Globus Monitoring and Discovery

Ben Clifford

USC/ISI

The Globus Alliance

benc@isi.edu

Talk Outline

- XXX XX XXXX RESYNC THIS WITH ACTUAL TALK X XXX

MDS is

The Monitoring and Discovery System

MDS addresses two problems

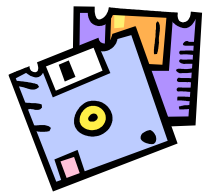
- Two problems
  - Monitoring
  - Discovery
- Look different but we can use similar techniques for both

Discovery

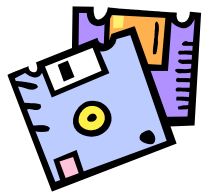
- Start with a task to perform on the grid
- For example, want to perform run a simulation

Discovering a grid resource

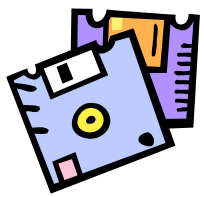
user



storage



Compute system



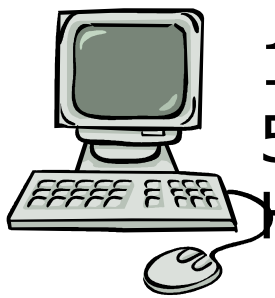
other

1. Which resources are relevant?

2. Which resources are best for task?

128 slow
CPUs
Low load

2 CPUs
100Gb disk
Medium load

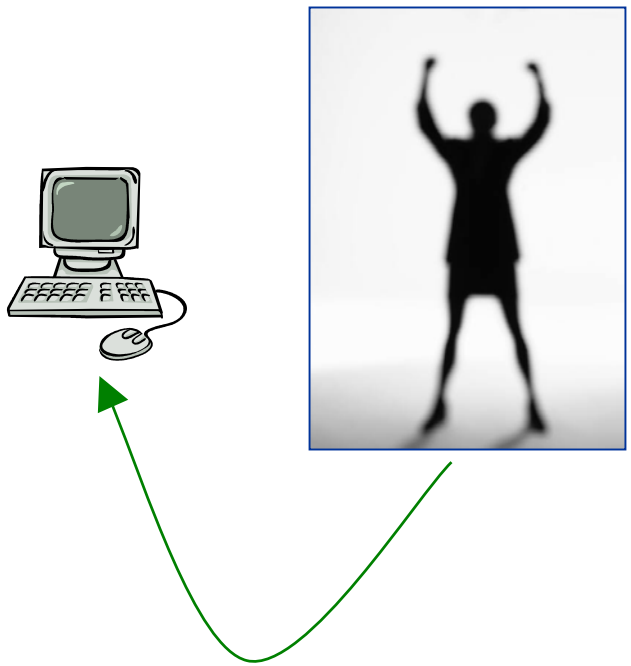


16 fast CPUs
500Gb disk
High load

3. Choose a resource

128 slow CPUs
Low load

4. Attempt to use resource



Job submission with GRAM
NOT part of MDS

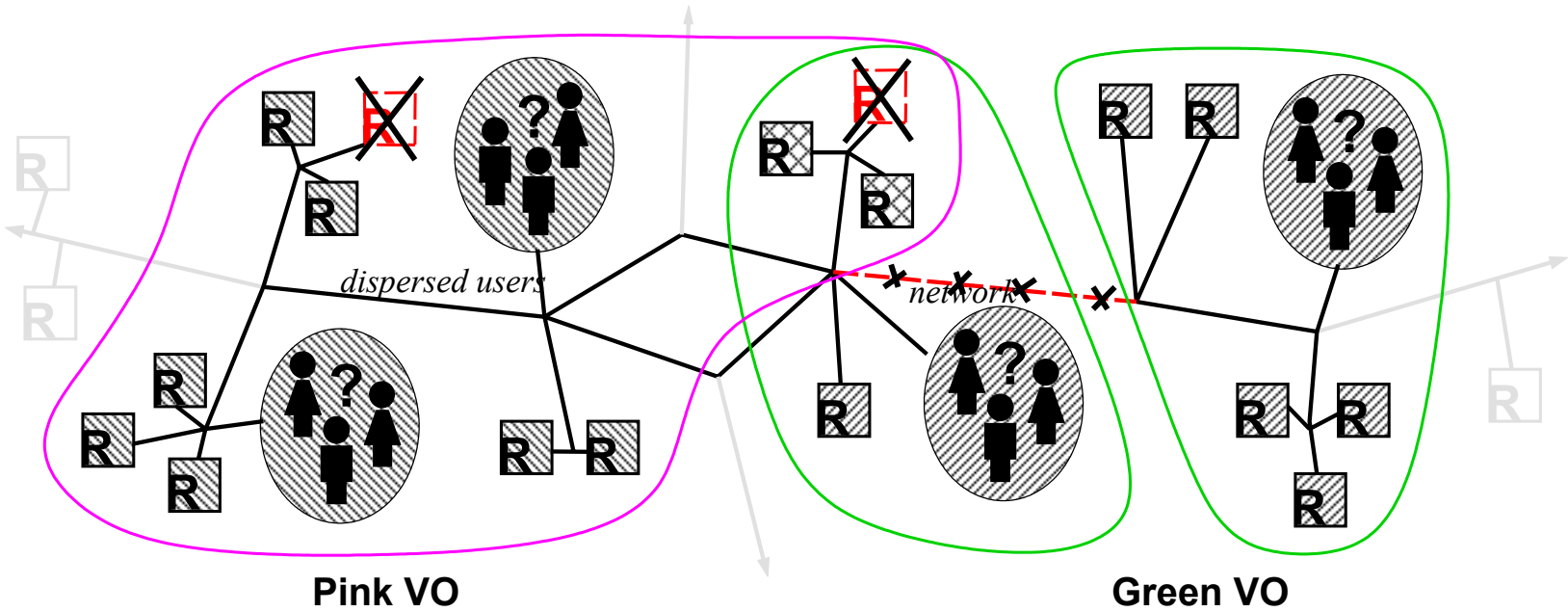
Monitoring grid resources

- TODO

What makes this difficult on the grid?

- Distributed users and resources
  - Sometimes unreliable network
- Variable resource status
  - Resources come up and go down without any centralised co-ordination
- Variable grouping
  - Different people belong to different groups (*Virtual Organisations*)
  - The grid is not cleanly partitioned

Resource Discovery/Monitoring



- Distributed users and resources
- Variable resource status
- Variable grouping
- Green VO has become partitioned because of network failure!

the globus alliance
www.globus.org

What does this look like to users?

ServiceGroup Overview

This WS-ServiceGroup is an Aggregating ServiceGroup, part of MDS4, a component of the Globus Toolkit.

This WS-ServiceGroup has 13 direct entries, 27 in whole hierarchy.

Resource Type	ID	Information	
GRAM		1 queues, submitting to 0 cluster(s) of 0 host(s).	detail
ServiceGroup		This WS-ServiceGroup has 2 direct entries, 2 including descendants.	detail
RFT		0 active transfer resources, transferring 0 files. 0 bytes transferred in 0 files since start of database.	detail
Unkown		Aggregator entry with no content from `https://ned-6.isi.edu:9000/wsrf/services/ManagedJobFactoryService`	detail
Unkown		Aggregator entry with no content from `https://viz-login.isi.edu:9000/wsrf/services/DefaultIndexService`	detail
Unkown		Aggregator entry with no content from `https://dc-user2.isi.edu:9000/wsrf/services/DefaultIndexService`	detail
ServiceGroup		This WS-ServiceGroup has 2 direct entries, 2 including descendants.	detail
Unkown		Aggregator entry with no content from `https://ned-4.isi.edu:9000/wsrf/services/ManagedJobFactoryService`	detail
RFT		0 active transfer resources, transferring 0 files. 0 bytes transferred in 0 files since start of database.	detail
ServiceGroup		This WS-ServiceGroup has 2 direct entries, 2 including descendants.	detail
Unkown		Aggregator entry with no content from `https://ned-7.isi.edu:9000/wsrf/services/ManagedJobFactoryService`	detail
RFT		0 active transfer resources, transferring 0 files. 0 bytes transferred in 0 files since start of database.	detail
ServiceGroup		This WS-ServiceGroup has 2 direct entries, 2 including descendants.	detail
RFT		0 active transfer resources, transferring 0 files. 0 bytes transferred in 0 files since start of database.	detail
Unkown		Aggregator entry with no content from `https://ned-3.isi.edu:9000/wsrf/services/ManagedJobFactoryService`	detail
Unkown		Aggregator entry with no content from `https://devrandom.isi.edu:9000/wsrf/services/DefaultIndexService`	detail
Unkown		Aggregator entry with no content from `https://viz-login.isi.edu:9000/wsrf/services/DefaultIndexService`	detail
ServiceGroup		This WS-ServiceGroup has 2 direct entries, 2 including descendants.	detail
RFT		0 active transfer resources, transferring 0 files. 0 bytes transferred in 0 files since start of database.	detail
Unkown		Aggregator entry with no content from `https://ned-2.isi.edu:9000/wsrf/services/ManagedJobFactoryService`	detail
ServiceGroup		This WS-ServiceGroup has 2 direct entries, 2 including descendants.	detail
RFT		0 active transfer resources, transferring 0 files. 0 bytes transferred in 0 files since start of database.	detail
Unkown		Aggregator entry with no content from `https://ned-5.isi.edu:9000/wsrf/services/ManagedJobFactoryService`	detail
RFT		28 active transfer resources, transferring 4 files. 158848396 bytes transferred in 8032 files since start of database.	detail
ServiceGroup		This WS-ServiceGroup has 2 direct entries, 2 including descendants.	detail
Unkown		Aggregator entry with no content from `https://ned-1.isi.edu:9000/wsrf/services/ManagedJobFactoryService`	detail
RFT		0 active transfer resources, transferring 0 files. 0 bytes transferred in 0 files since start of database.	detail

Examples of Useful Information

- Characteristics of a compute resource
  - Software available, networks connected to, load, type of CPU, disk space
- Characteristics of the Globus infrastructure
  - Hosts, resource managers, service availability
- Policy?

- More examples later on

Key Concepts

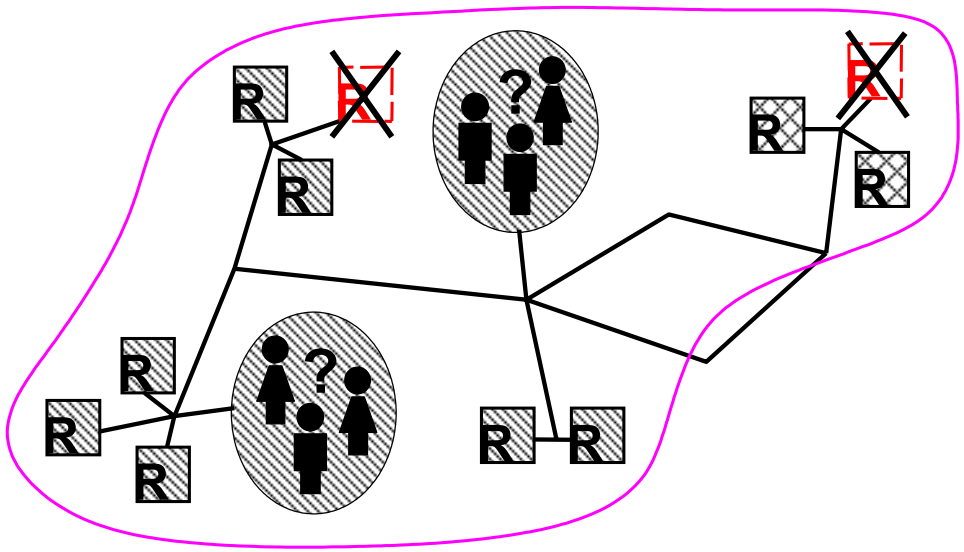
- Virtual Organizations (VOs)
  - Group together resources and users in related communities
  - Support community-specific discovery
  - Specialized views
  - Scalability

Virtual Organizations

- Collaborating individuals and institutions
  - Enable sharing of resources
  - Non-locality of participants
- Dynamic in nature
  - VOs come and go
  - Resources join and leave VOs
  - Resources change status and fail
- Community-wide goals
- Must not interfere with each other

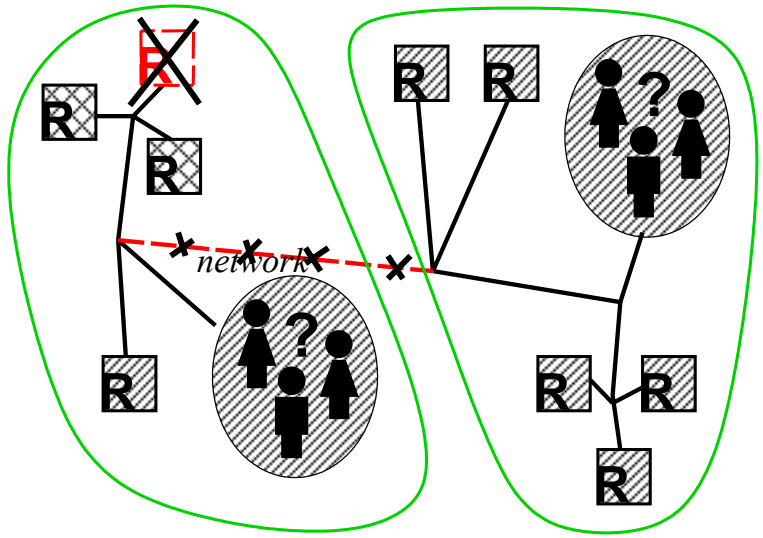
- In support of this, provide VO-wide resources for MDS
  - Other VO-wide resources, such as CAS for security

Pink VO

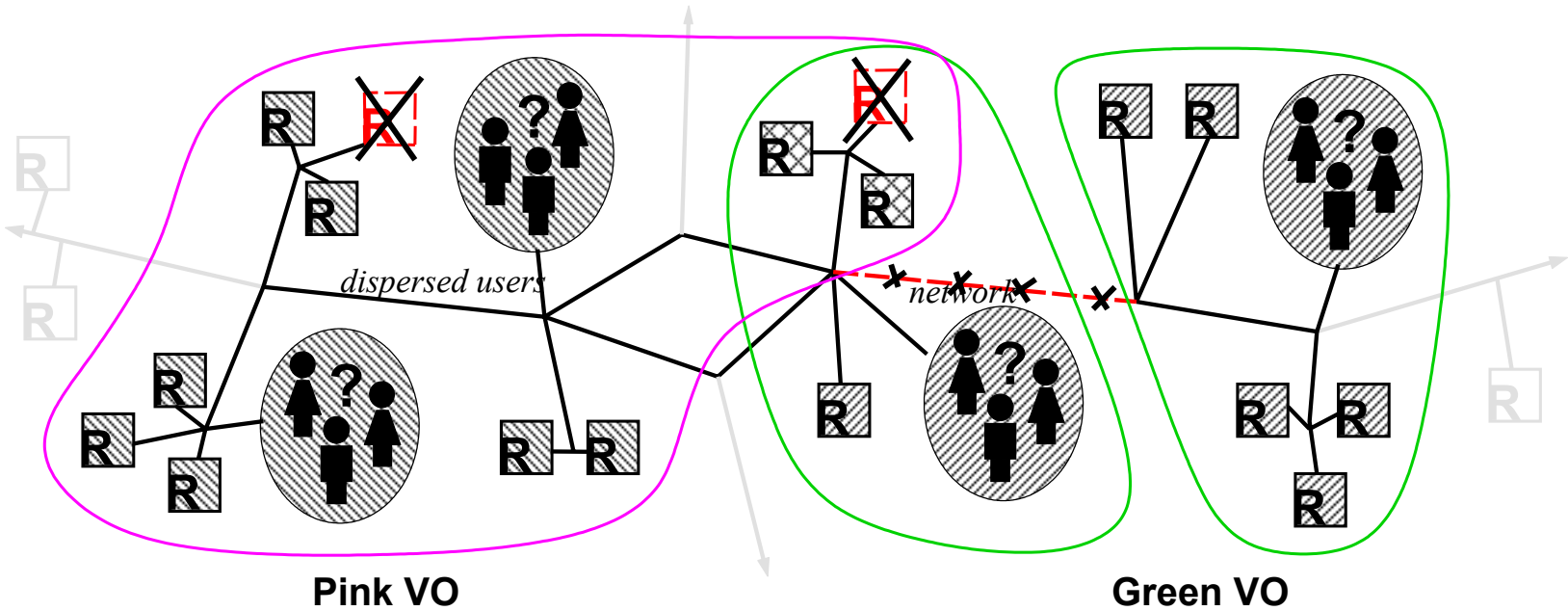


Pink VO

Green VO



The Grid



- Some resources are in both VOs
- Some resources are in neither VO
  - But are in other VOs

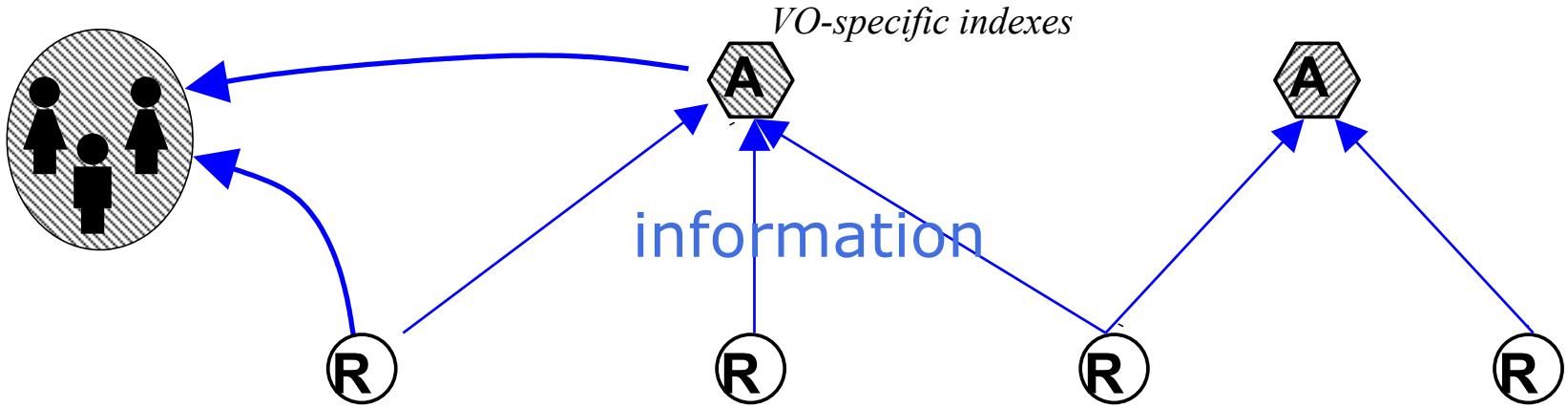
Scalability

- Large numbers
  - Many resources
  - Many users
- Independence
  - Resources shouldn't affect one another
  - VOs shouldn't affect one another
- Graceful degradation of service
  - "As much function as possible"
  - Tolerate partitions, prune failures

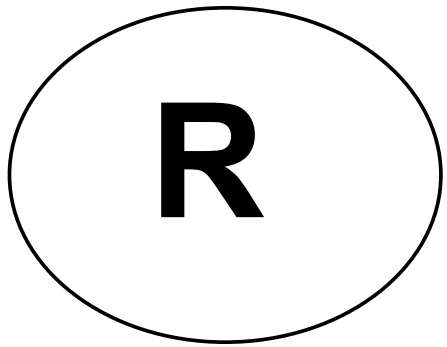
Grid Information: Facts of Life

- Information is always old
  - Time of flight; changing system state
- Distributed snapshot of state hard to obtain
- Components will fail
- Scalability and overhead
- Many different usage scenarios

MDS in a VO

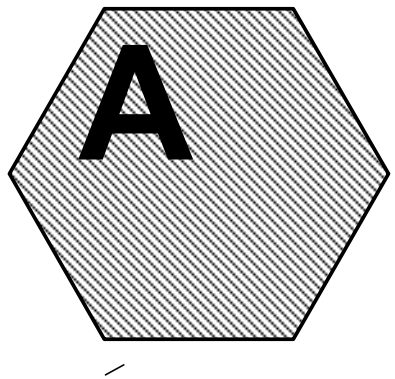


MDS in a VO
Components - Resources



- Resources
  - Things you can use on the grid
  - For example, GRAM installations
  - Resources have MDS information
    - Might publish themselves (eg. WSRF based services)
    - Might be collected by a separate probe (eg. GridFTP)
  - Resources do other things – for example, GRAM is primarily intended to run compute jobs

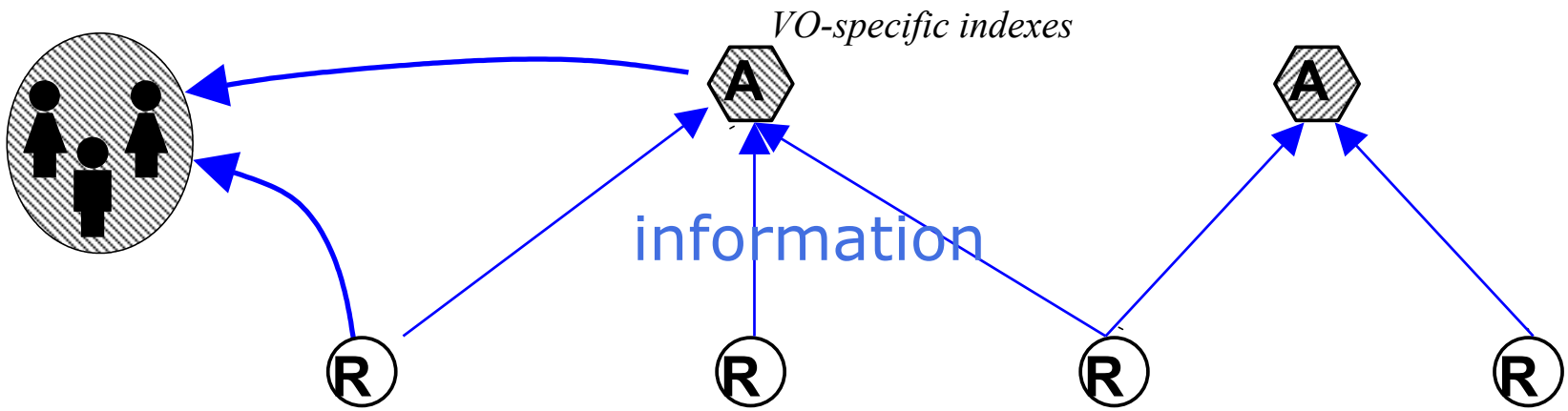
MDS in a VO Components



- Indexes
- Collect information from Resources and publishes in one place

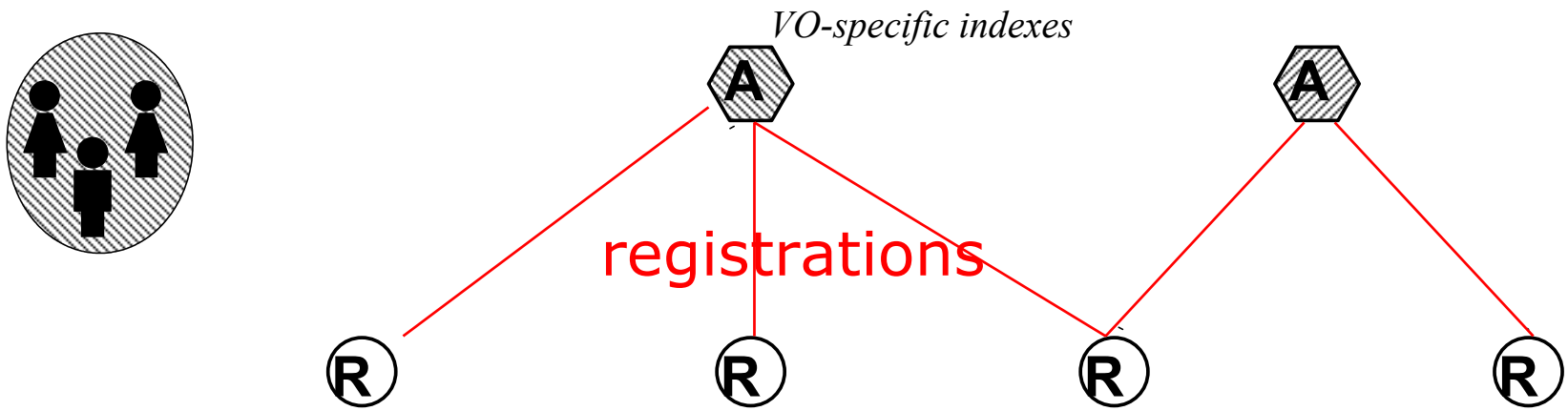
- A is for ‘Aggregator’

MDS in a VO
Flow of information



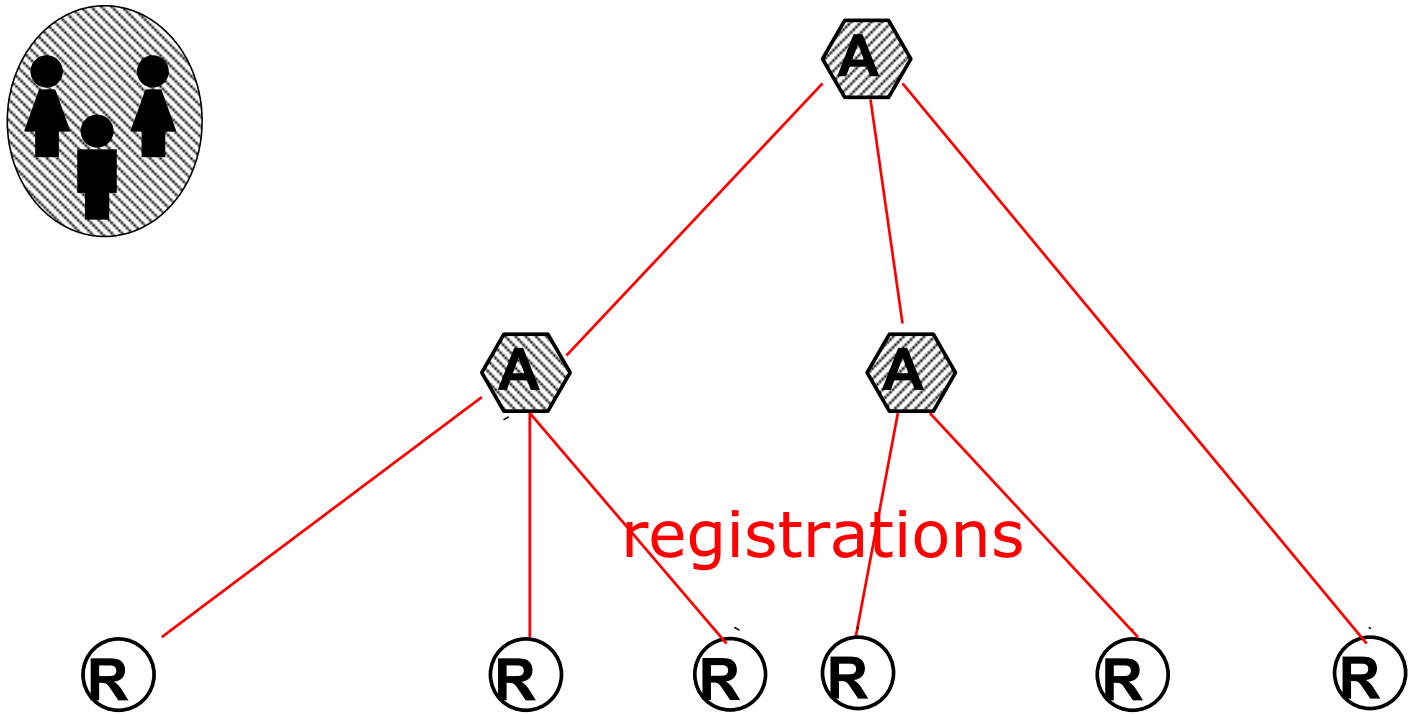
- Users can get information in two ways
  - Query from resources directly
    - Information more recent
    - Higher load on resources and on requestor
    - Need to know which resources exist in order to query them
  - Query from index
    - Do not need to know existence of each resource
    - Information is older

MDS in a VO
Constructing a hierarchy



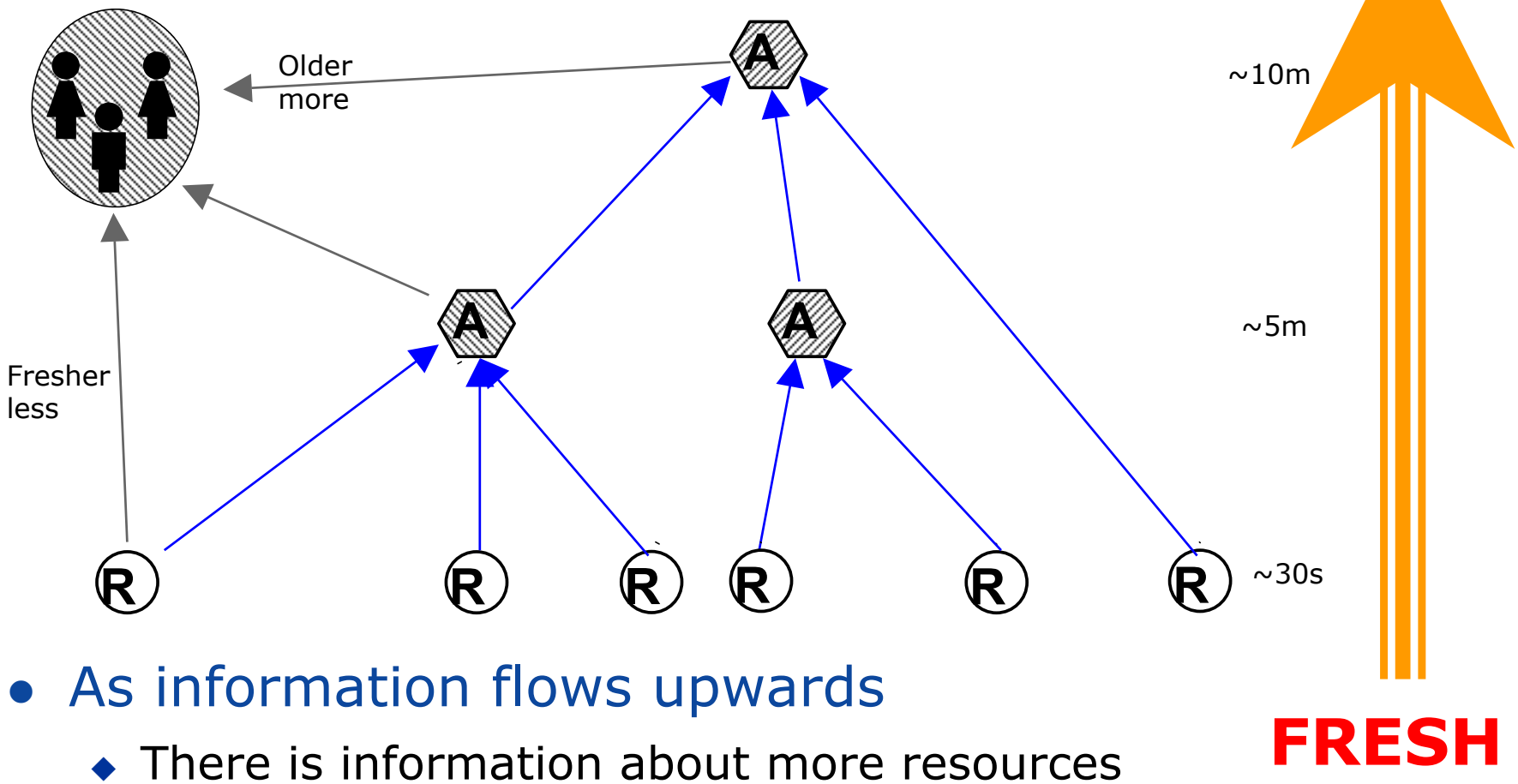
- Resources registered into Indexes
- Soft-state registration
  - Keeps index clean
  - Old entries disappear automatically
- Registrations configured at index or at resource

MDS in a VO
Constructing a hierarchy



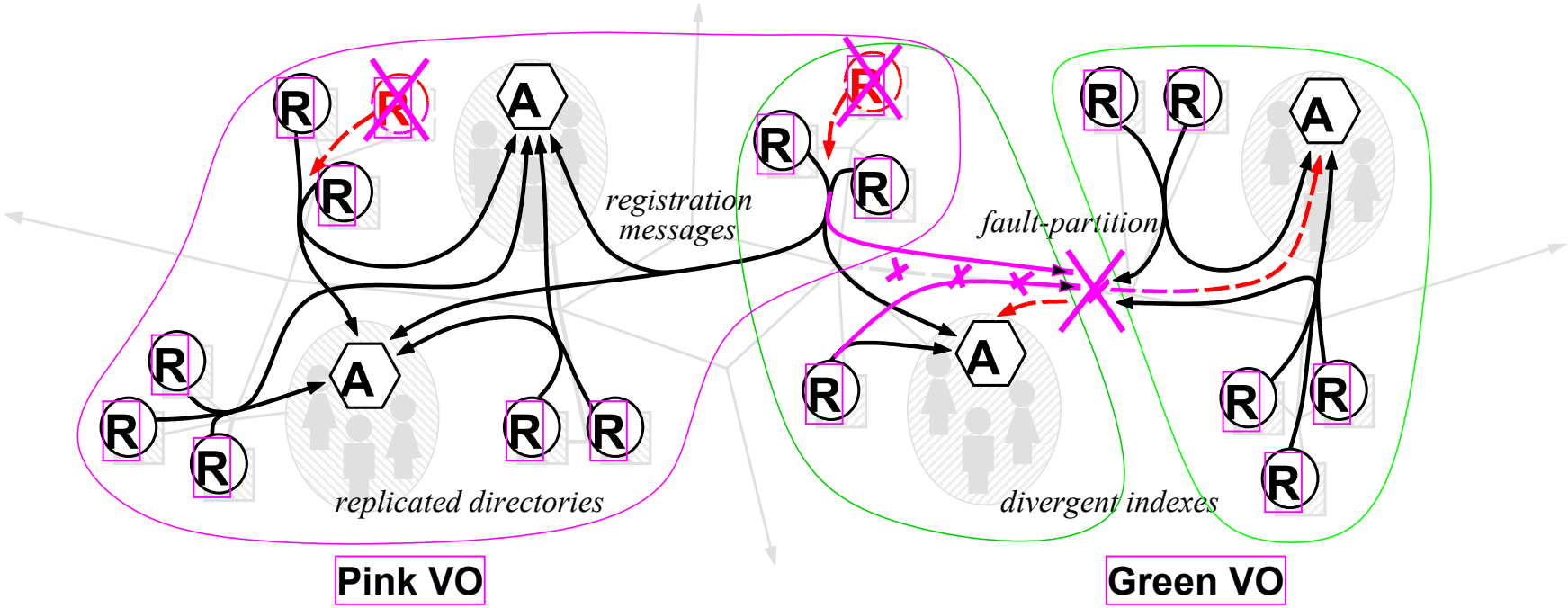
- Indexes can be registered to other indexes
- Index at top of slide contains information about all 6 resources

MDS in a VO
Age of information



- As information flows upwards
  - There is information about more resources
  - The information is generally older

Distributed Services



- Service scales with Grid growth
- Loose consistency model tolerates failures
- Interoperability through common protocols

MDS components

- MDS services
  - Index
  - Trigger
- Component-specific information
  - Queue status information for GRAM
- GT4 WSRF core provides underlying layer

Grid Resources

- MDS collects information about each Grid Resource
- WSRF resources can provide their own information
- Other types of Grid Resource have an MDS information source to collect the information.
- Much information is dynamic
  - Load, process information, storage information, etc.
- "White pages" lookup of resource information
  - Ex: How much memory does this compute resource have?
- "Yellow pages" lookup of resource options
  - Ex: Which queues on this resource allows large jobs?

Collective services

- Collective services aggregate information from multiple resource services, and process in specific ways
- Index service
- Trigger service

Index Service

- Maintain set of registered Grid Resources
  - Track incoming live registrations
  - Indexes can be registered hierarchically
- Cache of monitoring data for each Grid Resource
- Akin to web search engines
- “Which compute resources have 32 or more processors?”

Trigger Service

- List of rules
- Rules applied to monitoring data
- When rules match, action performed
  - Send e-mail to users/administrators

MDS Security

- GT WS core security allows operations to be restricted to certain listed users.
  - Restrict who can see service data
  - Restrict who can register into an index service

Gathering monitoring data

- Data can be gathered about Grid Resources in a number of ways.
  - WSRF
    - Most GT4 services
  - Protocol specific sources
    - GridFTP, RLS, other

WSRF

- Web Services Resource Framework
- Grid Resources <-> *WS-Resources*
  - XML-based *WS-Resource Properties*
    - Monitoring and discovery information
  - Standard access mechanisms
    - *Polling*
    - *Subscription (using* WS-Notification*)*

Information models

- Each information sources publishes information in XML according to some schema.
- Some times the author of the information source or the grid resource defines that schema.
- Some collaborative efforts to define common schemas– for example *GLUE* for compute information
- Schema typically written in XSD, but not required

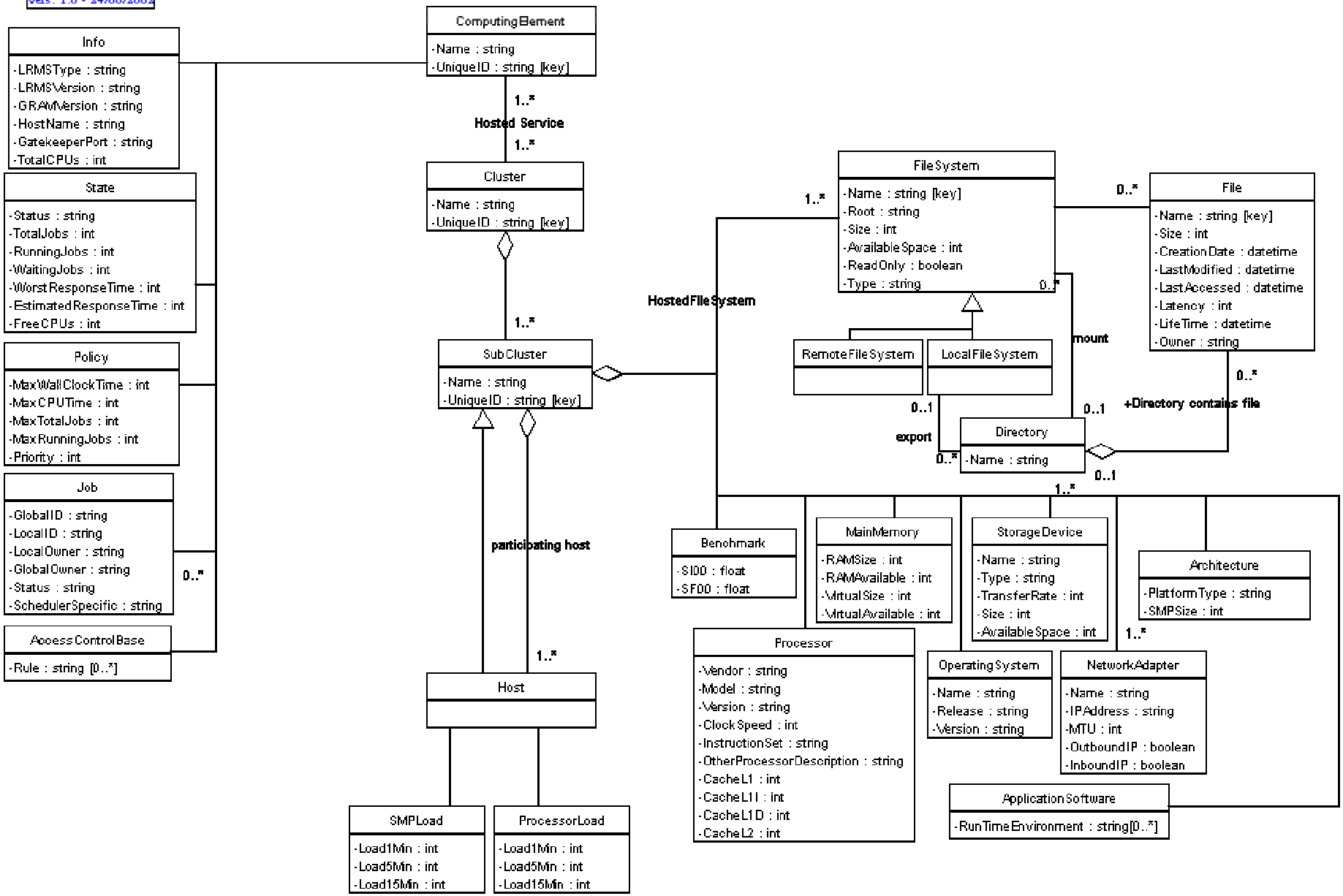
GLUE schema

- Grid Laboratory Uniform Environment
- Schema developed by DataTAG for EU/USA interoperability.
- Modelled in UML
- Implementations
  - XML version for MDS
    - Information collected from various cluster monitoring systems
  - Also: LDAP and SQL versions (used by older versions of MDS and other monitoring systems).

the globus alliance
www.globus.org

GLUE Schema v1.0

vers. 1.0 - 24/08/2002



GLUE schema example

```
<ce:Cluster ce:Name="stomoxys.isi.edu"
    ce:UniqueID="stomoxys.isi.edu"
    ogsi:availableUntil="2003-12-01T23:00:25.690Z"
    ogsi:goodFrom="2003-12-01T22:59:54.690Z"
    ogsi:goodUntil="2003-12-01T23:00:24.690Z">
<ce:SubCluster ce:Name="stomoxys.isi.edu"
    ce:UniqueID="stomoxys.isi.edu">
<ce:Host ce:Name="stomoxys.isi.edu"
    ce:UniqueID="stomoxys.isi.edu">
<ce:OperatingSystem ce:Name="Linux" ce:Release="2.4.18-14" />
 <ce:ProcessorLoad ce:Last15Min="074"  ce:Last1Min="095"
    ce:Last5Min="078"  />
...
</ce:Cluster>
```

MDS user interfaces

- General purpose UIs
  - Web browser based interface - WebMDS
  - Command line tools
- Specialized clients
  - Brokers

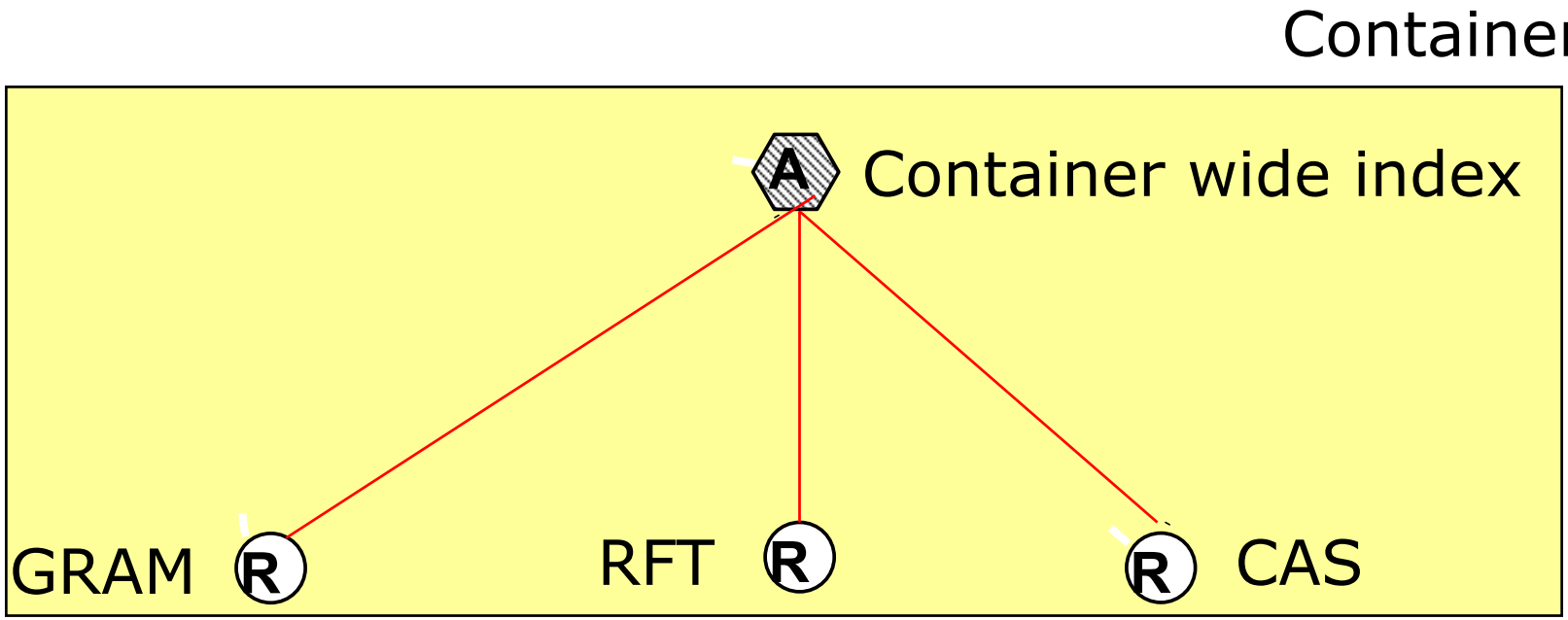
WebMDS

- Web-based interface to display monitoring information
- Easily extensible for new data using XSLT

MDS in the GT4 container

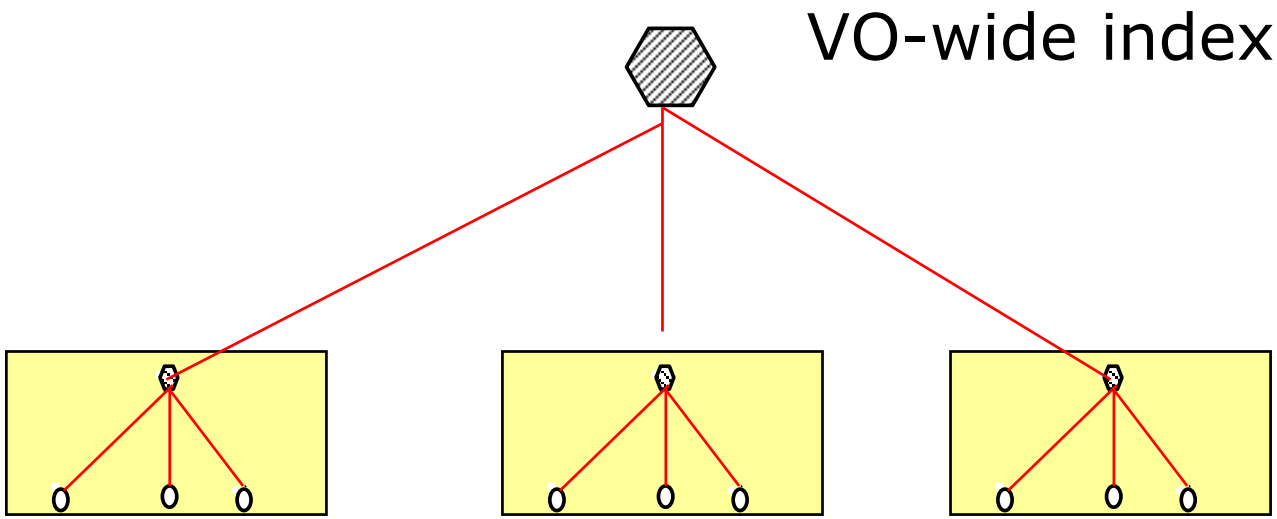
- Layout of MDS services in container
- Configuration

Containerwide index



- Each GT4 container has a local index
- Collects information about services in that container
- Each service registers to container index when correctly configured

VO-wide indexes



- Local indexes can be registered to VO wide indexes
- Config file at resource container or at VO index – contains URL for resource or VO index

Registering a container index into a VO index

- Config file:

`$GL/etc/etc/globus_wsrf_mds_index/hierarchy.xml`

- Two ways:
  - At the VO index
    - Configure the URLs of container indexes
    - Add lines like:
    - <upstream>http://myresource.isi.edu:8080/wsrf/services/DefaultIndexService</upstream>
  - At the resource containers
    - Configure the URLs of the VO indexes
    - Add lines like:
    - <downstream>http://myvo.org:8080/wsrf/services/DefaultIndexService</downstream>

Configuring GRAM to use a cluster monitoring system

- GRAM extracts and publishes cluster information from either Ganglia or Hawkeye
- $GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/gluerp.xml
- <defaultProvider> tag specifies whether to use Ganglia or Hawkeye or none.
- Uncomment appropriate example supplied in the config file

Programming with MDS4

- More information
  - Directly from resources
  - Gatewaying via other monitoring systems
- Write new collective functionality

WSRF APIs

- WSRF services can be implemented in a number of languages.
  - Java – Globus
  - C – Globus
  - Python – LBL
  - .NET – Virginia
- Write your service using GT4 WSRF core and you will get MDS compatibility almost magically.

Publish new information

1. Decide what information to publish into MDS and where
2. Define XML schema
3. Write code to publish info
    1. As part of a new service
    2. As part of existing implementation

Write new collective layer services

- Aggregator framework provides:
  - Common VO-level service functionality
    - Registration management
    - Collection of information from Grid Resources
  - Developer can plug in specialised functionality
    - Index, Trigger, as well as a prototype of an archiver all use this framework.

New ways to collect informtion

- Aggregator has pluggable sources
  - Collect via WSRF (poll or subscription)
  - Collect using custom executable
- Extensible – new ways to collect (new information sources) can be written in Java or as executables

Historical MDS

- MDS1 (Metacomputing Directory Service)
  - Centralized database
  - Globus 1.1.2 and earlier
  - Did not scale
  - Single point of failure
  - LDAP based
- MDS2 (Monitoring and Discovery Service)
  - Distributed services
  - In Globus Toolkit 1.1.3 and GT2.x
  - Two classes of server: GIIS and GRIS
  - LDAP based
  - Lazy caching presented scalabilty problem
- MDS3 (Monitoring and Discovery System)
  - Even more distributed services
  - Based around OGSI standard
  - In Globus Toolkit 3.x
- MDS4
  - Based around WSRF standards
  - In Globus Toolkit 4.x
  - More native components - (web UI, trigger service, ganglia, hawkeye)

Ways forward

- Archiving
  - Record historical monitoring data
- Partial notification
  - Send only small updates rather than a complete result set
  - Helps scalability
- Gateway into other monitoring/discovery systems
  - Eg. monalisa
- More interesting monitoring data
- Security
  - Who can see what
  - Referrals

What now?

- Download the toolkit
  - http://www.globus.org/toolkit
- Globus support lists:
  - discuss@globus.org
  - developer-discuss@globus.org
- MDS web pages:
  - http://www.globus.org/mds