Execution Management and SLA Enforcement in Akogrimo

Antonios Litke, Kleopatra Konstanteli, Vassiliki Andronikou, Sotirios Chatzis, and Theodora Varvarigou

e-mail: ali@telecom.ntua.gr

Presentation Outline

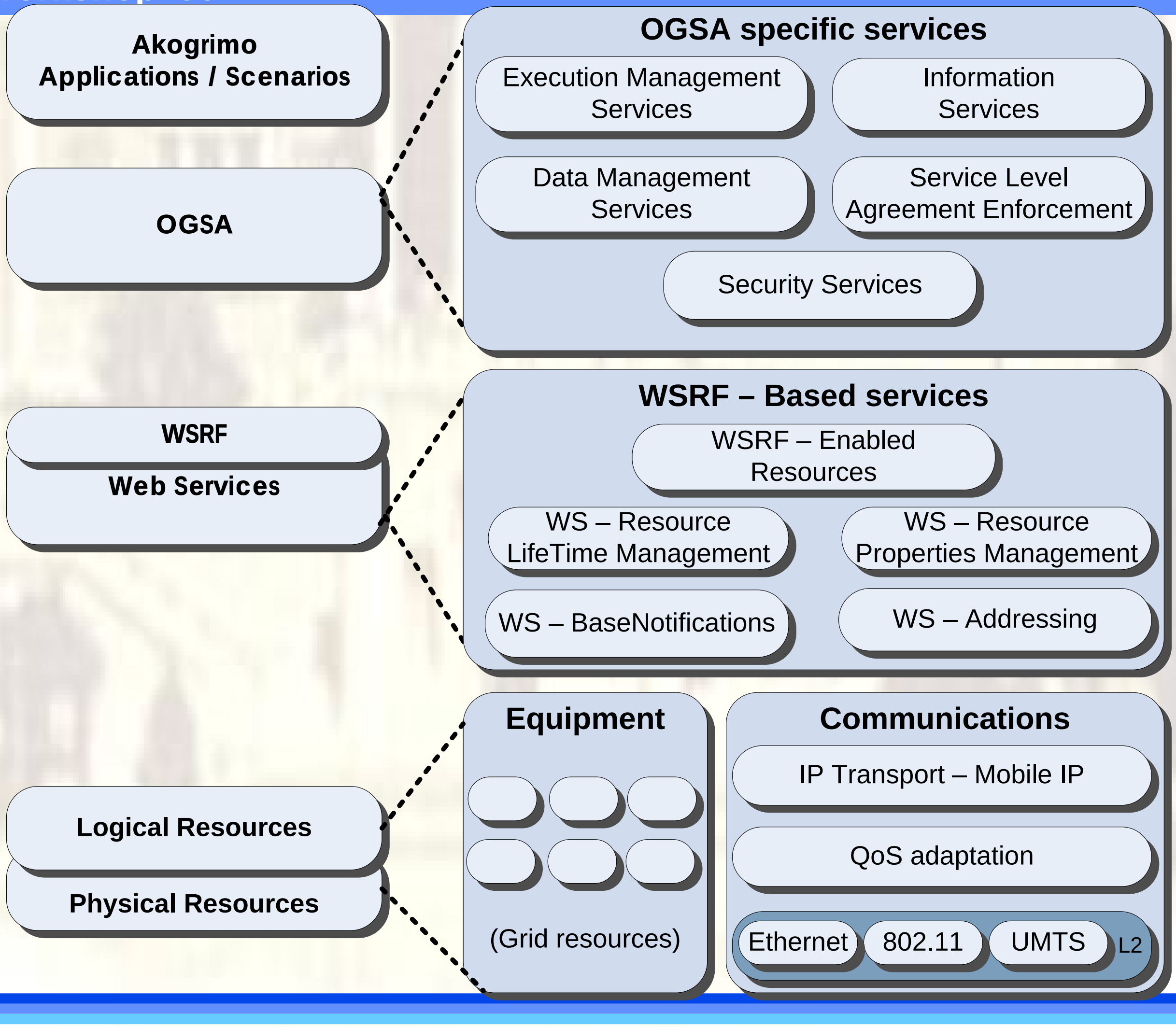
- ➢Architectural overview of Akogrimo
- ➢The need for SLAs
- ➢Component descriptions and interactions
- ➢Implementation issues
- ➢Related work
- ➢Conclusions
- ➢Questions/answers

OGSA-based Layered Architecture

An OGSA compliant architecture aims to

- Manage resources across distributed heterogeneous platforms
- Deliver seamless access control and quality of service
- Provide a common base for autonomic management solutions
- Define open, published interfaces

Akogrimo Architecture



Akogrimo Architecture

•Physical and logical resources layer

- Resources comprise each and every capability of the Grid

  •**Physical resources**, which include servers, storage, and network and Logical resources.

  •**Logical resources** are above the physical and provide additional functionality by virtualizing and aggregating the resources in the physical layer.

  •In Akogrimo communication resources are also incorporated in logical resources

The need for management of SLA

Grids and mobile Grids are dynamic environments subject to unpredictable changes:

- system or network failures, system performance degradation, removal of machines, variations in the cost of resources,

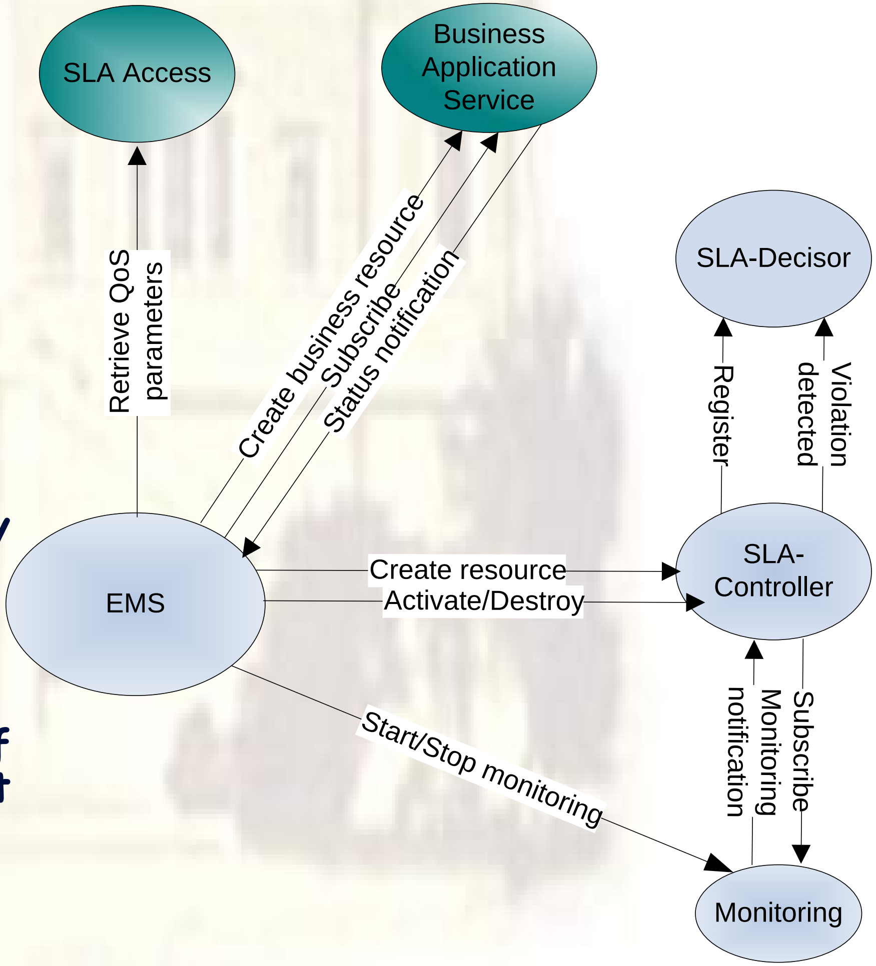
Execution Management components should take care of conformance to the contractual terms of SLAs

- EMS system monitors and manages the execution of the job until its completion
- In case of violations, it takes actions to dynamically rectify them in such a way as to meet the terms defined in the related SLA

Grid Infrastructure Services

- Addresses the performance issues while conforming to the determined Service Level Agreement (SLA).

- Execution Management Services:
  - Instantiating and managing tasks
- SLA Enforcement:
  - SLA contractual terms that especially influence the execution of jobs
- Monitoring:
  - status of the execution, availability of services, and of gathering the relevant information



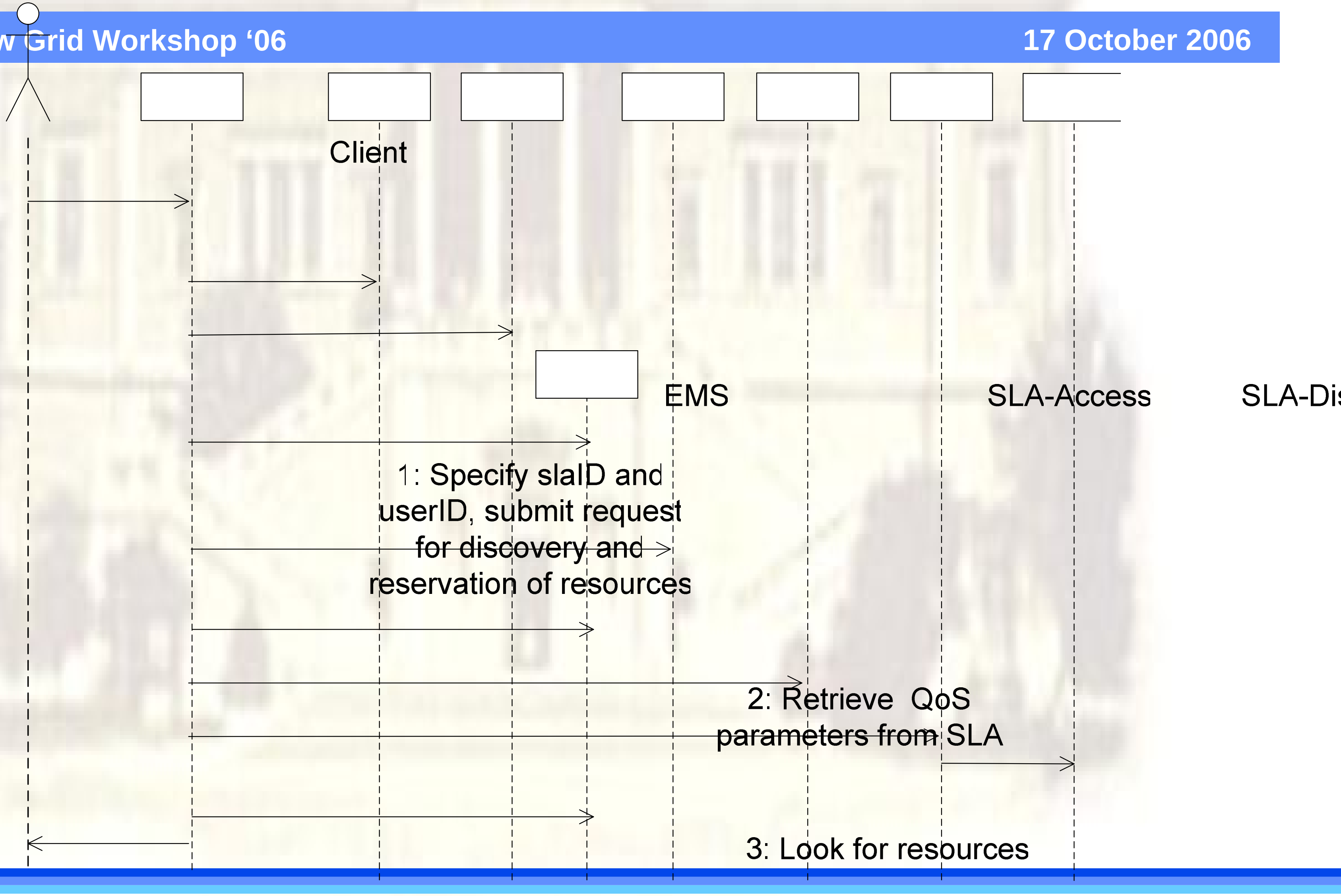
Example of QoS Parameters in SLAs

```
<QoSParams>
        <QoSItems>
                <QoSItem>
                        <param>cpuSpeed</param>
                        <paramValue>3.5</paramValue>
                        <paramType>MHz</paramType>
                        <threshold>40%</threshold>
                </QoSItem>
                <QoSItem>
                        <param>diskSpace</param>
                        <paramValue>1</paramValue>
                        <paramType>GB</paramType>
                        <threshold>20%</threshold>
                </QoSItem>
                <QoSItem>
                        <param>memory</param>
                        <paramValue>1</paramValue>
                        <paramType>GB</paramType>
                        <threshold>50%</threshold>
                </QoSItem>
                <QoSItem>
                        <param>networkBandwidth</param>
                        <paramValue>GOLD</paramValue>
                        <paramType></paramType>
                        <threshold></paramValue>
                </QoSItem>
        </QoSItems>
</QoSParams>
```

Discovery and Reservation phase

Service	Interaction	Interface
SLA-Access	Step 2	*ObjQoS getQoSParameters(String slaID)*
SLA-Discovery	Step 3	*ArrayList findSuitableResources(objQoS objQoSData)*
Business service	Step 5	*EndpointReferenceType create()*
QoS Broker	Step 6	*Boolean setQoS(EndpointReferenceType serviceEPR, integer networkBundleType)*
SLA-Controller	Step 8	*EndpointReferenceType create()*
	Step 9	*Boolean registerToServiceDecisor(EndpointReferenceType slaControllerEPR, EndpointReferenceType serviceEPR)*

Discovery and Reservation MSC



Client

EMS

SLA-Access

SLA-Di

1: Specify slaID and userID, submit request for discovery and reservation of resources

2: Retrieve QoS parameters from SLA

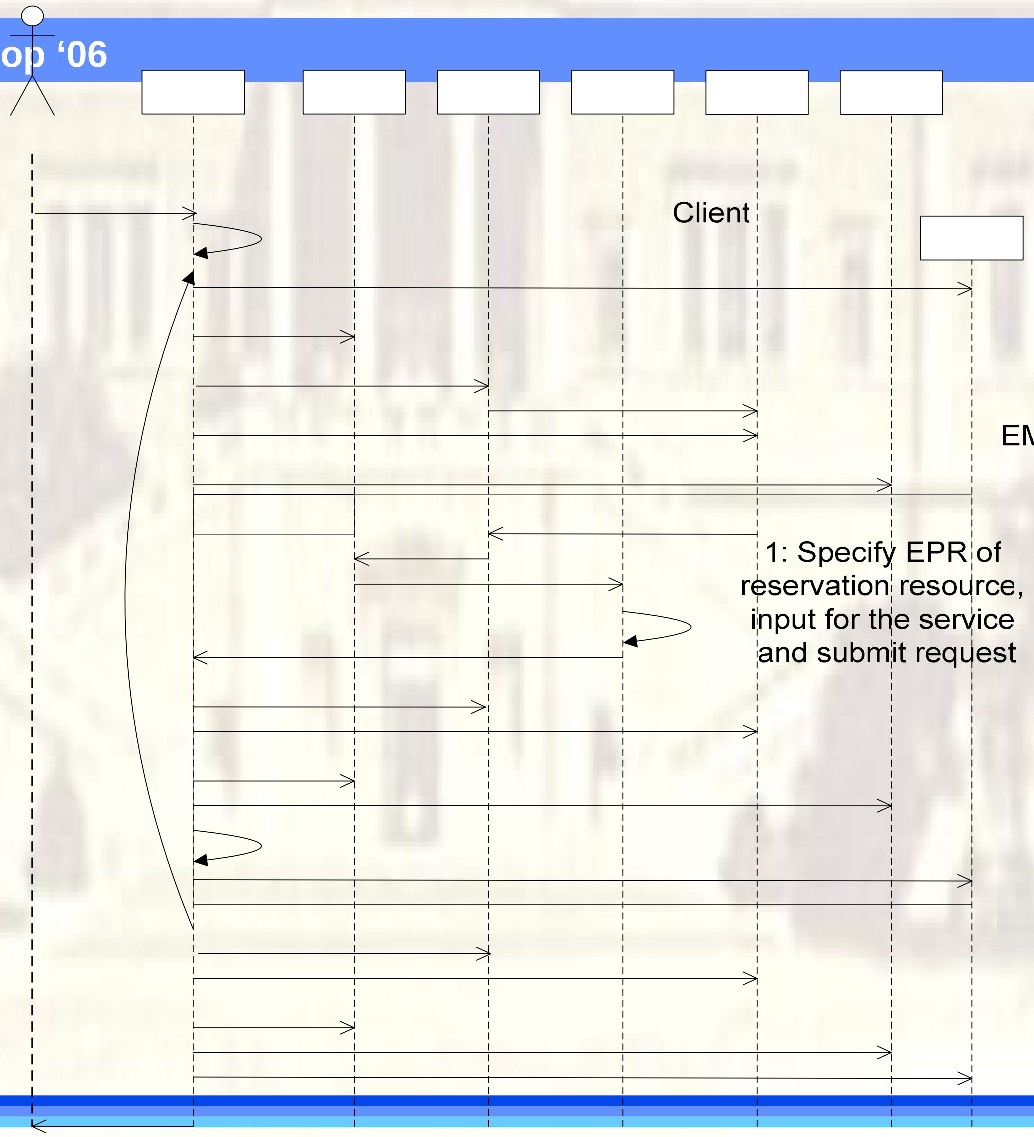
3: Look for resources

4: Match? Create a service reservation

Execution and Monitoring phase

Service	Interaction	Interface
SLA-Controller	Step 4	*Boolean activateServiceController(EndpointReferenceType serviceEPR, objQoS objQoSData, String slaID)*
	Step 11	*Boolean receiveInformationByMonitoring(EndpointReferenceType serviceEPR, objQoS objQoSData)*
	Step 17, 24	*Void destroy()*
SLA-Decisor	Step 12	*Boolean setViolation(EndpointReferenceType serviceEPR, Violation violation)*
Monitoring	Step 5	*Boolean startMonitoring(EndpointReferenceType serviceEPR, objQoS objQoSData, String slaID)*
	Step 14, 21	*Boolean stopMonitoring(EndpointReferenceType serviceEPR)*
Metering	Step 6	*org.oasis.wsn.SubscribeResponse subscribe(org.oasis.wsn.Subscribe subscribeRequest)*
	Step 7	*Boolean startMetering(EndpointReferenceType serviceEPR)*
	Step 16, 23	*Boolean stopMetering(EndpointReferenceType serviceEPR)*
Business	Step 8	*Not standard (depends on the e-Health service)*

Execution and Monitoring MSC


Client
EMS
SLA-Controlle
1: Specify EPR of reservation resource, input for the service and submit request
2. Find reservation resource
3. Re
4: Activate SLA-Controller resource
5: Invoke Monitoring Servic
7: Inv
8: Star
9: Loop (Monitor

Implementation issues

- Globus Toolkit 4, and especially WS-GRAM and MDS4
- SLA contracts management are developed using the WSRF.NET platform
- Compelling implementation and technological challenge to see whether:
  - The 2 main grid services development tools, implement the WS-related specifications in a transparent and interoperable way

Related work

- **GRIA (Grid Resources for Industrial Applications)**
  - designed and developed the GRIA middleware based on Web Services
  - supports the confirmation of a service offer through establishment of SLA and extensions of existing SLAs
- **GRASP (Grid-based Application Service Provision)**
  - SLA management subsystem including the service provision negotiation based on QoS criteria and monitoring of the feasibility of the contract
- **Unigrids (Uniform Interface to Grid Services)**
  - SLA framework and cross-Grid brokering services in order to support Grid economics
  - integrating a Web Services Agreement-based resource management framework into the Unicore Grid middleware

…and what is special in Akogrimo?

1) SLA parameters do include network QoS parameters

2) Negotiation, reservation and execution are made on this basis also

3) Utilization of both toolkits: GT4 and WSRF.NET

4) Decision on appropriate action when violation and estimation of the significance of the violation through a dedicated component

Conclusion

•We presented an architecture to manage execution and enforce SLA in OGSA based grids and mobile Grids

- Discovery and reservation phase
- Execution and monitoring phase

•The proposed design has been implemented in GT4 and WSRF.NET

•Advancing attributes of SLA mangement with use of network resources

•For future research: To define an economic model based on how these violations affect the efficiency of Grids

Thank you!

Questions?