

# DCERPC and Endpoint Mapper

Andreas Schneider <asn@samba.org>

Red Hat

May 11th, 2011



# DCERPC and Endpoint Mapper

- 1 DCERPC**
  - How does RPC work?
- 2 Endpoint Mapper**
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server**
  - Overview
  - Robustness
  - Scalability
- 4 Why?**
  - Franky
  - FreeIPA



# DCERPC and Endpoint Mapper

- 1 DCERPC
  - How does RPC work?
- 2 Endpoint Mapper
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server
  - Overview
  - Robustness
  - Scalability
- 4 Why?
  - Franky
  - FreeIPA

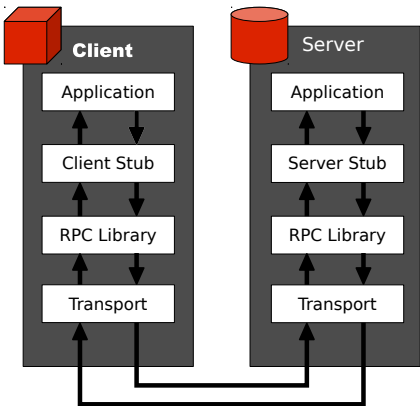


# Abbreviations

- DCE: Distributed (Disturbed) Computing Environment
- RPC: Remote Procedure Call
- NDR: Network Data Representation
- IDL: Interface description language

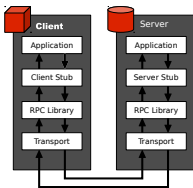


# The RPC process



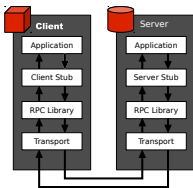
# Application

- spoolss: Printing application displaying a list of printers
- regedit: Display all values of a key



# Client Stubs

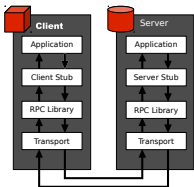
- spoolss: Your application calling `dcerpc_spoolss_EnumPrinters`
- regedit: Your application calling `dcerpc_winreg_EnumValues`



# Run-time Library

RPC client implementation creating a RPC bind

- Establishes the connection
- Authenticates the user





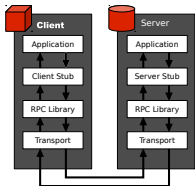
# Transports

- ncacn\_np: SMB Named Pipes transport
- ncacn\_ip\_tcp: DCE/RPC over TCP/IP
- ncalrpc: Local interprocess communication
- ncacn\_http: DCE/RPC over HTTP
- ncadg\_ip\_udp, ncacn\_at\_dsp, ncacn\_nb\_ipx, ncacn\_dnet\_nsp, ...



# Run-time Library

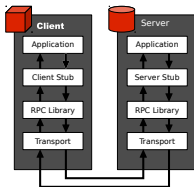
- The RPC Server accepting a connection over a transport and creating the RPC bind
- After successful authentication it calls the Server Stub



# Server Stubs

This unmarshals the packet and calls the application implementation

- spoolss: `_spoolss_EnumPrinters`
- regedit: `_winreg_EnumValues`



# DCERPC and Endpoint Mapper

- 1 DCERPC
  - How does RPC work?
- 2 Endpoint Mapper
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server
  - Overview
  - Robustness
  - Scalability
- 4 Why?
  - Franky
  - FreeIPA



# Abbreviations

- EPM: Endpoint Mapper
- UUID: Universally Unique Identifier (man uuidgen)
- NDR: Network Data Representation



# Terminology

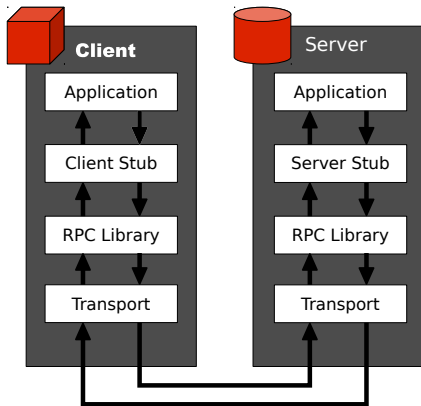
The Endpoint Mapper is a portmapper.

- Endpoint: An endpoint could be a port or a pipe and provide several interfaces
- Interface: An interface is a RPC service provided by an endpoint

The named pipe `\\PIPE\netlogon` can be used for netlogon and lsarpc connections.



# Remember: The RPC process



# Endpoint operations

- Each RPC service allocates one or more endpoints dynamically on server startup
- Endpoint mapper maintains information about those endpoints
- The Endpoint Mapper listens on port 135 TCP/IP or on `\\PIPE\epmapper`





# DCERPC and Endpoint Mapper

- 1 DCERPC
  - How does RPC work?
- 2 Endpoint Mapper
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server
  - Overview
  - Robustness
  - Scalability
- 4 Why?
  - Franky
  - FreelPA



# Function overview

The most important function of the endpoint mapper.

- **epm\_Insert** Add specified entries to an endpoint map.
- **epm\_Delete** Delete specified entries from an endpoint map.
- **epm\_Lookup** Lookup entries in an endpoint map.
- **epm\_Map** Apply some algorithm to an endpoint map to produce a list of protocol towers. (Provide an uuid and get an endpoint)
- **epm\_LookupHandleFree** Free an `epm_Lookup` or `epm_Map` `entry_handle`.



redhat®

# Example

Wireshark trace ...



# An endpoint tower

A tower has up to 6 floors, 4 at least

- 1** Floor1: Provides the RPC interface identifier (netlogon uuid).
- 2** Floor2: Transfer syntax (NDR encoded)
- 3** Floor3: RPC protocol identifier (ncacn\_tcp\_ip, ncacn\_np, ...)
- 4** Floor4: Port address (e.g. TCP Port: 49156, PIPE)
- 5** Floor5: Transport (e.g. IP:192.168.51.10, NB:krikkit)
- 6** Floor6: Routing



# DCERPC and Endpoint Mapper

- 1 DCERPC
  - How does RPC work?
- 2 Endpoint Mapper
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server
  - Overview
  - Robustness
  - Scalability
- 4 Why?
  - Franky
  - FreeIPA



# RPC Endpoints

- Added support for TCP/IP and NCALRPC
- Other processes can register at EPM (OpenChange) over NCALRPC



# DCERPC and Endpoint Mapper

- 1 DCERPC
  - How does RPC work?
- 2 Endpoint Mapper
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server
  - Overview
  - Robustness
  - Scalability
- 4 Why?
  - Franky
  - FreeIPA



# Robustness

## Client

- RPC service tries to register several times
- After successful registration we do connection monitoring

## Server

- We monitor the client connection
- If it goes away, delete the endpoints





# DCERPC and Endpoint Mapper

- 1 DCERPC
  - How does RPC work?
- 2 Endpoint Mapper
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server
  - Overview
  - Robustness
  - Scalability
- 4 Why?
  - Franky
  - FreeIPA



# Pre-fork

We started to implement a mutex locking based pre-fork model.

- Parent binds all sockets and then forks a number of children
- Childs have a lock around the accept(3) call
- Prototype working for our spoolss daemon



# DCERPC and Endpoint Mapper

- 1 DCERPC
  - How does RPC work?
- 2 Endpoint Mapper
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server
  - Overview
  - Robustness
  - Scalability
- 4 Why?
  - Franky
  - FreeIPA



# Franky

- A lot of infrastructure has been created for Franky
- EPM allows us to have multiple daemons



# DCERPC and Endpoint Mapper

- 1 DCERPC
  - How does RPC work?
- 2 Endpoint Mapper
  - Concept
  - Functions and Details
- 3 Samba3 RPC Server
  - Overview
  - Robustness
  - Scalability
- 4 Why?
  - Franky
  - FreelPA



# FreelPA

FreelPA is something like Active Directory but for Linux only.

- We want to be able to do forest trusts with Active Directory
- For this we need LSA and Netlogon (SAMR)
- pdb\_ipa and 'net rpc trust'



## Questions & Answers

- Slides <http://www.samba.org/~asn/>

