

# Platform for Metro Style Apps

Paul van Wingerden  
[paulvanw@microsoft.com](mailto:paulvanw@microsoft.com)  
Technical Evangelist  
Microsoft

# Windows reimagined



New fast and  
fluid Start  
screen

Touch,  
mouse,  
keyboard

Everything great about  
Windows 7 we made even  
better

# Great experience across all hardware

Tablets to  
laptops to all-  
in-ones



Touch,  
mouse,  
keyboard

Highest  
power to the  
most efficient  
chipset





# demo

Windows 8 reimagined

# Windows 8 Experience

- Fast and fluid user experience

Responsive, alive, beautiful

- Immersive and full-screen

Focuses attention on your apps

- Touch-first with full keyboard and mouse

Enables your choice of interaction

- Web of apps working together

Apps are connected to each other and the cloud

- Experience for all PC devices and architectures

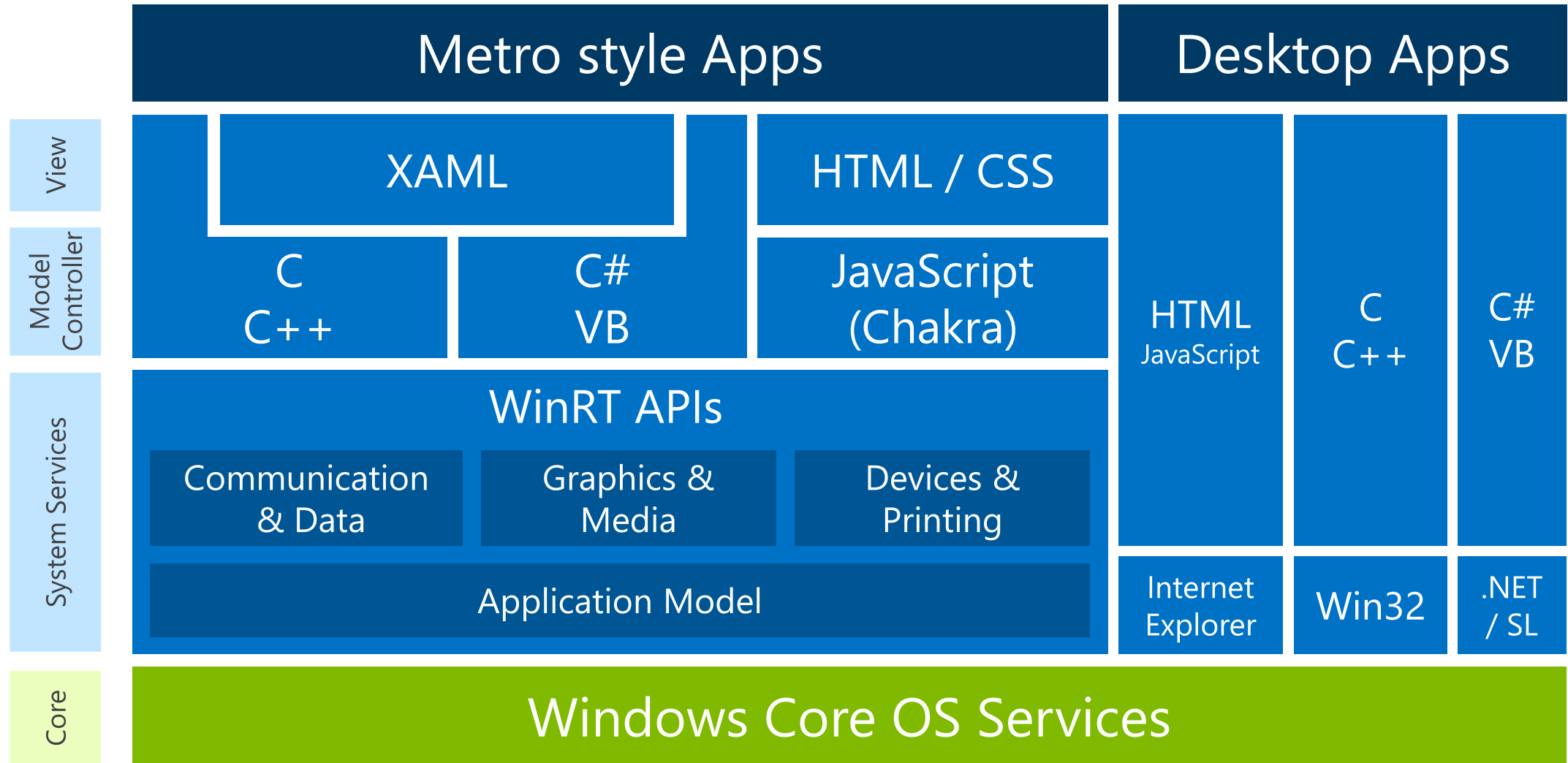
- No compromise across new form factors, desktops and laptops

# Platform Investments

# Key Investments

- ▶ New Windows Runtime (WinRT)
- ▶ New Windows 8 User Experience
- ▶ New App Execution Environment
- ▶ Windows Store

# Windows 8 Platform





# Windows Run-time (WinRT)

The background is a solid orange color with several faint, white-outlined geometric shapes scattered across it. These shapes include circles of various sizes and teardrop-like shapes, some of which are partially cut off by the edges of the frame.

# Hello WinRT in C++

## demo

# What did we see...

- WinRT APIs are native
- XAML used from C++
- Portable C++ code

# Metro style application APIs

## User Interface

HTML5/CSS

XAML

DirectX

Controls

Data Binding

SVG

Tiles

Input

Accessibility

Printing

## Devices

Geolocation

Portable

Sensors

NFC

## Media

Playback

Capture

PlayTo

Visual Effects

## Communications & Data

Contracts

Local & Cloud Storage

Web

Notifications

Streams

Memory  
Management

XML

Networking

SMS

## Fundamentals

Application Services

Threading/Timers

Memory Management

Authentication

Cryptography

Globalization



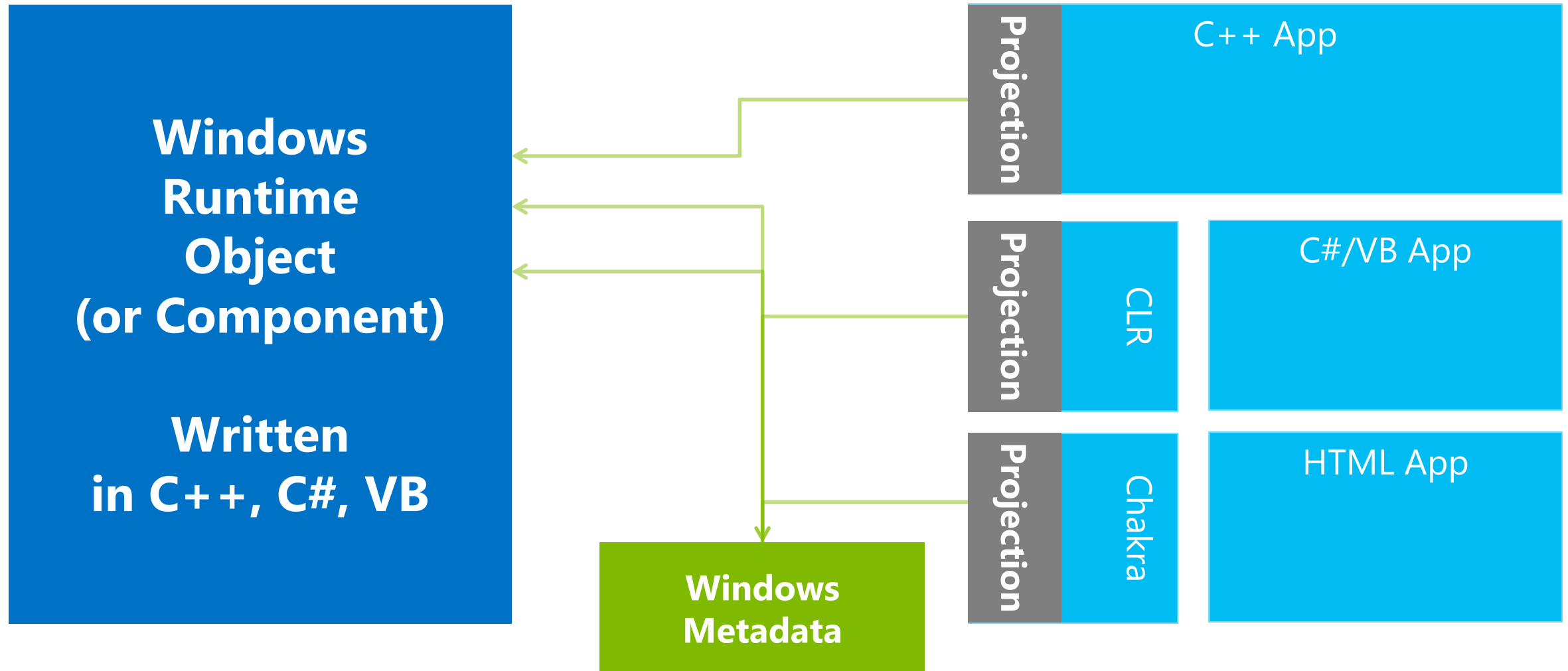
# demo

WinRT in C#

# What did we see...

- WinRT APIs are asynchronous
- WinRT APIs provide rich access to device, OS and services
- WinRT APIs feel natural in C#, Javascript and C++

# Language projections



# User Experience



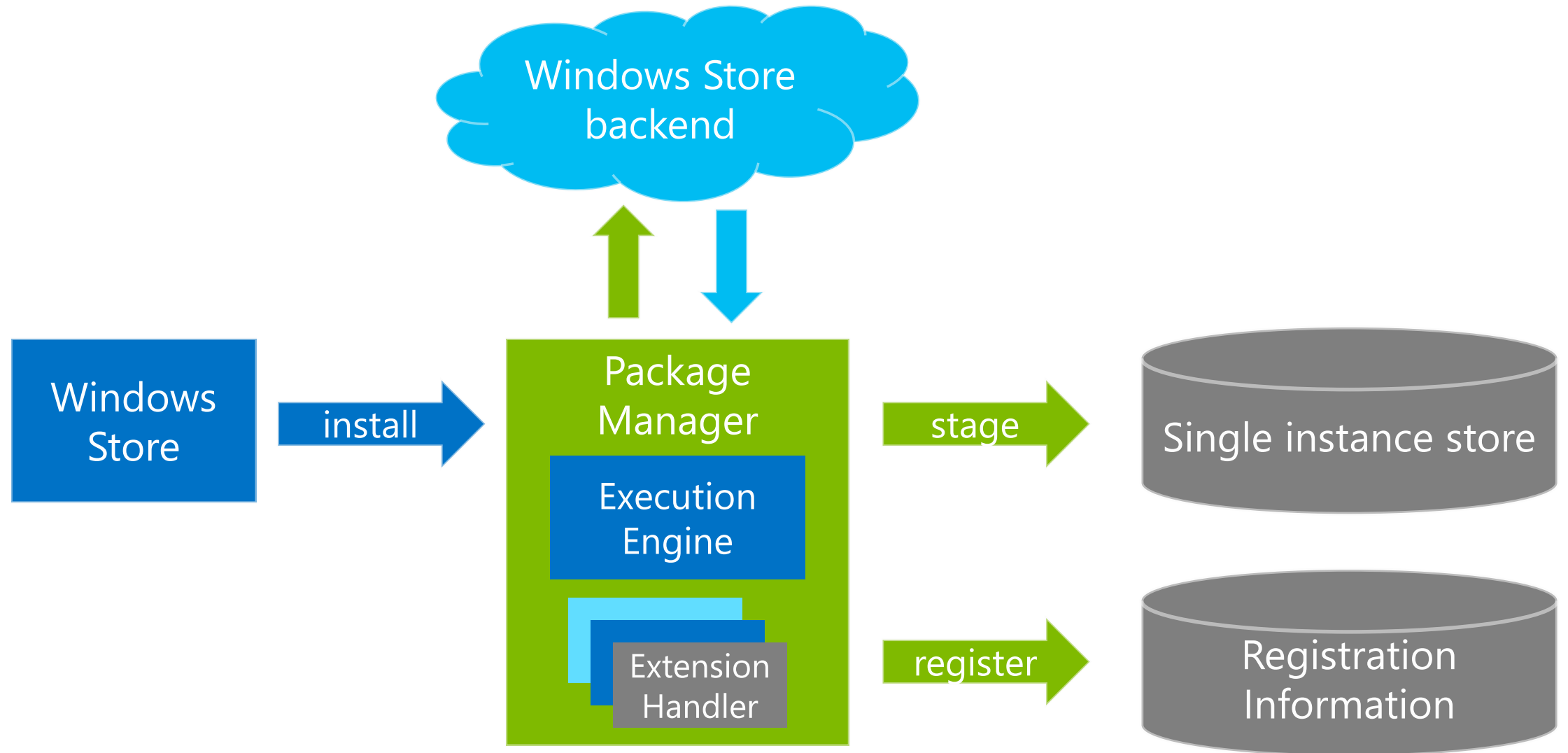
# Metro style apps

(more later)

- ▶ Leverage Metro style design
- ▶ Be Fast and fluid
- ▶ Snap and scale beautifully
- ▶ Use the right contracts
- ▶ Invest in a great Tile
- ▶ Feel connected and alive
- ▶ Roam to the cloud
- ▶ Embrace Metro principles

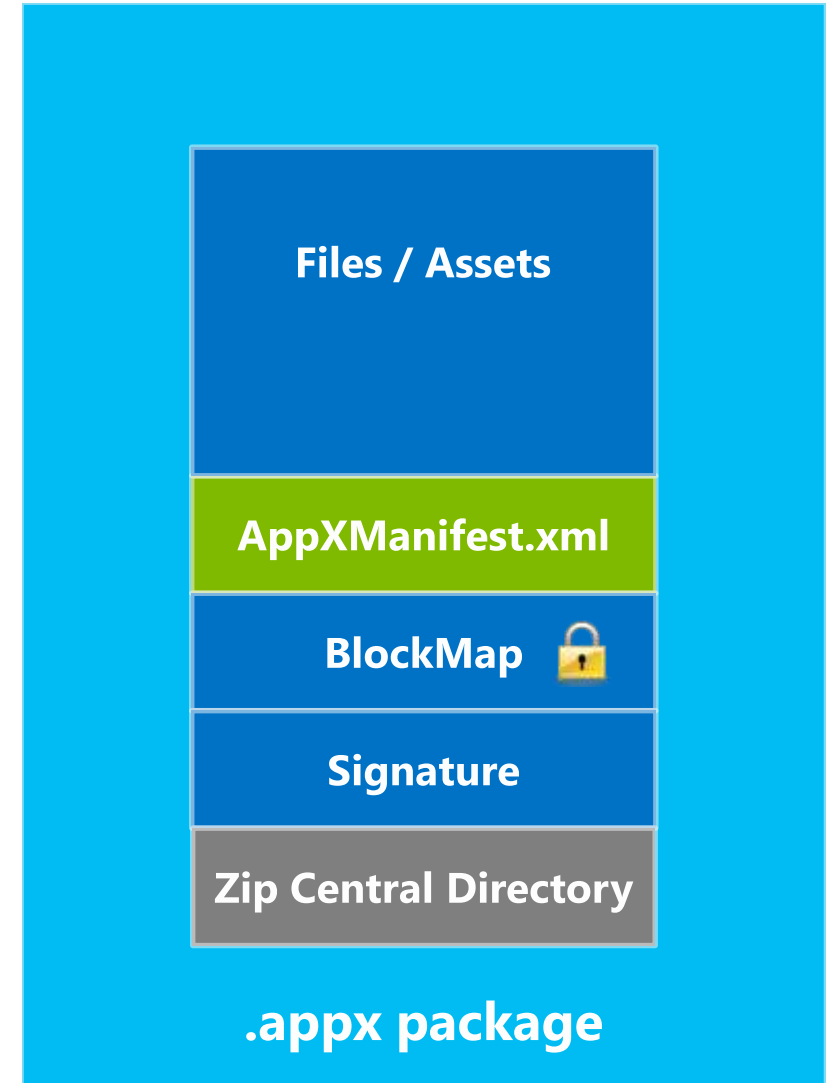
# App Execution Environment

# Metro style app acquisition



# Building blocks of an app package

- **App Manifest** contains all information needed to deploy the app
- **Blockmap** contains hashes of all files within the app package
- **Signature** validates the integrity of the app package



# App manifest and capabilities

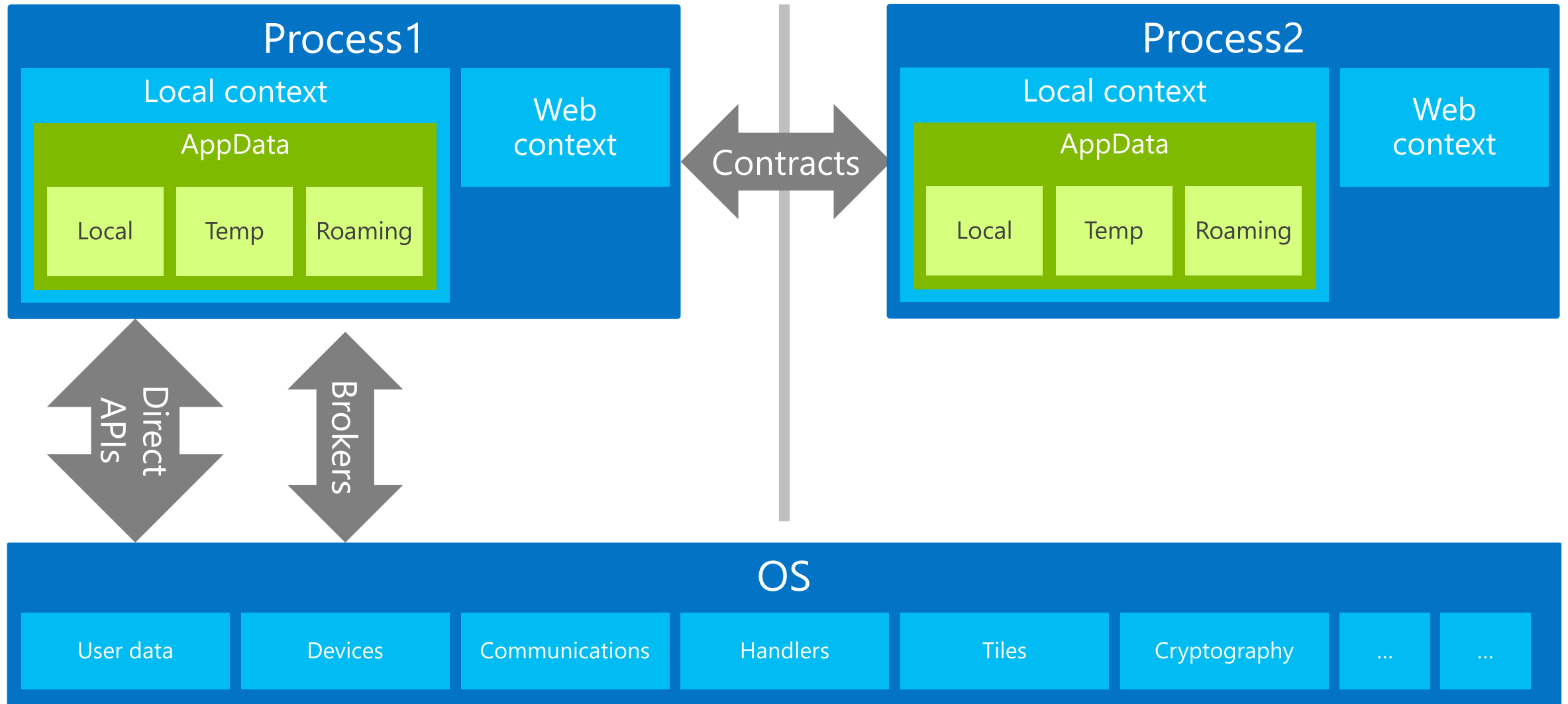
- Explicitly declares the app integration endpoints
  - File capabilities (music/pictures/videos/documents libraries, removable storage)
  - Devices (webcam, microphone, location, sms, proximity); extensible to new device classes
  - Network and identity (internet, private network, credentials)
  - File type associations
  - App contracts (search, share, etc.)
- Contained in appxmanifest.xml; Visual Studio provides editor



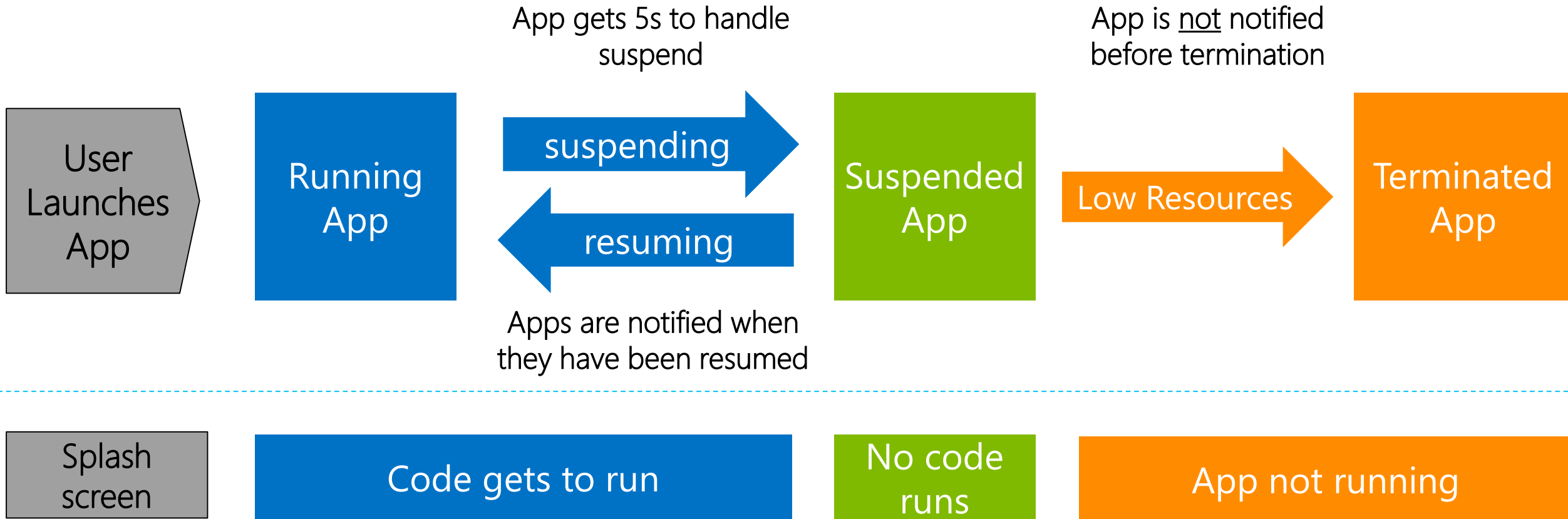
# Capabilities

# demo

# Process Isolation



# Process state transitions





# App data vs. User data

## App data

Settings

Files

Session State

Local Cache

## User data

Documents

Photos

Music

Videos

# Roaming

App uses data normally



Data syncs  
for the user's  
Microsoft account

# Recap: App execution

Confident	Connected	Powerful
Capabilities driven High isolation	Seamless data access Data Roaming	Performant Native platform

The background is a solid green color with several white circles of varying sizes scattered across it. Some circles are complete, while others are partially cut off by the edges of the frame.

# Windows Store

# The Windows Store

(more later)

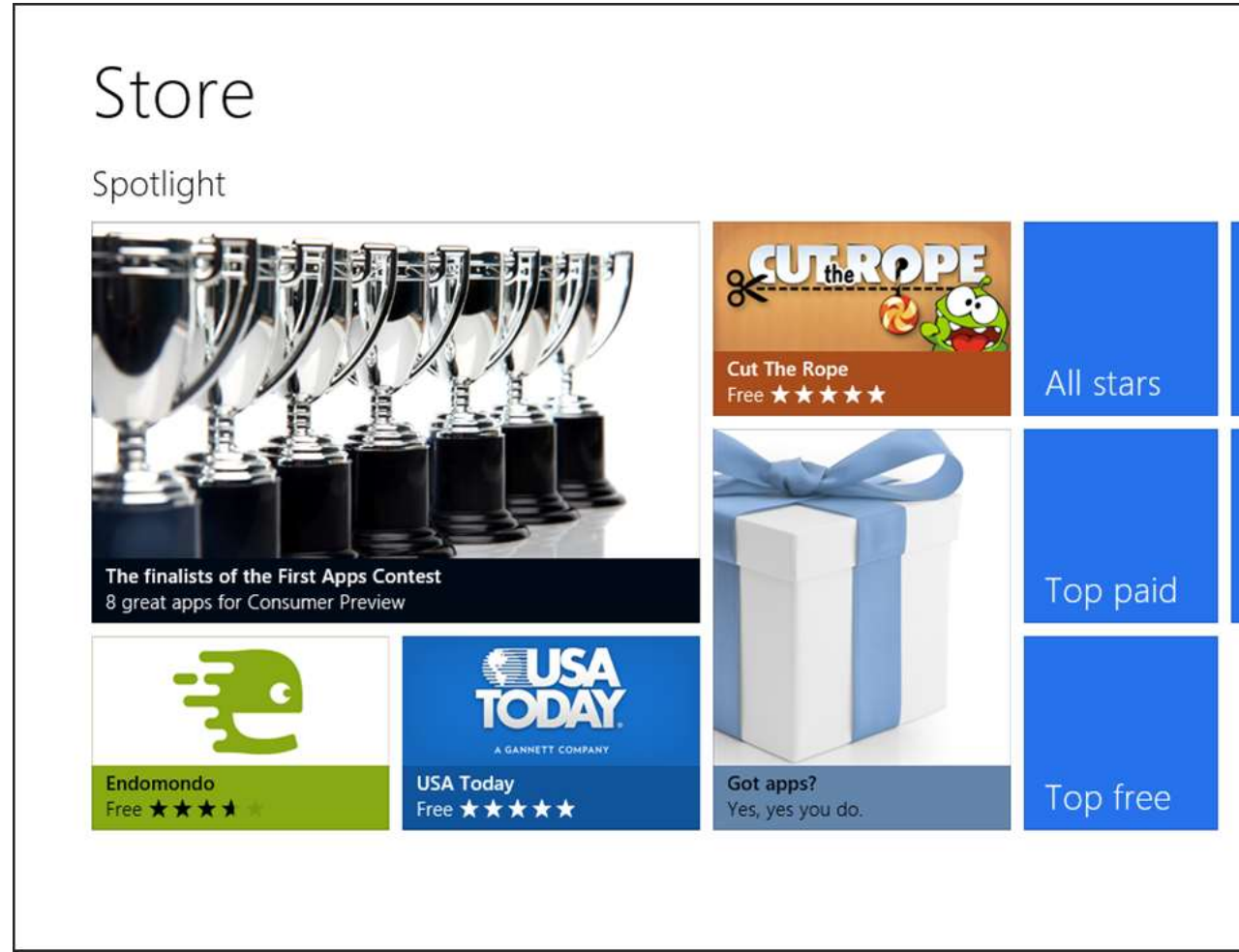
Unprecedented Reach

Optimized for Discoverability

Transparent Approval Process

Flexible Business Models

Maximizing Developer Revenue



The background is a solid green color with several white circles of varying sizes scattered across it. Some circles are complete, while others are partially cut off by the edges of the frame. The circles are distributed in the top-left, top-right, and bottom-right areas, leaving the bottom-left area relatively clear.

# Recap

# Windows reimagined

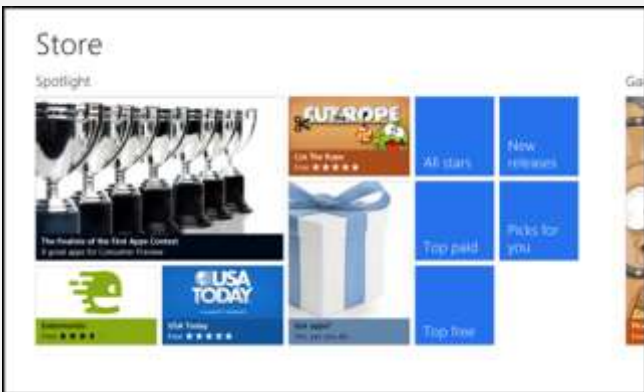
A new Metro style UI where **touch is a first-class citizen** along with full mouse-and-keyboard support



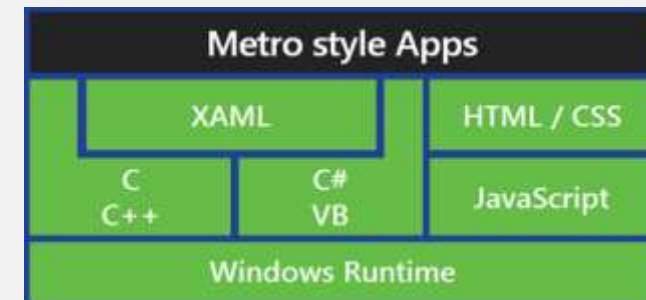
Designed from the chipset up for **multiple form-factors** – tablets, laptops, desktops & all-in-ones



The **Windows Store on every device** with a full commerce platform and flexibility



**New development models** built on WinRT, including native support for HTML/CSS/JS, C#/XAML, C++/DirectX



The Microsoft logo is centered on a solid blue background. The logo itself is white and features the word "Microsoft" in a bold, italicized sans-serif font, followed by a registered trademark symbol (®). The background is decorated with several white-outlined squares of various sizes, some of which are partially cut off by the edges of the frame, creating a geometric pattern.

***Microsoft®***



# In-Box Controls for Metro Style Apps

Button



Checkbox



Radio Button



Hyperlink

<http://www.buildwindows.com>

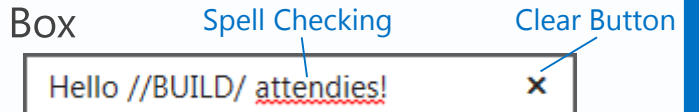
Combo Box

California

Grid View



Text Box



Progress Ring



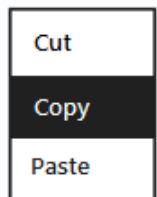
Progress Bar



Password



Context Menu



Flyout



List View



Semantic Zoom



Rating



Radio Button



Flip View



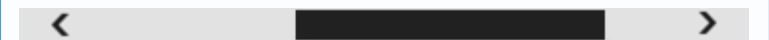
List Box



Panning Indicator



Scroll Bar



Toggle Switch



Tooltip



App Bar

