The data, they are a-changin'
(ReComp: Your Data Will Not Stay Smart Forever)

Panta Rhei
(Heraclitus, through Plato)



(*)

Paolo Missier, Jacek Cala, Eldarina Wijaya

School of Computing Science,

Newcastle University

{firstname.lastname}@ncl.ac.uk

TAPP'16

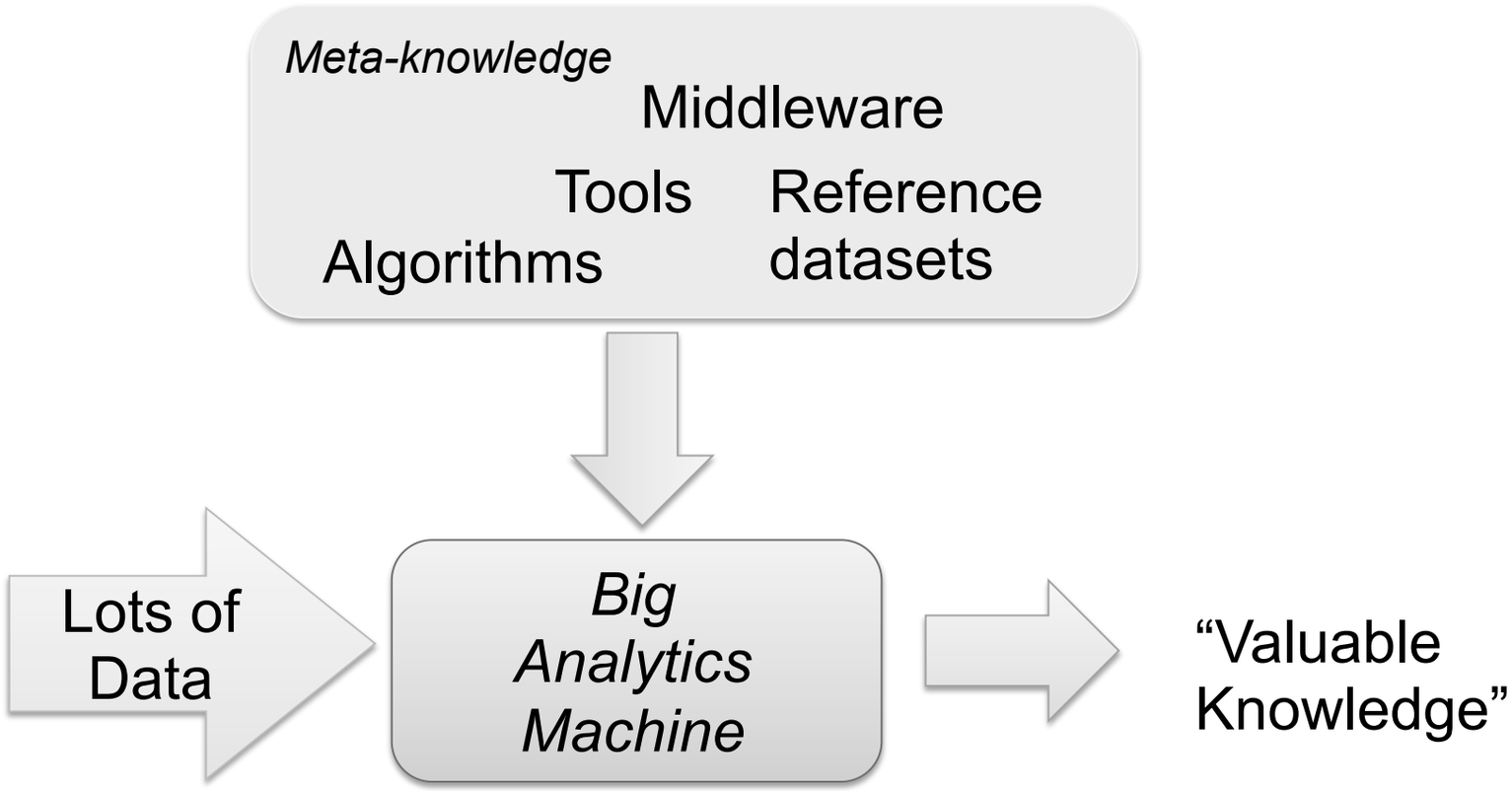
McLean, VA, USA

June, 2016

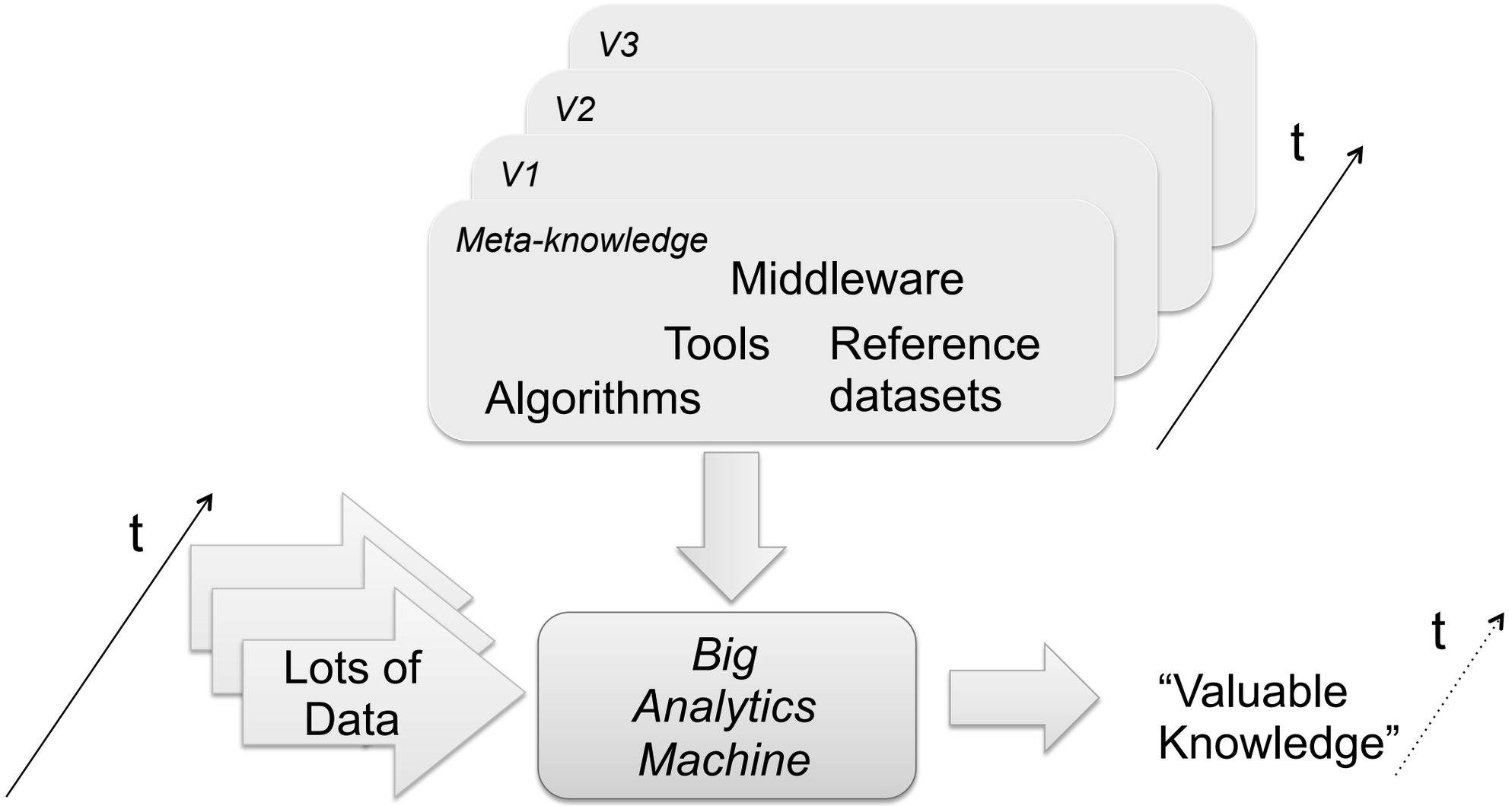
Engineering and Physical Sciences Research Council

(*) Painting by Johannes Moreelse

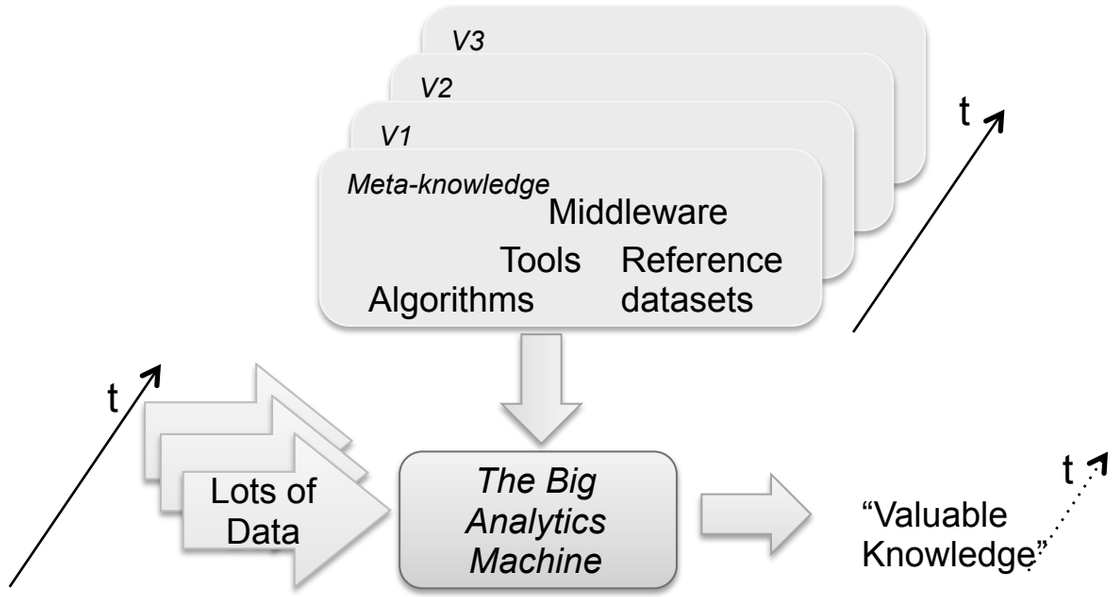
Data to Knowledge



The missing element: time



Your Data Will Not Stay Smart Forever



Observe change

- In input data
- In meta-knowledge

Assess and measure

- knowledge decay

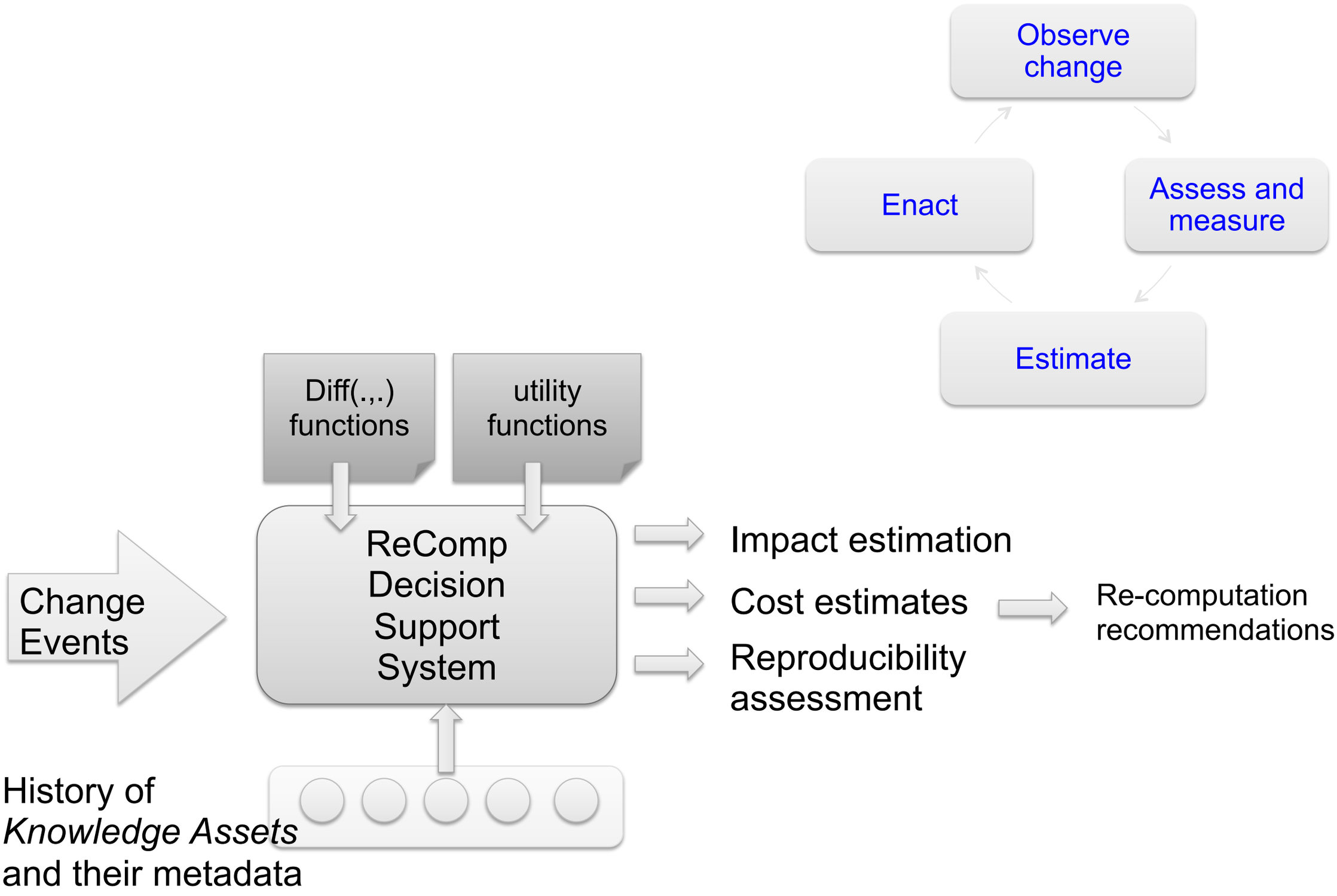
Estimate

- Cost and benefits of refresh

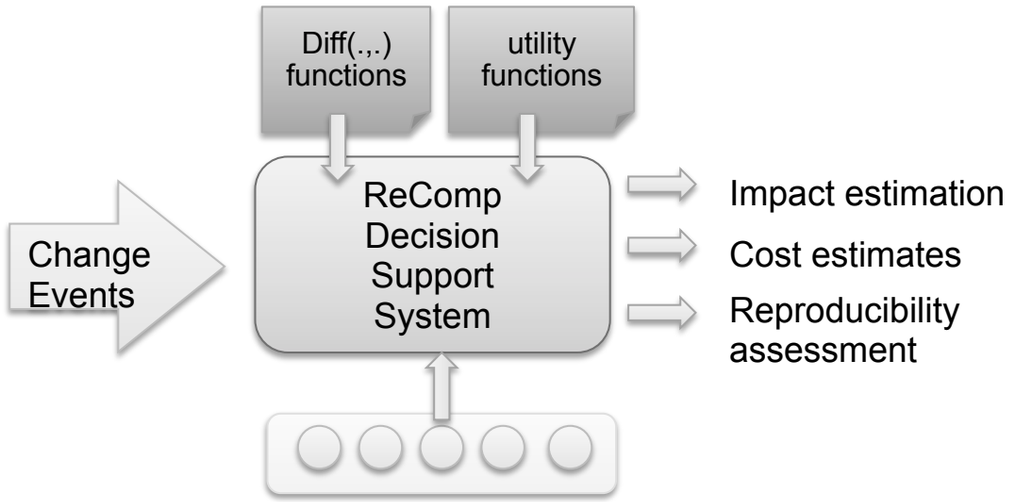
Enact

- Reproduce (analytics) processes

The *ReComp* decision support system



ReComp concerns



1. Observability (transparency)

How much can we observe?

- Structure
- Data flow

Provenance

2. Change detection: inputs, outputs, external resources

Can we quantify the extent of changes? → *diff()* functions

3. Impact assessment

Can we quantify knowledge decay?

4. Control: reaction to changes

How much re-computation control do we have on the system?

- Scope: Which instances?
- Frequency: how often?
- Re-run Extent: how much?

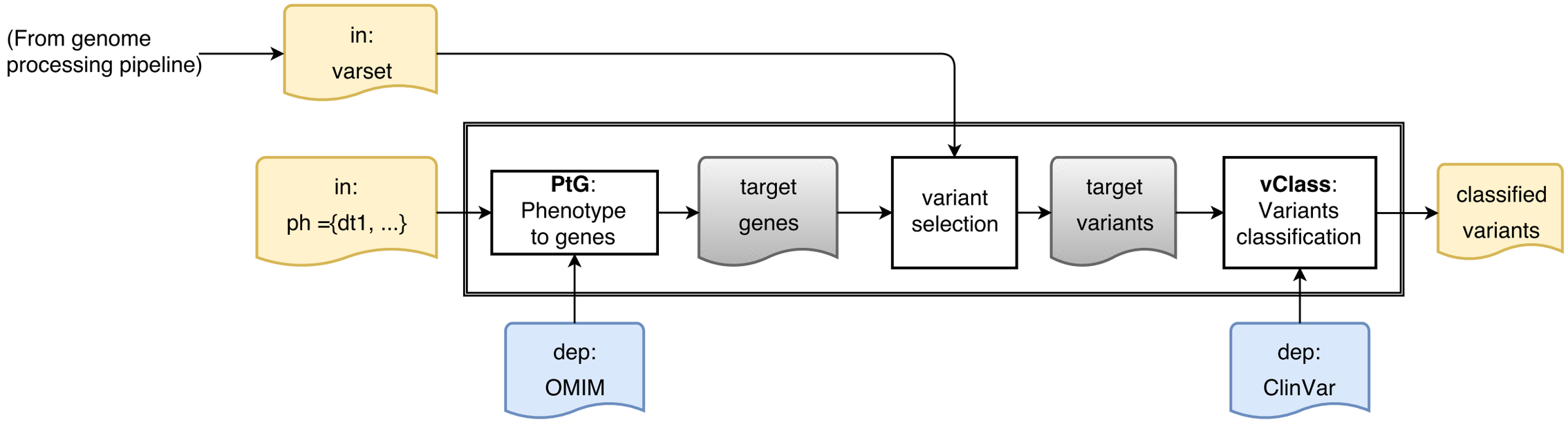
Reproducibility

- *Virtualisation*
- *Smart re-run*

This talk: White box ReComp -- initial experiments

	White box	Black box
Structure (static view)	Dataflow - eScience Central, Taverna, VisTrails… Scripting: - R, Matlab, Python...	- Packaged components - Third party services
Data dependencies (runtime view)	Provenance recording: • Inputs, • Reference datasets, • Component versions, • Outputs	• Input • Outputs • No data dependencies • No details on individual components
Cost	• Detailed resource monitoring • Cloud → £££	• Wall clock time • Service pricing • Setup time (eg model learning)

Example: genomics / variant interpretation



SVI is a classifier of likely variant deleteriousness:

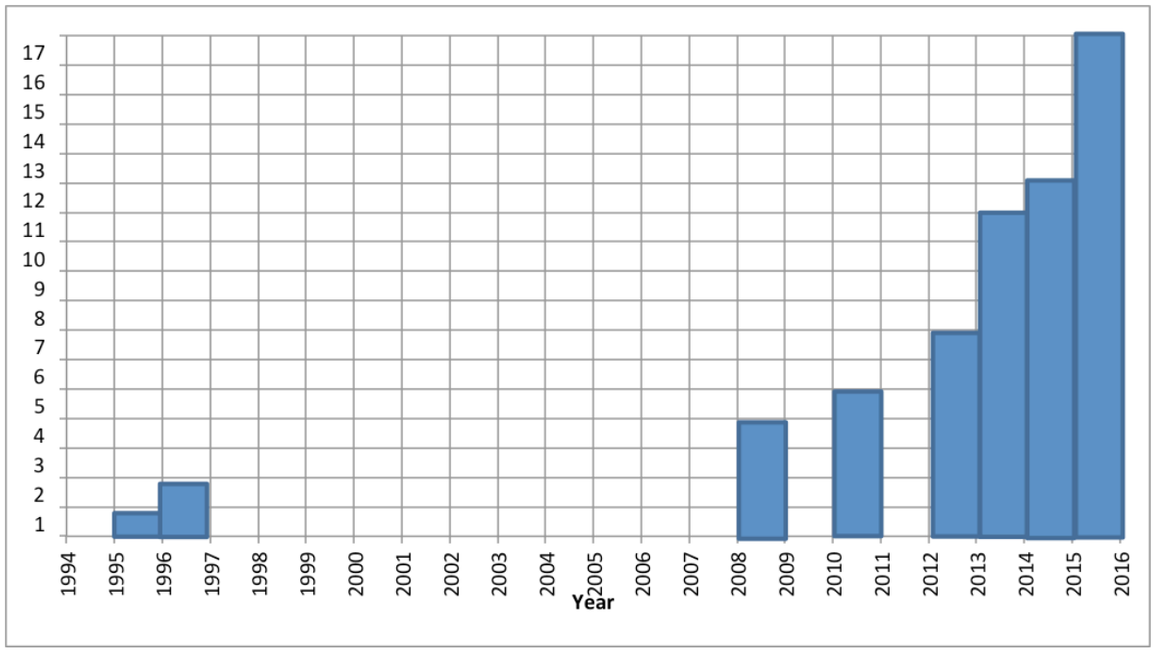
$$y = \{(v, class) | v \in varset, class \in \{red, amber, green\}\}$$

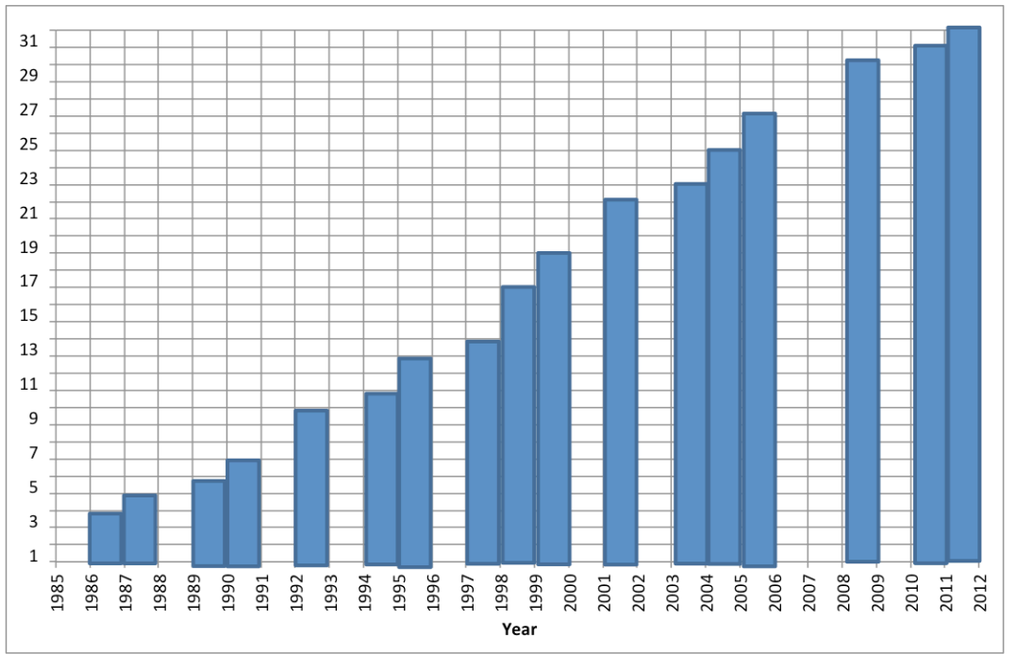
- Definitely deleterious
- Uncertain diagnosis
- Definitely benign

OMIM and ClinVar changes

Sources of changes:

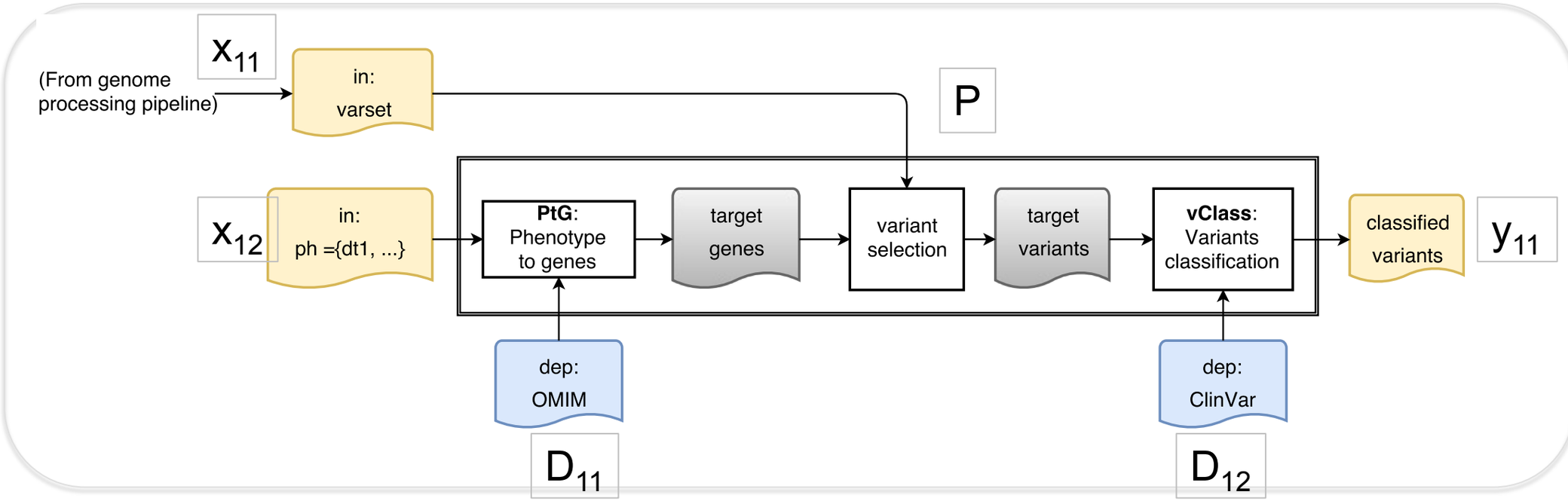
- Patient variants → improved sequencing / variant calling
- ClinVar, OMIM evolve rapidly
- New reference data sources





CLINVAR / OMIM relevant changes over time for a patient cohort
(Newcastle Institute of Genetics Medicine)

For each run *i*:



Observables:

Inputs X = {$x_{i1,}$ x_{12}, …}

Outputs y = {y_{i1}, y_{i2},…}

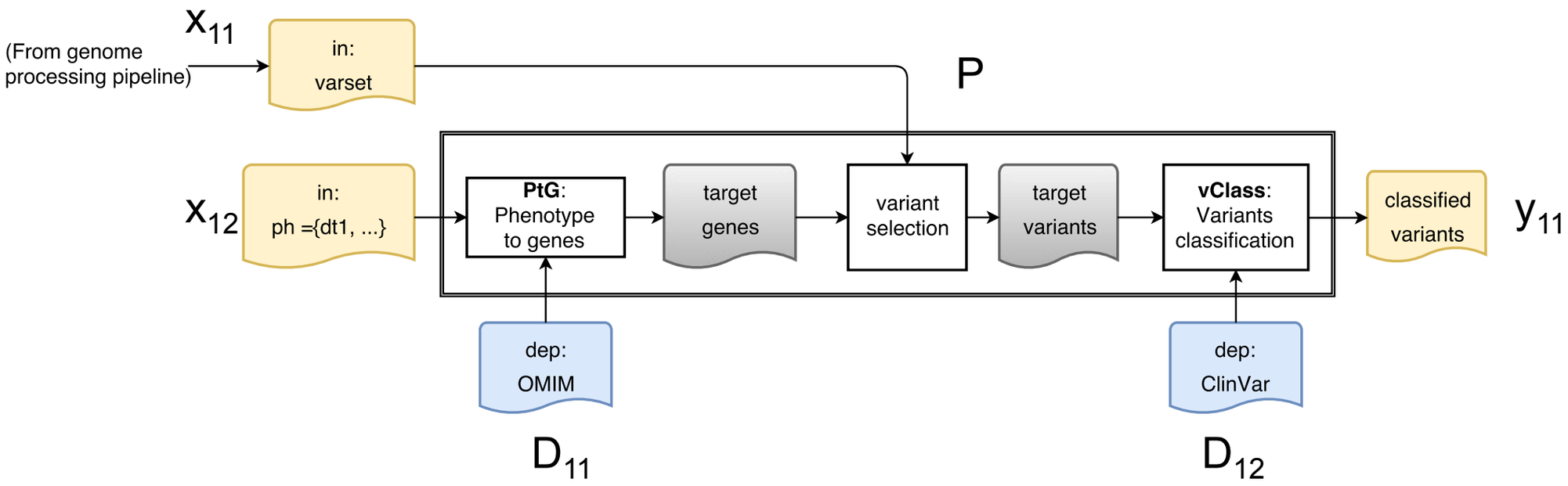
Dependencies D_{11}, D_{12}, ...

Variable-granularity provenance prov(y)

Granular Cost(y) → single-block level

Granular Process structure P → workflow graph

White-box provenance

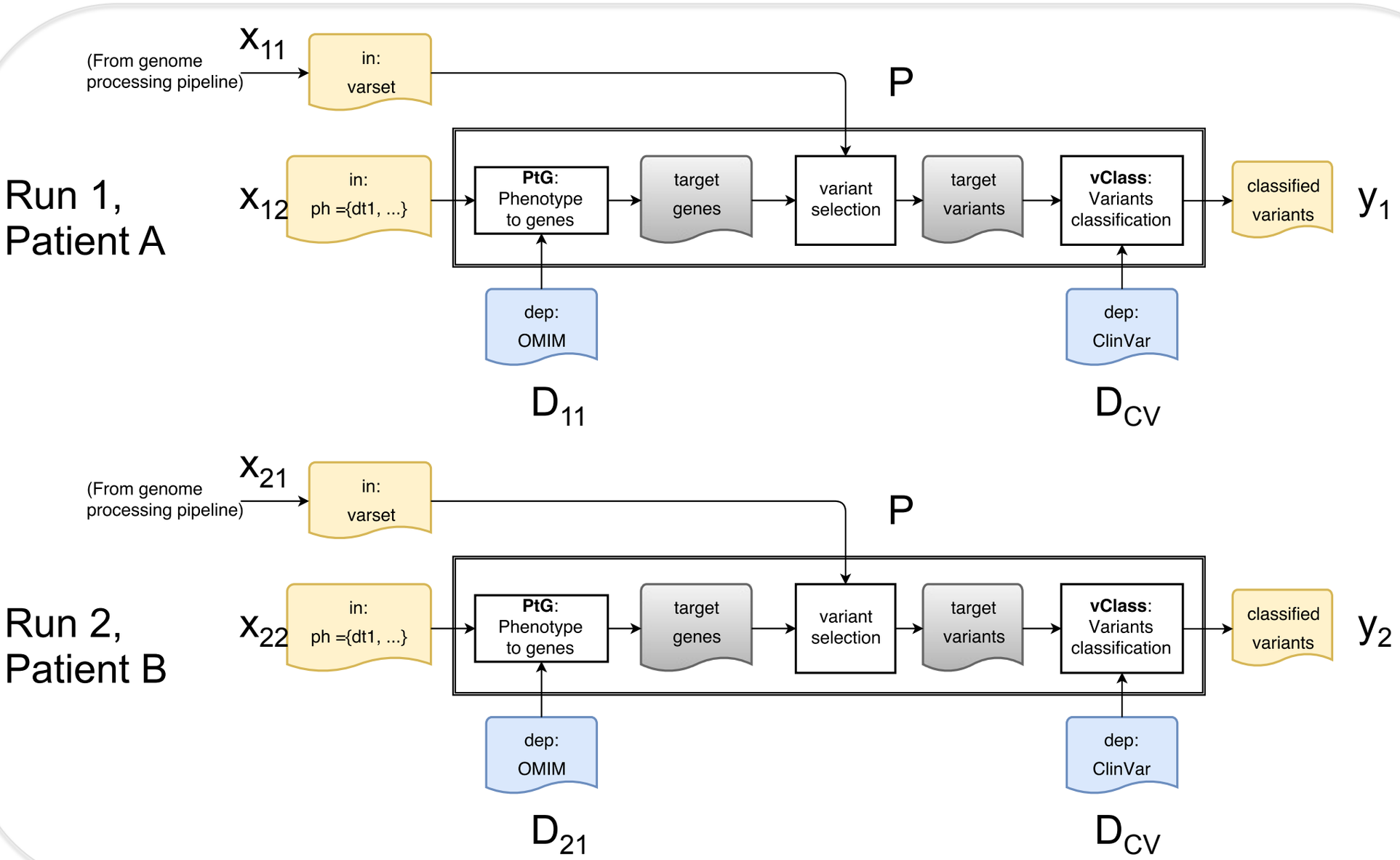


Granular: $used(P_j, d_{ij}, [prov : role = dep]), d_{i,j} \in D_i \in \mathbf{D}$

Coarse:

$$\begin{aligned}
&\texttt{entity}(\texttt{om}, [\texttt{prov:type} = \text{'OMIM'}, version = \texttt{'v'}])\\
&\texttt{entity}(\texttt{ph}, [\texttt{prov:type} = \text{'prov:collection'}])\\
&\texttt{entity}(\texttt{cv}, [\texttt{prov:type} = \text{'CV'}, version = \texttt{'v'}])\\
&\texttt{entity}(\texttt{vars}, [\texttt{prov:type} = \text{'prov:collection'}])\\
&\texttt{used}(\texttt{PtG}, \texttt{om}, [\texttt{prov:role} = \texttt{'dep'}])\\
&\texttt{used}(\texttt{PtG}, \texttt{ph}, [\texttt{prov:role} = \texttt{'input'}])\\
&\texttt{used}(\texttt{vClass}, \texttt{cv}, [\texttt{prov:role} = \texttt{'dep'}])\\
&\texttt{used}(\texttt{vClass}, \texttt{vars}, [\texttt{prov:role} = \texttt{'input'}])
\end{aligned}$$

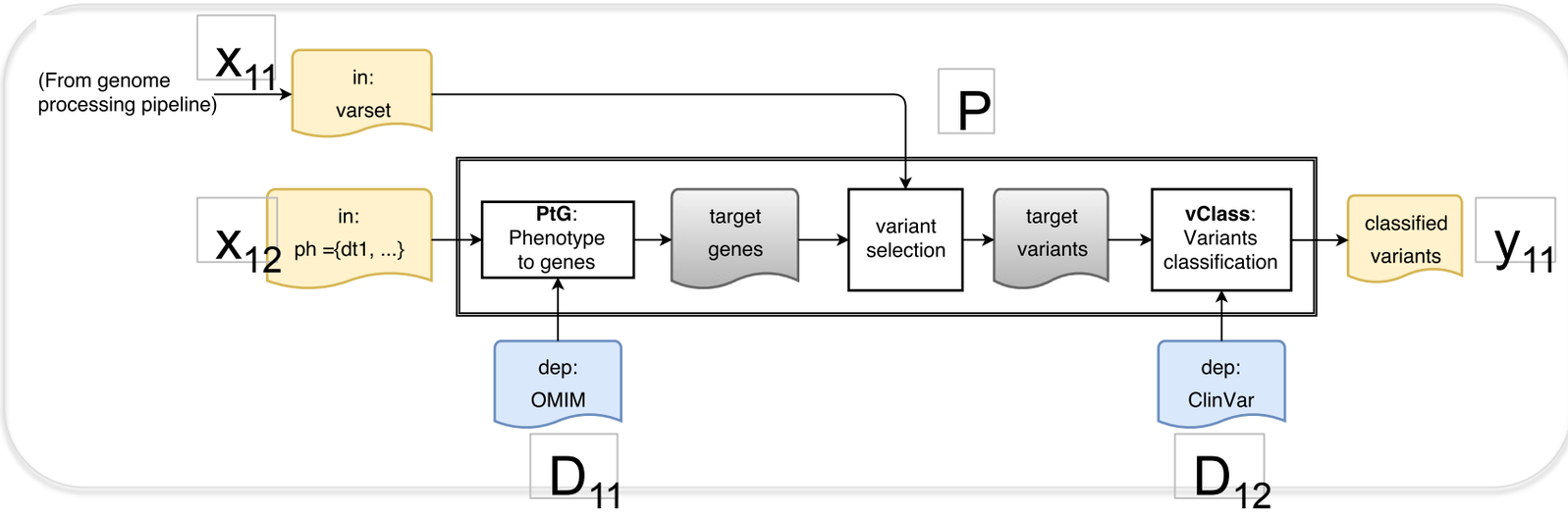
A history of runs



History database:

$$H = \{h(\mathbf{y}, v) = \langle P^v, \mathbf{D}^v, \mathbf{x}^v, prov(\mathbf{y}^v), cost(\mathbf{y}^v)\rangle\}$$

ReComp questions



- Scope: Which instances?

Which patients within the cohort are going to be affected by change in input/reference data?

- Re-run Extent: how much?

Where in each process instance is the reference data used?

- Impact: why bother?

For each patient in scope, how likely is that any patient's diagnosis will change?

- Frequency: how often?

How often are updates available for the resources we depend on?

1. History DB $H = \{h(\mathbf{y}, v) = \langle P^v, \mathbf{D}^v, \mathbf{x}^v, prov(\mathbf{y}^v), cost(\mathbf{y}^v)\rangle\}$

2. Measurables changes:

Input diff: $diff_{in}(x_i^v, x_i^{v'})$ → one patient at a time

Output diff: $diff_{out}(y_i^v, y_i^{v'})$ → has the change had any impact?

Dependencies $diff_d(D_i^v, D_i^{v'})$ → affects entire cohort → scoping

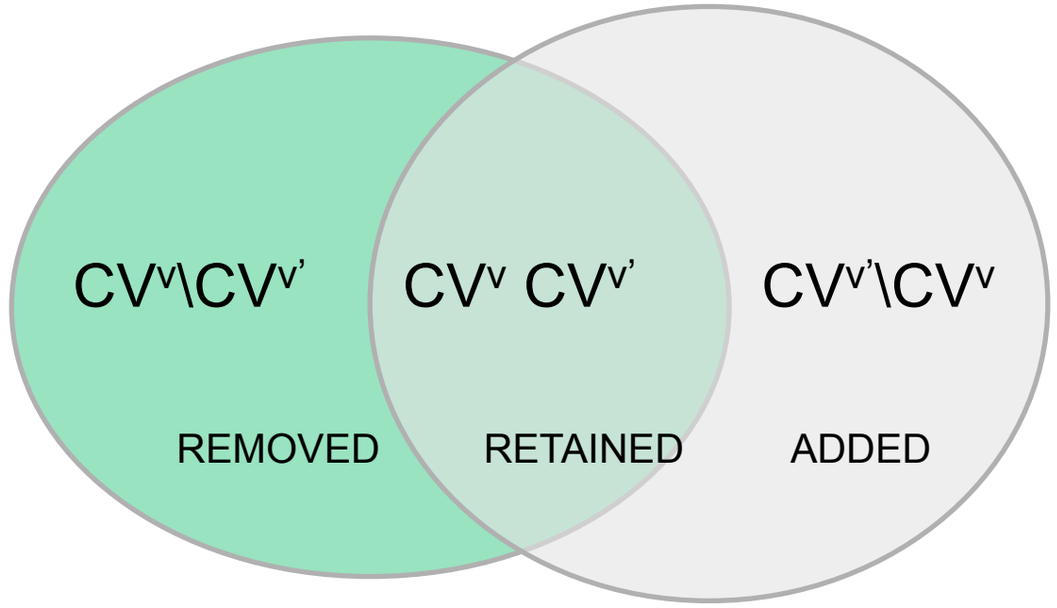
Example:

$$diff_{OM}(OM^v, OM^{v'}) = \{t \in DT | genes(t, OM^v) \neq genes(t, OM^{v'})\}$$

$$diff_{CV}(CV^v, CV^{v'}) =$$
$$\{var \in V | varstatus(var, CV^v) \neq varstatus(var, CV^{v'})\}$$
$$\cup\, CV^{v'} \setminus CV^v \cup CV^v \setminus CV^{v'}$$

The ClinVar diff function – two steps

1. Generic set-theoretic operations:



2. A ClinVar SVI specific filter:

REMOVED

CV^{REM}

$\{ var \in V | varstatus(var, CV^{v}) \} \in \{ \text{'pathogenic'}, \text{'benign'} \}$

RETAINED

CV^{CHG}

$\{ var \in V | varstatus(var, CV^{v}) \neq varstatus(var, CV^{v'}) \wedge (varstatus(var, CV^{v'}) \in \{ \text{'pathogenic'}, \text{'benign'} \} \vee varstatus(var, CV^{v}) \in \{ \text{'pathogenic'}, \text{'benign'} \}) \}$

ADDED

CV^{ADD}

$\{ var \in V | varstatus(var, CV^{v'}) \in \{ \text{'pathogenic'}, \text{'benign'} \} \}$

Up to 99%selectivity

Querying provenance to determine Scope and Re-run Extent

Case 1: Granular provenance

Given observed changes in resources $diff_d(D_i^v, D_i^{v'})$

1. Scoping: For each $d_{ij} \in diff_d(D_i^v, D_i^{v'})$

History instance:

$$h(\mathbf{y}, v) = \langle P^v, \mathbf{D}^v, \mathbf{x}^v, prov(\mathbf{y}^v), cost(\mathbf{y}^v) \rangle \in H$$

is in the scope $S \subseteq H$ if

$$\texttt{used}(P_j, d_{ij}, [\texttt{prov:role} = \texttt{'dep'}]) \in prov(\mathbf{y}^v)$$

P_j is added to $P_{scope}(y)$

2. Re-run Extent:

1. Find a partial order on $P_{scope}(y)$
2. Re-run starts from each of the earliest P_j such that their output is available as persistent intermediate result

see for instance Smart Run Manager [1]

[1] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows*, 2005, Wiley.

Querying provenance to determine Scope and Re-run Extent

Case 2: Coarse-grained provenance

Scoping: Any instance that depends on any D_{ij} is in scope:

For each $d_{ij} \in diff_d(D_i^v, D_i^{v'})$

$P_{scope} = \{P_j\}$, where:

$$\texttt{used}(P_j, D_i, [\texttt{prov:role} = \texttt{'dep'}]), \quad d_{ij} \in D_i$$

This is trivial for a homogenenous run population, but
H may contain run history for many different workflows!

Re-run Extent:
The mechanism from the fine-grained case still works

Assessing impact and cost

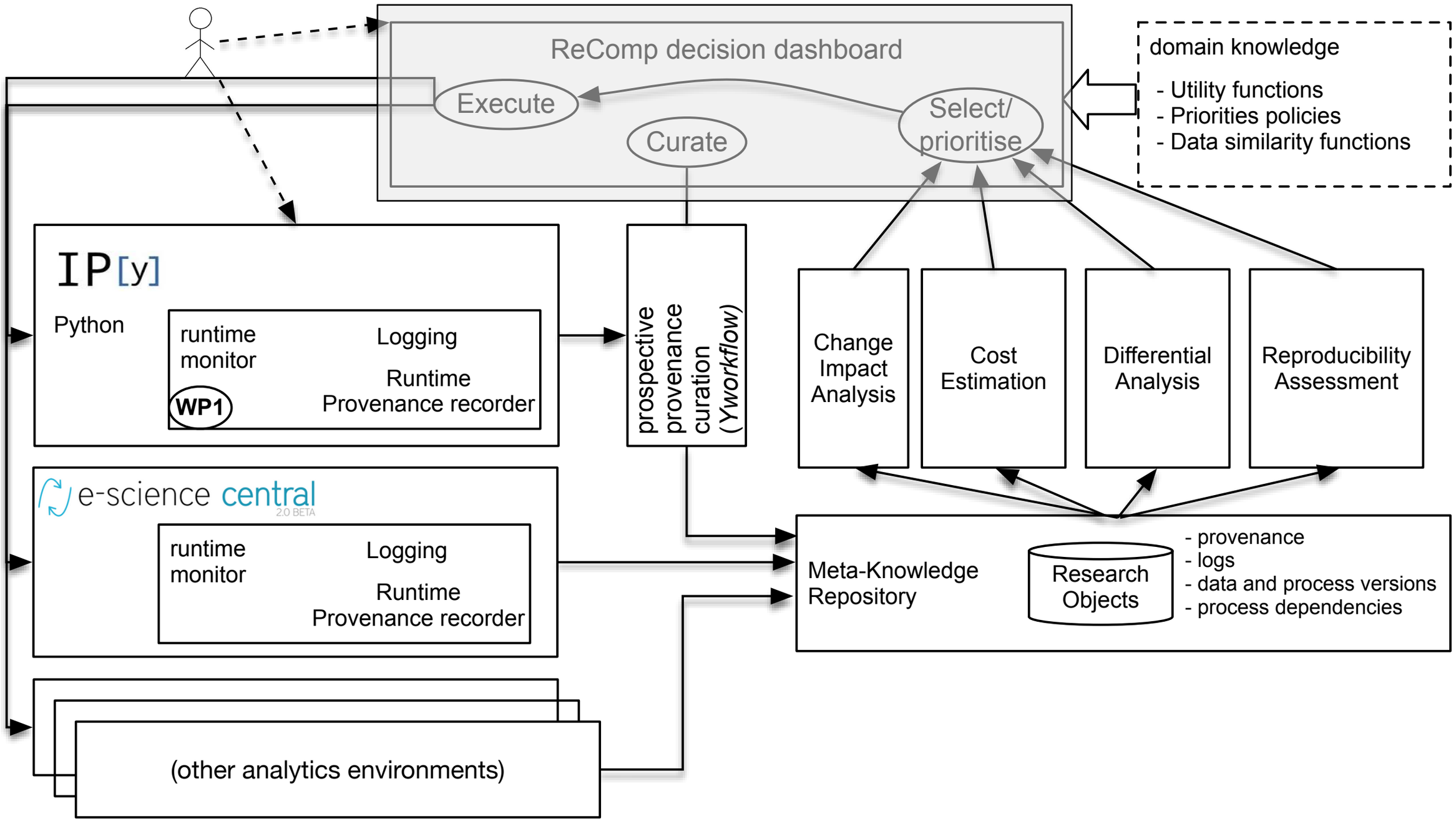
Approach: small-scale re-comp over the population in scope

1. Sample instances *S'* ⊆ *S* from the population in scope S
2. Perform partial re-run on each instance $h(\mathbf{y_i}, v) \in S'$, generating new outputs $\mathbf{y_i'}$
3. Compute $diff_{out}(y_i^v, y_i^{v'})$
4. Assess impact (user-defined) and *cost(y')*
5. Estimate cost difference *diff(cost(y), cost(y'))*

ReComp user dashboard and architecture

ReComp is a Decision Support System

Impact, cost assessment → ReComp user dashboard



Current status and Challenges

Implementation in progress

Small scale experiments on scoping / partial re-run

- Test cohort of about 50 (real) patients
- Short workflows runs (about 15 mins), observable cost savings
- (preliminary results)

Main challenge: deliver a generic and reusable DSS

From eScience Central → **To** generic dataflow, scripting (Python)

From

eSc prov traces → PROV-compliant but idiosincratic patterns

Python → noWorkflow traces

To: Canonical PROV patterns + queries + H DB implementation

ReComp: http://recomp.org.uk/

References

[1] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows*, 2005, Wiley.

[2] Ikeda, Robert, Semih Salihoglu, and Jennifer Widom. Provenance-Based Refresh in Data-Oriented Workflows. In *Procs CIKM*, 2011

[3] R. Ikeda and J. Widom. Panda: A system for provenance and data. *Procs TaPP10*, 33:1–8, 2010.

[4] D. Koop, E. Santos, B. Bauer, M. Troyer, J. Freire, and C. T. Silva. Bridging workflow and data provenance using strong links. In *Scientific and statistical database management*, pages 397–415. Springer, 2010. ISBN 3642138179.

[5] P. Missier, E. Wijaya, R. Kirby, and M. Keogh. SVI: a simple single-nucleotide Human Variant Interpretation tool for Clinical Use. In *Procs. 11th International conference on Data Integration in the Life Sciences*, Los Angeles, CA, 2015. Springer.