

DOR (Digital Object Registry)

- * Provides Meas Data Arch Srvc (and additional archive services)
- * Allows Operator/Researcher to archive Meas Data Files, and retrieve them
- * From another I&M srvc, MDA srvc can provide these basic functions: put/update file; get file; delete file
- * Interfaces to the MDA srvc include: https; also ftp?; also sftp?
- * When file is first introduced, it is assumed that file contains type info (extension), metadata, and "file self description" info. A wide range of files and associated metadata is permitted by the MDA srvc.
- * When file is first introduced, MDA srvc assigns a unique identifier, which is a handle, that is attached to the file, and can be used to retrieve the file
- * When file is first introduced, or at any later time, the MDA srvc provides a convenient service to allow Operator/Researcher to add identification metadata so that files are stored in an organized fashion, and can be found by searches/queries.
- * MDA srvc allows Operator/Researcher to view organized files, and find them by searches/queries. Multiple views are supported.
- * Each file is "owned" by a GENI slice and one or more users (operators/researchers)
- * The MDA srvc allows the owner to specify who has read and/or write access to the file.
- * The MDA srvc utilizes the mechanisms provided by the CF to authenticate and authorize users.

MDA Srvc protocol and API

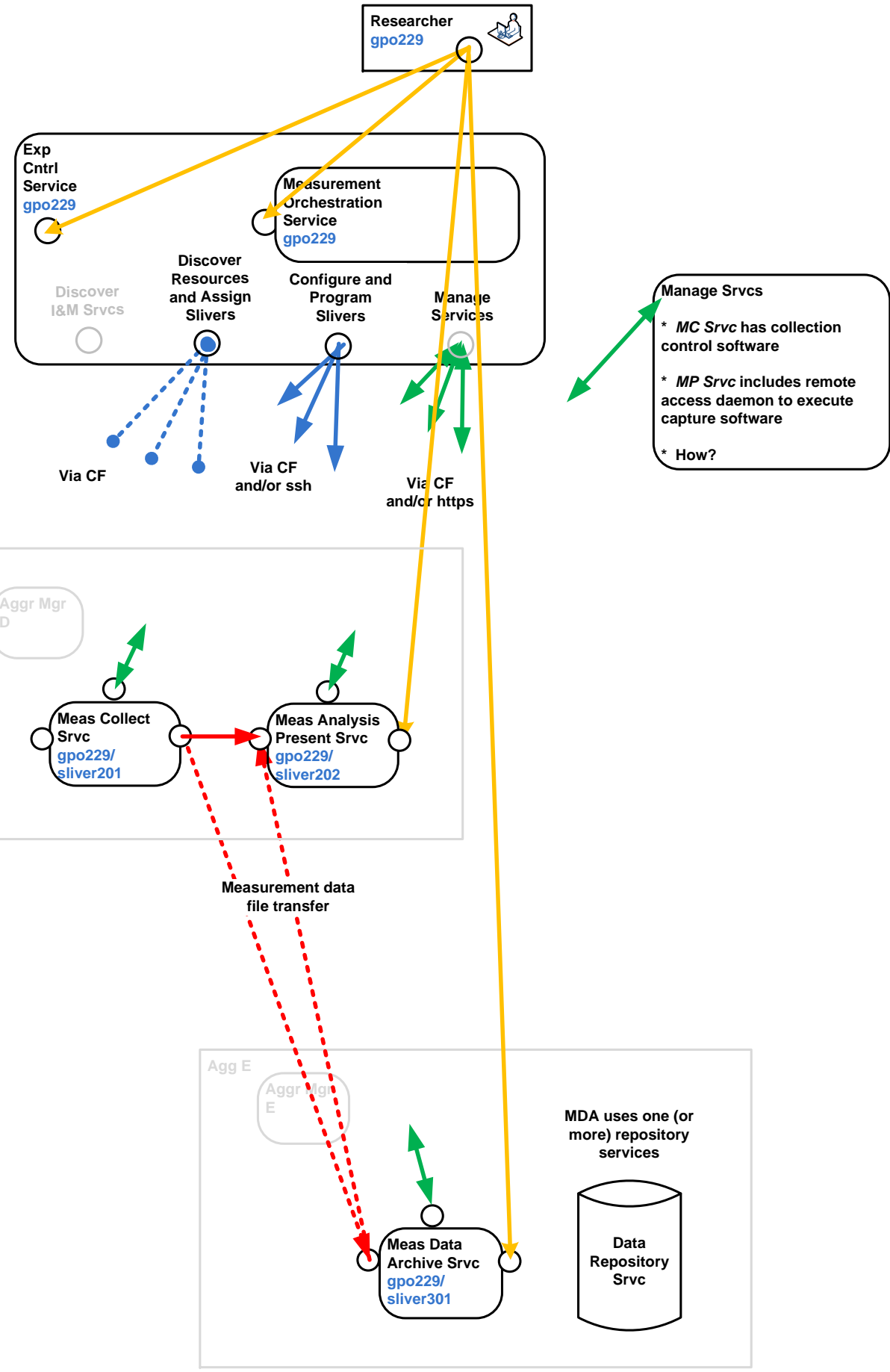
- * Interfaces to the MDA srvc include: https; scp or sftp
- * From another I&M srvc, MDA srvc can provide these basic functions: put/update file; get file; delete file
- * When file is first introduced, it is assumed that file contains type info (extension), metadata, and "file self description" info. A wide range of files and associated metadata is permitted by the MDA srvc.
- * When file is first introduced, MDA srvc assigns a unique identifier, which is a handle, that is attached to the file, and can be used to retrieve the file
- * When file is first introduced, or at any later time, the MDA srvc provides a convenient service to allow Operator/Researcher to add identification metadata so that files are stored in an organized fashion, and can be found by searches/queries.
- * MDA srvc allows Operator/Researcher to view organized files, and find them by searches/queries. Multiple views are supported.

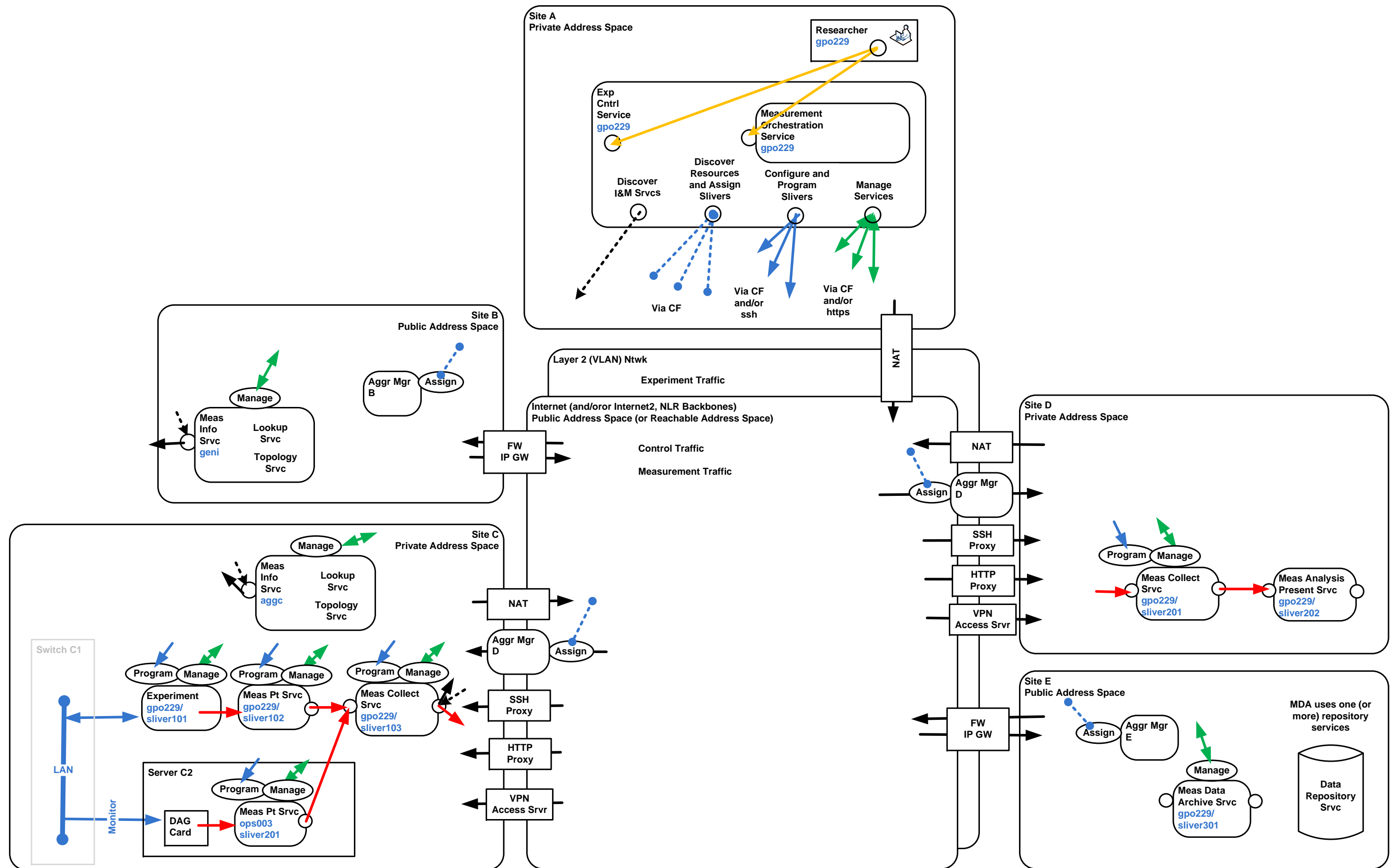
Assume: API as specified by CNRI for ADL (Advanced Distributed Learning) srvc (CNRI)

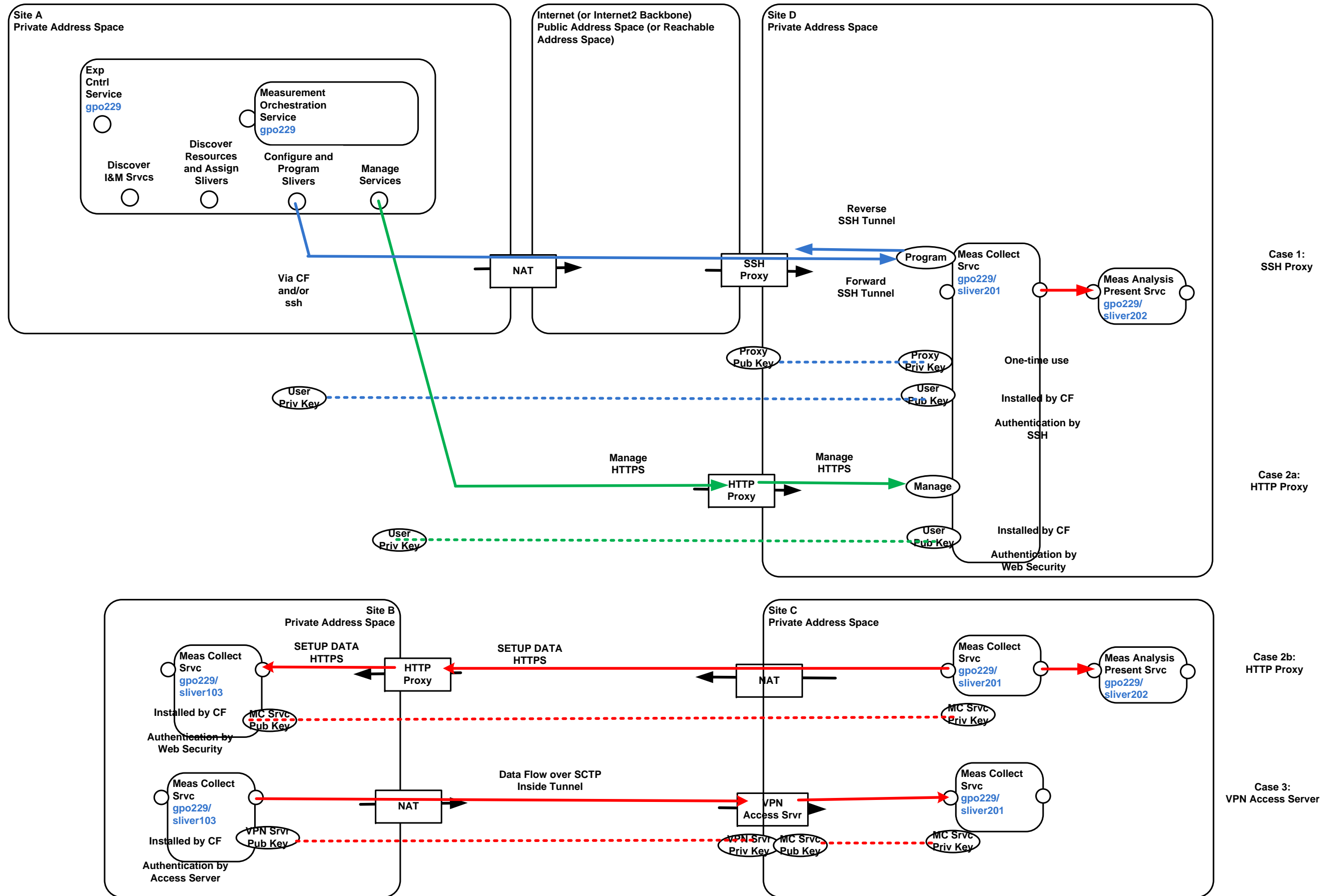
MDA Srvc authentication and authorization

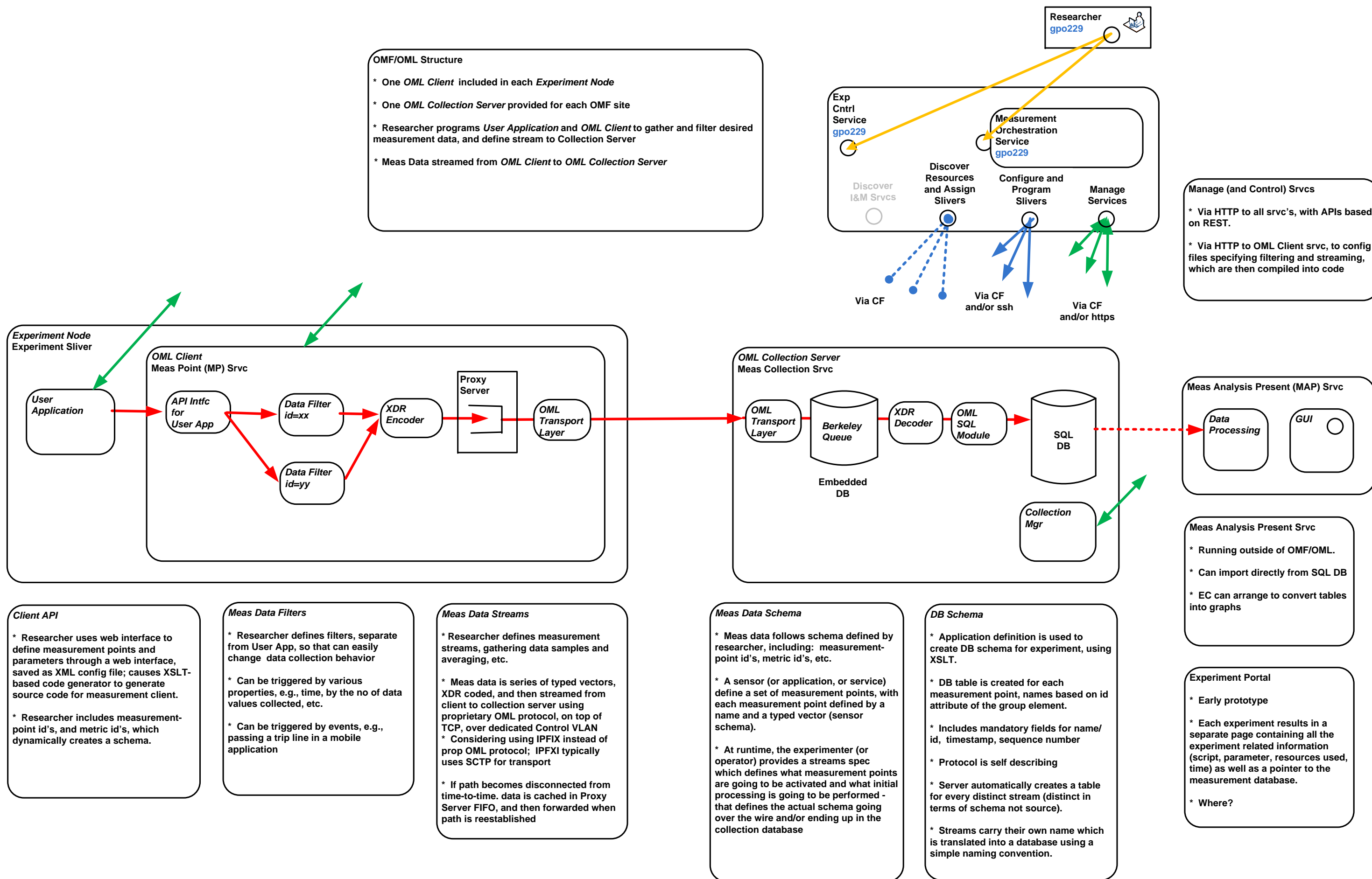
- * Each file is "owned" by a GENI slice and one or more users (operators/researchers)
- * The MDA srvc allows the owner to specify who has read and/or write access to the file.
- * The MDA srvc utilizes the mechanisms provided by the CF to authenticate and authorize users.

Assume: CF drops public keys of authorized users into MDA srvc, so that: presence of key indicates an "account" on the MDA srvc; additional info indicates nature of access (CNRI)









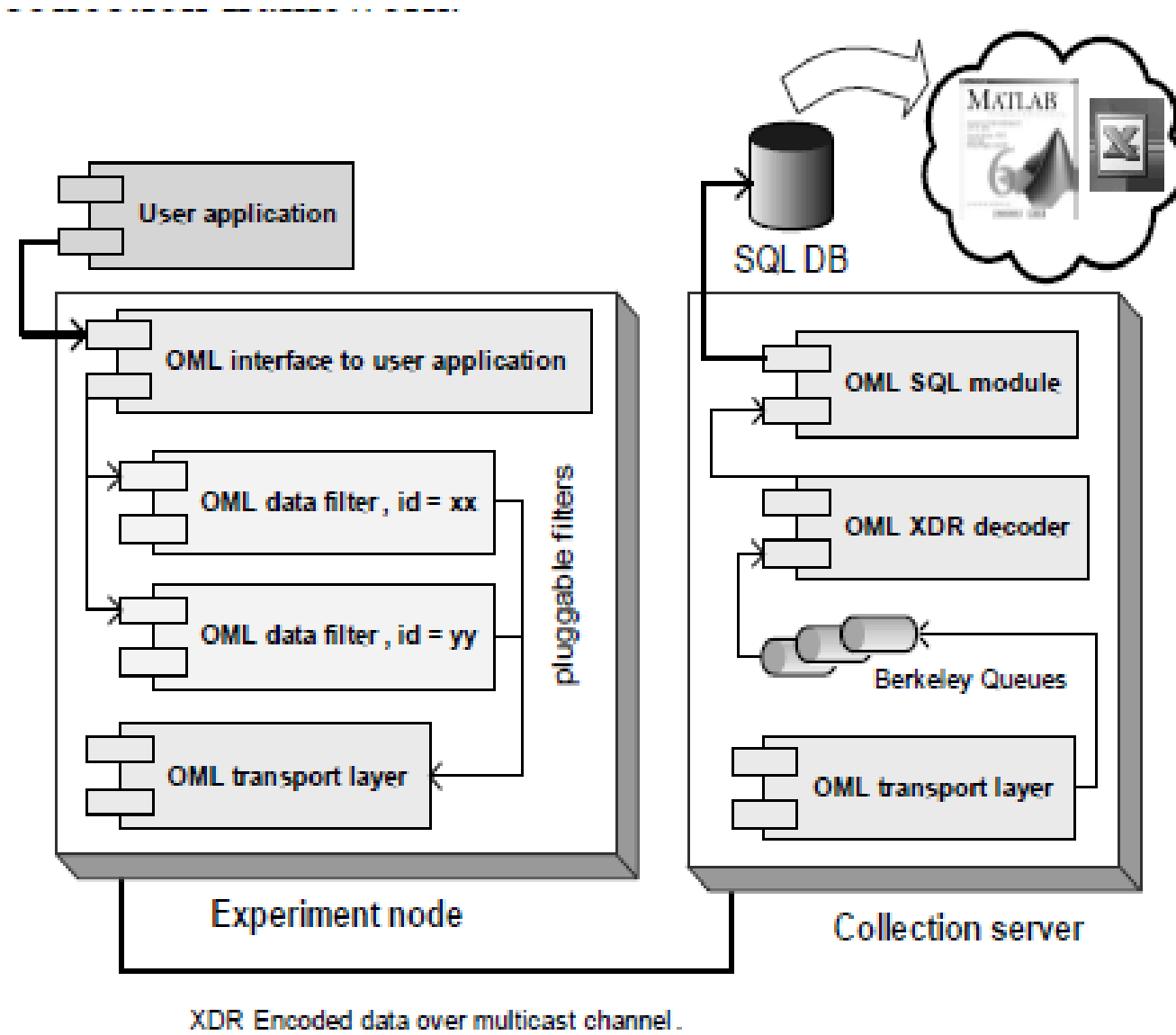
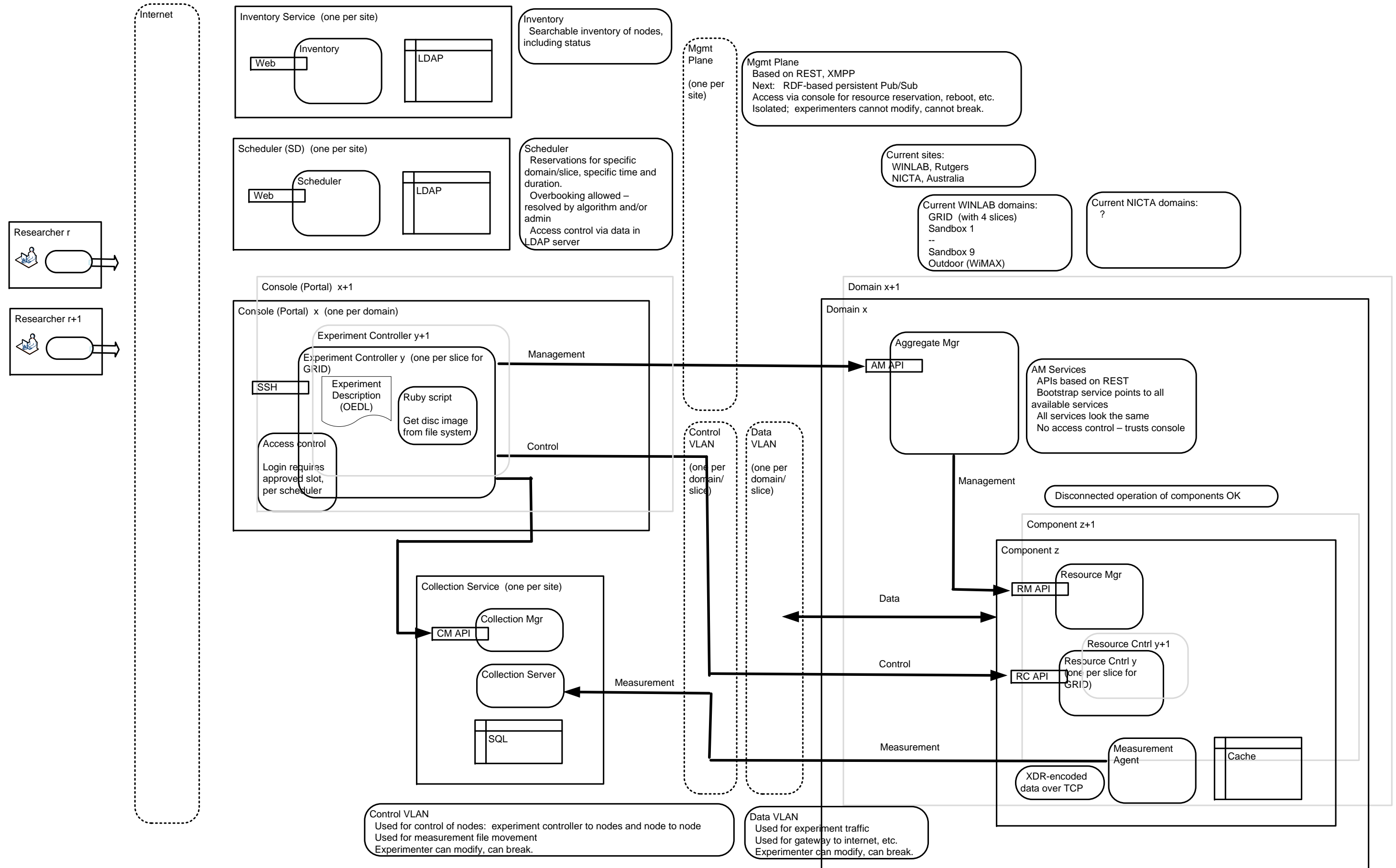
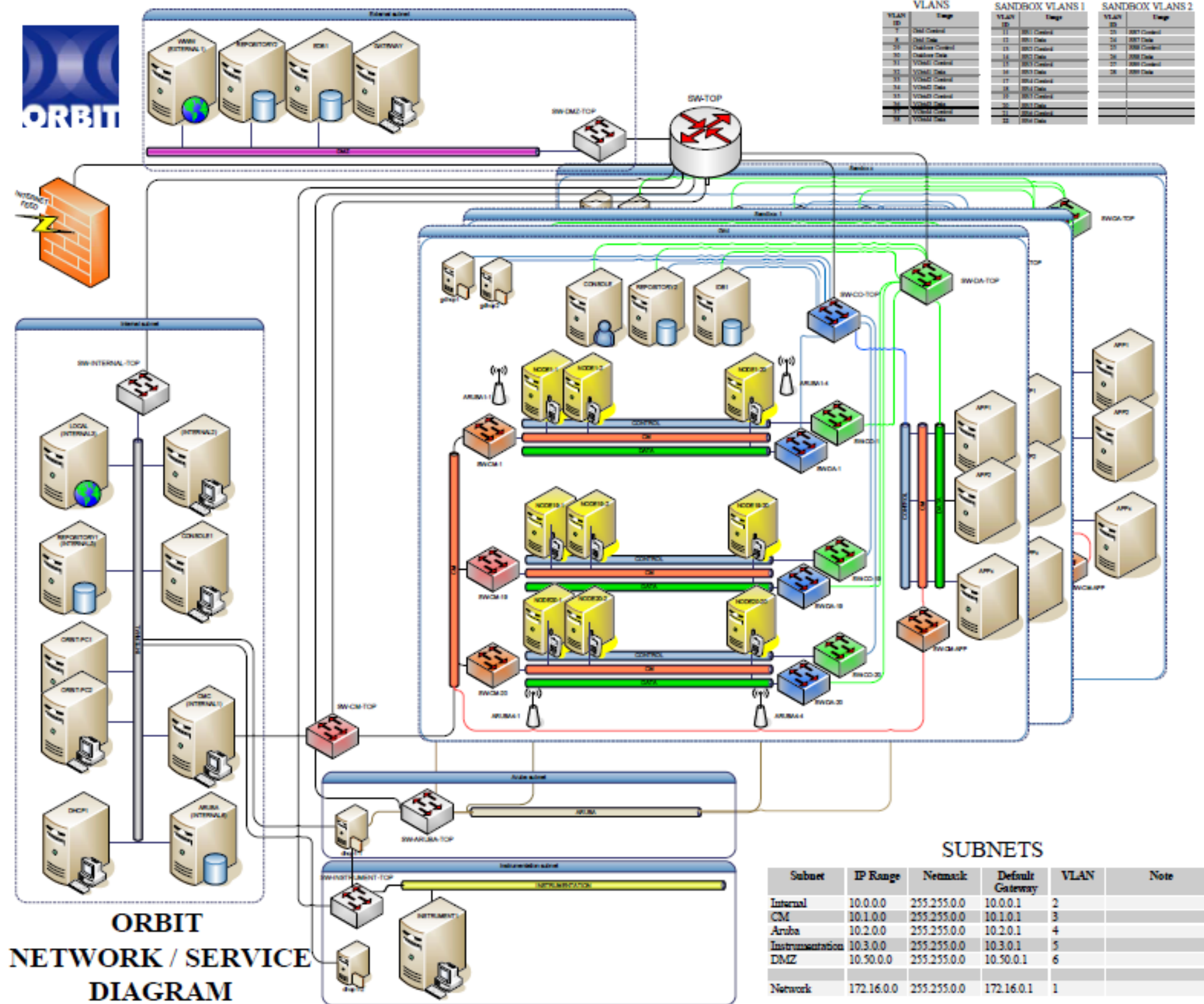


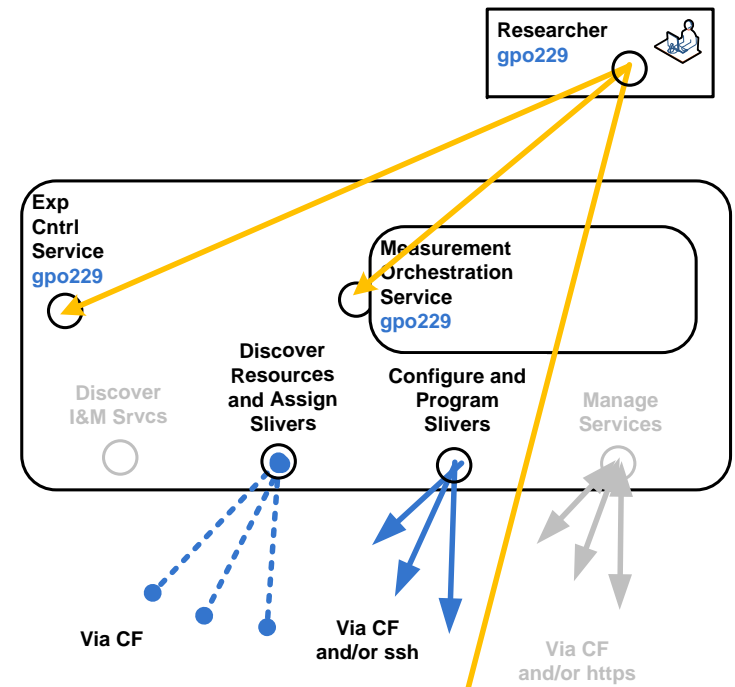
Figure 1. OML component architecture





Instrumentation Tools for Researchers (Students)

- * Each instance contained in one Aggregate (site)
- * Integrated with protoGENI CF
- * Meas Orch (MO) svc:
 - * Adds MO svc to protoGENI Exp Cntrl Svc that runs in Researcher's browser
 - * When Res specifies RSpec for experiment slice, it adds host for *Meas Controller (MC)* sliver
 - * Uses info from manifest to identify slivers to be monitored, and dynamically creates config files for SNMP daemon and other capture software
 - * Then, it downloads software for *MC* sliver, and it adds *MP* svc to each *Experiment Sliver*



Manage Svcs

- * *MC* Svc has collection control software
- * *MP* Svc includes remote access daemon to execute capture software
- * How?

MP Svc is automatically loaded into each Experiment Sliver

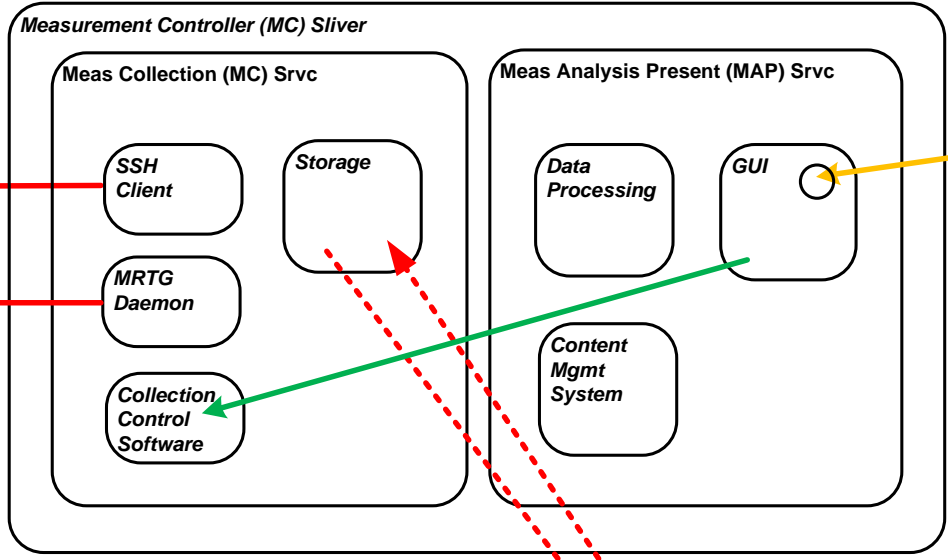
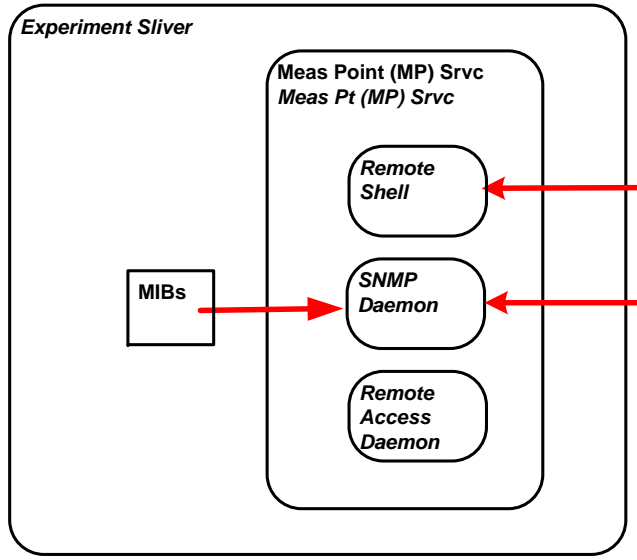
MC Sliver is automatically loaded into each Experiment Slice

Authorization

- * Emulab (ssh) key distribution mechanism used to authorize MC to get data from MPs

Security Concern

- * SNMP walk of MIBs



Portal to MCs

- * How does this work?

Software	Information	Display Type
SNMPd	Routing Table	Table
	IP Traffic	Graph
	ICMP Traffic	Graph
	TCP Traffic	Graph
	UDP Traffic	Graph
	CPU Utilization	Graph
	Memory Utilization	Graph
	Total Network Traffic	Graph
	Link-specific Traffic	Graph
	Link-specific Unicast Traffic	Graph
ssh/arp	ARP cache	Table
ssh/netstat	TCP streams	Table
ssh/netstat	UDP listeners	Table
ssh/ps	Process list	Table
ssh/lsmod	Installed Kernel Modules	Table

MP Svc includes:

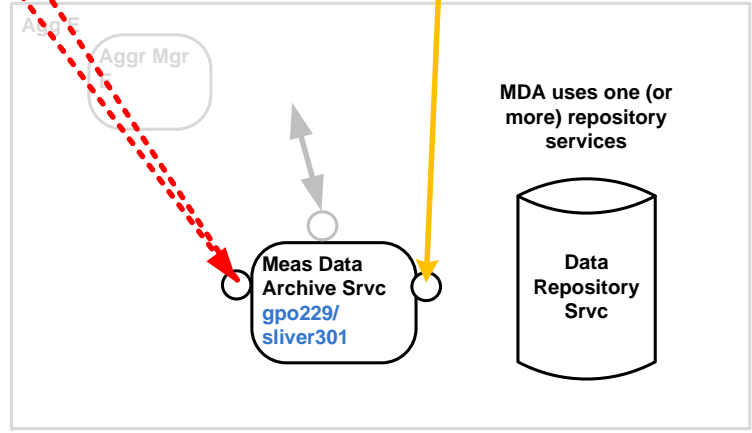
- * SNMP daemon, using existing MIBs, or added MIBs
- * tcpdump
- * netflow
- * custom monitoring code based on pcap library to collect packet stats not captured by SNMP daemon
- * ps, vmstat, and SNMP, to capture OS info, such as CPU load, memory load, routing table configs, ARP caches, loaded modules, etc.
- * remote access daemon to to execute capture software

Meas Data Schema

- * none defined
- * metadata not defined

Meas Data Arch Svc

- * svc not yet defined
- * messages not yet defined



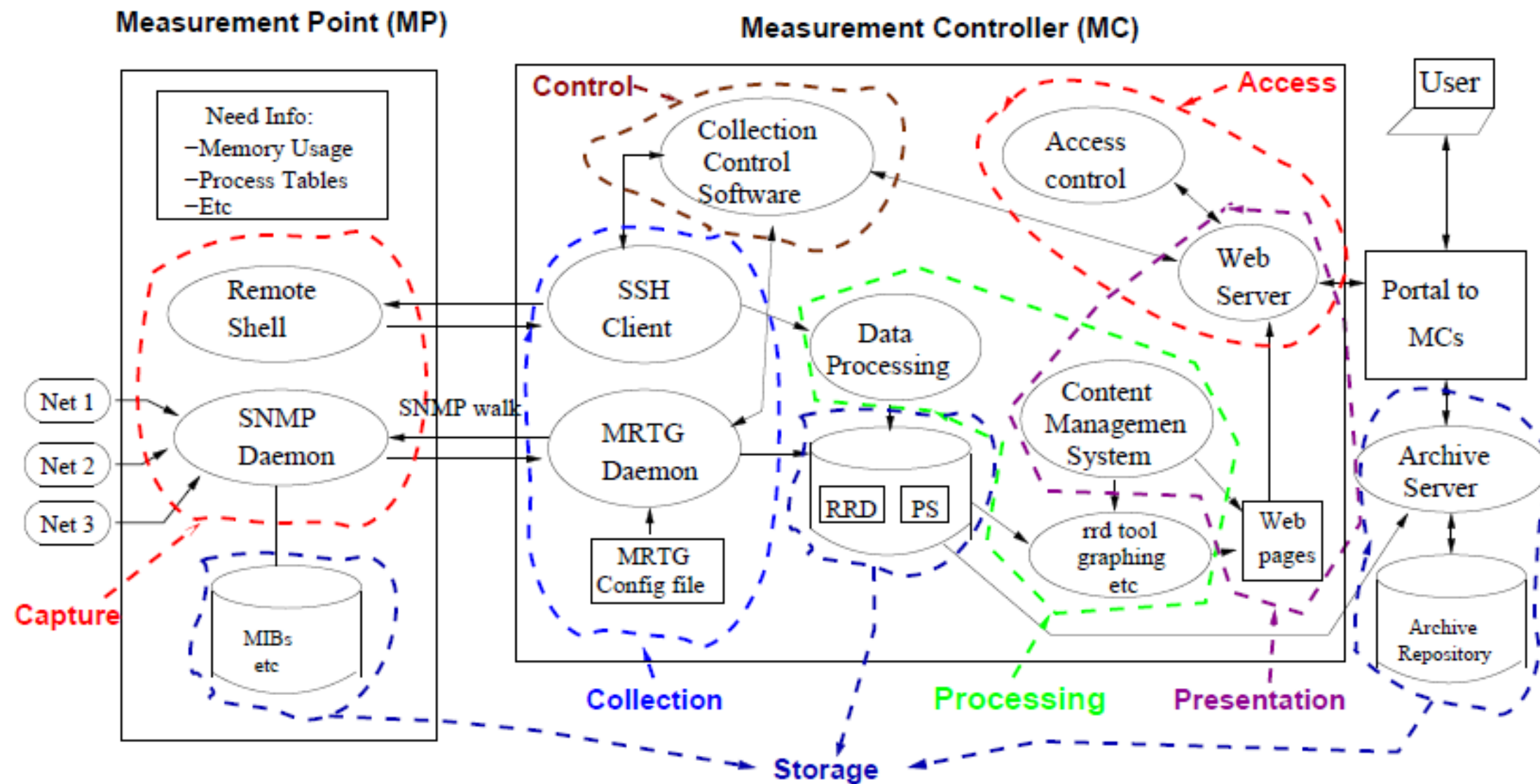


Figure 3: The Architectural Components of the INSTOOLS Toolset

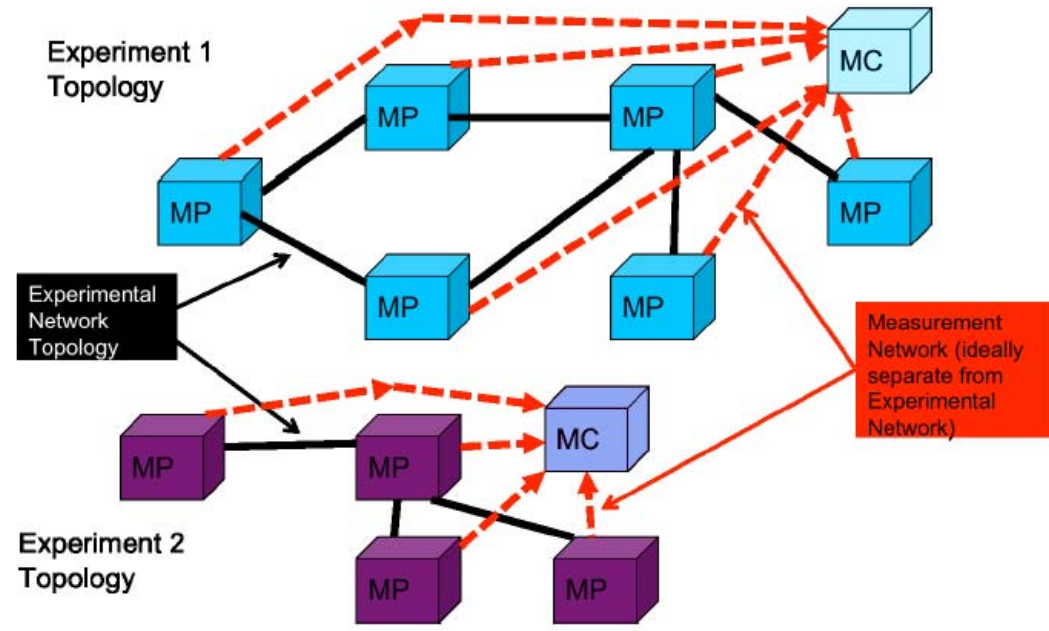


Figure 1: Each experiment/slice has its own MC and instrumentation and measurement network.

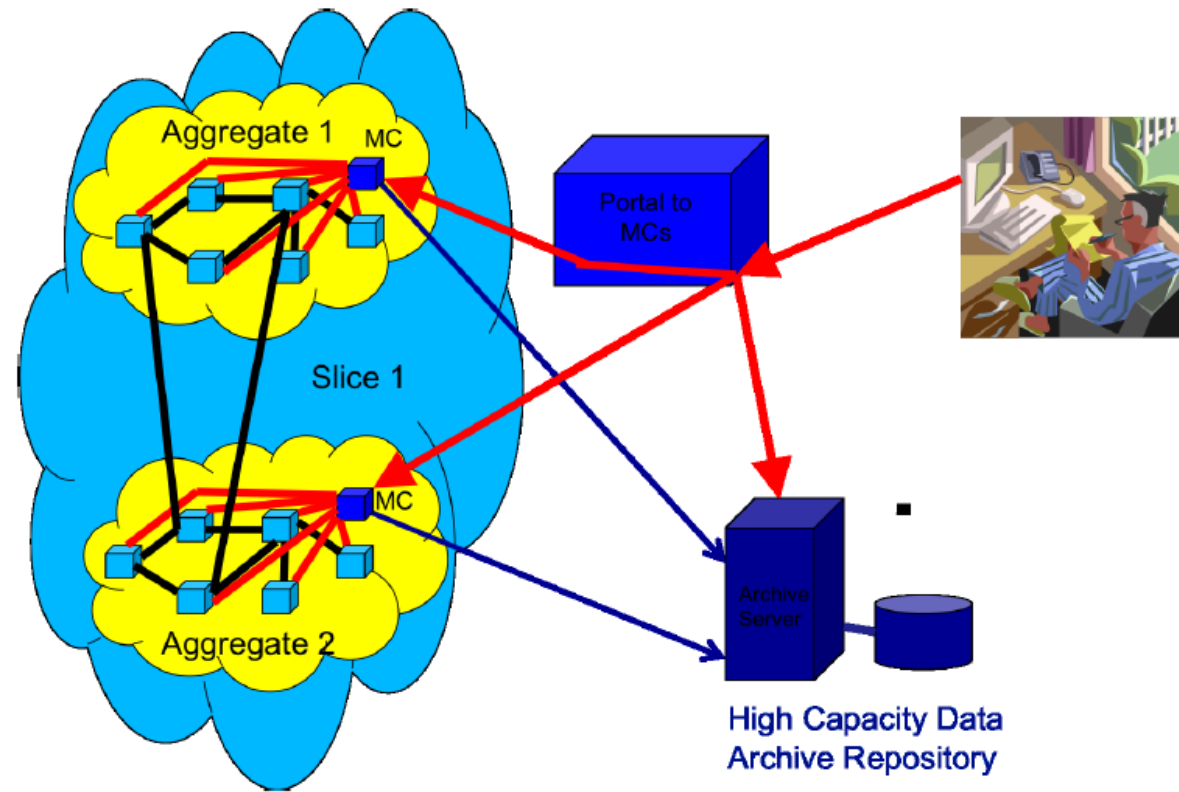
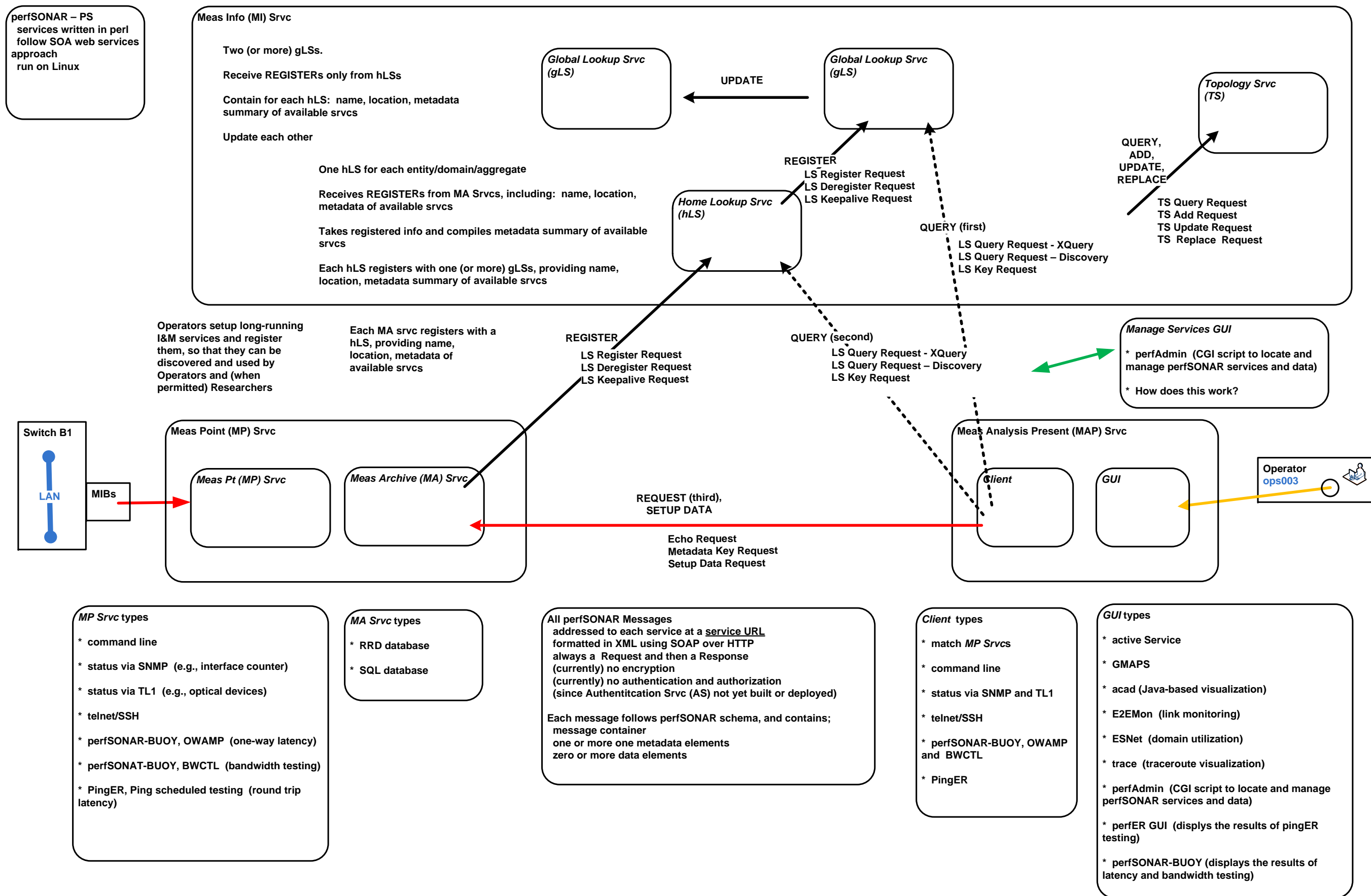


Figure 2: The MC Portal provides users with a single point of entry to their measurement data.



All perfSONAR Messages addressed to each service at a service URL formatted in XML using SOAP over HTTP always a Request and then a Response (currently) no encryption (currently) no authentication and authorization (since Authentication Srvc (AS) not yet built or deployed)

Each message follows perfSONAR schema, and contains; message container one or more one metadata elements zero or more data elements

Scalable Framework for Representation and Exchange of Network Measurements

Jason Zurawski, Martin Swany
Department of Computer and Information Sciences
University of Delaware, Newark, DE 19716
{zurawski, swany}@cis.udel.edu

Dan Gunter
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
dkgunter@lbl.gov

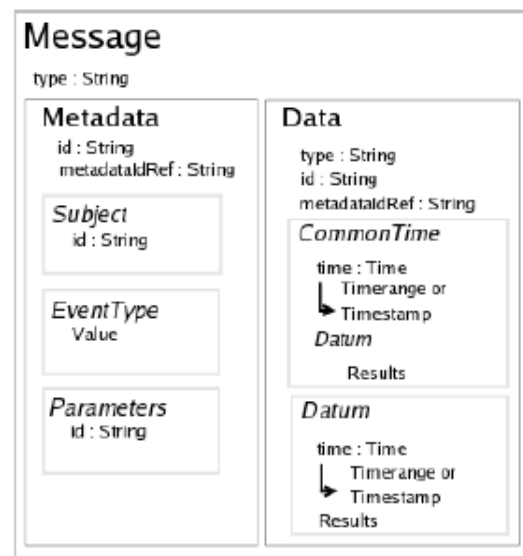
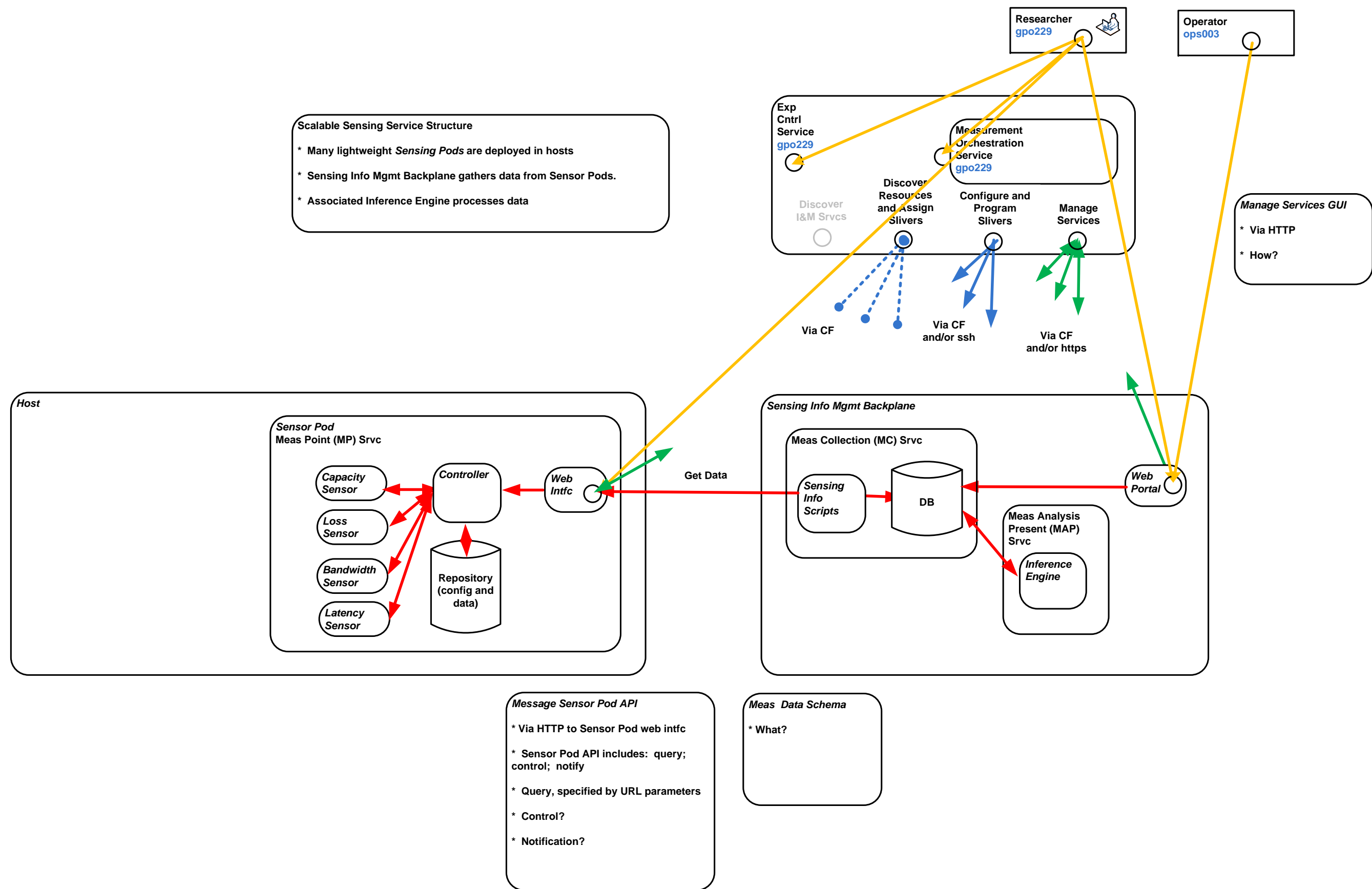
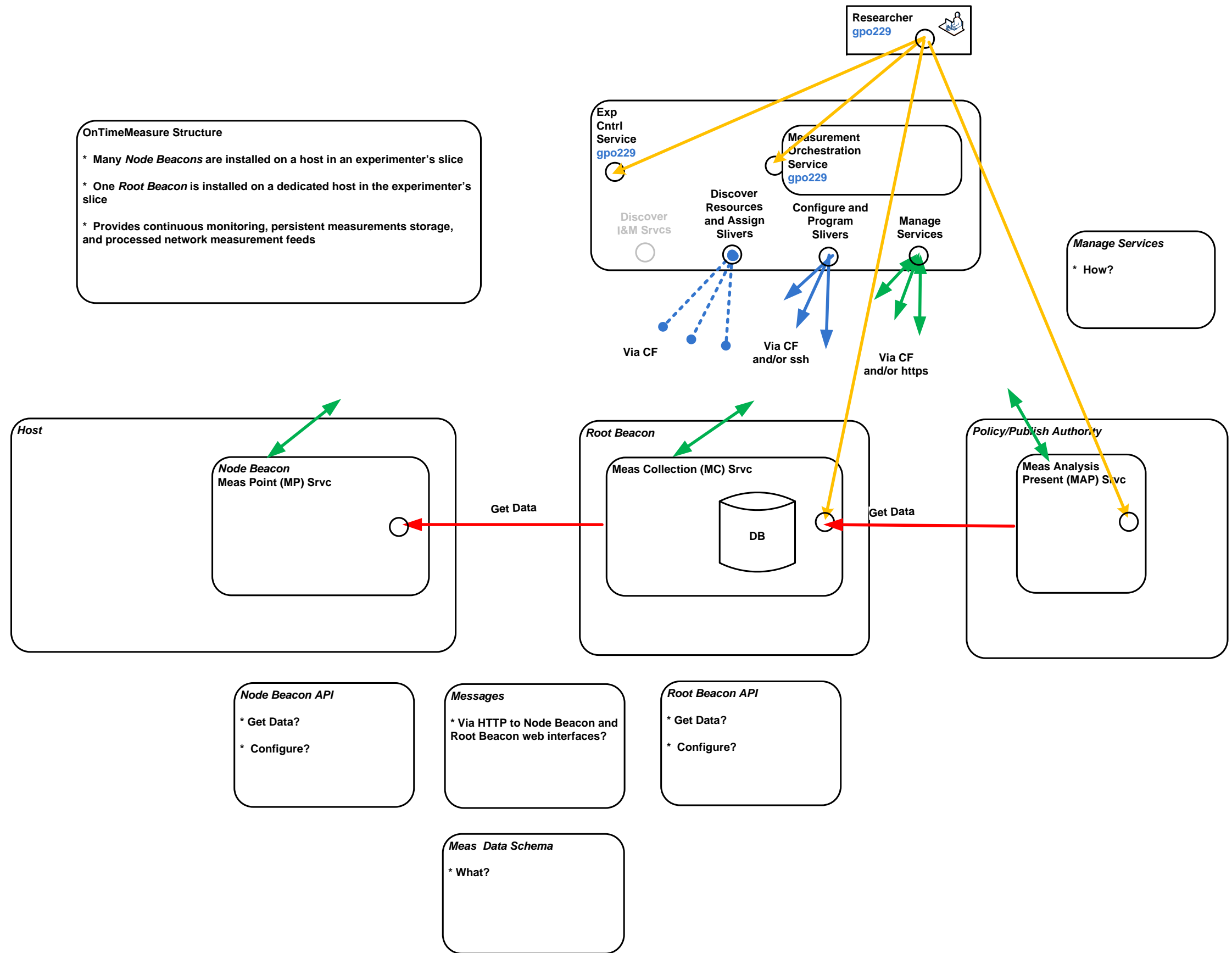


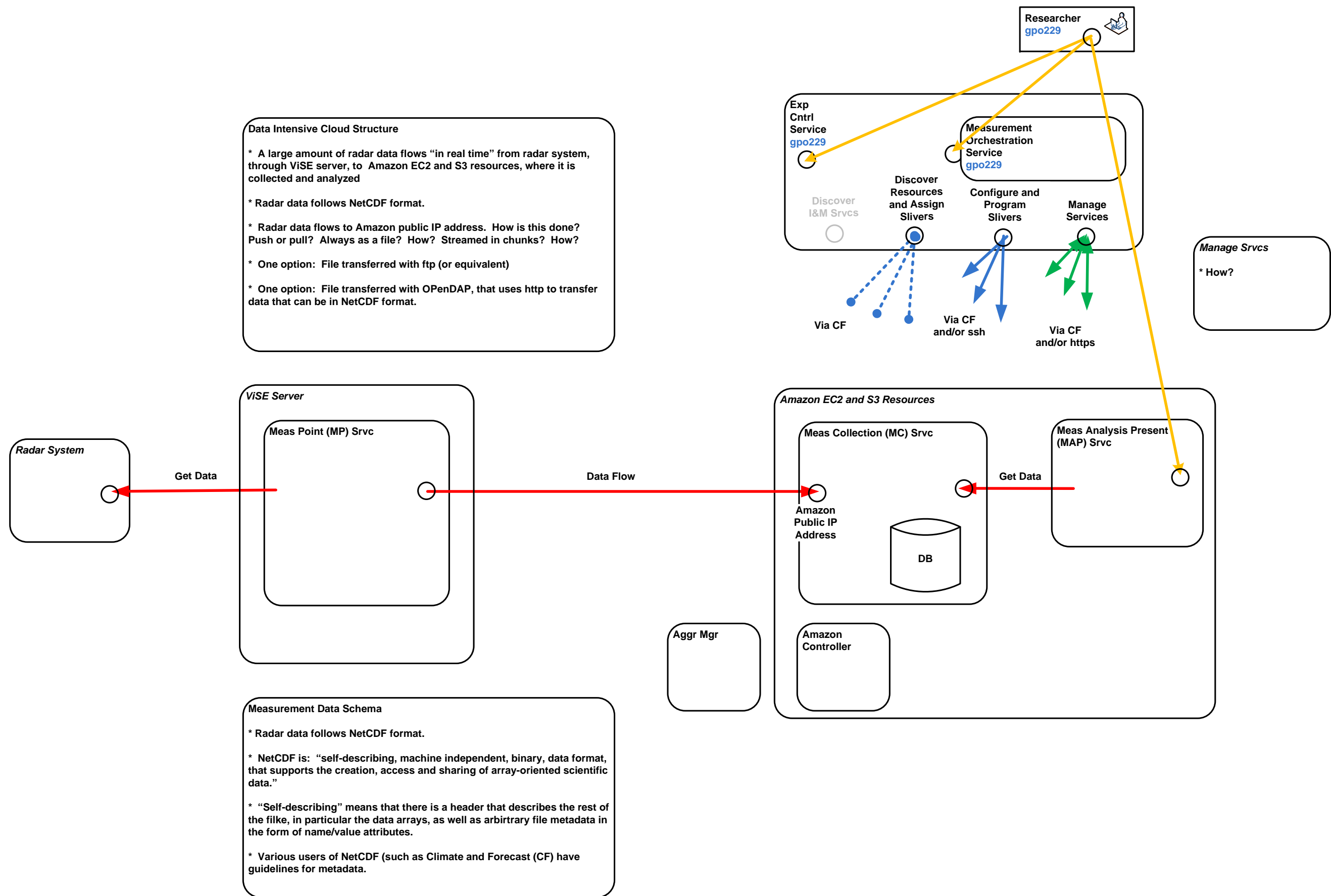
Figure 1. NM-WG Base Schema

The metadata section is subdivided into three parts, only the first of which is required:

- Subject – The physical or logical entity being described. For example, a host pair or router address. Like the subject of the sentence: *Host A to Host B measured ICMP latency is 100ms.*
- EventType – The canonical name of the aspect of the subject being measured, or the actual event (i.e. “characteristic”) being sought. Like the object of the sentence: *Host A to Host B measured ICMP latency is 100ms.*
- Parameters – The way in which the description is being gathered or performed. For example, command-line arguments to *traceroute* or whether the round-trip delay packet used ICMP or UDP. Like the descriptive clause of the sentence: *When you use 100 byte packets, Host A to Host B ICMP latency is 100ms.*







DOR (Digital Object Registry)

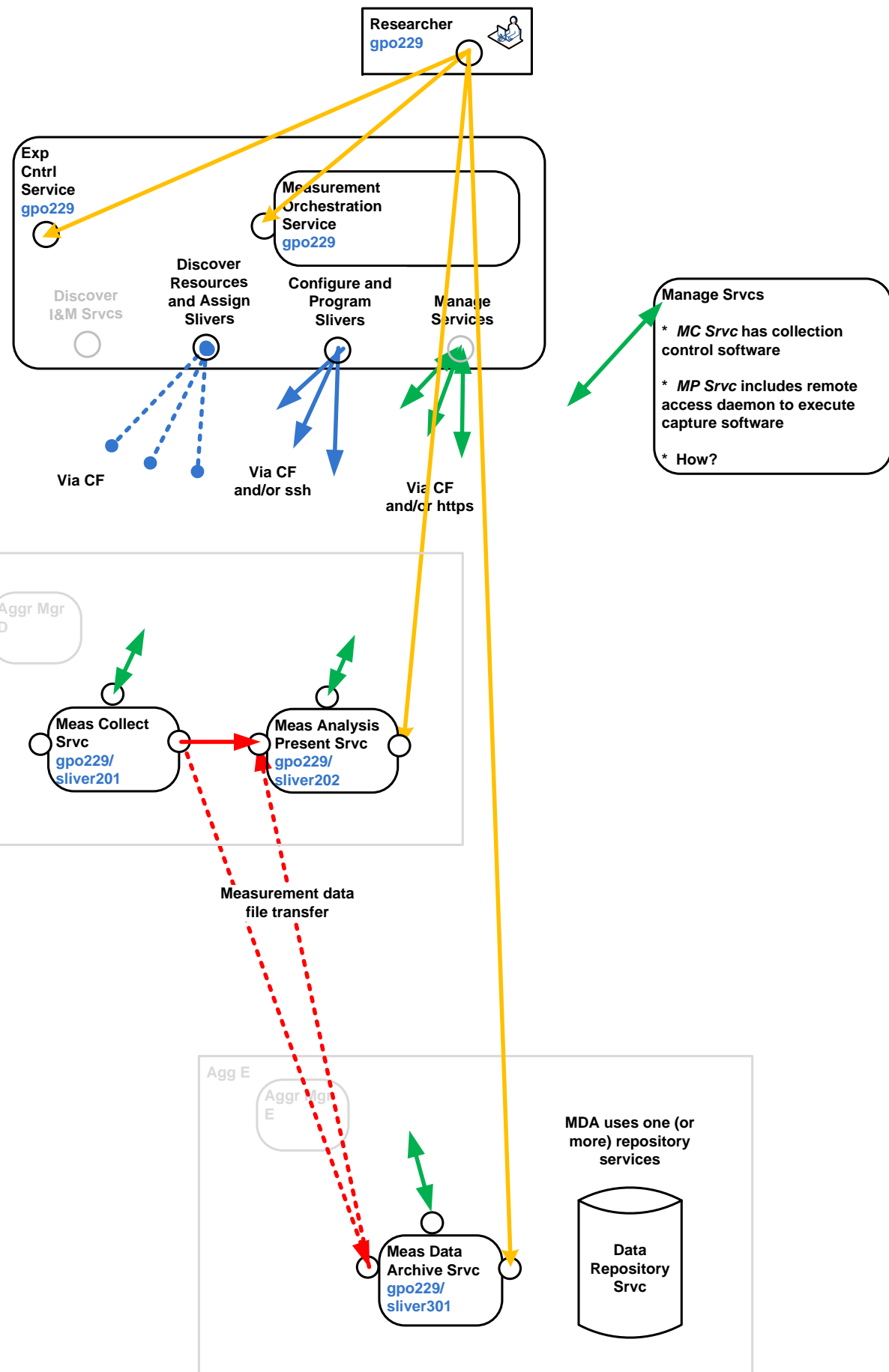
- * Provides Meas Data Arch Svc (and additional archive services)
- * Allows Operator/Researcher to archive Meas Data Files, and retrieve them
- * From another I&M svc, MDA svc can provide these basic functions: put/update file; get file; delete file
- * Interfaces to the MDA svc include: https; also ftp?; also sftp?
- * When file is first introduced, it is assumed that file contains type info (extension), metadata, and "file self description" info. A wide range of files and associated metadata is permitted by the MDA svc.
- * When file is first introduced, MDA svc assigns a unique identifier, which is a handle, that is attached to the file, and can be used to retrieve the file
- * When file is first introduced, or at any later time, the MDA svc provides a convenient service to allow Operator/Researcher to add identification metadata so that files are stored in an organized fashion, and can be found by searches/queries.
- * MDA svc allows Operator/Researcher to view organized files, and find them by searches/queries. Multiple views are supported.
- * Each file is "owned" by a GENI slice and one or more users (operators/researchers)
- * The MDA svc allows the owner to specify who has read and/or write access to the file.
- * The MDA svc utilizes the mechanisms provided by the CF to authenticate and authorize users.

MDA Svc protocol and API

- * Interfaces to the MDA svc include: https; scp or sftp
 - * From another I&M svc, MDA svc can provide these basic functions: put/update file; get file; delete file
 - * When file is first introduced, it is assumed that file contains type info (extension), metadata, and "file self description" info. A wide range of files and associated metadata is permitted by the MDA svc.
 - * When file is first introduced, MDA svc assigns a unique identifier, which is a handle, that is attached to the file, and can be used to retrieve the file
 - * When file is first introduced, or at any later time, the MDA svc provides a convenient service to allow Operator/Researcher to add identification metadata so that files are stored in an organized fashion, and can be found by searches/queries.
 - * MDA svc allows Operator/Researcher to view organized files, and find them by searches/queries. Multiple views are supported.
- Assume: API as specified by CNRI for ADL (Advanced Distributed Learning) svc (CNRI)

MDA Svc authentication and authorization

- * Each file is "owned" by a GENI slice and one or more users (operators/researchers)
 - * The MDA svc allows the owner to specify who has read and/or write access to the file.
 - * The MDA svc utilizes the mechanisms provided by the CF to authenticate and authorize users.
- Assume: CF drops public keys of authorized users into MDA svc, so that: presence of key indicates an "account" on the MDA svc; additional info indicates nature of access (CNRI)



MDA Srvc file organization, views and queries

* When file is first introduced, or at any later time, the MDA srvc provides a convenient service to allow Operator/Researcher to add identification metadata so that files are stored in an organized fashion, and can be found by searches/queries.

* MDA srvc allows Operator/Researcher to view organized files, and find them by searches/queries. Multiple views are supported.

Assume: Basic view follows current Emulab tree-like file structure, from ops.emulab.net server (Evan)

```

/proj          <--- This folder contains all the sub-folders for each project
../proj_A
../proj_B
.
.
.
../PNI         <--- a real project name
...../deltas
...../exp      <--- This folder contains all the sub-folders for each experiment
...../exp_A
...../exp_B
.
.
.
...../test1    <--- a real experiment name
...../archive
...../bin
...../datastore
...../logs
...../swapinfo
...../tbdata   <--- important folder contains all kinds of experiment data and logs
...../activity.log
...../assign.log
...../environment
...../eventkey
...../linktest
...../lmap
...../lmap.gz
...../ltpmap
...../ltpmap.gz
...../nsfile.ns
...../startexp.log
...../swapexp.log
...../test1.ns
...../test1.png
...../topomap
...../topomap.gz
...../webkey
...../tftpboot
...../tmp
...../groups
...../images
...../logs
...../rpms
...../tarfiles
...../templates
...../tiplogs

/groups        <--- Each project has a default group with the same name of the project. And each project also has other groups of users(with different
privileges) associated with it
../PNI -> /proj/PNI

/usr/testbed   <--- This folder contains the testbed software

/users        <--- This folder contains user accounts just like regular /home folder in Linux system
../chyz198    <--- This is my account registered in Utah Emulab, all my personal keys are here

/share        <--- This folder used to keep the data intend to be shared by all users

```