

# Driving Innovation with Fusion Teams

Extending the Reach of SAP Using Low-Code Tools

# Hello!

Welcome to ASUG!

# What We'll Cover

- Assessing the Current State of SAP Development
- A Fresh Approach to SAP Development
- Breaking Down Barriers to Productivity
- Unlocking Productivity with Fusion Teams
- Wrap-Up

# **Assessing the Current State**

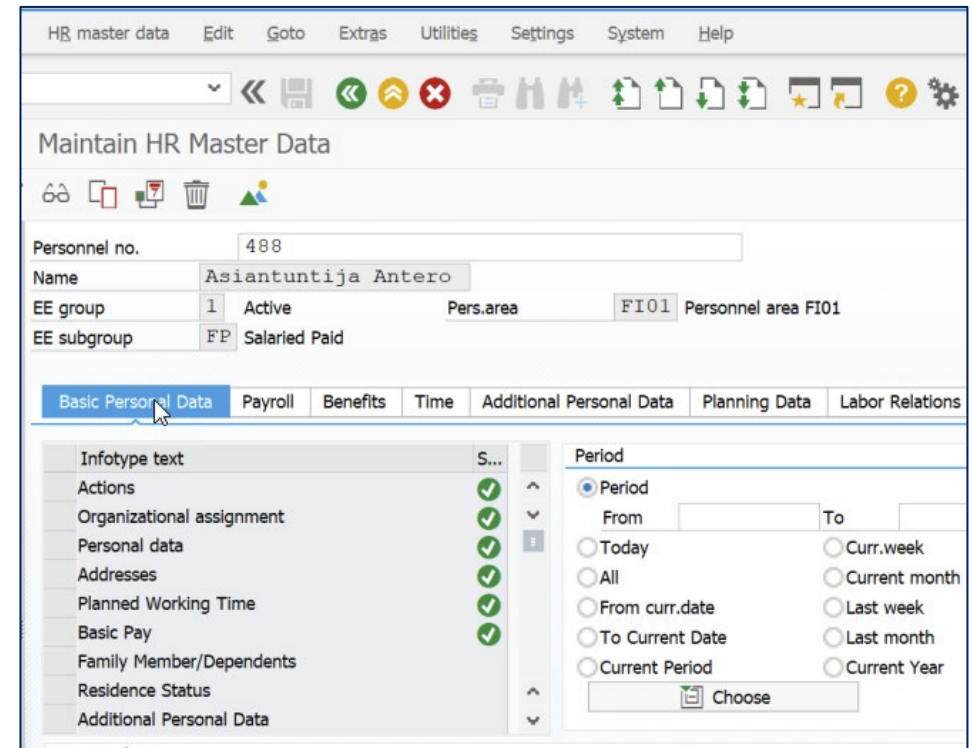
**Understanding the Current State of SAP Development**

# Common Challenges

- The needs from the business outweigh the capacity of IT
- The business has a difficult time articulating what it wants/needs
- Custom SAP solutions (e.g., Fiori apps) are expensive to build and maintain
- There are limitations to what can be built using ABAP & NetWeaver technologies
- SAP change management is slow and complex
- Apps / solutions only address *part* of a process problem

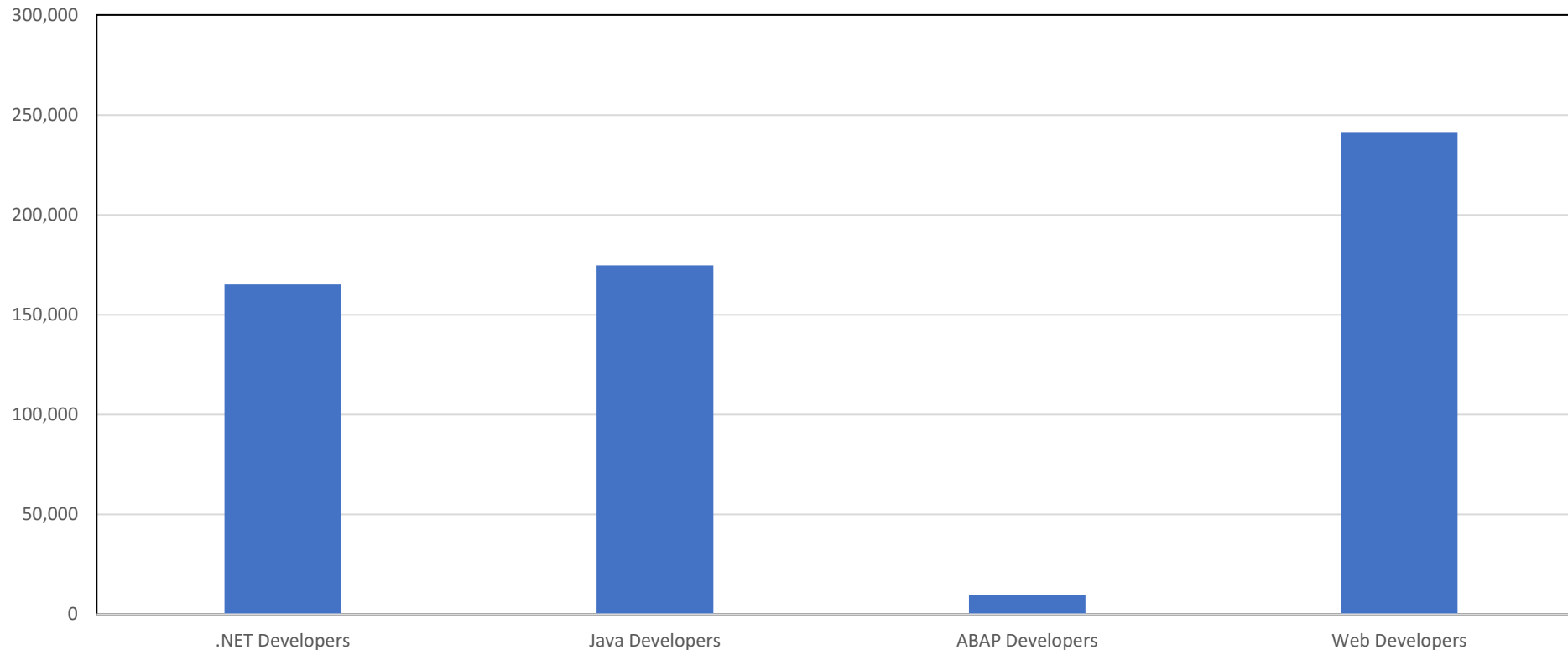
# Limited User Experience

- Many users still primarily transact in the legacy SAP GUI
- Processes are fragmented
- Workflow / automation is sparse
- Lack of transparency / analytics
- Minimal support for mobility
- Limited notifications



# Resource Constraints

Developer Availability in the US



Reference: <https://www.zippia.com/sap-abap-developer-jobs/demographics/>

# Developer Productivity Issues

- Programming languages provide abstractions that make it easier to solve complex problems
- There's a direct correlation between the quality of abstraction and developer productivity:
  - Ex: ABAP (as a 4GL) made developers more efficient creating reports and other enhancements
- Although there have been incremental improvements to ABAP over the years, productivity curves have largely plateaued



# **A Fresh Approach**

**Getting Started with Low-Code Development Platforms**

# A New Abstraction Model

- Low-code development platforms (LCDP) have introduced a new style of development focused on:
  - [Model-driven](#) or declarative design
  - Reusable components and templates (think Lego™)
  - Visual programming using graphical WYSIWYG editor tools
  - Automatic code generation
  - Simplified logic (think Excel formulas)



# Why Low-Code Tools Matter

1. LCDPs significantly lower the barrier to development, unlocking the door for [citizen developers](#)
2. They also make developers much more efficient
  - Developer productivity here at Bowdark is up **over 60%** with low-code tools
3. They offer unprecedented agility (e.g., DevOps, ALM)
4. They unlock many new app scenarios
5. They significantly reduce the TCO for custom solutions

# WYSIWYG Designer Tools



Power Apps | Form

Search

Environment: D365 PO Test

James Wood

Back | Form field | Component | Form settings | Switch to classic | Save | Publish

**Table columns**

Search

+ New table column

☒ Show only unused table columns

- Created By
- Created By (Delegate)
- Created On
- Currency
- 0.0 Exchange Rate
- Modified By
- Modified By (Delegate)
- Modified On
- Owning Business Unit
- Purchase Price (Base)
- Status
- Status Reason

**New Equipment**  
Equipment

**General**

Name	---
Category	---
Description	---
Vendor	---

**Manufacturing Information**

Manufacturer	---
Model Number	---
Serial Number	---
Order Number	---

**Price and Assignment**

Purchase Date	---	Purchase Price	---
Assigned To	---	Assigned On	---

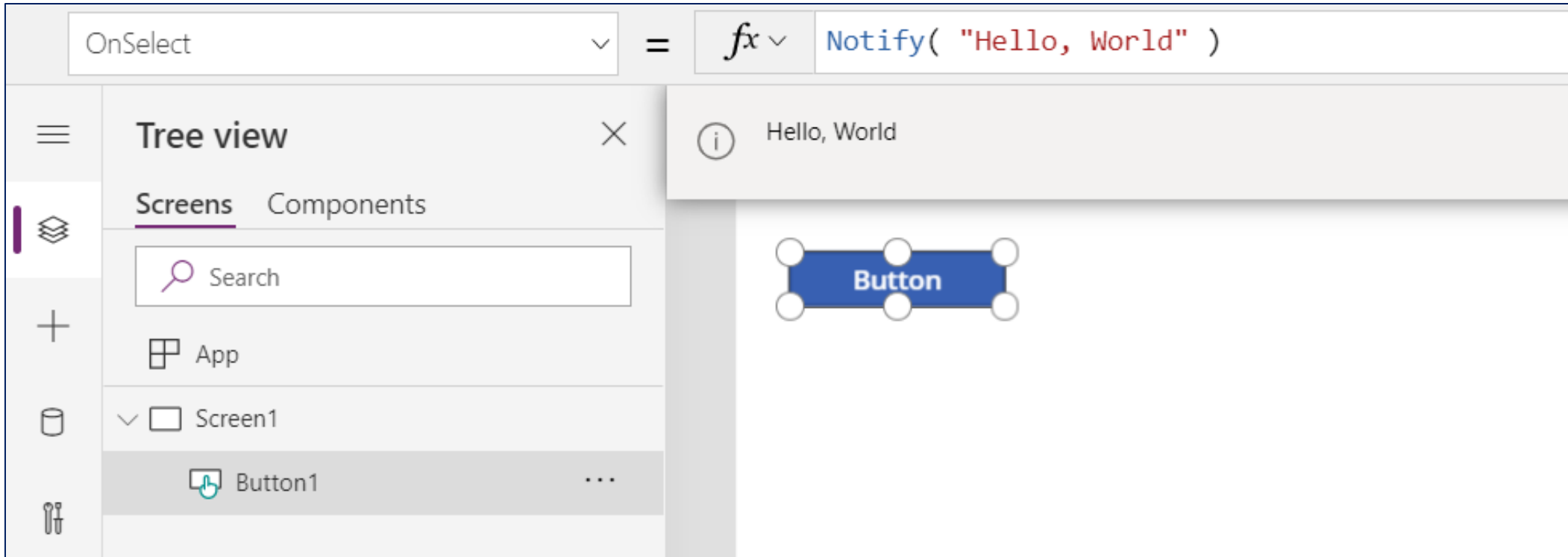
**Customer Information**

Customer Asset	No	Customer	---
----------------	----	----------	-----

Equipment main form

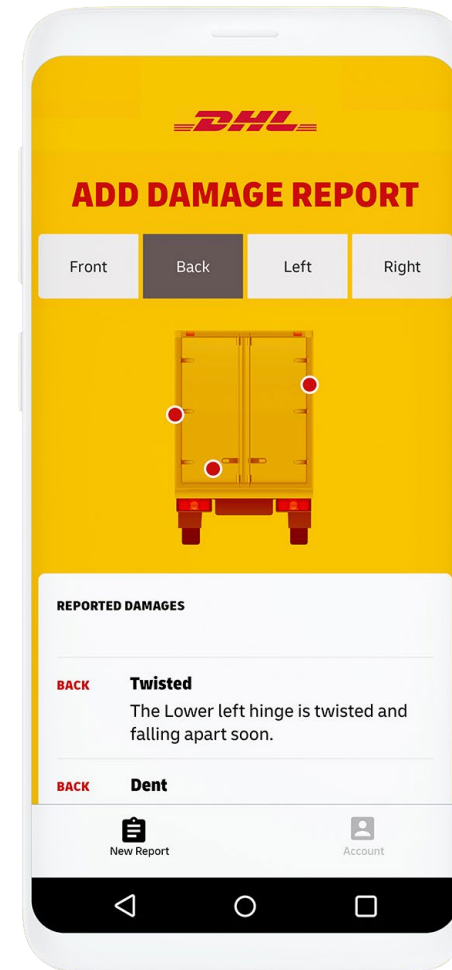
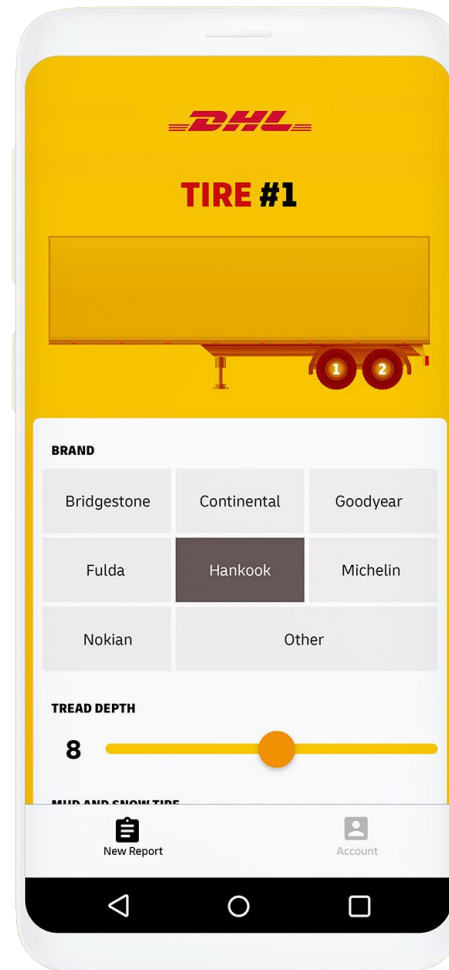
Desktop (1920 x 1080) | Show Hidden: Off | 64%

# Simplified App Logic

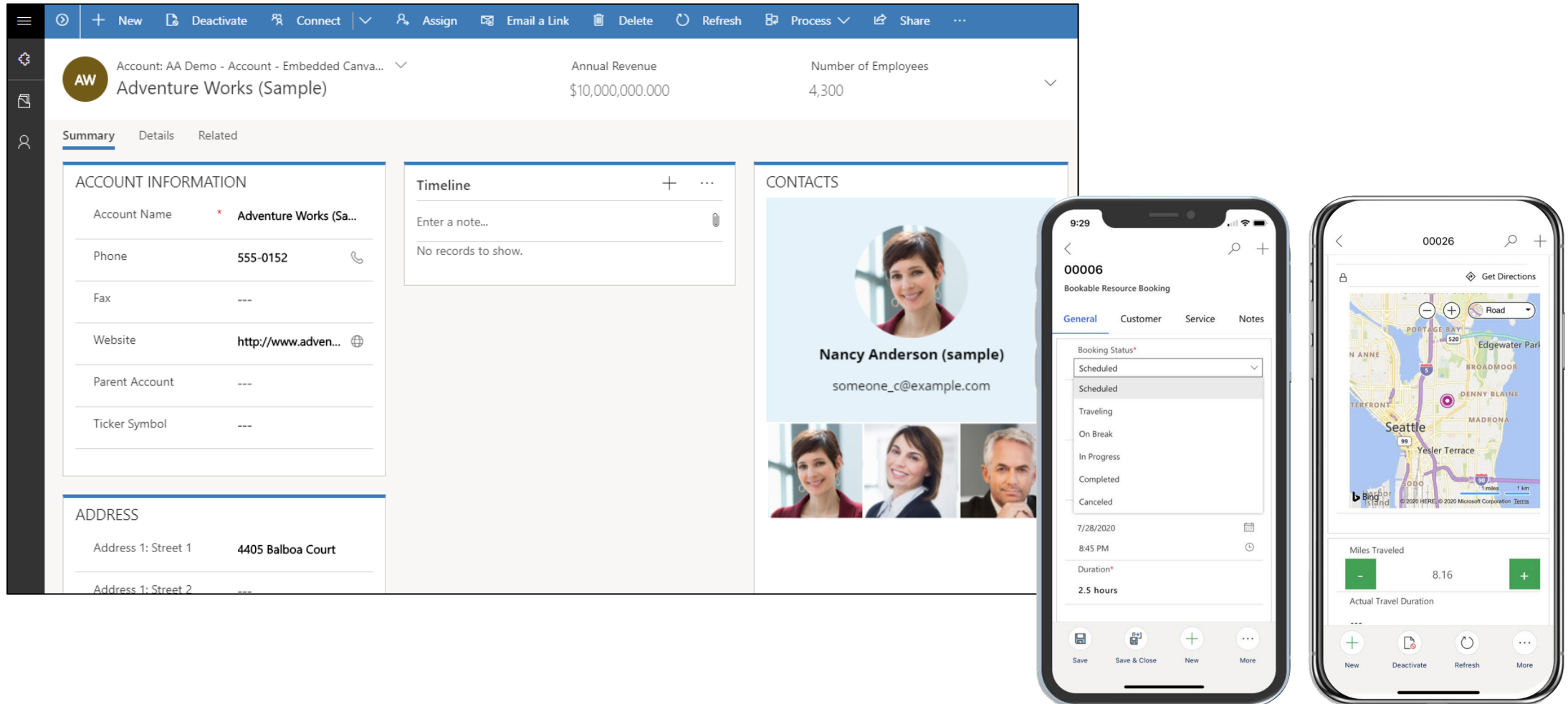


The screenshot displays a user interface for defining app logic. At the top, a logic rule is defined as `OnSelect = fx Notify( "Hello, World" )`. Below this, a 'Tree view' sidebar on the left contains a search bar and a hierarchical list of components: 'App', 'Screen1', and 'Button1'. The 'Button1' component is currently selected. The main workspace on the right shows a visual representation of a blue button with the text 'Button' and six white circular handles at its corners. Above the workspace, a status bar displays an information icon and the text 'Hello, World'.

# SAP AppGyver Examples



# MS Power Apps Examples



The image displays a Microsoft Power Apps canvas application interface for an account named "Adventure Works (Sample)". The app features a top navigation bar with various actions like New, Deactivate, Connect, Assign, Email a Link, Delete, Refresh, Process, Share, and more. The main content area is divided into sections: Account Information, Timeline, and Contacts.

**Account Information:**

- Account Name: Adventure Works (Sa...)
- Phone: 555-0152
- Fax: ---
- Website: http://www.adven...
- Parent Account: ---
- Ticker Symbol: ---

**Timeline:**

- Enter a note...
- No records to show.

**Contacts:**

- Nancy Anderson (sample)
- someone\_c@example.com

**Address:**

- Address 1: Street 1: 4405 Balboa Court
- Address 1: Street 2: ---

Two smartphones are overlaid on the right side of the screen, showing mobile views of the app:

- The first smartphone displays a "Bookable Resource Booking" screen for "00006". It shows a "Booking Status" dropdown menu with options: Scheduled, Traveling, On Break, In Progress, Completed, and Canceled. Below the menu, it shows a date "7/28/2020", a time "8:45 PM", and a duration of "2.5 hours".
- The second smartphone displays a map view of Seattle, showing locations like Edgewater Park, Broadmoor, Denny Blaine, Madrona, and Yesler Terrace. It includes a "Miles Traveled" section with a value of 8.16 and an "Actual Travel Duration" section.

# Just Add ~~Water~~ APIs

- REST (OData) APIs are the lifeblood of LCDPs
- When defined *properly*, APIs can be:
  - Used to create new user experiences
  - Mixed-and-matched to create app mash-ups (ex: SAP + Salesforce)
  - Leveraged by a wider audience of developers
  - Shared across the enterprise



# From APIs to Connectors



Office 365



Outlook



SharePoint



OneDrive



MS Teams



Slack



Notifications



SAP



Salesforce



Dynamics 365



Workday



Adobe Sign



DocuSign



Stripe



SQL Server



Oracle



Mainframes



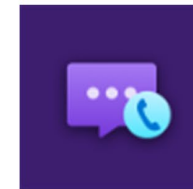
AWS



FTP / Files



ArcGIS



Azure  
Communication  
Services

# More Than Just Apps

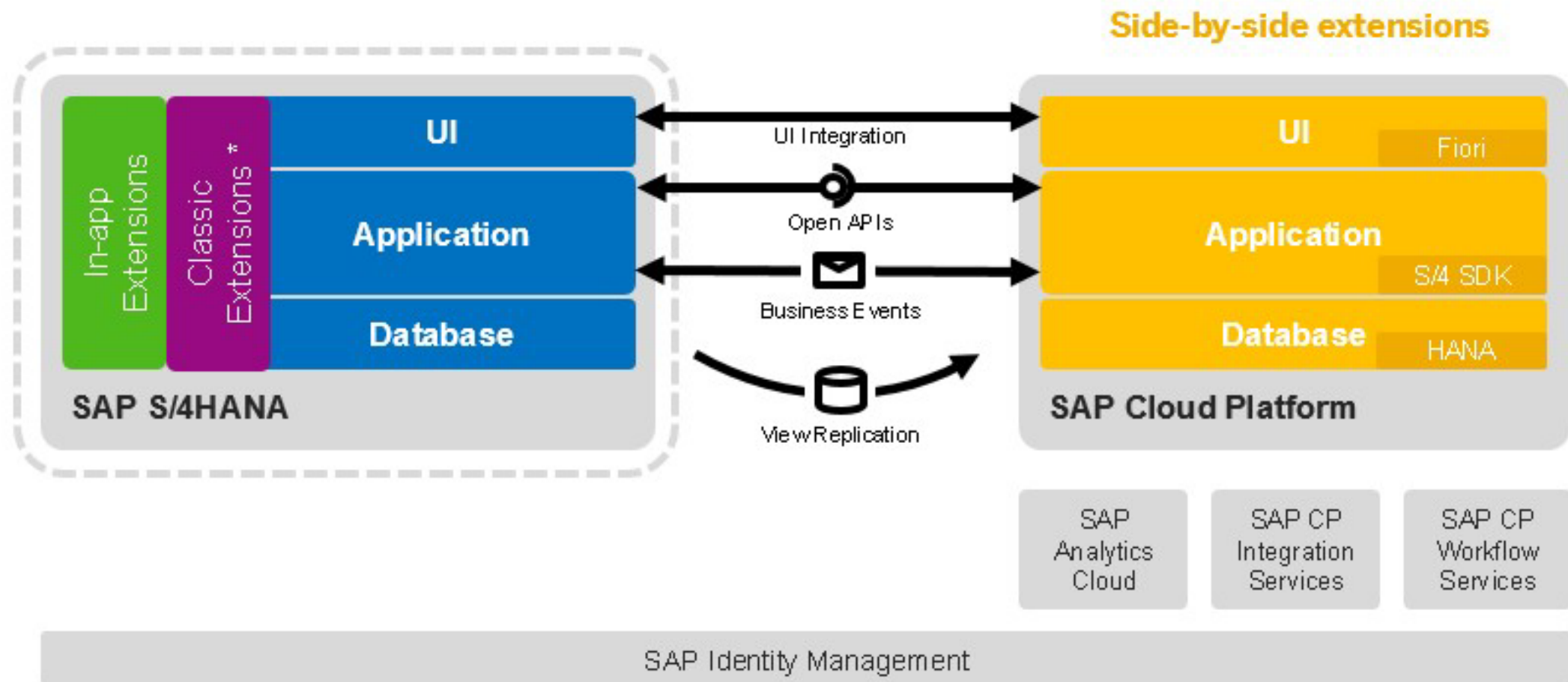
- Besides web and mobile apps, LCDPs can also be used to (rapidly) develop:
  - Self-service portals
  - Spreadsheet replacements (Excel + Access)
  - Interfaces
    - Ex: Microsoft provides over 700 connectors OOTB to easily connect SAP with various cloud and on-premises business systems
  - Workflows
  - Automations (RPAs)
  - AI & machine learning-based solutions
  - Reports & dashboards
  - Chatbots

“

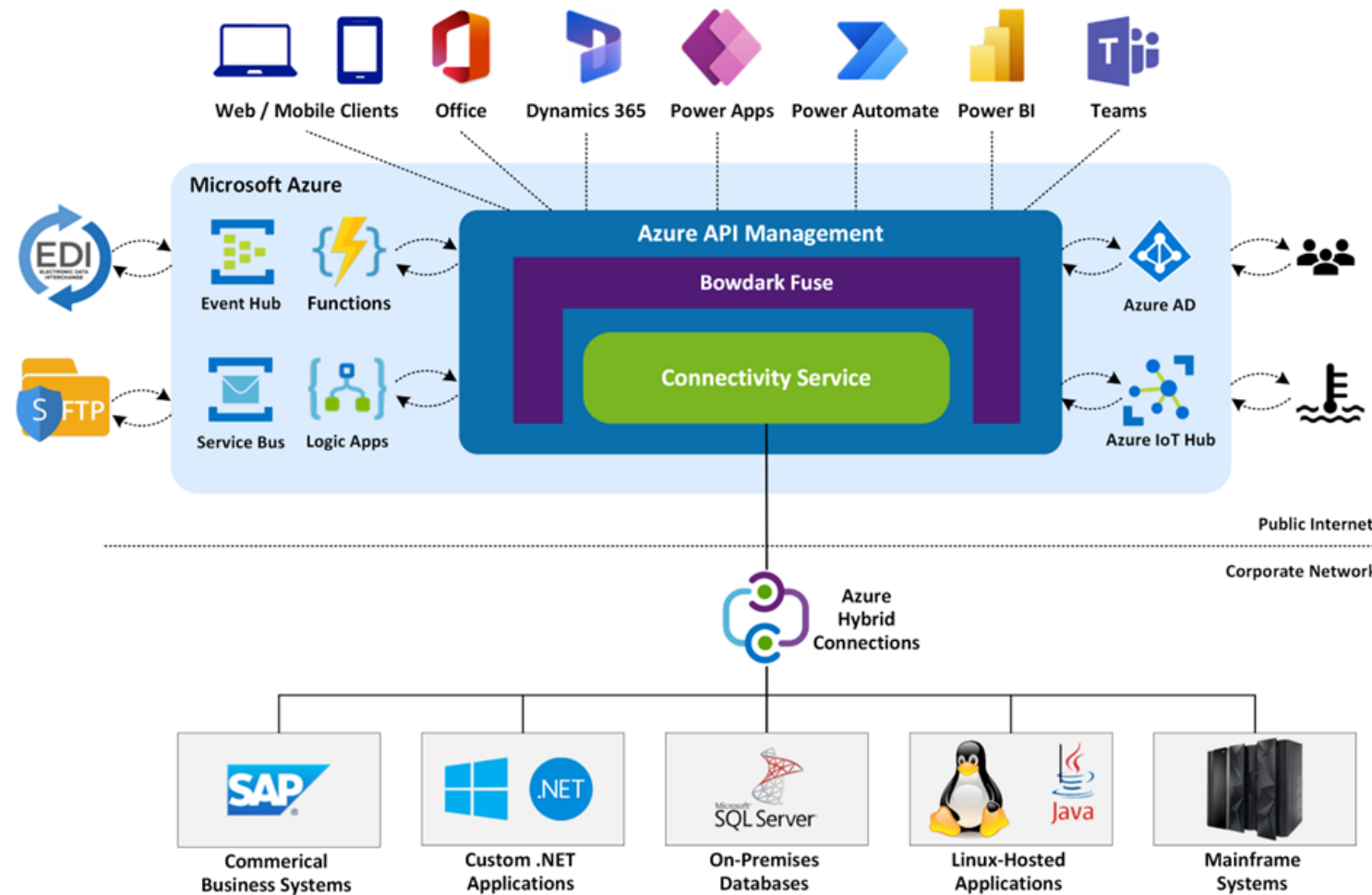
*“The best app is no app”*

- Eleanor Roosevelt (Probably)

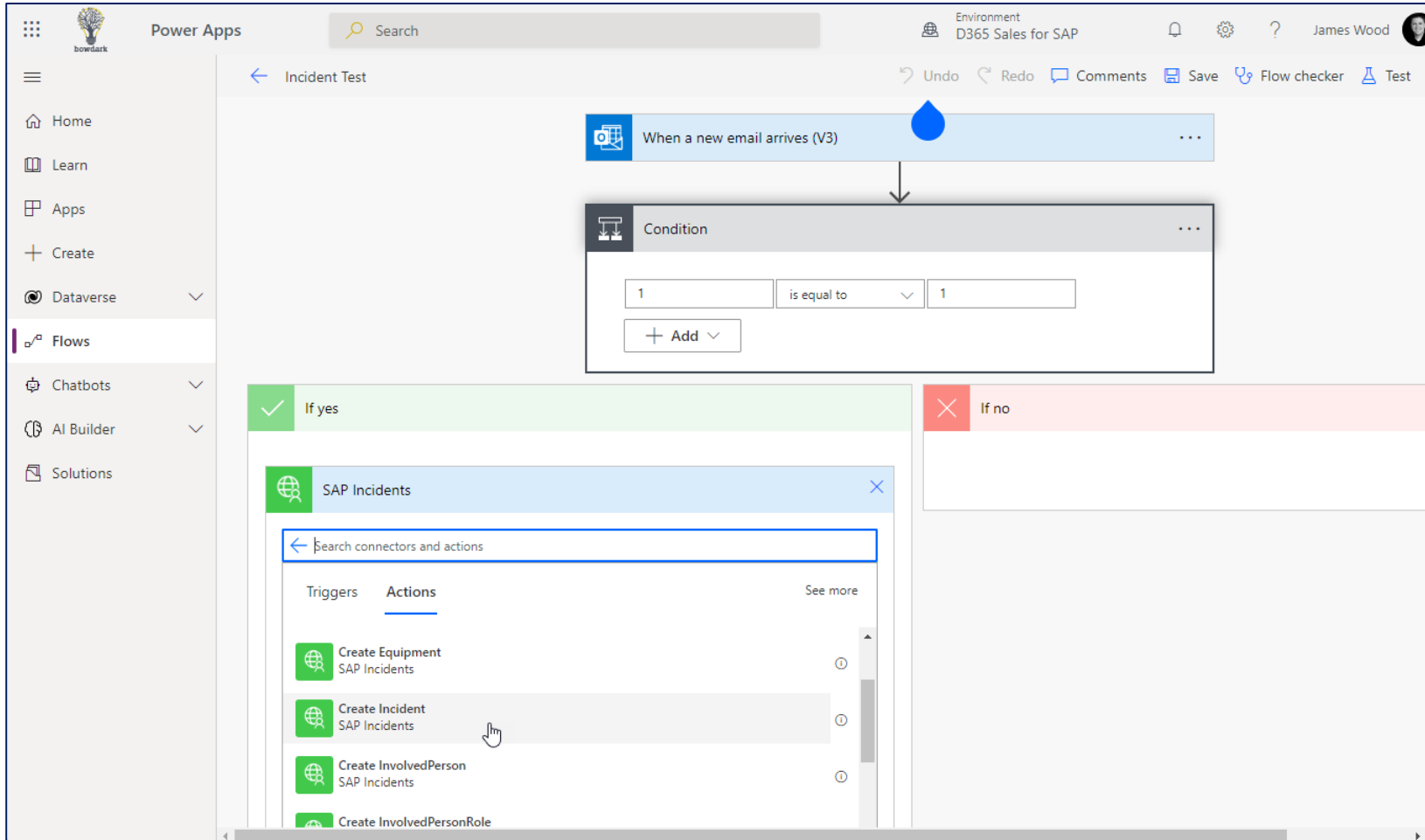
# Pro-Code Extension Concept



# Low-Code Extension Concept

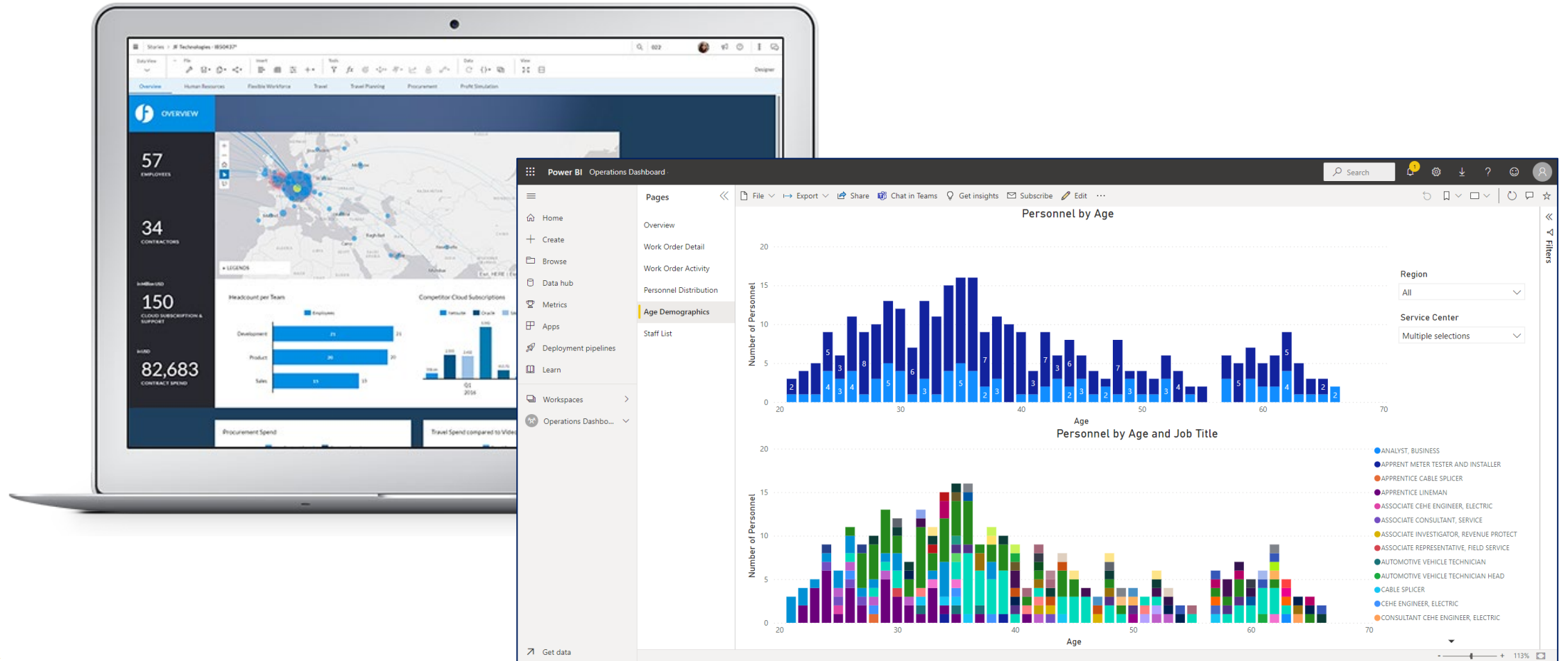


# Workflows & Automations



The screenshot displays the Microsoft Power Apps interface for editing a workflow named "Incident Test". The top navigation bar includes the "Power Apps" title, a search bar, and the environment "D365 Sales for SAP". The left sidebar shows the navigation menu with options like Home, Learn, Apps, Create, Datasource, Flows (selected), Chatbots, AI Builder, and Solutions. The main workspace shows the workflow steps: a trigger "When a new email arrives (V3)" followed by a "Condition" step. The condition is configured with the value "1" and the operator "is equal to" and "1". Below the condition, there are two paths: "If yes" (green) and "If no" (red). The "If yes" path is expanded, showing a search for "SAP Incidents" connectors and actions. The search results list several actions, with "Create Incident SAP Incidents" highlighted by a mouse cursor.

# Self-Service Analytics



# **Breaking Down Barriers**

**Extending the Reach of SAP**



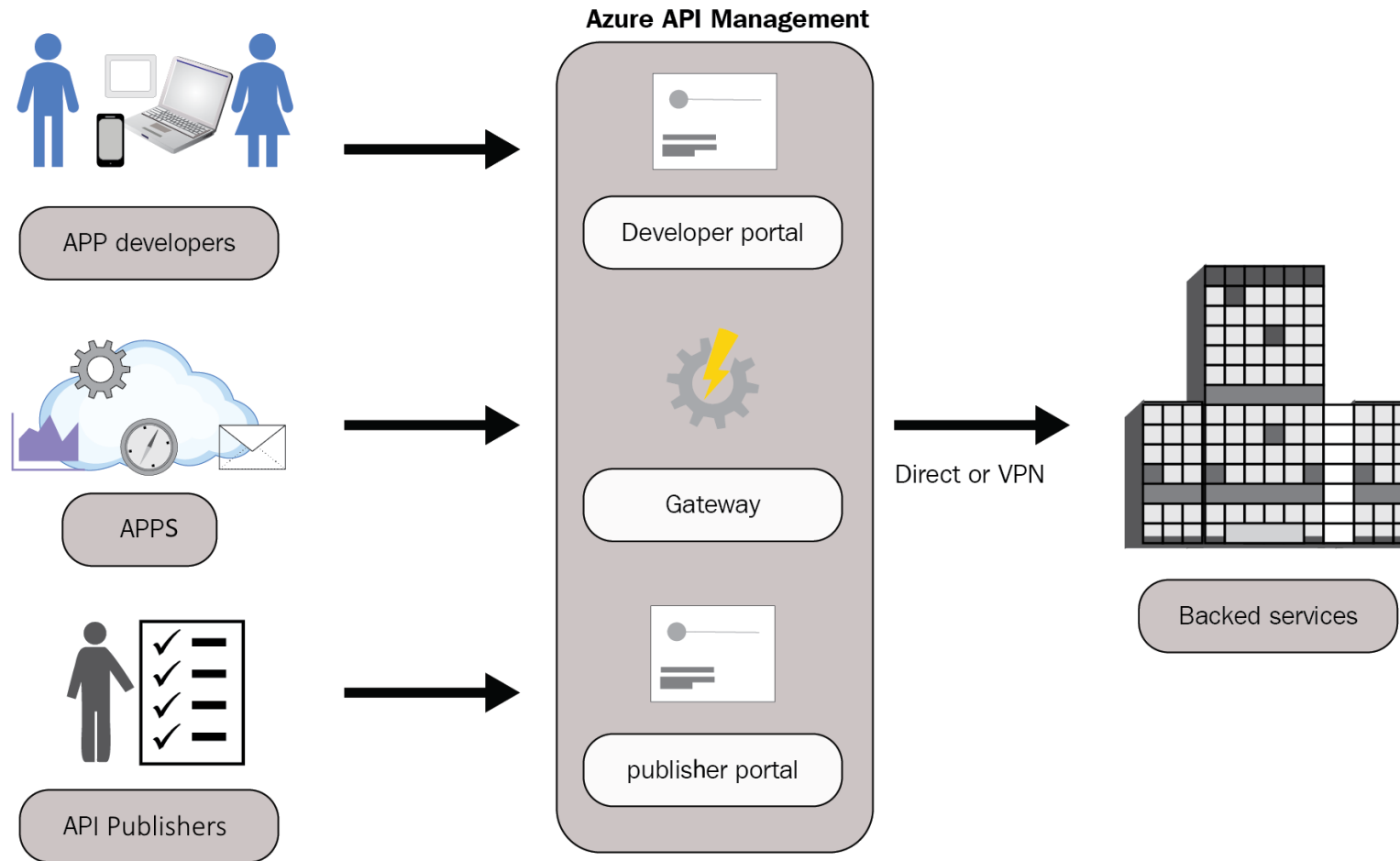
# The Need for SAP APIs

- APIs (specifically [OData](#) APIs) are essential for (securely) exposing SAP functionality to LCDPs
- Although SAP is continually expanding its API catalog (e.g., through the [SAP API Business Hub](#), etc.), there are still many gaps
- Bottom Line: Most apps/solutions will require the development of custom APIs (at least until a critical mass of APIs is attained)

# Stretching Pro-Code Resources

- Using a low-code approach, pro-code development can be limited to just API development
- With SAP Gateway and HANA these APIs can come together quickly:
  - SAP data can be modeled using [Core Data Services](#) (CDS)
  - From here, there are many [tools](#) that make it easy to (securely) expose SAP data through OData services
- These investments offer many repeat benefits as they can be reused in other app scenarios

# Publishing APIs for Reuse



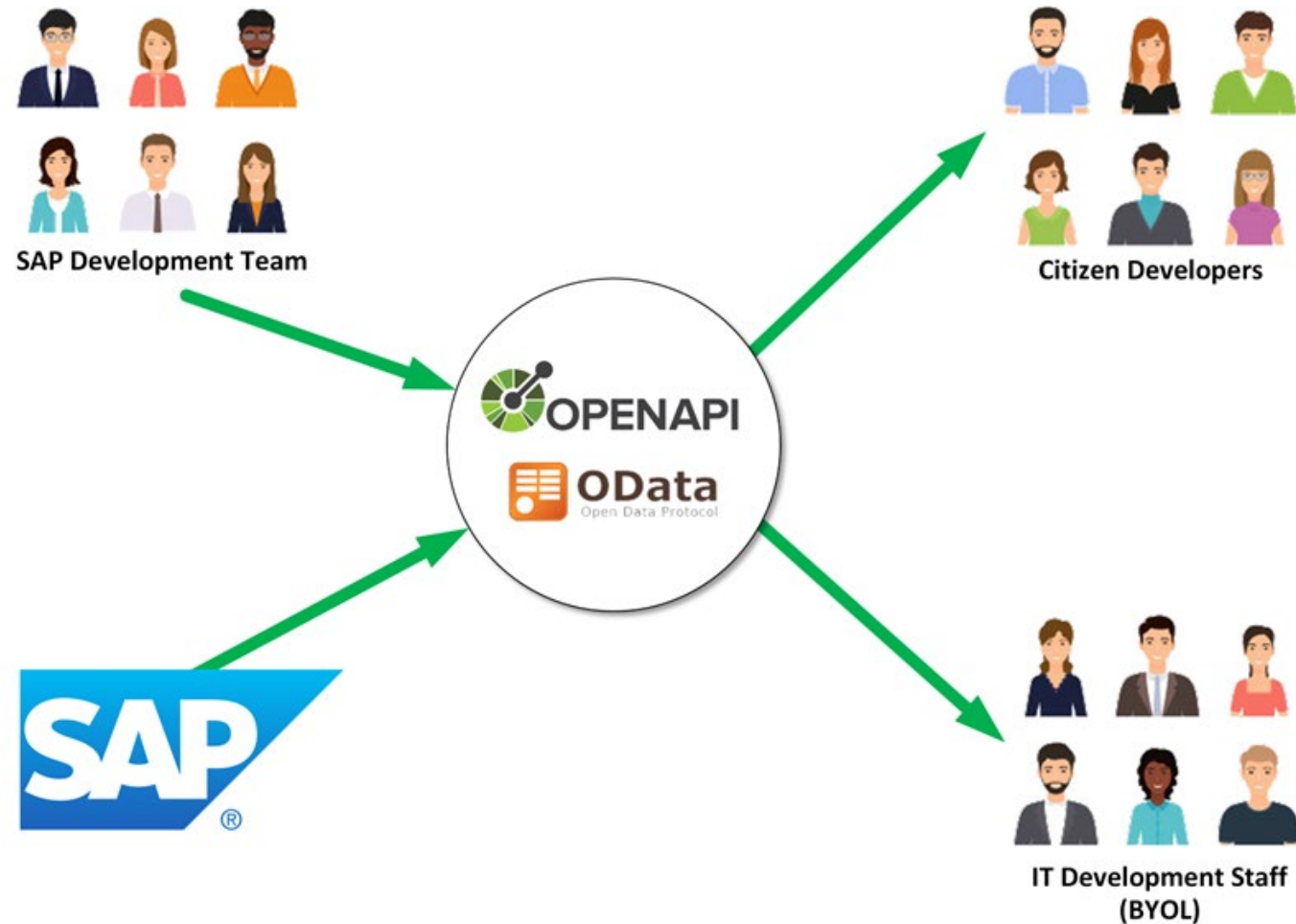
# **Getting Started with Fusion Teams**

**Enabling a Wider Audience for SAP Development**

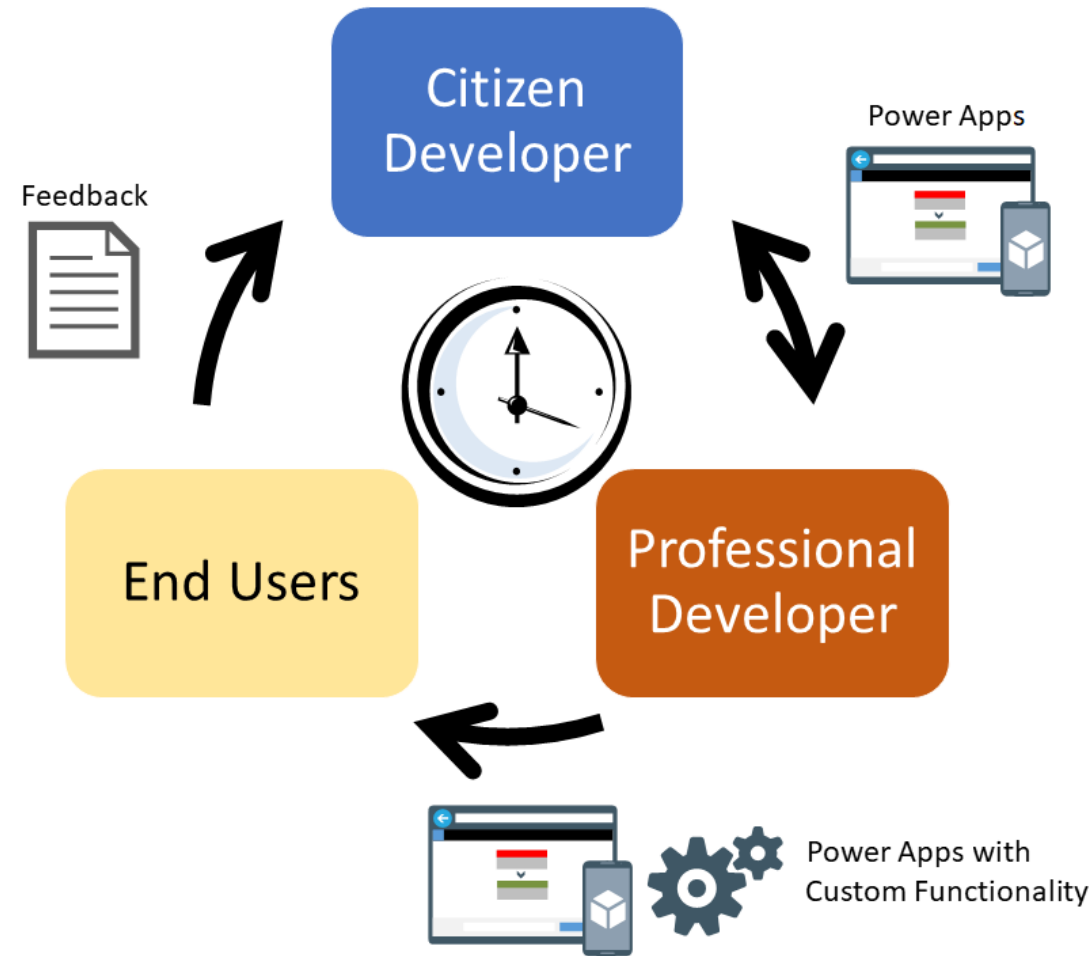
# Introducing Fusion Teams

- Gartner defines [fusion teams](#) as “multidisciplinary digital business teams that blend technology and business domain expertise to drive initiatives to create digital products and solutions”
- This combined approach allows the team to play to their respective strengths:
  - Business analysts and SMEs can focus on the improving business processes
  - IT / pro-code developers can fill in technical gaps where needed

# Low-Code + Pro-Code Unite



# A New Development Paradigm



# **Wrap-Up**



# Key Points to Take Home

- SAP pro-code development still has its place but there are some notable limitations that make it difficult to gain traction with digital transformation initiatives
- Low-code development platforms can be used to innovate around the edges with SAP
- Regardless of your direction, it's vitally important to focus your attention on SAP API enablement
- Embracing the fusion team concept can supercharge your digital transformation journey

# Q&A

Questions ==> { Answers }

# Thank you!

Office Phone: (972) 691-2101

Email: [info@bowdark.com](mailto:info@bowdark.com)

Web: <https://www.bowdark.com>