Next Generation Grid Enable Collaborative System

VRVS worldwide collaborative production service
From VRVS to EVO

Philippe Galvez
California Institute of Technology

International ICFA Workshop on HEP Networkir
Daegu, Korea May 27, 2005

National Science Foundation
WHERE DISCOVERIES BEGIN

Outlines

- VRVS: Background Information
- VRVS Deployment;
- VRVS Usage and Statistics

➔ From VRVS To EVO:
An End-to-End Self Managed RTC Infrastructure

Objective

The "Virtual Room Videoconferencing System" (VRVS) has been developed since 1995 in order to provide a low cost, bandwidth- efficient, extensible means for videoconferencing and remote collaboration over networks within the High Energy and Nuclear Physics communities.

May 2005:

- **16,200 Users Registered from 120 Countries, more than 1100 world wide meetings involving more than 4500 users (total 6000 hours) per month**
- **It is first Very Large Distributed System (using the Grid concept) deployed and used today in Production.**

VRVS Project Timeline

Version

- **v0.4** — 1995 Caltech/CMS group started the development of a full Web based user interface for videoconferencing.
- **v1.0** — In January 1997, pushed by strong demands from the LHC experiments, the Caltech/CMS group started a production prototype Web-based service named the Virtual Room Videoconferencing System (VRVS).
- **v2.0** — During 1998-2000 VRVS was widely recognized by the Research and Education Communities worldwide. It became a core technology for IP-based video & multimedia services in Internet2.
- **v2.5** — July 2001 VRVS is first system to support multiple protocols (Mbone, H.323, MPEG) for collaboration over IP network.
- **v3.0** — February 2003, first re-architect of the VRVS System (97% Code redone) – User Authentication, Database, AccessGrid Support, .. As of May 2005: 16200 users registered, 22000 meetings performed
- **v4.0** — July 2005, second re-architect of the VRVS System to become a Globally Distributed Self Managed End-to-End Real-time Infrastructure

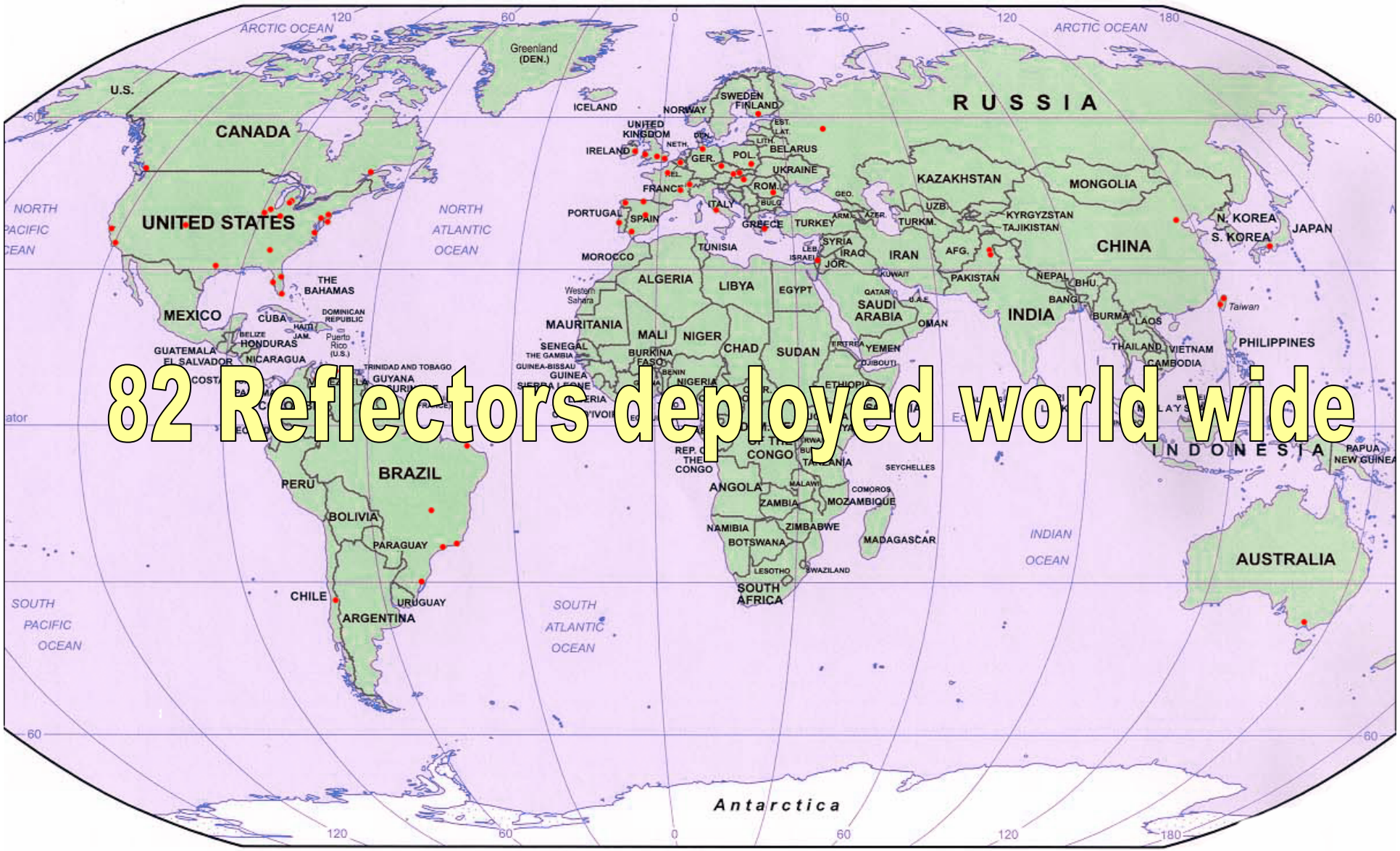
VRVS: What it is ?

- VRVS is a realtime distributed system which provides a scalable communication infrastructure for large collaboration dispersed all over the world.
- Different technologies and protocols are supported (and mixed) and allow users to connect their preferred videoconference.
- Supports Mbone, H.323, SIP, QuickTime, Access Grid, JMF and MPEG2.
- The system is composed of 1 main server and several reflectors (network servers) spread around the world.

VRVS Web Service Design

- **Unified Web User Interface** to schedule and join/leave a meeting **independently of the application.**
- **Multi-platform: Windows, Linux, MacOS and Unix.**
- **Easy to use: Everybody knows how to click on a web page today.**
- **Virtual Room Concept, Scheduling; Create a virtual space were people can exchange real-time information.**
- **Join or Leave a Collaborative session anytime.** ***Do not need to know in advance how many participants and booked ports capacity. Just announce the meeting and people will join from anywhere.***
- **Full Documentation and Tutorial**
- **Self service: Don't need a technician or expert to organize and join a conference.**

VRVS Reflectors Deployment World Wide

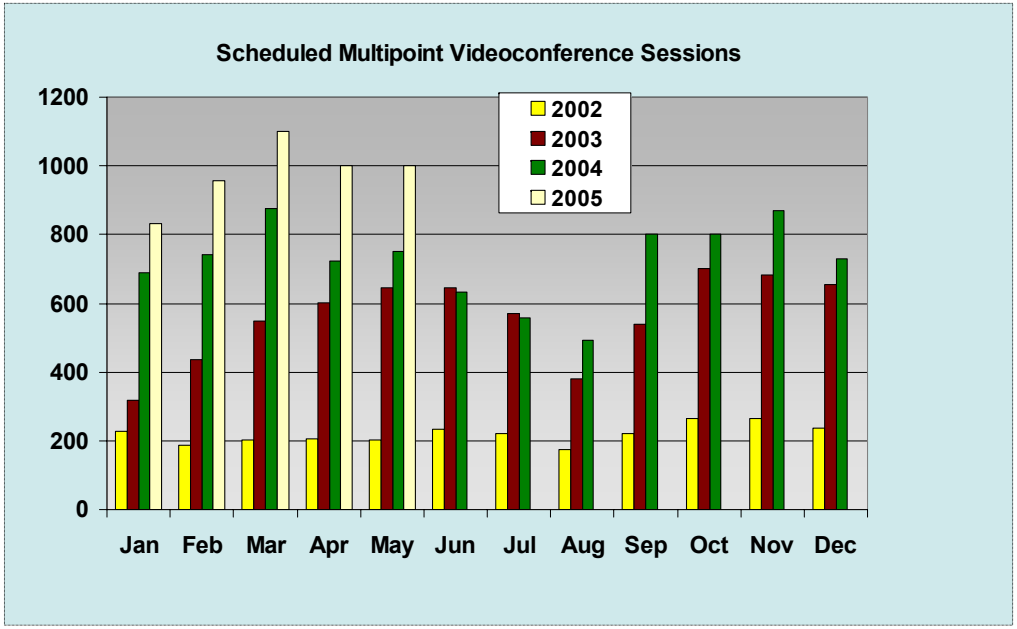


VRVS registered users and current usage

16,200 different Users Registered from 120 Countries

USA	3685
Spain	1768
France	1222
Italy	1115

Switzerland, Germany, Brazil, UK, Slovakia, Taiwan, Chile, Greece, Argentina, Japan, Russia, Canada, etc…



Average of 1100 world wide meetings involving more than 4500 users (total 6000 hours) per month

Workshops/ Conferences	Since 2004/01/01
Workshop	277
Lecture	33
Conference	265
Seminar	160
Tutorial	39

VRVS Worldwide Collaboration

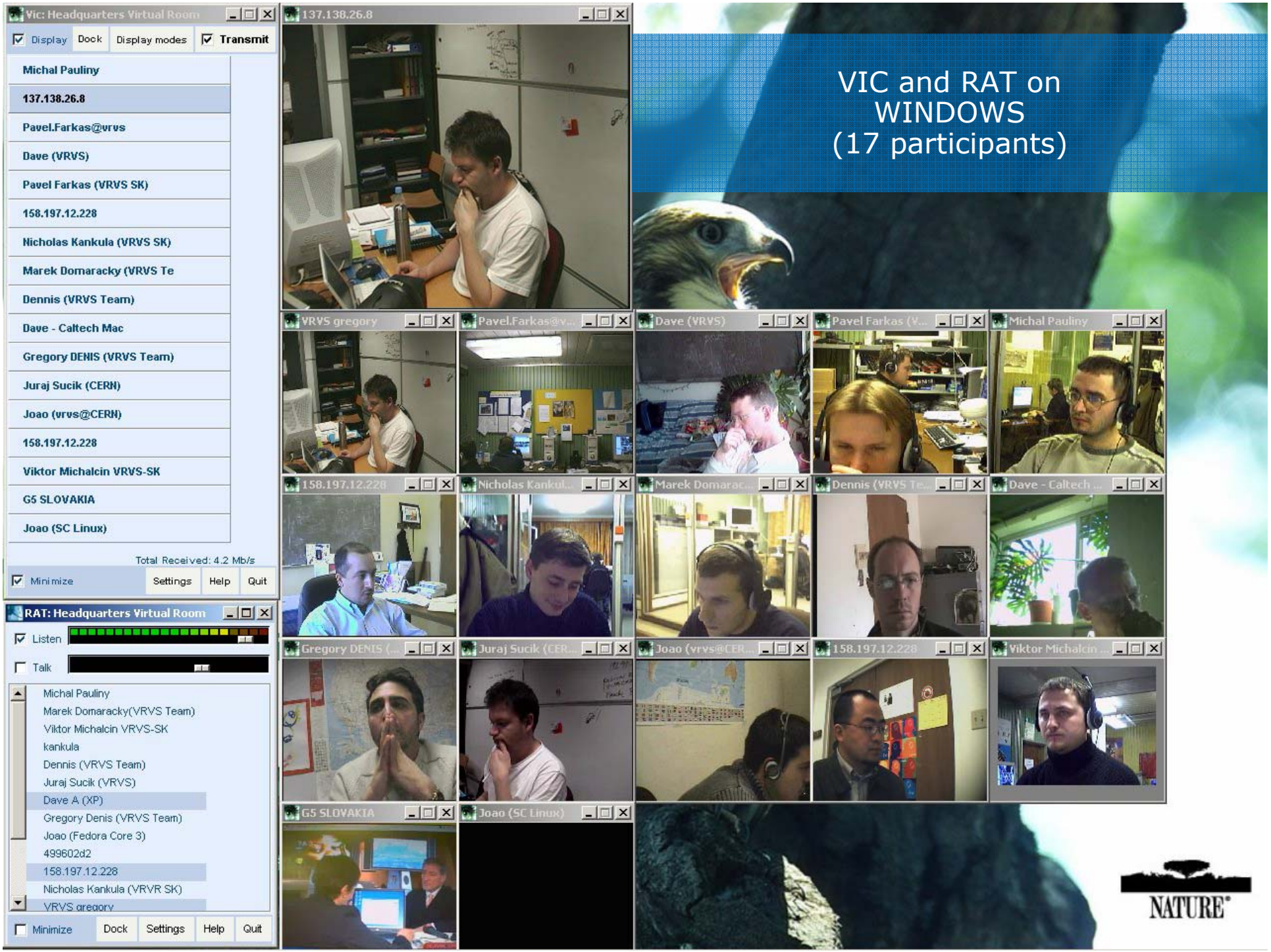
The VRVS team collaborates with Research and Education Networks and major Research Projects around the globe to provide the academic community with a unique and reliable real-time infrastructure supporting all protocols for advanced collaboration

VRVS National Research and Education Network (NREN)

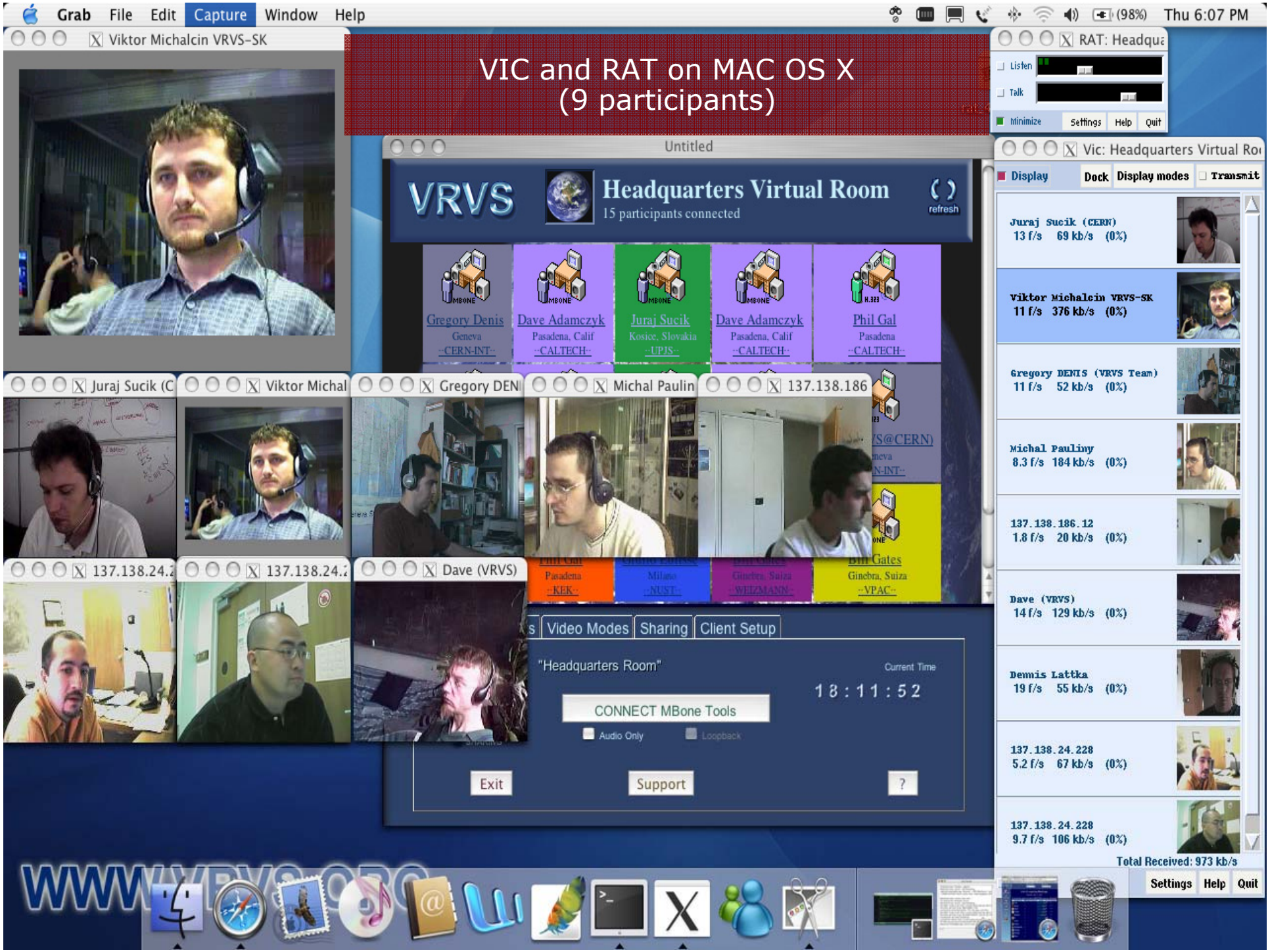
Communities and Project:

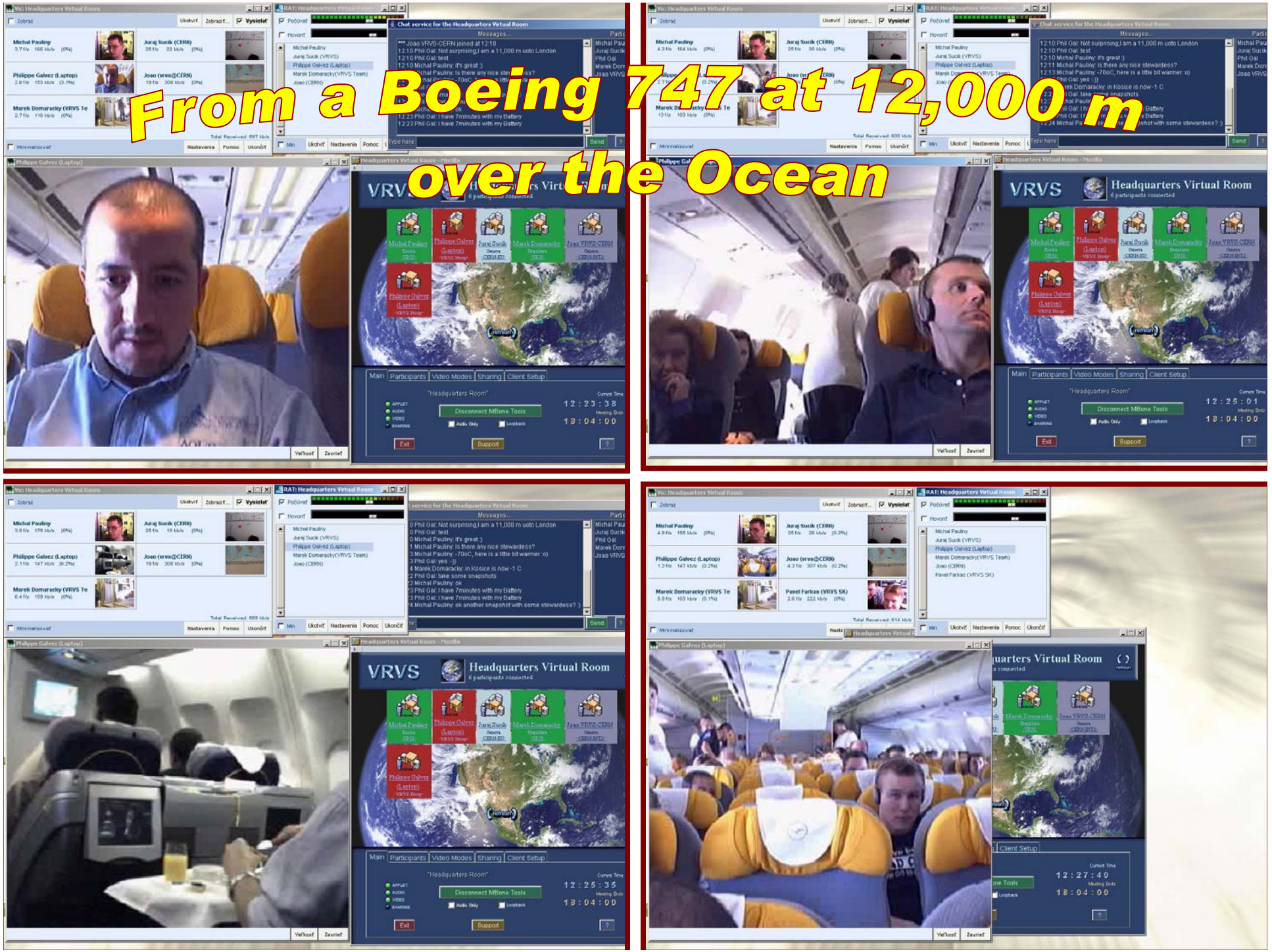
Internet2 **(U.S)**, GEANT2/DANTE **(Europe)**, RedIRIS **(Spain)**, RNP **(Brazil)**, REUNA **(Chile)**, RENATER **(France)**, SANET **(Slovakia)**, INFN **(Italy)**, FUNET **(Finland)**, REACCIUN2 (Venezuela) and AMPATH **(America)**, **GLORIAD…**

Others are in process.

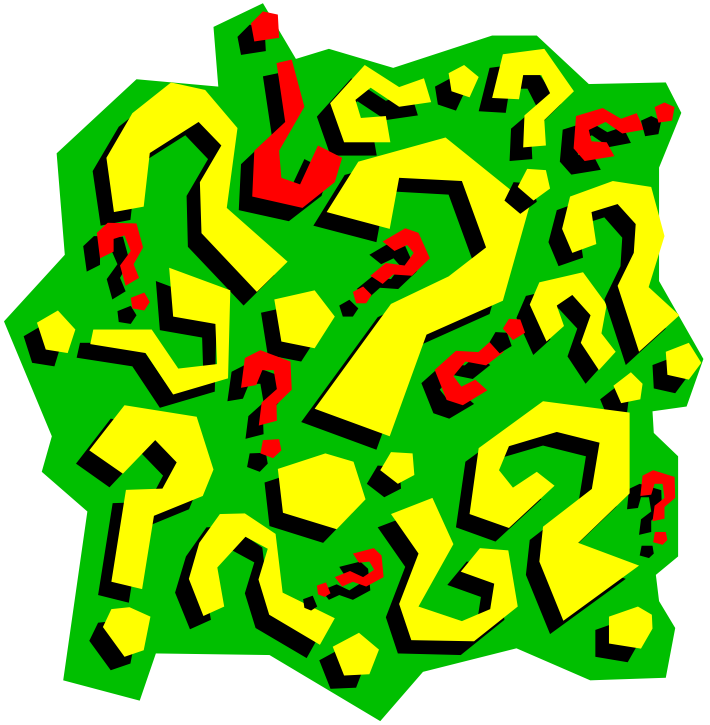

Vic: Headquarters Virtual Room
Display
Dock
Display modes
Transmit
Michal Pauliny
137.138.26.8
Pavel.Farkas@vrvs
Dave (VRVS)
Pavel Farkas (VRVS SK)
158.197.12.228
Nicholas Kankula (VRVS SK)
Marek Domaracky (VRVS Te
Dennis (VRVS Team)
Dave - Caltech Mac
Gregory DENIS (VRVS Team)
Juraj Sucik (CERN)
Joao (vrvs@CERN)
158.197.12.228
Viktor Michalcin VRVS-SK
G5 SLOVAKIA
Joao (SC Linux)
Total Received: 4.2 Mb/s
Minimize
Settings
Help
Quit
RAT: Headquarters Virtual Room
Listen
Talk
Michal Pauliny
Marek Domaracky(VRVS Team)
Viktor Michalcin VRVS-SK
kankula
Dennis (VRVS Team)
Juraj Sucik (VRVS)
Dave A (XP)
Gregory Denis (VRVS Team)
Joao (Fedora Core 3)
499602d2
158.197.12.228
Nicholas Kankula (VRVR SK)
VRVS gregory
Minimize
Dock
Settings
Help
Quit
137.138.26.8
VRVS gregory
Pavel.Farkas@v...
Dave (VRVS)
Pavel Farkas (V...
Michal Pauliny
158.197.12.228
Nicholas Kankul...
Marek Domarac...
Dennis (VRVS Te...
Dave - Caltech ...
Gregory DENIS (...
Juraj Sucik (CER...
Joao (vrvs@CER...
158.197.12.228
Viktor Michalcin ...
G5 SLOVAKIA
Joao (SC Linux)
VIC and RAT on
WINDOWS
(17 participants)
NATURE


MSSR - KM
Dennis (VRVS Team)
Juraj Sucik (CERN)
Gregory DENIS (VRVS Te
TnUAD2 - Trencin
Iosif Legrand
U of Zilina (Melo,Gintner)
TUKE Kosice (CNL)
Viktor Michalcin (VRVS-SK
U of Zilina (Melo,Gintner)
TnUAD - Trencin
131.215.116.60
MSSR - KM
UMB Banska Bystrica
Julian Bunn
PF UPJS Kosice
128.227.89.161
Dave (VRVS)
VIC on LINUX
(17 participants)


VIC and RAT on MAC OS X
(9 participants)
Grab File Edit Capture Window Help
Thu 6:07 PM
Viktor Michalcin VRVS-SK
RAT: Headqua
Listen
Talk
Minimize
Settings
Help
Quit
Untitled
VRVS
Headquarters Virtual Room
15 participants connected
refresh
Gregory Denis
Geneva
--CERN-INT--
Dave Adamczyk
Pasadena, Calif
--CALTECH--
Juraj Sucik
Kosice, Slovakia
--UPJS--
Dave Adamczyk
Pasadena, Calif
--CALTECH--
Phil Gal
Pasadena
--CALTECH--
Vic: Headquarters Virtual Ro
Display
Dock
Display modes
Transmit
Juraj Sucik (C
Viktor Michal
Gregory DEN
Michal Paulin
137.138.186
137.138.24.2
137.138.24.2
Dave (VRVS)
Video Modes
Sharing
Client Setup
"Headquarters Room"
Current Time
18:11:52
CONNECT MBone Tools
Audio Only
Loopback
Exit
Support
?
Juraj Sucik (CERN)
13 f/s 69 kb/s (0%)
Viktor Michalcin VRVS-SK
11 f/s 376 kb/s (0%)
Gregory DENIS (VRVS Team)
11 f/s 52 kb/s (0%)
Michal Pauliny
8.3 f/s 184 kb/s (0%)
137.138.186.12
1.8 f/s 20 kb/s (0%)
Dave (VRVS)
14 f/s 129 kb/s (0%)
Dennis Lattka
19 f/s 55 kb/s (0%)
137.138.24.228
5.2 f/s 67 kb/s (0%)
137.138.24.228
9.7 f/s 106 kb/s (0%)
Total Received: 973 kb/s
Settings
Help
Quit
WWW.VRVS.ORG


From a Boeing 747 at 12,000 m
over the Ocean

What's Next ?

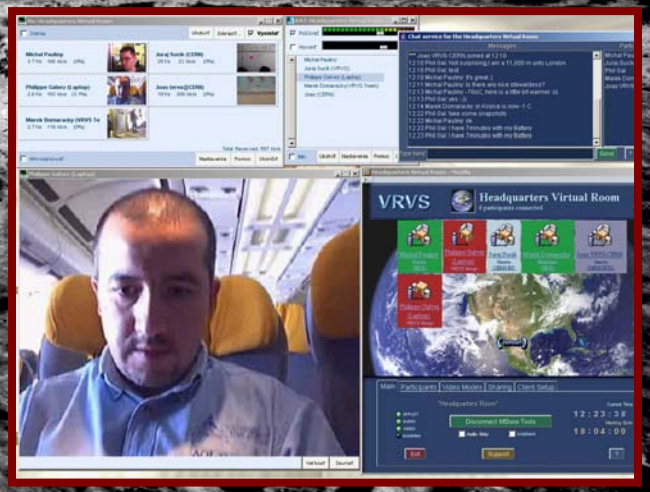


Caltech

NASA'S PLANETARY PHOTOJOURNAL
DEVELOPMENT TEAM

VRVS in Space

Not for the LHC program

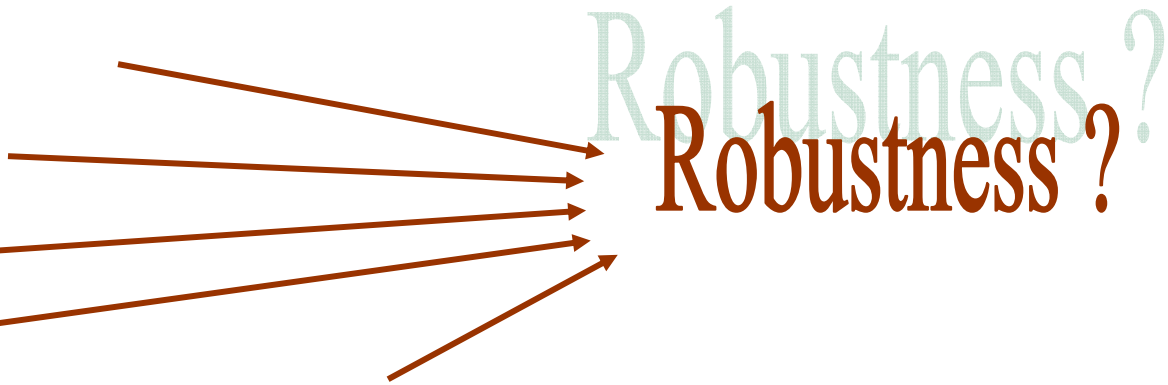


From VRVS To EVO: End-to-End Self Managed RTC Infrastructure

Developing/Deploying a Very Robust and Scalable Real-Time Collaborative (RTC) Infrastructure is a real problem that no-one was able to fix yet !

Main critical RTC issues:

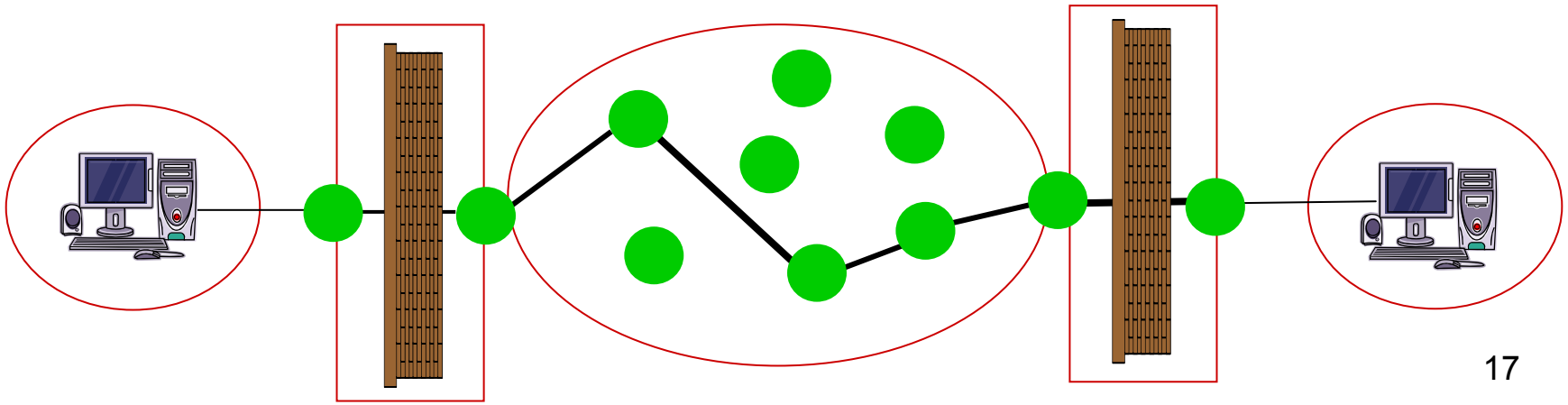
- **Manageability**
- **Reliability**
- **Scalability**
- **Security**
- **Functionality / Usability**



From VRVS To EVO: End-to-End Self Managed RTC Infrastructure

What's wrong ? Why can we not achieve total reliability/robustness when deploying a RTC Infrastructure ?

- The Real-Time Collaborative environment is a living environment, constantly changing, evolving. In addition devices/domains/nodes are managing by several independent technical and administrative entities



Solutions ?

By creating a "living" RTC Infrastructure capable to react/adapt to the change of the environment in real-time transparently to the end-user

Caltech

Building a Scalable RTC Infrastructure

- **Multicast Network: It is not any more a global solution**
  - **After more than 10 years of development/deployment, It reaches its maximum level of reliability/robustness**
  - **It is just a transport mechanism**
    - **No QoS**
    - **No awareness of the Application level**
    - **No data processing as required by the some application (mixing audio, select video streams, ….)**
  - **Extremely difficult to troubleshot because it crosses several administrative domains**

Today's solution:

Building an Overlay Network by deploying intelligent Software Agents communicating using Unicast connections and capable of using Multicast transport when it makes more sense

Panda Software Agent

Intelligent Software Agent to create an Overlay Network

Some functionalities:

- **Dynamic registration** to high level directory services
- **Automatic re-activation** of components and services
- **Automatic and secure** code update
- **Continuous monitoring** of network quality (packet loss, jitter, latency) between its peers and its possible peers
- **Automatic rerouting** to obtain the best performance/quality
- **Encryption** between reflectors and between reflector and VRVS clients
- **Automatic Alarm notifications** when monitored parameters (system or network) go beyond a **preset threshold**
- **Dynamically provides services** (video, audio, data,..) that matches the current **resources/capabilities** to the end users/applications
- Provides access to **real-time and historical data**

Koala Software Agent Functionality

Intelligent Software Agent running in the End-System

Some functionalities:

- New Java-based VRVS client that will perform Dynamic Registration to high level directory services – (Multi-OS support)
- Automatic detection of the system parameters (CPU, Memory,..) hardware components (Audio card, video card, …), services capabilities (video, audio, …), network environment and capabilities (wireless environment, DSL, available bandwidth, …)
- Dynamically gets services (video, audio, data,..) that matches the current resources/capabilities to end users/applications
- Continuous monitoring of network quality (packet loss, jitter) latency) and Automatic rerouting of packets
- Automatic Alarm notifications when monitored parameters (system or network) go beyond a preset threshold.

Building a Core RTC Infrastructure

Directory Services

Directory Services

Register the Service offered

Notification to others members

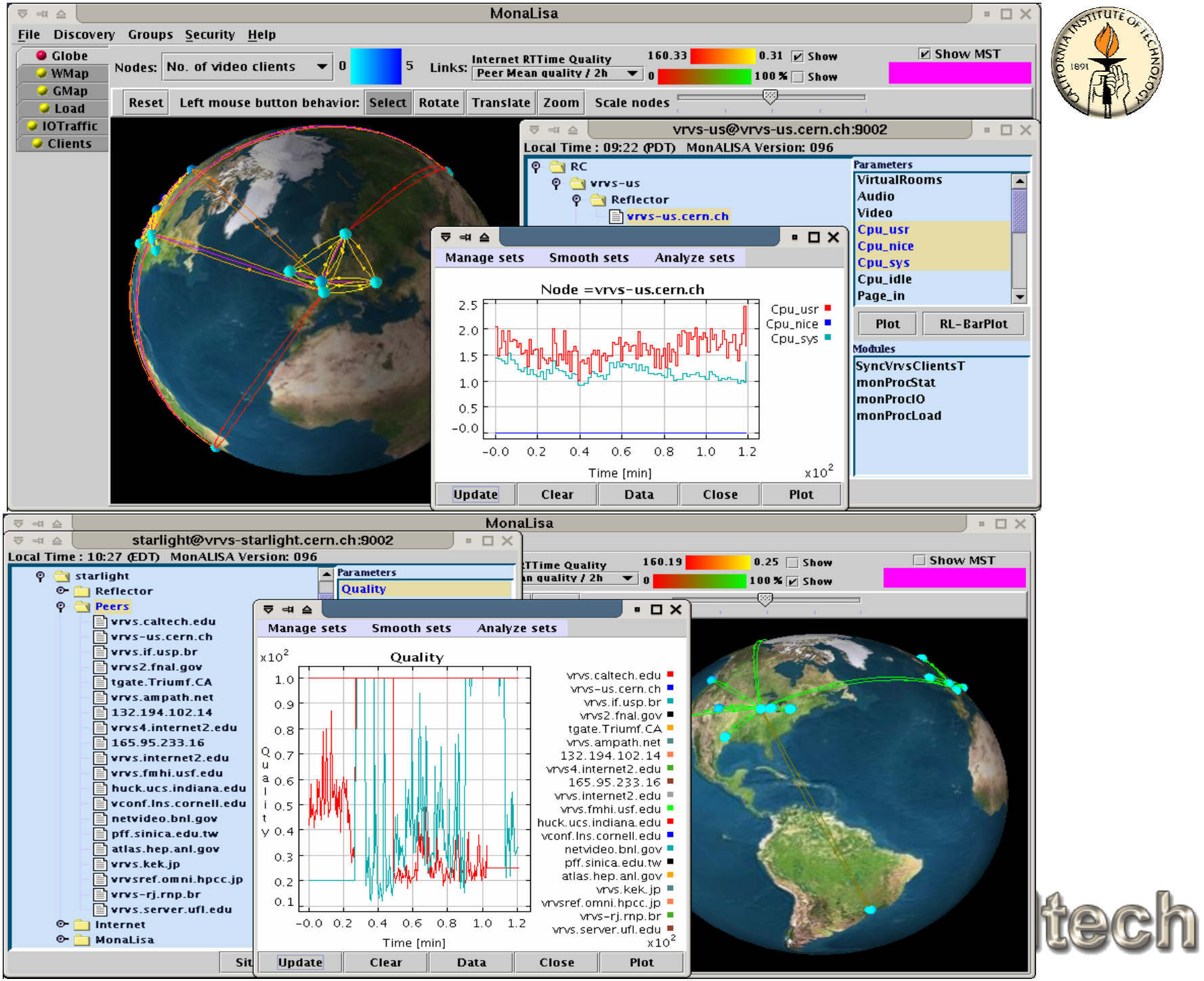
Access Monitoring Information and Remote Administrated the Different Services

Monitoring Service

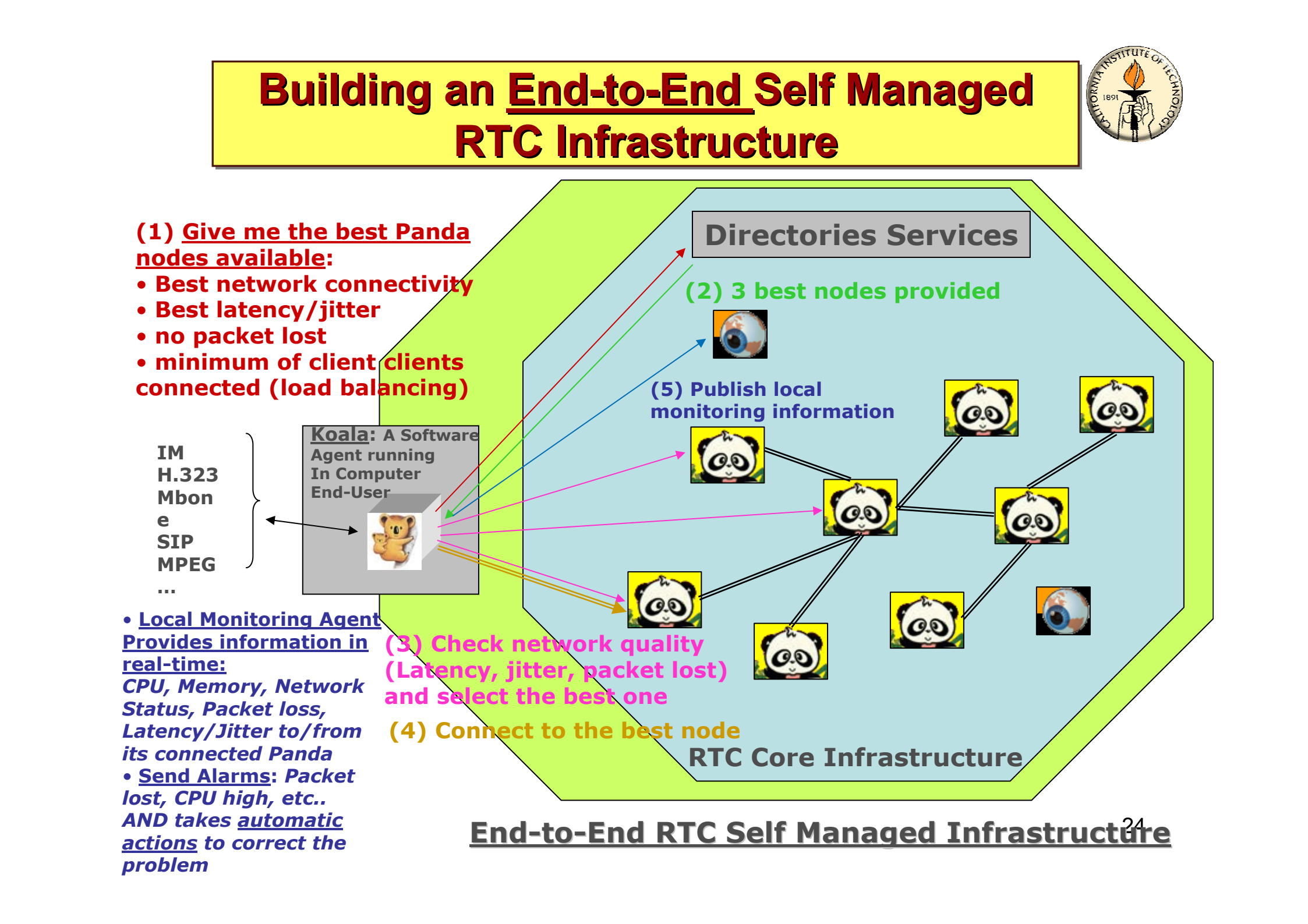
GUI for Monitoring

- Send Monitoring Information in real-time *CPU, Memory, Network Status, Packet loss, Latency/Jitter with its peers, number of clients connected, etc..*
- Send Alarms to Administrators: *Packet lost, CPU high, etc..*

Overlay Network built using Intelligent Software Agent: Panda

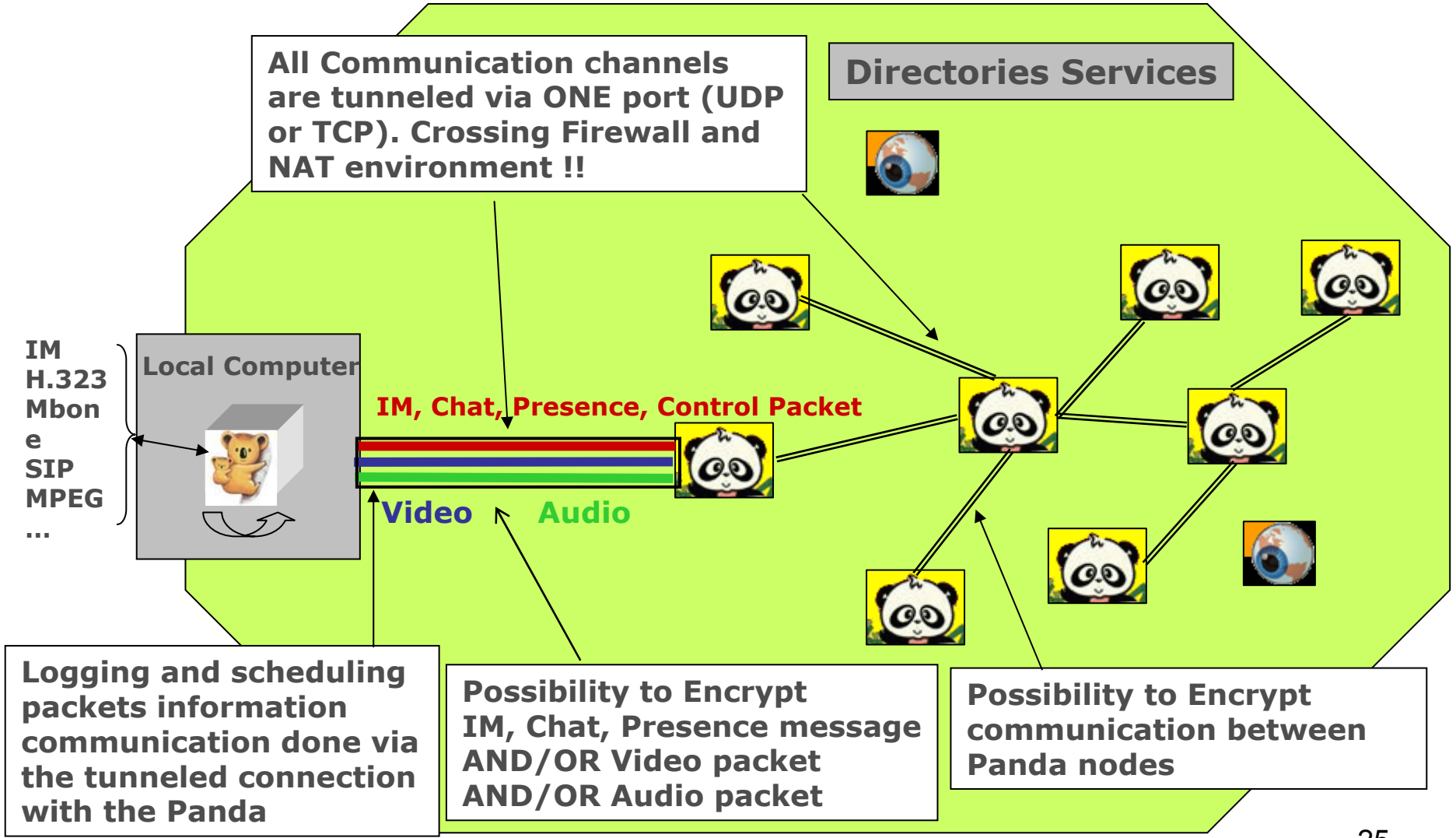

MonaLisa
File Discovery Groups Security Help
Globe
WMap
GMap
Load
IOTraffic
Clients
Nodes: No. of video clients
Links: Internet RTTime Quality
Peer Mean quality / 2h
Show MST
Reset
Left mouse button behavior: Select Rotate Translate Zoom
Scale nodes
vrvs-us@vrvs-us.cern.ch:9002
Local Time : 09:22 (PDT) MonALISA Version: 096
RC
vrvs-us
Reflector
vrvs-us.cern.ch
Parameters
VirtualRooms
Audio
Video
Cpu_usr
Cpu_nice
Cpu_sys
Cpu_idle
Page_in
Plot
RL-BarPlot
Modules
SyncVrvsClientsT
monProcStat
monProcIO
monProcLoad
Manage sets
Smooth sets
Analyze sets
Node =vrvs-us.cern.ch
Time [min]
Update
Clear
Data
Close
Plot
MonaLisa
starlight@vrvs-starlight.cern.ch:9002
Local Time : 10:27 (EDT) MonALISA Version: 096
starlight
Reflector
Peers
vrvs.caltech.edu
vrvs-us.cern.ch
vrvs.if.usp.br
vrvs2.fnal.gov
tgate.Triumf.CA
vrvs.ampath.net
132.194.102.14
vrvs4.internet2.edu
165.95.233.16
vrvs.internet2.edu
vrvs.fmhi.usf.edu
huck.ucs.indiana.edu
vconf.lns.cornell.edu
netvideo.bnl.gov
pff.sinica.edu.tw
atlas.hep.anl.gov
vrvs.kek.jp
vrvsref.omni.hpcc.jp
vrvs-rj.rnp.br
vrvs.server.ufl.edu
Internet
MonaLisa
Quality
Manage sets
Smooth sets
Analyze sets
Quality
Time [min]
Update
Clear
Data
Close
Plot

Building an End-to-End Self Managed RTC Infrastructure



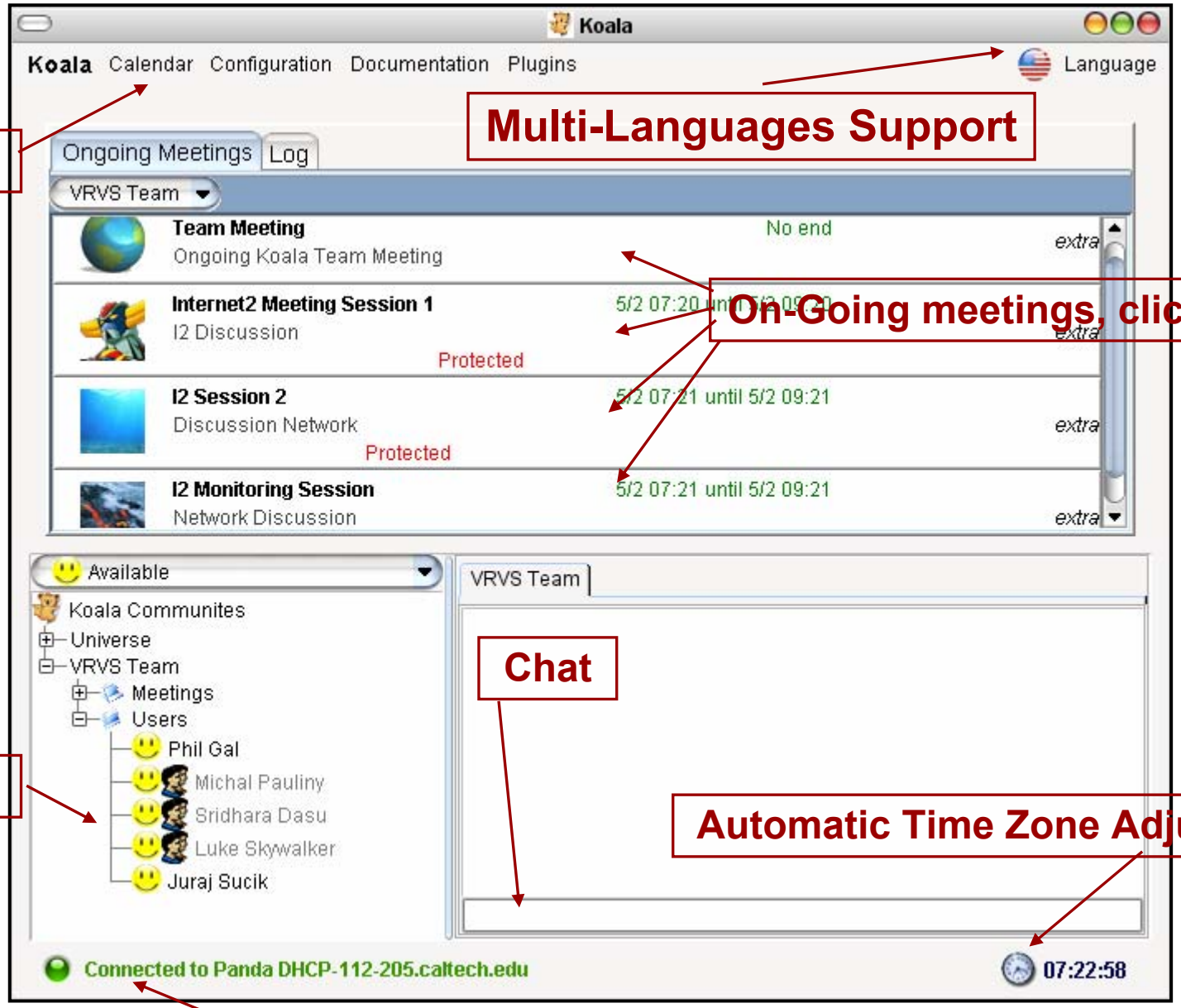
End-to-End RTC Self Managed Infrastructure

Building an End-to-End Self Managed RTC Infrastructure



End-to-End RTC Self Managed Infrastructure

The Koala Client (Java)



Bookings

Multi-Languages Support

On-Going meetings, click to join

Presence

Chat

Automatic Time Zone Adjustment

Connection to the Panda

Video and Audio Client improvements

- improvements done on decoder side have influence mainly on compatibility with H.323 hardware devices
  - RTP part was changed to avoid artifacts in decoded video from some H.323 clients and to fix bad positions of macroblocks in video stream from MCUs and Polycom VSX
  - bug which causes skipping of the last macroblock in RTP packet was fixed (avoids artifacts in decoded video)

Polycom VSX7000 - before

Polycom VSX7000 - now

New Codec Integration : H.263

- compare to H.261, there are two significant differences:

Resolution of video signal

- standardized picture formats: sub-QCIF (128x96), QCIF (176x144), CIF (352x288), 4CIF (704x576), 16CIF (1408x1152)
- custom picture formats: resolution up to 2048 x 1152

Level of compression and quality

- Arithmetic Coding instead of VLC - significantly fewer bits produced
- Unrestricted Motion Vector Mode - larger vectors and vectors can point outside the picture
- Advanced Prediction Mode - 4 8x8 vectors instead of one 16x16 vector - results in less blocking artifacts
- PB-frames mode - two pictures encoded as one unit
- enhanced error resilience capabilities
- half pixel precision used for motion compensation + many others

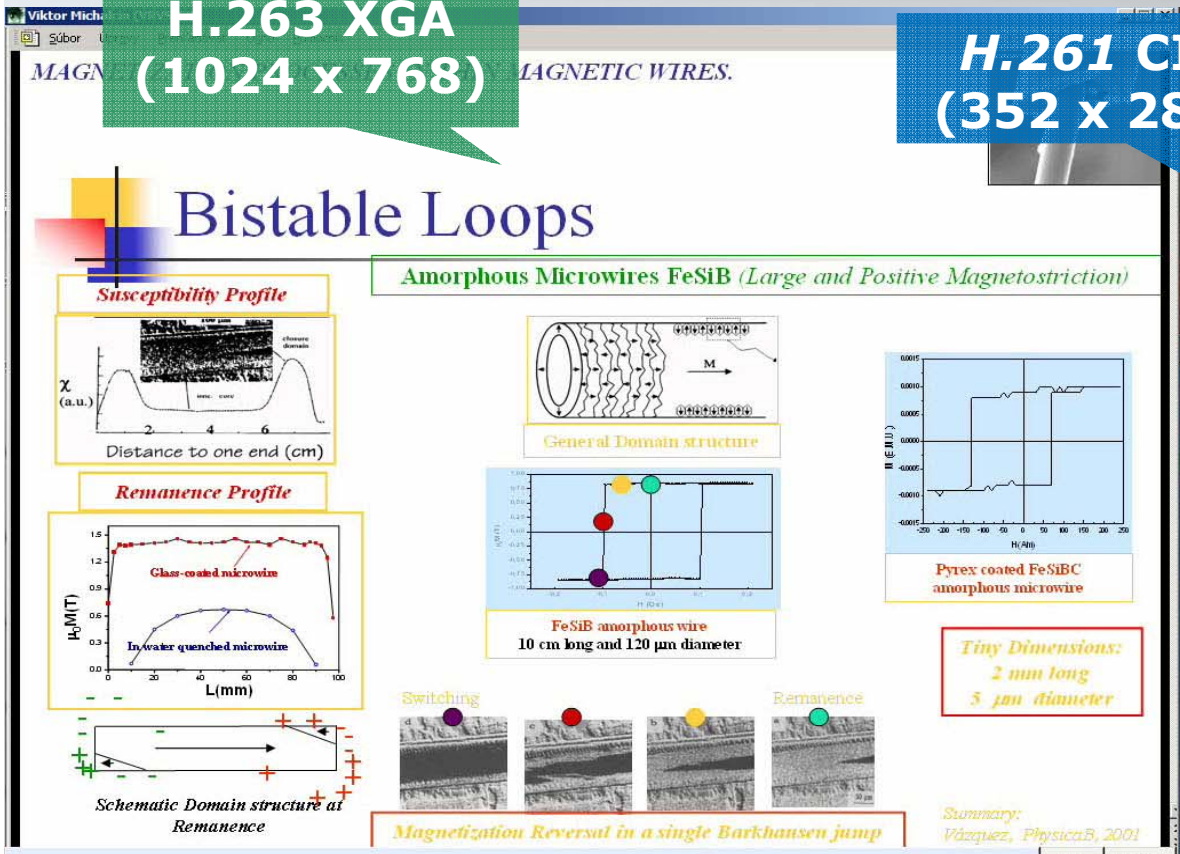
H.263 - Desktop Sharing

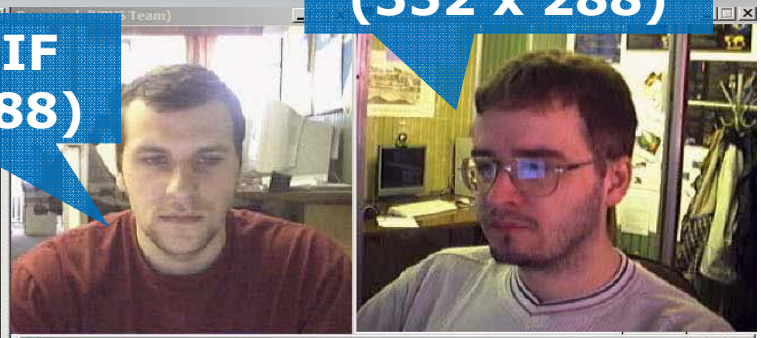
H.263 XGA (1024 x 768)

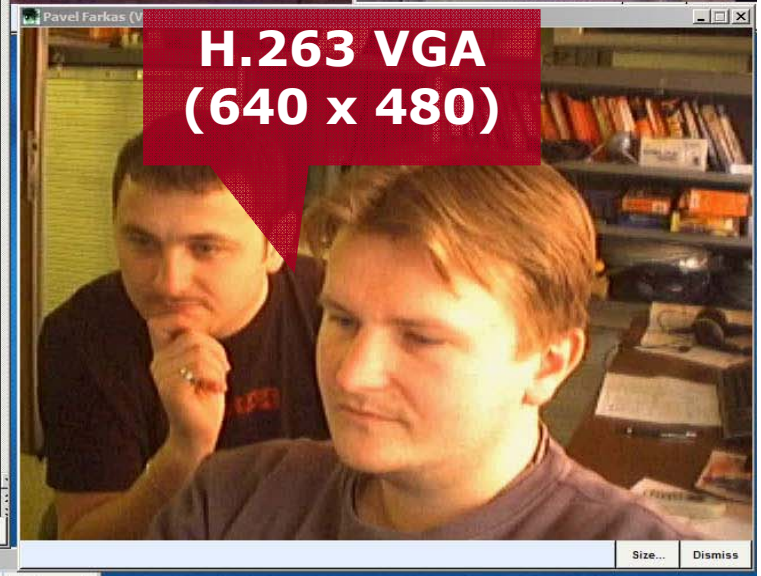
H.261 CIF (352 x 288)

H.261 CIF (352 x 288)

H.263 VGA (640 x 480)







PocketVRVS – the mobile solution

Documentation

Download PocketVRVS package

Booking

Connect to any ongoing meeting

PocketVRVS
web interface

Control Panel
Talk, Listen, Send and Receive controls

Video Screen
Remote Video, Own Video in PIP window

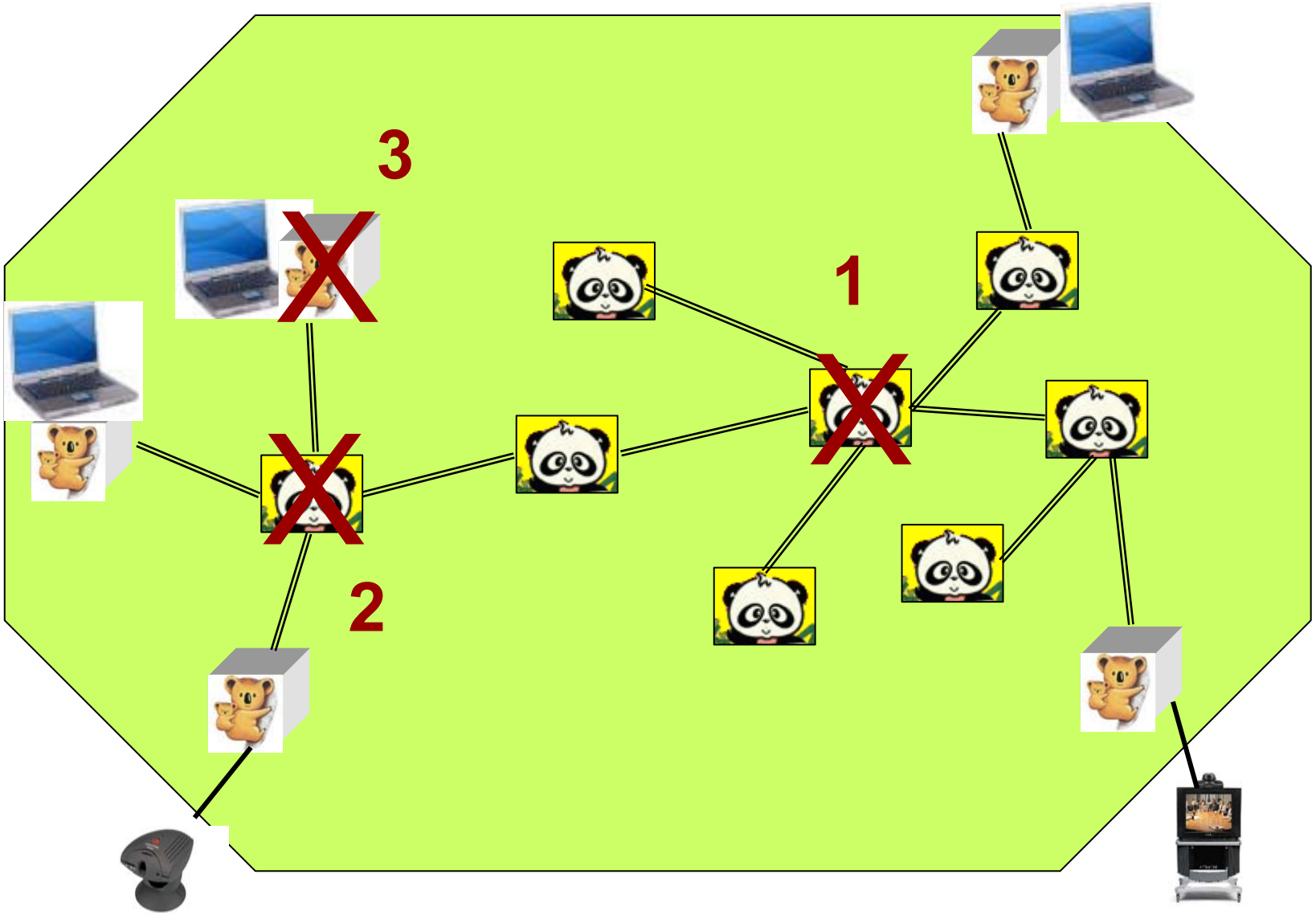
Session Info
Virtual Room, Participant's name, ...

Program Menu
Connection and Session Settings

PocketVRVS
application GUI

EVO: End-to-End Self Managed and Secure RTC Infrastructure

Demonstration: Let's break the System !



Thank you

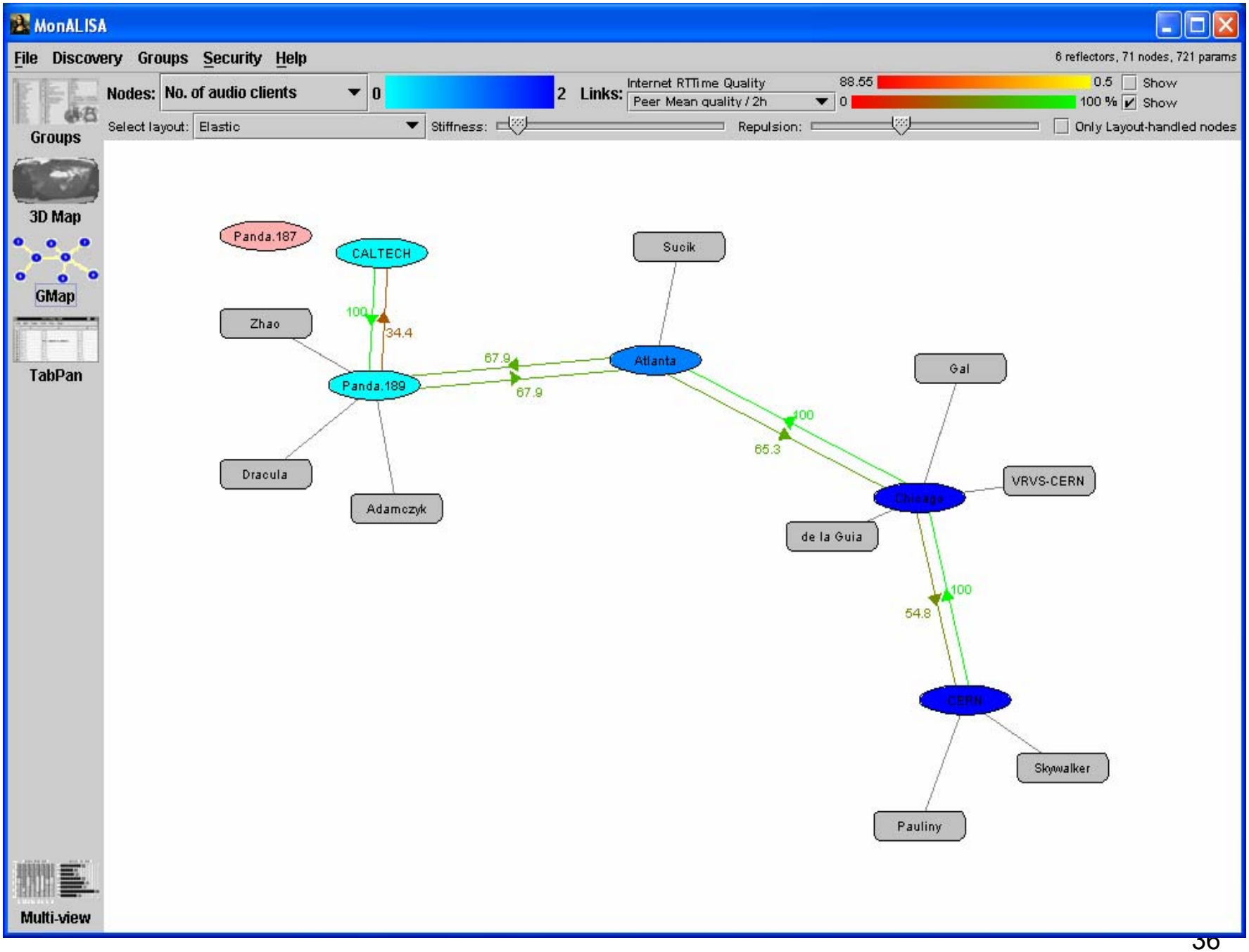
www.VRVS.org

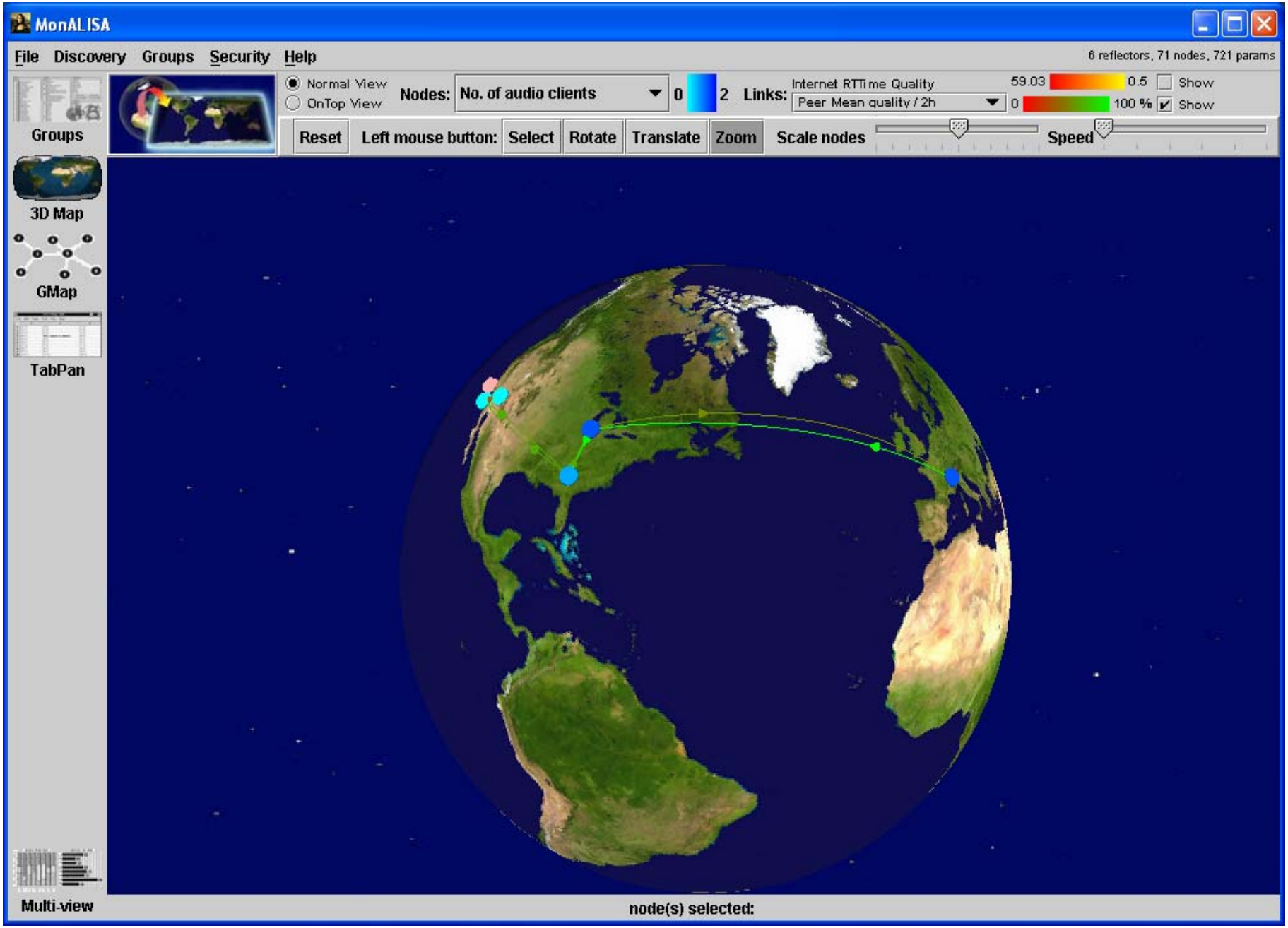
Contact@VRVS.org

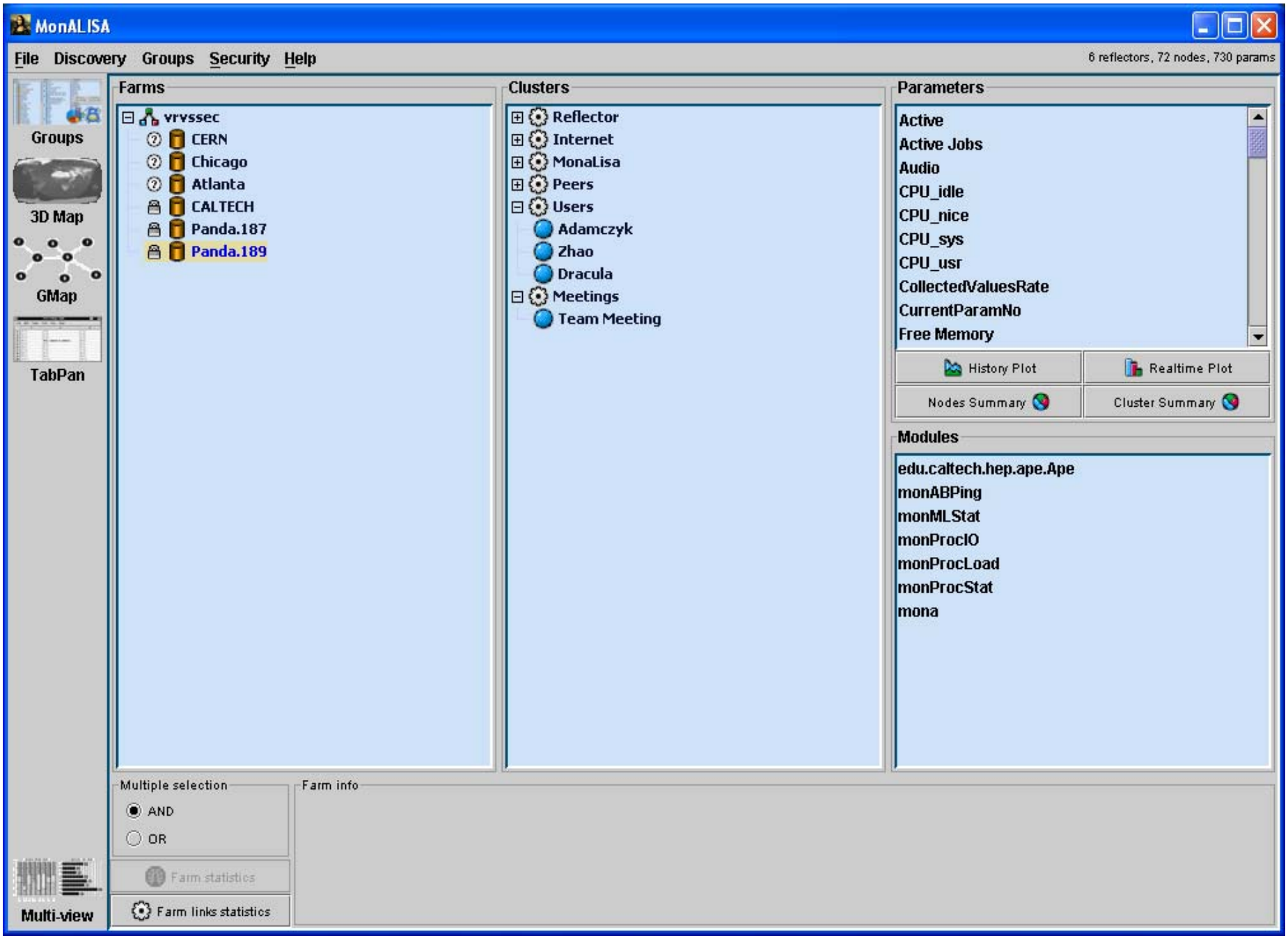
Support@VRVS.org

Additional Slides


Koala
Koala Calendar Configuration Documentation Plugins
Language
Team Meeting
Log
Phil Gal
Joao VRVS-CEF
Juraj Sucik
Dolores de la...
Gregory Denis
Michal Pauliny
Show Client Controls
Leave this meeting
Available
Users
Dave Adamczyk
Phil Gal
Nan Zhao
Count Dracula
Joao VRVS-CERN
Juraj Sucik
Dolores de la Guia
Gregory Denis
Michal Pauliny
VRVS Team
Luke Skywalker
Team Meeting
Connected to Panda vrvs-bkup-starlight.cern.ch
08:09:13


MonALISA
File Discovery Groups Security Help
6 reflectors, 71 nodes, 721 params
Nodes: No. of audio clients 0 2 Links: Internet RTTime Quality 88.55 0.5 Show
Peer Mean quality / 2h 0 100 % Show
Select layout: Elastic Stiffness: Repulsion: Only Layout-handled nodes
Groups
3D Map
GMap
TabPan
Multi-view
Panda.187
CALTECH
Sucik
Zhao
100
34.4
67.9
67.9
Atlanta
Gal
Panda.189
100
65.3
Dracula
Adamczyk
VRVS-CERN
Chicago
de la Guia
100
54.8
CERN
Skywalker
Pauliny


MonALISA
File
Discovery
Groups
Security
Help
6 reflectors, 71 nodes, 721 params
Groups
3D Map
GMap
TabPan
Multi-view
Normal View
OnTop View
Nodes:
No. of audio clients
0
2
Links:
Internet RTTime Quality
59.03
0.5
Show
Peer Mean quality / 2h
0
100 %
Show
Reset
Left mouse button:
Select
Rotate
Translate
Zoom
Scale nodes
Speed
node(s) selected:


MonALISA
File
Discovery
Groups
Security
Help
6 reflectors, 72 nodes, 730 params
Groups
3D Map
GMap
TabPan
Multi-view
Farms
vrvssec
CERN
Chicago
Atlanta
CALTECH
Panda.187
Panda.189
Clusters
Reflector
Internet
MonaLisa
Peers
Users
Adamczyk
Zhao
Dracula
Meetings
Team Meeting
Parameters
Active
Active Jobs
Audio
CPU_idle
CPU_nice
CPU_sys
CPU_usr
CollectedValuesRate
CurrentParamNo
Free Memory
History Plot
Realtime Plot
Nodes Summary
Cluster Summary
Modules
edu.caltech.hep.ape.Ape
monABPing
monMLStat
monProcIO
monProcLoad
monProcStat
mona
Multiple selection
AND
OR
Farm statistics
Farm links statistics
Farm info