

# *Information Dissemination (ID)*

DAIS F2F, Redwood Shores January 2004



Cecile Madsen (IBM)  
Chris Kantarjiev (Oracle)  
Dieter Gawlick (Oracle)  
Inderpal Narang (IBM)  
Mahadevan Subramanian (IBM)  
Shailendra Mishra (Oracle)  
Vitthal Gogate (IBM)

## *Topics*

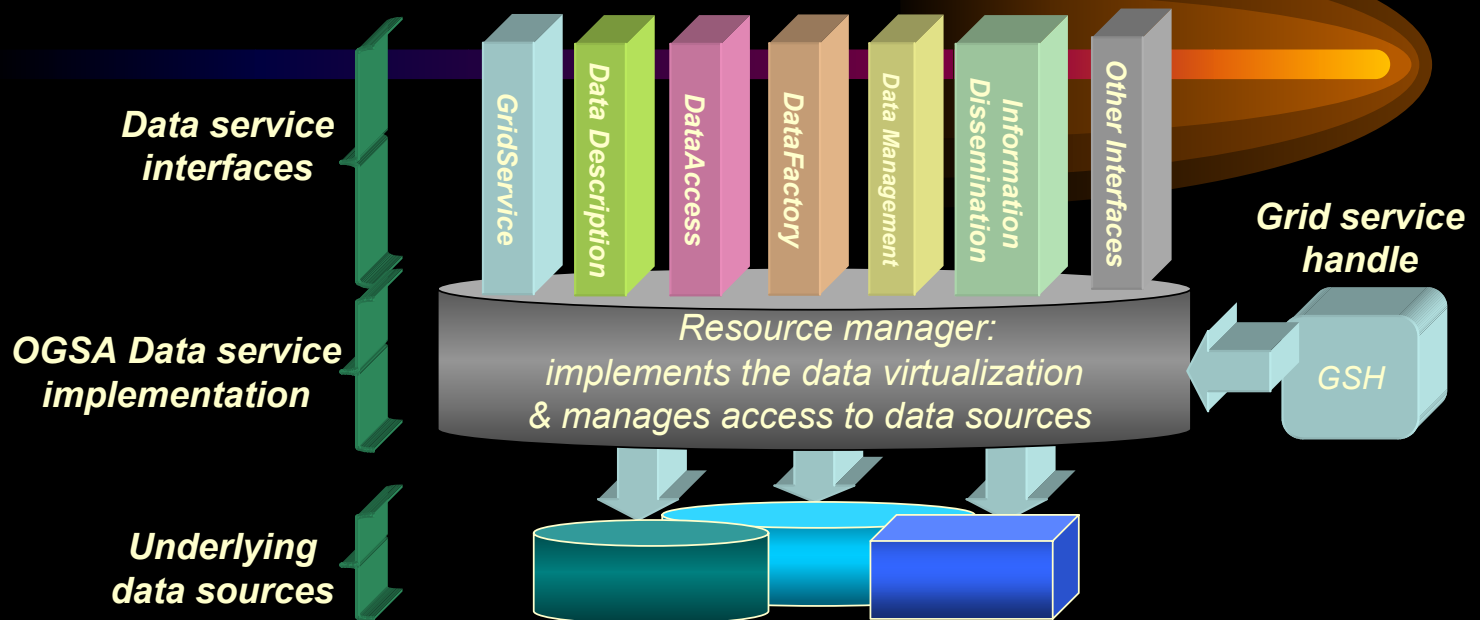


- Information Dissemination (ID) model summary
- Issues raised in DAIS October at ANL, Chicago & Manchester F2F
- ID's solution to sample DAIS Scenarios
- ID Simplification
- What's Next

# *Information Dissemination (ID)*

- Asynchronous data and event distribution model
  - 3<sup>rd</sup> party data delivery, data replication
  - Based on pub/sub
  - Dynamic operations (publication/consumption)
  - Administrative tasks and operational tasks with authorization and rules (secure, flexible)
  - Reliable, only-once delivery semantics
  - Consistency requirements (transactional)
  - Tracking and auditing of data
  - Support of open data transport protocols

# OGSA Data Service (OGSADS)



1. ID is an independent port type defined at the same level as Data Access and other port types
2. OGSA Data Service is source, target (maybe) in ID model

# *ID Interfaces*



- Publication
  - publishing rules (what/who), publisher info, implicit/explicit
  - dynamic publication (no materialization)
- Subscription
  - interest in data, future data, events (changes to data)
  - filtering rules, subscriber info
- Propagation
  - defines target: may be a Data service or a Client
  - distribution/delivery rules (1 subscription, n propagations), may include scheduling, retention, authorization rules

# *ID Interfaces*



- Consumption
  - transformation, filtering by consumer at target
  - consumer may be different from subscriber
- Publish at a source
  - publishData
- Deliver at target (push)
  - deliverData, deliverEvents
- Retrieve from source (pull)
  - getData

# *Issues raised in DAIS F2F*

- ID portType - Data Access or Data Management ?
  - Proposal: Define ID as a separate portType
    - Has two sub-portTypes
      - IDProducer
      - IDConsumer
    - ID is defined at the same level as Data Access and Data Management port types in OGSADS.
- Should ID be decomposed to sub-portTypes ?
  - Please see above.
- How is data access done by ID ?
  - ID and Data Access – not related even on same OGSADS
  - ID may use Data Access internally to define publication, subscription etc.

## *Issues raised in DAIS F2F*

- What kind of monitoring facilities are provided by ID ?
  - ID offers monitoring capability through views:
    - **Administrative views** – PubSub, Propagation rules etc.
    - **Security views** – User privileges etc.
    - **Statistical views** - #(Bytes transferred), Last error etc.
  - The views are accessed through the DataManagement portType



## *Issues raised in DAIS F2F*

- How does ID handle transactional issues?
  - ID is message oriented
    - needs transaction support from either OGSADS or WS-Transactions for consistency, high performance & scalability.
  - For improved control ID needs:
    - recoverable read
    - fast commit for better performance.

Note: This is a requirement on OGSADS or WS-Transactions for exactly once propagation.

## *Issues raised in DAIS F2F*



- If these are not spawned as services, how would a client know about existing publication subscription ids ?
  - The understanding is that these can be queried from the OGSADS through the DataDescription portType or from the registry.

## *Issues raised in DAIS F2F*

- ID may use appropriate Query languages. Does this mean ID will contain realizations that are model specific ?
  - ID will support a specific query language, if the underlying OGSADS supports it otherwise an appropriate error will be raised.
- GDD offers negotiation/inspection through DataDescription port type...?
  - ID offers inspection..through DataDescription port type. Any negotiation or capabilities are exchanged via WS-Agreements.

## *Issues raised in DAIS F2F*

- I didn't understand section on "RPC related capability" or what GDS\_PRESENTATION means ?
  - Specification lists example capabilities which may or mayn't be needed by implementation. This capability indicates presentations understood by OGSADS.

## *Issues raised in DAIS F2F*

- Transformations are required to make the result of one request passable as parameters to another. Where would these be specified ?
  - ID allows transformation to happen at each phase of information dissemination. Transformation can be specified via rules through administrative calls.
- Expression [Dir.Ref]. This seems to assume that a standard exists for dir's. Is this so ?
  - We were referring to registry and not to any other dir specification.

# *Issues raised in DAIS F2F*



- Usage scenario for ID ?
  - Usage scenarios are covered in information paper on Data Distribution. We would like to clarify if you have some specific scenario in mind.
- Information centric model for data access and data mining on unreliable environments ?
  - We need more clarification on this alternate model.
- Information flow in ID – unidirectional ? How transactional model is used by ID Apps. ?
  - Please see use case section in information paper on ID for ID Apps.
  - Information flow is bi-directional unless you have a specific scenario in mind.

## *ID – DAIS Scenarios*



- Focus of ID is to cover information dissemination scenarios with wide range of operational characteristics
- ID is not interested in scenarios already covered by DAIS

## *ID – DAIS Scenarios*

<u>Scen.</u>	<u>DS</u>	<u>Greg's Ext</u>	<u>ID</u>	<u>Remarks</u>
1	Yes			Synchronous Query
2		Yes	Yes	
3	Yes		Yes	If DS supported a push interface.
4	Yes			Synchronous Update
5		?		Pull from Non.Svc.
6			Yes	
7			Yes	
8			Yes	



## *ID – DAIS Scenario – (2)*

- Analyst locates Global Dataservice:
  - `lookup(global_registry GDS)` returns DSGSH
- Analyst subscribes expressing interest in the data through a query
  - `IDProducer::createSubscription([implicitname=QueryPublication, SQL Query, scheduleat = 3PM], Analyst)` returns SubsID.
- Analyst specifies that result of the query be delivered to 3<sup>rd</sup> party, this is done through propagation rules
  - `IDProducer::createPropagation(ConsumerURI, [subscription=SubsID, scheduleat = 9PM, protocol=, deliveryFormat=WebRowSet])` returns propagationId2.
- At 9 PM the DSGSH uses SMTP to deliver data to the consumer

## *ID – DAIS Scenario – (3a)*

- Analyst locates Global Dataservice:
  - `lookup(global_registry GDS)` returns DSGSH
- Analyst subscribes expressing interest in the data through a query – note the `implicitname` clause in the subscription rule.
  - `IDProducer::createSubscription([implicitname=QueryPublication, SQL Query, scheduleat = 3PM], Analyst)` returns SubsID.
- The analyst asks 3<sup>rd</sup> party consumer to get result data from DSGSH by passing the handle to the customer.
- The consumer specifies the consumption rules and uses `getData` to retrieve the result of the data.
  - `IDConsumer::createConsumption([subscription=SubsID, dataConsumptionFormat=WebRowSet], Consumer)` returns `consumptionId`.
  - `IDProducer::getData(consumptionId)`

## *ID – DAIS Scenario – (3b)*

- The first three steps are same as (3a)
- The 3<sup>rd</sup> party consumer would specify a schedule to the data service (DSGSH)
  - IDProducer.createPropagation(ConsumerURI, [subscription=SubsID, scheduleat = 11PM, protocol= , deliveryFormat=WebRowSet]) returns propagationId.
- At 11PM, DSGSH, would use the protocol mentioned for propagationId to send result data to the consumer at consumerURI.

## *ID – DAIS Scenario – (3c)*

- The first three steps are same as (3a)
- In this case at G1, we do createPropagation to G2
  - IDProducer::createPropagation(G2GSH, [subscription=SubsID, scheduleat = 11PM, deliveryFormat=WebRowSet]) returns propagationId.
- At 11PM, data gets pushed to G2
- Also, the other variation here is C subscribes to G2
  - IDProducer::createSubscription([implicitname=QueryPublication, SQL Query], Analyst) returns SubsID.
- The consumer specifies the consumption rules and uses getData to retrieve the result of the data.
  - IDConsumer::createConsumption([subscription=SubsID, dataConsumptionFormat=WebRowSet], Consumer) returns consumptionId.
  - IDProducer::getData(consumptionId)

# *ID Simplification*

- The following additional elements are assumed available to provide:
  - A name for a request
    - Provides reference for Alter, Start and Stop
  - The time or conditions of the executions(s) of a request
    - AT\_TIME | **ON\_DEMAND** | **SCHEDULE** | **EVENT**]
    - AT\_TIME implies there is one execution
    - ON\_DEMAND and SCHEDULE provides the ability for continuous execution e.g. for time = t1 to time = t2 execute forever
  - Specification determining the delivery

# *ID Simplification*

- DELIVERY [{RECIPIENT, INFORMATION, D\_SCHEDULE, QOS}, ...]
  - RECIPIENT [REQUESTOR, ADDRESS, EXPRESSION]
    - REQUESTOR identifies the issuer of the request; and needs to be explicitly specified if other recipients are named
    - ADDRESS identifies the address of a recipient along with a protocol, e.g., SMTP: [Joe@company.com](mailto:Joe@company.com)
    - EXPRESSION [directory reference, expression] identifies all recipients who are listed in the named directory and meet the expression.
  - INFORMATION [DATA | STATUS | FUNCTION]
    - INFORMATION identifies what is provided to specified recipient(s), data and the status, status only, or a function to allow transformations DATA is the default
  - D\_SCHEDULE allows the specification of a delivery schedule.

# *What's Next + Reference*



- What's Next:
  - agree on priority of to-do items
  - deliver new version of ID informational paper
  - any volunteers for some topic ?
- GGF9 Data Distribution Informational paper:
  - <http://www.cs.man.ac.uk/grid-db/documents.html>