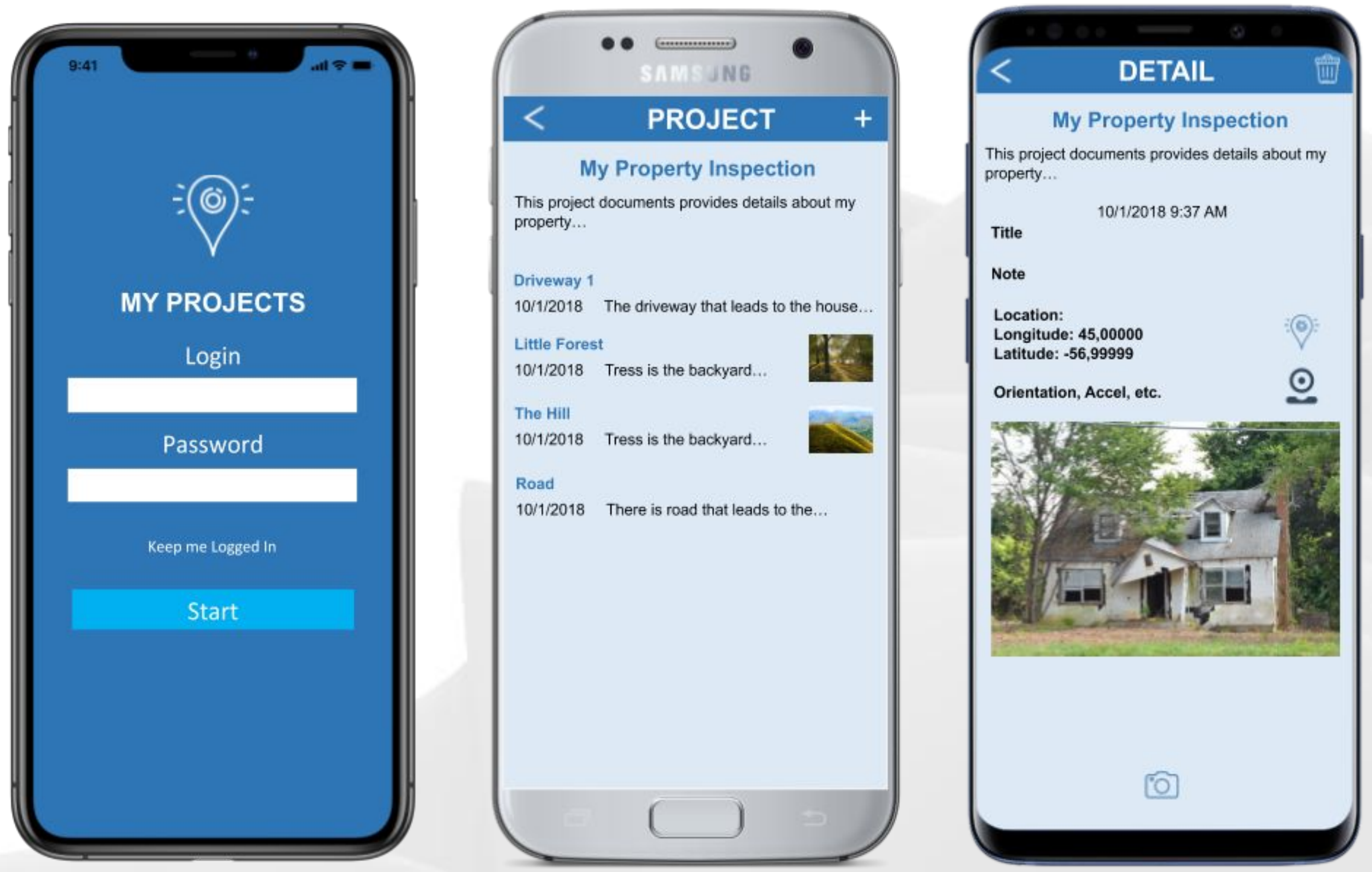
Mobile Development
with RAD Studio


MY PROJECTS
Login
Password
Keep me Logged In
Start
PROJECT
My Property Inspection
This project documents provides details about my property...
Driveway 1
10/1/2018 The driveway that leads to the house...
Little Forest
10/1/2018 Tress is the backyard...
The Hill
10/1/2018 Tress is the backyard...
Road
10/1/2018 There is road that leads to the...
DETAIL
My Property Inspection
This project documents provides details about my property...
10/1/2018 9:37 AM
Title
Note
Location:
Longitude: 45,00000
Latitude: -56,99999
Orientation, Accel, etc.

Stephen Ball
Pre-sales Director
Embarcadero Technologies
stephen.ball@embarcadero.com
@DelphiABall

embarcadero®

BLOG:
https://blogs.embarcadero.com/using-trestrequestdatasetadapter-to-update-data-via-a-restful-server-api/

Mobile Development with RAD Studio

- Mobile Development Landscape
- Building UI's for Mobile with FMX
    - UI Considerations for iOS and Android
    - Frames are cool!
    - Permissions
- App Store Deployment
- Q&A

Mobile Development with RAD Studio

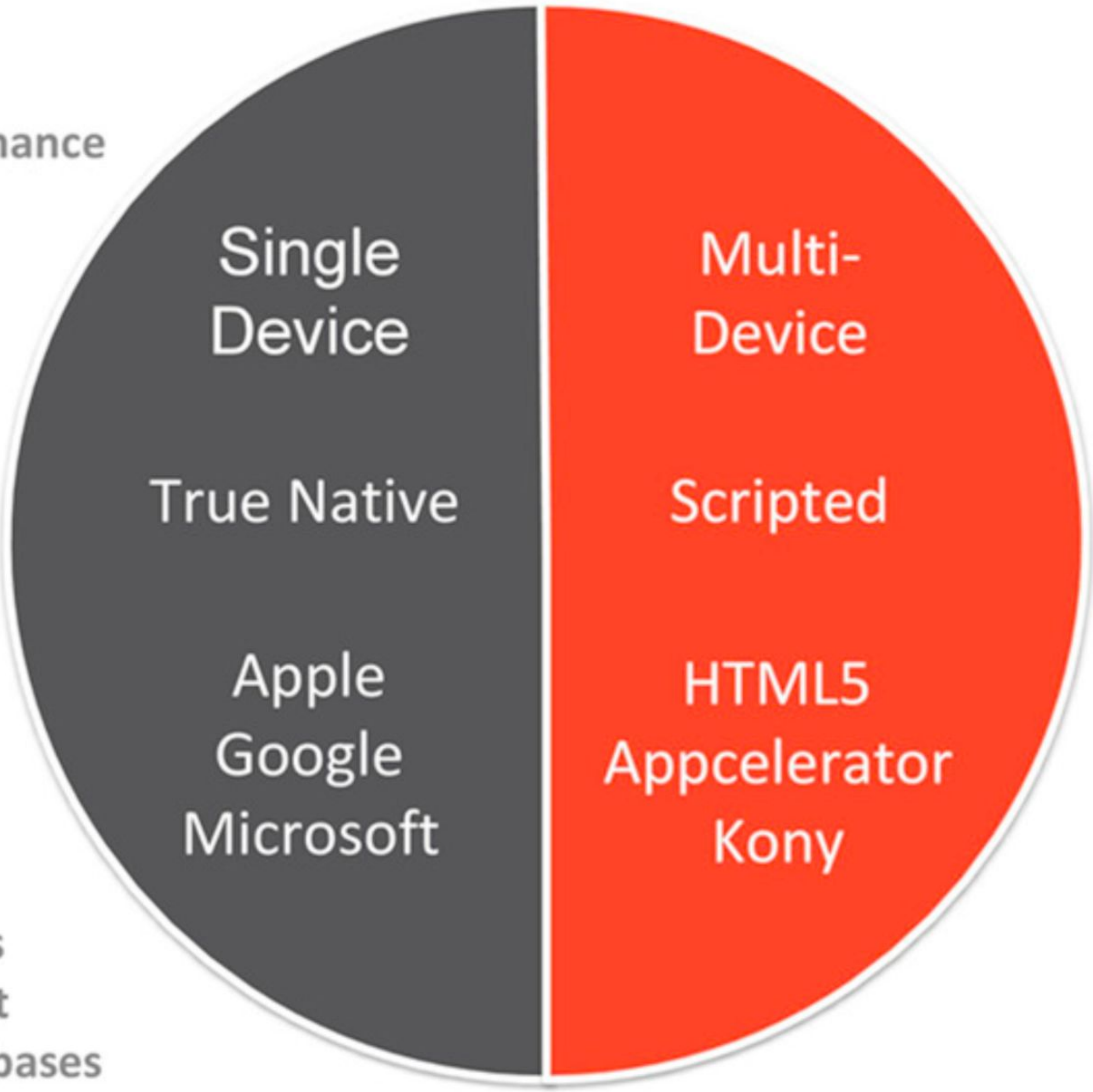
- Context
- Introduction to FireMonkey (FMX)
  - Objects / Interfaces / Components
  - Styles
  - LiveBindings & Data
- Building UI's for Mobile with FMX
  - UI Considerations for iOS and Android
  - Frames are cool!
  - LowCode Wizard
- App Store Deployment
- Q&A

Vendor Tools

Web Tech Based Tools

Pros
Native Performance
Native UX
Secure

Cons
Multiple teams
Higher dev cost
Multiple code bases

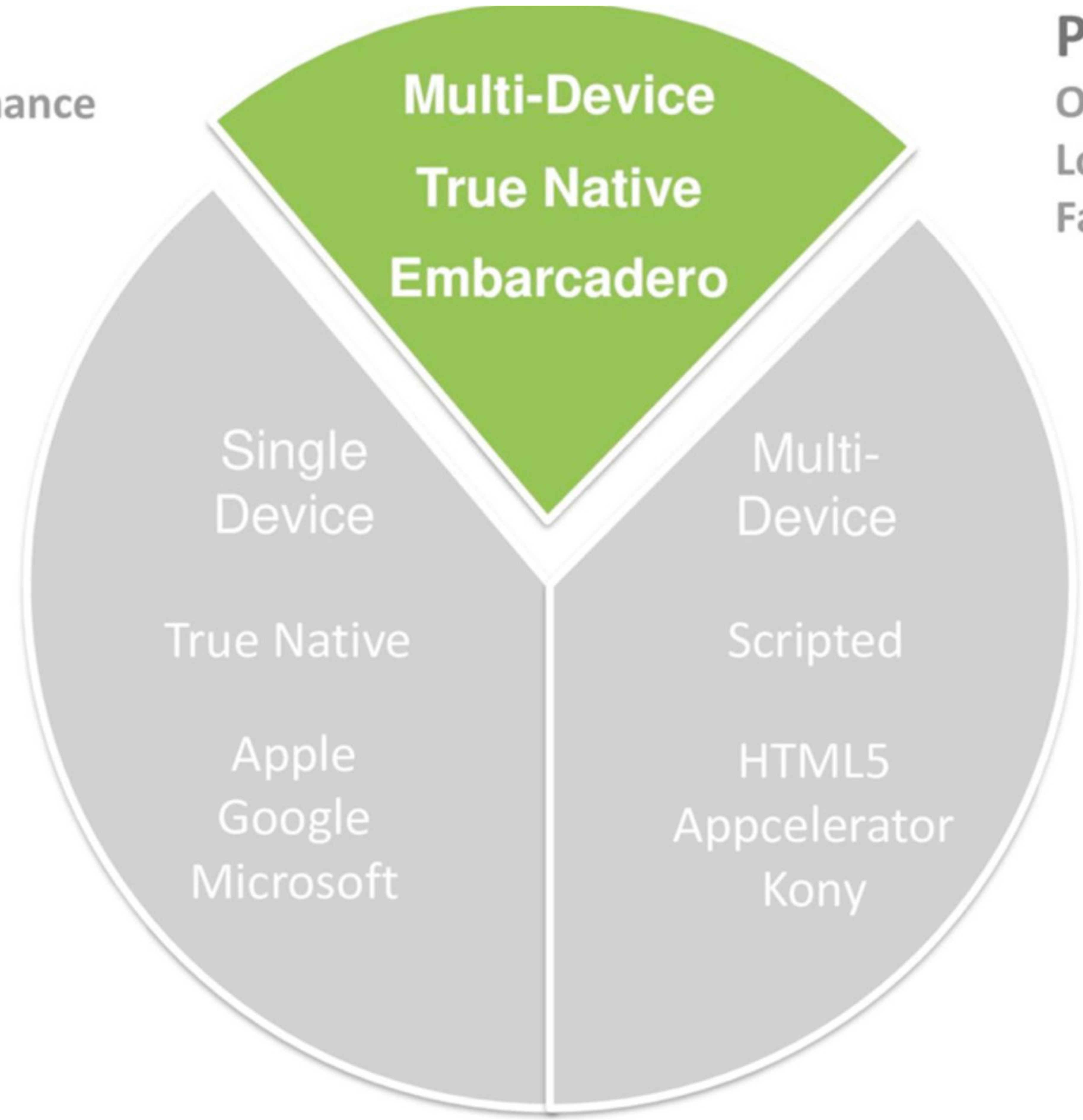


Pros
One team
Lower dev costs
Fast time to market

Cons
Non-Native UX
Unsecure
Script Performance

Pros

Native Performance
Native UX
Secure



Pros

One team
Lower dev costs
Fast time to market

Platform Vendor Tools

Apple iOS	Android	Windows
Info.plist	AndroidManifest.xml	Visual Studio Project
Swift or Objective-C Code	Java or Kotlin Code	C++ or C# Code
NIB File	Layout	XAML
Resources	Resources	Resources
Cocoa APIs	JNI & JDK Interface	Windows APIs

vs

RAD Studio

- Single Project file
- Pascal or C++ code
- FMX layout
  - Android Variant
  - iOS Variant
- Resources
- FMX & RTL Libraries
- Platform APIs

<u>One</u> Project
4 Platforms

RAD Studio

Single Project file

Pascal or C++ code

FMX layout

Android Variant

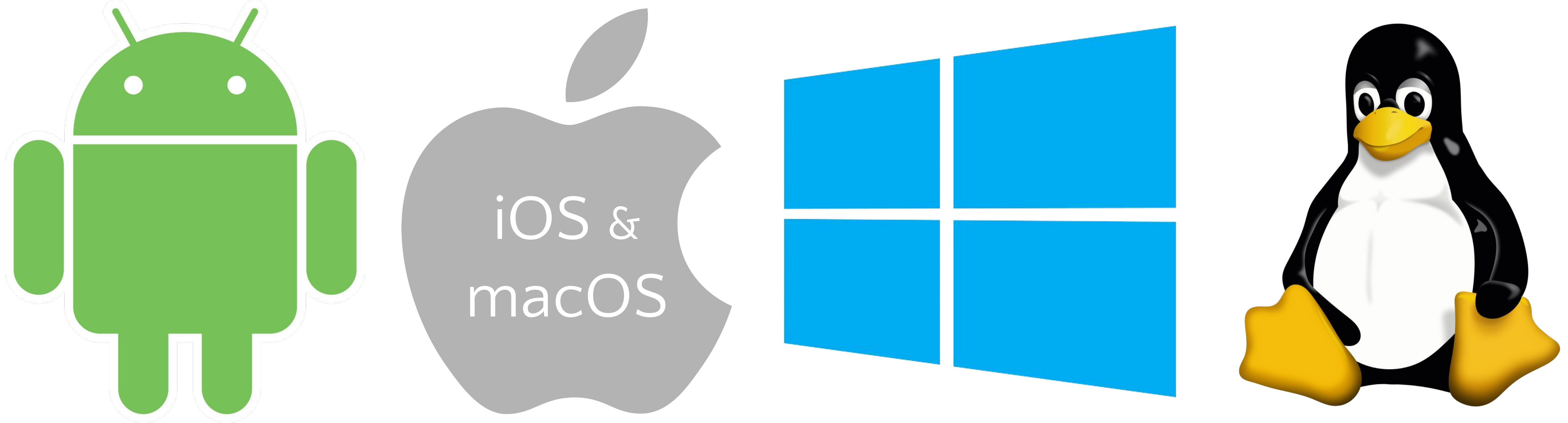
iOS Variant

Resources

FMX & RTL Libraries

Platform APIs

<u>One</u> Project
Every Platform



I need *how many* tools & languages?

Java for Android

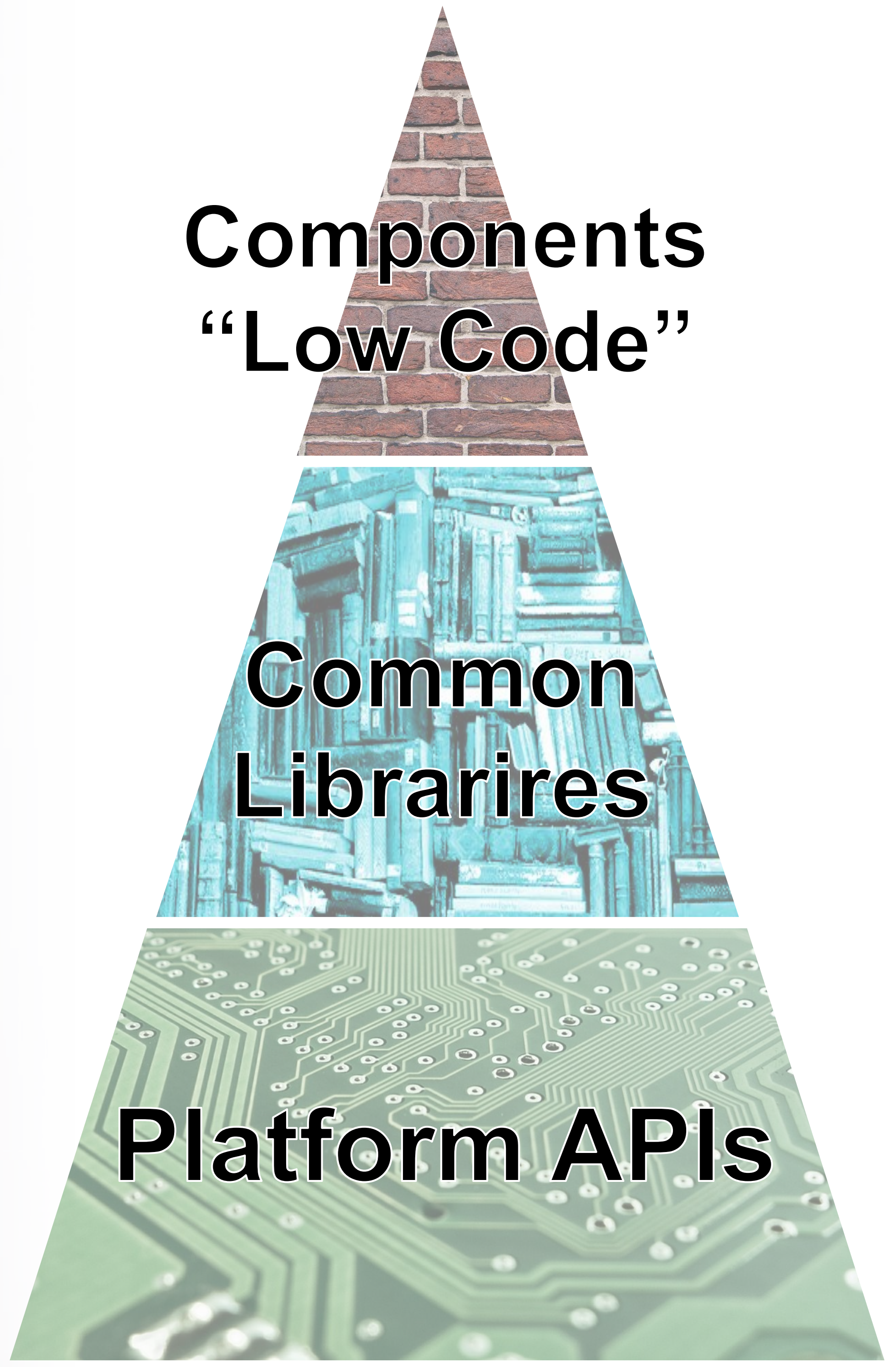
Swift for iOS & macOS

C# for Windows

That’s Where Embarcadero Comes In . .

- Embarcadero’s Delphi is a tool for the multi-device world
    - One project gives you native apps for all four platforms: Windows, iOS, Android, & macOS
    - Visual designer lets you design it once
    - Smart themes adapt the UI to correct platform behaviors
    - Create platform specific variations to tweak behavior on each platform if desired
    - Still with full access to platform APIs & 3rd party libraries

Supports all Three Layers of Development



- Each layer builds on previous
- Higher layers provide productivity benefits
- Lower layers provide more control of platform
- Apps typically written with a combination of layers
- Delphi gives you easy access to all layers!

No-Compromise Compiled Code

- Delphi compiles to native code, for Intel or ARM
  - Most compilers based on LLVM infrastructure
- No runtime needed
  - No dependency on execution environment (.NET, JVM)
  - No runtime compilation, JITer, or any other layer
- Advantages
  - Fast, secure, xcopy deployment (on desktop)

UI Widgets: Styled vs. Native or Both

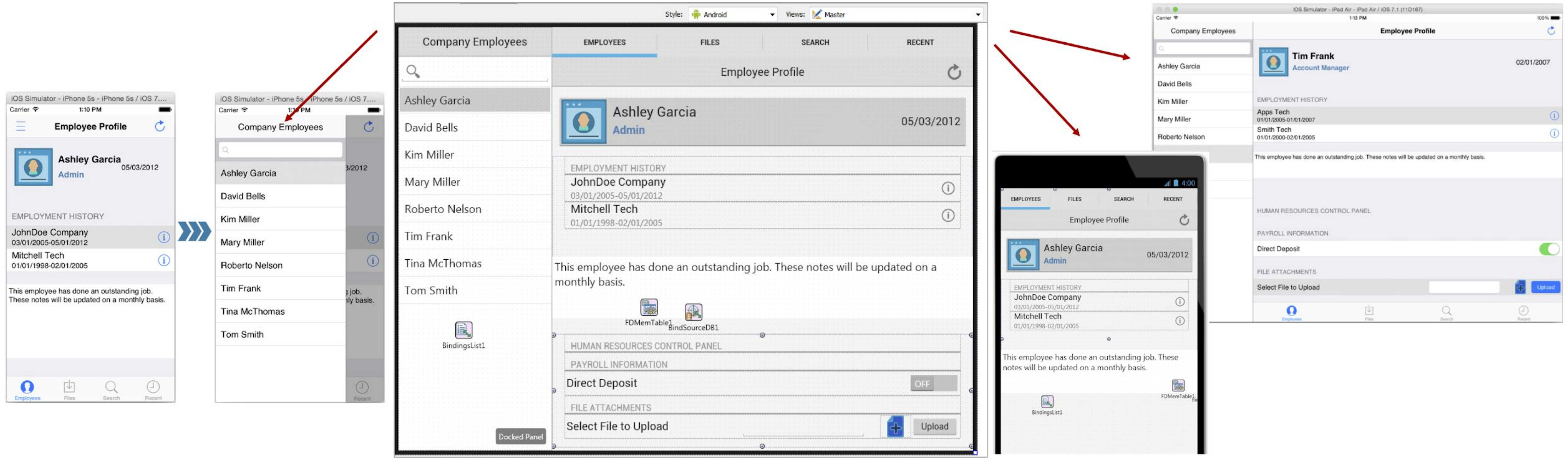
- Multi-platforms solutions offer
  - Styled (painted) controls only – often not even matching the platform
  - Native controls only – UI code changes significantly among platforms
  - “Forms” with a handful of controls

- FireMonkey
  - Extensive collection of styles controls, with platform specific styles (“pixel perfect”)
  - Key controls available as native controls (input controls and more)
  - Embedded complex native controls (maps, browser, etc.)

Best in Industry: FireUI Multi-Device Designer

Shared master and specific views

Visually customize forms for different platforms and form factors



Desktop First, Mobile Integrated

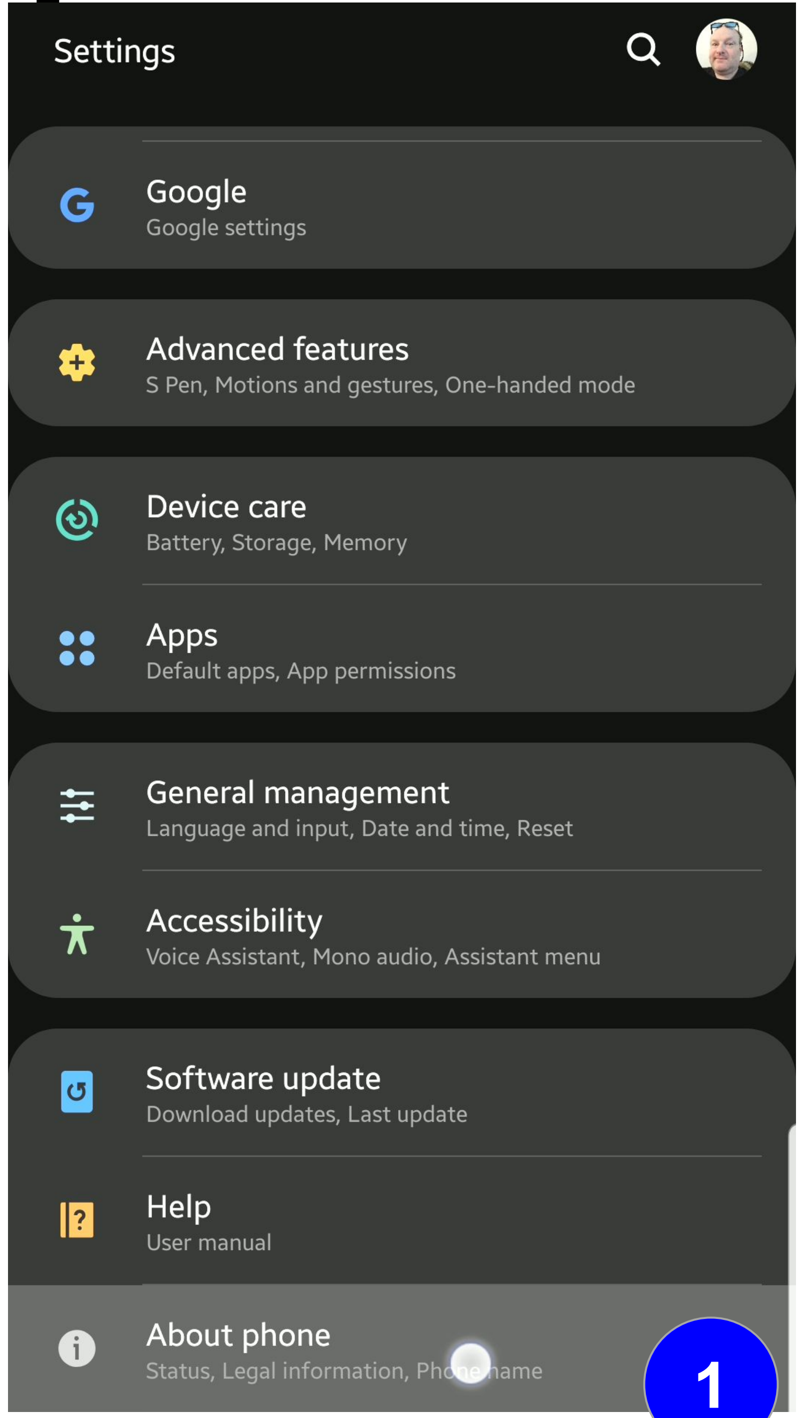
- A common scenario for business apps
- Keep the core business application on desktop
- Surface features on mobile with focused apps, notifications, etc
    - Ideally, use a shared back end
- Create companion apps via AppTethering
    - No server needed, just peer to peer to desktop app

Entering Developer Mode

- Enter Settings
1. Goto About Phone
2. Software Information
- Locate Build Number
3. Tap seven times
- Official instructions

https://developer.android.com/studio/debug/dev-options

- Some platforms are slightly different



Settings

Google
Google settings

Advanced features
S Pen, Motions and gestures, One-handed mode

Device care
Battery, Storage, Memory

Apps
Default apps, App permissions

General management
Language and input, Date and time, Reset

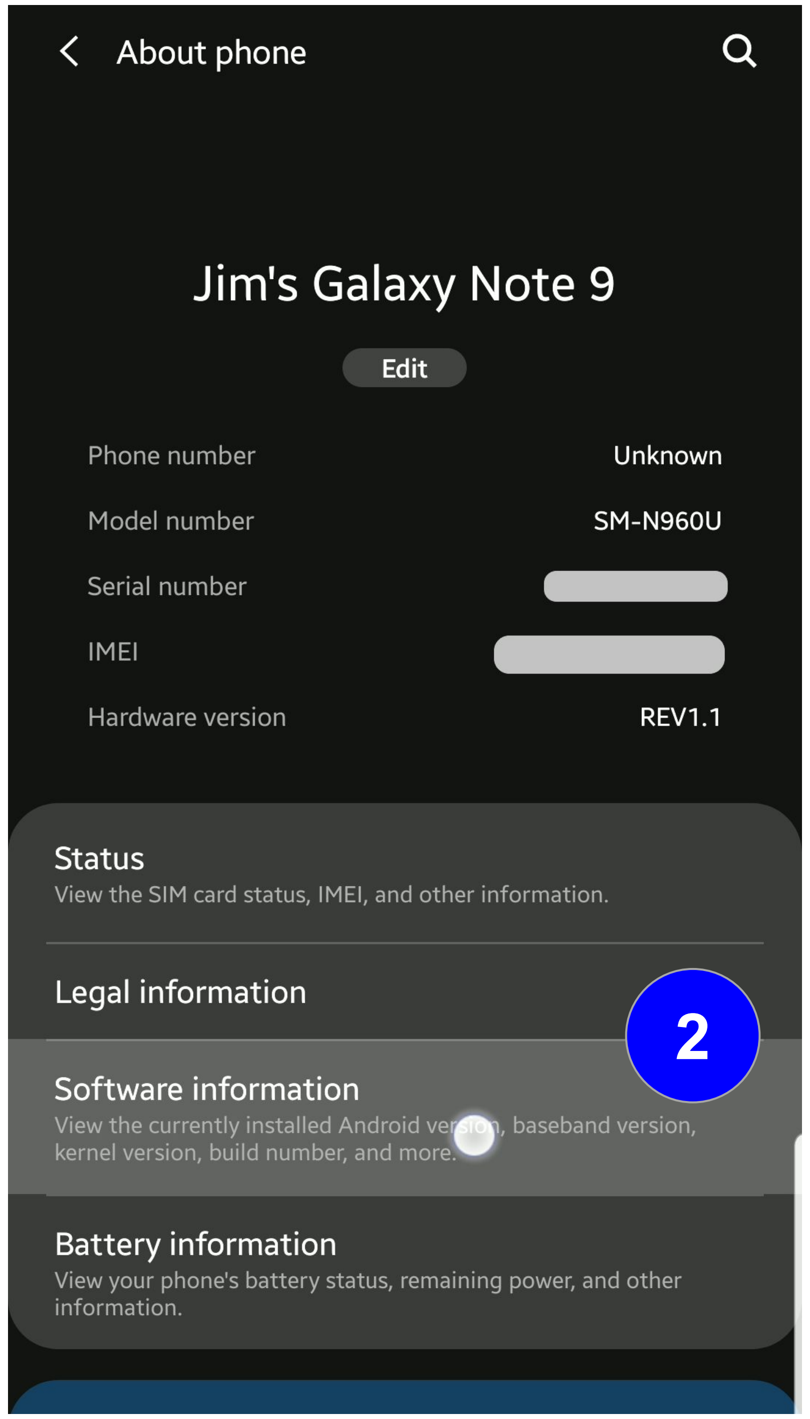
Accessibility
Voice Assistant, Mono audio, Assistant menu

Software update
Download updates, Last update

Help
User manual

About phone
Status, Legal information, Phone name

1



About phone

Jim's Galaxy Note 9

Edit

Phone number	Unknown
Model number	SM-N960U
Serial number	[redacted]
IMEI	[redacted]
Hardware version	REV1.1

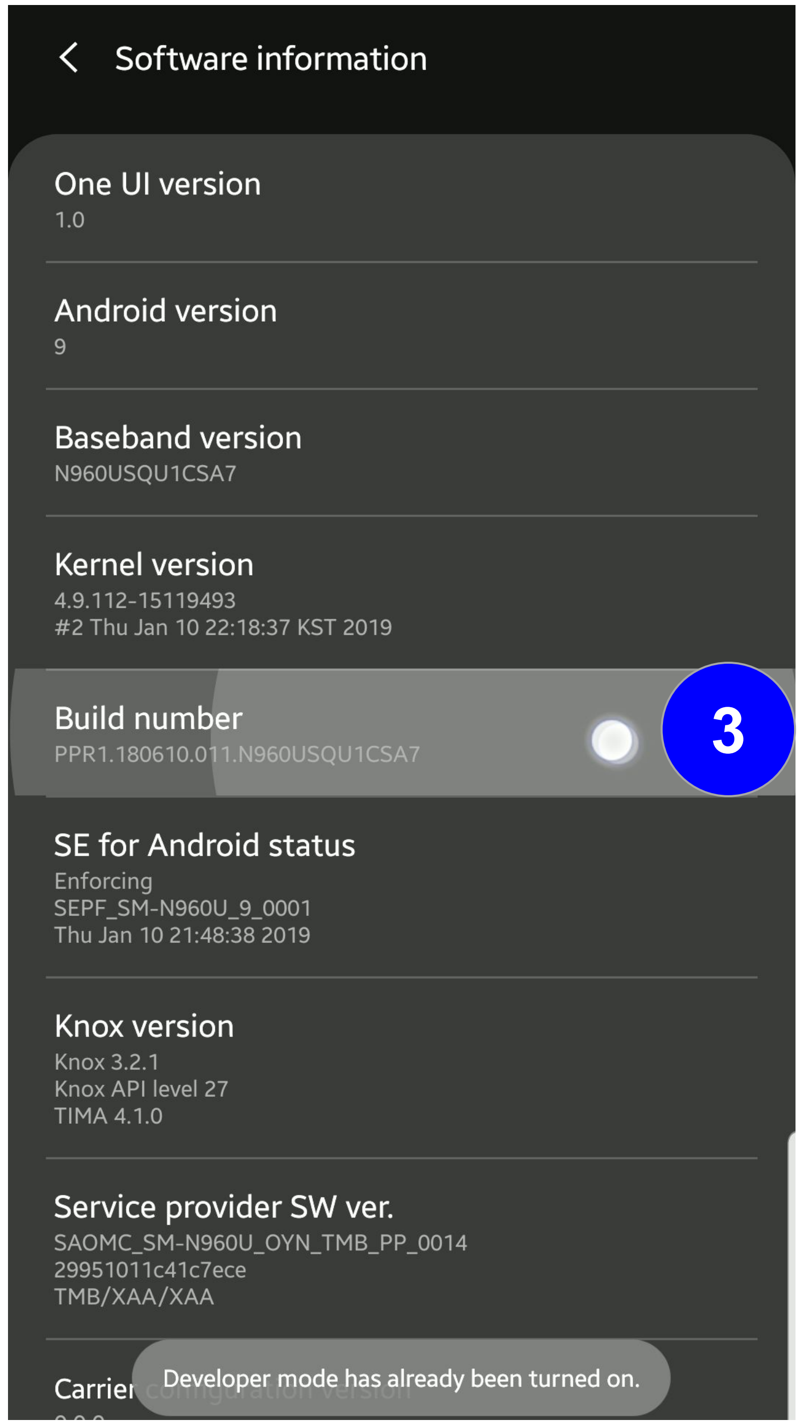
Status
View the SIM card status, IMEI, and other information.

Legal information

2

Software information
View the currently installed Android version, baseband version, kernel version, build number, and more.

Battery information
View your phone's battery status, remaining power, and other information.



Software information

One UI version
1.0

Android version
9

Baseband version
N960USQU1CSA7

Kernel version
4.9.112-15119493
#2 Thu Jan 10 22:18:37 KST 2019

Build number
PPR1.180610.011.N960USQU1CSA7

3

SE for Android status
Enforcing
SEPF_SM-N960U_9_0001
Thu Jan 10 21:48:38 2019

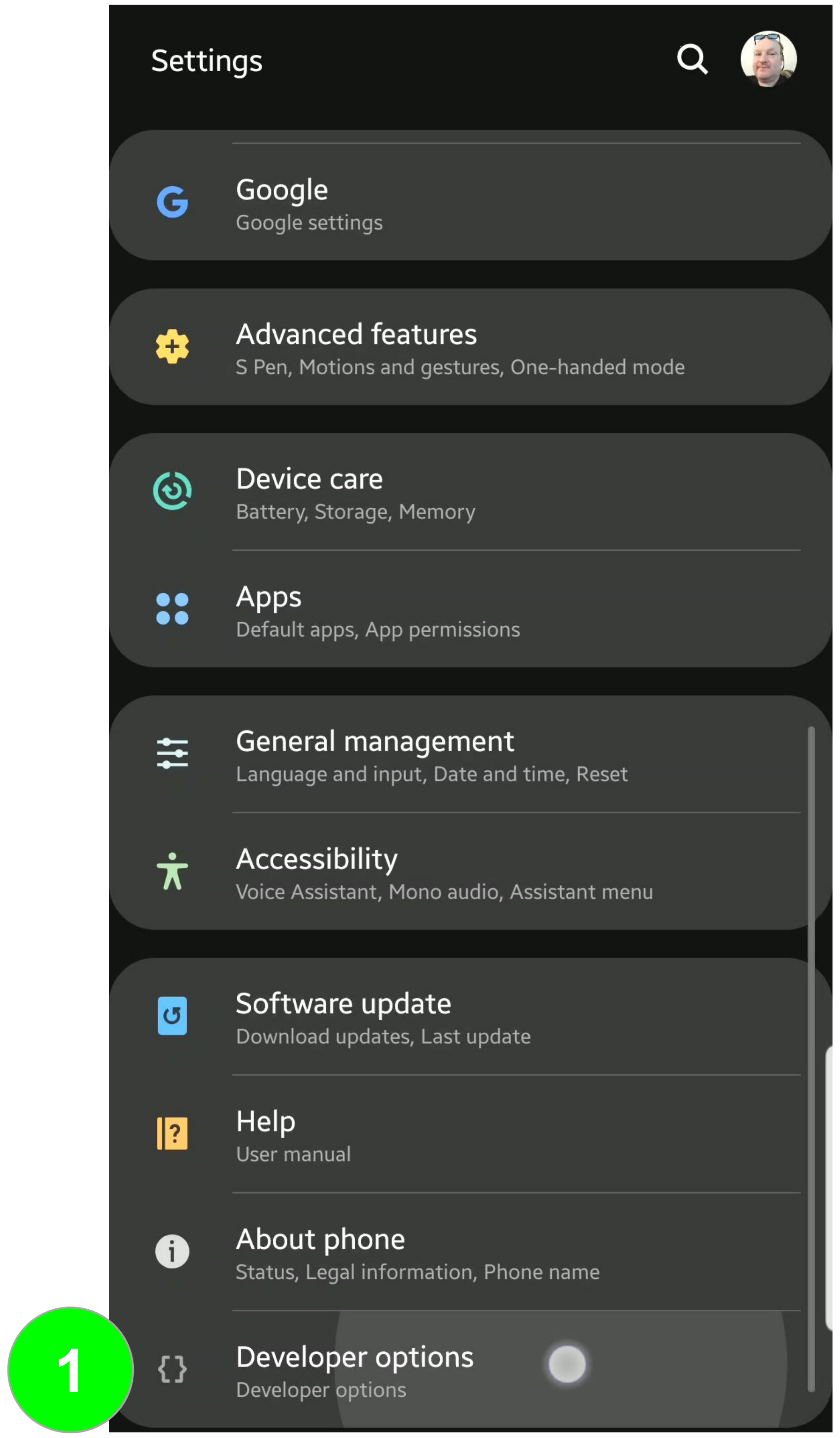
Knox version
Knox 3.2.1
Knox API level 27
TIMA 4.1.0

Service provider SW ver.
SAOMC_SM-N960U_OYN_TMB_PP_0014
29951011c41c7ece
TMB/XAA/XAA

Carrier

Developer mode has already been turned on.

Enable USB Debugging



Settings

Google
Google settings

Advanced features
S Pen, Motions and gestures, One-handed mode

Device care
Battery, Storage, Memory

Apps
Default apps, App permissions

General management
Language and input, Date and time, Reset

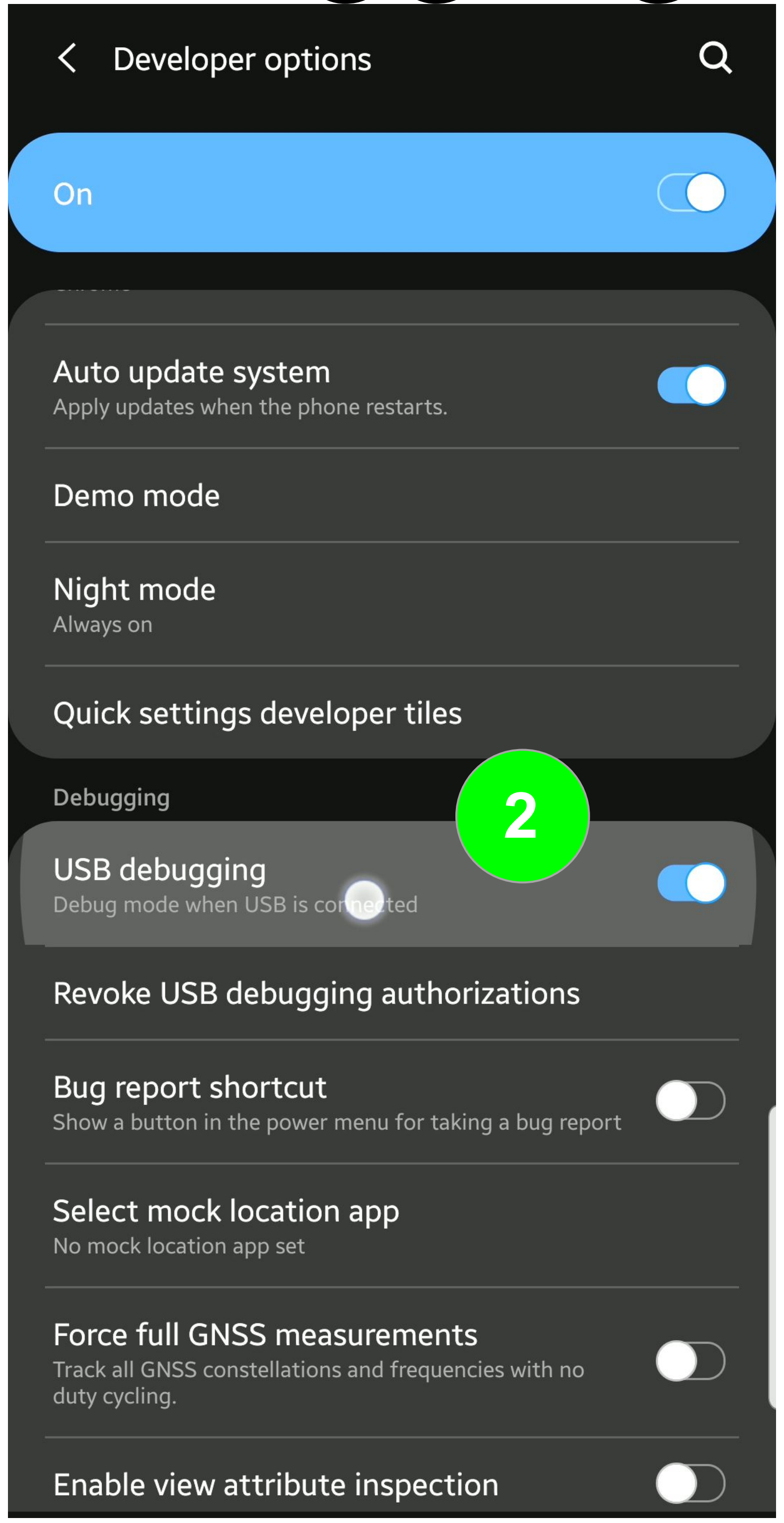
Accessibility
Voice Assistant, Mono audio, Assistant menu

Software update
Download updates, Last update

Help
User manual

About phone
Status, Legal information, Phone name

1 Developer options
Developer options



Developer options

On

Auto update system
Apply updates when the phone restarts.

Demo mode

Night mode
Always on

Quick settings developer tiles

Debugging 2

USB debugging
Debug mode when USB is connected

Revoke USB debugging authorizations

Bug report shortcut
Show a button in the power menu for taking a bug report

Select mock location app
No mock location app set

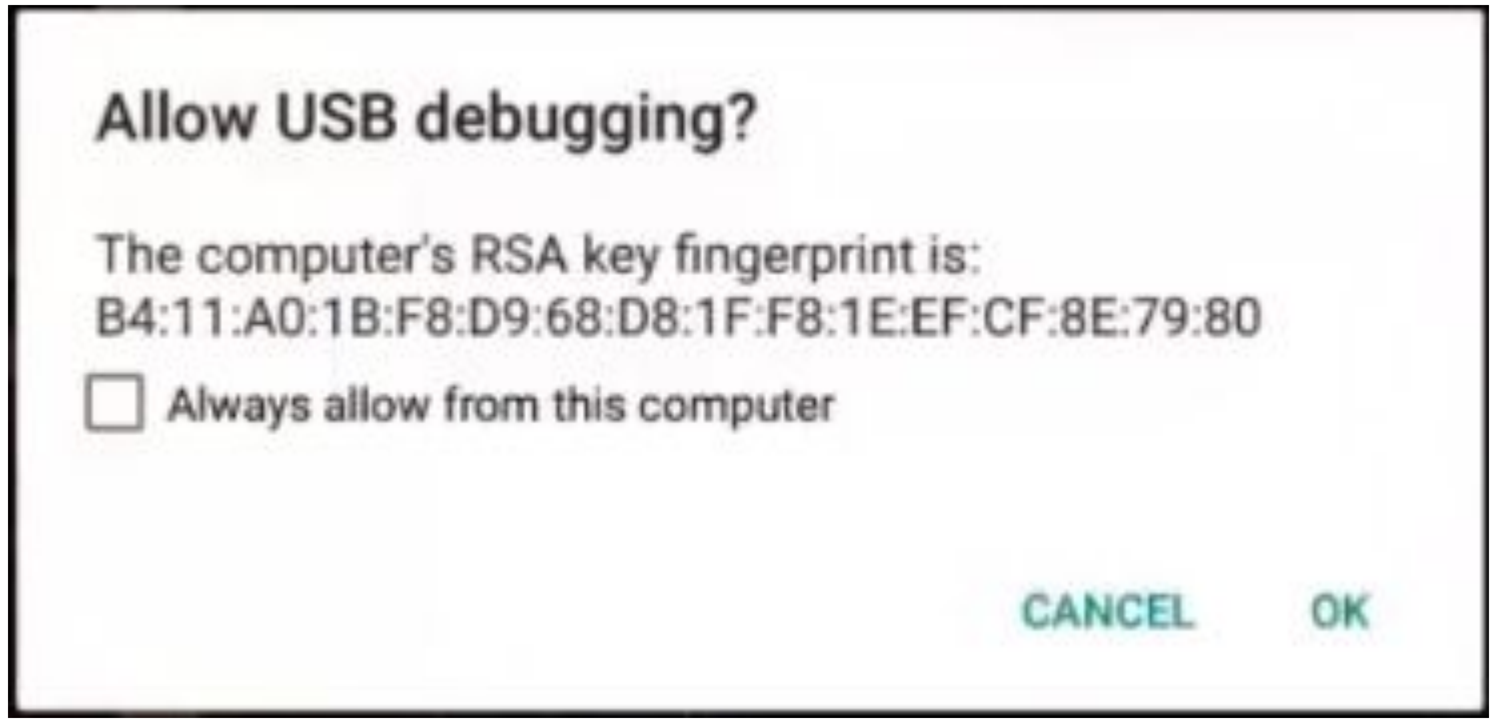
Force full GNSS measurements
Track all GNSS constellations and frequencies with no duty cycling.

Enable view attribute inspection

- After entering developer mode
- Go into Settings
- 1 Enter Developer options
- Scroll down to USB debugging
- 2 Enable USB debugging
- Confirm choice

Set up the Environment (ADB)

When you connect your device, you'll see a dialog similar to this on the device screen. Check the option to "always allow from this computer" and clock "OK"



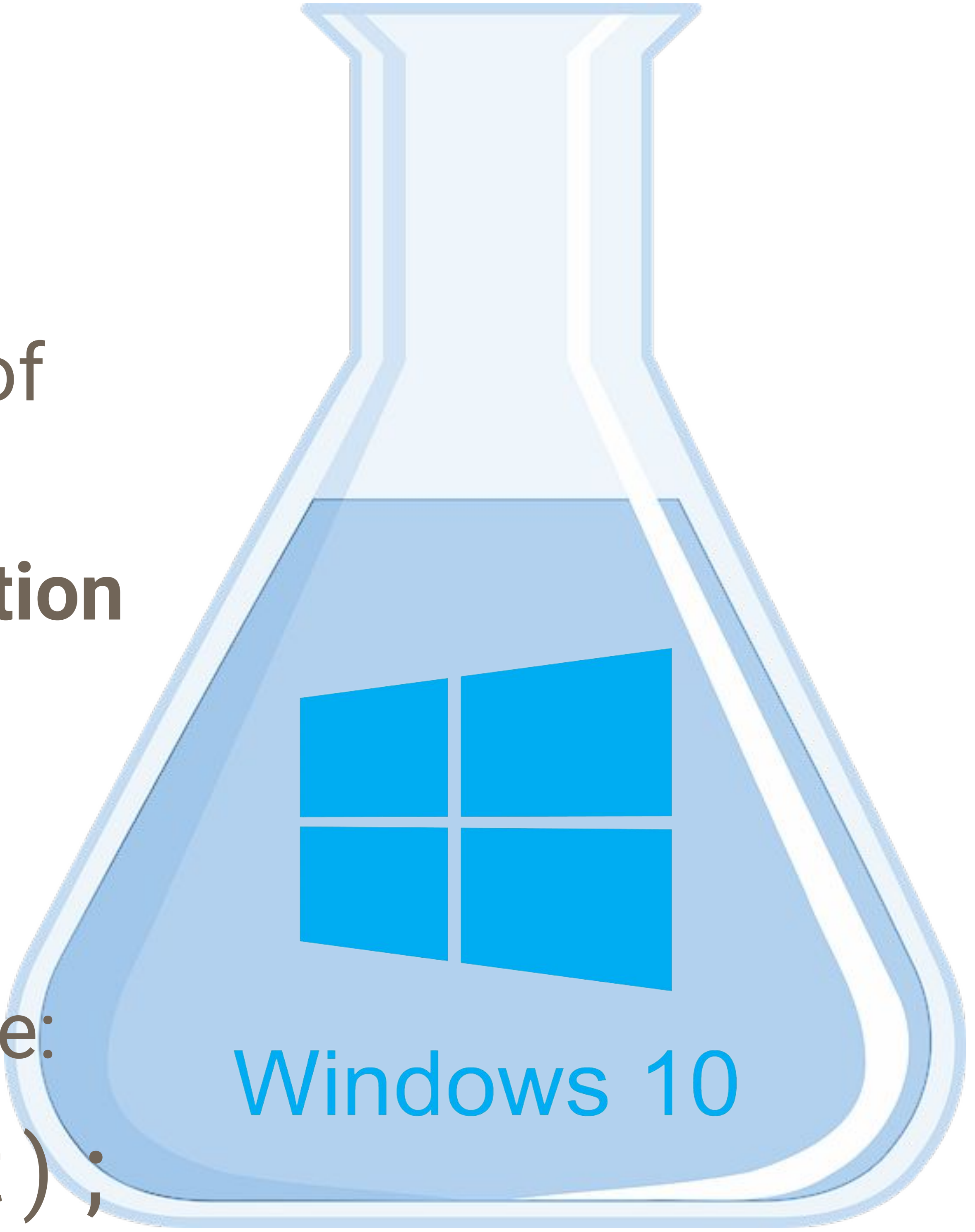
Allow USB debugging?

The computer's RSA key fingerprint is:
B4:11:A0:1B:F8:D9:68:D8:1F:F8:1E:EF:CF:8E:79:80

☐ Always allow from this computer

Lab 1 – Hello FireMonkey

- Create a FireMonkey app for Windows
- Add a TTabControl, then right click and "Add TTabItem"
- Include TLabel, TButton, TEdit, & TListBox on one of the TTabItems
- Notice TLabel & TButton have **Text** instead of **Caption** properties
- Notice the TTabControl has TabPosition of PlatformDefault
- In the OnClick event handler add the following code:
  - `ListBox1.Items.Add(Edit1.Text);`
- Compile and run your app on Windows

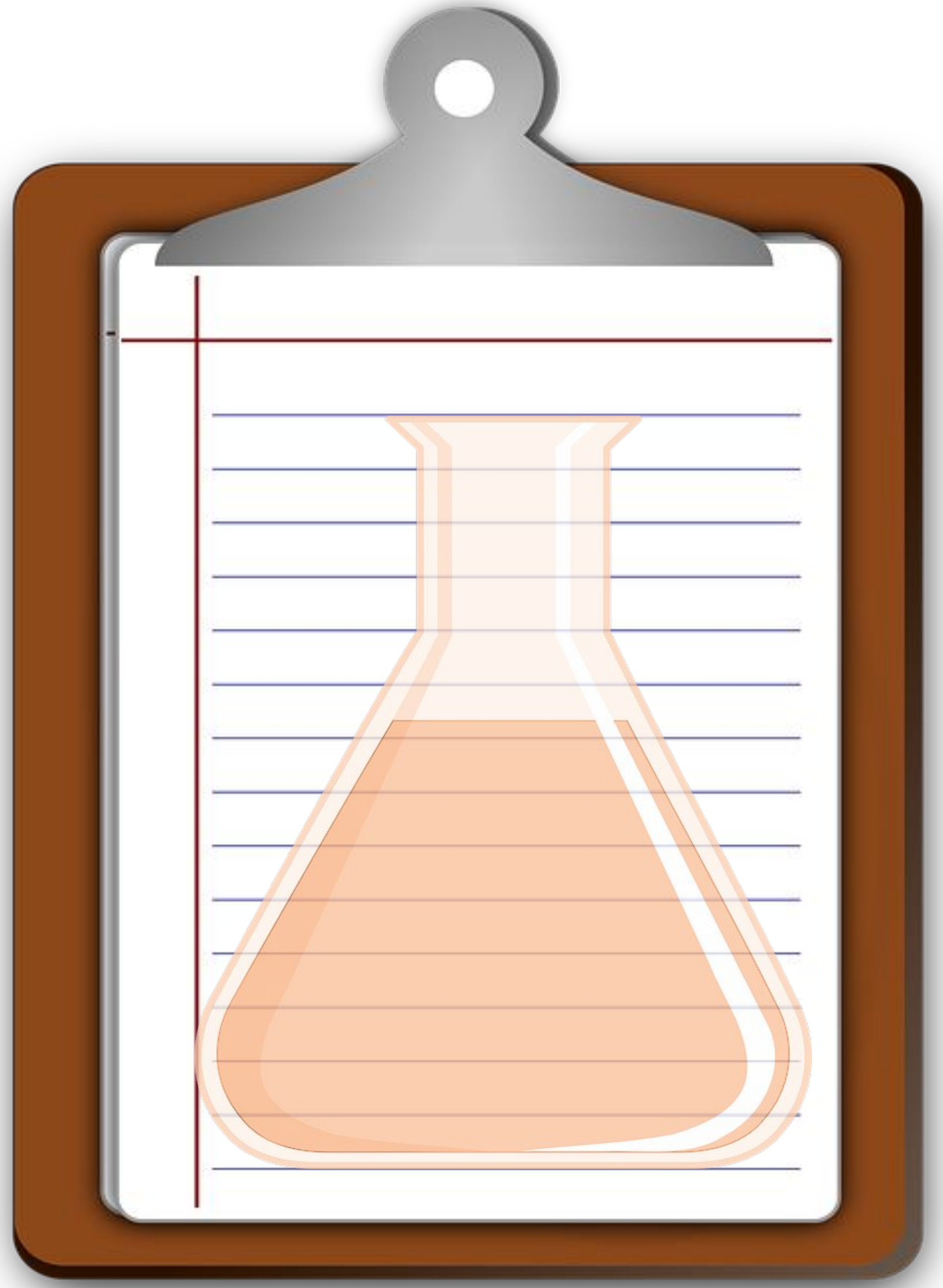


Beaker image https://pixabay.com/photo-23417/

Lab 2 – FireMonkey Platform Services

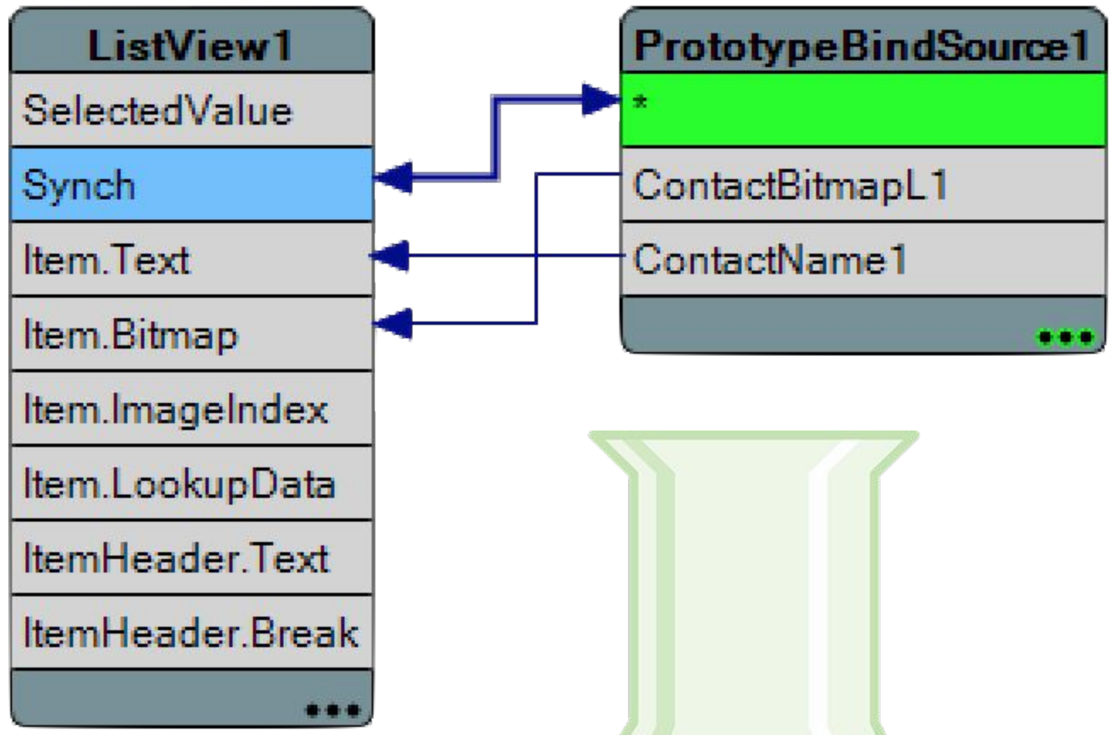
- Working with your lab from earlier, add Clipboard support
- Take a look at http://embt.co/PlatformServices
- Add FMX.Platform & FMX.Clipboard to your uses clause
- Interact with the clipboard via Platform Services

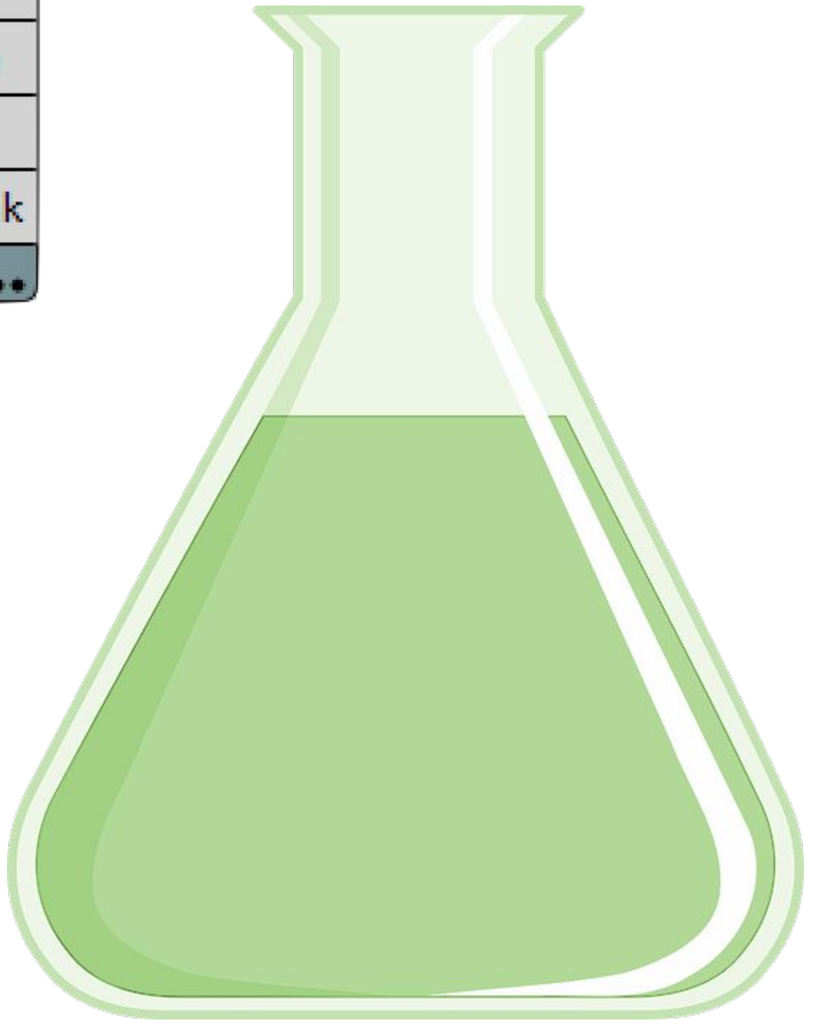
```
uses
  FMX.Platform, FMX.Clipboard;

procedure TFormFMXLab2.Button2Click(Sender: TObject);
var
  ClipboardService: IFMXExtendedClipboardService;
begin
  if TPlatformServices.Current.SupportsPlatformService(
   IFMXExtendedClipboardService, ClipboardService)
  then
  begin
    if ClipboardService.HasText then
      ListBox1.Items.Add(ClipboardService.GetText);
    ClipboardService.SetText('Hello FireMonkey');
  end;
end;
```


Clipboard image https://pixabay.com/photo-155885/

Lab 3 – FireMonkey LiveBindings

- Add a TPrototypeBindSource
  - Dbl Click -> Add ContactBitmapsL & ContactNames
- Add TListView
  - Find the ItemAppearance.ItemAppearance property
    - Change it from ListItem to ImageListItem
  - Set ItemAppearanceObjects.ItemObjects.Visible to False
- Right Click ListView1 -> Bind Visually…
  - Connect:
    - Sync -> *
    - Item.Text -> ContactName1
    - Item.Bitmap -> ContactBitmatL1



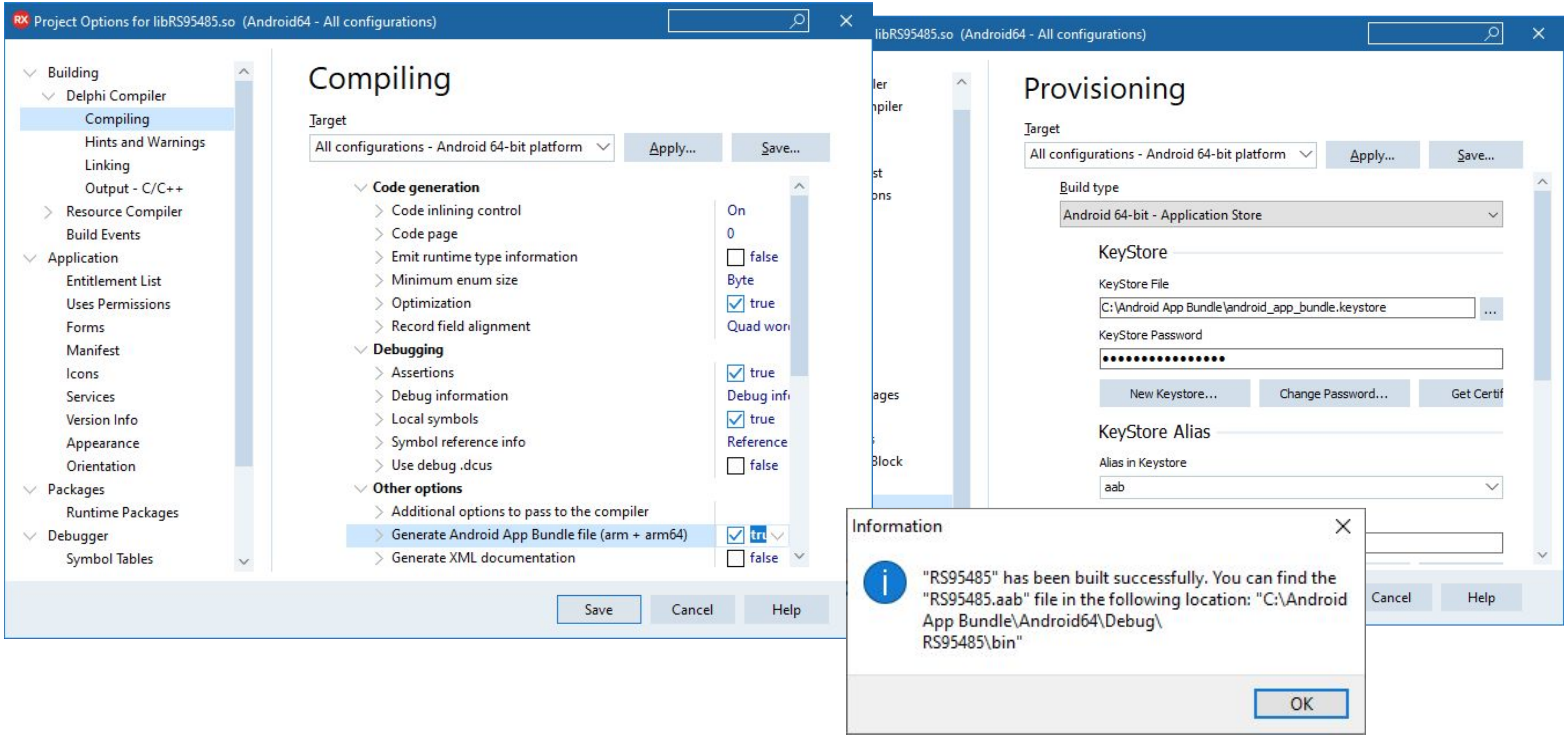


AppStore

Key Points

BLOG:
https://blogs.embarcadero.com/using-trestrequestdatasetadapter-to-update-data-via-a-restful-server-api

AAB Steps



Compiling

Target: All configurations - Android 64-bit platform

Code generation	
Code inlining control	On
Code page	0
Emit runtime type information	false
Minimum enum size	Byte
Optimization	✓ true
Record field alignment	Quad wor
Debugging	
Assertions	✓ true
Debug information	Debug inf
Local symbols	✓ true
Symbol reference info	Reference
Use debug .dcus	false
Other options	
Additional options to pass to the compiler	
Generate Android App Bundle file (arm + arm64)	✓ tru
Generate XML documentation	false

Provisioning

Target: All configurations - Android 64-bit platform

Build type: Android 64-bit - Application Store

KeyStore

KeyStore File: C:\Android App Bundle\android_app_bundle.keystore

KeyStore Password: ••••••••••••••••

KeyStore Alias

Alias in Keystore: aab

Information

"RS95485" has been built successfully. You can find the "RS95485.aab" file in the following location: "C:\Android App Bundle\Android64\Debug\RS95485\bin"

Mobile Development With RAD Studio

Q&A

Stephen Ball
Pre-sales Director
Embarcadero Technologies
stephen.ball@embarcadero.com
@DelphiABall

embarcadero®

Handout

Save slide as PDF

Getting Mobile with RAD Studio

RAD Studio

Why RAD Studio for Mobile Development?

Market trend data clearly shows Mobile is now the dominant platform for end-user engagement, with Android the biggest platform on the market today. It is, therefore, more important than ever to have a mobile strategy in place as part of a complete software vision. When looking to develop a mobile application, a number of key considerations come into play, including **User Experience, Data Security**, and **Total Development Costs** (Covered in depth in the White Paper – RAD Studio Guide For Managers). Delphi is the only language that answers all these challenges (without compromise) by enabling you to build fully compiled **true native applications**, from a **single code base** for all platforms. Delphi achieves this using the same compiler as Xcode on iOS, and by compiling to the NDK on Android (same layer used by games developers for speed and performance), with mature run-time libraries based on modern OOP practices, allowing you to reach all platforms at the same time, everytime, with the lowest total cost of ownership, the fastest speed in development, and strongest security in place.

Install & Initial Setup

You can download a free 30 day trial for RAD Studio Architect from https://www.embarcadero.com/products/rad-studio/start-for-free
Registering for the trial will send you an email with your trial key and will start the download

- The Install notes on DocWiki.Embarcadero.com give extra information. (such as minimum spec)
- Recommend you add Android and iOS as Target Platforms during installation. Along with the Java SDK. Alternatively you can add **Target Platforms**, **Help**, **Samples**, etc by opening ***Tools > Manage Platforms*** in the IDE.

[VIDEO] IDE Setup and First Android App **https://www.youtube.com/watch?v=39IplWi0oal**

- Adding Platforms
- Checking SDK's
- Connecting your Device
- Deploying and debugging a Hello World App.

[VIDEO] Setup and Deploy of first iOS App **https://www.youtube.com/watch?v=Pgg4009SCEk**

- Checking Platforms
- Introduction to PAServer ready to add SDK's
- Connecting your Device
- Deploying and debugging a Hello World App.

Videos

Other popular Getting Started videos | **Mobile Weather REST Client** | **Mobile App with local data storage** | **Building & deploying your first native mobile apps for iOS and Android**

Platform Requirements

Android : The same machine (or VM) your IDE is installed in will do (Windows 10). You also need to ensure the JDK and SDK are installed for Android. (check ***Tools > Manage Platforms > Extras***). if you manually install the SDKs, please check pathings in ***Tools > Options > Deployment > SDK Manager***

iOS : To deploy onto iOS hardware, you need to be enrolled in the Apple Developer Program and have access to Apple hardware (legal thing in the Apple EULAs for app signing). RAD Studio uses PAServer as the bridge to the host mac to enable the device to run and debug from the IDE.

Low Code Wizard

- **Low code Wizard** : Under ***Tools > GetIt Package Manager*** Install the "FireMonkey App Low Code Wizard". Then *File > New > Other > Delphi > Multi-Device >* ***FireMonkey Template App.***
- This wizard generates a outlined project with frames, unit test, and best practice MVC examples.
- Alternatively use **File > New > Multi-device Application** - to create a blank project.
- **Top Tip** - As you see from the Low Code Wizard, Frames can make mobile app development a lot easier. Check out TFrameStand in GetIt

Frameworks and Libraries

- **FireMonkey (FMX)** is the UI framework used for visually designing stunning cross platform apps.
- FMX uniquely enables developers to choose native or styled controls for ultimate platform fidelity.
- Control styles default to the platform look and feel for the best, expected, user experience, but can also be overridden for a truly customized design by using a single "StyleBook" component..
- **FMX** supports styles & **[Video] 60+ no code shader effects**. (such as Glow Effects to highlight required fields to end users)
- Core multi-platform runtime library include Database access library (FireDAC) and core System libraries e.g. Generics, Parallel Programming Library, JSON, IOUtils, Regular Expressions NetEncoding and more.
- **[Video] Language Features and RTL libraries introduction -** http://bit.ly/WelcomeToDelphi
- **Checkout the Samples folder for examples of UI Controls, Sensors, and Frameworks in action.**

Prototype Faster!

- To support differences in platform standards (e.g. Tabs at Top on Android and bottom on iOS) FMX controls includes options for setting values to "Platform Default".
- **On Device Live Preview with FireUI** - RAD Studio enables you to view your form, with prototype data as you build the UI, and how it looks differently on different platforms. Checkout FireUI
- InterBase + FireDAC give you have a highly powerful cross platform database that works across the development cycle on Windows, Linux, macOS, iOS & Android.
- InterBase ToGo run time license is Included in Enterprise and Architect editions, and includes the IoT Award winning Change Views, for tracking field level changes in the data layer, enabling developers to build apps faster with local data storage they can sync remotely when back online.

Need Advice?

Embarcadero pre-sales engineers are available to discuss your project requirements and help point you in the right direction. Ask your local sales rep for an appointment..

@EmbarcaderoTech