



# MonALISA

*MONitoring Agents using a Large  
Integrated Services Architecture*

**An Agent Based, Dynamic Service System to Monitor,  
Control and Optimize Distributed Systems**

**ICFA WORKSHOP**

**Daegu, May 2005**



**Iosif Legrand**

**California Institute of Technology**



# MonALISA is A Dynamic, Distributed Service Architecture



- Real-time monitoring is an essential part of managing distributed systems. The monitoring information gathered is necessary for developing higher level services, and components that provide automated decisions, to help operate and optimize the workflow in complex systems.
- The MonALISA system is designed as an ensemble of autonomous multi-threaded, self-describing agent-based subsystems which are registered as dynamic services, and are able to collaborate and cooperate in performing a wide range of monitoring tasks. These agents can analyze and process the information, in a distributed way, to provide optimization decisions in large scale distributed applications.
- **An agent-based architecture provides the ability to invest the system with increasing degrees of intelligence; to reduce complexity and make global systems manageable in real time**



# The MonALISA Architecture Provides:



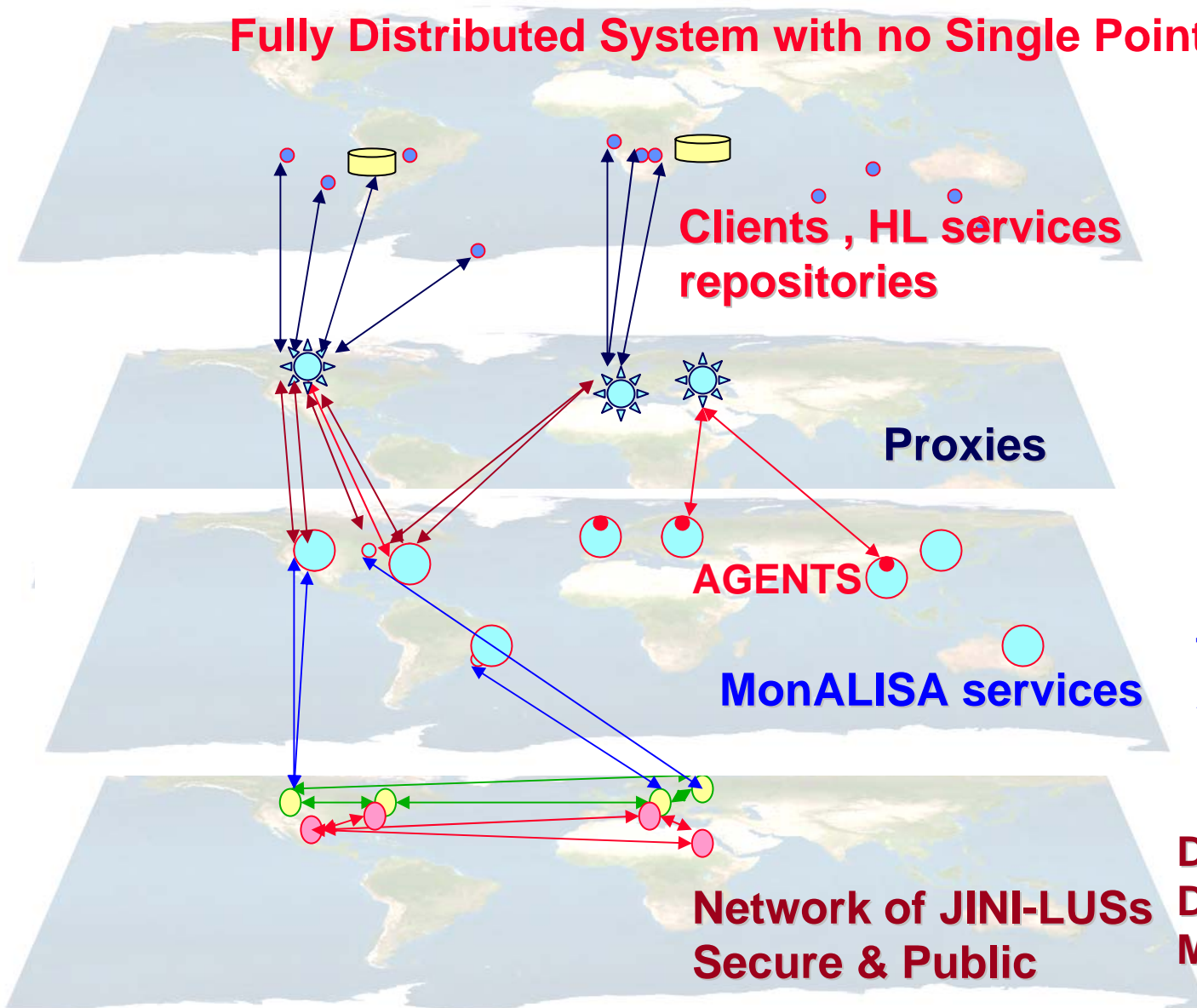
- Distributed **Registration and Discovery** for Services and Applications.
- Monitoring all aspects of complex systems :
  - ❑ System information for computer nodes and clusters
  - ❑ Network information : WAN and LAN
  - ❑ Monitoring the performance of Applications, Jobs or services
  - ❑ The End User Systems, its performance
- Can **interact with any other services** to provide in near real-time customized information based on monitoring data
- Secure, remote **administration** for services and applications
- **Agents to supervise applications**, to restart or reconfigure them, and to notify other services when certain conditions are detected.
- The MonALISA framework can be used **to develop higher level decision services**, implemented as a distributed network of communicating agents, to perform global optimization tasks.
- **Graphical User Interfaces** to visualize complex information



# The MonALISA Discovery System & Services



**Fully Distributed System with no Single Point of Failure**



**Global Services or Clients**

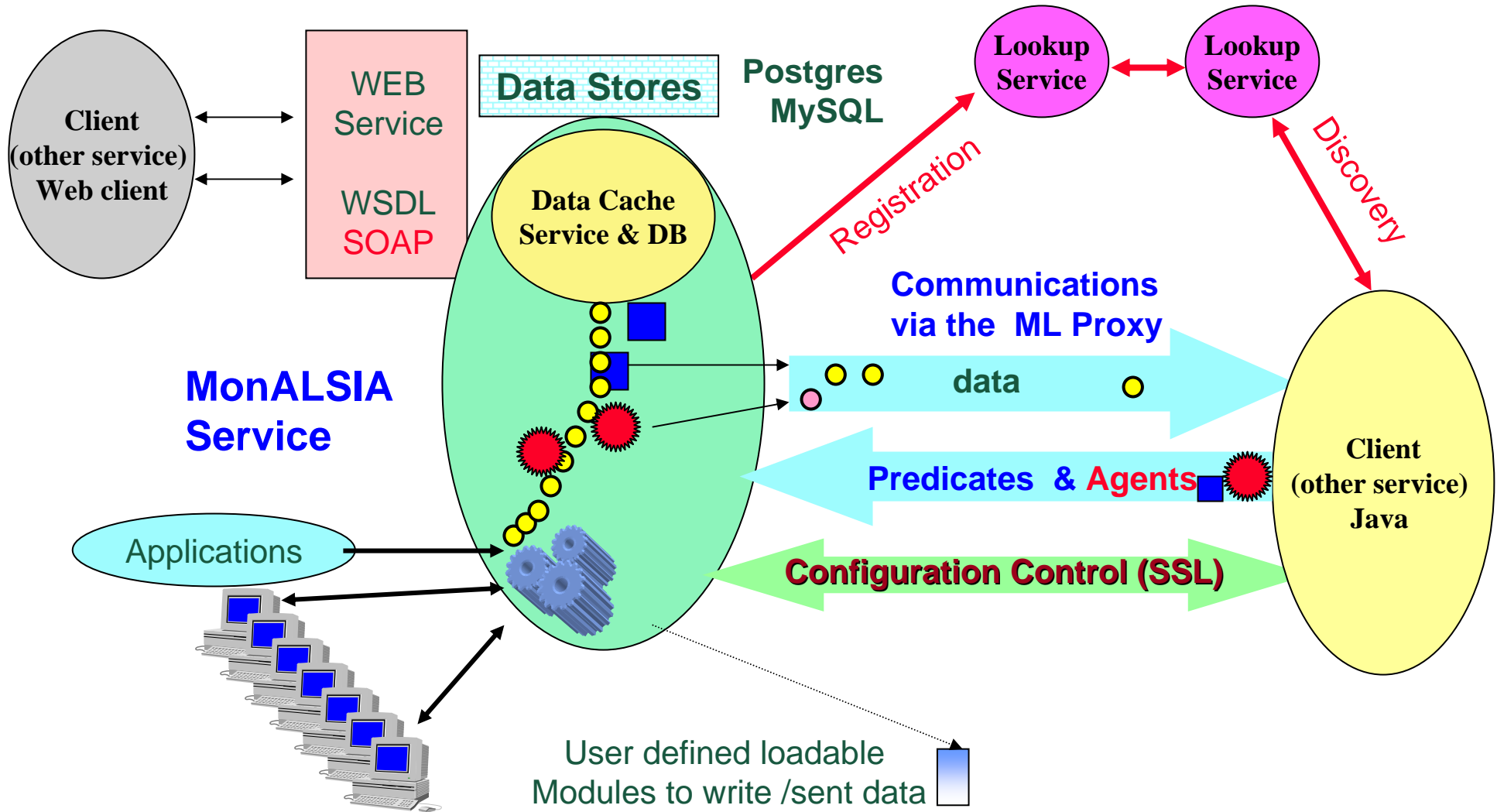
**Dynamic load balancing  
Scalability & Replication  
Security AAA for Clients**

**Distributed System  
for gathering and  
Analyzing Information.**

**Distributed Dynamic  
Discovery- based on a lease  
Mechanism and REN**



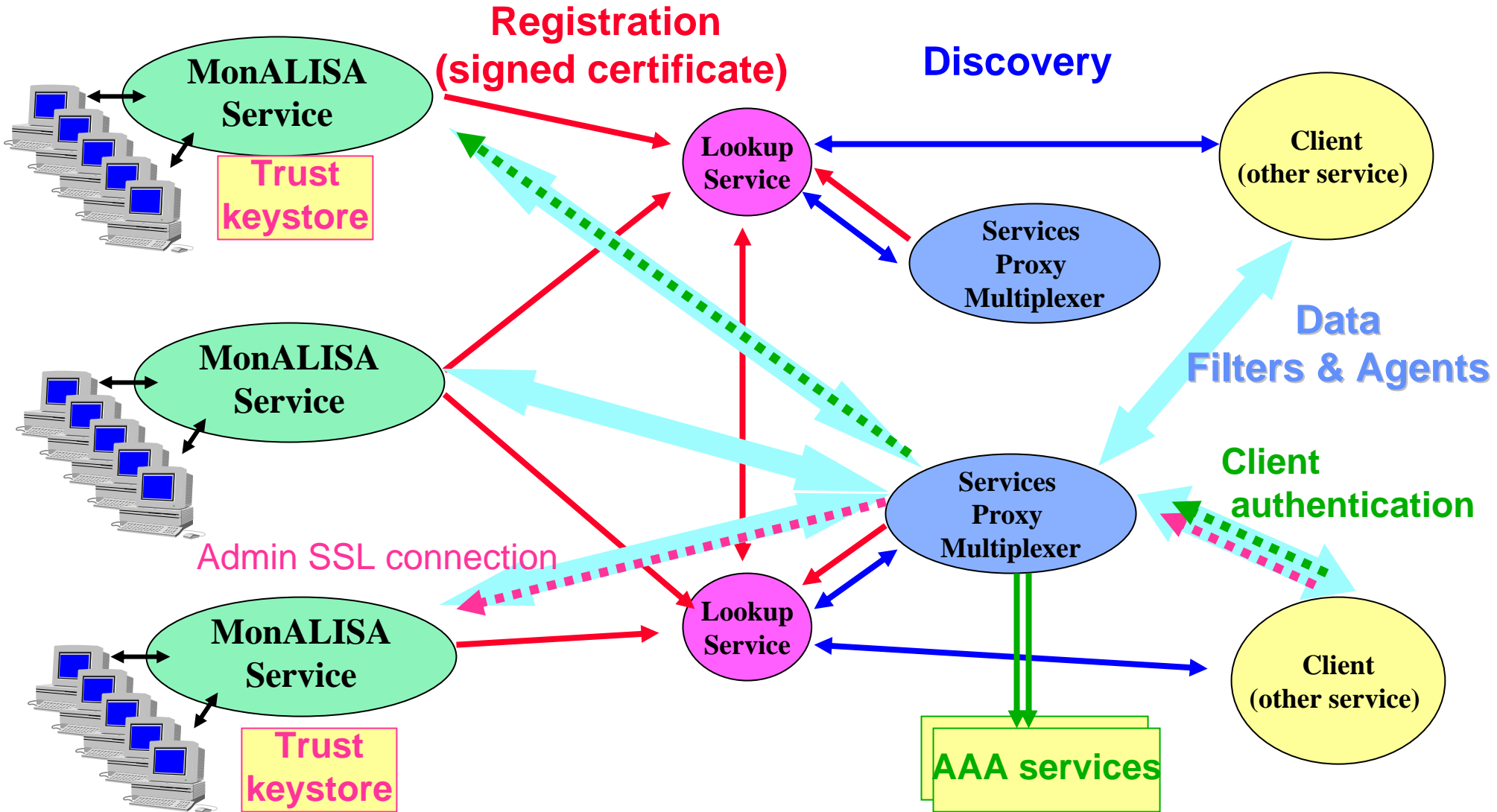
# MonALISA service & Data Handling





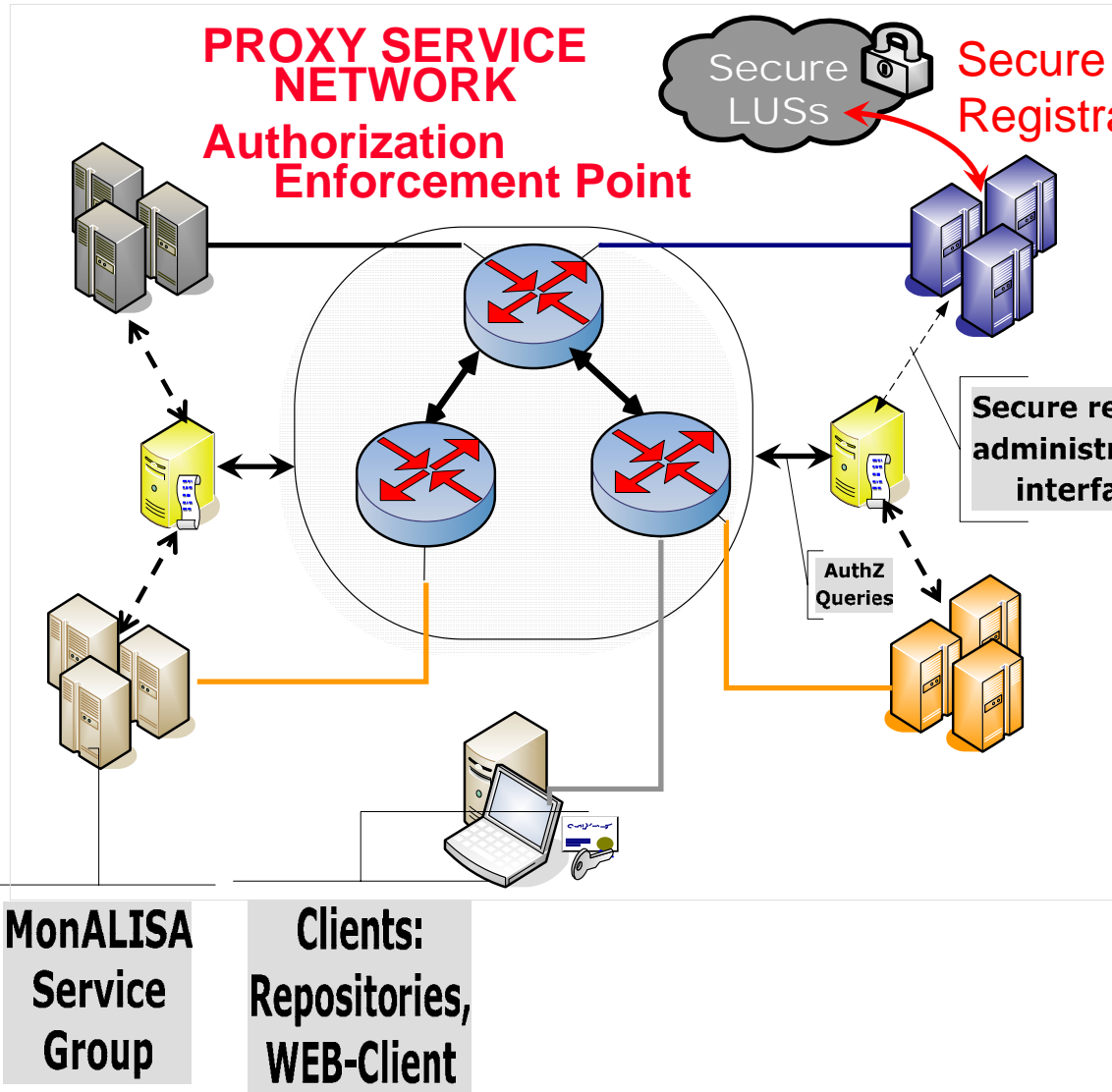
# Registration / Discovery

## Admin Access and AAA for Clients





# Security in the MonALISA System



## SSL/TLS, PKIX, GSS-API

- 1) **Community-based trust relationships.** Multiple MonaLisa services may be operated by a community. The community memberships is maintained in specialized Authorization Services
- 2) **Flexible communication protection**
- 3) **Secure registration in LUSs** based on an X.509 host or site certificate
- 4) **Auditing**



# Communities using MonALISA

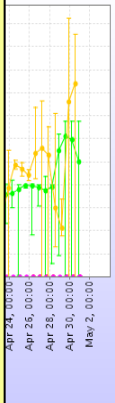


- ❖ Grid3  
~40 sites in US and 1 Korea
- ❖ CMS-US sites
- ❖ CMS
- ❖ CDF
- ❖ D0 SAR
- ❖ ABILENE backbone
- ❖ GLORIAD
- ❖ STAR
- ❖ ALICE
- ❖ VRVS System
- ❖ RoEduNET backbone
- ❖ INTERNET2 PIPES
- ❖ OSG
- ❖ LHCb



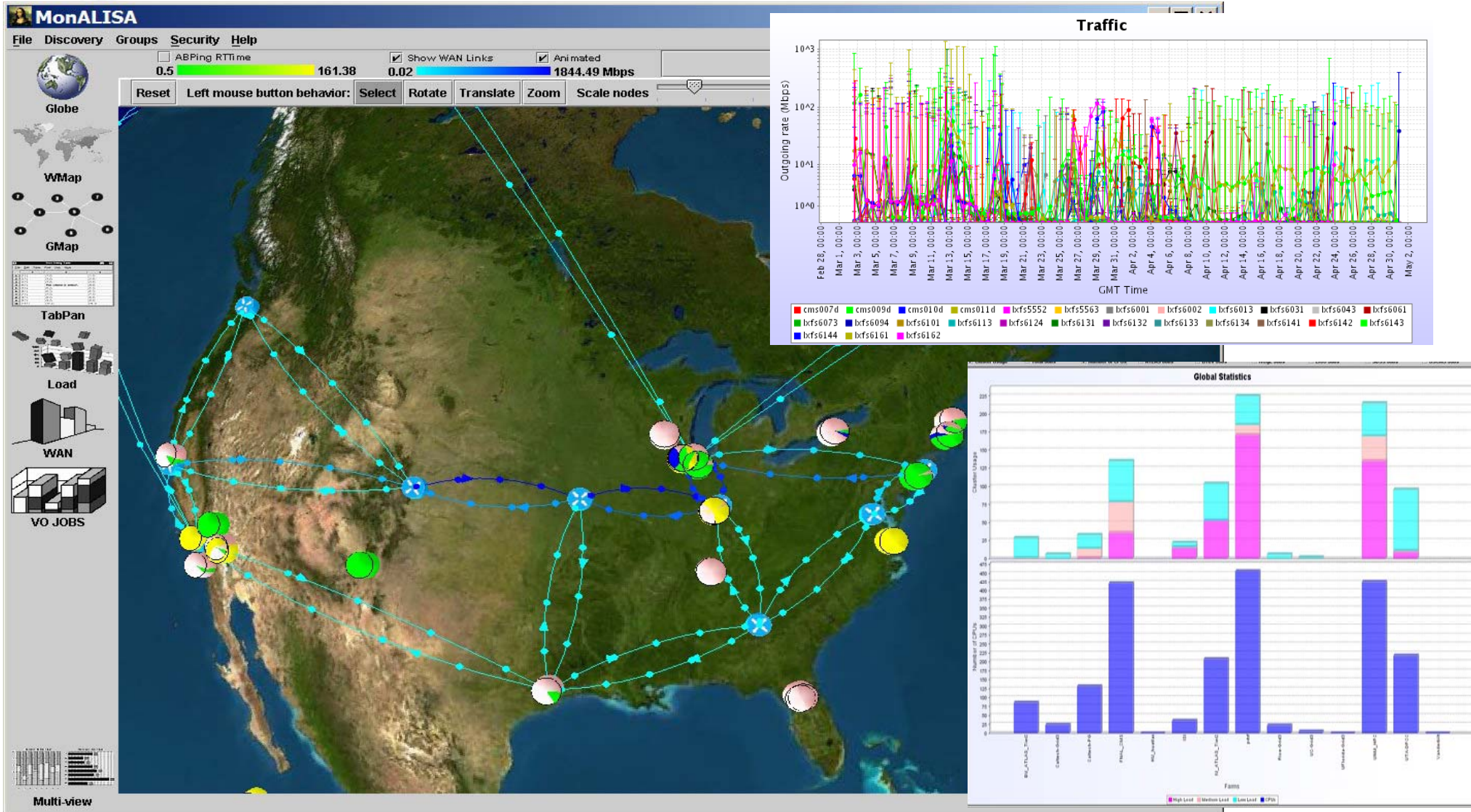
It has been used for Demonstrations at:

- SC2003
- Telecom 2003
- WSIS 2003
- SC 2004
- I2 2005



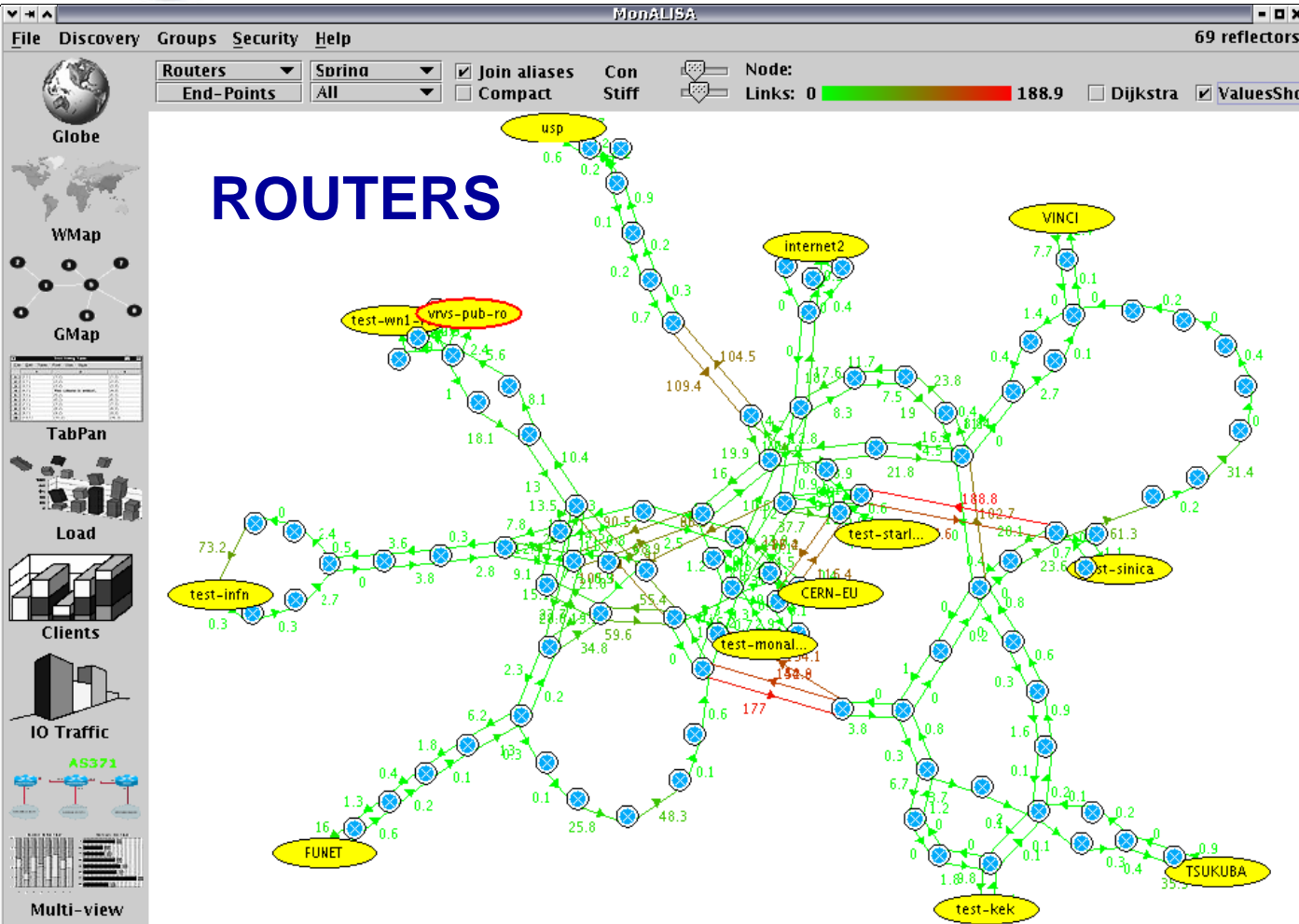


# Monitoring I2 Network Traffic, Grid03 Farms and Jobs

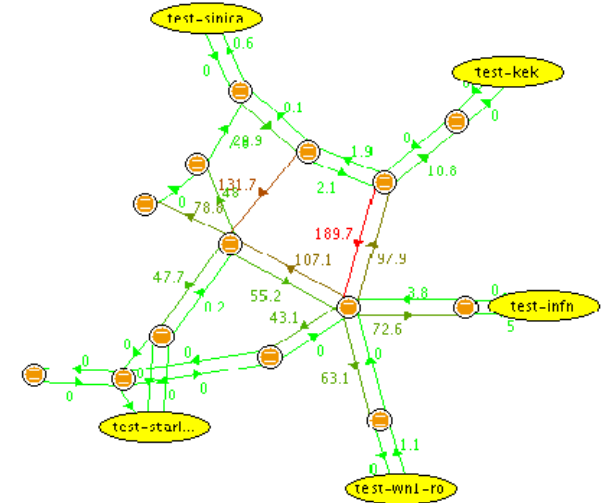




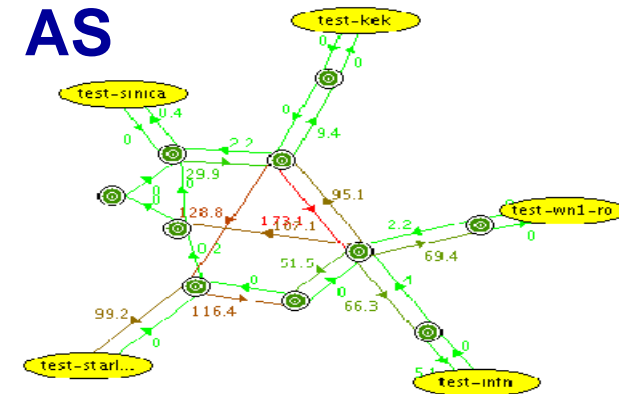
# Monitoring Network Topology Latency, Routers



## NETWORKS

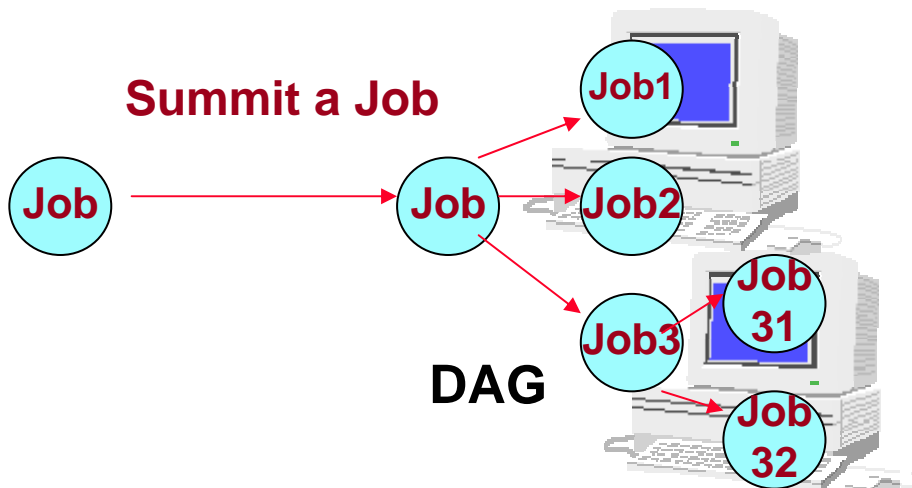
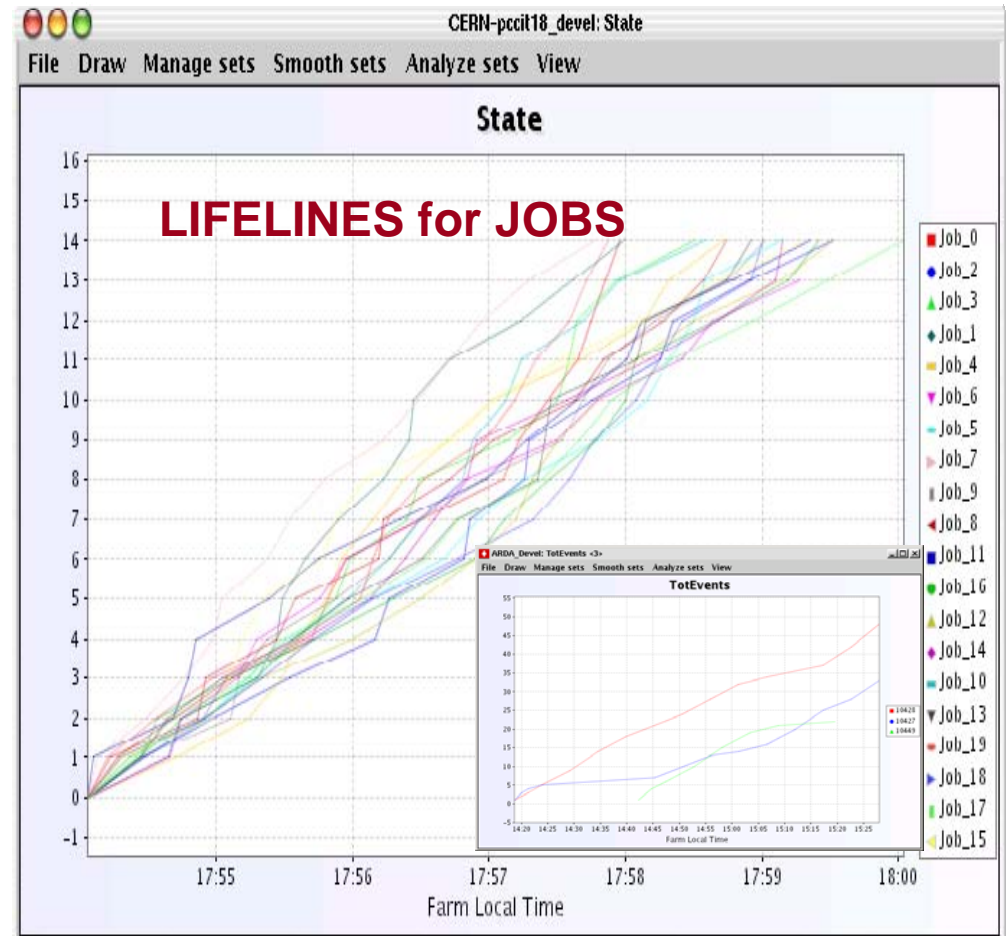
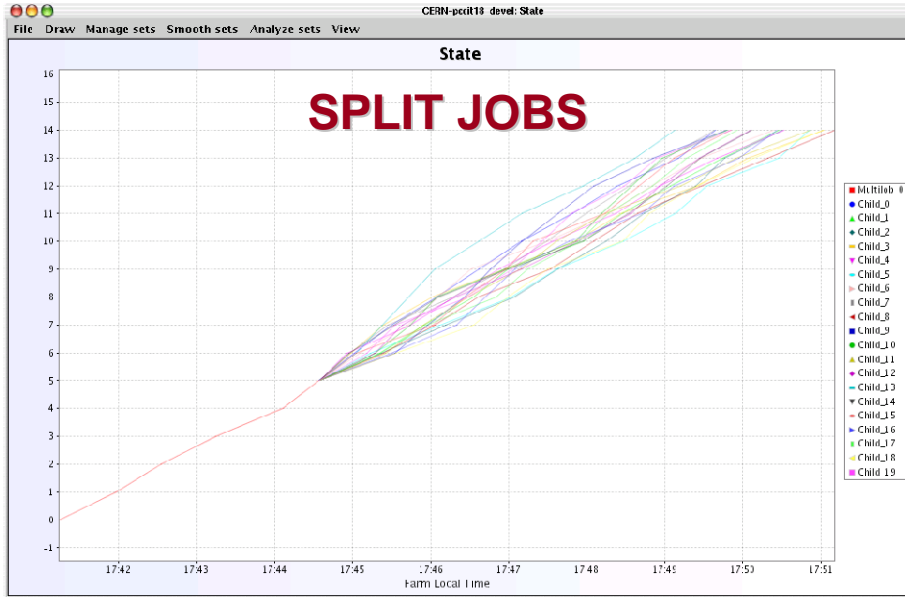


## AS





# Monitoring the Execution of Jobs and the Time Evolution

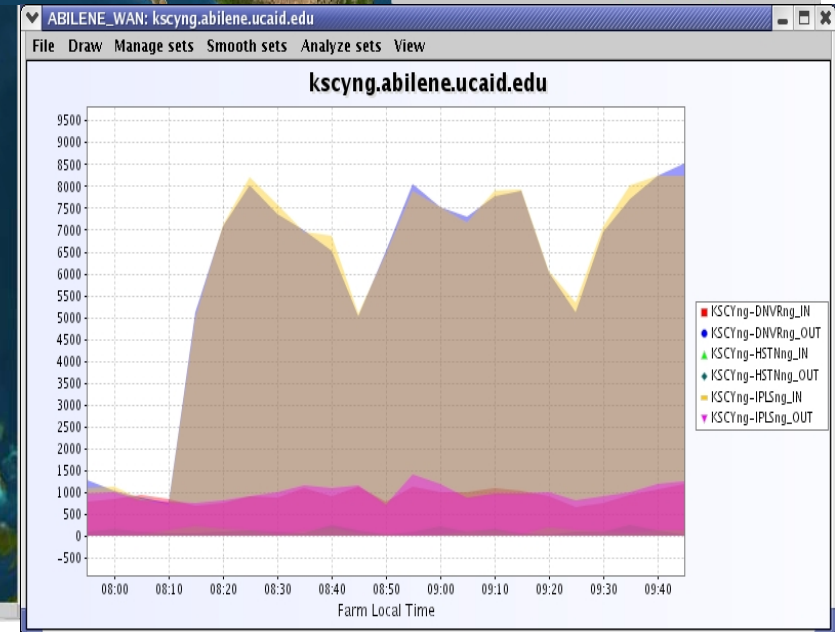
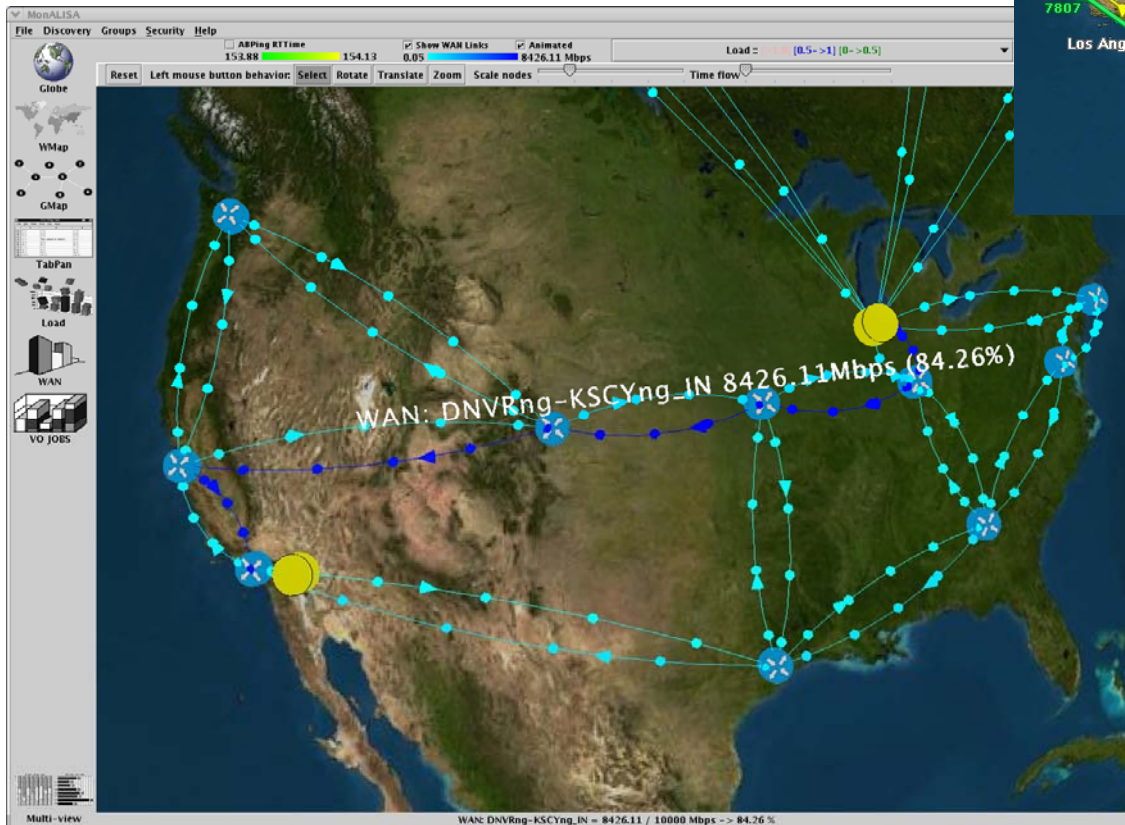
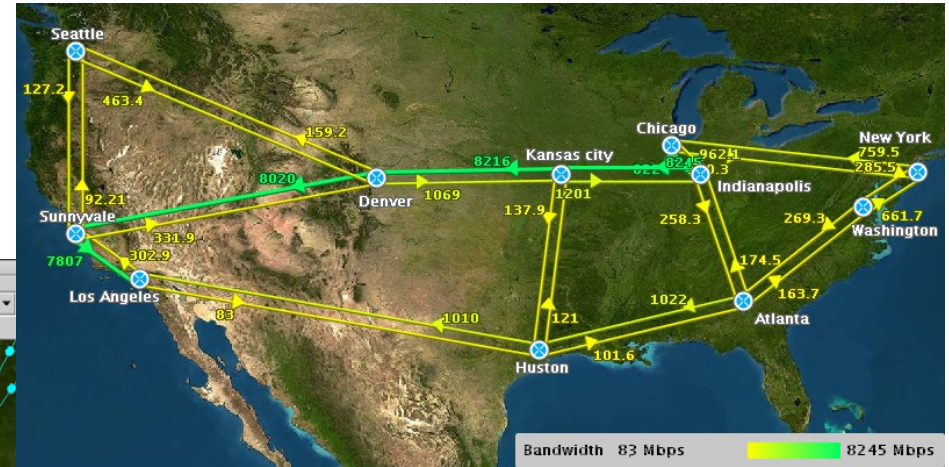




# Monitoring ABILENE backbone Network

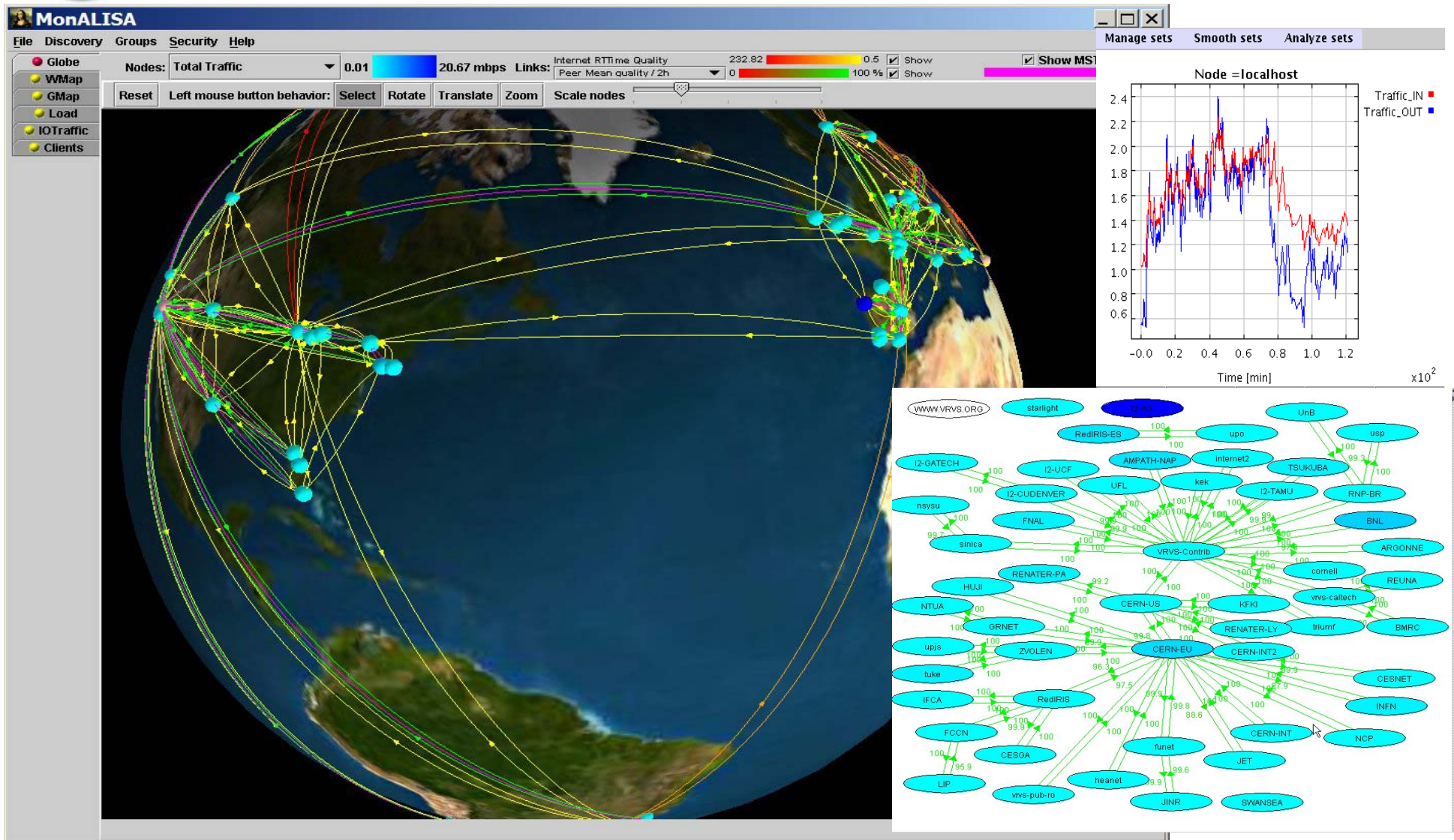


- ◆ Test for a Land Speed Record
- ◆ ~ 7 Gb/s in a single TCP stream from Geneva to Caltech





# Monitoring VRVS Reflectors and Communication Topology





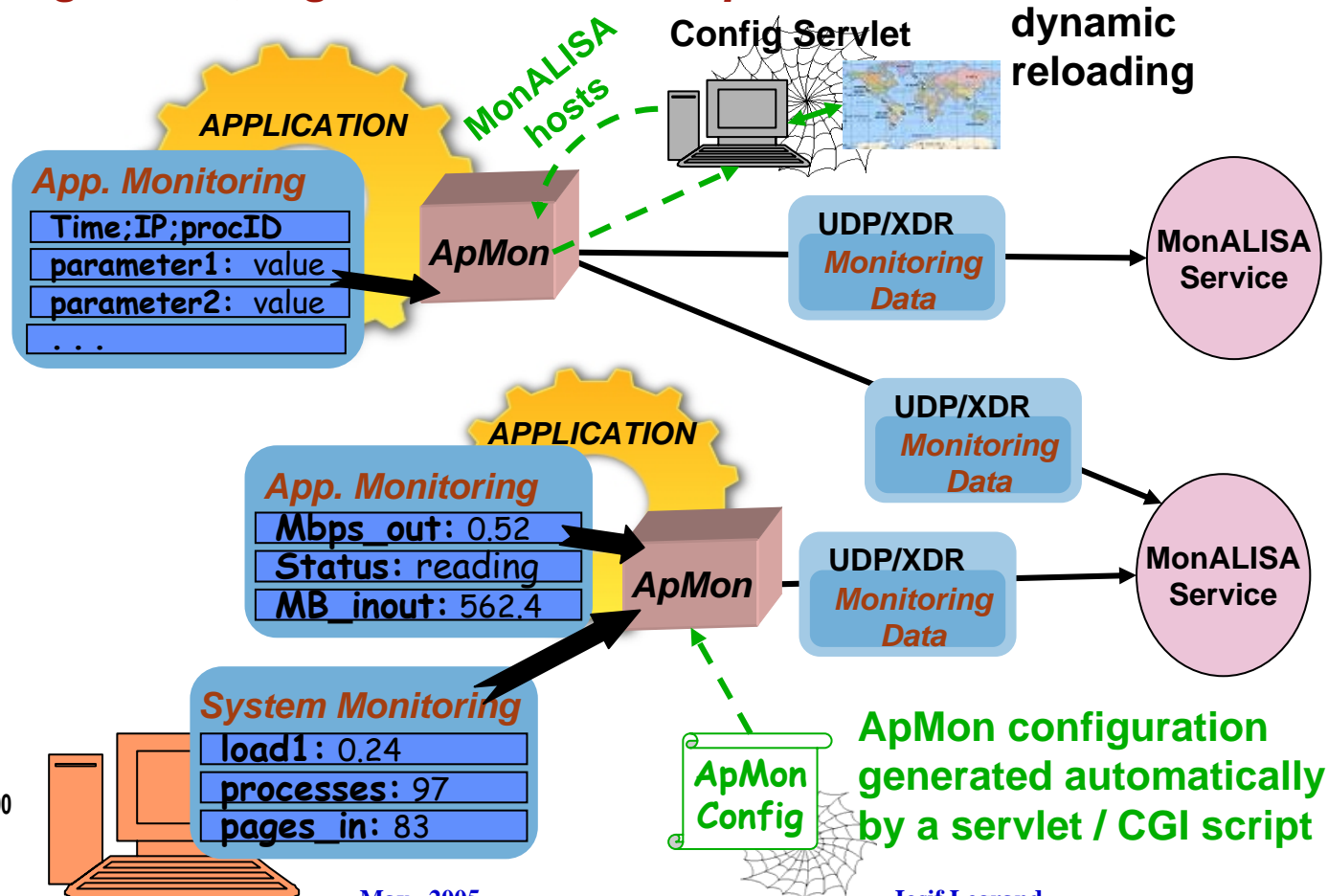
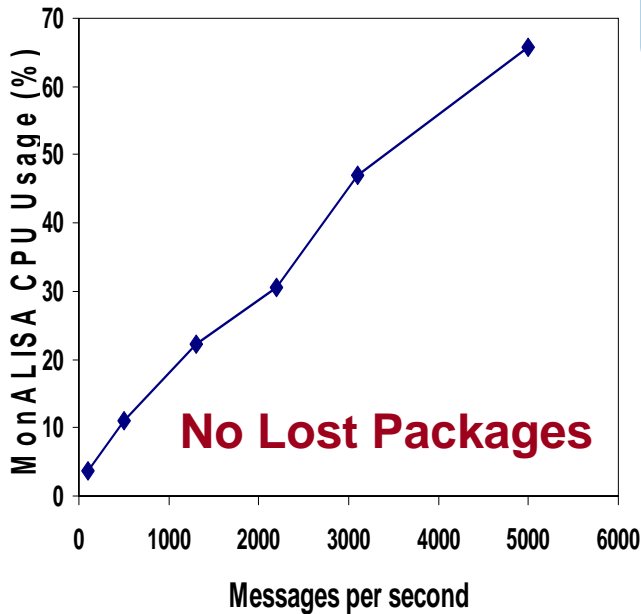
# ApMon – Application Monitoring



Library of APIs (C, C++, Java, Perl, Python) that can be used to send any information to MonALISA services

➤ Flexibility, dynamic configuration, high communication performance

- Automated system monitoring
- Accounting information





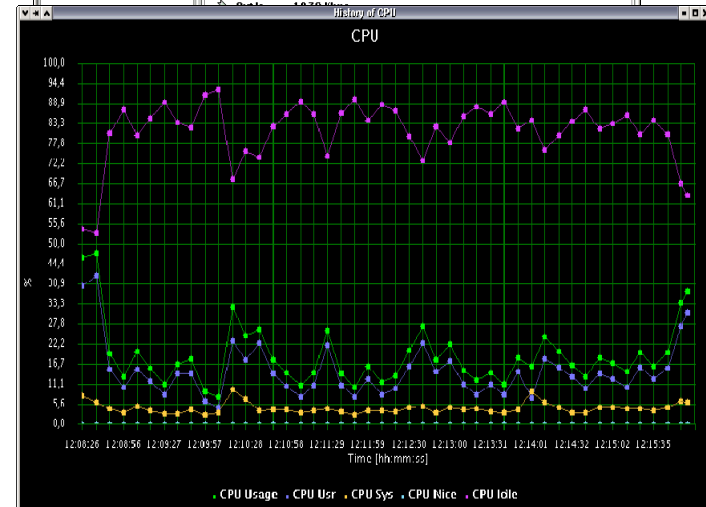
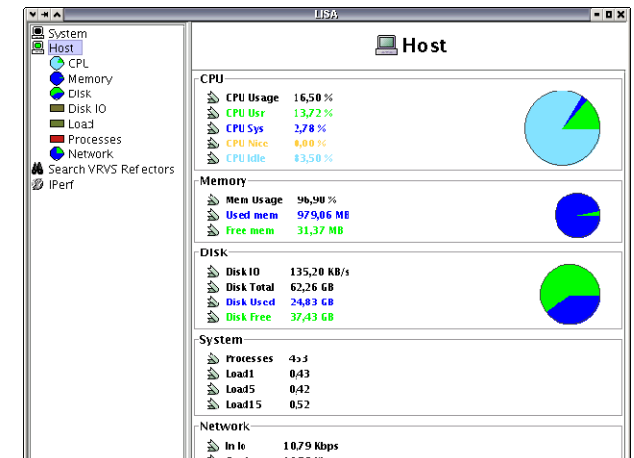
# LISA- Localhost Information Service Agent

## End To End Monitoring Tool



A lightweight Java Web Start application that provides complete monitoring of the end user systems, the network connectivity and can use the MonALISA framework to optimize client applications

- ◆ It is very easy to deploy and install by simply using any browser.
- ◆ It detects the system architecture, the operating system and selects dynamically the binary parts necessary on each system.
- ◆ It can be easily deployed on any system. It is now used on all versions of Windows, Linux, Mac.
- ◆ It provides complete system monitoring of the host computer:
  - ◆ CPU, memory, IO, disk, ...
  - ◆ Hardware detection
  - ◆ Main components, Audio, Video equipment,
  - ◆ Drivers installed in the system
  - ◆ Provides embedded clients for IPERF (or other network monitoring tools, like Web 100 )
  - ◆ A user friendly GUI to present all the monitoring information.

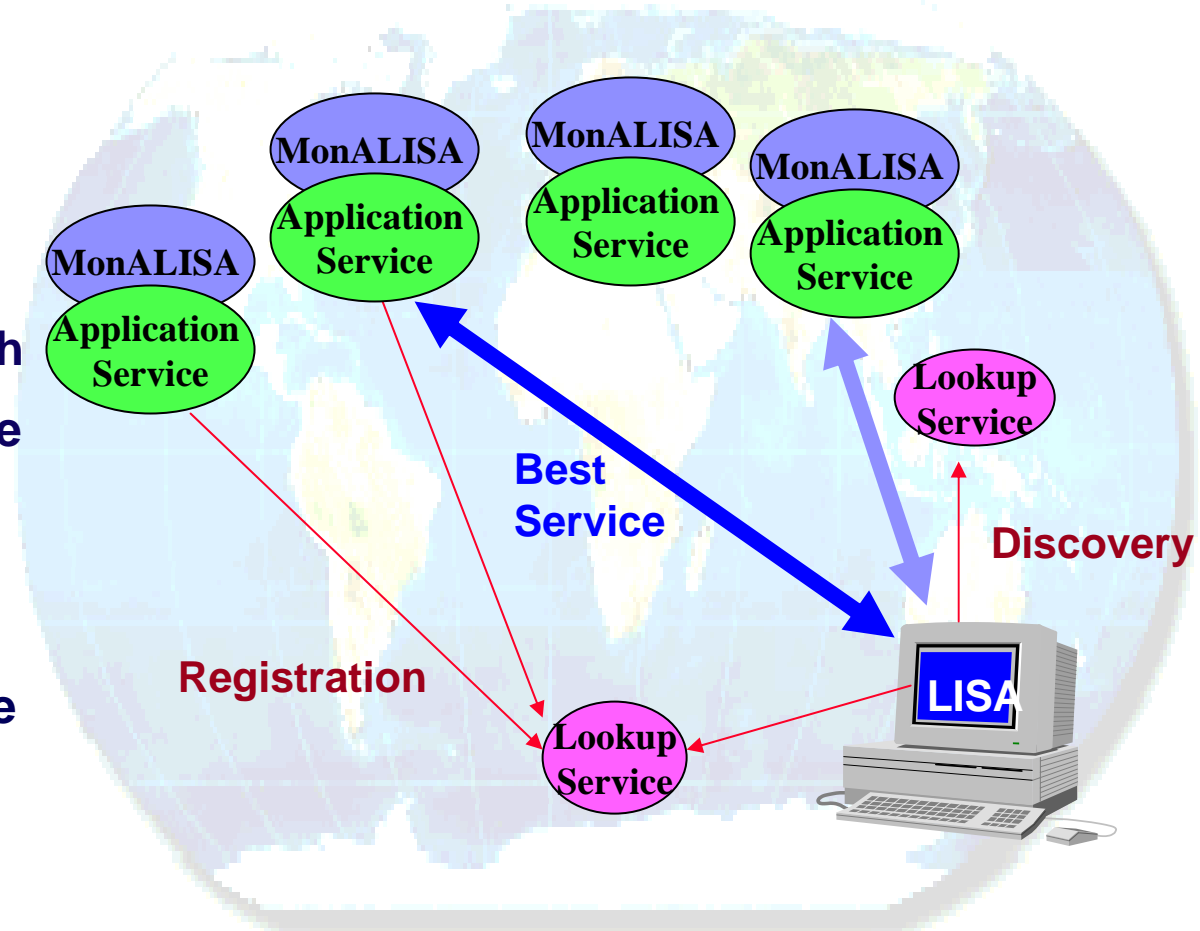




# LISA- Provides an Efficient Integration for Distributed Systems and Applications

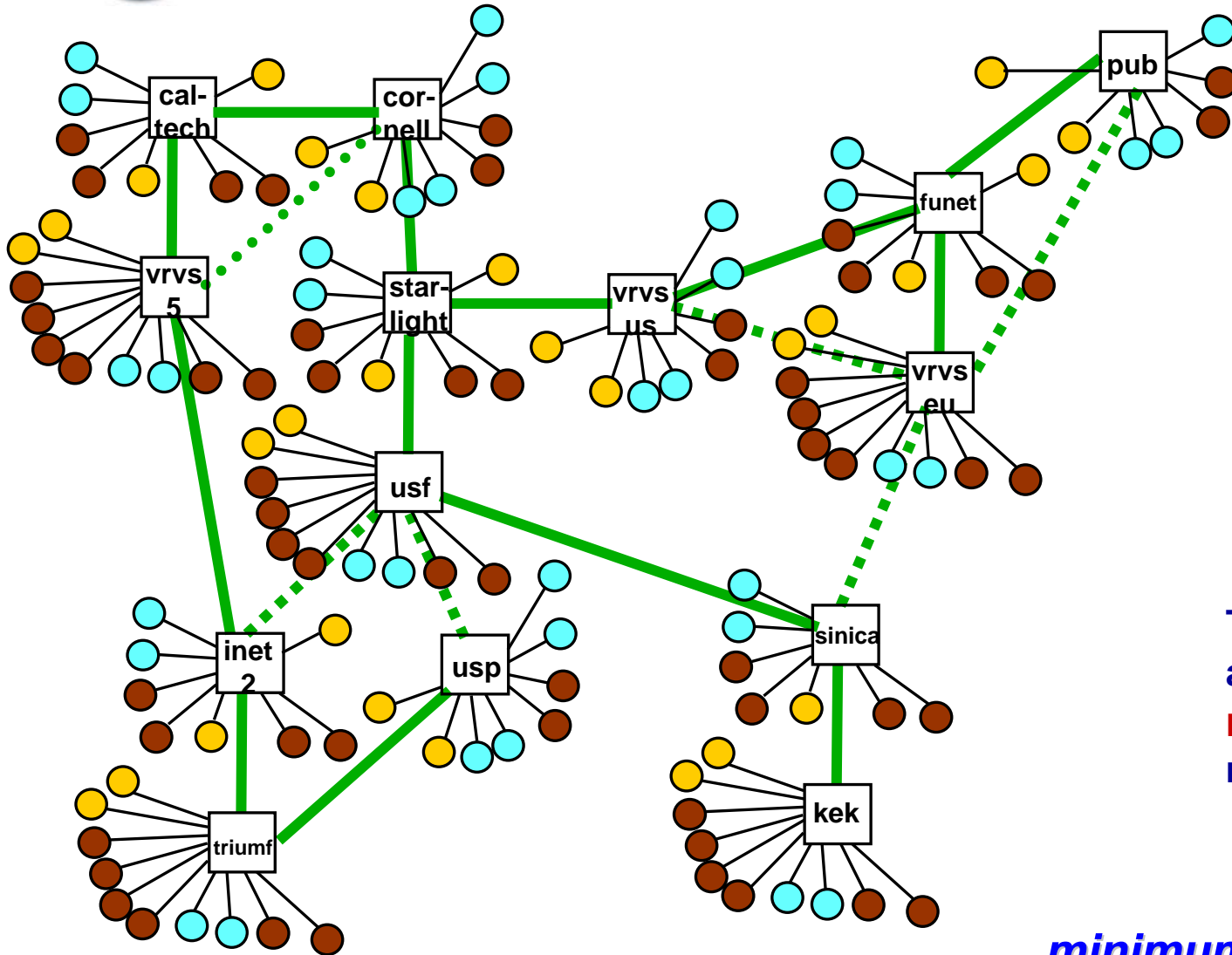


- ◆ It is using external services to identify the real IP of the end system, its network ID and AS
- ◆ Discovers MonALISA services and can select, based on service attributes, different applications and their parameters (location, AS, functionality, load ... )
  - Based on information such as AS number or location, it determines a list with the best possible services.
  - Registers as a listener for other service attributes (eg. number of connected clients).
  - Continuously monitors the network connection with several selected services and provides the best one to be used from the client's perspective.
  - Measures network quality, detects faults and informs upper layer services to take appropriate decisions





# Communication in the Distributed Collaborative System



**Reflectors** are hosts that interconnect users by permanent IP tunnels.

The active IP tunnels must be selected so that there is no cycle formed.



**Tree**

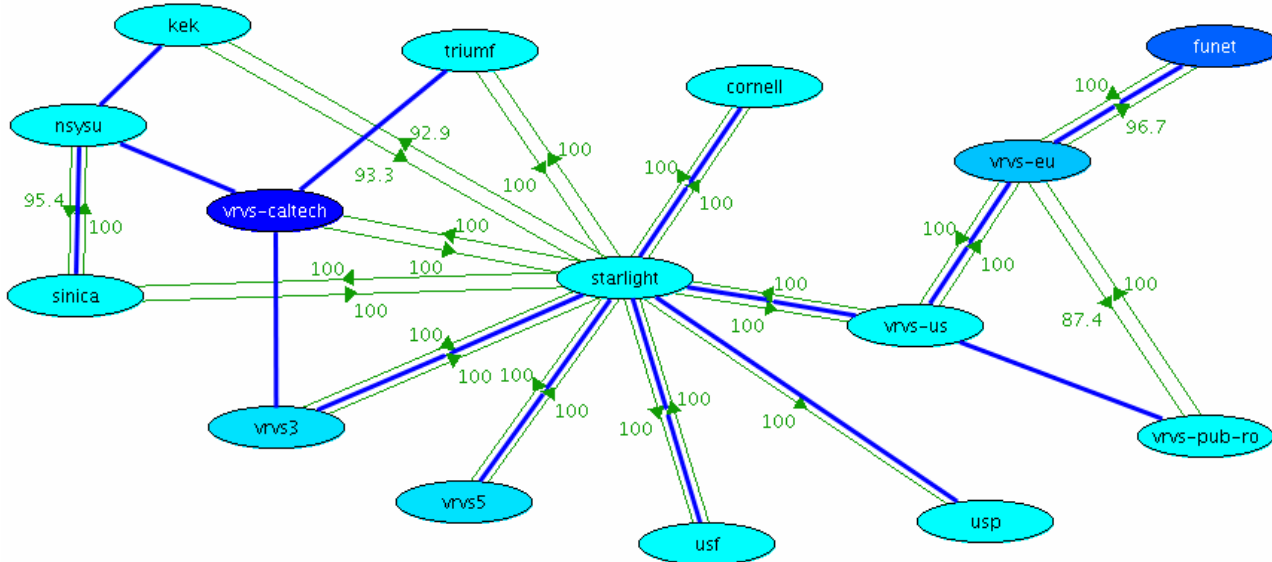
The selection is made according to the **real-time measurements** of the network performance.

$$w(T) = \sum_{(v,u) \in T} w((v,u))$$

**minimum-spanning tree (MST)**

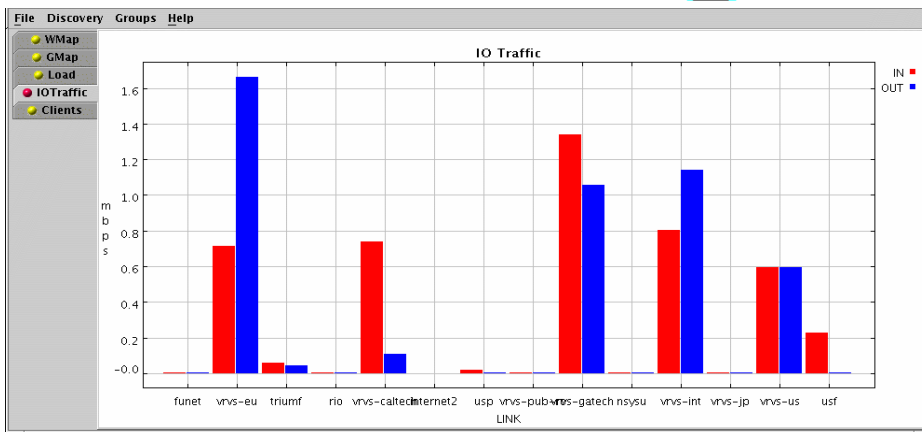


# Creating a Dynamic, Global, Minimum Spanning Tree to optimize the connectivity



A weighted connected graph  $G = (V, E)$  with  $n$  vertices and  $m$  edges. The quality of connectivity between any two reflectors is measured every 2s. Building in near real time a minimum-spanning tree  $T$

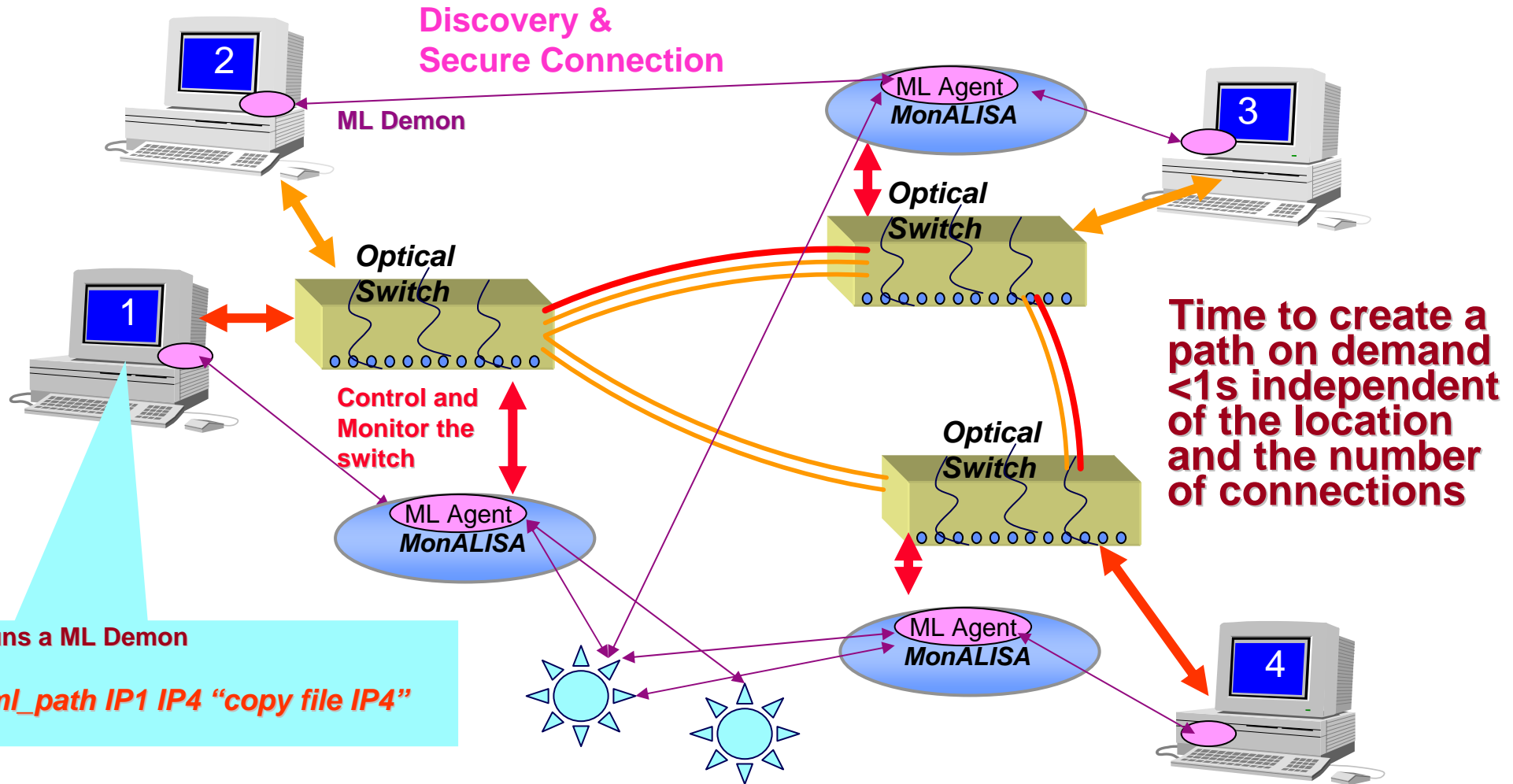
$$w(T) = \sum_{(v,u) \in T} w((v,u))$$







# MonALISA agents to create on demand on an optical path or tree



Discovery & Secure Connection

ML Demon

Optical Switch

Control and Monitor the switch

ML Agent MonALISA

ML Agent MonALISA

Optical Switch

Optical Switch

ML Agent MonALISA

Time to create a path on demand <1s independent of the location and the number of connections

Runs a ML Demon

>ml\_path IP1 IP4 "copy file IP4"

ML proxy services used in Agent Communication



# Monitoring Optical Switches Agents to Create on Demand an Optical Path



The screenshot displays the MonALISA monitoring interface. On the left is a 3D map of the world with several nodes connected by colored lines. The top center features a bar chart titled "Sys300Power" showing power levels for various ports (Port\_29 to Port\_40). The top right shows a network topology diagram with nodes labeled "glimmer" and "calient" connected by lines. The bottom right shows a detailed view of a "calient3" node with a grid of 15 ports.

**Sys300Power Data:**

Port	Power_IN
Port_29	-55
Port_28	-50
Port_30	-55
Port_27	-55
Port_26	-55
Port_25	-55
Port_47	-15.355
Port_46	-50
Port_41	-55
Port_40	-55

**Network Topology:**

- Nodes: glimmer1, glimmer2, glimmer3, glimmer4, calient1, calient2, calient3, calient4, calient5.
- IP Addresses: 137.138.42.36, 137.138.42.58, 137.138.42.62, 192.91.244.25, 137.164.30.231, 137.164.30.232, 137.164.30.234.

**calient3 Node Details:**

10.15a.3	10.9a.8	2	3
4	5	6	7
8	9	10	11
12	13	14	15



# Test Setup for Controlling Optical Switches

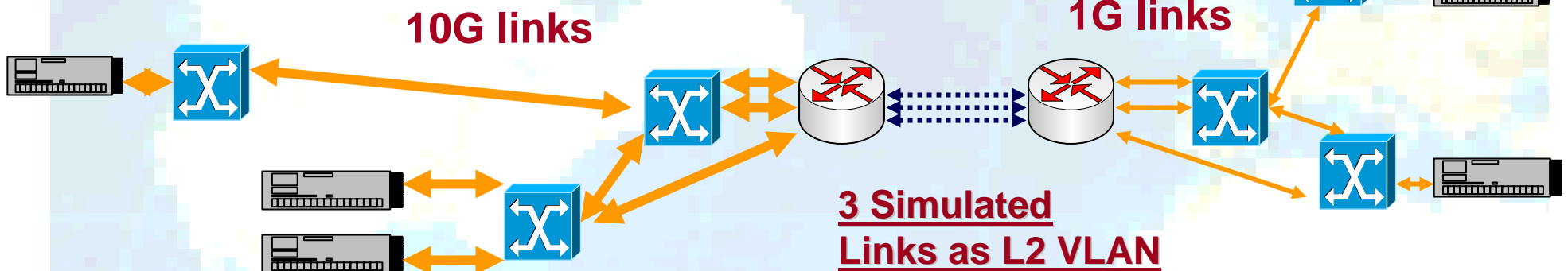


CALIENT (LA)



Glimmerglass (GE)

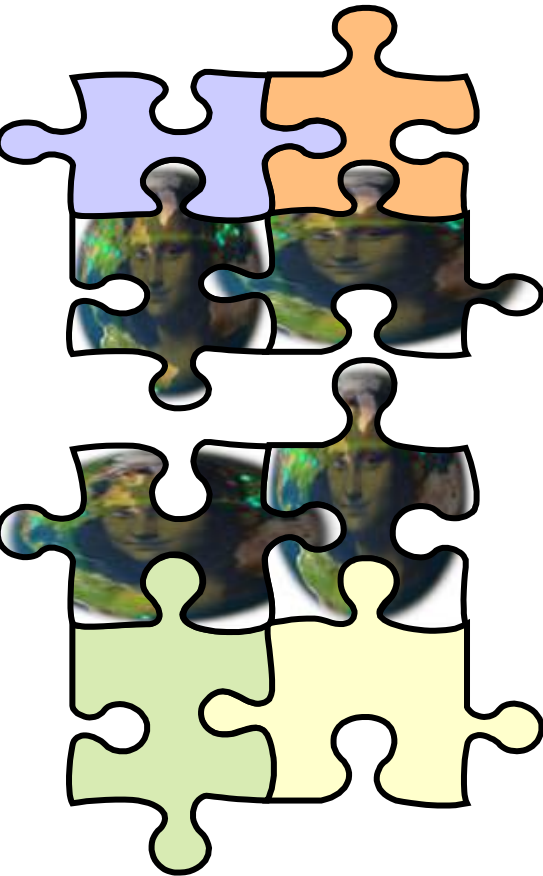
3 partitions on each switch  
They are controlled by a MonALISA service



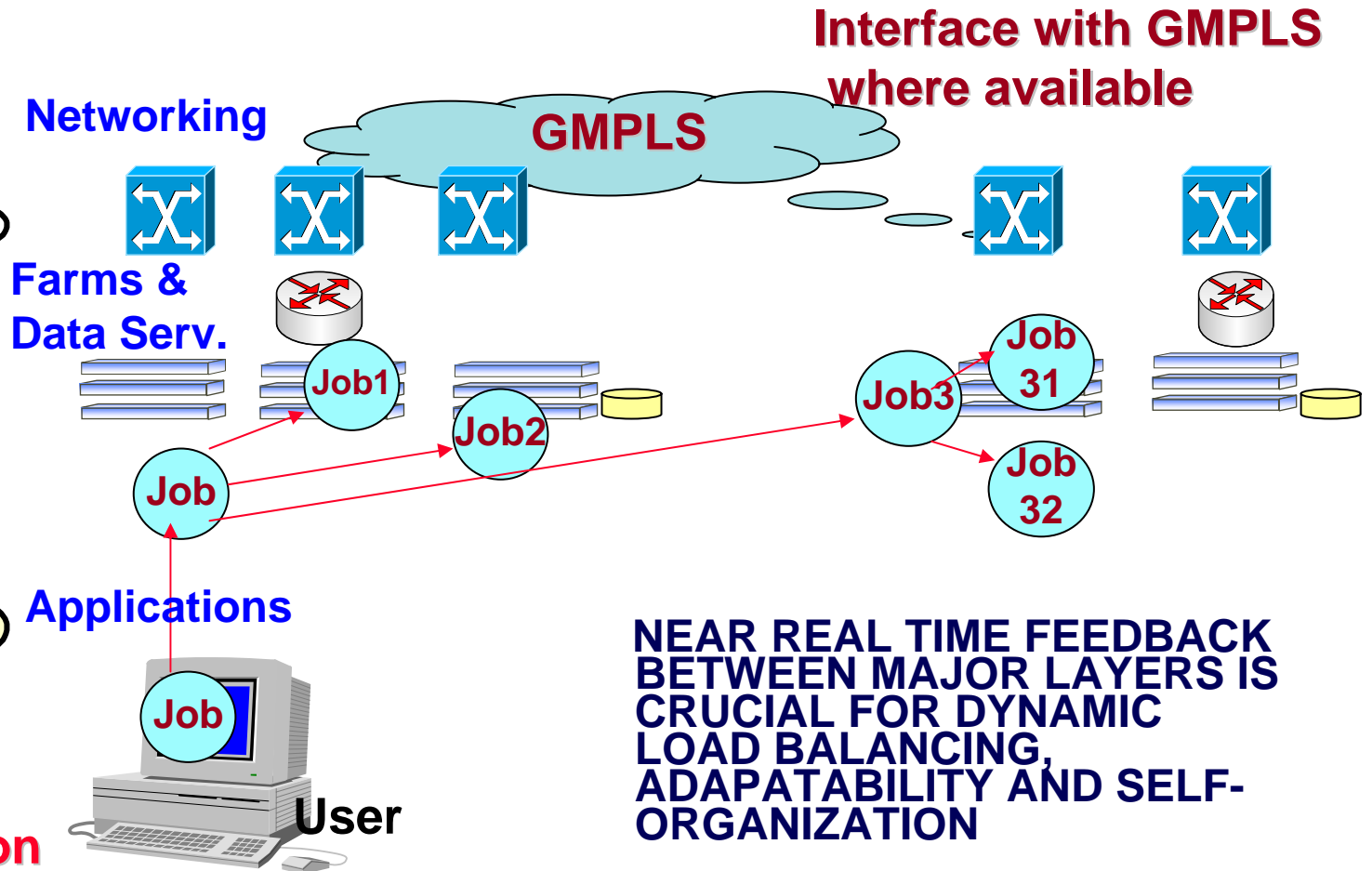
- ◆ Monitor and control switches using TL1
- ◆ Interoperability between the two systems
- ◆ End User access to service



# MonALISA is a framework to correlate information from different layers



**HELP to create Vertical Integration**



**NEAR REAL TIME FEEDBACK BETWEEN MAJOR LAYERS IS CRUCIAL FOR DYNAMIC LOAD BALANCING, ADAPATABILITY AND SELF-ORGANIZATION**



# SUMMARY



**MonaLISA is a fully distributed service system with no single point of failure. It provides reliable registration and discovery.**

- ◆ **MonALISA is interfaced with many monitoring tools and is capable to collect any information from different applications**
- ◆ **It allows to analyze and process information in real time, locally, using Filters or Agents that are dynamically deployed.**
- ◆ **Can be used to control and monitor any other applications. Agents can be used to supervise applications, to restart or reconfigure them, and to notify other services when certain conditions are detected.**
- ◆ **Provides a secure administration interface which allows to remotely control (start / stop/ reconfigure / upgrade) distributed services or applications.**
- ◆ **The Agent system in the MonALISA framework can be used to develop higher level services, implemented as a distributed network of communicating agents, to perform global optimization tasks.**

**It proved to be a stable and reliable distributed service system**

**~200 Sites running MonALISA**

**<http://monalisa.caltech.edu>**