Edg Testbed Experience

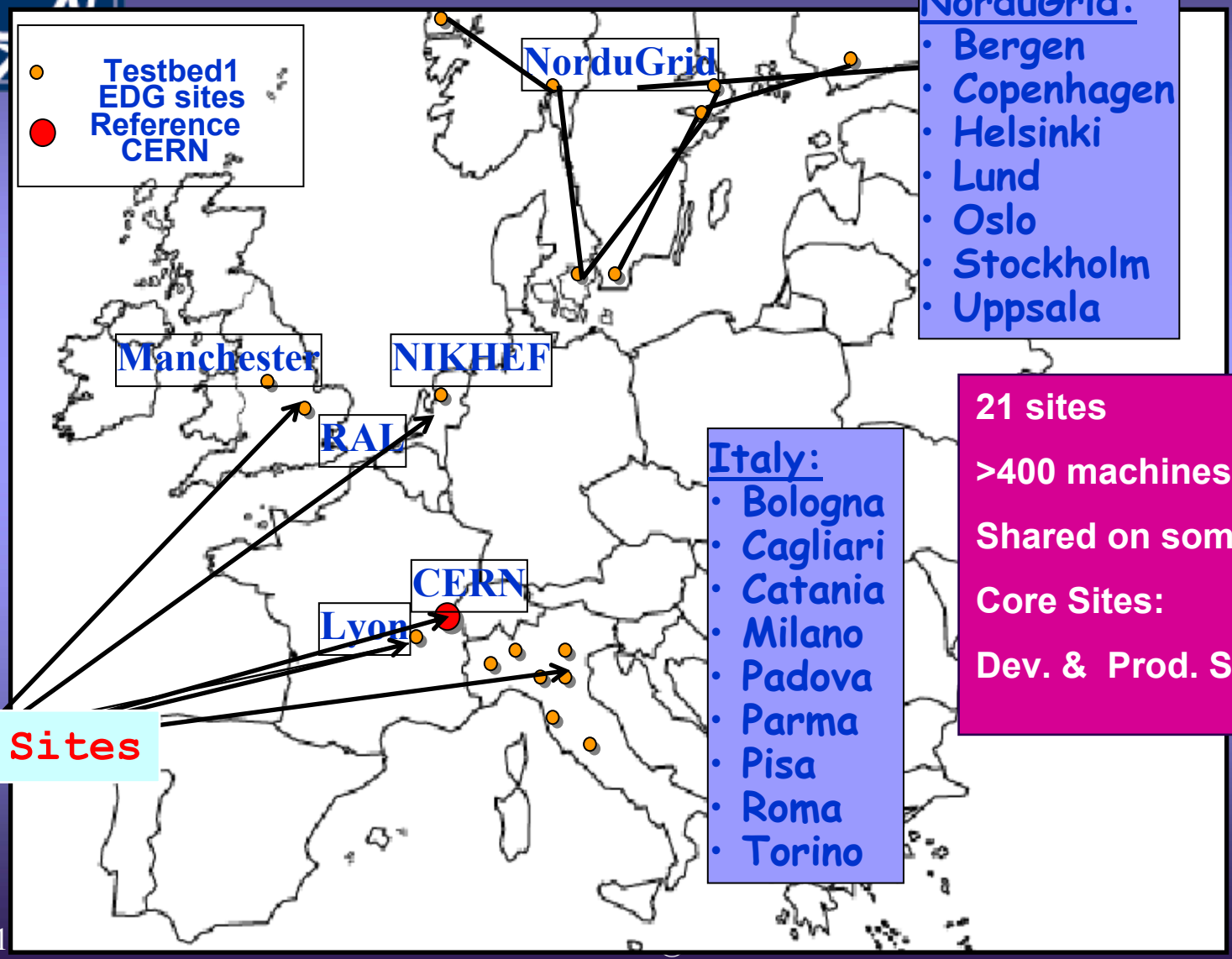
- EDG a short introduction
- Services provided by the EDG Middleware
- EDG Testbeds/CERN Testbeds
- Experience
  - LCFG (Install and Configuration)
  - Middleware
  - Grid Specific
  - Operation
  - Resources
- Summary

EDG http://www.edg.org

- European Data Grid (3 year project)
  - Project for middleware and fabric management
  - Emphasis on data intensive scientific computing
  - Large scale testbeds to demonstrate production quality
  - Based on globus
- Applications
  - HEP, Biology/Medical Science, Earth Observation
  - Organized into VOs (Virtual Organizations)
- Main Partners

IBM-UK, CS-SI (Fr), Datamat (It) + 12 Research and University Institutes

EDG http://www.edg.org



NorduGrid:
- Bergen
- Copenhagen
- Helsinki
- Lund
- Oslo
- Stockholm
- Uppsala

Italy:
- Bologna
- Cagliari
- Catania
- Milano
- Padova
- Parma
- Pisa
- Roma
- Torino

21 sites

>400 machines

Shared on some sites

Core Sites:

Dev. & Prod. Sites

Services

- Authentication
  - GSI Grid Security Infrastructure based on PKI (openSSL)
  - Globus Gatekeeper, Proxy renewal service…
- GIS
  - Grid Information Service, MDS (Metacomputing Dir. Service) based on LDAP, VOs
- Storage Management
  - Replica Catalog (LDAP), GDMP, GSI enabled ftp,RFIO …
- Resource Management
  - Resource Broker, Jobmanger,
  - Jobsubmission
  - Batch System (PBS,LSF)
  - Logging and Bookkeeping …

Services II

- Services are interdependent
- Composite Services
  - Require their own Database (MySQL, Postgres, …)
  - CondorG
- Services are mapped to logical machines
  - UI User Interface, CE Computing Element (Gateway)
  - RB Resource Broker, SE Storage Element,
  - WN Worker Node (Batch Node), RC Replica Catalog, IS (MDS)
  - VO server, Proxy Server (Proxy renewal for long jobs)
  - LCFG server for installation and configuration
- Services impose constraints on the setup
  - Shared File system required between some services

Services III (What Runs Where)

Deamon (only grid software)	UI	IS	CE	WN	SE	RC	RB
Globus Gatekeeper	-	-	✓	-	-	-	-
Replica Catalog	-	-	-	-	-	✓	-
GSI-enabled FTPd	-	-	✓	-	✓	-	✓
Globus MDS	-	✓	✓	-	✓	-	-
Info-MDS	-	✓	✓	-	✓	-	-
Resource Broker	-	-	-	-	-	-	✓
Job Submission	-	-	-	-	-	-	✓
Information Index	-	-	-	-	-	-	✓
Logging & Bookkeeping	-	-	-	-	-	-	✓
Local Logger	-	-	✓	-	✓	-	✓
CRL Update	-	-	✓	-	✓	-	✓
Grid mapfile Update	-	-	✓	-	✓	-	✓
RFIO	-	-	-	-	✓	-	-
GDMP	-	-	-	-	✓	-	-

A Grid Testbed

Sharing part of the file system

UI
User Interface

CE
Computing Element

WN
Worker Node

SE
Storage Element

Minimal Testbed

Minimum for using the GRID

Proxy
Proxy renewal

RB
Resource Broker

RC
Replica Catalog

MDS
Metacomputing Directory Service

LCFG
Installation Server

Cern Grid Testbeds

http://marianne.in2p3.fr/datagrid/giis/cern-status.html

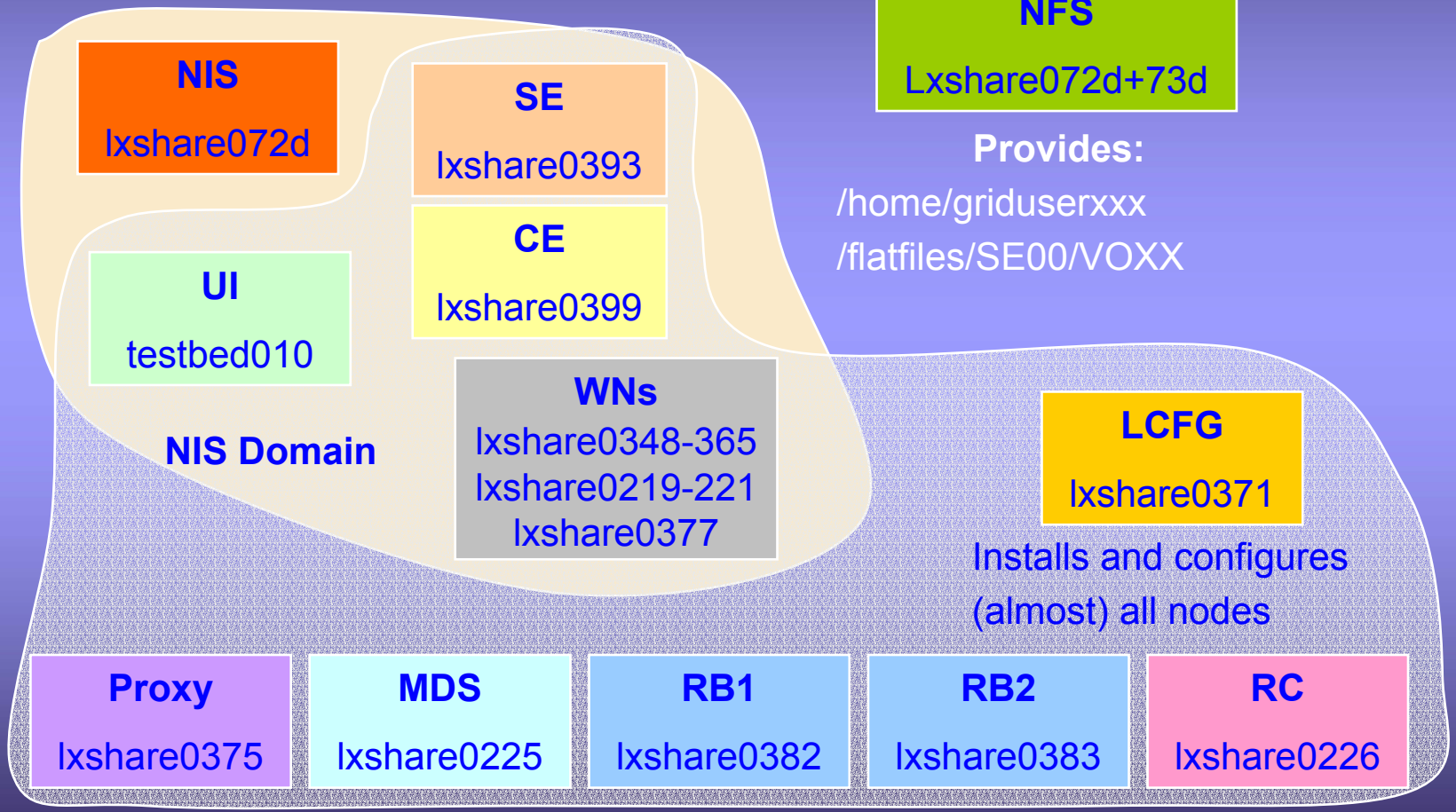
- **>90 Nodes**
- **3 Major Testeds**
  - **Production (Application Testbed): 27 Nodes**
    - Stable release (v.1.2.2) few updates, security fixes
    - Frequent restarts of services (daily)
    - Test production of applications
    - Demonstrations (every few weeks)
  - **Development: 7Nodes**
    - Changing releases, test versions, multiple changes/day
    - Very unstable, service restarts, problem tracing
  - **Major Release: 7Nodes**
    - Development, porting (globus1x -> 2.x, RH6.2 -> 7.2
- **Many Minor Testbeds**
  - **Used by developers for unit testing and first integration**

Cern Grid Testbeds II

http://marianne.in2p3.fr/datagrid/giis/cern-status.html

- **Infrastructure**
  - **Linux RH 6.2 (now almost standard)**
  - **2 NFS servers with 1Tbyte mirrored disk**
    - For user directories on Uis
    - For shared /home on CEs and WNs
    - To provide storage for the SEs (visible on WNs)
  - **NIS server to manage users (not only CERN users)**
  - **LCFG servers for installation and configuration**
    - Different versions
- **CA Certification Authority**
  - **To provide CERN users with X509 user certificates**
  - **To provide CERN with host and service certs.**
  - **Hierarchical system (Registration A.) mapped to experiments**

Cern Production Testbed



LCFG(ng)

LocalConFiGuration System

http://www.lcfg.org Univerity of Edinburgh

- **Described by Olof Barring**
- **For CERN we added support for PXE/DHCP to allow network based install**
- **LCFG works quite well if:**
  - **Configuration has been well tested (Install and Update) and many identical nodes are managed (WNs)**
  - **All services are configured by working LCFG-objects**
  - **The number of different machine types is not too large**
    - Only one directory/server to handle the configuration
  - **You know and respect the limitations of the tool**
    - Example: Only 4 partitions are supported

LCFG(ng) I
LocalConFiGuration System

http://www.lcfg.org Univerity of Edinburgh

- **We used LCFG from start of testbed1 on**
  - **Testing a config/install tool and new middleware at the same time was not a very good idea**
- **LCFG in a rapid changing development system:**
  - **Configuration is different almost each time**
  - **A working update doesn't mean a node will install (better in ng)**
  - **Limited information about how an update went (only on the client) (now improved in ng)**
  - **Configuration objects are not always invoked predictable**
  - **Incomplete Objects**
    - For many of the middleware components the conf objects have been in permanent development
    - Some objects wipe out the required manual changes

LCFG(ng) II
LocalConFiGuration System

http://www.lcfg.org Univerity of Edinburgh

- **LCFG in a rapid changing development system:**
  - **Developer machines are very hard to manage**
    - Local installed RPMs are replaced by LCFG (perfect in a production system)
    - How to keep a system in sync with the releases and not remove the developers work?
  - **User management with LCFG**
    - Usable for accounts like root, service accounts
    - Not practical for real users (no pwd change by user)
  - **No good support for installing only selected services on an already running system (farm integration)**

LCFG(ng) III
LocalConFiGuration System

http://www.lcfg.org Univerity of Edinburgh

Solutions:

- Separate releases of tools/middleware +++
- Developer Machines (+/-)
  - Install machines and turn off LCFG
  - On request reinstall the machines
  - Far too many different machines in many different states
- Verifying --
  - Local written small tools, manual checks (lots of them)
- Missing/Defect Objects ---
  - Test LCFG server and clients for developers
  - Since edg is a dynamic project this will stay with us for a some time
- Managing Users: ++
  - Root and System users by LCFG
  - Everything else with NIS

Install and Configuration

Summary:

- Using a tool is mandatory
  - Only way to reproduce configurations
  - Only way to manage a large number of different setups
  - Middleware developers have to be trained to write config obj for their software!!! (And forced to deliver them with the software)
- Using the projects tool
  - Best way to test the tool
  - Some tests have to be done before and SysAdmins have to know the tool before it is used for the testbeds
- Network based install
  - A highly desirable feature
- Room for improvement

Middleware

- **EDG is a R&D project**
  - **Many services are fragile**
  - **Very complex fault patterns (every release creates new)**
  - **The "right" way to do things has for some services still to be discovered**
  - **The site model used during development was not realistic enough**
    - Management of: Storage, Scratch Space
    - Scalability
  - **Middleware packages depend on conflicting versions of software (compiler, Python …)**
  - **Some components are resource hungry**
  - **Process from working binary to deployable RPM not always reliable**

Middleware

Solutions:

- Ad hoc creation of monitoring/ restart tools +/-
- Setting up multiple instances of the service
- Giving feedback about missing functionality +
- Providing a few upgraded machines (memory) +
- Edg is putting an autobuild system in place ++

Grid Specific

- New model of authentication (via certificate)
  - Updated information from remote sites needed for operation (regular)
  - Integration with Kerberos based systems is not trivial
  - Site policies (differ)
  - Many users don't understand the model (limited lifetime of proxies)
- The wide area effect
  - Simple configuration error on a site can bring a grid services down
  - Finding errors without access to remote sites is complicated (impossible to fix sometimes)
- No central grid wide system administration
  - For changes the SysAdmins on the remote sites have to be contacted
- Changes propagate slowly through the grid
  - Config changes, user authentication changes etc.

Grid Specific

Solutions:

- Efforts to integrate local and grid accounts (KCA)
- The middleware has to handle failing remote services in a more robust way (addressed by edg)
- To speedup the effect of changes test releases are tested first local at CERN and then on the 5 core sites
  - As a result the CERN testbed sees the highest rate of changes
- Communication via meetings and mailing lists
  - A real problem for SysAdmins. Extreme rate of mails from edg users, developers and administrators

Operation I

Usages of the testbeds:

- Development & Unit testing
  - Done by GRID experienced users
  - Need quick responds on requests (hours-day)
  - Conflicts over resources (need n+2 nodes for WPx now!)
- Demonstrations and Tutorials
  - High visibility activities done by experienced users
  - Problems have to be solved at all costs (work intensive)
  - Conflicts between ordinary users and demonstrators (communication)
- "Data Challenges" and Extensive Tests
  - Done by medium to high experienced users
  - Requires allocation of resources (storage/CPUs)
  - Mean downtime of services is critical (weeks-month)

Operation II

Usages of the testbeds:

- Integration Testing
  - Done by GRID experienced users (Iteam)
  - Need very quick responds on requests (hour)
  - Conflicts over resources (one integration testbed only)
  - Conflicts over schedule (WP1 first then WP2 or ??)
- Casual Users
  - Done by all level of users
  - Many users not aware of the current status of the testbeds
  - Create bursts of support work (especially new users)
  - Expect same quality of service as delivered by local farms (but this is a wide area distributed R&D project…)
  - Low level user support is currently done by the SysAdmins of the core sites….

Operation III

Solutions:

- Hierarchy of testbeds +++
  - Testbeds for different purposes with scheduled use
  - Helps a lot, but running them costs many resources
- User Education / Tutorials ++
  - Produces local experts that can handle trivial problems
- Integration of existing production systems
  - Ease resource allocation problem (soon to be done)
- Central user support +/-
  - Currently setup by edg

Operation IV

Running a CA:

- CA and all certificates have limited lifetime
  - A lot of renewal work
  - Regular issuing of new Certificate Revocation Lists
- CA has to be offline
  - Copy requests to floppy, down the stairs, process, floppy, send…
- Certificates come in many different flavors
  - User, host, service,….
  - Consumers often can't specify exactly what they need (trial and error)
- Certificates are not user friendly
  - Moving form node to node you have to carry your key/cert with you
  - Additional password needed
  - Keys/Certs/pwds get lost
  - Proxies have to be initialized and have a limited lifetime
  - Complex fault patterns (CA.Crls, GridMap-files, certs…)

Operation V

Solutions:

- Delegation of the registration process to experiments ++++
  - Team leaders run a RA (they check the requestors) and sign the requests with their certificate
- Tools for handling large number of requests ++
  - Semi automatic system in place and used
- We are building up local expertise (slow)
- Base certs on Kerberos credentials
  - We exploring the automatic generation of short term certificates based on kerberos credentials (KCA) Running prototype

Resources

Hardware

- Currently close to 100 nodes for linux 6.2 and one major version of edg
- Soon 6.2 + 7.2 and two versions of the edg software in parallel ?????

RH6.2 EDG 1.2.3	RH7.3 EDG 2.0.x	RH6.2 EDG 2.0.x	RH7.3 EDG 1.2.3
Production	Production		Production
	Integration	Integration	Integration
Debugging	Developer	Developer	Developer

Resources

Human Resources

- 2.5 persons spread over 5 for running the show (not enough)
- Number of different configuration (will increase)
- Number of different services (will increase)
- Lack of stability and monitoring of services
  (will improve over time)
- Tracking down problems in a distributed system
- Fast responds problematic (planned activities are interrupted)
- Number of nodes is secondary (scaling from 3 to 20 WN)
- Manual interventions during setup
  - Error prone
  - Time consuming
  - Training difficult (especially the manual part changed frequently)
- Demonstrations require one person watching the system

Summary

- Grid testbeds are complex
  - Many services
  - Many changes
  - Maturity of services is still low
  - Interdependencies
- Install and configuration tools are essential
- Grid core sites are very resource intensive
- Administrators need a detailed understanding of the services and their fault patterns
- Administrators have to handle rapid change
- For user support dedicated service needed