

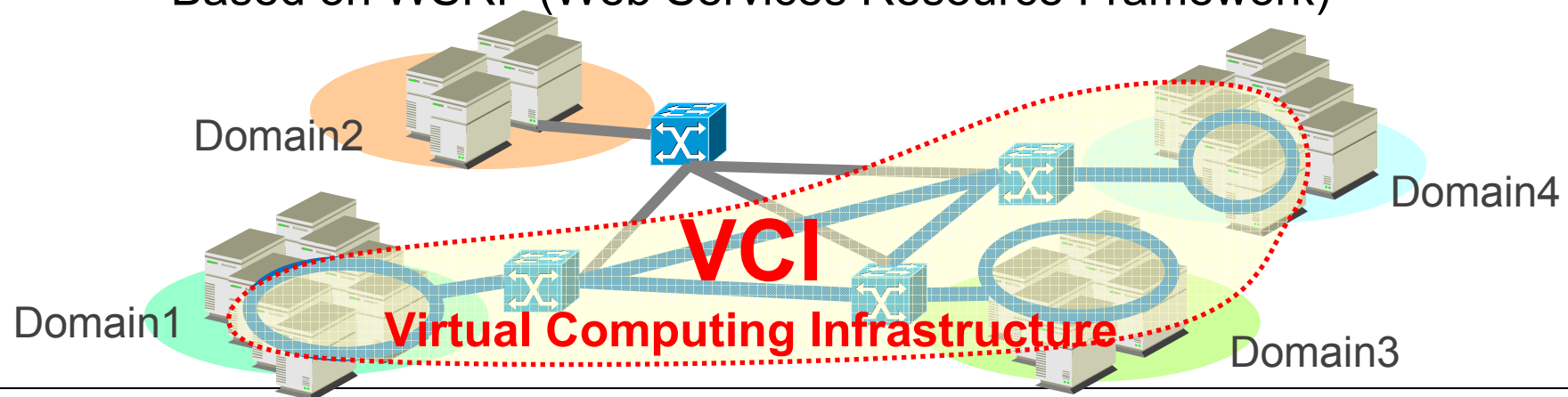
Design of a Domain Authorization-based Hierarchical Distributed Resource Monitoring System in cooperation with Resource Reservation

Atsuko Takefusa, Hidemoto Nakada, Seiya Yanagita,
Fumihiro Okazaki, Tomohiro Kudoh, Yoshio Tanaka

National Institute of Advanced Industrial Science and Technology
(AIST)

Grid and Network

- Grid and Network provisioning technologies enabled high-quality virtual computing infrastructure (VCI) spanning **multiple domains**
- We have developed the **GridARS** co-allocation framework [JSSPP2007]
 - Negotiates with **multi-domain resource managers**
 - Co-allocates suitable resources guaranteed requested performance and timeframes with **advance reservation**
 - Based on WSRF (Web Services Resource Framework)



Inter-domain Demonstration by G-lambda (Japan) and EnLIGHTened (the US)

- In the fall of 2006, the world's first inter-domain coordination of resource managers for in-AR of network and computers
- Co-allocation by GridARS (GL) and HARC (EL)
- 3 network domains and 10 cluster sites
- Parallel MPI and HD video stream applications over dynamic VCI



Issues for Providing VCI (1/2)

- **Monitoring of reserved and dynamic resources**
 - Constituent resources are provisioned and distributed
→ Comprehensive resource monitoring of VCI required
 - General monitoring tools (e.g. Ganglia) do not support cooperation with resource reservation
 - Our reservation resource monitor (**RRM**)
 - Gathers and provides reservation status (e.g. time tables, reserved/activated/released)
 - NOT provide resource utilization (e.g. CPU, transfer rate)
- **Authorization of monitoring**
 - VCI resources provided from multiple administrative domains
 - Ganglia and RRM discloses all resource information
 - Fine grain authorization required for each monitoring information item

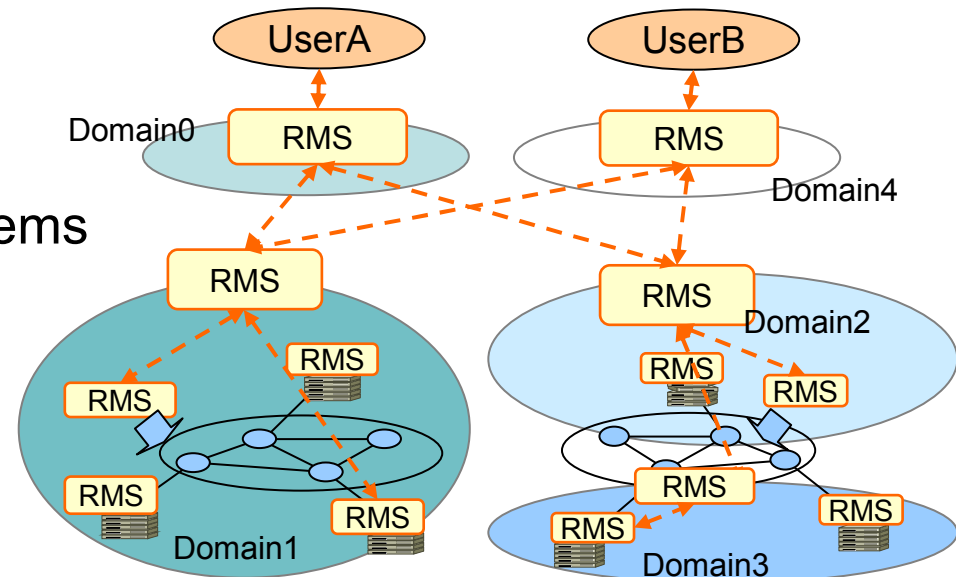
Issues for Providing VCI (2/2)

- **Hierarchical architecture**

- Resource management domains are composed hierarchically in actual environments
- End user may not informed details of VCI, especially in commercial services
- Authorization and filtering monitoring information **at each domain** important

- **Interoperability**

- Cooperation of multiple resource management systems (RMS) **developed by multiple providers**



Contributions

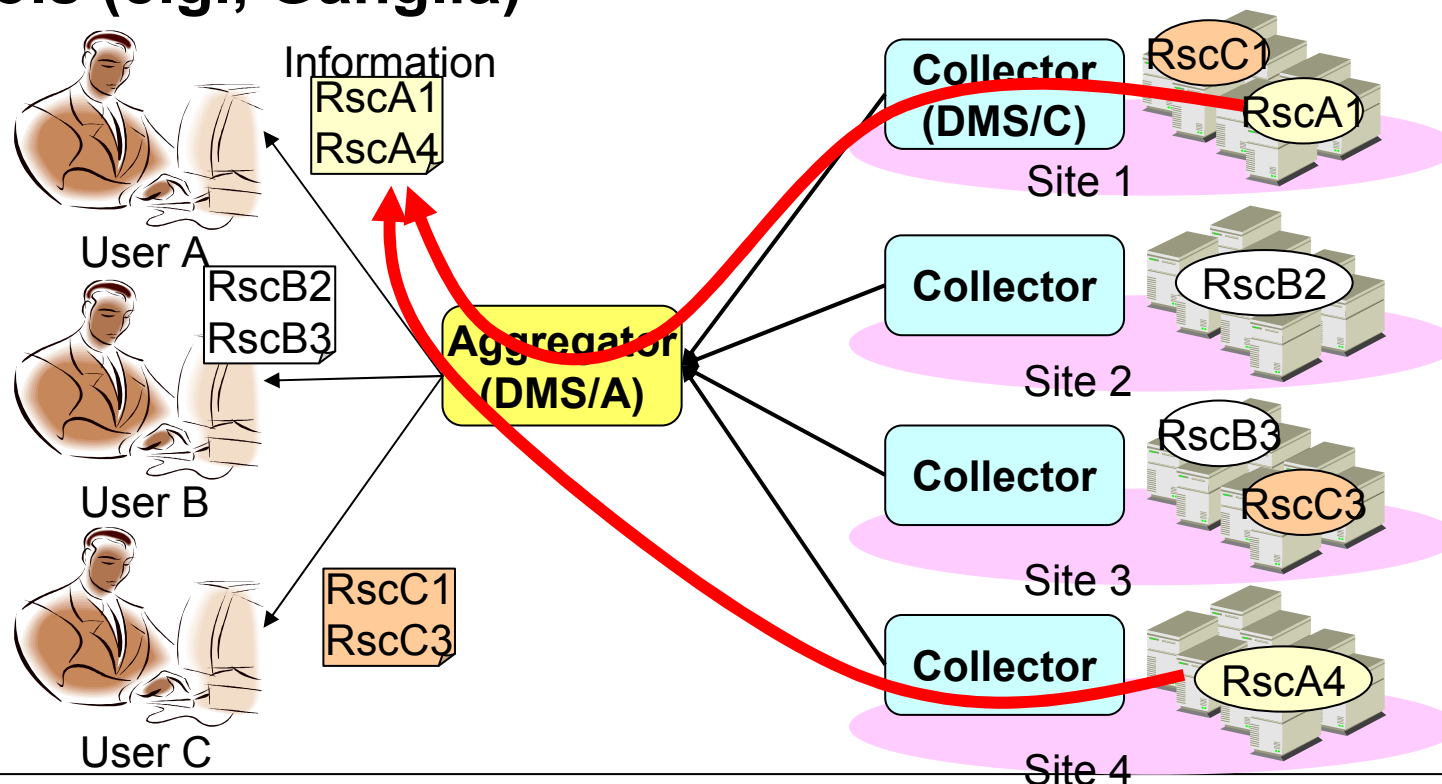
- **Distributed Monitoring System (DMS) for VCI**
 - DMS in cooperation with Resource Reservation based on the **GridARS** RMS (Resource Management System)
 - Fine grain authorization based on **XACML**
 - Hierarchical architecture by **Aggregator** and **Collector**
 - Aggregators gather monitoring information from related Aggregators and Collectors and provide requesters
 - Collectors monitor the VCL resources and filter the information by each domain policy
 - Standard interfaces and technologies
 - WSRF/GSI, XACML, GLUE, v. 2.0 and the extension
- **Confirm the feasibility of DMS**
 - DMS prototype developed using Globus Toolkit 4 and an XACML reference implementation by Sun Microsystems

Outline

- Background
- Design of Distributed Monitoring System (DMS)
 - System architecture
 - Cooperation with Resource Reservation
 - Fine grained authorization using XACML
 - Interface, protocol sequence, and data representations
 - Monitoring reserved resources
- DMS prototype and the experiments
 - XACML-based policy description
 - Overheads in WSRF implementation and authorization
- Related work
- Conclusions and future work

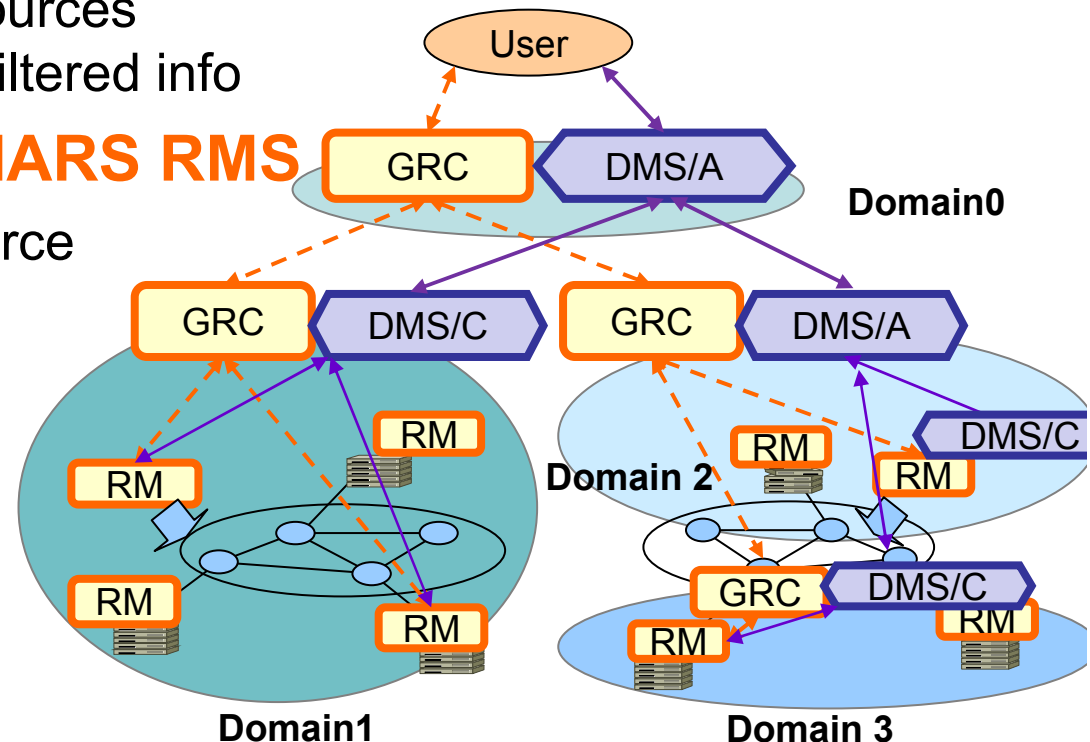
DMS Characteristics

- Monitoring information are filtered and managed by **each resource domain** ↔ **Central DB / central AuthZ**
- Each user get own VCI info and cannot get other user's task and resource information ↔ **General monitoring tools (e.g., Ganglia)**



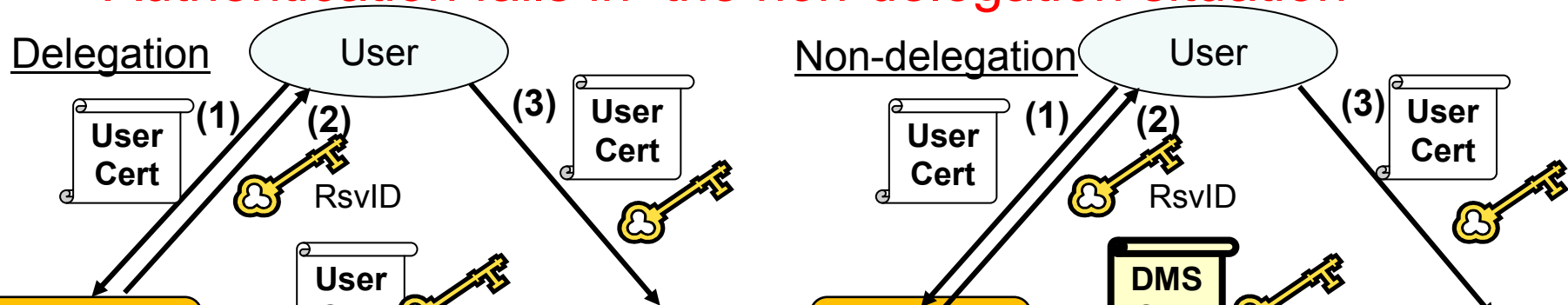
DMS System Architecture

- Hierarchical Architecture by **Aggregators (DMS/A)** and **Collectors (DMS/C)**
 - Common interface
 - DMS/A requests and gather monitoring info
 - DMS/C monitors user resources periodically and provides filtered info
- **DMS** Co-works with **GridARS RMS**
 - **RMS** consist of Grid resource coordinators (**GRC**) and computing / network resource managers (**RM**)
 - User requests resource co-allocation from RMS and the monitoring from DMS in Domain0



Cooperation with Resource Reservation and Authentication

- User requests resource co-allocation from GridARS RMS over WSRF/**GSI** and receives **reservation ID (RsvID)**, which allows to get the reservation information
 - Using the RsvID, User requests the monitoring from DMS
 - GSI (Grid Security Infrastructure), based on PKI, supports authentication and delegation
- **Authentication fails in the non-delegation situation**



RMS and DMS must be served on the same WS container
 or RMS and DMS must make some sort of agreement

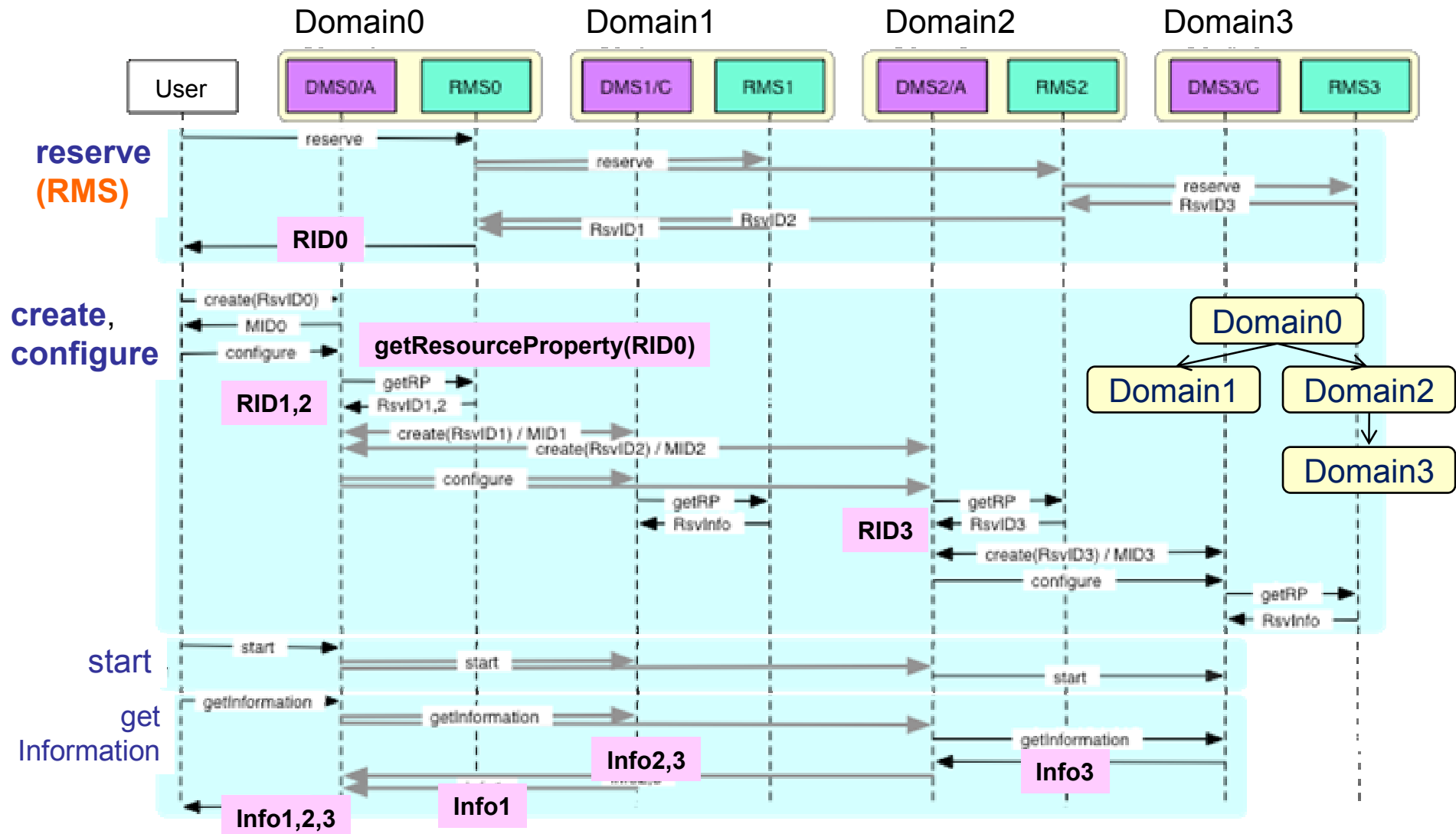
DMS Interface

Operation	Function	Input / Output
create	Start a monitoring process	RsvID by RMS / MID (Monitoring ID) by DMS
configure	Specify collecting information	MID, information list / -
start	Start monitoring at MA	MID(, time) / -
stop	Stop monitoring at MA	MID(, time) / -
getInformation	Get specified monitoring information	MID(, timeframe) / monitoring information

AuthZ using XACML

- Aggregators and Collectors use the same DMS I/F
- MID provided for each monitoring request
- Information list : list of target monitoring information items
- DMS supports subscribe/unsubscribe based on WS-Notification

DMS Protocol Sequence



Standard Interfaces and Technologies

- For Interoperability, DMS applies standards
- Standard data representation
 - **GLUE, v. 2.0** for resource monitoring information and the extensions for time series representation and network resources
 - **GNS-WSI (Grid Network Service - Web Services Interface)**, defined by the G-lambda project, for network resource information (→OGF NSI-WG)
- Standard technologies
 - WSRF / GSI
 - XACML

Example of ComputingActivity

```

<ComputingActivity>
  <ID>test_comp</ID><Name>test application</Name>
  <StartTime>2008-08-01T00:00:00Z</StartTime>
  <EndTime>2008-08-02T00:00:00Z</EndTime>
  <RequestedTotalWallTime>86400</RequestedTotalWallTime>
  <State>ACTIVATED</State>
  <RequestedSlots>2</RequestedSlots>
  <RequestedApplicationEnvironment>gridmpi</RequestedApplicationEnvironment>
  <Monitoring>
    <Timestamp>2008-08-01T01:00:00Z</Timestamp>
    <UsedTotalWallTime>3600</UsedTotalWallTime>
    <UsedTotalCPUTime>1800</UsedTotalCPUTime>
    <UsedMainMemory>1024</UsedMainMemory>
  </Monitoring>
  <Monitoring>...</Monitoring>
</ComputingActivity>
  
```

← Time series information (Extension)

Example of NetworkActivity

```

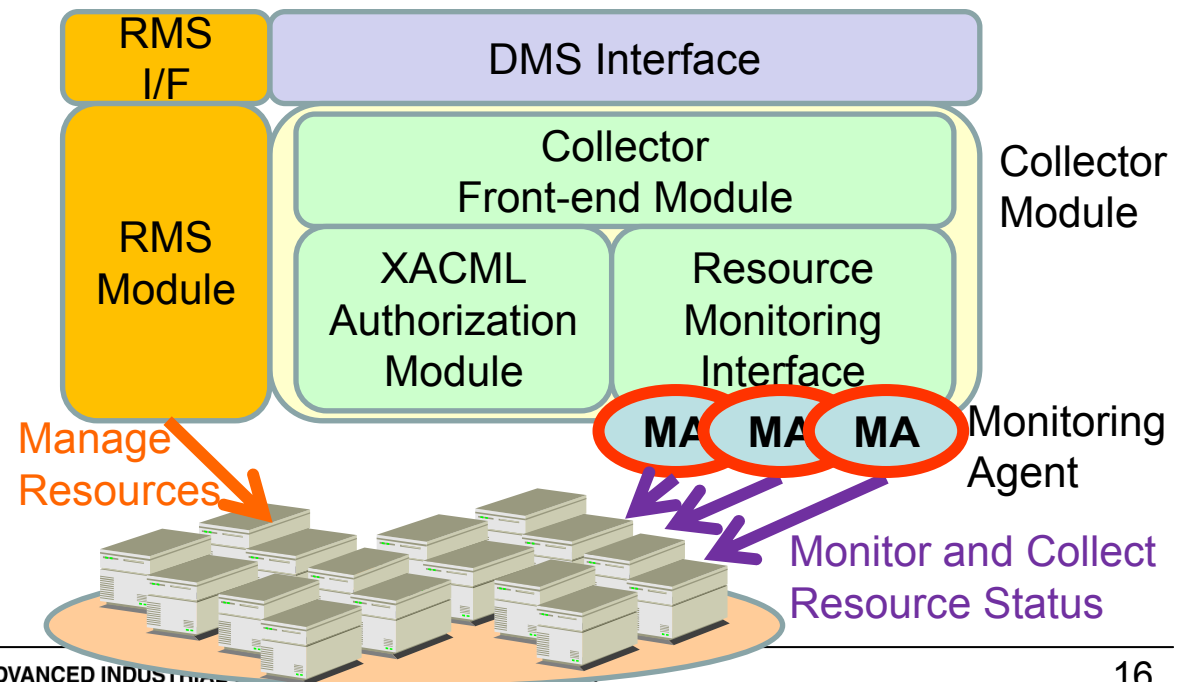
<NetworkActivity>
  <ID>test_net</ID>
  <StartTime>2008-08-01T00:00:00Z</StartTime>
  <EndTime>2008-08-02T00:00:00Z</EndTime>
  <RequestedTotalWallTime>86400</RequestedTotalWallTime>
  <State>ACTIVATED</State>
  <Path xmlns="http://www.glambda.net/schemas/gnswsi3/ndl-aist">
    <APoint><Name>AKB</Name></APoint>
    <ZPoint><Name>TKB</Name></ZPoint>
    <Bandwidth>1000</Bandwidth>
  </Path>
  <Monitoring stream="Down">
    <Timestamp>2008-08-01T01:00:00Z</Timestamp>
    <UsedTotalWallTime>3600</UsedTotalWallTime>
    <Transfer>65536</Transfer>
    <CurrentBandwidth>16</CurrentBandwidth><PeakBandwidth>256</PeakBandwidth>
    <Ping><Roundtrip>2.352</Roundtrip><TTL>64</TTL></Ping>
    <Jitter>0.1234</Jitter><Latency>3.456</Latency>
  </Monitoring>
</NetworkActivity>
  
```

← Network representation (GNS-WSI)

← Time series information (Extension)

Monitoring of Reserved Resources

- **MAs (Monitoring Agents) in Collectors**
 - Actual monitoring entity for each data item (e.g., CPU load, latency, network throughput)
 - The data items are filtered and provided the requester (Users or Aggregators)



Monitoring of Computing Resources

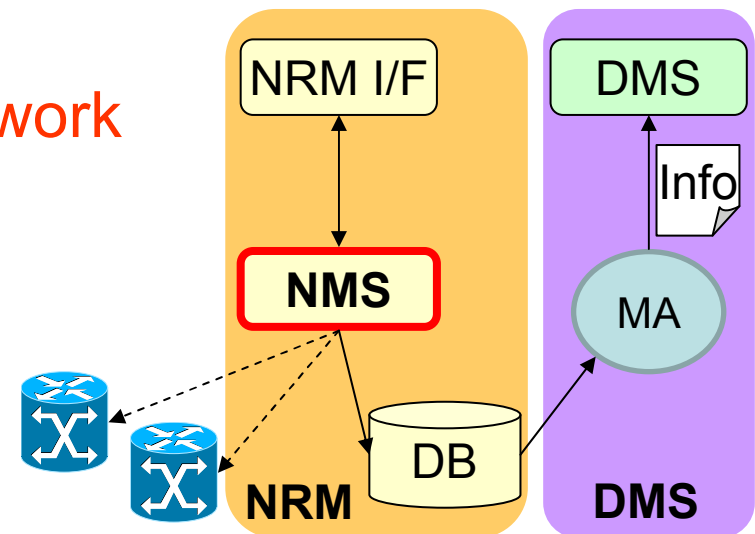
- Computing resource information
 - a. Reservation information
 - RsvID, # of CPUs, Memory size, start/end time, status
 - b. Reserved resource usage
 - CPU utilized time, memory and disk usage **for the reservation**
 - c. Information related to running applications
 - Execution logs of running applications registered in-advance
- Collection methods
 - a. Provided by CRM via RMS I/F using the RsvID
 - b. Collected via the process information I/F provided by each OS and via log files (e.g., proc and psacct)
 - c. Provided by the specified files for each application

Monitoring of Network Resources

- Network resource information
 - a. Reservation information
 - RsvID, endpoints, peak bandwidth, start/end time, status
 - b. Reserved resource (path network) usage
 - link status (Up/Down), packet statistical information
(Input/Output of transmission, destroy, and error packets)

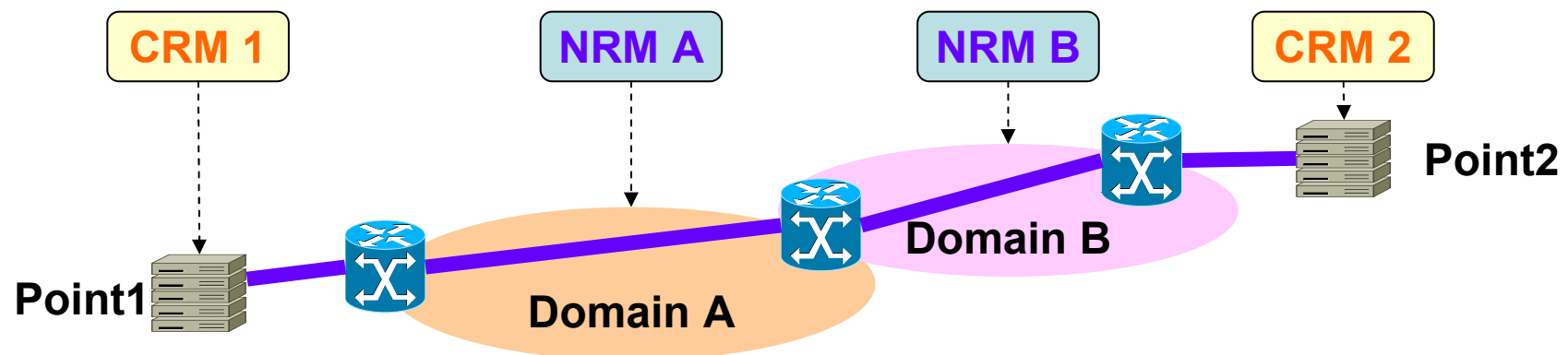
- Collection methods

- a, b are provided by **NMS (Network Management System)** in NRM
- **NMS** is managing paths and confirming the integrity
- NMS monitors the network I/Fs by using **SNMP** or **CLI**



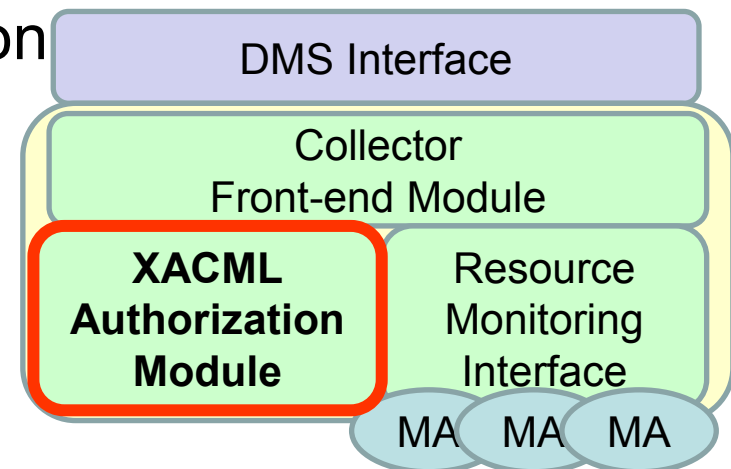
Monitoring of End-to-end Information

- NRM may not measure **end-to-end** throughput and delay
 - Termination points may belong to different domains
 - **CRMs collect the end-to-end information**
- End-to-end monitoring in CRM
 - The related CRMs exchange allocated computer information at the reservation start time
 - Each CRM collects end-to-end information between its node and other node by using “ping”



Collectors' Fine Grained Authorization using XACML

- Collectors authorize each user access to monitoring information by using **XACML**
 - e**X**tensible **A**ccess **C**ontrol **M**arkup **L**anguage
- **XACML**
 - Defines fine grained control of authorized activities and a policy AuthZ model
 - Enables flexible policy description



A Policy Example: allow access within a specified time

```

<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  PolicyId="TimeRangePolicy1"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Description> Between 9am and 11pm local time, allow anyone to access. </Description>
  <Target>
    <Subjects><AnySubject/></Subjects>
    <Resources><AnyResource/></Resources>
    <Actions><AnyAction/></Actions>
  </Target>
  <Rule RuleId="PermitDuringSpecifiedTimeRange" Effect="Permit">
    <Condition
      FunctionId="http://gridars.aist.go.jp/dms/Collector/XACML/TimeFunction">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#time">09:00:00</AttributeValue>
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#time">23:00:00</AttributeValue>
      </Condition>
    </Rule>
    <Rule RuleId="DenyAllOthers" Effect="Deny"/>
  </Policy>

```

A Policy Set Example

```

<PolicySet PolicySetId="PolicySet1"
  PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
    combining-algorithm:deny-overrides">
  <Description> Example policy set. </Description>
  <Target>
    <Resources><AnyResource/></Resources>
  </Target>
  <PolicyIdReference>DNMatchPolicy1</PolicyIdReference>
  <PolicyIdReference>TimeRangePolicy1</PolicyIdReference>
</PolicySet>

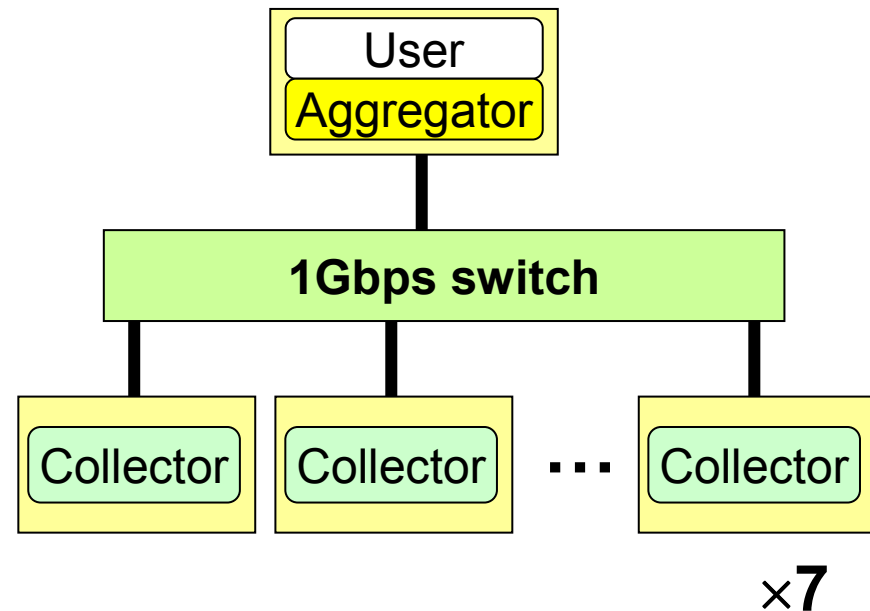
```

DMS Prototype and Experiments

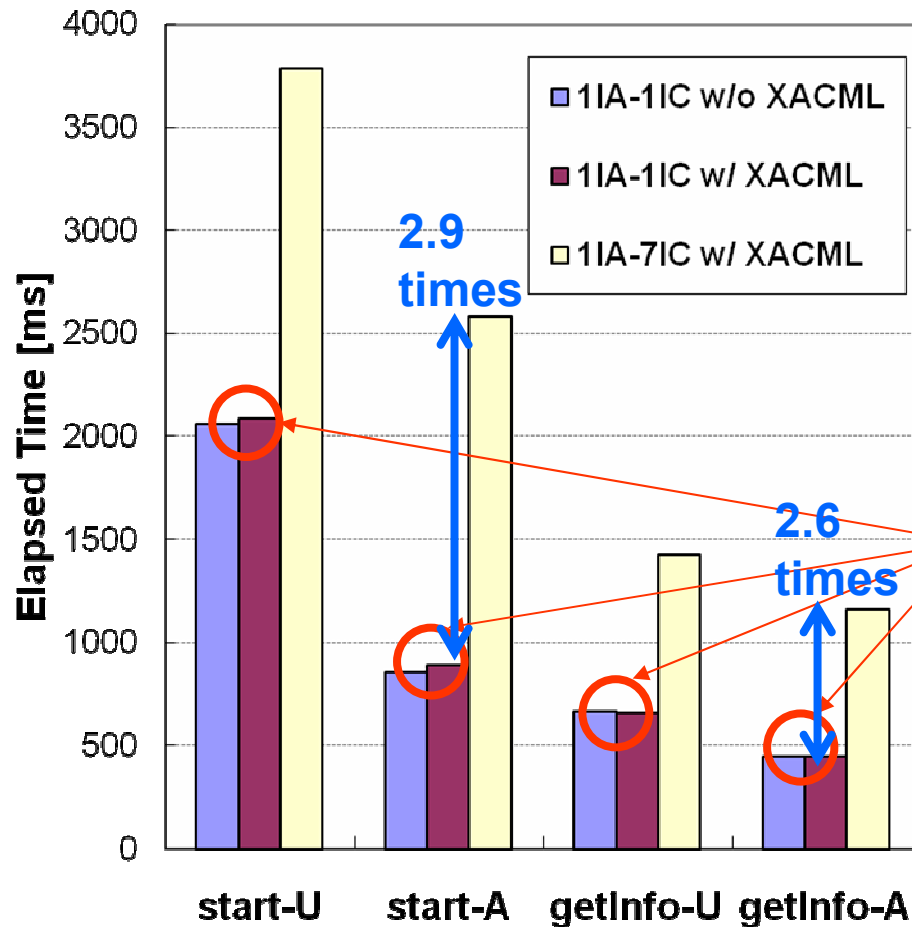
- DMS prototype has developed using
 - [Globus Toolkit 4](#) (WSRF)
 - GSI (Grid Security Infrastructure) - AuthN, SSL, grid-mapfile
 - [Sun Microsystems' Java-based reference implementation](#) (XACML)
- Experiments using the DMS prototype
 - Experiment 1: WSRF/GSI overheads in getting information
 - Experiment 2: XACML authorization overheads

Experimental Environment

- All Nodes:
 - Pentium 4 2.8GHz, 1GB Memory, CentOS4.3
 - Latencies: 40 [us]
- Applied a policy set includes:
 - If request issued in the specified time frame or not
 - If DN in user cert is in grid-mapfile or not
- Collected information
 - `$ ps ux -cols 80 -u <user>`



Experiment 1: WSRF/GSI Overheads

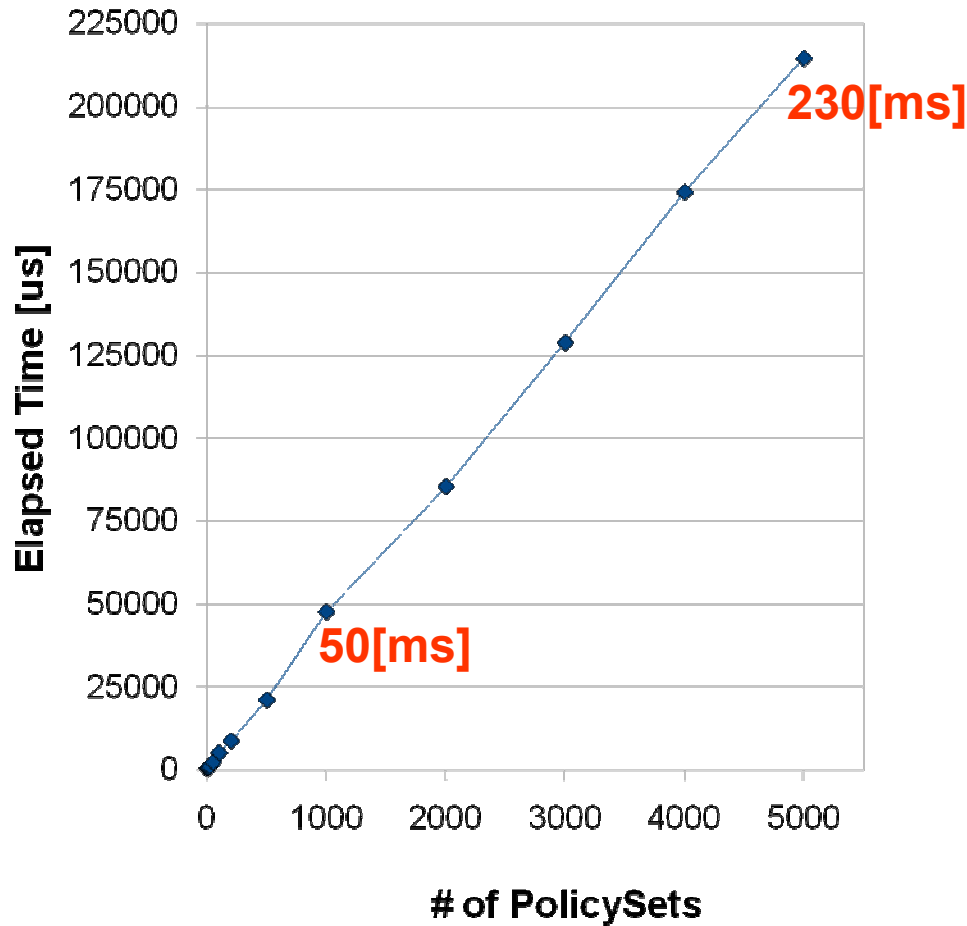


- Notation

- start : configure and start
- getInfo : getInformation
- -U: from user
- -A: from aggregator

- Comparison of 1A-1C w/ and w/o XACML shows less AuthZ overheads
- The ratios of 1A-7C and 1A-1C remain 2.6-2.9 times because of concurrent operations from Aggregator → Response times are not proportional

Experiment 2: XACML Overheads



- Notation
 - Elapsed times of XACML AuthZ decision
 - 1 PolicySet includes 2 policies
 - Not include policy reading time and Java GC time
- AuthZ overheads is approximately proportional
- The overhead is 230 ms for 5000 set
 → XACML overhead is negligible over total time

Related Work (1/2)

- Globus Toolkit, v. 4.2
 - Provides XACML-based general AuthZ framework
 - AuthZ granularity is “service” (↔ “each info item”)
- MonALISA [CHEP2004]
 - Stores a central registry with monitoring information collected by general tools (e.g., Ganglia and MRTG)
 - Does not authorize access to the information
- Inca [SC2004]
 - Grid monitoring system for TeraGrid administrators
- MonALISA and Inca gather and manage the information from multiple domains in a central DB

Related Work (2/2)

- WMSMonitor [Grid2008]
 - Monitoring tool for workload and job lifecycles for EGEE gLite
 - Supports requirements of various user categories
- AMon [Grid2008]
 - Monitoring system for HEPCG (High Energy Physics Community Grid) in D-Grid
 - Collects status, resource usage, and output of the applications
- WMSMonitor and AMon do NOT support AuthZ in each resource management domain
- None of the related systems deal with VCI, reserved in-advance and provisioned dynamically

Conclusions

- **Distributed Monitoring System (DMS) for VCI**
 - DMS in cooperation with Resource Reservation based on the **GridARS** RMS
 - Fine grain authorization based on **XACML**
 - Hierarchical architecture by **Aggregator** and **Collector**
 - Standard interfaces and technologies
 - WSRF/GSI, XACML, GLUE, v. 2.0 and the extension
- **Confirm the feasibility of DMS**
 - Developed a DMS prototype
 - Response times over WSRF/GSI are acceptable and AuthZ overheads are negligible

Future Work

- Continue to develop the DMS system based on the proposed design
- Develop a graphical user interface
- Apply a variety of access policies by XACML
 - provide basic set of policies for domain administrators

Acknowledgements

This work was partly funded by the National Institute of Information and Communications Technology.