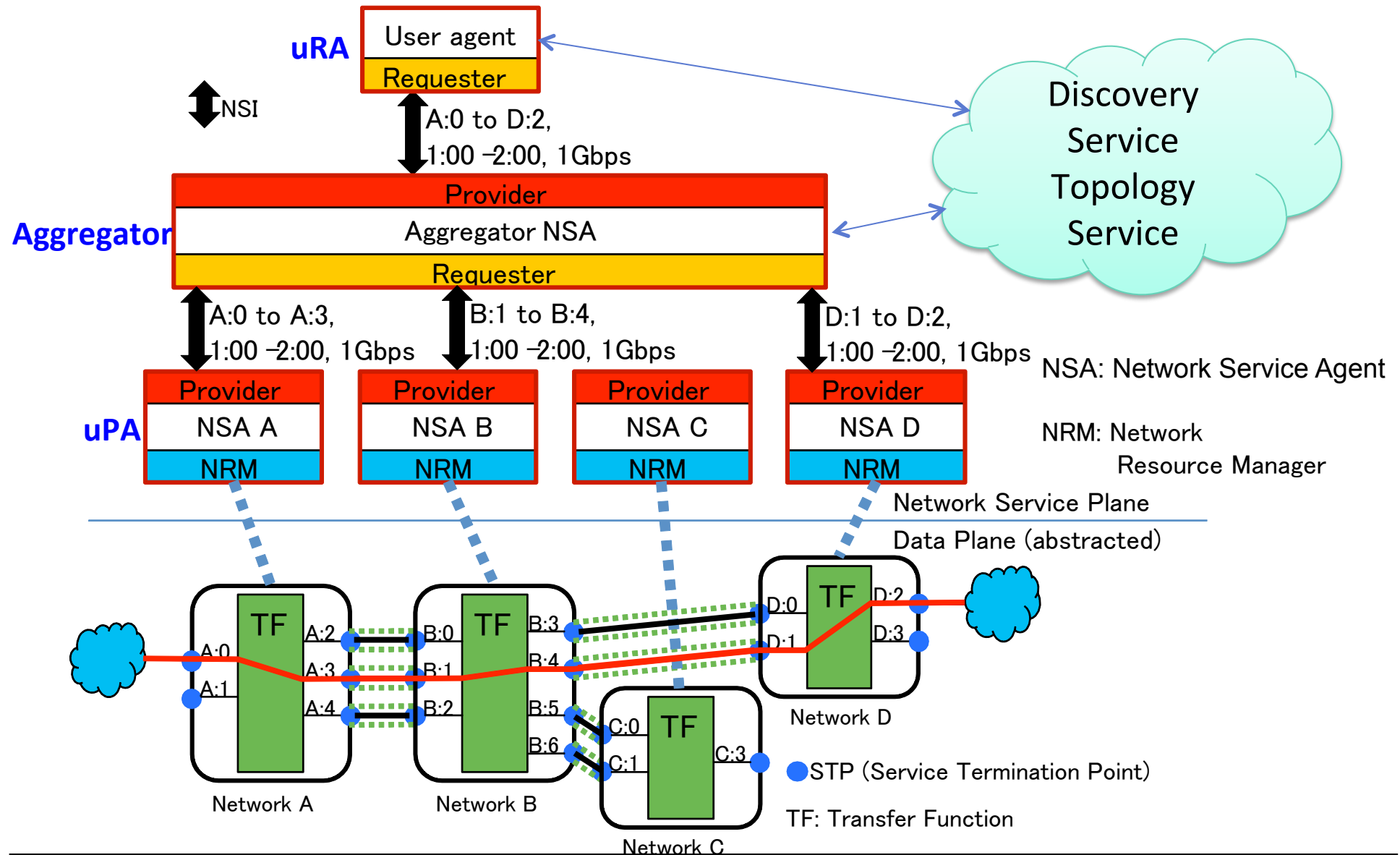


# G-lambda/A Implementations and Tools for NSI CS v. 2.0

Atsuko Takefusa

National Institute of Advanced Industrial  
Science and Technology (AIST)

# NSI Architecture and the Connection Service (CS) Protocol

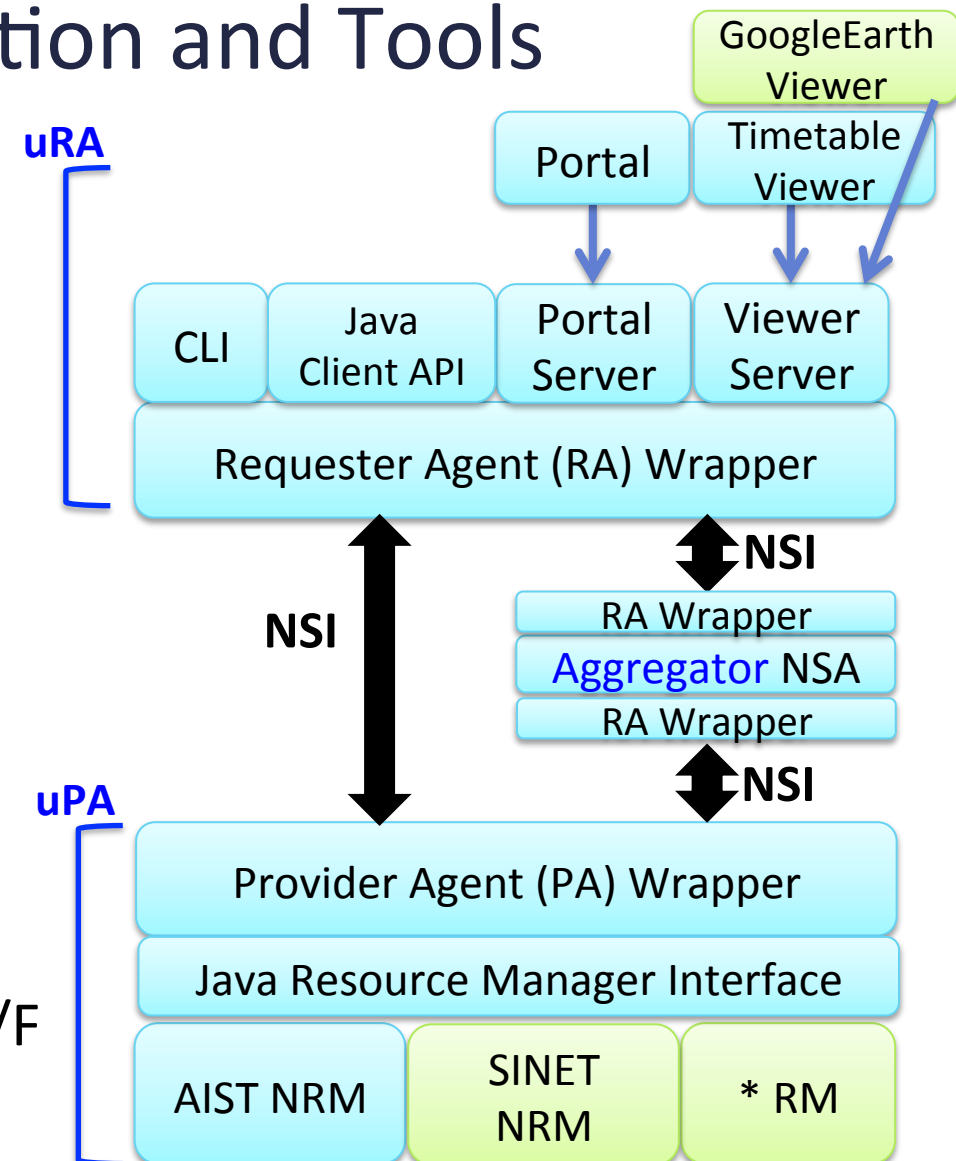


# Contributions

- AIST has developed G-lambda/A NSA implementations and tools for NSI CS v. 2.0
  - uRA modules
  - Aggregator NSA
  - uPA modules
- Our package is an open-source software and used by third parties
  - SINET PA uses our package on top of their own network provisioning system
  - UltraGrid application, Google Earth viewer

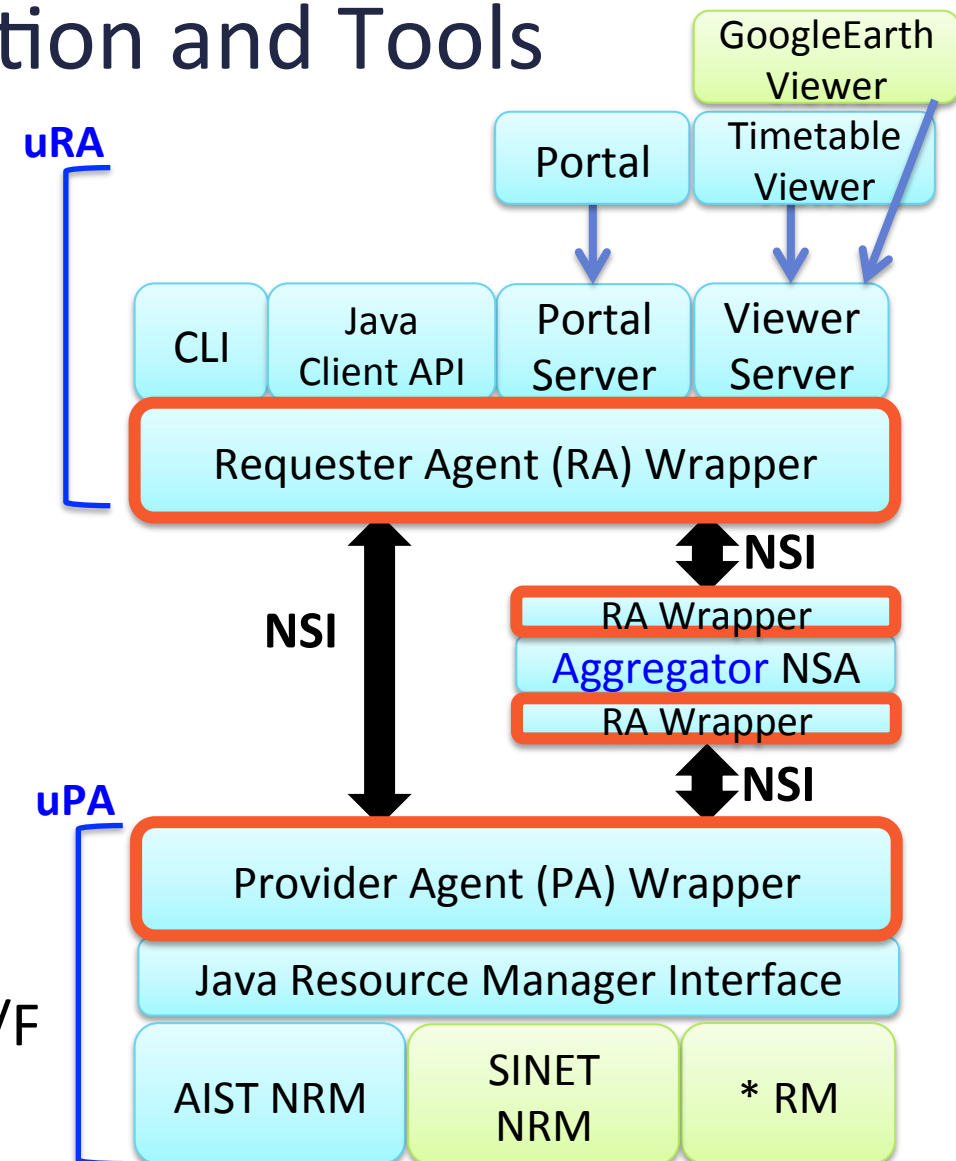
# Overview of the G-lambda/A Implementation and Tools

- RA/PA Wrappers
- uRA modules
  - CLI
  - Java Client API
  - Portal Server and Portal
  - Viewer Server and Timetable Viewer
- Aggregator NSA
- uPA modules
  - Java Resource Manager I/F
  - AIST NRM



# Overview of the G-lambda/A Implementation and Tools

- RA/PA Wrappers
- uRA modules
  - CLI
  - Java Client API
  - Portal Server and Portal
  - Viewer Server and Timetable Viewer
- Aggregator NSA
- uPA modules
  - Java Resource Manager I/F
  - AIST NRM

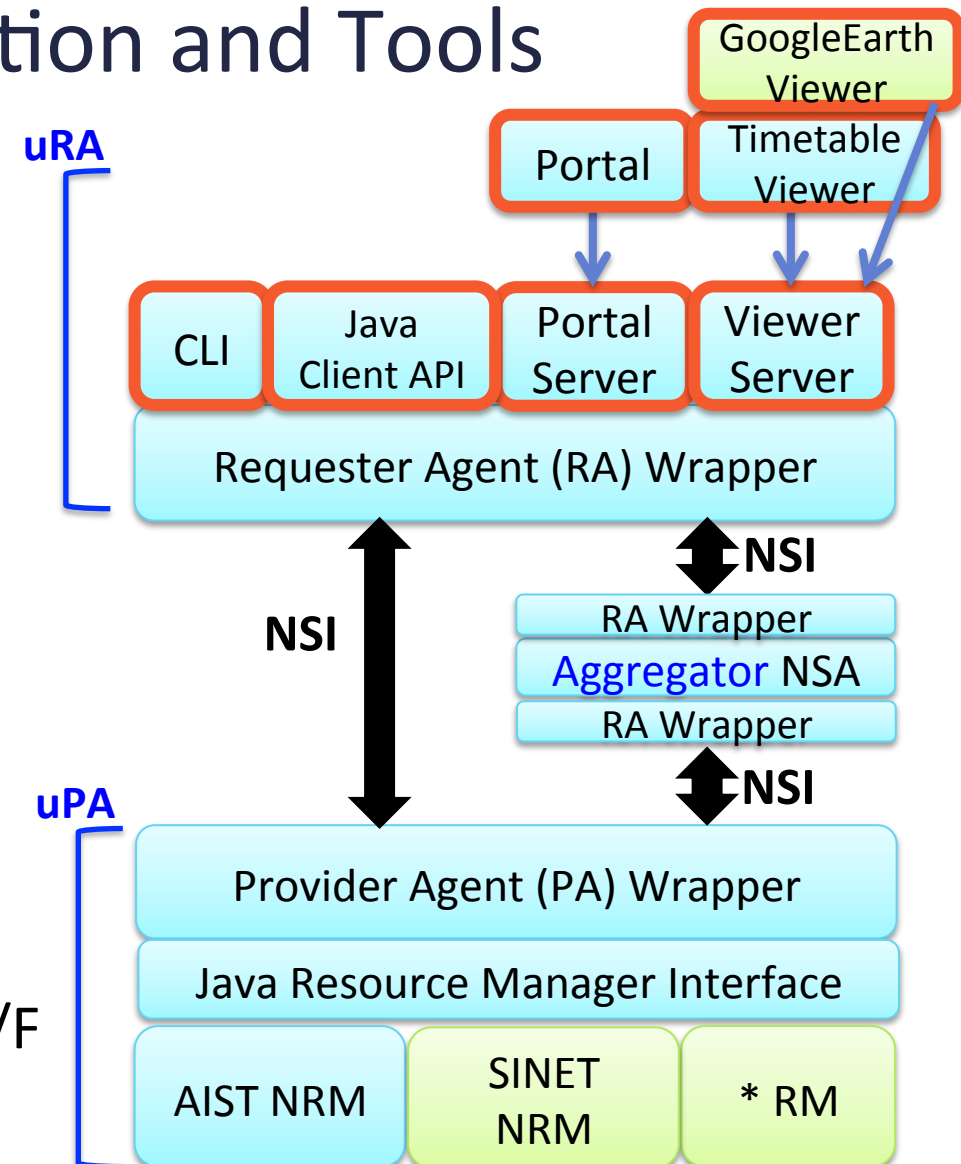


## RA/PA Wrappers

- All of G-lambda/A NSAs written in Java are running on the Jetty Java application server
- RA and PA Wrappers are generated from the NSI CS v. 2.0 wsdl and schemas using Apache CXF, an open-source web services framework

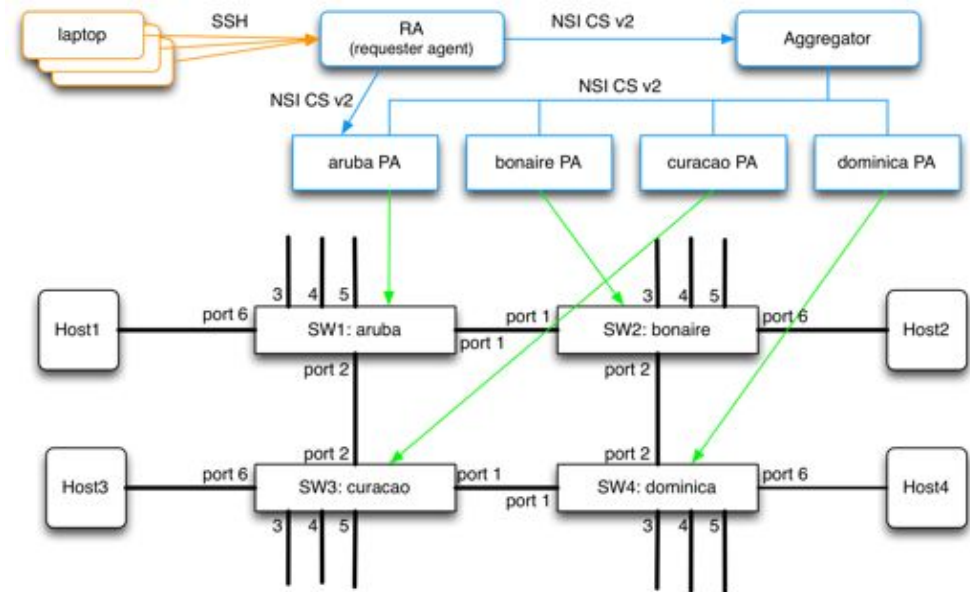
# Overview of the G-lambda/A Implementation and Tools

- RA/PA Wrappers
- uRA modules
  - CLI
  - Java Client API
  - Portal Server and Portal
  - Viewer Server and Timetable Viewer
- Aggregator NSA
- uPA modules
  - Java Resource Manager I/F
  - AIST NRM



# uRA module: CLI (Command Line I/F)

- Enables to send a NSI CS request to Aggregators and uPAs
- Provides commands:  
reserve, commit/abort,  
provision, release, terminate  
and query
- Confirmed, failed and  
notification messages are  
asynchronously shown up  
on the user terminal
- Used at NSI hands-on  
tutorial at TIP2013 and  
APAN36





# uRA module: Java Client API

Pseudo code

- Allows to develop uPA applications without detailed NSI CS knowledges
- Supports to access to PAs with http/https user and password authentication and OAuth2
- Application examples
  - Request Scheduler at GLIF2013 and SC13 demonstrations
  - UltraGrid application developed by CzechLight

```
// Create NSI2Client instance
NSI2Client client = new NSI2Client(
    providerNSA, providerURI, requesterNSA,
    requesterURI, replyWaitMsec, listener);

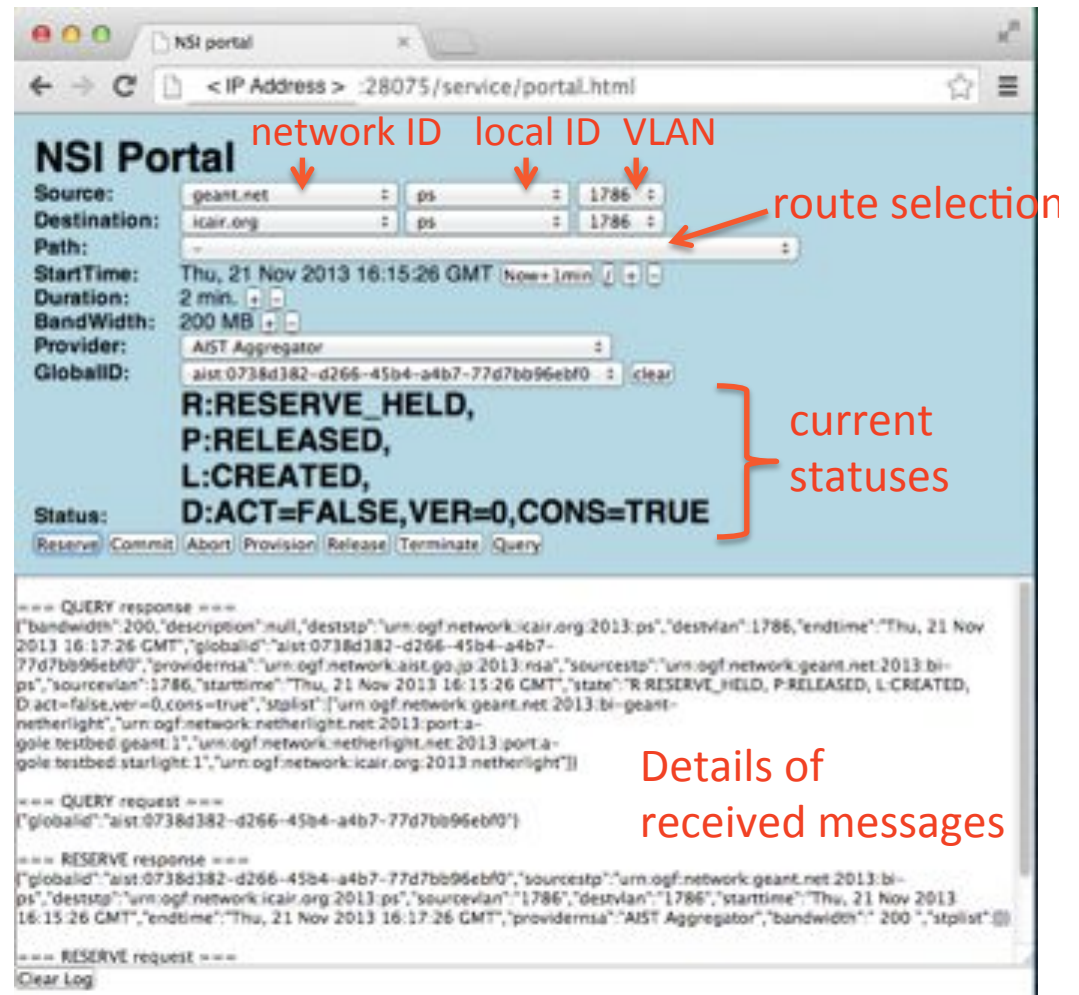
// Create criteria instance
ReservationRequestCriteriaType criteria =
    TypesBuilder.makeReservationRequestCriteriaType(
        schedule, srcstp, vlan, deststp, vlan, capacity);
criteria.setVersion(currentVersion);

// Send reserve request
// rsvReply returns when cf msg has received
ReserveReply rsvReply = client.reserve(
    connectionId, globalReservationId,
    description, criteria);

// Send reserveCommit request
ReserveCommitReply commitReply =
    client.reserveCommit(reply.getConnectionId());
```

# uRA module: Portal Server and Portal

- Portal Server
  - Provides simple REST I/F required for portal
- Portal
  - Provide simple GUI to send, receive and show each NSI CS v. 2 operation



The screenshot shows the NSI Portal web interface. Red arrows and text annotations highlight specific features:

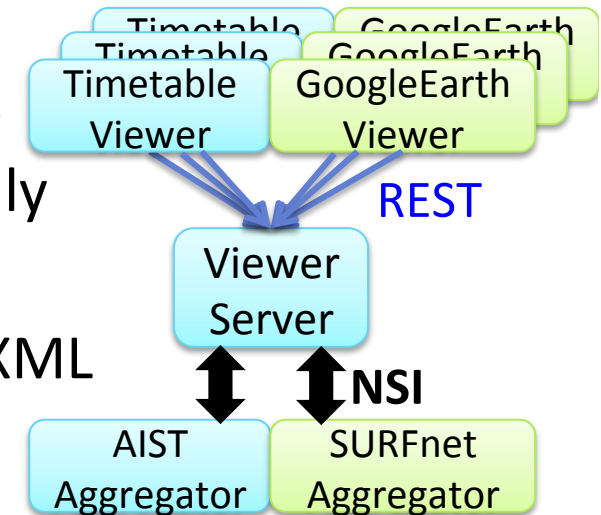
- network ID**: Points to the 'Source' field containing 'geant.net'.
- local ID**: Points to the 'Destination' field containing 'icair.org'.
- VLAN**: Points to the 'Path' field containing 'ps'.
- route selection**: Points to the 'GlobalID' field containing 'aist:0738d382-d266-45b4-a4b7-77d7bb96ebf0'.
- current statuses**: Points to the 'Status' section showing 'R:RESERVE\_HELD, P:RELEASED, L:CREATED, D:ACT=FALSE, VER=0, CONS=TRUE'.
- Details of received messages**: Points to the log area at the bottom showing a 'QUERY response' and a 'RESERVE request'.

The interface includes fields for Source, Destination, Path, StartTime, Duration, BandWidth, Provider, and GlobalID. It also has a 'Status' section with a list of current statuses and a log area for received messages.

# uRA module: Viewer Server and Viewers

## • Viewer Server

- Retrieves all of reservation info from the SURFnet and AIST aggregators periodically
  - via query request w/o connectionId
- Gives viewer instances reservation info XML via REST I/F



## • Viewers at GLIF2013 and SC13 demo

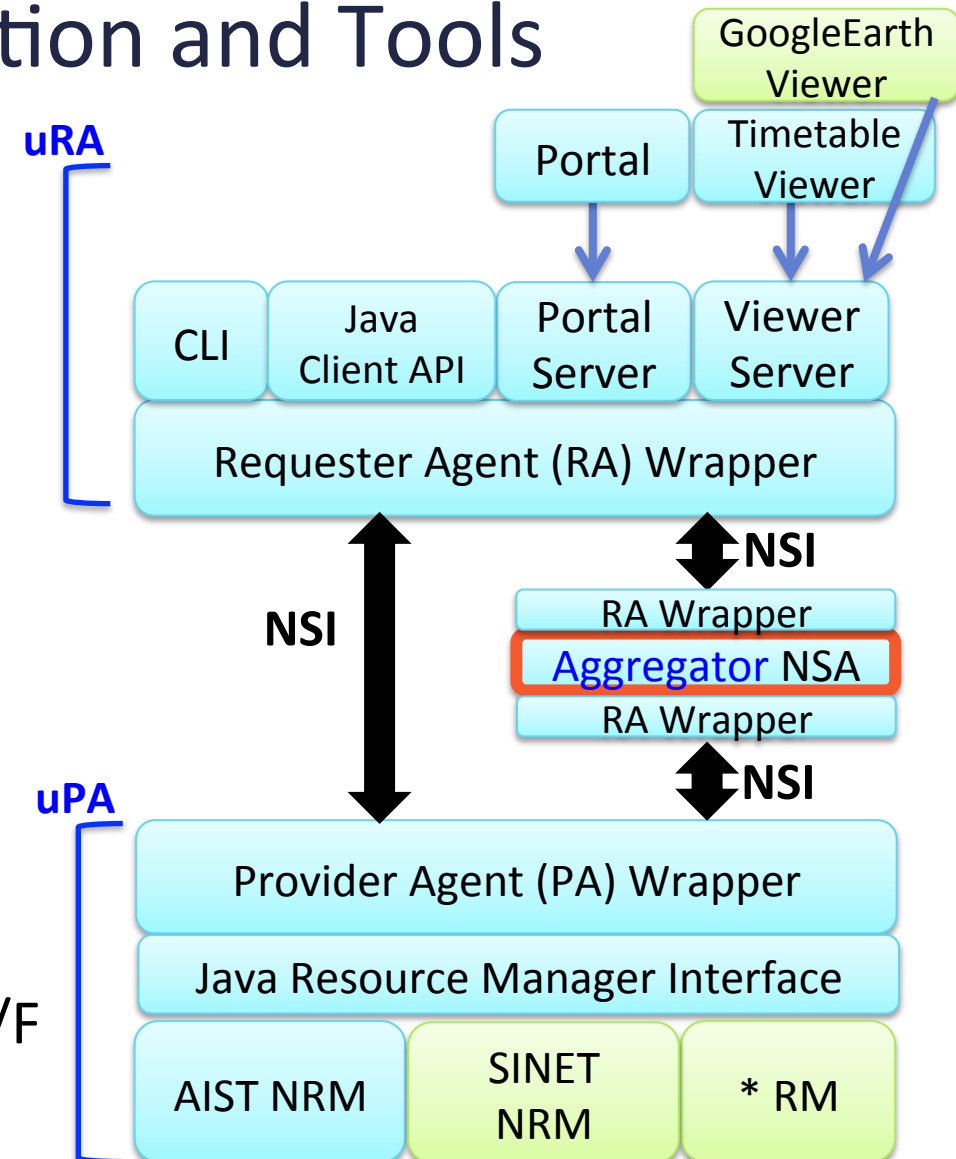


Google Earth Viewer  
developed by KDDI

Timetable Viewer  
developed by AIST

# Overview of the G-lambda/A Implementation and Tools

- RA/PA Wrappers
- uRA modules
  - CLI
  - Java Client API
  - Portal Server and Portal
  - Viewer Server and Timetable Viewer
- Aggregator NSA
- uPA modules
  - Java Resource Manager I/F
  - AIST NRM

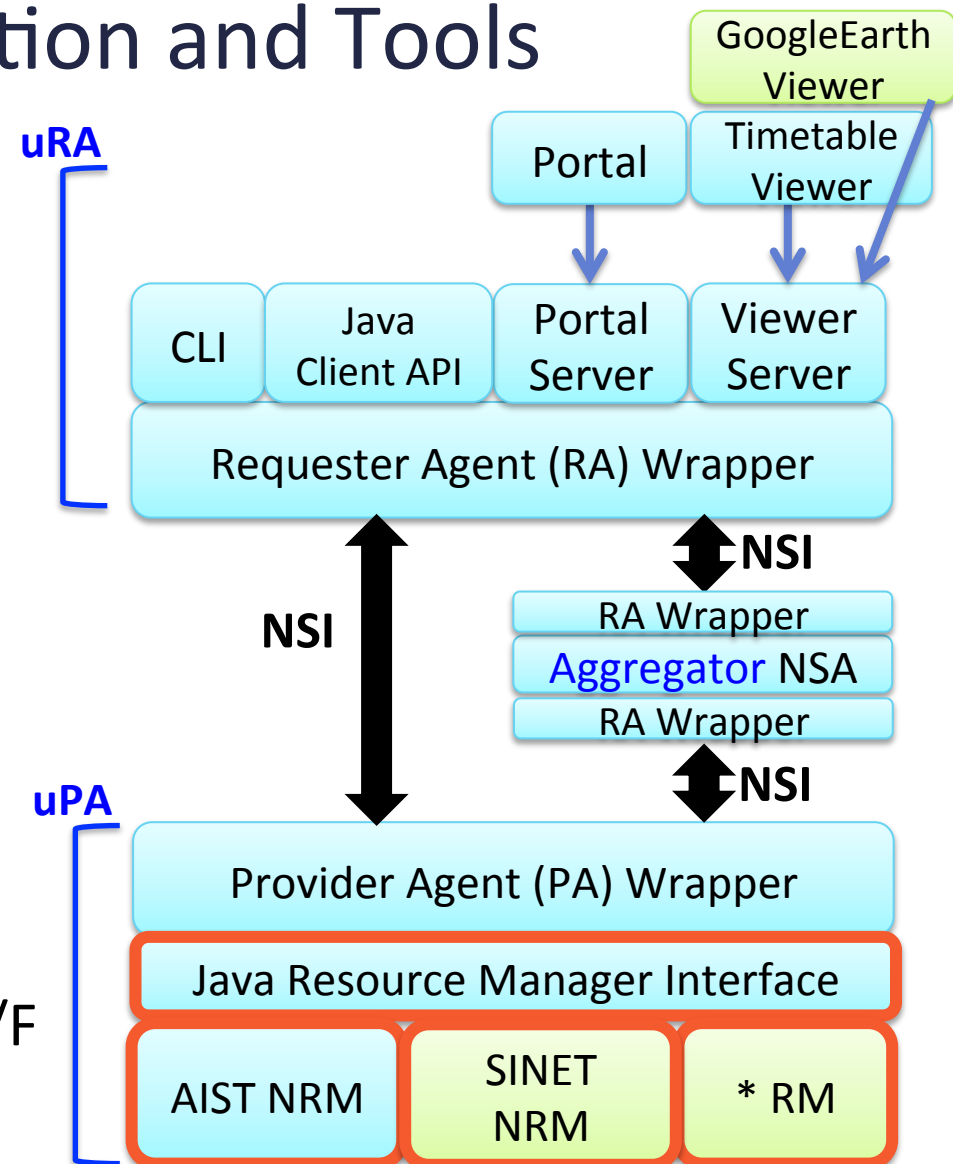


# Aggregator NSA

- Downloads topology files, described by each network provider, and calculates available paths between all of STPs in advance
- For each request, Aggregator sends requests to the corresponding PAs simultaneously, and returns aggregated response to the requester
- Can negotiate multiple PA implementations with various authentication models
  - http/https user and password authentication and OAuth2

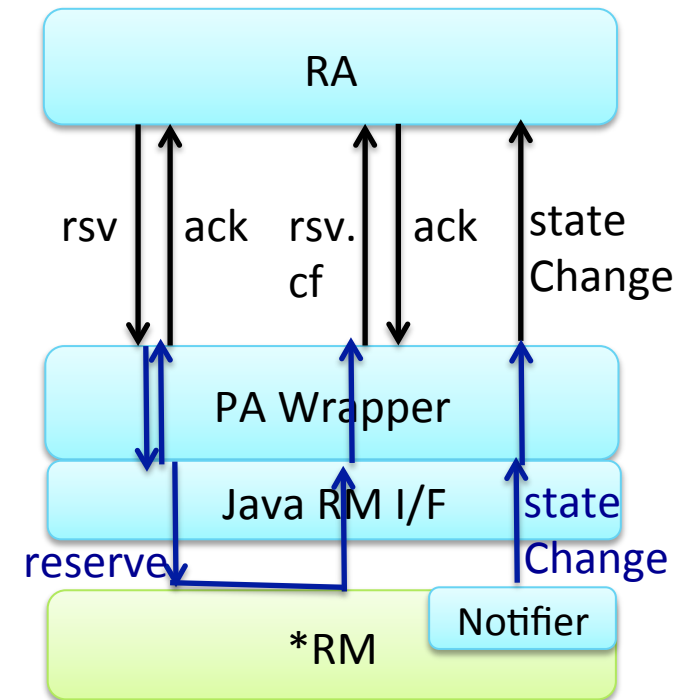
# Overview of the G-lambda/A Implementation and Tools

- RA/PA Wrappers
- uRA modules
  - CLI
  - Java Client API
  - Portal Server and Portal
  - Viewer Server and Timetable Viewer
- Aggregator NSA
- uPA modules
  - Java Resource Manager I/F
  - AIST NRM



# Java Resource Manager Interface

- Provides SPI (Service Provider I/F) to allow to develop a NSI CS v. 2-compliant PA
  - Receives CS operations (reserve, provision, release, terminate) and returns ack and confirmed/failed messages to the requester
  - Replies query-related operations
  - Absorbs the difference between minor wsdl/schema changes
- “Notifier” provides callback functions to send notifications from NRM
  - void dataPlaneStateChange()
  - void errorEvent()
- SINET PA provided by NII and AIST PA are developed using the Java RM I/F
  - SINET PA uses the I/F on top of their own network provisioning system



# Java Resource Manager SPI

```
public interface NSI2ResourceManagerBase {  
    public void setNotifier(Notifier notifier);  
    public void commit(CommonHeaderType header, String connectionId) throws  
ServiceException;  
    public void abort(CommonHeaderType header, String connectionId) throws ServiceException;  
    public void provision(CommonHeaderType header, String connectionId) throws  
ServiceException;  
    public void release(CommonHeaderType header, String connectionId) throws  
ServiceException;  
    public void terminate(CommonHeaderType header, String connectionId) throws  
ServiceException;  
}
```

```
public interface EthernetResourceManager {  
    public void reserve(CommonHeaderType header, String connectionId, String  
globalReservationId, String description, EthernetCriteria criteria) throws ServiceException;  
    public void modify(CommonHeaderType header, String connectionId, String  
globalReservationId, String description, EthernetCriteria criteria) throws ServiceException;  
}
```



# Summary

- AIST has developed G-lambda/A NSA implementations and tools for NSI CS v. 2.0
  - uRA modules: CLI, Java Client API, Portal Server and Portal, Timetable Viewer and Viewer Server
  - Aggregator NSA
  - uPA modules: Java Resource Manager I/F, AIST NRM
- NSI Application, PA and viewer by using our package have been developed by third parties
  - SINET PA, UltraGrid application, Google Earth Viewer
- The package is an open-source software and available at <http://www.g-lambda.net/gridars>
- Portal REST I/F and viewer(monitoring) architecture and its I/F need to be standardized