ArcGIS Runtime:
Building Cross-Platform Apps

Mike Branscomb
Michael Tims
Tyler Schiewe
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
Tyler Schiewe

Qt/QML
Michael Tims

.NET/Xamarin
Mike Branscomb
Java
Tyler Schiewe
Cross platform Java Development

• “Write once, run anywhere”
• ArcGIS Java Runtime SDK is aimed at desktop platforms
• Sits on the ArcGIS Runtime core architecture (C++) via JNI
• JavaFX for building modern, native-looking GUIs
• Massive ecosystem of mature, open-source libraries to use
Development and Deployment

IDE

JAR file

Compile

Deploy

Windows

JRE

JAR file

Runtime Core

.dll file

Linux

JRE

JAR file

Runtime Core

.so file

Mac OS X

JRE

JAR file

Runtime Core

.dylib file

Development environment

Cross platform deployments
Cross platform Java Development

- Layout uses a native window
- Styling with CSS
- Optionally use markup for layout (FXML)

Be mindful of file paths
- C:\Users\Tyler\Projects\MyData.geodatabase
- /Users/Tyler/Projects/MyData.geodatabase

```xml
<StackPane fx:controller="com.esri.samples.mysample.SampleController"
    xmlns:fx="http://javafx.com/fxml" stylesheets="/css/style.css">
  <MapView fx:id="mapView"/>
  <HBox StackPane.alignment="TOP_CENTER" maxWidth="200" maxHeight="50" spacing="5" styleClass="panel-region">
    <Label text="Click me: "/>
    <Button fx:id="myButton" onAction="#myEvent"/>
  </HBox>
</StackPane>
```
javapackager -deploy -native -outdir packages -outfile SampleViewer
    -srcDir ./ -srcfiles java-se-sample-viewer-1.0.jar
    -srcfiles jniLibs -srcfiles resources -srcfiles samples-data
    -appclass com.esri.sampleviewer.App -name "SampleViewer" -title "Sample Viewer"
Summary

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

- **Cons**
  - Clients must have Java installed
  - Not targeted for mobile applications or web apps
  - “Verbose” syntax
The Qt Company – qt.io

- 20 Years in cross-platform development
- 1 Framework
  - Code once, run anywhere on any device
- Over 1 Million downloads of latest framework version
- 800,000 developers (indie, corporate) worldwide
- Open-source community
- Proven to speed time-to-market by 50% for cross-platform
<table>
<thead>
<tr>
<th>API</th>
<th>Windows</th>
<th>Linux</th>
<th>macOS</th>
<th>Android</th>
<th>iOS</th>
<th>WinUWP</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>QML</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

*Future Release*
1 SDK, 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
QML API
Example QML code

Highly readable JSON/CSS-like syntax

Declarative UI elements

Imperative JavaScript Code to handle events

ArcGIS Runtime

Dynamic property binding
Qt SDK – System setup
Make time to setup your builds

• Compiler, SDK dependencies
  - iOS: Xcode compiler
  - Windows: Visual Studio compiler
  - Linux: GCC compiler
  - Android: Android NDK and SDK

• Setup once, same code across platforms

• IDE: Qt Creator - Cross-platform IDE
  - Design, develop, test, deploy from one tool
Qt SDK – ArcGIS Extras
Extra APIs provided

• UI display scale factor
  - Qt 5.6 brings out-of-the-box High-DPI support

• Core APIs exposed to QML
  - File IO
  - Application settings
  - AppStudio's AppFramework also provides more functionality
Qt SDK – Coming soon…
APIs not currently available in Qt framework

- Push notifications
- Qt Speech (text to speech) – Tech Preview
- Background/suspended location updates – Qt 5.9!
- Access photo gallery on Android
Qt Quick Controls
Out-of-the-box controls for QML

- macOS

- Android – Nexus 5
Qt Quick Controls 2
Common UI controls for QML

- Material style (Google)
- Universal style (Windows)
Qt SDK – 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/licensing
Qt SDK – Pros and Cons

APIs not currently available in Qt framework

• Pros
  - Same modules
  - Same code
  - Same workflow
  - Same look and feel
  - Access to device sensors
  - Open source community

• Cons
  - Incomplete / initial APIs from Qt
  - Common abstraction API for iPhone and Android SDKs
  - Look and feel of the native platform
  - Qt framework can increase apps size
.NET / Xamarin
Mike Branscomb
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
  - Universal Windows Platform apps (UWP)
  - Xamarin.Android
  - Xamarin.iOS
  - Xamarin.Forms (Android, iOS, and 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

Internal

Win32

Windows Runtime

iOS

Android

OS

Public SDK

.NET Framework

.NET Core

Mono / Xamarin

ArcGIS Runtime SDK for .NET (C#)

ArcGIS Runtime Core (C++)

Interop

Xamarin Forms

Common UI

Native UI
  • MapView, SceneView

Common SDK
  • Maps, scenes, layers, symbols, feature data, geocoding, routing, portal, rasters, offline/syncing, geometry, geometry engine...

Where all the magic happens

- Win32
- .NET Framework
- .NET Core
- Mono / Xamarin
- UWP
- iOS
- Android
- WPF
- UWP
- iOS
- Android
- OS
- Win32
- .NET Framework
- .NET Core
- Mono / Xamarin
- WPF
- UWP
- iOS
- Android
- OS
ArcGIS Runtime SDK for .NET & Xamarin
A simpler view...

Operating System

- ArcGIS Runtime SDK for .NET (C#)
- Xamarin Forms
- .NET

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 API

Work Offline

Native User Experience

Access Native APIs
Please Take Our Survey on the **Esri Events App**!

Download the Esri Events app and find your event

Select the session you attended

Scroll down to find the survey

Complete Answers and Select “Submit”