Implementing SAM replication in Samba 3

- Richard-Guillaume Renard <rrenard@idealx.com>

> IDEALX solutions

> SAM replication howto

> BDC side

> PDC side

IDEALX
the Open Source leader in France

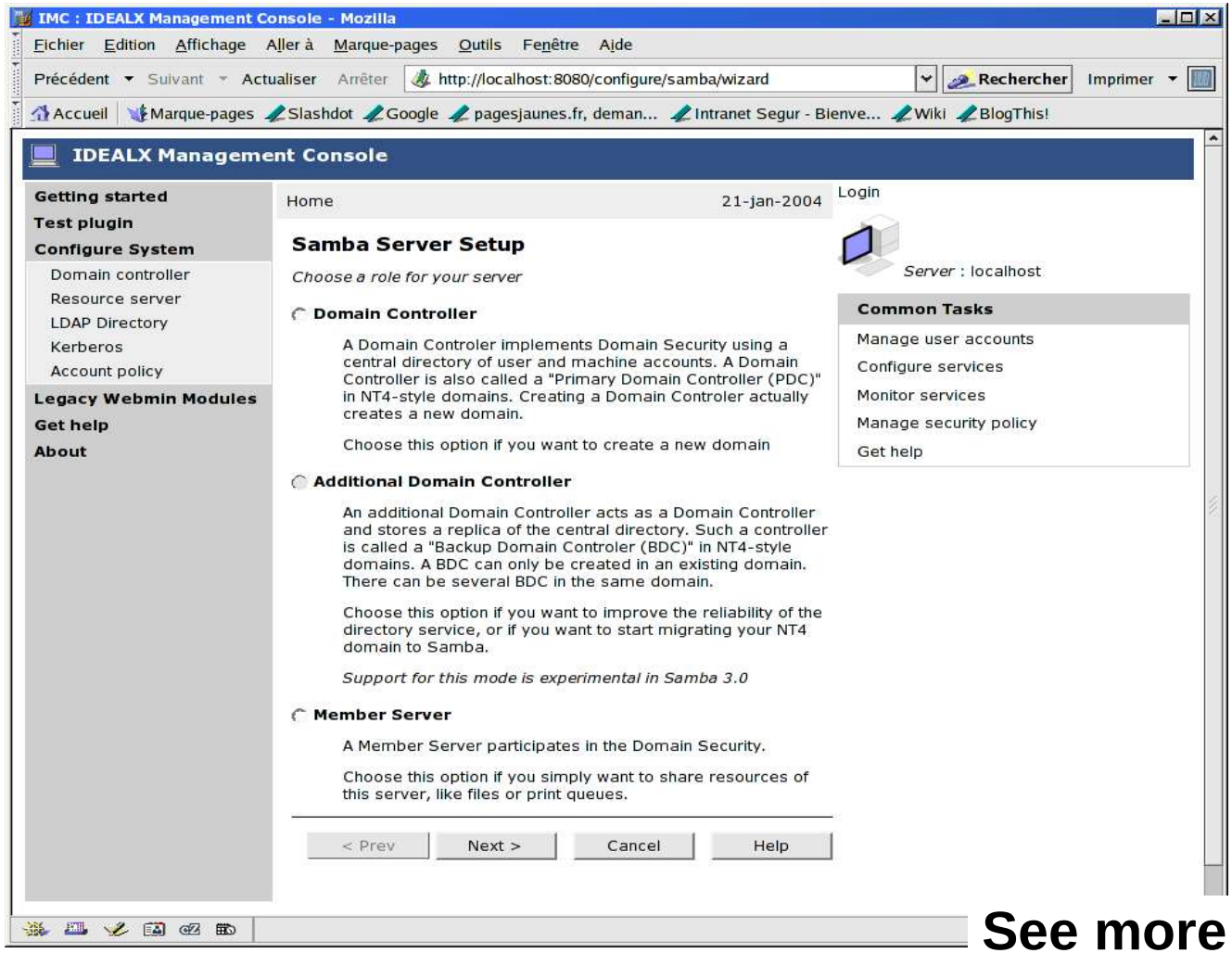
- Founded in feb. 2000
- Engineering Team : 60 people in january, growing...
- Main Shareholder : Caisse des Dépôts et Consignation
  (largest french investor, financing leading e-gov. projects like electronic citizen cards)

- *IDEALX is the reference Open Source partner for French Fortune-100 and Government*
- *IDEALX User Club gathers requests from corporations and enable partial financing of Samba developments (Thanks to Gaz de France for funding the TSE work)*

Best Open Source Solutions

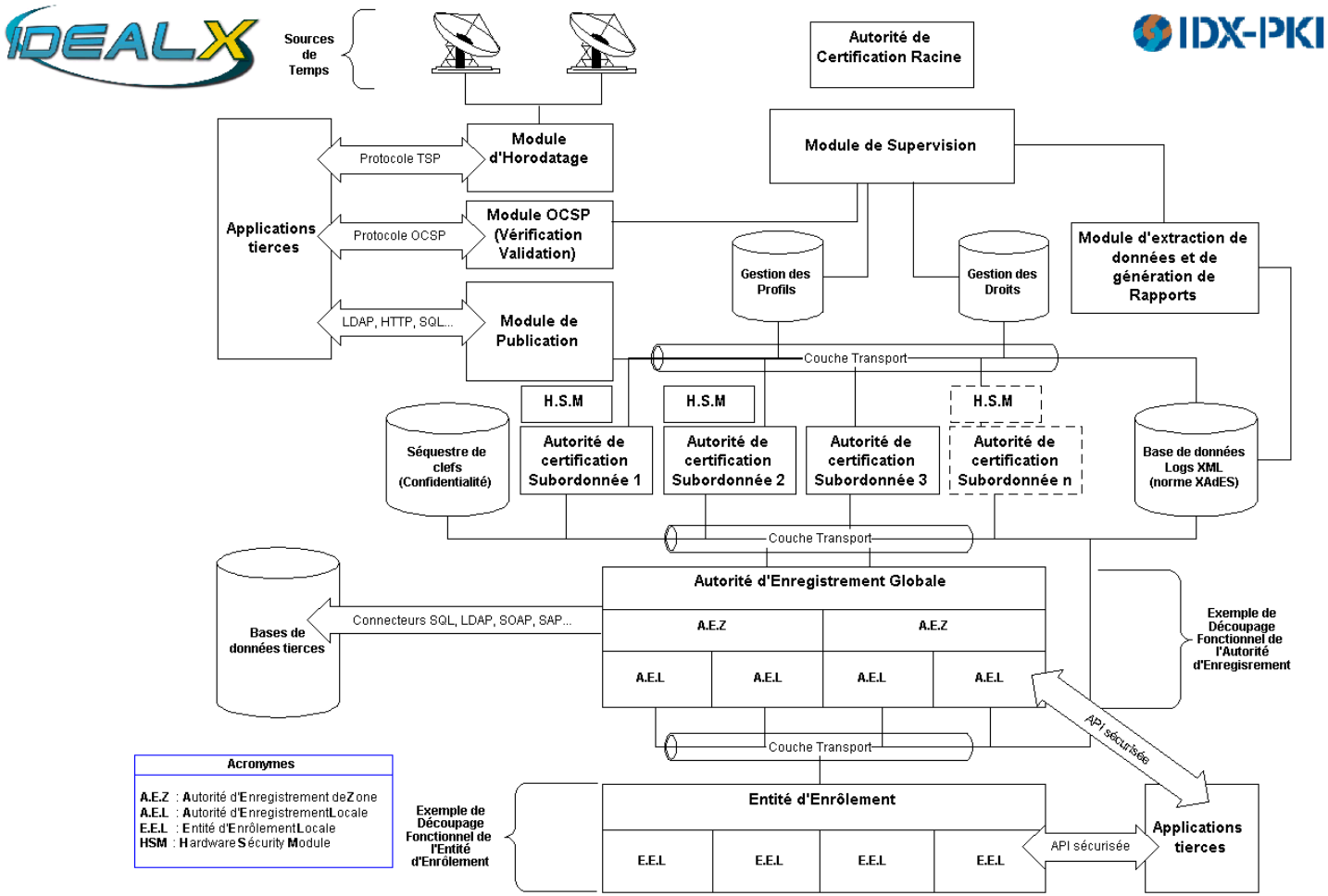
- > Security solutions
  - *IDX-PKI is the #1 PKI in France*
  - *Beating every large proprietary editor*
  - *Expanding in Europe*
  - *Recommended in the latest O'Reilly book ("PKI Open Source")*
- > Infrastructure Solutions
  - *Migration Server is an integrated package of OSS components for NT replacement projects (Samba, LDAP, DNS, DHCP, etc.)*
  - *Adopted by most of the retail sector (Auchan, Décathlon, Castorama/King Fisher, etc.)*
  - *New IMC platform for developping user friendly Web console tools*

IMC in action...



See more at idealx.org

Did we mention our PKI offering... ?



See details at idx-pki.idealx.org

“And now for something completely different...”

Why implementing NT replication protocol

Allow NT and Samba to work together.

- in organizations when DCs are at distant locations

Easier migration

Smoother transition, and test period

Samba could use this mechanism for replication too

We could add some Samba specific fields.

NT4 Domain

One master called PDC (Primary Domain Controller)

- Holds the master copy of the SAM database

Many secondary called BDC (Backup DC)

- Holds read-only copy of the SAM

SAM composed of 3 Databases

- Accounts
- Built-in
- LSA

Synchronisation Mechanism

When a modification is made on the PDC's SAM, a notification message is sent to BDCs.

Notification message contains serial number for each Databases.

Synchronisation can occur immediately after a change

- accout policy change, locked account

or after a delay (5 to 15 min)

- adding/deleting users/groups

Synchronisation Mechanism

Replication is a “pull” process, BDC makes a call on the \\NETLOGON pipe of the PDC for changes after receiving notification

When asking, BDC gives its databases state (serial n°)

The three databases are independent for sync, database is specified when the request is made.

Two Synchronisation Types

Partial Sync (NetDatabaseDeltas)

- contains only recent changes made on the domain
- based on sequence number

Full Sync (NetDatabaseSync)

- contains all SAM database informations
- initiated when BDC enters in a domain
- when BDC is out of sync
- if crash happened during last sync

Notification Message

Notification message contains serial n° for each DBs, this number have two states:

- current database state
  - BDC should be in sync, does not ask for deltas
- 0xFFFFFFFF
  - BDC not in sync, ask for deltas

If given serial number is wrong or outdated a Full Sync is requested by the PDC

Notification Message

...
Microsoft Windows Logon Protocol
Command: Announce Change to UAS or SAM (0x0a)
Low Serial Number: 350
Date/Time: 1073551413
Pulse: 7200
Random: 1
PDC Name: PDC Domain Name: DOMAIN
Unicode PDC Name: PDC Unicode Domain Name: DOMAIN
DB Count: 3
DBChange Info Structure: index 0
Database Index: 0
Large Serial Number: 4294967295
NT Date/Time: Jan 8, 2004 09:43:33.448547363
DBChange Info Structure: index 1
Database Index: 1
Large Serial Number: 2
NT Date/Time: Jan 8, 2004 08:54:53.791530609
DBChange Info Structure: index 2
Database Index: 2
Large Serial Number: 30
NT Date/Time: Jan 8, 2004 08:56:40.934349060
...

DatabaseDeltas

Reply contains an array of SAM objects

- users
- groups

and synchronisation points

- modification_count delta

The next serial number that should be used for the next sync is contained before the delta array

DatabaseDeltas

DCE RPC
Microsoft Network Logon
Operation: NetrDatabaseDeltas (7)
AUTHENTICATOR: return_authenticator
Credential: CDB74399FCD60FB8
Timestamp: Oct 13, 2033 10:40:20.000000000
MODIFIED_COUNT: domain modified count
Modify Count: 384
DELTA_ENUM_ARRAY: deltas
Referent ID: 0x00167420
Num Deltas: 1
DELTA_ENUM: deltas
Referent ID: 0x0016d2d0
Max Count: 1
DELTA_ENUM:User VMNTBDC3$
Delta Type: User (5)
DELTA_ID_UNION:
DELTA_UNION: VMNTBDC3$
Delta Type: User (5)
DELTA_USER: VMNTBDC3$

DatabaseSync

Serial number is contained in DOMAIN delta

Microsoft Network Logon
- Operation: NetrDatabaseSync2 (16)
- AUTHENTICATOR: return_authenticator
  - Credential: 2EA86FD3F8C14890
  - Timestamp: Feb 7, 1970 05:02:11.000000000
- Sync Context: 1
- DELTA_ENUM_ARRAY: deltas
  - Referent ID: 0x00165f88
  - Num Deltas: 11
  - DELTA_ENUM: deltas
    - Referent ID: 0x00165fd0
    - Max Count: 11
    - **DELTA_ENUM:Domain VMTEST**
    - DELTA_ENUM:Group
    - DELTA_ENUM:User
    - DELTA_ENUM:Group Member

BDC side – what is needed

We need to act upon reception of a notification event

- Mailslot\NETLOGON

Need to store serial number for next sync

Should consider our local SAM as a read-only copy

BDC side – what is working

Understands notification message

Ask PDC for deltas or full sync

Supported domain operations :

- adding/deleting user, group
- modifying username
- all password policy options

Based on “vampire” code, which is moved into rpc_client/cli_netlogon_util.c

Serial numbers stored in tdb

BDC side – what is left to do

SAM attributes

- local groups
- privileges, trusted domains

Add a system to prevent modifications of the SAM by command line tools or RPC calls ?

BDC side – suggestions, shortcomings

Why not temporarily store deltas before applying ?

Current implementation requires smbd to be started before nmbd (because of the need to get PDC name)

Should we remove our local SAM when full sync is requested ?

NT PDC seems to see Samba as a second class BDC

- PDC keeps sending notifications
- immediate notifications are not immediately sent to us

PDC side – what is needed

We need to support all SAM attributes, even if we don't use them.

Need to keep a list of all our BDCs and their synchro status

Need to keep track of operations done on the domain (users, groups, policy, etc ...), all or only a part ?

Some fields are still unknown

Serial numbers

Serial numbers seems to have their own life

- sometimes incrementation makes sense
  - 1 for a user or group, 2 for domain infos
- sometimes not
  - 4 deltas, increment is only 3

As a PDC should we care about ?

PDC side – what exists

rpc_parse/parse_net.c contains stuff to marshall / unmarshall deltas

client code

PDC side – what's left

2 missing RPC calls

- NetDatabaseSync
- NetDatabaseDeltas

All the system to keep track of the modifications done on the SAM

The notification system

Roadmap

The goal is to have something that works by the end of

- June

So it can be tested and merged into 3.x