Cluster improvements in Samba4

Günther Deschner
<gd@samba.org>

samba@redhat

- Samba developers in various teams (Storage, Identity, etc.)
- Several active contributors to Samba codebase
- Member of the Red Hat Gluster Storage team

Agenda

- Clustering compared: Samba/CTDB and Windows
- SMB3 Cluster Improvements
- SMB Transparent client failover: Witness Service
  - State of Witness implementation in Samba & Demo
- Remote cluster management: CLUSAPI
  - State of CLUSAPI implementation in Samba & Demo
- Further reading & Q/A

Clustering compared: Disclaimer

- File Server scenarios, not clustering of other applications like e.g. database servers
- Ignore shared storage options (e.g. CSV) and their setup

Clustering compared: Samba/CTDB

- **Cluster consists of multiple, usually cloned servers (nodes)**
- **Nodes share databases (record replication) & configuration**
- **Most common: domain member role**
  - All nodes share the same machine account
  - Only one sAMAccountName and secret
  - No distinction between nodes and cluster
- **Typically: Clients access cluster via public IP address(es)**
- **Management:**
  - cli: “ctdb” client tool
  - messaging: e.g. via smbcontrol client tool

Clustering compared: Windows

- **Cluster consists of multiple independent servers (nodes)**
- **Databases and configuration are kept in sync**
- **Most common: domain member role**
  - Each node keeps independent account in AD
  - Each cluster is a virtual server (DNS, IP addresses, machine account + SPNs)
- **Typically: Clients access cluster via public IP address(es)**
- **Management:**
  - “Failover Cluster Manager”
  - Power Shell commands

Windows File Server Cluster Types

- **Failover Cluster ("File Server for general use")**
  - "A failover cluster is a group of independent computers that work together to increase the availability and scalability of clustered roles"
  - All fileshares online only on one node at a time ("active-passive")
- **Scale Out Cluster ("Scale-Out File Server for application data")**
  - "Scale-Out File Server is a feature that is designed to provide scale-out file shares that are continuously available for file-based server application storage"
  - All fileshares simultaneously online on all nodes ("active-active")
  - Adding additional nodes transparently to the application to "scale out"
  - Available since SMB3 (Windows 2012)

SMB3 Cluster improvements

- SMB Transparent Failover (Witness)
- SMB Multichannel
- SMB Direct (RDMA)

SMB Transparent Client Failover: Witness

- New DCE/RPC Service to “witness” availability of other services, in particular SMB3 connections
- Prompt and explicit notifications about failures in highly available systems
- Allows Continuous Availability of SMB shares in clustered environments
- Controlled way of dealing with reconnects instead of detecting failures due to timeouts
- Available since Windows 2012 (SMB3)

Recap:

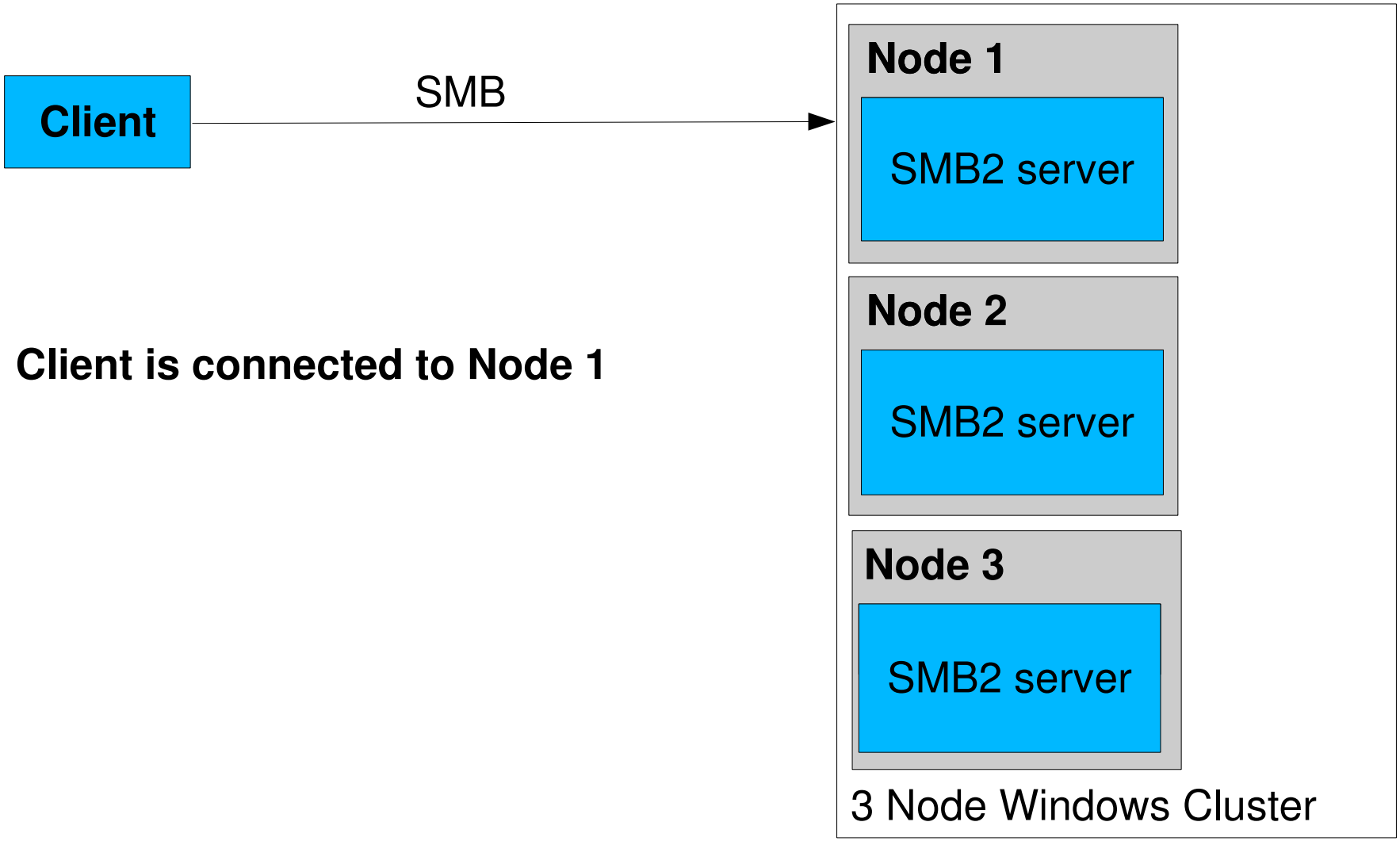
Client Failover with SMB1/2/3

(Windows and Samba/CTDB)

Failover in SMB1/SMB2

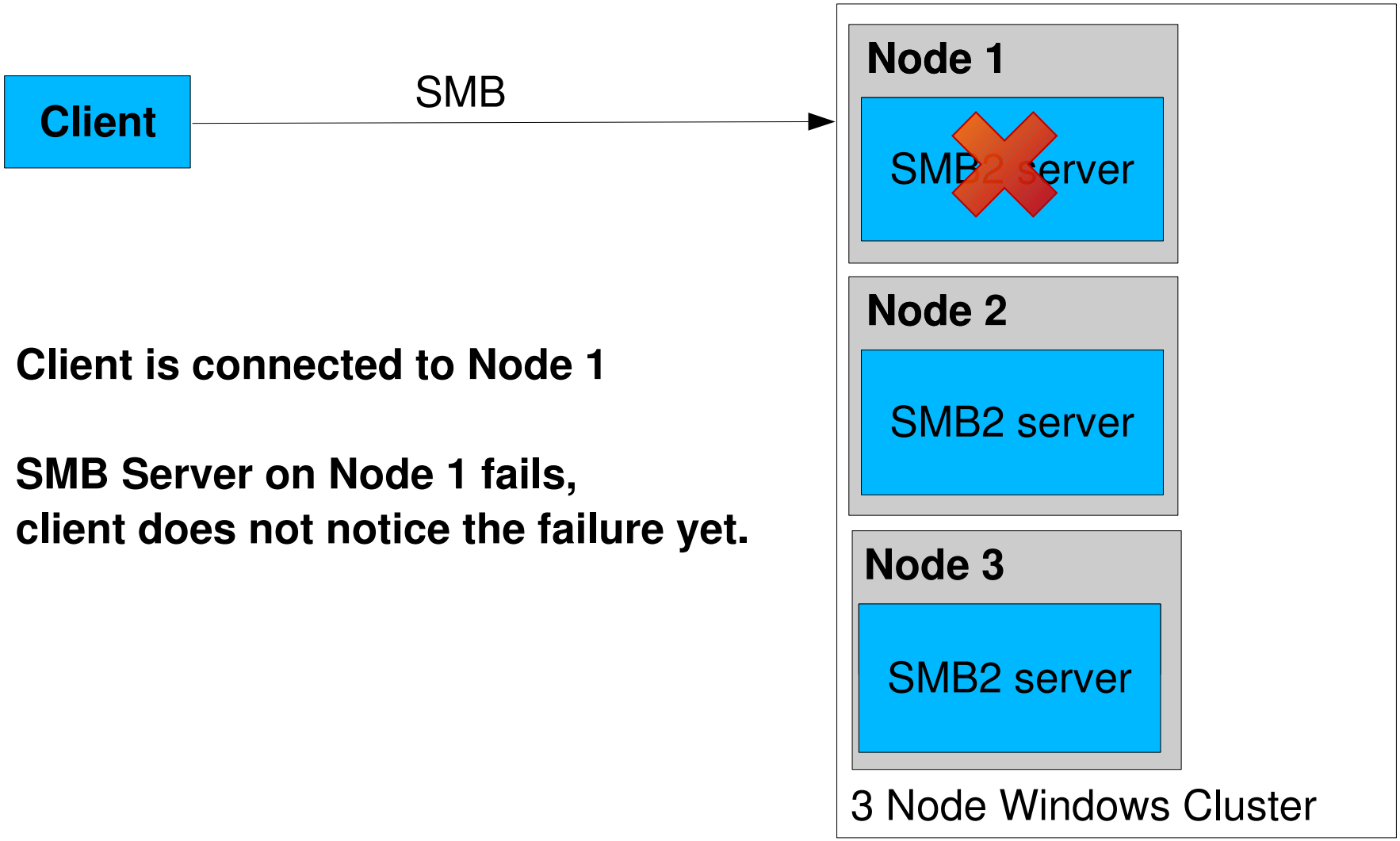
- Uncontrolled, clients detect unavailability by running into timeouts or by using keep alive mechanisms
- Clients reconnect after TCP/IP connection timeout
- Slow, unreliable, unpredictable
- Not all applications deal with stale connections good enough

Failover in SMB1/SMB2



Client is connected to Node 1

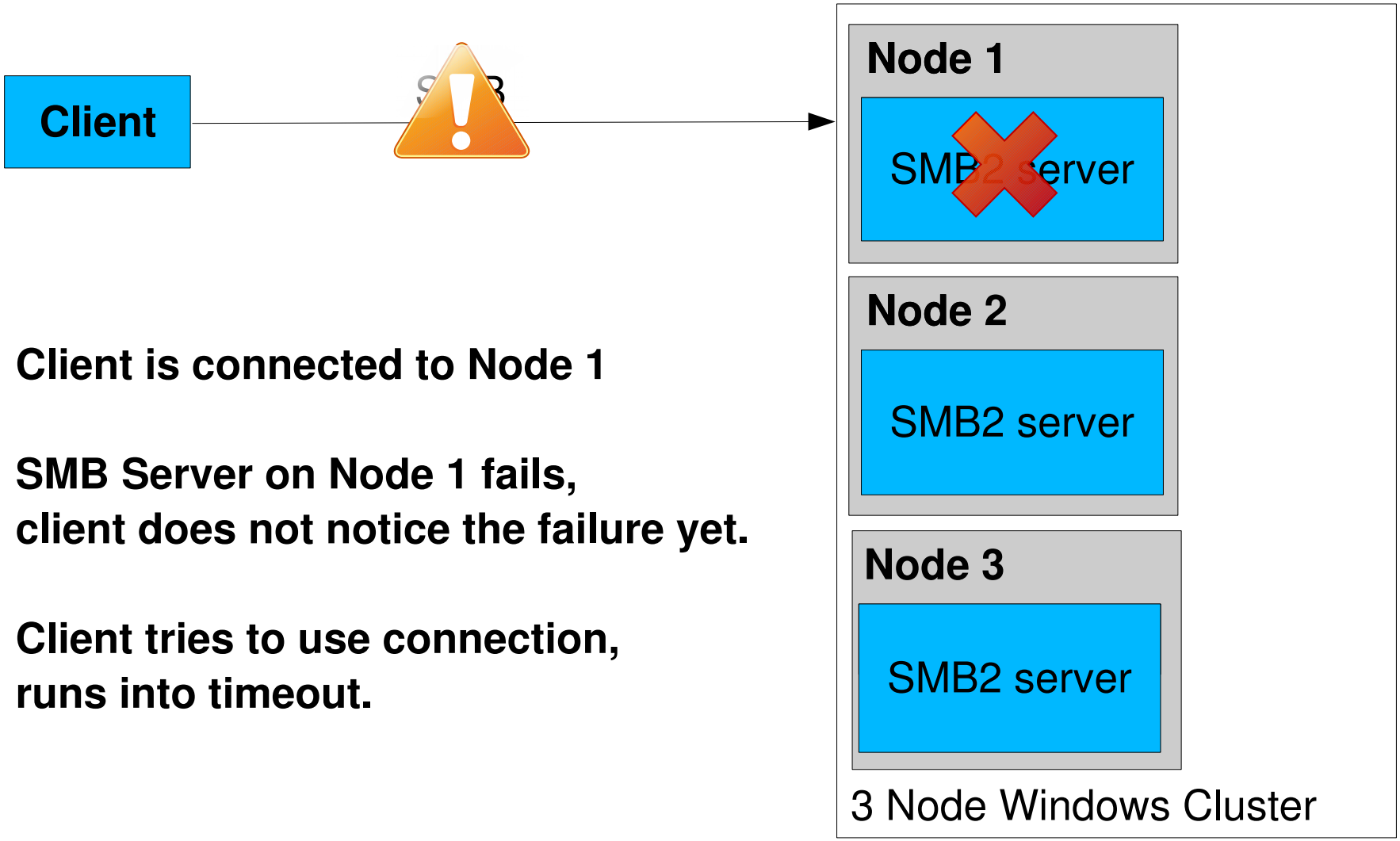
Failover in SMB1/SMB2



Client is connected to Node 1

SMB Server on Node 1 fails,
client does not notice the failure yet.

Failover in SMB1/SMB2

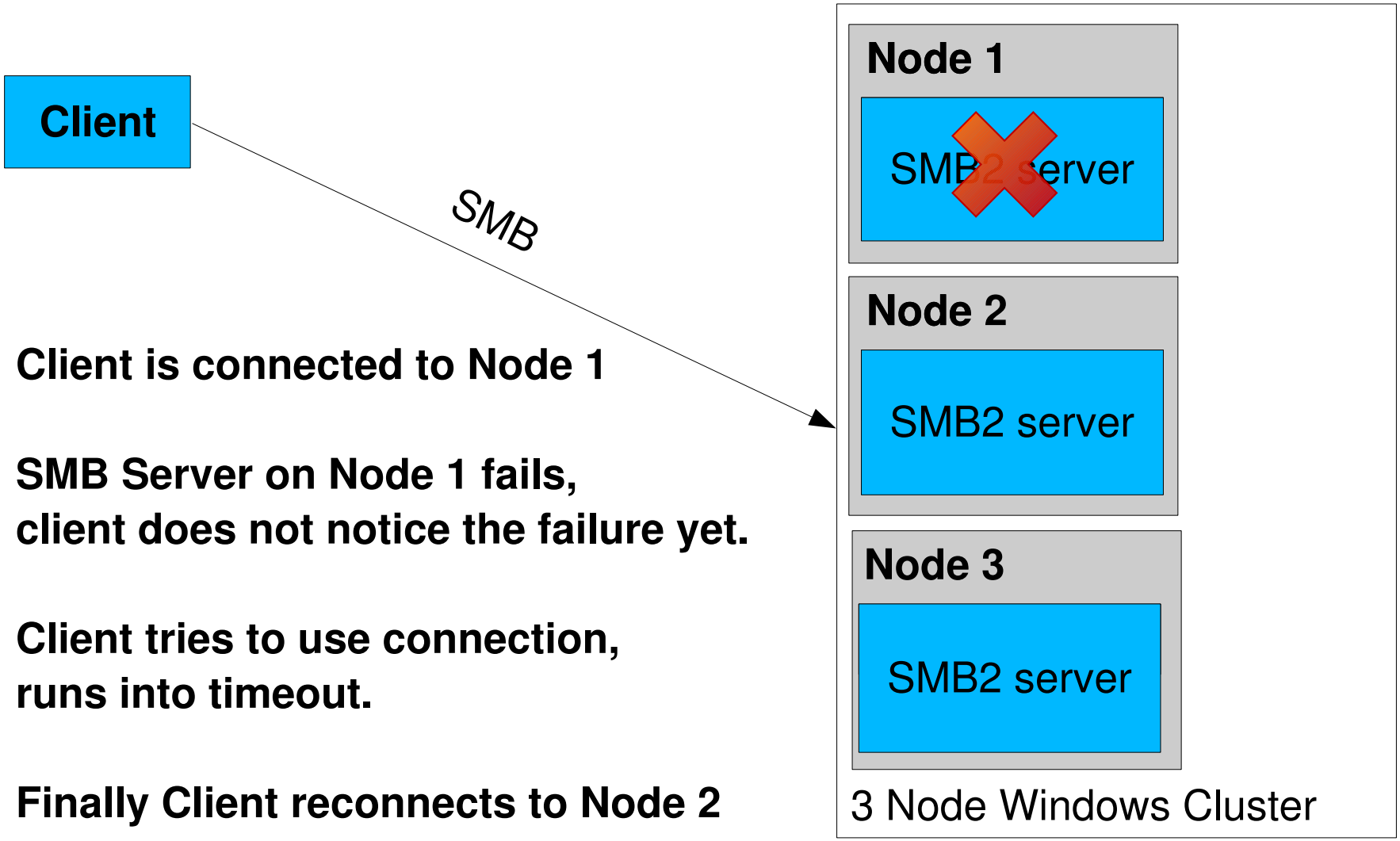


Client is connected to Node 1

SMB Server on Node 1 fails,
client does not notice the failure yet.

Client tries to use connection,
runs into timeout.

Failover in SMB1/SMB2



Client is connected to Node 1

SMB Server on Node 1 fails,
client does not notice the failure yet.

Client tries to use connection,
runs into timeout.

Finally Client reconnects to Node 2

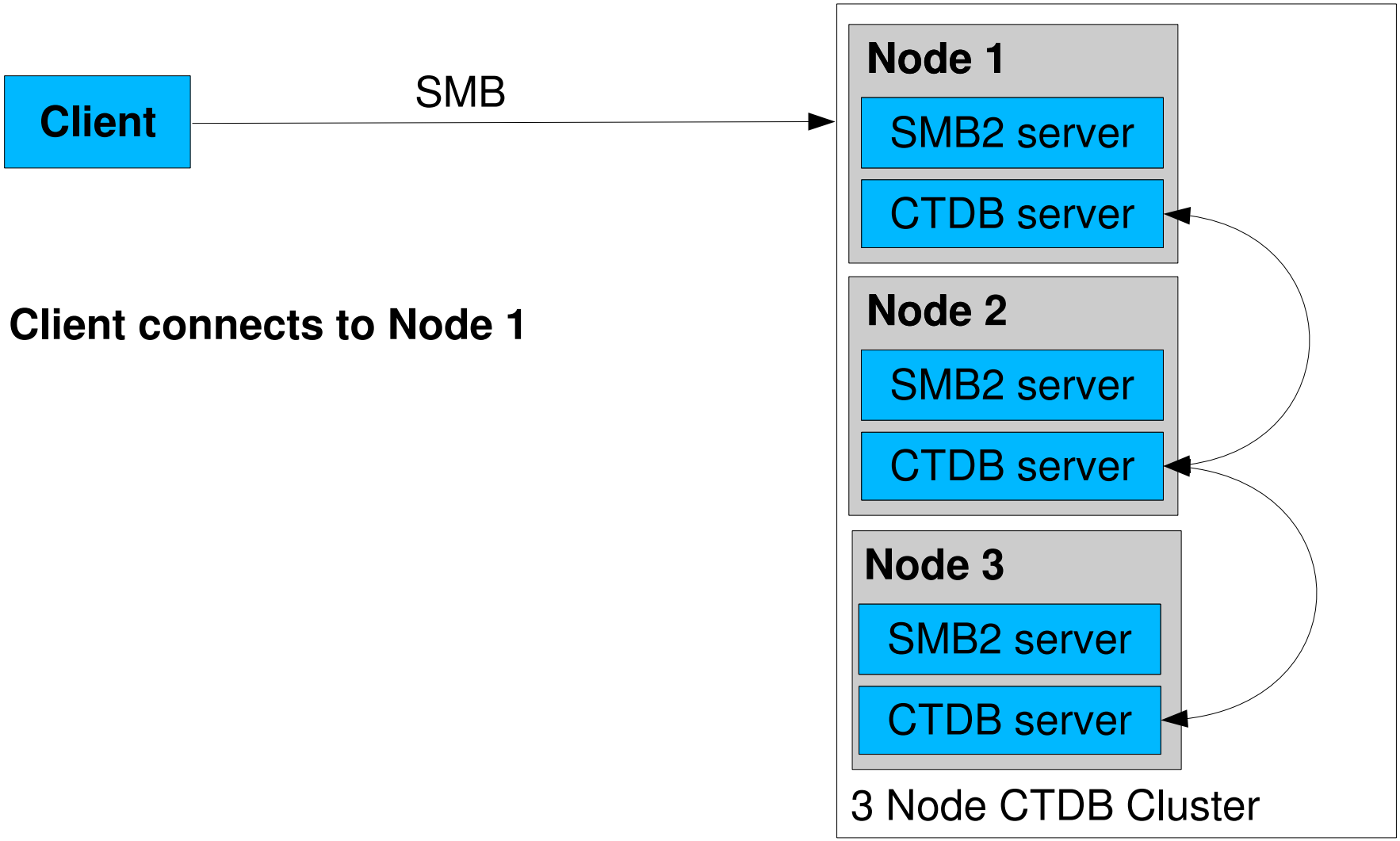
Failover in SMB1/SMB2 with CTDB

- Since 2007, a Samba cluster with CTDB is usually aware of failures before the client is
- In case of failure CTDB can proactively route the clients to another node
- With CTDB the cluster coordinates the failover, not the client

Failover in SMB1/SMB2 with CTDB

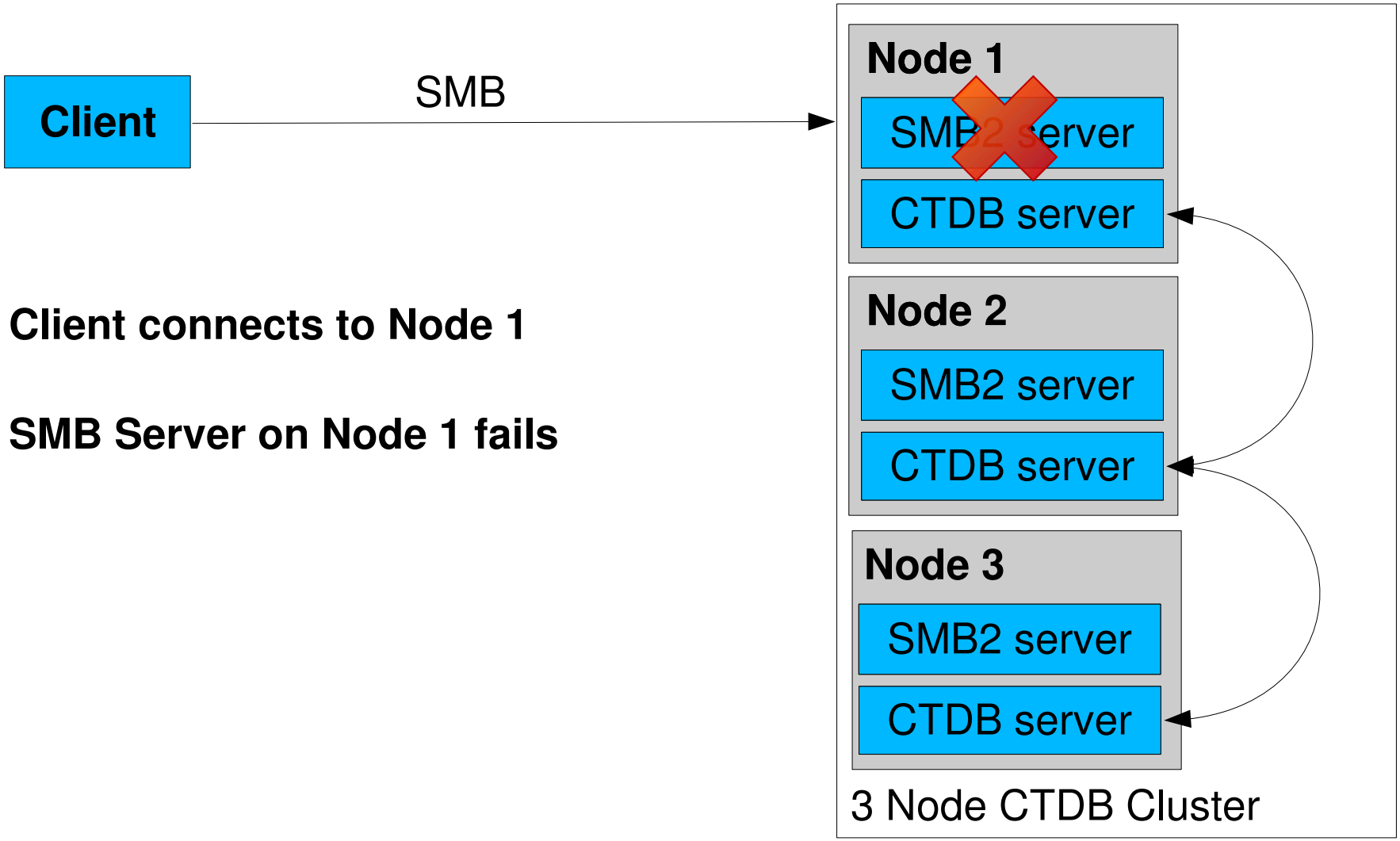
- CTDB uses Tickle ACKs to speedup recovery
- Tickle ACKs:
  - are TCP ACK packets with invalid sequence and acknowledge numbers
  - cause a TCP client to reestablish a connection with proper sequence numbers, immediately
  - were invented/discovered by tridge while working on CTDB

Failover in SMB1/SMB2 with CTDB



Client connects to Node 1

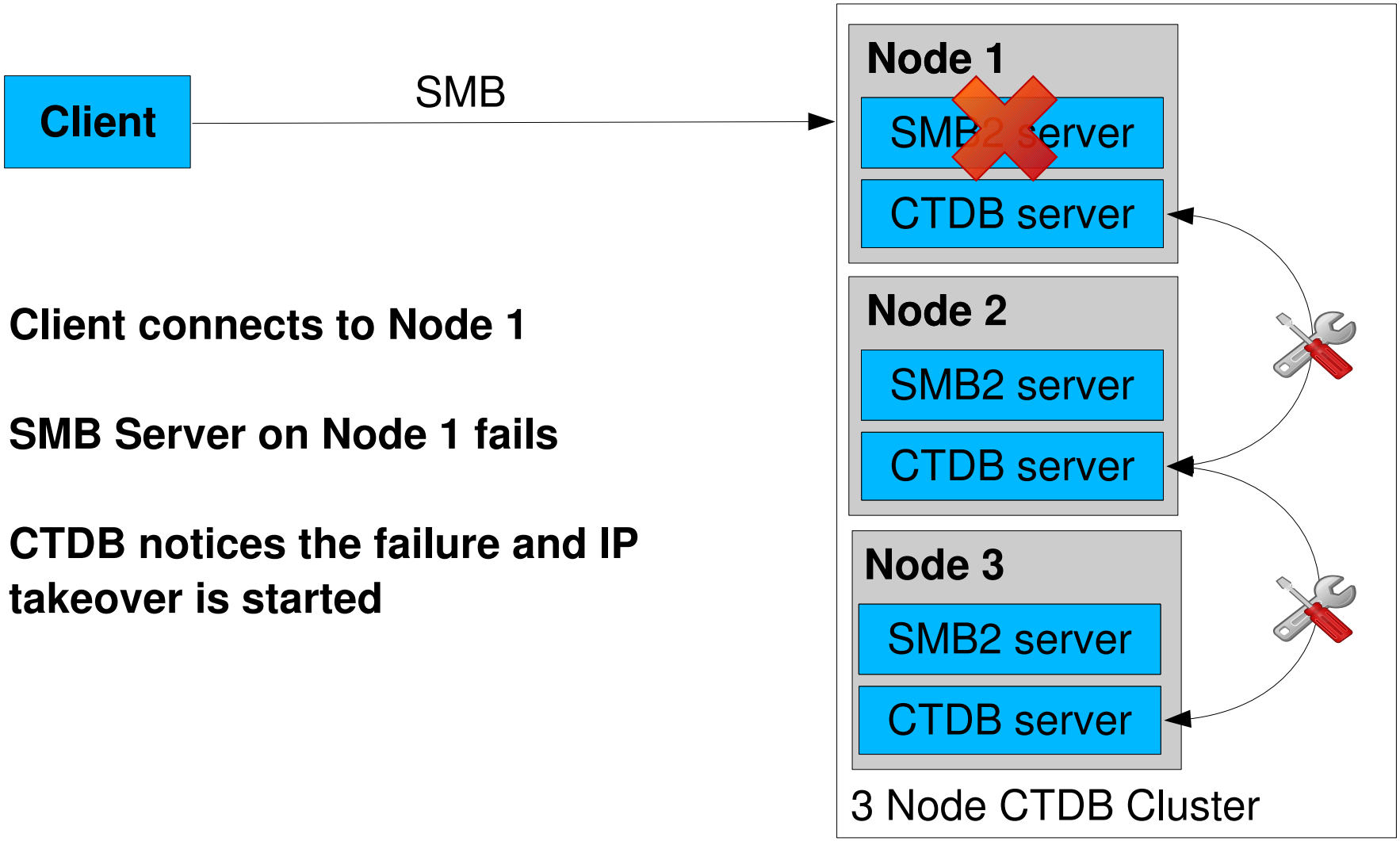
Failover in SMB1/SMB2 with CTDB



Client connects to Node 1

SMB Server on Node 1 fails

Failover in SMB1/SMB2 with CTDB

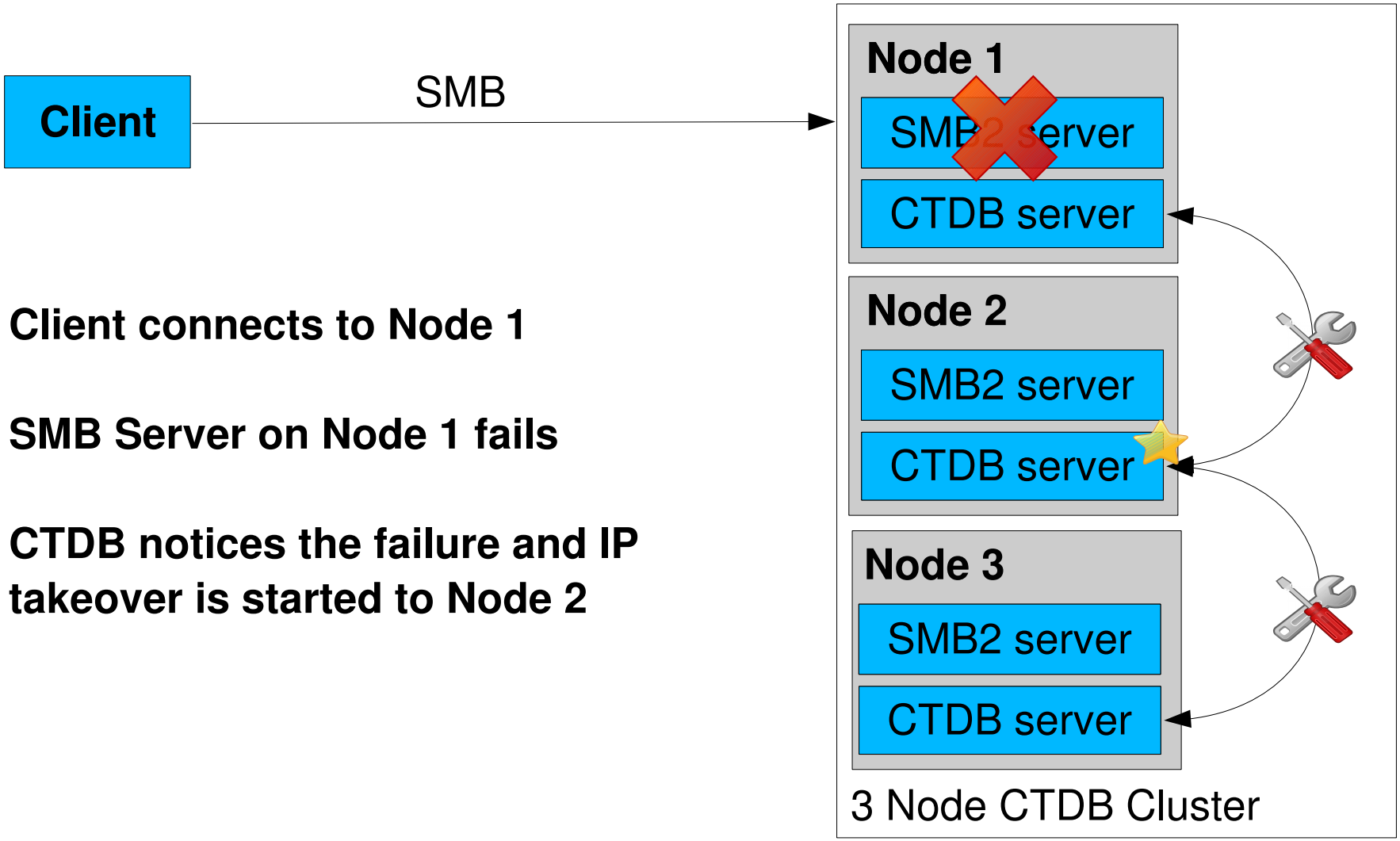


Client connects to Node 1

SMB Server on Node 1 fails

CTDB notices the failure and IP takeover is started

Failover in SMB1/SMB2 with CTDB

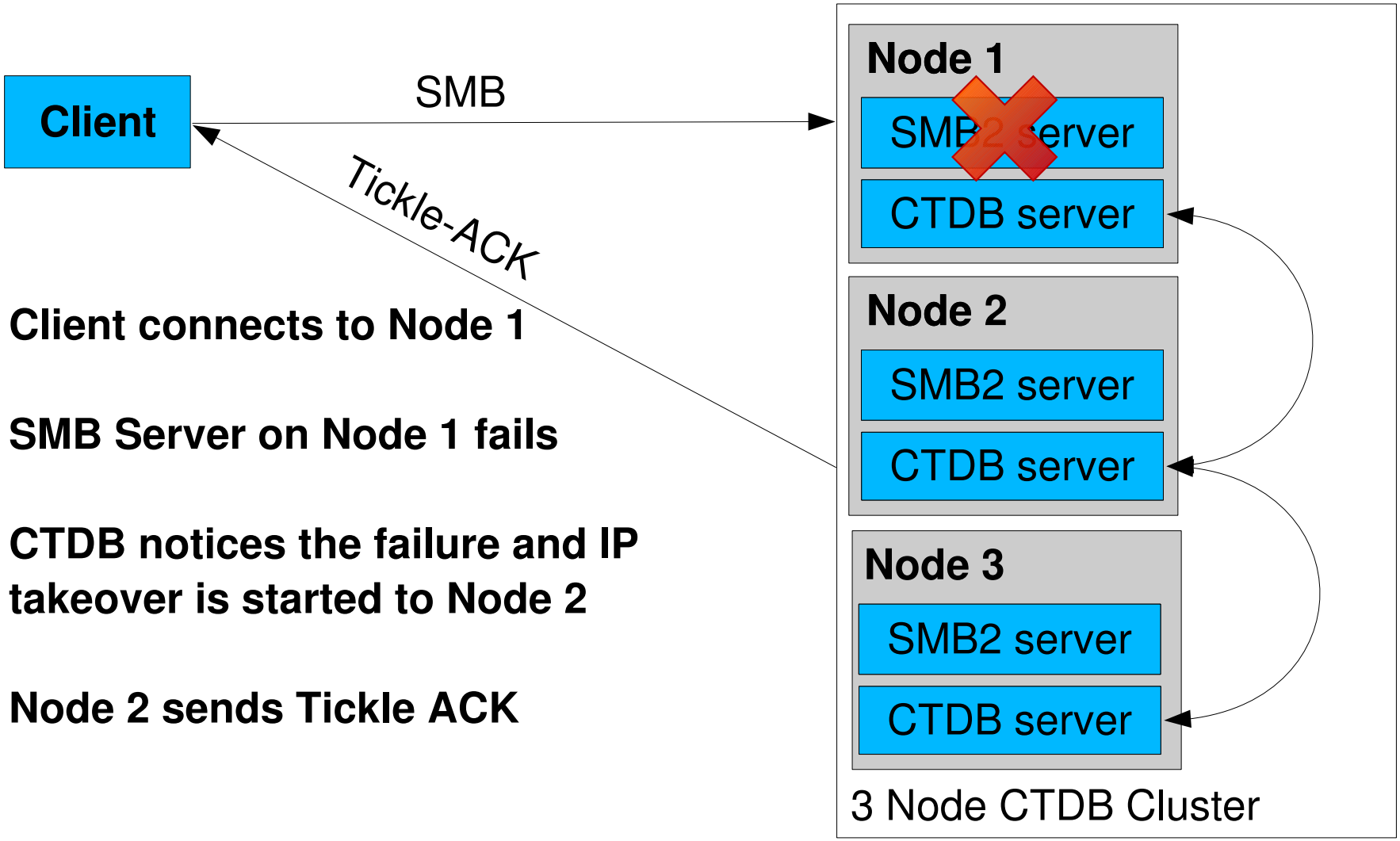


Client connects to Node 1

SMB Server on Node 1 fails

CTDB notices the failure and IP takeover is started to Node 2

Failover in SMB1/SMB2 with CTDB



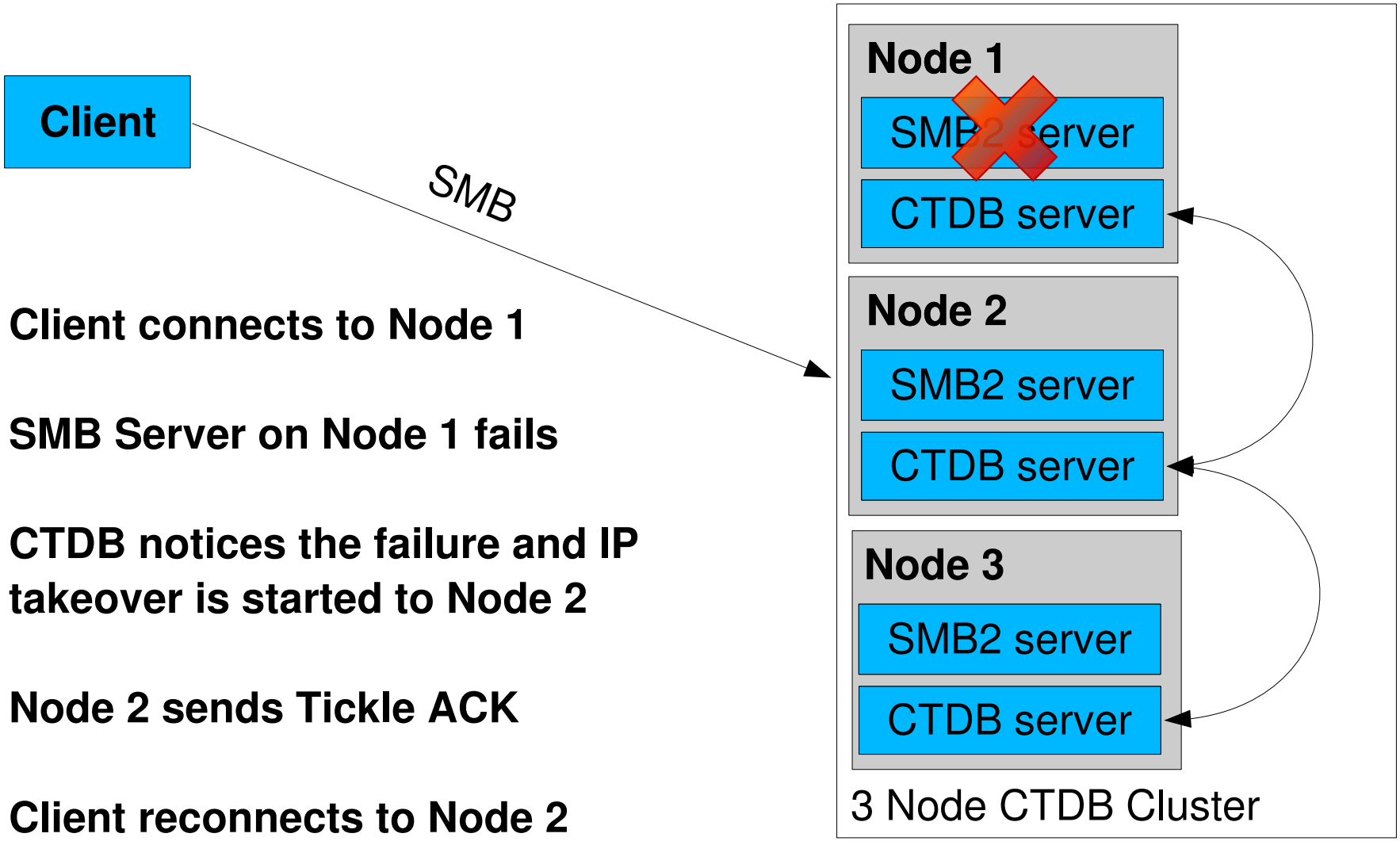
Client connects to Node 1

SMB Server on Node 1 fails

CTDB notices the failure and IP takeover is started to Node 2

Node 2 sends Tickle ACK

Failover in SMB1/SMB2 with CTDB



Client connects to Node 1

SMB Server on Node 1 fails

CTDB notices the failure and IP takeover is started to Node 2

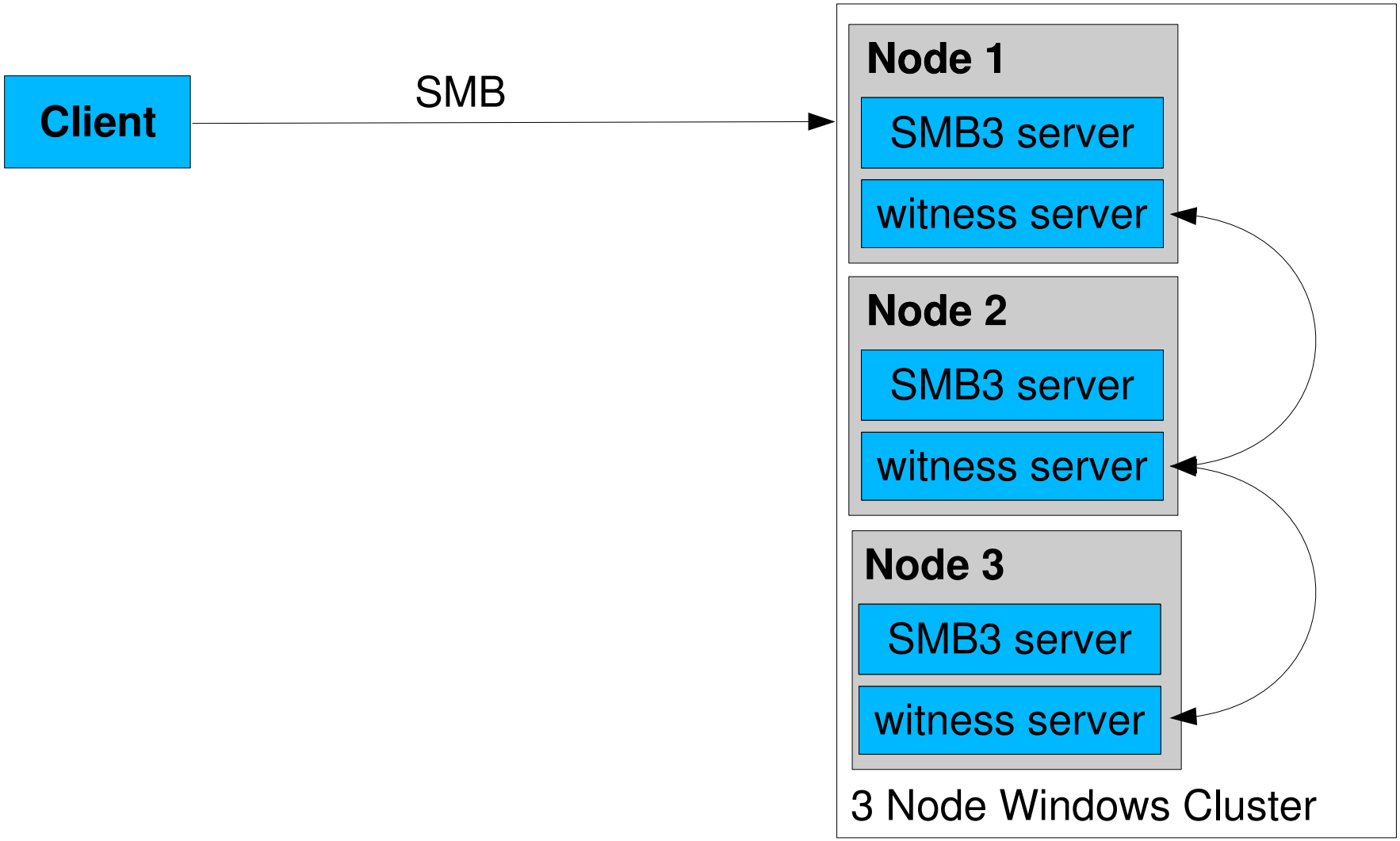
Node 2 sends Tickle ACK

Client reconnects to Node 2

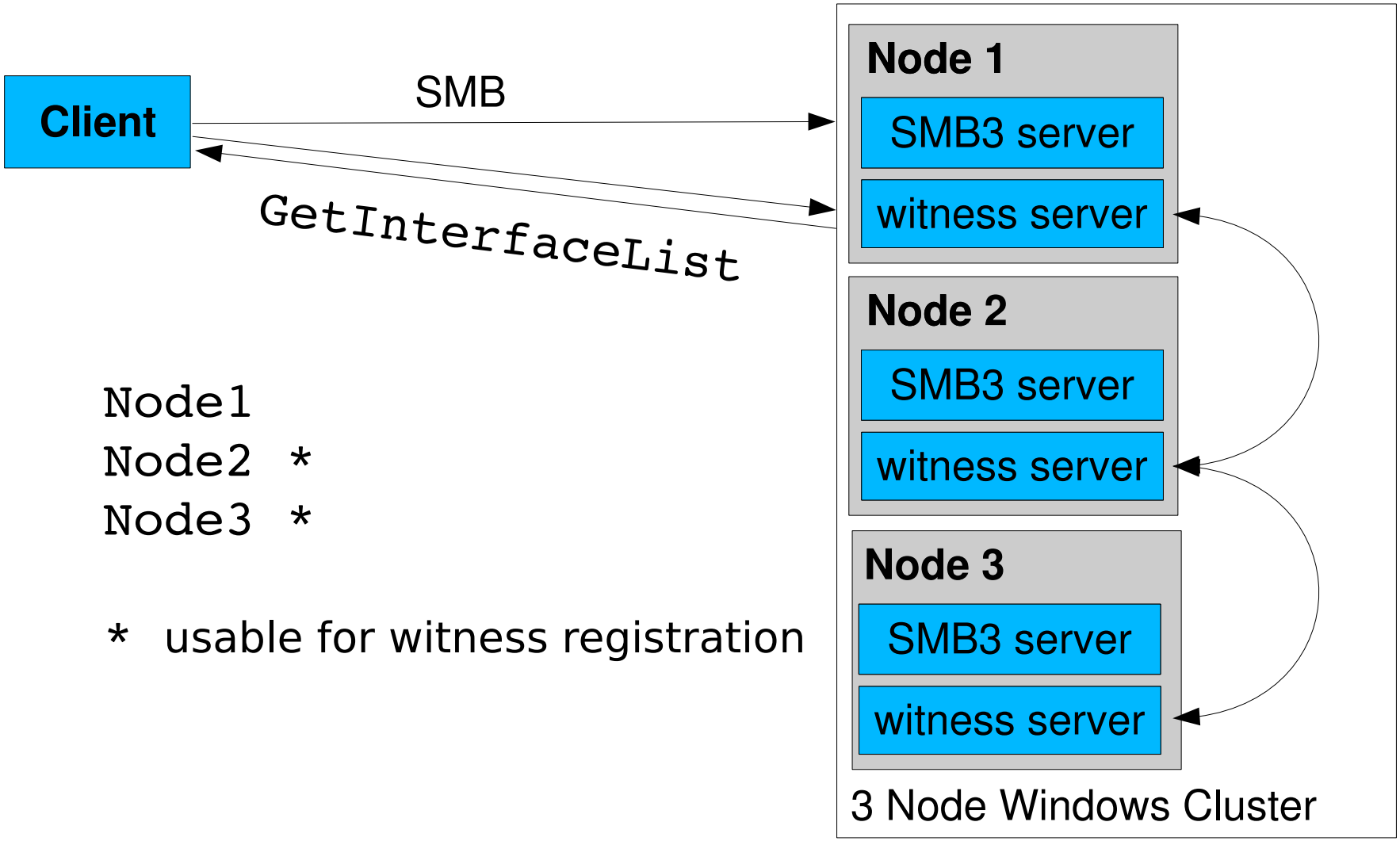
Failover in SMB3

- SMB3 achieves transparent failover via several new features:
  - Continuous Availability
  - Persistent Handles
  - Witness
- This leads to faster recovery from unplanned node failures
- Also allows planned and controlled migration of clients between cluster nodes

Failover in SMB3



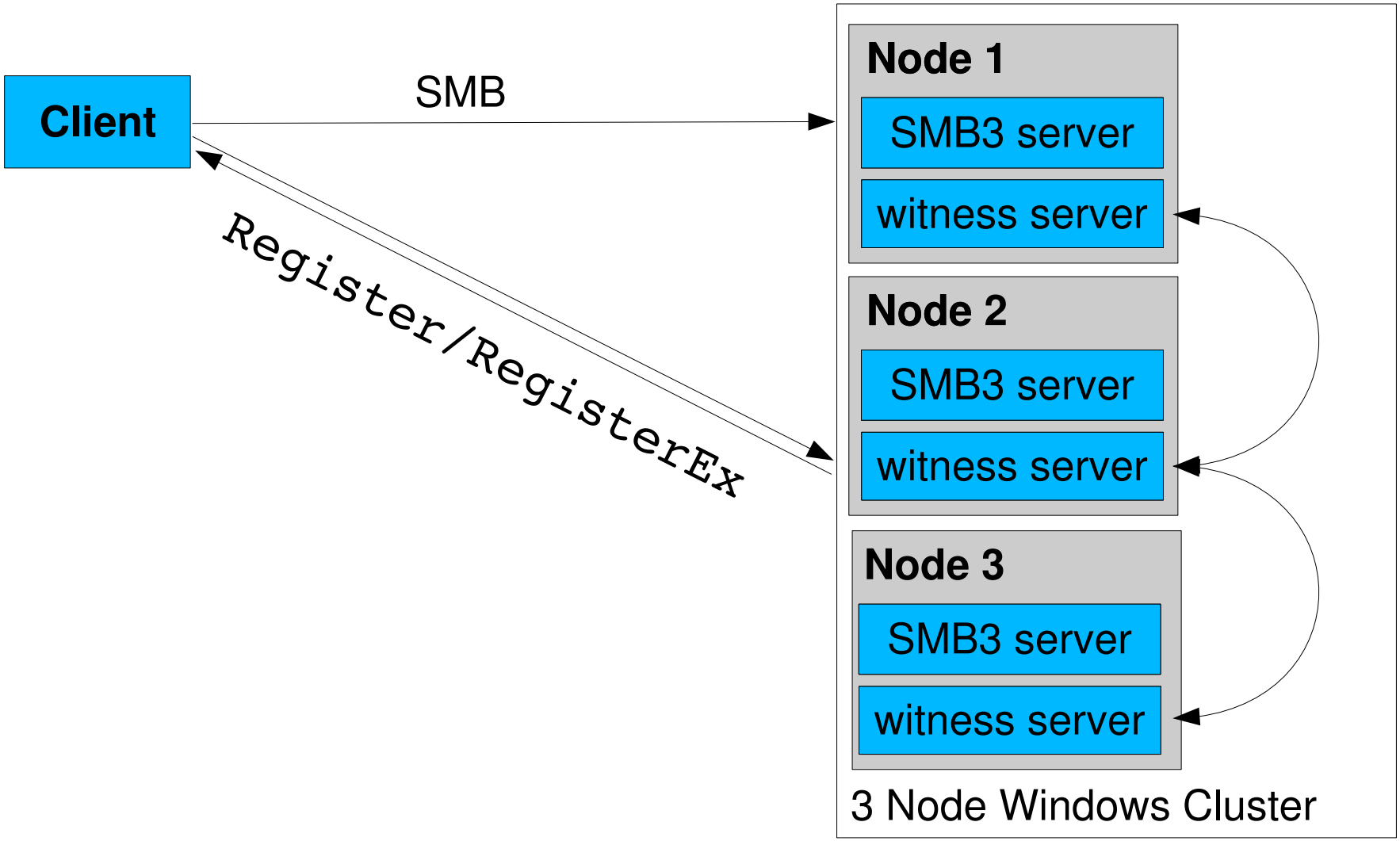
Failover in SMB3



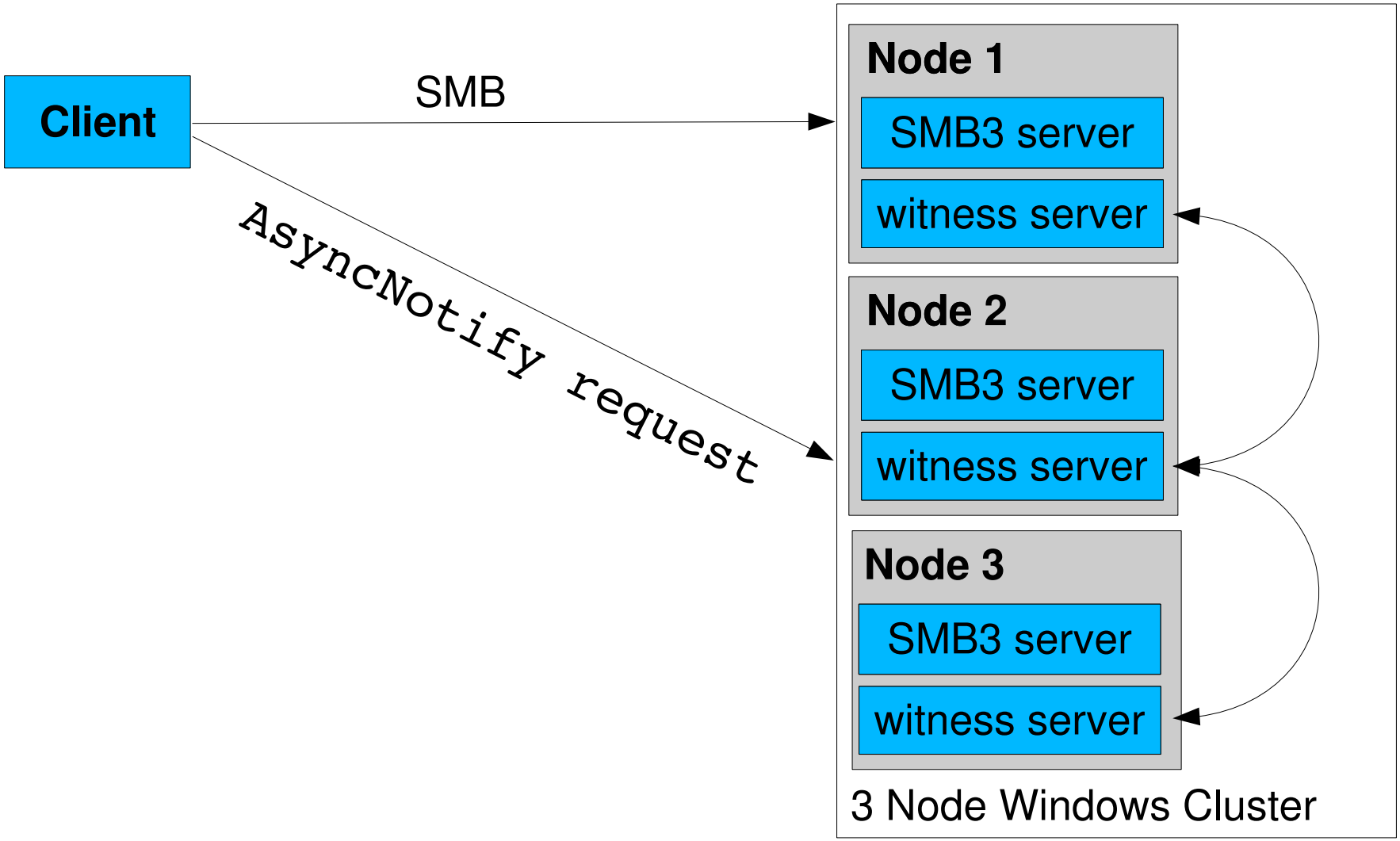

```
Node1
Node2  *
Node3  *
```

* usable for witness registration

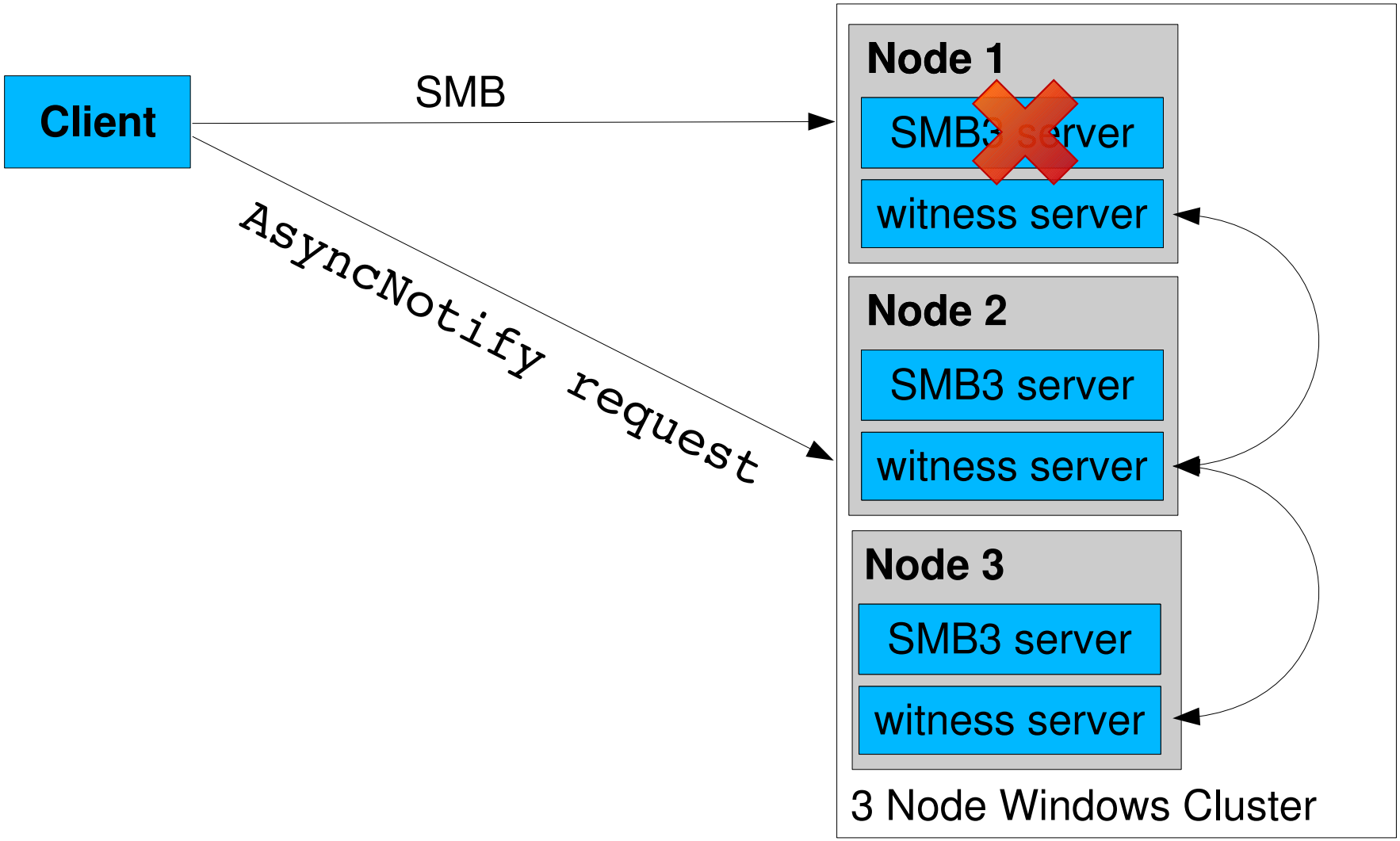
Failover in SMB3



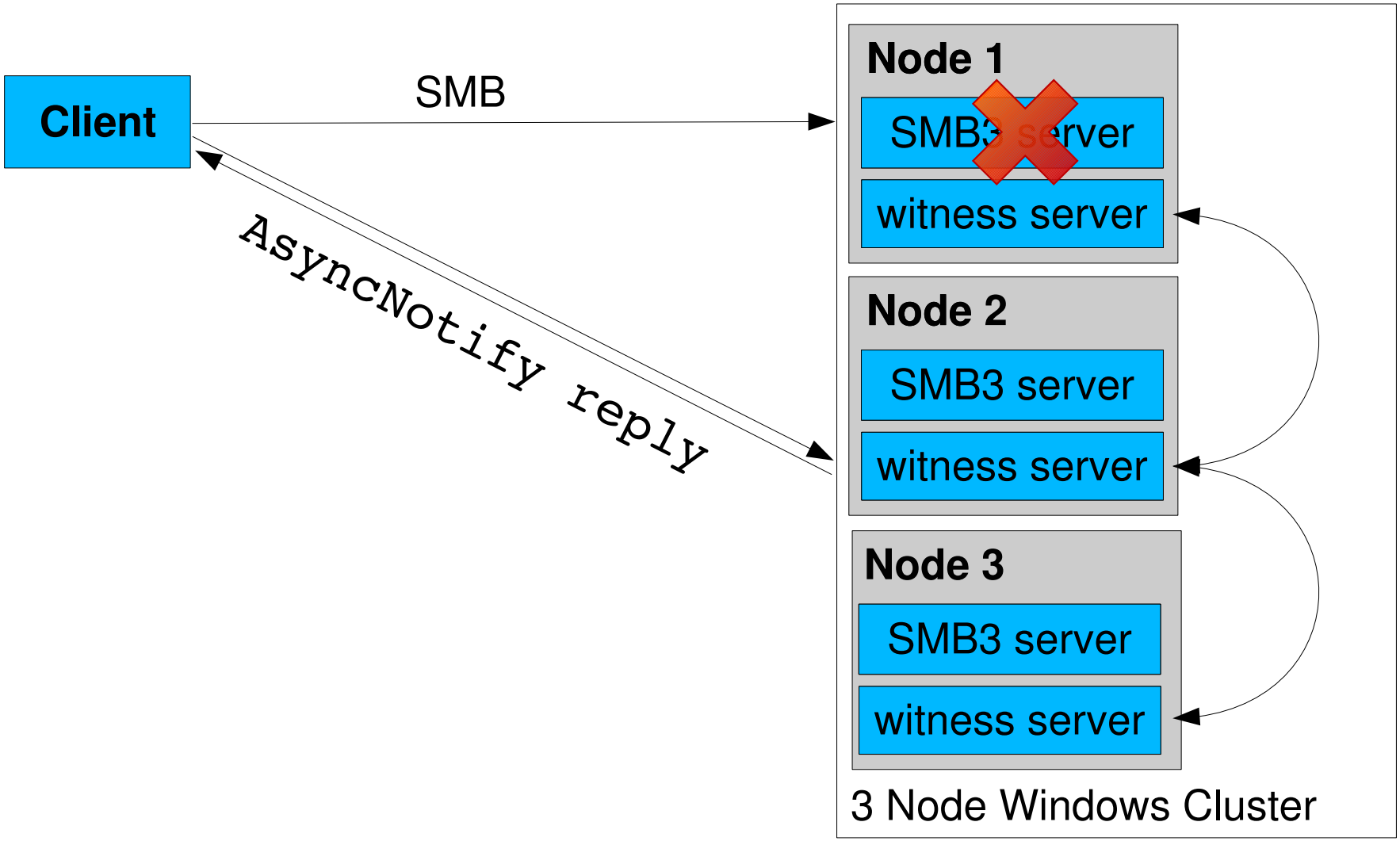
Failover in SMB3



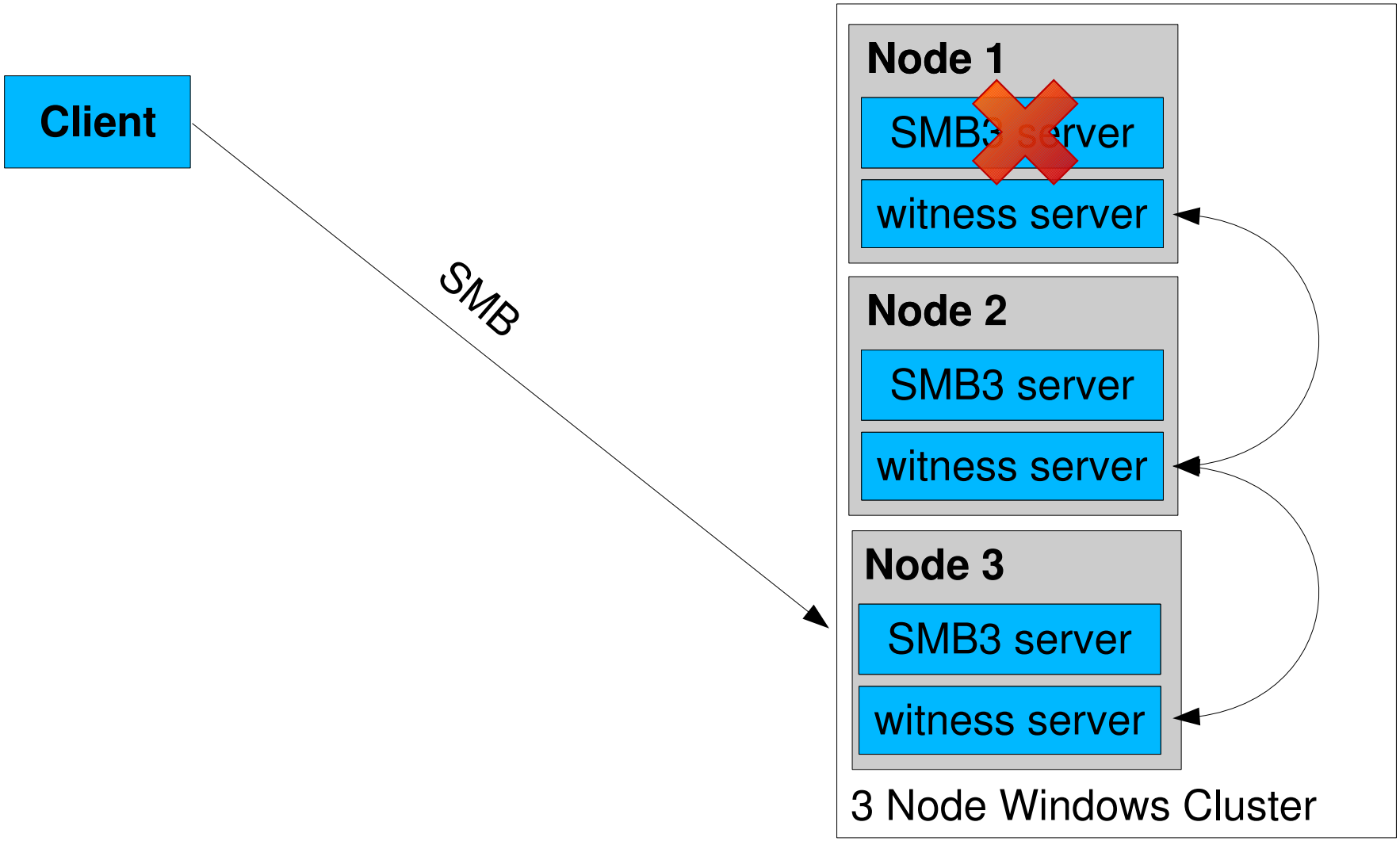
Failover in SMB3



Failover in SMB3



Failover in SMB3



Relationship to SMB3 protocol

- Per share flag enables use of Witness Protocol
- MS-SMB2: “The specified share is present on a server configuration which provides monitoring of the availability of share through the Witness service specified in [MS-SWN]”
- SMB2 TREE_CONNECT Response Capability Flag: SMB2_SHARE_CAP_CLUSTER = 0x00000040
- Windows 10 clients won't use witness unless also the SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY bit is set
- Currently for testing:
  - smbd:announce CLUSTER = yes
  - smbd:announce CA = yes

The witness interface

- Surprisingly short spec (only 47 pages)
- Version 1, SMB 3.0 (Windows 2012, Windows 8)
- Version 2, SMB 3.02 (Windows 2012 R2, Windows 8.1)
- Only 5 opcodes in the interface:
  - `_witness_GetInterfaceList`
  - `_witness_Register`
  - `_witness_Unregister`
  - `_witness_AsyncNotify`
  - `_witness_RegisterEx` (witness version 2)

GetInterfaceList

```
DWORD WitnessrGetInterfaceList(
    [in] handle_t Handle,
    [out] PWITNESS_INTERFACE_LIST * InterfaceList);
```

- Returns list of network interfaces with IPv4 and/or IPv6 addresses
- Each interface carries information about the interfaces version, state and whether it is a good candidate for witness use

Witness_InterfaceInfo

```
interfaces: struct witness_interfaceInfo
   group_name       : 'MTHELENA'
   version          : WITNESS_UNSPECIFIED_VERSION (-1)
   state            : WITNESS_STATE_AVAILABLE (1)
   ipv4             : 192.168.56.108
   ipv6             : ::
   flags            : 0x00000005 (5)
         1: WITNESS_INFO_IPv4_VALID
         0: WITNESS_INFO_IPv6_VALID
         1: WITNESS_INFO_WITNESS_IF
```

RegisterEx

```
DWORD WitnessrRegisterEx(
    [in] handle_t Handle,
    [out] PPCONTEXT_HANDLE ppContext,
    [in] ULONG Version,
    [in] [string] [unique] LPWSTR NetName,
    [in] [string] [unique] LPWSTR ShareName,
    [in] [string] [unique] LPWSTR IpAddress,
    [in] [string] [unique] LPWSTR ClientComputerName,
    [in] ULONG Flags,
    [in] ULONG KeepAliveTimeout);
```

- Available with Windows 2012 R2 (Witness v2)
- Witness keepalive as client can define KeepAliveTimeout
- Server returns with ERROR_TIMEOUT after KeepAliveTimeout has expired (Windows 8.1 default 120 seconds)

AsyncNotify

```
DWORD WitnessrAsyncNotify(
    [in] handle_t Handle,
    [in] PCONTEXT_HANDLE_SHARED pContext,
    [out] PRESP_ASYNC_NOTIFY * pResp);
```

- Asynchronous call
- Clients send request and wait, and wait, and wait...
- Only in the event of a notification issued by the cluster the client receives a reply
- Witness keep-alive mechanism available in Witness v2 (SMB 3.02)

AsyncNotify call

- 4 different events are currently defined in the protocol:
- WITNESS_NOTIFY_RESOURCE_CHANGE
  - Notify about a resource change state (available, unavailable)
- WITNESS_NOTIFY_CLIENT_MOVE
  - Notify a connected client to move no another node
- WITNESS_NOTIFY_SHARE_MOVE (only v2)
  - Notify that a share has been moved to another node
- WITNESS_NOTIFY_IP_CHANGE (only v2)
  - Notify about an ip address change (online, offline)

DCE/RPC requirements

- EPMAPPER with ncacn_ip_tcp support
- DCE/RPC sign & seal (SPNEGO,KRB5,NTLMSSP)
- Asynchronous DCE/RPC server (for MS-PAR, MS-CMRP, MS-SWN, etc.)
  - Samba4
  - Samba3 by David Disseldorp <ddiss@samba.org>
  - New DCE/RPC infrastructure from Metze
- mgmt service (Remote DCE/RPC service management)
  - Two implementations available, none is published yet.
  - mgmt_inq_princ_name() for different node principals

witnessd server

- Standalone binary, using new infrastructure invented for spoolssd
- Independent binary so any Samba server problem does not interfere with witness messaging
- Needs to register for at least 4 notification events (messaging)
- Configuration and possibly Server State store
- Very close integration with CTDB:
  - CTDB maintains all available cluster state information
  - CTDB already has mechanisms to communicate failures between the nodes
  - CTDB could easily reuse tickle-ack hooks for witness notifications

Witness support in Samba

- Early PoC implementation by Gregor Beck and Stefan Metzmacher from 2012
- Wireshark dissector for witness protocol → upstream
- Full IDL and torture tests in Samba Git repository → upstream
- Witness Service is on Samba Roadmap as a funded project
  - Goal was Samba 4.4/4.5 but delayed (badlock, multichannel, etc.)

Witness support in Samba

- First working server implementation built with Samba 4 DCE/RPC together with Jose Rivera <jarrpa@samba.org> and Stefan Metzmacher <metze@samba.org> (SDC 2015)
- BUT:
  - Samba 4 DCE/RPC services are not meant to be run on a domain member (samba binary would normally not even startup)
  - Samba 4 is not ctdb aware
- We need to finish the common DCE/RPC infrastructure!
- TODO: Persistent state of witness registrations
- Witness server support comes along with persistent handles

Witness admin tool

- Frontend for management tasks of witness server:
  - listing of active, connected clients
    (shared state stored in distributed database)
  - Tool to manually move Clients to other nodes
    (similar to Move-SmbWitnessClient PowerShell cmdlet)
  - Tool to move share to other node
- Currently implemented as part of the smbcontrol management and messaging tool
  - “smbcontrol witnessd witnessnotify”
  - subcommands: change, move, sharemove, ipnotify, getstate

TODO: Witness Client

- Several existing SMB clients would benefit from supporting Witness, including:
  - CIFS Kernel module
  - smbclient
  - Libsmbclient
- Witness client engine for smbclient can be written today
- Alternatives to CTDB could be introduced for the purposes of tracking the state of resources in the cluster (e.g. Pacemaker).

Witness
DEMO

Witness testing

- rpcclient witness command set
- smbtorture local.ndr.witness
  - Just tests correctness of the NDR marshalling/unmarshalling
- smbtorture rpc.witness
  - Test correctness of the DCE/RPC calls
- Fundamental problem: how to test a cluster ? How to test resource changes? How to test node failures ?
- Windows Failover Cluster Manager does resource changes with yet another DCE/RPC protocol

Remote Cluster Management: clusapi

- "Failover Cluster Management API" (MS-CMRP)
  - > 200 opcodes
  - > 600 pages protocol spec
  - Used by Microsoft "Failover Cluster Manager"
- DCE/RPC based interface (over ncacn_ip_tcp[seal])
- Two protocol versions:
  - Version 2: ncacn_udp
  - Version 3: ncacn_ip_tcp
- Samba has IDL (version 3) and a growing torture test suite
  → upstream
- Wireshark dissector

CLUSAPI wireshark dissector

failover_cluster_manager_ctdb.pcapng [Wireshark 2.1.0-git (v2.1.0rc0-3006-gaa76b27 from master)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: clusapi | Expression... | Clear | Apply | Save

No.	Time	Source	Destination	Protocol	Length	Info
927	8.456977051	192.168.124.1	192.168.124.116	CLUSAPI	194	GetClusterName request
929	8.458059899	192.168.124.116	192.168.124.1	CLUSAPI	210	GetClusterName response
930	8.459796614	192.168.124.1	192.168.124.116	CLUSAPI	146	OpenClusterEx request
931	8.460558882	192.168.124.116	192.168.124.1	CLUSAPI	146	OpenClusterEx response
932	8.461469270	192.168.124.1	192.168.124.116	CLUSAPI	146	CreateEnum request
937	8.462943092	192.168.124.116	192.168.124.1	CLUSAPI	274	CreateEnum response
939	8.464209445	192.168.124.116	192.168.124.1	CLUSAPI	258	GetClusterVersion2 response
943	8.542445170	192.168.124.116	192.168.124.1	CLUSAPI	210	GetClusterName response
946	8.544178024	192.168.124.116	192.168.124.1	CLUSAPI	258	GetClusterVersion2 response
947	8.547960525	192.168.124.1	192.168.124.116	CLUSAPI	146	GetRootKey request
964	8.551353631	192.168.124.116	192.168.124.1	CLUSAPI	146	GetRootKey response
965	8.556839830	192.168.124.1	192.168.124.116	CLUSAPI	210	QueryValue request[Malformed Packet]
966	8.558250786	192.168.124.116	192.168.124.1	CLUSAPI	146	QueryValue response, Error: WERR_MORE_DATA
967	8.560952188	192.168.124.1	192.168.124.116	CLUSAPI	210	QueryValue request[Malformed Packet]
968	8.563172635	192.168.124.116	192.168.124.1	CLUSAPI	210	QueryValue response

```
0000  22 dc 01 29 79 d3 52 54  00 10 42 1e 08 00 45 00   "..)y.RT ..B...E.
0010  00 b4 01 de 40 00 40 06  be 9f c0 a8 7c 01 c0 a8   ....@.@. ....|...
0020  7c 74 ce b2 04 00 58 f1  b8 be eb 82 2d 7d 80 18   |t....X. ....-}..
0030  00 ed 7a 6d 00 00 01 01  08 0a 14 b5 b4 92 14 b5   ..zm.... ........
0040  b4 92 05 00 00 83 10 00  00 00 80 00 10 00 02 00   ........ ........
0050  00 00 34 00 00 00 00 00  03 00 b2 b8 7d b9 63 4c   ..4..... ....}.cL
0060  cf 11 bf f6 08 00 2b e2  3f 2f 42 e5 81 ee d9 aa   ......+. ?/B.....
0070  08 5d d0 dd 7f d9 c2 64  36 5f e9 80 1a 7f 6c 8f   .]......d 6_....l.
0080  c1 9b 66 c8 f7 e9 80 62  52 ae d5 43 3a 4e 4c 94   ..f....b R..C:NL.
0090  9f 2f cd d1 96 2e 4c 90  b8 a7 8e 96 de e6 f1 c7   ./....L. ........
00a0  b1 83 57 25 b3 0d 46 bc  69 3b 0a 06 0c 00 00 00   ..W%..F. i;......
```

Frame (194 bytes) | Decrypted data (64 bytes) | Decrypted stub data (64 bytes) | Decrypted NTLMSSP Verifier (12 bytes)

File: "/home/gd/clusapi/failover_... | Packets: 6434 · Displayed: 2077 (32.3%) · Load time: 0:00.194 | Profile: Default

CLUSAPI architecture

- **Objects in the interface are categorized as:**
  - Cluster
  - Nodes
  - Resources
  - ResourceTypes
  - Groups
  - Networks
  - Interfaces
  - Registry
- **Operations are often implemented as Controls, allowing easy additions to the API**

CLUSAPI architecture

- Methods for manipulating Cluster Nodes:
- clusapi_OpenNode{Ex}
- clusapi_GetNodeState
- clusapi_GetNodeId
- clusapi_PauseNode{Ex}
- clusapi_ResumeNode
- clusapi_NodeControl
- clusapi_CloseNode
- And many, many more.

CLUSAPI architecture

- Methods for manipulating Cluster Nodes:
- clusapi_OpenNode{Ex}
- clusapi_GetNodeState
- clusapi_GetNodeId
- clusapi_PauseNode{Ex}
- clusapi_ResumeNode
- clusapi_NodeControl
- clusapi_CloseNode
- And many, many more.

```
CLUSCTL_NODE_UNKNOWN                          = 0x04000000,
CLUSCTL_NODE_GET_CHARACTERISTICS              = 0x04000005,
CLUSCTL_NODE_GET_FLAGS                        = 0x04000009,
CLUSCTL_NODE_GET_NAME                         = 0x04000029,
CLUSCTL_NODE_GET_ID                           = 0x04000039,
CLUSCTL_NODE_GET_CLUSTER_SERVICE_ACCOUNT_NAME =
0x04000041,
CLUSCTL_NODE_ENUM_COMMON_PROPERTIES           = 0x04000051,
CLUSCTL_NODE_GET_RO_COMMON_PROPERTIES         = 0x04000055,
CLUSCTL_NODE_GET_COMMON_PROPERTIES            = 0x04000059,
CLUSCTL_NODE_SET_COMMON_PROPERTIES            = 0x0440005E,
CLUSCTL_NODE_VALIDATE_COMMON_PROPERTIES       = 0x04000061,
CLUSCTL_NODE_ENUM_PRIVATE_PROPERTIES          = 0x04000079,
CLUSCTL_NODE_GET_RO_PRIVATE_PROPERTIES        = 0x0400007D,
CLUSCTL_NODE_GET_PRIVATE_PROPERTIES           = 0x04000081,
CLUSCTL_NODE_SET_PRIVATE_PROPERTIES           = 0x04400086,
CLUSCTL_NODE_VALIDATE_PRIVATE_PROPERTIES      = 0x04000089
```

CLUSAPI architecture

- “Property Lists” encapsulate registry content everywhere

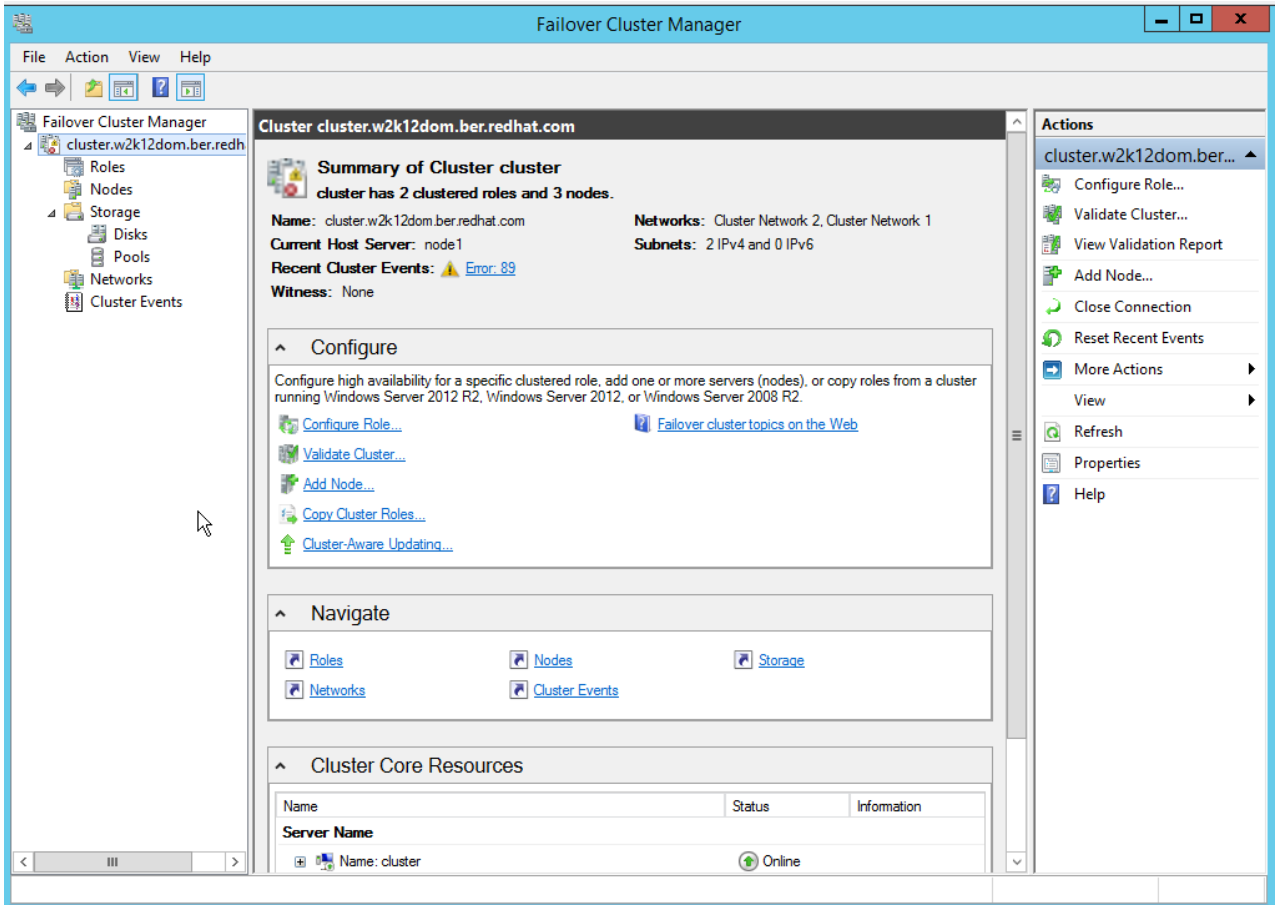
```
&list: struct clusapi_PROPERTY_LIST
          propertyCount            : 0x0000000c (12)
          propertyValues: ARRAY(12)
              propertyValues: struct clusapi_propertyValue
                  syntax_name              : CLUSPROP_SYNTAX_ENDMARK (0)
                  size                     : 0x00000000 (0)
                  buffer                   : 'NodeName'
                  padding                  : DATA_BLOB length=0
                  PropertyValues: struct clusapi_propertyValues
                      Syntax                   : CLUSPROP_SYNTAX_LIST_VALUE_SZ (65539)
                      Size                     : 0x00000018 (24)
                      Buffer                   : DATA_BLOB length=24
  [0000] 43 00 54 00 44 00 42 00   5F 00 4E 00 4F 00 44 00   C.T.D.B. _.N.O.D.
  [0010] 45 00 5F 00 31 00 00 00                              E._.1...
                      Padding                  : DATA_BLOB length=0
                  end_mark                 : CLUSPROP_SYNTAX_ENDMARK (0)
              propertyValues: struct clusapi_propertyValue
                  syntax_name              : CLUSPROP_SYNTAX_ENDMARK (0)
                  size                     : 0x00000000 (0)
                  buffer                   : 'NodeHighestVersion'
                  padding                  : DATA_BLOB length=0
                  PropertyValues: struct clusapi_propertyValues
                      Syntax                   : CLUSPROP_SYNTAX_LIST_VALUE_DWORD (65538)
                      Size                     : 0x00000004 (4)
                      Buffer                   : DATA_BLOB length=4
  [0000] 80 25 08 00                                          .%..
                      Padding                  : DATA_BLOB length=0
                  end_mark                 : CLUSPROP_SYNTAX_ENDMARK (0)
  ...
```

CLUSAPI frontends on Windows

- Power Shell commands
  - “Get-Command -Module FailoverClusters” lists 84 different cluster commands
  - Failover Cluster Cmdlets in Windows PowerShell https://technet.microsoft.com/en-us/library/hh847239.aspx
  - Allows full remote management of clusters

CLUSAPI frontends on Windows

- “Failover Cluster Manager”



Can we implement this in Samba as well?

CLUSAPI implementation problems

- **“Failover Cluster Manager” on Windows insists on contacting DCOM interfaces which Samba currently does not support**

CLUSAPI implementation problems

- ~~**“Failover Cluster Manager” on Windows insists on contacting DCOM interfaces which Samba currently does not support**~~

CLUSAPI implementation problems

- ~~"Failover Cluster Manager" on Windows insists on contacting DCOM interfaces which Samba currently does not support~~
- Namespace:
  - Windows has Cluster name and Node names
  - Samba has only one name on all nodes
  - Workaround: compose virtual node names: CTDB_NODE_$NODEID
- Requirement of asynchronous DCE/RPC infrastructure for notifications
  - Mostly required for the Failover Cluster Manager

CLUSAPI implementation problems

- Omnipresent exposure of registry as configuration state in the interface
  - Windows keeps entire cluster and node state in the registry
- Registry synchronization
  - Windows: non-replicated Windows registry, each node independent
  - Samba: registry is identical on all nodes (via ctdb)
- Ignoring the exact layout and content of cluster registry as on Windows leads to immediate crashes of the Failover Cluster Manager

CLUSAPI registry view (W2K12R2)

Registry Editor

File Edit View Favorites Help

- Cluster
  - Checkpoints
  - Dependencies
  - Groups
    - 19d652e8-a925-408d-9564-821ed3d4dc98
    - 39183aba-37b0-4685-a417-2e3beae7fa49
    - 5fea2112-6434-415a-a354-23722539a0f2
    - 7c8eecc0-b981-4547-9487-673e4db41047
    - f30e76bb-f01e-4d3e-9a14-6458b2e05e63
    - f4eb128f-1e54-495d-bc42-9bc1f968d310
  - LocalIdentity
  - NetworkInterfaces
    - 538e4b05-a2cd-4c97-8c9d-1014d957c718
      - AddressPrefixList
    - 5ca7dd19-de44-4db2-bf33-b84454695cf4
      - AddressPrefixList
    - 5d7d9309-df48-469c-b0d8-1a3fa3849e3d
      - AddressPrefixList
    - a1008f5a-8257-44bd-a833-6048276b59cd
    - f5ffb65d-e4c2-43aa-8229-587f535ca285
  - Networks
    - 28c3b898-a49e-4f02-a368-d353384491c6
      - PrefixList
        - 0000
    - 9a11c763-b9a7-4ff6-ac29-3ba88df1b478
      - PrefixList
        - 0000
  - Nodes
    - 1
    - 2
    - 3
  - Quorum
  - Resources
    - 19c240d4-a322-4475-9aa8-afd395595db9
      - Parameters
      - Shares
    - 21a5bb4a-b064-46c7-a814-206e86ced3f7

Name	Type	Data
(Default)	REG_SZ	(value not set)
AdminAccessPo...	REG_DWORD	0x00000001 (1)
AvailableStorage	REG_SZ	f30e76bb-f01e-4d3e-9a14-6458b2e05e63
BackupInProgress	REG_DWORD	0x00000000 (0)
ClusterGroup	REG_SZ	5fea2112-6434-415a-a354-23722539a0f2
ClusterInstanceID	REG_SZ	8d655874-d47d-4f4e-b919-2bace184d5d5
ClusterName	REG_SZ	cluster
ClusterNameRes...	REG_SZ	35a9a63e-c784-4247-8ecf-29a02ead7036
FromNode	REG_SZ	NODE1
LastOperation	REG_SZ	CreateSubkey
LastUpdated	REG_SZ	2/17/2015 4:04:53 PM
OS	REG_SZ	Windows Server 2012 R2 Datacenter
OSVersion	REG_SZ	6.3 (9600)
PaxosTag	REG_SZ	403:403:5816
PrimordialPoolId	REG_SZ	{885e9a73-cdcd-42c2-96b6-a1571e58ccaa}
QuorumArbitrat...	REG_DWORD	0x0000005a (90)
RecentEventsRe...	REG_QWORD	0x1d04c66cb5cfb41 (130688393575988033)
Security Descrip...	REG_BINARY	01 00 14 80 8c 00 00 00 9c 00 00 00 14 00 00 00 1c 00...
SharedVolumes...	REG_SZ	C:\ClusterStorage
SubsystemId	REG_SZ	{6924f3ee-14c2-46aa-abda-221c19436049}

Computer\HKEY_LOCAL_MACHINE\Cluster

CLUSAPI support in Samba

- Basic CLUSAPI v3 implementation in Samba
- Several Cluster Power Shell commands already work against Samba
- “Failover Cluster Manager” can do basic operations on CTDB Nodes

CLUSAPI DEMO

Further reading

- **Microsoft Protocol Documentation:**
  - **MS-SWN: Service Witness Protocol**
  - **MS-CMRP: Failover Cluster Management Protocol**
- **SMB 2.x and SMB 3.0 Timeouts in Windows**
  http://blogs.msdn.com/b/openspecification/archive/2013/03/27/smb-2-x-and-smb-3-0-timeouts-in-windows.aspx
- **Failover Cluster Cmdlets in Windows PowerShell**
  https://technet.microsoft.com/en-us/library/hh847239.aspx
- **Samba Wiki**
  https://wiki.samba.org/index.php/Samba3/SMB2#Witness_Notification_Protocol

Questions and answers

- Mail gd@samba.org
- #samba-technical on irc.freenode.net
- WIP branches:
  - https://git.samba.org/?p=gd/samba/.git;a=shortlog;h=refs/heads/master-witness
  - https://git.samba.org/?p=gd/samba/.git;a=shortlog;h=refs/heads/master-clusapi
  - https://git.samba.org/?p=gd/wireshark/.git;a=shortlog;h=refs/heads/master-clusapi

Thank you for your attention!

www.redhat.com
www.samba.org

<gd@samba.org>