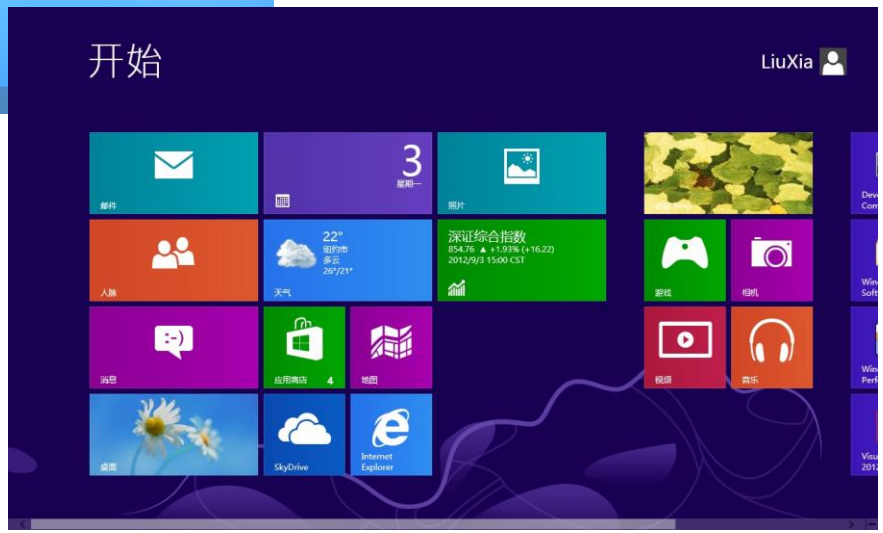
Windows RT Introduction

Part1 - The Architecture

Windows 8 Metro & Desktop

METRO_MONITOR_MODE::
MMM_DESKTOP



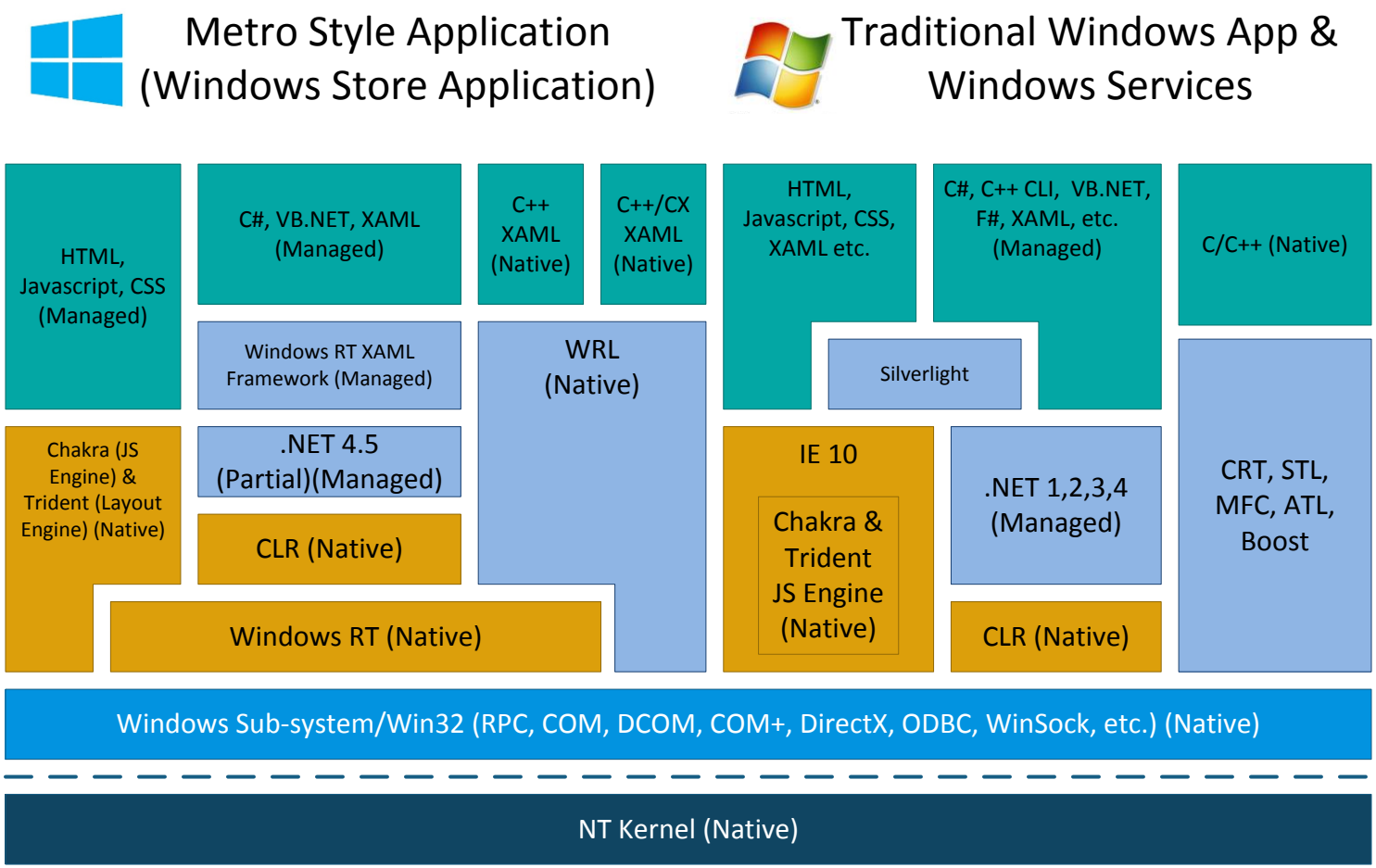
METRO_MONITOR_MODE::
MMM_METRO

Difference Between Metro & Desktop App

Display	Life Cycle	Interaction
Message loop based GDI or DirectX paint.	Managed by system configuration (windows service) or manually (windowed application).	Complex tasks, precise control.
No message loop. Pure DirectX powered graphics.	Managed by system resource manager.	Simple task, touch oriented, manual free.

Share & Communication	Portablility	OS Access
Named pipe, MM file, Socket, etc. All the way you want.	x86, x64	Opened
16 built-in contracts. (e.g. account picture provider, autoplay, camera, print)	x86, x64, ARM Processor	Restricted

The Windows API Architecture



Win RT is JUST for Metro App

- Win RT is designed for client metro application.
- Win RT is aimed at Metro style client application deployed on Windows 8 / Windows 8 ARM, and possibly Windows Phone 8.
- Win RT is for simple deployment, no shared library will be deployed, nor does inter-process communications.

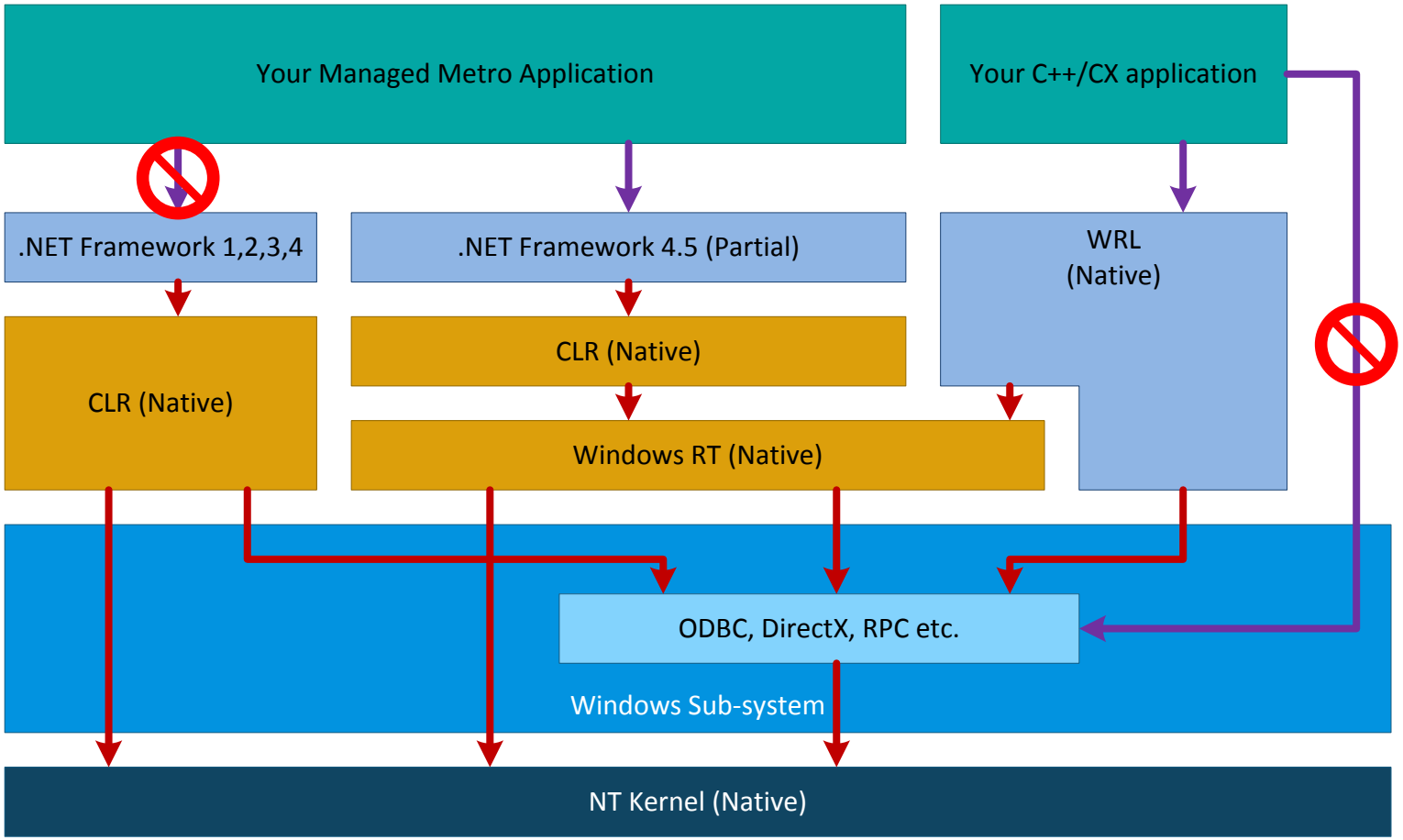
Win RT Relies on COM

- Win RT is reinvention of Window API.
- The WRL is the first-class library to consume Win RT, however, you can use ATL or any technique that supports COM programming to consume Win RT.

Win RT Restricts Direct Access To Windows sub-system

- The safest way to make system call is going through Win RT.
- Most traditional Win32 API is not support and only partial of .NET framework 4.5 is supported.
- Making system call in a metro application (either using C to make direct call or using P-Invoke under .NET Framework) is ok technically, while may cause runtime exception or be prohibited by Windows Store.
- Considerable effort may be paid even if you have large native COM codebase.
- Using C++/CX seems like the best way to develop new metro style application (rapid, portability, efficiency, accuracy, library safe).

Explain



Why C++/CX

```
HSTRING hstring;
HRESULT hr;
IInspectable *pInspApp;
const wchar_t *appClassName = L"Windows.UI.Xaml.Application";

hr = ::RoInitialize(RO_INIT_MULTITHREADED);
CHECK_RESULT(hr);

hr = ::WindowsCreateString(appClassName, static_cast<UINT32>(::wcslen(appClassName)), &hstring);
CHECK_RESULT(hr);

hr = ::RoActivateInstance(hstring, &pInspApp);
::WindowDeleteString(hstring);
CHECK_RESULT(hr);

Windows::UI::Xaml::IApplication *pApplication;
hr = pInspApp->QueryInterface(__uuidof(pApplication), reinterpret_cast<void**>(&pApplication));
pInspApp->Release();
CHECK_RESULT(hr);

pApplication->Run();
CHECK_RESULT(hr);
```

VS

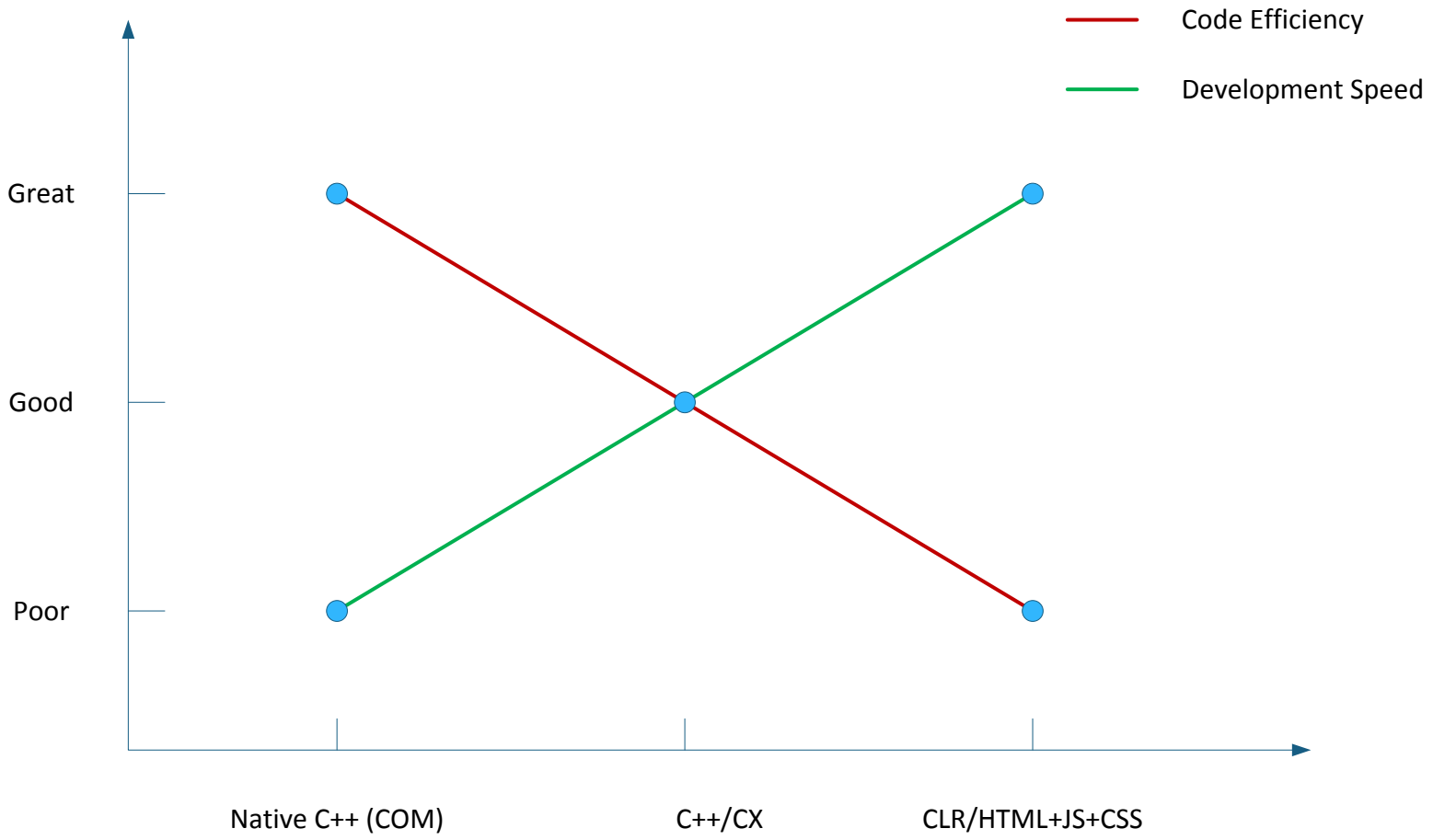
```
int main(lang::array<Platform::String^>^ args) {
    auto app = ref new App();
    app->Application::Run();
}
```

C++/CX is NATIVE

- C++/CX (Native) and C++ CLI (Managed) are almost syntactically the ~same~.
- No garbage collector exists. The “ref class” is reference counted. The delete logic is wrapped just like “Visual Basic 6”.
- Inter-reference might exist and cause memory leak. Weak Reference to the rescue.

- Personally, I think C++/CX is introduced because we cannot use visual basic 6 in Visual Studio 2012 :-).

Method Comparision



Cheat Sheet

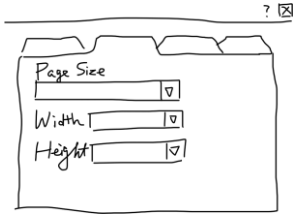
1. Support early versions of Windows OS.
2. Complex operations, precise control.
2. Used as Windows services.
3. Heavy IPC interactions.

Desktop App

1. Support Windows 8 under different processors (ARM incl.)
2. Portable to Windows Phone 8.
3. Simple and have rich client experience.

Metro App

1. Rapid prototype development.
2. Efficiency is not that critical.
3. Have large CLR codebase.



C#/HTML+JS+CSS

1. Critical on performance and battery efficiency.
2. Heavy graphics operations (e.g. Games).
3. Precise memory management.
4. Cross compiler code.

C++ or C++/CX

References

- Win RT: An Object Orientated Replacement for Win32. http://www.infoq.com/news/2011/09/WinRT-API
- An Accurate Windows 8 Platform Architecture Diagram?. http://www.bitcrazed.com/post/2012/01/27/An-Accurate-Windows-8-Platform-Architecture-Diagram.aspx
- Win8, Metro Apps and Silverlight. http://wildermuth.com/2011/9/20/Win8_Metro_Apps_and_Silverlight
- A bad picture is worth a thousand long discussions. http://dougseven.com/2011/09/15/a-bad-picture-is-worth-a-thousand-long-discussions/
- Windows RT Wikipedia. http://en.wikipedia.org/wiki/Windows_RT
- Windows Runtime Wikipedia. http://en.wikipedia.org/wiki/WinRT
- Windows Runtime XAML Framework Wikipedia. http://en.wikipedia.org/wiki/Windows_Runtime_XAML_Framework
- Windows 8 and Windows RT Compatible Logo Usage Guidelines. http://msdn.microsoft.com/en-us/library/windows/hardware/jj591630.aspx
- Windows API reference for Metro style apps. http://msdn.microsoft.com/en-us/library/windows/apps/br211377.aspx

Q/A