SRM 2.2

Shaun de Witt (s.de.witt@rl.ac.uk)

Introduction

- Design
- Dependencies
- Set-up
- Requirements
- Support
- Upcoming

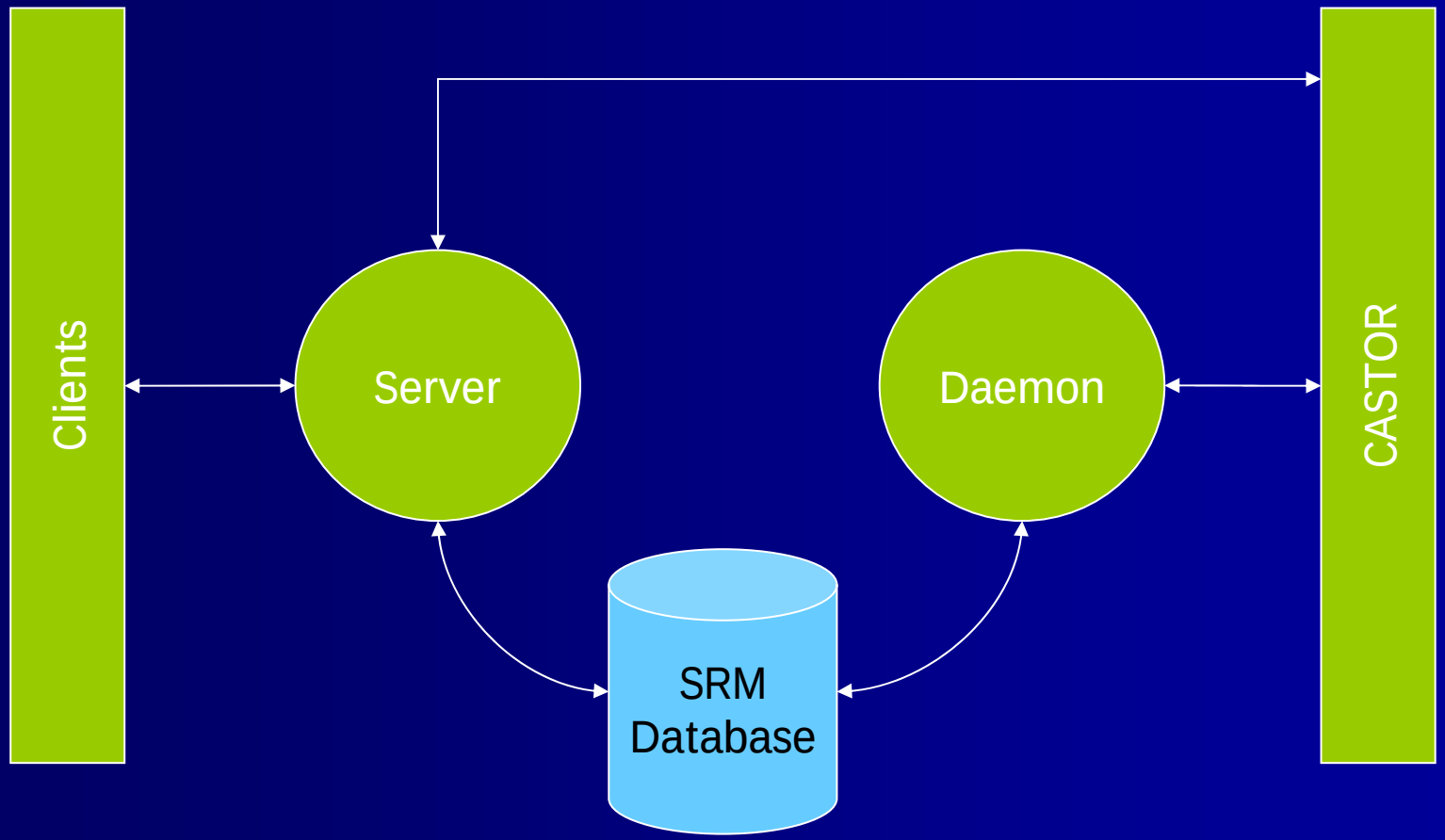
Design

- Database Centric
  - Heavy re-use of CASTOR core components
    - Thread library in daemon
    - DLF for both
  - Request Flow depends on request type
    - not all functions hit SRM database...
    - ...but all of them hit the CASTOR database

Design

- Secure
  - Uses GSI authentication
    - Cgsi plugin for gsoap 2.7
  - VOMS ready
    - whatever that means
- Server and Daemon multithreaded
  - configurable # threads
  - Throttled number of heavyweight threads
    - srmLs and possibly srmChangeSpaceForFiles

Design



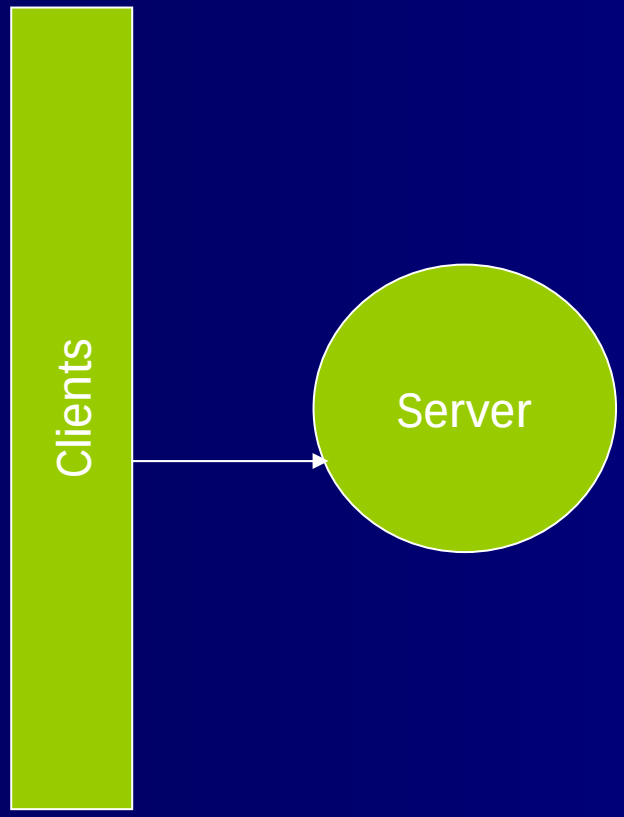
Data Flows

- Two types of request in SRM spec
- Synchronous
  - Client waits for response
  - Typically does not go through SRM database of daemon
  - Server talks directly to nameserver and/or stager
- Asynchronous
  - Request stored in SRM database.
  - Associated with srmStatusOfXXX call

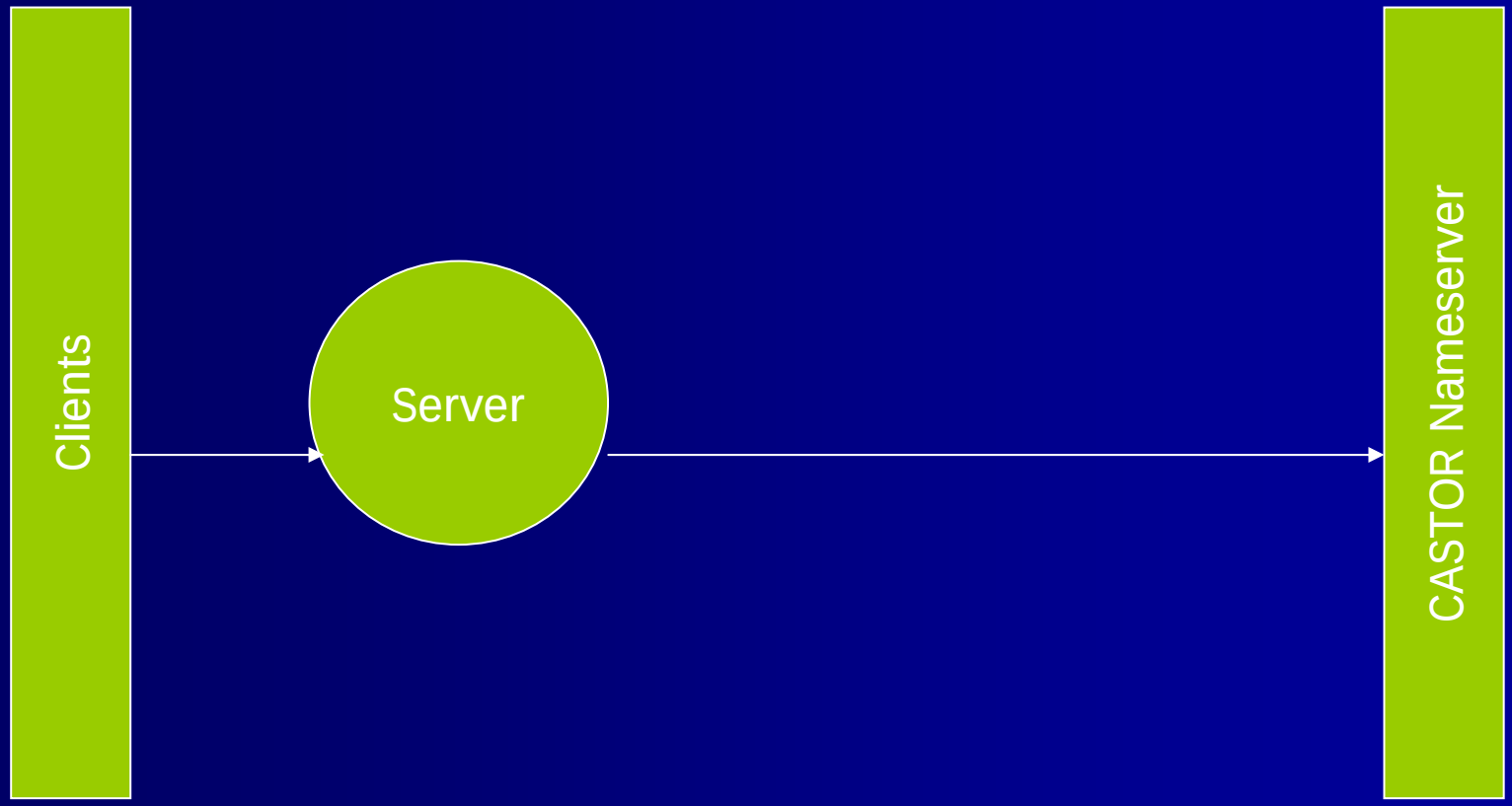
DataFlows

- Two Examples
- Directory Function (sync)
  - e.g. srmMkdir, srmRmdir
- PUT function (async)
  - srmPrepareToPut
  - srmStatusOfPutRequest
  - srmPutDone

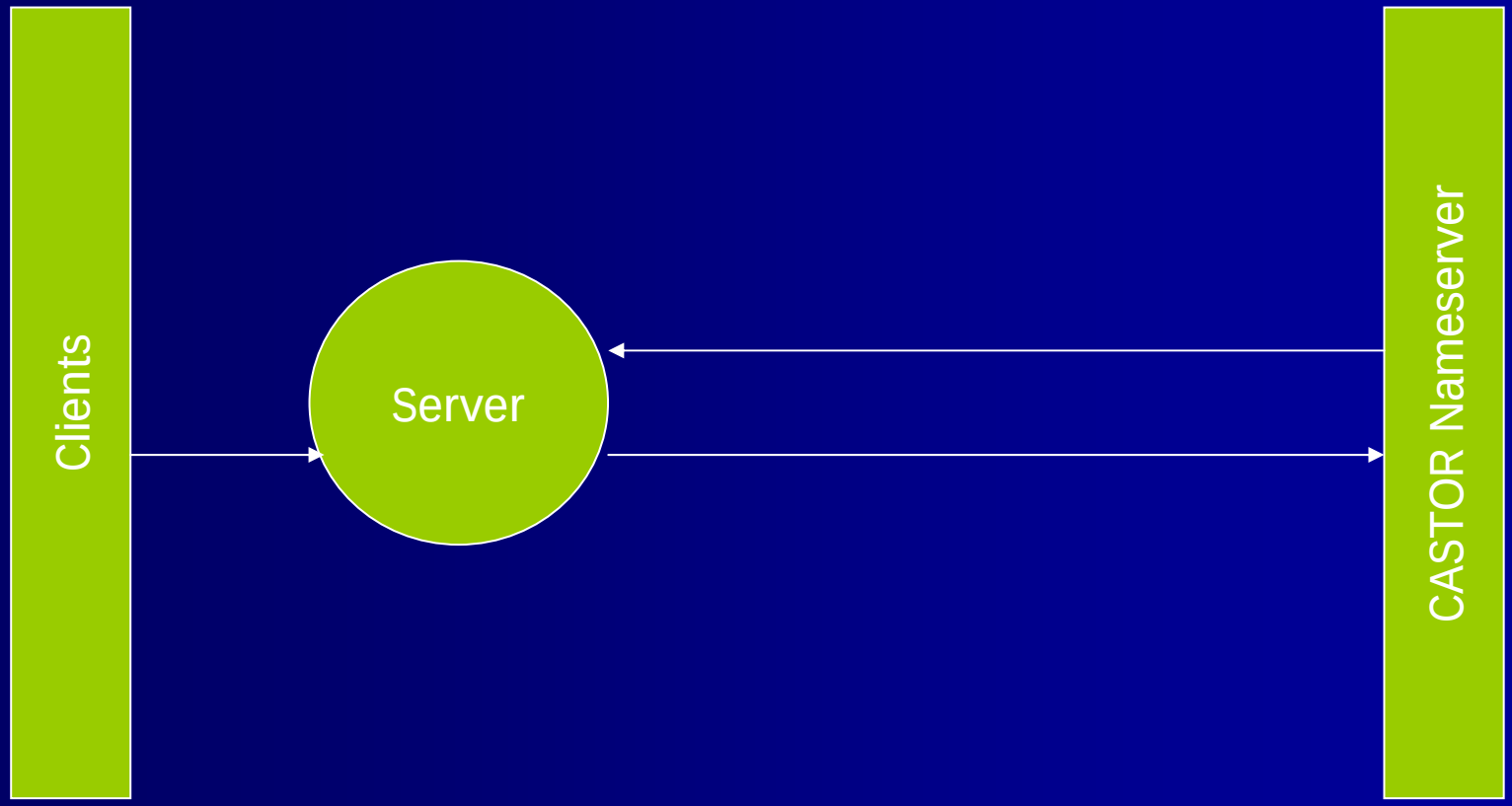
Directory Functions



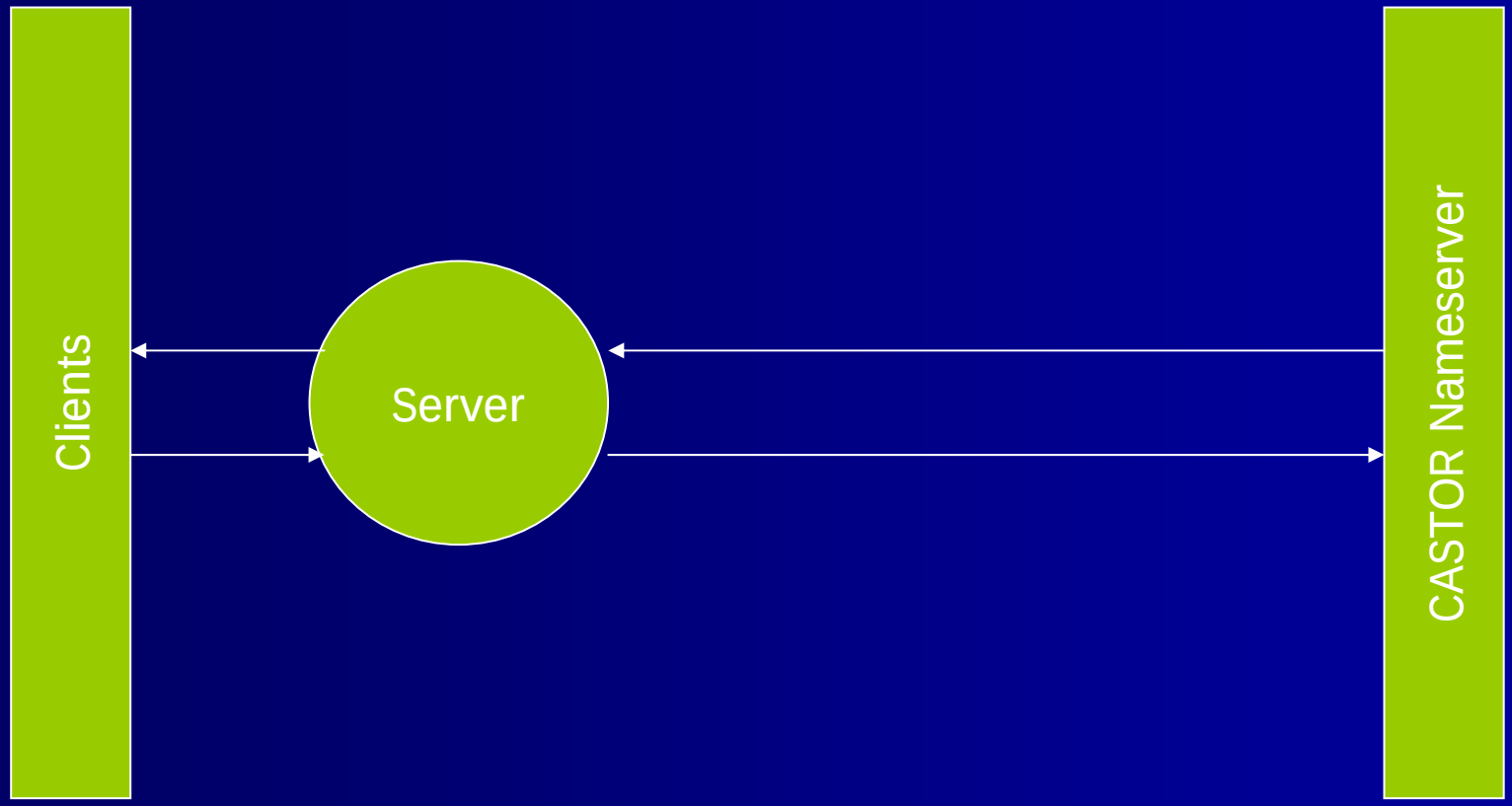
Directory Functions



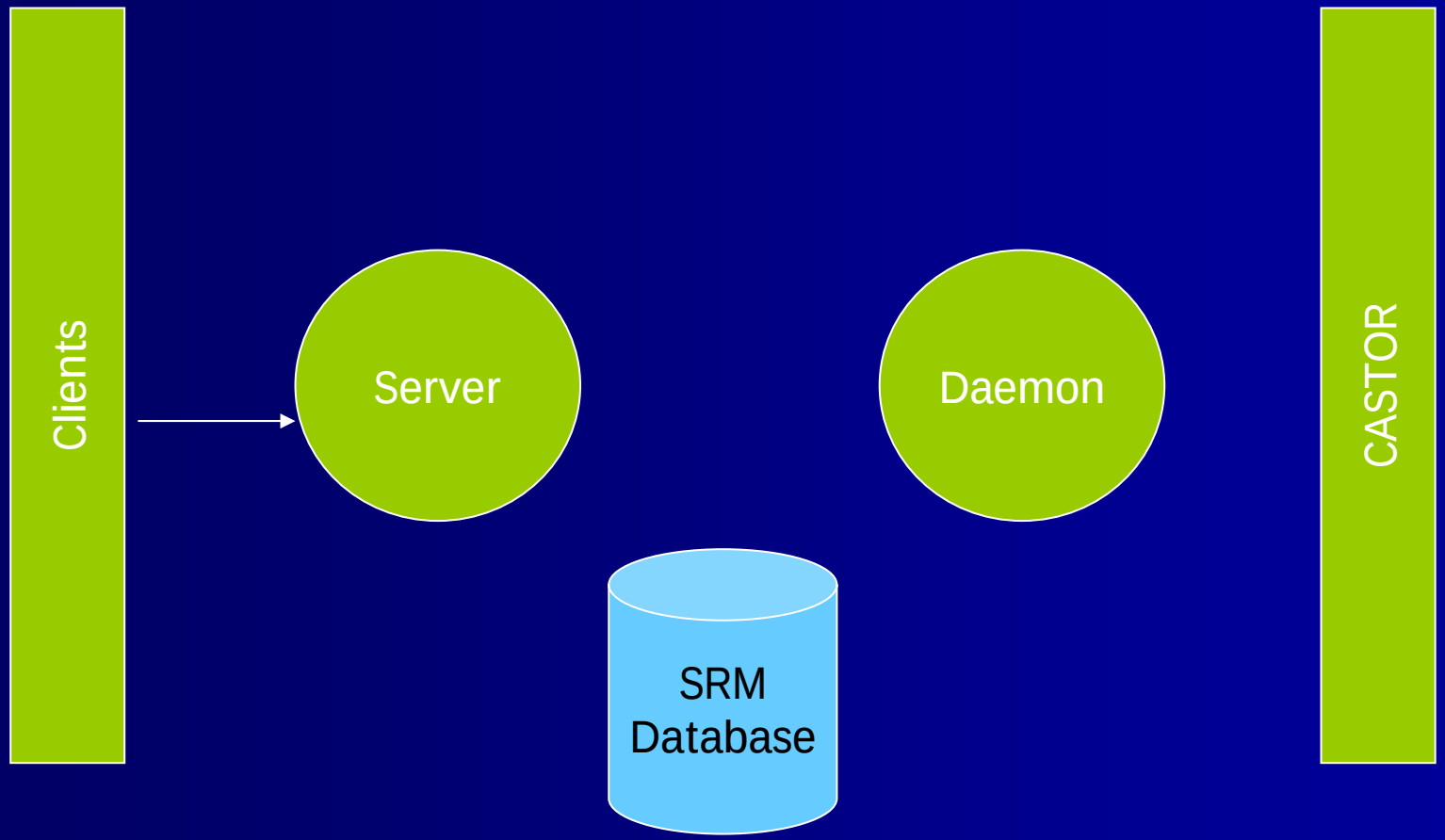
Directory Functions



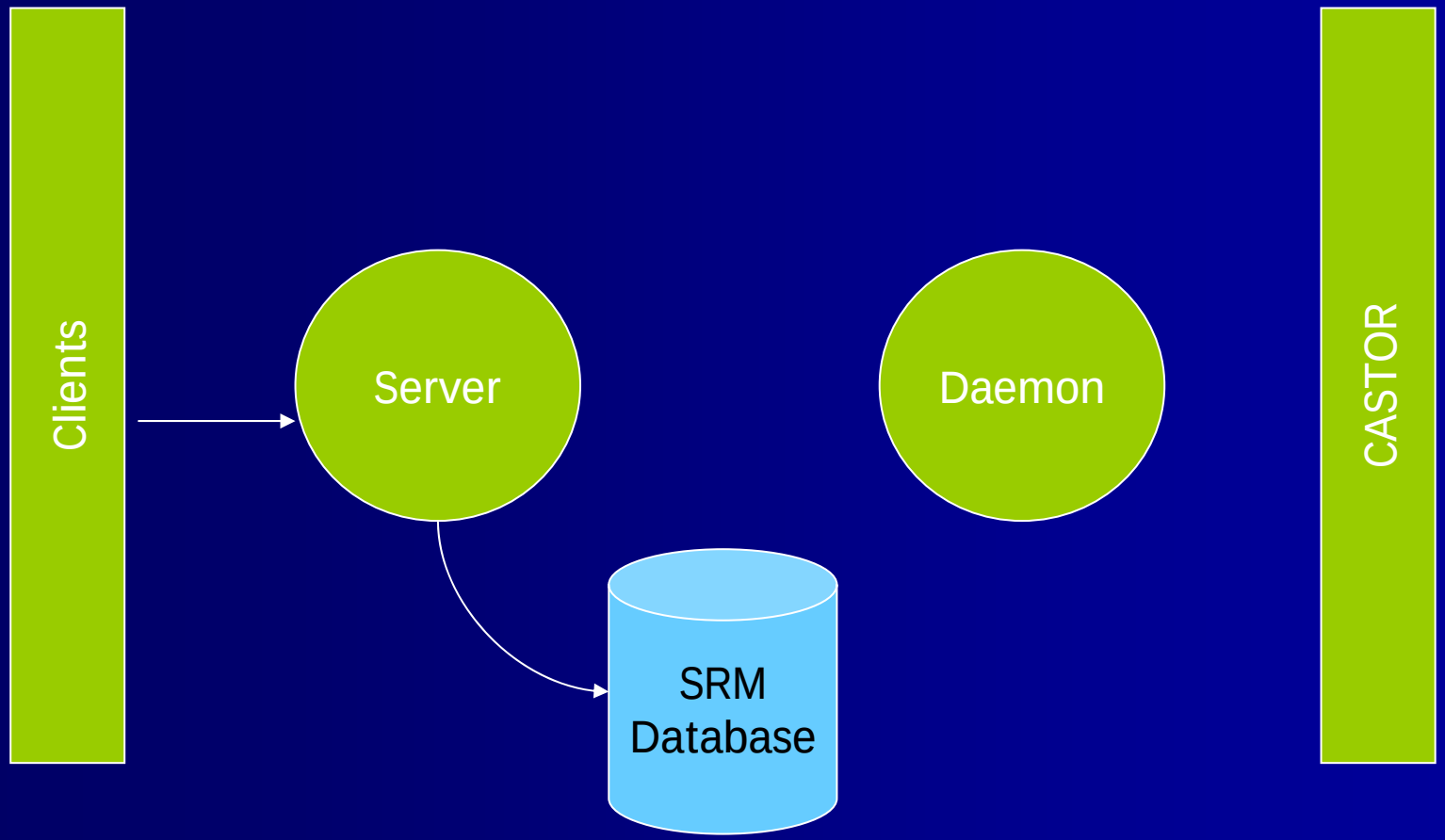
Directory Functions



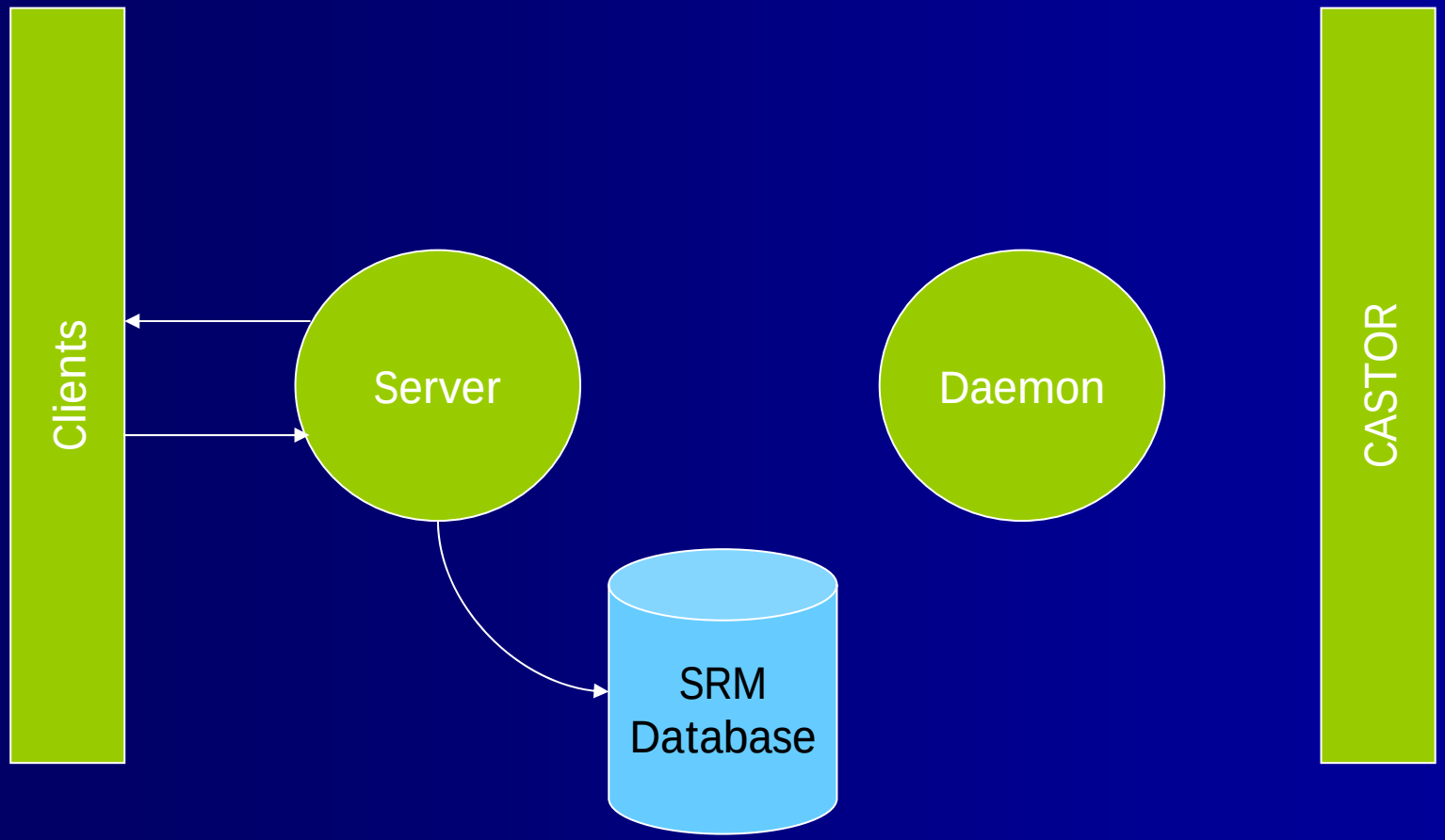
PUT operation (Prepare)



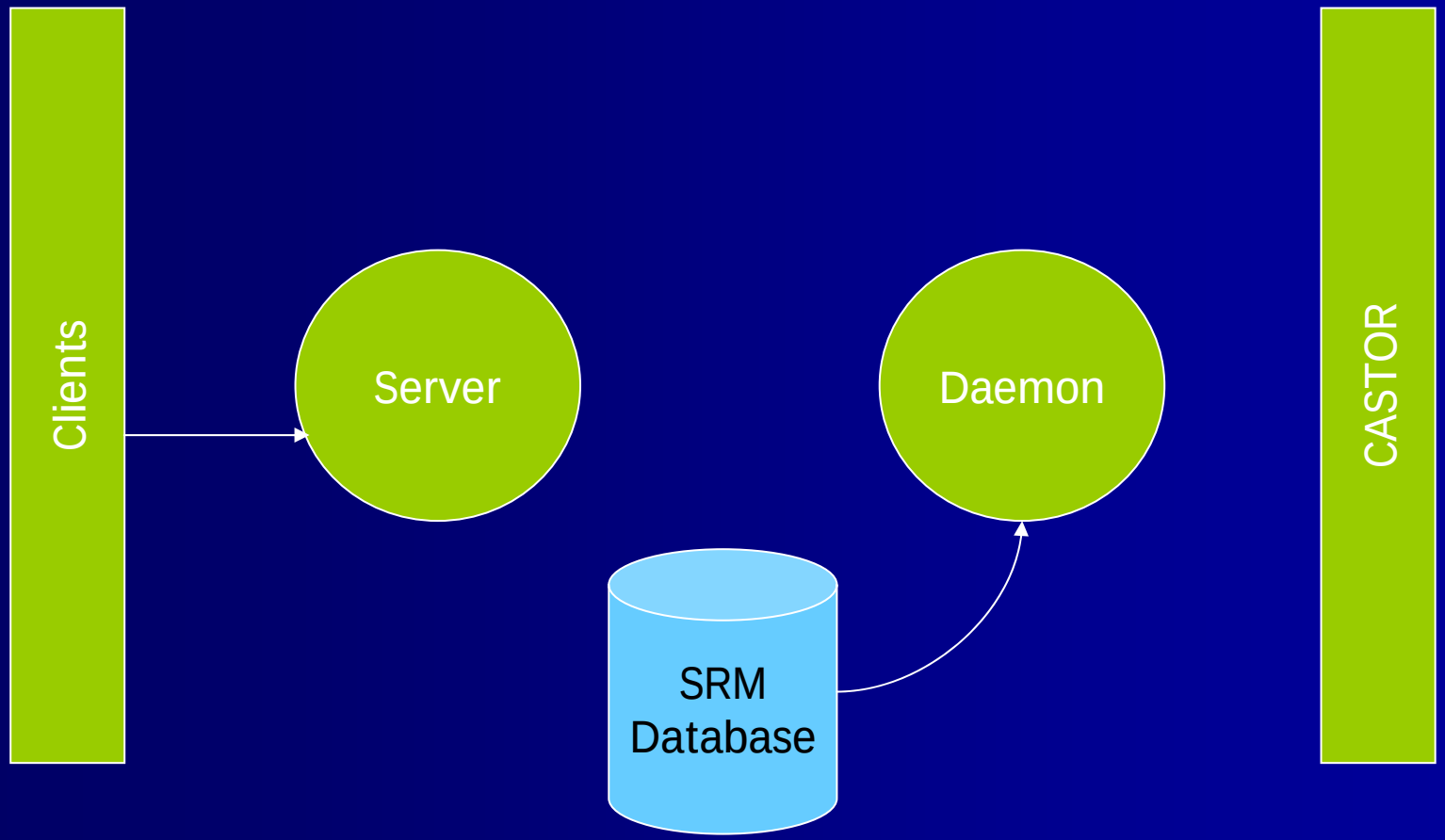
PUT operation (Prepare)



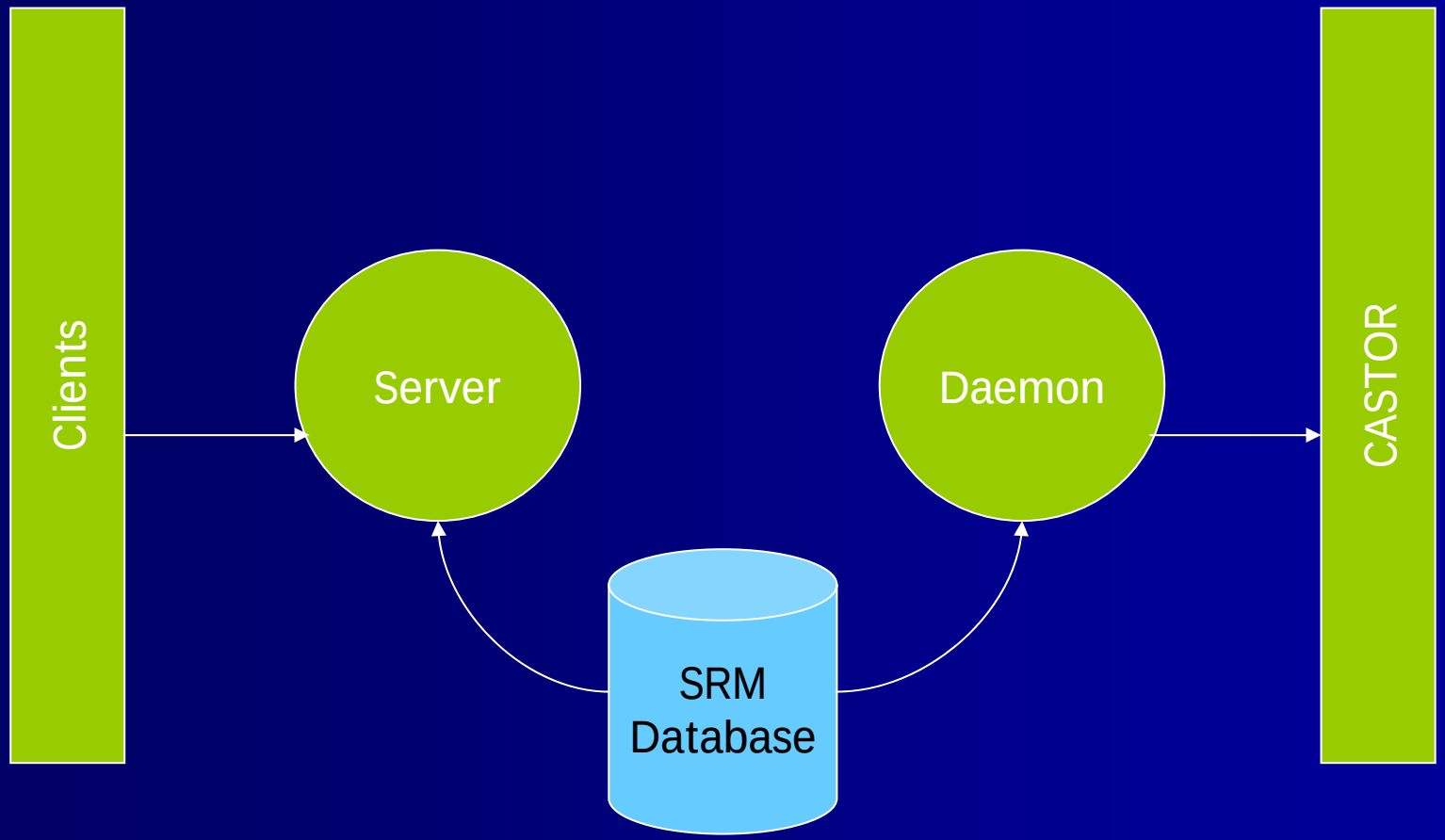
PUT operation (Prepare)



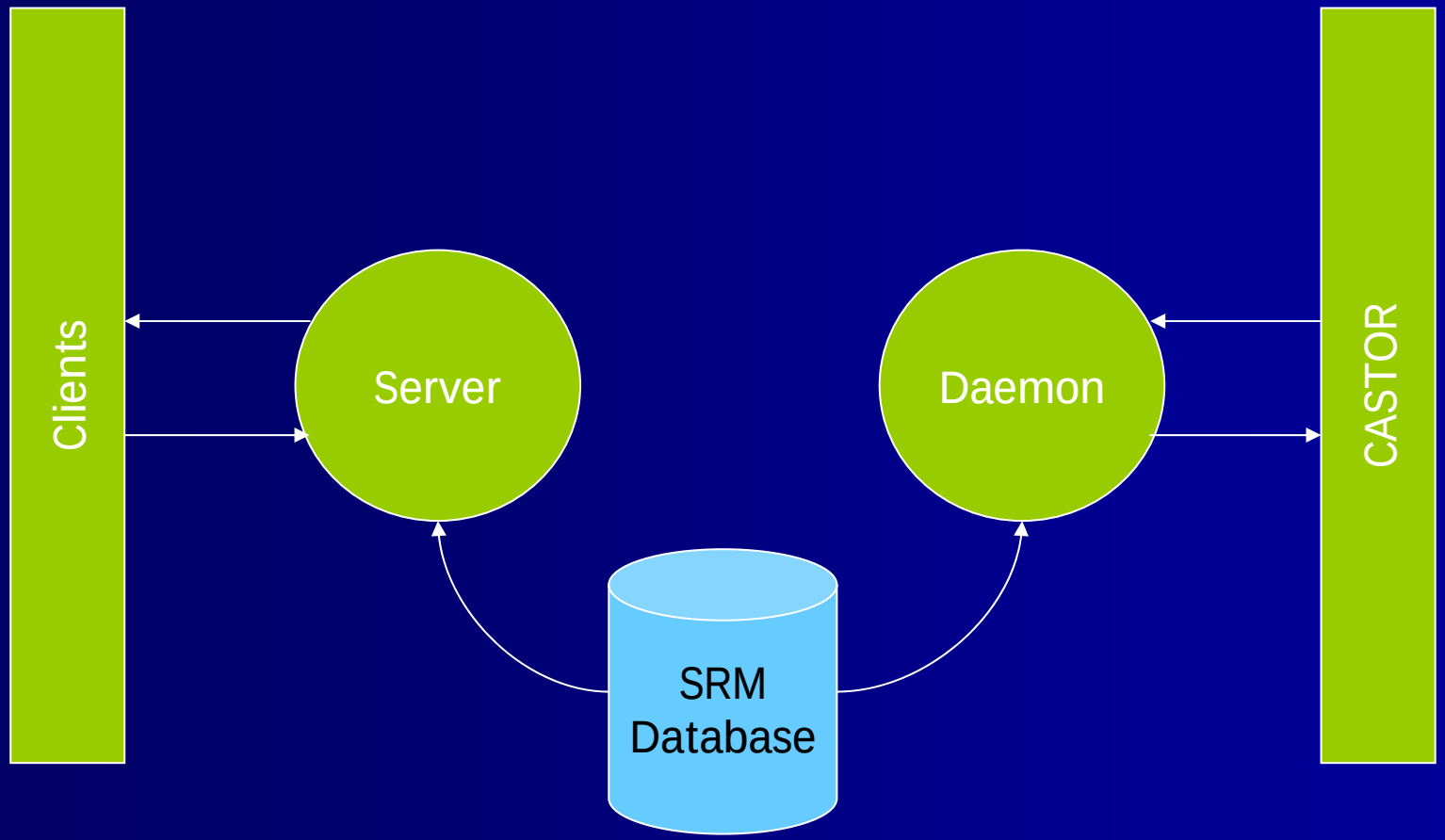
PUT operation (Status)



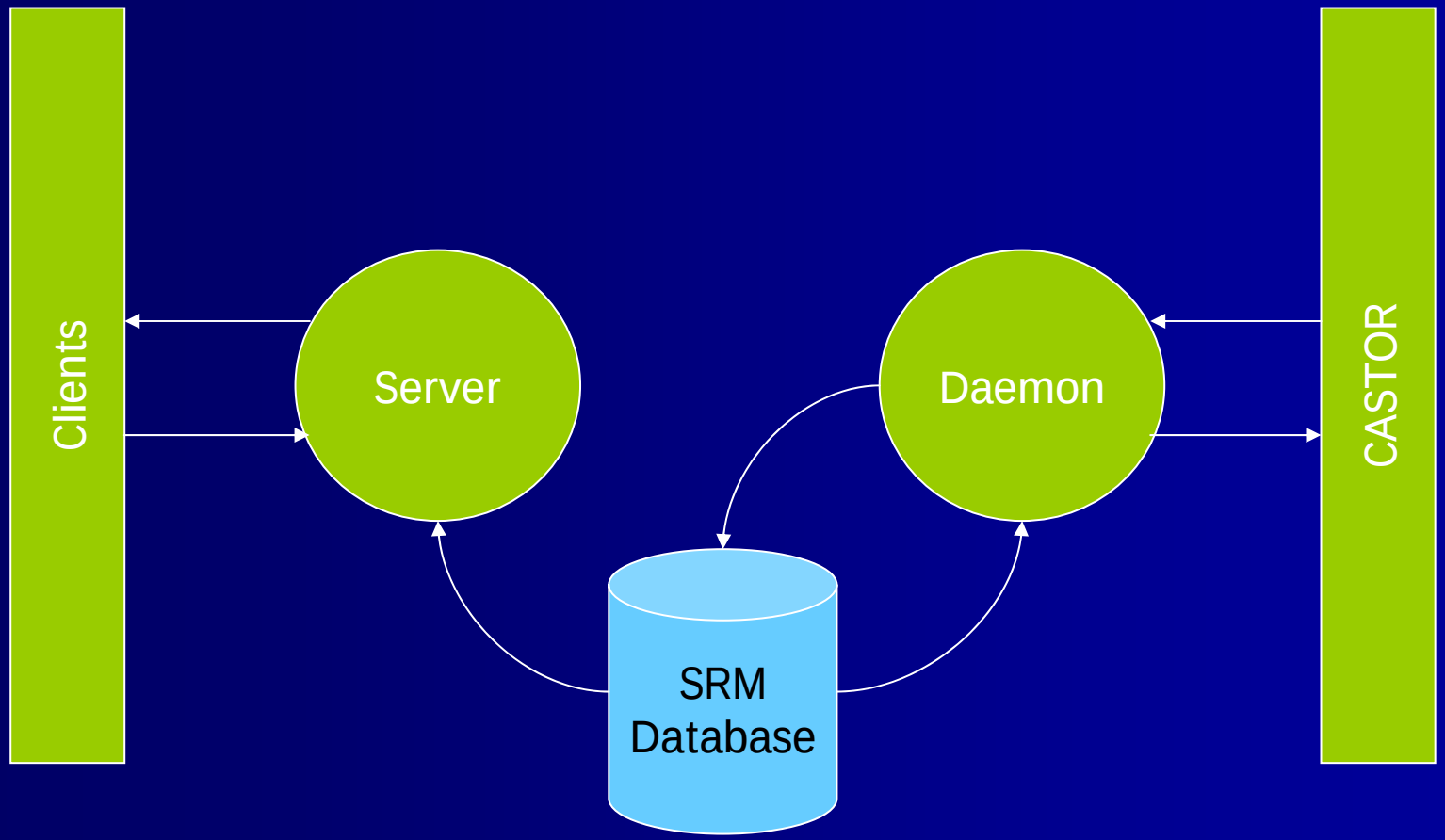
PUT operation (Status)



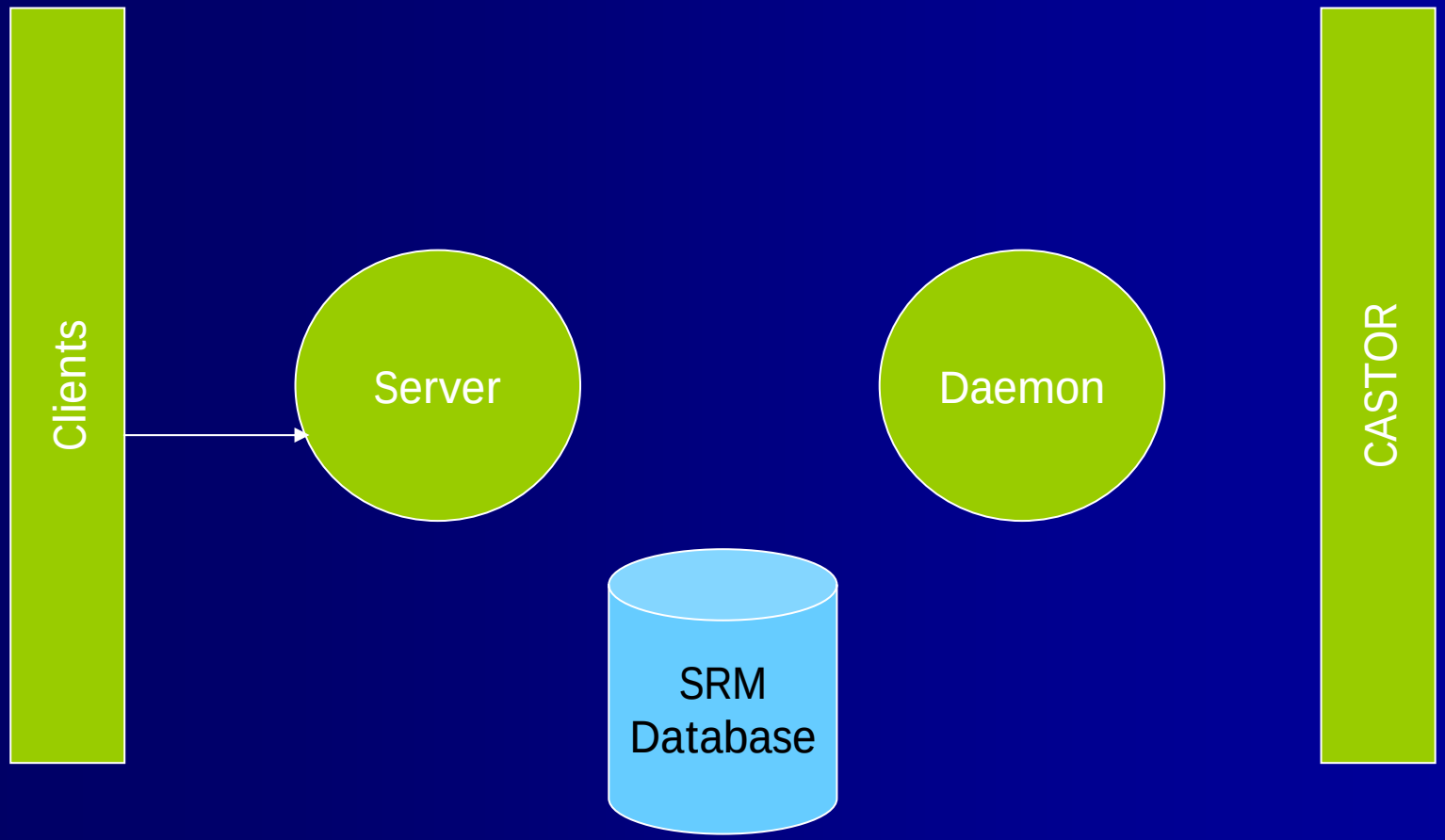
PUT operation (Status)



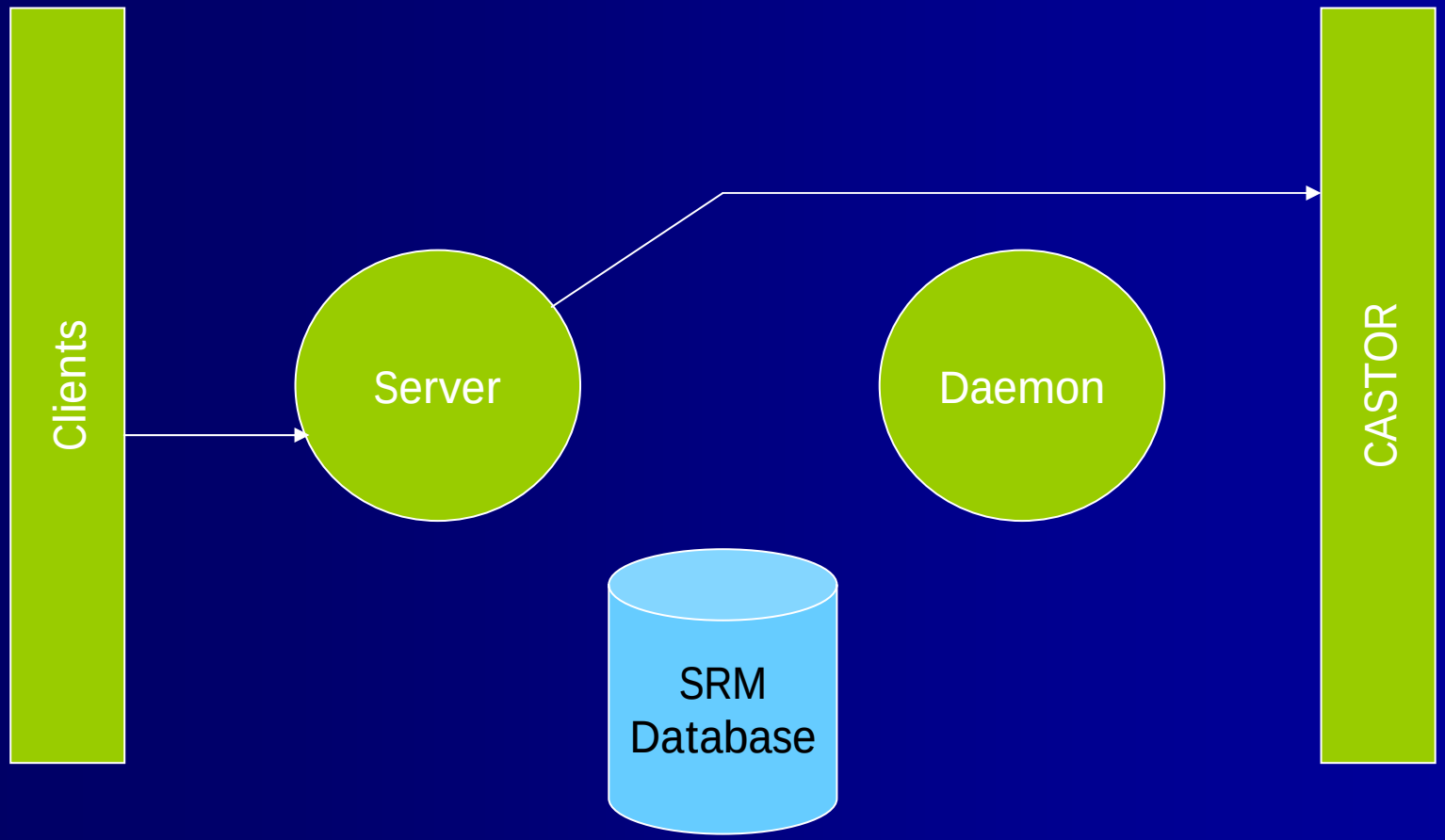
PUT operation (Status)



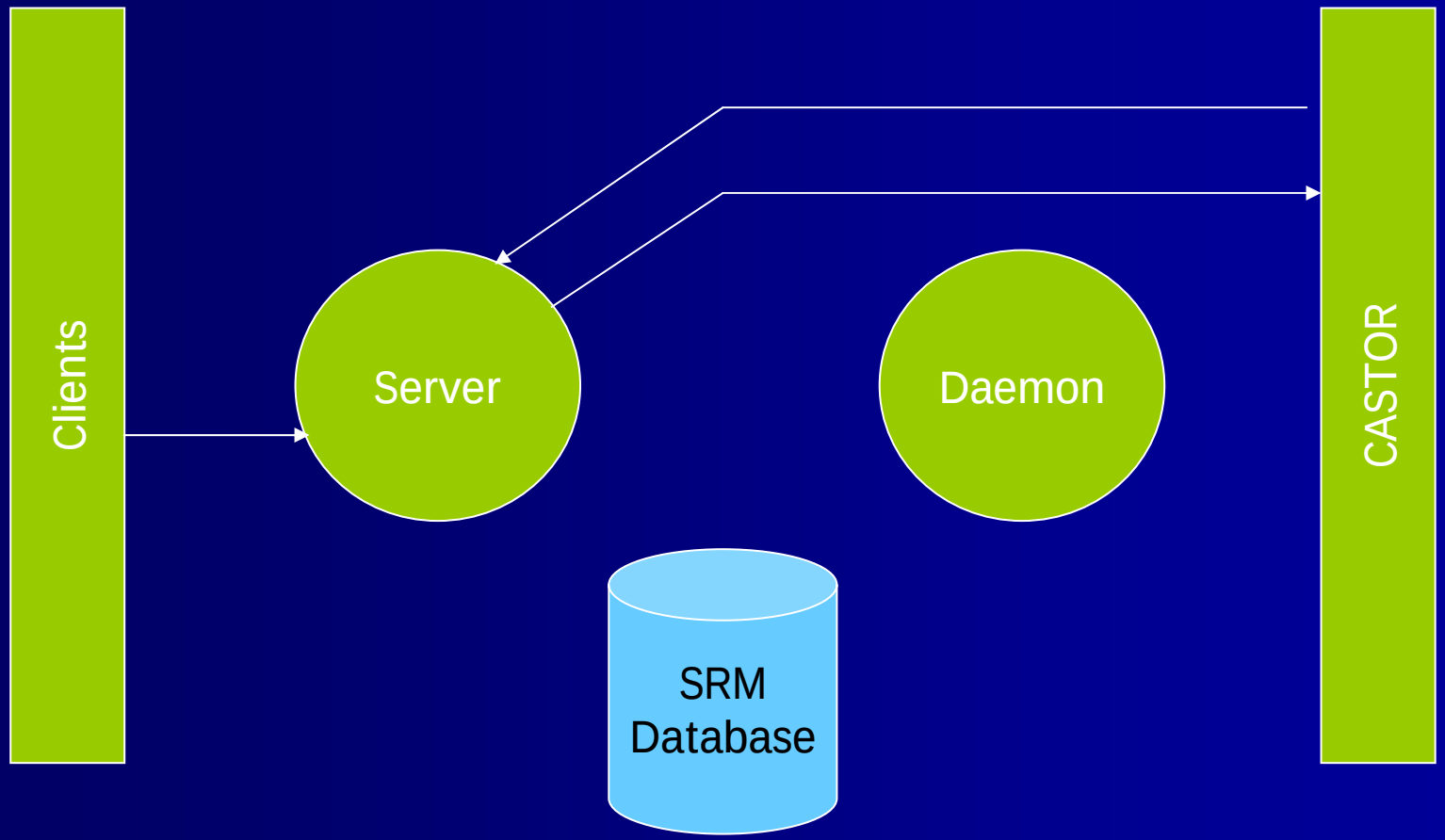
PUT operation (Done)



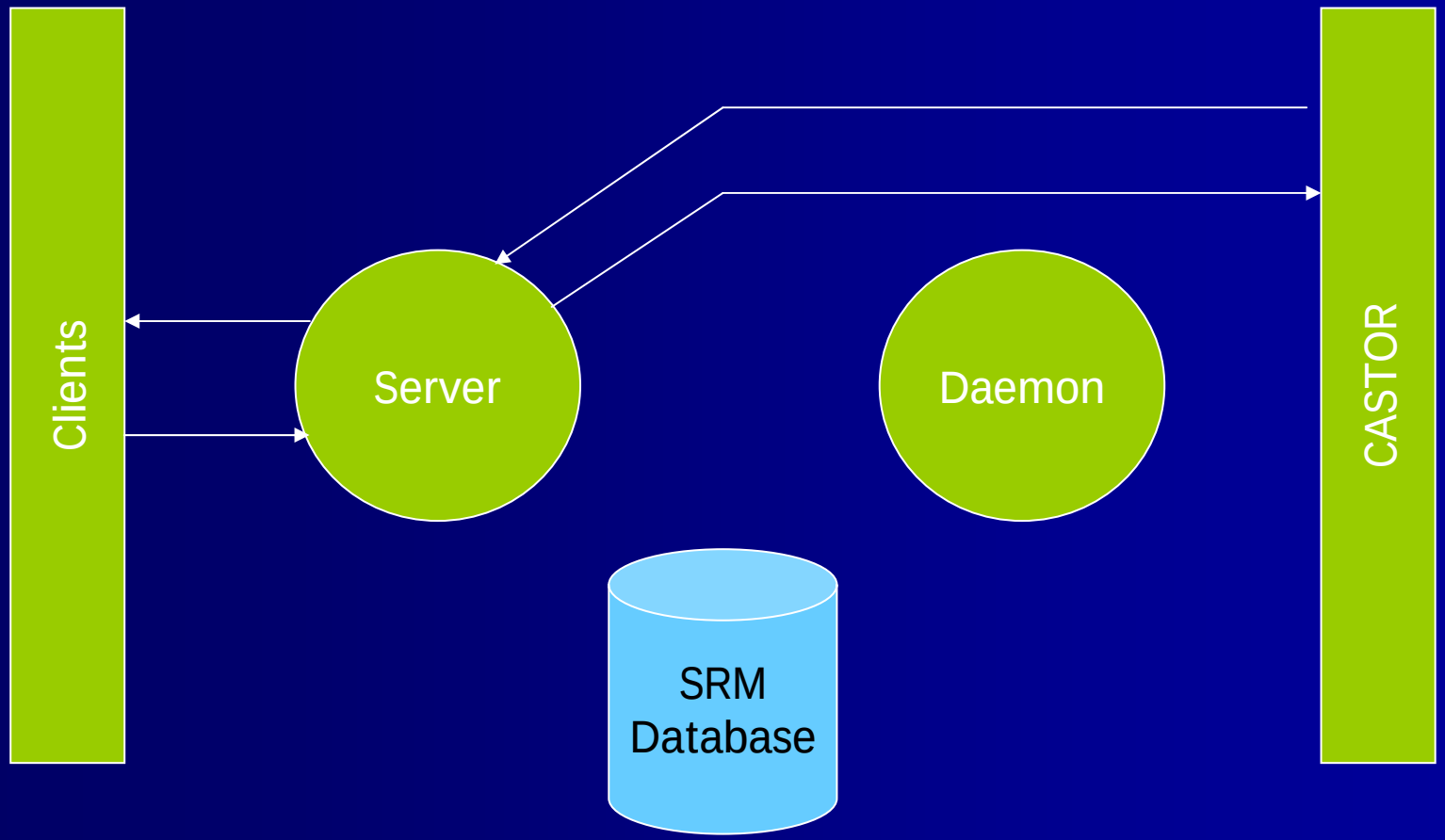
PUT operation (Done)



PUT operation (Done)



PUT operation (Done)



Dependencies

- Since everything done using RPMs should be no trouble working then out ☺
- CASTOR:
  - 2.1.1 or later
- CGSI-GSOAP plug-in:
  - 2.7-1.1.14 or later
- Globus Toolkit
  - vdt_globus_essentials-VDT1.2.2rh9_LCG-2
- VOMS
  - glite-security-voms-api-c-1.6.16
- Oracle
  - as per castor stager

Dependencies

- Server and Daemon in separate rpms
  - allows ease of deployment on separate nodes
- Both have same dependencies
  - srmCopy is daemon side operation that needs security libraries

Set-Up

- Two configuration files
  - srm2.conf
    - general configuration
  - srm2_storagemap.conf
    - Maps storage areas (space tokens) to service class
- Oracle
  - Script to create tables
  - Need to set up tnsnames, ORASTAGERCONFIG
  - Need to pre-populate storageArea table

SRM2.conf

- NOTIFYHOST
  - hostname where srmDaemon is running
- NOTIFYPORT
  - port numbers which daemon is listening on
    - one for GET/PUT
    - one for COPY
- PINTIME
  - Garbage Collection Weighting
  - Not implemented by default in CASTOR

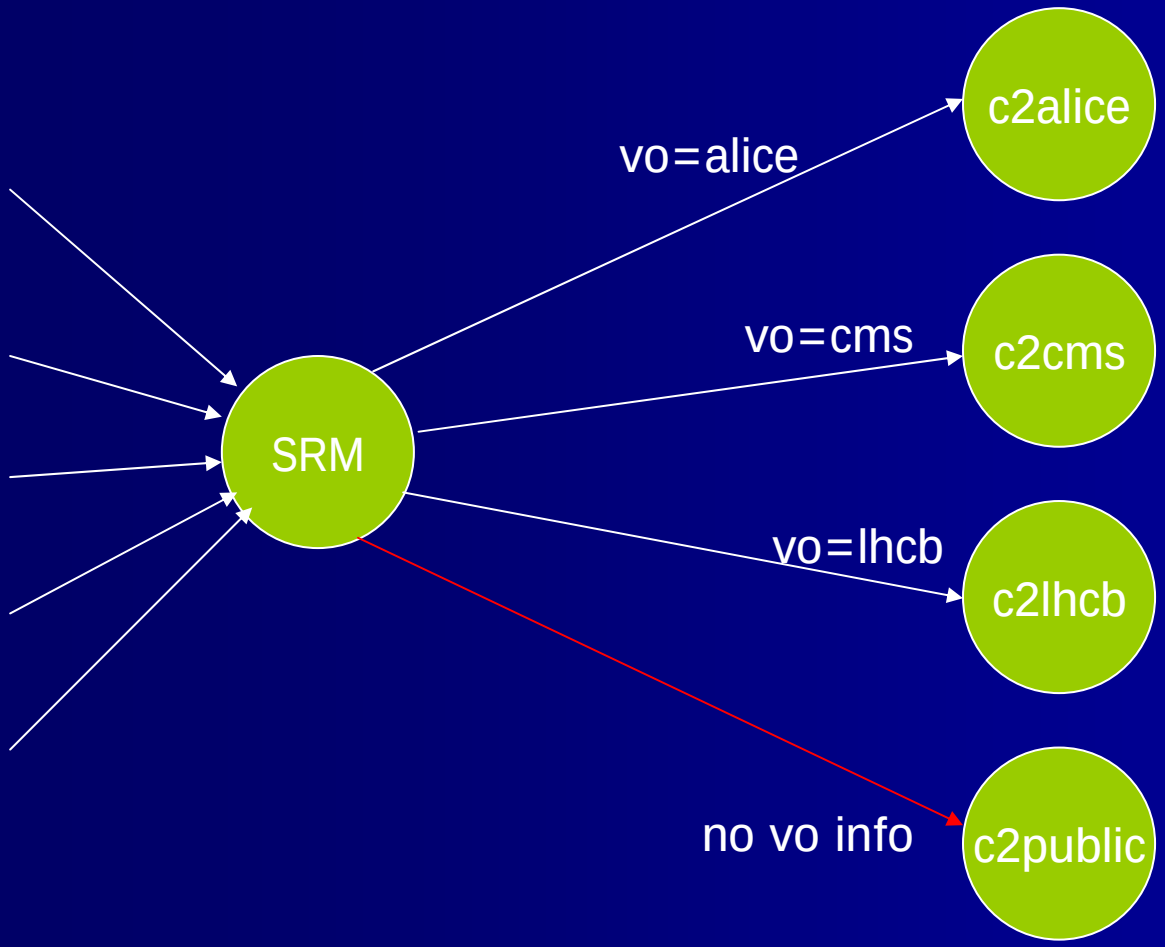
SRM2.conf

- MAXLSRETURNS/MAXLSCOUNT
  - Max # returns from srmLs, and max allowable offset
    - prevents too heavy load on srm
- SERVERTHREADS/LSTHREADS
  - SERVERTHREADS is max # of threads in srmServer
  - LSTHREADS is max # of Ls requests
    - avoids accidental DNS attacks

SRM2.conf

- Stager mapping
  - SRM <VO> <STAGE_HOST>
  - default:
    - SRM STAGE_HOST <blah>
    - For single castor instance sites

SRM2.conf


c2alice
vo=alice
vo=cms
c2cms
SRM
vo=lhcb
c2lhcb
no vo info
c2public

srm2_storagemap.conf

- Maps SRM retention policy/access latency to space token
  - hence to service class
- Format
  - <VO> TAPE*M*DISK*M* <service class>

srm2_storagemap.conf

```
ALICE       TAPE1DISK1      alimdc
ALICE       TAPE1DISK0      wan
ALICE       TAPE0DISK1      default
ALICE       default         default

DEFAULT     TAPE1DISK1      dteam
DEFAULT     TAPE1DISK0      dteam
DEFAULT     TAPE0DISK1      dteam
DEFAULT     default         dteam
```

srm2_storagemap.conf

- NOTE
  - Disk0 => Garbage Collection
  - Disk1 => No Garbage Collection
  - Tape1 => Migration
  - Tape0 => No Migration
- All should have default specified
  - Request with no space token or retentionPolicy info.

Oracle Setup

- Must initially set up service classes in CASTOR
  - Service class name reused
- Populate StorageArea Table
  - by hand
  - VO production manager run srmReserveSpace, then update entry showing data available
    - preferred since gets association between storage area and VO for free

Oracle Setup

- StorageArea table:
  - spaceToken VARCHAR2(2048)
  - tokenDescription VARCHAR2(2048)
  - storageLifetime INTEGER
  - svcClass VARCHAR2(2048)
  - storageType NUMBER
  - id INTEGER PRIMARY KEY
  - srmUser INTEGER
  - parent INTEGER
  - storageStatus INTEGER

Setup

- ALICE example
- Service Classes:
  - alimdc (Tape1Disk0)
  - recovery (Tape0Disk1)
  - wan (Tape1Disk0)
  - default (Tape1Disk0)
  - spare (Tape0Disk0)

srm2.conf

```
#ALICE          TAPE1DISK1      not supported
ALICE           TAPE1DISK0      alimdc
ALICE           TAPE0DISK1      recovery
ALICE           default         default
```

Requirements

- Suitable database and table space
  - Current at RAL and CERN SRM uses same database instance as stager
  - Unclear of load, may need its own oracle instance
- Suitable port open
  - 8443 is normally used

Differences

- Space Reservation
  - Spaces are static
    - No user defined spaces
  - Only suitable VOMS role allowed to reserve space
- Pinning
  - Pinning is advisory
  - Pinning represents GC weight

Space reservation

- Permitted user issues srmReserveSpace
  - Spacetoken assigned UUID, and set to space requested
    - UUID assigned initially since this help Tier-1 sites
  - Description can be meaningful
  - Alert sent to DLF
  - Manual update of space table once disks and pools are ready.

Problems

- Deployment problems
  - Related to cgsi plugin for gsoap
- Changes in Standard
  - Things missed and inconsistencies
- External pressure
  - Last three months – optimistically 25% of time

Support

- Supported by Giuseppe Lo Presti (CERN) and myself (RAL)
  - Details to be worked out
  - Informal e-mails should get reasonable response
    - but need to integrate into tracking s/w at CERN or RAL?
  - srm mailing list might be resurrected for support
- On-site support for deployment at other sites to be discussed
- Client tools being tested now.

Upcoming

- ChangeSpaceForFiles
  - Could be done wholly through SRM but easier internally to CASTOR
- UpdateSpace
  - Needs reimplementing similar to ReserveSpace
- Lots and lots of bug fixes
  - Finally getting lots of testers!
- Per VO threading
  - To try to stop 1 VO killing everyone
- 2 weeks intensive effort coming up (external influences permitting)