SMB3 in Samba

Multi-Channel and Beyond

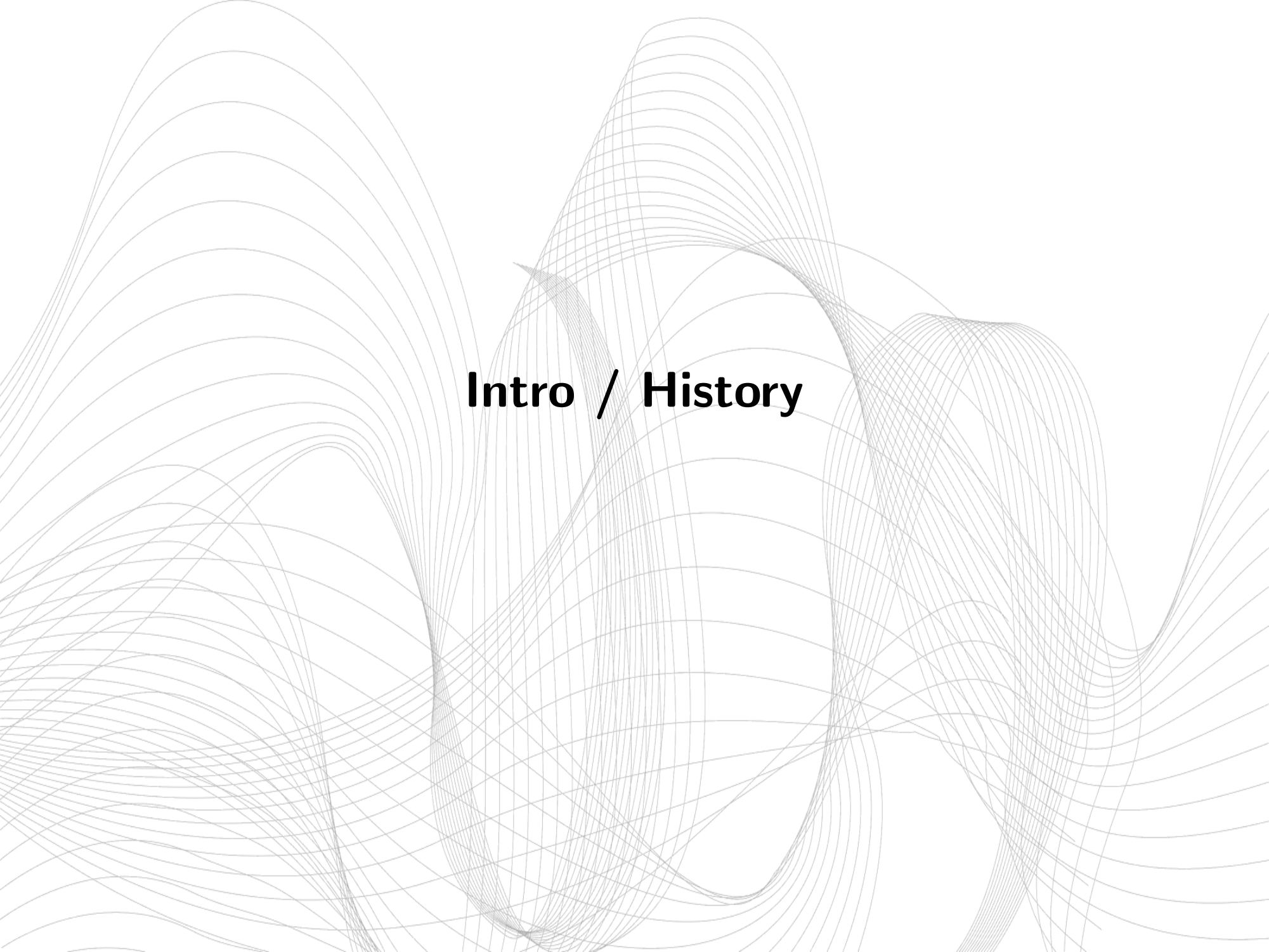
Michael Adam

Red Hat / samba.org

2016-04-20

agenda

- History of SMB
- History of Samba
- SMB 2+
- SMB 2+ in Samba
- SMB3 Multi-Channel
- Outlook: SMB3 over RDMA
- Outlook: SMB3 Clustering/Witness
- Outlook: SMB3 Persistent Handles



Intro / History

SMB - the alien protocol

- SMB - Server Message Block
- 1983: created by Barry Feigenbaum, IBM
  Turn DOS INT 21h local file access into network
- Microsoft:
  - Lan Manager (from 1990)
  - Windows for Workgroups (from 1992)
- On top of NetBIOS, TCP port 139
- from Windows 2000: directly on TCP port 445

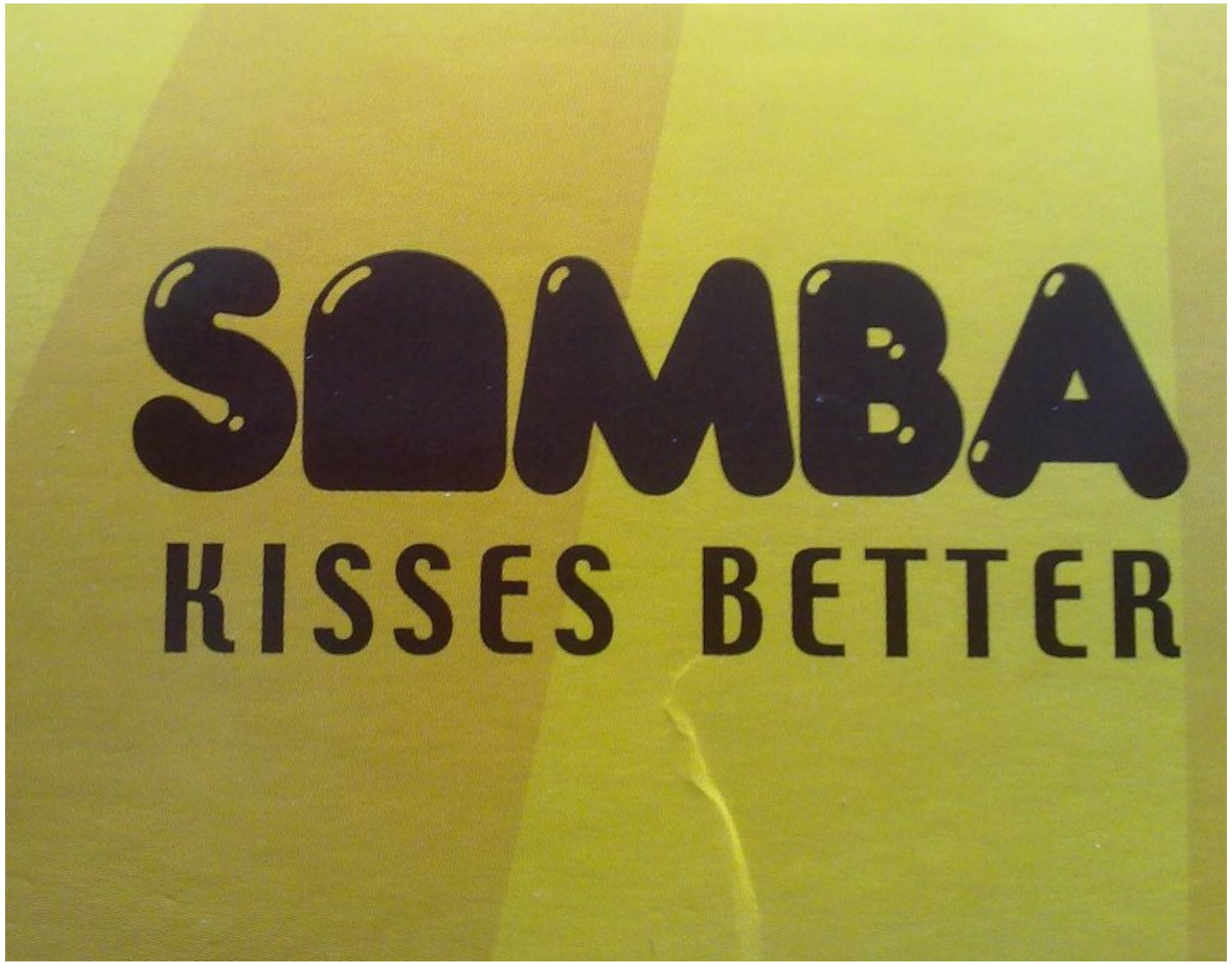
SMB versions > 1

- SMB 2.0: 2006 - Windows Vista
- SMB 2.1: 2009 - Windows 7/Server 2008R2
- SMB 3.0: 2012 - Windows 8/Server 2012
- SMB 3.0.2: 2014 - Windows 8.1/Server 2012R2
- SMB 3.1.1: 2015 - Windows 10/Server 2016

Enter Samba ...

- ... implements SMB ...
- ... *old* Open Source project ...
- ... opens windows to a wider world ... ☺
- ...

Samba...



Samba...

According to openhub.net, Samba

"...has had 101,614 commits made by 363 contributors representing 1,637,229 lines of code"

- present on millions of NAS devices and routers
- one of the oldest OSS projects (24 years)
- large codebase and small but very active development team

Samba - History

- 1992/01: start of the project
- 1.5: 1993/12: (nbserver)
- 1.9.16: 1996/05: CVS, Samba Team
- 2.0: 1999/01: domain-member, +SWAT
- 2.2: 2001/04: NT4-DC
- 3.0: 2003/09: AD-member, Samba4 project started
- 3.2: 2008/07: GPLv3, experimental clustering
- 3.3: 2009/01: clustering [with CTDB]
- 3.4: 2009/07: merged S3+S4 code
- 3.5: 2010/03: experimental SMB 2.0
- 3.6: 2011/09: SMB 2.0
- 4.0: 2012/12: AD/DC, SMB 2.0 durable handles, 2.1, 3.0
- 4.1: 2013/10: stability
- 4.2: 2015/03: AD trusts, SMB2.1 leases, perf, include CTDB
- 4.3: 2015/09: spotlight, new FileChangeNotify, SMB 3.1.1
- 4.4: 2016/03: Multi-Channel core, ...

Samba - Today

- Performant, scalable SMB file server
  ⇒ Ongoing SMB3 implementation
- Active Directory domain member with `winbindd`
  ⇒ flexible, performant, clusterable
- Full Active Directory Domain Controller
  (Kerberos KDC, LDAP, DNS, Trusted Domains, etc)
  "AWS Directory Service" is powered by Samba AD
- Established SMB clients for Linux:
  cifs.ko, libsmbclient (nautilus, dolphin, konqueror)
- Comprehensive testsuite
  ⇒ wrappers now published outside of Samba: cwrap.org
- IDL compiler, autogenerated DCE/RPC code
  ⇒ another 1,141,095 lines of code
- Powerful python(3) bindings, partly autogenerated

SMB3

SMB3

SMB3 (2012) introduced SMB clustering:

- Clustering - Witness
- Continuous Availability - Persistent Handles
- Scale Out

Additionally:

- Transport encryption
- Multi-Channel
- RDMA transport (SMB Direct)

SMB Features - in Samba

- SMB 2.0:
  - durable file handles [4.0]
- SMB 2.1:
  - multi-credit / large mtu [4.0]
  - dynamic reauthentication [4.0]
  - leasing [4.2]
  - resilient file handles [PoC]
- SMB 3.0:
  - new crypto (sign/encrypt) [4.0]
  - secure negotiation [4.0]
  - durable file handles v2 [4.0]
  - persistent file handles [design/PoC]
  - multi-channel [4.4 (experimental)]
  - SMB direct [design]
  - cluster features [design]
    - witness [WIP+]
- SMB 3.0.2: [4.3]
- SMB 3.1.1:
  - negotiate contexts, preauth: [4.3]

Multi-Channel

Multi-Channel - General

multiple transport connections in one SMB(3) session

- **channel**: transport connection bound to a session
- client decides which connections to bind and to use
- session is valid as long as at least one channel is intact

two purposes

1. increase throughput:
   - use multiple connections of same type
2. improve fault tolerance:
   - channel failure: replay/retry detection

Multi-Channel - General

use case: channels of different type/quality

- use only the channels of best quality
- fall back to inferior channels if superior ones fail
- e.g.: laptop switching between WiFi and LAN (?)

Multi-Channel - Windows/Protocol

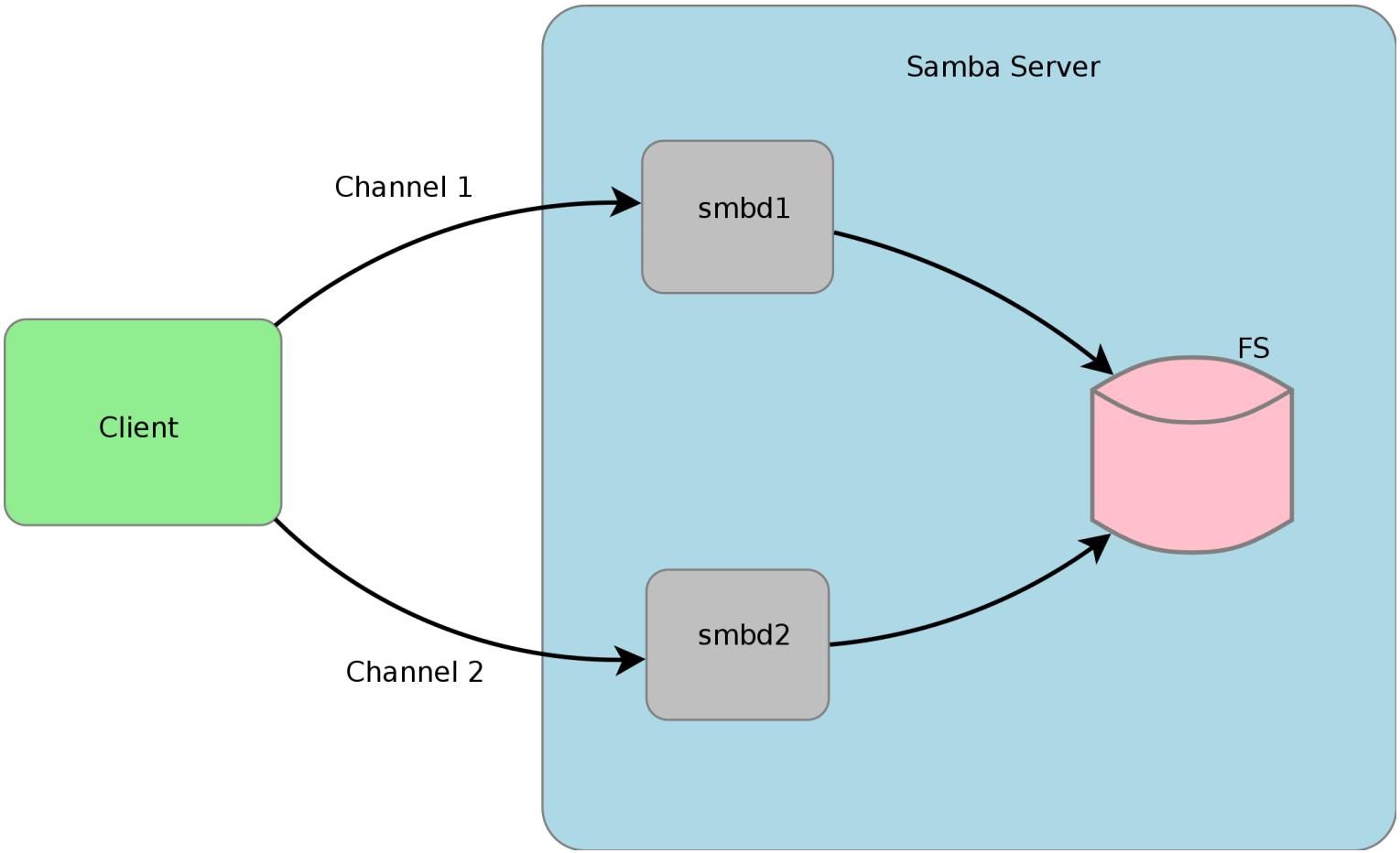
1. establish initial session on TCP connection
2. find interfaces with interface discovery: `FSCTL_QUERY_NETWORK_INTERFACE_INFO`
3. bind additional TCP (or later RDMA) connection (channel) to established SMB3 session (*session bind*)
4. Windows: uses connections of same (and best) quality
5. Windows: binds only to a single node
6. replay / retry mechanisms, epoch numbers

Multi-Channel ∈ Samba

samba/smbd: multi-process

- **Originally:** process ⇔ TCP connection
- **Idea:** transfer new TCP connection to existing smbd
- **How?** ⇒ use fd-passing (sendmsg/recvmsg)
- **When?**
  - *Natural choice*: at SessionSetup (Bind)
  - Idea: as early as possible, based on ClientGUID
    ⇒ per ClientGUID single process model

Multi-Channel ∈ Samba

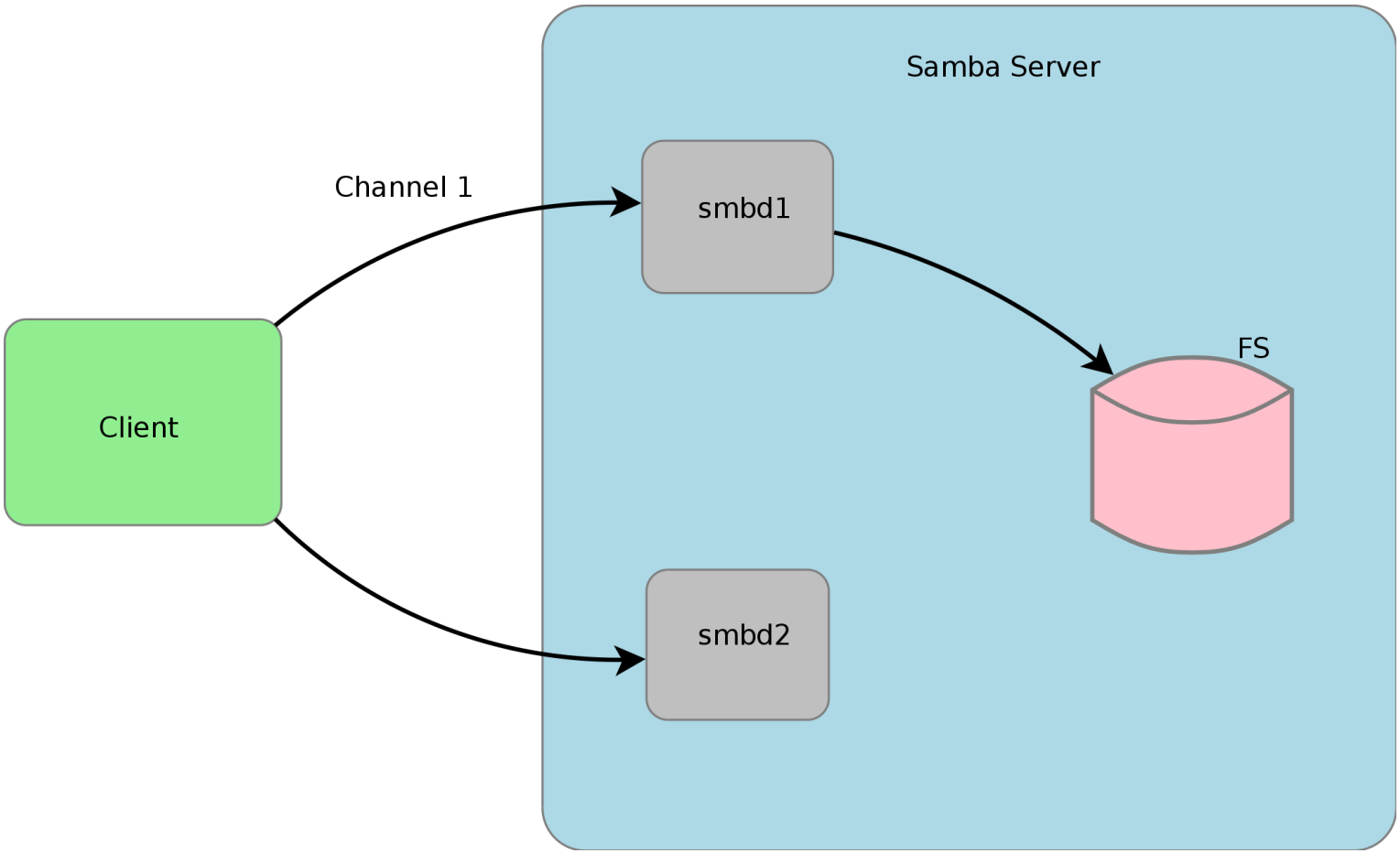


Multi-Channel ∈ Samba

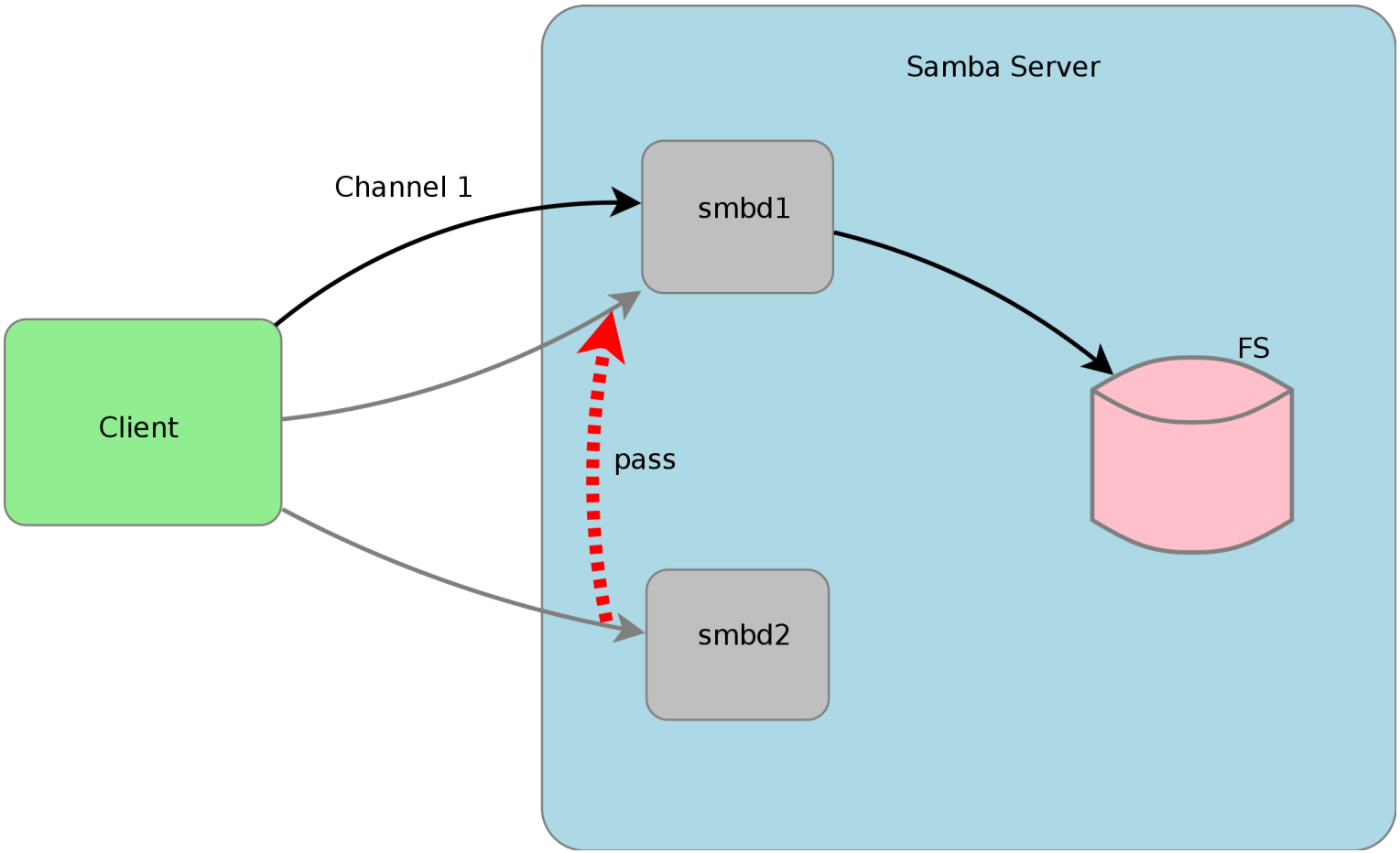
samba/smbd: multi-process

- **Originally:** process ⇔ TCP connection
- **Idea:** transfer new TCP connection to existing smbd
- **How?** ⇒ use fd-passing (sendmsg/recvmsg)
- **When?**
  - *Natural choice*: at SessionSetup (Bind)
  - Idea: as early as possible, based on ClientGUID
    ⇒ per ClientGUID single process model

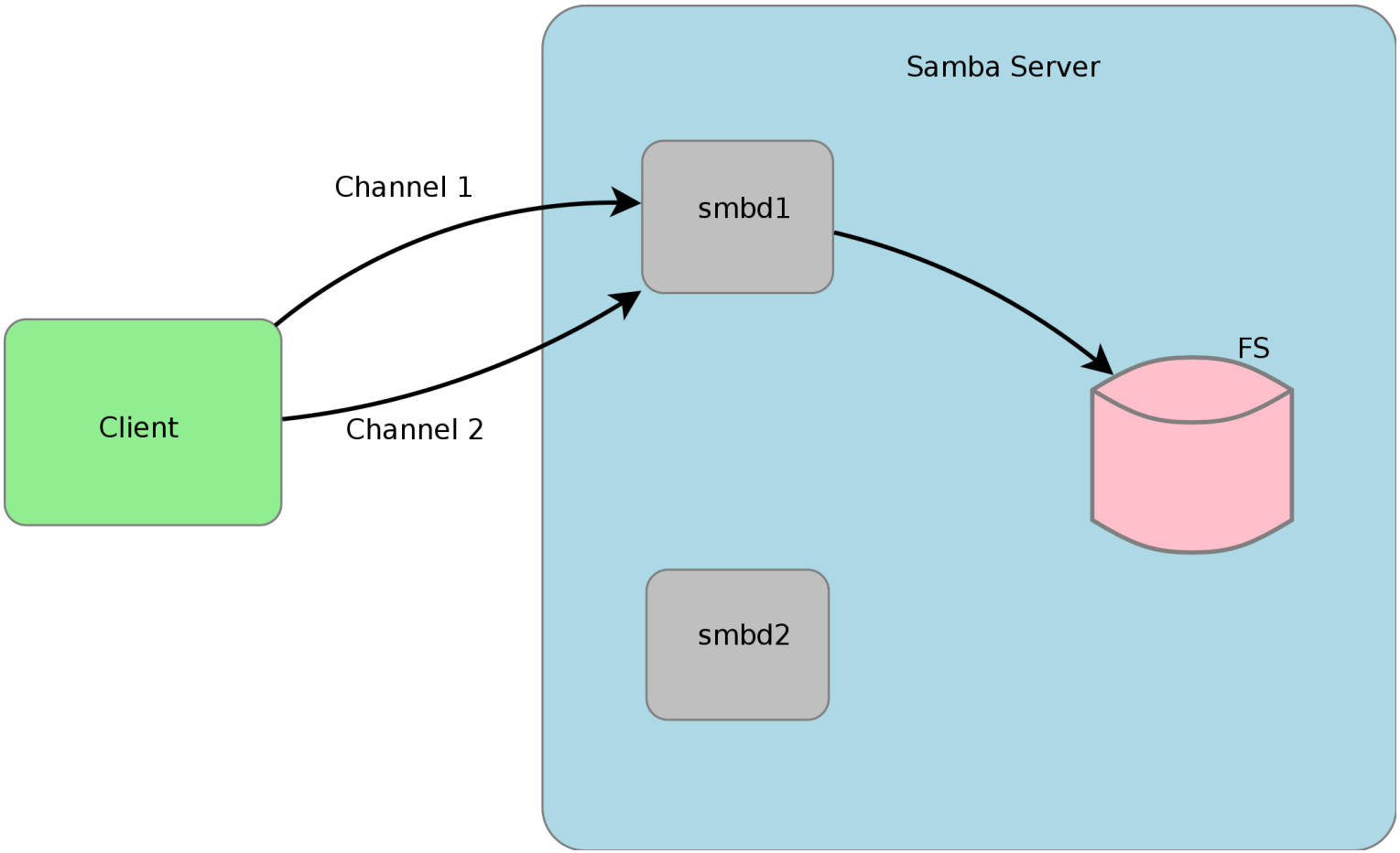
Multi-Channel ∈ Samba



Multi-Channel ∈ Samba



Multi-Channel ∈ Samba



Multi-Channel ∈ Samba

samba/smbd: multi-process

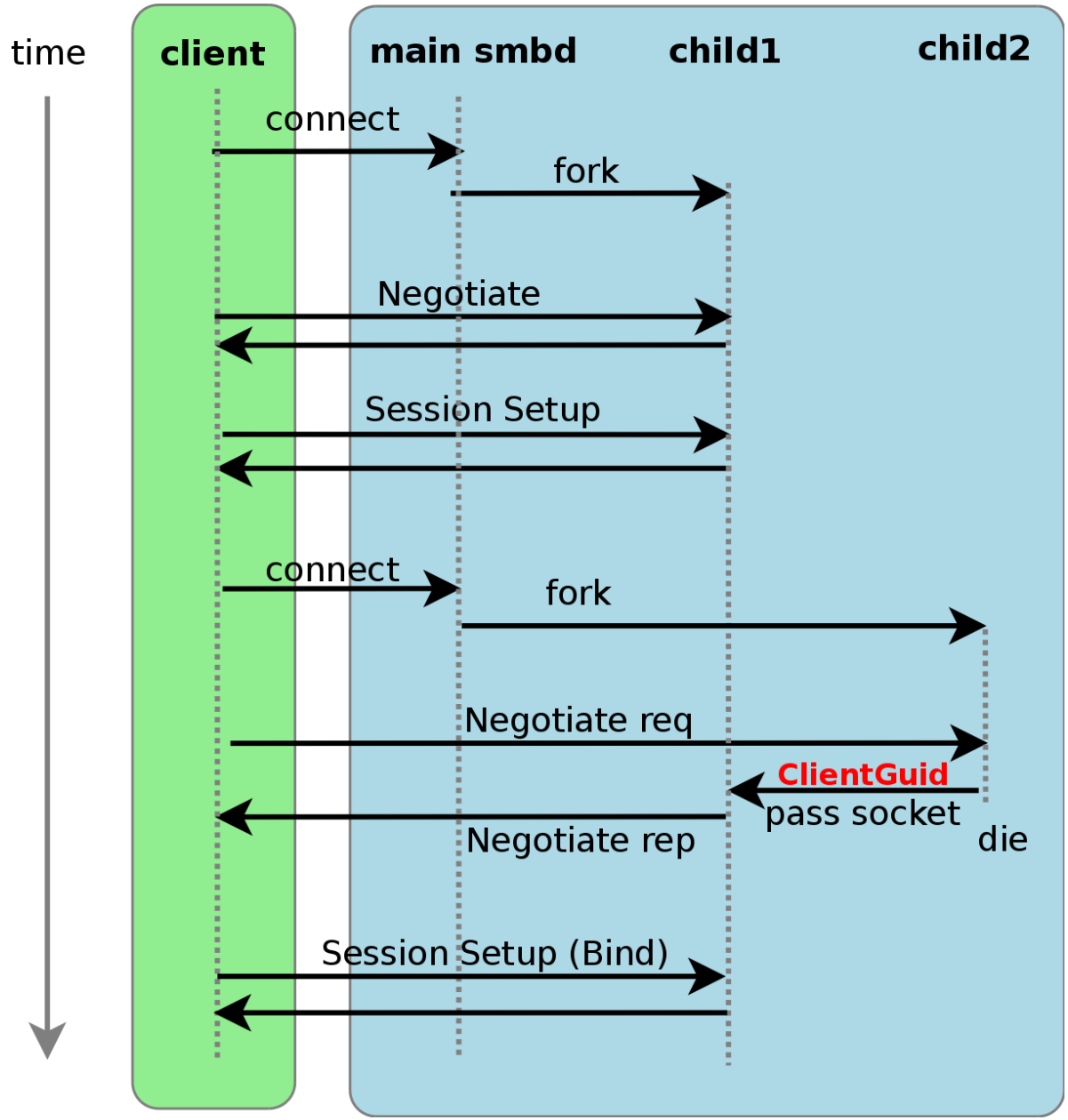
- **Originally:** process ⇔ TCP connection
- **Idea:** transfer new TCP connection to existing smbd
- **How?** ⇒ use fd-passing (sendmsg/recvmsg)
- **When?**
  - *Natural choice*: at SessionSetup (Bind)
  - Idea: as early as possible, based on ClientGUID
    ⇒ per ClientGUID single process model

Multi-Channel ∈ Samba

samba/smbd: multi-process

- **Originally:** process ⇔ TCP connection
- **Idea:** transfer new TCP connection to existing smbd
- **How?** ⇒ use fd-passing (sendmsg/recvmsg)
- **When?**
  - *Natural choice*: at SessionSetup (Bind)
  - Idea: as early as possible, based on ClientGUID
    ⇒ per ClientGUID single process model

Multi-Channel ∈ Samba : pass by ClientGUID



Multi-Channel ∈ Samba : pass by ClientGUID

Wait a minute - what about performance?

- Single process...
- But we use short-lived worker-pthreads for I/O ops!
- Benchmarks and tunings still to be done.

Multi-Channel ∈ Samba : Status

1. messaging rewrite using unix dgm sockets with sendmsg [DONE,4.2]
2. add fd-passing to messaging [DONE,4.2]
3. preparations in internal structures [DONE,4.4]
4. prepare code to cope with multiple channels [DONE,4.4]
5. implement smbd message to pass a tcp socket [DONE,4.4]
6. transfer connection in Negotiate (by ClientGUID) [DONE,4.4]
7. implement session bind [DONE,4.4]
8. implement channel epoch numbers [DONE,4.4]
9. implement interface discovery [DONE(linux/conf),4.4]
10. implement test cases [WIP(isn't it always?... ☺)]
11. implement fd-passing in socket-wrapper [WIP]
12. implement lease break replay [TODO]

Multi-Channel ∈ Samba : Status

1. messaging rewrite using unix dgm sockets with sendmsg [DONE,4.2]
2. add fd-passing to messaging [DONE,4.2]
3. preparations in internal structures [DONE,4.4]
4. prepare code to cope with multiple channels [DONE,4.4]
5. implement smbd message to pass a tcp socket [DONE,4.4]
6. transfer connection in Negotiate (by ClientGUID) [DONE,4.4]
7. implement session bind [DONE,4.4]
8. implement channel epoch numbers [DONE,4.4]
9. implement interface discovery [DONE(linux/conf),4.4]
10. implement test cases [WIP(isn't it always?... ☺)]
11. implement fd-passing in socket-wrapper [WIP]
12. implement lease break replay [TODO]

Multi-Channel ∈ Samba : Details from smbXsrv.idl

for MSG_SMBXSRV_CONNECTION_PASS

```
typedef struct {
    NTTIME                  initial_connect_time;
    GUID                    client_guid;
    hyper                   seq_low;
    DATA_BLOB               negotiate_request;
} smbXsrv_connection_pass0;
```

Multi-Channel ∈ Samba : Details from `smbXsrv.idl`

layering before

```
smbXsrv_session
  ->smbXsrv_connection
```

layering now

```
smbXsrv_session
  ->smbXsrv_client
    ->smbXsrv_connections
```

Multi-Channel $\in$ Samba: TODOs

- Replay lease breaks upon channel failure (server $\rightarrow$ client)
- teach socket_wrapper fd-passing ($\Rightarrow$ selftest...)
- clustering integration (CTDB)

Multi-Channel ∈ Samba : Clustering/CTDB

Special considerations

- channels of one session only to one node !
- do not bind connections to CTDB public IPs (can move)!
- ⇒ add static IPs on public interfaces
  use these for interface discovery

Multi-Channel ∈ Samba : Clustering/CTDB

Special considerations

- channels of one session only to one node !
- do not bind connections to CTDB public IPs (can move)!
- ⇒ add static IPs on public interfaces
  use these for interface discovery

Multi-Channel Demo

Outlook: SMB Direct

SMB Direct : SMB3 over RDMA

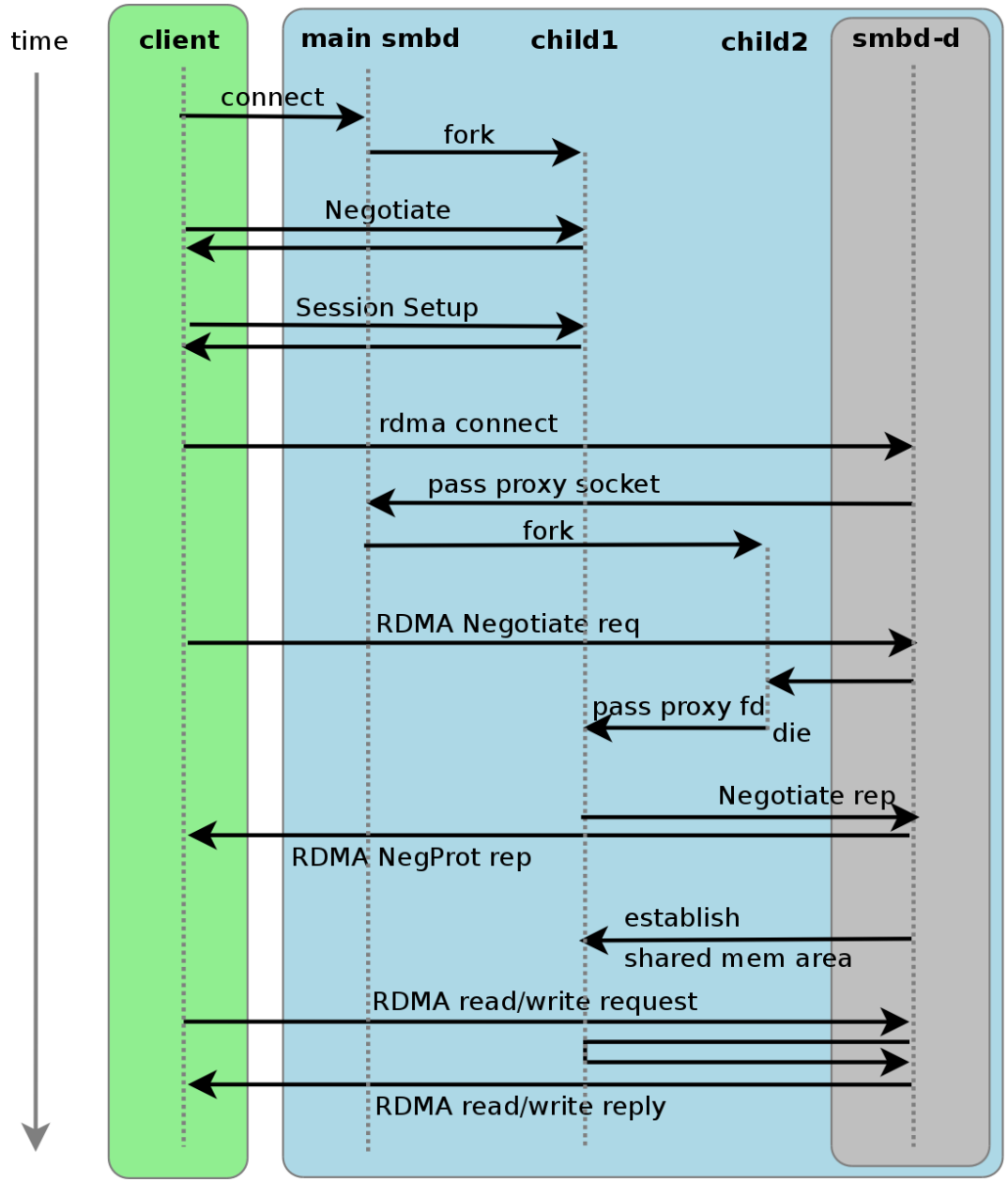
Windows/Protocol

- requires multi-channel
- start with TCP, bind an RDMA channel
- SMB Direct: small wrapper protocol to put SMB into RDMA
- reads and writes use RDMA write/read
- protocol/metadata via send/receive

SMB Direct ∈ Samba

- wireshark dissector: [DONE]
- Samba:
  - prereq: multi-channel [ess.DONE]
  - buffer / transport abstractions [WIP]
- problem with RDMA libraries:
  - not fork safe
  - no fd-passing
- ⇒ central RDMA proxy
  - PoC/dev: user space daemon
  - production: kernel module

SMB Direct ∈ Samba

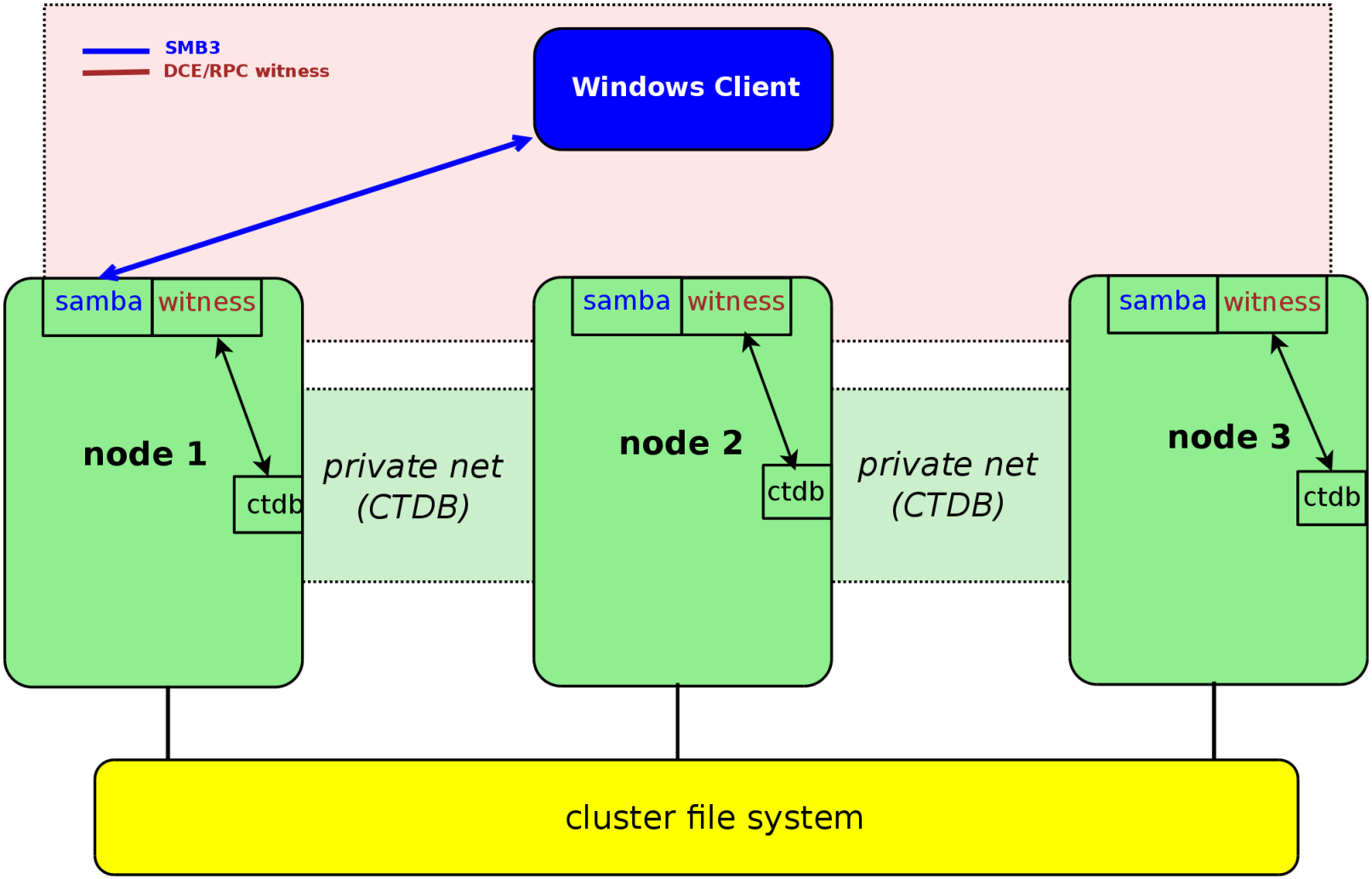


Outlook: clustering / witness

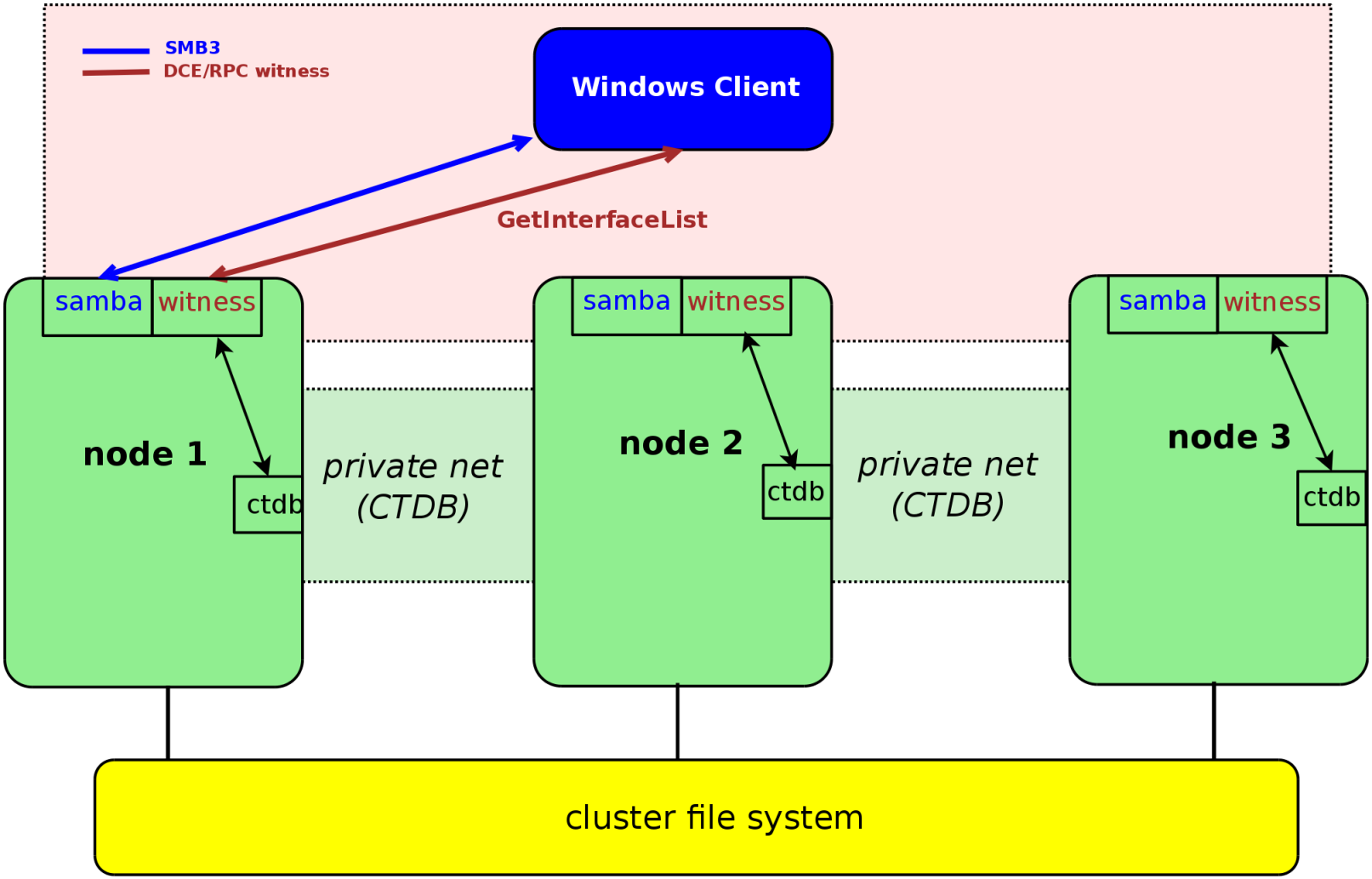
Witness - General

- New DCE/RPC Service to “witness” availability of IPs, shares, ...
- $\Rightarrow$ Faster fail-over of clients in the cluster
- Prompt, explicit, and controlled notifications about failures (CTDB tickle-ACKs are implicit)
- Available since SMB3 (Windows 8 / Windows Server 2012)

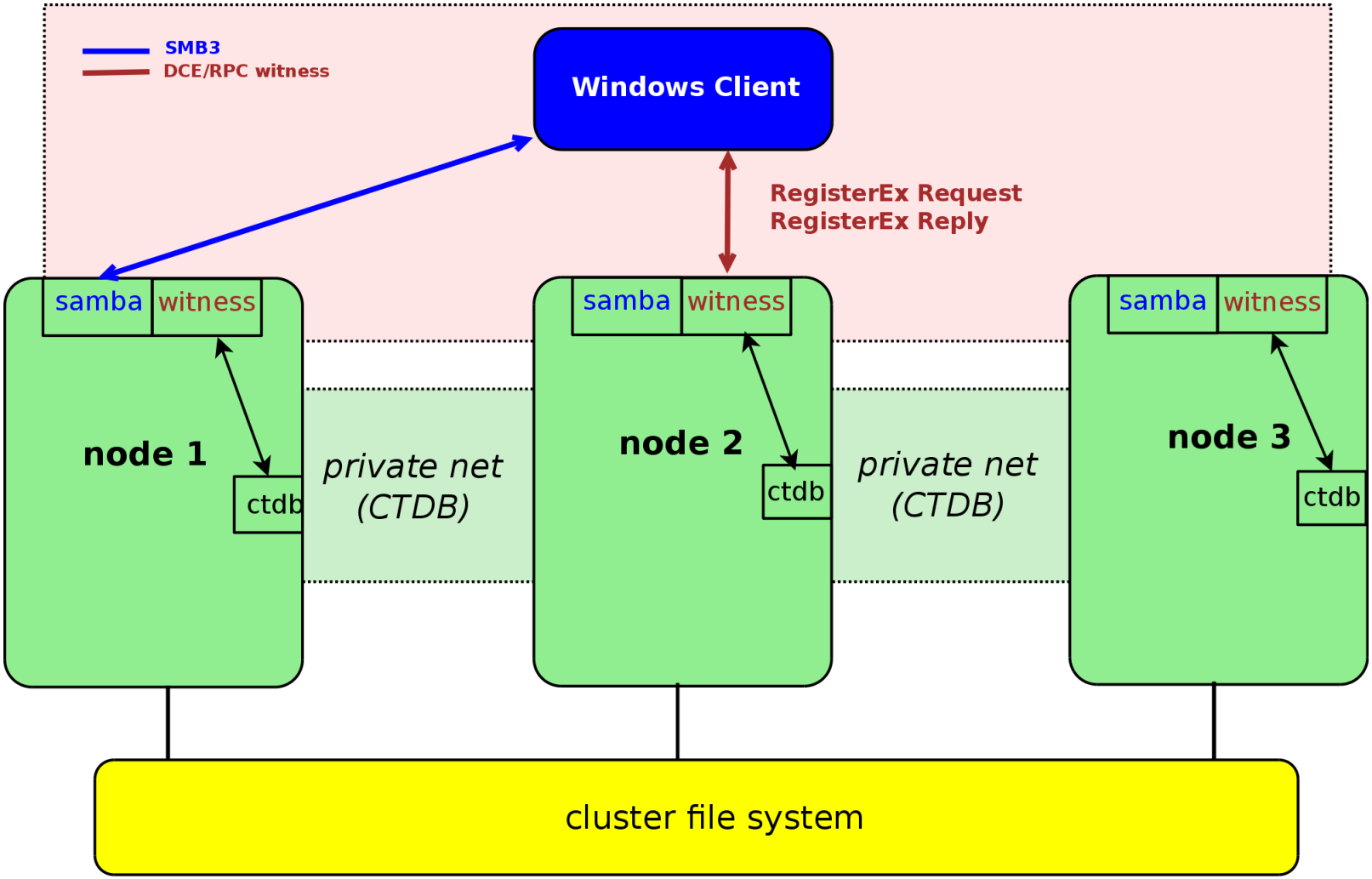
Witness - Failover with SMB3 in a Samba/CTDB cluster



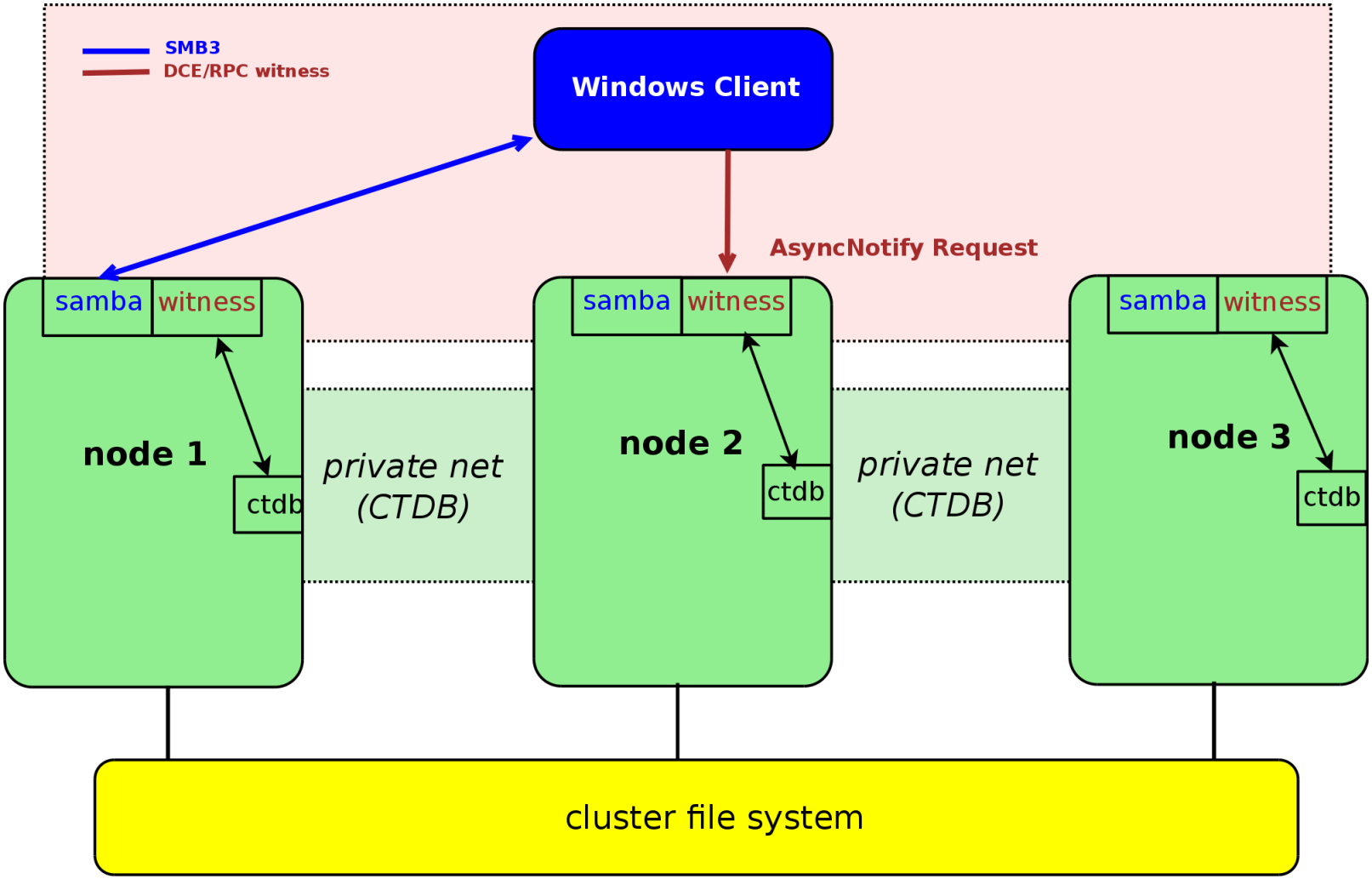
Witness - Failover with SMB3 in a Samba/CTDB cluster



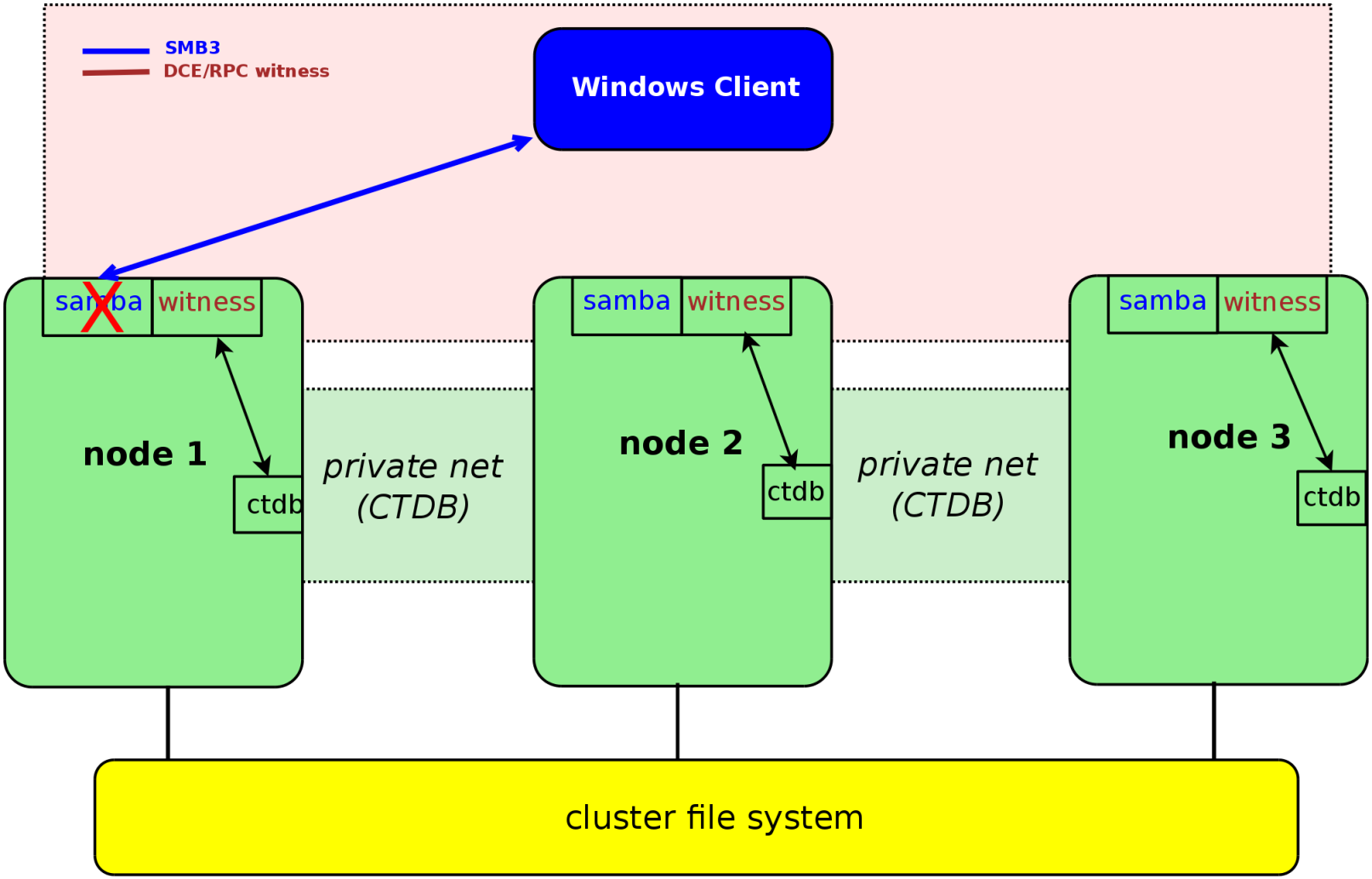
Witness - Failover with SMB3 in a Samba/CTDB cluster



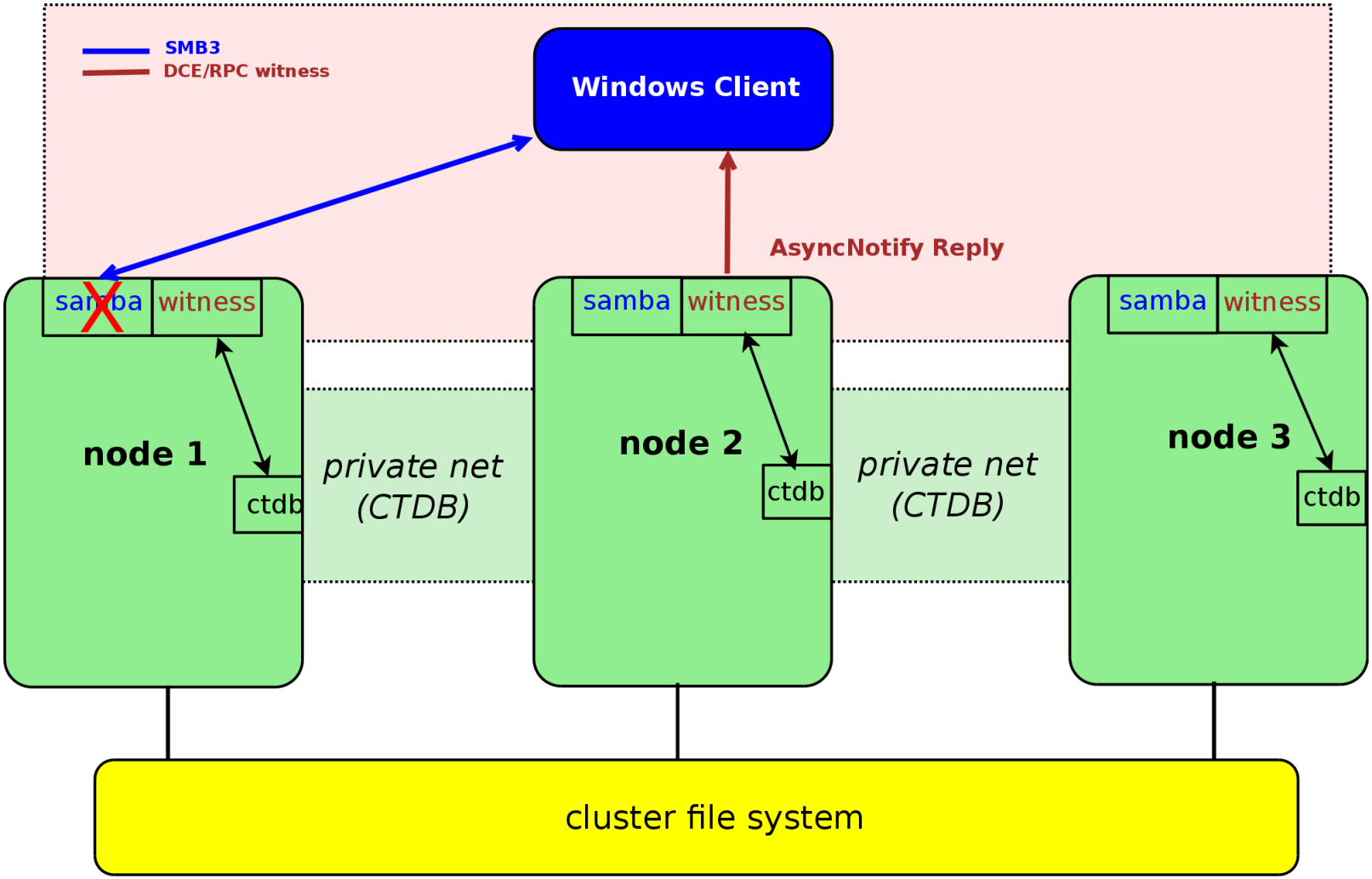
Witness - Failover with SMB3 in a Samba/CTDB cluster



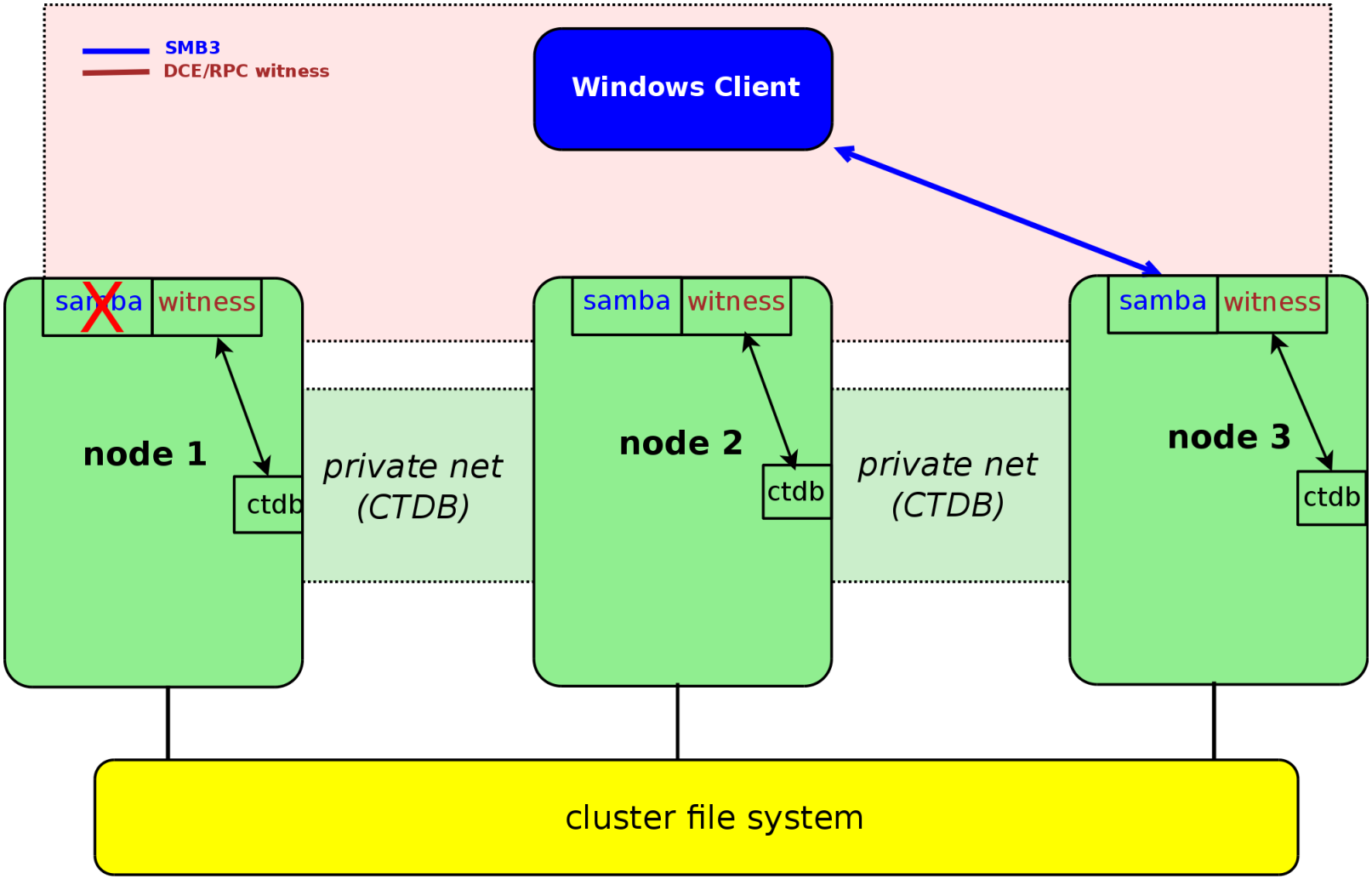
Witness - Failover with SMB3 in a Samba/CTDB cluster



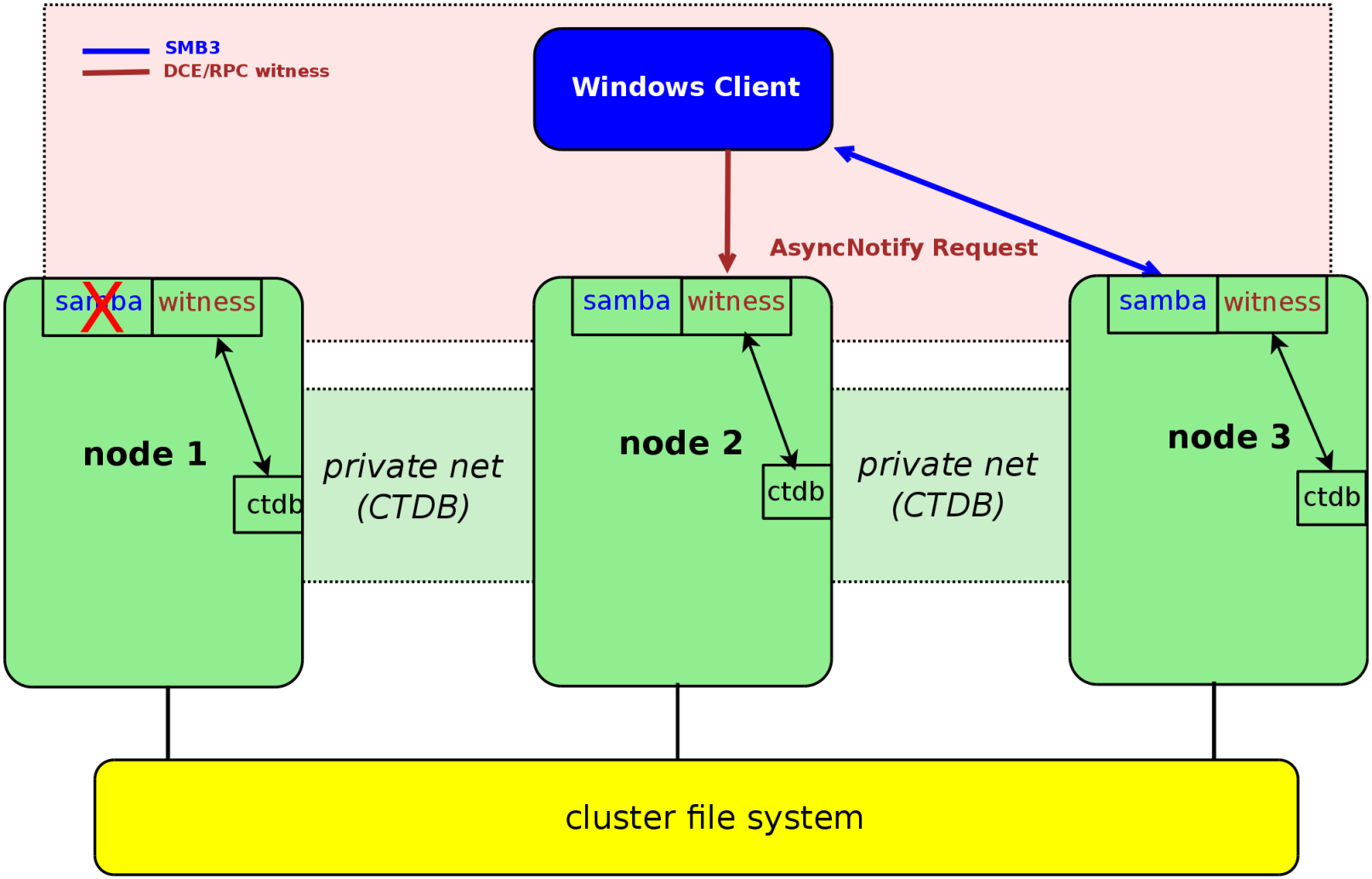
Witness - Failover with SMB3 in a Samba/CTDB cluster



Witness - Failover with SMB3 in a Samba/CTDB cluster



Witness - Failover with SMB3 in a Samba/CTDB cluster



Witness - Samba

Currently under development in Samba

- PoC implementation available
- TODO(wip): new async DCE/RPC infrastructure
- `https://wiki.samba.org/index.php/Samba3/SMB2#Witness_Notification_Protocol`
- WIP branch: `https://git.samba.org/?p=gd/samba/.git;a=shortlog;h=refs/heads/master-witness`

Samba Witness service will cause Windows clients to reconnect...

- when client admin tool is used
- when CTDB (or any other cluster resource control manager) moves resources or IP addresses

Outlook: persistent handles

Persistent File Handles

- available on 'Continuously Available' SMB3 shares
- allows disconnected clients to reconnect
- like durable handles, but with strong guarantees!

Persistent Handles : Challenges

- protocol is easy
- persistence/guarantees are hard
- strategies:
  - filesystem spcific
  - generic, with tdb extensions

Wrapping up...

What's next ?

- SMB3 Multi-Channel: finishing moves
- SMB3 Witness service: async RPC
- SMB3 Persistent Handles / CA
- SMB3 over RDMA (SMB direct)
- Multi-Protocol access (NFS, SMB...)
- SMB2+ Unix Extensions ⇒ See Jeremy's Talk!

Thanks for your attention!

Questions?

obnox@samba.org

obnox@redhat.com

https://git.samba.org/?p=obnox/slides/2016-04-vault.git