/HEXACON/

Akamai

Exploring Ancient Ruins to Find Modern Bugs

Stiv Kupchik & Ophir Harpaz

whoweare

Stiv Kupchik

Security Researcher

Akamai

@kupsul

Ophir Harpaz

Security Research team lead

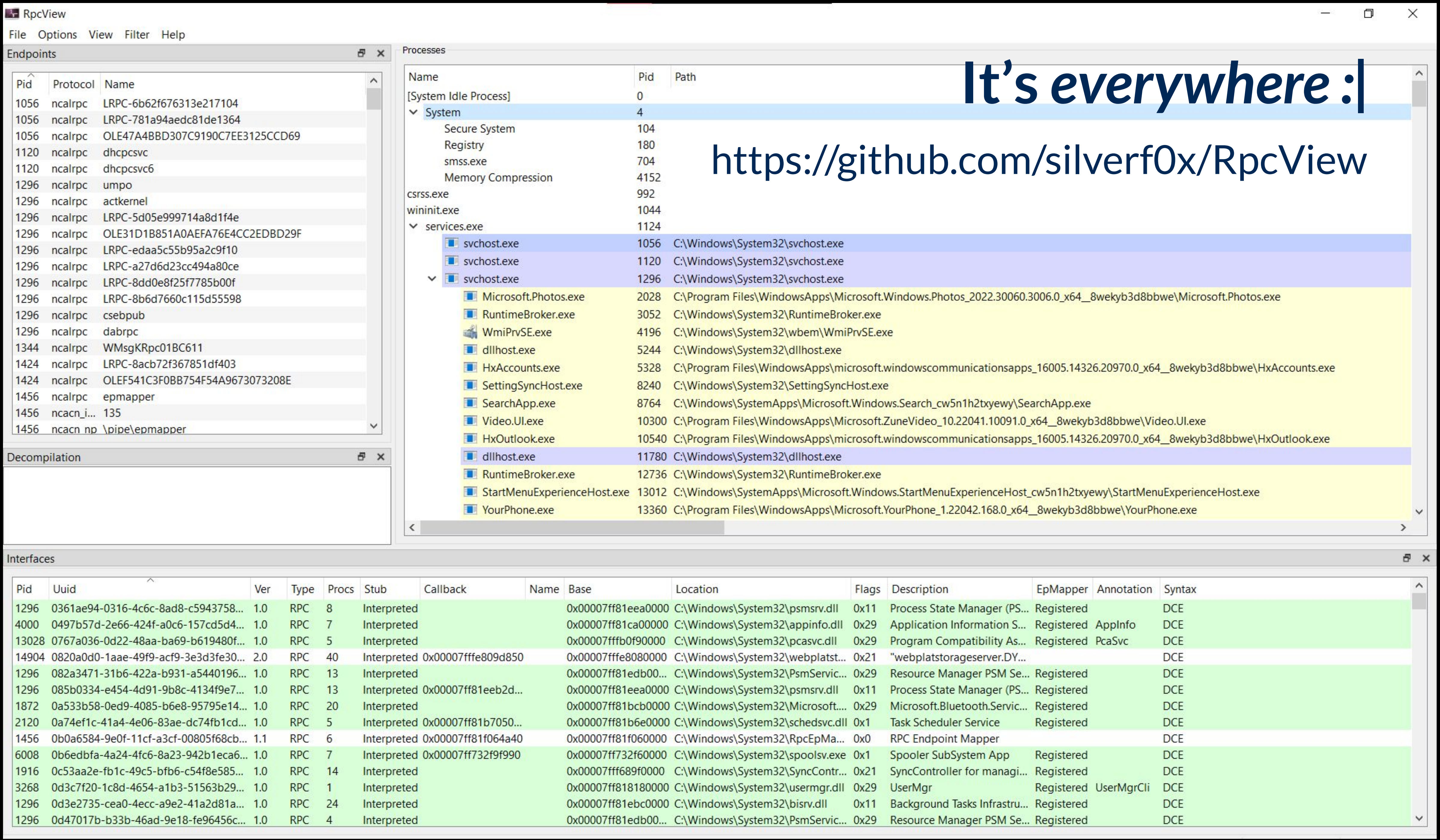
Akamai

@OphirHarpaz

Why MS-RPC?

It’s *everywhere* :|

https://github.com/silverf0x/RpcView



Endpoints

Pid	Protocol	Name
1056	ncalrpc	LRPC-6b62f676313e217104
1056	ncalrpc	LRPC-781a94aedc81de1364
1056	ncalrpc	OLE47A4BBD307C9190C7EE3125CCD69
1120	ncalrpc	dhcpcsvc
1120	ncalrpc	dhcpcsvc6
1296	ncalrpc	umpo
1296	ncalrpc	actkernel
1296	ncalrpc	LRPC-5d05e999714a8d1f4e
1296	ncalrpc	OLE31D1B851A0AEFA76E4CC2EDBD29F
1296	ncalrpc	LRPC-edaa5c55b95a2c9f10
1296	ncalrpc	LRPC-a27d6d23cc494a80ce
1296	ncalrpc	LRPC-8dd0e8f25f7785b00f
1296	ncalrpc	LRPC-8b6d7660c115d55598
1296	ncalrpc	csebpub
1296	ncalrpc	dabrpc
1344	ncalrpc	WMsgKRpc01BC611
1424	ncalrpc	LRPC-8acb72f367851df403
1424	ncalrpc	OLEF541C3F0BB754F54A9673073208E
1456	ncalrpc	epmapper
1456	ncacn_i...	135
1456	ncacn_np	\pipe\epmapper

Decompilation

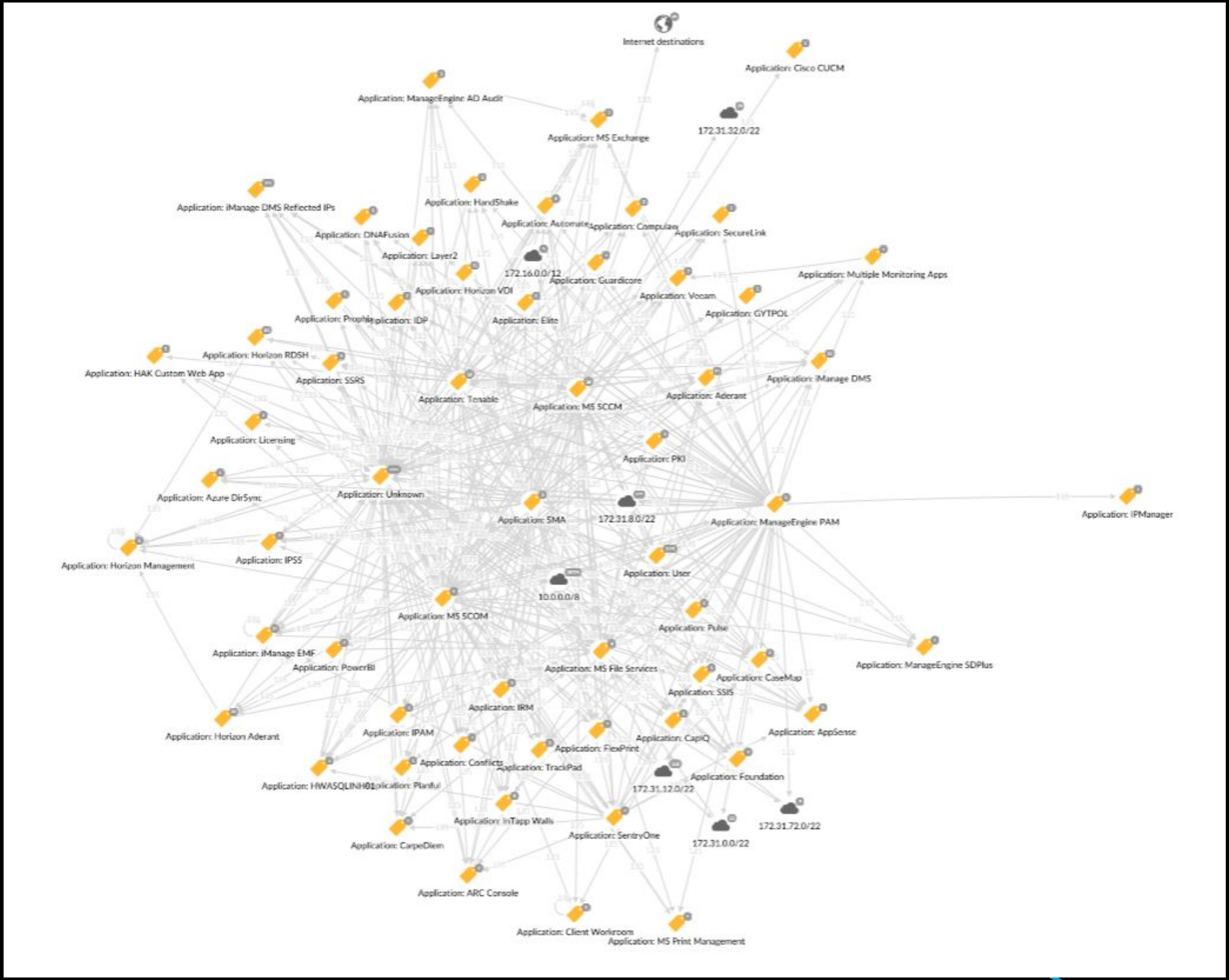
Processes

Name	Pid	Path
[System Idle Process]	0	
System	4	
Secure System	104	
Registry	180	
smss.exe	704	
Memory Compression	4152	
csrss.exe	992	
wininit.exe	1044	
services.exe	1124	
svchost.exe	1056	C:\Windows\System32\svchost.exe
svchost.exe	1120	C:\Windows\System32\svchost.exe
svchost.exe	1296	C:\Windows\System32\svchost.exe
Microsoft.Photos.exe	2028	C:\Program Files\WindowsApps\Microsoft.Windows.Photos_2022.30060.3006.0_x64__8wekyb3d8bbwe\Microsoft.Photos.exe
RuntimeBroker.exe	3052	C:\Windows\System32\RuntimeBroker.exe
WmiPrvSE.exe	4196	C:\Windows\System32\wbem\WmiPrvSE.exe
dllhost.exe	5244	C:\Windows\System32\dllhost.exe
HxAccounts.exe	5328	C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_16005.14326.20970.0_x64__8wekyb3d8bbwe\HxAccounts.exe
SettingSyncHost.exe	8240	C:\Windows\System32\SettingSyncHost.exe
SearchApp.exe	8764	C:\Windows\SystemApps\Microsoft.Windows.Search_cw5n1h2txyewy\SearchApp.exe
Video.UI.exe	10300	C:\Program Files\WindowsApps\Microsoft.ZuneVideo_10.22041.10091.0_x64__8wekyb3d8bbwe\Video.UI.exe
HxOutlook.exe	10540	C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_16005.14326.20970.0_x64__8wekyb3d8bbwe\HxOutlook.exe
dllhost.exe	11780	C:\Windows\System32\dllhost.exe
RuntimeBroker.exe	12736	C:\Windows\System32\RuntimeBroker.exe
StartMenuExperienceHost.exe	13012	C:\Windows\SystemApps\Microsoft.Windows.StartMenuExperienceHost_cw5n1h2txyewy\StartMenuExperienceHost.exe
YourPhone.exe	13360	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.22042.168.0_x64__8wekyb3d8bbwe\YourPhone.exe

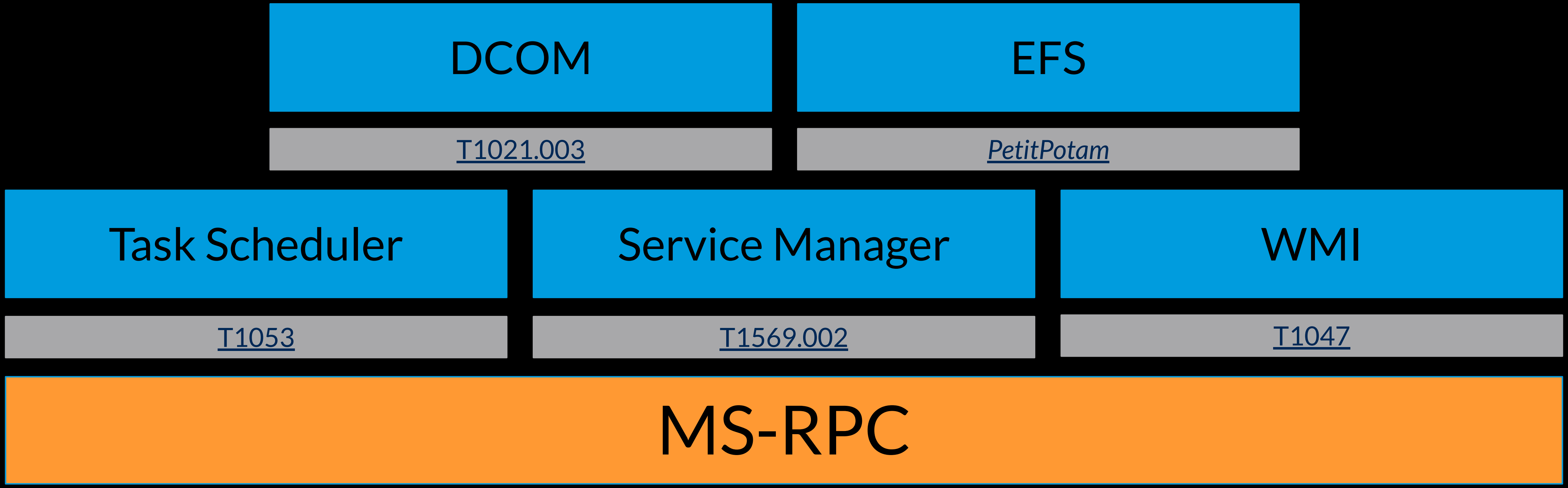
Interfaces

Pid	Uuid	Ver	Type	Procs	Stub	Callback	Name	Base	Location	Flags	Description	EpMapper	Annotation	Syntax
1296	0361ae94-0316-4c6c-8ad8-c5943758...	1.0	RPC	8	Interpreted			0x00007ff81eea0000	C:\Windows\System32\psmsrv.dll	0x11	Process State Manager (PS...	Registered		DCE
4000	0497b57d-2e66-424f-a0c6-157cd5d4...	1.0	RPC	7	Interpreted			0x00007ff81ca00000	C:\Windows\System32\appinfo.dll	0x29	Application Information S...	Registered	AppInfo	DCE
13028	0767a036-0d22-48aa-ba69-b619480f...	1.0	RPC	5	Interpreted			0x00007fffb0f90000	C:\Windows\System32\pcasvc.dll	0x29	Program Compatibility As...	Registered	PcaSvc	DCE
14904	0820a0d0-1aae-49f9-acf9-3e3d3fe30...	2.0	RPC	40	Interpreted	0x00007fffe809d850		0x00007fffe8080000	C:\Windows\System32\webplatst...	0x21	"webplatstorageserver.DY...			DCE
1296	082a3471-31b6-422a-b931-a5440196...	1.0	RPC	13	Interpreted			0x00007ff81edb00...	C:\Windows\System32\PsmServic...	0x29	Resource Manager PSM Se...	Registered		DCE
1296	085b0334-e454-4d91-9b8c-4134f9e7...	1.0	RPC	13	Interpreted	0x00007ff81eeb2d...		0x00007ff81eea0000	C:\Windows\System32\psmsrv.dll	0x11	Process State Manager (PS...	Registered		DCE
1872	0a533b58-0ed9-4085-b6e8-95795e14...	1.0	RPC	20	Interpreted			0x00007ff81bcb0000	C:\Windows\System32\Microsoft....	0x29	Microsoft.Bluetooth.Servic...	Registered		DCE
2120	0a74ef1c-41a4-4e06-83ae-dc74fb1cd...	1.0	RPC	5	Interpreted	0x00007ff81b7050...		0x00007ff81b6e0000	C:\Windows\System32\schedsvc.dll	0x1	Task Scheduler Service	Registered		DCE
1456	0b0a6584-9e0f-11cf-a3cf-00805f68cb...	1.1	RPC	6	Interpreted	0x00007ff81f064a40		0x00007ff81f060000	C:\Windows\System32\RpcEpMa...	0x0	RPC Endpoint Mapper			DCE
6008	0b6edbfa-4a24-4fc6-8a23-942b1eca6...	1.0	RPC	7	Interpreted	0x00007ff732f9f990		0x00007ff732f60000	C:\Windows\System32\spoolsv.exe	0x1	Spooler SubSystem App	Registered		DCE
1916	0c53aa2e-fb1c-49c5-bfb6-c54f8e585...	1.0	RPC	14	Interpreted			0x00007fff689f0000	C:\Windows\System32\SyncContr...	0x21	SyncController for managi...	Registered		DCE
3268	0d3c7f20-1c8d-4654-a1b3-51563b29...	1.0	RPC	1	Interpreted			0x00007ff818180000	C:\Windows\System32\usermgr.dll	0x29	UserMgr	Registered	UserMgrCli	DCE
1296	0d3e2735-cea0-4ecc-a9e2-41a2d81a...	1.0	RPC	24	Interpreted			0x00007ff81ebc0000	C:\Windows\System32\bisrv.dll	0x11	Background Tasks Infrastru...	Registered		DCE
1296	0d47017b-b33b-46ad-9e18-fe96456c...	1.0	RPC	4	Interpreted			0x00007ff81edb00...	C:\Windows\System32\PsmServic...	0x29	Resource Manager PSM Se...	Registered		DCE

… and between everyone



Network Attacks Over MS-RPC



DCOM

T1021.003

EFS

PetitPotam

Task Scheduler

T1053

Service Manager

T1569.002

WMI

T1047

MS-RPC

Yet not much public research

Most information boils down to:

- MSFT documentation
- Several research-oriented blog posts
- Few public vulnerabilities

Why so?

Potential impact:

Lateral Movement & Privilege Escalation

Our agenda for today

❏ MS-RPC introduction and overview

❏ Security flaws in MS-RPC

❏ Automating our RPC research

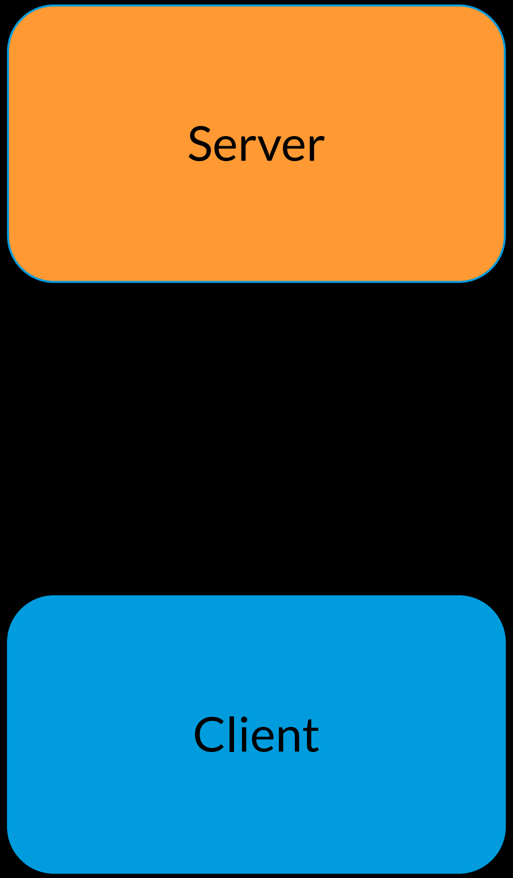
❏ A 0-day in a Windows service

MS-RPC Overview

Terminology you’ll soon master

- Interface
- {M}IDL
- Transport
- Endpoint
- Binding

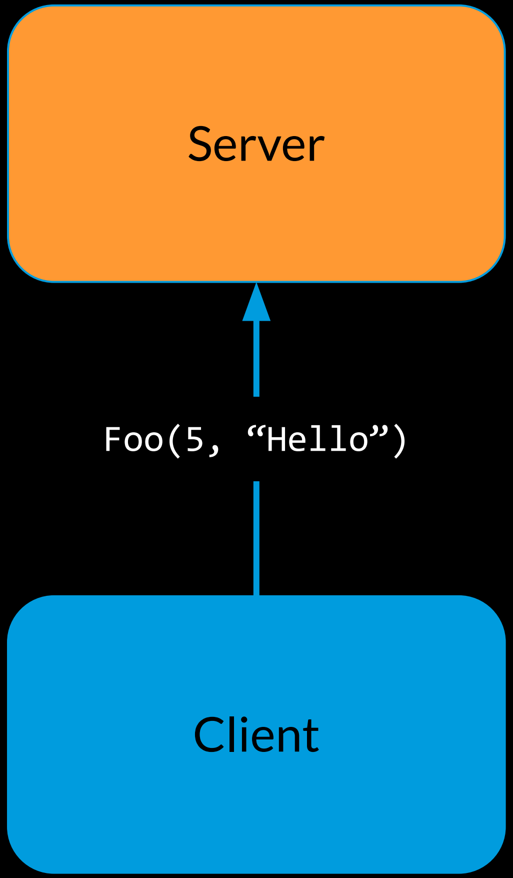
The RPC Client-Server Model



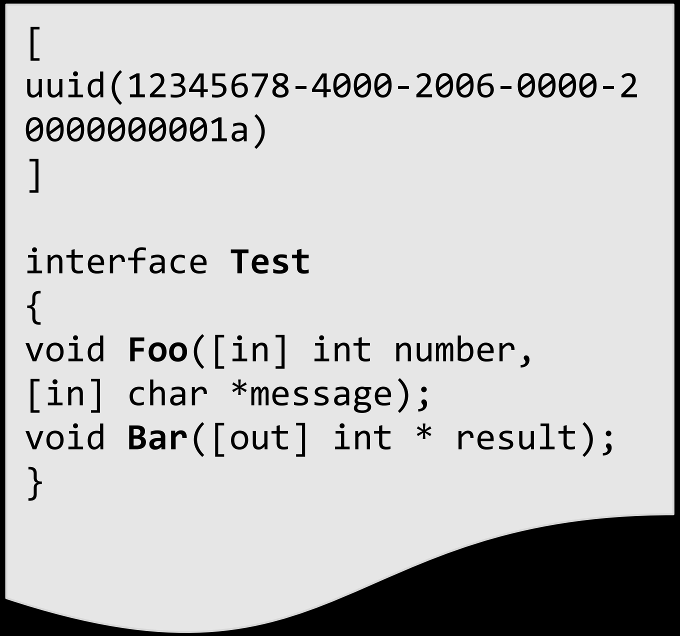
Server

Client

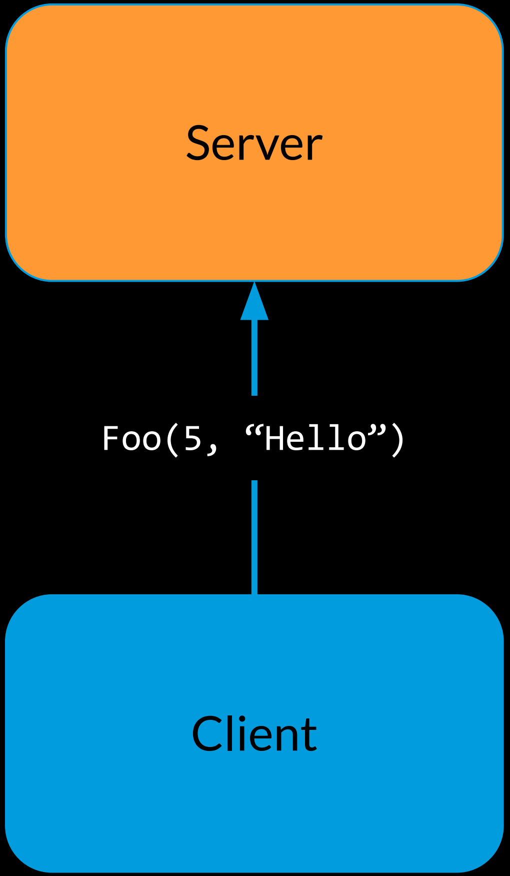
The RPC Client-Server Model



The RPC Client-Server Model



```
[
uuid(12345678-4000-2006-0000-2
0000000001a)
]

interface Test
{
void Foo([in] int number,
[in] char *message);
void Bar([out] int * result);
}
```

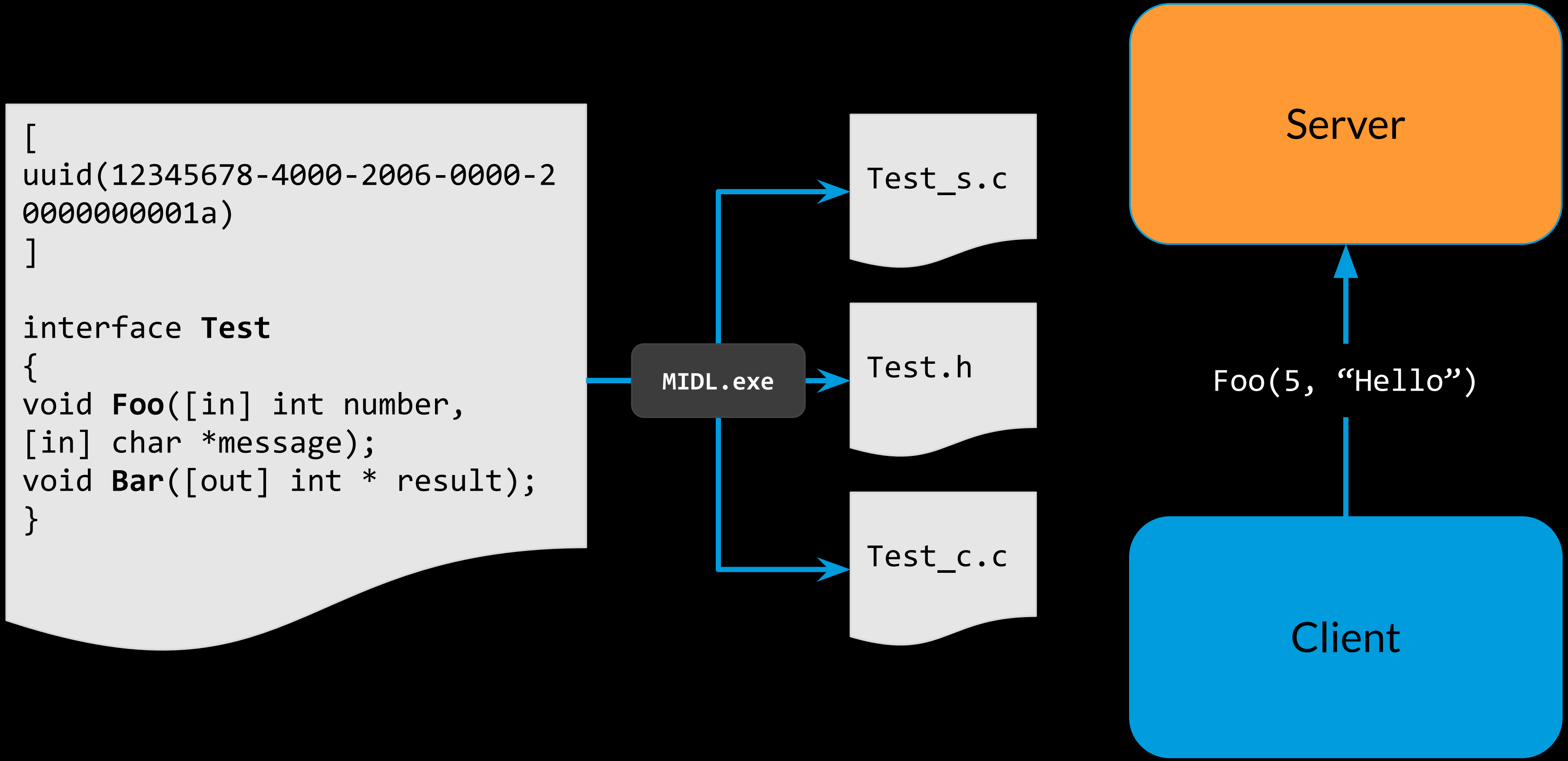


Server

Foo(5, “Hello”)

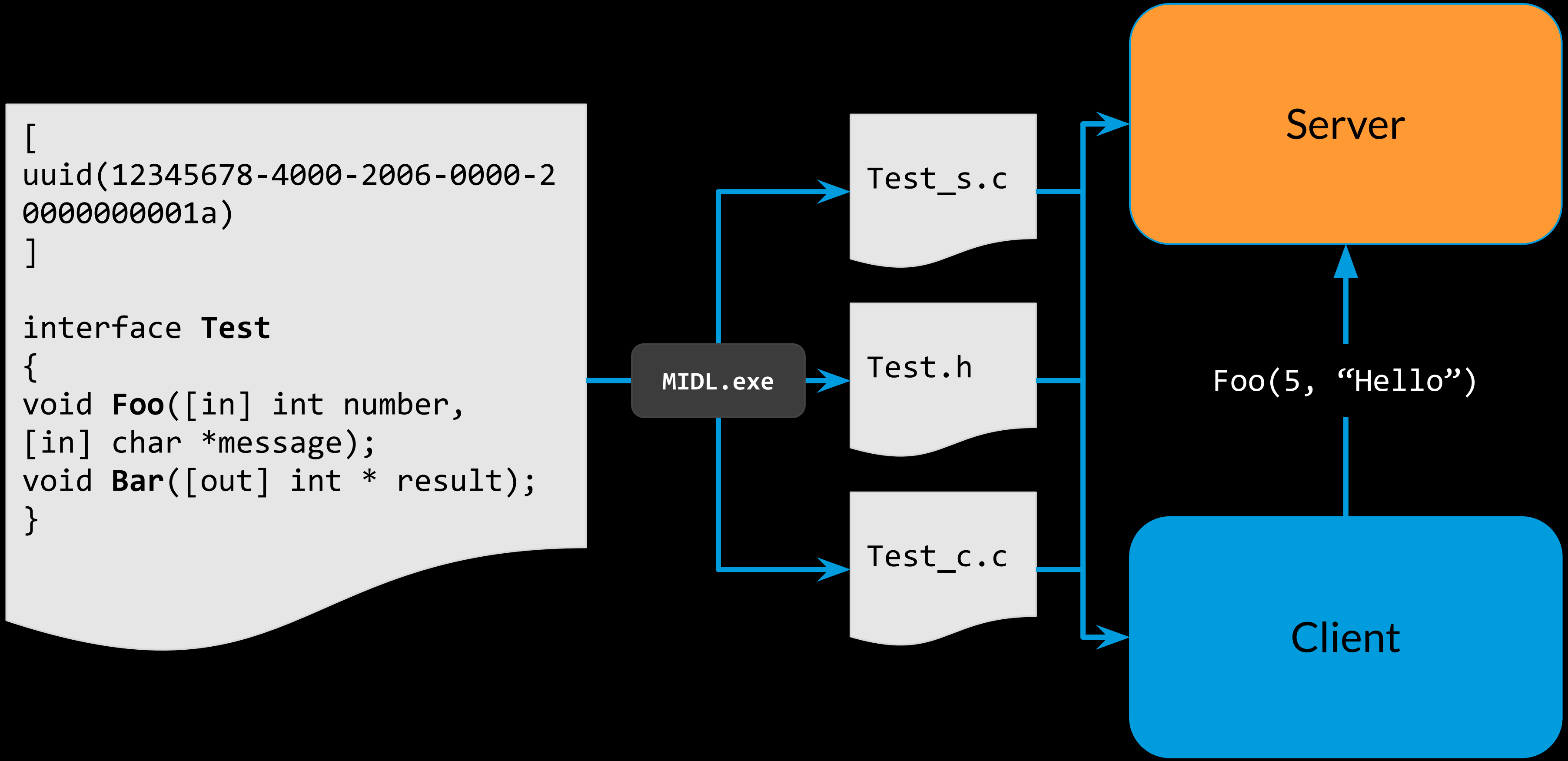
Client

The RPC Client-Server Model



```
[
uuid(12345678-4000-2006-0000-2
0000000001a)
]

interface Test
{
void Foo([in] int number,
[in] char *message);
void Bar([out] int * result);
}
```

The RPC Client-Server Model



```
[
uuid(12345678-4000-2006-0000-2
0000000001a)
]

interface Test
{
void Foo([in] int number,
[in] char *message);
void Bar([out] int * result);
}
```

Endpoints

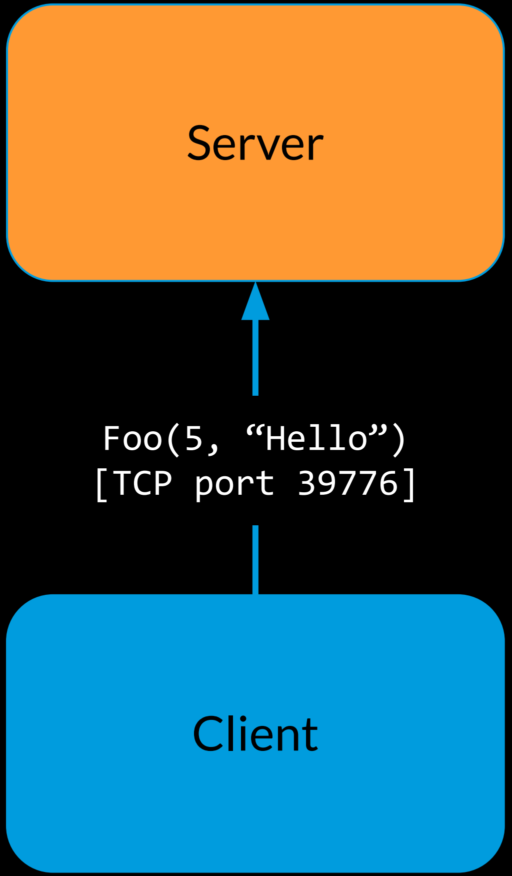
- The server registers an *endpoint* using a certain *transport*

Transports	Protocol Sequence	Endpoints
TCP	`ncacn_ip_tcp`	<port number>
Named pipe	`ncacn_np`	<pipe name>
UDP	`ncadg_ip_udp`	<port number>
ALPC	`ncalrpc`	<ALPC port>
HTTP	`ncacn_http`	<hostname>
Hyper-V socket	`ncacn_hvsocket`	<UUID>

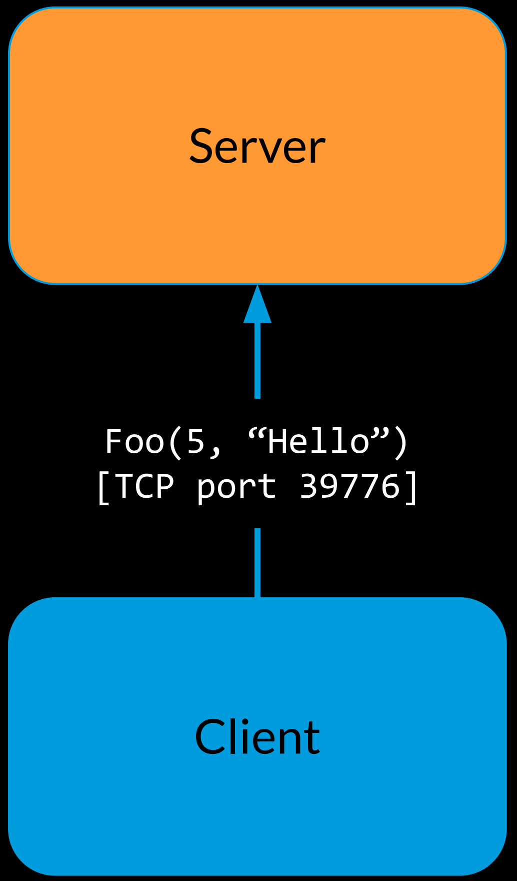
- Interfaces and endpoints are registered separately

Well-Known Endpoints

Dynamic Endpoints



Well-Known Endpoints

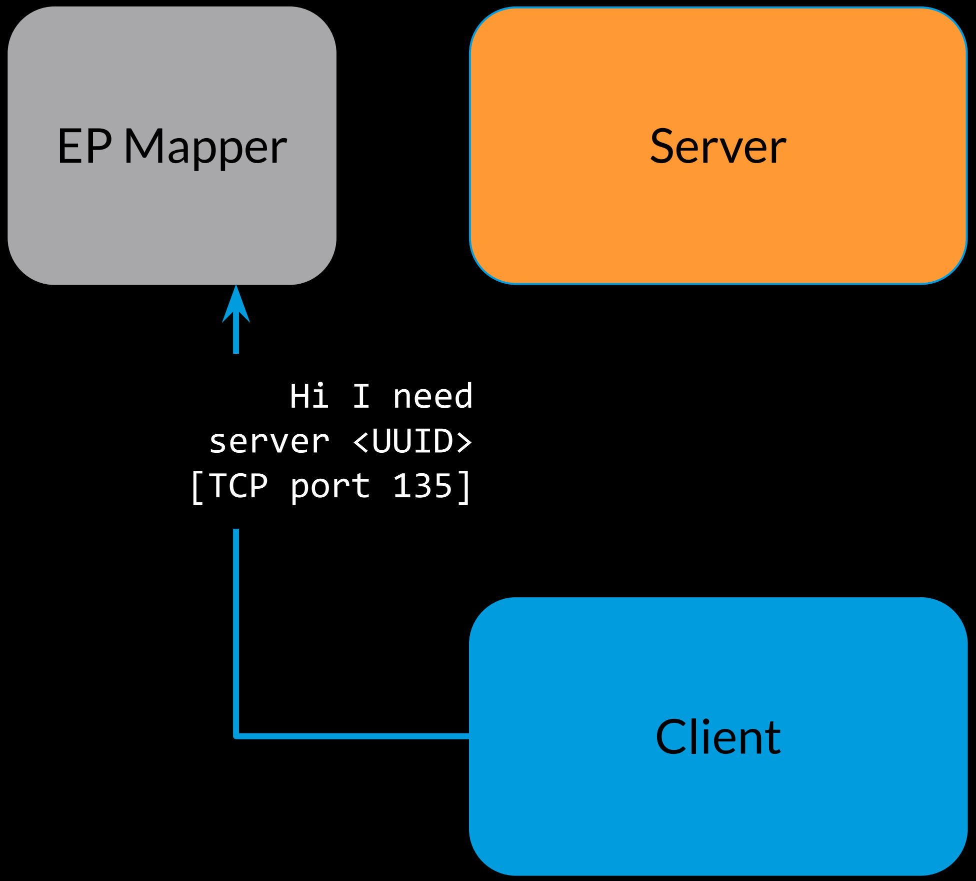


Server

Foo(5, “Hello”)
[TCP port 39776]

Client

Dynamic Endpoints



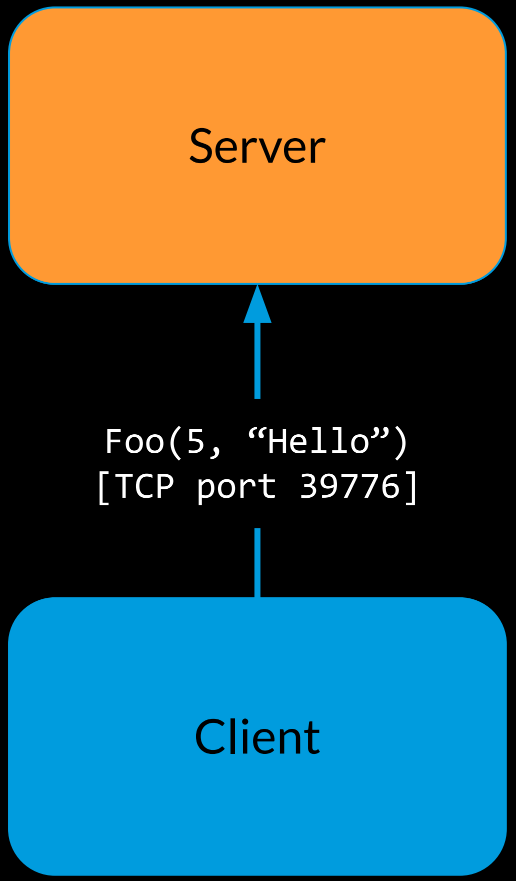
EP Mapper

Server

Hi I need
server <UUID>
[TCP port 135]

Client

Well-Known Endpoints

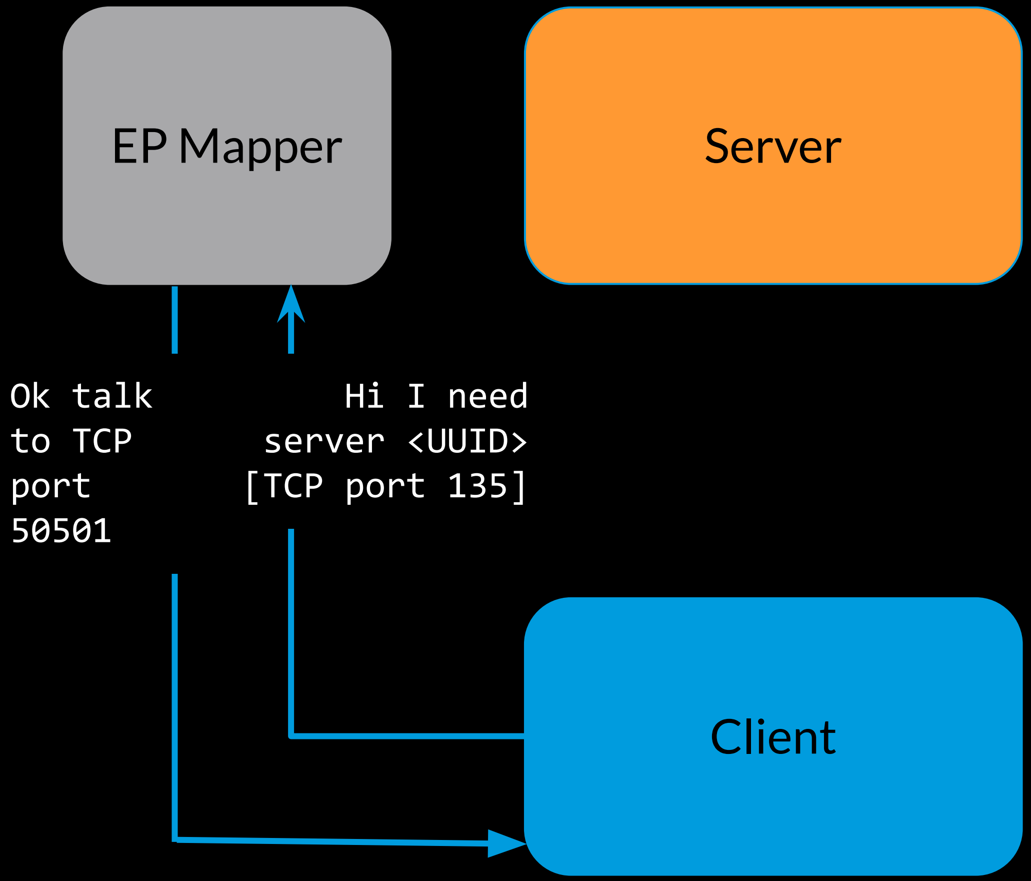


Server

Foo(5, “Hello”)
[TCP port 39776]

Client

Dynamic Endpoints



EP Mapper

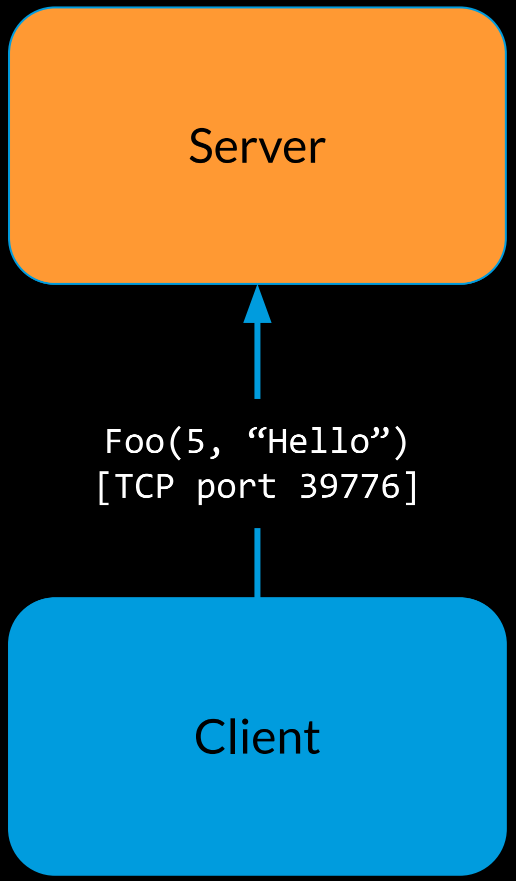
Server

Ok talk
to TCP
port
50501

Hi I need
server <UUID>
[TCP port 135]

Client

Well-Known Endpoints

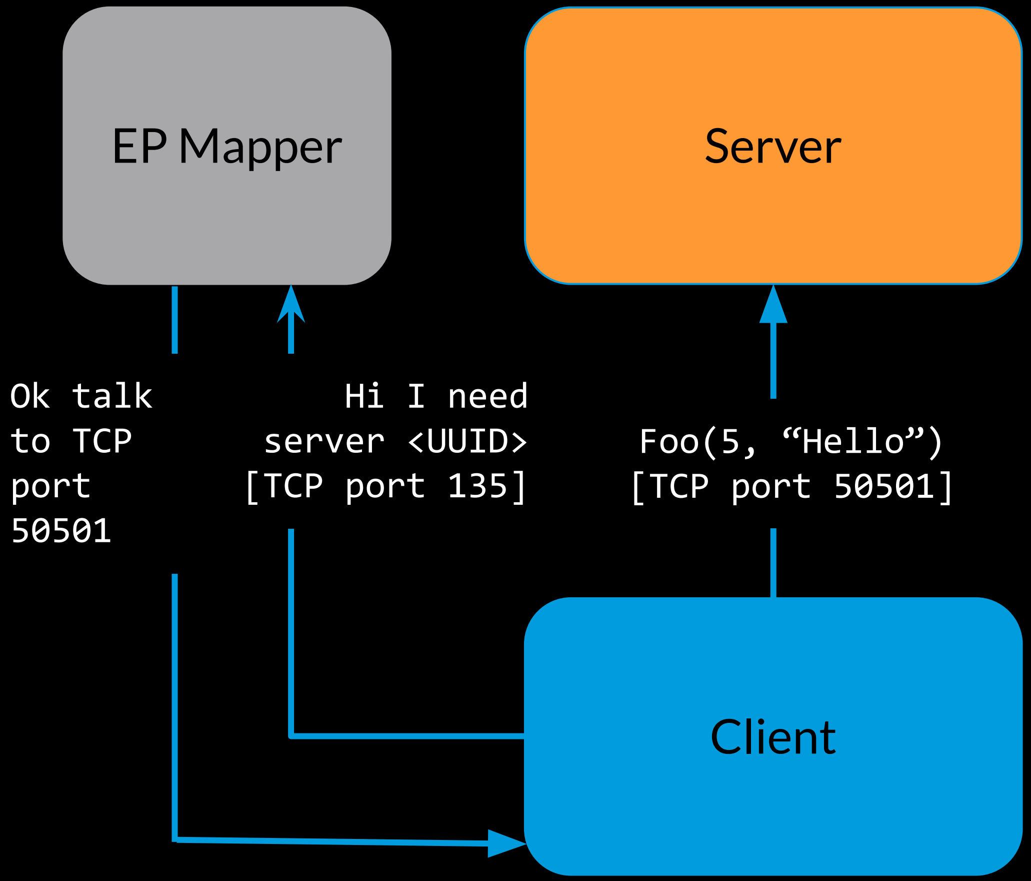


Server

Foo(5, “Hello”)
[TCP port 39776]

Client

Dynamic Endpoints



EP Mapper

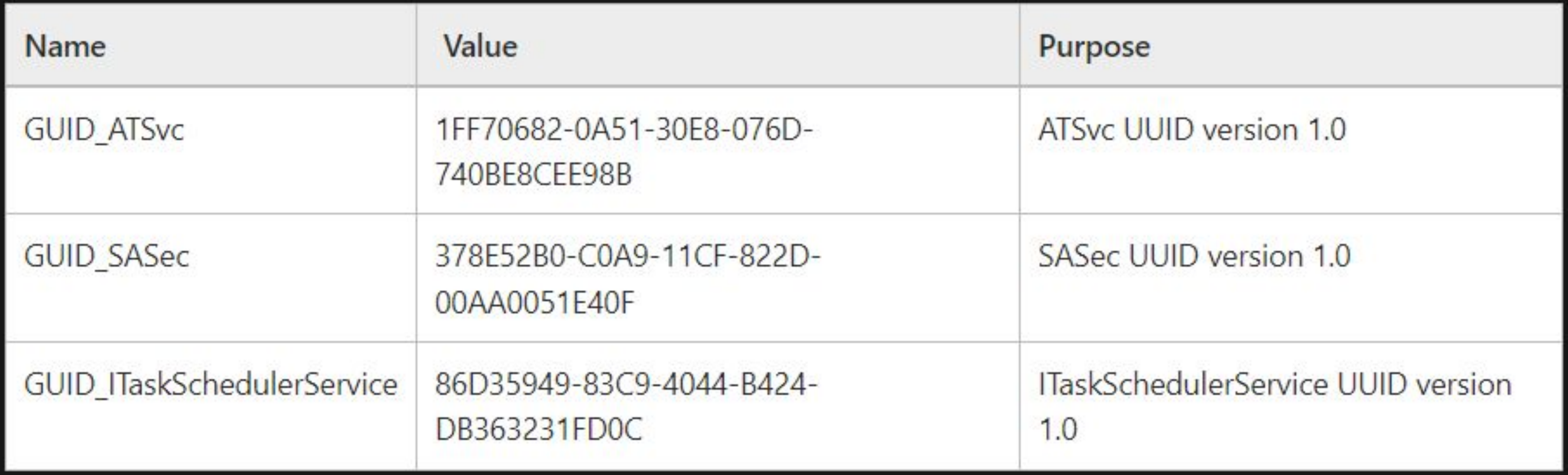
Server

Ok talk
to TCP
port
50501

Hi I need
server <UUID>
[TCP port 135]

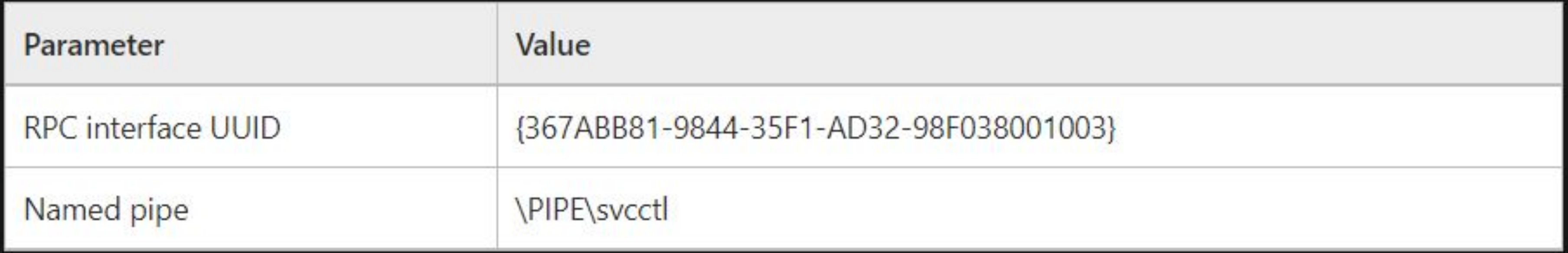
Foo(5, “Hello”)
[TCP port 50501]

Client



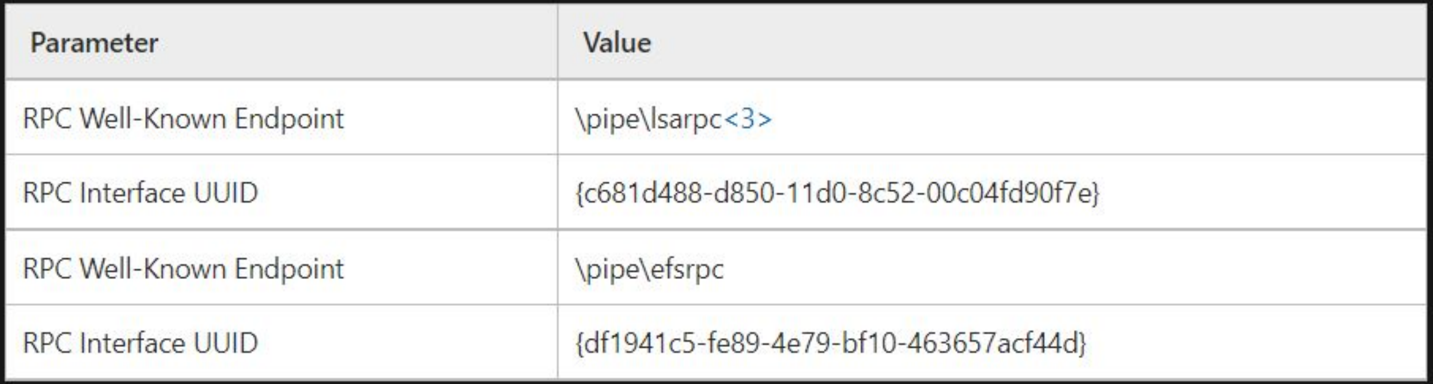
Name	Value	Purpose
GUID_ATSvc	1FF70682-0A51-30E8-076D-740BE8CEE98B	ATSvc UUID version 1.0
GUID_SASec	378E52B0-C0A9-11CF-822D-00AA0051E40F	SASec UUID version 1.0
GUID_ITaskSchedulerService	86D35949-83C9-4044-B424-DB363231FD0C	ITaskSchedulerService UUID version 1.0

Task Scheduler Service Remoting Protocol



Parameter	Value
RPC interface UUID	{367ABB81-9844-35F1-AD32-98F038001003}
Named pipe	\PIPE\svcctl

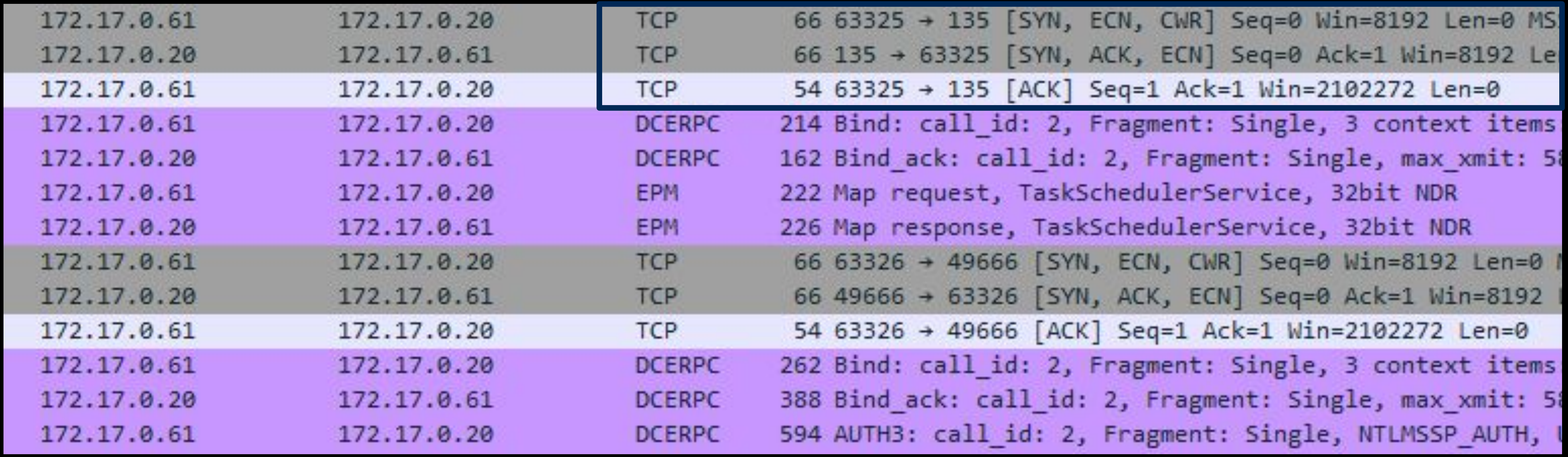
Service control manager remote protocol



Parameter	Value
RPC Well-Known Endpoint	\pipe\lsarpc<3>
RPC Interface UUID	{c681d488-d850-11d0-8c52-00c04fd90f7e}
RPC Well-Known Endpoint	\pipe\efsrpc
RPC Interface UUID	{df1941c5-fe89-4e79-bf10-463657acf44d}

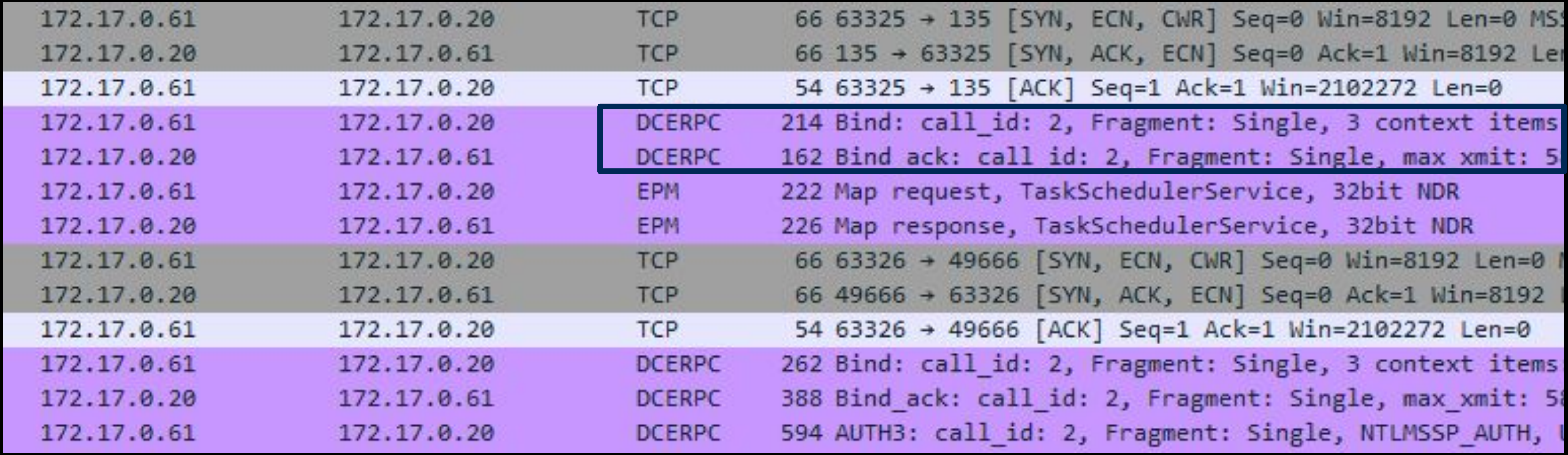
Encrypting File System Remote (EFSRPC) Protocol

Task Scheduler Endpoint Resolution



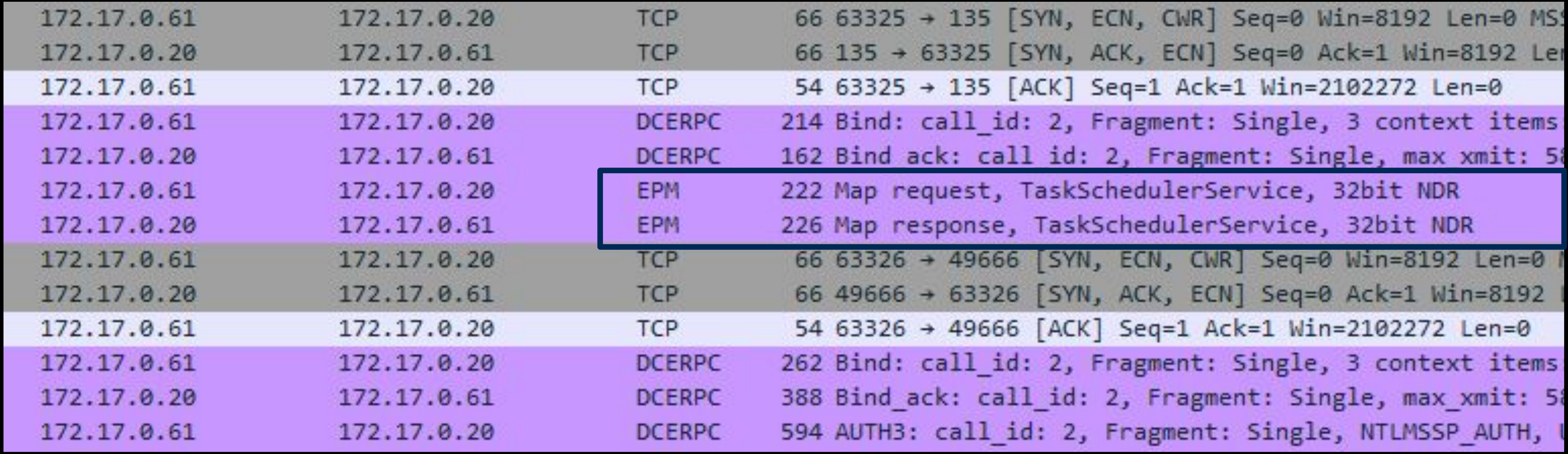
```
172.17.0.61    172.17.0.20    TCP       66 63325 → 135 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MS
172.17.0.20    172.17.0.61    TCP       66 135 → 63325 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Le
172.17.0.61    172.17.0.20    TCP       54 63325 → 135 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC   214 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC   162 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 58
172.17.0.61    172.17.0.20    EPM      222 Map request, TaskSchedulerService, 32bit NDR
172.17.0.20    172.17.0.61    EPM      226 Map response, TaskSchedulerService, 32bit NDR
172.17.0.61    172.17.0.20    TCP       66 63326 → 49666 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0
172.17.0.20    172.17.0.61    TCP       66 49666 → 63326 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192
172.17.0.61    172.17.0.20    TCP       54 63326 → 49666 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC   262 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC   388 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 58
172.17.0.61    172.17.0.20    DCERPC   594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH,
```

Task Scheduler Endpoint Resolution



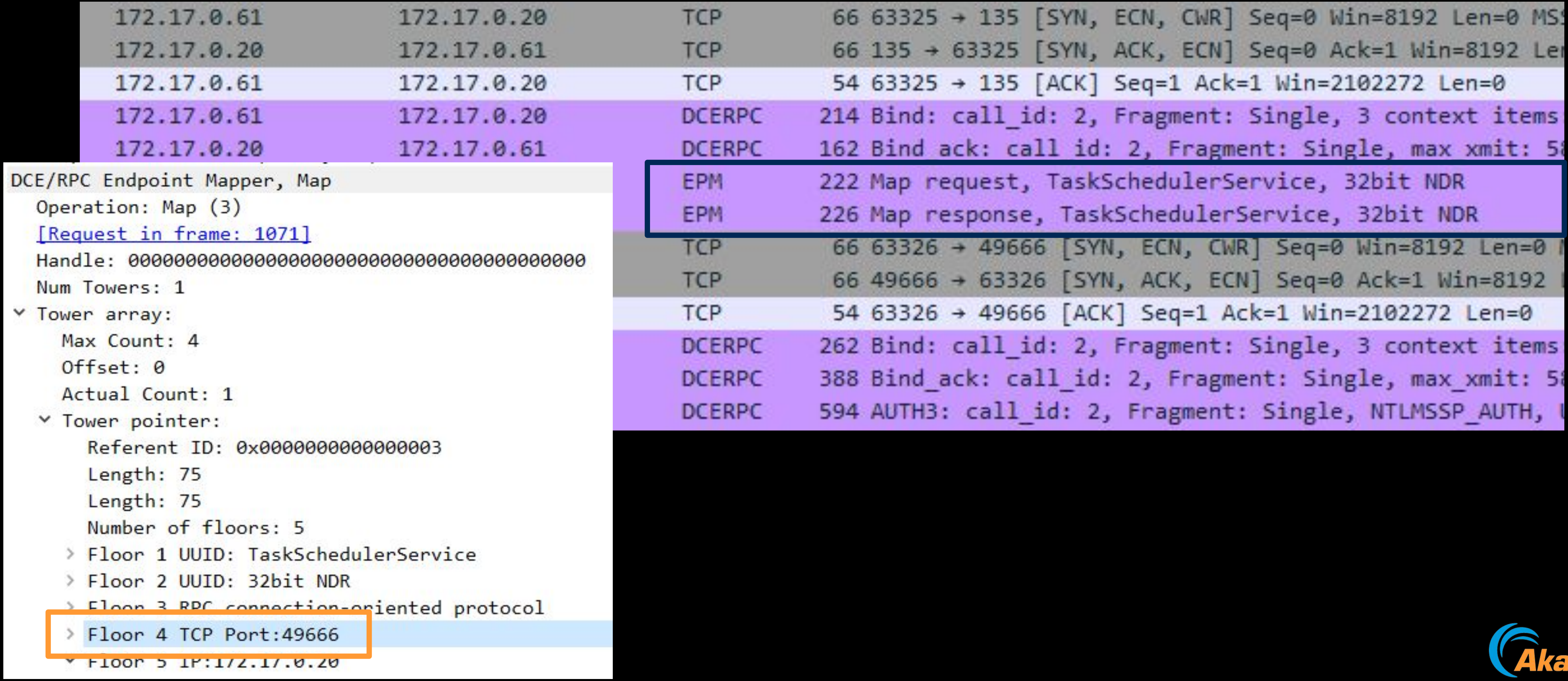
```
172.17.0.61    172.17.0.20    TCP       66 63325 → 135 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MS
172.17.0.20    172.17.0.61    TCP       66 135 → 63325 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Le
172.17.0.61    172.17.0.20    TCP       54 63325 → 135 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC   214 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC   162 Bind ack: call id: 2, Fragment: Single, max xmit: 5
172.17.0.61    172.17.0.20    EPM      222 Map request, TaskSchedulerService, 32bit NDR
172.17.0.20    172.17.0.61    EPM      226 Map response, TaskSchedulerService, 32bit NDR
172.17.0.61    172.17.0.20    TCP       66 63326 → 49666 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0
172.17.0.20    172.17.0.61    TCP       66 49666 → 63326 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192
172.17.0.61    172.17.0.20    TCP       54 63326 → 49666 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC   262 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC   388 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5
172.17.0.61    172.17.0.20    DCERPC   594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH,
```

Task Scheduler Endpoint Resolution



```
172.17.0.61    172.17.0.20    TCP       66 63325 → 135 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MS
172.17.0.20    172.17.0.61    TCP       66 135 → 63325 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Le
172.17.0.61    172.17.0.20    TCP       54 63325 → 135 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC   214 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC   162 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5
172.17.0.61    172.17.0.20    EPM      222 Map request, TaskSchedulerService, 32bit NDR
172.17.0.20    172.17.0.61    EPM      226 Map response, TaskSchedulerService, 32bit NDR
172.17.0.61    172.17.0.20    TCP       66 63326 → 49666 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0
172.17.0.20    172.17.0.61    TCP       66 49666 → 63326 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192
172.17.0.61    172.17.0.20    TCP       54 63326 → 49666 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC   262 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC   388 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5
172.17.0.61    172.17.0.20    DCERPC   594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH,
```

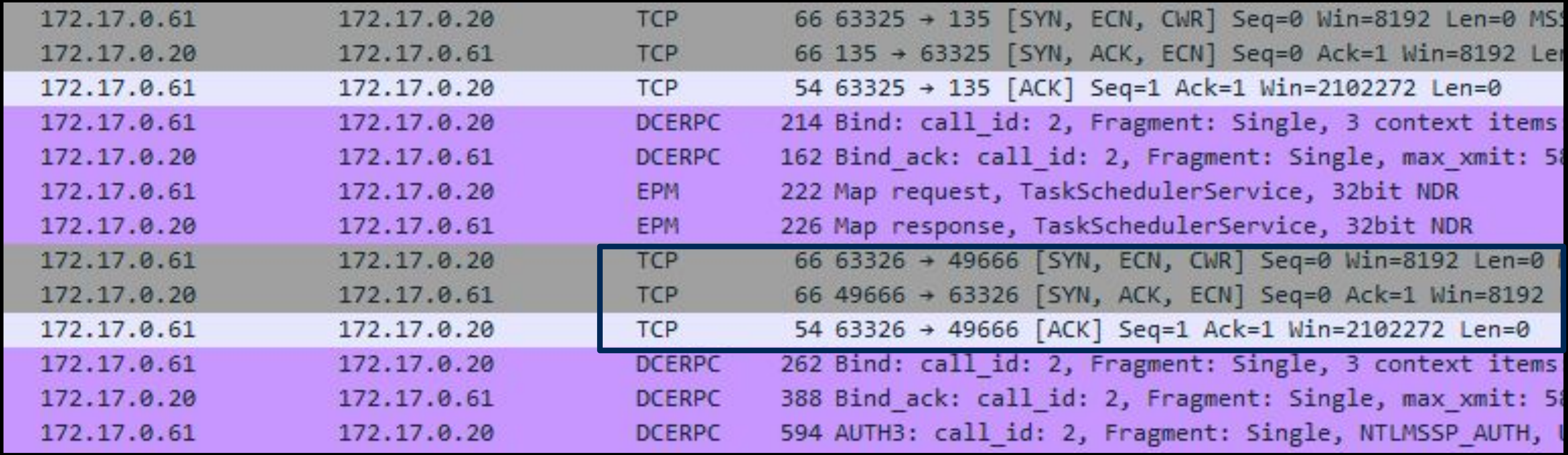
Task Scheduler Endpoint Resolution



```
172.17.0.61     172.17.0.20     TCP       66 63325 → 135 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MSS
172.17.0.20     172.17.0.61     TCP       66 135 → 63325 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Len
172.17.0.61     172.17.0.20     TCP       54 63325 → 135 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61     172.17.0.20     DCERPC   214 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20     172.17.0.61     DCERPC   162 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 58
                                EPM      222 Map request, TaskSchedulerService, 32bit NDR
                                EPM      226 Map response, TaskSchedulerService, 32bit NDR
                                TCP       66 63326 → 49666 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0
                                TCP       66 49666 → 63326 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192
                                TCP       54 63326 → 49666 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
                                DCERPC   262 Bind: call_id: 2, Fragment: Single, 3 context items
                                DCERPC   388 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5
                                DCERPC   594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH,
```

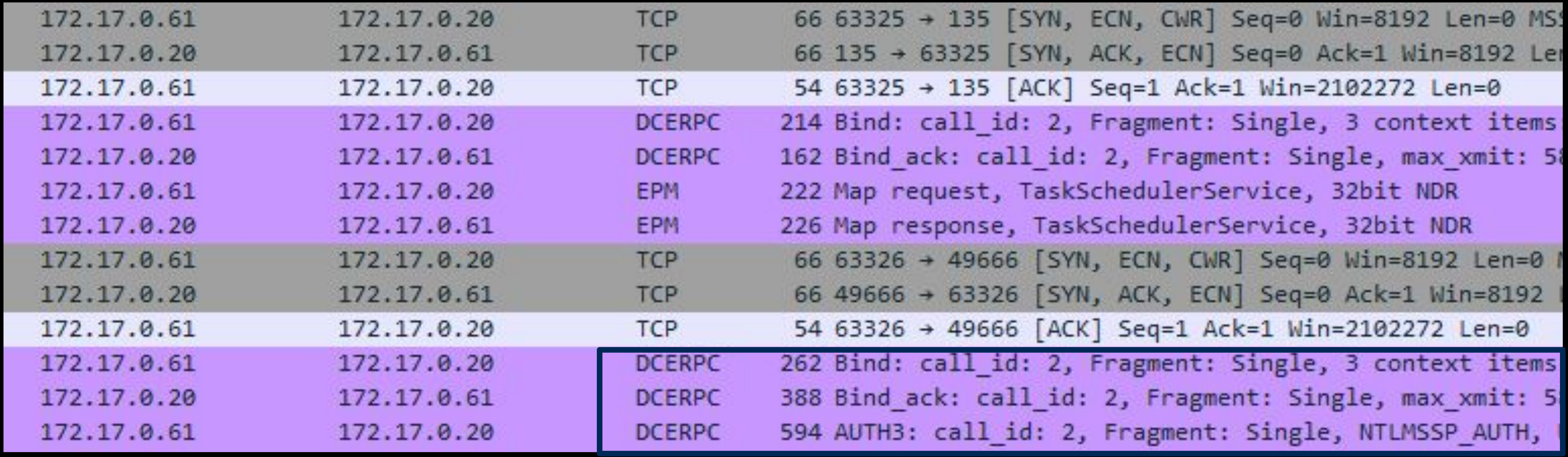
```
DCE/RPC Endpoint Mapper, Map
   Operation: Map (3)
   [Request in frame: 1071]
   Handle: 0000000000000000000000000000000000000000
   Num Towers: 1
 ˅ Tower array:
     Max Count: 4
     Offset: 0
     Actual Count: 1
   ˅ Tower pointer:
       Referent ID: 0x0000000000000003
       Length: 75
       Length: 75
       Number of floors: 5
     › Floor 1 UUID: TaskSchedulerService
     › Floor 2 UUID: 32bit NDR
     › Floor 3 RPC connection-oriented protocol
     › Floor 4 TCP Port:49666
     ˅ Floor 5 IP:172.17.0.20
```

Task Scheduler Endpoint Resolution



```
172.17.0.61    172.17.0.20    TCP      66 63325 → 135 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MS
172.17.0.20    172.17.0.61    TCP      66 135 → 63325 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Le
172.17.0.61    172.17.0.20    TCP      54 63325 → 135 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC  214 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC  162 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5
172.17.0.61    172.17.0.20    EPM     222 Map request, TaskSchedulerService, 32bit NDR
172.17.0.20    172.17.0.61    EPM     226 Map response, TaskSchedulerService, 32bit NDR
172.17.0.61    172.17.0.20    TCP      66 63326 → 49666 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0
172.17.0.20    172.17.0.61    TCP      66 49666 → 63326 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192
172.17.0.61    172.17.0.20    TCP      54 63326 → 49666 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC  262 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC  388 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5
172.17.0.61    172.17.0.20    DCERPC  594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH,
```

Task Scheduler Endpoint Resolution

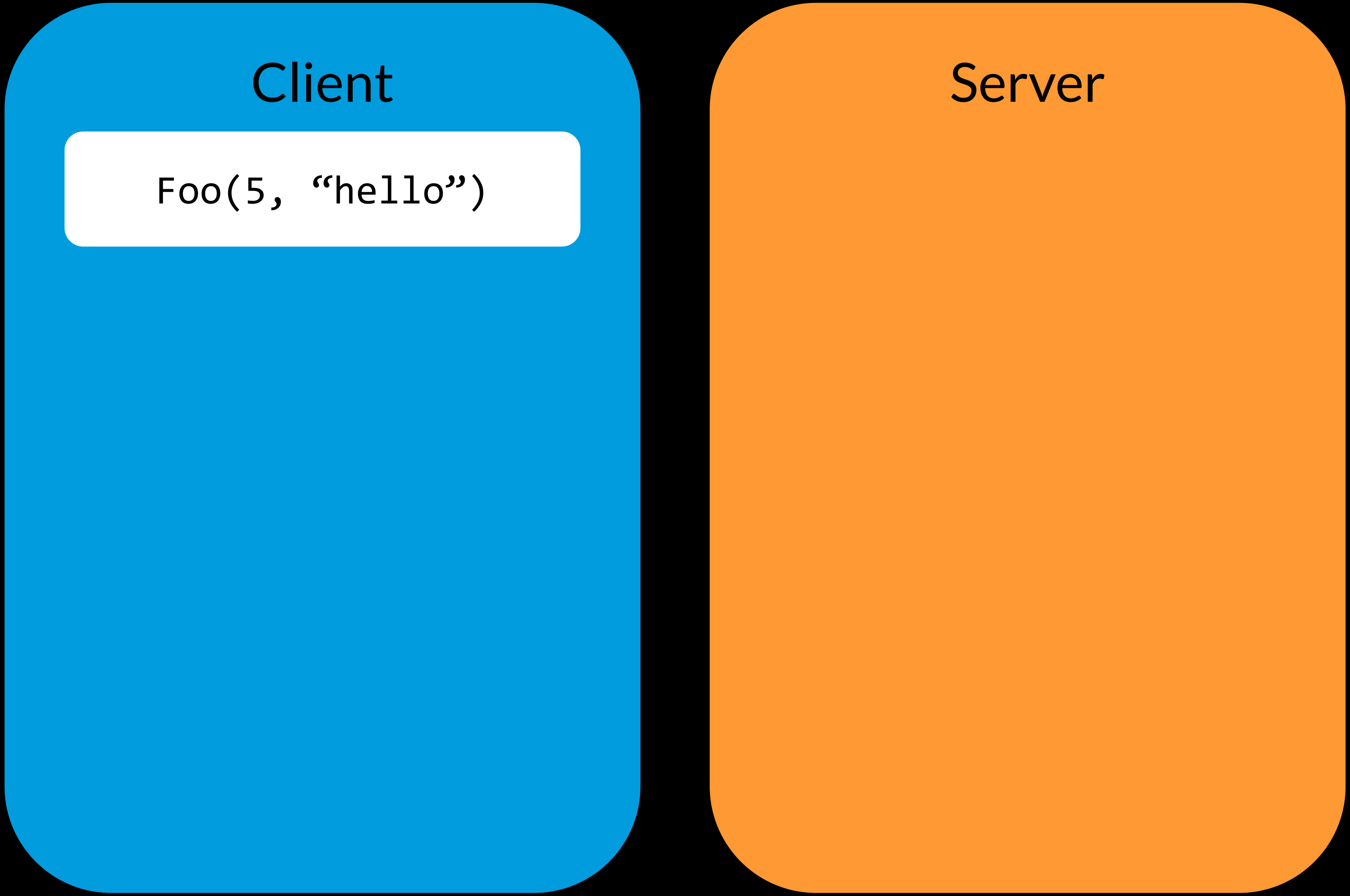


```
172.17.0.61    172.17.0.20    TCP       66 63325 → 135 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MS
172.17.0.20    172.17.0.61    TCP       66 135 → 63325 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Le
172.17.0.61    172.17.0.20    TCP       54 63325 → 135 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC   214 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC   162 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5
172.17.0.61    172.17.0.20    EPM      222 Map request, TaskSchedulerService, 32bit NDR
172.17.0.20    172.17.0.61    EPM      226 Map response, TaskSchedulerService, 32bit NDR
172.17.0.61    172.17.0.20    TCP       66 63326 → 49666 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0
172.17.0.20    172.17.0.61    TCP       66 49666 → 63326 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192
172.17.0.61    172.17.0.20    TCP       54 63326 → 49666 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
172.17.0.61    172.17.0.20    DCERPC   262 Bind: call_id: 2, Fragment: Single, 3 context items
172.17.0.20    172.17.0.61    DCERPC   388 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5
172.17.0.61    172.17.0.20    DCERPC   594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH,
```

Binding

- The representation of a session between a client and a server
  - Practically, a handle
  - Client and server can manipulate binding data using designated functions
  - Used for authentication (among other things)

An RPC Call’s Flow

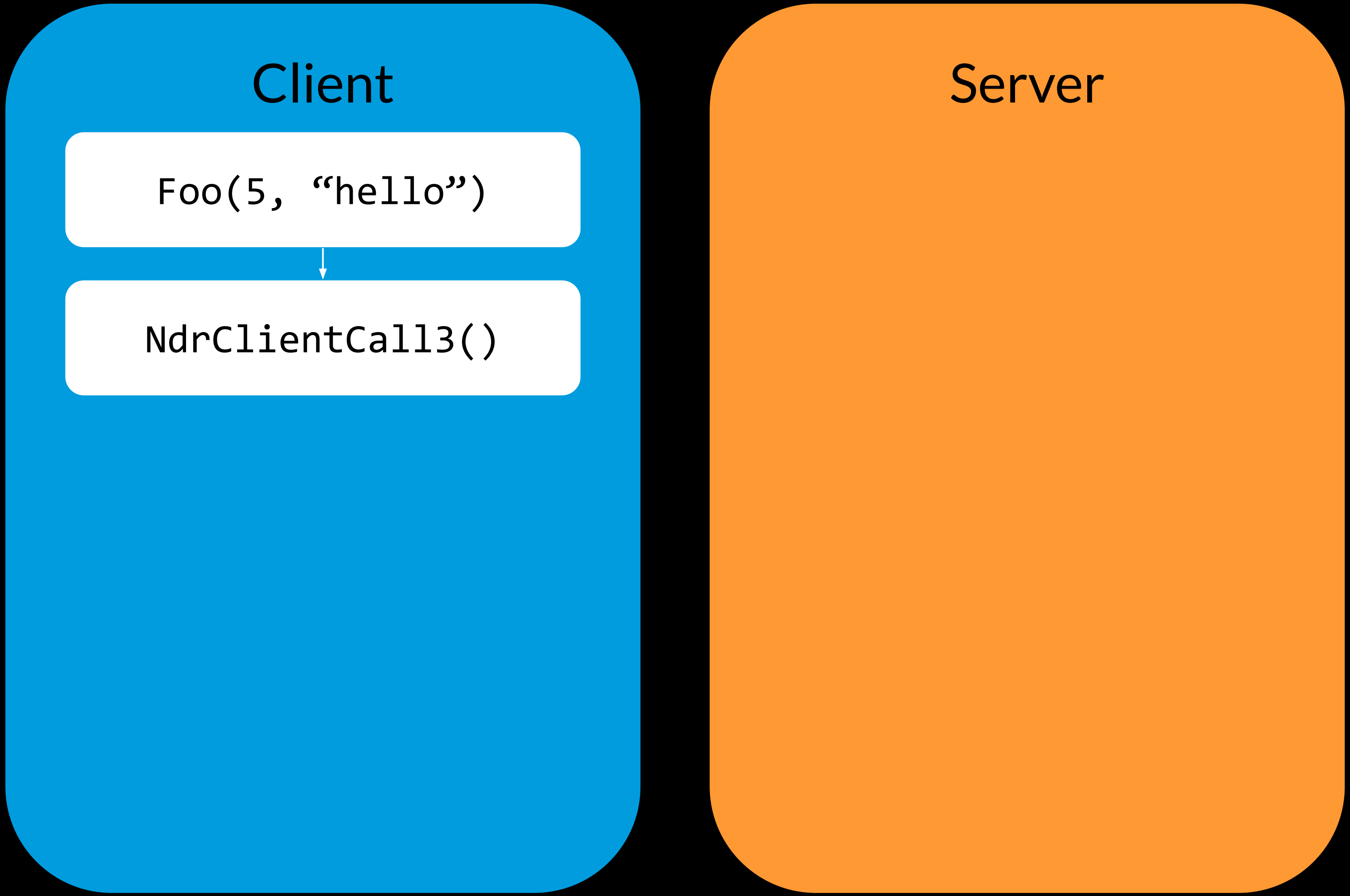


Client

```
Foo(5, “hello”)
```

Server

An RPC Call's Flow



Client

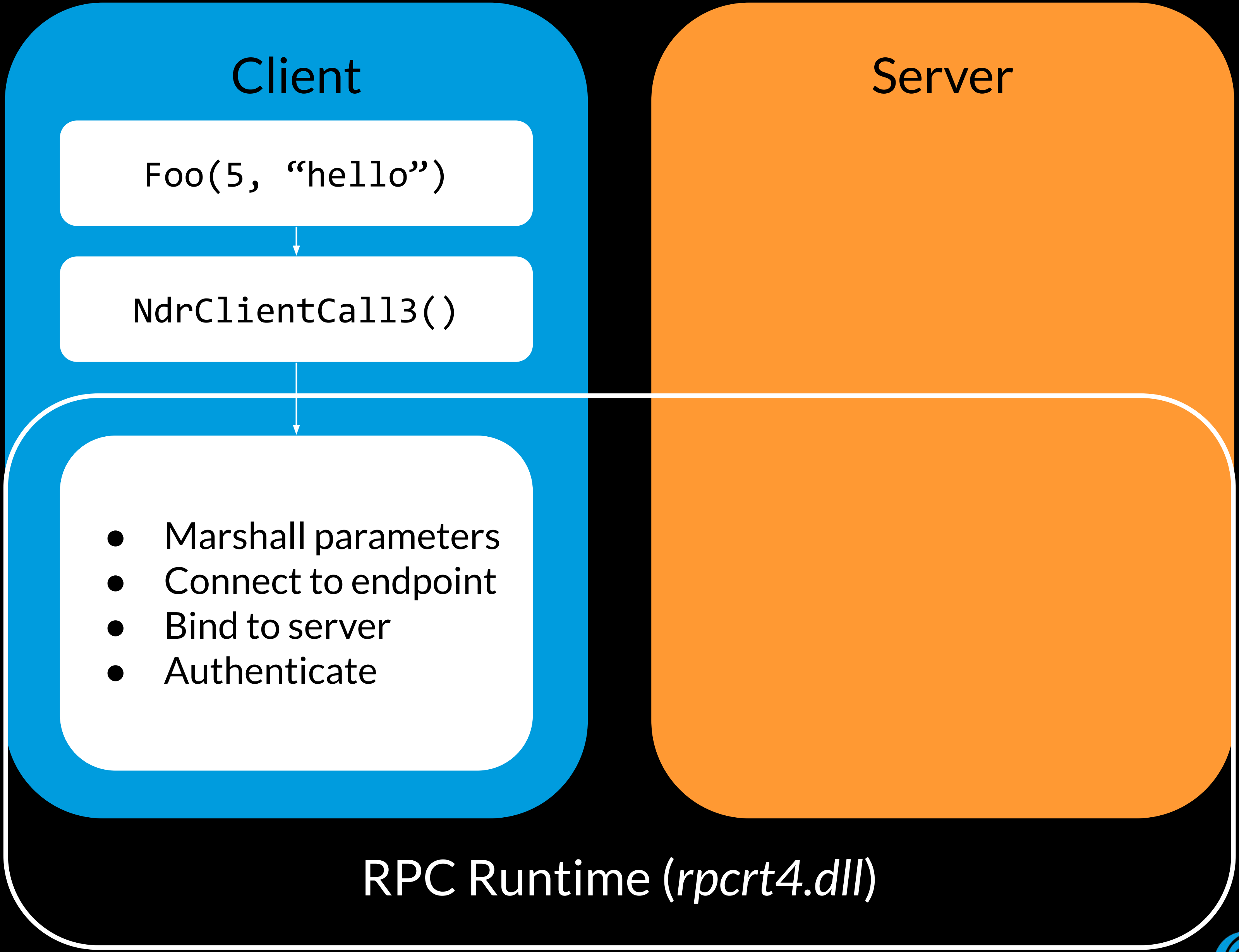
```
Foo(5, “hello”)
```

```
NdrClientCall3()
```

Server

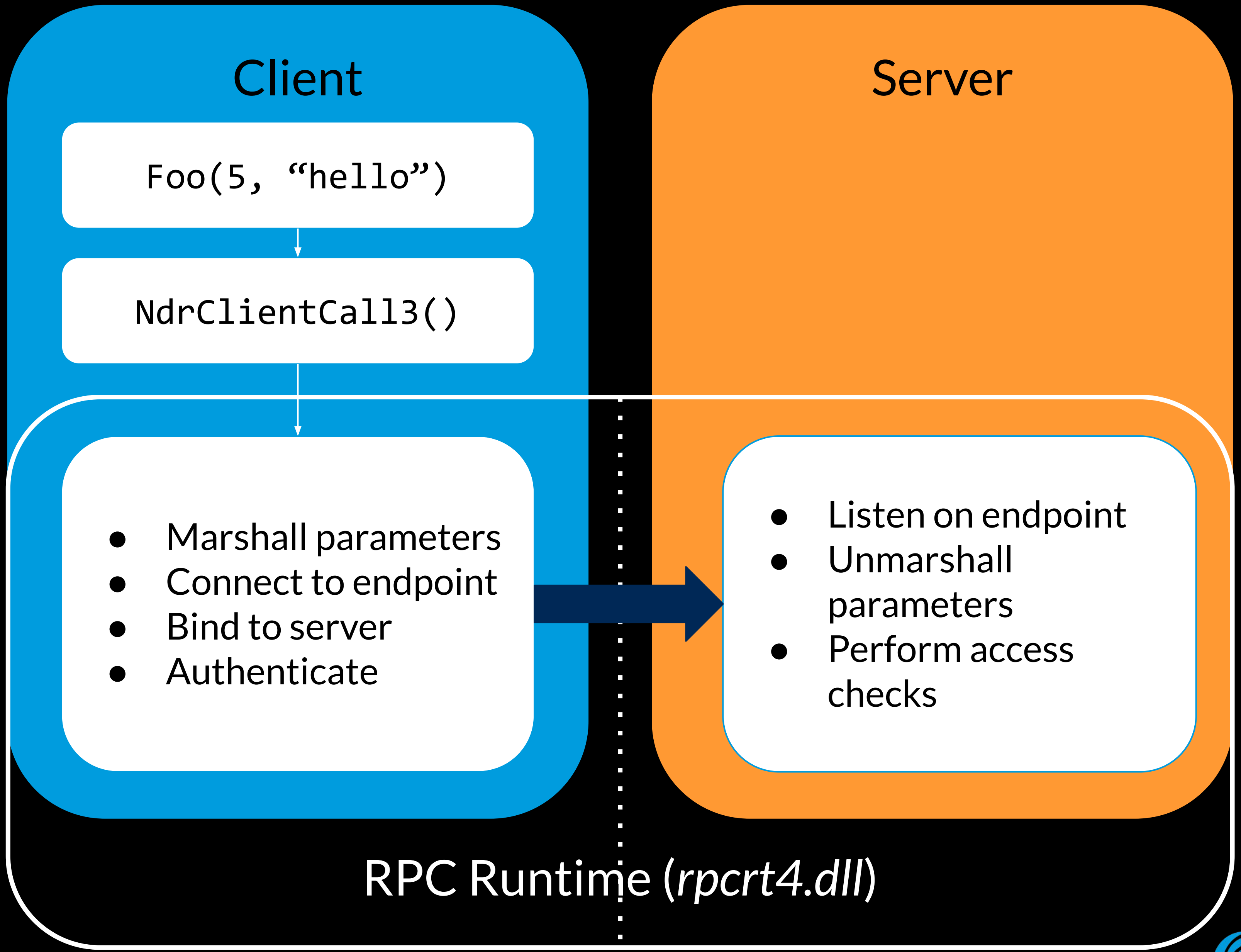
Akamai

An RPC Call's Flow



- Marshall parameters
- Connect to endpoint
- Bind to server
- Authenticate

An RPC Call’s Flow



Client

`Foo(5, “hello”)`

`NdrClientCall3()`

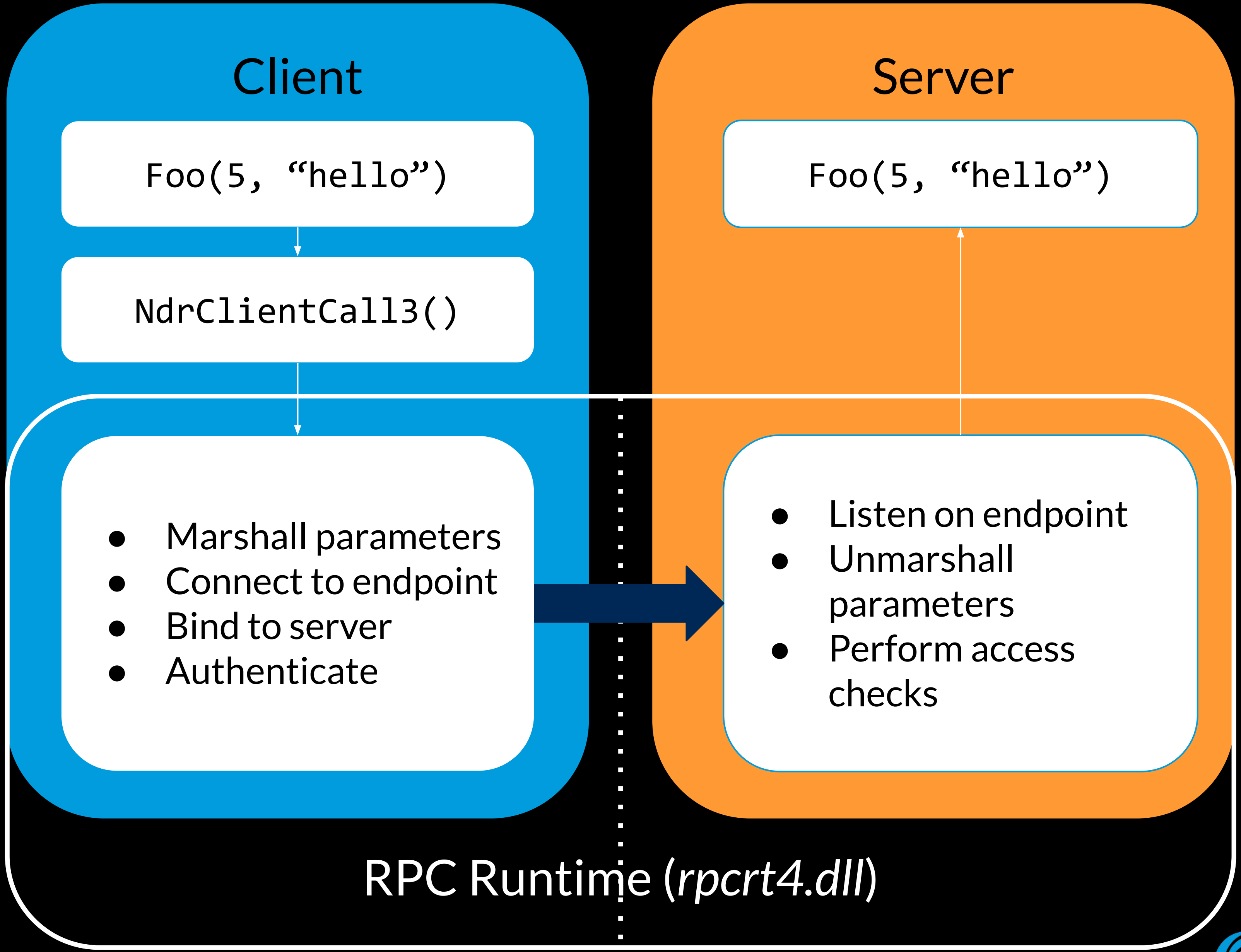
- Marshall parameters
- Connect to endpoint
- Bind to server
- Authenticate

Server

- Listen on endpoint
- Unmarshall parameters
- Perform access checks

RPC Runtime (*rpcrt4.dll*)

An RPC Call's Flow



Client

```
Foo(5, “hello”)
```

```
NdrClientCall3()
```

- Marshall parameters
- Connect to endpoint
- Bind to server
- Authenticate

Server

```
Foo(5, “hello”)
```

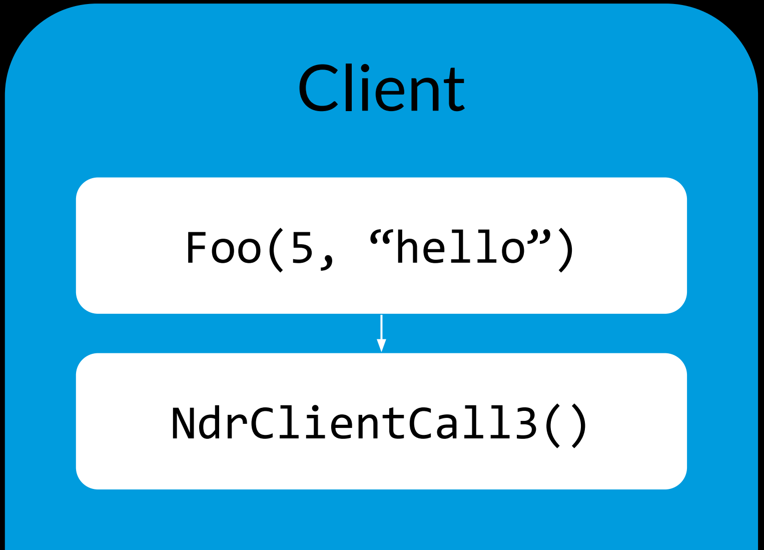
- Listen on endpoint
- Unmarshall parameters
- Perform access checks

RPC Runtime (*rpcrt4.dll*)

Zooming In

IDL:

```
void Foo([in] int number,
         [in] char* message);
```



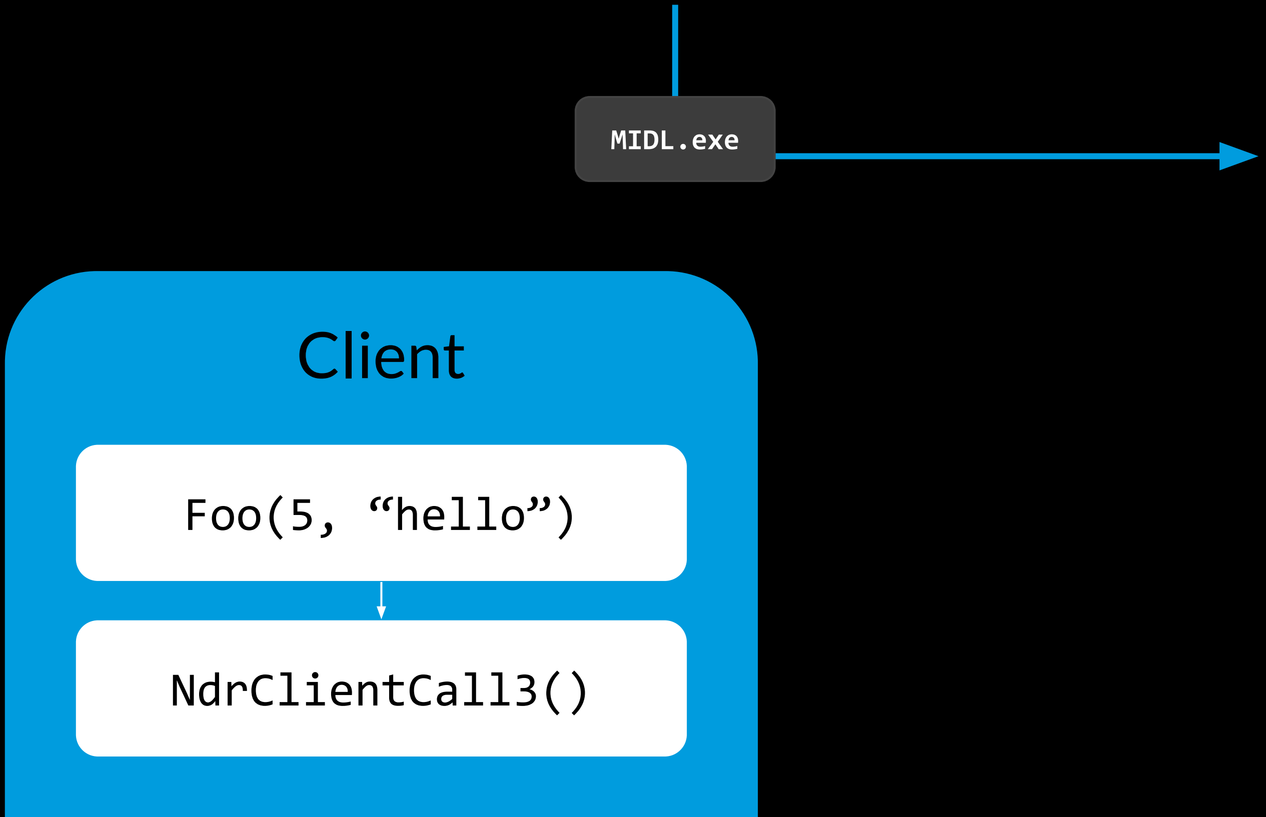
Zooming In

IDL:

```
void Foo([in] int number,
         [in] char* message);
```

Test c.c:

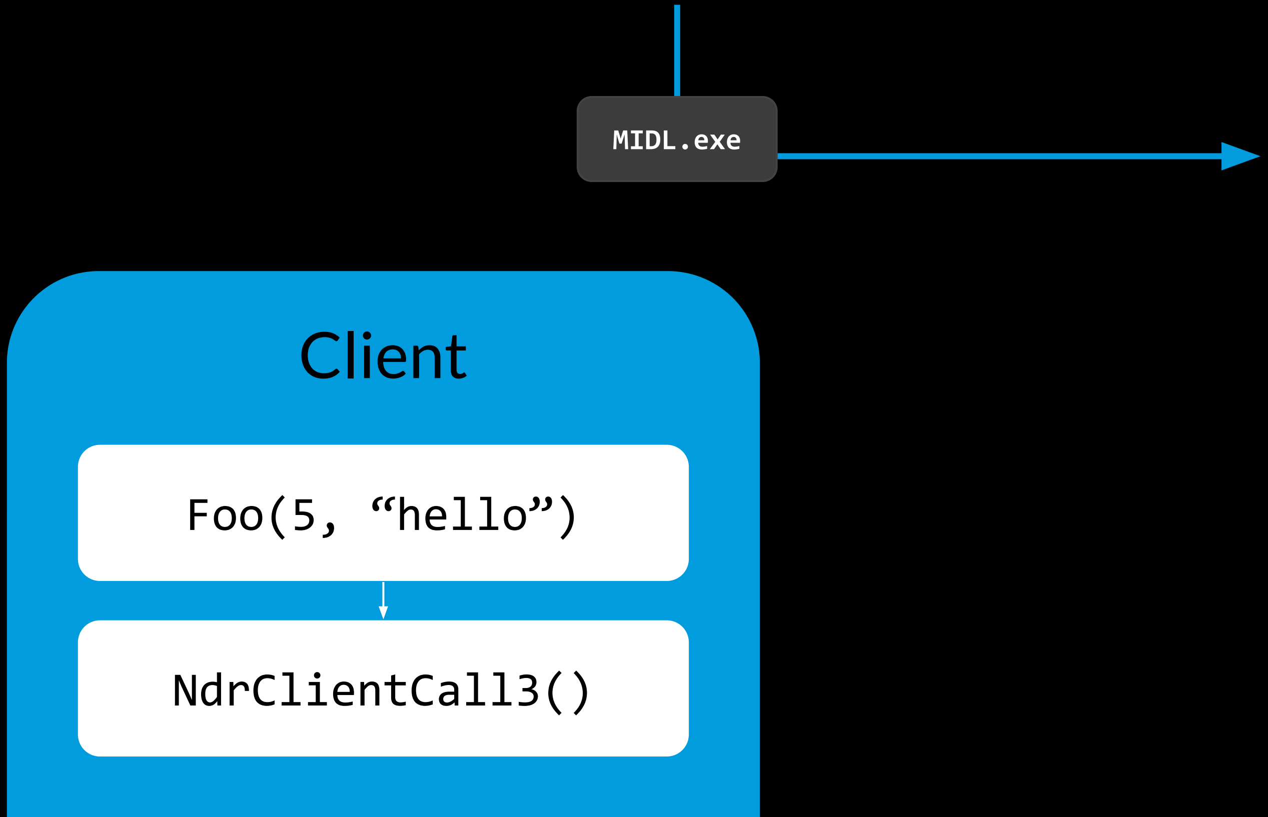
```
void Foo(
  handle_t IDL_handle,
  int number,
  unsigned char *message) {

  NdrClientCall3(
  (PMIDL_STUBLESS_PROXY_INFO
  )&Test_ProxyInfo, 0, 0,
  IDL_handle, number, message);
}
```



Zooming In

IDL:

```
void Foo([in] int number,
         [in] char* message);
```



MIDL.exe

Client

Foo(5, “hello”)

NdrClientCall3()

Test c.c:

```
void Foo(
  handle_t IDL_handle,
  int number,
  unsigned char *message) {

  NdrClientCall3(
  (PMIDL_STUBLESS_PROXY_INFO
  )&Test_ProxyInfo, 0, 0,
  IDL_handle, number, message);
}
```

Opnum

Quick Recap

- Interface – describes server functionality [UUID]
- Transport – the communication medium [protocol sequence]
- Endpoint – destination to connect to [port, pipe name, etc.]
- Binding – represents a client-server session [binding handle]

MS-RPC (In-)Security

Authenticated Binding

- Binding which carries authentication information
    - The server can register an authentication service provider

```
RPC_STATUS RpcServerRegisterAuthInfo(
  RPC_CSTR                  ServerPrincName,
  unsigned long             AuthnSvc,
  RPC_AUTH_KEY_RETRIEVAL_FN GetKeyFn,
  void                      *Arg
);
```

Authenticated Binding

- Binding which carries authentication information
    - The server can register an authentication service provider
    - The client can then authenticate using that provider

```
RPC_STATUS RpcServerRegisterAuthInfo(
  RPC_CSTR                  ServerPrincName,
  unsigned long             AuthnSvc,
  RPC_AUTH_KEY_RETRIEVAL_FN GetKeyFn,
  void                      *Arg
);
```

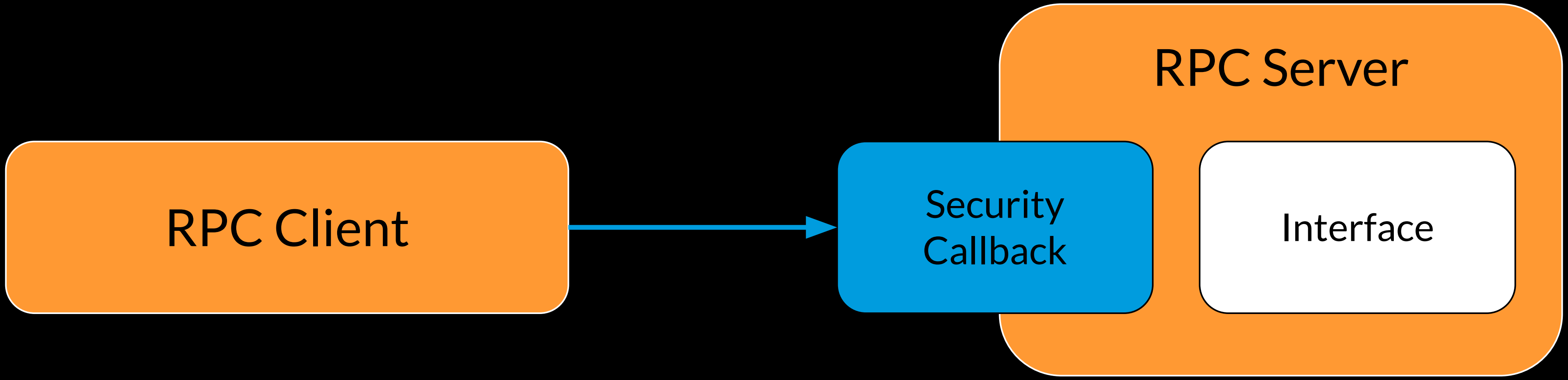
Authenticated Binding

- Binding which carries authentication information
  - The server can register an authentication service provider
  - The client can then authenticate using that provider
- End result: a security context - a “security binding”

```
RPC_STATUS RpcServerRegisterAuthInfo(
  RPC_CSTR                  ServerPrincName,
  unsigned long             AuthnSvc,
  RPC_AUTH_KEY_RETRIEVAL_FN GetKeyFn,
  void                      *Arg
);
```

Security Callback

```
RPC_STATUS RpcIfCallbackFn(
       RPC_IF_HANDLE InterfaceUuid,
       void *Context
)
{...}
```



IAS (Internet Authentication Service)

```
RPC_STATUS CIasRpcServer::RpcIfSecurityCallback(RPC_IF_HANDLE InterfaceUuid, void
*Context) {
  …
  if ( !I_RpcBindingIsClientLocal(0i64, &ClientLocalFlag) && ClientLocalFlag ) {
    if ( !RpcBindingInqAuthClientW(Context, 0i64, 0i64, &AuthnLevel, 0i64, 0i64)
      && AuthnLevel >= RPC_C_AUTHN_LEVEL_PKT_PRIVACY
      && CIasRpcServer::IsCorrectProtseq(&hBinding)
      && CIasRpcServer::IsAccessGranted(v3, &hBinding) )
    {
      return RPC_S_OK;
    }
  }
  return RPC_S_ACCESS_DENIED;
}
```

IAS (Internet Authentication Service)

```
RPC_STATUS CIasRpcServer::RpcIfSecurityCallback(RPC_IF_HANDLE InterfaceUuid, void
*Context) {
  …
  if ( !I_RpcBindingIsClientLocal(0i64, &ClientLocalFlag) && ClientLocalFlag ) {
    if ( !RpcBindingInqAuthClientW(Context, 0i64, 0i64, &AuthnLevel, 0i64, 0i64)
      && AuthnLevel >= RPC_C_AUTHN_LEVEL_PKT_PRIVACY
      && CIasRpcServer::IsCorrectProtseq(&hBinding)
      && CIasRpcServer::IsAccessGranted(v3, &hBinding) )
    {
      return RPC_S_OK;
    }
  }
  return RPC_S_ACCESS_DENIED;
}
```

IAS (Internet Authentication Service)

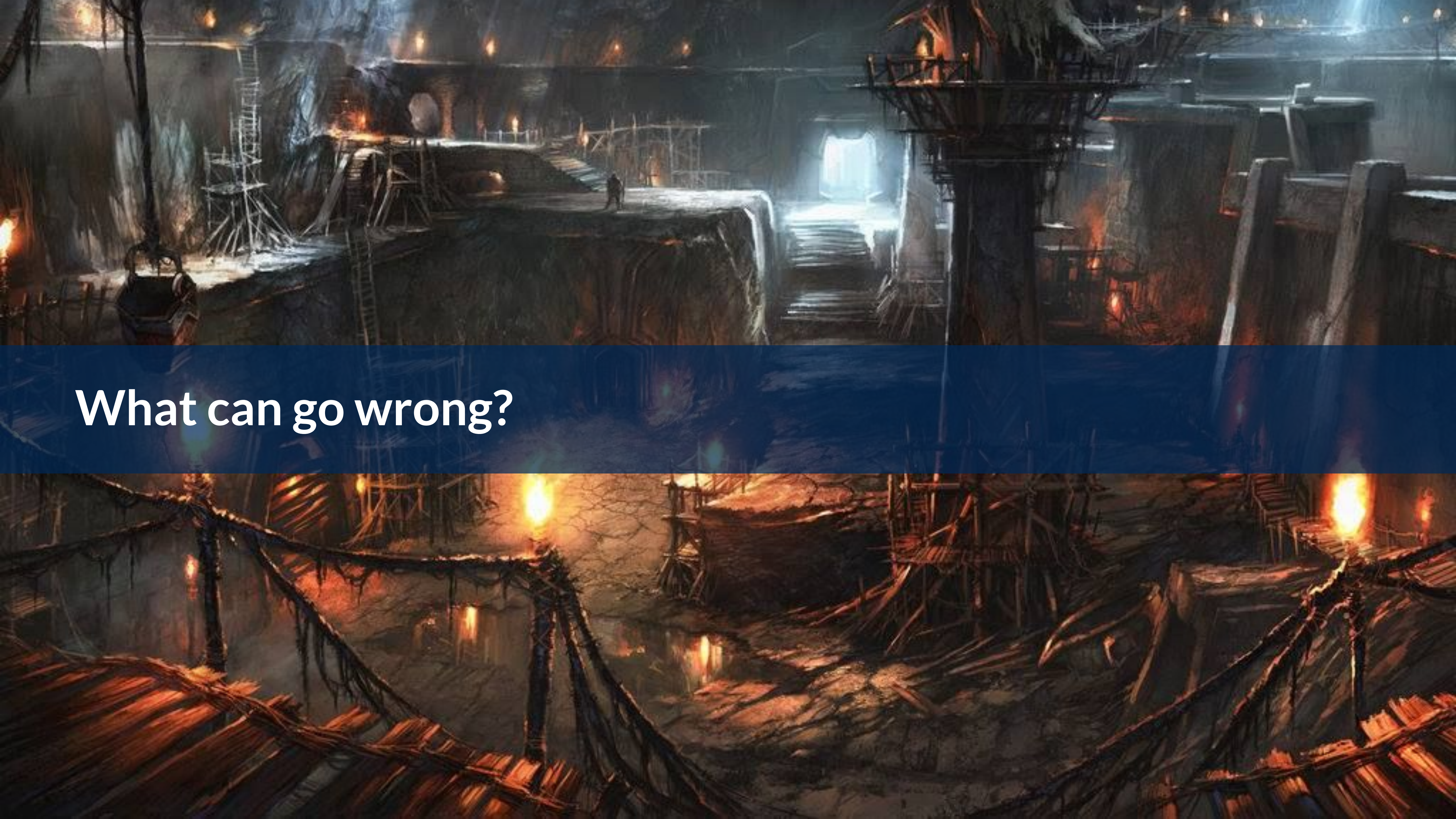
```
RPC_STATUS CIasRpcServer::RpcIfSecurityCallback(RPC_IF_HANDLE InterfaceUuid, void
*Context) {
  …
  if ( !I_RpcBindingIsClientLocal(0i64, &ClientLocalFlag) && ClientLocalFlag ) {
    if ( !RpcBindingInqAuthClientW(Context, 0i64, 0i64, &AuthnLevel, 0i64, 0i64)
      && AuthnLevel >= RPC_C_AUTHN_LEVEL_PKT_PRIVACY
      && CIasRpcServer::IsCorrectProtseq(&hBinding)
      && CIasRpcServer::IsAccessGranted(v3, &hBinding) )
    {
      return RPC_S_OK;
    }
  }
  return RPC_S_ACCESS_DENIED;
}
```

LSASS

```
RPC_STATUS LsaRpcIfCallbackFn(RPC_IF_HANDLE InterfaceUuid, void *Context)) {
  …
  LastError = RpcServerInqCallAttributesW(a2, &RpcCallAttributes);
  …
  if ( RpcCallAttributes.OpNum >= 0x86u ) return RPC_S_PROCNUM_OUT_OF_RANGE;
  …
  v6 = *((_DWORD *)&LsapRPCFunctionProperties + 2 * RpcCallAttributes.OpNum);
  if ( !_bittest(&v6, RpcCallAttributes.ProtocolSequence) )
    return RPC_S_PROTSEQ_NOT_SUPPORTED;
  …
}
```

LSASS

```
RPC_STATUS LsaRpcIfCallbackFn(RPC_IF_HANDLE InterfaceUuid, void *Context)) {
  …
  LastError = RpcServerInqCallAttributesW(a2, &RpcCallAttributes);
  …
  if ( RpcCallAttributes.OpNum >= 0x86u ) return RPC_S_PROCNUM_OUT_OF_RANGE;
  …
  v6 = *((_DWORD *)&LsapRPCFunctionProperties + 2 * RpcCallAttributes.OpNum);
  if ( !_bittest(&v6, RpcCallAttributes.ProtocolSequence) )
    return RPC_S_PROTSEQ_NOT_SUPPORTED;
  …
}
```



What can go wrong?

Security Callback Caching

Security Callback Caching

- Security callback results are cached by default

Security Callback Caching

- Security callback results are cached by default
- Cache is per security context

Security Callback Caching

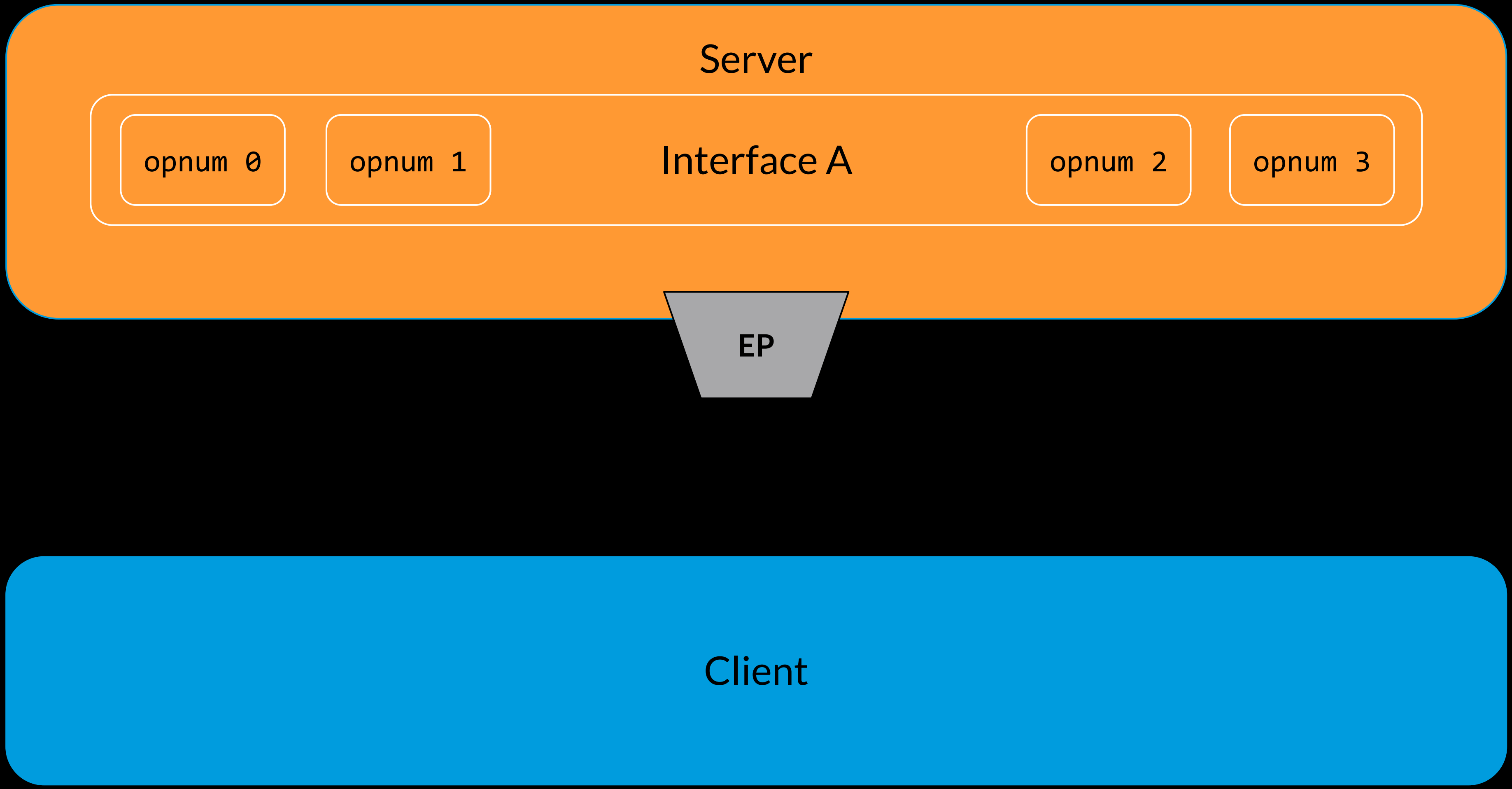
- Security callback results are cached by default
- Cache is per security context
    - No authentication? No cache

Security Callback Caching

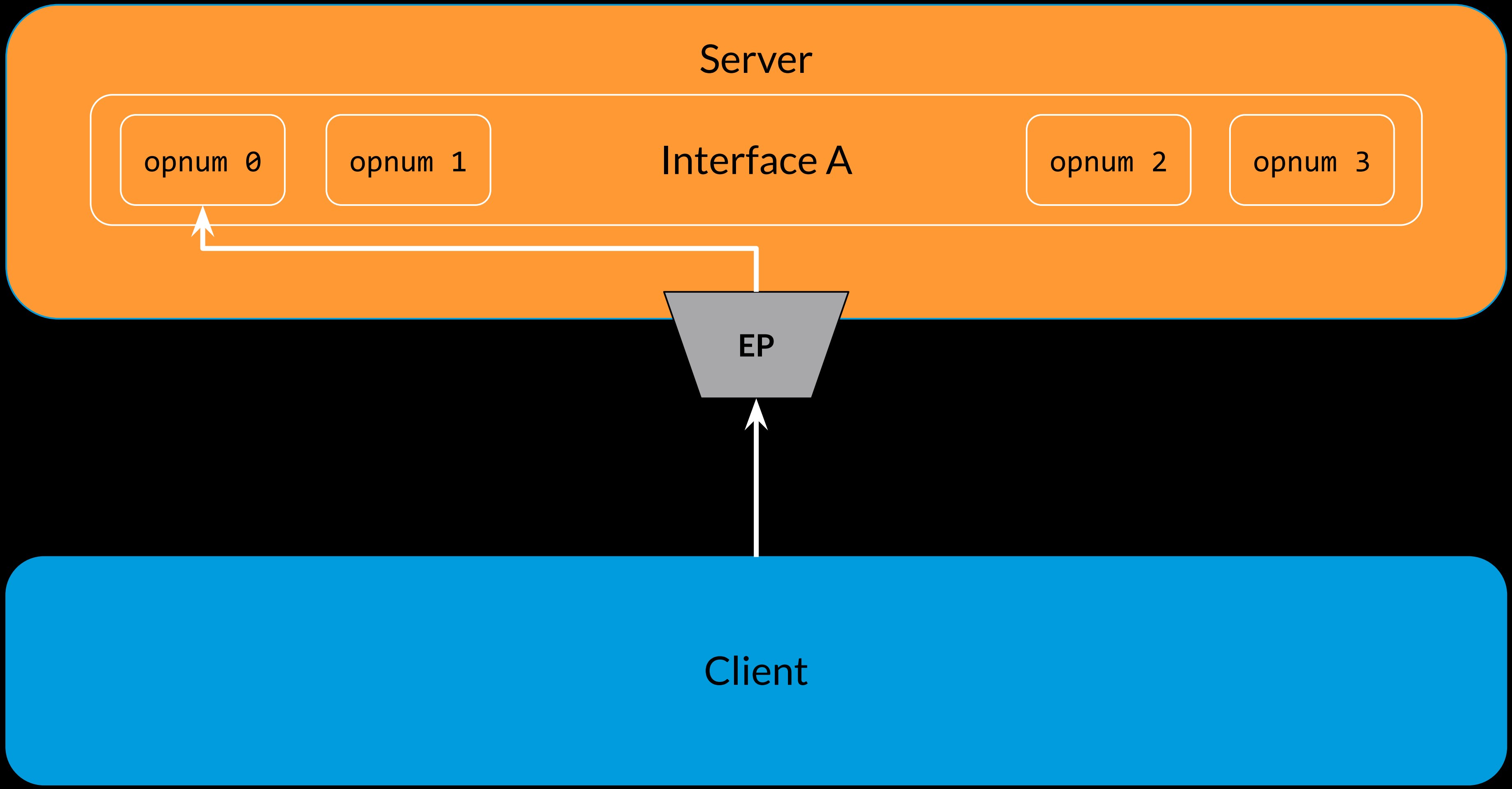
- Security callback results are cached by default
- Cache is per security context
  - No authentication? No cache

```
#define RPC_IF_SEC_NO_CACHE        0x40
#define RPC_IF_SEC_CACHE_PER_PROC  0x80
```

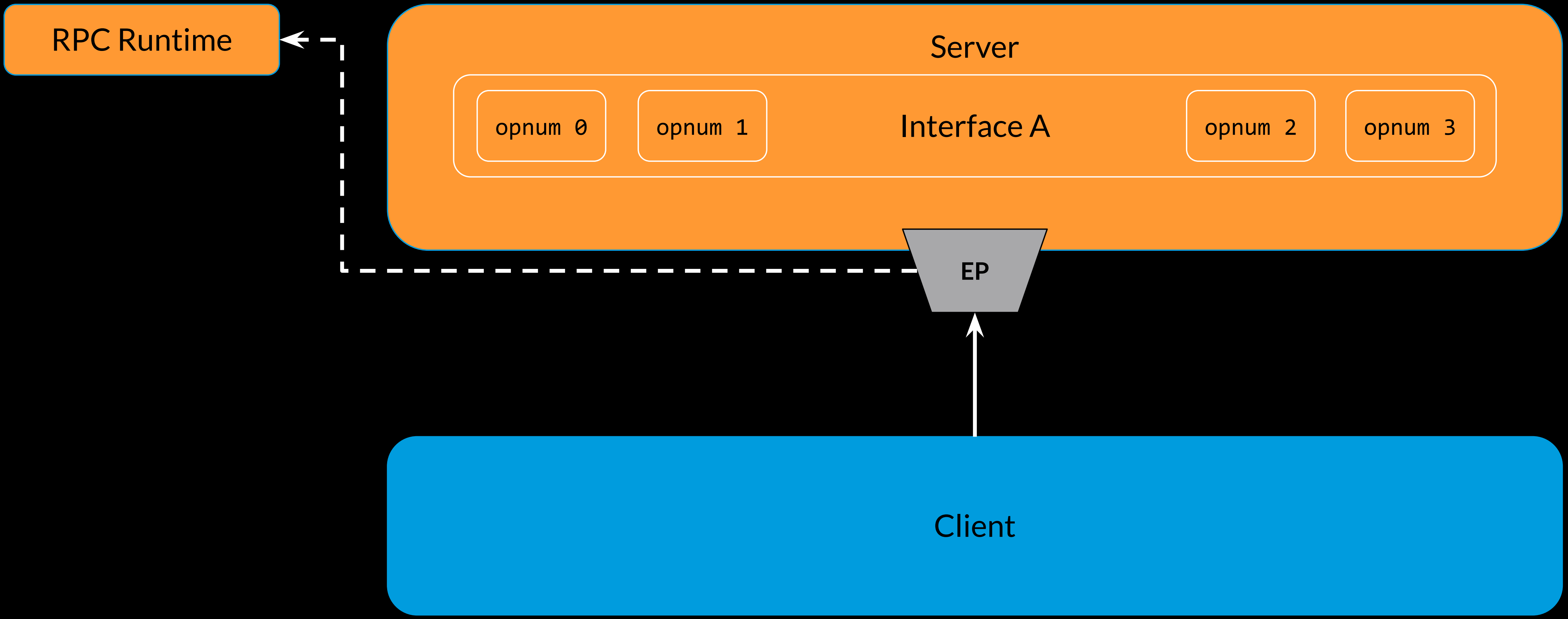
Security Callback Caching



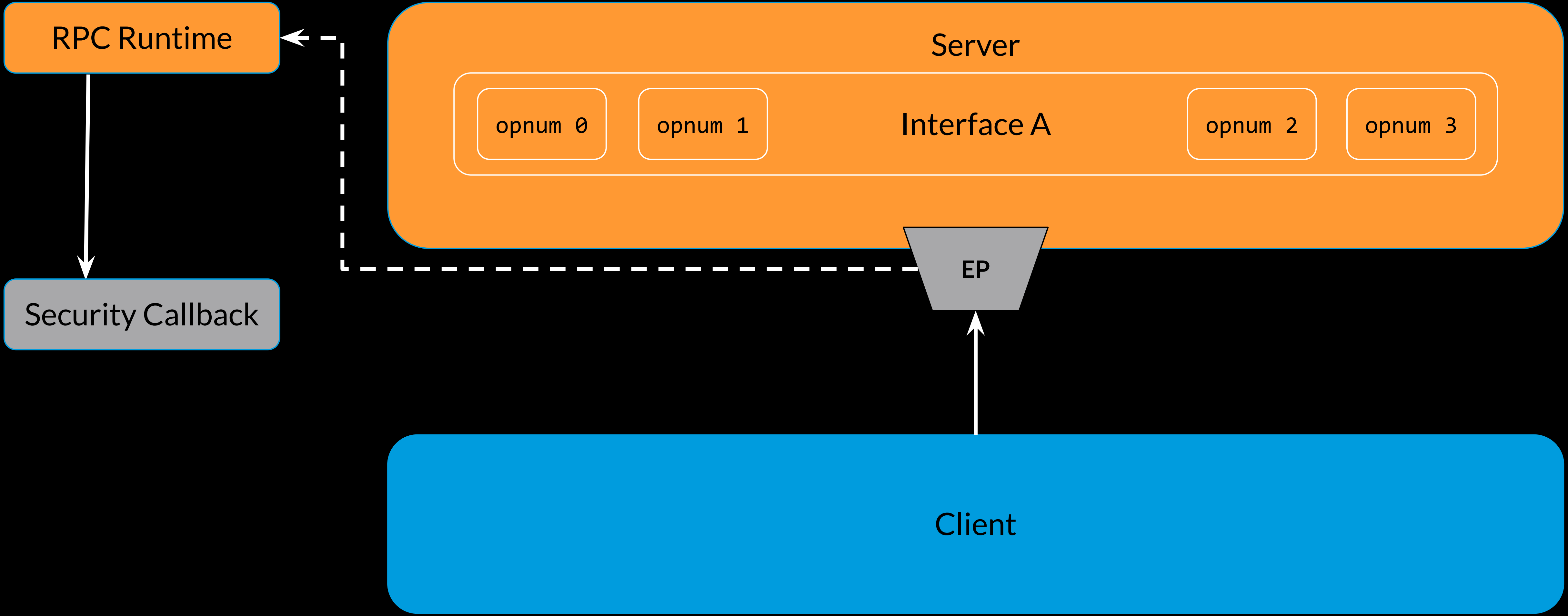
Security Callback Caching



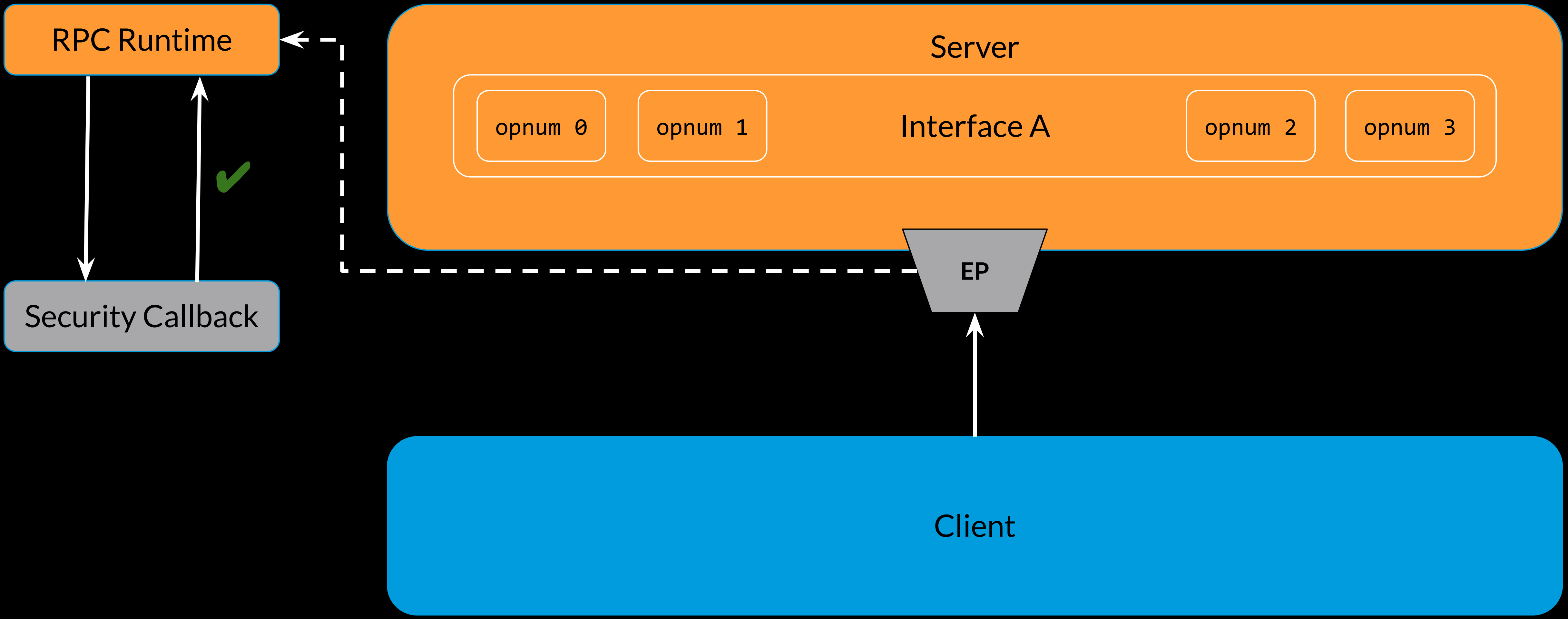
Security Callback Caching



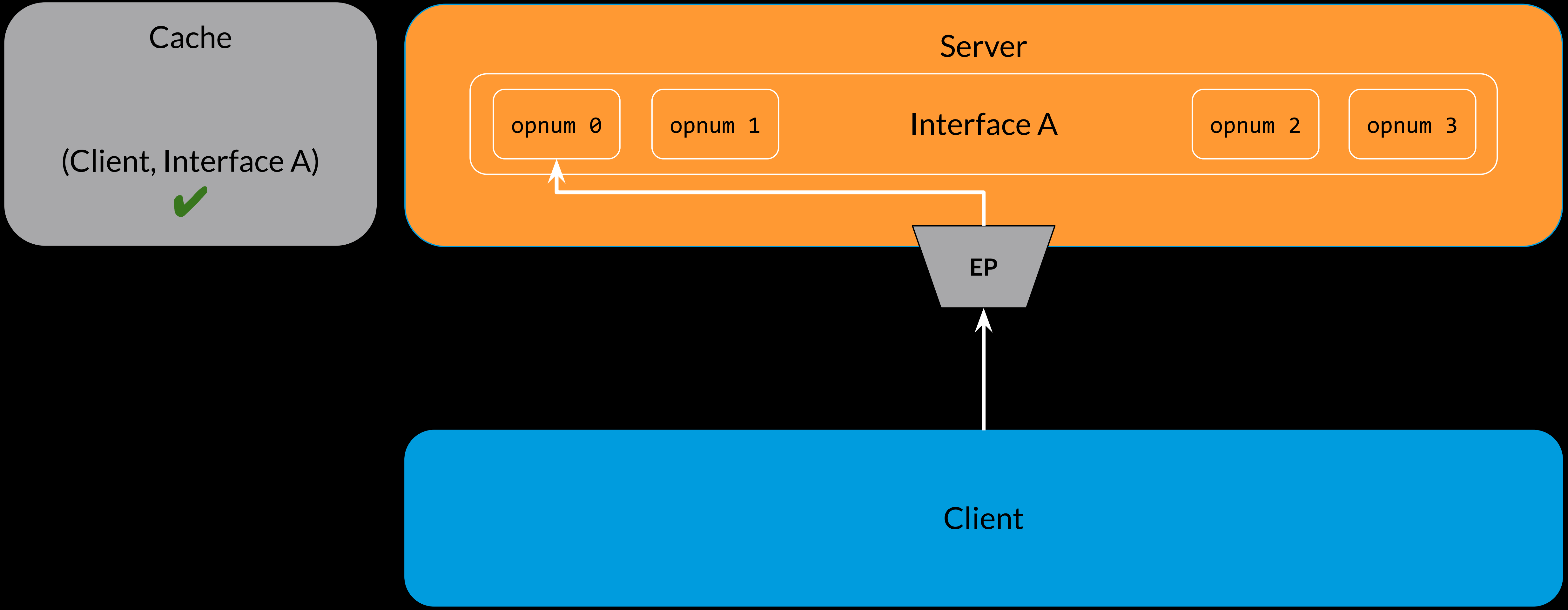
Security Callback Caching



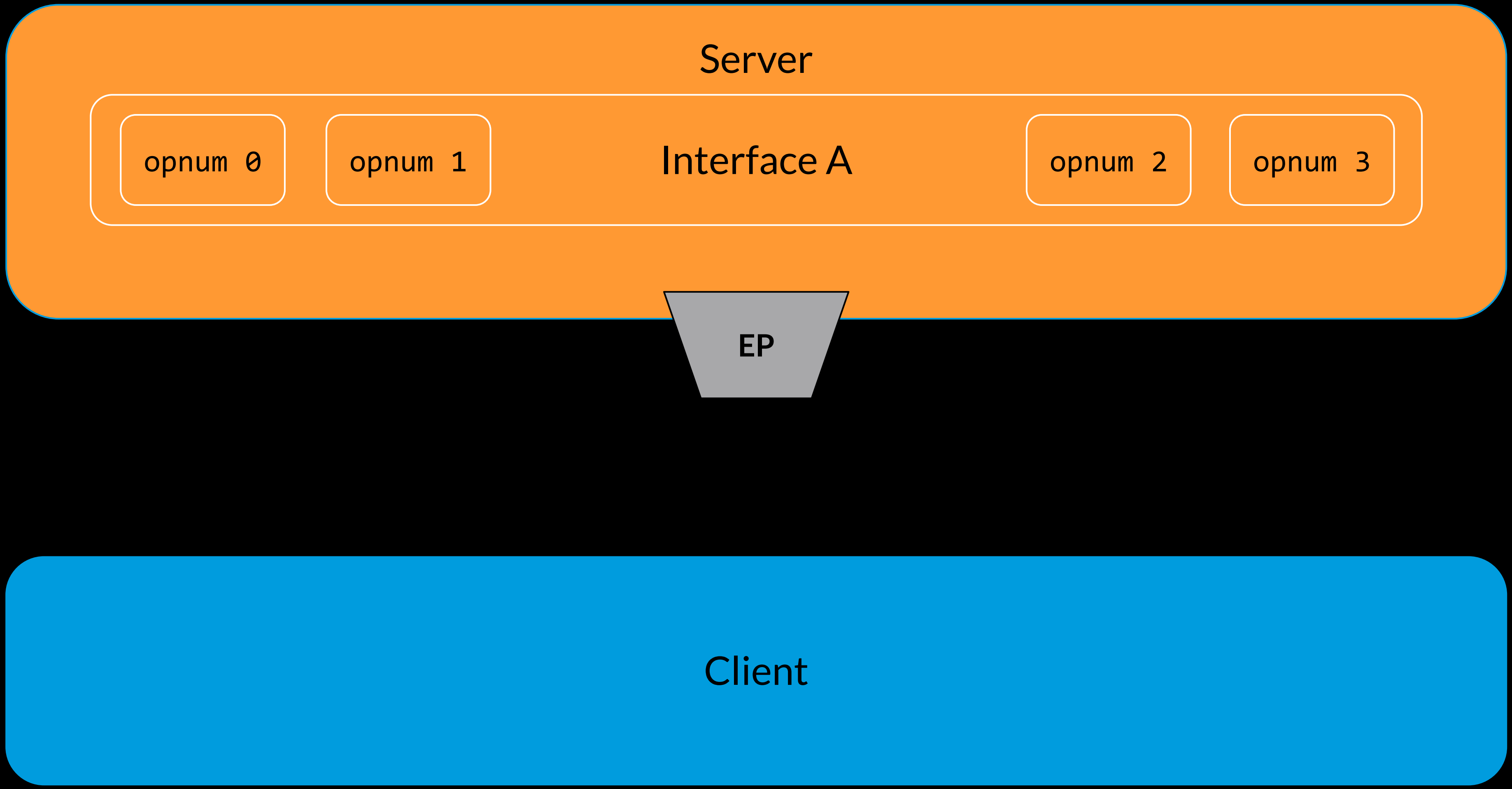
Security Callback Caching



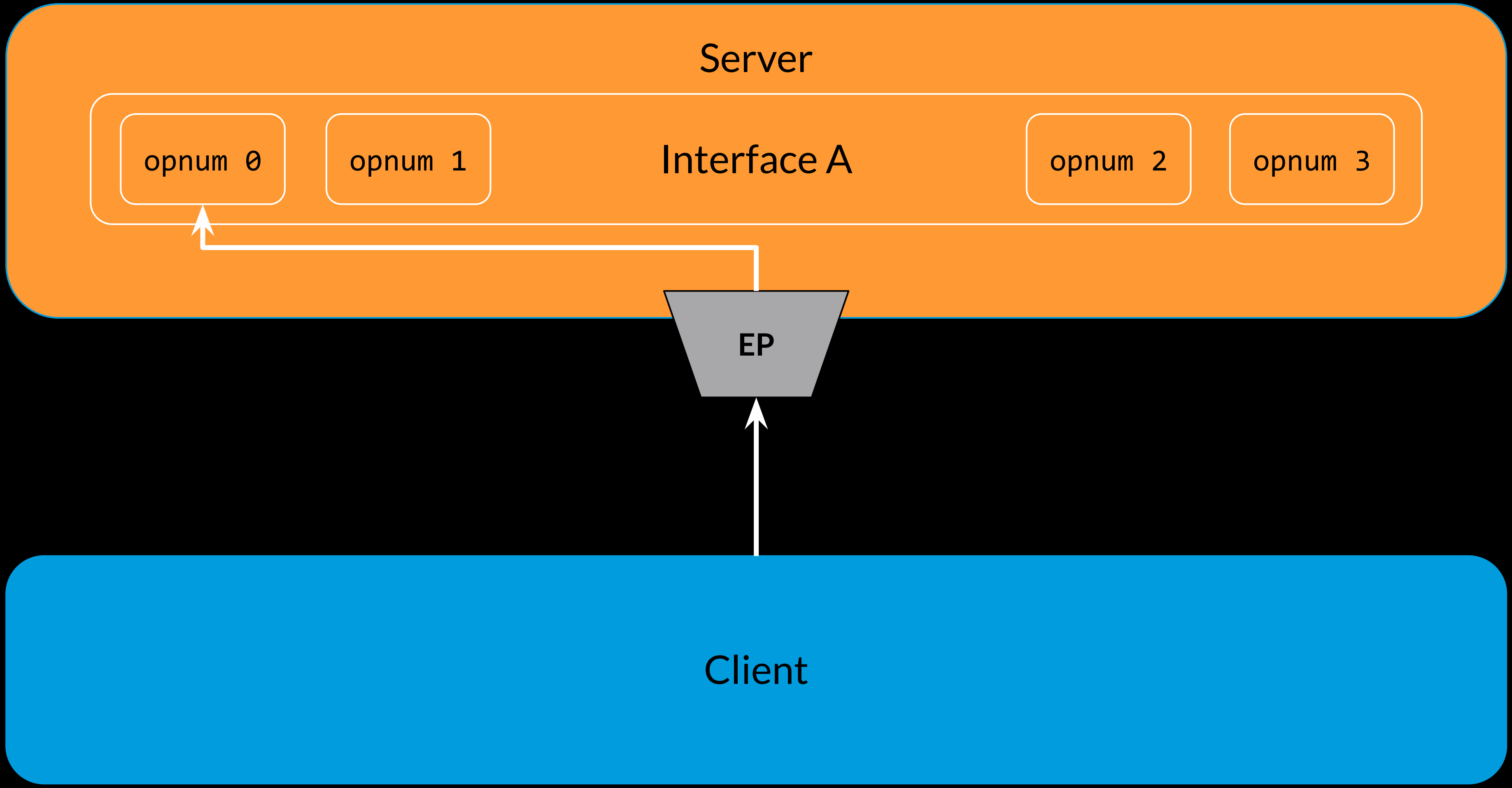
Security Callback Caching



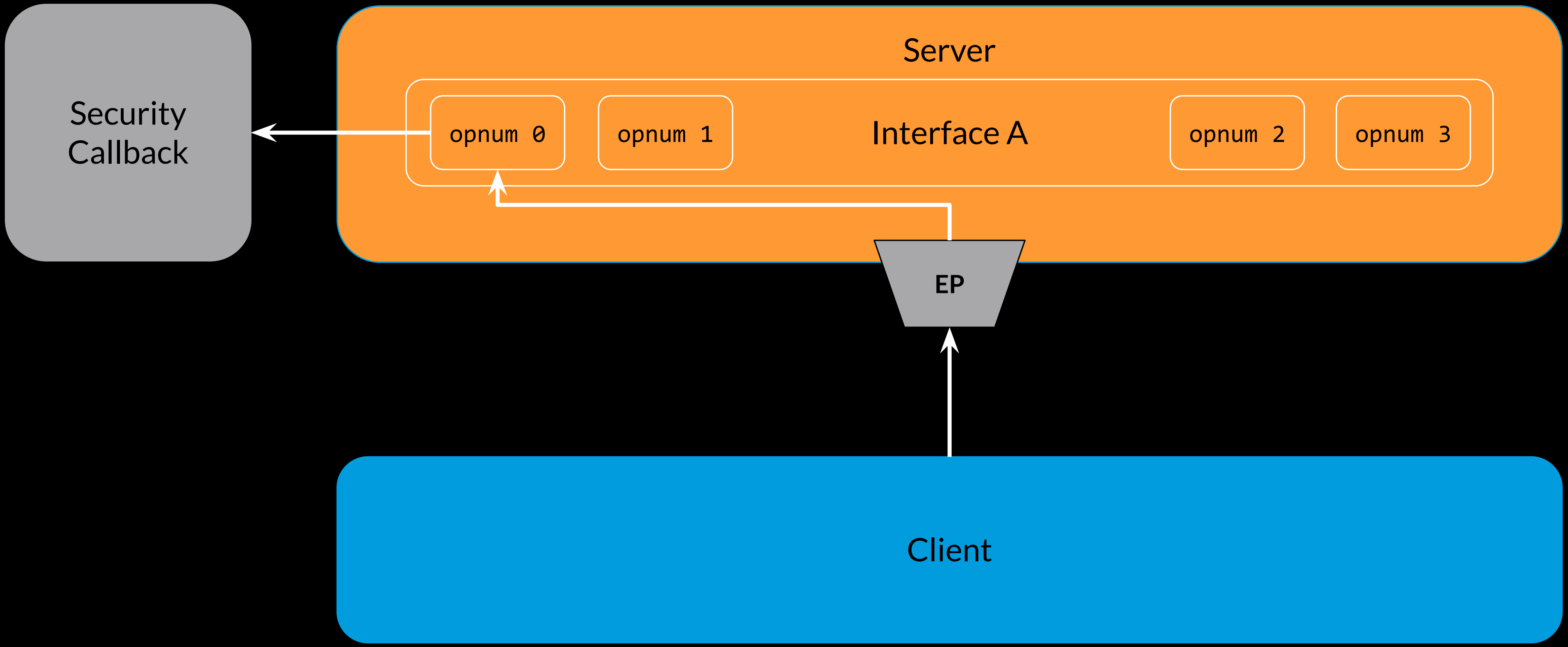
Security Callback Caching Bypass



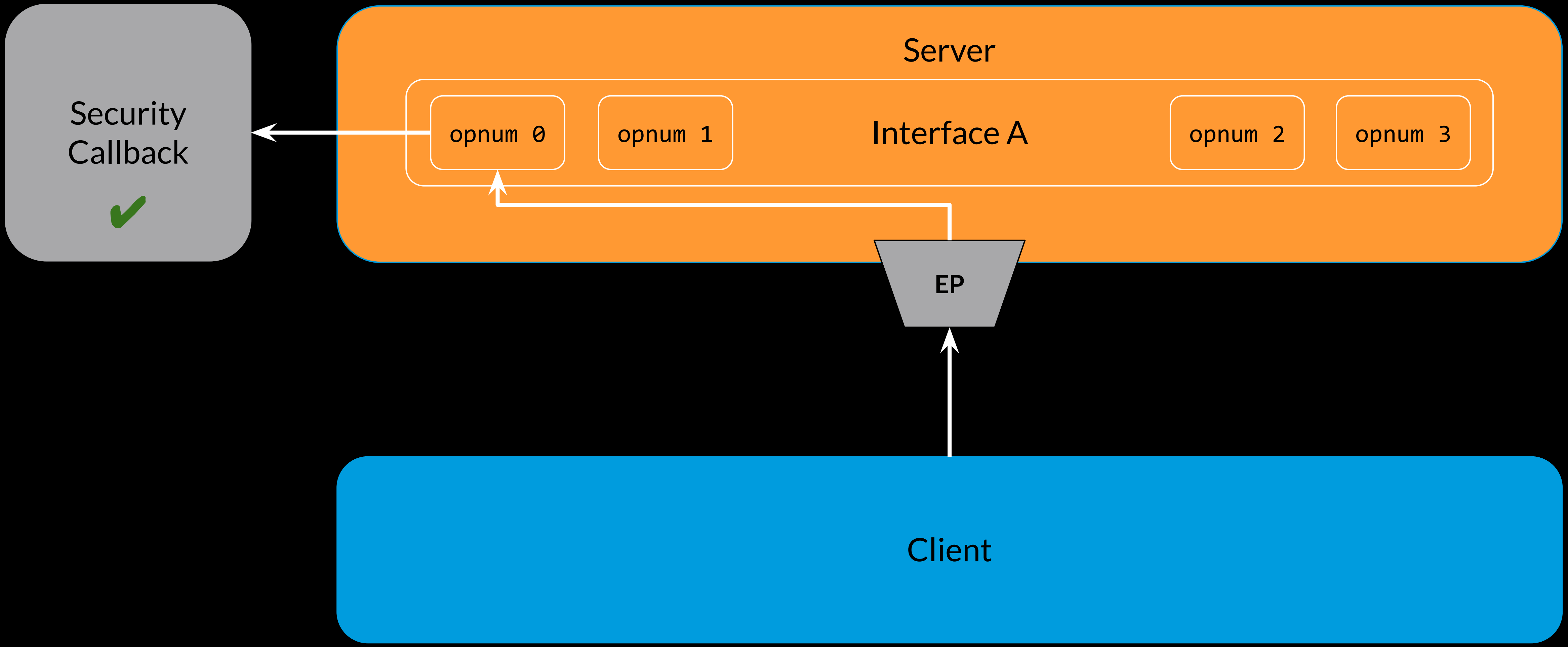
Security Callback Caching Bypass



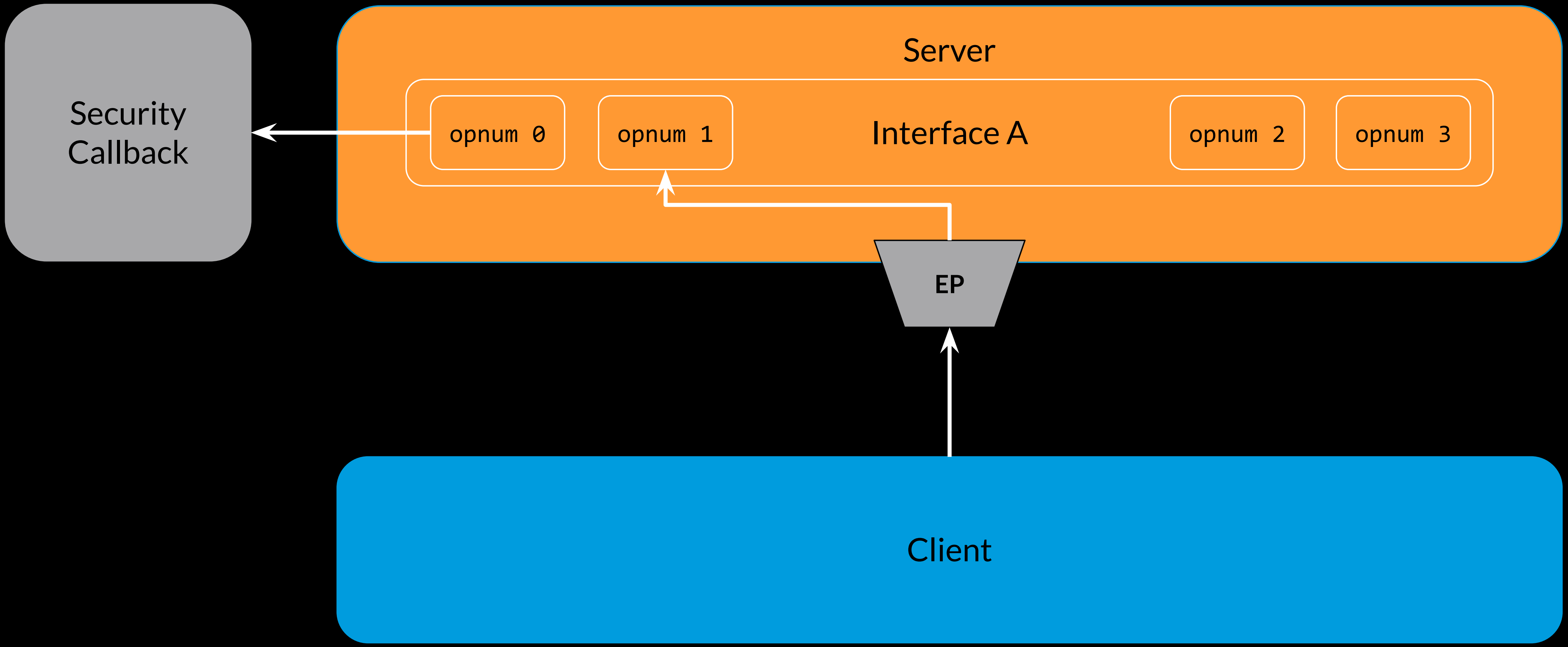
Security Callback Caching Bypass



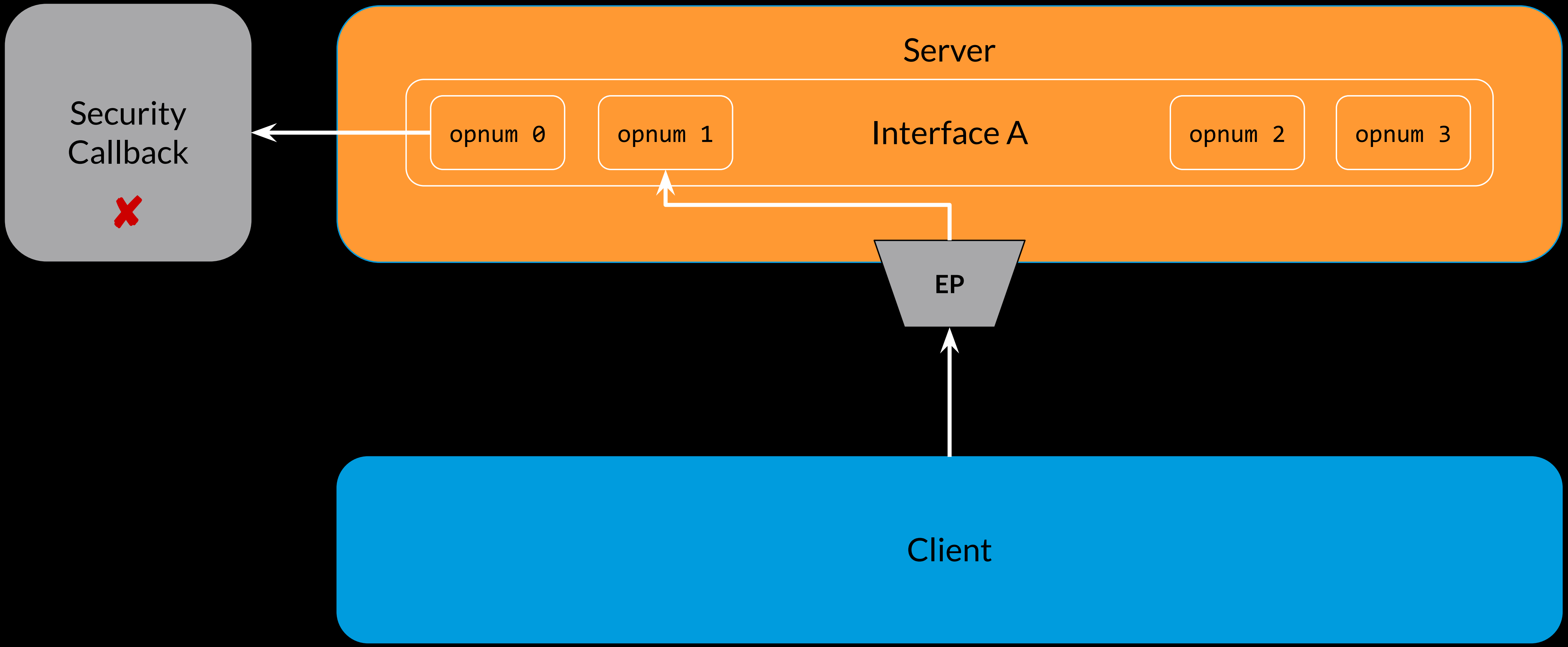
Security Callback Caching Bypass



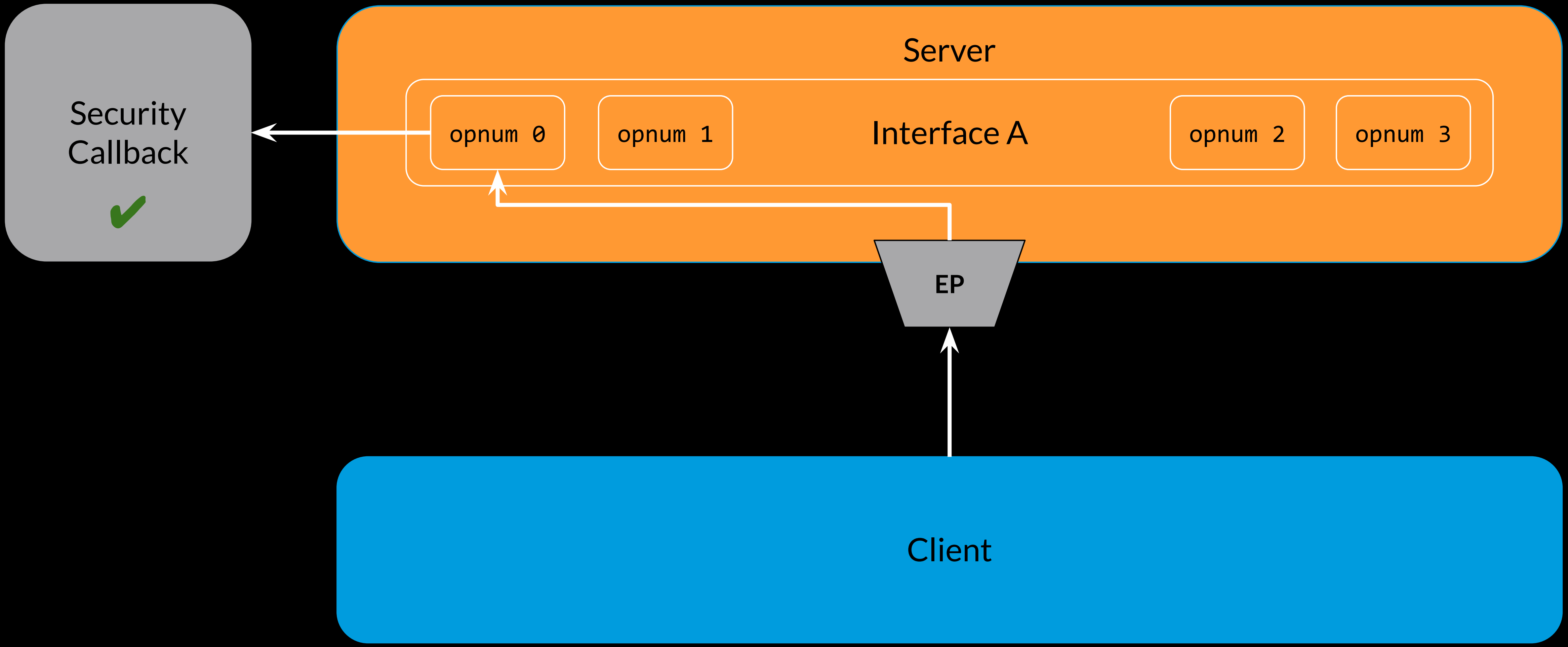
Security Callback Caching Bypass



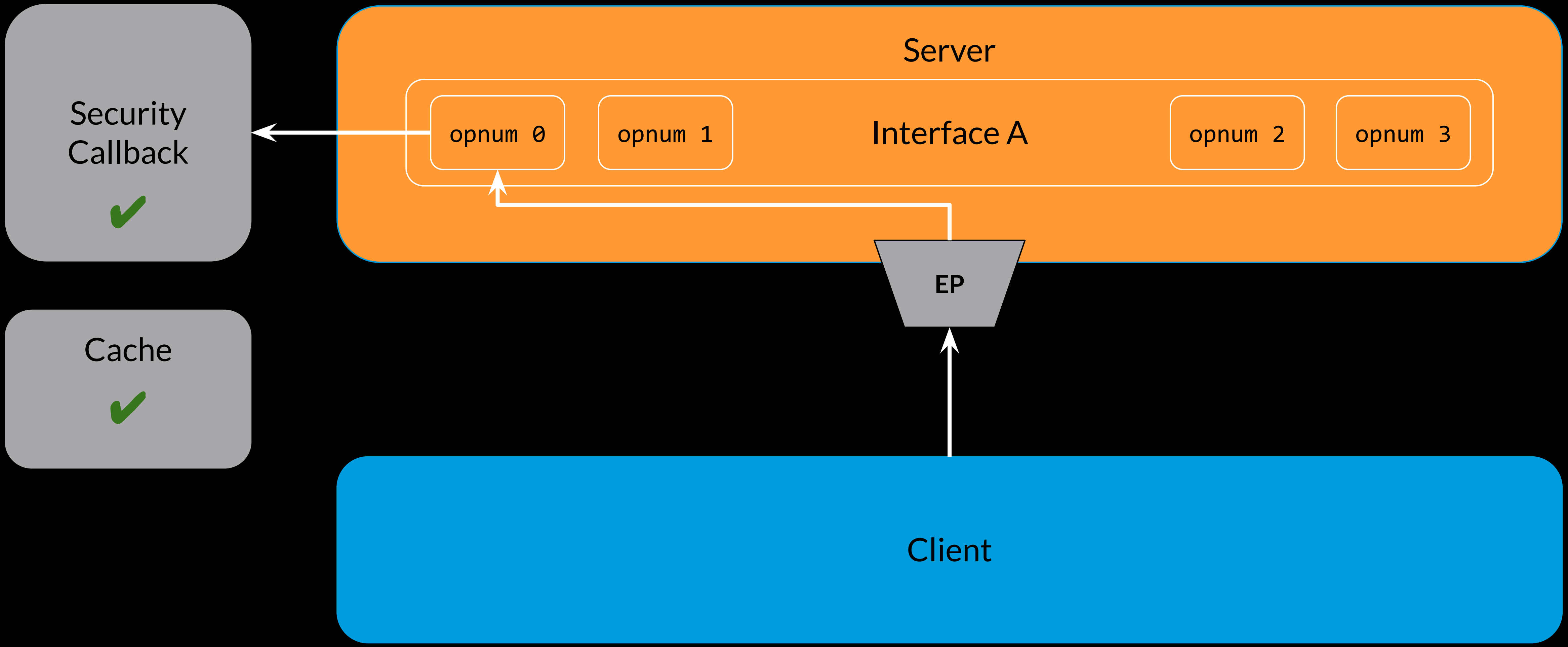
Security Callback Caching Bypass



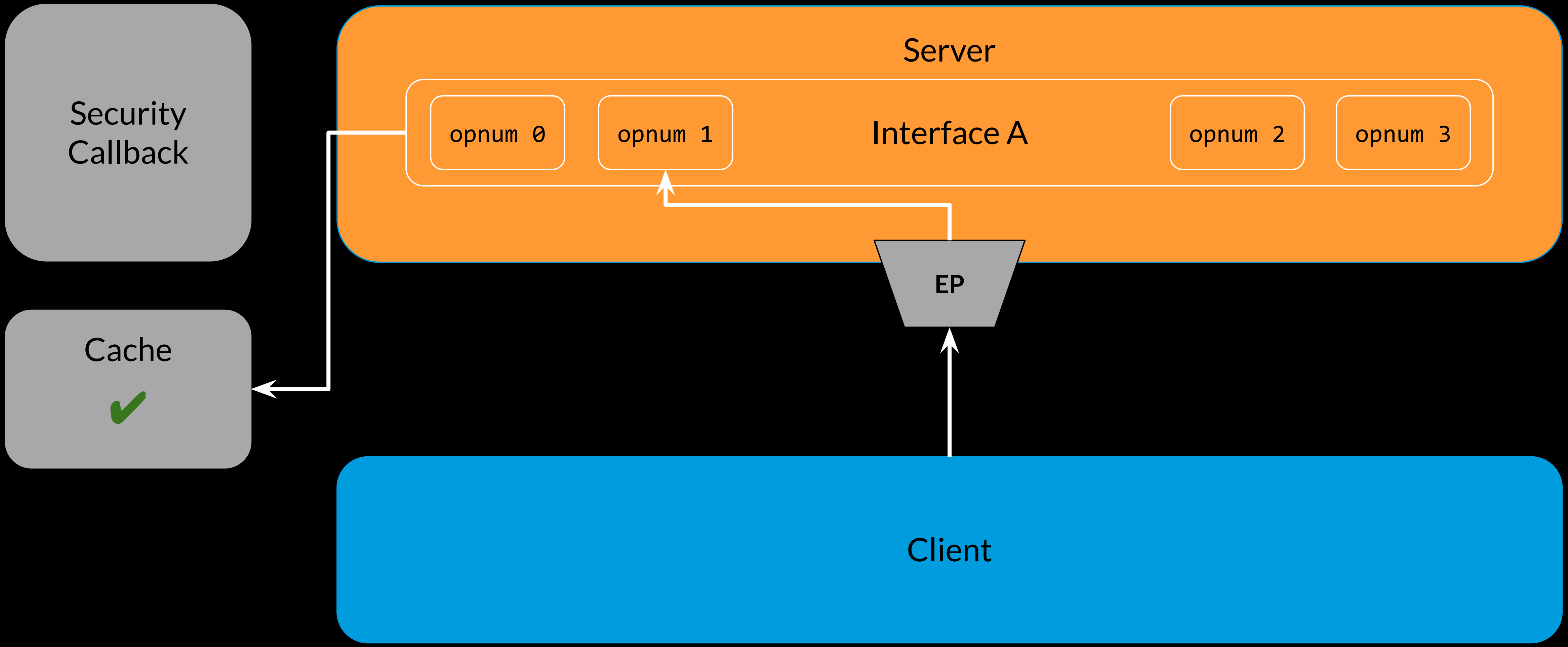
Security Callback Caching Bypass



Security Callback Caching Bypass



Security Callback Caching Bypass



Akamai

MS-RPC (in)Security — Recap

- RPC connections are unauthenticated by default

MS-RPC (in)Security — Recap

- RPC connections are unauthenticated by default
  - RPC servers have to register with a provider

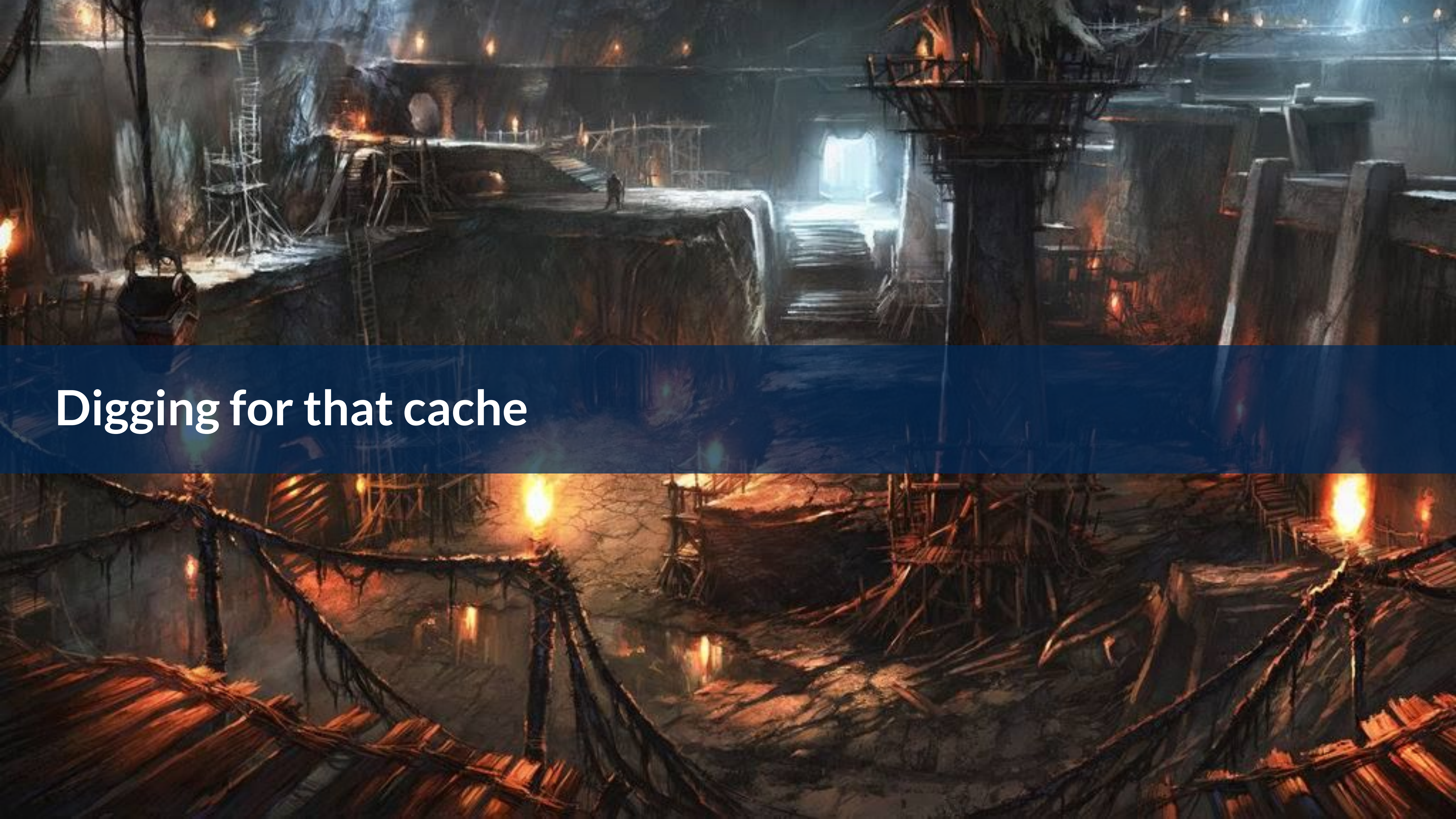
MS-RPC (in)Security — Recap

- RPC connections are unauthenticated by default
  - RPC servers have to register with a provider
- A security callback is a custom access check function

MS-RPC (in)Security — Recap

- RPC connections are unauthenticated by default
  - RPC servers have to register with a provider
- A security callback is a custom access check function
  - It is cached by default
  - Caching can lead to a bypass attack

Digging for that cache



Scraping Windows OS for RPC Interfaces

What’s interesting?

1. What interfaces and functions are exposed
2. How they’re registered

What interfaces and functions are exposed?

```
struct RPC_IF_HANDLE {
    UINT                  Length;
    RPC_SYNTAX_IDENTIFIER InterfaceId;
    RPC_SYNTAX_IDENTIFIER TransferSyntax;
    PRPC_DISPATCH_TABLE   DispatchTable;
    UINT                  RpcProtseqEndpointCount;
    PRPC_PROTSEQ_ENDPOINT RpcProtseqEndpoint;
    RPC_MGR_EPV_PTR_T     DefaultManagerEpv;
    void const PTR_T      InterpreterInfo;
    UINT                  Flags;
}
```

**** Defined in rpcdcep.h***

What interfaces and functions are exposed?

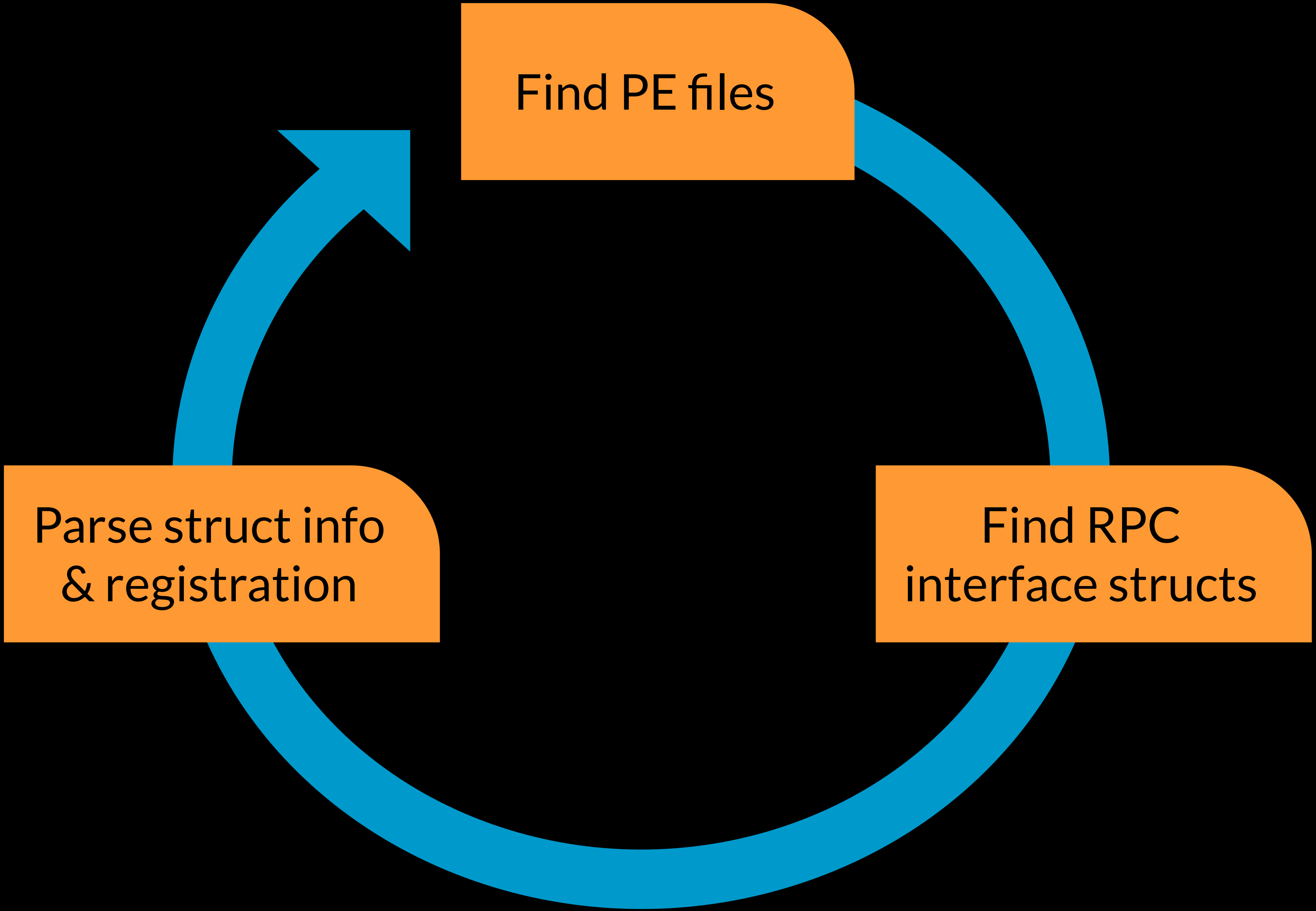
```
off_14006F1E8   dq offset  AddImage            ; DATA XREF: .rdata:000000014006F998↓o
                dq offset  IsImageMounted
                dq offset  RemoveImage
```

** interface 6d9fe472-30f1-4708-8fa8-678362b96155* in *wimserv.exe*

How the interfaces are registered

```
RpcServerRegisterIfEx(
  &<interface_addr>,
  0,
  0,
  <flags>,
  0,
  <security_callback>
);
```

Scraping Windows OS for RPC Interfaces



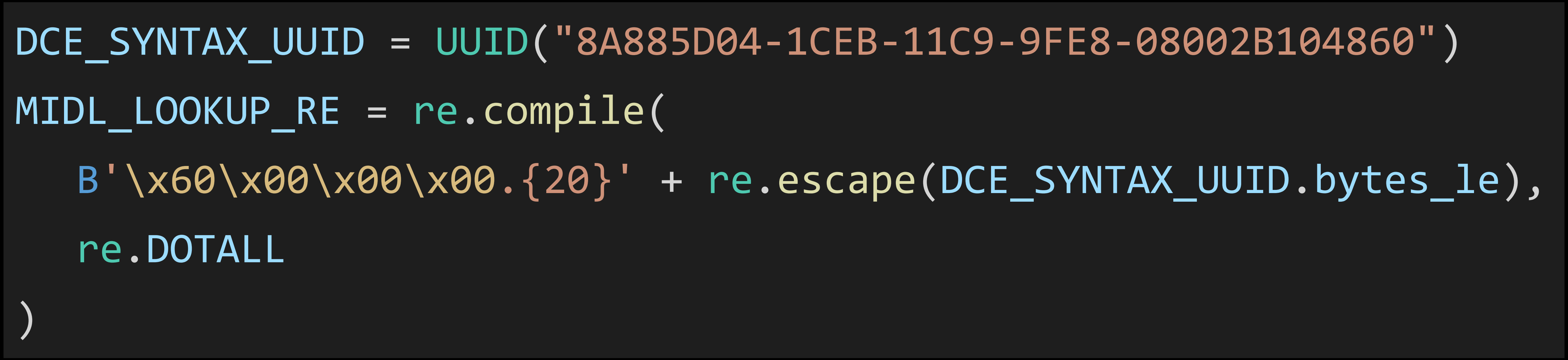
Find PE files

Find RPC interface structs

Parse struct info & registration

```
dword_18002F280   dd    60h                    ; struct size
                  dd    6BFFD098h              ; server interface UUID
                  dw    0A112h
                  dw    3610h
                  dq    5A347EF8C3463398h
                  dw    1                      ; server interface version major
                  dw    0                      ; server interface version minor
                  dd    8A885D04h              ; transfer syntax UUID
                  dw    1CEBh
                  dw    11C9h
                  dq    6048102B0008E89Fh
                  dw    2                      ; transfer syntax version major
                  dw    0                      ; transfer syntax version minor
                  dd    0                      ; alignment
                  dq    offset unk_180030320   ; dispatch table
                  dd    0                      ; endpoint count
                  dd    0                      ; alignment
                  dq    0                      ; endpoint array
                  dq    0                      ; default endpoint management
                  dq    offset off_180030870   ; interpreter info
                  dq    6000000h               ; flags
```

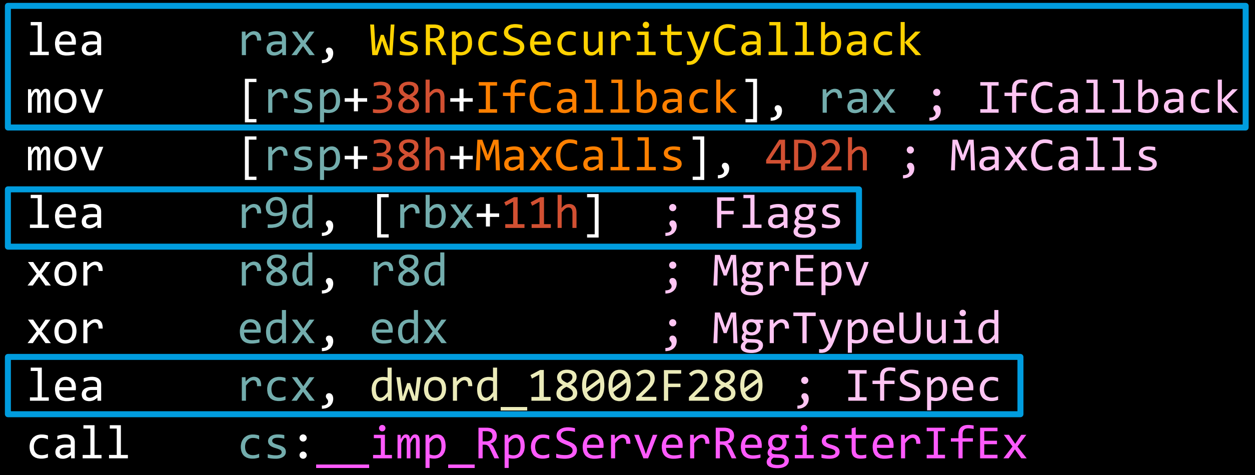
RPC Interface Lookup



```python
DCE_SYNTAX_UUID = UUID("8A885D04-1CEB-11C9-9FE8-08002B104860")
MIDL_LOOKUP_RE = re.compile(
    B'\x60\x00\x00\x00.{20}' + re.escape(DCE_SYNTAX_UUID.bytes_le),
    re.DOTALL
)
```

Disassembling Registration Parameters

- Using a disassembler, find all *RpcServerRegisterIf…* xrefs
- Parse function call arguments:



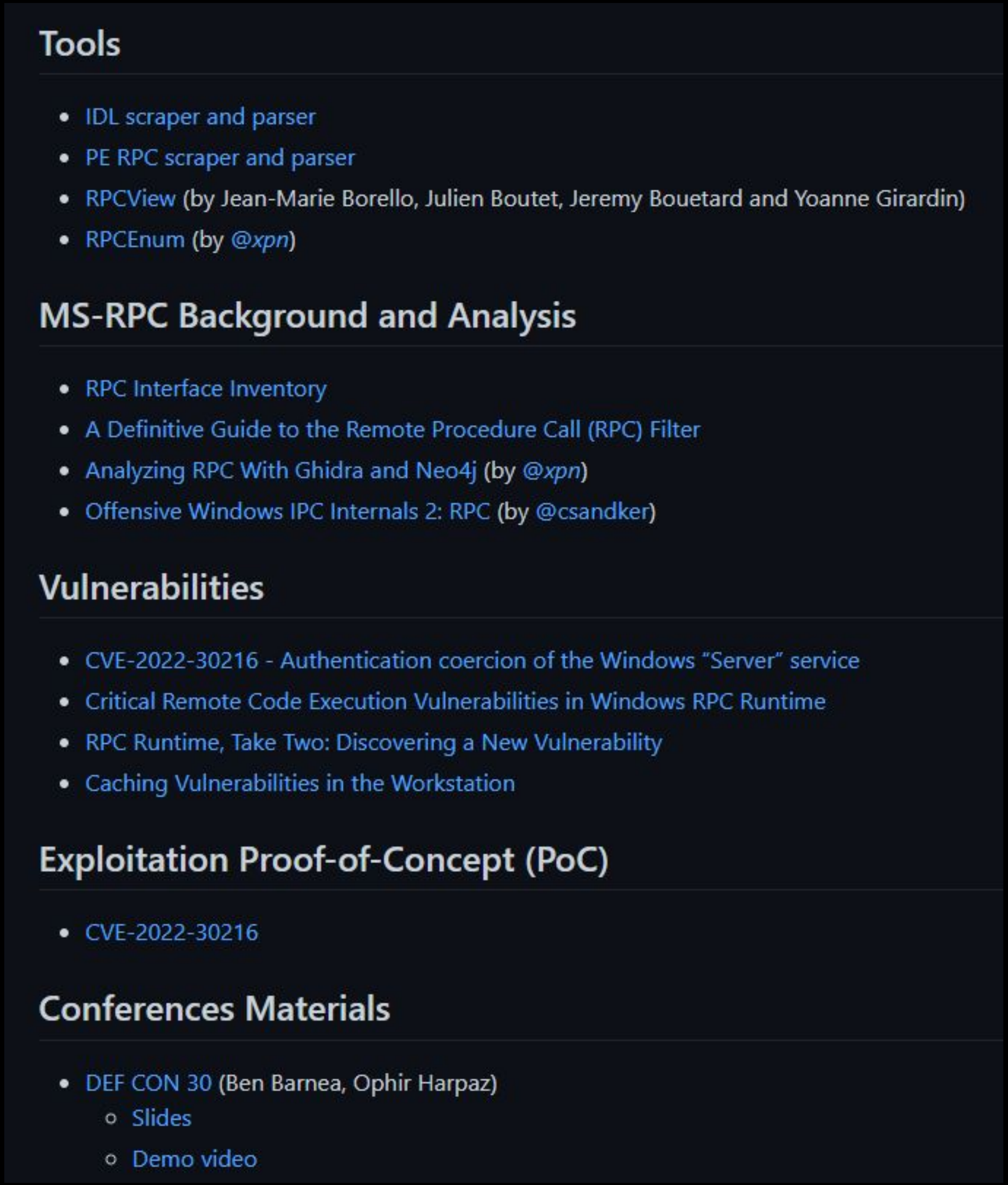
```
lea     rax, WsRpcSecurityCallback
mov     [rsp+38h+IfCallback], rax ; IfCallback
mov     [rsp+38h+MaxCalls], 4D2h ; MaxCalls
lea     r9d, [rbx+11h]  ; Flags
xor     r8d, r8d        ; MgrEpv
xor     edx, edx        ; MgrTypeUuid
lea     rcx, dword_18002F280 ; IfSpec
call    cs:__imp_RpcServerRegisterIfEx
```

Output:

```
"wimserv.exe": {
        "6d9fe472-30f1-4708-8fa8-678362b96155": {
            "number_of_functions": 3,
            "functions_pointers": [
                "0x140002650",
                …],
            "function_names": [
                "AddImage",
                …],
            "role": "server",
            "interface_address": "0x14006f9f0"
        },
```

RPC Toolkit

RPC Toolkit



Tools

- IDL scraper and parser
- PE RPC scraper and parser
- RPCView (by Jean-Marie Borello, Julien Boutet, Jeremy Bouetard and Yoanne Girardin)
- RPCEnum (by *@xpn*)

MS-RPC Background and Analysis

- RPC Interface Inventory
- A Definitive Guide to the Remote Procedure Call (RPC) Filter
- Analyzing RPC With Ghidra and Neo4j (by *@xpn*)
- Offensive Windows IPC Internals 2: RPC (by @csandker)

Vulnerabilities

- CVE-2022-30216 - Authentication coercion of the Windows "Server" service
- Critical Remote Code Execution Vulnerabilities in Windows RPC Runtime
- RPC Runtime, Take Two: Discovering a New Vulnerability
- Caching Vulnerabilities in the Workstation

Exploitation Proof-of-Concept (PoC)

- CVE-2022-30216

Conferences Materials

- DEF CON 30 (Ben Barnea, Ophir Harpaz)
  - Slides
  - Demo video

RPC Vulnerability Research Methodology - Recap

- RPC interface information can be found in PE files

RPC Vulnerability Research Methodology - Recap

- RPC interface information can be found in PE files
- By scraping the filesystem, and analyzing PE files we can:
  - Find all exposed functions

RPC Vulnerability Research Methodology - Recap

- RPC interface information can be found in PE files
- By scraping the filesystem, and analyzing PE files we can:
  - Find all exposed functions
  - Check if there’s a security callback and if caching is enabled

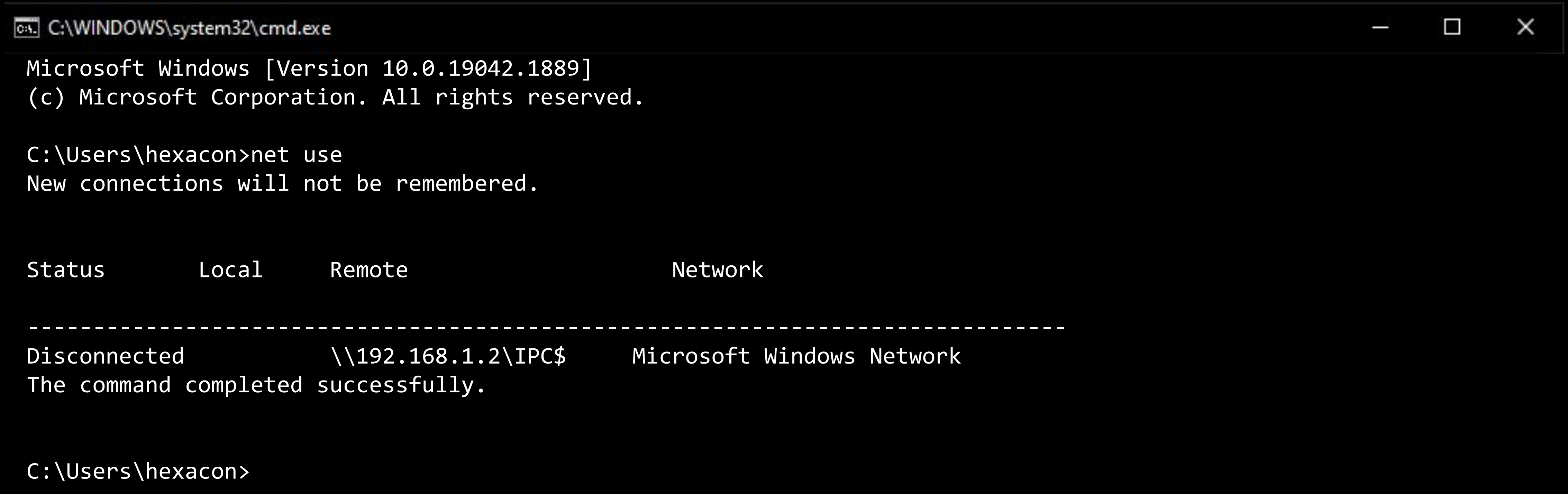
Akamai

CVE or it didn’t happen

Caching exploit discovery, attack flow & demo

The Workstation service (i.e. *LanmanWorkstation*)

Accessible through the `\pipe\wkssvc` named pipe



```
Microsoft Windows [Version 10.0.19042.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hexacon>net use
New connections will not be remembered.


Status       Local     Remote                    Network

-------------------------------------------------------------------------------
Disconnected           \\192.168.1.2\IPC$        Microsoft Windows Network
The command completed successfully.


C:\Users\hexacon>
```

Interface Registration

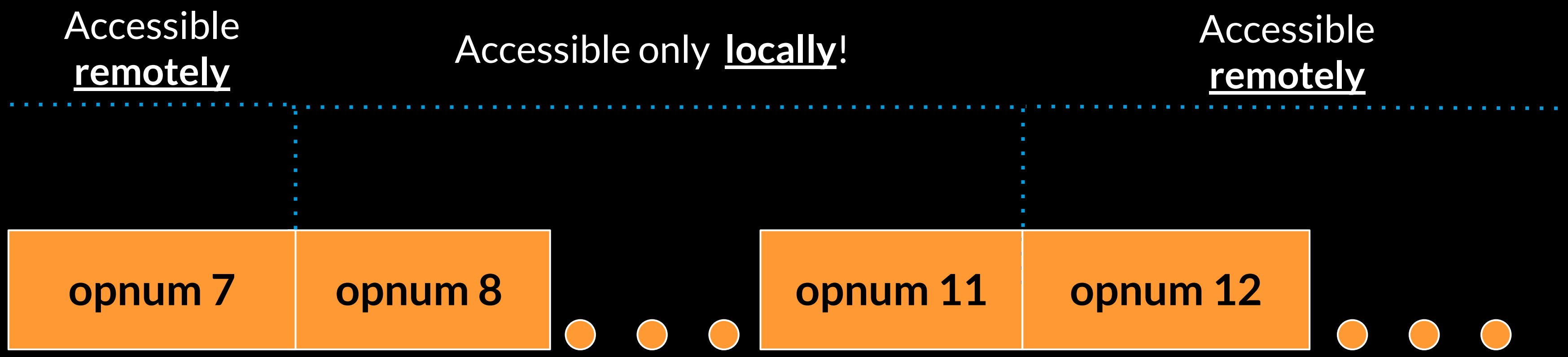
```
RpcServerRegisterIfEx(
  &unk_18002F280,
  0,
  0,
  0x11,
  0x4D2,
  WsRpcSecurityCallback
);
```

Interface Registration

```
                      RpcServerRegisterIfEx(
    RPC_IF_HANDLE  →    &unk_18002F280,
                        0,
                        0,
            Flags  →    0x11,
                        0x4D2,
Security Callback  →    WsRpcSecurityCallback
                      );
```

Workstation's Security Callback

```
if (
      (RpcCallAttributes.OpNum - 8) <= 3
      && (RpcCallAttributes.IsClientLocal != 1)
)
   return ERROR_ACCESS_DENIED;
```



Accessible **remotely**

Accessible only **locally**!

Accessible **remotely**

opnum 7

opnum 8

opnum 11

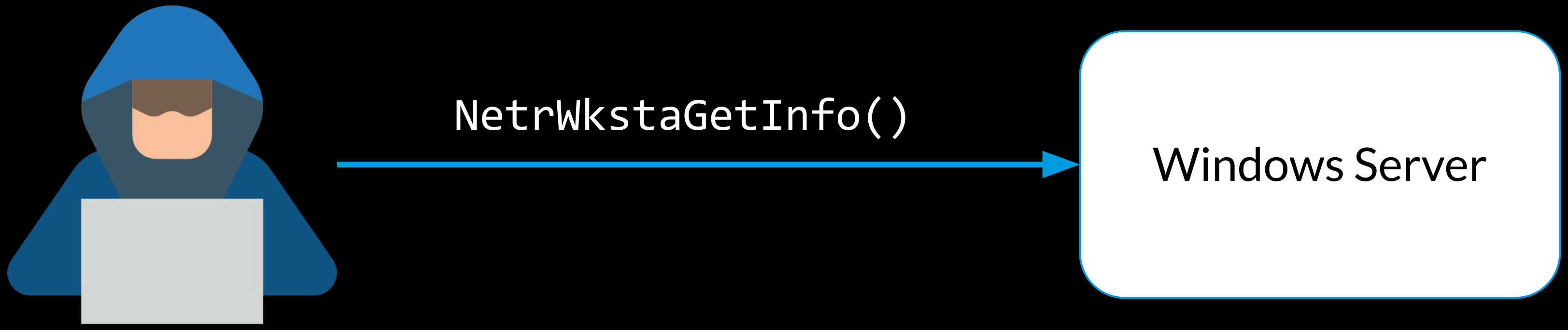
opnum 12

What’s the Cache?

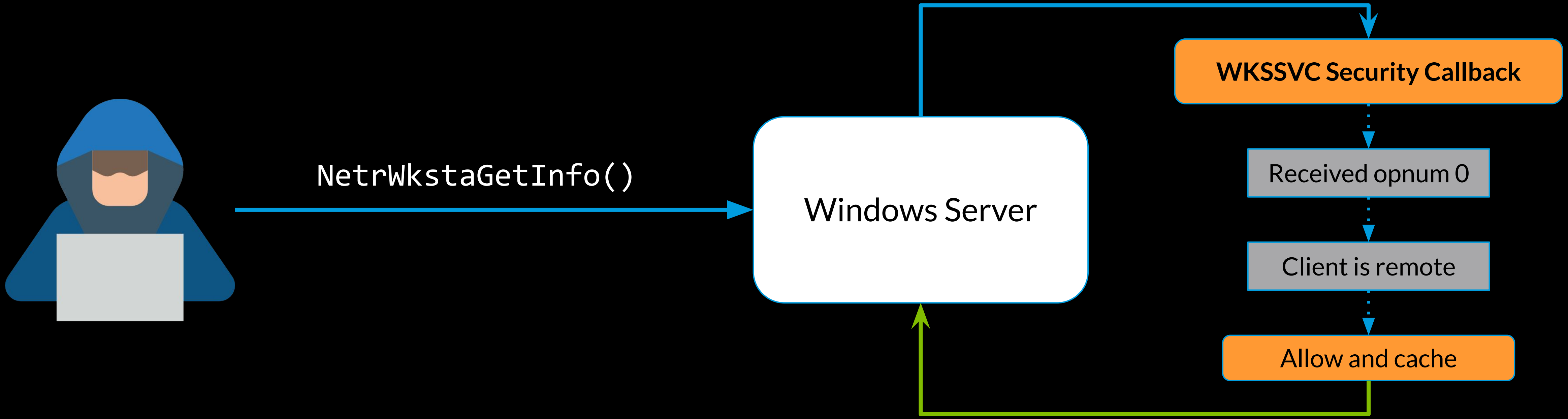
What’s the Cache?

```
off_18002E970   dq offset NetrWkstaGetInfo
                dq offset NetrWkstaSetInfo
                dq offset NetrWkstaUserEnum
                dq offset NetrWkstaUserGetInfo
                dq offset NetrWkstaUserSetInfo
                dq offset NetrWkstaTransportEnum
                dq offset NetrWkstaTransportAdd
                dq offset NetrWkstaTransportDel
                dq offset NetrUseAdd
                dq offset NetrUseGetInfo
                dq offset NetrUseDel
                dq offset NetrUseEnum
                dq offset NetrUnjoinDomain
                dq offset NetrWorkstationStatisticsGet
                dq offset NetrUnjoinDomain
```

Attack Flow



Attack Flow



NetrWkstaGetInfo()

Windows Server

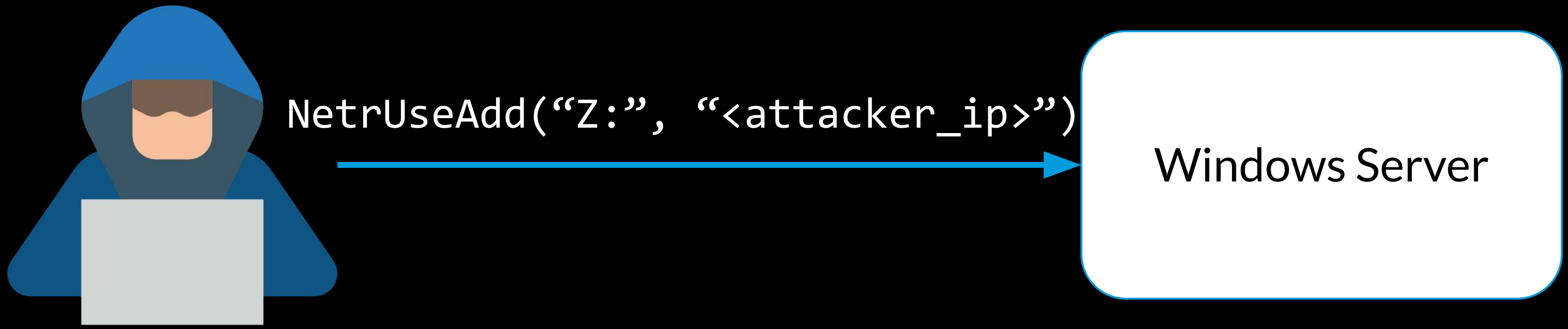
WKSSVC Security Callback

Received opnum 0

Client is remote

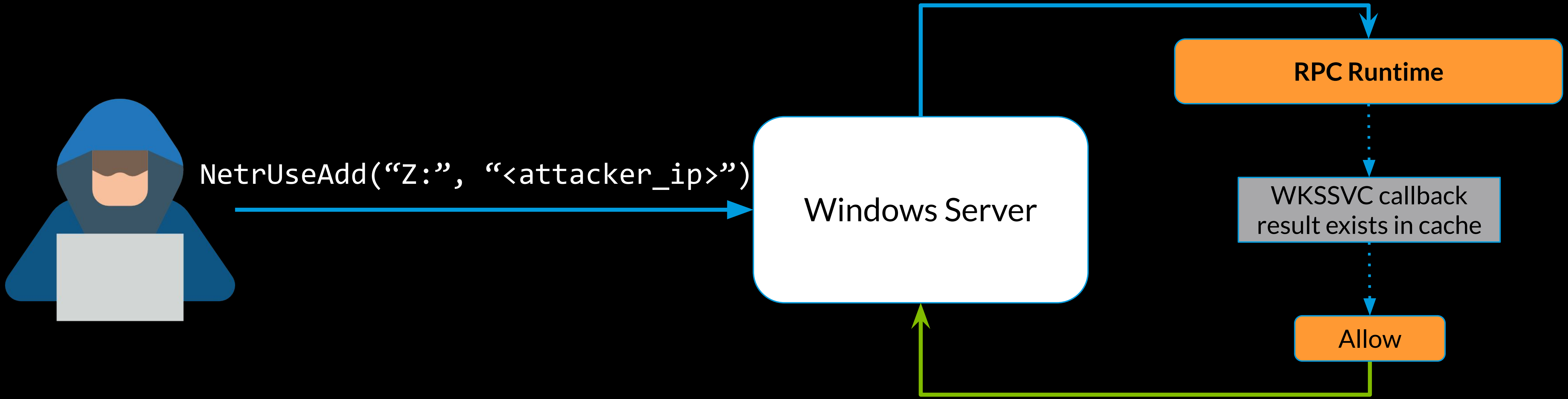
Allow and cache

Attack Flow

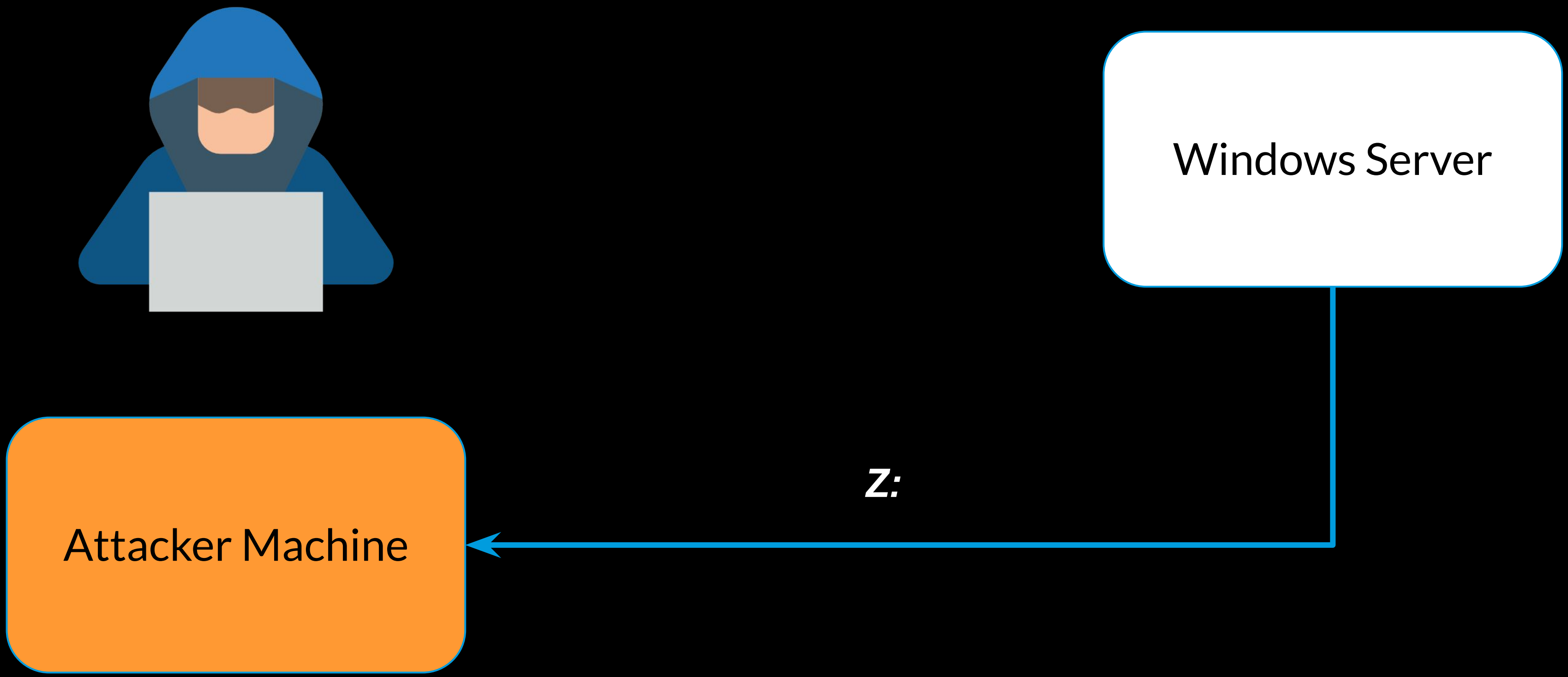


NetrUseAdd(“Z:”, “<attacker_ip>”)

Attack Flow


NetrUseAdd(“Z:”, “<attacker_ip>”)
Windows Server
RPC Runtime
WKSSVC callback result exists in cache
Allow

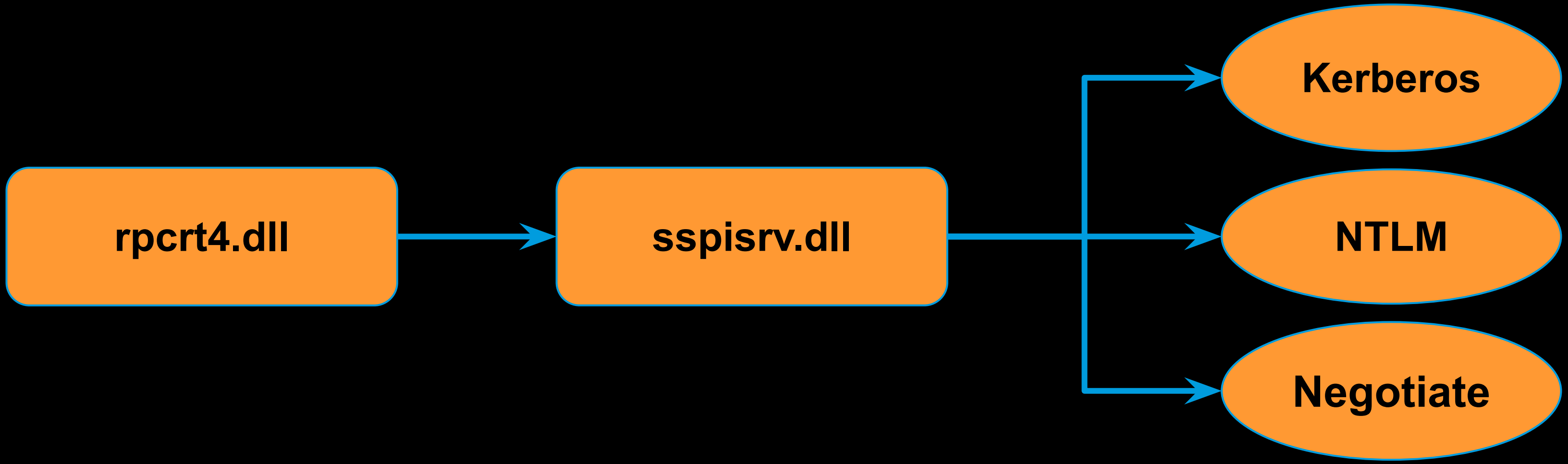
Attack Flow



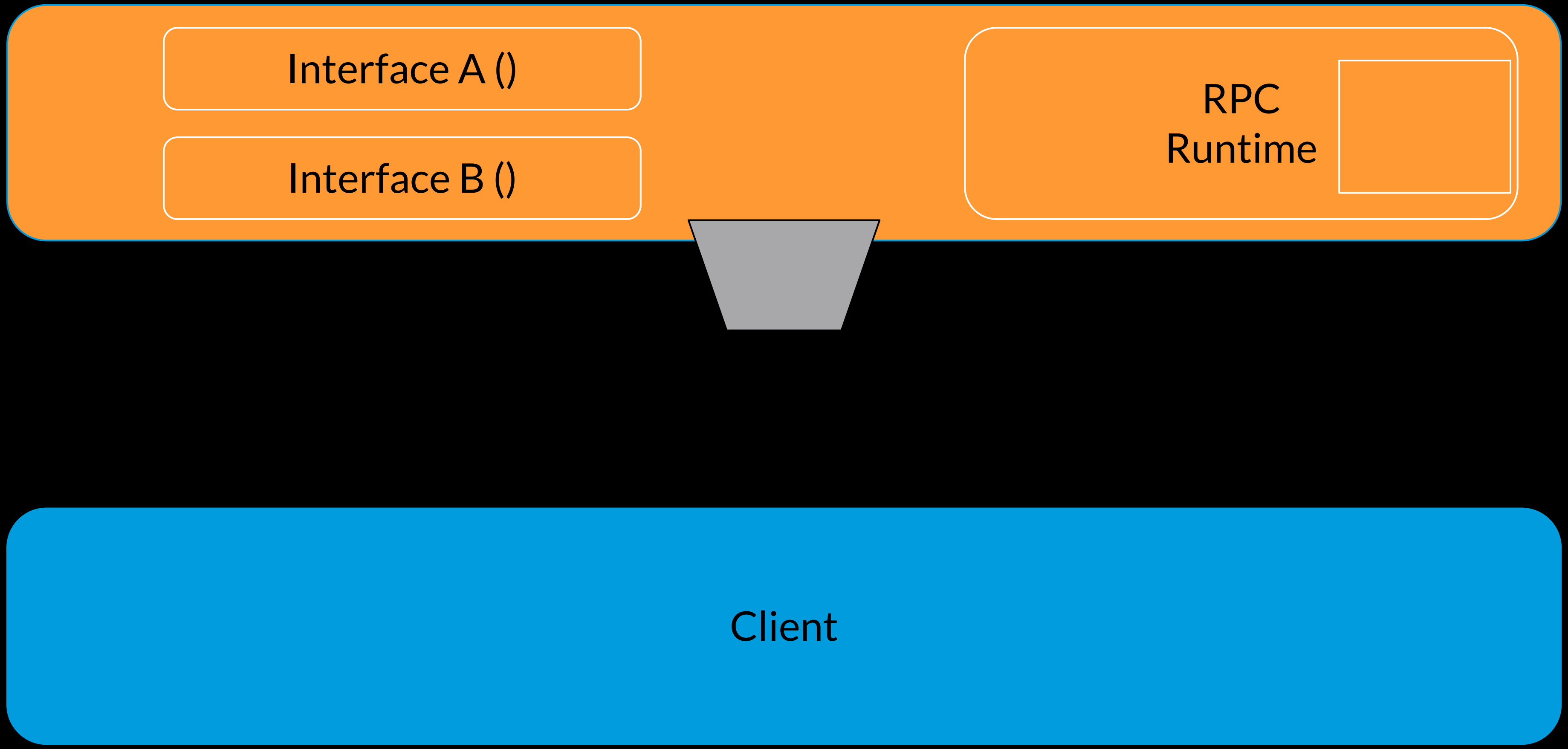
YOU CVE YET?

“SSPI Multiplexing”

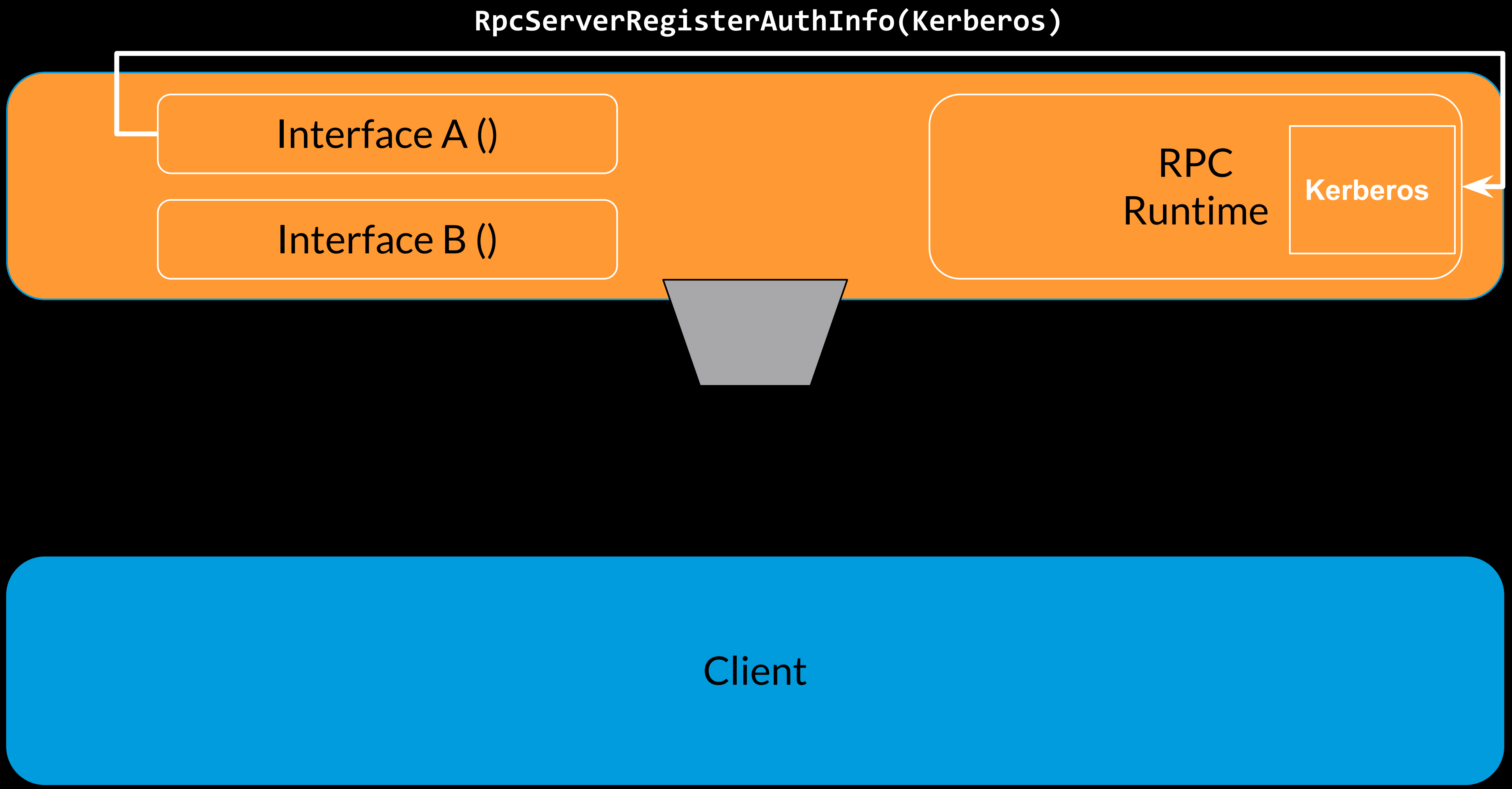
- Authentication in RPC is implemented with the Security Support Provider Interface (SSPI)
- RPC servers wishing to use authentication must instruct the RPC runtime to load the corresponding SSPI



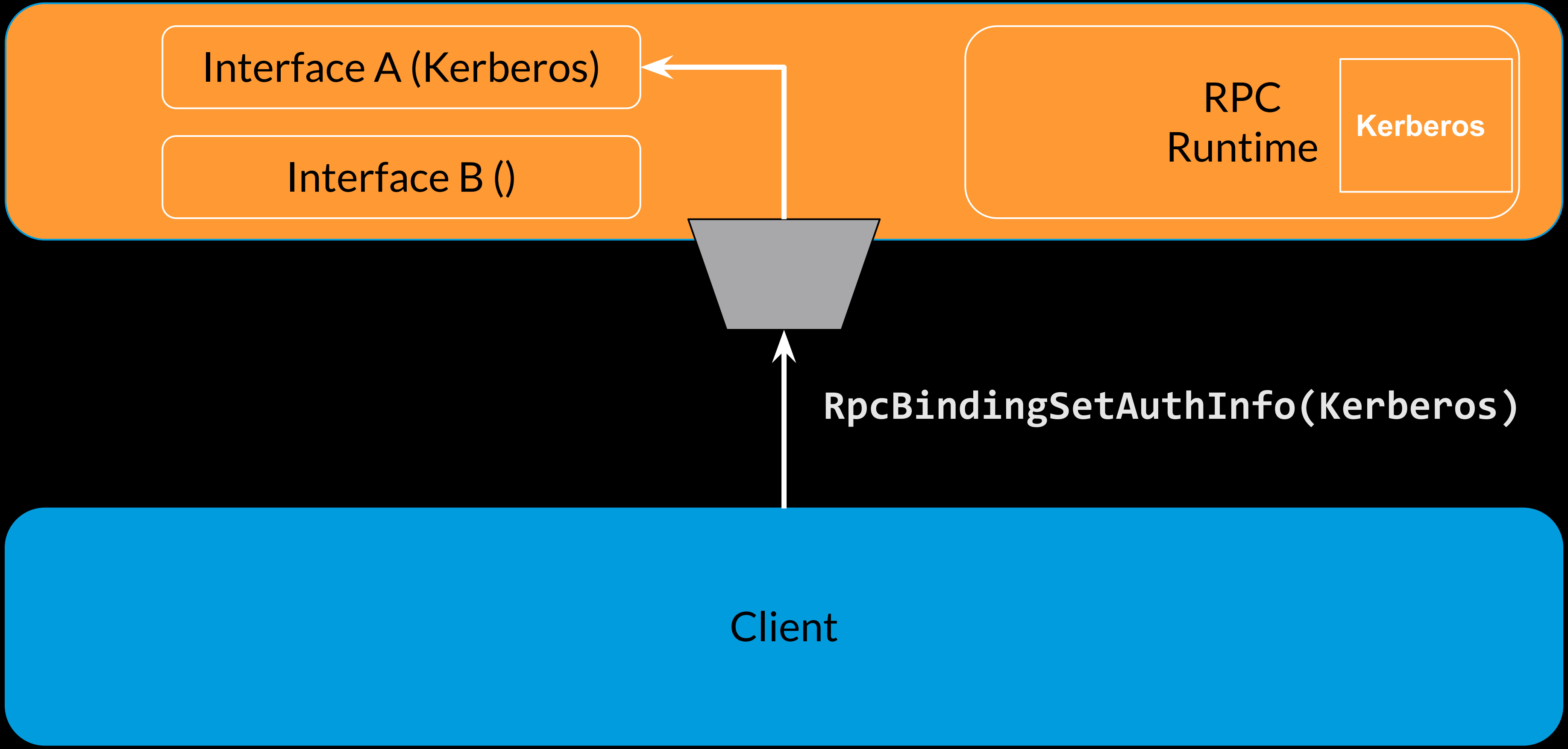
“SSPI Multiplexing”



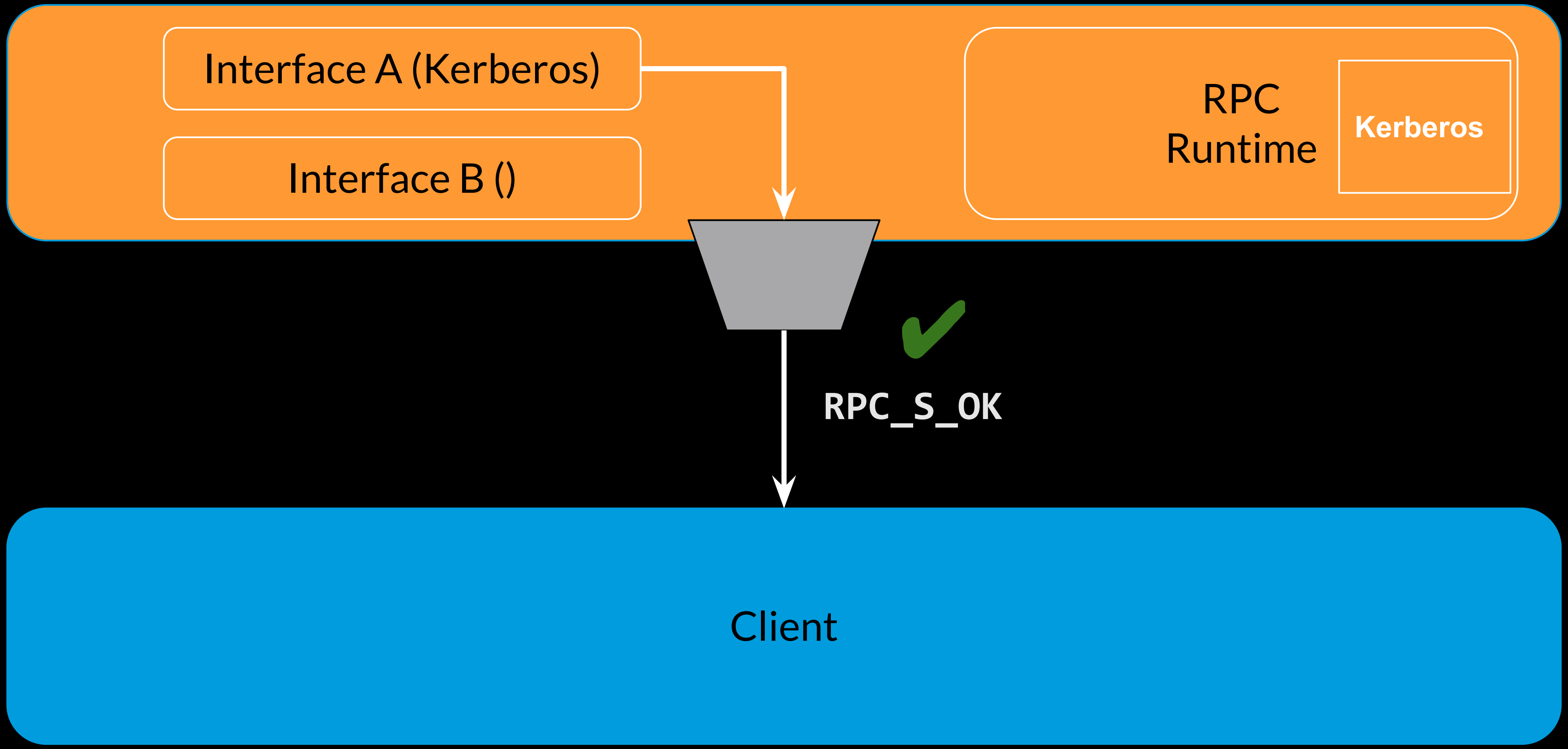
“SSPI Multiplexing”



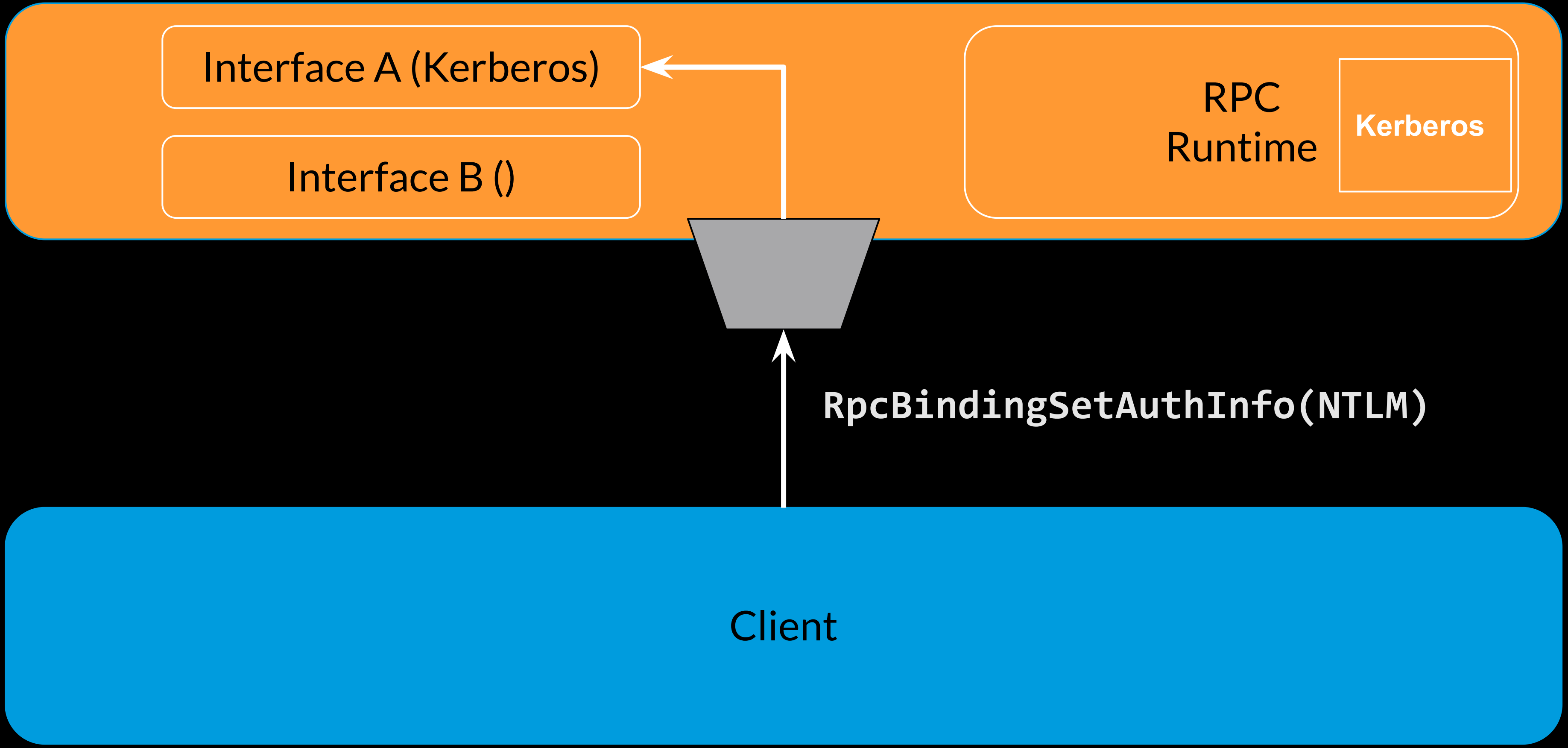
“SSPI Multiplexing”



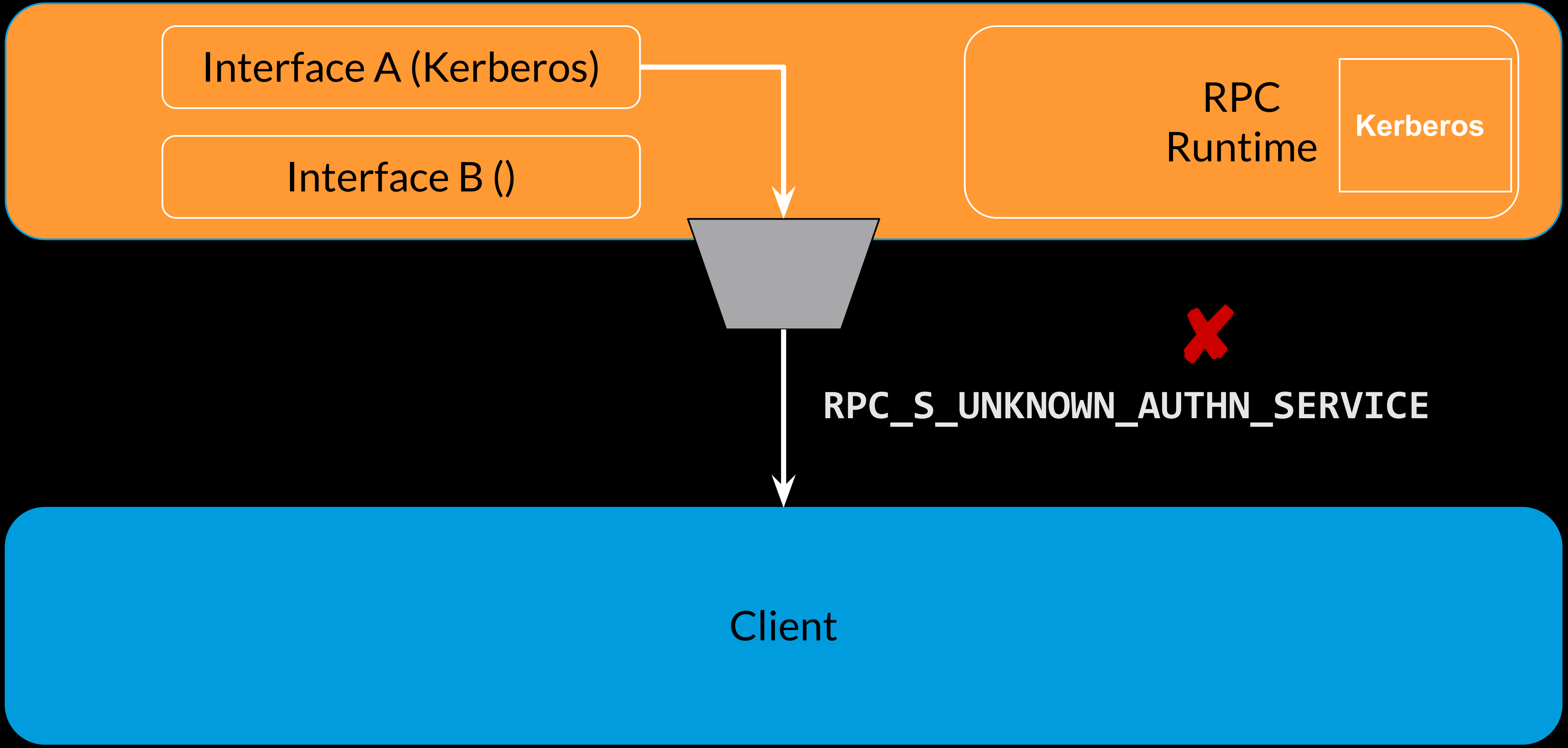
“SSPI Multiplexing”



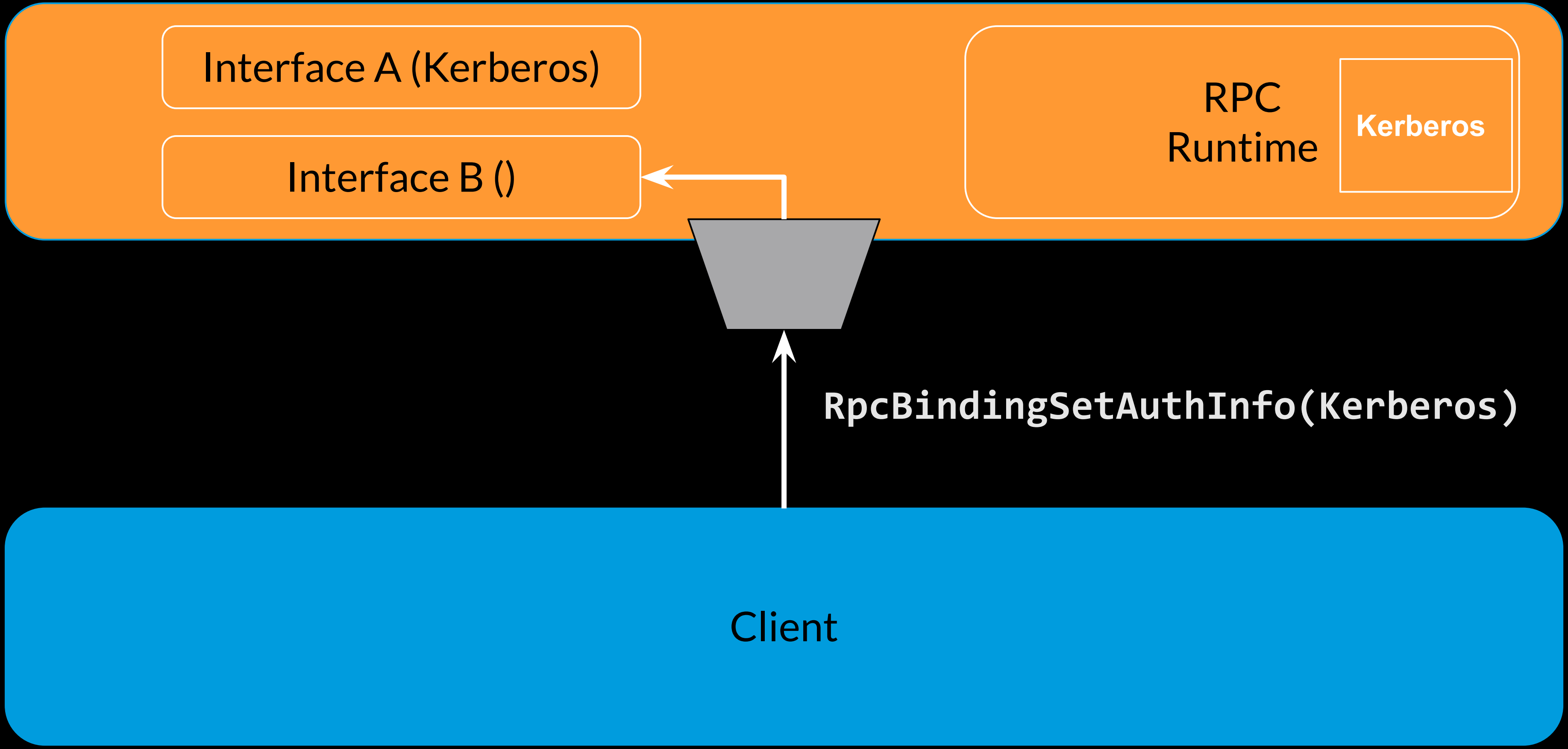
“SSPI Multiplexing”



“SSPI Multiplexing”



“SSPI Multiplexing”



Interface A (Kerberos)

Interface B ()

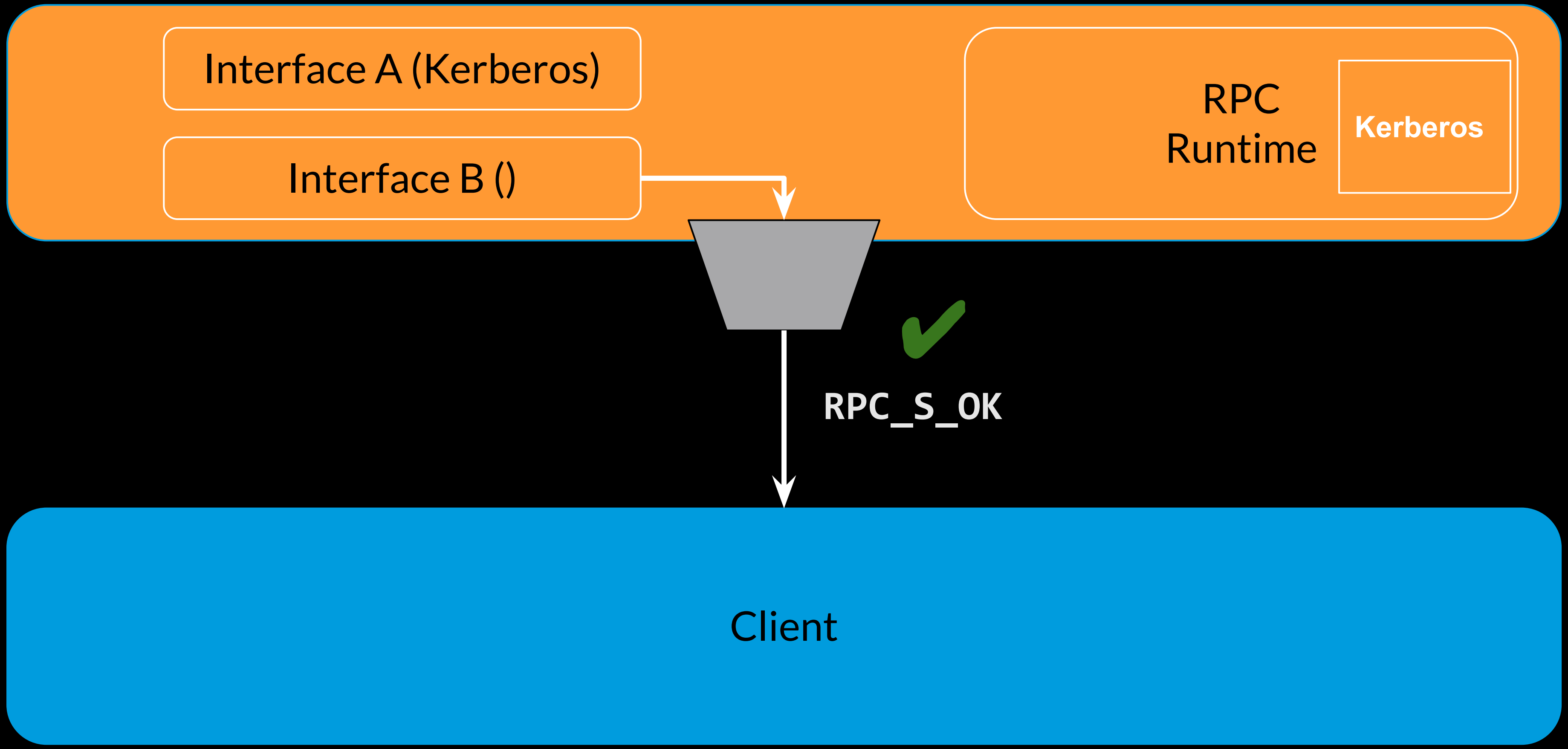
RPC Runtime

Kerberos

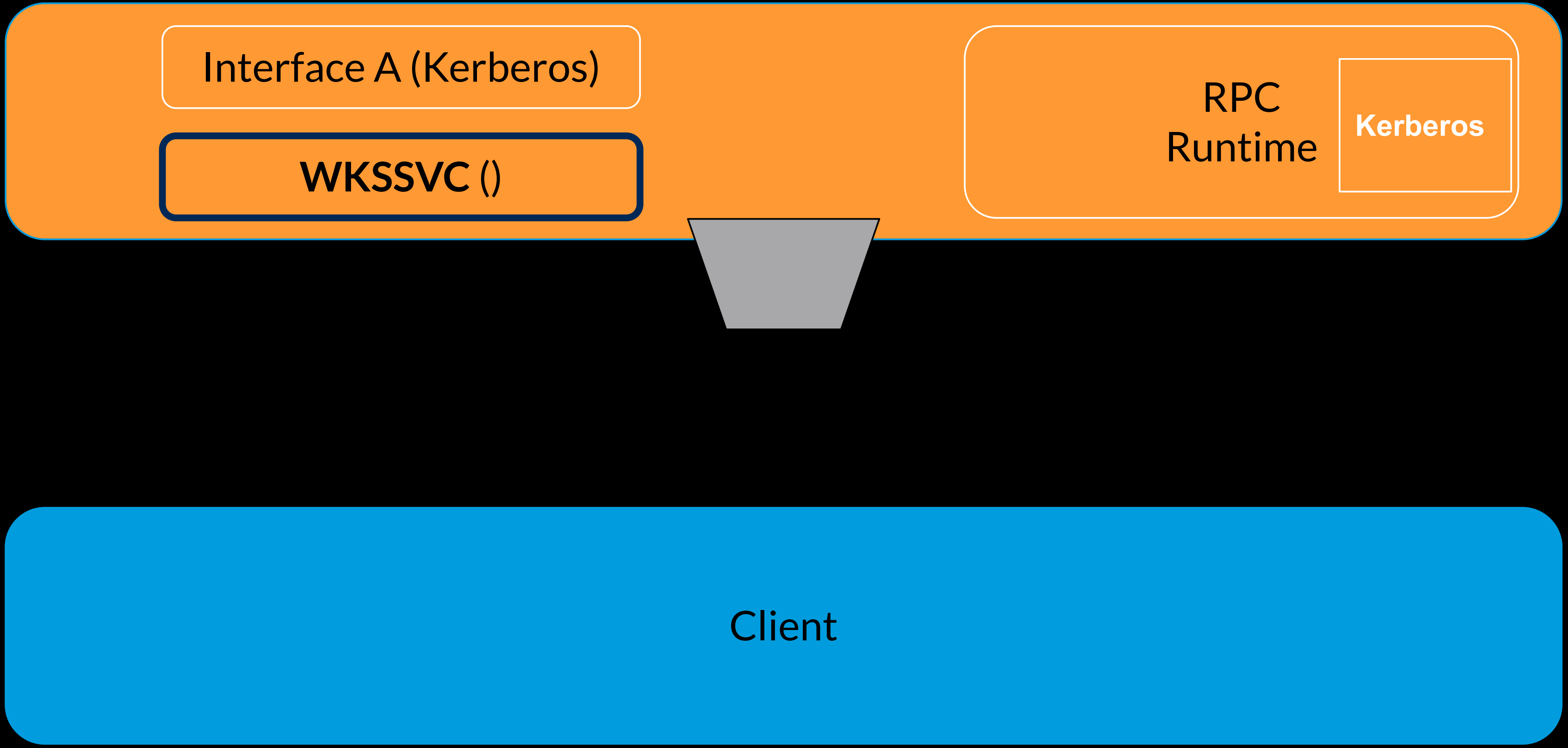
RpcBindingSetAuthInfo(Kerberos)

Client

“SSPI Multiplexing”



“SSPI Multiplexing”



Interface A (Kerberos)

WKSSVC ()

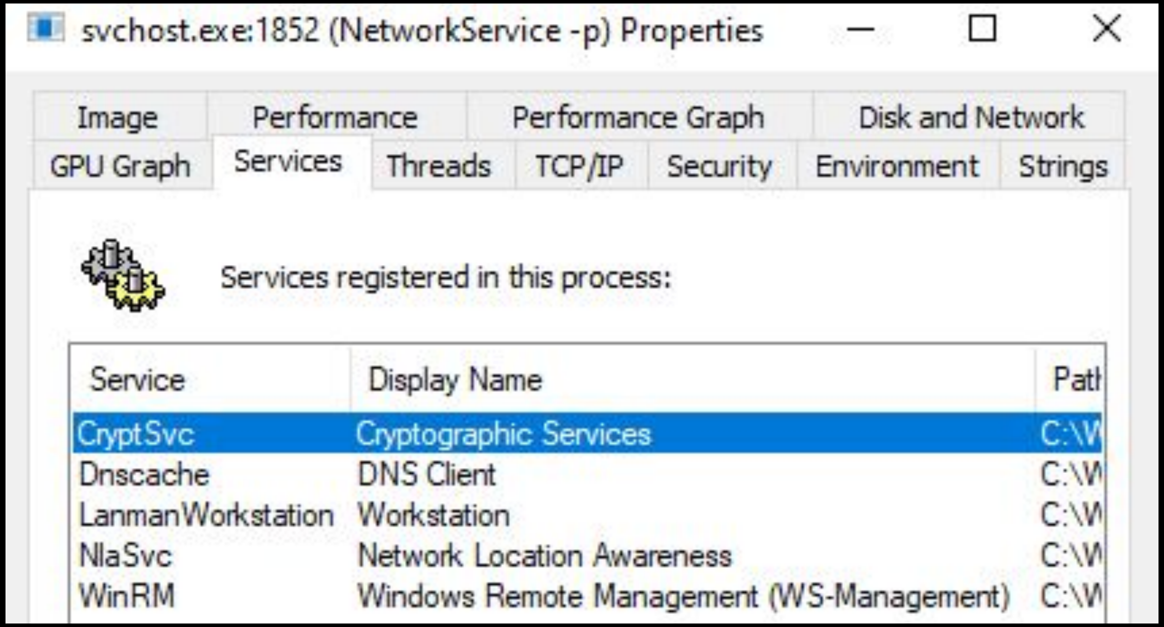
RPC Runtime

Kerberos

Client

Plex That WKS

- WKSSVC is part of the *NetworkProvider* service group
- Other services in that group register auth providers



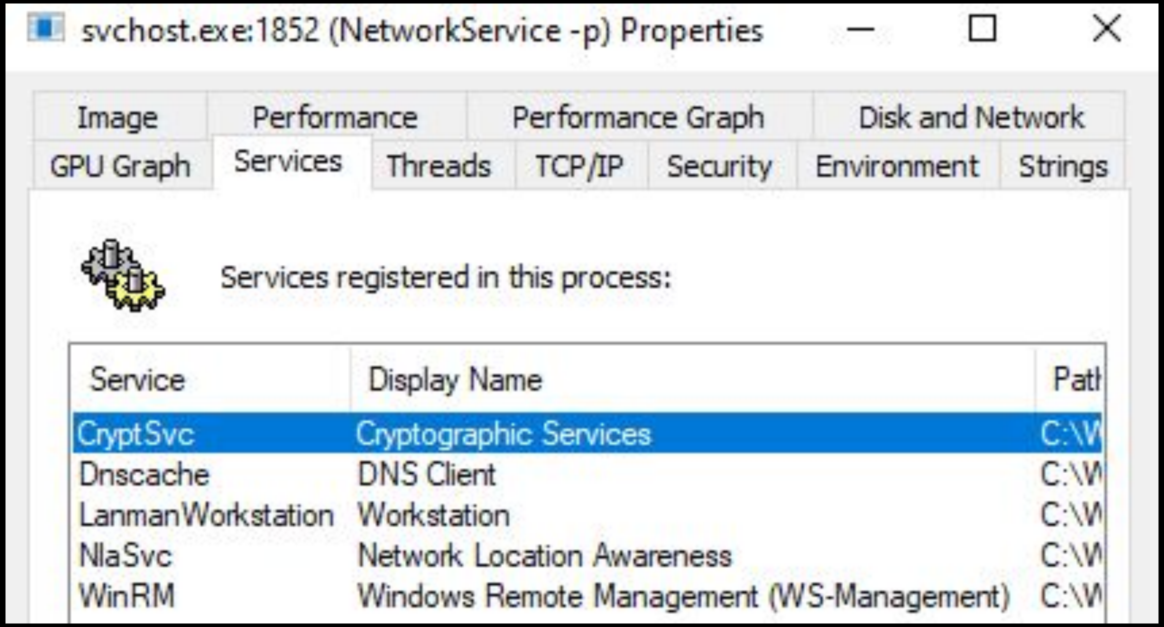
svchost.exe:1852 (NetworkService -p) Properties

Services registered in this process:

Service	Display Name	Patl
CryptSvc	Cryptographic Services	C:\W
Dnscache	DNS Client	C:\W
LanmanWorkstation	Workstation	C:\W
NlaSvc	Network Location Awareness	C:\W
WinRM	Windows Remote Management (WS-Management)	C:\W

Plex That WKS

- WKSSVC is part of the *NetworkProvider* service group
- Other services in that group register auth providers
- Multiplexing breaks with Windows 10 1703+
    - Service separation on by default

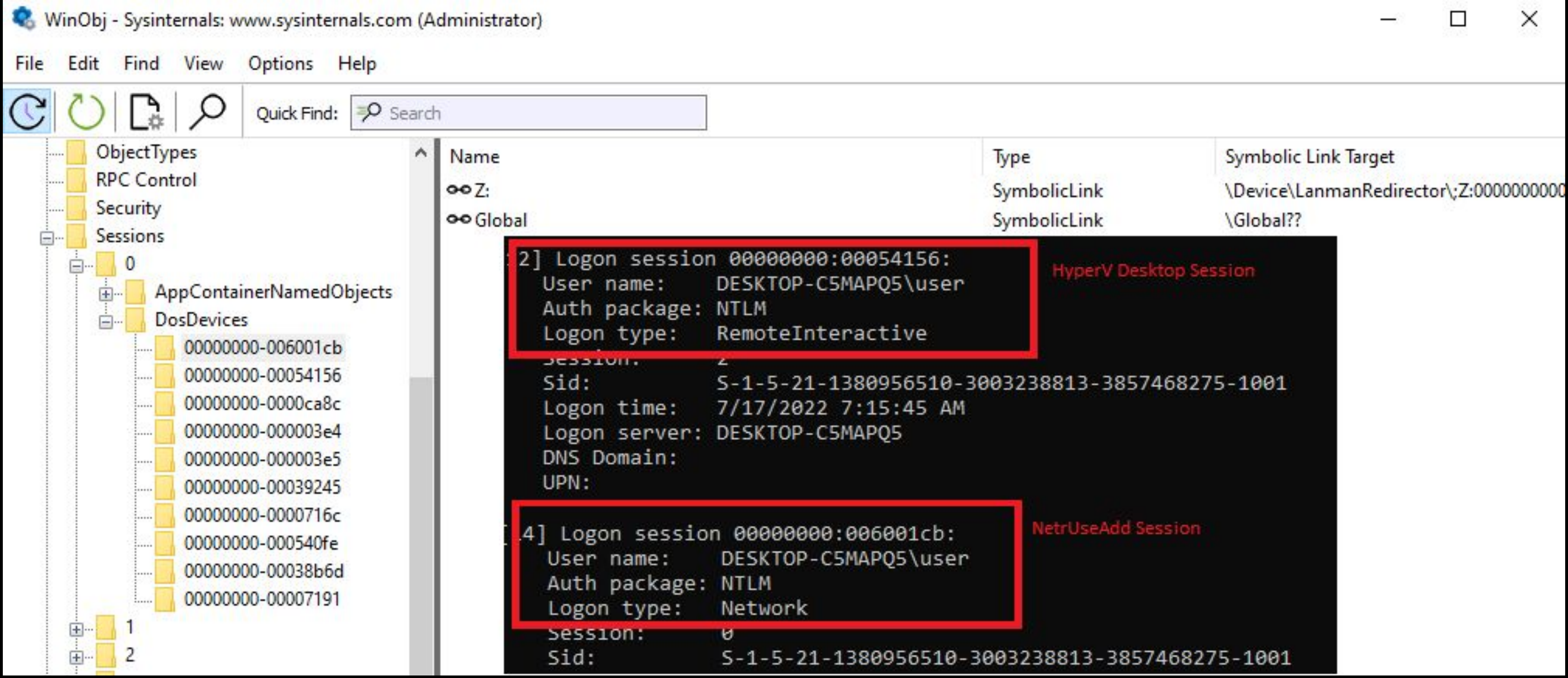


Services registered in this process:

Service	Display Name	Path
CryptSvc	Cryptographic Services	C:\W
Dnscache	DNS Client	C:\W
LanmanWorkstation	Workstation	C:\W
NlaSvc	Network Location Awareness	C:\W
WinRM	Windows Remote Management (WS-Management)	C:\W

YOU CVE YET?

So Close, Yet So Far Away



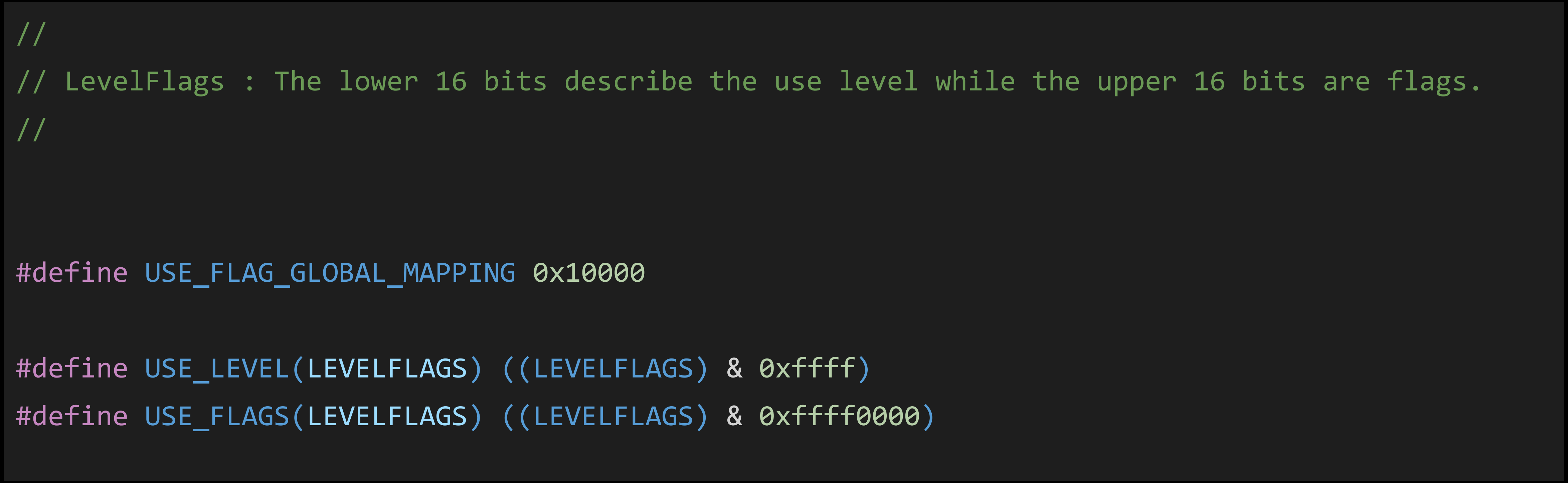
Name	Type	Symbolic Link Target
Z:	SymbolicLink	\Device\LanmanRedirector\;Z:0000000000
Global	SymbolicLink	\Global??

```
2] Logon session 00000000:00054156:
   User name:    DESKTOP-C5MAPQ5\user
   Auth package: NTLM
   Logon type:   RemoteInteractive
   Session:      2
   Sid:          S-1-5-21-1380956510-3003238813-3857468275-1001
   Logon time:   7/17/2022 7:15:45 AM
   Logon server: DESKTOP-C5MAPQ5
   DNS Domain:
   UPN:

[4] Logon session 00000000:006001cb:
   User name:    DESKTOP-C5MAPQ5\user
   Auth package: NTLM
   Logon type:   Network
   Session:      0
   Sid:          S-1-5-21-1380956510-3003238813-3857468275-1001
```

HyperV Desktop Session

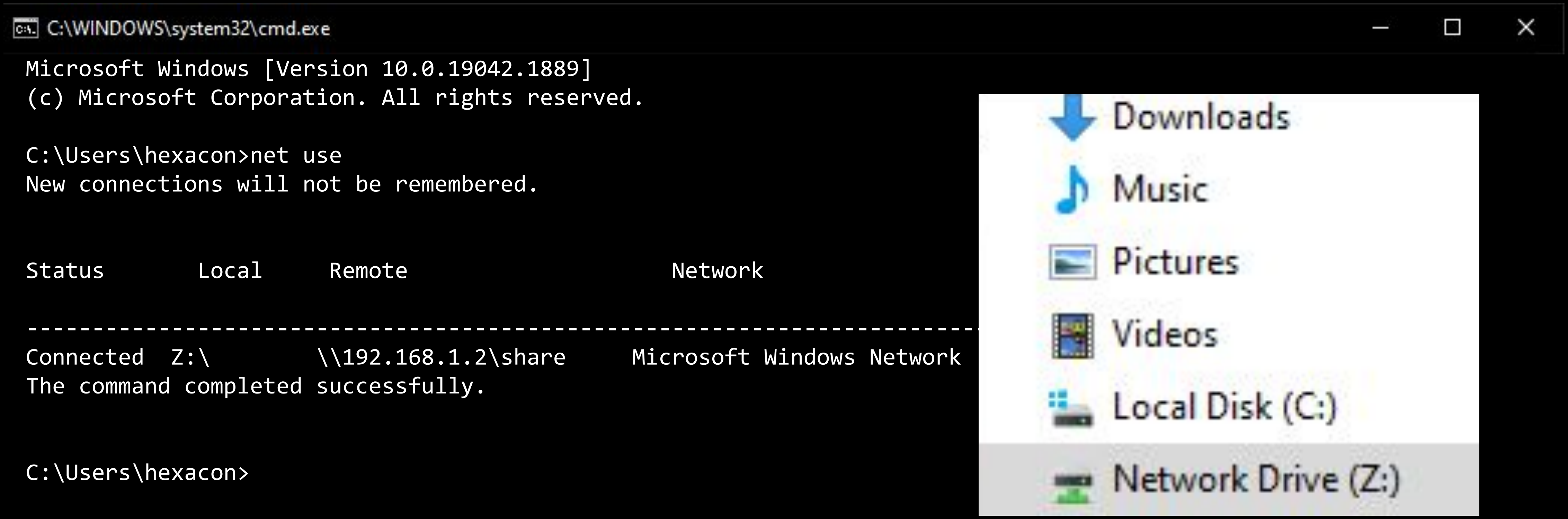
NetrUseAdd Session

Never Give Up, Never Give In



```
//
// LevelFlags : The lower 16 bits describe the use level while the upper 16 bits are flags.
//


#define USE_FLAG_GLOBAL_MAPPING 0x10000

#define USE_LEVEL(LEVELFLAGS) ((LEVELFLAGS) & 0xffff)
#define USE_FLAGS(LEVELFLAGS) ((LEVELFLAGS) & 0xffff0000)
```

** Defined in LMUse.h*

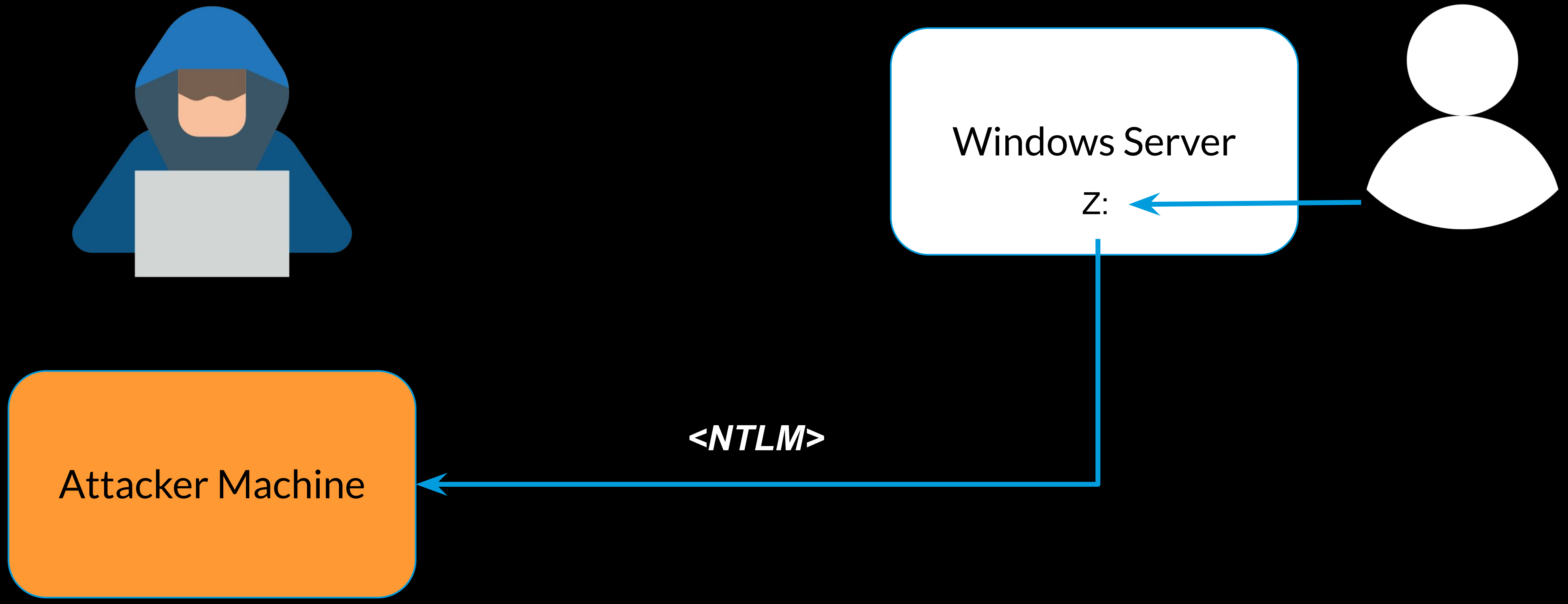


```
Microsoft Windows [Version 10.0.19042.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hexacon>net use
New connections will not be remembered.


Status       Local     Remote                    Network

-------------------------------------------------------------------------
Connected   Z:\        \\192.168.1.2\share     Microsoft Windows Network
The command completed successfully.


C:\Users\hexacon>
```

CVE-2022-38034 — Elevation of Privilege

- Create global mapping to a file share in our control
- Requirements:
  - Windows version earlier than Windows 10 version 1703
  - OR any Windows machine with less than 3.5GB RAM

Attack Flow



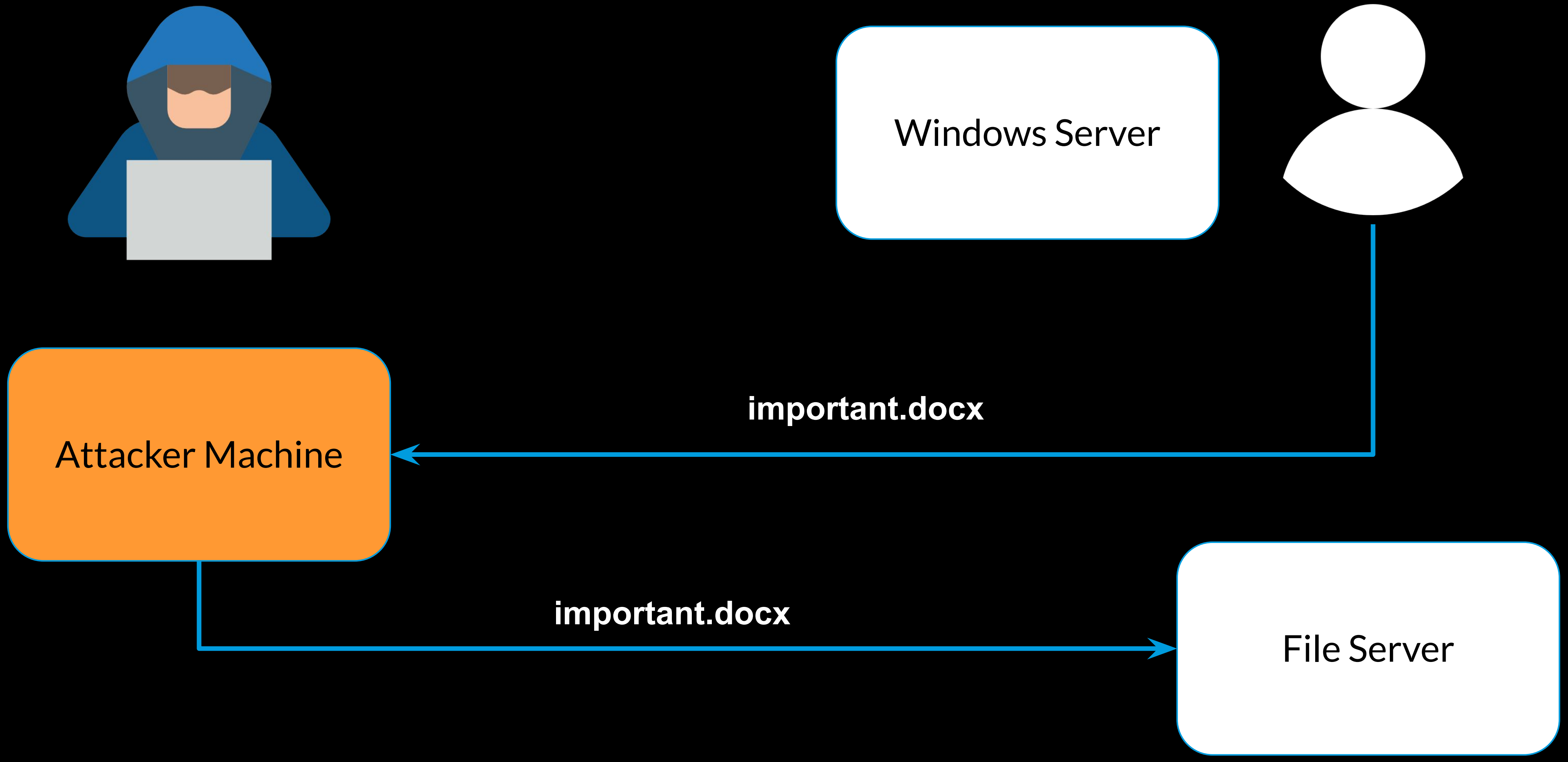
Windows Server

Z:

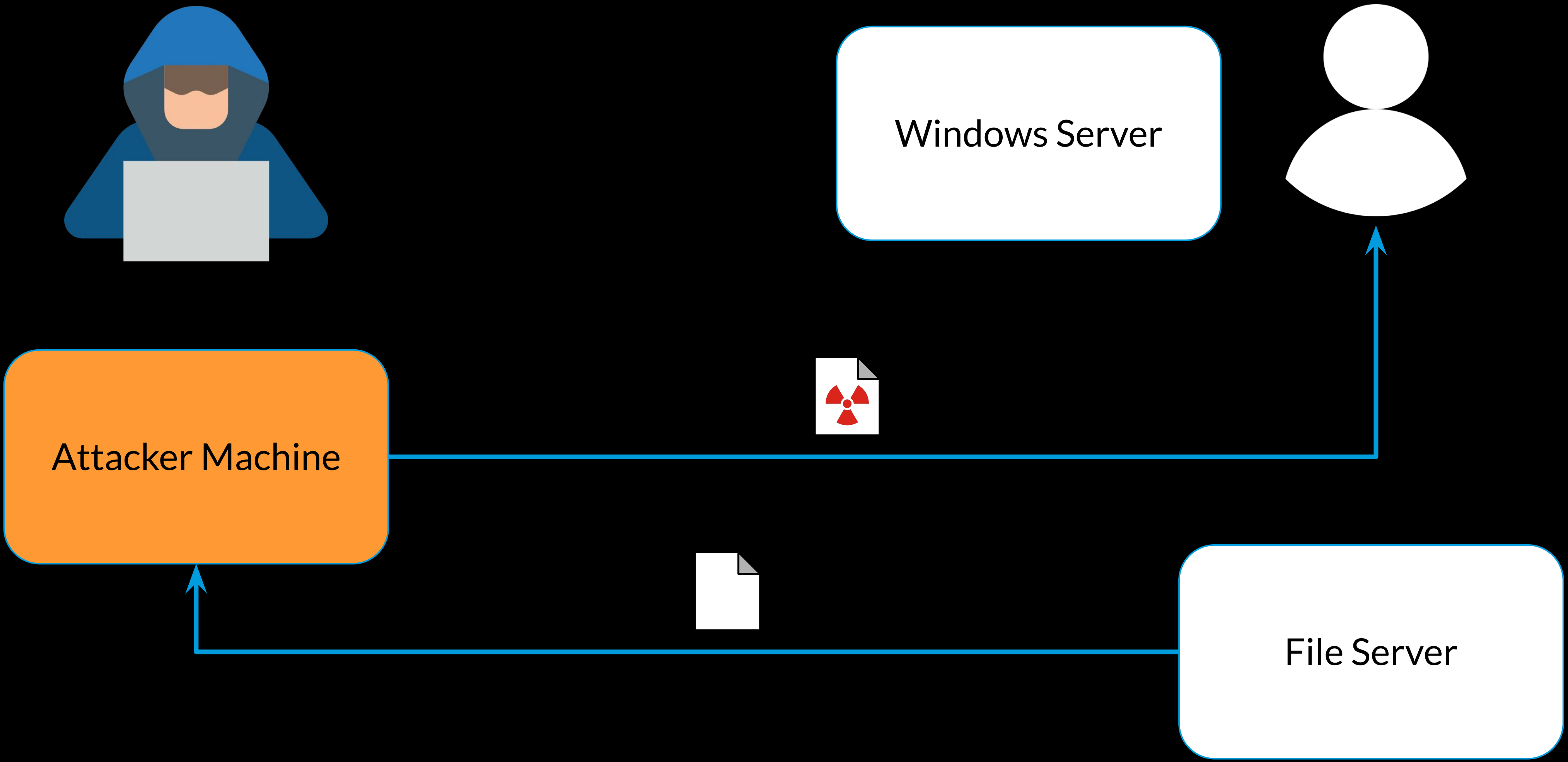
<NTLM>

Attacker Machine

Attack Flow


Windows Server
Attacker Machine
important.docx
important.docx
File Server

Attack Flow



Windows Server

Attacker Machine

File Server

Exploit Demo

Summary

- Security callbacks are an interesting attack surface
- We share automation tools & resources in our RPC Toolkit
- Future research directions
  - More services
  - Caching attacks that don’t involve opnums
  - More automation

Thank you

Questions?

@kupsul

@OphirHarpaz

[RPC Toolkit]

[Cold Hard Cache – Bypassing RPC Interface Security with Cache Abuse]