



Enabling Grids for E-scienceE

NPM Interfaces

JRA4 F2F, Edinburgh, 12-13 July 2005

Alistair K Phipps (A.Phipps@ed.ac.uk)

University of Edinburgh

www.eu-egee.org



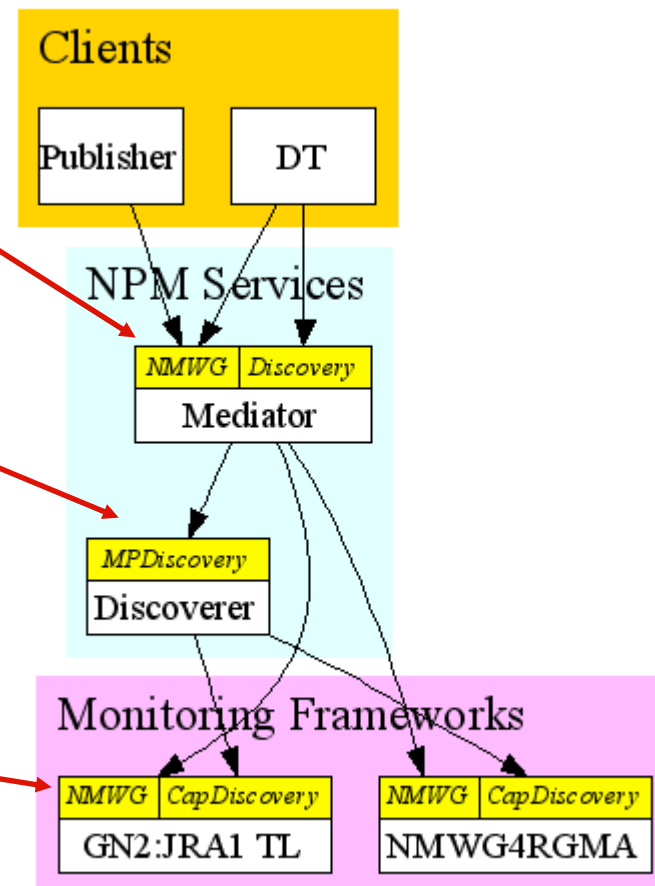
Information Society



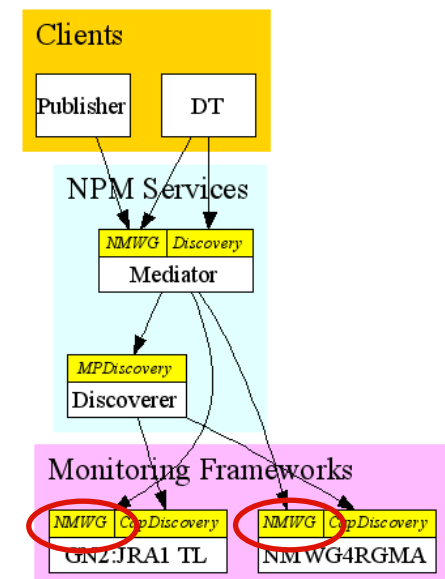
- **Covers:**
 - Interface operations for the different components
 - Outstanding issues with the interfaces
- **Objectives:**
 - Reach agreement on interface operations
 - Resolve outstanding issues

- **Three different types of service in NPM:**

- Mediator web services:
 - NMWG (with optional hopList)
 - Discovery (for measurement points / capabilities)
- Discoverer web service:
 - Discovery (for measurement points / capabilities / Framework URIs)
- Framework (e.g. GN2:JRA1 TL and NMWG4RGMA) web services:
 - NMWG
 - Discovery (for capabilities) - optional



- Framework NMWG service
- Allows a NM-WG Request to be submitted and a Report to be returned
- `report.NetworkMeasurementReport (1)`
`submit (`
`request.NetworkMeasurementRequest request (1),`
`) throws Fault`



- NPM Mediator NMWG service
- Allows a NM-WG Request to be submitted and a Report to be returned
- An optional list of hops can be sent; multiple Requests will be generated, one per hop, and results aggregated to give a single Report

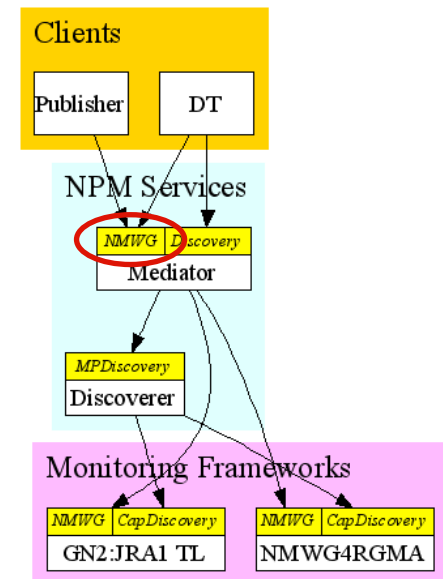
```
report.NetworkMeasurementReport (1)
```

```
submit (
```

```
    request.NetworkMeasurementRequest request (1),
```

```
    request.Node[] hops (0..n)
```

```
) throws Fault
```



- **Framework Discovery service - provides discovery of capabilities (called CapDiscovery on diagram)**

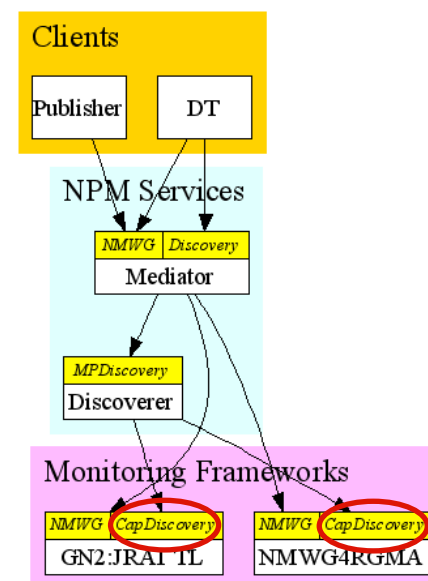
- **Uses NM-WG elements but defined by us**

```

capability.Capability[] (0..n)
getMeasurementCapabilities(
    request.Node sourceMeasurementPoint (1),
    request.Node destinationMeasurementPoint (1),
    Token characteristic (0..1),
    request.TimeInformation timeInformation (0..1)
) throws Fault
    
```

```

struct capability.Capability {
    Token characteristic (1),
    request.Statistics[] statisticList (0..n),
    request.ParameterSet parameterSet (0..1),
    request.TimeInformation timeInformation (0..1)
}
    
```

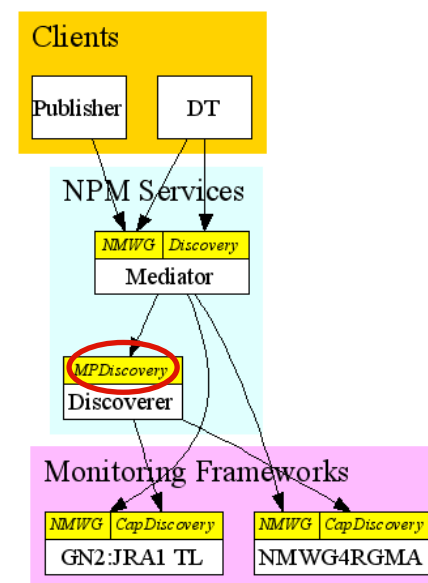


- Discoverer Discovery service - provides Monitoring Point information (MPDiscovery on diagram)

```
request.Node[] (0..n)
getSourceMeasurementPoints(
    Token characteristic (0..1)
) throws Fault
```

```
request.Node[] (0..n)
getDestinationMeasurementPoints(
    request.Node sourceMeasurementPoint (1),
    Token characteristic (0..1)
) throws Fault
```

```
request.Path[] (0..n)
getMeasurementStreams(
    Token characteristic (0..1)
) throws Fault
```



- Also provides Framework URIs for a particular measurement stream

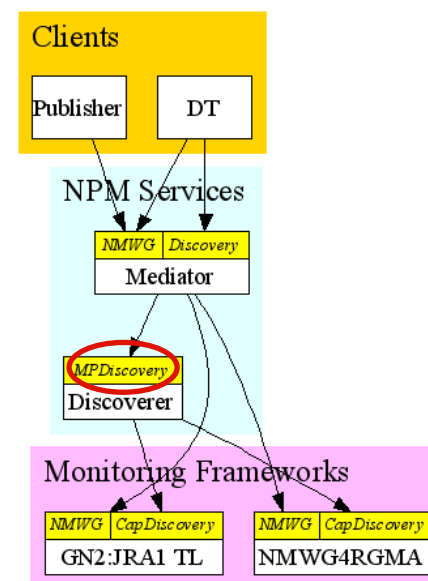
```
request.Node (1)
```

```
getFrameworkWSForMeasurementStream (
```

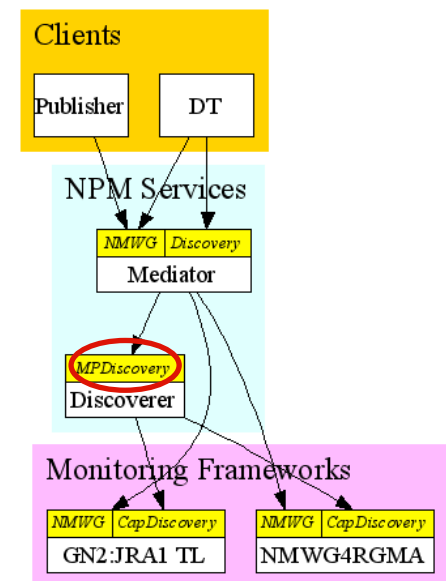
```
    request.Path measurementStream (1),
```

```
    Token characteristic (1)
```

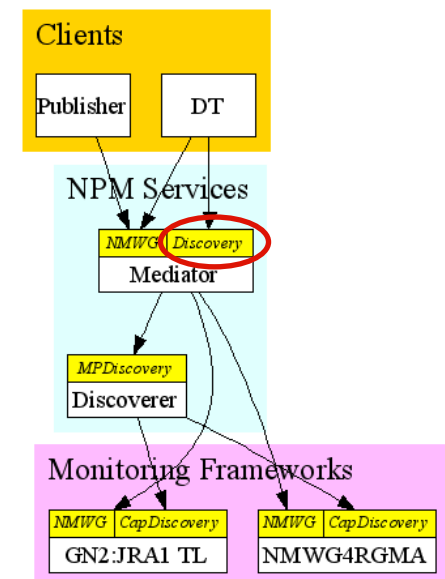
```
) throws Fault
```



- Also does pass-through of capability discovery - sends request on to Framework (same schema)
- This is so the Discoverer can potentially answer the request - e.g. for a Framework that does not support capability discovery



- Mediator discovery service (Discovery on diagram) provides all the same operations as Discoverer Discovery, except for retrieving the Framework URI (so it does MP retrieval and capability discovery)
- All operations are relayed to Discoverer
- Discovery service provided on Mediator as Mediator acts as single point of contact for clients
- However, Discovery service is separate from NMWG service so that we can be NM-WG compatible and allow other NM-WG compatible clients to connect that do not require discovery



- **There are issues with the interface as it stands - listed on following slides**
- **For discussion! Some proposals (also for discussion).**

- **Interface specifies additional optional element on submit (on Mediator NMWG service) to allow a list of hops to be specified**
- **Is this still required with v1 schema changes?**

- **Draft schema had constructs like:**
 - element e { xsd:int, element unit { xsd:token } }
- **Cause problems for automatic stub generation ("mixed" type)**
- **Should look like:**
 - element e { element value { xsd:int }, element unit { xsd:token } }
- **Fixed in v1 schema?**

- **Designed interface on assumption that NM-WG Fault schema is available - need to somehow return faults...**
- **Communication, returned by:**
 - Mediator, when it cannot contact the Discoverer
 - Mediator, when it cannot contact the Framework for a hop
 - Discoverer, when it cannot contact the Framework for a capability query (returned in turn by the Mediator)
- **Authorisation, returned by:**
 - Mediator, when the Framework web service refuses establishment of communications due to the user credential not being authorised to connect
 - Framework, when the user credential is not authorised for the specified NM-WG request (returned in turn by the Mediator)
 - Framework, when the user credential is not authorised for the specified Discovery request (returned in turn by the Discoverer and then the Mediator)

- **System**, returned by all entities on internal error (e.g. factory method failed)
- **No Data**, returned by:
 - Discoverer `getDestinationMeasurementPoints` when the specified source does not have any destinations (returned in turn by the Mediator)
 - Discoverer `getFrameworkWSForMeasurementStream` when the specified source/destination/characteristic is not in the list of measurement streams
 - Framework `submit` when there is no data available for the specified request (returned in turn by the Mediator)
 - Framework `getMeasurementCapabilities` when the measurement stream does not exist between specified source and destination at specified time (returned in turn by the Discoverer and Mediator)

- **Request, returned by:**
 - Discoverer getDestinationMeasurementPoints when the specified source is not in the list of source measurement points (returned in turn by the Mediator)
 - Discoverer getFrameworkWSForMeasurementStream when the specified source or destination is not in the list of valid measurement points
 - Framework submit when there are unsupported elements within the NM-WG request and the 'required' attribute is set on those elements, where relevant (returned in turn by the Mediator)
- **Discussion:**
 - Status of faults in NM-WG v1?
 - Alternatives

- **NM-WGv1 does not define a schema for discovery (still?)**
- **We have defined a custom discovery schema that uses NM-WG elements**
- **Is this a problem?**
- **Does NM-WGv2 define a discovery schema? Does anyone else?**
- **Proposal:**
 - Clients may interact with Mediator simply using NM-WG interface if they do not need discovery. They adhere to our custom interface if they want discovery.
 - Discoverer discovery is a totally separate interface and just happens to use elements from NM-WG schema.
 - Framework capability discovery optional (next slide).
 - Implication: not a problem.

- **Because capability discovery uses a custom schema, it's optional**
- **How can we detect and interact with Frameworks that do not support capability discovery?**
- **Proposal:**
 - Discoverer must know whether a Framework supports discovery
 - Discoverer does not pass on capability requests to such Frameworks, and simply returns whatever it can based on (source, destination, characteristic) data

- **Statistic element:**

- TimeInterval (e.g. 20050505231530 - 20050505232030)
- Value (e.g. '20')
- Unit (e.g. 'ms')
- Name (e.g. 'max')

- **Result element:**

- TimeInterval (e.g. 20050505231530 - 20050505232030)
- e.g. AvailableBandwidth
 - bandwidthBottleneck { Hop },
 - MTBF { xsd:int, element units { token } },
 - downs { xsd:int, element units { token } },
 - medianOutageLength { xsd:int, element units { token } },
 - percentUp { xsd:int, element units { token } }

- **Problem:**
 - If we use Result, every element is mandatory (and we don't have them all).
- **DJRA4.2 solution:**
 - NMWG4RGMA returns Statistic for everything except AchievableBandwidth - that is the only one for which you can request raw and the only one for which they have all the elements!
 - Not a very good solution.
- **Proposal:**
 - Only use Statistic ("raw" included)
 - How does this fit with v1 schema changes?

- **Options for TimeInterval / MaxResults interpretation:**
 - Return "maxResults" results closest to "time" between "time-minusTimeTolerance" and "time+plusTimeTolerance"
 - what if you want a sampling across a time interval?
 - Return "maxResults" results distributed evenly between "time-minusTimeTolerance" and "time+plusTimeTolerance"
 - doesn't take any account of desired "time" - what if you want the single latest result? (clearly wrong)
 - Return "maxResults" results, with one closest to "time" and additional results evenly spaced either side of "time" between "time-minusTimeTolerance" and "time+plusTimeTolerance"
 - what if you really do want the 5 values closest to the specified "time"?

- **Any other issues, comments, questions on the interfaces?**