



powered by
alpha Transform[®]

CUSTOM APPLICATION PLATFORM FOR FIELD WORK MANAGEMENT

The speed of no-code plus the power of low-code



mazia|EVP[®]
MOBILE & WEB | INTERGRATIONS | REPORTING

FIELD WORK

What is the challenge?

Backoffice/Management has reduced visibility into what is happening in the field

Data collection, reporting and analysis is often delayed, inaccurate, and manually constructed

Legacy data access and capture methods have deficiencies

Typical software solutions can be complex to implement, or restricting to use

Being offline in the field cannot be avoided and should not hinder performance

OPTIONS FOR IMPLEMENTING A BUSINESS APPLICATION



OPTIONS FOR IMPLEMENTING A BUSINESS APPLICATION

Building or buying a solution					
	“Native” development 	Low-code development 	No-code build 	SaaS platform 	COTS application(s) 
Implementation time:	Months/years	Weeks/months	Hours/days	Hours/days	Hours/weeks/months/years
Expertise required:	Software developer	Low-Code developer	Basic IT knowledge	Basic IT knowledge	Varies
Flexibility of features:	Maximum	Significant	Limited	Fixed	Fixed
Lifecycle support effort:	Major	Limited	Limited	None	Limited/none

mazia|EVP[®]
MOBILE & WEB | INTERGRATIONS | REPORTING

powered by
 alpha Transform™



mazia|EVP[®]
MOBILE & WEB | INTERGRATIONS | REPORTING

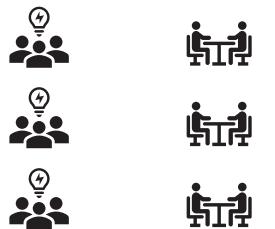
powered by
 alpha Transform™



COMPARING TRANSFORMWITH CUSTOM APP DEVELOPMENT

Standard mobile development					
Customer idea	Design	Expert dev team	• UI • Business logic • Mobile features • Offline • Local storage • Authentication • Database connection(s) • Link(s) to backend	Build/Test/Publish/Debug	Delivered mobile app
			 Weeks / Months to build Hours / Days to update	 (with risk of delay/failure)	

Multiple apps required ?



3x the apps:
3x the build time or 3x the teams

Low/no code mobile development using Alpha TransForm					
Customer idea	Design	Pre-built, data-driven mobile app	• Configure form data fields • Design form layout • Add custom workflows	Configure/Publish	Delivered mobile app
			 Hours to build Minutes to update		 No risk (testable instantly)

Multiple apps required ?

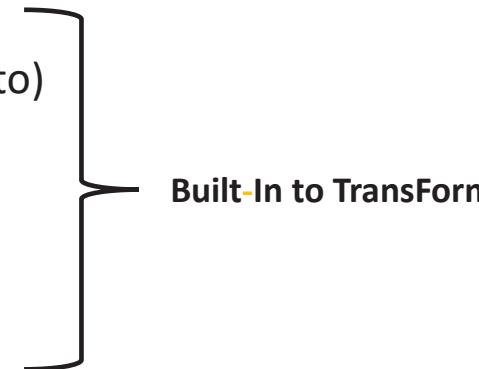


It is the same app !

CAN A PRE-BUILT MOBILE APP BE POWERFUL AND FLEXIBLE ENOUGH ?

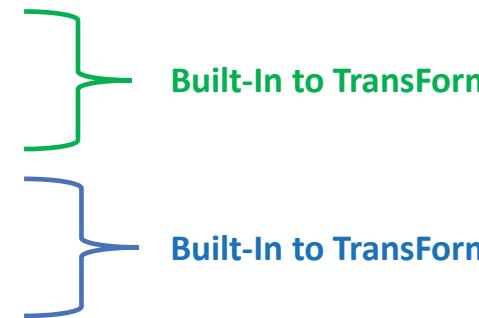
Standard Data Capture Apps need to include

- Various Field Types (Text, Date, Signature, Location, Photo)
- Connection to a Database
- User Interface – Responsive design
- Search / Sort / Filter of forms
- Authentication
- Android & iOS

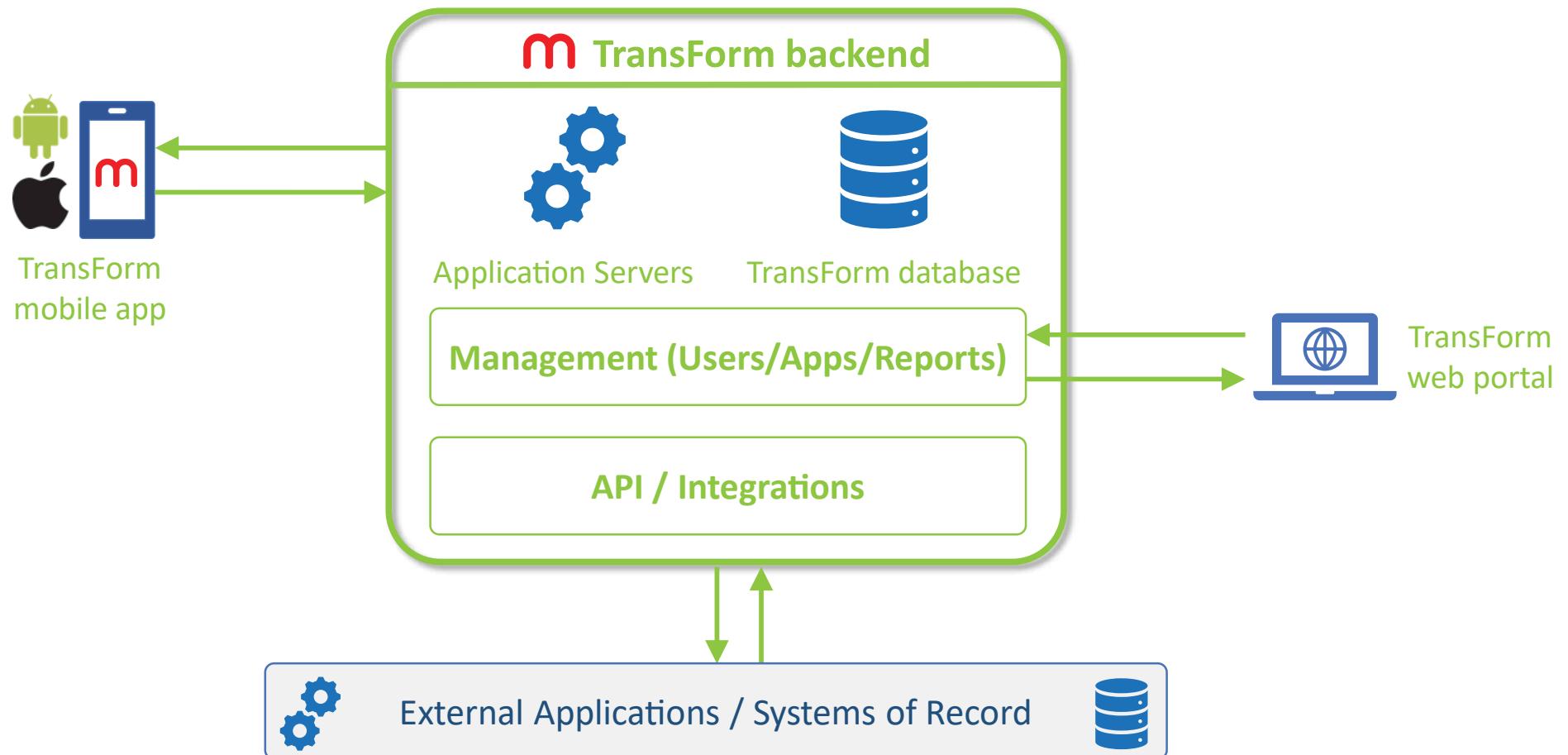


Advanced Data Capture Apps can include

- User Permissions
- API, Webhooks, and Triggers
- On-line & Off-line support
- In-app calculations and validations
- Look-ups; local and remote
- On-device Data (databases and files)



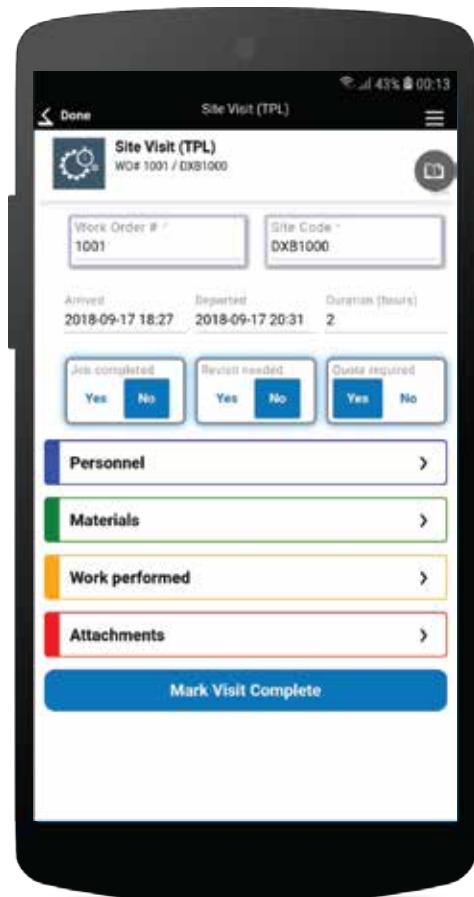
MOBILE AS PART OF A COMPLETE SYSTEM



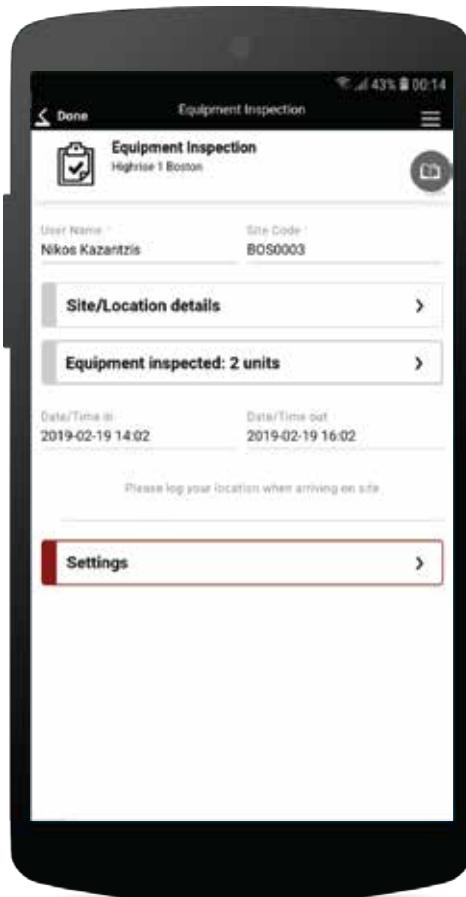
mazia|EVPTM
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha TransformTM

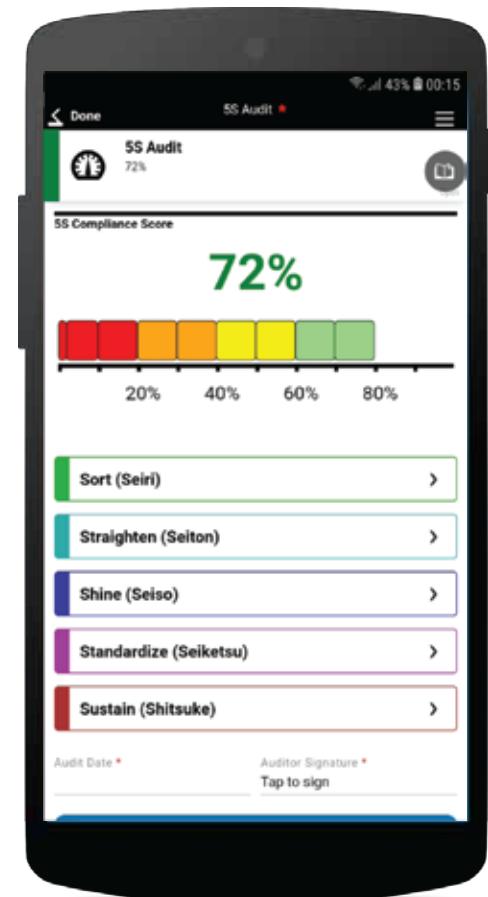
EXAMPLE



Site Visit



Equipment Inspection

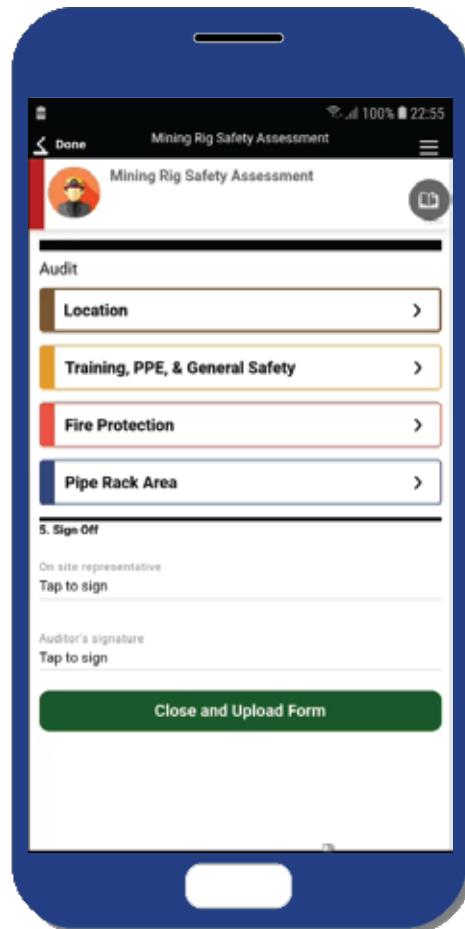


5S Audit

mazia|EVPTM
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha
TransformTM

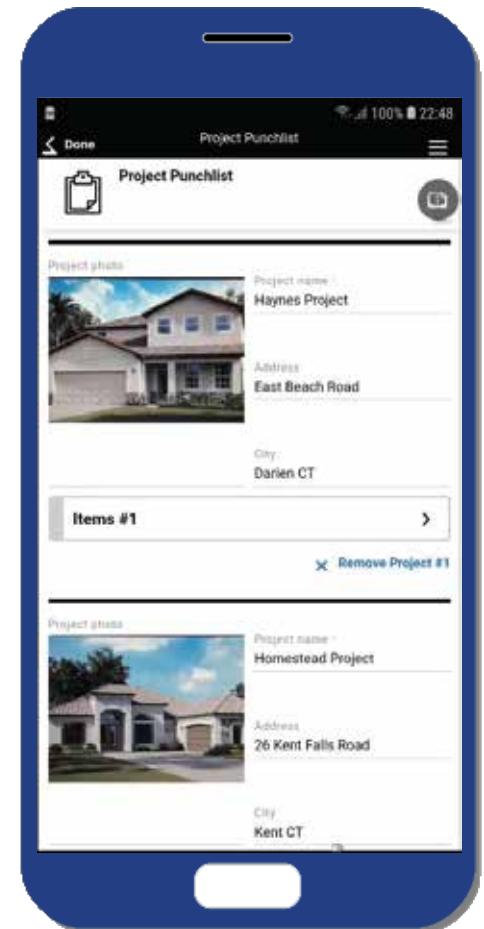
EXAMPLE



Mining Rig Safety Assessment



Perioperative Nursing Record

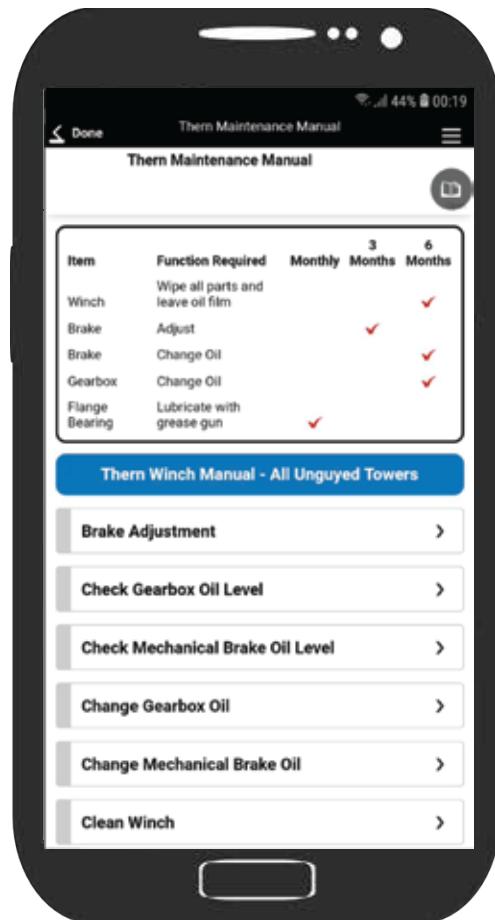


Project Punchlist

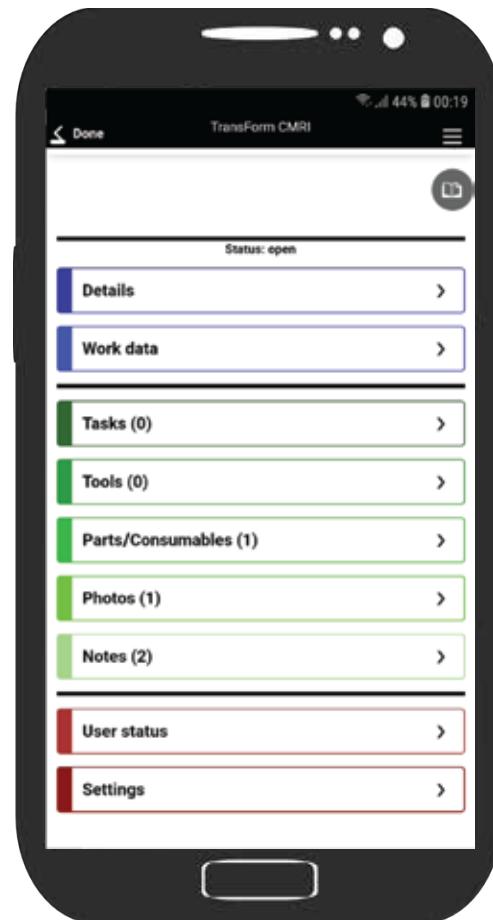
mazia|EVPTM
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha TransformTM

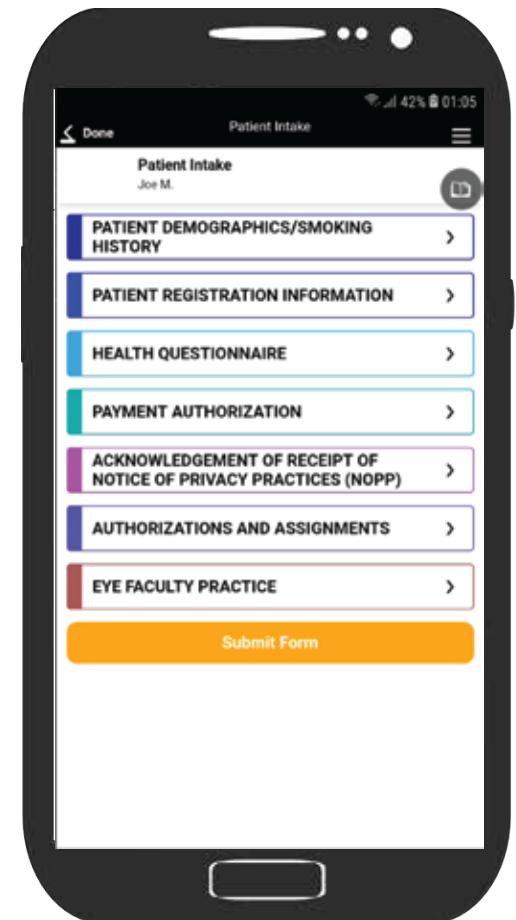
EXAMPLE



Maintenance Manual



Maintenance/Rollout/Inventory



Patient Intake

EXAMPLE

Daily Railyard Environmental Inspection

DAILY RAILYARD ENVIRONMENTAL INSPECTION
Return Completed Inspection to Environmental Dept.
Inform Shift Supervisor Immediately if Any Item of Concern is Noticed.

The purpose of this inspection is to determine if any items of environmental concern exist. Visually inspect the railyard for the items below. Be sure to observe each stormwater inlet.

Area Transload Road

Any signs of spills or debris?	Yes	No	Not applicable	Any other items of concern?	Comments/Notes*
Are signs of oil, paint, or acidic chemicals present?	Yes	No	Not applicable	Any other items of concern?	Comments/Notes*

X. Delete Inspection #1

Section 1, Tracks A-D (8 tracks)

Any signs of spills or debris?	Yes	No	Not applicable	Any other items of concern?	Comments/Notes*
Are signs of oil, paint, or acidic chemicals present?	Yes	No	Not applicable	Any other items of concern?	Comments/Notes*

X. Delete Inspection #2

+ Add a new inspection

Note: New findings were corrected. Dr = removed debris, placed into waste drum.

Note the location of any observed items of concern on Figure 1 (Site Diagram).

Site Diagram

Railyard inspection

OSHA Self-Inspection Checklist

Use this self-inspection checklist to identify probable hazards and whether they are under control. Use the 'N/A' button next to a section if it does not apply to the current area being inspected.

Employer Posting	Not Answered	Set N/A
Recordkeeping	All Answered	Set N/A
Safety and Health Program	All Answered	Set N/A
Medical Services & First Aid	All Answered	Set N/A
Fire Protection	All Answered	Set N/A
Personal Protective Equipment	All Answered	Set N/A
General Work Environment	Not Answered	Set N/A
Walkways	Not Answered	Set N/A
Floor and Wall Openings	Not Answered	Set N/A
Stairs and Stairways	Not Answered	Set N/A
Elevated Surfaces	Not Answered	Set N/A
Exiting or Egress - Evacuation	Not Answered	Set N/A

OSHA Self-Inspection Checklist

Job Safety Analysis

Date Site No.

Telecom Division Project Name Project Number

Subcontractor Name Subcontractor Field Supervisor

Site Location (GPS) Site Photo Tap to capture image

Photo Comments

JOB SITE INFORMATION AND HAZARDS

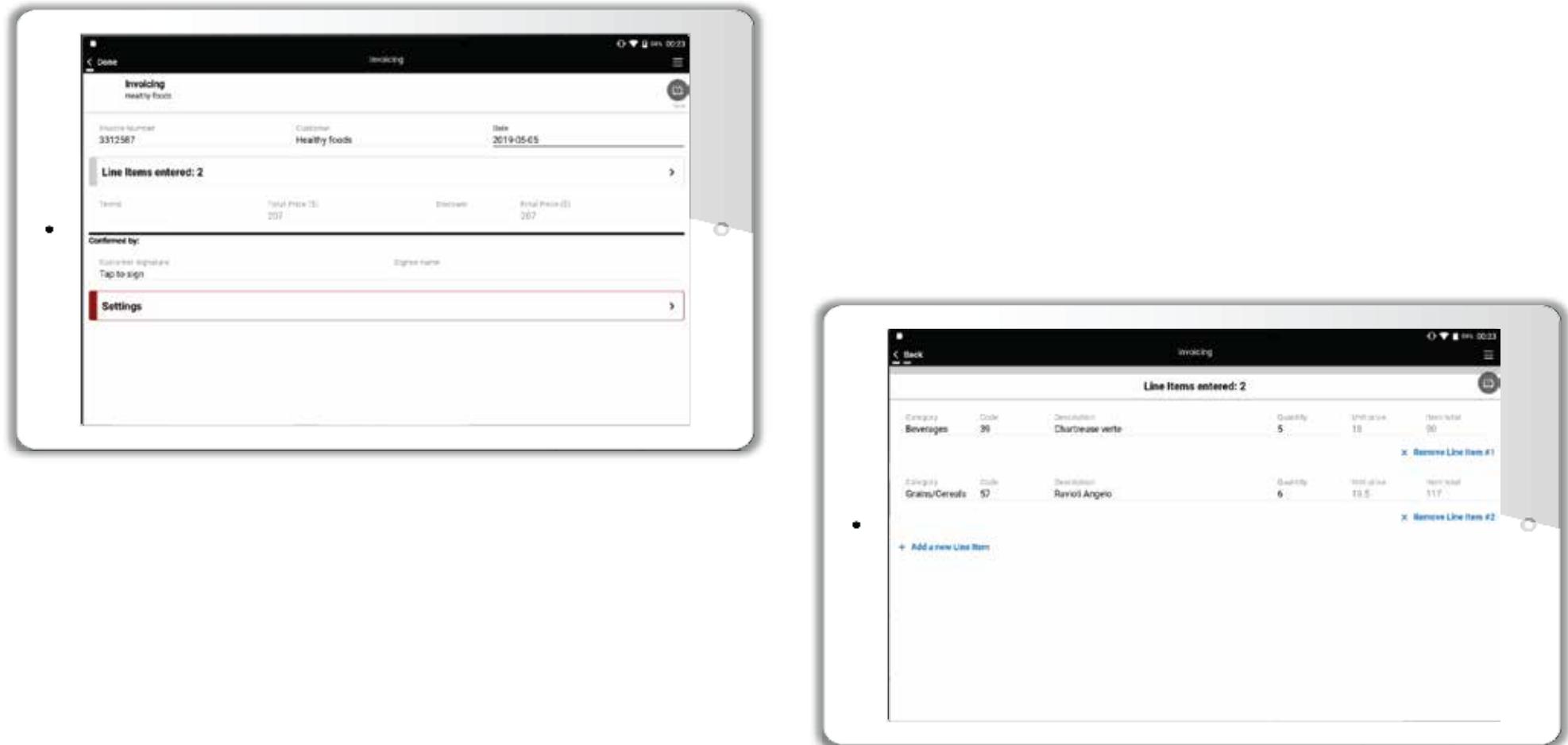
JOB HAZARD ANALYSIS

REQUIREMENTS/PREPPLANNING

PERSONNEL

Job Safety Analysis

EXAMPLE



Invoicing

mazia|EVPTM
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha
Transform™

WHY USE TRANSFORM?

Line-of-business

Digitize field data capture with no programming expertise

Utilize photo capture, barcode scanning, geolocation and other mobile features

“Same-day” deployment of complete apps to mobile

Only a single app to download from App-Stores –regardless of the number of apps deployed

Software developers / system integrators

Backoffice/Management has reduced visibility into what is happening in the field

Manage lifecycle for a larger number of projects –easily support 10s or 100s of different deployed apps

No App-Store submissions required to deploy and update apps

Built-in low-code programming language to extend no-code capabilities

Rich integration options for connecting with other IT systems

MaziaEVP Tranform 2 minute Demo

mazia|EVPTM
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha Transform™

FORM DESIGN QUICKSTART

The screenshot shows the alpha Transform Designer interface. At the top, there's a navigation bar with links for Home, Designer (which is selected), Permissions, and Management Console. On the right side of the header, there's a user profile for Nikos Kazantzis, Main Account, with a note that the login expires in 2 days (11-Jul-19). There are also Logout and Help buttons.

The main area is divided into three sections:

- Quick Reference:** A sidebar containing a list of quick reference items, each with a small icon and a label. Some items have '+' and '-' buttons next to them. The list includes:
 - As text
 - 1.23 :number
 - 12 :signed
 - 123 :integer
 - 0.00 :currency
 - 📞 :phone
 - 📅 :date
 - 🕒 :datetime
 - ⌚ :timenow
 - ⏱ :stopwatch
 - 📸 :photo
 - 🖨 :scanner
 - 🔊 :audio
 - ✍ :signature
 - 📍 :location
 - ☰ : item 1, item 2, item 3
 - 📋 : Yes, No
 - A : Group Heading
 - A. : Group Heading:datagroup
 - A : ## Section Heading
 - A. : ## Section Heading:datagroup
 - As : ## Instructions
 - ➡ : Submit Form
 - ➡ : Submit Form:status|new status
- Quickstart Text:** A large central text editor area with a toolbar at the top featuring '#Prefix', 'Title text', and 'Options' buttons. Below the toolbar is a text input field with placeholder text: 'Type Quickstart text here...'. At the bottom of this section are 'Save' and 'Cancel' buttons.
- Simple Preview:** A large rectangular area on the right side of the interface, currently empty, which is used for previewing the form design.

mazia|EVP[®]
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha
Transform™

BUILD APPS BY CONFIGURATION (CODING OPTIONAL)

A WYSIWYG builder allows creation of complex applications, that can include:

- a rich set of data capture field types
- datagroups in single or hierarchical levels
- conditional branches (IF/ELSE)
- layout/UI
- pages/sections/navigation

The screenshot shows the alpha Transform WYSIWYG builder on the left and a generated mobile application preview on the right.

Builder View (Left): A sidebar menu lists various configuration elements such as 'sectionStart', 'field', 'pageStart', 'dataGroupStart', 'sectionEnd', and 'conditional logic'. A large list of data field types is displayed, including 'Quickstart-style Input', 'Data Field: Typed Text', 'Data Field: Number', 'Data Field: Phone Number', 'Data Field: Signed Number', 'Data Field: Integer', 'Data Field: Currency', 'Data Field: Date', 'Data Field: DateTime', 'Data Field: Time Now', 'Data Field: Time Now Countdown', 'Data Field: Stopwatch', 'Data Field: Photo', 'Data Field: Audio', 'Data Field: Signature', 'Data Field: Scanner', 'Data Field: Location', 'Data Field: List', 'Data Field: Button List', 'Heading Group', 'Heading Section', 'Heading Instructions', 'Section Start', 'Section End', 'Change Status', 'Page Start', 'Page End', 'Go To Next Page', 'Go To Target', 'Define Target', 'Countdown', 'Action Button', 'IF', 'ELSE IF', 'ELSE', 'ENDIF', 'Data Group Start', 'Data Group End', 'HTML', and 'Set Error Indication'.

Preview View (Right): A mobile application titled 'Equipment Inspection' is shown. It includes fields for 'User Name' (John Doe) and 'Site Code' (BOS1000). Below these are sections for 'Site/Location details' (Site/Building name: Indigo Tower, Address: 28, Athens Avenue, City: Boston, State/Region: MA, Zip Code: 02379, Country: USA, Latitude: 42.010101, Longitude: -71.010101), 'Site notes' (Contact Mike Harris at reception (439-911488)), and buttons for 'Navigate to Site' and 'Weather forecast'. At the bottom, there are sections for 'Equipment inspected: 0 units', 'Details #1', and 'Equipment data #1'.

LAY OUT THE UI TO MAXIMIZE EFFICIENCY

Responsive controls that allow apps to:

- operate in portrait and landscape
- fit on small devices
- take advantage of available screen area of larger devices

The screenshot illustrates the alpha TransForm platform's responsive layout features. On the left, the 'Designer' tab is active, showing the XML code for an 'Equipment Inspection' form. A red box highlights a sectionStart element, with a yellow callout 'Define sections enclosing elements to layout'. On the right, the 'Preview' tab shows the resulting mobile application interface. A red box highlights a responsive section, with a yellow callout 'Define responsive layout per mobile device size/orientation'. The preview shows a user profile, site location details (Indigo Tower, 28 Athens Avenue), and inspection results (Equipment Type: Generator). Buttons for 'Navigate to Site' and 'Weather forecast' are also visible.

```
<?xml version="1.0" encoding="UTF-8"?>
<form>
    <!-- User information -->
    <sectionStart>
        <field>next_user_name / "User Name" (John Doe)</field>
        <field>iterSiteCode / "Site Code" (BOS1000)</field>
    </sectionEnd>
    <pageStart>"Site/Location details"</pageStart>
    <field>nextSite_building_name / "Site/Building name" (Indigo Tower)</field>
    <field>nextAddress / "Address" (28, Athens Avenue)</field>
    <sectionStart>
        <field>text city / "City" (Boston)</field>
        <field>text state_region / "State/Region" (MA)</field>
        <field>text zip_code / "Zip Code" (02379)</field>
    </sectionEnd>
    <sectionEnd>
        <field>nextSite_notes / "Site notes" (Contact Mike Harris at reception (439-911488))</field>
        <button>bnNavigate = "Navigate to Site"</button>
        <button>bnWeather = "Weather forecast"</button>
    </sectionEnd>
    <pageEnd>
    <pageStart>"Equipment Inspected: [&{EquipmentGroup}] units"</pageStart>
    <dataGroupStart>equipmentGroup "Equipment Group"</dataGroupStart>
        <sectionStart>
            <field>number equipmentNumber / "Equipment # (auto)" (1)</field>
            <field>iterEquipmentType / "Equipment Type" (Generator)</field>
        </sectionEnd>
        <sectionEnd>
    <pageStart>"Details"</pageStart>
    <pageStart>"Equipment data"</pageStart>
</form>
```

INDIVIDUAL APPLICATION THEMING

Personalize apps with:

- built-in icon library
- custom SVG icons
- text/background/indicator colors



ADVANCED FEATURES - CUSTOM EMBEDDED BUSINESS LOGIC

Add powerful client-side (offline) calculations, actions and behaviors using the “low code” TransForm Programming Language (TPL):



Data input validation

- Verify value or range
- Check input limits
- Compare with other values – on device or external



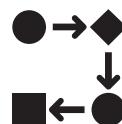
Web service REST API integration

- Call any web service to retrieve list items
- Call any web service to validate data
- Push data to web service



Calculations

- Client-side calculated values (sums, averages, products) from entered data
- Rich set of built-in functions that include date arithmetic
- Can be linked to events for automatic re-calculations

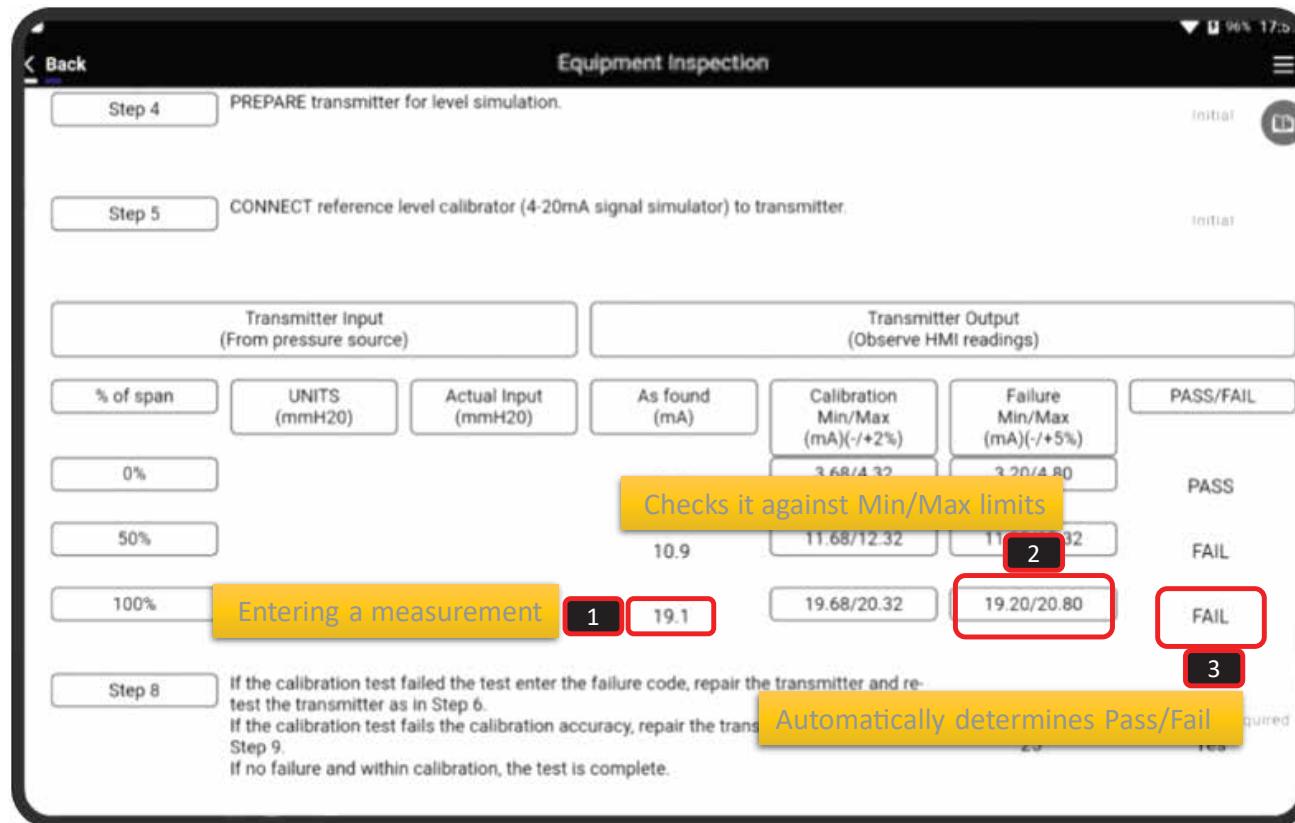


Work flow

- Manage form “state”
- Trigger actions based on “state”
- Process “state” based on entered data

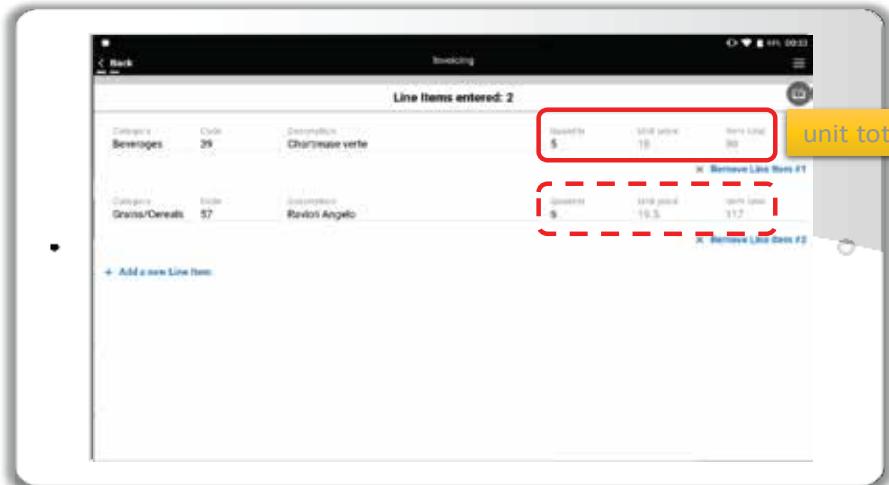
TPL EXAMPLES

Validation of entered data within a permitted range of values to produce Pass/Fail results automatically

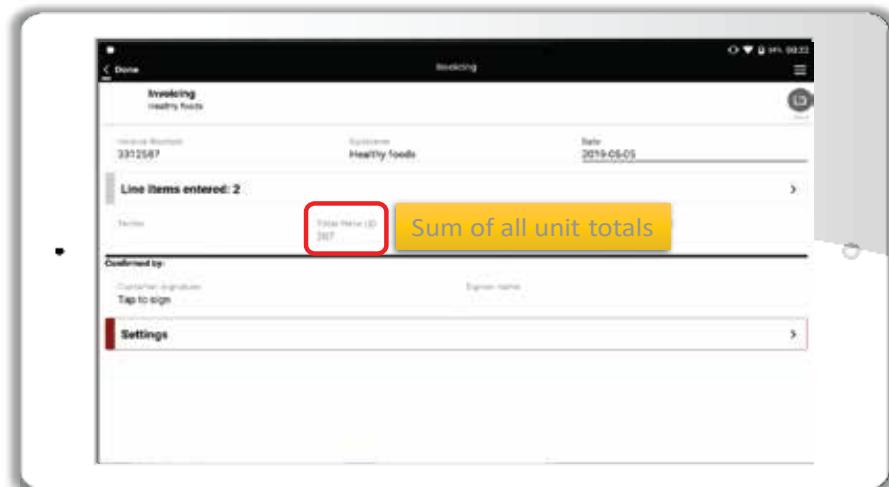


TPL EXAMPLES

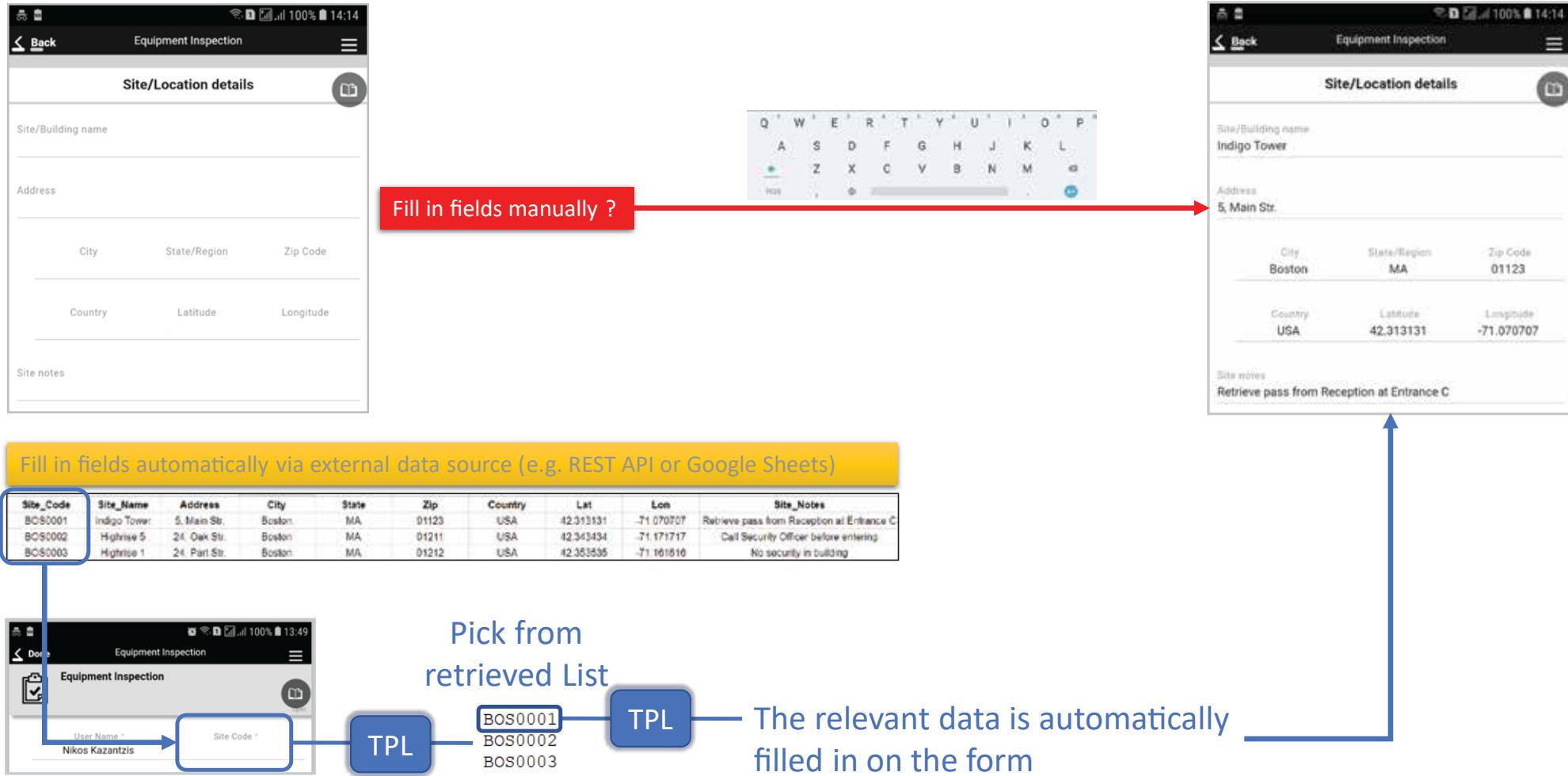
Calculation of product and sum values across arbitrary quantity of elements



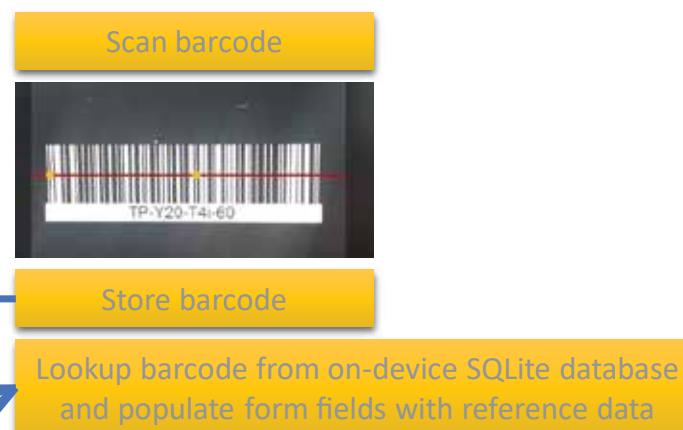
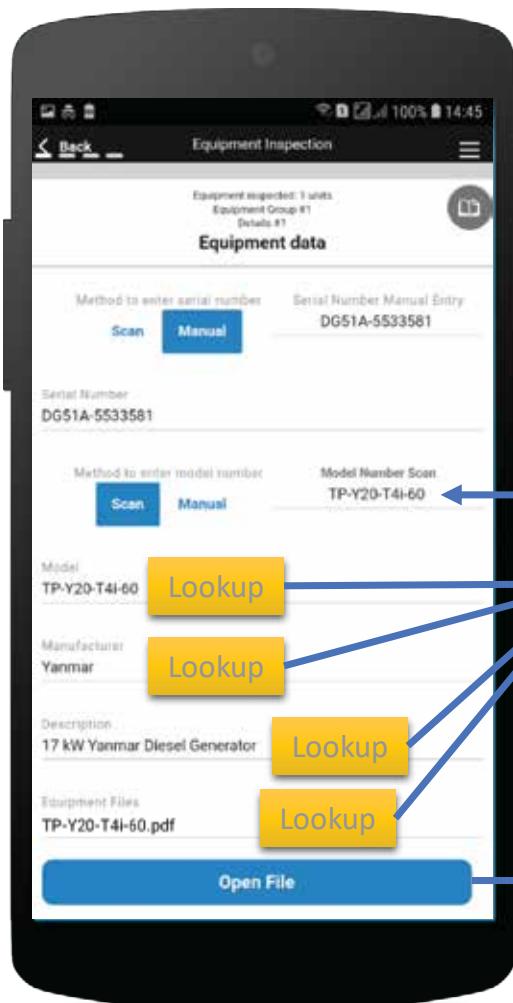
(only defined once, automatically applied to all added rows)



TPL EXAMPLES: DYNAMIC LISTS AND REMOTE LOOKUPS



TPL EXAMPLES: ON DEVICE LOOKUPS AND FILES



Access relevant on-device files (PDFs, videos, images etc.) via suitable native mobile device apps

Complete process can operate offline

mazia|EVP[®]
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha Transform[™]

TPL SCRIPTING - CODE EDITOR

A built-in TPL code editor:

- event-based at field and app level
- rich function library with documentation
- custom user functions with arguments

The screenshot shows the alpha Transform application interface with the 'Designer' tab selected. On the left, there's a form with fields like 'Invoice Number', 'Customer', 'Date', and 'Line Items entered: {len(#lineItemsArray)}'. Below this is a table for 'Item Group 1' with columns for 'Category', 'Code', 'Description', 'Quantity', 'Unit price', 'Item total', 'Terms', 'Total Price (\$)', and 'Discount'. Each column has an 'Editor' button and a 'Changed' status indicator. On the right, the main area is a code editor titled 'Tester' containing TPL script. The script handles 'changed_lineItemsArray.quantity' events, calculates totals, applies discounts, and rounds values. To the right of the code editor is a sidebar with tabs for 'Reference', 'Code', and 'Fields'. The 'Code' tab is active, showing a 'Function List' and a 'Function Families' section with links to various TPL functions like abs, acos, ajaxGET, args, array, etc. At the bottom of the sidebar is a table titled 'All Built-in Functions' listing over 100 functions categorized by type (math, ajax, execution, array) and their descriptions.

```
ON *changed_lineItemsArray.quantity
ON *changed_lineItemsArray.quantity
    rounded_itemtotal = round(#quantity * #unitPrice,2)
    ##itemtotal = rounded_itemtotal
    tempTotal = 0
    FOR i = 0 to len(#lineItemsArray)-1
        tempTotal = tempTotal + #lineItemsArray[i].quantity *
#lineItemsArray[i].unitPrice
    ENDFOR
    tempFinal = tempTotal
    IF #discount > 0
        tempFinal = tempFinal * (1-discount/100)
    ENDEF
    rounded_totalPrice = round(tempTotal,2)
    rounded_finalPrice = round(tempFinal,2)
    $totalPrice = rounded_totalPrice
    $finalPrice = rounded_finalPrice
ENDON
```

All Built-in Functions	106
abs(number)	math +
acos(number)	math +
ajaxGET(url, timeout)	ajax +
ajaxSendRequest(type, url, timeout, body, headers)	ajax +
args(argnum)	execution +
argslen()	execution +
array()	array +
array(val1, val2, ...)	array +
arrayConcat(array1, value1, value2, ...)	array +
arrayCopy(array)	array +
arrayCopyMap(array, namelin, namelout, name2in,...)	array +
arrayFilter(array, tests)	array +
arrayFilterIndex(array, tests)	array +
arrayFirst(array, tests)	array +
arrayFirstIndex(array, tests)	array +
arrayIndexOf(array, text, start)	array +
arrayInsert(array, start, val1, val2,...)	array +
arrayJoin(array, separator)	array +
arrayObjValue(array, name)	array +
arrayPop(array)	array +
arrayPush(array, val1, val2, ...)	array +
arrayRemove(array, start, count)	array +

TPL - A POWERFUL TOOL, NOT A BARRIER

TPL is an optional capability

Only a small percentage of Excel users write VBA code.

Yet millions of people continue to use Excel very effectively without knowing about VBA.

Similarly, sophisticated forms/apps can be built in TransForm without TPL.

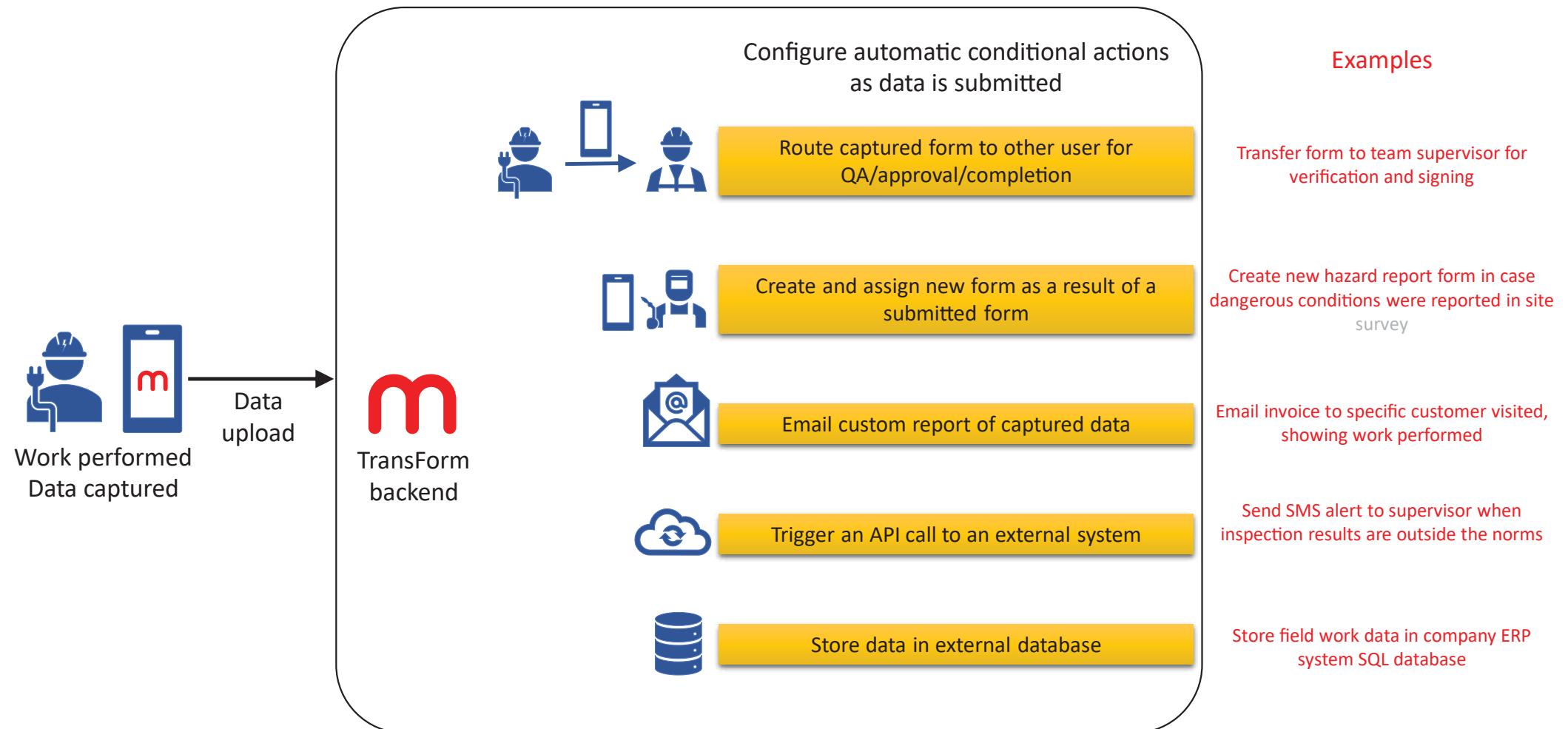
TPL is the enabler for even more advanced features and capabilities.

TPL is low code

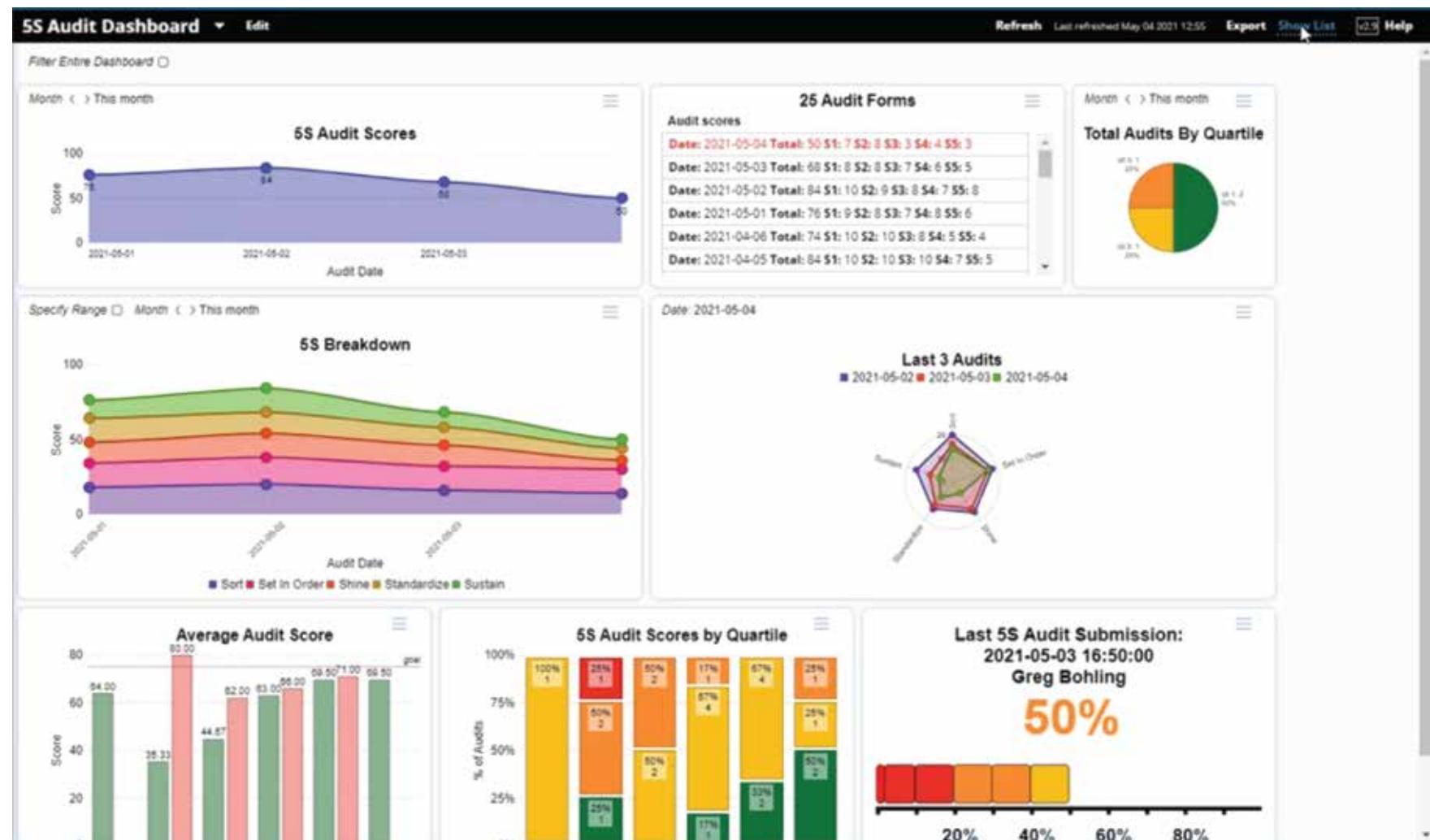
TPL is not a general-purpose programming language. It is a “low code” implementation that was developed specifically for a data capture app like TransForm.

Learning to use TPL is much simpler compared to a general-purpose computer language – existing developers can work proficiently and produce powerful results in a matter of hours/days.

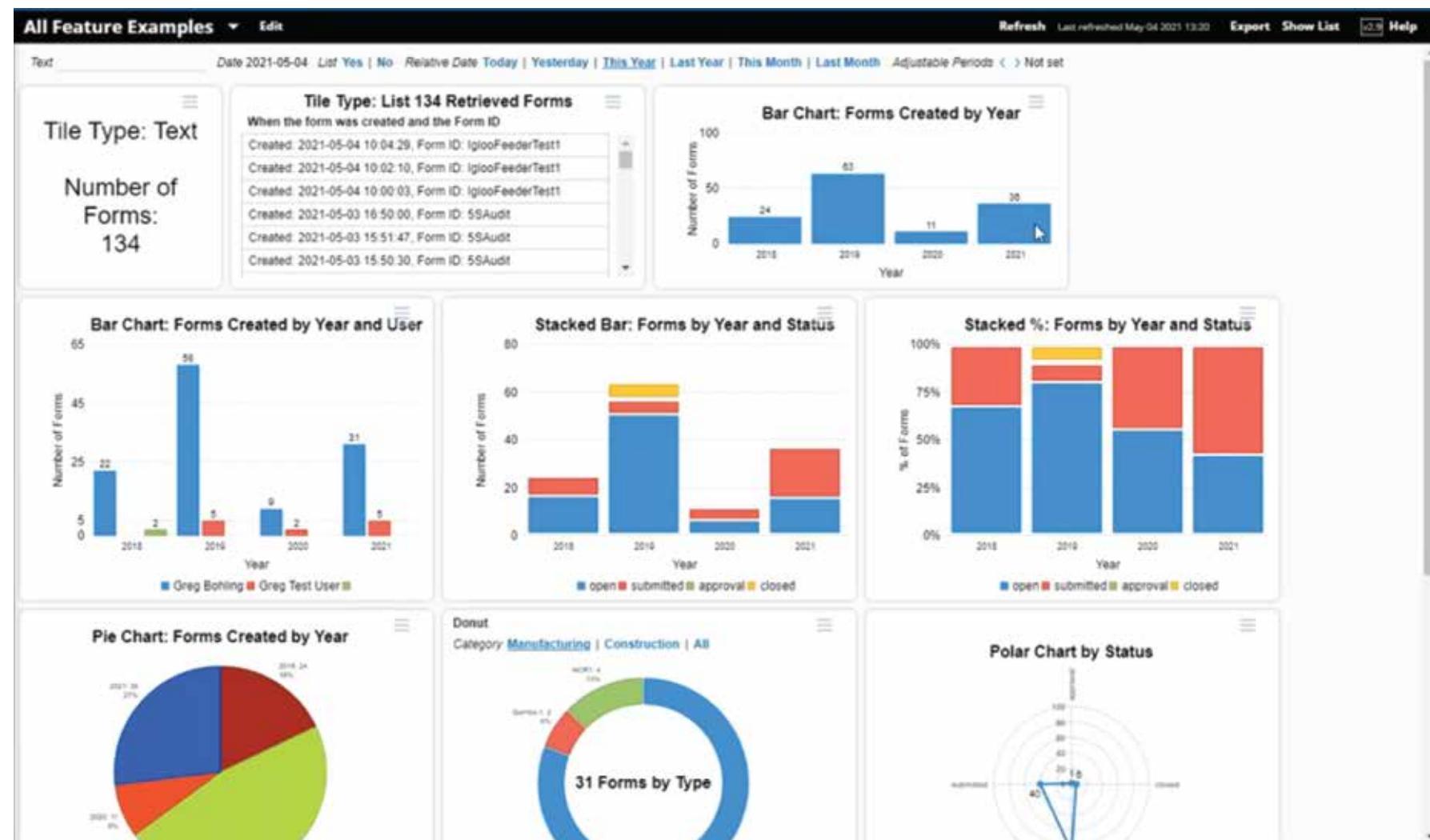
TRANSFORM: ENABLING ADVANCED DIGITAL WORKFLOWS



DASHBOARDS



DASHBOARDS



DASHBOARDS

SS Audit Dashboard ▾ Edit

Filter: Customize Dashboard Tile #2

Import/Export Save Cancel

Properties Filtering Sort Order User Inputs Tile Type: Chart List Text

Tile Size: Wide, Medium, Narrow, Small

Template Text (TPL template):

```
<span style="font-family:Open Sans;">@audit...</span>
```

Chart Title: "Audit Forms"

List Title: "Audit scores"

Titles: Chart: "(@dash.rows) Audit Forms", No legend

TPL Functions (advanced)

```
"function (@auditSummary @data=arg0) {if (@data.length > 0) {return @data[0].score;}}"
```

+ () ✎ X ↑ ↓ Auto-Preview

TPL Functions:

```
itn=<itn> & "<b>Total:</b> " & data.totalScore  
itn=<itn> & "<b>S1:</b> " & data.sortScore & "<b>S2:</b> " &  
data.setInOrderScore  
itn=<itn> & "<b>S3:</b> " & data.shineScore & "<b>S4:</b> " &  
data.standardizeScore  
itn=<itn> & "<b>S5:</b> " & data.sustainScore  
itn=<itn> & "<span>"  
return itn
```

The screenshot shows a dashboard titled "SS Audit Dashboard" with a configuration dialog open for "Customize Dashboard Tile #2". The dialog includes tabs for Properties, Filtering, Sort Order, User Inputs, Tile Type (set to Chart), and a preview section with Import/Export, Save, and Cancel buttons. The preview shows three tiles: a list of audit forms, a pie chart titled "Total Audits By Quartile" with segments for 25%, 25%, 25%, and 50%, and a card with audit submission details (1-05-03 16:50:00, Greg Bohling, 50%) and a progress bar at 50%.

mazia|EVP®
MOBILE & WEB | INTERGRATIONS | REPORTING

powered by
alpha
Transform™

DASHBOARDS

< Dashboard

Last refreshed 1 minute ago [Refresh](#)

[Check All](#) [Clear All](#)

	Status	Created	User Name	Details	# Photos	Place Born
<input type="checkbox"/>	Open	October 23, 2017	Dan Bricklin		 7 / 100	PA
<input type="checkbox"/>	★ Open	December 4, 2017	Dan Bricklin		 1 / 100	
<input type="checkbox"/>	★ Open	June 25, 2018	Dan Bricklin		 1 / 100	PA
<input type="checkbox"/>	★ Open	August 22, 2018	Dan Bricklin		 3 / 100	PA
<input type="checkbox"/>	★ Open	August 22, 2018	Dan Bricklin		 1 / 100	
<input type="checkbox"/>	★ Open	August 22, 2018	Dan Bricklin		 1 / 100	

Close > **Form Contents**

[Form View](#) [Editable Fields >](#)

General Sample

Oct 23, 2017 3:28pm

The purpose of this form is to serve as a general test for usability and training.

Please fill out the entire form.

There is no need to provide truthful answers. This is not a test for collecting real data but rather a test of using this app.

Overall Information

Is this the first time you are filling out this form? *

Yes

In which state or province were you born? *

Pennsylvania

Your birth date *

1951-07-16

Do you wish to include refrigerator inventory?

Actions

[Export Selected Form](#)

[View Comments](#)

[Delete Selected Form](#)

User Name or Queue
Dan Bricklin

User ID Queue
danbricklin@bricklin.com

[Revert](#) [Save changes](#)

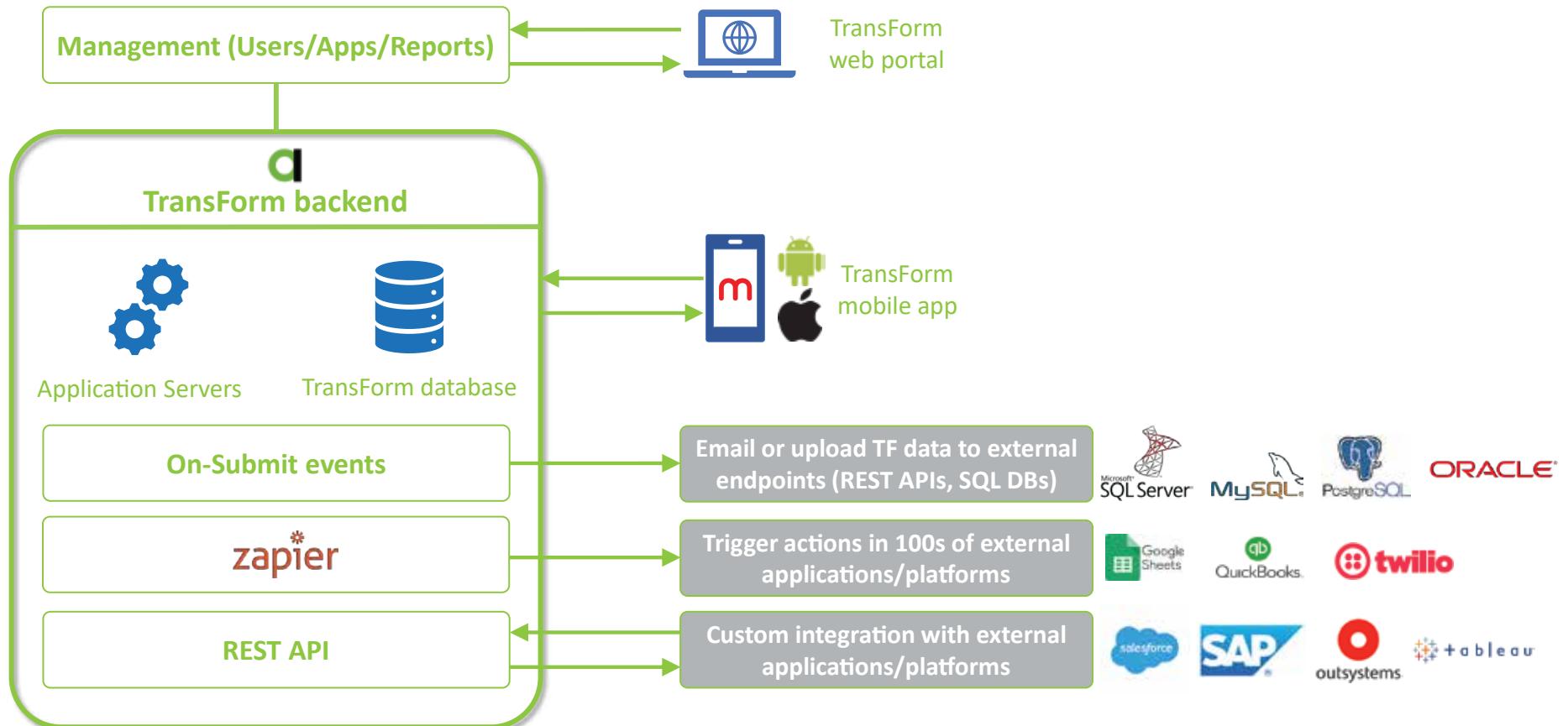
Prevent Filler from downloading
 Yes No

Change Status
Currently: Open

Submitted

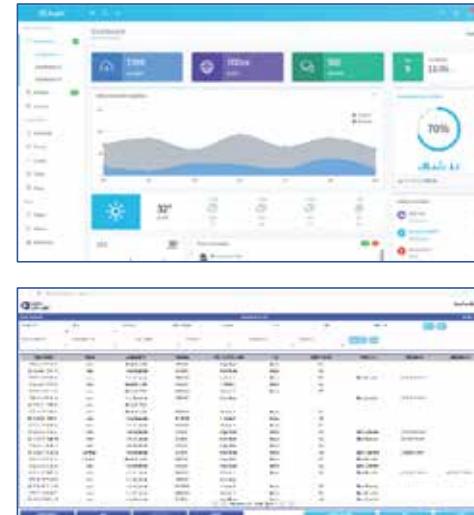
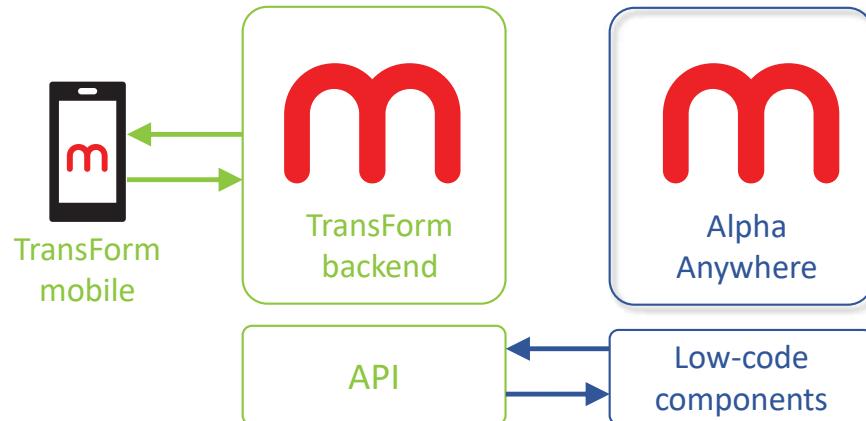
Closed

MOBILE AS PART OF A COMPLETE SYSTEM



TRANSFORM + ALPHA ANYWHERE - INTEGRATED SOLUTION PLATFORM

Use Alpha Anywhere to build customized backoffice applications to display/analyze/report-on the captured data

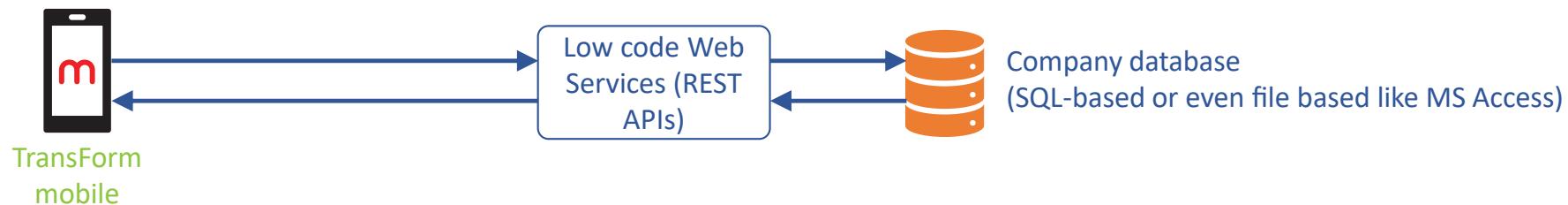


Dashboards/Dispatching



Custom PDF Reports

Use Alpha Anywhere to build Web Services on top of company databases to access them directly from TransForm mobile

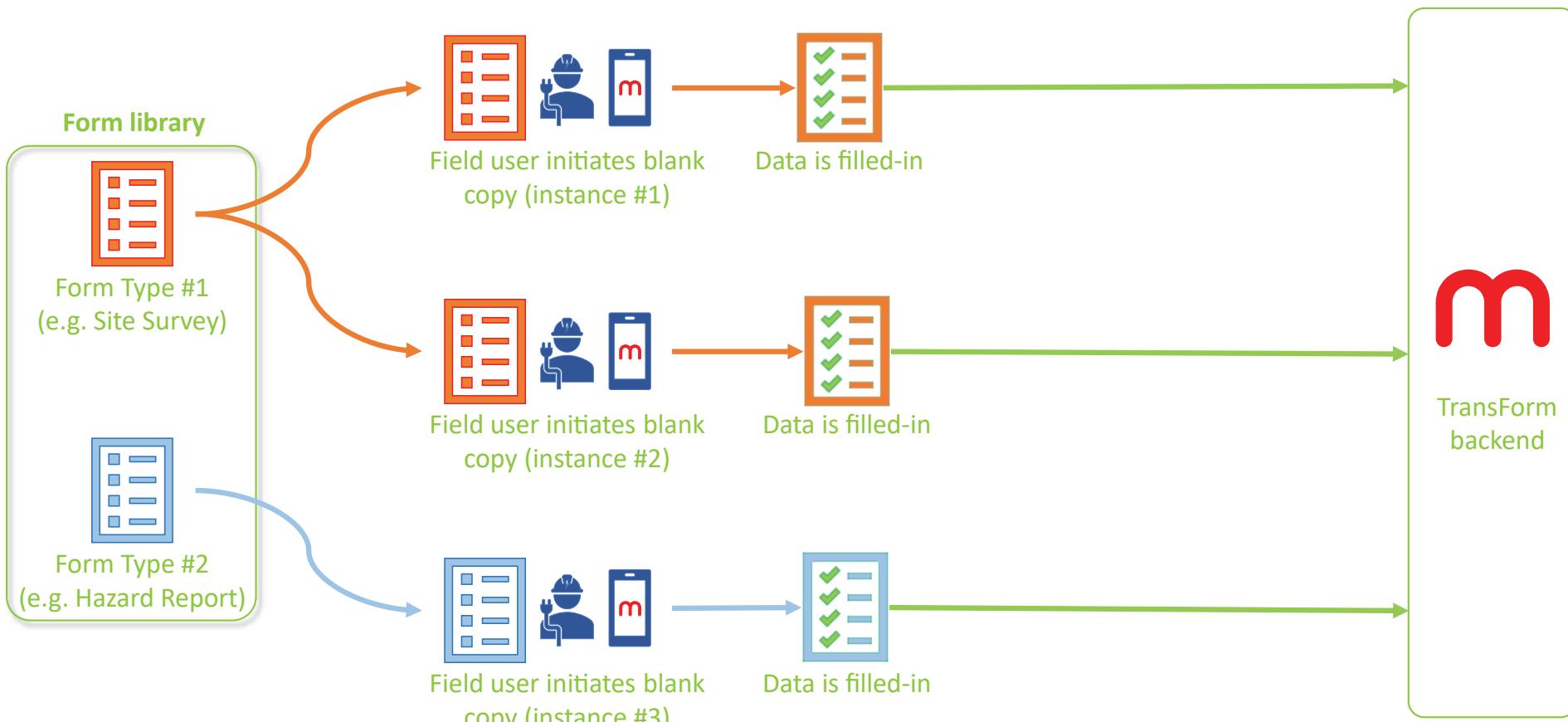


mazia|EVP™
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha Transform™

TRANSFORM WORKFLOW: FIELD-INITIATED DATA CAPTURE (AD-HOC)

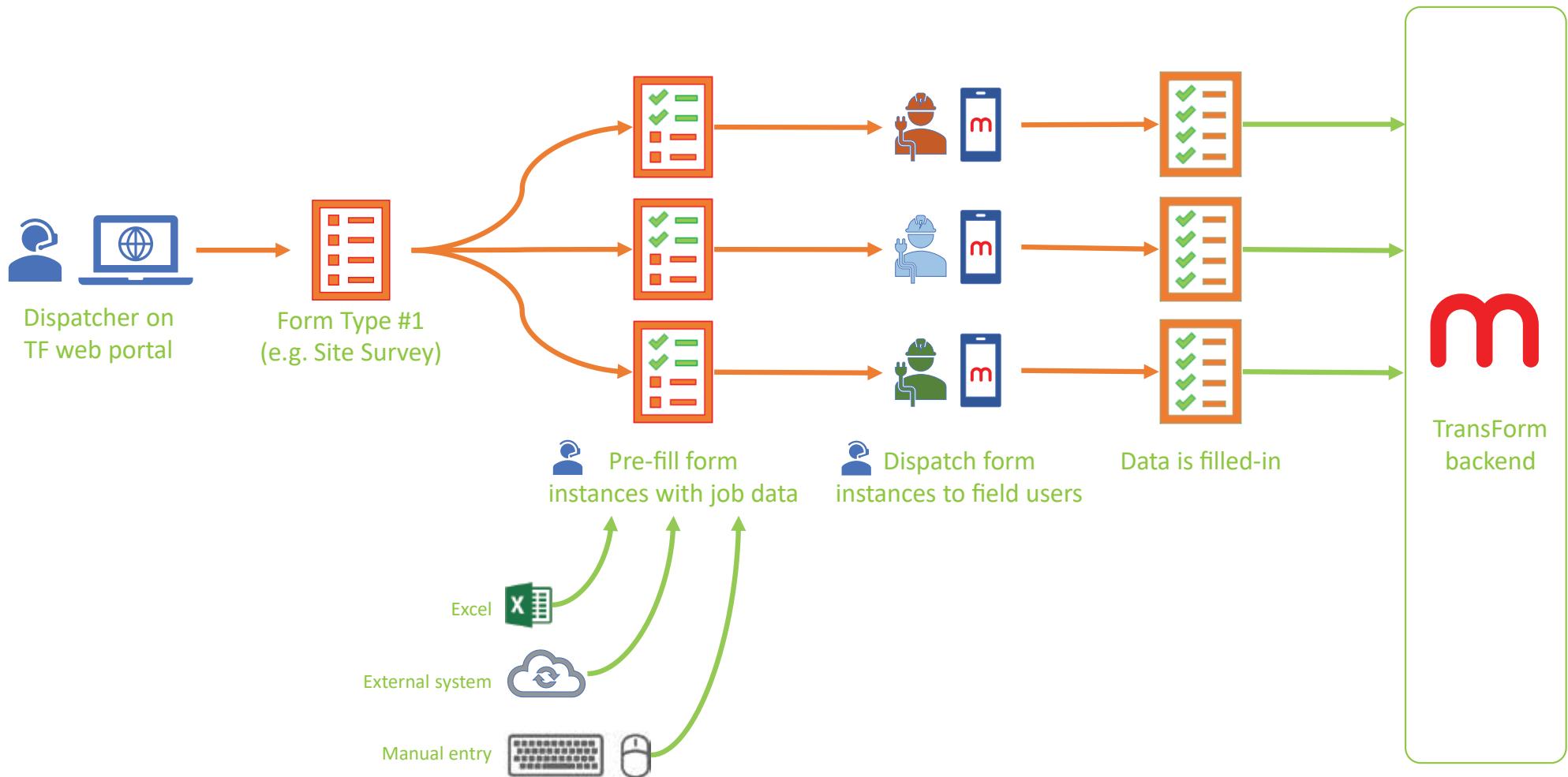
TransForm workflow: field-initiated data capture (ad-hoc)



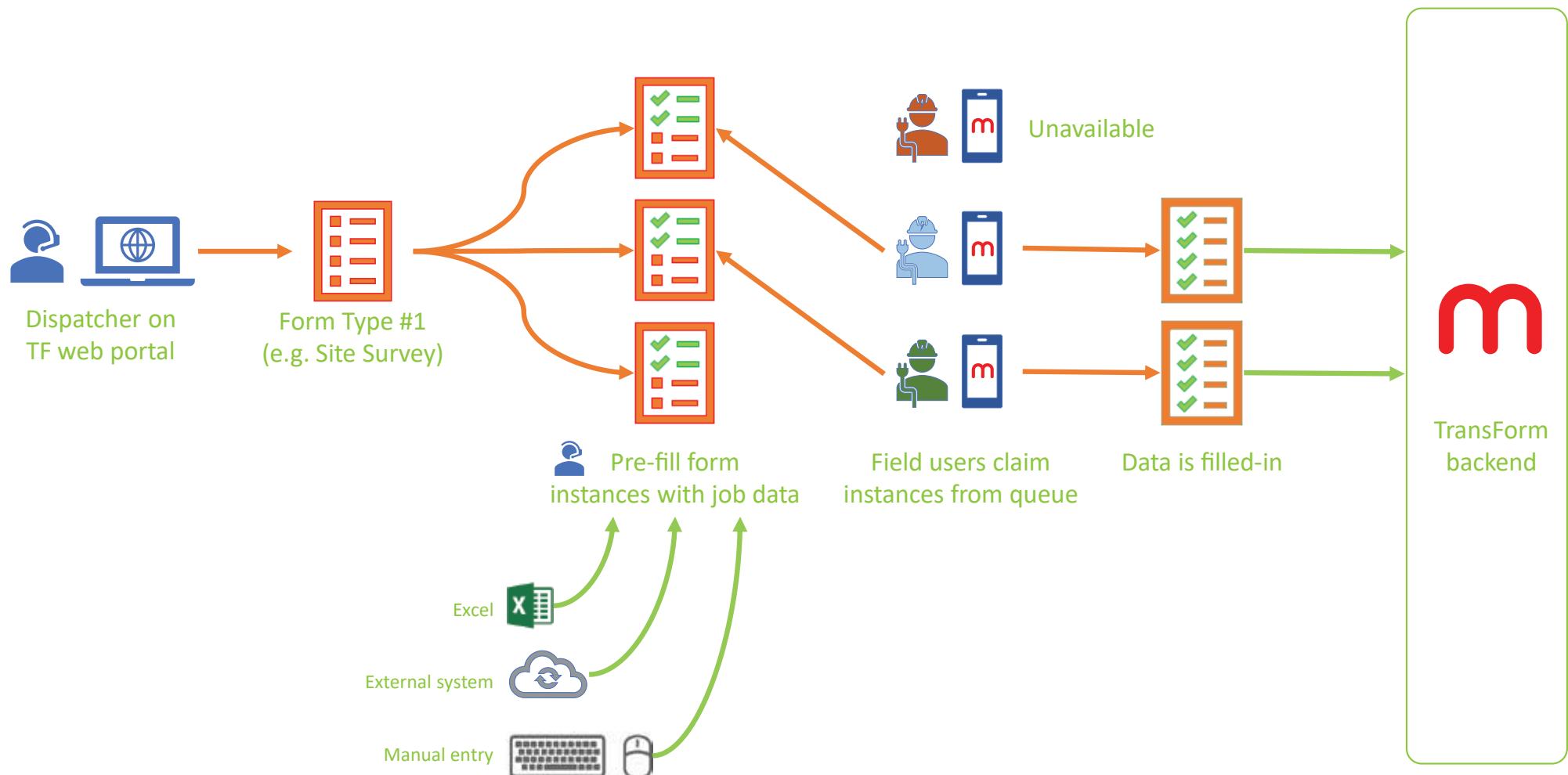
mazia|EVPTM
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha Transform™

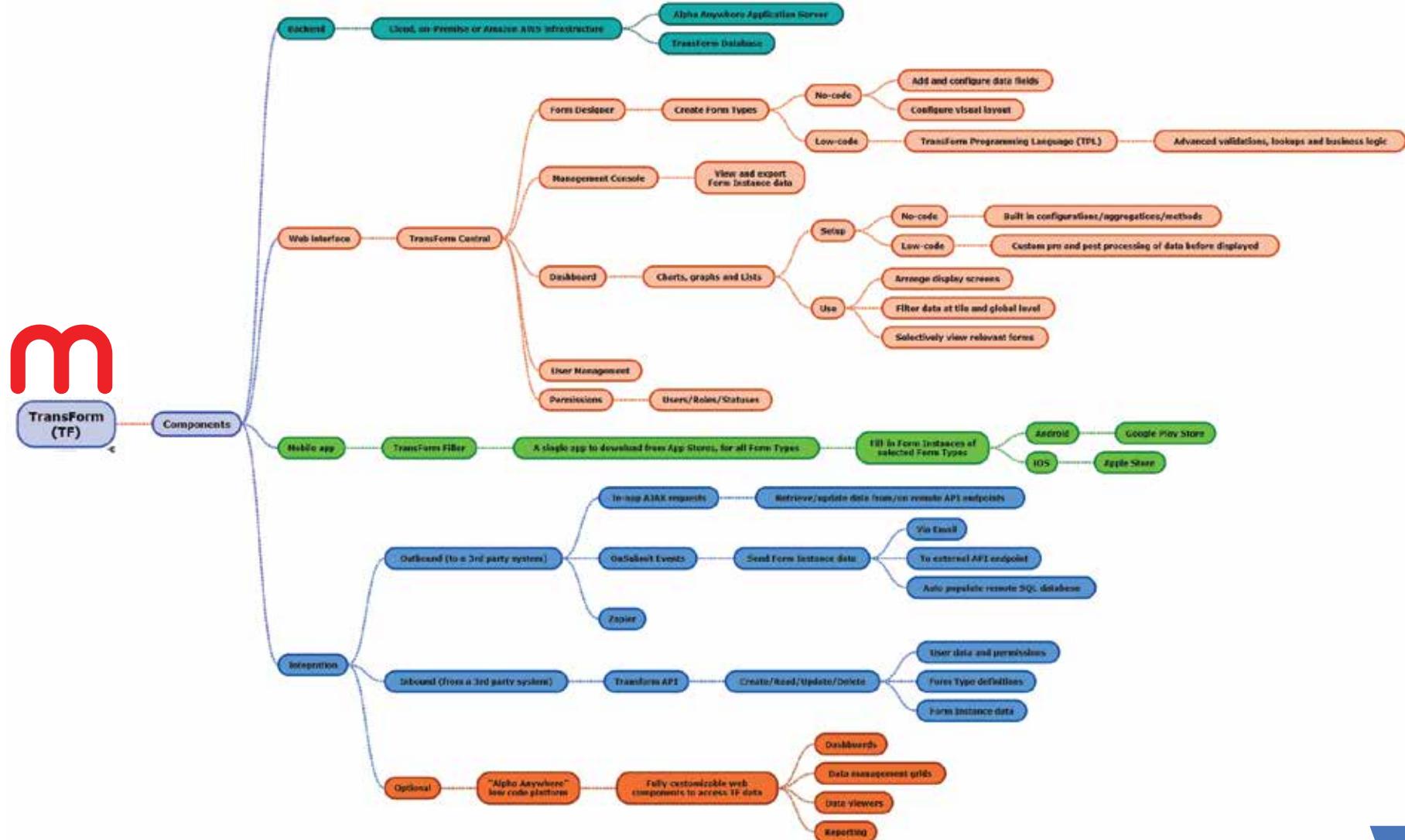
TRANSFORM WORKFLOW: DISPATCH



TRANSFORM WORKFLOW: QUEUING



TRANSFORM PLATFORM



mazia|EVPTM
MOBILE & WEB | INTEGRATIONS | REPORTING

powered by
alpha TransformTM

TPL DEBUGGING

The screenshot shows the mazia|EVP form designer interface. At the top, there are tabs for 'Functions' and 'ON Handlers'. Below these, a code editor displays TPL code. The code includes a function definition for '@convertToCelsius' and an event handler for 'ON *changed_tempF'. The event handler calls the function and stores the result in '#convertedTemp'. The code editor has line numbers and syntax highlighting. To the right of the code editor is a 'Execution' panel. This panel has tabs for 'Local', '\$', '#\$', '\$\$', 'Preview', and 'Preview JSON'. The 'Local' tab is selected, showing variables: 'tempF' with value '15' and 'convertedTemp' with value '15'. There is also a checkbox labeled 'Copy to Execution Form Data'. Below the execution panel is another code editor window, which is identical to the one above it, showing the same TPL code.

Built-in TPL debugger inside the form designer

The screenshot shows a mobile phone displaying the mazia|EVP mobile TPL debugger. The screen is divided into sections for code, local variables, and preview. The code section shows the same TPL code as the desktop version. The local variables section shows 'convertedTemp' with value '100' and 'tempF' with value '212'. The preview section is currently empty. Navigation icons for back, forward, and search are visible at the bottom of the mobile screen.

Mobile TPL debugger

 +27 (0) 11 326 6273
 +27 (0) 21 825 9725
 +27 (0) 84 836 4284
 talk2us@maziaeyp.com

 www.maziaeyp.com
 maziaeyp
 maziaeyp
 maziaeyp

 Bentley Office Park,
Block 9 First Floor,
Corner Rivonia & Wessel Road,
Johannesburg, 2128