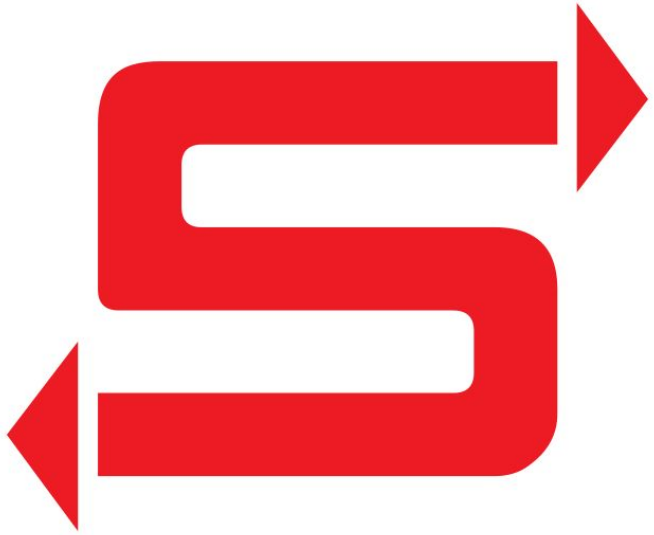

Birds of a Feather: Concurrent version update

SambaXP 2026

Günther Deschner

Rolling updates

- Welcome to the first BoF session at SambaXP!
- Continuation of discussion recently held at Samba Online Gathering (Friday Apr. 10th)
- “Rolling updates” is one of the major and long requested features for Samba
- Actually: what is this feature all about?

Rolling updates - Definition

- **Users expect Samba and its components to be updated individually without any interruption of the service they provide**

Rolling updates - Examples

- Distributed systems (e.g. Ceph) typically provide this feature by having *versioning* in all public APIs
  - Only when all participants share the same version new features can be provided

Rolling updates - Current situation

- How do things look like in Samba today?

Rolling updates - Current situation

- Very limited support of versioned interfaces used by Samba clients and servers in general
- Upgrade-only path (on the fly conversion of databases), no downgrades possible
- No mixed versioned Samba/CTDB clusters
  - => Unavoidable service interruption during upgrade
  - => Unacceptable in enterprise use cases!

Rolling updates - Current situation

- Do we really need versioning in our subsystems?
  - Many exposed subsystems of Samba have not changed a lot / at all over the years
  - Others have changed significantly!

Rolling updates - Current situation

- Safeguards protecting for version mismatches
- Requirement of matching Samba versions on all cluster nodes

  (g_lock.tdb)

  - `allow unsafe cluster upgrade = no`
- CTDB supports minor version differences

  (e.g. 4.24.0 and 4.24.1)

  - https://wiki.samba.org/index.php/Upgrading_a_CTDB_cluster

    (AllowMixedVersions CTDB tunable)

Rolling updates - The challenge

- **Significant challenge to bring versioning to an unversioned distributed system!**

Rolling updates - Versioning

- What is “versioning” ?
- Different approaches in Samba code:
  - Versioning in TDB databases:
    - Distinct versioning database records (e.g. `passdb.tdb` “`INFO/VERSION`” 4)
    - Use of IDL unions and infolevels for database records (e.g. `smbXsrv.idl` based tdbs)
    - Repository of versions in a cluster: `smbXsrv_version_global.tdb`
  - Versioning in messaging and all other API services

Rolling updates - Design considerations

- Major design considerations:
  - Provide rolling upgrades from current version (if possible)
  - Avoid breaking changes (if possible)
  - Incremental rollout of rolling upgrade support over multiple Samba releases
  - Consider containerized workloads with different versions accessing shared databases

Rolling updates - Use cases

- Focus on use-cases:
  - Winbind/CTDB as authentication/authorization source for external services (NFS Ganesha)
    - limited set of subsystems
    - limited set of tdbs involved
    - limited set of messaging operations
  - Mixed client and servers per node
    - Example: utilities reading and writing to smbprofile.tdb (Prometheus exporter, smbstatus)
  - Clustered SMB service

Rolling updates - Design proposal

- Design proposal by Xavi Hernandez:
  - https://gitlab.com/xhernandez/samba-rolling-upgrades
  - Focus on low-level TDB versioning
  - => Community feedback: TDB changes too complex and “too low-level” and making shift to alternative CTDB database backends also too difficult

Rolling updates - Discussion

- What are the possible next steps?
  - Significantly invest into test automation to
    - verify current interfaces
    - prevent any future breaking changes to established and public interfaces
  - Convert Samba subsystems to use IDL based structures
  - Define abstraction APIs on top of dbwrap controlled databases
- Versioned APIs is not enough, who orchestrates?

Q&A

Questions?