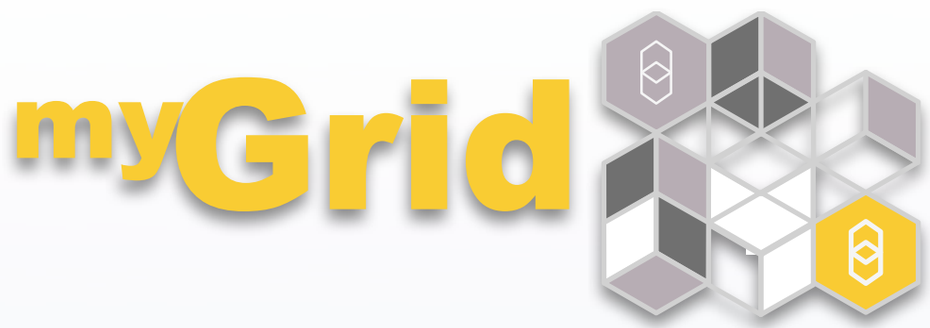


A taste of the future

Taverna 2

Stian Soiland

e-Science and "The Grid" for Bio/Health Informaticians
Manchester, 2008-01-15

Overview #1

Iteration

- Implicit iteration
- Errors while iterating
  - t2 error documents
- Error recovery
  - Retry
  - Failover (Alternate)

Overview #2

Taverna 1 limitations

- Long pipeline slows down workflow
- Large data difficulties
- t1 hacks and best practice workarounds

Taverna 2 improvements

- Streaming of workflow data
- Data types
- Data references and identifiers

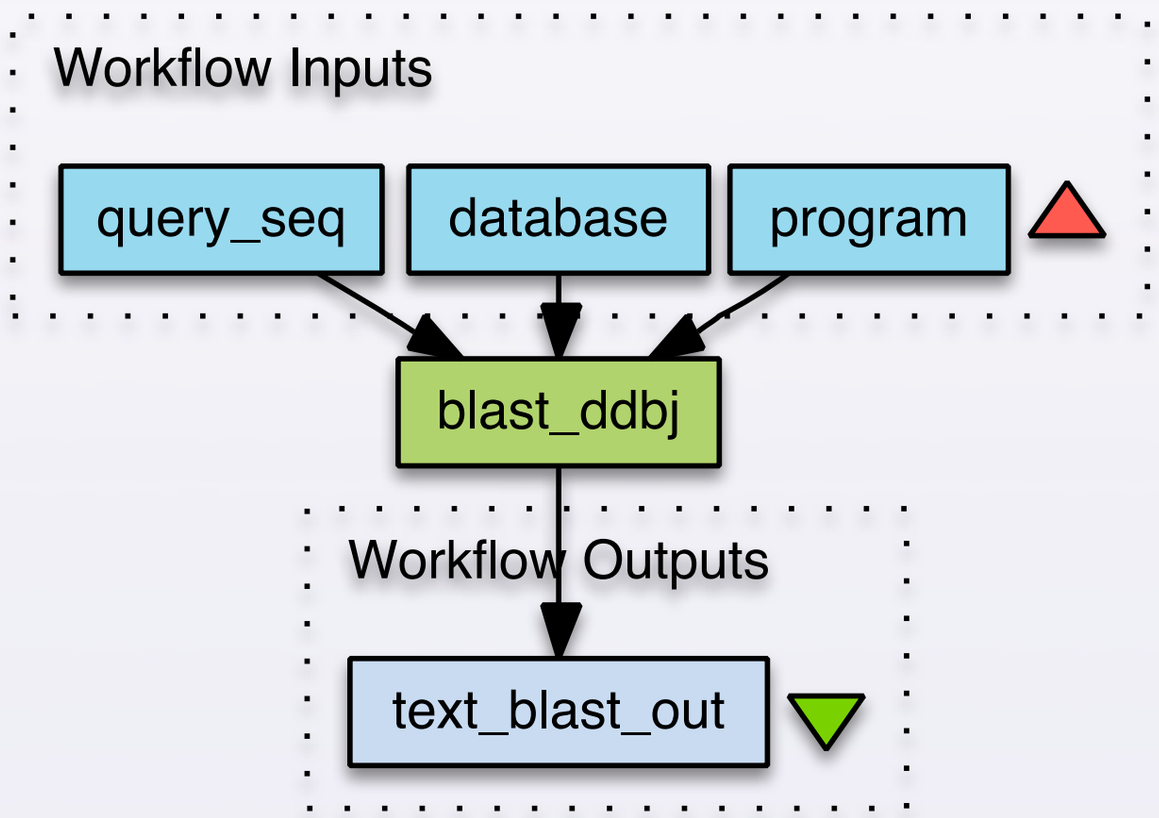
Overview #3

Taverna 2 future

- Multiple enactors
- The invocation stack (scary)

Iteration: Simple workflow

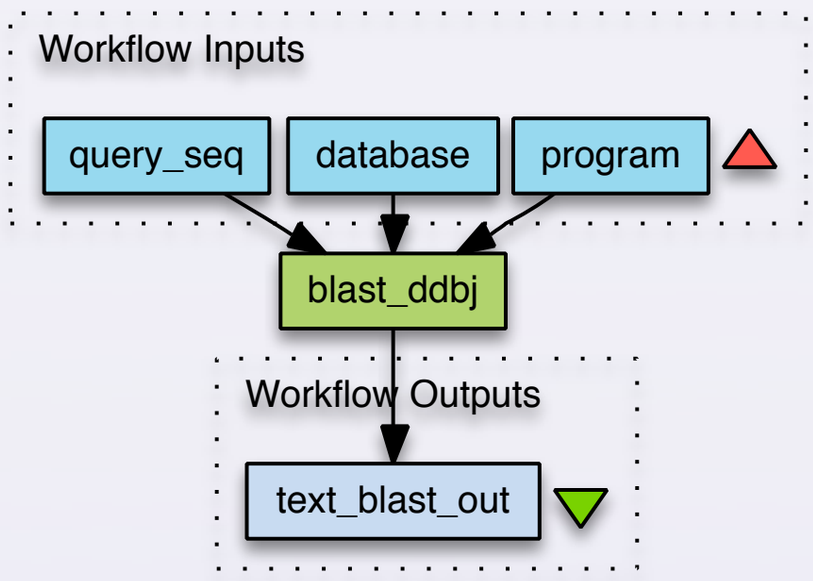
Assume a very simple workflow:



Paul Fisher, http://www.myexperiment.org/workflows/23

Input ports' syntactic type

All inputs of blast_ddbj have *syntactic type* 'text/plain'

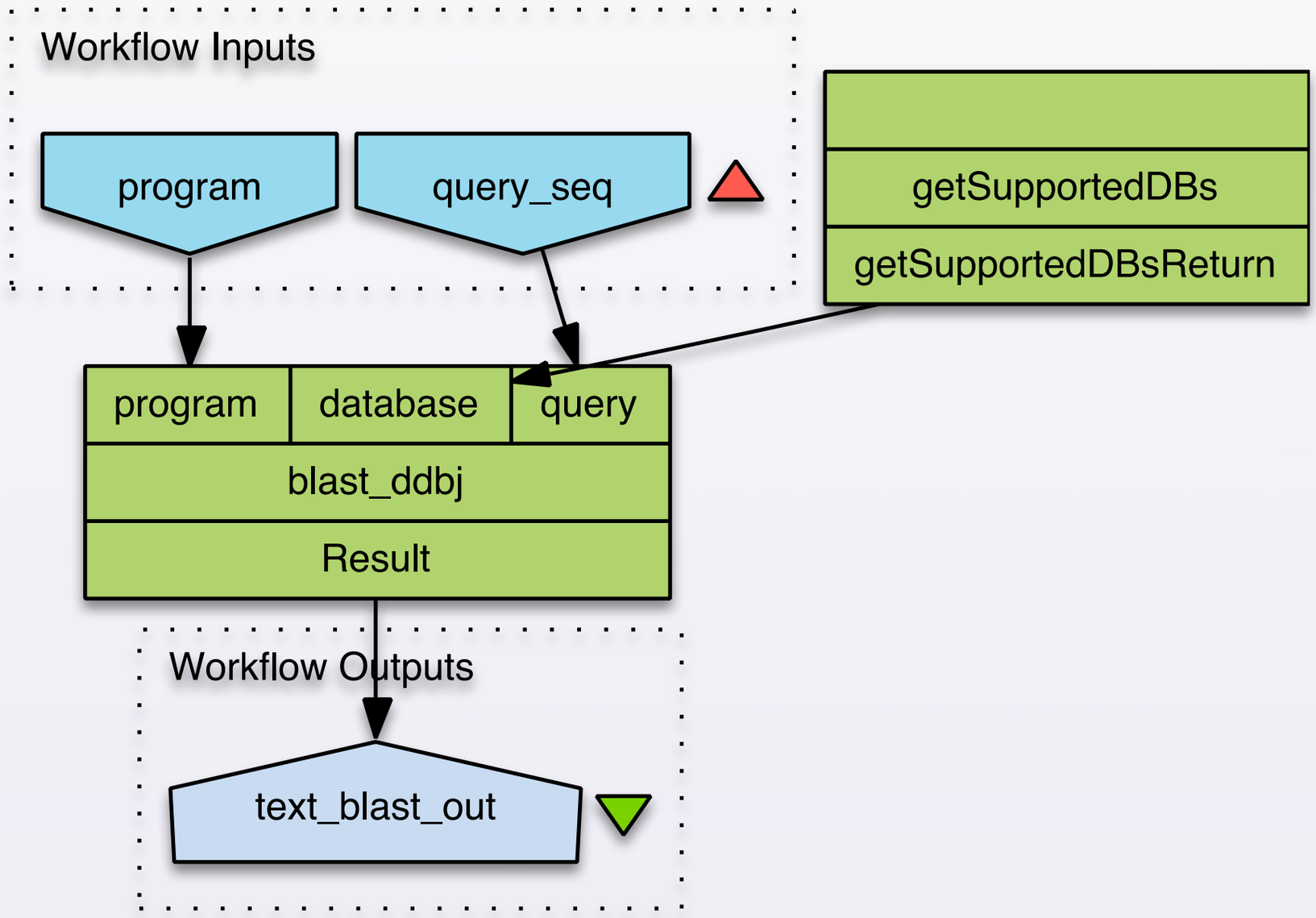


List type

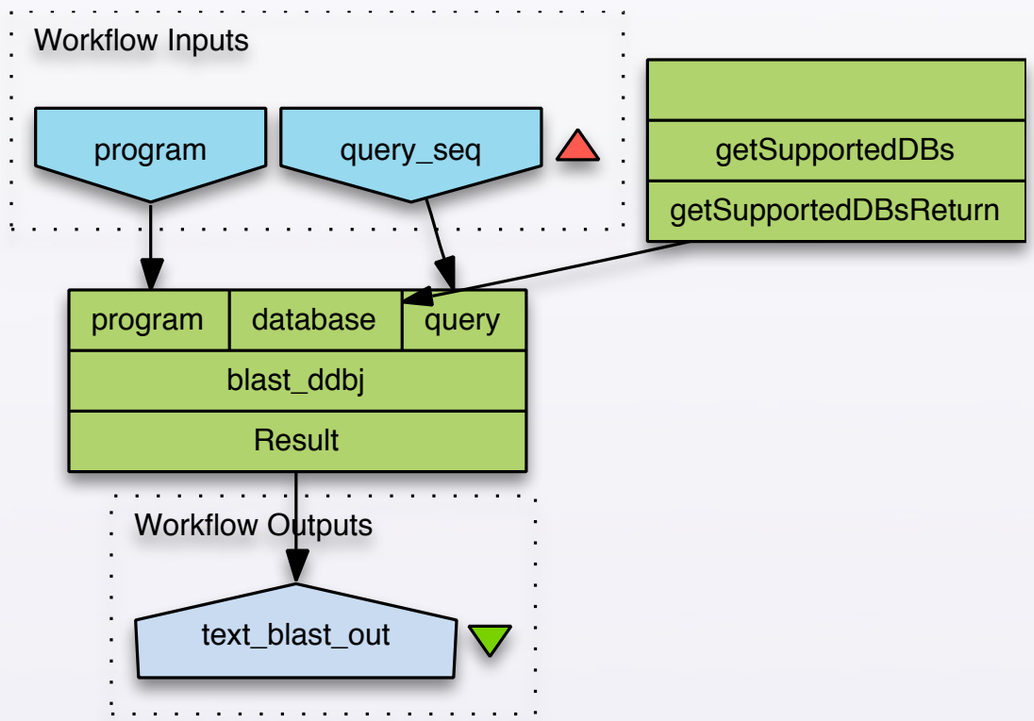
- But here's a service that outputs l('text/plain')

- The l() means a return of **list of strings**

Implicit iteration



Implicit iteration results



- Workflow output is now a **list** of blast reports, one for each database
- See whiteboard for Stian's scribblings

t2 terminology: list **depth**

- **l**('text/plain') --> depth **1**
- **l(l**('text/plain')) --> depth **2**
- ..

and of course..

- text/plain --> depth **0**

t1 issue: Errors in iterations

- What if the fourth element of an iteration fails?

Fourth element fails rest of wf

Taverna Workbench v1.7.0.0

Design | Results | T2 Activity palette preview | Taverna 2 preview | Discover

fail_on_4th 10:12 AM

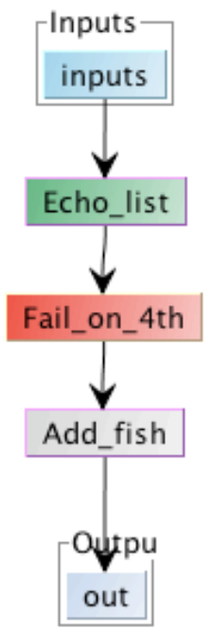
Save as XML | Save to disk | Save to disk as website | Excel | Close

Status | Results | Process report

Processor statuses

Ty...	Name	Last event	Eve...	Event detail	Breakpo...
	Echo_list	ProcessComplete	Jan ...		
	Fail_on_4th	ServiceFailure	Jan ...	Error running beanshell script: Sourced file: inline evaluation of: ``if (in.equals("4")) { ...	
	Add_fish	ProcessScheduled	Jan ...		

Graph | Intermediate inputs | Intermediate outputs



Why? To keep the ordering

Immediate output: “not 4”

List
urn:lsid:net.sf.taverna:dataCollection:27f0f17a-5792-4de7-906b-441790628ef6
text/plain
0
urn:lsid:net.sf.taverna:dataItem:b3caa31f-8bed-4eba-bd89-217f3486a30f
text/plain
1
urn:lsid:net.sf.taverna:dataItem:16e907a3-7c50-4390-9844-fef980816cc4
text/plain
2
urn:lsid:net.sf.taverna:dataItem:af2bd484-ca3c-
text/plain
3
urn:lsid:net.sf.taverna:dataItem:9b8ff841-8451-
text/plain
not 4
urn:lsid:net.sf.taverna:dataItem:41a59649-1ed9
text/plain
5
urn:lsid:net.sf.taverna:dataItem:07e259de-5f6e-

Results:

List
urn:lsid:net.sf.taverna:dataCollection:4a045c70-4fb7-464e-b379-8b2404f98e13
text/plain
Fish0
urn:lsid:net.sf.taverna:dataItem:2b4626c7-b19a-4790-bd54-628ad45ddd0b
text/plain
Fish1
urn:lsid:net.sf.taverna:dataItem:ae169480-209e-4238-aaf4-b34e3cd00870
text/plain
Fish2
urn:lsid:net.sf.taverna:dataItem:2c0b339b-0afd-418e-b7eb-3b623c250540
text/plain
Fish3
urn:lsid:net.sf.taverna:dataItem:4ce9d70c-4cee-4804-8ae6-56543a7719ea
text/plain
Fishnot 4
urn:lsid:net.sf.taverna:dataItem:42798cd9-f29c-488e-89d1-b654eb029481
text/plain
Fish5
urn:lsid:net.sf.taverna:dataItem:478bb49c-1df0-4ea1-9c9a-c77d8e6097ec

But #4 is not a result!

in t2: Error documents

List
urn:lsid:net.sf.taverna:dataCollection:27f0f17a-5792-4de7-906b-441790628ef6

text/plain
0
urn:lsid:net.sf.taverna:dataItem:b3caa31f-8bed-4eba-bd89-217f3486a30f

text/plain
1
urn:lsid:net.sf.taverna:dataItem:16e907a3-7c50-4390-9844-fef980816cc4

text/plain
2
urn:lsid:net.sf.taverna:dataItem:af2bd484-ca3c-4c63-9ee5-e08fbbaf8d0f

text/plain
3
urn:lsid:net.sf.taverna:dataItem:9b8ff841-8451-43c2-ae03-8d0ec2d345b7

Error in processor *Fail_on_4th*:
`java.lang.Exception: Can't handle 4`

text/plain
5
urn:lsid:net.sf.taverna:dataItem:07e259de-5f6e-4b23-b8ca-fe3dc3e3960a

Error documents acts as placeholders for the results that “should have been there”

t2: Propagation of errors

- Error documents float through the workflow as if they were data

List
urn:lsid:net.sf.taverna:dataCollection:4a045c70-4fb7-464e-b379-8b2404f98e13

text/plain
Fish0
urn:lsid:net.sf.taverna:dataItem:2b4626c7-b19a-4790-bd54-628ad45ddd0b

text/plain
Fish1
urn:lsid:net.sf.taverna:dataItem:ae169480-209e-4238-aaf4-b34e3cd00870

text/plain
Fish2
urn:lsid:net.sf.taverna:dataItem:2c0b339b-0afd-418e-b7eb-3b623c250540

text/plain
Fish3
urn:lsid:net.sf.taverna:dataItem:4ce9d70c-4cee-4804-8ae6-56543a7719ea

Error in processor *Fail_on_4th*:
`java.lang.Exception: Can't handle 4`

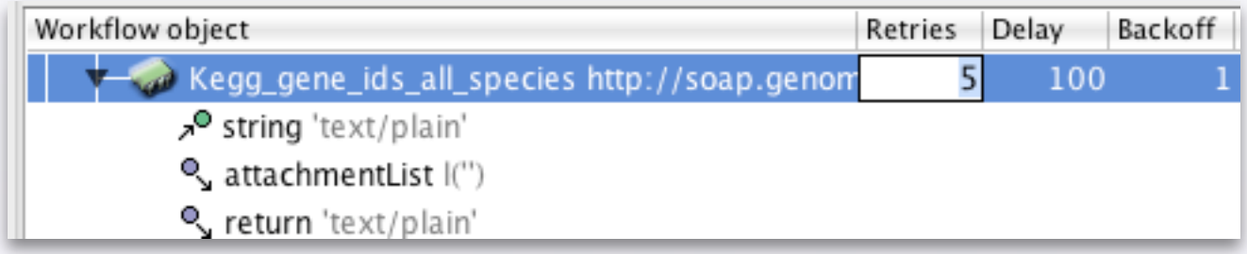
text/plain
Fish5
urn:lsid:net.sf.taverna:dataItem:478bb49c-1df0-4ea1-9c9a-c77d8e6097ec

t1: Error recovery

- Taverna already has two main ways to recover from errors

Retries

- Can specify how many times to retry (Retries)
- And how much to sleep (delay) before trying again (in milliseconds)



- Advanced: Can incrementally increase the delay using a multiplication factor (Backoff)
  - Stay close to 1, say 1.1, to avoid exponential growth

Alternate processors

- Other services can be dragged onto a processor as an alternate

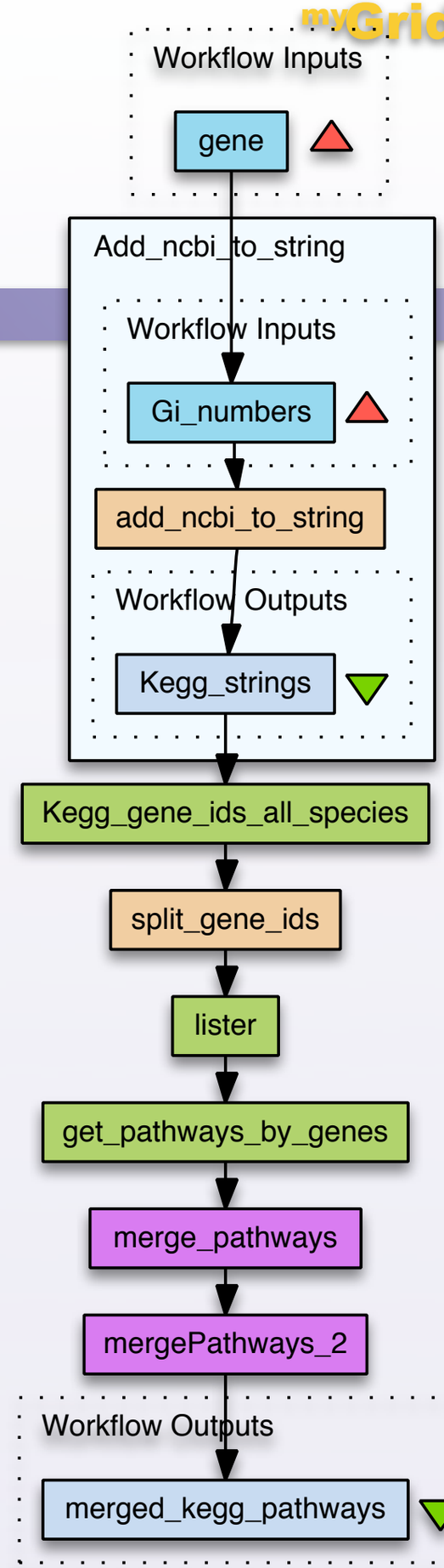
- Special case: Abstract processor
  - Non-invocable processor
  - Can add implementations as alternates later

t2: Activities and processors

- Taverna 2 separates between:
  - Processor (part of workflow)
  - Activity (invokes service)
- The processor performs the iteration, retries, failovers, etc
- The service contacts a concrete service (say at a SOAP endpoint)
- Processor can have 0 or more activities
- Abstract processor will mean no activities

Workflow as a pipeline

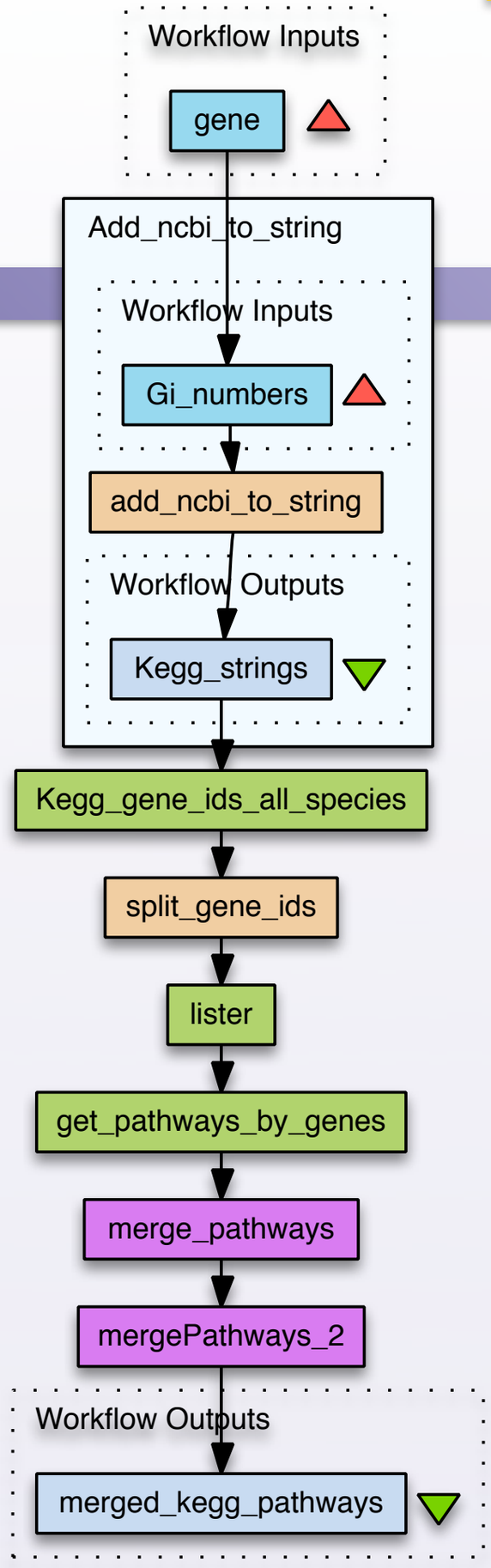
- OK, this one I did not make up:



Paul Fisher, http://www.myexperiment.org/workflows/15

Slow iteration

- Watch sleep_2s.xml in Taverna as Stian explains why this could take a while with 10-15 genes as inputs



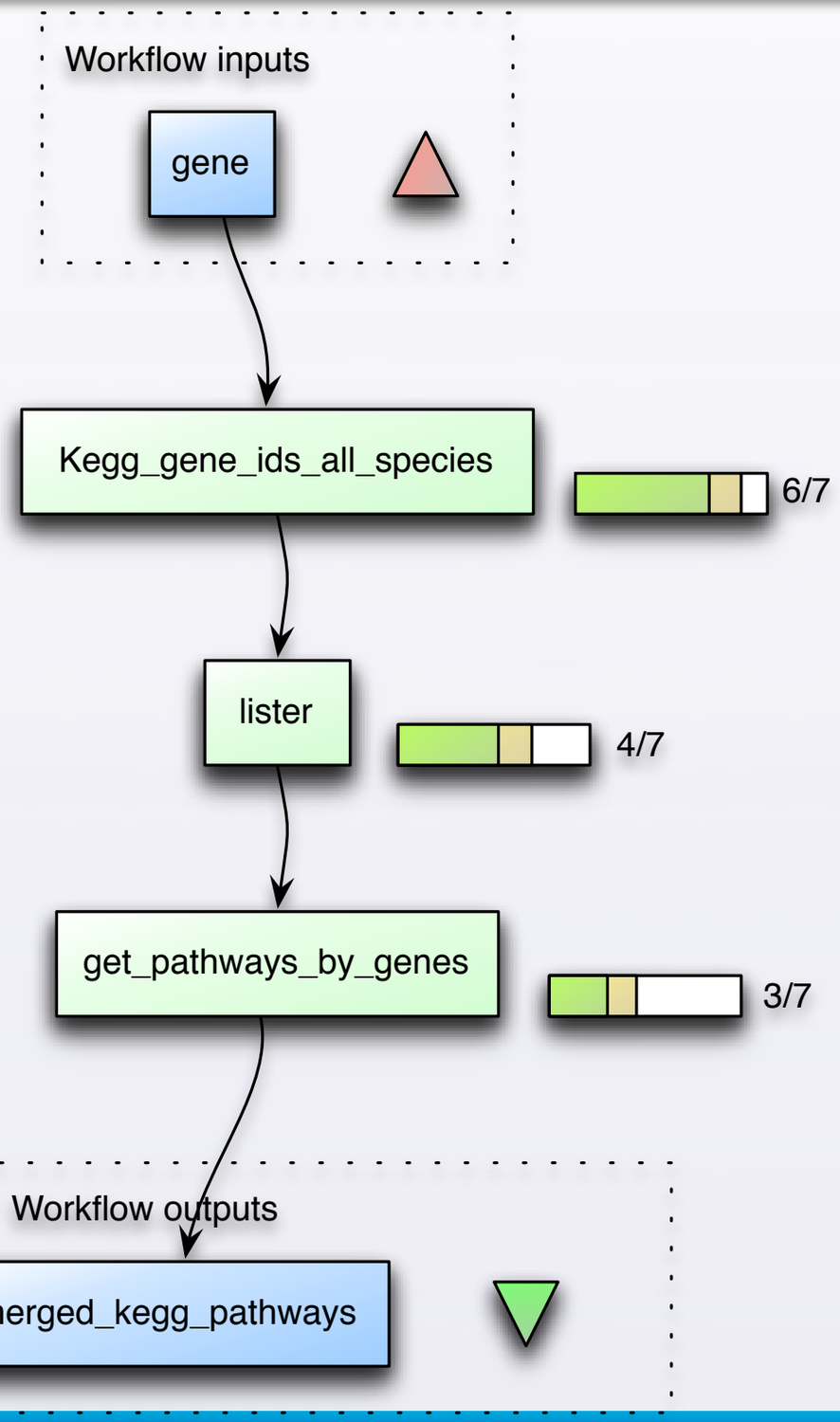
Quick t1 hack: More threads

- Increase threads slightly for the slow ones

Workflow object	Retries	Delay	Backoff	Threads	Critical
Workflow inputs					
gene					
Workflow outputs					
merged_kegg_pathways					
Processors					
split_gene_ids	0	0	1	1	
merge_pathways	0	0	1	1	
mergePathways_2	0	0	1	1	
get_pathways_by_genes http://soap.genome.j	0	0	1	3	
Kegg_gene_ids_all_species http://soap.genom	0	0	1	3	
Add_ncbi_to_string	0	0	1	1	
lister http://phoebus.cs.man.ac.uk:8081/axis/	0	0	1	3	

t2: Streaming

- See **whiteboard** for manual iteration by Stian



t2: Services can stream too

- Services that returns lists, such as BioMart
  - Create single data items for each line
  - See whiteboard
- Can even arrive out of order
- Infinite lists
  - Instruments
  - Updates
  - “Push”

Also slow: Big stuff

- t1: Anything bigger than a few MBs can be very slow
  - Mainly because it's stored in memory
  - Sent up and down several times over slow network connection in XML format

Big stuff over the wire

- t1 workarounds:
  - Instead of:
    - GATTGAGAGACCCAGAGG….
  - return a service-local identifier:
    - sequence:23231
  - See **whiteboard**
  - Or..re-use existing identification schemes
    - URI (Might even be fetchable)
    - LSID (Could be resolvable)
    - Specialised scheme (Uniprot, pubmed, etc)

Quick t1 hack: Data Proxy

- myGrid's Data Proxy wraps WSDL services
- Extracts selected elements of the schema and stores them to disk
- See **whiteboard**
- Problem: Server still needs to handle big data

t2: Big stuff

- t2: Several data managers can store data in:
  - memory
  - files
  - on another machine
- t2: Lists and errors are also data

t2 data types (Entities)

- Literal
  - Integer, Float, Double, etc.
  - Small strings (< 80 chars)
- Data document
  - With references to stored bytes/strings
- Error document
  - Error message
- List
  - Containing any of the above

t2: Reference schemes

- Data documents are just pointers to the real data
- References of any type
- Services might produce and consume references
- .. or still do old-style data on the wire

t2: Entity identifiers

- We try to practice what we preach, so t2 also uses references internally
- Lists, data documents and error documents have identifiers
- So a list of data documents don't actually contain the data documents, just the identifiers of those documents

t2: List with identifiers

The gory details.. can you spot the references?

```
<ent:entityList>
 <identifier>
    urn:t2data:list://6785b681-3cb3-4c99-9601-67c1d3379f67/
     69350508-83c3-4e7c-bd11-e97e7f22507c/1
 </identifier>
 <entity>
    urn:t2data:ddoc://6785b681-3cb3-4c99-9601-67c1d3379f67/
          a0fed96c-181f-40b9-9f22-13225b3b7c34
 </entity>
 <entity>
    urn:t2data:ddoc://6785b681-3cb3-4c99-9601-67c1d3379f67/
          d8d2c827-f8fa-4f8e-a555-d23f35ee5043
 </entity>
</ent:entityList>
```

t2: List with identifiers

The gory details.. can you spot the references?

```
<ent:entityList>
  <identifier>
    urn:t2data:list://6785b681-3cb3-4c99-9601-67c1d3379f67/
     69350508-83c3-4e7c-bd11-e97e7f22507c/1
  </identifier>
  <entity>
    urn:t2data:ddoc://6785b681-3cb3-4c99-9601-67c1d3379f67/
          a0fed96c-181f-40b9-9f22-13225b3b7c34
  </entity>
  <entity>
    urn:t2data:ddoc://6785b681-3cb3-4c99-9601-67c1d3379f67/
          d8d2c827-f8fa-4f8e-a555-d23f35ee5043
  </entity>
</ent:entityList>
```

t2: Data with multiple refs

The gory details.. can you spot the references?

```
<bean:dataDocument>
  <identifier>
     urn:t2data:ddoc://6785b681-3cb3-4c99-9601-67c1d3379f67/
                       d8d2c827-f8fa-4f8e-a555-d23f35ee5043
  </identifier>
  <reference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:type="file:fileReferenceScheme">
     <file>/Users/stain/kegg.esdl</file>
  </reference>
  <reference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:type="http:httpReferenceScheme">
     <url>http://genes.org/kegg.esdl</url>
  </reference>
  <reference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="blob:blobReferenceScheme">
     <namespace>6785b681-3cb3-4c99-9601-67c1d3379f67</namespace>
     <id>a82fed20-8bdc-41fc-b1b0-84c96d7d8266</id>
  </reference>
</bean:dataDocument>
```

t2: Data with multiple refs

The gory details.. can you spot the references?

```
<bean:dataDocument>
  <identifier>
     urn:t2data:ddoc://6785b681-3cb3-4c99-9601-67c1d3379f67/
                       d8d2c827-f8fa-4f8e-a555-d23f35ee5043
  </identifier>
  <reference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:type="file:fileReferenceScheme">
     <file>/Users/stain/kegg.esdl</file>
  </reference>
  <reference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:type="http:httpReferenceScheme">
     <url>http://genes.org/kegg.esdl</url>
  </reference>
  <reference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:type="blob:blobReferenceScheme">
     <namespace>6785b681-3cb3-4c99-9601-67c1d3379f67</namespace>
     <id>a82fed20-8bdc-41fc-b1b0-84c96d7d8266</id>
  </reference>
</bean:dataDocument>
```

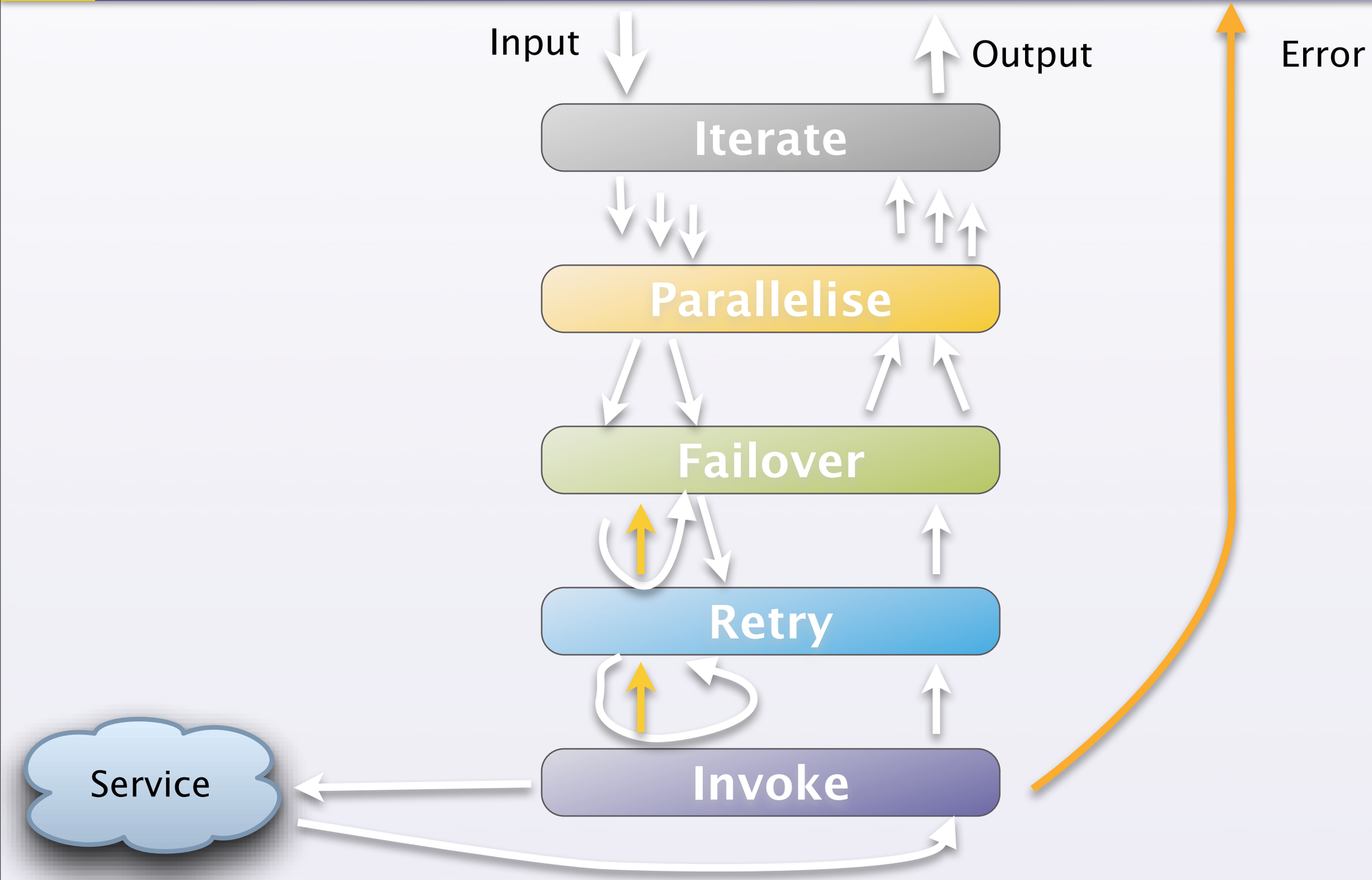
t2 future: Multiple enactors

- A workflow can run on a combination of machines
- Enactors communicate using a p2p protocol
- Entities and data are shared if needed
- Cooperate to get access to resources

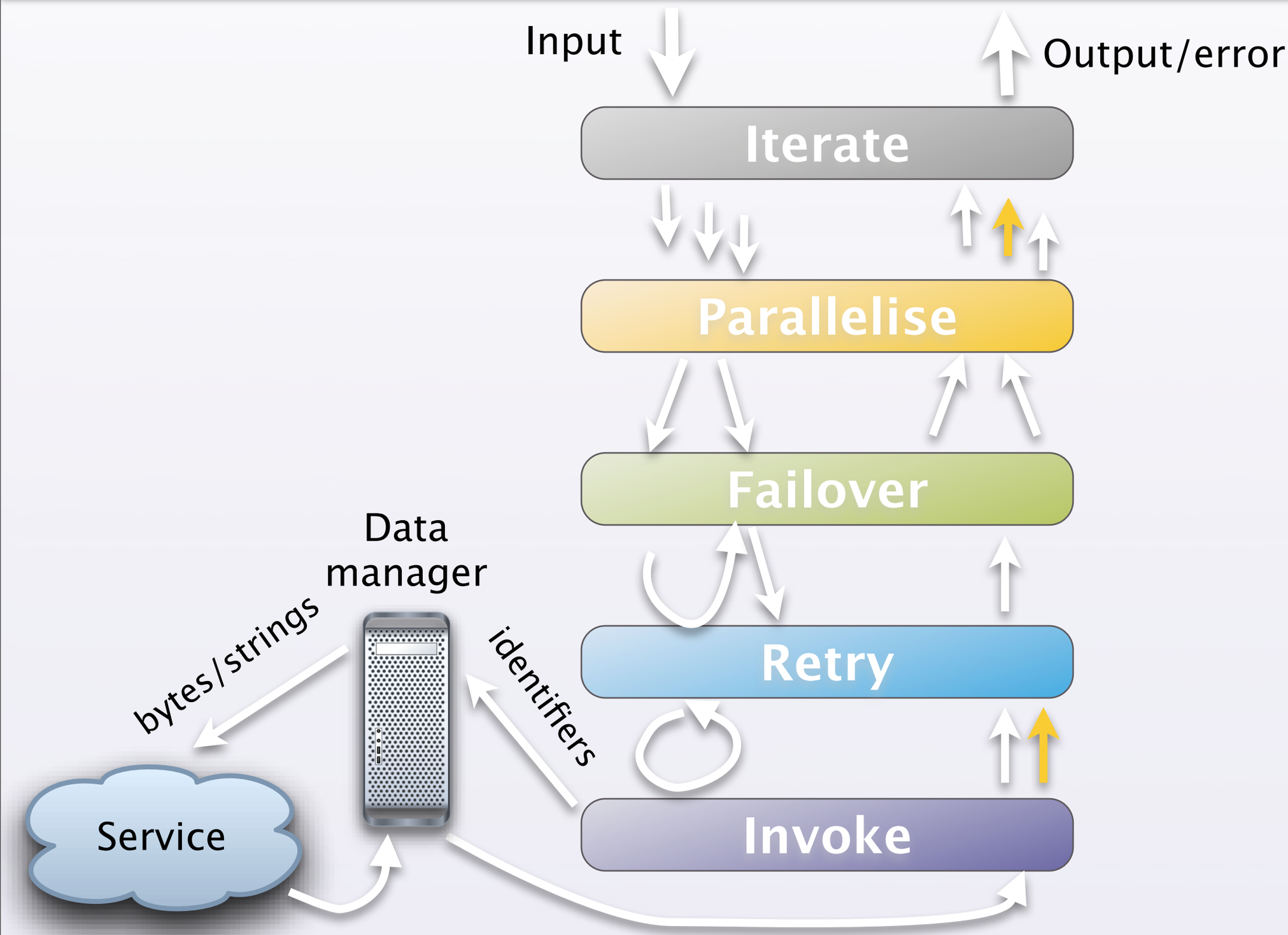
Recap of workflow

- Processor contains activities
- Processor does:
  - Implicit iteration
  - Delay and retry
  - Failover to alternate activity
  - Map to selected activity
- Activity does:
  - Invoke the service
  - Register result data

Taverna 1 invocation stack



Taverna 2 default stack



t2: stack can be changed

- For advanced users..
  - Failover first, then retry
  - Failover for the full iteration
  - Look up services at run-time
  - Parallelise on several computers
  - … anything

Where do we go next with t2?

- Taverna 1.7 was relased 2007-12-17
  - Includes a t2 “taster” plugin
    - Still quite buggy, but can test workflow for compatability
  - New available services palette
- More updates for t2 to follow in 2008
- Improving/hardening APIs to ease the transition for 3rd party projects like BioMoby
- Update GUI for new features
- Grid and security