Fast Exploration of the QSAR Model Space with e-Science Central and Windows Azure

Simon Woodman

Jacek Cala

Hugo Hiden

Paul Watson

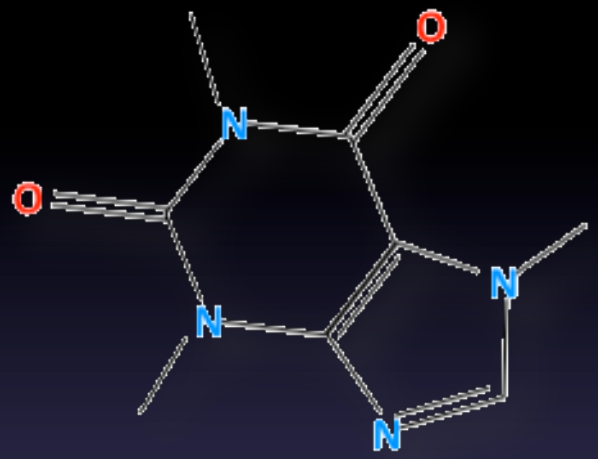
Newcastle University

e-science central

The Problem

What are the properties of this molecule?

Toxicity



Biological Activity

Solubility

Perform experiments

Time consuming

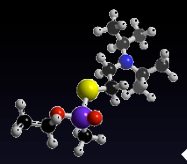
Expensive

Ethical constraints

e-science central

QSAR

Quantitative Structure Activity Relationship

Activity ≈ *f(*  *)*

More accurately, Activity related to a *quantifiable* structural attribute

Activity ≈ *f(logP, number of atoms, shape....)*

Currently > 3,000 recognised attributes

http://www.qsarworld.com/

e-science central

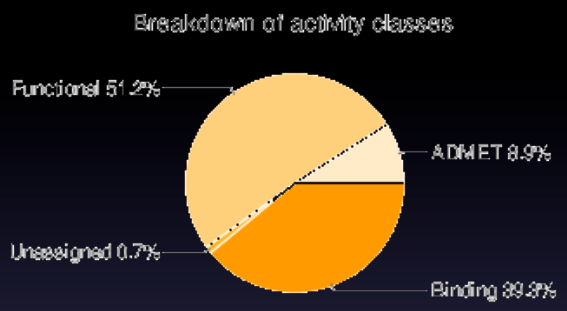
The alternative to Experiments

Predict likely properties based on similar molecules

CHEMBL Database: data on 622,824 compounds, collected from 33,956 publications

WOMBAT Database: data on 251,560 structures, for over 1,966 targets

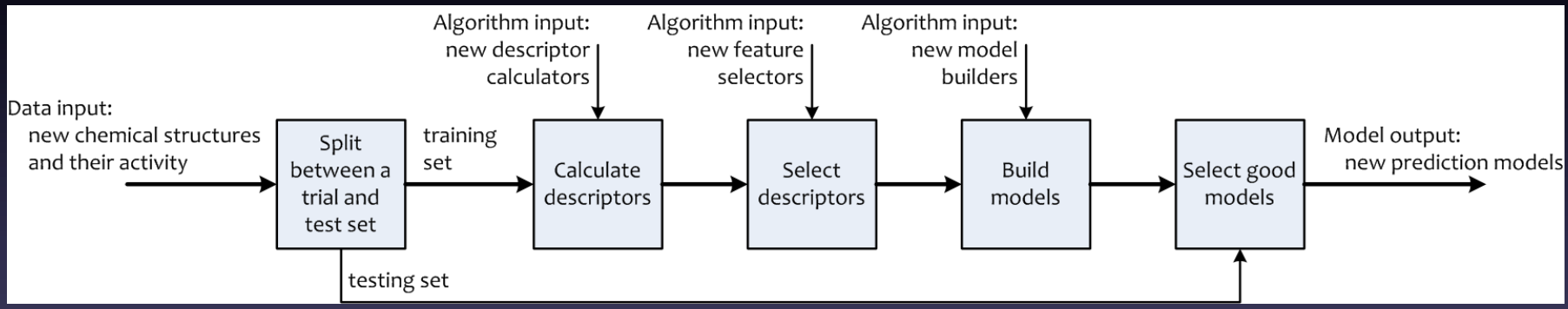
WOMBAT-PK Database: data on 1230 compounds, for over 13,000 clinical measurements



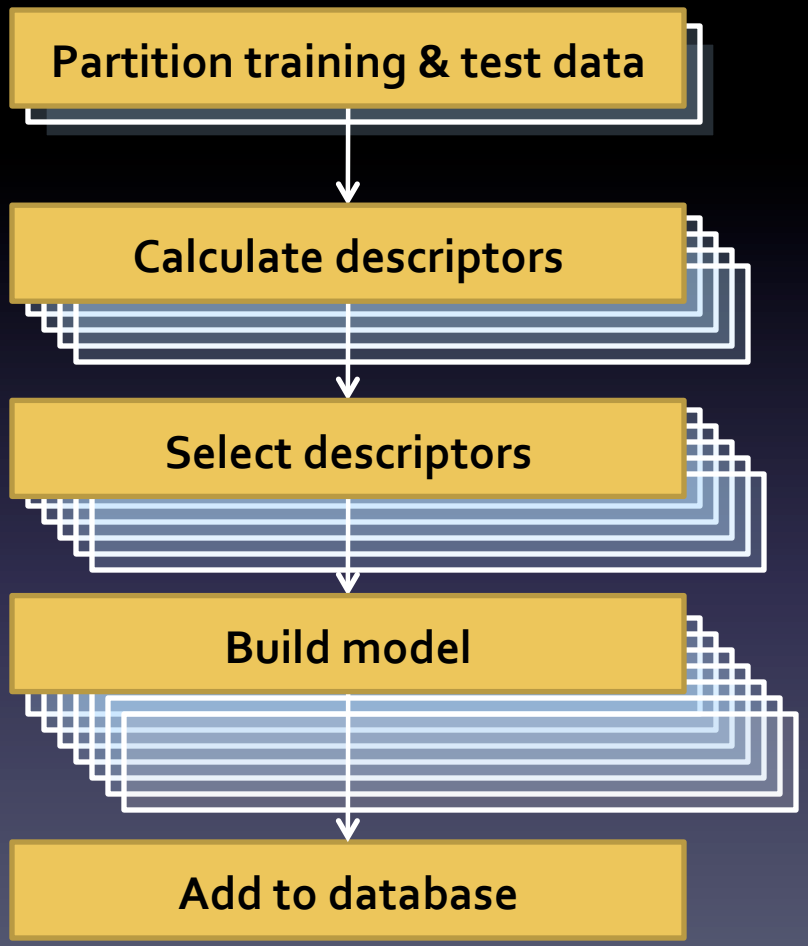
All these databases contain structure information and numerical activity data

e-science central

Method


Algorithm input:
new descriptor
calculators
Algorithm input:
new feature
selectors
Algorithm input:
new model
builders
Data input:
new chemical structures
and their activity
Split
between a
trial and
test set
training
set
Calculate
descriptors
Select
descriptors
Build
models
Select good
models
Model output:
new prediction models
testing set

Branching Workflows



Random split
80:20 split

Java CDK descriptors
C++ CDL descriptors

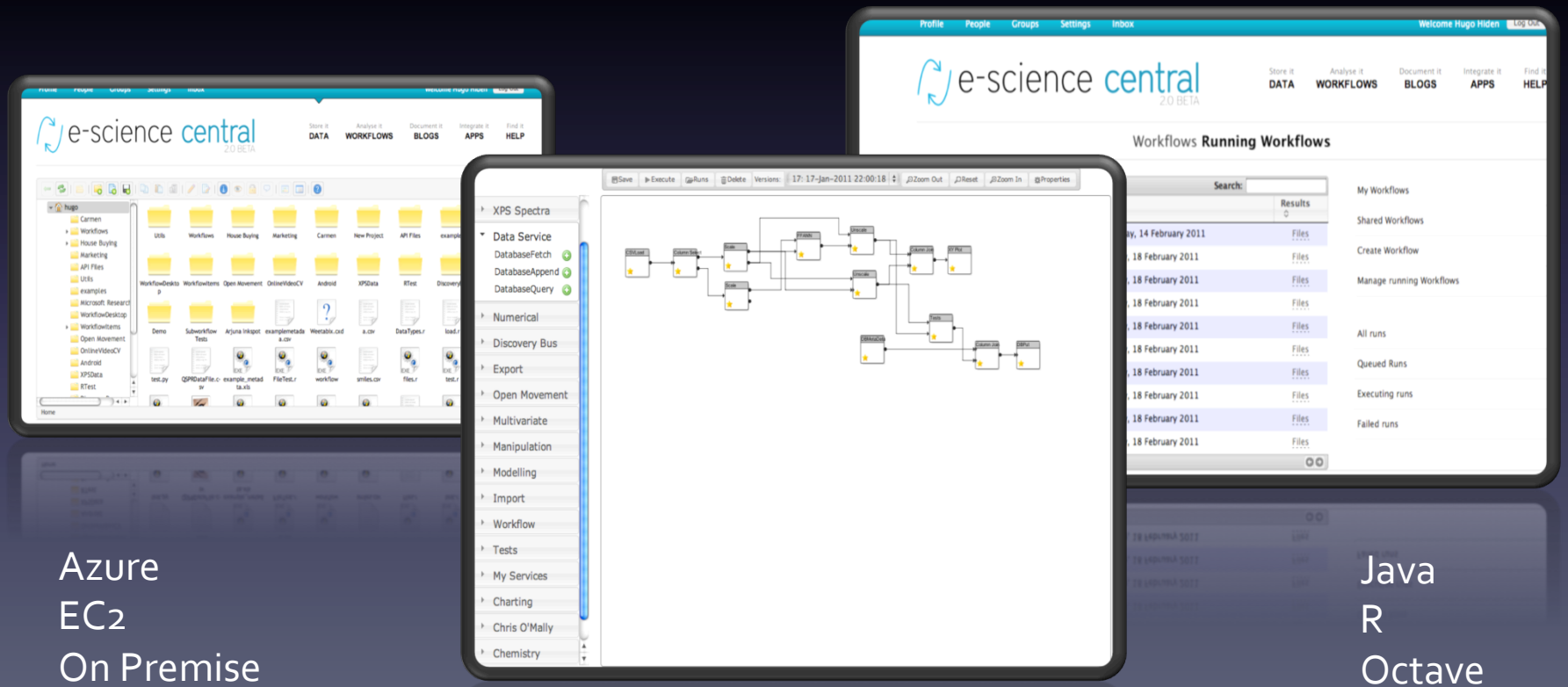
Correlation analysis
Genetic algorithms
Random selection

Linear regression
Neural Network
Partial Least Squares
Classification Trees

e-science central

e-Science Central

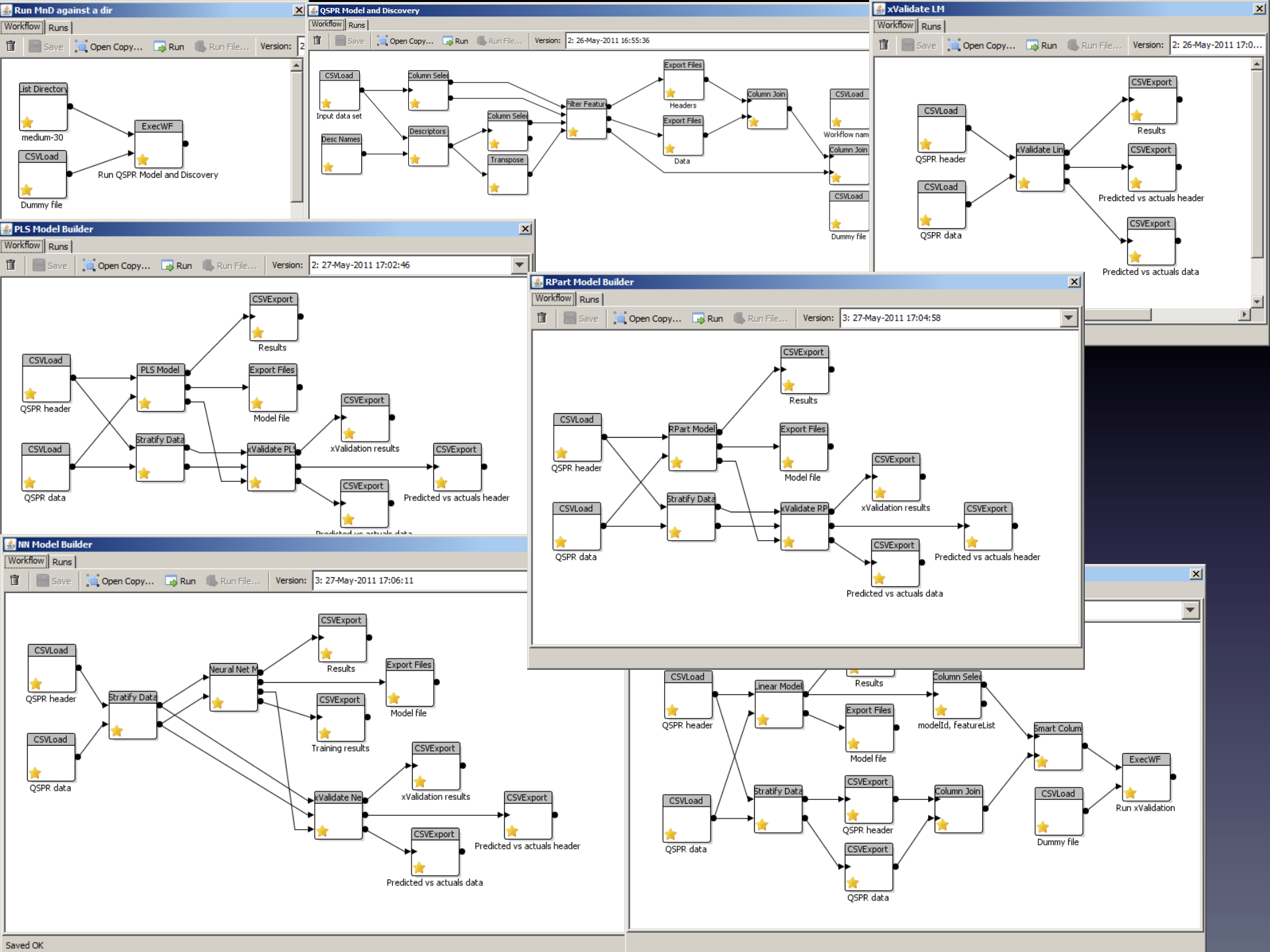
Platform for cloud based data analysis



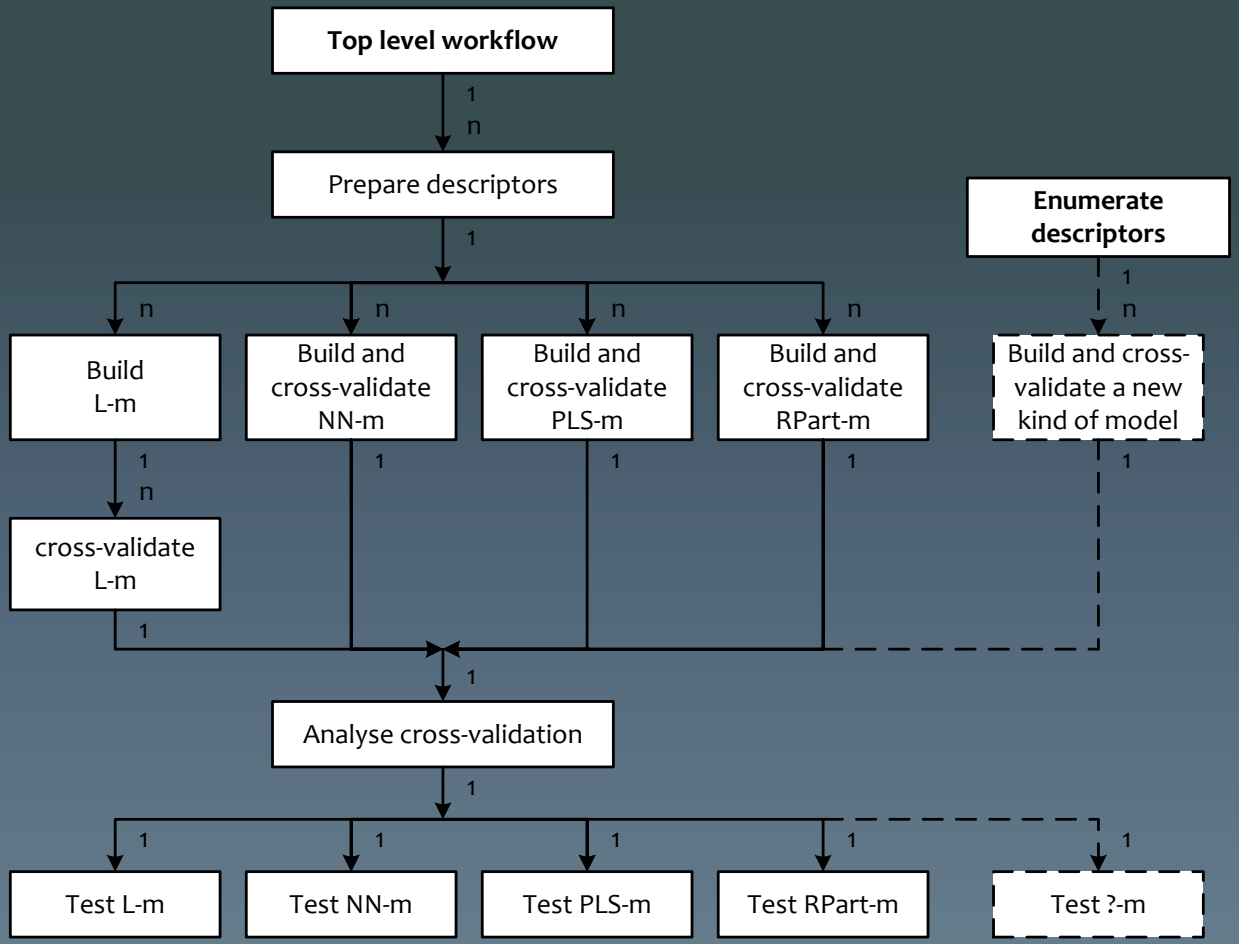
Azure
EC2
On Premise

e-science central

Java
R
Octave
Javascript

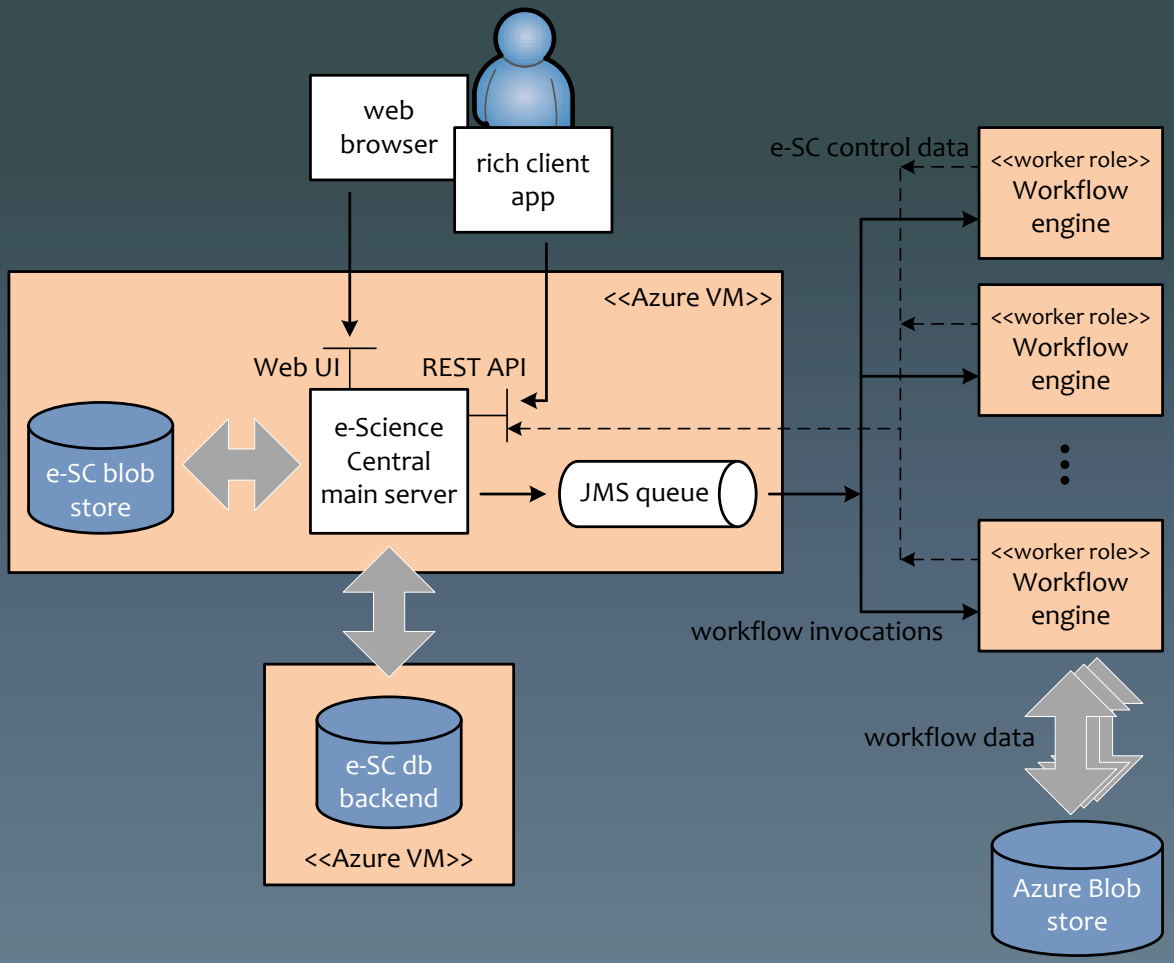

Run MnD against a dir
Workflow
Runs
Save
Open Copy...
Run
Run File...
Version:
List Directory
medium-30
ExecWF
CSVLoad
Dummy file
Run QSPR Model and Discovery
QSPR Model and Discovery
2: 26-May-2011 16:55:36
CSVLoad
Input data set
Column Sele
Desc Names
Descriptors
Transpose
Filter Featur
Export Files
Headers
Data
Column Join
Workflow nam
Dummy file
xValidate LM
2: 26-May-2011 17:0...
QSPR header
QSPR data
xValidate Lin
CSVExport
Results
Predicted vs actuals header
Predicted vs actuals data
PLS Model Builder
2: 27-May-2011 17:02:46
PLS Model
Model file
Stratify Data
xValidate PLS
xValidation results
Predicted vs actuals header
RPart Model Builder
3: 27-May-2011 17:04:58
RPart Model
xValidate RP
NN Model Builder
3: 27-May-2011 17:06:11
Neural Net M
Training results
xValidate Ne
Linear Model
modelId, featureList
Smart Colum
ExecWF
Run xValidation
Saved OK

QSAR Implementation



e-science central

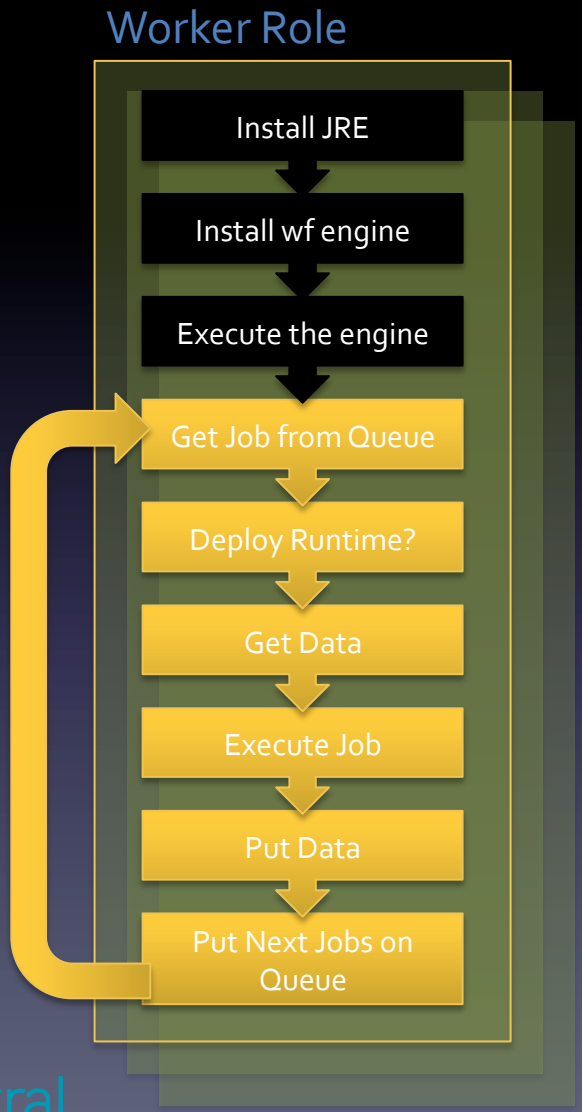
Azure e-Science Central



e-science central

Workflow Architecture

- Single Message Queue
  - Worker Failure Semantics
  - Elasticity
- Runtime Environments
  - R
  - Octave
  - Java
- Deployed only once



e-science central

Results

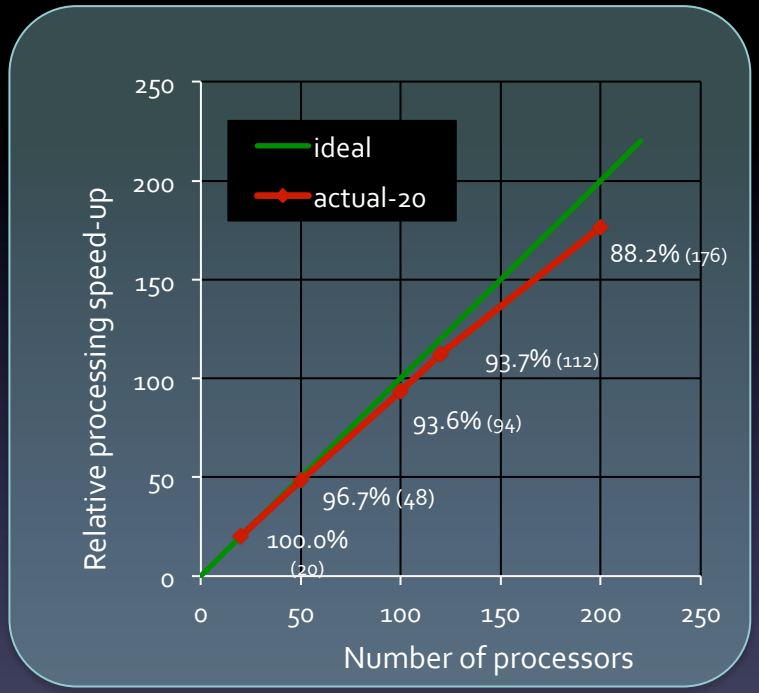
- 460K workflow executions
- 4.4M service calls

- 250k models
  - Linear Regression
  - PLS
  - Rpartitioning
  - Neural Net

- QSAR Explorer
  - Browse
  - Search
  - Get Predictions

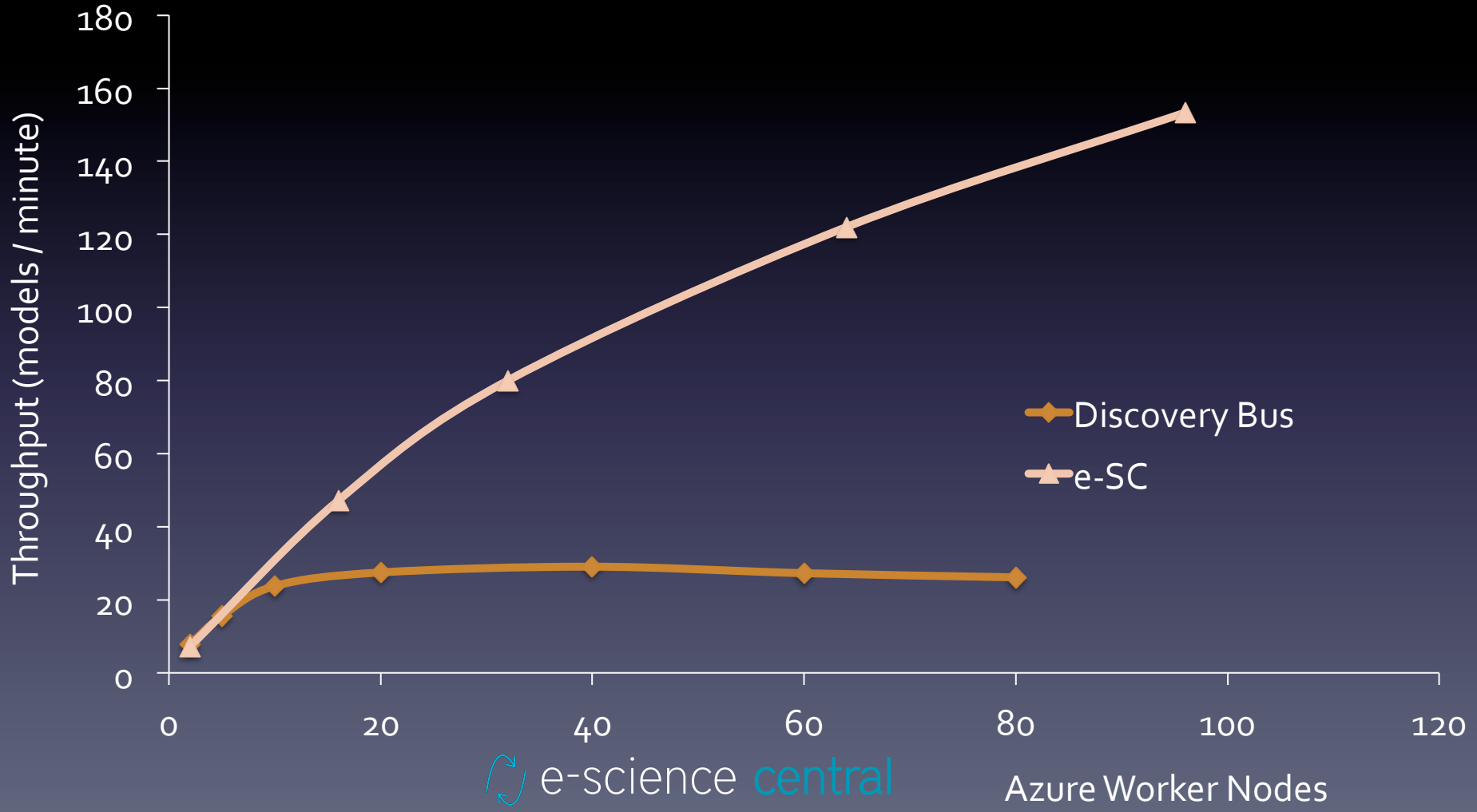
Evaluation

Number of cores	1 + server	20 + server	200 + server
Response time	11d 20h 03m 40s	13h 00 m 11s	1h 28m 28s
Speed-up	1	20	176
Cost	$62.64	$40.32	$51.84



e-science central

Evaluation


Throughput (models / minute)
180
160
140
120
100
80
60
40
20
0
0
20
40
60
80
100
120
Discovery Bus
e-SC
e-science central
Azure Worker Nodes

Cloud Applicability

- Bursty
  - ChEMBLdb updates (delta 10%)
  - New Modelling Methods (???)
- Performance depends on how *chatty* the problem is
  - Deploy (incl download) dependencies once
  - Avoid storage bottlenecks

Performance is great but …

Drug Development requires us to capture the

data and the process

Provenance Requirements

- How was a model generated?
  - What algorithm?
  - What descriptors
- Are these results reproducible?

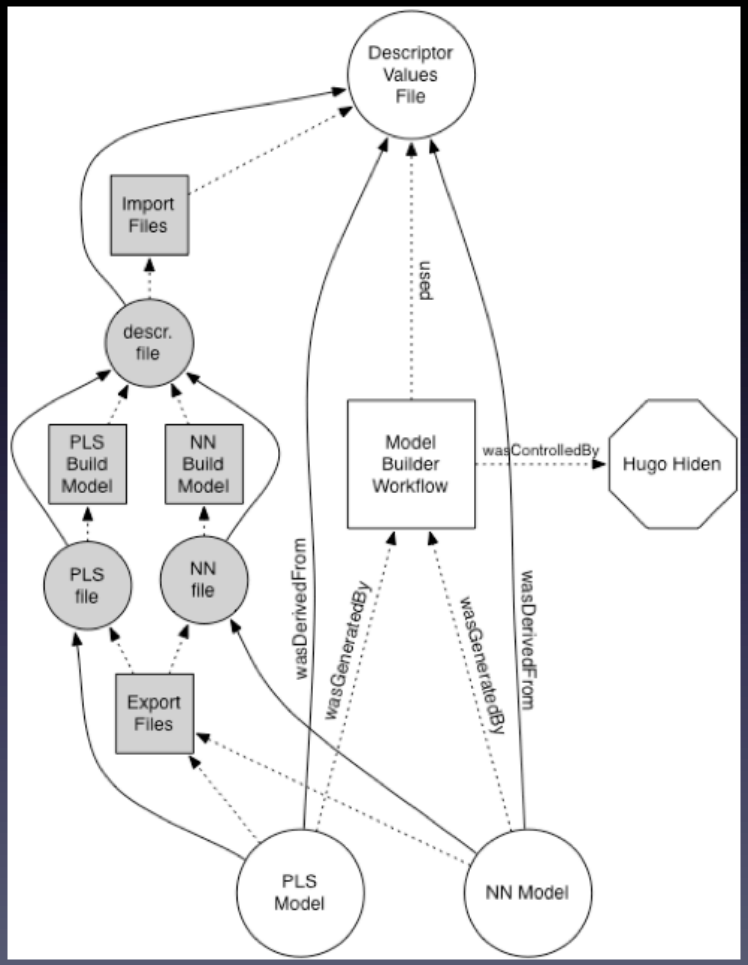
- How have bugs manifested?
  - Which models affected
  - How do we regenerate affected models?
- Performance Characteristics
- How do we deal with new data?

Storing Provenance

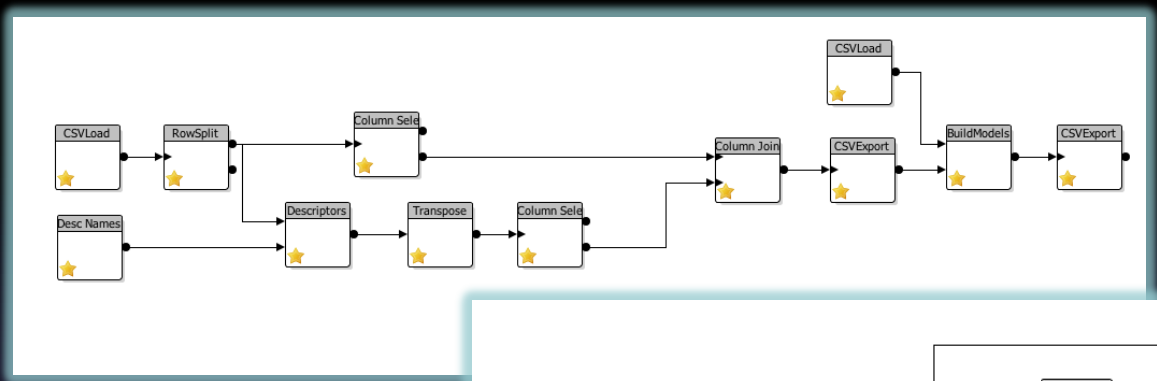
- Neo4j
  - Open Source Graph Database
  - Nodes/Relationships + properties
  - Querying/traversing
- Access
  - Java lib for OPM
  - e-SC library built on top of OPM lib
  - REST interface
- Options for HA and Sharding for performance

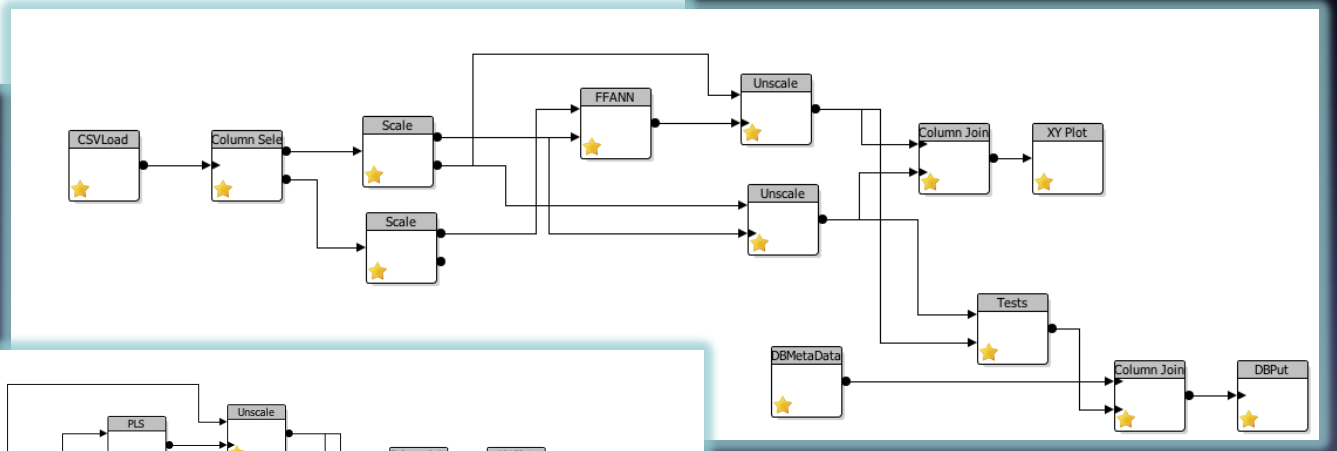
Provenance Model

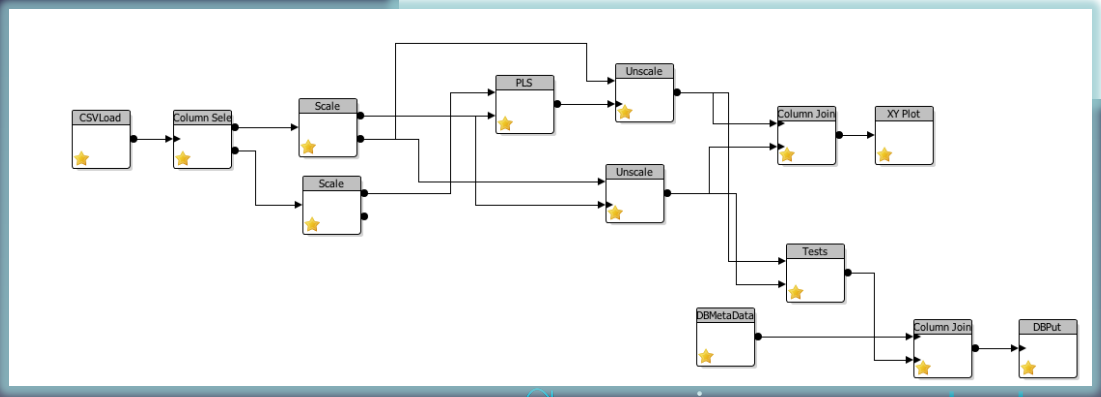
- Based on OPM
  - Processes, Artifacts, Agents
- Directed Graph
- Multiple views of provenance
  - Dependent on security privileges



Multiple Linked Workflows



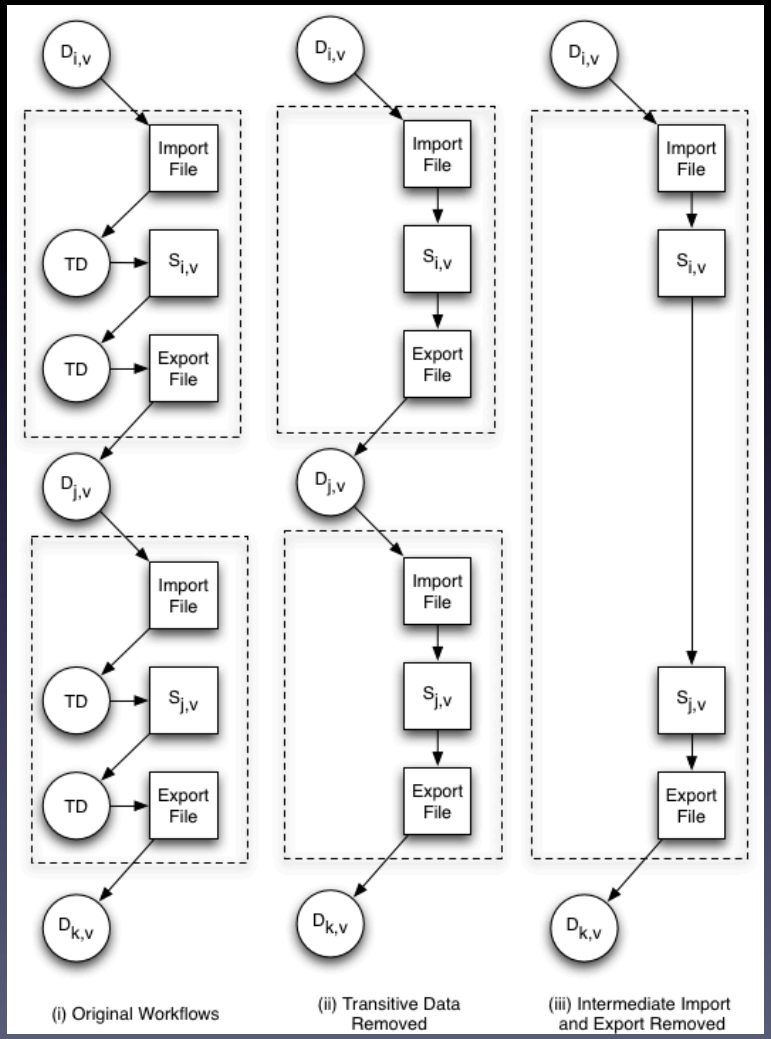




e-science central

Spanning Multiple Workflows

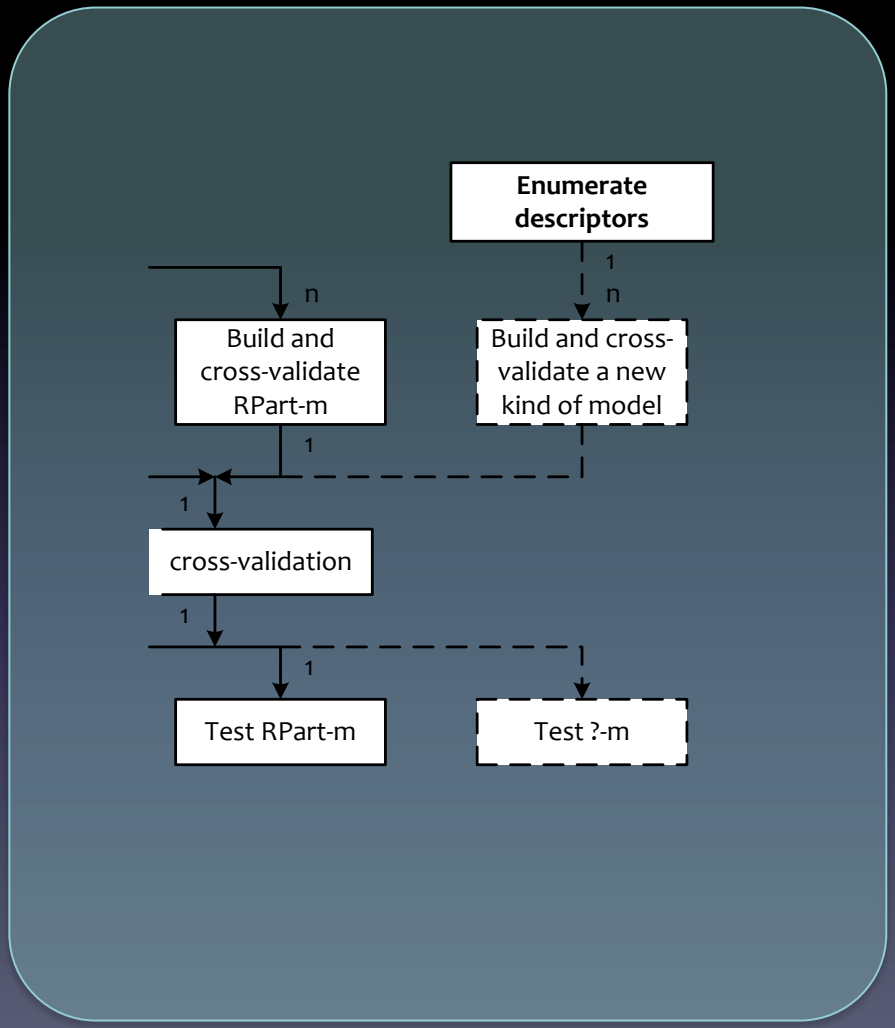
- Regenerate Data with services updated
  - Not everything in a single workflow
- Create a single "Virtual Workflow" spanning multiple physical workflows
- Work performed on data sets by different people over a period of time



e-science central

Adding new model builders

1. Add new block
2. Mine the provenance
3. Dynamically create virtual workflows
4. One invocation per data set



- Work in progress…

e-science central

Future Work

- Scale 200+ nodes
  - Database is bottleneck
- Provenance visualization
- Meta-QSAR
  - Provenance Mining

Questions?

- Thank you to our generous funders
  - EU FP7 - VENUS-C (RI-261565)
  - RCUK – SiDE (EP/G066019/1)