Likewise Software

Let's go to the library - advantages and difficulties of creating componentised DCE/RPC environment

Brian Koropoff
Rafal Szczesniak

Agenda

Small journey across the layers:

- SMB transport
- Named pipes
- DCE/RPC runtime
- MS-RPC libraries

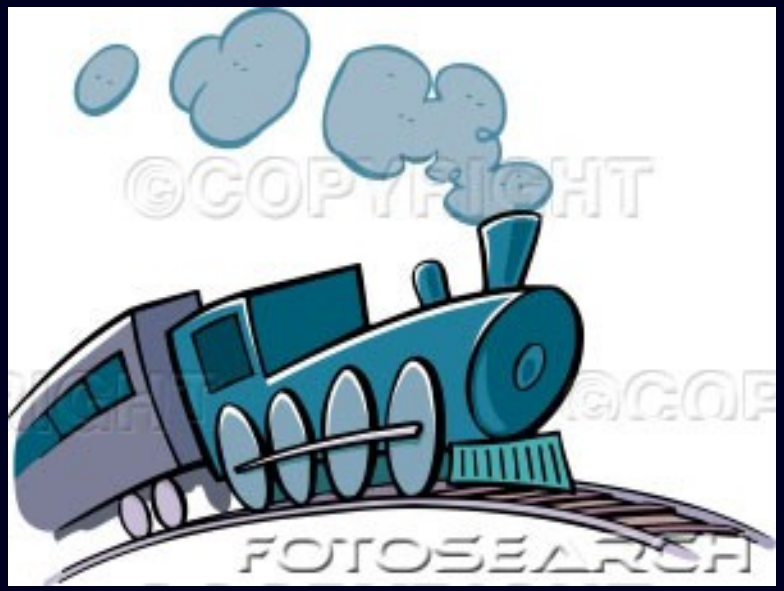
Foundations

- DCE/RPC environment released by Novell/PADL, originally based on OSF libraries
- Samba 3 libsmbclient providing support for named pipes read/write
- Samba 4 idl files providing details of interface

Likewise®

MS-RPC API
DCE/RPC runtime
named pipes muxer

libsmbclient



SMB/CIFS client library

- Originally not designed to carry named pipe traffic, but to provide POSIX compatibility layer
- Pretty well established API though not very extendible

SMB/CIFS client library

Changes made:

- Extended to provide NT-style open (NTcreate)
- Kerberos credentials cache supported in authentication callback
- Access to SMB session key necessary for cryptography in MS-RPC layer
- ABI remains the same

SMB/CIFS client library

Likewise®

Possible improvement:

- Small part of the code could turn into named pipes library

Likewise®

MS-RPC API

DCE/RPC runtime

libsmbclient

named pipes muxer

Named pipes muxer

- Thin layer providing access to SMB named pipes via unix domain sockets
- A connection between DCE/RPC libraries and SMB client library, thus allowing to call remote procedures over named pipes/SMB transport

Named pipes muxer

The muxer works in two modes:

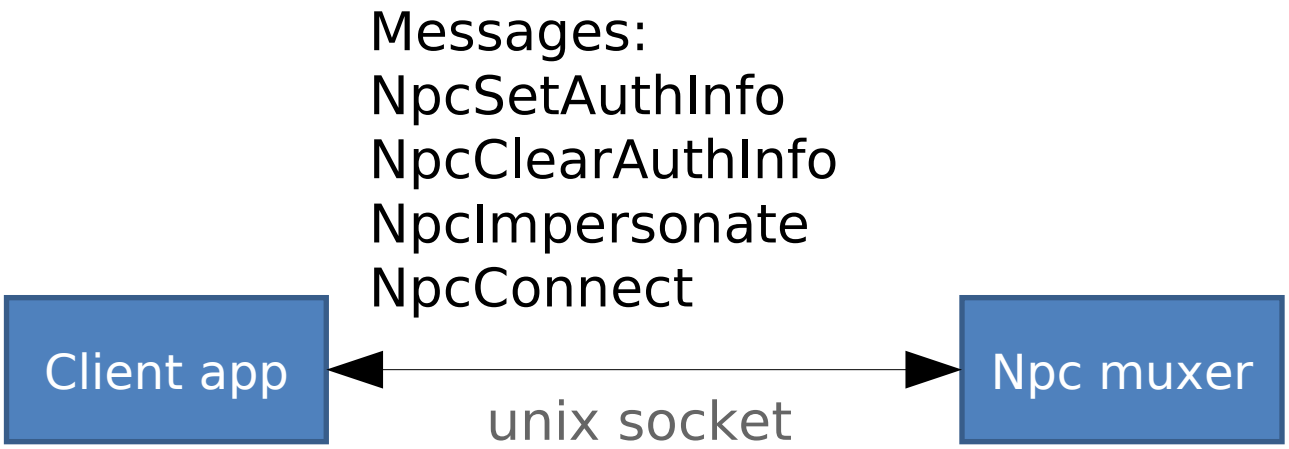
- Daemon listening for commands on unix domain socket
- Transparent proxy forwarding the traffic between the client socket and named pipe

NP muxer – daemon

libnpc library allows sending messages to the muxer

- Credentials handling
- Impersonation
- Transport layer's session key handling
- Opening connection and entering proxy mode

NP muxer – daemon



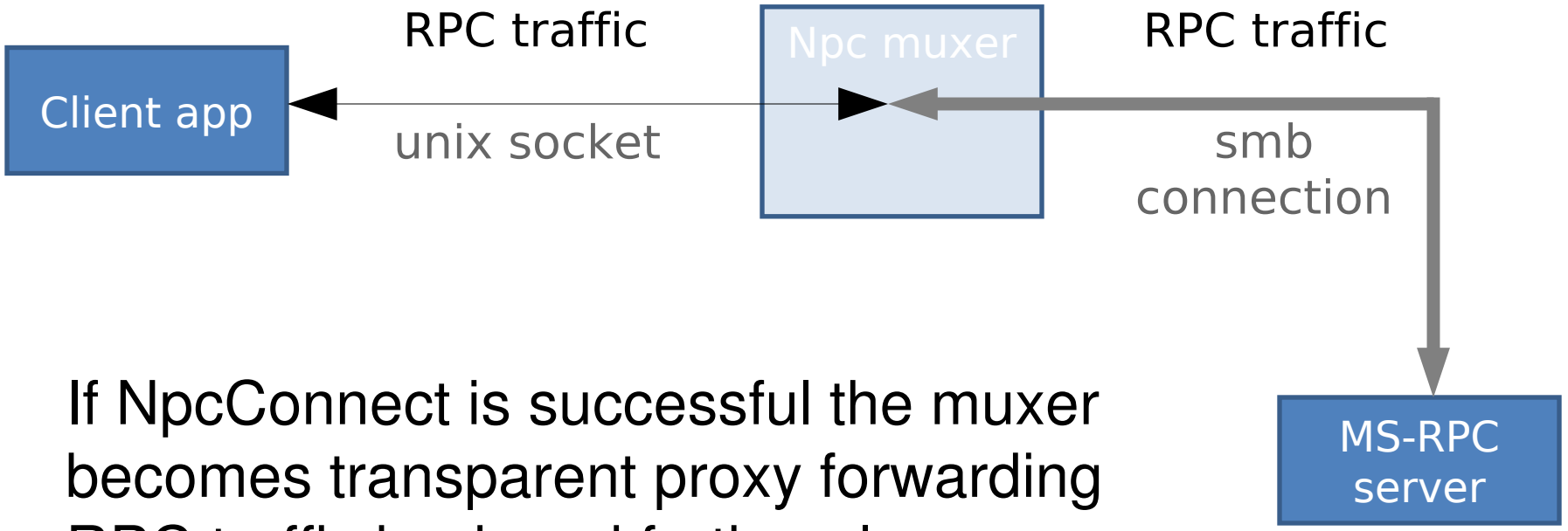
NP muxer – proxy

Likewise®

Connecting remote named pipe enters the proxy mode:

- Remote host's IPC$ is connected
- Named pipe is opened
- Established session key is returned along with status code

NP muxer – proxy



If NpcConnect is successful the muxer becomes transparent proxy forwarding RPC traffic back and forth as long as the connection is active.

NP muxer – future

Likewise®

Possible changes/extensions:

- Separate functions for sending/receiving PDU's (instead of proxy mode)
- Access to all transport layer's data for MS-RPC layer

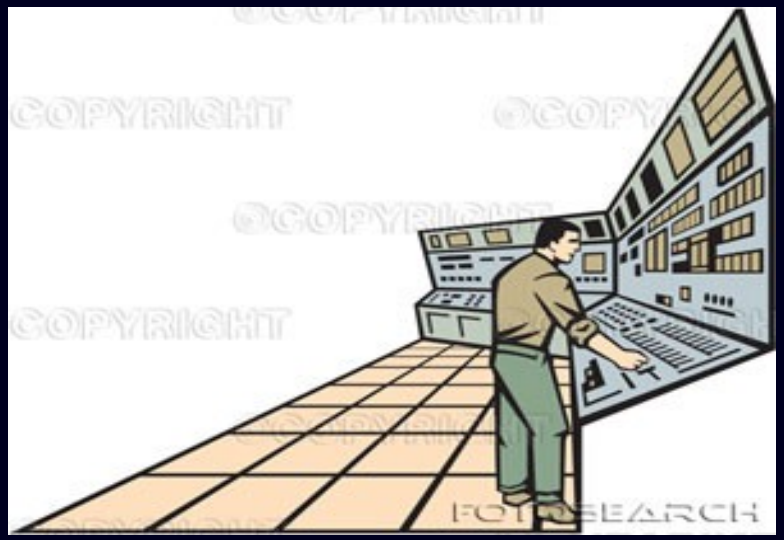
Likewise®

MS-RPC API

named pipes muxer

libsmbclient

DCE/RPC runtime



DCE/RPC environment

OSF heritage:

- Machinery that enables a client to call remote function on a server
- Includes IDL compiler
- Heavily based on threads
- Did not include ncacn_np endpoint support (named pipes)
- Used by Microsoft ages ago for their own DCE/RPC environment

DCE/RPC – changes made

Likewise®

What had to be done to make it useful:

- IDL compiler cleanup to make use of certain MS-specifics in function calls
- Missing piece – ncacn_np endpoint support (named pipes)
- Threads – a big cleanup
- Impersonation token

DCE/RPC compared

Likewise®

Existing DCE/RPC implementations

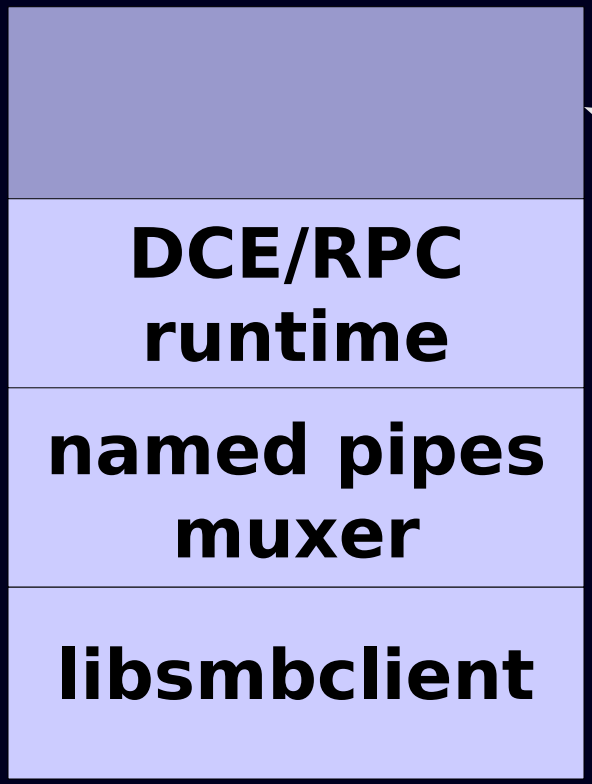
	Samba 3	Samba 4	Likewise DCE/RPC
IDL compiler	PIDL (Perl script)	PIDL (Perl script)	C code + Flex/Bison + C preprocessor
generated code	Human readable	Human readable	Definitely machine readable
Multitasking	Processes	Processes + state machines	Threads

DCE/RPC – changes made

Next steps to be done:

- Support for GSS authenticated binds
- Even more severe threads overhaul
- Further IDL compiler extensions

Likewise®



MS-RPC

MS-RPC libraries

- Provide interfaces for well known named pipes – Lsa, Samr, Netlogon, etc.
- Idl files are based on Samba 4 IDLs
- Different calling convention than what Samba 4 uses – the same result
- All functions are synchronous

MS-RPC libs – string handling

Likewise®

- 2-byte unicode strings are used, just like on the wire
- A separate libunistr library has been developed to make our life easier
- Originally the code (and most importantly idl's) have been tested on Windows

MS-RPC compared

Likewise®

Existing MS-RPC implementations

	Samba 3	Samba 4	Likewise DCE/RPC
Function execution	Synchronous	Asynchronous	Synchronous
RPC layer status	Status code	Status code returned by _recv function	Exception code
String encoding	Single byte	Single byte	2-byte
RPC function result	Status code	Stored in RPC function's I/O struct	Status code returned by rpc function

MS-RPC libs – future

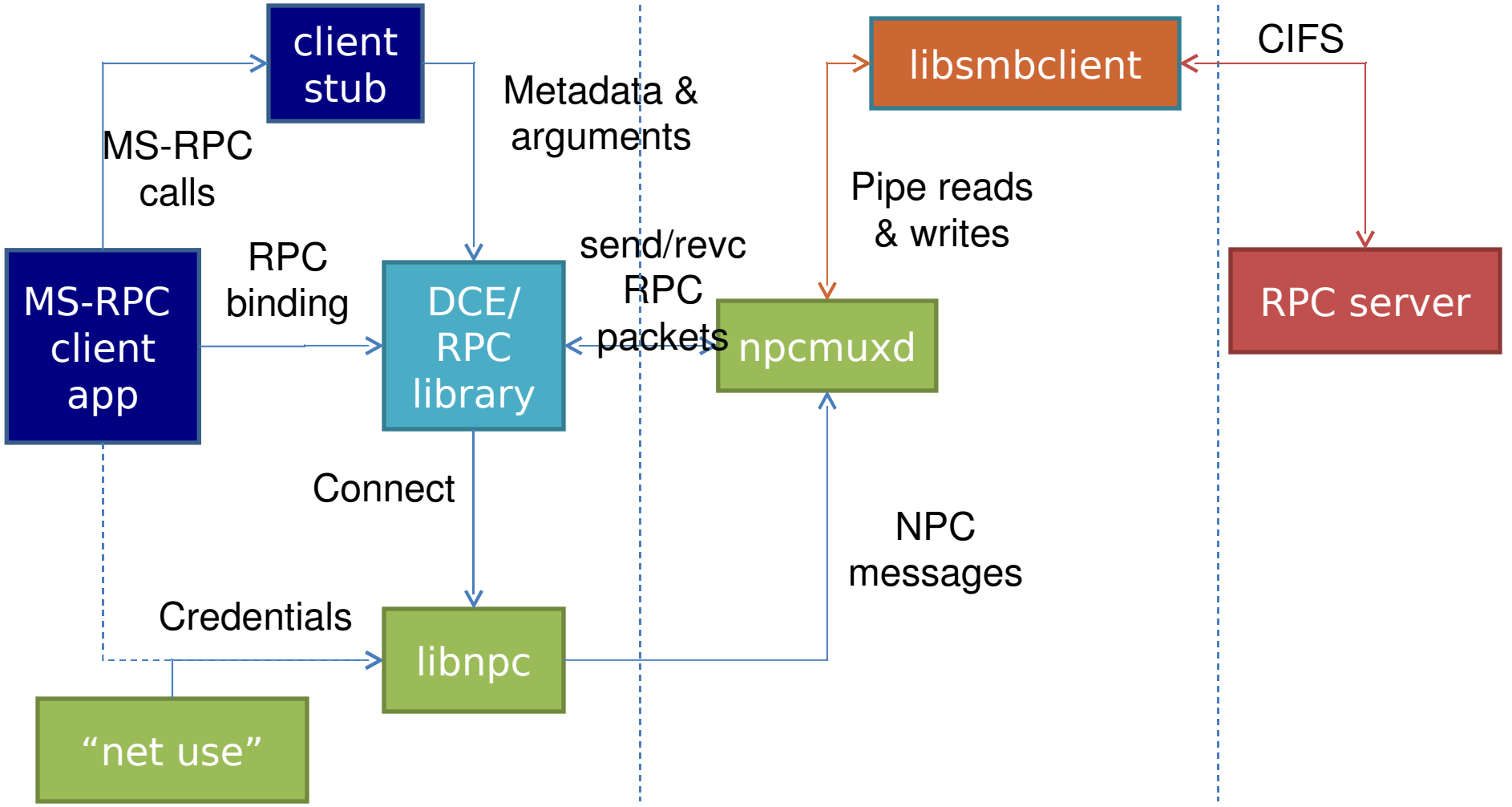
Changes coming soon:

- More complete set of MS-RPC functions in existing pipes
- Another MS-RPC pipe supported (spoolss)

Changes coming (not necessarily soon):

- Server side

Architecture Overview



Extra library

Likewise®

NetAPI

MS-RPC API

DCE/RPC runtime

named pipes muxer

libsmbclient

NetAPI – management library

Likewise®

- Designed to be the exact replacement of windows network management library (NetApi)
- Provides the same API and the same header file (LM.h)
- Allows to use the same source code with no changes on both Windows and Linux
- There's more APIs than “official” ones :)

NetAPI – management library

Let's make our life easier...

	Samr	Net
Arguments	server/domain name username	server/domain name username
Number of function calls involved	~5	1
Intermediary arguments	~6 (handles, ids, access rights)	none!

NetAPI – future

Changes/extensions:

- 100% of original NetAPIs supported
- Server side (\pipe\wkssvc)
- Functions taking 1-byte strings (just for convenience)