

LCLS Production Network & Servers

Terri Lahey

4 May 2006

LCLS Production Networks/Servers

Terri Lahey

lahey@slac.stanford.edu

Goal

- Build production hosts, workstations and networks
- Finalize LCLS host requirements so we can start to implement
- Overview of current production systems: what's interesting, not interesting, what's different
- Increase dialog!

Outline

- Engineering Teams
- Use SCCS services where possible
- Production Infrastructure & Shared Servers
- Software Development & Release to Production
- Monitoring Production Systems
- Host System Configuration Management
- Proposed Infrastructure
- SW Tools Needed
- Procurement/Testing Status
- Ethernet Architecture
- What's Next?

Engineering Teams

- SCCS: (network and security)
Gary Buhrmaster, Antonio Ceseracciu,
Charles Granieri, Fred Hooker
- LCLS: (network)
Doug Murray
- CPE: (network and servers)
Ken Brobeck, Jim Knopf,
Terri Lahey, Jingchen Zhou

Use SCCS services where possible

- Security:
 - identify model and DOE/Office of Science requirements
 - Scan networks in a scheduled manner (production very controlled)
- Network Design and Physical Layer
 - SLAC standards to achieve more reliable networks
 - Central Management with strong liaison to Controls
 - Current Equipment/Design Knowledge
- SCCS manages Oracle, web servers, & OS versions. Servers reside at MCC
- Use AFS for CVS repository, development, & main web server (mirror to MCC)

Production infrastructure

- Can run LCLS 24x7 standalone so all required services at MCC
- All required files reside on MCC NFS server & all hosts mount NFS volumes
- Network disconnect point so we can run isolated, if needed
- MCC NFS server is used for staging data, not large storage area
- Move data to SCCS NFS toaster space and use SCCS tape backup facilities
- Recommend Network Attached Storage for data from archiver, etc.
- MCC saves all “history” data forever. Solve for LCLS, too.

Individual Accelerator Realm share some servers

- MCC: “SCP” for HLA, DNS
 - NFS server: NFS and TFTP
 - Oracle *
 - Elog *
 - Artemis (*)
 - Web server for documentation (also servers specific to LCLS) *
- * Managed centrally by SCCS for improved security

Software Development

- Develop on SCCS servers
- Keep CVS repository on AFS
- readonly /ref
- Build and install on AFS
- Release code & configurations to MCC NFS with release escalation
- Method to revert to previous version

Overview of Production Release

- cvs commit
- build & install on AFS using scripts started by Kristi & Steph
- levels of release (can reduce) tst,dev,new,prod
- path that supports access to levels of release
- automated distribute (move) to MCC NFS for production, uses rdist, supports host and IOC software
- supports EPICS files types, Oracle, Matlab, startup
- can build with non-SLAC makefile (eg. cmlog) and escalate and distribute

Monitor Systems and Process Restart

- ALH to monitor system resources, network ping test, and EPICS applications & restart processes
- process management system to show status & restart processes without knowing exact host (use same startup files as ALH restart)
- iocConsole.pl to connect to IOCs and keep log of console activity

Hosts: System Administrators

- Use SCCS-supported versions of operating systems (RHEL3 ... RHEL4) & applications where possible
- Patch operating systems and update (linux and solaris – standalone and taylor)
- Automate maintenance of production hosts – creating new system for LINUX (up2date/yum/?)
- Reduce maintenance load and improve security by using taylor where possible
- Centralized Log server & security monitoring
- Use existing servers where possible (e.g. elog)

Proposed Infrastructure

- AFS dependencies can lead to lost beamtime
- Implement Production servers that provide all identified systems, including code released to NFS (hosts mount NFS)
- Support all required tools, including serving current source and objects for gnu debugger
- Mitigation Plan for supporting access to AFS if crisis arises.

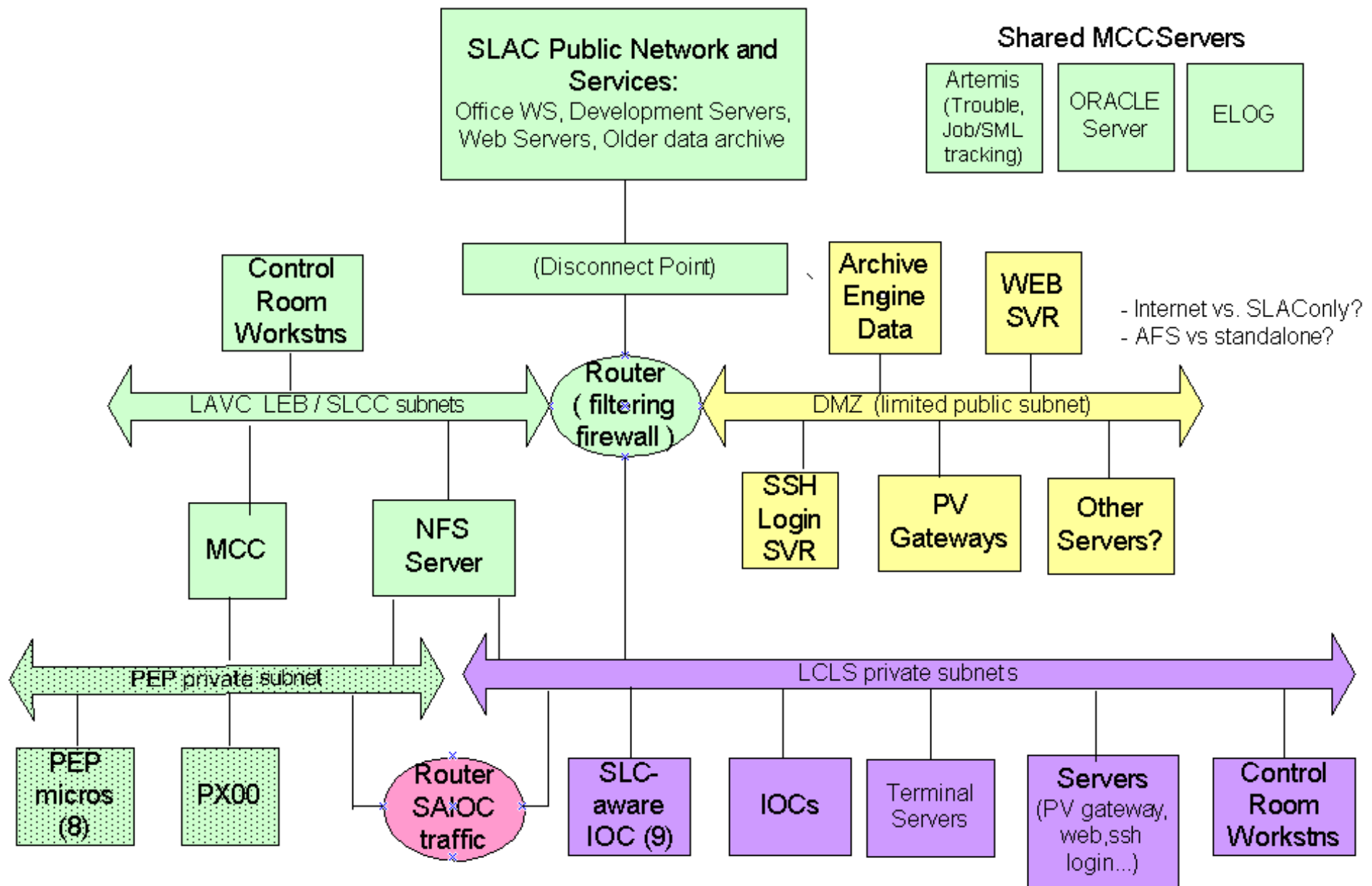
Software Development Tools Needed

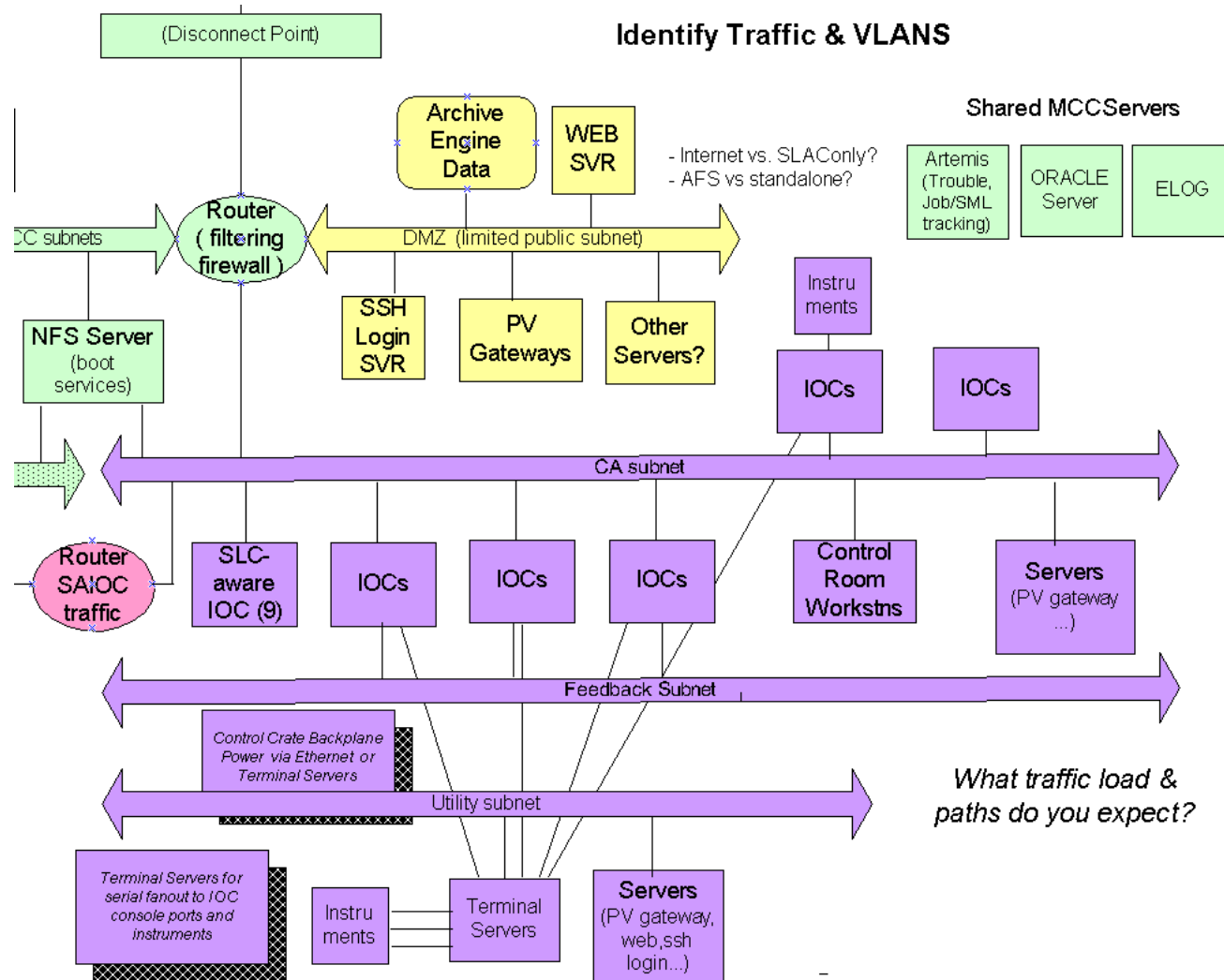
- Software development tools needed during commissioning:
 - gdb debugger over tcpip: client side gui needs access to source and objects. Server side on the IOC needs access to objects. These must be in-synch.
 - gdb debugger over serial line (console) is needed to debug some pieces, eg. tcpip stack. It changes the speed and passes binary over the serial line. Will iocConsole support this?
 - release code changes quickly, and keep the main source code repository up-to-date with changes. (For example, avoid multiple versions of source code. Avoid moving changes back to the repository after checkout on the accelerator.) Source and Objects for gdb are readonly.
 - Some services might not be identified until commissioning (eg. need arises during commissioning).

Procurement/Installation/Testing

- Networking/Computing Infrastructure needed in S20, MCC, S29
- Network switches arrived
- fiber ordered and installation details in progress
- testing DIGI terminal server
- ordered server for implementing production Linux server
- Desktops: testing quad card Linux desktop. Complementary product is new Sun Ray product 2fs
- PX00 performance testing is critical during PEP run

LCLS Accelerator Network





What's next

- Identify and implement VLANs needed.
 - What traffic loads/paths do your systems have?
 - What broadcast and multicast traffic do you plan?
- Test 11g in S20 for wireless coverage
- Additional tests of SLCaware IOC and improve monitoring of traffic to avoid interference between PEP & LCLS programs
- Filtering Router or Firewall?
- Complete design and design review of production hosts and networks & documents
- Have you planned for multiple versions of EPICS
- WBS for hosts & network
- Integration of plans with other networks (timing, MPS, feedback, etc.)
- What additional requirements do you have?