

HPSS Futures

Danny Teaff
June 18, 2002



HPSS Executive Committee

- ✓ Development objectives and deployment strategy for the next two HPSS releases will be discussed at the HPSS Executive Committee meeting on July 11-12 in Houston.
- ✓ Meeting announcement was sent to the HPSS EC reflector last month.
- ✓ HPSS EC members are encouraged to participate.
- ✓ Contact Dick Watson or Bob Coyne for details.

HPSS Executive Committee

- ✓ In addition to the planned DCE replacement effort, *candidate areas* to be discussed include:
 - Full Linux port
 - Small File performance enhancements
 - ADIC Scalar Tape support
 - Unattended monitoring
 - Automatic device add/delete
 - Scripted installs
 - Support for multiple Cluster File Systems
 - NFSV4
 - Multiple Distributed Movers and SAN exploitation
 - Grid FTP

General

- ✓ DCE replacement
 - **Planned for Release 5.2.**
 - **Subteam formed and actively working to address this item.**
 - **Communication channels established with product group in Austin.**
- ✓ Other content is to be defined.
 - **Some *candidate* areas are defined later in this presentation.**

DCE Replacement



✓ Current Usage:

– Threads

- Majority of HPSS processes are multi-threaded applications.
- 38 unique pthread APIs used (32 exist in POSIX library, 6 interfaces must be written).
- 1801 instances of pthread calls (87 require name changes only, 637 require syntax).

– Timing Services

- Needed for time synchronization of nodes hosting HPSS server components and security server. Time skew must be managed for the security server to function.
- No explicit application calls to timing services.
- NTP or other time synchronization service may be substituted.

DCE Replacement



– **UUID support**

- UUIDs used to identify HPSS objects (e.g. server ids, site ids, bitfile ids, storage segment ids, virtual volume ids, storage map ids).
- UUID generation from command line.
- APIs used to for the following: create a new UUID, create a NULL UUID, convert a UUID to string, convert a string to UUID format, return TRUE if 2 UUIDs are equal, return true if a UUID is NULL, compare 2 UUIDs, create a hash value for a UUID

– **Directory Services**

- Store and retrieve binding information so clients can find and bind to a server.
- Store security objects for each HPSS servers for authorization and authentication of server requests. ACLs on the security objects are used to define what principals / groups can access a server.

DCE Replacement



- **Remote Procedure Calls (RPCs)**
 - Primary method for control communication.
 - Key characteristics of usage:
 - Enable client to bind to a particular server.
 - Dispatch RPCs as separate threads.
 - Support for separate thread pools per interface.
 - Security for authentication.
 - 39 unique rpc_ calls used.
- **Security**
 - 49 unique sec_ / gss_ APIs used (255 instances).
 - Authentication of servers and clients.
 - GSS-API Support for handling Secure RPCs and to provide security to distributed applications that handle network communications (i.e. pftp/ftp).
 - Cross-cell authentication (federated namespace).

DCE Replacement



- Allow the client to set up an outbound security context.
- Verify incoming token and return the authentication source name and mechanism.
- Allow caller (client or server) to sign and encrypt a message.
- Allow caller (client or server) to decrypt and indicate if the message is signed.
- Allow context to be passed that identifies a client by uid, gid, and Realm ID. [local file system transfer]
- Determine client authorization access to HPSS server interface objects.

DCE Replacement - RPCs



Current options are:

- 1) Build a RPC replacement (described below).
- 2) Utilize solution from the IBM DCE team.

- ✓ Provide a simple communication infrastructure for HPSS to provide an HPSS replacement for DCE RPC
- ✓ Simple RPC runtime library
 - **Provide interfaces that look and behave much like the ones provided currently.**
 - **Model on existing functionality.**
 - Non-DCE Client Library and Gateway .
- ✓ Data definition IDL files are converted to XDR.

DCE Replacement - RPCs

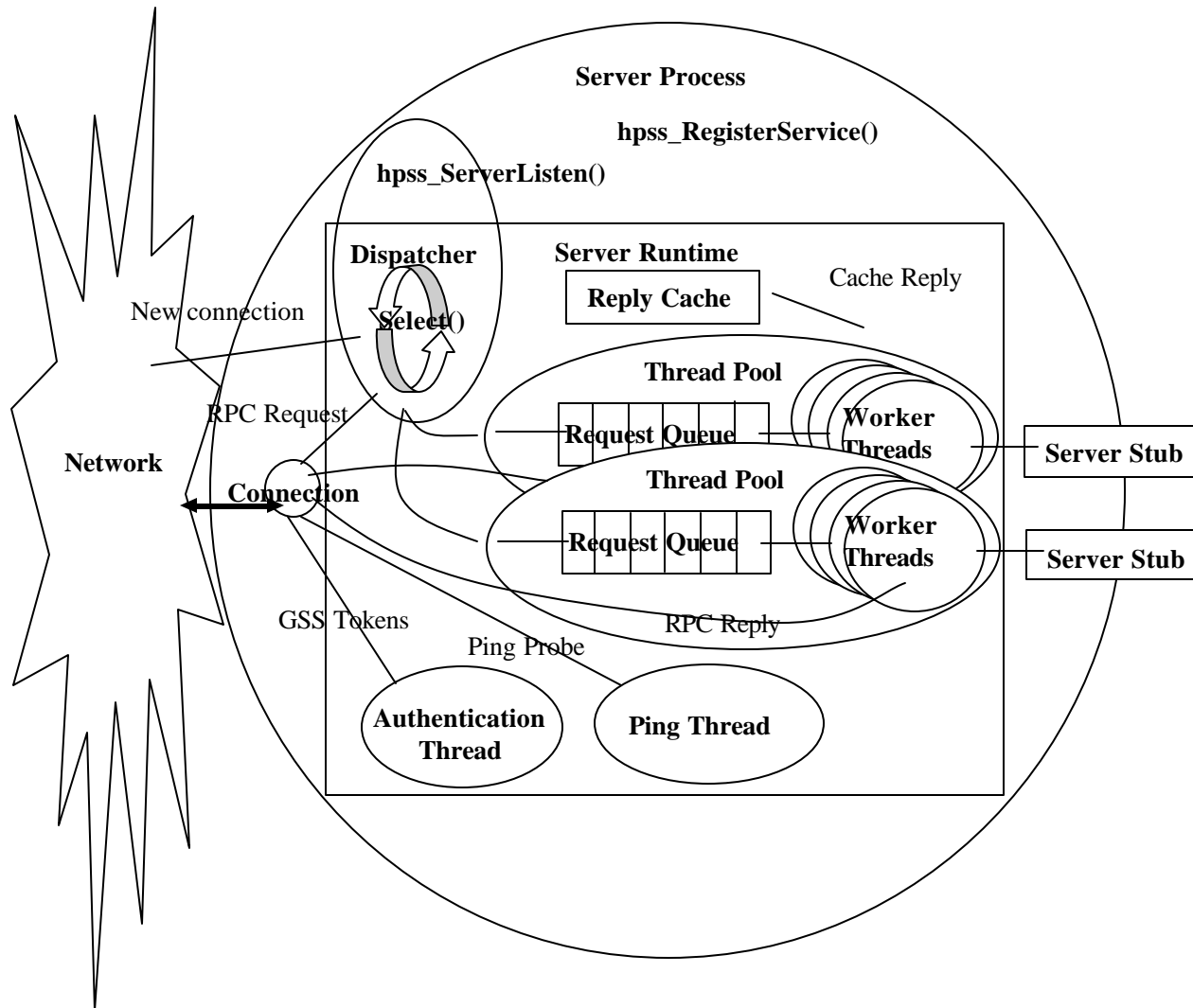


- ✓ Interface definition IDL files remain in IDL format.
- ✓ Simple IDL parser that generates client and server RPC stub code.
- ✓ GSS API compliant library to provide security services.
- ✓ Server Runtime Structure
 - **Simple runtime will be comprised of:**
 - Dispatcher thread
 - Request queue
 - Pool of RPC worker threads
 - Authentication thread
 - Reply cache
 - Ping Thread

DCE Replacement - RPCs



– Server Runtime Diagram



DCE Replacement - RPCs



✓ Server Interfaces

– **Standard RPC Runtime Interfaces.**

- Example: Register service, Listen for requests, etc...

– **Memory Management Interfaces.**

- Duplicate functionality provided by DCE RPC runtime.

Example: `rpc_ss_allocate_enable()`, `rpc_ss_allocate_enable`, `rpc_ss_allocate()`, etc...

- List of allocated memory per RPC.
- Pthread thread-specific data management interfaces.

✓ Client Runtime Structure

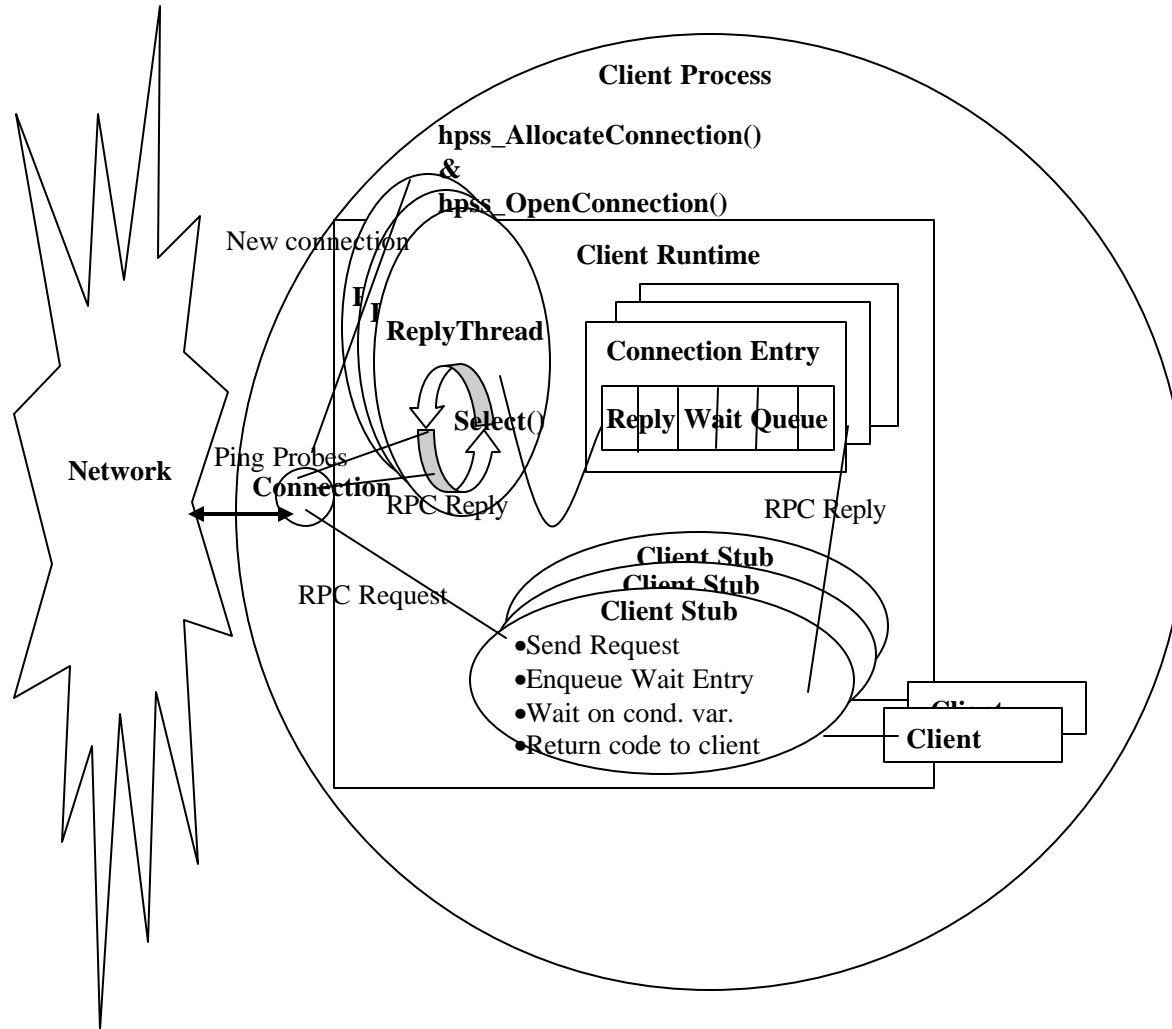
– **Simple runtime will be comprised of:**

- Reply thread per connection.
- Client threads that issue RPC requests.

DCE Replacement - RPCs



– Client Runtime Diagram



DCE Replacement - RPCs



- ✓ **Marshalling**
 - **IDL data structure definitions will be converted to XDR.**
 - Perl script used for Non-DCE function.
 - XDR files are used to generate the marshalling routines.
- ✓ **Client and Server Stubs**
 - **Generated by a simple IDL parser.**
 - Utilizing lex and yacc.
 - **Use core runtime communication primitives.**
 - Example: `hpss_SendRequest()` & `hpss_GetReply()` .
 - Example: `hpss_DecodeRPCArgs()` & `hpss_EncodeRPCArgs`.

DCE Replacement - RPCs



- ✓ Security
 - **GSS API library**
 - Wrapper or glue layer between HPSS and the GSS API implementation.
 - **GSS handshake to obtain network context.**
 - **Context used to provide integrity or privacy.**

DCE Replacement - RPCs



✓ Location Services

– Utilize RPC port map service.

- Servers are assigned unique RPC program number during configuration.
- Host name and RPC program number are stored in server configuration data.
- RPC runtime registers server with the port map service.

– HPSS Location Server.

- Returns IP address and RPC program number for a server
 - Specified by a UUID.
- Location servers assigned a well-know RPC program number.
- One Location server per physical host.

DCE Replacement - Threads

- ✓ Generate wrappers for POSIX supported thread calls.
 - Handle name and syntax changes.
 - Return -1 instead of errno.
- ✓ Generate code for non-supported POSIX thread calls:
 - **thread_delay_np, thread_get_expiration_np, pthread_getunique_np, thread_lock_global_np, pthread_mutexattr_setkind_np, pthread_unlock_global_np.**



DCE Replacement - Timing Services

- ✓ Use NTP (the Internet-standard Network Time Protocol) or other Timing Service of choice



DCE Replacement - UUID Support

- ✓ Re-implement these interfaces. Interface calls will remain syntactically the same as the DCE interfaces. This will avoid unnecessary changes to existing HPSS code.
- ✓ Maintain same UUID format (timestamp, hardcoded version, hardcoded variant, clock sequence, node).
- ✓ Also, may may be able pick up existing UUID software.



DCE Replacement - Directory Services

- ✓ Develop directory services based upon LDAP.
- ✓ Use IBM Directory Server.
- ✓ Store registry information.



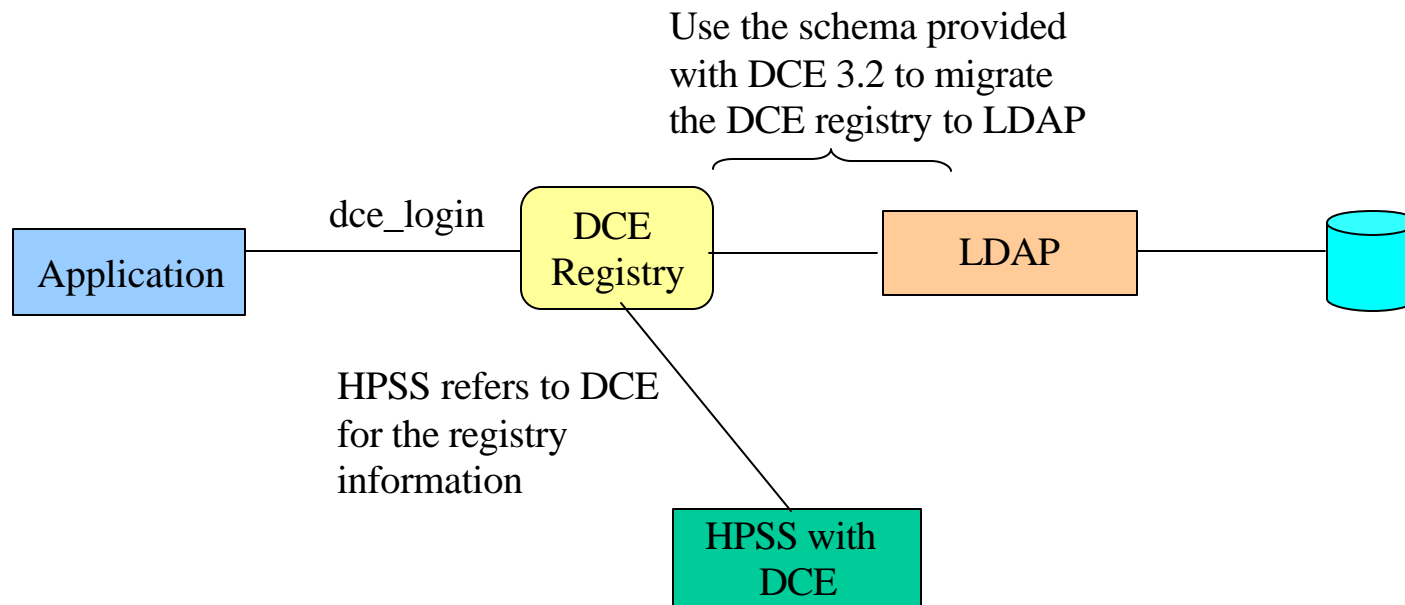
DCE Replacement - Security Services

- ✓ A combination of Kerberos and LDAP services will be used to support the Security Services.
- ✓ Kerberos will supply the authentication between service communication/authentication.
- ✓ Security information will be maintained in a DB2 table to support the required HPSS Server communication authentication (i.e Security /
- ✓ Kerberos will be used to authenticate the principal/password user information.
- ✓ Support cross-cell by using Kerberos cross-realm functionality.



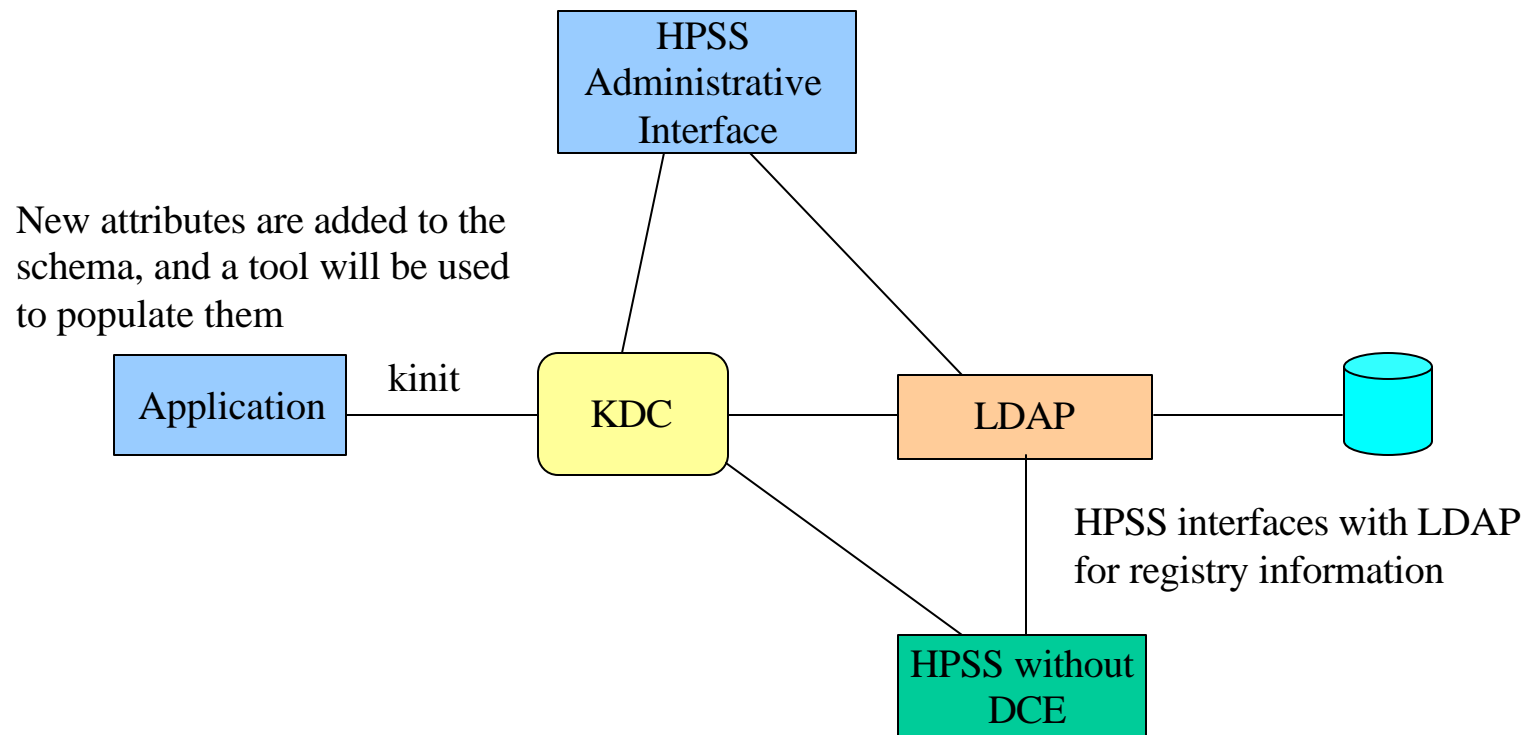
DCE Replacment - Migration Path Phase 1

HPSS 4.5/5.1 (using DCE 3.2)



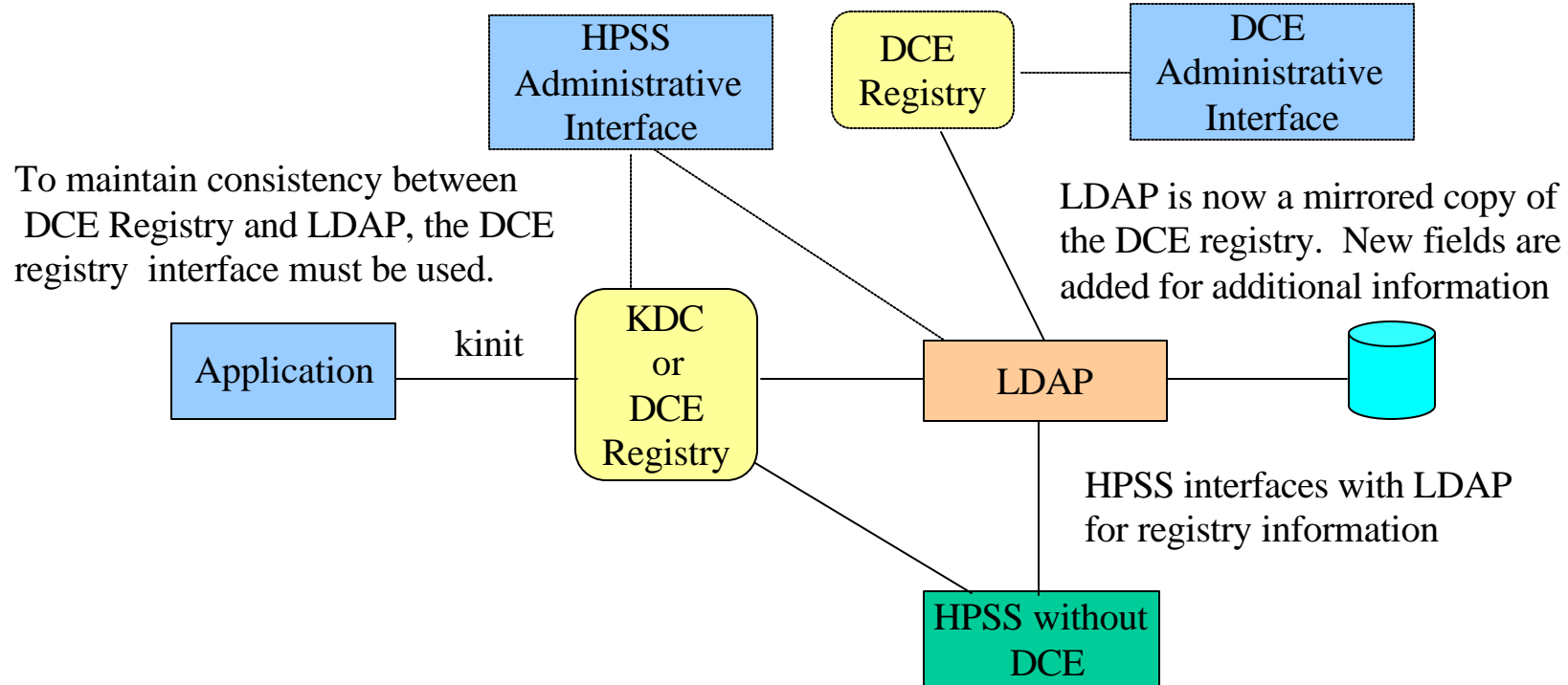
DCE Replacement - Migration Path Phase 2

Sites not requiring DCE



DCE Replacement - Migration Path Phase 2

Sites wanting to maintain their DCE Registry



Other Future Requirements Candidates



- ✓ SAN Exploitation - IRMs
 - **Stay tuned for SAN presentation.**
- ✓ Multiple Distributed Movers (MDMs)
 - **Manage each device with multiple Movers.**
 - **Select Movers based on client / device affinity.**
 - For each device, list of supporting Movers.
 - For each Mover, list of TCP/IP addresses.
 - Optimally select best Mover. If possible (client affinity), use shared memory for the transfer.
 - **Fail-over to alternate Mover following administrator action.**
 - **Load balancing based on ordering of lists of Movers supporting a device.**
 - If possible, select first Mover that matches client address.

Other Future Requirements Candidates



– LAN-less SAN Support

- Movers and potentially clients connected to the SAN.
- NFS Daemons / XFS Servers connected to the SAN.

– Server-less SAN Support (3rd Party Copy)

- Builds on Phase 1 changes
- SCSI-3 Extended Copy commands sent to SAN copy manager used to initiate device to device copies
- Movers convert offset / length into SCSI LBA / block count
- SAN id used to associate a disk or tape device with a SAN copy manager. Device selection based on device and copy manager association

Other Future Requirements Candidates



- ✓ Full Linux Port
 - Client API and Mover currently ported.
 - Port remaining server components.
- ✓ Dynamically Add Volumes
 - Support adding or modifying drives without requiring the PVL, PVR, and Mover to be recycled.
- ✓ New Device Support
 - Support new devices of interest to the HPSS user community when available.
- ✓ NFSV4
- ✓ Grid FTP
 - Utilize Grid FTP protocols and syntax for parallel FTP.

Other Future Requirements Candidates



- ✓ Migration Request Counts per Family
 - Request Count in Migration Policy applies to a Storage Class.
 - May result in files within a family being spread across multiple tapes.
 - Change would add new Request Count to apply to a family. This would limit the number of migration streams to the count supplied in this new count.
- ✓ DMAPI Support for GPFS
 - Port archived filesystem DMAPI code for GPFS support.
 - Support migration and staging of data in parallel across Movers.
 - Mirror file name and attributes in HPSS to allow read access from HPSS.