NPM Architecture

JRA4 F2F, Edinburgh, 12-13 July 2005

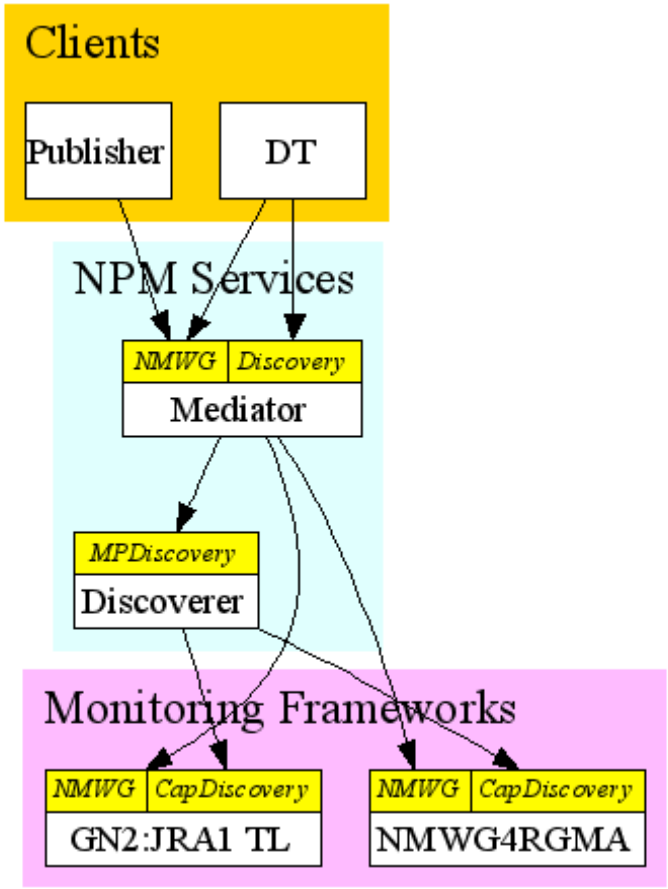
Alistair K Phipps (A.Phipps@ed.ac.uk)
University of Edinburgh

About this presentation

- **Covers:**
  - NPM architecture - the major components and how they interact
  - NPM security architecture - how the components ensure security in their interactions
- **Objectives:**
  - Ensure everyone knows the plans for the MJRA4.6 NPM architecture and is happy with them

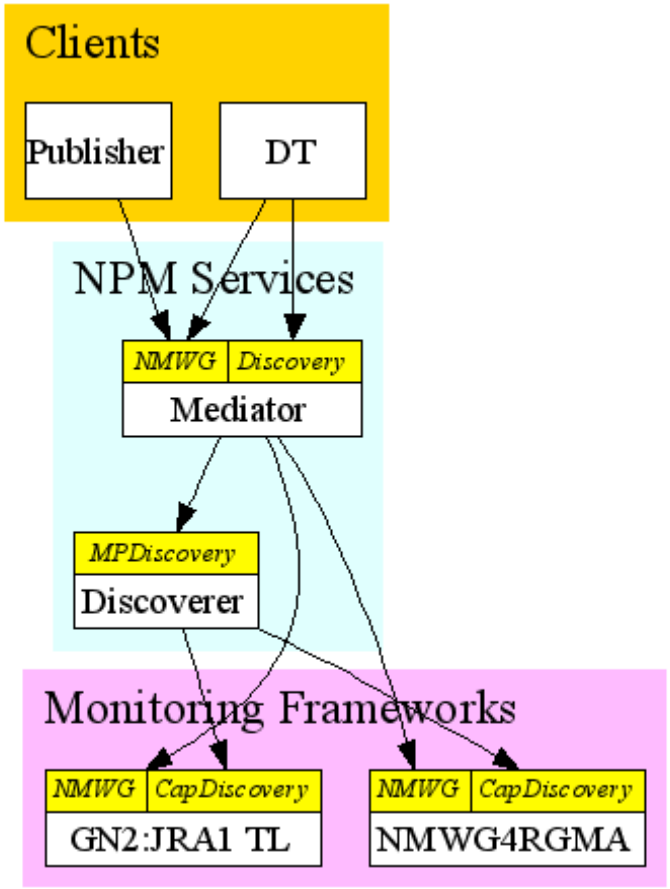
Overview

- **Clients make requests of the NPM Mediator**
- **NPM Mediator looks up information about monitoring framework services from the NPM Discoverer**
- **Monitoring framework services provide network monitoring data**
- **Planned monitoring frameworks:**
  - NMWG4RGMA: returns WP7 data by retrieval from R-GMA. Similar to DJRA4.2 service but new design and implementation.
  - GN2:JRA1 TL: returns data from the GN2:JRA1 monitoring framework. A TL instance acts as a single point of contact for many GN2:JRA1 measurement archives. TL stands for "Translation Layer" – it translates NM-WGv1 into NM-WGv2 and between different security mechanisms and credentials.



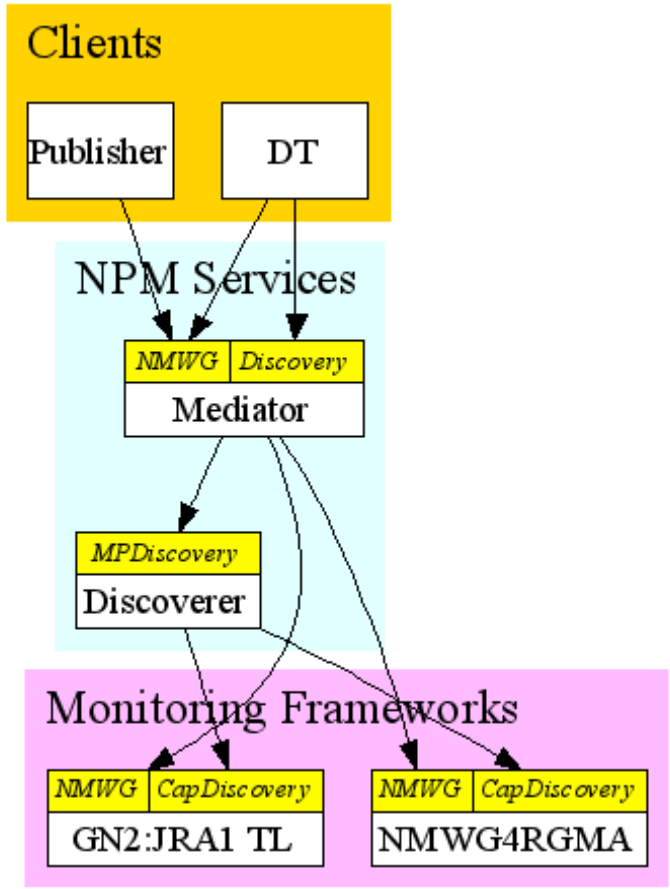
NM-WG Request

- NM-WG Requests sent by clients to Mediator “NMWG” service
- Mediator finds URI for monitoring framework web service from Discoverer “MPDiscovery” service
- Mediator sends request to relevant monitoring framework web service – e.g. NMWG4RGMA “NMWG” service
- Framework web service returns NM-WG Report
- Mediator returns NM-WG Report to client



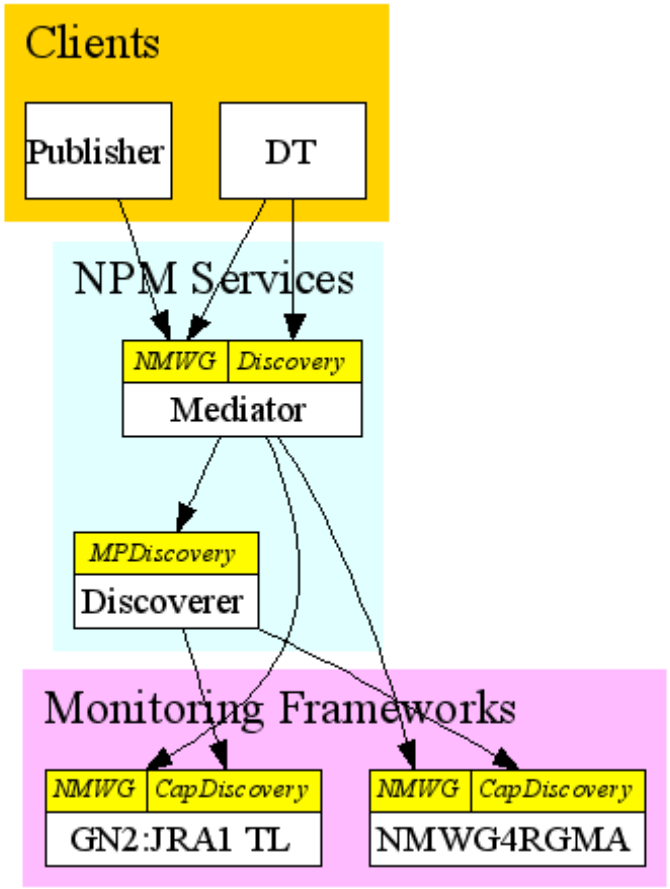
Basic Discovery

- Discovery requests, such as fetching a list of measurement points, are sent by clients to Mediator “Discovery” service
- Mediator passes request to Discoverer “MPDiscovery” service
- Discoverer has a statically configured internal list of (Source MP, Destination MP, Characteristic, Framework “NMWG” service URI, Framework “CapDiscovery” service URI)
- If the request can be answered from this list, the response is returned
- Q: Why not have clients interact with Discoverer directly?
- A: Because Mediator provides single point of contact for clients.



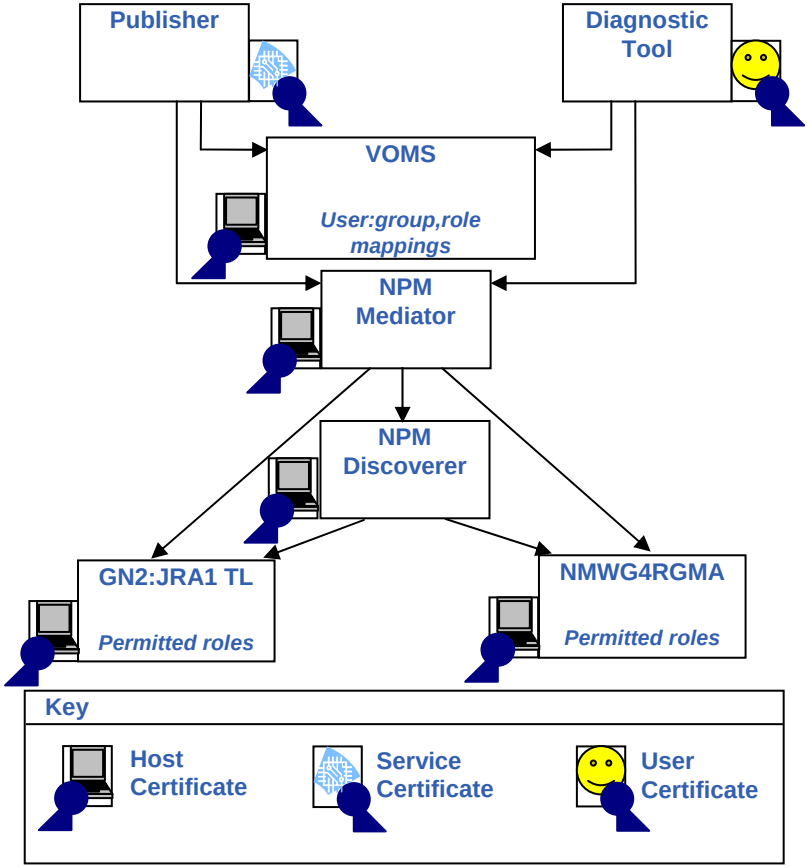
Capability Discovery

- The client may also send a “capability query” which requests information about statistics supported and times when data is available.
- Mediator again sends this to Discoverer
- Discoverer cannot answer this from its internal list, so relays it to Framework “CapDiscovery” service (optional service)
- Response returned from Framework relayed via Discoverer and Mediator to client (possibly with some processing added)
- Q: Why not store the capability information in the Discoverer?
- A: Too much information that is dynamically changing.



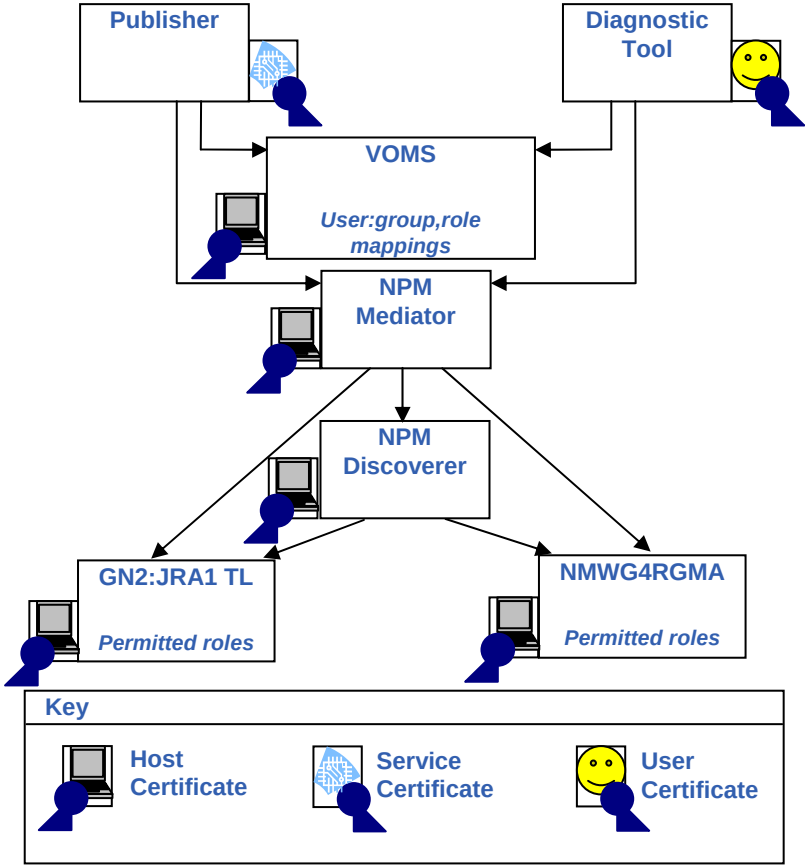
Security overview

- **DT has user certificate**
  - Actually a delegated proxy if DT is a JSP
- **Service certificate loaded into Publisher**
- **Transport layer security used in communications**



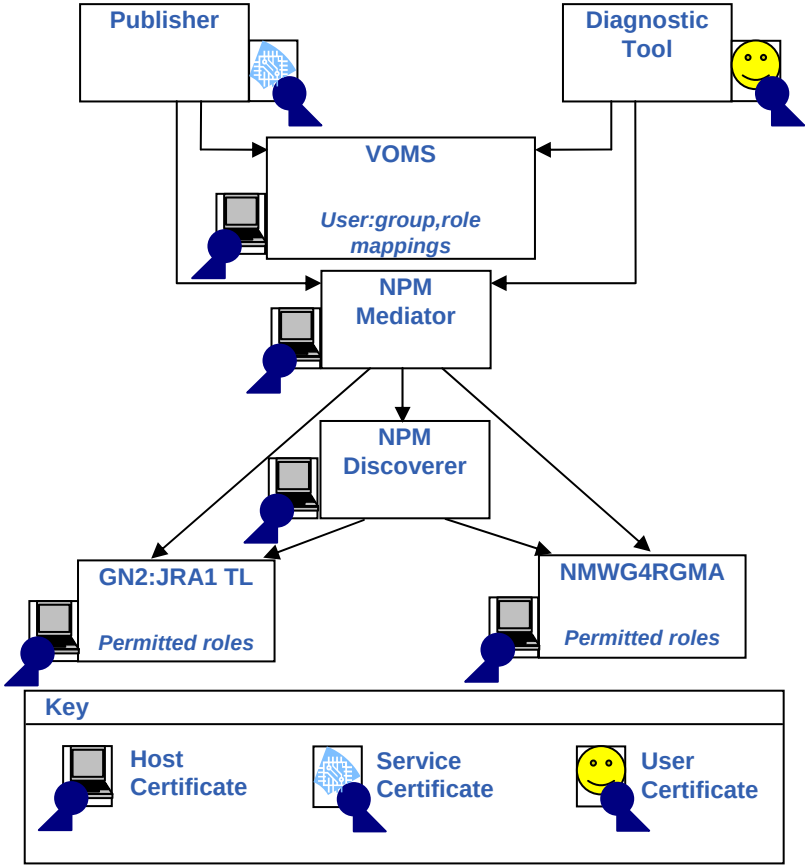
Request Security

- Client (publisher/DT) generate proxy certificate in conjunction with VOMS
- VOMS Attribute Certificate (AC) is added as extension to proxy
- Proxy is delegated to Mediator
- NM-WG Request sent to Mediator
- Mediator communicates with Frameworks, using delegated proxy
- Frameworks retrieve roles from AC in Proxy, compare with list of permitted roles for Request
- If authorised, Report returned.



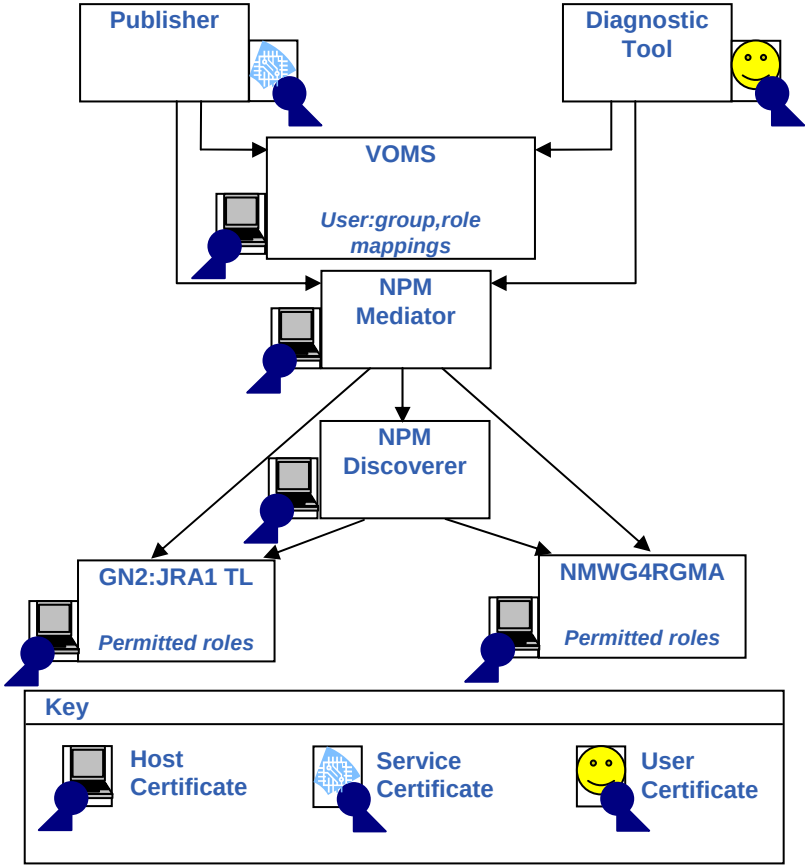
Basic Discovery Security

- Again, proxy with AC delegated to NPM Mediator
- Discovery Request sent to Mediator
- Mediator communicates with Discoverer using delegated proxy, but…
- Information stored in Discoverer (source MP, destination MP, characteristic, framework URI) considered public information – no access control. User DN just used for auditing (if required).
- Q: Why not have a list of roles permitted to retrieve data from discoverer?
- A: Because it should be up to the Frameworks who gets this information, but there is no way for us to delegate the permitted roles to them without things getting very complicated. GN2:JRA1 at least is happy with this information being public – it is the monitoring data they want to protect, not what they measure.



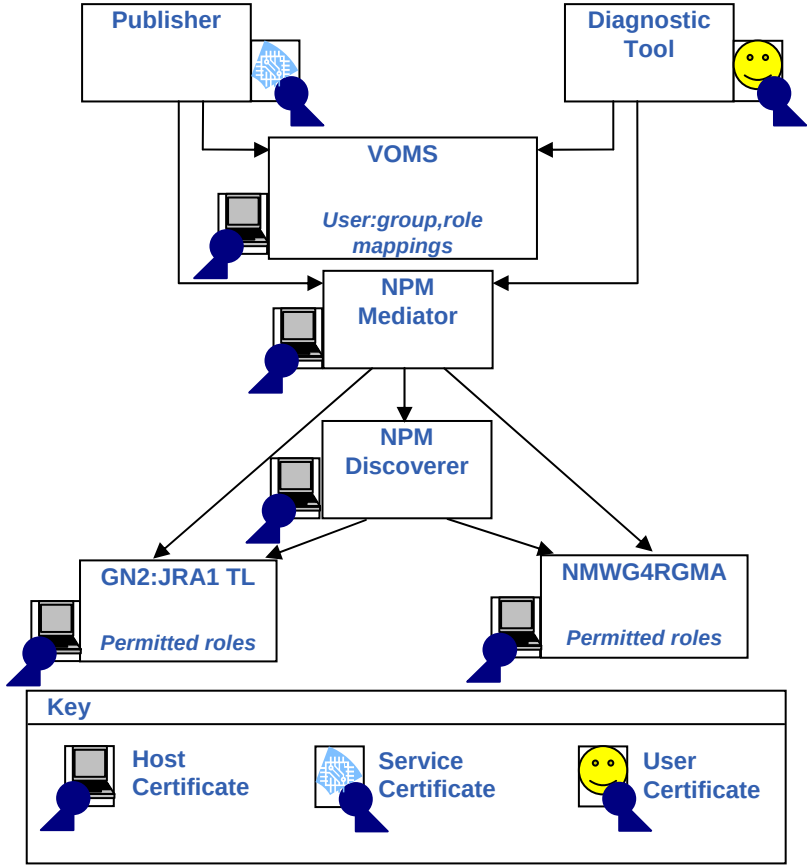
Capability Discovery Security

- Again, proxy with AC delegated to NPM Mediator
- Capability Request sent to Mediator
- Mediator delegates proxy to Discoverer (additional delegation step)
- Mediator communicates with Discoverer using delegated proxy
- Discoverer communicates with Framework using delegated proxy
- Framework carries out authorisation, similar to when a NM-WG Request received



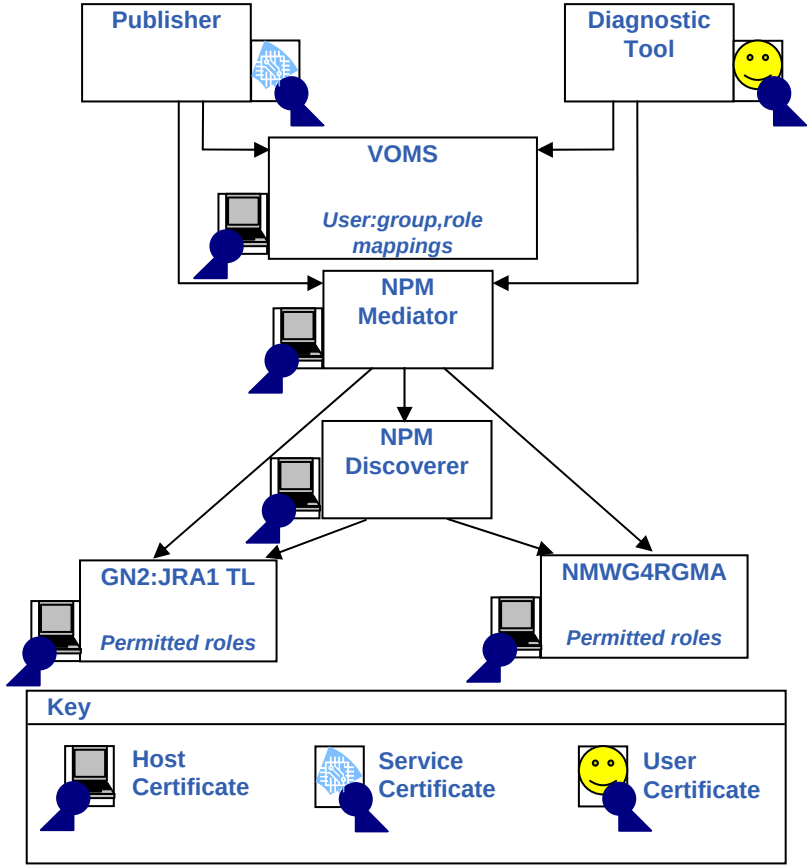
Framework Security (1)

- When the monitoring framework authorises a request, it knows the DN and role of the user that initiated the request
- However, the proxy is never delegated to the monitoring framework
- This means a separate security mechanism must be used within the framework



Framework Security (2)

- For GN2:JRA1 TL, the EGEE VO roles will be mapped to GN2:JRA1 users (more details in tomorrow's TL session)
- For NMWG4RGMA, the mapping is still to be decided (discussion in tomorrow's NMWG4RGMA session)



Summary

- **What's missing?**
  - Discovery caching – iteration 2? Data from the discoverer, such as web service URIs, should probably be cached by the Mediator.
  - Aggregation – iteration 2? Overall architecture should be compatible, but still requires raw data to be transferred from framework to Mediator.
  - Data caching – iteration 2? If we do aggregation then raw data should be cached by the Mediator.
  - Dynamic discoverer. The discoverer will read the information about monitoring points, characteristics and web service URIs from a configuration file and is therefore "static". We do not currently plan to include dynamic discovery.
  - Discovery of Mediator by clients and Discoverer by Mediator. This might be possible through use of gLite service-discovery, but we do not plan to address this in MJRA4.6.
  - Discoverer security. We intend to classify all information stored by the discoverer as public in MJRA4.6.
- **Comments?**