

**COM IN
60
SECONDS**

James Forshaw (@tiraniddo)
Infiltrate 2017

Agenda

- Component Object Model (COM) Internals
- Attacking COM
 - Enumerating attack surface for EoP
 - Reverse engineering COM components
- Bugs and “Features”

We'll be using my OleViewDotNet tool throughout.

<https://github.com/tyranid/oleviewdotnet>

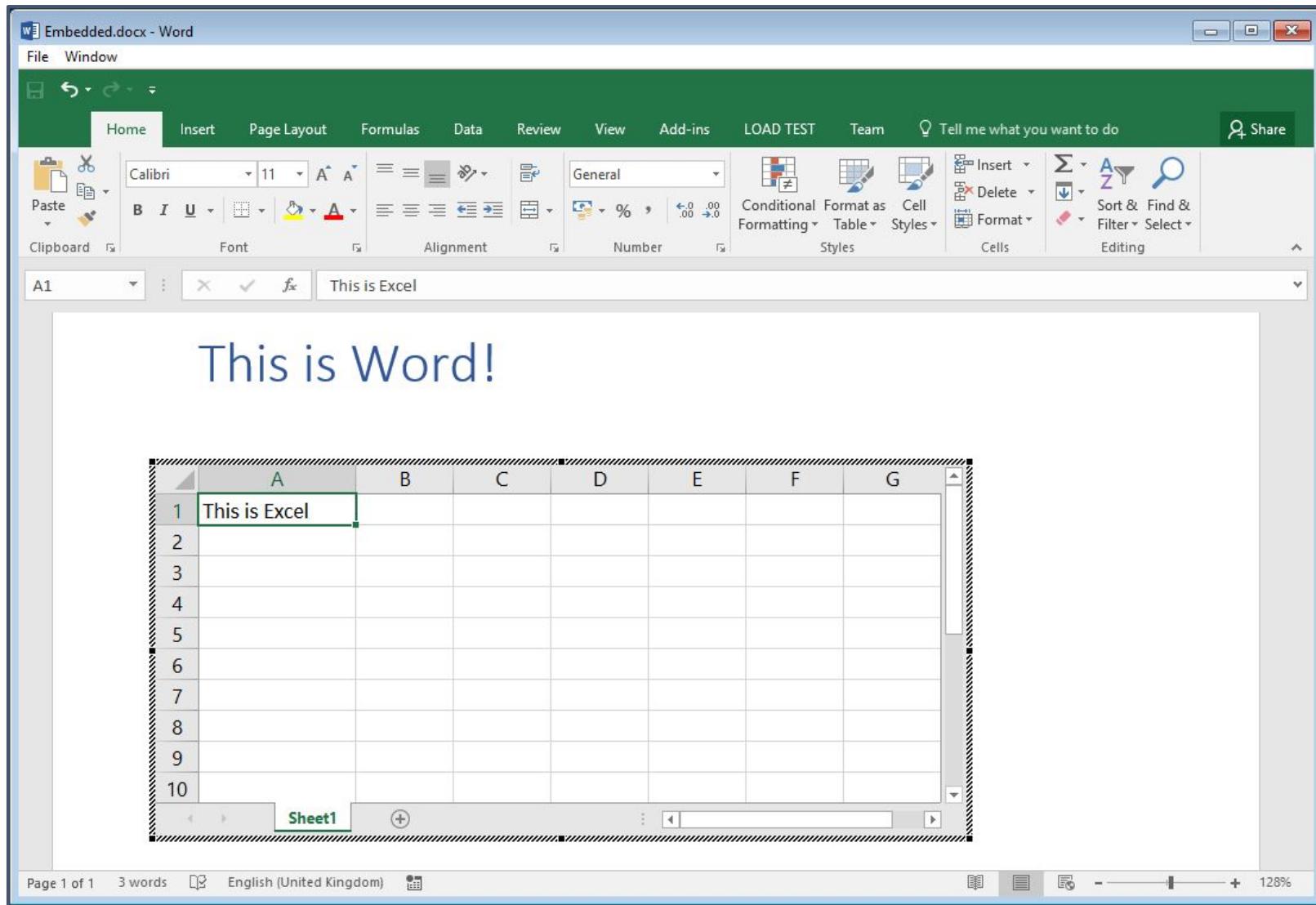
COM Internals



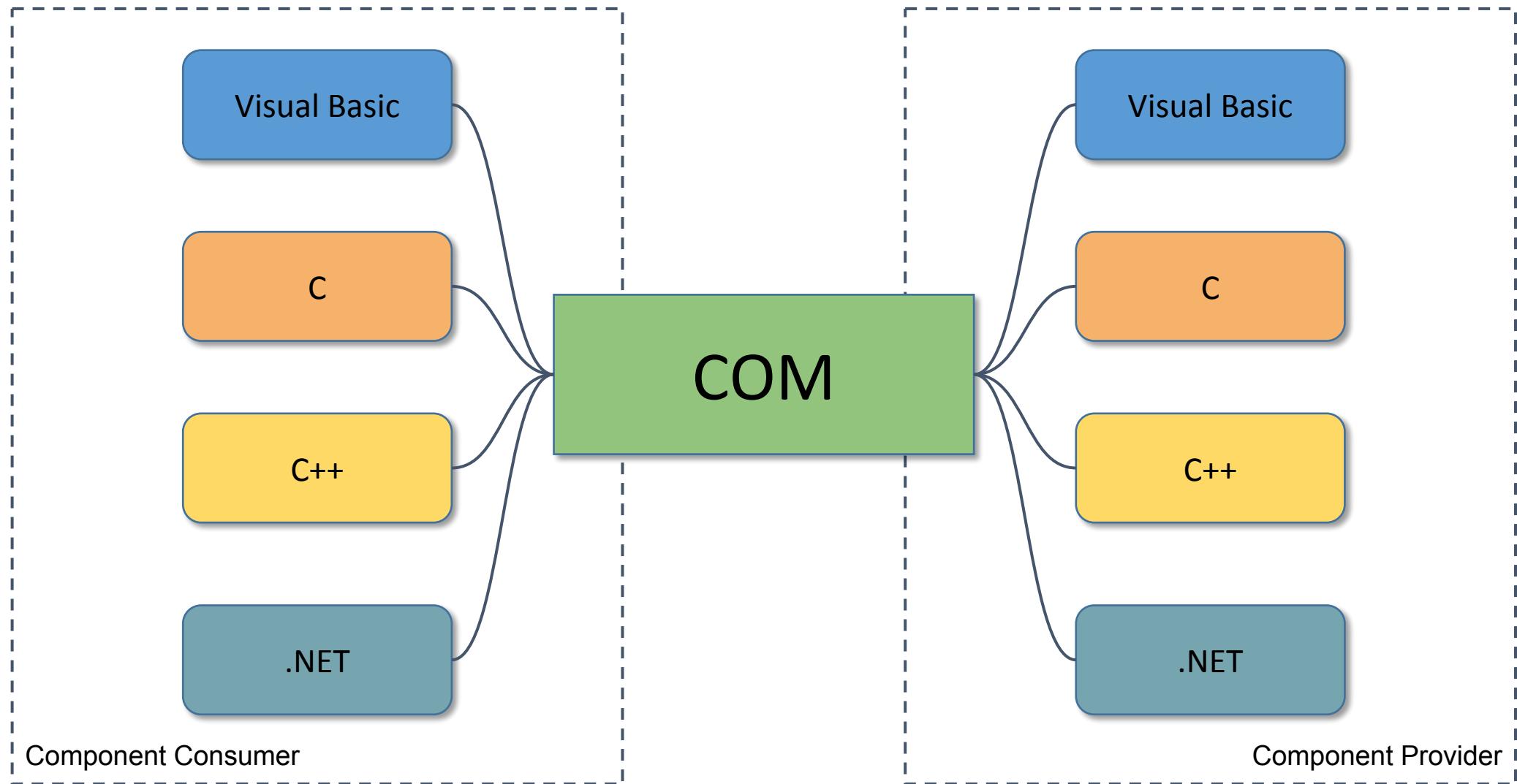
**“Any sufficiently complex
middleware is
indistinguishable from
magic.”**

Arthur C. Clarke’s Third Law of Software Development

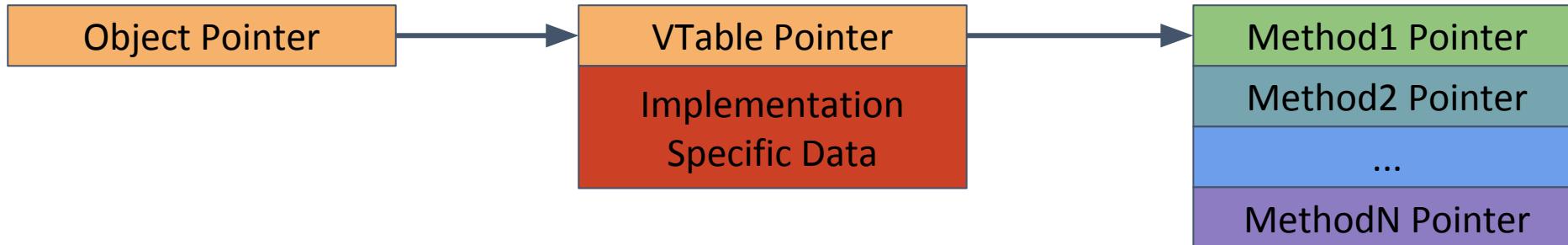
In the Beginning was OLE



Interoperability Heaven



Common ABI



```
struct ObjectVTable {  
    void (*SetInt)(struct Object* This, int i);  
    int  (*GetInt)(struct Object* This);  
};
```

```
struct Object {  
    struct ObjectVTable* Vtbl;  
    // Implementation specific data follows.  
};
```

```
struct Object* obj;  
obj->Vtbl->SetInt(obj, 1234);
```

Object pointer is first.
Arguments passed left to right

VTable Pointer at start

The Casting Problem

```
struct Interface1 {  
    virtual void A() = 0;  
};  
  
struct Interface2 {  
    virtual void B() = 0;  
};  
  
class Object : public Interface1,  
               public Interface2 {  
    void A() {}  
    void B() {}  
};
```

Define a
pure virtual
“Interface”

Derive from
Interface and
implement

```
// Okay (mostly).  
Interface1* intf1 = new Object;  
// Incredibly bad idea.  
Interface2* intf2 = (Interface2*)intf1;
```

No no no!!!

IUnknown, the Root of all COM Evil

```
DEFINE_GUID(IID_IUnknown, 00000000-0000-0000-C00-00000000046");
struct IUnknown {
    HRESULT QueryInterface(GUID& iid, void** ppv);
    LONG AddRef();
    LONG Release();
};
```

Used to reference count the object.

Interfaces defined using a 128 bit Globally Unique ID.

```
struct Interface1 : public IUnknown {};
struct Interface2 : public IUnknown {};

Interface2* intf2;
if (intf1->QueryInterface(IID_Interface2,
                           (void**)&intf2) >= 0) {
    // Success, we can call methods.
    intf2->Release();
}
```

Cast is a GIANT code smell!
↖(ツ)↗

Class Registration

The screenshot shows the Windows Registry Editor window. The left pane displays a tree view of registry keys under the root key `HKEY_CLASSES_ROOT\CLSID\{00000316-0000-0000-C000-000000000046}\InprocServer32`. The right pane shows a table with three columns: Name, Type, and Data. One entry is visible in the table:

Name	Type	Data
<code>ab</code> (Default)	REG_SZ	combase.dll

The status bar at the bottom of the window shows the full path: `Computer\HKEY_CLASSES_ROOT\CLSID\{00000316-0000-0000-C000-000000000046}\InprocServer32`.

Class Factories

```
DEFINE_GUID(IID_ClassFactory,
    "00000001-0000-0000-C000-000000000046");

struct IClassFactory : public IUnknown {
    HRESULT CreateInstance(
        IUnknown **pUnkOuter,
        REFIID riid,
        void **ppvObject);

    HRESULT LockServer(BOOL fLock);
};
```

Creating Class Factories and Instances

```
HRESULT CoGetClassObject (
```

REFCLSID	rclsid,
DWORD	dwClsContext,
COSERVERINFO	*pServerInfo,
REFIID	riid,
LPVOID	*ppv

```
) ;
```

Specify remote server
information is required
(more on this later).

Specifies what type of server to lookup:

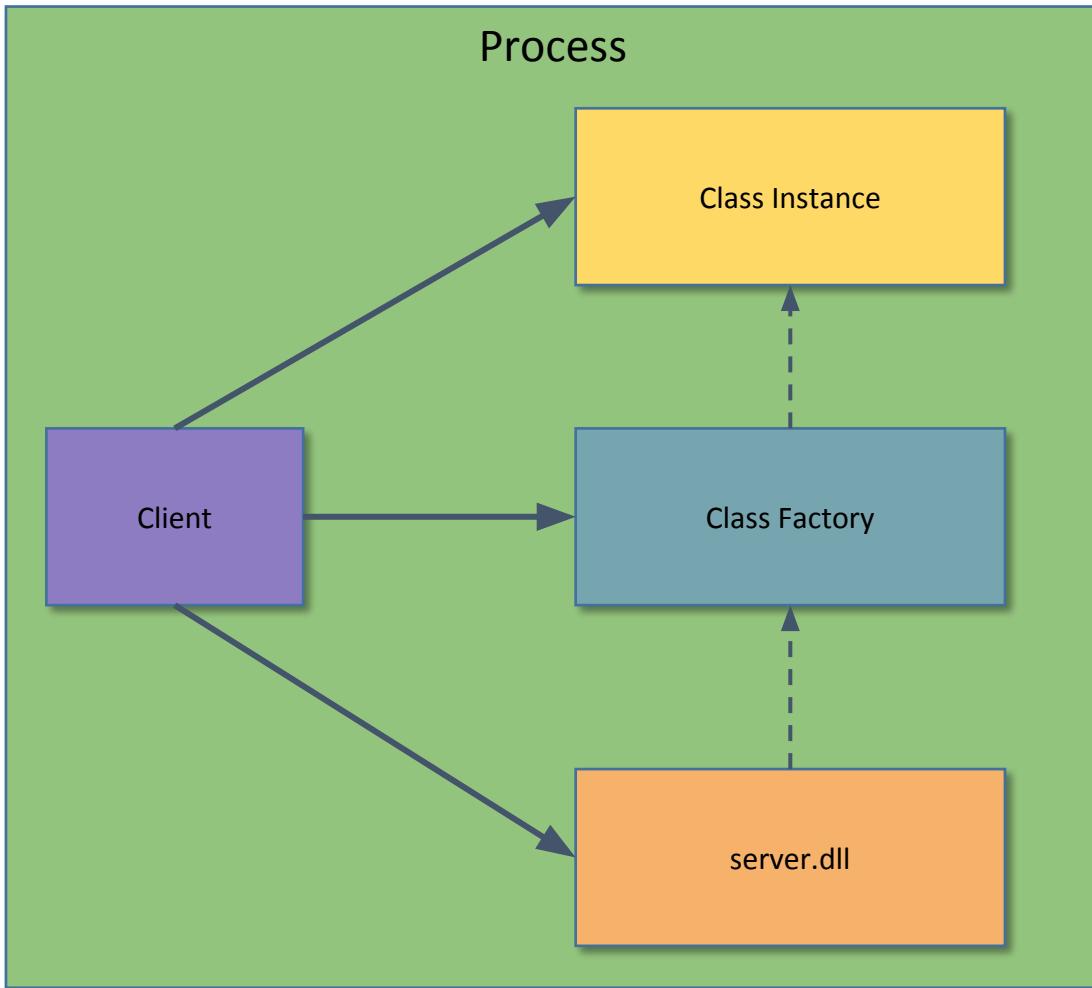
- CLSCTX_INPROC_SERVER
- CLSCTX_INPROC_HANDLER
- CLSCTX_LOCAL_SERVER
- CLSCTX_REMOTE_SERVER

```
HRESULT CoCreateInstanceEx (
```

REFCLSID	rclsid,
IUnknown	*punkOuter,
DWORD	dwClCtx,
COSERVERINFO	*pServerInfo,
DWORD	dwCount,
MULTI_QI	*pResults

```
) ;
```

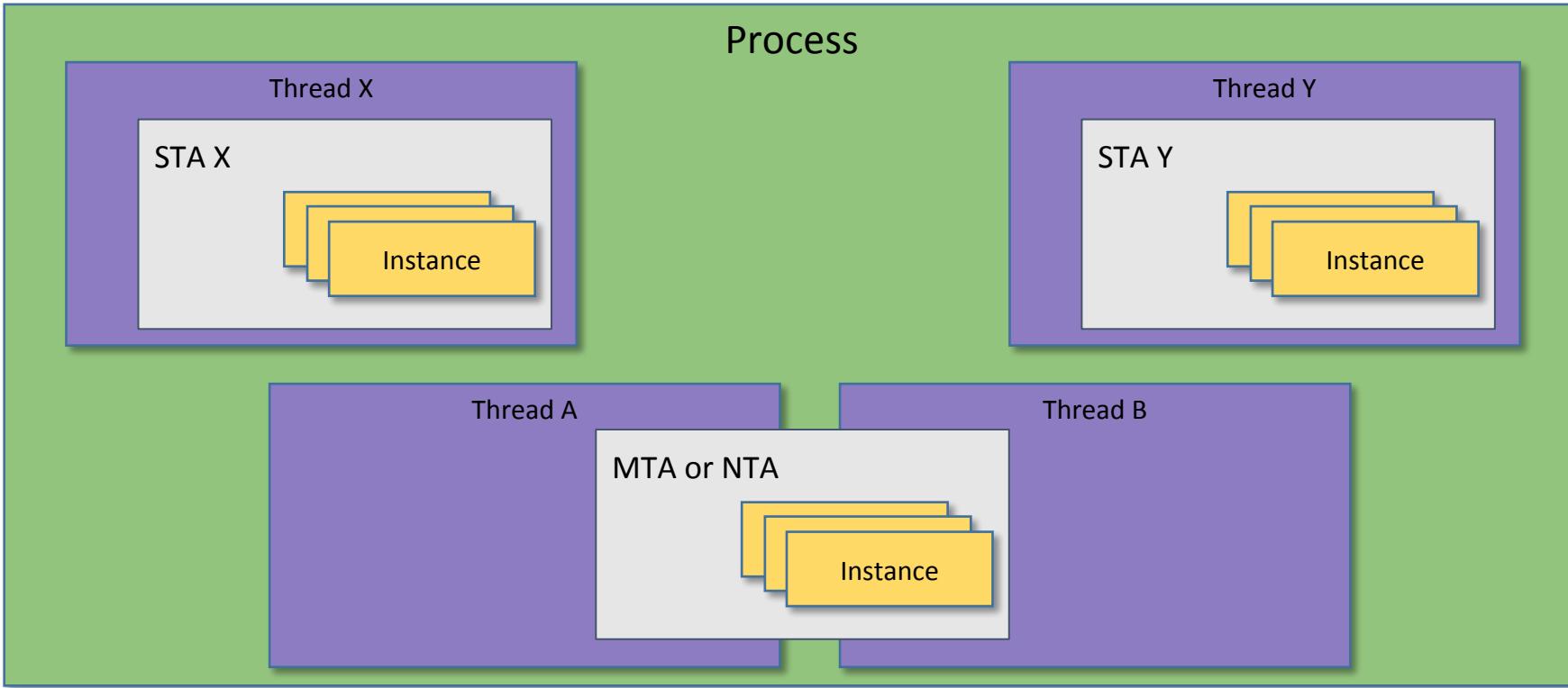
In-Process Server



- DLL server filename specified in *InProcServer32* key.
- DLL loaded into process and class factory created by calling exported method:

```
HRESULT DllGetClassObject (REFCLSID rclsid,  
                          REFIID riid,  
                          LPVOID *ppv);
```

COM Apartments

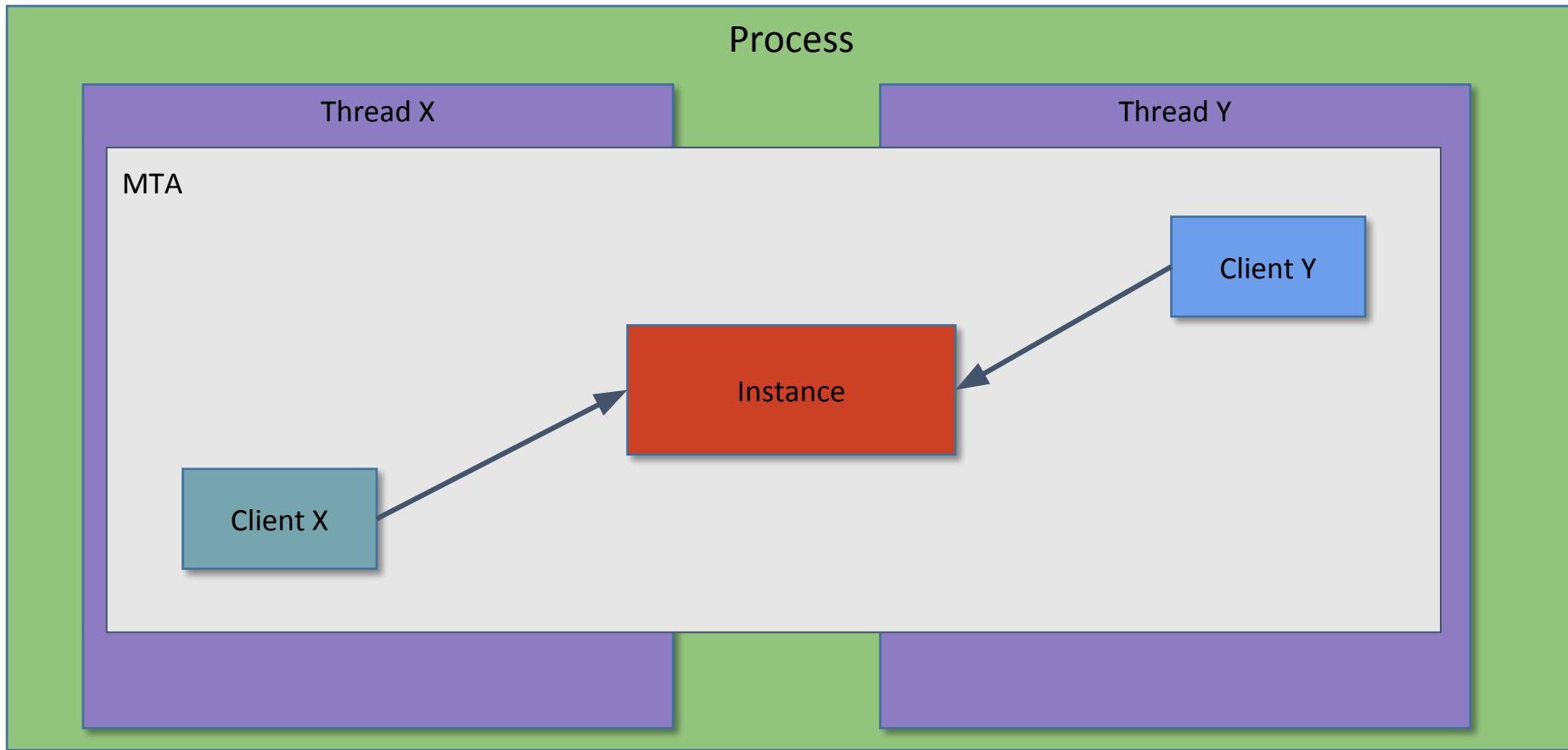


```
HRESULT CoInitializeEx(
```

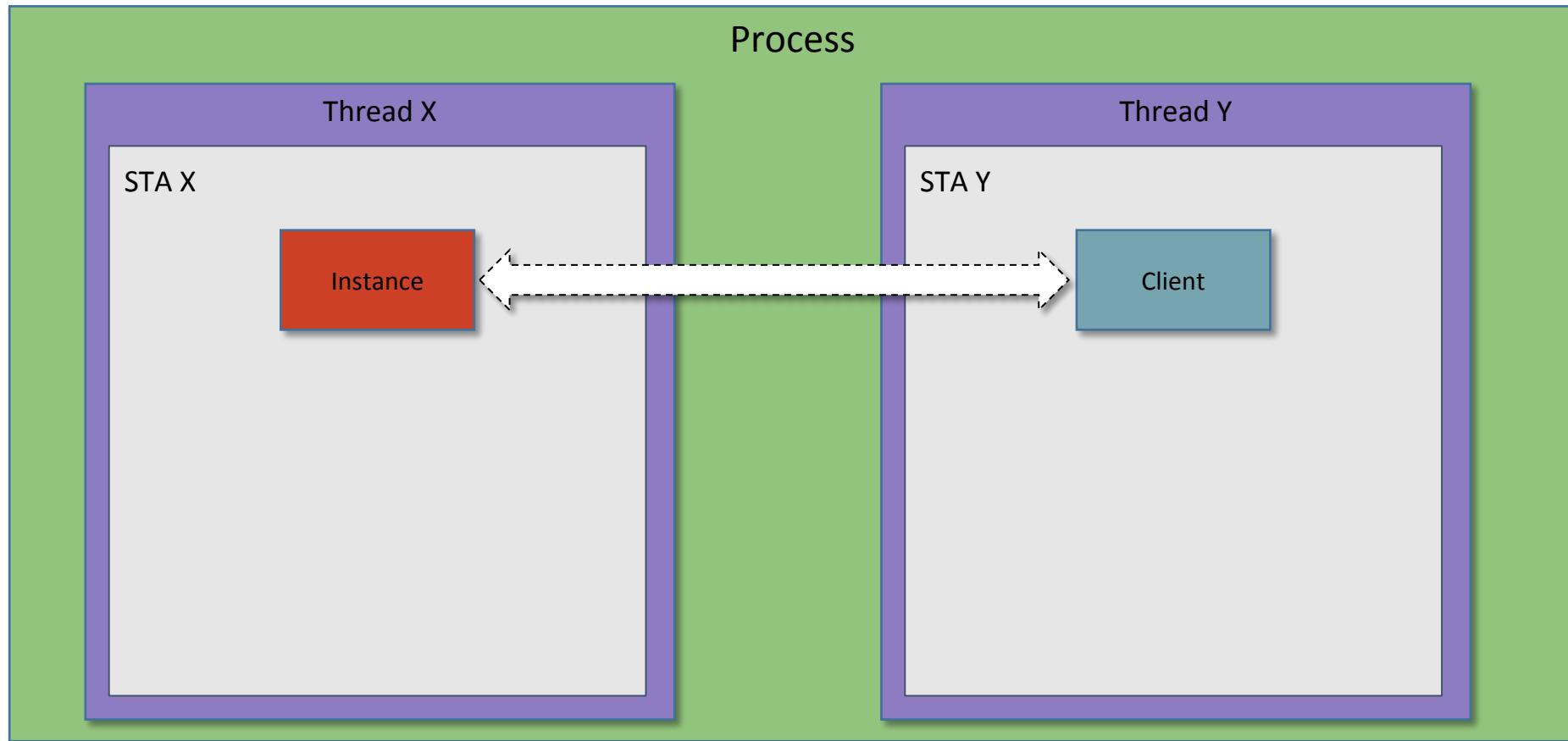
LPVOID pvReserved,	COINIT_APARTMENTTHREADED
DWORD dwCoInit	or
	COINIT_MULTITHREADED

```
) ;
```

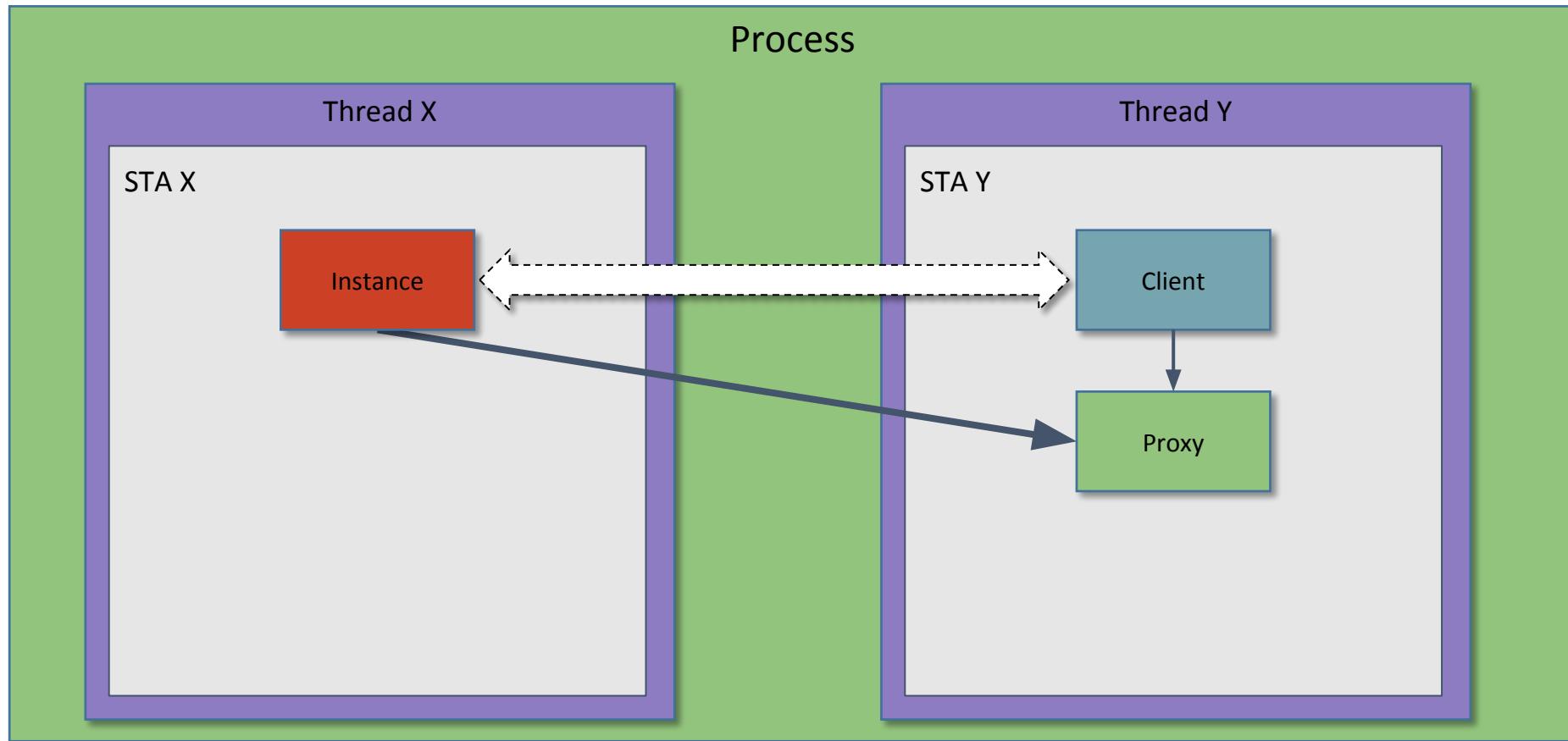
Multi-Threaded Apartments (MTA)



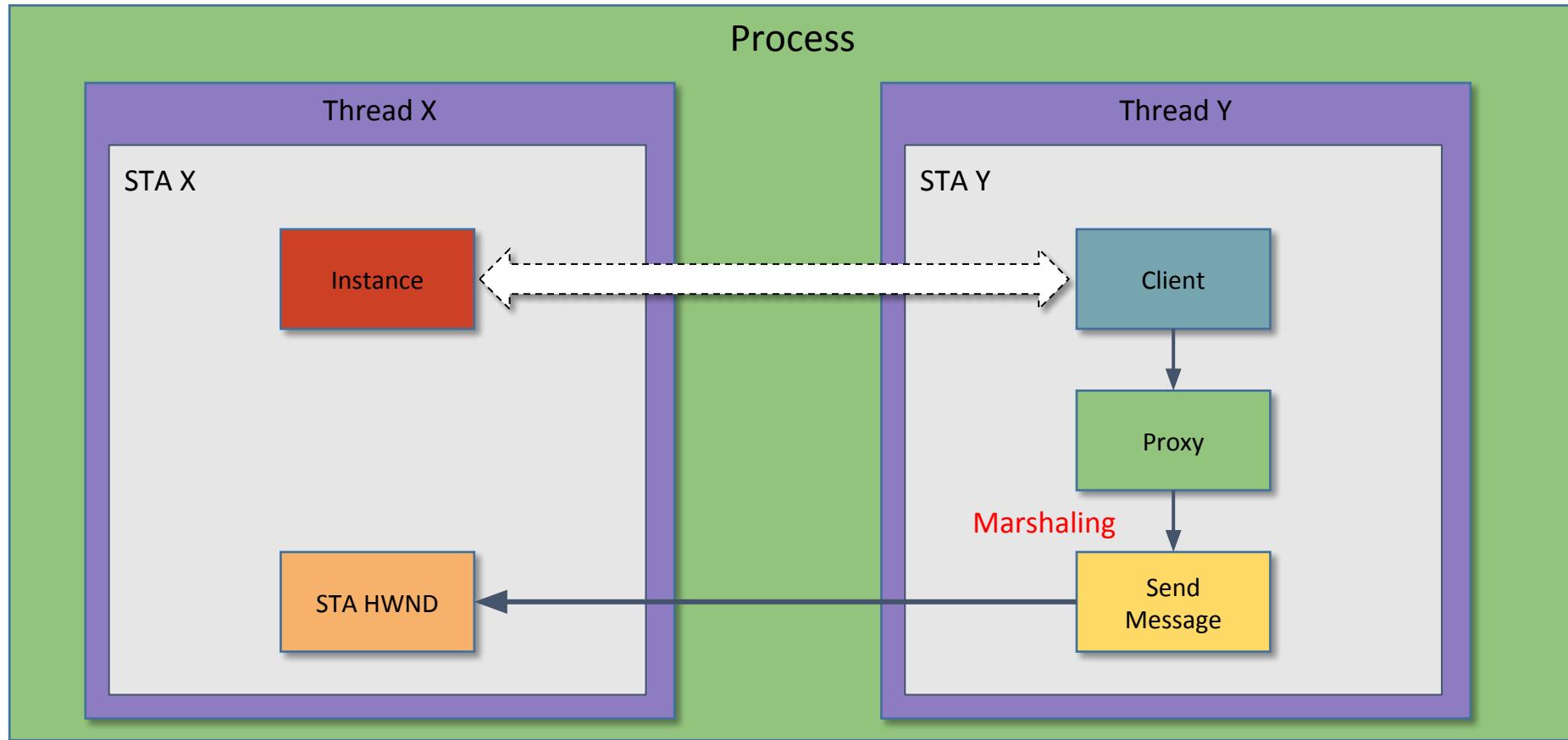
Single Threaded Apartments (STA)



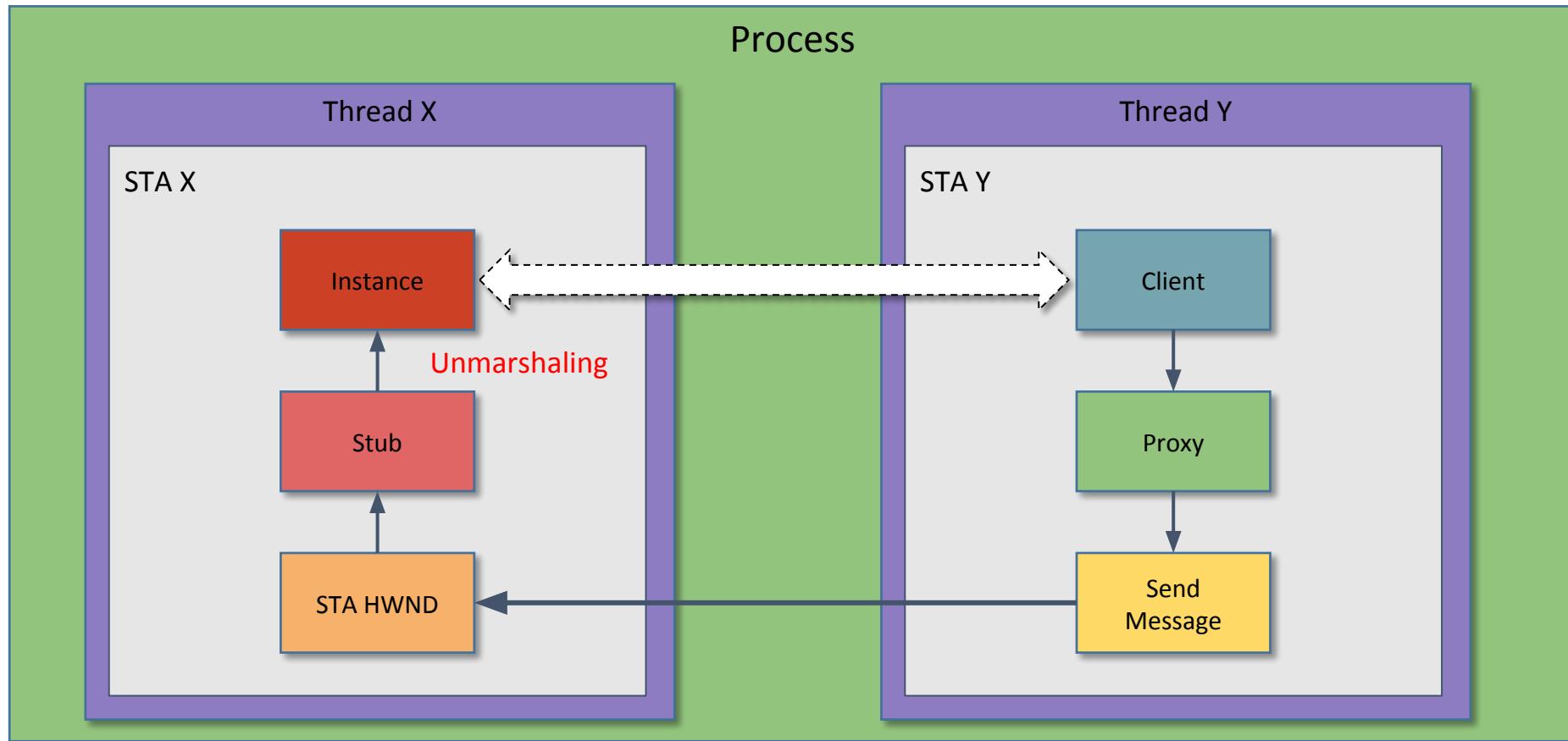
Single Threaded Apartments (STA)



Single Threaded Apartments (STA)



Single Threaded Apartments (STA)

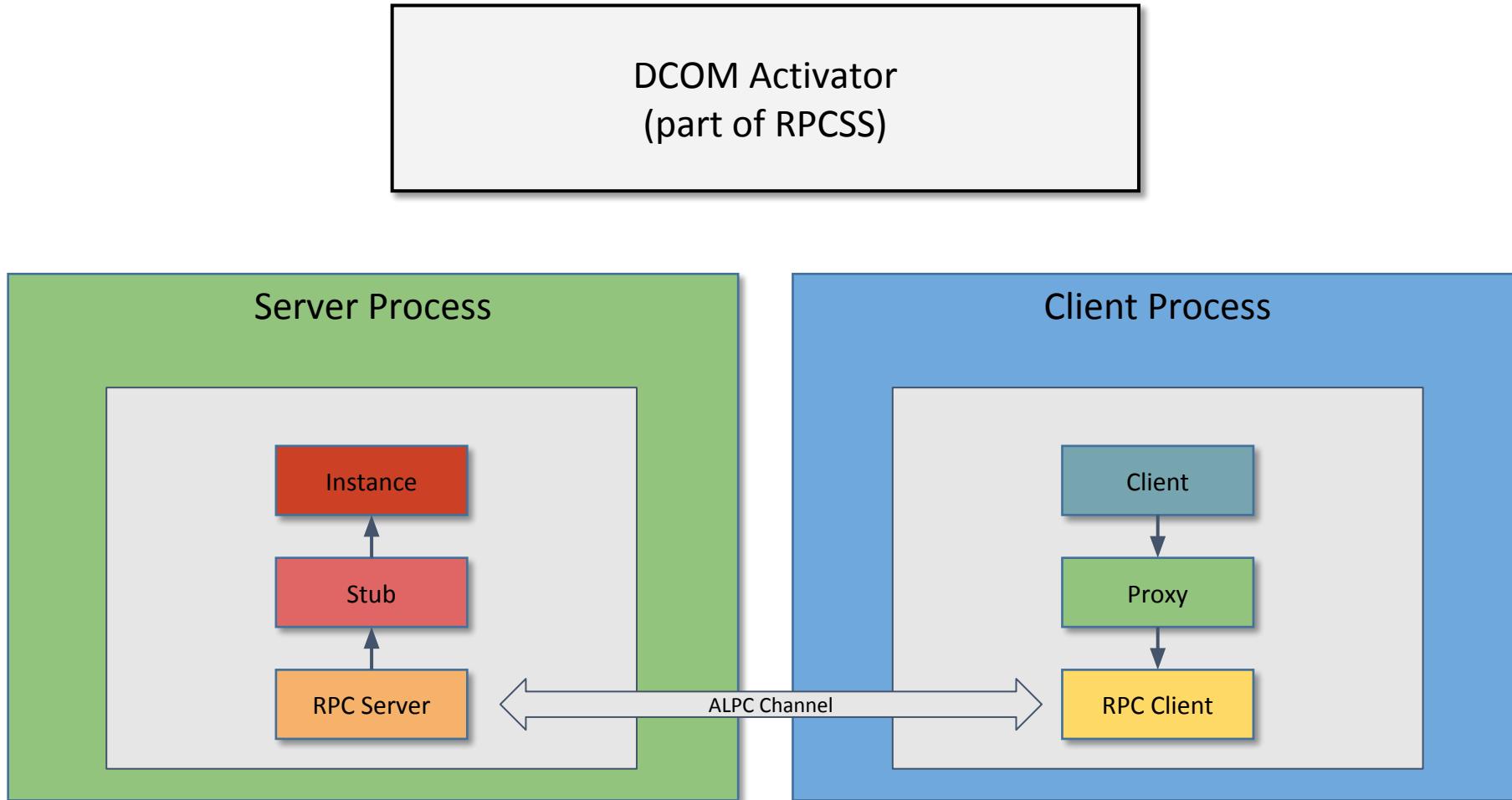


The Mystery Window

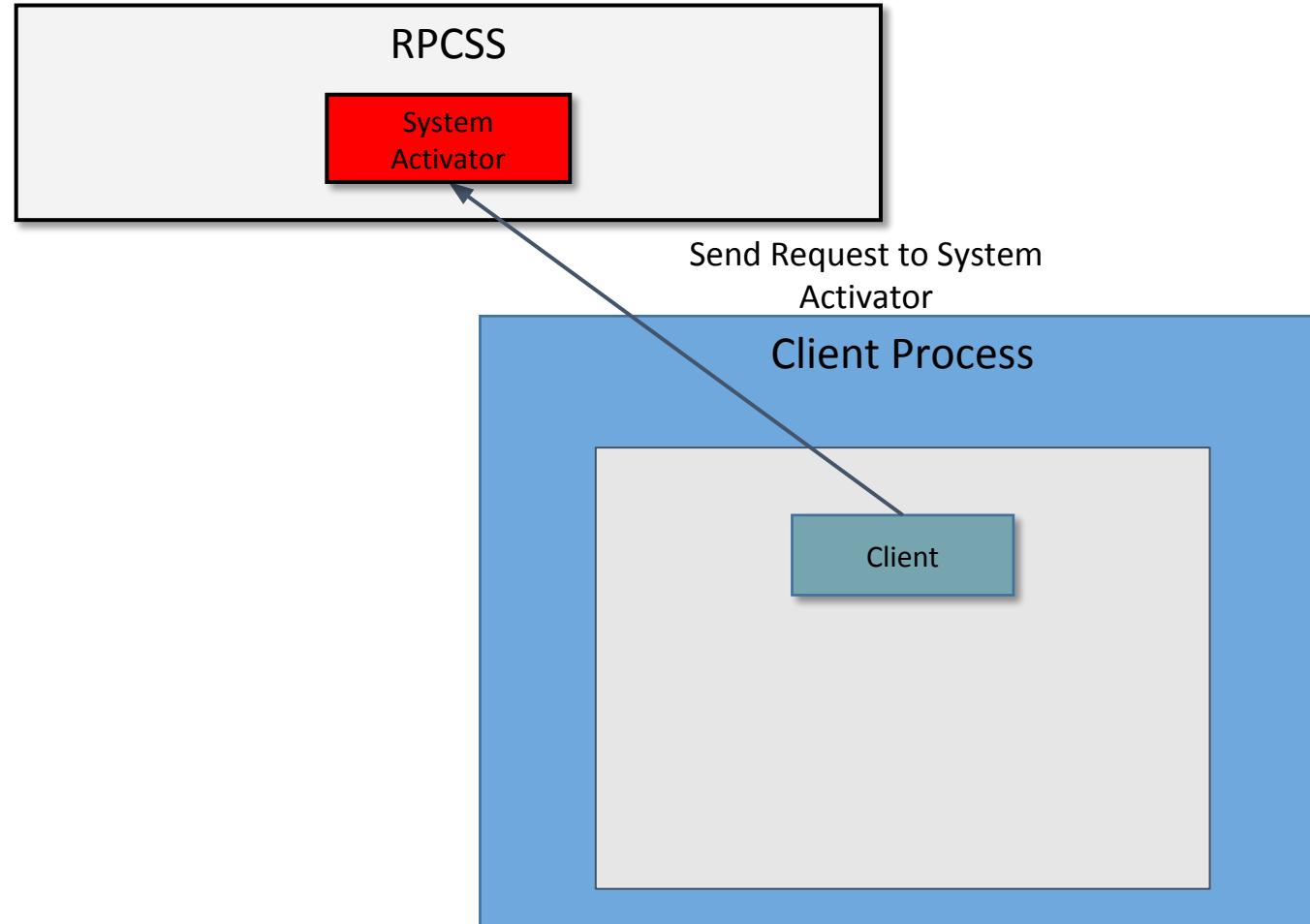
```
HWND hwnd = FindWindowEx(HWND_MESSAGE, NULL, NULL, NULL);
while (hwnd) {
    WCHAR name[256] = {};
    if (GetWindowText(hwnd, name, _countof(name))
        && _wcsnicmp(name, L"ole", 3) == 0) {
        DWORD pid = 0;
        DWORD tid = GetWindowThreadProcessId(hwnd, &pid);
        printf("%p %5d %5d %ls\n", hwnd, pid, tid, name);
    }
    hwnd = GetNextWindow(hwnd, GW_HWNDNEXT);
}
```

HWND	PID	TID	NAME
002906E4	21300	13416	OleMainThreadWndName
00510942	4316	8464	OLEChannelWnd
005F0A4A	4316	21232	OLEChannelWnd
00210D3E	4316	6028	OLEChannelWnd
001A048E	5880	19768	OLEChannelWnd
005C077E	5880	16680	OLEChannelWnd
004E08A2	5880	12464	OLEChannelWnd
000702E4	19600	20756	OleMainThreadWndName
00530832	5880	12636	OleMainThreadWndName
004E07EE	10952	18772	OLEChannelWnd
00110D58	10952	19568	OleMainThreadWndName
003D0302	5404	19032	OLEChannelWnd
004B0B7A	2244	2416	OleMainThreadWndName
001E09D8	19828	19832	OleMainThreadWndName

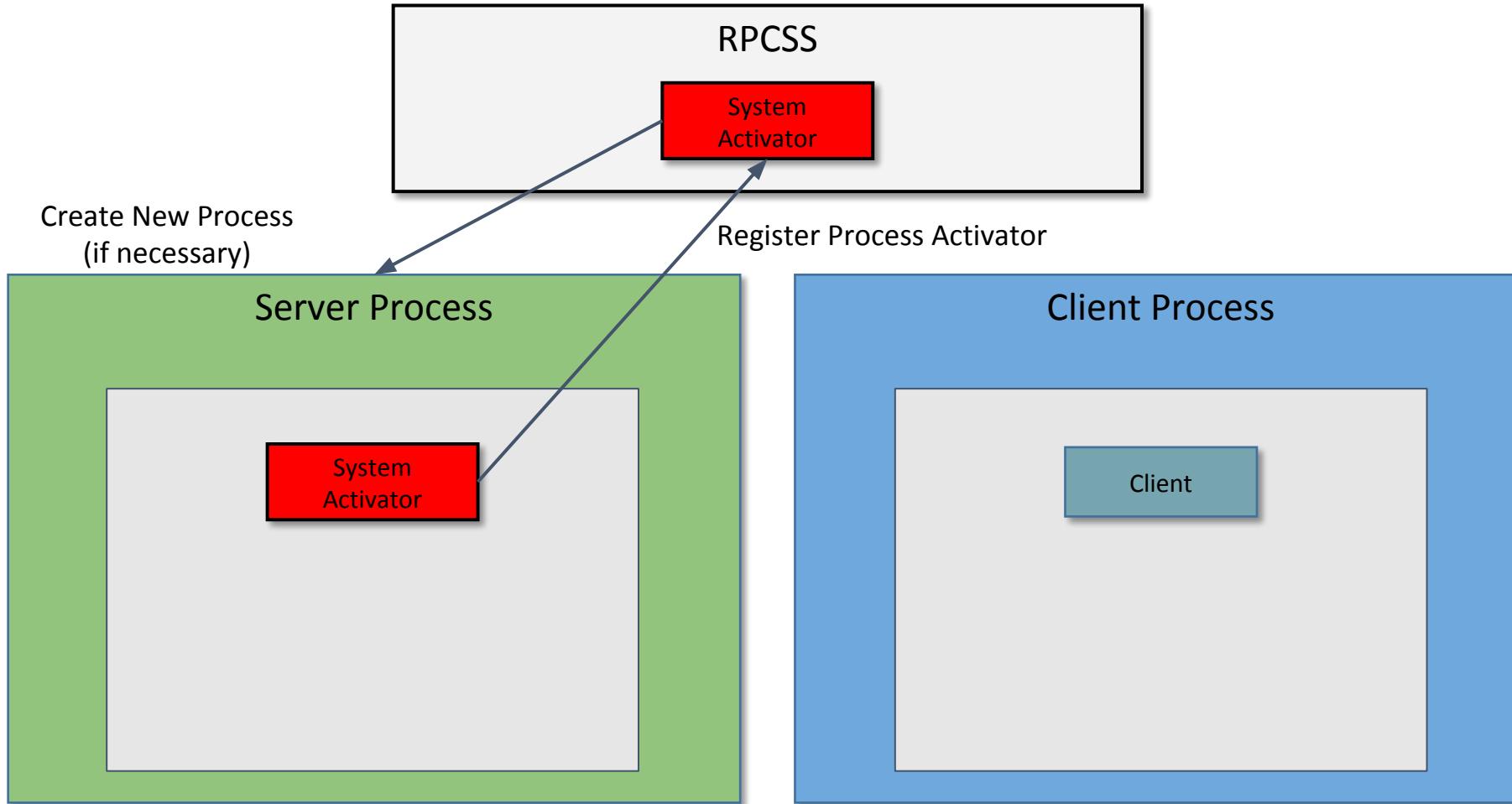
Local Server Activation



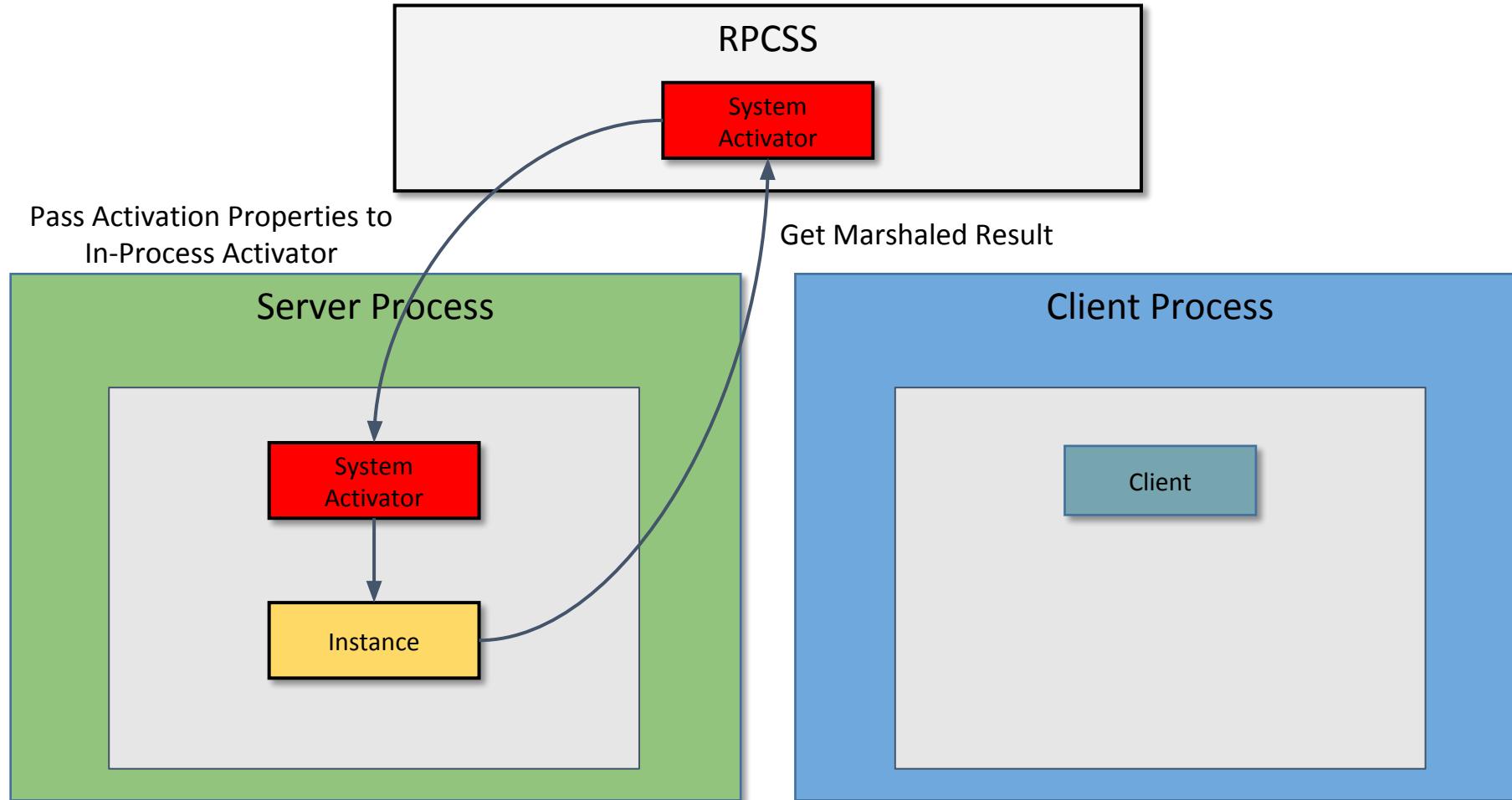
Local Server Activation



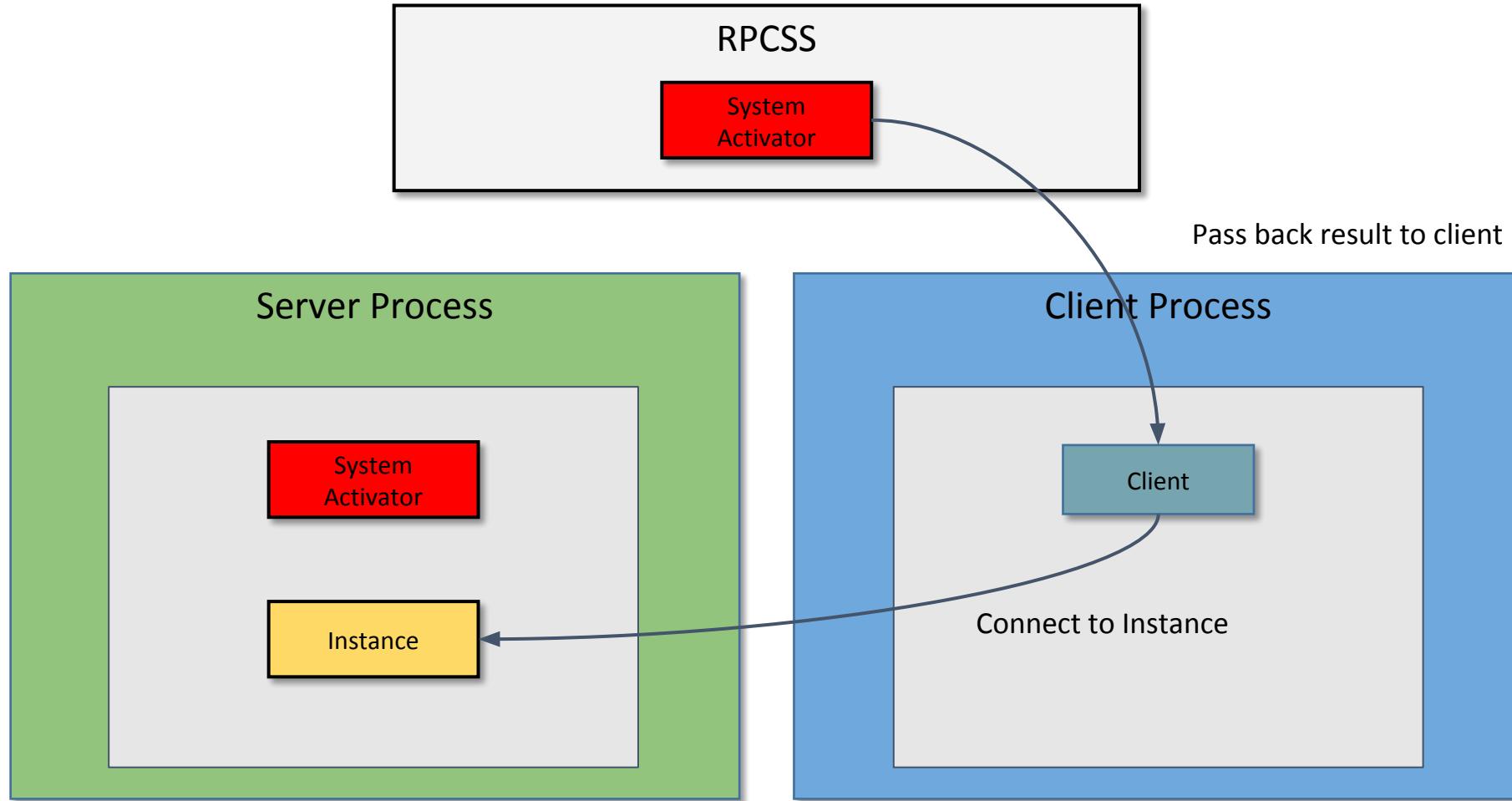
Local Server Activation



Local Server Activation



Local Server Activation



System Activator

```
DEFINE_GUID(IID_ISystemActivator,
            "000001a0-0000-0000-c000-000000000046")
struct ISystemActivator : public IUnknown {
    HRESULT GetClassObject(
        IActivationPropertiesIn *pActProperties,
        IActivationPropertiesOut **ppActProperties);
    HRESULT CreateInstance(
        IUnknown *pUnkOuter,
        IActivationPropertiesIn *pActProperties,
        IActivationPropertiesOut **ppActProperties);
};
```

Activation Properties In



```
struct CustomHeader {  
    DWORD totalSize;  
    DWORD headerSize;  
    DWORD dwReserved;  
    DWORD destCtx;  
    DWORD cIfs;  
    CLSID classInfoClsid;  
    CLSID *pclsid;  
    DWORD *pSizes;  
    CustomOpaqueData *opaqueData;  
};
```

List of GUIDs and Sizes of
following Property Blobs

```
struct InstantiationInfoData {  
    CLSID classId;  
    DWORD classCtx;  
    DWORD actvflags;  
    long fIsSurrogate;  
    DWORD cIID;  
    DWORD instFlag;  
    IID *pIID;  
    DWORD thisSize;  
    COMVERSION clientCOMversion;  
};
```

CLSID to
create.

List of IIDs to
query for.

```
enum ACTIVATION_FLAGS {  
    ACTVFLAGS_DISABLE_AAA,  
    ACTVFLAGS_ACTIVATE_32_BIT_SERVER,  
    ACTVFLAGS_ACTIVATE_64_BIT_SERVER,  
    ACTVFLAGS_NO_FAILURE_LOG,  
    ACTVFLAGS_WINRT_LOCAL_SERVER,  
    ACTVFLAGS_WINRT_PER_USER_OK,  
    ACTVFLAGS_APPCONTAINER,  
};
```

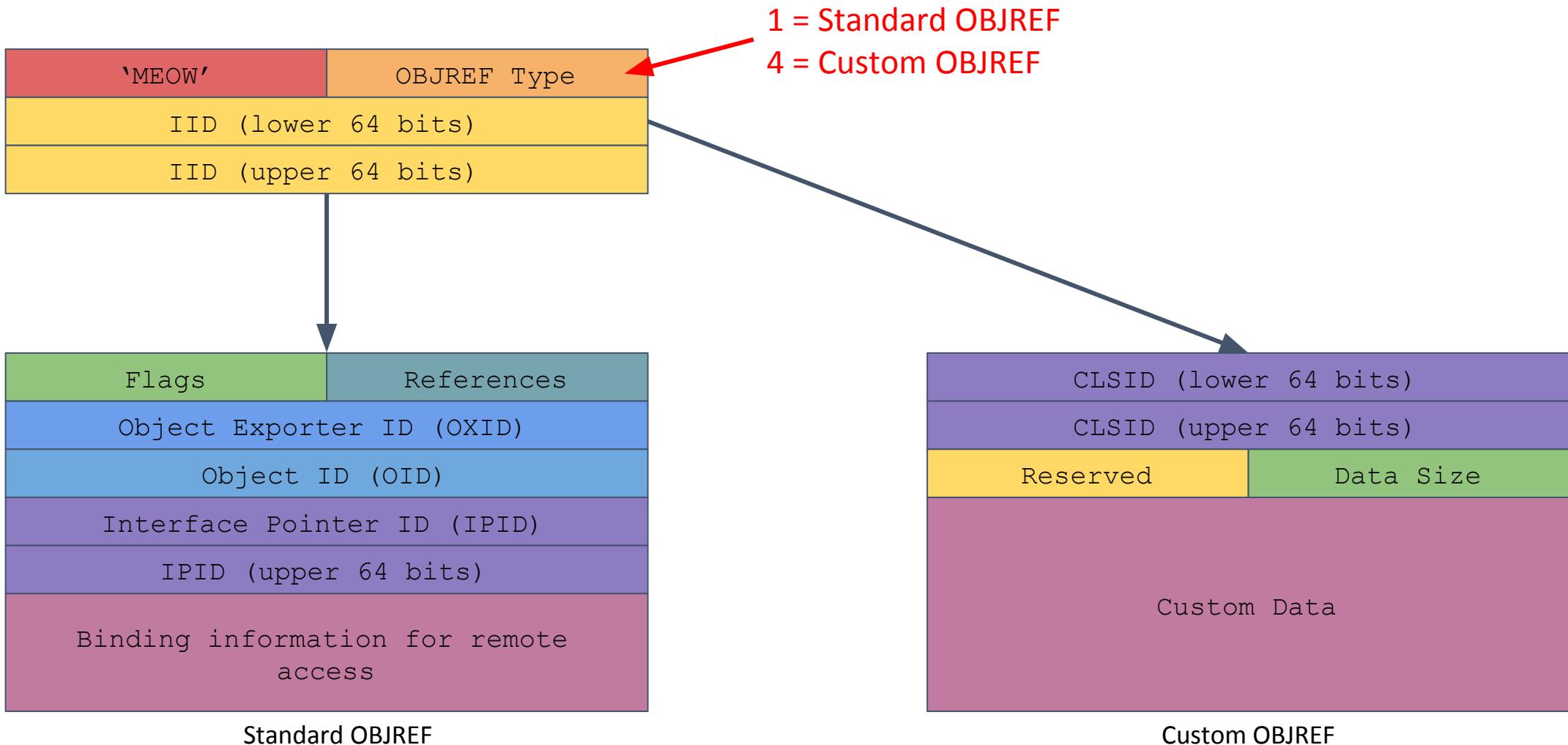
Activation Properties Out

```
struct PropsOutInfo {
    DWORD cIfs;
    IID *piid;
    HRESULT *phresults;
    MInterfacePointer **ppIntfData;
};
```

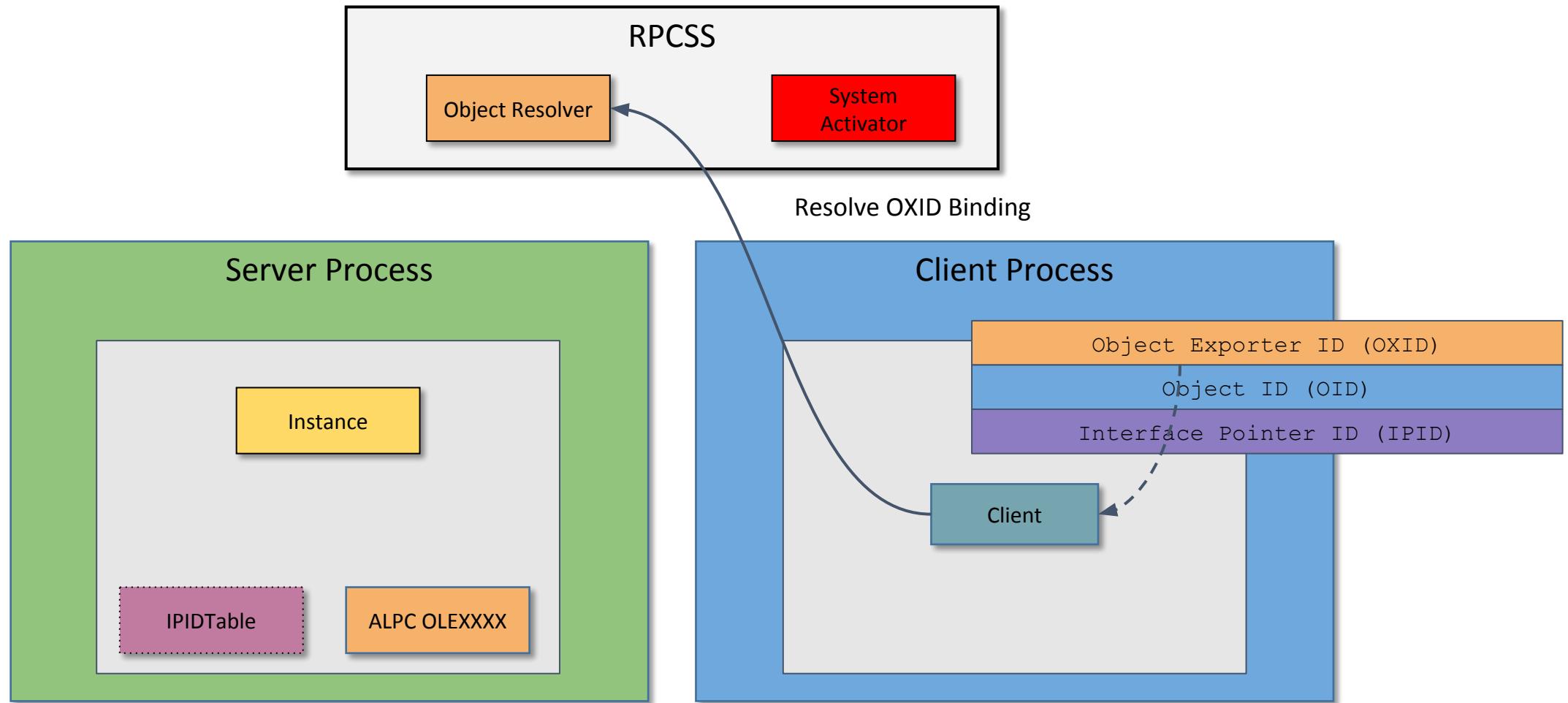
```
struct MInterfacePointer {
    unsigned long ulCntData;
    byte abData[];
};
```

00000000	4D 45 4F 57 01 00 00 00 A0 01 00 00 00 00 00 00	MEOW....
00000010	C0 00 00 00 00 00 00 00 46 00 00 00 00 01 00 00 00	À.....F.....
00000020	19 14 A0 F8 BF 3D 72 6A D5 41 68 BD 84 C3 08 E7	... øj=rjÖAhñ.Ã.ç
00000030	03 5C 00 00 38 16 E4 46 37 53 80 9C 87 A9 08 45	.\\..8.äF7S...@.E
00000040	00 00 00 00

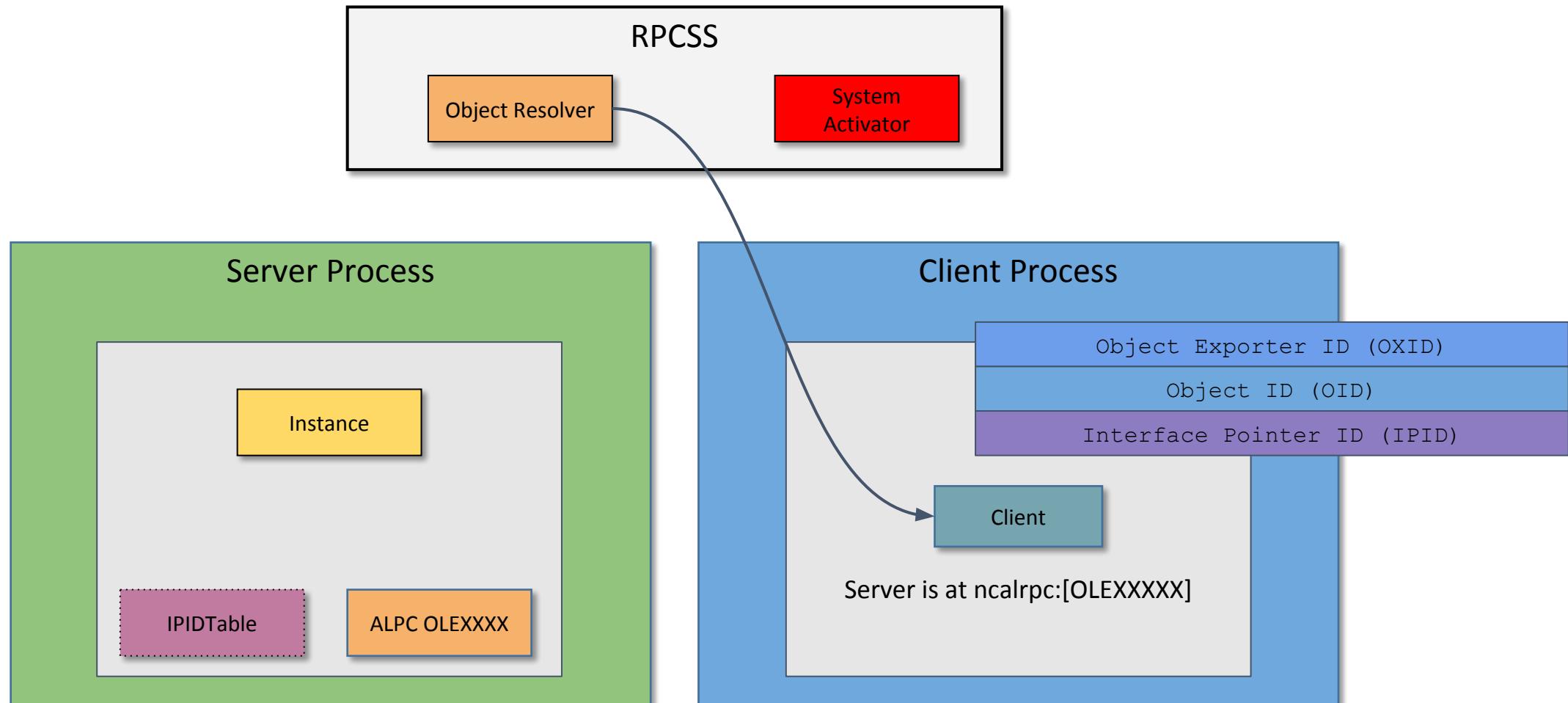
It's like Marshaling Cats



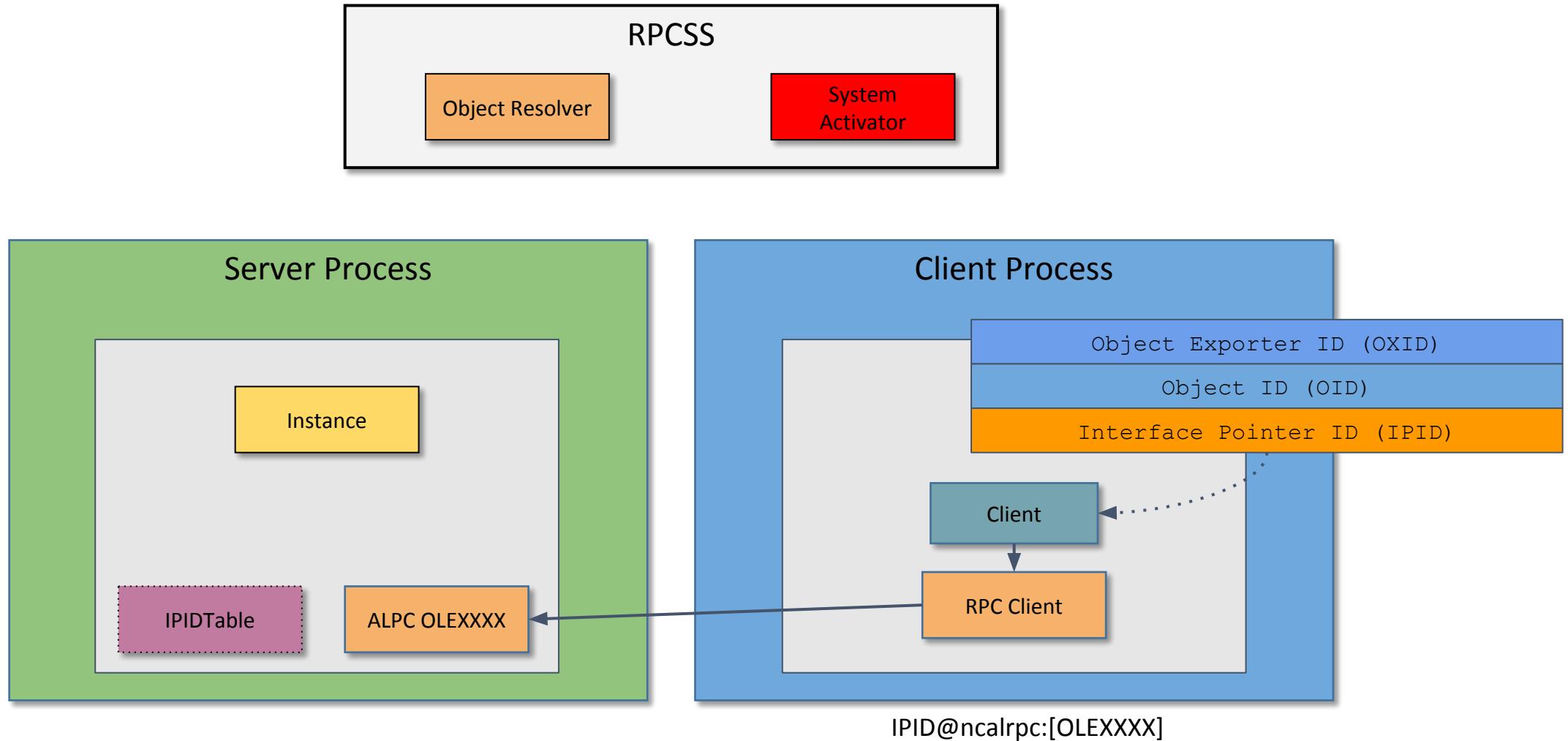
Standard Unmarshaling



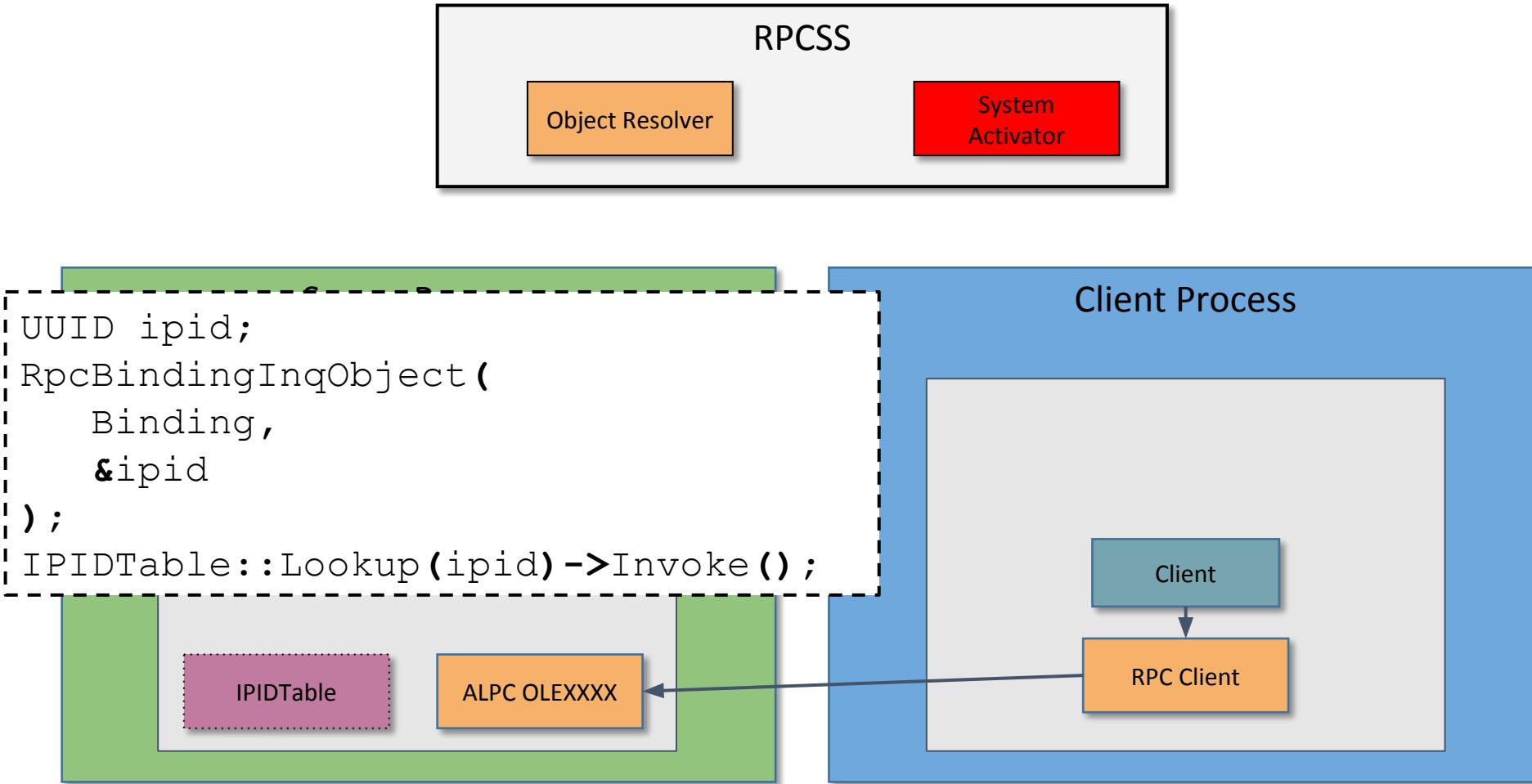
Standard Unmarshaling



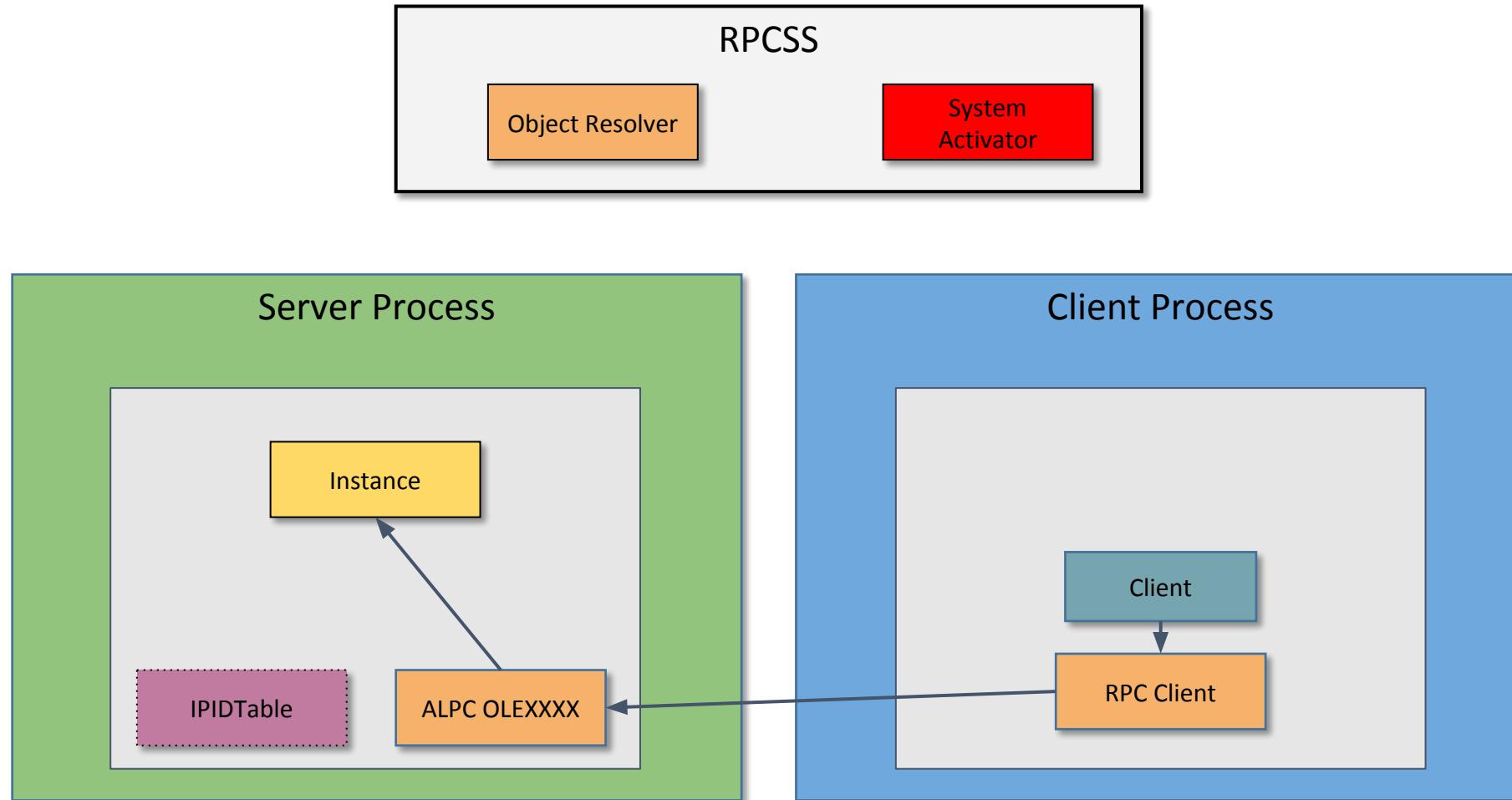
Standard Unmarshaling



Standard Unmarshaling



Standard Unmarshaling



Interface Proxies and Stubs

The image displays two windows illustrating interface proxies and stubs:

Registry Editor (Left Window): Shows the Windows Registry structure under `Computer\HKEY_CLASSES_ROOT\Interface`. A red arrow points from the `ProxyStubClsid32` key to the `IClaimedMagneticStripeReader` interface in the **OleViewDotNet** window.

OleViewDotNet 64bit (Right Window): Shows the supported interfaces for a specific CLSID (`e1ba88ba-7a83-421a-a05d-71...`). The `IClaimedMagneticStripeReader` interface is highlighted in blue.

Supported Interfaces:

- IMagneticStripeReaderBankCardDataReceivedEventArgs
- IPosPrinterStatusUpdatedEventArgs
- ICashDrawerStatusUpdatedEventArgs
- ITypedEventHandler_2_Windows__CDevices__CPointOfService__CClaimedPosP...
- IAsyncOperationCompletedHandler_1_Windows__CDevices__CPointOfService__...
- IBarcodeScannerStatusUpdatedEventArgs
- IJournalPrinterCapabilities
- IAsyncOperation_1_Windows__CDevices__CPointOfService__CClaimedMagnetic...
- IBarcodeScannerDataReceivedEventArgs
- IAsyncOperation_1_Windows__CDevices__CPointOfService__CCashDrawer
- IClaimedMagneticStripeReader** (highlighted)
- IClaimedBarcodeScanner
- ITypedEventHandler_2_Windows__CDevices__CPointOfService__CClaimedBarc...

Interface Proxy-Stub Factory

```
DEFINE_GUID(IID_IPSFactoryBuffer,
    "D5F569D0-593B-101A-B569-08002B2DBF7A")
struct IPSFactoryBuffer : public IUnknown
{
    HRESULT CreateProxy(
        IUnknown *pUnkOuter,
        REFIID riid,
        IRpcProxyBuffer **ppProxy,
        void **ppv);

    HRESULT CreateStub(
        REFIID riid,
        IUnknown *pUnkServer,
        IRpcStubBuffer **ppStub);
};
```

IRemUnknown

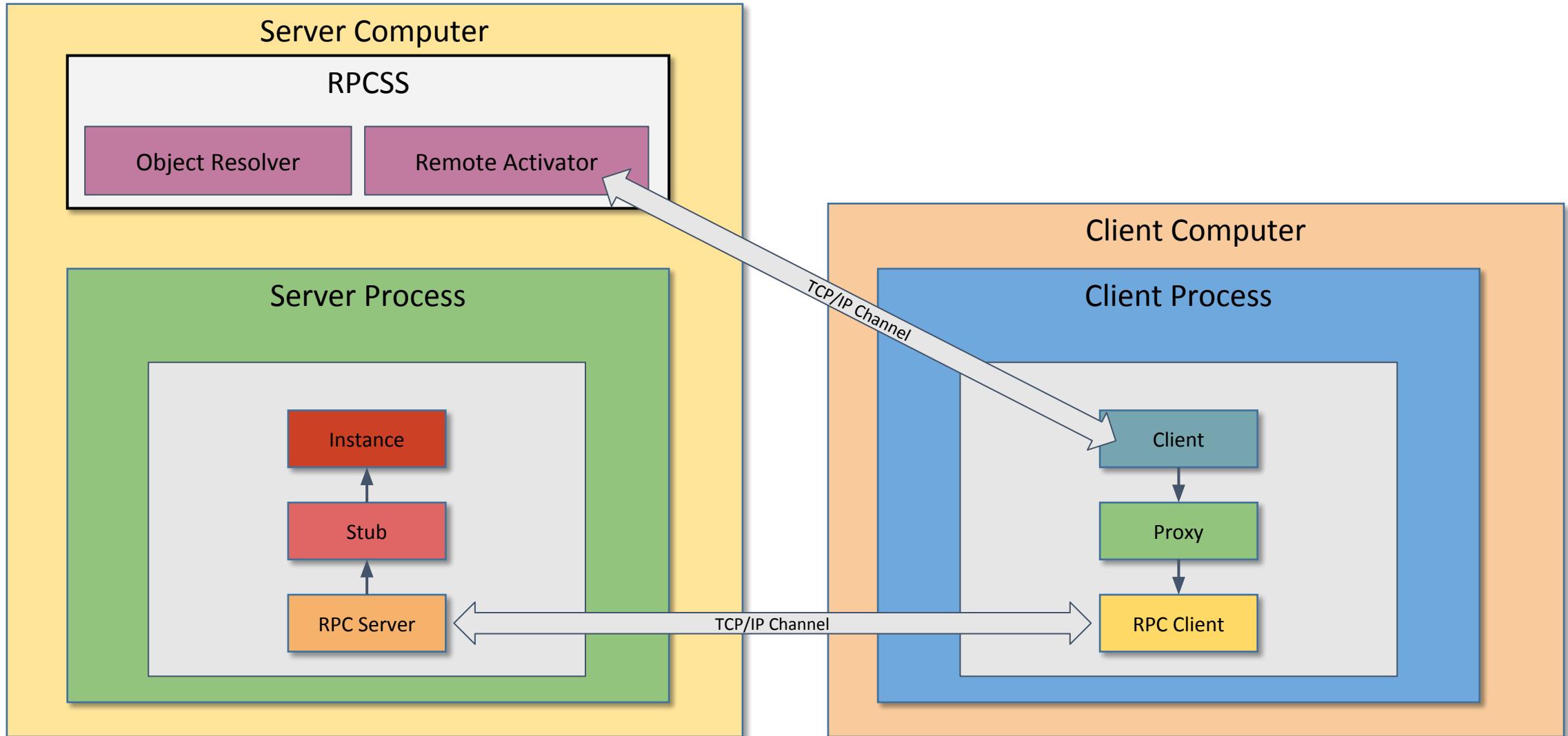
```
struct customREMOTE_REPLY_SCM_INFO {
    OXID Oxid;
    DUALSTRINGARRAY *pdsaOxidBindings;
    IPID ipidRemUnknown;
    DWORD authnHint;
    COMVERSION serverVersion;
};
```

```
DEFINE_GUID(IID_IRemUnknown,
            "00000131-0000-0000-C000-00000000046")
struct IRemUnknown : public IUnknown {
    HRESULT RemQueryInterface(
        REFIID ripid,
        unsigned long cRefs,
        unsigned short cIids,
        IID *iids,
        PREMQIRESULT *ppQIResults);

    HRESULT RemAddRef(
        unsigned short cInterfaceRefs,
        REMINTERFACEREF InterfaceRefs[],
        HRESULT *pResults);

    HRESULT RemRelease(
        unsigned short cInterfaceRefs,
        REMINTERFACEREF InterfaceRefs[]);
};
```

Remote Server



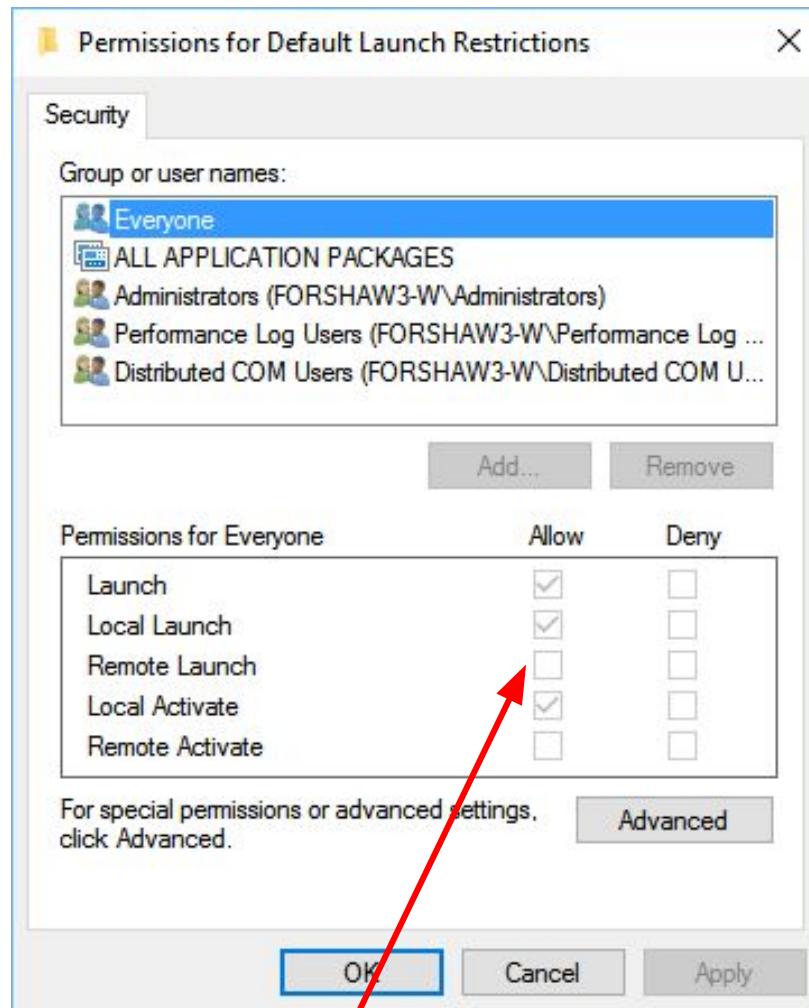
COSERVERINFO

```
struct COSERVERINFO {  
    DWORD dwReserved1;  
    LPWSTR pwszName;  
    COAUTHINFO *pAuthInfo;  
    DWORD dwReserved2;  
};
```

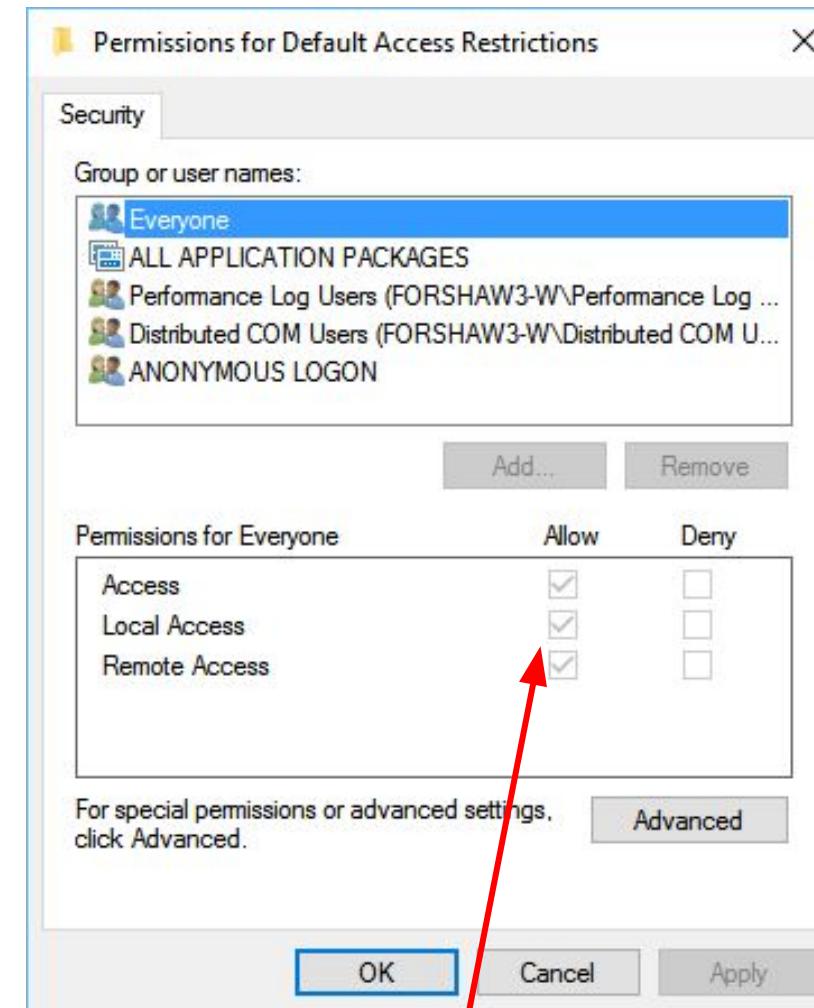
```
struct COAUTHINFO {  
    DWORD dwAuthnSvc;  
    DWORD dwAuthzSvc;  
    LPWSTR pwszServerPrincName;  
    DWORD dwAuthnLevel;  
    DWORD dwImpersonationLevel;  
    void* pAuthIdentityData;  
    DWORD dwCapabilities;  
};
```

Can specify better authentication level when connecting such as CALL or even better PKT_PRIVACY

COM Security Restrictions

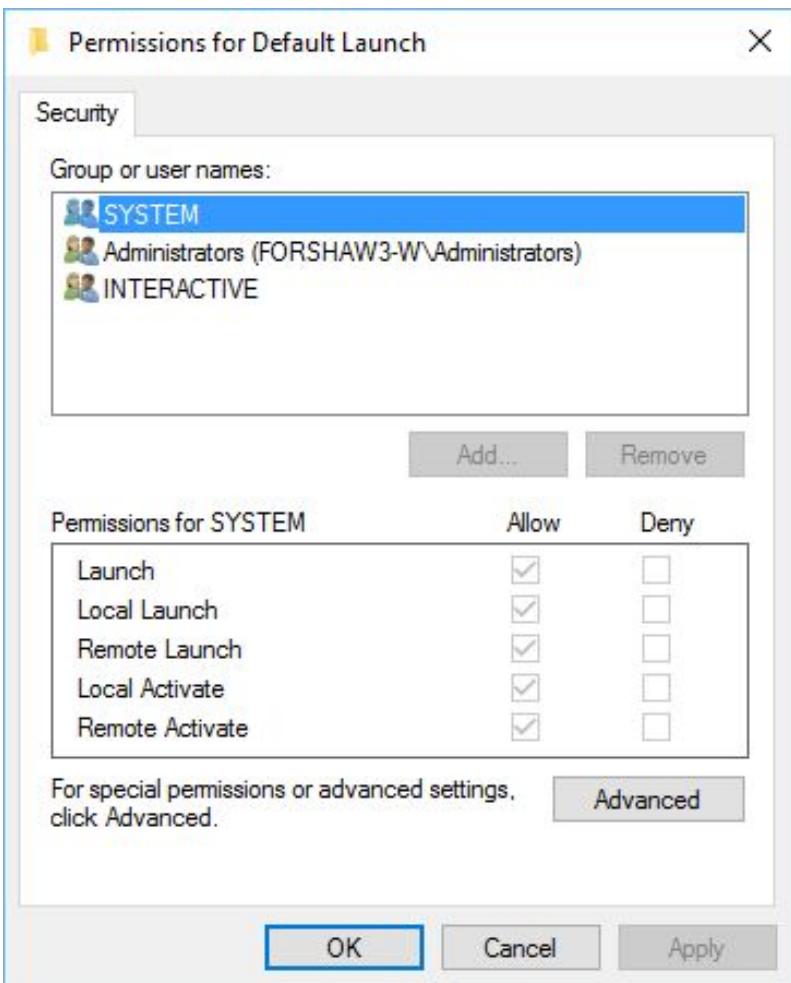


'Everyone' not allowed to launch a new object remotely

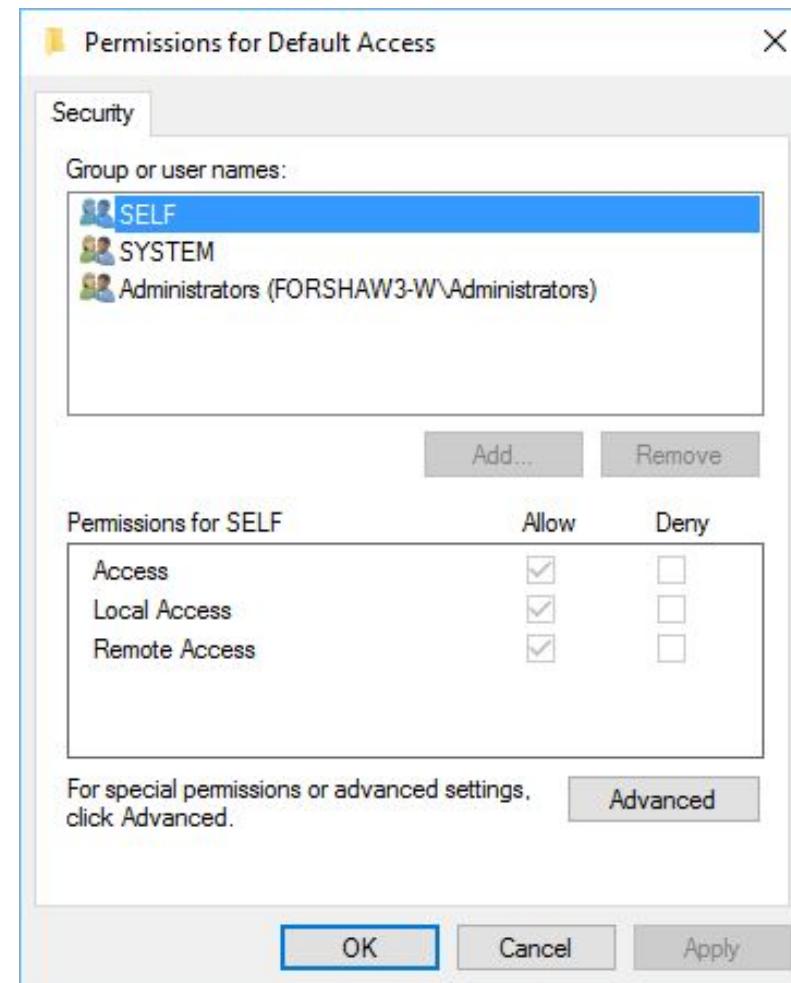


'Everyone' can access an existing object remotely⁷⁰

COM Security



Launch = Create a new instance of the server.
Activate = Create new object on existing server.
Enforced in RPCSS



Access = Call methods on existing objects.
Enforced in Server Process
SELF = Process Token User SID

Integrity Levels

The screenshot shows the OleViewDotNet 64bit application interface. The main window displays a list of registry entries under the tab 'AppIDs with IL'. A specific entry, 'AxInstSv', is selected and expanded, revealing its sub-entries including 'CIEAxIInstallerService Class'. A context menu is open over this entry, with the option 'Properties' highlighted. A secondary window titled 'Permissions for AxInstSv Launch' is displayed, showing the 'Security' tab. The 'Group or user names:' section lists 'Everyone' and 'S-1-15-3-4096'. Below this, the 'Advanced Security Settings for AxInstSv Launch' section is shown, with the 'Owner' set to 'Administrators (FORSHAW3-W\Administrators)'. A red box highlights the 'Integrity level: Low Mandatory Level' text. The 'Permissions' section at the bottom lists two entries: 'Allow Everyone Access Special' and 'Allow S-1-15-3-4096 Access Special'. The status bar at the bottom of the main window indicates 'Showing 55 of 55 entries'.

Security Through CoInitializeSecurity

```
HRESULT CoInitializeSecurity(  
    PSECURITY_DESCRIPTOR pSecDesc,  
    LONG cAuthSvc,  
    SOLE_AUTHENTICATION_SERVICE * asAuthSvc,  
    void * pReserved1,  
    DWORD dwAuthnLevel,  
    DWORD dwImpLevel,  
    void * pAuthList,  
    DWORD dwCapabilities,  
    void * pReserved3  
);
```

Optional SD:
NULL = No Access Security!

EOAC_APPID
: pSecDesc is a AppID GUID
EOAC_ACCESS_CONTROL
: pSecDesc is a pointer to an IAccessControl implementation
EOAC_NO_CUSTOM_MARSHAL
: Disables custom marshaling in the process.

Security Through AppID

The screenshot shows the OleViewDotNet 64bit application window with the title bar "OleViewDotNet 64bit". The menu bar includes File, Registry, Object, Security, and Help. A toolbar with icons for Open, Save, Copy, Paste, and others is visible. The main area displays the "Ifsvc Properties" dialog. The tabs at the top are CLSID, Supported Interfaces, AppID (which is selected), Service, and Type Library. The properties listed are:

Name:	Ifsvc
AppID:	020FB939-2C8B-4DB7-9E90-9527966E38E5
Run As:	N/A
Service:	Ifsvc
Flags:	None

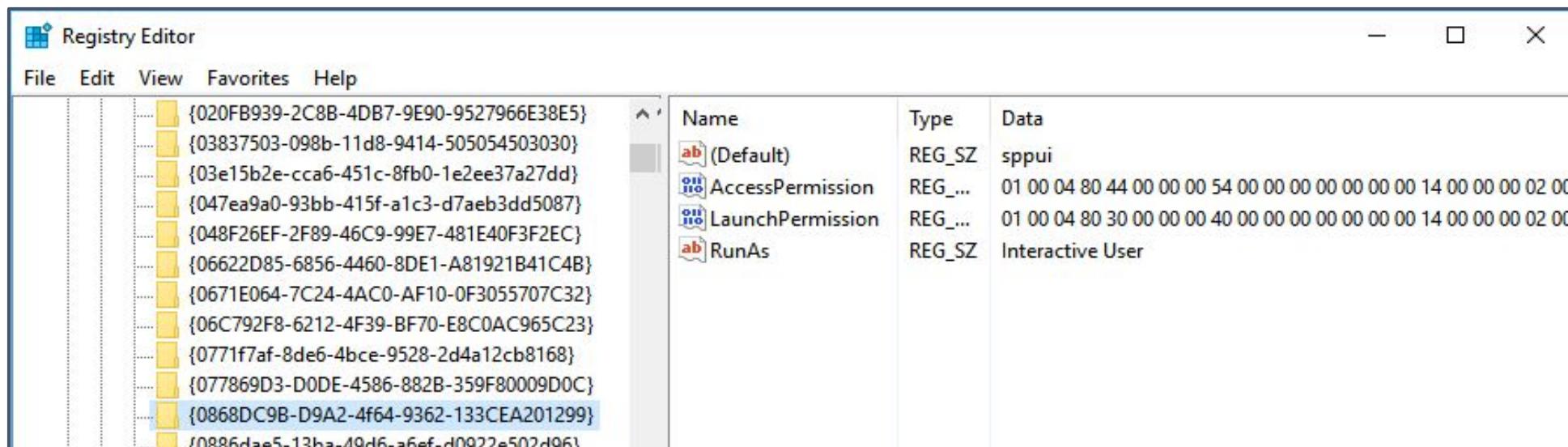
Below these fields are two sections with red boxes around them:

- Launch Permission:** O:BAG:BAD:(A;;CCDCSW;;;SY)(A;;CCDCSW;;;WD)(A;;CCDCSW;;;BA)(A;;CCDCSW;;;IU)(A;;CCDCSW;;;AC)(A;;CCDCSW;;;LS)S:(ML;;NX;;LW) [View](#)
- Access Permission:** O:BAG:BAD:(A;;CCDCSW;;;SY)(A;;CCDCSW;;;WD)(A;;CCDCSW;;;BA)(A;;CCDCSW;;;IU)(A;;CCDCSW;;;AC)(A;;CCDCSW;;;LS)S:(ML;;NX;;LW) [View](#)

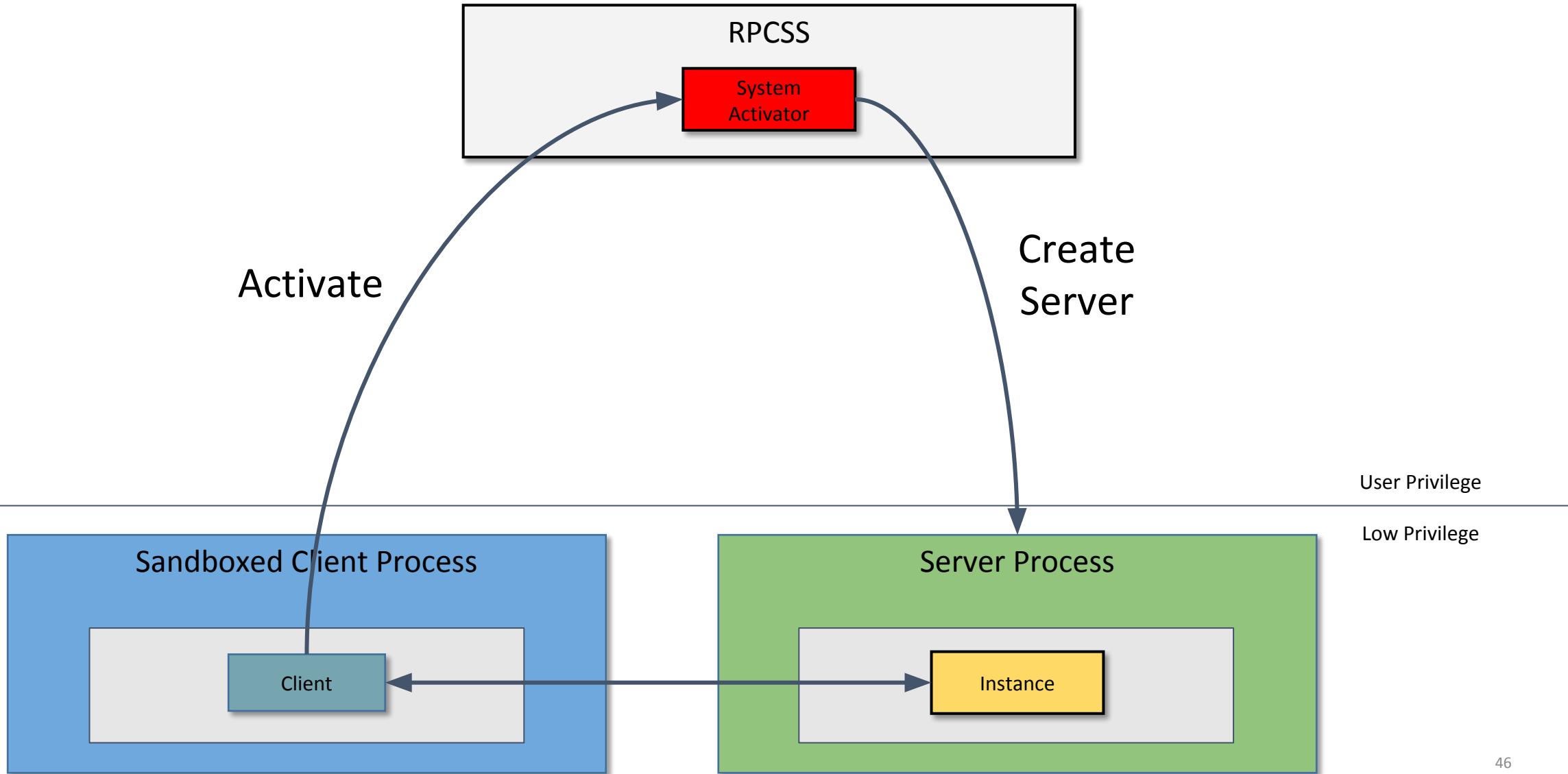
The "DLL Surrogate:" field shows "N/A". On the right side of the dialog, there is a vertical "Object Properties" panel with icons for Class, Object, and Type Library.

Application IDs

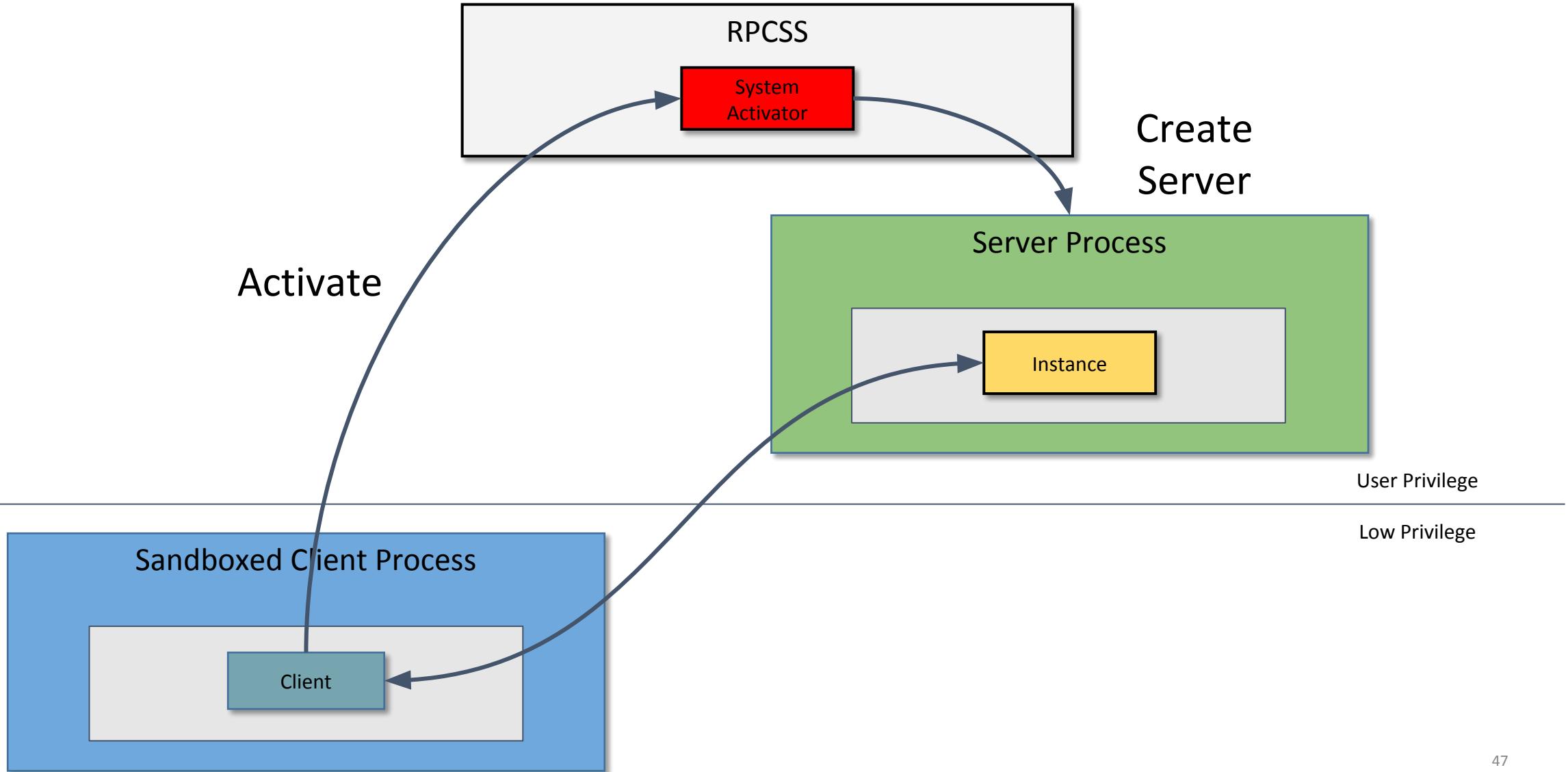
- AppIDs can configure more than just Launch/Access security
- Used to specify a number of features:
 - Allows you to specify a DLL Surrogate. This allows you to create in-process DLL servers as OOP local servers
 - Specify the object is hosted in a Windows service rather than a separate executable
 - Specify running as interactive user.



Activate as Activator (AAA)



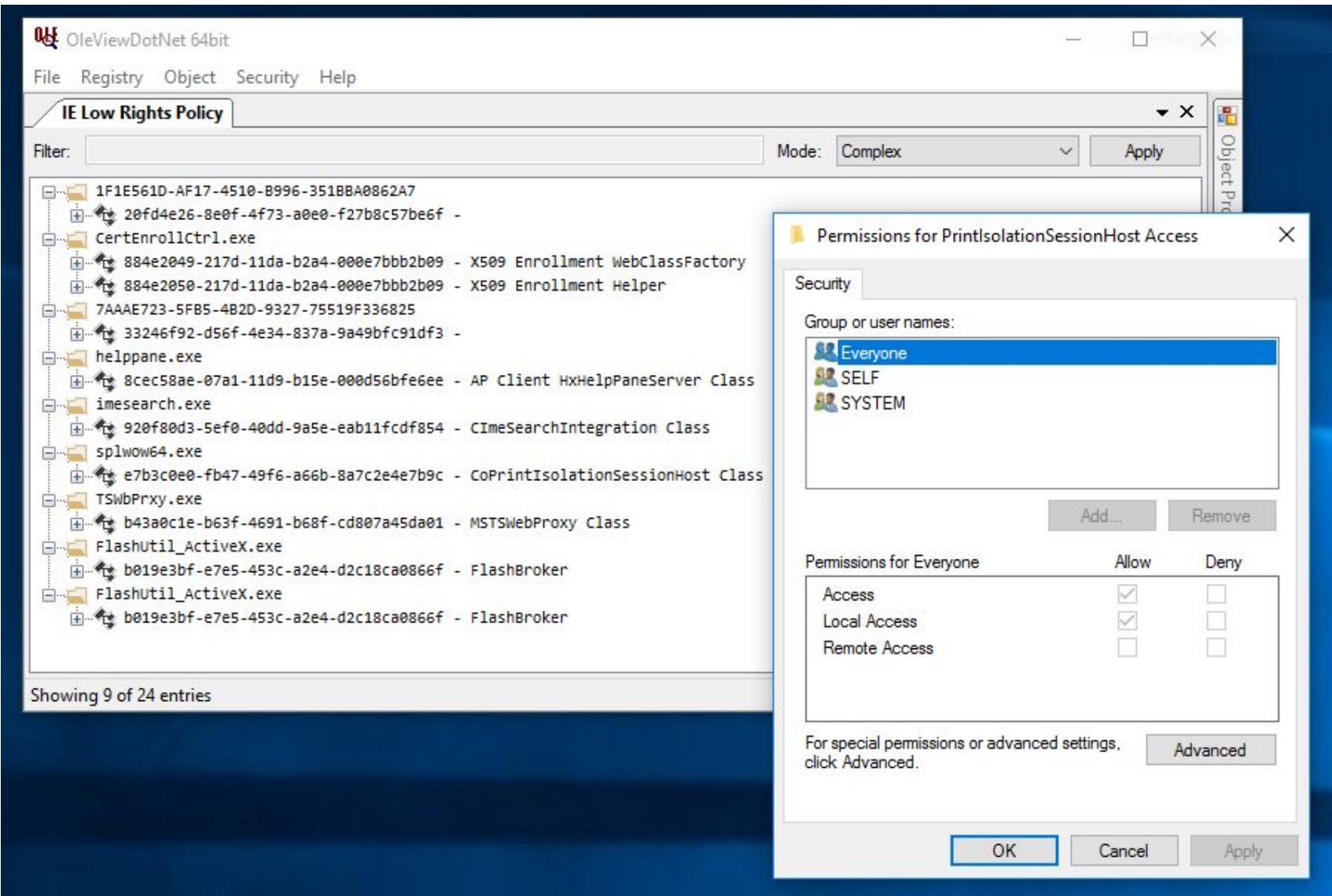
RunAs Interactive User



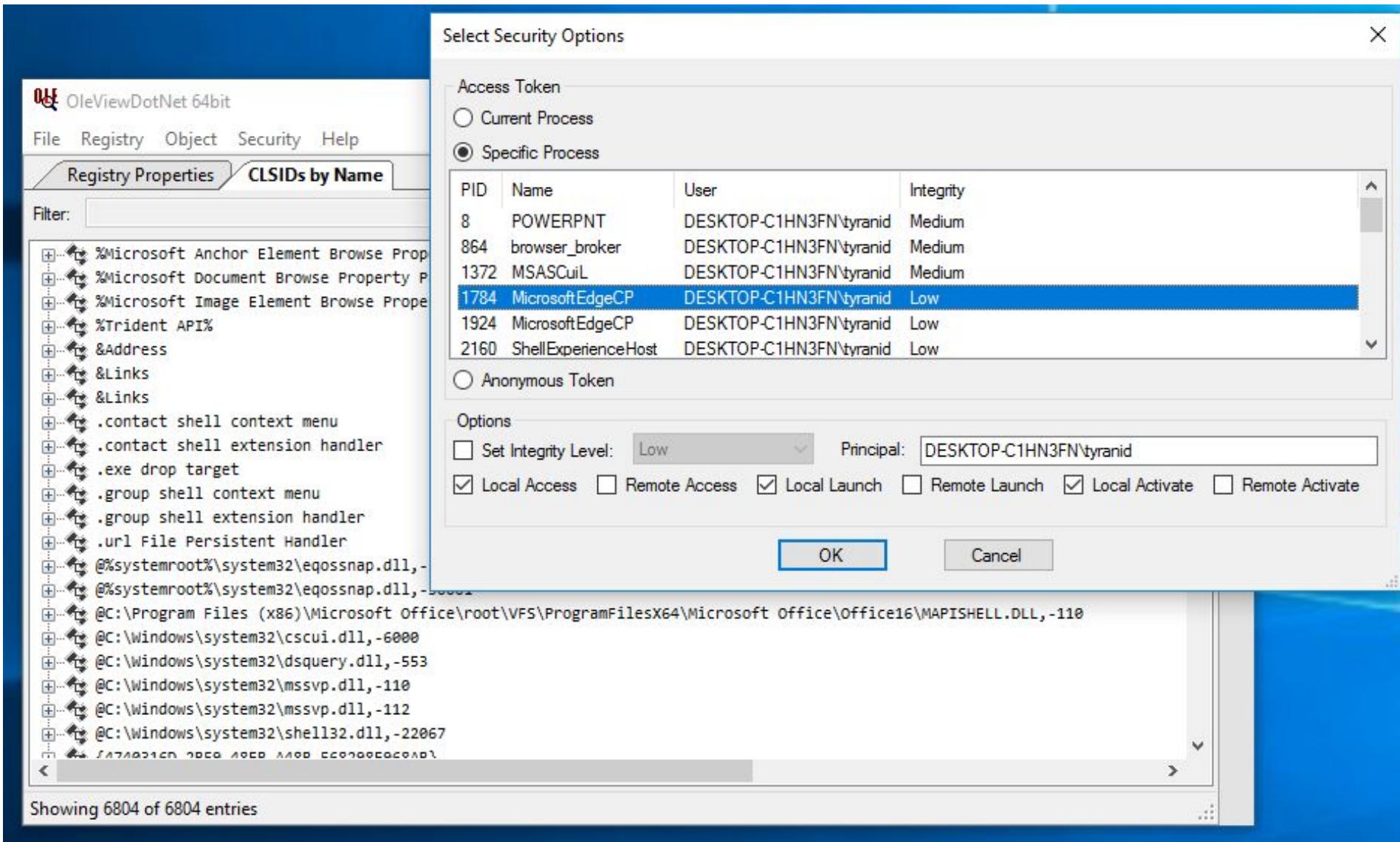


Enumerating Attack Surface and Reverse Engineering

Sandbox: Activation via Broker



Sandbox: Activation via RPCSS



Edge is Watching You PWN

OleViewDotNet 64bit

File Registry Object Security Help

AppIDs from LPAC

AppIDs from AC

Object Properties

Filter: Mode: Accessible Apply

Filter: Mode: Accessible Apply

Edge + LPAC
~20 CLSIDs

Edge + AC
~40 CLSIDs

51

42CBFAA7-A4A7-47BB-B422-BD10E9D02700	Diagnostics Hub Standard Collector Service
A463FCB9-6B1C-4E0D-A80B-A2CA7999E25D	SmartScreen
AA0B85DA-FDDF-4272-8D1D-FF9B966D75B0	CPrintTicket WoW Services
B0316D0C-DA2F-40E0-9F91-F600CAF042DC	69B1A7D7-C09E-40E9-A1DF-688007A2D9E4
	9A4B1918-0A2F-4422-890D-35B3F455999C
	A4FBCBC6-4BE5-4C3D-8AB5-88873357A23E
	BA6EE7D8-190D-423A-93CC-1270E6599195
	C658E5BD-817B-41C8-8FB6-5B2B386A40EA
	DE50C7BB-FAA7-4A7F-BA47-BF0EFCFE433D
	DF46CD07-4F86-42F0-8FA9-35C3CE55D77B
	MtfTransportServerDCOM
DataExchangeHost	DataExchange Host
editionupgradebroker	EditionUpgradeBroker
lfsvc	lfsvc

2A947841-0594-48CF-9C53-A08C95C22B55	
42CBFAA7-A4A7-47BB-B422-BD10E9D02700	
5E176815-9A63-4A69-810F-62E90D36612A	
9D73451F-6BFC-47C7-95FB-46598431BC19	
A463FCB9-6B1C-4E0D-A80B-A2CA7999E25D	
AA0B85DA-FDDF-4272-8D1D-FF9B966D75B0	
B0316D0C-DA2F-40E0-9F91-F600CAF042DC	
BrowserBrokerServer	
CE0E0BE8-CF56-4577-9577-34CC96AC087C	
CoreDpusSvr	
DataExchangeHost	
editionupgradebroker	
F1425A67-1545-44A2-AB59-8DF1020452D9	
F72671A9-012C-4725-9D2F-2A4D32D65169	
F8842F8E-DAFE-4B37-9D38-4E0714A61149	
InstallAgent	
InstallAgentUserBroker	
lfsvc	
Local Service Credential UI Broker	
OOBE Bio Enrollment	
PaymentsSvc	

In-process Reverse Engineering

The screenshot shows the OleViewDotNet 64bit application window. The title bar reads "OleViewDotNet 64bit". The menu bar includes File, Registry, Object, Security, and Help. The main window title is "Shockwave Flash Object Prop...". Below it are tabs for CLSID, Supported Interfaces, and Type Library, with "Supported Interfaces" selected. A sub-header "Interfaces:" is followed by a "Refresh" button. The main table lists ten interfaces with their names, IIDs, method counts, and VTable offsets. A red dashed box highlights the "VTable Offset" column, and a red arrow points from the text "VTable RVAs for interface" to this column. The "Object Properties" pane on the right is partially visible.

Name	IID	Methods	VTable Offset
ICanHandleException	C5598E60-B307-11D1-B27D-006008C3FBFB	4	Flash.ocx+0x134A748
IConnectionPointContainer	B196B284-BAB4-101A-B69C-00AA00341D07	3	Flash.ocx+0x134A658
IDataObject	0000010E-0000-0000-C000-000000000046	3	Flash.ocx+0x134A5F8
IDispatch	00020400-0000-0000-C000-000000000046	7	Flash.ocx+0x1349F68
IOleCommandTarget	B722BCCB-4E68-101B-A2BC-00AA00404770	3	Flash.ocx+0x134A830
IOleControl	B196B288-BAB4-101A-B69C-00AA00341D07	3	Flash.ocx+0x134A3E0
IOleInPlaceActiveObject	00000117-0000-0000-C000-000000000046	3	Flash.ocx+0x134A4D8
IOleInPlaceObject	00000113-0000-0000-C000-000000000046	3	Flash.ocx+0x134A5A0
IOleObject	00000112-0000-0000-C000-000000000046	3	Flash.ocx+0x134A418
IOleWindow	00000114-0000-0000-C000-000000000046	2	Flash.ocx+0x134AFA0

VTable RVAs for interface

OOP Reverse Engineering

OleViewDotNet 64bit

File Registry Object Security Help

CLSIDs by Local Server FlashBroker Properties

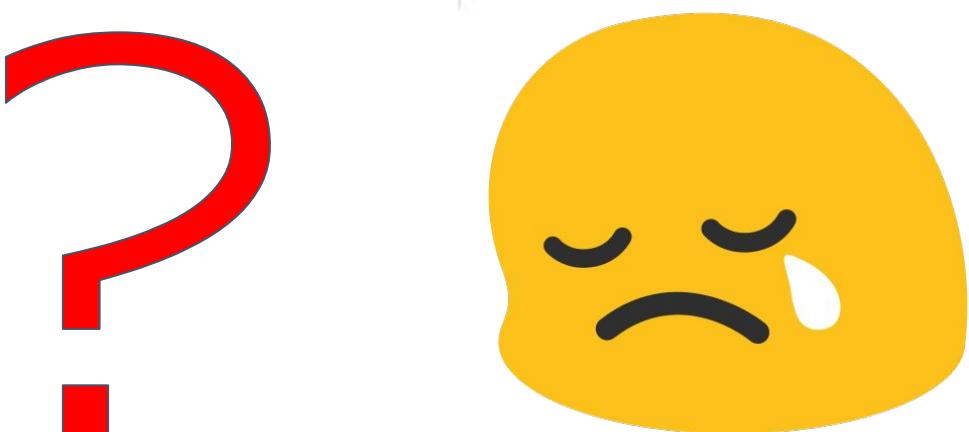
CLSID Supported Interfaces Type Library Elevation

Interfaces: Refresh

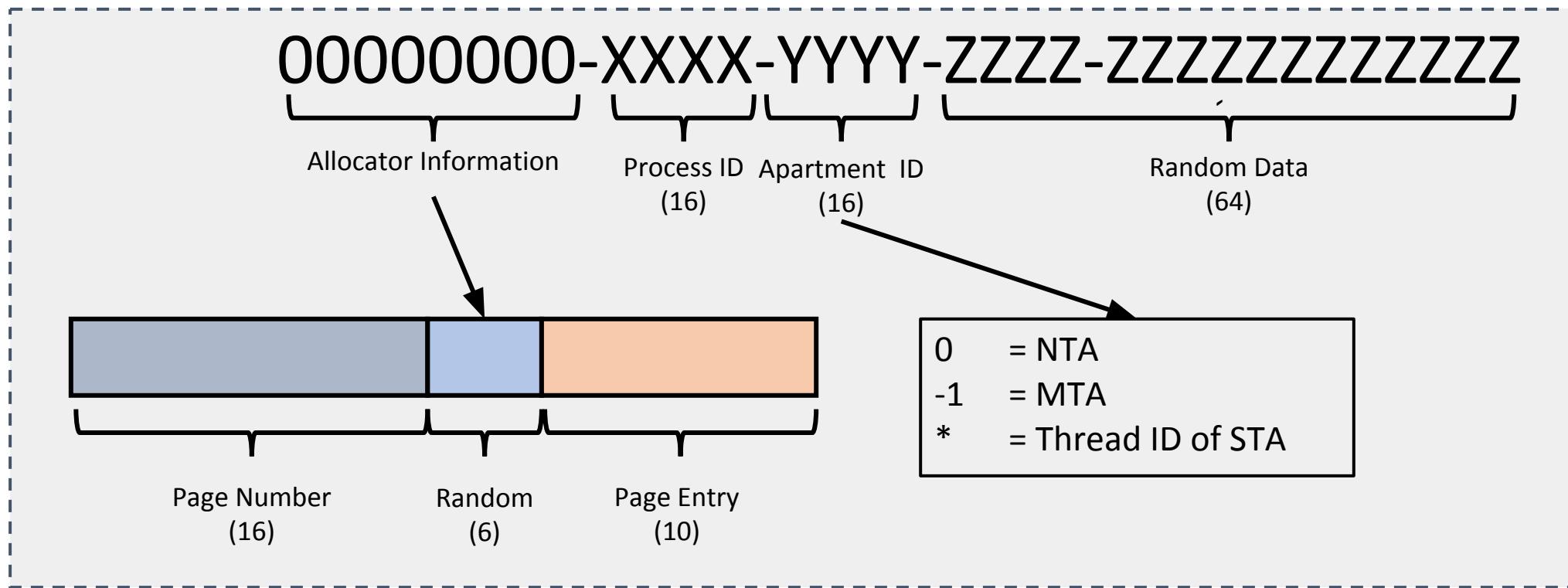
Name	IID	Methods	VTable Offset
IDispatch	00020400-0000-0000-C000-000000000046	7	
IFlashBroker6	299817DA-1FAC-4CE2-8F48-A108237013BD	3	
IMarshal	00000003-0000-0000-C000-000000000046	9	
IUnknown	00000000-0000-0000-C000-000000000046	3	

?

Object Properties



IPIPID Structure



Tracking Down OOP VTable

The screenshot shows the OleViewDotNet application window. The title bar reads "OleViewDotNet - Administrator - 64bit". The menu bar includes File, Registry, Object, Security, and Help. A tab labeled "Windows Management and Ins..." is selected. The main pane displays the properties of the service:

Properties:

- CLSID: 8BC3F05E-D86B-11D0-A075-00C04FB68820
- Name: Windows Management and Instrumentation
- Server: <APPID HOSTED>

A large red watermark "Create Instance" is overlaid on the center of the window.

Interfaces:

Name	IID	Viewer
IMarshal	00000003-0000-0000-C000-000000000046	No
IUnknown	00000000-0000-0000-C000-000000000046	No
IWbemLevel1Login	F309AD18-D86A-11D0-A075-00C04FB68820	No
IWbemLoginClientID	D4781CD6-E5D3-44DF-AD94-930EFE48A887	No

Operations ▾

Tracking Down OOP VTable

OleViewDotNet - Administrator - 64bit

File Registry Object Security Help

Windows Management and Instr...

Properties:

CLSID 8BC3F05E-D86B-11D0-A075-00C04FB68820
Name Windows Management and Instrumentation
Server <APPID HOSTED>

Marshal

Interfaces:

Name	IID	Viewer
IMarshal	00000003-0000-0000-C000-000000000046	No
IUnknown	00000000-0000-0000-C000-000000000046	No
IWbemLevel1Login	F309AD18-D86A-11D0-A075-00C04FB68820	No
IWbemLoginClientID	D4781CD6-E5D3-44DF-AD94-930EFE48A887	No

Operations ▾

OleViewDotNet - Administrator - 64bit

File Registry Object Security Help

Windows Management and Instr... Marshal Viewer - Standard

OBJREF Type: Standard IID: F309AD18-D86A-11D0-A075-00C04FB68820 IID Name: IWbemLevel1Login

Standard Flags: 0x0 Public Refs: 1

OXID: 0x4DDFA1BDC588A3DB OID: 0x2DA8A0652473E24F

IPID: 0001482F-01A8-0000-F9F2-3464FAED95DA Apartment: NTA

Process ID: 424 Process Name: svchost

String Bindings:

Tower ID	Network Address
Tcp	DESKTOP-C1HN3FN
Tcp	10.0.2.15
Tcp	192.168.56.103

Authentication Service Principal Name

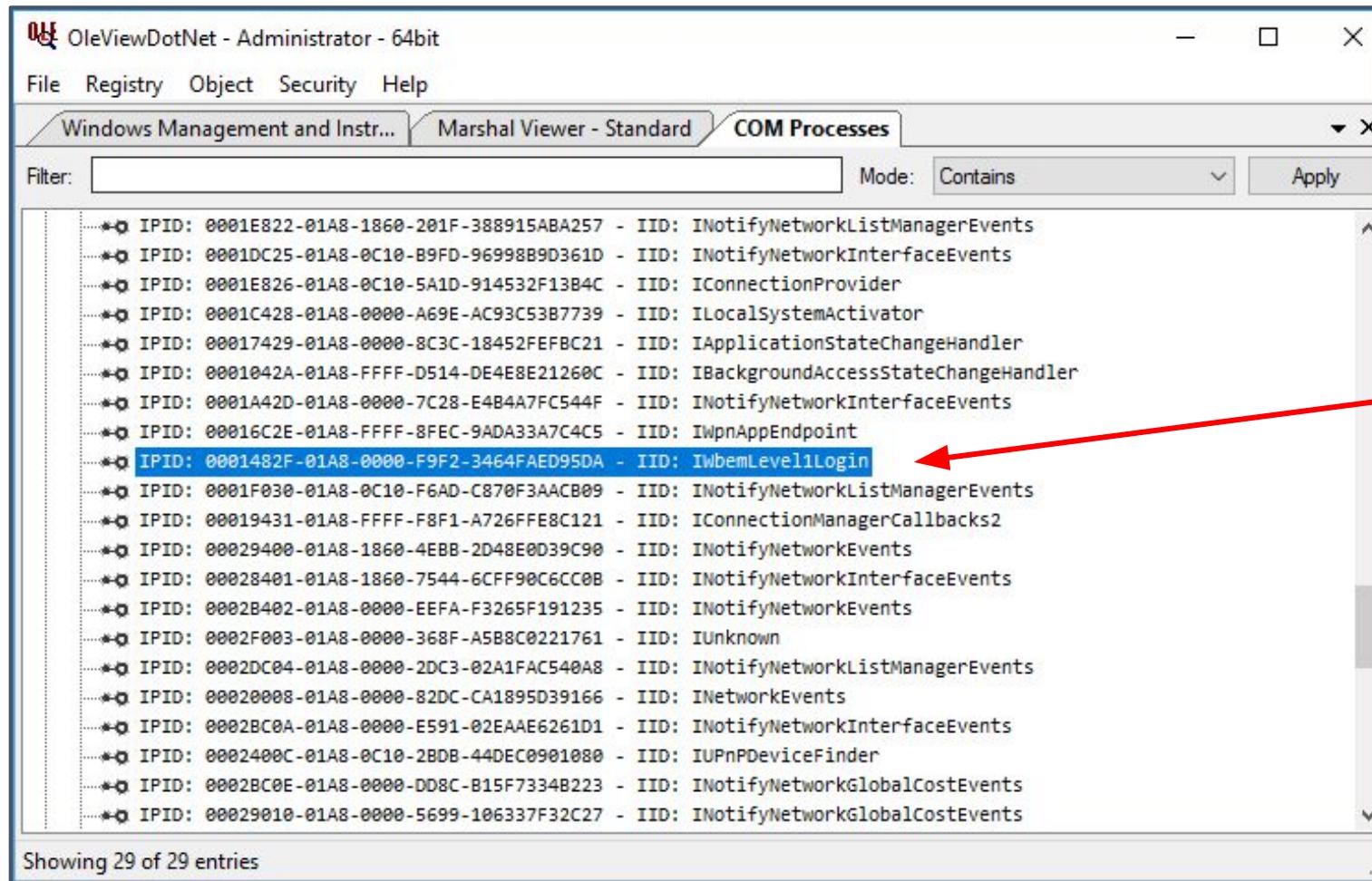
GSS_Negotiate

NegoExtender

GSS_Kerberos

PID from IPID

Tracking Down OOP VTable



The screenshot shows the OleViewDotNet application window titled "OleViewDotNet - Administrator - 64bit". The menu bar includes File, Registry, Object, Security, and Help. Below the menu is a toolbar with tabs: Windows Management and Instr..., Marshal Viewer - Standard, and COM Processes (which is selected). A filter bar with "Filter:" and "Mode: Contains" dropdown, and an "Apply" button is present. The main pane displays a list of entries, each consisting of an IPID and its corresponding IID and interface name. An arrow points from the text "Find IPID in Process List" to the IPID entry highlighted in blue.

IPID	IID	Interface
0001E822-01A8-1860-201F-388915ABA257	-	INotifyNetworkListManagerEvents
0001DC25-01A8-0C10-B9FD-9699889D361D	-	INotifyNetworkInterfaceEvents
0001E826-01A8-0C10-5A1D-914532F13B4C	-	IConnectionProvider
0001C428-01A8-0000-A69E-AC93C53B7739	-	ILocalSystemActivator
00017429-01A8-0000-8C3C-18452FEFB21	-	IApplicationStateChangeHandler
0001042A-01A8-FFFF-D514-DE4E8E21260C	-	IBackgroundAccessStateChangeHandler
0001A42D-01A8-0000-7C28-E4B4A7FC544F	-	INotifyNetworkInterfaceEvents
00016C2E-01A8-FFFF-8FEC-9ADA33A7C4C5	-	IPnPAppEndpoint
0001482F-01A8-0000-F9F2-3464FAED95DA	-	IWbemLevel1Login
0001F030-01A8-0C10-F6AD-C870F3AACB09	-	INotifyNetworkListManagerEvents
00019431-01A8-FFFF-F8F1-A726FFE8C121	-	IConnectionManagerCallbacks2
00029400-01A8-1860-4EBB-2D48E0D39C90	-	INotifyNetworkEvents
00028401-01A8-1860-7544-6CFF90C6CC0B	-	INotifyNetworkInterfaceEvents
0002B402-01A8-0000-EEFA-F3265F191235	-	INotifyNetworkEvents
0002F003-01A8-0000-368F-A5B8C0221761	-	IUnknown
0002DC04-01A8-0000-2DC3-02A1FAC540A8	-	INotifyNetworkListManagerEvents
00020008-01A8-0000-82DC-CA1895D39166	-	INetworkEvents
0002BC0A-01A8-0000-E591-02EAAE6261D1	-	INotifyNetworkInterfaceEvents
0002400C-01A8-0C10-2BDB-44DEC0901080	-	IUPnPDeviceFinder
0002BC0E-01A8-0000-DD8C-B15F7334B223	-	INotifyNetworkGlobalCostEvents
00029010-01A8-0000-5699-106337F32C27	-	INotifyNetworkGlobalCostEvents

Showing 29 of 29 entries

Find IPID in Process List

Tracking Down OOP VTable

The image shows two windows from the OleViewDotNet application.

The left window is titled "COM Processes". It contains a list of entries, each showing an IPID and its corresponding IID and interface names. One entry is highlighted with a blue box:

- * IPID: 0001E822-01A8-1860-201F-388915ABA257 - IID: INotifyNetworkListManagerEvents
- * IPID: 0001DC25-01A8-0C10-B9FD-9699889D361D - IID: INotifyNetworkInterfaceEvents
- * IPID: 0001E826-01A8-0C10-5A1D-914532F13B4C - IID: IConnectionProvider
- * IPID: 0001C428-01A8-0000-A69E-AC93C53B7739 - IID: ILocalSystemActivator
- * IPID: 00017429-01A8-0000-8C3C-18452FEFBC21 - IID: IApplicationStateChange
- * IPID: 0001042A-01A8-FFFF-D514-DE4E8E21260C - IID: IBackgroundAccessSt
- * IPID: 0001A42D-01A8-0000-7C28-E4B4A7FC544F - IID: INotifyNetworkInter
- * IPID: 00016C2E-01A8-FFFF-8FEC-9ADA33A7C4C5 - IID: IWpnAppEndpoint
- * IPID: 0001482F-01A8-0000-F9F2-3464FAED95DA - IID: IWbemLevel1Login
- * IPID: 0001F030-01A8-0C10-F6AD-C870F3AACB09 - IID: INotifyNetworkListM
- * IPID: 00019431-01A8-FFFF-F8F1-A726FFE8C121 - IID: IConnectionManagerC
- * IPID: 00029400-01A8-1860-4EBB-2D48E0D39C90 - IID: INotifyNetworkEvent
- * IPID: 00028401-01A8-1860-7544-6CFF90C6CC0B - IID: INotifyNetworkInter
- * IPID: 0002B402-01A8-0000-EEFA-F3265F191235 - IID: INotifyNetworkEvent
- * IPID: 0002F003-01A8-0000-368F-A5B8C0221761 - IID: IUnknown
- * IPID: 0002DC04-01A8-0000-2DC3-02A1FAC540A8 - IID: INotifyNetworkListM
- * IPID: 00020008-01A8-0000-82DC-CA1895D39166 - IID: INetworkEvents
- * IPID: 0002BC0A-01A8-0000-E591-02EAAE6261D1 - IID: INotifyNetworkInter
- * IPID: 0002400C-01A8-0C10-2BDB-44DEC0901080 - IID: IUPnPDeviceFinder
- * IPID: 0002BC0E-01A8-0000-DD8C-B15F7334B223 - IID: INotifyNetworkGloba
- * IPID: 00029010-01A8-0000-5699-106337F32C27 - IID: INotifyNetworkGloba

The right window is titled "IPID: 0001482F-01A8-0000-F9F2-3464FAED95DA". It displays detailed information for this specific IPID, including its IID, Flags, Interface, Stub, OXID, References, PID, and STA HWND. A red arrow points to the "VTable" field, which contains the value "wbemcore+0xDA448".

IPID	IID	IID Name	Flags	Interface	VTable	Stub	VTable	OXID	References	PID	Apartment	STA HWND
0001482F-01A8-0000-F9F2-3464FAED95DA	F309AD18-D86A-11D0-A075-00C04FB68820	IWbemLevel1Login	IPIDF_SERVERENTRY	0x13AFD765910	wbemcore+0xDA448	0x13AF52B64F0	wbemsvc+0x765	C588A3DB-A1BD-4DDF-6117-1CF7D8ED199D	Strong: 5, Weak: 0, Private: 0	424	NTA	0x0

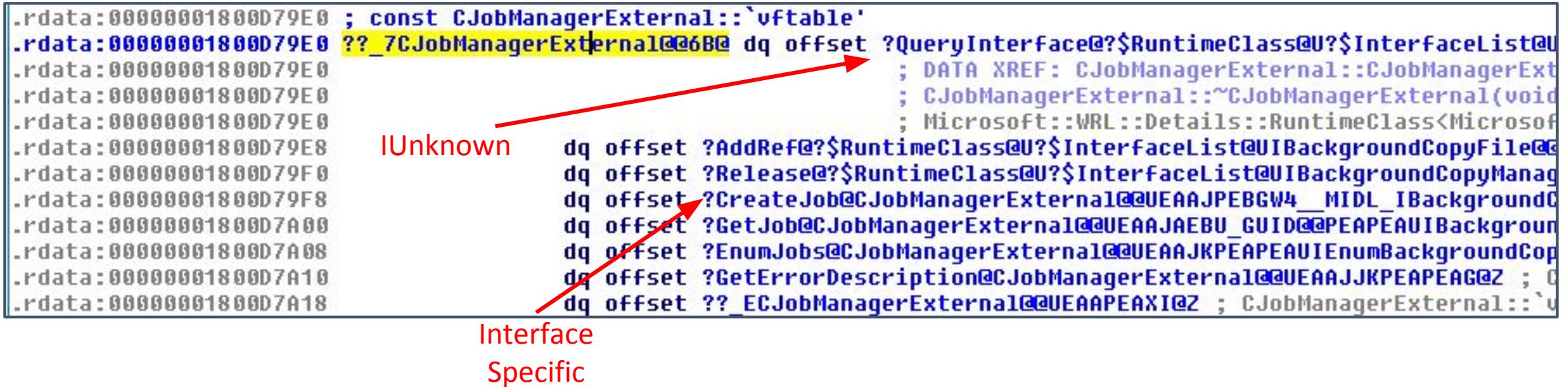
Showing 29 of 29 entries

VTable RVA

VTable Reverse Engineering

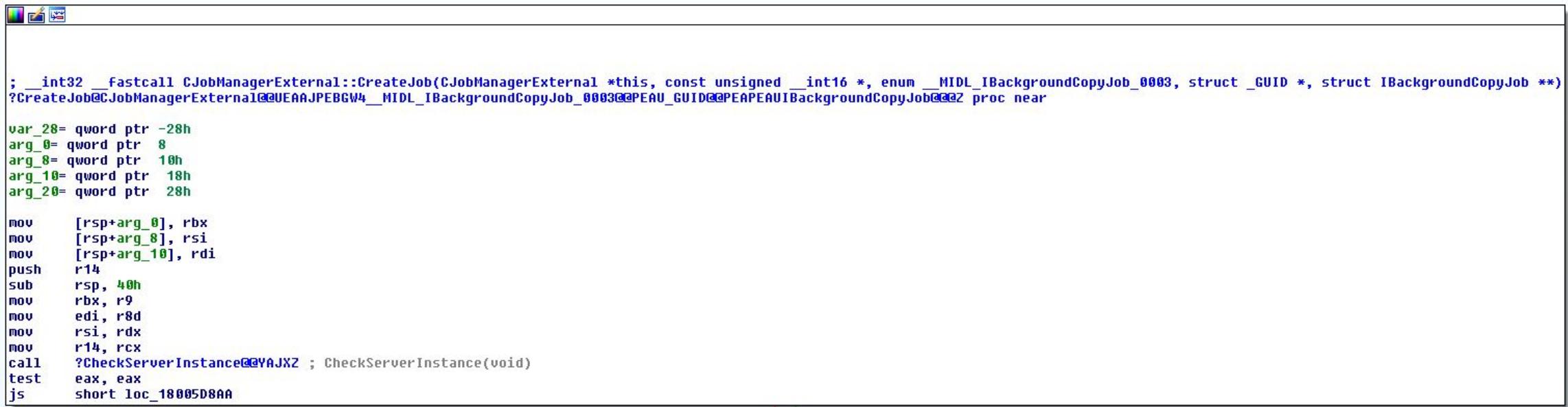
```
.rdata:00000001800D79E0 ; const CJobManagerExternal::`vtable'  
.rdata:00000001800D79E0 ??_7CJobManagerExternal@6B@ dq offset ?QueryInterface@?$RuntimeClass@U?$InterfaceList@U  
.rdata:00000001800D79E0  
.rdata:00000001800D79E0  
.rdata:00000001800D79E0  
.rdata:00000001800D79E0  
.rdata:00000001800D79E8 IUnknown dq offset ?AddRef@?$RuntimeClass@U?$InterfaceList@UIBackgroundCopyFile@@  
.rdata:00000001800D79F0 dq offset ?Release@?$RuntimeClass@U?$InterfaceList@UIBackgroundCopyManag  
.rdata:00000001800D79F8 dq offset ?CreateJob@CJobManagerExternal@@UEAJPEBGW4__MIDL_IBackgroundC  
.rdata:00000001800D7A00 dq offset ?GetJob@CJobManagerExternal@@UEAAJAEBU_GUID@@PEAPEAUIBackgroun  
.rdata:00000001800D7A08 dq offset ?EnumJobs@CJobManagerExternal@@UEAAJKPEAPEAUIEnumBackgroundCop  
.rdata:00000001800D7A10 dq offset ?GetErrorDescription@CJobManagerExternal@@UEAAJJKPEAPEAG@Z ; 0  
.rdata:00000001800D7A18 dq offset ??_ECJobManagerExternal@@UEAAPEAXI@Z ; CJobManagerExternal::`v
```

Interface Specific



Interface Information - Public Symbols

- Most Windows components come with public symbols available
- Most COM code written in C++
- So, use C++ managed names to recover some parameter information



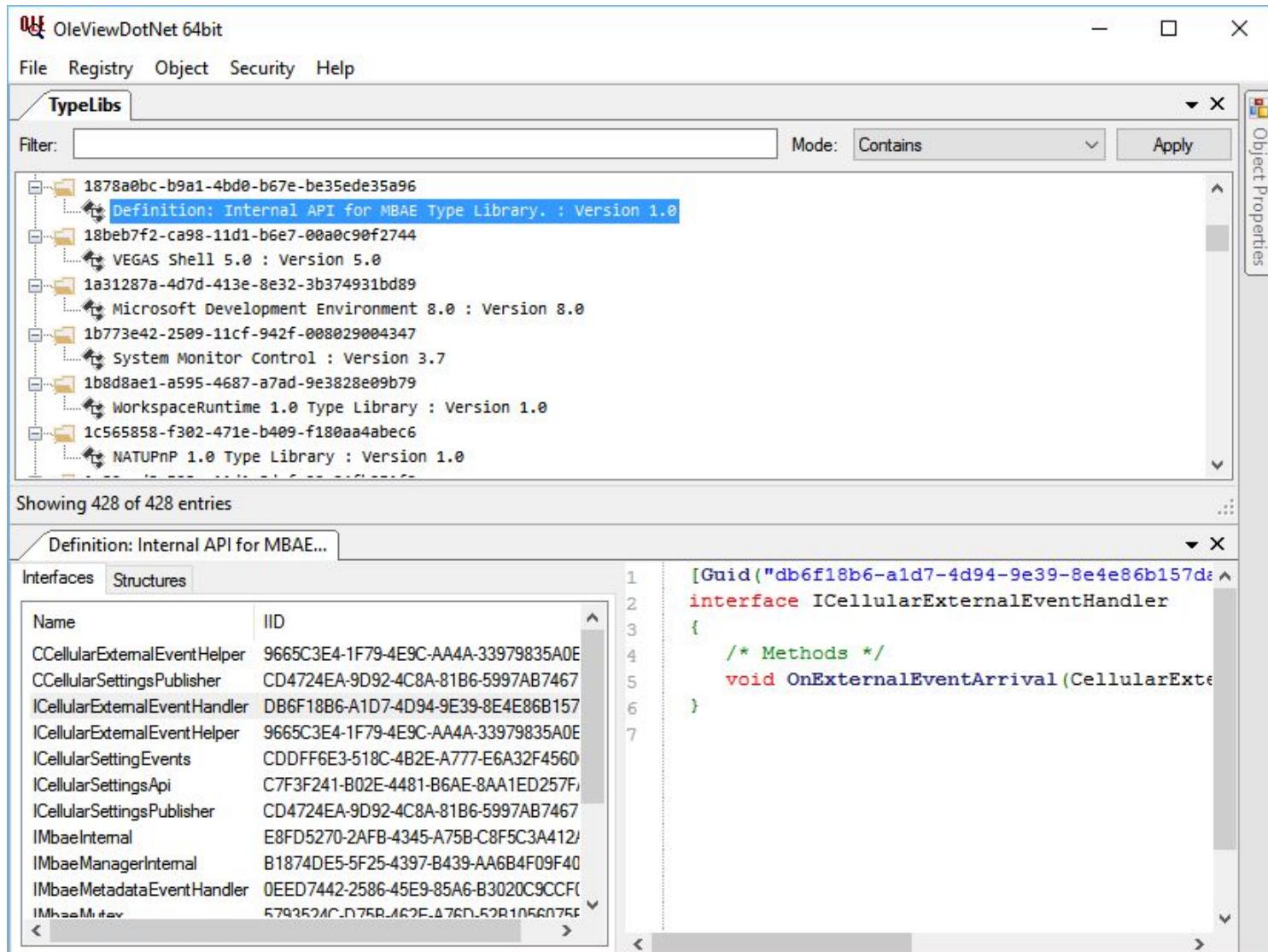
The screenshot shows a debugger window displaying assembly code. The code is annotated with various labels and comments in blue, likely indicating public symbols or managed names. The assembly instructions include moves to registers (mov), pushes to the stack (push), and calls to functions (call). The code appears to be part of a Windows component, specifically related to job management.

```
; __int32 __fastcall CJobManagerExternal::CreateJob(CJobManagerExternal *this, const unsigned __int16 *, enum __MIDL_IBackgroundCopyJob_0003, struct _GUID *, struct IBackgroundCopyJob **)
?CreateJob@CJobManagerExternal@@UEAJPEBGW4__MIDL_IBackgroundCopyJob_0003@PEAU_GUID@@PEAPEUIBackgroundCopyJob@@Z proc near

var_28= qword ptr -28h
arg_0= qword ptr 8
arg_8= qword ptr 10h
arg_10= qword ptr 18h
arg_20= qword ptr 28h

mov    [rsp+arg_0], rbx
mov    [rsp+arg_8], rsi
mov    [rsp+arg_10], rdi
push   r14
sub    rsp, 40h
mov    rbx, r9
mov    edi, r8d
mov    rsi, rdx
mov    r14, rcx
call   ?CheckServerInstance@@YAJXZ ; CheckServerInstance(void)
test   eax, eax
js     short loc_18005D8AA
```

Interface Information - Type Libraries



Interface Information - Proxy/Stub NDR

OleViewDotNet 64bit

File Registry Object Security Help

Proxy CLSIDs tpcps.dll

Interfaces Structures

Name	IID
IInkManager	A5558057-9B96-46BA-94ED-982E684A9A6
IInkObject	B9C4A0C1-16ED-4DC2-B34A-4E83032658
IInkPoint	3776F33D-6BF8-4ADD-9C7E-946AB4A771
IInkStroke	251F1257-1DCB-4AD0-A826-4F9E326FE49
ILattice	946665F9-71E3-447B-A896-3359DA41153
IPenService	9E358D23-02B2-4CCD-9FEE-6B75EE8DD!
IPenServiceSampleCollector	B0ECE4A1-FF78-43FA-B06E-EBEAAA1BFD
IPenServiceTablet	5EE46C56-1537-4CC5-8A93-A7ED4D1393
IRecoAsyncResults	250CEF9C-121F-493D-ADCD-7A2A6823C0
IRecoContext	E6DAB875-75AF-4C8A-9665-2B6A44DD0F
IRecognizer	3A182AD6-596A-4070-A574-73941817B67
IRecoManager	C2E8F101-5D03-42EE-B90A-35255783103
IRecoWordList	37ADC645-ACE6-4A31-B6A1-FD1F4EF480
IRenderingContext	4E6B4F16-5A0C-4815-9AA2-DE231F5AAA
IRenderInk	538A9C7B-858A-4FF1-9769-62B6D74993C
IStrokeGeometry	11F962C5-242E-4D4D-B205-0F3AB3562F1
IStrokeSet	2080FF4F-297F-4F66-AA83-CACA65F6721
ITablet	1CB2EFC3-ABC7-4172-8FCB-3BC9CB93E2
ITablet2	C247F616-BBEB-406A-AED3-F75E656599
ITablet3	AC0E3951-0A2F-448E-88D0-49DA0C4534
ITabletContext	45AAAF04-9D6F-41AF-8FD1-FCD6D4B2F

```
[Guid("c2e8f101-5d03-42ee-b90a-352557831039")]
interface IRecoManager : IUnknown {
    HRESULT Proc3 /* Stack Offset: 8 */ [In, Out] int* p0,
    HRESULT Proc4 /* Stack Offset: 8 */ [In] struct Struct,
    HRESULT Proc5 /* Stack Offset: 8 */ [In] GUID* p0, /*
    HRESULT Proc6 /* Stack Offset: 8 */ [Out] GUID* p0, /*
    HRESULT Proc7 /* Stack Offset: 8 */ [In] GUID* p0, /*
    HRESULT Proc8 /* Stack Offset: 8 */ [In] GUID* p0, /*
    HRESULT Proc9 /* Stack Offset: 8 */ [Out] IRecoWordList
}
```

62

Bugs and "Features"

Activation Properties In SPD

```
struct SpecialPropertiesData {  
    unsigned long dwSessionId;  
    long fRemoteThisSessionId;  
    long fClientImpersonating;  
    long fPartitionIDPresent;  
    DWORD dwDefaultAuthnLvl;  
    GUID guidPartition;  
    DWORD dwPRTFlags;  
    DWORD dwOrigClсх;  
    DWORD dwFlags;  
    DWORD dwPid;  
    unsigned __int64 hwnd;  
    DWORD ulServiceId;  
    DWORD Reserved[4];  
};
```

Choosing a Session ID?

UAC Related Stuff

```
enum SPD_FLAGS {  
    SPD_FLAG_USE_CONSOLE_SESSION,  
    SPD_FLAG_USE_DEFAULT_AUTHN_LVL,  
    SPD_FLAG_USE_SERVER_PID,  
    SPD_FLAG_USE_LUA_LEVEL_ADMIN,  
    SPD_FLAG_COAUTH_USER_IS_NULL,  
    SPD_FLAG_COAUTH_DOMAIN_IS_NULL,  
    SPD_FLAG_COAUTH_PWD_IS_NULL,  
    SPD_FLAG_USE_LUA_LEVEL_HIGHEST  
};
```

Session and Elevation Monikers

Session-to-Session Activation with a Session Moniker

Session-to-session activation (also called cross-session activation) allows a client process to start (activate) a local server process on a specified session. This feature is available for applications that are configured to run in the security context of the interactive user, also known as the "RunAs Interactive User" object activation mode. For more information about security contexts, see [The Client's Security Context](#).

Distributed COM (DCOM) enables object activation on a per-session basis by using a system-supplied **Session Moniker**. Other system-supplied monikers include [file monikers](#), [item monikers](#), generic [composite monikers](#), [anti-monikers](#), [pointer monikers](#), and [URL monikers](#).

The COM Elevation Moniker

The COM elevation moniker allows applications that are running under user account control (UAC) to activate COM classes with elevated privileges. For more information, see [Focus on Least Privilege](#).

When to Use the Elevation Moniker

The elevation moniker is used to activate a COM class to accomplish a specific and limited function that requires elevated privileges, such as changing the system date and time.

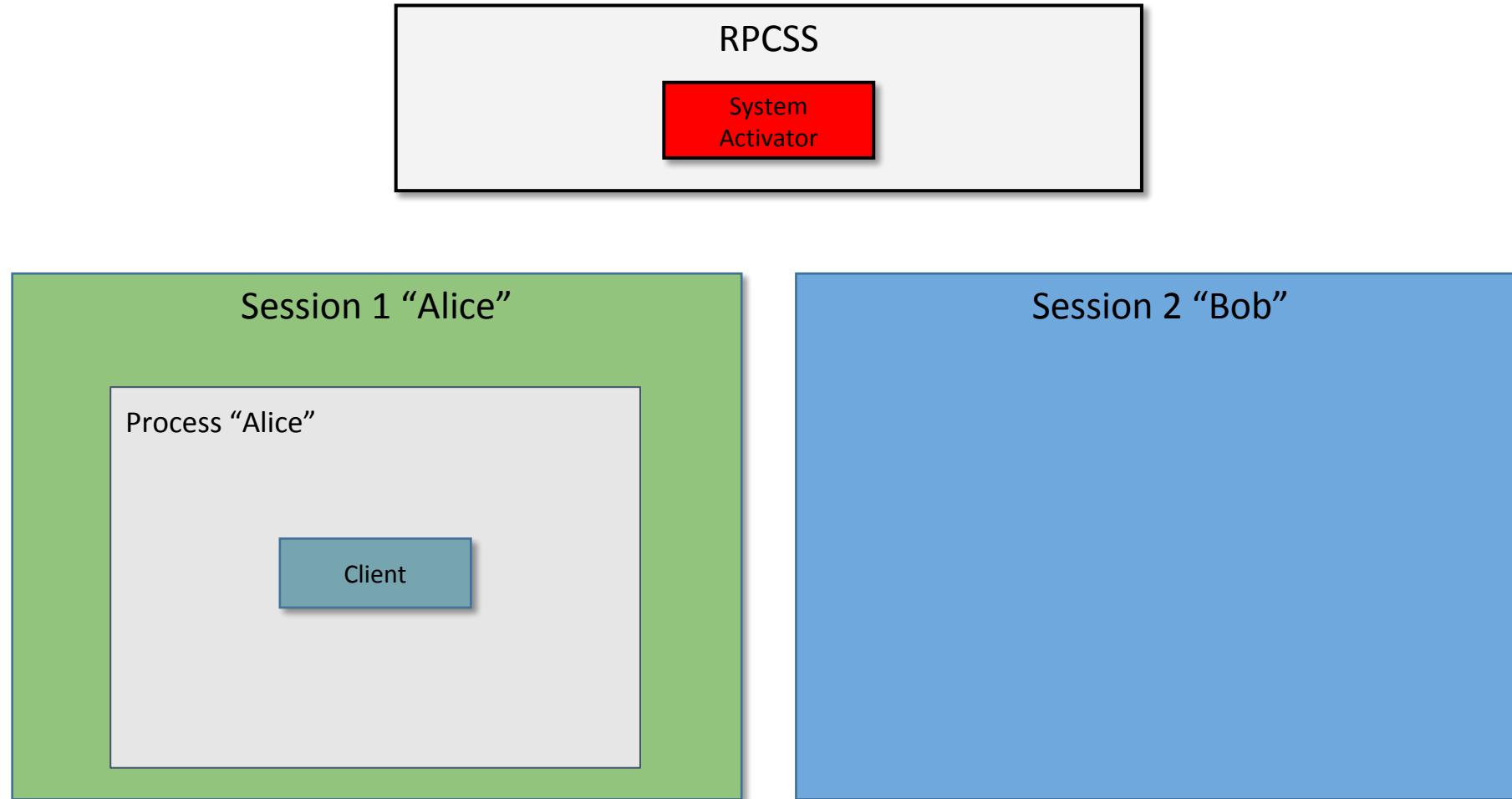
Elevation requires participation from both a COM class and its client. The COM class must be configured to support elevation by annotating its registry entry, as described in the Requirements section. The COM client must request elevation by using the elevation moniker.

Monikers and Binding

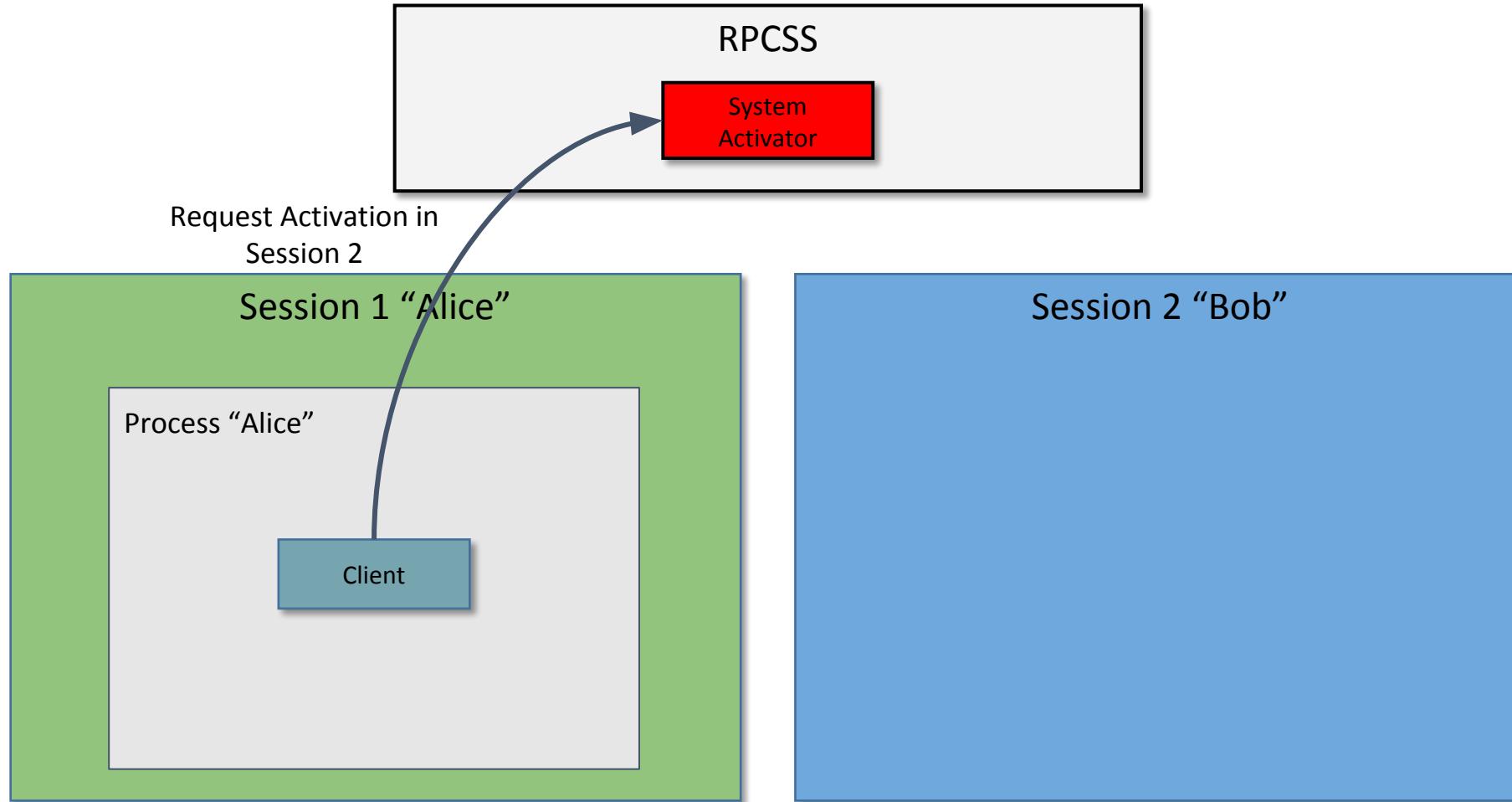
```
HRESULT CoCreateInstanceInSession( DWORD session,
REFCLSID rclsid, REFIID riid, void ** ppv) {
BIND_OPTS3 bo = {};
WCHAR wszCLSID[50];
WCHAR wszMonikerName[300];

StringFromGUID2(rclsid, wszCLSID, _countof(wszCLSID));
StringCchPrintf(wszMonikerName, _countof(wszMonikerName),
L"session:%d!new:%s", session, wszCLSID);
bo.cbStruct = sizeof(bo);
bo.dwClassContext = CLSCTX_LOCAL_SERVER;
return CoGetObject(wszMonikerName, &bo, riid, ppv);
}
```

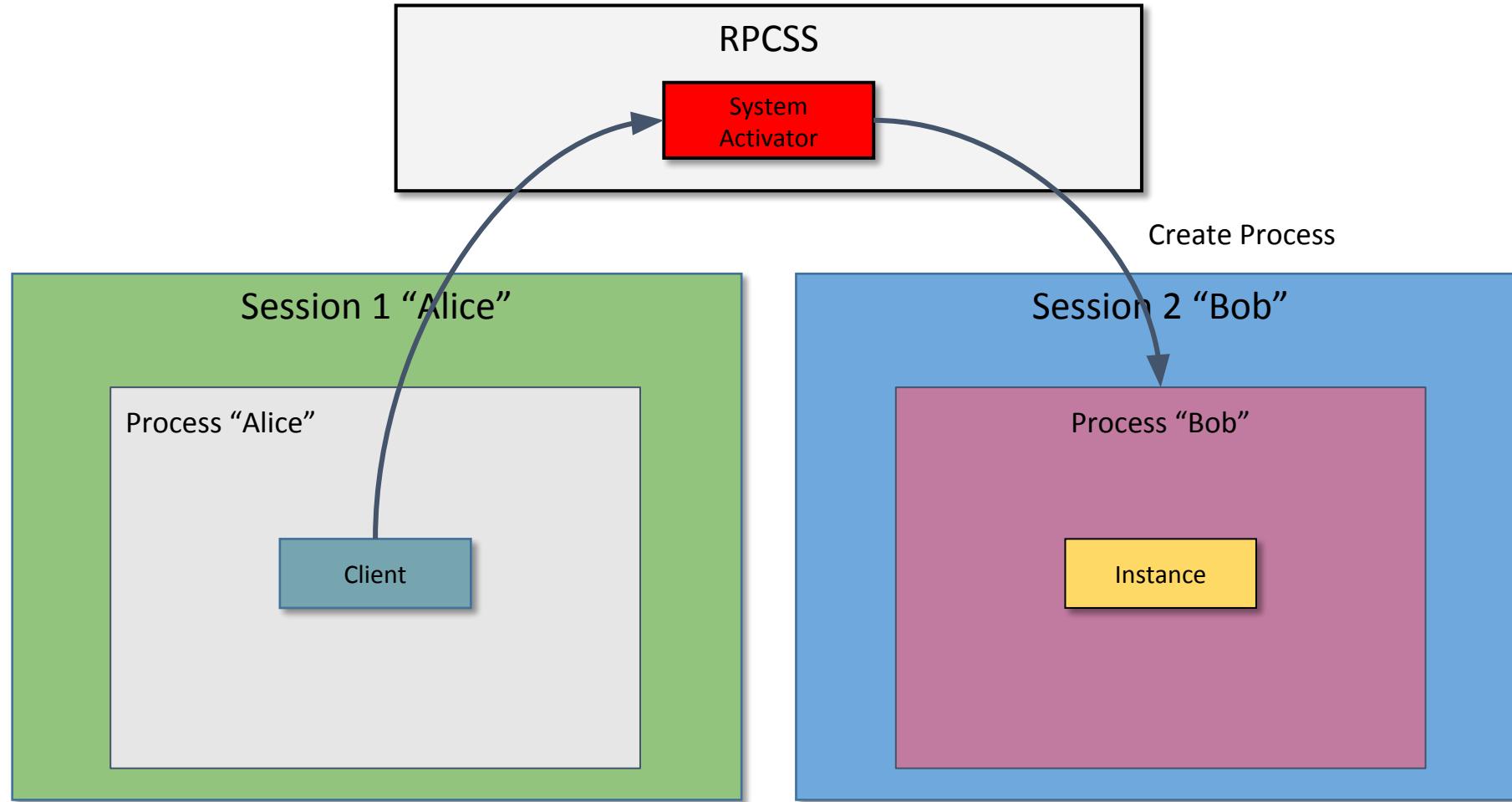
Session Moniker in Action



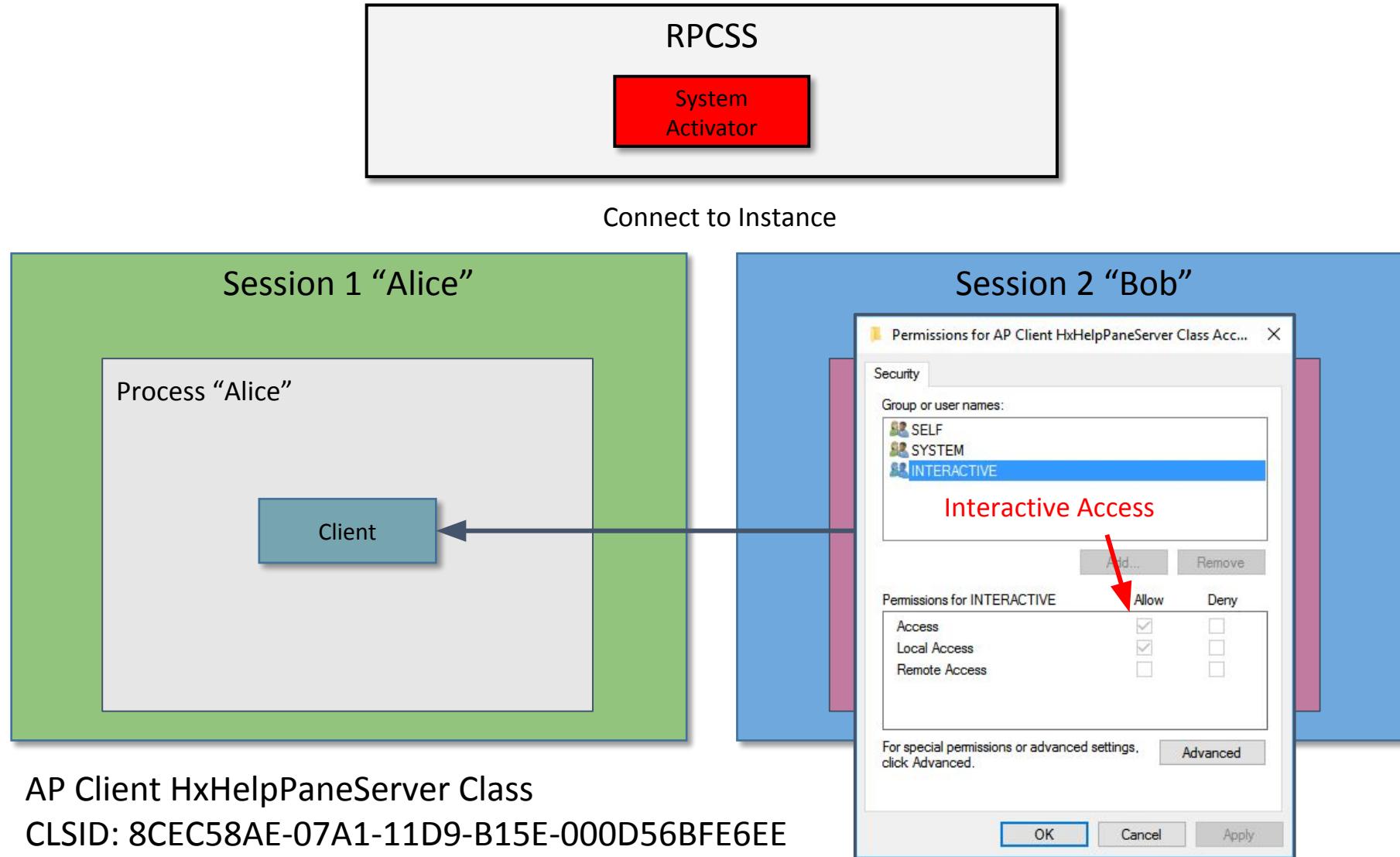
Session Moniker in Action



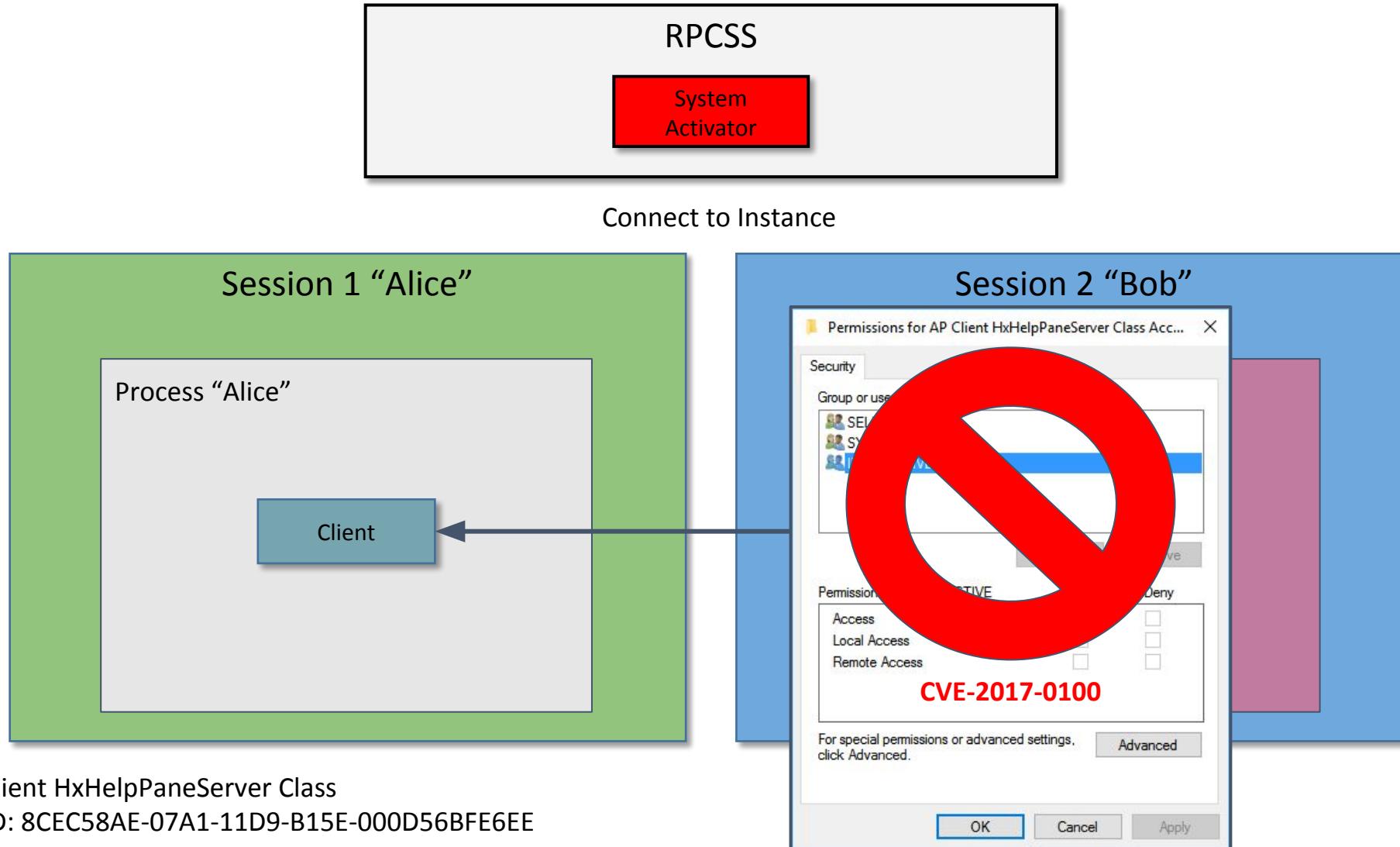
Session Moniker in Action



Session Moniker in Action



Session Moniker in Action



Abusing Information Disclosure

The screenshot shows two windows from the OleViewDotNet 64bit application.

Left Window (COM Processes):

- File Registry Object Security Help
- Filter: [] Mode: Contains [] Apply
- IPID 0000140b-0e64-0e68-e75c-d4cd455824f1 Properties:
- CLSID 00000000-0000-0000-0000-000000000000
- Interfaces:

Name	IID	Viewer
ILocalSystemActivator	00000132-0000-0000-C000-000000000046	No
IMarshal	00000003-0000-0000-C000-000000000046	No
ISystemActivator	000001A0-0000-0000-C000-000000000046	No
IUnknown	00000000-0000-0000-C000-000000000046	No

- Operations []

Right Window (Properties):

- IPID 0000140b-0e64-0e68-e75c-d4cd455824f1 Properties:
- CLSID 00000000-0000-0000-0000-000000000000
- Interfaces:

Name	IID	Viewer
ILocalSystemActivator	00000132-0000-0000-C000-000000000046	No
IMarshal	00000003-0000-0000-C000-000000000046	No
ISystemActivator	000001A0-0000-0000-C000-000000000046	No
IUnknown	00000000-0000-0000-C000-000000000046	No

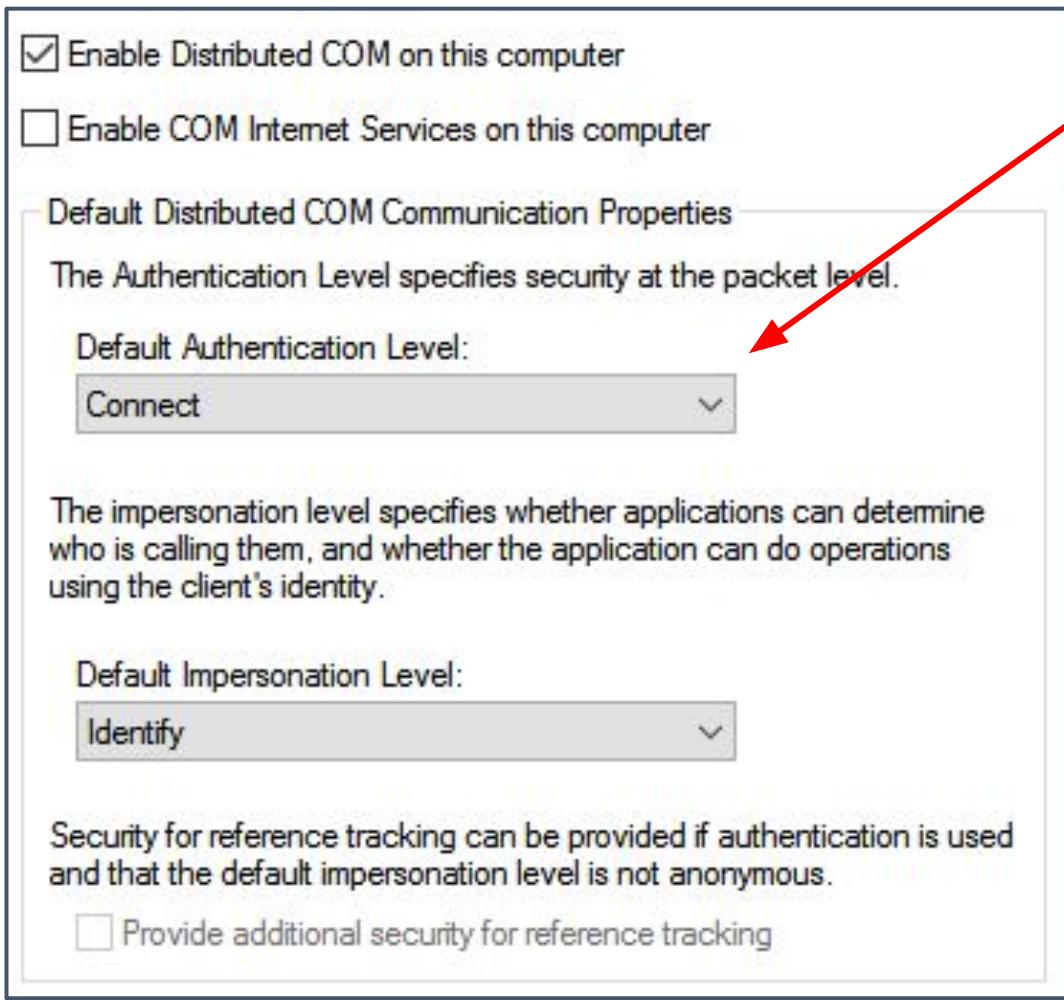
- Operations []

A red arrow points from the highlighted entry in the left window to the text "Unmarshal" in the right window.

You Can't Help Loving DCOM



Plain Text Communications?



Defaults to CONNECT Authentication Level?

Authentication Level	
CONNECT	Authenticates the credentials of the client and server.
CALL/PKT	Same as CONNECT but also prevents replay attacks.
PKT_INTEGRITY	Same as CALL/PKT but also verifies that none of the data transferred between the client and server has been modified.
PKT_PRIVACY	Same as PKT_INTEGRITY but also ensures that the data transferred can only be seen unencrypted by the client and the server.

Good old Wireshark

*Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dcerpc && ip.addr == 192.168.56.102

No.	Time	Source	Destination	Protocol	Length	Info
126	58.048448	192.168.56.50	192.168.56.102	DCERPC	138	Bind_ack: call_id: 2, Fragment: Single, max_xm...
127	58.049465	192.168.56.102	192.168.56.50	IOXIDResolver	78	ServerAlive2 request IOXIDResolver V0
128	58.050659	192.168.56.50	192.168.56.102	IOXIDResolver	210	ServerAlive2 response[Long frame (2 bytes)]
145	58.062317	192.168.56.102	192.168.56.50	DCERPC	325	Bind: call_id: 3, Fragment: Single, 1 context ...
147	58.063403	192.168.56.50	192.168.56.102	DCERPC	290	Bind_ack: call_id: 3, Fragment: Single, max_xm...
148	58.063842	192.168.56.102	192.168.56.50	DCERPC	274	Alter_context: call_id: 3, Fragment: Single, 1...
149	58.064777	192.168.56.50	192.168.56.102	DCERPC	159	Alter_context_resp: call_id: 3, Fragment: Sing...
150	58.069377	192.168.56.102	192.168.56.50	ISystemActivator	854	RemoteCreateInstance request
151	58.101186	192.168.56.50	192.168.56.102	ISystemActivator	1022	RemoteCreateInstance response
169	58.118628	192.168.56.102	192.168.56.50	DCERPC	412	Bind: call_id: 2, Fragment: Single, 3 context ...

> [No Specification Available: 00000000]

 ` IActProperties
 CntData: 728

 ` OBJREF
 Signature: MEOW (0x574f454d)
 Flags: OBJREF_CUSTOM (0x00000004)
 IID: 000001a2-0000-0000-c000-000000000046

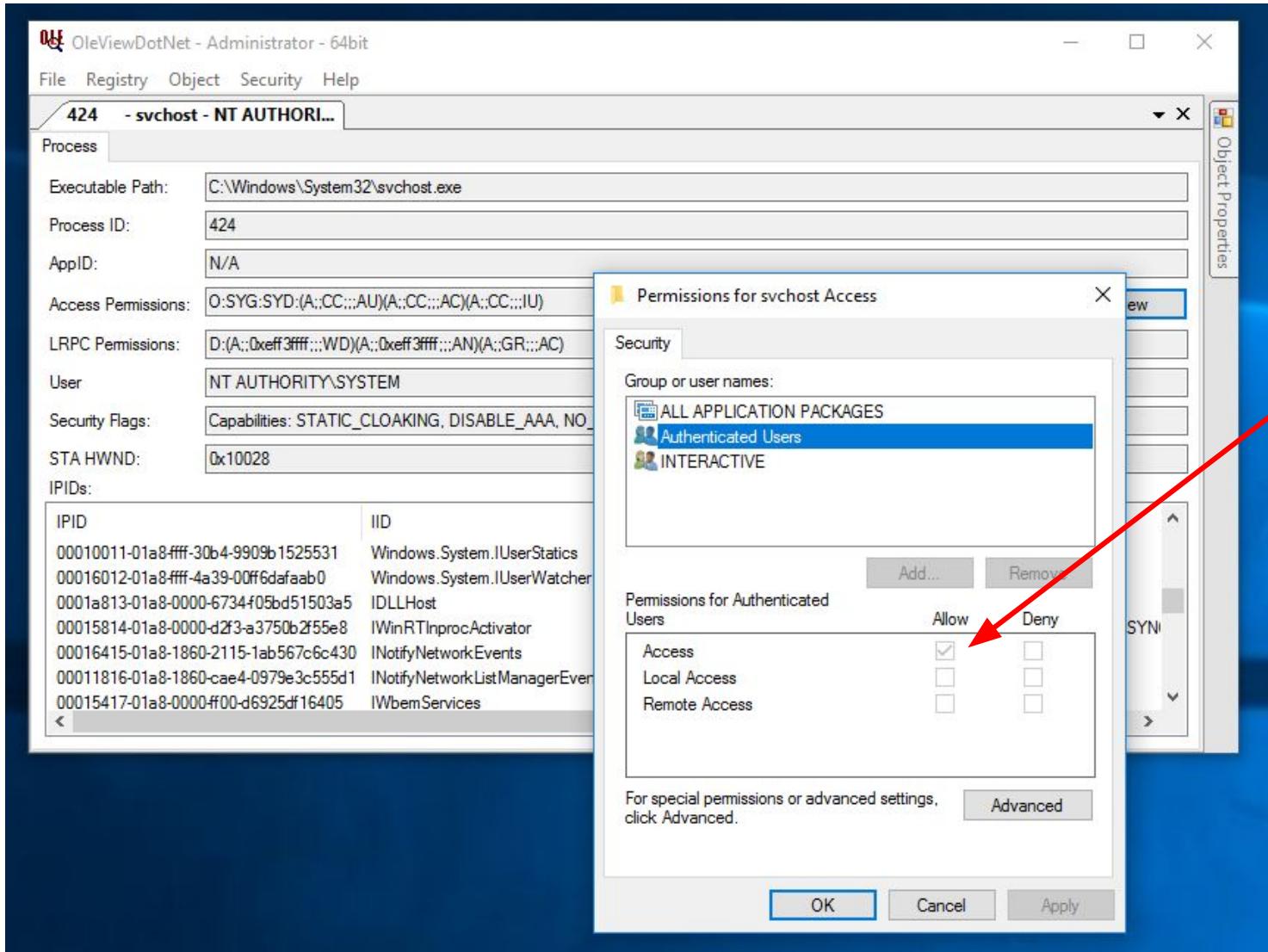
Hex	Dec	ASCII
0070	00 00 00 00 00 02 00 d8 02 00 00 00 4d 45ME
0080	4f 57 04 00 00 00 a2 01 00 00 00 00 00 c0 00	OW.....
0090	00 00 00 00 00 46 38 03 00 00 00 00 00 c0 00F8..
00a0	00 00 00 00 00 46 00 00 00 00 b0 02 00 00 a0 02F... .
00b0	00 00 00 00 00 00 01 10 08 00 cc cc cc cc b0 00
00c0	00 00 00 00 00 00 a0 02 00 00 c0 00 00 00 00 00
00d0	00 00 02 00 00 00 06 00 00 00 00 00 00 00 00 00
00e0	00 00 00 00 00 00 00 00 00 00 00 02 00 04 00
00f0	02 00 00 00 00 00 06 00 00 00 b9 01 00 00 00 00
0100	00 00 c0 00 00 00 00 00 00 46 ab 01 00 00 00 00 F....
0110	00 00 c0 00 00 00 00 00 00 46 a5 01 00 00 00 00 F....

OBJREF (dcom.objref), 728 bytes

Packets: 478 · Displayed: 216 (45.2%) · Dropped: 0 (0.0%) · Profile: Default

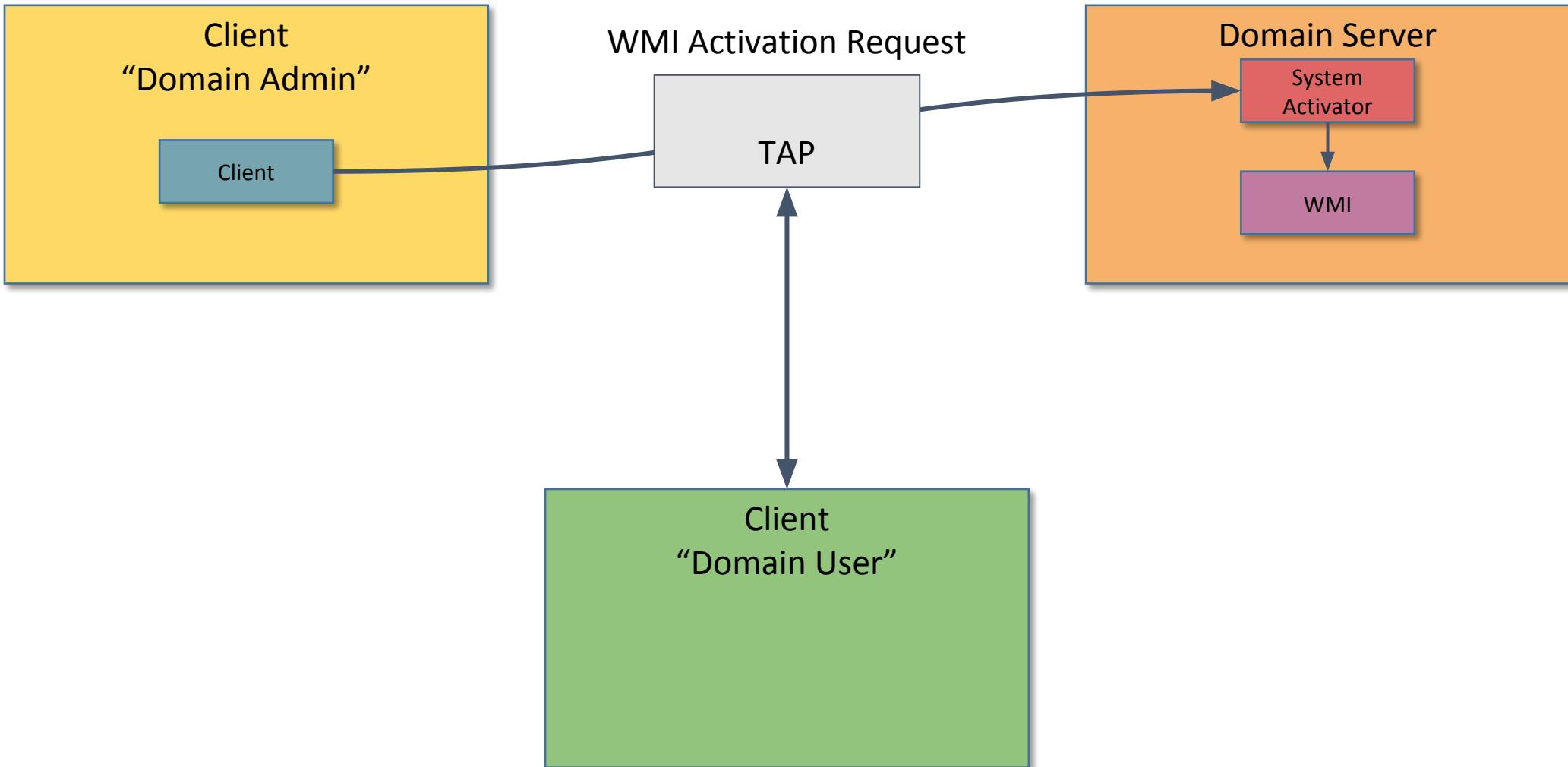
75

Access Permissions for WMI

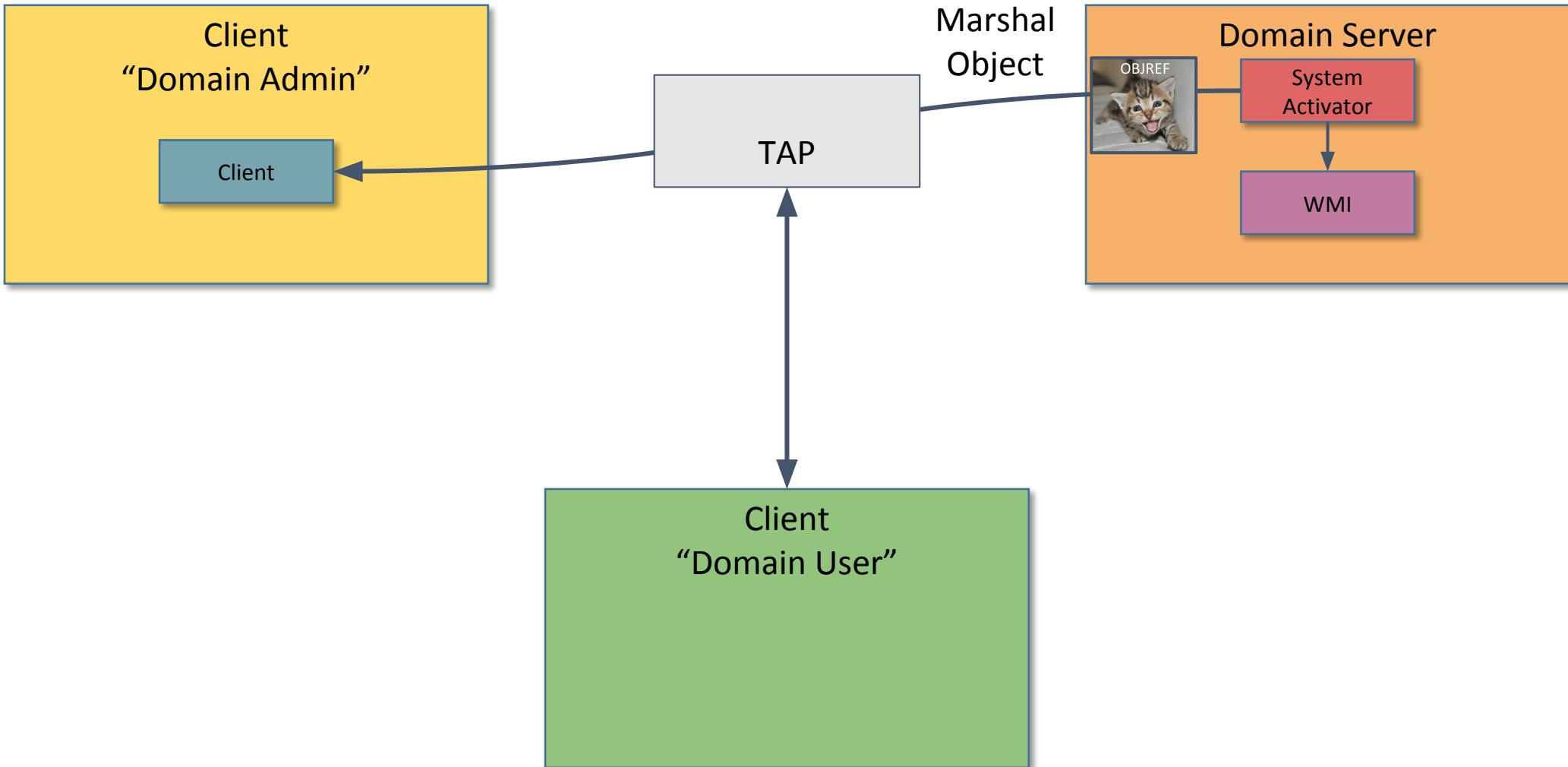


All authenticated
users can access WMI
remotely

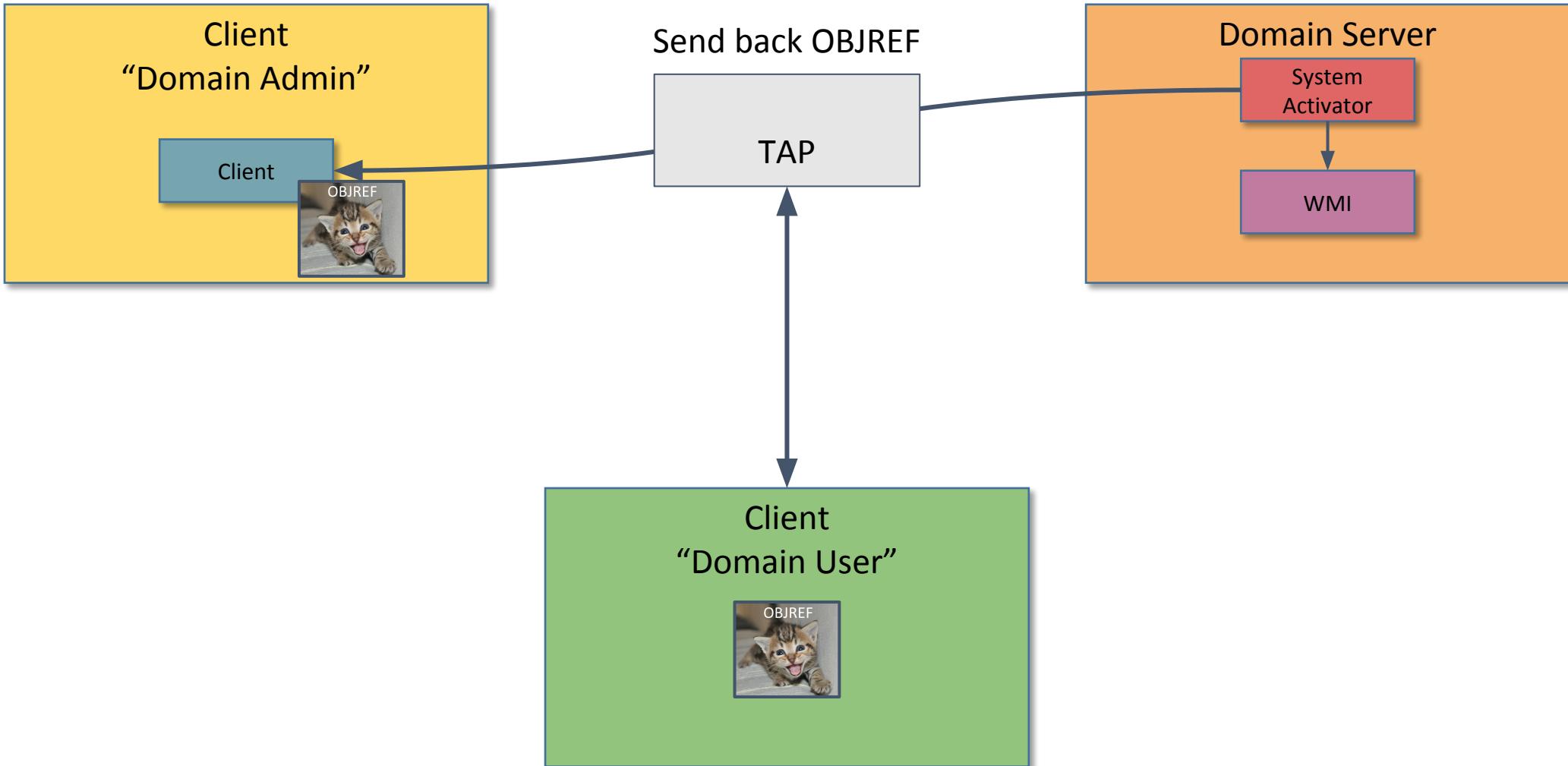
MEOW-Jacking!



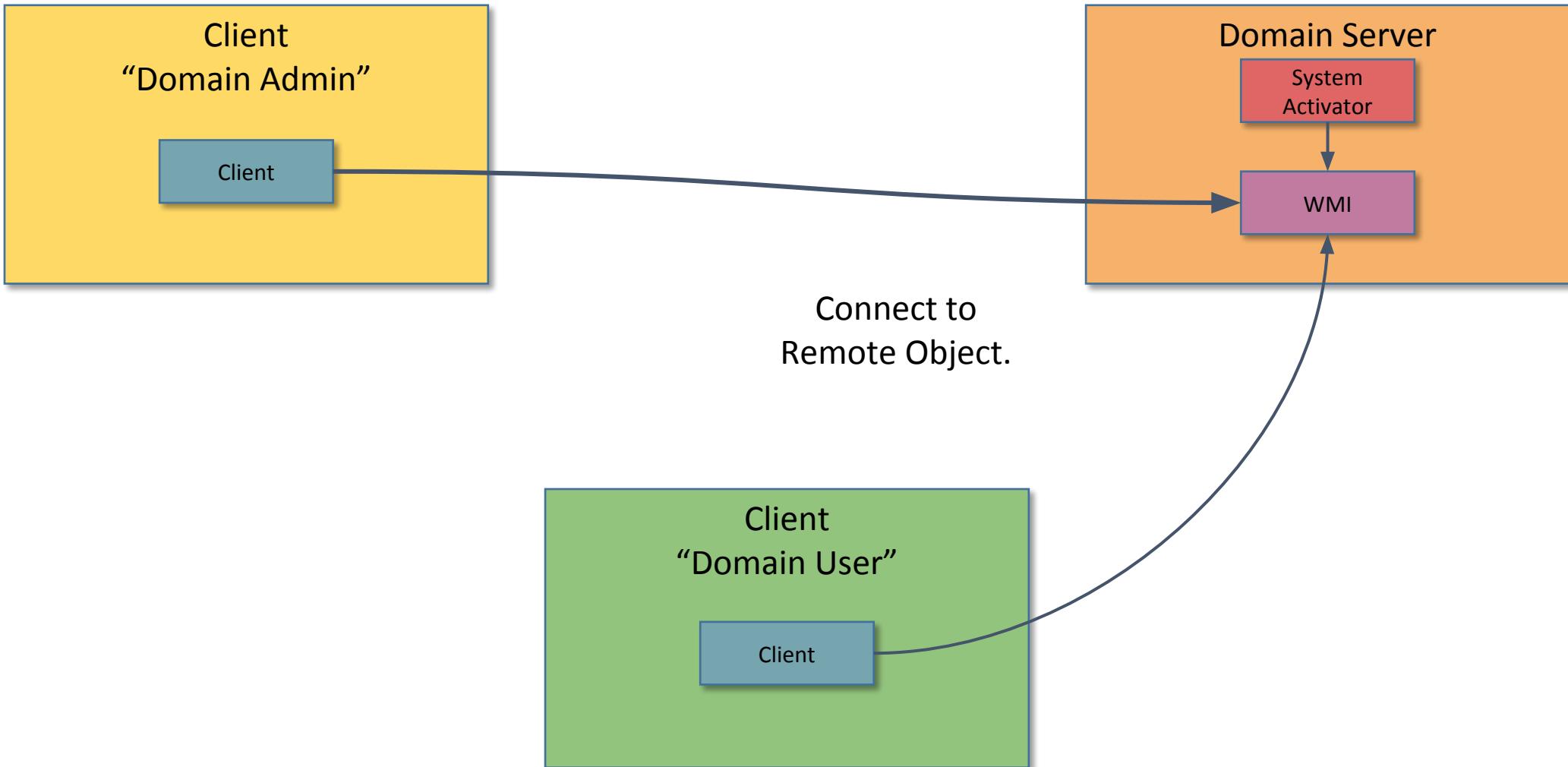
MEOW-Jacking!



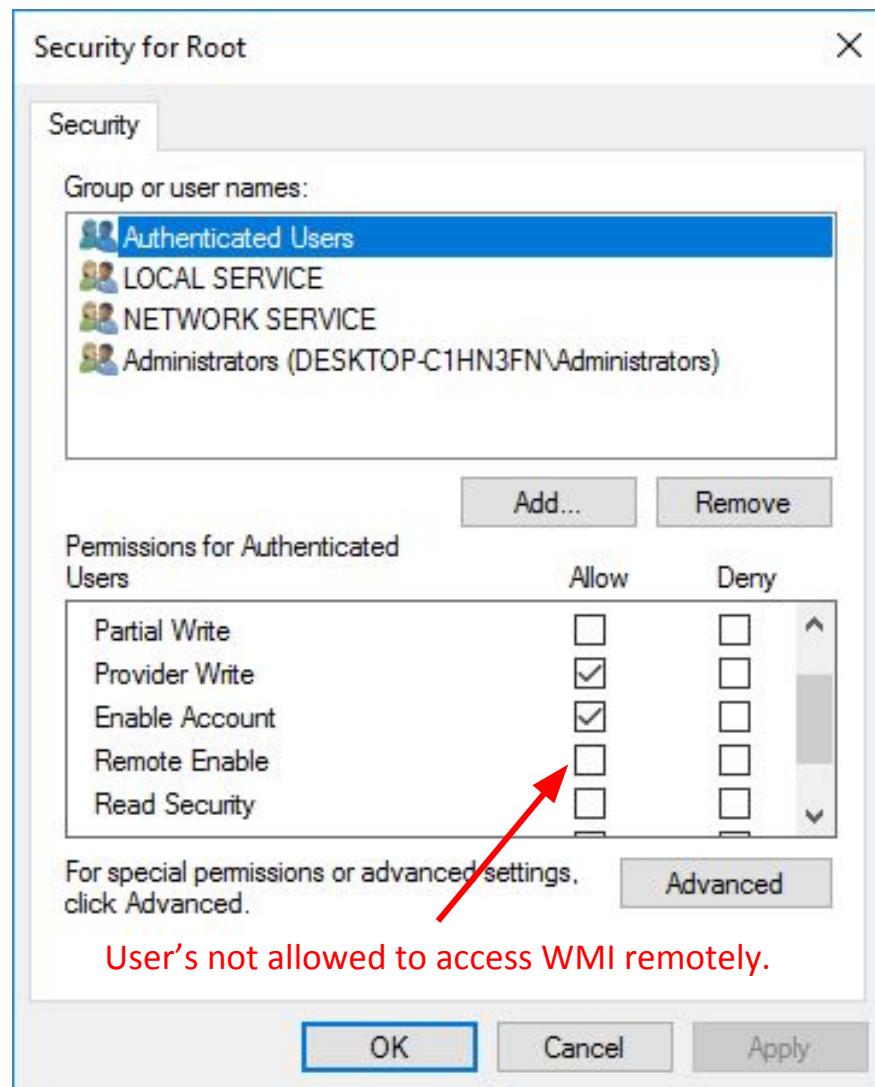
MEOW-Jacking!



MEOW-Jacking!



Not as Bad as it Could Be



The screenshot shows the 'OleViewDotNet - Administrator - 64bit' application. The 'COM Processes' tab is selected. A red box highlights a list of COM objects under the 'Services' category, including Delivery Optimization, Geolocation Service, User Profile Service, Shell Hardware Detection, Windows Management Instrumentation, Windows Push Notifications System Service, and Windows Update. To the right of this list, the text 'Plenty of other potential COM objects though.' is displayed in red. At the bottom of the application window, the text 'Showing 29 of 29 entries' is visible.

File Registry Object Security Help

COM Processes

Filter: Mode: Contains Apply

8 - POWERPNT - DESKTOP-C1HN3FN\tyranid
92 - svchost - NT AUTHORITY\LOCAL SERVICE
424 - svchost - NT AUTHORITY\SYSTEM

Services

Delivery Optimization
Geolocation Service
User Profile Service
Shell Hardware Detection
Windows Management Instrumentation
Windows Push Notifications System Service
Windows Update

IPID: 0000AC00-01A8-0000-3C84-5EFC3F736181 - IID: IRundown
IPID: 0000BC01-01A8-0000-5099-8B16DA72B808 - IID: ILocalSystemActivator
IPID: 00001C02-01A8-0000-9ABD-1E5A8512A0C6 - IID: ILocalSystemActivator
IPID: 00002C03-01A8-FFFF-5418-B03A5E682A4F - IID: IRundown
IPID: 0000E404-01A8-0000-B6EB-112DDEE18C4A - IID: ILocalSystemActivator
IPID: 0000CC05-01A8-0000-845A-67026922296E - IID: ILocalSystemActivator
IPID: 0000B406-01A8-0000-A878-8DC9BEEFB1C - IID: ILocalSystemActivator
IPID: 00001807-01A8-0000-80BC-EBDF64F1D4BE - IID: INotifyNetworkInterfaceEvents
IPID: 00003008-01A8-0000-C34B-ABB9AEDE8617 - IID: IUnknown
TBD: 00000000 01A8 0000 1ABC 7E5F00E1A0C5 TTD: TUnknown

Showing 29 of 29 entries



DEMO TIME



**Thanks for Listening
Any Questions?**