

STORAGE DEVELOPER CONFERENCE



Fremont, CA  
September 12-15, 2022

*BY Developers FOR Developers*

A **SNIA** Event

# Making DCE/RPC Calls Time Bound

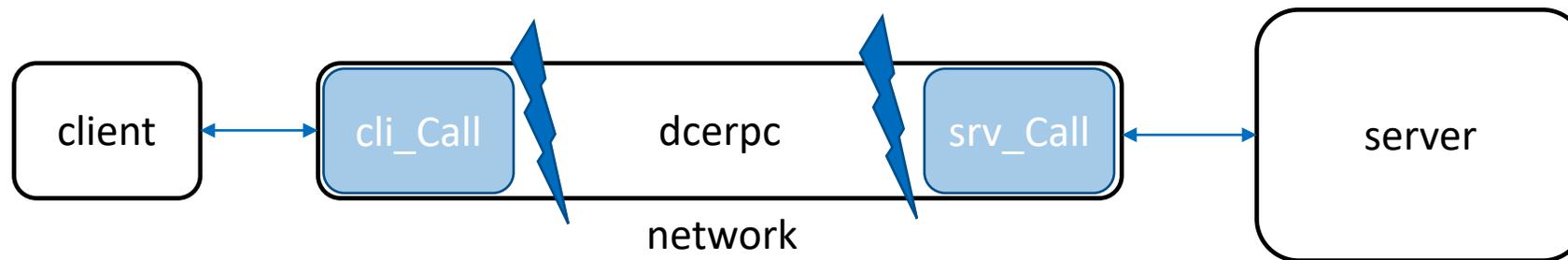
An approach

Amit Gaurav, Principal Engineer, Dell Technologies

# DCE/RPC Overview

# DCE/RPC Overview

- Distributed Computing Environment - Remote Procedure Calls
  - Developed by Open Software Foundation / BSD License
- Underlying remote execution is transparent to caller
- A DCE/RPC call has two parts
  - *cli\_CALL* that client application invokes.
  - *srv\_CALL* that server application implements.
- Client waits until server completes the call, blocking the thread
- Message transport can be SMB or HTTP



# The Issue with DCE/RPC

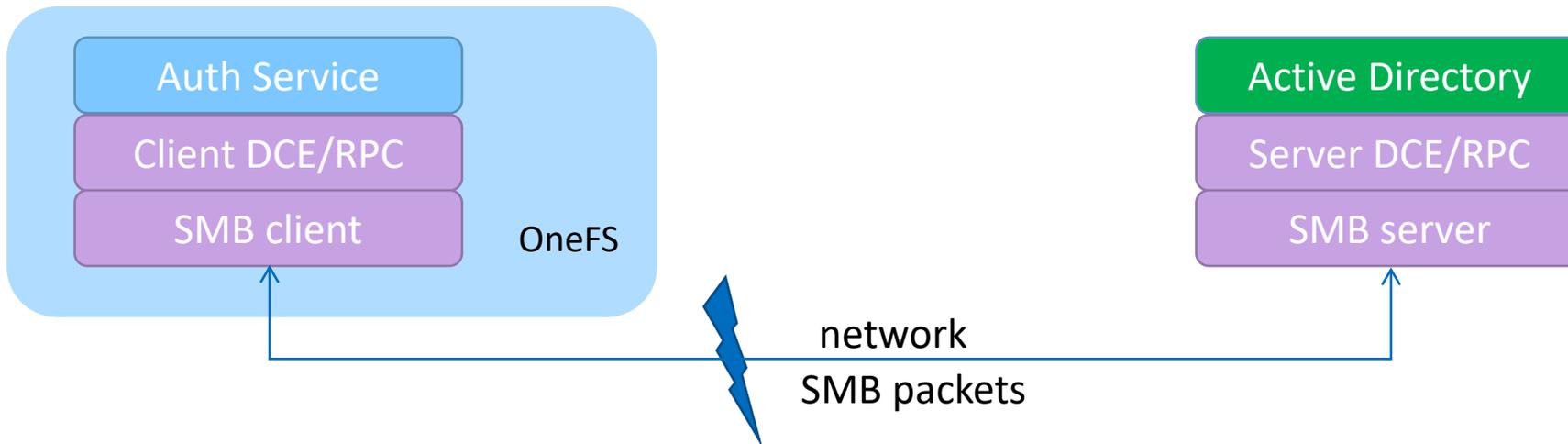
- The calls are synchronous by nature
- The design of DCE/RPC follows the same paradigm of local calls
  - Remote calls are blocked till completion
- Remote execution is more complicated and error prone
- Network blips can cause indefinite wait for calling thread
  - At times, results in cascaded waits across threads
- Depends a lot on “good” server

# DCE/RPC in OneFS

How we use the library

# DCE/RPC in OneFS

- OneFS leverages DCE/RPC to communicate with Active Directory
  - DCE 1.1 Likewise Implementation
  - netlogon, lsarpc, samr
- The communication with Active Directory is through SMB transport
- Home-grown SMB client “Re-director” to communicate with AD



# DCE/RPC in OneFS ... continued

- **Basic Requirement**

- A waiting call must not remain waiting forever
- Caller should be able to decide the time allotted for a waiting call

- **Issues**

- *lsarpc, netlogon, samr* calls are blocking. Threads wait until completion
- A delayed response/no response causes cascaded delays
- Threads use system resources, cannot put all threads to blocking waits
- Calls cannot be canceled from within DCE/RPC
- Relies on server to cancel. If server does not respond, wait.

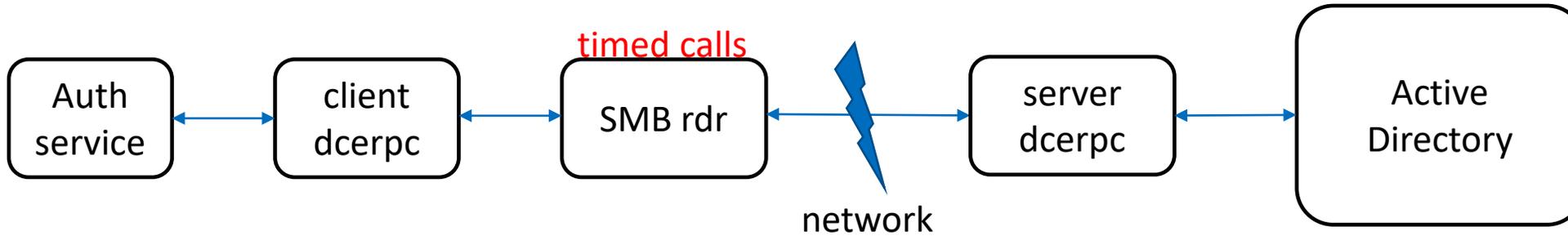
# Timeouts in DCE/RPC

- DCE/RPC does support timeouts but is not straightforward
- DCE timer thread to track all the operations
- Upon timeout, facility to cancel a call
  
- Cancel call requests server to cancel the call
- Nothing could be done if server does not respond to the call
- No guarantee the timeout would be honored

# Asynchronous DCE/RPC

- Requires change in cli\_ and srv\_ interfaces
  - Need one more parameter to return “promise”.
- Need changes in IDL (Interface Definition Language) library that generates client and server DCE/RPC calls
  - MS-RPC has changed the IDL to support async calls
- Requires considerable changes both in interface as well as transport layer
- *Is there a way to meet requirement without messing up with DCE/RPC?*

# Using SMB Client for Timeouts

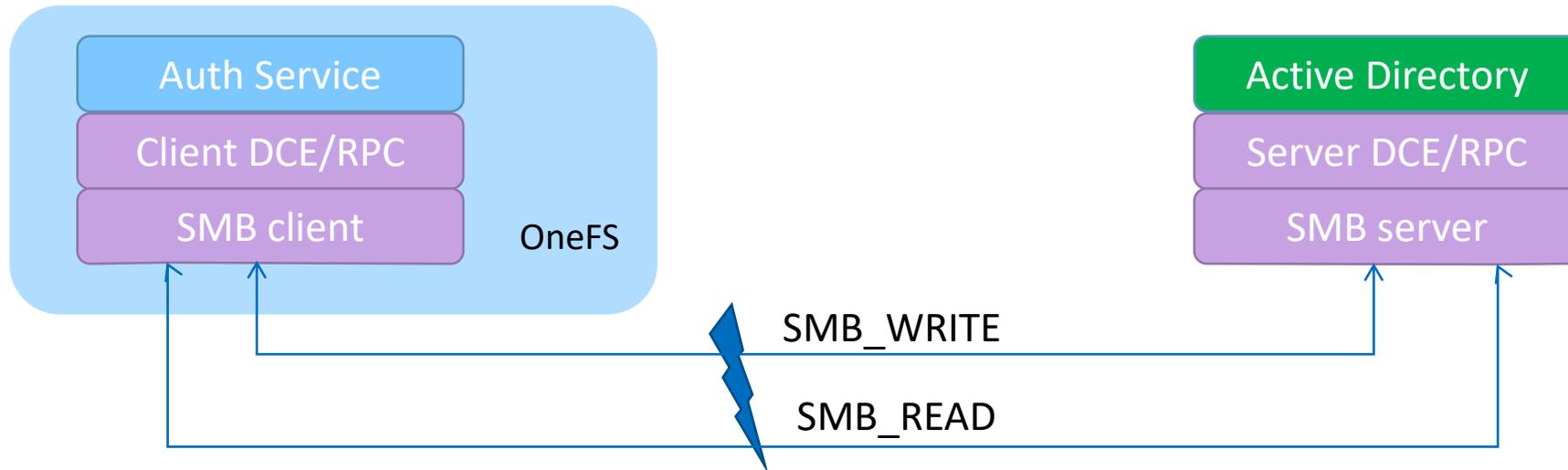


- The calls from DCE/RPC will still be synchronous
- Introduce timeouts at the SMB client layer (SMB rdr)
- Timeout information can be passed to SMB client from DCE/RPC
  - A separate data structure containing call timeouts

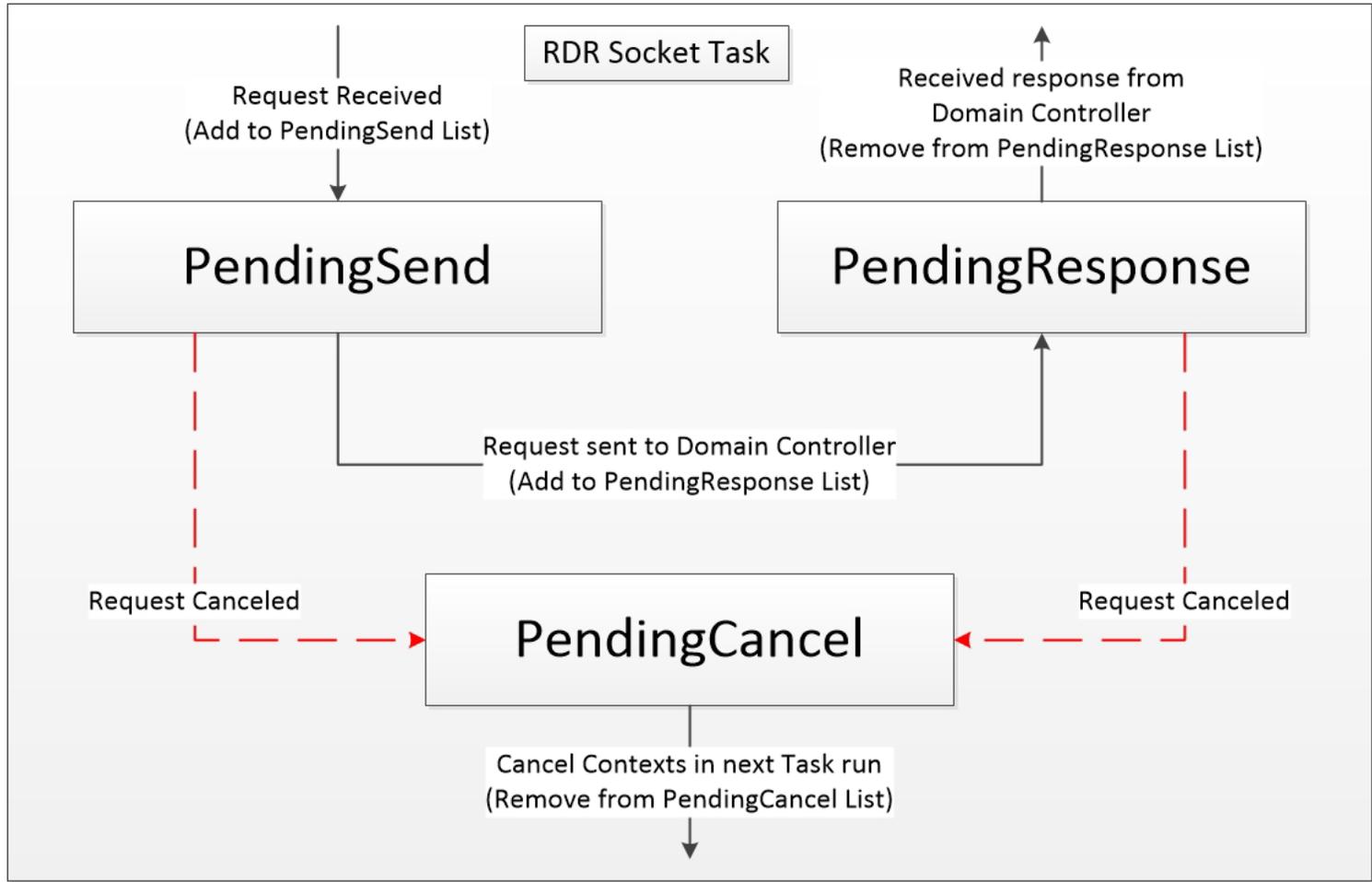


# Using SMB Client for Timeouts contd...

- Every DCE/RPC is translated into two network calls at SMB client layer
  - SMB WRITE: To send request to the AD server
  - SMB READ: To receive response from the AD server
- Creating connection requires additional SMB\_\* calls
- For most hung calls, SMB\_READ response is never received



# SMB Client (Redirector)



# SMB Client (Redirector)

## ■ SMB WRITE

- One network call to send the RPC request to server
- All calls pending ack are in “**PendingSend**” queue
- *Move the call from **PendingSend** to **PendingCancel** upon timeout*

## ■ SMB READ

- One network call to receive call response (for SMB WRITE) from server
- All calls pending response are in “**PendingResponse**” queue
- *Move the call from **PendingResponse** to **PendingCancel** upon timeout*

## ■ A separate thread to cancel all **PendingCancel** items

## ■ As of now, no cancel request is sent to AD

## ■ Delayed response will be treated as no-op

# Finishing Thoughts

- A timeout at SMB client layer achieves timed DCE/RPC calls without tinkering with the library
- More graceful
- Provides guarantee of timed calls irrespective of response
- Limited changes
  
- ROI is important

# Q & A

[Amit.Gaurav@dell.com](mailto:Amit.Gaurav@dell.com)