# EGEE - NAREGI inter-operation Information and Monitoring Service

National Institute of Informatics

& Hitachi, Ltd.

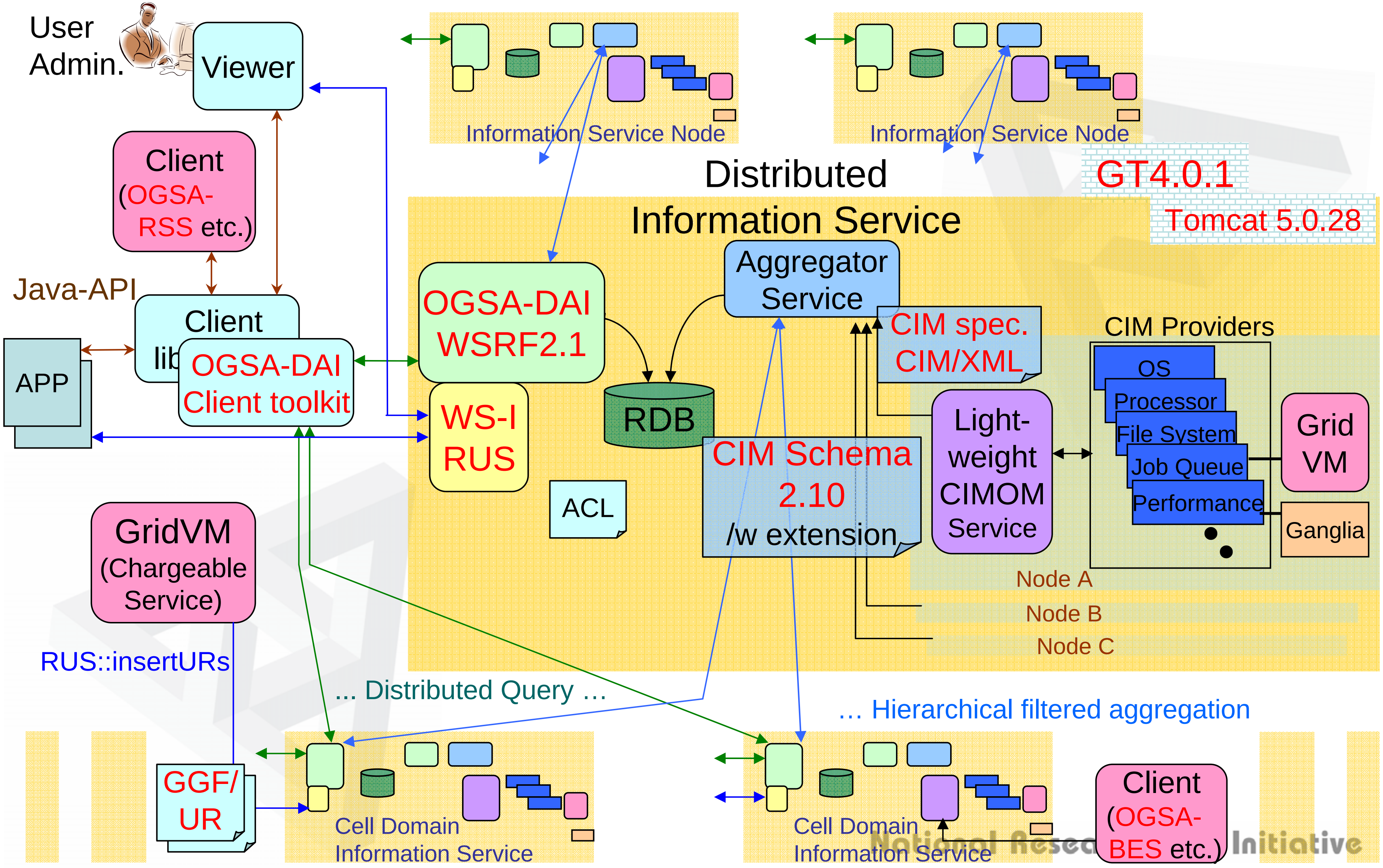
2006/03/21

National Research Grid Initiative

# Information Service Characteristics

- Basic syntax:
  - Resource description schemas (e.g., GLUE, CIM)
  - Data representations (e.g., XML, LDIF)
  - Query languages (e.g., SQL, XPath)
  - Client query interfaces (e.g., WS Resource Properties queries, LDAP, OGSA-DAI)
- Semantics:
  - What pieces of data are needed by each Grid (various previous works & actual deployment experiences already)
- Implementation:
  - Information service software systems (e.g., MDS, BDII)
  - The ultimate sources of this information (e.g., PBS, Condor, Ganglia, WS-GRAM, GridVM, various grid monitoring systems, etc.).

# NAREGI Information Service



User
Admin.

Viewer

Client
(OGSA-
RSS etc.)

Java-API

APP

Client
lib

OGSA-DAI
Client toolkit

GridVM
(Chargeable
Service)

RUS::insertURs

GGF/
UR

Information Service Node

Information Service Node

Distributed
Information Service

GT4.0.1

Tomcat 5.0.28

Aggregator
Service

OGSA-DAI
WSRF2.1

WS-I
RUS

RDB

ACL

CIM spec.
CIM/XML

CIM Schema
2.10
/w extension

Light-
weight
CIMOM
Service

CIM Providers

OS

Processor

File System

Job Queue

Performance

Grid
VM

Ganglia

Node A

Node B

Node C

... Distributed Query ...

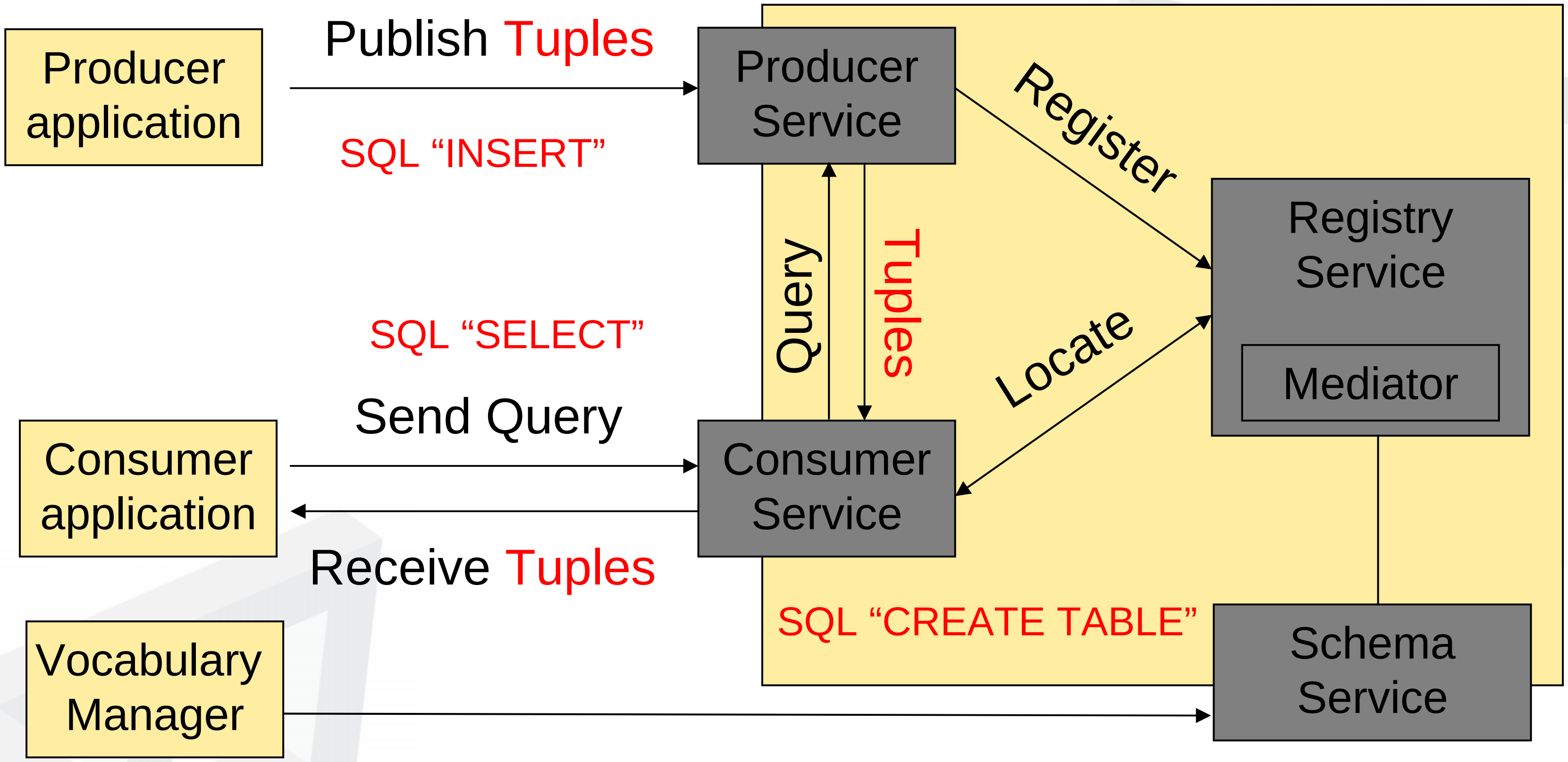
… Hierarchical filtered aggregation

Cell Domain
Information Service

Cell Domain
Information Service

Client
(OGSA-
BES etc.)

# Relational Grid Monitoring Architecture



- An implementation of the GGF Grid Monitoring Architecture (GMA)
- All data modelled as tables: a single schema gives the impression of one virtual database for VO

# Syntax Interoperability Matrix

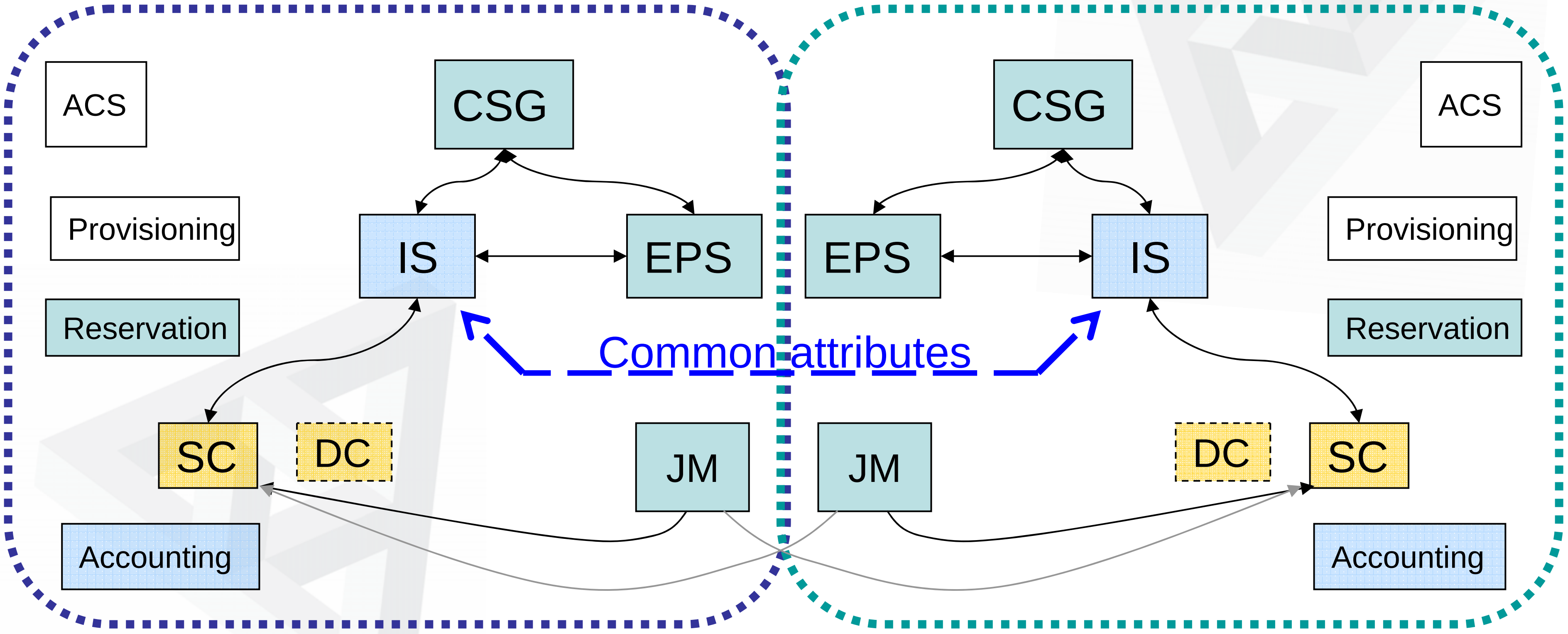
| Grid | Schema | Data | Query Lang | Client IF | Software |
|---|---|---|---|---|---|
| Tera-Grid | GLUE | XML | XPath | WSRF RP Queries | MDS4 |
| OSG | GLUE | LDIF | LDAP | LDAP | BDII |
| NAREGI | CIM 2.10+ext | Relational | SQL | OGSA-DAI<br>WS-I RUS | CIMOM +<br>OGSA-DAI |
| EGEE/ LCG | GLUE | LDIF | LDAP | LDAP | BDII |
| | | Relational | SQL | R-GMA i/f | R-GMA |
| NorduGrid | ARC | LDIF | LDAP | LDAP | GIIS |

# Low Hanging Fruit
# “Just make it work by GLUEing”

- Identify the minimum common set of information required for interoperation in the respective information service
- Employ GLUE and extended CIM as the base schema for respective grids
- Each Info service in grid acts as a information provider for the other
- Embed schema translator to perform schema conversion
- Present data in a common fashion on each grid ; WebMDS, NAREGI CIM Viewer, SCMSWeb, …

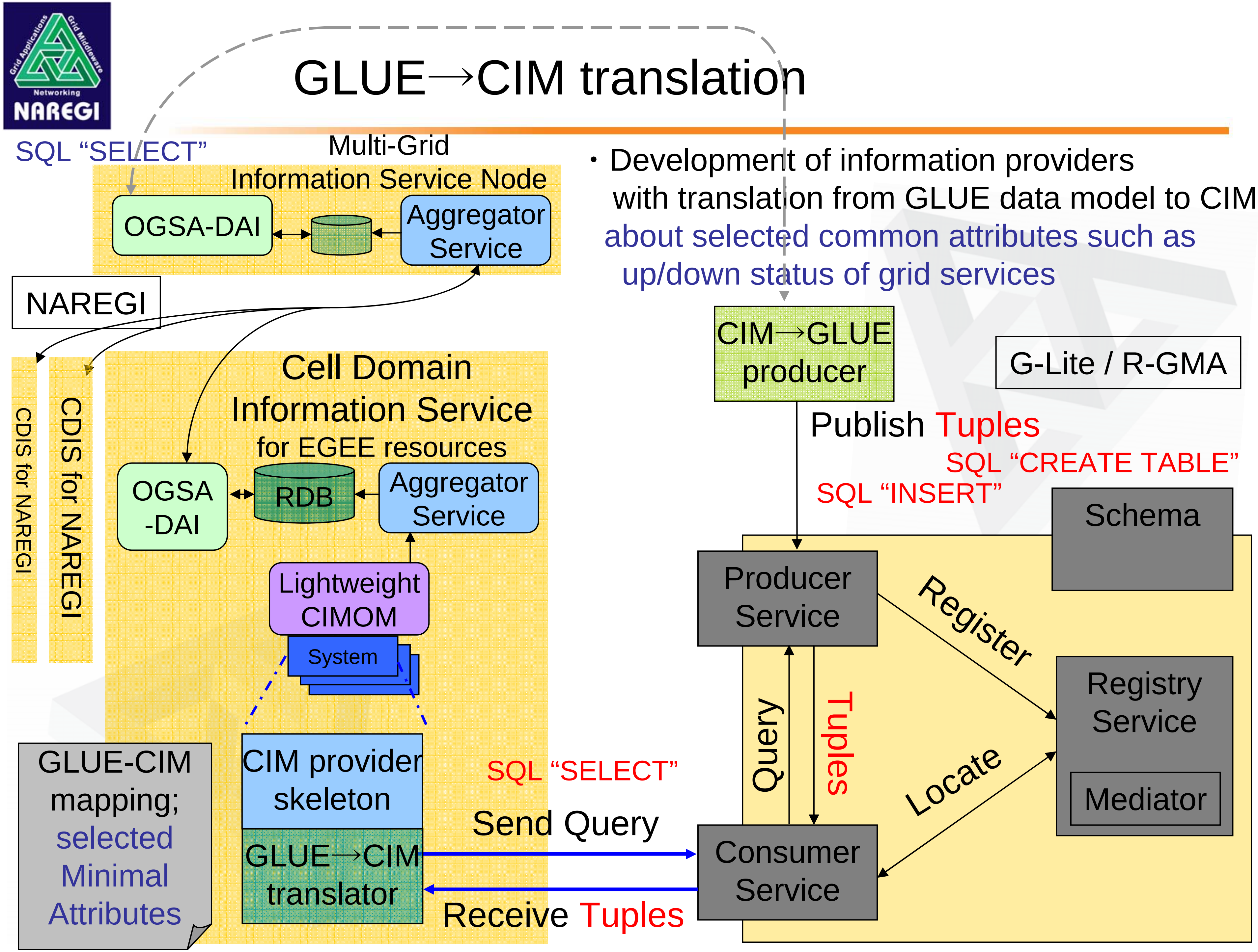
# Minimal Common Attributes

- Define minimal common set of attributes required
- Each system components in the grid will only access the translated information



# GLUE→CIM translation

- Development of information providers with translation from GLUE data model to CIM about selected common attributes such as up/down status of grid services

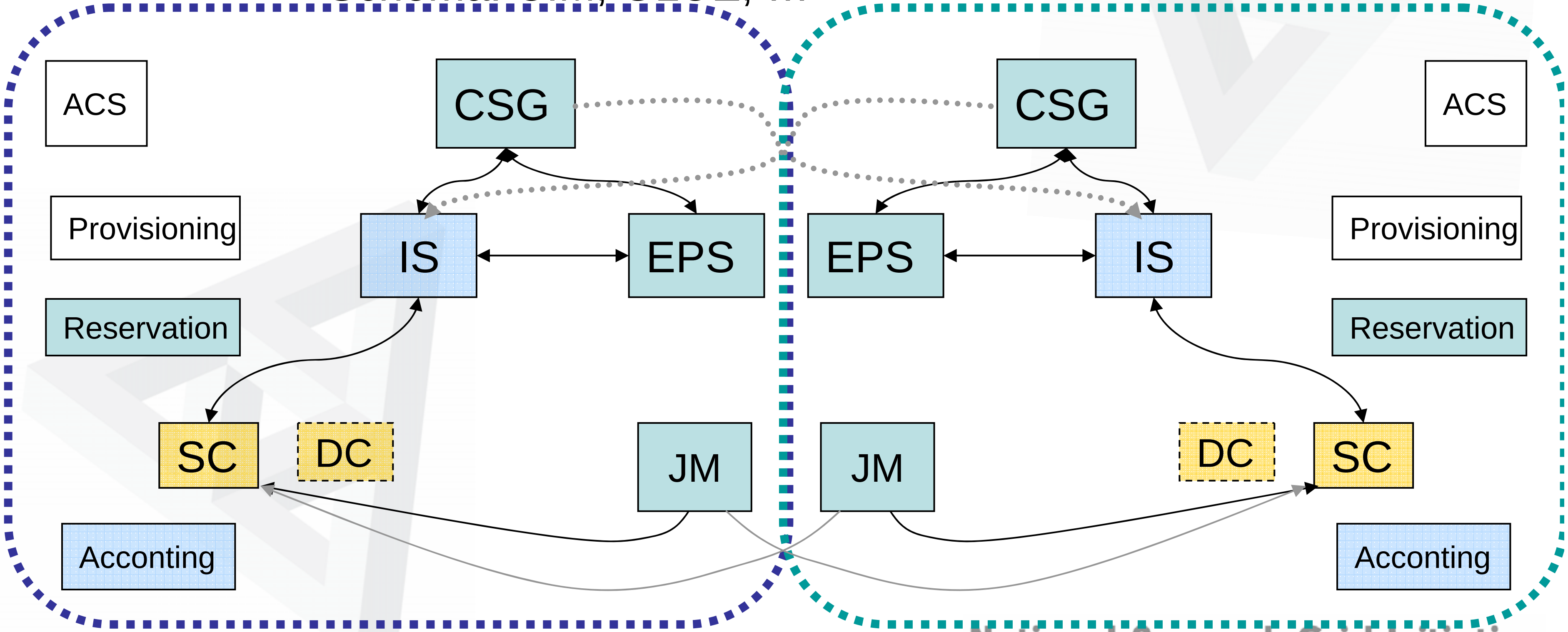


# Interface level adaptation … in long term

When the CSG accesses multi-Grid Information Services for resource discovery

CSG has to know the differences in the consumer interfaces

- Interfaces: Subscription / Query, push/pull
- Query Language
- Data format: XML, …
- Schema: CIM, GLUE, …



# NAREGI Information Service

1. Overview

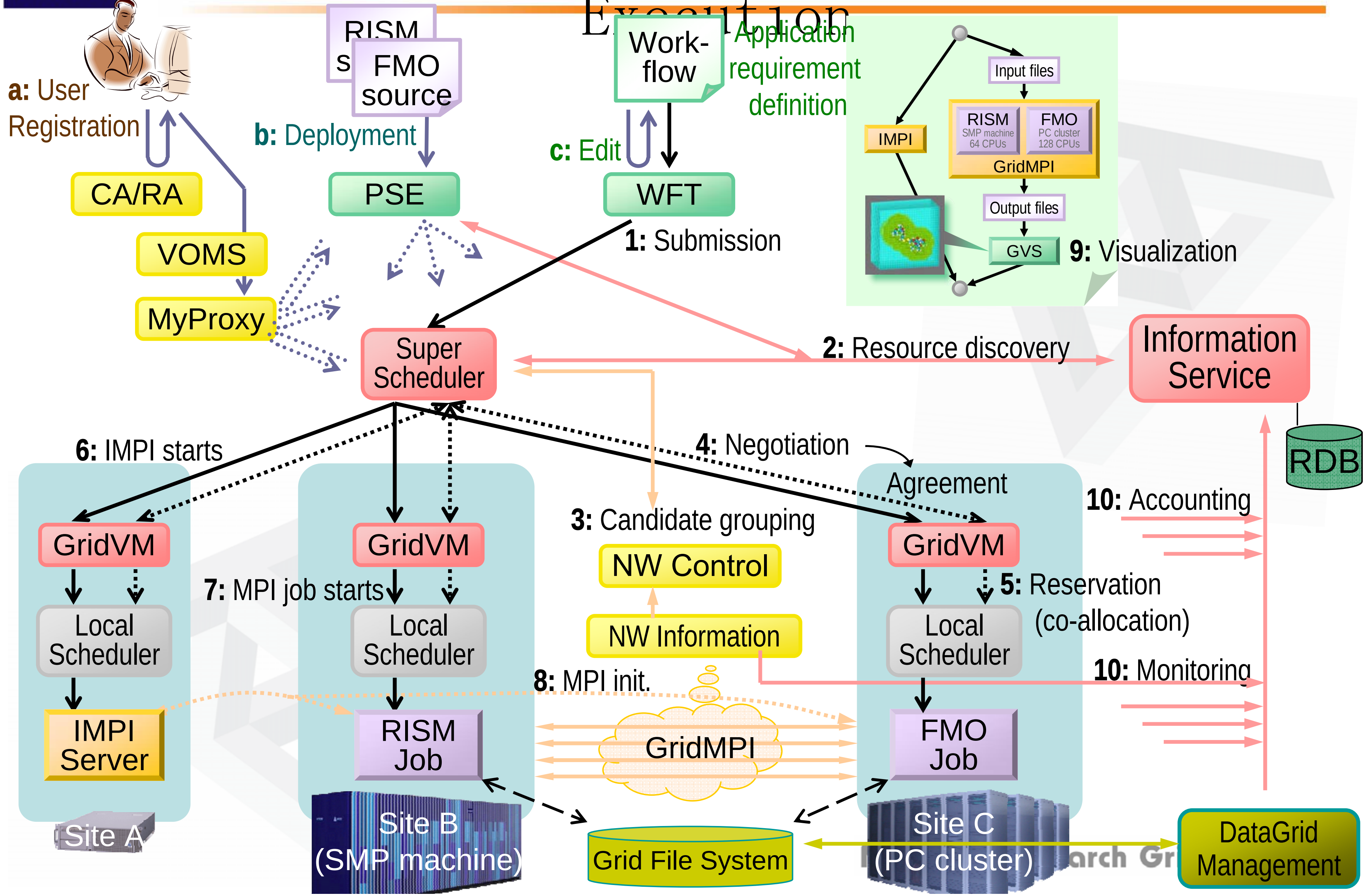
2. Resource information schema

3. Publisher interface

4. Consumer interface

5. VO information service

# Scenario for Inter-site MPI Job Execution



a: User Registration

CA/RA

VOMS

MyProxy

RISM source

FMO source

b: Deployment

PSE

Work-flow

c: Edit

WFT

Application requirement definition

Input files

IMPI

RISM
SMP machine
64 CPUs

FMO
PC cluster
128 CPUs

GridMPI

Output files

GVS

9: Visualization

1: Submission

Super Scheduler

2: Resource discovery

Information Service

RDB

6: IMPI starts

4: Negotiation

Agreement

3: Candidate grouping

NW Control

NW Information

10: Accounting

GridVM

GridVM

GridVM

7: MPI job starts

5: Reservation (co-allocation)

Local Scheduler

Local Scheduler

Local Scheduler

8: MPI init.

10: Monitoring

IMPI Server

RISM Job

GridMPI

FMO Job

Site A

Site B (SMP machine)

Grid File System

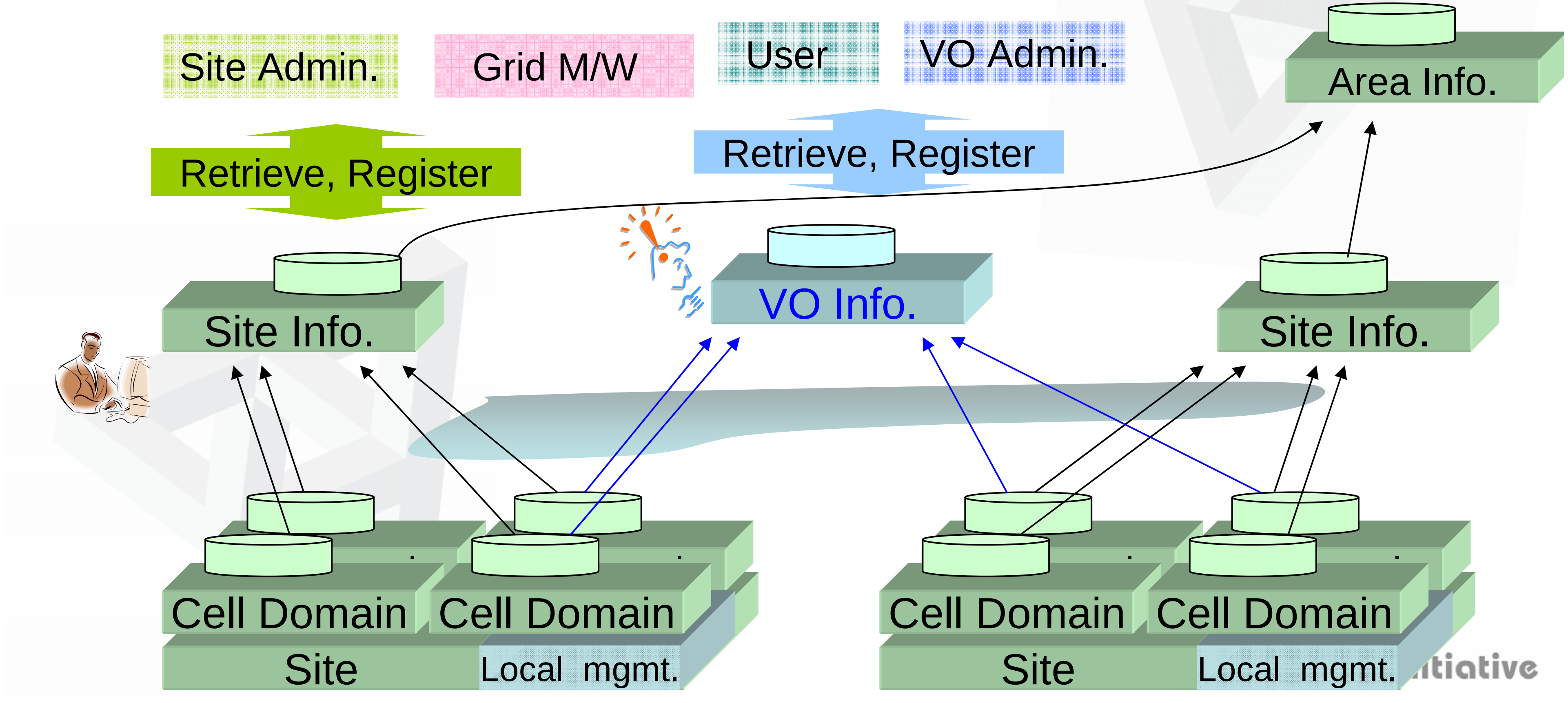
Site C (PC cluster)

DataGrid Management

# Distributed Information Service

Distributed Info.Services maintain various kind of information across multiple administrative domains and VOs.

Clients can search useful information to help Resource Broker for job execution, VO management, etc.



# Functions

■ Discovery

Aggregated resource information is accumulated to RDB (PostgreSQL), Resource can be discovered by SQL query.

■ Monitoring

Information of Job Queue and local scheduler managed by GridVM is served.
Utilization of existing monitoring systems ; e.g. Ganglia.

■ Accounting

Usage Records provided by GridVMs are collected and maintained.
Users can search and summarize their records by global id,
even if their jobs are executed across multiple sites.

■ Logging

Job information and Syslog are monitored and accumulated and support the cause investigation of abnormal phenomena/activities.

■ Registry

PSE registers application information and deployment information.
NAREGI M/W components register information of their service access points.

■ VO Management

Information Service for each VO.

# Features

## Support for Resource Brokering and Accounting …

{ GridVMs provide information about Job Queue and Job Usage.
Resource Brokers consume the information using SQL query.

○ General schema (based on CIM) for resource description.
→ can satisfy requirement of other middleware.
can include existing / standard schema.
(⊃ GGF / JSIM, UR Schema)

○ Aggregated CIM objects are accumulated to RDB.
→ Resource discovery by using SQL query,
Analysis of time-series data.

| CIM | RDB |
|---|---|
| Class | Table |
| Instance | Record (row) |
| Property | Field (column) |

○ Implemented as secure Grid Service.
(on GT3→GT4 with OGSA-DAI, RUS)

○ Hierarchical access to distributed large DB.

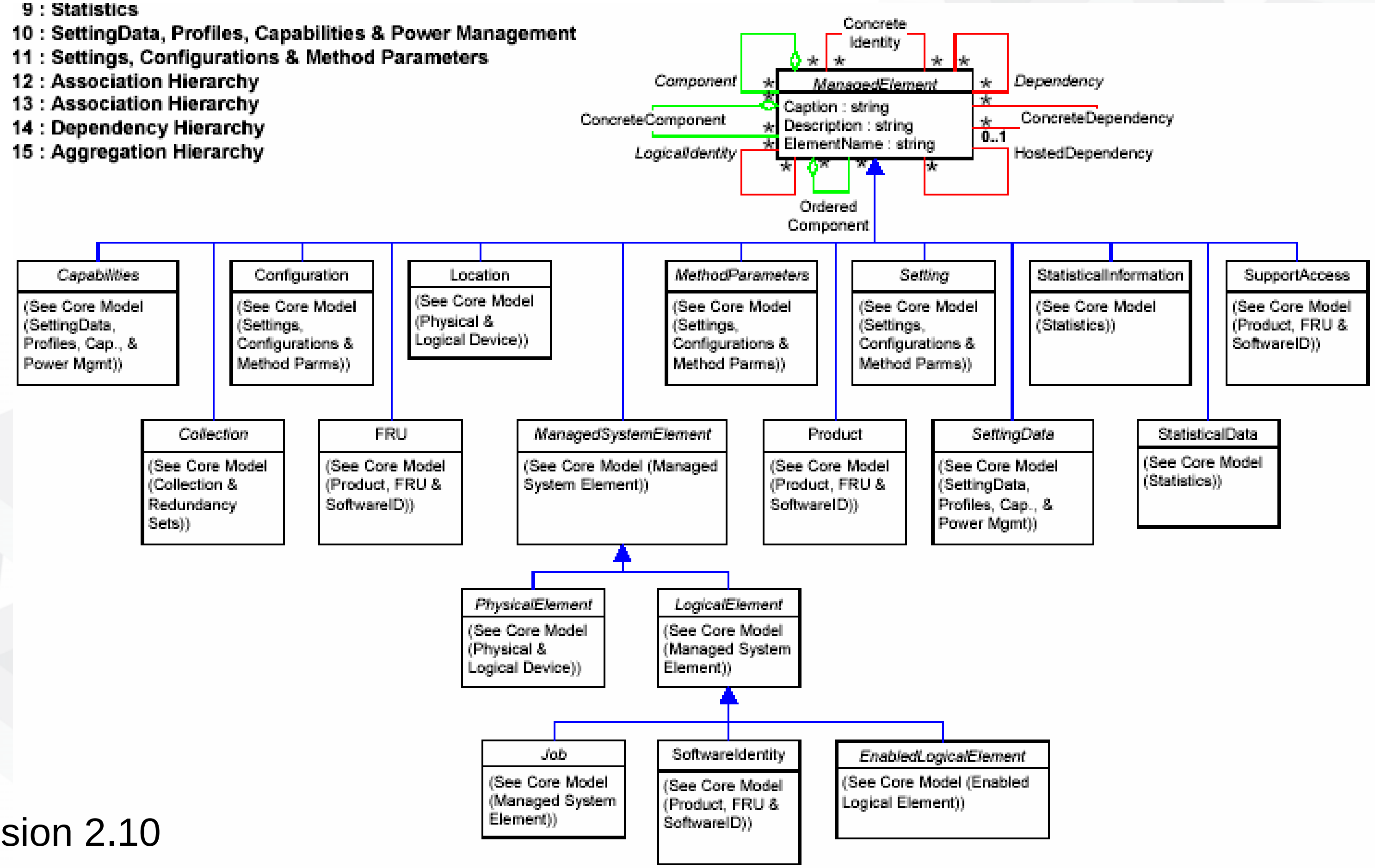
○ Support for VO management

# Common Information Model

DMTF(Distributed Management Task Force) formulates

CIM Schema : Abstract object-oriented model very widely about the administrativ information of the computers and has over thousand classes.

WBEM : Interface to access to administrative information.



9 : Statistics
10 : SettingData, Profiles, Capabilities & Power Management
11 : Settings, Configurations & Method Parameters
12 : Association Hierarchy
13 : Association Hierarchy
14 : Dependency Hierarchy
15 : Aggregation Hierarchy

ManagedElement
Caption : string
Description : string
ElementName : string

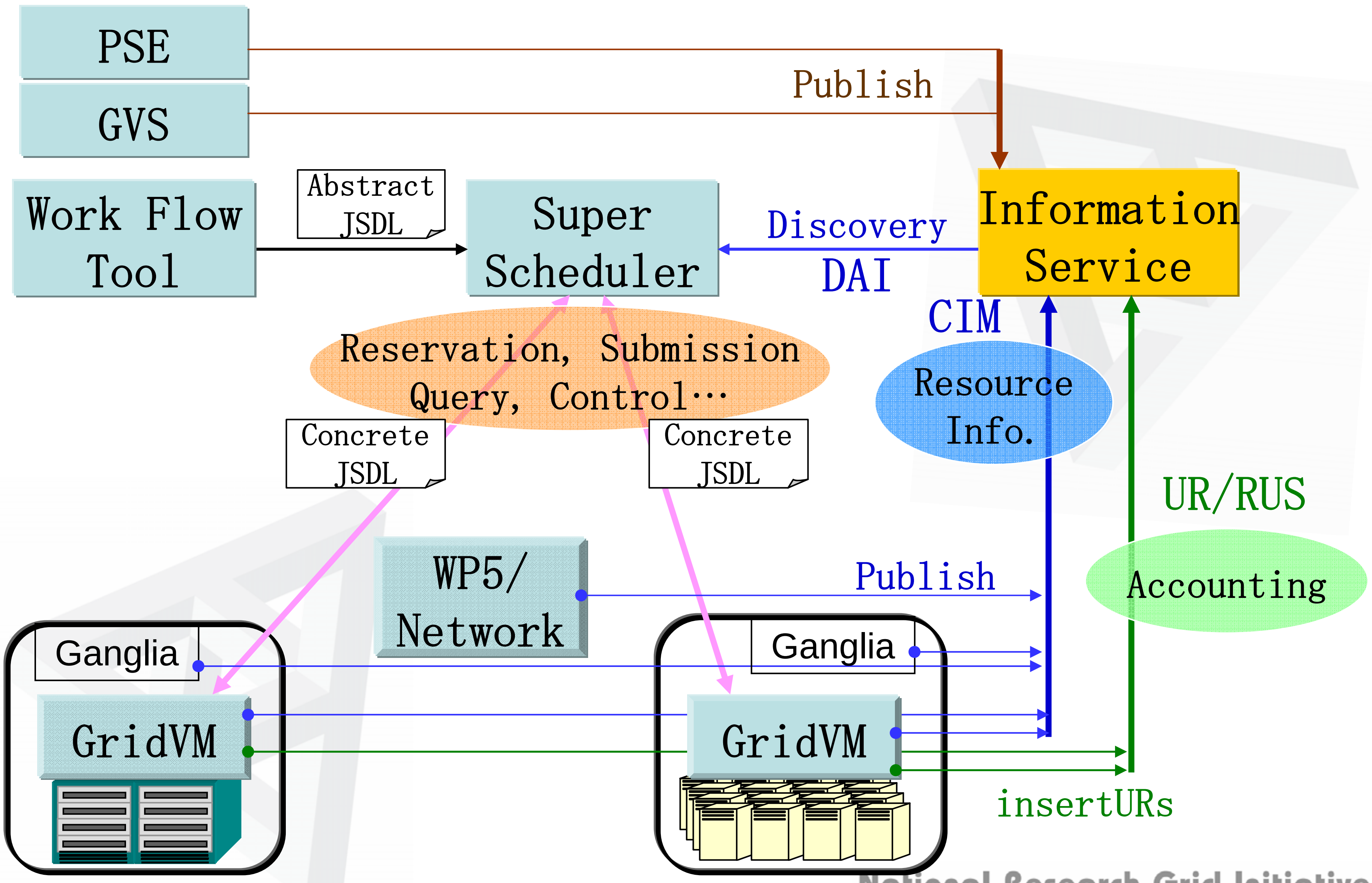
Component
ConcreteComponent
LogicalIdentity
Concrete Identity
Dependency
ConcreteDependency
HostedDependency
Ordered Component
0..1

Capabilities (See Core Model (SettingData, Profiles, Cap., & Power Mgmt))
Configuration (See Core Model (Settings, Configurations & Method Parms))
Location (See Core Model (Physical & Logical Device))
MethodParameters (See Core Model (Settings, Configurations & Method Parms))
Setting (See Core Model (Settings, Configurations & Method Parms))
StatisticalInformation (See Core Model (Statistics))
SupportAccess (See Core Model (Product, FRU & SoftwareID))
Collection (See Core Model (Collection & Redundancy Sets))
FRU (See Core Model (Product, FRU & SoftwareID))
ManagedSystemElement (See Core Model (Managed System Element))
Product (See Core Model (Product, FRU & SoftwareID))
SettingData (See Core Model (SettingData, Profiles, Cap., & Power Mgmt))
StatisticalData (See Core Model (Statistics))
PhysicalElement (See Core Model (Physical & Logical Device))
LogicalElement (See Core Model (Managed System Element))
Job (See Core Model (Managed System Element))
SoftwareIdentity (See Core Model (Product, FRU & SoftwareID))
EnabledLogicalElement (See Core Model (Enabled Logical Element))

www.dmtf.org

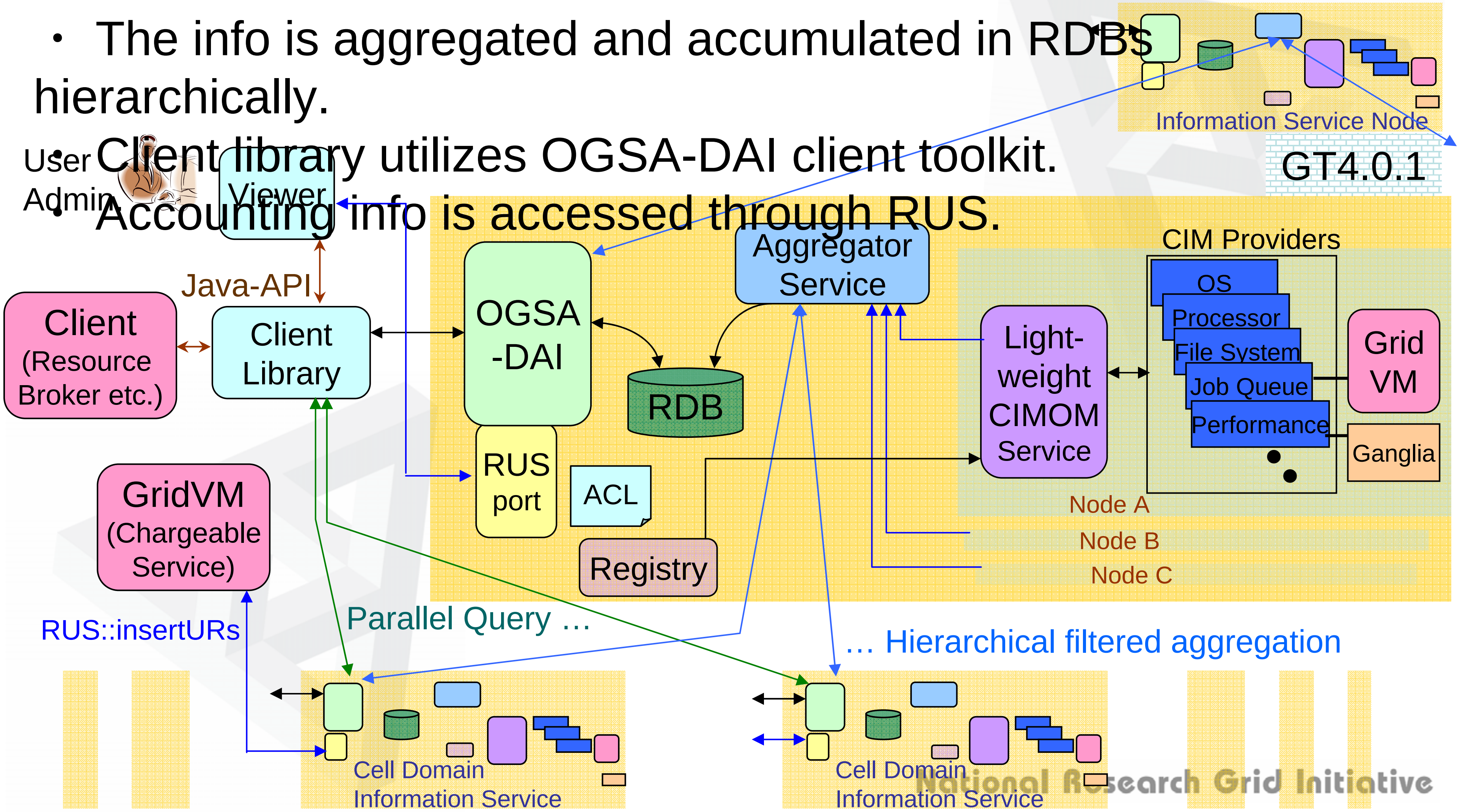
CIM Schema: Version 2.10

# Relationship

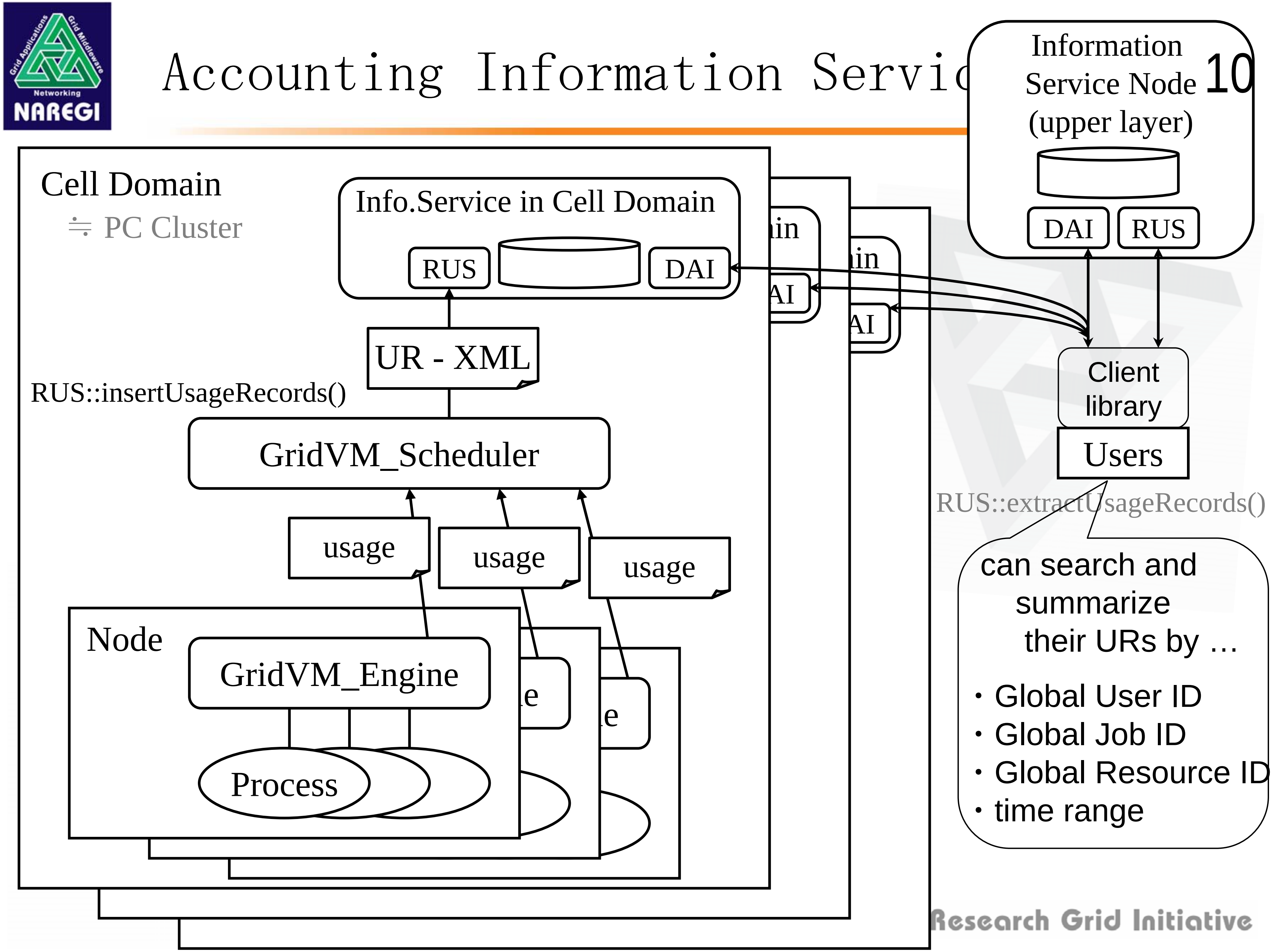


# Internal Structure

- CIMOM Service classifies info according to CIM based schema.
- The info is aggregated and accumulated in RDBs hierarchically.
- Client library utilizes OGSA-DAI client toolkit.
- Accounting info is accessed through RUS.



# Accounting Information Servic



Information Service Node (upper layer)

DAI

RUS

Cell Domain

≒ PC Cluster

Info.Service in Cell Domain

RUS

DAI

ain

AI

ain

AI

UR - XML

RUS::insertUsageRecords()

GridVM_Scheduler

usage

usage

usage

Node

GridVM_Engine

e

e

Process

Client library

Users

RUS::extractUsageRecords()

can search and summarize their URs by …

- Global User ID
- Global Job ID
- Global Resource ID
- time range

1. Overview

2. Resource information schema

3. Publisher interface

4. Consumer interface

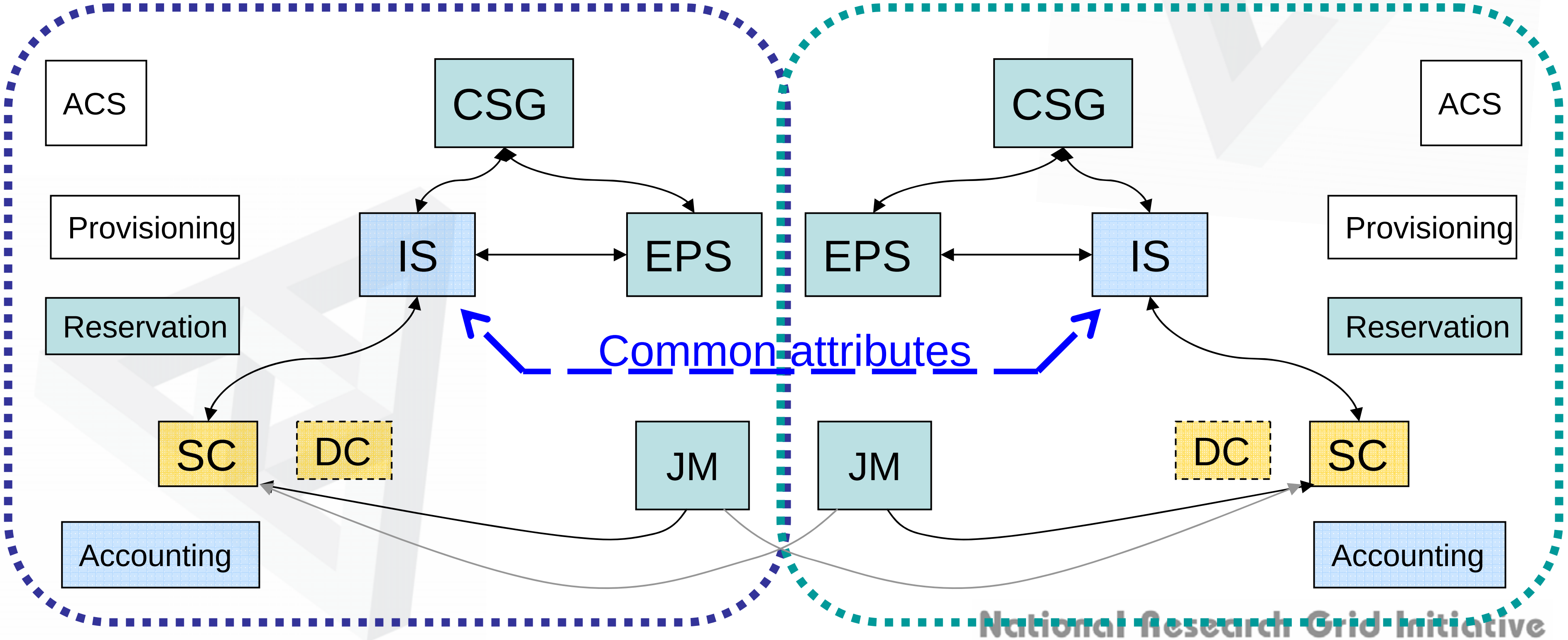
5. VO information service

# Minimal Common Attributes

In case information services share multi-Grid resource information,

Information services have to maintain common attributes for CSG to generate Candidate Sets.

What attributes should be common?



# Candidates

## Service

Type : [pre]ws-gram-pbs, LRMS, Scheduler, GridFTP, RFT, MDS4/IS, RLS, SRB, etc
Version : e.g. 4.0.1
Host : e.g. tg-grid1.uc.teragrid.org
Port : e.g. 2119
Path : e.g. /jobmanager-pbs
URL : e.g. https://png1037.naregi.org:9000/wsrf/services/gridvm/GridVMJobFactoryService
Status : e.g. enabled
VO/group/role to be authorized

other candidates : Functionality, Outage start/end

## Software

Package name : Runtime environment, MPI
Version
Description

other candidates :

# Candidates

## Queue

Name, Unique ID
Number of CPUs {Total, Free}
Status
Number of jobs {Total, Running, Waiting}
Policy : Max {Wall time, CPU time, Total jobs, Running jobs}
VO/group/role to be authorized

other candidates : Estimated traversal time

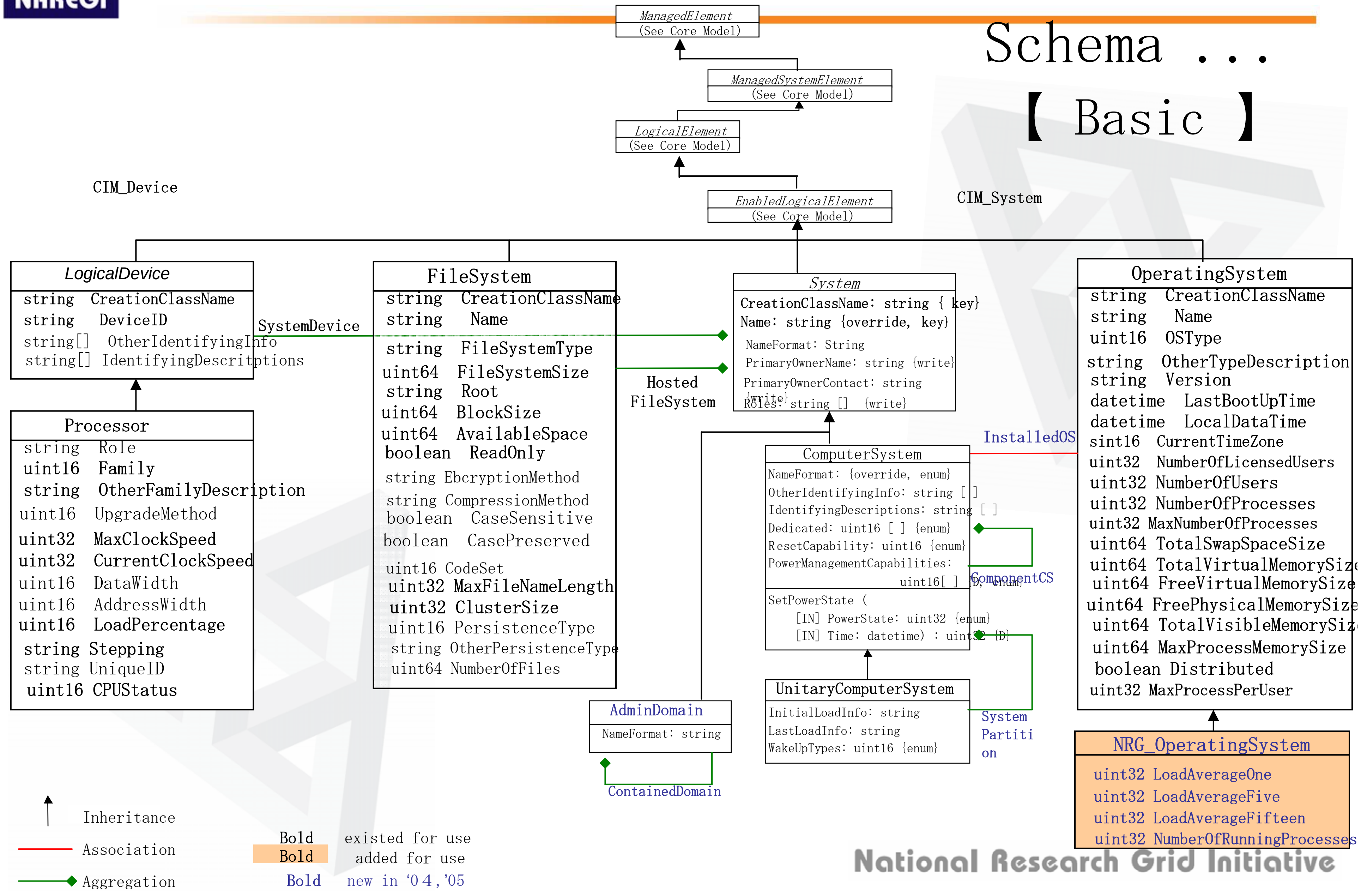
## Cluster ～ Host

Type : heterogeneous / homogeneous
Name, Unique ID
Total nodes
Storage device name
size
available space
type

Host name, unique ID
Processor type
speed
Total memory
Operating system
SMP size

other candidates : accepted CA

# Information Model based on CIM Schema ... 【 Basic 】



ManagedElement
(See Core Model)

ManagedSystemElement
(See Core Model)

LogicalElement
(See Core Model)

EnabledLogicalElement
(See Core Model)

CIM_Device

CIM_System

**LogicalDevice**
string CreationClassName
string DeviceID
string[] OtherIdentifyingInfo
string[] IdentifyingDescritptions

SystemDevice

**Processor**
string Role
uint16 Family
string OtherFamilyDescription
uint16 UpgradeMethod
uint32 MaxClockSpeed
uint32 CurrentClockSpeed
uint16 DataWidth
uint16 AddressWidth
uint16 LoadPercentage
string Stepping
string UniqueID
uint16 CPUStatus

**FileSystem**
string CreationClassName
string Name
string FileSystemType
uint64 FileSystemSize
string Root
uint64 BlockSize
uint64 AvailableSpace
boolean ReadOnly
string EbcryptionMethod
string CompressionMethod
boolean CaseSensitive
boolean CasePreserved
uint16 CodeSet
uint32 MaxFileNameLength
uint32 ClusterSize
uint16 PersistenceType
string OtherPersistenceType
uint64 NumberOfFiles

Hosted FileSystem

**System**
CreationClassName: string { key}
Name: string {override, key}
NameFormat: String
PrimaryOwnerName: string {write}
PrimaryOwnerContact: string {write}
Roles: string [] {write}

InstalledOS

**ComputerSystem**
NameFormat: {override, enum}
OtherIdentifyingInfo: string [ ]
IdentifyingDescriptions: string [ ]
Dedicated: uint16 [ ] {enum}
ResetCapability: uint16 {enum}
PowerManagementCapabilities: uint16[ ] {D, enum}
SetPowerState (
[IN] PowerState: uint32 {enum}
[IN] Time: datetime) : uint32 {D}

ComponentCS

**UnitaryComputerSystem**
InitialLoadInfo: string
LastLoadInfo: string
WakeUpTypes: uint16 {enum}

System Partition

**AdminDomain**
NameFormat: string

ContainedDomain

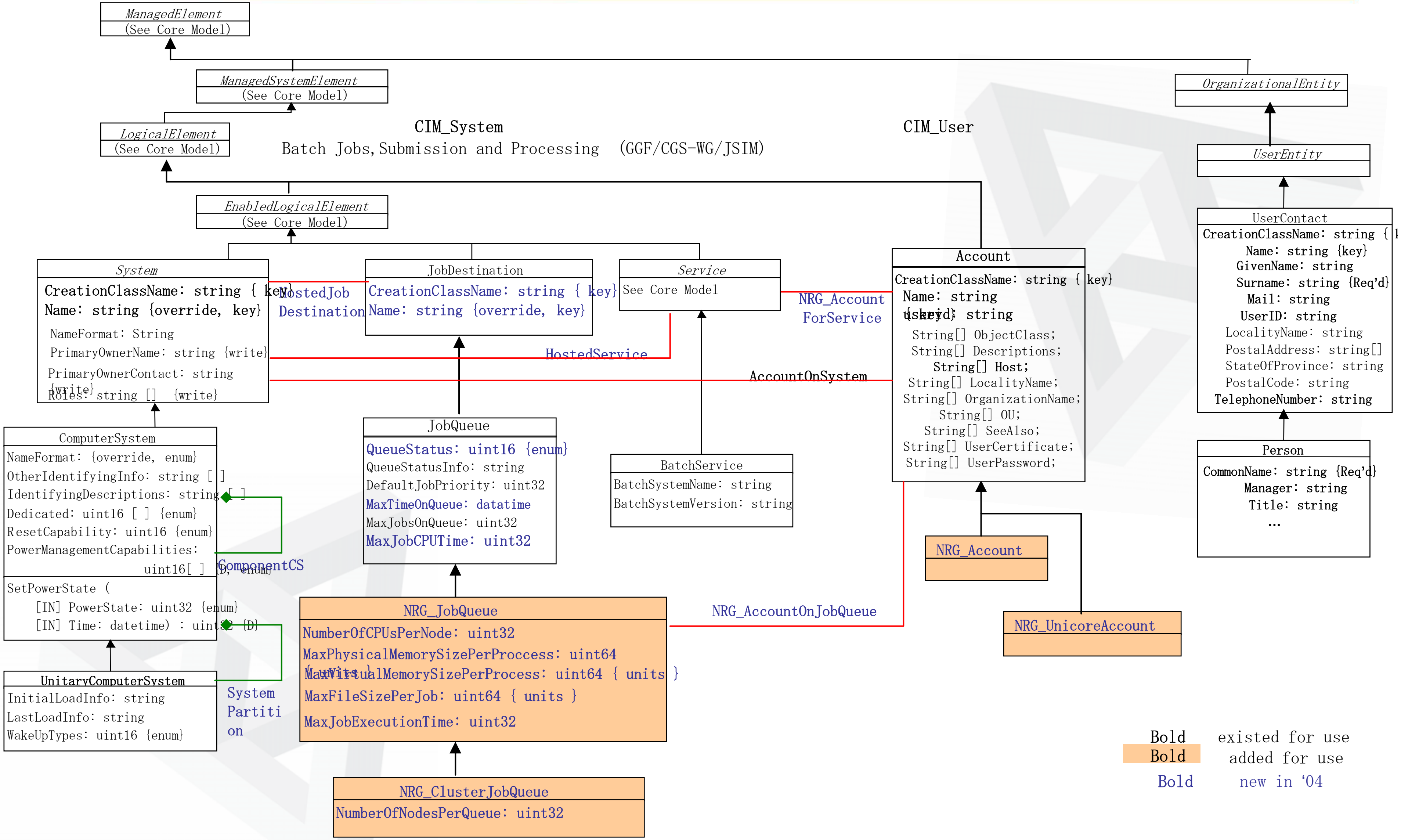
**OperatingSystem**
string CreationClassName
string Name
uint16 OSType
string OtherTypeDescription
string Version
datetime LastBootUpTime
datetime LocalDataTime
sint16 CurrentTimeZone
uint32 NumberOfLicensedUsers
uint32 NumberOfUsers
uint32 NumberOfProcesses
uint32 MaxNumberOfProcesses
uint64 TotalSwapSpaceSize
uint64 TotalVirtualMemorySize
uint64 FreeVirtualMemorySize
uint64 FreePhysicalMemorySize
uint64 TotalVisibleMemorySize
uint64 MaxProcessMemorySize
boolean Distributed
uint32 MaxProcessPerUser

**NRG_OperatingSystem**
uint32 LoadAverageOne
uint32 LoadAverageFive
uint32 LoadAverageFifteen
uint32 NumberOfRunningProcesses

↑ Inheritance
Association
Aggregation

Bold existed for use
Bold added for use
Bold new in '04,'05

# Schema for Job Queue



*ManagedElement*
(See Core Model)

*ManagedSystemElement*
(See Core Model)

*LogicalElement*
(See Core Model)

CIM_System
Batch Jobs, Submission and Processing (GGF/CGS-WG/JSIM)

CIM_User

*OrganizationalEntity*

*UserEntity*

*EnabledLogicalElement*
(See Core Model)

*System*
CreationClassName: string { key}
Name: string {override, key}
NameFormat: String
PrimaryOwnerName: string {write}
PrimaryOwnerContact: string {write}
Roles: string [] {write}

HostedJob Destination

JobDestination
CreationClassName: string { key}
Name: string {override, key}

*Service*
See Core Model

NRG_Account ForService

HostedService

AccountOnSystem

Account
CreationClassName: string { key}
Name: string {key}
UserId: string
String[] ObjectClass;
String[] Descriptions;
String[] Host;
String[] LocalityName;
String[] OrganizationName;
String[] OU;
String[] SeeAlso;
String[] UserCertificate;
String[] UserPassword;

UserContact
CreationClassName: string { key}
Name: string {key}
GivenName: string
Surname: string {Req'd}
Mail: string
UserID: string
LocalityName: string
PostalAddress: string[]
StateOfProvince: string
PostalCode: string
TelephoneNumber: string

Person
CommonName: string {Req'd}
Manager: string
Title: string
...

ComputerSystem
NameFormat: {override, enum}
OtherIdentifyingInfo: string [ ]
IdentifyingDescriptions: string [ ]
Dedicated: uint16 [ ] {enum}
ResetCapability: uint16 {enum}
PowerManagementCapabilities: uint16[ ] {D, enum}
SetPowerState (
[IN] PowerState: uint32 {enum}
[IN] Time: datetime) : uint32 {D}

ComponentCS

UnitaryComputerSystem
InitialLoadInfo: string
LastLoadInfo: string
WakeUpTypes: uint16 {enum}

System Partition

JobQueue
QueueStatus: uint16 {enum}
QueueStatusInfo: string
DefaultJobPriority: uint32
MaxTimeOnQueue: datatime
MaxJobsOnQueue: uint32
MaxJobCPUTime: uint32

BatchService
BatchSystemName: string
BatchSystemVersion: string

NRG_Account

NRG_UnicoreAccount

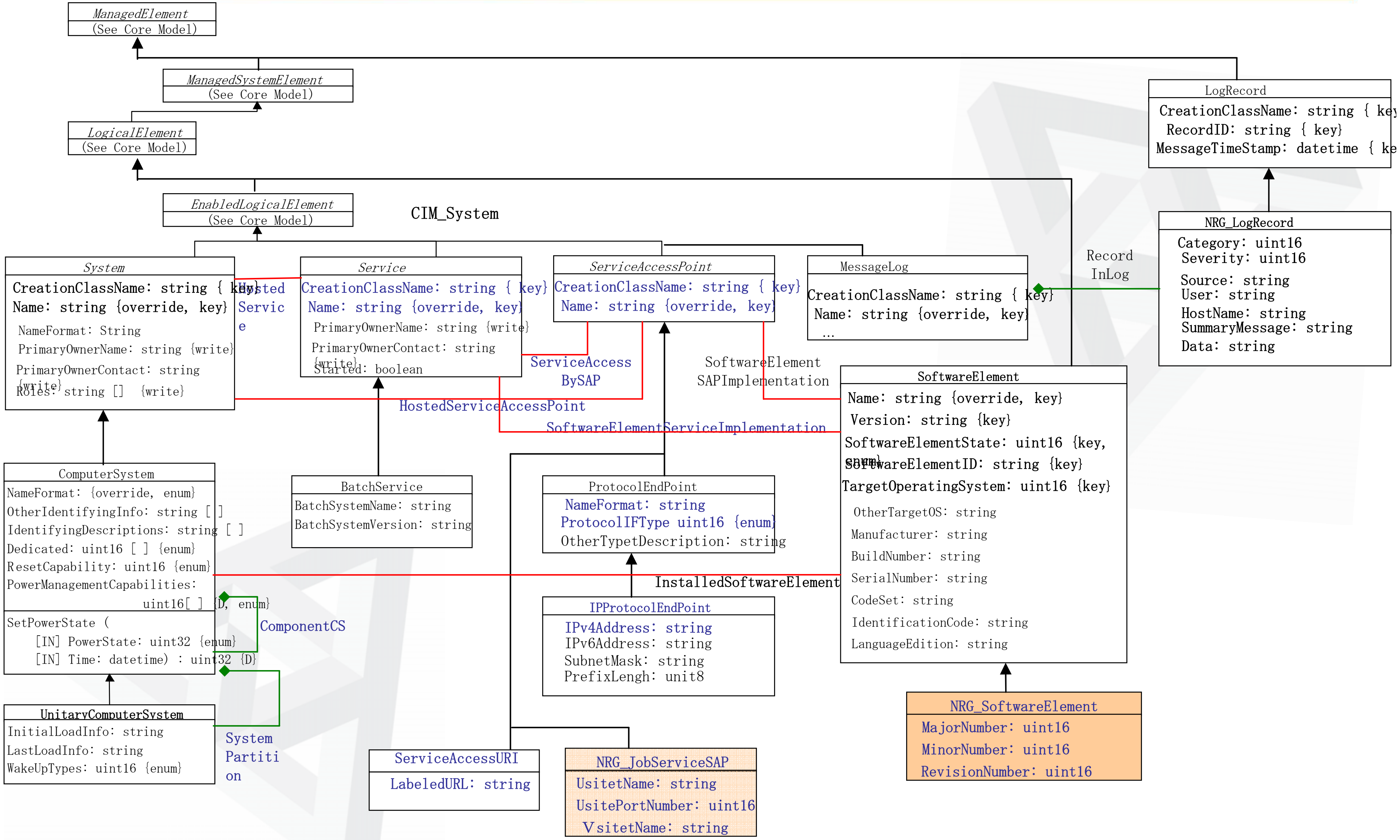
NRG_JobQueue
NumberOfCPUsPerNode: uint32
MaxPhysicalMemorySizePerProccess: uint64 { units }
MaxVirtualMemorySizePerProcess: uint64 { units }
MaxFileSizePerJob: uint64 { units }
MaxJobExecutionTime: uint32

NRG_AccountOnJobQueue

NRG_ClusterJobQueue
NumberOfNodesPerQueue: uint32

Bold existed for use
Bold added for use
Bold new in '04

# Schema for Software, Service, Log



*ManagedElement*
(See Core Model)

*ManagedSystemElement*
(See Core Model)

*LogicalElement*
(See Core Model)

*EnabledLogicalElement*
(See Core Model)

CIM_System

*System*
CreationClassName: string { key}
Name: string {override, key}
NameFormat: String
PrimaryOwnerName: string {write}
PrimaryOwnerContact: string {write}
Roles: string [] {write}

Hosted Service

*Service*
CreationClassName: string { key}
Name: string {override, key}
PrimaryOwnerName: string {write}
PrimaryOwnerContact: string {write}
Started: boolean

*ServiceAccessPoint*
CreationClassName: string { key}
Name: string {override, key}

MessageLog
CreationClassName: string { key}
Name: string {override, key}
…

LogRecord
CreationClassName: string { key}
RecordID: string { key}
MessageTimeStamp: datetime { ke

Record InLog

NRG_LogRecord
Category: uint16
Severity: uint16
Source: string
User: string
HostName: string
SummaryMessage: string
Data: string

ServiceAccess BySAP

SoftwareElement SAPImplementation

HostedServiceAccessPoint

SoftwareElementServiceImplementation

SoftwareElement
Name: string {override, key}
Version: string {key}
SoftwareElementState: uint16 {key, enum}
SoftwareElementID: string {key}
TargetOperatingSystem: uint16 {key}
OtherTargetOS: string
Manufacturer: string
BuildNumber: string
SerialNumber: string
CodeSet: string
IdentificationCode: string
LanguageEdition: string

ComputerSystem
NameFormat: {override, enum}
OtherIdentifyingInfo: string [ ]
IdentifyingDescriptions: string [ ]
Dedicated: uint16 [ ] {enum}
ResetCapability: uint16 {enum}
PowerManagementCapabilities: uint16[ ] {D, enum}
SetPowerState (
[IN] PowerState: uint32 {enum}
[IN] Time: datetime) : uint32 {D}

ComponentCS

BatchService
BatchSystemName: string
BatchSystemVersion: string

ProtocolEndPoint
NameFormat: string
ProtocolIFType uint16 {enum}
OtherTypetDescription: string

InstalledSoftwareElement

IPProtocolEndPoint
IPv4Address: string
IPv6Address: string
SubnetMask: string
PrefixLengh: unit8

UnitaryComputerSystem
InitialLoadInfo: string
LastLoadInfo: string
WakeUpTypes: uint16 {enum}

System Partition

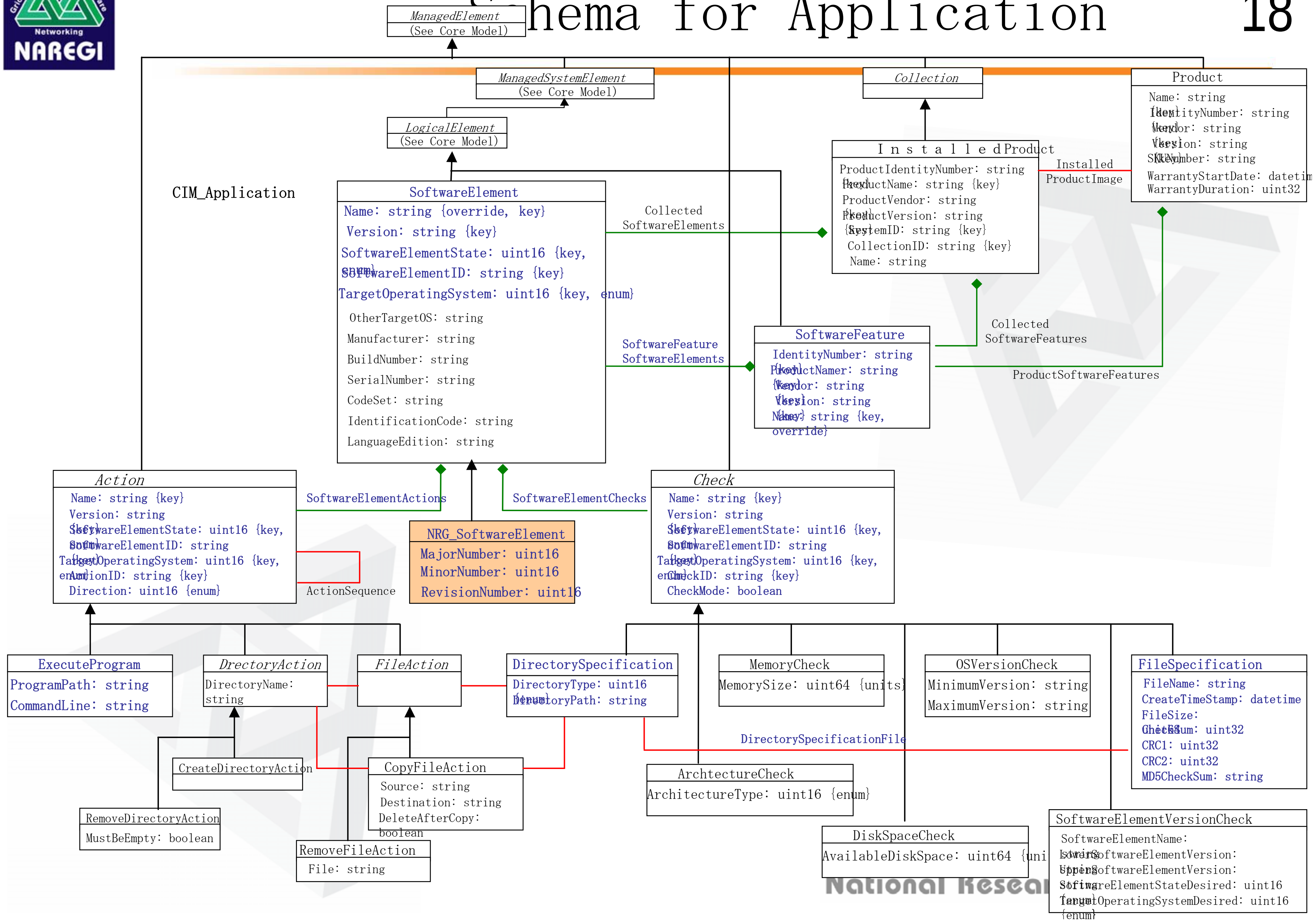
ServiceAccessURI
LabeledURL: string

NRG_JobServiceSAP
UsitetName: string
UsitePortNumber: uint16
VsitetName: string

NRG_SoftwareElement
MajorNumber: uint16
MinorNumber: uint16
RevisionNumber: uint16

: used for α version

# Schema for Application



CIM_Application

ManagedElement
(See Core Model)

ManagedSystemElement
(See Core Model)

LogicalElement
(See Core Model)

Collection

Product
Name: string {key}
IdentityNumber: string {key}
Vendor: string {key}
Version: string {key}
SKUNumber: string
WarrantyStartDate: datetim
WarrantyDuration: uint32

InstalledProduct
ProductIdentityNumber: string {key}
ProductName: string {key}
ProductVendor: string {key}
ProductVersion: string {key}
SystemID: string {key}
CollectionID: string {key}
Name: string

Installed ProductImage

SoftwareElement
Name: string {override, key}
Version: string {key}
SoftwareElementState: uint16 {key, enum}
SoftwareElementID: string {key}
TargetOperatingSystem: uint16 {key, enum}
OtherTargetOS: string
Manufacturer: string
BuildNumber: string
SerialNumber: string
CodeSet: string
IdentificationCode: string
LanguageEdition: string

Collected SoftwareElements

SoftwareFeature SoftwareElements

SoftwareFeature
IdentityNumber: string {key}
ProductNamer: string {key}
Vendor: string {key}
Version: string {key}
Name: string {key, override}

Collected SoftwareFeatures

ProductSoftwareFeatures

Action
Name: string {key}
Version: string {key}
SoftwareElementState: uint16 {key, enum}
SoftwareElementID: string {key}
TargetOperatingSystem: uint16 {key, enum}
ActionID: string {key}
Direction: uint16 {enum}

SoftwareElementActions

ActionSequence

NRG_SoftwareElement
MajorNumber: uint16
MinorNumber: uint16
RevisionNumber: uint16

SoftwareElementChecks

Check
Name: string {key}
Version: string {key}
SoftwareElementState: uint16 {key, enum}
SoftwareElementID: string {key}
TargetOperatingSystem: uint16 {key, enum}
CheckID: string {key}
CheckMode: boolean

ExecuteProgram
ProgramPath: string
CommandLine: string

DirectoryAction
DirectoryName: string

FileAction

CreateDirectoryAction

RemoveDirectoryAction
MustBeEmpty: boolean

CopyFileAction
Source: string
Destination: string
DeleteAfterCopy: boolean

RemoveFileAction
File: string

DirectorySpecification
DirectoryType: uint16 {enum}
DirectoryPath: string

DirectorySpecificationFile

MemoryCheck
MemorySize: uint64 {units}

OSVersionCheck
MinimumVersion: string
MaximumVersion: string

FileSpecification
FileName: string
CreateTimeStamp: datetime
FileSize: uint64
CheckSum: uint32
CRC1: uint32
CRC2: uint32
MD5CheckSum: string

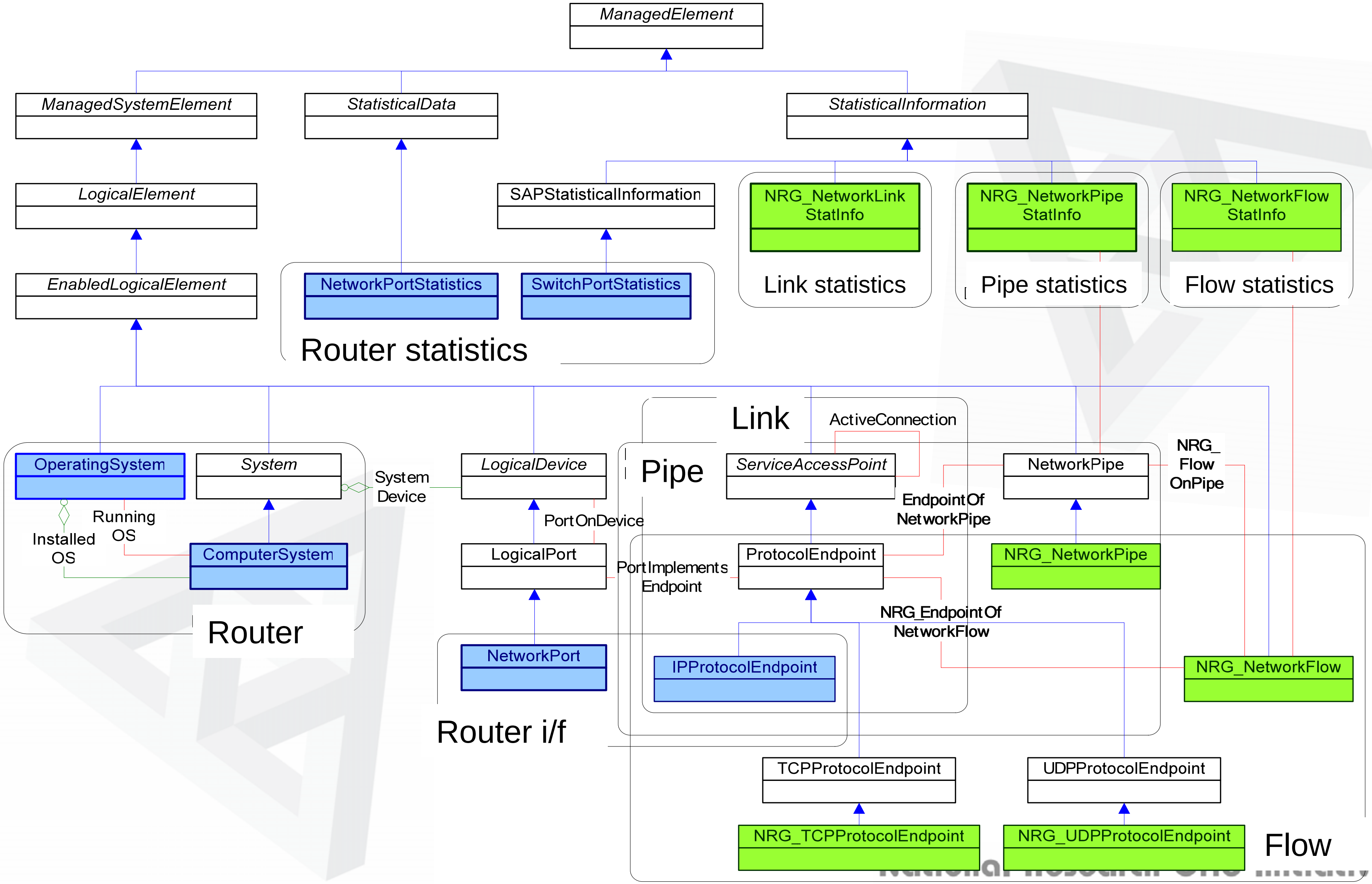
ArchtectureCheck
ArchitectureType: uint16 {enum}

DiskSpaceCheck
AvailableDiskSpace: uint64 {uni

SoftwareElementVersionCheck
SoftwareElementName: string
LowerSoftwareElementVersion: string
UpperSoftwareElementVersion: string
SoftwareElementStateDesired: uint16 {enum}
TargetOperatingSystemDesired: uint16 {enum}

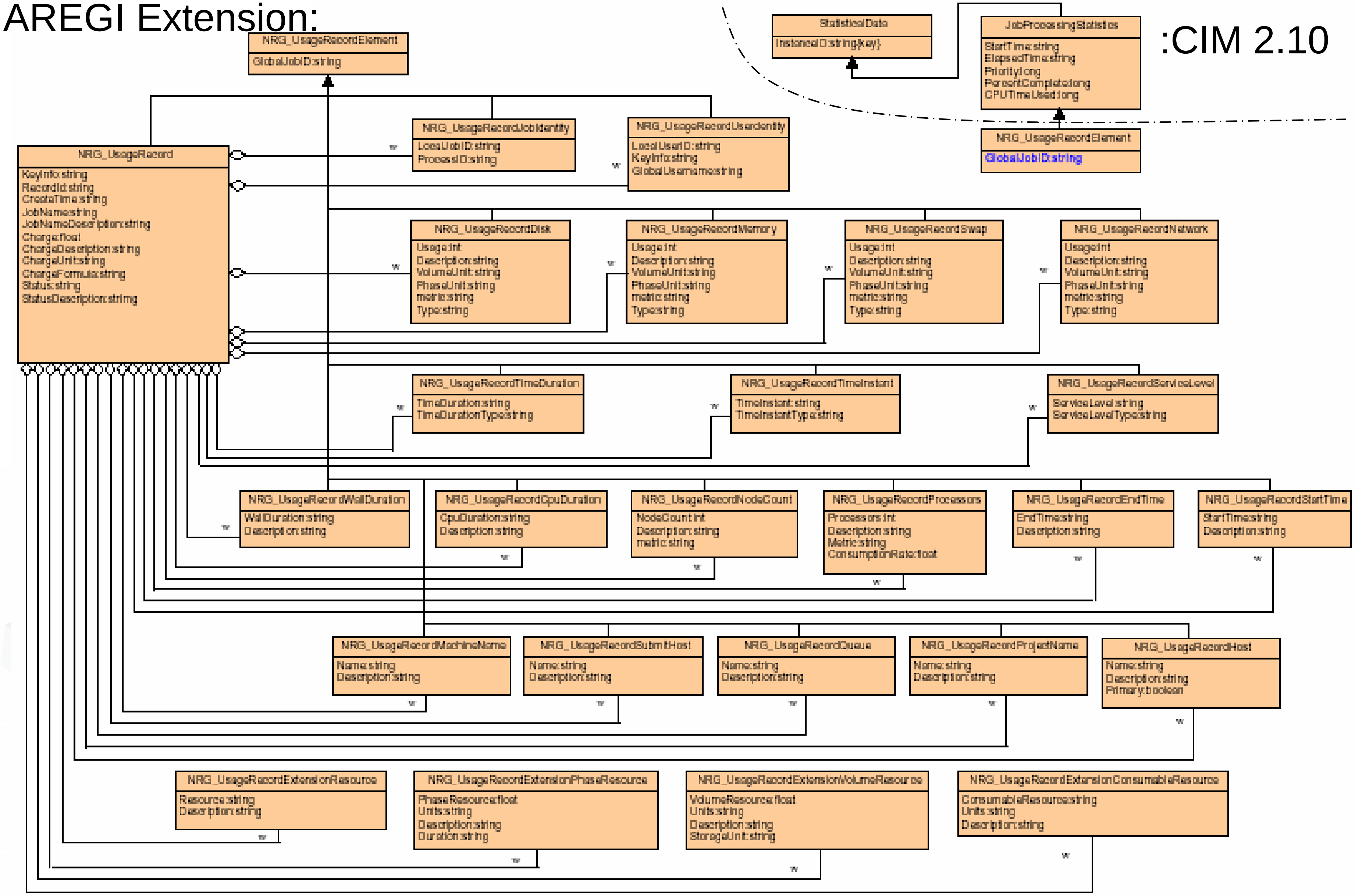
National Resear

# Schema for Network



# Schema for Usage Record

NAREGI Extension:



:CIM 2.10

StatisticalData
InstanceID:string{key}

JobProcessingStatistics
StartTime:string
ElapsedTime:string
Priority:long
PercentComplete:long
CPUTimeUsed:long

NRG_UsageRecordElement
GlobalJobID:string

NRG_UsageRecordElement
GlobalJobID:string

NRG_UsageRecord
KeyInfo:string
RecordId:string
CreateTime:string
JobName:string
JobNameDescription:string
Charge:float
ChargeDescription:string
ChargeUnit:string
ChargeFormula:string
Status:string
StatusDescription:strimg

NRG_UsageRecordJobIdentity
LocalJobID:string
ProcessID:string

NRG_UsageRecordUserIdentity
LocalUserID:string
KeyInfo:string
GlobalUsername:string

NRG_UsageRecordDisk
Usage:int
Description:string
VolumeUnit:string
PhaseUnit:string
metric:string
Type:string

NRG_UsageRecordMemory
Usage:int
Description:string
VolumeUnit:string
PhaseUnit:string
metric:string
Type:string

NRG_UsageRecordSwap
Usage:int
Description:string
VolumeUnit:string
PhaseUnit:string
metric:string
Type:string

NRG_UsageRecordNetwork
Usage:int
Description:string
VolumeUnit:string
PhaseUnit:string
metric:string
Type:string

NRG_UsageRecordTimeDuration
TimeDuration:string
TimeDurationType:string

NRG_UsageRecordTimeInstant
TimeInstant:string
TimeInstantType:string

NRG_UsageRecordServiceLevel
ServiceLevel:string
ServiceLevelType:string

NRG_UsageRecordWallDuration
WallDuration:string
Description:string

NRG_UsageRecordCpuDuration
CpuDuration:string
Description:string

NRG_UsageRecordNodeCount
NodeCount:int
Description:string
metric:string

NRG_UsageRecordProcessors
Processors:int
Description:string
Metric:string
ConsumptionRate:float

NRG_UsageRecordEndTime
EndTime:string
Description:string

NRG_UsageRecordStartTime
StartTime:string
Description:string

NRG_UsageRecordMachineName
Name:string
Description:string

NRG_UsageRecordSubmitHost
Name:string
Description:string

NRG_UsageRecordQueue
Name:string
Description:string

NRG_UsageRecordProjectName
Name:string
Description:string

NRG_UsageRecordHost
Name:string
Description:string
Primary:boolean

NRG_UsageRecordExtensionResource
Resource:string
Description:string

NRG_UsageRecordExtensionPhaseResource
PhaseResource:float
Units:string
Description:string
Duration:string

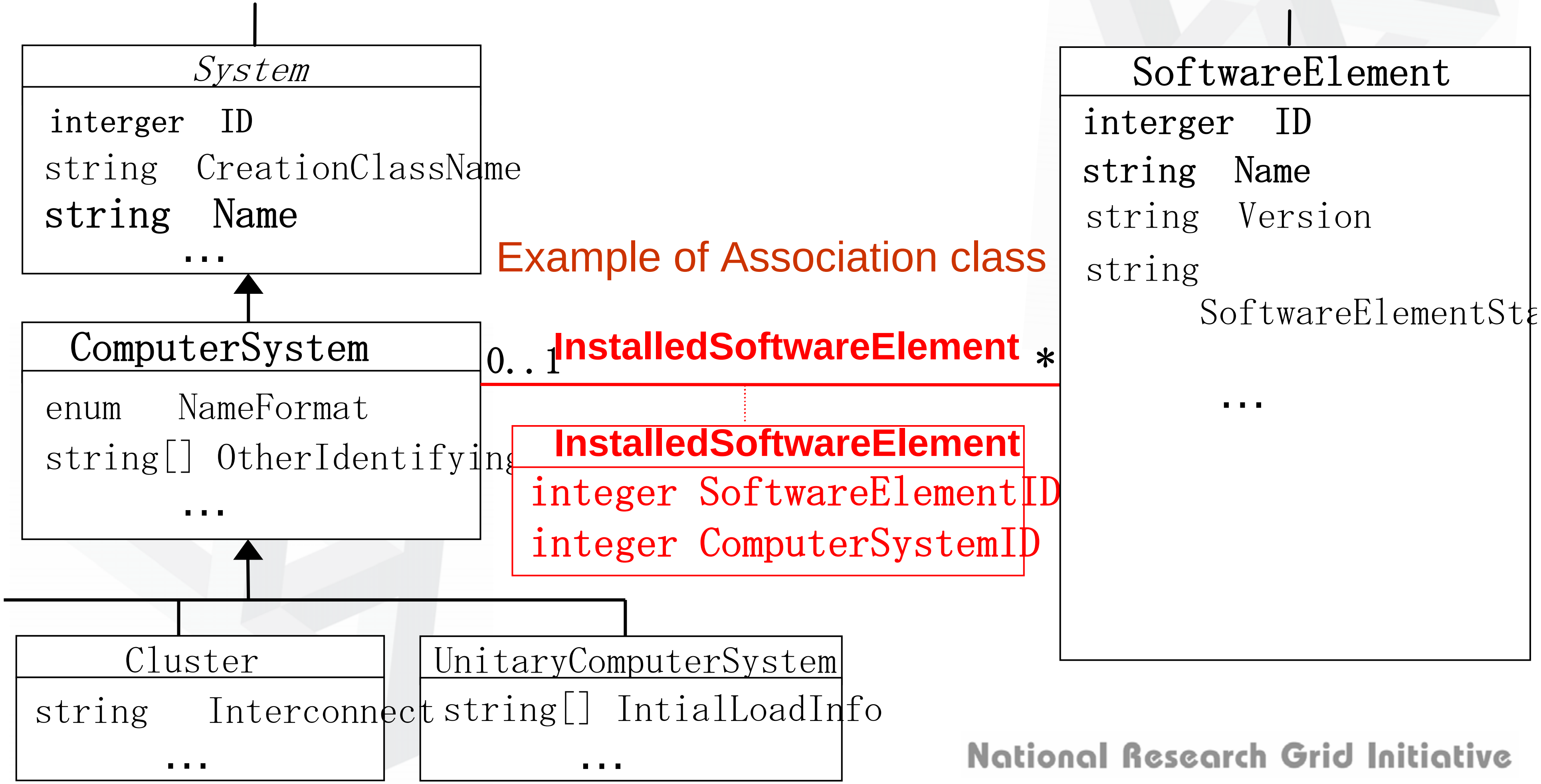
NRG_UsageRecordExtensionVolumeResource
VolumeResource:float
Units:string
Description:string
StorageUnit:string

NRG_UsageRecordExtensionConsumableResource
ConsumableResource:string
Units:string
Description:string

# Mapping to PostgreSQL (example)

- A class in CIM Schema corresponds to a table in RDB.
- An association class has pair of key properties of 2 classes and is used to join the tables.

| CIM | ORDB |
|---|---|
| Class | Table |
| Instance | Record (row) |
| Property | Field |
| Key | ID |



Example of Association class

*System*
interger ID
string CreationClassName
string Name
...

ComputerSystem
enum NameFormat
string[] OtherIdentifying
...

Cluster
string Interconnect
...

UnitaryComputerSystem
string[] IntialLoadInfo
...

0..1 InstalledSoftwareElement *

InstalledSoftwareElement
integer SoftwareElementID
integer ComputerSystemID

SoftwareElement
interger ID
string Name
string Version
string SoftwareElementSta
...

# SQL query to RDB

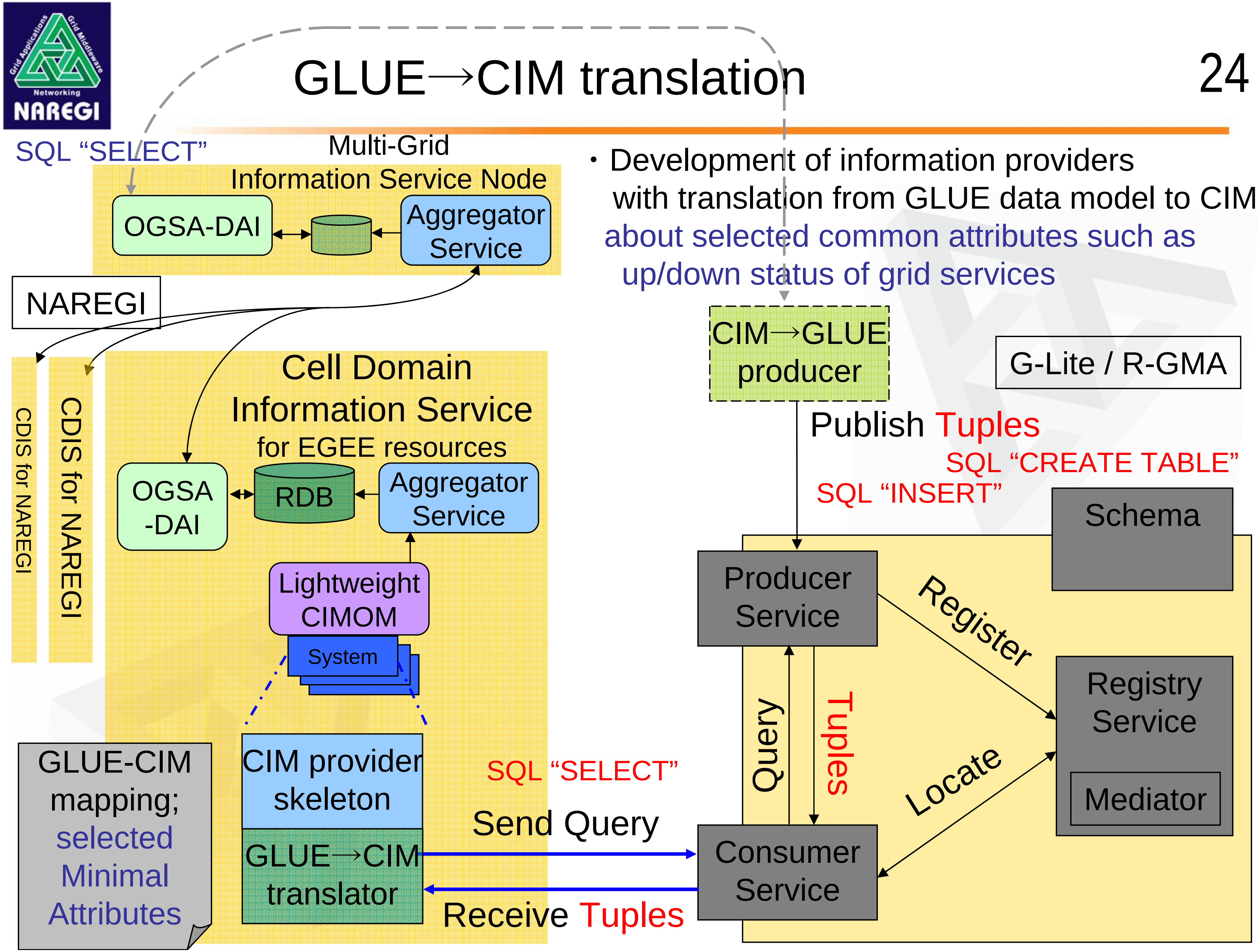
- SQL query through Association class :

```
SELECT  Name
FROM     CIM_ComputerSystem
WHERE
  (
        /*  Join condition with Association class   */
     CIM_InstalledSoftwareElement.SoftwareElementID = CIM_SoftwareElement.ID

  AND
        /*   Join condition with Association   */
     CIM_InstalledSoftwareElement.ComputerSystemID = CIM_ComputerSystem.ID

  ) AND (
        /*   Condition for this search   */
     CIM_SoftwareElement.Name = ‘intel-ifort8’
  );
```

1. Overview
2. Resource information schema
3. Publisher interface
4. Consumer interface
5. VO information service

# GLUE→CIM translation

- Development of information providers with translation from GLUE data model to CIM about selected common attributes such as up/down status of grid services



SQL “SELECT”

Multi-Grid Information Service Node

OGSA-DAI

Aggregator Service

NAREGI

CDIS for NAREGI

CDIS for NAREGI

Cell Domain Information Service for EGEE resources

OGSA -DAI

RDB

Aggregator Service

Lightweight CIMOM

System

GLUE-CIM mapping; selected Minimal Attributes

CIM provider skeleton

GLUE→CIM translator

SQL “SELECT”

Send Query

Receive Tuples

CIM→GLUE producer

G-Lite / R-GMA

Publish Tuples

SQL “CREATE TABLE”

SQL “INSERT”

Schema

Producer Service

Register

Registry Service

Mediator

Query

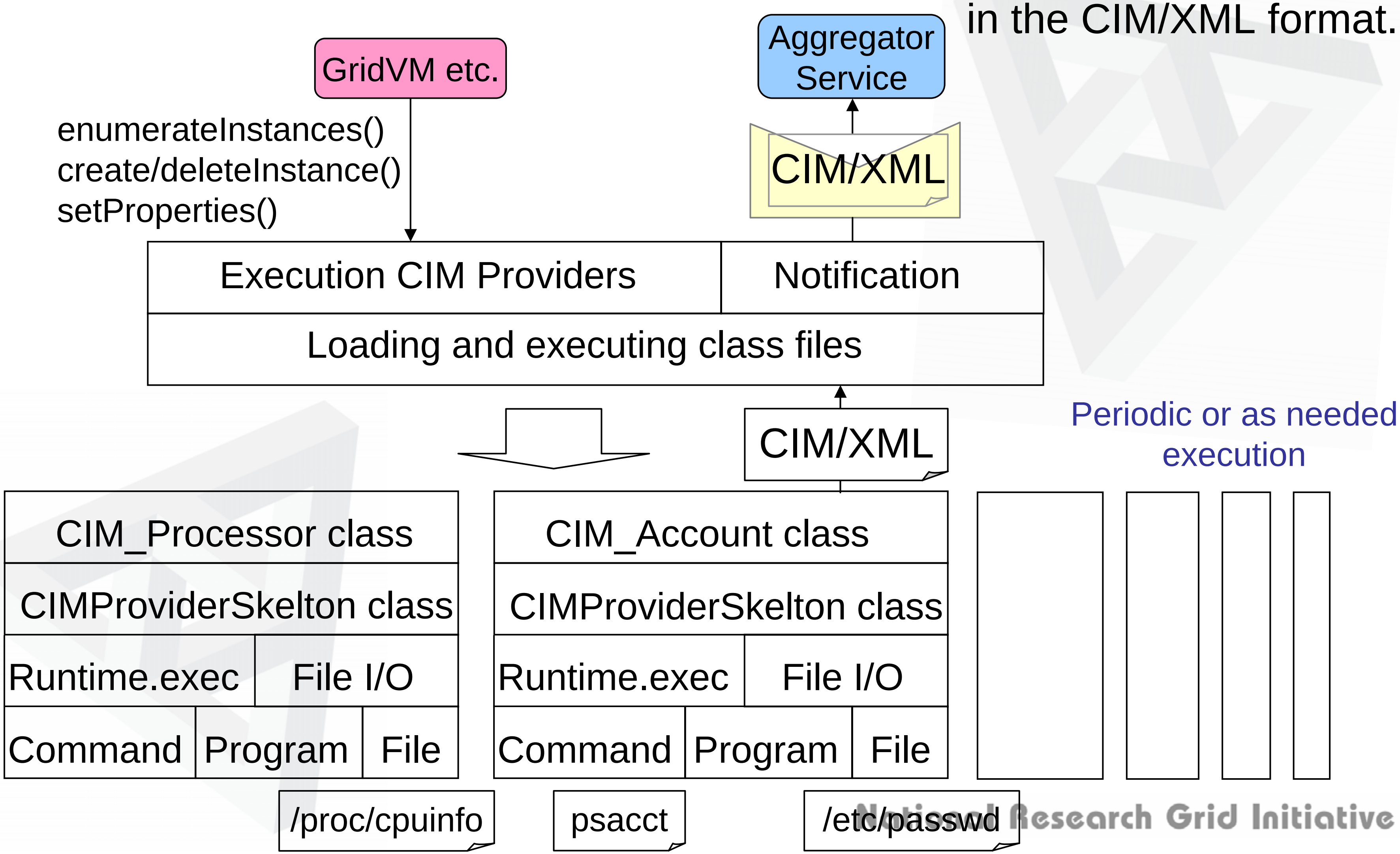
Tuples

Locate

Consumer Service

# Light-weight CIMOM

We developed Grid Service that manages CIM Provider classes and transmits resource information to AggregateService (～IndexService) in the CIM/XML format.



# Information Provider plug-in

Developers of NAREGI M/W can easily implement provider software.

- CIM provider classes extend CIMProviderSkelton class.
  Association provider classes extend CIMAssociationProviderSkelton class.

- The Skelton class has
  execProvider() : starting point of the provider,
  createCIMInstance() ,
  addInstance(cimInstance) : notifies to RDB, etc.

- CIMInstance class has
  addKeyBinding(key, type, value)
  addProperty(name, type, value) .

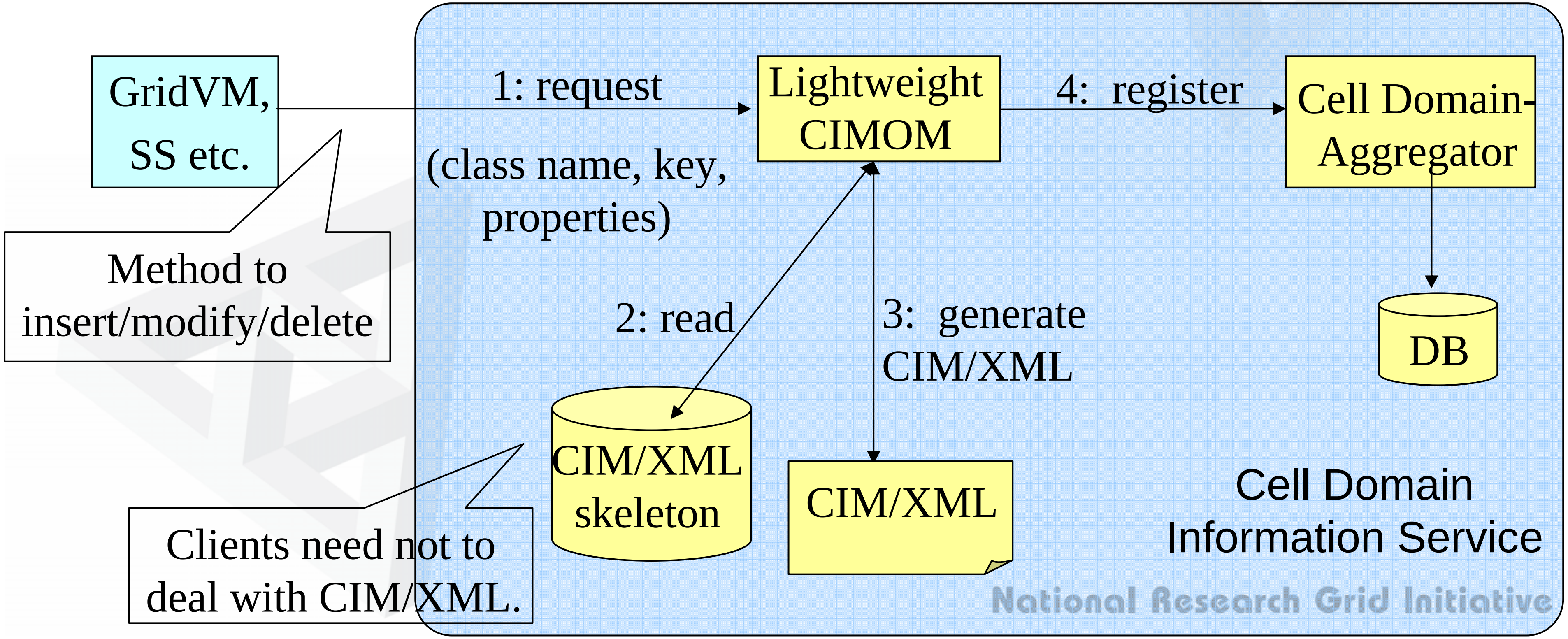
- Providers are put in $GLOBUS_LOCATION/lib/ directory.

# Example

```
import java.io.*;
import java.util.*;
public class NRG_Account extends CIMProviderSkelton {
    public NRG_Account()  {}
    public void execProvider() throws Exception {
        try {
            FileInputStream inFile = openFile("accountList.txt");  //  Account Information is in the file.
            BufferedReader buf_in = new BufferedReader(new InputStreamReader(inFile));
            String buf;
            while((buf = buf_in.readLine()) != null) {
                String userid = buf.trim();
                if(userid.length() < 1) {
                    continue;
                }
                CIMInstance cimInstance = createCIMInstance();
                // KEYBINDING
                cimInstance.addKeyBinding("UserID", "string", userid);
                addInstance(cimInstance);
            }
        } finally {
           closeFile();
}   }   }
```

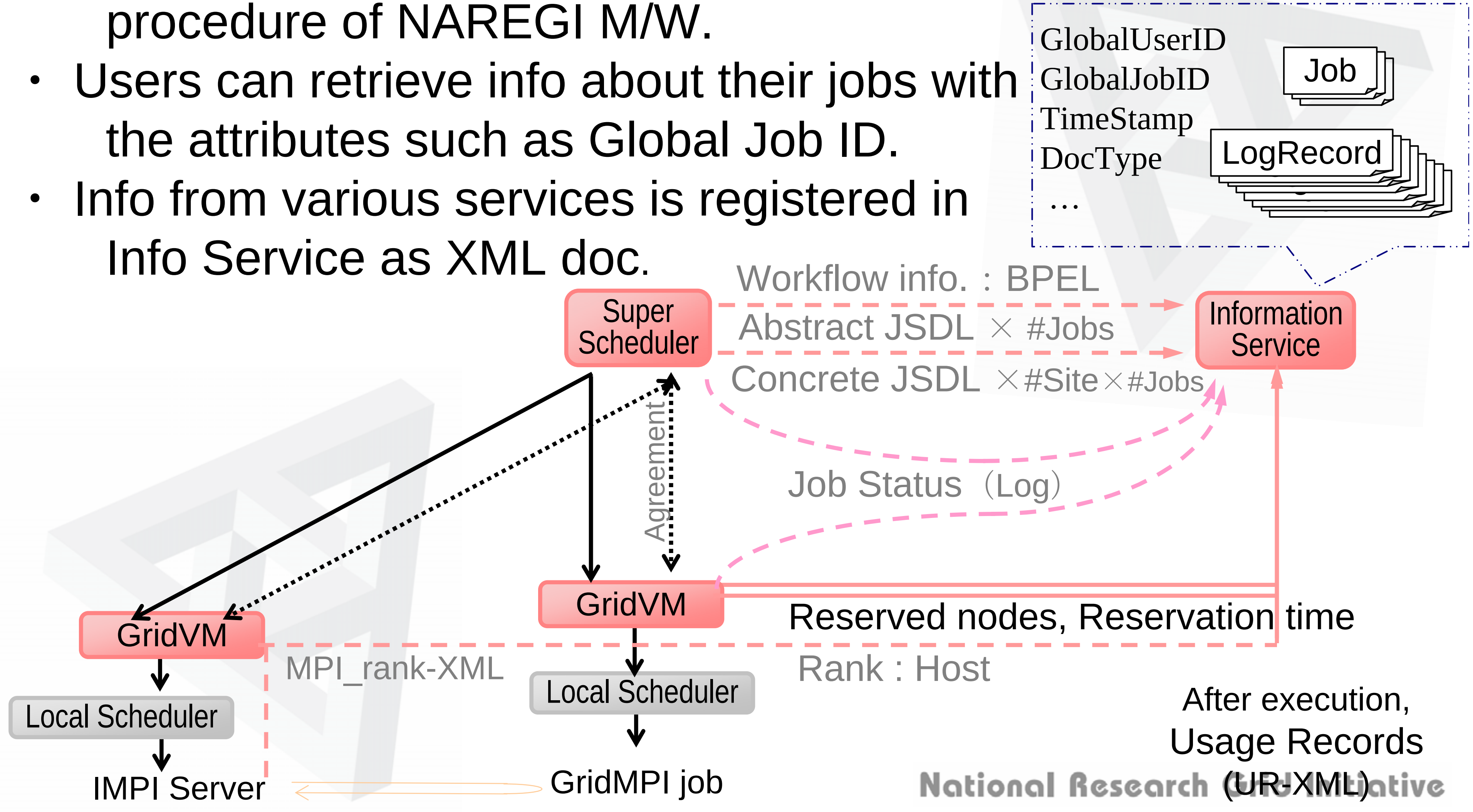
# create/deleteInstance(), setProperpies()

- A CIM instance (row in RDB) can be created/altered/deleted through above operations of Lightweight CIMOM.
- A Set of attributes as retrieval keys and XML document as detailed information can be stored in RDB. (NRG_xxxLog class)
- AuthZ for publishing info. : gridmap-file (M/W or Admin : allowed)

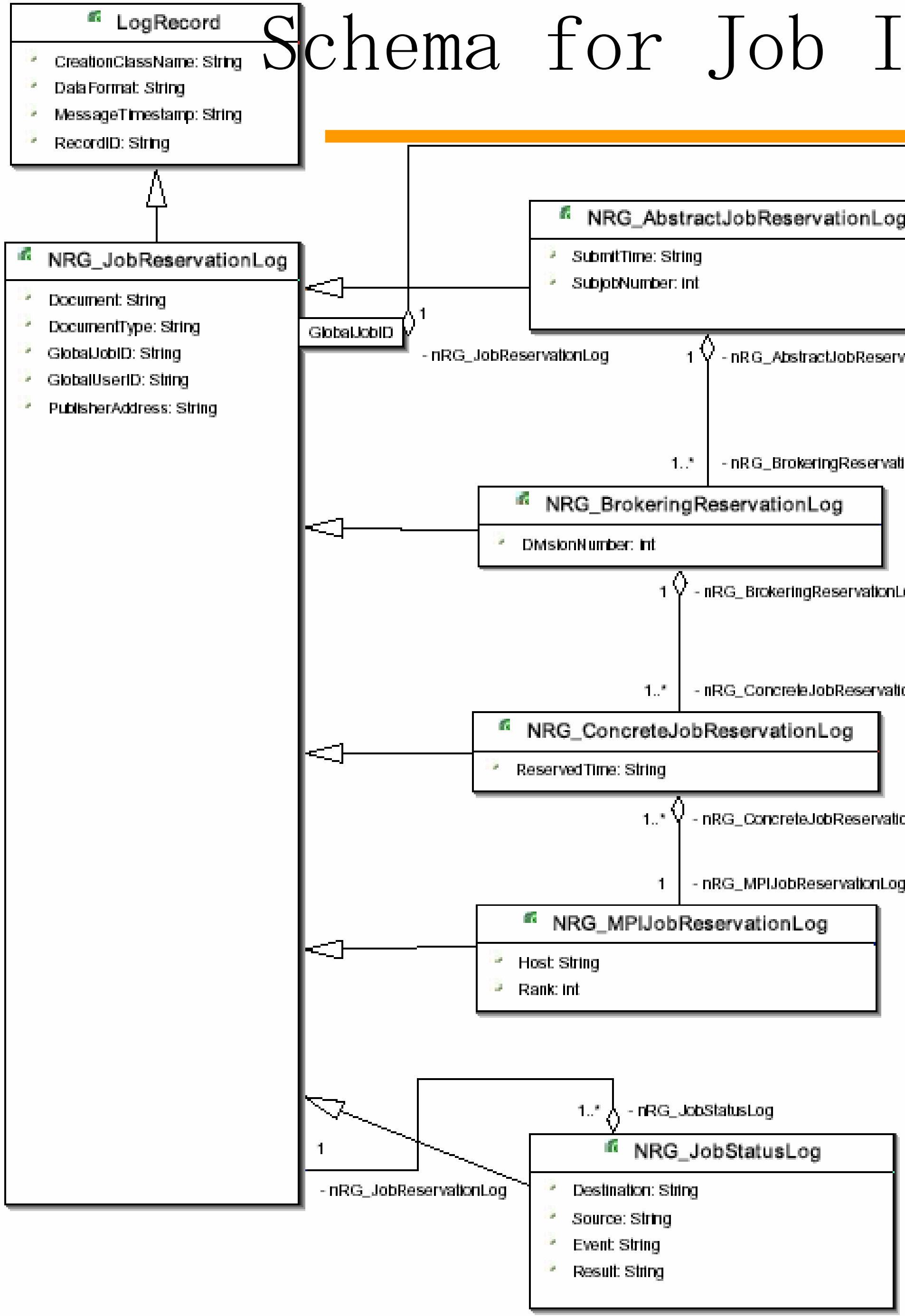

GridVM, SS etc.
1: request
(class name, key, properties)
Lightweight CIMOM
4: register
Cell Domain-Aggregator
Method to insert/modify/delete
2: read
3: generate CIM/XML
DB
CIM/XML skeleton
CIM/XML
Cell Domain Information Service
Clients need not to deal with CIM/XML.

# Repository of Job Information

- Information Service traces processes of job execution mgmt, where the job info described by users gets concrete in the procedure of NAREGI M/W.
- Users can retrieve info about their jobs with the attributes such as Global Job ID.
- Info from various services is registered in Info Service as XML doc.



# Schema for Job Information



BPEL

Abstract JSDL

Concrete JSDL,
Reserved Node

Rank : Host

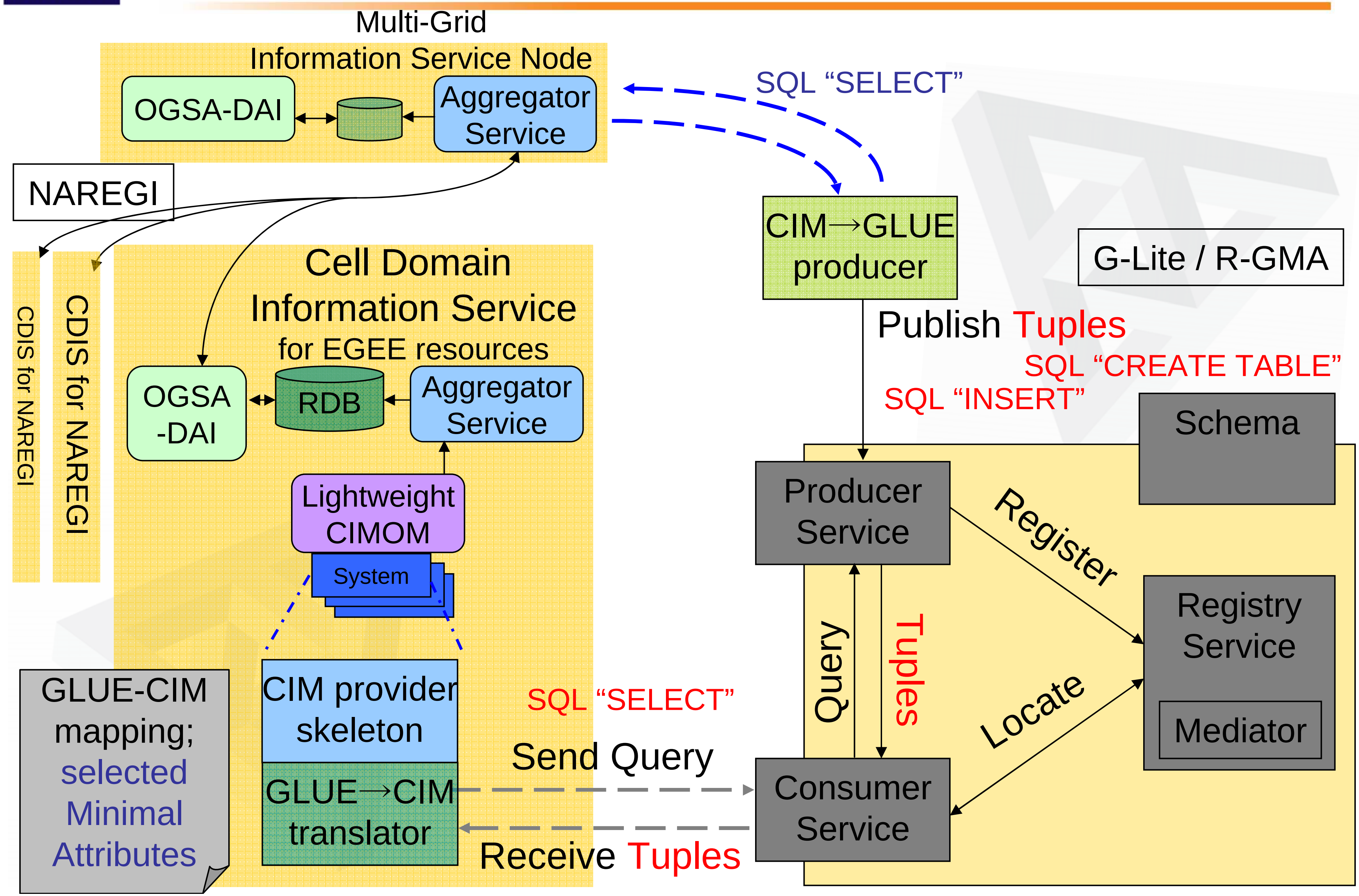
Job status

# Example of XML about Reserved Nodes

```
<?xml version="1.0"?>
<Reservation xmlns="http://www.naregi.org/infoservice/namespaces/sbc"
      xmlns:xsi=" http://www.w3.org/2001/XMLSchema-instance”
      xsi:schemaLocation="http://www. naregi.org/infoservice/namespaces/sbc/sbcfile.xsd>
<Job>
  <GlobalJobID>87407426632466317</GlobalJobID>
  <SubmittingUserName>/C=JP/O=NII/CN=Saeki</SubmittingUserName>
  <VOAttributeName>/wp1.naregi.org/InfoService</VOAttributeName>
  <JobType>GridMPI</JobType>
  <ApplicationName>FMO</ApplicationName>
  <Executable>gamess00.x</Executable>
</Job>
<Site>
  <TargetHost>pbg1003.naregi.org</TargetHost>
  <IMPIClientId>0</IMPIClientId>
  <SitesCoallocated>7<SitesCoallocated>
</Site>
<Node>
  <HostName>pbg1004.naregi.org</HostName>
  <HostName>pbg1003.naregi.org</HostName>
  <HostName>pbg1004.naregi.org</HostName>
  <HostName>pbg1003.naregi.org</HostName>
</Node>
</Reservation>
```

1. Overview
2. Resource information schema
3. Publisher interface
4. Consumer interface
5. VO information service

# CIM→GLUE translation

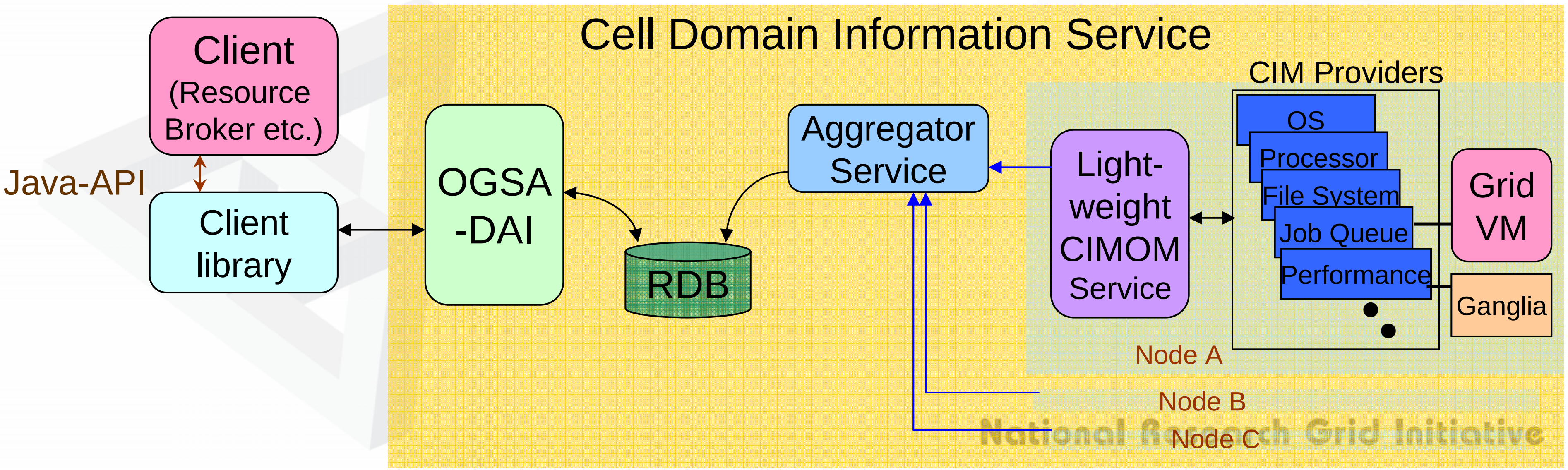


# Query i/f to Cell Domain Info.Service

- class SQLClientWSRF
  - SQLResult[] cellDomainQuery(String[] names, String sql)

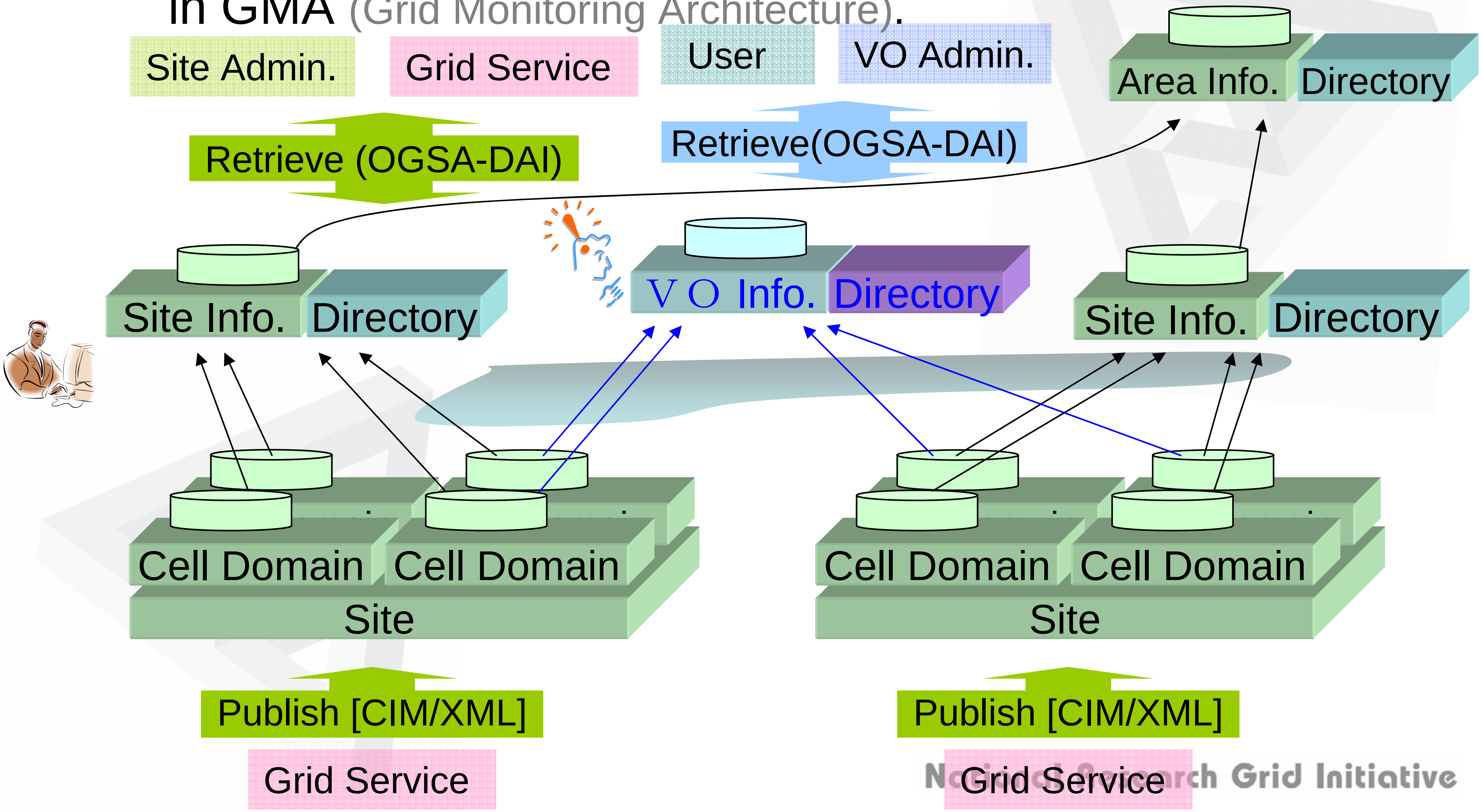
    names : Names of Cell Domains ... Scope of query,
    sql : SQL expression ... SELECT, CREATE VIEW,

- class SQLResult
  - String getTargetName() : Name of Cell Domain,
  - String[] getHostName() : Hosts within the target domain,
  - ResultSet getResultSet() : Result of the query,
  - void discard()



# Multi-Domain connection： GMA feature

- Cell Domain Info Services are hierarchically connected.
- Info Service Nodes in the upper layer play a role of Directory in GMA (Grid Monitoring Architecture).



# Index to Multi-Domain Info Services

- class SQLClientWSRF
  - SQLClientWSRF(String nodeURL)

    nodeURL : URL of target “Information Service Node” in upper layer.
  - IndexInfo getIndexInfo()

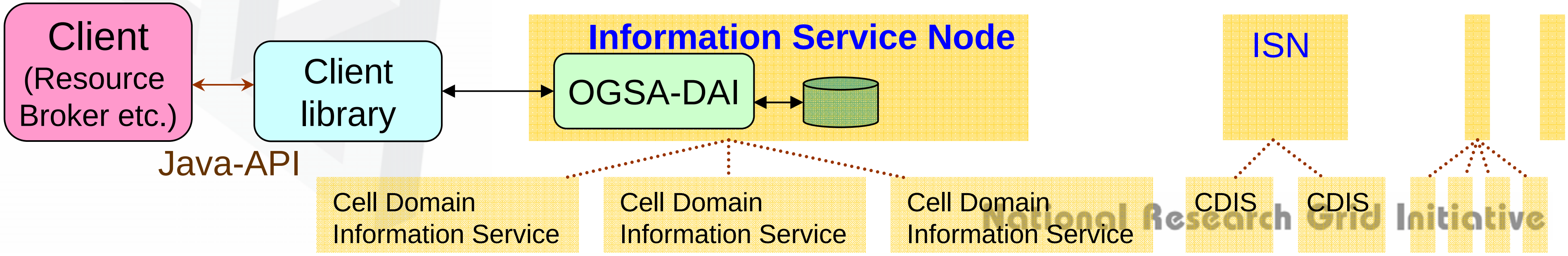
◇ class IndexInfo

- String[] getCellDomainNames() : Cell Domains in lower layer of the target noc
- String[] getHostNames(cellDomainName) : Hosts in the specified Cell Doma
- String getOwnerCellDomainName(hostName)

  : Cell Domain with the specified host,
- String[] getContainerCellDomainNames (cimClassName)

  : Cell Domains with specified class information in the lower layer,
- String[] getContainerHostNames (cimClassName)

  : Hosts with specified class information in the lower layer.



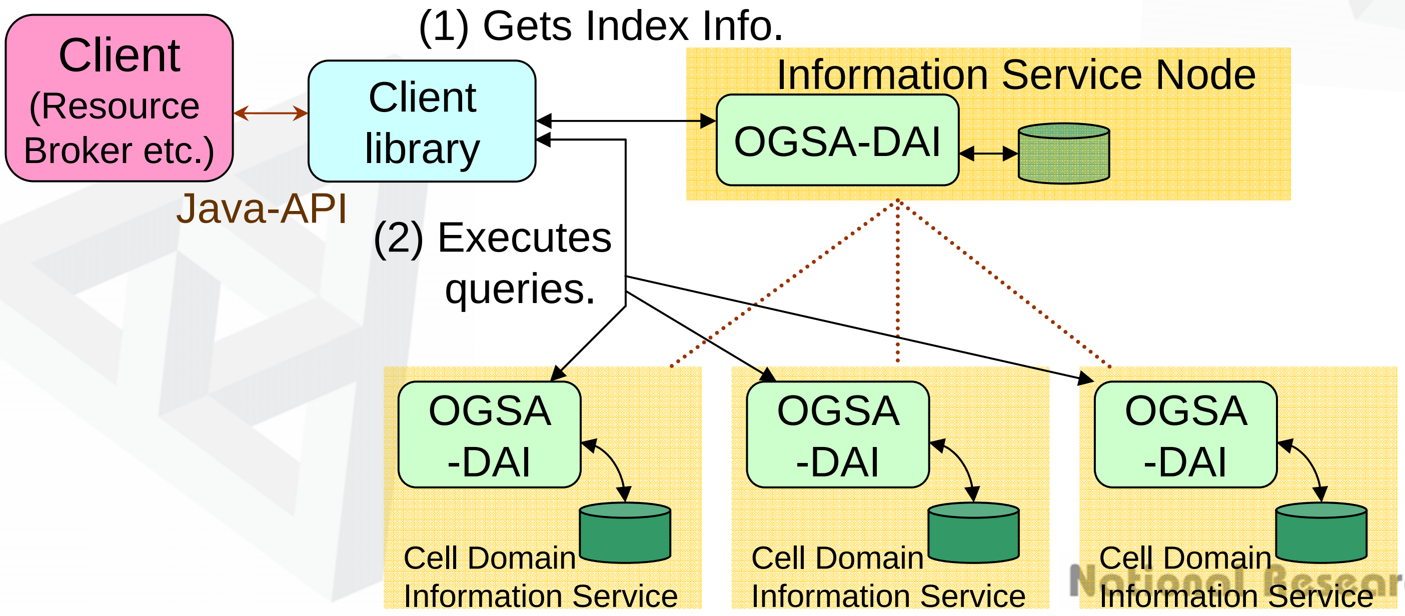
# Query i/f to Multi-Domain Info. Services

- class SQLClientWSRF
  - SQLClientWSRF(String nodeURL)

    nodeURL : URL of target Information Service Node in upper layer.

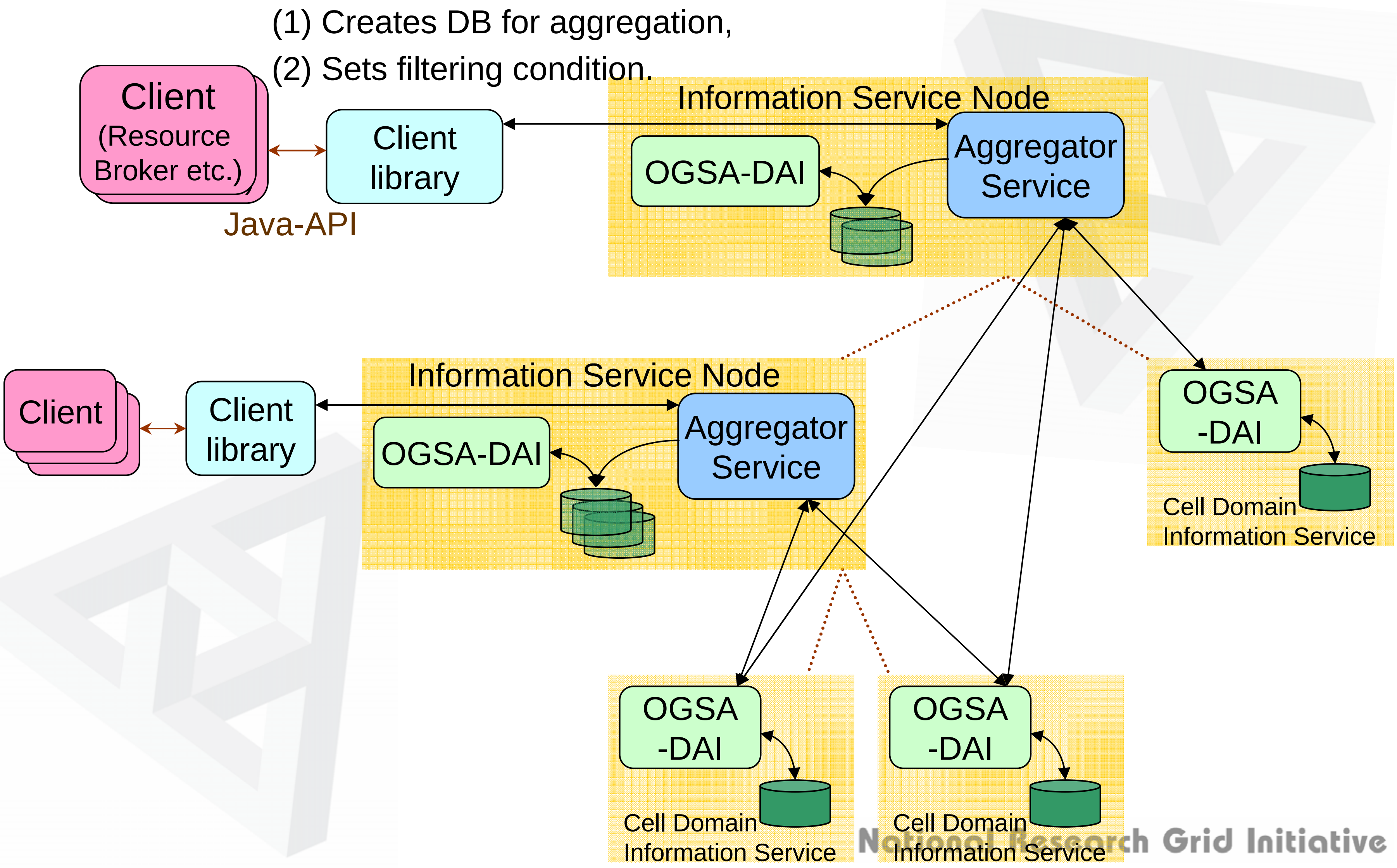
  - SQLResult[] cellDomainQuery(String[] names, String sql)

    names : Names of Cell Domains ... Scope of query,
    = null ⇒ target = all Cell Domains in lower of the target node,
    sql : SQL expression ... SELECT, CREATE VIEW.



# Filtered Aggregation to IS Node

NAREGI M/W components can create their DBs in Information Service Nodes.



# Filtered Aggregation i/f

- ◆ class SQLClientWSRF
  - ・ SQLClientWSRF(String nodeURL)

    nodeURL : URL of target Information Service Node in upper layer.

  - ・ ClassAggregateHandle createClassAggregate()

    : creates DB for filtered aggregation in the target ISN,

◇ class ClassAggregateHandle

・boolean store(String absoluteFilePath) : saves the created handle.

  - ・ ClassAggregateHandle loadClassAggregateHangle() : loads the saved handle

  - ・ boolean addAggregateClass

    (ClassAggregateHandle handle, String className,

    String filterSqlWhereClause, int refreshFequency, String freqencyUnit)

    className : Name of CIM class to be aggregated to the DB in IS Node,
    filterSqlWhereClause : Condition of instances to be aggregated.

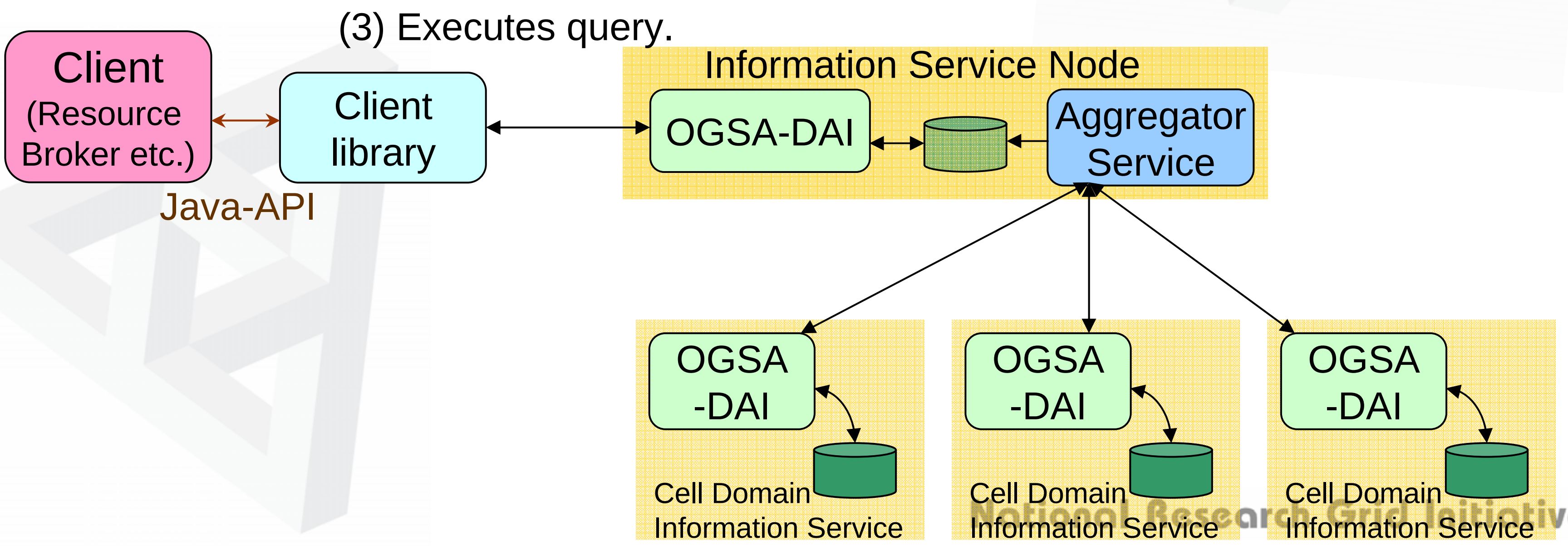
# Query i/f to Information Service Node

- class SQLClientWSRF
  - SQLClientWSRF(String nodeURL)

    nodeURL : URL of target Information Service Node in upper layer.

  - SQLResult[] nodeQuery (ClassAggregateHandle handle, String sql)

    handle : handle of target DB in the IS Node ... Scope of query,
    sql : SQL expression ... SELECT, CREATE VIEW.


(3) Executes query.
Client (Resource Broker etc.)
Client library
Java-API
Information Service Node
OGSA-DAI
Aggregator Service
OGSA -DAI
Cell Domain Information Service
OGSA -DAI
Cell Domain Information Service
OGSA -DAI
Cell Domain Information Service

1. Overview
2. Resource information schema
3. Publisher interface
4. Consumer interface
5. VO information service

# VO Management in NAREGI

a) VOMS Server manages VO membership.

Same as EGEE.

b) VO Information Service maintains information about the VO.

Information about computer systems to which members of the VO have access right. <=  Extention of CIM_Account, JobQueue.

c) Local Authorization : gridmap-file ＆ Resource usage mgmt. for VOs by GridVM.

Limits of resource usage for VOs are described in policy of local sites. … {Wall time, CPU time, Disk size}

d) VO members can use appropriate resources according to local authz. policy.

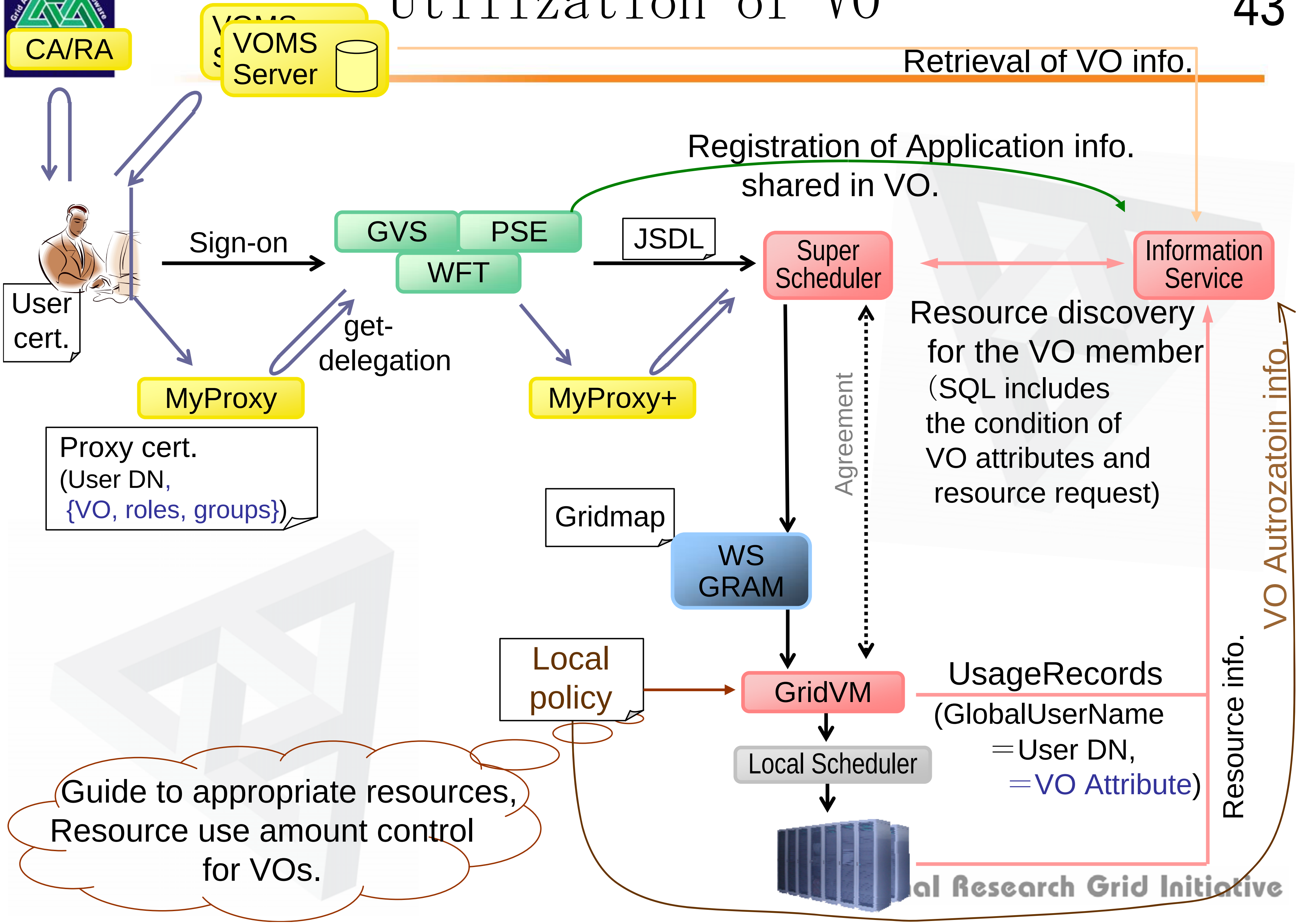
・The policy information is reflected in the Information Service.

・The Super Scheduler tries to find resources with the condition of users' attributes in VO and requests about resource usage.

・GridVM registers Usage Records to Information Service.

・PSE registers association between VOs and deployed application to IS.

# Utilization of VO



CA/RA

VOMS Server

Retrieval of VO info.

Registration of Application info. shared in VO.

Sign-on

GVS

PSE

WFT

JSDL

Super Scheduler

Information Service

User cert.

get-delegation

MyProxy

MyProxy+

Proxy cert. (User DN, {VO, roles, groups})

Resource discovery for the VO member （SQL includes the condition of VO attributes and resource request)

Agreement

VO Autrozatoin info.

Gridmap

WS GRAM

Local policy

GridVM

UsageRecords (GlobalUserName ＝User DN, ＝VO Attribute)

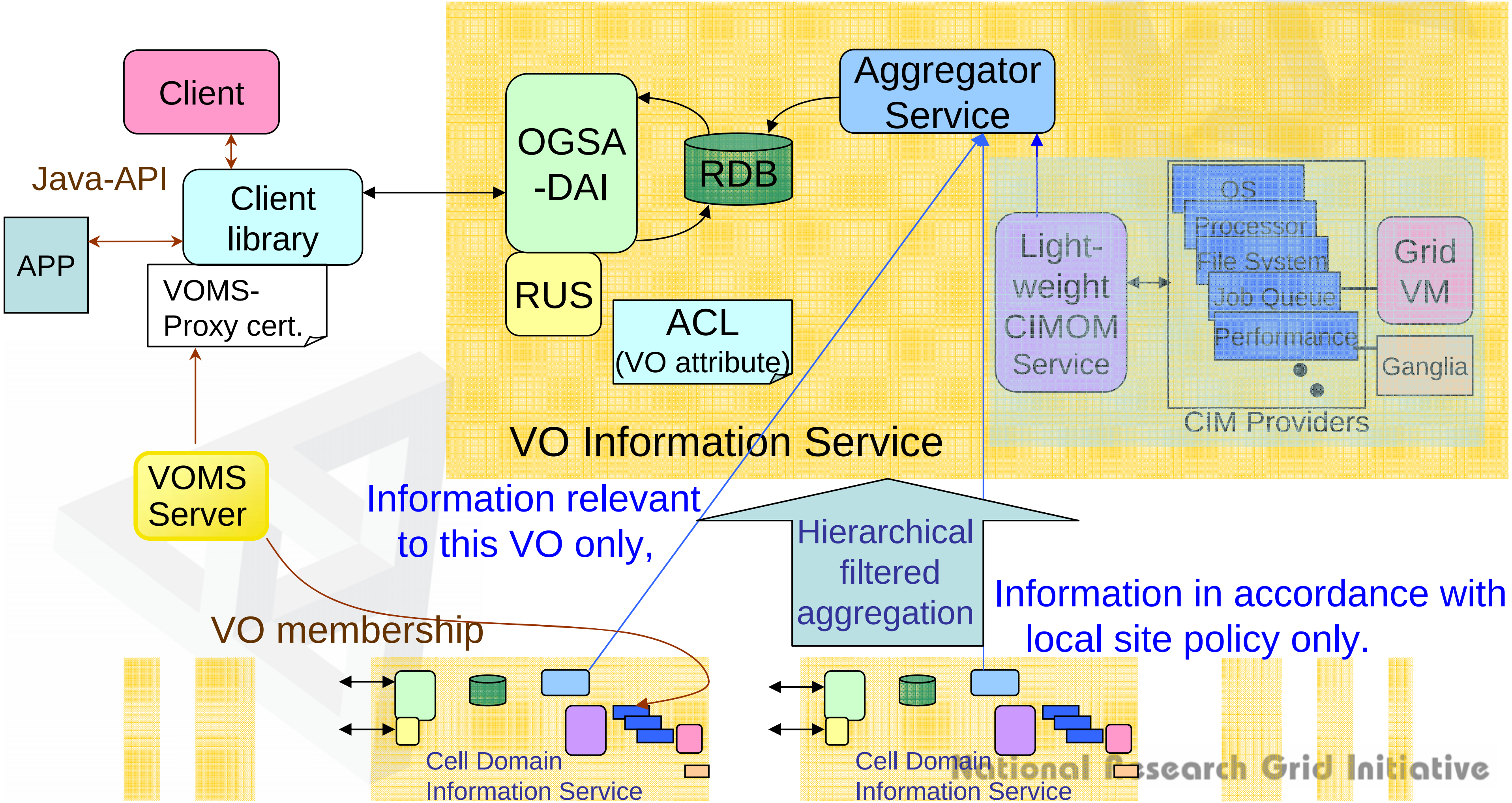
Resource info.

Local Scheduler

Guide to appropriate resources, Resource use amount control for VOs.

# VO Information Service

An Information Service Node that extracts information relevant to the VO from "Cell Domains" with appropriate filter of aggregation.



Information relevant to this VO only,

Information in accordance with local site policy only.

# Local AuthZ info. in Information Service

## (1) Extension of Account and JobQueue class

NRG_VomsAccount class,

where Name attribute is fqan.

(fully qualified attribute name)

/VO[/group[/subgroup(s)]][/Role=role][Capability=cap]

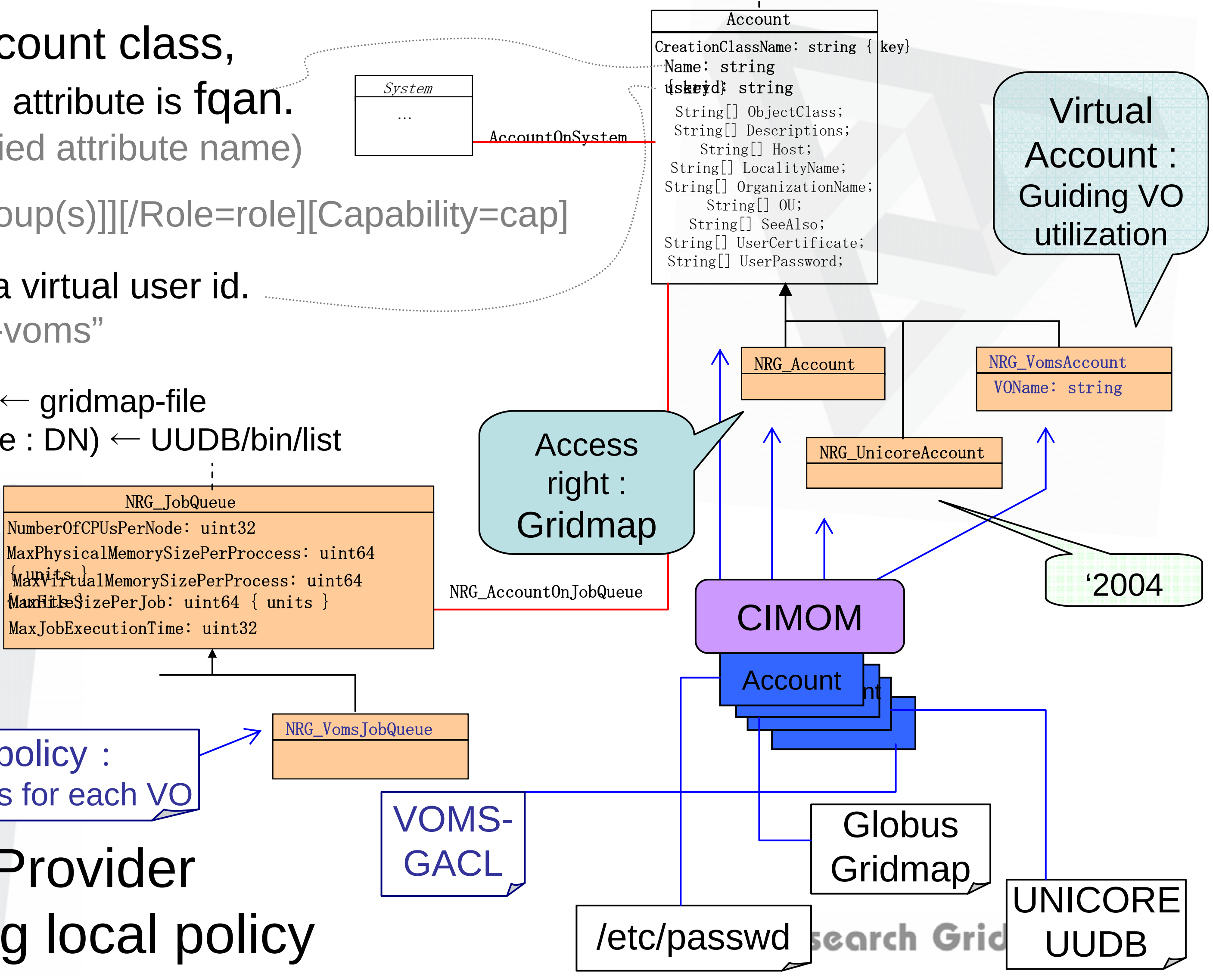
Userid attribute is a virtual user id.

e.g. “naregi-voms”

c.f.

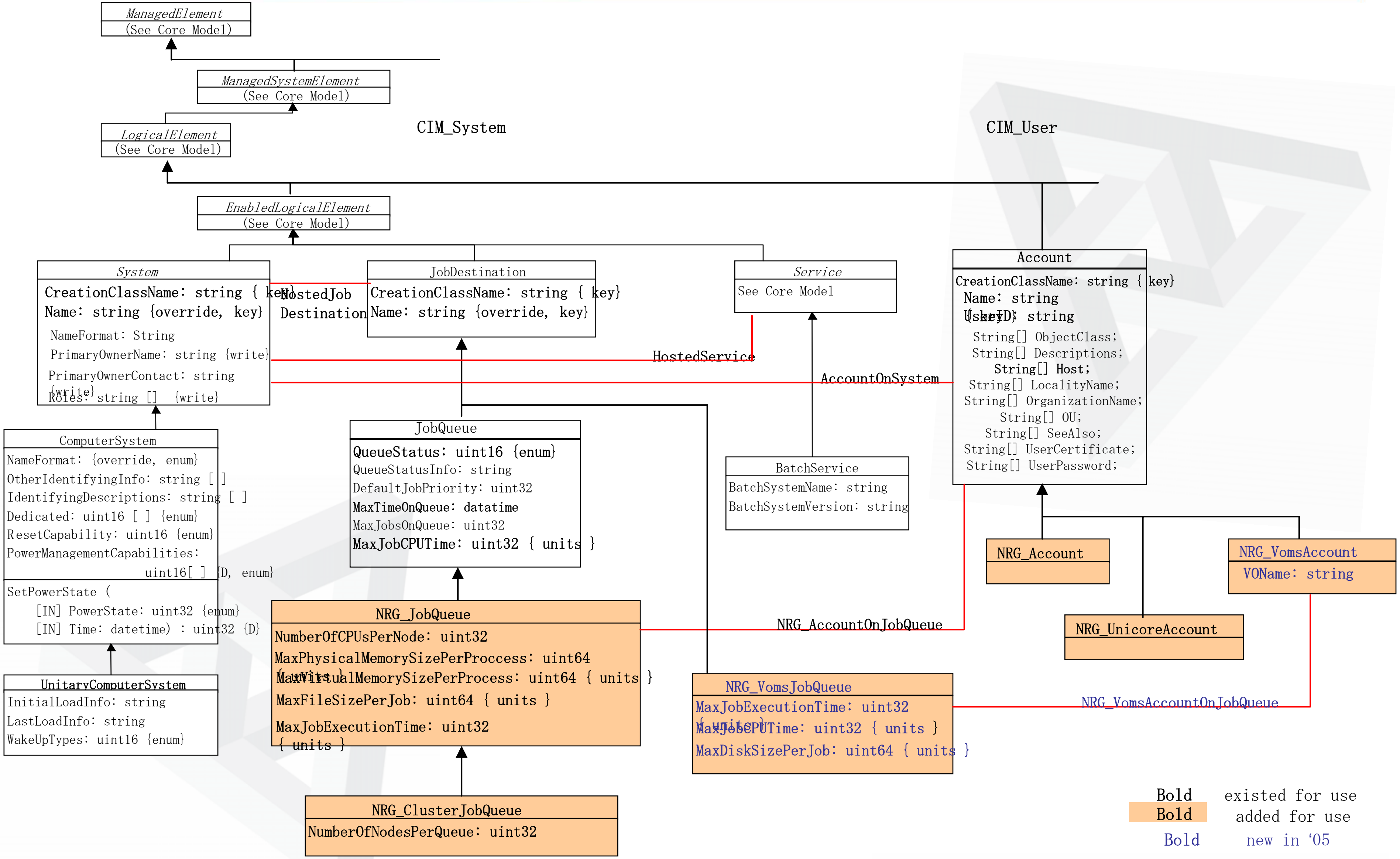
NRG_Account (Name : DN) ← gridmap-file

NRG_UnicoreAccount (Name : DN) ← UUDB/bin/list



## (2) Information Provider interpreting local policy

# Local AuthZ information Schema



*ManagedElement*
(See Core Model)

*ManagedSystemElement*
(See Core Model)

*LogicalElement*
(See Core Model)

CIM_System

CIM_User

*EnabledLogicalElement*
(See Core Model)

*System*
CreationClassName: string { key}
Name: string {override, key}
NameFormat: String
PrimaryOwnerName: string {write}
PrimaryOwnerContact: string {write}
Roles: string [] {write}

HostedJob Destination

JobDestination
CreationClassName: string { key}
Name: string {override, key}

*Service*
See Core Model

HostedService

AccountOnSystem

Account
CreationClassName: string { key}
Name: string
UserID: string {key}
String[] ObjectClass;
String[] Descriptions;
**String[] Host;**
String[] LocalityName;
String[] OrganizationName;
String[] OU;
String[] SeeAlso;
String[] UserCertificate;
String[] UserPassword;

ComputerSystem
NameFormat: {override, enum}
OtherIdentifyingInfo: string [ ]
IdentifyingDescriptions: string [ ]
Dedicated: uint16 [ ] {enum}
ResetCapability: uint16 {enum}
PowerManagementCapabilities: uint16[ ] {D, enum}
SetPowerState (
[IN] PowerState: uint32 {enum}
[IN] Time: datetime) : uint32 {D}

JobQueue
**QueueStatus: uint16 {enum}**
QueueStatusInfo: string
DefaultJobPriority: uint32
**MaxTimeOnQueue: datatime**
MaxJobsOnQueue: uint32
**MaxJobCPUTime: uint32 { units }**

BatchService
BatchSystemName: string
BatchSystemVersion: string

NRG_Account

NRG_VomsAccount
VOName: string

NRG_JobQueue
NumberOfCPUsPerNode: uint32
MaxPhysicalMemorySizePerProccess: uint64 { units }
MaxVirtualMemorySizePerProcess: uint64 { units }
MaxFileSizePerJob: uint64 { units }
MaxJobExecutionTime: uint32 { units }

NRG_AccountOnJobQueue

NRG_UnicoreAccount

UnitaryComputerSystem
InitialLoadInfo: string
LastLoadInfo: string
WakeUpTypes: uint16 {enum}

NRG_VomsJobQueue
MaxJobExecutionTime: uint32 { units }
MaxJobCPUTime: uint32 { units }
MaxDiskSizePerJob: uint64 { units }

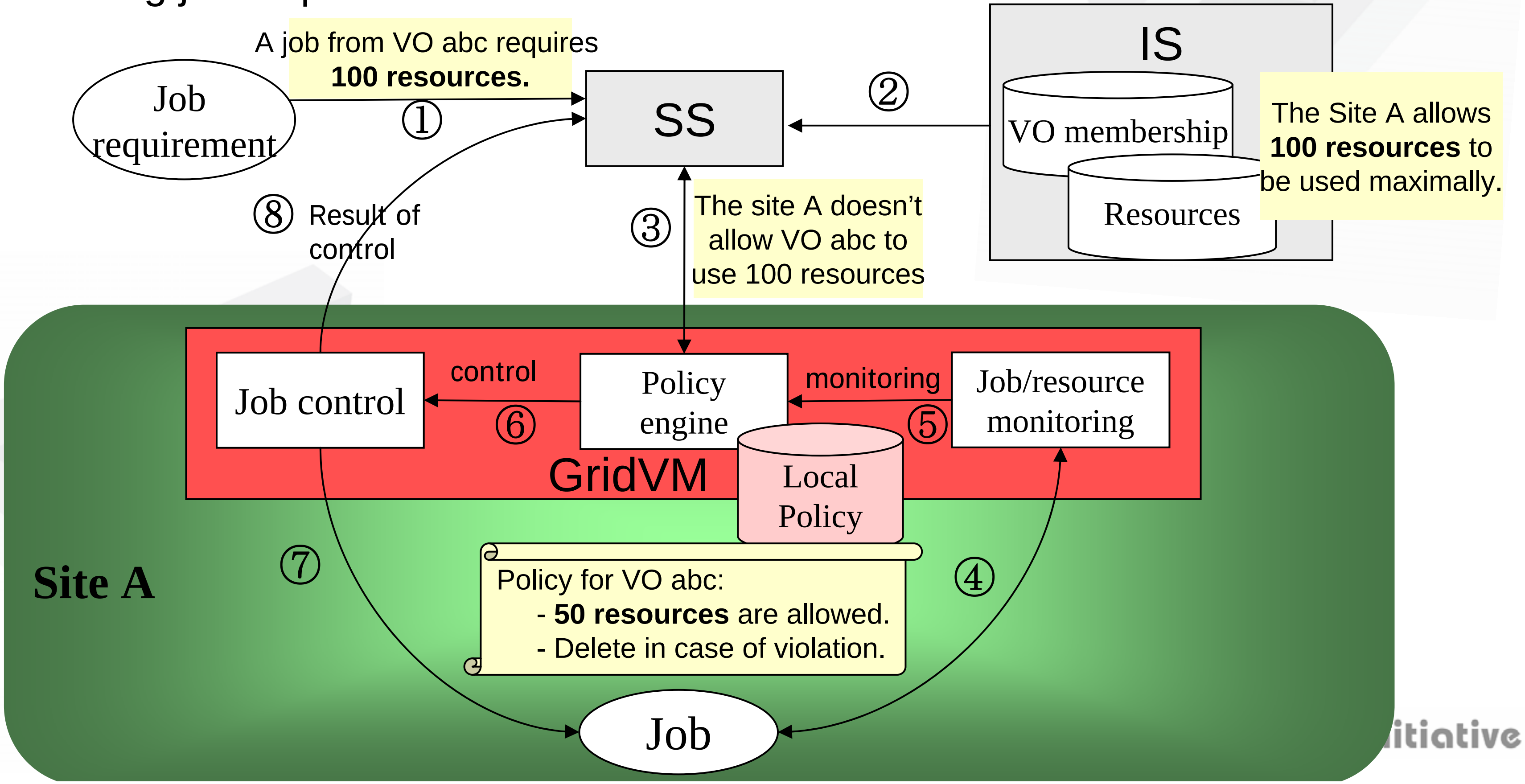
NRG_VomsAccountOnJobQueue

NRG_ClusterJobQueue
NumberOfNodesPerQueue: uint32

| | |
|---|---|
| **Bold** | existed for use |
| **Bold** | added for use |
| **Bold** | new in '05 |

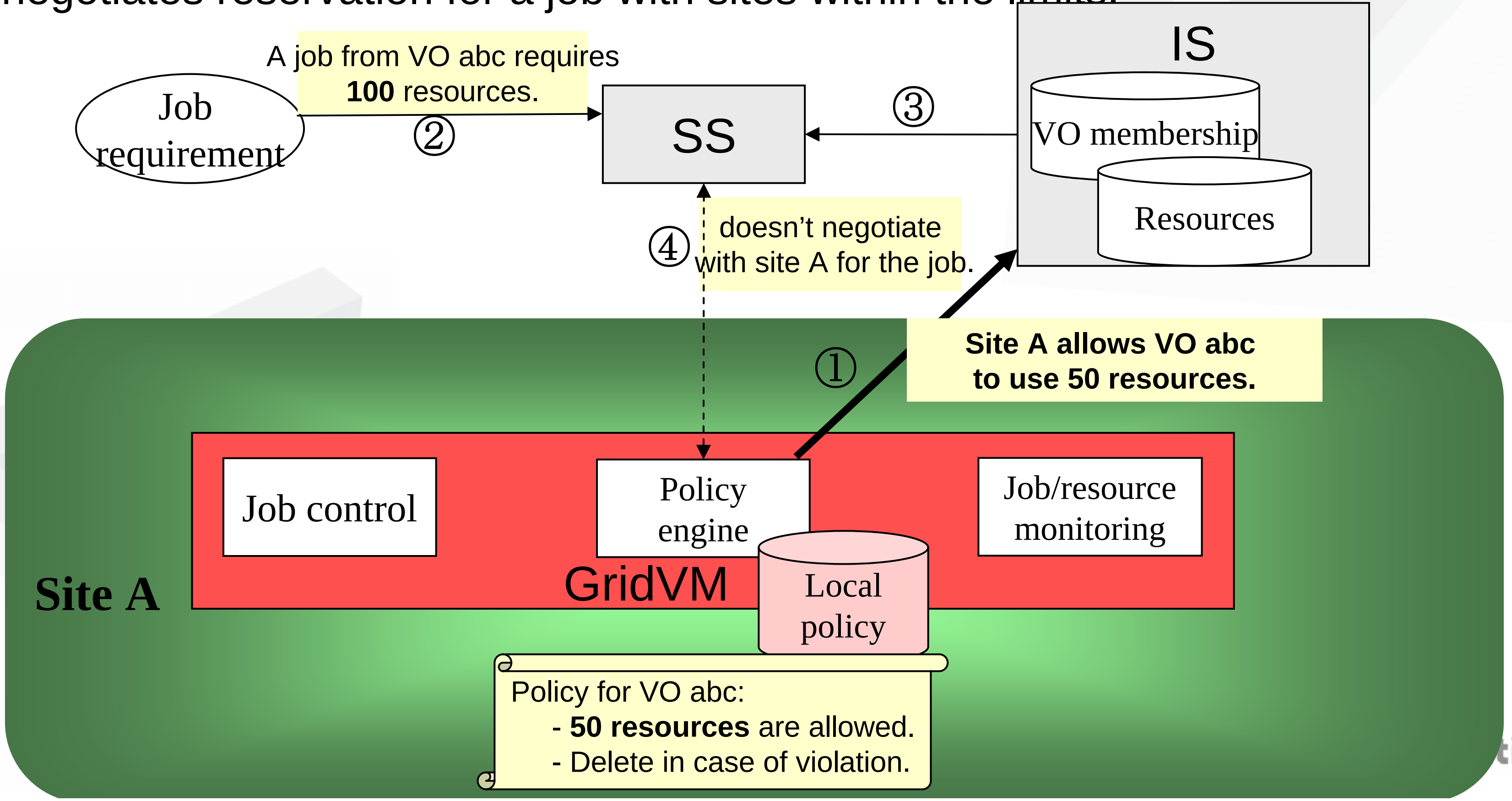
# Resource Usage Control for VO

- GridVM monitors and controls resource use amount of each job according to local policy.
- In case Information Service **doesn't** know about the local policy,
  - the reservation request is refused even if the Super Scheduler decided the site matching job requirement and resource information in the Information Service.


Job requirement
A job from VO abc requires **100 resources.**
①
SS
②
IS
VO membership
Resources
The Site A allows **100 resources** to be used maximally.
⑧ Result of control
③
The site A doesn't allow VO abc to use 100 resources
Job control
control
⑥
Policy engine
monitoring
⑤
Job/resource monitoring
GridVM
Local Policy
Site A
⑦
Policy for VO abc:
- **50 resources** are allowed.
- Delete in case of violation.
④
Job

# Brokering of VO jobs with Local policy

- GridVM provides Information Service information about amount of resources in each site allowing each VO to use.
  - Limits of Wall time, CPU time and Disk Size for a job executed in each site.
- Super Scheduler refers to it for resource brokering,
  - negotiates reservation for a job with sites within the limits.


Job requirement
A job from VO abc requires **100** resources.
②
SS
③
IS
VO membership
Resources
④
doesn't negotiate with site A for the job.
①
**Site A allows VO abc to use 50 resources.**
Job control
Policy engine
Job/resource monitoring
GridVM
Local policy
Site A

Policy for VO abc:
- **50 resources** are allowed.
- Delete in case of violation.

# Accounting for VO utilization

```
<?xml version="1.0" encoding="UTF-8"?>
<JobUsageRecord xmlns="http://www.gridforum.org/2003/ur-wg"
    xmlns:urwg="http://www.gridforum.org/2003/ur-wg"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.gridforum.org/2003/ur-wg file:/Users/bekah/Documents/GGF/URWG/urwg-schema.09.xsd">
  <RecordIdentity urwg:recordId="http://pbg2002.naregi.org/PBS.1234.0" urwg:createTime="2005-07-11T14:26:56Z" />
  <JobIdentity>
    <GlobalJobId>87461495154</LocalJobId>
    <LocalJobId>PBS.1234.0</LocalJobId>
  </JobIdentity>
  <UserIdentity>
    <LocalUserId>unicore</LocalUserId>
    <GlobalUserName>
      EMAILADDRESS=ysaeki@grid.nii.ac.jp, CN=Yuji Saeki, O=National Research Grid Initiative, C=JP
    </GlobalUserName>
  </UserIdentity>
  <UserIdentity>
    <LocalUserId>naregi-voms</LocalUserId>
    <GlobalUserName>/naregi.org/wp1/info-service/Role=Developer</GlobalUserName>
  </UserIdentity>
  <UserIdentity>
    <LocalUserId>naregi-voms</LocalUserId>
    <GlobalUserName>/naregi.org/office/Role=Staff</GlobalUserName>
  </UserIdentity>
  <Status>completed</Status>
  <Memory urwg:storageUnit="MB">1234</Memory>
  <Processors>4</Processors>
  <NodeCount>2</NodeCount>
      …
      …
</JobUsageRecord>
```

multiple UserIdentity in a Usage Record :
… set of {LocalUserId, GlobalUserName}

# Application for VO utilization

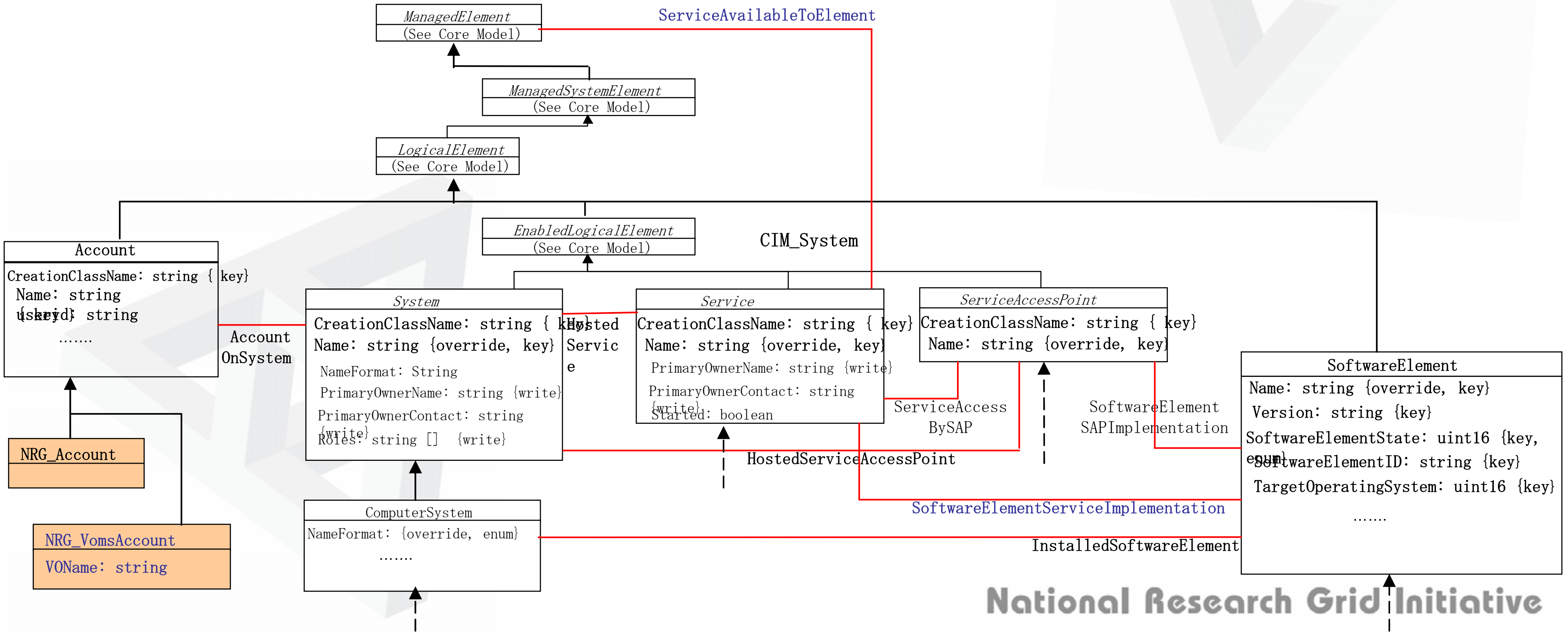
- Sharing among VO members

  … what members（group, role） are allowed to execute it

  → expressed in Association class with NRG_VomsAccount

- Selection of Application/Software managed by system admins,

  → filter to aggregate information to VO Information Service.


ManagedElement
(See Core Model)
ServiceAvailableToElement
ManagedSystemElement
(See Core Model)
LogicalElement
(See Core Model)
EnabledLogicalElement
(See Core Model)
CIM_System
Account
CreationClassName: string { key}
Name: string {key}
userid: string
…….
Account
OnSystem
NRG_Account
NRG_VomsAccount
VOName: string
System
CreationClassName: string { key}
Name: string {override, key}
NameFormat: String
PrimaryOwnerName: string {write}
PrimaryOwnerContact: string {write}
Roles: string [] {write}
Hosted
Service
Service
CreationClassName: string { key}
Name: string {override, key}
PrimaryOwnerName: string {write}
PrimaryOwnerContact: string {write}
Started: boolean
ServiceAccessPoint
CreationClassName: string { key}
Name: string {override, key}
ServiceAccess
BySAP
SoftwareElement
SAPImplementation
HostedServiceAccessPoint
SoftwareElementServiceImplementation
ComputerSystem
NameFormat: {override, enum}
…….
InstalledSoftwareElement
SoftwareElement
Name: string {override, key}
Version: string {key}
SoftwareElementState: uint16 {key, enum}
SoftwareElementID: string {key}
TargetOperatingSystem: uint16 {key}
…….

# Summary

We developed Information Service in Cell Domain in ‘2003 as a component of NAREGI Information Service.

CIM & RDB on GT3 using open

2004～

Proper implementation of Grid Info. Service ...

RDB centric with Lightweight CIMOM,

Scalable monitoring (multi-domain),

Secure accounting (Access control),

Interface to NAREGI MiddleWare,

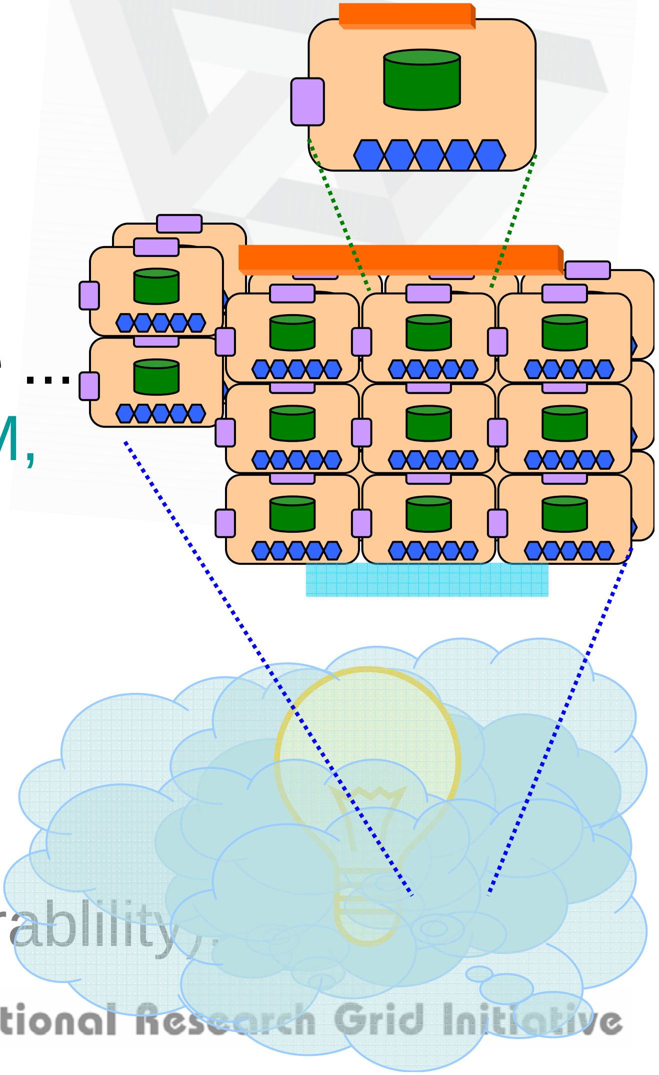
2006～

Virtual Organization Management …

OGSA Information Service (inter-operablility),

VO hosting service,

Support for stable management



# Low Hanging Fruit
# “Just make it work by GLUEing”

- Identify the minimum common set of information required for interoperation in the respective information service
- Employ GLUE and extended CIM as the base schema for respective grids
- Each Info service in grid acts as a information provider for the other
- Embed schema translator to perform schema conversion
- Present data in a common fashion on each grid ; WebMDS, NAREGI CIM Viewer, SCMSWeb, …

# 参考

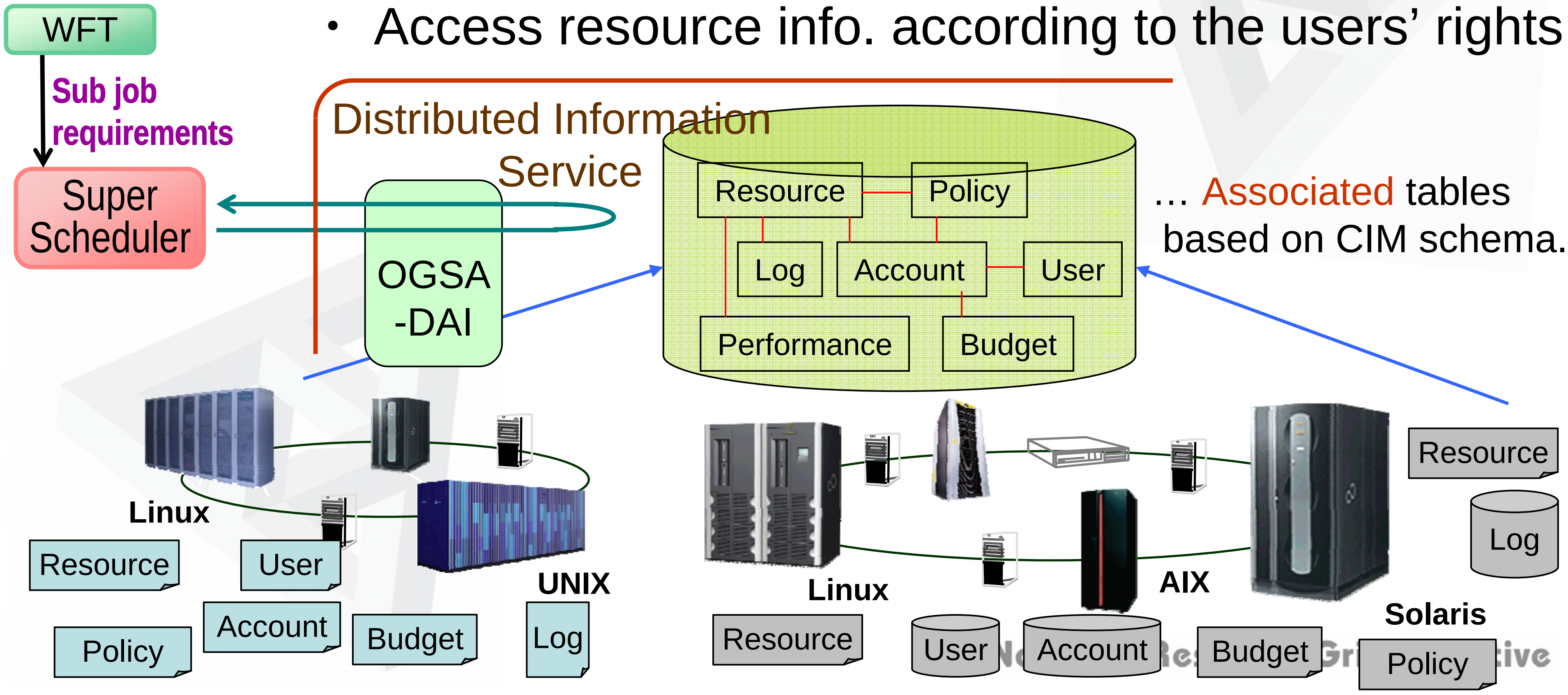
# Resource Discovery

Distributed Info.Services maintain various kind of information;

CPU, Memory, OS, Job Queue, Account, Usage Record, etc.etc.

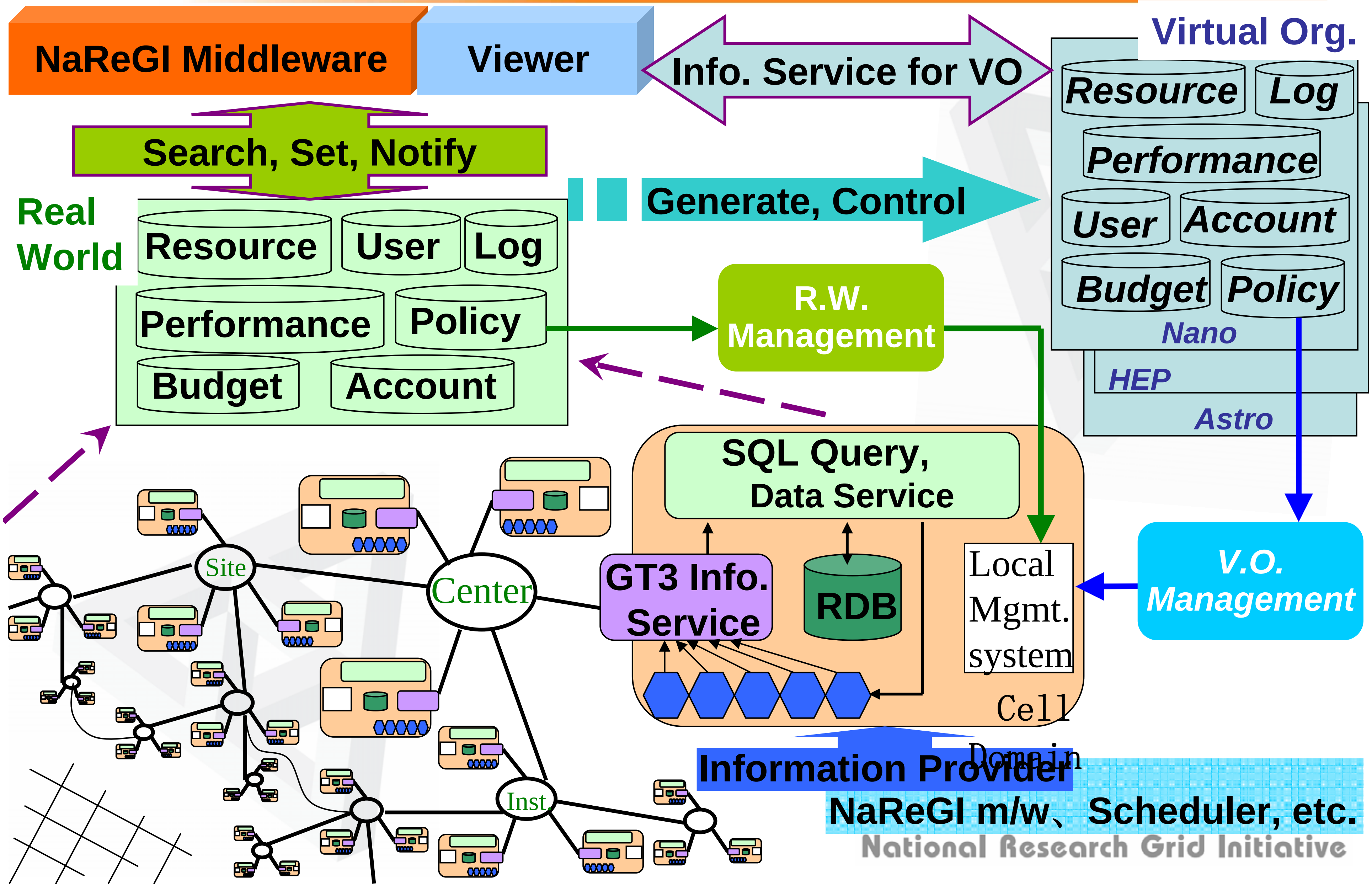
across multiple administrative domains,

- Abstract heterogeneous resources (CIM schema) → RD
- Retrieve resource DB through Grid Service(OGSA-DA
- Access resource info. according to the users' rights.



… Associated tables based on CIM schema.

# Distributed Information Service ①



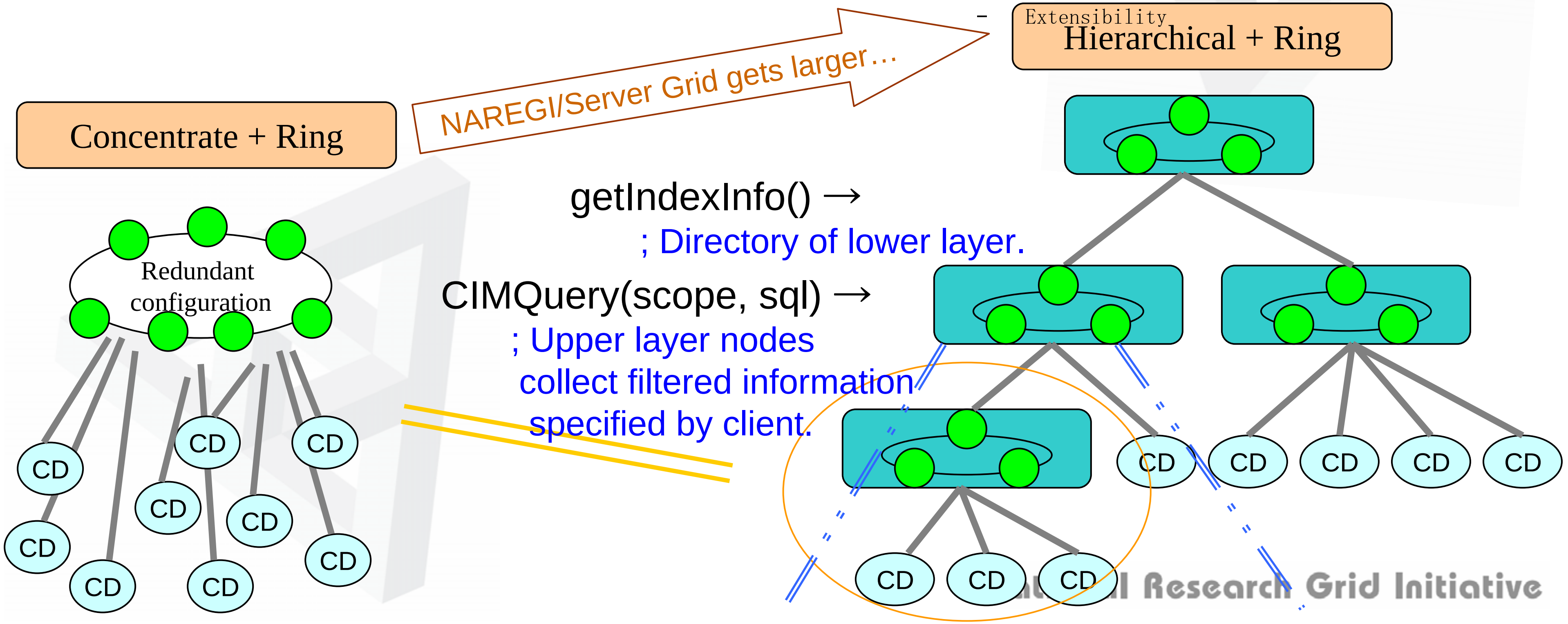
# Example ; SS→IS （α version)

```
SELECT DISTINCT ON ("MaxNumberOfNodes","Hostname", "QueueName")
            "Hostname","UsiteName","UsitePort","VsiteName","QueueName",
            "MaxNumberOfNodes","UserName","UserID"
FROM "BrokeringTable"
WHERE (("SoftName" = 'gcc') AND ("SoftMajorNumber" >= 3) AND
          ("SoftMinorNumber" >=2) AND ("SoftRevisionNumber" >= 0)) AND
     (("PMemory" >= 32768)) AND (("VMemory" >= 32768)) AND
     (("CPU" = 179)) AND (("MaxNumberOfNodes" >= 7)) AND
     (("TasksPerHost" >= 2)) AND
     (("UserName" = 'EMAILADDRESS=ysaeki@grid.nii.ac.jp, CN=Yuji Saeki,
                              O=National Research Grid Initiative, C=JP')) AND
     (("Hostname" != 'pbg1012.naregi.org'))
  ORDER BY "MaxNumberOfNodes" ASC LIMIT 10;
```

# Multi-Domain connection

- NAREGI is a Server Grid,
  - managed by managers of resource pools.
- Resource pool is
  - Large scale,
  - relatively Static, however,
  - composed of Multiple administrative domains.

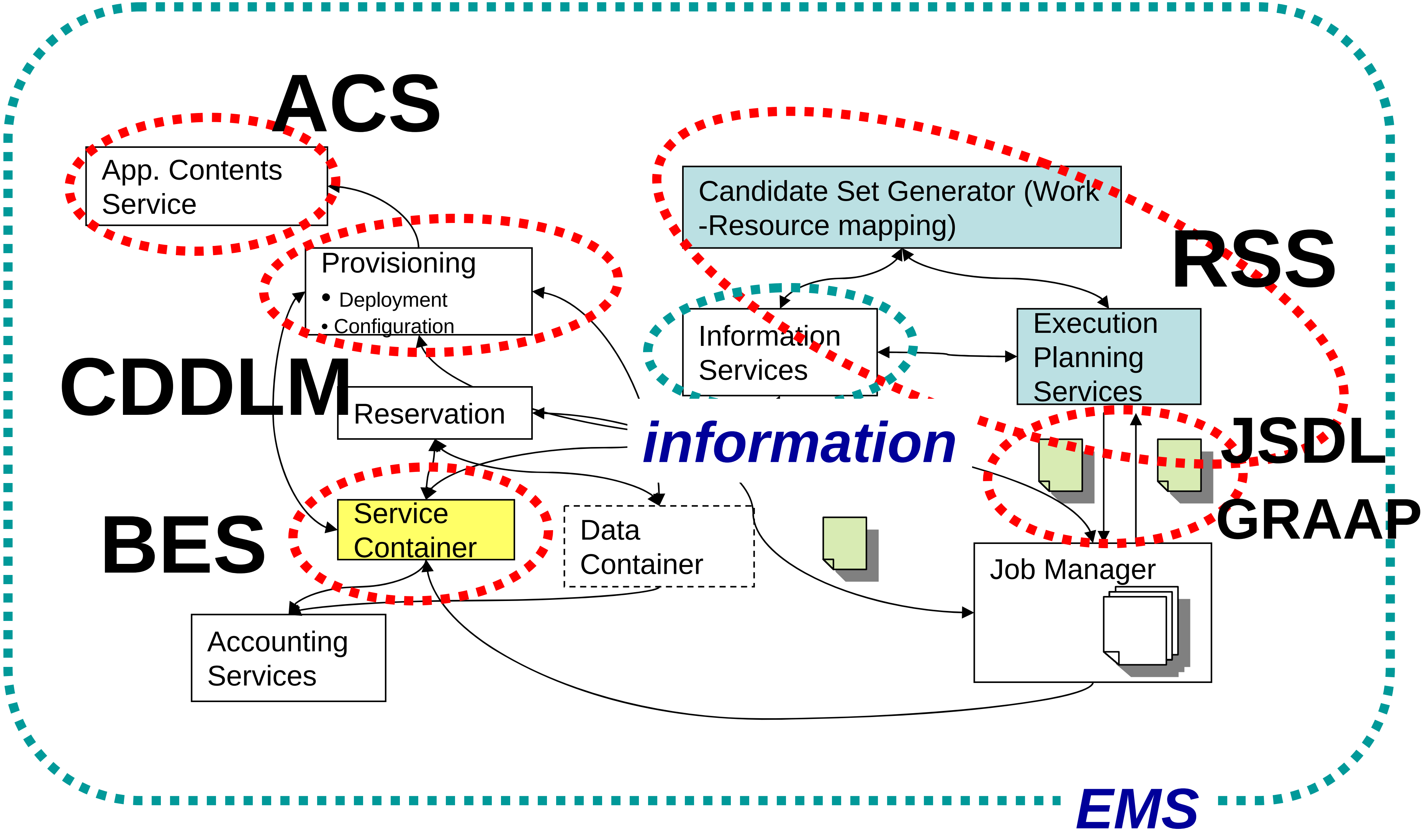
- Points in terms of topology are
  - Scalability
  - Managability
  - Fault Tolerance
  - Information Coherence
  - Security
  - Extensibility



getIndexInfo() →
; Directory of lower layer.

CIMQuery(scope, sql) →
; Upper layer nodes collect filtered information specified by client.

# OGSA-EMS: Collaborative Work Example



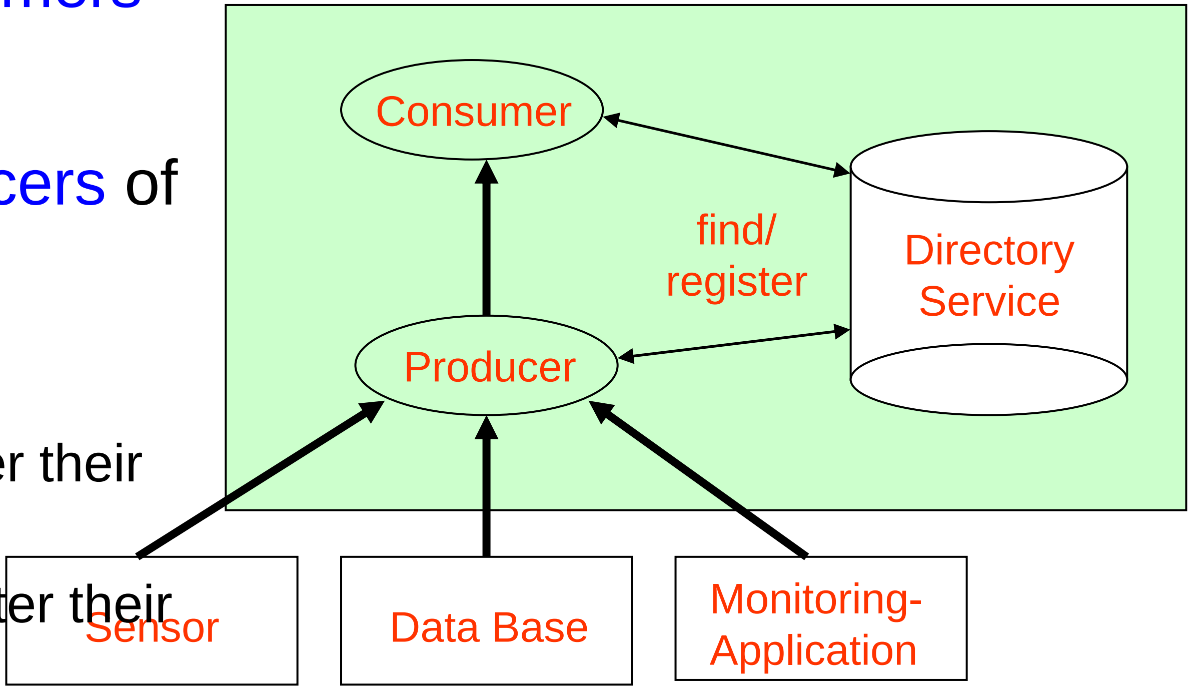
# CIM in OGSA

- CIM is the information model that best satisfies the requirements
  - “Low barrier of entry” is a concern
- OGSA-WG intends to use the CIM “framework”
  - Details and further commitment need more work
  - Data model TBD
    - OASIS WSDM and corresponding CIM mapping are candidates for the WSRF basic profile
- Information models specs are possibly CIM profiles plus OGSA extensions, plus guideline doc

# A “Virtual” distributed data warehouse

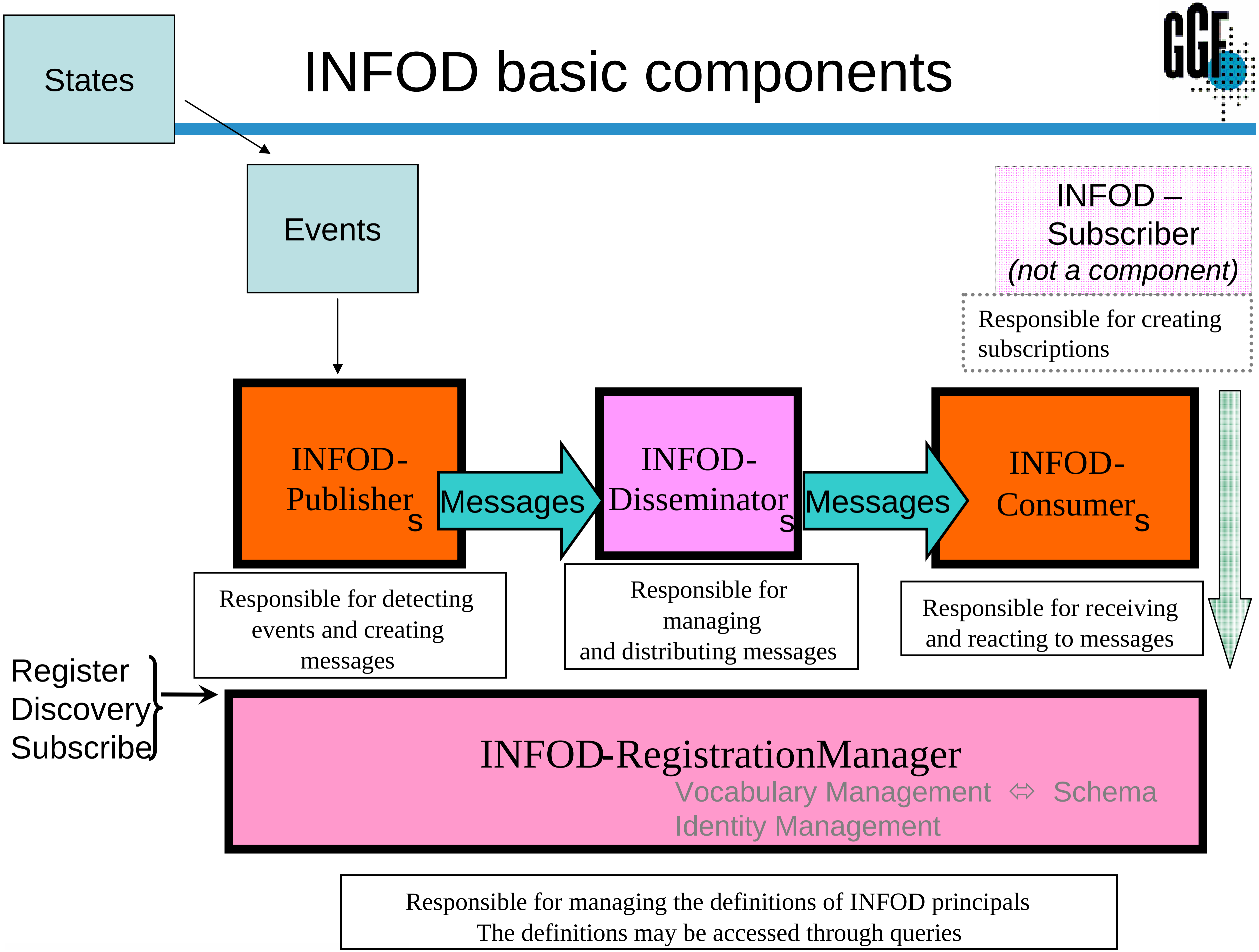
The *Grid Monitoring Architecture* (GMA) of the Global Grid Forum distinguishes between:

- Distributed Consumers of information
- Distributed Producers of information
- Directory Service
  - Producers register their *supply*
  - Consumers register their *demand*



GMA separates matching of consumers to producers and delivery of data from producers to consumers

# INFOD basic components



Responsible for creating subscriptions

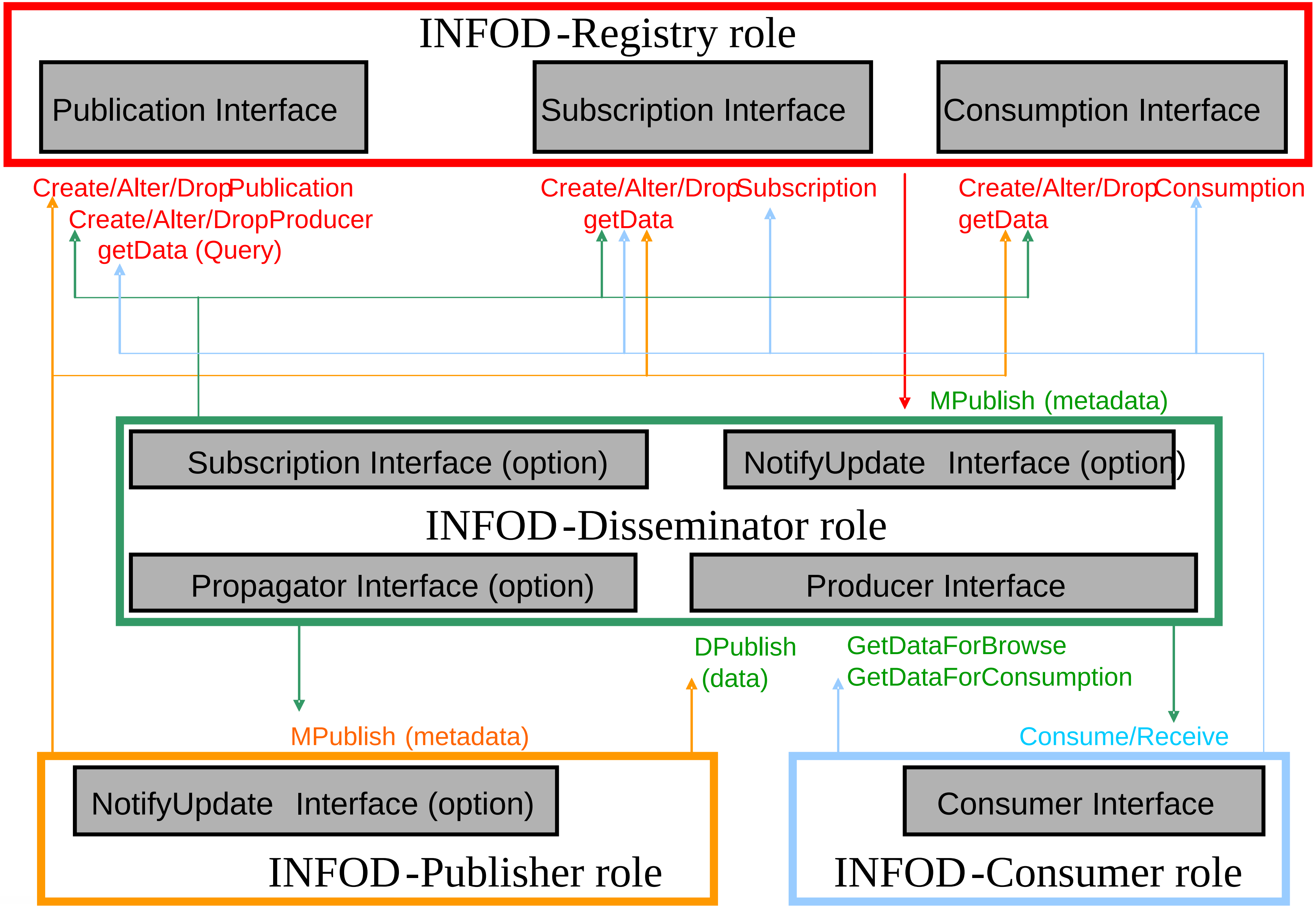
Responsible for detecting events and creating messages

Responsible for managing and distributing messages

Responsible for receiving and reacting to messages

Responsible for managing the definitions of INFOD principals
The definitions may be accessed through queries

# INFOD Interfaces



INFOD-Registry role

Publication Interface

Subscription Interface

Consumption Interface

Create/Alter/Drop Publication

Create/Alter/Drop Producer

getData (Query)

Create/Alter/Drop Subscription

getData

Create/Alter/Drop Consumption

getData

MPublish (metadata)

Subscription Interface (option)

NotifyUpdate Interface (option)

INFOD-Disseminator role

Propagator Interface (option)

Producer Interface

DPublish (data)

GetDataForBrowse

GetDataForConsumption

MPublish (metadata)

Consume/Receive

NotifyUpdate Interface (option)

INFOD-Publisher role

Consumer Interface

INFOD-Consumer role