



WIN-015

ROOM
B

Windows 10 for Phone の 全て

日本マイクロソフト株式会社
デベロッパー エバンジェリズム統括本部
エバンジェリスト

高橋 忍



WIN-015

ROOM
B

Windows Phone と Windows 10 Mobile のお話

日本マイクロソフト株式会社
デベロッパー エバンジェリズム統括本部
エバンジェリスト

高橋 忍

Agenda

Windows 10 Mobile

Windows 10 Mobile まで

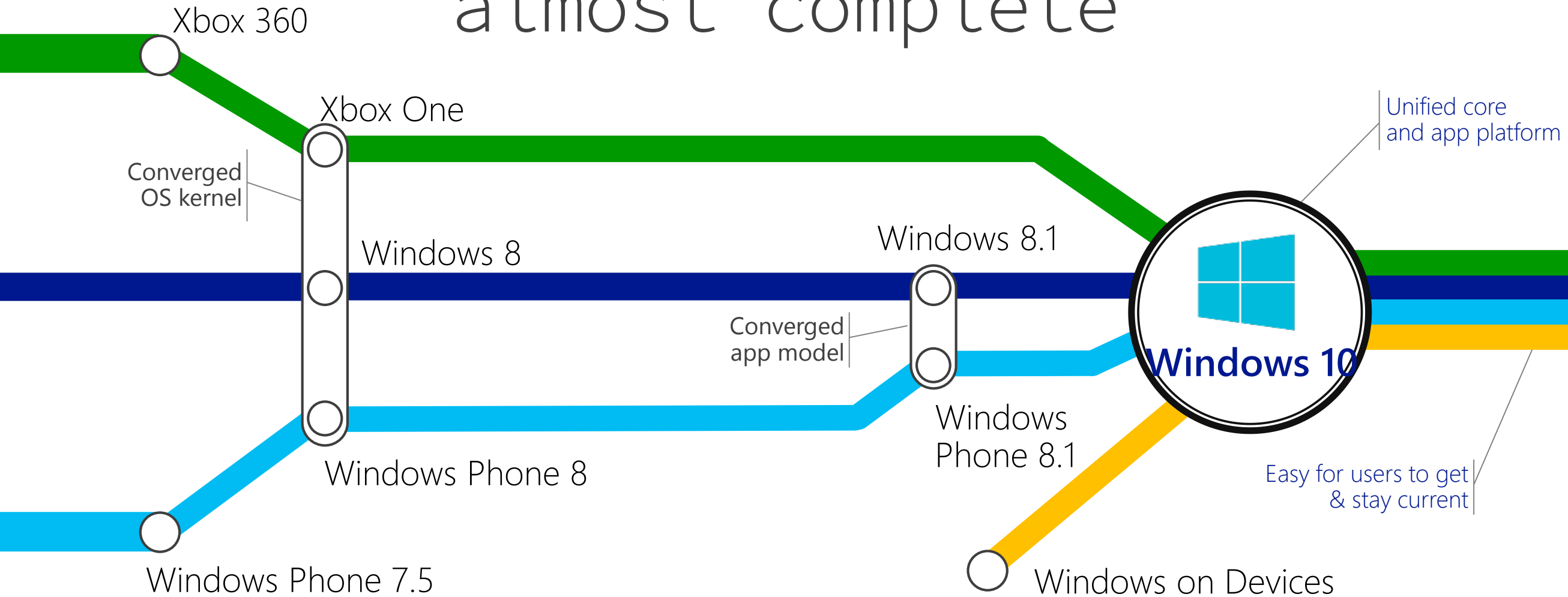
Windows 10 Mobile とは

Universal Windows Platform App

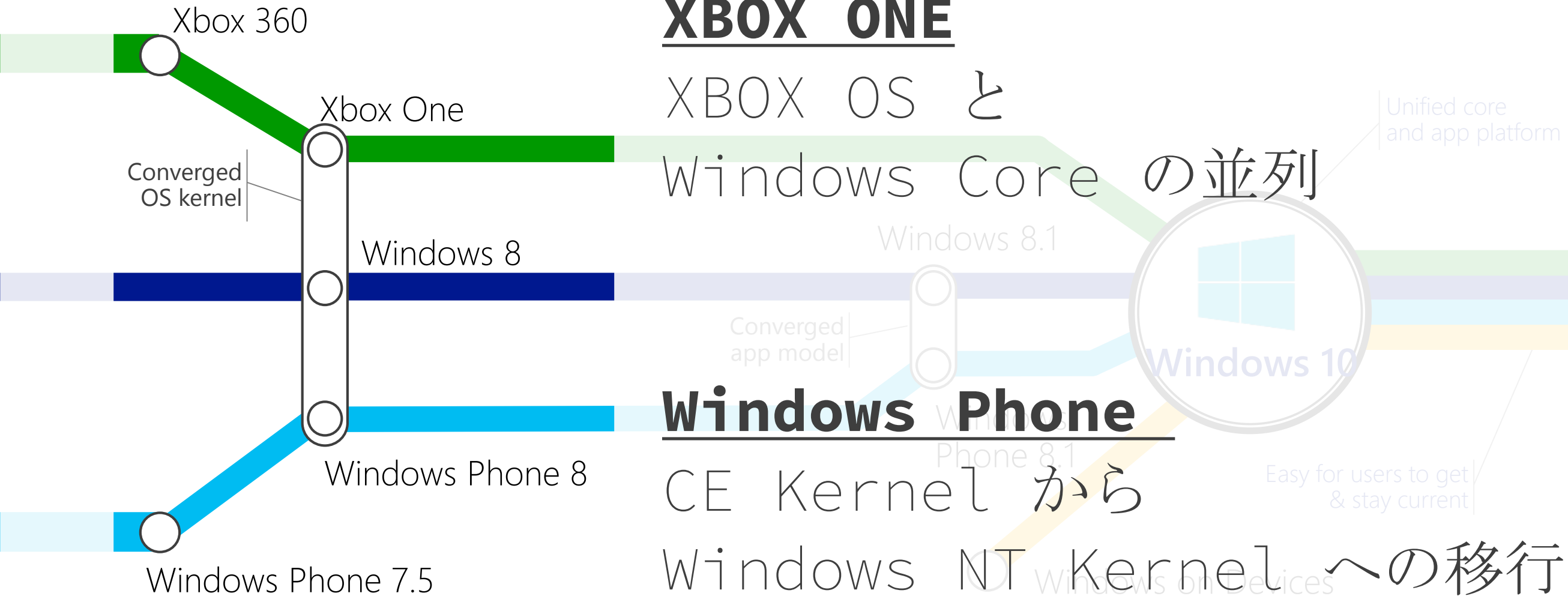


Windows 10 Mobile まで

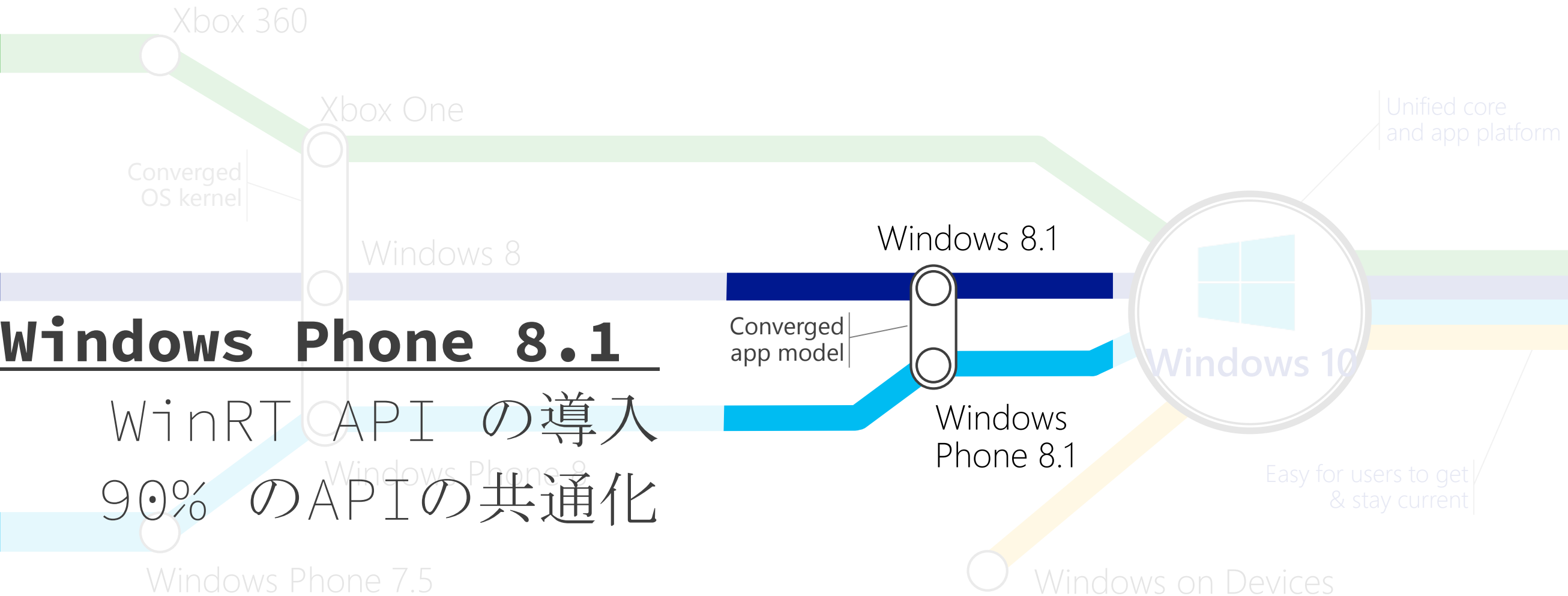
Our Windows convergence is almost complete



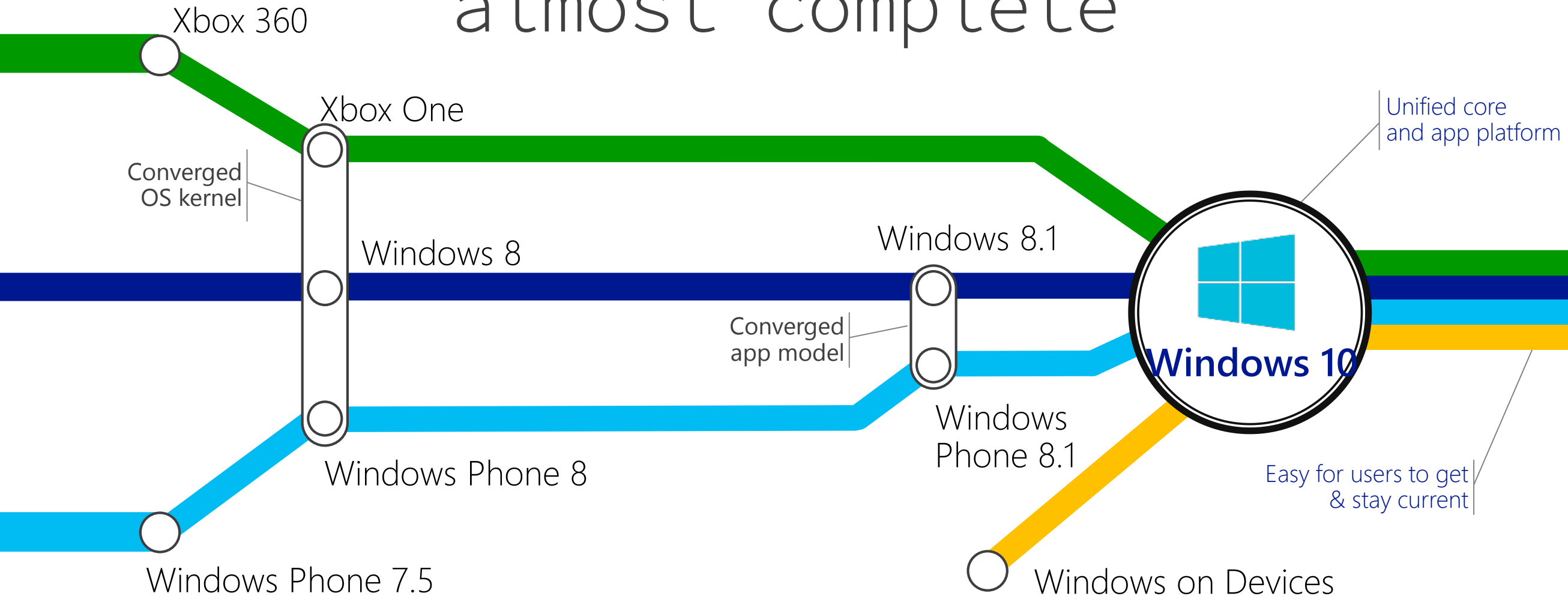
8.0 Kernel



8.1 Application Model



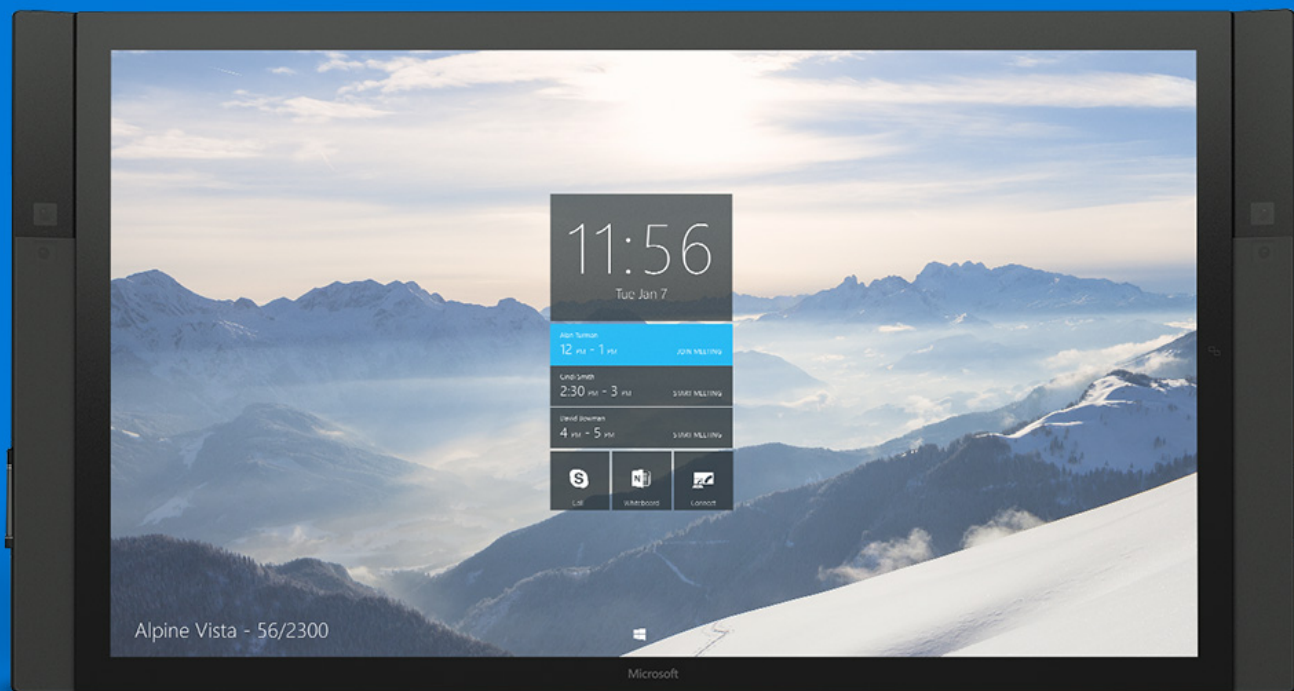
Our Windows convergence is almost complete



Windows Phone 10



Windows 10 Mobile



 Windows 10



Windows 10 Lineup

- Windows 10 Home
- Windows 10 Pro
- Windows 10 Enterprise
- Windows 10 Education
- **Windows 10 Mobile**
- **Windows 10 Mobile Enterprise**
- Windows 10 IoT Core



Windows 10 Mobile

*今後仕様が変更になる場合がございます

います

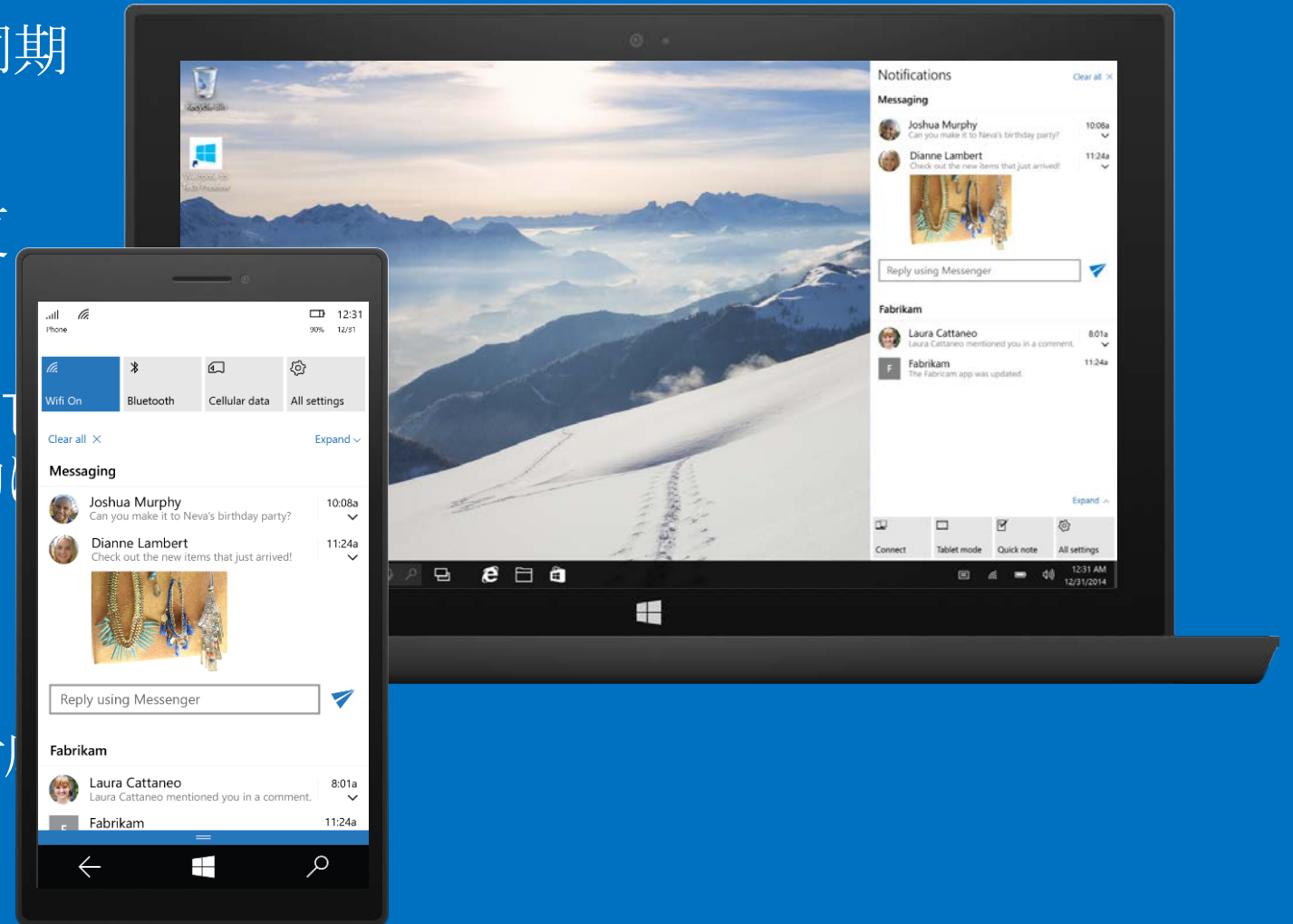
- Firmware **UEFI** 2.3.1
- Memory **512MB** for 32bit
- Storage **>=4GB** of Flash (SD Card require for 4GB Device)
- Graphics DirectX **9**
- Button **Power**, Volume **Up/Down**
Start, Back, search for Phone with WVGA display
- Screen Size 3-**7.99**''
- Resolution 800x480 (WVGA) –
2560x2048 (OS7GA)

Windows 10 Mobile とは

Windows 10 Mobile

Action Center 拡張、デバイス間同期
Microsoft Edge が標準ブラウザに
メールとカレンダーがOutlookに変更
Office はUWPを提供

Continuum for Windows 10 Mobile
外部モニタと接続してデスクトップ的
マップを拡張・日本語地図の対応
Bluetooth Profile の拡張
キーボードの拡張・大画面に対する対応
UWPとなったPeople, Map, Photos



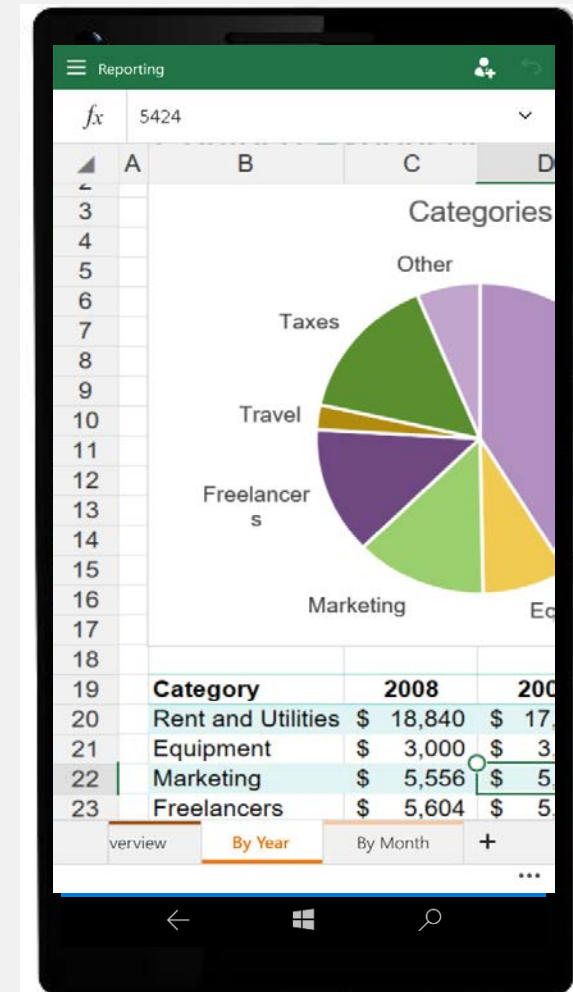
Continuum

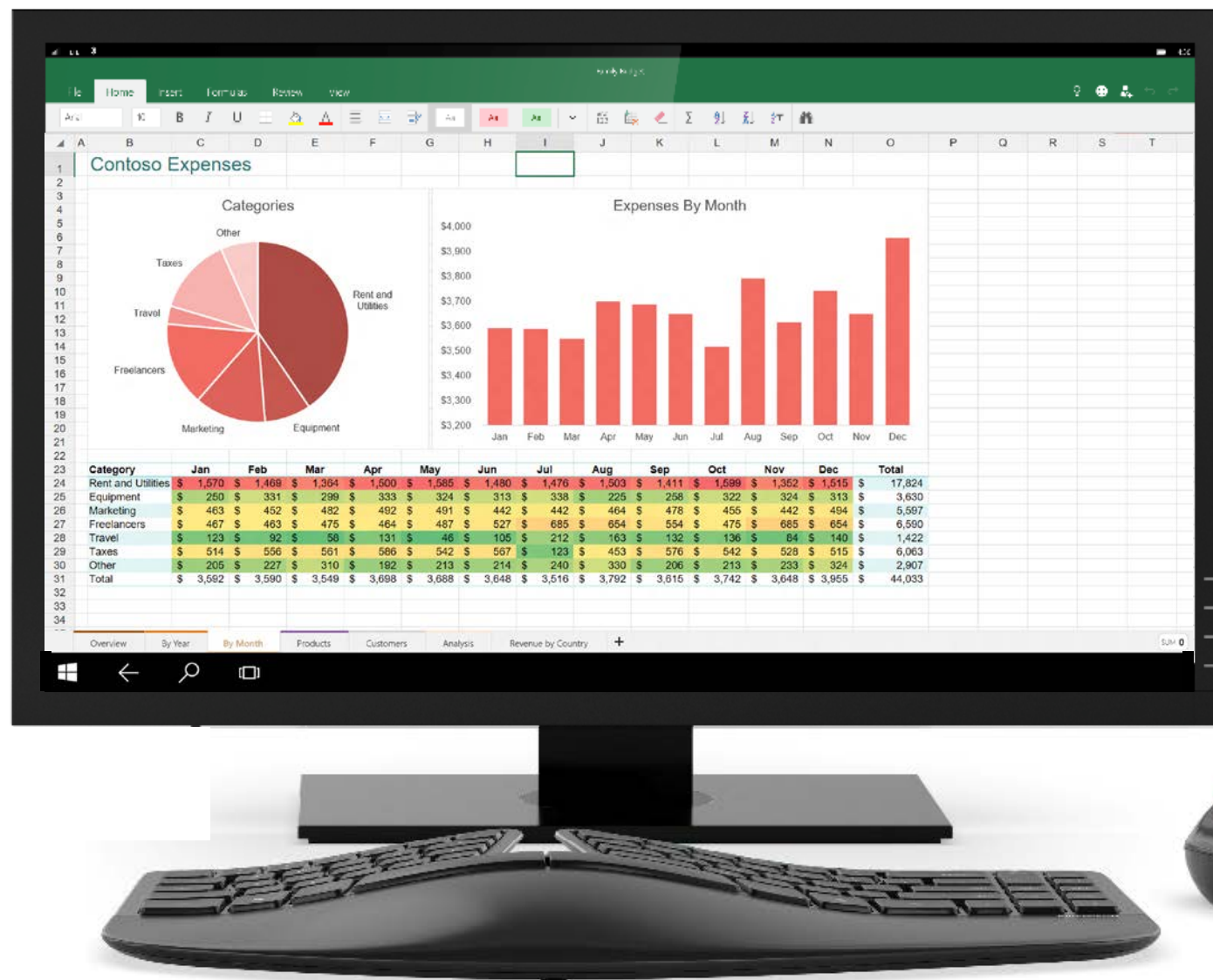
Phone で使える外部モニタ連携

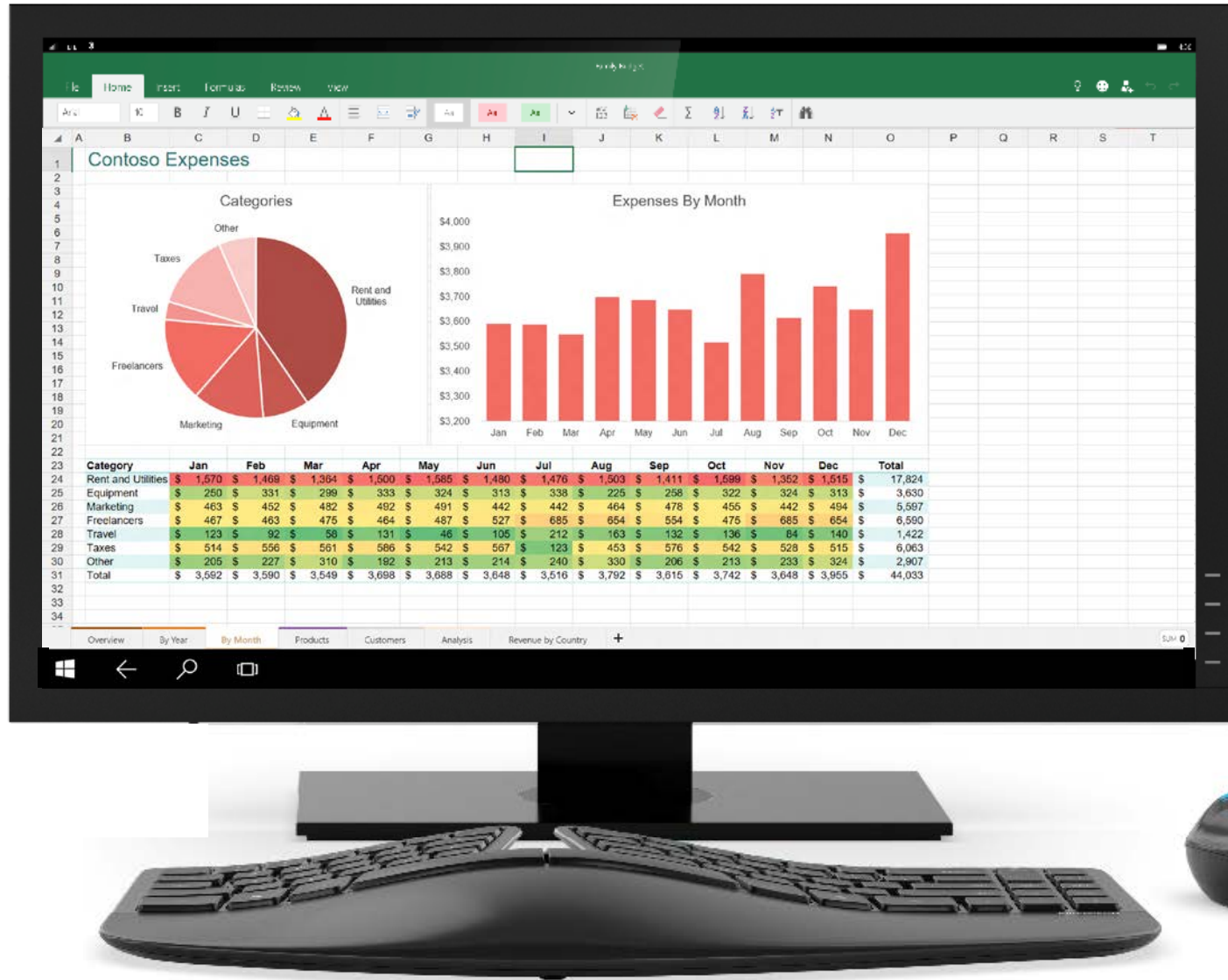
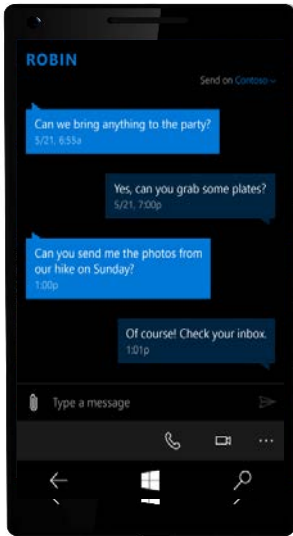
- デスクトップ的な使い方
- 接続して表示するだけ
- デバイスはコントローラ
- 電話しながらも



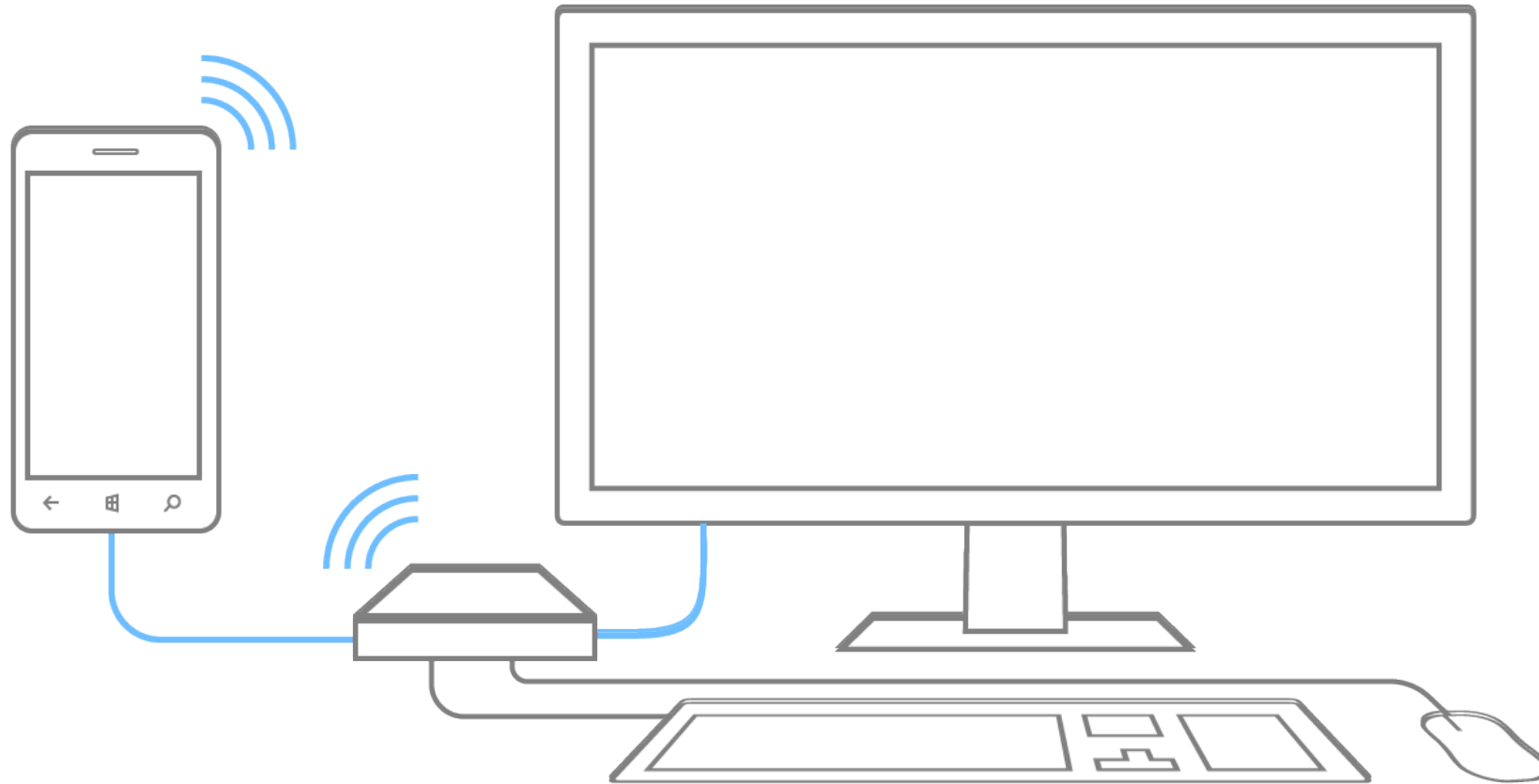
Sometimes a phone is
not enough



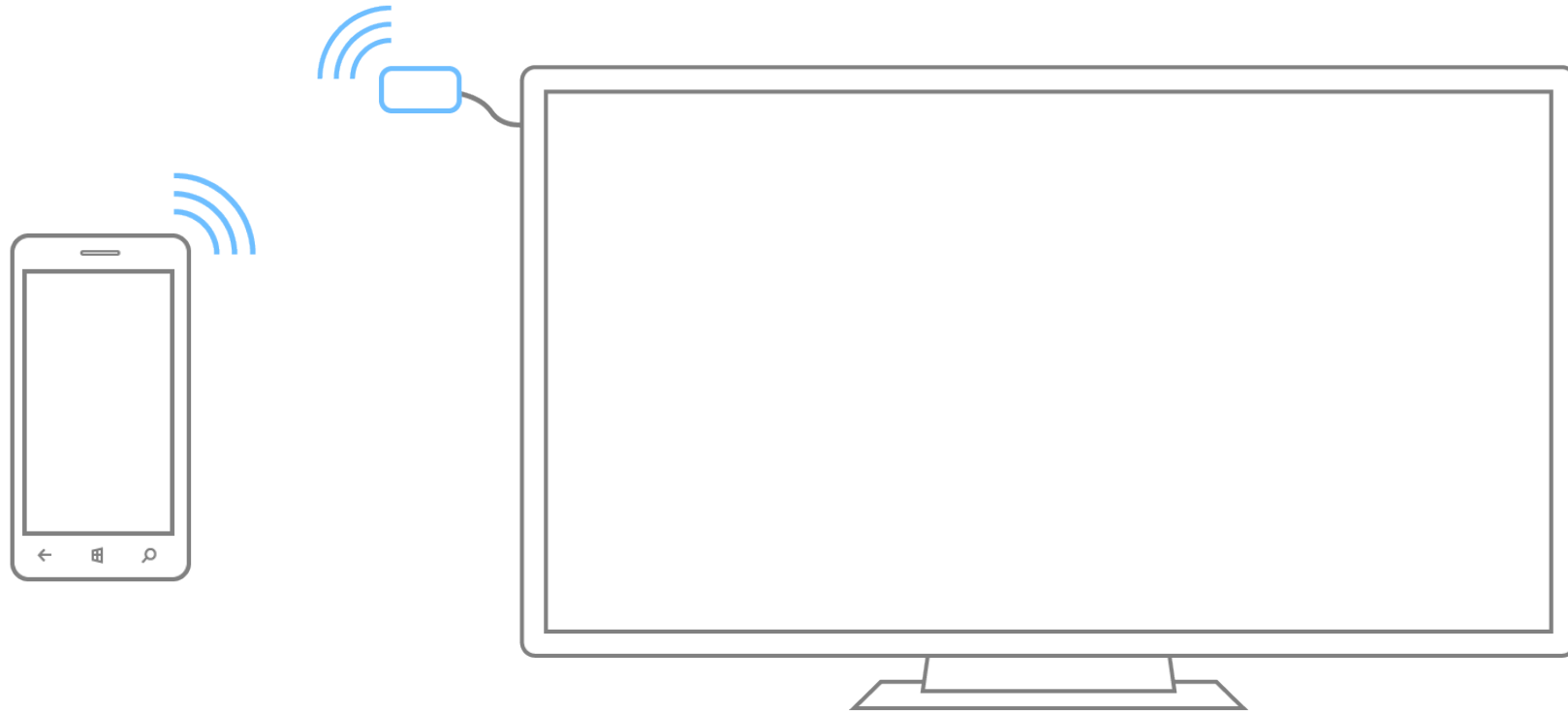




Wireless dock



Wireless dongle



Enterprise Mobility

2要素認証

シングルサインオン

ユーザーデータと業務データの分離管理

信頼されたアプリのみがユーザーデータに接続

企業データ保護機能をモバイルでも

マルウェアに対するハードウェアプロテクト

サンドボックスモデルによるアプリからの保護

暗号化による情報の保護



Enterprise Data Protection

One experience

Data is isolated

Data is encrypted at rest

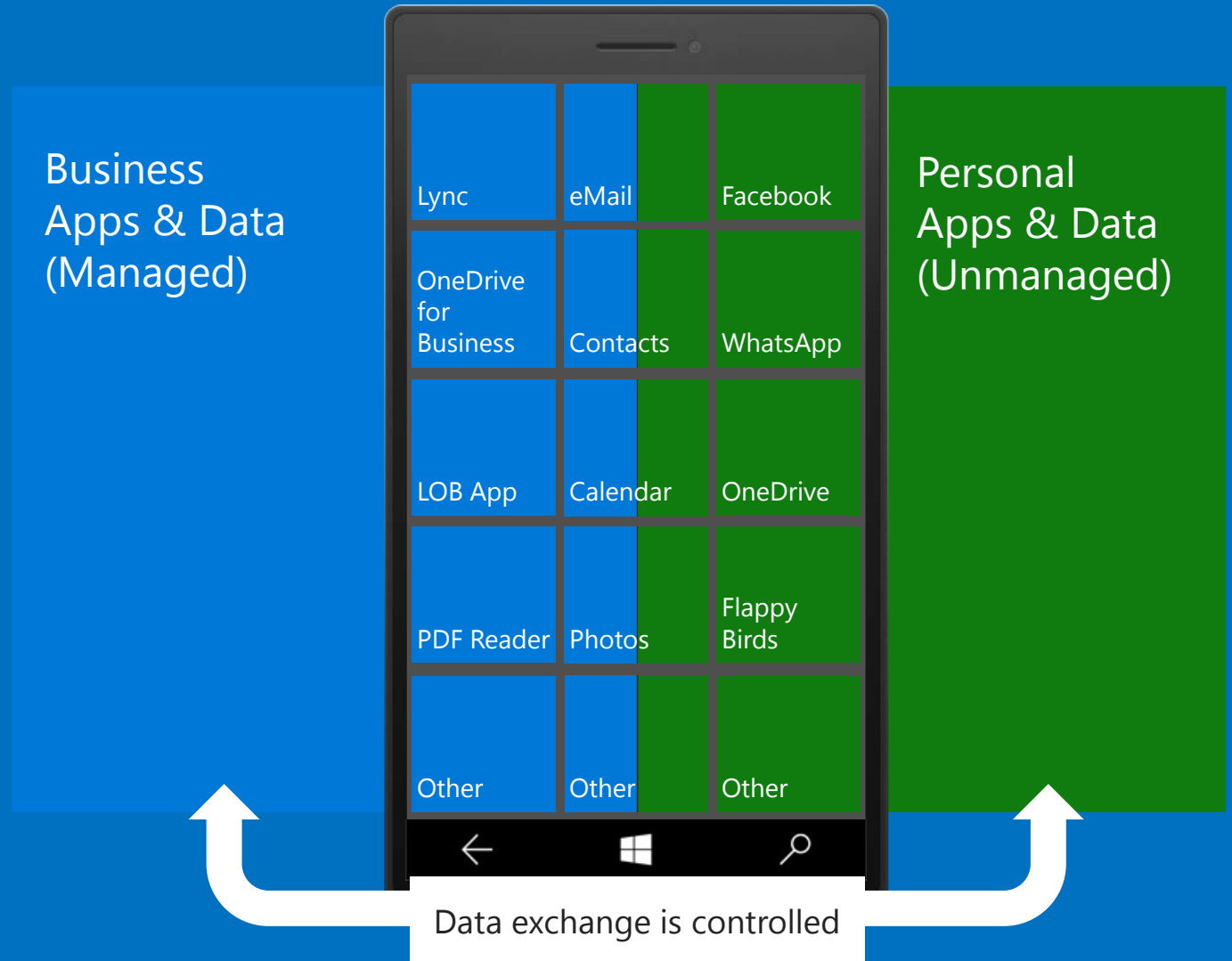
MDM managed

Block/audit data exchange

Organization holds keys

Enlightened Apps

APIs for 3rd party apps



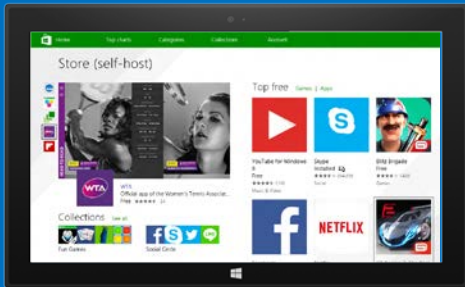
Ref : CSP

<https://msdn.microsoft.com/en-us/library/windows/hardware/dn920025>



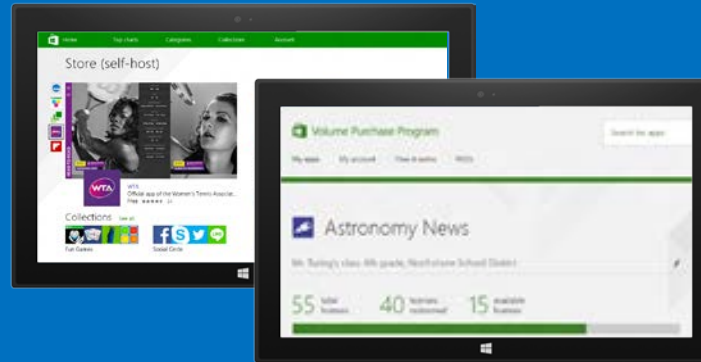
A Store open for business

Windows Store



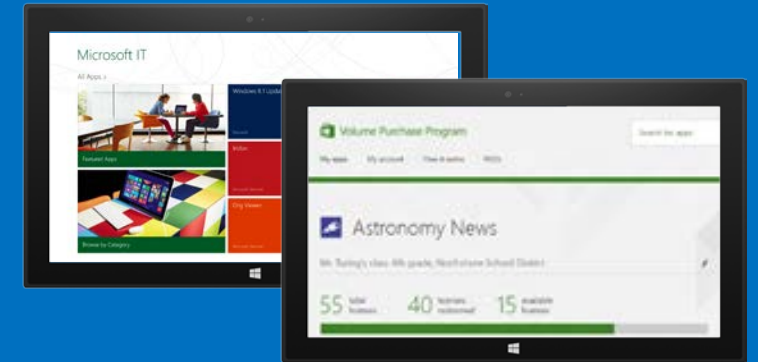
- ストアアプリ
- Microsoft Account でサインイン
- クレジットカード、ギフトカード、PayPal、Airpay、INICIS、キャリア課金での支払い

Business Store Portal



- ストアアプリ
- Azure Active Directory の利用も可
- Private organization store for the org's preferred or LOB apps
- クレジットカード、PO/invoiceでの支払い
- アプリのオフライン配布
- ストアアプリのライセンス管理

“Company Portal”



- 業務用ストアアプリのサイドローディング
- Windows Store を通じてのアプリの配布

Universal Windows Platform App

Windows 10 Mobile と Application

- 動作するアプリケーション

- Windows Phone 7.x/8.x Silverlight Application (XAP)
- Windows Phone 7.x/8.x WinRT Application (AppX)
- Universal Application
- もちろん、デスクトップアプリケーションは (Windows 10 といっても) 使えない
- Universal Windows Platform App (Desktop 専用APIもあるので注意)

- 特にユーザーエクスペリエンスは変わらず

- 差分ダウンロード・アップロード
- 外部ストレージへのアプリのインストール
- 非公開・ベータテスト用の公開も引き続き可能に

UWP App で出来るになったこと

- MediaElement の複数起動
- レスポンシブデザイン
- アプリ間連携（URIアクティベーション）の拡張：双方向
- パブリッシャー共有ストレージフォルダ
- カスタムセンサー、バッテリー関連等
- 広告用Bluetooth企画サポート（Beacon）
- Wi-Fi Direct の更新
- ファイル検索APIの提供

Mobile Device Family API

Contract: 固有機能

<https://msdn.microsoft.com/en-us/library/windows/apps/dn706138.aspx>

- Windows.ApplicationModel.Activation.**WebUISearchActivatedEventsContract**
- Windows.ApplicationModel.**Calls**.CallsPhoneContract
- Windows.ApplicationModel.SocialInfo.**SocialInfoContract**
- Windows.ApplicationModel.**CommunicationBlocking**.CommunicationBlockingContract
- Windows.ApplicationModel.**Wallet**.WalletContract
- Windows.Devices.SmartCards.**SmartCardBackgroundTriggerContract**
- Windows.Devices.SmartCards.**SmartCardEmulatorContract**
- Windows.Phone.**PhoneContract**
- Windows.Phone.StartScreen.**DualSimTileContract**
- Windows.Security.EnterpriseData.**EnterpriseDataContract**
- Windows.Security.**ExchangeActiveSyncProvisioning**.EasContract
- Windows.System.UserProfile.**UserProfileContract**
- Windows.UI.WebUI.Core.**WebUICommandBarContract**

Microsoft Mobile Extension SDK

参照マネージャー - App5

フィルターの適用先: App5 に該当する SDK

Windows Universal の検索 (C 🔍)

名前: Microsoft Mobile Extension SDK for Universal App Platform

バージョン: 10.0.0.1

Targets: UAP 10.0.0.1

[詳細情報](#)

	名前	バージョン
	Behaviors SDK (XAML)	12.0
	Microsoft .NET Core Runtime Package for Windo...	1.0
	Microsoft Desktop Extension SDK for Universal A...	10.0.100...
<input type="checkbox"/>	Microsoft Mobile Extension SDK for Universal Ap...	10.0.0.1
	Microsoft Visual C++ AppLocal Runtime Packag...	14.0
	Microsoft Visual C++ AppLocal Runtime Packag...	14.0
	Microsoft Visual C++ 2013 Runtime Package for...	12.0
	Microsoft Visual C++ 2013 Runtime Package for...	14.0

アセンブリ

プロジェクト

共有プロジェクト

Windows Universal

コア

拡張

参照

C:\Program Files
(x86)
 \Windows Kits
 \10
 \Extension
 SDKs

移植する

Windows 10 Mobile 限定

特定のデバイスファミリー向けにアプリを提供する (Desktop/Mobile/Xbox/IoT/...)

設定

Package.appxmanifest

設定例

<Dependencies>

<TargetDeviceFamily Name="Windows.Mobile"
MinVersion="10.0.1.0" ... />

元は Windows.Universal

XAML に関する変更点

- デフォルトのテーマが異なります
 - デスクトップでは Light がデフォルトです (8.1 では Dark がデフォルト)
 - 古い設定をしたい場合は `Set RequestedTheme="Dark"`
- AppBar と CommandBar
 - `IsSticky=true` makes it not light dismissible
- SearchBox はなし
 - AutoSuggestBox を使います
- Date/TimePicker
 - 表示が異なります
- Pivot control is converged

XAML Upgrade Breaking Changes

- GridView/ListView
 - GridView – horizontal の代わりに Vertical Flow
 - 選択モデルに対する新しい表示
 - スワイプジェスチャーは推奨しない
- SemanticZoom
 - グループヘッダーをタップしてズームアウト、ピンチズーム
 - Mobileではズームアウトなし
- ContentDialog
 - MessageDialog よりより柔軟に
- Some missing ‘UnConverged’ Styles
 - いくつかの Phone* & Phone コントロールスタイルがない

ネームスペースの変更

8.1

```
<Package
  xmlns="http://schemas.microsoft.com/appx/2010/manifest"
  xmlns:m2="http://schemas.microsoft.com/appx/2013/manifest">

  <Identity Name="78ec24f6-149a-47aa-b5a2-f854ab7119f5"
    Publisher="CN=samjar"
    Version="1.0.0.0" />

  <Properties>
    <DisplayName>My Awesome App</DisplayName>
    <PublisherDisplayName>samjar</PublisherDisplayName>
    <Logo>Assets¥StoreLogo.png</Logo>
  </Properties>

  <Resources>
    <Resource Language="en-us"/>
    < Resource m2:Scale="180"/>
    < Resource m2:DXFeatureLevel="dx11"/>
  </Resources>
```

...

10.0

```
<Package
  xmlns="http://schemas.microsoft.com/appx/manifest/foundation/windows10"
  xmlns:uap="http://schemas.microsoft.com/appx/manifest/uap/windows10">

  <Identity Name="a6db432c-3f41-4475-ae4f-0e3168636cd2"
    Publisher="CN=samjar"
    Version="1.0.0.0" />

  <Properties>
    <DisplayName>My Amazing App</DisplayName>
    <PublisherDisplayName>samjar</PublisherDisplayName>
    <Logo>Assets¥StoreLogo.png</Logo>
  </Properties>

  <Resources>
    <Resource Language="en-us"/>
    <Resource uap:Scale="180"/>
    <Resource uap:DXFeatureLevel="dx11"/>
  </Resources>
```

...

頭にuap: が必要なコントロール

- documentsLibrary
- picturesLibrary
- videosLibrary
- musicLibrary
- enterpriseAuthentication
- sharedUserCertificates
- removableStorage
- Appointments
- Contacts

Example

```
<Capabilities>  
  <uap:Capability Name="documentsLibrary"/>  
  <uap:Capability Name="removableStorage"/>  
</Capabilities>
```

対象デバイスファミリー

8.1

```
<Prerequisites>
```

```
  <OSMinVersion>6.3.0</OSMinVersion>
```

```
  <OSMaxVersionTested>6.3.0</OSMaxVersionTested>
```

```
</Prerequisites>
```

10.0

```
<Dependencies>
```

```
  <TargetDeviceFamily
```

```
    Name="Windows.Universal"
```

```
    MinVersion="10.0.10053.0"
```

```
    MaxVersionTested="10.0.10053.0" />
```

```
</Dependencies>
```

VisualElement の違い

8.1

```
<m2:VisualElements
  DisplayName="My Awesome App"
  Square150x150Logo="Assets¥Logo.png"
  Square30x30Logo="Assets¥SmallLogo.png"
  Description="My Awesome App"
  BackgroundColor="#464646"
  ForegroundText="light"
  ToastCapable="True">
<m2:SplashScreen Image="Assets¥SplashScreen.png" />
<m2:DefaultTile
  ShortName="MyApp"
  Wide310x150Logo="310x150.png"
  Square310x310Logo="310x310.png"
  Square70x70Logo="70x70.png"
  DefaultSize="square150x150Logo"/>
<m2:ApplicationView MinWidth="width320"/>
</m2:VisualElements>
```

10.0

```
<uap:VisualElements
  DisplayName="My Amazing App"
  Square150x150Logo="Assets¥Logo.png"
  Square44x44Logo="Assets¥SmallLogo.png"
  Description="My Amazing App"
  BackgroundColor="#464646">
  <uap:SplashScreen
    Image="Assets¥SplashScreen.png" />
  <uap:DefaultTile
    ShortName="MyApp"
    Wide310x150Logo="310x150.png"
    Square310x310Logo="310x310.png"
    Square71x71Logo="71x71.png"/>
</uap:VisualElements>
```

ネームスペースの変更

8.1

```
<m2:VisualElements
  DisplayName="My Awesome App"
  Square150x150Logo="Assets¥Logo.png"
  Square30x30Logo="Assets¥SmallLogo.png"
  Description="My Awesome App"
  BackgroundColor="#464646"
  ForegroundText="light"
  ToastCapable="True">
<m2:SplashScreen Image="Assets¥SplashScreen.png" />
<m2:DefaultTile
  ShortName="MyApp"
  Wide310x150Logo="310x150.png"
  Square310x310Logo="310x310.png"
  Square70x70Logo="70x70.png"
  DefaultSize="square150x150Logo"/>
<m2:ApplicationView MinWidth="width320"/>
</m2:VisualElements>
```

10.0

```
<uap:VisualElements
  DisplayName="My Amazing App"
  Square150x150Logo="Assets¥Logo.png"
  Square44x44Logo="Assets¥SmallLogo.png"
  Description="My Amazing App"
  BackgroundColor="#464646">
  <uap:SplashScreen
    Image="Assets¥SplashScreen.png" />
  <uap:DefaultTile
    ShortName="MyApp"
    Wide310x150Logo="310x150.png"
    Square310x310Logo="310x310.png"
    Square71x71Logo="71x71.png"/>
</uap:VisualElements>
```

タイ尔サイズ

8.1

```
<m2:VisualElements
  DisplayName="My Awesome App"
  Square150x150Logo="Assets¥Logo.png"
  Square30x30Logo="Assets¥SmallLogo.png"
  Description="My Awesome App"
  BackgroundColor="#464646"
  ForegroundText="light"
  ToastCapable="True">
<m2:SplashScreen Image="Assets¥SplashScreen.png" />
<m2:DefaultTile
  ShortName="MyApp"
  Wide310x150Logo="310x150.png"
  Square310x310Logo="310x310.png"
  Square70x70Logo="70x70.png"
  DefaultSize="square150x150Logo"/>
<m2:ApplicationView MinWidth="width320"/>
</m2:VisualElements>
```

10.0

```
<uap:VisualElements
  DisplayName="My Amazing App"
  Square150x150Logo="Assets¥Logo.png"
  Square44x44Logo="Assets¥SmallLogo.png"
  Description="My Amazing App"
  BackgroundColor="#464646">
  <uap:SplashScreen
    Image="Assets¥SplashScreen.png" />
  <uap:DefaultTile
    ShortName="MyApp"
    Wide310x150Logo="310x150.png"
    Square310x310Logo="310x310.png"
    Square71x71Logo="71x71.png"/>
</uap:VisualElements>
```

なくなった Elements

8.1

```
<m2:VisualElements
  DisplayName="My Awesome App"
  Square150x150Logo="Assets¥Logo.png"
  Square30x30Logo="Assets¥SmallLogo.png"
  Description="My Awesome App"
  BackgroundColor="#464646"
  ForegroundText="light"
  ToastCapable="True">
  <m2:SplashScreen Image="Assets¥SplashScreen.png" />
  <m2:DefaultTile
    ShortName="MyApp"
    Wide310x150Logo="310x150.png"
    Square310x310Logo="310x310.png"
    Square70x70Logo="70x70.png"
    DefaultSize="square150x150Logo"/>
  <m2:ApplicationView MinWidth="width320"/>
</m2:VisualElements>
```

10.0

```
<uap:VisualElements
  DisplayName="My Amazing App"
  Square150x150Logo="Assets¥Logo.png"
  Square44x44Logo="Assets¥SmallLogo.png"
  Description="My Amazing App"
  BackgroundColor="#464646">
  <uap:SplashScreen
    Image="Assets¥SplashScreen.png" />
  <uap:DefaultTile
    ShortName="MyApp"
    Wide310x150Logo="310x150.png"
    Square310x310Logo="310x310.png"
    Square71x71Logo="70x70.png"/>
</uap:VisualElements>
```


Back Button のハンドリング

- Single converged back button API

Example:

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    var frame = Window.Current.Content as Frame;
    if (frame.CanGoBack)
    {
        SystemNavigationManager.GetForCurrentView().AppViewBackButtonVisibility = AppViewBackButtonVisibility.Visible;
    }
    else
    {
        SystemNavigationManager.GetForCurrentView().AppViewBackButtonVisibility = AppViewBackButtonVisibility.Collapsed;
    }

    Window.UI.Core.SystemNavigationManager.GetForCurrentView().BackRequested += Page_BackRequested;
}
```

チェックしたい技術

IME関連のAPIを拡張

Win32時代に合ったIME関連のAPIを一部提供

- 入力に対する変換候補リストの取得
`TextConversionGenerator.GetCandidateAsync`
- 入力に対する予測変換候補リストの取得
`TextPredictionGenerator.GetCandidateAsync`
- 漢字に対する読み仮名の取得
`TextReverseCenversionGenerator.ConvertBackAsync`

カメラキャプチャー

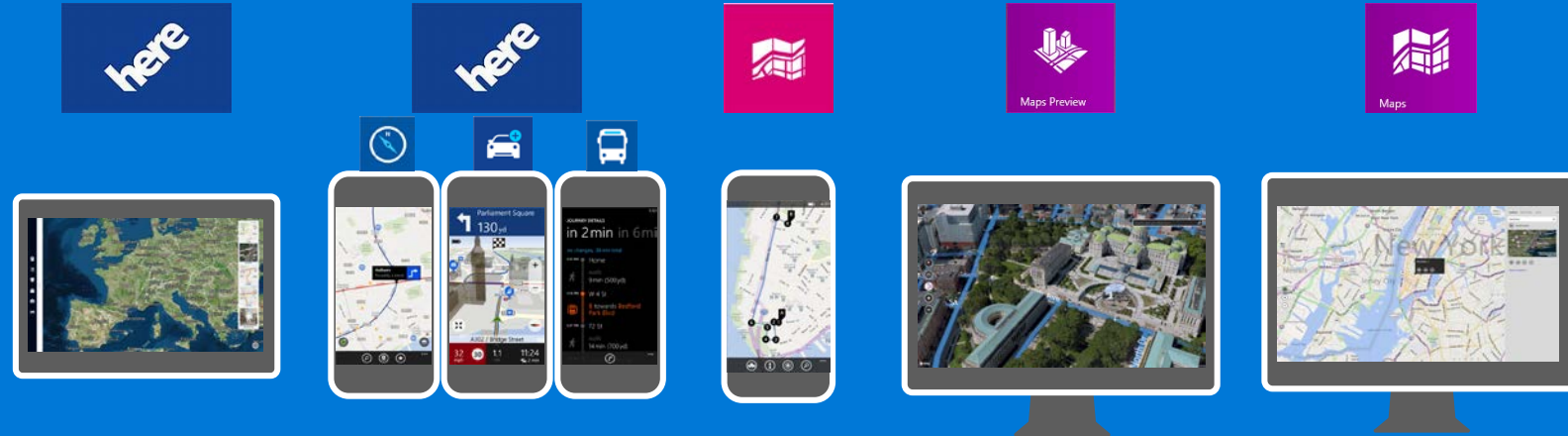
- Software Bitmap
- Preview Frame Access†
- Face Detection†
- Pause/Resume Recording
- Video Stabilization
- Video Frame
- Geotag Helper†
- Variable Photo Sequence
- Lamp (flash) without Camera†*
- Camera Profiles†*
- Smooth Zoom*
- HDR Video*
- Square Video Recording
- Scene Analysis
- Enterprise Policy & Control

† Sample available today

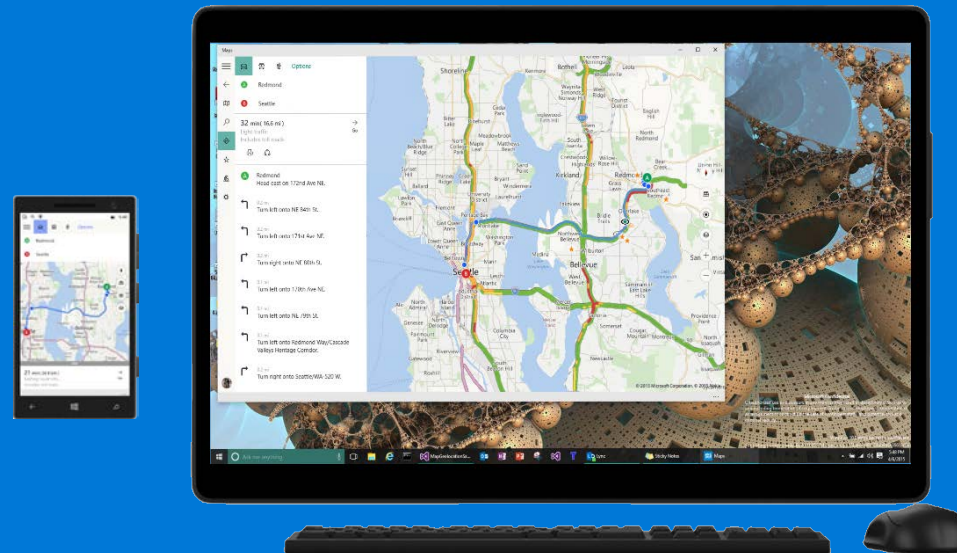
* Requires Windows 10 devices

Map Control

Windows 8.1



Windows 10



MapControl Basics

- **ZoomLevel** – 1 to 20 (max zoomed in)
- **Heading** – 0 (north), 90(west), 180(south), 270(east) to 360
- **DesiredPitch** – 0(top down) to 64(horizon)



```
xmlns:maps="using:Windows.UI.Xaml.Controls.Maps"
```

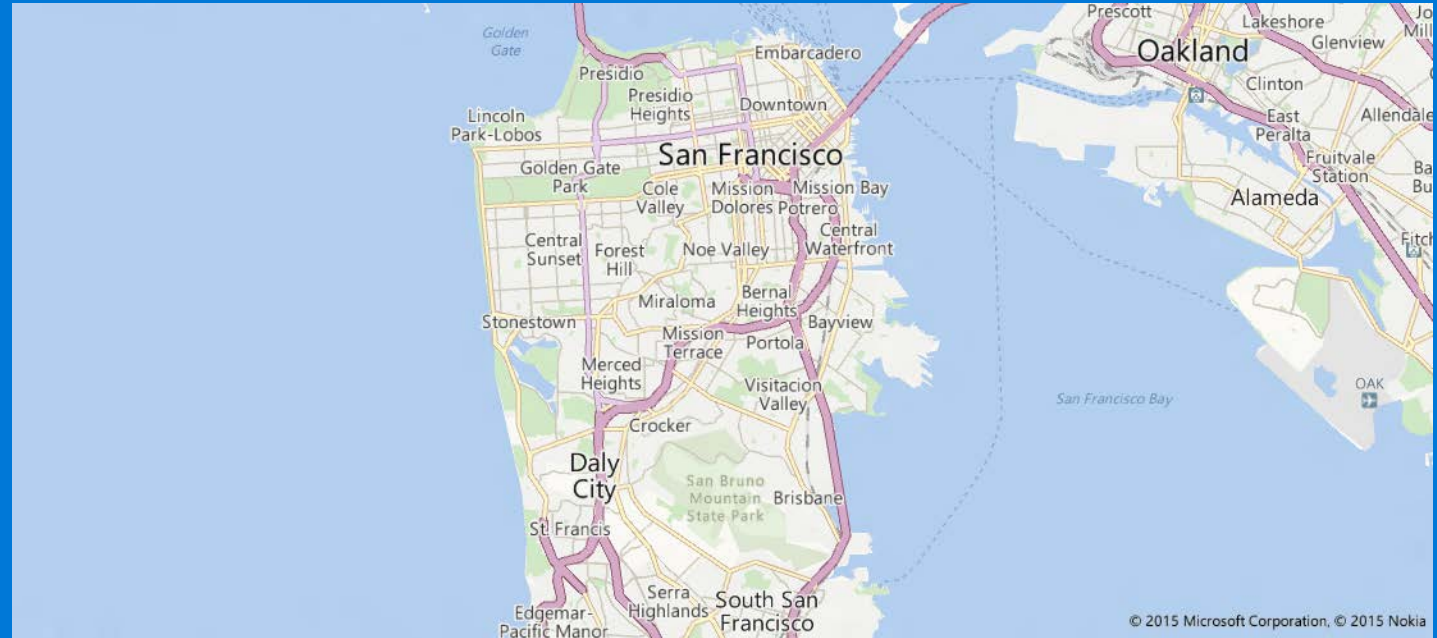
```
<Grid>
```

```
    <maps:MapControl x:Name="myMap" />
```

```
</Grid>
```

MapControl Basics

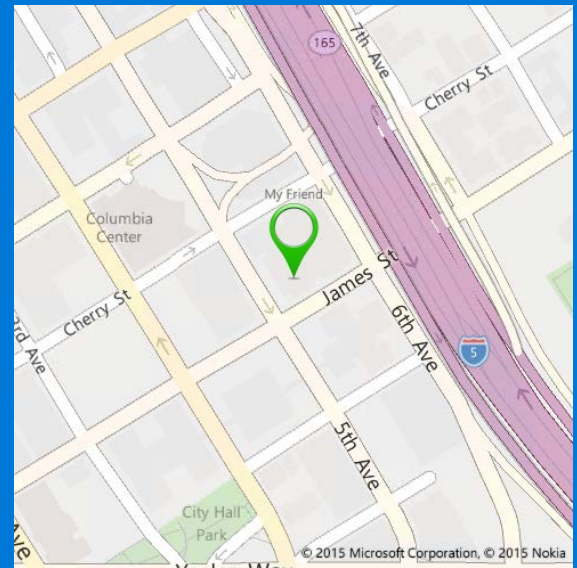
Center the Map



```
myMap.Center = new Geopoint(new BasicGeoposition
{
    // Geopoint for San Francisco
    Latitude = 37.733795,
    Longitude = -122.446747,
});
myMap.ZoomLevel = 12;
```


MapControl Basics

Adding MapIcon



```
MapIcon mapIcon1 = new MapIcon();  
mapIcon1.Location = myMap.Center;  
mapIcon1.NormalizedAnchorPoint = new Point(0.5, 1.0);  
mapIcon1.Image = RandomAccessStreamReference.CreateFromUri(new  
Uri("ms-appx:///Assets/MapPin.png"));  
mapIcon1.CollisionBehaviorDesired =  
MapElementCollisionBehavior.RemainVisible;  
myMap.MapElements.Add(mapIcon1);
```

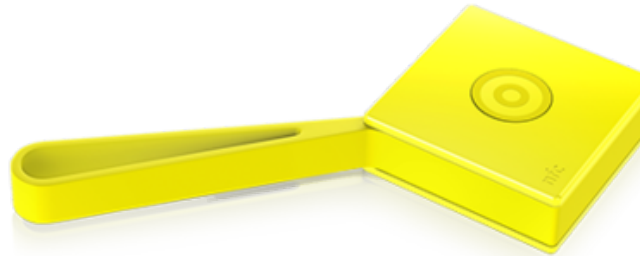

Bluetooth LE "beacon"?



Advertisements



Presence Detection



Proximity



Windows as the Beacon

Bluetooth New APIs in Windows 10

Windows.Devices.Bluetooth.Advertisement

広告用にシンプルかつ強力なAPI

Watcher – ADの受信. **Publisher** – 限定配信.

Windows.Devices.Bluetooth.Background

Fully converged. Trigger details classes.

Windows.ApplicationModel.Background

Power efficient watcher. Requires Hardware Offload.

Background Audio

Rich Player App

- アプリケーション UI を表示
- 対ユーザープレイリストの管理
- バックグラウンドタスクとの通信連携

Background Media Player

- プロセス間通信の管理
- MediaPlayer の表示

Low Memory Background Task

- 内部プレイリストの管理
- MediaPlayer のコントロール
- プレイヤーアプリでのUI変更の調整
- SMTC の実相

System Media Transport Controls

- OS-level UI 表示
- ユーザーからの Bluetooth / Kinect などのハードウェアボタンによるメディアコントロールを許可



バックグラウンドプレイヤーへのメッセージ送信

```
private void nextButton_Click(object sender, RoutedEventArgs e)
{
    BackgroundMediaPlayer.SendMessageToBackground("next");
}
```

バックグラウンドプレイヤーへの詳細メッセージ送信

```
private void NextButton_Click(object sender, RoutedEventArgs e)
{
    MessageService.SendMessageToBackground(new SkipNextMessage());
}

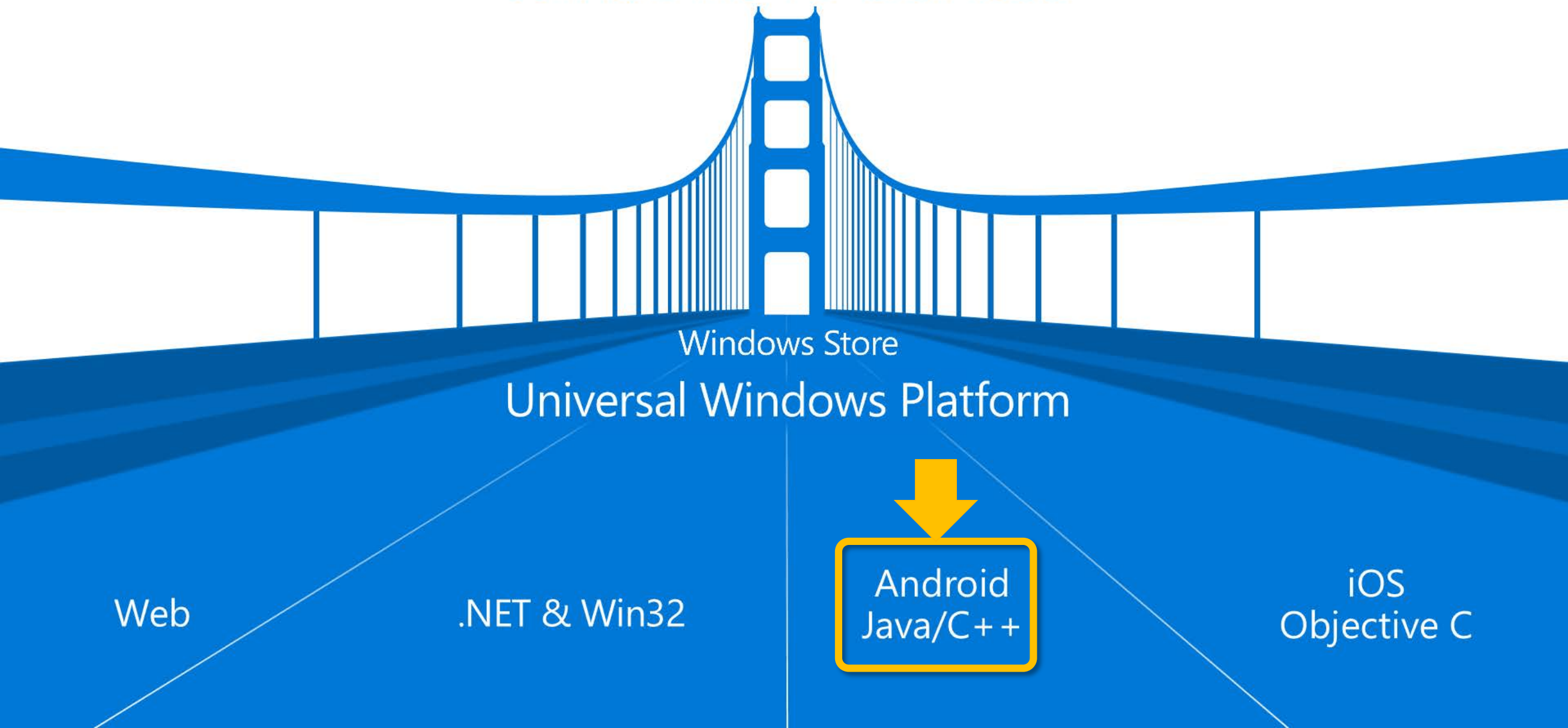
public static void SendMessageToBackground<T>(T msg)
{
    var payload = new ValueSet();
    payload.Add(MessageService.MessageType, typeof(T).FullName);
    payload.Add(MessageService.MessageBody, JsonSerializer.ToJson(msg));
    BackgroundMediaPlayer.SendMessageToBackground(payload);
}
```

プレイリストの作成：ギャップレスオーディオ
プレイリストAPIs は自動的にギャップレス
転送を提供

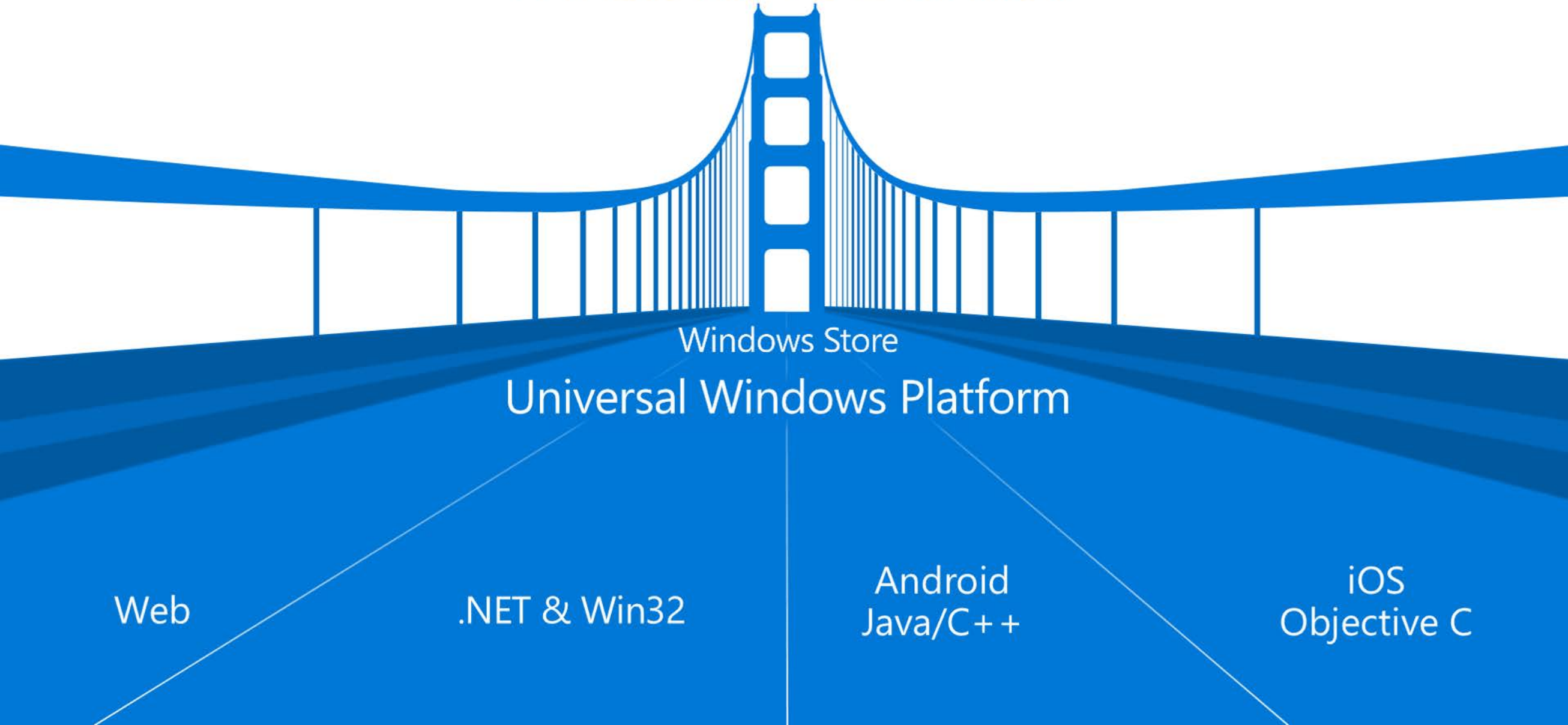
```
MediaPlaybackList playbackList = new MediaPlaybackList();
foreach (var song in songs)
{
    var source = MediaSource.CreateFromUri(song.MediaUri);
    source.CustomProperties[TrackIdKey] = song.MediaUri;
    source.CustomProperties[TitleKey] = song.Title;
    source.CustomProperties[AlbumArtKey] = song.AlbumArtUri;
    playbackList.Items.Add(new MediaPlayerItem(source));
}
BackgroundMediaPlayer.Current.SetPlaybackSource(playbackList);
```

Bridge

開発者の投資効果を最大化



1 Billion Windows 10 Devices

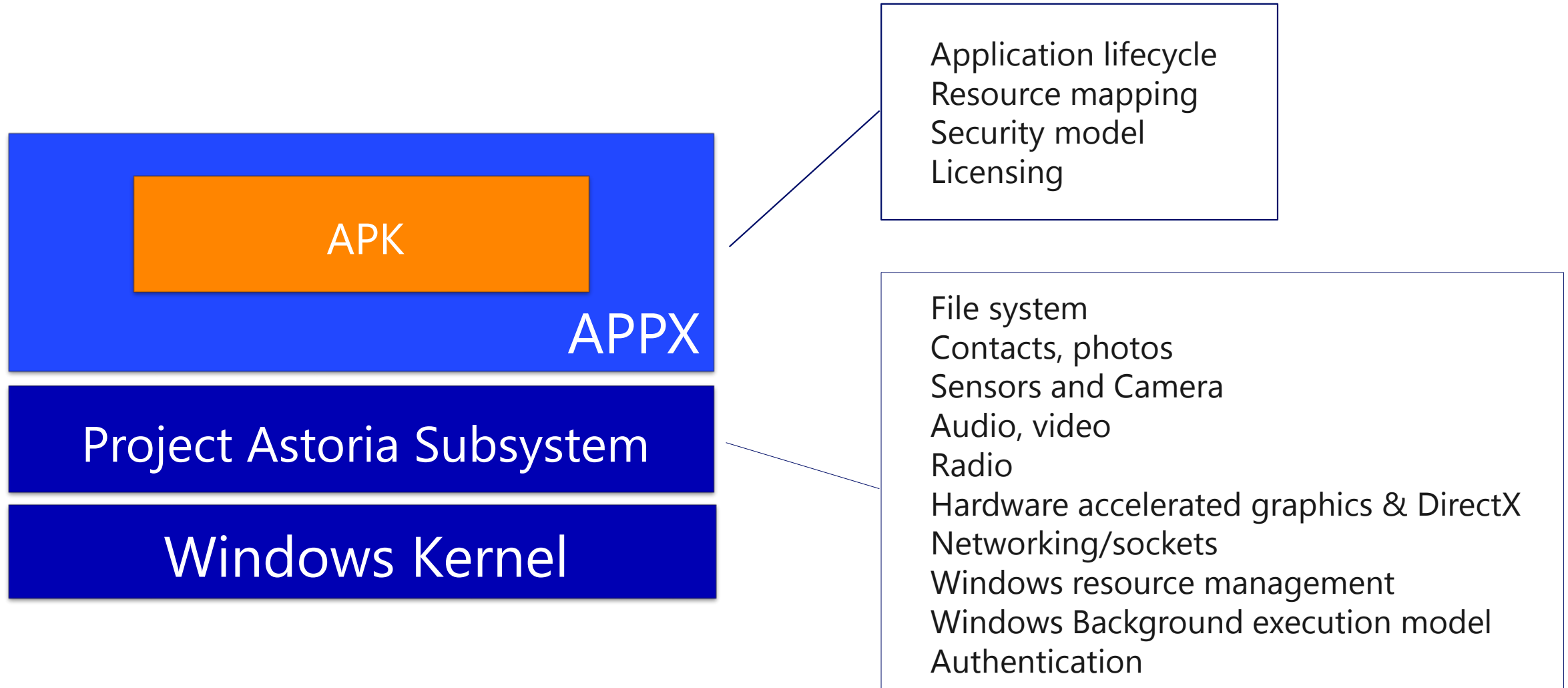


Project Astoria

- Android Apk を変換して Appx にするツールの提供
- バイナリの再利用、ソースコードは使わない
- ARM バイナリのみ / ARM デバイスのみ
=Windows 10 Mobile でのみ対象

Project Astoria Platform

Universal Windows Platform のポリシーに準拠



Project Astoria ワークフロー



まとめ

ひきつづき

Windows 10 Mobile を
愛してあげてください

