Windows 8 WinRT Interop

Windows Runtime (WinRT) is the new platform designed for Windows 8 Metro-style applications. WinRT is implemented in native code (i.e. .NET Framework is not used by WinRT), but you can target WinRT from C++/CX, .NET languages or JavaScript. WinRT replaces a large portion of Win32 and .NET BCL, which become inaccessible. WinRT places an emphasis on asynchrony, making it mandatory for any operation potentially taking more than 50ms to complete. This is done to ensure smooth UI performance which is especially important for touch-based user interfaces like Metro.

WinRT is built on top of an advanced version of COM. Below are some differences between WinRT and COM:

- Objects are created using `RoCreateInstance`.
- All objects implement the `IInspectable` interface which in turn derives from the familiar `IUnknown` interface.
- Supports .NET-style properties, delegates and events (instead of sinks).
- Supports parameterized interfaces (“generics”).
- Uses the .NET metadata format (.winmd files) instead of TLB and IDL.
- All types derive from `Platform::Object`.

Despite borrowing many ideas from .NET, WinRT is implemented entirely in native code, so the CLR is not required when calling WinRT from a non-.NET language.

Microsoft implemented *language projections* that map WinRT concepts to language-specific concepts, whether the language in C++/CX, C# or JavaScript. For example, C++/CX is a new language extension of C++ which manages reference counting automatically, translates WinRT object activation `(RoActivateInstance)` to a C++ constructor, converts `HRESULT`s to exceptions, converts “retval” arguments to return values, etc.

When the caller and callee are both managed, the CLR is smart enough to make the call directly and there is not inerop involved. For calls that cross a native to managed boundary, regular COM interop is involved. When both the caller and callee are implemented in C++, and the callee’s header files are available to the caller, no COM interop is involved and the call is very fast, otherwise, a COM `QueryInterface` needs to be done.