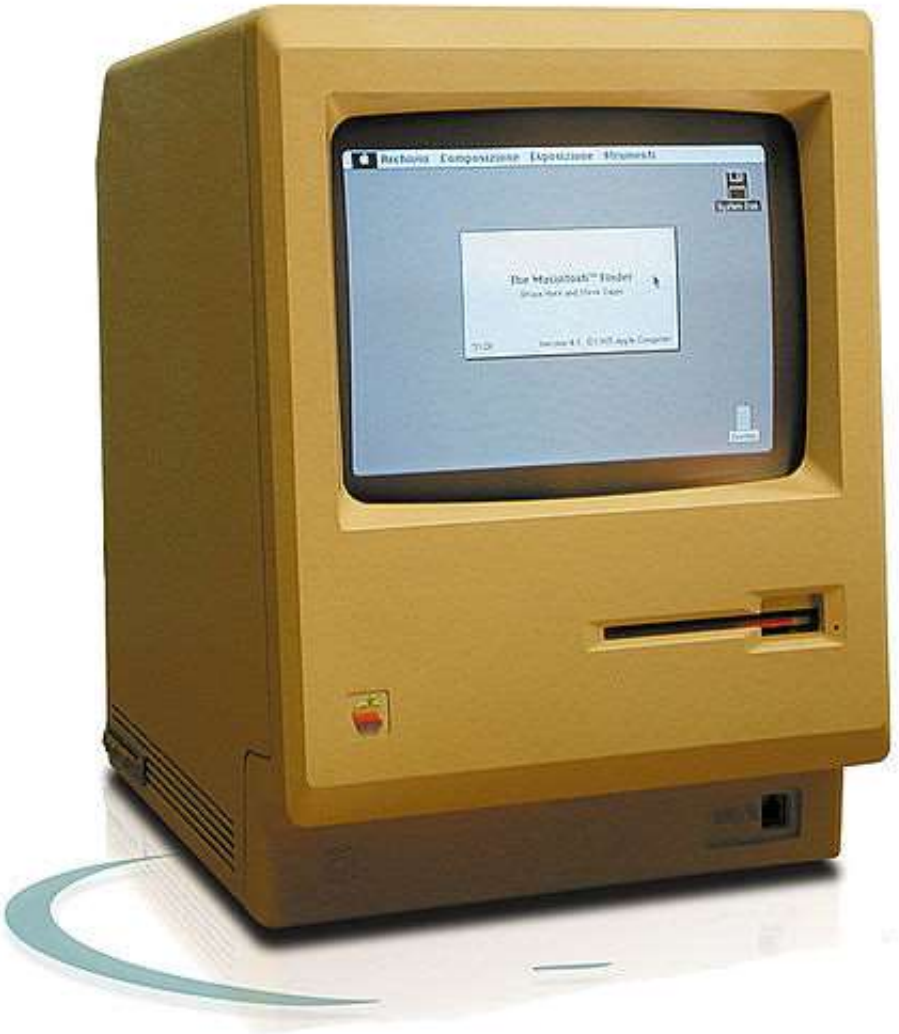
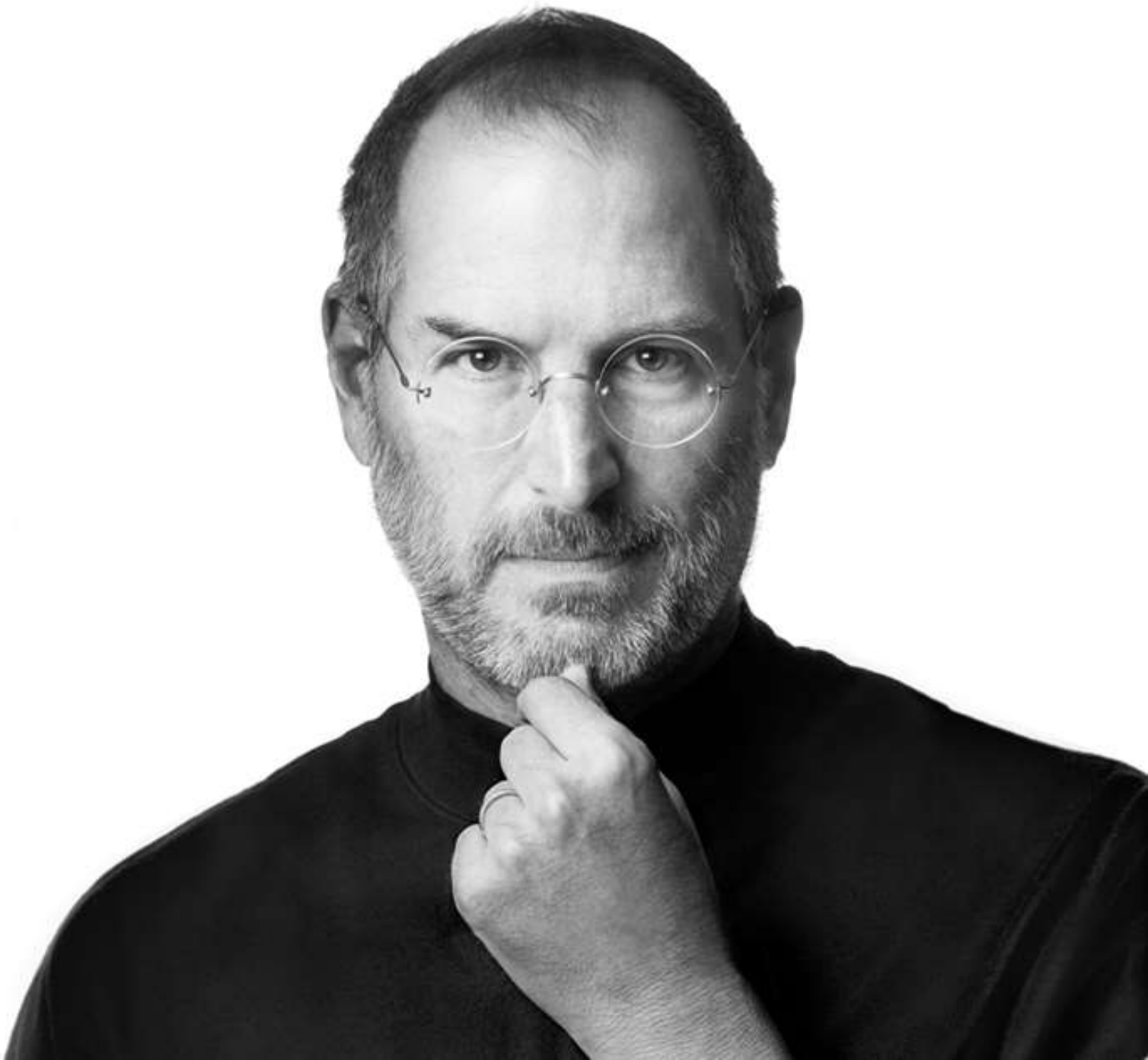
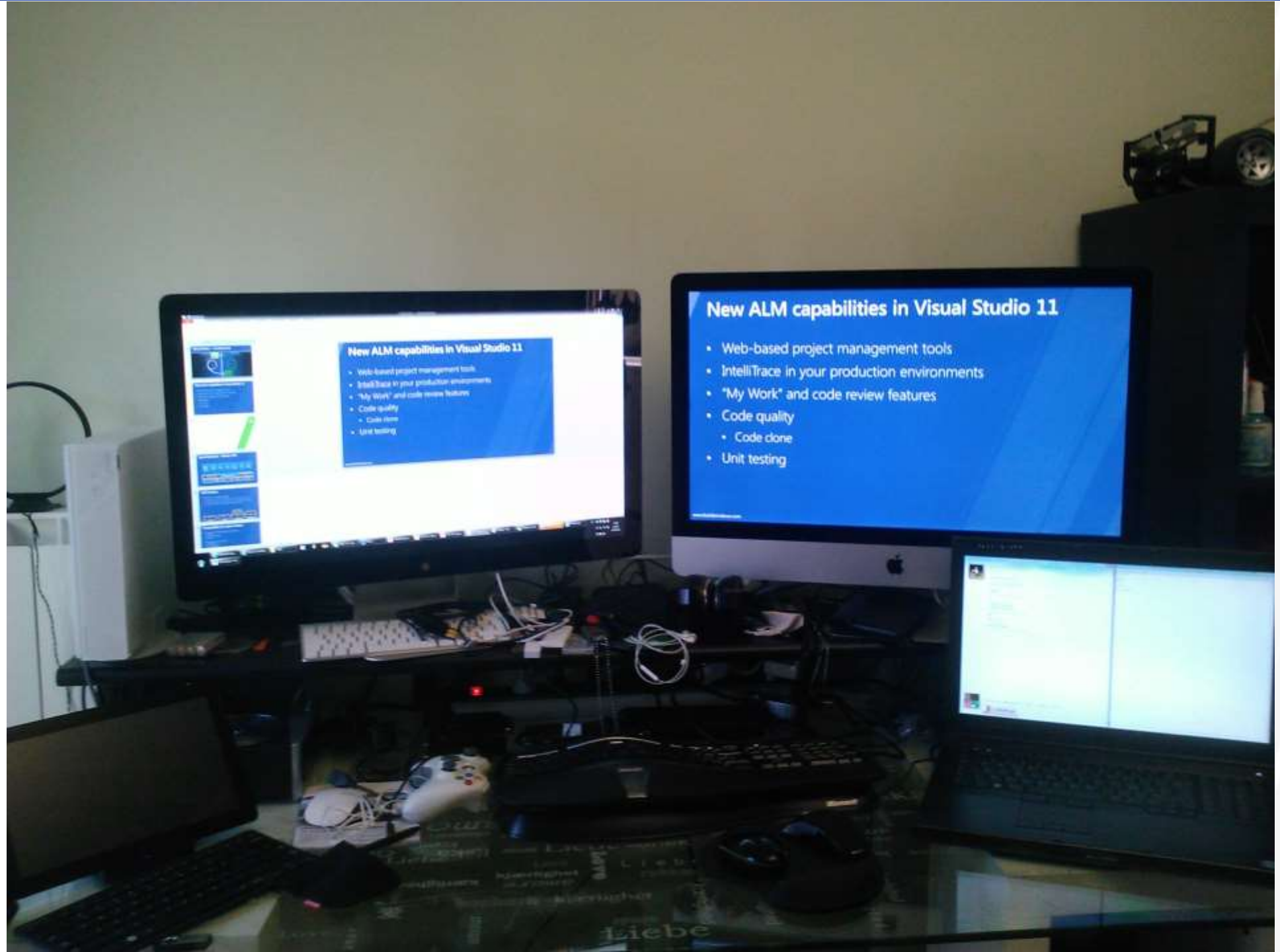

# Benvenuti e GRAZIE !!!!





# Steve Jobs

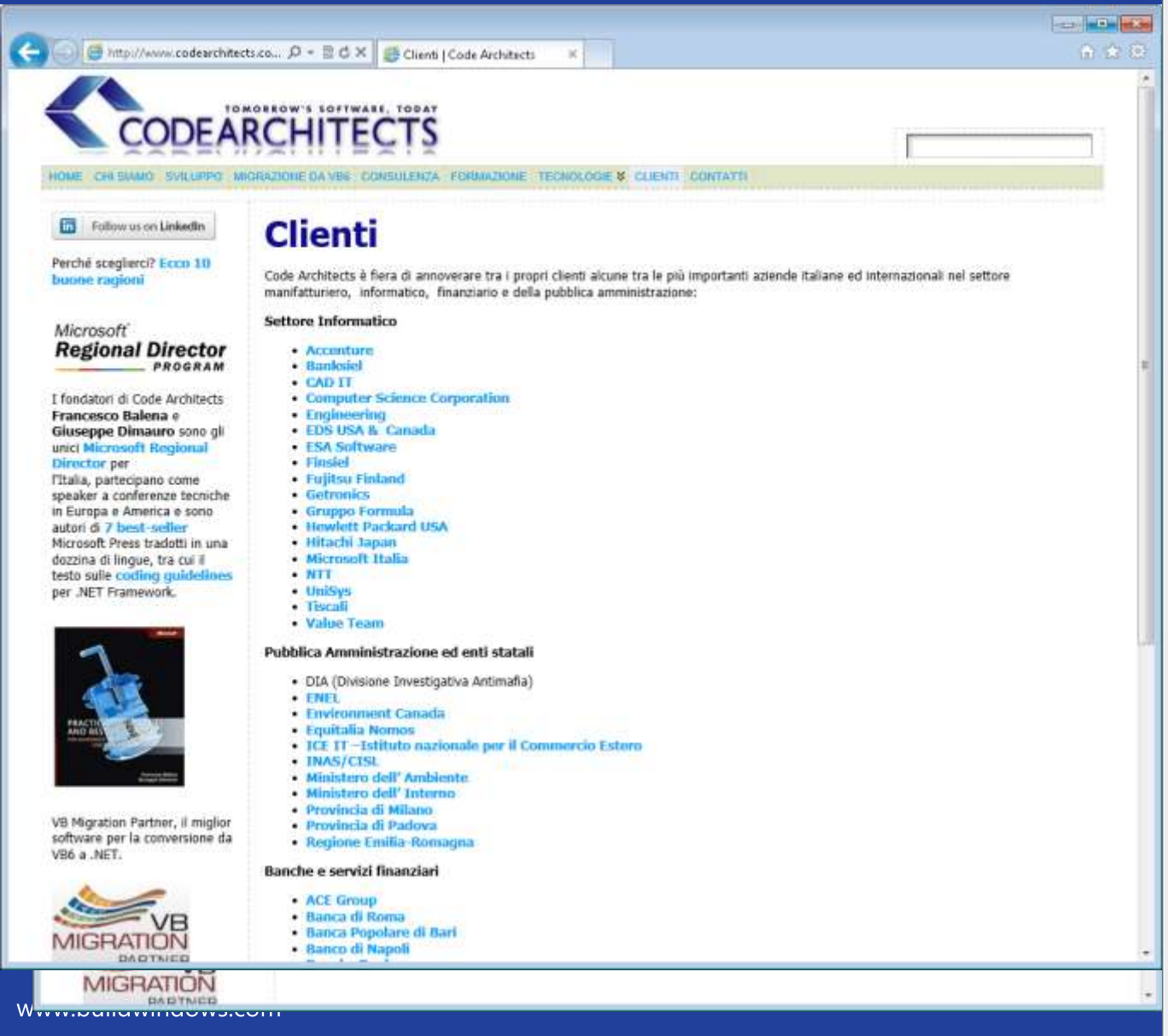
1955-2011

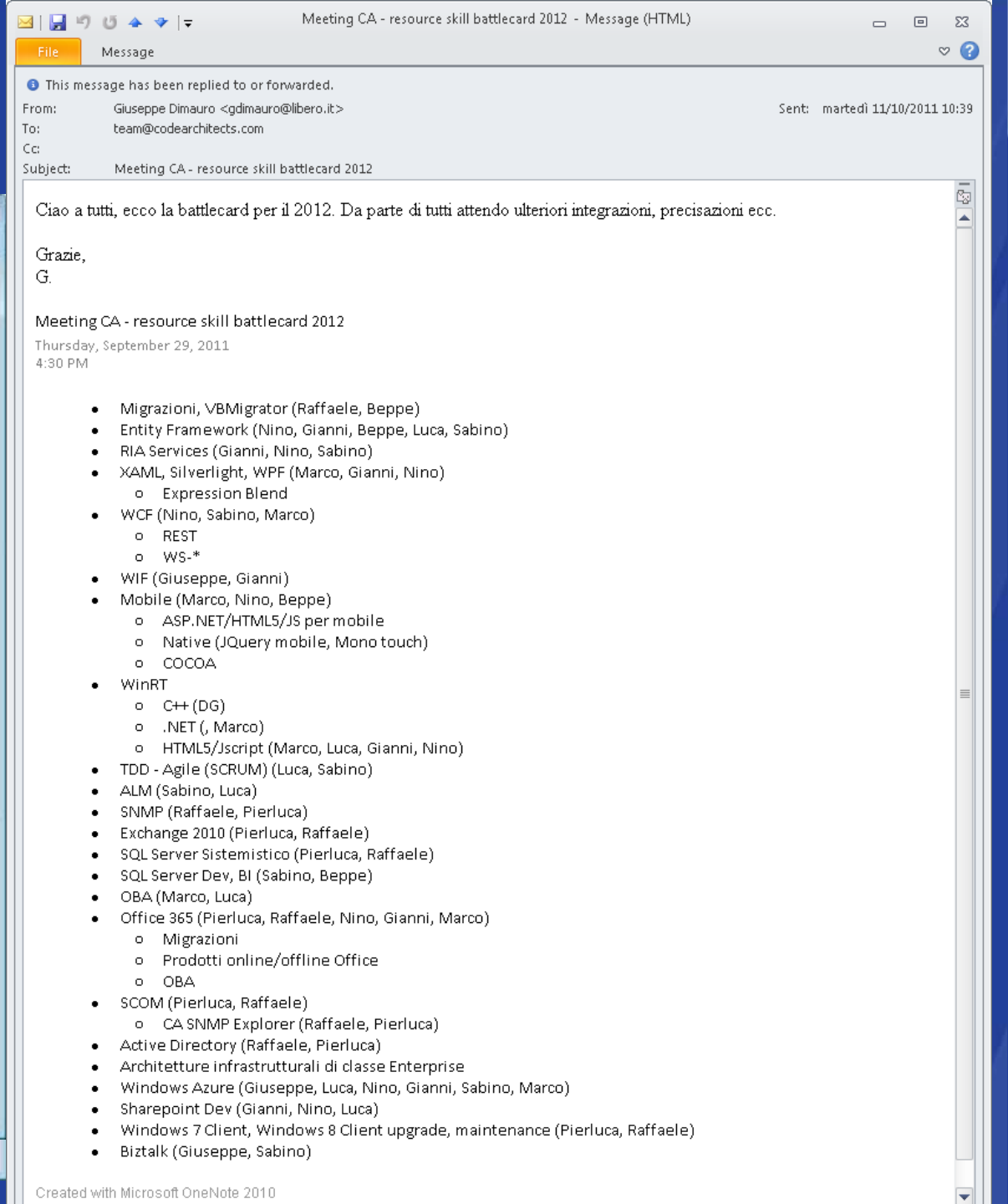

New ALM capabilities in Visual Studio 11
Web-based project management tools
IntelliTrace in your production environments
"My Work" and code review features
Code quality
Code clone
Unit testing
New ALM capabilities in Visual Studio 11
Web-based project management tools
IntelliTrace in your production environments
"My Work" and code review features
Code quality
Code clone
Unit testing

# What's new in Windows 8

Giuseppe Dimauro
CTO Code Architects srl
Microsoft MSDN Regional Director
gdimauro@codearchitects.com
http://www.codearchitects.com
http://www.vbmigration.com

# Code Architects srl



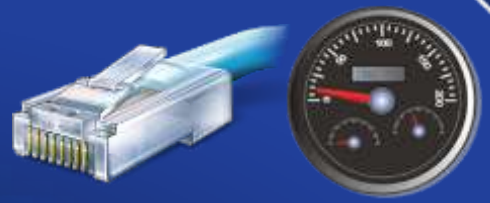
# What's new …

- … in Windows Server 2012
- … in Windows 8
- … in Visual Studio 11
- … in .NET Framework 4.5
- … in HTML5, Javascript and CSS3
- … in C++
- … inside Windows Azure Storage
- …

# *Hardware Inflection Points*

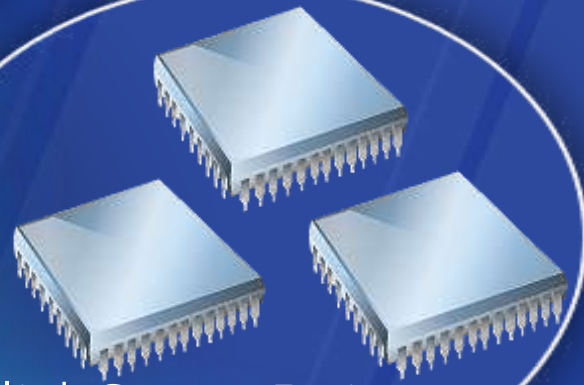
Virtualization

High Speed Networking



Multi-sock ts


Solid State Drives



# What's new in Windows Server 2012 & Windows 8

## L'esperienza del cloud a servizio delle aziende

In the **past**,

Windows focused on being a great OS

for **a server** and **its devices**

Going forward,

Windows focuses on being a great OS

for **lots of servers** and

the **devices between them**

whether they are **physical** or **virtual**

# Windows Server 8

# Windows Server ~~8~~

# Windows Server 2012 is the best Cloud Optimized OS

# 7 ways Windows Server 2012 pays for itself

- Windows Server 2012 supersaver No. 1: Storage Spaces
- Windows Server 2012 supersaver No. 2: Hyper-V 3.0
- Windows Server 2012 supersaver No. 3: PowerShell 3.0
- Windows Server 2012 supersavers No. 4: Failover clusters
- Windows Server 2012 supersaver No. 5: Data deduplication
- Windows Server 2012 supersaver No. 6: SMB 3.0
- Windows Server 2012 supersaver No. 7: Scale-out file server

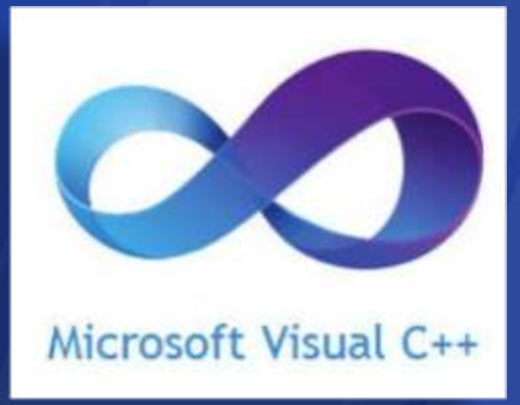
# Windows Server 2012

- Interfaccia utente Metro style UX e applicazioni
  - Tecnologie DirectX come: Direct3D, Direct2D, DirectWrite
  - Stile "Aero" ottenibile via Desktop Window Manager (DWM)
- Alta disponibilità
  - Reboot less driver upgrade, Timeout Detection e Recovery (TDR), GPU preemption e fault tolerance
- Innovazioni
  - UEFI GOP, Headless system, Collaboration e Remote Desktop Access API, DirectCompute
- Grafica sul server
  - Nuovo Windows Display Driver Model (WDDM)
  - WDDM è stato ottimizzato per scenari «Server»

# What is Graphics on Server?

- Administrator logging in to GPU for compute and server purposes
- Administrator logging in to Windows to manage a server

# C++ Accelerated Massive Parallelism




## Cosa:

- Parte integrante di C++ & Visual Studio prossima versione
- Librerie STL/Boost -like per pattern di parallelismo su grandi array
- Costruito su Direct3D

## Perchè

- Performance
- Produttività
- Portabilità

## Come:

```
#include <amp.h>
using namespace concurrency;
void AddArrays(int n, int * pA, int * pB, int * pC)
{
    array_view<int,1> a(n, pA);
    array_view<int,1> b(n, pB);
    array_view<int,1> sum(n, pC);

    parallel_for_each( sum.grid,
        [=](index<1> idx) restrict(direct3d) {
            sum[idx] = a[idx] + b[idx];
        }
    );
}
```

# Windows 8 Enterprise integration

- AppLocker
- Windows To Go
  - Startup da usb-pen
- DirectAccess
  - VPN diretta “automatica”
- BranchCache
  - Tecnologia ottimizzata per WAN
- RemoteFX virtualization
  - GPU virtualizzate e condivisibili via internet, client ecc
- Network File System
- Windows 8 Applications *Side Loading*

SAC-646T

# Manage a highly-efficient environment at scale using the Windows Management Framework (WMF)

# Server Virtualization

- Windows Server 2012 with Hyper-V provides a complete virtualization platform, providing increased scalability and performance with an elastic and flexible solution from the datacenter to the cloud. It's now easier than ever for organizations to realize cost savings from virtualization and to optimize server hardware investments.

# Server virtualization

Complete virtualization platform

Increased scalability and performance

Elastic and flexible—from the datacenter to the cloud

## TOP FEATURES OF SERVER VIRTUALIZATION TO MEET CUSTOMER CHALLENGES

1. Hyper-V Network Virtualization
2. Shared-nothing live migration
3. Massive scale
4. Cluster enhancements
5. Hyper-V Replica
6. Hardware offloading
7. Virtual Fibre Channel
8. Guest NUMA support
9. Runtime memory configuration
10. Hyper-V network switch

# Scale enhancements

| System | Resource | Maximum number | | Improvement factor |
|---|---|---|---|---|
| | | Windows 2008 R2 | Windows Server 2012 | |
| Host | Logical processors on hardware | 64 | 320 | 5× |
| | Physical memory | 1 TB | 4 TB | 4× |
| | Virtual processors per host | 512 | 1,024 | 2× |
| Virtual machine | Virtual processors per virtual machine | 4 | 64 | 16× |
| | Memory per virtual machine | 64 GB | 1 TB | 16× |
| | Active virtual machines | 384 | 1,024 | 2.7× |
| | Virtual disk size | 2 TB | 64 TB | 32× |
| Cluster | Nodes | 16 | 64 | 4× |
| | Virtual machines | 1,000 | 4,000 | 4× |

# VDI

- Virtual Desktop Infrastructure in Windows Server 2012 makes it possible for users to access IT from virtually anywhere on popular devices, providing them a rich Windows experience while ensuring enhanced data security and compliance.

# Virtual Desktop Infrastructure

- Access virtually anywhere, on any device
- Full Windows experience anywhere
- Enhanced data security and compliance

## TOP FEATURES FOR VDI

1. SIMPLIFIED USER CONNECTION
2. USER PROFILE DISKS
3. FAIR SHARE SESSION VIRTUALIZATION
4. INTELLIGENT PATCHING
5. RDS SMART CACHE
6. QUICK VDI WIZARD
7. CONCURRENT REDIRECT
8. REMOTE FX ADAPTIVE GRPHICS
9. REMOTE FX OVER WAN
10. REMOTE FX USB REDIRECTION & TOUCH

# Networking

- Windows Server 2012 provides an array of new and enhanced features that help reduce networking complexity, making it easier to connect users to IT resources, efficiently manage your datacenter and private clouds, and easily link your infrastructure with public cloud services.

# Networking

Easy to connect users to IT resources

Efficient management of datacenters and private clouds

Ability to link your infrastructure with Public Cloud Services

## TOP FEATURES OF NETWORKING TO MEET CUSTOMER CHALLENGES

1. NIC Teaming
2. Network virtualization
3. DHCP server failover
4. Single Root I/O Virtualization (SR-IOV)
5. Resource Metering
6. Dynamic Virtual Machine Queue
7. IP Address Management (IPAM)
8. Quality of Service (QoS)
9. BranchCache
10. SMB Direct and Multichannel

# Identity and Access

- Windows Server 2012 makes it easier for administrators to protect corporate resources, build strong data access management and protection, and simplify the deployment of identity management infrastructure.

# Security and access

Protection of corporate resources

Data access management and protection

Simplified deployment and management of identity infrastructure

## TOP FEATURES FOR SECURITY & ACCESS

1. Dynamic Access Control
2. Classification
3. DirectAccess
4. Simpler deployment of Active Directory
5. Domain Name System Security Extensions
6. Active Directory virtualization
7. Active Directory cloning
8. Kerberos constrained delegation
9. Private virtual LAN (PVLAN)
10. Multitenant security and isolation

# Storage

- Windows Server 2012 provides greater efficiency, performance, and innovation through diverse storage choices. New features and enhancements for disk, network, and storage area network storage solutions provide greater choice and flexibility while ensuring high availability and uptime.

# Storage

Efficiency, performance, and innovation through diverse storage choices

Continuous availability through new features that preserve uptime

Cost efficiency through storage, management, and other capabilities

1 Storage spaces

2 File system improvements

3 Thin provisioning

4 Cluster Shared Volume

5 SMB 3.0 for workloads

6 Offloaded data transfer

7 Transparent failover

8 NFS support

9 iSCSI target

10 Storage management

# Server Management and Automation

- Windows Server 2012 offers excellent total cost of ownership as an integrated platform with comprehensive, multi-server manageability that provides increased management efficiency, simplified deployment of servers and server roles, and resilient automation.

# Management and automation

Increased management efficiency

Simplified deployment and virtualization

Resilient and simple automation

TO MEET CUSTOMER CHALLENGES

1. Multiserver management
2. Role and feature deployment
3. Integrated console
4. 2,400 cmdlets
5. Disconnected Sessions
6. Windows PowerShell Workflow
7. Windows PowerShell Web Access
8. Windows PowerShell ISE
9. Script Sharing
10. Syntax Simplification and IntelliSense

# Web and Application Platform

- Windows Server 2012 provides enormous flexibility through hybrid, symmetrical, and web-based applications on-premises and in the cloud, increased scalability and elasticity for applications, and enhanced support for open frameworks and open source.

# Web and application platform

Flexibility through hybrid and symmetrical applications

Increased scalability and elasticity for applications

Enhanced support for open frameworks and open source

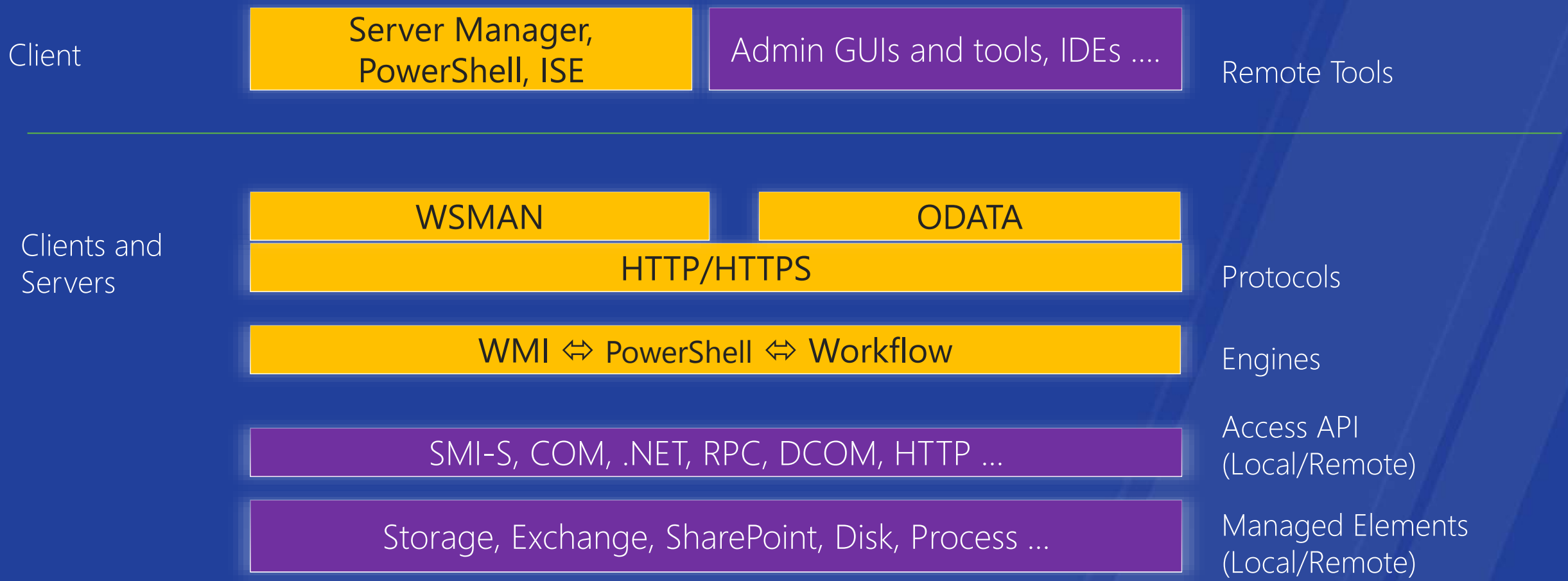
## TOP FEATURES OF WEB AND APPLICATION PLATFORM TO MEET CUSTOMER CHALLENGES

1. Website density and mutlitenancy
2. Application Initialization
3. Dynamic IP Restrictions
4. CPU Throttling
5. NUMA-Aware Scalability
6. Centralized SSL management
7. "Just works" with PHP and node.js
8. Embracing web standards
9. Application symmetry
10. Comming Development Platform

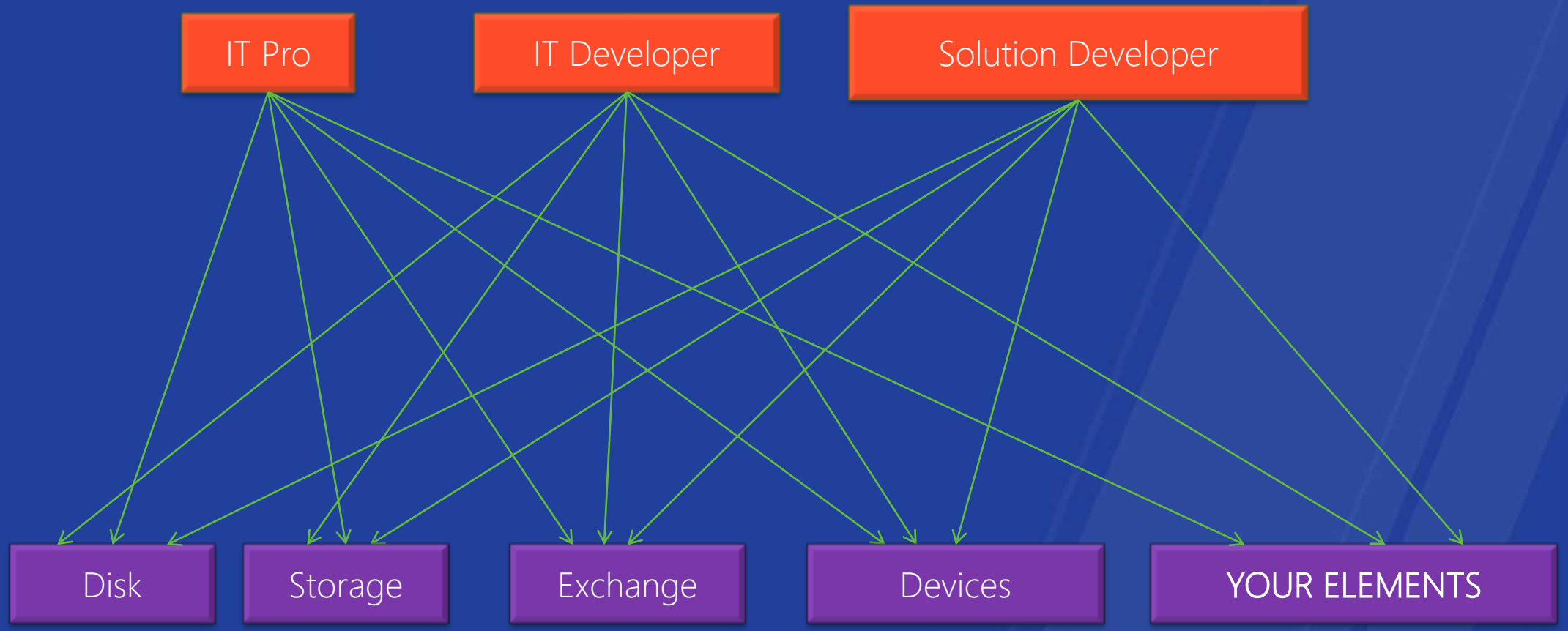
# A Cloud Optimized OS needs a different management UX

Server Manager is a thin layer on top of a platform designed to minimize the cost and effort to create robust multi-machine management solutions

# Windows Management Framework

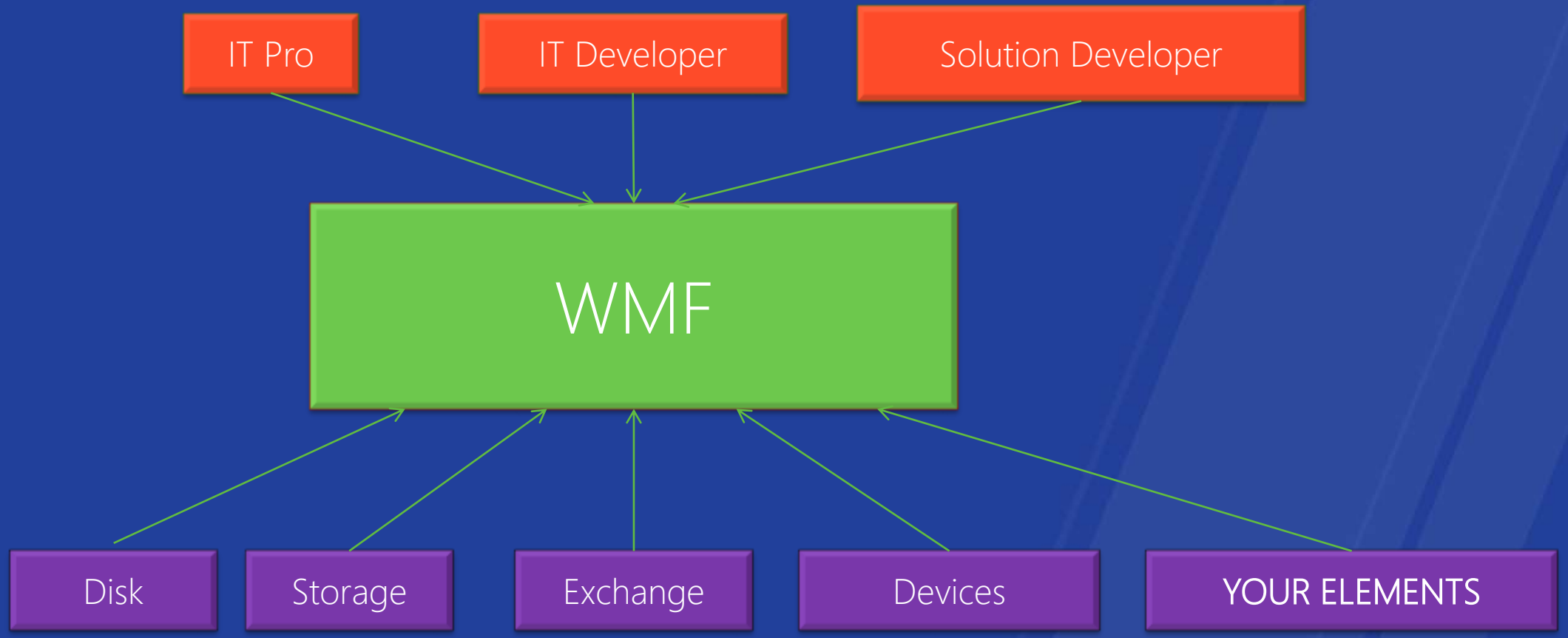


# The N x M Problem



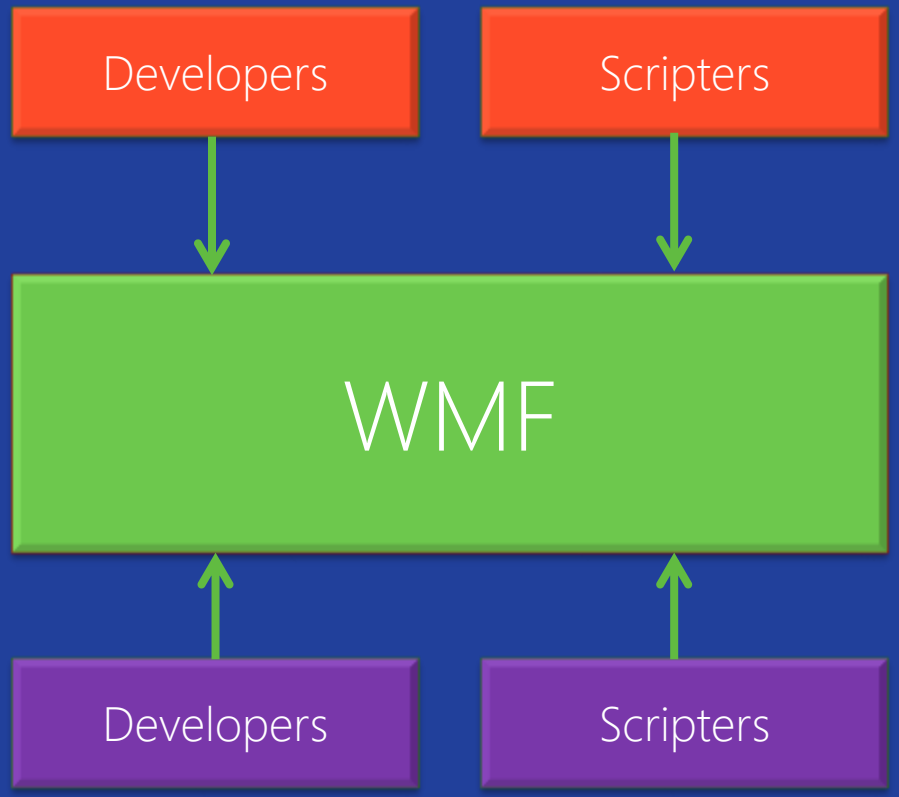
# The Solution

OR MAXIMIZING THE VALUE OF LEVERAGING THE WINDOWS MANAGEMENT FRAMEWORK



# A cloud optimized OS requires formal and informal management

# Developers AND Scripters



# Windows 8 provides the best cloud optimized management platform with great new standards-based APIs and cmdlets

# The New CIM Interface

- CIM = Common Information Model
- Simple API and cmdlets to manage standards-based devices
  - Windows and/or non-Windows
- Native and managed
- Supports both WSMan and DCOM
- Follows the latest .NET framework design patterns:
  - For async: Supports IObservable <T> / integration with Rx (Reactive Extensions Framework)

# Old WMI API – Query Operation

```
//Connect to remote computer
ManagementScope scope = new ManagementScope("\\\\RemoteComputer\\root\\cimv2");
scope.Connect();
// Create a results watcher object and handler for results and completion
ManagementOperationObserver results = new ManagementOperationObserver();
// Attach handler to events for results and completion
results.ObjectReady += new  ObjectReadyEventHandler(this.NewObject);
results.Completed += new     CompletedEventHandler(this.Done);

//Build a query system for Operating System information
SelectQuery query = new SelectQuery("Win32_Process");
// Instantiate an object searcher with this query
ManagementObjectSearcher searcher = new ManagementObjectSearcher(scope,query);
// Call the asynchronous overload of Get to start the enumeration
searcher.Get(results)
//////////////////Call backs for Async Operations ///////////////
private void NewObject(object sender,  ObjectReadyEventArgs obj)
{
}

private void Done(object sender, CompletedEventArgs obj)
{
}
```

# New CIM API – Asynch Query Operation

```
//Connect to remote computer
CimSession session = CimSession.Create("RemoteComputer");

//Perform the async operation – uses IObservable pattern
IObservable<CimInstance> results = session.QueryInstancesAsync(
            "root\Cimv2", "WQL",  "Select * from Win32_Process");

results.Subscribe(this);

//////////////////Standard IObserver implementation //////////////////
void IObserver<CimInstance>.OnCompleted()
{
}
void IObserver< CimInstance >.OnError(Exception error)
{
}
void IObserver< CimInstance >.OnNext(CimInstance value)
{
}
```

# New CIM API – Query Operation (DCOM)

```
//Connect to remote computer
CimSession session = CimSession.Create("RemoteComputer", new DcomSessionOption())

//Perform the async operation – uses IObservable pattern
IObservable<CimInstance> results = session.QueryInstancesAsync(
            "root\Cimv2", "WQL",  "Select * from Win32_Process");

results.Subscribe(this);

//////////////////Standard IObserver implementation //////////////////
void IObserver<CimInstance>.OnCompleted()
{
}
void IObserver< CimInstance >.OnError(Exception error)
{
}
void IObserver< CimInstance >.OnNext(CimInstance value)
{
}
```

# Windows PowerShell Workflow

- IT Pros often automate the management of their multicomputer environments by running sequences of long-running tasks or workflows that can affect multiple managed computers or devices at the same time. Windows PowerShell Workflow lets IT Pros and developers apply the benefits of workflows to the automation capabilities of Windows PowerShell.

# Windows PowerShell Workflow

- A workflow is a sequence of automated steps or activities that execute tasks on or retrieve data from one or more managed nodes (computers or devices). These activities can include individual commands or scripts. Windows PowerShell Workflow helps IT Pros and developers to author sequences of multi-machine management activities (which usually are long-running, repeatable, frequent, parallelizable, interruptible, stoppable, or restartable) as workflows. By design, workflows can be resumed from an intentional or accidental suspension or interruption, such as a network outage, reboot, or power loss.

# Benefits of Windows PowerShell Workflow

- Windows PowerShell Workflow manages the distribution, sequencing, and completion of multicomputer tasks, freeing users and administrators to focus on higher level tasks. The following list describes some of the benefits of Windows PowerShell Workflow:

# Benefits of Windows PowerShell Workflow

- Take advantage of the PowerShell scripting syntax
  - IT Pros can reuse their existing PowerShell scripting skills to author script-based workflows using the extended PowerShell language. Apart from being easy to author, PowerShell script-based workflows provide the additional benefit of sharing by simply pasting them into an email or publishing them online.

# Benefits of Windows PowerShell Workflow

- Multi-machine management
  - Simultaneously run long-running tasks as workflows on up to hundreds of managed nodes. Windows PowerShell Workflow includes a built-in library of common management parameters for workflows, enabling multi-machine management scenarios such as PSComputerName and PSConfigurationName.

# Benefits of Windows PowerShell Workflow

- Take advantage of the PowerShell scripting syntax
- Multi-machine management
- Single task execution of complex processes
- Robustness: Automated failure recovery
- Persistence
- Connection and action retries
- Ability to connect and disconnect
- Scheduling
- Workflow and connection throttling

# When to use

- When to use Windows PowerShell Workflow instead of a cmdlet/script …
  - You need to perform a long-running task that combines multiple steps in a sequence.
  - You need to perform a task that runs on multiple computers.
  - You need to perform a task that requires checkpointing or persistence.
  - You need to perform a long-running task that is asynchronous, restartable, parallelizable, or interruptible.
  - You need the task to run at scale or in high availability environments, potentially requiring throttling and connection pooling.

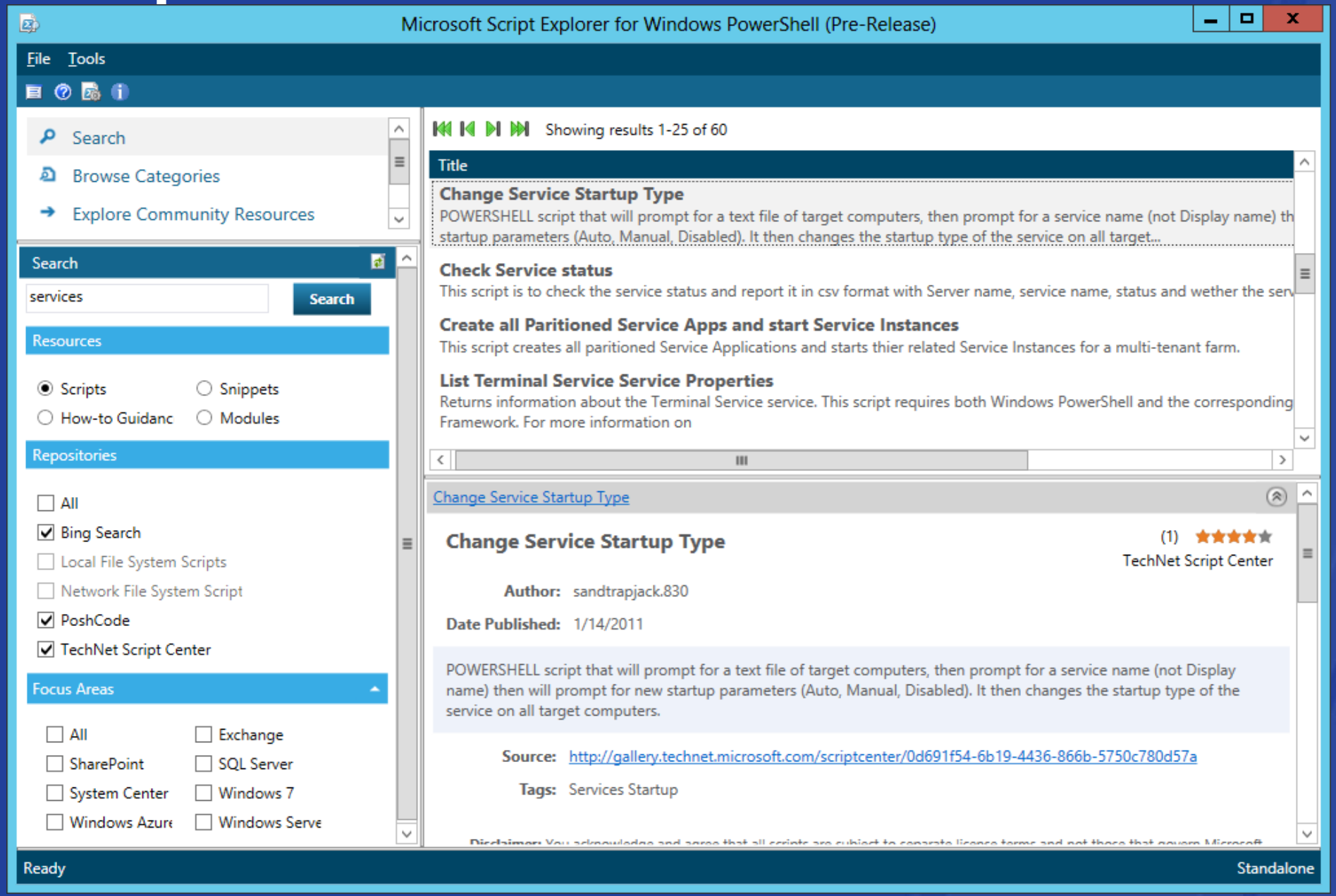
# WF Long-running sample

```
Workflow Invoke-LongWorkflow
{
        Write-Output -InputObject "Loading some information..."
        Start-Sleep -Seconds 10
        Write-Output -InputObject "Performing some action..."
        Start-Sleep -Seconds 10
        Write-Output -InputObject "Cleaning up..."
        Start-Sleep -Seconds 10
}
```

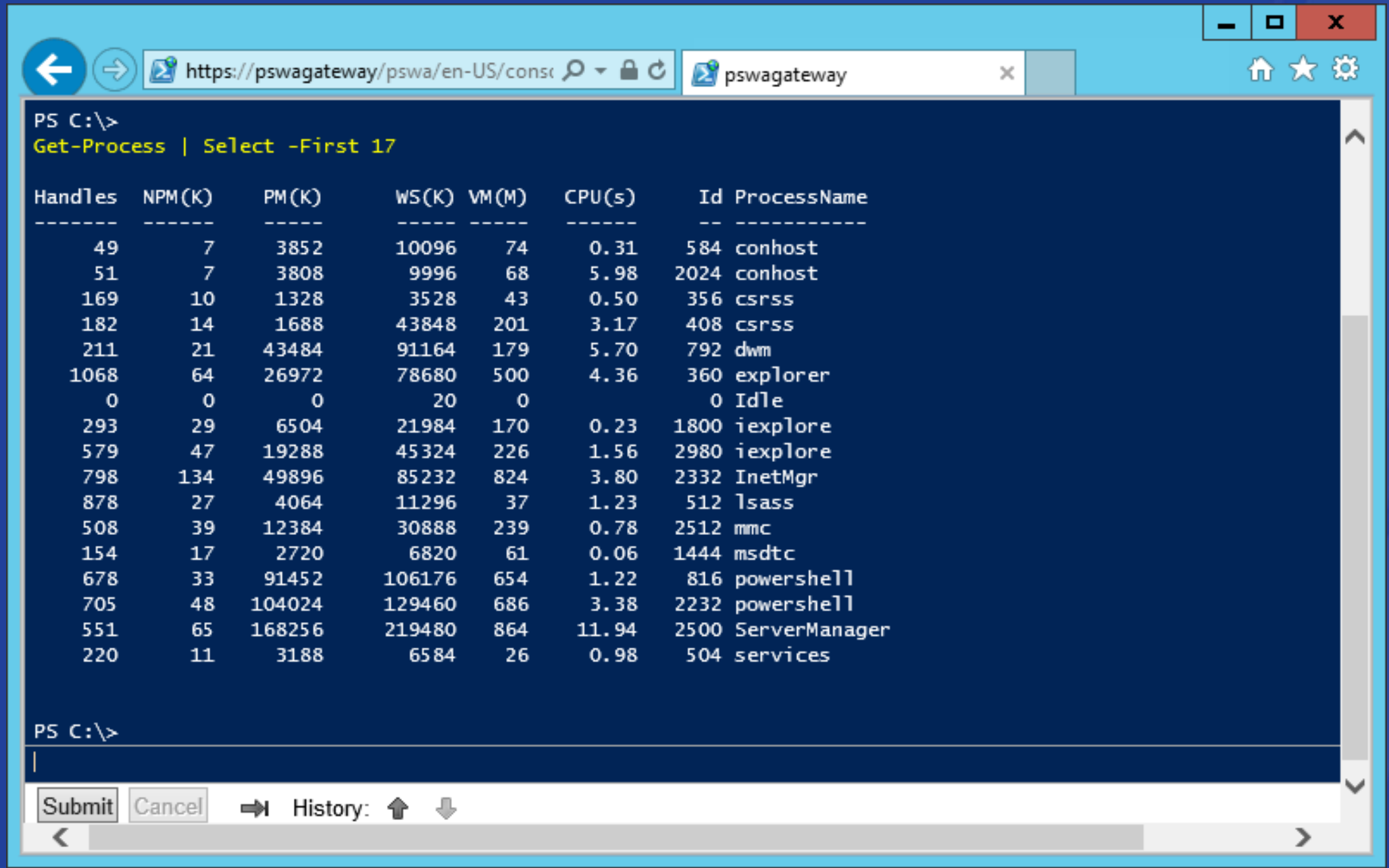
*Running as a job:*

```
Invoke-LongWorkflow –AsJob –JobName LongWF
```

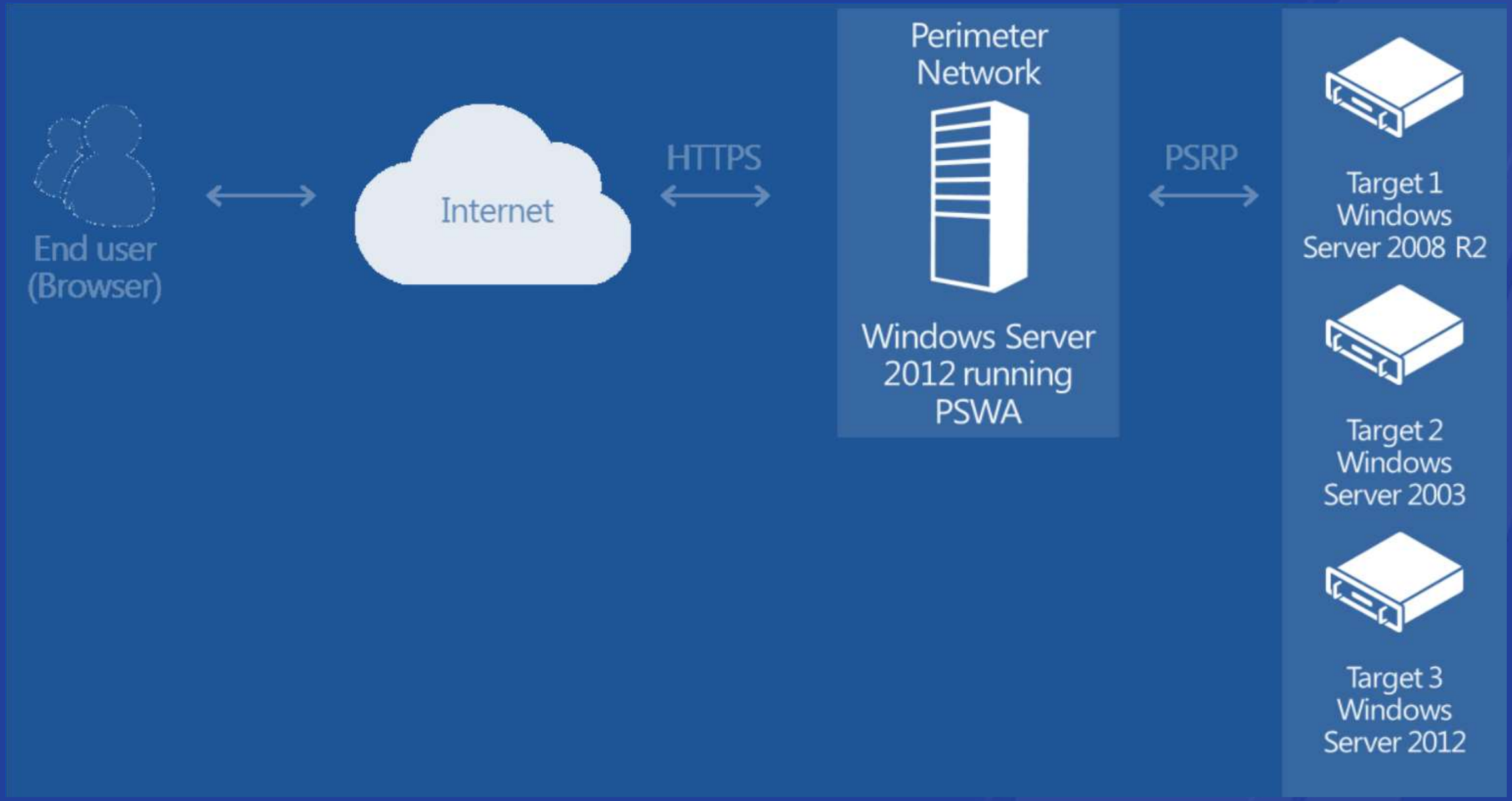
# Script Explorer



# Powershell Web Access



# Powershell Web Access Infrastructure



Windows 8 makes it easy to develop solutions of new generation targeting devices from smart phones through servers

# The PowerShell Interface

- Managed API and cmdlets provide rich, multi-machine automation
- V2 provided rich 1-1, fan-out and fan-in remoting
- New APIs and Cmdlets focus on minimizing the cost of producing robust cloud automation:
  - Robust connections
  - Remote Connect / Disconnect
  - Control workflows
  - Scheduling jobs
- On-the-fly compilation allows PowerShell scripts to run up to 6x faster

# Windows 8 minimizes the effort to build multi-machine workflow-based solutions

# Programing Workflows

- Same API and Cmdlets as PowerShell
- Connect to a different ConfigurationName
  - microsoft.powershell.workflow
- Workflows are exposed as Cmdlets with
  - A lot of new parameters
  - Rich control semantics including Suspend and Resume
    - Workflows can survive system reboots!

PowerShell jobs allow you to uniformly coordinate the activities of multiple remote heterogeneous management engines

# New Job API

- Single job manager/multiple job sources
  - Registering a new JobSourceAdapter (JSA) with the job manager makes all *-Job cmdlets work with custom job types
- JobSourceAdapter provides custom job implementations:
  - Windows 8: Workflow jobs, scheduled jobs
  - Create your own. E.g. products that support job scheduling, workflow, long running tasks
- JSA is based on a new abstract class (Job2)
- New for Windows 8: Suspend / Resume

# New Job API - continued

- Implement details, such as how to load and create a running instance of a job in the JSA
- Implement a JSA which gets job definition info from arbitrary sources, such as Web Services, databases, …
- Package the JSA as a PowerShell module.
- Importing the module automatically registers all JSAs

# Cloud management solutions need to manage from down-level OSs and need to manage everything

# Windows Management Framework

- Intend to make it available on W7, WS08, WS08/R2
- PowerShell remoting will always work
  - From the lowest/highest invoker to the lowest/highest device
  - E.g. V2 running on XP can manage W8, W8 can manage XP
- W8 will be the best managed OS
  - Small set of cmdlets and WMI providers will go down-level

# NanoWBEM* now named Open Management Interface (OMI)

- NanoWBEM is Microsoft's freely licensable CIM server for Linux - so PowerShell can manage non-Windows machines & devices.
  - CIM = Common Information Model
  - Portable, next-generation CIM server (or CIMOM) that implements key WBEM standards
  - Written by MSFT on Linux for support non-Windows Systems
  - Common provider interface with WMI
  - Readily licensable to others

*Web Based Enterprise Management

# NanoWBEM vs. OpenPegasus

| | OpenPegasus | NanoWBEM | Factor |
|---|---|---|---|
| Requests per sec. (binary protocol) | 260 | 20,000 | 76x |
| Image object size | 8,000 KB | 150 KB | 53x |
| Trivial provider size | 30 KB | 3 KB | 10x |
| Virtual memory size | 54 MB | 1.5 MB | 36x |
| Resident set size | 8,500 KB | 500 KB | 17x |

# Call To Action

Leverage CIM, PowerShell and Workflow to build powerful, robust, multi-machine heterogeneous management solutions

- With code using our APIs
- With Cmdlets using scripting
- With a combination

# Building great Windows 8 systems

# The expanding set of PC options

| | | | |
|---|---|---|---|
| Processing & Graphics | Power & Headroom | ⟷ | Low-power components |
| Storage | A breadth of options | ⟷ | Non-rotational storage |
| Idle power states | Sleep/Hibernate | ⟷ | Connected standby |
| Primary input | Mouse & Keyboard | ⟷ | Touch, pen |
| Sensors | Few/none | ⟷ | Light, motion, orientation, presence |

Windows 8 is designed to run efficiently on the widest range of PCs.

With powerful features
for every PC form factor.

# Windows 8: Nuove aspettative

- Fast and fluid touch experience
- Long battery life
- Fun and intuitive
- Connecting to devices
- Confident and safe from malware

# Fast and fluid touch

# Fast and fluid touch experience

- System operations are effortless
- Panning gestures are smooth
- Multipoint manipulations are immediate
- Targeting with fingers is precise
- Typing on the screen is quick and efficient
- App behavior is consistent

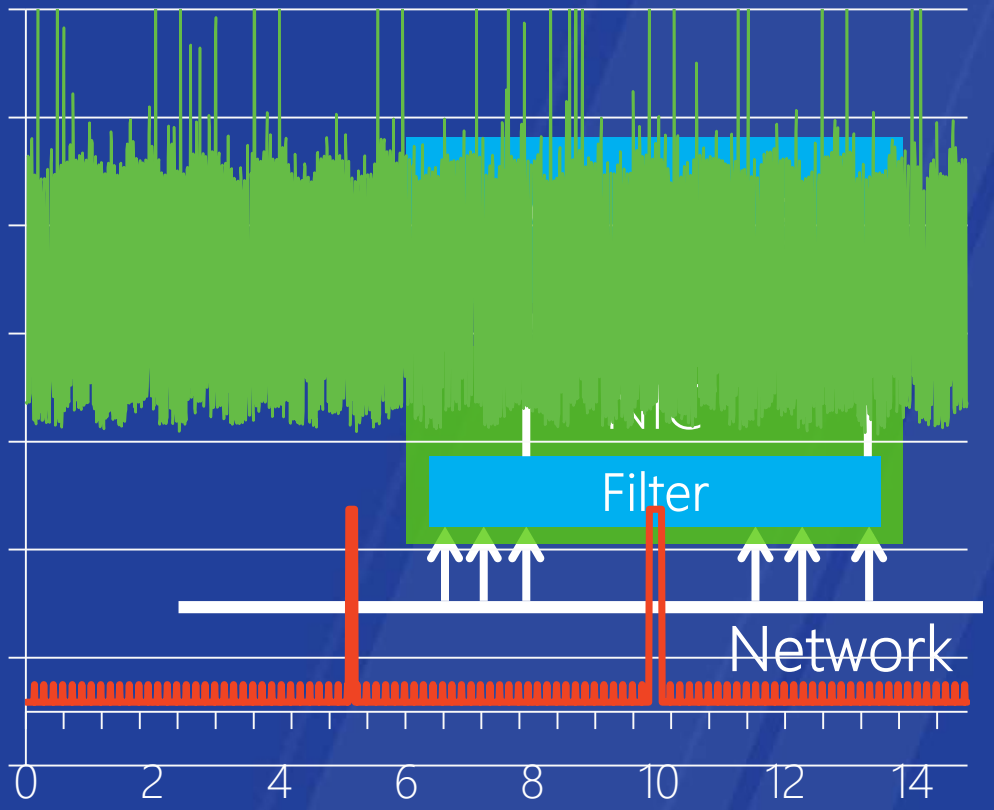
# Long battery life

# Durata della batteria

- Almeno una giornata di batteria con uso intensivo delle applicazioni
- Riuscire a mantenere la connessione internet aperta anche con bassa carica della batteria
- Aggiornamento continuo dello stato della applicazioni
- On/Off istantaneo (sia sleep che power on-off)

# Achieving long battery life

- Connected Standby
- Network offload / coalescing
- Low power memory / components
- Low-powered busses
- Solid-state boot volumes



# Fun and Intuitive

# PCs that are fun and intuitive

- Aware of the environment
- Reattività elevata
- Applicazioni molto integrate fra loro

# Making sensors standard equipment

- Supporto nativo a sensori integrati in Windows
  - ALS, gyroscope, accelerometer, compass, IR
  - HID-compatible class driver for USB or I2C
  - Driver model for other low-power peripheral busses
- Sensor Fusion integrates input from multiple sensors to represent complex actions (motion/orientation)
- Facilmente accessibili da applicazioni Metro style

# Connecting to devices

# Customers expect device connectivity

- >10 billion USB device esistenti
- centinia di milioni di printer utilizzate
- Stampanti, mouse e altre periferiche **critiche** per la produttività quotidiana degli information worker
- Camera + PC sono ormai uno standard di fatto
- Nuovi dispositivi stanno emergendo rapidamente (es. fitness)

# New device connectivity in Windows 8


**USB 3** offers up 10x speed of USB 2.0, higher power for charging, lower power requirements


**Bluetooth LE** delivers low power wireless connectivity for longer battery life or new classes of low power devices


**Wi-Fi Direct** provides high bandwidth, secure, peer-to-peer device connectivity between PCs and devices


**Near-field Communication** is a low bandwidth, short range communication technology perfectly suited for person-to-person sharing and simplifying device pairing

# Confident and safe

# Customers expect confidence and safety

- Starts fast and is responsive
- Protection against malware
- Personal and corporate data is safe

# Delivering confidence and safety

- Fast, beautiful boot
- Reduce root/boot kit attacks
- Enhance malware detection
- Encrypt and protect user data
- Push-button reset

+ BitLocker

# What's new in Visual Studio 11

# Agenda

- overview Visual Studio 11
- Tool per Windows, Cloud & Web
- Application Lifecycle Management

# Visual Studio 11
# Experiences and Promises

user experience

developer experience

team experience

# experiences

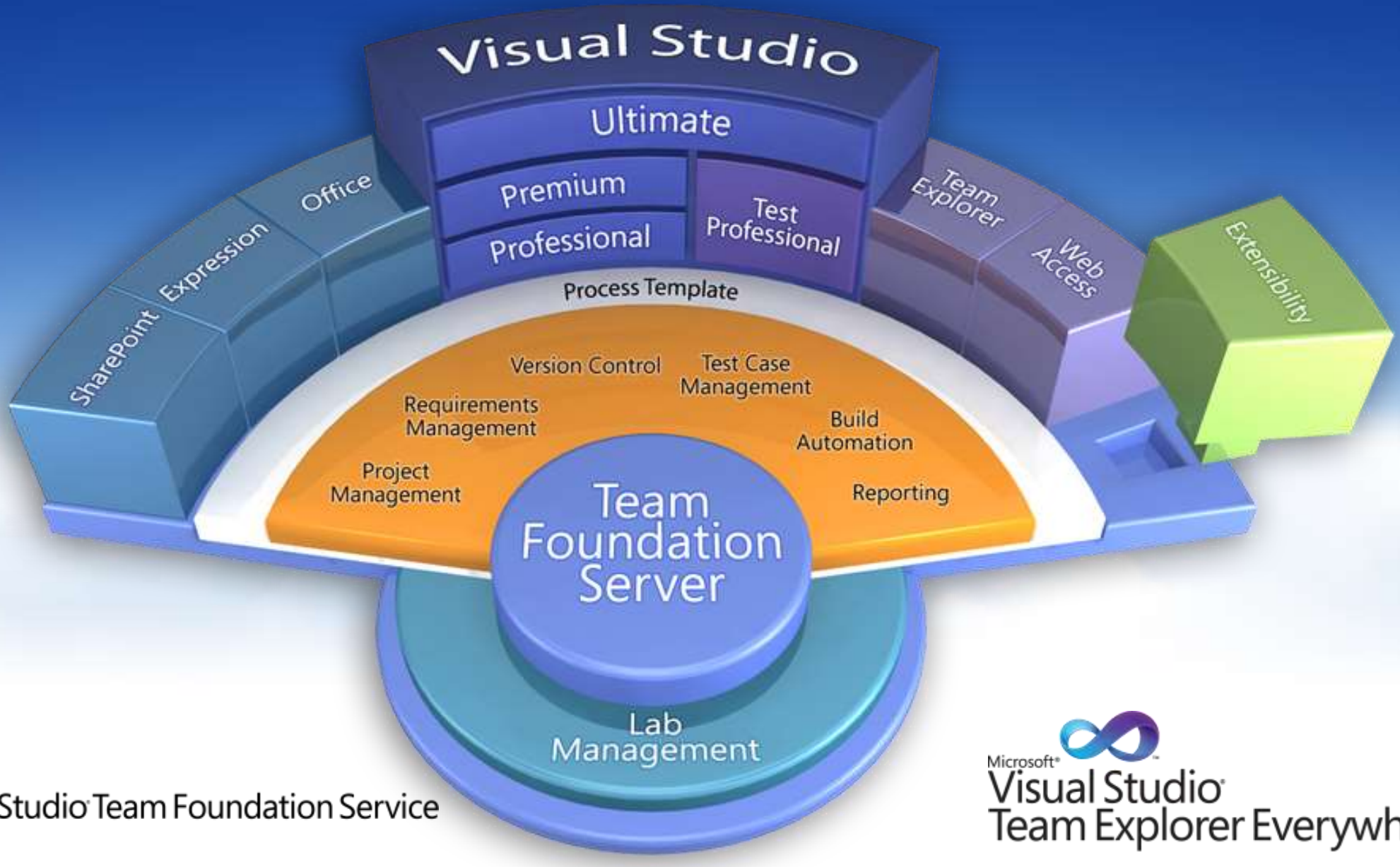
productivity

integration

platforms

# promises

# The Visual Studio Family




Microsoft Visual Studio Team Foundation Service Preview

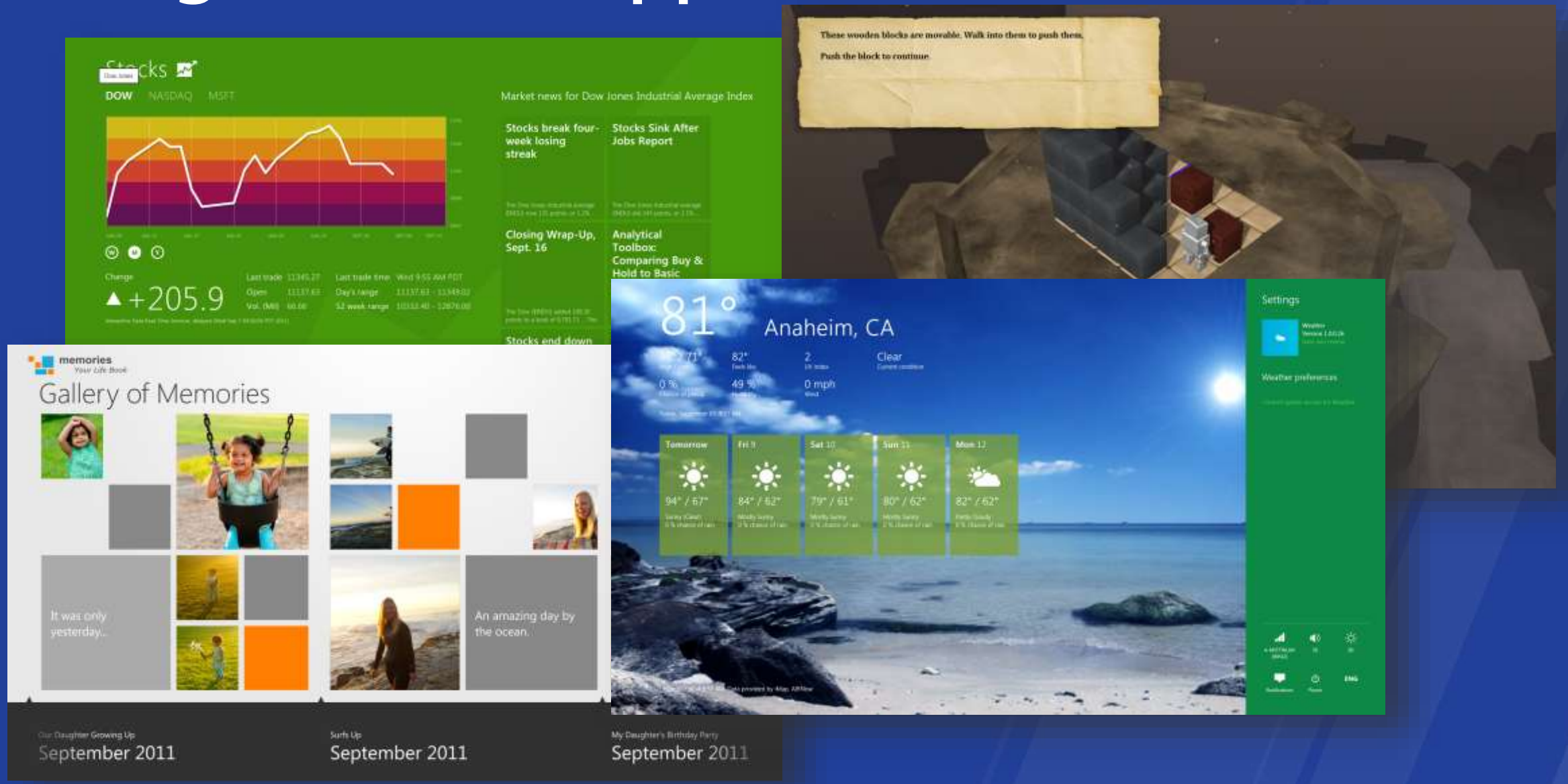
Microsoft Visual Studio Team Explorer Everywhere 2010

# Covering //build/ end to end

- Windows
  - Applicazioni Windows 8 Metro style
  - Applicazioni Windows esistenti
- Web/Cloud
  - ASP.NET
  - Azure
- ALM (Application Live Management)
  - Team Foundation Server/Team Foundation Service
  - Collaborazione e qualità

# Building for Windows

# Building Windows Applications



# Visual Studio 2012 is *the* tool for Windows 8

- Applicazioni Metro style
  - Visual Basic, C#, C++ & HTML5/JS
- Applicazioni Windows desktop
- Windows Drivers

# Windows 8 Metro style applications

- New designer Available today!
- Built on shared architecture with Expression Blend
- Key features
  - Object creation
  - Layout
  - Property editing
  - Create and reuse resources
  - Configurable design-time
  - XAML editor with IntelliSense

# Expression Blend for HTML

- Authoring of Metro style HTML Apps for Windows 8
  - Design-centric
  - Native HTML / CSS
- Solo per applicazioni Windows 8, non siti web
- Visual authoring con UI JavaScript generato dinamicamente
- Nessuna astrazione di piattaforma
  - Focus su concetti, non su sintassi
- Code – Design Workflow con Visual Studio

# Core advances for Visual Studio 11

- IDE productivity improvements
- Local, simulator and remote debugging
- Integrated application packaging
- Other topics
  - Project compatibility
  - Async
  - Localization
  - Unit testing
  - Profiling

# demo

## Windows Metro style apps

Building Windows Metro style applications

# DirectX applications

- Nuove template per giochi DirectX di tipo Metro style
- C++ AMP
  - http://blogs.msdn.com/b/nativeconcurrency/archive/2012/01/30/c-amp-sample-projects-for-download.aspx
- Integrated game asset management
  - Models
  - Textures

# DirectX applications

- IDE support for HLSL
  - IntelliSense
  - Debugging
- Integrated graphical debugging
  - Links GPU debugging with your actual game source and assets

# demo

## DirectX Applications

New tooling for building DirectX applications

# Building for the Cloud & Web

# Building Windows Azure Applications

- Windows Azure SDK 1.5
- Service Bus v2 con code
- Profile applications running in Windows Azure
- Create ASP.NET MVC3 Web Roles
- Manage multiple service configurations in one cloud project
- Improved validation of Windows Azure packages
- Add deployment projects to existing ASP.NET, MVC or WCF projects via a right click

# Use HTML5 the way that works best for you

- Programmatori HTML e JavaScript si troveranno particolarmente a loro agio con Windows 8
- IE10 è il più veloce e completo browser sul mercato
- Visual Studio 11 contiene il miglior editor CSS, JavaScript; con feature avanzate di debugging e profiling
- ASP.NET provides a set of controls and frameworks that make HTML5 development even more productive and accessible

# New HTML & CSS tool support

- IDE Enhancements for HTML & CSS
  - New HTML5 & CSS snippets
  - Extract to user control
  - Match end-tag when editing start-tag
  - Smart indent
  - Outlining
  - Comment/uncomment support
- Better IntelliSense
  - Auto reduce
  - IntelliSense for code nuggets within attributes
  - Additional IntelliSense triggers

# Better JavaScript tools

- Improved IntelliSense:
  - Performance – Built on top of IE's JavaScript engine
  - New features:
    - Go To Definition
    - Overloads for methods (using new Xml comment <signature>)
    - Implicit references
    - Extensibility – user can provide a JavaScript extension that can modify the IntelliSense
- Editing:
  - Brace matching and highlighting
  - Outlining

# Making diagnostics easier

- DOM Explorer
  - Inspect & modify the live DOM in IE or WWA
  - Style tracing for CSS
  - Layout debugging
- JavaScript console
  - Interactive REPL console against the page
  - Console API
  - Error reporting from the host
    - Syntax, parsing & network issues
  - Single and multi-line input with history

# Extending and improving ASP.NET

- Mobile
  - MVC4 will allow automatically detect on phone
  - Mobile optimized views / pages : can distinguish between slate and phone
  - Support JQuery mobile; mobile starter set; easier to theme mark-up for mobile and desktop
  - Mobile NuGet will allow mobile support
- Language & Controls
  - Supporting full ECMAScript v5
  - Better experience for using JQuery
  - JQuery controls : Datagrid etc being built by Microsoft
- Web Platform
  - MVC app support pre-built AJAX views : client side paging, sorting
  - MVC will support webforms
  - Debug while running; can select something that is running and have it highlight where code is etc
  - Added SEO capabilities

# Page Inspector

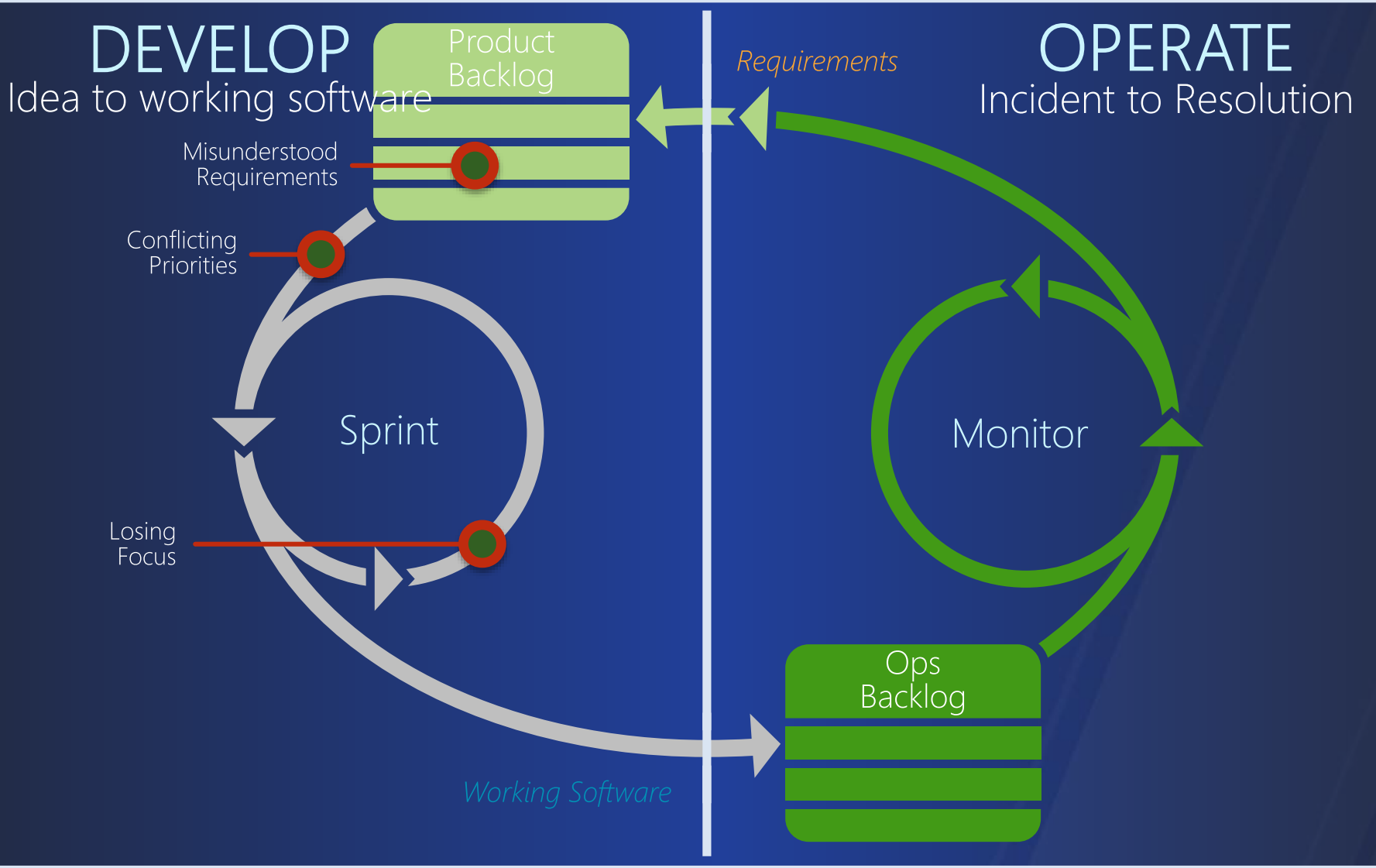
- A new tool Visual Studio 11 wave tool for troubleshooting of ASP.NET Web Sites and Applications
  - Bring together browser tools, ASP.NET, and source code into one integrated troubleshooting experience
  - Easily diagnose issues in Web Application Projects and Web Sites
  - Pinpoint the source code behind a browser selection in just one click
  - Reverse engineer HTML/CSS/JS in the browser to actual source files in just one click

# Application Lifecycle Management

# ALM is for everyone

- Application Lifecycle Management isn't just for large companies– The Agile Manifesto
  - Individuals and interactions (over process & tools)
  - Working software (over comprehensive documentation)
  - Customer collaboration (over contract negotiation)
  - Responding to change (over following a plan)

# What Matters – Creating Value



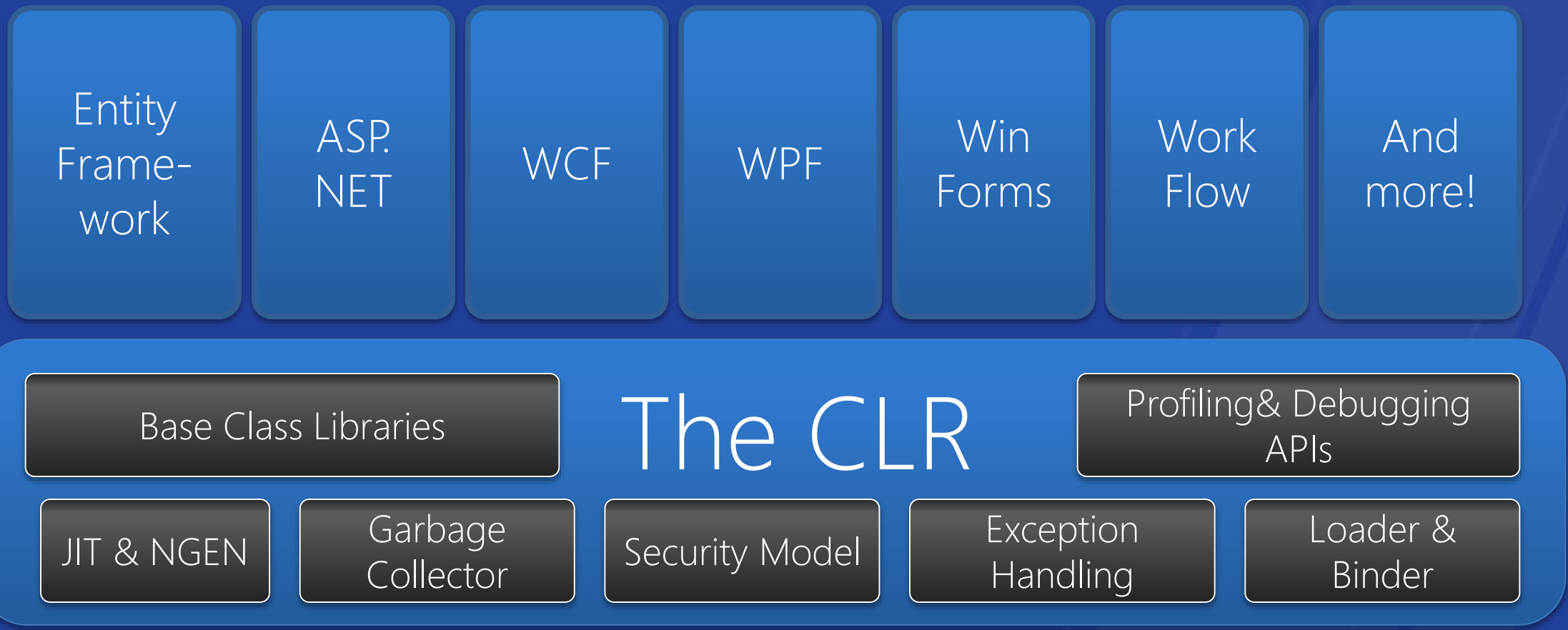
# New ALM capabilities in Visual Studio 11

- Web-based project management tools
- IntelliTrace in your production environments
- “My Work” and code review features
- Code quality
  - Code clone
- Unit testing

TOOL-834T

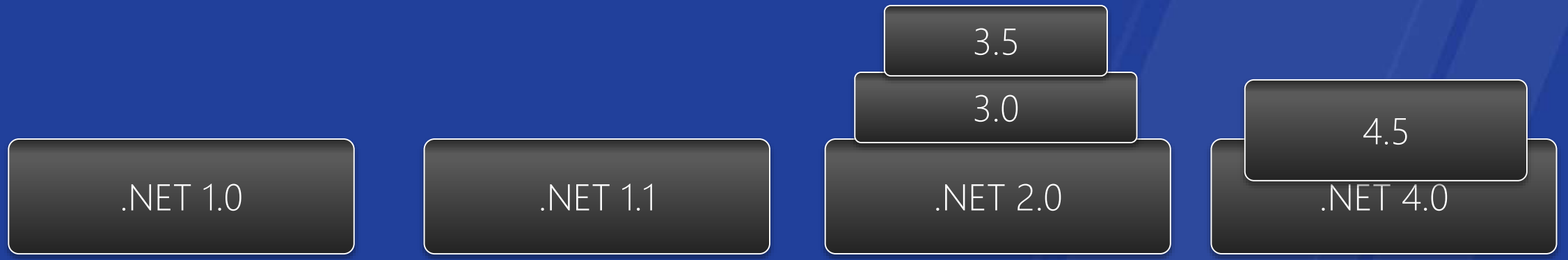
# What's new in .NET 4.5

# Quick Reminder – What Is .NET



# .NET Versions

- .NET 4.5 is an in-place update
  - You can chain it in for your installers on Win7, Win2K8, etc.
  - You get it automatically with Windows 8 and Windows Server 8
  - On Windows Update for supported downlevel OSes



# Compatibility of in-place releases

- Compat for in-place releases is challenging
  - Bug fixes
  - Perf improvements
  - Internals

- Ways we preserve compat
  - Compat lab
  - Review bug fixes
  - Review features
  - Rerun old tests unchanged
  - Tools to check for common problems

# Call to Action: Compat Testing

- Binary compat
  - Try your 4.0 app on 4.5
  - Try your 3.5 app on 4.0; if it works, try on 4.5
- Source compat
  - Same, but recompile
- Serialization compat
  - 3.5 or 4 client talking to your .NET 4.5 server
  - 4.5 client saving/opening files for your 3.5/4 client

- Try it now, and we'll fix the bugs

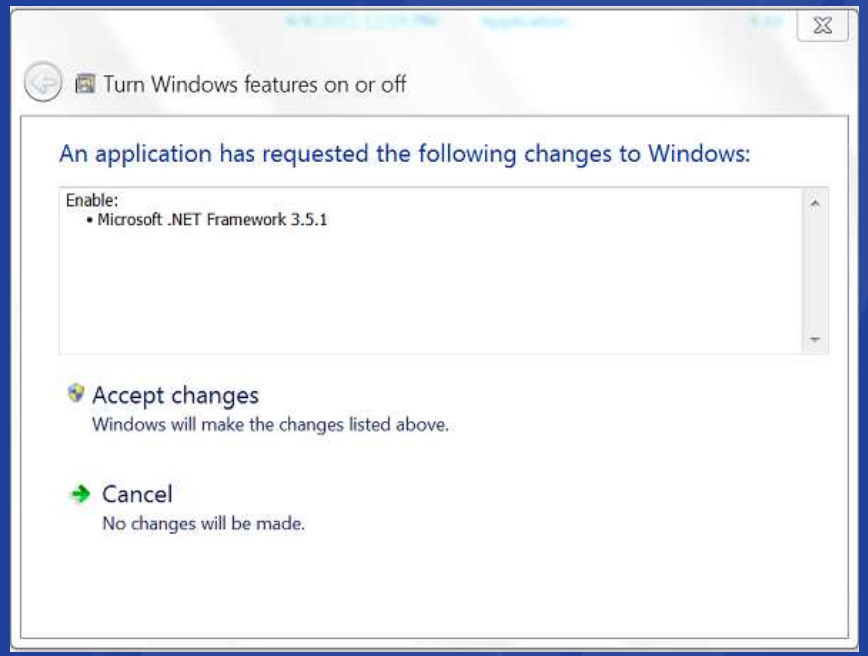
| | |
|---|---|
| **LEARN MORE** | http://blogs.msdn.com/b/clrteam/ "Compat Testing" |

# .NET 3.5 Feature-On-Demand

- Each .NET version on a user's machine has costs
  - Size of images
  - Servicing cost
- But want Win8 desktop highly compatible with Win7

- Solution: Feature-on-demand.
  - Option 1 (preferred)
    - Your setup tries to install 3.5 MSI
    - 3.5 MSI shim installs correct Win8 version
    - Typical for apps that support XP
  - Option 2:
    - App tries to use 3.5
    - App use detected, 3.5 is installed, restart app
  - Edge cases:
    - App lights up when 3.5 is installed

- OEMs and Enterprise can pre-install 3.5



# Windows 8 Support

In this section:

Windows Runtime
.NET for Metro Style Apps
Portable Libraries

# Support for Windows Runtime

```
using Windows.Storage;
using Windows.Storage.Pickers;

var picker = new FileOpenPicker();
picker.FileTypeFilter.Add("*");
var files
foreach (S
{
    listBo
}
```

Windows Runtime code feels natural for C# a

LEARN MORE

531- Using the Windows runtime from C# and Visual Basic

**Hands on Labs:**

699 Building Your First Metro Style Application Using C#
700 A Tour of Building Metro Style Applications Using C# Part 1
701 A Tour of Building Metro Style Applications Using C# Part 2
702 Playing Media in Metro Style Applications Built Using C#
703 Exploring Data Binding in Metro Style Applications Using C#
705 Using Windows 8 Controls in your Metro Style Applications Built Using C#
1000 Using Windows 8 Templates to Build Metro Style Applications Using C#

# .NET for Metro style apps

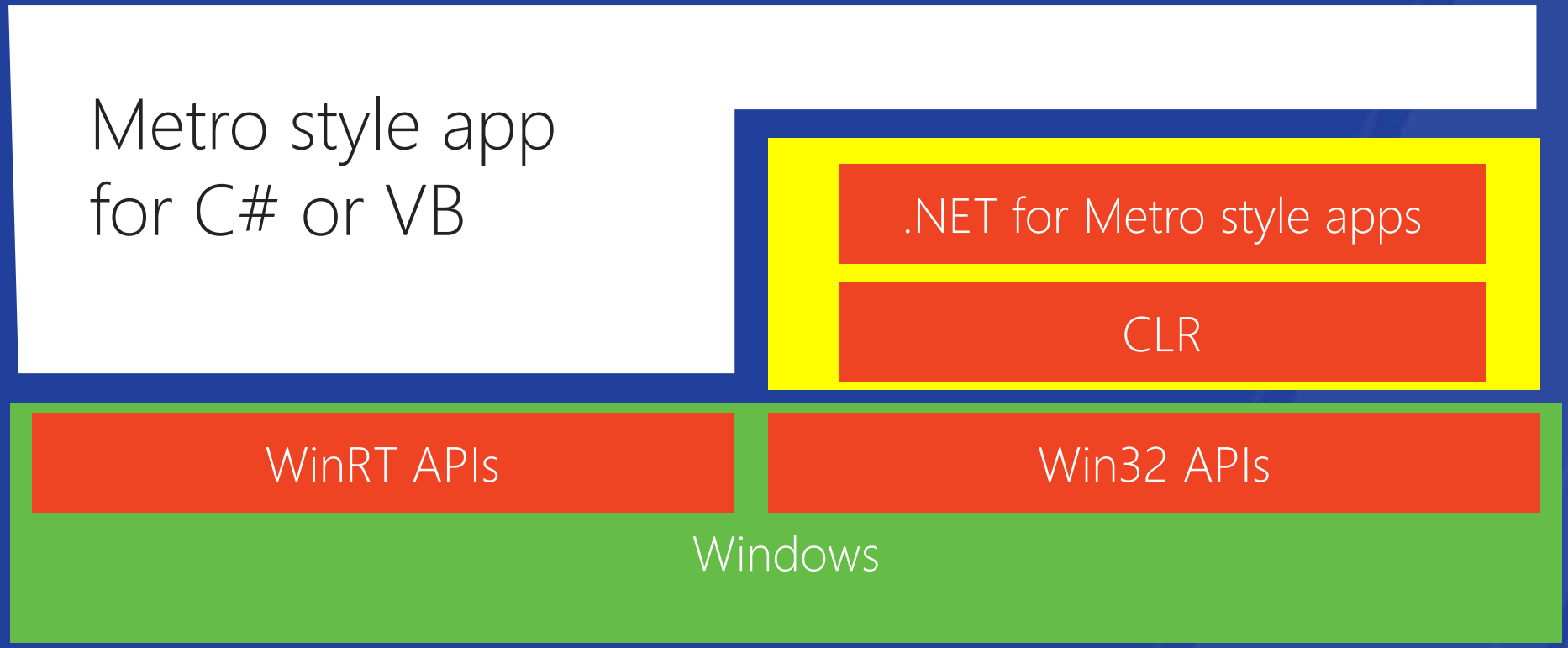
- *Provide simple, well-designed, and light-weight base class library APIs to C# and VB developers programing Metro style apps*

- API Cleanup
  - No obsolete or inapplicable APIs
  - No duplication with Windows Runtime APIs
  - Natural for C# & VB developers



| LEARN MORE | 930: A .NET Developer's View of Windows 8 App Development |
| --- | --- |

# .NET and WinRT




In this section:

Async programming
Task Parallel Library Improvements
Improved server Garbage Collector
Faster ASP.NET startup

# Await: More responsive applications

- Client UI Code
  - Easy to write client UI code that doesn't block
- Business logic
  - Easy to write code that fetches data from multiple sources in parallel
- Server code
  - Better scalability – no need to have a thread per request.
- New APIs in BCL, ASP.NET, ADO.NET, WCF, XML, WPF

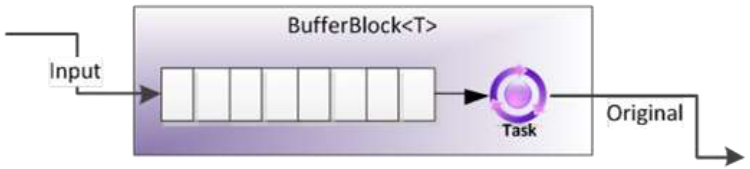
```
async void DisplayUserInfo(string userName) {
    var image = FetchUserPictureAsync(userName);
    var address = FetchUserAddressAsync(userName);
    var phone = FetchUserPhoneAsync(userName);
    await Task.WhenAll(image, address, phone);
    DisplayUser(image.Result, address.Result,
                phone.Result);
}
```

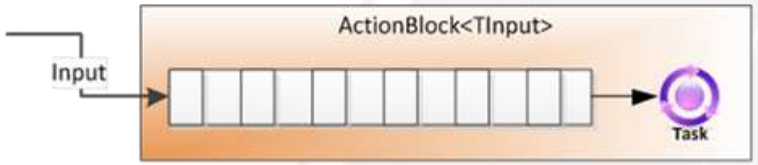
**LEARN A LOT MORE**

804: Building IIS and ASP.NET Apps with the Power of Async
810: C# and Visual Basic in VS 11: Windows 8 Async Made Simple
829: The Zen of Async: Best Practices for Best Performance

Hands on Lab:
995 Asynchronous Programming in the .NET Framework 4.5 with Visual C# and Visual Basic

# TPL Dataflow



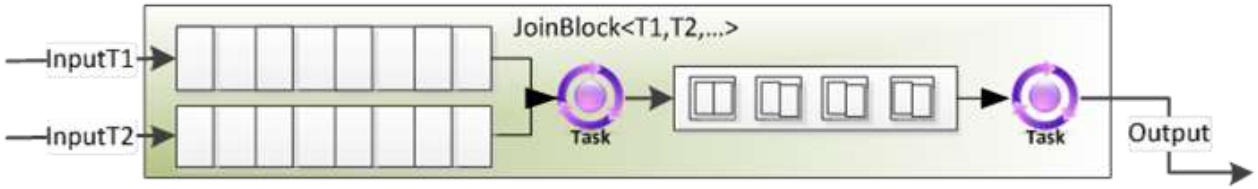



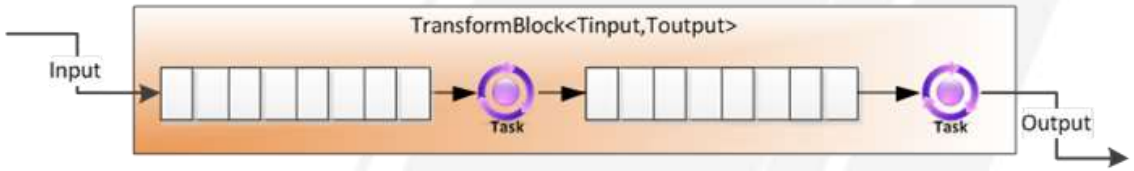

Complements .NET 4 parallel programming

.NET 4: “Data first.  Now set up computation.”

New in 4.5: “Set up computation. Now here’s data.”

Primitives for in-process message passing



# Other Parallel Computing Additions

- Combinators
  - Task.WhenAll, Task.WhenAny
- Timer integration
  - Task.Delay(TimeSpan), CancellationTokenSource.CancelAfter(TimeSpan)
- Task scheduling
  - ConcurrentExclusiveSchedulerPair
- Fine-grained control
  - DenyChildAttach, HideScheduler, LazyCancellation, EnumerablePartitionerOptions
- ThreadLocal<T>.Values
- PERFORMANCE (“it’s just faster”)

| LEARN MORE | 808: Building Parallelized Applications with .NET and Visual Studio |
| --- | --- |

# GC Improvements

- Background Server GC
  - Shorter pauses when doing Gen2 Collections
- (Server GC) **Scalable Marking** for full blocking GCs
- Large Object Heap Allocation Improvements
  - Better use of free space on LOH
  - (Server only) Balancing the LOH allocations across processors

| LEARN MORE | 813 Deep Dive into the Kernel of the .NET Framework |
| --- | --- |

# Faster ASP.NET Startup

- Two ways to run ASP.NET
  - Start app, keep it running
  - Start when a request comes in (e.g. Hosters)

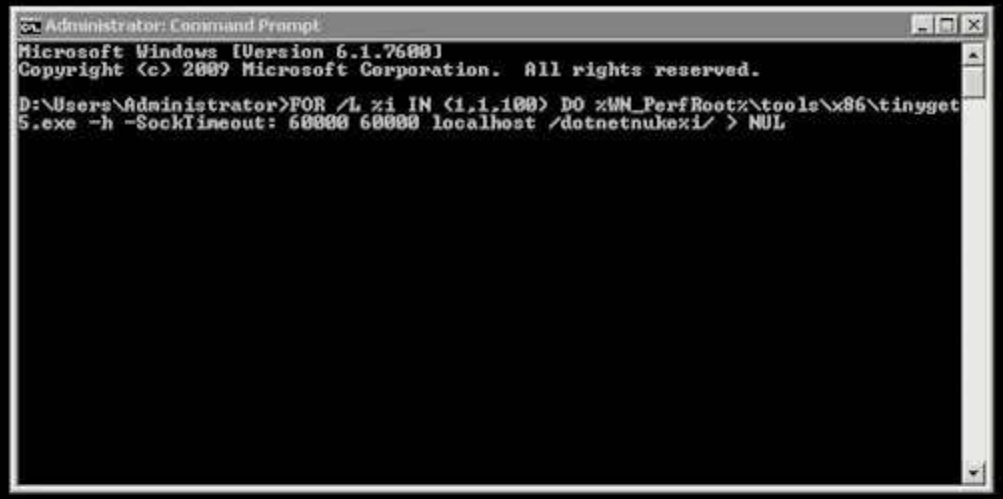
- 35% faster cold start
  - Multi-core JIT
  - Windows Server 8 pre-fetch option

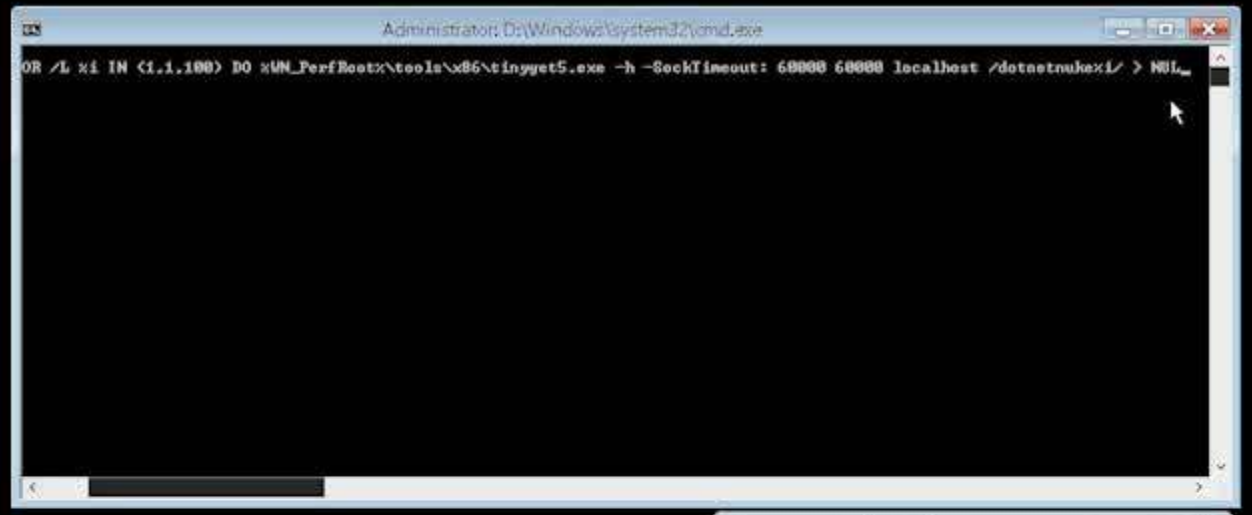
- Working set improvements

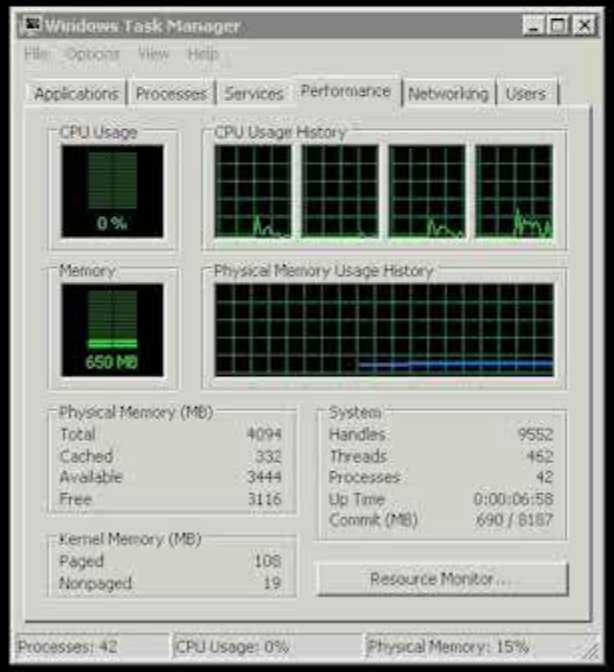


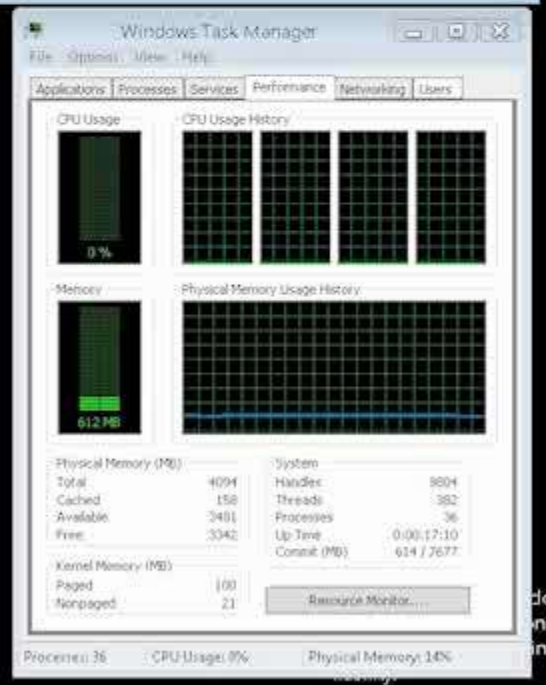
LEARN MORE

837: Optimize your Website Using ASP.NET and IIS8




Windows Server 2008R2 w/ Dev10

Windows Server 8 w/ Dev11

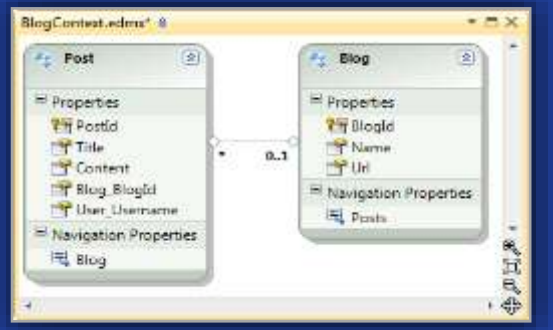
# Better Data Access

In this section:

Entity Framework
ADO.NET
WCF
Websockets

# Developer Workflows

## Designer Centric




## Code Centric

### New Database

**Model First**
- Create .edmx model in designer
- Generate database from .edmx
- Classes auto-generated from .edmx

**Code First**
- Define classes & mapping in code
- Database auto-created at runtime

### Existing Database

**Database First**
- Reverse engineer .edmx model
- Classes auto-generated from .edmx

**Code First**
- Define classes & mapping in code (Reverse engineer tools available)

# Pillars of EF 4.5

- Improve Developer Productivity
  - Enums
  - Migrations
  - Batch Sproc Import
  - Designer highlighting and coloring
- Enable SQL Server and Azure Features
  - Spatial (Geometry and Geography)
  - Table-valued functions
  - Sprocs with multiple result sets
- Increase Enterprise Readiness
  - Multiple diagrams per model
  - TPT query optimizations
  - Automatic compiled LINQ queries

EF 4.5
(.NET 4.5)

EF 4.1
(Magic Unicorn)

EF 4
(.NET 4.0)

EF 1
(.NET 3.5 SP1)

**LEARN MORE**

812: Creating immersive data experiences with Entity Framework
904: F# 3.0: data, services, Web, cloud, at your fingertips

# WCF

- Scalable modern communication stack
  - Interoperable UDP multi-cast channel
  - TCP support for high-density scenarios (partial trust)
  - Async
  - Improved streaming support

- Continued commitment to simplicity
  - Further config simplification, making WCF throttles/quotas smarter & work for you by default!
  - Better manageability via rich ETW & e2e tracing

| | |
|---|---|
| LEARN MORE | 800 – Building data-driven HTML5 apps with WCF RIA Services<br>798 - Building Web APIS in Windows Azure with WCF to reach any device |

# Websockets

- New bidirectional communications standard (W3C & IETF)
  - Low latency
  - Low bandwidth

- Client: IE10, .NET, Win 8 Metro style apps
- Server: Windows 8 Server – ASP.NET, WCF, IIS

- Example applications:
  - Stock tickers, chat applications, interactive games

| LEARN MORE | 807: Building Real-Time Web Apps with WebSockets using IIS, ASP.NET and WCF |
| --- | --- |

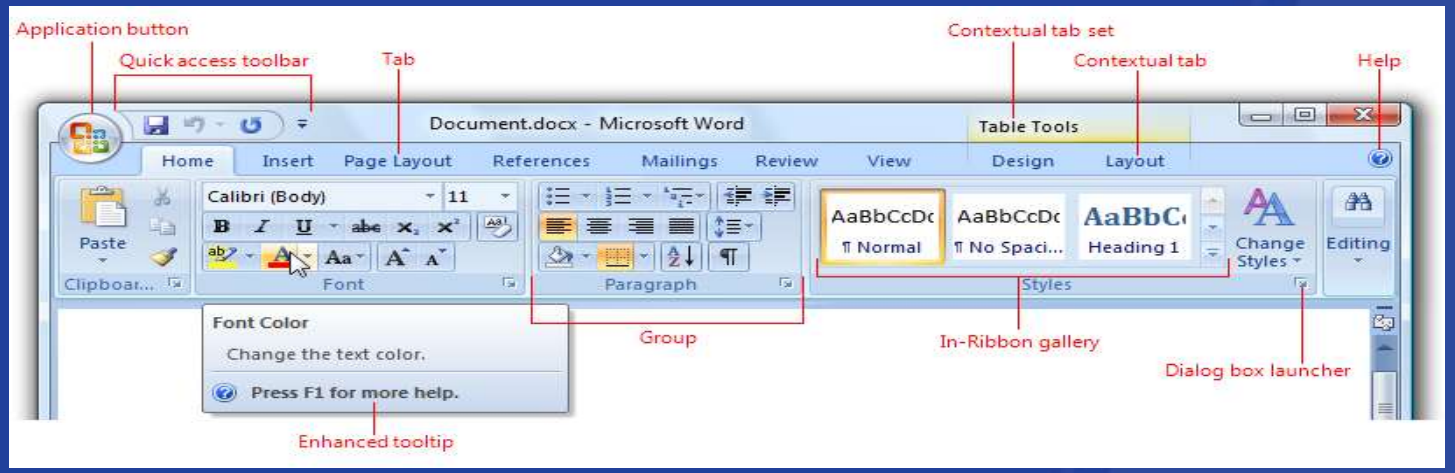
# Top Developer Requests

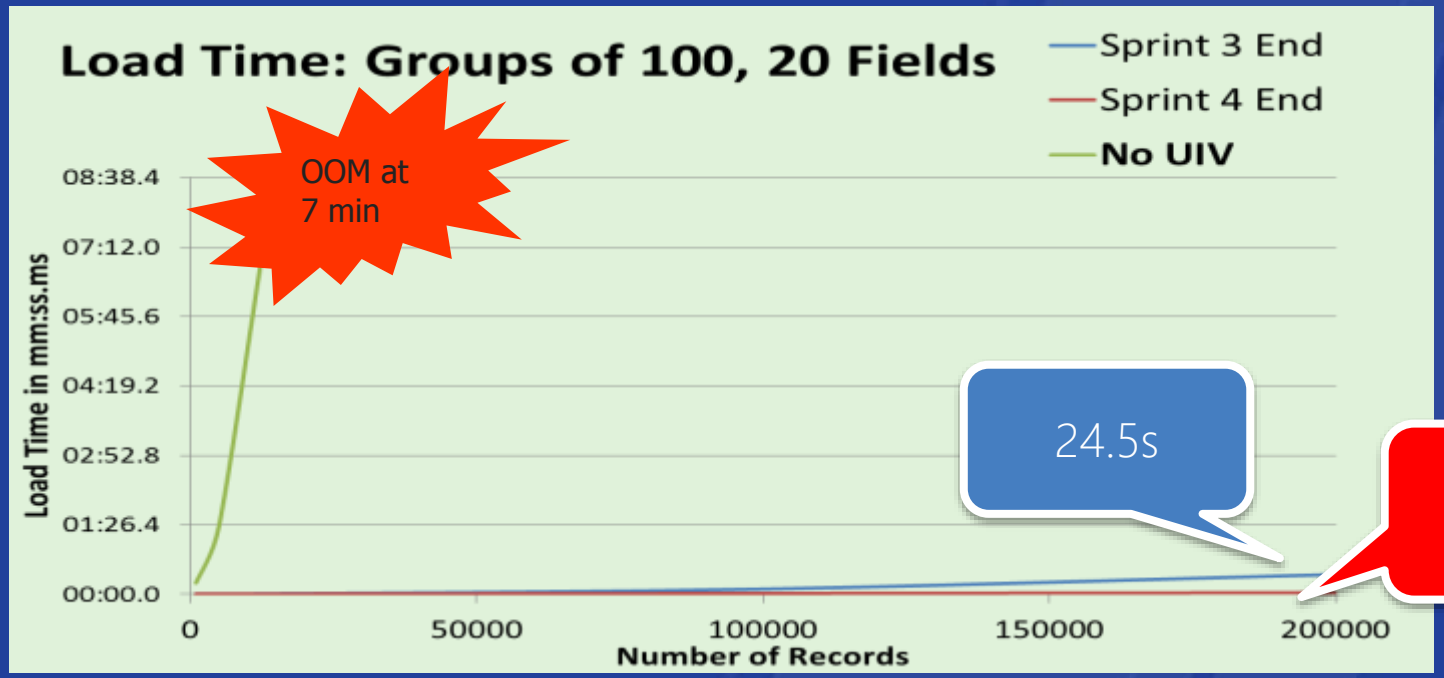
In this section:

WPF
Workflow
BCL
MEF
ASP.NET

# WPF

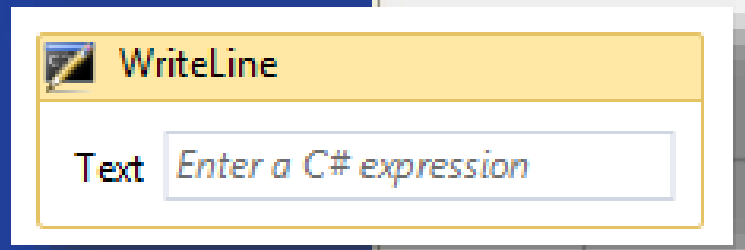
- Ribbon licensing
- Itemscontrol scaling
- MVVM Improvements



# Workflow

- C# Expressions
- State machines
- Versioning
- Contract First Services
- Runtime Performance
- Designer Improvements
  - Performance
  - Usability
  - Better Integration with VS







**LEARN MORE**

801: Event driven, long-running apps with Windows Workflow
867: Building Apps with Windows Workflow and Windows Azure

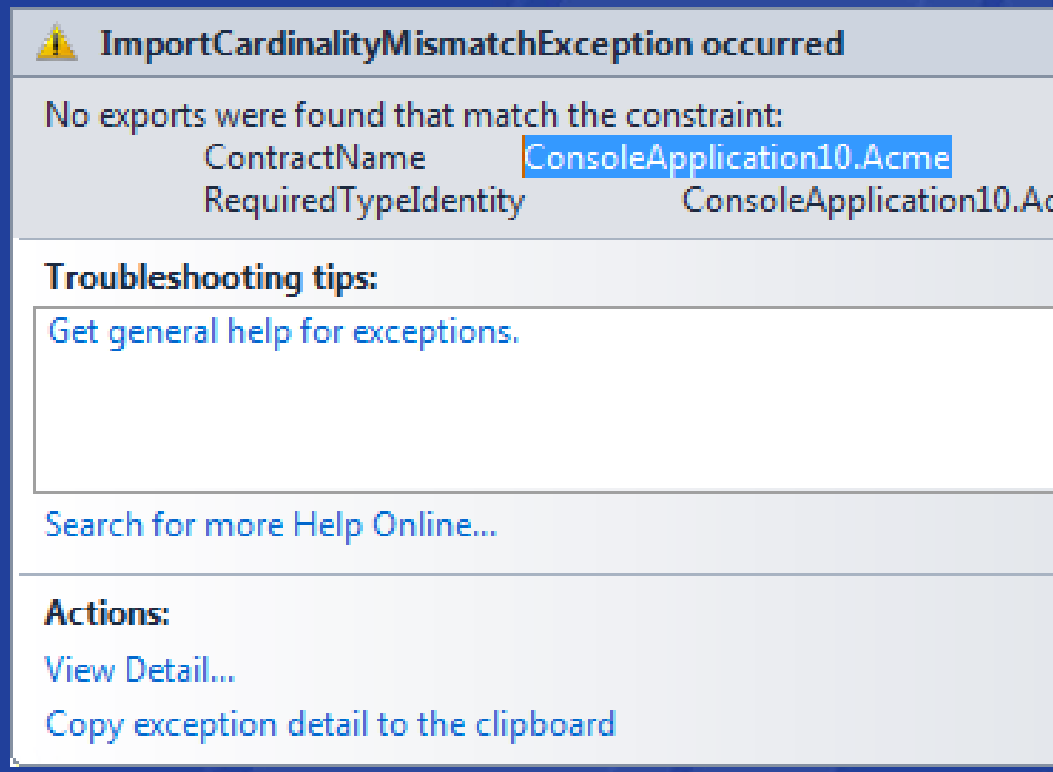
# Base Class Libraries

- Improvements to
  - WeakReferences
  - ArraySegment
  - Streams
  - ReadOnlyDictionary
  - Compression
  - Bigger than 2GB Objects

# New MEF Features

- All your objects are MEF now
  - Generics
  - POCO
  - Explicit Wiring (wire specific MEF parts the way YOU want)

- MEF problems are easy to diagnose
  - Break on First Chance Exceptions
  - Visualize the exception
  - Fix your problem!



# ASP.NET

- Cloud
  - ASP.NET providers natively support SQL Azure
- Better Web than ASP.NET 4
  - HTML 5 Templates and Support
  - Web Form Model Binders
  - Unobtrusive Javascript Validators
  - Anti-XSS Encoders

**LEARN MORE**

796 - ASP.NET 4.5 loves HTML5, CSS3 & JavaScript
906 - Create rich, data-driven Web apps with ASP.NET 4.5 Web Forms
797: It's not a great phone app without ASP.NET services and push notifications

# Review

.NET 4.5 is an in-place update
Help us make sure it is highly compatible.

# .NET 4.5 makes it easy and natural to write Metro style apps using C# and VB

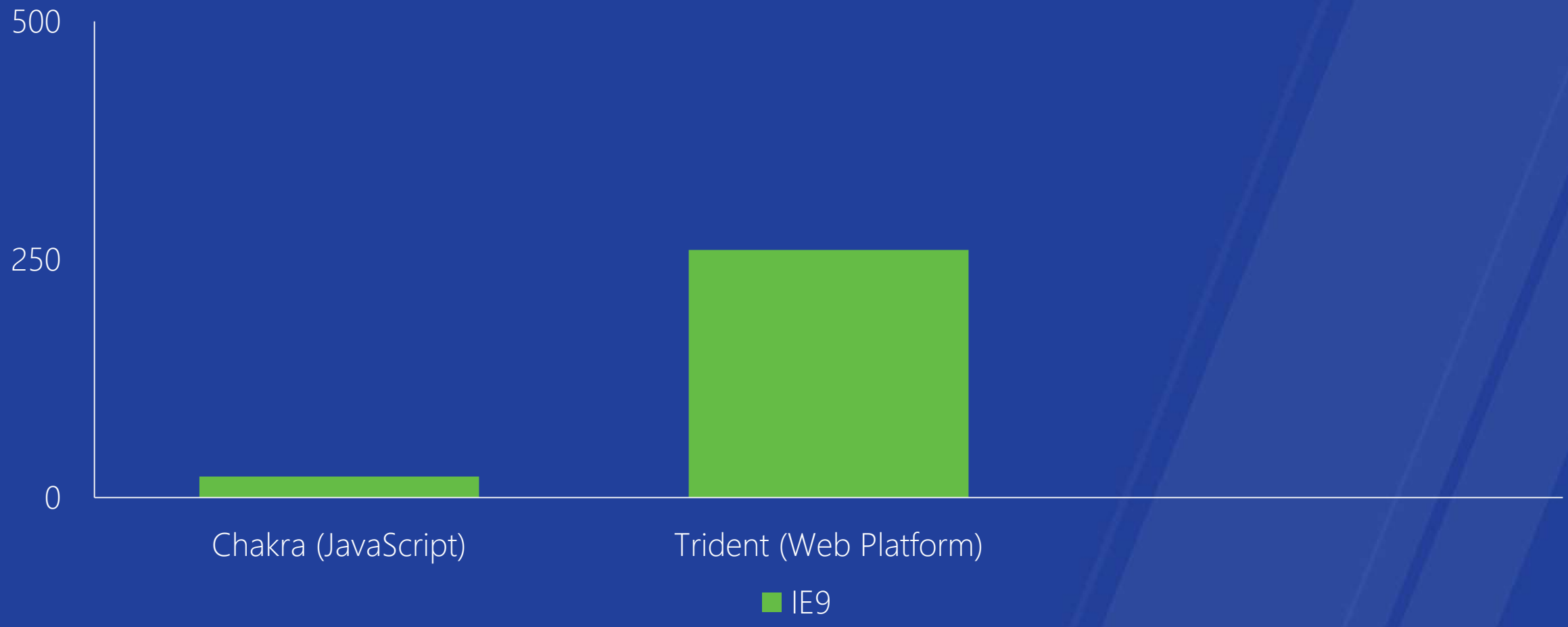
# .NET 4.5 makes your apps run faster: Faster ASP.NET startup, fewer pauses due to Server GCs, and great support for Asynchronous programming

.NET 4.5 gives you easy, modern access to your data, with support for Entity Framework Code First, and recent SQL Server features, and Websockets

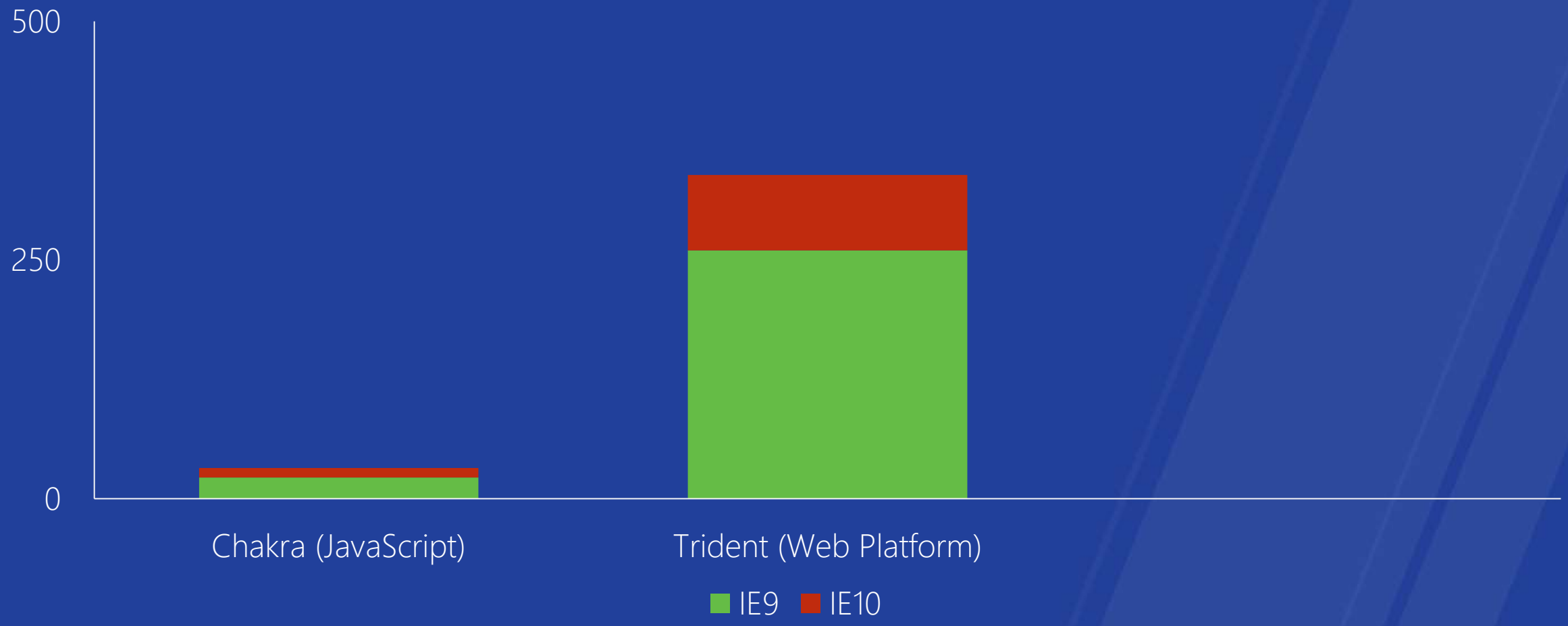
# .NET 4.5 addresses the top developer requests in WPF, Workflow, BCL, MEF, and ASP.NET

# Using the Windows Runtime from JavaScript

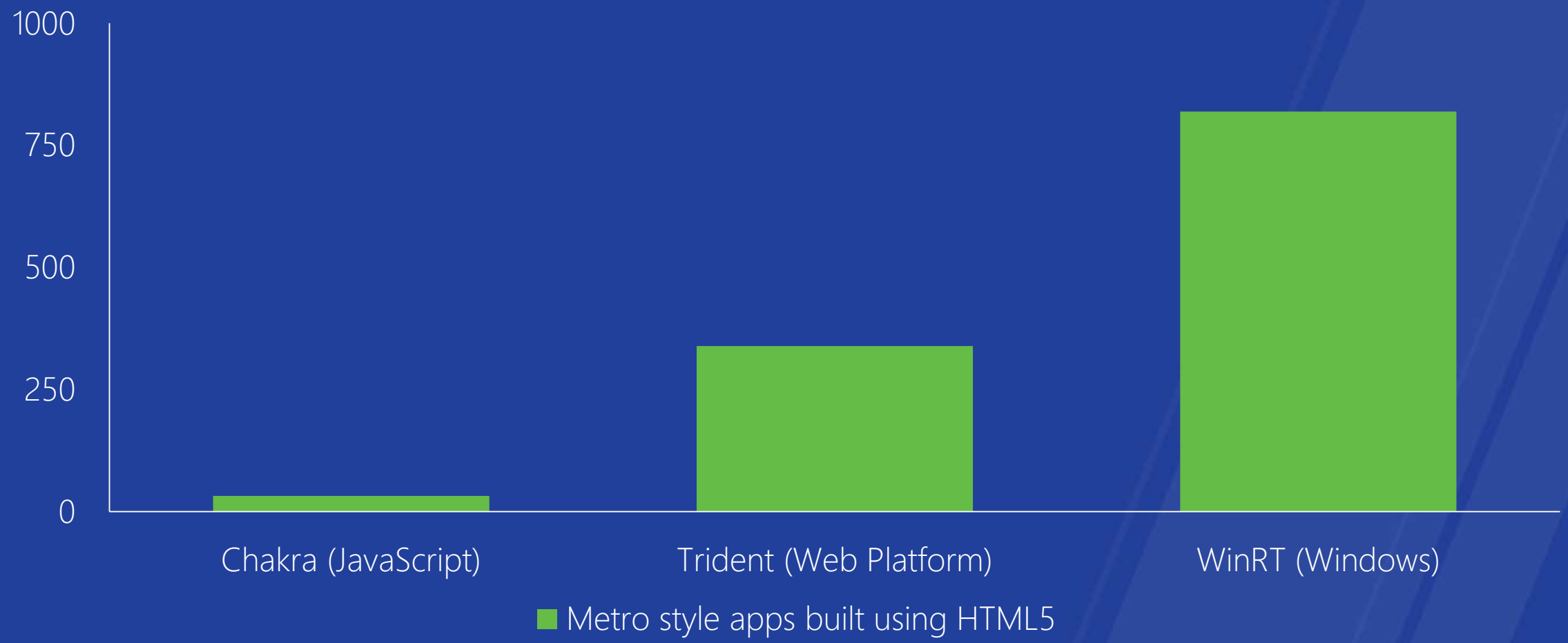
# APIs available in JavaScript Metro style apps



# APIs available in JavaScript Metro style apps



# APIs available in JavaScript Metro style apps



The Windows Runtime provides a rich and expansive set of APIs for JavaScript developers.

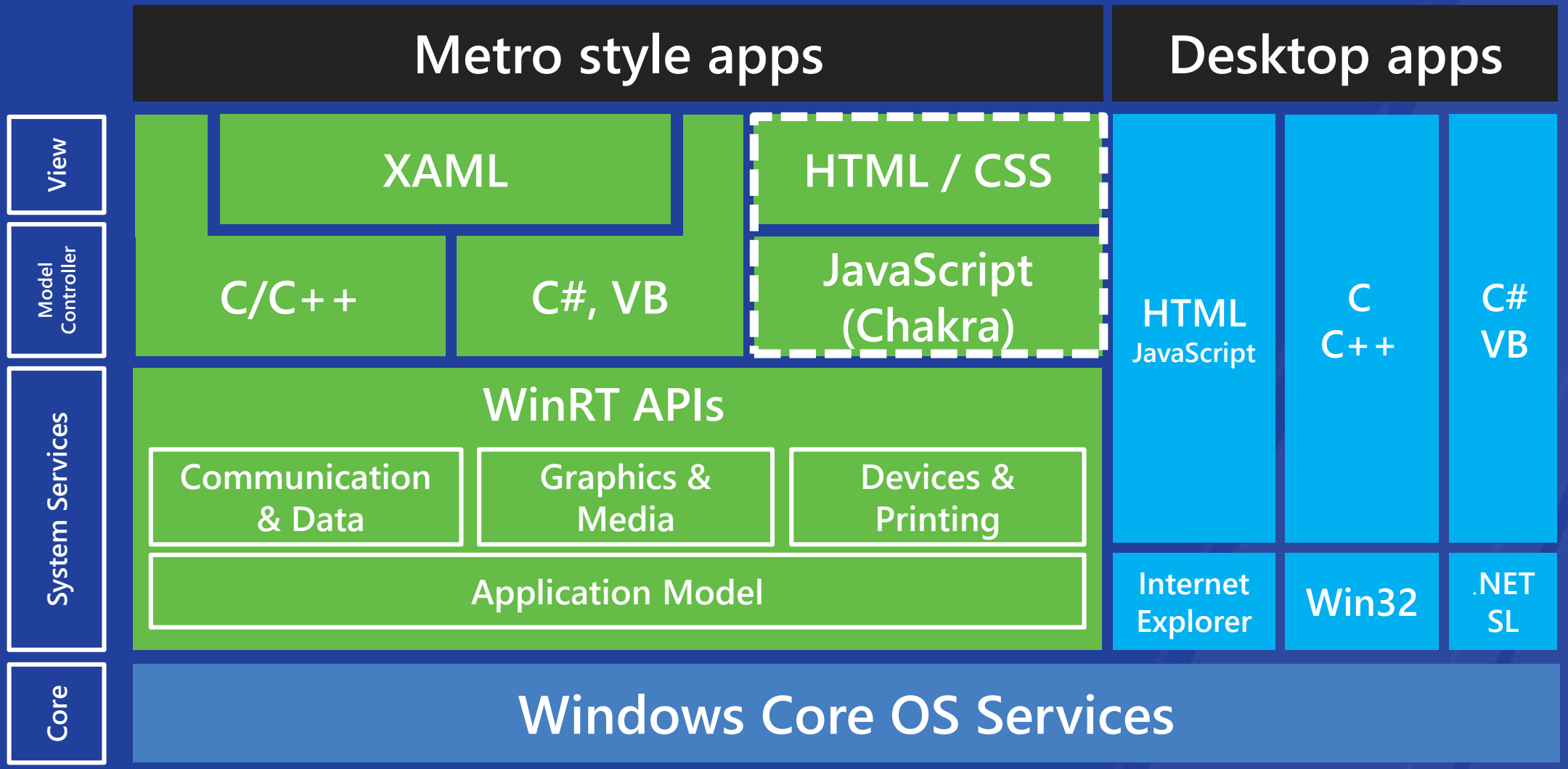
# demo

## A tour of WinRT from JavaScript

Using Visual Studio to explore and experiment with Windows Runtime APIs

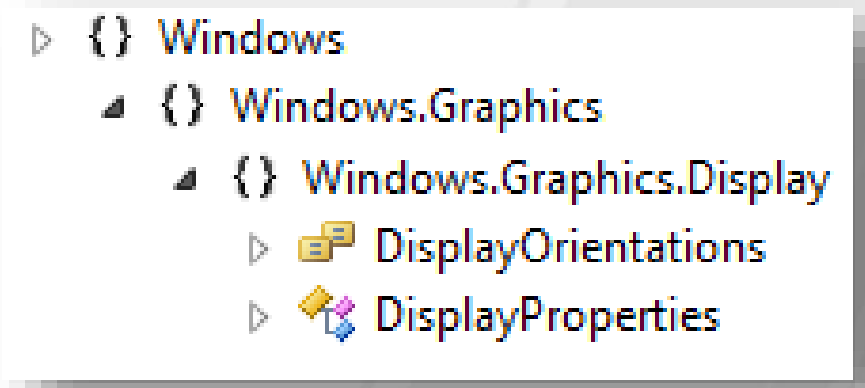
# WinRT and JavaScript Basics

# Windows 8


Metro style apps
Desktop apps
View
Model Controller
System Services
Core
XAML
HTML / CSS
C/C++
C#, VB
JavaScript (Chakra)
WinRT APIs
Communication & Data
Graphics & Media
Devices & Printing
Application Model
HTML JavaScript
C C++
C# VB
Internet Explorer
Win32
.NET SL
Windows Core OS Services

The Windows Runtime provides platform neutral metadata describing APIs – namespaces, classes, enums, methods, events and more…

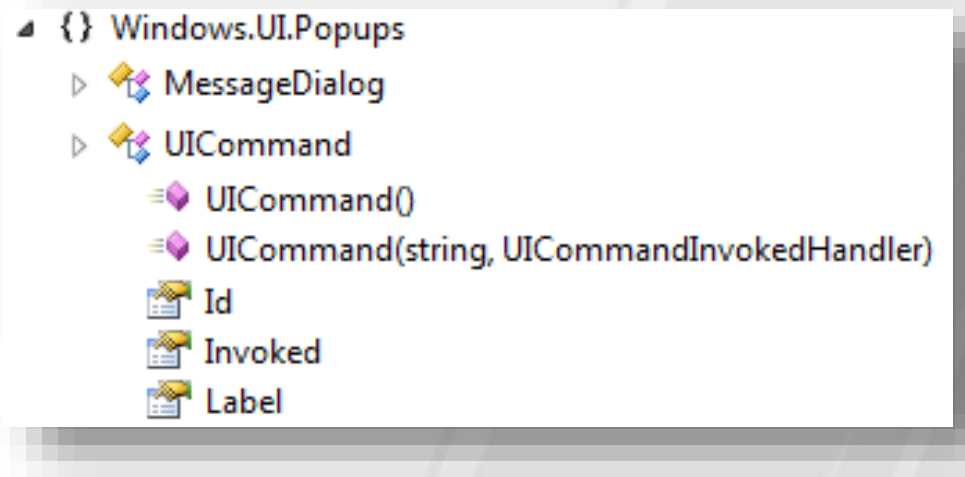
# Namespaces




```
// Namespaces are objects
var win = Windows;
var graphics = Windows.Graphics;
var display = Windows.Graphics.Display;

// Namespaces contain namespaces, classes and enums
var displayProperties = display.DisplayProperties;
var orientations = display.DisplayOrientations;
```

# Classes



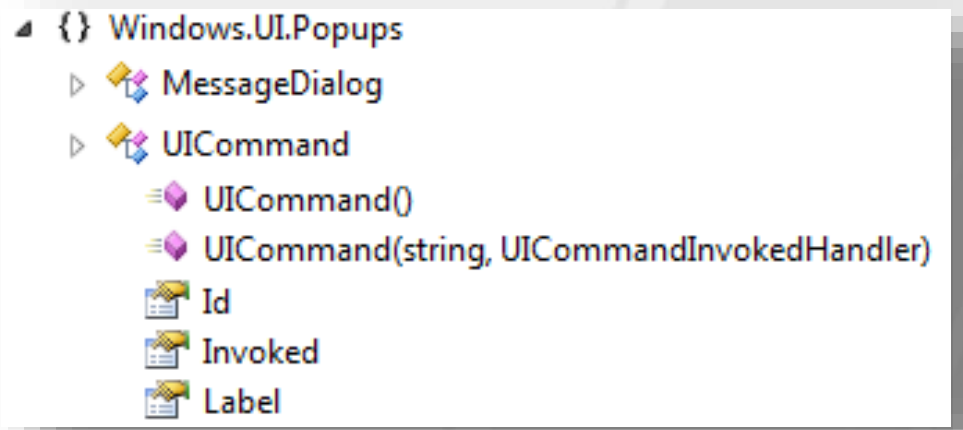

```
var Popups = Windows.UI.Popups,
    UICommand = Popups.UICommand,
    MessageDialog = Popups.MessageDialog;

var dialog = new MessageDialog("What next?");

var customCommands = [
  new UICommand("Say hi", function () { console.log("hello"); }),
  new UICommand("Say bye", function () { console.log("goodbye"); })
];

dialog.defaultCommandIndex = 0;
dialog.commands.replaceAll(customCommands);
dialog.showAsync().then();
```

# Members and parameters



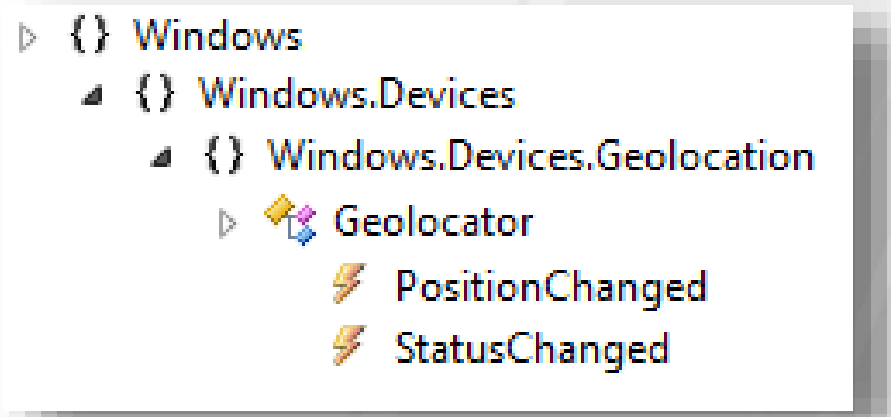

```
var Popups = Windows.UI.Popups,
    UICommand = Popups.UICommand,
    MessageDialog = Popups.MessageDialog;

var dialog = new MessageDialog("What next?");

var customCommands = [
  new UICommand("Say hi", function () { console.log("hello"); }),
  new UICommand("Say bye", function () { console.log("goodbye"); })
];

dialog.defaultCommandIndex = 0;
dialog.commands.replaceAll(customCommands);
dialog.showAsync().then();
```

# Events




```
var geo = Windows.Devices.Geolocation;
var locator = new geo.Geolocator();

// on<eventname> for simple cases
locator.onpositionchanged = function (ev) {
  console.log("longitude = " + ev.position.coordinate.longitude);
  console.log("latitide = " + ev.position.coordinate.latitude);
}

// addEventListener/removeEventListener for multicast, removing
locator.addEventListener("statuschanged", onStatusChanged);

function onStatusChanged(ev) {
  console.log(ev.target + "--" + ev.status);
  locator.removeEventListener("statuschanged", onStatusChanged);
}
```

# demo

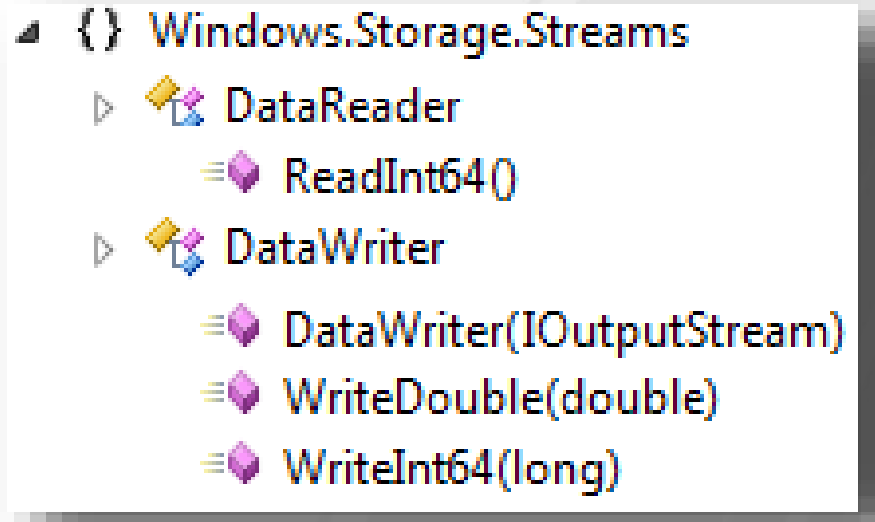
## Extending an app with WinRT – Part 1

Using APIs from the Windows Runtime to enhance the functionality of a Metro style app written in JavaScript

# WinRT and JavaScript **Details**

Many concepts used in Windows Runtime APIs are similar to DOM APIs, but there are also some new patterns.

# Basic types

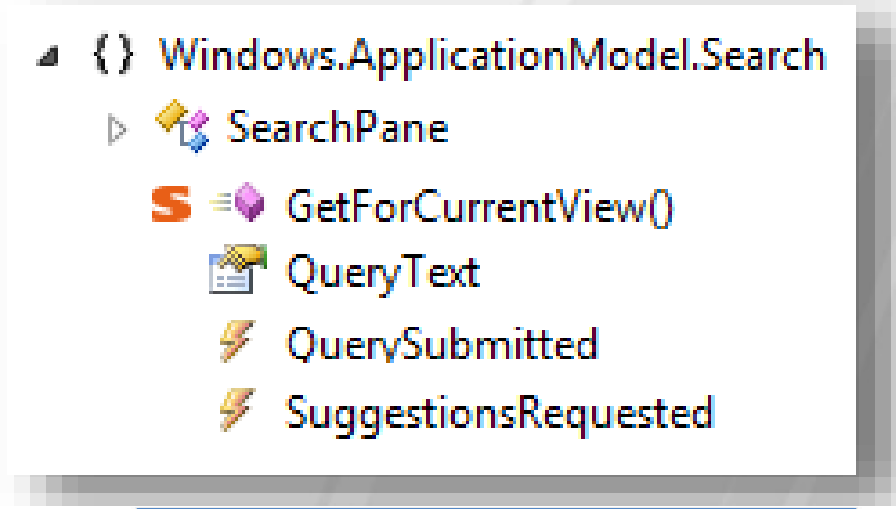



```
var Sockets = Windows.Networking.Sockets;
var socket = new Sockets.DatagramSocket();
socket.onmessagereceived = function(ev) {
  var dataReader = ev.getDataReader();
  var writer = new Windows.Storage.Streams.DataWriter(
                                        socket.outputStream);

  writer.writeDouble(3.14);
  writer.writeDouble("3.14");
  writer.writeDouble(null);
  writer.writeDouble({valueOf: function() { return 24; }});

  var intVal = dataReader.readInt64();
  writer.writeInt64(intVal);
};
```

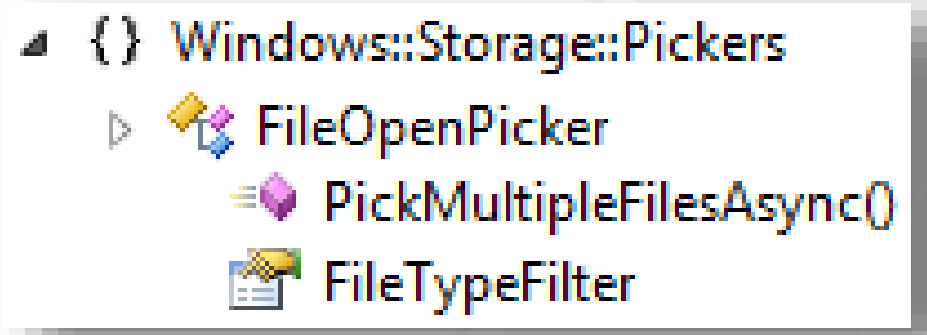
# Static members




```
var Launcher = Windows.System.Launcher,
    Uri = Windows.Foundation.Uri;

var searchPane =
      Windows.ApplicationModel.Search.SearchPane.getForCurrentView();
searchPane.onsuggestionsrequested = function (e) {
  var results = ["Windows", "Office", "XBox", "Visual Studio"];
  e.request.searchSuggestionCollection
           .appendQuerySuggestions(results);
}
searchPane.onquerysubmitted = function (e) {
  var uri = new Uri("http://www.bing.com//search?q="+e.queryText);
  Launcher.launchDefaultProgram(uri);
}
```

# Collections




```
var picker = Windows.Storage.Pickers.FileOpenPicker();

picker.fileTypeFilter.replaceAll([".jpg", ".png"]);

picker.pickMultipleFilesAsync().then(function (files) {
  div.innerHTML += "# of files:" + files.length + "<br/>";
  div.innerHTML += "1st file:" + files[0].fileName + "<br/>";
  files.forEach(function (file) {
    div.innerHTML+= file.fileName + "<br/>";
  });
});
```

# Async - Promises

```
promise.then(completeHandler,
             errorHandler,
             progressHandler)
```

# Async- basic

```
var picker = Windows.Storage.Pickers.FileOpenPicker();

picker.fileTypeFilter.replaceAll([".jpg", ".png"]);

picker.pickMultipleFilesAsync().then(function (files) {
  div.innerHTML += "# of files:" + files.length + "<br/>";
  files.forEach(function (file) {
    div.innerHTML+= file.fileName + "<br/>";
  });
});
```

# WinRT components and JavaScript

Windows Runtime Components written in C# and C++ can add custom native code to your JavaScript Metro style apps.

# JavaScript and C# WinRT component

```
// C#
namespace CustomWinRTComponent {
    public interface IMoreMath { }
    public sealed class MoreMath : IMoreMath {
        public static double Sinh(double x) {
            return Math.Sinh(x);
        }
    }
}

// JavaScript
var MoreMath = CustomWinRTComponent.MoreMath;
var y = MoreMath.sinh(0.7);
```

# JavaScript and C++ WinRT component

```
// C++
namespace CustomWinRTComponent {
    public ref class Math sealed {
    public:
        static double Sinh(double d) { return sinh(d); }
    };
}
```

```
// JavaScript
var MoreMath = CustomWinRTComponent.MoreMath;
var y = MoreMath.sinh(0.7);
```

# demo

## Custom WinRT components in JavaScript

Authoring and using custom native code components
from a Metro style app written in JavaScript

Microsoft®


//build/
windows