Zebra DevCon 2023

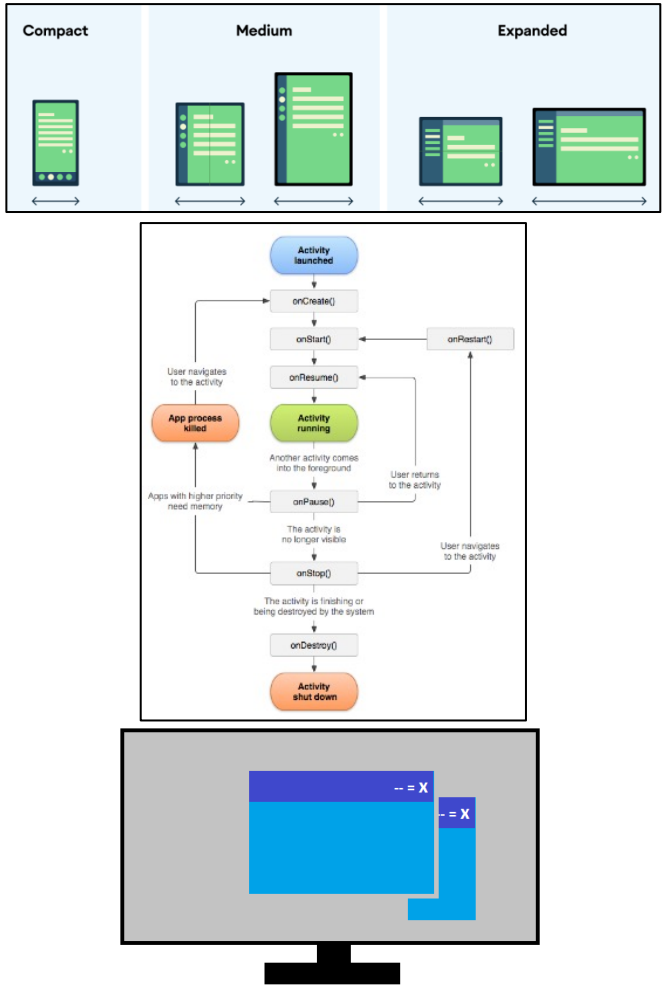
How your application can power a workstation or POS Station with Workstation Connect

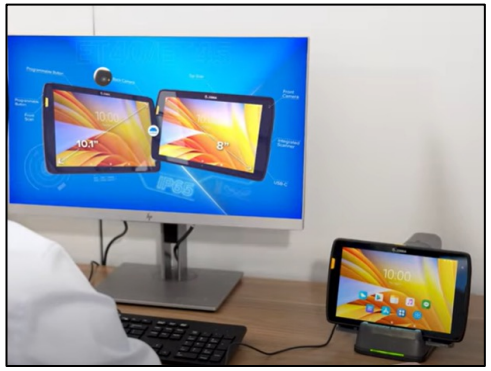
Nicola De Zolt L.
Software Engineer,
Zebra Technologies

Zebra Workstation Connect

Agenda / Teaser

Zebra DevCon 2023





Zebra DevCon 2023

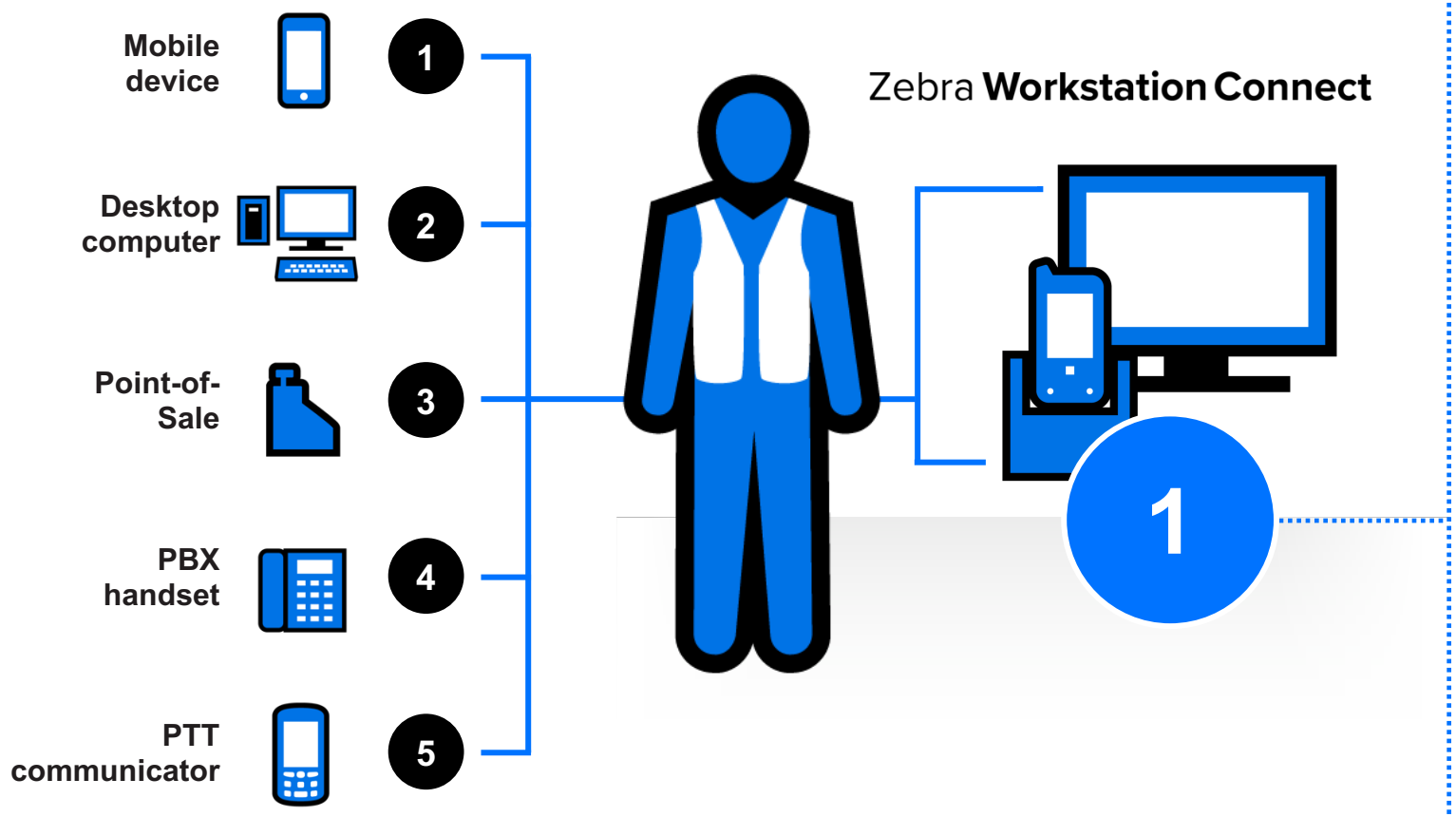
Why is Workstation Connect important for our Customers?

Use Case: Retail Associate

Zebra DevCon 2023

Store associates use FIVE different systems with different operating systems at different locations to get the job done…

…but ONE Zebra mobile device with Workstation Connect does it all.



Mobile device
Take inventory, restock shelves, putaway incoming shipments, price markdowns/markups, pick online orders, assist customers in aisle and more

Desktop computer
Assist customers with tasks best suited for a large display – assisted selling, product information and comparisons, complex in-store orders, employee training and more

Point-of-Sale
Process payments for purchases and returns

PBX handset
Call other store locations to check stock, call managers, call and receive calls from customers and more

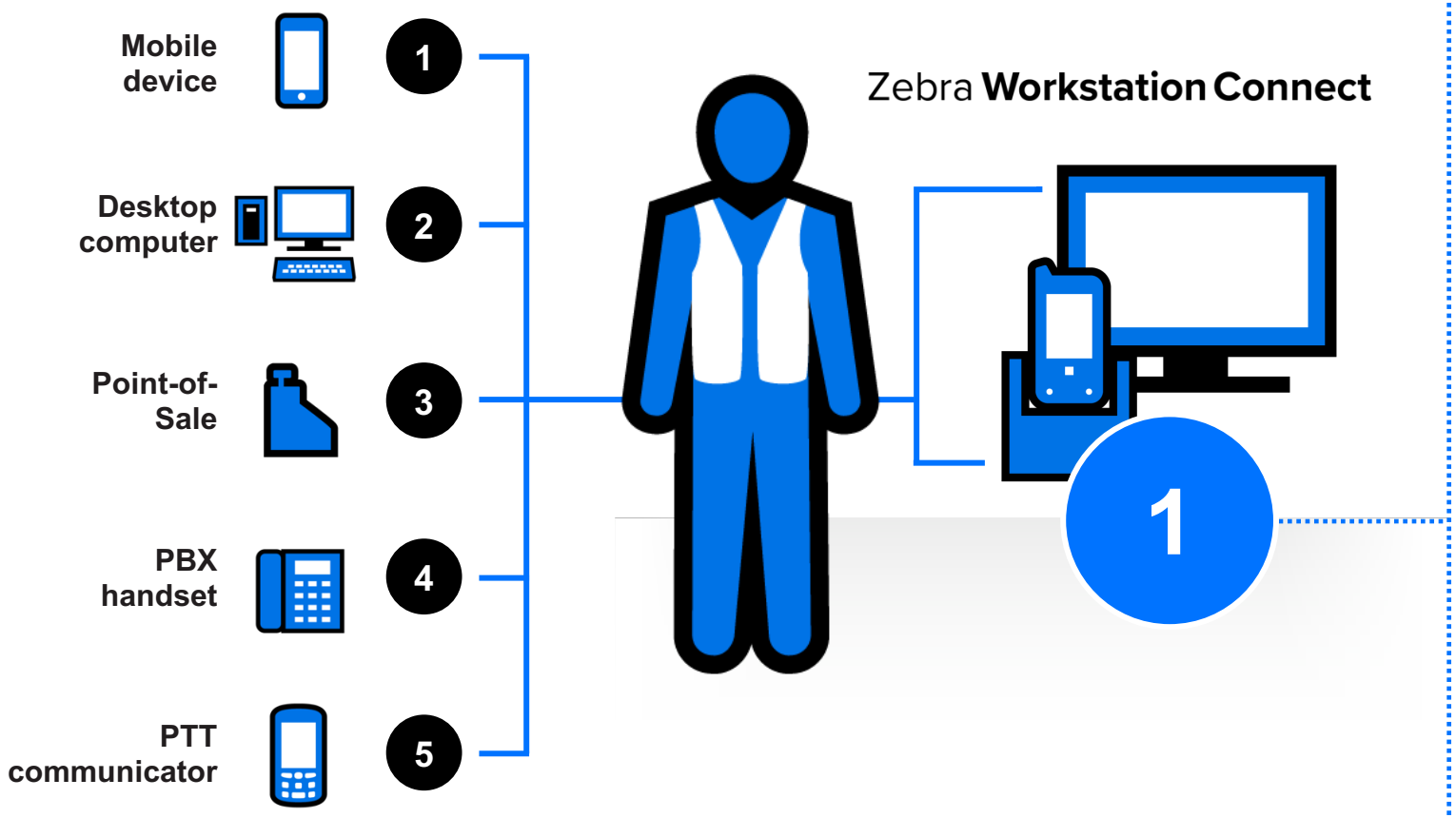
PTT communicator
Instant communications with co-workers and managers for tasks that require immediate attention

Use Case: Retail Manager

Zebra DevCon 2023

Store managers use FIVE different systems with different operating systems at different locations to get the job done…

…but ONE Zebra mobile device with Workstation Connect does it all.



Mobile device
Assist customers and associates on the sales floor, task management for associates

Desktop computer
Access store/corporate systems on a large display: manage inventory, place orders, schedule employees, manage sales promotions, adjust prices, perform sales analysis, create performance reports and more

Point-of-Sale
Assist with POS-related transaction and technical issues; perform overrides

PBX handset
Communications with other store locations/ corporate office

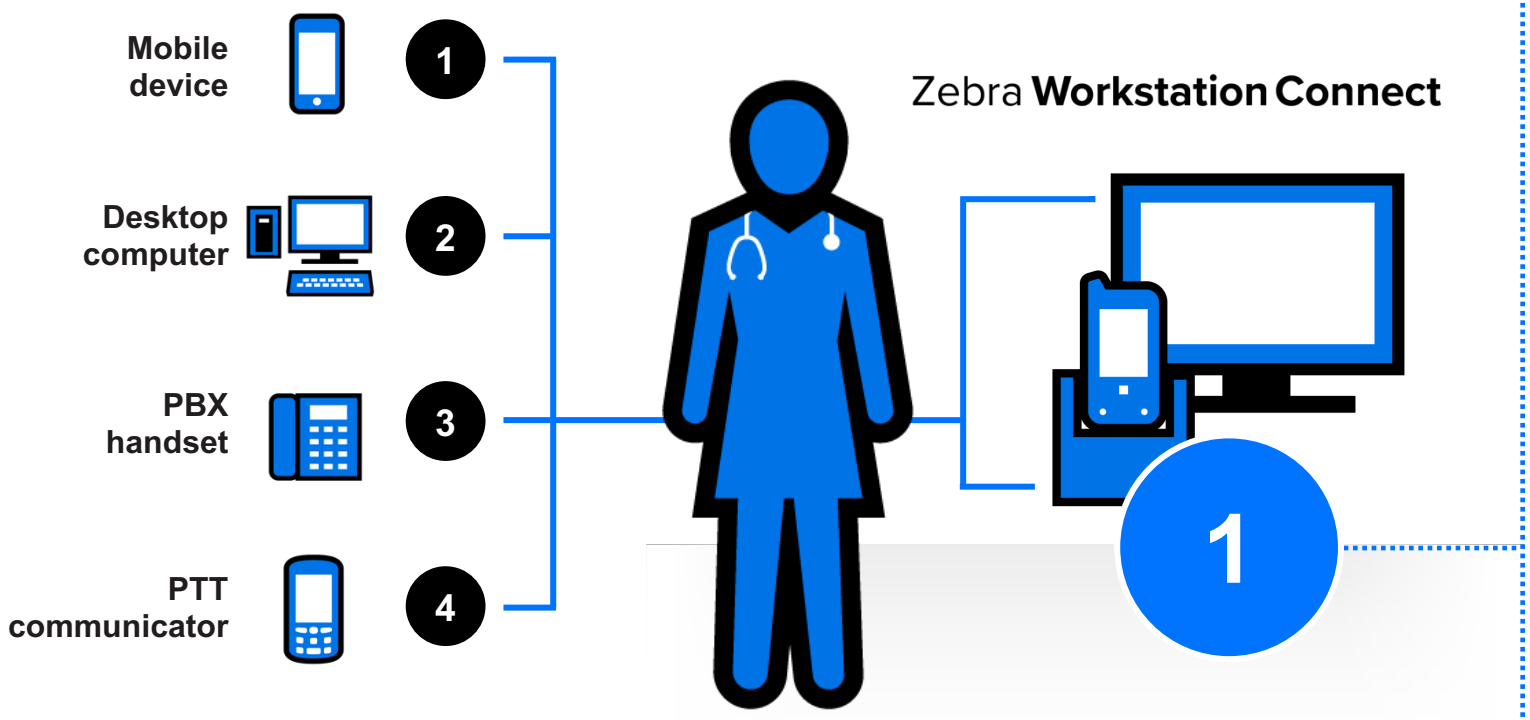
PTT communicator
Instant communications with associates; storewide or departmental staff communications

Use Case: Healthcare

Zebra DevCon 2023

Nurses use FOUR different systems with different operating systems at different locations to get the job done…

…but ONE Zebra mobile device with Workstation Connect does it all.



Mobile device
Perform activities at patient bedside — access EHR; scan wristbands for identity verification; scan specimen labels for tracking; capture images of wounds/incisions

Desktop computer
Access systems to complete data entry/ administrative tasks, update patient EHRs, complete admissions/discharge forms and more

PBX handset
Collaborate with physicians, surgeons, specialists, lab technicians, patient family members and more

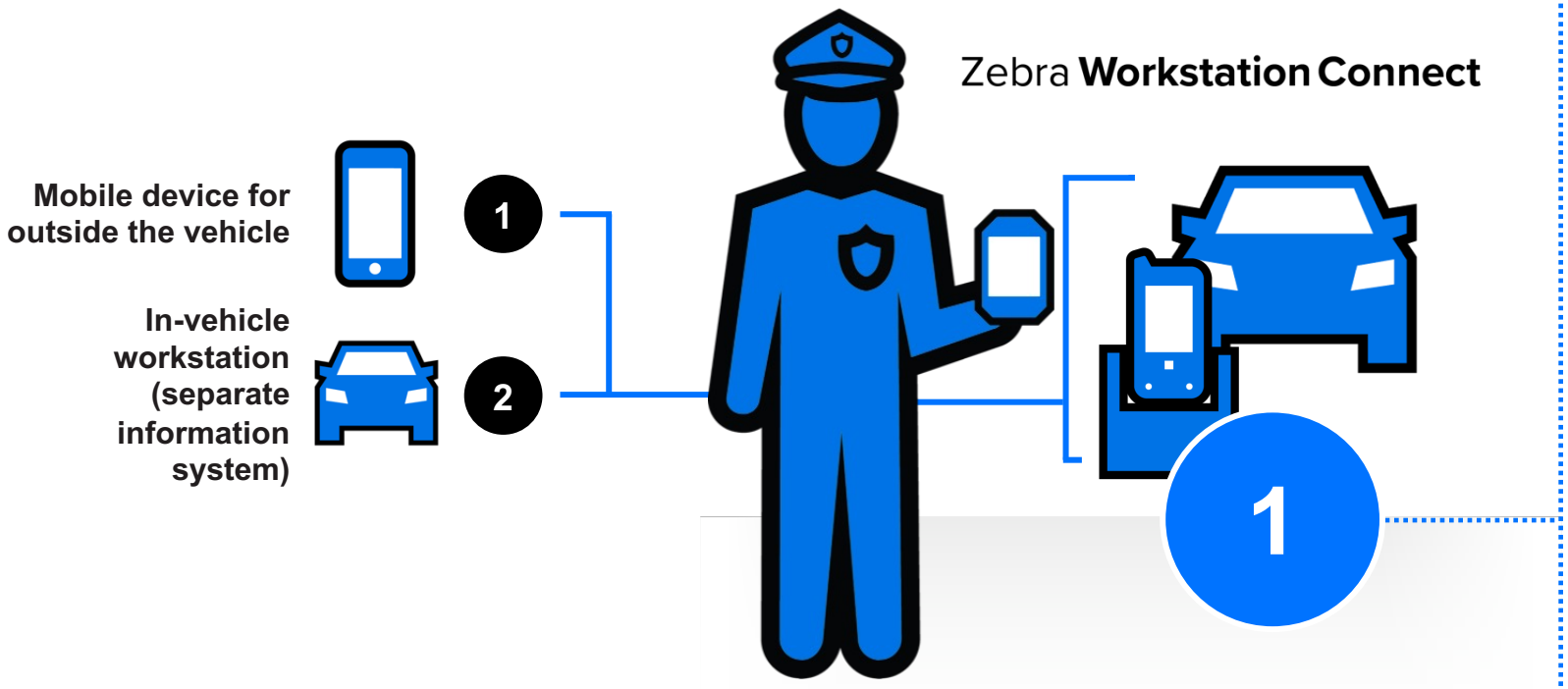
PTT communicator
Instantly reach other caregivers, security personnel, environmental services and more for time-sensitive issues

Use Case: Public Safety

Zebra DevCon 2023

Public safety officers need to access different information systems INSIDE and OUTSIDE the vehicle…

…but ONE Zebra mobile device with Workstation Connect does it all.



Mobile device (outside of the vehicle)
Scan driver's licenses; scan/capture images of vehicle registration; capture photos of license plates, vehicles and vehicle damage at accident sites

In-vehicle workstation
Complete online forms to issue tickets; document incidents; access databases such as the Department of Motor Vehicles (DMV); county/state/national criminal history; wants/warrants; probation records; call responses.

A single mobile device with a Workstation Connect in-vehicle cradle **condenses two systems in one**; Officers simply lift the mobile computer from the cradle to take the mobile-driven workstation out of the car, providing seamless access at all times

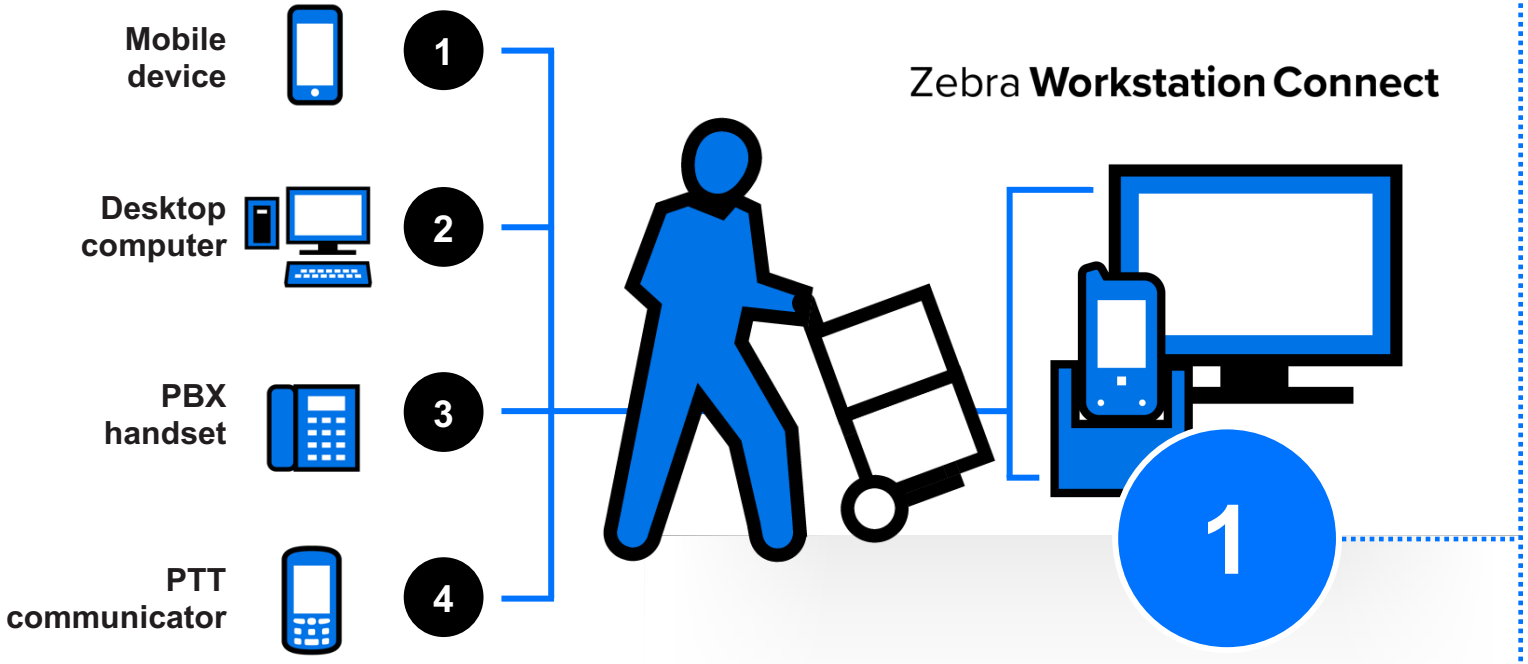
** In-vehicle compatible cradle for Workstation Connect coming in 2023 (3rd party validation)*

Use Case: Warehouse

Zebra DevCon 2023

Workers in warehouses use FOUR different systems with different operating systems at different locations to get the job done…

…but ONE Zebra mobile device with Workstation Connect does it all.



Mobile device
Accessing and completing work orders for put-away, picking, packing, shipping and receiving.

Desktop computer
Print paper barcode or RFID labels; print shipping labels and bills of lading

PBX handset
Warehouse floor workers collaborate with co-workers and managers; managers can communicate team members and other managers, plus corporate management

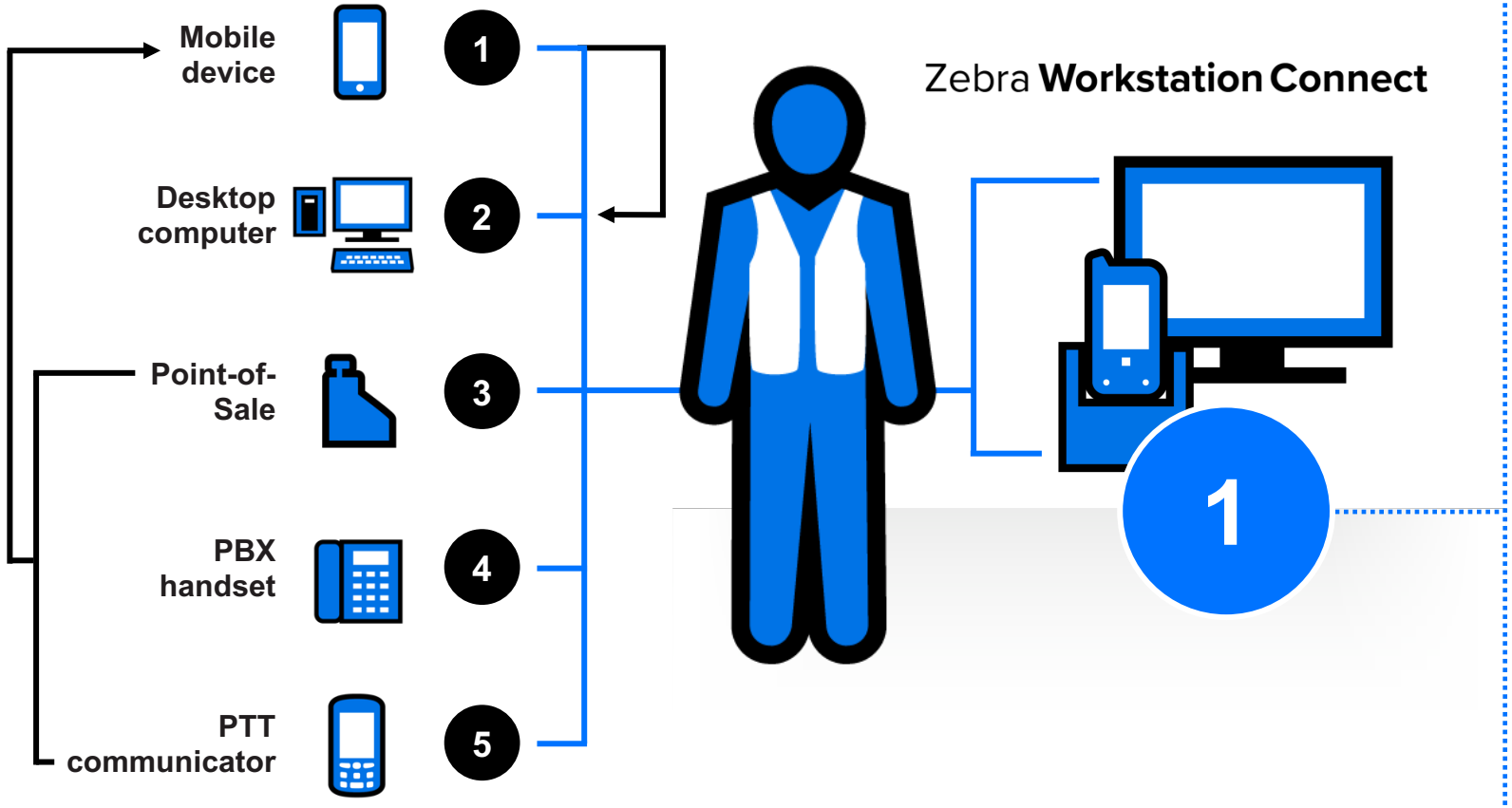
PTT communicator
Warehouse floor workers can instantly reach co-workers and managers for time-sensitive issues; managers can reach one or more specific team members or the entire team, instantly

Use Case: Hospitality

Zebra DevCon 2023

Associates in hotels use FIVE different systems with different operating systems at different locations to get the job done…

…but ONE Zebra mobile device with Workstation Connect does it all.



Mobile device
Assist guests on the spot throughout the hotel and hotel grounds

Desktop computer
Improve concierge services and the guest experience with a large screen to show guests area restaurants and other local activities that are available. Managers can better manage work schedules, email and voice communications.

Point-of-Sale
Ring up guest purchases anywhere in the hotel

PBX handset
Communications with co-workers, managers, guests and corporate headquarters

PTT communicator
Instant communications between co-workers and co-workers and their managers, departmental or all staff communications

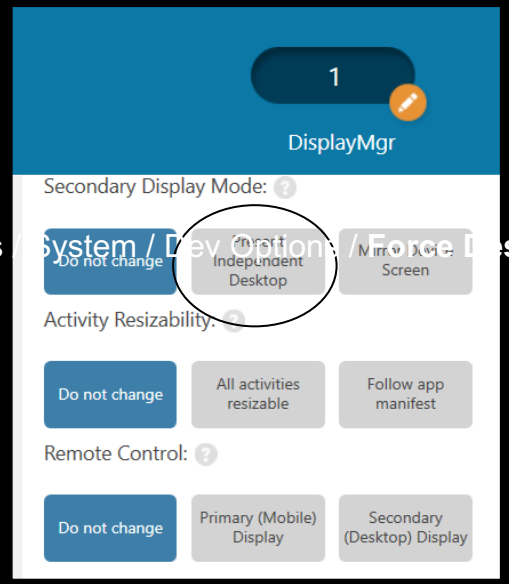
Zebra DevCon 2023

Large screen ready

Large screen ready

Enable the Multi-window mode

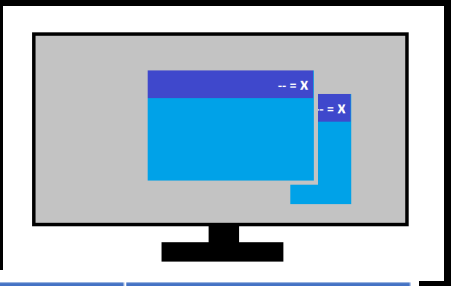
- Multi-window mode
  - Enable your app to run in multi-window mode alongside other apps either in free-form mode
    - Enable the Desktop Mode, either via Stagenow / Display Manager / **Secondary Display Mode** or in Android Settings / System / Dev Options / Force Desktop **Mode** (else it's in mirror mode)
    - Give end users a **full desktop** experience ➔ Show the Action bar
      e.g. set android:theme="@style/Theme.AppCompat"
      in the application or activity settings of the app's Manifest.xml
  - Use the WindowMetrics API to accurately determine the size of the app window
    - Current and maximum Window width/height
    - Is the window the Top activity
    - https://developer.android.com/reference/android/view/WindowManager
  - The Display Manager helps to deal with the multi-monitor scenarios
    https://developer.android.com/reference/android/hardware/display/DisplayManager



Large screen ready

Activity lifecycle in Multi-window mode

- Activity lifecycle in multi-window mode
  - Several new actions on a window trigger a state change



Move to Desktop	Maximize / Normal Size	Window is resized	Back from Minimized	Window is dragged	Window is closed (X)	Minimized
onPause onStop onDestroy onCreate onStart onResume	onPause onStop onDestroy onCreate onStart onResume	Only on significant changes onPause onStop onDestroy onCreate onStart onResume	onStart onResume	No state change	onPause onStop onDestroy	onPause onStop

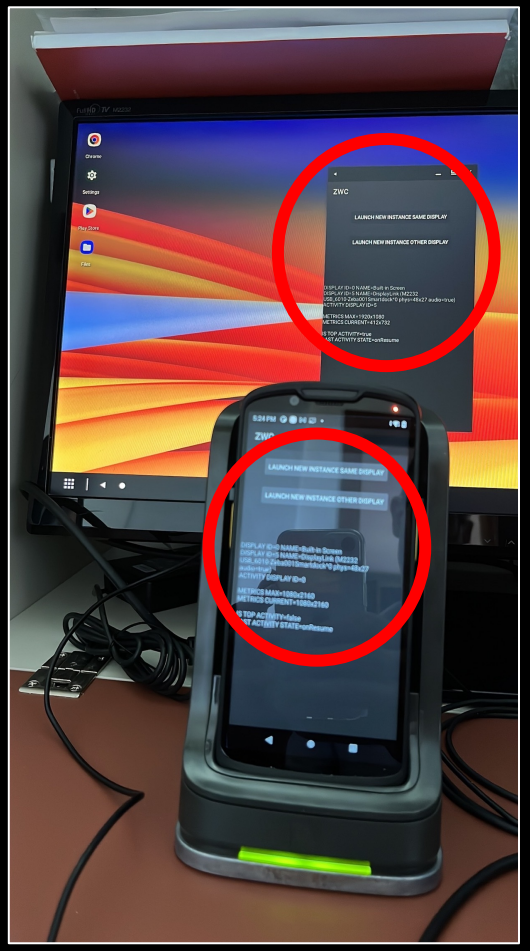
Full lifecycle

Partial lifecycle

Large screen ready

Activity lifecycle in Multi-window mode

- **Multi-instance:** the same activity can launch on the same or different monitor several times.
  Start the activity with Flags FLAG_ACTIVITY_LAUNCH_ADJACENT and FLAG_ACTIVITY_NEW_TASK set and check Manifest's android:launchMode
- **Multi-resume:** since API 29+ (Android 10 and above) all activities remain RESUMED when the device is in multi-window mode – before API 29, only the topmost activity in the RESUMED state. All other visible activities are STARTED but are not RESUMED.
- **Application vs. Activity context:**
  - Application's building blocks: Activities, Services, Broadcast receivers, Content providers
  - Activity context is UI-based and adjusted for the display area where it appears
- **Datawedge:** use DW Intent APIs to SOFT_SCAN_TRIGGER the scanner – the hardware button is not supported for windows on the 2nd screen



Large screen ready

Saving UI States

Zebra DevCon 2023

- Your app should retain and restore state during configuration changes and resumes ongoing processes such as media playback, filled-in forms”

User Expectation	When actions occur
• A consistent user experience through configuration changes and activity recreation. • States to be preserved	• Rotating the device • Entering multi-window mode • Resizing the application while in multi-window mode or a free-form window • Folding a foldable device with multiple displays • Changing the system theme, such as dark mode versus light mode • Changing the font size • Changing the system or app language • Connecting or disconnecting a hardware keyboard • Connecting or disconnecting a dock

If restricted the App enters *compatibility mode*

Large screen ready

Saving UI States

Zebra DevCon 2023

- Your app should retain and restore state during configuration changes and resumes ongoing processes such as media playback, filled-in forms
- **Bridging the gap between user expectations and system behavior**
  - Reference https://developer.android.com/topic/libraries/architecture/saving-states
  - Saved instance states facilities offered by Android depending on the UI technology used:
    - Jetpack Compose: rememberSaveable.
    - Views: onSaveInstanceState() API.
    - ViewModels: SavedStateHandle.
- Examples of persisted UI states in the next slides, for different platforms >>

Large screen ready

Saving UI States – Native Android Views

Zebra DevCon 2023

- The default implementation takes care of most of the UI per-instance state for you by calling View.onSaveInstanceState() on each view in the hierarchy that has an id
- And by saving the id of the currently focused view (all of which is restored by the default implementation of onRestoreInstanceState(Bundle)).
- If you override this method to save additional information not captured by each individual view, you will likely want to call through to the default implementation
- Reference
  https://developer.android.com/reference/android/app/Activity#onSaveInstanceState(android.os.Bundle)

Large screen ready

Saving UI States – .NET / MAUI

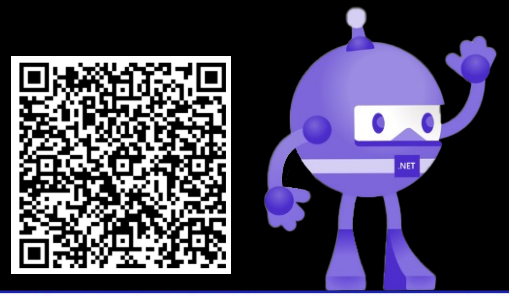
Zebra DevCon 2023

- Same as for native Android if using savedInstanceState bundle
- Reference https://learn.microsoft.com/en-us/xamarin/android/app-fundamentals/activity-lifecycle/saving-state

- Adding customized key/value pairs as an additional example
- Reference project datawedge-MAUI-SampleApp

```
protected override void OnCreate(Bundle savedInstanceState) {
    base.OnCreate(savedInstanceState);
    RegisterReceivers();

    //showing saved states
    try
    {
        String savedDatetime = savedInstanceState.GetString("time");
        if (savedDatetime is not null)
            WeakReferenceMessenger.Default.Send("Saved DateTime=" + savedDatetime);
    }catch(Exception ex) {
        WeakReferenceMessenger.Default.Send("No previously saved instance available");
    }
}
```



```
protected override void OnSaveInstanceState(Bundle outState)
{
    String currentDatetime = DateTime.Now.ToString();
    outState.PutString("time", currentDatetime);
    base.OnSaveInstanceState(outState);
}
```

Large screen ready

Saving UI States – Chrome / Enterprise Browser / JS

Zebra DevCon 2023

- **Chrome** – v114
  - Keep in mind what actions cause a page to reload

Move to Desktop/ Device	Maximize / Normal Size	Window is resized	Back from Minimized	Window dragged	Tab Navigation	Minimized	Pull to refresh	Manual Reload, Link Navigation
//	//	//	//	//	//	//	Reloads if not disabled	Reloads

  - Use JS localStorage in conjunction with window.onload and window.onbeforeunload to save/retrieve UI State data
  - Also works with local html files served to Chrome via content provider
  - Reference code: https://ndzl.github.io/zwc.html
- **Chrome:** To suppress the annoying Pull-to-refresh feature, causing unwanted page reloading (behavior not related to Workstation Connect)
  - No chrome://flags found for that
  - Need adding overscroll-behavior: none; in each HTML page
- Reference https://developer.chrome.com/blog/overscroll-behavior/

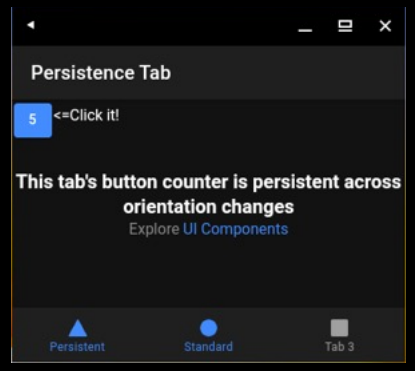
Large screen ready

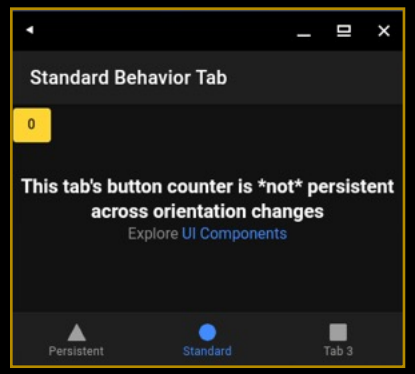
Saving UI States – Ionic sample app

- Ionic standard behavior is not automatically saving UI states

- To achieve persistence on window events, so while working with ZWC
  - Import { Storage } from '@ionic/storage-angular';
  - Initialize storage with a call to storage.create()
  - Save key/value pairs with await this._storage?.set(key, value);
  - Retrieve a value in the ionViewDidEnter() – called at load time - with a call to await Promise.resolve(this._storage?.get(key));

- Reference https://github.com/NDZL/ionic-ang-cap-PersistOnReload/blob/main/tab1.page.ts





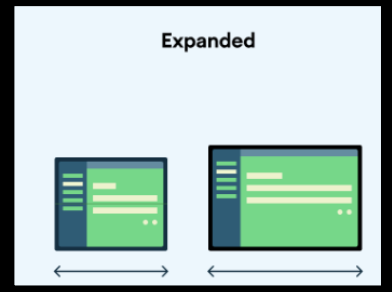
Zebra DevCon 2023

Large screen *optimized*

Large screen optimized
Checklist

Zebra DevCon 2023

- Reference https://developer.android.com/guide/topics/large-screens/get-started-with-large-screens#large_screen_optimized
- **Support different screen sizes:** use window size classes (compact, medium, expanded) to design the UI
  - Classes are identified by dp breakpoints (width/height < a fixed dp number)
    Design your UI using density-independent pixels (dp) as your unit of measurement
    Vertical scrolling ➔ Available width is more important than available height
  - Classes allow avoiding referencing the device type (phone, tablet, external display,…)
- Adaptive/Responsive UIs: fit a wide range of screen sizes as well as different orientations and device states
  - **Flexibility:** making optimal use of the available space and adjusting when the available space changes
  - **Continuity:** seamless user experience while transitioning from one window size to another
  - **Best layout:** ConstraintLayout – relative positions, spatial relationships
    LinearLayout is ok but heavier when nested
- **Avoid**
  - Locking your app to a specific orientation or aspect ratio
  - Trying to determine whether it's a phone or tablet
- "a11y" (Accessibility) – Guarantee navigation by Tab, Arrow keys and Keyboard shortcuts
- External input devices should provide superior UX: mouse right-click for contextual menus, wheel for zoom



Zebra DevCon 2023

Zebra Workstation Connect

Zebra Workstation Connect (ZWC)

Introduction

- What is Workstation Connect?
  - Workstation Connect is a powerful solution that allows you to quickly and easily turn Zebra mobile computers and tablets into workstations and POS stations, on demand.
  - It's made of
    - **A piece of hardware** – a kind of hub where an external monitor, mouse, or keyboard can be connected
    - **An app** (com.zebra.workstationconnect.release) available on Google Play or the Zebra website – this is needed to configure ZWC using JSON profiles and managed configurations
  - Additional tools for ZWC
    - Stagenow
      - To set the external monitor mode Desktop/Mirror
      - To allow a 3rd party app to invoke ZWC APIs, if not done programmatically
    - Zebra License Manager app or Stagenow to license budget line devices (e.g. ET4x, TC21, TC26)

Zebra Workstation Connect (ZWC)

Three personas - Configuration rules – The “swim lanes”

Zebra DevCon 2023

- Workstation Connect allows configuration changes to be made using two parallel paths
  - An administrator sends Managed Configurations to Zebra Workstation Connect using an EMM
  - A developer submits JSON strings to Zebra Workstation Connect

  Any time a particular setting is configured through either path, the new value replaces any value previously configured for that setting
- It is recommended that only one path is used to configure any single setting to reduce the potential for confusion
  - Alternatively Administrator and Developer should strictly agree on each swim lane. E.g.
    - **Administrator** ← limits its action to configure global features (e.g. external display mode, UI feature accessible to users, any other global/static setting)
    - **Developer** ← limits its action to application contextual needs (e.g. screen resolution, shortcut creation for the current user, any other role-related/dynamic setting)
- In the UI, the change made by the device user is temporary. It remains in effect until the device gets undocked
  - Any value the device user sets is reverted to the configured value when a new session begins (which occurs upon the next docking event)
  - If both the administrator or developer has not set a configuration and the user does, the configuration does not change until the administrator, developer, or user changes it
  - The changes remain in effect across multiple sessions until the device user changes it or configures it through the administrator and developer paths

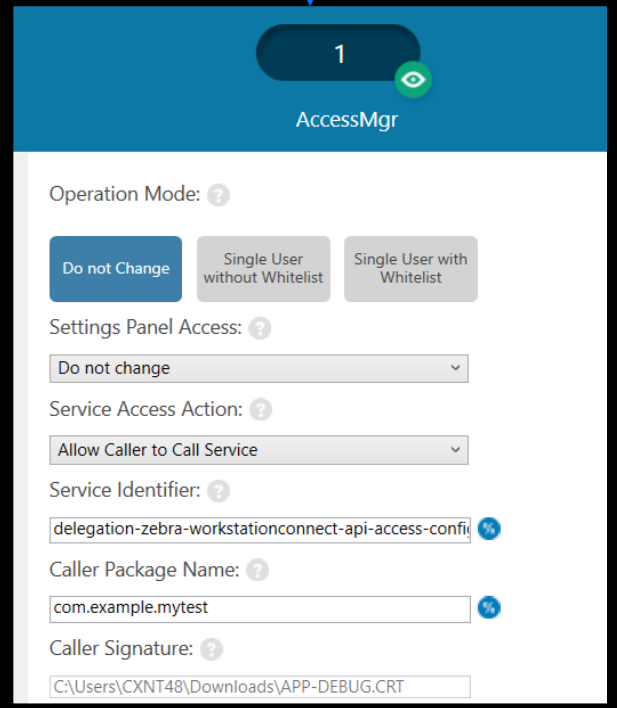
Zebra Workstation Connect (ZWC)

The Developer Use Case

Zebra DevCon 2023

Two stages to interface ZWC – Details in the ZWC Developer Guide

1. Authorize your app to call the ZWC developer interface.
   Options:
   - Stagenow – use Access Manager / Allow Caller to Call Service
   - EMDK – Same as Stagenow, apply a Profile calling Access Manager
     Details in this source code
     https://github.com/NDZL/ZVA_sERVICE_ALLOW_CALLER/blob/2201959672ac09cca88ca2aee6ac6ff1e4e84f02/app/src/main/java/com/zebra/adminapp/MainActivity.java#L124
   - Be sure the app you are going to authorize is already installed



2. Feed in your JSON Configuration to the developer interface
   - Bind the specific ZVA Service "com.zebra.workstationconnect.developerservice.DeviceManagementService“
   - Cast the binder to the service’s AIDL
   - Pass the JSON configuration by calling the **processZVARequest**(<json config>) method
     Details in the next slide and here
     https://github.com/NDZL/DEVCON23_ZWC/blob/3e998bcf3157eb787fd8ef38eba68014bb58a956/app/src/main/java/com/ndzl/zwc/HDLauncherActivity.java#L89 v

Zebra Workstation Connect (ZWC)

Building your JSON Configuration

Zebra DevCon 2023

- JSON basic schema to pass to **processZVARequest** ➔ like the one on the left-hand side of this slide (“appRestrictionsSchema” - valid for the ZWC app currently available)
- To import desktop shortcuts, add the **configDesktopShortcuts** key (right-hand side) to the **managedProperty** array
- configDesktopShortcuts valueString is obtained from an Export action on the secondary display. You need to escape quotes when pasting them in the outer JSON. Drop the checksum if values are changed since exporting.

```
{
	"kind": "androidenterprise#appRestrictionsSchema",
	"productId": "com.zebra.oemconfig.common",
	"managedProperty": [{
			"key": "desktopOrientation",
			"valueString": "0"
		},
		{
			"key": "persistConfiguration",
			"valueString": "1"
		},
		{
			"key": "resetConfiguration",
			"valueString": "1"
		},
		{
			"key": "configurationMode",
			"valueString": "Immediately"
		},
		{
			"key": "zwcConfigurationVersion",
			"valueString": "3003"
		}
	]
}
```

```
{
	"key": "configDesktopShortcuts",
	"valueBundle": [{
		"key": "shortcutImportFile",
		"valueString":
"[{\"addedDatetime\":1688272393625,\"appInfo\":{\"activityName\":\"com.android.camera.CameraLauncher\",\"iconImageString\":null,\"imagePath\":null,\"packageName\":\"com.symbol.zcam\"},\"groupInfo\":null,\"id\":\"sc2023.07.02.06:33:13.6e11e7\",\"modifiedDatetime\":1688272393626,\"shortcutInfo\":null,\"type\":\"application\"},{\"addedDatetime\":1688272404815,\"appInfo\":{\"activityName\":\"com.android.calculator2.Calculator\",\"iconImageString\":null,\"imagePath\":null,\"packageName\":\"com.google.android.calculator\"},\"groupInfo\":null,\"id\":\"sc2023.07.02.06:33:24.f06c04\",\"modifiedDatetime\":1688272404815,\"shortcutInfo\":null,\"type\":\"application\"},{\"addedDatetime\":1688272447787,\"appInfo\":{\"activityName\":\"com.google.android.keep.activities.BrowseActivity\",\"iconImageString\":null,\"imagePath\":null,\"packageName\":\"com.google.android.keep\"},\"groupInfo\":null,\"id\":\"sc2023.07.02.06:34:07.07eba0\",\"modifiedDatetime\":1688272447791,\"shortcutInfo\":null,\"type\":\"application\"}],{\"checksum\":\"306048d9ad9f503f952f3e8312e0d3ea\"}"
	}]
}
```

Drop “checksum” if the above values are changed

Zebra Workstation Connect (ZWC)

Roadmap

Zebra DevCon 2023

Next big feature of ZWC

- “**Secure mass deployment** of workstation connect managed app configurations via a JSON file delivered by SSM.”
  - This aims at making life easier for Administrators who need to manage several remote units via EMM
  - Avoids managed configurations on the EMM UI
  - Check the Lifeguard updates release notes to learn when this feature has been delivered

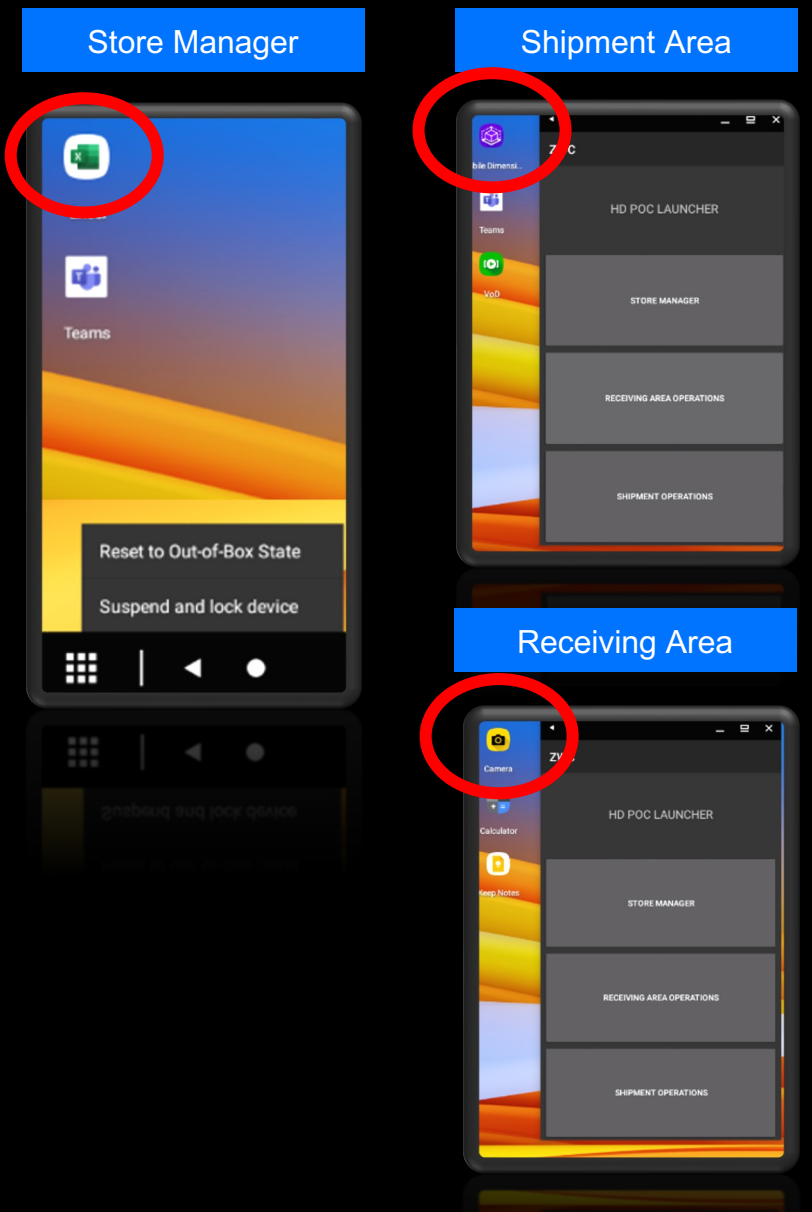
Zebra DevCon 2023

Workstation At Work

Workstation Connect Use Case

One Configuration – Multiple Roles

- Recall the configurations' *swim lanes* mentioned earlier
- Use case
  - One App Launcher – with role-based login, Apps launch on the external display when available
  - 3 roles: Store Manager, Worker in the Receiving Area, Worker in the Shipment Area
  - Depending on the user's role logged in, different ZWC Configurations are applied at runtime
    - ✓ The Launcher is shown on the external display as soon as it's available
    - ✓ On the external display only role-relevant links are shown
    - ✓ LOB Apps preferably run on the external display
- Benefits: each user is focused on their specific job, distractions are lowered, and user actions are sandboxed though still flexible



Workstation Connect At Work

Source code https://github.com/NDZL/DEVCON23_ZWC

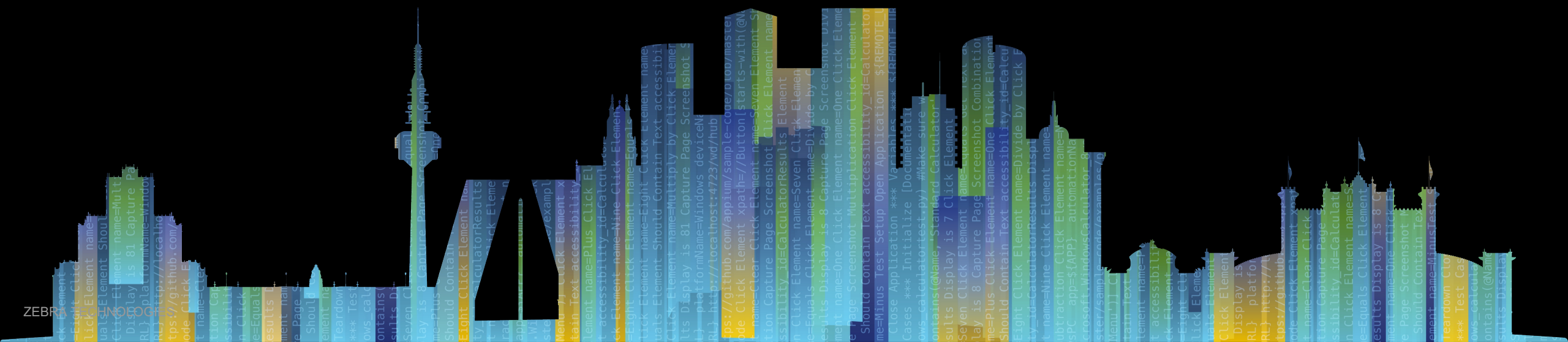
Zebra DevCon 2023

- The Launcher is shown on the external display as soon as it's available ➔ Register a Display Listener and move the window to the external display
  - ```
    DisplayManager displayManager = (DisplayManager) getSystemService(Context.DISPLAY_SERVICE);
    displayManager.registerDisplayListener(new DisplayManager.DisplayListener() {
        @Override
        public void onDisplayAdded(int displayId) {  //As a multi-screen-aware app, detect a 2nd screen when available
    ```
  - ```
    //move this launcher to the 2nd screen when available
    finish();
    ActivityOptions ao =ActivityOptions.makeBasic();
    ao.setLaunchDisplayId(displayId);
    Bundle bao = ao.toBundle();
    starterIntent.setFlags(  Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(starterIntent, bao);
    ```
- On the external display only role-relevant links are shown
  - ```
    String dataSet = loadJSONFromAsset( jsonConfig );
    String response = iServiceBinder.processZVARequest(dataSet);
    ```

- LOB Apps preferably run on the external display
  - Check this document https://source.android.com/docs/core/display/multi_display/activity-launch
  - Manage at runtime where apps are displayed with setLaunchDisplayId(..)
    ```

Zebra DevCon 2023

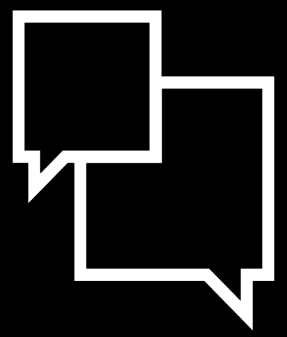
Recap



Recap

Zebra DevCon 2023

What	Why
Android Large Screen APIs	• Learn the foundations of multi-screen / multi-window systems • Migrate your code by tiers to support large screens
Zebra Workstation Connect	• Leverage Zebra's solution to upgrade your customer's experience
A Warehouse PoC	• Combine your large screens knowledge to build productive multi-role apps



Questions

Zebra DevCon 2023

ZEBRA

Thank You