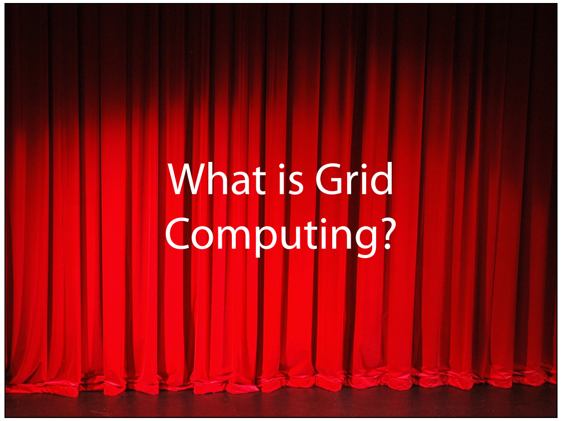
Mathias Dalheimer

Fraunhofer ITWM

Competence Center
High Performance
Computing

More material:
http://md.gonium.net



Virtualization

If you ask me:
Grid Computing is about Virtualization.

Grid problem: coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.

If you ask Foster, Kesselman & Tücke:
The real and specific problem that under lies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.

Several components

coordinated resource sharing and problem solving

Imagine a set of resources: For example, ...

... a cluster, ...

... a network and ...

... a radio telescope.

coordinated resource sharing and problem solving

Now, people may want to use these three resources in a coordinated way:
- Get their share of the resource
- Use all three resources at the same time, e.g. to receive some signal, transfer it to the cluster, and process the data.
- They don’t want to deal with the specific interfaces of the telescope or the cluster.
- They just want to solve their problem, i.e. examine the radiation of a pulsar.

in dynamic, multi-institutional virtual organizations

multi-institutional: ...

... while the cluster may be in Germany, e.g. the Fraunhofer Resource Grid, ...

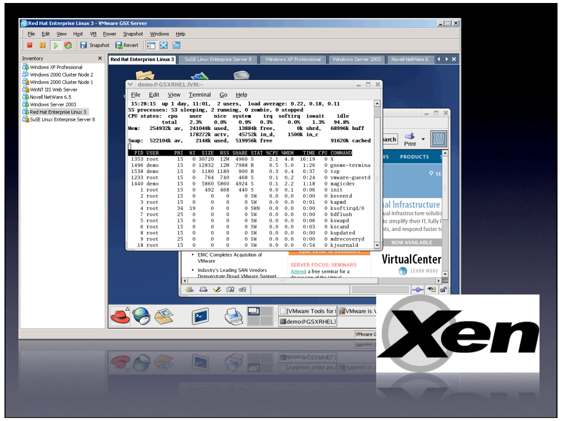
... the telescope may be in the US.

in dynamic, multi-institutional virtual organizations

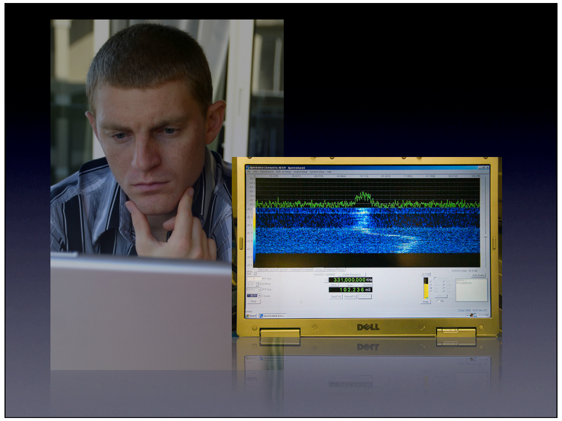
multi-institutional: So we have different organizations collaborating.
- Maybe scientists from Germany and the US are working on the same project, this can then be called a “virtual organization” (VO)
- Usually, a VO is volantile/dynamic: The members change very often.

Virtualization?

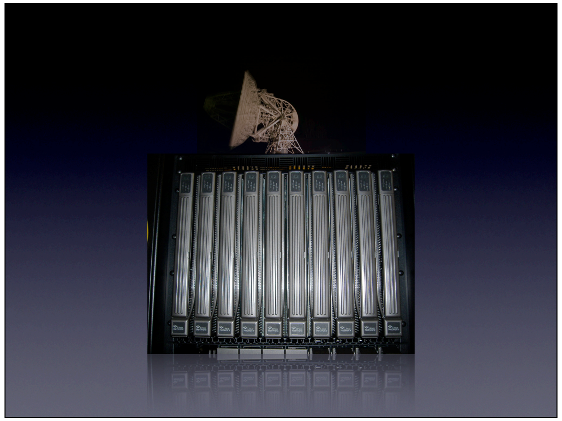
So: What has this to do with virtualization?



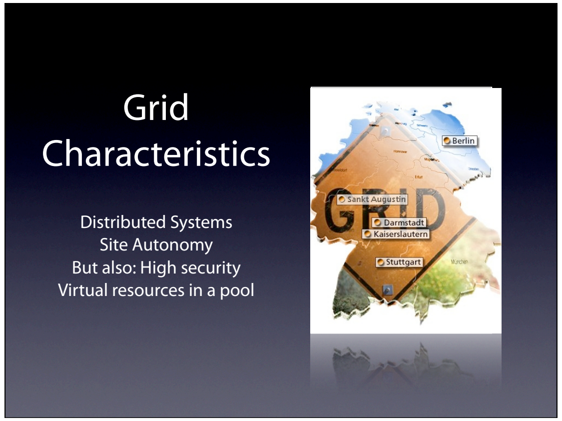
you might know techniques like Xen and VMware
- provide virtual PCs in some way.
- but this is not what I mean. (at least not now)

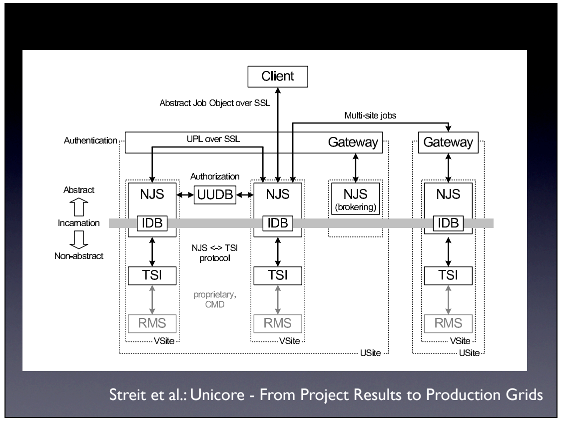


In this context:
- The scientist doesn’t need to know where the cluster is
- Or how to obtain a login
- Or how to run a job on it (different site configurations).
- It just looks like a cluster thingy (Dings), that can be used.
-> Virtual machine that has a radio telescope and analysis processing power built in.

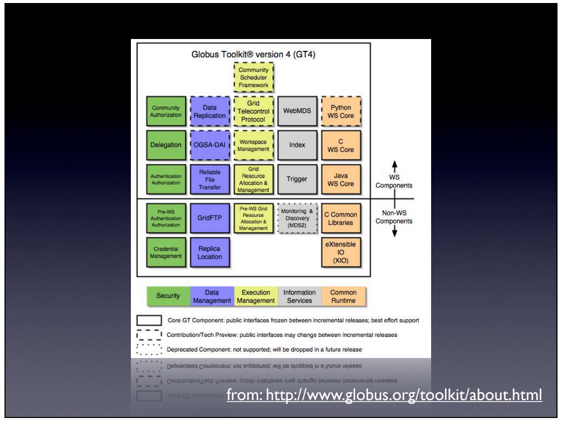


- hmm, ok, I am not the graphics guy - just imagine some duck tape around everything.

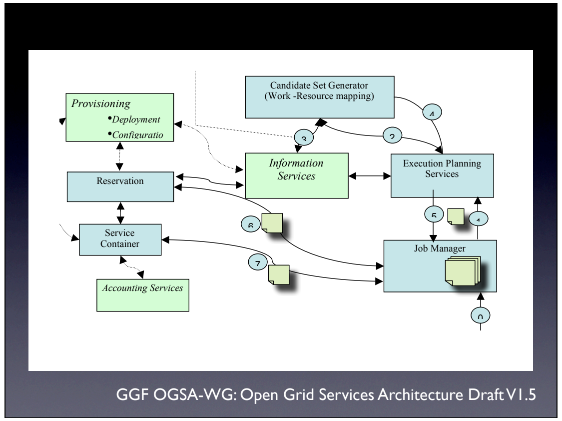




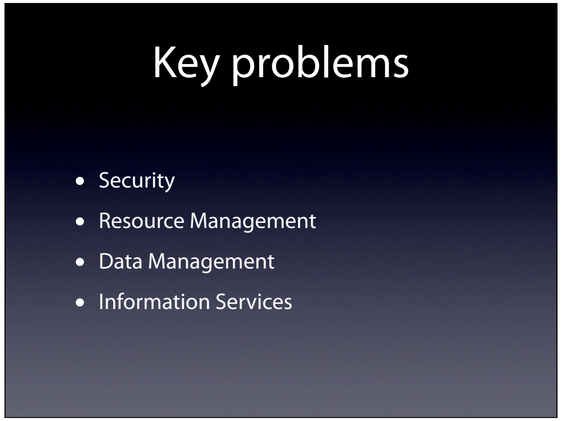
Unicore architecture
- End-to-end model
- Architecture is clearly visible



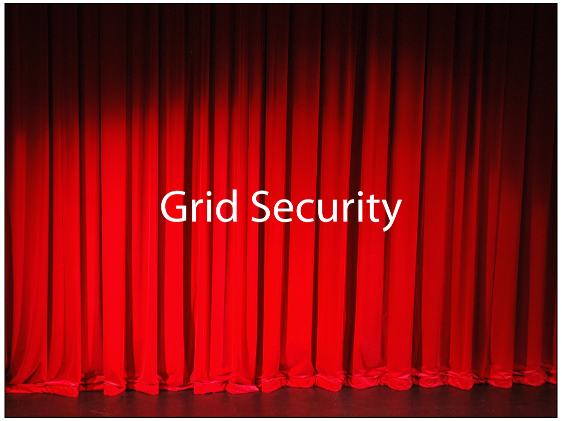
Globus Toolkit V4
- Service-oriented architecture
- A lot of WSRF-compliant services



Open Grid Service Architecture
- A big working group in the Global Grid Forum
- Tries to standardize the architecture used in grids
- Make components interchangeable



- Security: How to authenticate and authorize users in a decentralized way?
- RM: How to manage resources without a central entity? How to schedule?
- DM: How to handle huge amounts of data?
- IS: How to retrieve information from other sites? How to know there is another site?



Security = Encryption
+ Authentication
+Authorization
+ etc.

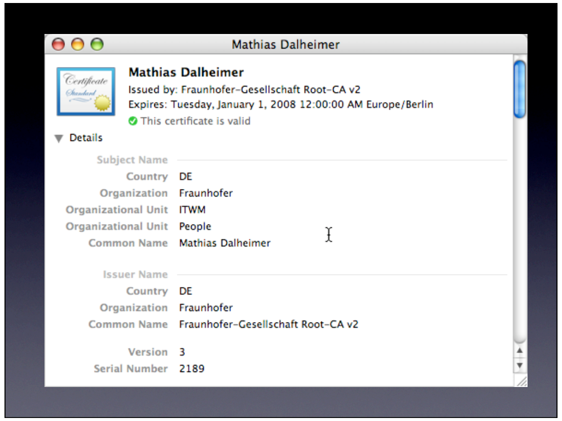
Encryption

- Data needs to be encrypted when sent over unsafe networks
- Ususally done with TLS (Successor of SSL): Data is sent through an encrypted TCP tunnel.
- But: How to know that the TCP tunnel is pointing at the receiver?

Authentication

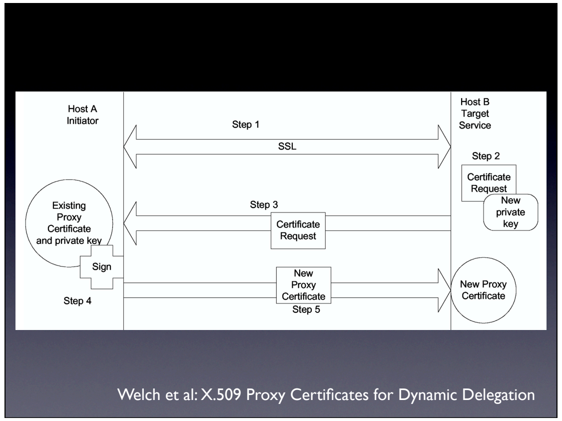
- Answers the question: “What is the identity of my counterpart?”
- Easy when dealing with just one organization: The system administration issues accounts, maintained in a central system (e.g. LDAP).
- More difficult in a distributed environment: How to do this for several organizations?

- X.509 is a standard for digital certificates (RFC 2459)
- Think of an electronic passport
- Unlike GPG, there is a Certificate Authority (CA) that issues the certs
- An X.509 cert represents the identity of a user.



- This is one of my certs
- Issued by the Fraunhofer CA
- Has an expiration date
- Describes who I am (DN), who the CA is
- So: The CA signs a RSA keypair which I can use.
- A X.509 cert has:
  * A DN of the holder
  * A DN of the CA
  * Public and private keys, signed by the CA
  * Reference to the CA policy
  * Reference to the cert revocation list.

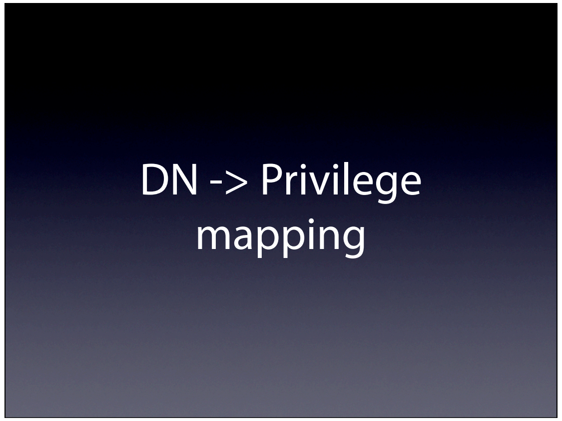
- Delegation: Transfer the rights of a user to a service
- Problem: The user doesn’t want to reveal his private passphrase
Two ways: Either use message based security (Unicore) or use proxy certs (Globus)



Dynamic Delegation with X.509 certs (simplified)
Goal: Delegate the user’s rights to a target service without the exchange of private keys.
- Step 1: Establish an integrity protected channel.
- Step 2: Target service generates a new public and private keypair.
- Step 3: Create a certificate request (CR) with the public key.
- Step 4: Initiator uses his own private key to sign the CR. Within the proxy certificate, the “signer” field is filled with the user’s public key (or another proxy certificates public key)
- Step 5: The new proxy certificate is sent back to the target service.

- We have encryption and authentication by now.

- Authorization: What is the user (once authenticated) allowed to do?
- Enforce the site policy.
- Relatively simple: When authentication is done, we know who is asking for the service.



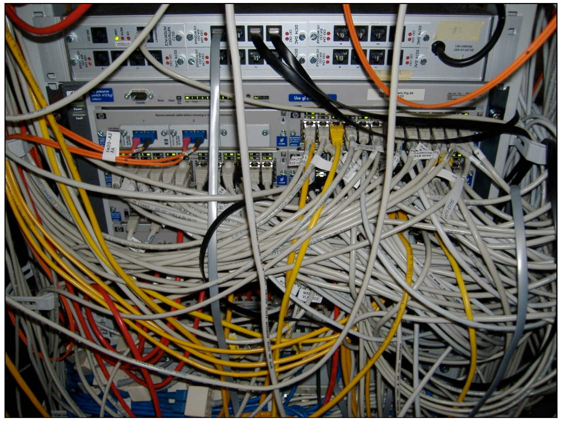
- A mapping of the distinguished name (DN) to the local privileges must be made.
- This can be done locally: A service maps DNs to local users.
- The user's privileges granted to the requestor.
- Usually: Unix security model.
- Unicore: UUDB, Globus: gridmap-file

- etc. is missing
- Usual security precautions: Installed updates, bugfixes, firewalls, ...
- Single Sign-On: Reduces the risk of lost & stolen passwords (and is convenient for the user)

What is a grid resource?

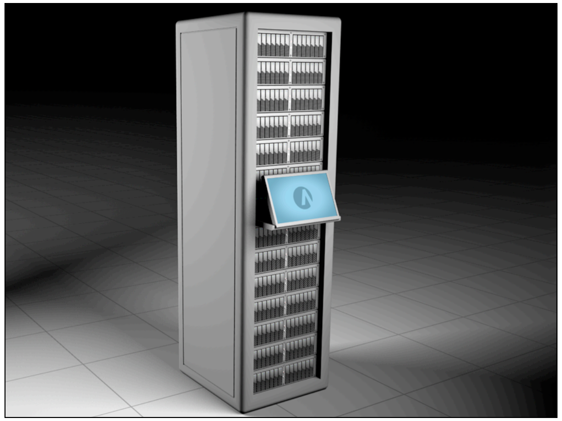
e.g. a cluster

or a radio telescope



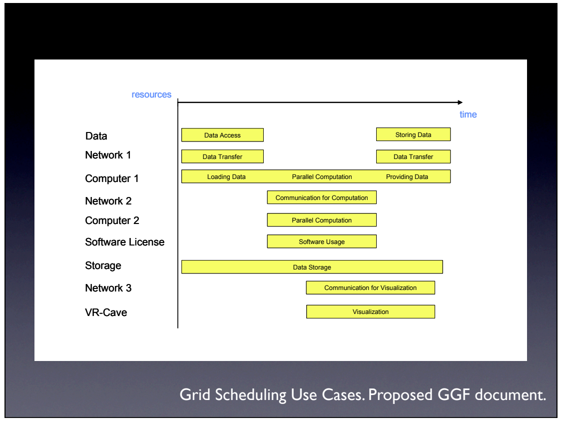
network resources

(graphic) workstations

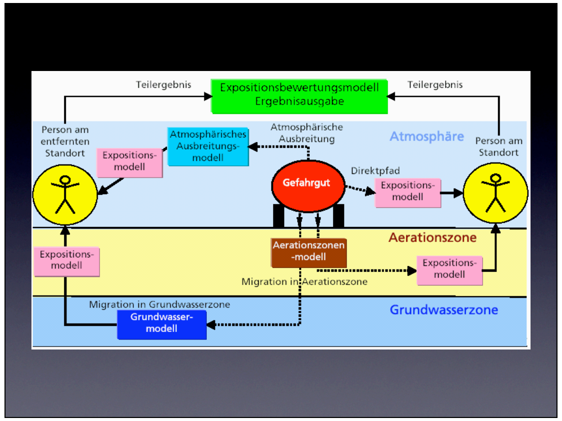


storage resources.

Managing resources in the grid
- There is maintenance to do -> not covered here
- Assign jobs to resources, monitor them
- Relatively easy for one resource
- But often, jobs have a difficult workflow.



Example of a complex job
- Coallocation
- Software licenses
- ...



Gekoppelte Modelle: ERAMAS
- Gekoppelte Modelle zur Bewertung der Schadstoffausbreitung im Grundwasse, im Boden und in der Luft.

Grid Resource Management

- Resource Discovery
- Information Gathering
- Job Execution

Schopf: Ten Actions when Grid Scheduling

Refer to Schopf, “Ten steps when grid scheduling”
- Resource Discovery: which resources are available?
- Information Gathering: what is the current situation of the resources
- Job Execution: run and monitor the job on the selected resource

Resource Discovery

- Authorization filtering
- Application requirement definition
- Minimal requirement filtering

Deals with the search for available resources, ends with a list of execution candidates
- Authorization filtering: determine the set of resources the user submitting the job has access to.
- Application requirement definition: determine what the application requirements are (Architecture, CPU, Memory, OS, Libs)
- Minimal requirement filtering: create a list of resources that fulfill the applications requirements.

Information Gathering

- Dynamic information gathering
- System selection

Determines where to execute the job.
- Dynamic information gathering: the current situation on each location needs to be considered, e.g. free cluster nodes, load, network IO.
- System selection: based on the gathered information, an execution location will be selected. One needs to consider network transfers, cluster walltimes, local policies, pricing, reliability...

Job Execution

- Advance reservation
- Job submission
- Preparation tasks
- Monitor progress
- Cleanup tasks

- Advance reservation: Especially needed when doing coallocation. Coallocation refers to several jobs running in parallel on different resources, advance reservation (hopefully) ensures resource availability for the given job.
- Job submission: the job is submitted to the resource
- preparation tasks: prepare the resource for job execution, e.g. copy input files, create directories, stage application
- monitor progress
- job completion: notify the user
- cleanup tasks: copy the results and delete temporary files.

Different architectures

There are 3 architectures for grid schedulers.

Centralized Scheduling

A central scheduler manages all resources
- not scalable
- difficult with multiple organizations / policies

Hierarchical Scheduling

A high-level scheduler receives jobs and assigns them to local resource schedulers (typical setup)

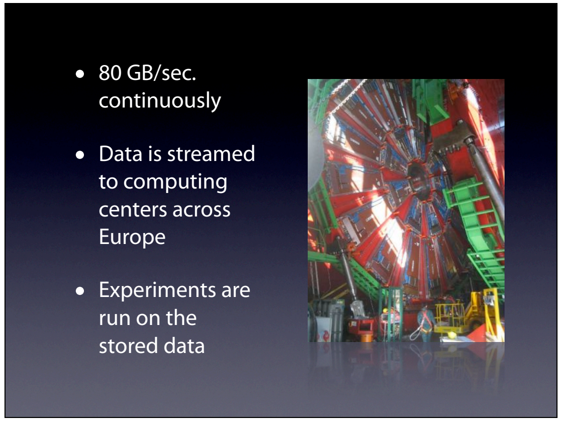
Decentralized Scheduler

There is no central scheduler, but a distributed queue. Each local scheduler retrieves its jobs from the distributed queue.

Negotiation

Often, it is necessary that a resource guarantees a certain QoS
- Service level agreements need to be made
- Usually in a negotiation process, see e.g. GRAAP-WG of GGF.

EGEE = Enabling Grids for E-Science in Europe
An EU-funded project that aims at creating a grid infrastructure for researchers in Europe
Motivated by the Large Hadron Collider and its experiments,
situated at CERN in Switzerland



Data management challenges

- Manage huge amounts of data
- Provide data where it is needed
- Determine where it is needed
- Help to find specific datasets

GridFTP

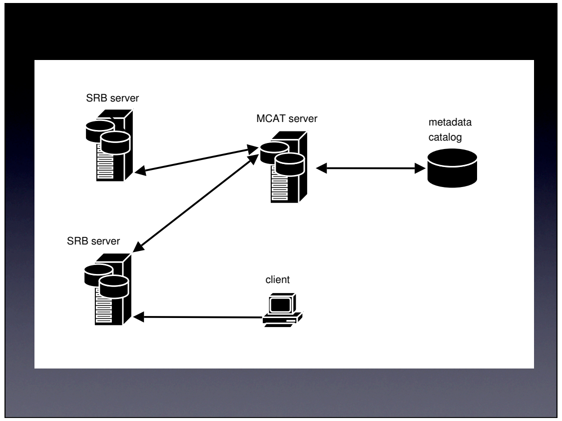
- Add-on to FTP:
  - Uses GSI to authenticate users
  - All data transfers are encrypted
  - Allows third-party transfers
  - Striped File transfer
  - Partial transfers

Striped file transfers: Think of RAID Level 0
In addition: Automatic negotiation of TCP buffer/window sizes.

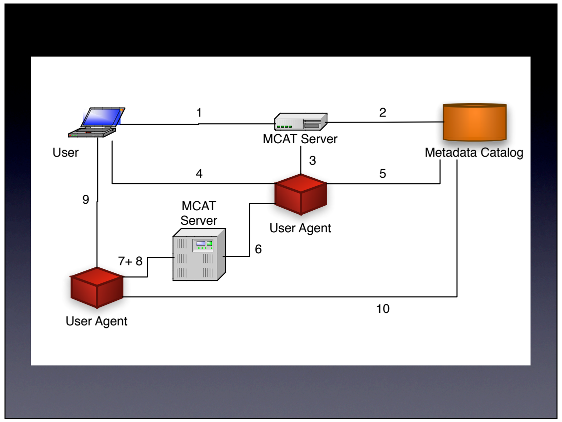
But: Not sufficient

- Doesn't abstract from the resource
- No search mechanisms for specific datasets
- Doesn't integrate different storage technologies
- No support or replicas and caches

Storage Resource Broker
- Developed by the San Diego Supercomputing Center (SDSC)
- Provides a comprehensive system for data management:
    * Integration of data with meta-data
    * Provides sophisticated storage management (Replication & Caching)



A SRB zone consists of the following entities:
- Exactly one metadata-catalog which stores information about physical volumes, metadata etc.
    * Contains all metadata
    * Implemented on top of a DBMS
    * Responsible for the abstraction from physical resources - associate logical names with datasets
    * Search for datasets based on associated metadata
    * User authentication
- Several SRB-Servers have storage resources like DBs, filesystems and tapes attached.
- Special MCAT-Server which accesses the metadata catalog.

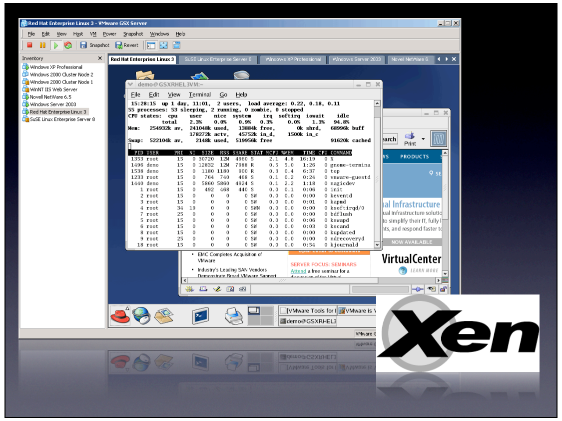


Workflow of a data access operation:
(1) SRB-Client connects to the MCAT Server and tries to authenticate
(2) The MCAT-Server compares this credential to the one stored in the metadata-catalog
(3) If the authentication is valid, an agent process is created. Note: The agent processes are usually running on the MCAT server.
(4) The client submits its request to the agent.
(5) The agent authorizes the query using the metadata catalog
(6) If the data is stored on another SRB server, the agent connects to it.
(7) A local agent is created for request processing.
(8) The local agent accesses the storage resource
(9) Results are sent back to the client

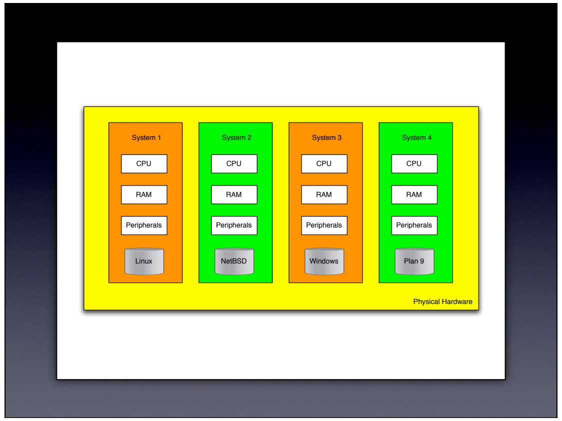
Additional concepts

- Fine-grained access control
- Ticket mechanism allows temporary access delegation
- Automatic or manual replication of datasets
- Caching on fast media while archiving on slow media

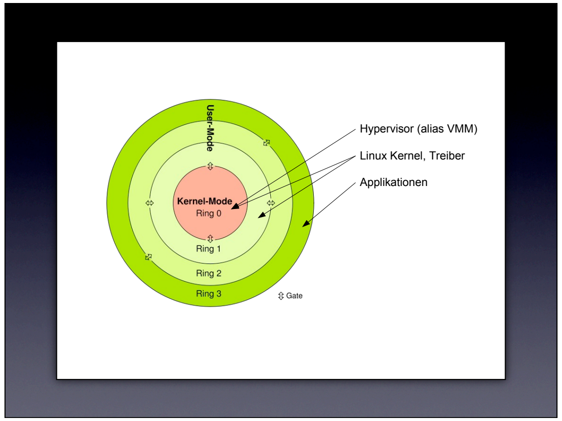
Speak VMWare, Xen, OpenVZ, ...



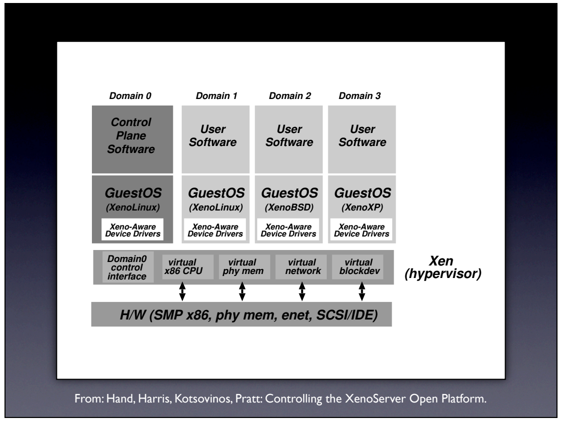
Now, I talk about this stuff.
- I will focus on Xen - other virtualization products are not included here, but may be used as well



Xen virtualizes the physical hardware of a system
- The hardware is shared between different instances (possibly different OS)
- Each instance “sees” its own CPU, memory etc.
- Peripherals are mapped in the virtual machine
- For Xen: OS has to be adjusted to the Xen VMM

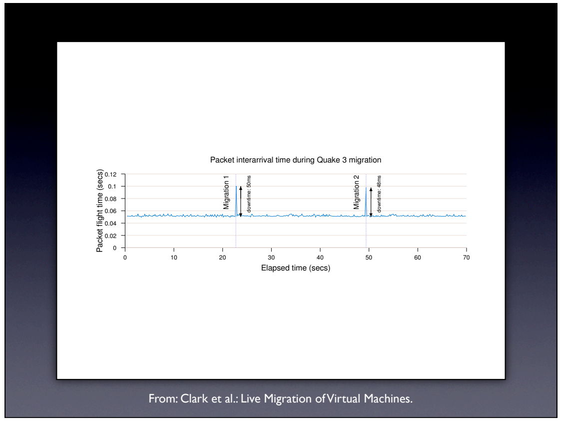


x86 Ring modes
- protection of certain operations (accessing memory, IO etc. only in ring 0)
- typically, kernel in ring 0, apps in ring 3
- xen shifts kernel to ring 1 (partially), mostly hypervisor in ring 0
- hypervisor multiplexes different kernels

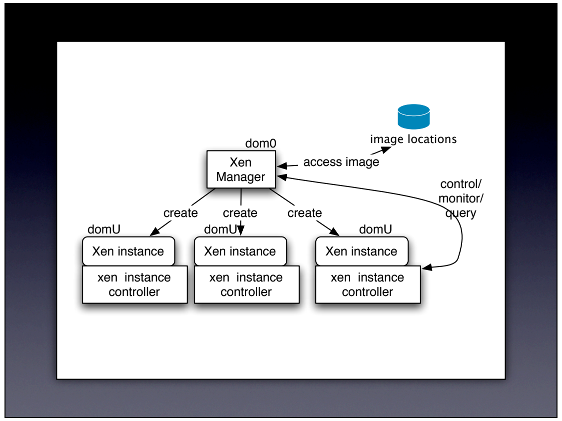


- Domain0 is privileged - administration of all other domains can be done here.
- Domain0 provides the drivers etc - the hypervisor is only responsible for dispatching CPU, memory etc.
- Device drivers are split in two parts:
    - the xen part (real driver in dom0)
    - the domU stubs
    -> The OS must be adjusted to the xen architecture (newer CPUs: special instructions that work around it)

Virtual machine images are containers for the whole virtual system.
-> Installation can be done and tested on a single system,
deployment in the wild
-> For commercial codes: License can be part of the image.
-> Migration of running systems is easy

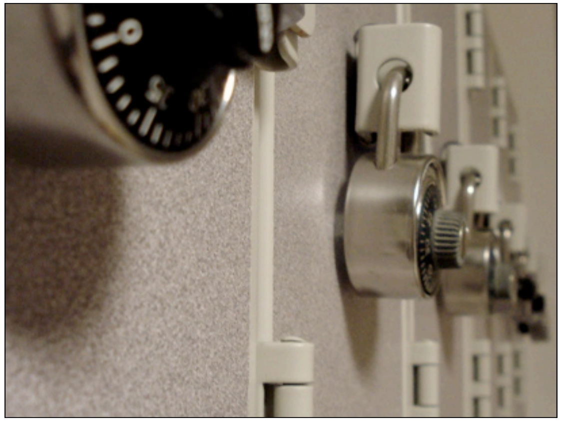


Live migration of a quake server
- packet flight time increases by 50 ms
- system remains fully operational!
-> You will investigate further during the exercises



Deployment:
- Images are copied from a central repository and started.
- How to decide whether a system is up and running *correctly*?

Quality assurance can be done with the image infrastructure:
- Use only tested and certified (minimal) Images
- Specification of hosting-environment can be done, “certified gold provider”
- Image maintenance: only the “golden master” must be updated



Secured Containers:
- Data can be shipped with the container (encrypted)
- Decryption in memory during execution of the image
- Better security: No data lies around, goes in backup system, ...
- The providers may distinguish themselves by providing different levels of security (“gold provider”)

Cloud Computing:
- Do not provide complex services
- But allow people to launch their complete server images
- Billing per CPU-hour and storage used
- currently in beta stage
- find more at http://www.amazon.com/gp/browse.html?node=201590011

http://md.gonium.net