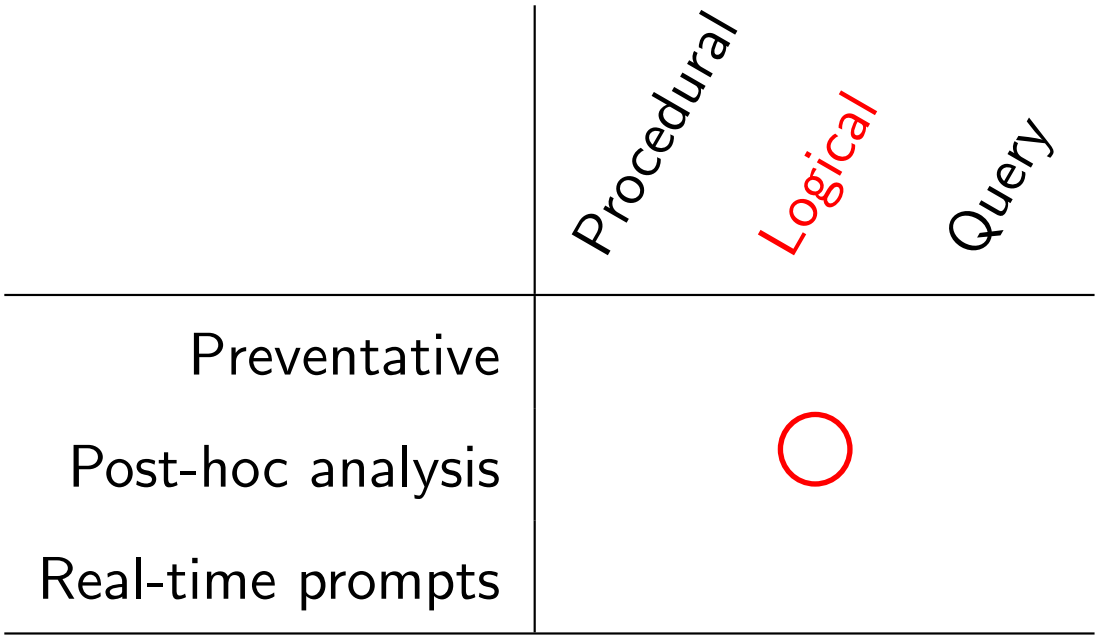
Language model requirements

- Human comprehensible
  - Easy to author
  - Easy to understand
- Computer interpretable
  - (Critical)
- Transformation needed?

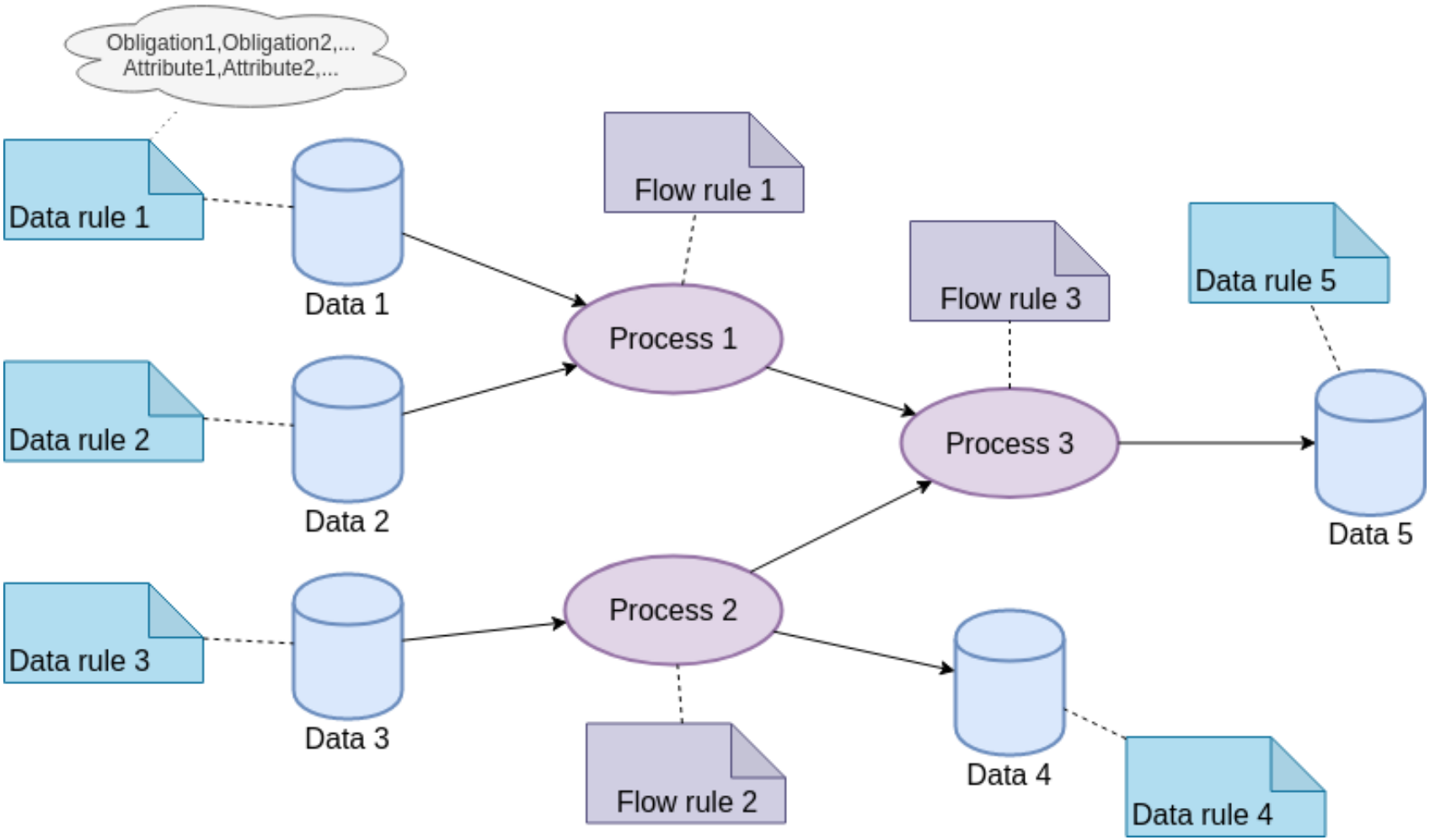
Solution space

	Procedural	Logical	Query
Preventative			
Post-hoc analysis			
Real-time prompts			

Solution space

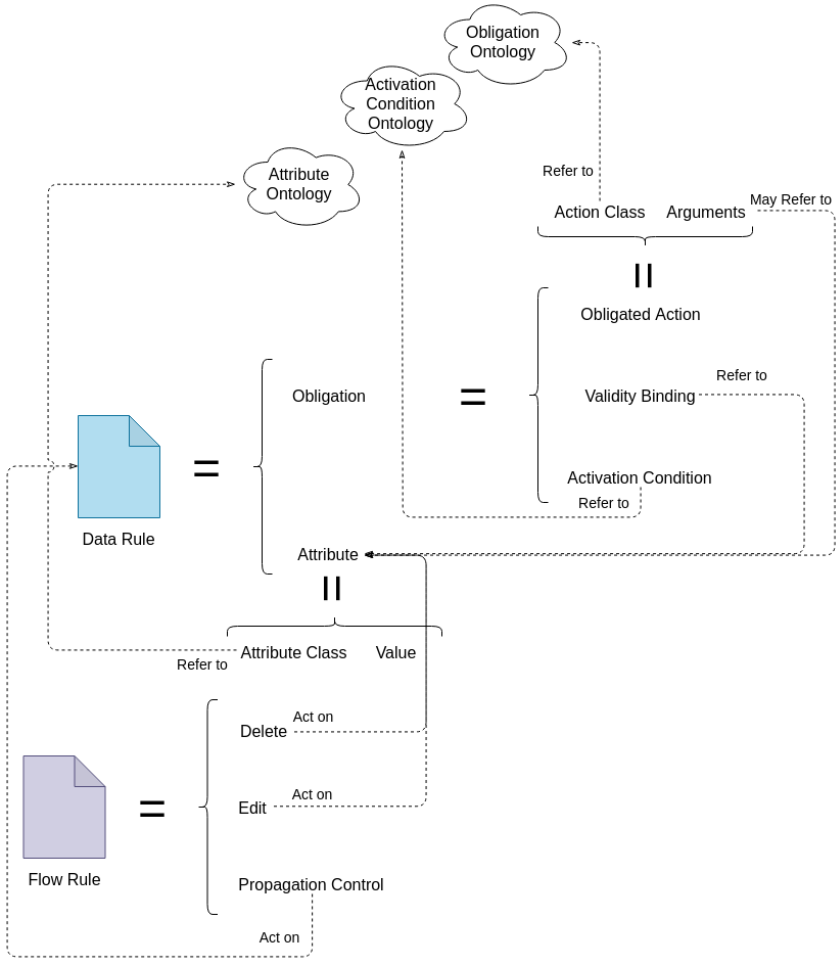


Let rules *flow* with data and *change* with processing



Use ontologies for interoperability

- Ontologies for different purposes
  - Obligation type
  - Attribute class
  - Activation condition
- Rule propagation control



Provenance as input

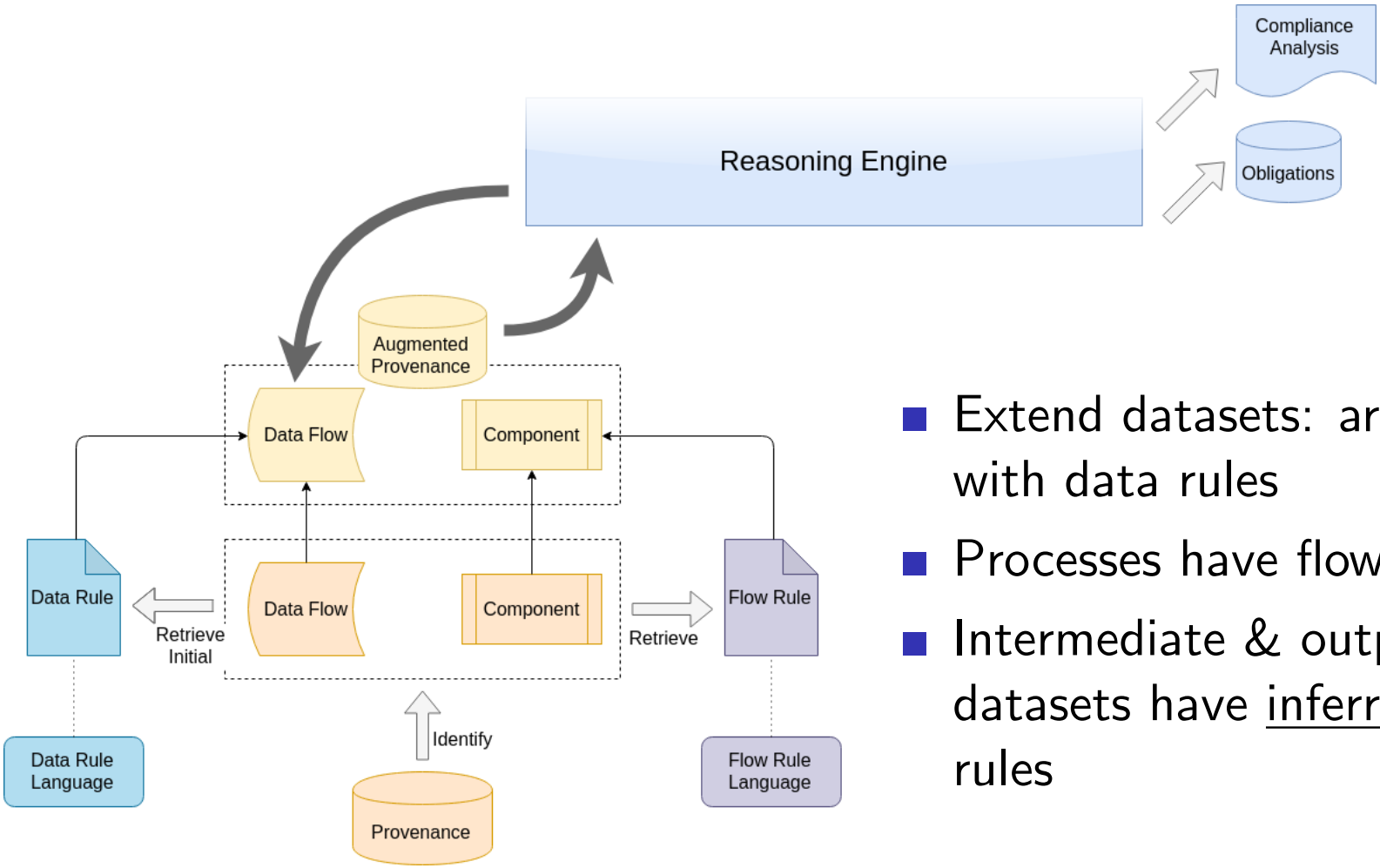
Why *provenance*?

- History
- Standardized
- Extensible
- Interoperable

Provenance provides

Integrated information flow from systems, software and tools

Working rule compliance checker



- Extend datasets: arrive with data rules
- Processes have flow rules
- Intermediate & output datasets have inferred rules

Example: simple yet complicated

Data providers often want data users to...

> Account data use: "Report the number of times you use our data."

Example: simple yet complicated

"Report the number of times you use our data."

Define *use*:

- *every time* as input to a process
- when as *initial input*
- when resulting in a *successful result*

Define *report time*:

- *whenever a new use occurs*
- *when finished*

Example encoding

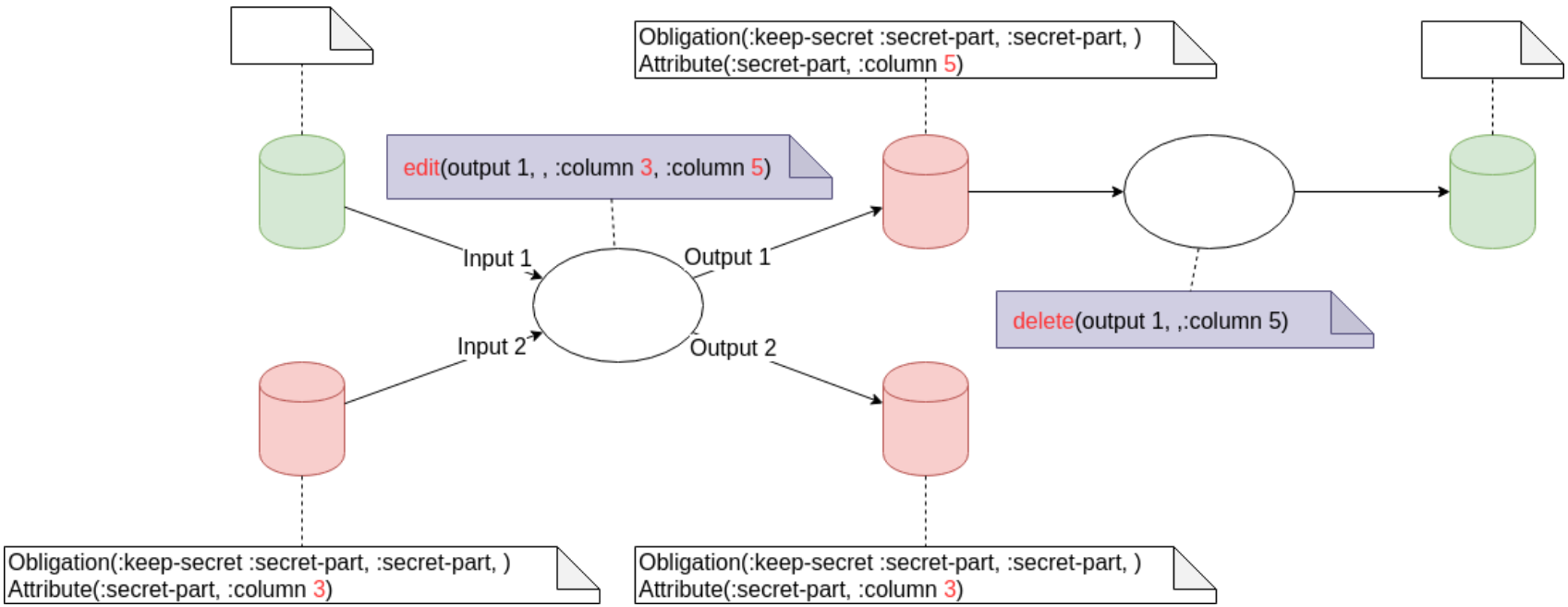
Report the number of times you use our data *every time* as input to a process

```
Obligation(:report :source, :source, :WhenAsInput)
Attribute(:source, "some-source")
```

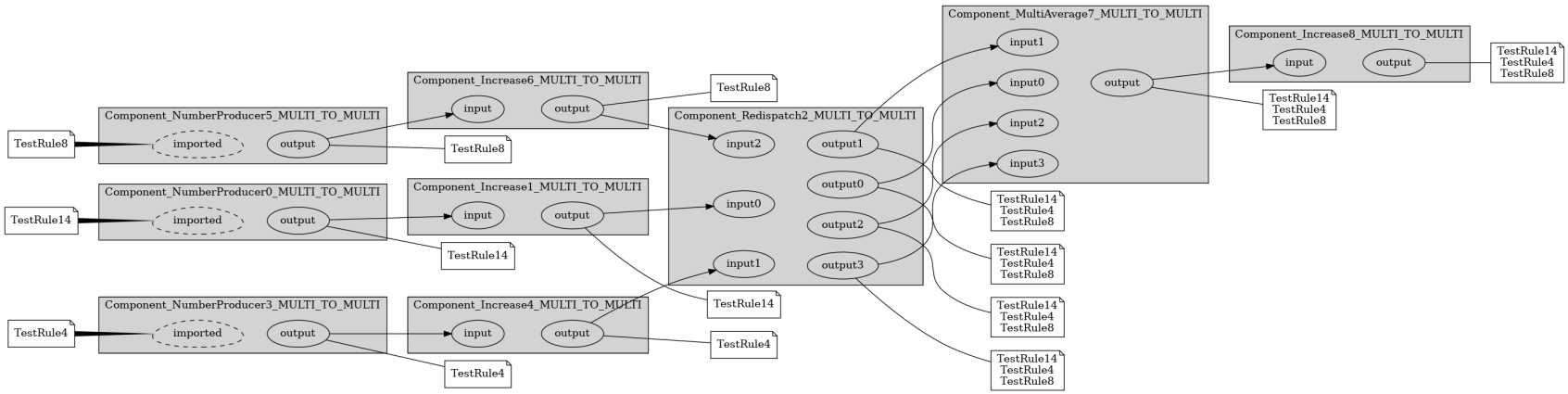
Report the number of times you use our data when as *initial input*

```
Obligation(:report :source, :source, :WhenImported)
Attribute(:source, "some-source")
```

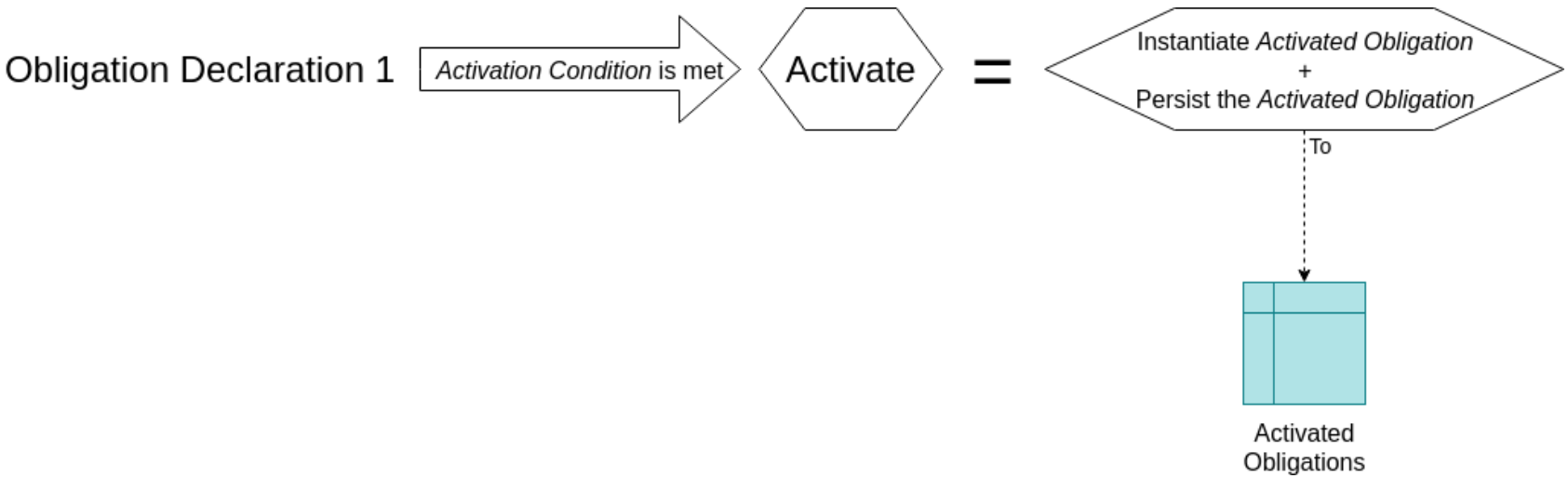
Flow Rule


Obligation(:keep-secret :secret-part, :secret-part,)
Attribute(:secret-part, :column 5)
edit(output 1, , :column 3, :column 5)
Input 1
Output 1
Input 2
Output 2
delete(output 1, ,:column 5)
Obligation(:keep-secret :secret-part, :secret-part,)
Attribute(:secret-part, :column 3)
Obligation(:keep-secret :secret-part, :secret-part,)
Attribute(:secret-part, :column 3)

Merge


Component_NumberProducer5_MULTI_TO_MULTI
imported
output
TestRule8
Component_NumberProducer0_MULTI_TO_MULTI
imported
output
TestRule14
Component_NumberProducer3_MULTI_TO_MULTI
imported
output
TestRule4
Component_Increase6_MULTI_TO_MULTI
input
output
TestRule8
Component_Increase1_MULTI_TO_MULTI
input
output
TestRule14
Component_Increase4_MULTI_TO_MULTI
input
output
TestRule4
Component_Redispatch2_MULTI_TO_MULTI
input2
input0
input1
output1
output0
output2
output3
TestRule8
TestRule14
TestRule4
Component_MultiAverage7_MULTI_TO_MULTI
input1
input0
input2
input3
output
TestRule14
TestRule4
TestRule8
Component_Increase8_MULTI_TO_MULTI
input
output
TestRule14
TestRule4
TestRule8

Activation


Obligation Declaration 1
Activation Condition is met
Activate
=
Instantiate Activated Obligation
+
Persist the Activated Obligation
To
Activated
Obligations

Future work

- Improve logical foundation
- More rule forms
  - wider range of use cases
- Supporting systems
  - operational integration & tools

Question time & Acknowledgement

Questions?

Special thanks to Christian Pagé, Wim Som de Cerff and Luca Trani who replied to our initial survey, to Alessandro Spinuso who provided testing provenance database (and helped the survey) and to everyone who provided help.

Activation - extra