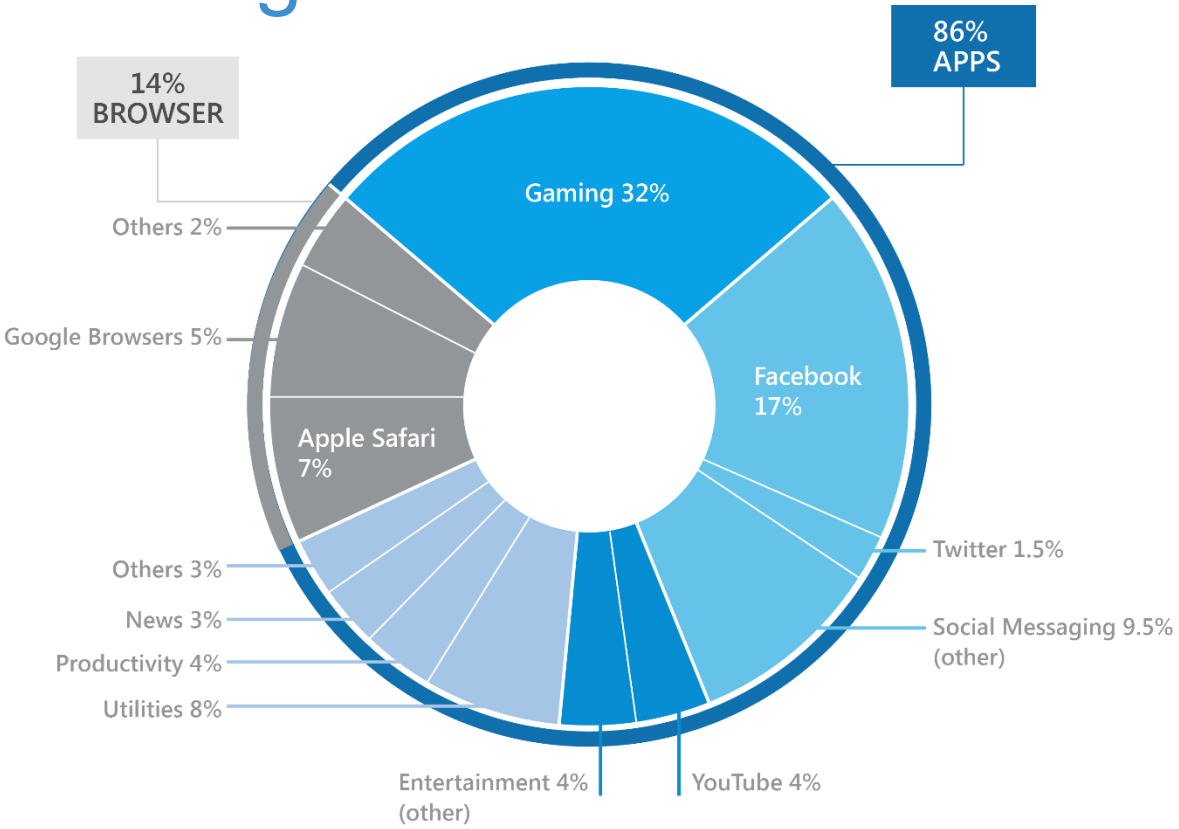
Developing Cross-Platform Mobile Apps Using Xamarin and PI Web API

Presented by **Marcos Vainer Loeff**

Introduction

What's Driving Mobile?

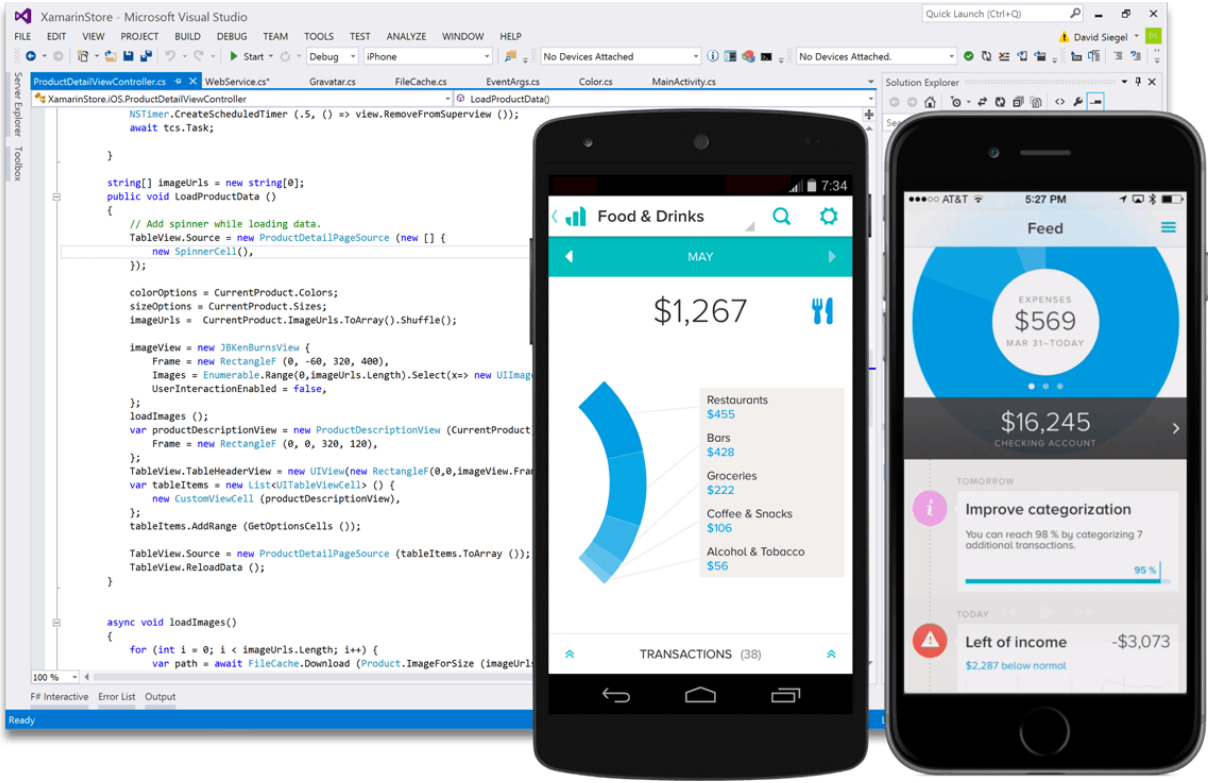


People Expect Great Experiences!

Facebook HTML

Facebook Native

Introducing Xamarin

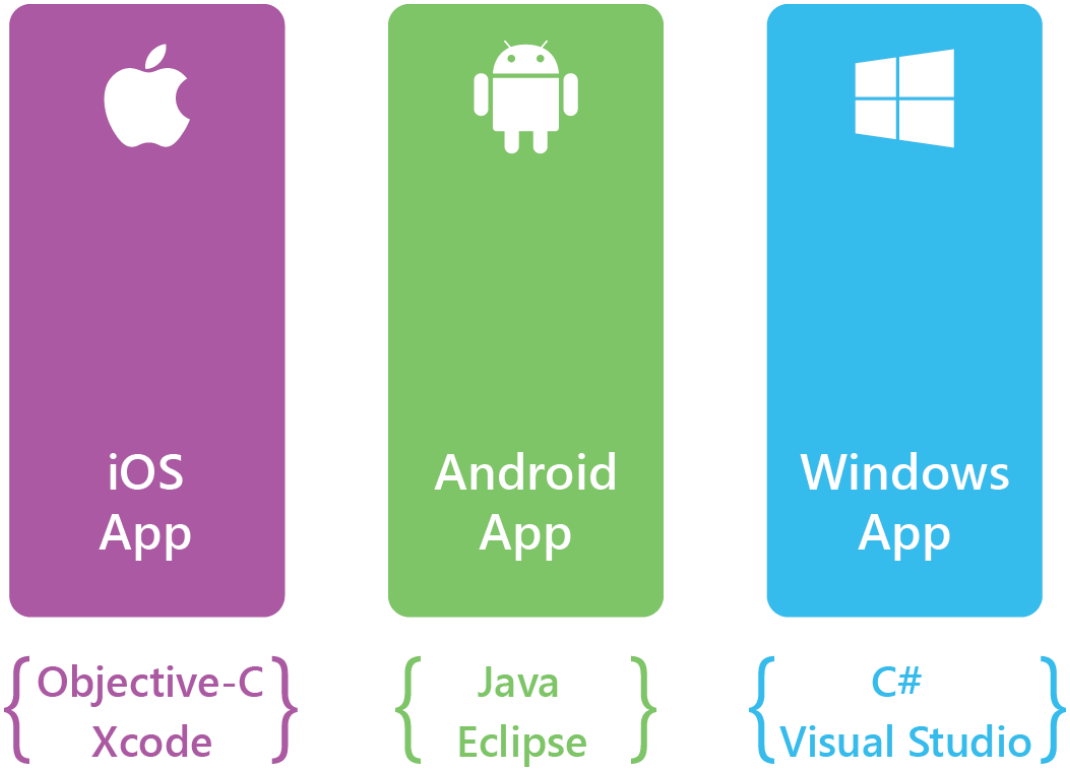


Xamarin Visual Studio

C# runs on 2.6 billion devices

Silo Approach:

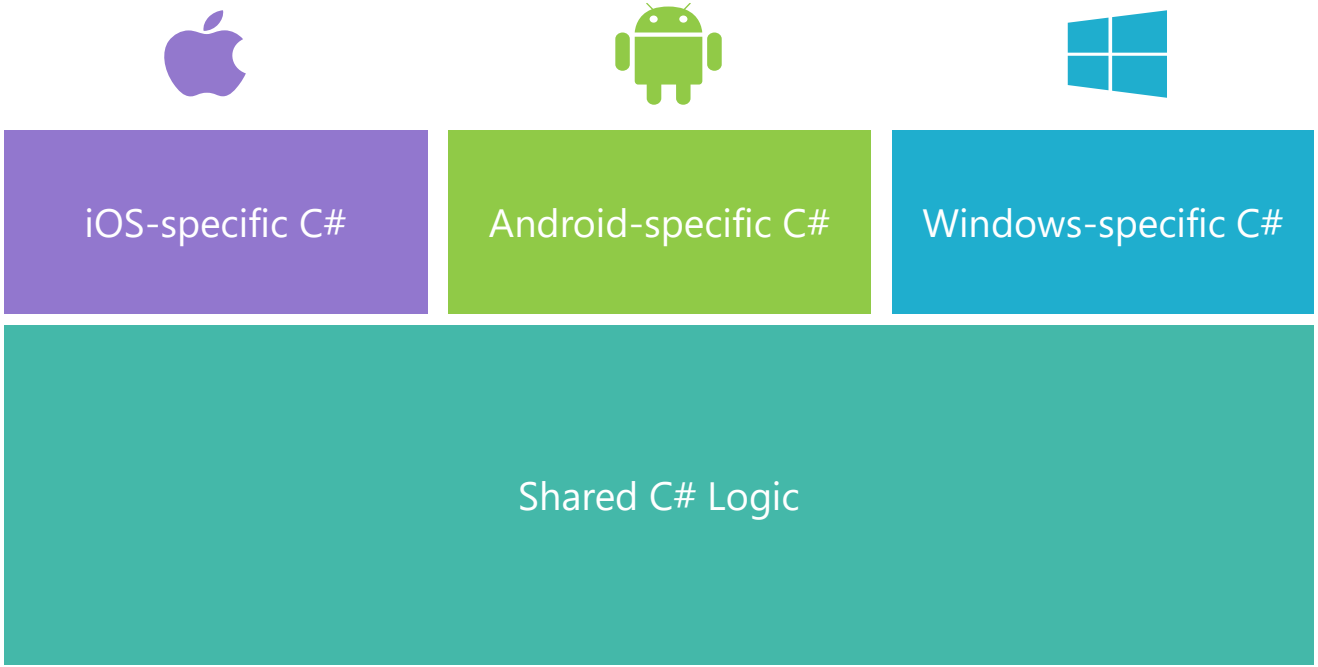
Build Apps Multiple Times



Mobile Agility with Xamarin and .NET

- Increase Your Reach
- Delight Your Customers
- High Quality Apps
- Lower Development Costs

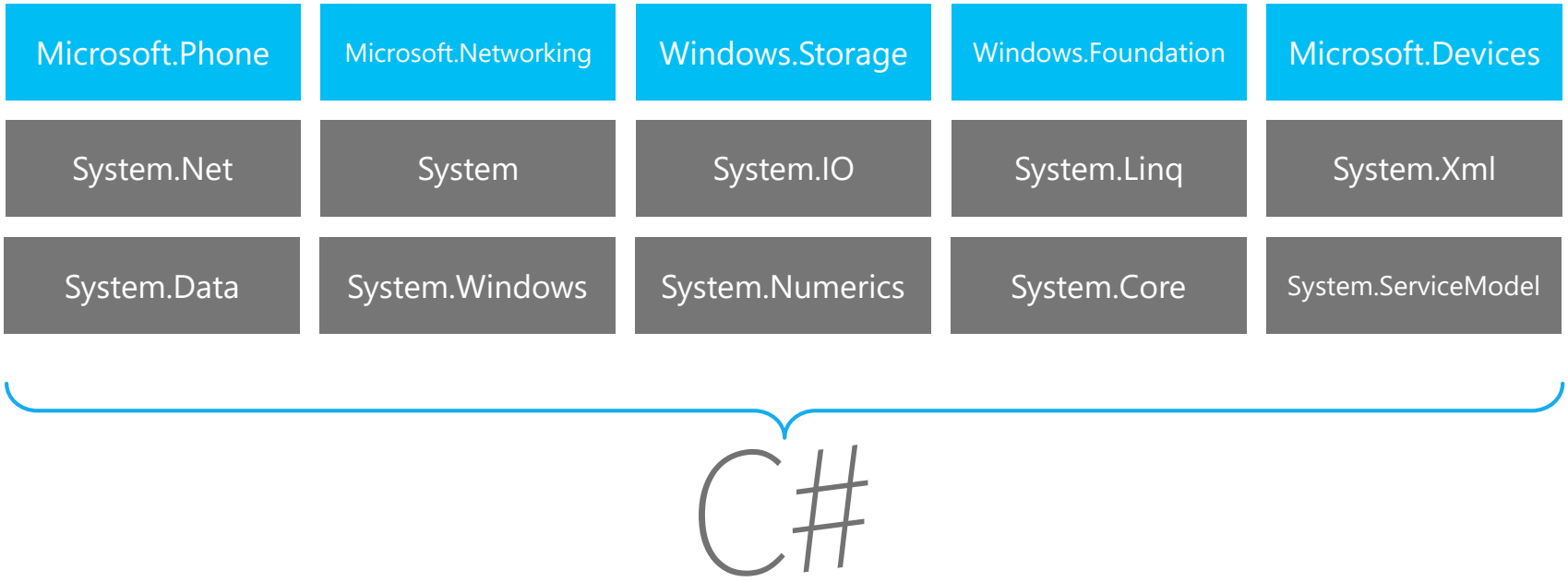
Xamarin’s Cross-platform approach



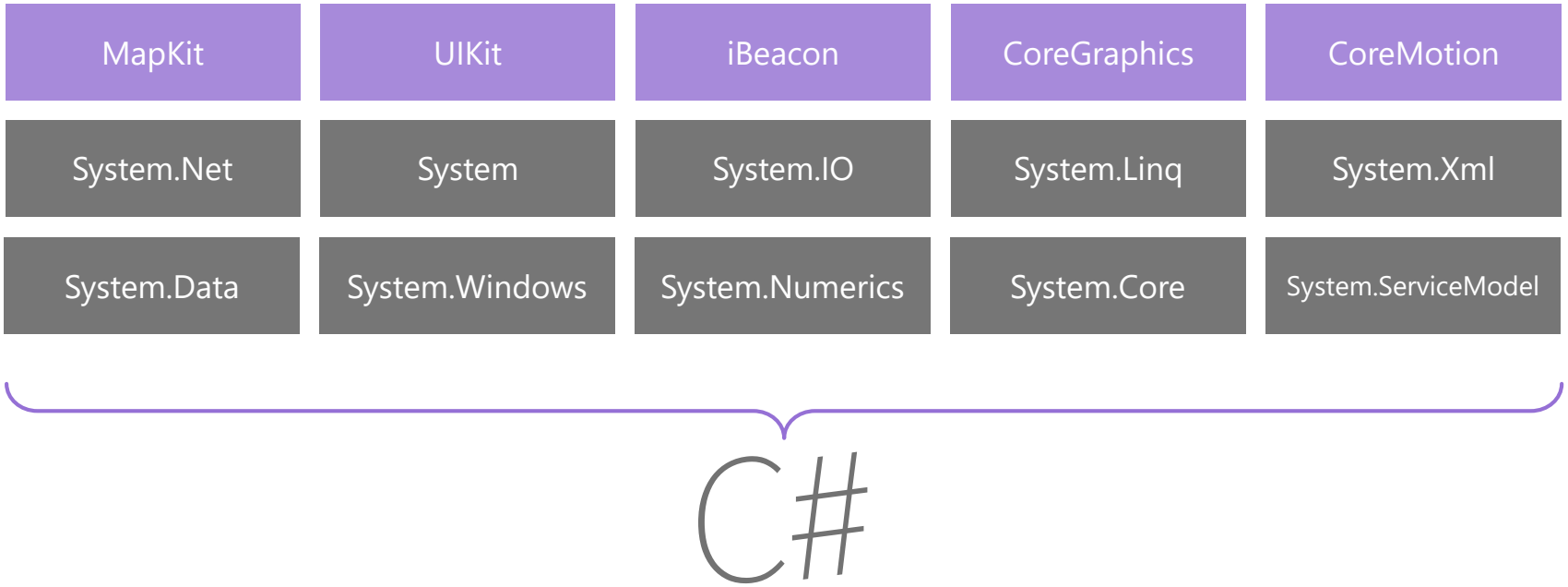
Shared C# codebase • 100% native API access • High performance

How does Xamarin work?

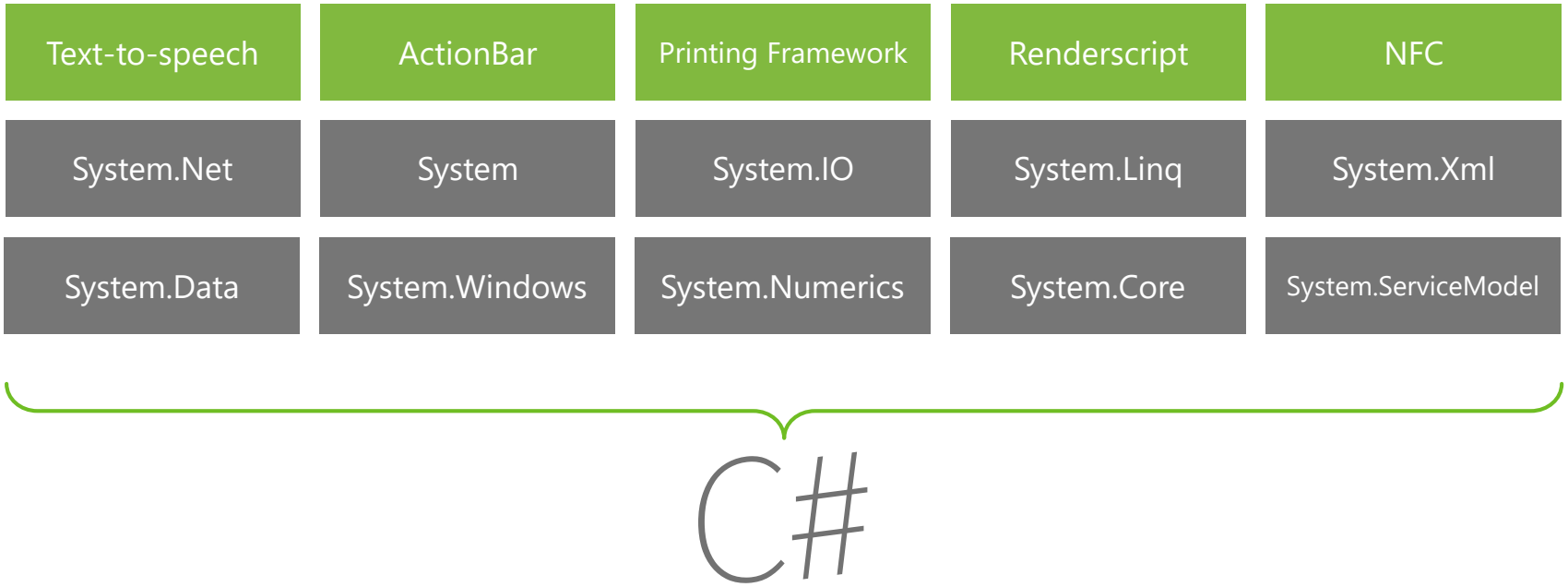
Windows APIs



iOS – 100% API Coverage

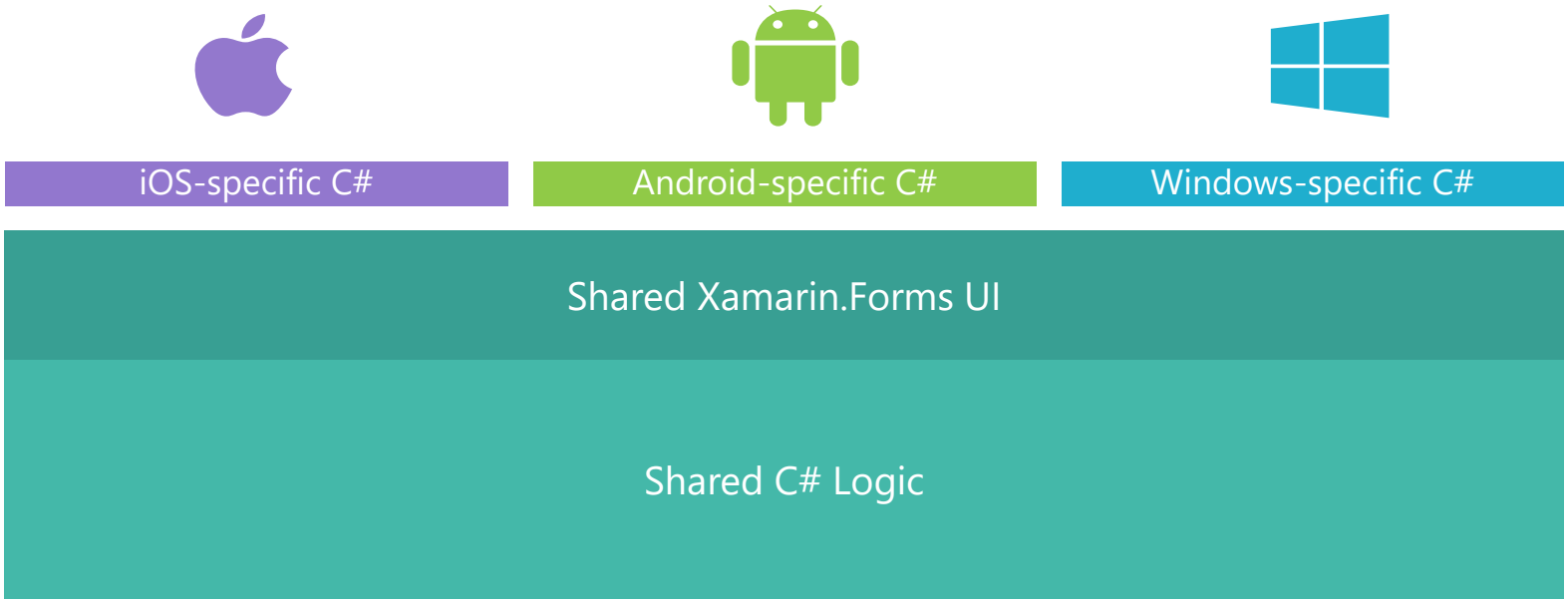


Android – 100% API Coverage



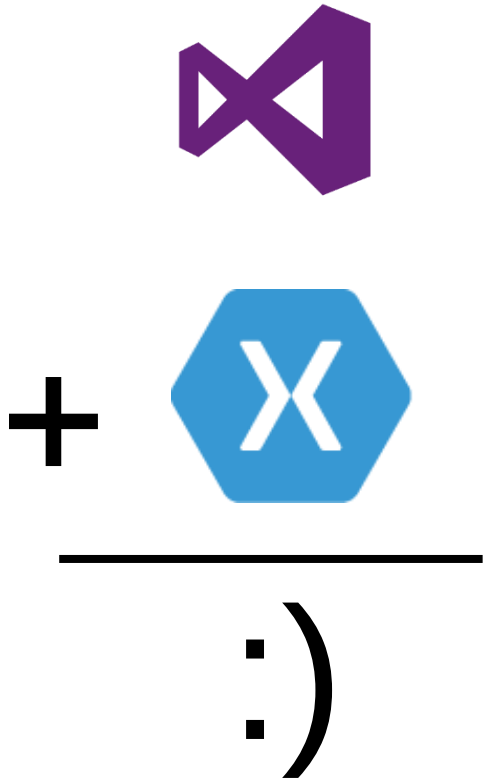
What is Xamarin.Forms?

Xamarin.Forms



How do I build a Xamarin App?

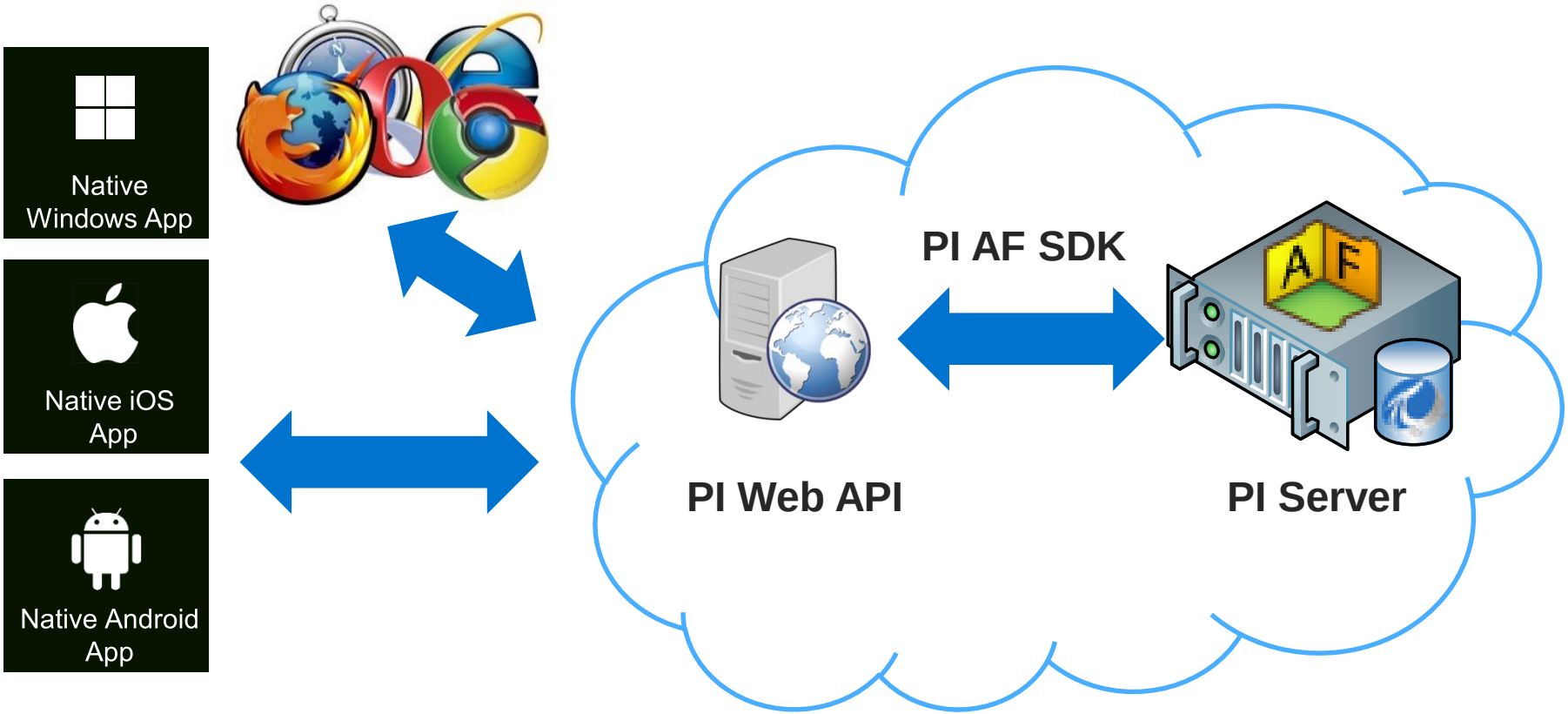
Xamarin for Visual Studio



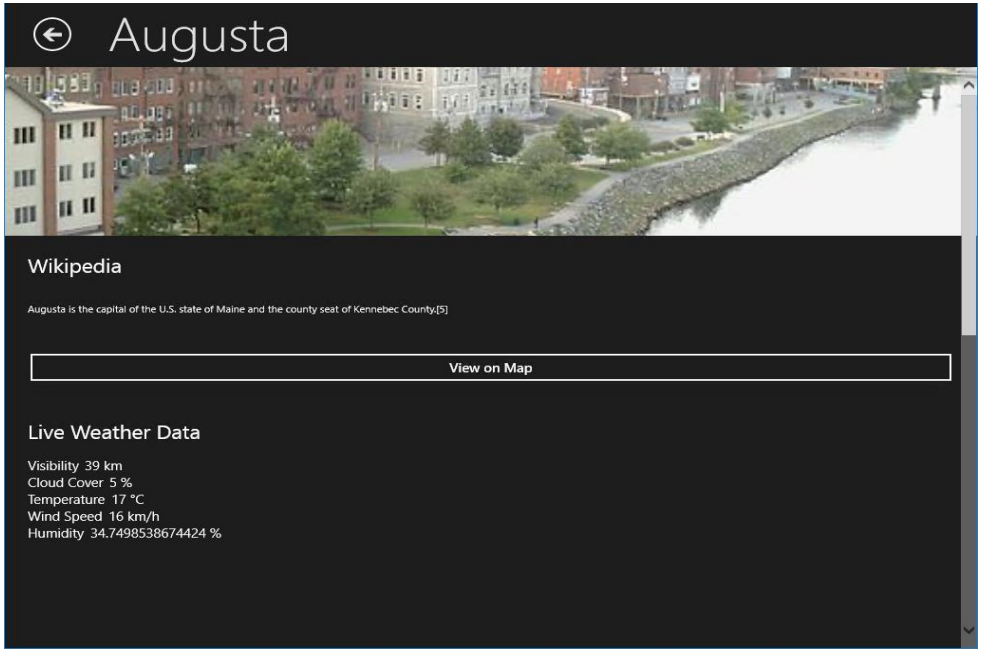
- Available for Windows and Mac
- Create, code, build, debug, package, and publish Xamarin applications for iOS, Android, and Windows from within Visual Studio

How do I develop a Xamarin app on top of the PI System?

How do I develop a Xamarin app on top of the PI System?



The Xamarin App for this lab



Exercise 1: Getting data from PI Web API

```
private async Task<Cities> GetCitiesDataNoBatch()
{
    //Exercise 1
    string url = baseUrl + @"assetdatabases?path=\\pifitness-srv2\Weather";
    string response = await DownloadWebData(url);
    PIObject dbData = JsonConvert.DeserializeObject<PIObject>(response);


    url = baseUrl + "assetdatabases/" + dbData.WebId + "/elementattributes?attributeNameFilter=*Wikipedia%20Thumbnail%20Url*";
    response = await DownloadWebData(url);

    PIListObject imageAttributes = JsonConvert.DeserializeObject<PIListObject>(response);
    List<string> webIds = imageAttributes.GetItemsWebIds();
    url = baseUrl + "streamsets/value?";
    foreach (string webId in webIds)
    {
        url = url + string.Format("webId={0}&", webId);
    }
    url = url.Substring(0, url.Length - 1);


    response = await DownloadWebData(url);
    PIListObject imagesValues = JsonConvert.DeserializeObject<PIListObject>(response);
    Cities cities = new Cities(imageAttributes, imagesValues);
    return cities;
}
```

Exercise 2: Xamarin Forms Maps

```
public class MapPage : ContentPage
{
    private City city;

    -- references
    public MapPage(City selectedCity)
    {
        //Exercise 2
        this.city = selectedCity;
        Position position = new Position(city.Latitude, city.Longitude);

        MapSpan mapSpan = MapSpan.FromCenterAndRadius(position, Distance.FromMiles(40));
        var map = new Map(mapSpan);
        var pin = new Pin();
        pin.Label = city.Name;
        pin.Position = position;
        map.Pins.Add(pin);
        Title = "Map";
        map.IsShowingUser = false;
        map.HeightRequest = 100;
        map.WidthRequest = 960;
        map.VerticalOptions = LayoutOptions.FillAndExpand;
        var stack = new StackLayout { Spacing = 0 };
        stack.Children.Add(map);
        Content = stack;
    }
}
```

```
– references
public async void OnViewMapBtnClicked(object sender, EventArgs e)
{
    //Exercise 2
    await Navigation.PushAsync(new MapPage(SelectedCity));
}
```

Exercise 3: Learning how to use ListView

```
<StackLayout Orientation="Vertical" Spacing="14" Padding="24,14,24,14">
  <Label Text="Live Weather Data" FontSize="Medium"  />
  <ListView x:Name="WeatherListView" ItemSelected="OnListViewItemSelected">
    <ListView.ItemTemplate>
      <DataTemplate>
        <ViewCell>
          <ViewCell.View>
            <StackLayout Orientation="Horizontal" >
              <Label Text="{Binding Property}"></Label>
              <Label Text="{Binding Value}"></Label>
            </StackLayout>
          </ViewCell.View>
        </ViewCell>->
      </DataTemplate>
    </ListView.ItemTemplate>
  </ListView>
 </StackLayout>
</StackLayout>
```

```
0 references
public async void OnListViewItemSelected(object sender, SelectedItemChangedEventArgs args)
{
    //Exercise 3
    WeatherData weatherData = ((ListView)sender).SelectedItem as WeatherData;
    await Navigation.PushAsync(new PITrendsPage(weatherData));
}
```

Exercise 4: Integrating your app with PI Coresight

```
public partial class PITrendsPage : ContentPage
{
    private WeatherData weatherData;
    -- references
    public PITrendsPage(WeatherData weatherData)
    {
        Title = string.Format("PI Coresight Trend - {0} - {1}", weatherData.CityName, weatherData.Property);
        this.weatherData = weatherData;
        InitializeComponent();
        Action actionDefault = () =>
        {
            Browser.Source = weatherData.GetCoresightUrl(false);
        };
        Action actionAndroid = () =>
        {
            Browser.Source = weatherData.GetCoresightUrl(true);
        };
        Device.OnPlatform(null, actionAndroid, actionDefault, actionDefault);
    }
}
```

Exercise 5: Using PI Web API Batch

```
private async Task<Cities> GetCitiesDataWithBatch()
{
    //Exercise 5
    Dictionary<string, PIBatchRequest> globalBatch = new Dictionary<string, PIBatchRequest>();
    PIBatchRequest batchGetDbWebId = new PIBatchRequest()
    {
        Method = "GET",
        Resource = baseUrl + @"assetdatabases?path=\\pifitness-srv2\Weather"
    };

    PIBatchRequest batchGetElementsWebIds = new PIBatchRequest()
    {
        Method = "GET",
        Resource = baseUrl + "assetdatabases/{0}/elementattributes?attributeNameFilter=*Wikipedia%20Thumbnail%20Url*",
        Parameters = new List<string>() { "$.1.Content.WebId" },
        ParentIds = new List<string>() { "1" }
    };

    PIBatchRequest batchGetAttributesValues = new PIBatchRequest()
    {
        Method = "GET",
        RequestTemplate = new PIRequestTemplate()
        {
            Resource = baseUrl + "streams/{0}/value"
        },
        Parameters = new List<string>() { "$.2.Content.Items[*].WebId" },
        ParentIds = new List<string>() { "2" }
```

```
            Parameters = new List<string>() { "$.2.Content.Items[*].WebId" },
            ParentIds = new List<string>() { "2" }
        };

        globalBatch.Add("1", batchGetDbWebId);
        globalBatch.Add("2", batchGetElementsWebIds);
        globalBatch.Add("3", batchGetAttributesValues);
        string json = JsonConvert.SerializeObject(globalBatch);
        HttpContent httpContent = new StringContent(json, Encoding.UTF8, "application/json");
        string url = baseUrl + "/batch";
        string response = await PostWebData(url, httpContent);
        Dictionary<string, PIBatchResponse> batchData = JsonConvert.DeserializeObject<Dictionary<string, PIBatchResponse>>(response);
        PIListObject imageAttributes = JsonConvert.DeserializeObject<PIListObject>(batchData["2"].Content.ToString());
        JArray items = (JArray)(batchData["3"].Content["Items"]);
        PIValue[] piValues = new PIValue[items.Count];
        for (int i = 0; i < items.Count; i++)
        {
            piValues[i] = JsonConvert.DeserializeObject<PIValue>(items[i]["Content"].ToString());
        }
        Cities cities = new Cities(imageAttributes, piValues);
        return cities;
    }
```

Contact Information

Marcos Vainer Loeff
mloeff@osisoft.com
Development Support Engineer
OSIsoft, LLC

감사합니다

谢谢

Danke

Merci

Gracias

Thank You

ありがとう

Спасибо

Obrigado

https://www.surveymonkey.com/r/QMBJP2G