Agent Framework For Intelligent Data Processing

Rahul Ramachandran, Sara Graves, Sunil Movva and Xiang Li

Information Technology and Systems Center
University of Alabama in Huntsville
rramachandran@itsc.uah.edu
(256)-824-5157

UAH
The University of Alabama in Huntsville

Data Processing and Mining

- Terminology:
  - data processing = data preparation
- Observations:
  - Majority (80%) of the scientist's time is spent preparing the data
  - Remainder (20%) of the time is used for actual analysis and mining
  - There are few data preparation/processing steps are commonly used on many different kinds of data especially imagery data
    - Subsetting, Data Format Translation, Subsampling

Can we automate Science Data Processing?

- Yes, but the current methodology involves using:
  - scripts with explicit domain knowledge
  - local data, services
- Adapt to the changing computational environment
  - Web services
    - Semantic Web
  - Grid Services
    - Semantic Grid
- User specifies a goal instead of steps with detailed information required to achieve the goal
  - The different services with proper details are chained together to execute the users task
  - Example:
    - *Subset temperature fields over Florida from TMI dataset for Summer 2002*

Challenges

- *Specifying a query (Natural Language Processing Research)*
- *Optimal service composition (Planning Research*)

✓ **Metadata**

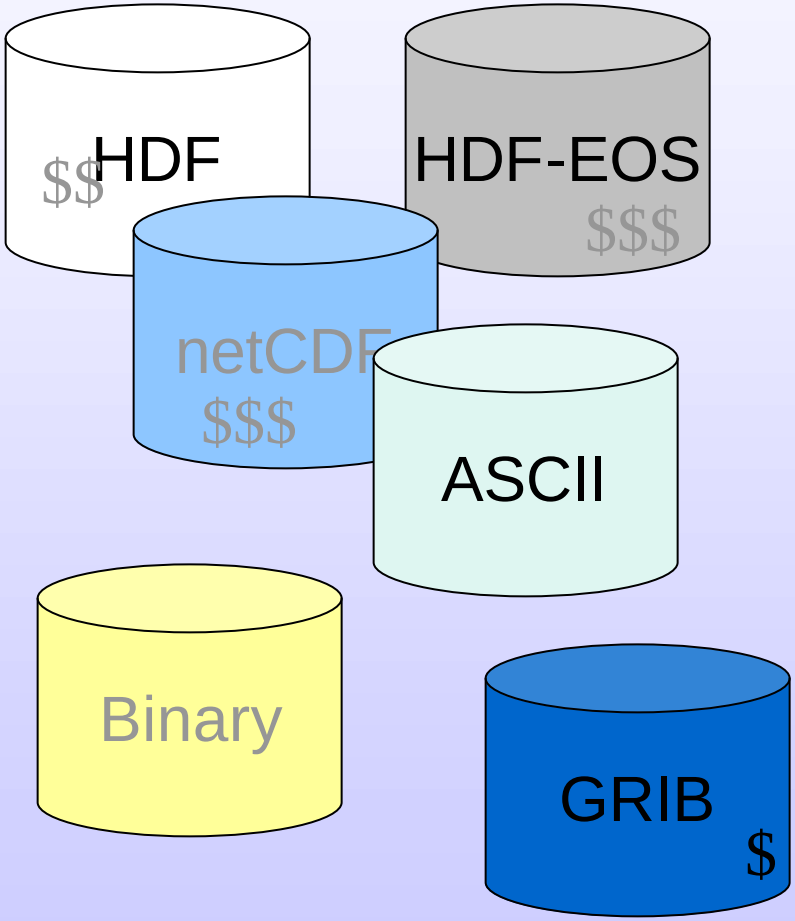
- **Services require a rich set of metadata**
  - **Structural metadata: to provide full description of the data file in bits & bytes, to allow application to read the data**
  - **Semantic metadata: to provide meaning of the data along with a context, to allow application to understand what it has read and how to use it**

✓ **Data Processing Framework**

Project Objectives

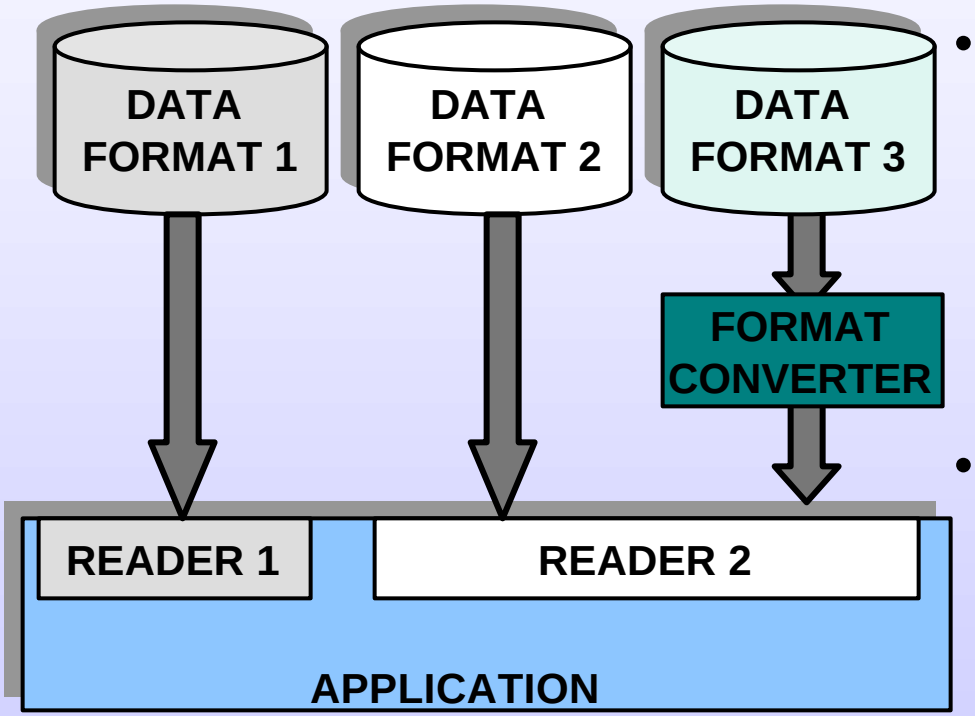
- Design a Metadata Solution
  - Use Earth Science Markup Language (ESML) for structural metadata description
  - Enhance ESML description by providing semantic metadata by leveraging ontologies
- To design and implement a proof of concept Agent Framework (MIDAS) to:
  - leverage the semantic metadata and ontologies for reasoning
  - provide an intelligent, automated processing capabilities for Science data
  - be scalable and adaptable
  - integrate and exploit distributed standalone applications, web services and grid services

Metadata Solution: Background



- **Earth Science Data Characteristics**
  - Different formats, types and structures (18 and counting for Atmospheric Science alone!)
  - Some formats lack metadata where as others are metadata rich ($)
- **Heterogeneity leads to Data usability problem**

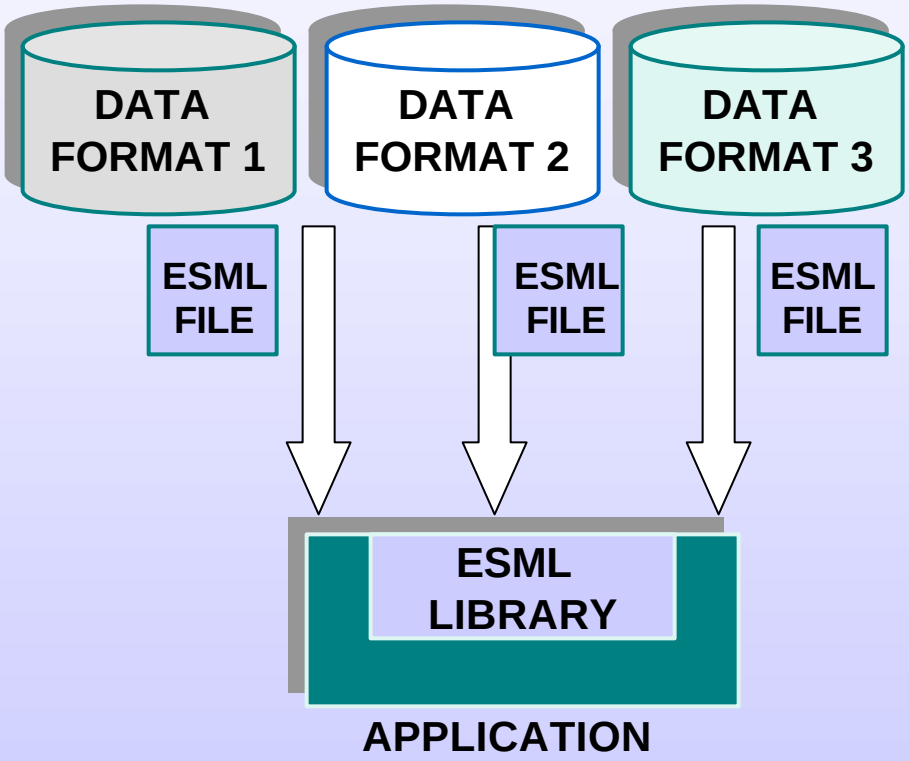
Metadata Solution: Background



Data Usability Problem

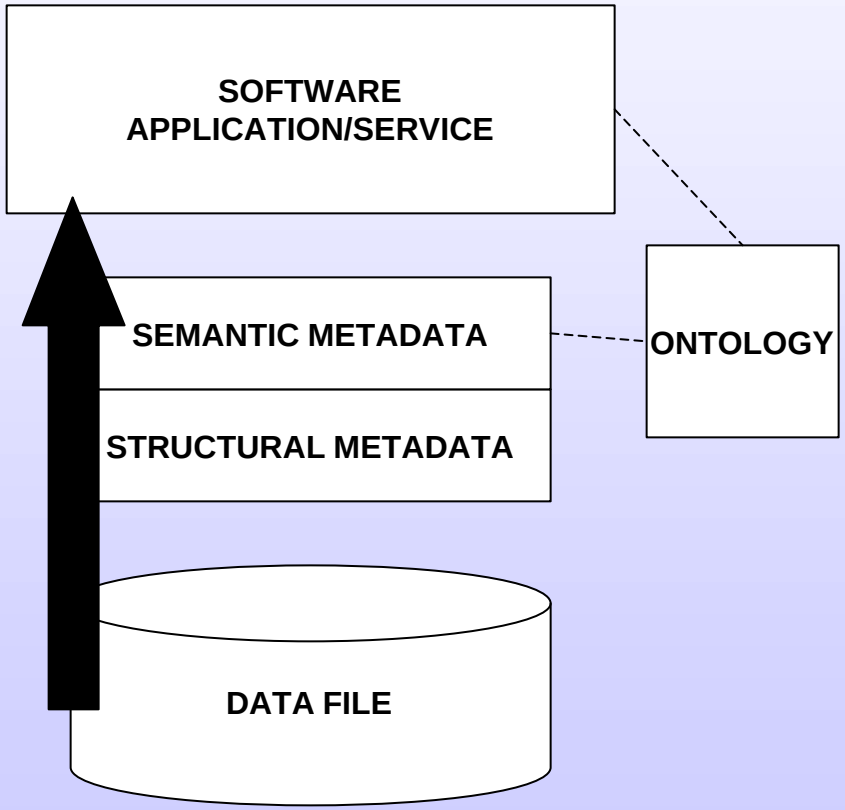
- Requires specialized code for every format
  - Difficult to assimilate new data types
  - Makes applications tightly coupled to data
- One possible solution - enforce a Standard Data Format
  - Not practical for legacy datasets

Metadata Solution:
Earth Science Markup Language (ESML)



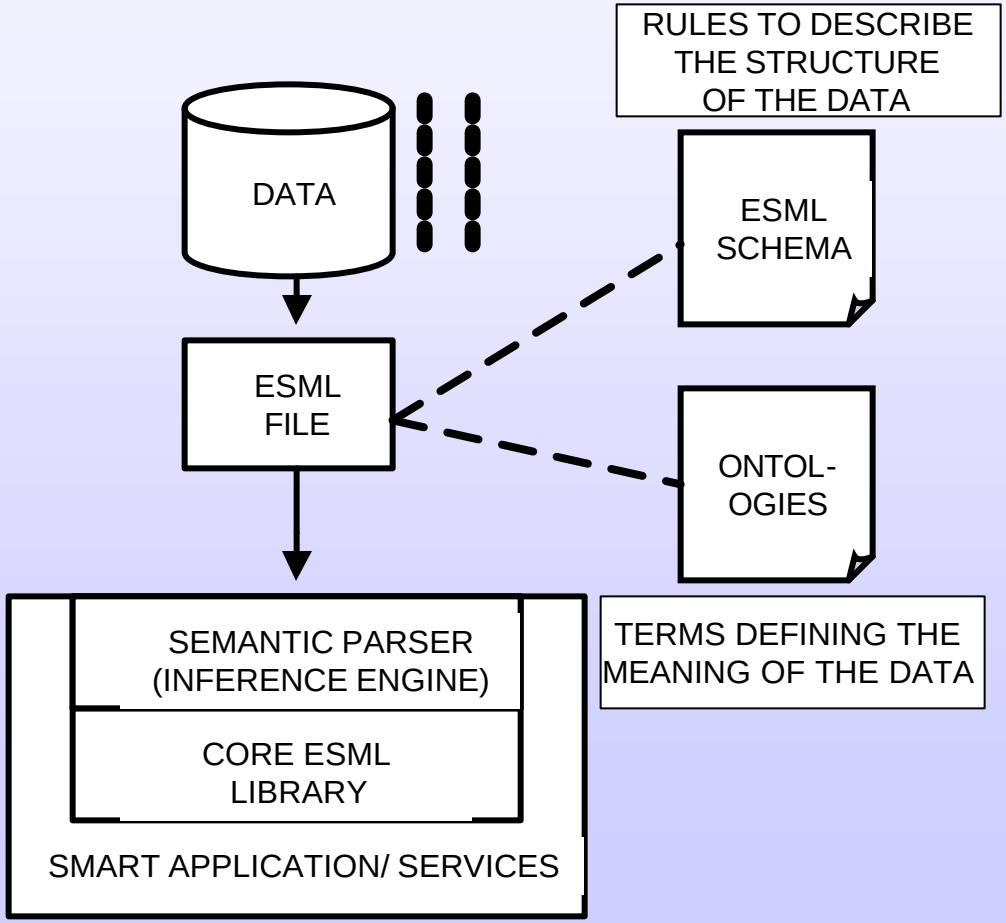
- ESML (external metadata) files containing the structural description of the data format
- Applications utilize these descriptions to figure out how to read the data files resulting in data interoperability for applications

Metadata Solution:
Vertical Metadata Integration via Ontologies



- Horizontal Metadata Integration
  - Mediation services
  - Yellow page services
- Vertical Metadata Integration focuses on semantics for the use of the data by an application
- Both Structural and Semantic metadata are required

Metadata Solution: Extending ESML with Ontologies



- ESML Schema provides structural metadata
- Extend ESML schema by embedding semantic terms in the ESML Description File to provide a complete description of the data
- Allow various science communities can create their own ontologies (for example, SWEET) and use them with ESML Description Files for their data

UAH
The University of Alabama in Huntsville

Metadata Solution: Example of Embedding Semantics in ESML

```
<<a:ESML xmlns:a="ESML" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="ESML.xsd" >
 <SyntacticMetaData>
  <Binary>
   <Structure instances="1" name="SampleSet">
    <Array occurs="100">
     <Array occurs="100">
      <Field name="UWind" type="Int32" order="LittleEndian"/>
     </Array>
    </Array>
    <Array occurs="100">
     <Array occurs="100">
      <Field name="DimX" type="Int32" order="LittleEndian"/>
     </Array>
    </Array>
    <Array occurs="100">
     <Array occurs="100">
      <Field name="DimY" type="Int32" order="LittleEndian"/>
     </Array>
    </Array>
   </Structure>
  </Binary>
 </SyntacticMetaData>
</a:ESML>
```

Original ESML Description File containing only structural metadata

ESML Description File embedded with Semantic Tags defined in separate ontologies

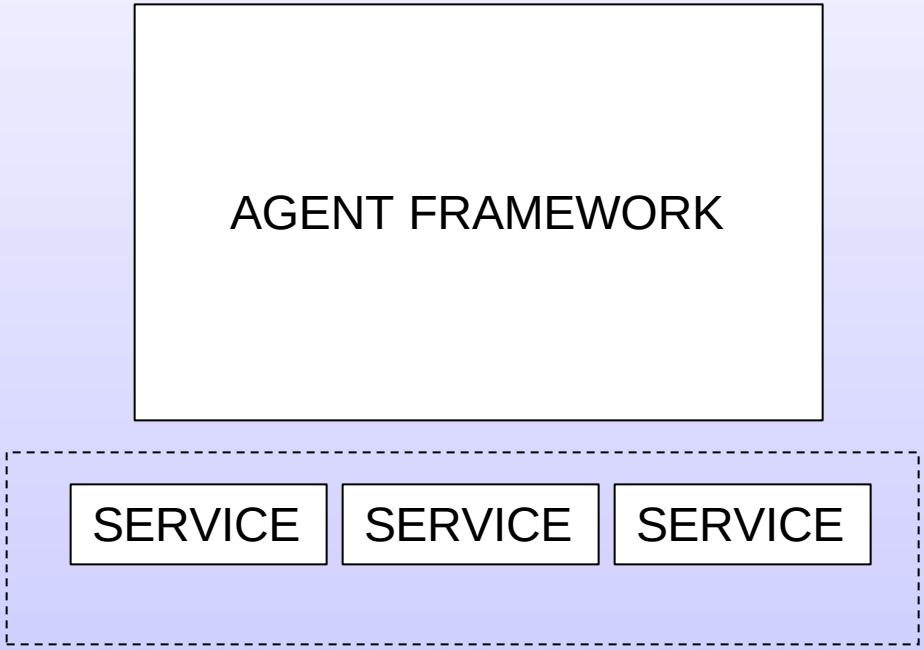
```
<a:ESML xmlns:a="ESML" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="ESML"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns="http://www.itsc.uah.edu/esml-ex#">
   <SemanticMetaData>
        <Latitude rdf:ID="DimX"/>
        <Longitude rdf:ID="DimY"/>
        <DataField rdf:ID="UWind"/>
        <DataSet rdf:ID="SampleSet">
             <hasField rdf:resource="#DimX"/>
             <hasField rdf:resource="#DimY"/>
             <hasField rdf:resource="#UWind"/>
        </DataSet>
   </SemanticMetaData>
   <SyntacticMetaData>
        <Binary>
             <Structure instances="1" name="SampleSet">
             ……................................................
             </Structure>
        </Binary>
   </SyntacticMetaData>
</a:ESML>
```

MIDAS Agent Framework Design: Layer Architecture

- *Infrastructure Layer*: provides the environment that agents can act upon, i.e. services
- *Agent Layer*: contains the agents used to achieve the overall goal of the framework
- *Organization Layer*: defines the organizational structure of the system which is important for agents interaction
- *Coordination Layer*: defines coordination methods required to resolve conflicts and select the next agent
- *Constraint Layer*: verifies whether the system goals are met and interfaces with the users/user interface

MIDAS Agent Framework Design: Infrastructure Layer

- Consists of services domain
  - Search
  - Subsetting
  - Data Format Translation
  - Calibration
  - Navigation
  - Reprojection
  - Visualization
  - Aggregation
  - Fusion
  - Mining



MIDAS Agent Framework Design: Agent Characteristics and Types

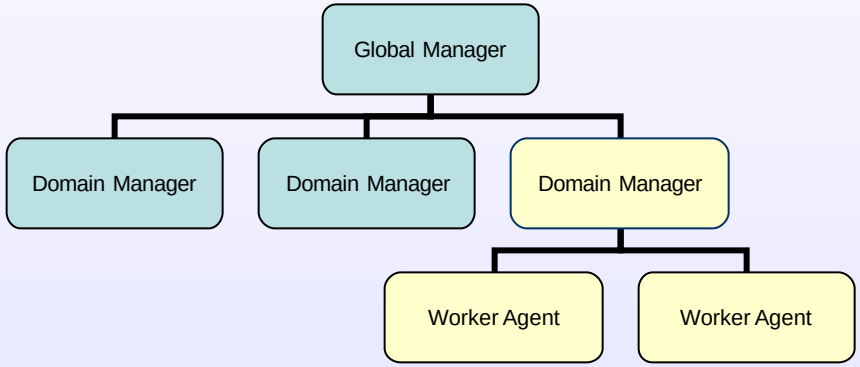
- All the agents have the following characteristics:
  - Role
  - Behavior- methods/functions that it can act upon
  - State – store information of itself and the world it perceives
  - Intelligence/Knowledge– to be able to make decisions via ontologies and a reasoning engine or via a machine learning algorithm or via heuristic algorithm
  - Communication protocol – to interact with other agents
- Types of Agents in the Layer
  - Manager Agents
    - Global Manager
    - Domain Manager
  - Worker Agents

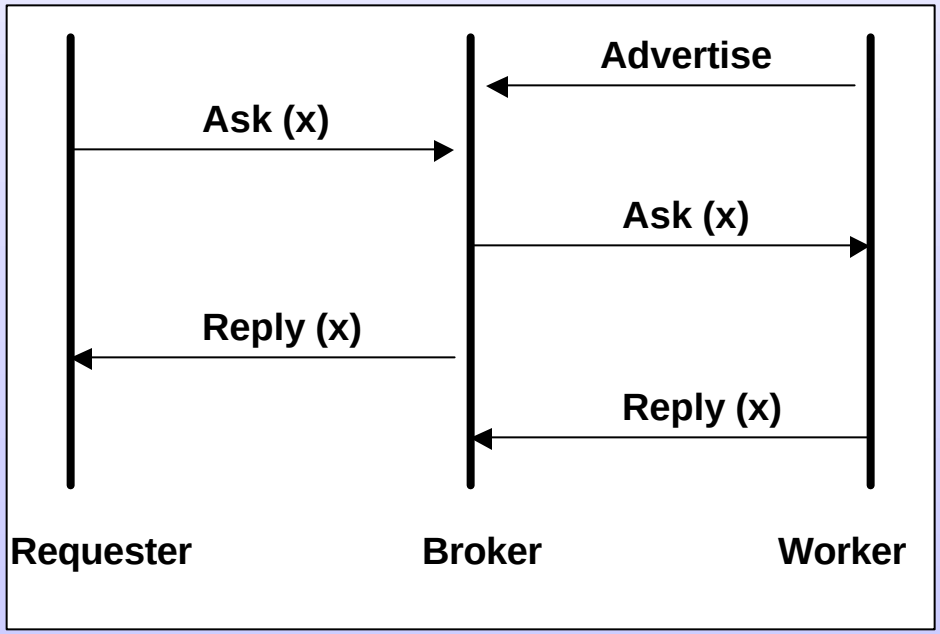
MIDAS Agent Framework Design: Agent Description

- **Global Manager Agent:** Given an user input request, distribute the work and collate results
- **Domain Manager Agent:**
  - Keeps a registry of all the Worker agents in its domain
  - All Worker agents advertise their capabilities to the Manager Agent
  - Parses the incoming message and uses an ontology to find the “correct” Worker agent
  - Polls Worker agents for results
  - Fires and Hires Agents
- **Worker Agent**: Uses ESML semantic metadata and ontologies to **map** input message to API requirements of the Service
  - Example:
    - Navigate the data
    - Map Parameter Concept to Field Name(s)
    - Map Spatial Concept to Bounding Box
    - Map Temporal Concepts to Time Range

UAH
The University of Alabama in Huntsville

MIDAS Agent Framework Design: Organization and Coordination Layer





Organization Layer

- A simple tree structure with a global manager and number of domains is used.
- Each of the domains contains a Manager agent and Worker agents
- Advantage: scalable design that will allow addition of new domains to the overall framework

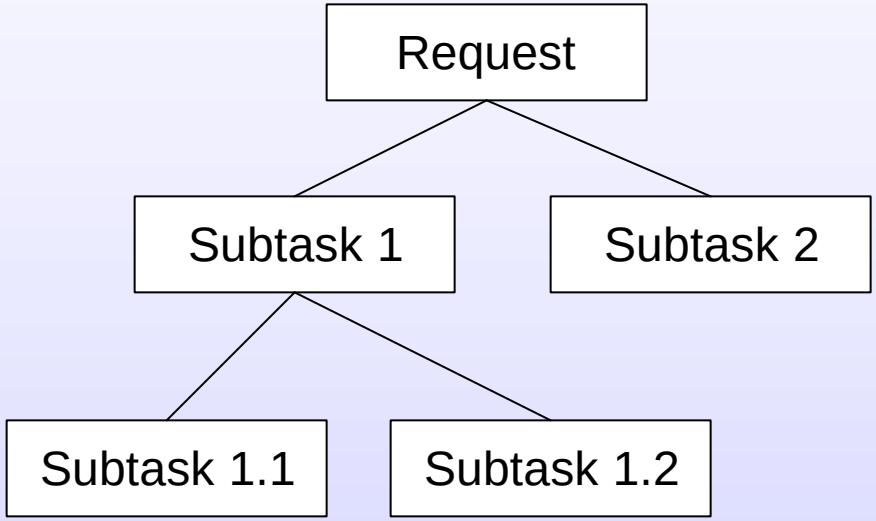
Coordination Layer

- A broker model is used
- Advantage: unlike matchmaker or a contract-net, this model allows the broker to shoulder some responsibility of finding the right agent and returning the result

MIDAS Agent Framework Design: Constraint Layer and Performatives

Constraint Layer

- By using a tree to represents requests, one can check the goal achievement.
- When all the leaf and intermediary nodes are satisfied, resulting in completion of the root node, the task has been accomplished.

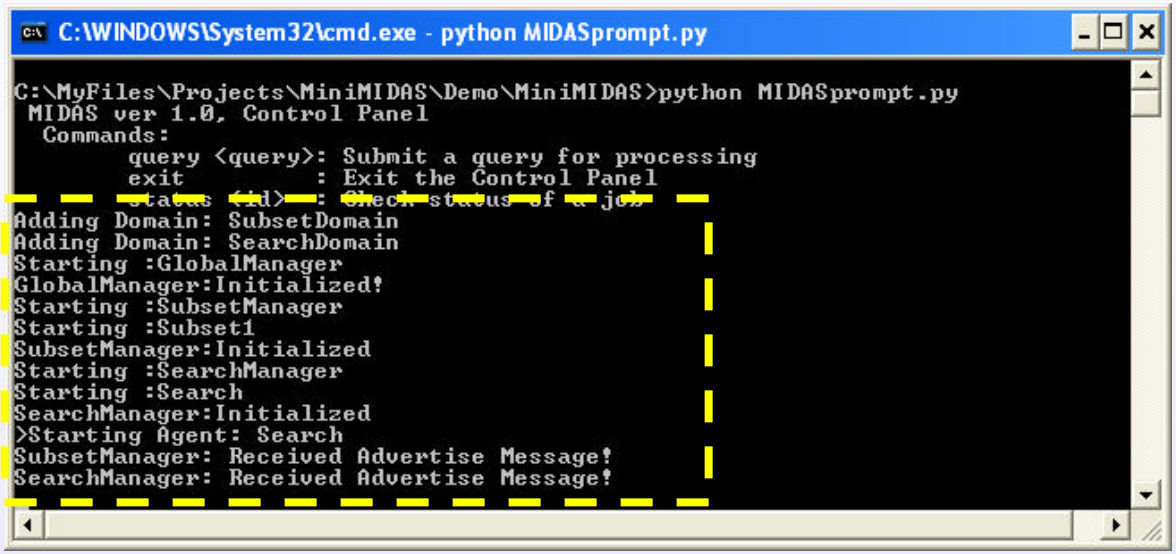


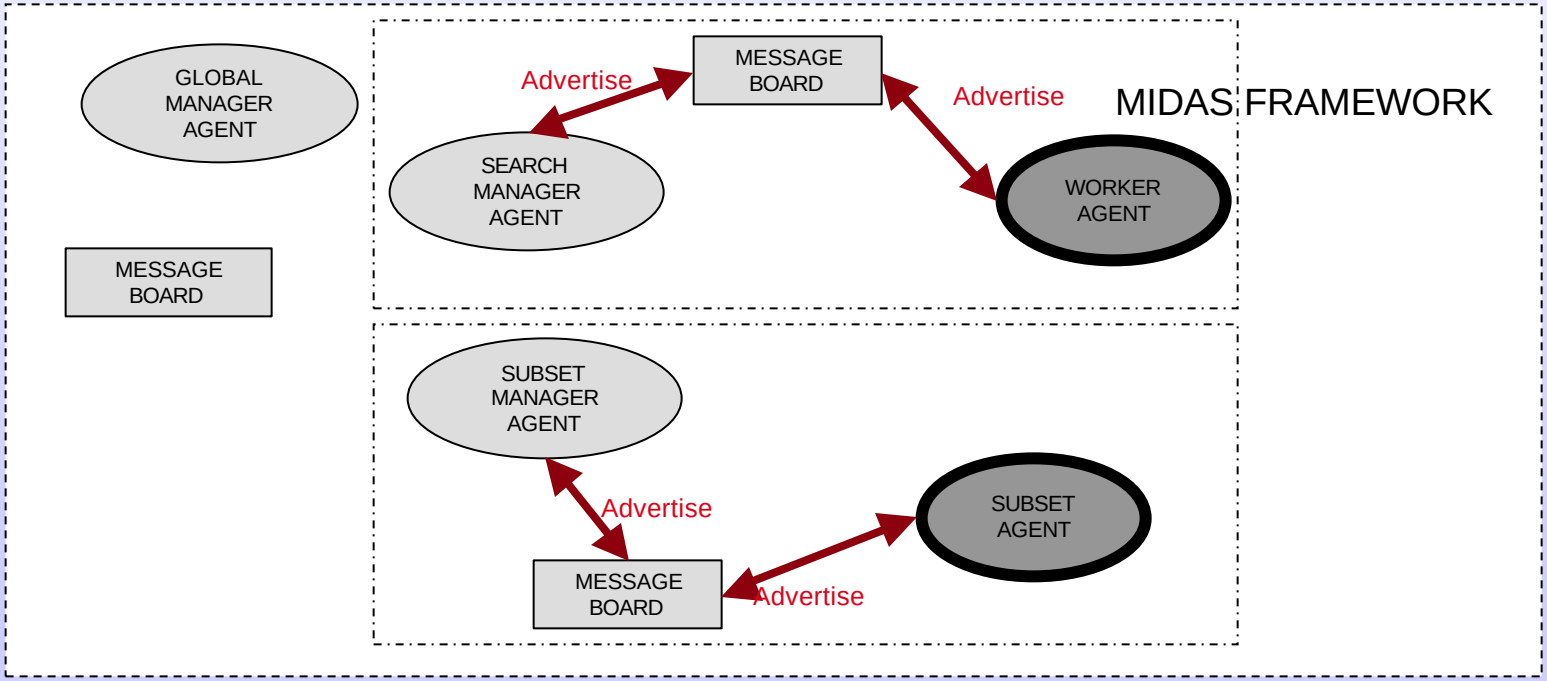
Performatives

- Performatives are the permissible "speech acts" agents use to interact
- Partial set derived from KQML (Knowledge Query and Manipulation Language- UMBC)
- Basic Responses:
  - Error, Sorry
- Query: (Evaluate, AskStatus)
- Capability: (Advertise)

```
advertise
:content <performative>
:language KQML
:ontology <word>
:force <word>
:sender <word>
:receiver <word>
```

MIDAS: Initialization

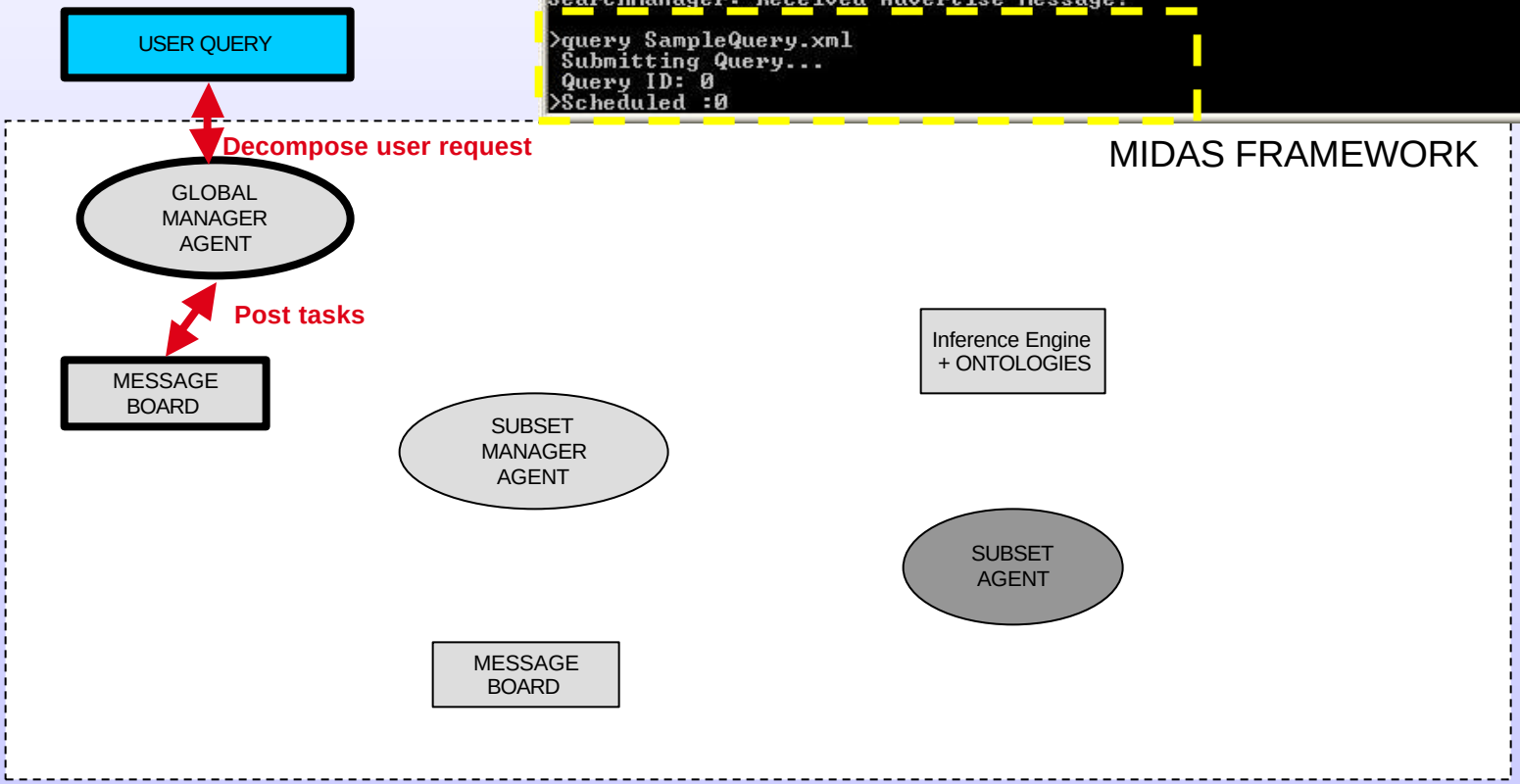




MIDAS : Subset Query

```
C:\WINDOWS\System32\cmd.exe - python MIDASprompt.py
C:\MyFiles\Projects\MiniMIDAS\Demo\MiniMIDAS>python MIDASprompt.py
 MIDAS ver 1.0, Control Panel
  Commands:
        query <query>: Submit a query for processing
        exit         : Exit the Control Panel
        status <id>  : Check status of a job
Adding Domain: SubsetDomain
Adding Domain: SearchDomain
Starting :GlobalManager
GlobalManager:Initialized!
Starting :SubsetManager
Starting :Subset1
SubsetManager:Initialized
Starting :SearchManager
Starting :Search
SearchManager:Initialized
>Starting Agent: Search
SubsetManager: Received Advertise Message!
SearchManager: Received Advertise Message!

>query SampleQuery.xml
 Submitting Query...
 Query ID: 0
>Scheduled :0
```



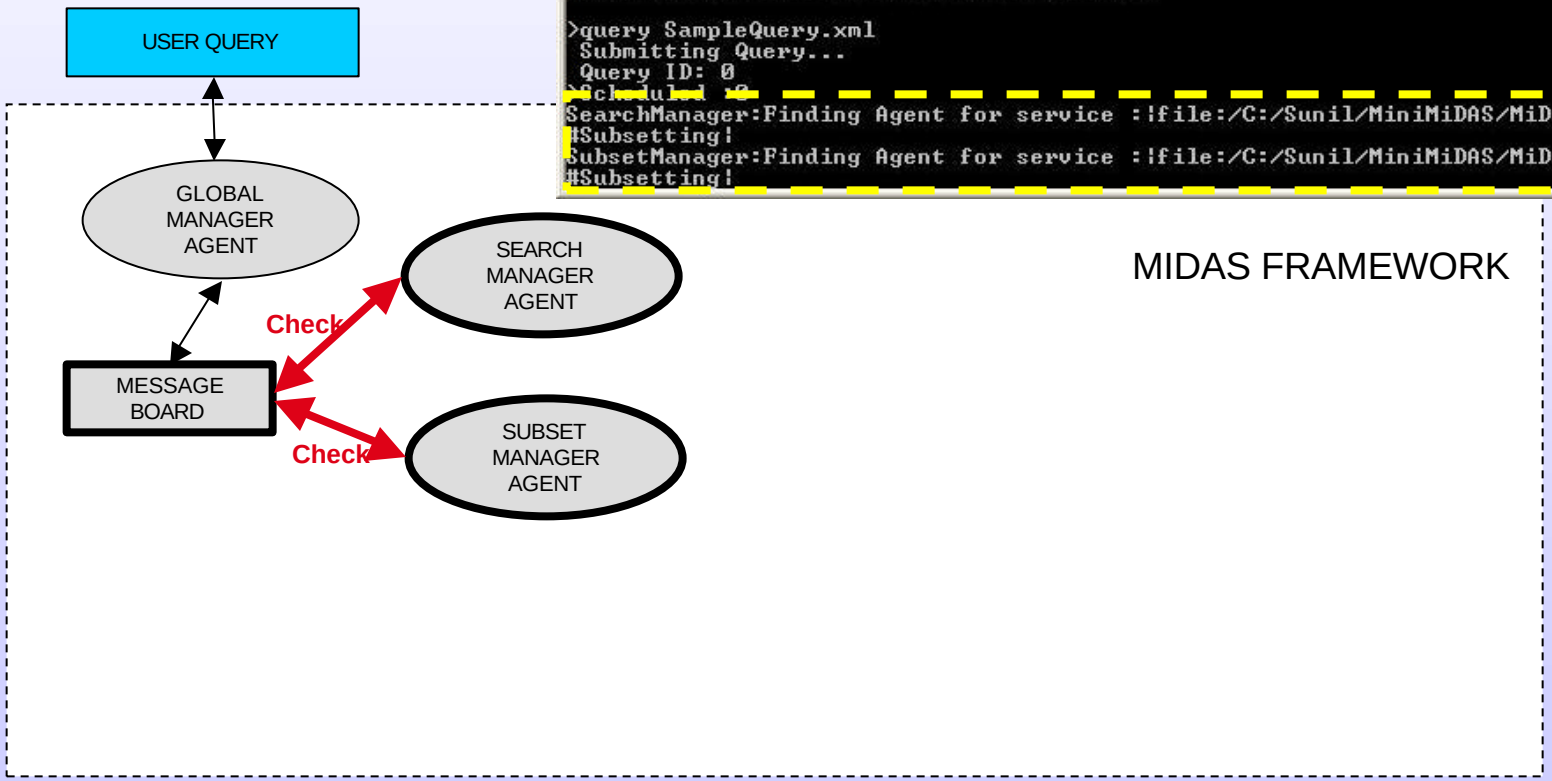
SUBSETTING SERVICE

ESML DESCRIPTION FILE

MIDAS : Subset Query

```
C:\WINDOWS\System32\cmd.exe - python MIDASprompt.py
        exit          : Exit the Control Panel
        status <id>   : Check status of a job
Adding Domain: SubsetDomain
Adding Domain: SearchDomain
Starting :GlobalManager
GlobalManager:Initialized!
Starting :SubsetManager
Starting :Subset1
SubsetManager:Initialized
Starting :SearchManager
Starting :Search
SearchManager:Initialized
>Starting Agent: Search
SubsetManager: Received Advertise Message!
SearchManager: Received Advertise Message!

>query SampleQuery.xml
 Submitting Query...
 Query ID: 0
>Scheduled 10
SearchManager:Finding Agent for service :|file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting|
SubsetManager:Finding Agent for service :|file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting|
```



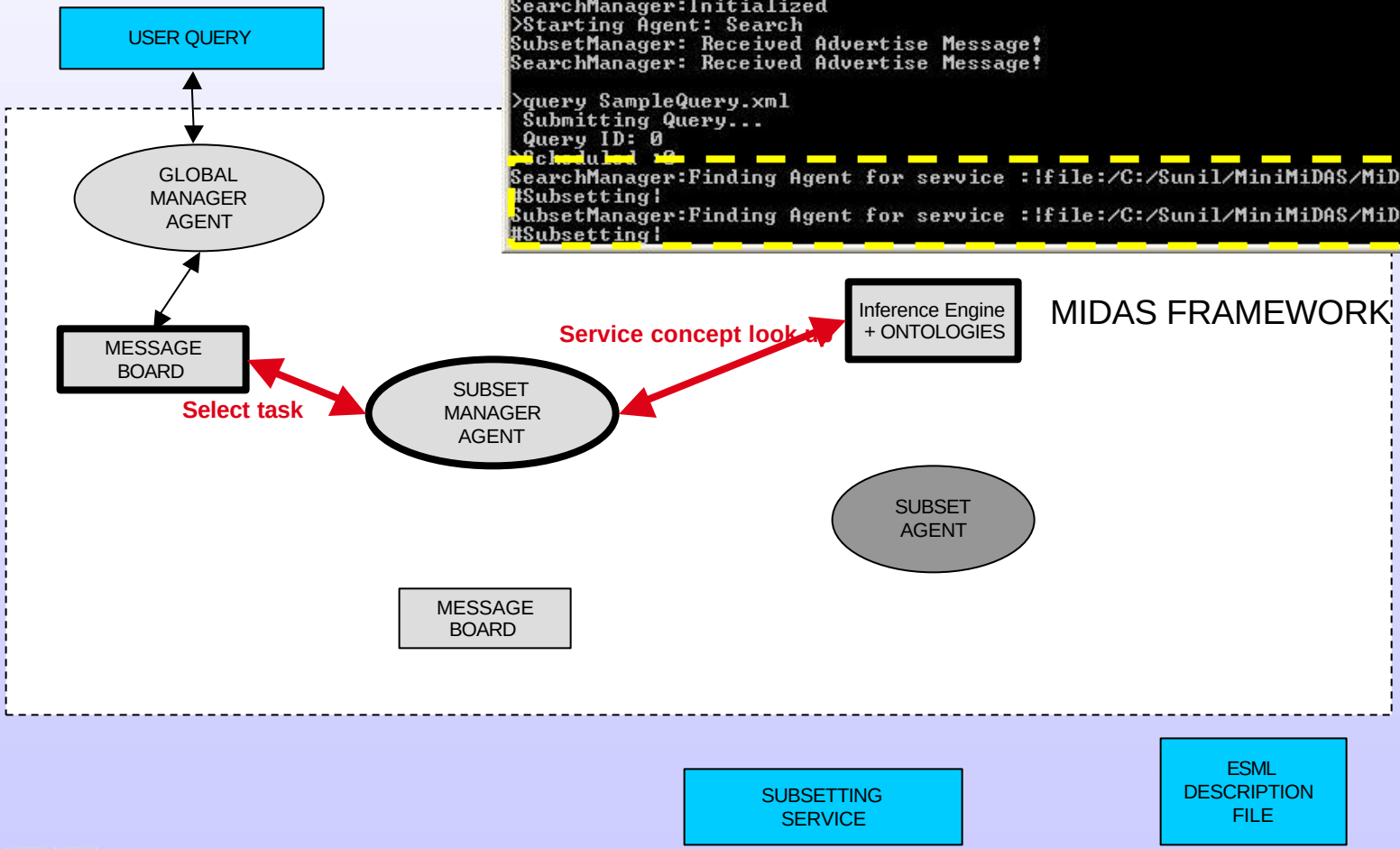
SUBSETTING SERVICE

ESML DESCRIPTION FILE

MIDAS: Subset Query

```
C:\WINDOWS\System32\cmd.exe - python MIDASprompt.py
        exit          : Exit the Control Panel
        status <id>   : Check status of a job
Adding Domain: SubsetDomain
Adding Domain: SearchDomain
Starting :GlobalManager
GlobalManager:Initialized!
Starting :SubsetManager
Starting :Subset1
SubsetManager:Initialized
Starting :SearchManager
Starting :Search
SearchManager:Initialized
>Starting Agent: Search
SubsetManager: Received Advertise Message!
SearchManager: Received Advertise Message!

>query SampleQuery.xml
 Submitting Query...
 Query ID: 0
>Scheduled 0
SearchManager:Finding Agent for service :|file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting|
SubsetManager:Finding Agent for service :|file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting|
```

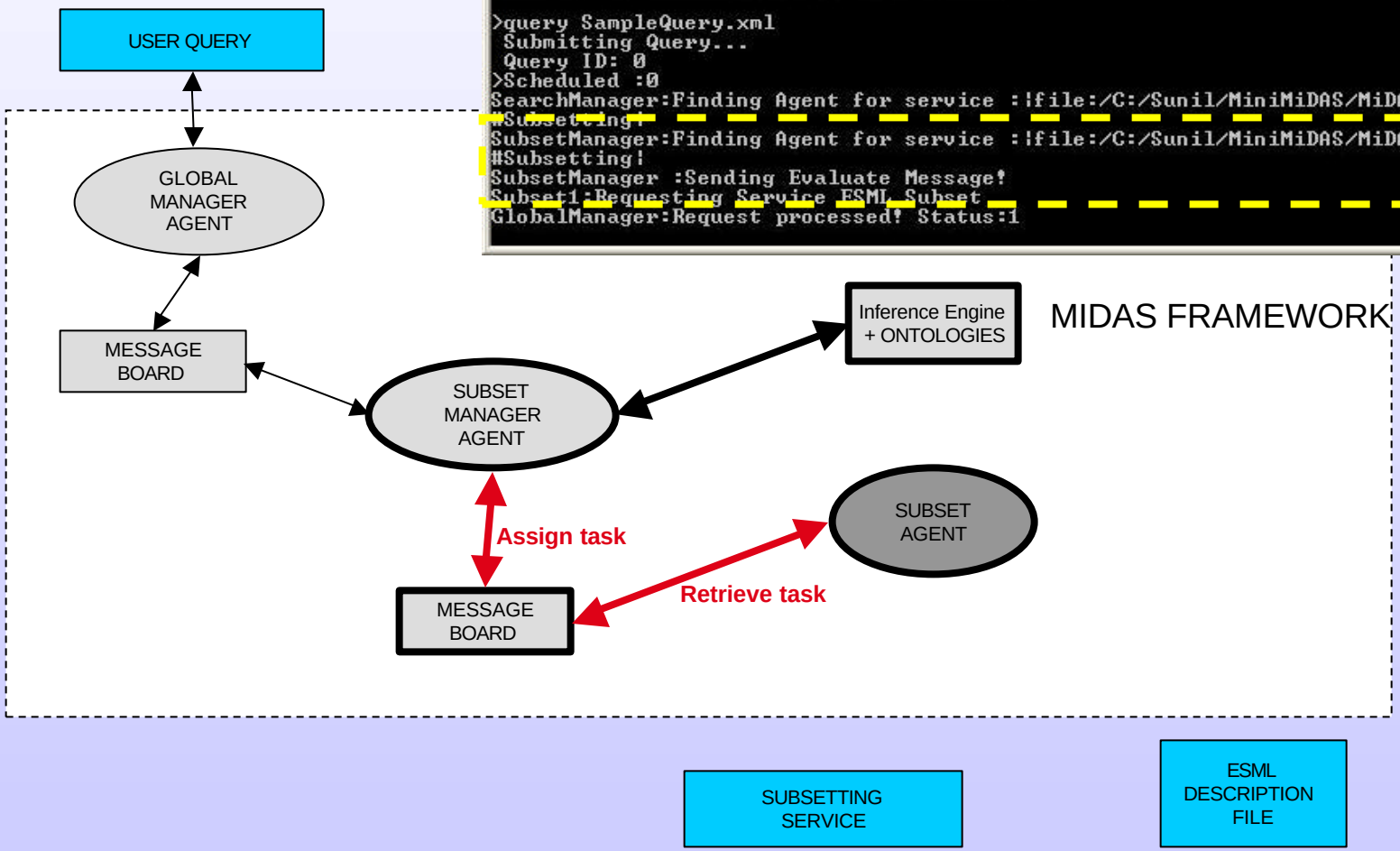


UAH
The University of Alabama in Huntsville

MIDAS: Subset Query

```
C:\WINDOWS\System32\cmd.exe - python MIDASprompt.py
Starting :GlobalManager
GlobalManager:Initialized!
Starting :SubsetManager
Starting :Subset1
SubsetManager:Initialized
Starting :SearchManager
Starting :Search
SearchManager:Initialized
>Starting Agent: Search
SubsetManager: Received Advertise Message!
SearchManager: Received Advertise Message!

>query SampleQuery.xml
 Submitting Query...
 Query ID: 0
>Scheduled :0
SearchManager:Finding Agent for service :!file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting!
SubsetManager:Finding Agent for service :!file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting!
SubsetManager :Sending Evaluate Message!
Subset1:Requesting Service ESML Subset
GlobalManager:Request processed! Status:1
```

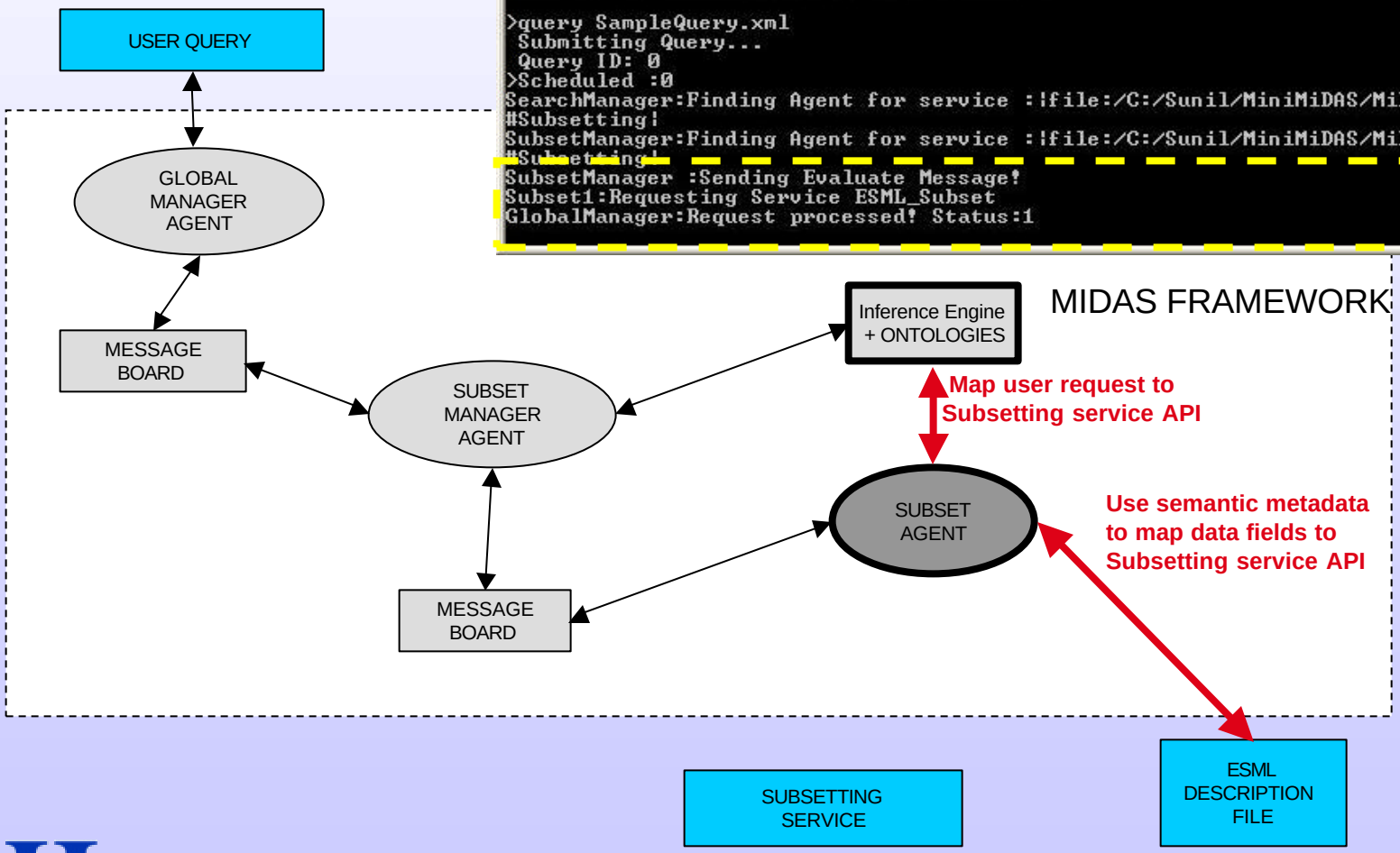


UAH
The University of Alabama in Huntsville

MIDAS: Subset Query

```
C:\WINDOWS\System32\cmd.exe - python MIDASprompt.py
Starting :GlobalManager
GlobalManager:Initialized!
Starting :SubsetManager
Starting :Subset1
SubsetManager:Initialized
Starting :SearchManager
Starting :Search
SearchManager:Initialized
>Starting Agent: Search
SubsetManager: Received Advertise Message!
SearchManager: Received Advertise Message!

>query SampleQuery.xml
 Submitting Query...
 Query ID: 0
>Scheduled :0
SearchManager:Finding Agent for service :!file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting!
SubsetManager:Finding Agent for service :!file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting!
SubsetManager :Sending Evaluate Message!
Subset1:Requesting Service ESML_Subset
GlobalManager:Request processed! Status:1
```

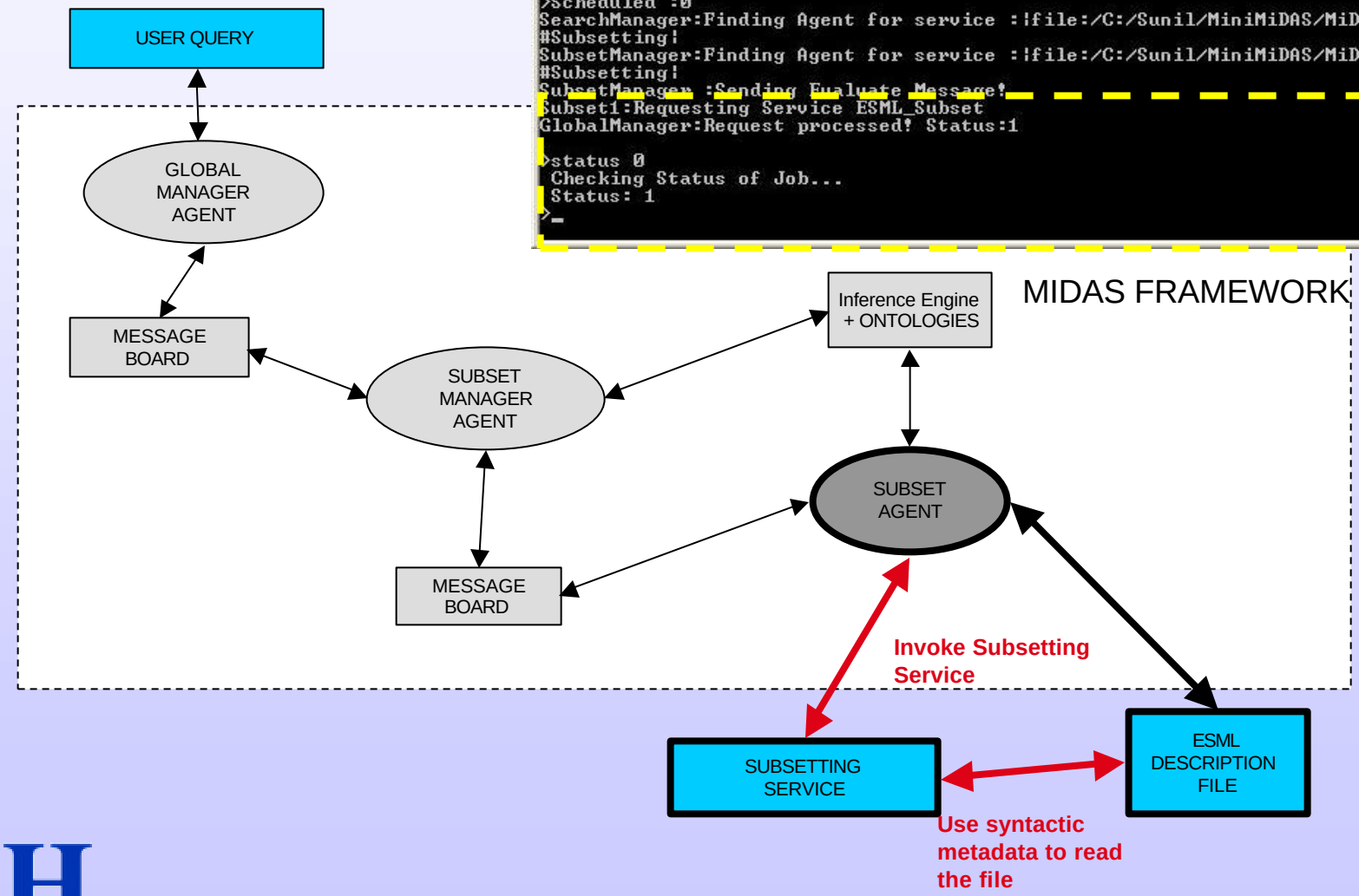


MIDAS: Subset Query

```
C:\WINDOWS\System32\cmd.exe - python MIDASprompt.py
Starting :SearchManager
Starting :Search
SearchManager:Initialized
>Starting Agent: Search
SubsetManager: Received Advertise Message!
SearchManager: Received Advertise Message!

>query SampleQuery.xml
 Submitting Query...
 Query ID: 0
>Scheduled :0
SearchManager:Finding Agent for service :!file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting!
SubsetManager:Finding Agent for service :!file:/C:/Sunil/MiniMiDAS/MiDASOntology
#Subsetting!
SubsetManager :Sending Evaluate Message!
Subset1:Requesting Service ESML_Subset
GlobalManager:Request processed! Status:1

>status 0
 Checking Status of Job...
 Status: 1
>_
```



UAH
The University of Alabama in Huntsville

MIDAS Summary

- Contains three data preparation/processing domains with services for:
  - Search
  - Data Format Translation
    - Any ESML supported format to Binary
  - Subsetting
- Leverages both Semantic and Structural metadata in ESML
  - Unlike existing hardwired services
- Uses Ontologies and inference engines for reasoning
- Can be adapted for both Web and Grid services

Questions?

Additional ESML Information:
http://esml.itsc.uah.edu

Contact Information:
rramachandran@itsc.uah.edu
(256)-824-5157