

1st Hands-on dCache workshop
24. Januar 2008
DESY Hamburg



dCache

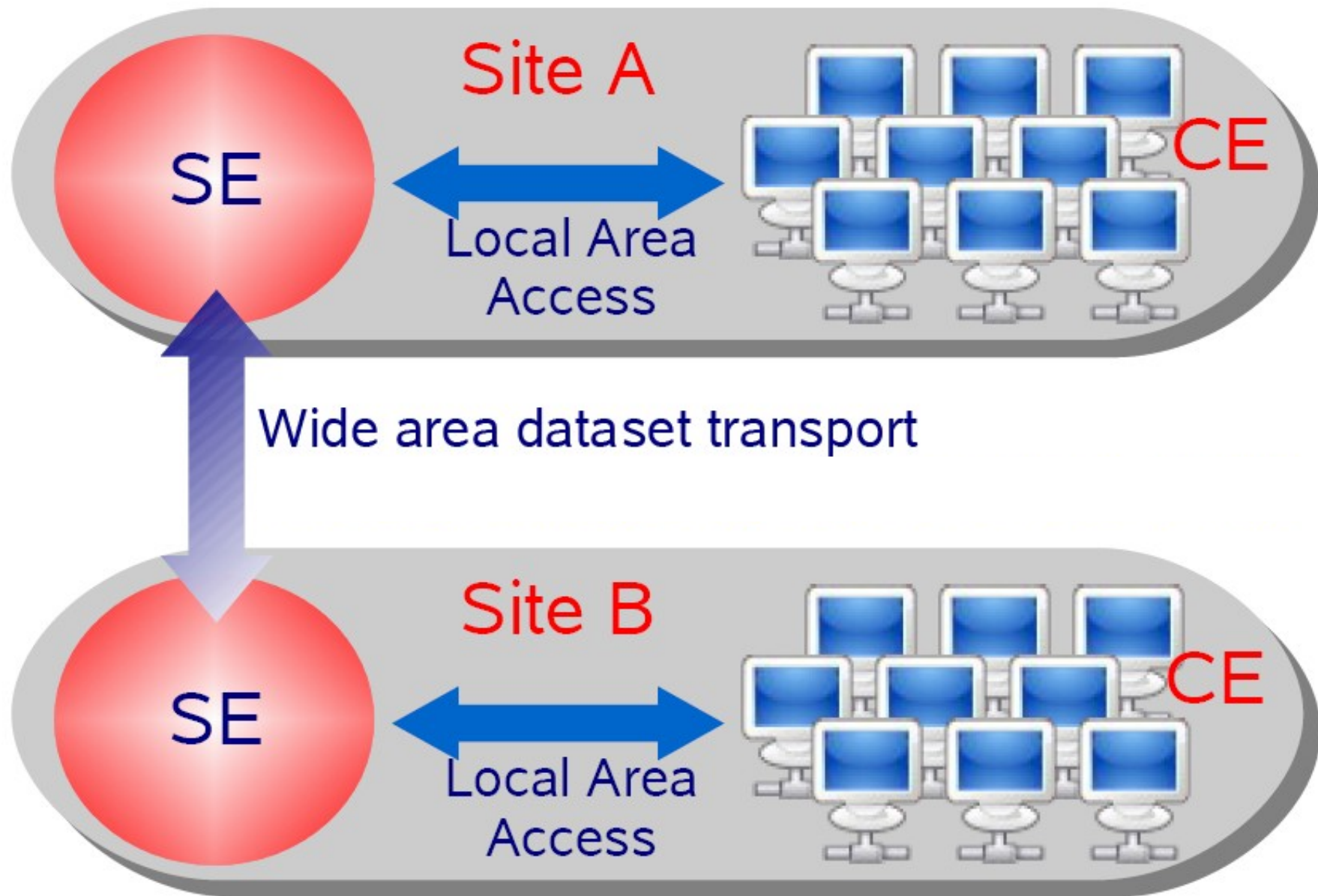
An Introduction

Martin Radicke





Introducing the Grid StorageElement (SE)

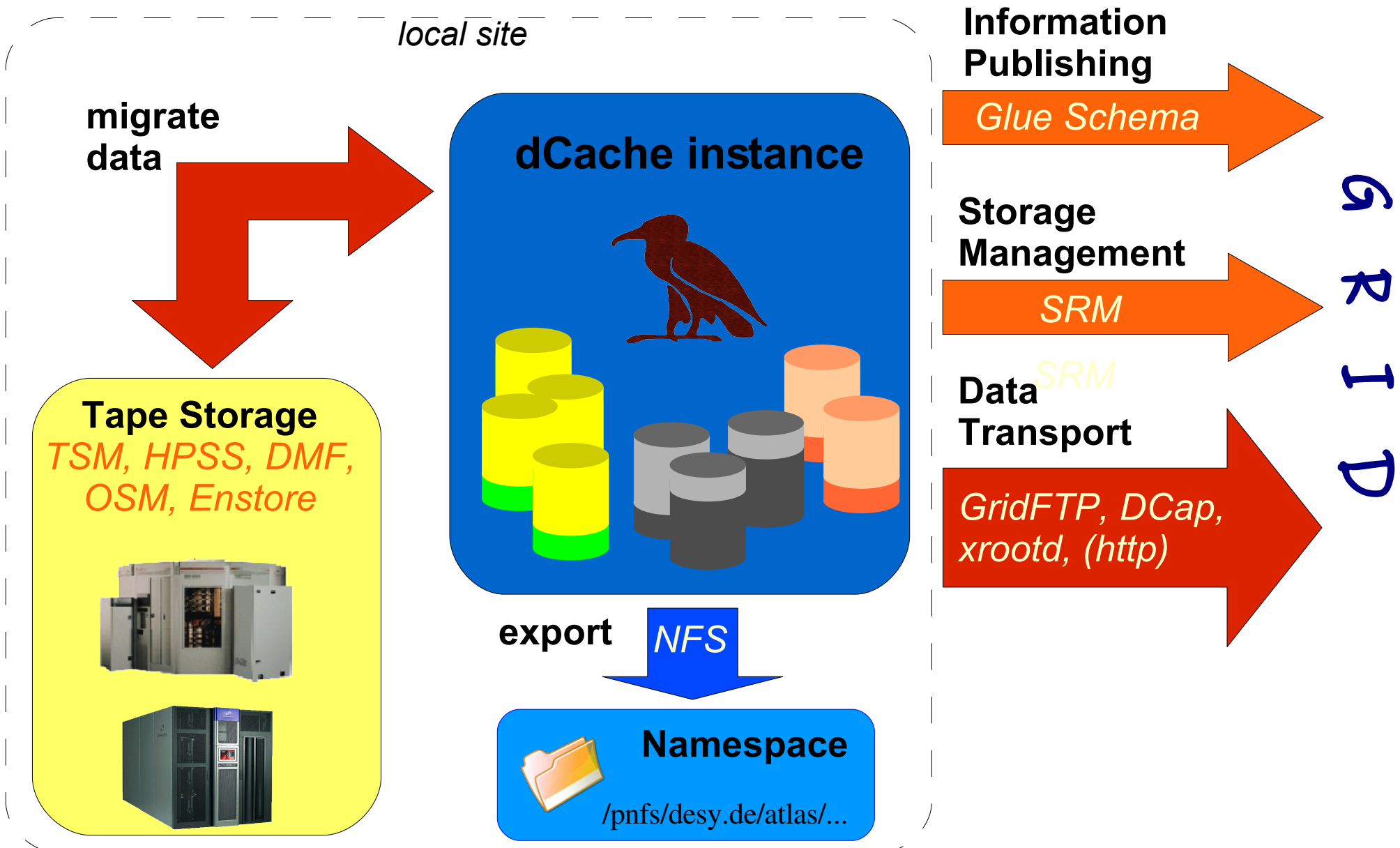




- ▶ serving large amounts of data locally
 - access from the local site (from the workernodes that form together the CE)
 - huge number of parallel file requests
 - low latency, Posix-like random access
- ▶ exchanging big datasets across datacenters
 - file streaming from/to SEs on remote sites
- ▶ storage control
 - space reservation for guaranteed streaming bandwidth
 - space properties (tape only, tape + disk, disk only)
- ▶ information publishing
 - available space (max, free, used), types of spaces
 - Which VO owns which space?



The dCache SE – a bird's view





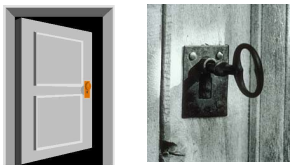
- ▶ managed storage on off-the-shelf hardware
- ▶ combines hundreds of commodity disk servers to get a huge PetaByte-scale data store
- ▶ strictly separates between namespace and data repositories → increased fault tolerance
- ▶ allows several copies of a single file for distributed data access
- ▶ internal load balancing using cost metrics and inter-pool-transfers
- ▶ automatic file replication on high load (hotspot detection)



- single rooted namespace service
- one central instance ensures consistency, allowing multiple file copies

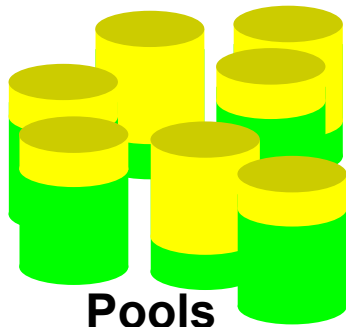


- Storage Resource Manager
- identified by Storage URL (SURL)
- selects Door based on load and agreed transfer protocol



Doors

- protocol-specific entry points to the file repository
- identified by Transport URL (TURL)
- contacted by Clients to initiate file transfer



Pools

- diskcache holding 0..n copies of files registered in namespace
- get selected and do the actual file transfers
- recycle cached files according to LRU algorithm, if space needed
- migrates files to and from Tertiary Storage



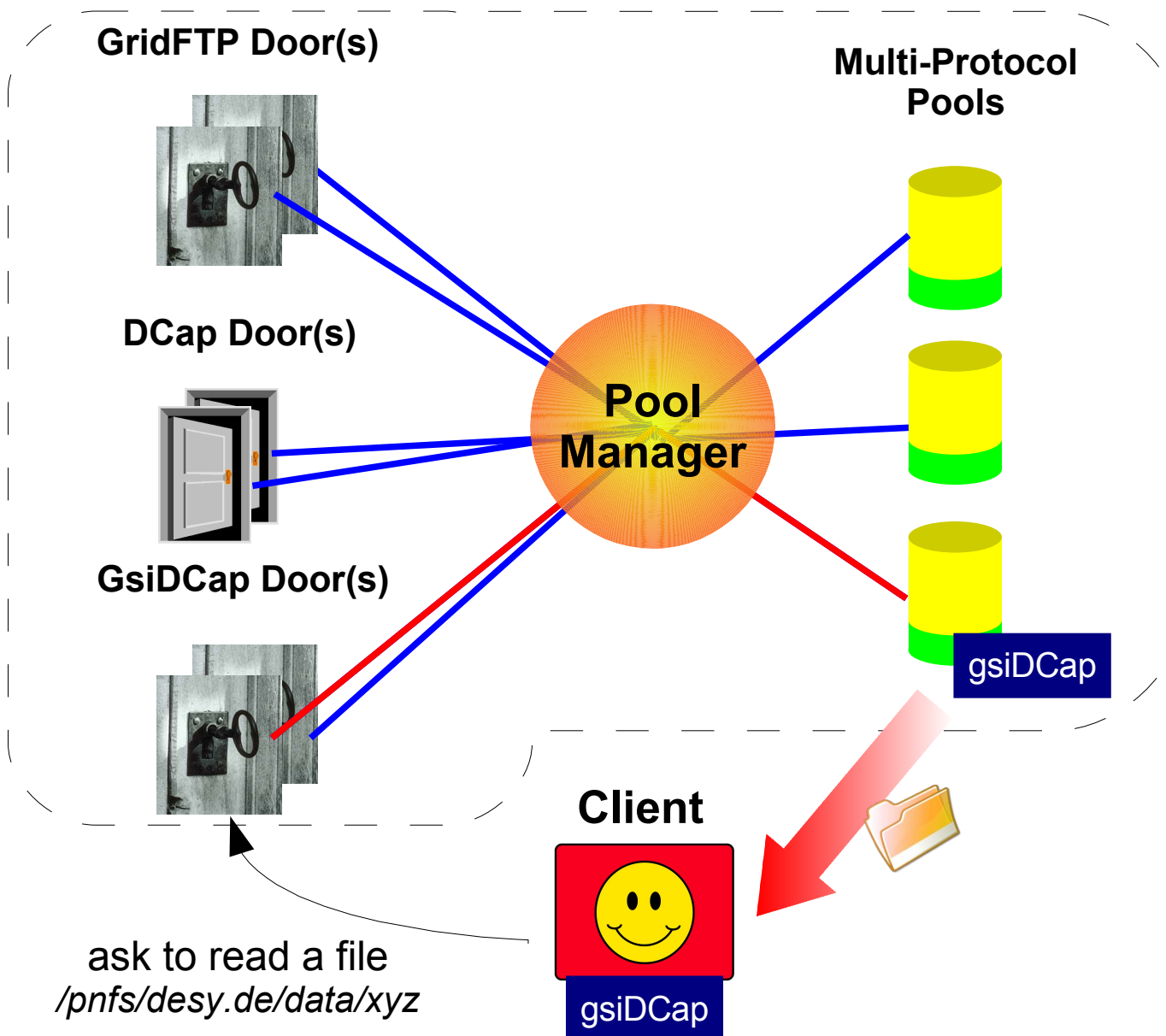


- ▶ done by the **PoolManager** (the “heart of dCache”)
- ▶ Select a set of pools which matches the following criteria
 - Protocol
 - Dataflow Direction (read, write, restore from tape, pool2pool)
 - Directory Subtree
 - Client IP Subnet
- ▶ Out of these, select the “best” target pool with lowest cost
 - cpu load (number of running transfers)
 - free disk space

➡ **chosen pool is asked to fulfill request**



Logical flow of a file request





▶ local, posix-like access

- DCap – dCache's native protocol
 - client (-library) provided by dCache
 - GSI authentication supported
- xrootd - access from within ROOT

▶ remote, streaming access

- GsiFTP – the Grid standard
 - active, passive
 - multiple streams
- HTTP(s) – prototype available

*Tigran's talk will
give more details
on this*



- ▶ an interface for standardized access to grid SEs
- ▶ prepares file transfers
 - client $\Leftrightarrow$ SE (file up- and download)
 - SE $\Leftrightarrow$ SE (3rd -party transfer across grid sites)
 - negotiates the transfer protocol (mostly GsiFTP)
- ▶ advanced features
 - pre-staging of files
 - provides directory functions (e.g. ls, mkdir)
 - space management (reservation, ls)
- ▶ version 1 and 2.2 supported by dCache

*not covered
during this
workshop*

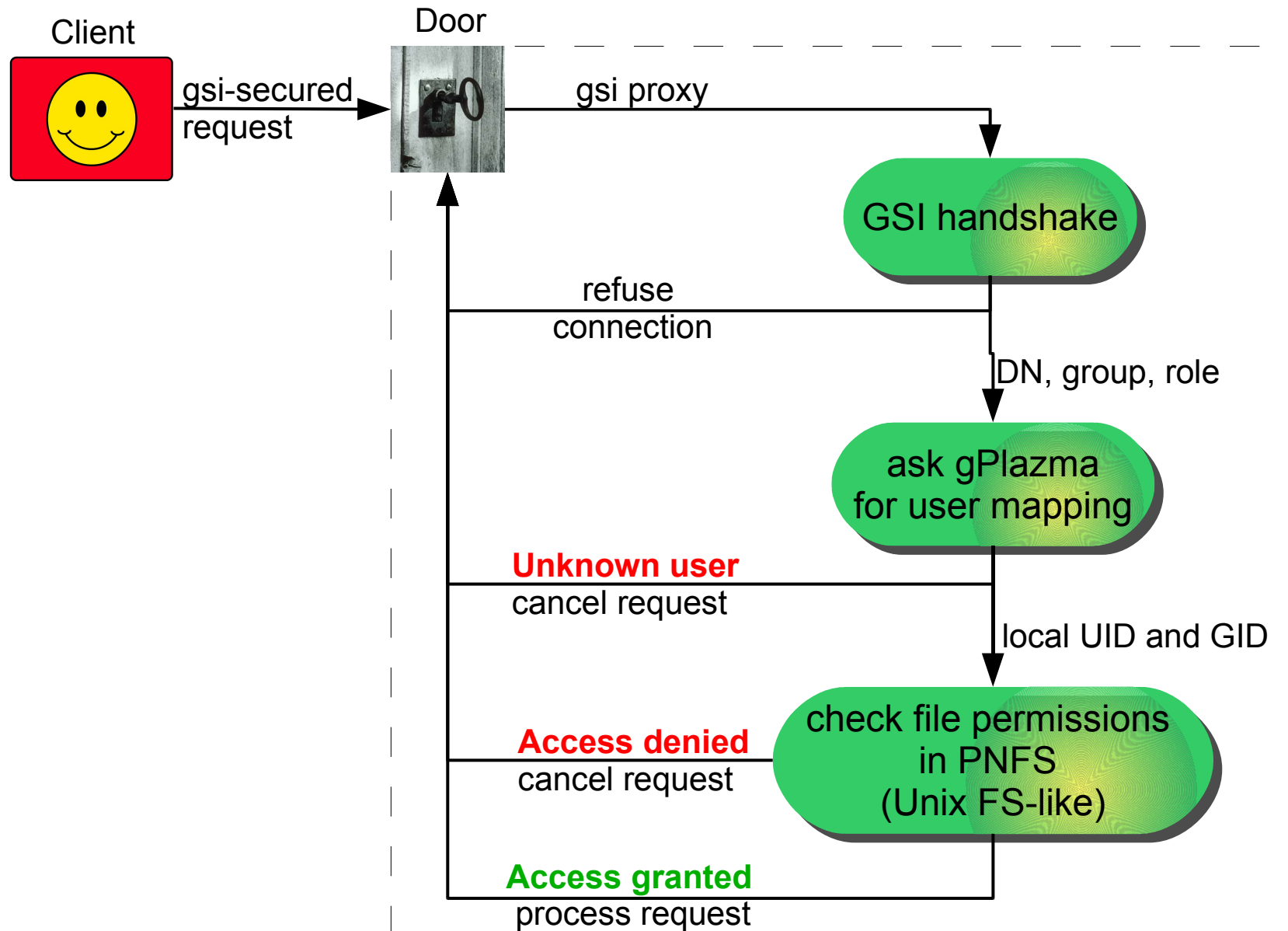


- ▶ one central user mapping service: **gPlazma**
 - used by all gsi-enabled processes (e.g. the gsiFtp door)
- ▶ **maps user DN + VOMS extensions to a local user account**
- ▶ 3 different methods
 - kpwd-file
 - dCache-specific solution
 - maps DN => local user id and group id
 - gridmap-file
 - legacy, grid-wide solution
 - queries VOMS server
 - maps DN => local user id and group id
 - VORoleMap-file
 - maps DN + Group + Role => local user id and group id
 - queries VOMS server
 - most advanced method (e.g. one user in two VOs)

used during this workshop

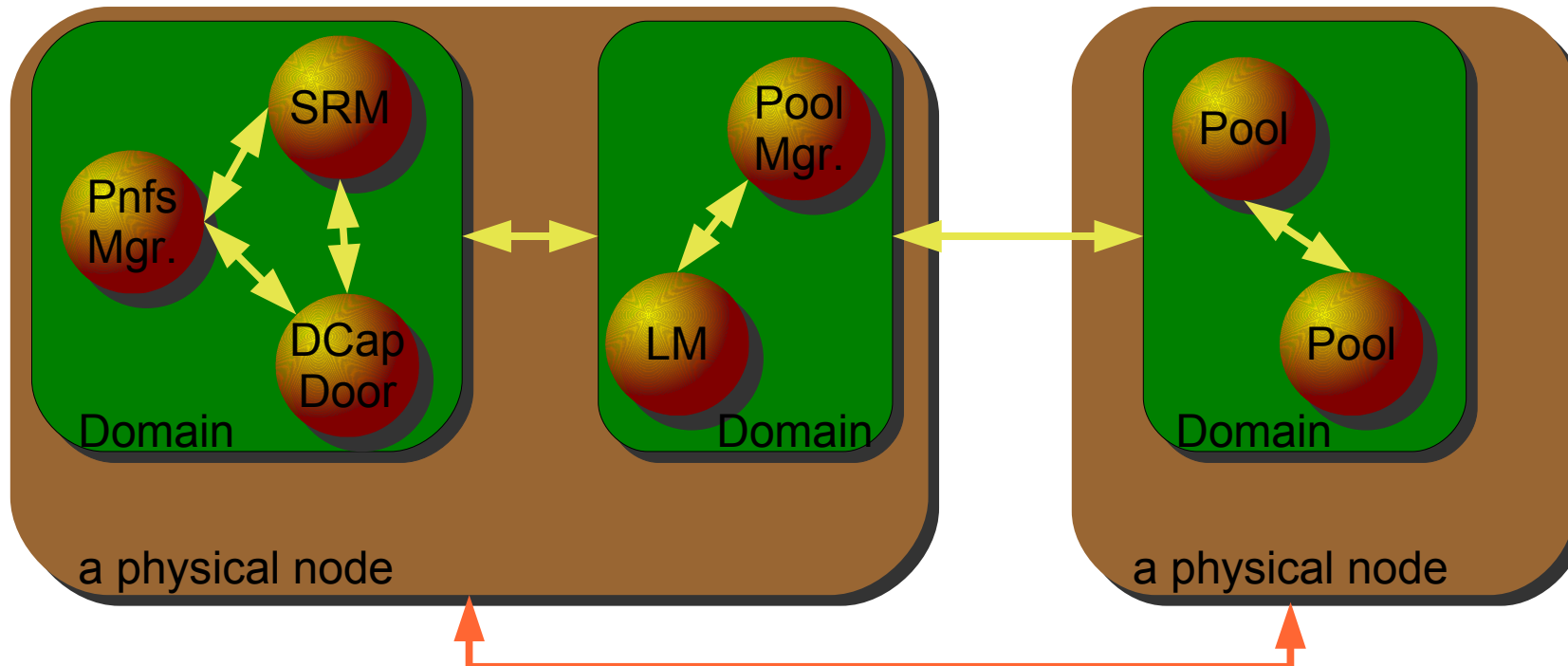


Authentication & Authorisation





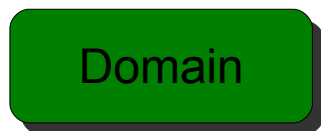
dCache as a distributed system



- small, individual entity
(= Thread)
- communicates with other cells



logical message bus



- container hosting cells
(= java process)
- multiple instances per machine possible



Local Area Network



Installation & Deployment



- ▶ current production version: dCache 1.7
- ▶ we are in transition to dCache 1.8
 - this workshop is already based on dCache 1.8.0
 - most Tier-1s already upgraded to 1.8
 -
- ▶ **recommended Version: dCache 1.8**
 - currently in dCache unstable repository only (yum)
 - or take it directly from www.dcache.org
 - will move to stable soon
- ▶ source code (SVN-WebView) available under special Open Source License
 - binaries for Linux, Solaris



- ▶ manual installation (standard, full flexibility)
- ▶ or: automatic installation via **YAIM** installer
 - a full single-host dCache instance in 15min as a starting point
 - supports adding of pool nodes or other specific node types
- ▶ Documentation on <http://www.dcache.org/>
 - dCache Book, Wiki (incl. 3rd party contributions)
 - publications and talks
 - news, announcements