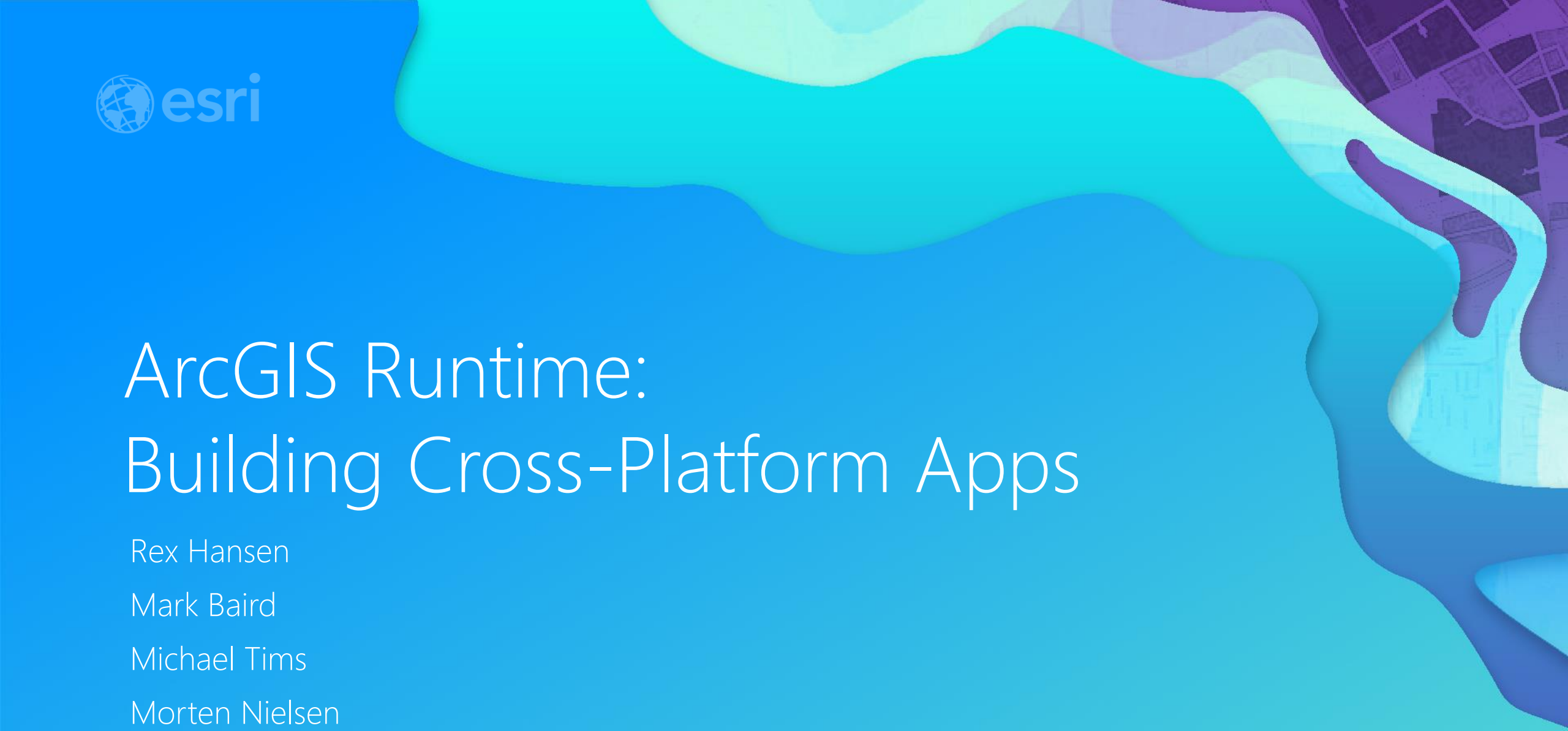


ArcGIS Runtime: Building Cross-Platform Apps

Rex Hansen

Mark Baird

Michael Tims

Morten Nielsen

2018 Esri DEVSummit Conference | Palm Springs, CA

Agenda

- Cross-platform review
- ArcGIS Runtime cross-platform options
  - Java
  - Qt
  - .NET

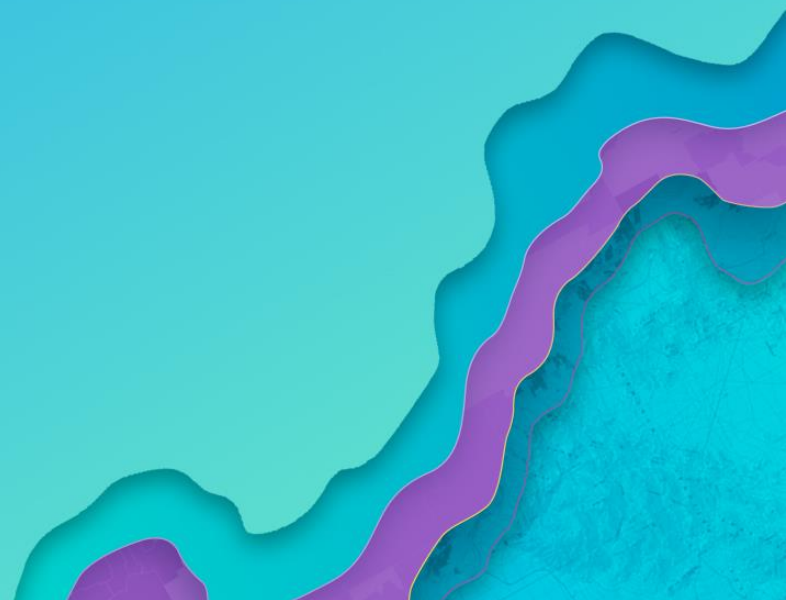
Native vs Web

- **Native** strategies offer the best device integration and the most out-of-the-box functionality for connected and offline workflows, but they require native development skills. You can use ArcGIS Runtime SDKs to create native apps.

- **Web** strategies use HTML, JavaScript, and CSS hosted on a web server and delivered to the user's device or desktop using a web browser. This strategy is best for connected workflows if you don't know the devices your users have and you need to reach a wide audience. You can use the ArcGIS API for JavaScript to create web client solutions.

Cross-platform native application considerations

- Benefits
  - Share application code
  - Enforces good design patterns
  - Makes your app available to more users
- Challenges
  - User experience of your app may vary
  - Handling platform idiosyncrasies (security, bugs, etc)
  - More testing
  - Development cost



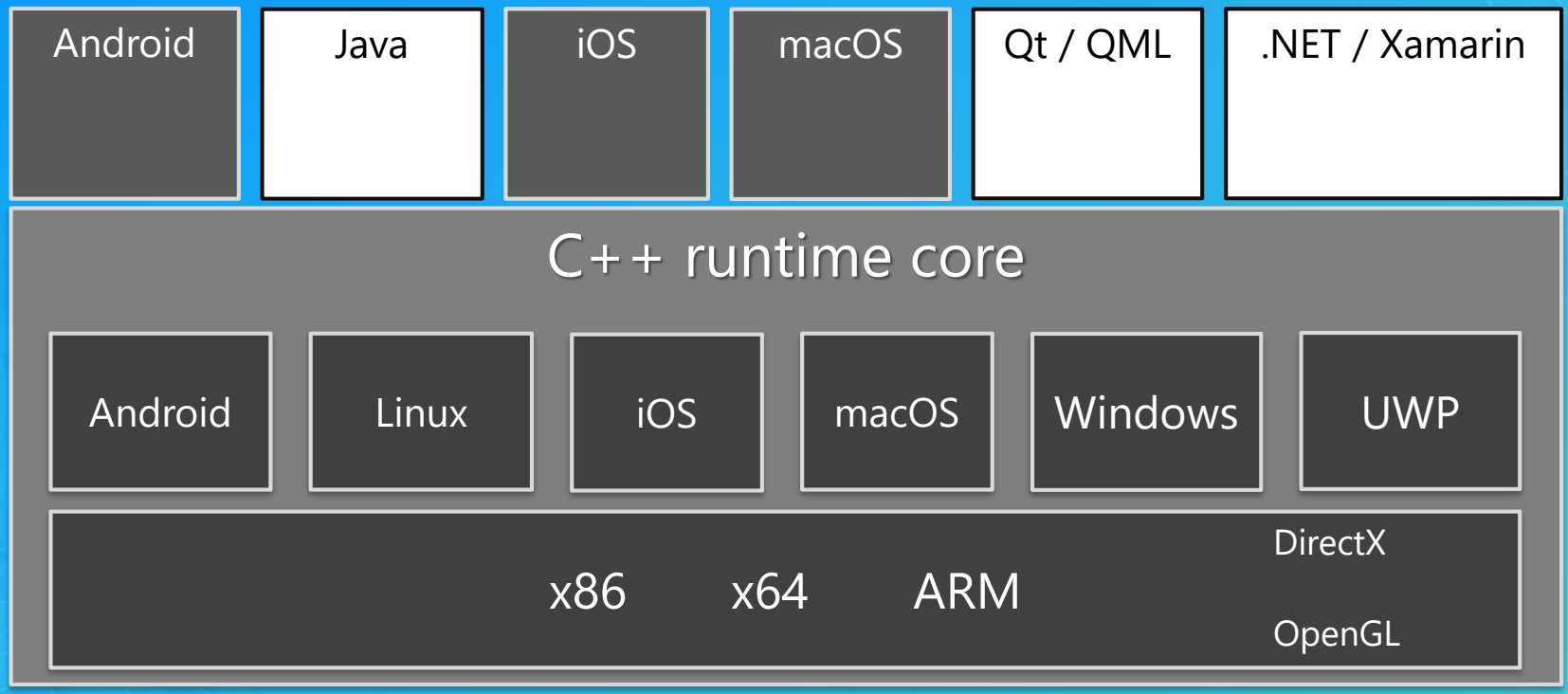
Building Native Apps on Multiple Platforms

- Which One is Right for Me?
  - Understand expectations of your users
  - Educate yourself and your team
  - Be prepared to learn something new

- Multiple players in the market
  - Java
  - Qt Company
  - Microsoft

ArcGIS Runtime cross-platform options

- All Runtime APIs built on common Runtime core



Java

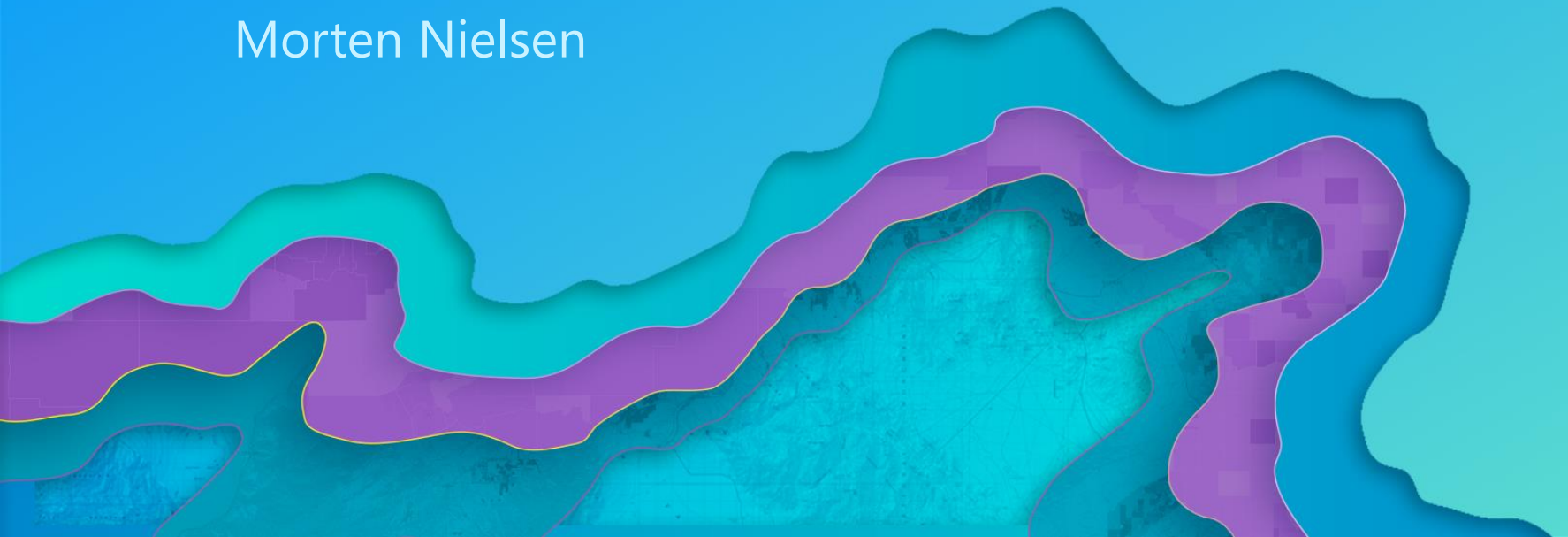
Mark Baird

Qt/QML

Michael Tims

.NET/Xamarin

Morten Nielsen



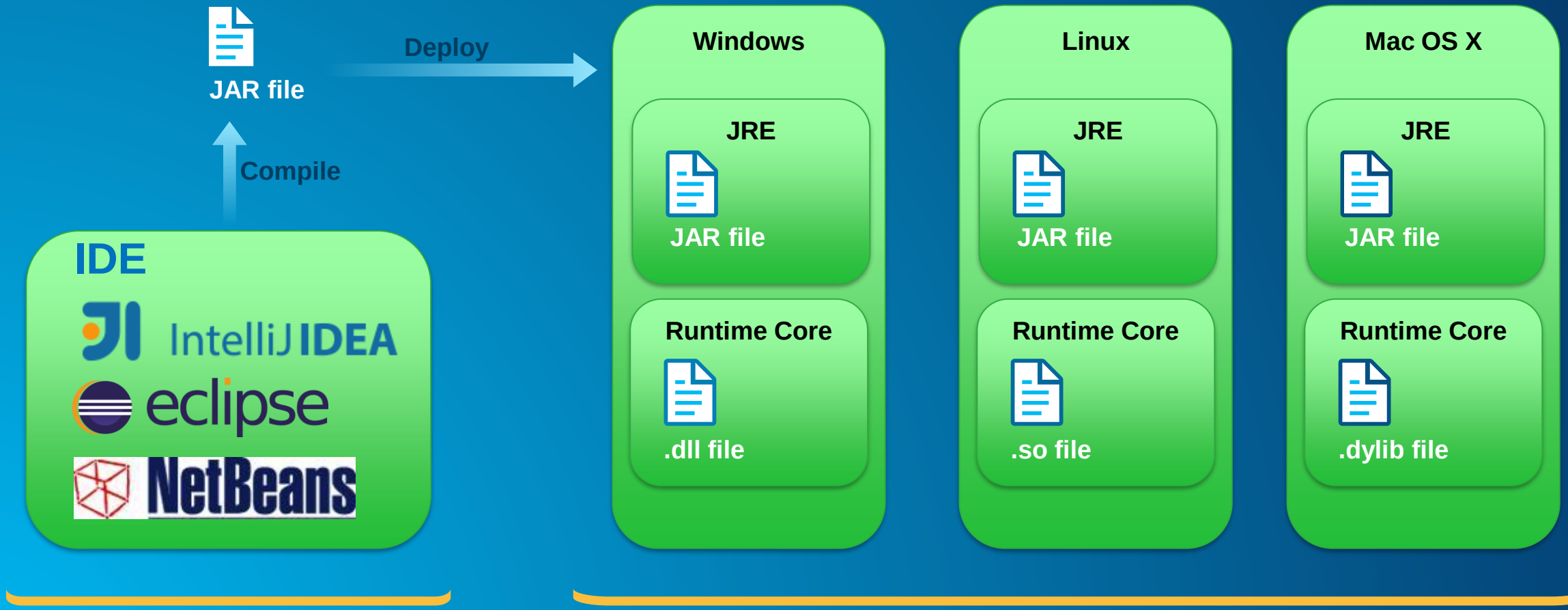
Java

Mark Baird

Cross platform Java Development

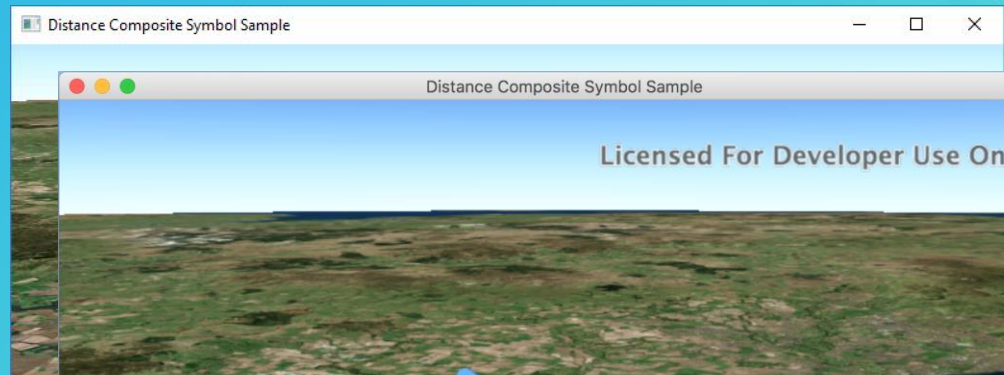
- Java is everywhere!
- Write applications for multiple platforms
- 100.2.1 Release supports Windows, Mac and Linux
- Java Runtime is aimed at desktop platforms (JavaFX based)
- Familiar architecture where applications run in a JVM
- Sits on the ArcGIS Runtime architecture

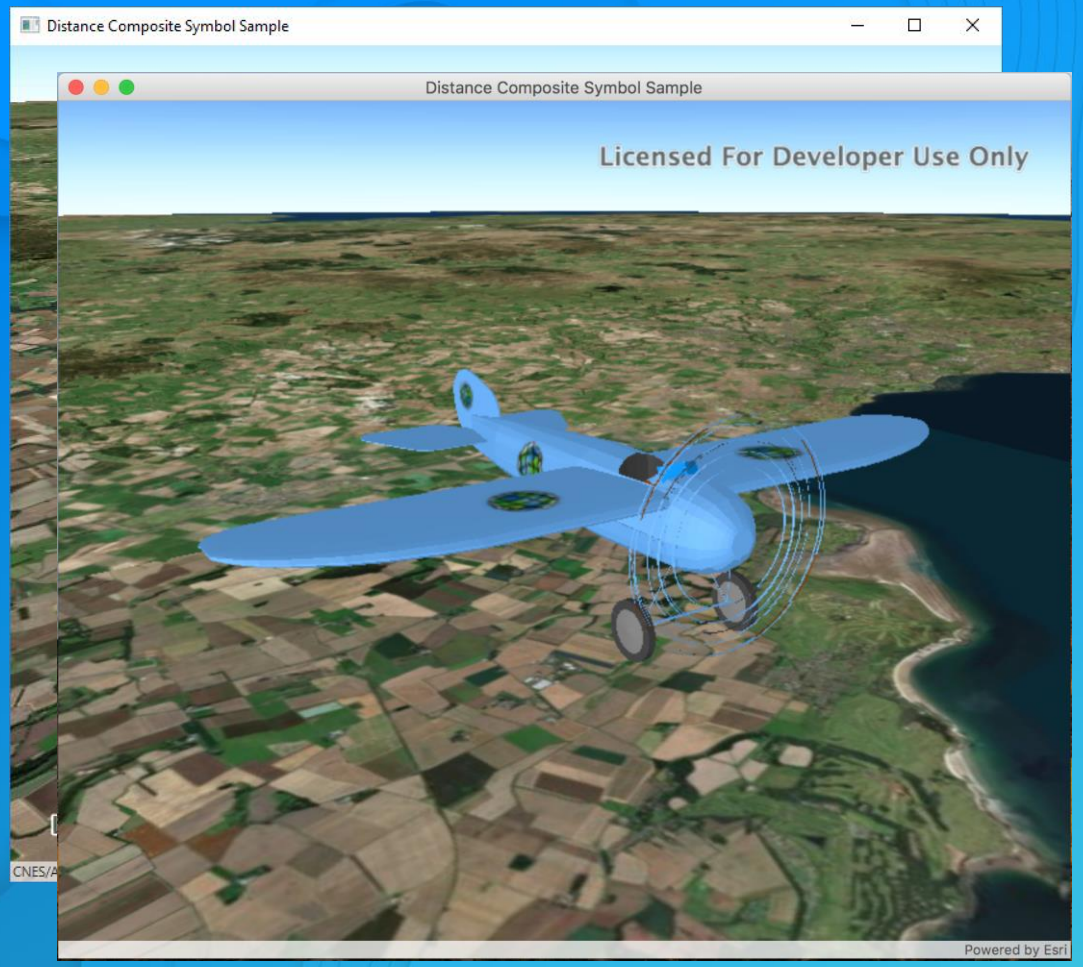
Java Runtime development and deployment



Considerations for writing cross platform in Java

- Generally it just works!
- JavaFX is consistent in behaviour on all platforms
- File systems need special consideration:
  - Windows compared to Mac and Linux:
    - C:\Users\Mark\Projects\MyData.geodatabase
    - /Users/Mark/Projects/MyData.geodatabase
  - If possible use relative path locations from your executable JAR
  - Use “/” as it works on Windows too.
    - But to be sure : System.*getProperty("file.separator");*
  - *Environment variables might help too.*





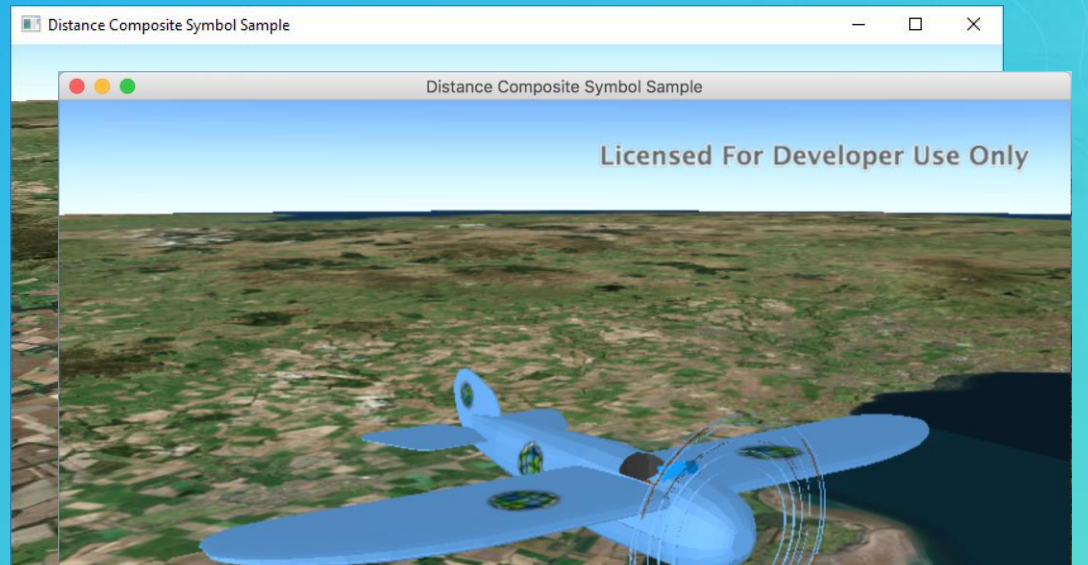
An App for Windows, Linux and a Mac

Mark Baird

Summary (development for desktop platforms)

- **Pros**
  - IDE and JRE is free
  - Deployments can be identical for ALL platforms
  - JavaFX apps style for the platform

- **Cons**
  - Apps will not work if JRE not present
    - Enterprise deployments of Java are easy
    - Self contained application packaging
  - Not for mobile applications



Qt/QML

Michael Tims

Qt

The Qt Company – www.qt.io

- What do you get with Qt?
  - A complete cross-platform software framework
  - Ready-made UI elements, C++ libraries, and tooling
- Over 1 million developers (indie, corporate) worldwide
- Open-source community
- Code less, create more, deploy everywhere

Platform Support

ArcGIS Runtime SDK for Qt

- Windows – x86, x64
- Linux – x64
- macOS – x64
- Android – armv7, x86
- iOS – armv7, arm64, sim
- Universal Windows Platform (UWP)?
- Linux Embedded arm?

2 APIs

ArcGIS Runtime SDK for Qt

- C++ API
  - Qt Widgets for UI
  - Qt Quick (QML) for UI
  - Modern C++ language – C++ 11
  - Fast performance – direct binding to the Runtime Core (C++)

- QML API
  - Qt Quick (QML) for UI
  - Declarative language
  - Imperative JavaScript business logic code
  - AppStudio (Survery 123)

QML API

Example QML API code

Highly readable JSON/CSS-like syntax

Declarative UI elements

Imperative JavaScript Code to handle events

```
Rectangle {
    MapView {
        id: mv
        anchors.fill: parent
        Map {
            id: map
            BasemapStreetsVector {}
        }
    }
     Button {
        anchors {
            left: parent.left
            top: parent.top
        }
        text: "Zoom to Hawaii"
        enabled: map.loadStatus === Enums.LoadStatusLoaded
        onClicked: {
            var point = ArcGISRuntimeEnvironment.createObject("Point", {
                            x: -157.564,
                            y: 20.677,
                            spatialReference: SpatialReference.createWgs84() } );
            mv.setViewpointCenterAndScale(point, 4000000.0);
        }
    }
}
```

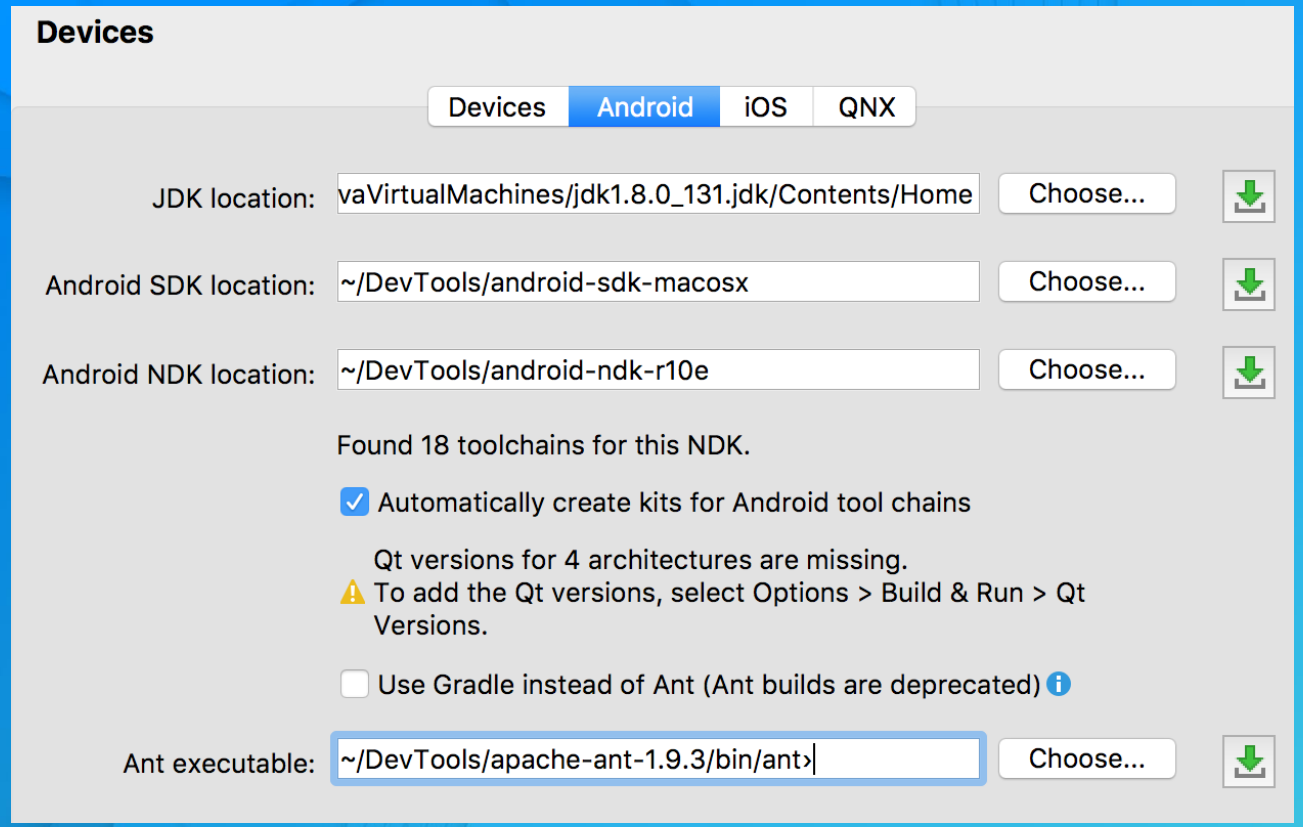
ArcGIS Runtime

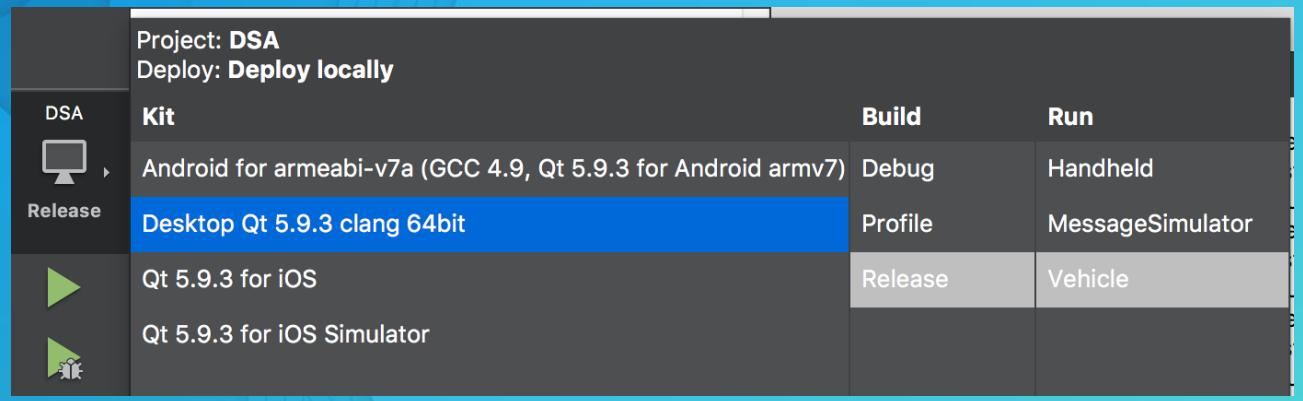
Dynamic property binding

Qt SDK – System setup

Set up your build environments

- Compiler, SDK dependencies
  - iOS: Xcode compiler
  - Windows: Visual Studio compiler, debugging tools
  - Linux: GCC compiler
  - Android: Android NDK and SDK
- Setup once, same cross-platform code
- IDE: Qt Creator - Cross-platform IDE
  - Design, develop, test, deploy from one tool





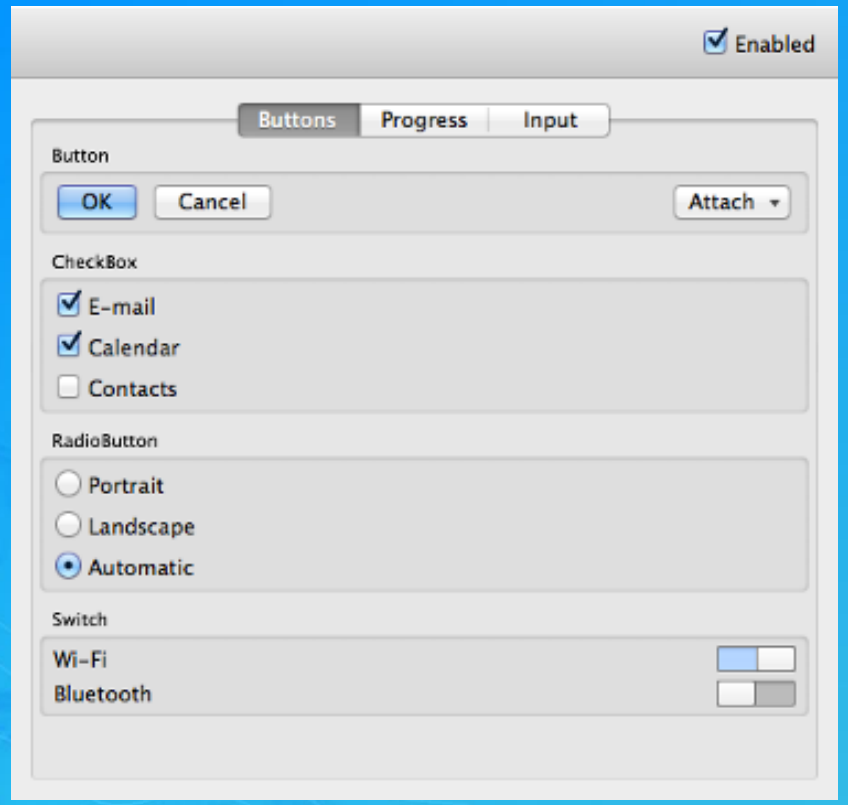
Qt SDK

Demo - Getting started

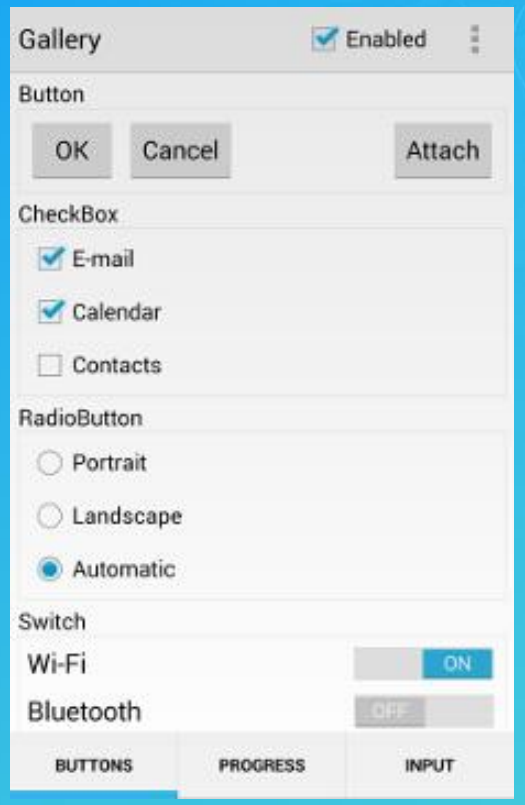
Qt Quick Controls

Ready-made UI elements

- **macOS**



- **Android – Nexus 5**

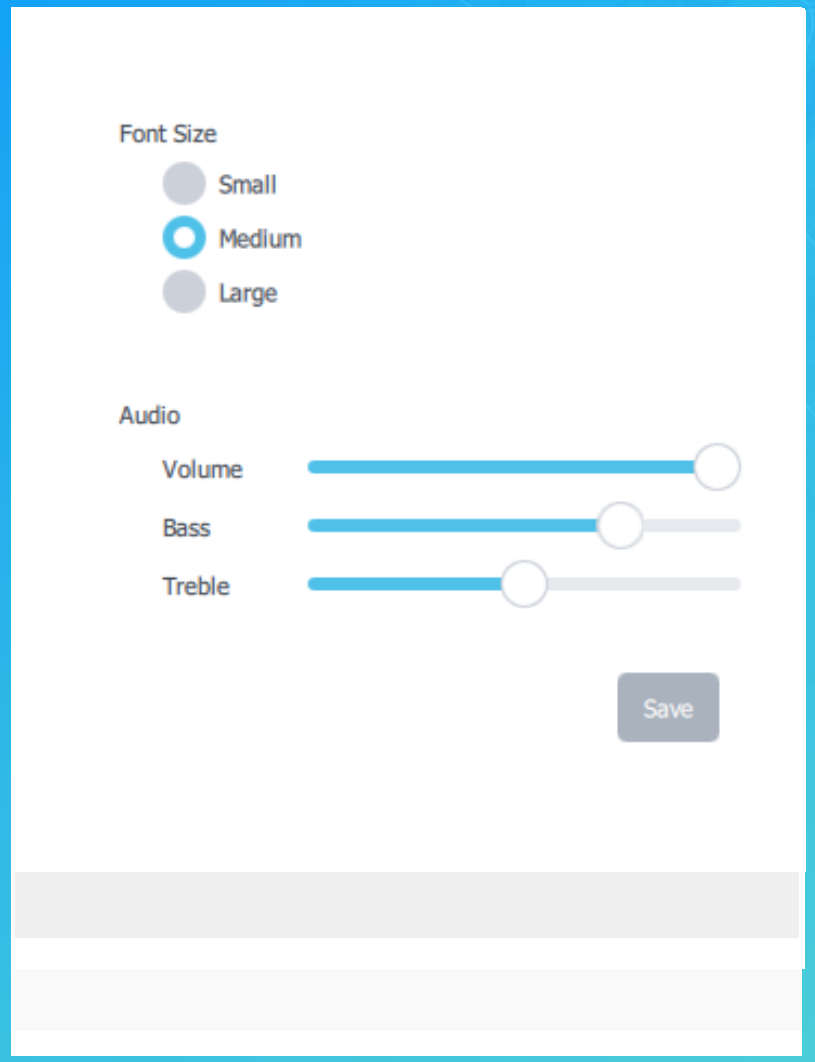


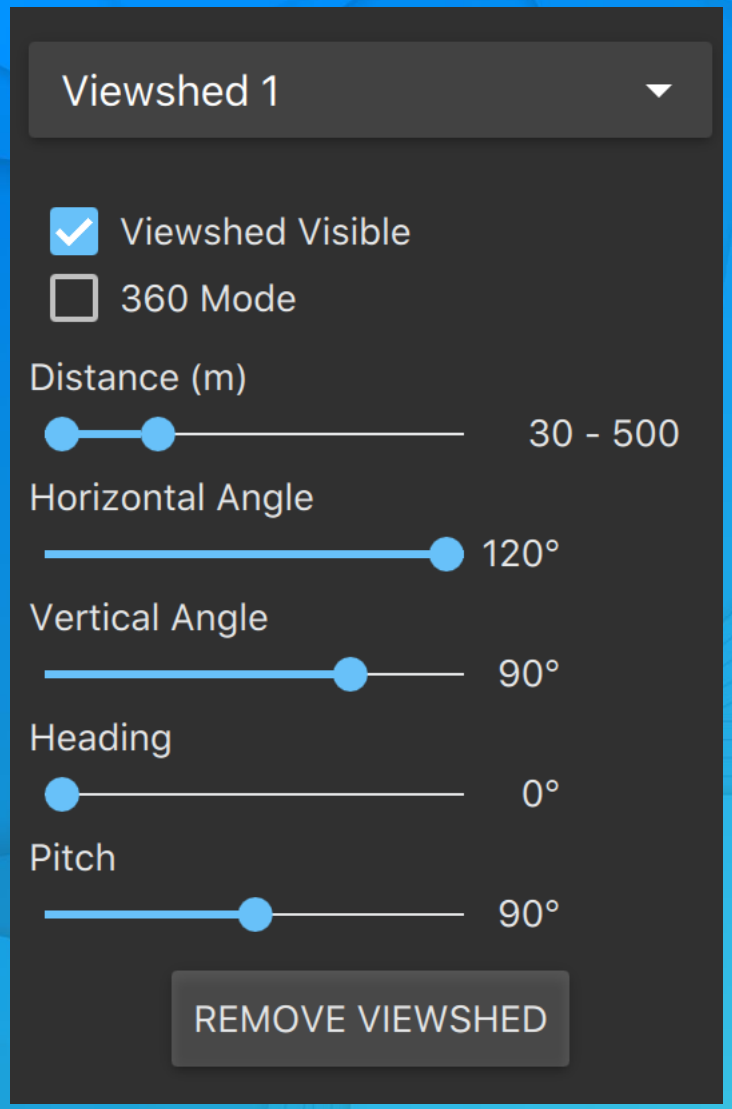
Qt Quick Controls 2

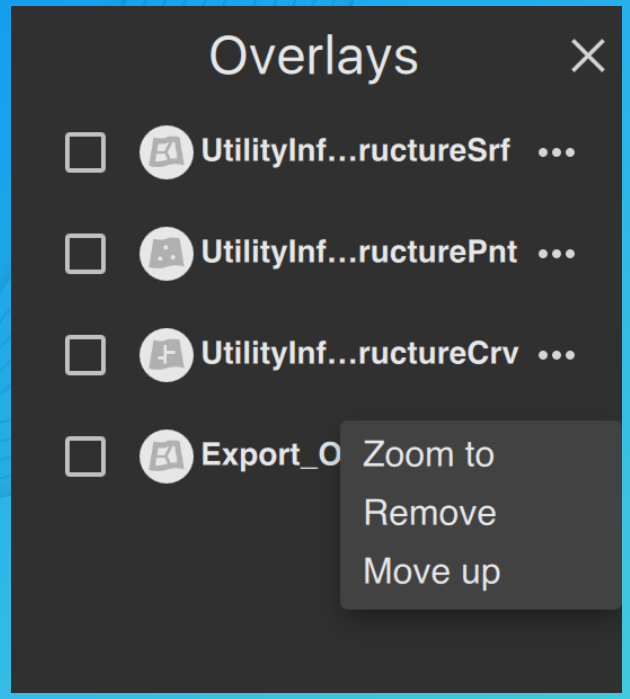
More ready-made UI elements

- Available Styles
  - Default style
  - Material style
    - Google’s guidelines
  - Universal style
    - Microsoft’s guidelines
  - Fusion style
    - Desktop-oriented look and feel
  - Imagine style
    - Based on image assets

- Default style

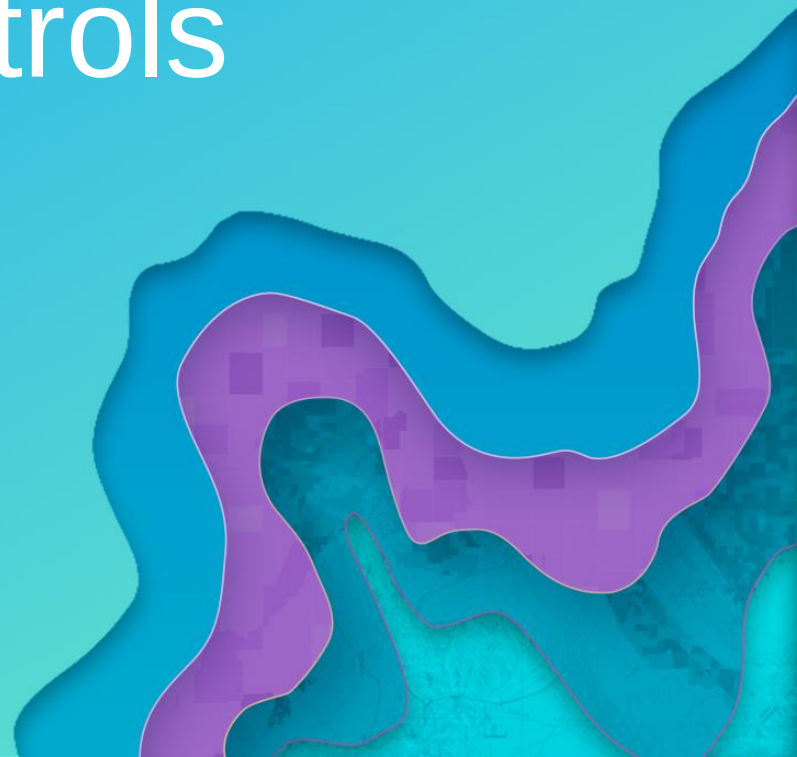






Qt Quick Controls

Demo



Qt/QML Limitations

Cross-platform hurdles

- Missing APIs
  - Manual implementation of Qt abstraction APIs
    - No push notification API
    - No access to Android photo gallery
    - No inter-App communication APIs
  - QML is designed for UI
    - C++ only APIs, i.e. networking, file IO, and text-to-speech
      - ArcGIS Extras and AppStudio Framework wrappers
- UI Limitations
  - Inconsistencies with native look and feel

Qt SDK framework licensing

Open Source or Commercial Use

- Qt for Application Development license model
- Dual-licensed under commercial and open source licenses
  - Commercial
    - Full rights to create and distribute software
  - Open Source
    - Qt 5.6.x - LGPLv2.1
    - Qt 5.7.x and up - GPL and LGPLv3
- More information - http://www.qt.io/download

Qt SDK – Pros and Cons

ArcGIS Runtime SDK for Qt

- Pros
  - Same modules
  - Same code
  - Same workflow
  - Same look and feel
  - Access to device sensors
  - Open-source community
- Cons
  - Incomplete native APIs
    - Common abstraction API for iPhone and Android SDKs
  - Look and feel of the native platform
  - Qt framework can increase apps size

.NET/Xamarin

Morten Nielsen

What is Xamarin?

- Xamarin
  - Based on the Mono runtime*
  - Compiles into a native Android or iOS app
  - Exposes ***all*** Android and iOS APIs

- Xamarin is not a cross-platform SDK. It's a cross-platform language (C#)
  - Most of .NET's core libraries are shareable code
  - UI code is very platform specific
  - Device code not shareable (Bluetooth, GPS, sensors etc)

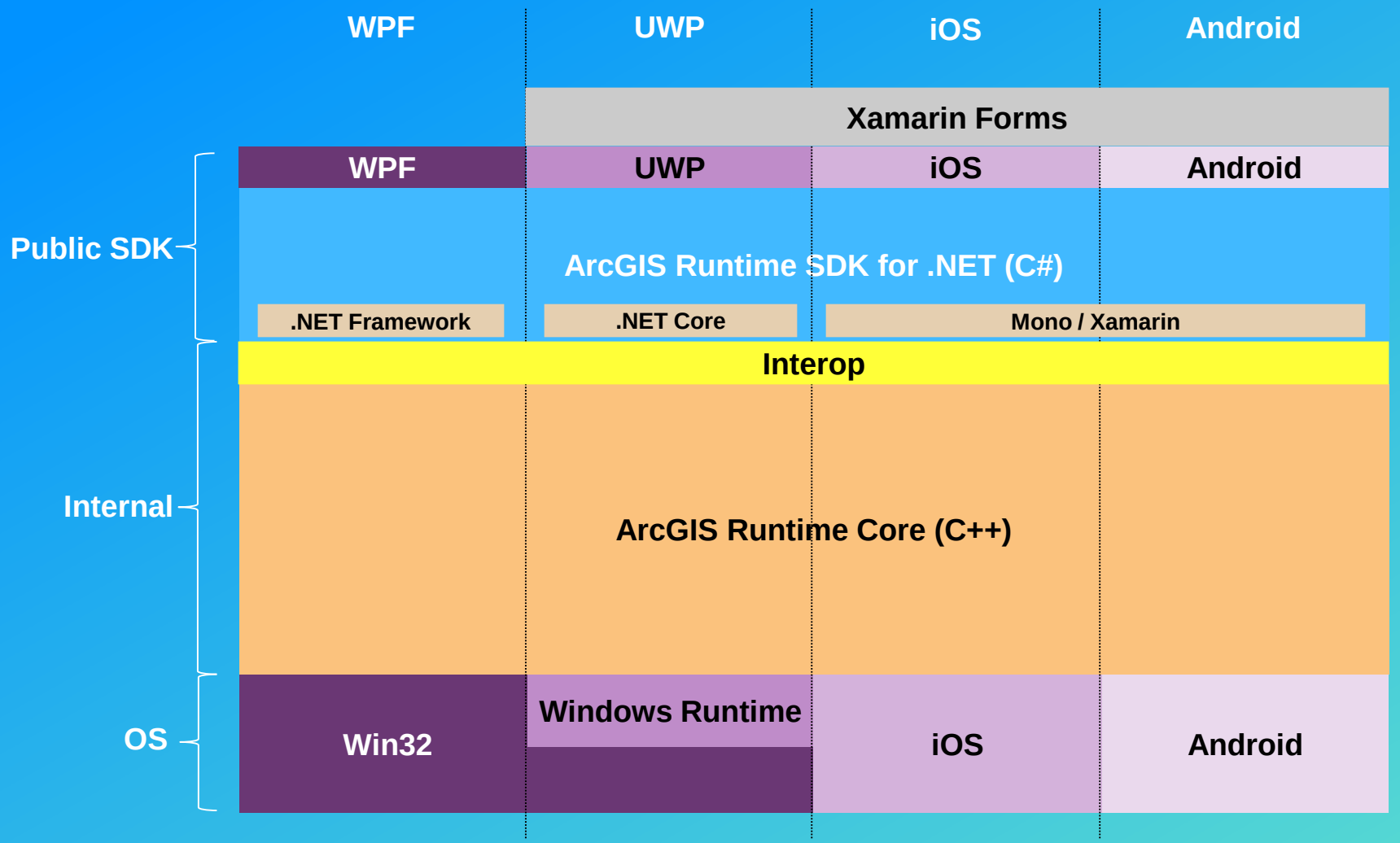
- Abstraction-libraries exist that simplifies this
  - Xamarin.Forms: Cross-platform UI framework which supports XAML
  - Lots of nuget-libraries

What is ArcGIS Runtime SDK for .NET

- **Supports:**
  - **WPF**
  - **Windows Universal app (UWP)**
  - **Xamarin Android**
  - **Xamarin iOS**
  - **Xamarin.Forms (Android, iOS, UWP)**
- **Exact same business logic code for all platforms.**
  - **Use native platform UI framework**
  - **Or use Xamarin.Forms abstraction to share UI logic as well**

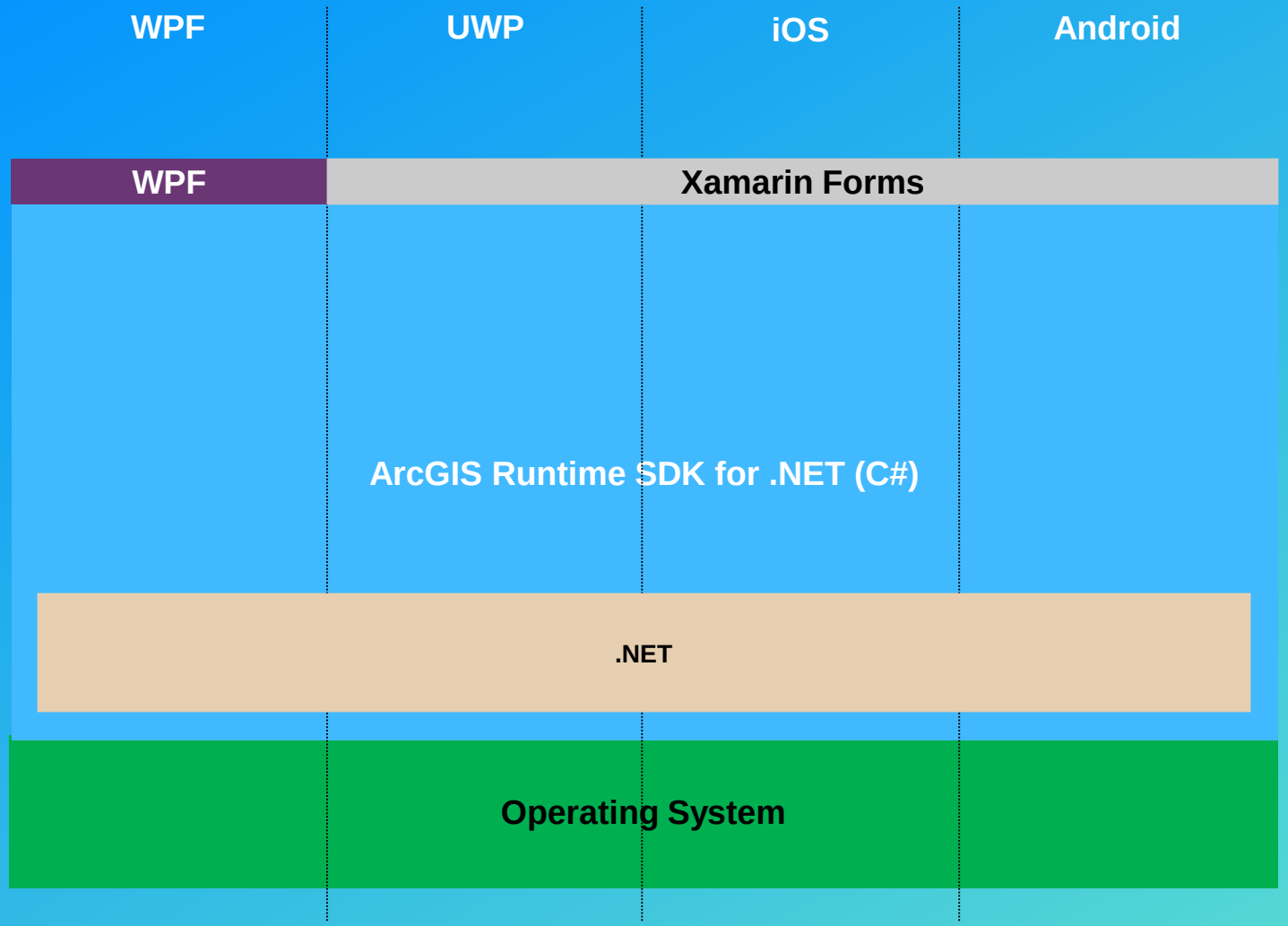
ArcGIS Runtime SDK for .NET & Xamarin

Architecture Diagram



ArcGIS Runtime SDK for .NET & Xamarin

A simpler view…



Native UI

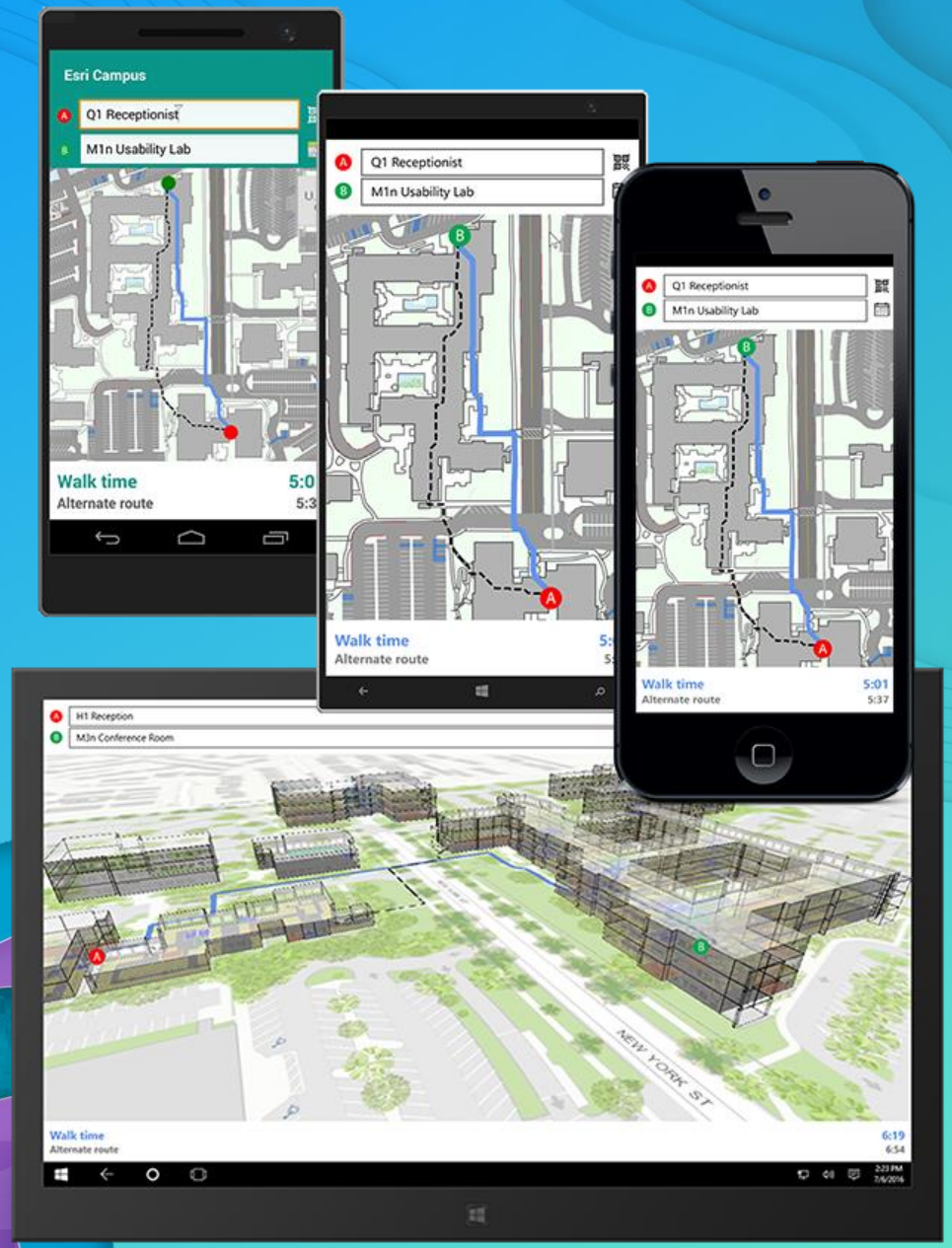
- *MapView, SceneView*

Common SDK

- *Maps, scenes, layers, symbols, feature data, geocoding, routing, portal, rasters, offline/syncing, geometry, geometry engine…*

Xamarin Demo

Source: http://esriurl.com/XamarinRouting



Pros and Cons

- **Pros:**
  - Xamarin is free and Open Source
    - Note: Visual Studio isn’t free for most commercial uses though
  - Target all platforms in a single IDE (Visual Studio), on a single OS*
  - Full access to all native platform APIs
- **Cons:**
  - Not 100% abstraction of all platform code
  - Xamarin Tooling (while greatly improving) can be a little buggy
  - *You need a Mac for iOS deployment

Questions?

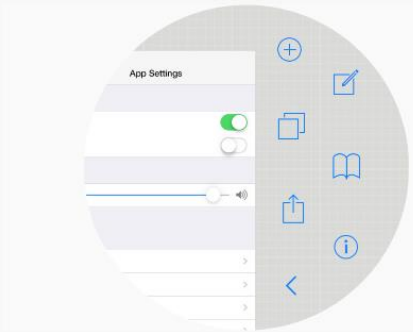
developers.arcgis.com/arcgis-runtime

What are ArcGIS Runtime SDKs?

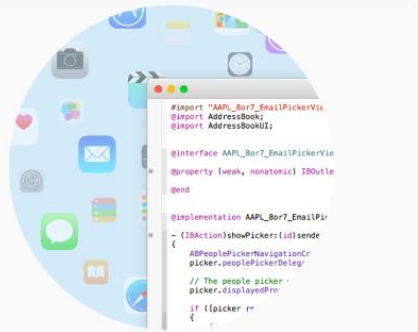
Built natively from the ground up using C++ and GPU acceleration, ArcGIS Runtime SDKs expose the full capability of the ArcGIS Platform to mobile, desktop, and embedded devices. Whether you're using ArcGIS Online or ArcGIS Enterprise or have disconnected users, ArcGIS Runtime SDKs let you do all things GIS, from simple map display or routing to advanced analysis.

Choosing the Right Esri API

Work Offline



Native User Experience

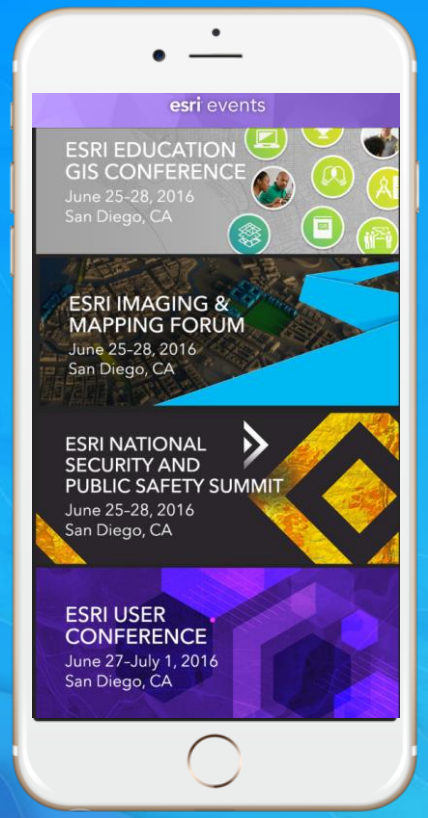


Access Native APIs

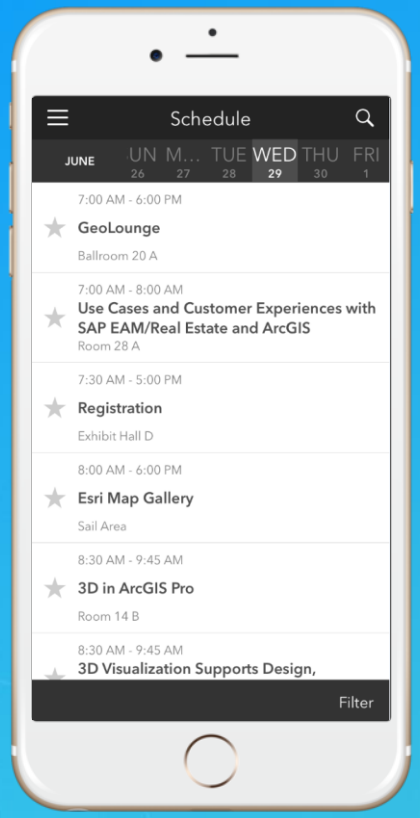
Please take our Survey

- **Your feedback allows us to help maintain high standards and to help presenters**

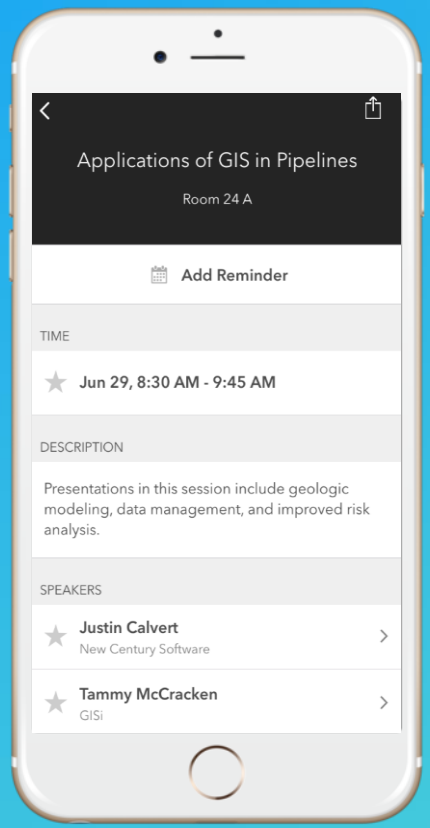
Find your event in the Esri Events App



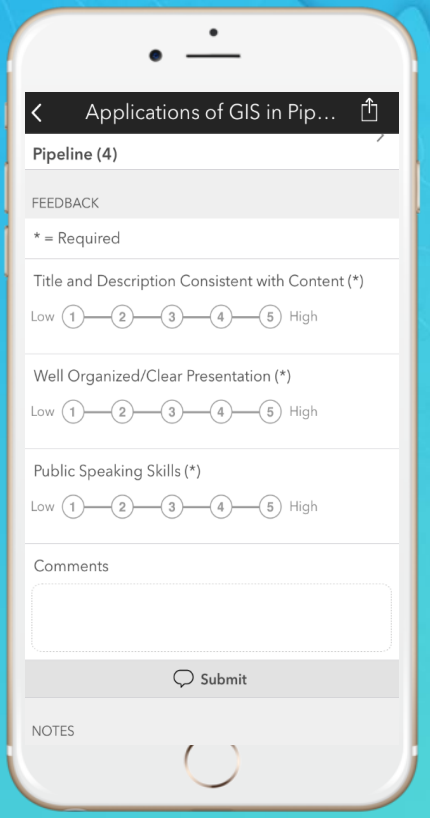
Find the session you want to review



Scroll down to the bottom of the session



Answer survey questions and submit




esri
THE
SCIENCE
OF
WHERE