Microsoft

tech·days Korea 2014
개발자 컨퍼런스

윈도우와 윈도우폰 앱을 한번에 - 유니버셜 윈도우 앱 개발

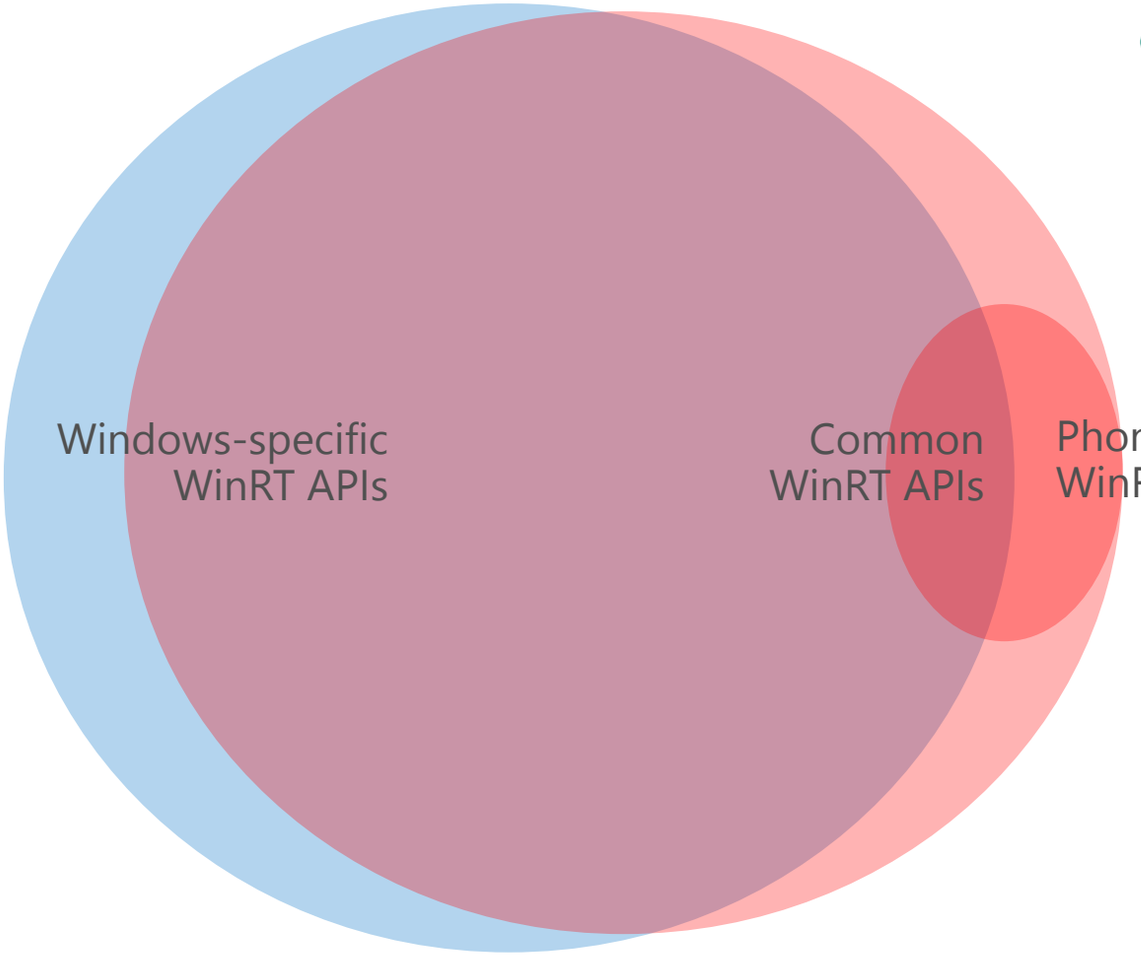
박중석 차장 | 한국마이크로소프트
http://blogs.msdn.com/jspark

목차

유니버셜 윈도우 앱 소개

윈도우 플랫폼간 API 통합

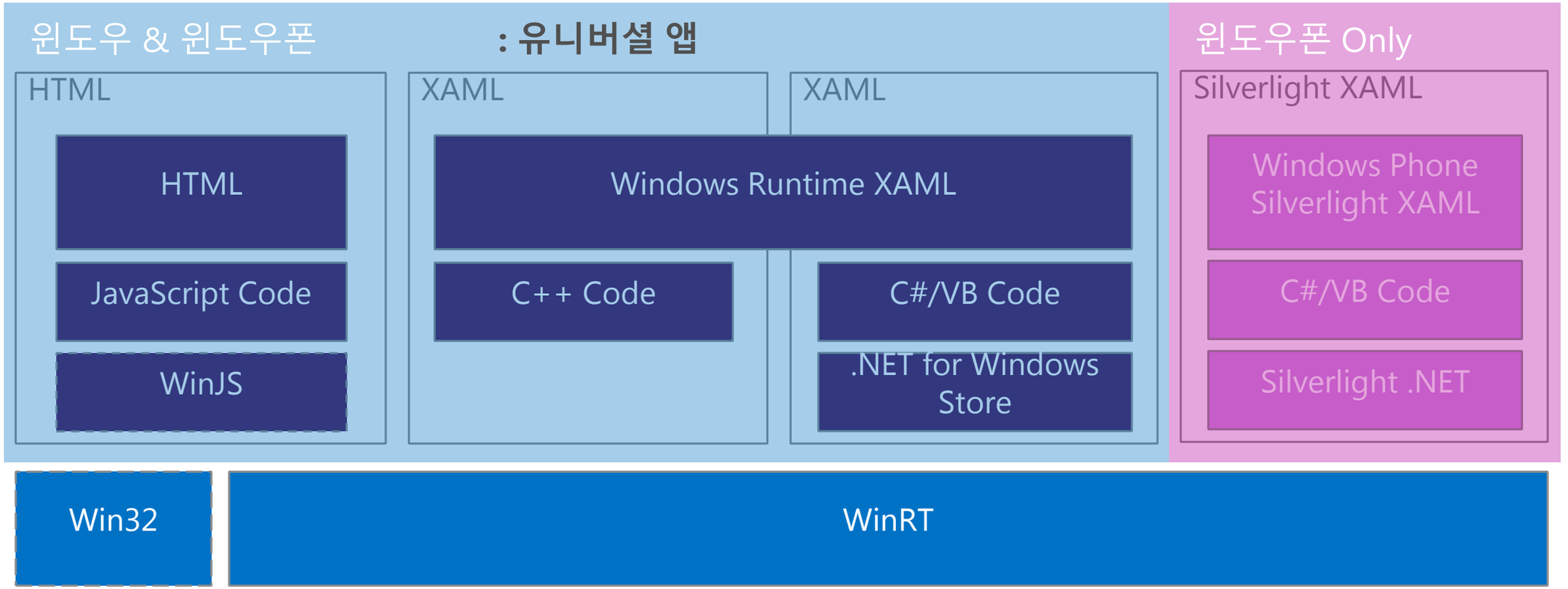
윈도우 런타임(WinRT)?
윈도우 플랫폼(폰과 클라이언트) 스토어 앱에서 사용되는 shared 런타임과 API



8.1 에서 크게 향상됨

- 목표는 100%
- 8.0에서 30% 미만
- 8.1에서 90% 이상

윈도우 & 윈도우폰 8.1 앱


윈도우 & 윈도우폰
: 유니버셜 앱
윈도우폰 Only
HTML
XAML
XAML
Silverlight XAML
HTML
JavaScript Code
WinJS
Windows Runtime XAML
C++ Code
C#/VB Code
.NET for Windows Store
Windows Phone Silverlight XAML
C#/VB Code
Silverlight .NET
Win32
WinRT

유니버셜 윈도우 앱

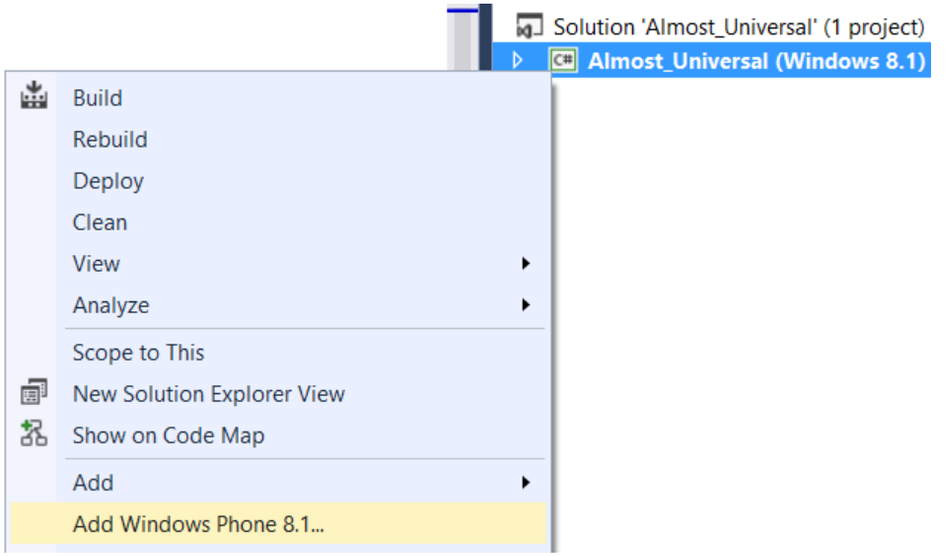
윈도우 8.1과 윈도우폰 8.1 플랫폼에서 구동되는 앱

윈도우 런타임 XAML 지원: C#, VB, C++ 그리고
HTML & Javascript/WinJS

유니버셜 윈도우 앱은 프로젝트 템플릿으로 생성
혹은 기존 윈도우폰 8.1이나 윈도우 8.1앱에서 변환

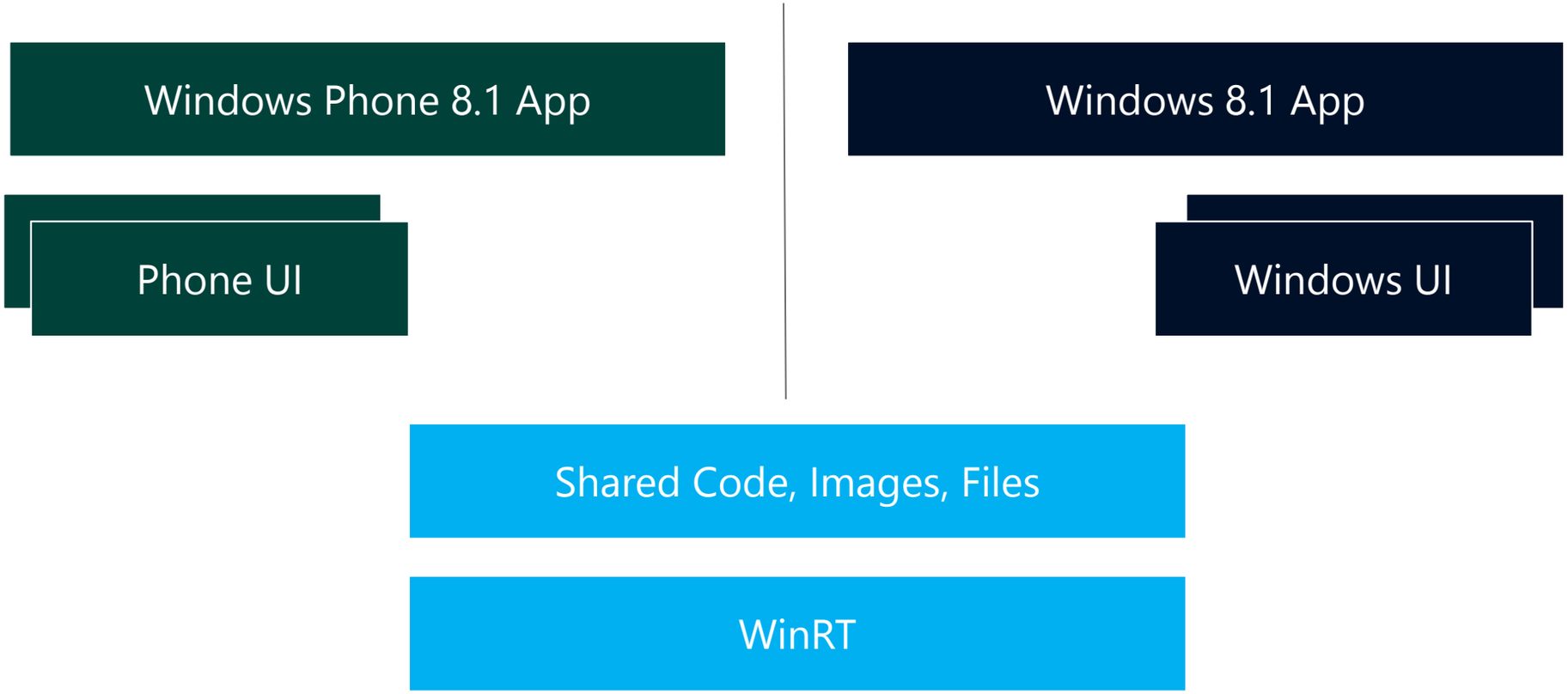
프로젝트를 유니버셜 앱으로 변환

기존 윈도우 8.1 앱에 shared 프로젝트를 포함한 윈도우폰 8.1앱을 추가 (반대도 가능)



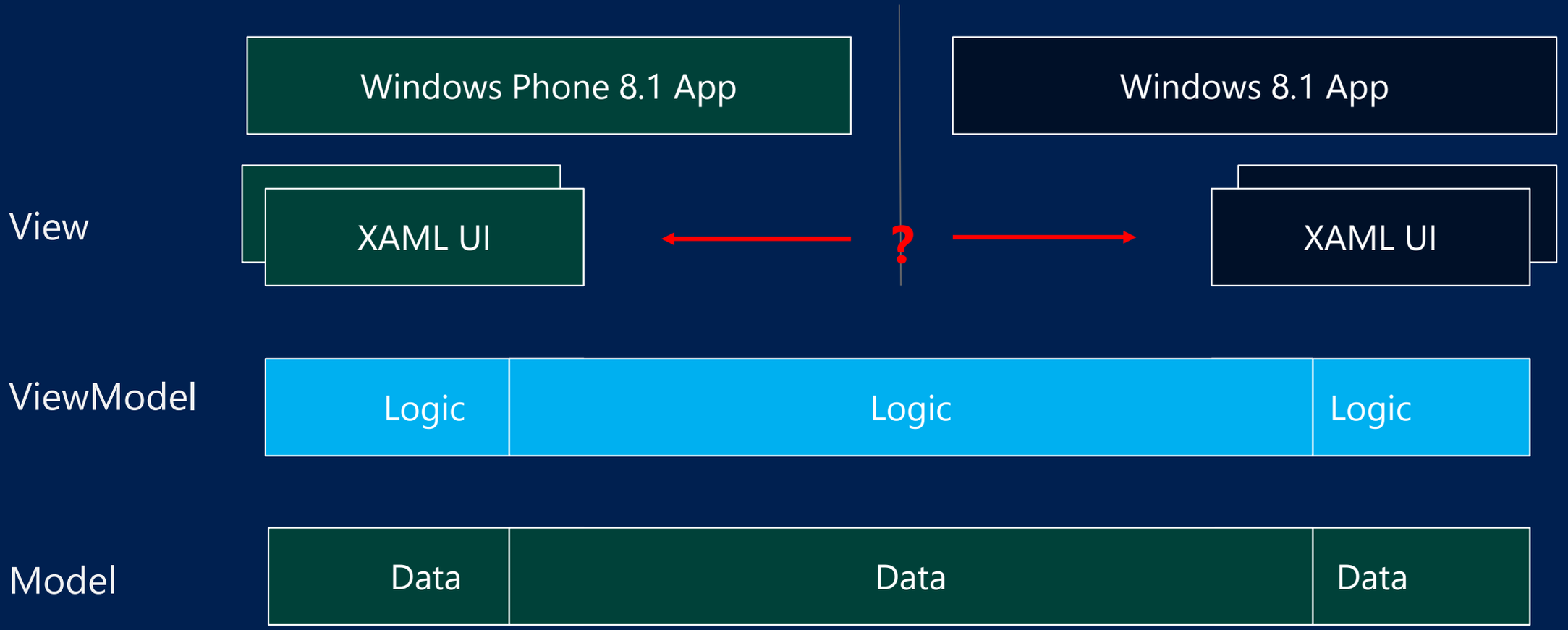
데모:
유니버셜 윈도우 앱 개발

'유니버셜'은 여전히 두개 앱이며, 더 많은 sharing



Sharing 전략

얼마나 많이 sharing 할 수 있을까?



코드 Sharing 방법

Shared 프로젝트 파일

대부분의 파일 타입 가능

Portable 클래스 라이브러리

라이브러리 & 윈도우 런타임 컴포넌트

"링크로 추가"

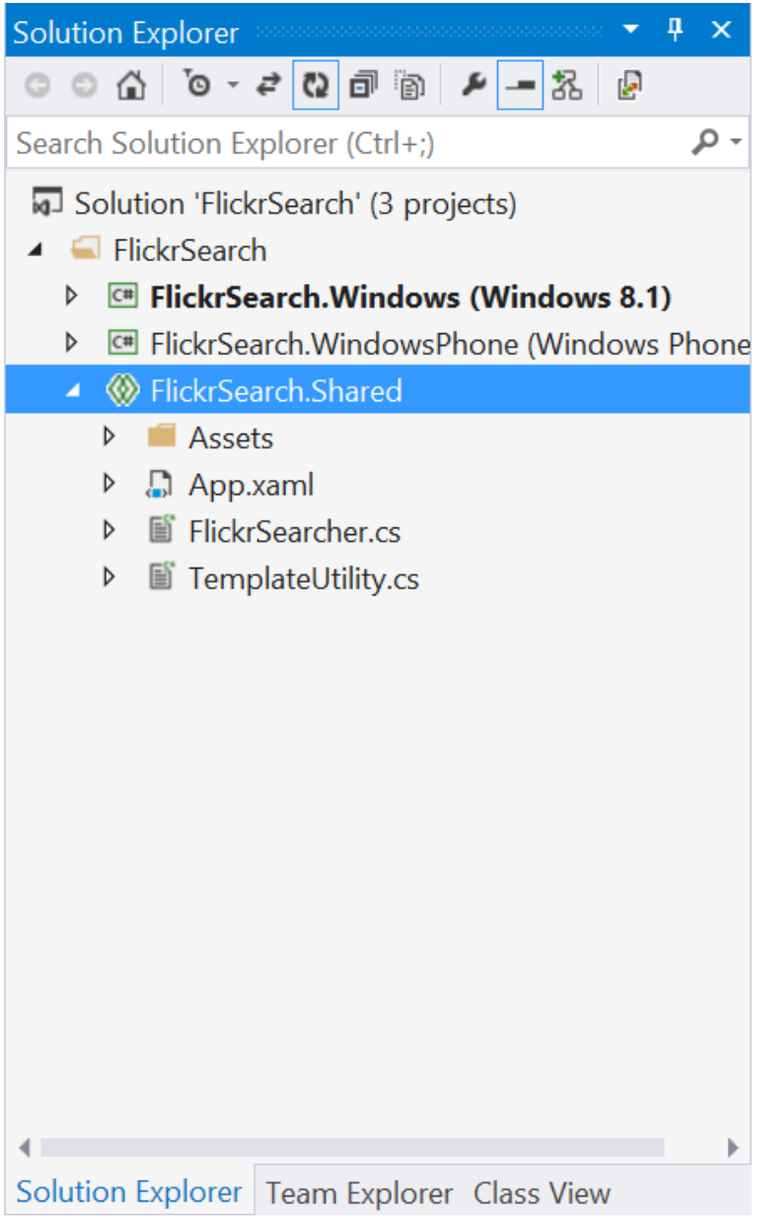
‘Shared’ 프로젝트

앱 간의 소스 sharing 허용

binary output 생성 못함

모든 아이템 지원

- 코드 파일
- XAML
- 이미지
- XML/JSON
- RESW

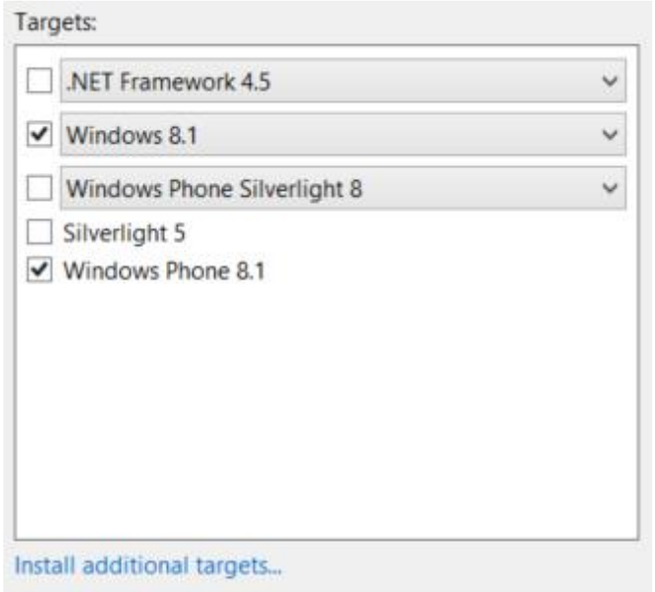


Portable 클래스 라이브러리

tech·days Korea 2014
개발자 컨퍼런스

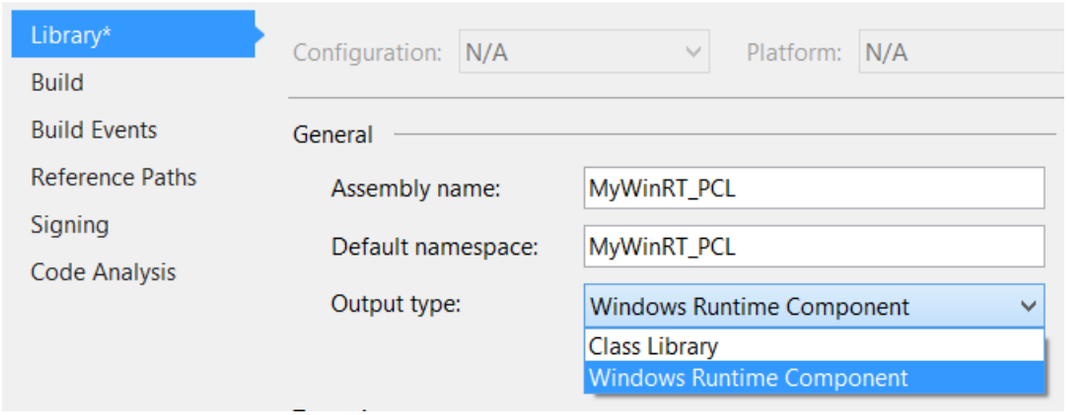
윈도우 유니버셜 앱

WinRT APIs 지원



Output으로 윈도우 런타임 컴포넌트

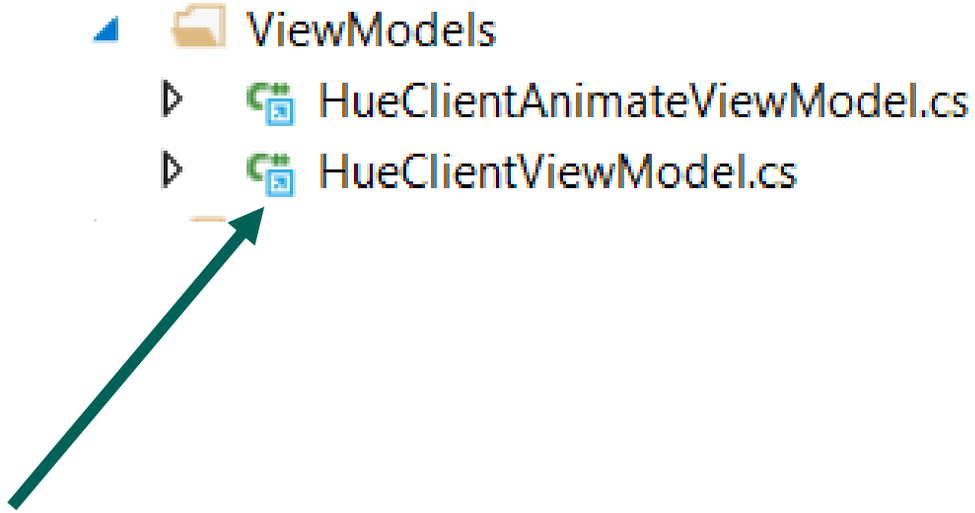
C++, Javascript 앱에서 라이브러리 활용



"링크로 추가"

Shared 프로젝트와 유사

추가 플랫폼 간에 공유 가능



코드 sharing 아키텍쳐

Separation of Concerns

로직으로 부터 UI를 분리

윈도우/윈도우폰 프로젝트의 UI

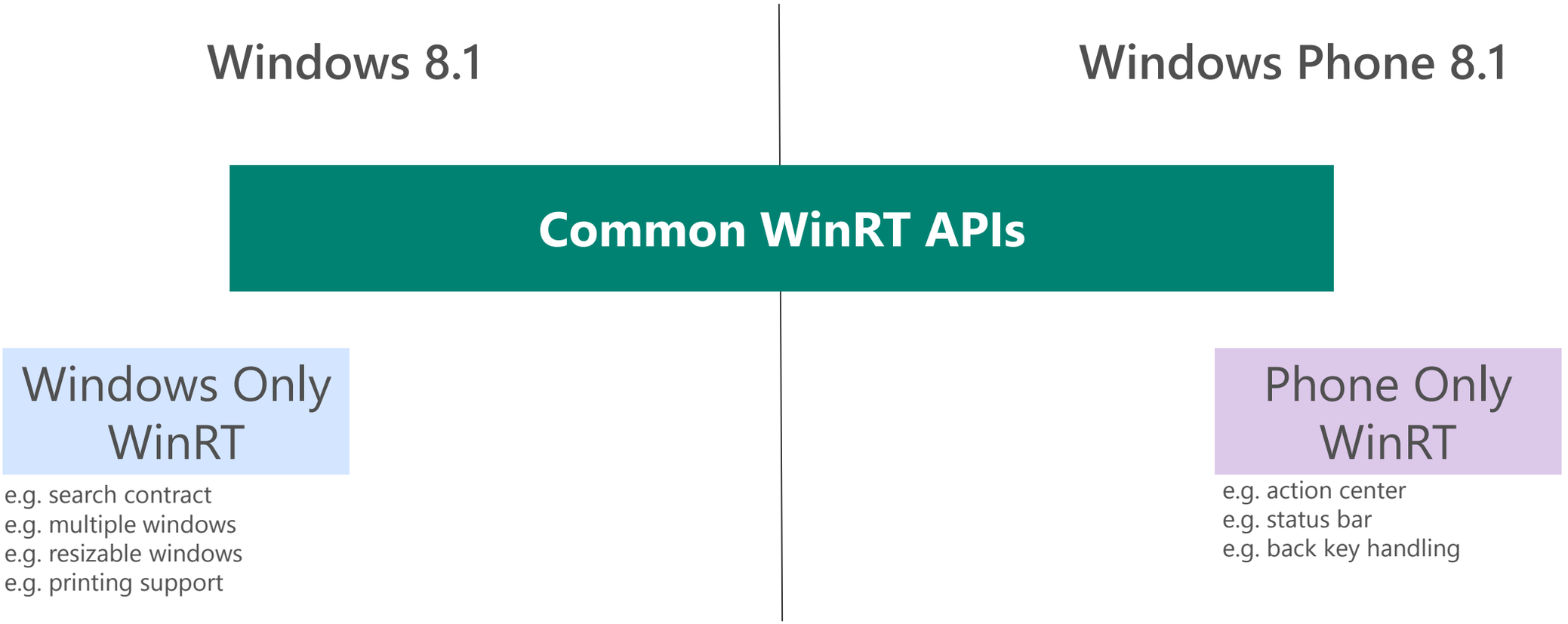
+ 플랫폼 특화된 API 집합 (some geolocation, media, sensors)

Shared 프로젝트의 로직

+ "의미가 있는" XAML 컴포넌트

코드 Sharing

WinRT across Windows+Phone 8.1



몇몇 Common APIs 는 Windows 및 Phone 에서 다르게 동작 할 수 있음

플랫폼 별로 다른 부분 처리 방법

- #if 조건부 컴파일
- 상속
- Partial 클래스

#if 조건부 컴파일

shared 코드에서, 플랫폼 별 차이를 처리하기 위한 코드를 포함하기 위해 상황에 맞게 #if 를 사용

Windows = WINDOWS_APP

Windows Phone = WINDOWS_PHONE_APP

예: 하드웨어 뒤로 가기 버튼은 윈도우폰에만 있음

```
#if WINDOWS_PHONE_APP
            Windows.Phone.UI.Input.HardwareButtons.BackPressed +=
                    this.HardwareButtons_BackPressed;
#endif
```

Sharing 코드: 상속

base 클래스에서 shared 기능 구현

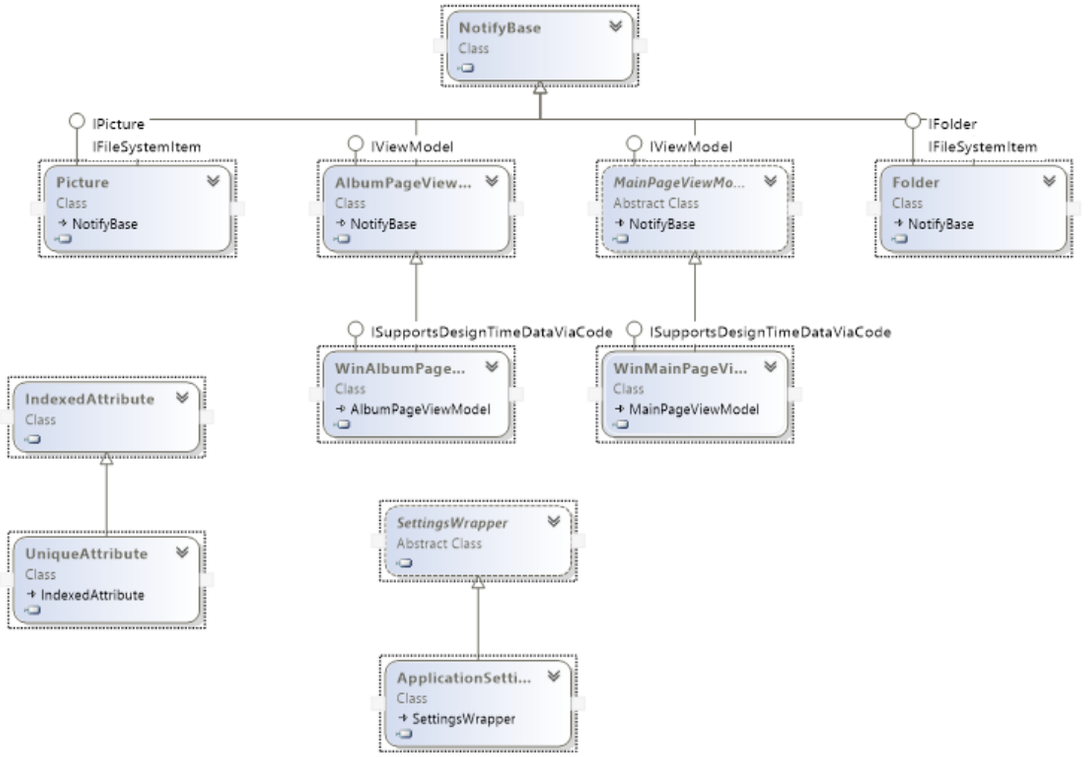
Shared 프로젝트에서

sub 클래스에서 플랫폼 특화 코드 구현

플랫폼 특화된 프로젝트에서

각 플랫폼 별로 특화된 기능을 구분하는데 유용

클래스를 abstract로 만들어서 플랫폼 특화된 구현을 하도록 강제할 수 있음



Sharing 코드: Partial 클래스 & 메소드

하나의 코드 파일에서 Shared 기능 작성

e.g.: DataSource.cs in the shared project

추가 코드 파일에서 플랫폼 특화 코드 작성

e.g.: DataSource.WP.cs in the platform-specific project

클래스는 partial 로 명시되고, 하나의 클래스로 컴파일됨

플랫폼 특화 기능을 분리함

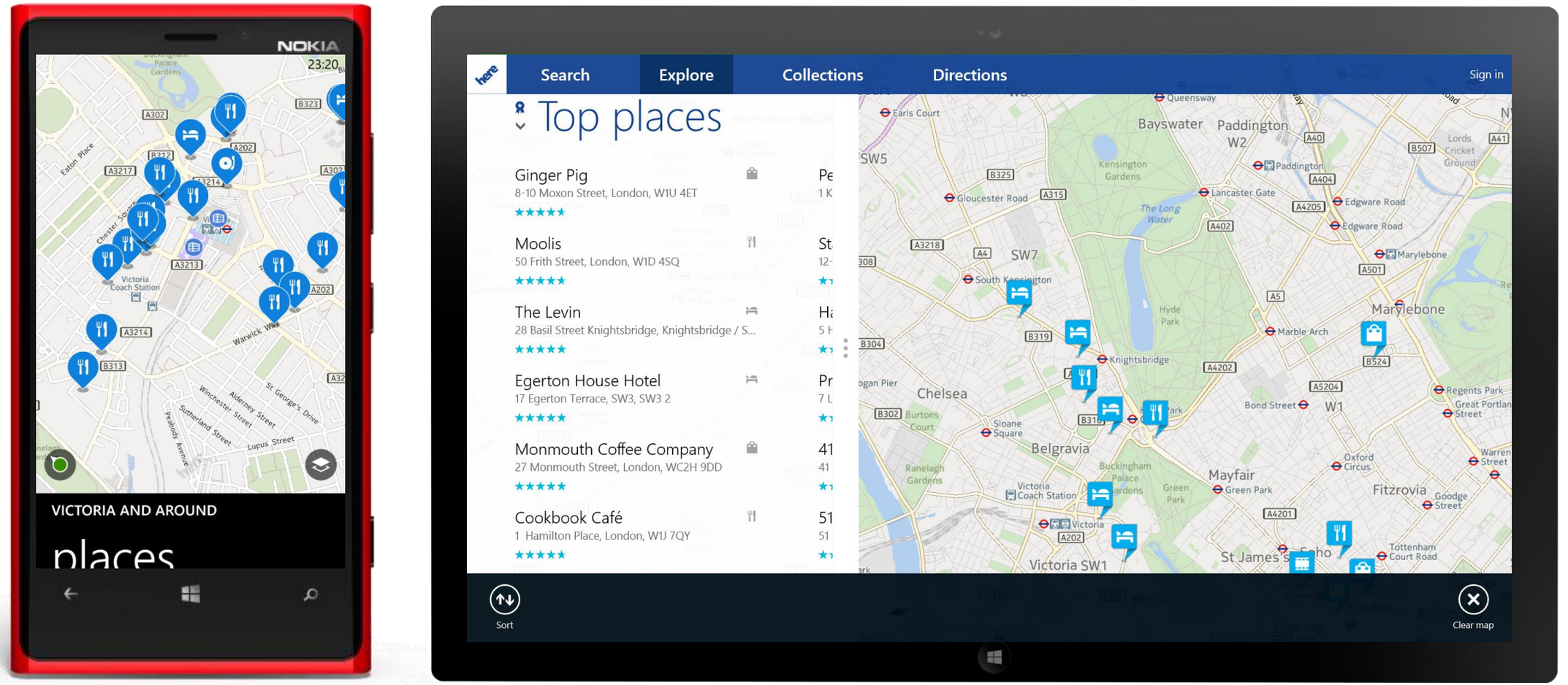
플랫폼 특화 로직을 분리하기 위한 방법으로 partial 메소드를 사용할 수도 있음

Sharing 코드: Partial 클래스 & 메소드

```
/// <summary>
/// DataSource.cs
/// </summary>
public partial class DataSource :IDataSource {
    public async Task<IEnumerable<IFolder>> RetrieveFolders(IFolder root) {
        ... // other logic
        var folders = await LoadFolders(root);
        ... // other logic
        return folders
    }
}

/// <summary>
/// DataSource.WP.cs
/// </summary>
public partial class DataSource {
    private async Task<IEnumerable<IFolder>> LoadFolders(IFolder root) {
        ...
    }
}
```

XAML Sharing

확인: 기기 특성과 사용성을 고려할 것



HERE maps on Windows (8.1)/Phone (8.0)

XAML 컨트롤 on Windows & Phone 8.1

common, same rendering

Button

Slider

ToggleSwitch

ProgressBar

etc (many more)

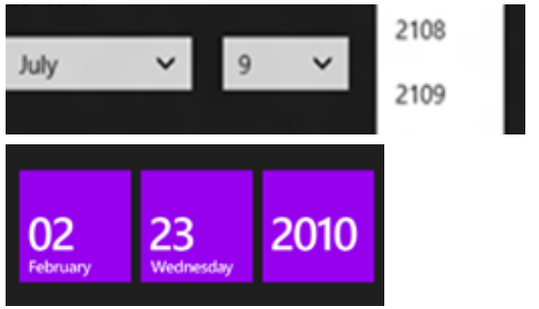
common, different content

Hub

ListView
GridView
etc.

common, different rendering

DatePicker



TimePicker
CommandBar
AppBar
etc.

unique

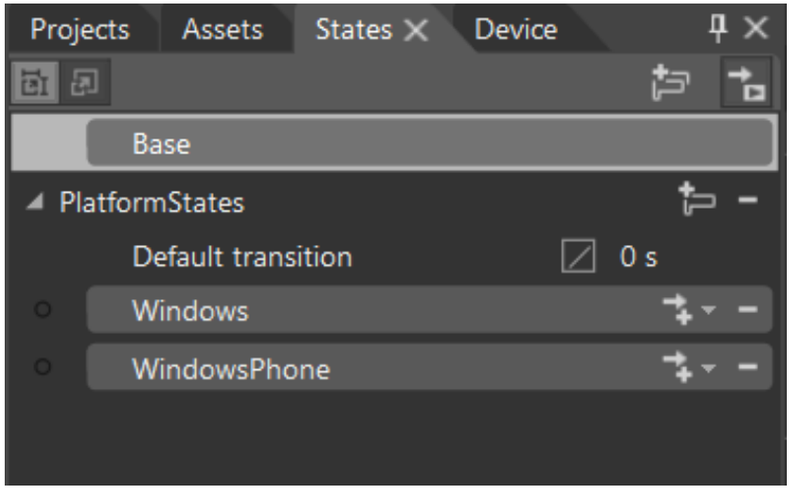
SearchBox

Pivot
ContentDialog
AutoSuggestBox
etc.

Shared XAML 컴포넌트

User Controls

Complete XAML pages



One technique:
Use Visual State Manager to handle layout changes

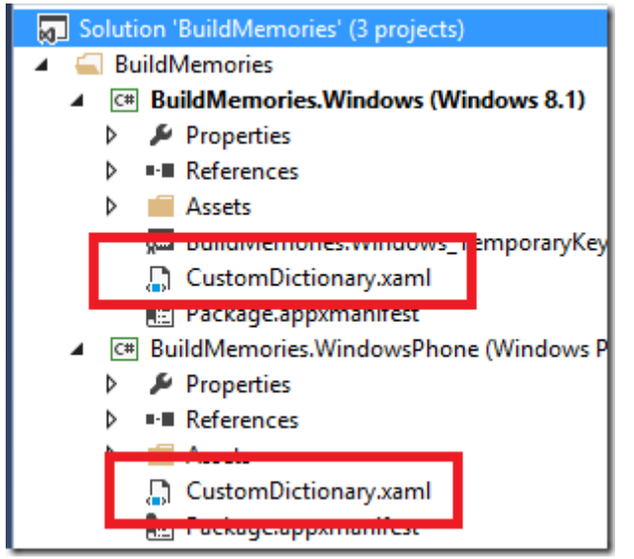
```
#if WINDOWS_APP
            var result = VisualStateManager.GoToState(this, "Windows", false);
#elif WINDOWS_PHONE_APP
            var result = VisualStateManager.GoToState(this, "WindowsPhone", false);
#endif
```

플랫폼 특화 XAML 처리 방법

tech·days Korea 2014
개발자 컨퍼런스

플랫폼 특화된 Resource Dictionaries를 사용

플랫폼 특화된 스타일과 데이터 템플릿을 포함



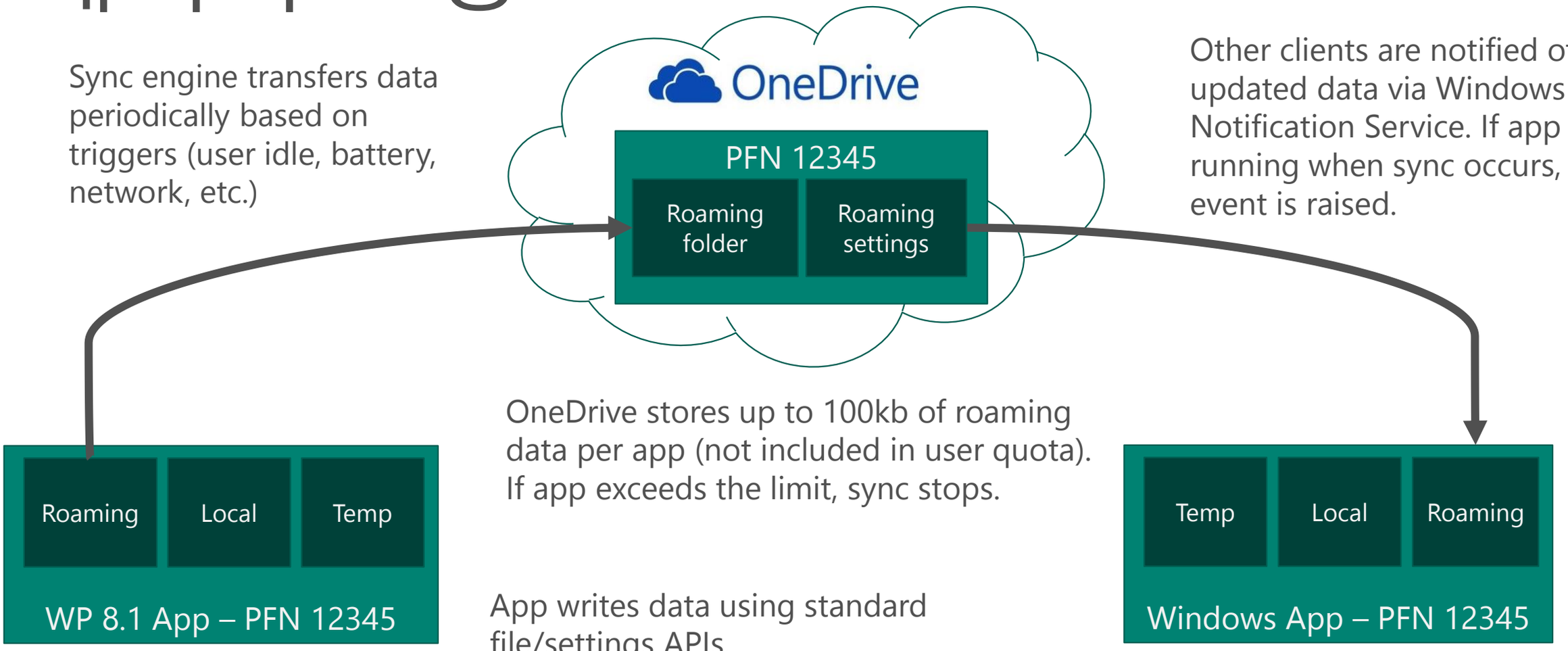
App.xaml에서 dictionaries를 합침:

```
<Application
    x:Class="FlickrSearch.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:FlickrSearch">
    <Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="CustomDictionary.xaml" />
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Application.Resources>
</Application>
```

상태 Sharing
:기기간 데이터 로밍과 OneDrive

데이터 로밍

Sync engine transfers data periodically based on triggers (user idle, battery, network, etc.)



Other clients are notified of updated data via Windows Notification Service. If app is running when sync occurs, an event is raised.

OneDrive stores up to 100kb of roaming data per app (not included in user quota). If app exceeds the limit, sync stops.

App writes data using standard file/settings APIs.

OneDrive 활용

Live SDK 다운로드

Controls and APIs that enable apps to access i nformation from OneDrive

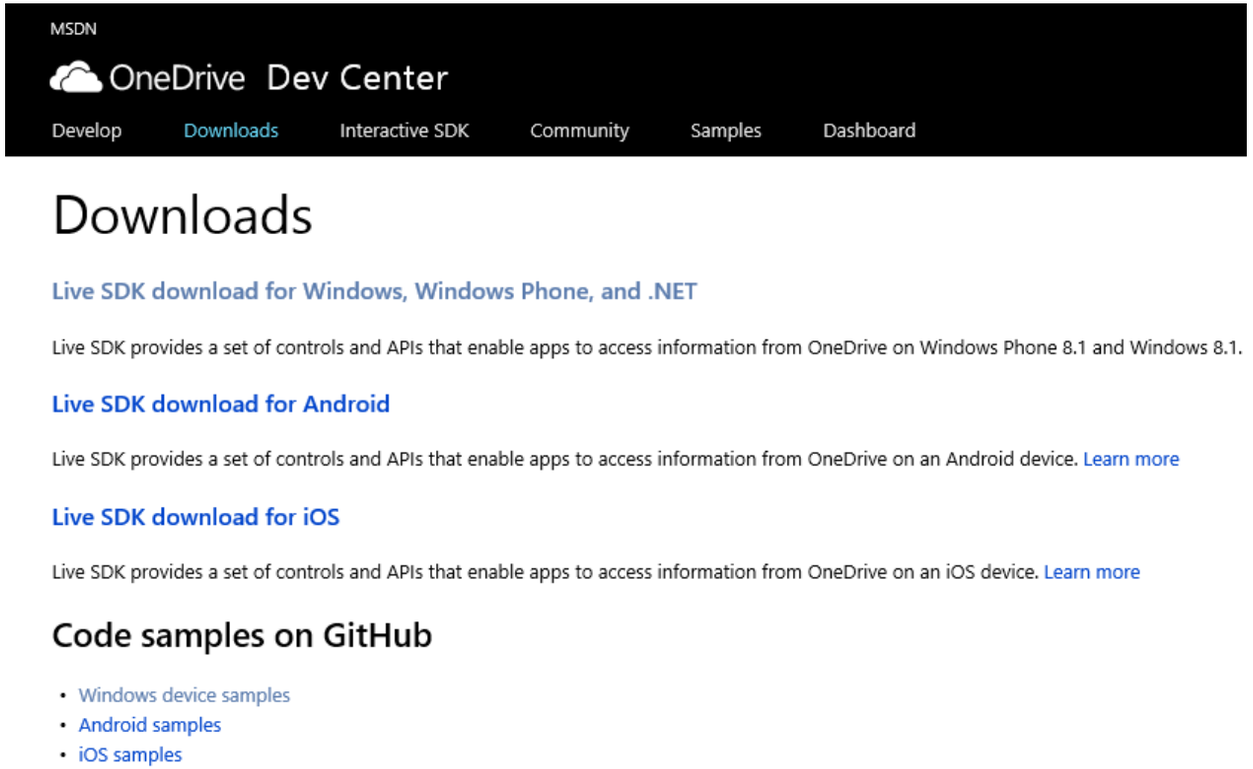
Live SDK 는 사용자의 신원확인 정보에 접속 및 앱에서 활용할 수 있음:

Upload and download photos, videos, docum ents, and other files in OneDrive

Personalize your users' experience

Take advantage of single sign-on using Micros oft Account in Windows 8.1 and Windows Pho ne 8.1

Create, read, and update contacts, calendars, and events in Outlook.com



정리…

유니버셜 앱 소개

Sharing 전략

코드 Sharing

XAML Sharing

상태 Sharing

참고자료

tech·days Korea 2014
개발자 컨퍼런스

Build for Both: Building Shared Apps for Windows Phone and Windows 8.1
http://channel9.msdn.com/Events/TechEd/NorthAmerica/2014/WIN-B363

Building Universal Apps with Visual Studio 2013 Update 2
http://wintellect.com/blogs/jprosise/building-universal-apps-with-visual-studio-2013-update-2

Using Visual Studio to Build XAML Converged Apps
http://channel9.msdn.com/Events/Build/2014/3-591

Developing Apps using the Common XAML UI Framework
http://channel9.msdn.com/Events/Build/2014/2-507

Microsoft

tech·days Korea 2014
개발자 컨퍼런스

Universal design me

유니버셜 윈도우 앱의 새로운 가능성

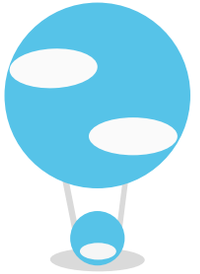
소개

tech·days Korea 2014
개발자 컨퍼런스

이종인

* 바람 인터렉티브 Founder
* Windows Platform Development MVP
* Windows Phone 앱 개발
  - Design Me
  - 대중교통허브

Contact
lunelake@outlook.com
http://www.facebook.com/BaramInteractive



design me

Design Me

나를 디자인 하는 서비스

삶의 목표에 다가가기 쉽게
만들어주기 위한 서비스

Design Me

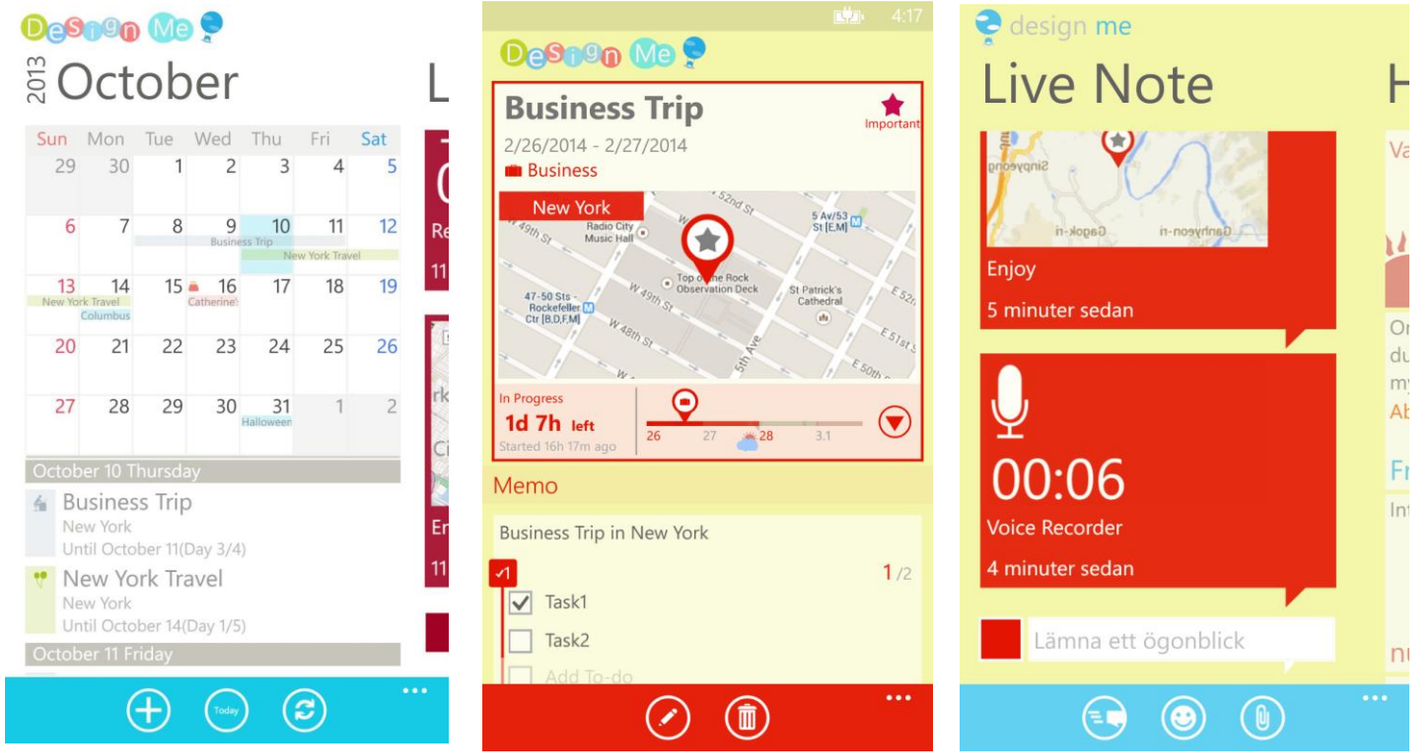
나를 디자인 하는 서비스

삶을 쉽게 계획하고
순간을 기록하기 쉽게

Design Me on Windows Phone

tech·days Korea 2014
개발자 컨퍼런스

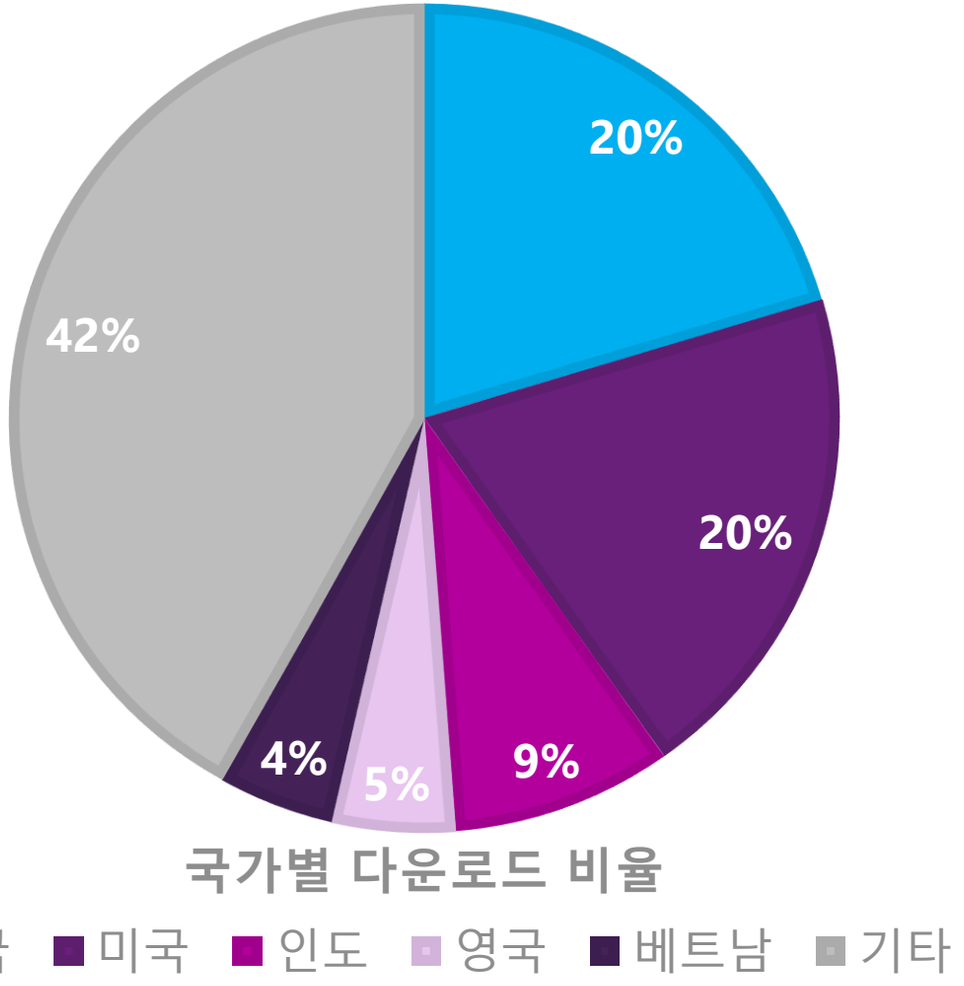
2013년 10월 출시 15개의 언어 지원



Design Me on Windows Phone

tech·days Korea 2014
개발자 컨퍼런스

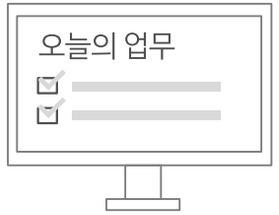
중국, 미국 스토어 유료앱 전체 순위 40위권 기록(2014.3)



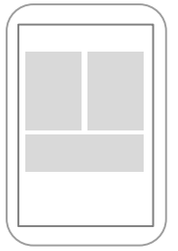
Demo:
design me

Design Me의 유저 시나리오

현재의 일정 및
할 일들 확인



위치 확인 및
특이 사항 메모



7:30am
일어나기

9:00am
업무 시작

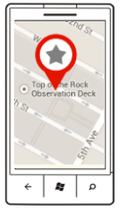
12:00pm
점심 먹기

3:00pm
출장

8:00pm
여가

11:00pm
하루 정리
및 마무리

알람 및
하루 일과 확인



밥먹는 곳
찾기

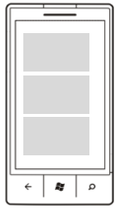
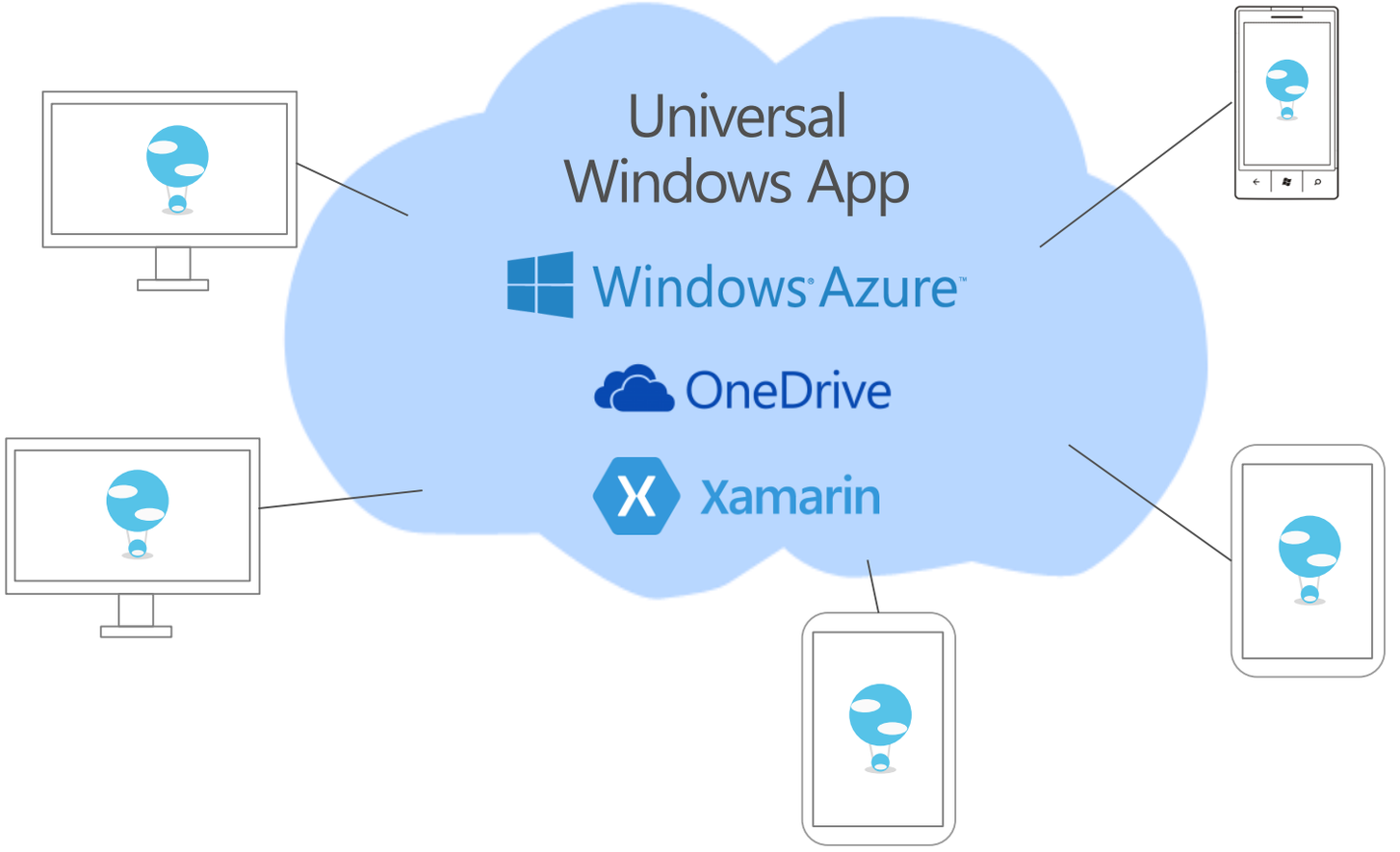


사진 기록하고
있었던 일 남기기

Design Me의 개발 목표

tech·days Korea 2014
개발자 컨퍼런스

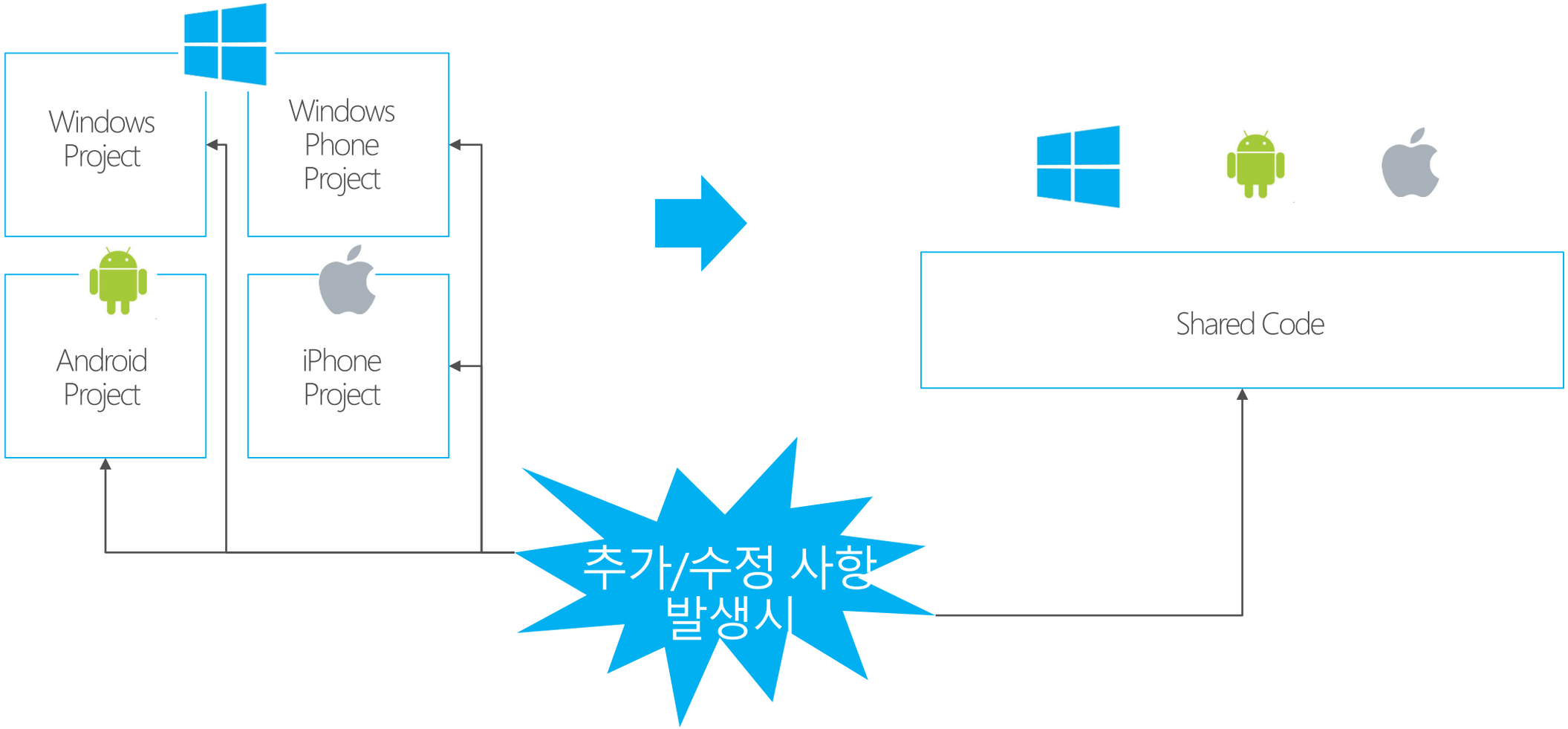
다양한 디바이스와 환경 속에서
유연하게 이어지는 서비스를 만드는 것



하나처럼 이어진 서비스를 위해서

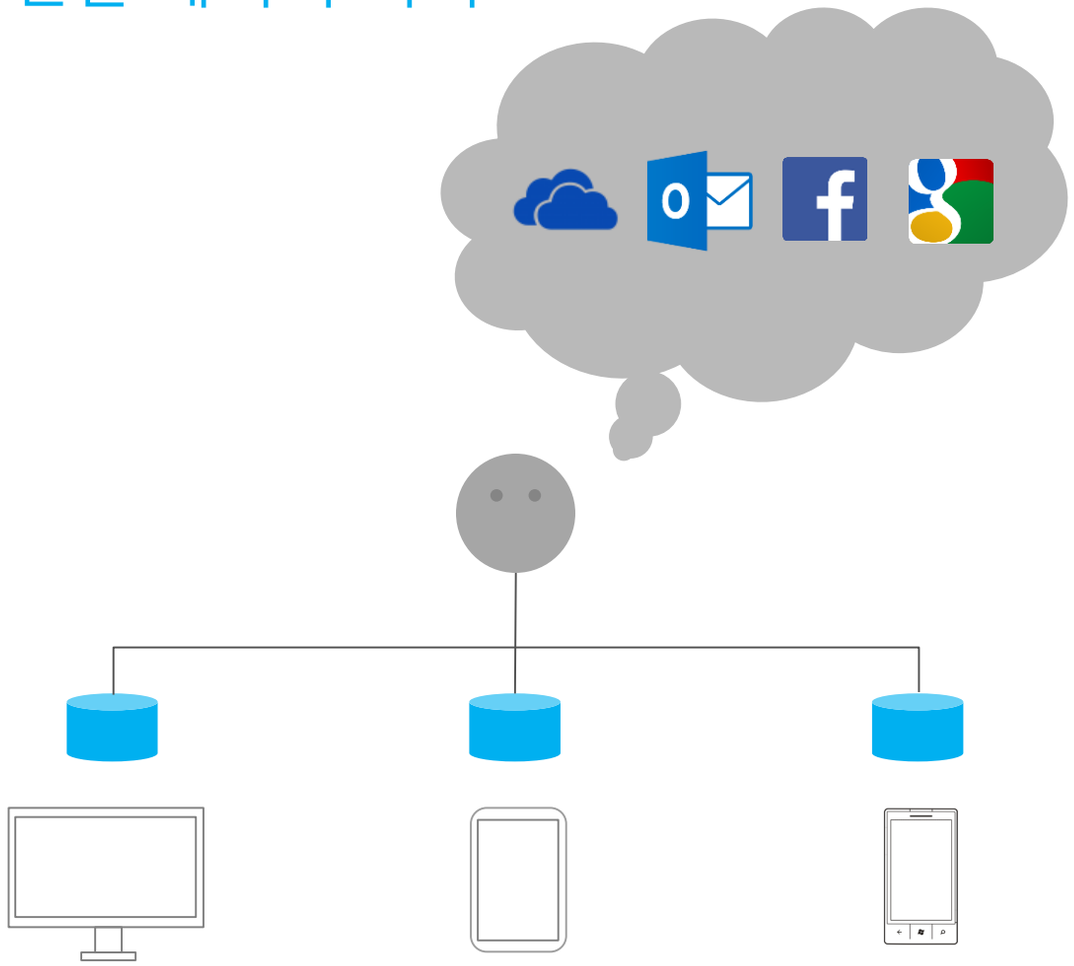
고려해야 할 사항

Cross platform – 코드 공유



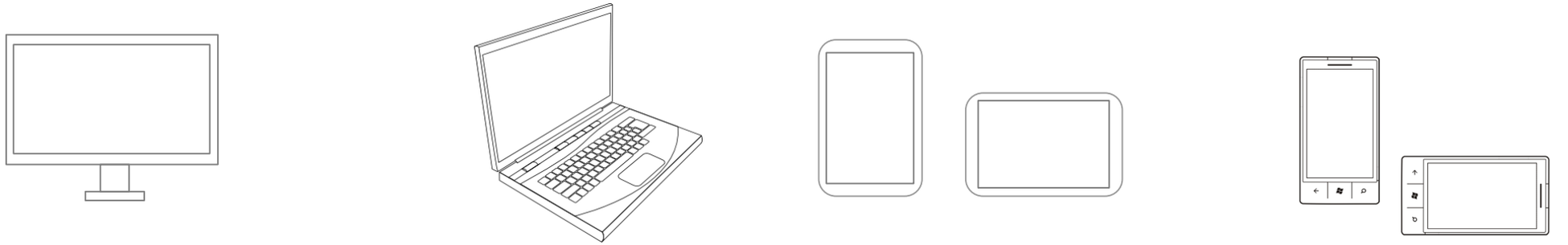
고려해야 할 사항

멀티 디바이스 환경에서 유연한 데이터 처리



고려해야 할 사항

디바이스의 크기와 환경에 따라 달라지는 유저들의 요구와
그로 인해 파편화된 경험들 합치기



생산성

기능

편집

이동성

단순함

보기

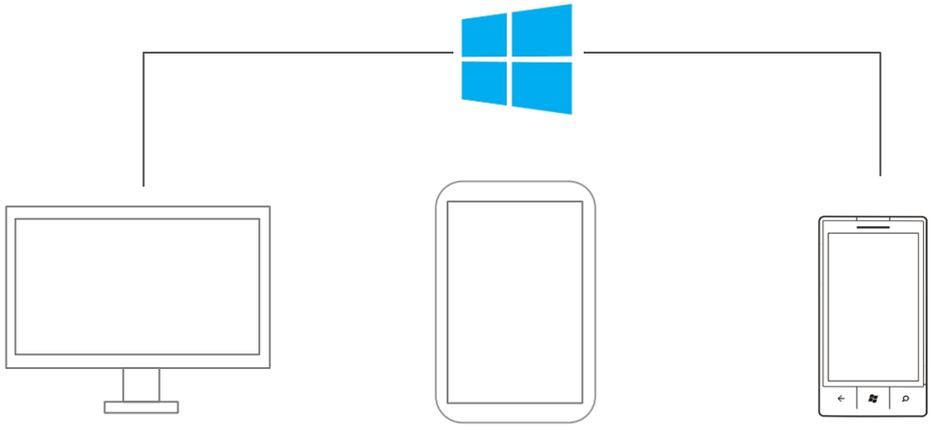
Universal Windows App

윈도우와 윈도우폰 개발을 한번에
그리고 Cross platform

Universal Windows App

tech·days Korea 2014
개발자 컨퍼런스

데스크탑, 태블릿, 폰을 아우르는 서비스를 한 프로젝트에서



Xamarin

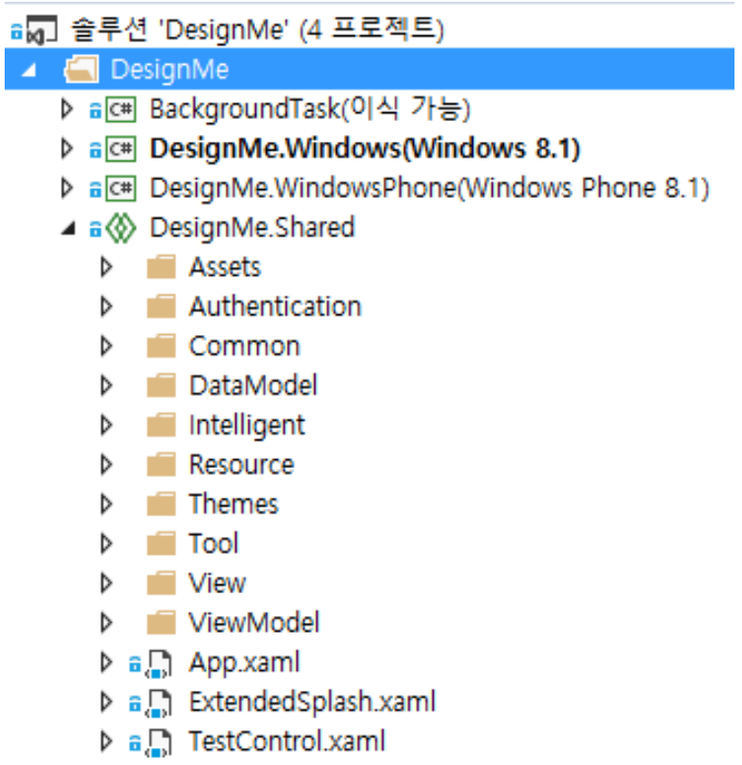
Universal
Windows App

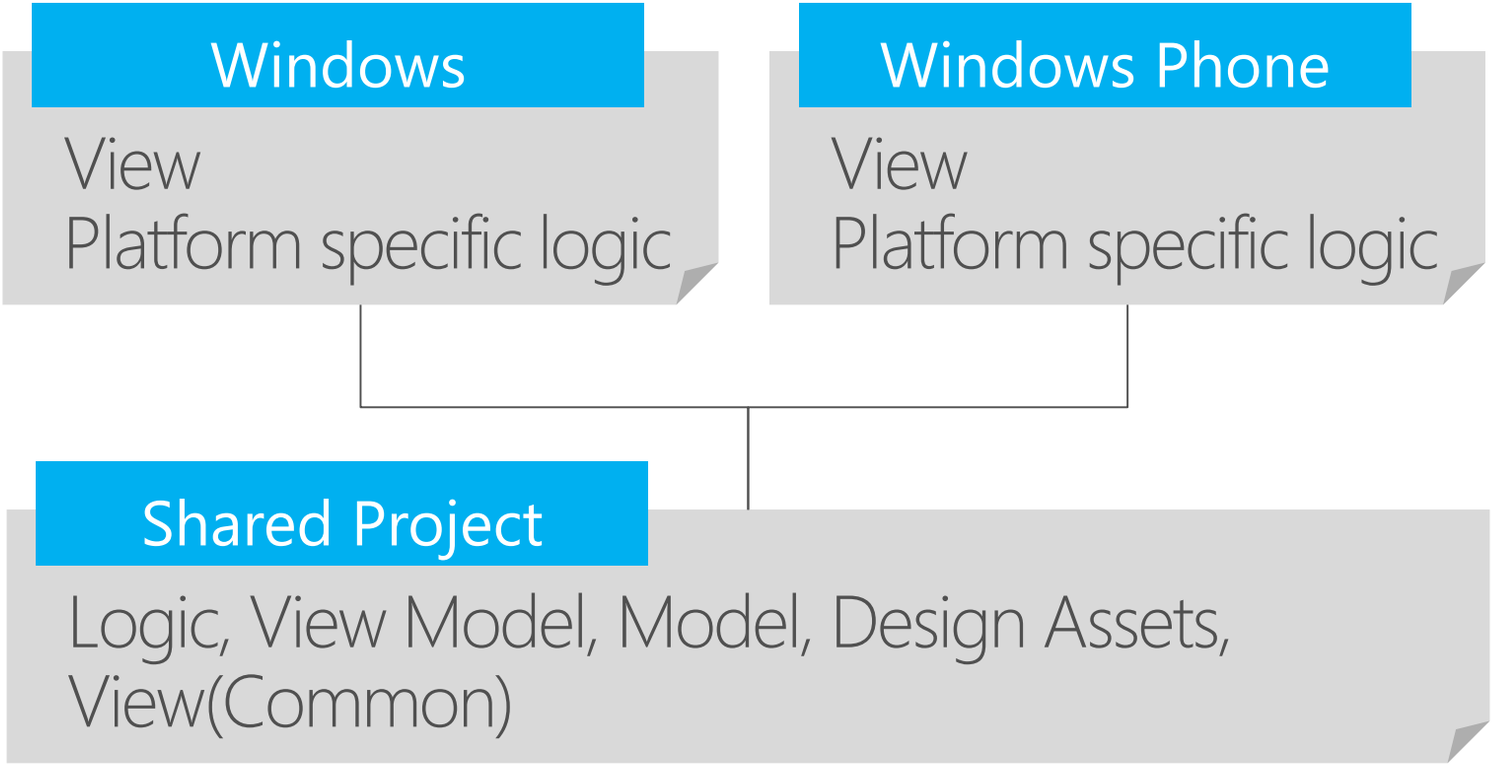
(Shared project, PCL)

Universal Windows App

tech·days Korea 2014 개발자 컨퍼런스

Shared Project

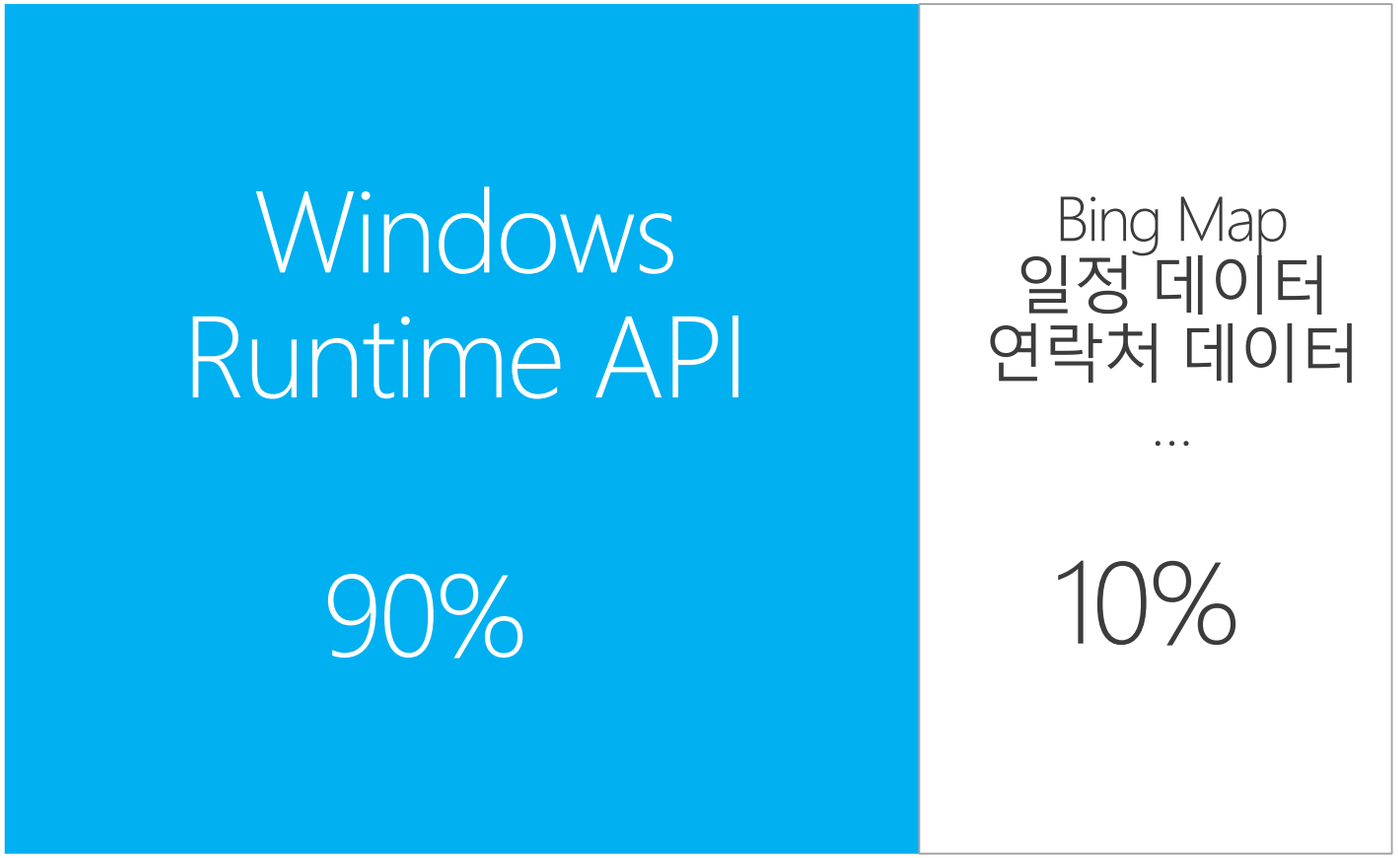




Universal Windows App

tech·days Korea 2014
개발자 컨퍼런스

통합이 안된 API들



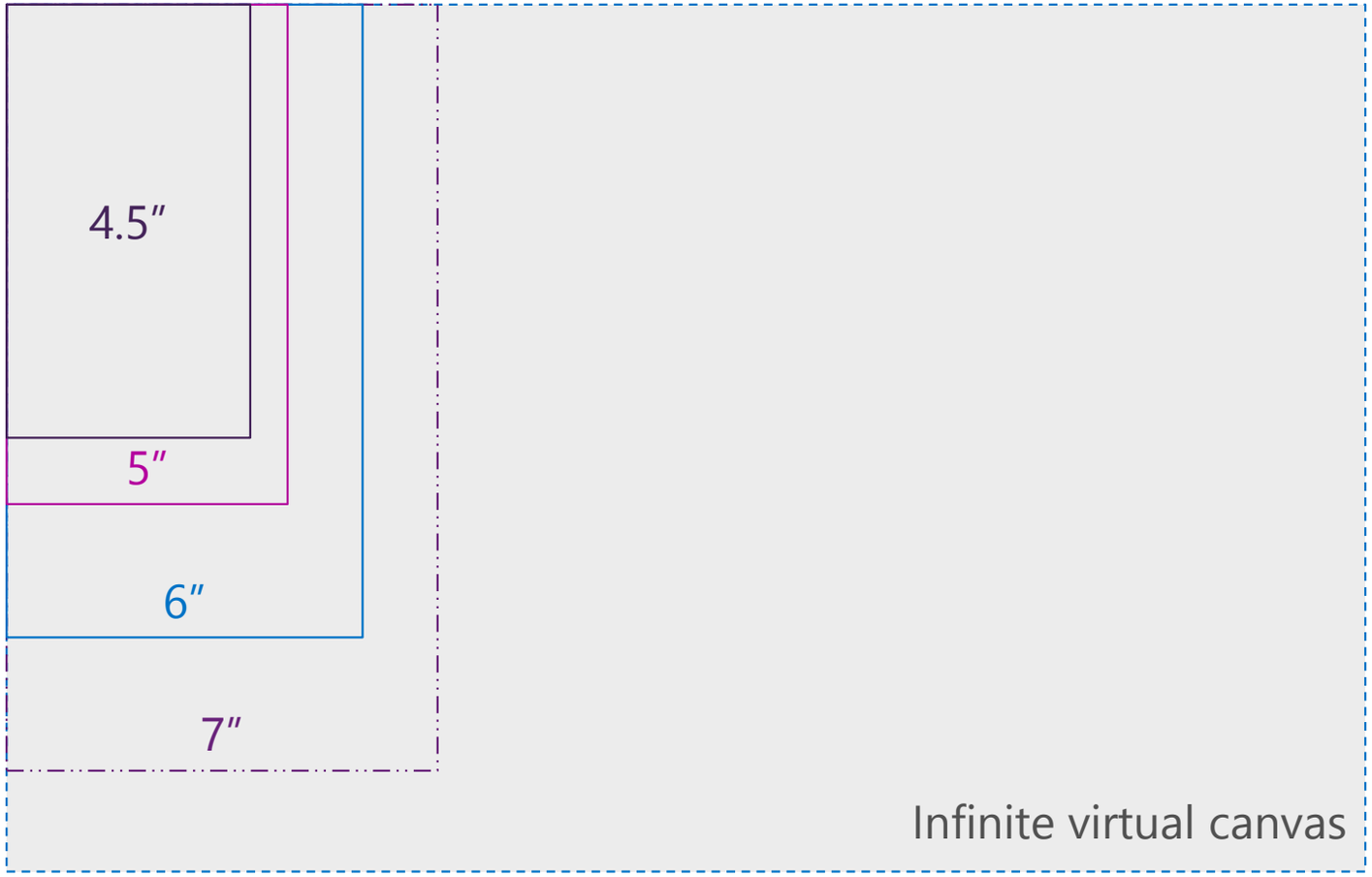
조건부 컴파일 + 별도 처리 시나리오

```
public void ClearMap()
{

        #if WINDOWS_APP
        _shapeLayer.Shapes.Clear();
         _pinLayer.Children.Clear();

        #elif WINDOWS_PHONE_APP
        _map.MapElements.Clear();
        _map.Children.Clear();
        #endif

}
```

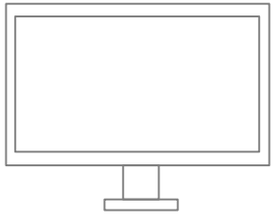
Universal Windows App

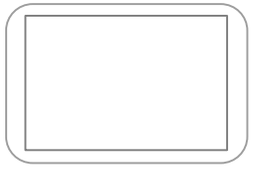
tech·days Korea 2014
개발자 컨퍼런스

화면 크기에 대응하는 UI

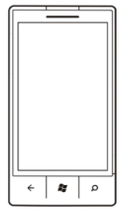


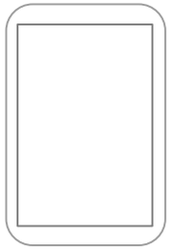
Landscape





Portrait





Universal Windows App

tech·days Korea 2014
개발자 컨퍼런스

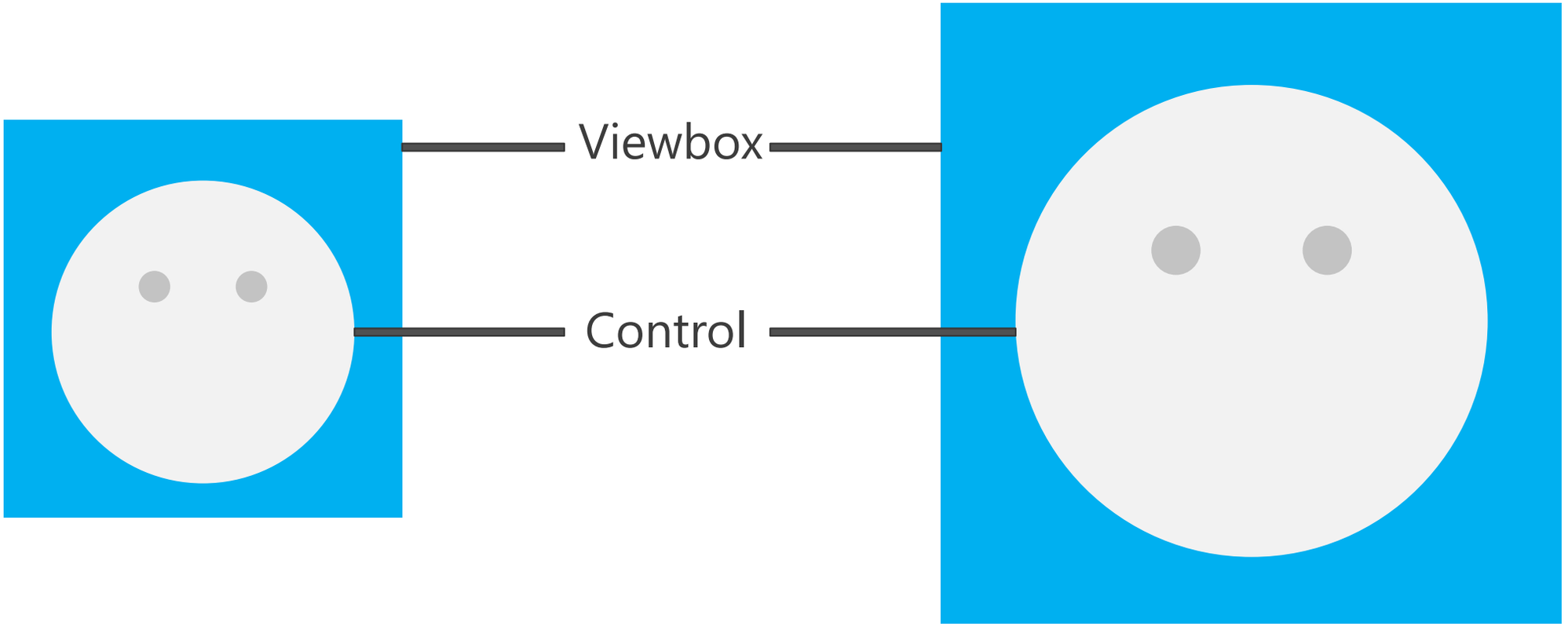
윈도우 사이즈가 변했을 때

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
        this.navigationHelper.OnNavigatedTo(e);
         Window.Current.SizeChanged += Current_SizeChanged;
}

protected override void OnNavigatedFrom(NavigationEventArgs e)
{
         this.navigationHelper.OnNavigatedFrom(e);
        Window.Current.SizeChanged -= Current_SizeChanged;
}

/// <summary>
/// 윈도우의 사이즈가 변했을 때 이벤트
/// </summary>
public void Current_SizeChanged(object sender, Windows.UI.Core.WindowSizeChangedEventArgs e)
{
         double windowWidth = e.Size.Width;
         double windowHeight = e.Size.Height;

          //로직
}
```

Universal Windows App

tech·days Korea 2014
개발자 컨퍼런스

Viewbox

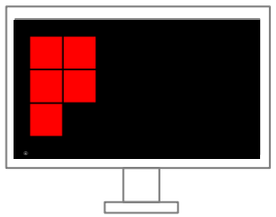


Universal Windows App

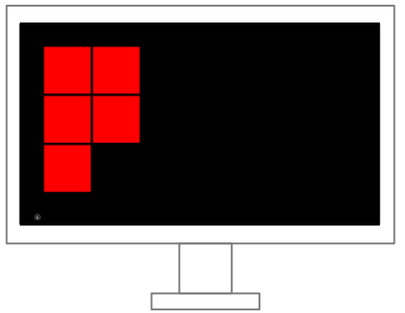
tech·days Korea 2014
개발자 컨퍼런스

Viewbox

Element 비율 따라 크기 조정



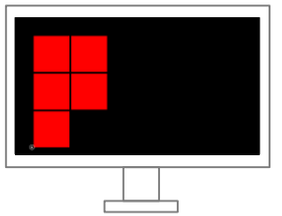
1366x768



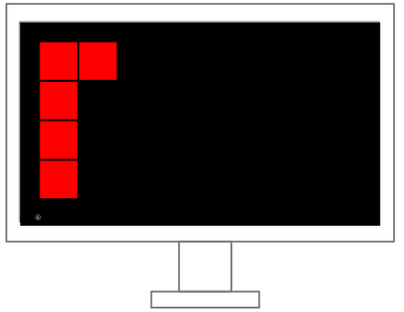
1920x1080

```
<Grid Background="Black">
    <Viewbox HorizontalAlignment="Left">
        <GridView Margin="100,100,0,0" Height="768" >
            <Rectangle Fill="Red" Height="200" Width="200" />
            <Rectangle Fill="Red" Height="200" Width="200" />
            <Rectangle Fill="Red" Height="200" Width="200" />
            <Rectangle Fill="Red" Height="200" Width="200" />
            <Rectangle Fill="Red" Height="200" Width="200" />
        </GridView>
    </Viewbox>
</Grid>
```

Element 크기 조정



1366x768



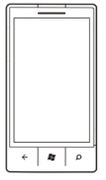
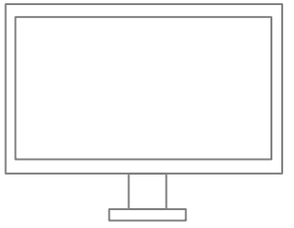
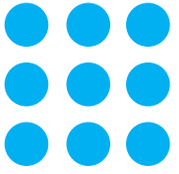
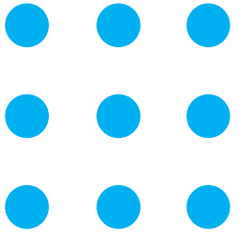
1920x1080

```
<Grid Background="Black">
    <GridView Margin="100,100,0,0">
        <Rectangle Fill="Red" Height="200" Width="200" />
        <Rectangle Fill="Red" Height="200" Width="200" />
        <Rectangle Fill="Red" Height="200" Width="200" />
        <Rectangle Fill="Red" Height="200" Width="200" />
        <Rectangle Fill="Red" Height="200" Width="200" />
    </GridView>
</Grid>
```

Universal Windows App

tech·days Korea 2014
개발자 컨퍼런스

스케일 이슈

	Windows Phone		Windows
			
DPI		>	
Font	글씨		글씨

플랫폼별 스타일 지정

```
<Style x:Key="TextblockStyle" TargetType="TextBlock">
    <Setter Property=" FontSize" Value=“17”></Setter>
</Style>
```

Resource dictionary

Windows Phone

Resource dictionary

Windows

Viewbox

Viewbox의 크기를 플랫폼별로 다르게 지정

Demo:
Universal design me

Less code, Share more!

Data binding, MVVM

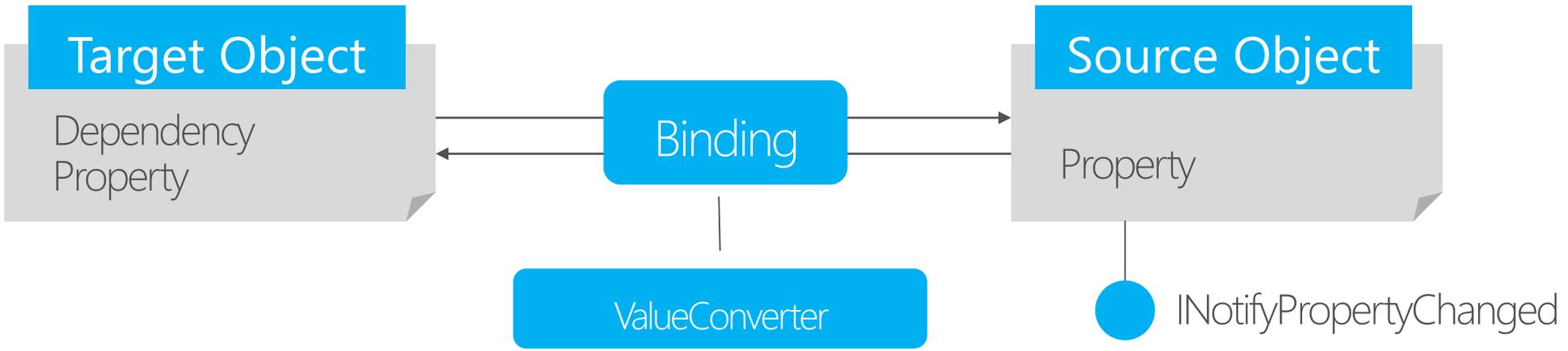
Data binding


```
<TextBlock Margin="-1,0,0,0" FontSize="40"
Foreground= “#FF444444”  Text= “Pages”  />
```

Data binding

```
<TextBlock Margin="-1,0,0,0" FontSize="40"
Foreground="{Binding TitleForegroundColor}"
Text="{Binding PagesTitle}" />
```

결과 Pages



Data binding

tech·days Korea 2014
개발자 컨퍼런스

복잡한 뷰 로직의 간결한 처리

Home

Textblock1
Textblock2
Textblock3
Textblock4

bool WhiteMode = true

Home

Textblock1
Textblock2
Textblock3
Textblock4

bool WhiteMode = false

기본 방식

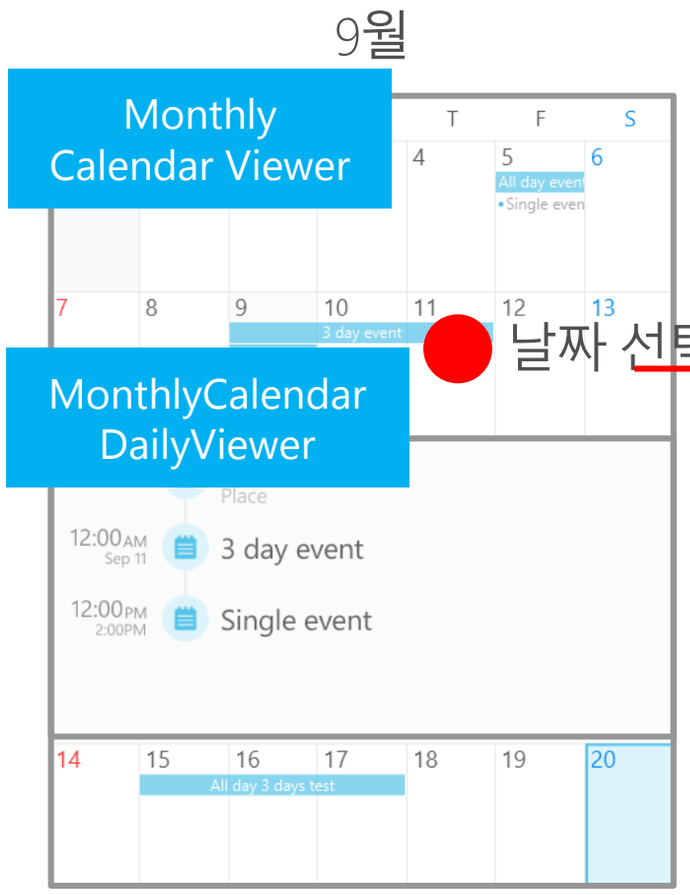
```
TitleTextblock.Foreground = new SolidColorBrush(Colors.White);
Textblock1.Foreground = new SolidColorBrush(Colors.White);
Textblock2.Foreground = new SolidColorBrush(Colors.White);
Textblock3.Foreground = new SolidColorBrush(Colors.White);
Textblock4.Foreground = new SolidColorBrush(Colors.White);
```

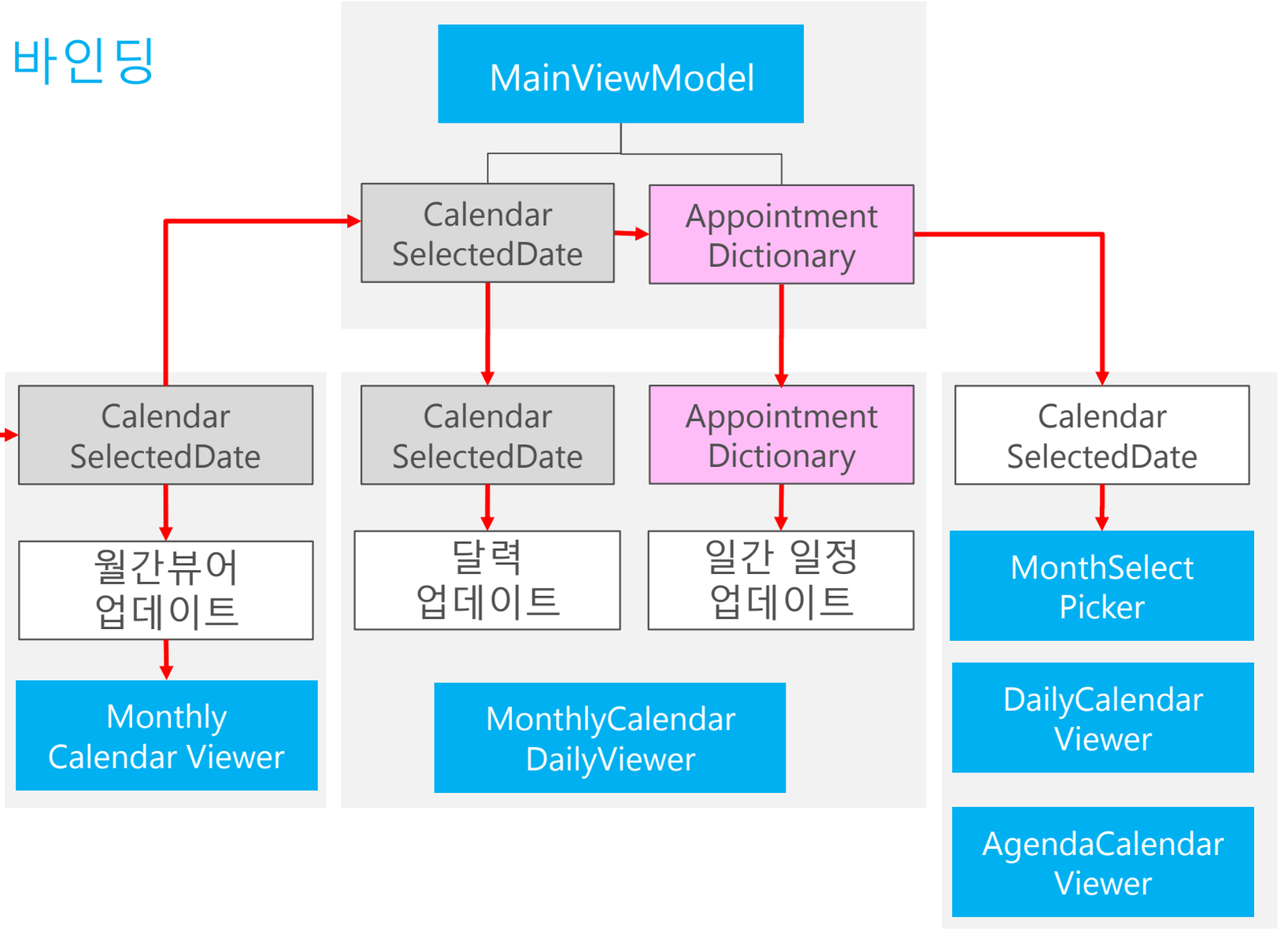
Data binding

```
public bool WhiteMode
{
   get{return _WhiteMode;}
   set
   {
      _WhiteMode = value;
      OnPropertyChanged();

      if (value == true)
        TitleForegroundColor = "#FFFFFFFF";
      else
        TitleForegroundColor = "#FF444444";
   }
}
```

Data binding

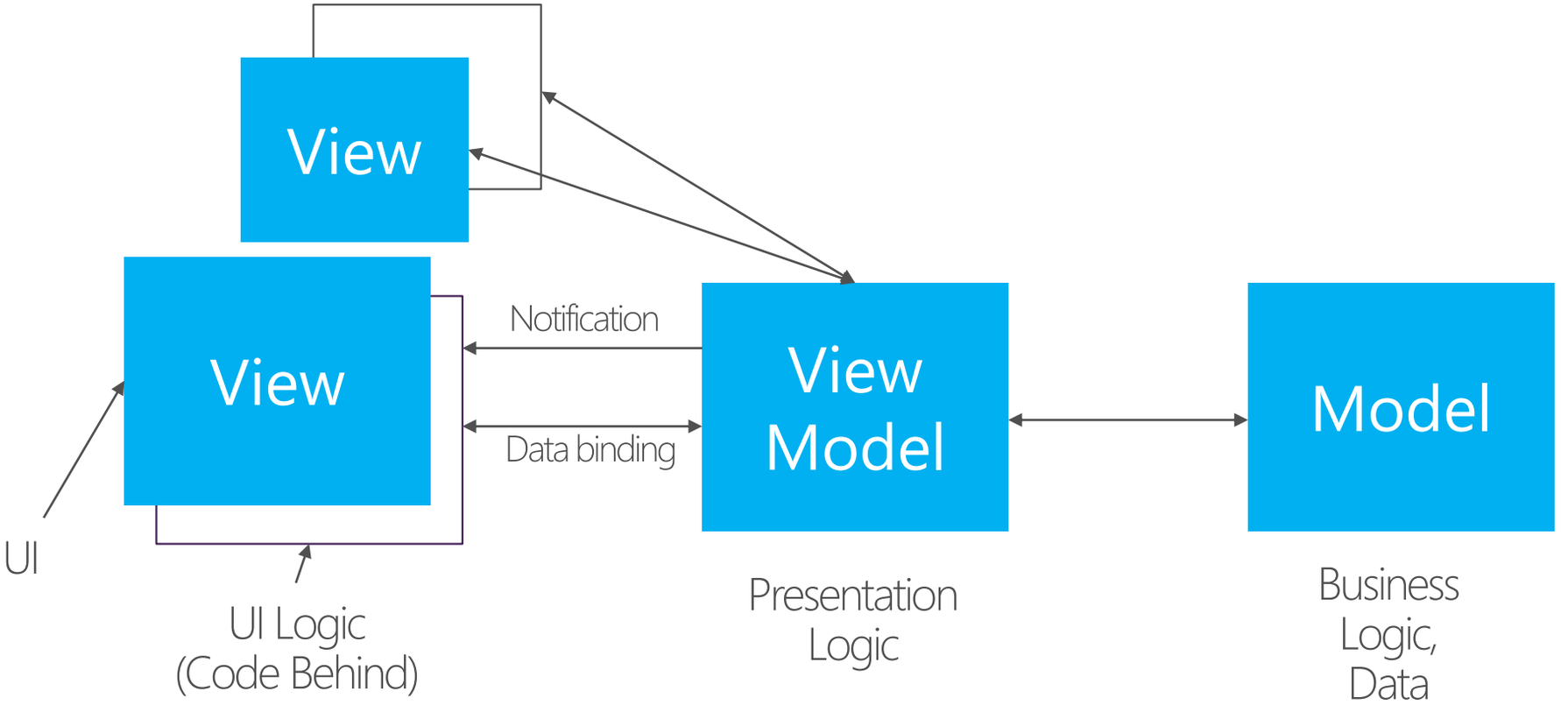
굴비 엮듯이 엮어지는 데이터 바인딩





Data binding

MVVM



Data binding

INotifyPropertyChanged

```
public class ViewModel : INotifyPropertyChanged
{
   private string _ID;
   public string ID
   {
      get{return _ID;}
      set{_ID = value;
           OnPropertyChanged();}
    }

   public event PropertyChangedEventHandler PropertyChanged;

   protected bool SetProperty<T>(ref T storage, T value, [System.Runtime.CompilerServices.CallerMemberName] String propertyName = null)
   {
        if (object.Equals(storage, value)) return false;

         storage = value;
         this.OnPropertyChanged(propertyName);
         return true;
   }

   protected void OnPropertyChanged([System.Runtime.CompilerServices.CallerMemberName] string propertyName = null)
   {
       if (PropertyChanged != null)
       {
           PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
       }
   }
}
```

Data binding

Control에 DependencyProperty 추가하기


```
public static readonly DependencyProperty CalendarSelectedDateProperty =
DependencyProperty.Register("CalendarSelectedDate", typeof(DateTime), typeof(MonthlyCalendarViewer), new PropertyMetadata(null, new
PropertyChangedCallback(OnCalendarSelectedDateChanged)));

        public DateTime CalendarSelectedDate
    {
        get { return (DateTime)GetValue(CalendarSelectedDateProperty); }
        set { base.SetValue(CalendarSelectedDateProperty, value); }
    }

    private static void OnCalendarSelectedDateChanged(DependencyObject d, DependencyPropertyChangedEventArgs e)
    {
        ((MonthlyCalendarViewer)d).OnCalendarSelectedDateChanged(e);
    }

    void OnCalendarSelectedDateChanged(DependencyPropertyChangedEventArgs e)
    {
        if (e.OldValue != null)
        {
            DateTime oldValue = (DateTime)e.OldValue;
            DateTime newValue = (DateTime)e.NewValue;
            if (oldValue.Month != newValue.Month || oldValue.Year != newValue.Year)
                DrawCalendarBase();
        }
        else
        {
            DrawCalendarBase();
            DrawCalendarContent();
            loaded = true;
        }
    }
```

```
        참조 16개
        public string ID
        {
            get
            {
                return _ID;
            }
            set
            {
                _ID = value;
                OnPropertyChanged();
            }
        }

        private string _CategoryColor;
        참조 11개
        public string CategoryColor
        {
            get
            {
                return _CategoryColor;
            }
            set
            {
                _CategoryColor = value;
                OnPropertyChanged();
            }
        }

        private string _CategoryName;
        참조 8개
        public string CategoryName
```

Demo:
Data binding

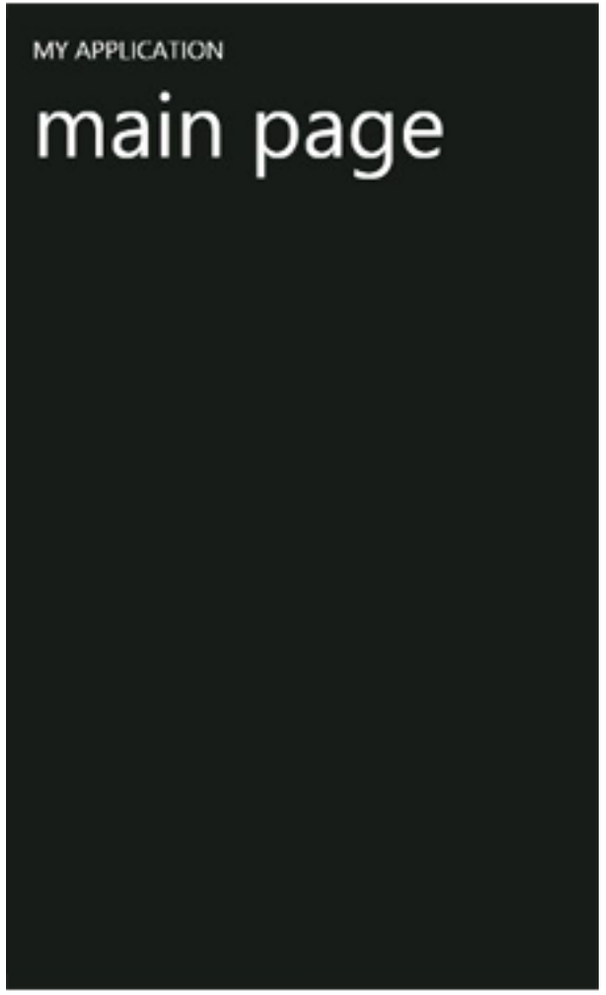
Data binding

tech·days Korea 2014
개발자 컨퍼런스

데이터 바인딩 이럴때 쓰세요!

앱의 뷰 로직이 복잡할때

Cross platform 지향
혹은 뷰 로직을 공유하고 싶을 때



단순한 앱에는 이 방법이 낭비일 수 있어요!

User data

멀티 디바이스에서 User data 처리

유저 데이터

tech·days Korea 2014
개발자 컨퍼런스

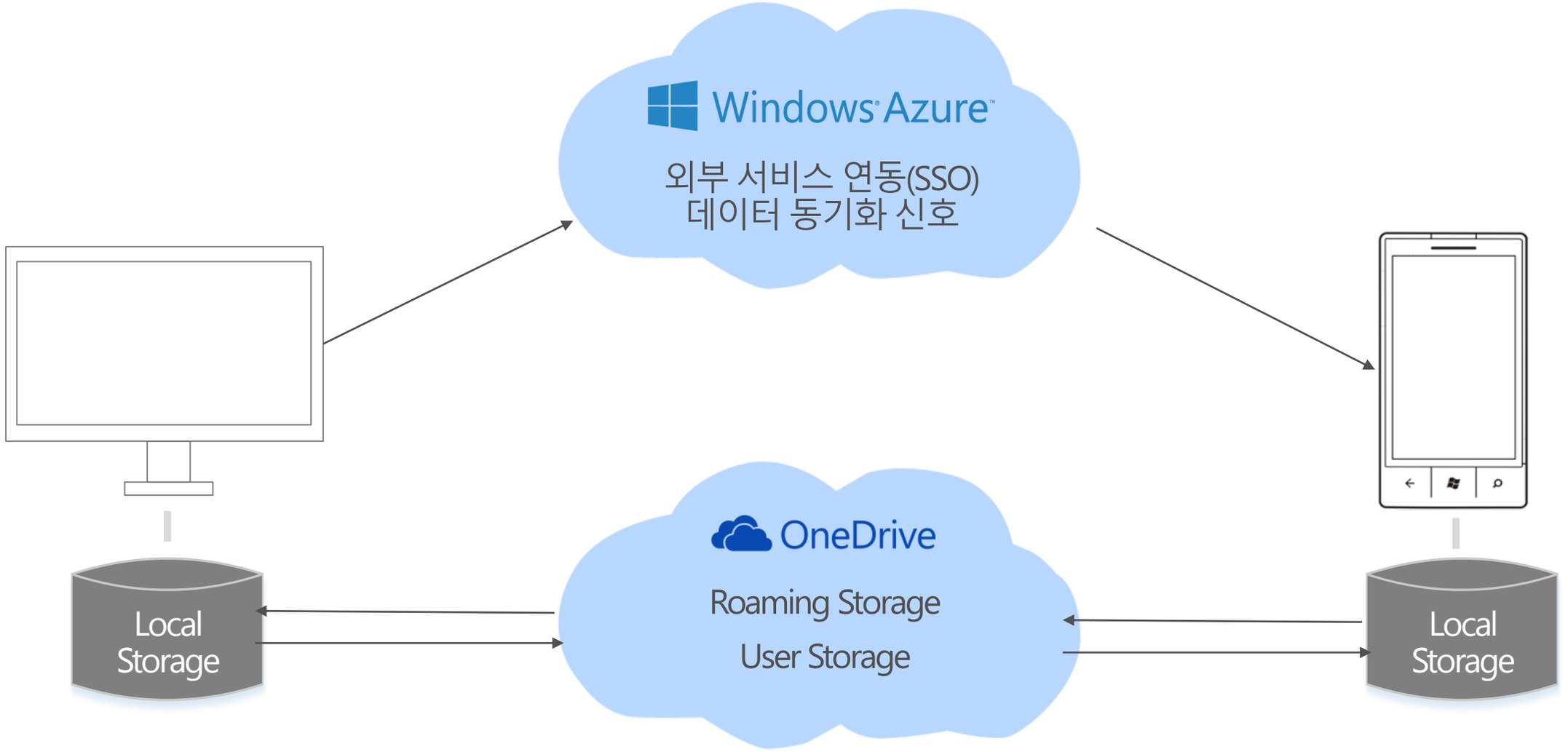
멀티디바이스에서의 데이터 동기화 시나리오

비용 지출X	비용 지출O
Roaming Storage (데이터가 100kb 미만, 간결한 설정 등, 즉결성 보장X)	서버 혹은 클라우드 서비스 사용
OneDrive (데이터가 100kb 이상, 데이터 용량이 큼)	

유저 데이터

tech·days Korea 2014
개발자 컨퍼런스

디자인 미의 유저 데이터 처리



유저 데이터

OnlineIdAuthenticator를 통한 Onedrive 동기화

```
public class WindowsAccountAuthenticator
{
        OnlineIdAuthenticator authenticator;

        public WindowsAccountAuthenticator()
        {
            authenticator = new OnlineIdAuthenticator();
        }

        /// <summary>
        /// 마이크로소프트 계정 로그인 및 인증
        /// </summary>
        public async Task SignIn()
        {
            var targetArray = new List<OnlineIdServiceTicketRequest>();
            targetArray.Add(new OnlineIdServiceTicketRequest("wl.signin", "DELEGATION"));

            var result = await authenticator.AuthenticateUserAsync(targetArray, CredentialPromptType.PromptIfNeeded);

            string AccessToken = result.Tickets[0].Value;
        }
}
```

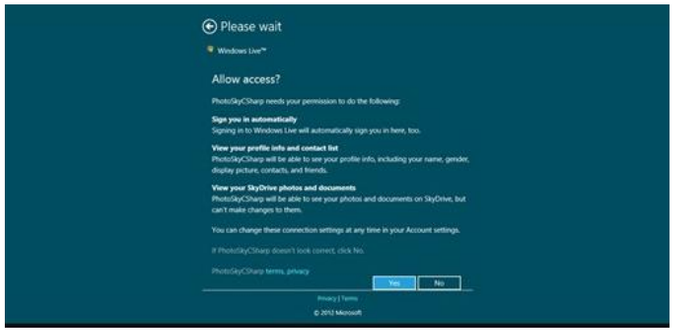
유저 데이터

tech·days Korea 2014
개발자 컨퍼런스

OnlineIdAuthenticator를 통한 Onedrive 동기화

OnlineIdAuthenticator
Sign in

Microsoft 계정 로그인
(이미 기기 연결시 생략)



사용자에게 권한 동의

Access
Token

Access Token 받아옴

Access Token으로
Onedrive와 동기화

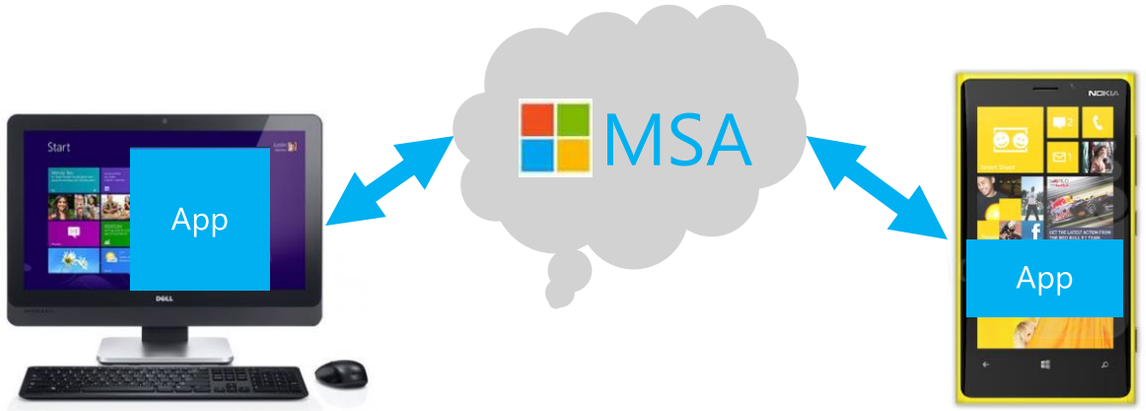
유저 데이터

tech·days Korea 2014
개발자 컨퍼런스

Credential Locker

API 가 User name과 Password를 안전하게 저장함, Roaming 됨

```
void SaveCredential(string username, string password)
{
    PasswordVault vault = new PasswordVault();
    PasswordCredential cred = new
PasswordCredential("MyAppResource", username, password);
    vault.Add(cred);
}

IReadOnlyList<PasswordCredential> RetrieveCredential(string
resource)
{
    PasswordVault vault = new PasswordVault();
    return vault.FindAllByResource(resource);
}
```



감사합니다:)

tech·days Korea 2014
개발자 컨퍼런스