Zebra DevCon 2023

ZEBRA

Choosing the best development environment

Nicola De Zolt L.

Software Engineer – Zebra Technologies

Choosing the best development environment

Agenda

Zebra DevCon 2023

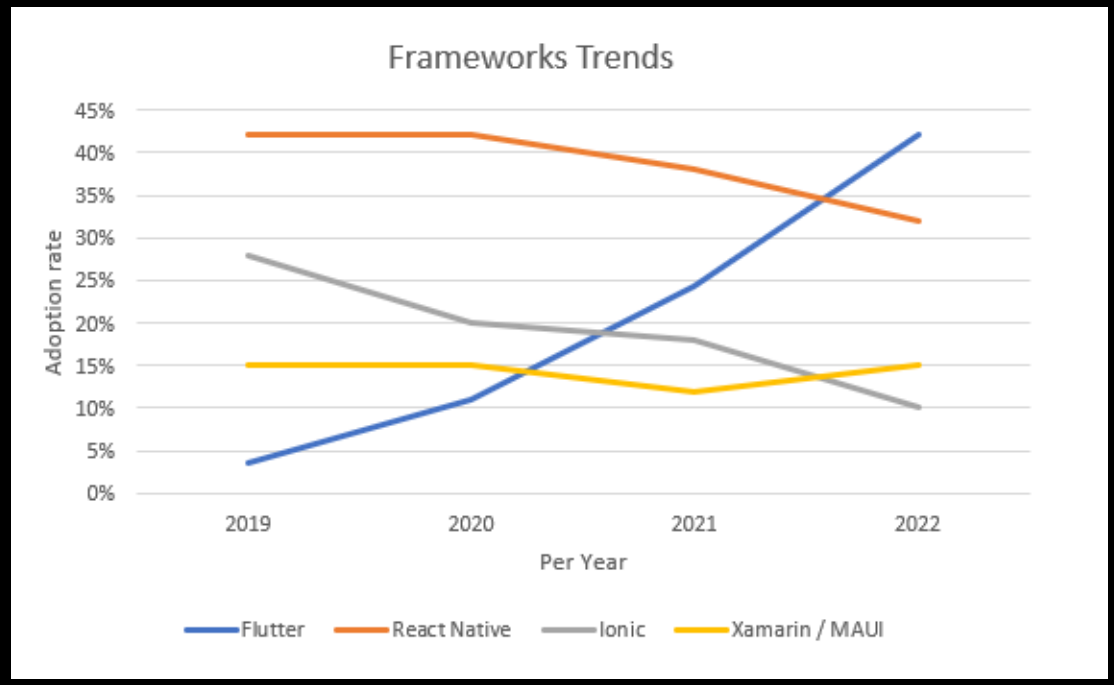
- Why look at cross-platform frameworks?
- Zebra Enterprise Browser
- Flutter
- Ionic
- React Native
- .NET MAUI
- Frameworks comparisons
- Q&A

Why look at cross-platform frameworks?

Reasons & Trends

Zebra DevCon 2023

- Reduced development costs
  - A single codebase that can be used to build apps for multiple platforms. Saves time and money
- Increased development speed
  - You only need to learn one codebase
  - Reuse code and components across different projects
- Reach a wider audience
  - Your app is available on multiple platforms
  - Great way to grow business and increase profits
- Improved user experience
  - Use the native features of each platform to create a more seamless and intuitive experience for your users
- Reduced maintenance costs
  - Maintain a single codebase
  - Use automated tools to update your app for different platforms



Framework	Adoption (2023) (*)	Trend
Flutter	~45%	Skyrocketing
React Native	~40%	Slightly decreasing
Xamarin / MAUI	~15%	Stable
Ionic	~10%	Declined

(*) Elaborated data from Statista and StackOverflow sources

Why look at cross-platform frameworks?

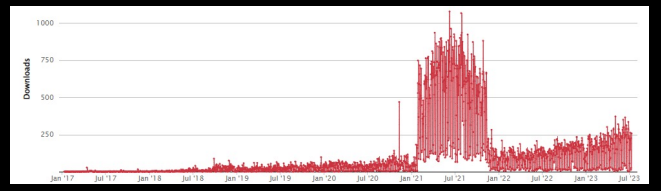
Indoors / Outdoors – The Zebra perspective

Zebra DevCon 2023

- Traditional Zebra mobile computers tools for developers
  - Library-based SDKs: EMDK
  - Intent APIs: Datawedge (DW), Workstation Connect (ZWC)
  - Javascript: Enterprise Browser (EB)

- Targeting
  - Java
  - Xamarin and .NET MAUI (npm, 277k)
  - Zebra Enterprise Browser
  - Any other environment supporting Intents

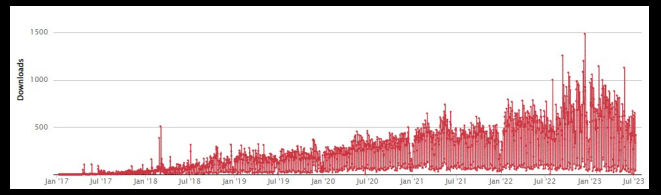
- Community trends
  - React Native *Datawedge* plugin: npm, 244k



  - Flutter *Datawedge* plugin: pub.dev

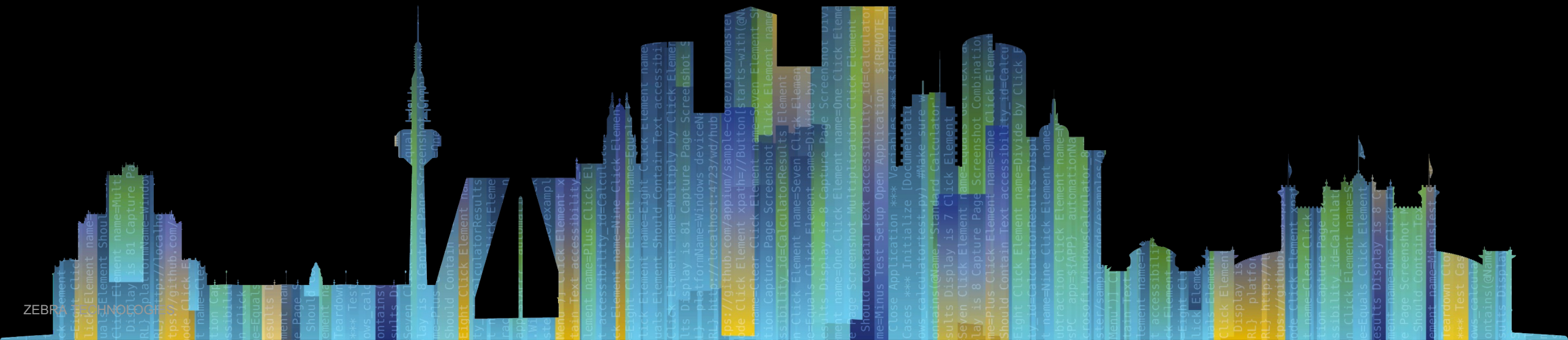
Not officially supported by Zebra!

  - Ionic integrated *Datawedge*, via Capacitor (https://ionic.io/blog/announcing-zebra-datawedge-integration)
  - Ionic-Cordova *DW* plugin still growing in popularity



Zebra DevCon 2023

Zebra Enterprise Browser

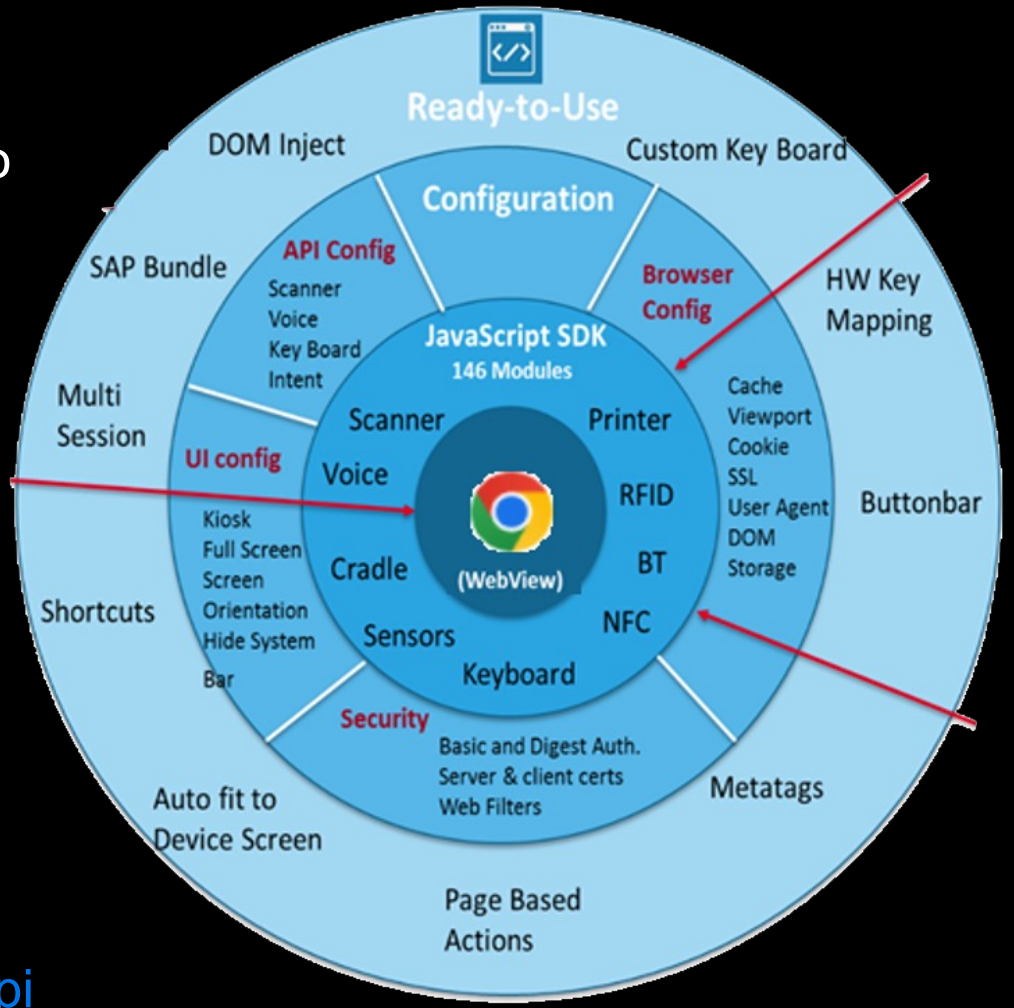


What is Enterprise Browser?

Introduction

- Enterprise Browser is a cross platform Industrial Browser which provides a rich JavaScript interface to Zebra value-adds such as
  - Android Intents
  - Barcode scanner
  - Printer
  - RFID
  - Keycapture
  - Custom keyboard etc
  - And camera, BT Scanners, Intent, Sensor etc.

- Exhaustive API set can be found below https://techdocs.zebra.com/enterprise-browser/3-4/api



Enterprise Browser SAP Bundle

EB provides ready to use configuration bundle for SAP ITS Mobile customers

- https://techdocs.zebra.com/enterprise-browser/3-4/guide/sapandroid/
- Customization Options
- UI auto customization via viewport
- SAP Button Height adjustment
- Ready to use function key layouts
- Transparent keyboards

- Button Keyboard show/hide mapped to H/W key
- Ready to use DataWedge scanning
- SAP Network errors handling
- H/W Keys remapped to quit, back, zoom-in, zoom-out of pages
- Orientation lock in Portrait/Landscape mode
- System Bar Hiding
- Webview configurations (Cookie,cache,DB,DomStorage)

Dom Injection

Dom Injection – Use Cases

Zebra DevCon 2023

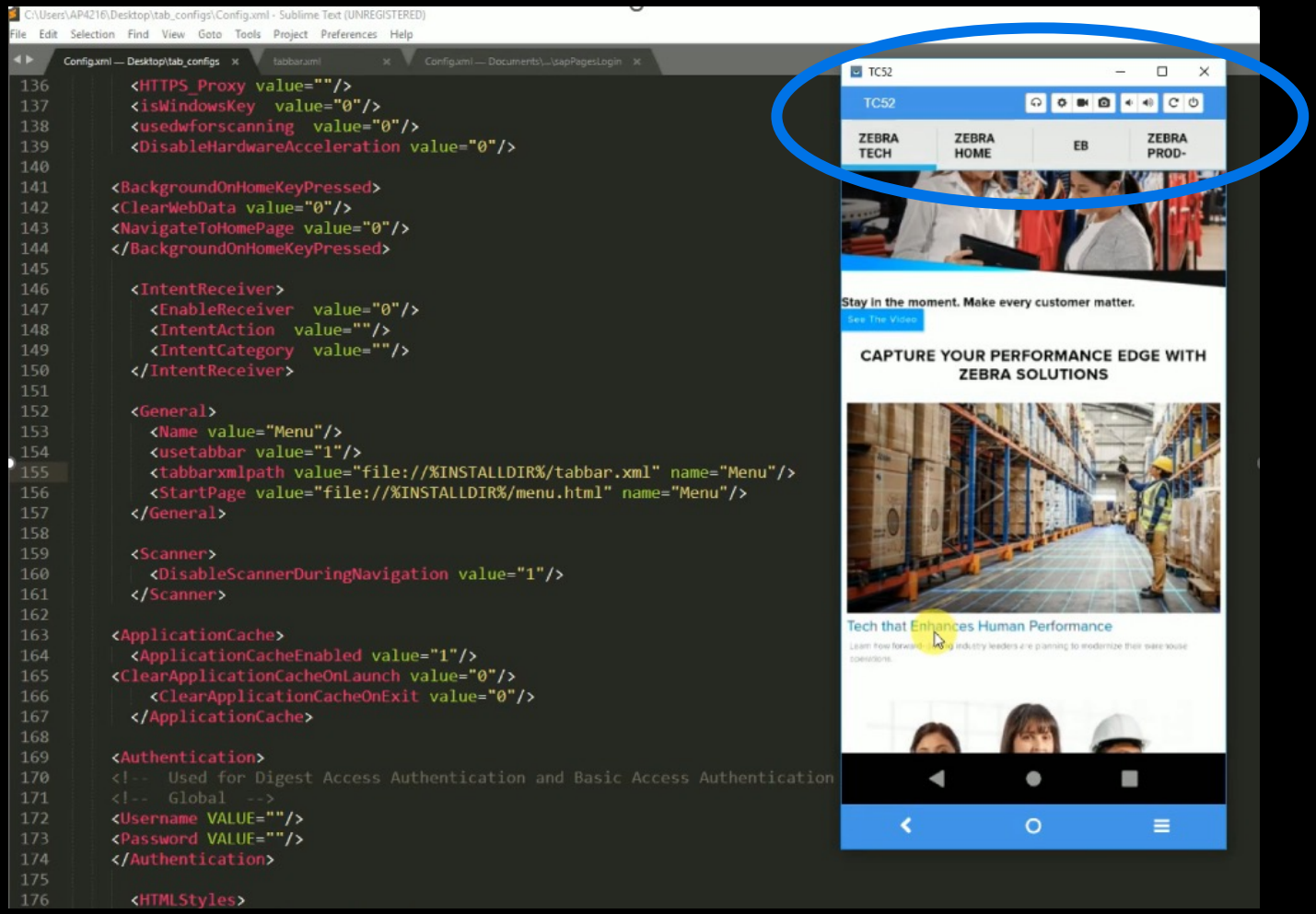
- I have a legacy web application running on desktop and I want to run them on Android devices and access device capabilities
- I am already running web application on the device. Now I want to add Enterprise and Zebra device capabilities
- I am running application on legacy WM/CE based devices. When I try to migrate to new Android devices, UI is not rendering well
- I have a mixed deployment of old and new devices. I only want to change the application behavior on new devices
- I do not want to change the server-side application logic which is very expensive to implement, validate, support and maintain
- I want to customize the application behavior specific to a (or a set of) device or a customer

The answer is Dom Injection….

Dom Injection provides a way for changing the runtime behavior of your application on the device without touching the server code on the existing backend

Multi-Session

Multi-Session – Tab approach

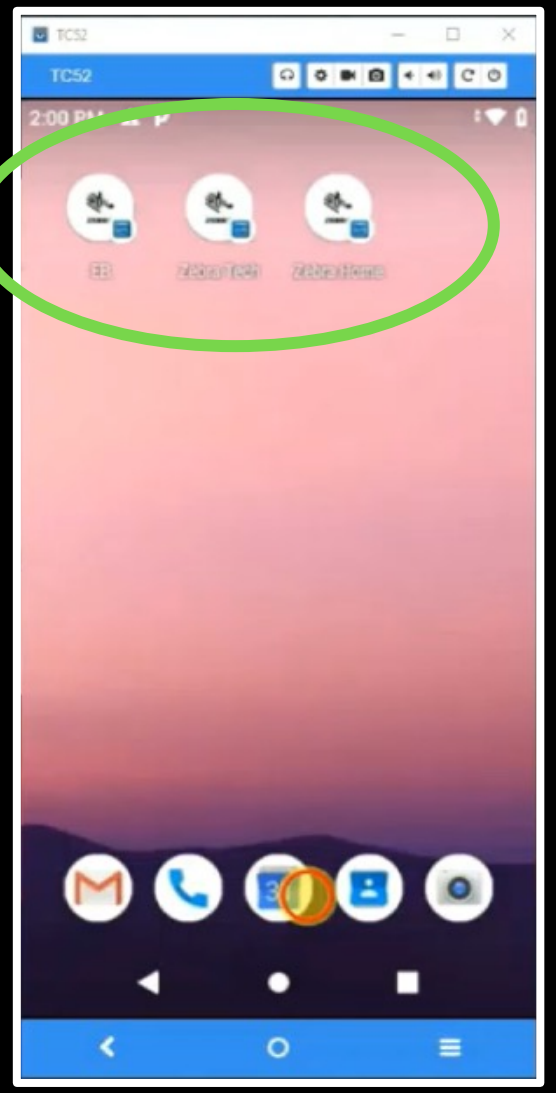


Multi-Session

Shortcut approach

- Separate shortcuts to launch web apps within Enterprise Browser, each maintaining own state
- Use the Shortcut Creator utility
- Define shortcuts within an XML file with separate configuration and icon

```
<Company>
  <Shortcut ID="0">
    <Name>Barcode</Name>
    <ConfigPath>/Barcode/Config.xml</ConfigPath>
    <IconPath>/Barcode/barcode_img.png</IconPath>
  </Shortcut>
  <Shortcut ID="1">
    <Name>Features</Name>
    <ConfigPath>/Features/Config.xml</ConfigPath>
    <IconPath>/Features/printer.jpg</IconPath>
  </Shortcut>
</Company>
```

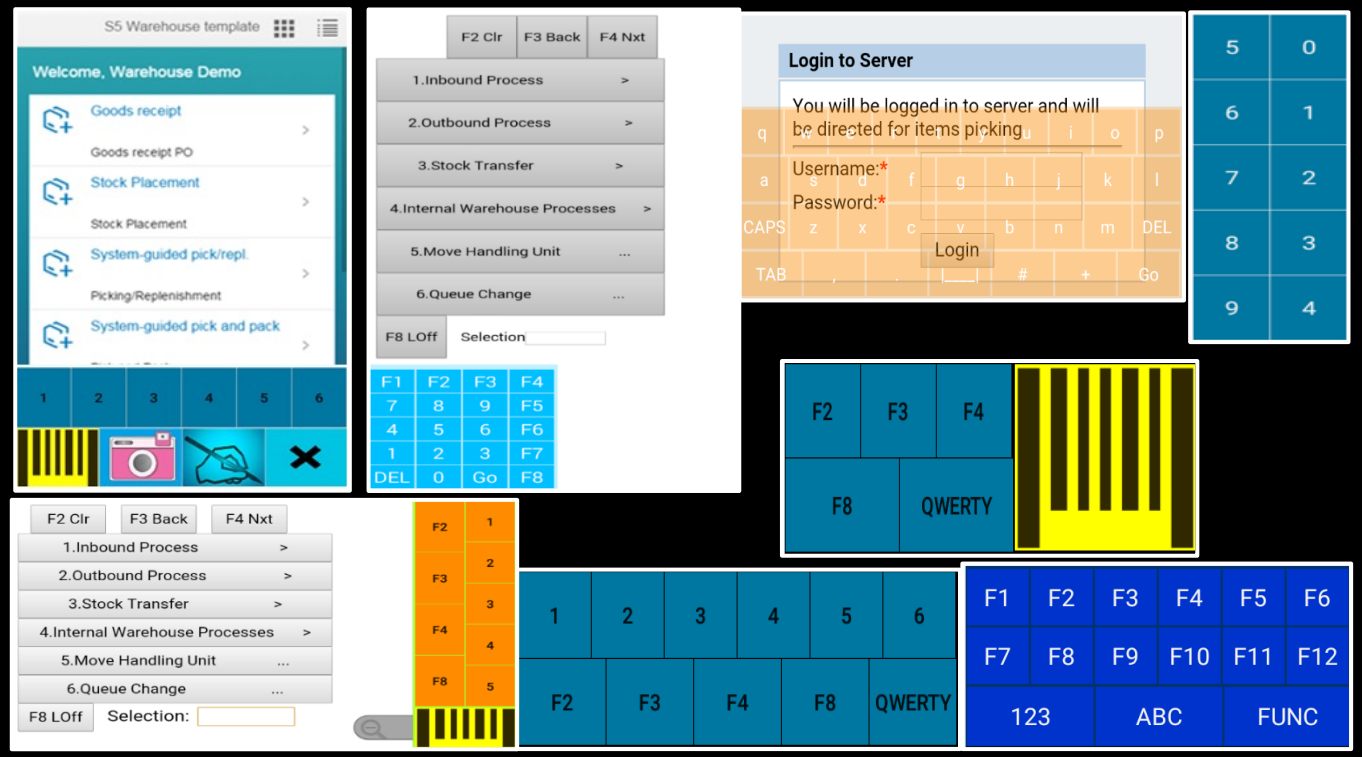


Enterprise Keyboard Designer (EKD)

Examples of Custom Key Layouts

Zebra DevCon 2023

- EKD is a Windows GUI tool that can be used to create customized key layouts
- Layouts created with EKD work on Zebra Android devices that use Zebra's Enterprise Keyboard (EKB) 3.2 (and later)
- Custom layouts can be displayed programmatically using Android intents or through DataWedge 7.4.44 (and later)
- EKD employs a drag-and-drop interface with control over fonts, images, key codes, layout transparency and many other layout properties.



Enterprise Browser

EB Internal Web Server

Zebra DevCon 2023

- You have the option to run a local, embedded web server on the device to service the application
- When multiple Webview applications are deployed, all can run from a single embedded server or use discrete servers, each running on a different port
- See also the WebFolder node to tell EB where the local application is hosted

Reference: https://techdocs.zebra.com/enterprise-browser/3-4/guide/configreference/#webserver

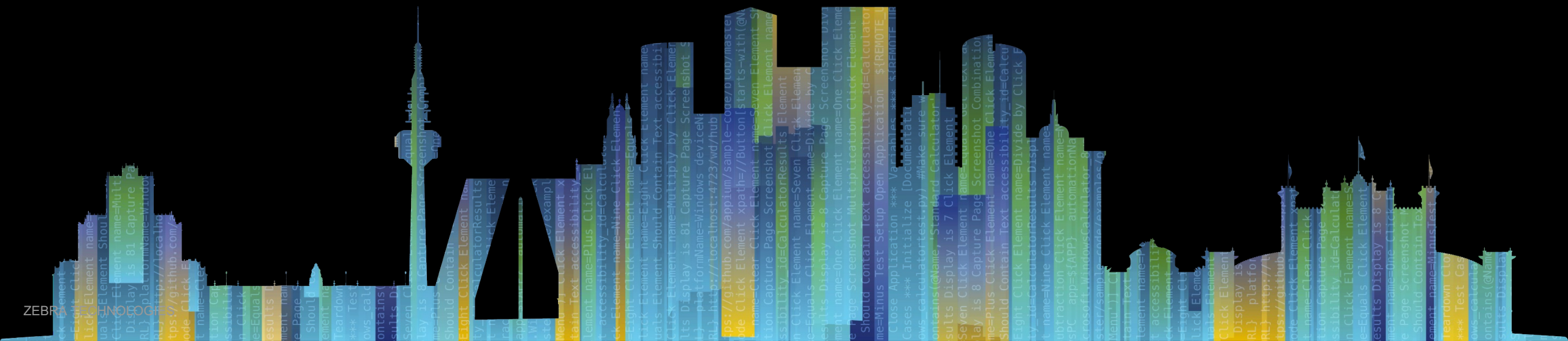
Zebra Enterprise Browser

Zebra DevCon 2023

	EB
Supported platforms	Zebra devices
Popularity	Highest among Zebra Web developers
Prominent Technology	Web plus Access to Zebra ***MX features*** and ***Android APIs***
Programming Language/ Skills	Web-related – HTML, JS
Pro's	Renders any WebApp and is capable of leveraging • **investments in JS and HTML** code as well as skill sets • specific features of Zebra devices and the Android OS
Con's / Drawbacks	Not available for platforms other than Zebra mobile computers (though applications can be designed to adapt to the underlying device)
Reference	https://techdocs.zebra.com/enterprise-browser/latest/guide/about/
License	Licensed
Integration to Zebra	Full

Zebra
DevCon 2023

Flutter



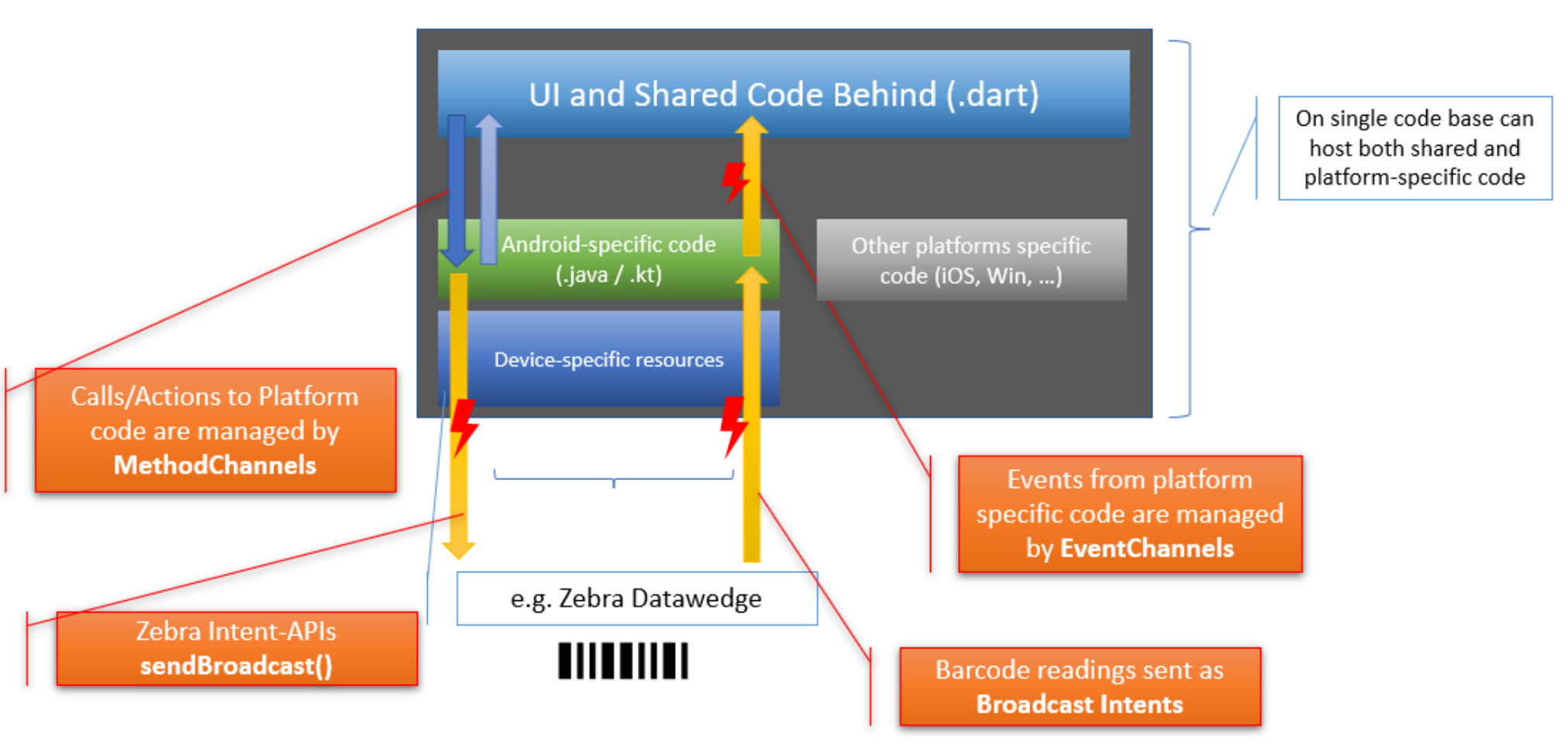
Flutter

Zebra DevCon 2023

	Flutter
Supported platforms	Android, iOS, Linux, macOS, Web, Windows
Popularity	Released in 2017, one of the most popular among Android developers, adoption doubling YoY.
Prominent Technology	UI Library Skia is platform agnostic, Powerful hot-reload capability
Programming Language	DART (C-style) object-oriented, strongly typed, client-optimized
Pros	Rich SDK, pre-built widgets, Open source, Efficient 2D rendering engine, Precise UI definition
Cons / Missing	Heavily using computing resources – large apps (starting from 38MB-release, 120MB-debug) Platform-specific plugins not coded in Dart; Weaker community support
IDE and License	Android Studio / Google grants you a non-transferable, non-exclusive, royalty-free limited license - https://docs.flutter.dev/tos
Plugin repo	https://pub.dev/
Integration to Zebra	See sample https://developer.zebra.com/blog/creating-flutter-applications-zebra-android-devices-datawedge using Platform Channels for platform's APIs in a non-Dart language Plus this deck's projects

Flutter

Native Flutter App operating Zebra Datawedge

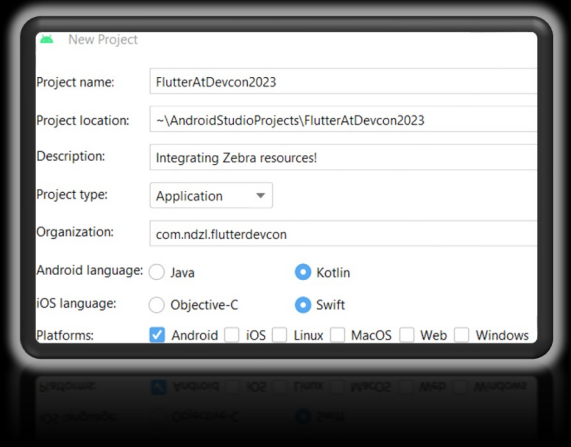


Flutter

Native app operating DW – Coding snapshots / Scanner gets triggered

Zebra DevCon 2023

Sample code



Shared Code

main.dart

```
89  static const platform = MethodChannel('com.ndzl.dw/ZebraDatawedgeMethodChannel');
90  String _methodChannelCallResult = '...';
91
92  Future<void> _softScanTriggerStart() async {
93    String callToPlatformMethodResult;
94    try {
95      final int result = await platform.invokeMethod('softScanTriggerStart');
```

Platform code (Android)

MainActivity.java

```
52      new MethodChannel(Objects.requireNonNull(getFlutterEngine()).getDartExecutor(),
53              name: "com.ndzl.dw/ZebraDatawedgeMethodChannel").setMethodCallHandler(
54
                { N.DZL }
55              new MethodCallHandler() {
                    { N.DZL }
56                  @Override
57                  public void onMethodCall(MethodCall call, Result result) {
58
59                      if (call.method.equals("softScanTriggerStart")) {
60                          dwStartScan();
```

Flutter

Native app operating DW – Pushing scanner data to the UI

Zebra DevCon 2023

```
{ N.DZL }
public class MainActivity extends FlutterActivity {

new EventChannel(Objects.requireNonNull(getFlutterEngine()).getDartExecutor(),
        name: "com.ndzl.dw/ZebraDatawedgeEventChannel").setStreamHandler(
        { N.DZL }
        new EventChannel.StreamHandler() {
            2 usages  { N.DZL }
            @Override
            public void onListen(Object arguments, EventSink events) {
                Log.i(TAG_LOG, msg: "EventChannel/onListen");
                dwEventChannelSink = events;

                //like onCreate for initialization, but with events available
                dwBroadcastReceiver = createDWBroadcastReceiver(events);
                IntentFilter filter = new IntentFilter();
                filter.addAction("com.ndzl.DW");
                filter.addCategory("android.intent.category.DEFAULT");
                registerReceiver(dwBroadcastReceiver, filter);
            }

    private BroadcastReceiver  createDWBroadcastReceiver(EventSink events) {
        Log.i(TAG_LOG,"createDWBroadcastReceiver");
        { N.DZL }
        return new BroadcastReceiver() {
            { N.DZL }
            @Override
            public void onReceive(Context context, Intent intent) {
                //
                String barcode_value = intent.getStringExtra("com.symbol.datawedge
                String barcode_type = intent.getStringExtra("com.symbol.datawedge.
                long epoch_scan_time =  intent.getLongExtra("com.symbol.datawedge.
                String readable_scan_time = (new Date(epoch_scan_time)).toGMTStrin
                Log.i(TAG_LOG,"onReceive DW="+ barcode_type+" "+barcode_value );
                if(events!=null)
                    events.success( "Scanned data: <"+barcode_value+">\nSymbology:
            }
        };
```

main.dart

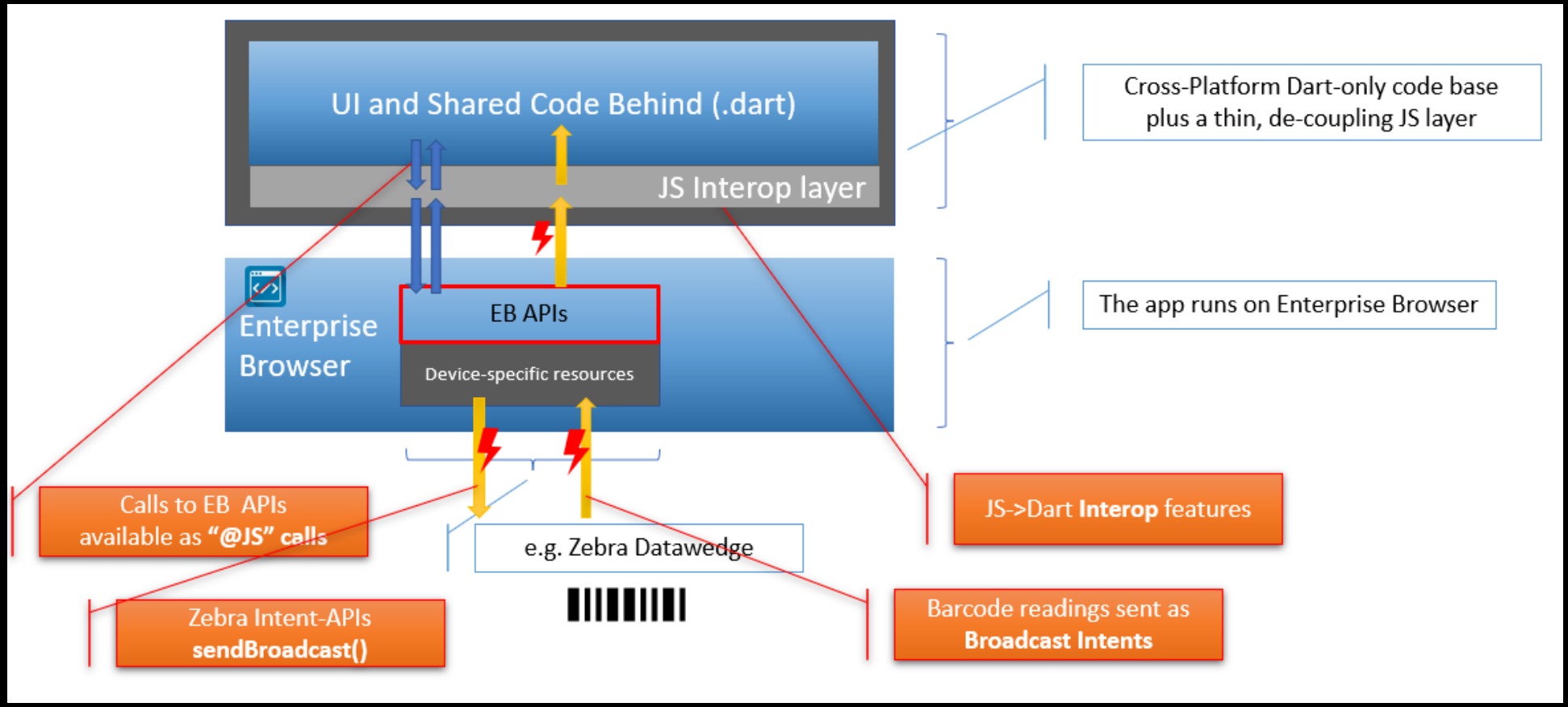
```
64  static const stream = EventChannel('com.ndzl.dw/ZebraDatawedgeEventChannel');
65
66  late StreamSubscription _streamSubscription;
67  String _latest_barcode = "N/A";
68
69  void _startListener() {
70    _streamSubscription = stream.receiveBroadcastStream().listen(_listenStream);
71  }
72
73  void _cancelListener() {
74    _streamSubscription.cancel();
75    setState(() {
76      _latest_barcode = "";
77    });
78  }
79
80  void _listenStream(value) {
81    debugPrint("Received From Native:  $value\n");
82    setState(() {
83      _latest_barcode = value;
84    });
```

Shared Code-Dart

Sample code

Flutter

Web app operating Datawedge via Enterprise Browser APIs



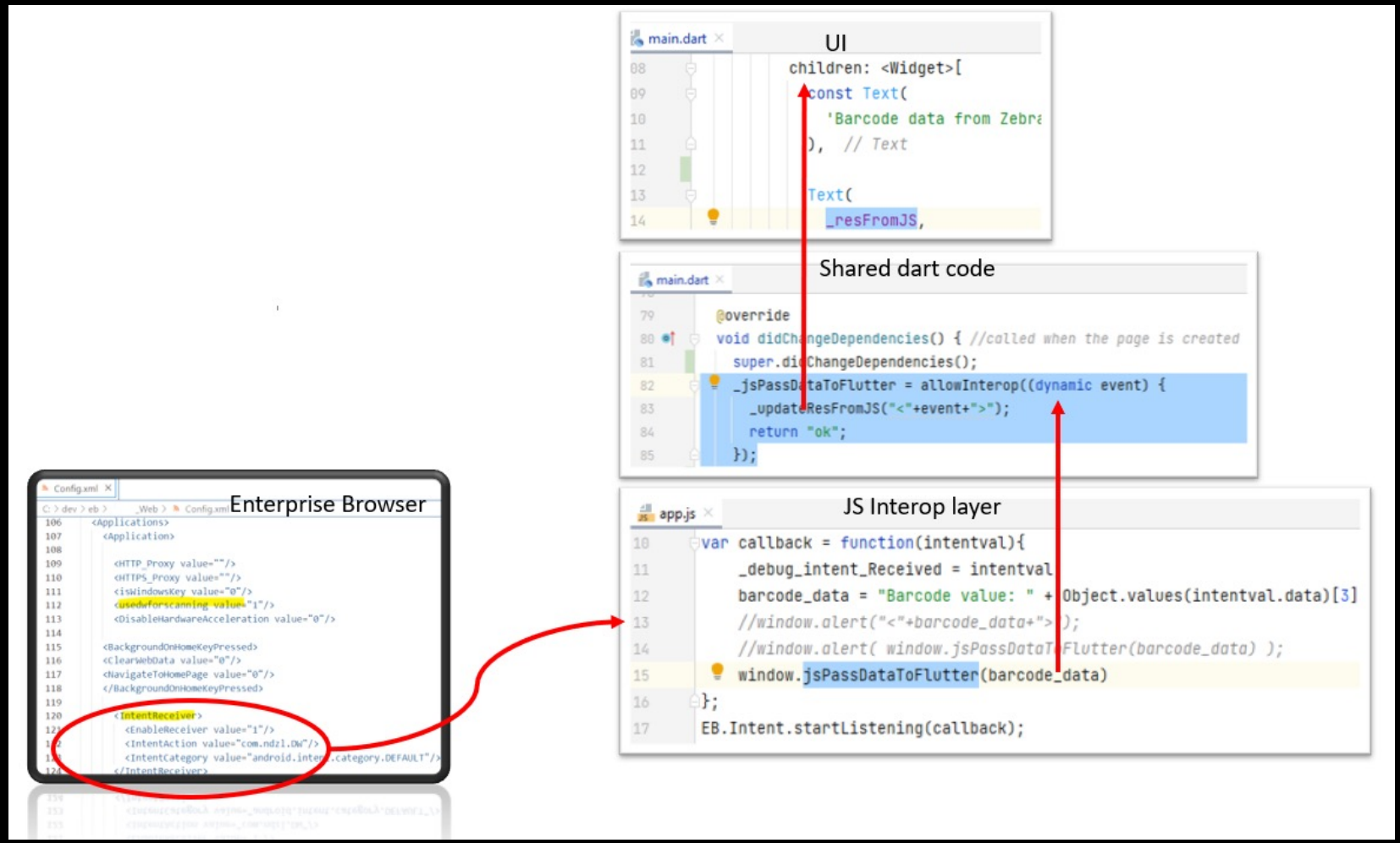
Flutter

Web app operating Datawedge via Enterprise Browser APIs

Zebra DevCon 2023

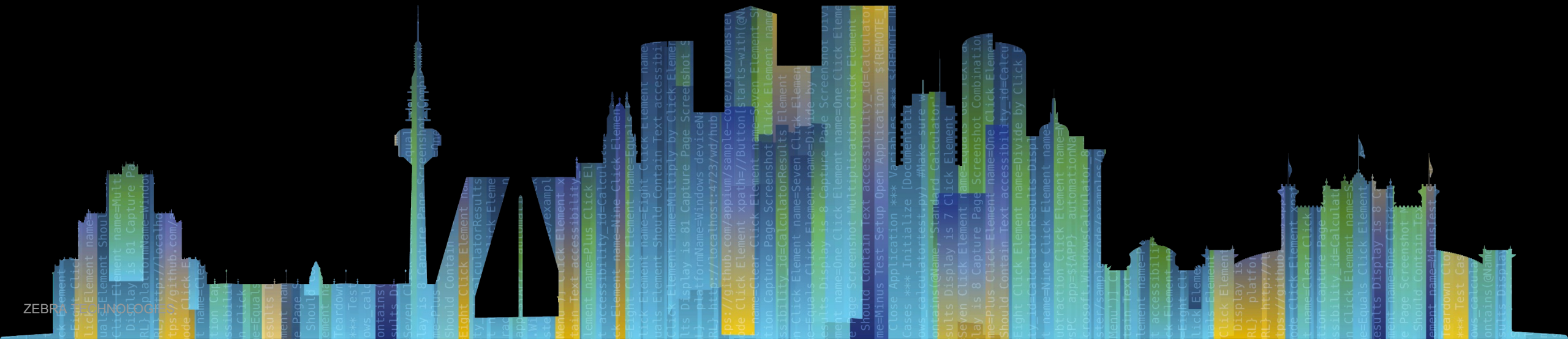
Coding snapshots

Sample code



Zebra
DevCon 2023

Ionic



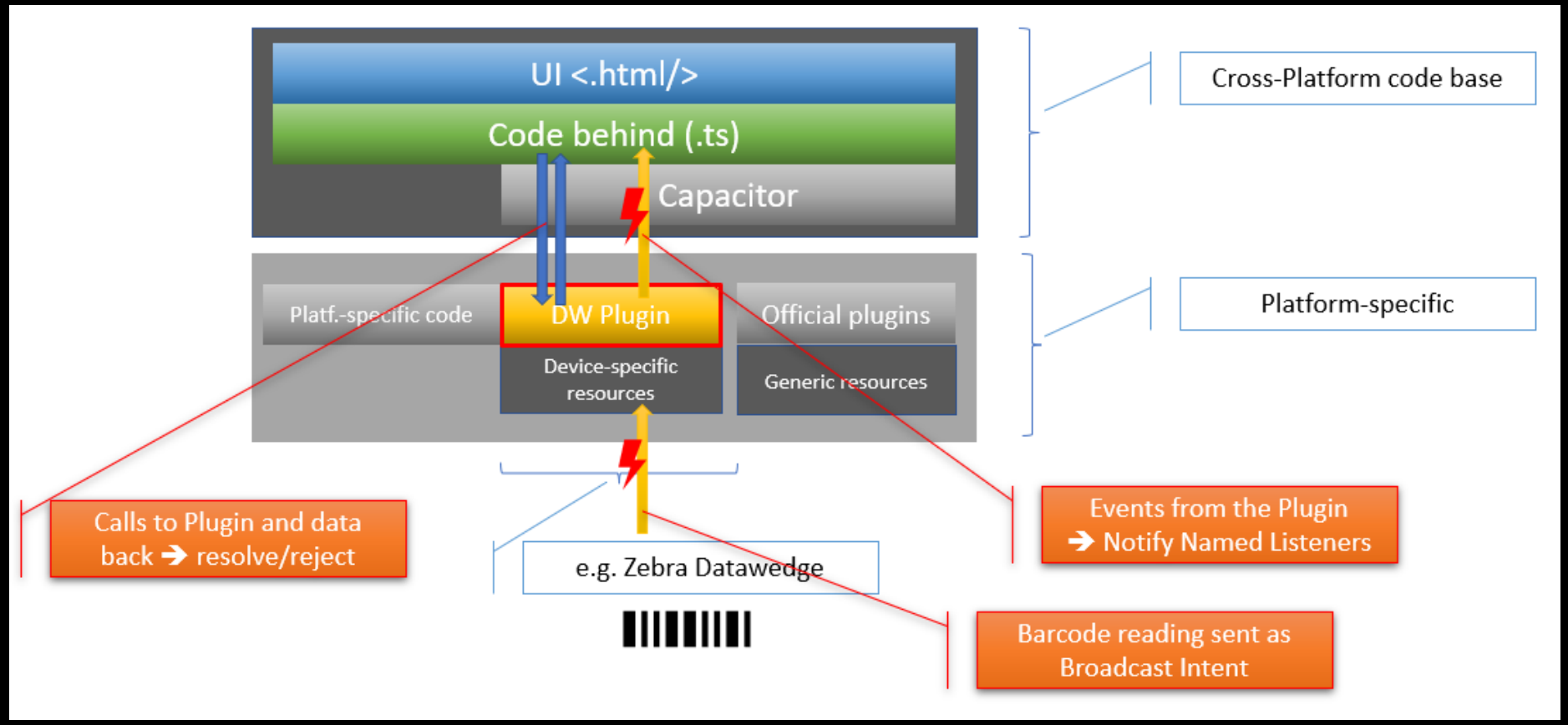
Ionic

Zebra DevCon 2023

	Ionic
Supported platforms	Ionic apps are truly cross-platform: able to run as an Android, iOS, Electron, and Progressive Web App (PWA), all from a single codebase
Popularity	Declining – 10y old, but with robust install base
Prominent Technology	Built-in support for JavaScript Frameworks (Vue, React, Angular), or use without any framework at all. Integrate Ionic with Capacitor to bring native capabilities to your app.
Programming Language /Skills	Web languages, HTML, CSS https://ionicframework.com/ https://ionic.io/resources/articles/capacitor-vs-cordova-modern-hybrid-app-development
Pros	Ionic apps run with a mixture of native code and web code, providing full access to native functionality if necessary
Cons / Drawbacks	Less performant because of Webview rendering, Rather complex for inexperienced developers. Community support is not as active as other platforms, though a thick knowledge base has been built over time.
Repos and License	npm – MIT
Integration to Zebra	Sample code https://github.com/ZebraDevs/DataWedge-Ionic-Demo https://ionic.io/blog/announcing-zebra-datawedge-integration

Ionic

Ionic-Angular Native App with Capacitor Plugin – Diagram

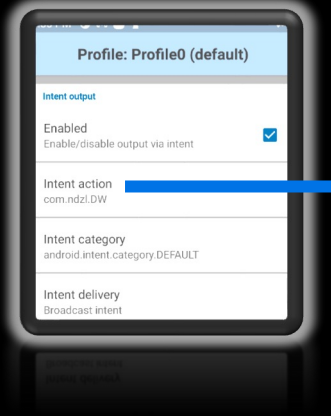


Ionic

Ionic-Angular Native App with Capacitor Plugin – Coding snapshots

Zebra DevCon 2023

Sample code




```
@CapacitorPlugin(name = "EchoPlugin")
public class EchoPluginPlugin extends Plugin {

protected void handleOnResume() {
    super.handleOnResume();
    registerIntentFilters();
}

1 usage
void registerIntentFilters()    {
    IntentFilter filter = new IntentFilter();
    filter.addAction("com.ndzl.DW");
    filter.addCategory("android.intent.category.DEFAULT");
    getContext().registerReceiver((context, intent) -> {
            if (intent.getAction().equals("com.ndzl.DW")) {

                /*ZEBRA*/
                String aimId = "N/A";
                String charset = "N/A";
                String codeId = intent.getStringExtra( name: "com.
                String data = intent.getStringExtra( name: "com.
                byte[] dataBytes = intent.getByteArrayExtra(
                //String dataBytesStr = bytesToHexString(data
                long epoch_timestamp = intent.getLongExtra(

                JSObject ret = new JSObject();
                ret.put("value", data);
                notifyListeners( eventName: "dwBarcodeReading", ret);

            }
```

```
export class Tab1Page {
  private _storage: Storage | null = null;
  constructor(private storage: Storage, private zone: NgZone) {
    this.initStorage();

    EchoPlugin.addListener('dwBarcodeReading', (info: any) => {
      console.log('tab1.page.ts intent '+info.value+' received from dwBarcodeReading'

      this.printDWreading(info.value);
    });

  }


async printDWreading(s: string){
  this.zone.run(() => {
    this.txtIn = s;
  });
}
```

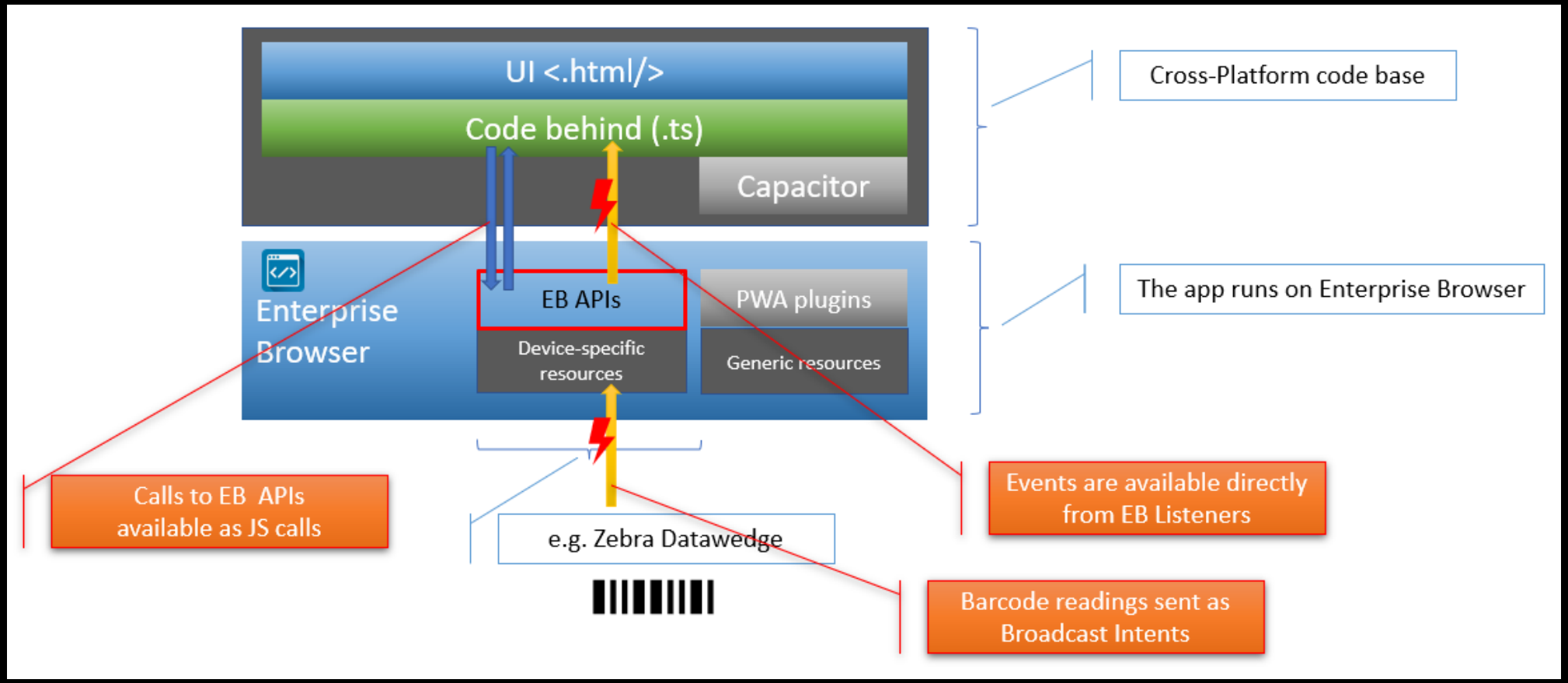
tab1.page.html M

src > app > tab1 > tab1.page.html > ion-content > ion-input

```
19      <ion-label>&lt;=Click it!</ion-label>
20      <br>
21      <ion-input label="Write something" [(ngModel)]="txtIn"  placeholder="Zebra"> </ion-input>
22
```

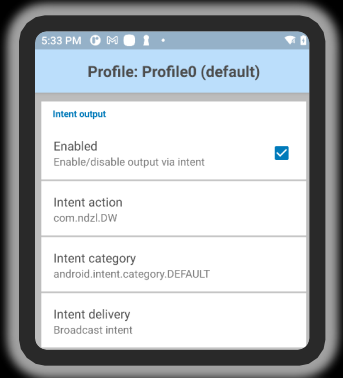
Ionic

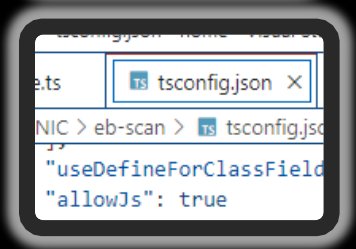
Ionic-Angular Web App operating DW via EB APIs – Diagram

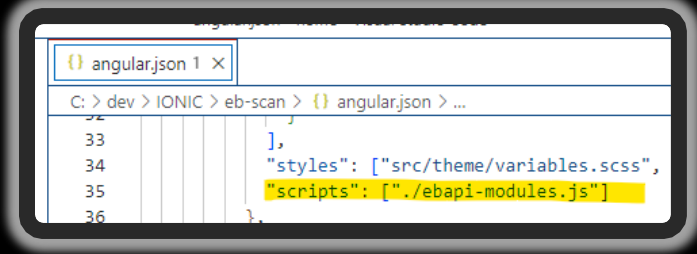


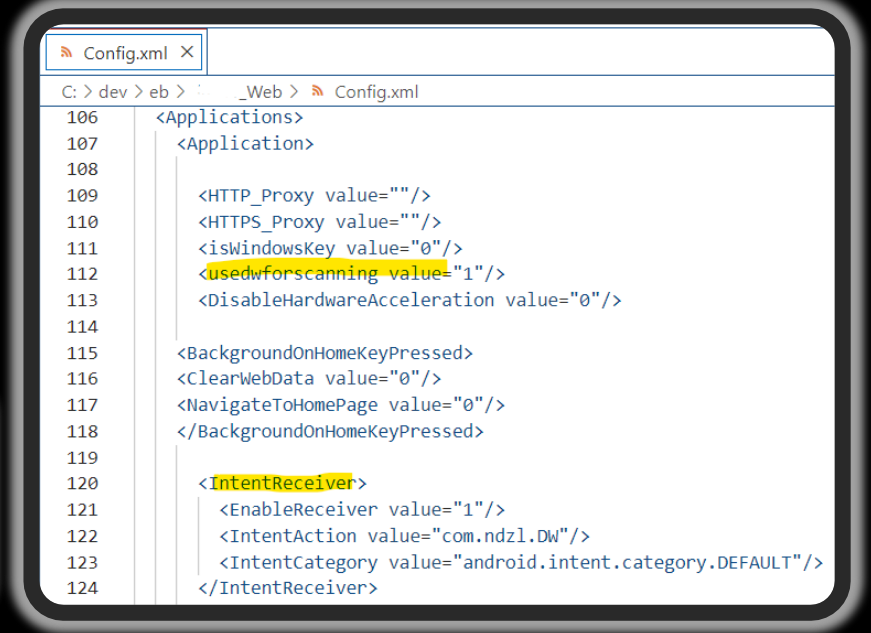
Ionic

Ionic-Angular Web App operating DW via EB APIs – Coding snapshotsz









Enterprise Browser configuration file

No need for a JS Interop. Layer like in Flutter

Typescript code

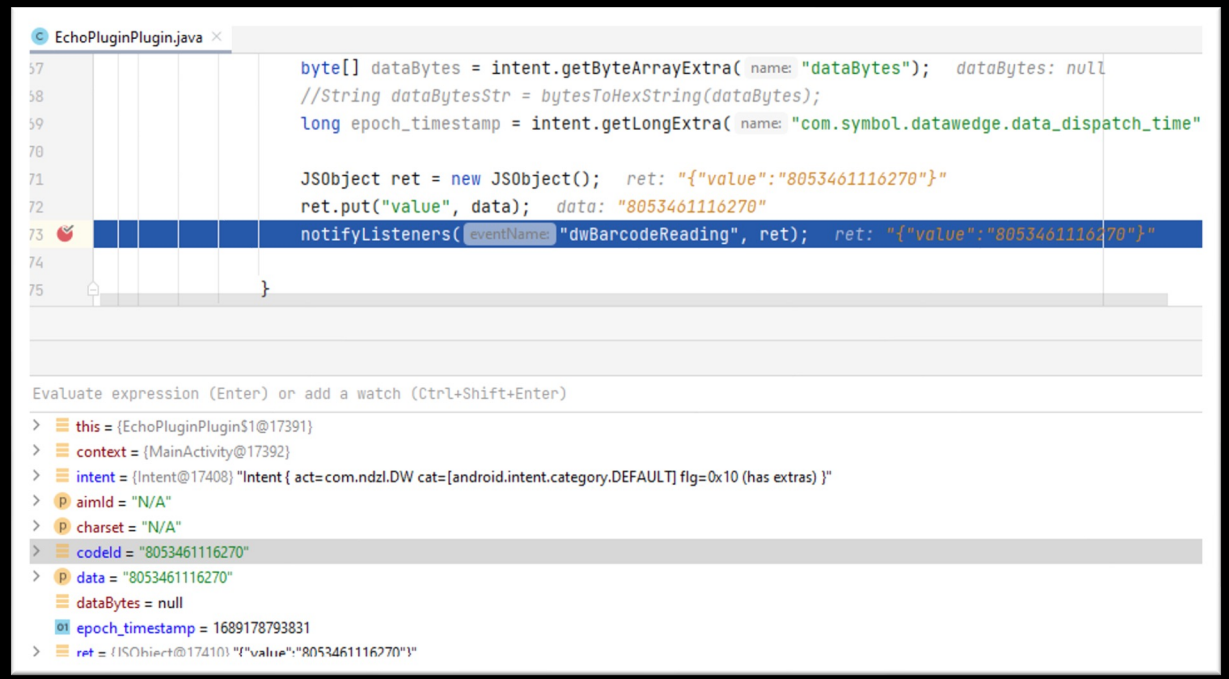
Sample code

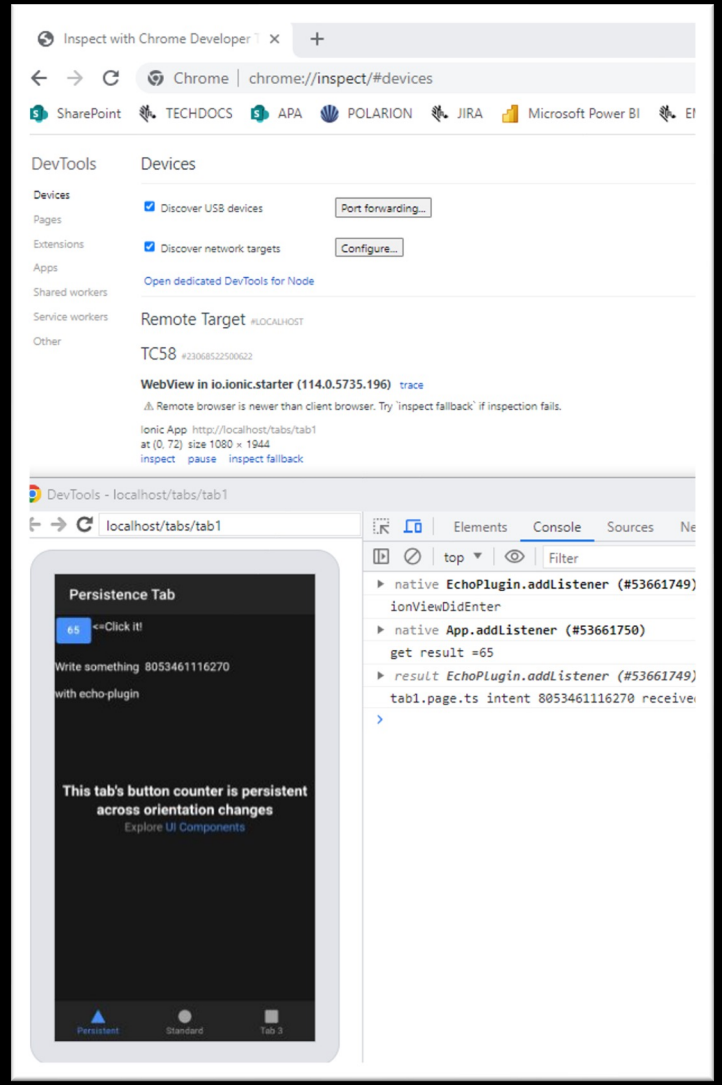
Ionic

Debugging

Zebra DevCon 2023

- Standard Android Studio Debugging features are available for native apps code-behind

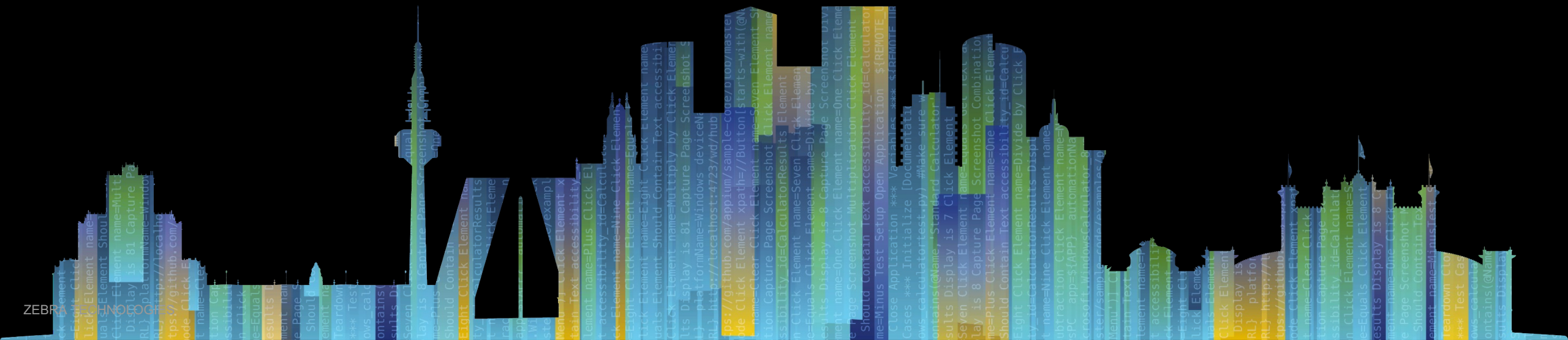




- UI Typescript code is rendered in a Webview even for native apps! So use chrome://inspect/#device

Zebra DevCon 2023

React Native



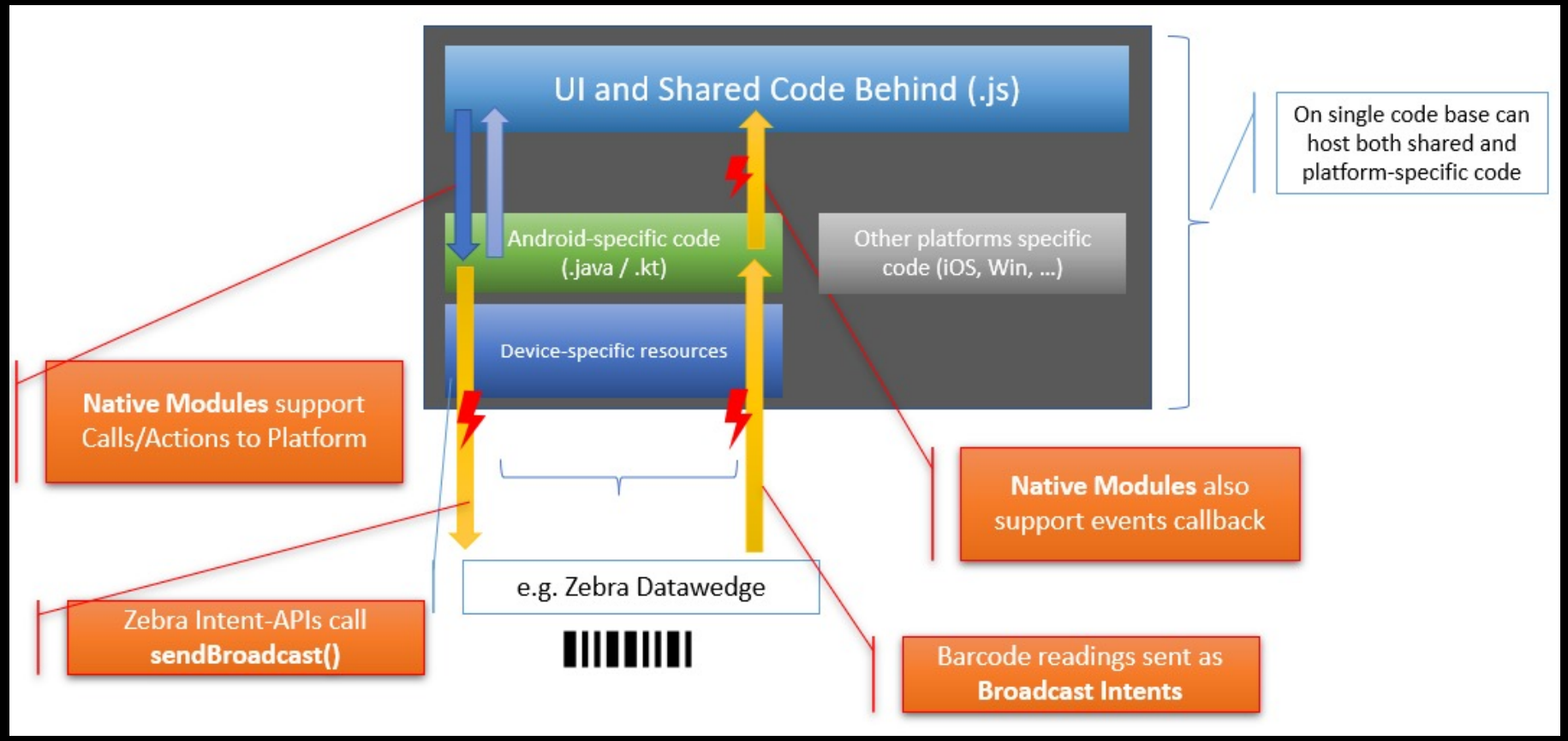
React Native

Zebra DevCon 2023

	React Native
Supported platforms	Android, Android TV, iOS, macOS, tvOS, Web, Windows and UWP + Oculus
Popularity	Head-to-head with Flutter – Slight decreasing but with vast install base
Prominent Technology	UI is built on native OS components, not emulated –App UI auto updates with OS
Programming Language	Javascript
Pros	Access to native functionalities, like camera and other features Use of platform-specific native code to optimize individual apps Hot reload; Facebook backed – reliable; Open Source
Cons / Missing	Not for complex UI animations or transitions Some mark RN as challenging for developers, steep learning curve, and difficult to debug; Some see RN as just built by Meta for Meta so you have to fit into their model
IDE and License	Free: Expo CLI or React Native CLI plus Android Studio or Xcode Paid: JetBrains WebStorm or Visual Studio
Integration to Zebra	An example is given here https://github.com/darryncampbell/DataWedgeReactNative and https://github.com/darryncampbell/DataWedge-Expo-Sample

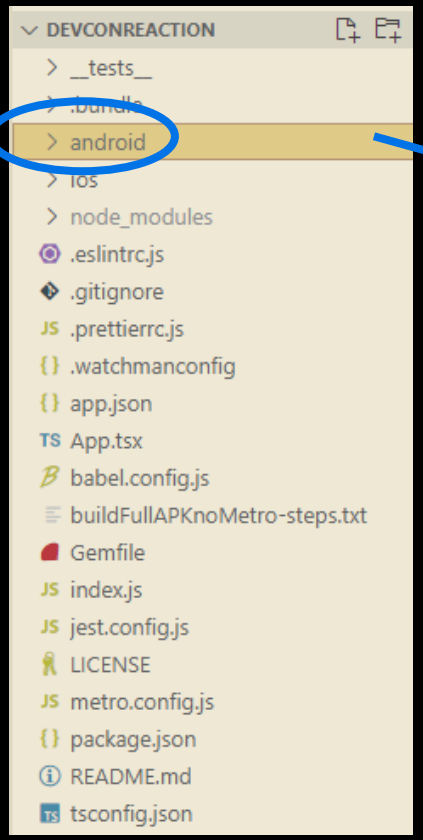
React Native

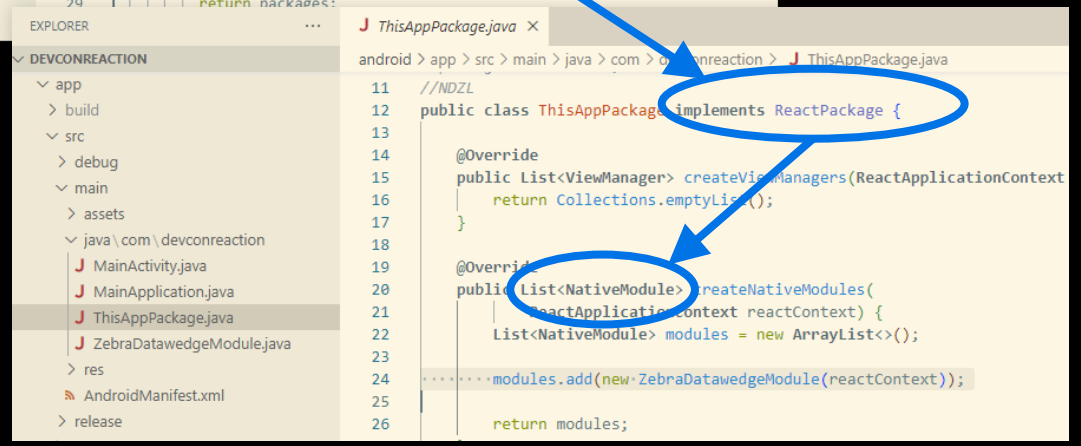
React Native App operating Zebra Datawedge – Diagram



React Native

React Native App operating DW – Coding Snapshots





Sample code

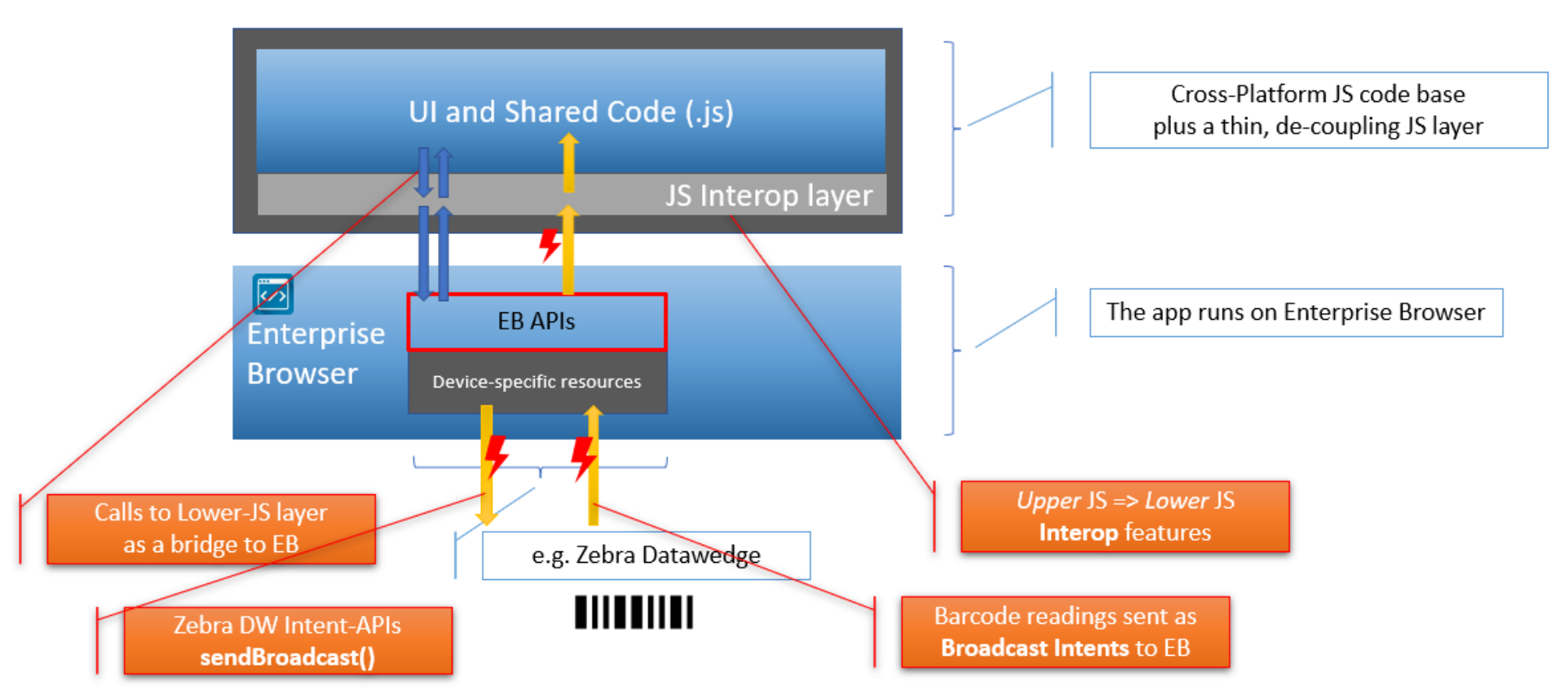
React Native

React Native App operating DW – Coding Snapshots

Zebra DevCon 2023

React Native

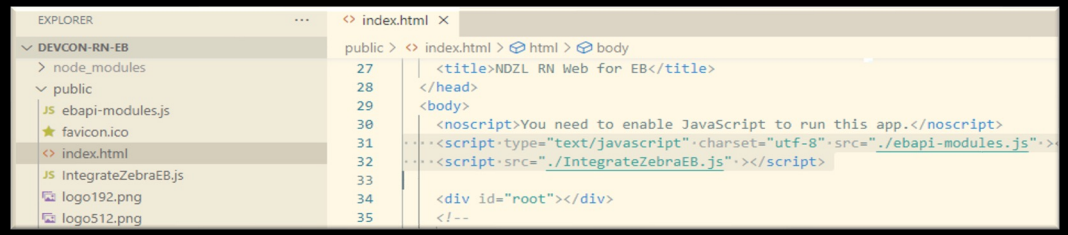
React Web App operating Zebra Datawedge via EB – Diagram



React Native

React Native Web App operating DW via EB – Coding Snapshots

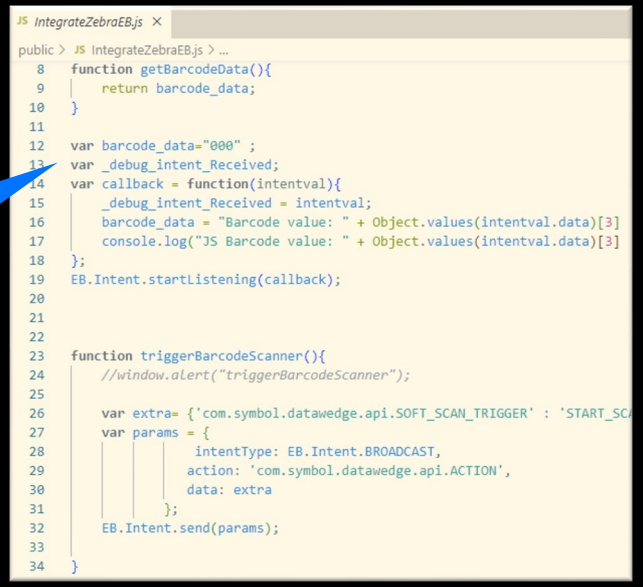
Zebra DevCon 2023



UI Container

Shared code

Interop Layer

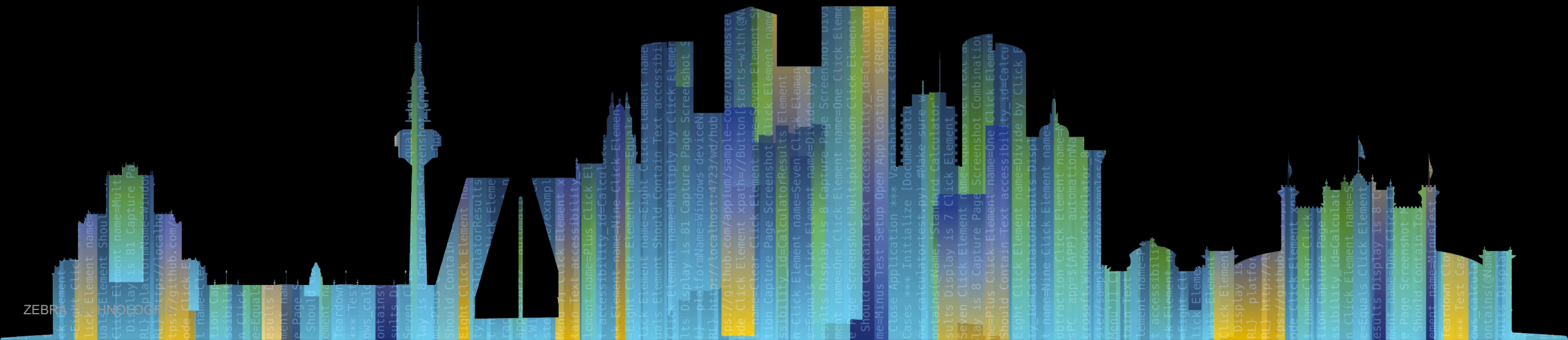


UI React Components

Sample code

Zebra
DevCon 2023

.NET MAUI



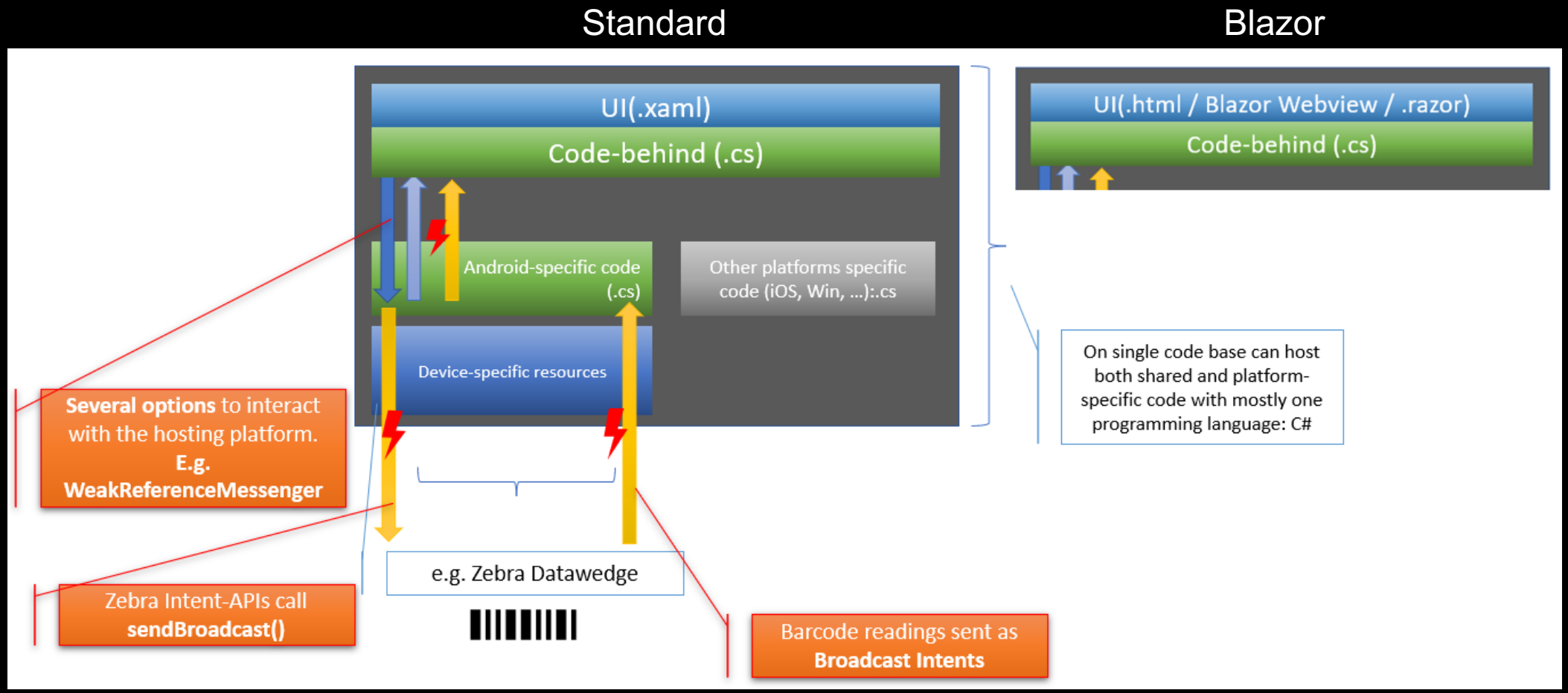
.NET MAUI (formerly Xamarin)

Zebra DevCon 2023

	MAUI
Supported platforms	iOS, Android, macOS, Windows
Popularity	
Prominent Technology	Microsoft .NET / Blazor
Programming Language /Skills	C# - Allows most of identical codebase across platforms
Pros	High native-like performance An extensive ecosystem with C#, .Net, and Microsoft Visual Studio Gives developers access to native APIs from Apple, Facebook, Google Platform-specific coding using C#
Cons / Missing	Updates often delayed before being reflected in the framework's tools Apps often larger than native apps Not suitable for complex Uis – poorer widgets library compared to Flutter
IDE and License	Visual Studio (Free / Paid)
Integration to Zebra	Officially supported. Sample code https://github.com/ZebraDevs/datawedge-MAUI-SampleApp and https://github.com/ZebraDevs/emdk-MAUI-Blazor-SampleApp

.NET MAUI

Standard and Blazor .NET Maui App operating Datawedge



.NET MAUI

Standard and Blazor .NET Maui App operating Datawedge - Coding

Zebra DevCon 2023

```
public class DWIntentReceiver : BroadcastReceiver
{
    0 references
    public override void OnReceive(Context context, Intent intent)
    {
        //System.Console.WriteLine("Here is DW on MAUI");
        if (intent.Extras != null)
        {
            string bc_type = intent.Extras.GetString("com.symbol.datawedge.label_type");
            string bc_data = intent.Extras.GetString("com.symbol.datawedge.data_string");

            WeakReferenceMessenger.Default.Send(bc_type + " "+ bc_data);
        }
```

Android-specific code

```
public class MainActivity : MauiAppCompatActivity
{
    0 references
    protected override void OnCreate(Bundle savedInstanceState) {
        base.OnCreate(savedInstanceState);
        RegisterReceivers();
```

UI Shared code-behind

```
public partial class MainPage : ContentPage
{
    int count = 0;

    0 references
    public MainPage()
    {
        InitializeComponent();

        WeakReferenceMessenger.Default.Register<string>(this, (r, m) =>
        {
            MainThread.BeginInvokeOnMainThread(() => { lbDisplayBarcodeData.Text += "\n"+m; });
        });
    }
```

```
<Label
    Text="Welcome to .NET Multi-platform App UI"
    SemanticProperties.HeadingLevel="Level2"
    SemanticProperties.Description="Welcome to dot net
    FontSize="16"
    HorizontalOptions="Center"
    x:Name="lbDisplayBarcodeData"
    />
```

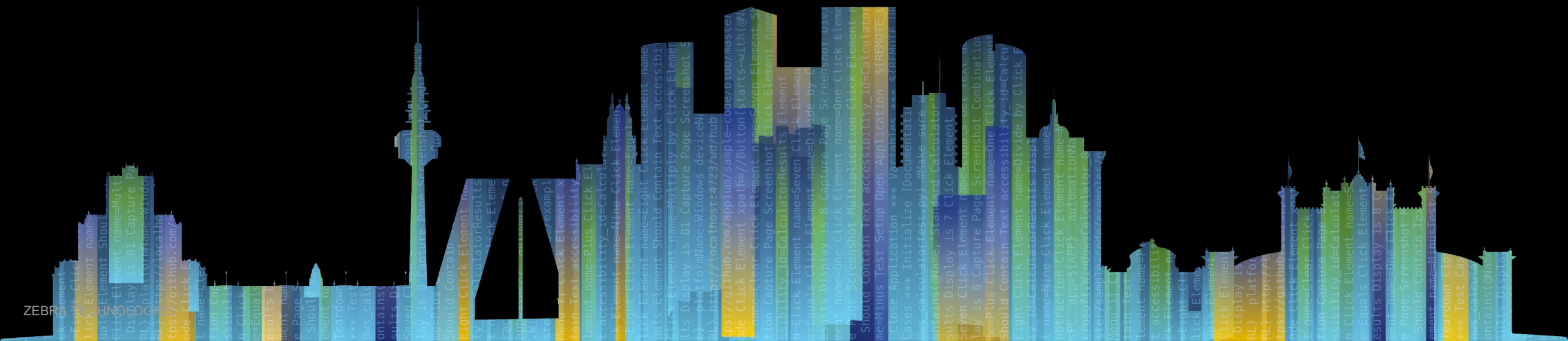
XAML UI

Sample code

DevTalk

Zebra DevCon 2023

Conclusions



Cross-platform development environments

Personal comparison of the Native solutions introduced earlier

Zebra DevCon 2023

	Flutter	Ionic-Angular-Cap	React Native	.NET MAUI
Nice feature(s) found	Flutter *doctor* Components licensing	Wide community KB	NativeModules allowed easy communic. between UI and Native layers	Super powerful inter-layers communicator
Not that nice	2 channels for upper/lower layers communication	-Capacitor is an additional project (plugin) -2 dev.env -2channel for communic.	assembleDebug/Release (full APK creation) not well doc. Default AS settings generating Metro-based APK	Just *native*, no web option
Experience in setting up the solution	Easy, well documented All done in Android Studio	Well documented, more complex than Flutter. Working template.	Fragmented, confusing: Expo vs RN CLI, npm vs yarn…	Well documented, Working template available
How many touch points to integrate DW	-2 channels setup, with 2 endpoints each -integrating EventSink in the Broadcast receiver -usual java DW code	-usual Java DW code -plus named listener notification added Easier than Flutter	-usual Java DW code -plus the NativeModules splicing	-usual C# DW code (copy/paste) -exchanging messages between platform and UI
Deliverable size (debug/rel)	starting from 38MB-release, 120MB-debug	6MB debug	55MB debug	9 MB debug
How did supporting AI performed (Copilot)	Good	Good	Good	Good

Cross-platform development environments

Comparing the Web solutions based on Enterprise Browser

Zebra DevCon 2023

	Flutter	Ionic-Angular	React Native for Web	~~.NET MAUI~~
Nice feature(s) found	Skinnier project	JS EB APIs directly integrated in Angular!	N/A	N/A
Not that nice	JS interop layer (app.js)	More complex to setup	JS interop layer (IntegrateZebraEB.js)	N/A as WebApp even though Razor using Webviews
Experience in setting up the solution and the JS integration	Events callback integration took longer	None	Couldn't push event data to UI without additional plugins	
How many touch points to integrate EB	-add JS interop layer -annotate @JS dart meth. -add standard EB Intent management for DW	Several: angular.json, tsconfig.json, plus the usual EB code to manage intents	-same Flutter complexity	
www folder size (debug/rel)	20MB	2MB	1MB	
How did supporting AI performed (Copilot)	Useless for @JS annotation	Coding was not an issue. No help in the configuration setup.	Didn't help to suggest a good inter-layer communic. solution	

Cross-platform development environments

The Developer Angle – Thoughts from personal experience and web searches

Zebra DevCon 2023

	Flutter	Ionic	React Native	.NET MAUI
Support				
Official doc quality	Comprehensive, up-to-date	Well regarded like Flutter's, more focused on the web technology	Has improved over time, but harder to cope with	Comprehensive but still under development for some topics
Community doc quality	Community is active in improving the overall documentation quality	Community contribution is valuable though less than Flutter's	RN heavily relies on Community contribution though for some complex topics lack in-depth explanations	Fewer community-generated resources compared to the other frameworks
Enterprise support / Contracts	**No services** commercialization or dedicated support	**Yes** - https://ionic.io/enterprise "Enterprise Support services, which provides access to our mobile experts, priority issue resolution, and a guaranteed response SLA."	**No service found** – Several bugs appear open and without SLA for resolution **No developer-focused business** like cloud service and frontend products found	**Yes** -Periodic updates/bug fixes versions are released by Microsoft -A Premier Support for Delevopers appears to be available

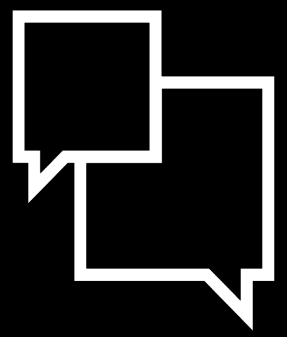
Cross-platform development environments

Links

Zebra DevCon 2023

Relevant sources used for comparison

- https://docs.flutter.dev/tos
- https://ionic.io/blog/who-is-going-to-support-your-next-mobile-app-project-hint-not-react-native-or-flutter
- https://reactnative.dev/community/support
- https://dotnet.microsoft.com/en-us/platform/support/policy/maui
  https://devblogs.microsoft.com/premier-developer/contact-us/



Questions

Zebra DevCon 2023

ZEBRA

Thank You