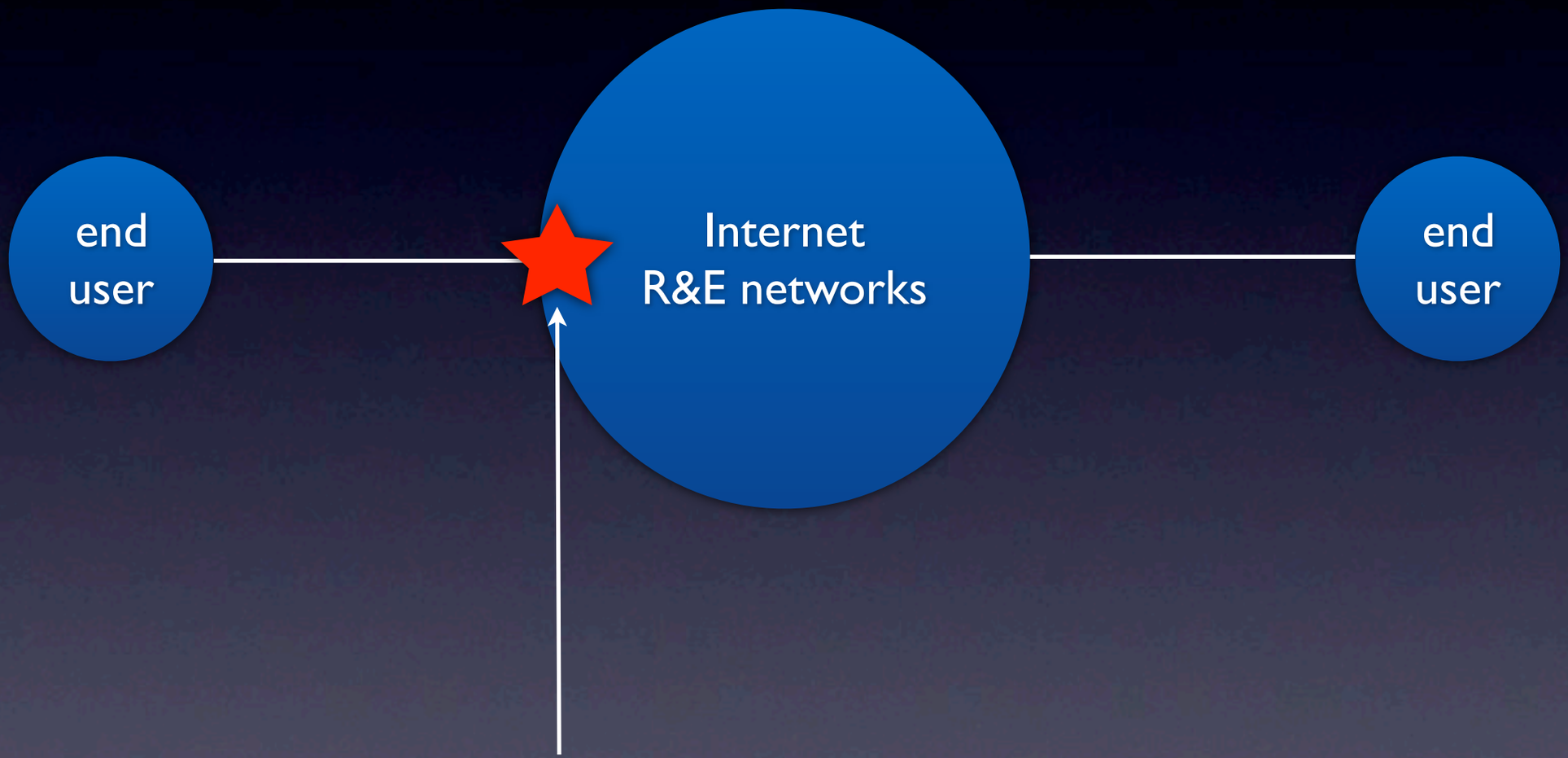
How NOC staff can help the users activities?

Yasuichi Kitamura (kita@jp.apan.net)
Jin Tanaka (tanaka@ote.kddi.com)
Yuichi Kurokawa (kurokawa@ote.kddi.com)
Takatoshi Ikeda(ikeda@ote.kddi.com)



You learned the technique here.

end user's interest

end user

communication performance

end user

- Detect the user's traffic.
- Performance between domains
- Other side issue

Detect the user's traffic.

- The user's traffic has reached you?
- The user's traffic can be transited at you?
- The user's traffic is transferred to the expected direction?

The user's traffic has reached you?

- Access network issue
  - End user side
    - Not quite sure who can be responsible
  - NOC side
    - you job
- Hosts issue

The user's traffic can be transited at you?

- your job
- router performance
- hardware issue
- routing information
- flow

The user's traffic is transferred to the expected direction?

- your job
- routing information
  - your side?
  - other sides?

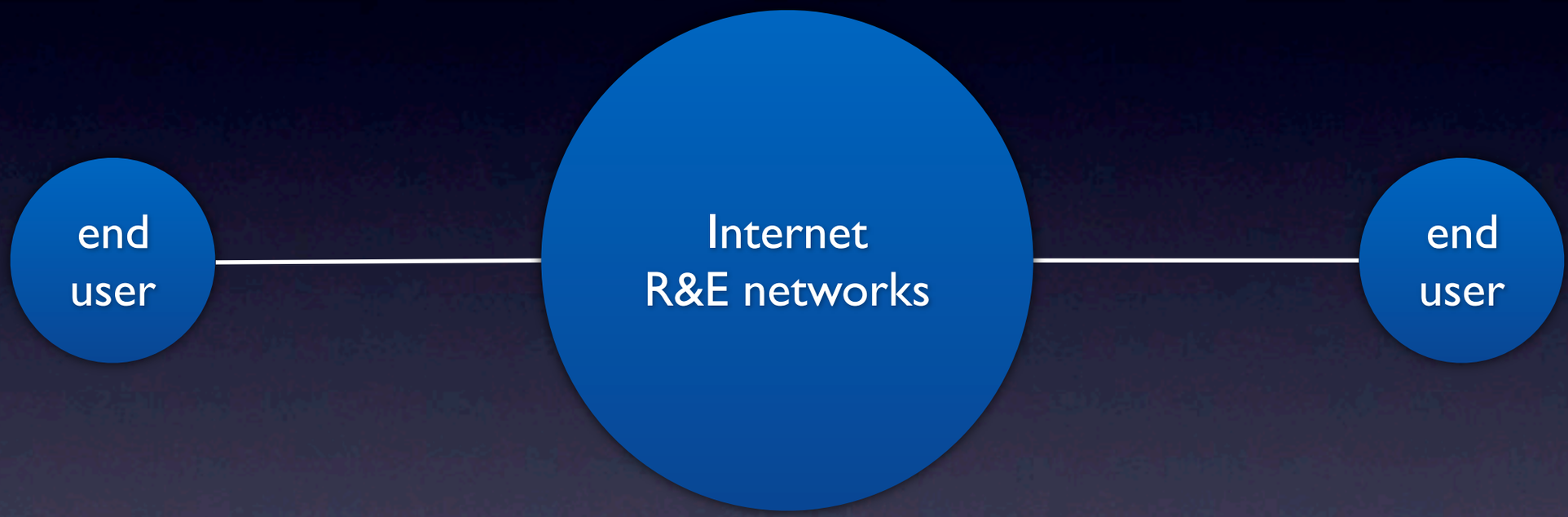
Performance between domains

- your job
- Available bandwidth
- Paths

Other side issue

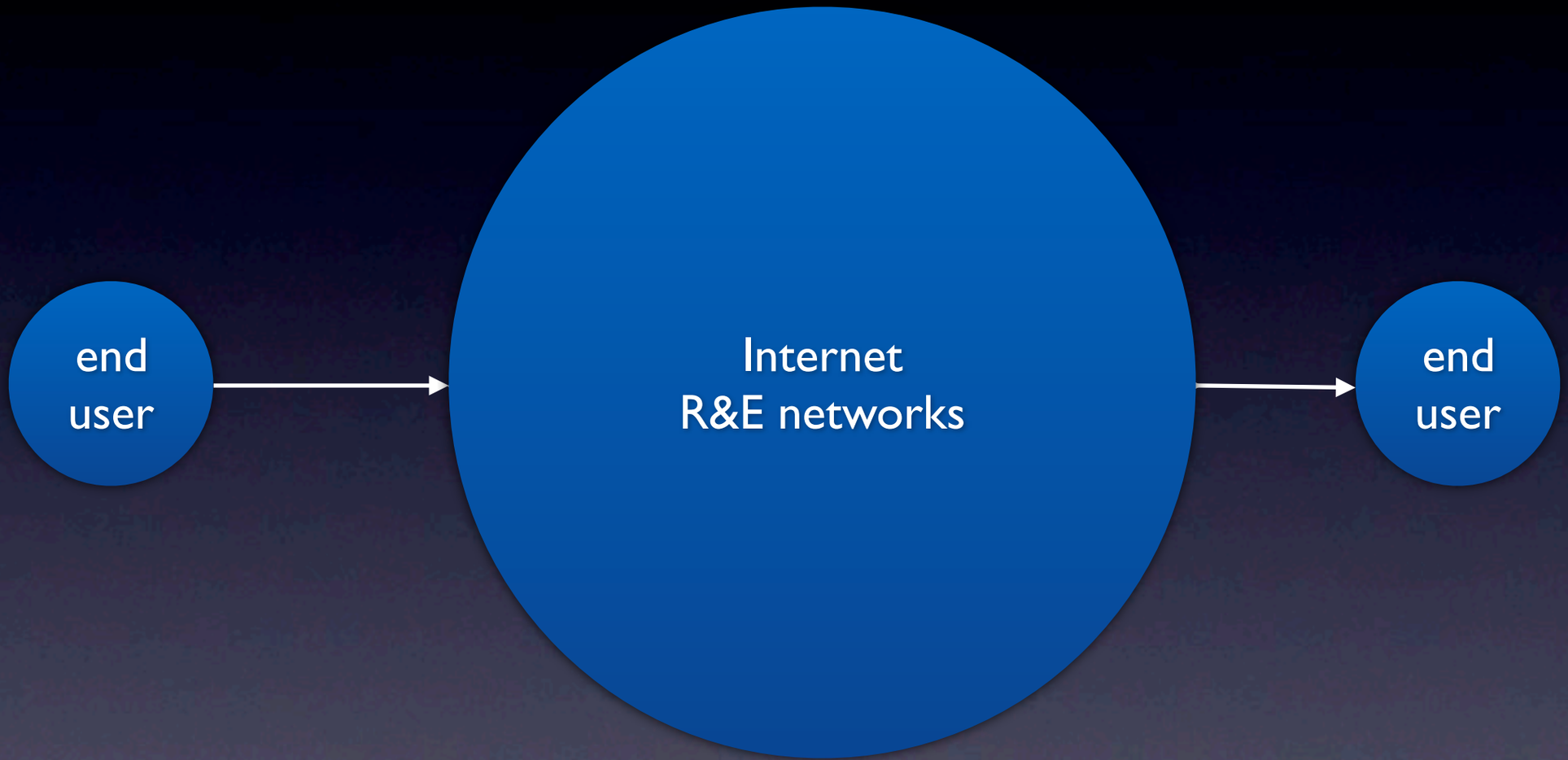
- NOC
  - How the other side is receiving your traffic?
  - How the other side is sending the return traffic?
- Access network
  - How the traffic reaches the end user?

Dynamic Circuit Network



- Internet part disappears.
- Direct connection is established virtually.

Compath



- The tool can detect if the packets are going through the expected routes.

How about others?

- ping
  - Most of the OSes have this command or the same function.
  - The target host is active?
  - Round Trip Time (RTT)
  - Packet loss rate
  - Timt To Live (TTL)

- ICMP/Ping Polling 2 -

- Optional Parameter
  - In case of daily operation
    - Packet size (byte)
    - Sending interval (sec)
    - Sending count (n)
    - Timeout (sec)
    - TTL (n)
    - Pattern (0x????)
    - etc.
  - At Monitoring system
    - Sending interval
    - Sending count
    - Time out

Set up the value which is adapted for critical level or service level!

Network Diagnostic Tool (NDT)

- Bottleneck Link Detection
- Duplex Mismatch Detection
- 2 clients are available.
  - command line
  - Web based client (Java applet)

NDT (Cont'd)

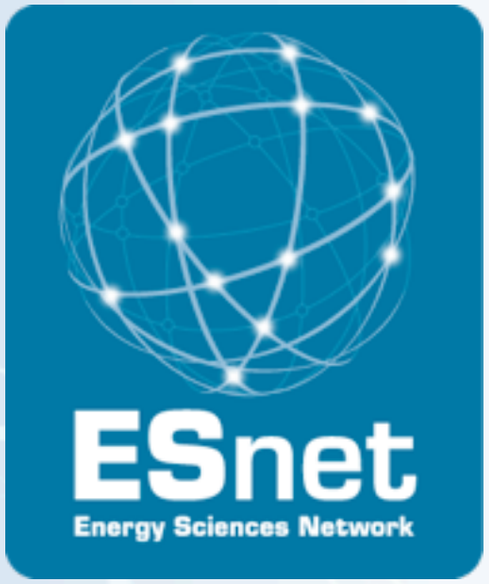
- How to use NDT?
  - Follow the installation guide of http://www.internet2.edu/pubs/ndt-cookbook.pdf
- Linux is required.
- Linux Kernel version sensitive

Netalyzr

- http://netalyzr.icsi.berkeley.edu/
- Java applet and no OS specification
- Tool for checking the end user side

Netalyzr (Cont'd)

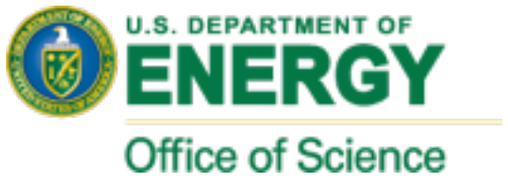
- Network address translation
- Network link properties
- Port filtering
- HTTP tests
- DNS tests
- Misc items
  - IPv6
  - clock drift



Iperf Tutorial

Jon Dugan <jdugan@es.net>

Summer JointTechs 2010, Columbus, OH



Outline

What are we measuring?

TCP Measurements

UDP Measurements

Useful tricks

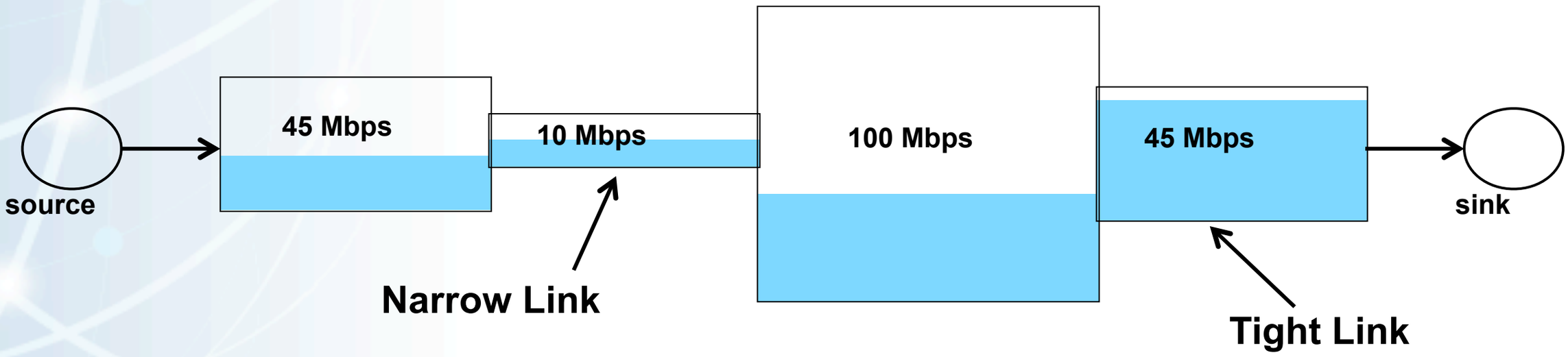
Iperf Development

What are we measuring?

Throughput? Bandwidth? What?

The term “throughput” is vague

- Capacity: link speed
  - Narrow Link: link with the lowest capacity along a path
  - Capacity of the end-to-end path = capacity of the narrow link
- Utilized bandwidth: current traffic load
- Available bandwidth: capacity – utilized bandwidth
  - Tight Link: link with the least available bandwidth in a path
- Achievable bandwidth: includes protocol and host issues



(Shaded portion shows background traffic)

Iperf data flow

Client is the sender
(data source)

Server is the receiver
(data sink)

Iperf discards
the data

TCP Measurements

TCP Measurements

Measures TCP Achievable Bandwidth

- Measurement includes the end system
- Sometimes called “memory-to-memory” tests
- Set expectations for well coded application

Limits of what we can measure

- TCP hides details
- In hiding the details it can obscure what is causing errors

Many things can limit TCP throughput

- Loss
- Congestion
- Buffer Starvation
- Out of order delivery

Example Iperf TCP Invocation

Server (receiver):

```
$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.0.1.5 port 5001 connected with 10.0.1.10 port 60830
[  4]  0.0-10.0 sec  1.09 GBytes    933 Mbits/sec
[  4] local 10.0.1.5 port 5001 connected with 10.0.1.10 port 60831
[  4]  0.0-10.0 sec  1.08 GBytes    931 Mbits/sec
```

Client (sender):

```
$ iperf -c 10.0.1.5
------------------------------------------------------------
Client connecting to 10.0.1.5, TCP port 5001
TCP window size:  129 KByte (default)
------------------------------------------------------------
[  3] local 10.0.1.10 port 60830 connected with 10.0.1.5 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.2 sec  1.09 GBytes   913 Mbits/sec
```

TCP performance: window size

Use TCP auto tuning if possible

- Linux 2.6, Mac OS X 10.5, FreeBSD 7.x, and Windows Vista

The –w option for Iperf can be used to request a particular buffer size.

- Use this if your OS doesn't have TCP auto tuning
- This sets both send and receive buffer size.
- The OS may need to be tweaked to allow buffers of sufficient size.
- See http://fasterdata.es.net/tuning.html for more details

Parallel transfers may help as well, the –P option can be used for this

To get full TCP performance the TCP window needs to be large enough to accommodate the Bandwidth Delay Product

TCP performance:
Bandwidth Delay Product

The amount of “in flight” data allowed for a TCP connection

BDP = bandwidth * round trip time

Example: 1Gb/s cross country, ~100ms

1,000,000,000 b/s * .1 s = 100,000,000 bits

100,000,000 / 8 = 12,500,000 bytes

12,500,000 bytes / (1024*1024) ~ 12MB

TCP performance: read/write buffer size

TCP breaks the stream into pieces transparently

Longer writes often improve performance

- Let TCP “do it’s thing”
- Fewer system calls

How?

- -l <size> (lower case ell)
- Example –l 128K

UDP doesn’t break up writes, don’t exceed Path MTU

TCP performance: parallel streams

Parallel streams can help in some situations

TCP attempts to be “fair” and conservative

- Sensitive to loss, but more streams hedge bet
- Circumventing fairness mechanism
  - 1 Iperf stream vs. n background: Iperf gets 1/(n+1)
  - x Iperf streams vs. n background: Iperf gets x/(n+x)
  - Example: 2 background, 1 Iperf stream: 1/3 = 33%
  - Example: 2 background, 8 Iperf streams: 8/10 = 80%

How?

- The –P option sets the number of streams to use
- There is a point of diminishing returns

TCP performance:
congestion control algorithm selection

Classic TCP (aka TCP Reno) is very conservative

Linux supports several different algorithms

- http://en.wikipedia.org/wiki/TCP_congestion_avoidance_algorithm
- CUBIC seems to work well for RE&E traffic flows

How?

- -Z allows the selection of a congestion control algorithm

UDP Measurements

UDP Measurements

UDP provides greater transparency

We can directly measure some things TCP hides

- Loss
- Jitter
- Out of order delivery

Use -b to specify target bandwidth

- Default is 1M
- Two sets of multipliers
  - K. m, g multipliers are 1000, 1000^2,1000^3
  - K, M, G multipliers are 1024, 1024^2,1024^3
- Eg, -b 1m is 1,000,000 bits per second

Example Iperf UDP Invocation

Server (receiver):

```
$ iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:   107 KByte (default)
------------------------------------------------------------
[  3] local 10.0.1.5 port 5001 connected with 10.0.1.10 port 65299
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  0.008 ms    0/  893 (0%)
```

Client (sender):

```
$ iperf -u -c 10.0.1.5 -b 1M
------------------------------------------------------------
Client connecting to 10.0.1.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 9.00 KByte (default)
------------------------------------------------------------
[  3] local 10.0.1.10 port 65300 connected with 10.0.1.5 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[  3] Server Report:
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  0.003 ms    0/  893 (0%)

[  3] Sent 893 datagrams
```

Useful tricks

Using Iperf to generate high rate streams

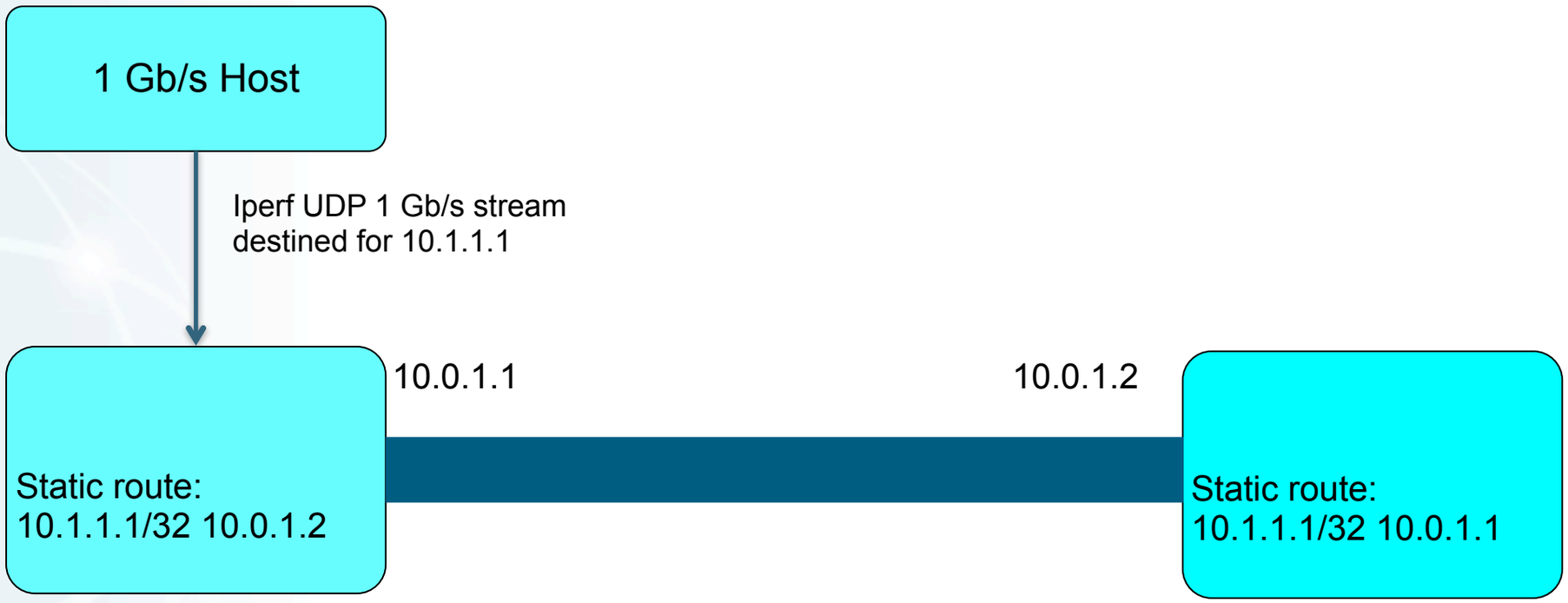
UDP doesn’t require a receiver

If you have good counters on your switches & routers those can be used to measure

Turns out UDP reception can be very resource intensive resulting in drops at the NIC at high rates (8-9 Gb/s)

Routing loops for fun and profit

Generate 10 Gb/s of traffic using a 1 Gb/s source host



Use the –T option to Iperf to control the number of times the traffic loops
Can also use firewall filters to discard a certain TTL range.
Other filters may be prudent as well.
Use firewall filters to count traffic on your router.

(Do not try this at home, the author is a highly insane network engineer.)

Iperf Development

Iperf Development
(Iperf is dead, long live Iperf)

ESnet

Iperf 2

- Iperf 2 is widely used
- Current version is 2.0.5 (released July 8, 2010)
- No further development, maintenance only
  - critical patches
  - sporadic releases, only when necessary

Iperf 3

- Currently in development
- Current version is 3.0b1 (released July 8, 2010)
- Weekly beta releases (starting this past Thursday)
- Eventually replace Iperf 2

Iperf 3: current status

Working

- TCP
- Control channel
  - Stream setup
  - Test parameter negotiation
  - Results Exchange
- Clean code!

Coming Soon

- UDP tests
- API with sane error reporting, library
- Timeline at: http://code.google.com/p/iperf/wiki/Iperf3Roadmap

More Information

Iperf 2:
http://sourceforge.net/projects/iperf/
Iperf 3:
http://code.google.com/p/iperf/

User Discussion:
iperf-users@lists.sourceforge.net
Developer Discussion:
iperf-dev@googlegroups.com

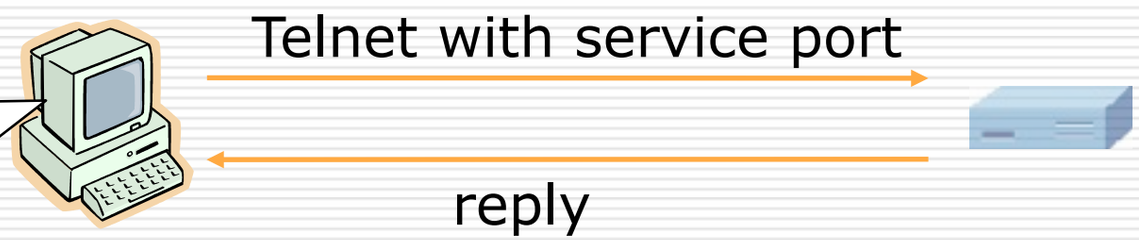
Network performance:
http://fasterdata.es.net/

Jon Dugan <jdugan@es.net>

UDP/TCP polling

- Effective in monitoring service ports of server
  - Using client for service
    - DNS - nslookup
  - Using telnet
    - WWW,SMTP,POP
  - Using tool
    - Radius - radping

```
bash-2.05$ telnet ns.jp.apan.net 80
Trying 203.181.248.3...
Connected to ns.jp.apan.net.
Escape character is '^]'.
get
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
<html><head>
<title>501 Method Not Implemented</title>
                    :
```



SNMP - Framework -

- SNMP: Simple Network Management Protocol
- Polling: UDP 161, Trap : UDP 162
- Protocol for monitoring/managing equipment via network
- Enables us to monitor the state and traffic of various equipment without being dependent on vender
- Management is realized by UDP between…
  - monitoring/managing server : Manager
    - e.g. HP Open View, Sun NNM
  - network equipment : Agent resides in device
    - e.g. Unix daemon、Cisco IOS
- Most general technique for acquiring detailed information from a router or a switch

SNMP - Version -

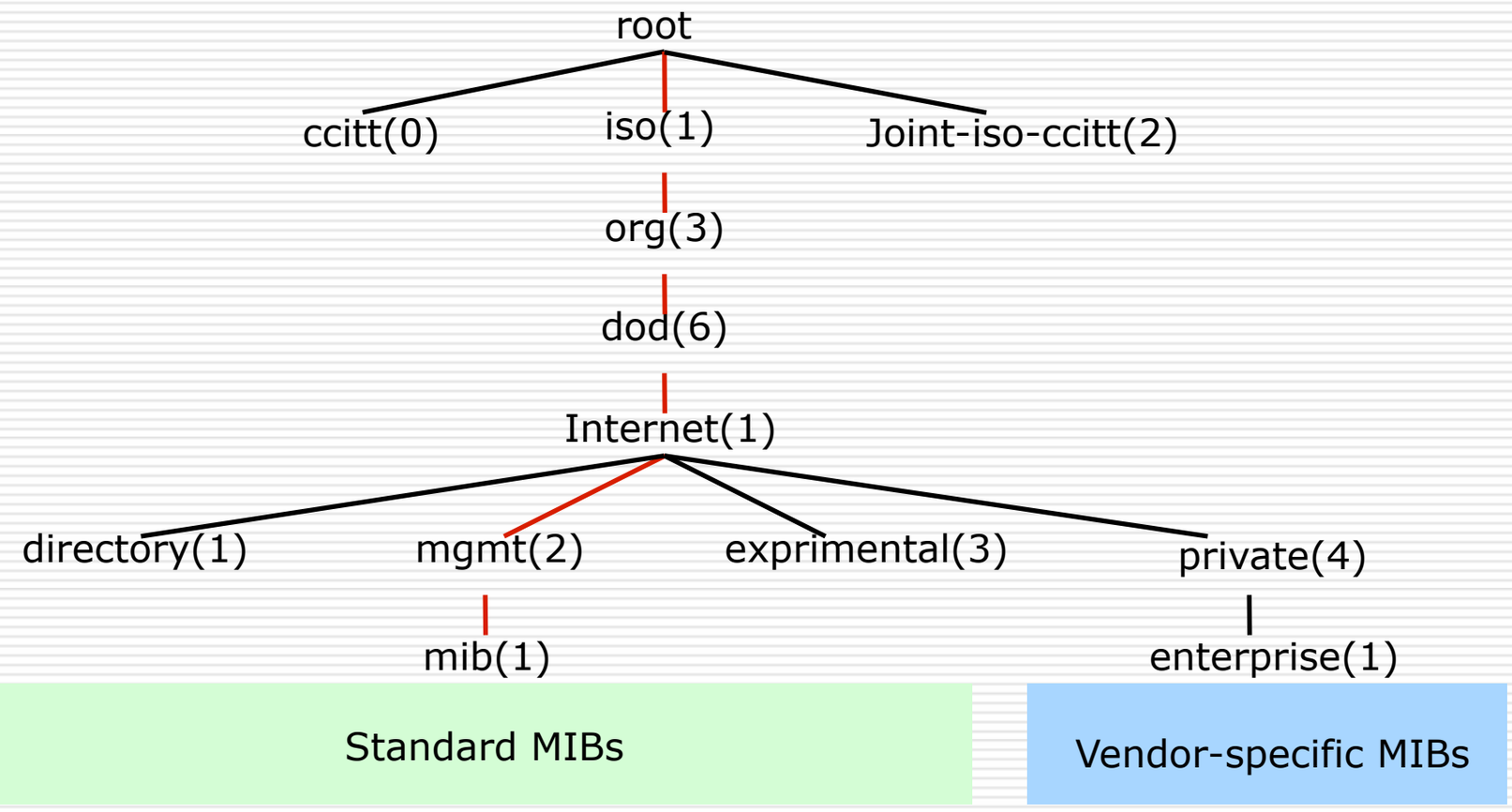
- SNMP v1 RFC1157
  - When Manager requests, Agent returns response
  - Agent sends trap when specific event has occurred
- SNMP v2 RFC1902
  - Basic of features are almost the same as those of v1
  - Additional regulation
    - 64bit counter：can deal with large numerical value
    - Get-bulk request： used to efficiently retrieve large blocks of data
    - Supports the use of encryption of messages
- SNMP v3 RFC2271～2275
  - Additional regulation
    - Various security function：MD5 user authentication,
      DES encryption

SNMP - MIB & OID -

- SNMP Manager can acquire the management information defined by MIB(Management Information Base) from Agent
  - Current version : MIBv2 RFC 1213
  - MIB is the aggregate of object (information) on the equipment which SNMP Agent holds
  - Identifier is defined for each object = OID
  - MIB performed by Agent is roughly divided into:
    - MIBv2 : standard, public, specified by IETF
    - Enterprise MIB : private, specified by vendor company

SNMP - MIB Tree -

- Objects are managed by the tree
- Expressed in a row of values divided by the period



SNMP - OID -

- OID Expression
  - iso(1). org(3). dod(6). internet(1). mgmt(2). mib2(1)

    -> .1.3.6.1.2.1

    e.g. sysDscr = .1.3.6.1.2.1.1.1 = mib-2.1.1 = system.1

Subtree Name	OID	Description
system	1.3.6.1.2.1.1	Defines a list of objects that pertain to system operation, such as the system uptime, system contact, and system name.
interfaces	1.3.6.1.2.1.2	Keeps track of the status of each interface on a managed entity. The interfaces group monitors which interfaces are up or down and tracks such things as octets sent and received, errors and discards, etc.
at	1.3.6.1.2.1.3	The address translation (at) group is deprecated and is provided only for backward compatibility. It will probably be dropped from MIB-III.
ip	1.3.6.1.2.1.4	Keeps track of many aspects of IP, including IP routing.
icmp	1.3.6.1.2.1.5	Tracks things such as ICMP errors, discards, etc.
tcp	1.3.6.1.2.1.6	Tracks, among other things, the state of the TCP connection (e.g., closed, listen, synSent, etc.).
udp	1.3.6.1.2.1.7	Tracks UDP statistics, datagrams in and out, etc.
egp	1.3.6.1.2.1.8	Tracks various statistics about EGP and keeps an EGP neighbor table.
transmission	1.3.6.1.2.1.10	There are currently no objects defined for this group, but other media-specific MIBs are defined using this subtree.
snmp	1.3.6.1.2.1.11	Measures the performance of the underlying SNMP implementation on the managed entity and tracks things such as the number of SNMP packets sent and received.

SNMP - SNMP Message -

- SNMP version : Check the version of SNMP(0 is for version 1)
- Community : Password between Manager and Agent
- PDU (Protocol Data Unit) : Actual command
  - **Manager -> Agent**
    - **GetRequest**
      - Used to request the values of one or more MIB variables
    - **GetNextRequest**
      - Used to read the values of variables in the MIB sequentially. It is often used to read through a table of values. After reading the Getrequest,

SNMP - SNMP Message Handling 1 -

SNMP Manager

SNMP Agent

GetRequest (What is the value of MIB?)

GetResponse (The value is XXXX!)

GetNextRequest
(What is the next value of MIB Tree ?)

GetResponse (The value is XXXX!)

SetRequest (Modify the value of OID)

GetResponse (The value is XXXX!)

Trap (Problem happened!)

SNMP - SNMP Message Handling 2 -

Command examples

GetRequest

```
inetapan@tools:~> snmpget -v2c -c xxxx tpr2.jp.apan.net  .1.3.6.1.2.1.2.2.1.4.136
IF-MIB::ifMtu.136 = INTEGER: 9192
```

GetNextRequest

```
inetapan@tools:~> snmpget -v2c -c xxxx tpr2.jp.apan.net system
SNMPv2-MIB::system = No Such Object available on this agent at this OID
inetapan@tools:~> snmpwalk -v2c -c xxxx tpr2.jp.apan.net system
SNMPv2-MIB::sysDescr.0 = STRING: m20 internet router, kernel 6.2R3.10
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.2636.1.1.1.2.2
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (423280751) 48 days, 23:46:47.51
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: tpr2
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 4
```

SetRequest

```
inetapan@tools:~> snmpset -v2c -c xxxx tppr.jp.apan.net system.sysLocation.0
system.sysLocation.0 = ""
inetapan@tools:~> snmpset -v2c -c yyyy tppr.jp.apan.net system.sysLocation.0 s  “Tokyo, JP “
system.sysLocation.0 =  “Tokyo, JP”
inetapan@tools:~> snmpset -v2c -c xxxx tppr.jp.apan.net system.sysLocation.0
system.sysLocation.0 =  “Tokyo, JP”
```

SNMP - Trap Message -

- The way for Agent to inform Manager about event of something undesirable
- Trap originates from Agent and is sent to the trap destination, as configured within Agent itself
- When Manager receives a trap, it needs to know how to interpret it
- PDU
  - **Enterprise**
    - vendor identification (OID) for the agent
  - **AgentAddress**
    - The IP address of the node where the trap was generated.
  - **Trap Type**
    - Generic / Specific (not used)
  - **Timestamp**
    - The length of time between the last re-initialization of the agent that issued a trap and the moment at which the trap was issued

Monitoring Software - HP OpenView -

- HP OpenView Network Node Manager ®

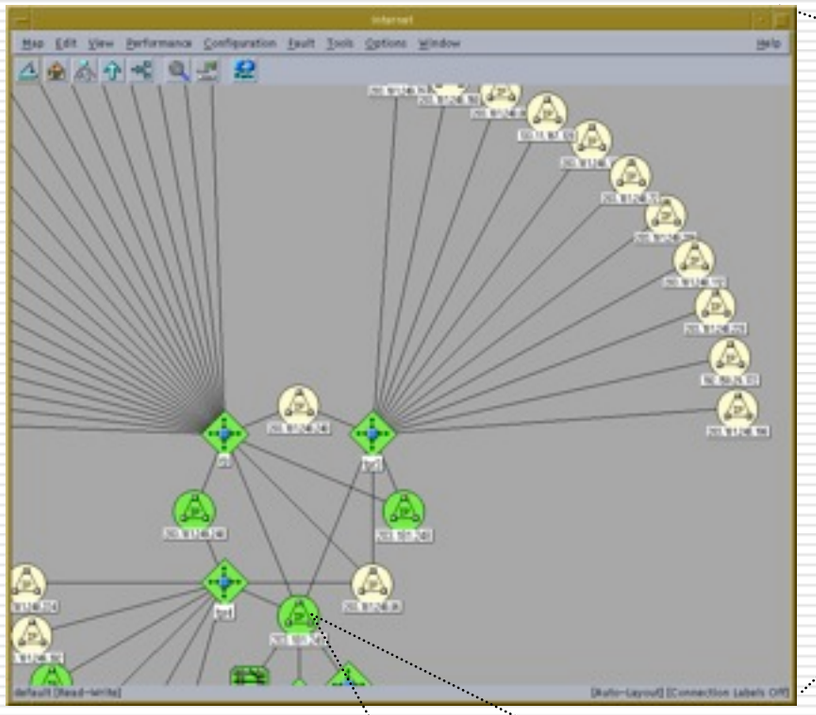
  http://www.openview.hp.com/products/nnm/index.html

- Overview
  - Auto discovery and mapping
  - Drill-down views (Hierarchy Map)
  - Fault monitoring : ICMP / SNMP polling
  - Event monitoring : Trap receiving/Event configuration
  - SNMP tools : Status polling
  - MIB Browser
  - Web-based reports
  - Extended software is enhanced
  - Platform : Windows 2000/XP, Solaris 8/9, HP-UX

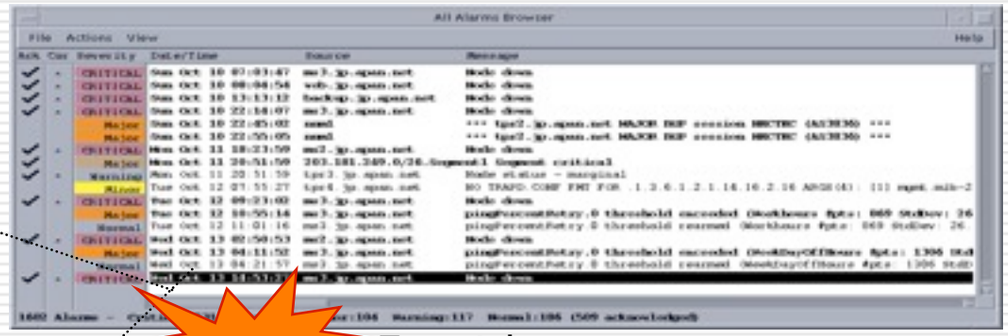
APAN-JP NOC monitors its network using OpenView mainly!

Monitoring Software
- HP OpenView Sample 1-

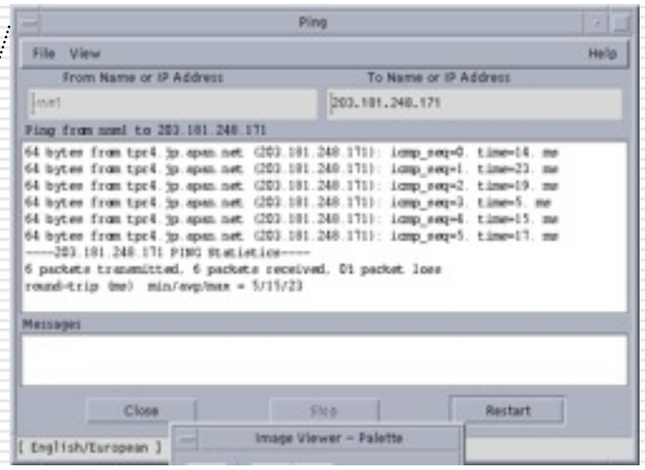
- OpenView Contracture



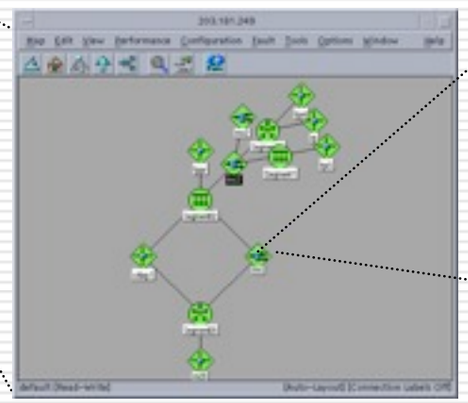
Network map



Event log



ICMP polling for connectivity check

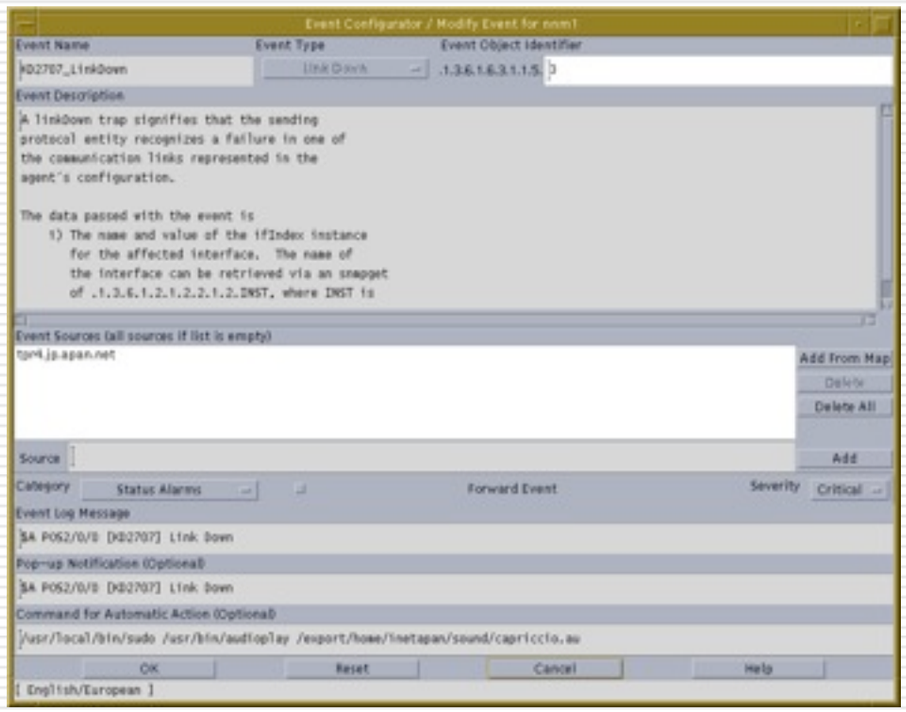


Network sub-map

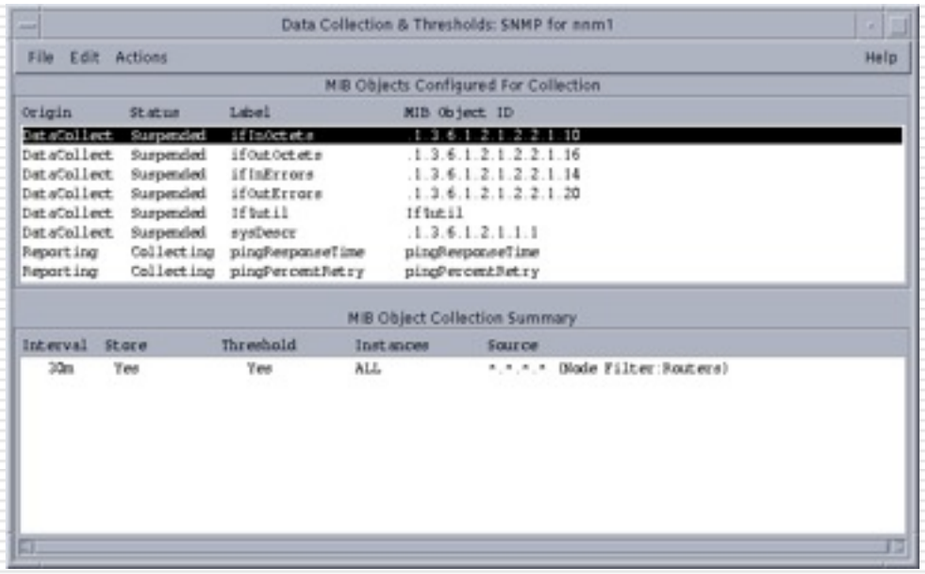
Router map

Monitoring Software
- HP OpenView Sample 2-

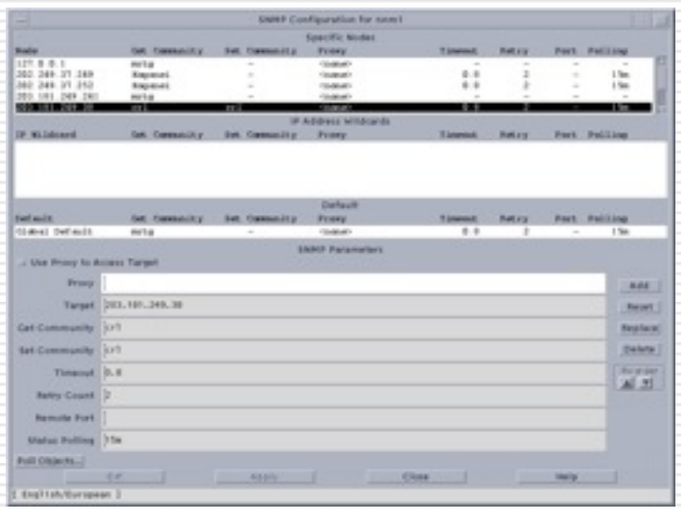
- OpenView Tools



Snmp configuration for polling
- parameters
- community



Event configuration



Data collection & Thresholds for SNMP

Monitoring Software
- Nagios Overview-

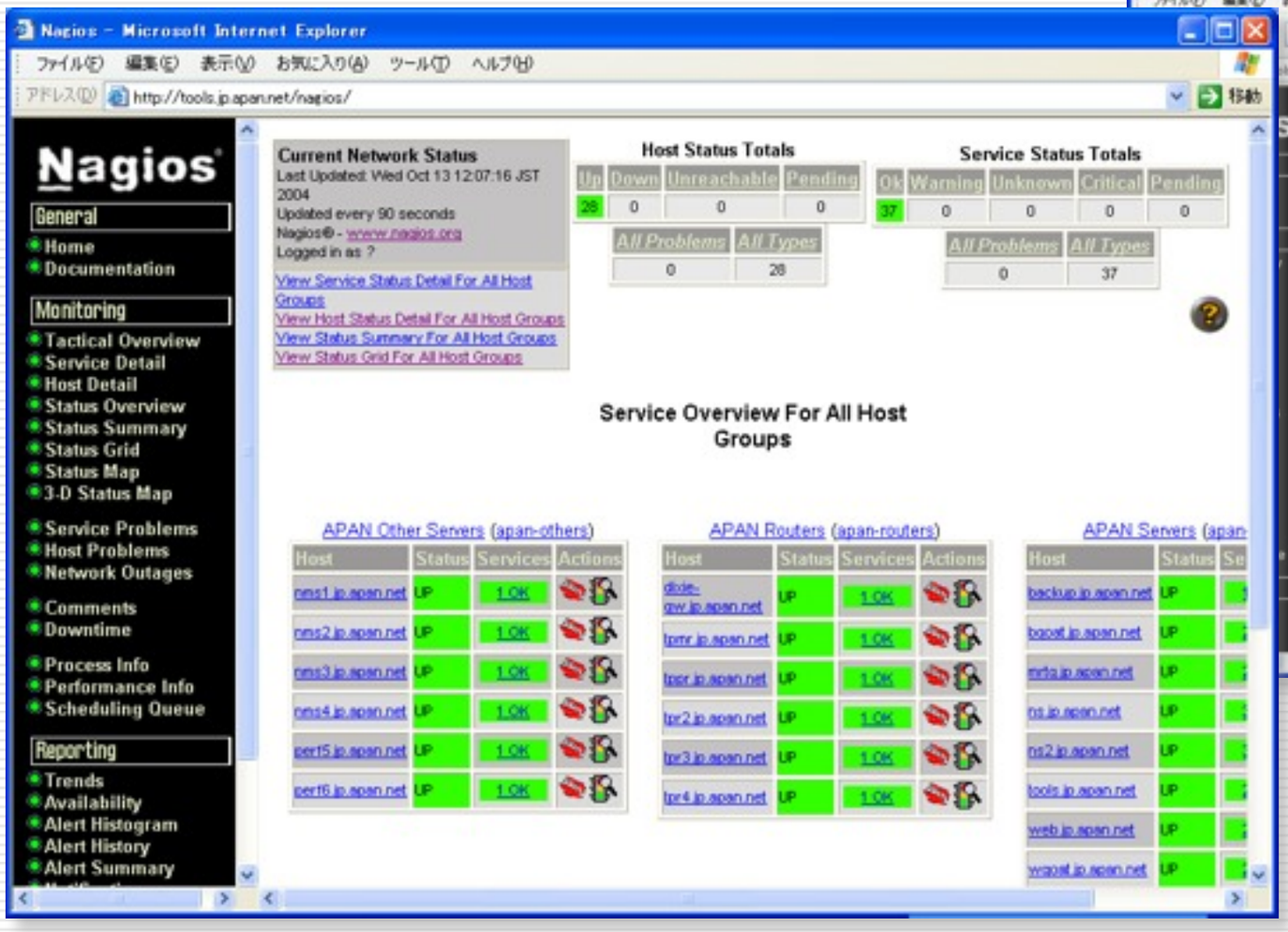
- Nagios ®
  - Freely available from http://www.nagios.org
- Overview
  - A host and service monitor designed to inform you of network and end system problems
  - Provides simple ping availability of resources on the network
  - Works with a set of “plugins” to provide local and remote host service status
  - Custom “plugins” are relatively easy to develop
  - Web-based monitoring system
  - Platform : Linux, UNIX

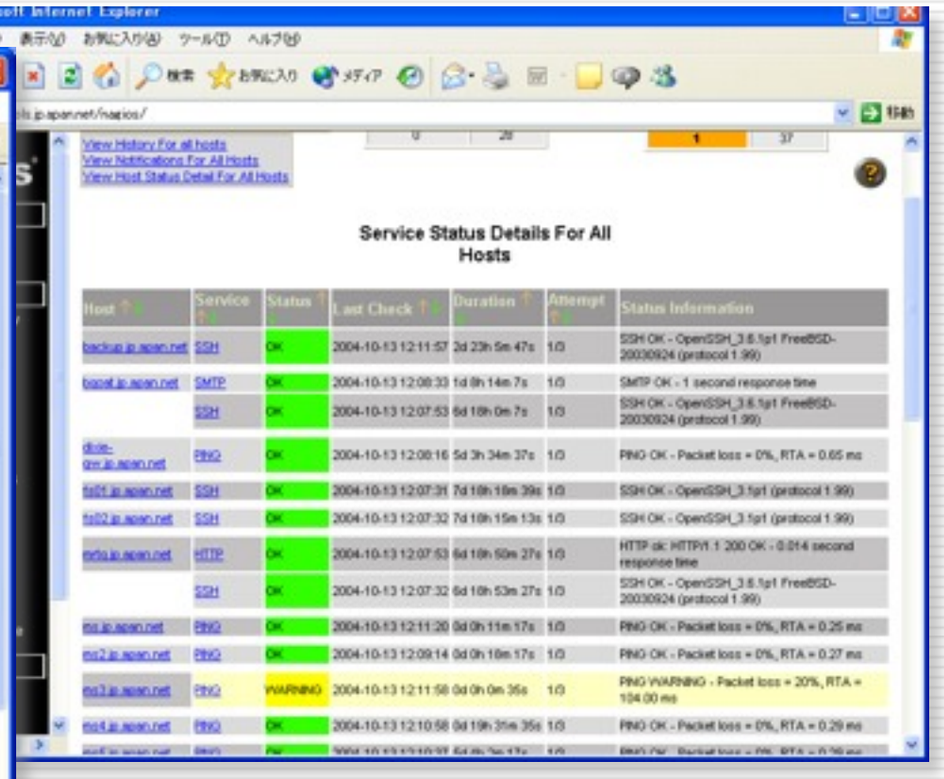
APAN-JP NOC uses Nagios as secondary monitoring system!

Monitoring Software
- Nagios Sample 1-

- Nagios

Service Overview For All Host Groups



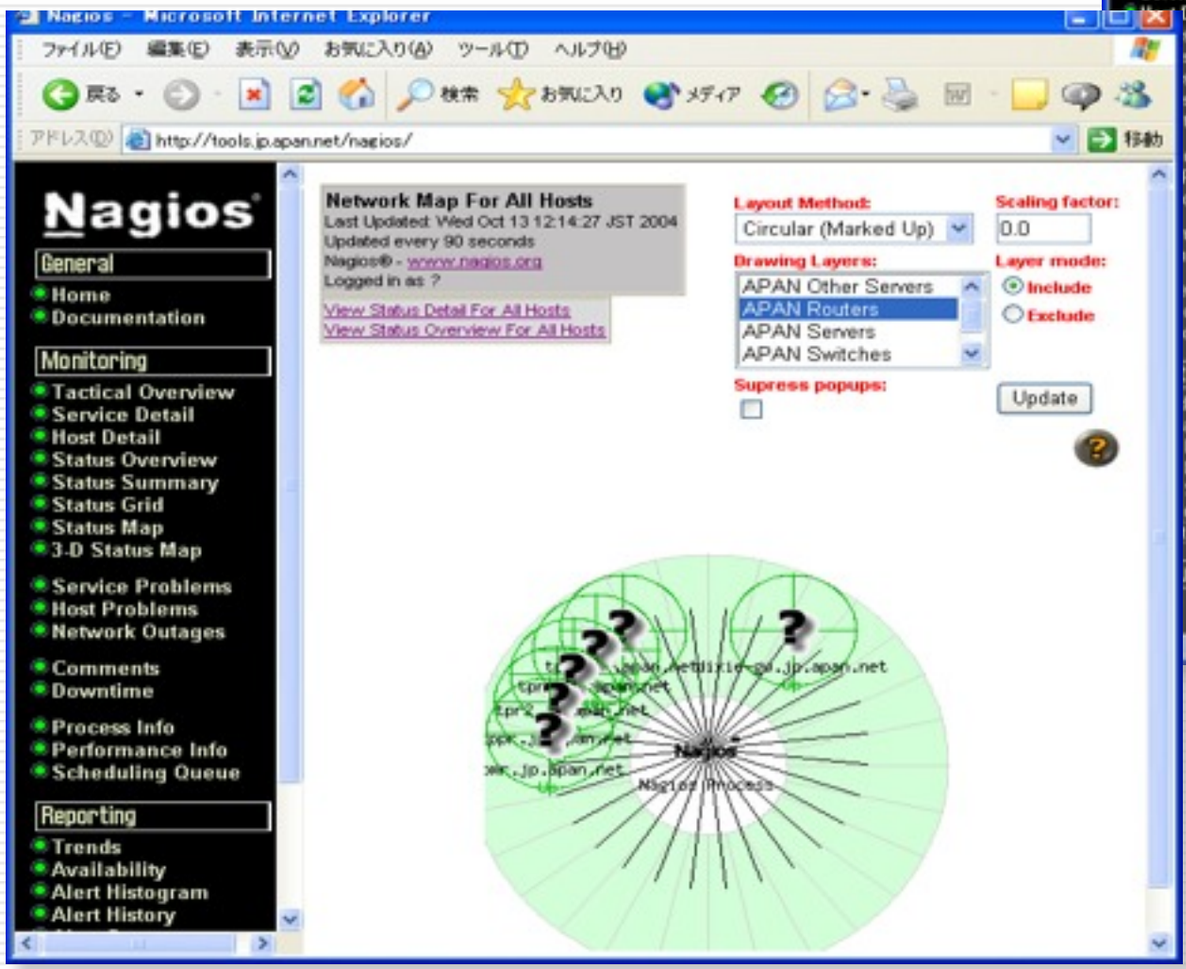


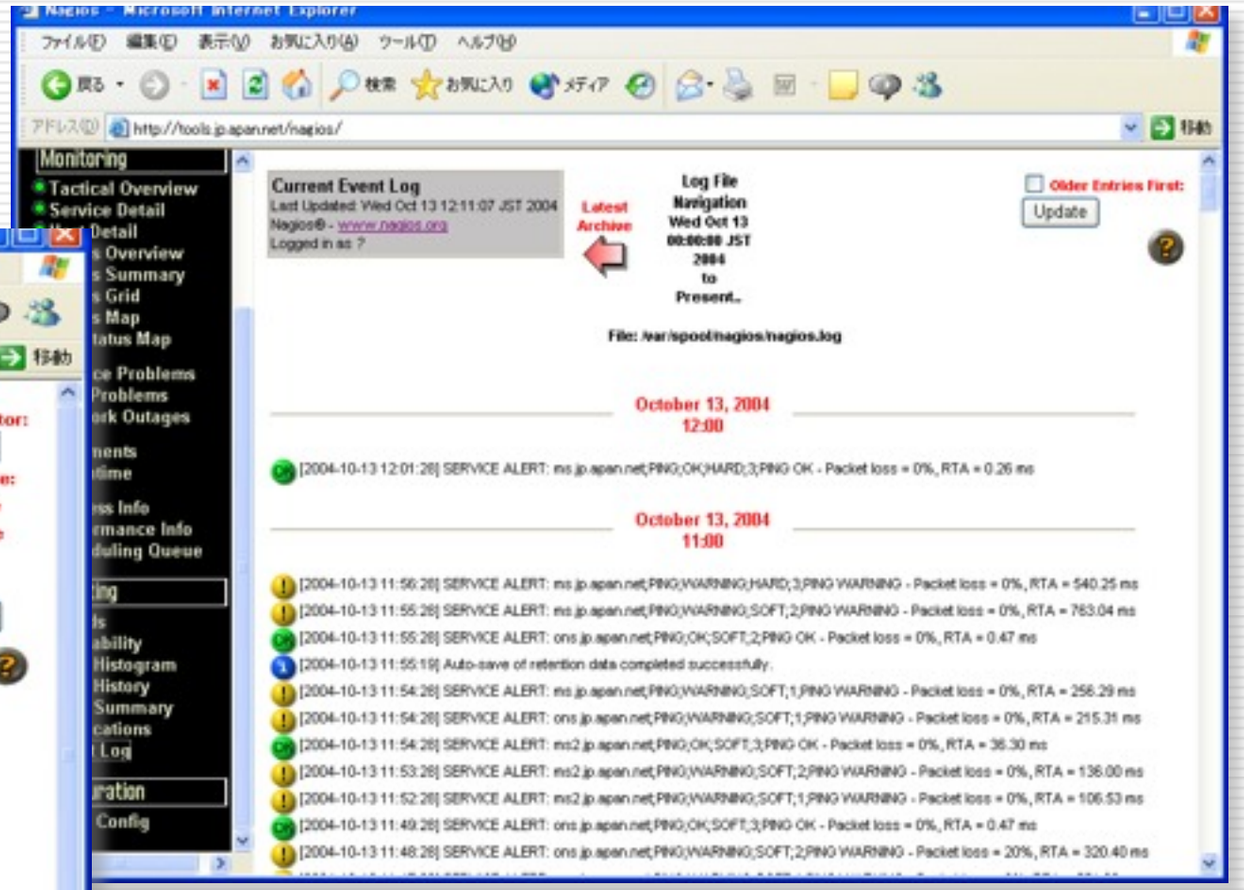
Service Status Details For All Hosts

Monitoring Software
- Nagios Sample 2-

- Nagios

Network Map For All Hosts





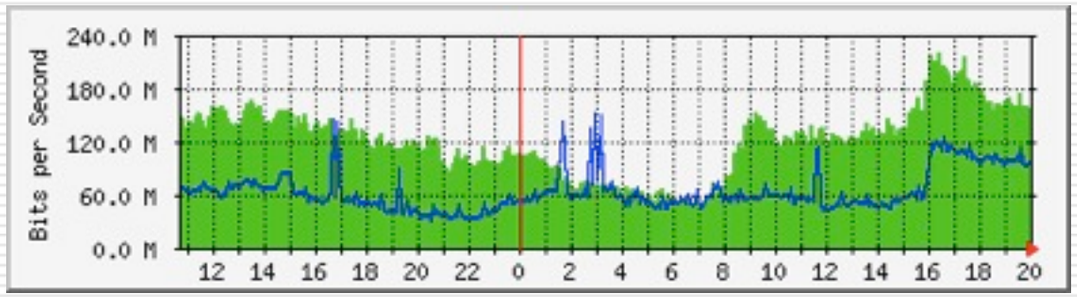
Event log

MRTG (Multi-Router Traffic Grapher)

- Overview
  - Monitors the load of network equipment using SNMP, mainly used for creation of traffic graph
  - Excellent graphing tool developed by Tobias Oetiker
  - Plots graph with any two variables against time, It is graph-ized with PNG format on HTML page
  - Able to create scripts to feed data into MRTG
  - Implements data collection, image, web-page collection
  - Very widely deployed in large networks and still being actively developed
  - Platform : UNIX system / Windows NT
  - Supports SNMPv2 : able to read 64bit counters
  - http://people.ee.ethz.ch/~oetiker/webtools/mrtg/

MRTG - Workflow -

- Display of graph



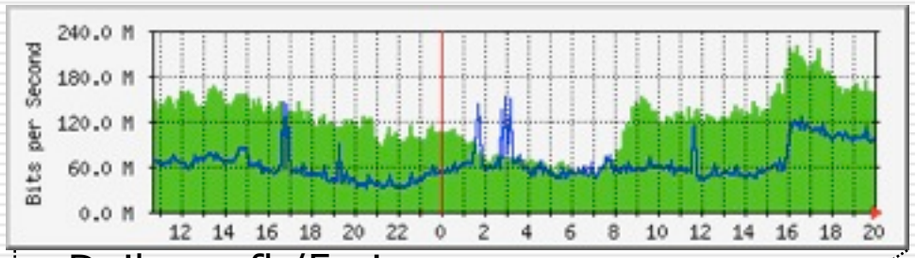
- Green area typically represents incoming maximum bits per second
- Blue line typically represents outgoing maximum bits per second

- Workflow

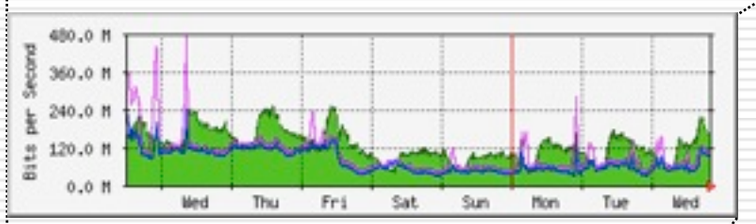
1. Read configuration file
2. Collect graphing data from network equipment, based on configuration
3. Update database file and generate graph
4. If required, generate HTML file

- MRTG performs above workflow then completes
- Since MRTG collects data of the past 5 minutes (default value of source code), it is desirable to set “crontab” for every 5 minutes

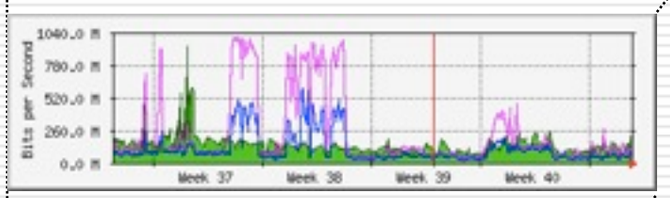
MRTG - Data Storage -



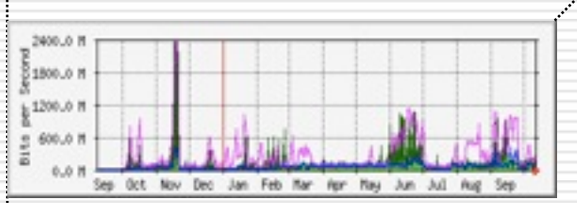
Daily grafh/5min



Weekly grafh/30min



Monthly grafh/2hours



Yearly grafh/1day

Rougher Resolution

- Data Storage
  - Keeps 5 minute data only for 2.5 days. The data is thrown away afterward.
    - There is no referring to historical data with high resolution
    - Keeps 1-day data for approx. 2 years

Interval	Num of record	Storage period	Graph
5 minutes	600	2.5 days	daily
30 minutes	600	12.5 days	Weekly
2 hours	600	50 days	Monthly
1 day	731	2 years	Yearly

MRTG - Configuration 1 -

- MRTG Configuration
  - cfgmaker
    - Helps to create configuration file form
    - Example

```
cfgmaker -global ‘WorkDir: /home/httpd/html/mrtg’ \
        -global "Options[_]: bits,growright’ \
        -output /home/httpd/html/mrtg/cfg/mrtg.cfg \n
       community@router.domain.name
```

Graph & log data: /home/httpd/html/mrtg

Configuration file: /home/https/html/cfg/mrtg.cfg

Option : unit = bits(bps), Horizontal axis = grow right way

- Detailed information

http://people.ee.ethz.ch/~oetiker/webtools/mrtg/cfgmaker.html

MRTG - Configuration 2 -

- Target Configuration
  - Target Expression
    - Target[<target name>]:<target kind>:<community>@<address>
      - <target name> : Identify equipment
      - <target kind> : Measurement item
      - <community> : SNMP community string
      - <address> : Hostname or IP address of equipment
  - SNMP data collection specification method
    - Basic / Port (ifindex)

      Target[myrouter]: 2:public@wellfleet-fddi.ethz.ch
    - Explicit OIDs / MIB Variables

      Target[myrouter]: 1.3.6.1.2.1.2.2.1.14.1&1.3.6.1.2.1.2.2.1.20.1:public@myrouter

      Target[myrouter]: ifInErrors.1&ifOutErrors.1:public@myrouter

      You can use cfgmaker to generate references with the options -- ifref=?
    - ifref=ip: Interface by IP
    - ifref=descrf: Interface by Description
    - ifref=name: Interface by Name
    - ifref=eth: Interface by Ethernet Address

MRTG - Configuration 3 -

Example of Configuration

```
Target[la]: ifHCInOctets\so-2/0/0&ifHCOutOctets\so-2/0/0:xxxxxxx@tpr2.jp.apan.net:::::2
MaxBytes[la]: 300000000
Title[la]: Traffic Analysis of TransPAC LA Link
PageTop[la]: <H1>Traffic Analysis of TransPAC LA link</H1>
WithPeak[la]: ymw
Directory[la]: tpr2
Options[la]: bits, growright

Target[la-err]: ifInErrors\so-2/0/0&ifOutErrors\so-2/0/0:xxxxxxx@tpr2.jp.apan.net
MaxBytes[la-err]: 300000000
Title[la-err]: Packet Error for TransPAC LA link
PageTop[la-err]: <H1>Packet Error for TransPAC LA link</H1>
Directory[la-err]: tpr2
Options[la-err]: growright, integer, nopercent
YLegend[la-err]: Number of Error Packets
ShortLegend[la-err]: n
Legend1[la-err]: Number of Error Packets for Incoming Traffic
Legend2[la-err]: Number of Error Packets for Outgoing Traffic
Legend3[la-err]: Peak of Number of Error Packets for Incoming Traffic
Legend4[la-err]: Peak of Number of Error Packets for Outgoing Traffic
LegendI[la-err]:  In:
LegendO[la-err]:  Out:
WithPeak[la-err]: w
```

MRTG - Comments -

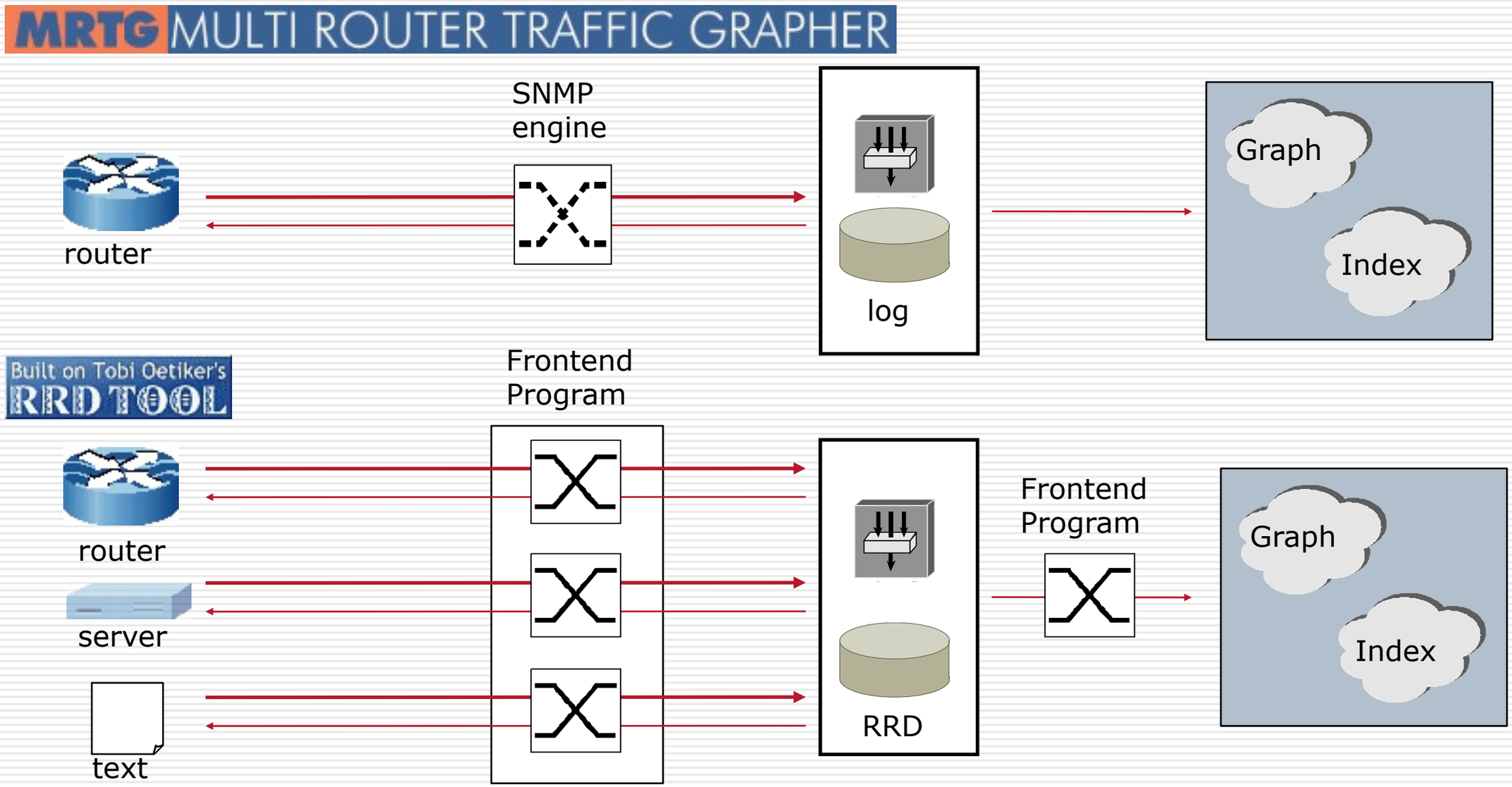
- Comments / Disadvantages
  - If you are to monitor a lot of devices (1000s), it is better to have a fast disk
  - If using external monitoring scripts, a fast processor and a lot of memory is necessary
  - Not particularly fast when compared to other data retrieval and storage schemes (Flat text files can slow down processing.)
  - MRTG can’t customize graphing periods
  - Flat text files are difficult to process when scripting against the data
  - Use 64bit counters with SNMPv2 for OC3-OC192 speed interface, GbE if it is 115Mbps traffic can wrap 32bit counters around in 5 minutes
  - MRTG can’t modify collected data which is summarized
  - Only two variables are available in processing a graph

RRDtool (Round Robin Database Tool)

- Overview
  - Successor to MRTG
  - Developed by the same developer of MRTG : Tobias Oetiker
  - Tool group for RRD can flexibly define data item, time interval, data amount, graph depiction, etc.
  - Binary file format that can store data at any interval for any length of time
    - File does not grow in size over time
  - Ability to make custom graphs across user-defined intervals
    - Ability to graph multiple variables on a single graph
  - Additional scripts are necessary in creating graphs and web-page
    - 25-30 percent faster than MRTG
  - Does not have the function to collect data
  - http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/

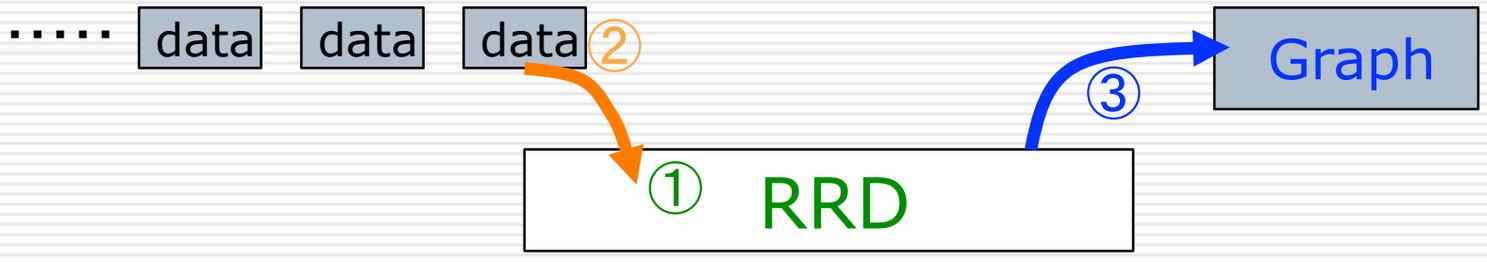
RRDtool - Architecture -

- Comparison of architecture between MRTG and RRD



RRDtool - Basic Usage -

- Basic usage of RRD tools
  - Set up new Round Robin Database (RRD) ・・・①
    - Define RRD used as vessel of data
    - Command : rrdtool create *filename*
  - Store new set of values into RRD periodically ・・・②
    - Write the data collected by frontend program in RRD
    - Command : rrdtool update *filename*
  - Generate Graph ・・・③
    - Create graph from data stored in one or several RRDs
    - Command : rrdtool graph *filename* (specify the graph name to generate)



RRDtool - Practice -

- Example
  - Object
    - Gigabit Ethernet Switch
  - Definition
    - Definition of RRD record

Interval	Num of RRD file	Storage Period	Graph
1 minute	360	6 hours	4 hours
5 minutes	576	2 days	Daily
2 hours	600	50 days	Monthly
1 day	731	2 years	Yearly
4 days	915	10 years	10 years

- Ability to describe peak graph from data of 1-day to 10-years

RRDtool - Create -

- Set up a new Round Robin Database (RRD)

Command Example

```
/usr/local/rrdtool-1.0.46/bin/rrdtool create \
/home/httpd/html/traffic/traffic_vlan.rrf \
–step 60 \
DS:vlan2in:counter60:0:125000000 \
DS:vlan2out:counter60:0:125000000 \
DS:vlan7in:counter60:0:125000000 \
DS:vlan7out:counter60:0:125000000 \
   :
RRA:AVERAGE:0.5:1:360 \
RRA:AVERAGE:0.5:5:576 \
RRA:AVERAGE:0.5:120:600 \
RRA:AVERAGE:0.5:1440:731 \
RRA:AVERAGE:0.5:5760:915 \
RRA:MAX:0.2:5:576 \
RRA:MAX:0.1:120:600 \
RRA:MAX:0.1:440:731 \
RRA:MAX:0.1:5760:915 \
```

- DS : Define the data item
  - COUNTER: continuous increasing counters
  - 60 : if no new data is supplied for more than 60 sec, it is considered as “unknown”
  - 0 : minimum acceptable value (byte)
  - 125000000 : maximum acceptable value (byte)
- RRA (Round Robin Archive) : Define the data consolidations
  - AVARAGE/MAX: average /maximum of consolidated of data
  - 0.5 : consolidation interval is be made up from *UNKNOWN* data while the consolidated value is still regarded as known.
    - Average 50%. MAX 20% or 10%
  - 1: consolidated data point where the data then goes into the archive
  - 360 : how many generations of data values are kept in RRA

RRDtool - Update -

- Stores a new set of values into RRD periodically
  - Data collection
    - Collect the data from targets using frontend program
      - Original tool
      - Cricket - http://cricket.sourceforge.net/
      - Orca - http://www.orcaware.com/orca/
      - SNAPP - http://sourceforge.net/projects/snapp/
  - Updating an RRD
    - Feed collected data into a RRD database using following commands

Command Example

```
rrdtool update /home/httpd/html/traffic/traffic_vlan.rrd
\
 --template in:out N:11222:1
```

- The name of the RRD you want to update.
- DS1: DS2
  The data sources are defined in the RRD
- 'N'=Update time is set to be the current time

RRDtool - Graph 1 -

Generating Graph -1-

Command Example

```
rrdtool graph /home/httpd/html/traffic/traffic.png -s -4h –w 800 –h 800 –a PNG \
 –t “VLAN Traffic” –v  “bit/s” \
DEF:vlan2in_ave=/home/httpd/html/traffic/traffic_vlan.rrd:vlan2in:AVERAGE \
DEF:vlan2out_ave=/home/httpd/html/traffic/traffic_vlan.rrd:vlan2out:AVERAGE \
DEF:vlan7in_ave=/home/httpd/html/traffic/traffic_vlan.rrd:vlan7out:AVERAGE \
DEF:vlan7in_ave=/home/httpd/html/traffic/traffic_vlan.rrd:vlan7out:AVERAGE \
CDEF:vlan2in_ave_bit=vlan2in_ave,8 * \
CDEF:vlan7in_ave_bit=vlan7in_ave,8 * \
CDEF:vlan2out_ave_bit=vlan2out_ave,-8 * \
CDEF:vlan7out_ave_bit=vlan7out_ave,-8 * \
AREA:vlan2in_ave_bit#ff5e5e:VLAN2-in \
STACK:vlan7in_ave_bit#5eff5e:VLAN7-in \
AREA:vlan2out_ave_bit#aa0101:VLAN2-out \
STACK:vlan7out_ave_bit#0101aa:VLAN7-out \
```

Options

-s: start time (default : seconds), -e: end seconds (default : seconds),
-w,h : width and height pixels, -a : image format GIF|PNG, -t : Graph title,
-v vertical-label text

RRDtool - Graph 2 -

- Generating a Graph -2-
  - DEF
    - Define virtual name for data source
      - DEF:<*vname*>=<*RRDfilename*>:<*DS-name*>:*CF*

        CF: consolidation function

        select AVARAGE, MAX, MIN, LAST (Newest data)
  - CDEF
    - Create new virtual data source by evaluating mathematical expression
      - CDEF:<*vname*>=*rpn-expression* (Reverse Polish Notation)
  - Graph depiction parameter
    - <Style>:<*vname*>#<*color*>:<*legend*>

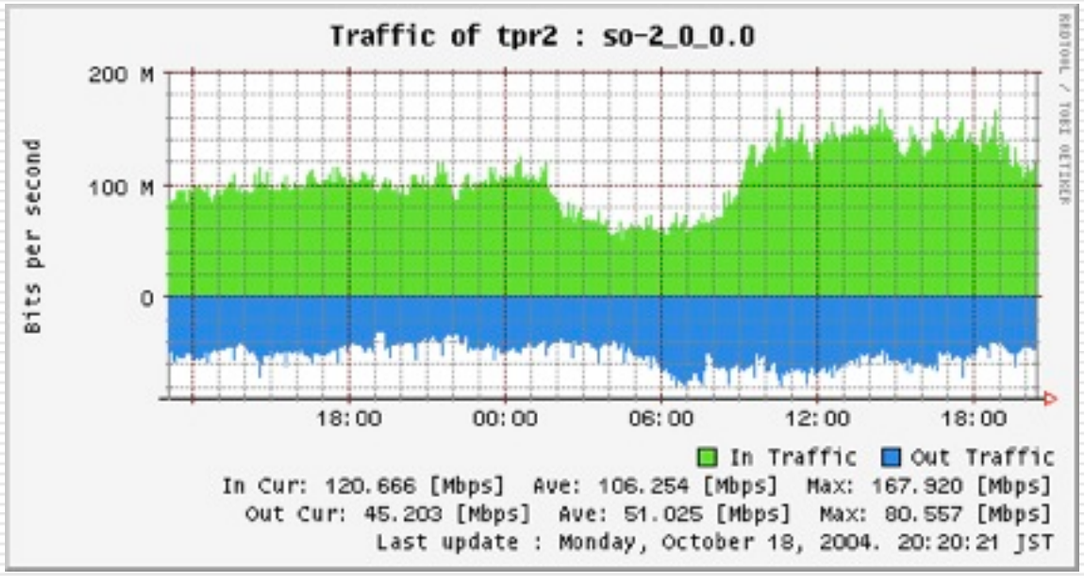
      LINE：Plot for the request data, using the color specified

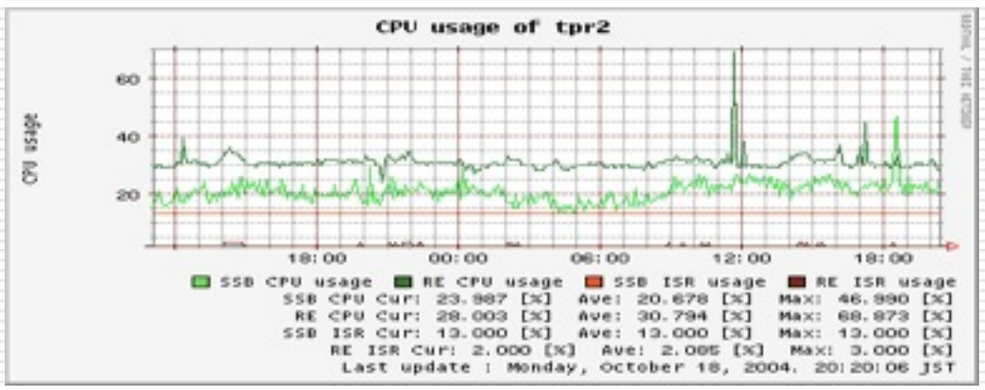
      AREA ：Area between 0 line and the graph line will be filled with the color specified

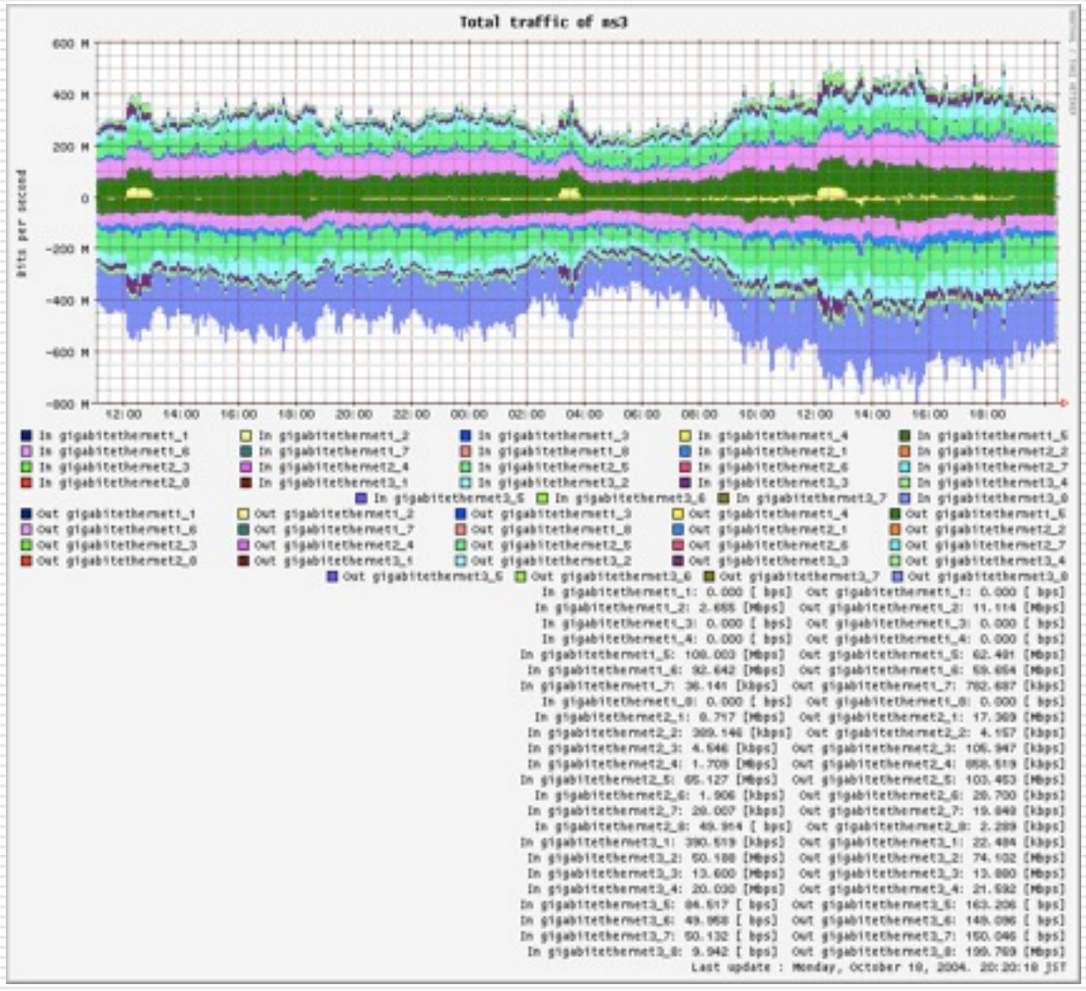
      STACK：Graph gets stacked on top of the previous LINE, AREA, or STACK graph
  - By updating graph generation periodically using “crontab”, you can see updated graphs on the Web

RRDtool - Sample -

- Sample Graph







http://mrtg.jp.apan.net/cricket/router-interfaces/

BWCTL (Bandwidth Control)

Users attempting to run bandwidth tests used to be not certain whether or not their test was scheduled in a time frame where other tests were not to run

- BWCTL is a resource allocation and scheduling daemon for arbitration of iperf tests
- BWCTL client application works by contacting a **bwctld** process on both endpoints of test systems
- Requires that NTP be running to synchronize the system clock
- Open mode：everyone can use
- Authentication mode：need to exchange AES key
- Support IPv6, Platform：UNIX systems
- Developed by Internet2 http://e2epi.internet2.edu/bwctl/

OWAMP (One-way Active Measurement Protocol)

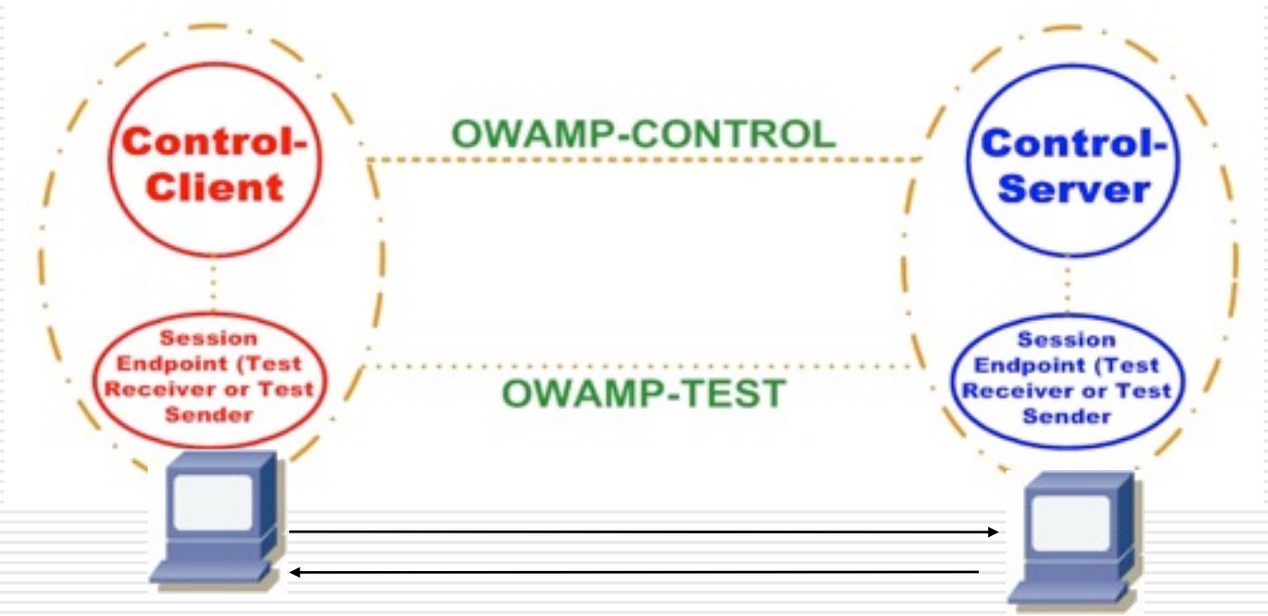
Roundtrip-based measurement can not identify the delay in each direction, especially when asymmetric routes are used

↓

- OWAMP is a command line client application and a policy daemon used to determine one way latencies between hosts
- It is possible to collect active measurement data
  - e.g., one-way delay, packet loss, jitter
- NTP must be setup correctly on the system to calculate a reasonable estimate of time error and to stabilize clock
- Support IPv6. Platform : UNIX systems
- Current Draft : draft-ietf-ippm-owdp-10.txt
- Developed by Internet2 http://e2epi.internet2.edu/owamp/

OWAMP - Protocol -

- Consists of two inter-related protocols
  - OWAMP-Control
    - Used to initiate, start/stop test sessions, and fetch test results
  - OWAMP-Test
    - Define the format of probe packet



- Sample measurement data

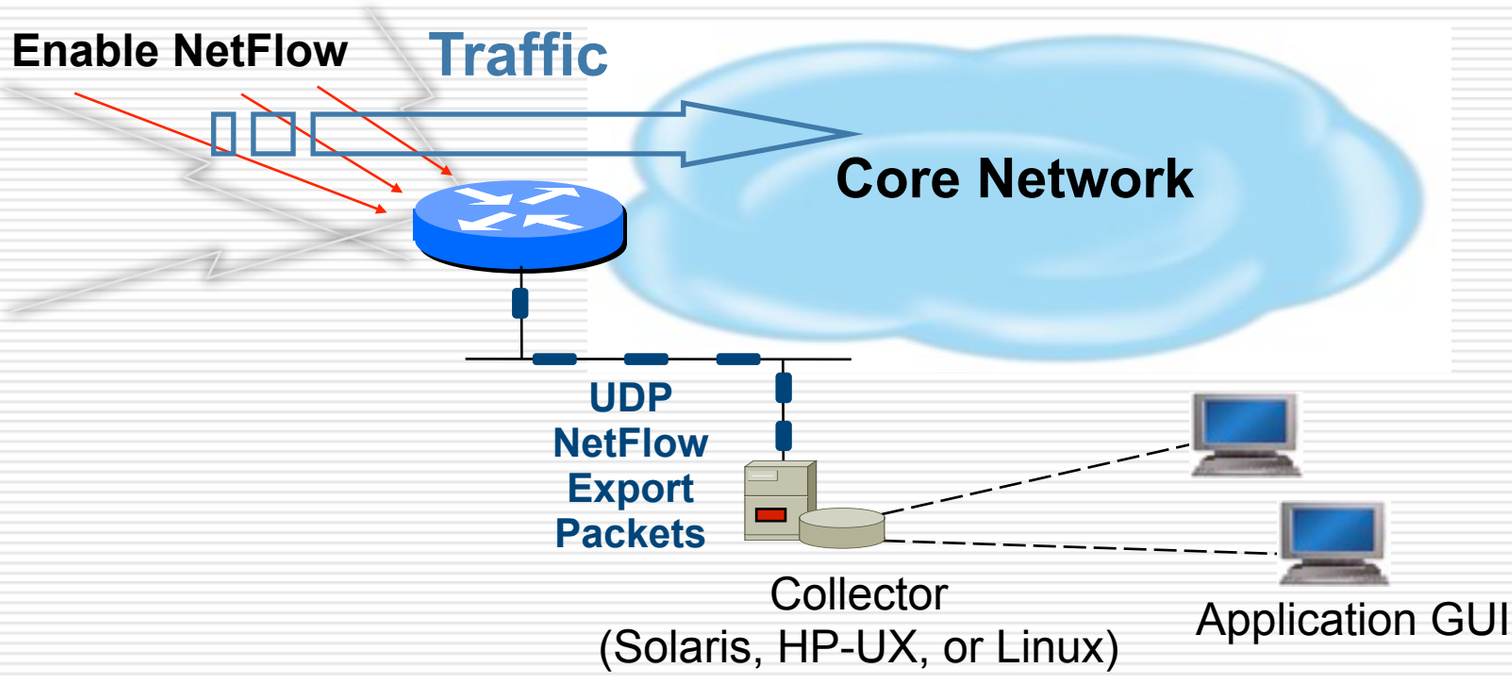
http://pe2.koganei.wide.ad.jp/cgi-bin/owd-stat

http://qpe.jp.apan.net/cgi-bin/owd-stat

Netflow - Overview -

- Overview
  - Enables IP traffic **flow** analysis without probes
  - Invented and patented by Cisco
    - Juniper (called cflowd), Foundry, ･･･ many venders are supporting
  - Flow cash data on routers is exported

    to a flow tool, so that traffic flow is to be analyzed



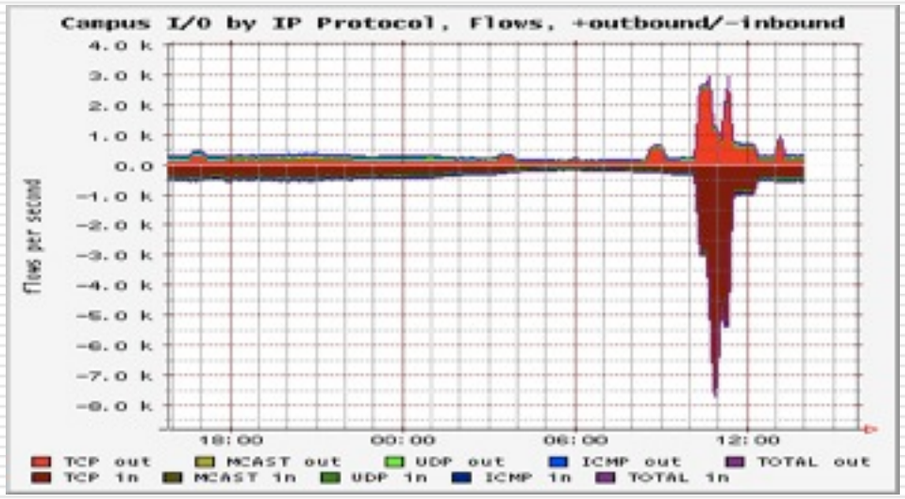
flow Definition:
- Source IP address
- Destination IP address
- Source port
- Destination port
- Layer 3 protocol type
- TOS byte (DSCP)
- Input logical interface (ifIndex)

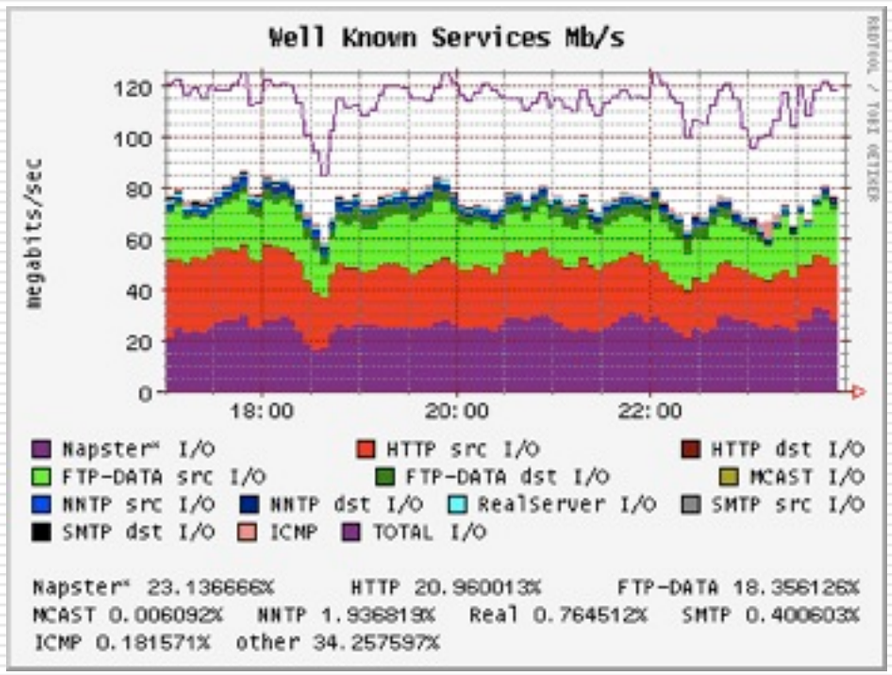
Netflow - Flow Data -

- Flow data export
  - Enable NetFlow on the router
    - There is difference in architecture between Cisco and Juniper routers
    - Take care! the load of a router does not become high!
      - Check CPU, memory, bandwidth, sampling rate
- Flow data collection & Analysis
  - Prepare the software for receiving flow-export data
    - flow-tools http://www.splintered.net/sw/flow-tools/
    - cflowd http://www.caida.org/tools/measurement/cflowd/
    - Cisco : NetflowCollector
  - Analyze traffic from raw data with software
    - flow-scan http://net.doit.wisc.edu/~plonka/FlowScan/
      (If you want to graph-ize analysis data, I recommend you to use RRDtool)
    - Cisco : CiscoWorks
      - Source and destination IP address
      - Source and destination TCP/UDP ports
      - Packet and byte counts
      - Routing information (next-hop address, source autonomous system (AS) number, destination AS number, source prefix mask, destination prefix mask)

Netflow - Example -

- Netflow Example





Introduction of other advanced tools

Abilene Router Proxy - Overview -

- Similar to Looking Glass, but with some advanced functions
- Web-form allows users to submit various commands to backbone routers
- Allows remote network operators to troubleshoot problems without contacting NOC
- Unix-based
- Uses scripted telnet to login to the routers and grab the output
- Not designed for high-speed access to backbone information
- Very useful operation tool among inter-domain network
- Enable us to view operational situation of almost all Abilene routers
- APAN Tokyo XP Router Proxy Service
  - http://monitor.jp.apan.net/routerproxy/


perfSONAR

July 7th 2010, perfSONAR Workshop – perfSONAR Tutorial

Jason Zurawski, Network Software Engineer, Research Liason

perfSONAR Preliminaries

Outline

- Motivation
  - A Typical Scenario
  - Possible Solutions
- What is *perfSONAR*?
  - Inception
  - Architecture Primer
  - Example Use Case
- Who is involved in *perfSONAR*?
  - perfSONAR-MDM
  - perfSONAR-PS
- Who is adopting *perfSONAR*?
- Practical Information
- Workshop Overview

Why Worry About Network Performance?

- Most network design lends itself to the introduction of flaws:
  - Heterogeneous equipment
  - Cost factors heavily into design – e.g. *Get what you pay for*
  - Design heavily favors ***protection*** and ***availability*** over performance
- Communication protocols are not advancing as fast as networks
  - *TCP/IP* is the king of the protocol stack
    - Guarantees reliable transfers
    - Adjusts to failures in the network
    - Adjusts speed to be *fair* for all
- User Expectations
  - *Big Science* is prevalent globally
  - The “8 Second Rule” is present in Scientific Communities too [1]

Motivation – A Typical Scenario

- User and resource are geographically separated
- Both have access to high speed communication network
  - LAN infrastructure - 1Gbps Ethernet
  - WAN infrastructure – 10Gbps Optical Backbone

Motivation – A Typical Scenario

- User wants to access a file at the resource (e.g. ~600MB)
- Plans to use COTS tools (e.g. “scp”, but could easily be something scientific like “GridFTP” or simple like a web browser)
- What are the expectations?
  - 1Gbps network (e.g. *bottleneck* speed on the LAN)
  - 600MB * 8 = 4,800 Mb file
  - User expects *line rate*, e.g. 4,800 Mb / 1000 Mbps = 4.8 Seconds
  - Audience Poll: Is this expectation too high?
- What are the realities?
  - Congestion and other network performance factors
  - Host performance
  - Protocol Performance
  - Application performance

Motivation – A Typical Scenario

- Real Example (New York USA to Los Angeles USA):

```
[zurawski@nms-rthr2 ~]$ scp zurawski@bwctl.losa.net.internet2.edu:pS-Performance
_Toolkit-3.1.1.iso .
pS-Performance_Toolkit-3.1.1.iso                2%   17MB   1.0MB/s   10:05 ETA_
```

- 1MB/s (8Mb/s) ??? 10 Minutes to transfer???
- Seems unreasonable given the investment in technology
  - Backbone network
  - High speed LAN
  - Capable hosts
- Performance realities as network speed decreases:
  - 100 Mbps Speed – 48 Seconds
  - 10 Mbps Speed – 8 Minutes
  - 1 Mbps Speed – 80 Minutes
- How could this happen? More importantly, why are there not more complaints?
- Audience Poll: Would you complain? If so, to whom?
- Brainstorming the above – where should we look to fix this?

Motivation – A Typical Scenario

- Expectation does not even come close to experience, time to debug. Where to start though?
  - Application
    - Have other users reported problems? Is this the most up to date version?
  - Protocol
    - Protocols typically can be tuned on an individual basis, consult your operating system.
  - Host
    - Are the hardware components (network card, system internals) and software (drivers, operating system) functioning as they should be?
  - LAN Networks
    - Consult with the local administrators on status and potential choke points
  - Backbone Network
    - Consult the administrators at remote locations on status and potential choke points (Caveat – do you [should you] know who they are?)

Motivation – A Typical Scenario

- Following through on the previous, what normally happens …
  - Application
    - This step is normally skipped, the application designer will *blame the network*
  - Protocol
    - These settings may not be explored. Shouldn't this be automatic (e.g. autotuning)?
  - Host
    - Checking and diagnostic steps normally stop after establishing connectivity. E.g. "can I ping the other side"
  - LAN Networks
    - Will assure "internal" performance, but LAN administrators will ignore most user complaints and shift blame to upstream sources. E.g. "our network is fine, there are no complaints"
  - Backbone Network
    - Will assure "internal" performance, but Backbone responsibilities normally stop at the demarcation point, blame is shifted to other networks up and down stream

Motivation – A Typical Scenario

- Stumbling Blocks to solving performance problems
  - Lack of a clear process
    - Knowledge of the proper order to approach problems is paramount
    - This knowledge is not just for end users – also for application developers and network operators too
  - Impatience
    - Everyone is impatient, from the user who wants things to work to the network staff and application developers who do not want to hear complaints
  - Information Void
    - Lack of a clear location that describes symptoms and steps that can be taken to mitigate risks and solve problems
    - Lack of available performance information, e.g the current status of a given network in a public and easily accessible forum
  - Communication
    - Finding whom to contact to report problems or get help in debugging is frustrating

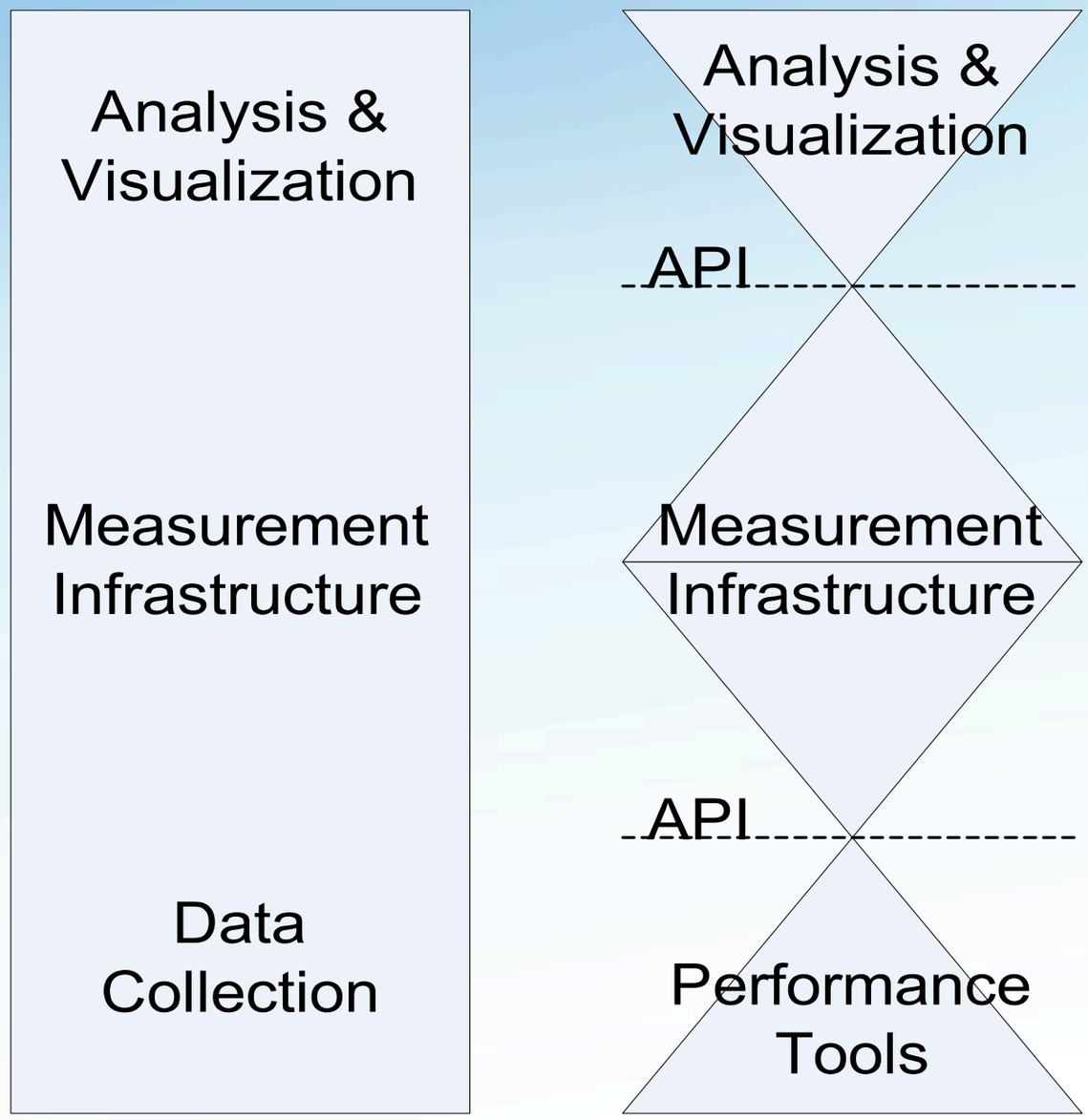
Motivation – Possible Solutions

- 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

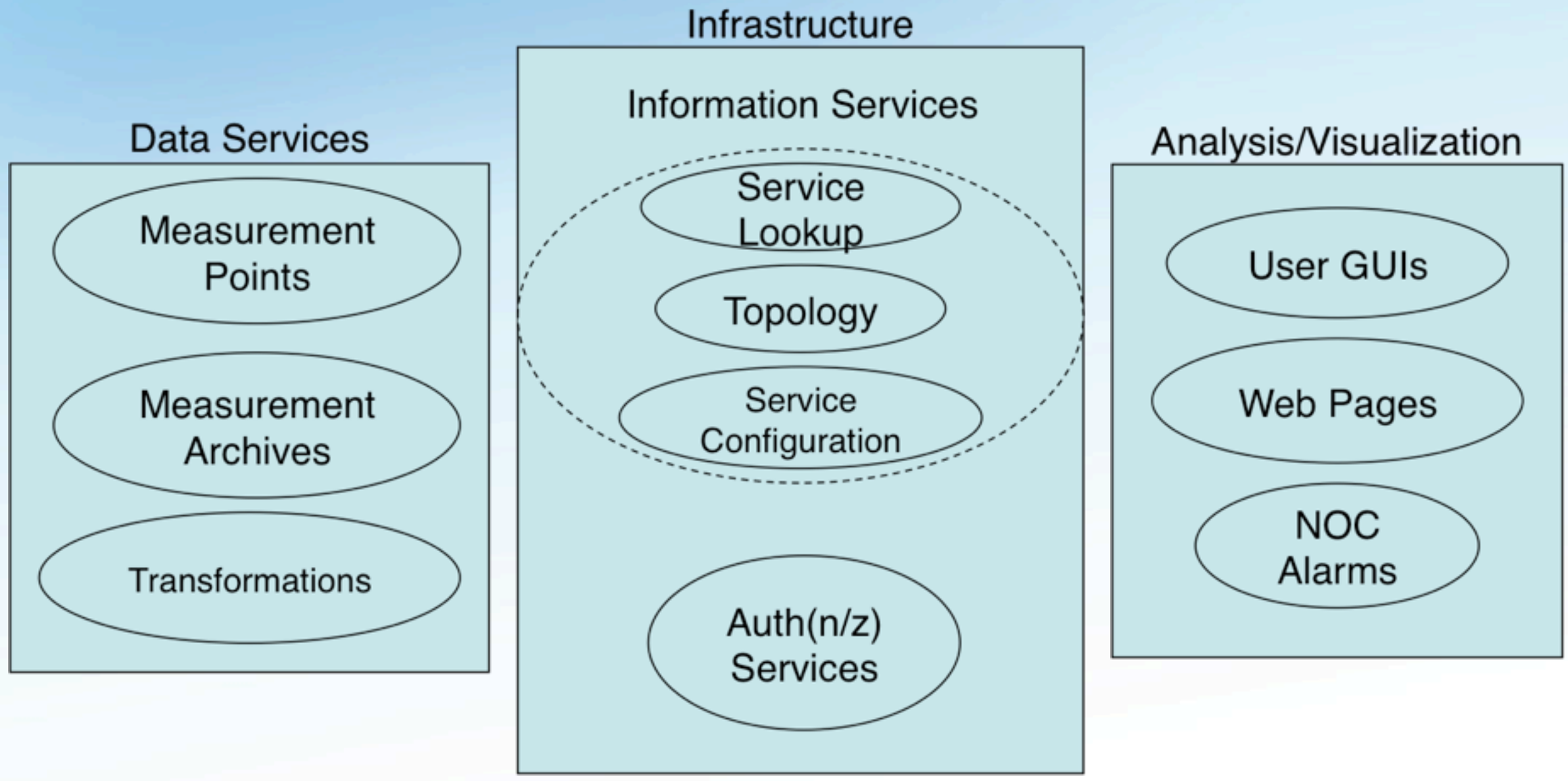
perfSONAR Architecture Overview

- 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
- perf*SONAR* Integrates:
  - Network measurement tools and archives (e.g. stored measurement results)
  - Data manipulation
  - Information Services
    - Discovery
    - Topology
  - Authentication and authorization

perfSONAR Architecture Overview

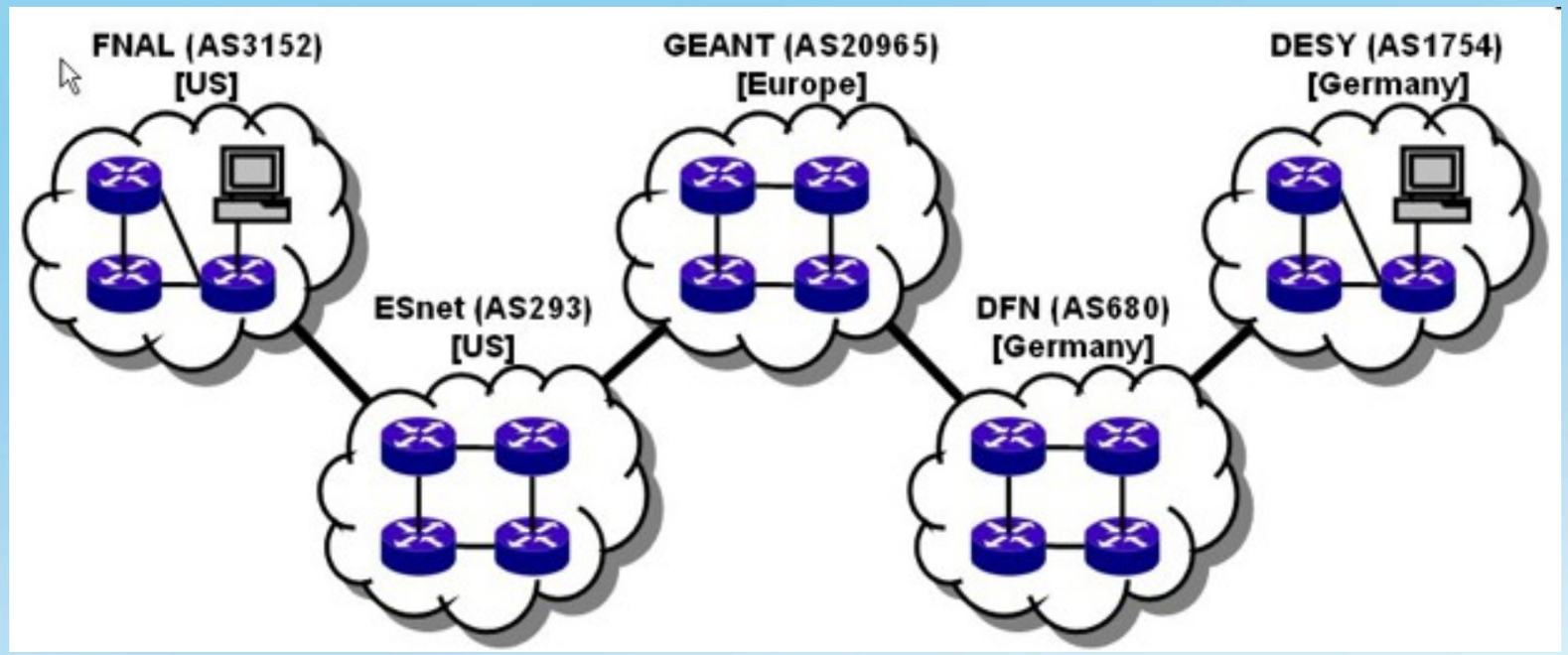
- The key concept of *perfSONAR* is that each entity (e.g. “services”) performs a function
  - Each service provides a limited set of functionality, e.g. collecting measurements between arbitrary points or managing the registration and location of distributed services
  - The service is a ***self contained*** 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

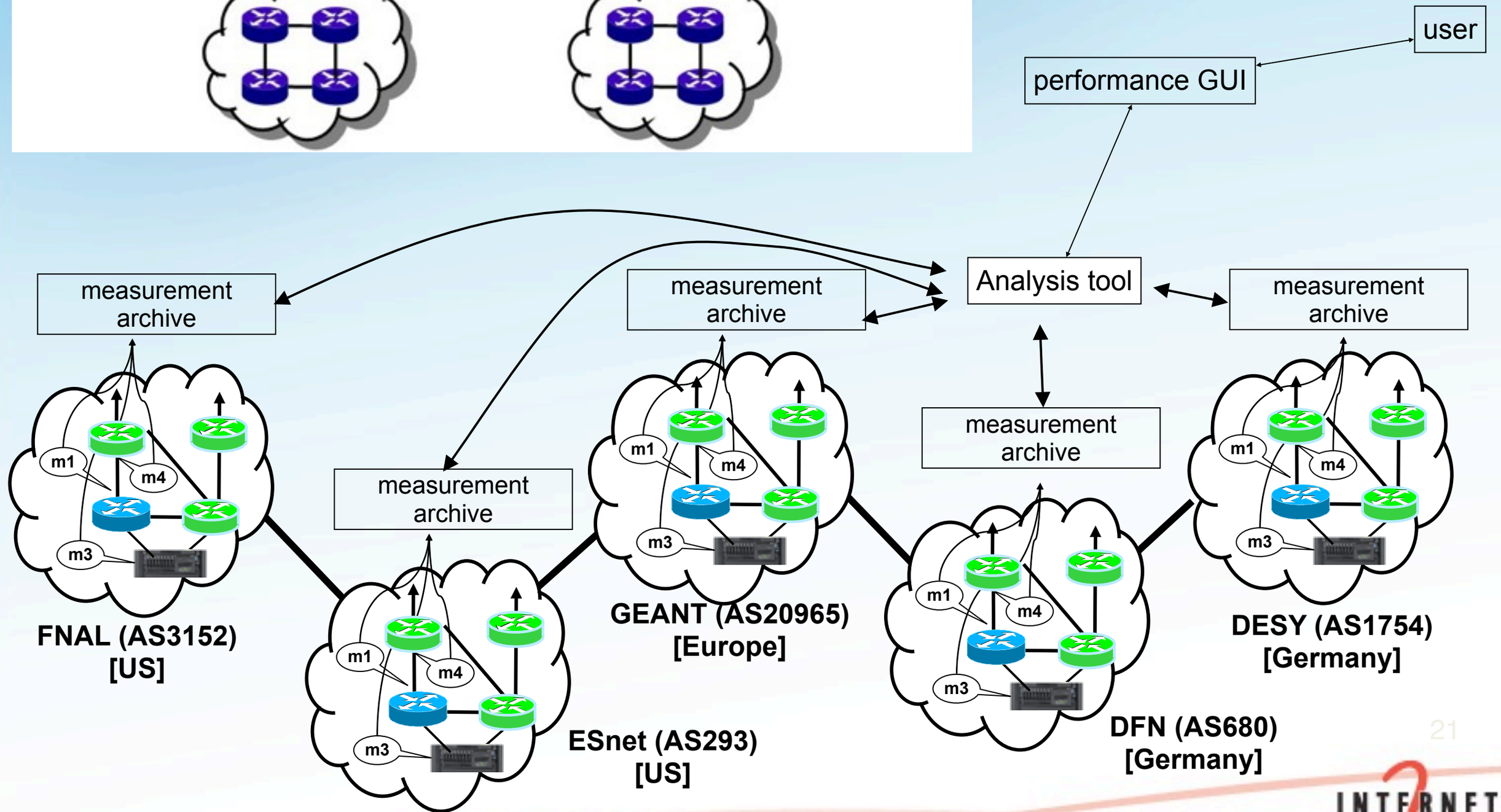


perfSONAR Architecture Overview

- 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



perfSONAR powered

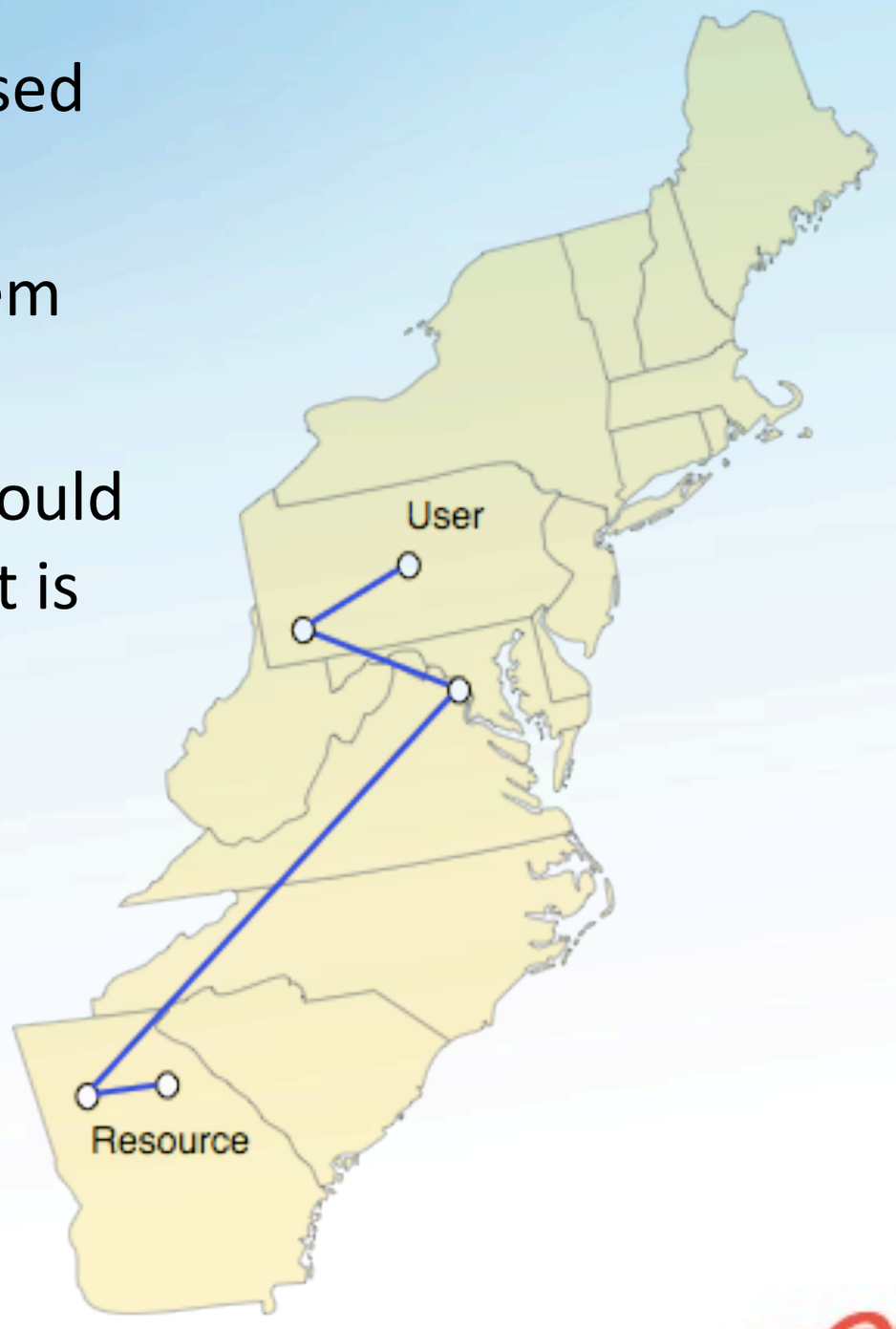
INTERNET 2

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?

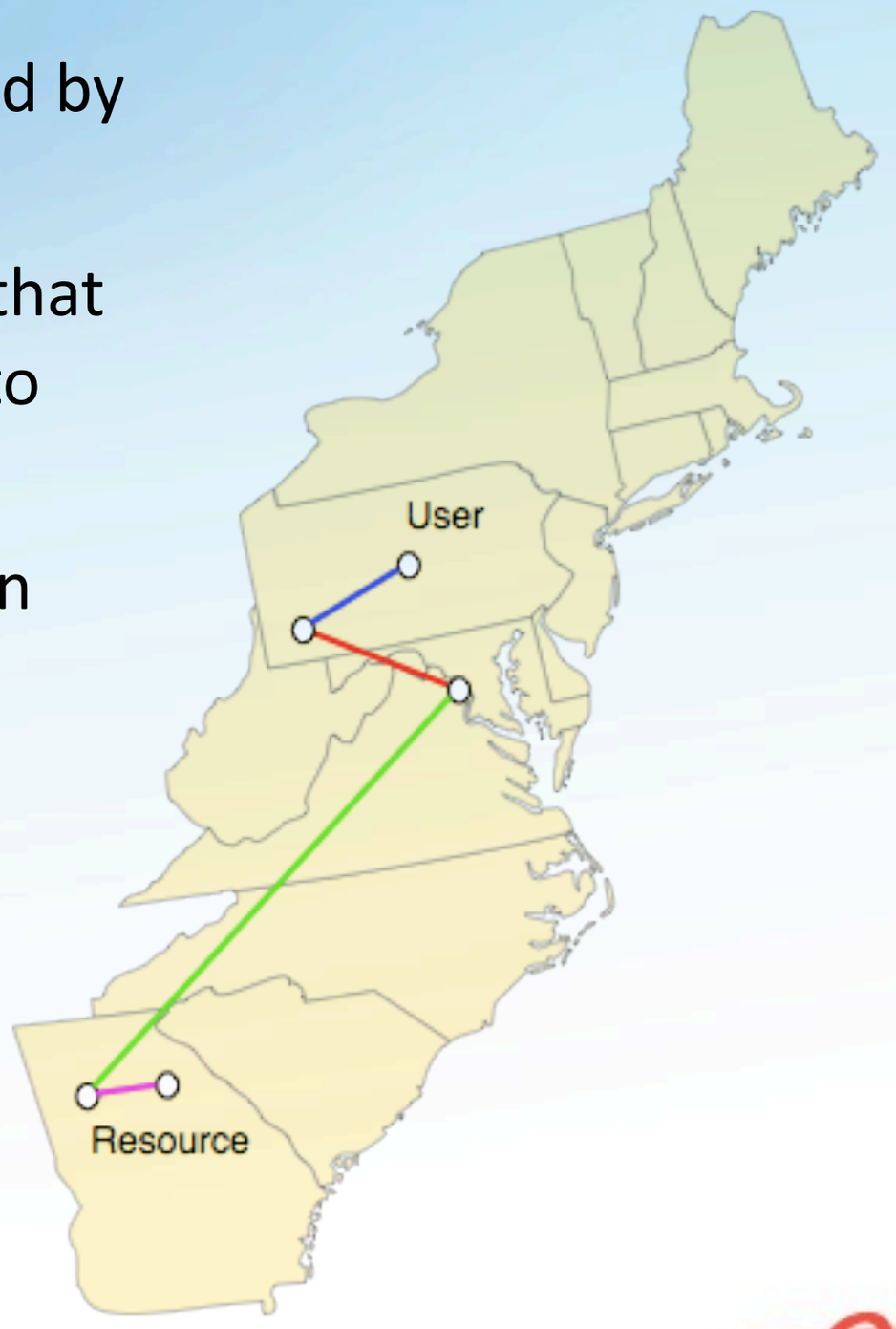
Example perfSONAR Use Case

- 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



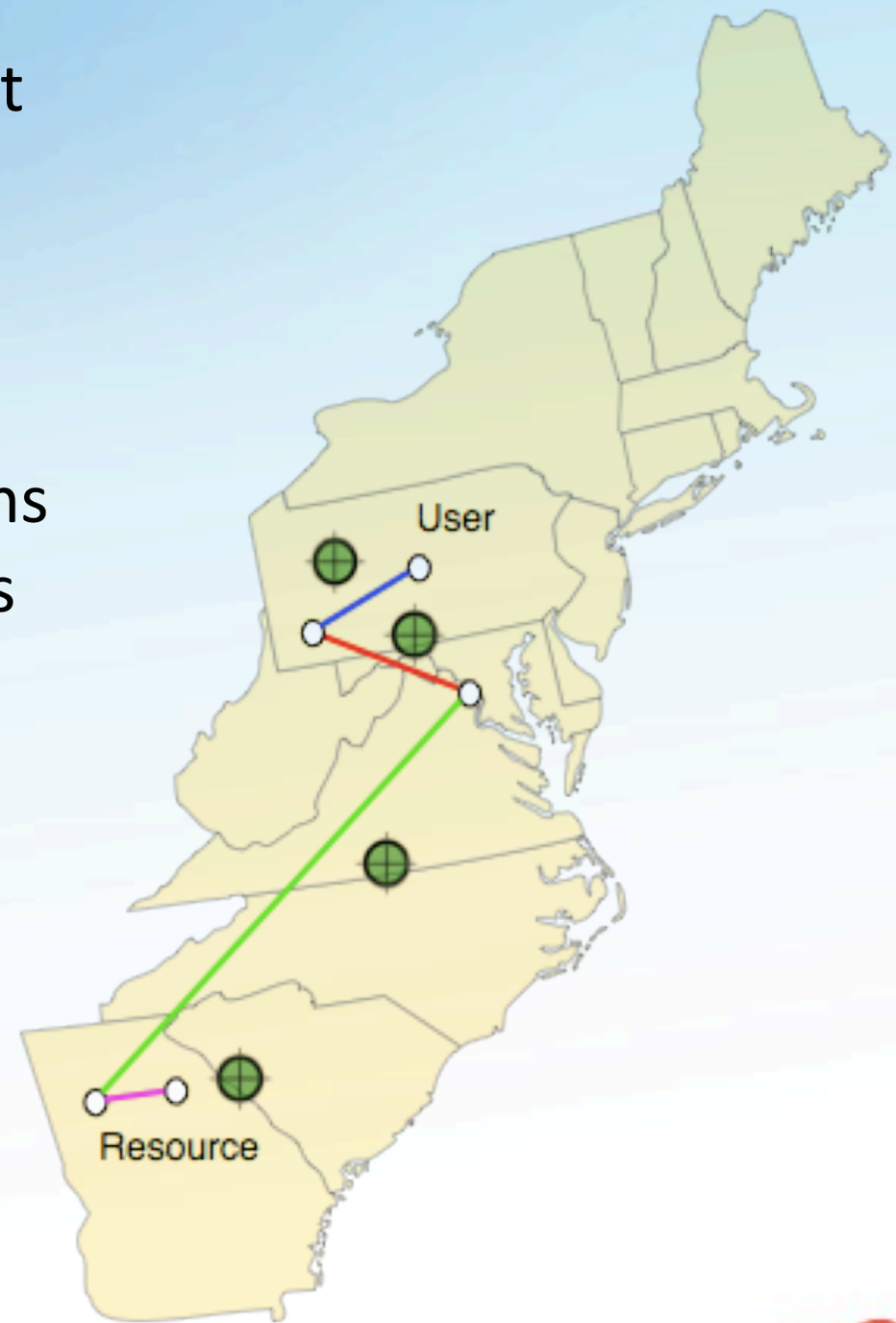
Example perfSONAR Use Case

- 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



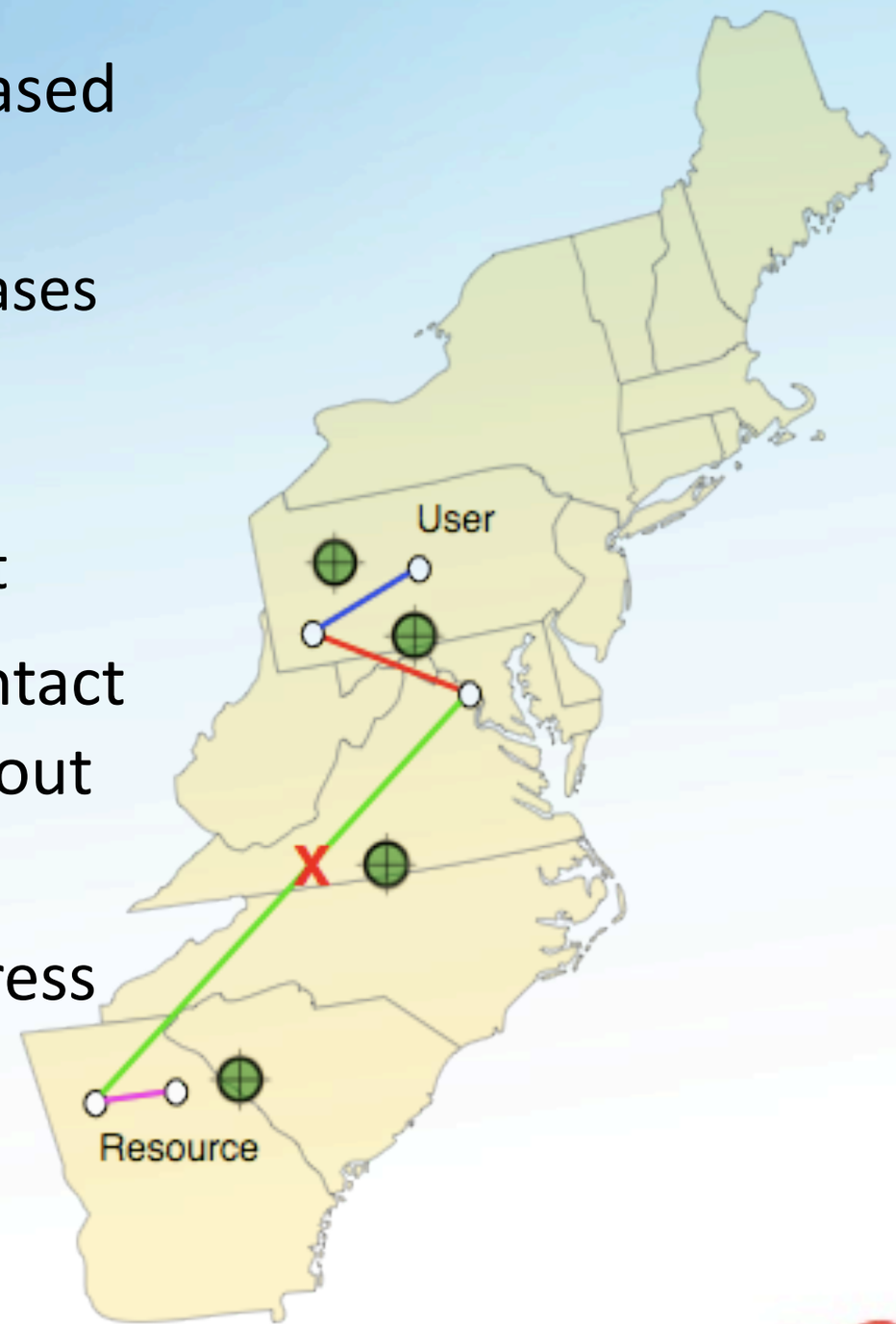
Example perfSONAR Use Case

- 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



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



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.

pS-performance ToolKit

- easy to start perfSONAR
- CentOS is required.
- perfSONAR
- regular monitoring tools
- WEB visualizing tools
- http://psps.internet2.edu/toolkit

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 manufactures

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/Oceana
    - 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
  - Louisianna 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 Illionios *
  - 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

What is currently available

Services useful for solving end-to-end performance problems

- Link Utilization (and errors, and discards)
  - SNMP or TL1
  - RRD/SQL Based
- Interface Status
  - Useful for multi-domain monitoring
- Ping/Traceroute beacons
  - Sources to test against for a given network
- ”looking glass”es
  - Run commands on network devises. Authentication will vary naturally
- Active measurements and active measurement results
  - Latency (and loss, duplicates): one-way and round-trip
  - Throughput (Iperf for now)
  - NDT and NPAD server locations
  - ‘Universal’ MP
    - Run many commands like ping, traceroute

How might perfSONAR be useful to you?

- A data source for your research
  - Active deployments (as in the ones just described) are all over.
  - Easy to locate using the ‘Lookup Service’
  - Data policy varies, but most are open
- A place to try out new algorithms and visualizations, and feed them back to production networks
  - Open Source Development model, your help is more than welcome!
- A place to install new data sources with the cooperation of production networks
  - Integrate new tools into the framework
  - Construct new data representations

For more information

- General and MDM impelmentation: 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://stats1.es.net/cgi-bin/tree.cgi
  http://www.perfsonar.net/activeServices/IS/
- More human-readable list of services: http://stats1.es.net/cgi-bin/directory.cgi
  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-user
  - 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-announce
  - 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 Preliminaries

July 7th 2010, perfSONAR Workshop – perfSONAR Tutorial

Jason Zurawski - Network Software Engineer, Research Liason

For more information, visit http://www.perfsonar.net

July 7th 2010, perfSONAR Workshop – perfSONAR Tutorial

Jason Zurawski, Network Software Engineer, Research Liason

perfSONAR Architecture

Outline

- Motivation and Goals
- Service Oriented Architecture (SOA)
- Architecture Overview
  - Service Components
  - Client Examples
- Framework Interactions
  - Deploying a Service
  - Simple Client/Service Interaction
    - Echo
    - Metadata Request
    - Data Request
  - Lookup Service Interaction

Motivation and Goals

- Thus far we have seen that the *perfSONAR* framework can be used to solve end-to-end performance problems for multi-domain paths.
- The framework is made up of several unique components and design considerations, all of which operate in a cooperative yet independent manner
  - Each functionality is separated into a specific function
  - Clients and servers interact through scripted, XML Based protocols
  - Measurement data is encoded in expressive XML formats

Service Oriented Architecture (SOA)

- Interoperable network measurement middleware (SOA):
  - Modular
  - Web services-based
  - Decentralized
  - Locally controlled
- Integrates:
  - Network measurement tools and data archives
  - Data manipulation
  - Information Services
    - Discovery
    - Topology
    - Authentication and authorization
- Based on:
  - Open Grid Forum Network Measurement (NM) Working Group schema
  - Currently attempting to formalize specification of *perfSONAR* protocols in the Network Measurement Control (NMC) working group.
  - Network topology description being defined in the Network Markup Language (NML) Working Group

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

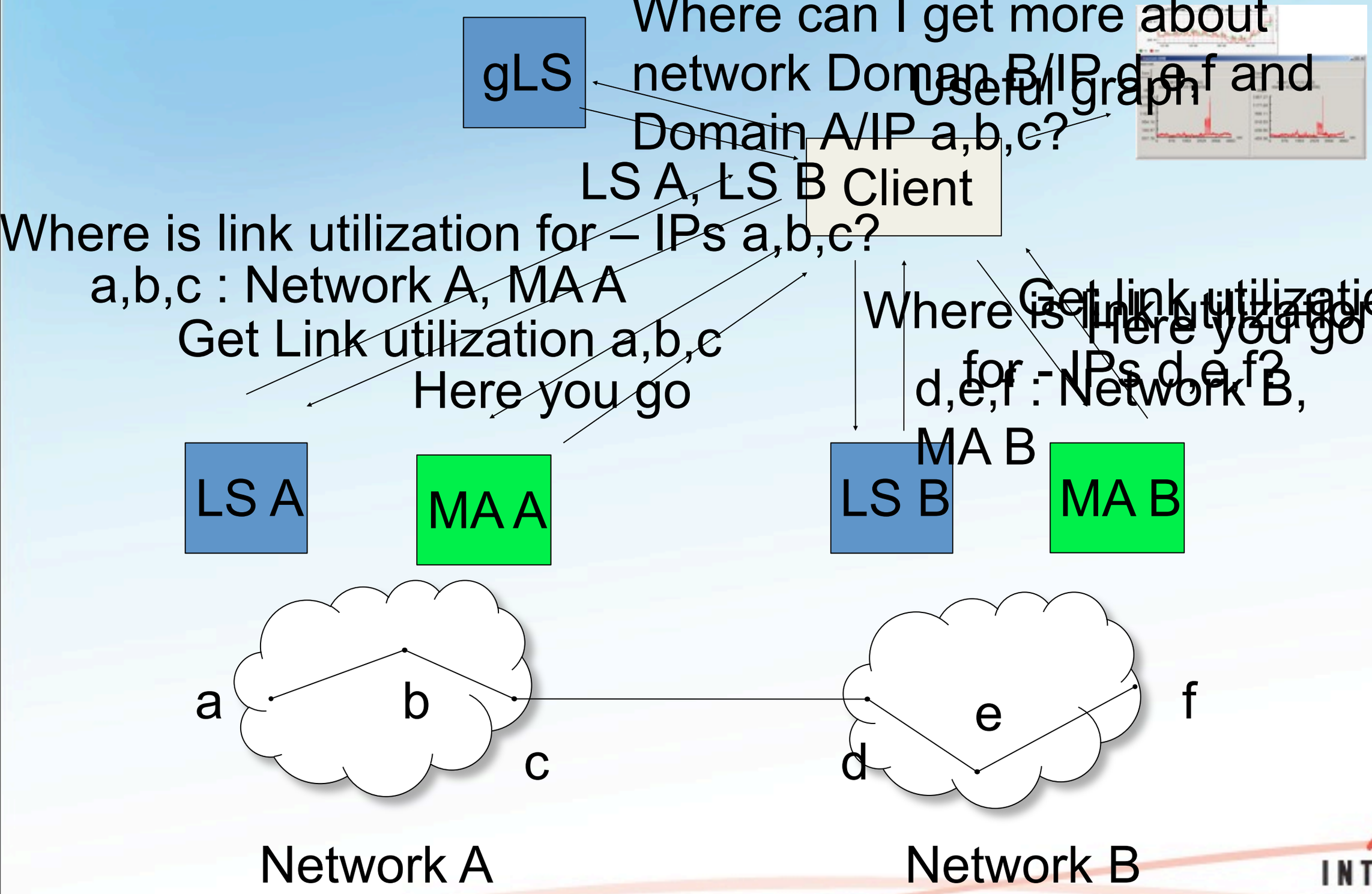
These services are specifically concerned with the job of network performance measurement and analysis

Service Oriented Architecture (SOA)

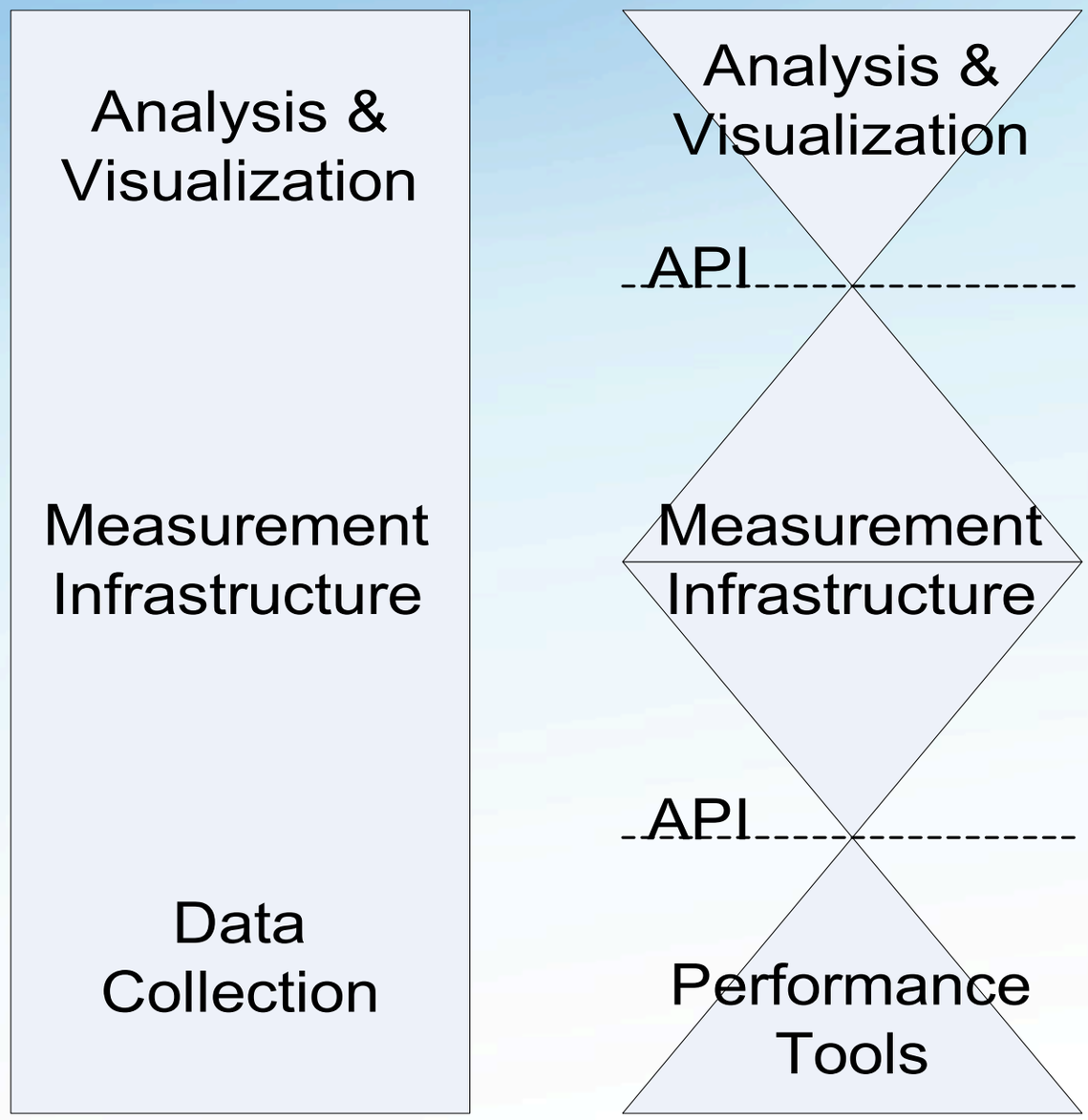
- Lookup Service
  - Allows the client to discover the existing services and other LS services.
  - Dynamic: services registration themselves to the LS and mention their capabilities, they can also leave or be removed if a service goes down.
- Topology Service
  - Make the network topology information 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

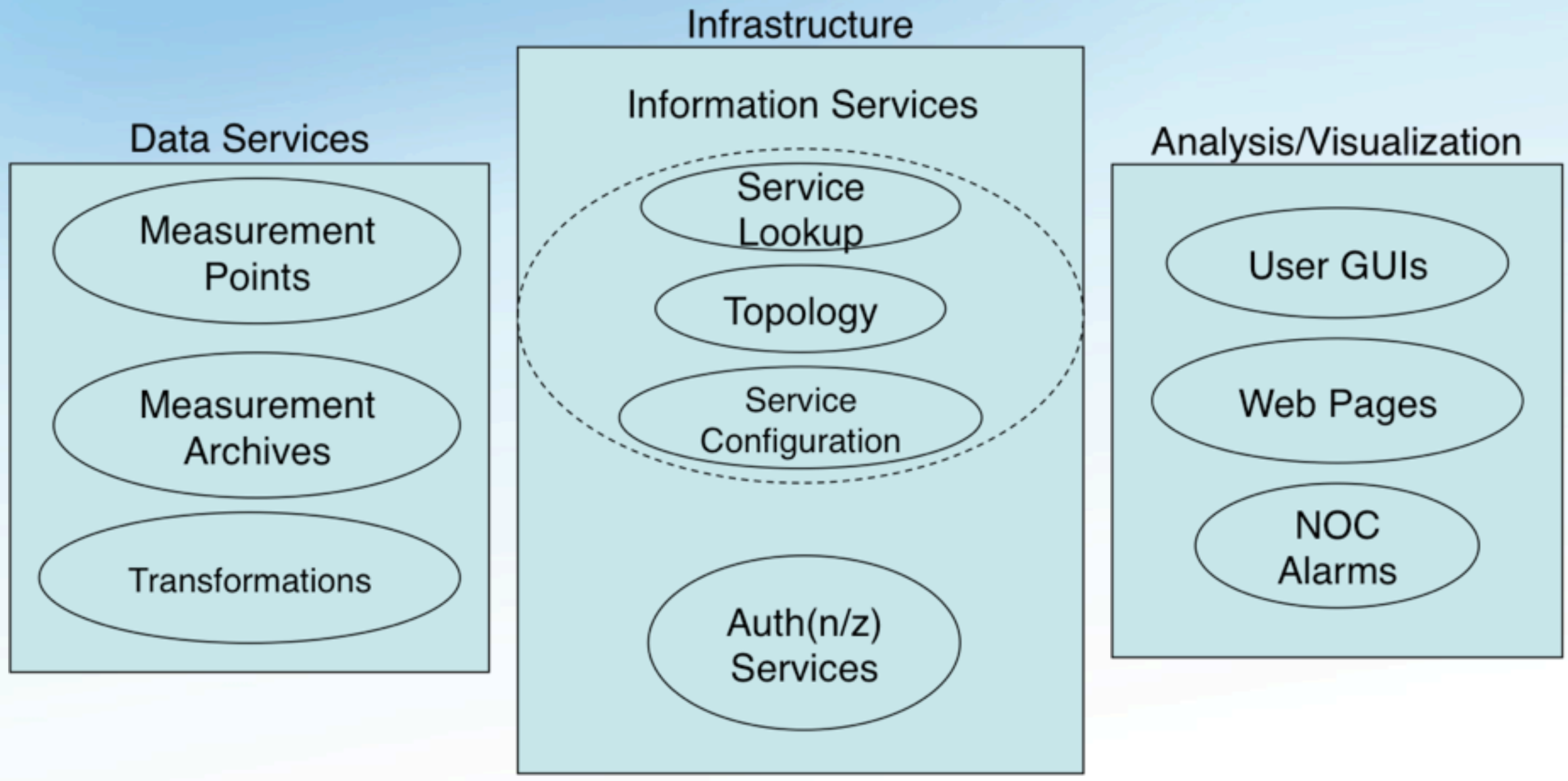
Service Oriented Architecture (SOA)



Architecture Overview



perfSONAR Architecture Overview



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
- Rolls 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)
- Rolls 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
- Rolls 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
- Rolls of the Transformation Service :
  - 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 (Planned):
  - 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 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
- Rolls 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

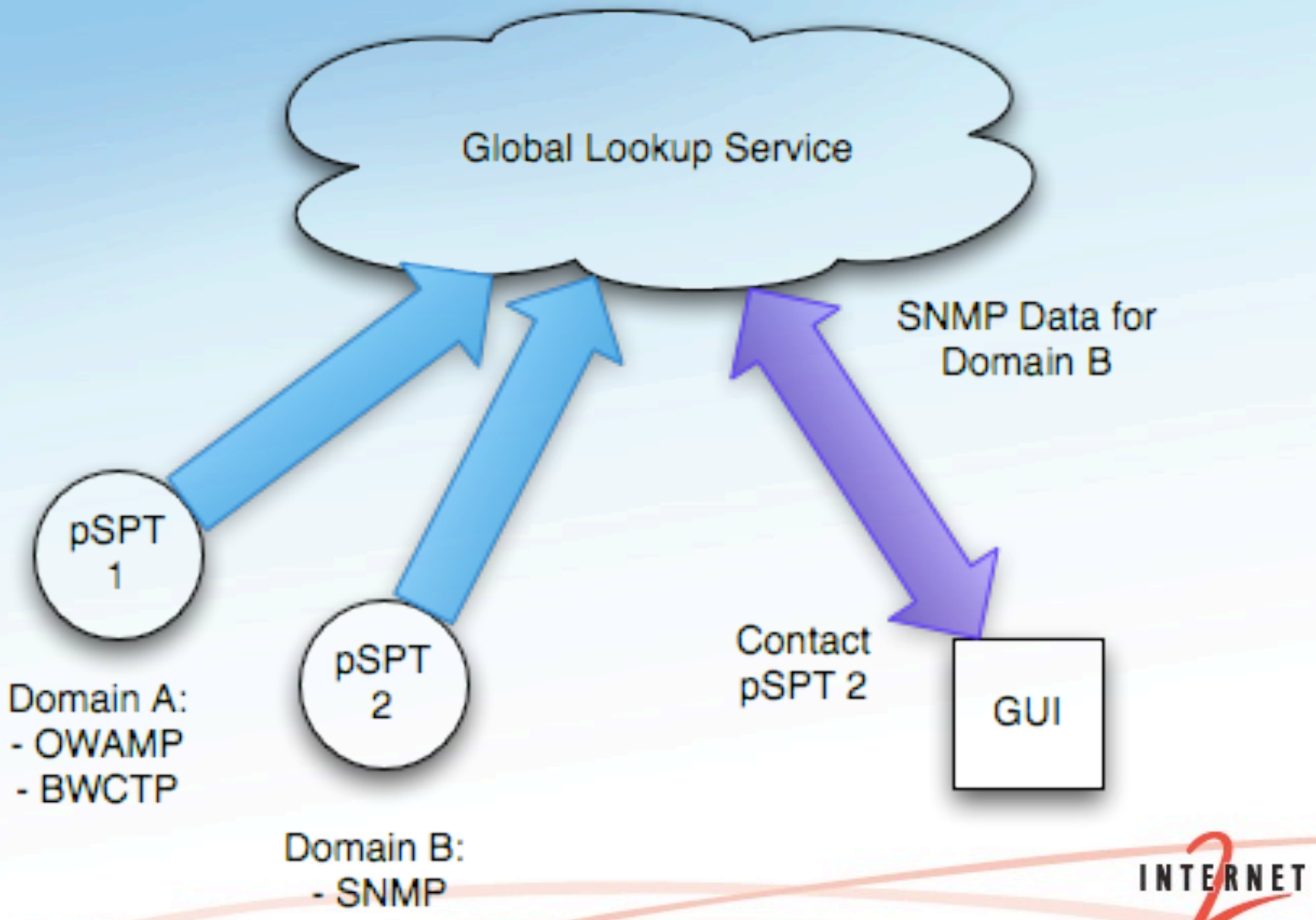
Architecture Overview - hLS

- 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 measurment (‘*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 - Communities

- Communities = Web 2.0 Content Tagging
  - Think Flicker (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):

Communities[1] This Host Participates In	
Internet2	Delete
perfSONAR-PS	Delete
Add New Community	

Popular Communities As Of 2009-09-22 08:02 (Click To Join)
Atlas CMS DOE DOE Sites DOE-SC-LAB ESnet GRNOC Internet2_CTP KREONET LHC RNP Sites USATLAS Utah

- 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 Washingont, 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

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
- Rolls 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

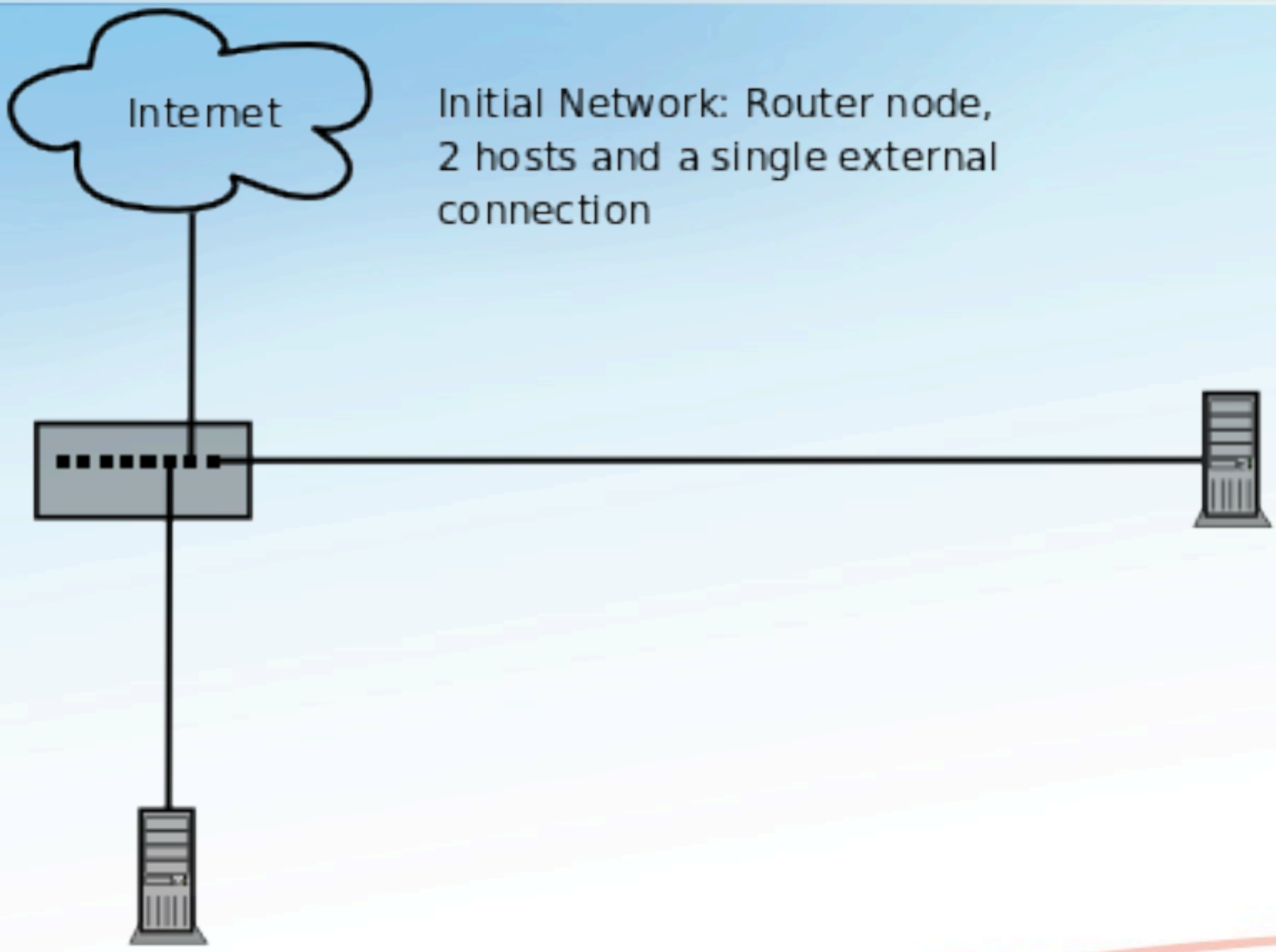
Framework Interaction

- The following examples will illustrate the mechanics of services and protocols:
  - Deploying a Service
  - Simple Client/Service Interaction
    - Echo
    - Metadata Request
    - Data Request
  - Lookup Service Interaction

Deploying a Service

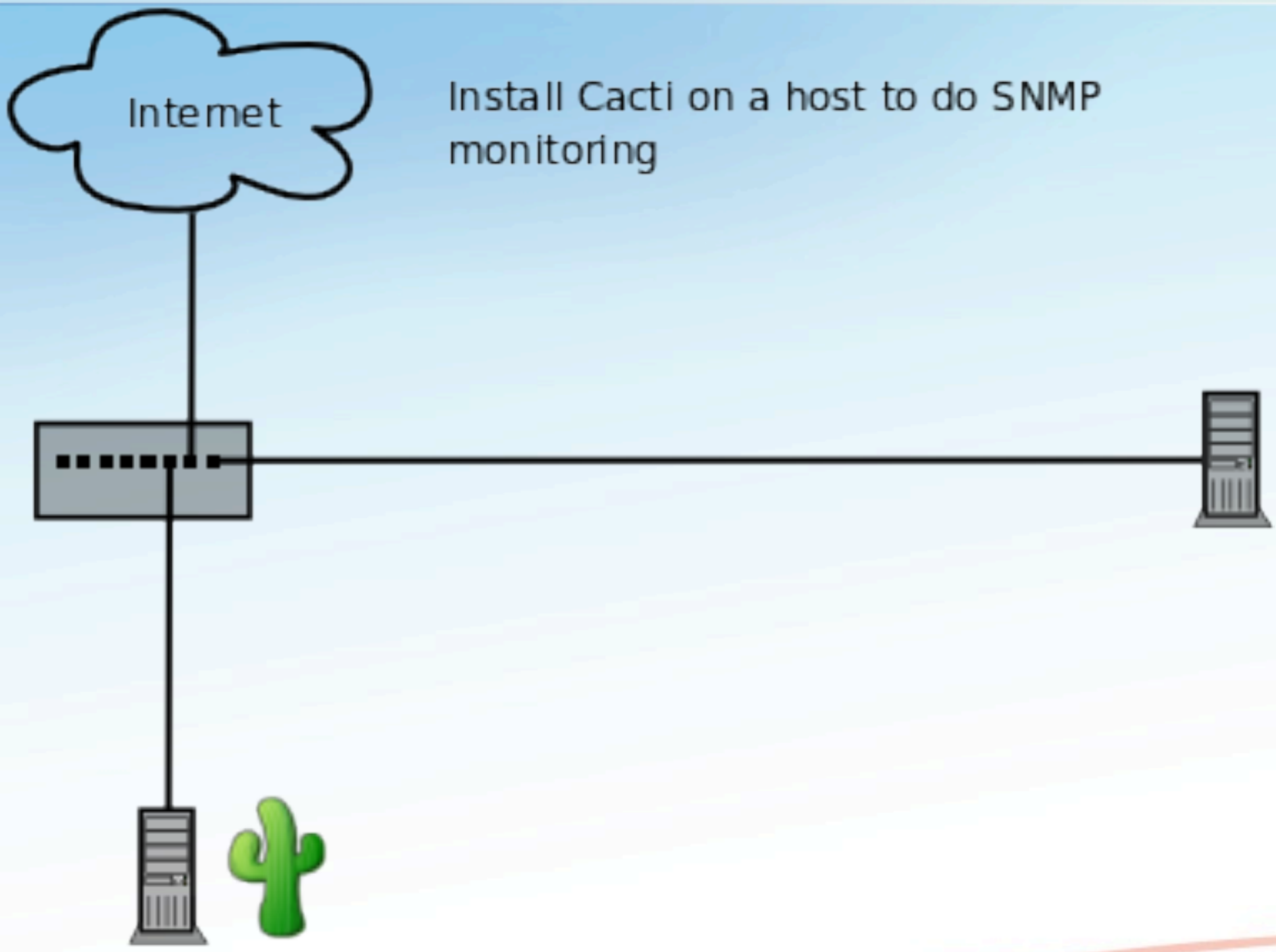
- A perfSONAR service is deployed alongside the measurement infrastructure
- Interactions with the lookup service and clients are described

Deploying a Service

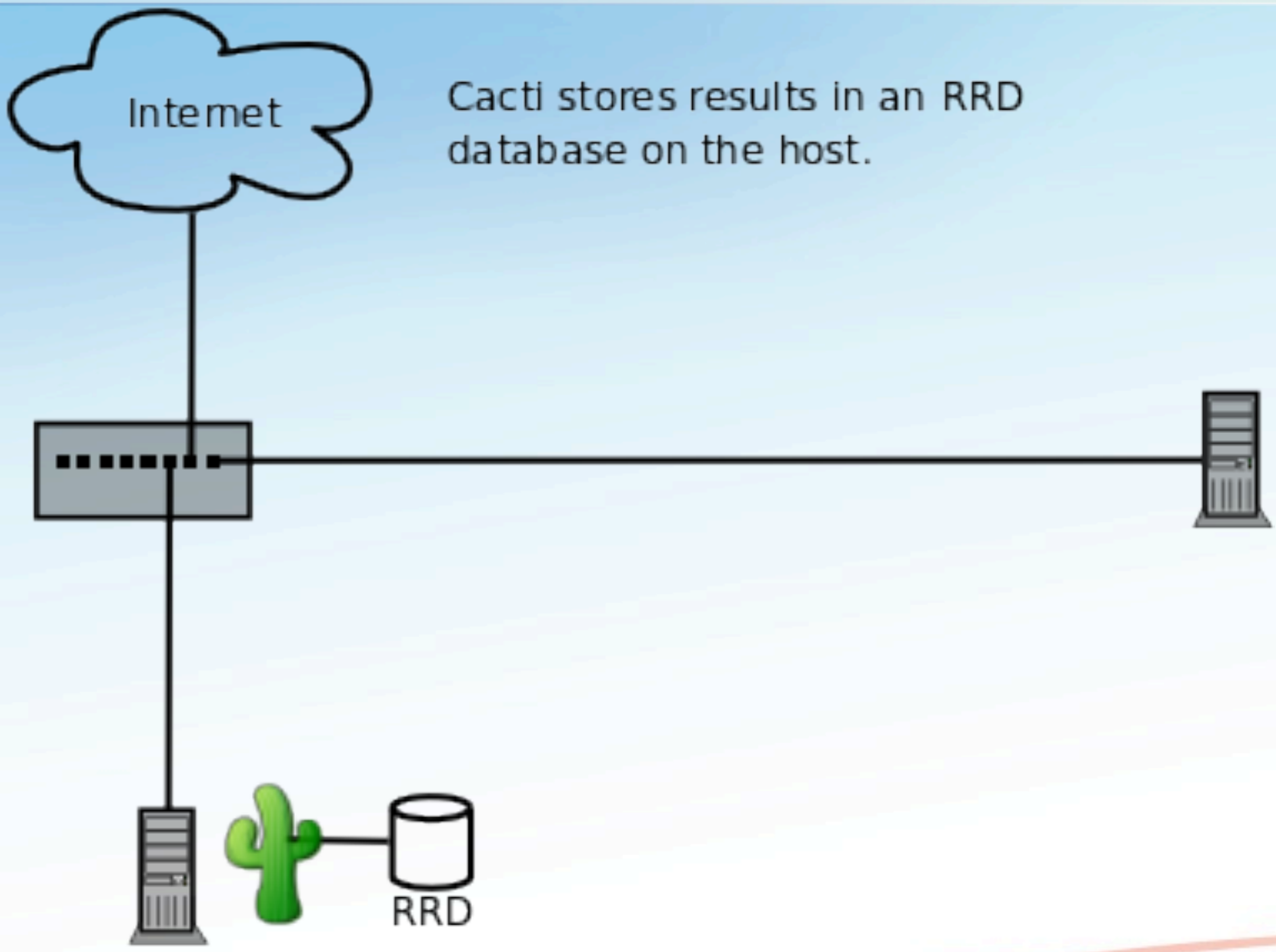


Initial Network: Router node, 2 hosts and a single external connection

Deploying a Service

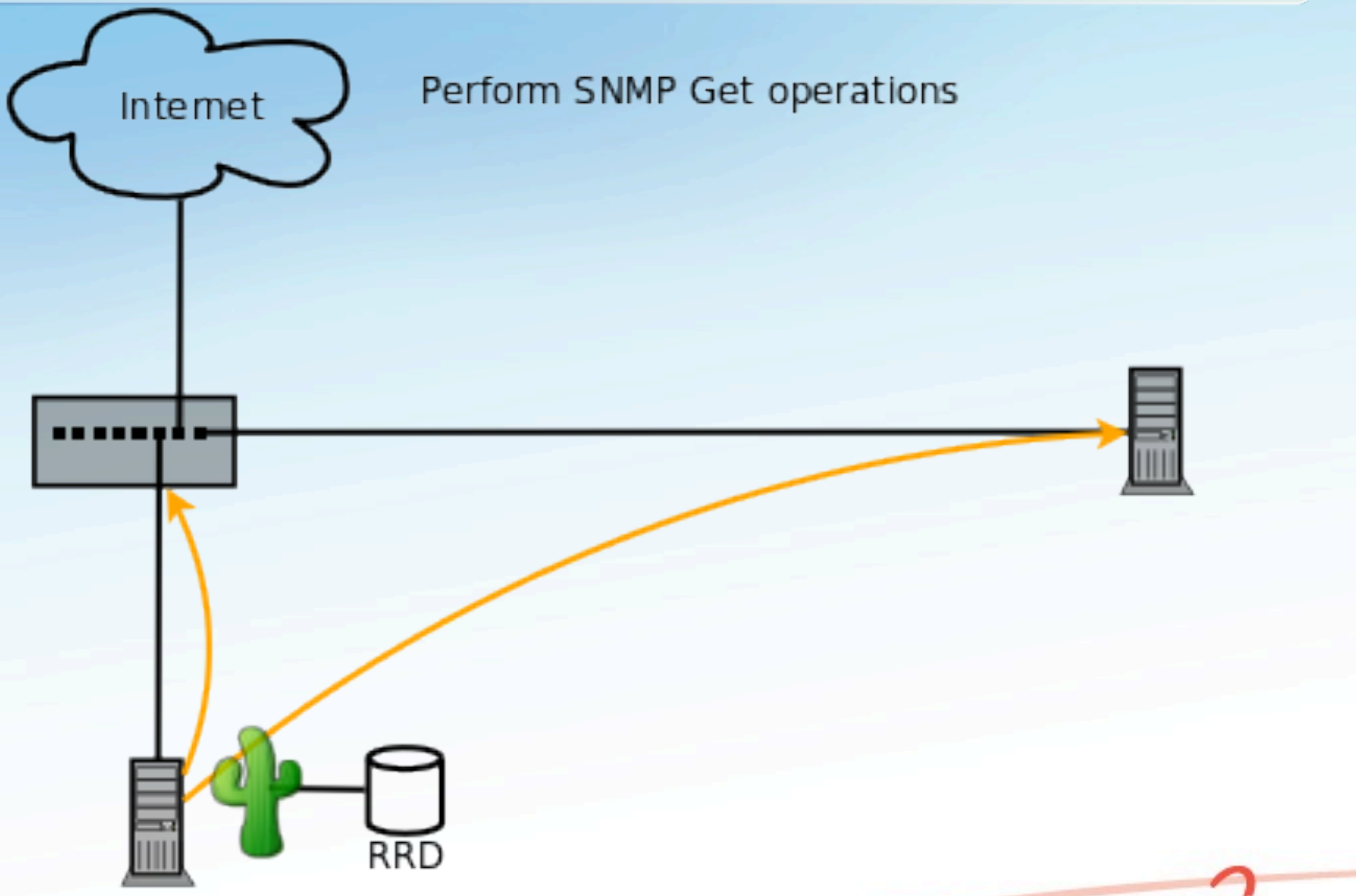


Deploying a Service

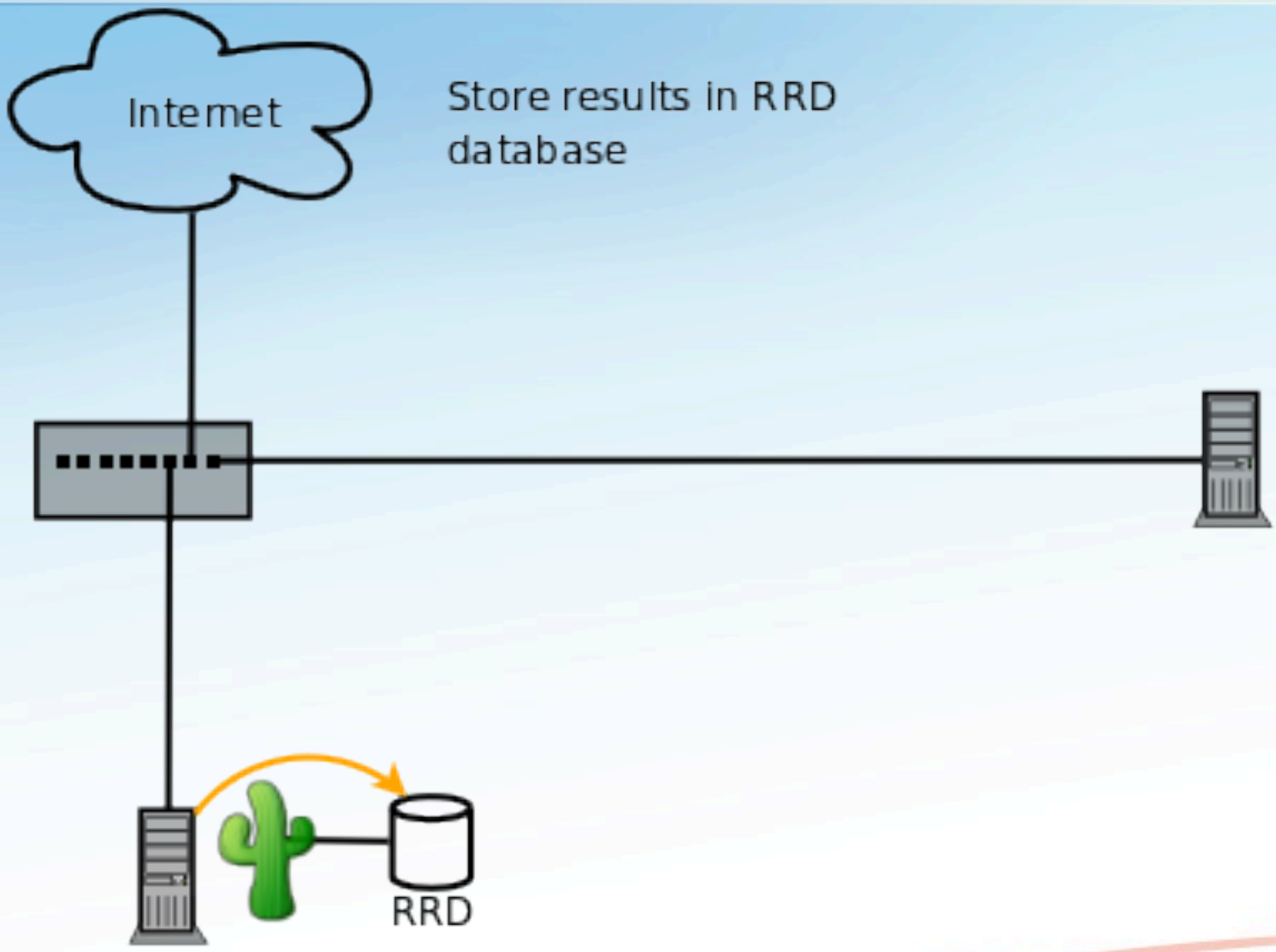


Cacti stores results in an RRD database on the host.

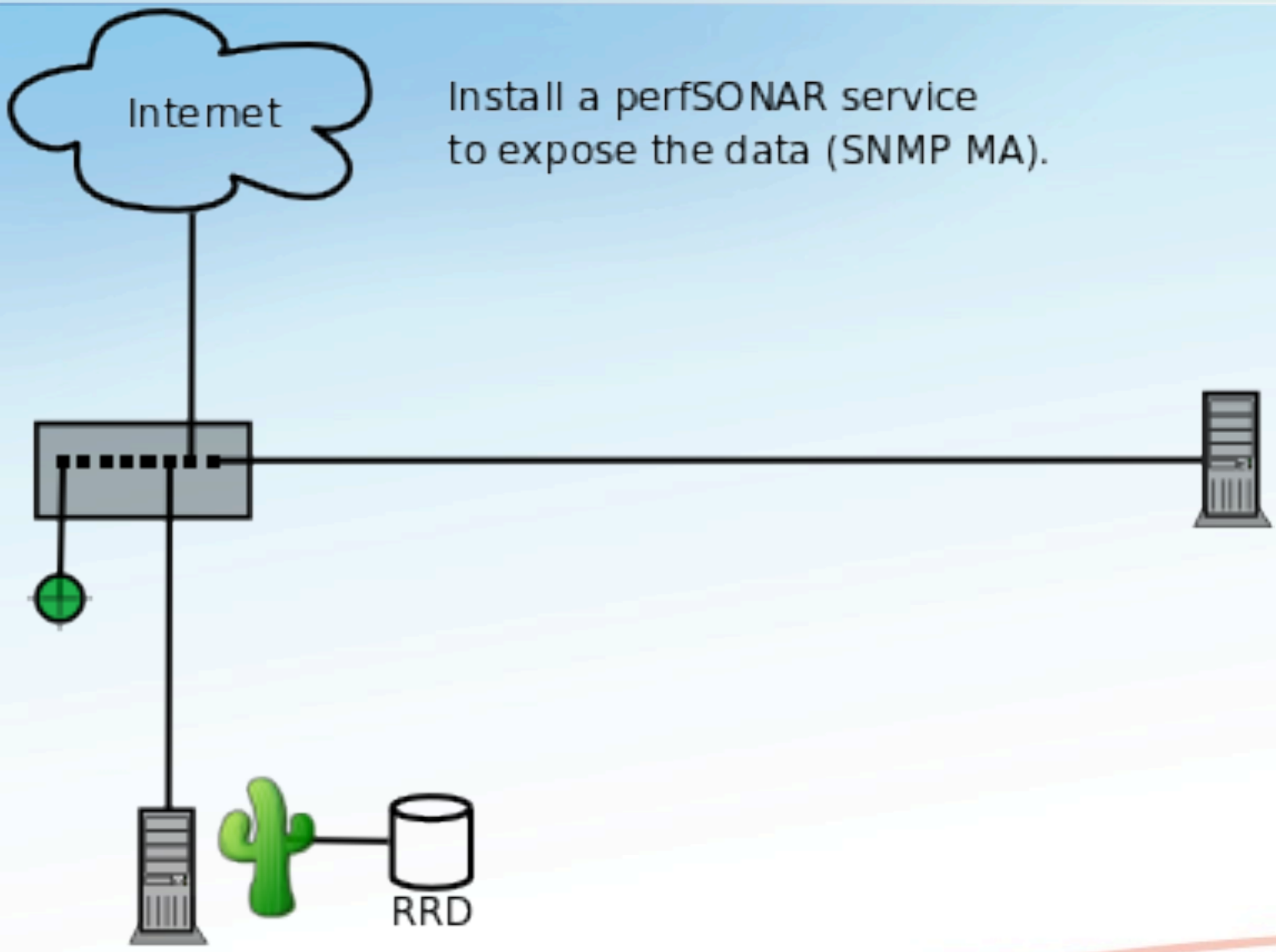
Deploying a Service



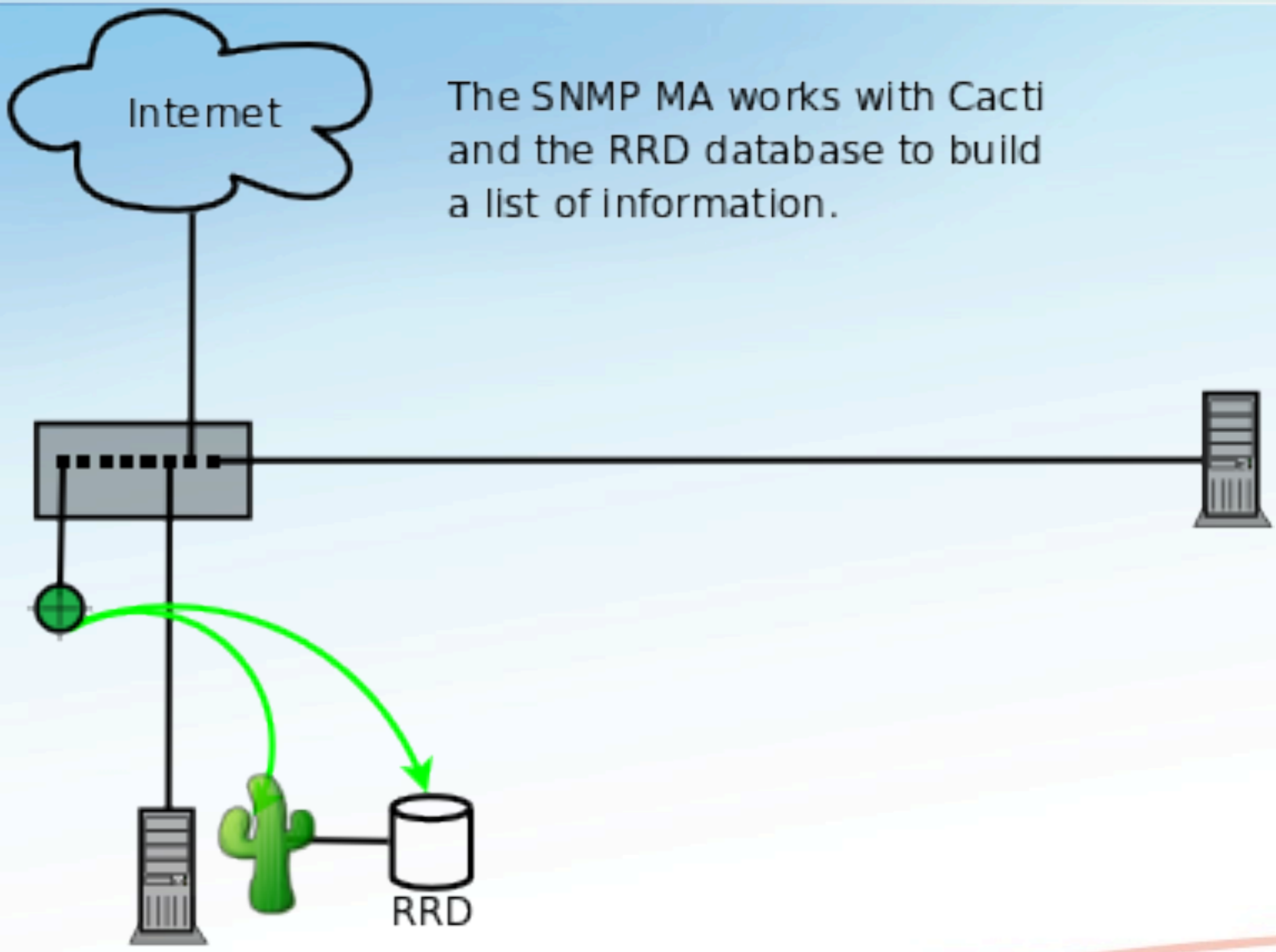
Deploying a Service



Deploying a Service

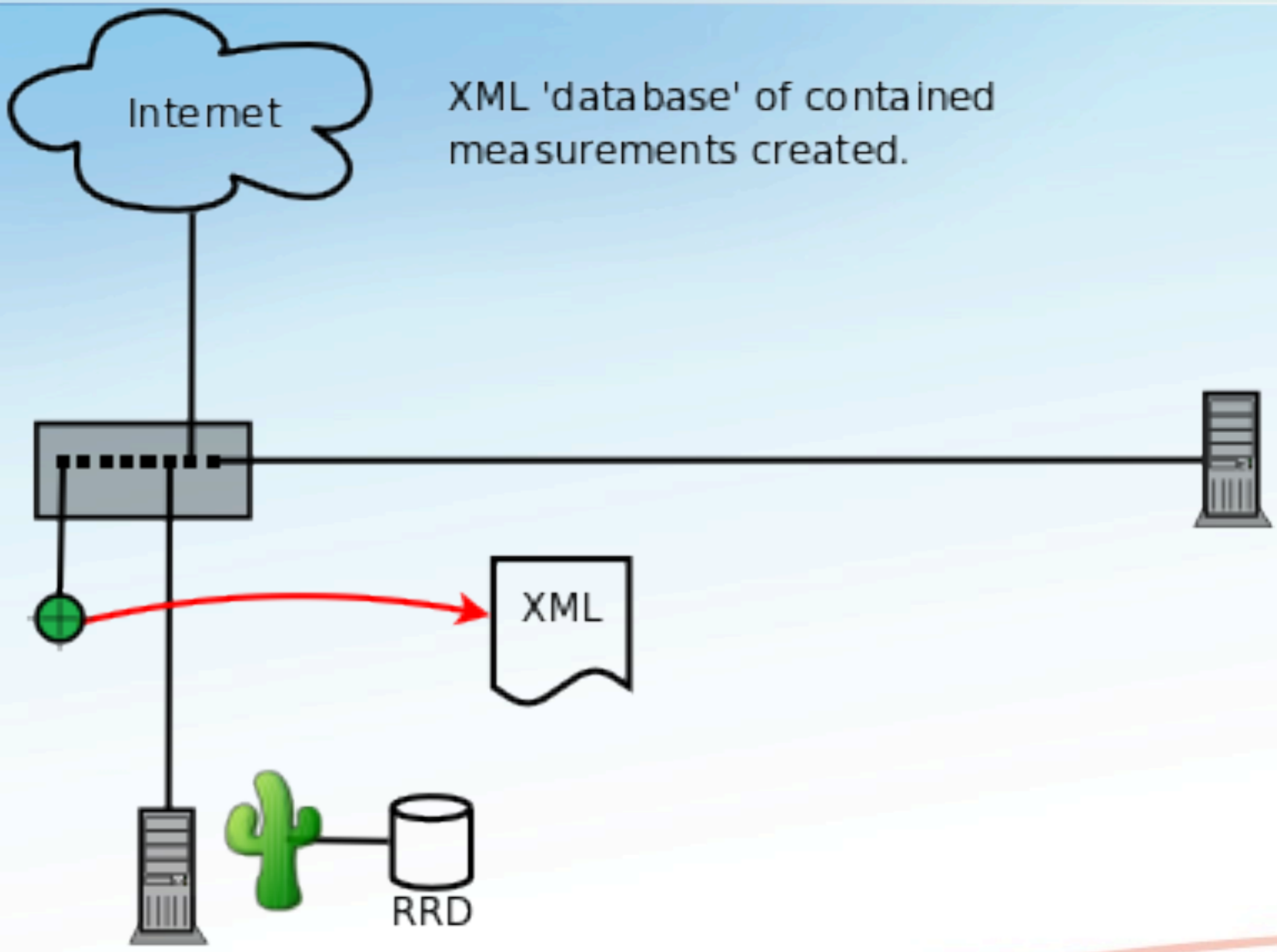


Deploying a Service

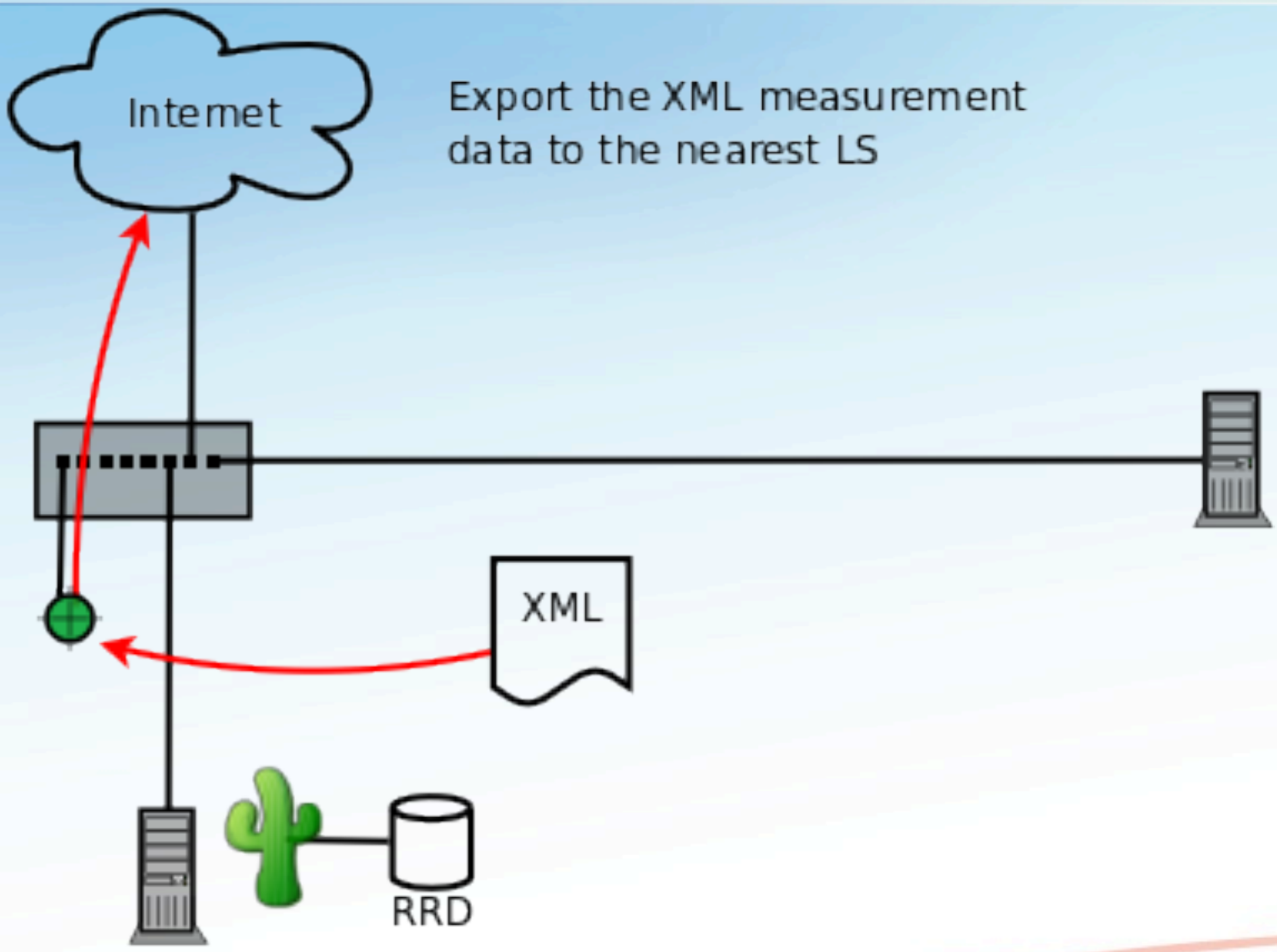


The SNMP MA works with Cacti and the RRD database to build a list of information.

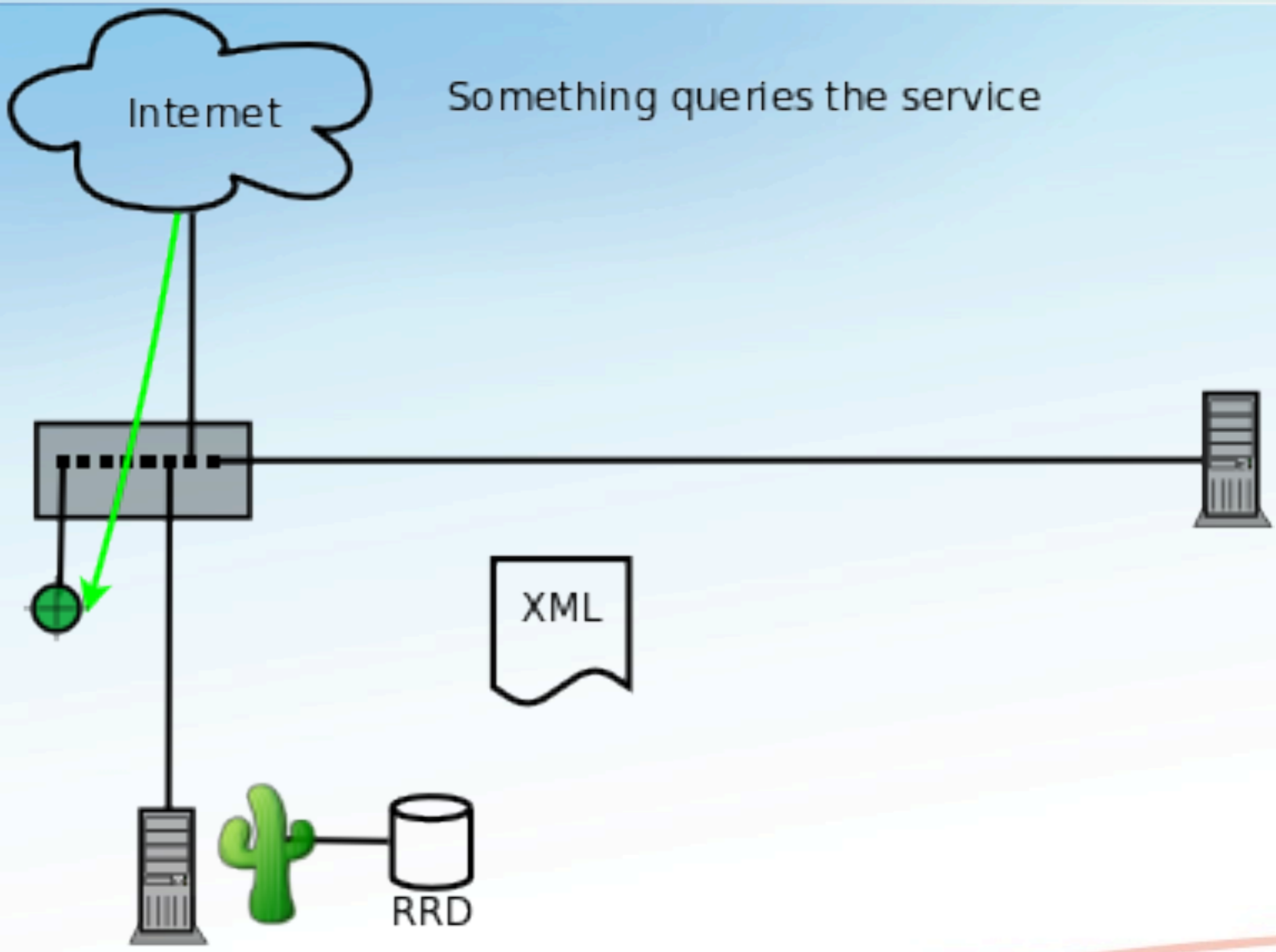
Deploying a Service



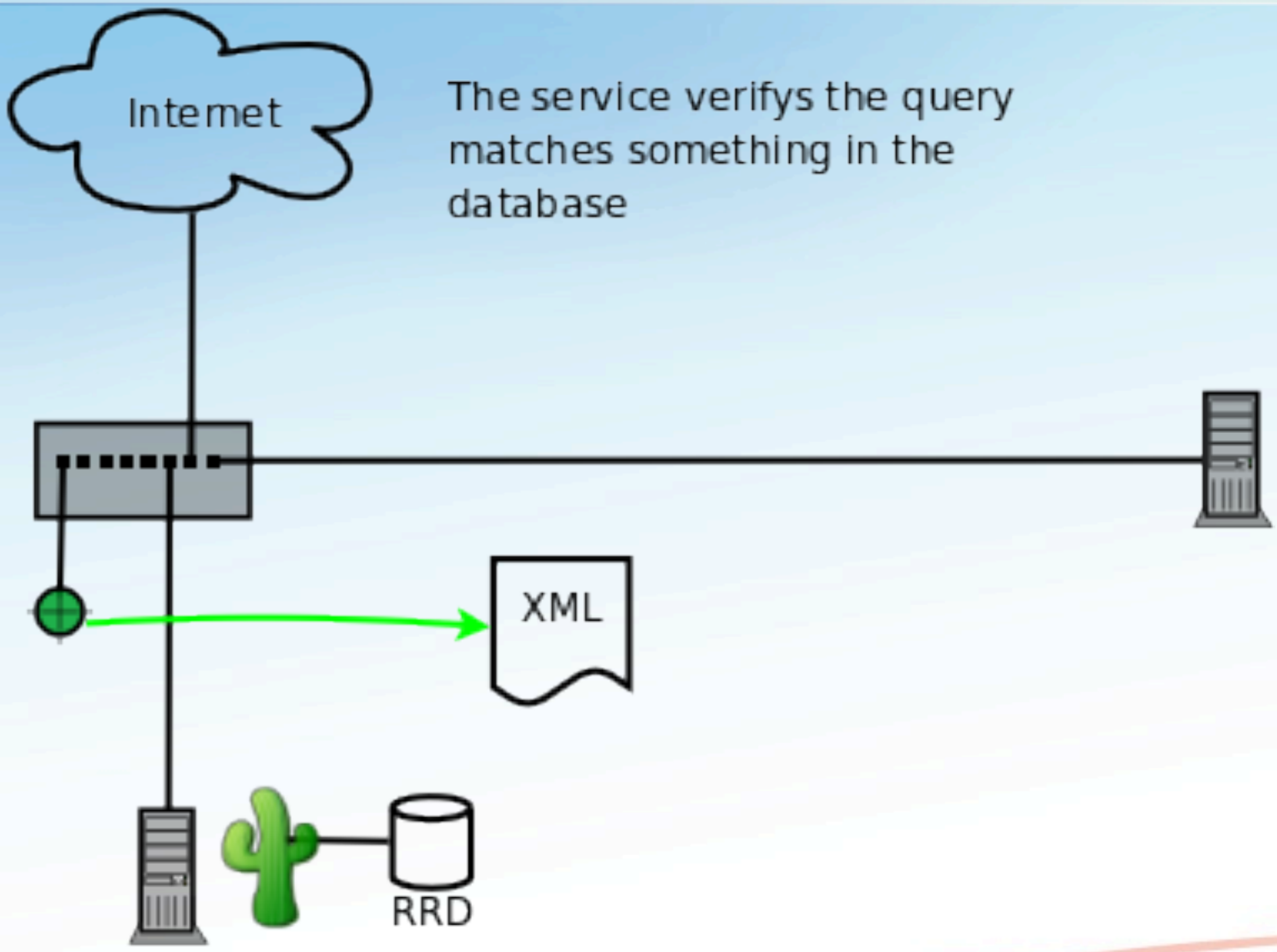
Deploying a Service



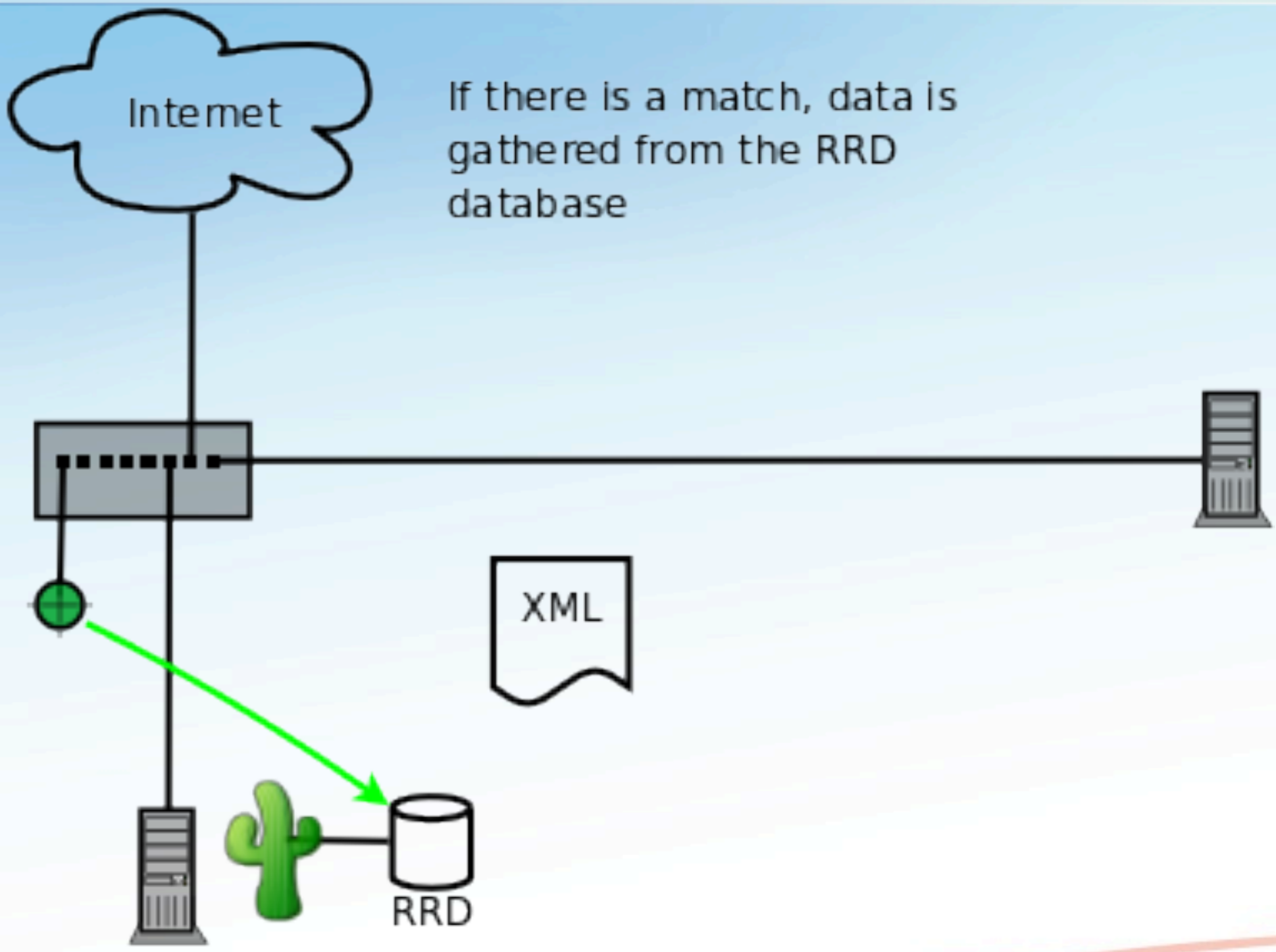
Deploying a Service



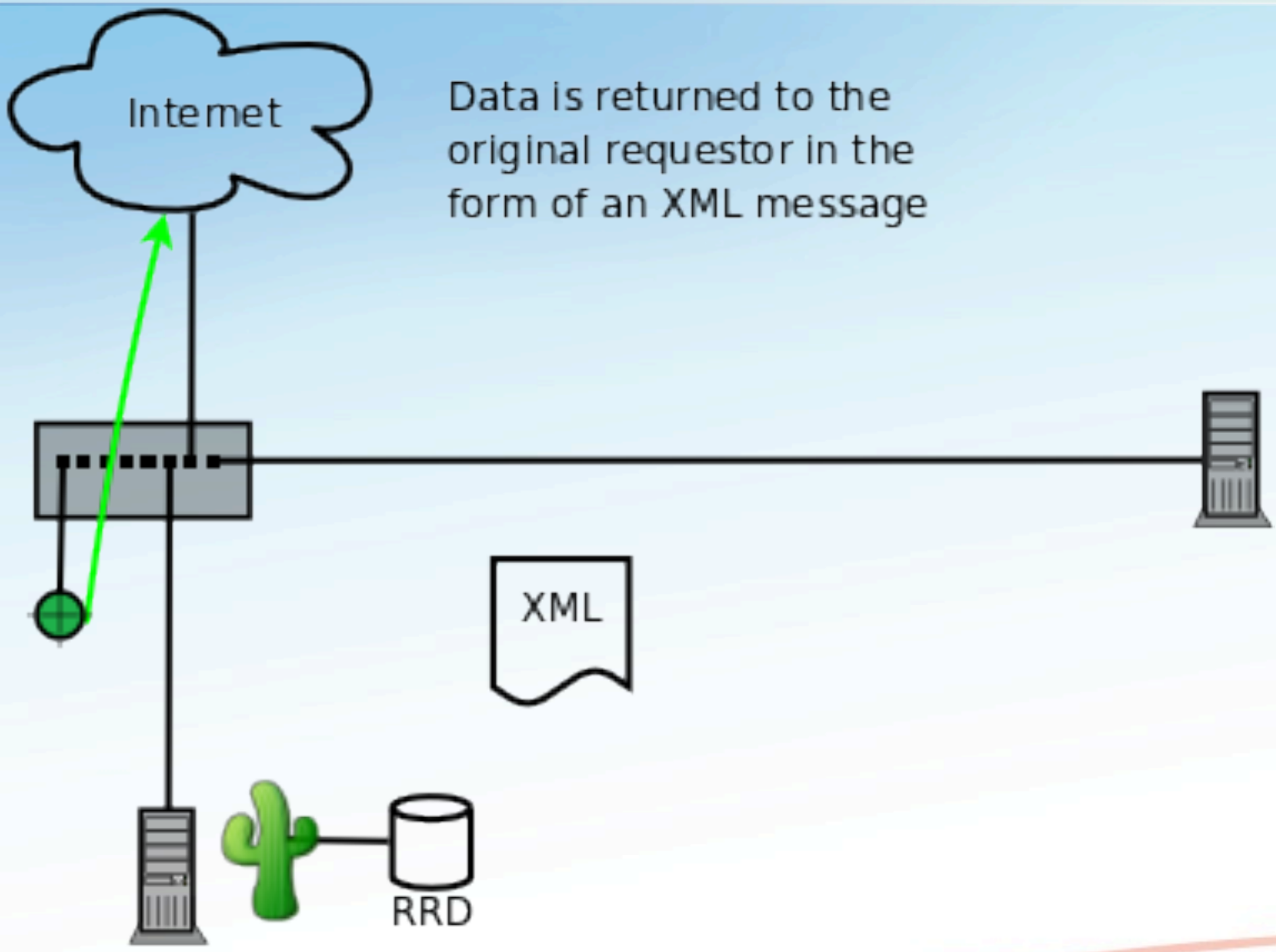
Deploying a Service



Deploying a Service



Deploying a Service

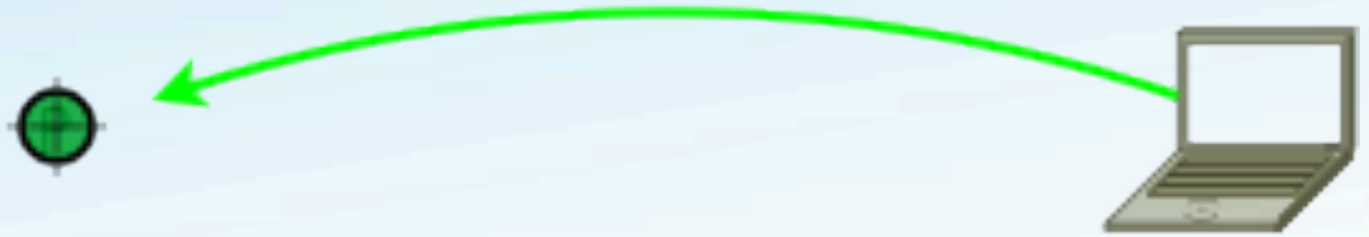


Client/Service Interaction

- 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

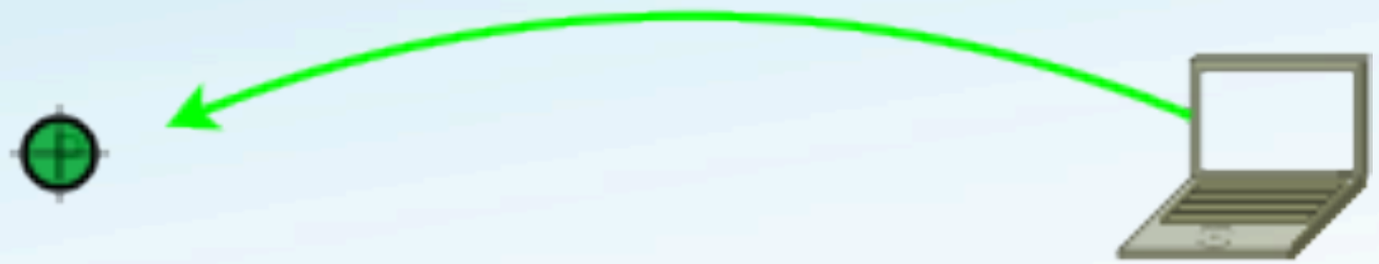
Service sends EchoResponse to verify

Client/Service Interaction

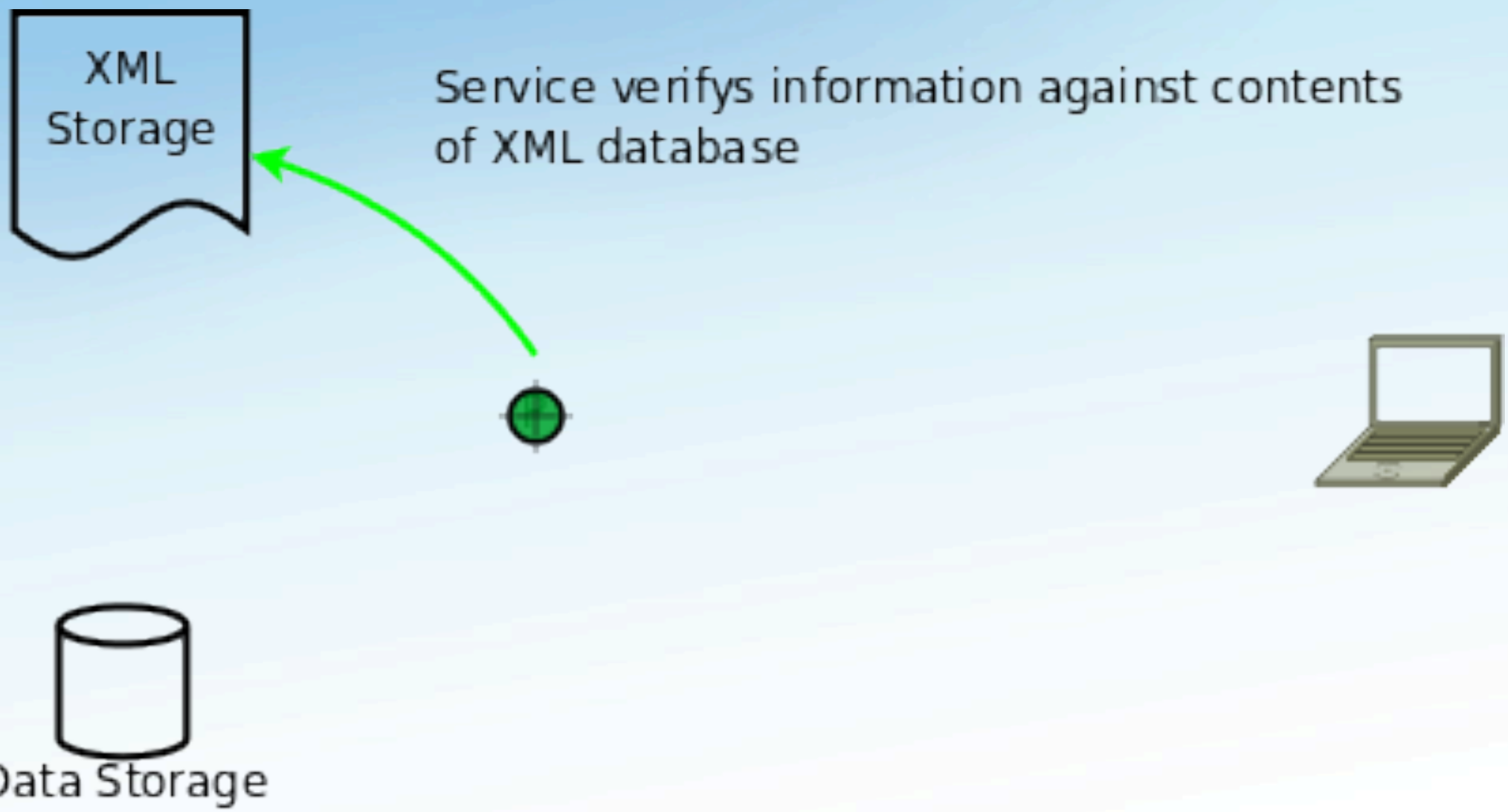
- MetadataKeyRequest
  - For a given (partial) metadata, ask the service to verify that it does or does not exist
  - Return a ‘key’, e.g. replayable token, to access the data

Client/Service Interaction

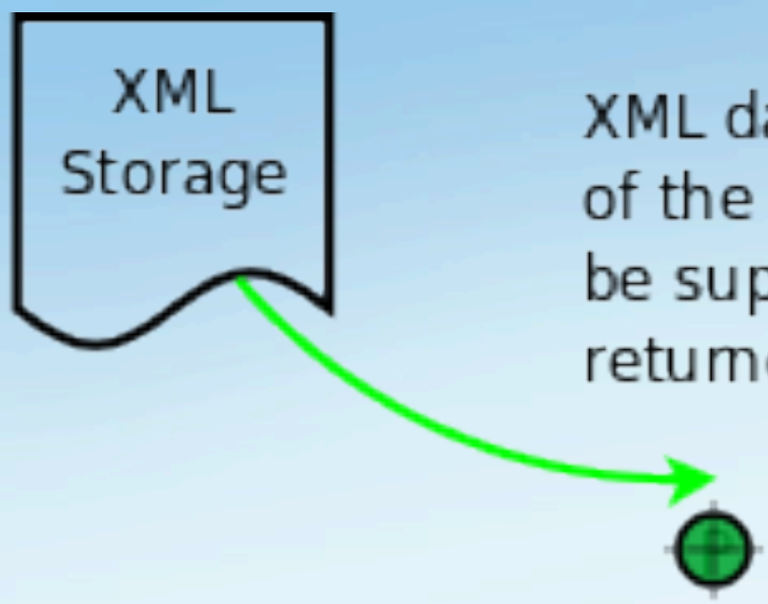
Client sends MetadataKeyRequest to check the status of a specific interface



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

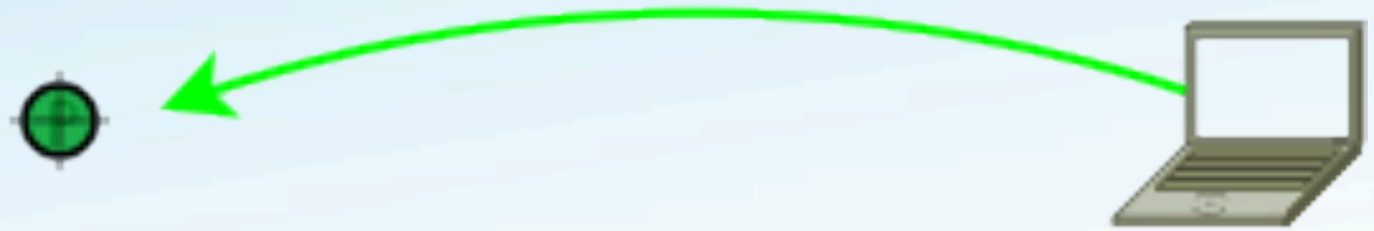
Service returns MetadataKeyResponse to client.

Client/Service Interaction

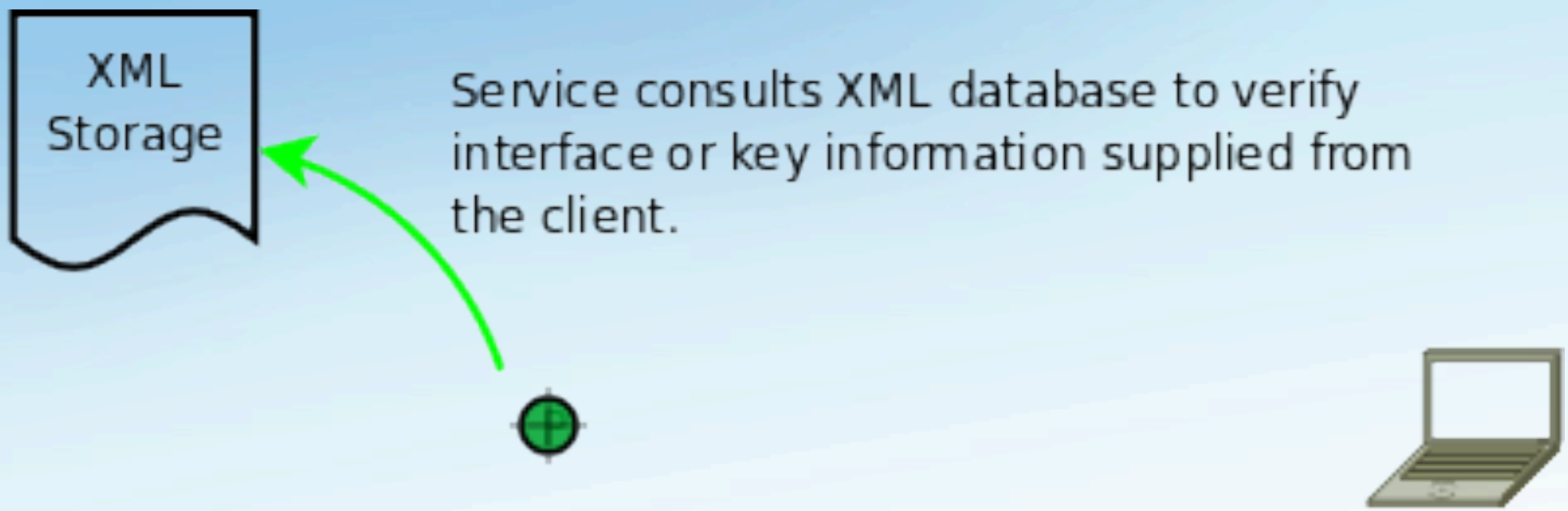
- SetupDataRequest
  - Given a key or (partial) metadata, return measurement information.
  - Can be ‘filtered’ by time to prevent getting more results than necessary.

Client/Service Interaction

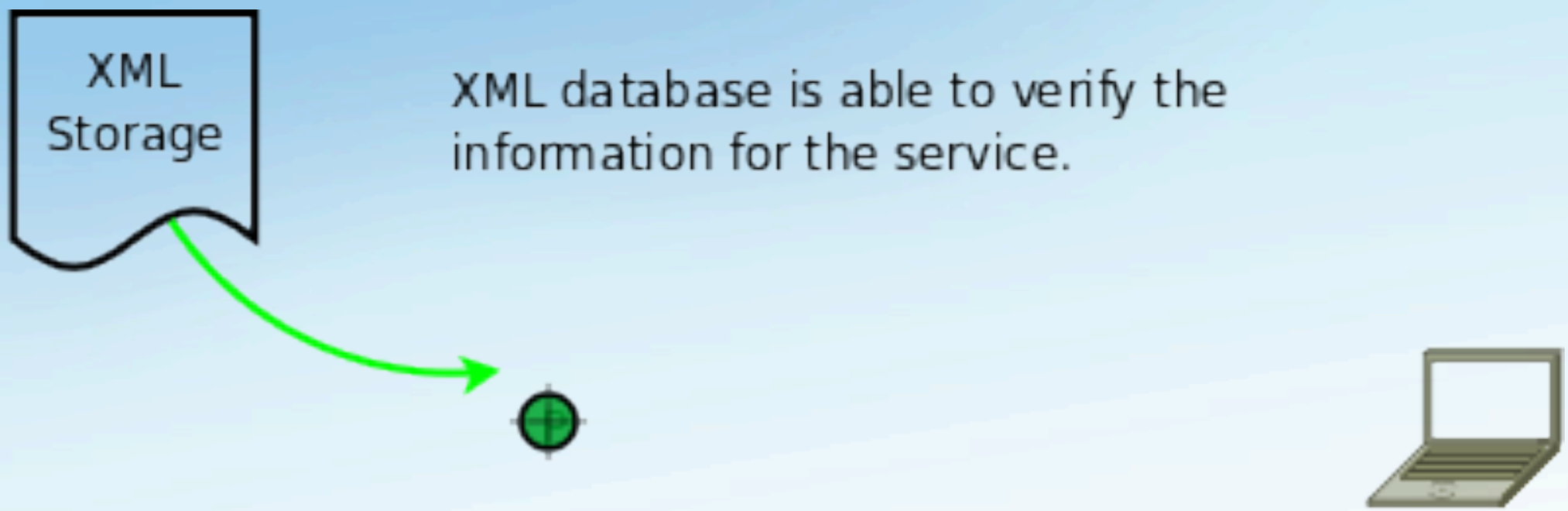
Client sends SetupDataRequest looking for interface data. Client may supply a key or metdata similar to a MetadataKeyRequest



Client/Service Interaction



Client/Service Interaction

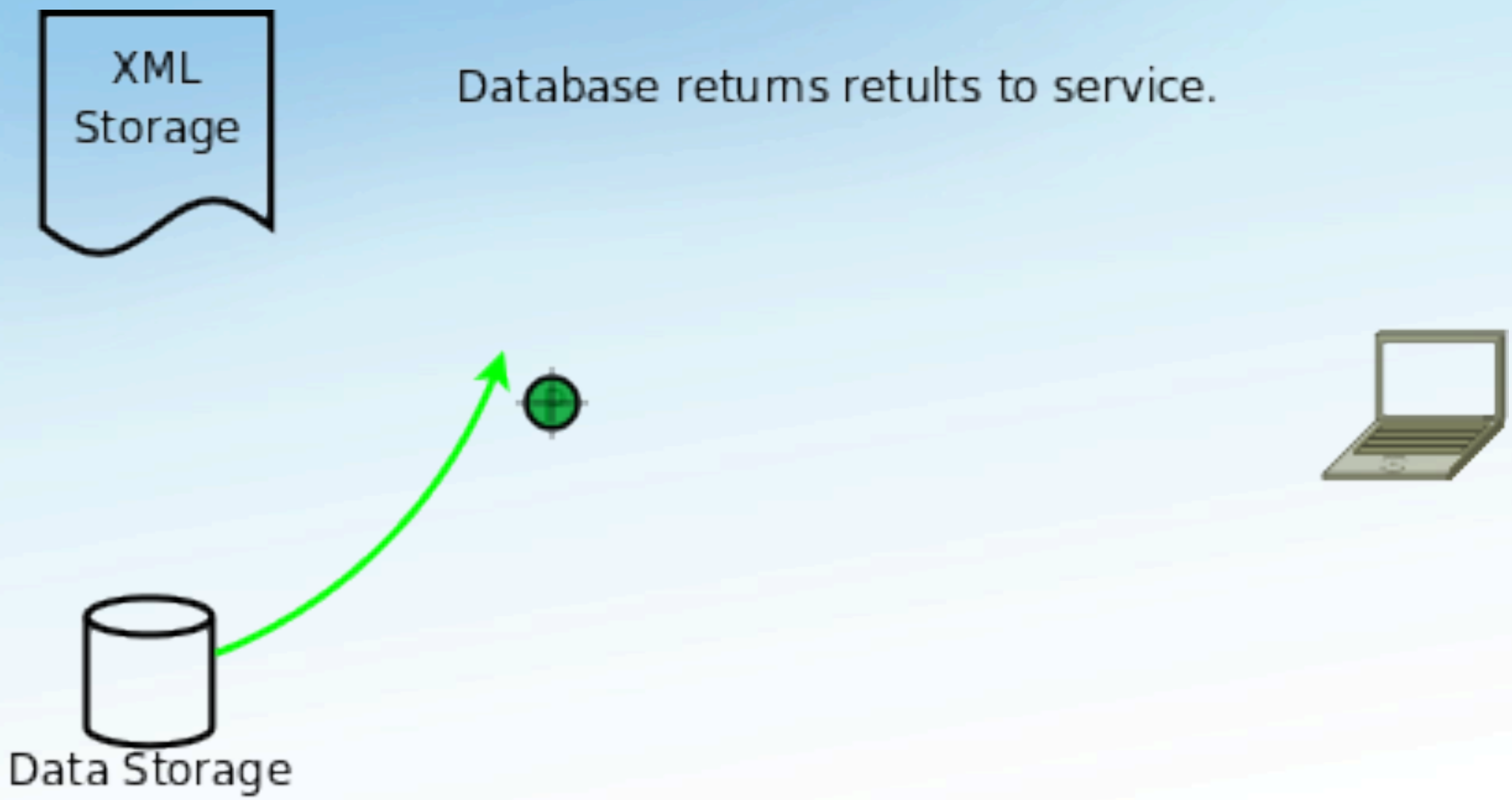


Client/Service Interaction

XML Storage

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

Client/Service Interaction



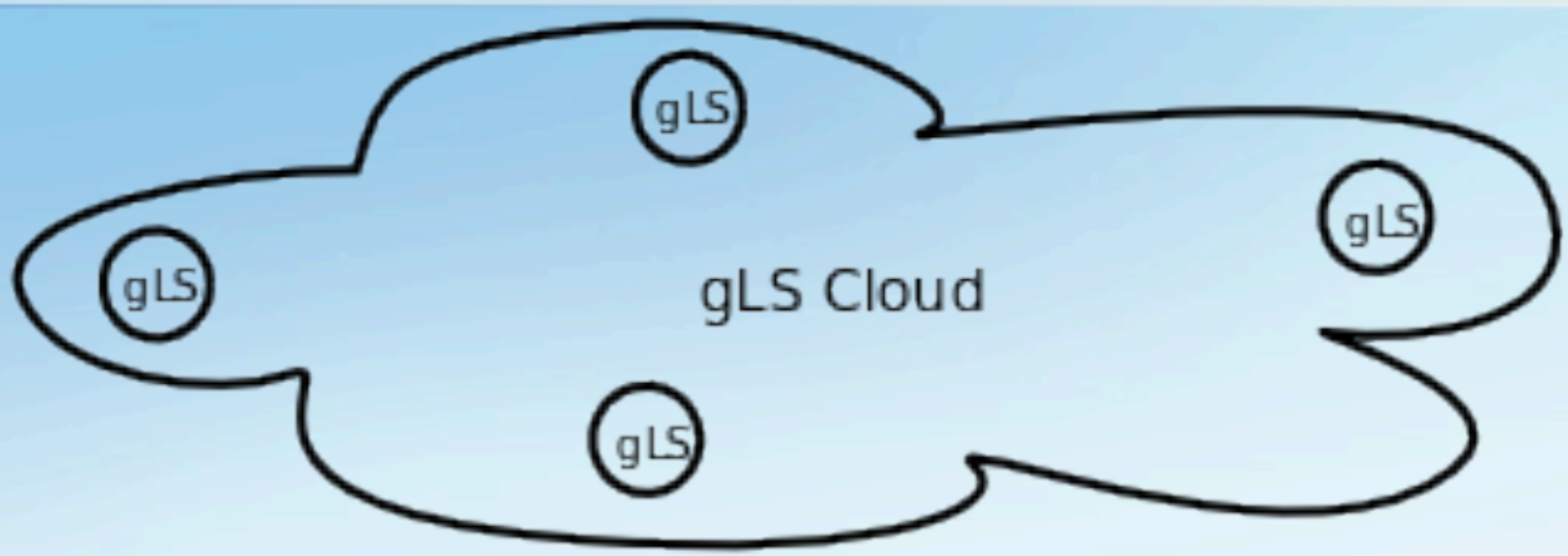
Client/Service Interaction

Service prepares SetupDataResponse for client.

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



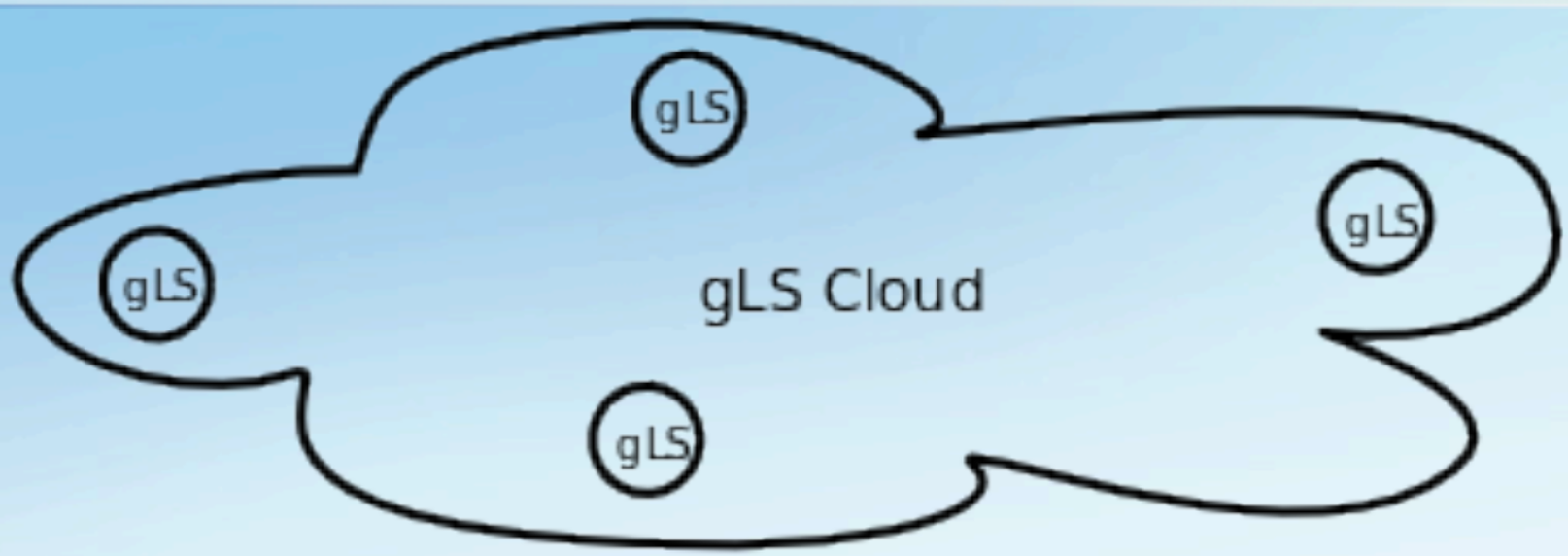
Initial Network, a domain has deployed some perfSONAR services and an hLS

hLS

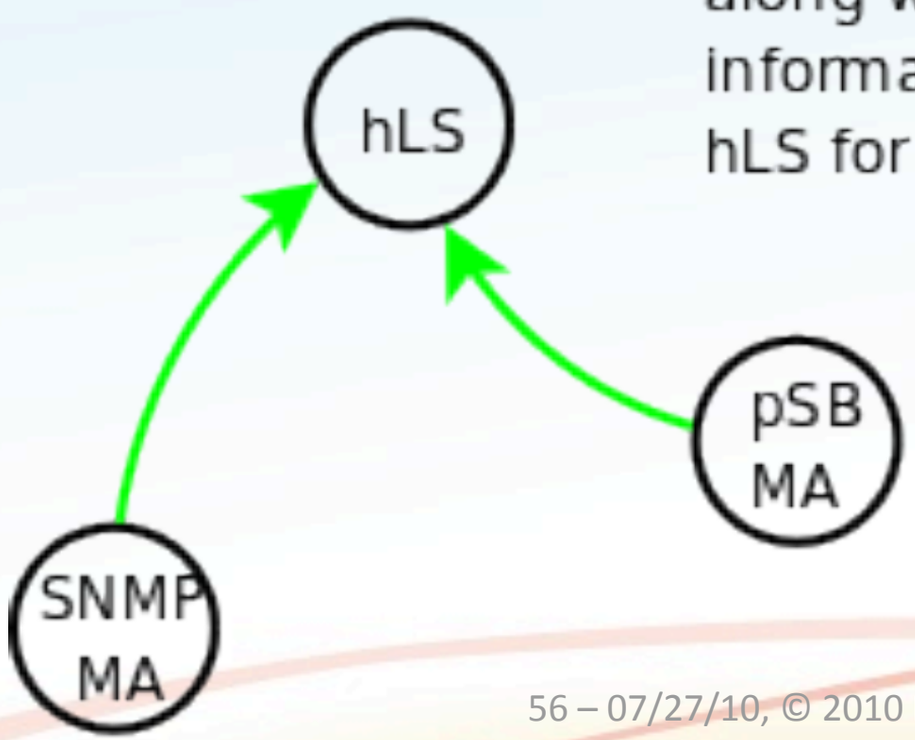
pSB MA

SNMP MA

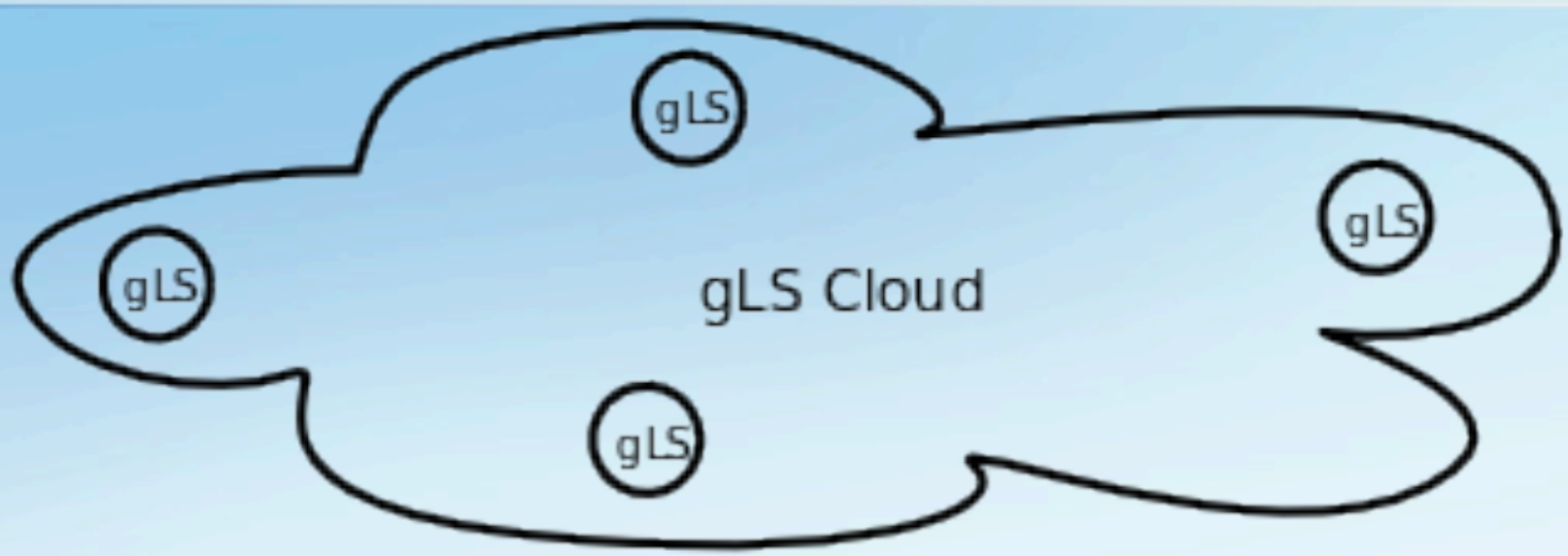
Lookup Service Interaction



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

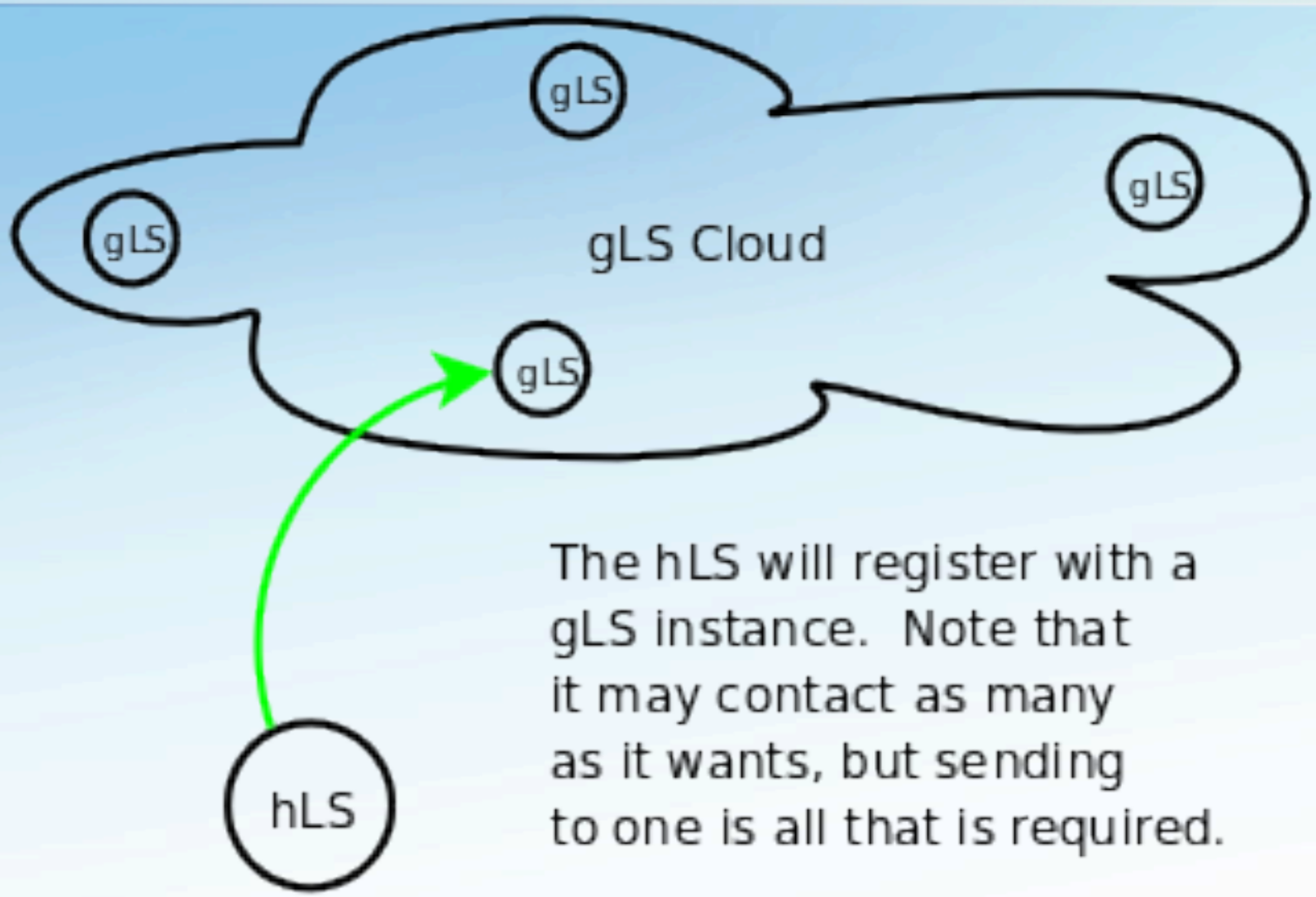


The hLS will 'summarize' the information it receives into a format that the gLSs want to see.

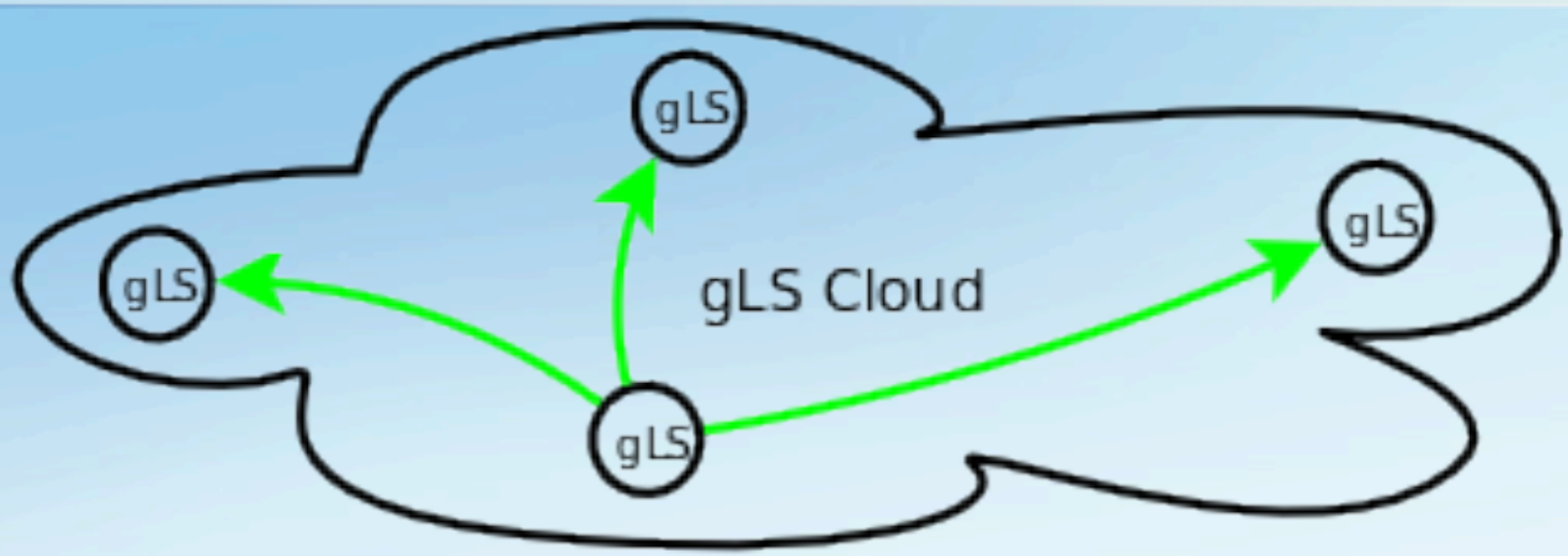
perfSONAR powered

INTERNET 2

Lookup Service Interaction



Lookup Service Interaction

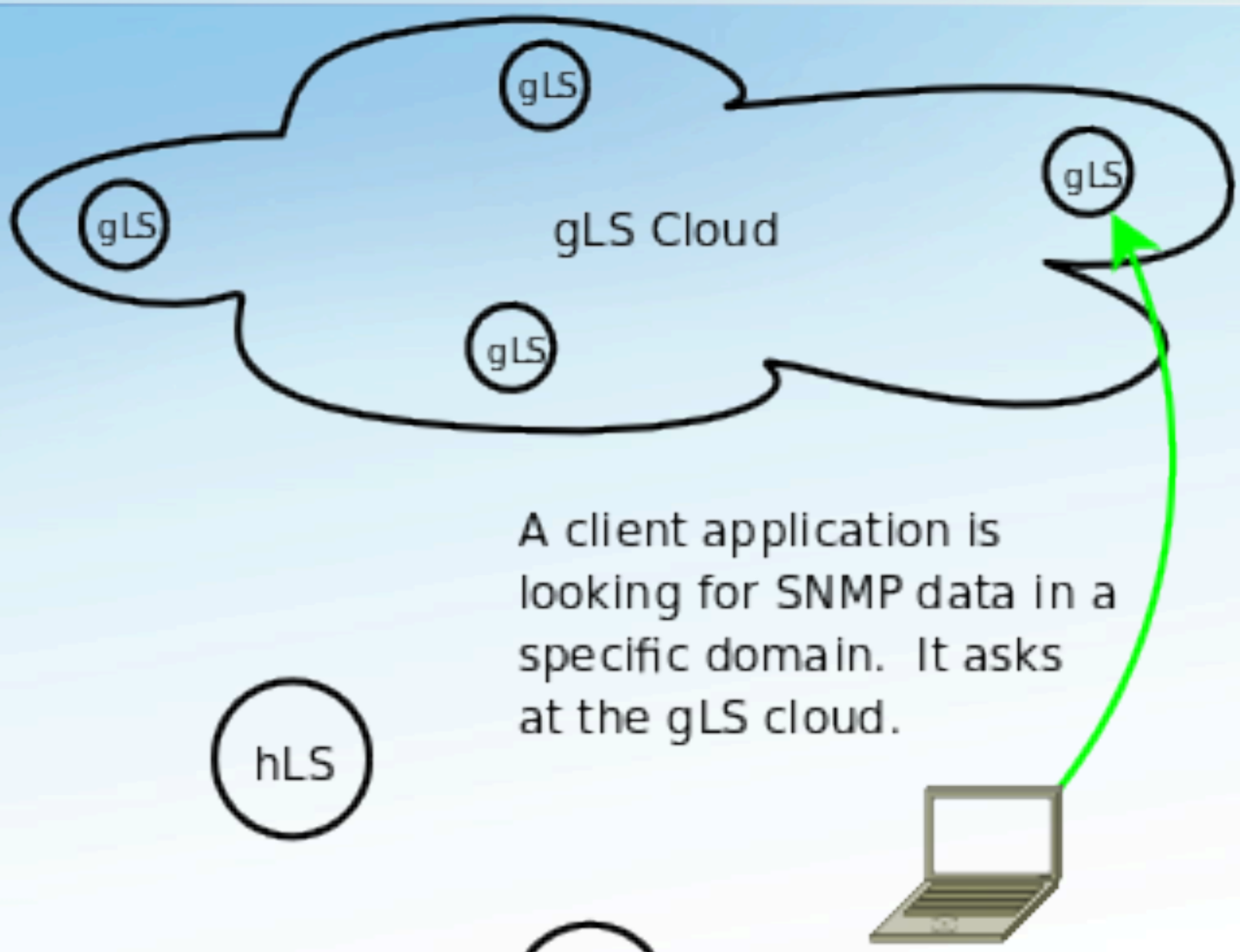


Periodically each gLS will share the hLSs it knows about with the others.

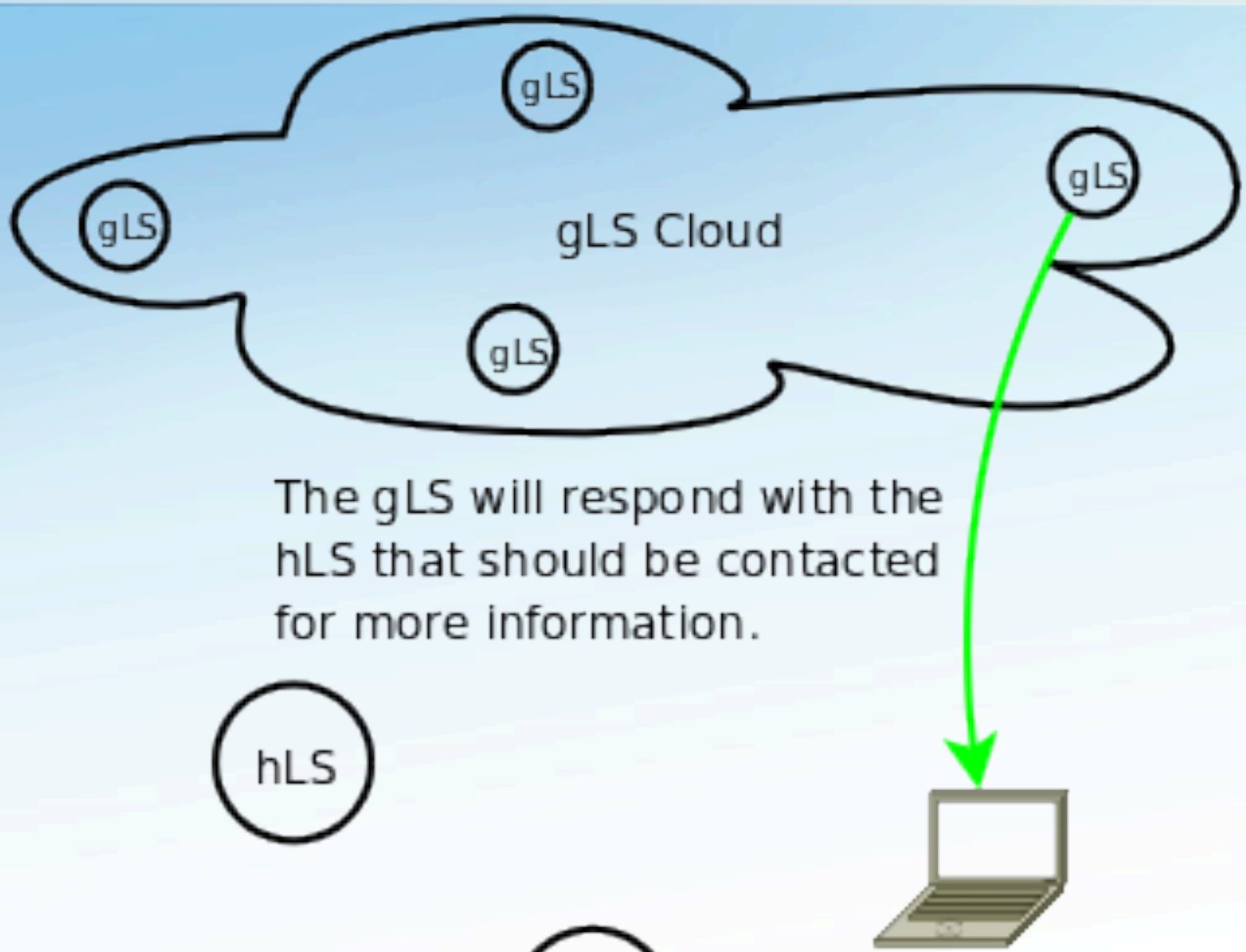
perfSONAR powered

INTERNET 2

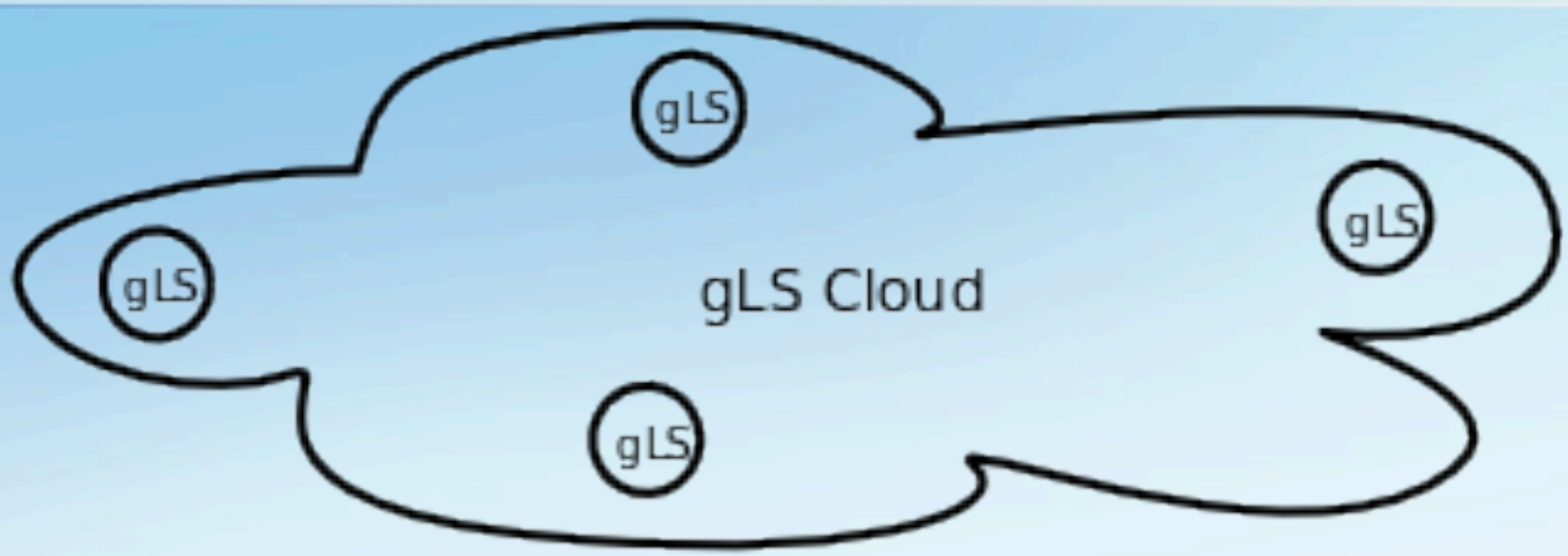
Lookup Service Interaction



Lookup Service Interaction

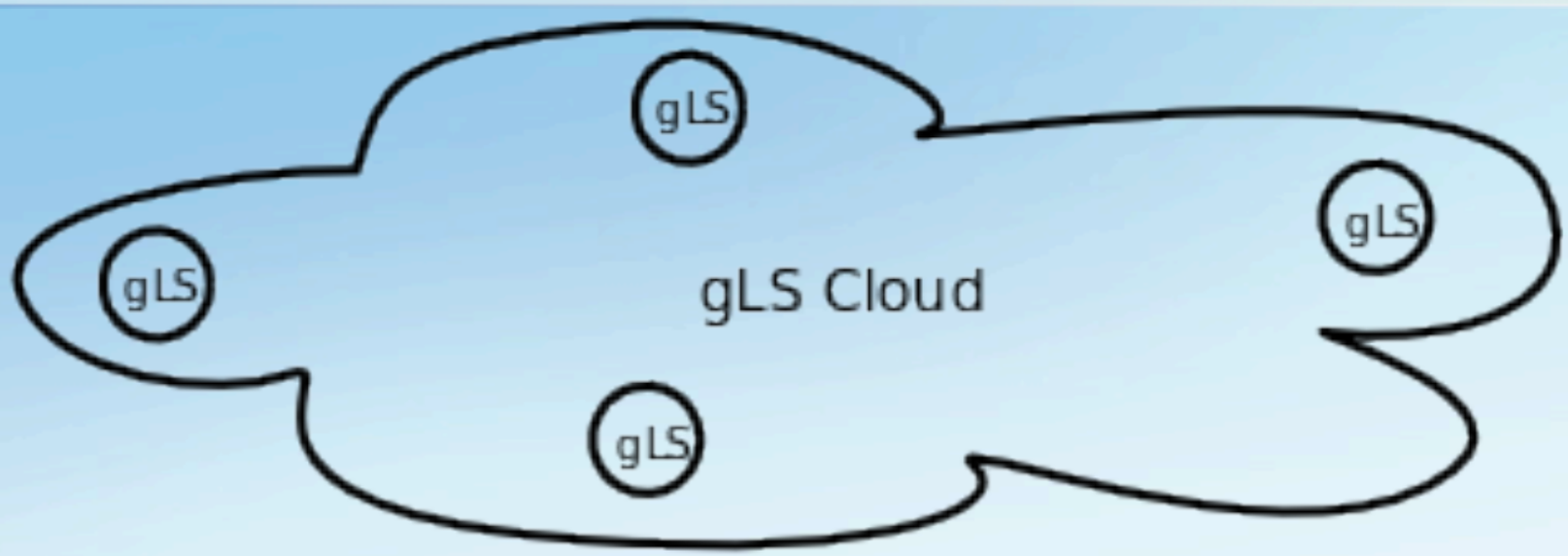


Lookup Service Interaction



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

Lookup Service Interaction

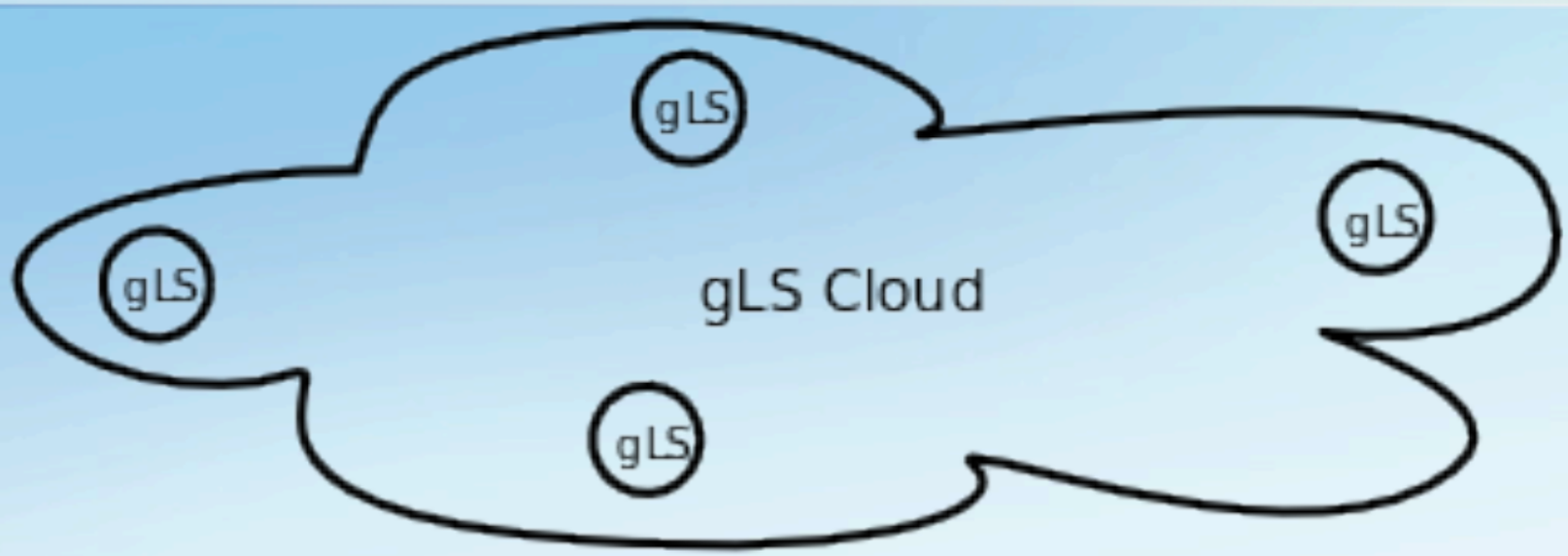


The hLS will respond with the service name that is likely to have the data the client wants.

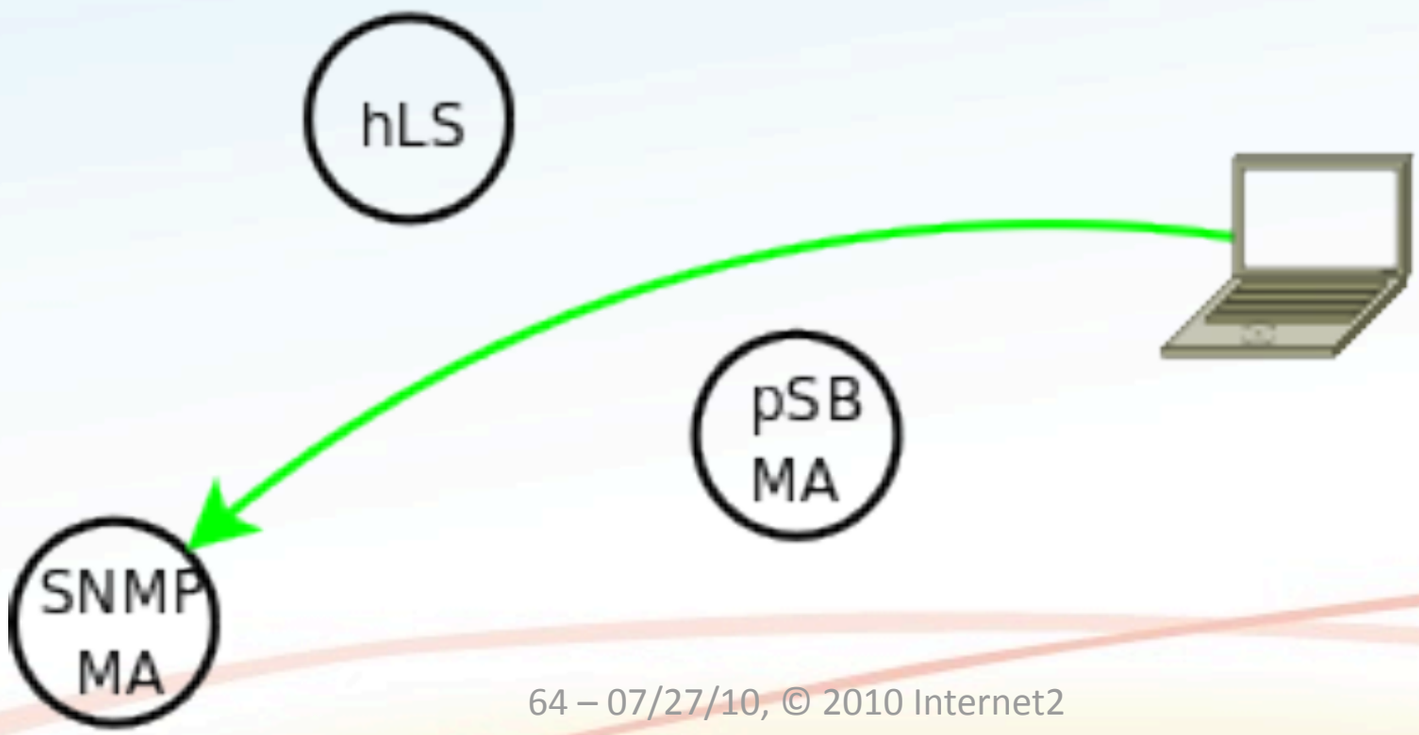
pSB MA

SNMP MA

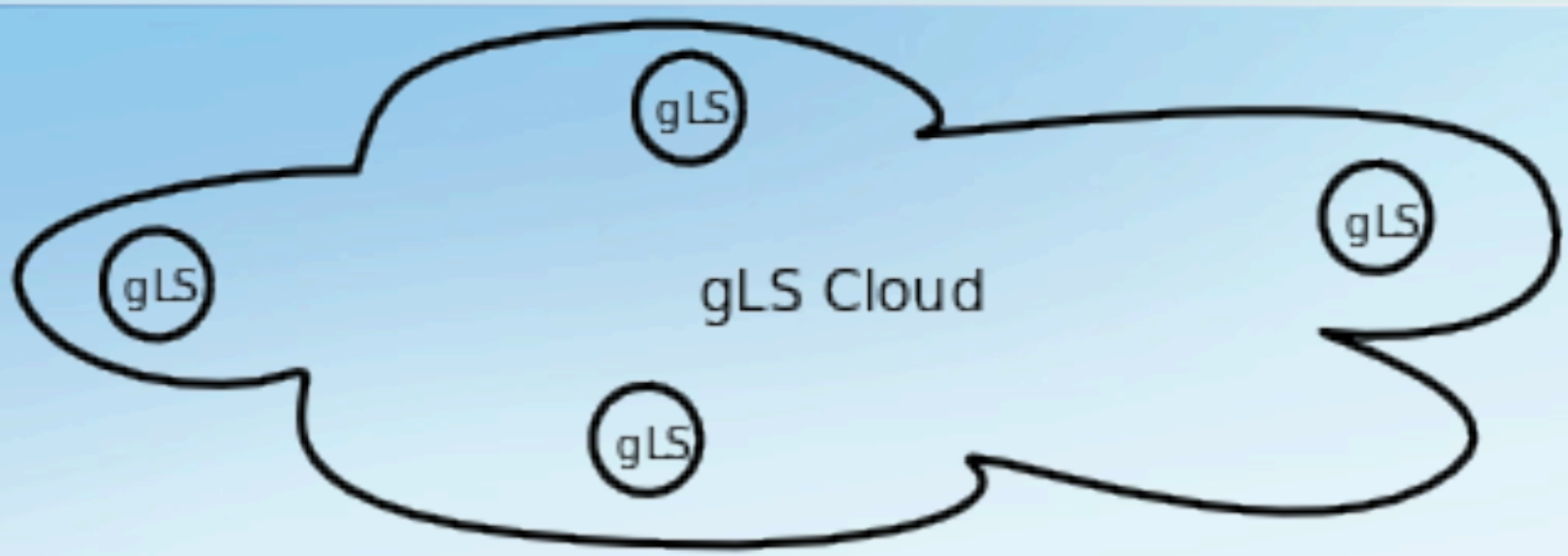
Lookup Service Interaction



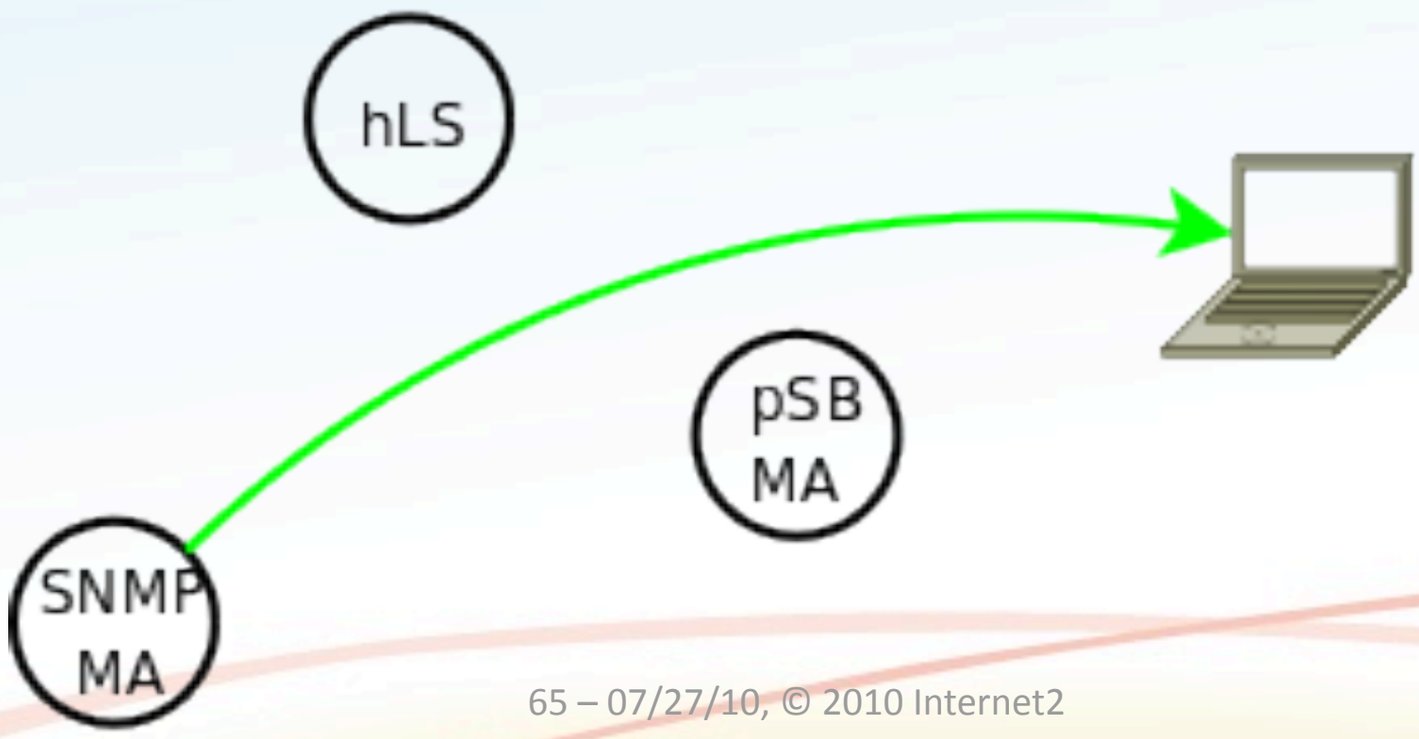
The client will contact the service suggested by the hLS for data.



Lookup Service Interaction



The service returns the data to the client.



Conclusions

- We have seen the Architecture
- We have seen how it works together
- Whats next?
  - Software Availability
  - Protocols
  - Client Design
  - Service Design

perfSONAR Architecture

July 7th 2010, perfSONAR Workshop – perfSONAR Tutorial

Jason Zurawski - Network Software Engineer, Research Liason

For more information, visit http://www.perfsonar.net

INTERNET2

July 7th 2010, perfSONAR Workshop – perfSONAR Tutorial

Jason Zurawski, Network Software Engineer, Research Liason

perfSONAR Use Cases

Outline

- Motivation
- How it ***Should*** Work
- How it ***Probably*** Works
- Use Cases
  - Cisco Telepresence
- Questions for Later in the Workshop…
- Reference Material (Not Reviewed)
  - Why is Science Data Different?
  - Identifying Common Network Problems
  - Additional Use Cases
    - Georgetown International Campus
    - USATLAS
    - REDDnet

Motivation

- Now that we have seen the purpose and makeup of the *perfSONAR* infrastructure, it's time to see what it can do in the real world
- *perfSONAR* is used by network engineers to identify many types of performance problem
  - A ***Divide and Conquer*** strategy is necessary to isolate problems
  - A ***structured methodology*** helps to eliminate duplicate or useless steps
  - *perfSONAR* works best when everyone participates, holes in deployment lead to holes in the problem solving phase
- The following sections will outline the proper deployment strategy and describe some real work use cases

How it ***Should*** Work

- To accurately and swiftly address network performance problems the following steps should be undertaken
  - Identify the problem: if there a user in one location is complaining about performance to another, get as much information as possible
    - Is the problem uni-directional? Bi-directional?
    - Does the problem occur all the time, frequently, or rarely?
    - Does the problem occur for only a specific application, many applications, or only some applications?
    - Is the problem reproducible on other machines?
  - Gather information about the environment
    - Hosts
    - Network Path
    - Configuration (where applicable)
    - Resources available

How it ***Should*** Work

- Cont.
  - Methodically approach the problem
    - Test using the same tool everywhere, gather results
    - Before moving on to the next tool, did you gather everything of value?
    - Are the results consistent?
  - After proceeding through all tools and approaches, form theories
    - Can the problem be isolated to a specific resource or component?
    - Can testing be performed to eliminate dead ends?
- Consider the following example:
  - International path
  - Problems noted
  - We know the path
  - We have tools available

Scenario: Multi-domain International Path

perfSONAR powered

INTERNET 2

Desirable Case: Expected Performance

perfSONAR powered

INTERNET 2

Typical: Poor Performance … Somewhere

perfSONAR powered

INTERNET 2

Typical: Poor Performance … Somewhere

perfSONAR powered

INTERNET 2

Solution: Test Points + Regular Monitoring

perfSONAR powered

INTERNET 2

perfSONAR: Backbone and Exchanges

perfSONAR: Regional Networks

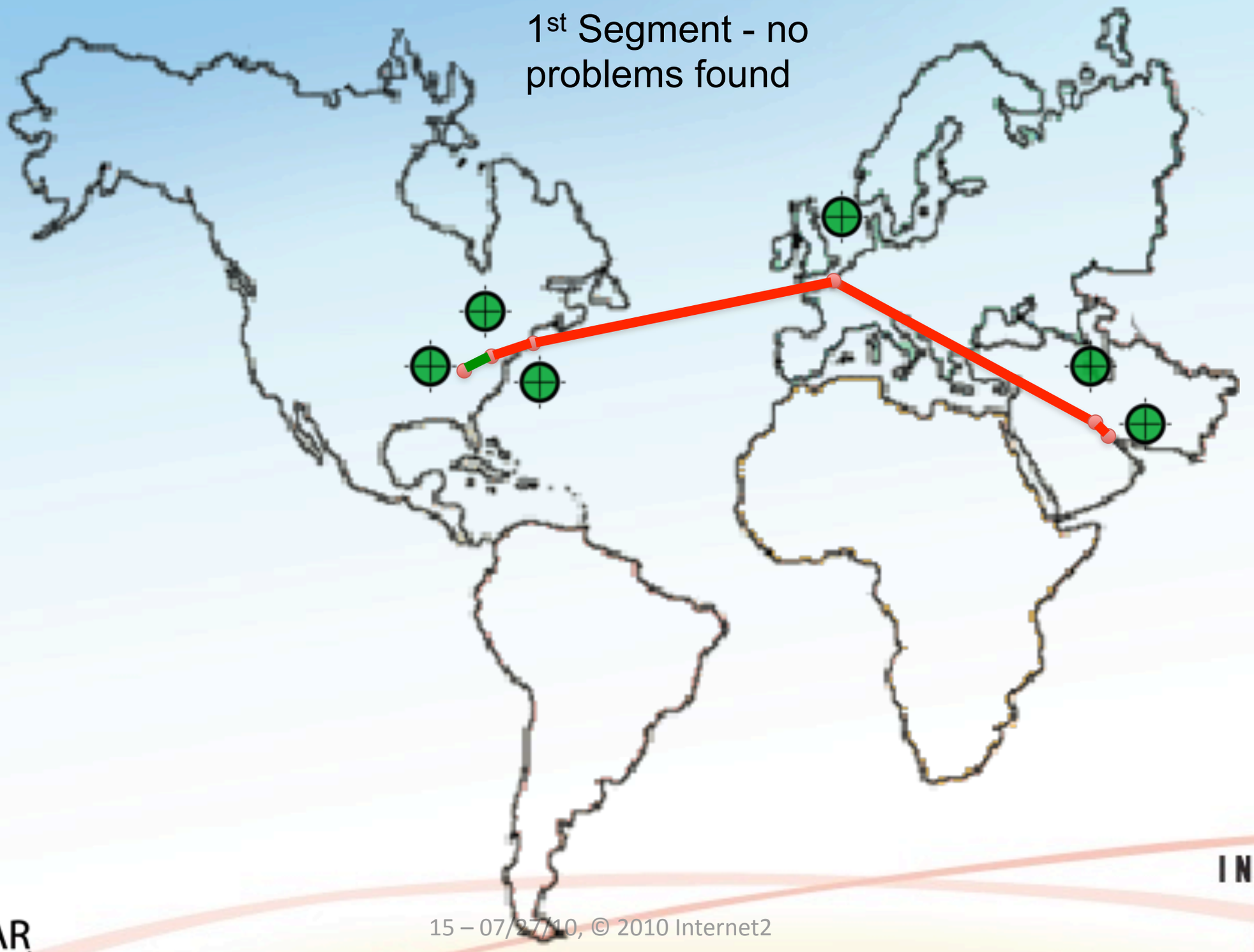
perfSONAR: Campus

perfSONAR powered

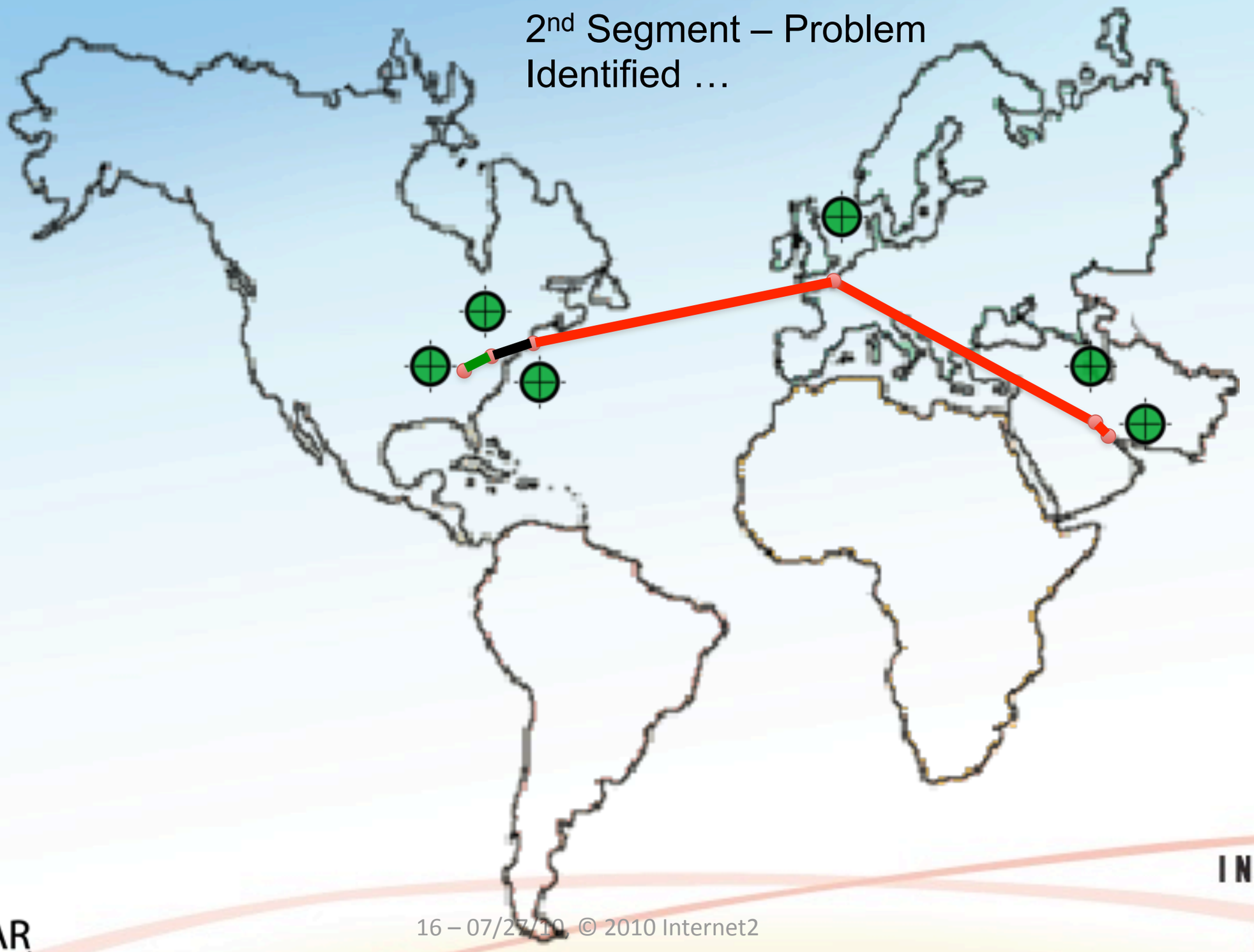
INTERNET 2

Path Decomposition – Isolate the Problem

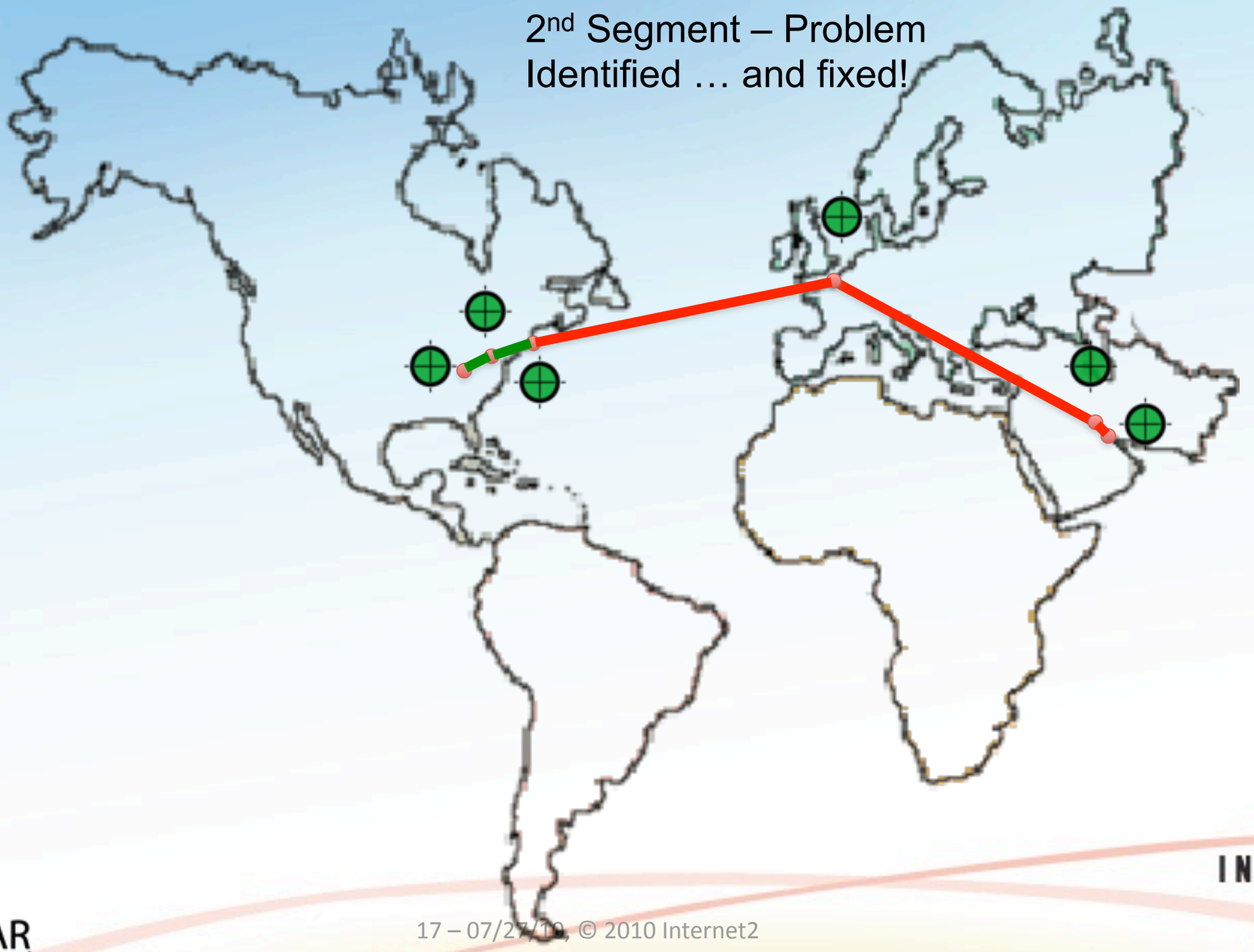
Path Decomposition – Isolate the Problem



Path Decomposition – Isolate the Problem



Path Decomposition – Isolate the Problem

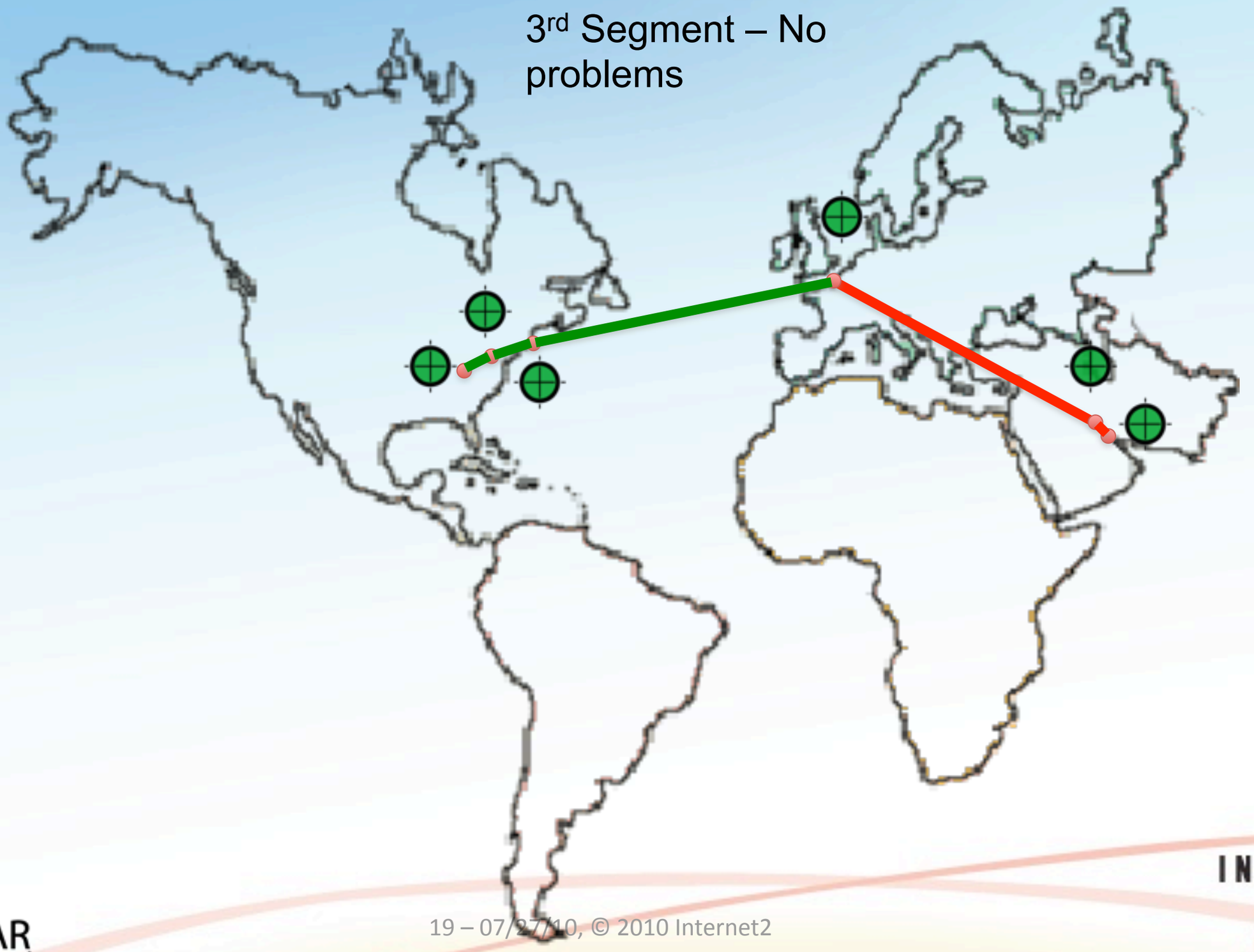


perfSONAR powered

INTERNET 2

Path Decomposition – Isolate the Problem

Path Decomposition – Isolate the Problem



Path Decomposition – Isolate the Problem

Path Decomposition – Isolate the Problem

Path Decomposition – Isolate the Problem

perfSONAR powered

INTERNET 2

Lessons Learned

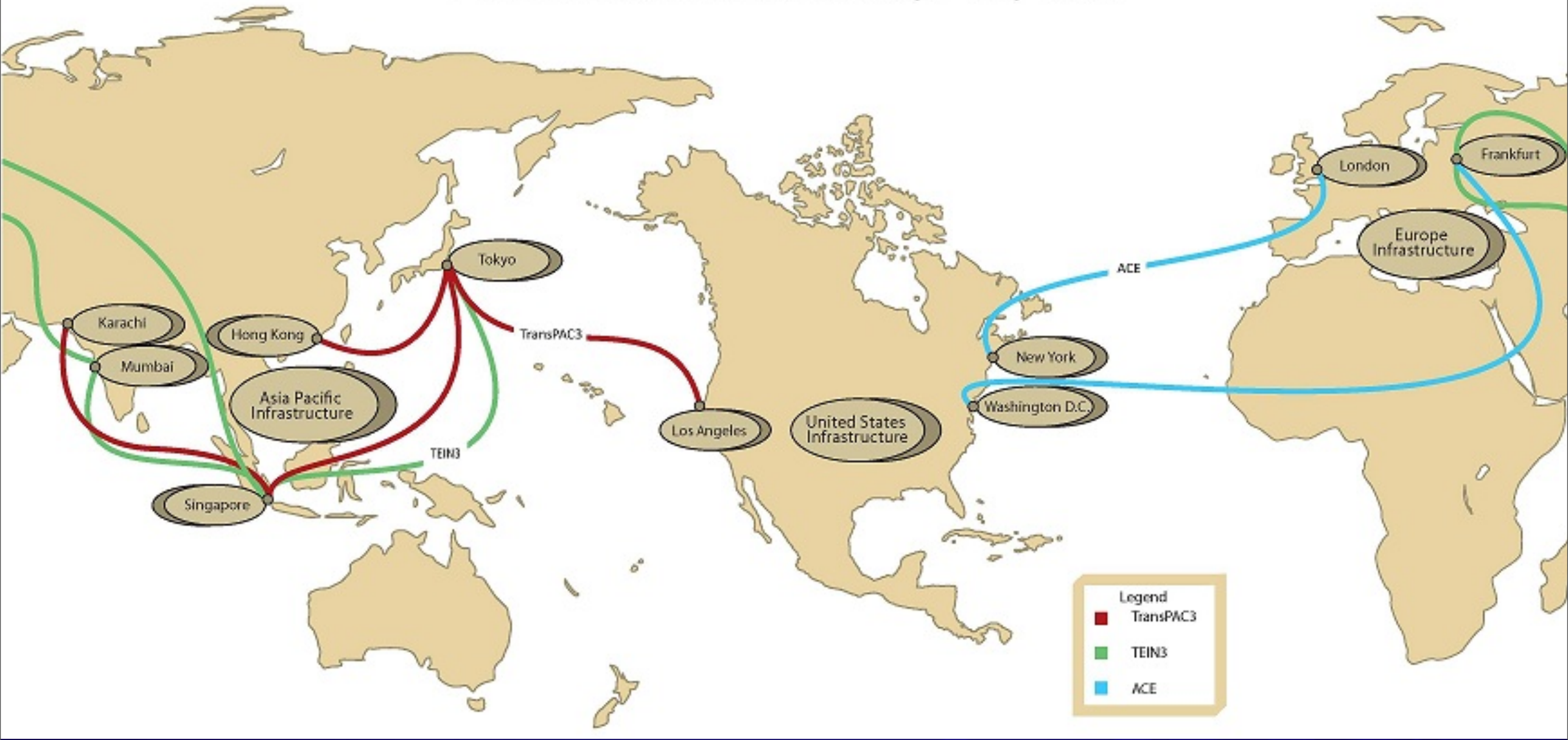
- Problem resolution requires proper tools
  - Specialized to given task (e.g. Bandwidth, Latency)
  - Widely available where the problems will be
- Isolating a problem is a well defined, multi-step process
  - Rigid set of steps – systematic approach to prevent causing new problems
- Diagnostics, as well as regular monitoring, can reveal true network performance

How it ***Probably*** Works

- If the suggested steps aren't taken (or followed in an ad-hoc manner), results will vary.
  - Skipping steps leads to missing clues
- Deployment and participation may vary, this leads to some gaps in the debugging process
- Consider the following example:
  - International path
  - Problems noted
  - We know the path
  - We have tools available - almost everywhere

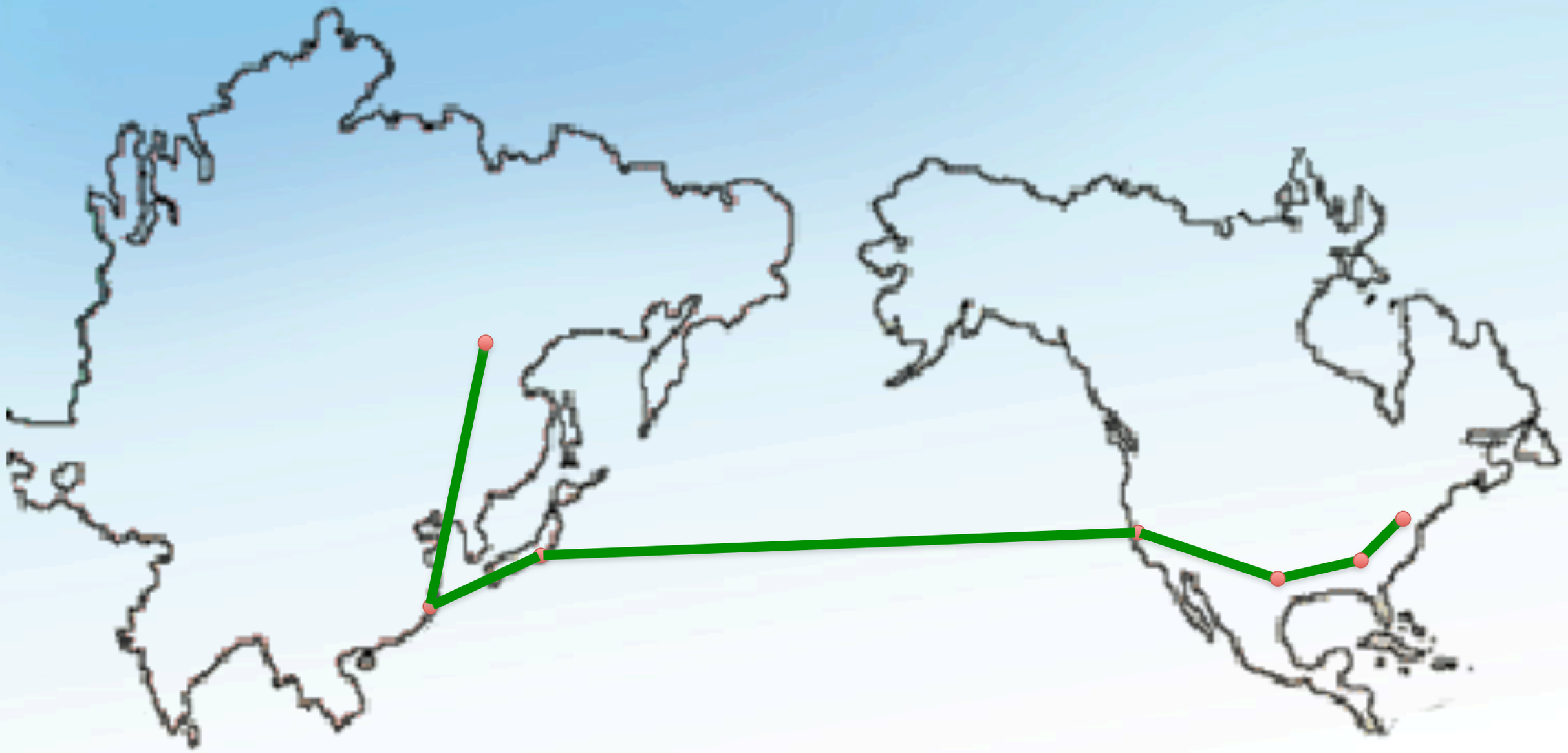
US-ACE-GN3-TEIN3-TP3

Possible Network Connectivity 1-July-2011



Scenario: Multi-domain International Path

Desirable Case: Expected Performance



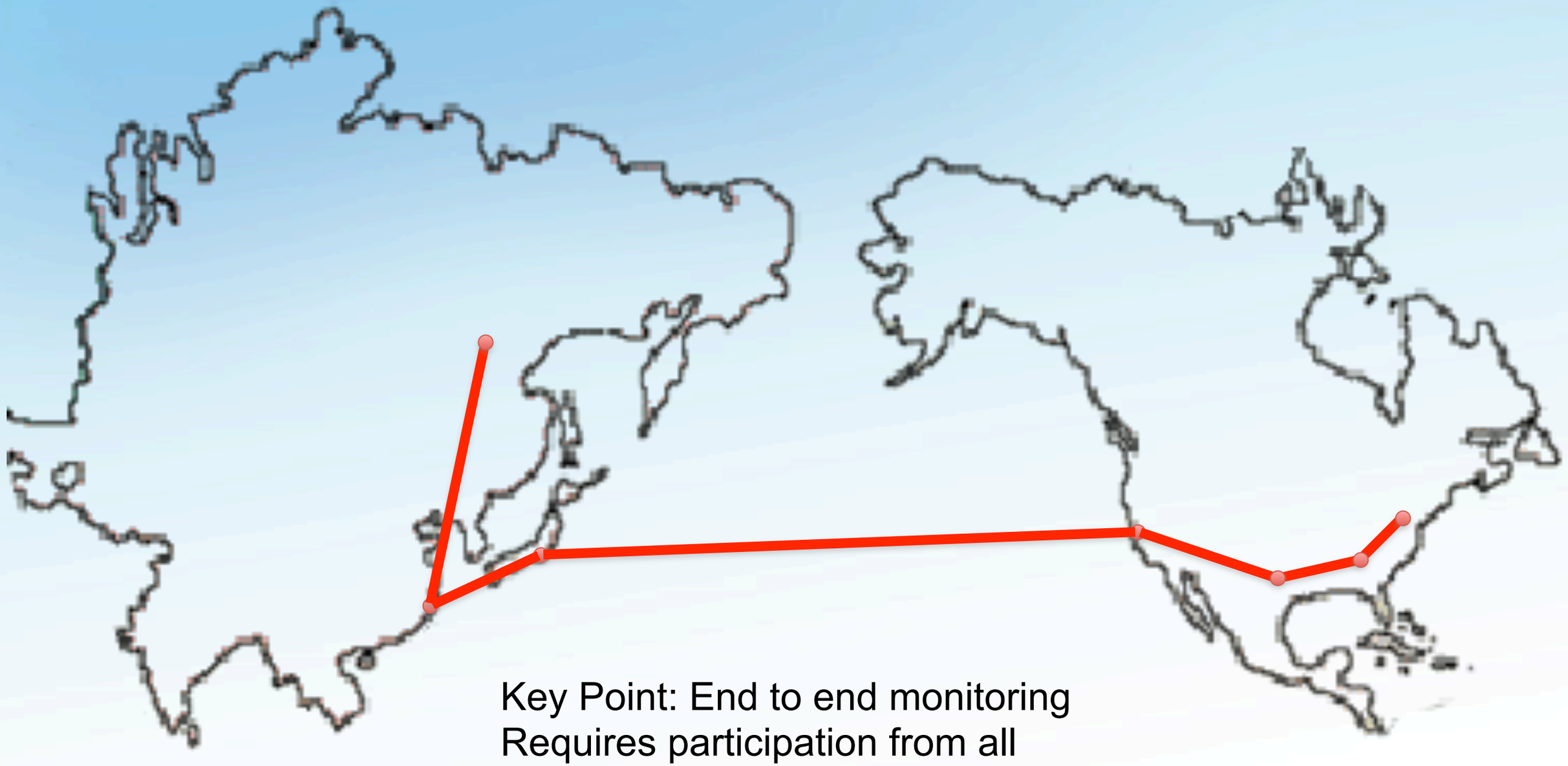
Typical: Poor Performance … Somewhere

Typical: Poor Performance … Somewhere

But where?

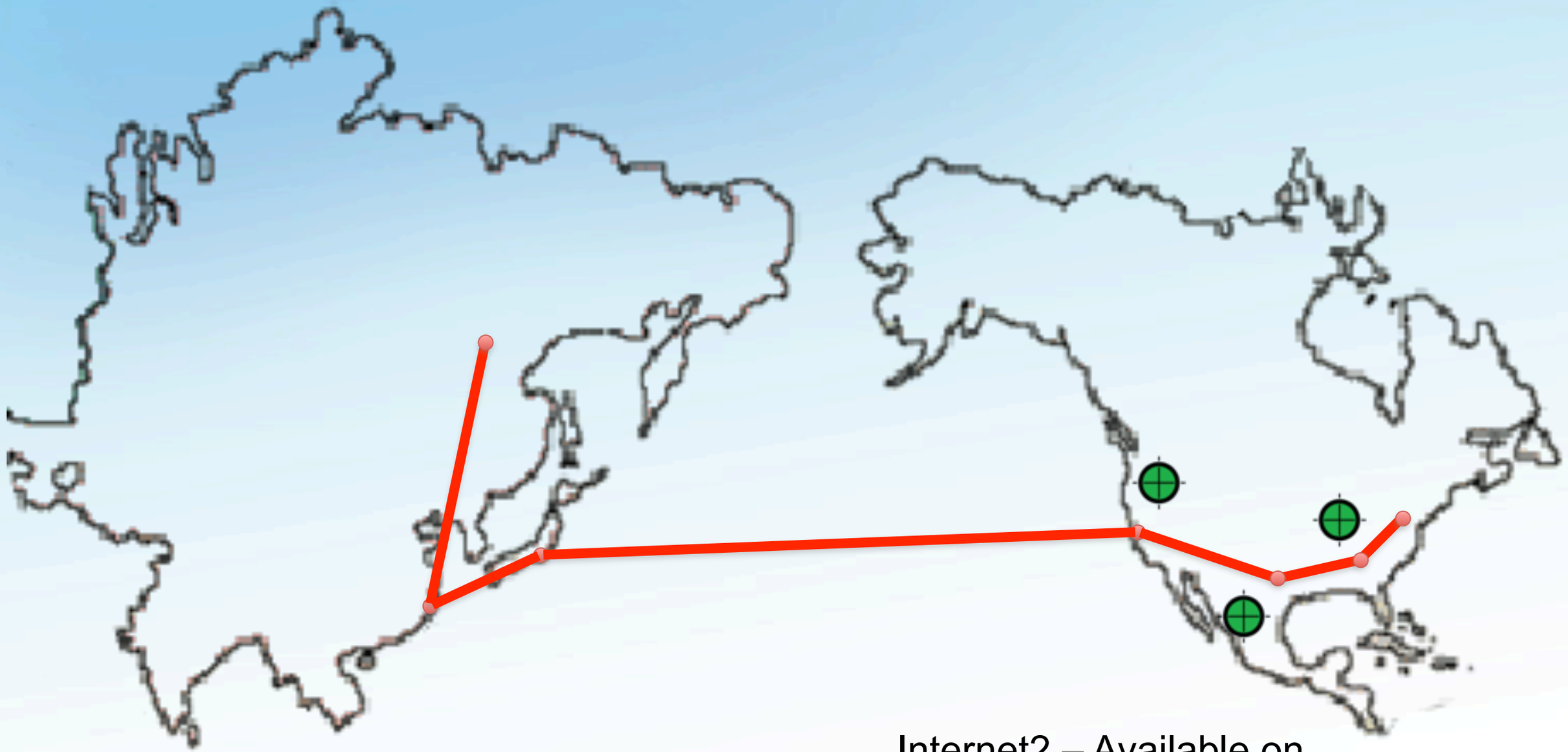
Solution: Test Points + Regular Monitoring

Solution: Test Points + Regular Monitoring



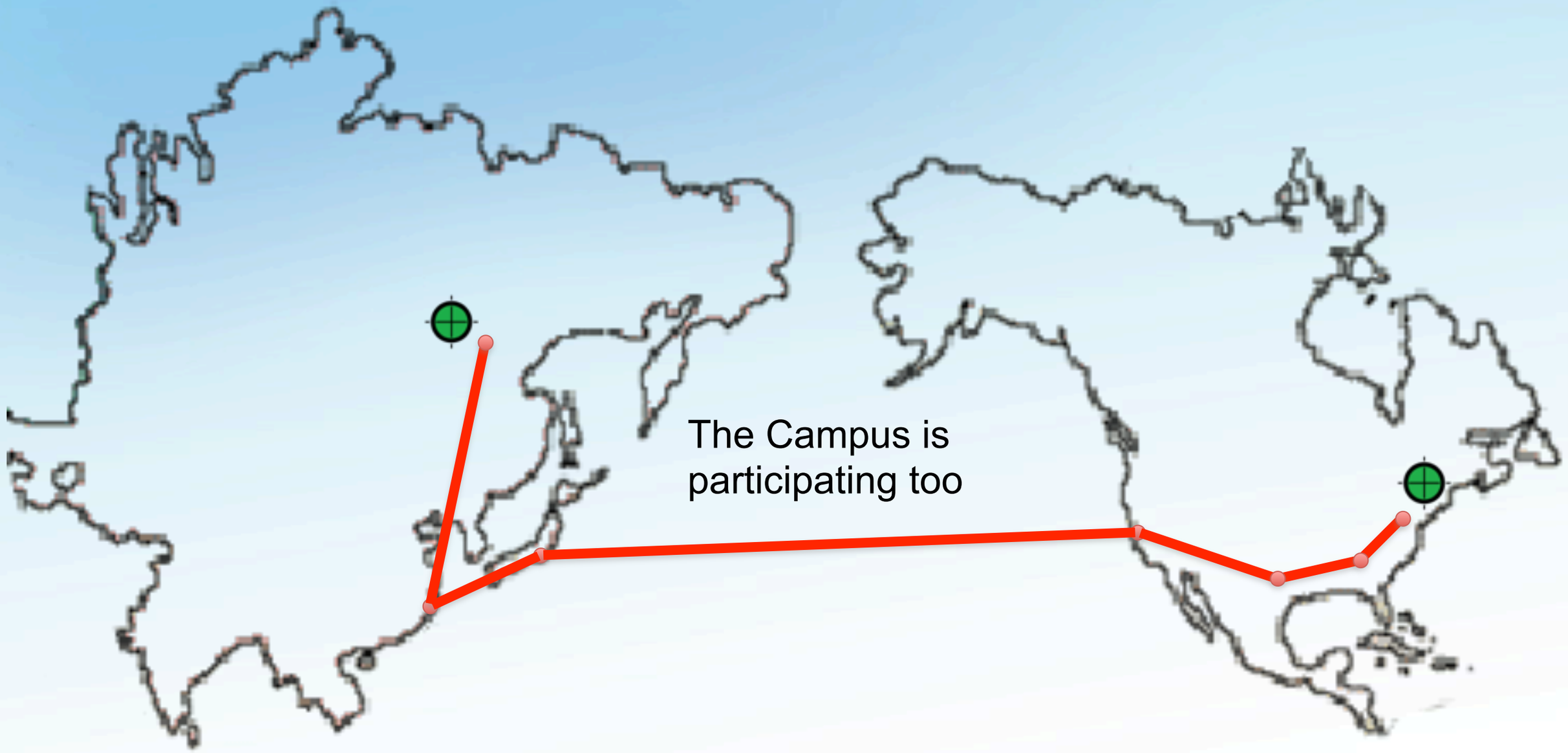
Key Point: End to end monitoring
Requires participation from all domains

Typical: Poor Performance … Somewhere

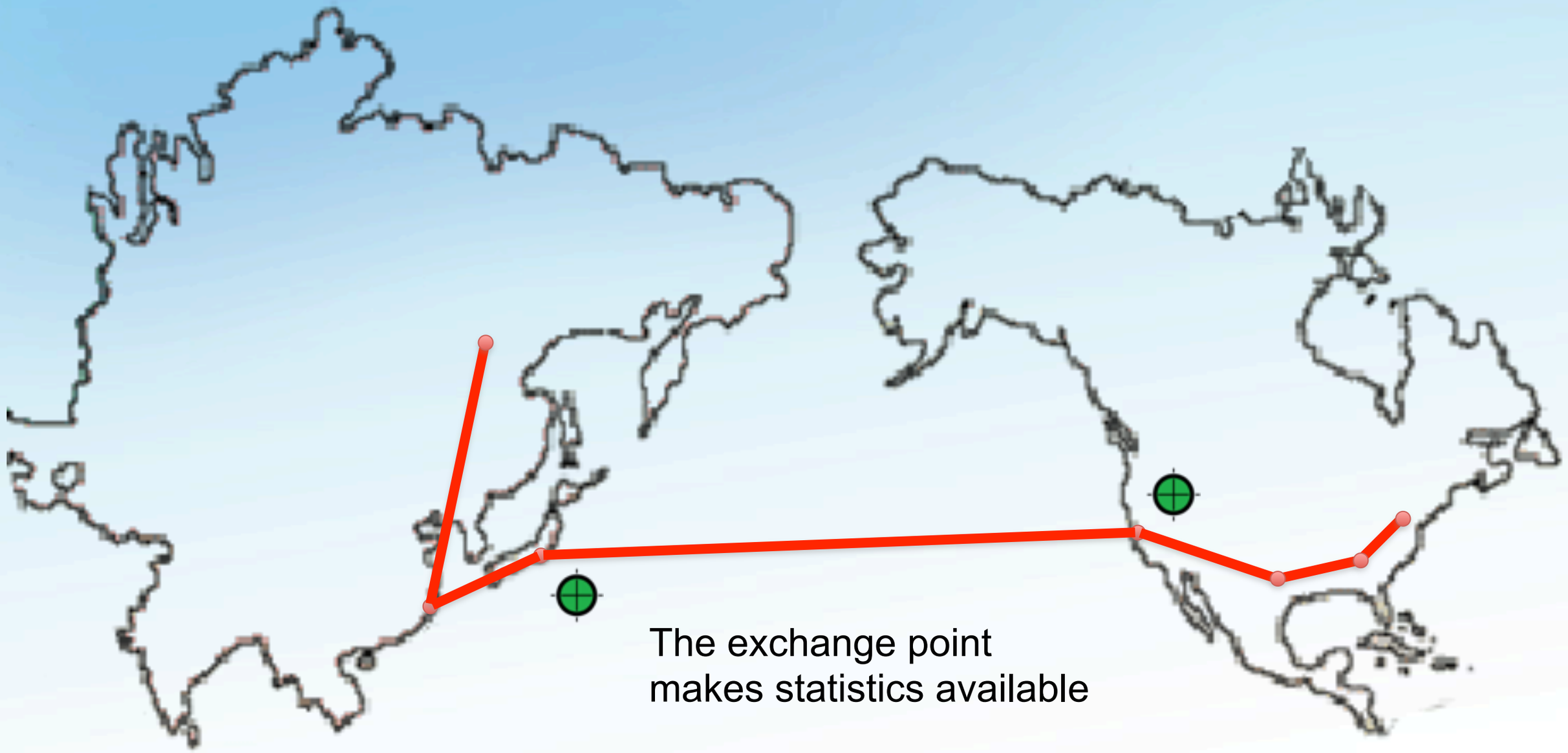


Internet2 – Available on the backbone

Typical: Poor Performance … Somewhere

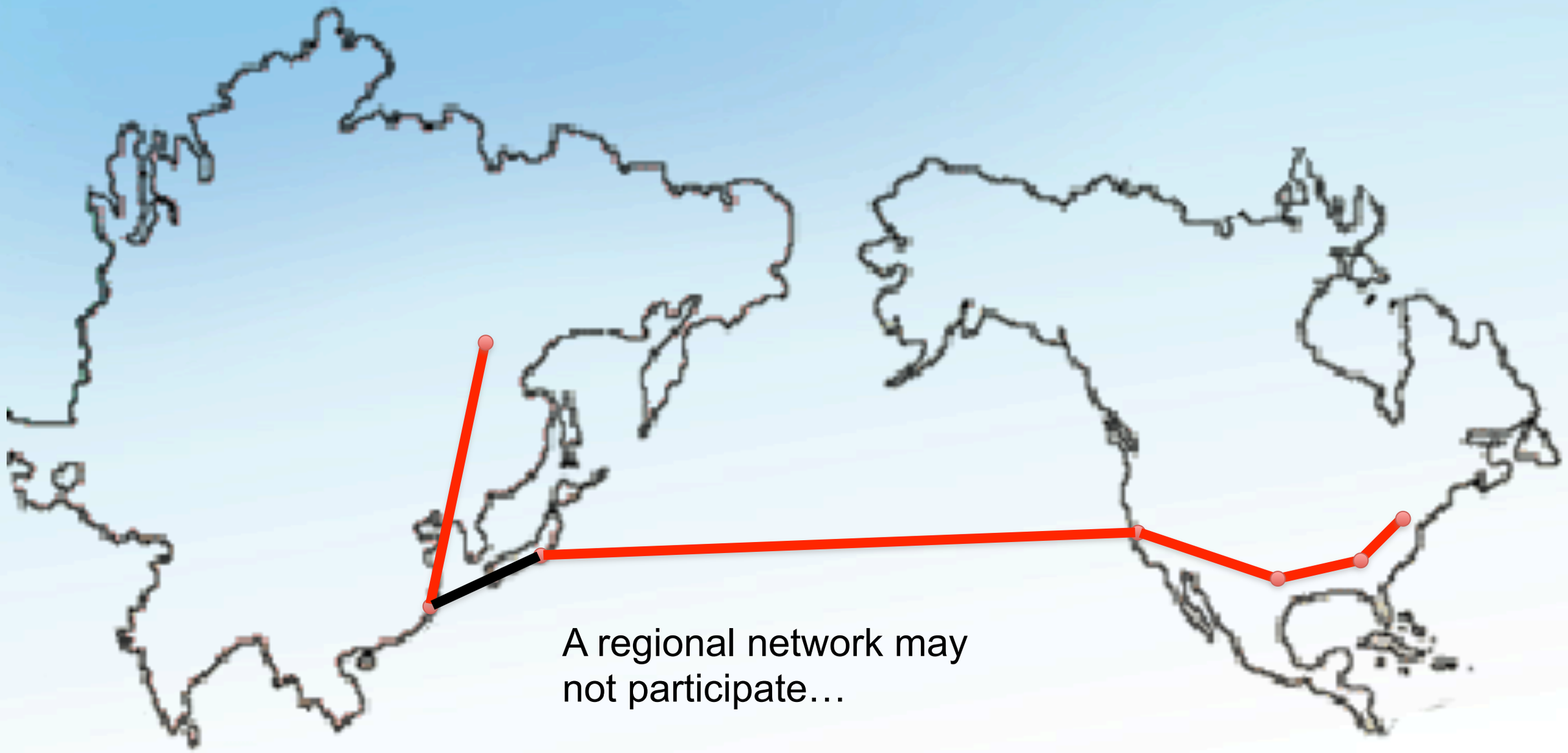


Typical: Poor Performance … Somewhere



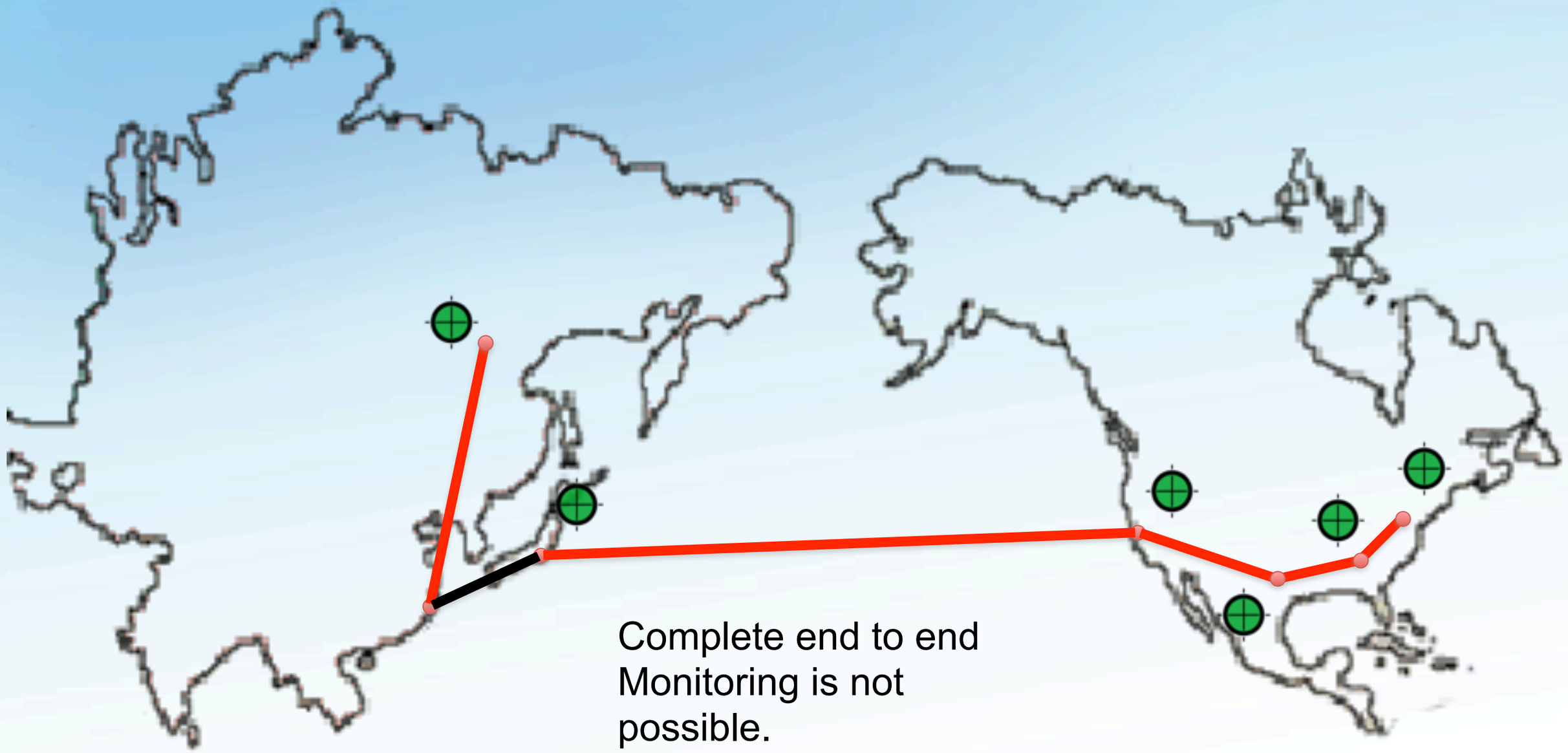
The exchange point
makes statistics available

Typical: Poor Performance … Somewhere



A regional network may not participate…

Typical: Poor Performance … Somewhere



Complete end to end Monitoring is not possible.

Lessons Learned

- Missing part of the path leaves us with a huge disadvantage
- May discover some problems through isolation on the path we know, could miss something
  - Most network problems occur on the demarcation between networks
  - Testing *around* the problem won't work (we still have to transit this network)

Use Cases

- The following use cases demonstrate use of perfSONAR tools to solve sometimes complex performance problems
  - Cisco Telepresence
    - Multi-domain path where performance guarantees dictate use of a specific application
  - Additional Use Cases (Not Presented)
    - Georgetown International Campus
      - Assuring quality, from one end of the world to another
    - USATLAS
      - Enabling *Big Science* through diagnostic checks and regular monitoring
    - REDDnet
      - Assuring clean paths for datamovement

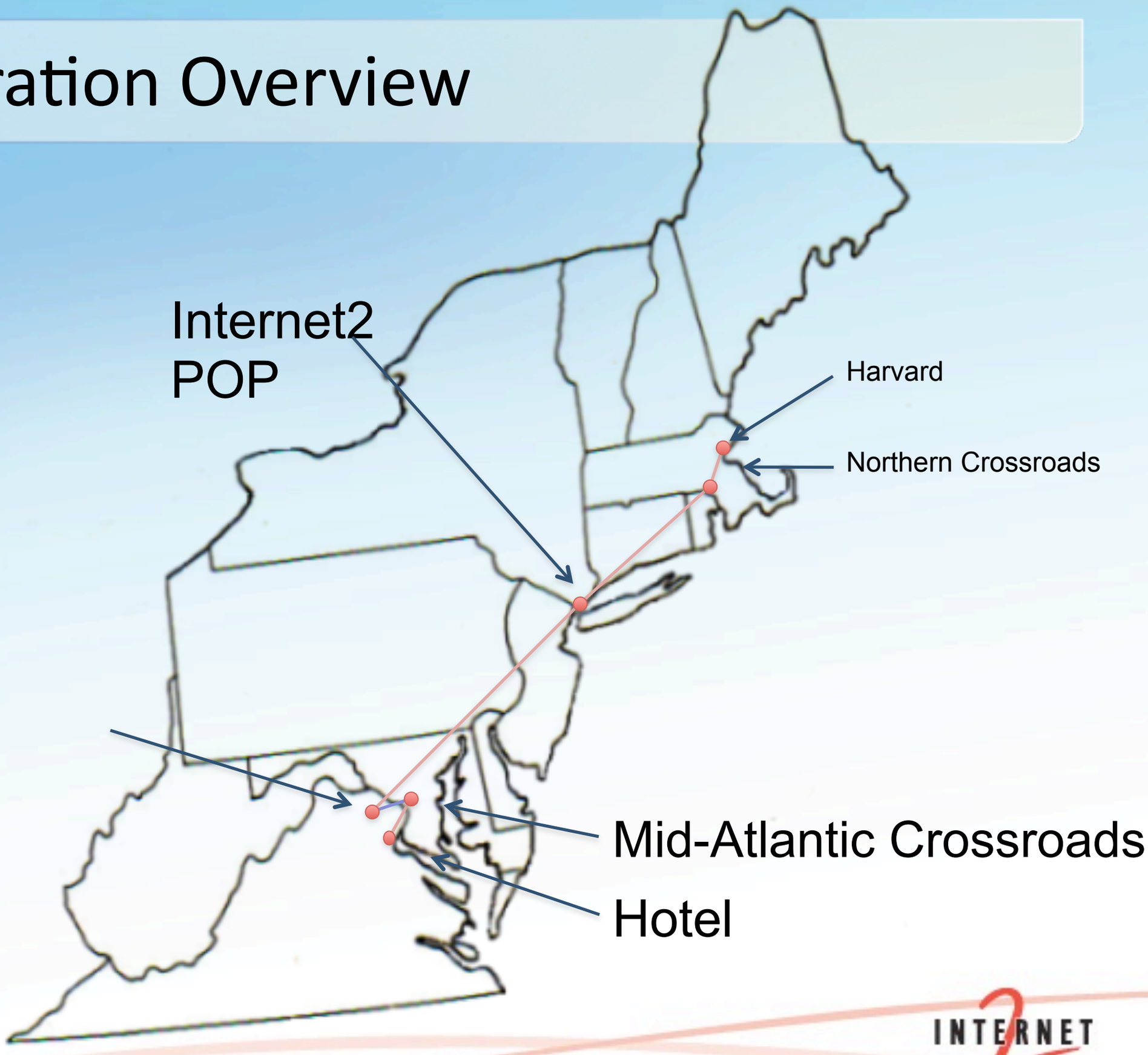
Cisco TelePresence Demo

- 2 Locations
  - Harvard University (Boston, MA)
  - Spring Member Meeting (Arlington, VA)
- Must meet or exceed performance expectations
  - < 10 ms Jitter (Packet Arrival Variation)
  - < 160 ms End-to-End Delay
  - < 0.05% Packet Loss
- Network Path spanned:
  - ~450 Miles
  - 4 Distinct Domains
    - Internet2
    - Mid Atlantic Crossroads (MAX)
    - Northern Crossroads (NOX)

perfSONAR powered

INTERNET 2

Demonstration Overview



perfSONAR powered

INTERNET 2