

# 1 Semantic Service Search with Feta Engine1.0

---

## 1.1 Introduction

Feta is the suite of components responsible for dealing with **service metadata** in <sup>my</sup>Grid. Feta is composed of two components, namely Feta Client and Feta Engine. Within this document we focus on the latter. We will be providing:

- An overview of the functionality of Feta Engine
- Describe its interface in detail (mainly for the developers of components that need to talk to the engine interface)

Readers interested in information on building and installing Feta Engine should refer to relevant documents delivered within the software distributions or within the **doc** section of the code trunk in the fetaEngine1.0 section of <sup>my</sup>Grid CVS.

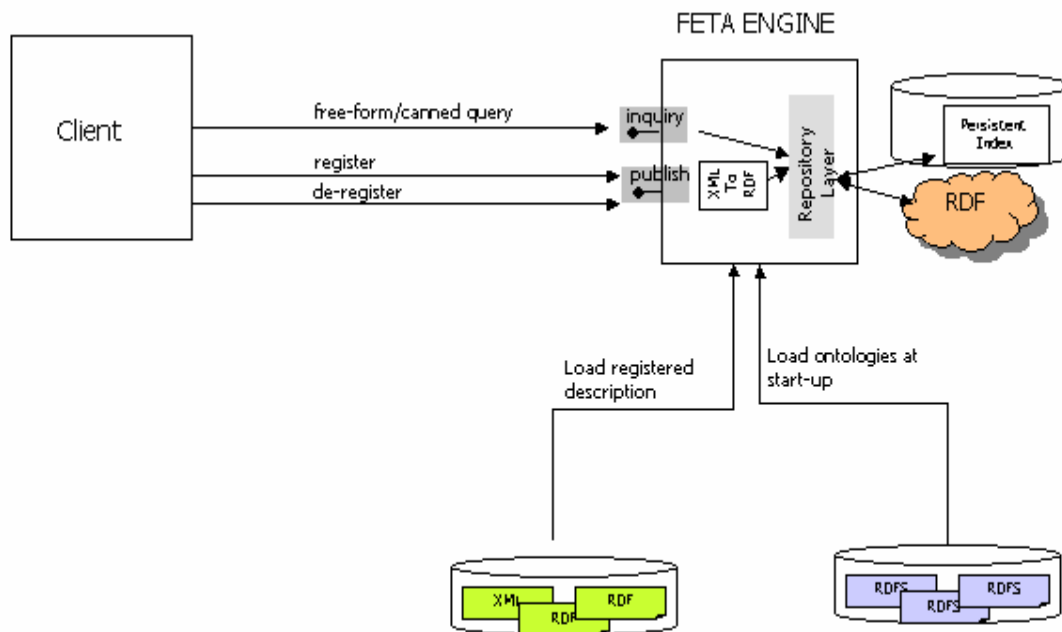
## 1.2 Overview

Feta Engine is a web-service based component that enables search over semantic descriptions of services. A Semantic description is information that complements a low-level description of a service (e.g. a WSDL or a SCUFL file.) and informs the potential service consumers on the domain-specific capabilities of a service.

A pre-requisite to having semantic descriptions is to have a common (agreed-upon) understanding of what a service (capability) is and what is the vocabulary that could be used to talk about capabilities of individual services in the bioinformatics domain. Within myGrid we have achieved these by:

- 1) Jointly developing a **Service Ontology** with the BioMoby project. This ontology outlines the service–schema vocabulary used within semantic service descriptions in the <sup>my</sup>Grid-BioMoby domain. (See Appendix A)
- 2) Developing a bioinformatics **Domain Ontology** that provides the vocabulary that could be used to semantically describe individual services and their capabilities. (See <http://phoebus.cs.man.ac.uk:8100/feta-beta/mygrid/ontology/mygrid-services-lite.rdfs>)

The Feta components are focused on generation aggregation and querying of service metadata wrt the above-described ontologies in <sup>my</sup>Grid. (A sample semantic service description is given in Appendix C.)



**Figure 1.** Feta Engine Architectural Overview.

As mentioned earlier, Feta Engine focuses on the aggregation and querying of service metadata. The architectural overview of this component is given in Figure 1.

- The engine provides two core categories of functionality, namely Publishing and Inquiry.
  - **Publish.** It should be noted that Feta Engine is not the primary source of content for the descriptions published to it. The engine simply holds a copy of we-accessible descriptions, locations of which are reported via the Publish operation. What Feta Engine primarily persists is the list of service description locations that has so far been registered to a Feta Engine.
  - **Inquiry.** Feta Engine allows for search over descriptions via either free form or canned queries over RDF based service metadata.
- Currently Feta Engine supports registration of semantic descriptions in both XML and RDF formats. Existence of the XML based representation is mainly due to technicalities related to the annotation tooling (in myGrid user generate semantic descriptions via the guidance of a data entry tool called Pedro). The support for XML representation is temporary and will probably be removed in future releases (i.e. when we have UI tooling fully operating over RDF).
- To enable semantic search over descriptions Feta Engine needs to be ontology-aware. Ontology-awareness of the engine has been achieved via a configurable start-up ontology loading mechanism as depicted in the Figure1.
- The engine abstracts away from the RDF repository realization by using a Repository Layer Interface. Within the current implementation we provide a Sesame 2 based realization for this layer.

## 1.3 Component Interface Description

Here we would like to provide more info on the semantics and syntax of the operations in the Feta Engine web service interface, the WSDL description of which could be found in Appendix D. Feta Engine could be accessed through the stub classes that come with the fetaEngine.jar. This however, is only useful for Java based clients. In order to make things easier for client developers using other languages (e.g. perl) we will be giving snapshots of SOAP messages that are consumed and produced by Feta Engine.

As previously said, the operations in the main functionality port type are grouped into two:

- The **inquiry** methods are
  - A **Canned Query** method, that accepts search requests conforming to the myGrid service search query schema, which could be found in the web service complex types given in Appendix E. The schema defines the following queries (FindByName, FindByDescription, FindByInput, FindByOutput, FindByTask, FindByMethod, FindByApplication, FindByResource, FindByResourceContent). The results of the Canned Query operation is the conjunct list of signature URL's of service descriptions that match all of the submitted canned queries. A sample canned query request (FindByApplication) search given below.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <FetaSearchRequest xmlns="http://mygrid.org.uk/2004/FETA">
      <fetaAtomicCannedRequest>
        <kindOfQuery>ByApplication</kindOfQuery>
        <paramValue>http://www.mygrid.org.uk/ontology#Basic_Local_Alignme
nt_Search_Tool</paramValue>
      </fetaAtomicCannedRequest>
    </FetaSearchRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

A sample response for this canned request is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <FetaSearchResponse xmlns="http://mygrid.org.uk/2004/FETA">
```

```

    <operationURI>file:///C:/work/code/mygrid/fetaEngine1.0/etc/testData/services_test.xml$alignment::blastx_ncbi$blastx_ncbi</operationURI>
    <operationURI>file:///C:/work/code/mygrid/fetaEngine1.0/etc/testData/serviceDescriptions.rdf$alignment::blastn_ncbi$run</operationURI>
  </FetaSearchResponse>
</soapenv:Body>
</soapenv:Envelope>

```

- A **Freeform Query** method, with which the clients can search over the entire service description repository. We provide this method, as we are aware that there can be cases where canned queries are not enough and the client would like to search over other aspects of a service (e.g. publisher info, service in/out parameter format). This method accepts SeRQL graph queries (with transport type String) executes them over the store and returns the result (with transport type String) to the user. Note that the client is responsible for interpreting the results in the free form query case.

A sample free query request message is given below

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <freeQueryIn xmlns="">
      CONSTRUCT DISTINCT {oper} mg:hasOperationNameText
{operName}
      FROM {serv} mg:hasServiceDescriptionLocation
{descloc},
      {serv} mg:hasServiceNameText {servName},
      {serv} mg:hasOperation {oper} mg:hasOperationNameText
{operName}
      WHERE operName LIKE &quot;*Oimim*&quot;; IGNORE CASE
      USING NAMESPACE
      mg = &lt;http://www.mygrid.org.uk/ontology#&gt;;
      moby = &lt;http://biomoby.org/RESOURCES/MOBY-
S/ServiceDescription#&gt;;
    </freeQueryIn>
  </soapenv:Body>
</soapenv:Envelope>

```

The response for the above request is given below

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope

```

```

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <freeQueryOut xmlns="">
      @prefix rdf: &lt;http://www.w3.org/1999/02/22-rdf-
syntax-ns#&gt; .
      @prefix a: &lt;http://www.w3.org/2000/01/rdf-
schema#&gt; .
      @prefix mg:
&lt;http://www.mygrid.org.uk/ontology#&gt; .
      @prefix moby: &lt;http://biomoby.org/RESOURCES/MOBY-
S/ServiceDescription#&gt; .
      _:node1181nj7t5x26 mg:hasOperationNameText
&quot;getOmimId&quot; .
    </freeQueryOut>
  </soapenv:Body>
</soapenv:Envelope>

```

- The **publish** methods are:
  - **Registration** method that accepts a signature URL for the service to be registered and returns a result designating the outcome of registration, either a Success or a Failure. As we support a hybrid approach, the given Signature URL could be pointing out to an RDF based description or an XML based description. Feta Engine resolves this SignatureURL into the actual document, converts it into an RDF representation (if it is in XML) and stores it in its in-memory RDF model. The descriptions inside the engine are indexed by their signature URLs. The index is a persisted list managed by the engine. If for a reason the Feta Engine web service is re-started, each document listed in the index will be loaded at start-up.

A publish request message is given below.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <publicationRequest xmlns="">
      file:///C:/work/code/mygrid/fetaEngine1.0/etc/testData/services\_test.xml
    </publicationRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

A response message to the above request would look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <FetaPublishResponse xmlns="http://mygrid.org.uk/2004/FETA">
      <publishResult>Success</publishResult>
      <publishMessage>Publish is successful</publishMessage>
    </FetaPublishResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

- The **de-registration** (or delete) method, again, accepts a SignatureURL and removes the description indexed by this URL both from RDF model and from the persistent index. We do not give an example for the de-registration here, as it is very similar to the message interactions of the registration case

## 1.4 To-Dos

In its current form the Feta Engine has no security, every body can make publish and inquiry. Security is currently being worked on in myGrid and once a project-wide security mechanism is devised and developed, it will be incorporated into Feta engine.

Currently all the UI based tooling related to generation of service metadata is being re-developed. The outcomes of this development activity might potentially have an effect on the Feta Engine interface and its query answering mechanisms.

## 1.5 Further Reading

- More information on the contributions of the Feta framework from a from a research perspective could be found here <http://twiki.mygrid.org.uk/twiki/pub/Mygrid/ServiceDiscovery/Feta.pdf>
- The latest news and future plans on Feta could be found on the wiki page <http://twiki.mygrid.org.uk/twiki/bin/view/Mygrid/ServiceDiscovery>