



Enabling Grids for  
E-science in Europe

[www.eu-egee.org](http://www.eu-egee.org)

*JRA1 meeting, 30 June 2004*

# Sharing common capabilities

**Dr Abdeslem DJAOUI / RAL**



**EGEE is a project funded by the European Union under contract IST-2003-508833**

# Content

- Faults and Versioning
- Messaging
- Logging
- Discussion on What is feasible for this project

# Beyond WSDL and SOAP?

- Applications need interoperable functionality beyond bit-level transport and communication layer
- Faults
  - GliteBaseFault?
    - Then Extend it
- Versioning
  - Backward compatible
  - What is expected from implementations
    - ignore non-understood extensions?
    - Fail?
    - Redirect message somewhere else
    - Inform client

# Basic WS Infrastructure issues

- WSDL + SOAP + HTTP
  - Does not guarantee application-level interoperability among implementations
- MUST also comply with
  - WS-I basic profile
    - XML 1.0, SOAP 1.1, WSDL 1.1, UDDI 2,
    - Assumes HTTP 1.1, SSL 3.0, TLS 1.0, ...
- Practical considerations:
  - What WSDL tools do we use?
    - Different tools support different bindings
  - Should Identify and Adopt Agreed features?
    - Document/Literal encoding?

# What do we share?

- Service implementations
  - the capability of the service is fixed
    - “Security services” is a good example
- WSDL Interfaces
  - Allows different implementations of the functionality
  - Allows the combination of more than one interface for creating new services
  - How do we deal with QoS
- Interfaces + Features and Properties
  - Fine Grained at level of PortType and operation
  - Different implementations with consistent semantics

# Messaging

- SOAP:
  - We get P2P messaging Framework for free (but it is stateless)
- Direct/Indirect messaging
  - Peer-to-peer (connection oriented)
    - Can support: security, transaction management, guaranteed delivery, ...
    - Can support queues
  - Brokers
    - Additional QoS: persistency, retention period, ...
    - Pub/sub
- What Message Formats do we need?
  - XML, Binary, ...
- Existing Message Oriented Middleware
  - JMS
  - .Net messaging
  - CORBA Notification
- WS-\* specs
  - WS-Events
  - WS-Notification

# Logging versus Messaging

- Messaging:
  - Move messages from Producers to Consumers with
    - Various persistency QoS (Store and Forward)
    - Various reliability QoS
- Logging is messaging with
  - Specific persistency QoS (Period of Time)
  - Either
    - Optimised for performance
    - Or Additional requirements at expense of performance
      - Secure Logs
  - Requirements for common message format
    - Apache Logging service
    - Log4J
- How about Accounting?
  - Are there any additional requirements in functionality or features?

# What is feasible?

- Guidelines on Tools, WS-I, Faults, Versioning
- Messaging Logging and Accounting based on R-GMA
  - R-GMA combines
    - Discovery capability
    - Messaging capability
      - Can this be the Basis for a common messaging framework?
      - Basis for Logging use?
        - » GMA and L&B common Framework
      - Basis for Accounting
        - » LCG Accounting