

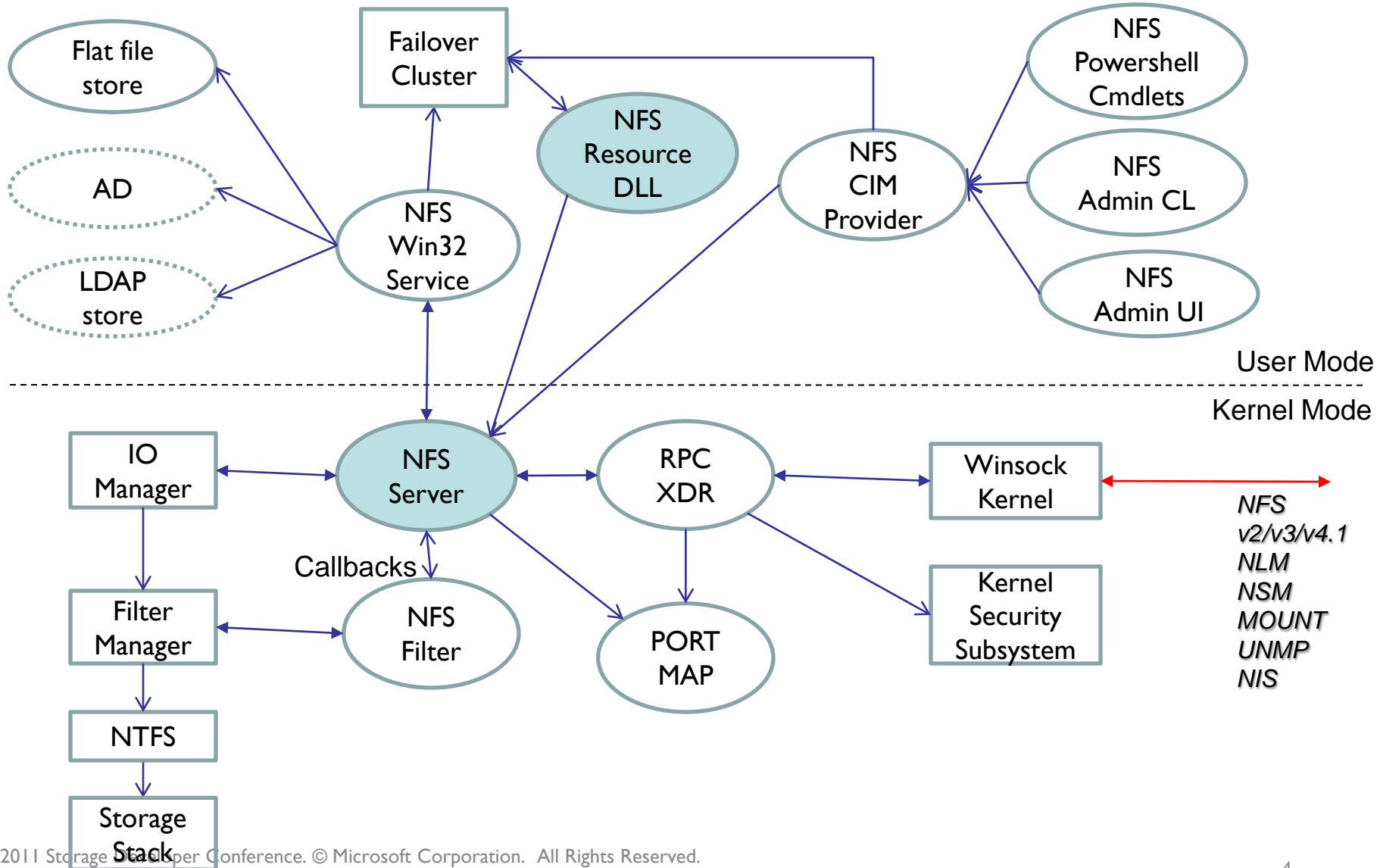
NFS High Availability in Windows

Roopesh Battepati
Microsoft Corporation

- ❑ Windows NFS Server Architecture Overview
- ❑ NFS Cluster Resource Architecture
 - ❑ Overview of Windows FC Resource Model
 - ❑ Windows NFS Resource DLL Overview
- ❑ NFS Virtual Server Architecture for HA
- ❑ NFS v4.1 Server progress

- ❑ $Availability = Uptime / (Downtime + Uptime)$
 - ❑ Decrease downtime for high availability
 - ❑ Faster failover = Higher availability
- ❑ How to make an application highly available in Windows Servers
 - ❑ Make application reboot-recoverable
 - ❑ Tell cluster about application
 - ❑ Create an application “resource” DLL
 - ❑ Cluster maintains configuration of resources
 - ❑ Monitor health of resources & moves resources if necessary

Windows NFS Server



NFS Server High Availability

Windows Failover Cluster Resource Groups



User mode

Kernel mode



Group of Virtual NFS Servers



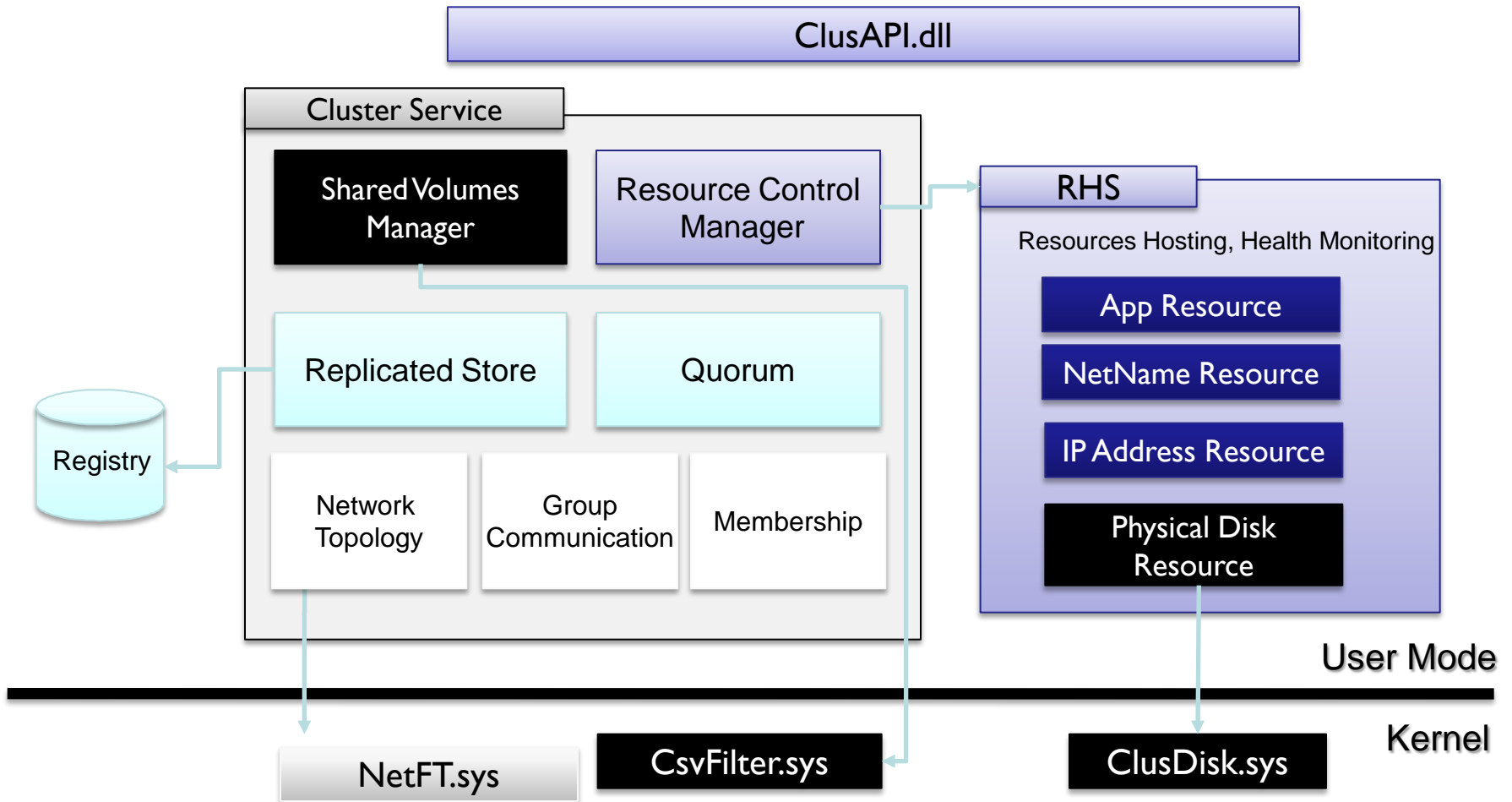
Network name

NFS Namespace

Cluster disk

Overview of Windows Failover Cluster Resource Model

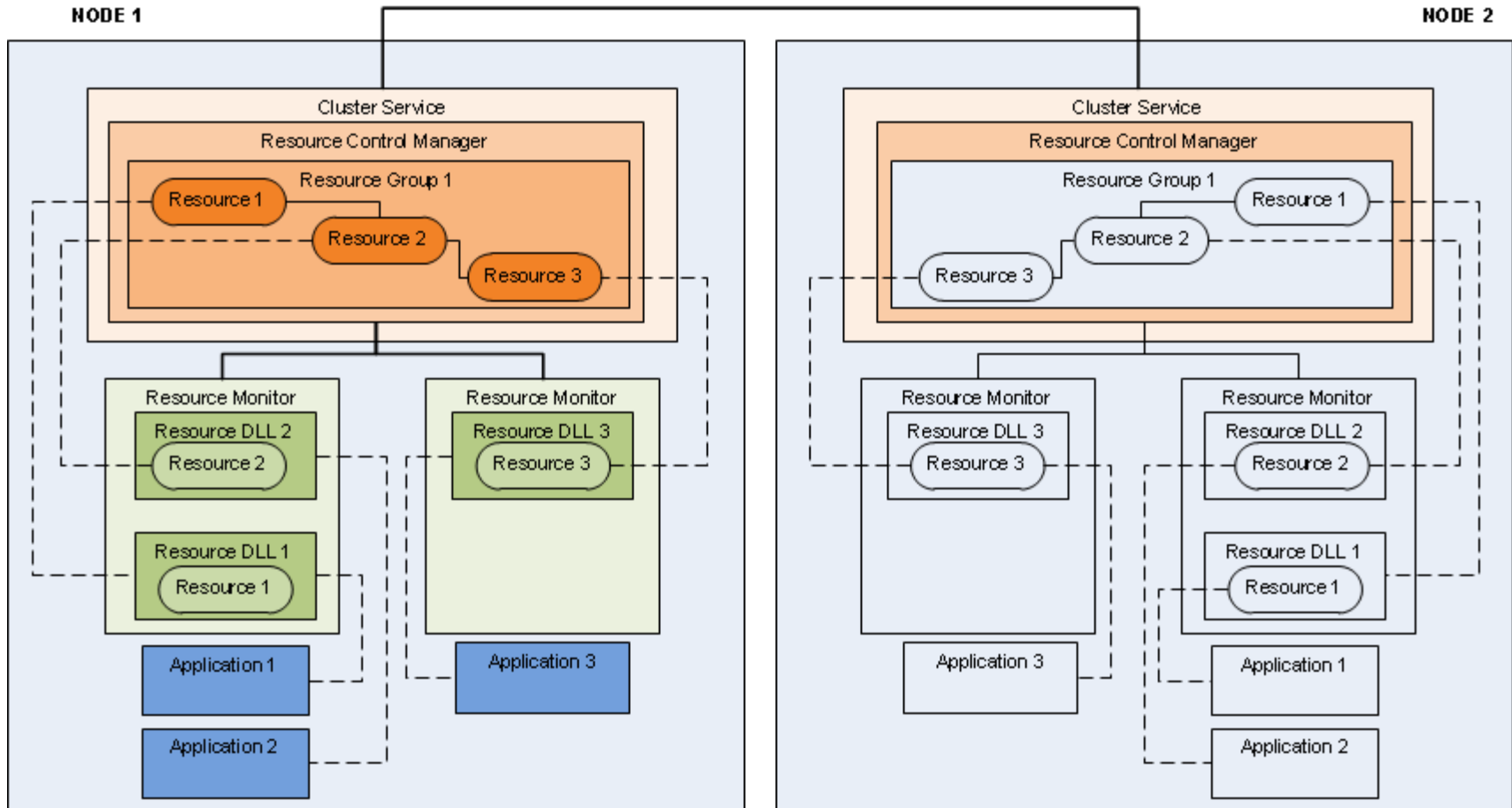
Windows Failover Cluster



- ❑ Represents an entity hosted by the cluster
 - ❑ Could be process, service, script, disk, IP address, file server, VM, etc.
- ❑ State machine
 - ❑ Online on at most one node at a time
- ❑ State transitions implemented in resource DLL
 - ❑ Implemented by application developer according to application semantics
 - ❑ Uses cluster APIs to interact with cluster, app-specific APIs to interact with the managed entity
 - ❑ Cluster invokes resource DLL via published

- ❑ Resources == Applications
- ❑ Resource code is untrusted code
- ❑ Isolated process per application
 - ❑ Optional multiple resources in same RHS
- ❑ Loose coupling
 - ❑ Created and maintained by RCM
 - ❑ Communication with RCM via RPC

Failover Cluster Resource Model



- Resource group is a collection of resources and their dependencies
 - AND (&) dependency
 - OR (||) dependency

- ❑ RCM maintains following 5 states for a resource:
 - ❑ Online
 - ❑ The resource is fully functional
 - ❑ Offline
 - ❑ The resource has been gracefully made non-functional
 - ❑ Failed
 - ❑ The resource has failed and is non-functional
 - ❑ Online Pending
 - ❑ The resource is in the process of coming online
 - ❑ Offline Pending
 - ❑ The resource is in the process of going offline

Resource Group States

- ❑ RCM maintains 4 states for a resource group:
 - ❑ Online
 - ❑ All resources in the group are online
 - ❑ Offline
 - ❑ All resources in the group are offline
 - ❑ Partial Online
 - ❑ Some resources in the group are in an online state and some are in an offline state but none of the resources are in a failed state
 - ❑ Failed
 - ❑ One or more resources in the group are in a failed state

RCM Actions on Resources

- Move
 - Causes a resource group to be hosted on another node in the cluster when initiated by an operator
- Failover
 - Causes a Resource Group to move to another node in the cluster due to a failure of a Resource (or multiple Resources) in the Resource Group
- Failback
 - Causes the Resource Group to move back to the original node in the cluster from which Failover was successfully performed, once the failure condition on original node has been remedied

- ❑ State transitions implemented in resource DLL
 - ❑ Cluster invokes resource DLL via published resource API
 - ❑ Open/Close: initialization/cleanup
 - ❑ Online: start the resource
 - ❑ Offline: stop the resource
 - ❑ Terminate: stop the resource immediately
 - ❑ IsAlive/LooksAlive: health check
 - ❑ Control: like an ioctl, extensible mechanism for configuration and app-specific operations

Startup & Function Table Exchange

```
DWORD WINAPI Startup(  
IN LPCWSTR  
IN DWORD  
IN DWORD  
IN PCLRES_CALLBACK_FUNCTION_TABLE  
CallbackFunctions,  
OUT PCLRES_FUNCTION_TABLE *ResDllFunctions);
```

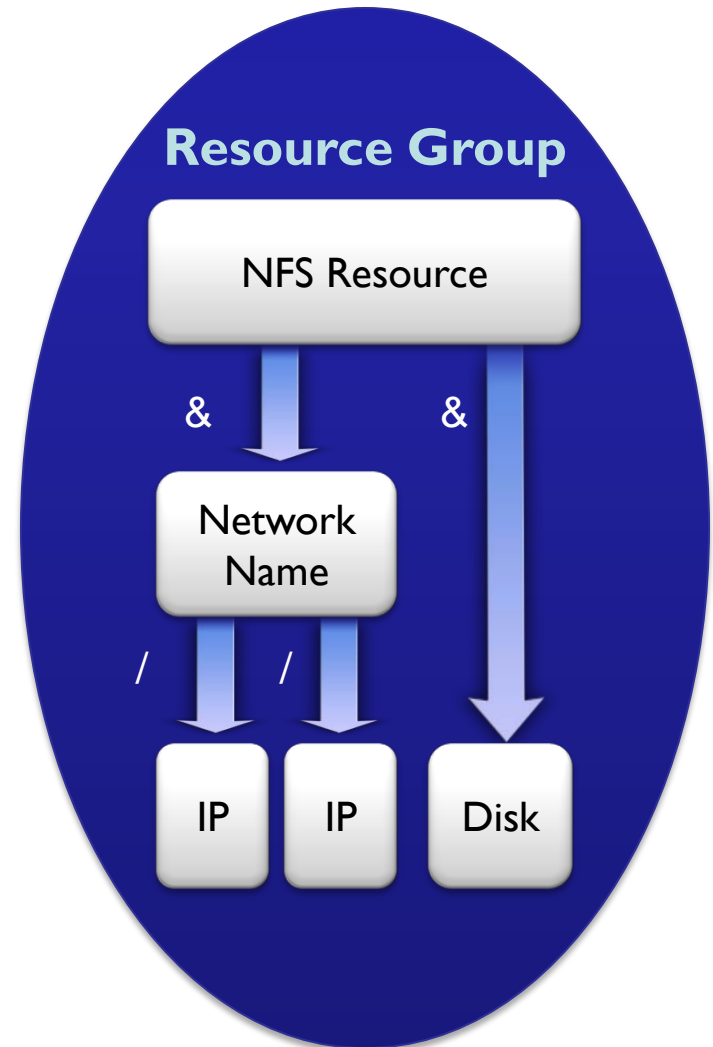
Startup & Function Table Exchange

```
□ CLRES_FUNCTION_TABLE {  
  POPEN_ROUTINE           Open;  
  PCLOSE_ROUTINE          Close;  
  PONLINE_ROUTINE         Online;  
  POFFLINE_ROUTINE        Offline;  
  PTERMINATE_ROUTINE      Terminate;  
  PLOOKS_ALIVE_ROUTINE    LooksAlive;  
  PIS_ALIVE_ROUTINE       IsAlive;  
  PRESOURCE_CONTROL_ROUTINE ResourceControl;  
  ...  
};
```

Windows NFS Resource DLL Overview

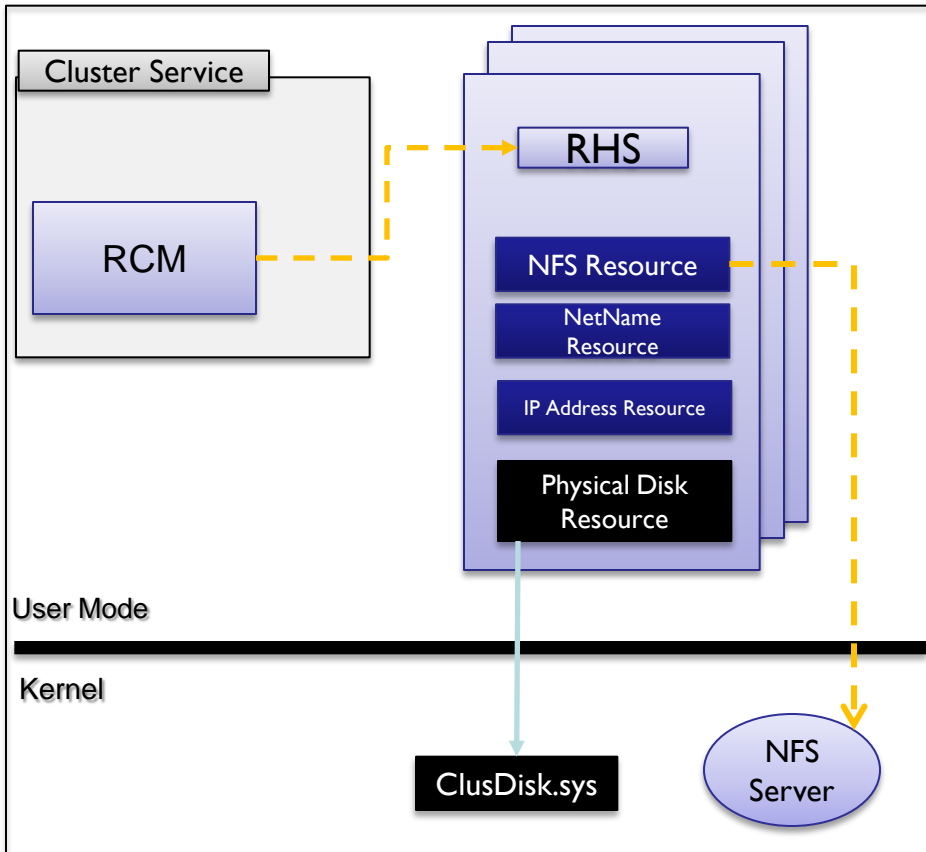
NFS Resource Group

- ❑ Container for NFS application resources
- ❑ Unit of NFS application failover
- ❑ Boundary of resource dependencies
 - ❑ Start/stop order
 - ❑ App-specific relationships
 - ❑ AND/OR

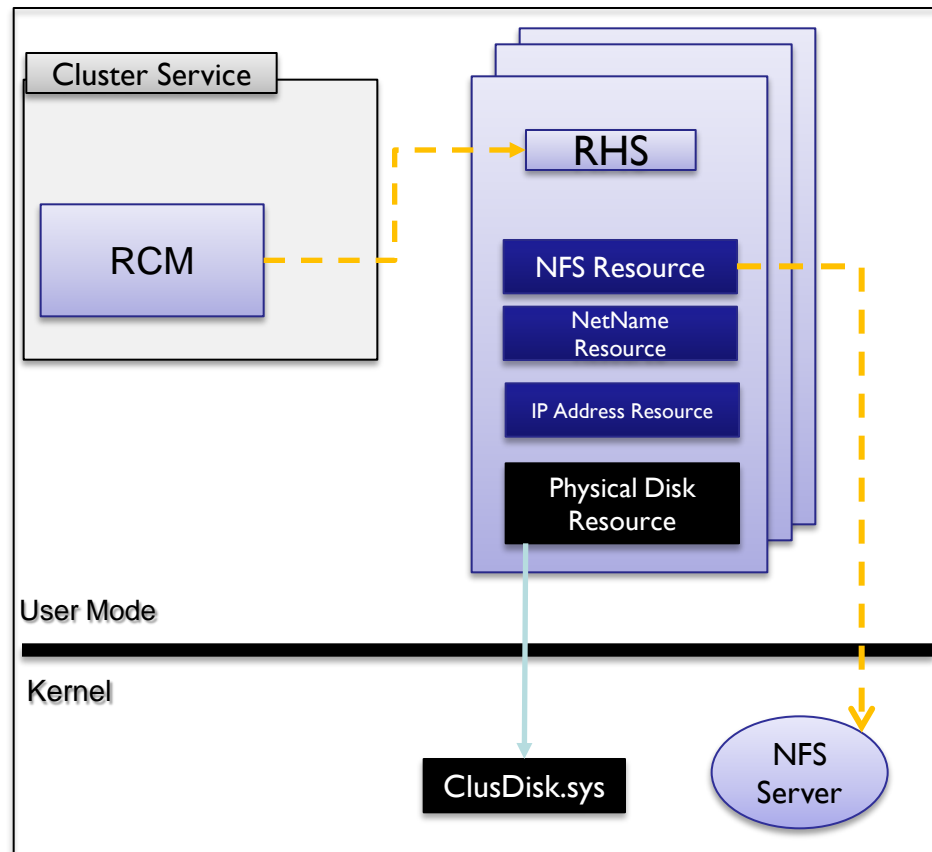


NFS Resources in Failover Cluster

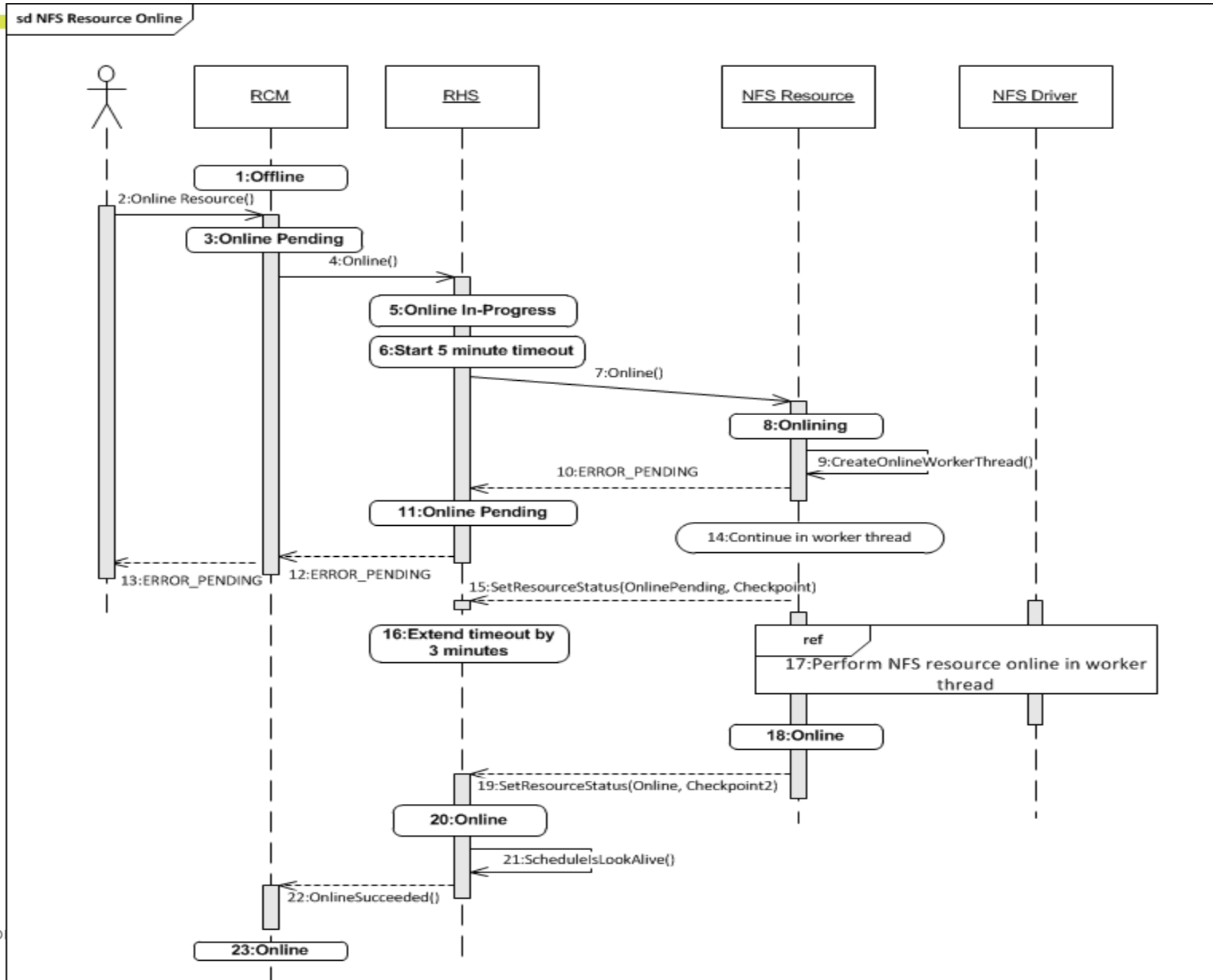
Failover Cluster Node 1



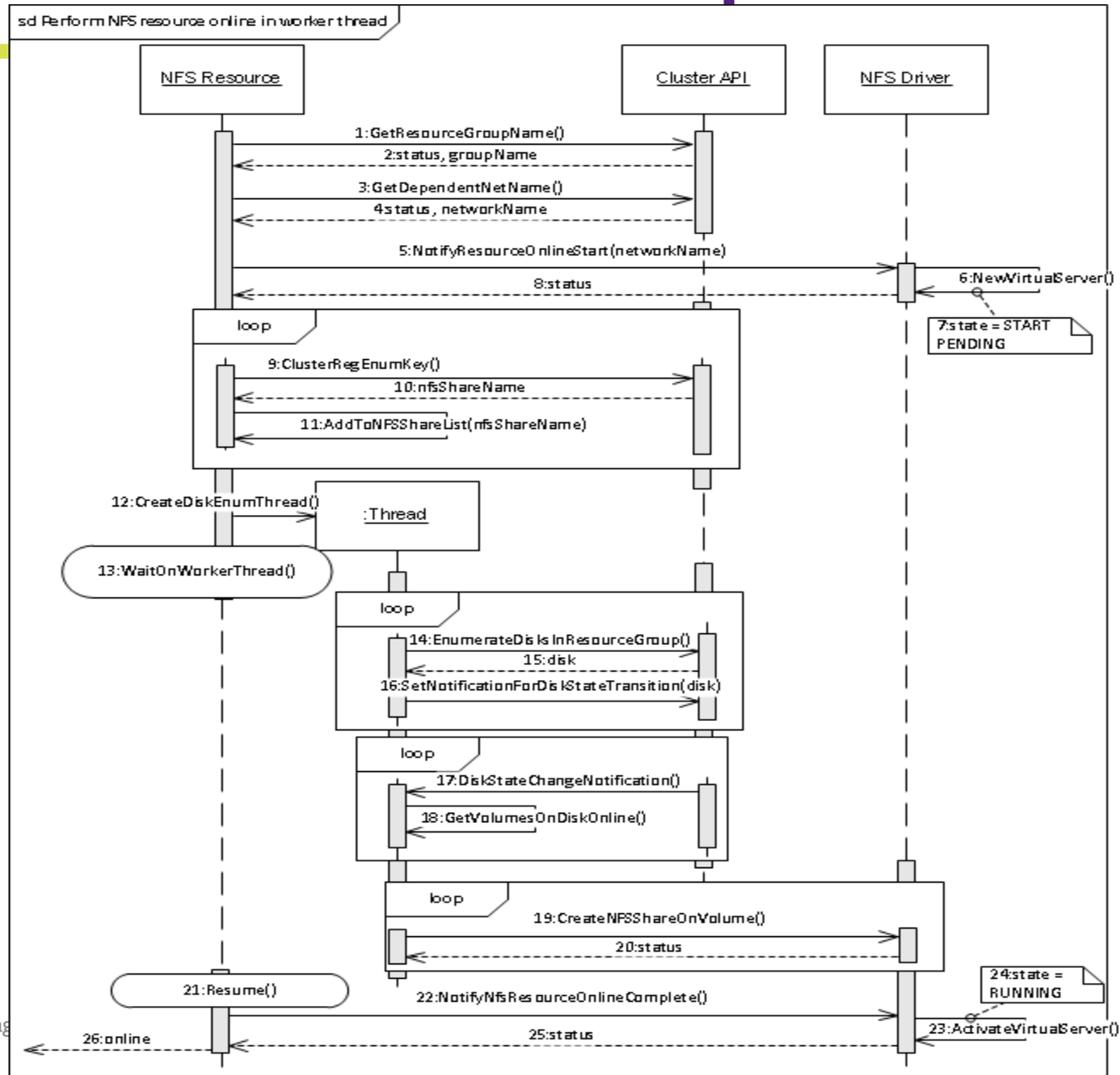
Failover Cluster Node 2



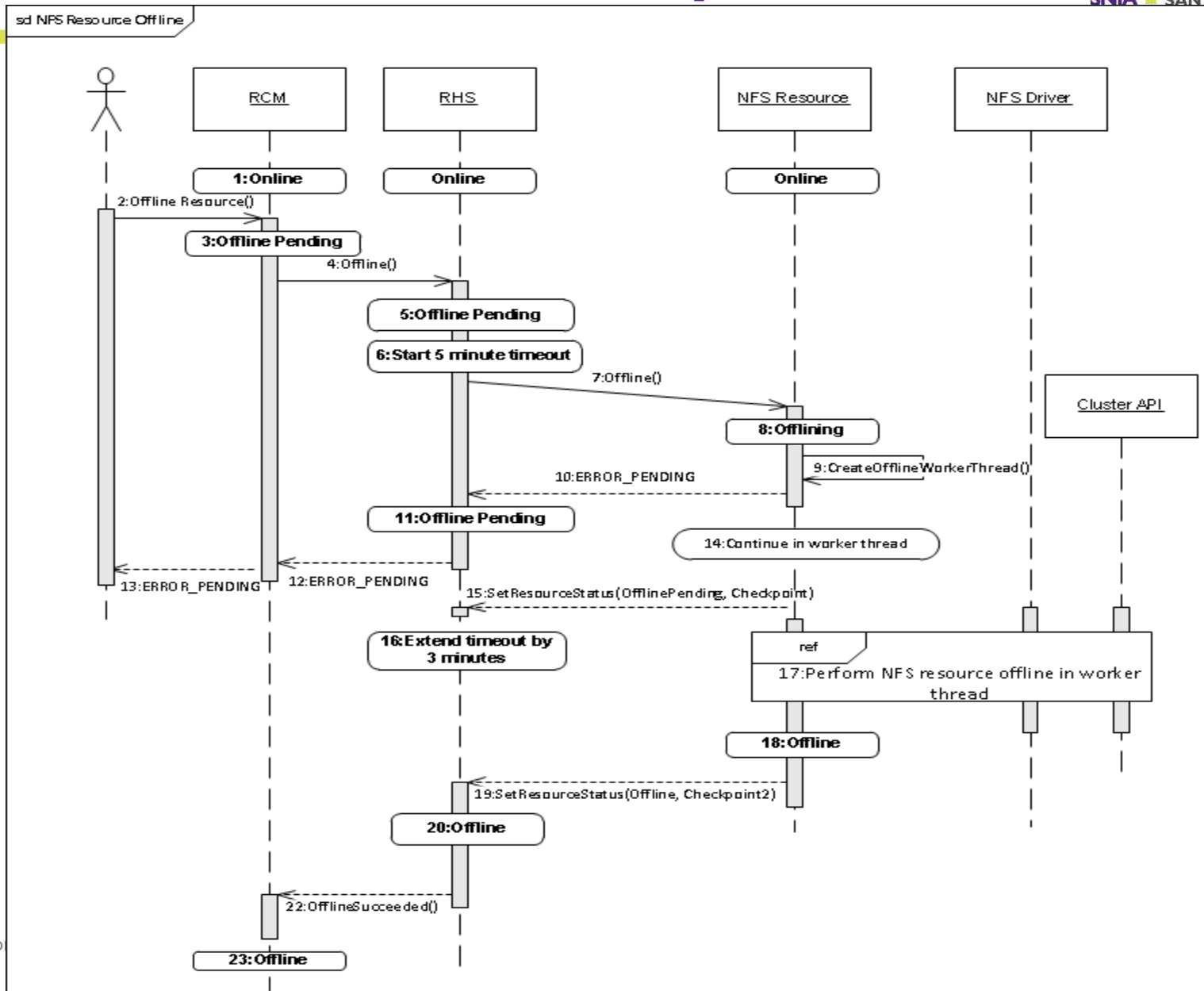
NFS Resource Online Sequence



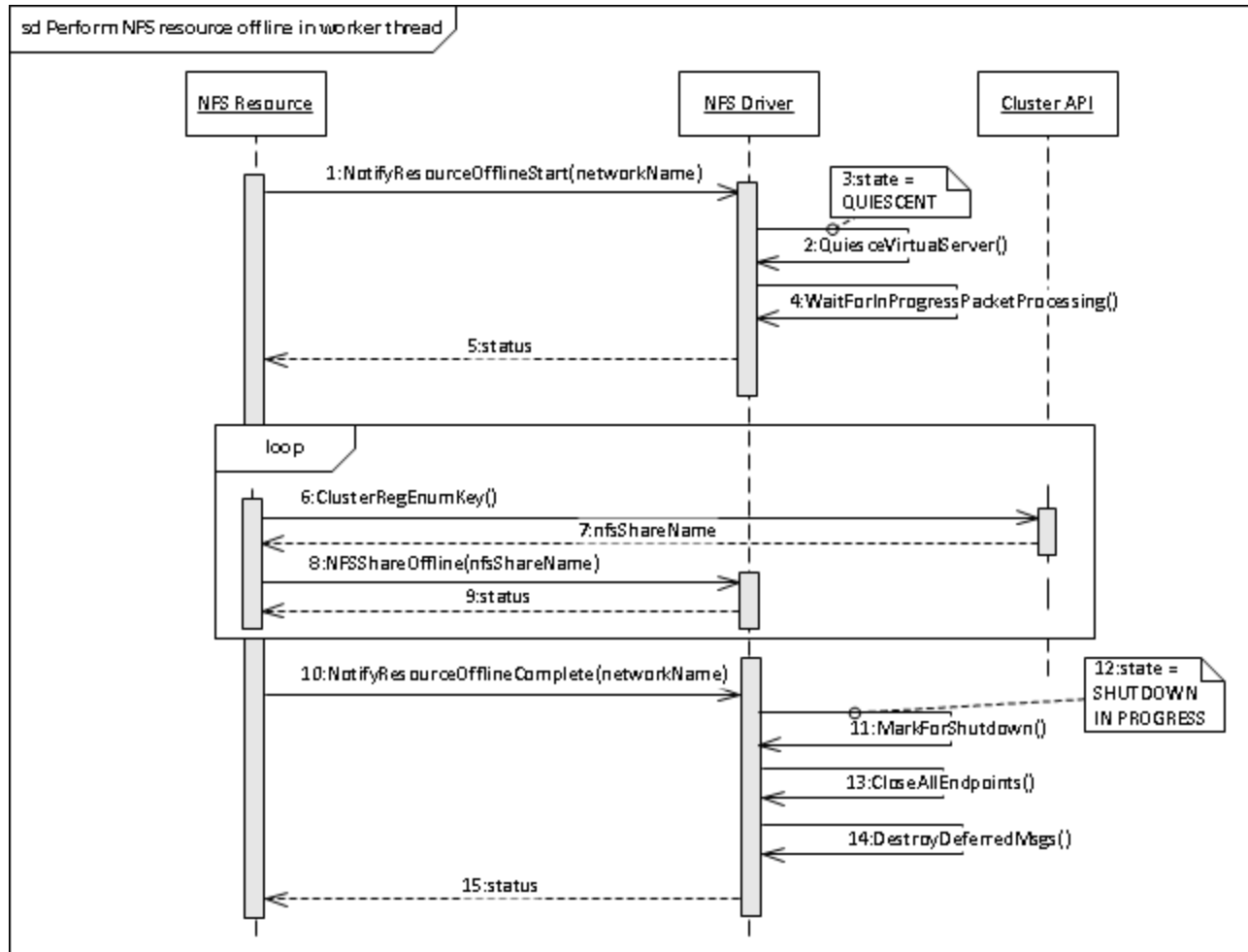
NFS Resource Online Sequence



NFS Resource Offline Sequence



NFS Resource Offline Sequence



Windows NFS Server Architecture for High Availability

- ❑ Container for NFS Driver “virtual” server instances
 - ❑ Independent namespaces
 - ❑ Independent network endpoints
 - ❑ IP address scoping for namespace
- ❑ Singleton VirtualServer0 for non-clustered resources
- ❑ As many additional virtual servers as there are FC NFS resources

NFS Virtual Server Manager

```
LIST_ENTRY DeferredEndpointList
LIST_ENTRY AllEndpoints
LIST_ENTRY VirtualServerList
NFS_VIRTUAL_SERVER VirtualServer0
ULONG      VirtualServer0Incarnation
PNP_CONTEXT *PnpContext
```

NFS Server – Virtual Server

- ❑ Unit of NFS Server failover
- ❑ Virtual Server Contains:
 - ❑ Collection of TCP endpoints (one per IP Address)
 - ❑ Collection of NFS Shares (namespace)
 - ❑ Referenced volumes backing the NFS Shares

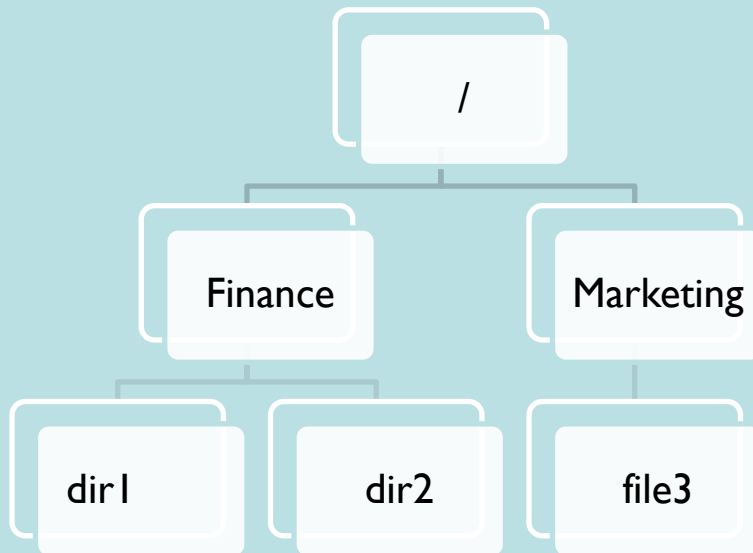
NFS_VIRTUAL_SERVER

```
GUID      Id
ULONG     Incarnation
UNICODE_STRING VsName
NFS_VS_STATE VsState
LIST_ENTRY DelayedMsgList
LIST_ENTRY EndpointList
NFS_SHARE_CTX NfsNamespace
```

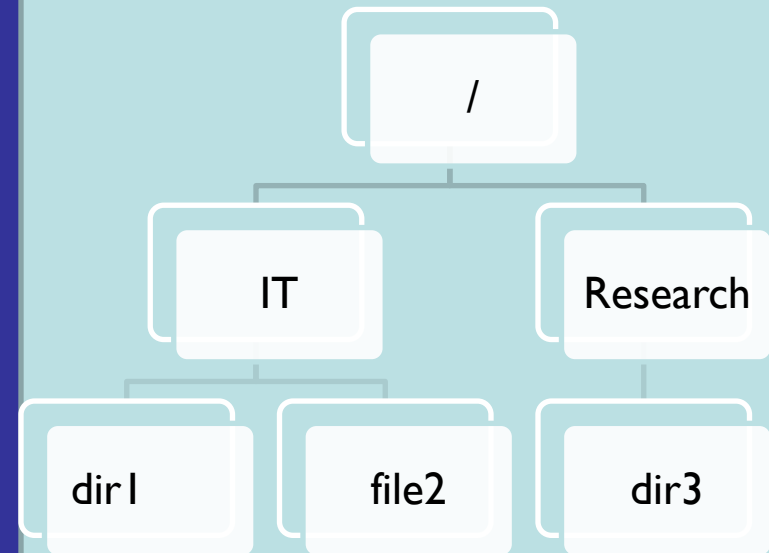
Example of Virtual Server

Physical Server

Virtual Server 1



Virtual Server 2



- ❑ NLM protocol is stateful
- ❑ NSM protocol provides a notification mechanism following client or server failure
- ❑ Recovery across server restart requires NFS Server to provide two guarantees:
 - ❑ NSM on NFS Server sends notifications to NFS Clients about server reboot
 - ❑ NFS Server enforces grace period to allow lock reclaims by NFS Clients after server reboot

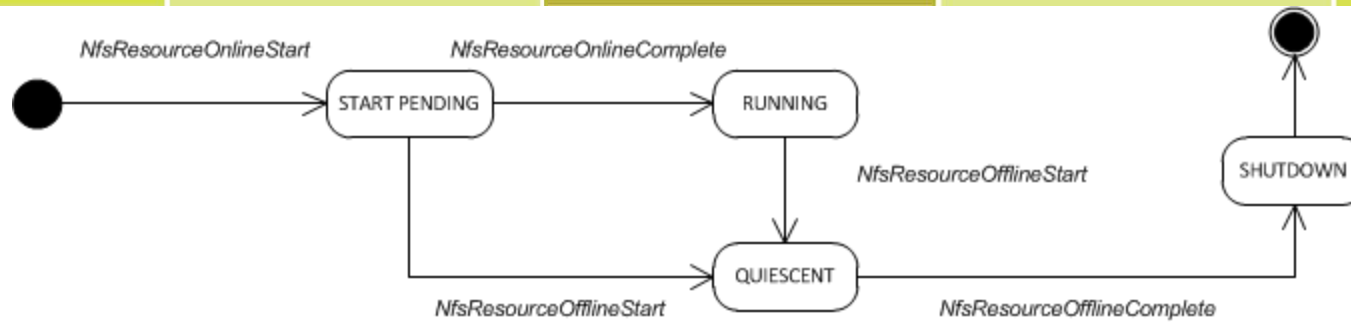
- ❑ One NLM lock file per volume
- ❑ Persistent lock record contains:
 - ❑ Caller name – identifies the client host
 - ❑ Lock file name, range, client address, protocol, version and procedure
 - ❑ Virtual server identifier
- ❑ Windows Failover Cluster allows individual NFS resource to be taken offline or failed over to other nodes
 - ❑ This requires NFS Server implement grace period per virtual server

- ❑ Advantages of Windows NFS Virtual Servers
 - ❑ NFS namespace scoping per netname
 - ❑ Per-IP endpoint registration to facilitate source IP on UDP reply packets
 - ❑ Grace period scoped per virtual server

NFS Server – IP Address Scoping

- ❑ NFS Server registers for IP address PnP notifications from the operating system
- ❑ When an IP address online notification is received, an LPC is performed to query Failover Cluster for the netname (and hence the NFS Virtual Server) that it should be scoped to
- ❑ All unknown endpoints are put in a deferred list while waiting for the Failover Cluster resource group, NFS Resource DLL and the NFS Virtual Server to come online

NFS Server – Virtual Server States



- ❑ As soon as **RUNNING** state is entered, NFS/NLM Grace Period process is initiated
- ❑ When in **START PENDING** state, RPC packets received are chained on a deferred list till **RUNNING** state transition is possible
- ❑ In **QUIESCENT** state, RPC packets are dropped

- ❑ Multiple ways to provision and manage NFS HA shares
 - ❑ NFS PowerShell cmdlets
 - ❑ NFS CIM (WMI) Provider API
 - ❑ Windows Failover Cluster Manager or Server manager UI

- ❑ Powershell Cmdlet:

- ❑ Create a new NFS HA share

- ❑ `new-nfsshare -name <sharename> -networkname <scoped cluster network name> -path <share path>`

- ❑ CIM API in MSFT_NfsServerTasks:

- ❑ `uint32 CreateShare([In]string Name,
[In] string Path,
[In] string NetworkName,
[In] string Authentication[],
...);`

Windows NFS v4.1 Server

❑ Scope and definition:

- ❑ Compliant with all mandatory aspects of RFC 5661
- ❑ Highly available – Windows Failover Clustering
- ❑ Identity Mapping support
 - ❑ passwd/group file mapping
 - ❑ Active Directory
 - ❑ ADLDS or 3rd party LDAP stores (RFC 2307 compliant)
 - ❑ User Name Mapping (legacy)
- ❑ RPCSEC_GSS – support for Krb5, Krb5i and Krb5p
- ❑ Multiprotocol access (SMB + NFS) to same share
- ❑ Volume mount point support

❑ Not currently implemented:

- ❑ ACL's
- ❑ Delegations
- ❑ Migration & Replication
- ❑ pNFS
- ❑ RDMA
- ❑ Other optional aspects of RFC 5661

- ❑ Windows NFS Server when deployed in Failover Cluster provides reliable platform for server application workloads
 - ❑ Tested with VMware ESX hypervisor vmdk storage

Questions?

Thank You!