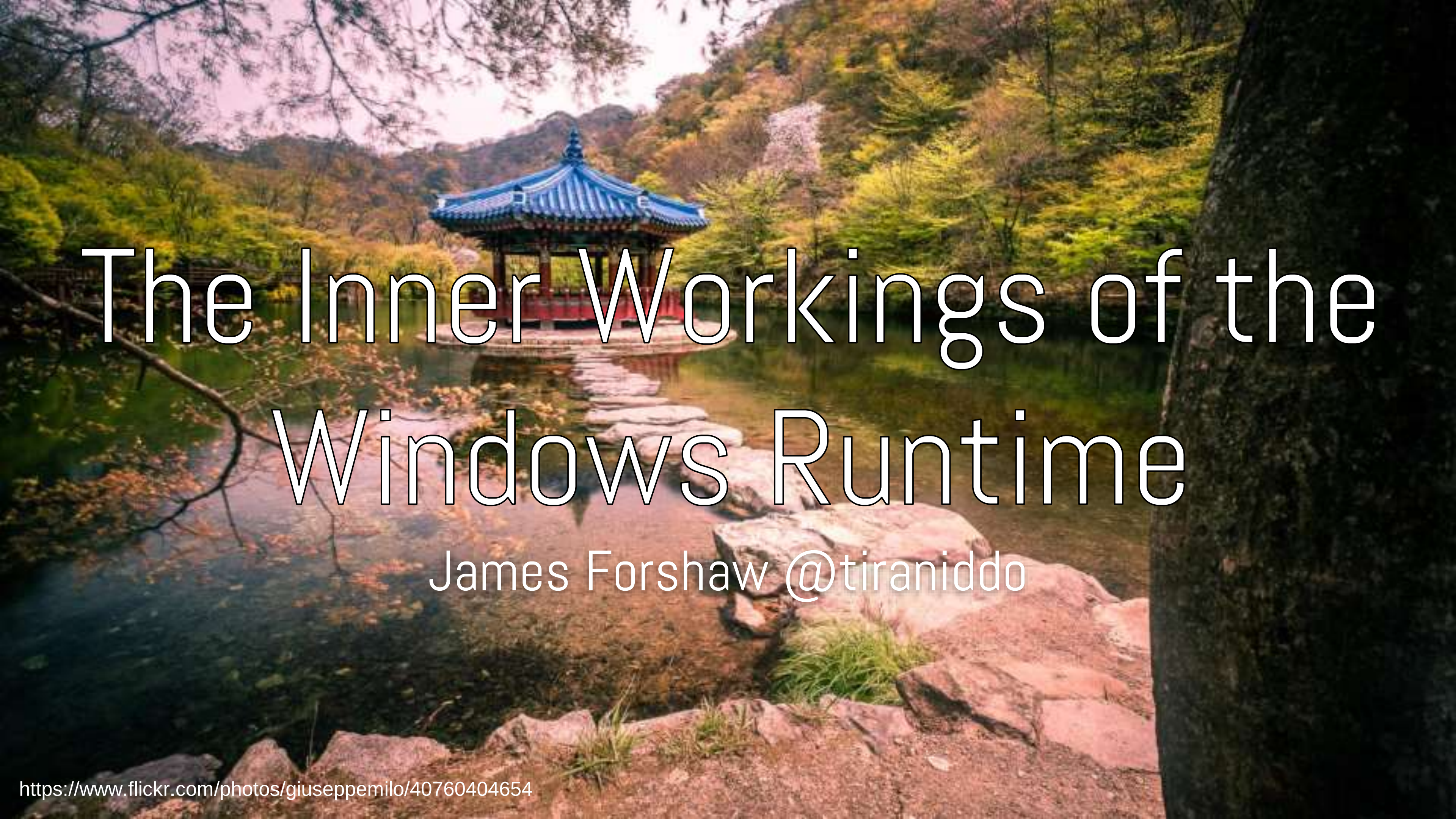


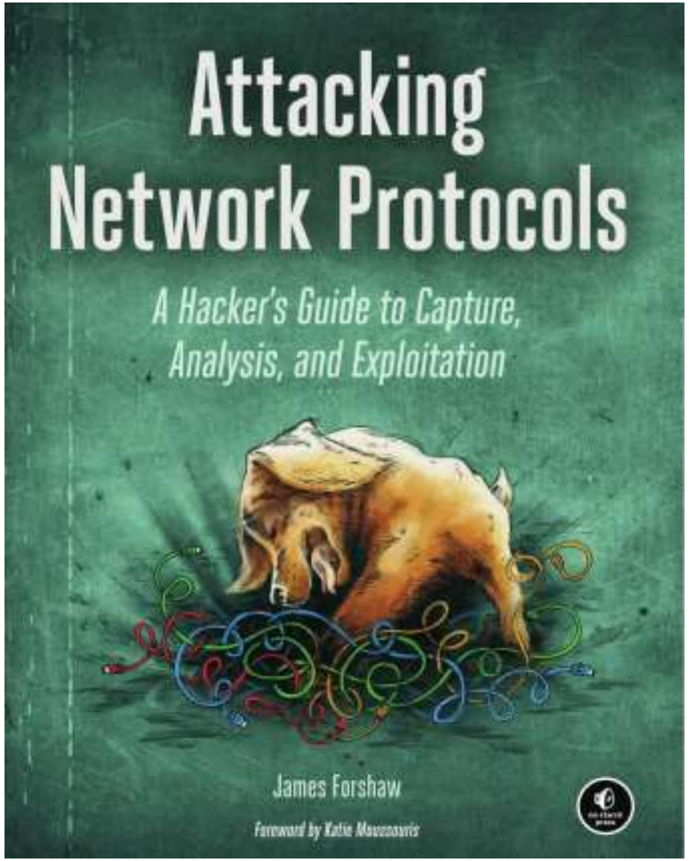
The Inner Workings of the Windows Runtime

James Forshaw @tiraniddo

https://www.flickr.com/photos/giuseppemilo/40760404654

Who am I?

- Researcher in Google's Project Zero
- Specialize in Windows
  - Especially local privilege escalation
  - Logical vulnerability specialist
- Author of a book on attacking network protocols
- @tiraniddo on Twitter.


Attacking Network Protocols
A Hacker's Guide to Capture, Analysis, and Exploitation
James Forshaw
Foreword by Katie Moussouris

Why Talk About Windows Runtime?

Understand the Technology

Aid to Reverse Engineering

Improve Security Research

Background Research

Demystifying COM

James Forshaw (@tiraniddo)

Troopers 2017

https://www.troopers.de/downloads/troopers17/TR17_Demystifying_%20COM.pdf

Windows RunTime

Hack In The Box 2012

Sébastien RENAUD srenaud@quarkslab.com

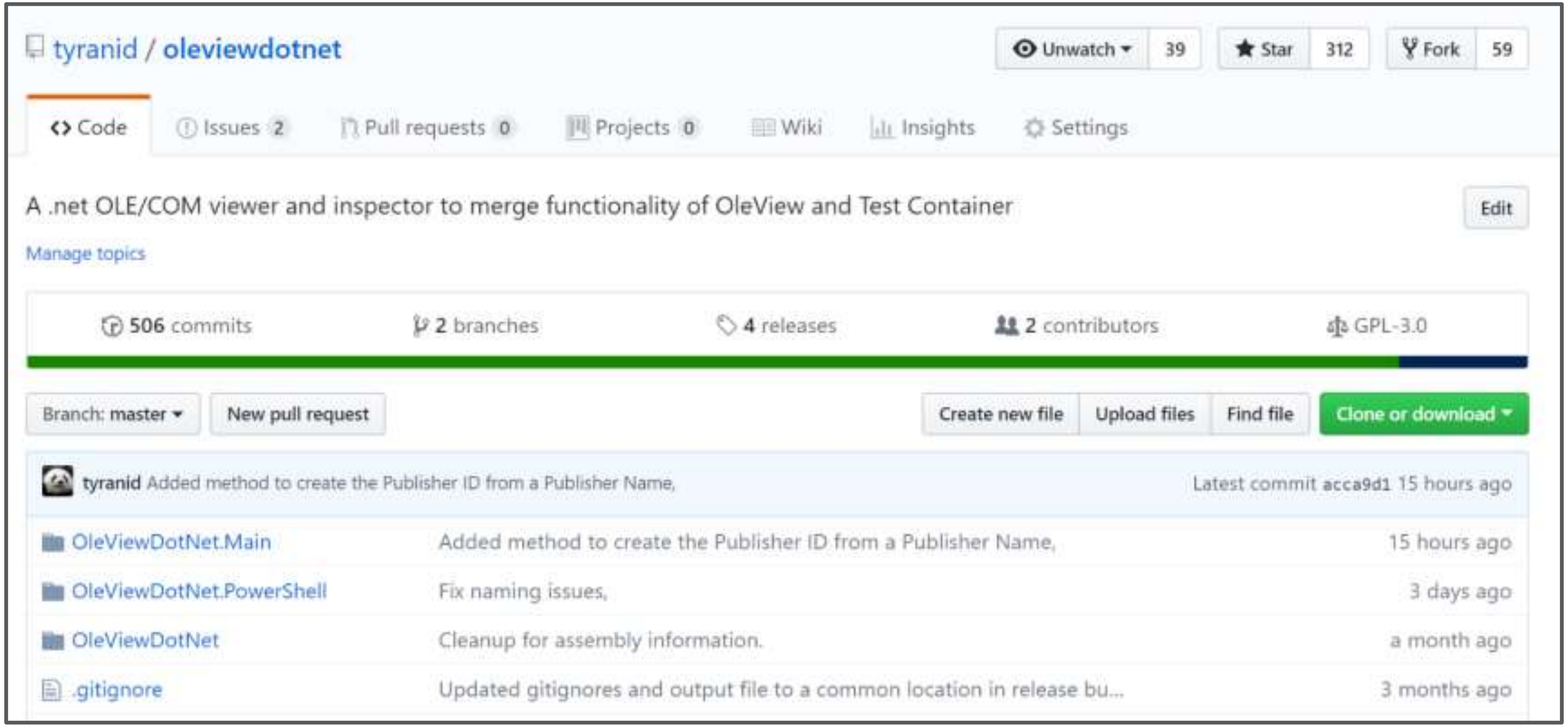
Kévin SZKUDLAPSKI kszkudlapski@quarkslab.com

QUARKSLAB INNOVATIVE SECURITY

https://conference.hitb.org/hitbsecconf2012ams/materials/D1T2%20-%20Sebastien%20Renaud%20and%20Kevin%20Szkudlapski%20-%20WinRT.pdf

This Talk is based on Windows 10 **1803/1809**

OleViewDotNet


tyranid / oleviewdotnet

A .net OLE/COM viewer and inspector to merge functionality of OleView and Test Container

506 commits | 2 branches | 4 releases | 2 contributors | GPL-3.0

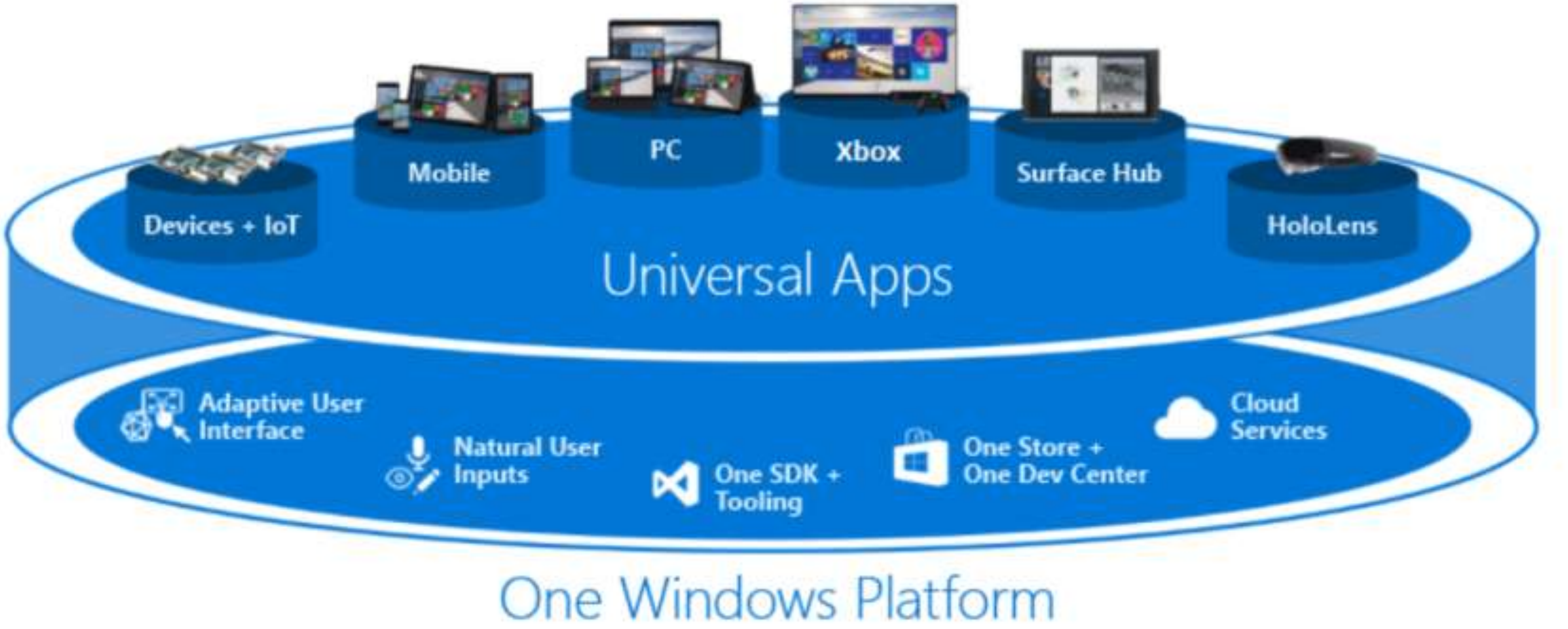
tyranid Added method to create the Publisher ID from a Publisher Name. Latest commit acca9d1 15 hours ago

OleViewDotNet.Main	Added method to create the Publisher ID from a Publisher Name.	15 hours ago
OleViewDotNet.PowerShell	Fix naming issues.	3 days ago
OleViewDotNet	Cleanup for assembly information.	a month ago
.gitignore	Updated gitignores and output file to a common location in release bu...	3 months ago

https://github.com/tyranid/oleviewdotnet

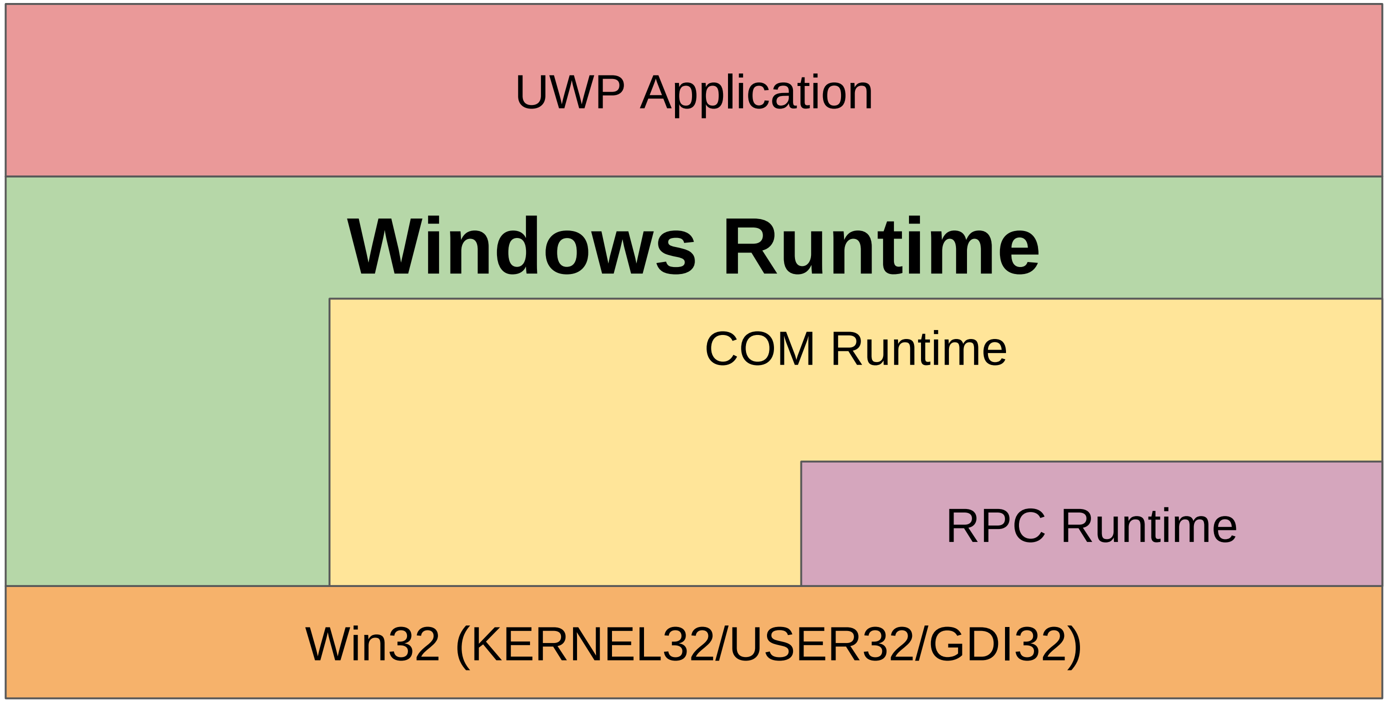
What’s the Universal Windows Platform (UWP)?

05/07/2018 • 10 minutes to read • Contributors all

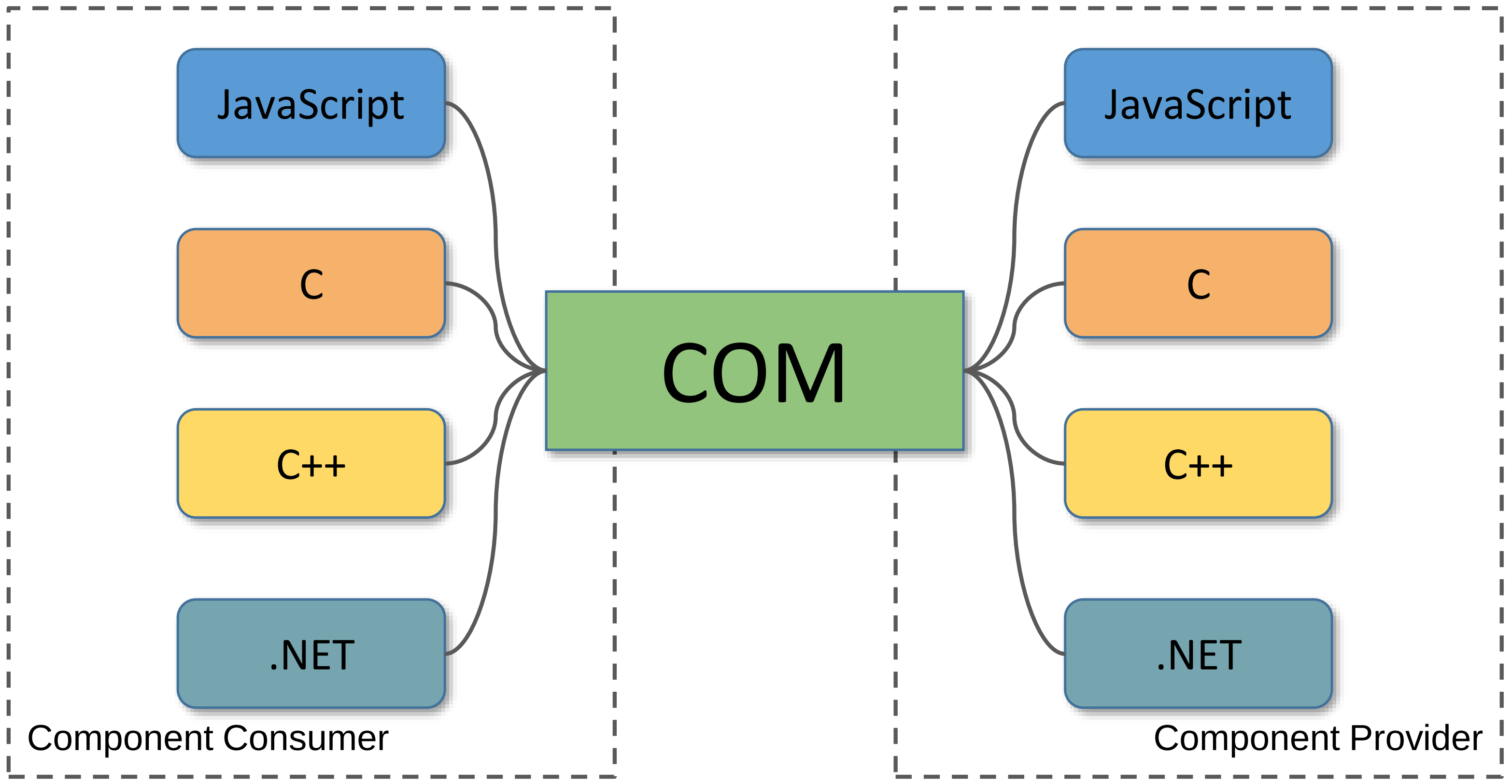


https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide

What’s the Windows Runtime (WinRT)?



COM Joins Everything Together



IInspectable the New Root of Evil

```
MIDL_INTERFACE("AF86E2E0-B12D-4c6a-9C5A-D7AA65101E90")
IInspectable : public IUnknown {
public:
    HRESULT GetIids(
        ULONG *iidCount,
        IID **iids);

    HRESULT GetRuntimeClassName(
        HSTRING *className);

    HRESULT GetTrustLevel(
        TrustLevel *trustLevel);
};
```

Get a list of interface IDs supported by class.

Get class name.

Get class trust level.

String Handles (HSTRING)

```
typedef struct HSTRING__{
    int unused;
} HSTRING__;

// Declare the HSTRING handle for C/C++
typedef HSTRING__* HSTRING;
```

Opaque string handle structure.

```
WindowsCreateString(
  PCNZWCH sourceString,
  UINT32  length,
  HSTRING *string
);
```

Reference counted on the heap.

```
WindowsCreateStringReference(
  PCWSTR         sourceString,
  UINT32         length,
  HSTRING_HEADER *hstringHeader,
  HSTRING        *string
);
```

Scoped on the stack.

The Real HSTRING

```
struct HSTRING_HEADER_INTERNAL {
  WINDOWS_RUNTIME_HSTRING_FLAGS flags;
  unsigned int length;
  unsigned int padding1;
  unsigned int padding2;
  const wchar_t *stringRef;
};
```

Used for stack scoped “reference” strings.

```
struct STRING_OPAQUE {
  HSTRING_HEADER_INTERNAL header;
  volatile int refcount;
  wchar_t string[1];
```

Inline string data and reference count for use on the heap

```
PCWSTR WindowsGetStringRawBuffer(
  HSTRING string,
  UINT32  *length
);
```

Call to get raw buffer and length.

Classic COM to Windows Runtime Functions

Description	***Classic COM***	***Windows Runtime***
Initialize COM Apartment	CoInitializeEx	RoInitialize
Initialize COM Security	CoInitializeSecurity	CoInitializeSecurity
Create Class Instance	CoCreateInstance	RoActivateInstance
Create Class Factory	CoGetClassObject	RoGetActivationFactory
Register Class Factory	CoRegisterClassObject	RoRegisterActivationFactories
Get Class Factory	DllGetClassObject	DllGetActivationFactory

Activation Factories

- Component classes can’t be directly ‘newed’ so WinRT defines a factory interface, *IActivationFactory.* Does not use *IClassFactory*.

```
DEFINE_GUID(IID_ActivationFactory,
     "00000035-0000-0000-C000-000000000046");
struct IActivationFactory : public IUnknown {
  HRESULT ActivateInstance(
    IInspectable **instance
  );
};
```

Activation Factories and Instances

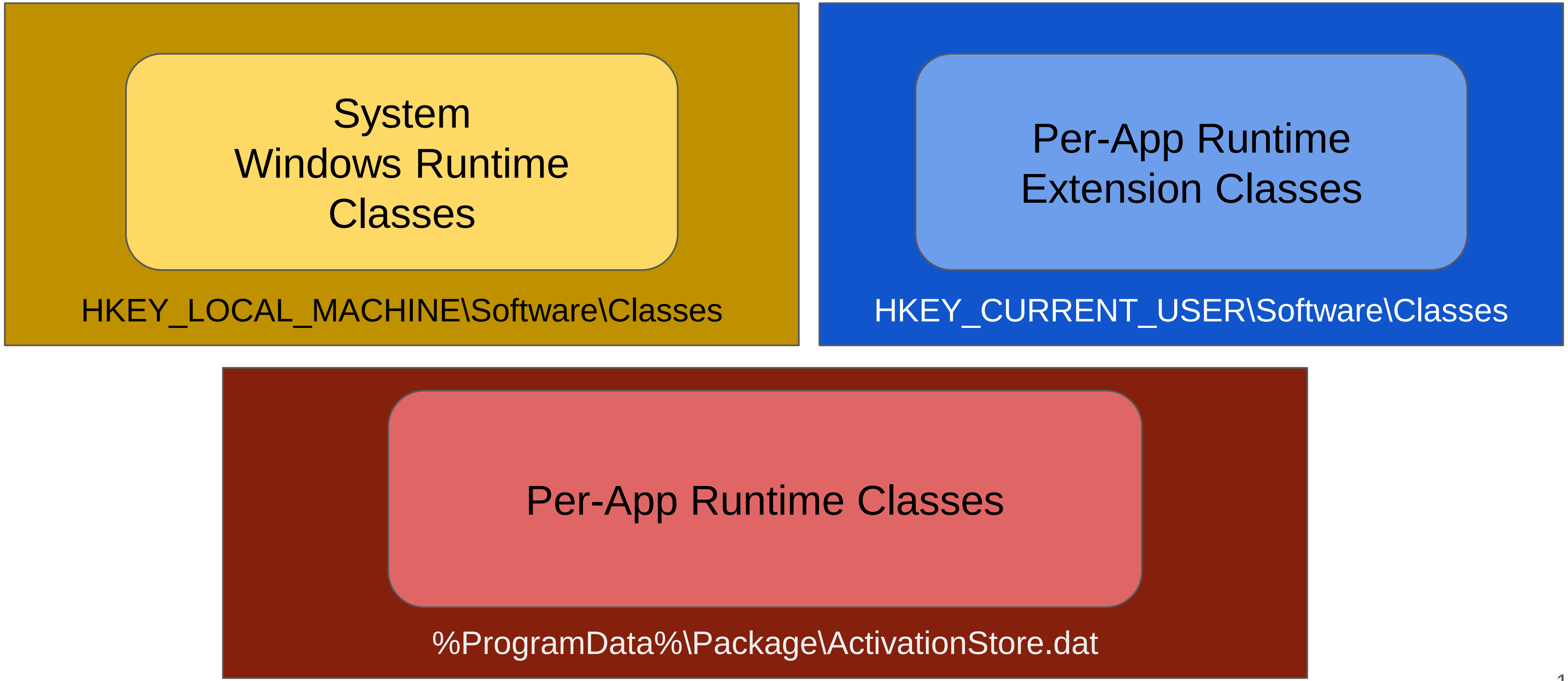
```
HRESULT RoGetActivationFactory(
  HSTRING          activatableClassId,
  REFIID           iid,
  LPVOID*          factory);
```

Abbreviated as ACID

```
HRESULT RoActivateInstance(
  HSTRING          activatableClassId,
  IInspectable**   instance,
);
```

Example ACID: **“Windows.Foundation.Uri”**

Runtime Class Registry Keys

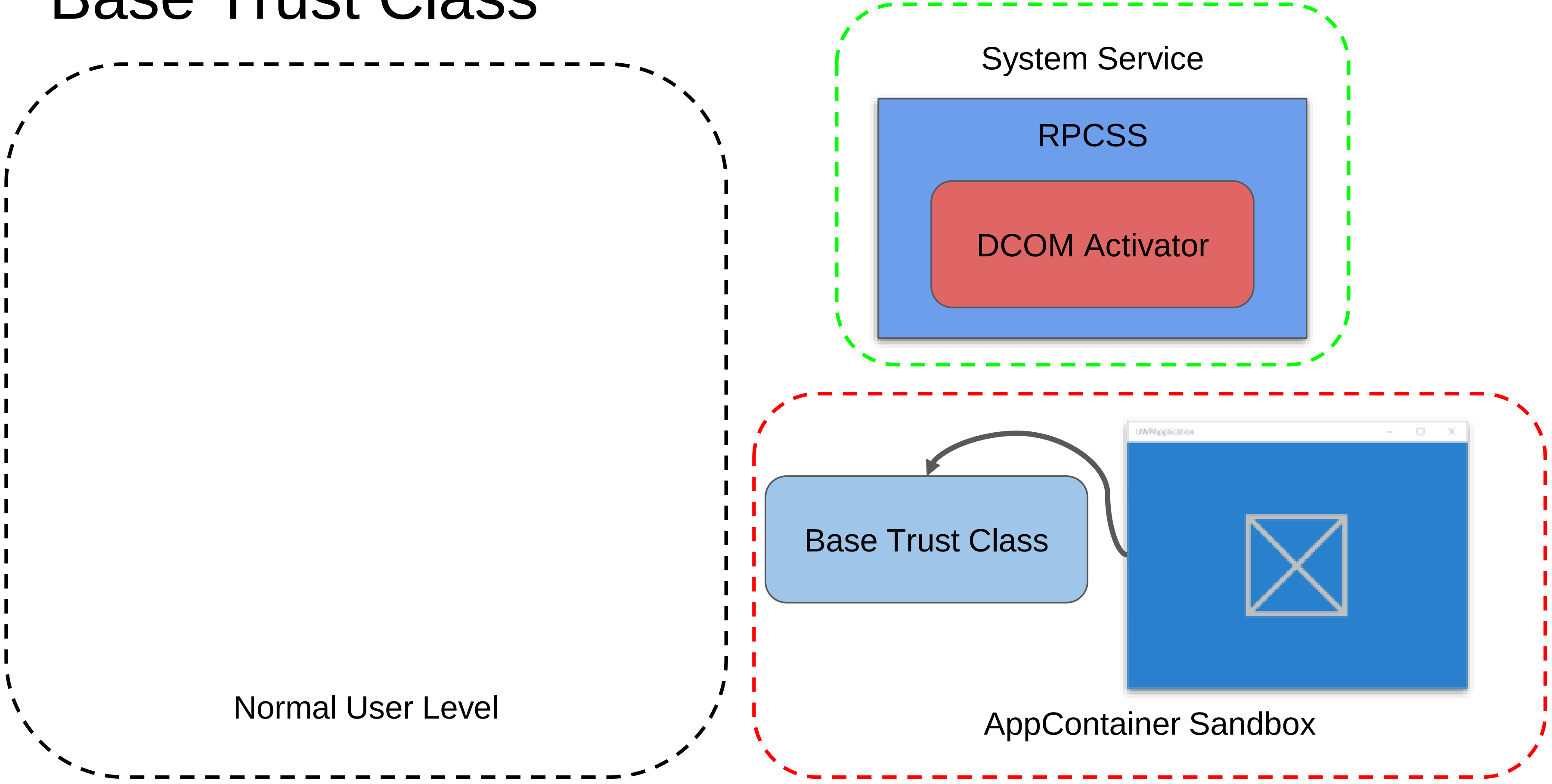

System
Windows Runtime
Classes
HKEY_LOCAL_MACHINE\Software\Classes
Per-App Runtime
Extension Classes
HKEY_CURRENT_USER\Software\Classes
Per-App Runtime Classes
%ProgramData%\Package\ActivationStore.dat

Class Trust Levels

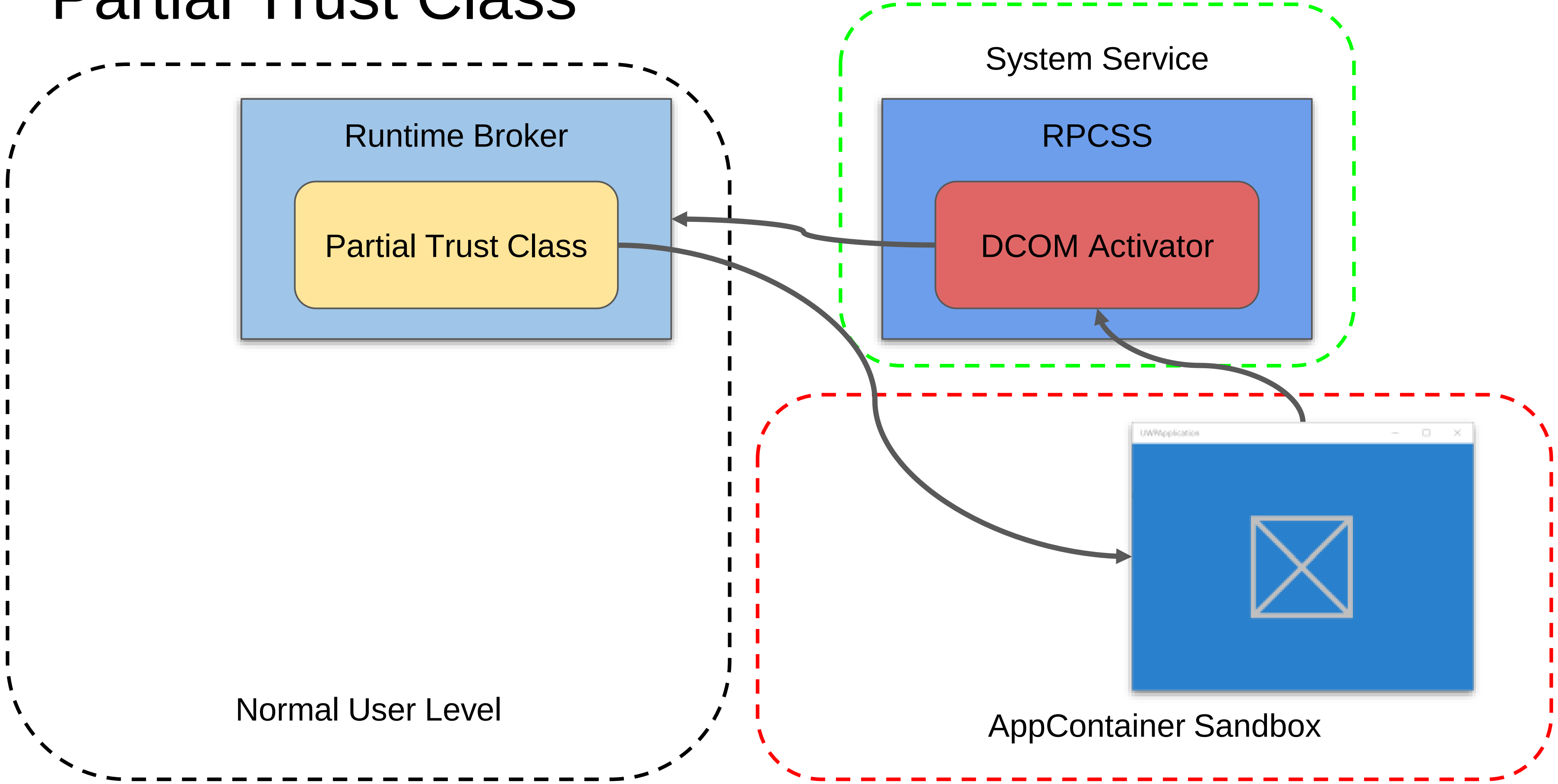
```
HRESULT GetTrustLevel(TrustLevel *trustLevel);
```

Full Trust	Can only be created in a fully trusted context
Partial Trust	Can be created in a sandbox context through a broker
Base Trust	Can be created in any context

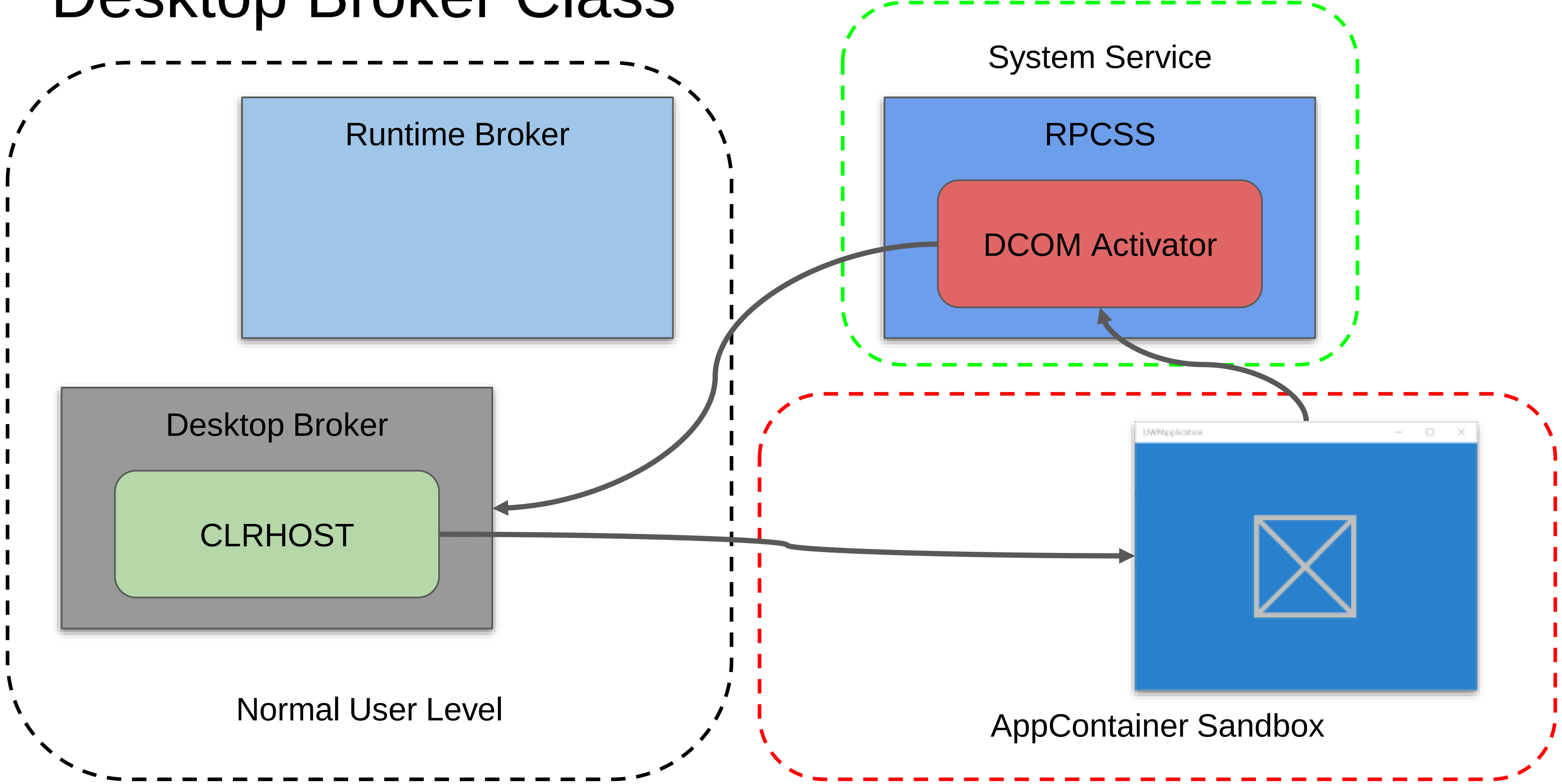
Base Trust Class



Partial Trust Class



Desktop Broker Class



DEMO 1

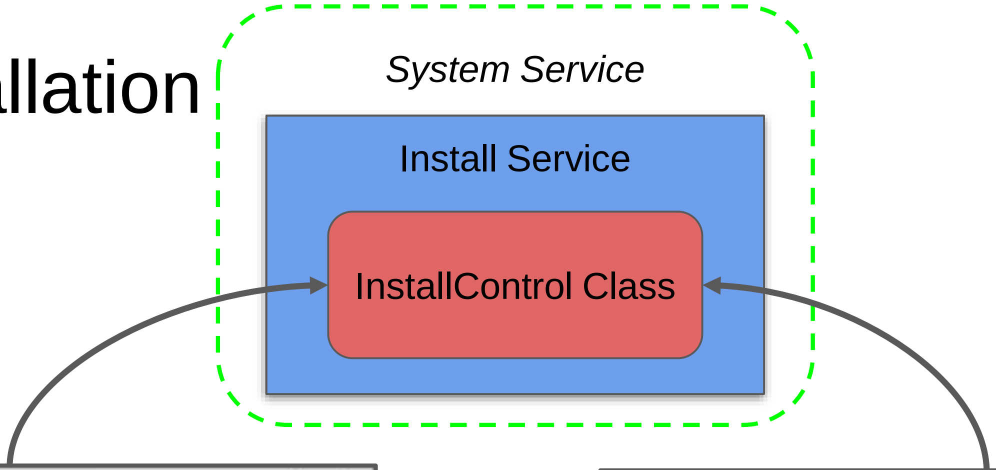
Application Manifest XML

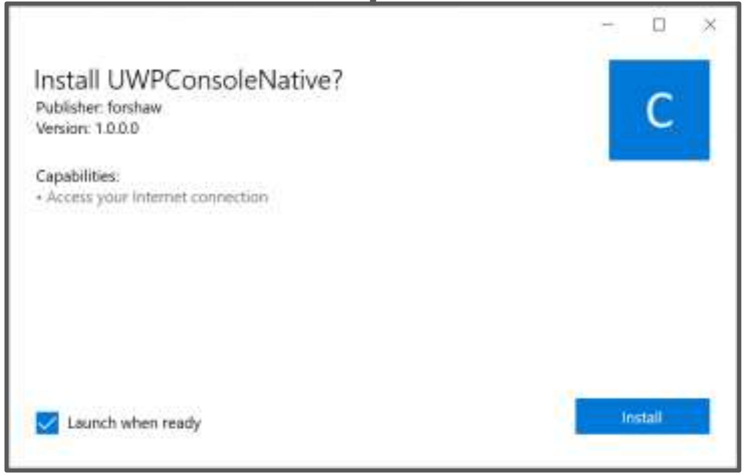
Package
Identity

```
<Package>
    <Identity Name="Microsoft.MicrosoftEdge"
              Publisher="CN=Microsoft Corporation, ..."
              Version="44.17763.1.0"
              ProcessorArchitecture="neutral"/>
    <Applications>
        <Application Id="MicrosoftEdge"
                     Executable="MicrosoftEdge.exe"
                     EntryPoint="MicrosoftEdge.App">
            ...
        </Application>
    </Applications>
</Package>
```

Application
Launch

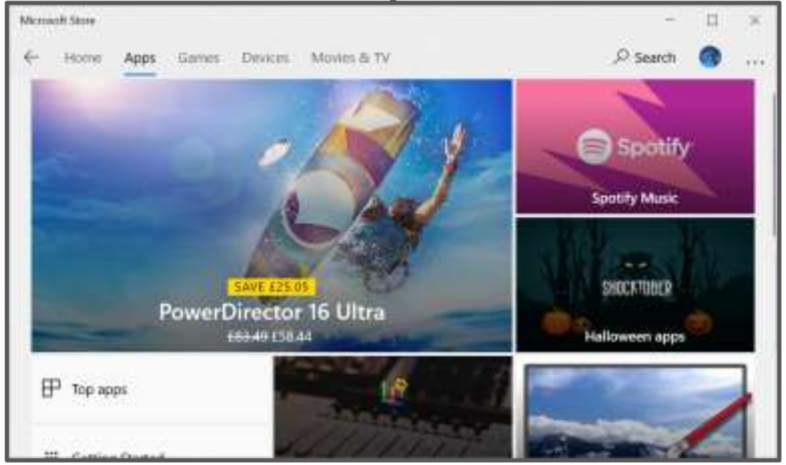
AppX Installation




Install UWPConsoleNative?
Publisher: forshaw
Version: 1.0.0.0

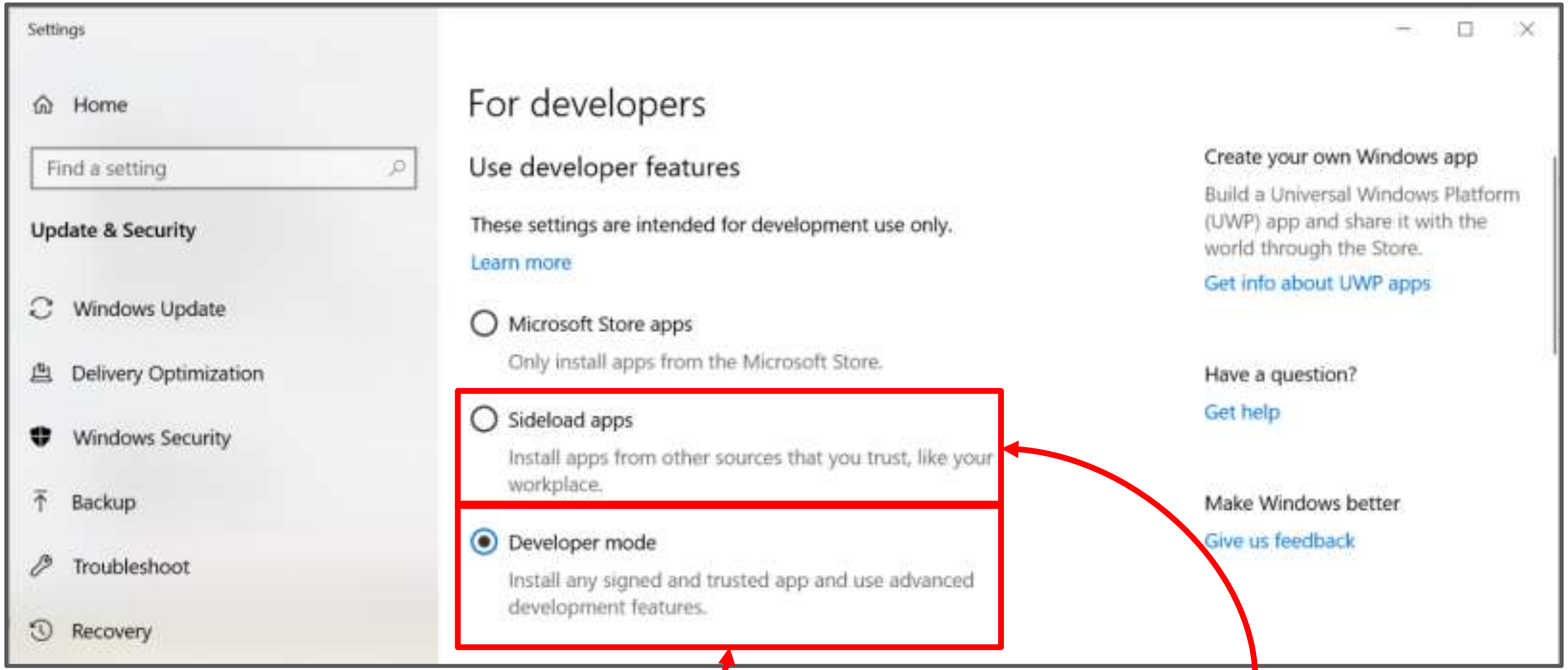
Capabilities:
• Access your Internet connection

Side-loading



Windows Store

Developer Mode and Sideloading


Settings

Home

Find a setting

Update & Security

Windows Update

Delivery Optimization

Windows Security

Backup

Troubleshoot

Recovery

For developers

Use developer features

These settings are intended for development use only.

Learn more

Microsoft Store apps

Only install apps from the Microsoft Store.

Sideload apps

Install apps from other sources that you trust, like your workplace.

[✓ Developer mode]

Install any signed and trusted app and use advanced development features.

Create your own Windows app

Build a Universal Windows Platform (UWP) app and share it with the world through the Store.

Get info about UWP apps

Have a question?

Get help

Make Windows better

Give us feedback

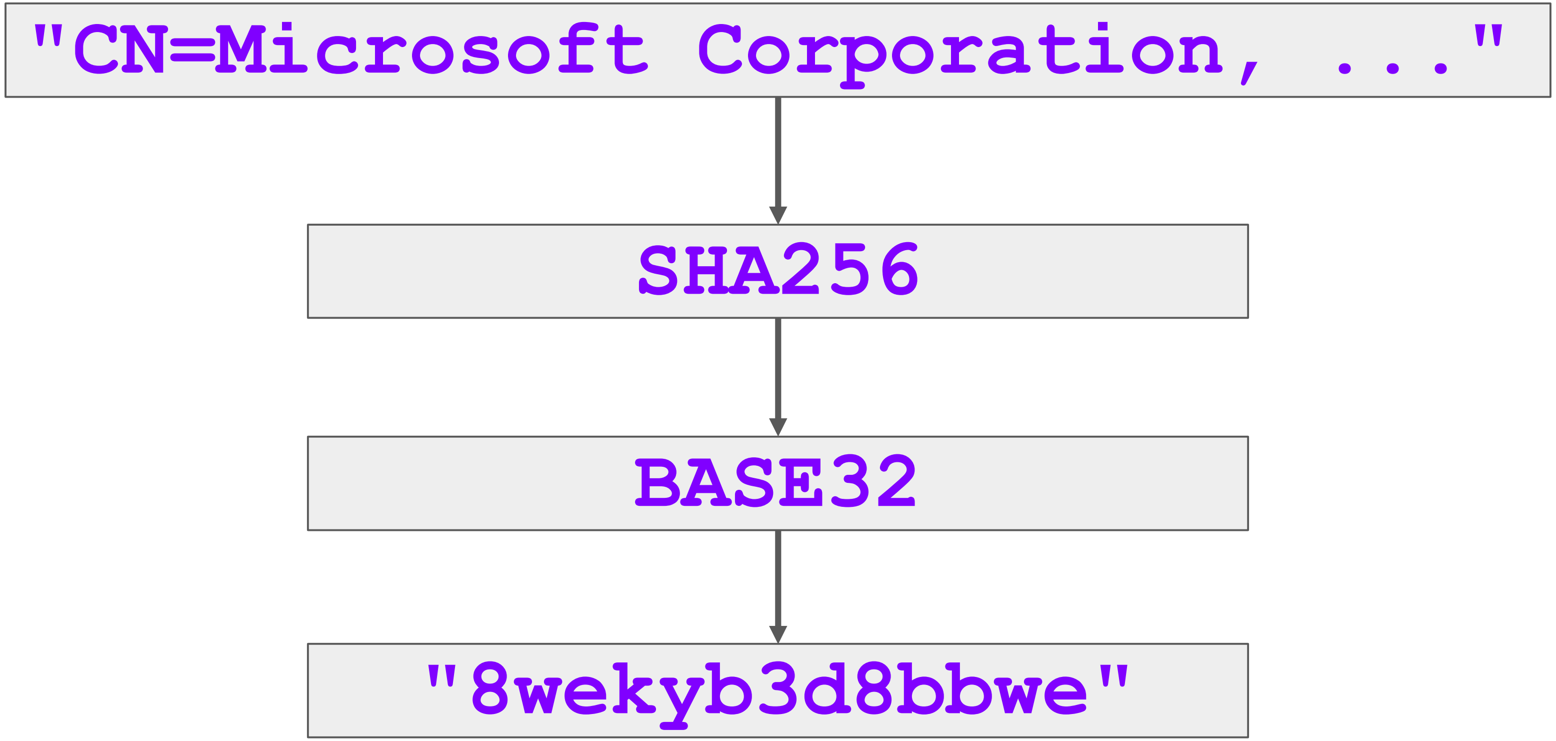
Enable Developer Mode to allow anything to install, even unsigned.

Sideloading allows any signed application from trusted CA root store.

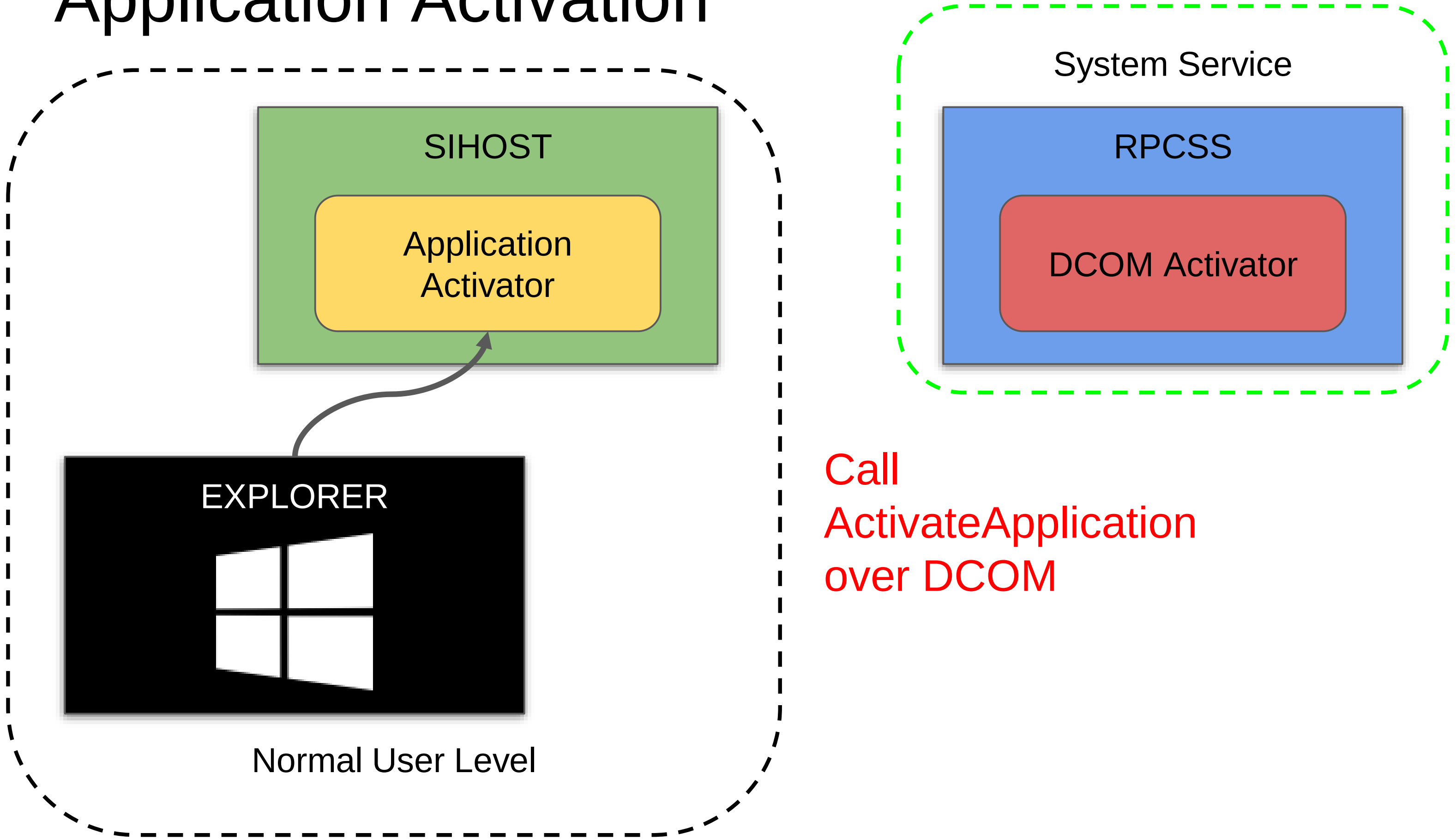
Building the System Application ID

Component	***Example***
Package Name	*Microsoft.MicrosoftEdge*
Publisher ID	*8wekyb3d8bbwe*
Package Family Name	*Microsoft.MicrosoftEdge_8wekyb3d8bbwe*
Package Full Name	*Microsoft.MicrosoftEdge_44.17763.1.0_neutral__8wekyb3d8bbwe*
Package Moniker	Same as Package Full Name
Package-Relative App ID	*App*
Application User Model ID	*Microsoft.MicrosoftEdge_8wekyb3d8bbwe!App*

Publisher to Publisher ID

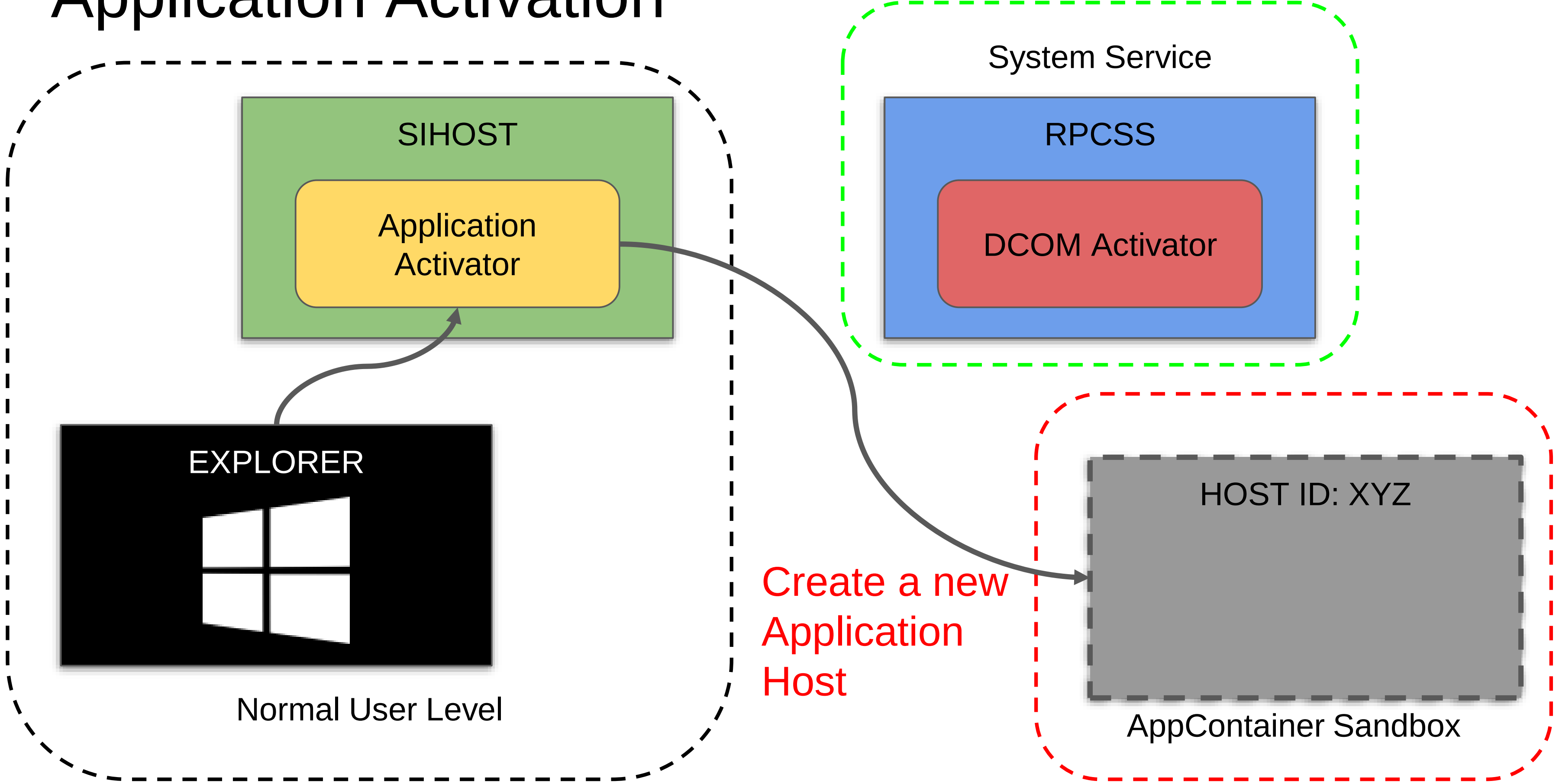


Application Activation

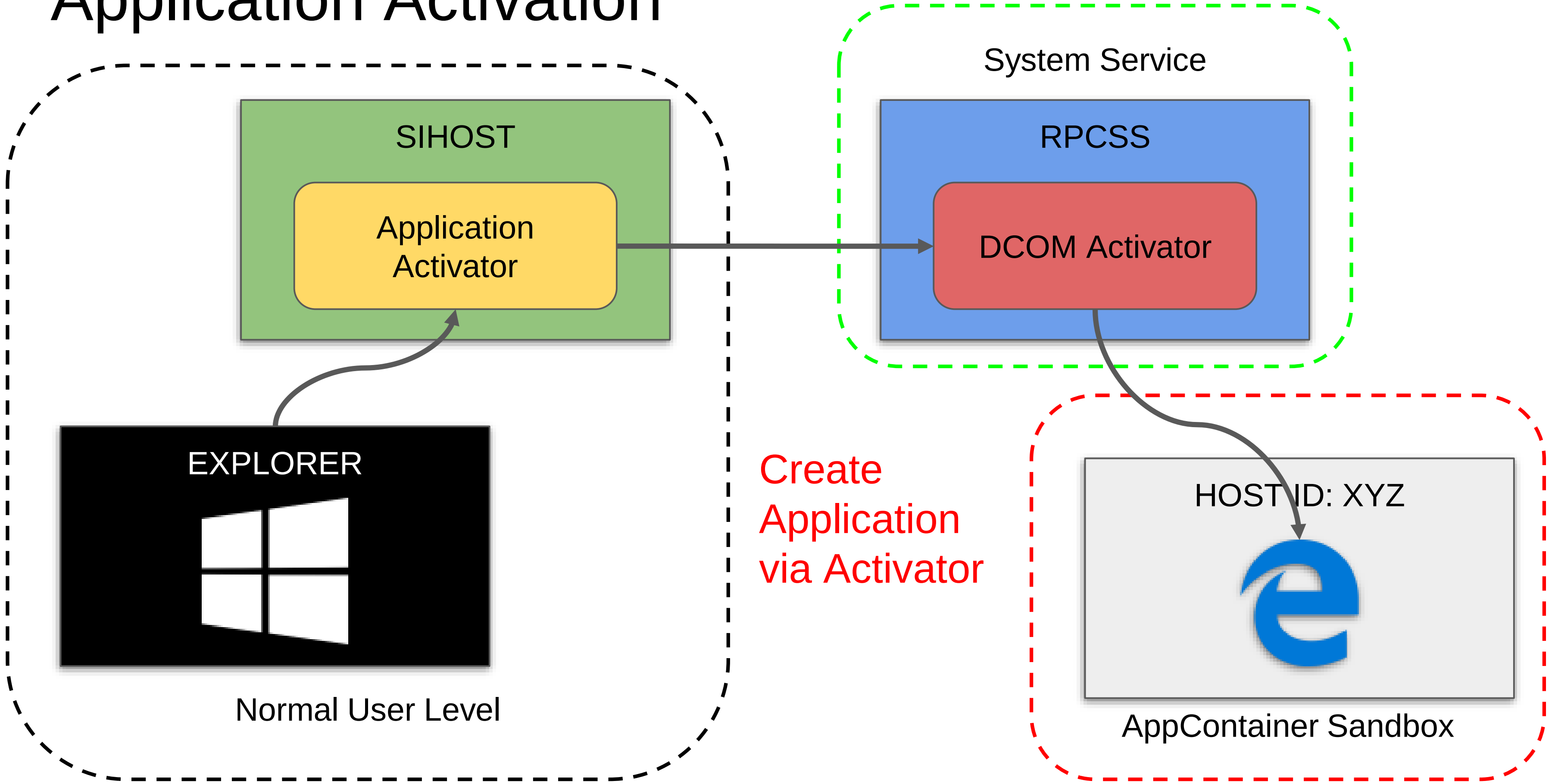


Call
ActivateApplication
over DCOM

Application Activation



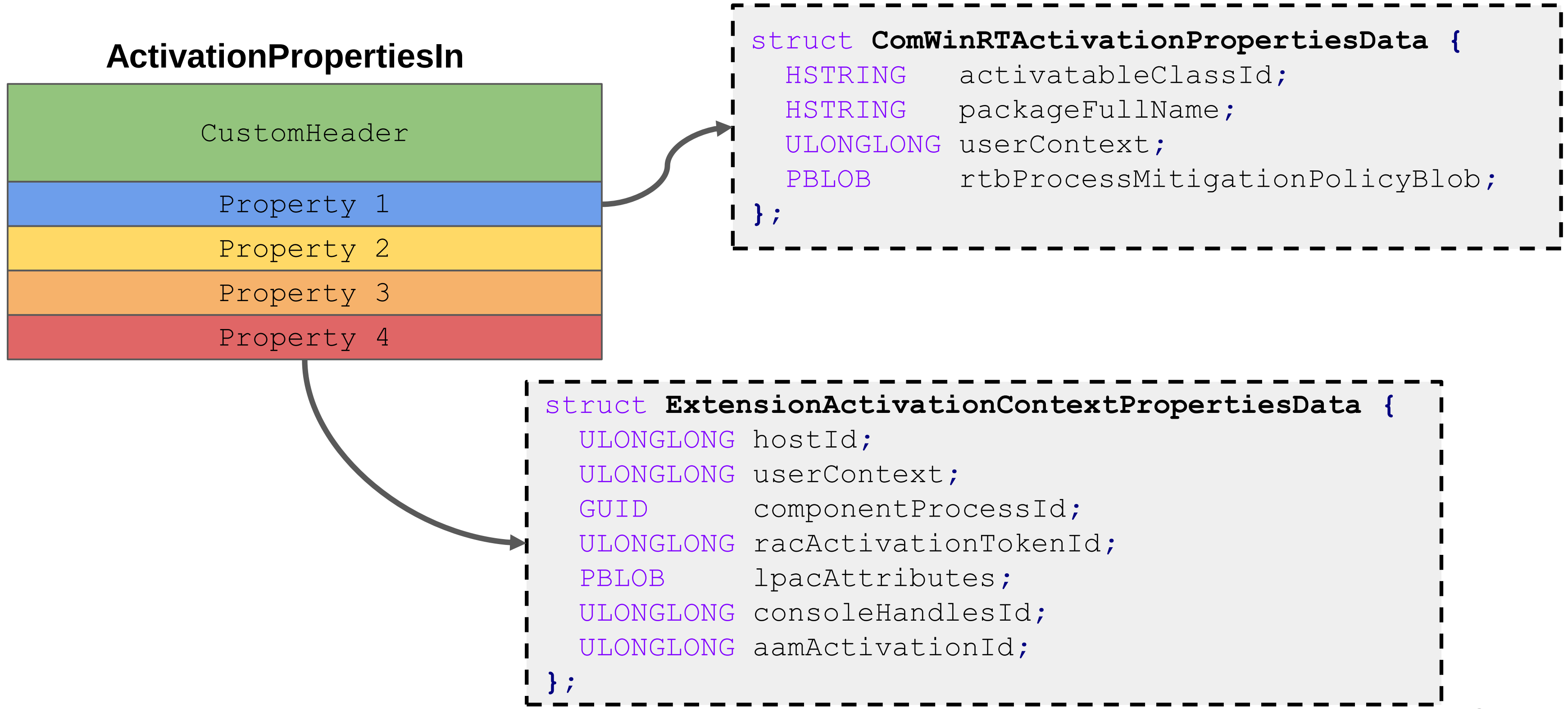
Application Activation



DCOM 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);
};
```

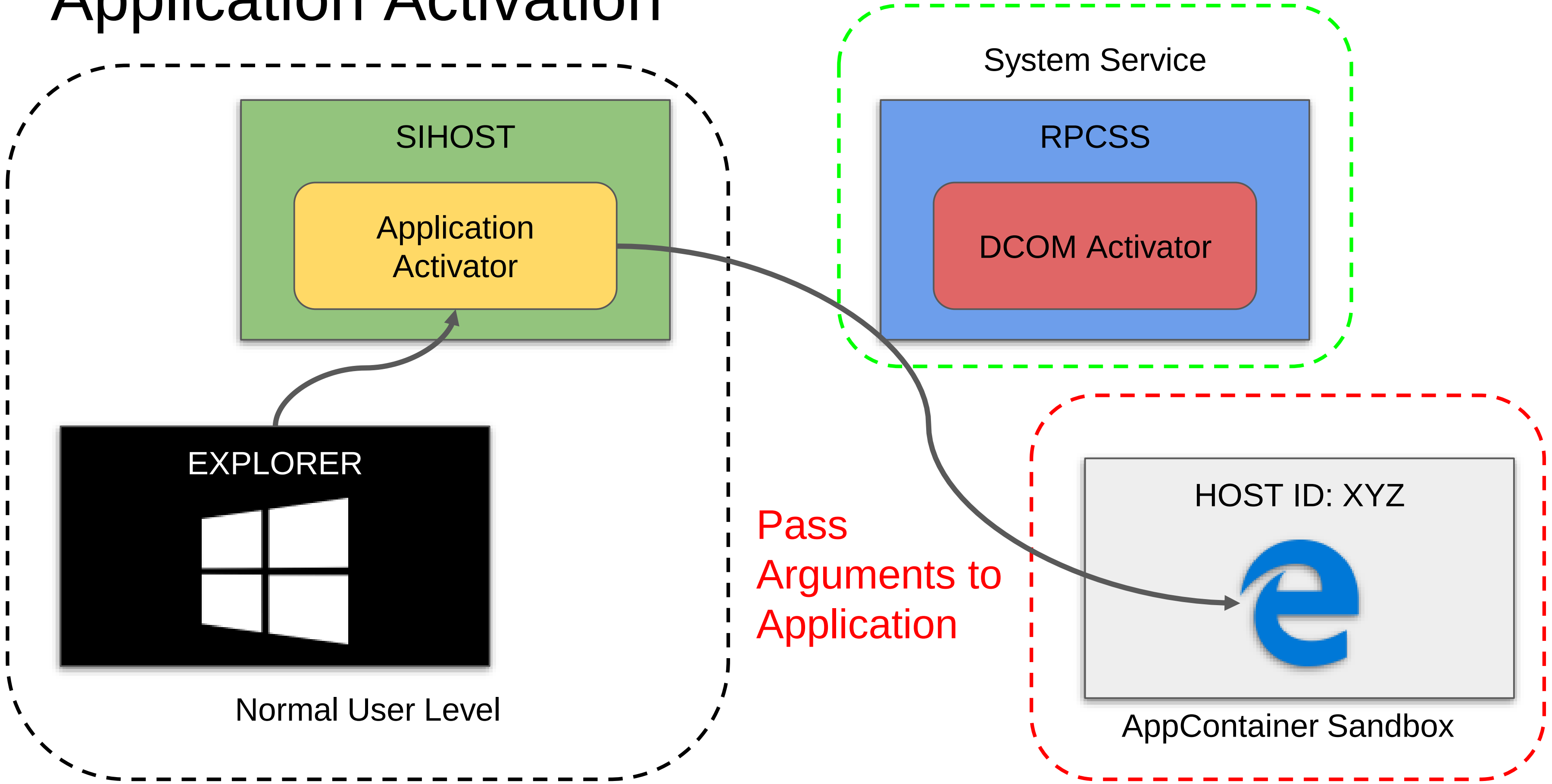
WinRT Activation Properties



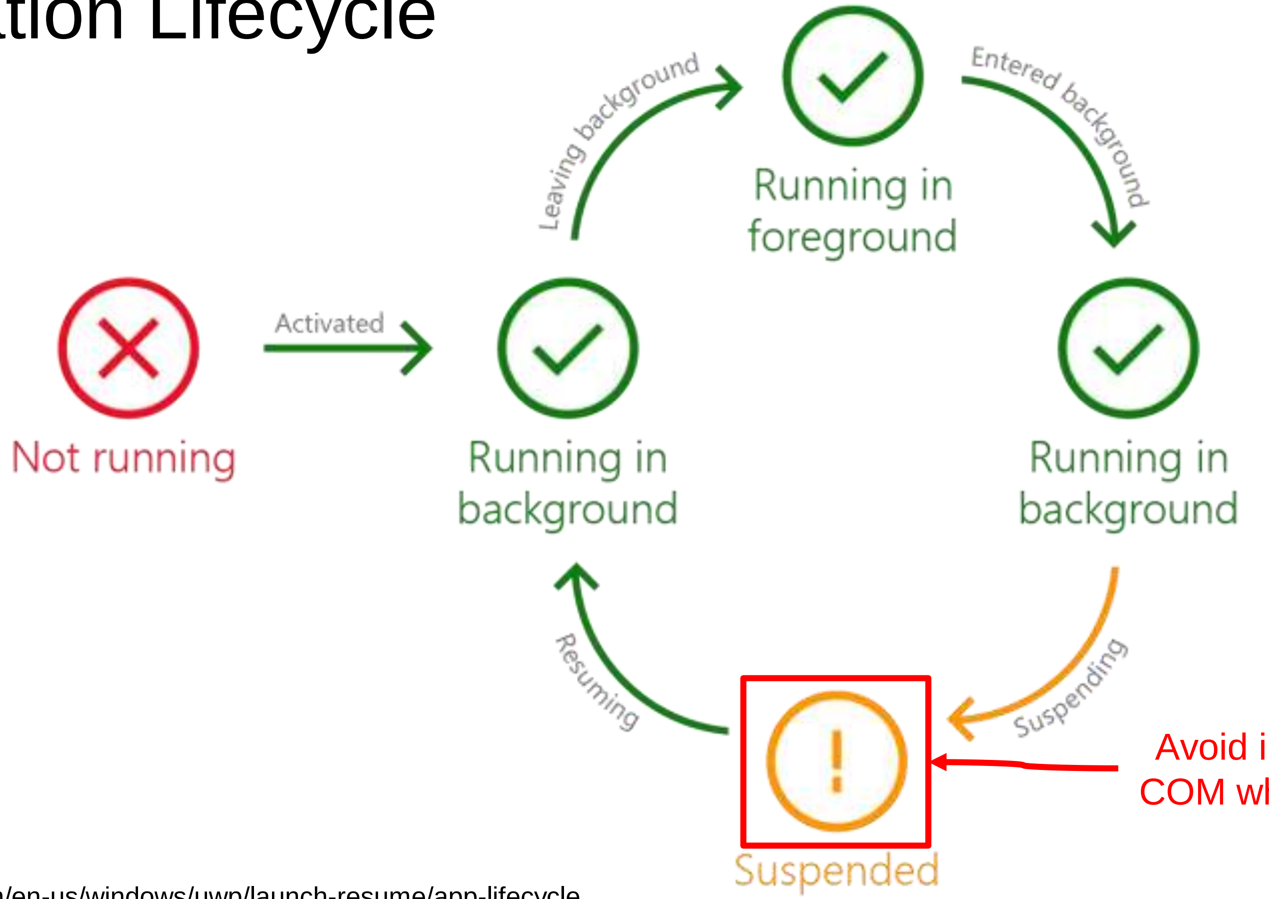
```
struct ComWinRTActivationPropertiesData {
  HSTRING   activatableClassId;
  HSTRING   packageFullName;
  ULONGLONG userContext;
  PBLOB     rtbProcessMitigationPolicyBlob;
};
```

```
struct ExtensionActivationContextPropertiesData {
  ULONGLONG hostId;
  ULONGLONG userContext;
  GUID      componentProcessId;
  ULONGLONG racActivationTokenId;
  PBLOB     lpacAttributes;
  ULONGLONG consoleHandlesId;
  ULONGLONG aamActivationId;
};
```

Application Activation



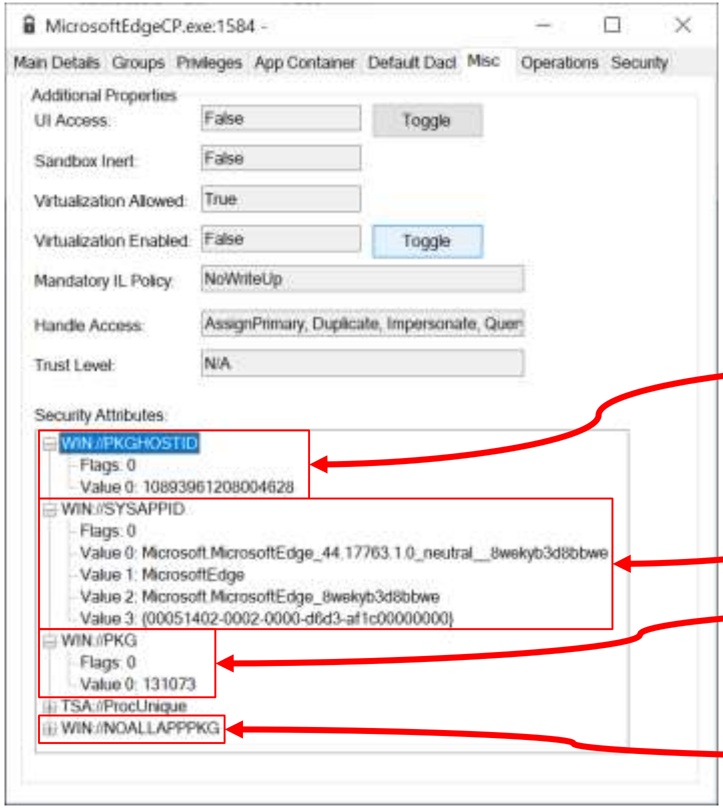
Application Lifecycle



Avoid interacting over COM when suspended.

https://docs.microsoft.com/en-us/windows/uwp/launch-resume/app-lifecycle

AppContainer Access Token Attributes



MicrosoftEdgeCP.exe:1584 -

Additional Properties

UI Access:	False
Sandbox Inert:	False
Virtualization Allowed:	True
Virtualization Enabled:	False
Mandatory IL Policy:	NoWriteUp
Handle Access:	AssignPrimary, Duplicate, Impersonate, Quer
Trust Level:	N/A

Security Attributes:

```
WIN://PKGHOSTID
  Flags: 0
  Value 0: 10893961208004628
WIN://SYSAPPID
  Flags: 0
  Value 0: Microsoft.MicrosoftEdge_44.17763.1.0_neutral__8wekyb3d8bbwe
  Value 1: MicrosoftEdge
  Value 2: Microsoft.MicrosoftEdge_8wekyb3d8bbwe
  Value 3: {00051402-0002-0000-d6d3-af1c00000000}
WIN://PKG
  Flags: 0
  Value 0: 131073
TSA://ProcUnique
WIN://NOALLAPPPKG
```

Caller needs SeTcbPrivilege to add or modify security attributes.

Application Host ID

System Application ID

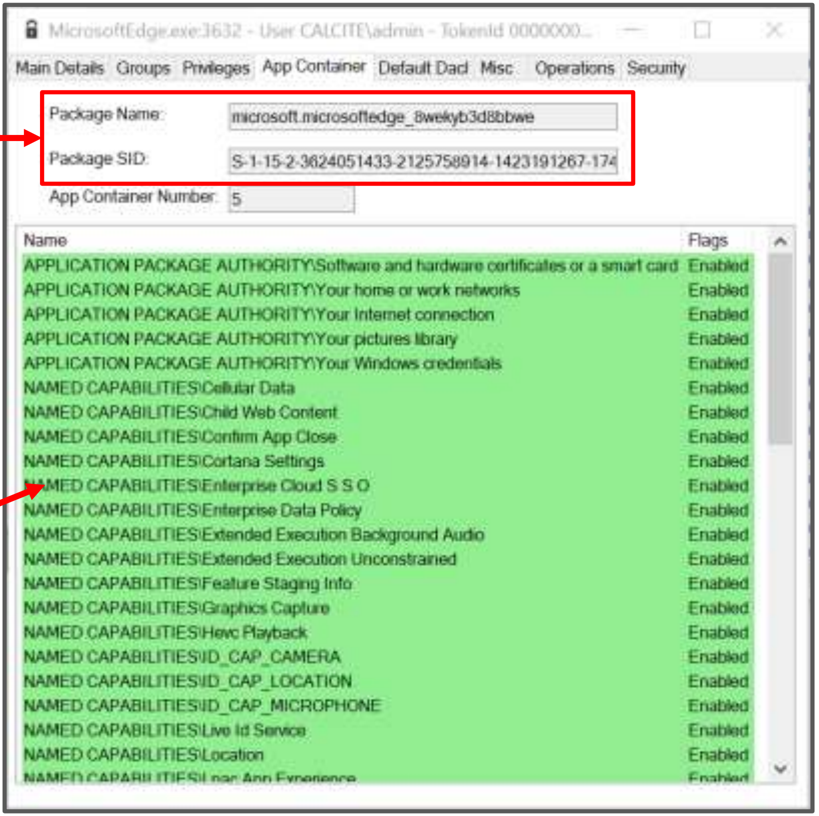
Package Flags

Low Privilege App Container

AppContainer SID and Capabilities

Package Family Name and Package SID

```
<Capabilities>
 <Capability Name="internetClient"/>
 <Capability Name="privateNetworkClientServer"/>
 <rescap:Capability Name="childWebContent"/>
 <rescap:Capability Name="confirmAppClose"/>
 <rescap:Capability Name="lpacCom"/>
 ...
 <DeviceCapability Name="location"/>
 <DeviceCapability Name="microphone"/>
 <DeviceCapability Name="webcam"/>
</Capabilities>
```



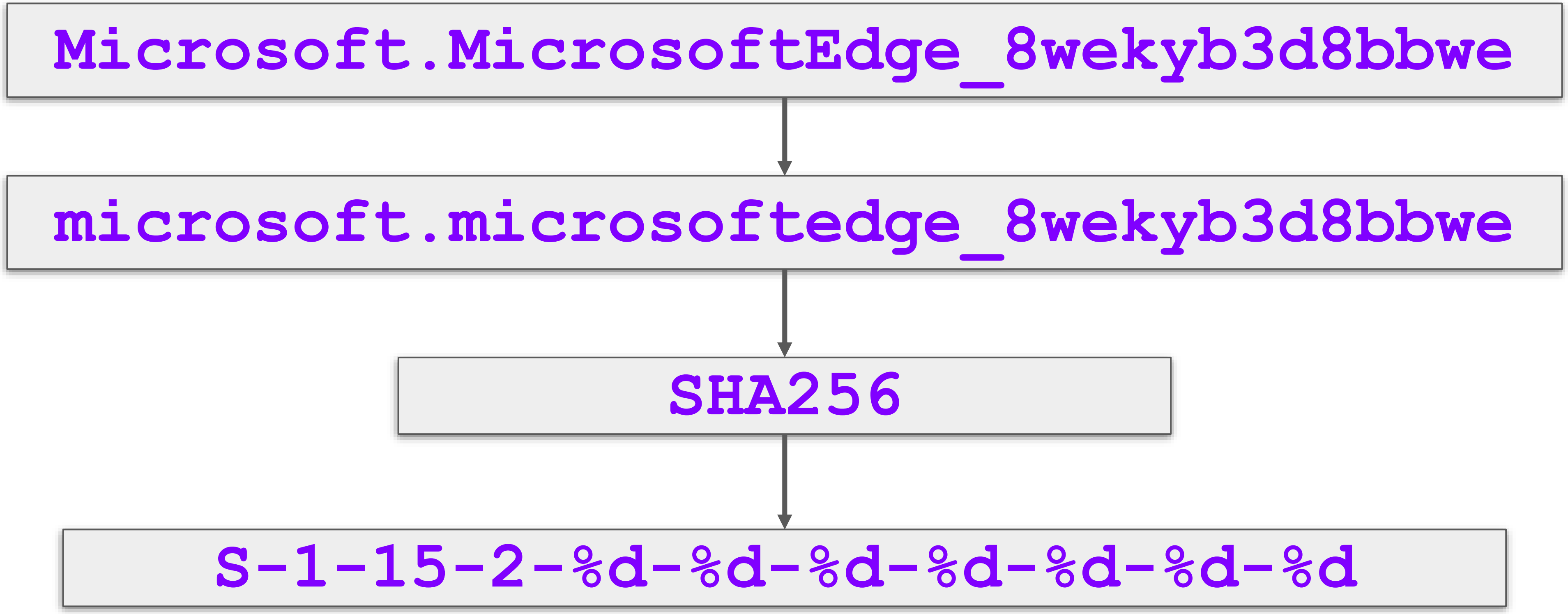
Package Name: microsoft.microsoftedge_8wekyb3d8bbwe

Package SID: S-1-15-2-3624051433-2125758914-1423191267-174

App Container Number: 5

Name	Flags
APPLICATION PACKAGE AUTHORITY\Software and hardware certificates or a smart card	Enabled
APPLICATION PACKAGE AUTHORITY\Your home or work networks	Enabled
APPLICATION PACKAGE AUTHORITY\Your Internet connection	Enabled
APPLICATION PACKAGE AUTHORITY\Your pictures library	Enabled
APPLICATION PACKAGE AUTHORITY\Your Windows credentials	Enabled
NAMED CAPABILITIES\Cellular Data	Enabled
NAMED CAPABILITIES\Child Web Content	Enabled
NAMED CAPABILITIES\Confirm App Close	Enabled
NAMED CAPABILITIES\Cortana Settings	Enabled
NAMED CAPABILITIES\Enterprise Cloud S S O	Enabled
NAMED CAPABILITIES\Enterprise Data Policy	Enabled
NAMED CAPABILITIES\Extended Execution Background Audio	Enabled
NAMED CAPABILITIES\Extended Execution Unconstrained	Enabled
NAMED CAPABILITIES\Feature Staging Info	Enabled
NAMED CAPABILITIES\Graphics Capture	Enabled
NAMED CAPABILITIES\Hevc Playback	Enabled
NAMED CAPABILITIES\ID_CAP_CAMERA	Enabled
NAMED CAPABILITIES\ID_CAP_LOCATION	Enabled
NAMED CAPABILITIES\ID_CAP_MICROPHONE	Enabled
NAMED CAPABILITIES\Live Id Service	Enabled
NAMED CAPABILITIES\Location	Enabled
NAMED CAPABILITIES\Lpac App Experience	Enabled

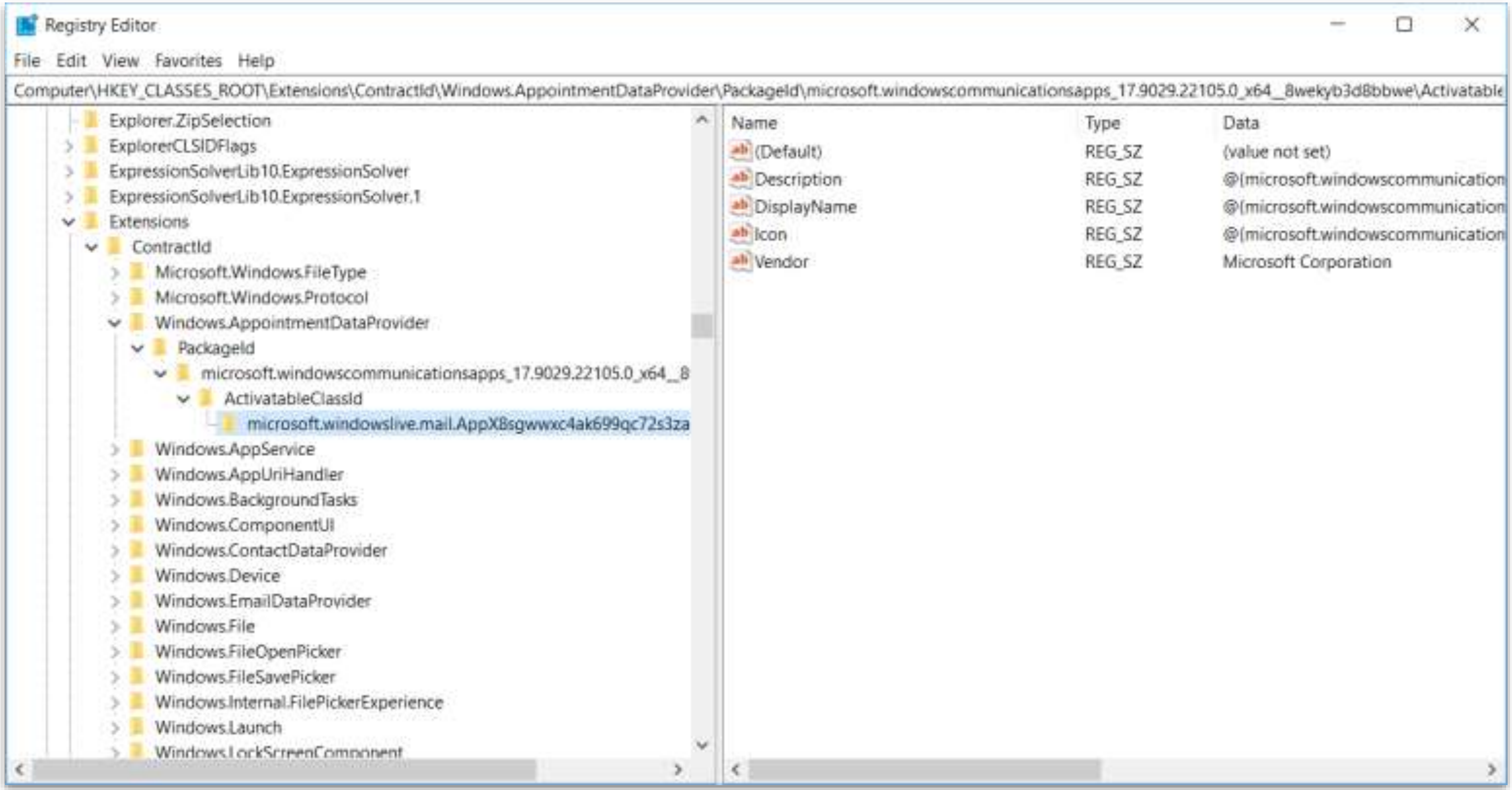
Package Family Name to Package SID


Microsoft.MicrosoftEdge_8wekyb3d8bbwe
microsoft.microsoftedge_8wekyb3d8bbwe
SHA256
S-1-15-2-%d-%d-%d-%d-%d-%d-%d

Windows Protocols

Contract ID	***Description***
Windows.Launch	Default Application Launch
Windows.Protocol	URI Protocol Handler
Windows.BackgroundTasks	Background Task
Windows.File	Launch and pass a file object
Windows.Search	Search request

Registered Extensions



Registry Editor

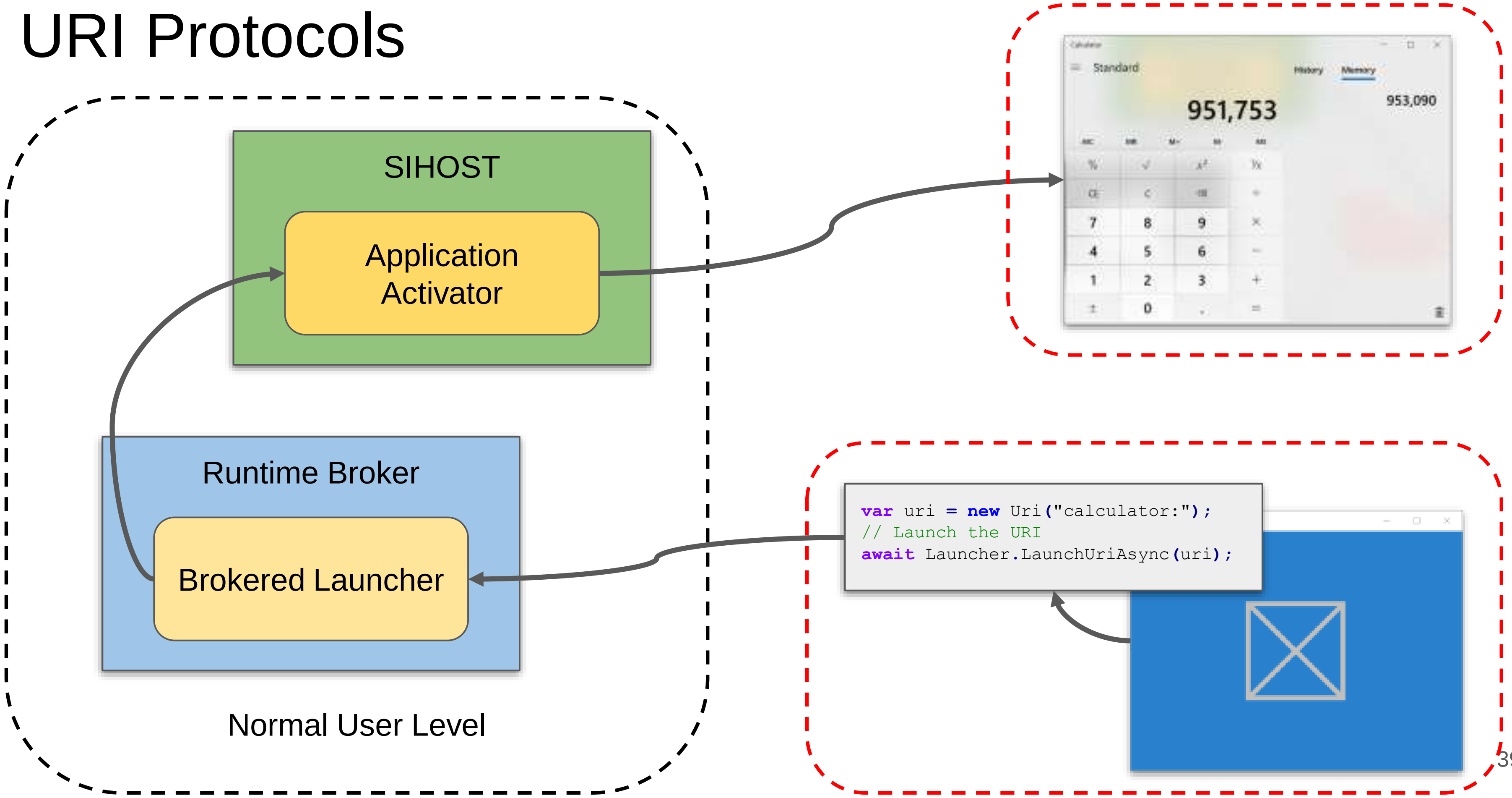
File Edit View Favorites Help

```
Computer\HKEY_CLASSES_ROOT\Extensions\ContractId\Windows.AppointmentDataProvider\PackageId\microsoft.windowscommunicationsapps_17.9029.22105.0_x64__8wekyb3d8bbwe\Activatable
```

```
  Explorer.ZipSelection
> ExplorerCLSIDFlags
> ExpressionSolverLib10.ExpressionSolver
> ExpressionSolverLib10.ExpressionSolver.1
v Extensions
  v ContractId
    > Microsoft.Windows.FileType
    > Microsoft.Windows.Protocol
    v Windows.AppointmentDataProvider
      v PackageId
        v microsoft.windowscommunicationsapps_17.9029.22105.0_x64__8
          v ActivatableClassId
              microsoft.windowslive.mail.AppX8sgwwxc4ak699qc72s3za
    > Windows.AppService
    > Windows.AppUriHandler
    > Windows.BackgroundTasks
    > Windows.ComponentUI
    > Windows.ContactDataProvider
    > Windows.Device
    > Windows.EmailDataProvider
    > Windows.File
    > Windows.FileOpenPicker
    > Windows.FileSavePicker
    > Windows.Internal.FilePickerExperience
    > Windows.Launch
    > Windows.LockScreenComponent
```

Name	Type	Data
(Default)	REG_SZ	(value not set)
Description	REG_SZ	@{microsoft.windowscommunication
DisplayName	REG_SZ	@{microsoft.windowscommunication
Icon	REG_SZ	@{microsoft.windowscommunication
Vendor	REG_SZ	Microsoft Corporation

URI Protocols

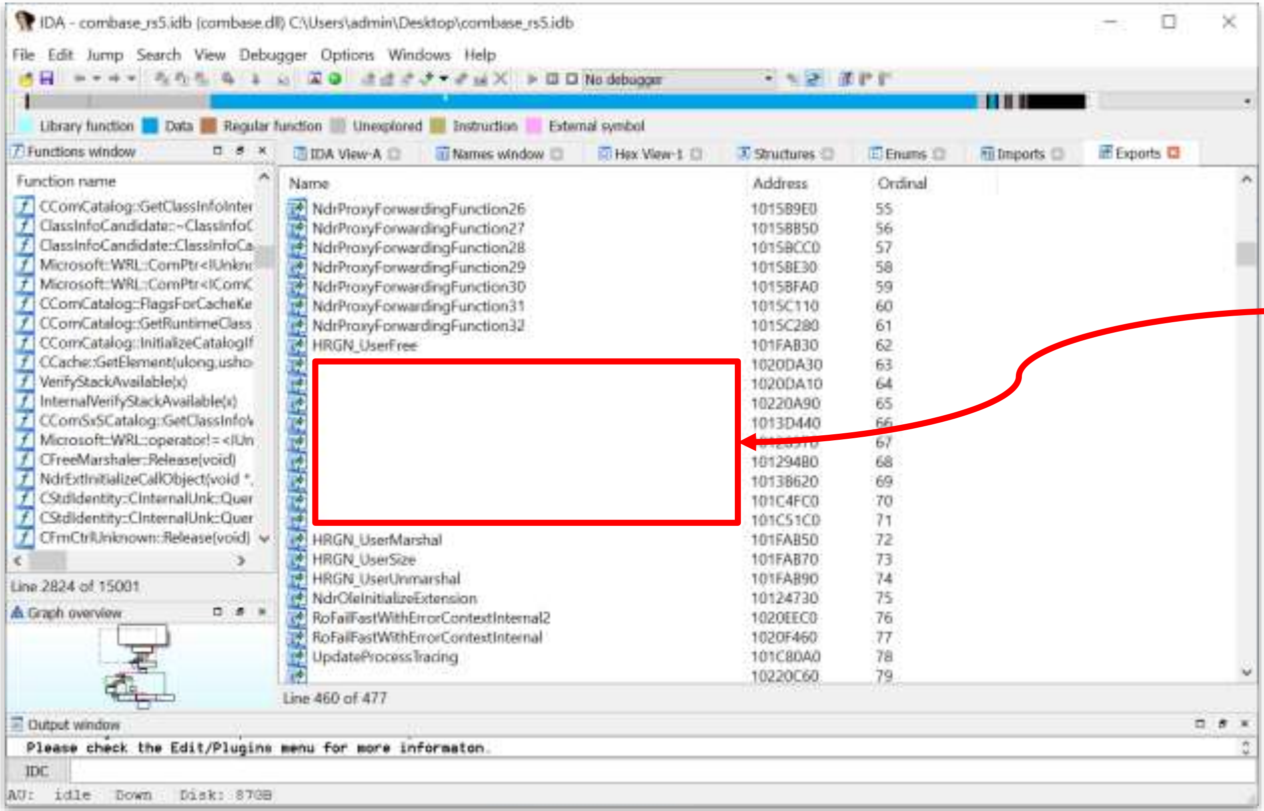


```
var uri = new Uri("calculator:");
// Launch the URI
await Launcher.LaunchUriAsync(uri);
```

Handling Activation

```
class App : Application {
  override void OnLaunched(LaunchActivatedEventArgs e) {
    // Handle Windows.Launch activation.
  }

  override void OnActivated(IActivatedEventArgs args) {
      if (args.Kind == ActivationKind.Protocol) {
         ProtocolActivatedEventArgs eventArgs =
            args as ProtocolActivatedEventArgs;
         // Handle Windows.Protocol activation
      }
   }
   // ...
}
```

The Undocumented Exports



Function name
CComCatalog::GetClassInfoInter
ClassInfoCandidate::~ClassInfoC
ClassInfoCandidate::ClassInfoCa
Microsoft::WRL::ComPtr<IUnkn
Microsoft::WRL::ComPtr<IComC
CComCatalog::FlagsForCacheKe
CComCatalog::GetRuntimeClass
CComCatalog::InitializeCatalogIf
CCache::GetElement(ulong,usho
VerifyStackAvailable(x)
InternalVerifyStackAvailable(x)
CComSxSCatalog::GetClassInfo\
Microsoft::WRL::operator!=<IUn
CFreeMarshaler::Release(void)
NdrExtInitializeCallObject(void *,
CStdIdentity::CInternalUnk::Quer
CStdIdentity::CInternalUnk::Quer
CFmCtrlUnknown::Release(void)

Line 2824 of 15001

Name	Address	Ordinal
NdrProxyForwardingFunction26	1015B9E0	55
NdrProxyForwardingFunction27	1015BB50	56
NdrProxyForwardingFunction28	1015BCC0	57
NdrProxyForwardingFunction29	1015BE30	58
NdrProxyForwardingFunction30	1015BFA0	59
NdrProxyForwardingFunction31	1015C110	60
NdrProxyForwardingFunction32	1015C280	61
HRGN_UserFree	101FAB30	62
	1020DA30	63
	1020DA10	64
	10220A90	65
	1013D440	66
	[illegible]	67
	101294B0	68
	1013B620	69
	101C4FC0	70
	101C51C0	71
HRGN_UserMarshal	101FAB50	72
HRGN_UserSize	101FAB70	73
HRGN_UserUnmarshal	101FAB90	74
NdrOleInitializeExtension	10124730	75
RoFailFastWithErrorContextInternal2	1020EEC0	76
RoFailFastWithErrorContextInternal	1020F460	77
UpdateProcessTracing	101C80A0	78
	10220C60	79

Line 460 of 477

```
Please check the Edit/Plugins menu for more informaton.
```

```
AU:  idle   Down   Disk: 870B
```

Where are the exported names?

Creating Instances in Packages

```
// Exported as Ordinal #135
HRESULT
RoActivateInstanceInPackage(HSTRING activatableClassId,
                            HSTRING packageMoniker,
                            IInspectable **instance) {

  return _RoActivateInstance(activatableClassId,
                             packageMoniker,
                             true,
                             0,
                             instance);
}
```

Doesn’t seem to be an equivalent for GetActivationFactory.

Extension Activation

```
// Exported as Ordinal #65
HRESULT RoGetExtensionRegistration(
  HSTRING contractId,
  HSTRING packageId,
  HSTRING activatableClassId,
  IExtensionRegistration **extensionRegistration);
```

```
IExtensionRegistration* reg =;
RoGetExtensionRegistration("Windows.Protocol",
        "Pkg_1.0.0.0_xxxxxxxxx", "Class", &reg);
reg->set_HostId(12345678);
IInspectable* obj;
reg->Activate(&obj);
```

Low Privilege/Restricted AC and Security Mitigations

```
[Guid("533148e2-ee0a-4b06-8500-7fda28f92ae2")]
interface IExtensionActivationContext {
    long HostId { get; set; }
    long UserContext { get; set; }
    long ComponentProcessId { get; set; }
    long RacActivationTokenId { get; set; }
    Blob LpacAttributes { get; set; }
    long ConsoleHandlesId { get; set; }
    int AAMActivationId {
}
```

```
HRESULT CoRegisterRacActivationToken(
        HANDLE     racActivationToken,
        PULONGLONG racActivationTokenId)
```

```
HRESULT GenerateProcThreadAttributeBlob(
        UINT entryCount, BLOB_ENTRY* blob,
        LPVOID *buffer, SIZE_T *bufferSize);
```

LpacAttributes Validation

```
HRESULT ValidateAttributeList(BLOB_ENTRY* blob,
                              UINT blobCount) {
  for(int i = 0; i < blobCount; ++i) {
    switch (blob[i].Attribute) {
      case ATTRIBUTE_MITIGATION_POLICY:
      case ATTRIBUTE_CHILD_PROCESS_POLICY:
      case ATTRIBUTE_ALL_APPLICATION_PACKAGES_POLICY:
      case ATTRIBUTE_WIN32K_FILTER:
      default:
        return E_NOTIMPL;
    }
    if (!ValidateAttribute(blob[i]))
      return E_INVALIDARG;
  }
  return S_OK;
}
```

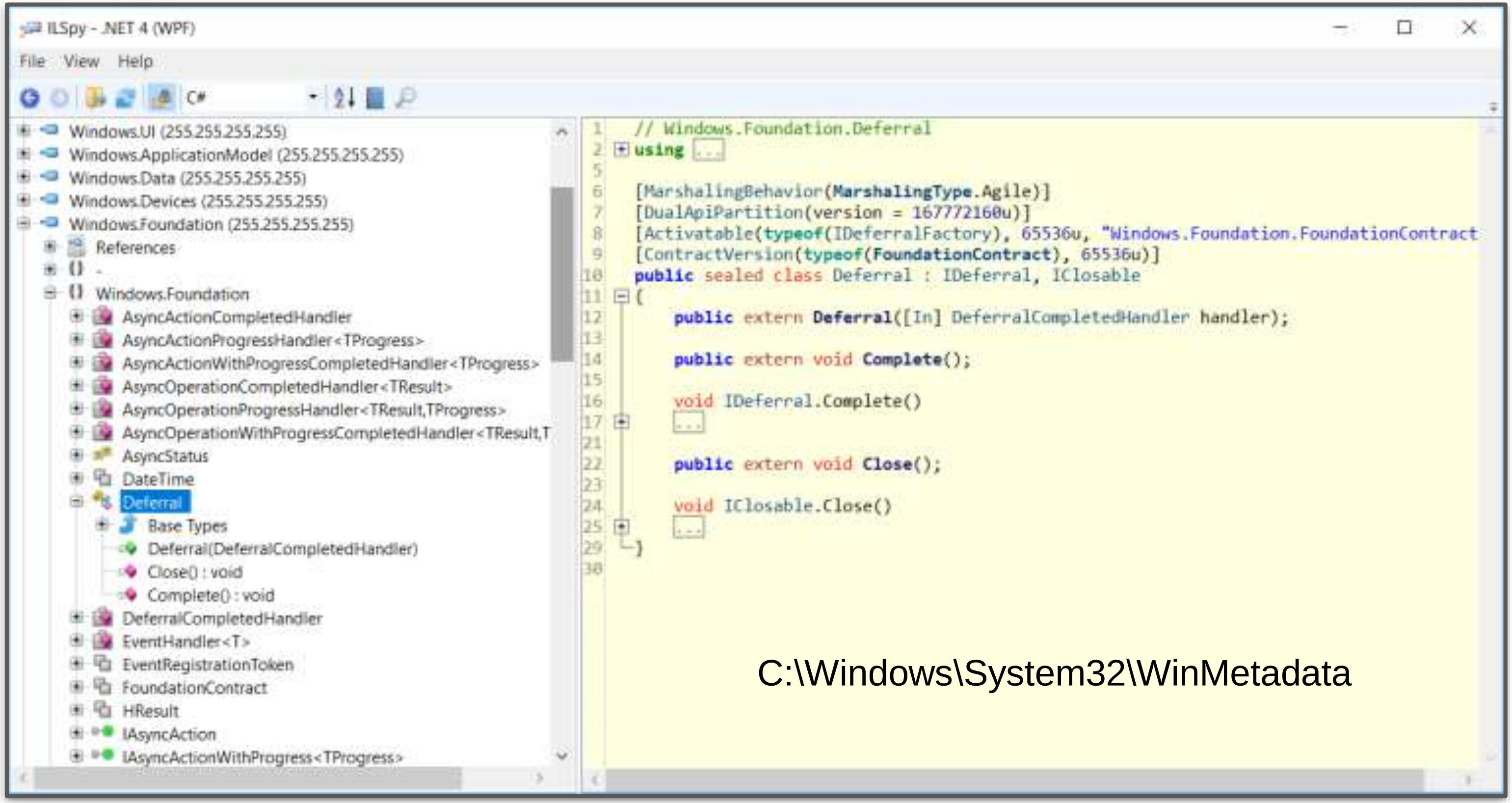
Limited set of attributes

Attribute data also validated

DEMO 2

Reverse Engineering Native Components

Windows Metadata



```
Windows.UI (255.255.255.255)
Windows.ApplicationModel (255.255.255.255)
Windows.Data (255.255.255.255)
Windows.Devices (255.255.255.255)
Windows.Foundation (255.255.255.255)
  References
  {} -
  {} Windows.Foundation
    AsyncActionCompletedHandler
    AsyncActionProgressHandler<TProgress>
    AsyncActionWithProgressCompletedHandler<TProgress>
    AsyncOperationCompletedHandler<TResult>
    AsyncOperationProgressHandler<TResult,TProgress>
    AsyncOperationWithProgressCompletedHandler<TResult,T
    AsyncStatus
    DateTime
    Deferral
      Base Types
      Deferral(DeferralCompletedHandler)
      Close() : void
      Complete() : void
    DeferralCompletedHandler
    EventHandler<T>
    EventRegistrationToken
    FoundationContract
    HResult
    IAsyncAction
    IAsyncActionWithProgress<TProgress>
```

```csharp
// Windows.Foundation.Deferral
using ...

[MarshalingBehavior(MarshalingType.Agile)]
[DualApiPartition(version = 167772160u)]
[Activatable(typeof(IDeferralFactory), 65536u, "Windows.Foundation.FoundationContract
[ContractVersion(typeof(FoundationContract), 65536u)]
public sealed class Deferral : IDeferral, IClosable
{
    public extern Deferral([In] DeferralCompletedHandler handler);

    public extern void Complete();

    void IDeferral.Complete()
    ...

    public extern void Close();

    void IClosable.Close()
    ...
}
```

C:\Windows\System32\WinMetadata

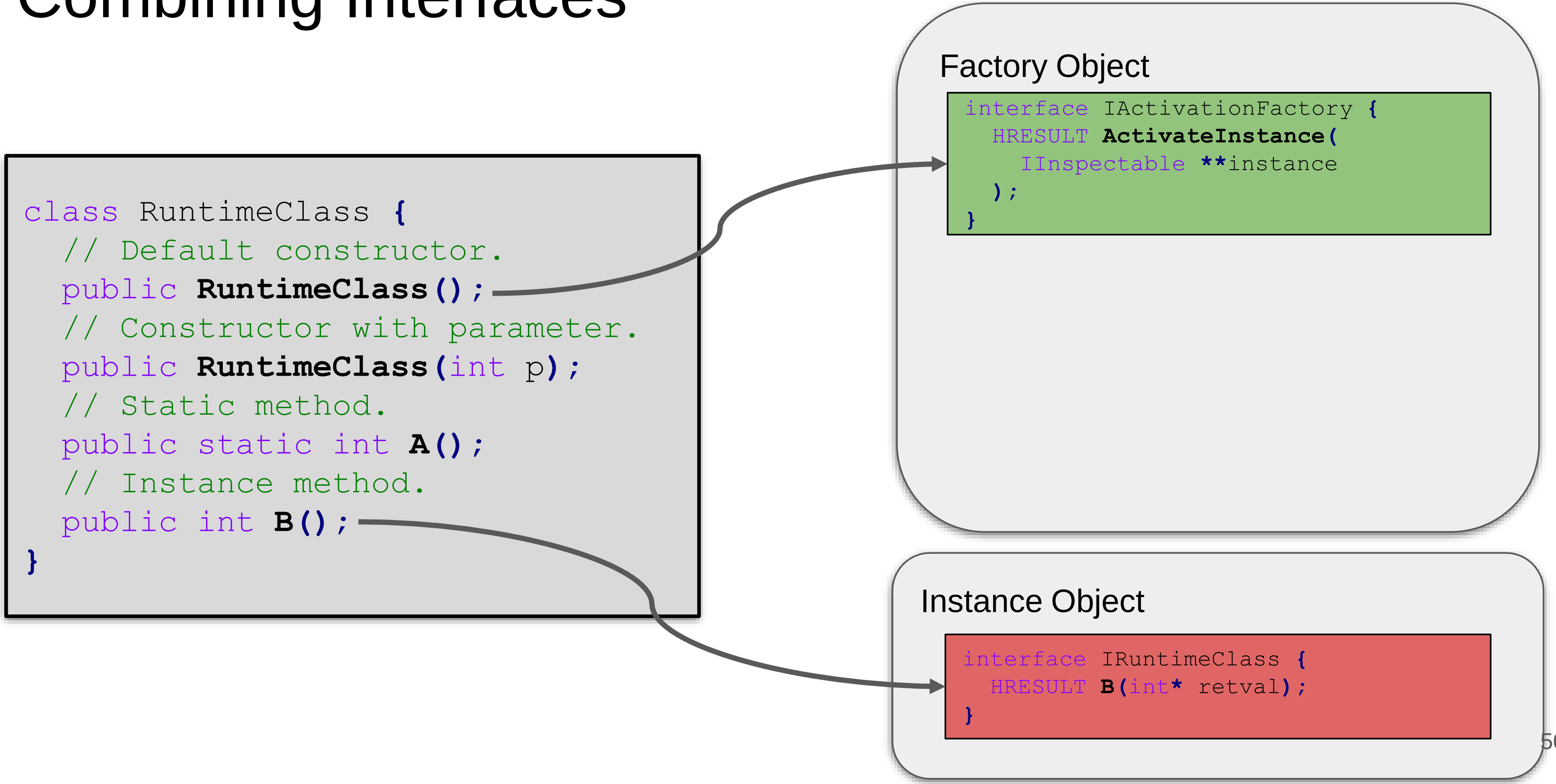
Combining Interfaces

```
class RuntimeClass {
  // Default constructor.
  public RuntimeClass();
  // Constructor with parameter.
  public RuntimeClass(int p);
  // Static method.
  public static int A();
  // Instance method.
  public int B();
}
```

Factory Object

Instance Object

Combining Interfaces



```
class RuntimeClass {
  // Default constructor.
  public RuntimeClass();
  // Constructor with parameter.
  public RuntimeClass(int p);
  // Static method.
  public static int A();
  // Instance method.
  public int B();
}
```

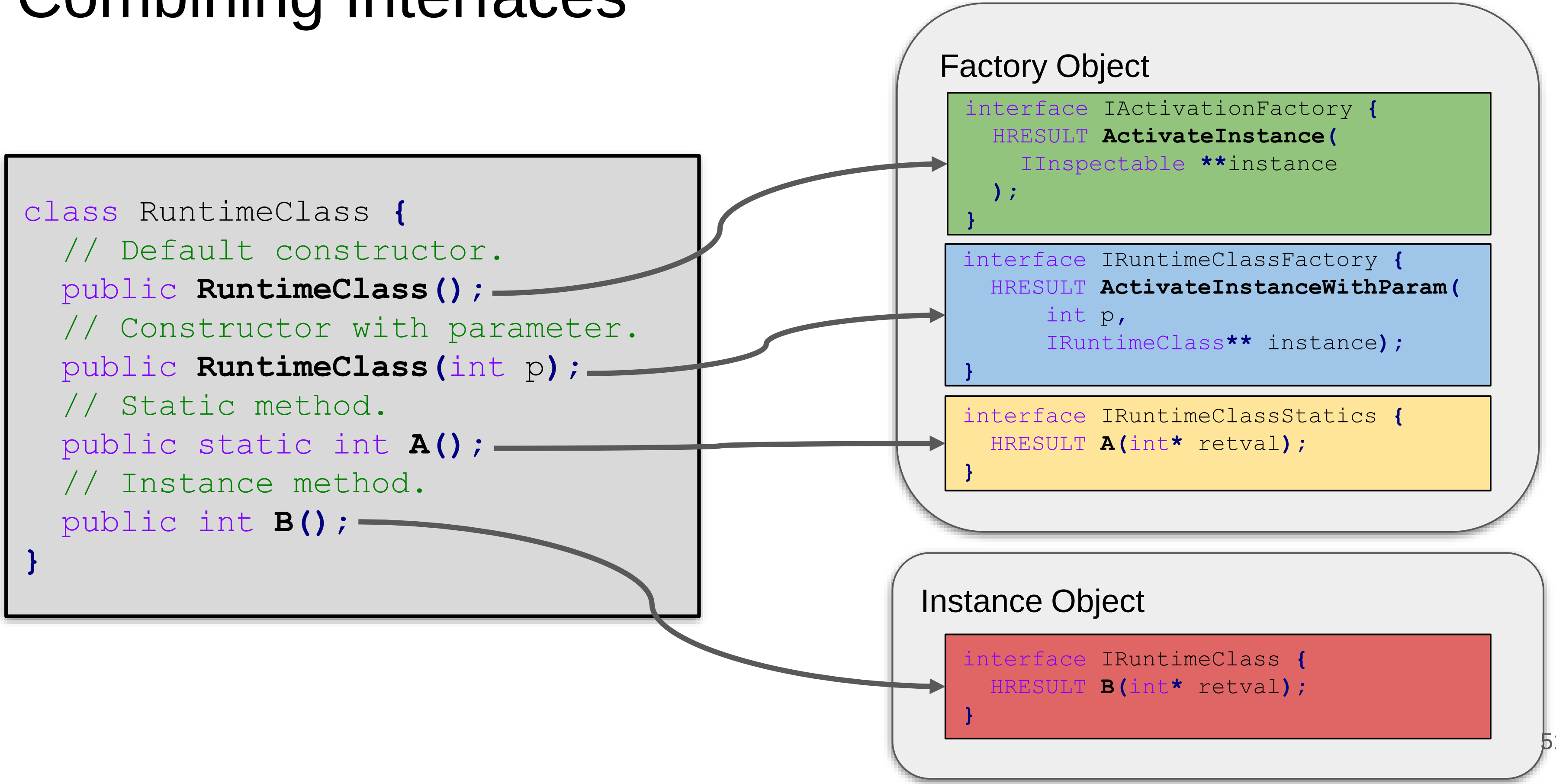
Factory Object

```
interface IActivationFactory {
  HRESULT ActivateInstance(
    IInspectable **instance
  );
}
```

Instance Object

```
interface IRuntimeClass {
  HRESULT B(int* retval);
}
```

Combining Interfaces



```
class RuntimeClass {
  // Default constructor.
  public RuntimeClass();
  // Constructor with parameter.
  public RuntimeClass(int p);
  // Static method.
  public static int A();
  // Instance method.
  public int B();
}
```

Factory Object

```
interface IActivationFactory {
  HRESULT ActivateInstance(
    IInspectable **instance
  );
}
```

```
interface IRuntimeClassFactory {
  HRESULT ActivateInstanceWithParam(
      int p,
      IRuntimeClass** instance);
}
```

```
interface IRuntimeClassStatics {
  HRESULT A(int* retval);
}
```

Instance Object

```
interface IRuntimeClass {
  HRESULT B(int* retval);
}
```

Finding the Implementation Binary

Get object for the class

```
PS> $cls = Get-ComRuntimeClass -Name "Class.Name"
```

If In-Process get DLL path

```
PS> $cls.DllPath
```

If OOP NormalExe get Server Exe Path

```
PS> $cls.ServerEntry.ExePath
```

If OOP service get Service name

```
PS> $cls.ServerEntry.ServiceName
```

Activation Entry Points

Exported from a DLL

```
HRESULT DllGetActivationFactory(
    HSTRING                 activatableClassId,
    IActivationFactory **factory
);
```

Called in an EXE

```
HRESULT RoRegisterActivationFactories(
  HSTRING                 *activatableClassIds,
  PFNGETACTIVATIONFACTORY *activationFactoryCallbacks,
  UINT32                  count,
  RO_REGISTRATION_COOKIE  *cookie
);
```

C++ Application Frameworks

C++/CX (Custom C++ dialect)

```
void App::OnLaunched(LaunchActivatedEventArgs^ e) {
    Handler^ handler = ref new Handler();
    handler->HandleLaunch("Launched");
}
```

C++/WRL (C++ 11)

```
HRESULT App::OnLauncher(ILaunchActivatedEventArgs* e) {
    ComPtr<IHandler> handler;
    HRESULT hr = Make<Handler>(&handler)
    if (FAILED(hr))
        return hr;
    HStringReference str(L"OnLaunched");
    return handler->HandleLaunch(str.Get());
}
```

C++/WINRT (C++ 17)

```
void App::OnLaunched(LaunchActivatedEventArgs const& e) {
    Handler handler = Handler();
    handler.HandleLaunch(hstring(L"Launched"));
}
```

IDL File

```
namespace WRLClass {
    [uuid(E74F1CF0-59C7-4CA6-BDE5-0F9DED9B4EF7),
        version(1.0), exclusiveto(WinRTClass)]
    interface IWinRTClass : IInspectable {
        HRESULT Add([in] int a, [in] int b,
                    [out, retval] int* value);
    }

    [version(1.0), activatable(1.0)]
    runtimeclass WinRTClass {
        [default] interface IWinRTClass;
    }
}
```

C++/WRL Implementation

```
class WinRTClass : public RuntimeClass<IWinRTClass> {
  InspectableClass(L"WRLClass.WinRTClass", BaseTrust)
public:
  HRESULT STDMETHODCALLTYPE Add(
         /* [in] */int a,
         /* [in] */int b,
         /* [retval, out] */int * value
  ) override {
    *value = a + b;
    return S_OK;
  }
};

ActivatableClass(WinRTClass);
```

Define base implementation of IInspectable

Interface Implementation

Define ActivationFactory

Finding Implemented Interfaces

```
HRESULT QueryInterface(REFIID riid, void** ppv) {
    bool handled = false;
    HRESULT hr = CustomQueryInterface(riid, ppv, &handled);
    if (FAILED(hr) || handled)
        return hr;

    return Super::AsIID(this, riid, ppv);
}
```

Overridable
Custom QI

Call AsIID
helper
method

AsIID Helper

Variadic Template

Handle Base Case

Specific CanCastTo

```
HRESULT AsIID(RuntimeClass<IT...>* implements,
              REFIID riid, void **ppv) {
    HRESULT hr = E_NOINTERFACE;
    if (riid == __uuidof(IUnknown)
     || riid == __uuidof(IInspectable)) {
        *ppv = implements->CastToUnknown();
        hr = S_OK;
    } else {
        hr = implements->CanCastTo(riid, ppv);
    }
    if (SUCCEEDED(hr))
        static_cast<IUnknown*>(*ppv)->AddRef();
    return hr;
}
```

CanCastTo Helper

Variadic Template Expanded

```
HRESULT RuntimeClass<I1, I2, I3> CanCastTo(REFIID riid,
                                          void* ppv) {
    if (riid == __uuidof(I1)) {
        ppv = static_cast<I1*>(this);
    } else if (riid == __uuidof(I2)) {
        ppv = static_cast<I2*>(this);
    } else if (riid == __uuidof(I3)) {
        ppv = static_cast<I2*>(this);
    } else {
        return E_NOINTERFACE;
    }
    return S_OK;
}
```

Test Each Interface

Sets the global symbol resolver to use WinDBG's DBGHELP

```
PS> Set-ComSymbolResolver "c:\windbg\dbghelp.dll"
```

Create a new instance of a COM object.

```
PS> $obj = New-ComObject -Class $cls
```

Get all known interfaces for class.

```
PS> $intf = Get-ComClassInterface $cls
```

Get all known IPIDs for an object and resolve method names (if symbols available).

```
PS> $ipids = Get-ComObjectIpid $obj `
                             -ResolveMethodNames
```

Format the COM IPIDs as text.

```
PS> $ipids | Format-ComProxy
```

Debugging Applications

Get all registered Windows.Launch Extensions

```
PS> Get-ComRuntimeExtension -Launch | `
                          Select PackageId, AppId
```

Start a package and debug it.

```
PS> windbg.exe -plmPackage PKGID -plmApp APPID
```

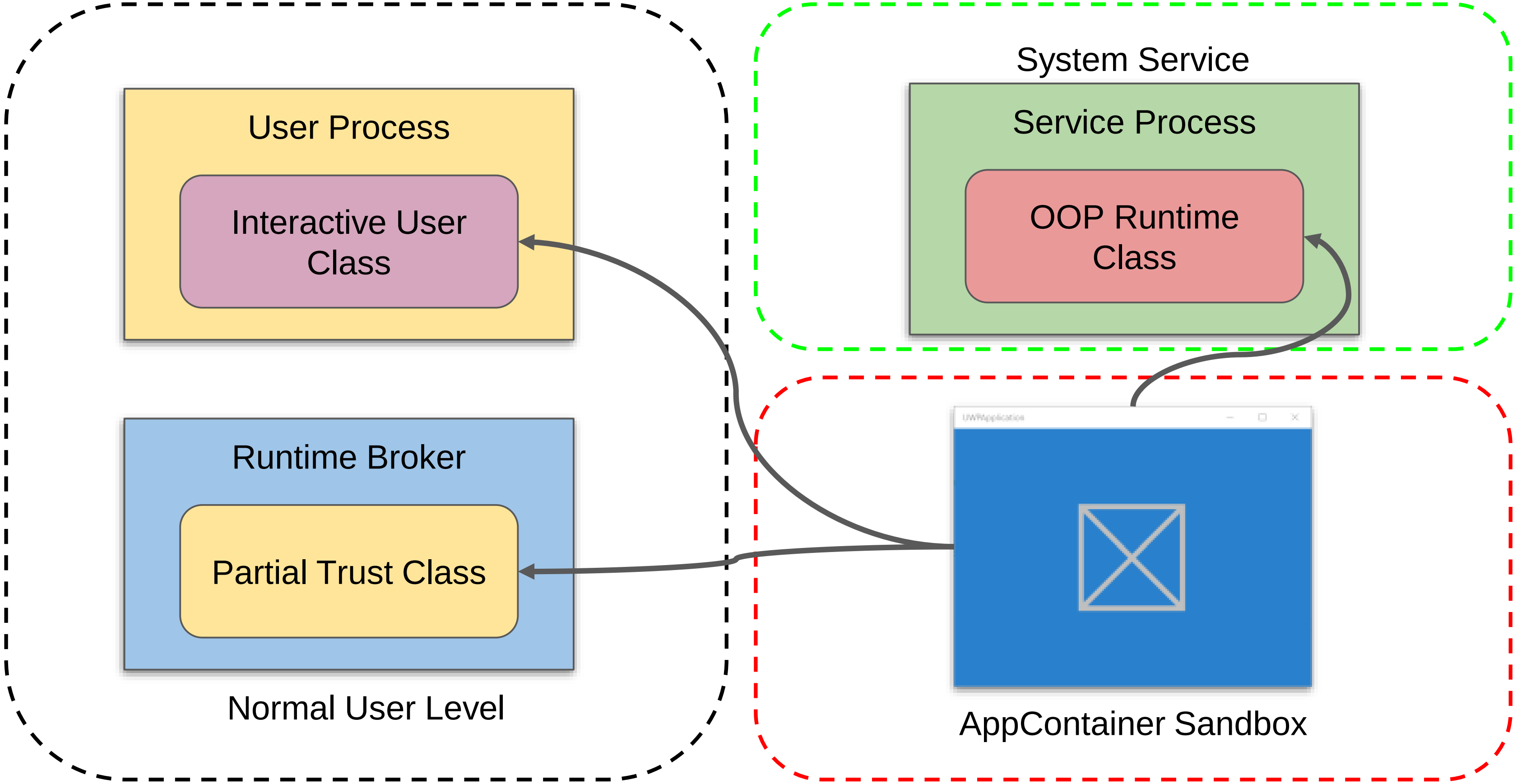
Enable debugging for a package

```
PS> plmdebug.exe /enableDebug PKGID DBGPATH.EXE
```

DEMO 3

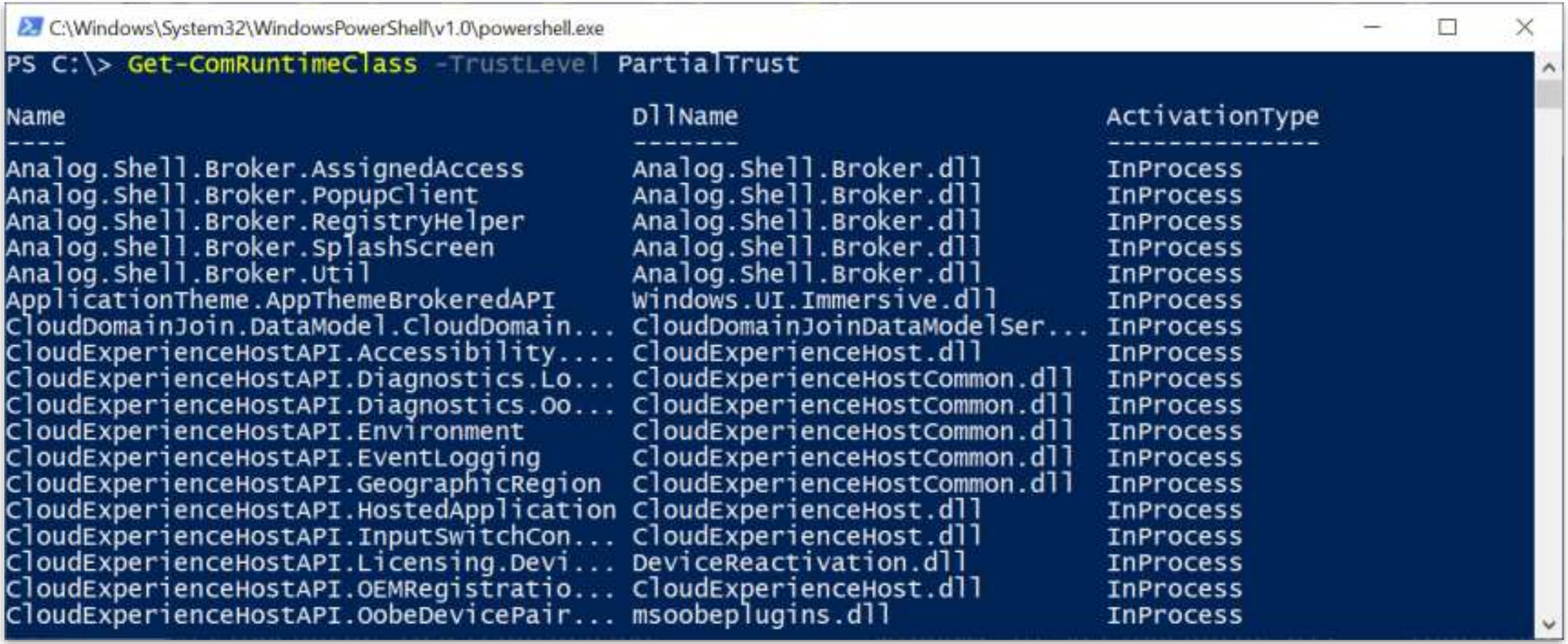
Windows Runtime Security

Sandbox Escape OOP Attack Surface



Partial Trust Brokered Classes

```
PS> Get-ComRuntimeClass -TrustLevel PartialTrust
```

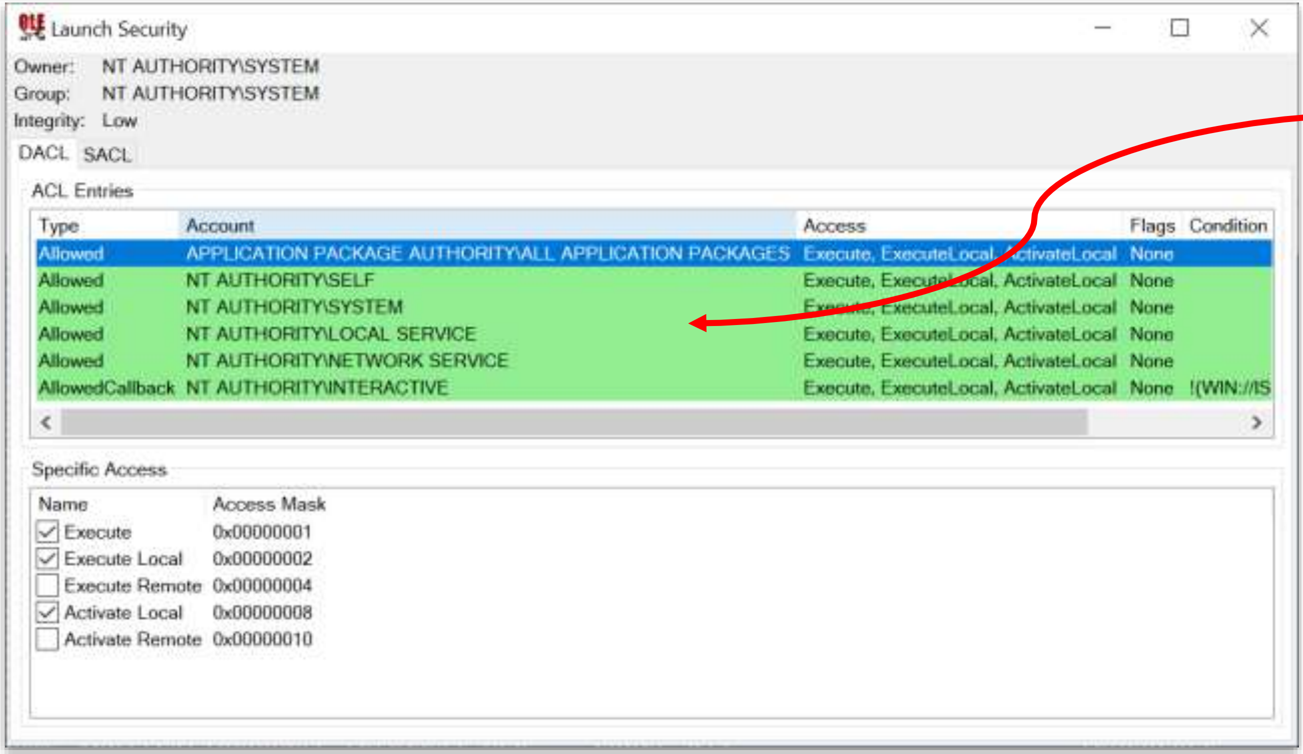


C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

```
PS C:\> Get-ComRuntimeClass -TrustLevel PartialTrust

Name                                     DllName                              ActivationType
----                                     -------                              --------------
Analog.Shell.Broker.AssignedAccess       Analog.Shell.Broker.dll              InProcess
Analog.Shell.Broker.PopupClient          Analog.Shell.Broker.dll              InProcess
Analog.Shell.Broker.RegistryHelper       Analog.Shell.Broker.dll              InProcess
Analog.Shell.Broker.SplashScreen         Analog.Shell.Broker.dll              InProcess
Analog.Shell.Broker.Util                 Analog.Shell.Broker.dll              InProcess
ApplicationTheme.AppThemeBrokeredAPI     Windows.UI.Immersive.dll             InProcess
CloudDomainJoin.DataModel.CloudDomain... CloudDomainJoinDataModelSer...       InProcess
CloudExperienceHostAPI.Accessibility.... CloudExperienceHost.dll              InProcess
CloudExperienceHostAPI.Diagnostics.Lo... CloudExperienceHostCommon.dll        InProcess
CloudExperienceHostAPI.Diagnostics.Oo... CloudExperienceHostCommon.dll        InProcess
CloudExperienceHostAPI.Environment       CloudExperienceHostCommon.dll        InProcess
CloudExperienceHostAPI.EventLogging      CloudExperienceHostCommon.dll        InProcess
CloudExperienceHostAPI.GeographicRegion  CloudExperienceHostCommon.dll        InProcess
CloudExperienceHostAPI.HostedApplication CloudExperienceHost.dll              InProcess
CloudExperienceHostAPI.InputSwitchCon... CloudExperienceHost.dll              InProcess
CloudExperienceHostAPI.Licensing.Devi... DeviceReactivation.dll               InProcess
CloudExperienceHostAPI.OEMRegistratio... CloudExperienceHost.dll              InProcess
CloudExperienceHostAPI.OobeDevicePair... msoobeplugins.dll                    InProcess
```

Partial Trust Class Default Permissions

```
PS> Show-ComSecurityDescriptor -RuntimeDefault
```



Launch Security

Owner: NT AUTHORITY\SYSTEM
Group: NT AUTHORITY\SYSTEM
Integrity: Low

DACL SACL

ACL Entries

Type	Account	Access	Flags	Condition
Allowed	APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	NT AUTHORITY\SELF	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	NT AUTHORITY\SYSTEM	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	NT AUTHORITY\LOCAL SERVICE	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	NT AUTHORITY\NETWORK SERVICE	Execute, ExecuteLocal, ActivateLocal	None	
AllowedCallback	NT AUTHORITY\INTERACTIVE	Execute, ExecuteLocal, ActivateLocal	None	!(WIN://IS

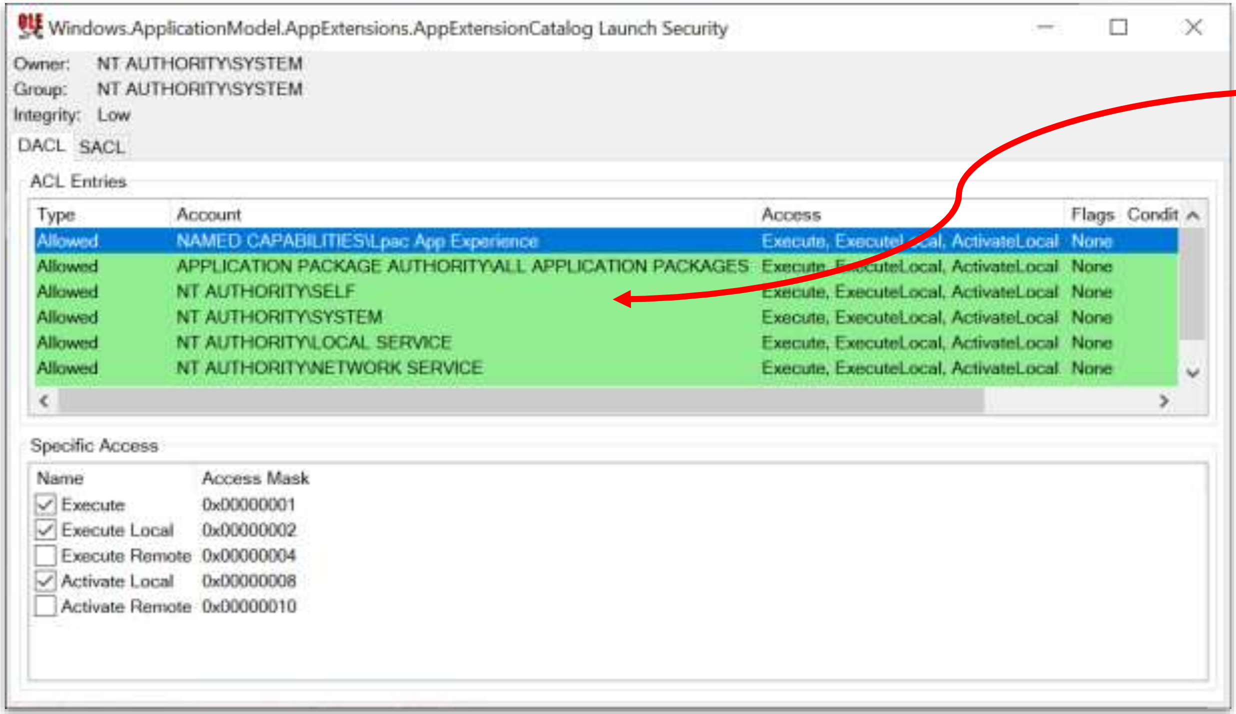
Specific Access

Name	Access Mask
✓ Execute	0x00000001
✓ Execute Local	0x00000002
☐ Execute Remote	0x00000004
✓ Activate Local	0x00000008
☐ Activate Remote	0x00000010

Allows all AC at the same user to access the class.

Class Specific Permissions

```
PS> Show-ComSecurityDescriptor $cls
```



Windows.ApplicationModel.AppExtensions.AppExtensionCatalog Launch Security

Owner: NT AUTHORITY\SYSTEM
Group: NT AUTHORITY\SYSTEM
Integrity: Low

DACL SACL

ACL Entries

Type	Account	Access	Flags	Condit
Allowed	NAMED CAPABILITIES\Lpac App Experience	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	NT AUTHORITY\SELF	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	NT AUTHORITY\SYSTEM	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	NT AUTHORITY\LOCAL SERVICE	Execute, ExecuteLocal, ActivateLocal	None	
Allowed	NT AUTHORITY\NETWORK SERVICE	Execute, ExecuteLocal, ActivateLocal	None	

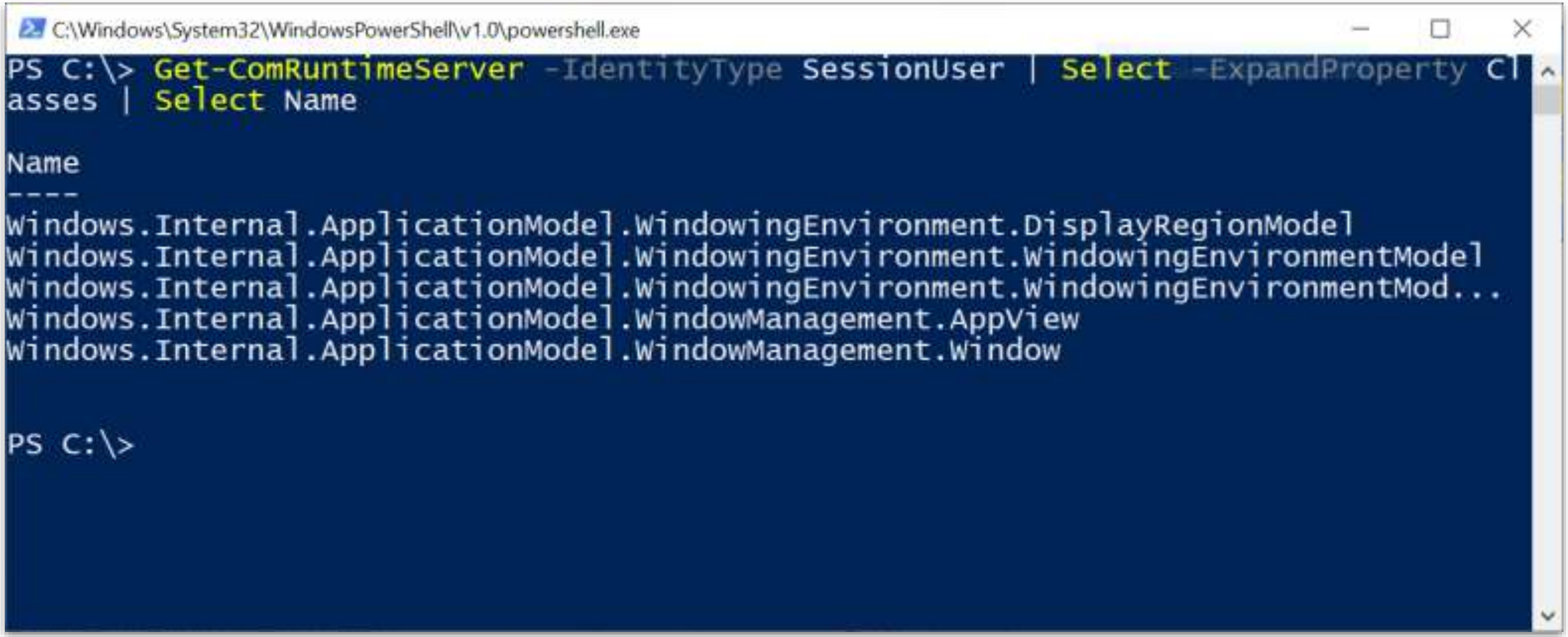
Specific Access

Name	Access Mask
✓ Execute	0x00000001
✓ Execute Local	0x00000002
☐ Execute Remote	0x00000004
✓ Activate Local	0x00000008
☐ Activate Remote	0x00000010

Adds the ***lpacAppExperience*** capability

Interactive User Classes

```
PS> Get-ComRuntimeServer -IdentityType SessionUser `
    | Select -ExpandProperty Classes
```

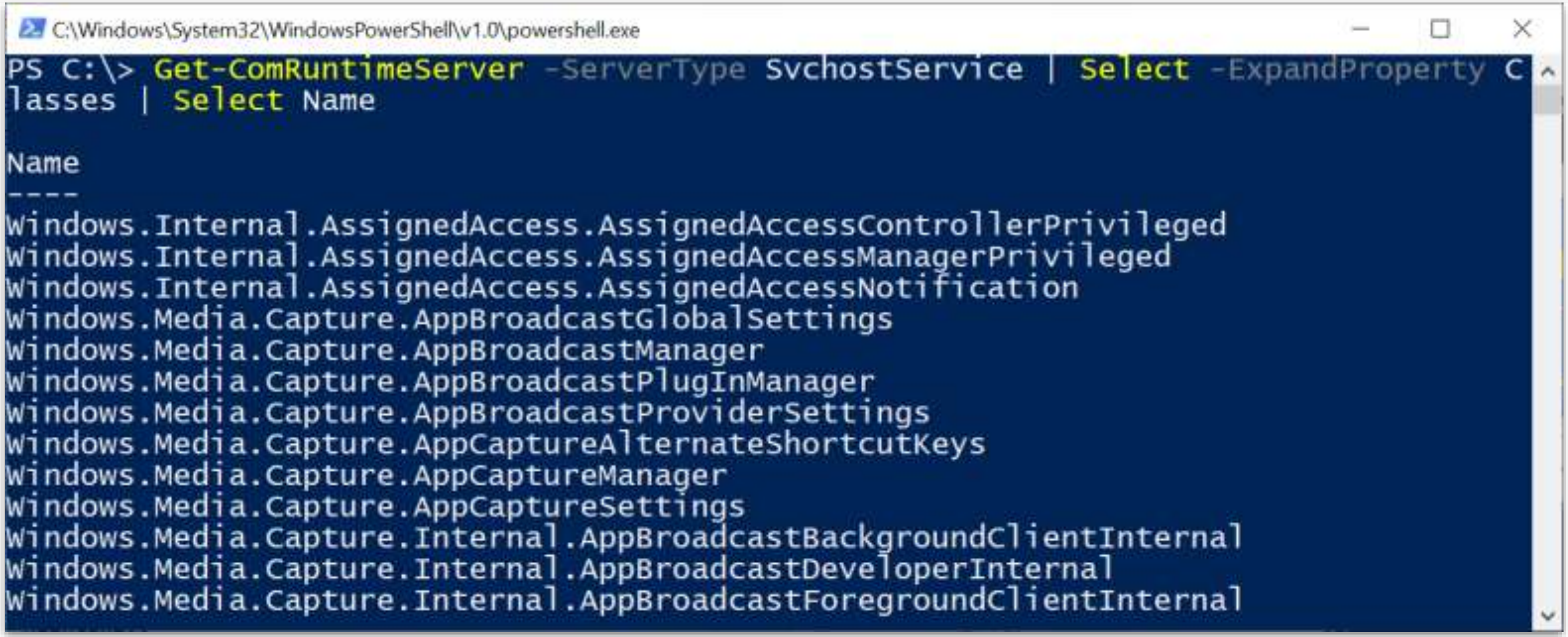


C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

```
PS C:\> Get-ComRuntimeServer -IdentityType SessionUser | Select -ExpandProperty Cl
asses | Select Name

Name
----
Windows.Internal.ApplicationModel.WindowingEnvironment.DisplayRegionModel
Windows.Internal.ApplicationModel.WindowingEnvironment.WindowingEnvironmentModel
Windows.Internal.ApplicationModel.WindowingEnvironment.WindowingEnvironmentMod...
Windows.Internal.ApplicationModel.WindowManagement.AppView
Windows.Internal.ApplicationModel.WindowManagement.Window


PS C:\>
```

System Service Classes

```
PS> Get-ComRuntimeServer -ServerType SvchostService `
      | Select -ExpandProperty Classes
```

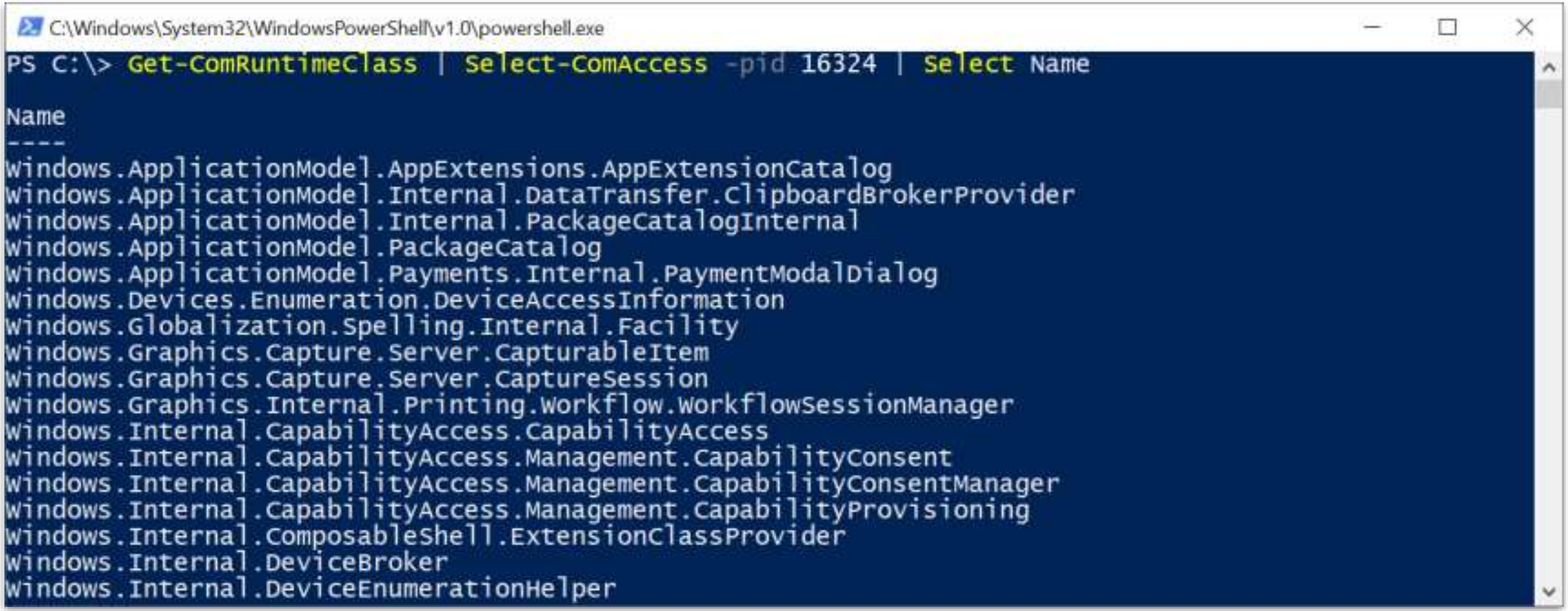


C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

```
PS C:\> Get-ComRuntimeServer -ServerType SvchostService | Select -ExpandProperty C
lasses | Select Name

Name
----
Windows.Internal.AssignedAccess.AssignedAccessControllerPrivileged
Windows.Internal.AssignedAccess.AssignedAccessManagerPrivileged
Windows.Internal.AssignedAccess.AssignedAccessNotification
Windows.Media.Capture.AppBroadcastGlobalSettings
Windows.Media.Capture.AppBroadcastManager
Windows.Media.Capture.AppBroadcastPlugInManager
Windows.Media.Capture.AppBroadcastProviderSettings
Windows.Media.Capture.AppCaptureAlternateShortcutKeys
Windows.Media.Capture.AppCaptureManager
Windows.Media.Capture.AppCaptureSettings
Windows.Media.Capture.Internal.AppBroadcastBackgroundClientInternal
Windows.Media.Capture.Internal.AppBroadcastDeveloperInternal
Windows.Media.Capture.Internal.AppBroadcastForegroundClientInternal
```

Finding Accessible Classes

```
PS> Get-ComRuntimeClass | Select-ComAcccess -pid X
```



C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

```
PS C:\> Get-ComRuntimeClass | Select-ComAccess -pid 16324 | Select Name

Name
----
Windows.ApplicationModel.AppExtensions.AppExtensionCatalog
Windows.ApplicationModel.Internal.DataTransfer.ClipboardBrokerProvider
Windows.ApplicationModel.Internal.PackageCatalogInternal
Windows.ApplicationModel.PackageCatalog
Windows.ApplicationModel.Payments.Internal.PaymentModalDialog
Windows.Devices.Enumeration.DeviceAccessInformation
Windows.Globalization.Spelling.Internal.Facility
Windows.Graphics.Capture.Server.CapturableItem
Windows.Graphics.Capture.Server.CaptureSession
Windows.Graphics.Internal.Printing.Workflow.WorkflowSessionManager
Windows.Internal.CapabilityAccess.CapabilityAccess
Windows.Internal.CapabilityAccess.Management.CapabilityConsent
Windows.Internal.CapabilityAccess.Management.CapabilityConsentManager
Windows.Internal.CapabilityAccess.Management.CapabilityProvisioning
Windows.Internal.ComposableShell.ExtensionClassProvider
Windows.Internal.DeviceBroker
Windows.Internal.DeviceEnumerationHelper
```

Package Name Checks

```
BOOL BrokerAuthenticateCOMCaller() {
  HANDLE token;
  CoImpersonateClient();
  OpenThreadToken(GetCurrentThread(), TOKEN_QUERY, &token);
  WCHAR family_name[255];
  ULONG family_name_length = 255;
  NTSTATUS status = RtlQueryPackageClaims(token,
                              family_name, &family_name_length);
  if (NT_SUCCESS(status))
    return wcsicmp(package_name, L"MicrosoftEdge") == 0;
  return FALSE;
}
```

Reads from WIN://SYSAPPID

HSTRING is a Counted String

```
UINT32 length;
PCWSTR str = WindowsGetStringRawBuffer(hString,
                                       &length);
// Might not be equal.
assert(wcslen(str) == length);
```

```
HRESULT WindowsStringHasEmbeddedNull(
  HSTRING string,
  BOOL    *hasEmbedNull
);
```

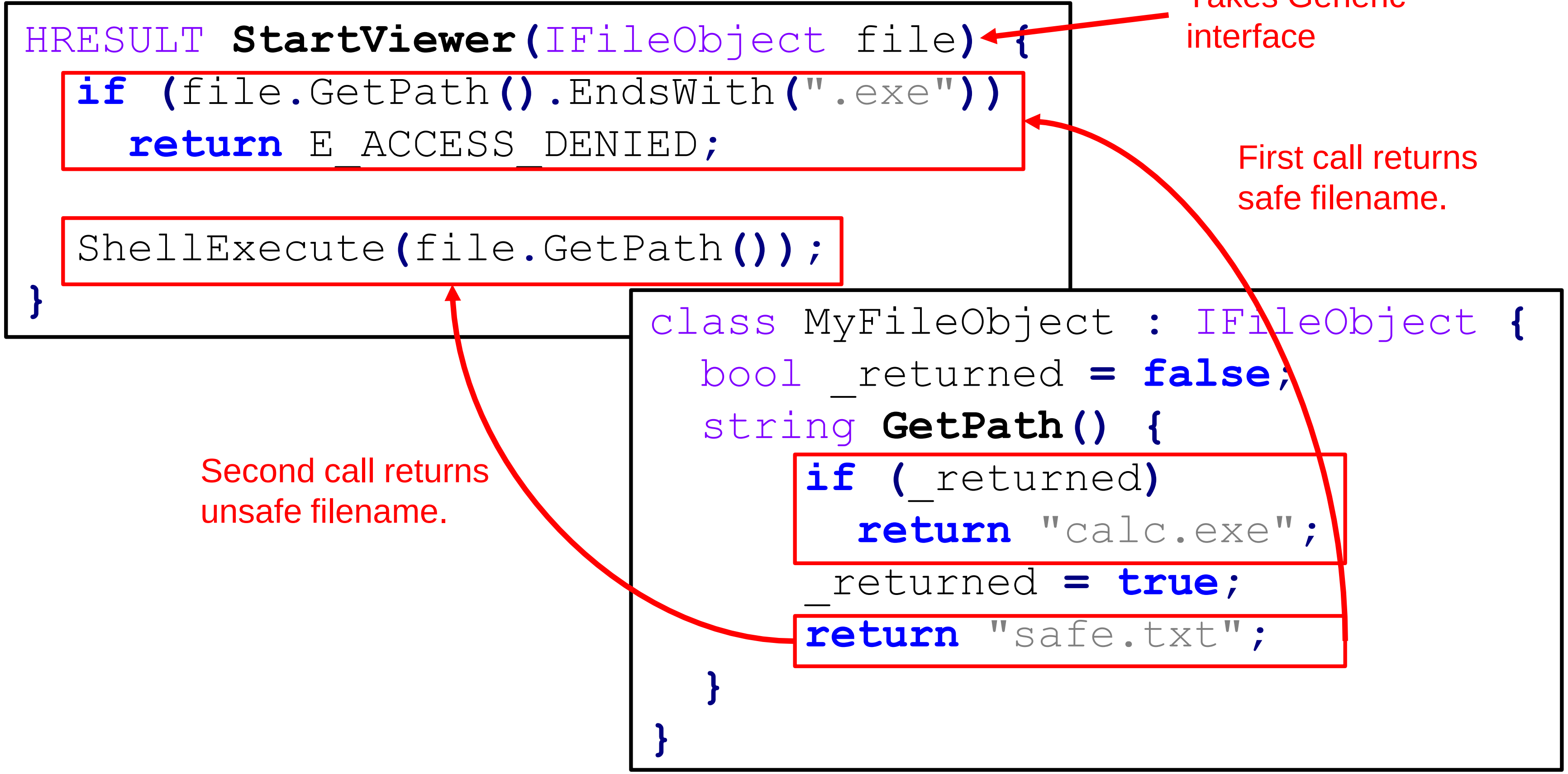
Incorrect Capability or Missing Security Checks

```
HANDLE CheckedCreateFile(string path) {
  // Get client token.
  HANDLE token;
  CoImpersonateClient();
  OpenThreadToken(GetCurrentThread(), &token);

  HANDLE ret = INVALID_HANDLE_VALUE;
  if (CapabilityCheck(token, L"internetClient")) {
    ret = CreateFile(path, ...);
  }

  return ret;
}
```

Checking for internetClient capability

But opening a file.

TOCTOU in Marshaled Interfaces

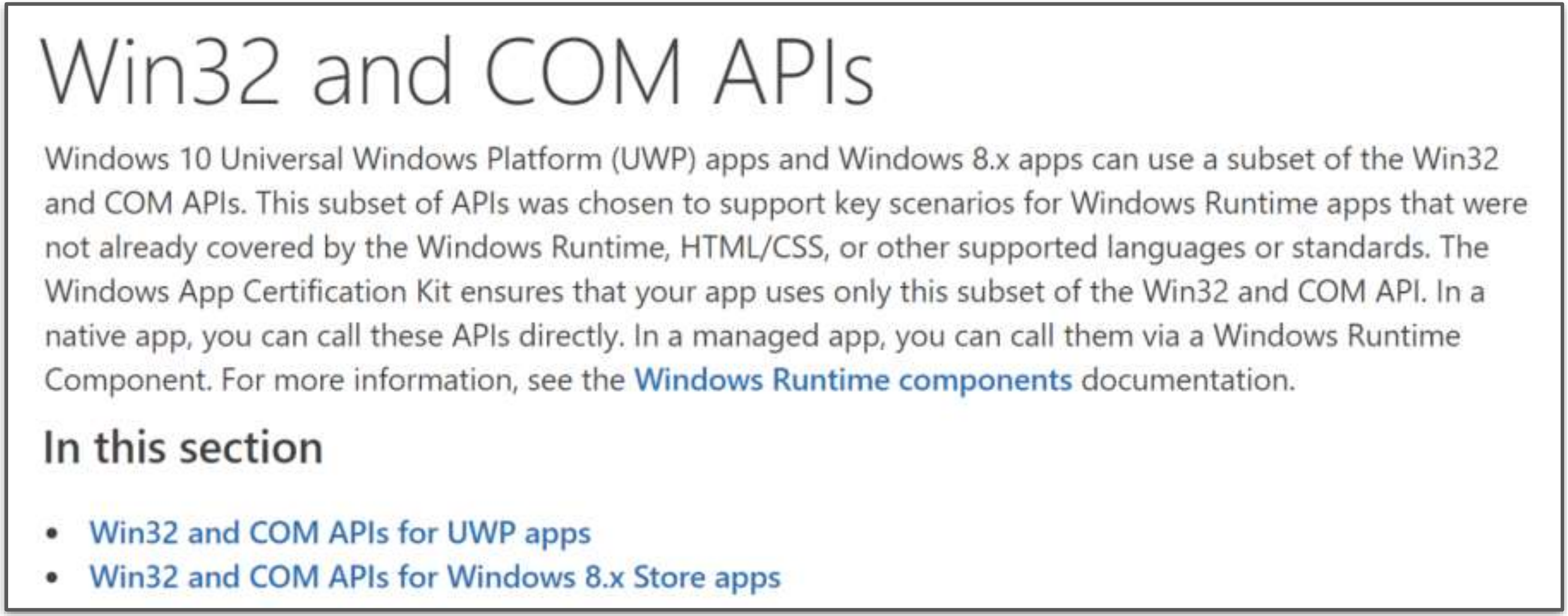


```
HRESULT StartViewer(IFileObject file) {
  if (file.GetPath().EndsWith(".exe"))
    return E_ACCESS_DENIED;

  ShellExecute(file.GetPath());
}
```

```
class MyFileObject : IFileObject {
  bool _returned = false;
  string GetPath() {
      if (_returned)
        return "calc.exe";
      _returned = true;
      return "safe.txt";
  }
}
```

Takes Generic interface
First call returns safe filename.
Second call returns unsafe filename.

The Challenges of Writing a Proof of Concept



Win32 and COM APIs

Windows 10 Universal Windows Platform (UWP) apps and Windows 8.x apps can use a subset of the Win32 and COM APIs. This subset of APIs was chosen to support key scenarios for Windows Runtime apps that were not already covered by the Windows Runtime, HTML/CSS, or other supported languages or standards. The Windows App Certification Kit ensures that your app uses only this subset of the Win32 and COM API. In a native app, you can call these APIs directly. In a managed app, you can call them via a Windows Runtime Component. For more information, see the Windows Runtime components documentation.

In this section

- Win32 and COM APIs for UWP apps
- Win32 and COM APIs for Windows 8.x Store apps

https://msdn.microsoft.com/en-us/library/windows/apps/br205757.aspx

CoCreateInstanceFromApp

```
#if !(WINAPI_PARTITION_DESKTOP | WINAPI_PARTITION_SYSTEM)
HRESULT CoCreateInstanceEx(
    REFCLSID      Clsid,
    ...,
    MULTI_QI*     pResults) {
    return CoCreateInstanceFromApp(Clsid, punkOuter,
           dwClsCtx, pServerInfo, dwCount, pResults);
}
```

If not a desktop application call FromApp version

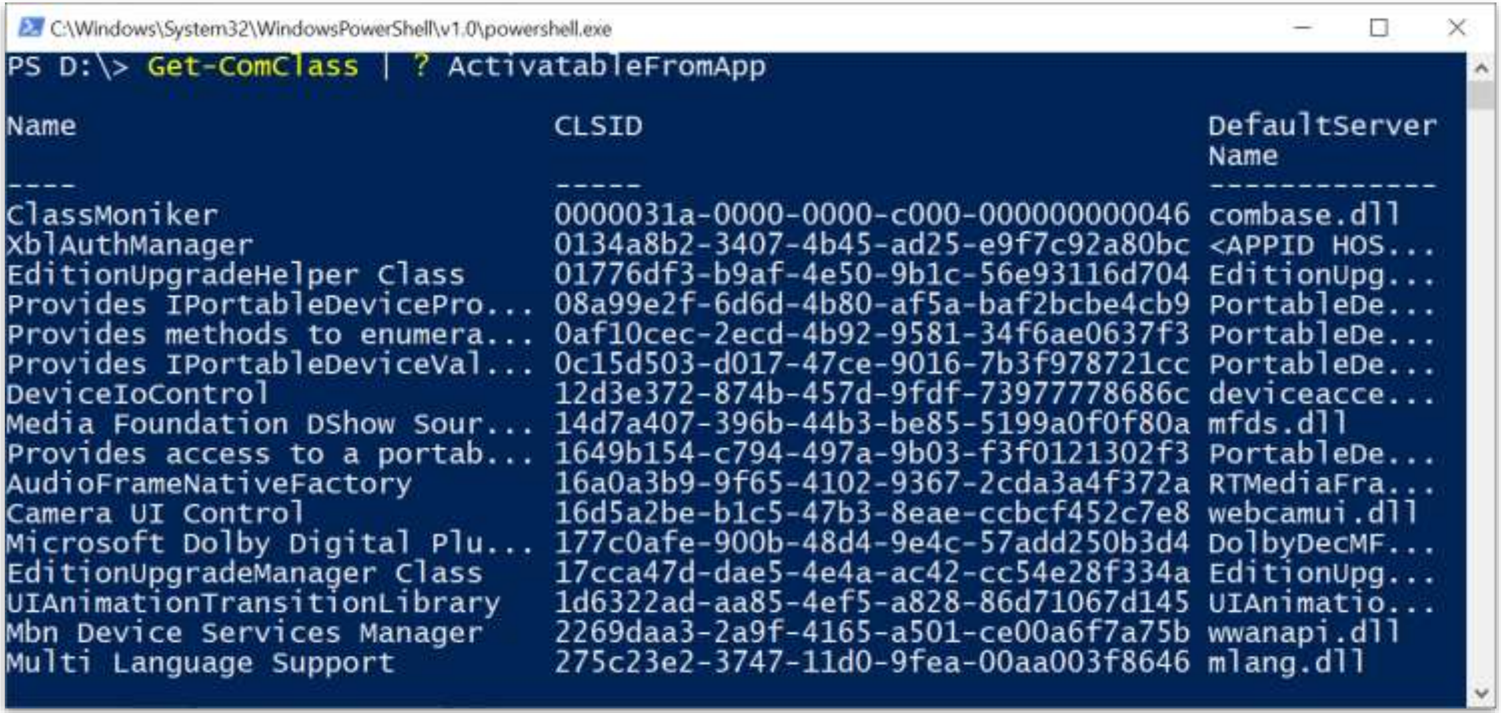
```
#else
```

Normal import.

```
HRESULT STDAPI CoCreateInstanceEx(REFCLSID Clsid, ...);

#endif // !(WINAPI_PARTITION_DESKTOP | WINAPI_PARTITION_SYSTEM)
```

COM Class Restrictions

```
PS> Get-ComClass | ? ActivatableFromApp
```


C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

```
PS D:\> Get-ComClass | ? ActivatableFromApp

Name                           CLSID                                 DefaultServer
                                                                     Name
----                           -----                                 -------------
ClassMoniker                   0000031a-0000-0000-c000-000000000046 combase.dll
XblAuthManager                 0134a8b2-3407-4b45-ad25-e9f7c92a80bc <APPID HOS...
EditionUpgradeHelper Class     01776df3-b9af-4e50-9b1c-56e93116d704 EditionUpg...
Provides IPortableDevicePro... 08a99e2f-6d6d-4b80-af5a-baf2bcbe4cb9 PortableDe...
Provides methods to enumera... 0af10cec-2ecd-4b92-9581-34f6ae0637f3 PortableDe...
Provides IPortableDeviceVal... 0c15d503-d017-47ce-9016-7b3f978721cc PortableDe...
DeviceIoControl                12d3e372-874b-457d-9fdf-73977778686c deviceacce...
Media Foundation DShow Sour... 14d7a407-396b-44b3-be85-5199a0f0f80a mfds.dll
Provides access to a portab... 1649b154-c794-497a-9b03-f3f0121302f3 PortableDe...
AudioFrameNativeFactory        16a0a3b9-9f65-4102-9367-2cda3a4f372a RTMediaFra...
Camera UI Control              16d5a2be-b1c5-47b3-8eae-ccbcf452c7e8 webcamui.dll
Microsoft Dolby Digital Plu... 177c0afe-900b-48d4-9e4c-57add250b3d4 DolbyDecMF...
EditionUpgradeManager Class    17cca47d-dae5-4e4a-ac42-cc54e28f334a EditionUpg...
UIAnimationTransitionLibrary   1d6322ad-aa85-4ef5-a828-86d71067d145 UIAnimatio...
Mbn Device Services Manager    2269daa3-2a9f-4165-a501-ce00a6f7a75b wwanapi.dll
Multi Language Support         275c23e2-3747-11d0-9fea-00aa003f8646 mlang.dll
```

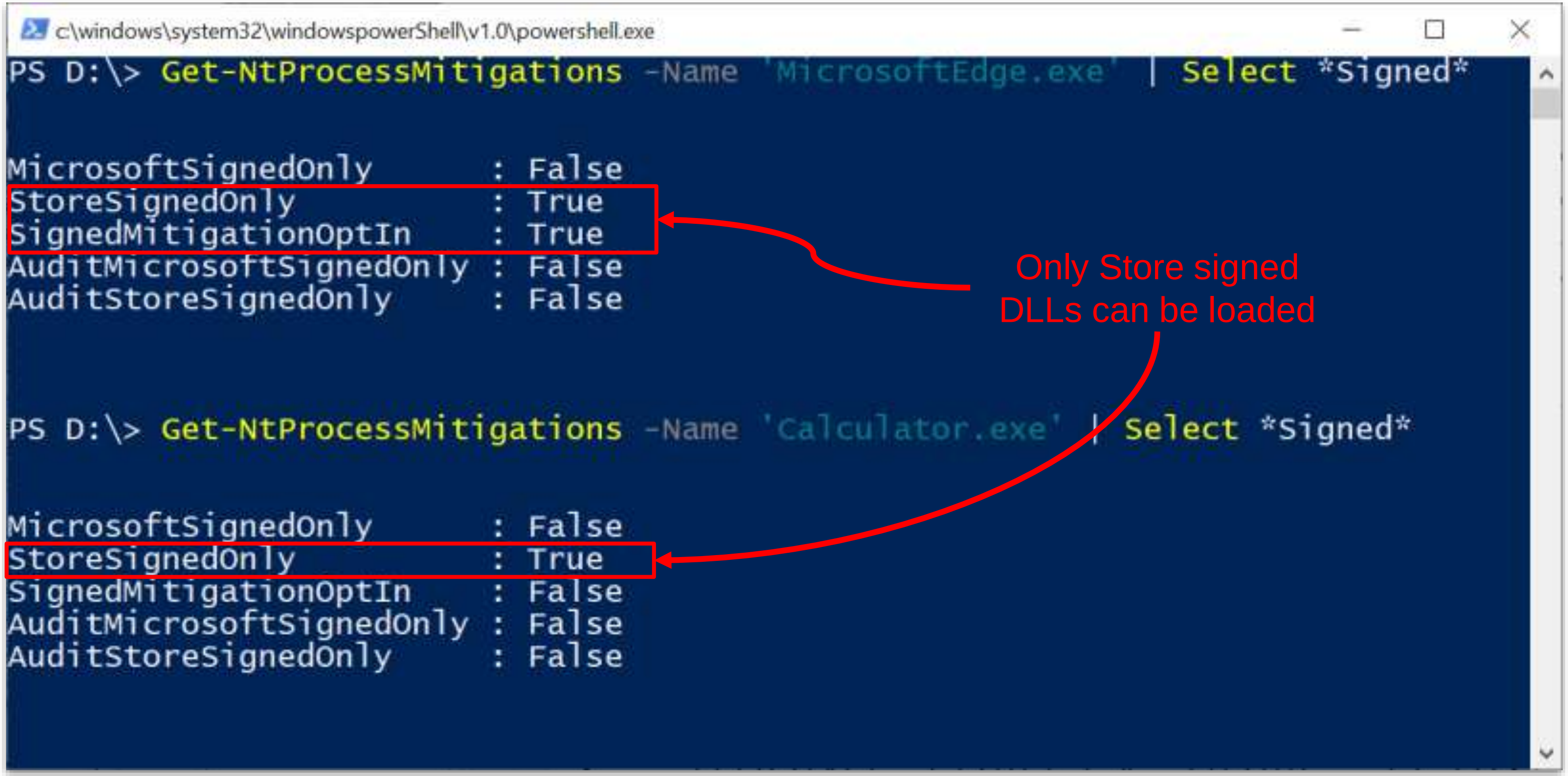
Load at Runtime

```
IUnknown* CreateObject(REFCLSID clsid) {
  HMODULE mod = GetModuleHandle(L"combase");
  fCCI pfCCI = (fCCI)GetProcAddress(mod, "CoCreateInstance");

  IUnknown* unk;
  pfCCI(clsid, nullptr, CLSCTX_SERVER, IID_PPV_ARGS(&unk));
  return unk;
}
```

Late bound call.

Probably wouldn't get through Store review process.

Inject a DLL Into Running Process



```
c:\windows\system32\windowspowerShell\v1.0\powershell.exe

PS D:\> Get-NtProcessMitigations -Name 'MicrosoftEdge.exe' | Select *Signed*


MicrosoftSignedOnly        : False
StoreSignedOnly            : True
SignedMitigationOptIn      : True
AuditMicrosoftSignedOnly   : False
AuditStoreSignedOnly       : False



PS D:\> Get-NtProcessMitigations -Name 'Calculator.exe' | Select *Signed*


MicrosoftSignedOnly        : False
StoreSignedOnly            : True
SignedMitigationOptIn      : False
AuditMicrosoftSignedOnly   : False
AuditStoreSignedOnly       : False
```

DEMO 4

Conclusions

- All based on familiar COM programming paradigms
- The Windows Runtime has many interesting attack surfaces
  - Attack surface which might be accessible remotely
  - Plenty of Sandbox to User and User to System privilege escalation routes
- Tooling is not quite there, making an effort with OleViewDotNet

谢谢
Any Questions?