Zebra
DevCon 2023

Zebra Pay

Miriam T. Ayala
Sr. Mgr. SW Engineering,
Enterprise Mobile Computing

Ian Hatton
Pre Sales Consultant Zebra
EMEA

Daniel Neamtu
EMEA Developer Advocate

Agenda

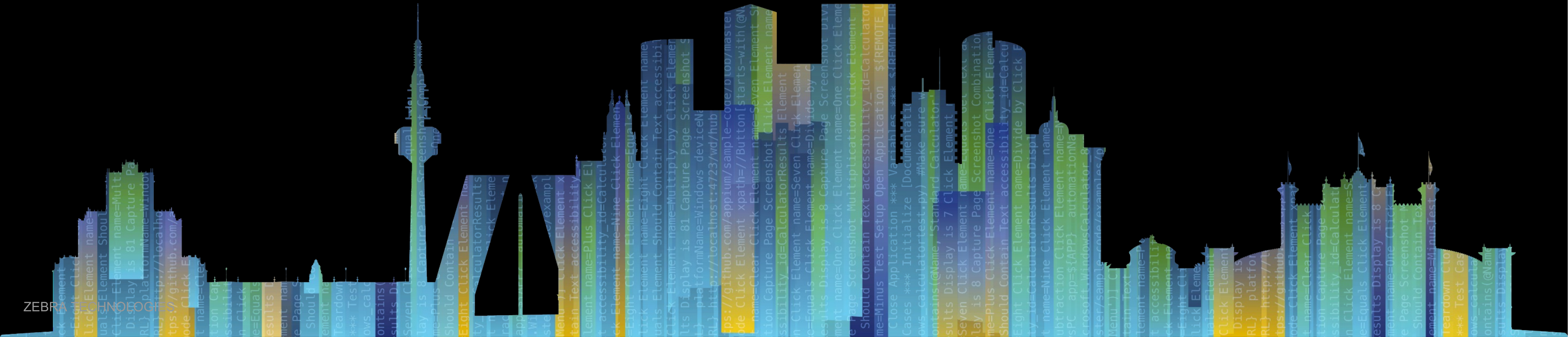
Zebra Pay

Zebra DevCon 2023

- Introduction to Zebra Pay
- Solution Architecture
- Zebra Pay Merchant Portal
- Zebra Pay Application Integration
- Notes, tips and references
- Questions

Zebra
DevCon 2023

Introduction to Zebra Pay



Enterprise PCI certified SPoC mobile payment solution

Everything you need to accept any card-based transaction, anywhere

Accept any type of transaction

Tap to pay with a digital wallet or NFC-enabled card, insert a chip-based card, accept PIN entry on screen, swipe an MSR card

Accept all major card brands

EMV certified to accept all major card brands*: Visa, Mastercard, American Express, Discover and Union Pay

Single pane of glass

Complete every payment transaction through the Zebra mobile device screen — including pin entry

Fits in one hand

Provides workers with payment simplicity — no need for a separate keypad for PIN entry or clunky BT connectivity. USB connection to payment reader provides stable connectivity.

*Supported card brands vary by acquirer and region

Easy to deploy

Everything you need for fast, easy and painless deployment

Pre-certified for specific markets

All the required certifications are included, and maintained — from acquirers to PCI security compliance, including EMV L3 certified for card brands with Global Payments, Worldpay* and Fiserv*. Since SPoC solution certification applies to Android OS version for COTS device, no per-device certification is needed

Easy-integration Zebra Pay API

Easily integrate any POS app with the PCI certified Zebra Pay app, alleviating your need to do so

Merchant Sandbox for easy testing

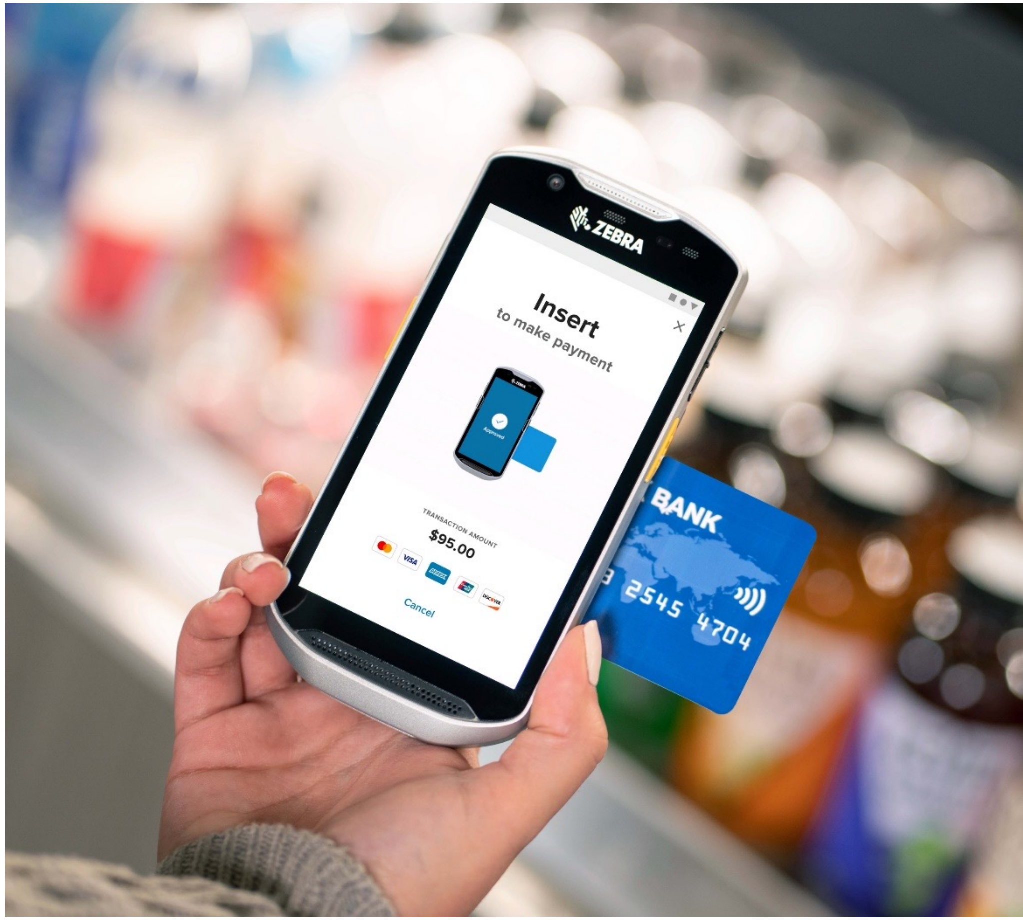
Easily test your Zebra Pay solution with the Merchant Sandbox, and you're ready to deploy — no waiting for certifications

Host integration included

Includes certified direct host integration with acquirers, starting with Global Payments in EU, (and soon to expanding into NA with Worldpay and Fiserv), eliminating development, integration and gateway fees

Complete device flexibility — no device lockdown required

Unlike other handheld mobile payment solutions, you can continue to use any apps on your Zebra mobile devices as well as install and manage Google Play and other apps via your EMM



*Zebra Pay for North America, supporting World Pay and Fiserv coming soon

Zebra Pay at work in …

Retail

Zebra DevCon 2023

Retail

Supercenters
Discount stores
Department stores
Specialty stores
Convenience stores

Applications

Line busting	Returns
Assisted selling	Pop-up sales
Off-site/outdoor purchasing	Order online – pay in-store or curbside

Zebra Pay at work in …

Hospitality

Zebra
DevCon 2023

Hospitality

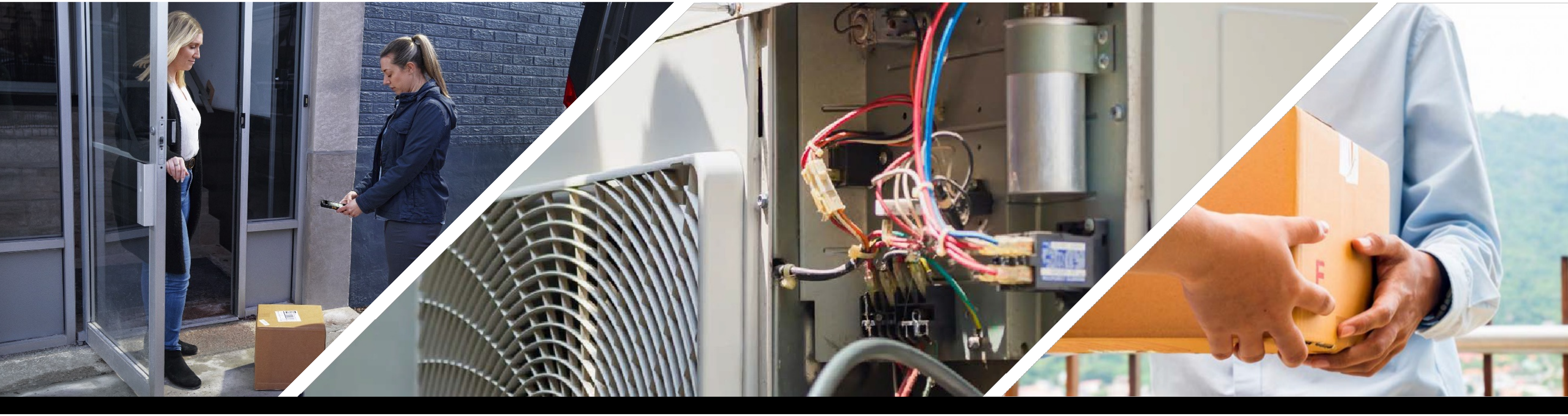
Amusement parks
Restaurants
Stadiums
Hotels
Airports

Applications

Ticket sales	Concierge sales
Pop-up sales	Line busting
Table-side payment	Service upgrades / pay at gate

Zebra Pay at work in …

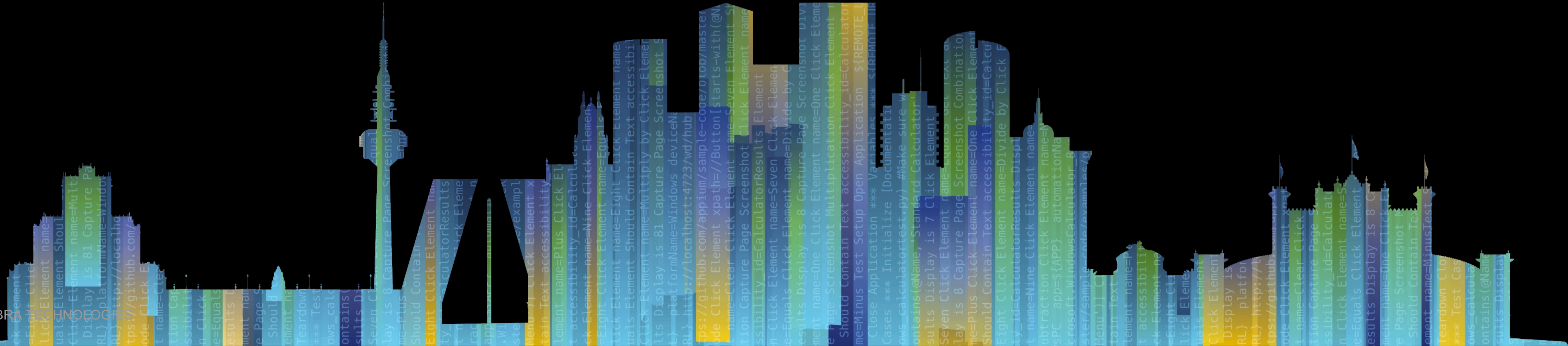
Field Services/T&L



Field Services	**Service calls** **Field services**	**Applications** Payment for services and parts
Transportation & Logistics	**Delivery** **Courier services** **Postal Services**	**Applications** Payment on delivery

Zebra
DevCon 2023

Zebra Pay Architecture



Zebra Pay Solution Components

PCI-SPoC Certified Enterprise Mobile Payment Solution

Zebra DevCon 2023

The Zebra Pay solution built for Zebra mobile devices using the Zebra PD20 card reader enables the collection and management of digital/card-based payments. Primary use case target: retail workflows.

Software Pin Entry on Zebra Mobile devices (SPoC) with Zebra Pay APK
Zebra Pay Android Application, expected to be available from Google Play Store, enables Credit and Debit acceptance on the TC52x, TC57x, TC53/TC58 running Android 10, and A11. [ET40/ET45 – coming soon].

Zebra Mobile Device Payment back or Payment Battery accessory
Elegantly connects Mobile Device to PD20 for a comfortable user experience

---

Zebra PD20 PCI-PTS certified Secure Card Reader supports various payment input modalities (insert, tap, and swipe)
Chip, NFC (Apple Pay, Google Pay, NFC Cards), or MSR

Zebra Pay Android API
Enables quick and easy integration of Zebra Pay with customer or integrator applications.

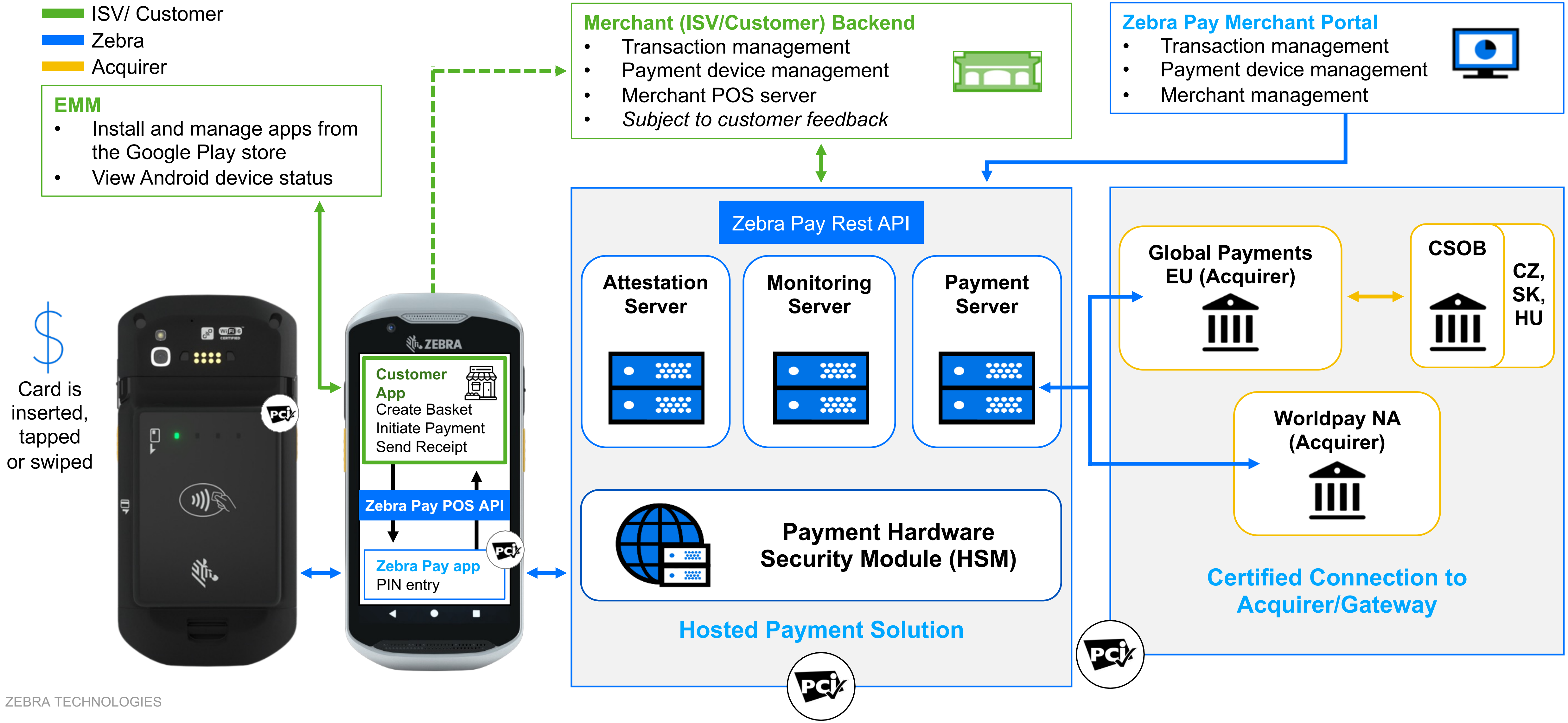
Zebra Pay Portal (Merchant & Admin)
The Merchant Portal provides tools and visibility to manage mobile devices, reader, user access, and transactions.

Zebra Pay Rest API ***(coming soon)***
The Rest API provides customers with the ability to pull device, reader, and transaction data from the Zebra Pay database in order to support customer cloud applications.

Zebra Pay architecture

Zebra Mobile Device with Zebra Pay

Zebra DevCon 2023



🟩 ISV/ Customer
🟦 Zebra
🟨 Acquirer

EMM
- Install and manage apps from the Google Play store
- View Android device status

Merchant (ISV/Customer) Backend
- Transaction management
- Payment device management
- Merchant POS server
- *Subject to customer feedback*

Zebra Pay Merchant Portal
- Transaction management
- Payment device management
- Merchant management

$ Card is inserted, tapped or swiped

Customer App
Create Basket
Initiate Payment
Send Receipt

Zebra Pay POS API

Zebra Pay app
PIN entry

Zebra Pay Rest API

Attestation Server

Monitoring Server

Payment Server

Payment Hardware Security Module (HSM)

Hosted Payment Solution

Global Payments EU (Acquirer)

CSOB

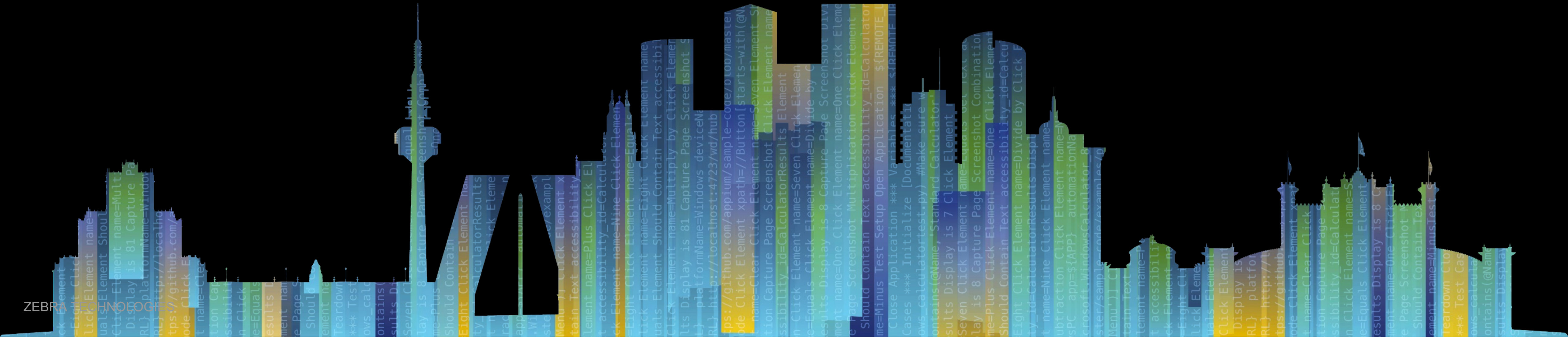
CZ, SK, HU

Worldpay NA (Acquirer)

Certified Connection to Acquirer/Gateway

Zebra
DevCon 2023

Zebra Pay Merchant Portal



Zebra Pay Merchant Portal

Easily manage all aspects of system usage

Zebra
DevCon 2023

Manage companies

View all store locations or a specific store

Easily onboard new stores

Manage mobile users, readers, transactions and credit settlements

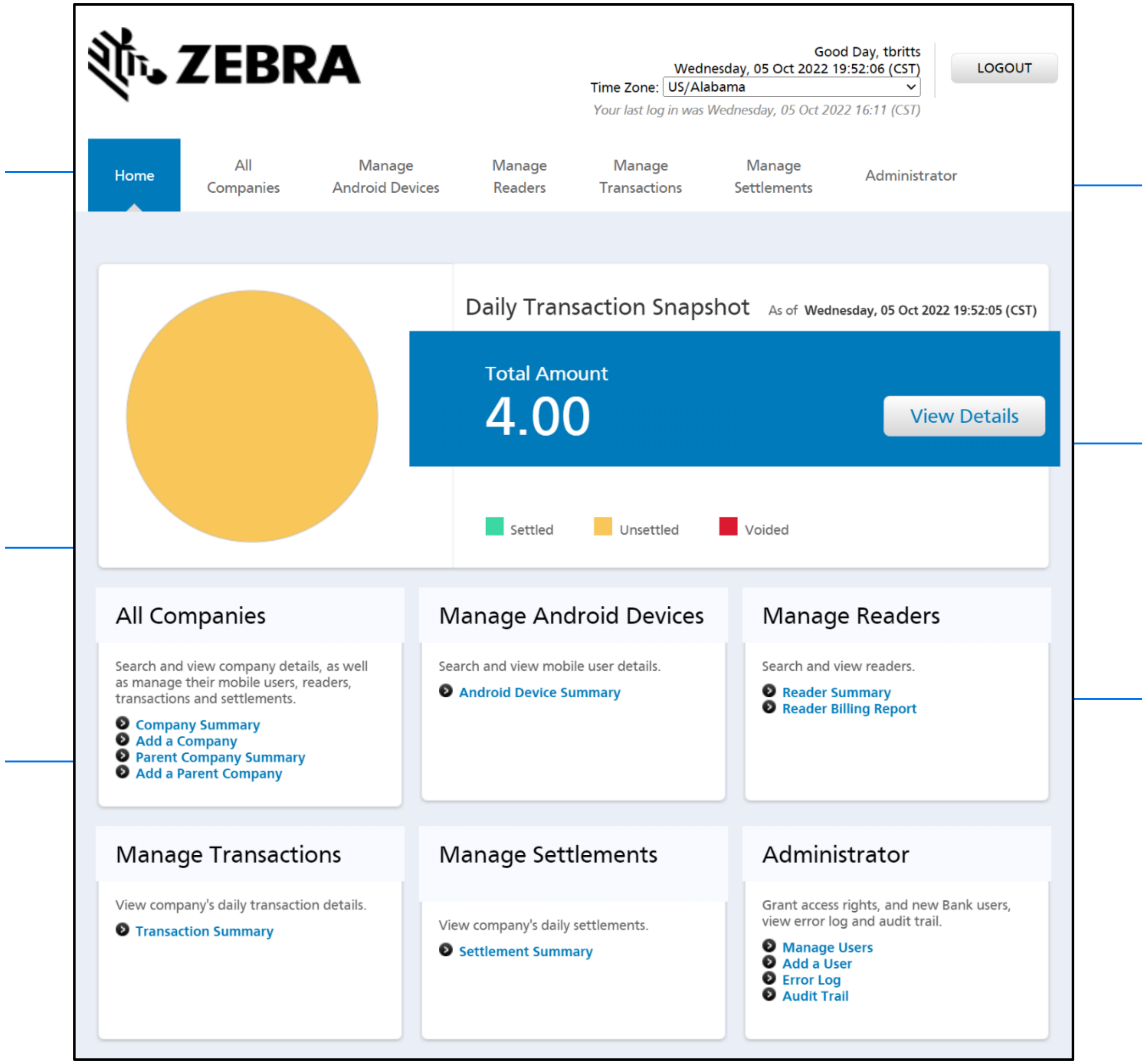
Manage Zebra Android devices

View all mobile user details

Manage readers

Monitor reader status

View, activate or suspend readers as needed



ZEBRA

Good Day, tbritts
Wednesday, 05 Oct 2022 19:52:06 (CST)
Time Zone: US/Alabama
Your last log in was Wednesday, 05 Oct 2022 16:11 (CST)

Daily Transaction Snapshot As of **Wednesday, 05 Oct 2022 19:52:05 (CST)**

Total Amount
4.00

🟩 Settled 🟨 Unsettled 🟥 Voided

All Companies

Search and view company details, as well as manage their mobile users, readers, transactions and settlements.

- ● Company Summary
- ● Add a Company
- ● Parent Company Summary
- ● Add a Parent Company

Manage Android Devices

Search and view mobile user details.

- ● Android Device Summary

Manage Readers

Search and view readers.

- ● Reader Summary
- ● Reader Billing Report

Manage Transactions

View company's daily transaction details.

- ● Transaction Summary

Manage Settlements

View company's daily settlements.

- ● Settlement Summary

Administrator

Grant access rights, and new Bank users, view error log and audit trail.

- ● Manage Users
- ● Add a User
- ● Error Log
- ● Audit Trail

Manage transactions

View the daily details of transactions Export Transaction to .csv

View settlements

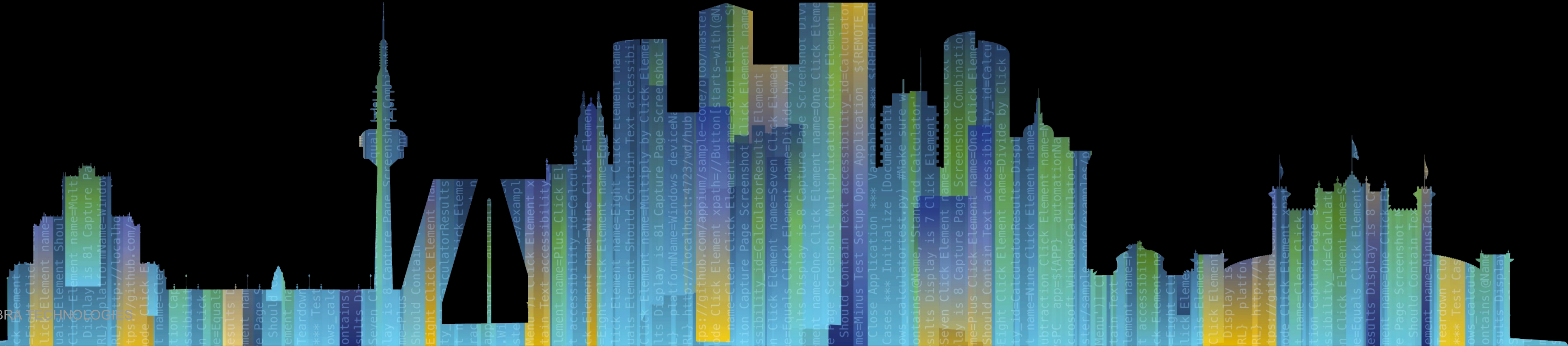
View the details of daily credit settlements

Administrator

Grant access rights
Add new users
View error logs and audit trail

Zebra
DevCon 2023

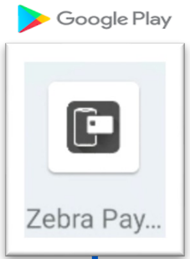
Zebra Pay Application Integration

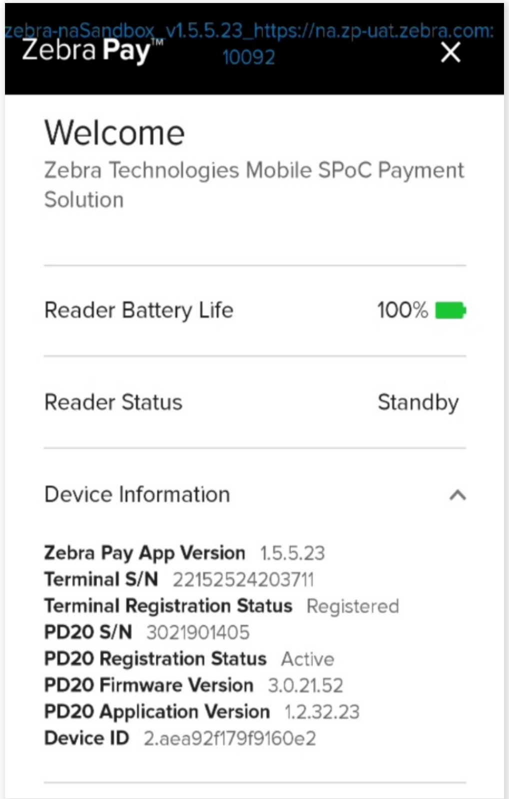


Zebra Pay Android Application

Installation and App Details

- Zebra Pay production application is only available from Google PlayStore; apk it is *not* available for sideloading.
- Devices and OS's Supported:
  - TC52x, TC52ax,T57x – A11, A10 (LG-June 2022 and later)
  - TC53, TC58 - A11
- Variants of App for Region (EU vs. NA) and Environment (Sandbox-Test vs. Production)
- Launching Zebra Pay app from the launcher brings up details UI with device and reader solution details
- PD20 FW (FW & App) is contained in the Zebra Pay App. upon launch, Zebra Pay will automatically connect to PD20 reader, check FW version, and update FW if required.
- Attestation checks validate that APK version is current and within acceptable date range. If outside permitted range, an attestation error will occur, and functionality will be blocked. Release notes will indicate if an update is mandatory and timeframe to transition.
  - It is strongly encouraged to validate new releases in Sandbox environment and update production units in a timely fashion.





Welcome

Zebra Technologies Mobile SPoC Payment Solution

Reader Battery Life 100%

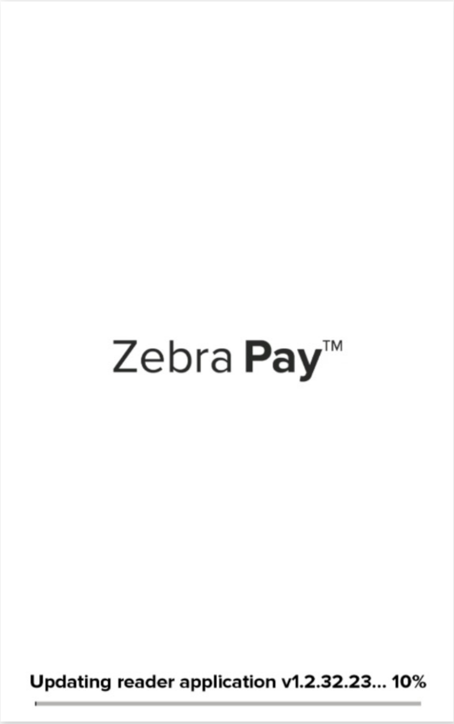
Reader Status Standby

Device Information

Zebra Pay App Version 1.5.5.23
Terminal S/N 22152524203711
Terminal Registration Status Registered
PD20 S/N 3021901405
PD20 Registration Status Active
PD20 Firmware Version 3.0.21.52
PD20 Application Version 1.2.32.23
Device ID 2.aea92f179f9160e2

Production:
com.zebra.zebrapay.eu.release
Sandbox:
com.zebra.zebrapay.eu.sandbox.release

Customer App



Updating reader application v1.2.32.23... 10%

Zebra Pay Application

Deployment Checks

- The Zebra PD20 Secure Card Reader may be used/shared across mobile devices (approval required on portal per company)
- Mobile device may only be registered to one Company (MMK) and one MID/TID at a time.
- Zebra PD20 Reader contains special HW and SW checks exists to ensure the integrity of the reader to protect against and check for tamper.
- HW must be registered (via POS API) and Approved (via Portal) before any Payment transactions will be processed
- Due to strict PCI-SPoC security measures, the following conditions must be met else payment transactions may fail:
  - Zebra Pay UI may not be overlayed
  - Uninterrupted application session
  - ADB debugging and developer settings must be disabled in production
    - Currently USB must be disabled for security purposes for Sandbox release app. We are looking at approaches to allow ADB to be enabled during development.
  - Mobile device integrity ensured (non-rooted device, etc.)
  - Camera must be disabled
  - Screen recording disabled
  - HW has not been tampered

Zebra Pay

High Level Application Flow

Staging:

Company and user set up, Customer & Zebra Pay App deployed to mobiles from Google Playstore, Firewall configuration, mobile device and reader registration and approvals

Transactions:

- POS Application* Scans Shoppers merchandise, calculates total and fires intent to launch Zebra Pay
- Zebra Pay presents Payment screen to Shopper requesting presentation of card
- Shopper presents payment card
- Zebra Pay UI displays card brand animation with jingle and/or notification screen (configuration settings)
- Zebra Pay via Intent call, delivers response to Customer Application
- POS Application generates and presents receipt

Customer App

Scans merchandise, calculates total amount to charge

Zebra Pay Android API

When ready to collect payment, app will fire intent with amount and credentials.

Zebra Pay launches and present UI. Shopper to presents card to the Zebra PD20 Reader, enters pin on mobile if required), and payment is collected.

*"POS Application" refers to the developer's application that interfaces with Zebra Pay application. See arch slide above.

Zebra Pay Android Library

Android APK and Native Library

- Customer Android App will use Zebra Pay Android POS API to configure and send **Android Intent to Zebra Pay** to:
  - Register Mobile Device and PD20 Secure Card Reader
  - Process Payment Transactions (Payment, Refund, or Void)
- Zebra Pay Android Native SDK: ssmobilethirdpartyintegration-x.xx.x.aar
- Zebra Pay asynchronously sends and Android call-back Intent to the customer Application with request results
- Key Configuration parameters include:
  - **Transaction Type**
  - **Transaction Details (**e.g. amount, etc)
  - **Zebra Pay environment (Region: NA/EU, Test =Sandbox/Production)**
  - **Name of POS Application's Call Back Intent (Zebra Pay will send with response data)**
  - **Merchant Company MMK** (aka Developer ID) [obtained from Portal]
    - Alphanumeric string
    - Generated when "Company" is created in Portal and associated with a MID
    - Passed from Mobile device in order to register it with Company

Note: Zebra only supports a native Android SDK. Developers of web-apps can use 3rd party tools to develop android plugins that call the Zebra Pay SDK.

Zebra Pay Android SDK

Straightforward APIs

Zebra DevCon 2023

Available Types

- Registration – Used to register your reader and get it validated from the Zebra Pay Portal
- Payment – Used to create a normal payment transaction
- Void Payment – Used to cancel a referenced transaction only if the approved transaction hasn't been settled yet
- Refund Payment – Used to create a refund transaction

ThirdPartyIntegrationRequestVO - Various parameters to be populated based on the requested Action Type:

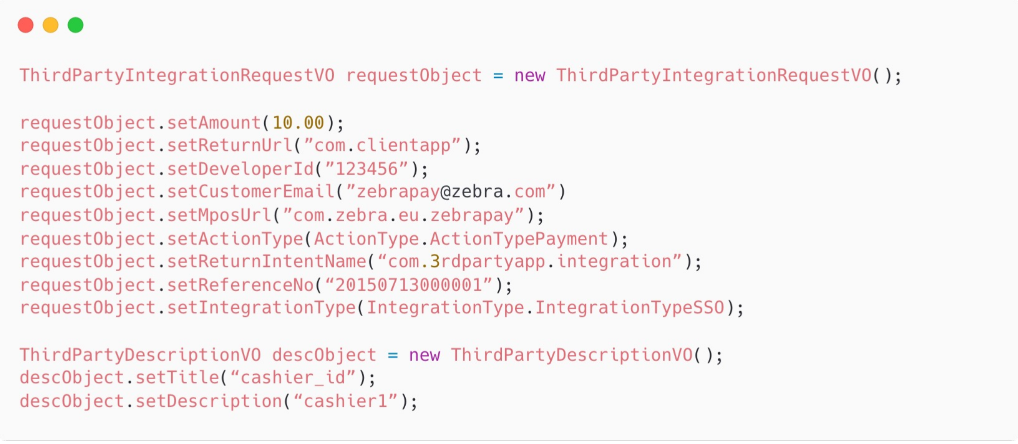
setActionType(<actionType>) – registration or payment type

Additional parameters (see example with sample fields)

ThirdPartyIntegration.requestMPOSIntegration(context, requestObject);

ThirdPartyIntegrationResponseVO

Sample Payment Request:

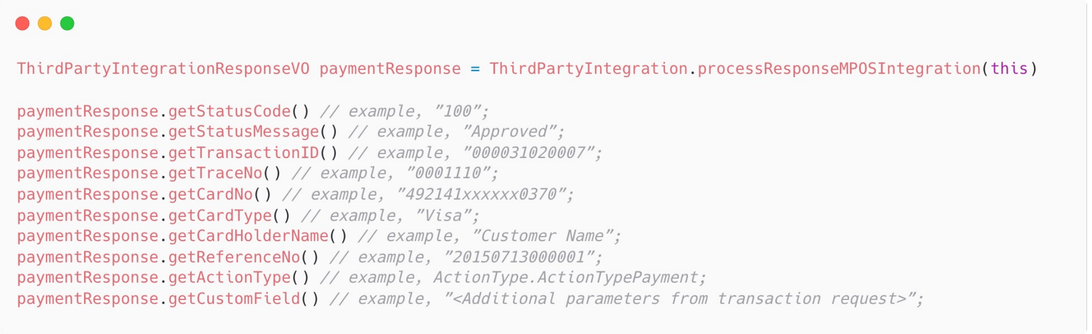


```
ThirdPartyIntegrationRequestVO requestObject = new ThirdPartyIntegrationRequestVO();

requestObject.setAmount(10.00);
requestObject.setReturnUrl(”com.clientapp”);
requestObject.setDeveloperId(”123456”);
requestObject.setCustomerEmail(”zebrapay@zebra.com”)
requestObject.setMposUrl(”com.zebra.eu.zebrapay”);
requestObject.setActionType(ActionType.ActionTypePayment);
requestObject.setReturnIntentName(“com.3rdpartyapp.integration”);
requestObject.setReferenceNo(“20150713000001”);
requestObject.setIntegrationType(IntegrationType.IntegrationTypeSSO);

ThirdPartyDescriptionVO descObject = new ThirdPartyDescriptionVO();
descObject.setTitle(“cashier_id”);
descObject.setDescription(“cashier1”);
```

Sample Response:



```
ThirdPartyIntegrationResponseVO paymentResponse = ThirdPartyIntegration.processResponseMPOSIntegration(this)

paymentResponse.getStatusCode() // example, ”100”;
paymentResponse.getStatusMessage() // example, ”Approved”;
paymentResponse.getTransactionID() // example, ”000031020007”;
paymentResponse.getTraceNo() // example, ”0001110”;
paymentResponse.getCardNo() // example, ”492141xxxxxx0370”;
paymentResponse.getCardType() // example, ”Visa”;
paymentResponse.getCardHolderName() // example, ”Customer Name”;
paymentResponse.getReferenceNo() // example, ”20150713000001”;
paymentResponse.getActionType() // example, ActionType.ActionTypePayment;
paymentResponse.getCustomField() // example, ”<Additional parameters from transaction request>”;
```

Zebra Pay Application UIs

Payment Processing

Zebra DevCon 2023

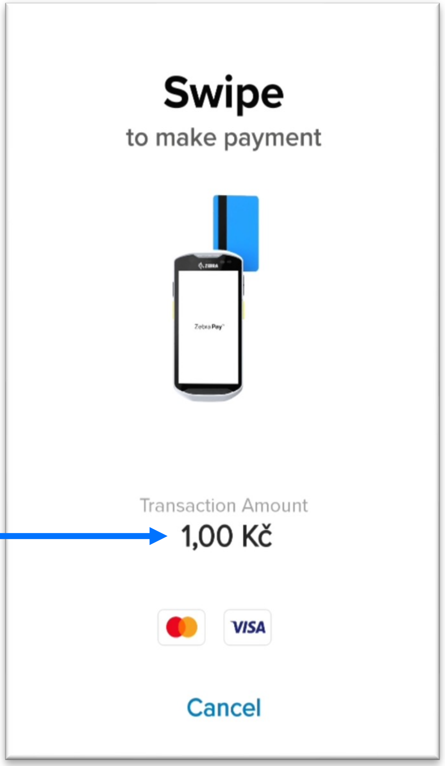
Supports Android language settings: English and Czech

Zebra Pay intent

POS App

Currency is derived from MID

"Tap", "Insert", "Swipe"



Swipe

to make payment

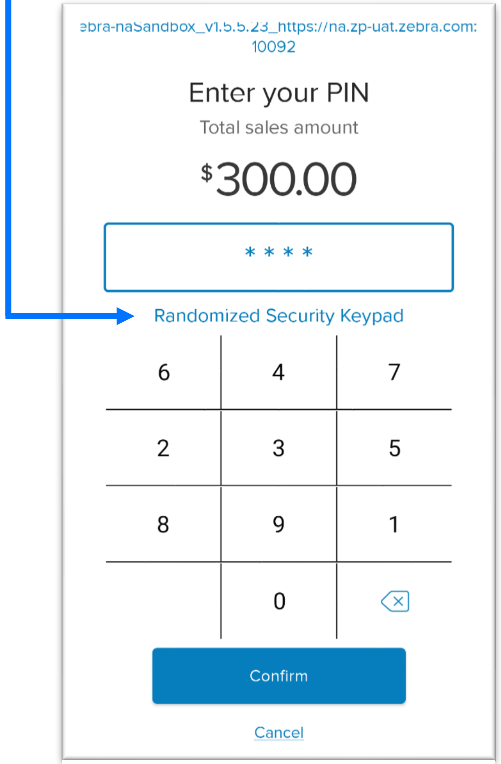
Transaction Amount

1,00 Kč

Cancel

Note: At Zebra Pay EU Rev A release only CZ country and currency supported.

Randomized Pin Entry (PCI-SPoC requirement)



ebra-naSandbox_v1.5.5.23_https://na.zp-uat.zebra.com:10092

Enter your PIN

Total sales amount

$300.00

* * * *

Randomized Security Keypad

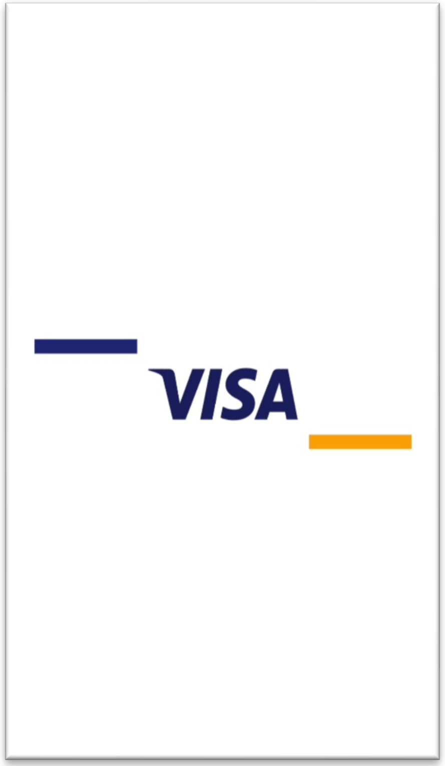
6	4	7
2	3	5
8	9	1
	0	⌫

Confirm

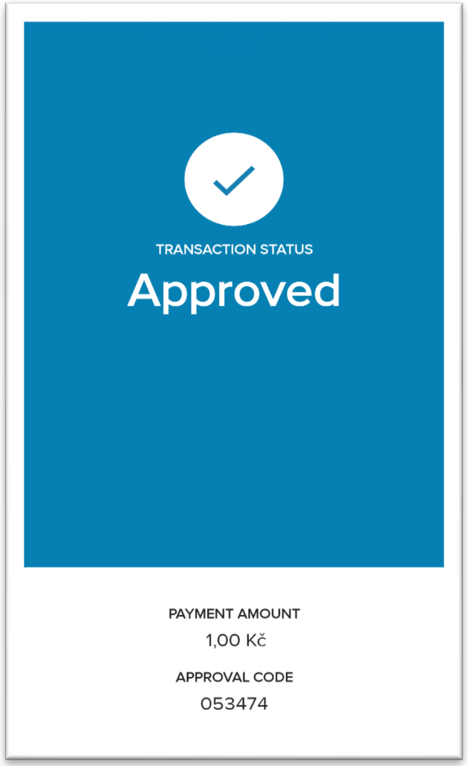
Cancel

Zebra Pay environment, version and region (.40=EU)

Optional Card Branding UI and accompanying jingle



Optional UI showing Approved, Declined, Remove Card



TRANSACTION STATUS

Approved

PAYMENT AMOUNT

1,00 Kč

APPROVAL CODE

053474

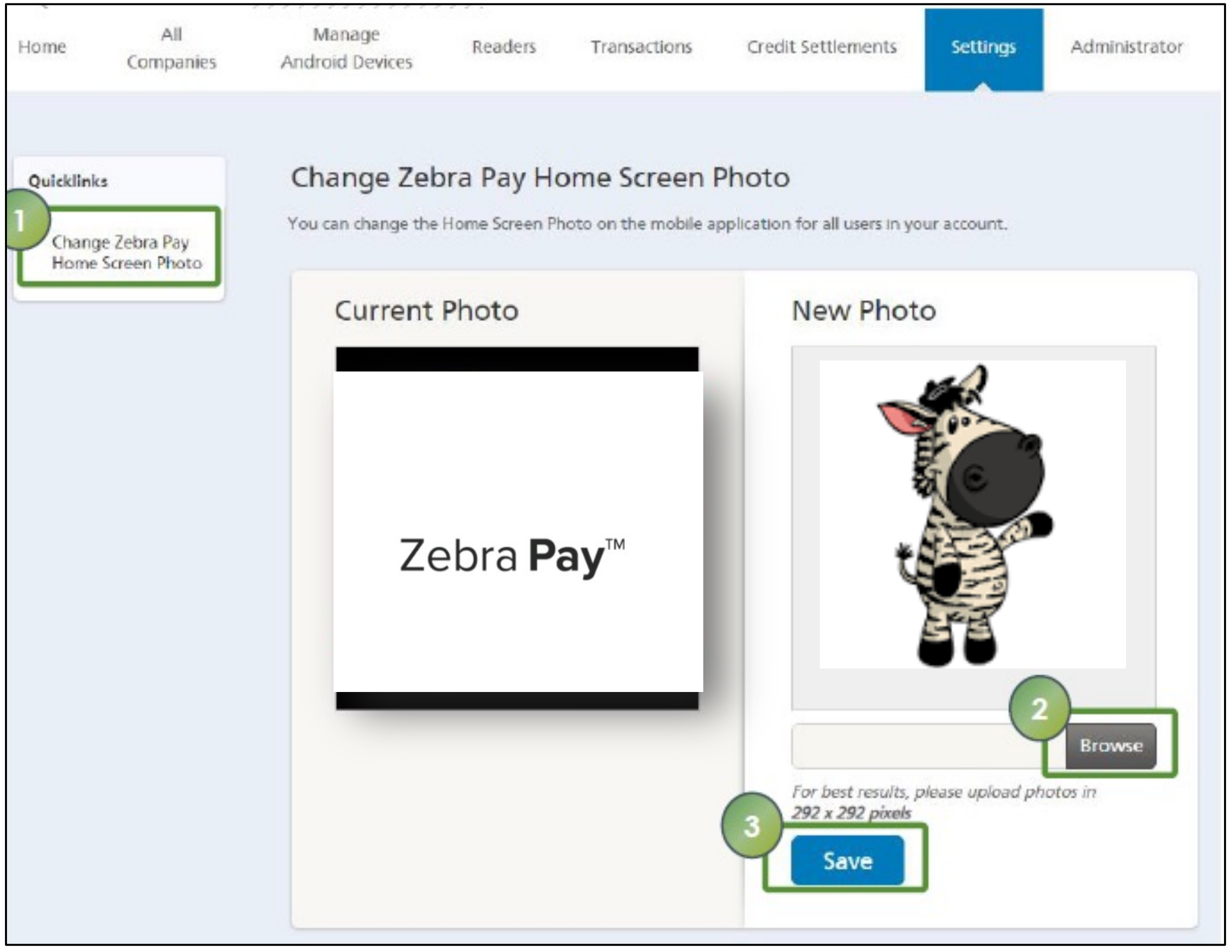
Note: Zebra Pay API returns transaction response to the calling application. Calling app controls UI thereafter.

Zebra Pay Application

Customizations

Zebra DevCon 2023

- addItemDescription(ThirdPartyDescriptionVO thirdPartyDescriptionVO)
  - By default, sensory branding is enabled (MC, Visa). To disable, set to true
    - thirdPartyDescriptionVO.setTitle("disable_sensory_branding"); thirdPartyDescriptionVO.setDescription("true");
- setMerchantData(String merchantData) – can be used for passing small string of data related to transaction that may be useful when reconciling settlement report
- Portal Driven Configuration of APK:
  - "Store Manager" role user can optionally customize the logo on the Zebra Pay Android App
  - The image will be shown during the Zebra Pay app loading screen
  - Minimum size: 292 x 292 pixels



Change Zebra Pay Home Screen Photo

You can change the Home Screen Photo on the mobile application for all users in your account.

For best results, please upload photos in 292 x 292 pixels

References

Zebra DevCon 2023

References: http://www.zebra.com/zebrapay

- Zebra Pay API Reference Guide
- Zebra Pay Solution User Guide – see FAQs
- ZebraPOSSample Android application source code available for reference to kick start development

Coming soon:

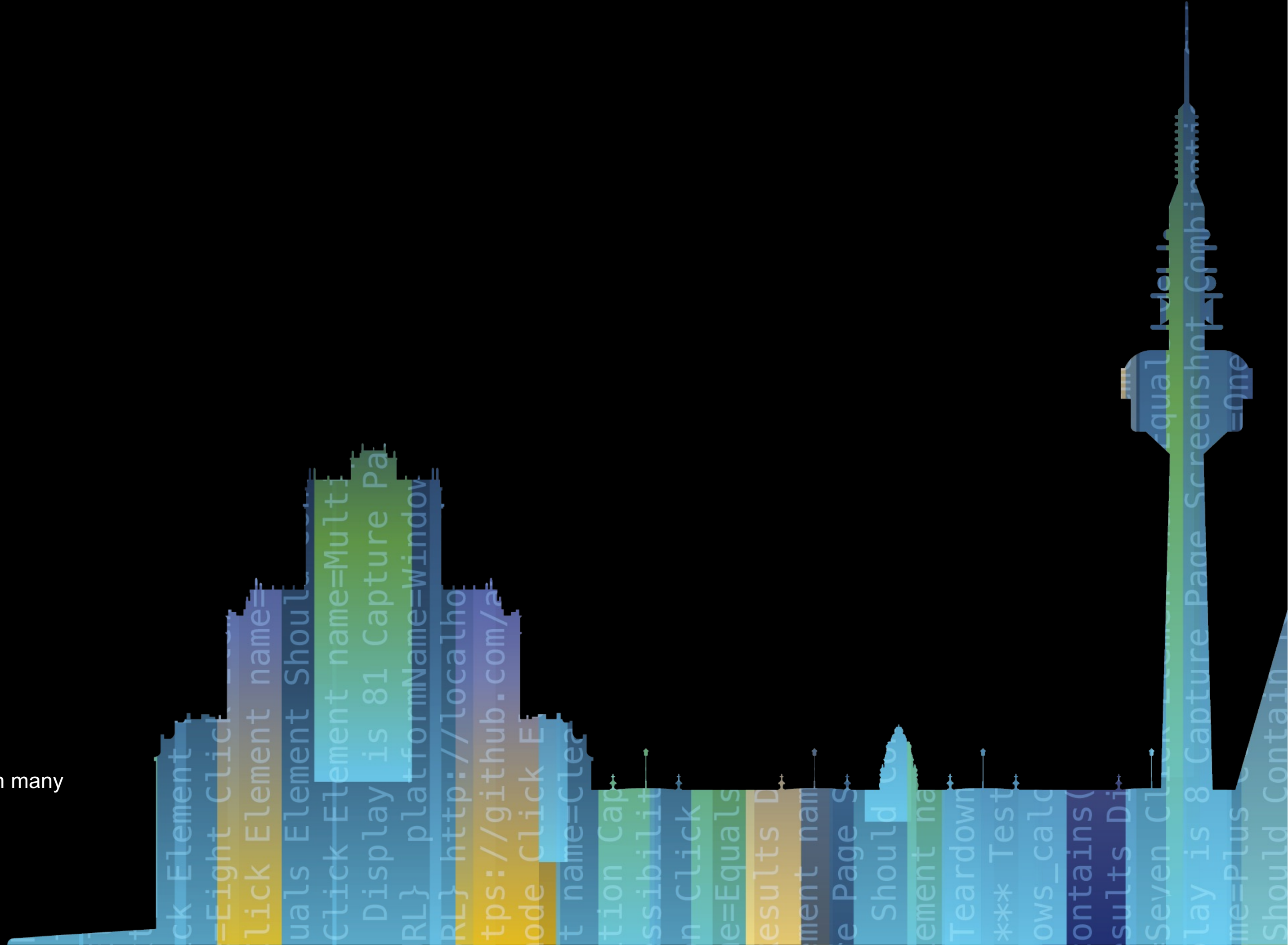
- Zebra Pay NA – WorldPay, Fiserv (future support for US)
- Expanding set of supported mobile devices
- Rest API – enables reading Mobile device, reader status and transaction details. In process of being vetted during customer Pilot. Note: Rest API supports read only functions and does not support any payment processing.

Questions

Zebra
DevCon 2023

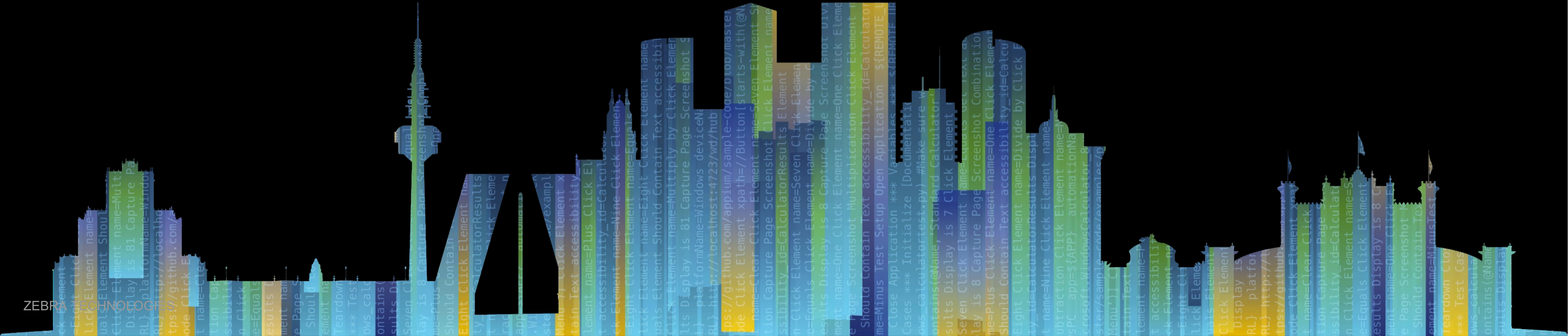
ZEBRA

Thank You



Zebra
DevCon 2023

Additional Developer Tips



Additional Developer Tips

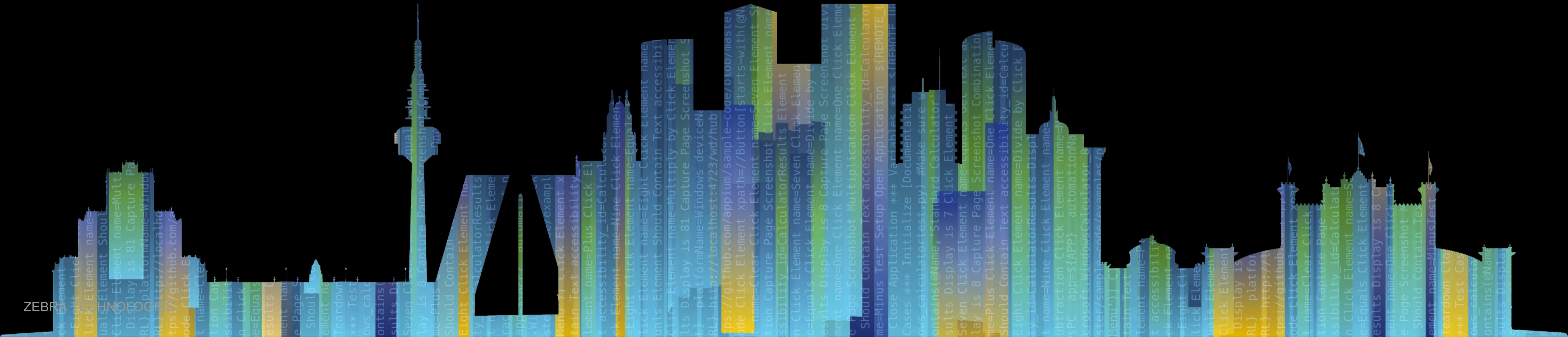
Common oversights and errors

Zebra DevCon 2023

- **StatusCode** responseObject.getStatusCode()]
  - StatusCode= if not 10 or 100, it's an error
- **Application/Solution**
  - Incorrect MMK (14015), mobile device was not registered (3029) or assigned TID in portal (3040), reader was not registered or approved (3032) invalid/missing parameter (see SDK), 6001=PD20 is not properly seated or insufficiently charged; Network Error on Mobile (-1001—1006) – check firewall settings, connection stability and network time is set on device
  - No response: ensure your config file has proper Call Back to receive Response intent from Zebra Pay into your app.
- **Attestation**
  - USB debugging and/or ADB was enabled, pop up/intervention of non-zebra pay app UI
  - PD20 Reader is tampered – reader was dropped; 1st and 4th LED flashing when power button pressed.
- **Host**
  - Card processing issue (e.g., card not supported, payment mode not supported etc.) .
  - Follow guidance provided in error and try again.

Zebra
DevCon 2023

Lifecyle from Development to Roll Out



Lifecycle from Development to Production

Onboarding and Development

Zebra DevCon 2023

- **Customer or ISV**
  - Engages with Zebra Pay Sales team
  - Obtains Test MID/TID credentials from Acquirer for use within Zebra Pay Sandbox (Test) environment
  - Purchases Zebra Mobile Device, Payment battery/accessory, PD20 Secure Reader (Test sku)
  - Zebra Pay Sales Team works with Zebra Product team to create Parent account in Sandbox Portal
  - Customer creates Company(s) in Sandbox Portal with their Test MIDs and associated TIDs
  - Customer shares Test Company MMK with their Development team
- **Application Development**
  - Zebra POS API call to initiate registration of mobile device
  - Upon success Customer’s Company Admin logs into Portal and assigns TID to available mobile device
  - Zebra POS API call to register reader
  - Upon success Customer’s Company Admin Approves the reader in Zebra Pay Portal
  - Zebra POS API calls to process (test) Payment transactions against the Company (MMK and validate application in test environment using Test Credit/Debit cards.
  - Review transactions on the Portal

Lifecycle from Development to Production

Ready for Production (Live transactions)

- When development and validation is Complete (in Sandbox) and customer is ready for Live Pilot or Production deployment, next steps:
  - Engages with Zebra Pay Sales team
  - Obtains Production MID/TID credentials from their Acquirer for use within Zebra Pay Production environment
  - Purchase Zebra PD20 Secure Reader (Production sku)
  - Zebra Pay Sales Team works with Zebra Product team to create Parent account in Production Portal
  - Customer populates Portal with their Production MID/TIDs and authorized user credentials
  - Customer shares MMK with Development or Deployment team
- Stage mobile devices deploying Zebra Pay (Production) APK from Google Playstore
- Customer app registers devices and readers – via Zebra POS API with appropriate MMK for use with the target Store's Mobile devices using Production MMK credentials
- Customer Manager/Admin – assigns TID credential to Mobile Devices and Approves Readers via the Merchant Portal
- Production devices are now ready to process live payment transactions