Using Samba in Enterprise products

Dr.-Ing. Jens-Peter Akelbein, Dr. Ingo Meents
Storage Systems Architecture
IBM Deutschland Research and Development GmbH

IBM at 100

Using open source software in Enterprise products

Functional requirements do not really change for open source software being used in an Enterprise product…

…while you implicitly expect

the system is always up and running 365x7x24.

large amounts of workload are consolidated on the system.

there is a large dependency of many users or business critical applications to the system.

capacity can be added to the system online.

maintenance procedures should always be performed online.

sophisticated disaster recovery scenarios are in place consisting of two or more systems.

the system recovers immediately whenever something unplanned happened.

Using open source software in Enterprise products

For high availability lets count the 9's (http://en.wikipedia.org/wiki/High_availability)

- 99,99% (four nines), 52,56 minutes downtime in a year
- 99,999% (five nines), 5,26 minutes downtime in a year
- 99,9999% (six nines), 31,5 seconds downtime in a year

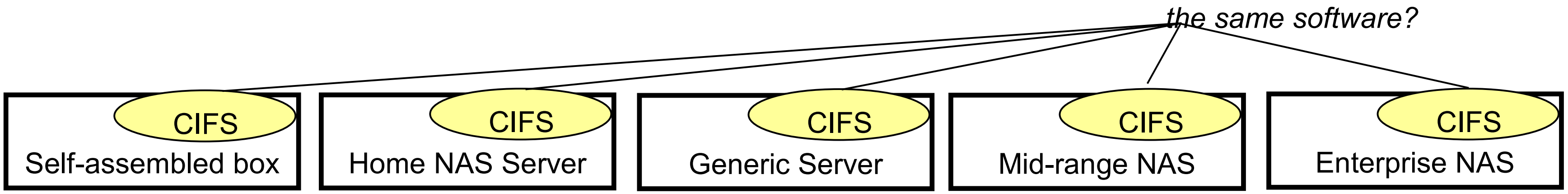
- How many updates implying downtime can you afford?

The system requires regular software updates without or with minimized downtime

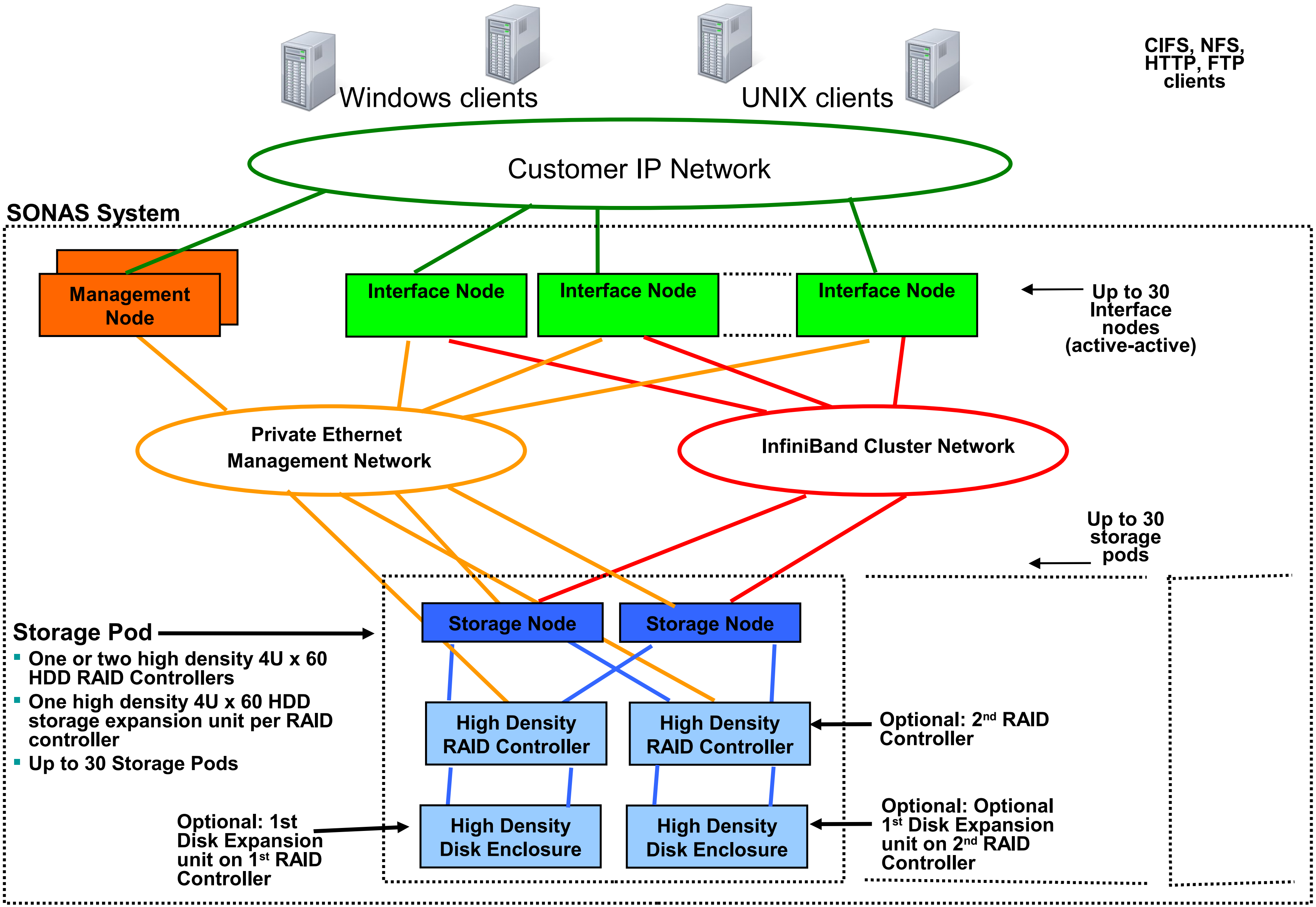
- You update your clustered system consisting out of two or multiple nodes concurrently
- You have more than one software level running concurrently
- You limit the system capability as less as possible i.e. only one node or controller is running at the same time having downlevel and updated instances running in parallel

The system is behind one or several firewalls

- with limited or no remote access
- limited or no ability to call home for the system
- sometimes with limited bandwidth not allowing to send back large volume of data



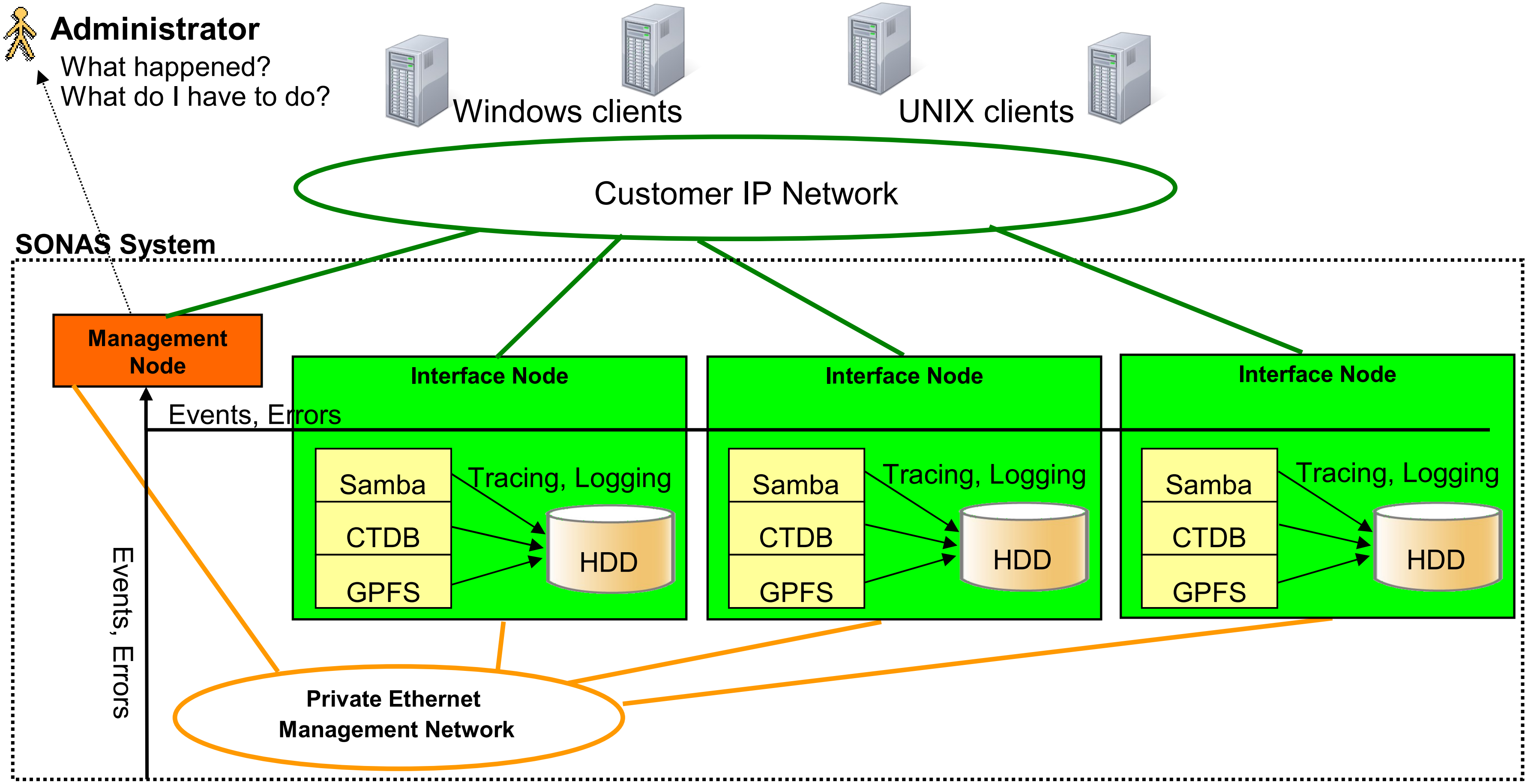
SONAS Enterprise System


Windows clients
UNIX clients
CIFS, NFS, HTTP, FTP clients
Customer IP Network
SONAS System
Management Node
Interface Node
Interface Node
Interface Node
Up to 30 Interface nodes (active-active)
Private Ethernet Management Network
InfiniBand Cluster Network
Up to 30 storage pods
Storage Node
Storage Node
High Density RAID Controller
High Density RAID Controller
Optional: 2nd RAID Controller
High Density Disk Enclosure
High Density Disk Enclosure
Optional: 1st Disk Expansion unit on 1st RAID Controller
Optional: Optional 1st Disk Expansion unit on 2nd RAID Controller

Storage Pod

- One or two high density 4U x 60 HDD RAID Controllers
- One high density 4U x 60 HDD storage expansion unit per RAID controller
- Up to 30 Storage Pods

Monitoring and upgrading of clustered nodes



- Internal HDD in Interface Nodes allow to log and trace large amounts of data
- Samba and CTDB logs are generated by several nodes
- Examples on what an administrator needs to know:
  - Why did one of my clients lost its connection?
  - A hardware part failed, how healthy is my system?
  - A maintenance task needs to be carried out. Which clients will be affected?
- Examples on what the product vendor needs to know:
  - For an upgrade a new Samba version is getting applied. Can I carry this out concurrently?
  - How can I develop and select patches allowing minimized number of non-concurrent upgrades?

Fault Isolation - First Fault Problem Solving

- Requirements
  - ideally enough diagnostic data is captured to understand the issue completely when a problem occurs
  - different means for different users
    - customer administrator
    - root administrator
    - level 1,2,3 support
    - development
    - audit
  - “dark site” support: no ssh, no login, no data offload
  - customer servicability vs. support servicability
  - if recreates are necessary they need to be done easily (->next slide)
- Challenges
  - Samba running in
    - debug lvl. <10 - often not enough data
    - debug lvl. 10 - a lot of data, performance impact

Whitelist Translation & Tracing Commands

- Syslog Translation
  - translating error code like the following

```
2010-11-12T13:53:42.622866-07:00 int003st001 winbindd[2186699]:   INTERNAL ERROR: Signal 11 in
 pid 2186699 (3.4.7-ctdb-14)
```

  - into a message that is being understood by the customer administrator
  - i.e. error code with explanation, user response, system response, etc.

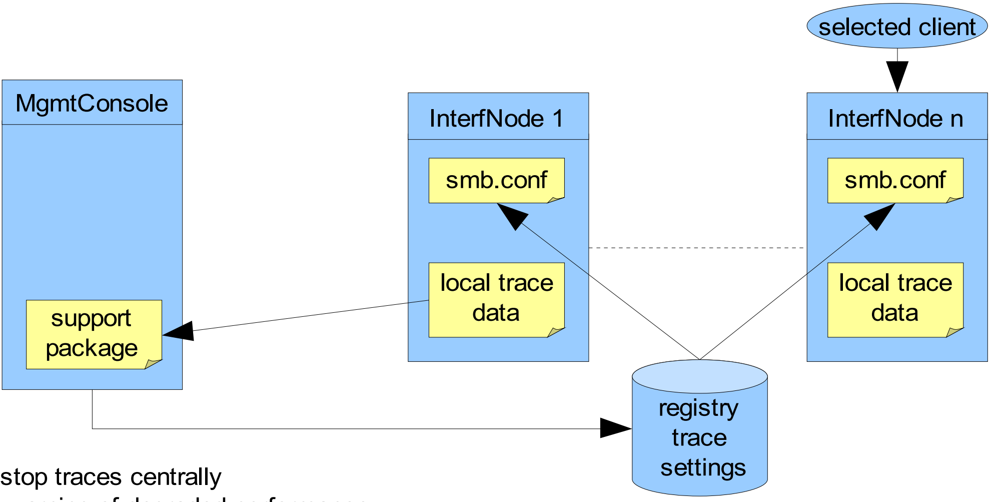
- Solution for lvl 10 traces: CLI commands

  starttrace cifs

  lstrace

  stoptrace

  download dump data



- start/stop traces centrally
- show warning of degraded performance
- offer support package for download

- Issue: restart is required - stopping deamons (killing connections) affects open connections

Concurrent Code Upgrades - Upgrading code while cluster serves I/O

- Requirements: ideally no downtime (--> 5 nines)
- Challenges
  - keeping changing data structures consistent across different versions
  - Samba / ctdb do not have versioned data structures for communication

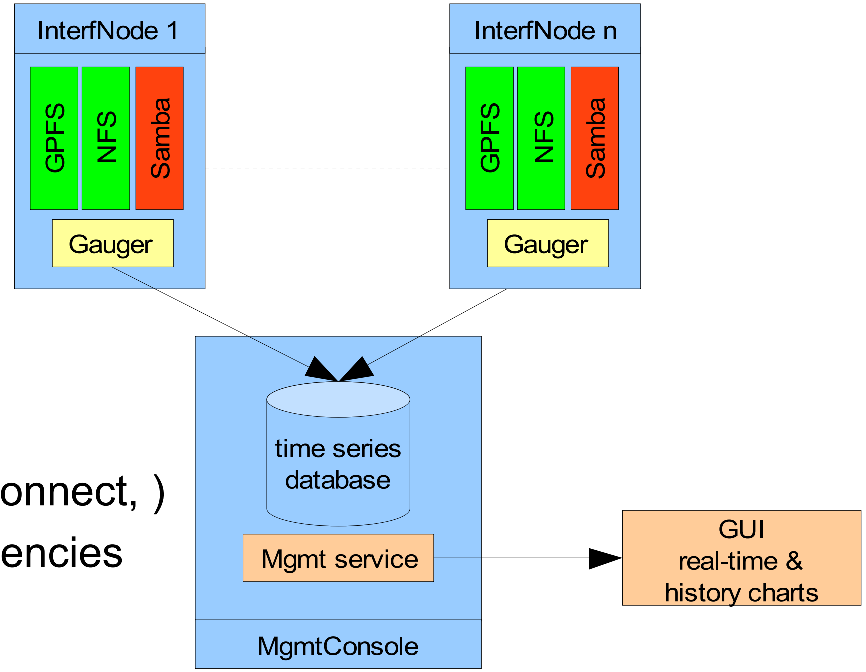
- Solution for code upgrades (2 node example)
  - disconnect node 1 ***failover of connections to node 2***
  - upgrade node 1
  - disallow any Samba/ctdb configuration changes by disabling services on node2
  - dump TDBs on node2 and send backups to node 1
  - shutdown ctdb on node2
  - restart ctdb on node 1 as disabled
  - restore all relevant persistent TDBs on node 1
  - allow configuration changes by fully restarting ctdb
  - upgrade node2
  - node 2 rejoins the cluster ***failback connections to node 1***

downtime for whole cluster

- Extension for n nodes: update each half of the cluster in turn

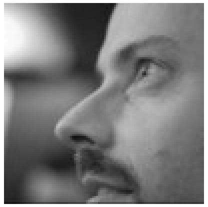
Performance Monitoring

- Requirements
  - need to see / analyze problems
  - capacity planning
- Challenge: Performance metrics are currently not supplied by Samba
- Desired metrics
  - number of connected clients
  - read/write throughput (MB/s)
  - read/write operations per seconds
  - operations counters (read, write, connect,)
  - operation time measurements / latencies
- Moreover
  - list of all connected clients
  - client statistics
- The metrics should be available at node, client, user, and share level.
- Challenge: aggregation of performance data of separate smbd procecesses



The Trivial (64bit transactional) Database - tdb2

- Trivial database: tdb
  - small, efficient database for key-value pairs, safe transactions
  - originally developed for Samba, other projects are using it as well
- The issue: current tdb format on disk is 32-bit, 4GB limits are approaching
- Improvements
  - API enhancements: nested transactions, thread safeness, and others
  - Performance and scalability enahncements: get beyond 4GB limit



- Author: Rusty Russell, IBM
- Design document
  - http://samba.2283325.n4.nabble.com/TDB2-Draft-Design-Document-td2480209.html
- Prototype available for Samba 4
  - in Comprehensive C Archive Network (http://ccan.ozlabs.org/info/tdb2.html)
- Among next steps:
  - Samba 3 build
  - tdb1 on-disk compatibility

Summary

- For using Samba in **Enterprise products**, you
  - have a mature open source software project with a very active community fulfilling a large amount of requirements today
  - require having the system up and running whatever happens
  - use enough hardware redundancy to maintain the system up and running whatever you need to replace
  - require having different software levels running concurrently during an upgrade
  - require sufficent (and no more than that) information gathering (logging, monitoring) during normal operation
  - scale your system to the next order of magnitude reconsidering architectural limitations for example due to 32 bit limitations
  - have a lot of improvements on your todo list :-)