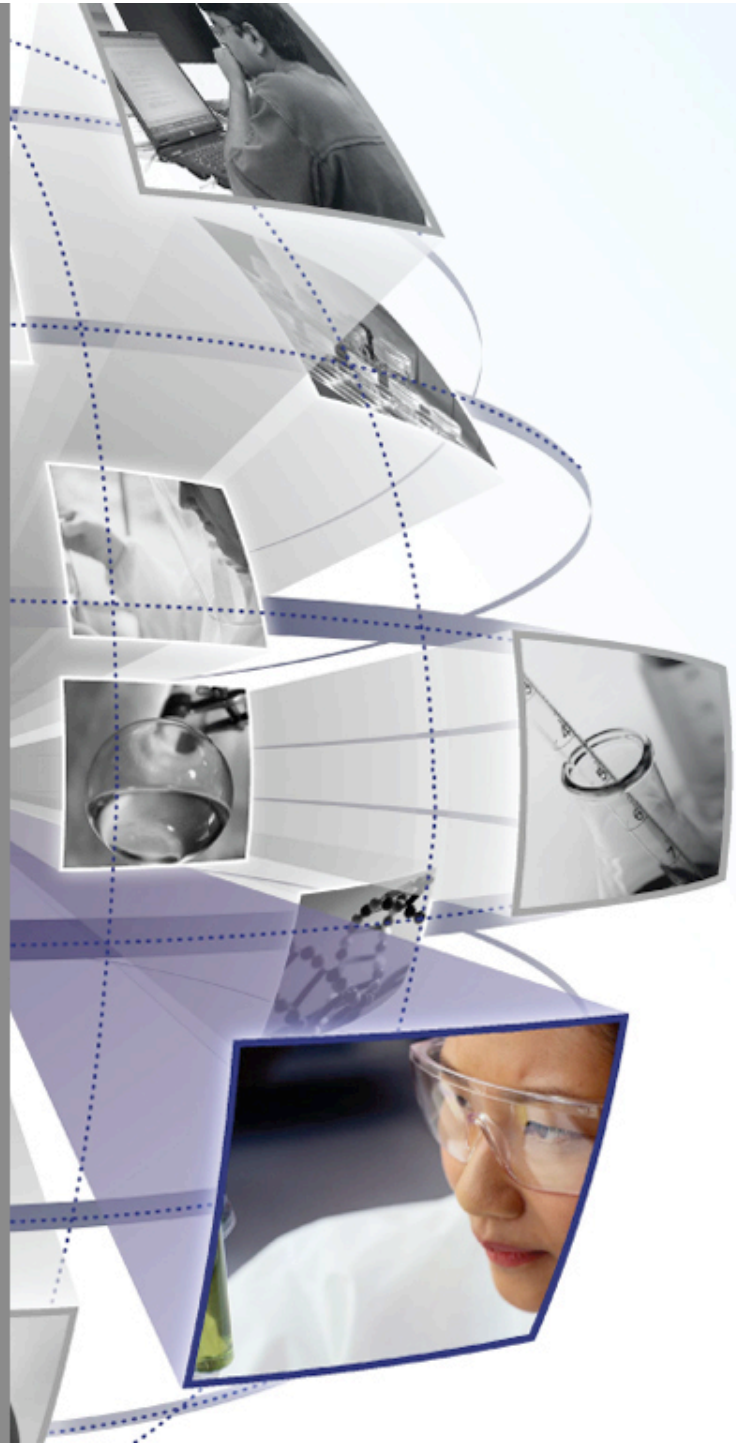
National Cancer Institute



caBIG™
cancer Biomedical Informatics Grid ™

Orchestrating Grid Services Using Taverna

Wei Tan UC
Stian Soiland-Reyes myGrid, UK
Ravi K Madduri ANL/UC

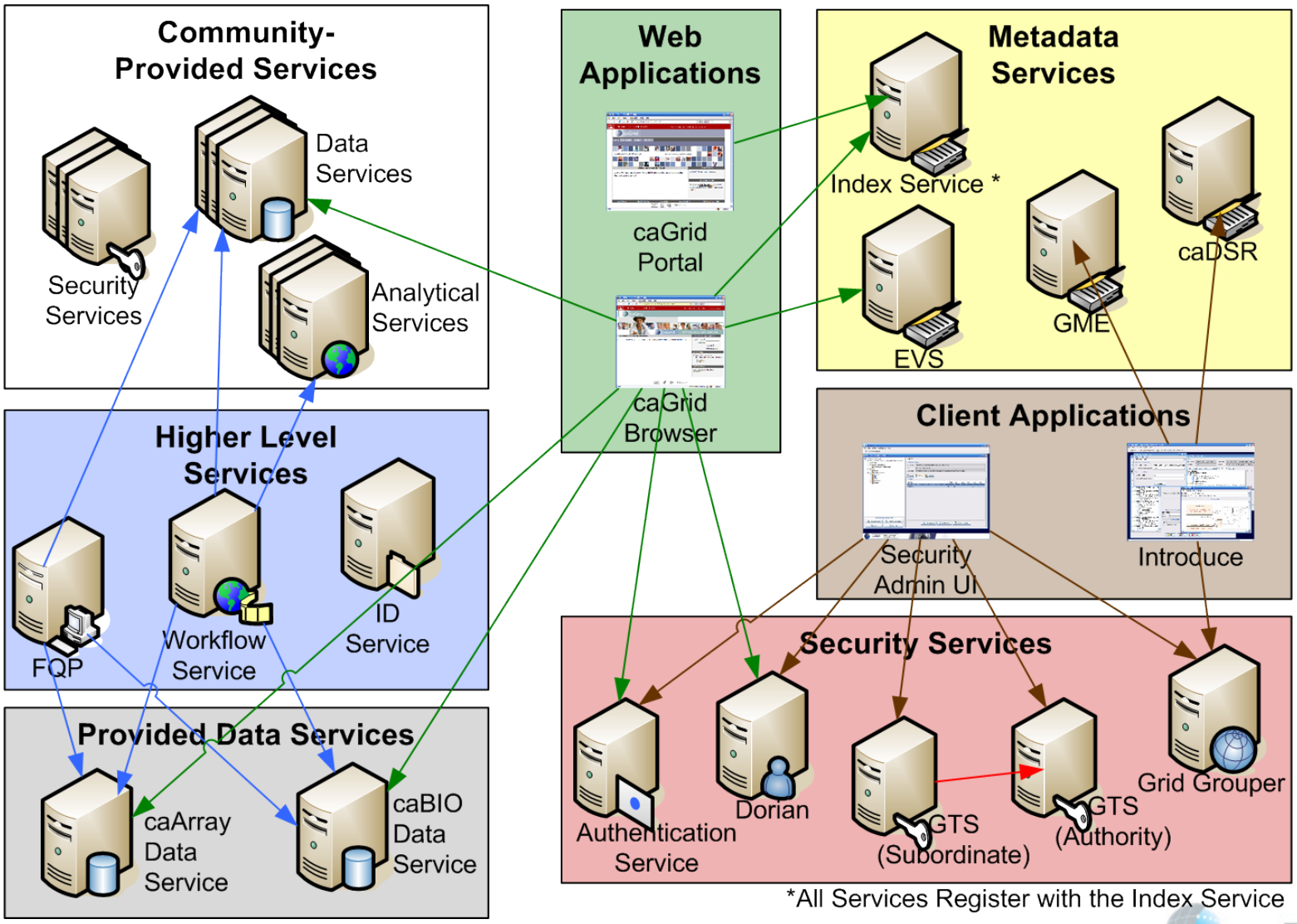
Agenda

- Background of Workflows in caGrid
- ICR Working group of caBIG
- Taverna
- Discussion

caGrid background
What is caGrid?

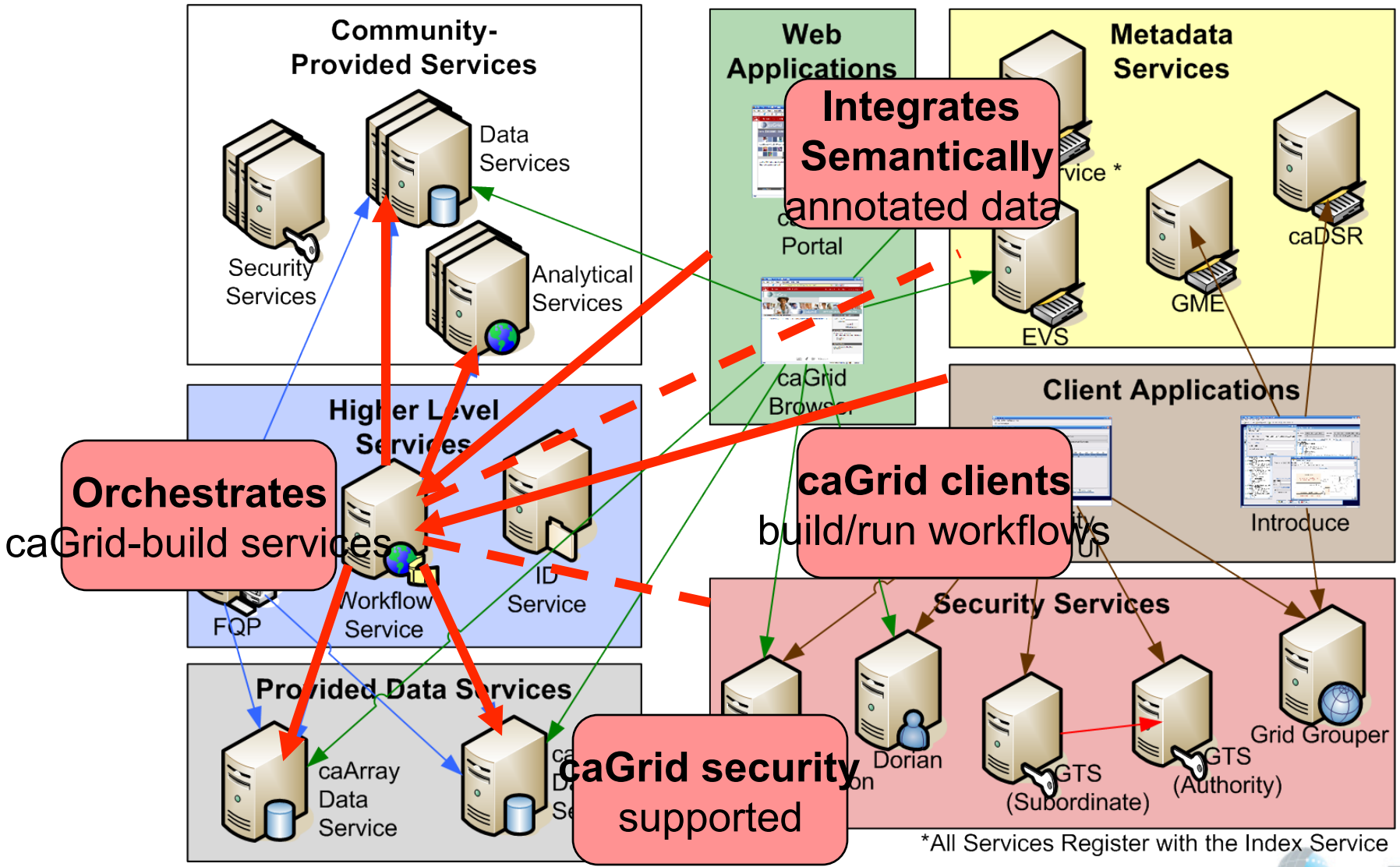
- **What is Grid?**
  - Evolution of distributed computing to support sciences and engineering
  - Sharing of resources (computational, storage, data, etc)
  - Secure Access (global authentication, local authorization, policies, trust, etc.)
  - Open Standards
  - Virtualization
- **What is caGrid?**
  - Development project of Architecture Workspace
    - Helping define and implement Gold Compliance
  - Implementation of Grid technology
    - Leverages open standards, community open source projects
  - No requirements on *implementation* technology necessary for compliance
    - Specifications will be created defining requirements for interoperability
    - caGrid provides core infrastructure, and tooling to provide “a way” to achieve Gold compliance
  - Gold compliance creates the ***G*** in caBI***G***™
    - **G**old => **G**rid => connecting Silver Systems

caGrid background
caGrid overview



caGrid background

Where does workflow fit in?



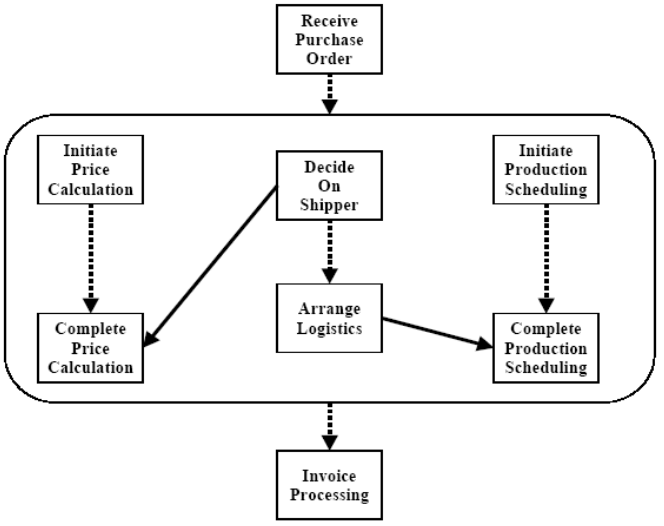
Workflow background
What is workflow?

- **The connecting of services to solve a problem that each individual service could not solve**
- **In bioinformatics, this is sometimes referred to as a pipeline**
- **Could mimic to some process in the real world**
- **Grid-aware scripting language**
- **Other possible definitions/uses of workflow**
  - Tracking samples in a LIMS
  - Tracking patient data through protocols in CTMS

caBIG cancer Biomedical Informatics Grid™

Workflow background
What is a service workflow?

- **High-level scripting for frequently executed tasks**
  - Often automates a manually driven sequence
  - Powerful manner of composing scripts from services
- **Benefits over regular programming**
  - Parallelism: not as easy to do in Java
  - Persistence: keeps track of state for long-running scripts
  - Better fault recovery: engine automatically retries failing calls
  - Powerful semantics for failure action – compensation handling
- **Canonical pattern for service workflows**
  - receive – input message and trigger to start
  - declare variables – all local to the workflow
  - Invoke services, assign variables, loops, etc
  - Return final results.



Workflow background
How does workflow fit into caBIG?

- **caBIG is a…**
  - Common, widely **distributed infrastructure** that permits the cancer research community to focus on innovation
  - Shared, harmonized set of terminology, data elements, and data models that facilitate information exchange
  - Collection of **interoperable applications** developed to common standards
  - **Cancer research data** is available for mining and **integration**
- **Workflow enables…**
  - Accessing **distributed services** in flexible patterns
  - **Integrating data and analytic services** with flexible control-flow patterns
    - Loops, conditionals, iteration over collections
  - Type-safety: verifying data-type correctness of arguments passed between services
  - Robustness: recover and continue long running workflows after failures
  - **Usability and integration**: specify workflows in graphical interfaces and scripted textual form
  - Record data provenance of workflow results

Workflow background
What is the BPEL?

- **Workflows in caGrid are described by the Business Process Execution Language (BPEL)**
  - Under standardization at OASIS
  - Integrates well with web services (WSDL)
- **Described in an XML document**
- **Work done via Service invocations**
  - “partner links” represent service endpoints
- **Looping, conditionals, parallel flows**
  - Specifies the order in which services are executed
- **Data objects copied from outputs to inputs**
  - Variables hold data
  - XPath used to select data
- **Event-driven message exchanges allowed**
- **Dynamic service discovery**

Workflow background
BPEL example

```
<receive createInstance="yes" operation="startWorkFlow"
      partnerLink="WorkFlowClientPartnerLinkType"
      portType="ns2:startWorkFlowPortType"
      variable="workFlowInputMessage" />
  <assign>
      <copy>
        <from expression=""1"" /> <to variable="indexCounterDuke" />
      </copy>
      <copy>
         <from part="parameters" query="/ns1:WorkFlowInputType/query"
               variable="workFlowInputMessage" />
         <to part="parameters" query="/ns1:query" variable="queryInputMessage" />
      </copy>
  </assign>

  <invoke inputVariable="queryInputMessage" operation="query"
          outputVariable="queryOutputMessage"
          partnerLink="RproteomicsDataLinkType" portType="ns1:RPDataPortType" />

  <assign>
    <copy>
      <from expression="count(bpws:getVariableData('queryOutputMessage', 'parameters',
                              '/ns1:queryResponse')/response/ns4:CQLQueryResult) div 2" />
      <to variable="countDuke" />
    </copy>
  </assign>
```

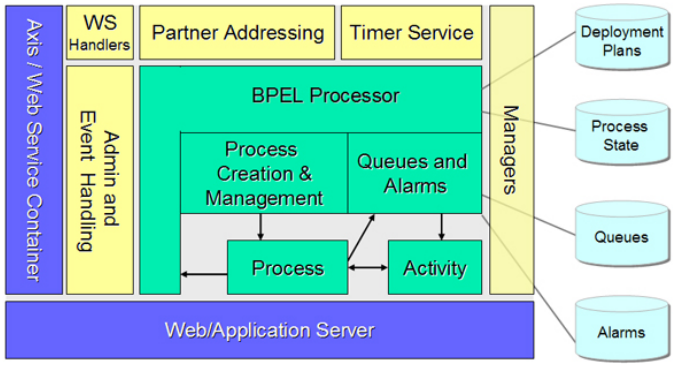
Workflow background
BPEL iteration example

```
<while condition="bpws:getVariableData('indexCounterDuke')
                    <= bpws:getVariableData('countDuke')">
   <sequence>
    <assign> ... </assign>
    <invoke operation="denoise_waveletUDWTWByValue"
            inputVariable=
               "denoise_waveletUDWTWByValueInputMessageDuke"
            outputVariable=
               "denoise_waveletUDWTWByValueOutputMessageDuke"
            partnerLink="DukeRproteomicsPartnerLinkType"
            portType="ns3:RProteomicsPortType" />
    <assign> ... </assign>
  </sequence>
</while>
```

Workflow in caGrid

How does caGrid implement workflow?

- **Workflow Factor Service (WFS)**
  - Grid service to create a new workflow
- **Workflow Service**
  - Grid service to access your created workflow
  - Start, pause, resume, cancel, getWorkflowOutput
- **caGrid integration**
  - Invoke grid services
  - Security (communication, message, conversation)
- **caGrid implementation**
  - Leverages the ActiveBPEL workflow engine
  - Workflows exposed as web services in ActiveBPEL, wrapped as grid services
  - Wraps the ActiveBPEL Admin Service
    - WFS submits a BPR (workflow package)
    - Accesses the created stateful web service



The future of caGrid workflow

- **Dynamic discovery**
  - Select workflow endpoints based on search criteria
- **Provenance**
  - Tracking all actions of workflows
- **Workflow management service enhancements**
  - Share workflows
- **Identifier Integration**
  - Demonstrate use of identifiers and out-of-band data transfer
- **Optimized data flow**
  - Pass data directly from service to service
- **Grid cache**
  - Storing intermediate results
  - Manipulate data by reference (via identifiers)

Users Reaction (from ICR Working Group)

- **Execute grid analytical and data services to accomplish a pre-defined task.**
- **Authoring of workflows should be easy.**
- **Submission of workflows should be easy.**

Results: Authoring Tool

- **Conducted extensive tool review of existing tools: http://gforge.nci.nih.gov/docman/view.php/332/7509/icr_workflow_tool_review_2007.doc**
- **Prioritized feature list: http://gforge.nci.nih.gov/docman/view.php/332/9690/icr_workflow_prioritized_feature_list_2007.xls**

Results: Authoring Tool - Taverna

- **Taverna:**
  - http://gforge.nci.nih.gov/docman/view.php/332/8007/caGrid_workflow_taverna_2007.ppt
  - http://gforge.nci.nih.gov/docman/view.php/332/8278/taverna_tom_oinn.doc
  - Ravi Madduri (Cagrid workflow point of contact) attended Taverna workshop
  - Prototype of Taverna to discover and invoke caGrid services

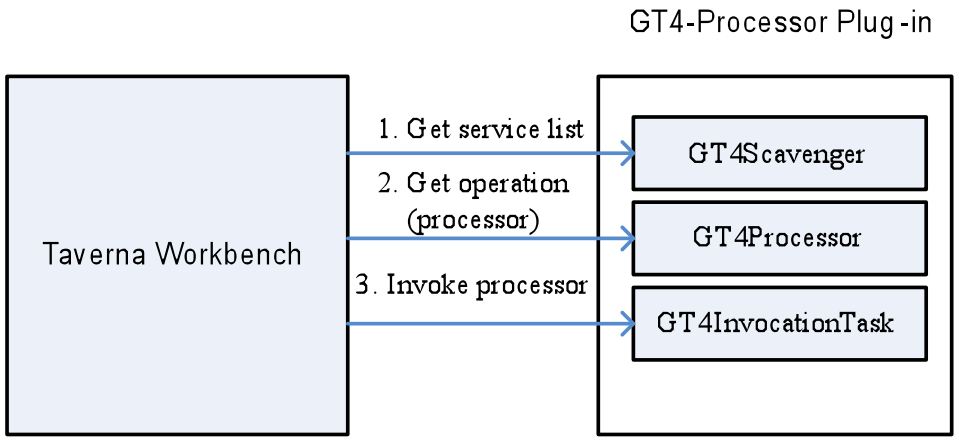
Taverna's Features

- Explicit modeling of data flow.
- Implicit iteration.
- Input/Output metadata. Taverna provides a plug-in called Feta, a semantic discovery tool, to support service discovery using the function description as well as input and output metadata, semantically described using the myGrid ontology. This metadata mechanism provides a feasible integration point with caGrid metadata and discovery services.

Taverna's Features (Cont..)

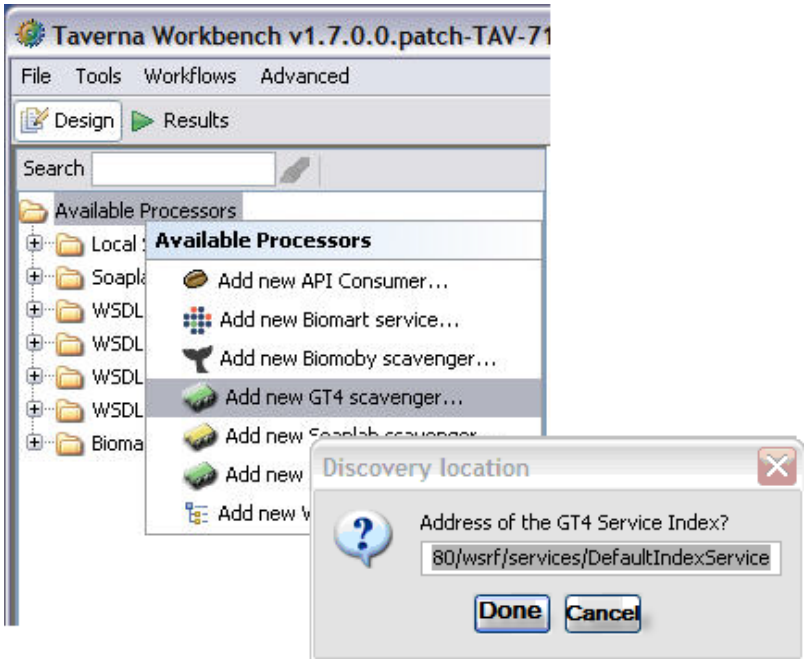
- Beanshell scripting and XML processing support. A custom processor that executes a beanshell (a flavor of dynamic Java) script can be defined inside a Taverna workflow. Beanshell scripts can be shared with others by wrapping them in a workflow and publishing the workflow on the Web. Taverna also provides a set of local processors supporting the processing of XML. By this means Taverna can leverage the processing power of Java and XML.
- Taverna is a nice tool which is easy to use for the non-IT specialists, and the Taverna workbench provides both build-time and run-time support for workflow modeling, debugging, tracing and provenance. This feature of Taverna is an obvious advantage over other solutions like BPEL, DAGMan, etc, because these languages and tools are designed for IT specialists and the domain users in caGrid have difficulties in using them. http://www.beanshell.org/

GT4 Plug-in for Taverna

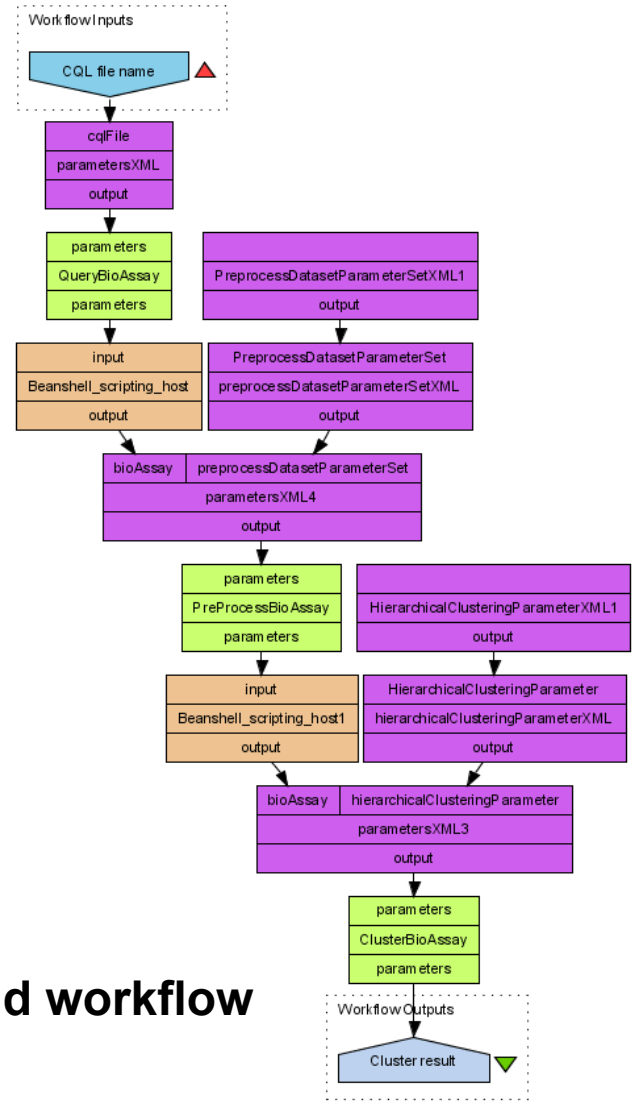


- ***Processor:*** **The concept of *processor* in a Taverna workflow is the same as the concept of task in a common workflow.**
- ***ProcessorTaskWorker:*** **Each *processor* interface corresponds to a *ProcessorTaskWorker* interface. *ProcessorTaskWorker* interface defines the action that is to be performed at runtime when workflow execution reaches to that processor.**
- ***Scavenger:*** **In Taverna, a *scavenger* is a logic collection of a set of processors. For example, a WSDL scavenger collects all the processors each of which corresponds to an operation defined inside that WSDL.**

GT4 Plug-in for Taverna (cont'd)

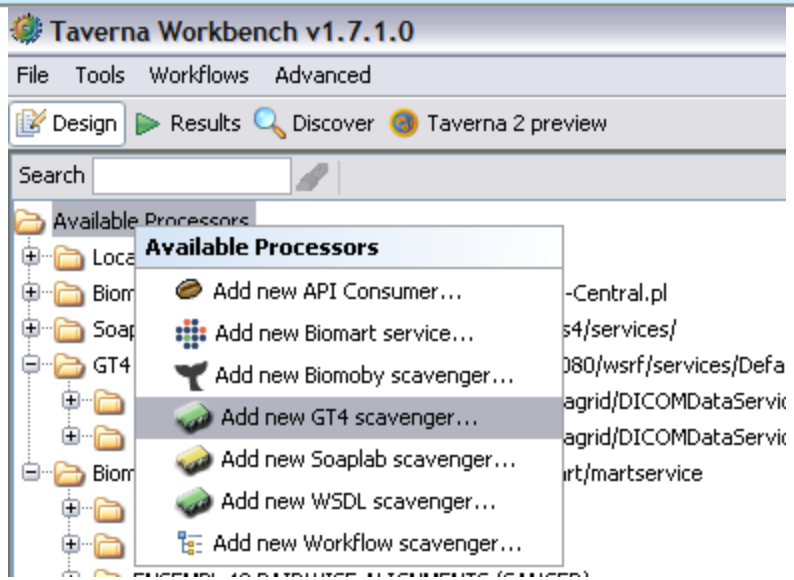


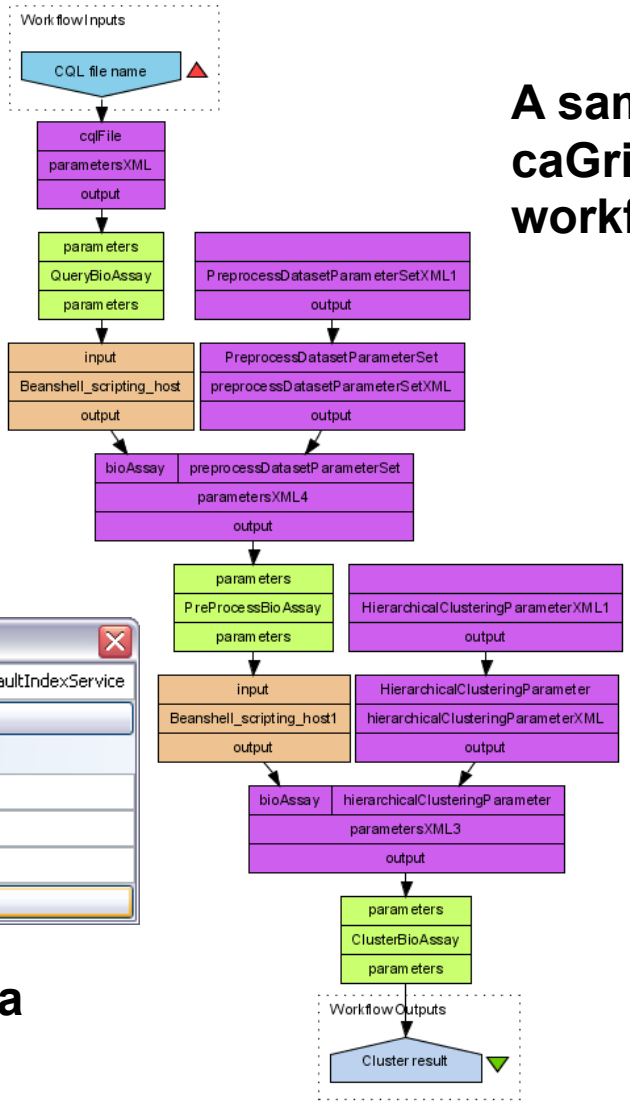
GT4 Scavenger



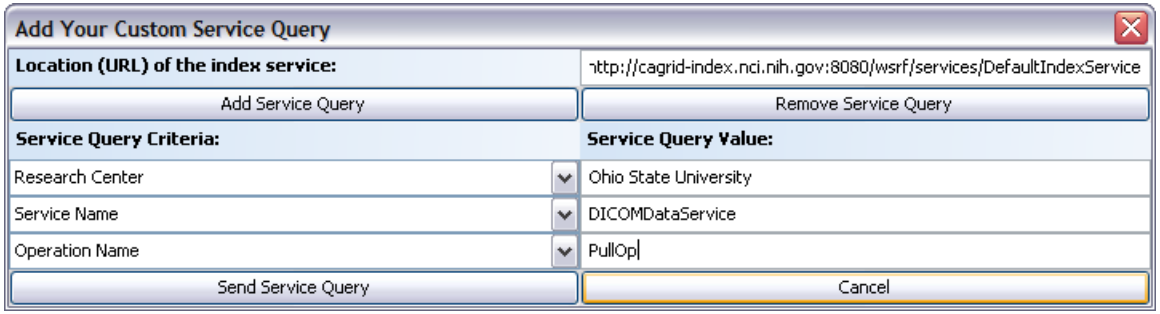
A sample caGrid workflow

GT4 Plug-in for Taverna (cont'd)



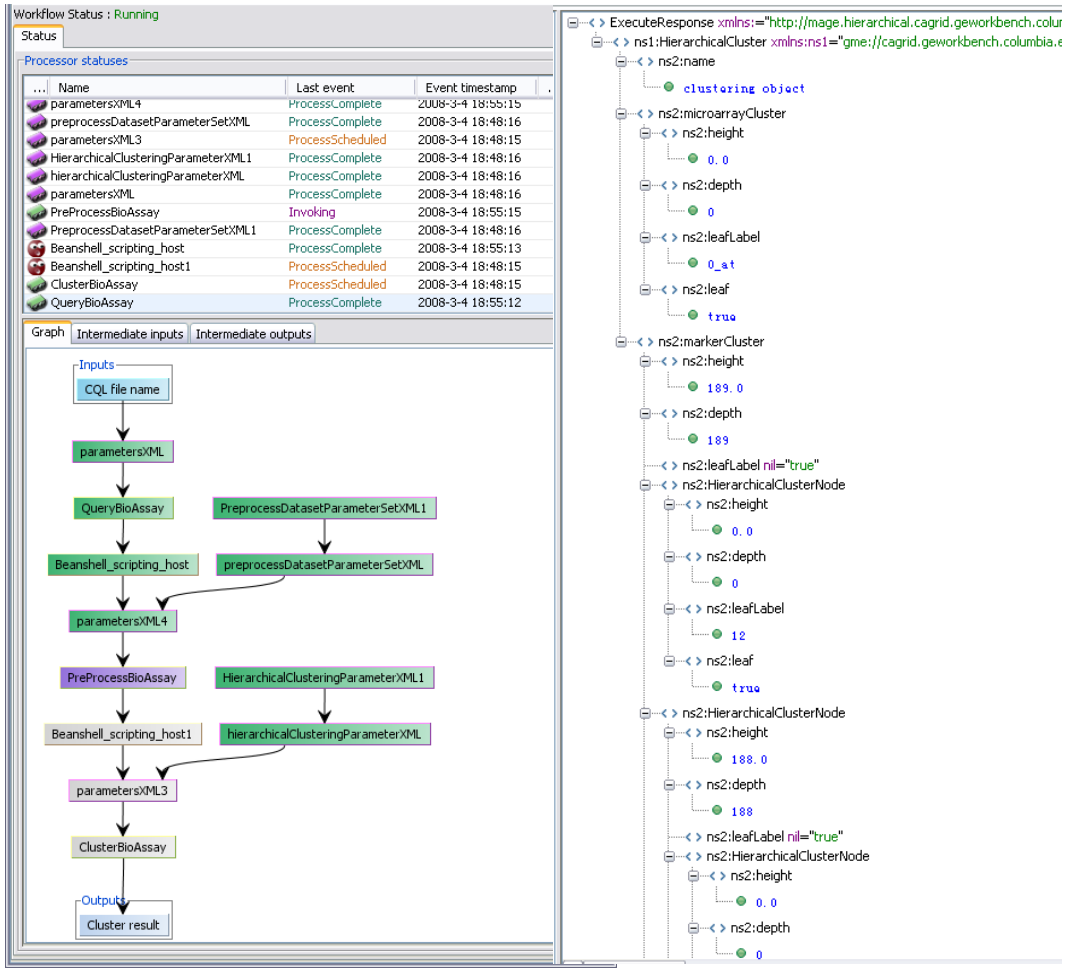


A sample caGrid workflow



GT4 Scavenger with semantic/metadata based caGrid service query

GT4 Plug-in for Taverna (cont'd)



Execution trace of the sample caGrid workflow

caBIG cancer Biomedical Informatics Grid™

Lessons learned

- **Taverna has nice features in modeling data-intensive flows over other languages (like BPEL):**
  - Data do not need to be explicit defined.
  - Support for implicit iteration.
  - A easy-to-extend invocation framework.
  - A nice workbench integrates modeling/ execution tooling.

Issues Identified from Taverna Integration – Potential Solutions

1. **The stability of services. When trying to create a workflow with the services listed, there were some stability issues with the services.** Services could "post" test data to their cvs location so others can, at the very least invoke the service with this test data.
2. **Service Specification.** Services should publish the boundaries of the inputs that can be accepted. For example, a service that can run jobs on a Teragrid resource on the back end is far more desirable to run large datasets than a "regular" caGrid service.
3. **Reusing workflows.** Once a workflow has been created and there is community value in reusing it, the workflow can be published. This will require metadata on the workflows.

Next Steps

- Integrating WS-RF Resource Pattern in service invocation
- Integrating Grid Security with Taverna
- Semantic discovery of services
- Workflow Builder based on metadata (more ambitious !)
- User acceptance.. We are going to try ☺
- Remote Execution of the workflows

Discussion

- Q & A
- Contact Ravi K Madduri at madduri@mcs.anl.gov