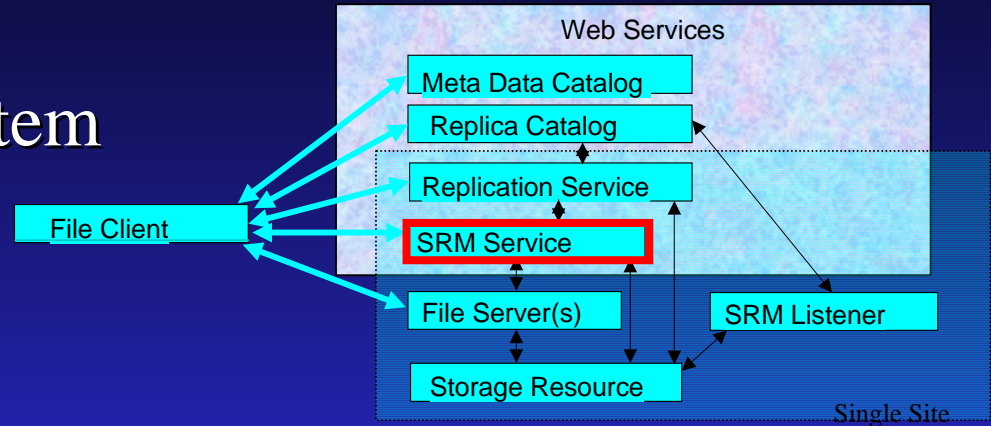# SRM Middleware Component

## ILDG-4 Workshop, May 2004

# Components: Storage Resource Manager

Manage storage system
- ◆ Disk only
- ◆ Disk plus tape

- ■ 3 party file transfers
- ■ Negotiate protocols for file retrieval (select a file server)
- ■ Auto stage a file on get (asynchronous operation)
- ■ Version 2.1 defined (collaboration)

Web Services

- Meta Data Catalog
- Replica Catalog
- Replication Service
- SRM Service
- File Server(s)
- SRM Listener
- Storage Resource
- File Client

Single Site

# SRM v2.1

## ILDG Essential Functions

With these functions, we could build a working grid. Even just implementing the first method could suffice.

- ◆ srmPrepareToGet

  Negotiates for retrival protocol, forces a stage to disk of offline files, returns URL

- ◆ srmRelease

  Let SRM know file is no longer needed

- ◆ srmCopy

  Move files from one SRM to another; pull/push (push uses srmPrepareToPut, srmPutDone)

# srmPrepareToGet

**srmPrepareToGet**

     In:      TUserID                   authorizationID,

        TGetFileRequest[]      arrayOfFileRequest,

           string[]                  arrayOfTransferProtocols,

           string                  userRequestDescription,

        TStorageSystemInfo    storageSystemInfo,

           TLifeTimeInSeconds  totalRetryTime


     Out:    TRequestToken       requestToken,

        TReturnStatus        returnStatus,

        TGetRequestFileStatus[]  arrayOfFileStatus

# Structures

```
typedef  struct {
     TSURLInfo                    fromSURLInfo,
     TLifeTimeInSeconds           lifetime, // pin time
TFileStorageType         fileStorageType,
TSpaceToken              spaceToken,
TDirOption               dirOption
     } TGetFileRequest
```

String to identify file

```
typedef  struct {
     TSURL            SURLOrStFN,
TStorageSystemInfo   storageSystemInfo
} TSURLInfo
```

# SRM v2.1

Categories of Functionality

- Put / get a file (negotiate for protocol)
- 3$^{rd}$ party file transfers (optional)
- Space reservation (for put and get/stage)
- Directory operations
- Manipulate file permissions

# SRM v2.1 conventions

- Protocol
  - ◆ SOAP, rpc style
- Authentication
  - ◆ Transport layer GSI (httpg)
- SURL (Site URL or SRM URL) conventions
  - ◆ SURL can contain or imply SOAP endpoint
  - ◆ SURL can imply location of the service's WSDL

# Implementation Status

- (Note: many sites have v1.1 implementations)
- Jefferson Lab has implemented v2.1 as front end to user files, daemon managed cache, and JLab's JasMine silo system (via abstract silo interface)
  - ◆ >90% of functions done
  - ◆ Working with GSI, strong authentication
  - ◆ Working on integrating with gridftp for srmCopy
  - ◆ Testing & debug for next several weeks
- Will provide as reference implementation to any ILDG members
- Will do interoperability testing when another SRM v2.1 implmentation is ready.