



Network Based On-demand/Grid System (NBGS)

**NBGS team
Cisco Systems**

February 2006

NBGS Phase I



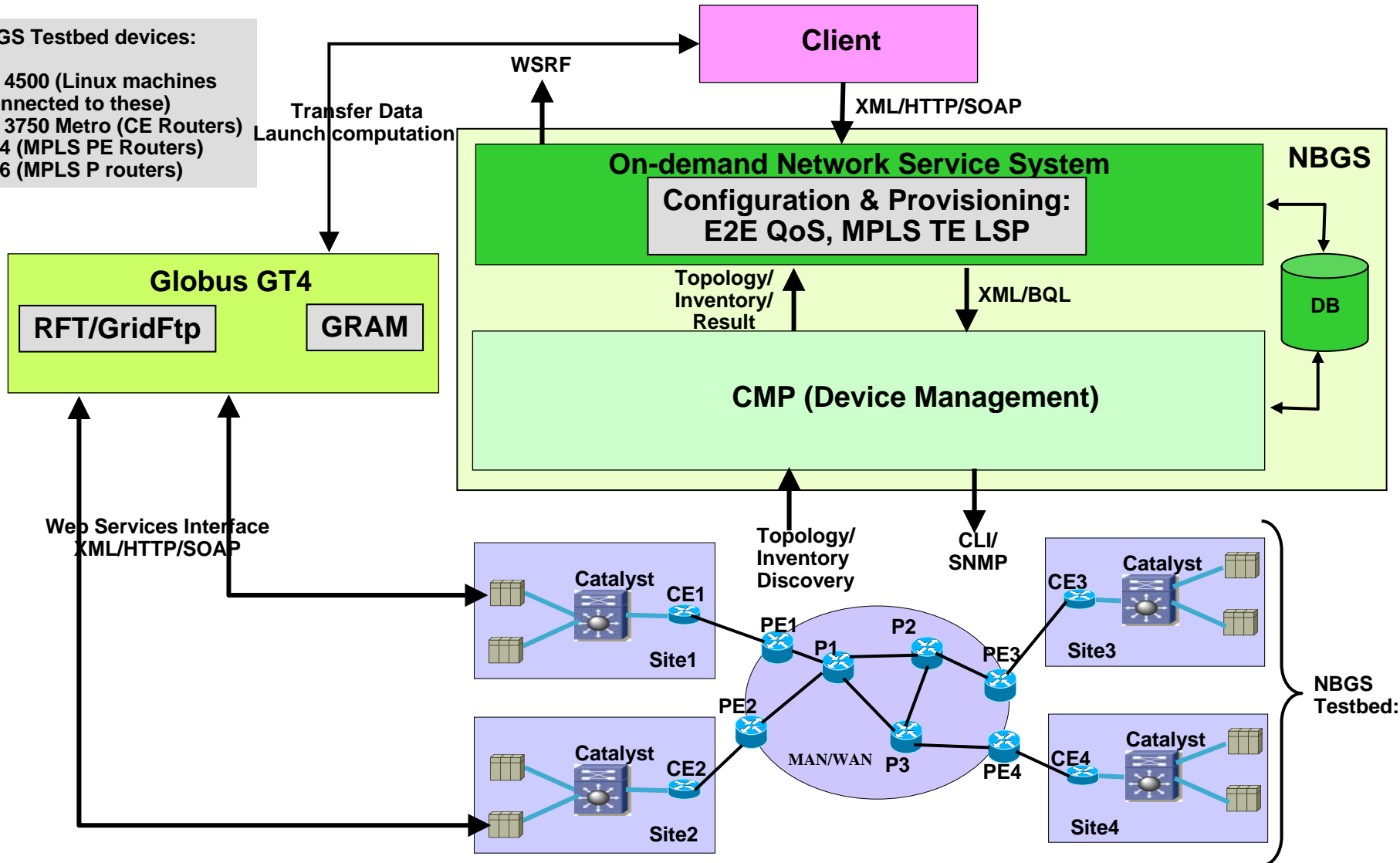
NBGS Features

- The main feature of NBGS is *on-demand, dynamic, automated* and *quick* provisioning of network resources that may be requested by Grid clients (Grid middleware, tools) before these clients can schedule application jobs to run on available Grid resources (compute resources)
- The network resources are requested via abstract and simple interfaces that hide the details of network resources and complex steps of underlying network provisioning
- For example, QoS is abstracted as only bandwidth and priority (via an abstract interface in NBGS called the *CosLink*)
- But the underlying QoS may be provided end-to-end as a combination of
 - Ethernet QoS (802.1p)
 - IP (based on TOS marking)
 - DiffServ (based on DSCP marking)
 - MPLS QoS (combination of MPLS EXP marking and MPLS TE LSP)

NBGS Architecture – Phase I

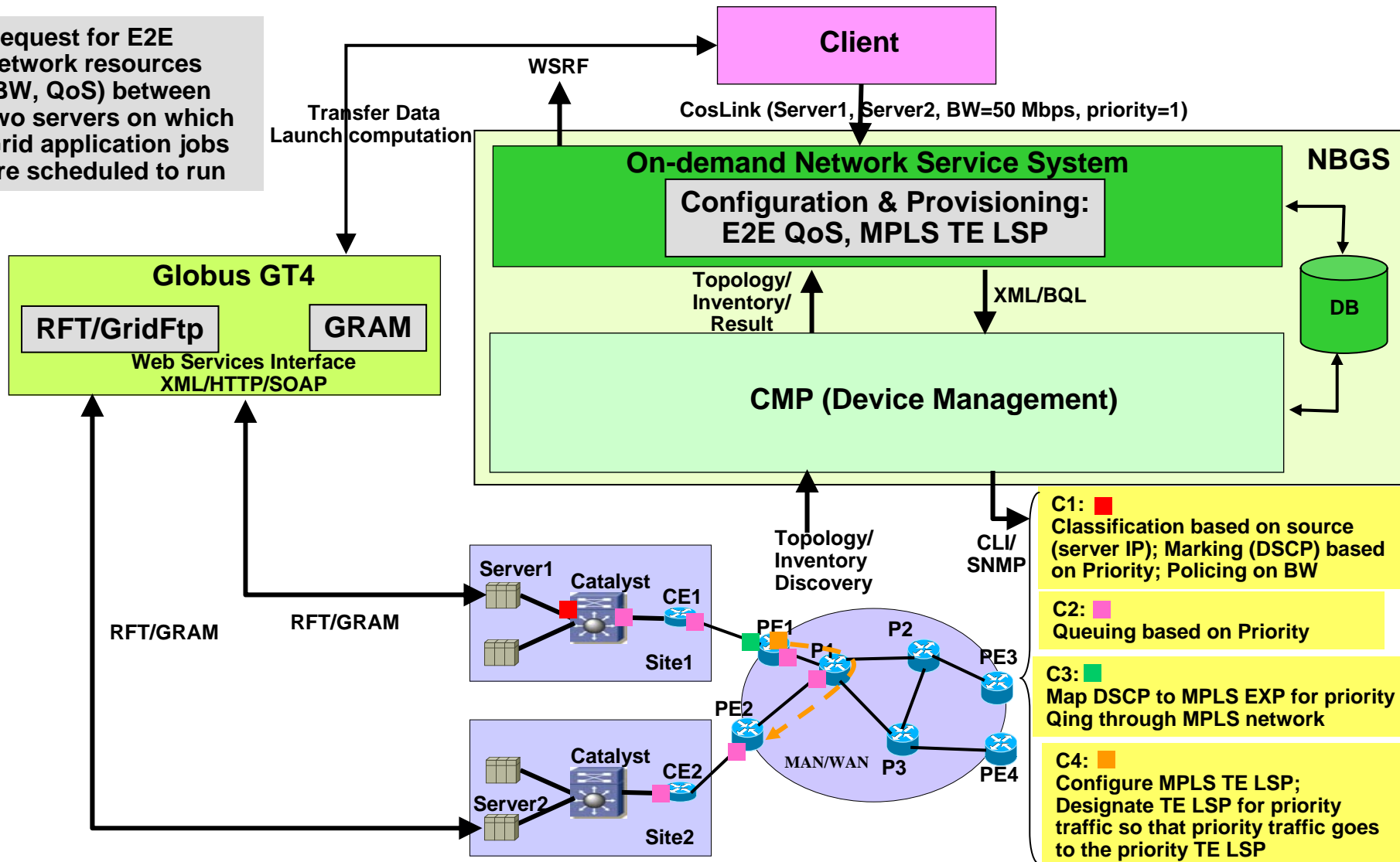
NBGS Testbed devices:

- Cat 4500 (Linux machines connected to these)
- Cat 3750 Metro (CE Routers)
- 7204 (MPLS PE Routers)
- 7206 (MPLS P routers)



NBGS Provisioning Flow

Request for E2E network resources (BW, QoS) between two servers on which Grid application jobs are scheduled to run



POC Demo - purpose

- The main purpose of the NBGS POC prototype project is to:
- Show on-demand, quick, and dynamic provisioning of network resources (QoS, bandwidth, MPLS TE LSP, etc.) as requested by Grid clients (Grid middleware, tools) before these clients can schedule jobs to run on available Grid resources (compute resources)
- Show that the existence of on-demand/dynamic network resource provisioning interfaces (provided by NBGS) may improve overall performance of Grid applications, that is,
 - application output (such as video rendering output) produced in shorter time-frame than is otherwise possible
 - application output is furnished over the network to the source in time and with better quality (for example, quality streaming of rendered video output to a client)

POC Grid Application – Post-production Video Rendering

- A high-quality video (of several GB size) is rendered on four Grid machines after editing actions are selected on the video (rendering is compute intensive → an hour-long video may take several hours to render)
- Brief workflow (the workflow is embedded in a script) of the POC demo:
 - Video editing actions (video/audio effects) are chosen using Cinelerra (Linux tool)
 - Video file (frames) is divided into a number of segments using Cinelerra.
 - Rendering computers (four machines in the demo) are selected (Cinelerra rendering engines are already loaded on these machines)
- Large file transfers may add substantial time in a loaded network to the overall processing time (from start to the time when output is produced and merged)
 - Need priority data transfer
→ need proper QoS
- Rendered video may be visualized during or after rendering (for checking edited result)
 - Stream video
 - Video on-demand
→ Need proper QoS

POC Grid Application – Post-production Video Rendering contd...

- For each selected (remote) Grid machine (Mk)
 - NBGS QoS/BW resource provisioning interface (CosLink) is invoked with the machine (E1), where the main script resides, as source, and Mk as destination
 - Data transferred to Mk (using Globus WS-GRAM/RFT)
 - When data transfer completes, network resources are claimed by invoking an NBGS interface (rendering computation may take long time, hence network resources can be returned to pool)
 - Rendering computation is invoked (via Globus WS-GRAM)
 - Once rendering completes (Globus WS-GRAM on Mk notifies E1), NBGS CosLink invoked again with Mk as source and E1 as destination
 - Rendered Data transferred to E1
 - All the rendered video segments are merged on E1
 - Video streaming client (C1) is selected
 - NBGS CosLink is invoked between E1 and C1
 - Rendered video is streamed to C1

POC Grid Application – Post-production Video Rendering contd...

- Various options are used in the demo to show difference between NBGS and non-NBGS based rendering:
 - Single machine rendering
 - Rendering on Grid (four machines) and output streaming *without* invocation of NBGS interfaces
 - Rendering on Grid (four machines) and output streaming *with* invocation of NBGS interfaces.

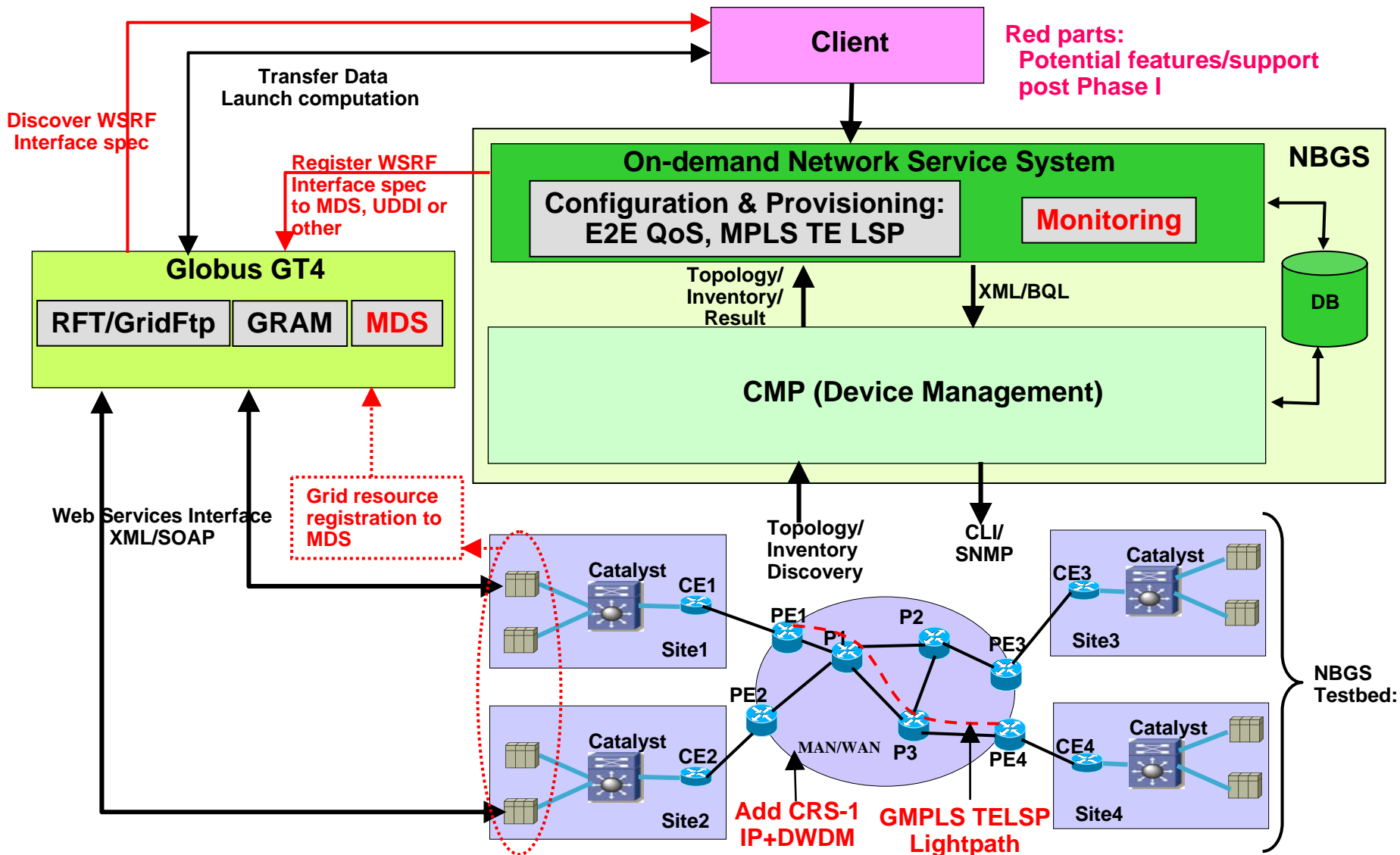
NBGS Team

Engineering	Field, other
<p>NMTG Project leads: Masum Z Hasan Wayne Clark</p> <p>NMTG development Manager: Dragan Milosavljevic</p> <p>NMTG development team: Eileen Liu Miro Krandic Umesh Tyagi</p> <p>NMTG development Director: Geng Lin</p> <p>NMTG VP, CTO Nino Vidovic</p>	<p>Horst Dumcke</p> <p>Dirk Schroetter</p> <p>Monique J Morrow</p> <p>Peter Tomsu</p>

NBGS Post Phase I



NBGS Architecture – Post Phase I



Towards all-encompassing Grid – our goal

Systems/ Infrastructure	Parallel Machines or Supercomputers (MPPs, Vector, NUMA)	Dedicated HPC Clusters	Scavenge low-end systems	Grid
Goal	Absolute HPC Performance	HPC with better Price and Performance ratio	Any application	Cost-effective resource sharing and on-demand resource access/provisioning across LAN, MAN, WAN
Targets	Large scale SMPs; special purpose	Clusters	PCs, Servers, Workstations	All on the left + Network factored in (L1/2/3/4 Network Resources factored in)
Coupling	Tight	Tight, Loose	Loose	Loose
Communication	Shared memory	Message passing	Any	Any
Network	Bus, backplane	Infiniband, GigE	All (LAN, MAN, WAN) and Internet scale	All (LAN, MAN, WAN) and Internet scale

Other potential post Phase I support



NBGS and Data Center



Typical Network with DC

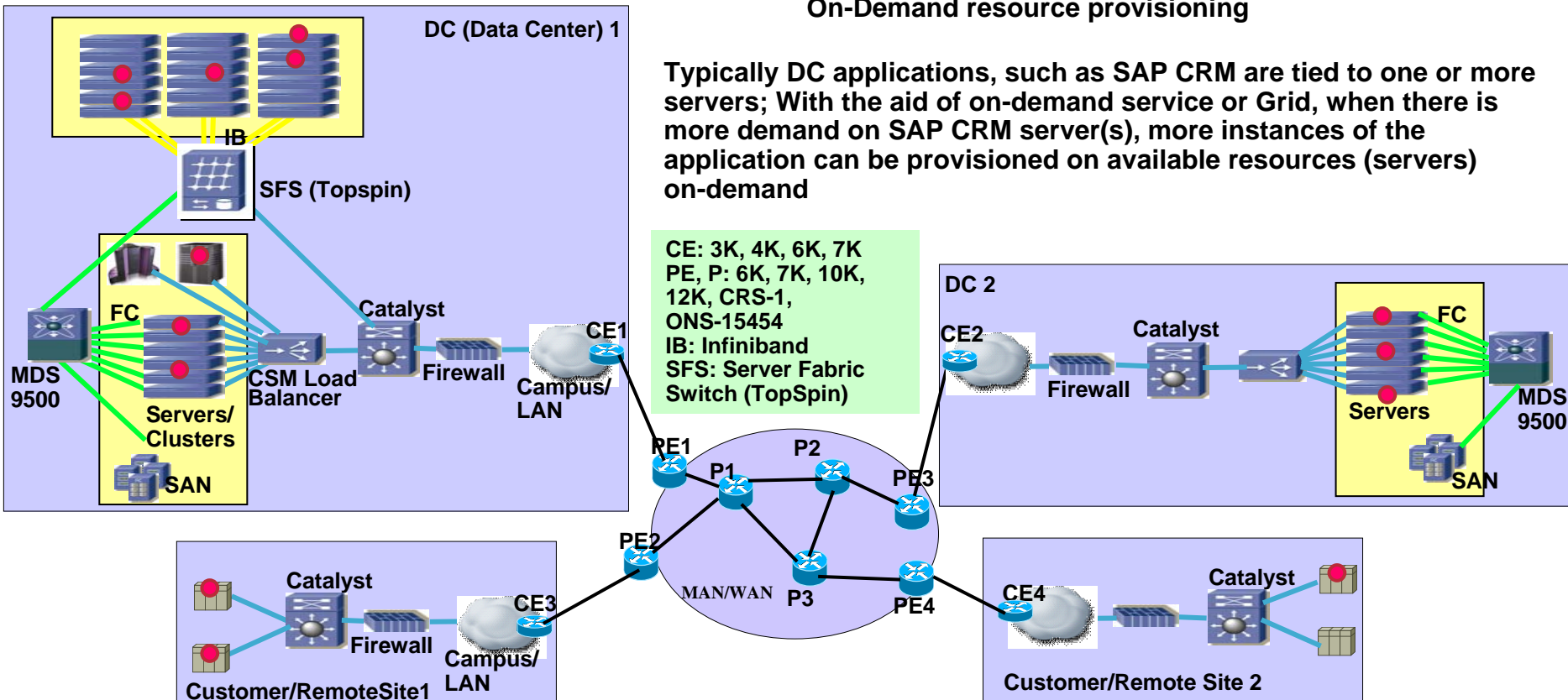
- Network with distributed DC and remote/customer sites
- Yellow region: typical Grid domain with L7 Grid resources
 - Compute: PC, Server, Cluster, Supercomputer
 - Storage

NBGS may provision only a segment of the network: for example, only within the DC and remote sites and not MAN/WAN

Available resources, shown as ● are advertised (via a directory service, such as Globus MDS); not all L7 resources may be Grid resources

A Grid client (such as a scheduler) checks resource availability information, selects useful resources, then loads application/task images on selected resources → On-Demand resource provisioning

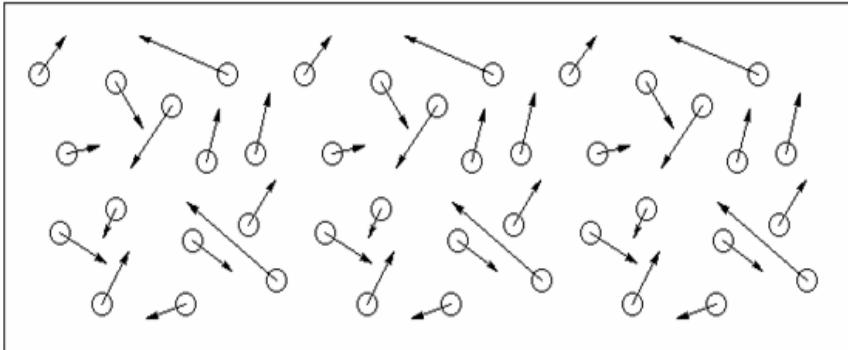
Typically DC applications, such as SAP CRM are tied to one or more servers; With the aid of on-demand service or Grid, when there is more demand on SAP CRM server(s), more instances of the application can be provisioned on available resources (servers) on-demand



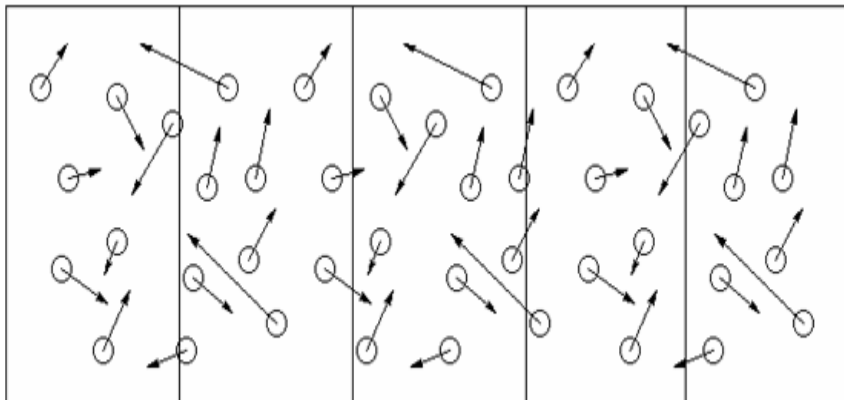
NBGS and HPC and Cluster Grid



HPC – Molecular Dynamics



Spatial partitioning of problem space



A B C D E

- Remote Visualization during Simulation

- Molecular Dynamics

- Application in Bio-technology: Protein Folding, Drug Design

- Extensive communication between processors exchanging large quantities of data at every time step even with optimization;

Reason:

- Atoms may interact over distances of several nanometers
- Atoms may freely wander from one region to another, requiring processors to renegotiate ownership of such atoms

- May require millions of time-steps

Remote Visualization during Simulation

200,000 Atoms per partition
 Precision: 64 bit
 Coordinates: 3 dim (x, y, z)
 Variables: 10
 Memory: $64 \times 3 \times 10^2 \times 10^5 = 384 \times 10^6 \sim 400\text{Mbit}$
 Time Step: 1 sec: 400Mbit generated at each time step
 Assuming 50% atoms interacts with other (neighbor) partition \rightarrow BW between nodes 200Mbps

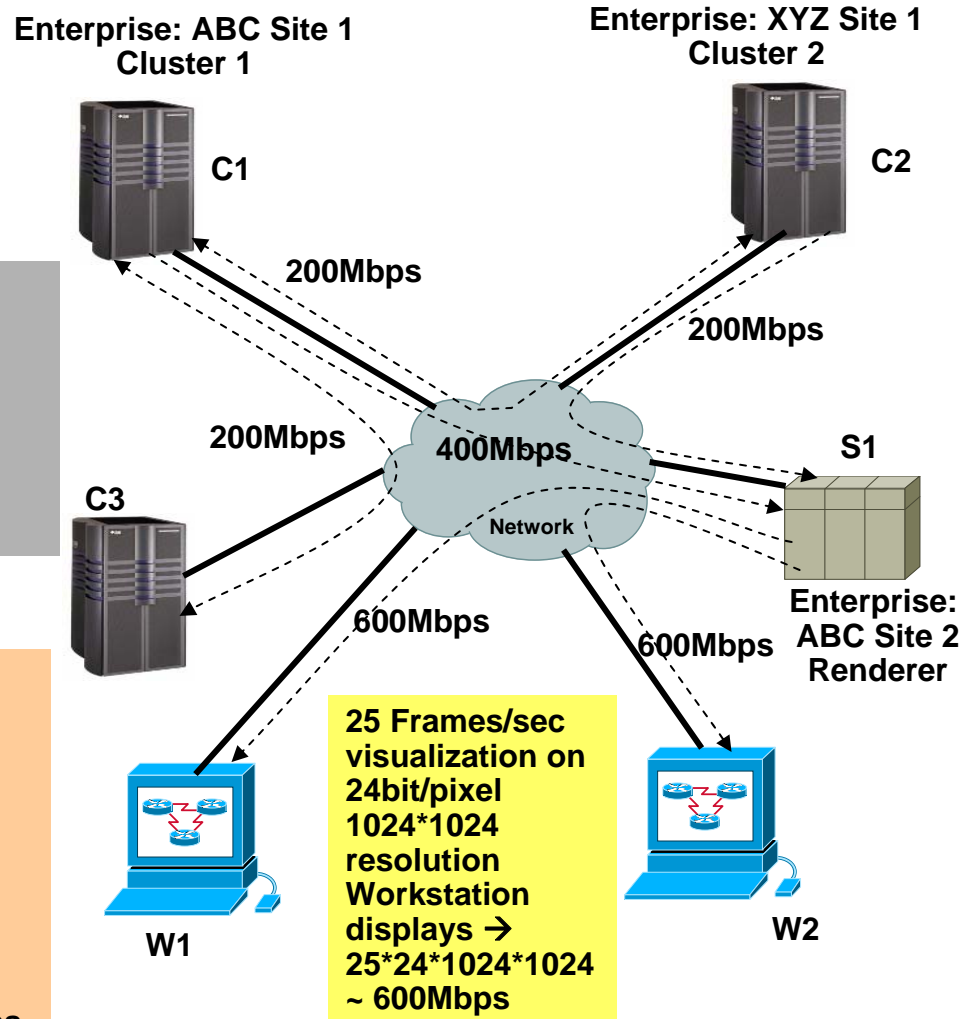
L7 resource requirements:

of CPUs: 16
 CPU speed: 800 Mhz
 Memory: 200MByte each
 Local Disk: 1GB each

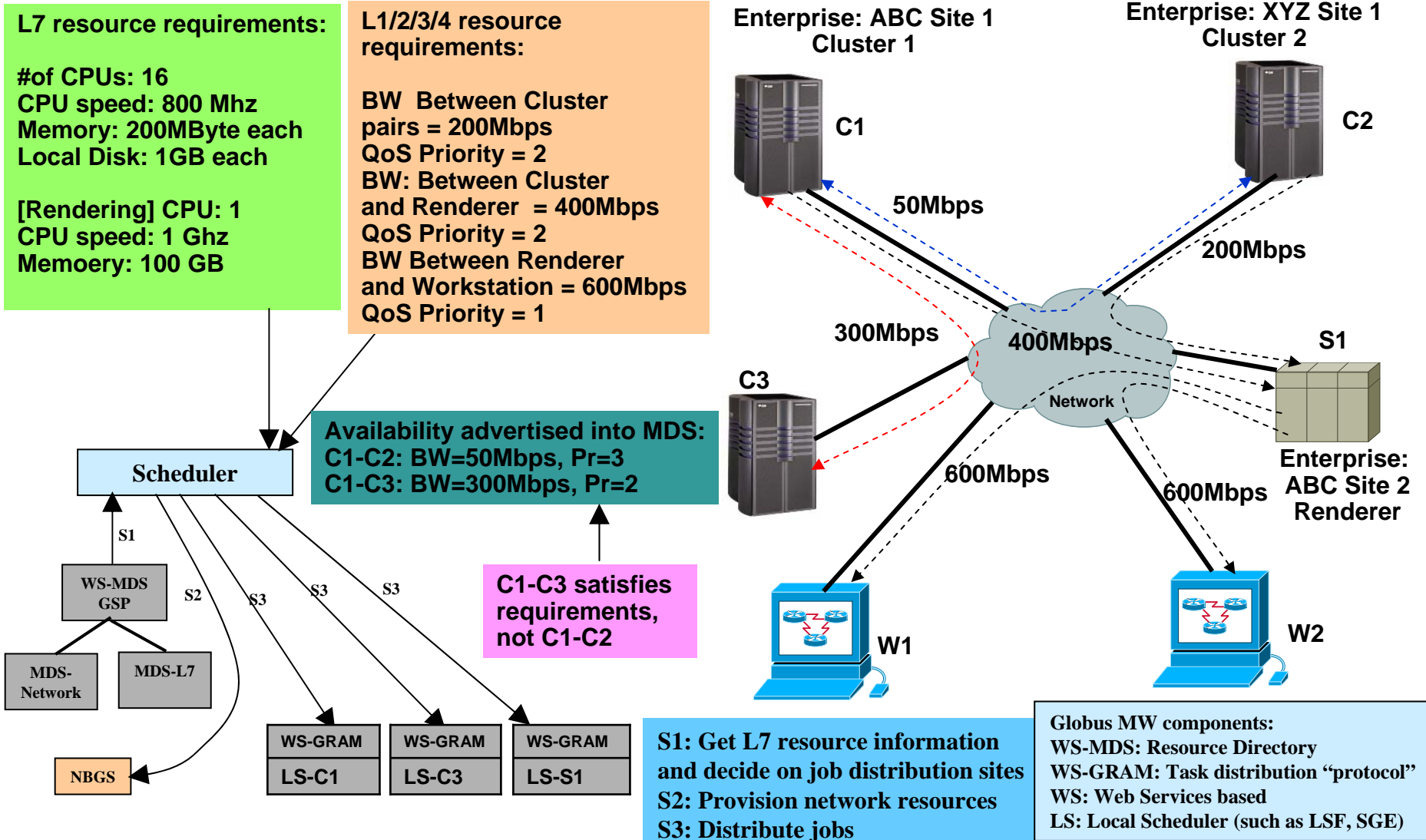
[Rendering] CPU: 1
 CPU speed: 1 Ghz
 Memory: 100 GB

L1/2/3/4 resource requirements:

BW Between Cluster pairs = 200Mbps
 QoS Priority = 2
 BW: Between Cluster and Renderer = 400Mbps
 QoS Priority = 2
 BW Between Renderer and Workstation = 600Mbps
 QoS Priority = 1



Use Case - Network-aware Task Distribution – Distribute tasks based on advertised configured resources - Provision network Resources on-demand



CISCO SYSTEMS

