SMB3 Multi-Channel in Samba

... Now Really!

Michael Adam

Red Hat / samba.org

sambaXP - 2016-05-11


SAMBA
EXPERIENCE
SAMBA
EXPERIENCE

Introduction


SMB
Sanitätshaus müller betten
CRAFTER

SMB - mini history

- SMB: created around 1983 by Barry Feigenbaum, IBM
- SMB in Lan Manager: around 1990
- SMB in Windows for Workgroups: from 1992
- SMB $\rightarrow$ CIFS: 1996
- SMB on TCP port 445: 2000 - Windows 2000
- 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


EXCELENCIA EM CAFÉ
SAMBA
FLAVOUR
espresso do brasil

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 ChangeNotify, SMB 3.0.2, 3.1.1
- 4.4: 2016/03: SMB3 Multi-Channel (experimental), ...

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 ChangeNotify, SMB 3.0.2, 3.1.1
- 4.4: 2016/03: SMB3 Multi-Channel (experimental), ...

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 ChangeNotify, SMB 3.0.2, 3.1.1
- 4.4: 2016/03: SMB3 Multi-Channel (experimental), ...

Apologies to our friends from Microsoft for writing "Multi-Channel"! ... ☺

But hey... How can we *partly* implement an SMB version?

SMB2 Capabilities - Negotiate

- SMB2_CAP_DFS (3.5, 3.6)
- SMB2_CAP_LEASING (4.2)
- SMB2_CAP_LARGE_MTU (4.0)
- SMB2_CAP_MULTI_CHANNEL (4.4)
- SMB2_CAP_PERSISTENT_HANDLES
- SMB2_CAP_DIRECTORY_LEASING
- SMB2_CAP_ENCRYPTION (4.0)

Other 'optional' SMB2 features

- Some create contexts - ok to ignore, e.g.:
  - durable handles (best-effort concept)
- fsctl/ioctls - ok (?) to return errors, e.g.:
  - FSCTL_QUERY_NETWORK_INTERFACE_INFO
  - FSCTL_LMR_REQ_RESILIENCY

So what's the big deal about SMB3?

SMB3 - what's the big deal?

SMB3 (2012) introduced SMB clustering:

- Clustering - Witness (HA / faster fail-over)
- Continuous Availability - Persistent Handles (guarantees!)
- Scale Out (all-active access)

Additionally:

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

from workstation to server workload

- databases (sql...)
- virtualtization (hyper-v)
- ...

SMB3 - what's the big deal?

SMB3 (2012) introduced SMB clustering:

- Clustering - Witness (HA / faster fail-over)
- Continuous Availability - Persistent Handles (guarantees!)
- Scale Out (all-active access)

Additionally:

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

from workstation to server workload

- databases (sql...)
- virtualtization (hyper-v)
- ...

SMB3 - what's the big deal?

SMB3 (2012) introduced SMB clustering:

- Clustering - Witness (HA / faster fail-over)
- Continuous Availability - Persistent Handles (guarantees!)
- Scale Out (all-active access)

Additionally:

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

from workstation to server workload

- databases (sql...)
- virtualtization (hyper-v)
- ...

SMB3 - what's the big deal?

SMB3 (2012) introduced SMB clustering:

- Clustering - Witness (HA / faster fail-over)
- Continuous Availability - Persistent Handles (guarantees!)
- Scale Out (all-active access)

Additionally:

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

from workstation to server workload

- databases (sql...)
- virtualtization (hyper-v)
- ...

Multi-Channel


NO ALCOHOL
BEYOND THIS
POINT

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

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

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

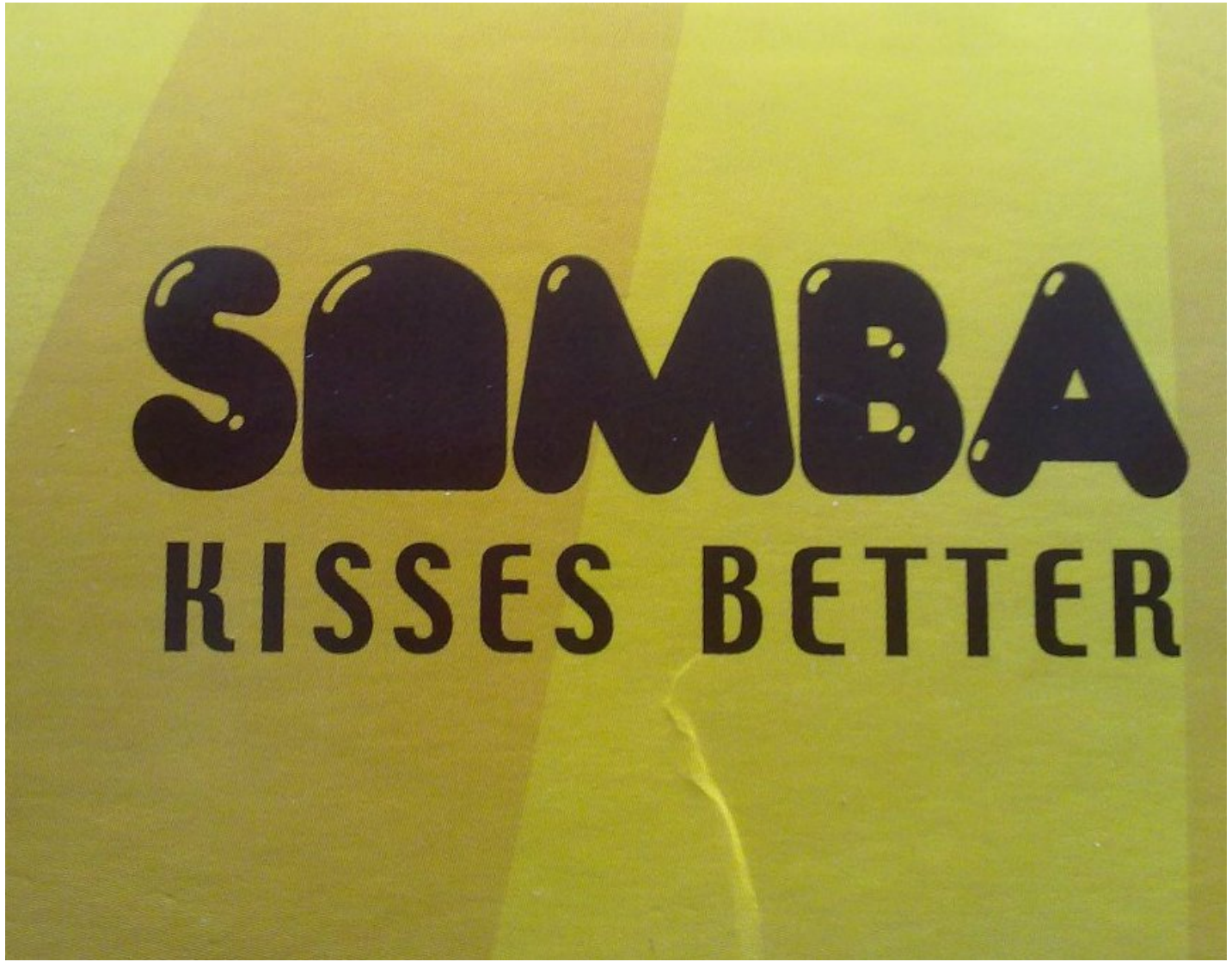
Multi-Channel - General

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, sequence numbers

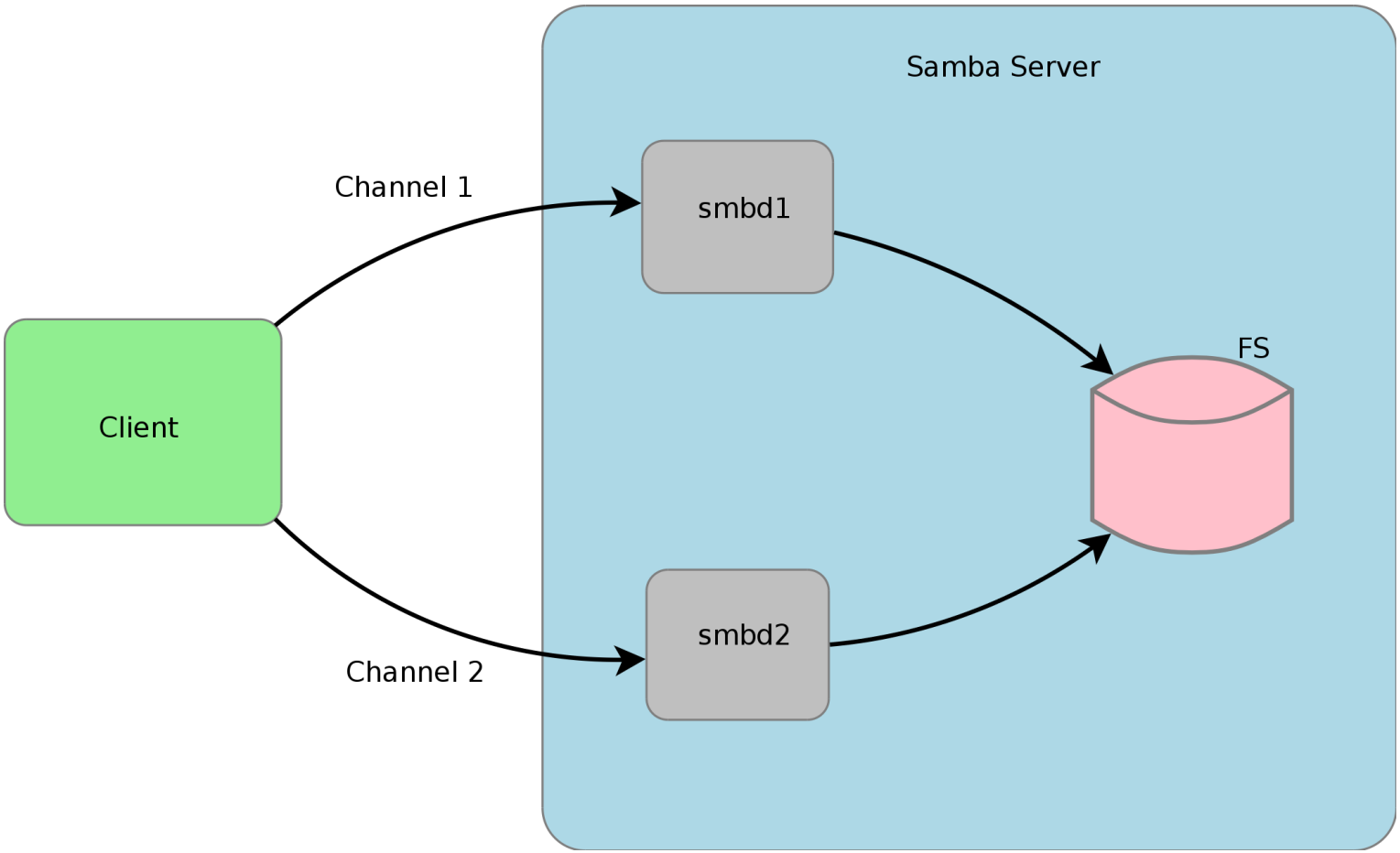

SAMBA
KISSES BETTER

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

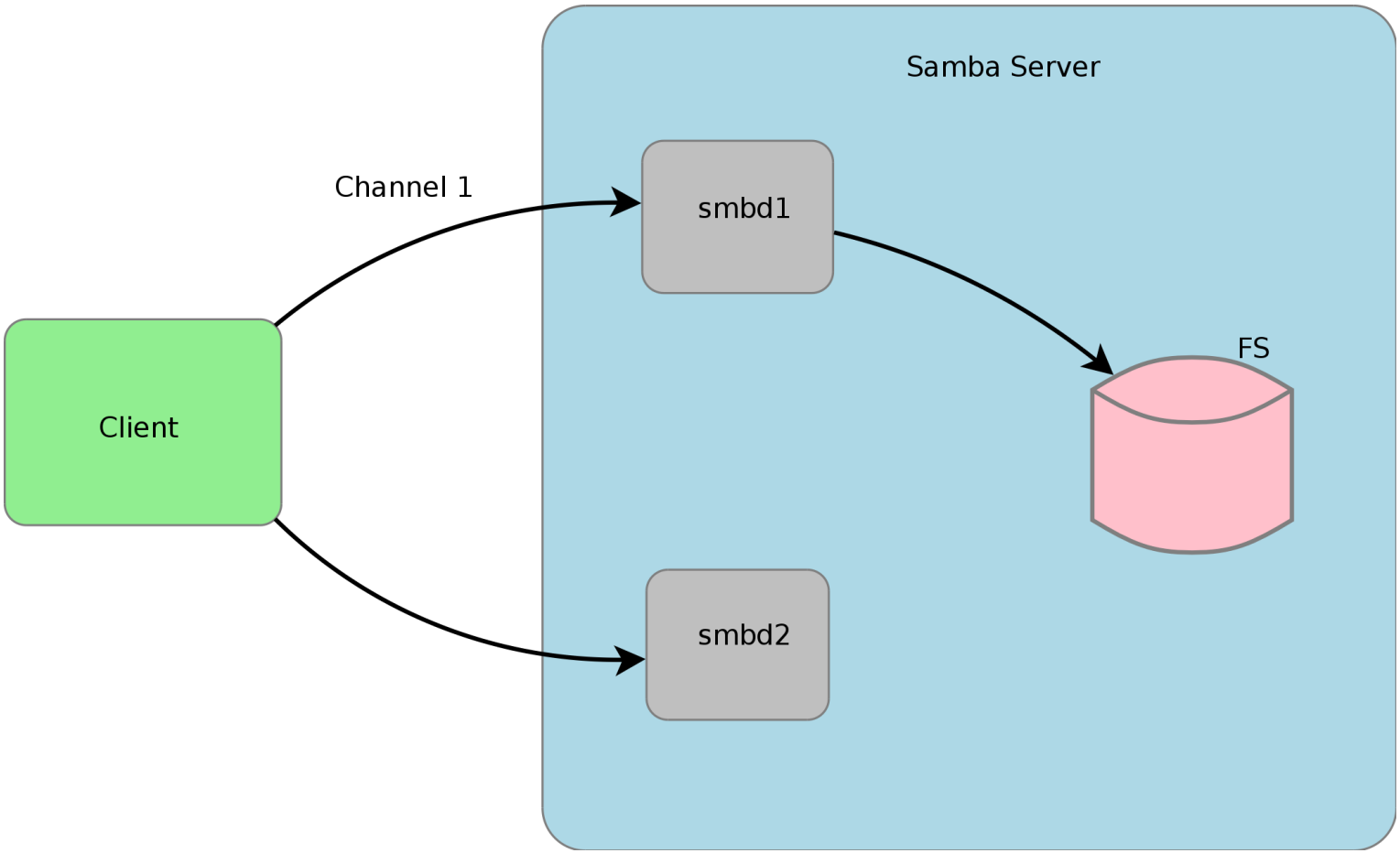


Multi-Channel ∈ Samba

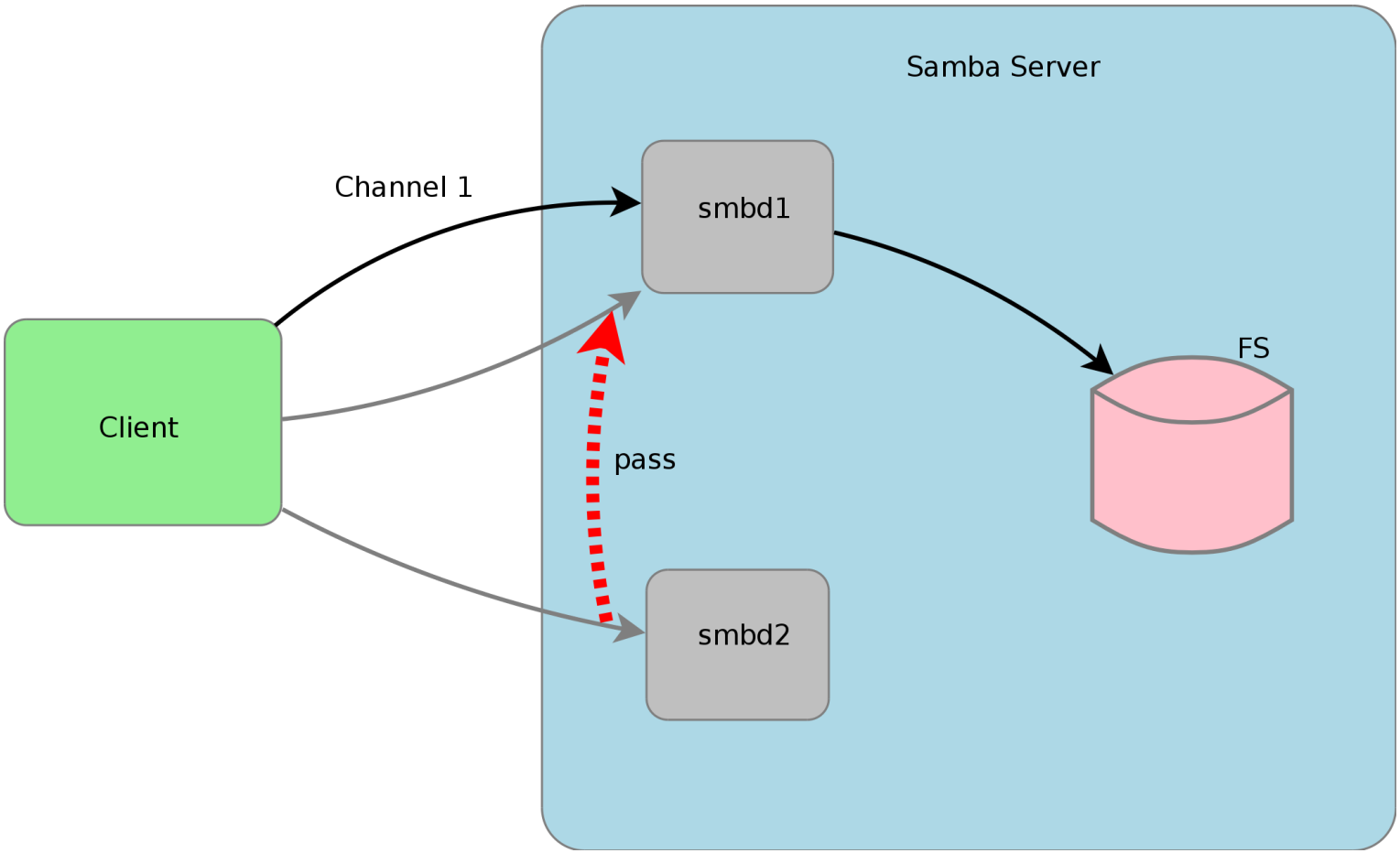
samba/smbd: multi-process

- **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

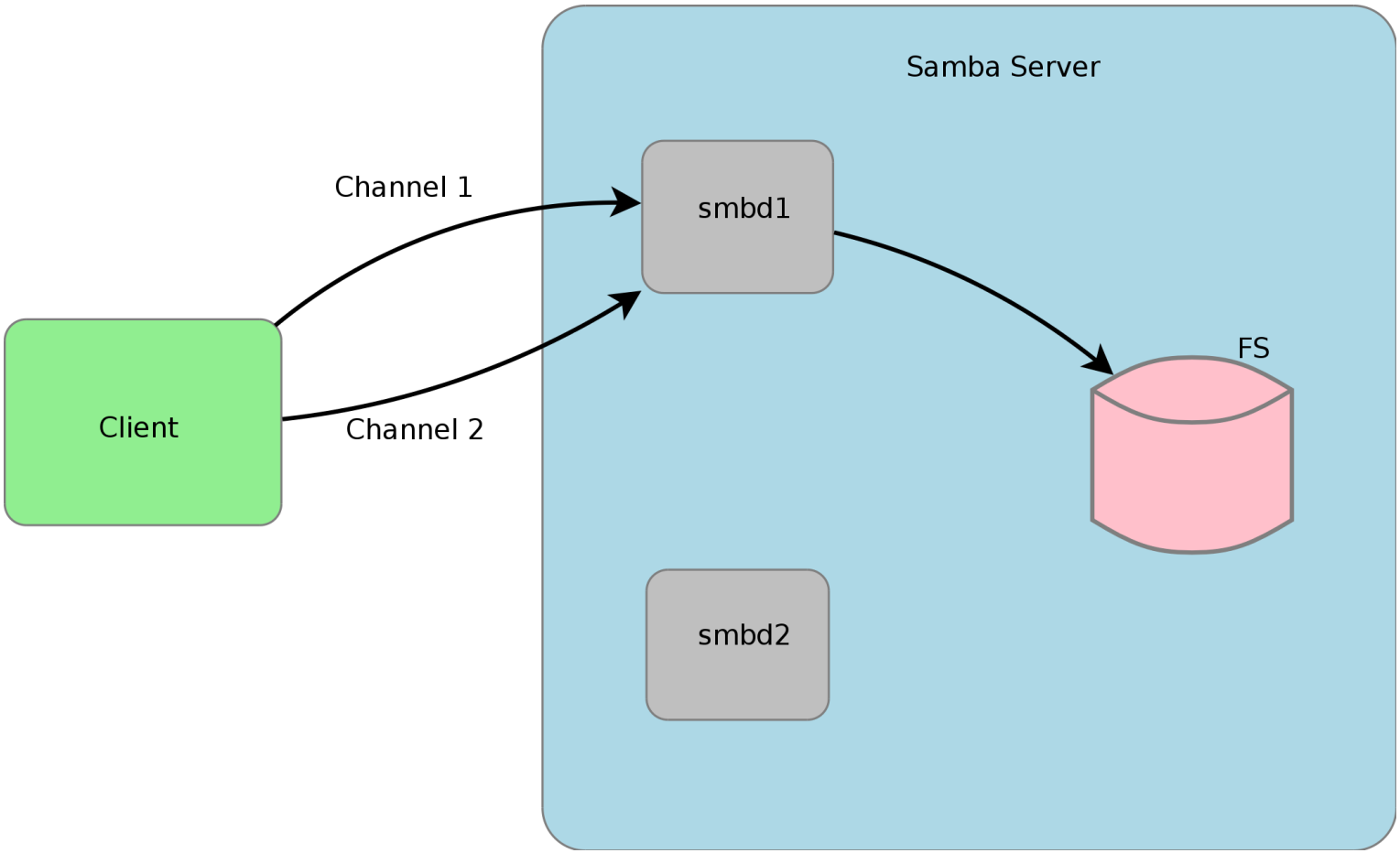
Multi-Channel ∈ Samba



Multi-Channel ∈ Samba



Multi-Channel ∈ Samba



Multi-Channel ∈ Samba

samba/smbd: multi-process

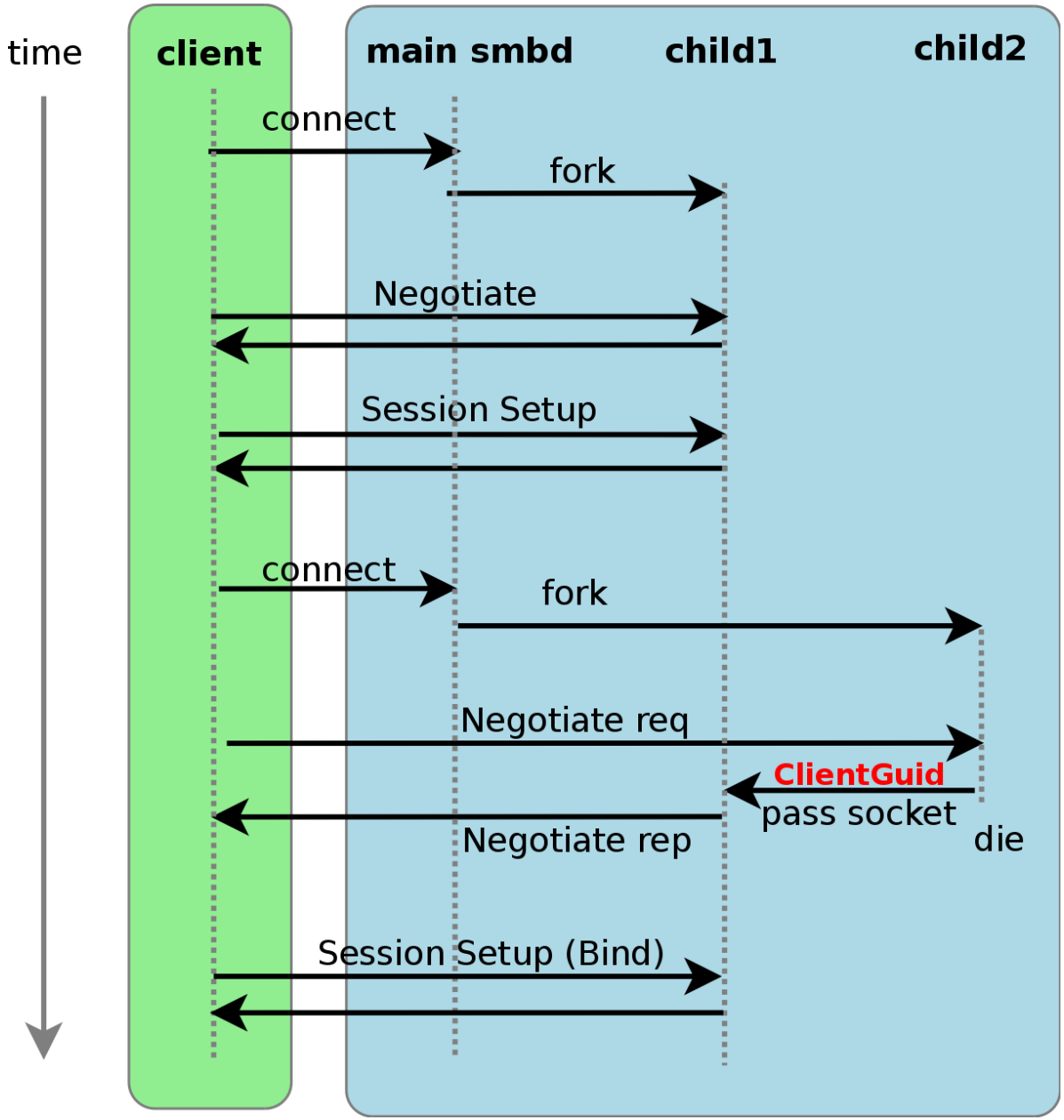
- **Originally:** process ⇔ TCP connection
- **Idea:** transfer new TCP connection to existing smbd
- **How?** ⇒ use fd-passing (sendmsg/recvmsg)
- **When?**
  - *Natural choice*: at SessionSetup (Bind)
  - Idea: as early as possible, based on ClientGUID
    ⇒ per ClientGUID single process model

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!
  ⇒ using multiple CPUs
- Benchmarks and tunings in progress

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,4.2–4.4]
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 sequence 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,4.2–4.4]
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 sequence 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 : How we got there

- Based on preparations in 4.2 and earlier (200+ patches)
  - Patches by Stefan Metzmacher, Michael Adam, Volker Lendecke, Anubhav Rakshit
- Since Summer 2015:
  - Polishing of large parts of massively WIP branch
  - Added new code (create replay, interface detection)
  - Result merged in units. Overall some 130 patches.
  - Patches by:
    - Michael Adam
    - Stefan Metzmacher
    - Günther Deschner
    - Anoop C S
    - Anubhav Rakshit
- Just made it as experimental feature into Samba 4.4

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: the newer patches

shell breakout...


START CROSSING
Watch For
Vehicles
FLASHING
DON'T START
Finish Crossing
If Started
18
TIMER
TIME REMAINING
To Finish Crossing
STEADY
DON'T CROSS
PUSH BUTTON
TO CROSS

Multi-Channel ∈ Samba : How to enable it

smb.conf

```
[global]
...
server multi channel support = yes
...
```

Multi-Channel ∈ Samba: TODOs

- teach socket_wrapper fd-passing (⇒ selftest...)
- Replay lease breaks upon channel failure (server → client)
  DANGER!
- clustering integration (CTDB)
  DANGER!

Multi-Channel ∈ Samba: TODOs

- teach socket_wrapper fd-passing (⇒ selftest...)
- Replay lease breaks upon channel failure (server → client)
  DANGER!
- clustering integration (CTDB)
  DANGER!

Multi-Channel ∈ Samba: TODOs

- teach socket_wrapper fd-passing (⇒ selftest...)
- Replay lease breaks upon channel failure (server → client)
  DANGER!
- clustering integration (CTDB)
  DANGER!

Multi-Channel $\in$ Samba: TODOs

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

Multi-Channel ∈ Samba : Clustering/CTDB

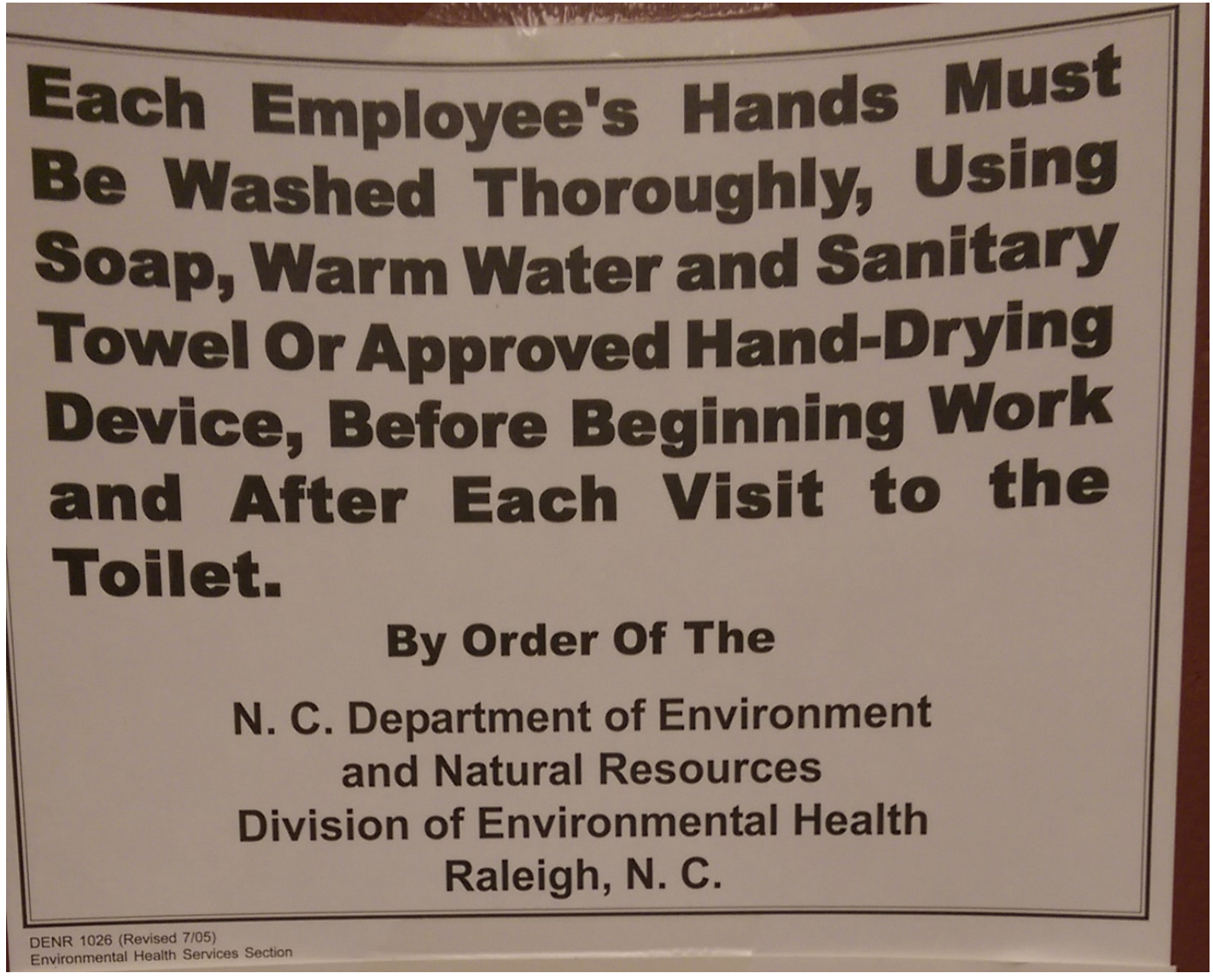
Special considerations

- channels of one session only to one node !
- do not bind connections to CTDB public IPs (can move)!
- problem: CTDB clustering transparent to SMB clients...

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)!
- problem: CTDB clustering transparent to SMB clients...


Each Employee's Hands Must Be Washed Thoroughly, Using Soap, Warm Water and Sanitary Towel Or Approved Hand-Drying Device, Before Beginning Work and After Each Visit to the Toilet.
By Order Of The
N. C. Department of Environment and Natural Resources
Division of Environmental Health
Raleigh, N. C.
DENR 1026 (Revised 7/05)
Environmental Health Services Section

Multi-Channel ∈ Samba : Clustering/CTDB

Plan for integration

- establish blacklist of addresses (e.g. CTDB public IPs)
- add static IPs to public interfaces
- optionally establish whitelist (interfaces ...)
- ⇒ list of allowed addresses
- only publish allowed addresses in interfaces info ioctl
- only give more than one address in interface info when asked via an allowed address
- deny session bind on non-allowed address

Multi-Channel ∈ Samba : Clustering/CTDB

Plan for integration

- establish blacklist of addresses (e.g. CTDB public IPs)
- add static IPs to public interfaces
- optionally establish whitelist (interfaces ...)
- ⇒ list of allowed addresses
- only publish allowed addresses in interfaces info ioctl
- only give more than one address in interface info when asked via an allowed address
- deny session bind on non-allowed address

Multi-Channel Demo

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 $\Rightarrow$ See Jeremy's Talk!

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 $\Rightarrow$ See Jeremy's Talk!

Thanks for your attention!

Questions?

obnox@samba.org

obnox@redhat.com

https://git.samba.org/?p=obnox/slides/2016-05-sambaxp.git
https://www.samba.org/ obnox/presentations/2016-05-sambaxp/*.pdf