



August 9th 2011, OSG Site Admin Workshop
Jason Zurawski – Internet2 Research Liaison

perfSONAR

Agenda

- Tutorial Agenda:
 - Network Performance Primer - Why Should We Care? (**30 Mins**)
 - Introduction to Measurement Tools (**20 Mins**)
 - Use of NTP for network measurements (**15 Mins**)
 - Use of the BWCTL Server and Client (**25 Mins**)
 - Use of the OWAMP Server and Client (**25 Mins**)
 - Use of the NDT Server and Client (**25 Mins**)
 - perfSONAR Topics (**30 Mins**)
 - Diagnostics vs Regular Monitoring (**20 Mins**)
 - Use Cases (**30 Mins**)
 - Exercises

Motivation – Possible Solutions

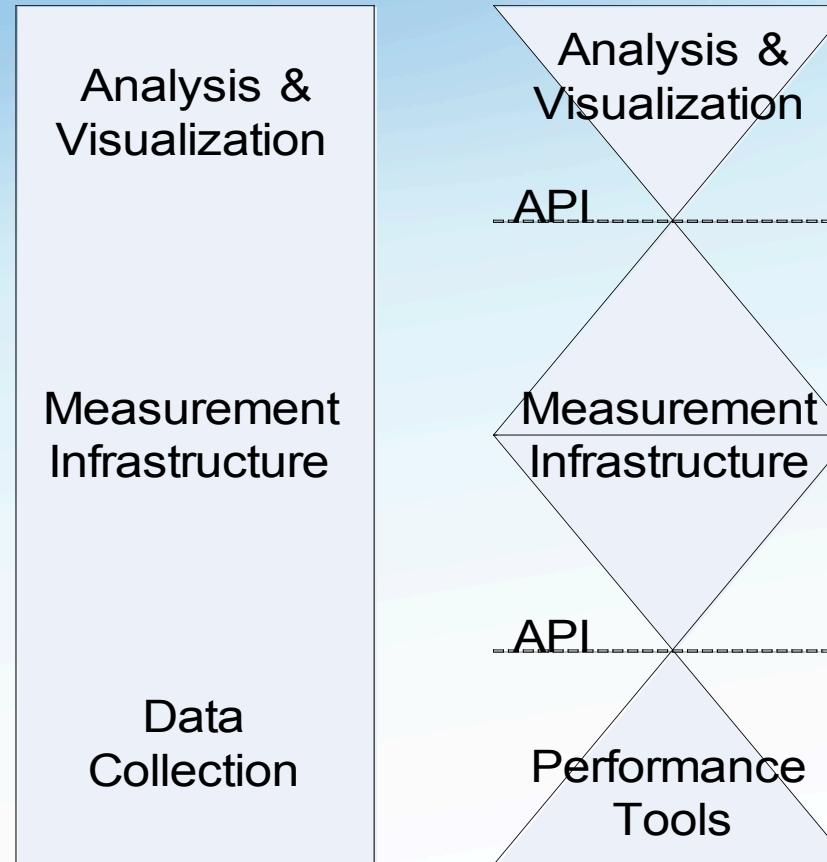
- Finding a solution to network performance problems can be broken into two distinct steps:
 - Use of *Diagnostic Tools* to locate problems
 - Tools that actively measure performance (e.g. Latency, Available Bandwidth)
 - Tools that passively observe performance (e.g. error counters)
 - *Regular Monitoring* to establish performance baselines and alert when expectation drops.
 - Using diagnostic tools in a structured manner
 - Visualizations and alarms to analyze the collected data
- Incorporation of either of these techniques must be:
 - *ubiquitous*, e.g. the solution works best when it is available everywhere
 - seamless (e.g. *federated*) in presenting information from different resources and domains



Motivation – Possible Solutions

- Desirable design features for any solution
 - Component Based
 - Functionality should be split into logical units
 - Each function (e.g. visualization) should function through well defined communication with other components (e.g. data storage)
 - Modular
 - Monolithic designs rarely work
 - Components allow choice of how to operate a customized end solution.
 - Accessible
 - Well defined interfaces (e.g. APIs)
- Initial design should facilitate future expansion

Motivation – Possible Solutions



What is perfSONAR?

- Most organizations perform monitoring and diagnostics of their own network
 - SNMP Monitoring via common tools (e.g. [MRTG](#), [Cacti](#))
 - Enterprise monitoring (e.g. [Nagios](#))
- Networking is increasingly a cross-domain effort
 - International collaborations in many spaces (e.g. science, the arts and humanities) are common
 - Interest in development and use of R&E networks at an all time high
- Monitoring and diagnostics ***must*** become a cross-domain effort
 - Complete view of all paths
 - Eliminate “who to contact” and “what to ask for” - 24/7 availability of diagnostic observations

What is perfSONAR?

- A collaboration
 - Production network operators focused on designing and building tools that they will deploy and use on their networks to provide monitoring and diagnostic capabilities to themselves and their user communities.
- An architecture & set of communication protocols
 - Web Services (WS) Architecture
 - Protocols established in the Open Grid Forum
 - Network Measurement Working Group ([NM-WG](#))
 - Network Measurement Control Working Group ([NMC-WG](#))
 - Network Markup Language Working Group ([NML-WG](#))
- Several interoperable software implementations
 - [perfSONAR-MDM](#)
 - [perfSONAR-PS](#)
- A Deployed Measurement infrastructure

perfSONAR Inception

- *perfSONAR* originated from discussions between [Internet2](#)'s End-to-End Performance Initiative ([E2Epi](#)), and the [Géant2](#) project in September 2004.
- Members of the [OGF](#)'s (then GGF) NM-WG provided guidance on the encoding of network measurement data.
- Additional network partners, including [ESnet](#) and [RNP](#) provided development resources and served as early adopters.
- The first release of *perfSONAR* branded software was available in July 2006 (Java based software).
- All *perfSONAR* branded software is open source
- All products looking to be labeled as *perfSONAR compliant* must establish protocol compliance based on the public standards of the OGF

Who is perfSONAR?

- The *perfSONAR* Consortium is a joint collaboration between
 - ESnet
 - Géant
 - Internet2
 - Rede Nacional de Ensino e Pesquisa (RNP)
- Decisions regarding protocol development, software branding, and interoperability are handled at this organization level
- There are at least two independent efforts to develop software frameworks that are *perfSONAR* compatible.
 - perfSONAR-MDM
 - perfSONAR-PS
 - Others? The beauty of open source software is we will never know the full extent!
- Each project works on an individual development roadmap and works with the consortium to further protocol development and insure compatibility

Who is perfSONAR-MDM?

- [perfSONAR-MDM](#) is made up of participants in the Géant project:
 - Arnes
 - Belnet
 - Carnet
 - Cesnet
 - CYNet
 - DANTE
 - DFN
 - FCCN
 - GRNet
 - GARR
 - ISTF
 - PSNC
 - Nordunet (Uninett)
 - Renater
 - RedIRIS
 - Surfnet
 - SWITCH
- perfSONAR-MDM is written in Java primarily and was designed to serve as the monitoring solution for the Large Hadron Collider (LHC) project.
- perfSONAR-MDM is available as Deb (Debian Compatible) and RPM (Red Hat Compatible) packages.

Who is perfSONAR-PS?

- perfSONAR-PS is comprised of several members:
 - ESnet
 - Fermilab
 - Georgia Tech
 - Indiana University
 - Internet2
 - SLAC
 - The University of Delaware
- perfSONAR-PS products are written in the perl programming language and are available for installation via source or RPM (Red Hat Compatible) packages
- perfSONAR-PS is also a major component of the pS Performance Toolkit – A bootable Linux CD containing measurement tools.

perfSONAR Adoption

- *perfSONAR* is gaining traction as an interoperable and extensible monitoring solution
- Adoption has progressed in the following areas:
 - R&E networks including backbone, regional, and exchange points
 - Universities on a national and international basis
 - Federal labs and agencies in the United States (e.g. *JET* nets)
 - Scientific Virtual Organizations, notably the LHC project
- Recent interest has also accrued from:
 - International R&E network partners and exchange points
 - Commercial Providers in the United States
 - Hardware manufacturers

perfSONAR Architecture Overview

- Interoperable network measurement middleware designed as a Service Oriented Architecture (SOA):
 - Each component is modular
 - All are Web Services (WS) based
 - The global *perfSONAR* framework as well as individual deployments are decentralized
 - All *perfSONAR* tools are Locally controlled
 - All *perfSONAR* tools are capable of federating locally and globally
- *perfSONAR* Integrates:
 - Network measurement tools and archives (e.g. stored measurement results)
 - Data manipulation
 - Information Services
 - Discovery
 - Topology
 - Authentication and authorization

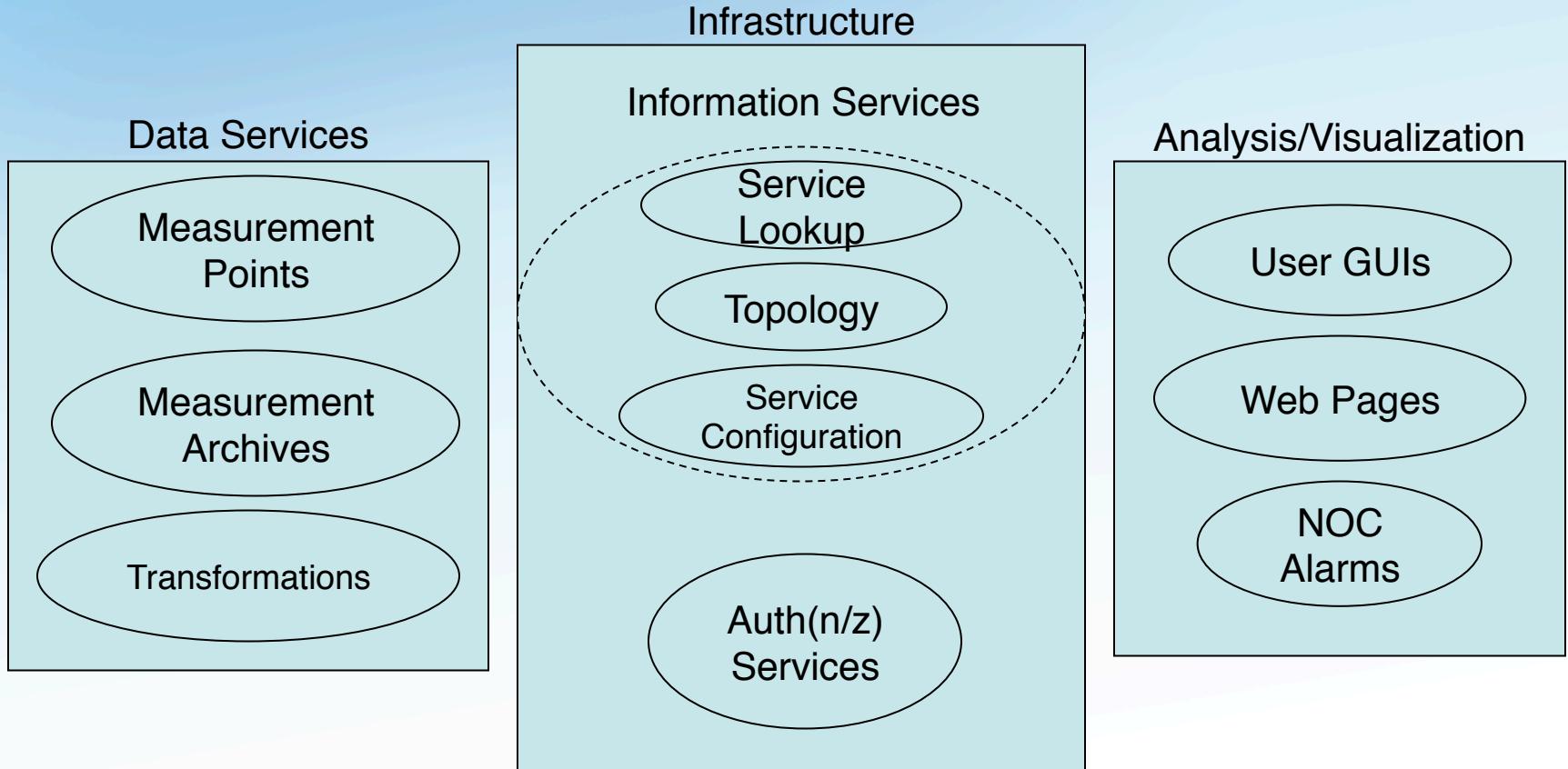


perfSONAR Architecture Overview

- The key concept of *perfSONAR* is that each entity (e.g. “services”) performs a function
 - Each service provides a limited set of functionality, e.g. collecting measurements between arbitrary points or managing the registration and location of distributed services
 - The service is a ***self contained*** and provides functionality on its own as well as when deployed with the remainder of the framework
- Services interact through exchanges
 - Standardized message formats
 - Standardized exchange patterns (e.g. a communication protocol)
- A collection of *perfSONAR* services within a domain is a ***deployment***
 - Deploying *perfSONAR* can be done *À la carte*, or through a complete solution
- Services federate with each other, locally and globally
 - Services are designed to automatically discover the presence of other *perfSONAR* components
 - Clients are designed with this distributed paradigm in mind



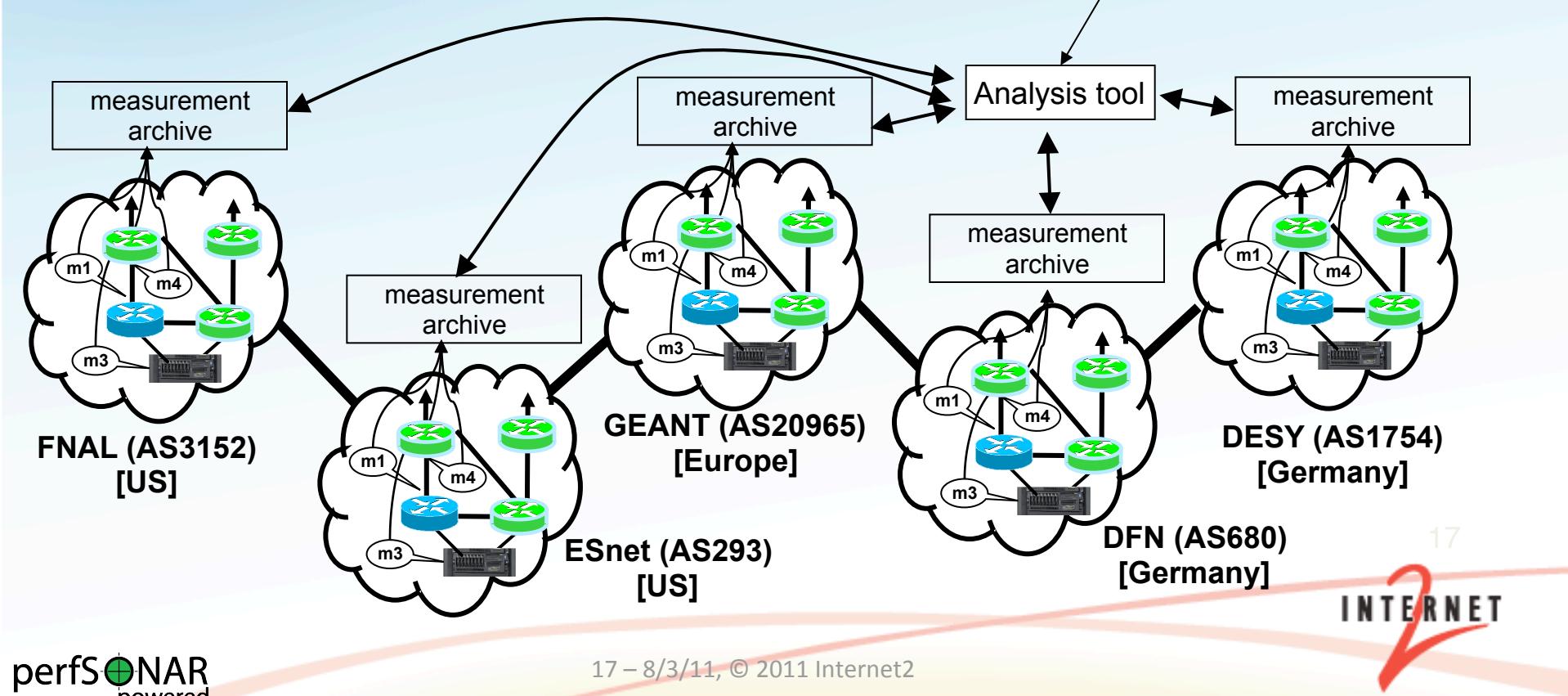
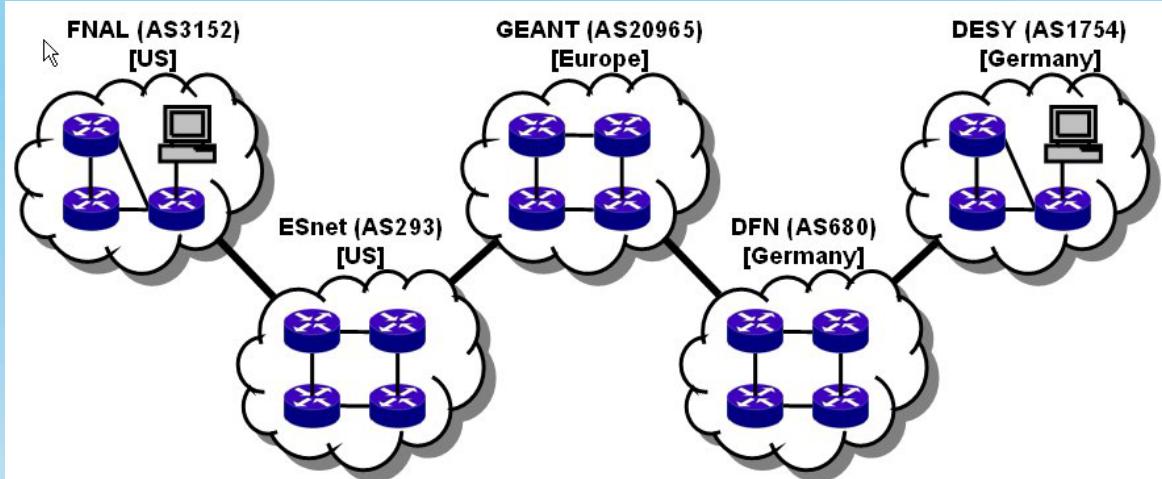
perfSONAR Architecture Overview



perfSONAR Architecture Overview

- A *perfSONAR* deployment can be any combination of services
 - An instance of the *Lookup Service* is required to share information
 - Any combination of data services and analysis and visualization tools is possible
- *perfSONAR* services have the ability to federate globally
 - The *Lookup Service* communicates with a confederated group of directory services (e.g. the *Global Lookup Service*)
 - Global discovery is possible through APIs
- *perfSONAR* is most effective ***when all paths are monitored***
 - Debugging network performance must be done *end-to-end*
 - Lack of information for specific domains can delay or hinder the debug process

Many collaborations are inherently multi-domain, so for an end-to-end monitoring tool to work everyone must participate in the monitoring infrastructure



Service Oriented Architecture (SOA)

- Measurement Point (MP) Service
 - Enables the initiation of performance tests
- Measurement Archive (MA) Service
 - Stores and publishes performance monitoring results
- Transformation Service
 - Transform the data (aggregation, concatenation, correlation, translation, etc)
- Resource protector
 - Arbitrate the consumption of limited resources
 - Other services delegate a limited portion of the authorization decision here

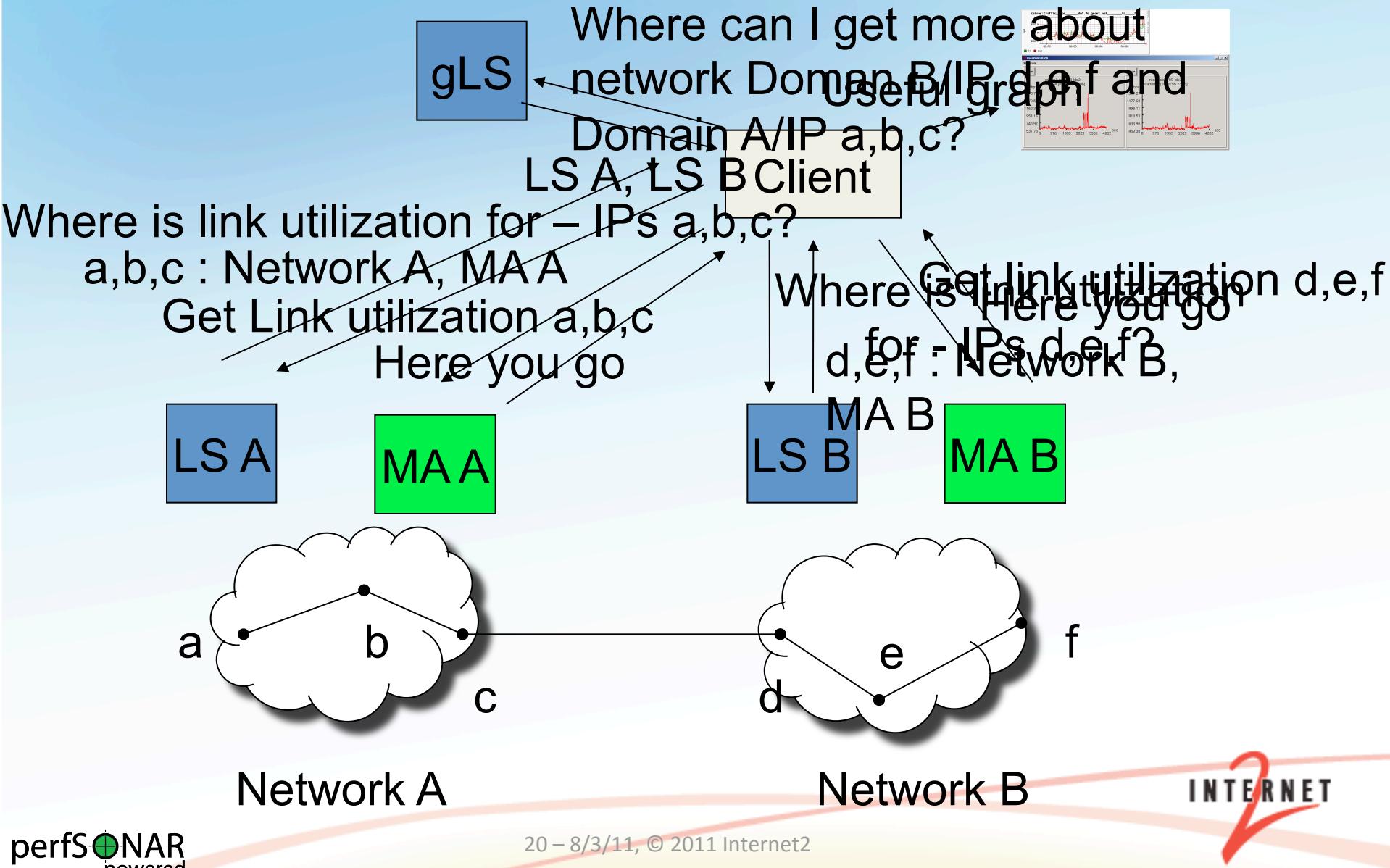
These services are specifically concerned with the job of network performance measurement and analysis

Service Oriented Architecture (SOA)

- **Lookup Service**
 - Allows the client to discover the existing services and other LS services.
 - Dynamic: services register themselves to the LS and mention their capabilities, they can also leave or be removed if a service goes down.
- **Topology Service**
 - Make the network topology information available to the framework.
 - Find the closest MP, provide topology information for visualisation tools
- **Authentication Service**
 - Authentication & Authorization functionality for the framework
 - Users can have several roles, the authorization is done based on the user role.
 - Trust relationship between networks

These services are the infrastructure concerned with discovering federating the available network services

Service Oriented Architecture (SOA)



Architecture Overview - MP

- Measurement Point (MP) form the *lowest layer* of the monitoring infrastructure
 - Directly interacts with the measurement tool
 - Can offer WS control over on-demand measurement
 - Can offer interface to a regular scheduled measurements
- Roles of the Measurement Point:
 - Utilize well known tools to perform measurements
 - Offer, at a minimum, cache storage of recently performed measurements
 - Interact with Measurement Archives (MAs) to archive stored measurements
- Examples:
 - perfSONAR-BUOY (OWAMP and BWCTL Testing)
 - PingER (Ping Testing)
 - Command Line MP

Architecture Overview - MA

- Measurement Archive (MA) stores the results of network and performance measurements
 - WS interface for storage and query
 - Interacts with backend databases (e.g. SQL, RRD)
- Roles of the Measurement Archive:
 - Expose historical and current measurements of diverse types
 - Draw data queries away from the Measurement Points (MPs)
- Examples:
 - perfSONAR-BUOY (OWAMP and BWCTL Data)
 - PingER (Ping Data)
 - SNMP MA/RRD MA
 - Status MA

Architecture Overview - TrS

- The Transformation Service (TrS) performs operations on data sets (e.g. aggregation, correlation).
 - WS interface
 - Potential to store well known operations, and replay later
- Roles of the Transformation Service:
 - Draw complex statistical queries from Measurement Archives
 - Provide a conduit for popular operations (e.g. running statistics over several changing dataset).
- Examples (Planned):
 - Path diagnostics tools
 - Combining multiple metrics (network path, utilization, latency, bandwidth)
 - Data presentation
 - Statistical results for raw measurements.

Architecture Overview - RP

- The Resource Protector (RP) monitors the relative performance and availability of the monitoring infrastructure
 - Knowledge of the services in a given deployment
 - Defined policy regarding access and resources
- Roles of the Resource Protector:
 - Protects the time and resources of services from being overrun
 - Too many queries from a single source
 - Too much data for a given query
 - Cooperate with the Authentication and Authorization (AA) entities
- Examples (Panned):
 - Data Protection
 - Limits the size, duration, or frequency of a query
 - Service Protection
 - Limits access to functionality of the service



Architecture Overview - LS

- The Lookup Service (LS) is a general name for the service and data discovery infrastructure
 - Facilitates service and data discovery through the concept of registration
 - “Summarizes” and distributes the job of location across layers of lookup
 - Home Lookup Services – Local cache of data for several services
 - Global Lookup Services – Works similar to DNS for locating information through general queries
- Roles of the Lookup Service :
 - Draws specific queries about the data and services away from the Measurement Points and Archives
 - Distribute information globally based on local conditions
 - Assure the ‘freshness’ of information in a dynamic infrastructure

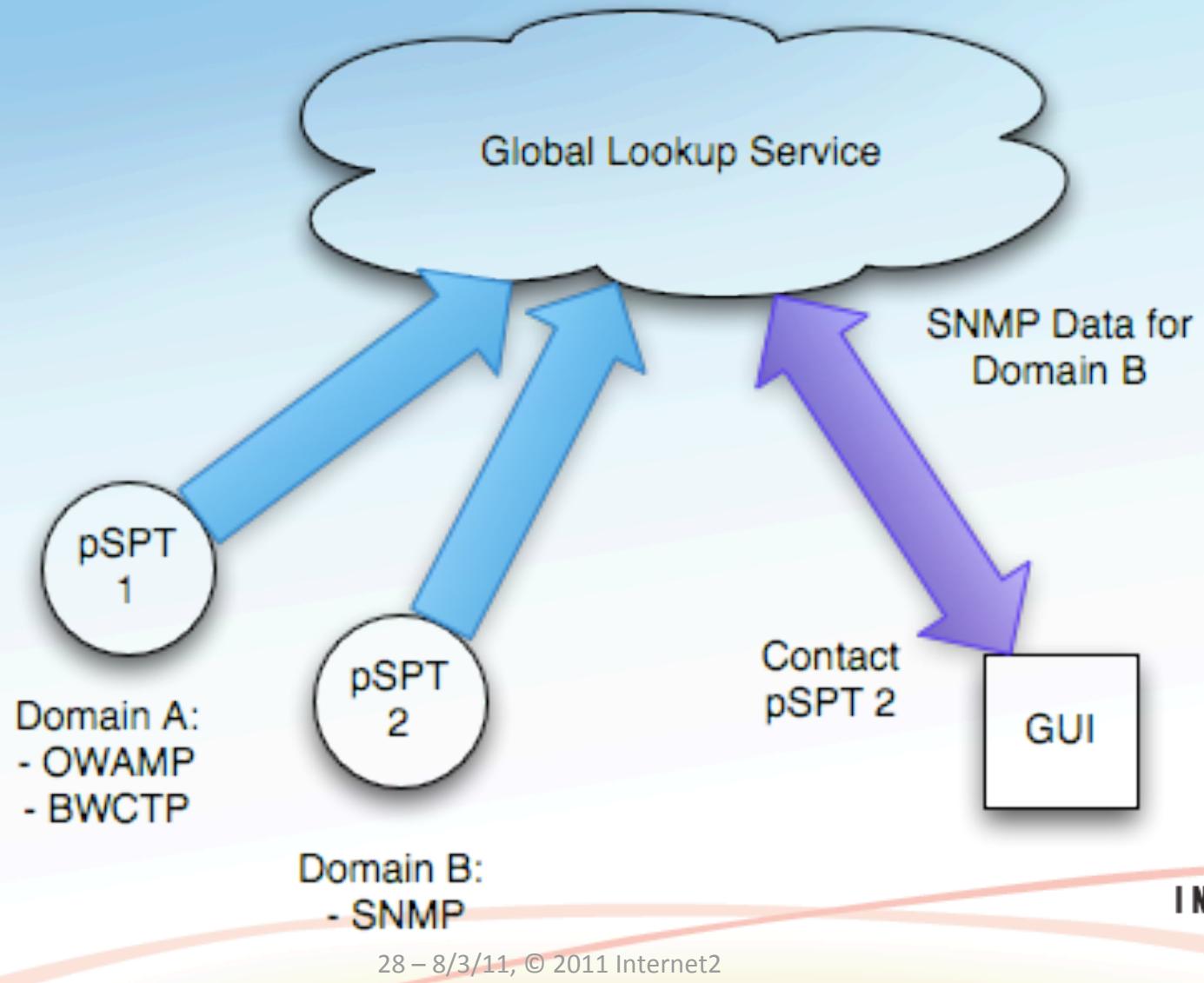
Architecture Overview - hLS

- The Home Lookup Service (hLS) interacts directly with the other portions of the perfSONAR framework
 - Recommended deployment is per domain
- Accepts *Registration* information from around the framework
 - E.g. An MA will register its name, location, and available *metadata*
 - Metadata = static portion of a measurement ('*subject*', not results)
- Responds to *Queries* about services and data
 - Services looking for each other (e.g. MP looking for an MA)
 - Client applications looking for data

Architecture Overview - gLS

- The Global Lookup Service (gLS) serves as the *oracle* of the perfSONAR framework
 - Global *cloud* of services cooperating together to distribute information
 - Manage the hLSs at the lower layer
- Accepts *Registration* information from hLSs **only**(!)
 - E.g. An hLS will register its name, location, and a *summary* of the services and data it contains
 - Summary = condensed list of domains, ip addresses, data types
- Responds to *Queries* about services and data
 - Similar to hLS queries, but more focused on *where* instead of *what*
 - Answer is typically an hLS to contact, not a direct result

Architecture Overview – Lookup Service



Architecture Overview - TS

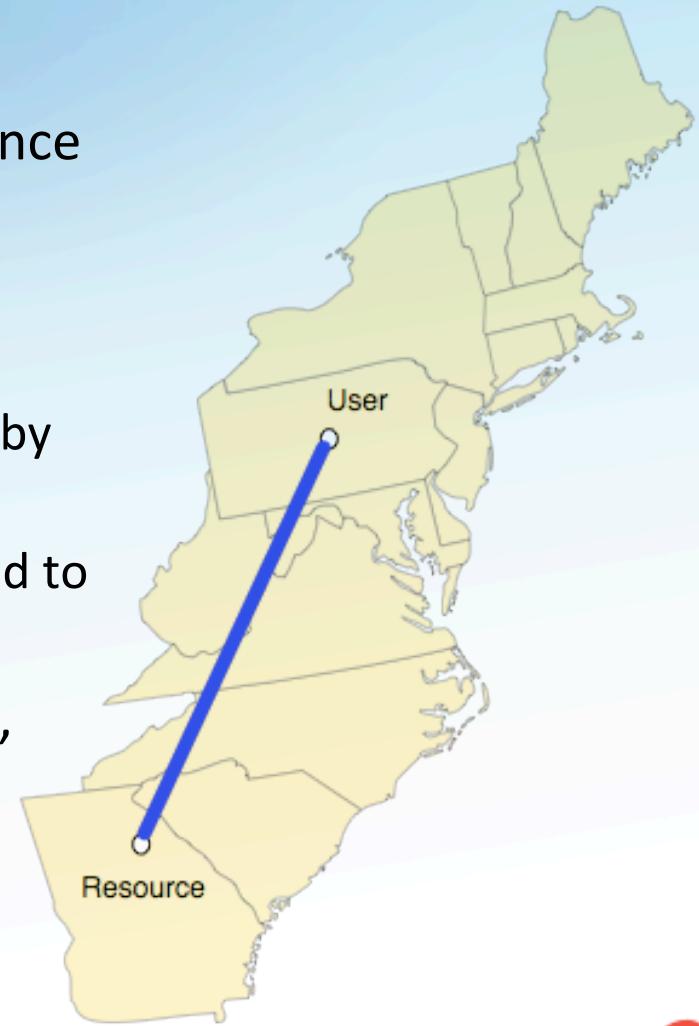
- The Topology Service (TS) gathers and stores network topology information similar to the Lookup Service (LS)
 - Interfaces with external network tools (Dynamic Circuits, NOC databases)
 - Provides a query interface
- Roles of the Transformation Service :
 - Gather network topology from various sources
 - Correlate information found in other TS sources to provide a complete view of network availability
 - Interface with measurement tools to associate measurements with specific portions of the infrastructure

Architecture Overview - AS

- The Authentication and Authorization (AS) service serves as a front end for identity management.
 - Identity management relies on assigning roles to a given user via *attributes*, e.g. permission to do something
 - The AS will communicate via WS with a client and pass along *credentials* in order to validate an action or task
 - The AS will protect access to services and data
- Rolls of the Authentication Service :
 - Validate services and clients given credentials
 - Act on behalf of the users to acquire the necessary permissions

Example perfSONAR Use Case

- perfSONAR should be used to diagnose an end-to-end performance problem
 - User is attempting to download a remote resource
 - Resource and user are separated by distance
 - Both are assumed to be connected to high speed networks
- Operation does not go as planned, where to start?



INTERNET
2

Example perfSONAR Use Case

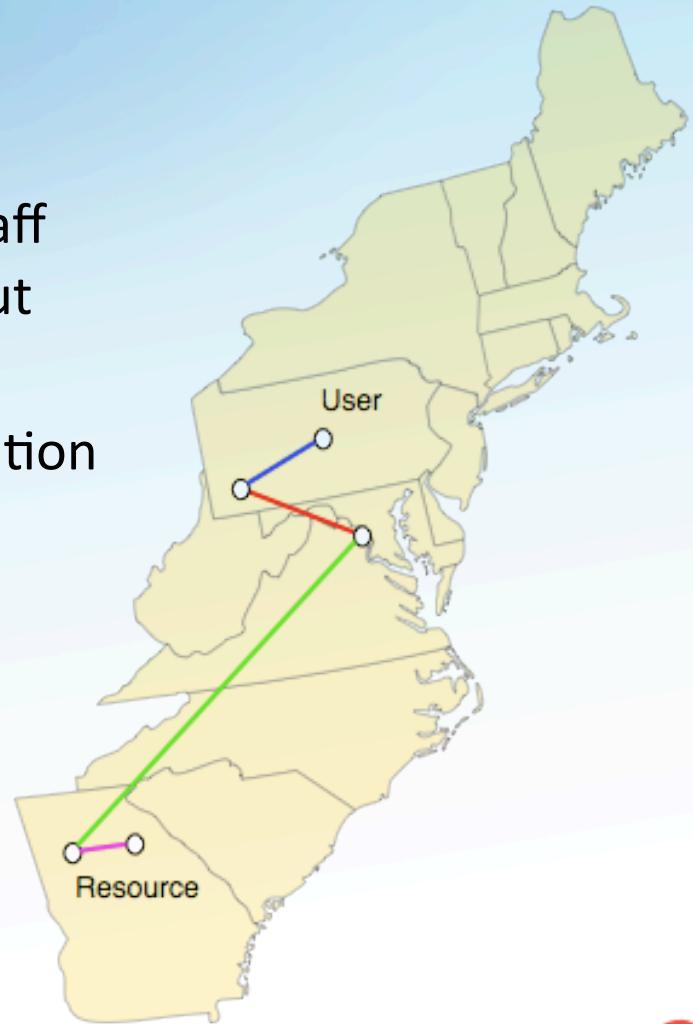
- Simple tools like ***traceroute*** can be used to determine the path traveled
- There could be a performance problem anywhere in here
- The problem may be something we could fix, but the chances are greater that it is not



INTERNET
2

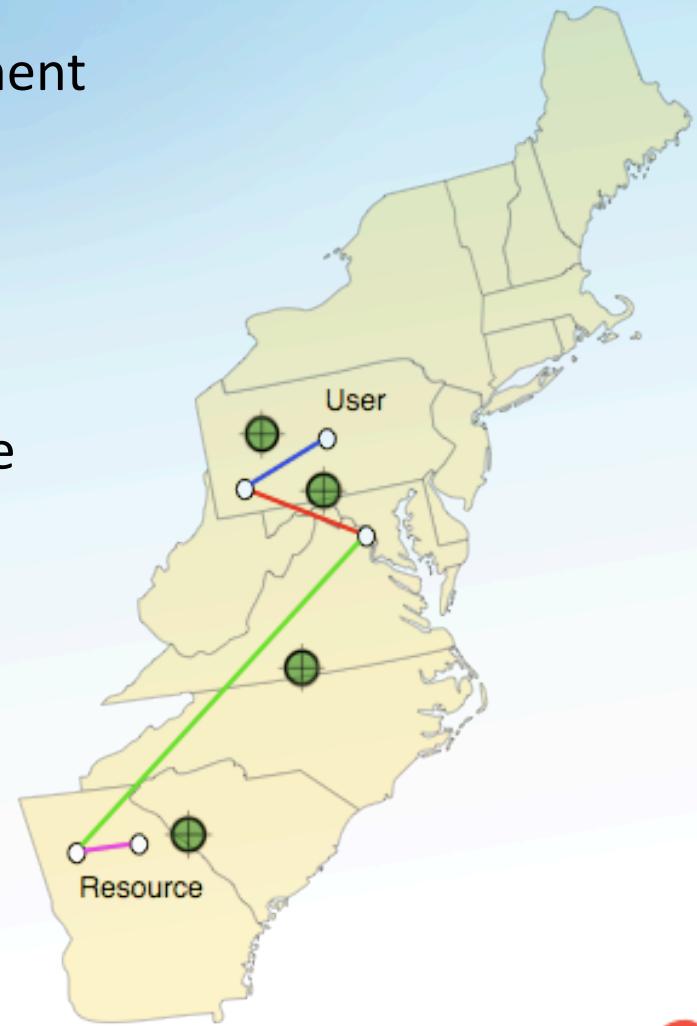
Example perfSONAR Use Case

- Each segment of the path is controlled by a different domain.
- Each domain will have network staff that could help fix the problem, but how to contact them?
- All we really want is some information regarding performance



Example perfSONAR Use Case

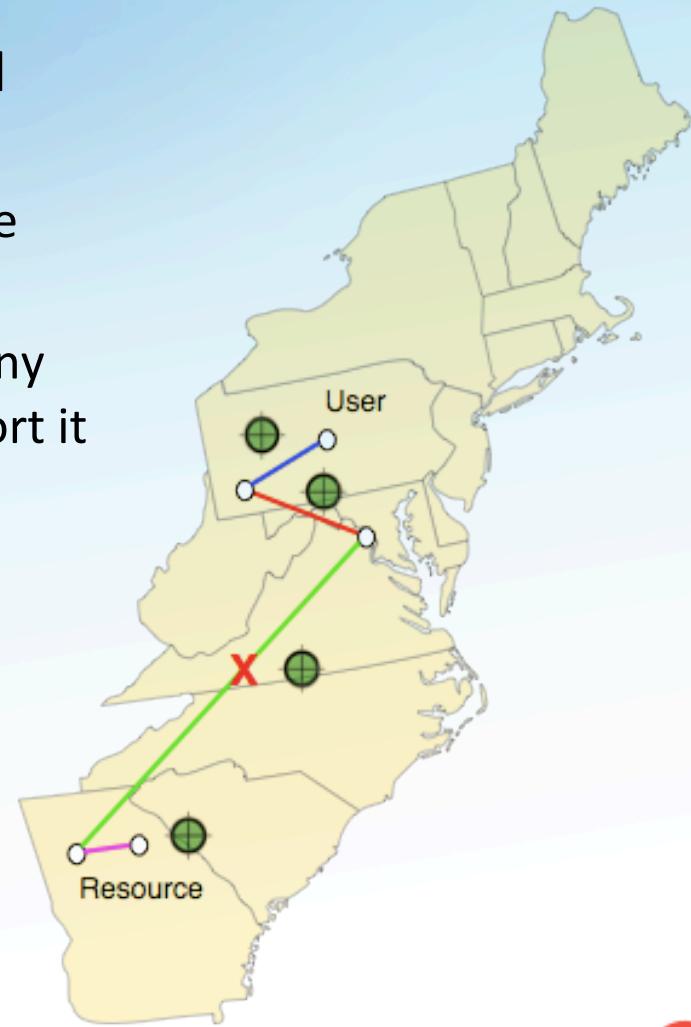
- Each domain has made measurement data available via perfSONAR
- The user was able to discover this automatically
- Automated tools such as visualizations and analyzers can be powered by this network data



INTERNET
2

Example perfSONAR Use Case

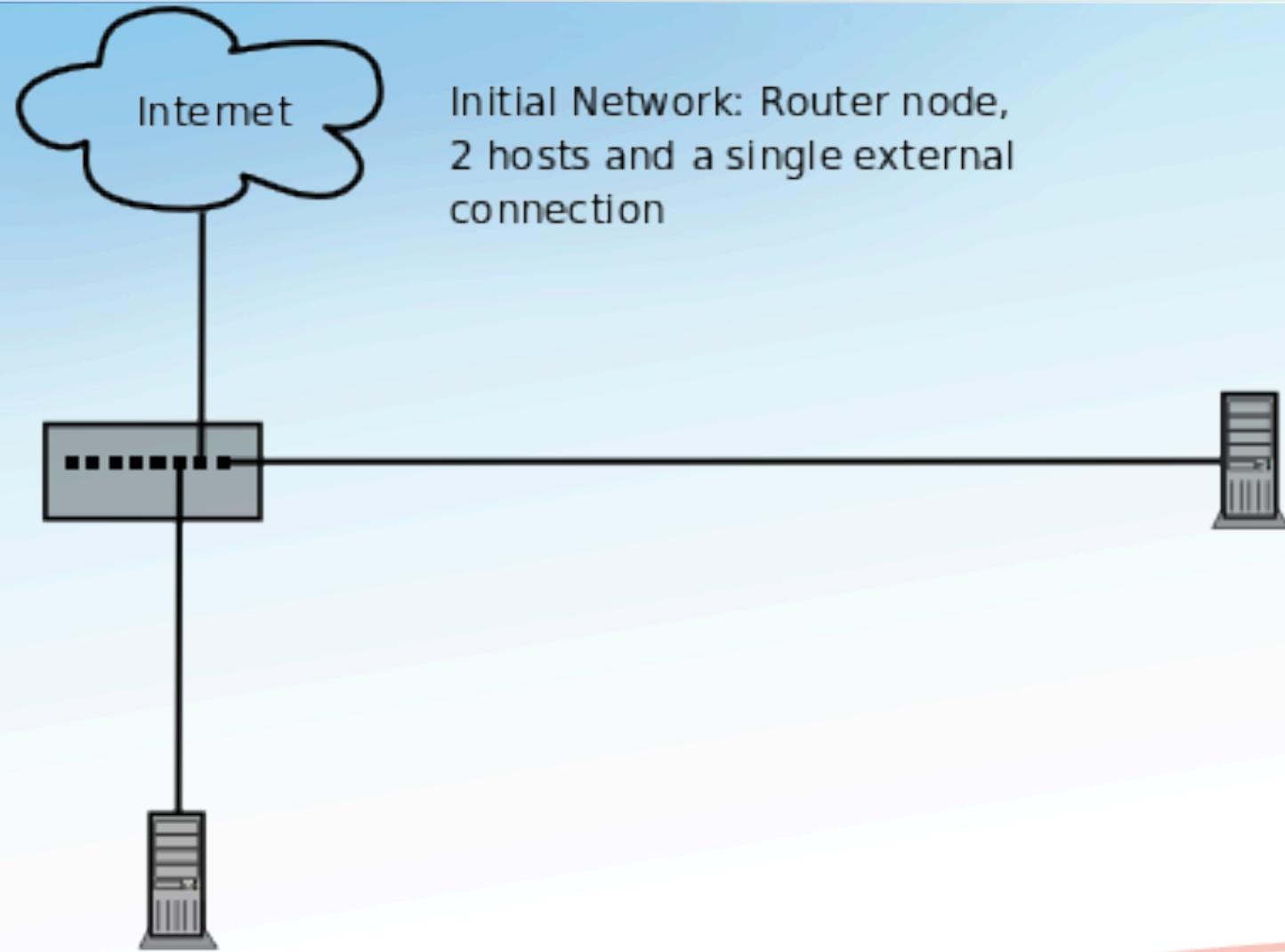
- In the end, the problem is isolated based on testing.
 - May have gone unnoticed in some cases (e.g. a “soft failure”)
 - Could have been observed by many others ... that didn’t think to report it
- The user (or operations staff) can contact the domain in question to inquire about this performance problem
- When fixed the transfer should progress as intended



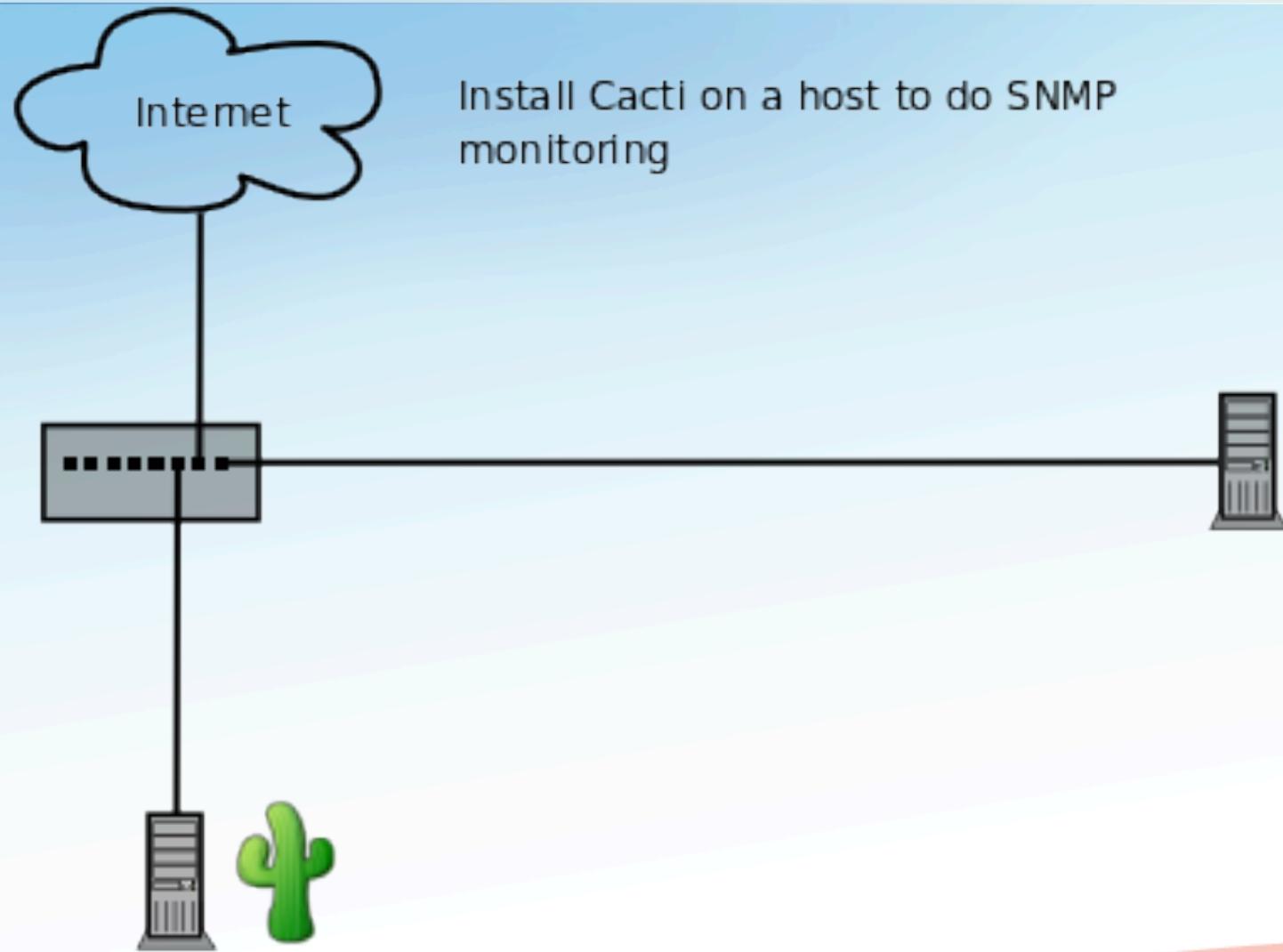
Deploying a Service

- A perfSONAR service is deployed alongside the measurement infrastructure
- Interactions with the lookup service and clients are described

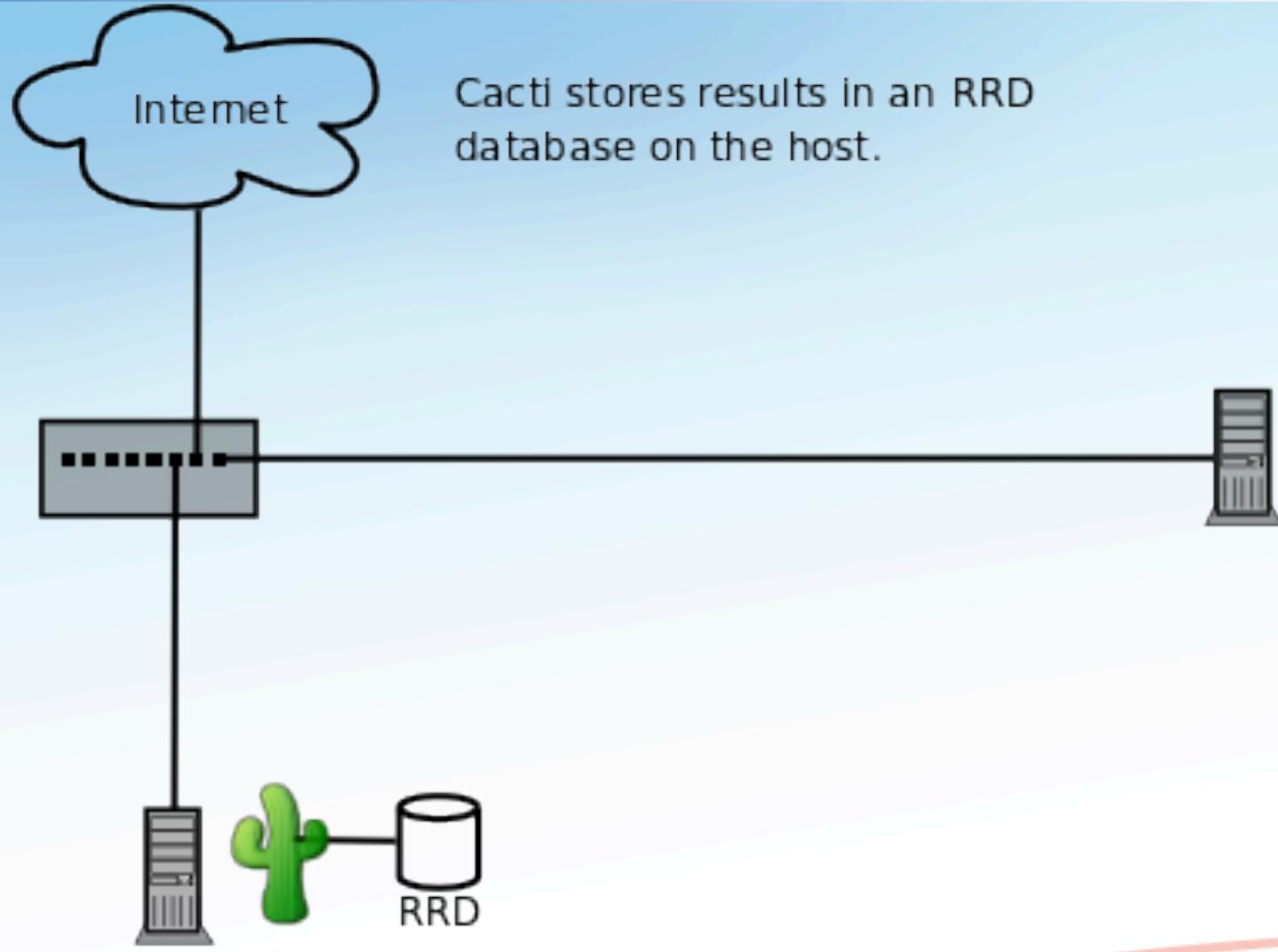
Deploying a Service



Deploying a Service

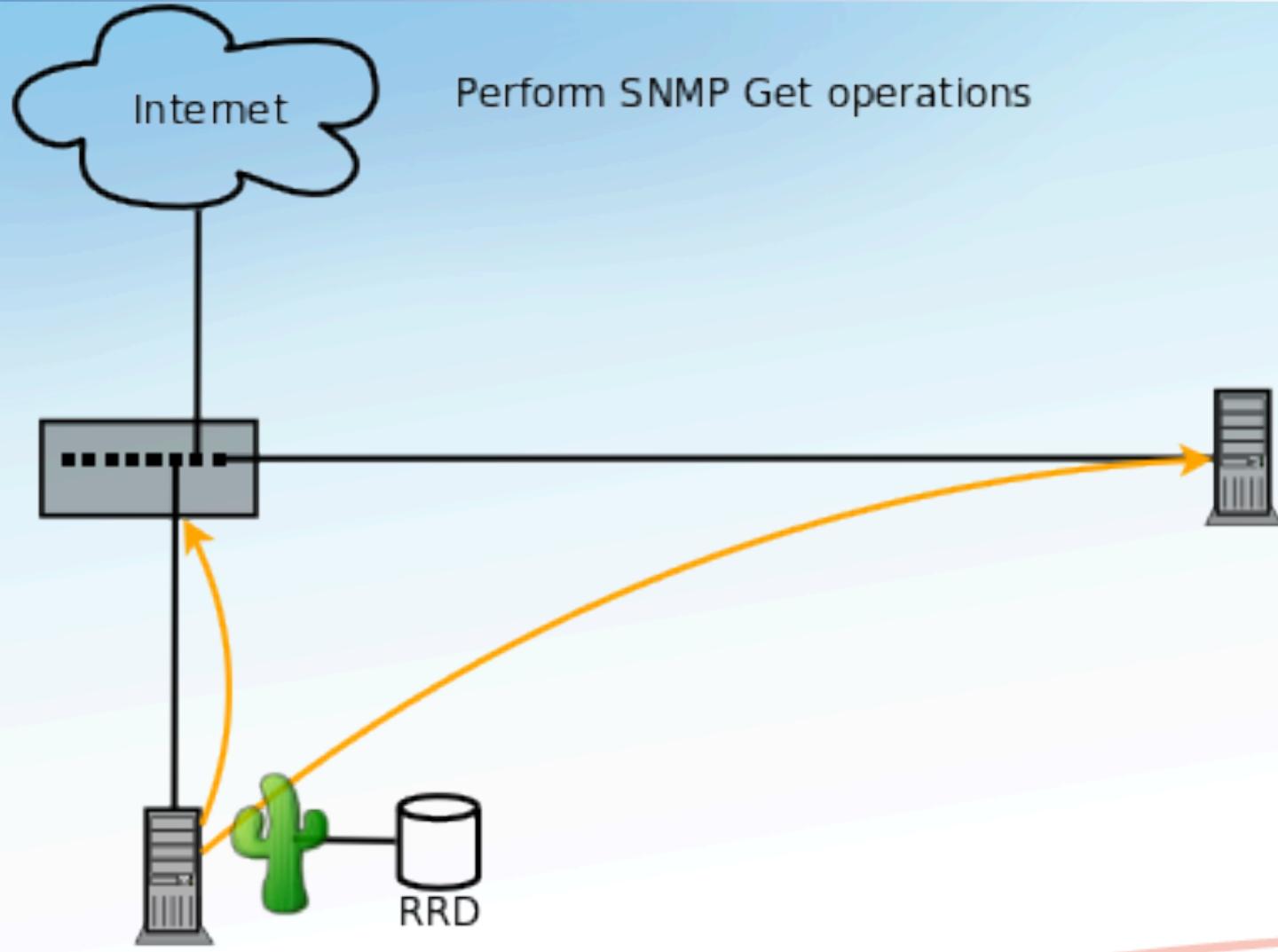


Deploying a Service

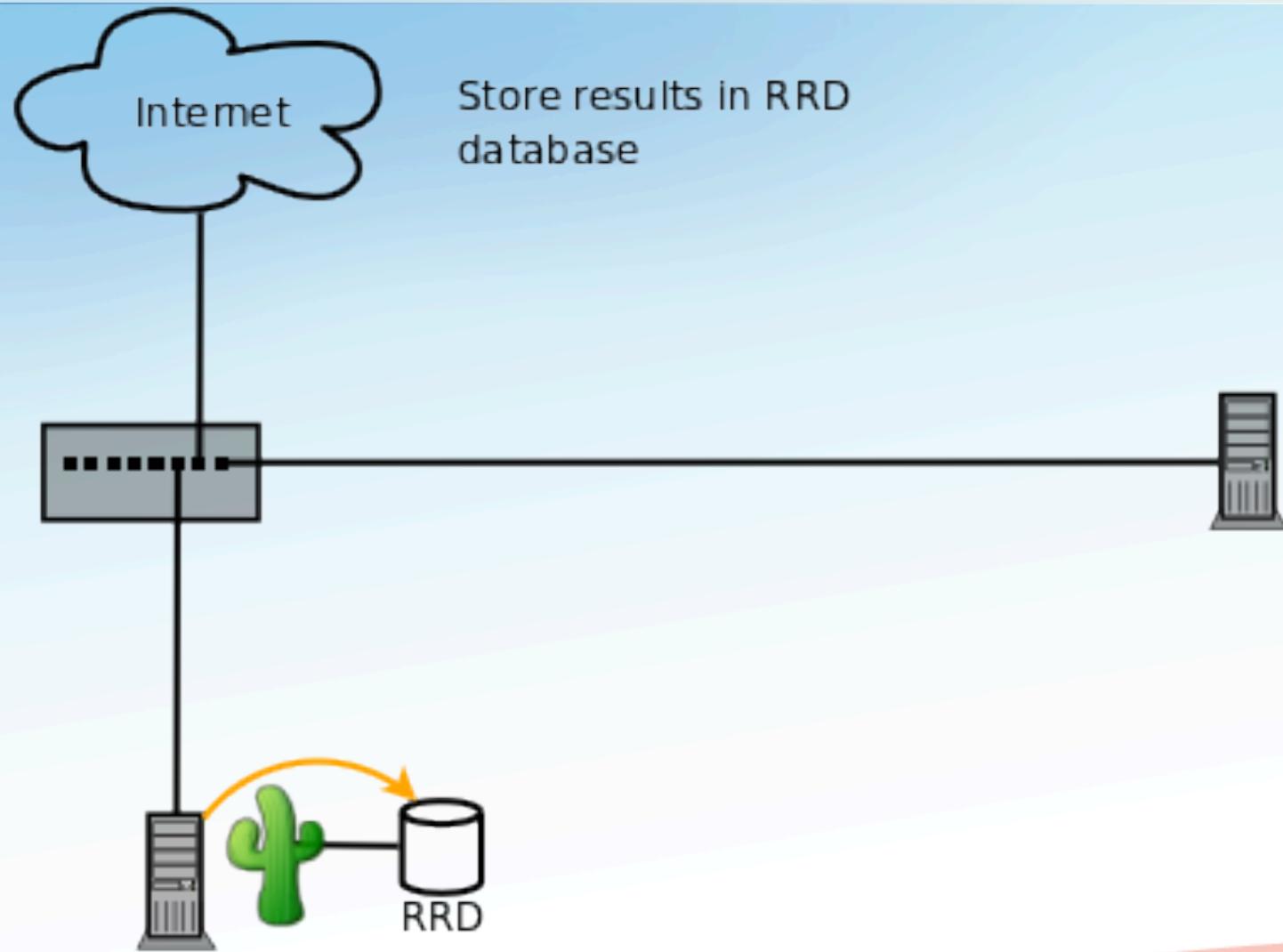


INTERNET
2

Deploying a Service

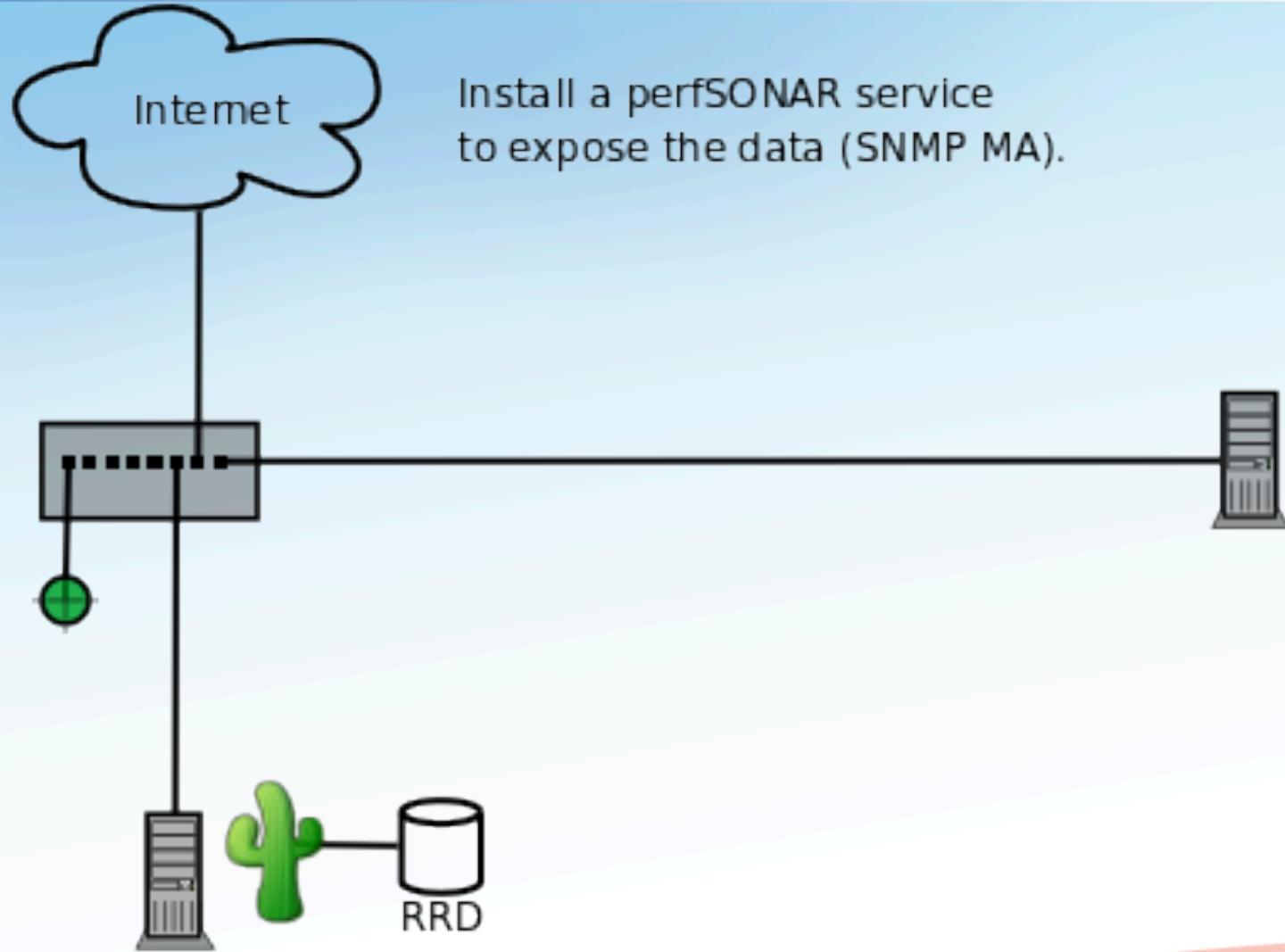


Deploying a Service

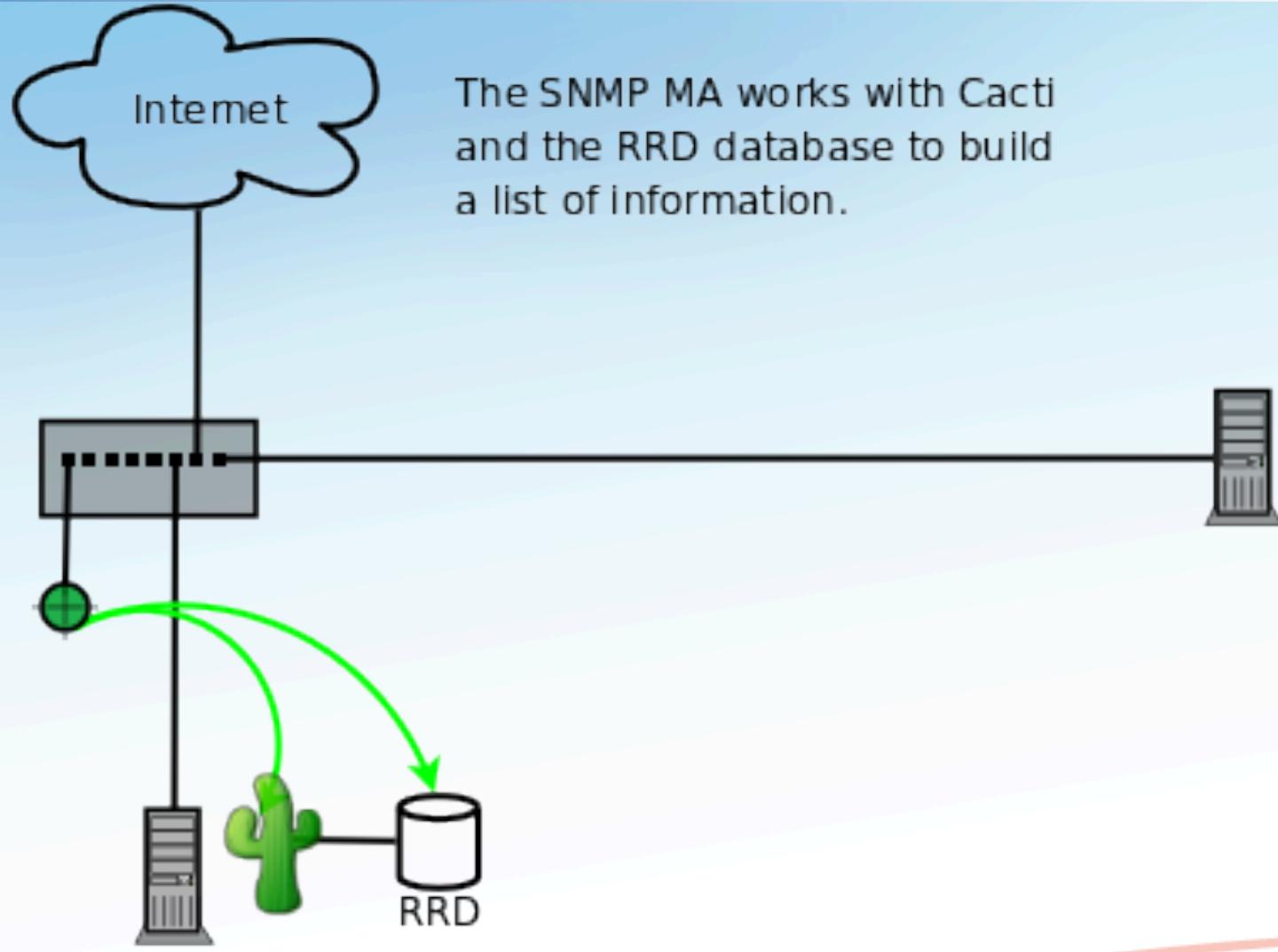


INTERNET
2

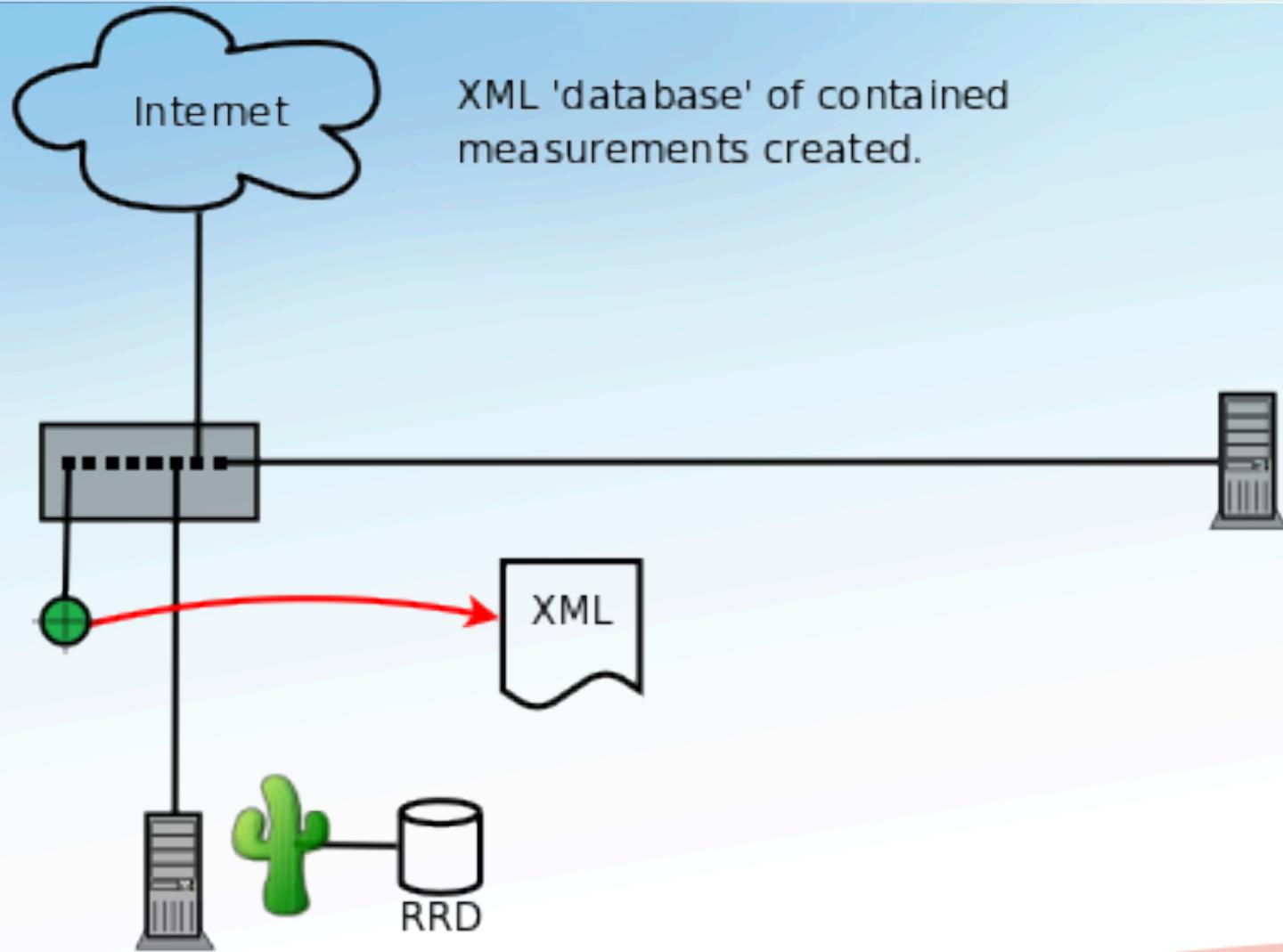
Deploying a Service



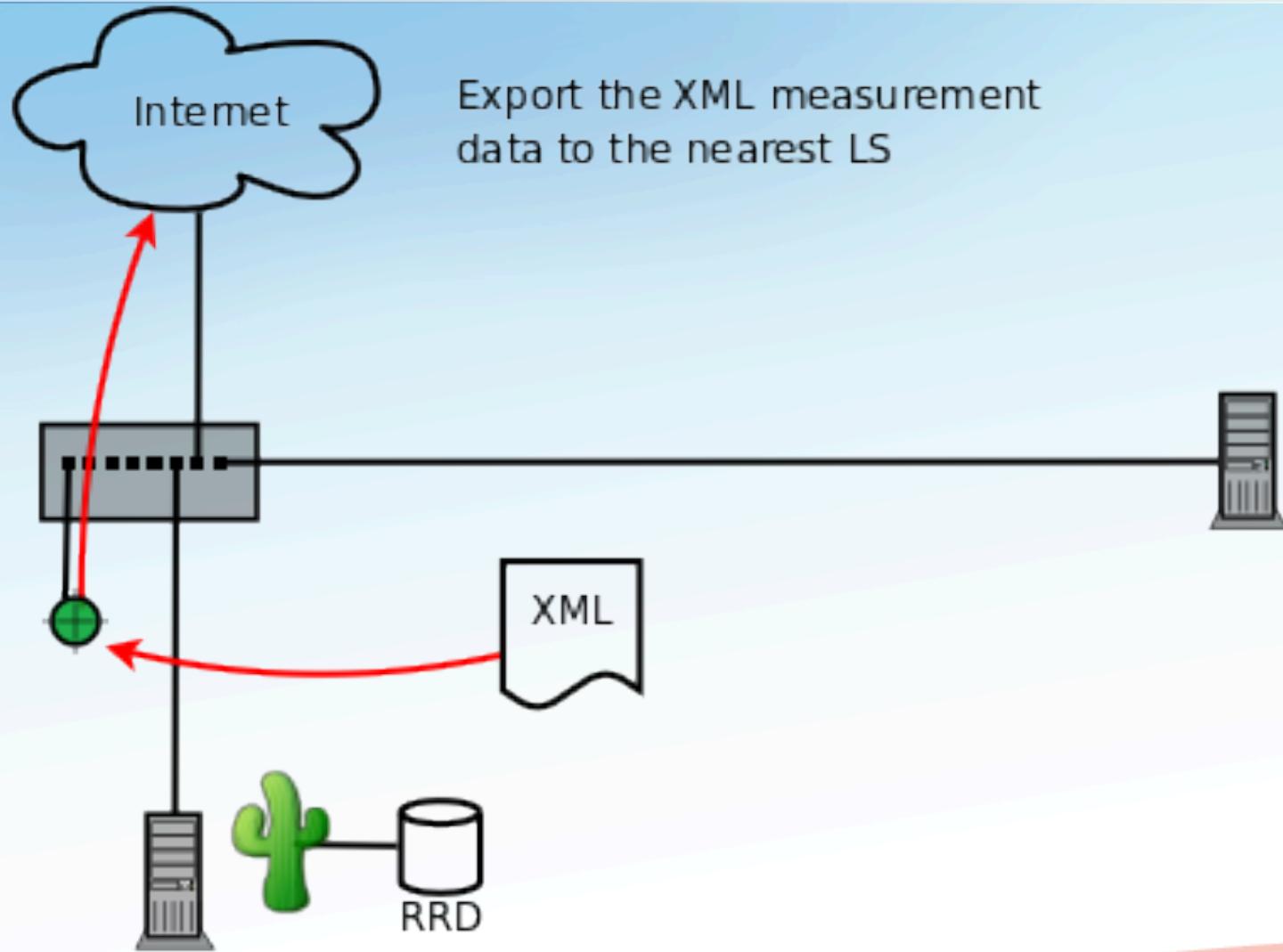
Deploying a Service



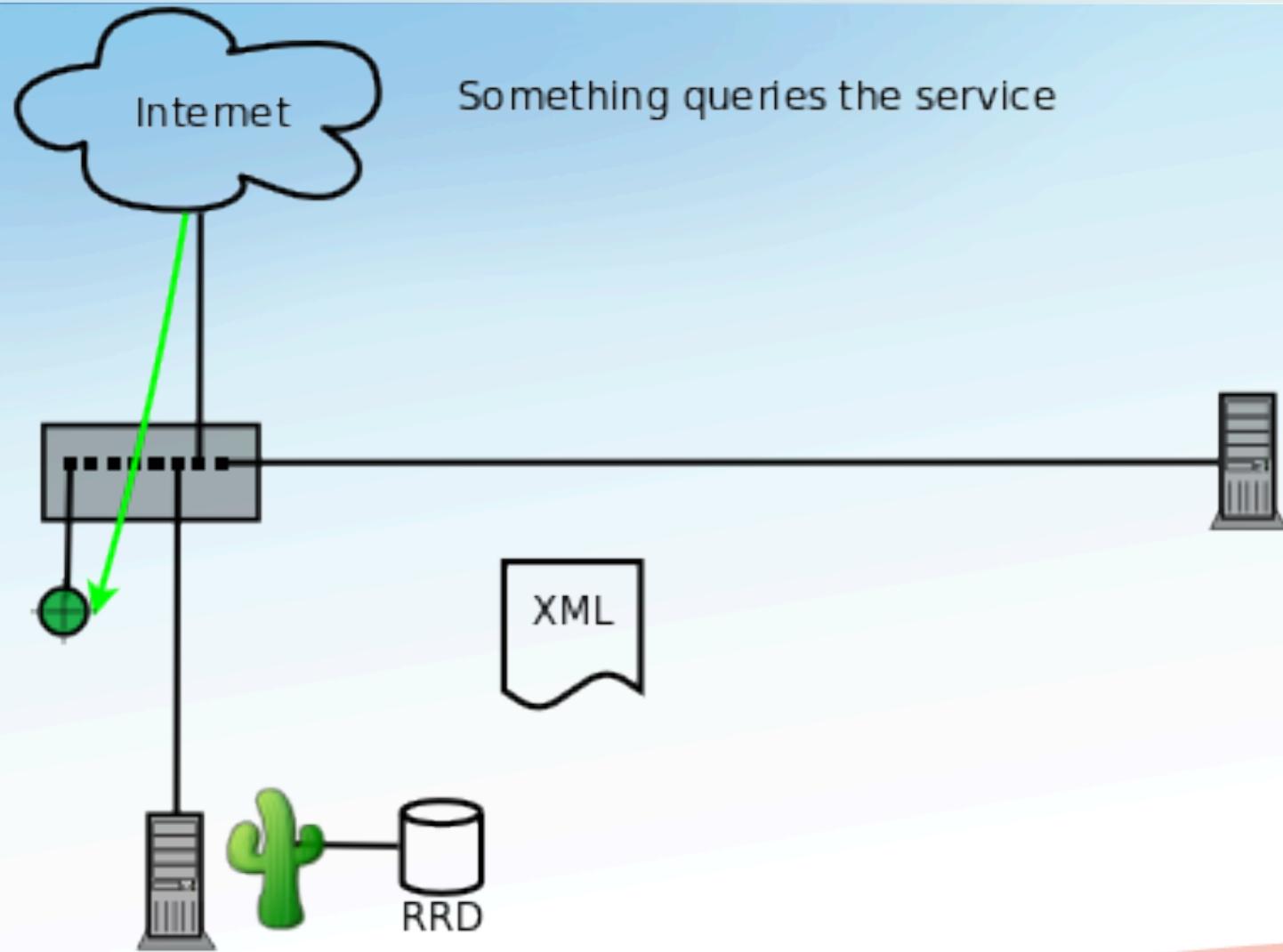
Deploying a Service



Deploying a Service

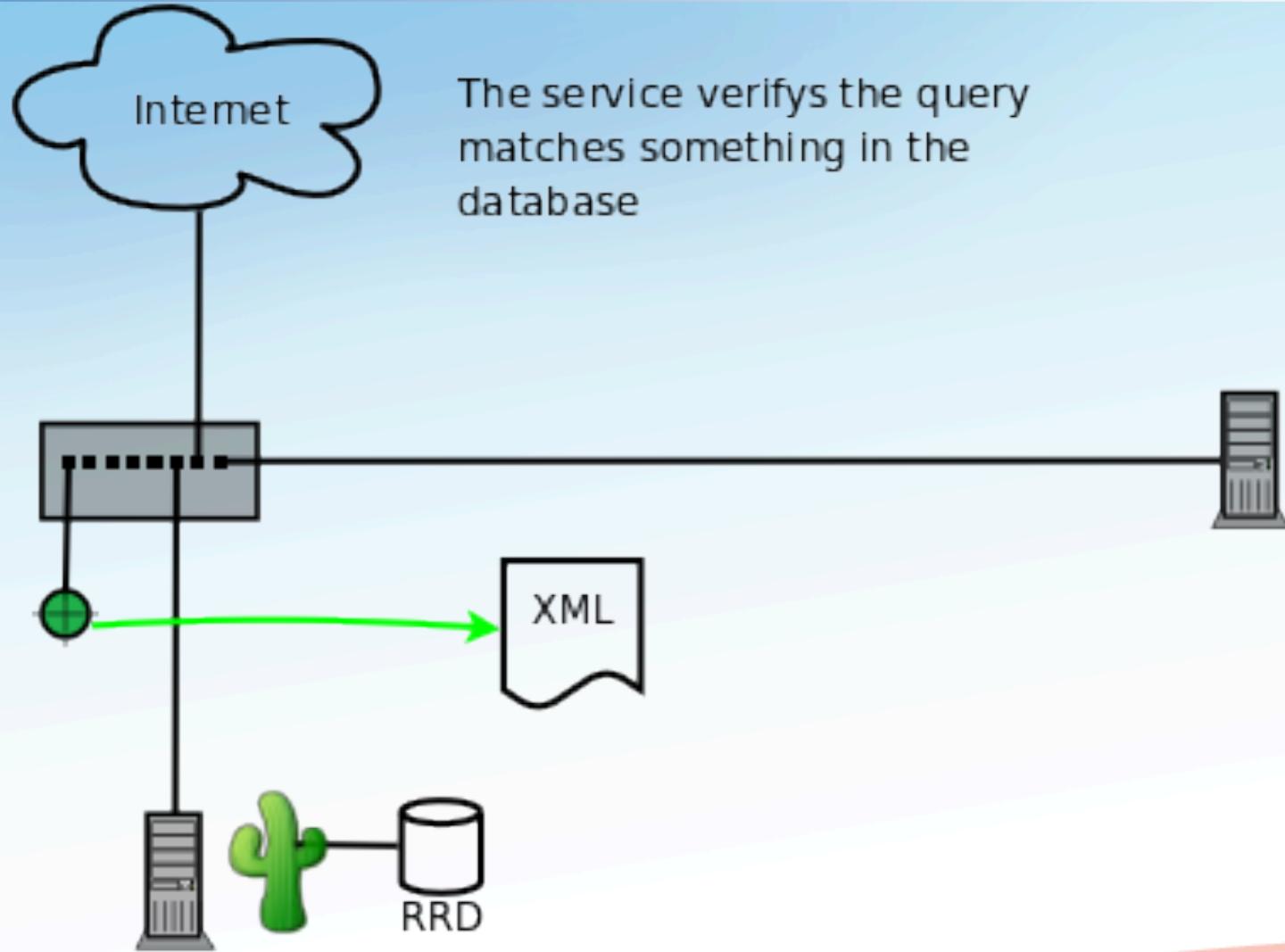


Deploying a Service



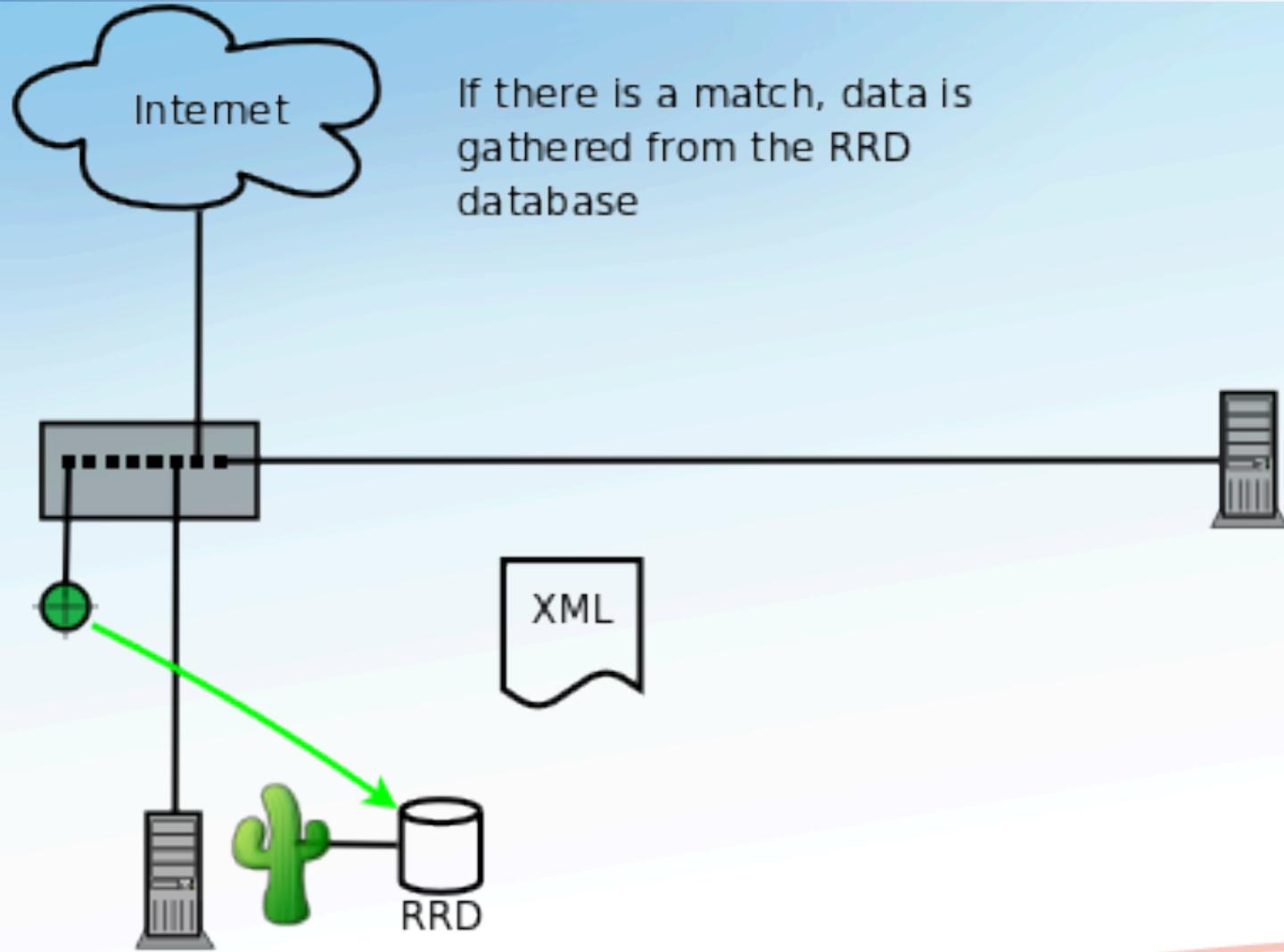
INTERNET
2

Deploying a Service

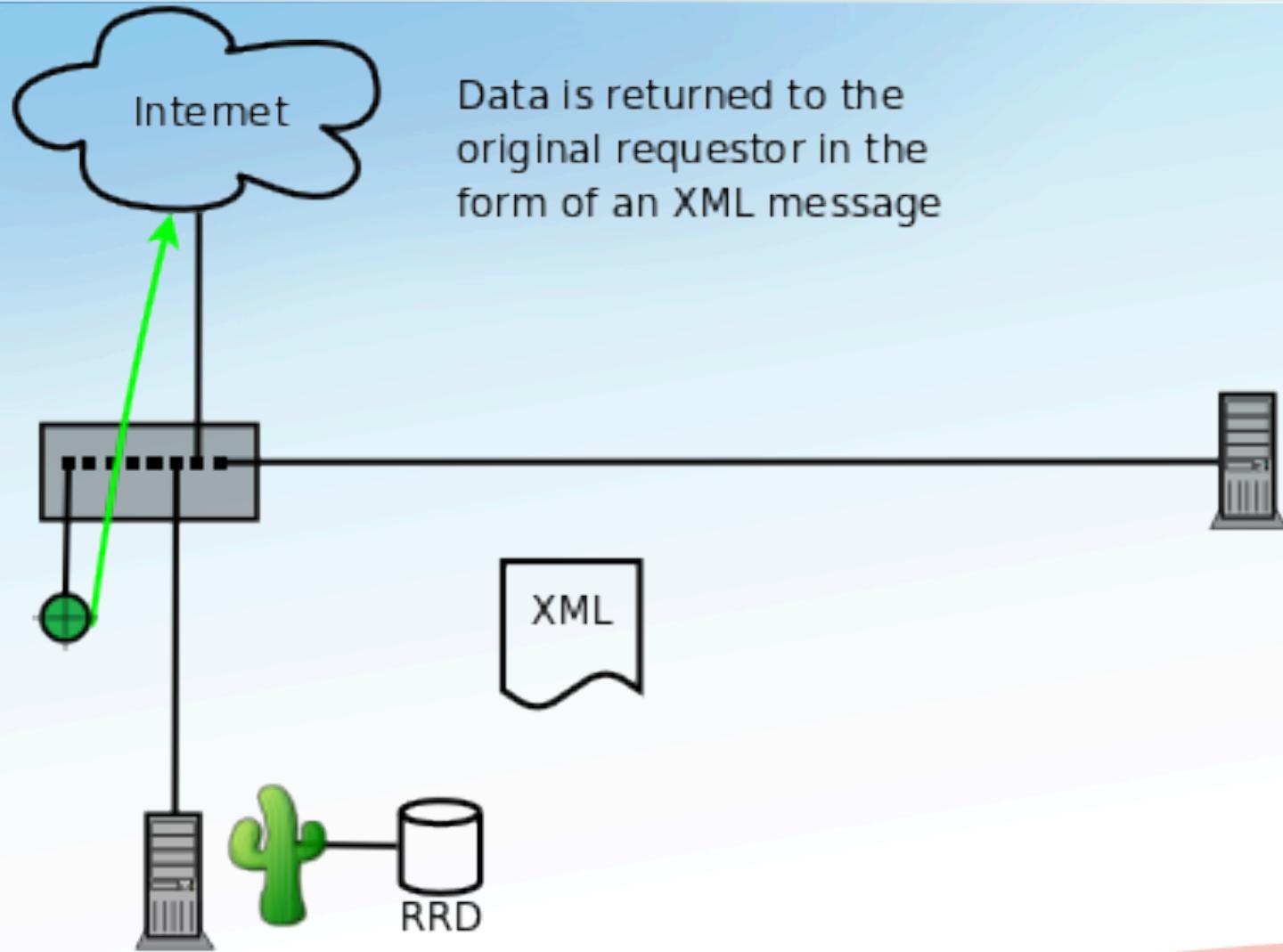


INTERNET
2

Deploying a Service



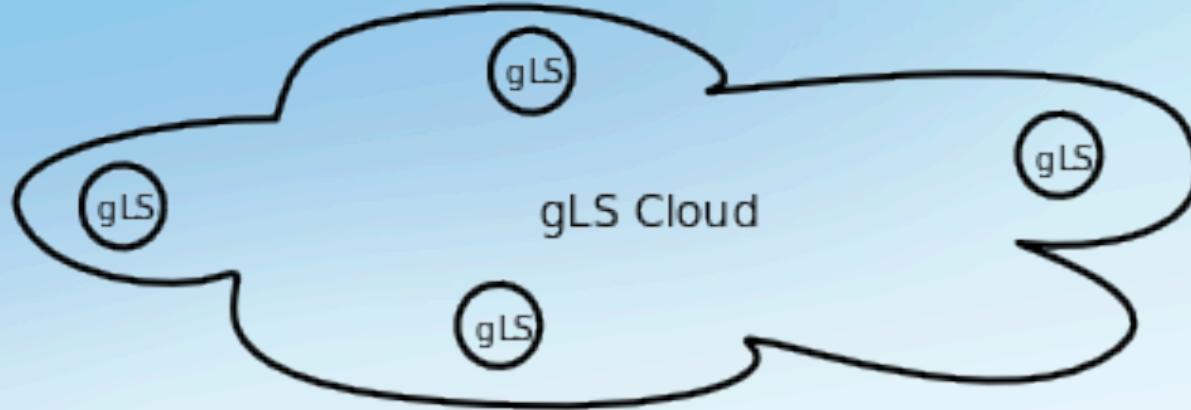
Deploying a Service



Lookup Service Interaction

- Services register with an hLS
- hLSs summarize what they know and pass to the gLSs
- gLSs exchange the information as needed
- Clients will need a multi-step process to find information
 - Query the gLS
 - Query the appropriate hLS
 - Query the appropriate services

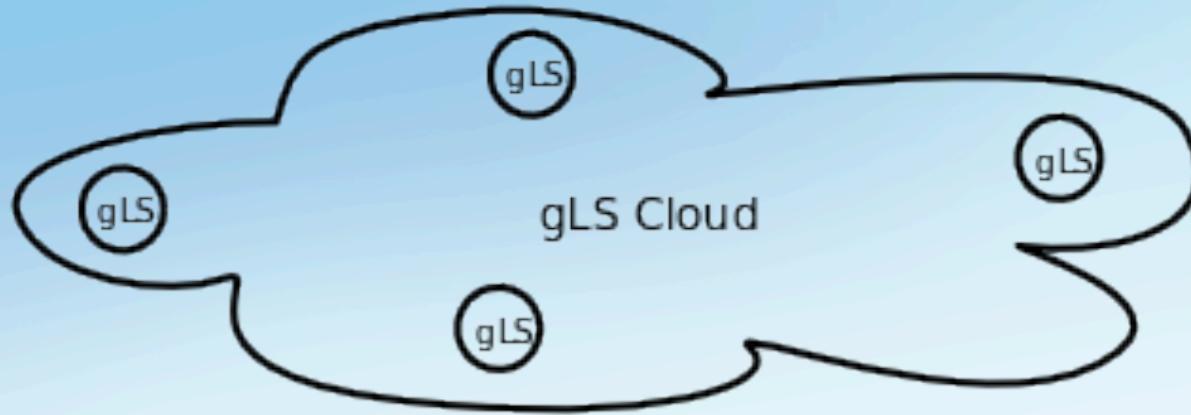
Lookup Service Interaction



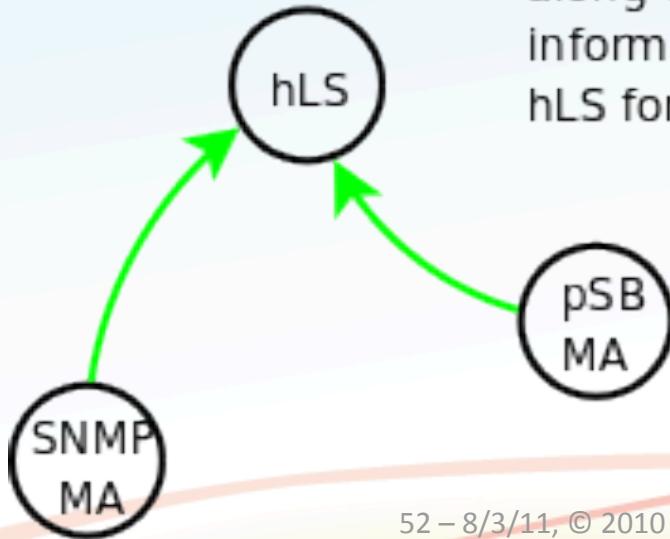
Initial Network, a domain has deployed some perfSONAR services and an hLS



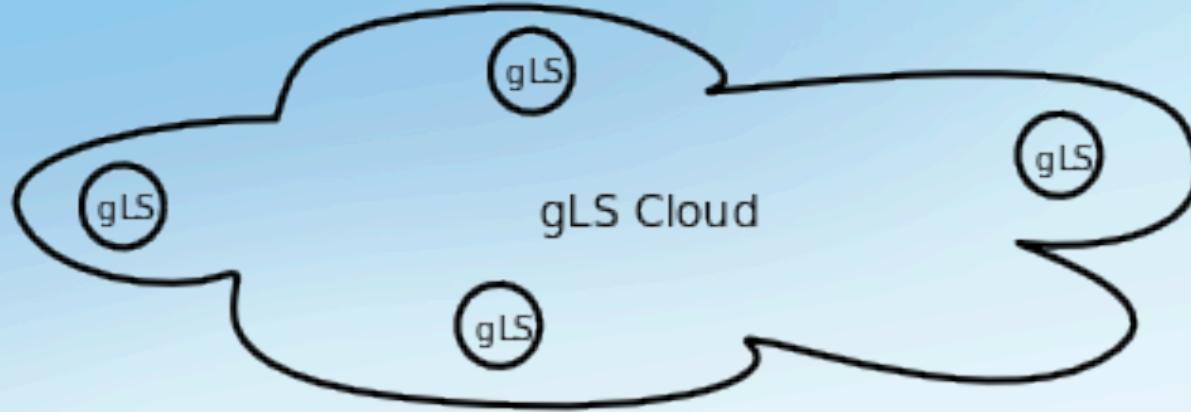
Lookup Service Interaction



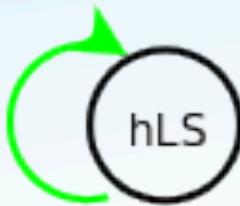
The services register with the hLS. Data they are collecting along with other service information is sent to the hLS for processing



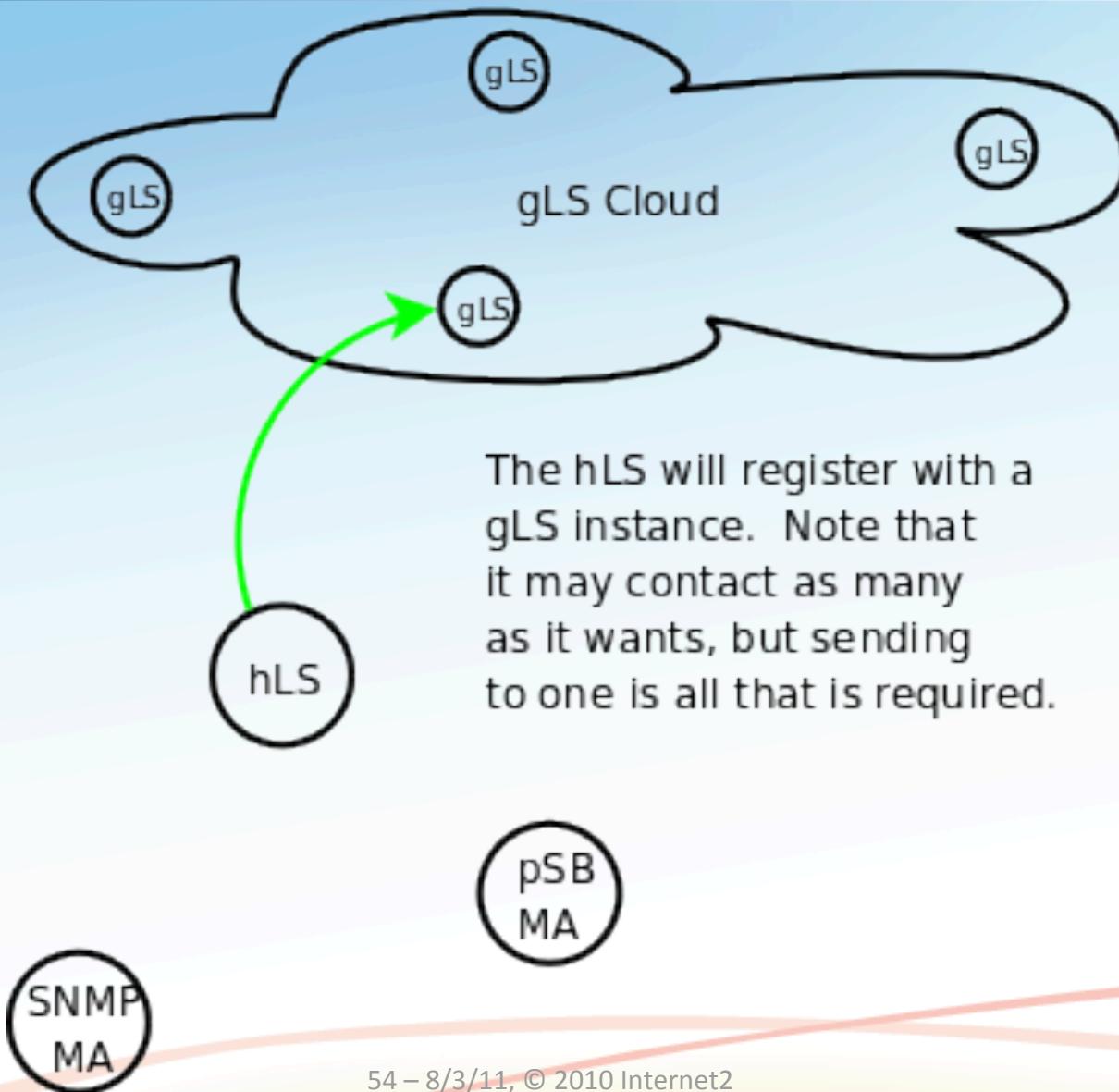
Lookup Service Interaction



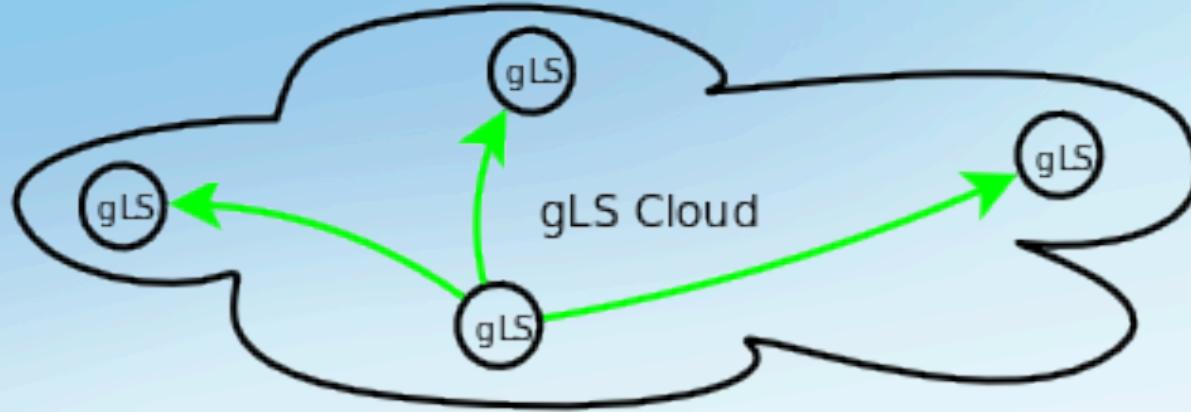
The hLS will 'summarize' the information it receives into a format that the gLSs want to see.



Lookup Service Interaction



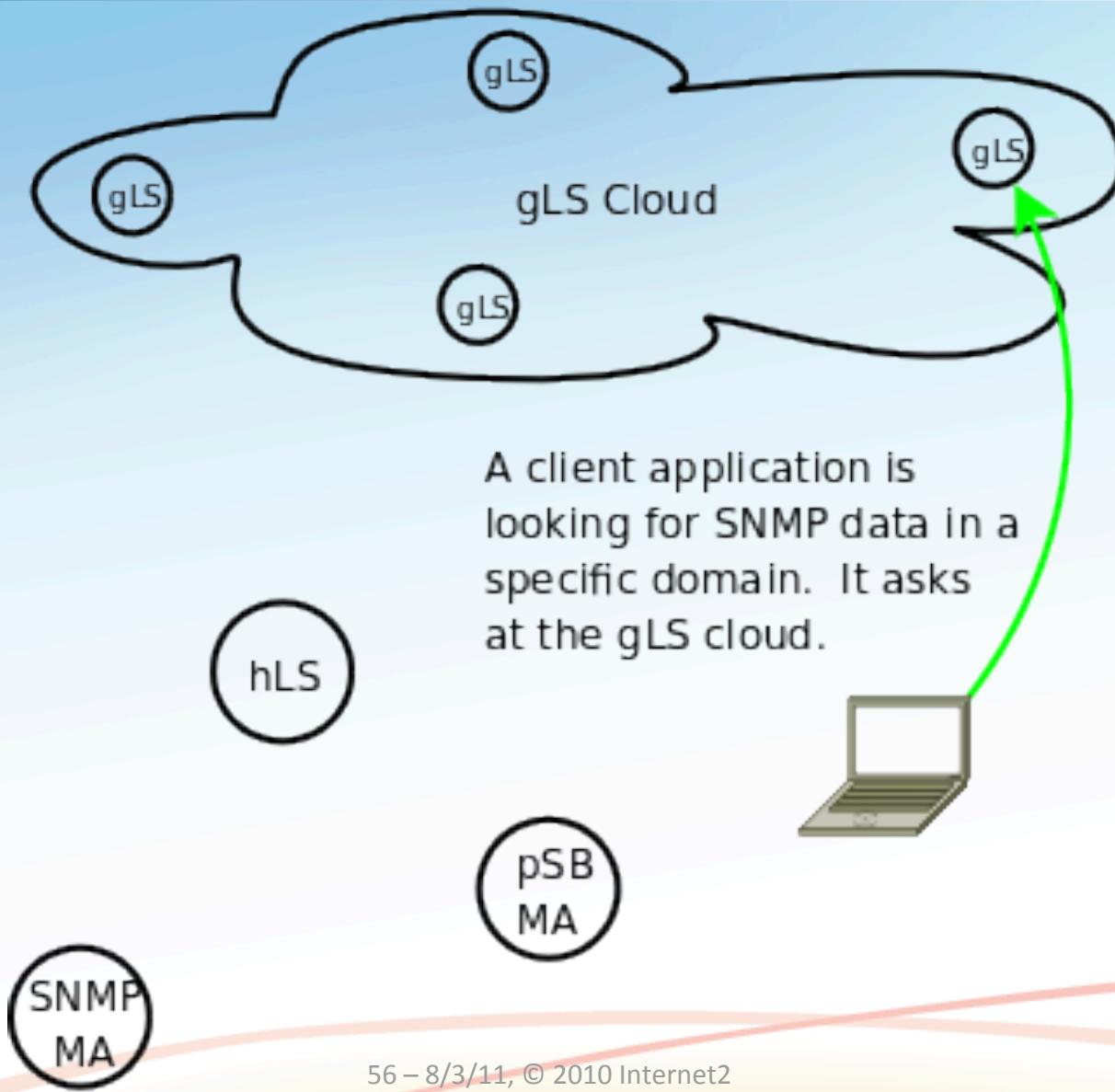
Lookup Service Interaction



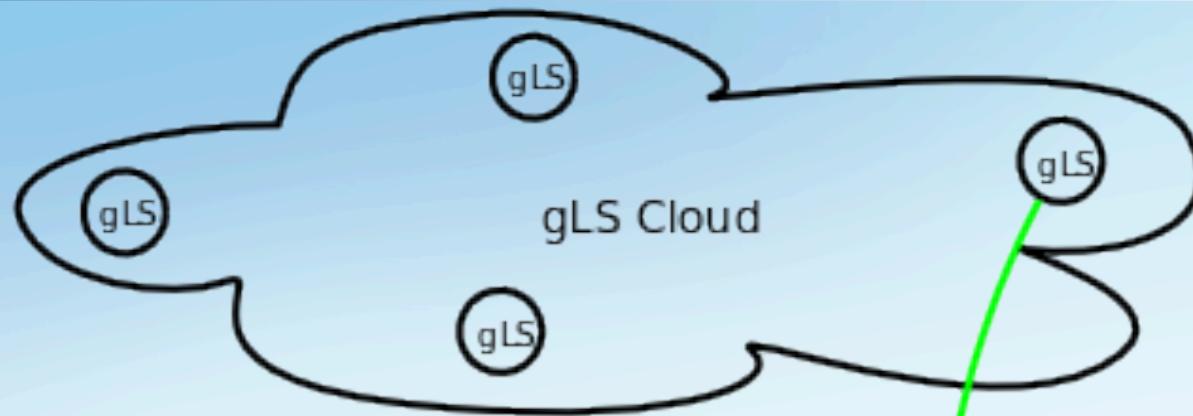
Periodically each gLS will share the hLSs it knows about with the others.



Lookup Service Interaction



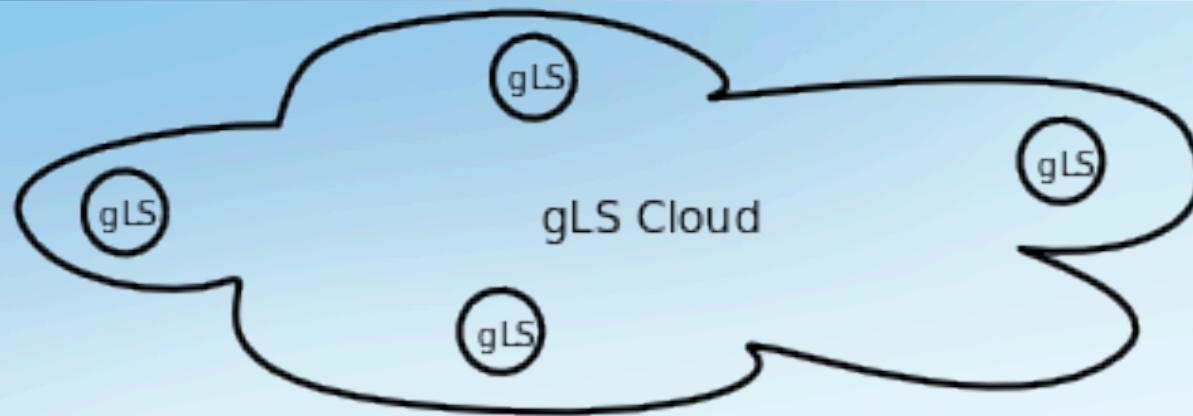
Lookup Service Interaction



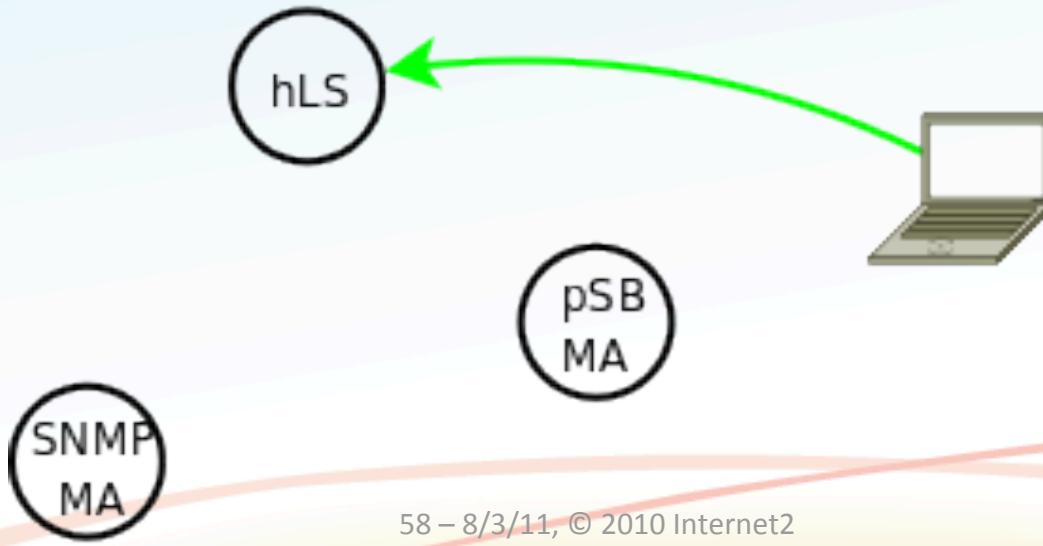
The gLS will respond with the hLS that should be contacted for more information.



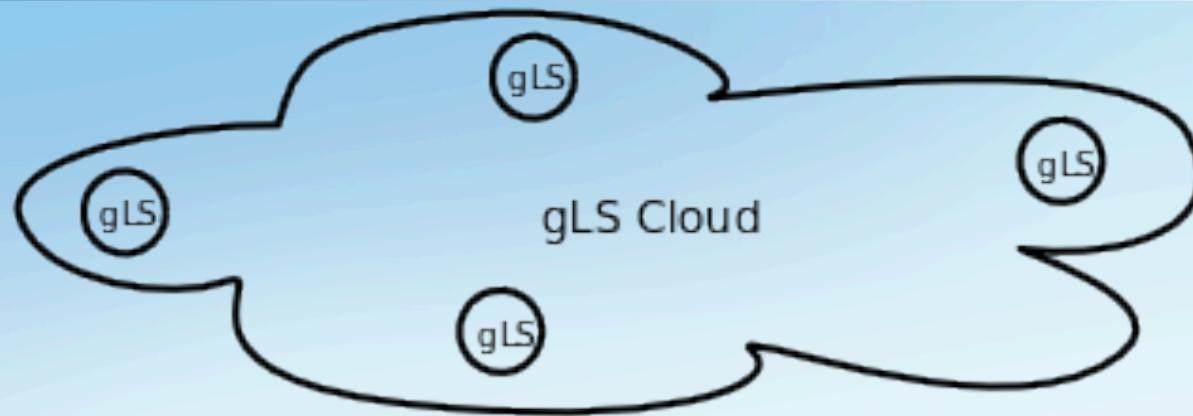
Lookup Service Interaction



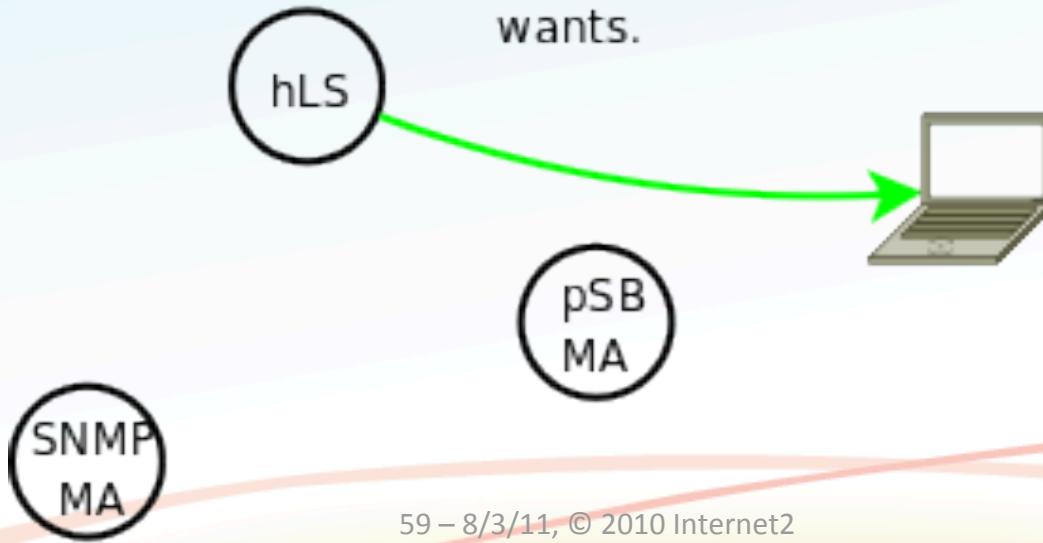
The client will ask a similar, perhaps more specific query to the hLS.



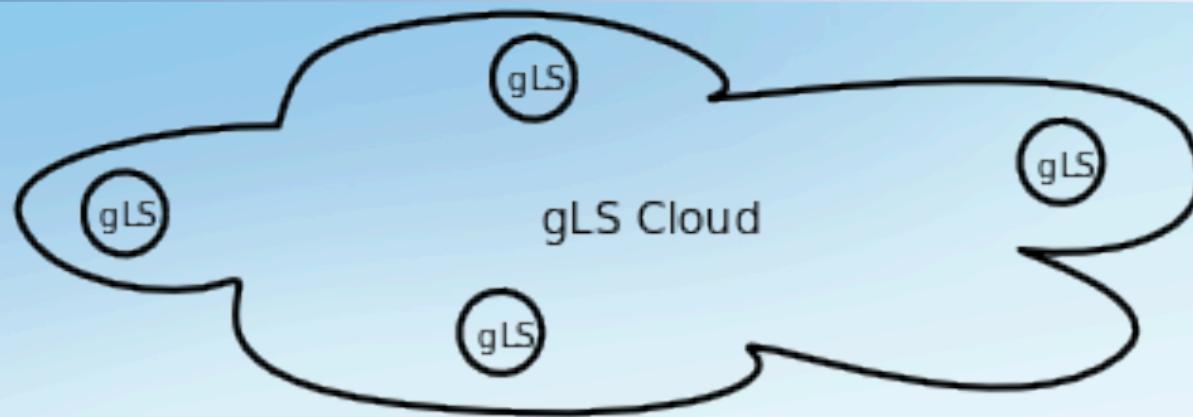
Lookup Service Interaction



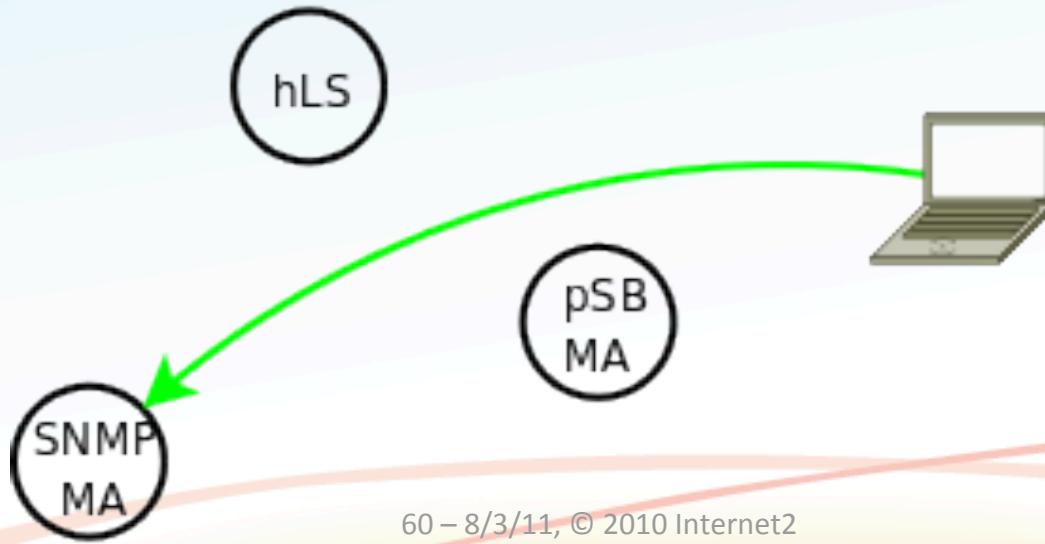
The hLS will respond with the service name that is likely to have the data the client wants.



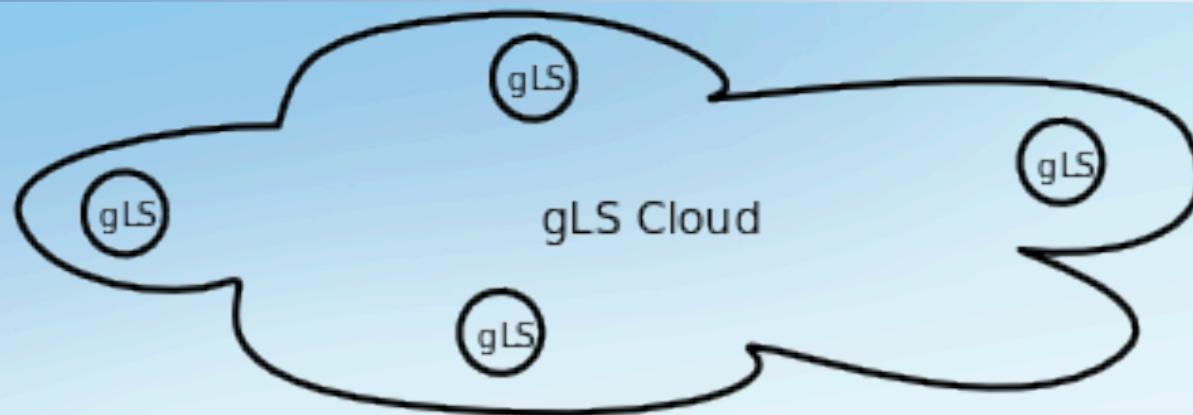
Lookup Service Interaction



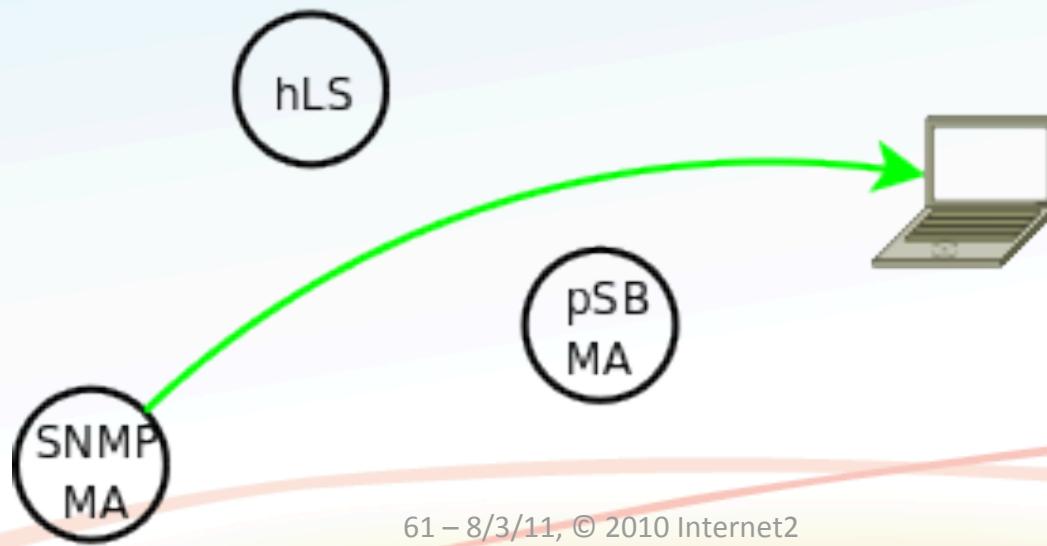
The client will contact the service suggested by the hLS for data.



Lookup Service Interaction



The service returns the data
to the client.



perfSONAR-PS Availability

- *perfSONAR-PS* is an implementation of the *perfSONAR* measurement infrastructure and protocols written in the *perl* programming language
- All products are available as platform and architecture independent source code.
- All products are available as **RPMs** (e.g. *RPM Package Manager*). The *perfSONAR-PS* consortium directly supports the following operating systems:
 - **CentOS** (version 5)
- RPMs are compiled for the x86 (should work w/ x86 64 bit) architecture.
- Functionality on other platforms and architectures is possible, but not supported. Attempts are done at the user's own risk.
 - Should work:
 - **Scientific Linux** (versions 4 and 5)
 - **Red Hat Enterprise Linux** (versions 4 and 5)
 - Harder, but possible:
 - **Fedora Linux** (any recent version)
 - **SuSE** (any recent version)
 - **Debian Variants** (...)

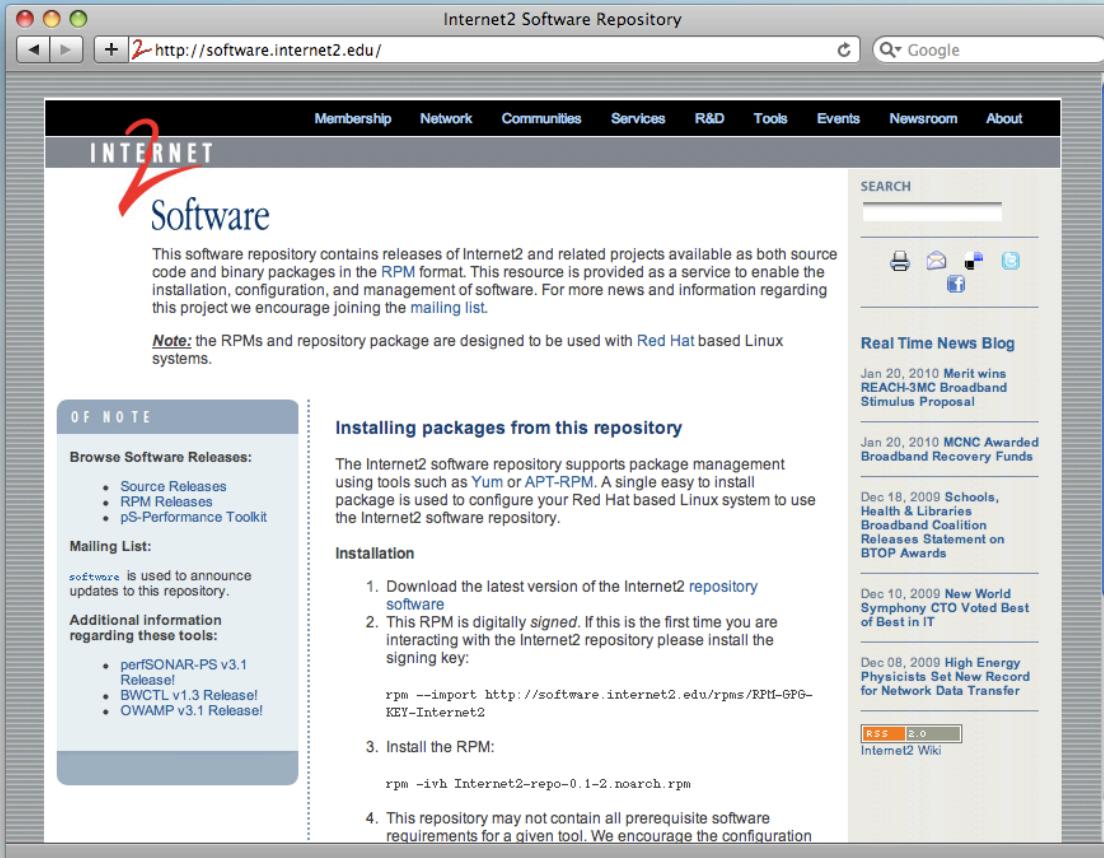


perfSONAR-PS Availability

- The pS Performance Toolkit (*pSPT*) is a Linux ISO image (e.g. a *LiveCD*) packed by Internet2 for both easy of installation and configuration of performance tools
 - Prior:
 - Based on Knoppix Linux
 - Current:
 - Based on CentOS (version 5) Linux
 - Designed for x86 architecture
 - No explicit support for x86 64 bit but compatibility is expected
- Product also contains other relevant measurement tools and *perfSONAR-PS* dependencies.
- Support structure is limited to the following goals:
 - Updated versions of all software (operating system and performance) with each release
 - Monitoring and alerts regarding critical security vulnerabilities for all software. Critical patches and releases available for severe cases
 - Semi annual (4 times per year) minor releases

perfSONAR-PS Availability

- *perfSONAR-PS* and the *pSPT* are available from
<http://software.internet2.edu>



The screenshot shows a web browser window displaying the Internet2 Software Repository. The URL in the address bar is <http://software.internet2.edu/>. The page features a header with links to Membership, Network, Communities, Services, R&D, Tools, Events, Newsroom, and About. On the left, there's a sidebar with sections for 'OF NOTE' (Browse Software Releases: Source Releases, RPM Releases, pS-Performance Toolkit; Mailing List: software mailing list), 'Additional information regarding these tools:' (perfSONAR-PS v3.1 Release!, BWCTL v1.3 Release!, OWAMP v3.1 Release!). The main content area has a heading 'Installing packages from this repository'. It explains that the repository supports package management using tools like Yum or APT-RPM. It provides instructions for installation, including a command line example: `rpm --import http://software.internet2.edu/rpms/RPM-GPG-KEY-Internet2`. Below this, there's a note about the repository not containing all prerequisite software requirements. To the right, there's a 'Real Time News Blog' section with recent posts and an RSS feed link.

perfSONAR-PS Availability

- To facilitate installation and updates on the supported platforms, installation is available through several package managers:
 - YUM
 - Up2date
 - APT-RPM
- Instructions to enable are available on
<http://software.internet2.edu>
- Installing software becomes a simple one step operation
 - Dependencies are managed by the operating system
 - Software is identified by name, and can be searched for

perfSONAR-PS Availability

- Using YUM to search for packages:

```
[zurawski@clean-centos5 ~]$ sudo yum search perfSONAR
perl-perfSONAR_PS-LookupService.noarch : perfSONAR_PS Lookup Service
perl-perfSONAR_PS-TopologyService.noarch : perfSONAR_PS Topology Service
perl-perfSONAR_PS-Status.noarch : perfSONAR-PS Status Service
perl-perfSONAR_PS-PingER-server.noarch : perfSONAR_PS PingER Measurement Archive and Collection System
perl-perfSONAR_PS-perfAdmin.noarch : perfSONAR_PS perfAdmin
perl-perfSONAR_PS-perfSONARBUOY-client.noarch : perfSONAR_PS perfSONARBUOY Web Service Client and Measurement System
perl-perfSONAR_PS-LSRegistrationDaemon.noarch : perfSONAR_PS Lookup Service Registration Daemon
perl-perfSONAR_PS-perfSONARBUOY-server.noarch : perfSONAR_PS perfSONARBUOY Measurement Archive and Collection System
perl-perfSONAR_PS-perfSONARBUOY-config.noarch : perfSONAR_PS perfSONARBUOY Configuration Information
perl-perfSONAR_PS-PingER-GUI.i386 : perfSONAR_PS PingER data charts GUI
perl-perfSONAR_PS-SNMPMA.noarch : perfSONAR_PS SNMP Measurement Archive
```



perfSONAR-PS Availability

- Using YUM to install packages:

```
[zurawski@clean-centos5 ~]$ sudo yum install owamp-client
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package owamp-client.i386 0:3.2rc1-1 set to be updated
--> Finished Dependency Resolution
Dependencies Resolved

=====
Package          Arch    Version      Repository      Size
=====
Installing:
owamp-client    i386   3.2rc1-1   Internet2      198 k

Transaction Summary
=====
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 198 k
Is this ok [y/N]:
```



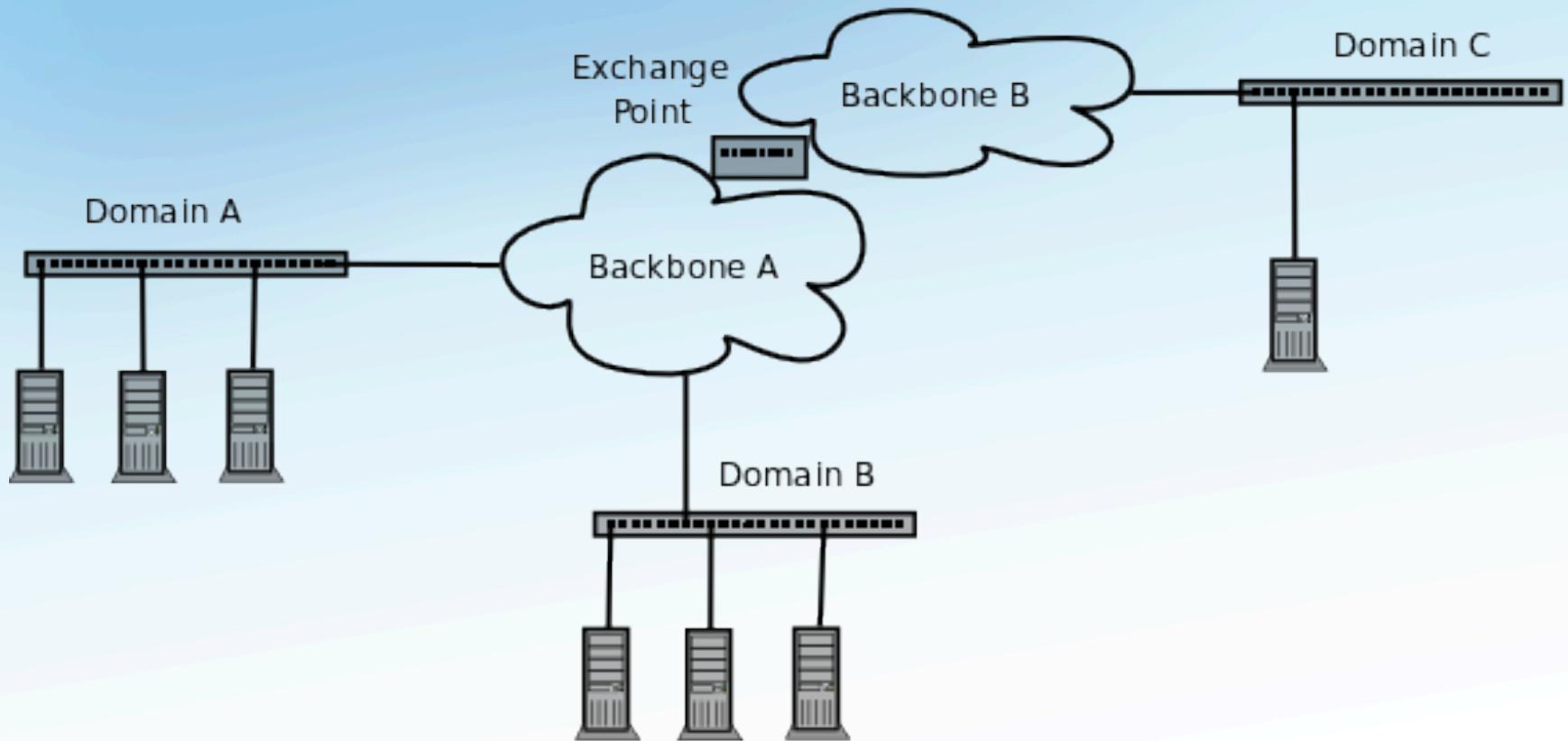
perfSONAR-PS Availability

- perfSONAR-PS is working to build a strong user community to support the use and development of the software.
- perfSONAR-PS Mailing Lists
 - Users List: <https://mail.internet2.edu/wws/subrequest/perfsonar-ps-users>
 - Announcement List:
<https://mail.internet2.edu/wws/subrequest/perfsonar-ps-announce>
- pSPT Mailing Lists
 - Users List:
<https://mail.internet2.edu/wws/subrequest/performance-node-users>
 - Announcement List:
<https://mail.internet2.edu/wws/subrequest/performance-node-announce>

perfSONAR-PS Tools

- *perfSONAR-PS* Tools can be broken into 3 categories:
 - Measurement Points and Archives
 - Information Services
 - Analysis tools and GUIs
- The following sections will explain the differences between the categories as well as describe the utility each offers.
- Development on *perfSONAR-PS* continues to be active and new services are added over time
- Contributions from the community with regards to ideas or development work are encouraged

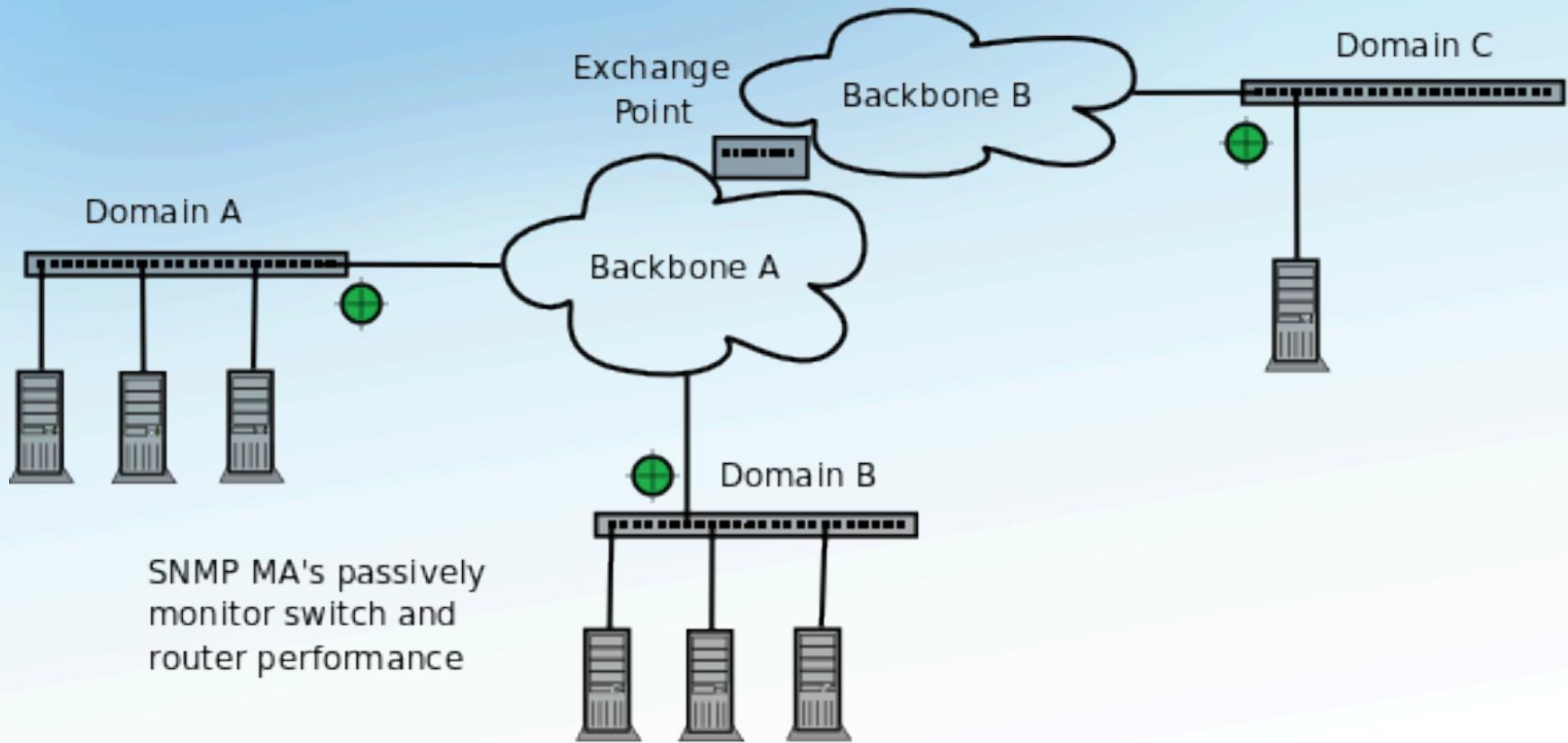
perfSONAR-PS Framework Deployment



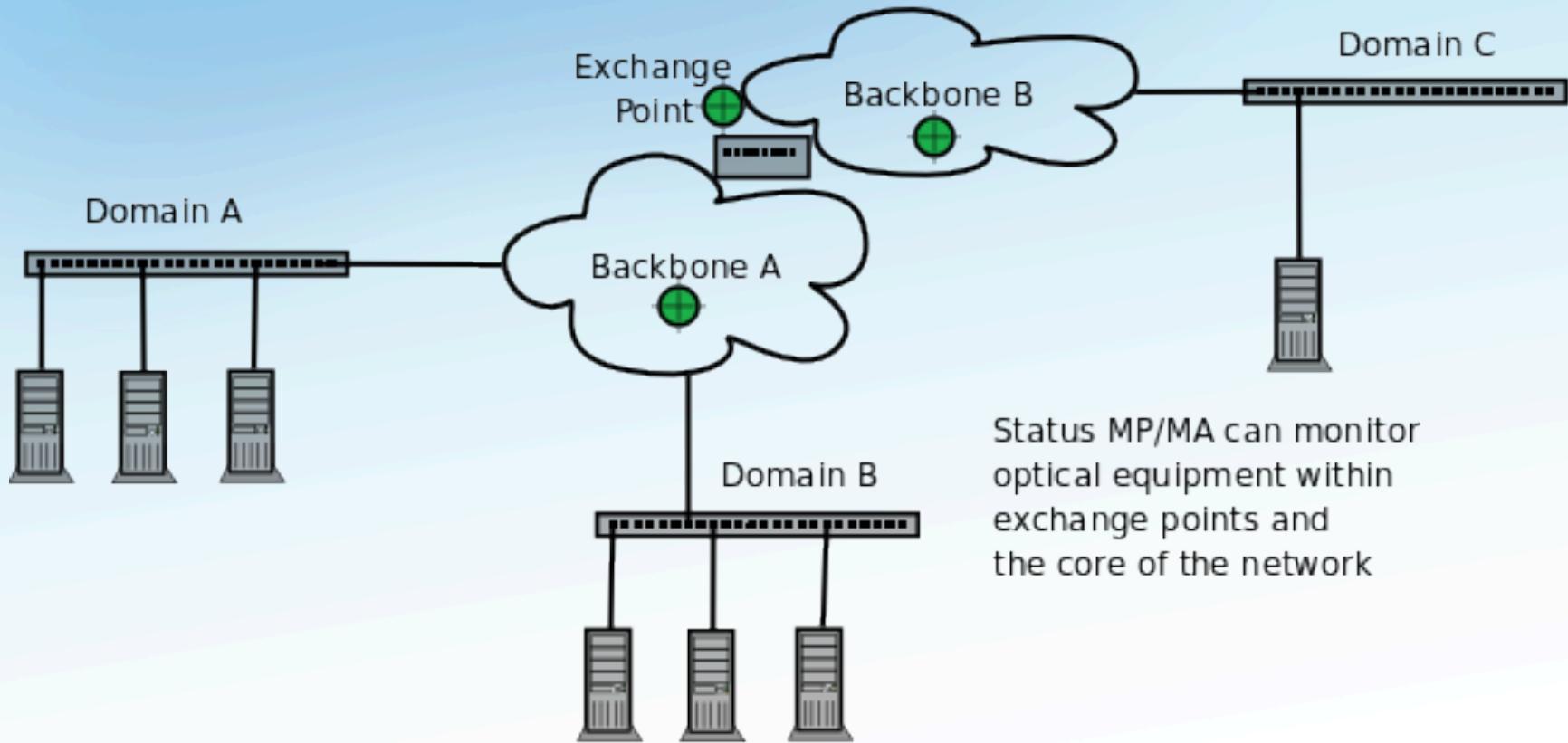
perfSONAR-PS Tools - MAs

- Measurement Archives store the results of measurements
 - Results can come from external sources (e.g. Cacti, MRTG, Cricket)
 - Results can come from a perfSONAR MP performing measurements
- Designed to run on any machine
 - Should be located in the same physical domain, but this is not a requirement
- Examples in *perfSONAR-PS*
 - **SNMP MA**: Results of SNMP collection (interface counters, etc.)
 - **Status MA**: Results of TL1 collection (optical devices)
 - **perfSONAR-BUOY**: OWAMP and BWCTL (one way latency and bandwidth testing) results
 - **PingER**: Ping (round trip latency) results.

perfSONAR-PS Framework Deployment



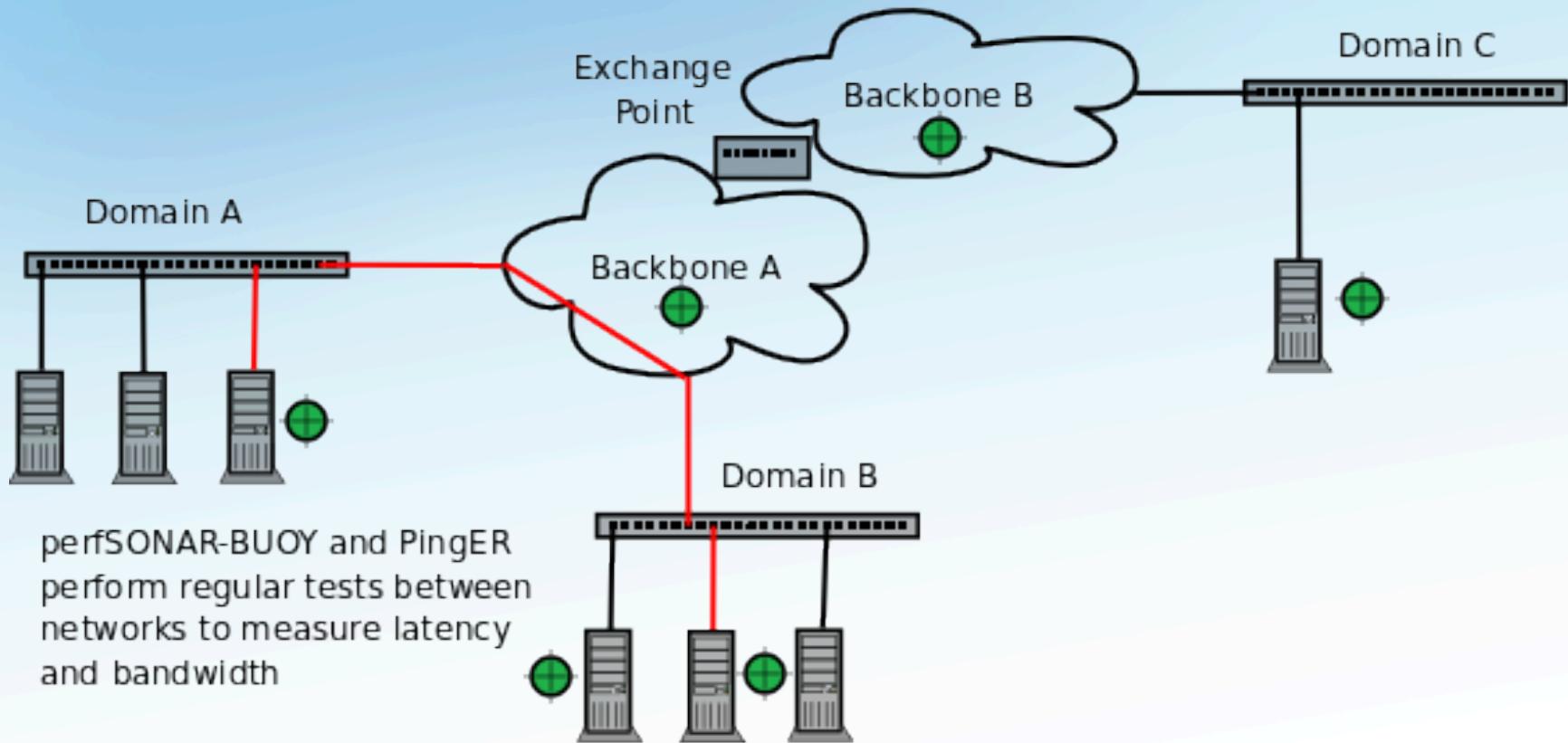
perfSONAR-PS Framework Deployment



perfSONAR-PS Tools - MPs

- Measurement Points perform measurements and store the results locally, or in Measurement Archives
 - Perform *on-demand* testing
 - Perform test according to a schedule
- Designed to run on any machine
 - Should be located in the same physical domain, but this is not a requirement
- Examples in *perfSONAR-PS*
 - ***Status MA Scripts***: Gather TL1 or SNMP results.
 - ***perfSONAR-BUOY***: OWAMP and BWCTL (one way latency and bandwidth testing) scheduled testing
 - ***PingER***: Ping (round trip latency) scheduled testing.

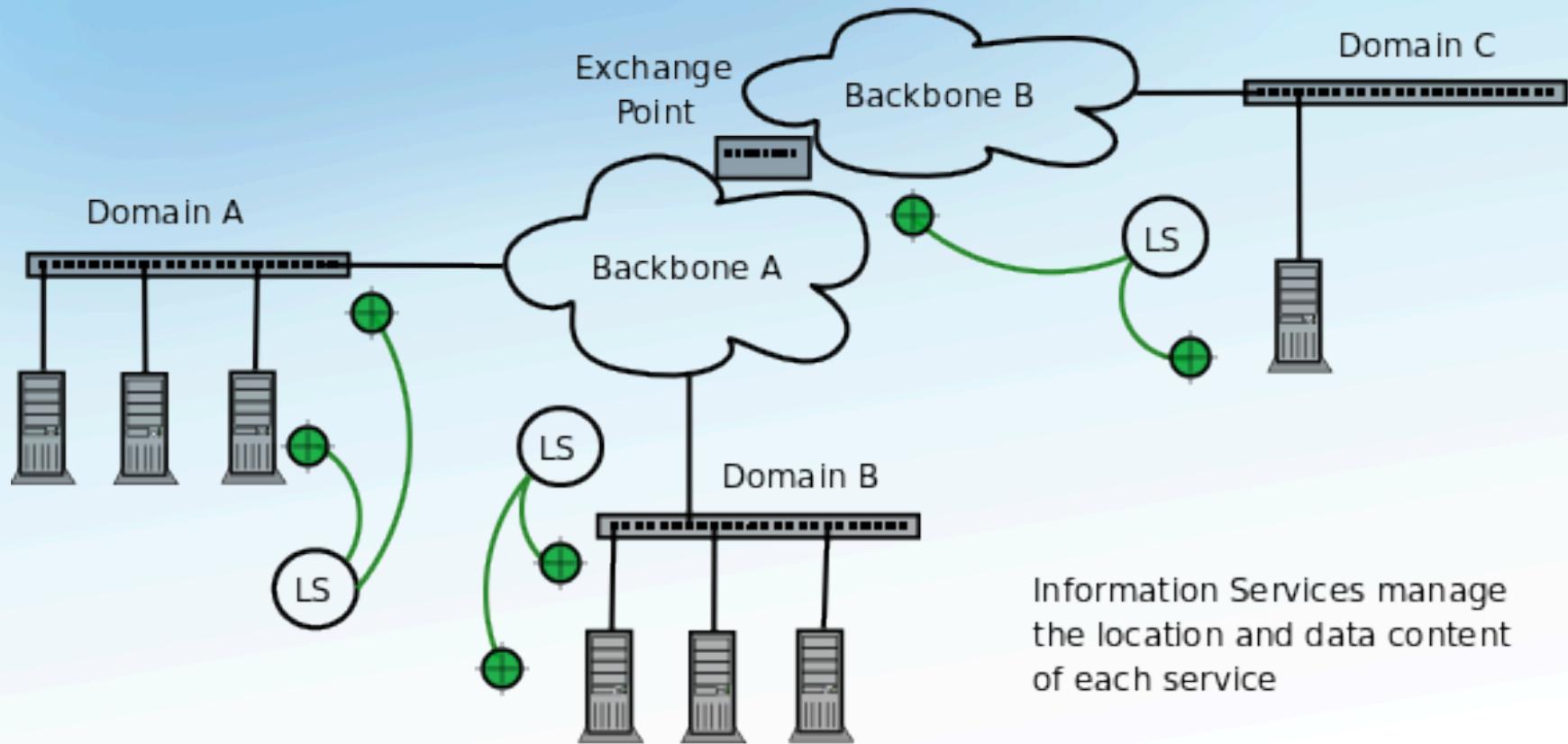
perfSONAR-PS Framework Deployment



perfSONAR-PS Tools - IS

- Information services are the *glue* of the framework
 - Collect statistics and status of deployed services
 - Answer queries from clients and services
 - Allow for federation, e.g. the sharing of information across domains
- Designed to run on any machine
 - Must be located in the same physical domain
- Examples in *perfSONAR-PS*
 - ***Lookup Service***: Manages service and data locality
 - ***Topology Service***: Manages topological knowledge for a domain

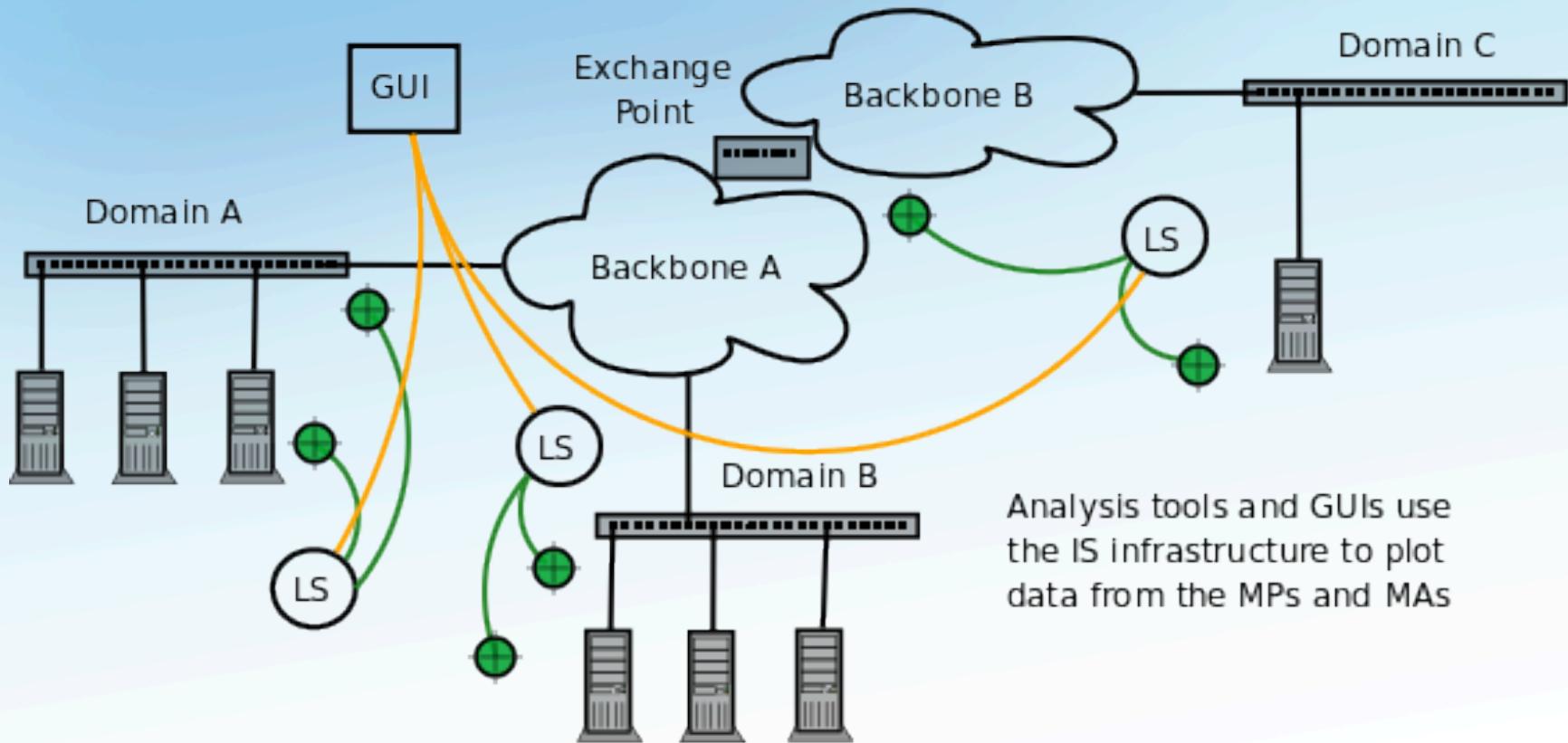
perfSONAR-PS Framework Deployment



perfSONAR-PS Tools - GUIs

- Analysis tools and GUIs plot data from *perfSONAR* services.
 - Utilize the IS infrastructure to locate specific data
 - Communicate with MAs to gather data
 - Communicate with MPs to perform live tests
- Designed to run on any machine
- Examples in *perfSONAR-PS*
 - ***perfAdmin***: CGI scripts to locate and manage *perfSONAR* services and data
 - ***PingER GUI***: Displays the results of PingER testing

perfSONAR-PS Framework Deployment



Analysis tools and GUIs use the IS infrastructure to plot data from the MPs and MAs



perfSONAR-PS Utility

- *perfSONAR-PS* appeals to both network users and operators:
 - **Operators:**
 - Easy deployment
 - Minimal maintenance
 - Results relevant to common problems (e.g. connectivity loss, equipment failure, performance problems)
 - **Users:**
 - Immediate access to network data
 - Cross domain capabilities
- Adoption is spreading to networks of all sizes
- The *perfSONAR-PS* framework has two primary high level use cases:
 - Diagnostic (e.g. *on-demand*) use
 - Monitoring Infrastructure

perfSONAR-PS Utility - Diagnostics

- The pS Performance Toolkit was designed for one-off diagnostic use
 - All tools preconfigured
 - Minimal installation requirements
 - Can deploy multiple instances for short periods of time in a domain
- Enhancements to the original design also allow for use in regular monitoring.
- Other *perfSONAR-PS* and performance tools can function in the same role as the toolkit
 - Requires longer installation/configuration phase
 - Requires semi-permanent home for services

perfSONAR-PS Utility - Monitoring

- Regular monitoring is an important design consideration for *perfSONAR-PS* tools
 - ***perfSONAR-BUOY*** and ***PingER*** provide scheduling infrastructure to create regular latency and bandwidth tests
 - The ***SNMP MA*** integrates with COTS SNMP monitoring solutions
- The pSPT is capable of organizing and visualizing regular tests
- NAGIOS can be integrated with *perfSONAR-PS* tools to facilitate alerting to potential network performance degradation

For more information

- General and MDM implementation: <http://www.perfsonar.net>
- The PS implementation: <http://psps.perfsonar.net>
- perfSONAR-PS tools and software: <http://software.internet2.edu>
- A hook to the global lookup service:
<http://www.perfsonar.net/activeServices/IS/>
- More human-readable list of services:
<http://www.perfsonar.net/activeServices/>

Mailing Lists

- Development (by approval of the project)
 - <https://lists.internet2.edu/sympa/subscribe/perfsonar-dev>
- User Support
 - <https://lists.internet2.edu/sympa/subscribe/perfsonar-ps-users>
 - <https://lists.internet2.edu/sympa/subscribe/performance-node-users>
- Announcements
 - <https://lists.internet2.edu/sympa/subscribe/perfsonar-ps-announce>
 - <https://lists.internet2.edu/sympa/subscribe/performance-node-announce>
- Working Groups
 - <https://lists.internet2.edu/sympa/subscribe/performance-wg>
 - <https://lists.internet2.edu/sympa/subscribe/is-wg>
 - <http://www.ogf.org/mailman/listinfo/nm-wg>
 - <http://www.ogf.org/mailman/listinfo/nmc-wg>
 - <http://www.ogf.org/mailman/listinfo/nml-wg>





perfSONAR

August 9th 2011, OSG Site Admin Workshop

Jason Zurawski – Internet2 Research Liaison

For more information, visit <http://www.internet2.edu/workshops/npw>

perfSONAR Adoption

- Networks
 - US National R&E Networks
 - ESnet, Internet2, NLR, NOAA
 - International R&E Networks/Exchange Points
 - APAN NOC, CSTNET, GEANT (and NREN Partners), Gloriad, JGN2, JPNet, KRNET, MANLAN, Starlight, TransPac2, RNP
 - US Regional R&E Networks
 - CENIC, CIC OmniPoP, Florida LambdaRail, Front Range GigaPoP, GPN, Indiana GigaPoP, LEARN, LONI, MAX, MCNC, Merit, MOREnet, Northern Lights GigaPoP, NOX, NYSERNet, OARnet, Pacific Northwest GigaPoP, SoX, UEN
- US Based Federal Labs/Facilities
 - ANL, BNL, DOE Headquarters, FNAL, LBNL, LLNL, MIT Lab for Nuclear Science, National Library of Medicine, NCAR, NCSA, NERSC, PNNL, PPPL, PSC, SLAC



perfSONAR Adoption

- International Sites
 - North America
 - Simon Frazier University (Canada), University of Western Ontario (London, Ontario, Canada), Camosun College (Canada)
 - South America
 - MonIPE - RNP (Rio de Janeiro, Brazil), Universidade Federal De Santa Catarina (Brazil), SPRACE (Brazil), UERJ (Brazil), Innova-Red (Buenos Aires, Argentina), UFSC (Florianopolis, Brazil), REUNA (Santiago, Chile), PUCP (Lima, Peru), MRREE (Lima, Peru), RAGIE2 (Universidad Mariano Galvez – Guatemala), UDESC (Brazil), UNIFACS (Salvador, Bahia, Brazil)
 - Europe/Middle East
 - Universitat Politecnica de Catalunya (Barcelona Spain), GEANT Affiliated NRENs (Europe), Northwestern (Doha, Qatar), Georgetown University (Doha, Qatar)
 - Africa
 - African Network Information Center, KRNET, UbuntuNet
 - Asia/Oceania
 - Chinese University of Hong Kong, Chonnam National University (South Korea), KISTI (Korea), Teritary Education Commission (Wellington, NZ), IHEP (China), Advanced Network Lab. at JNU (South Korea), Monash University (Melbourne, Australia), USTC ATLAS Tier-3 in Hefei (AnHui Province, China), NCHC (Taiwan), NICT (Japan), Thaisarn Nectec (Bangkok, Thailand)



perfSONAR Adoption

- Universities
 - Boise State University
 - Boston University *
 - California Institute of Technology **
 - The College of William and Mary
 - Colorado School of Mines
 - Colorado State University
 - Florida Atlantic University
 - George Mason University
 - Georgia Tech University
 - Georgetown University
 - Hat Creek Radio Observatory
 - Harvard University *
 - Hope College
 - Indiana University *
 - Indiana Purdue University *
 - Iowa State University
 - Johns Hopkins University
 - Leeward Community College
 - Louisiana State University
 - Massachusetts Institute of Technology (MIT) **
 - Merced County Office of Education
 - Michigan State University *
 - Middle Tennessee State University
 - North Dakota State University
 - Northwestern University
 - Oregon State University
 - Penn State University
 - Portland Community College
 - Purdue University **
 - Scripps College
 - Southern Methodist University *
 - Syracuse University
 - Texas A&M University
 - Tufts *
- Universities
 - University of California Irvine *
 - University of California Los Angles
 - University of California San Diego **
 - University of California Santa Barbara *
 - University of California Santa Cruz *
 - University of Chicago *
 - University of Connecticut
 - University of Delaware
 - University of Florida **
 - University of Hawaii
 - University of Houston
 - University of Illinois *
 - University of Maryland
 - University of Michigan *
 - University of Minnesota
 - University of Nebraska **
 - University of Northern Iowa
 - University of Oklahoma *
 - University of Oregon
 - University of Pennsylvania
 - University of South Dakota
 - University of South Florida
 - University of Texas *
 - University of Utah
 - University of Washington *
 - University of Wisconsin * **
 - Vanderbilt University **
 - Washington University in St. Louis
 - Wayne State University

* USATLAS

** USCMS



perfSONAR Adoption

- Commercial Networks/Organizations
 - AboveNet, BCNet, BBN, Cobham, EagleNet, KDL Inc., Northrop Gruman, Ocala Electric Company, Philadelphia Orchestra, Steelville Telephone Exchange, Viawest, Verizon
- Virtual Organizations
 - GENI, LIGO, LSST, MeasurementLab, REDDnet, USATLAS, USCMS
- Live pS Status:
 - Services: <http://www.perfsonar.net/activeServices>
 - Locations: <http://www.perfsonar.net/activeServices/IS>

Architecture Overview - Communities

- Communities = Web 2.0 Content Tagging
 - Think Flickr (tag your pictures with a category)
 - Think iTunes (tag your music with a genre)
- How does this help measurement lookup and discovery?
 - One more axis to search on
 - More human readable and understandable than IP address or hostnames
- Use as many (or as few) as required:
 - Networks (e.g. Campus, Regional, Network)
 - VO or Project (e.g. USATLAS, eVLBI, etc.)
 - Organization (DOE)
 - Other?

Architecture Overview - Communities

- Example: Some VO is setting up monitoring.
 - All sites want to test with each other
 - Not everyone is coming online at once, and VO membership may be volatile.
 - Strategy 1:
 - Central VO coordinator maintains a list of participants (and must update it often)
 - All monitoring is manual: add/remove test hosts when the list changes
 - Strategy 2:
 - VO recommends a tag for all new hosts
 - All VO members search for test hosts (periodically) that share this tag – N.B. the GUIs on the disk can organize this automatically

Architecture Overview - Communities

- Screenshot from the toolkit (when setting up the host):

The screenshot shows a web-based toolkit interface for managing host communities. At the top, a header reads "Communities [1] This Host Participates In". Below this, a table lists two communities: "Internet2" and "perfSONAR-PS", each with a "Delete" link. A blue "Add New Community" button is located below the table. The bottom section is titled "Popular Communities As Of 2009-09-22 08:02 (Click To Join)" and displays a word cloud of community names. The most prominent words in the cloud are "LHC", "USATLAS", and "Utah", with "Atlas", "CMS", "DOE", "DOE Sites", "DOE-SC-LAB", "ESnet", "GRNOC", "Internet2", "CTP", and "KREONET" also visible.

- Top: Communities the host has chosen to associate with
- Bottom: ‘Popular’ communities
 - The word cloud is based on what we found in the GLS – the larger the word = the more people that are using this classification

Architecture Overview - Communities

- List of hosts from the LHC community:

Test Members	
No Members In Test	
Add New Host	
Find Hosts To Test With	
Members Of LHC Community As Of 2009-09-22 08:02	
BWCTL Server at HEP, University of Pennsylvania in Philadelphia, PA, USA i2perf.hep.upenn.edu(128.91.45.144)	Add To Test
BWCTL Server at Internet2 in Washington, D.C., USA nms-rthr2-eth2.wash.net.internet2.edu(64.57.16.22)	Add To Test
BWCTL Server at Internet2 in Kansas City, MO, USA eth-2.nms-rthr2.kans.net.internet2.edu(64.57.16.214)	Add To Test
BWCTL Server at Internet2 in Houston, TX, USA eth-3.nms-rthr2.hous.net.internet2.edu(64.57.16.131)	Add To Test
BWCTL Server at Internet2 in New York, NY, USA nms-rthr2.newy32aoa.net.internet2.edu(64.57.17.66)	Add To Test



Client/Service Interaction

- EchoRequest
 - Sent to a service to test connectivity
 - Can be made arbitrarily complex by the service designer
 - Test backend storage
 - Test internal self-checks
 - Minimum is an ‘are you alive’ ping

Client/Service Interaction



Client sends EchoRequest to service to check liveness



Client/Service Interaction



Service sends Echo Response to verify



INTERNET
2

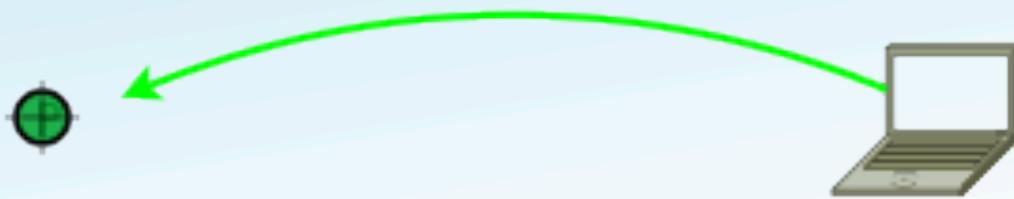
Client/Service Interaction

- MetadataKeyRequest
 - For a given (partial) metadata, ask the service to verify that it does or does not exist
 - Return a ‘key’ , e.g. replayable token, to access the data

Client/Service Interaction

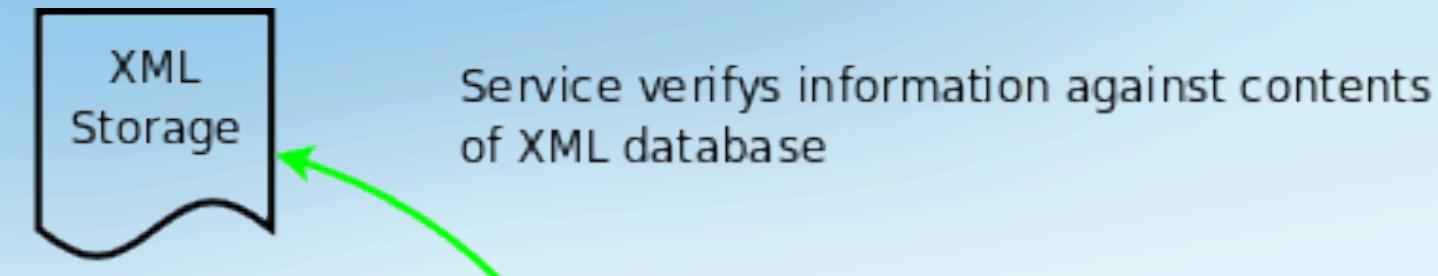


Client sends MetadataKeyRequest to check the status of a specific interface



INTERNET
2

Client/Service Interaction



Client/Service Interaction



XML database responds with results of the search - if successful a key will be supplied to the service to be returned to the client.



Client/Service Interaction



Service returns MetadataKeyResponse to client.



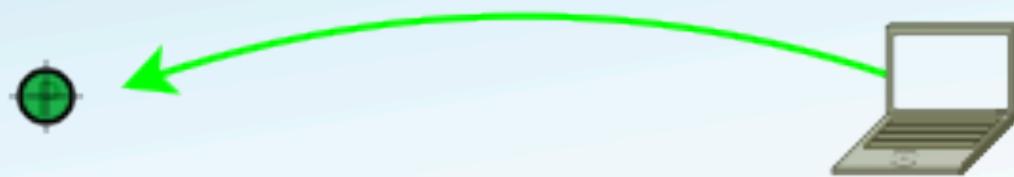
Client/Service Interaction

- SetupDataRequest
 - Given a key or (partial) metadata, return measurement information.
 - Can be ‘filtered’ by time to prevent getting more results than necessary.

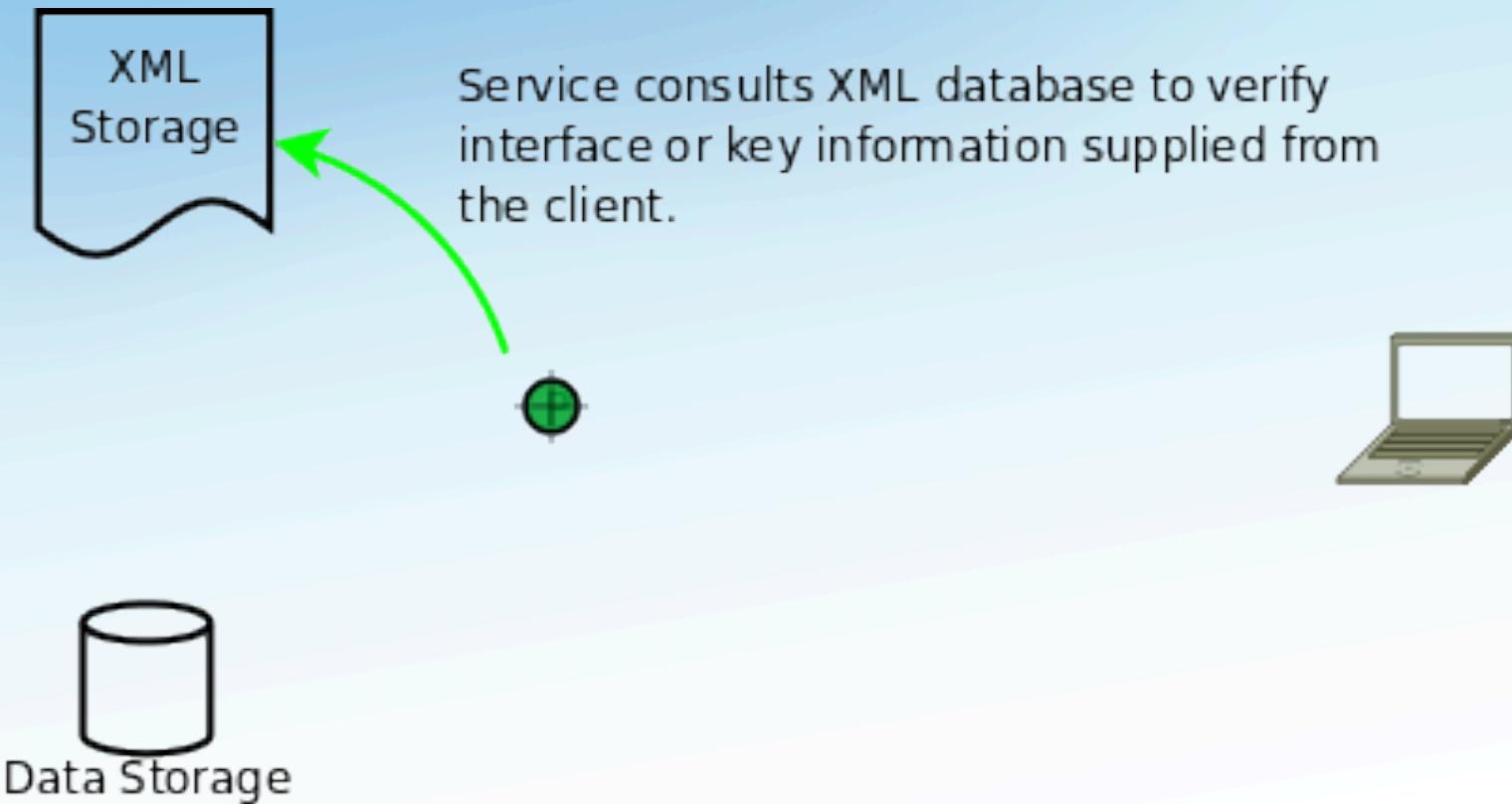
Client/Service Interaction



Client sends SetupDataRequest looking for interface data. Client may supply a key or metadata similar to a MetadataKeyRequest



Client/Service Interaction



Client/Service Interaction



XML database is able to verify the information for the service.

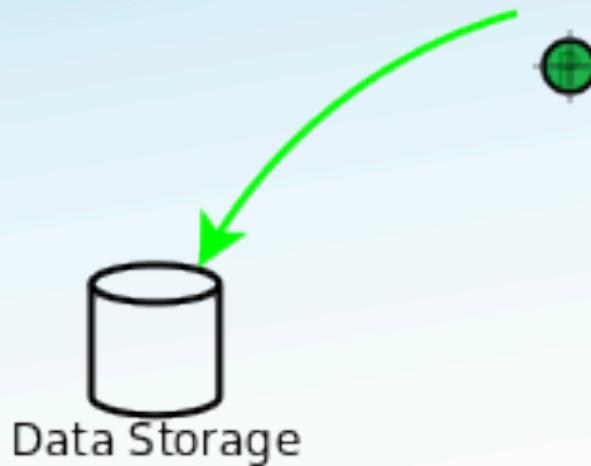


Data Storage

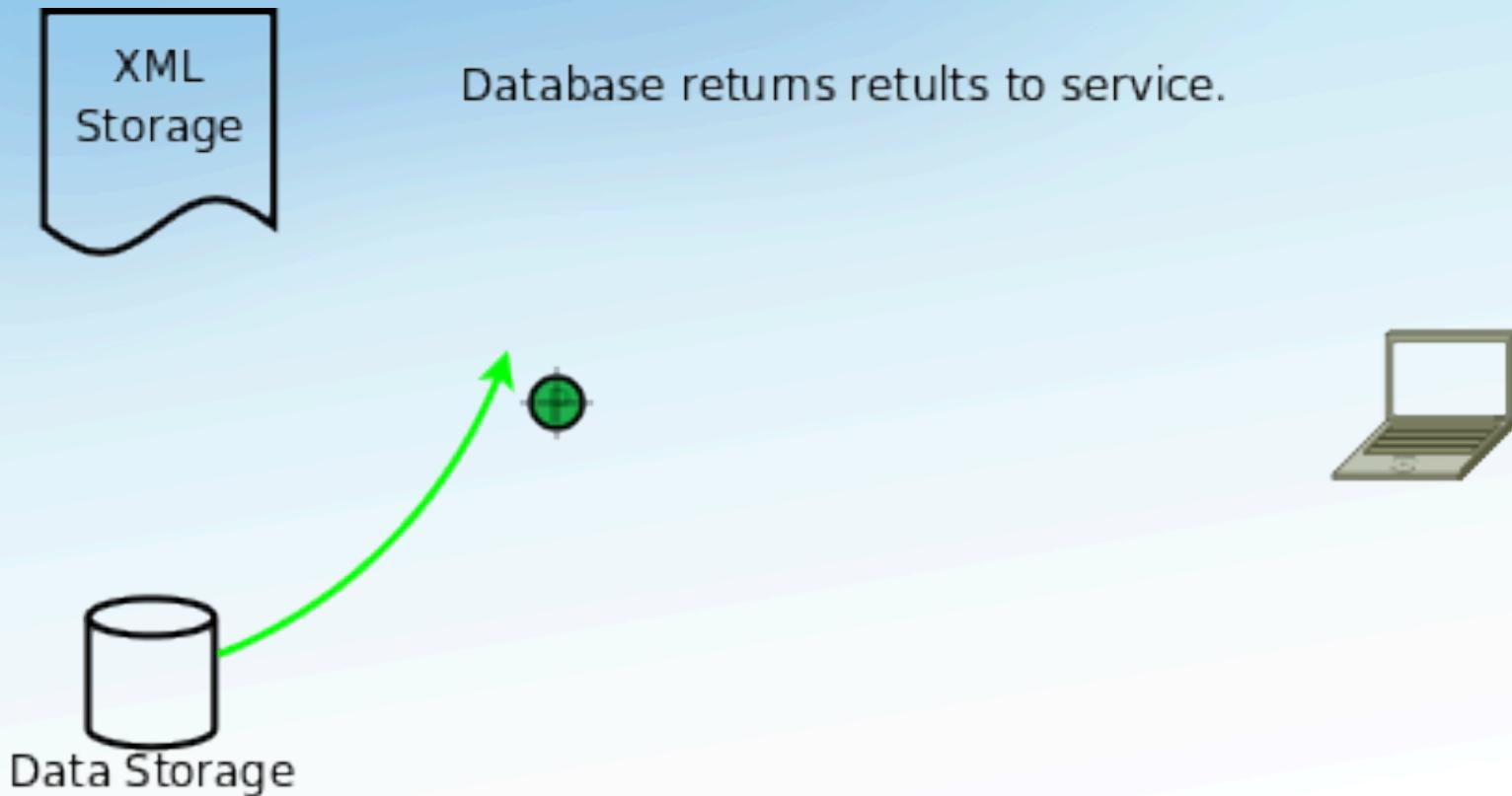
Client/Service Interaction



Service sends a specific query to the database backend. Query is database specific (e.g. RRD or SQL)



Client/Service Interaction



Client/Service Interaction



Service prepares SetupDataResponse for client.



Data Storage

