

The code first developer's toolbox for Power Platform integration

April Dunnam, Principal Cloud Advocate
Scott Durow, Solutions Architect, MVP

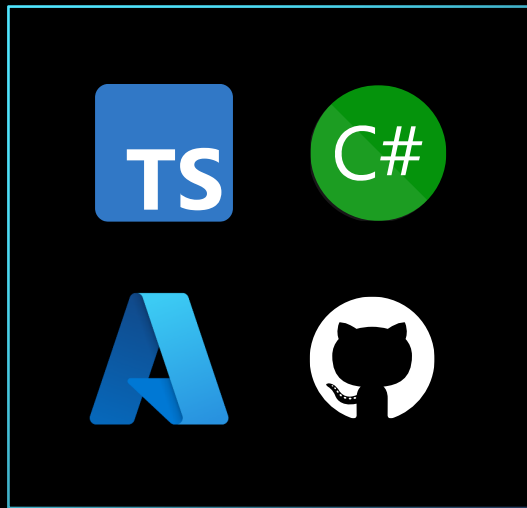


So you've been told to use
'Low-Code' ?!

"But I'm a pro-coder – now what?"



Why would I use the Power Platform instead of our existing software development techniques?





Power Apps

Application development



Power Automate

Process automation



Power BI

Business analytics



Power Virtual Agents

Intelligent bots



Power Pages

External websites



**Data
connectors**



**AI
Builder**



**Microsoft
Dataverse**

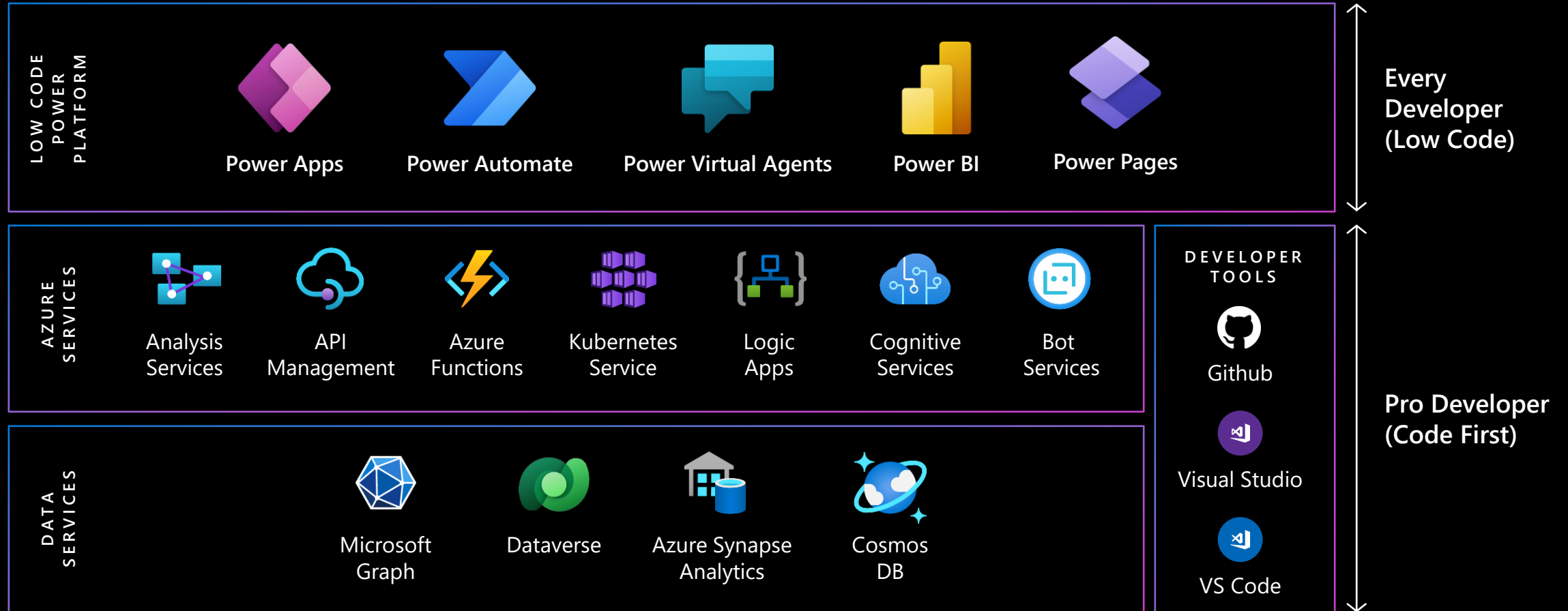


**Power
Fx**



**Managed
Environments**

Pro Developers + Power Platform = No Limits



VS APIs + Azure API Mgt + Power Platform

One of many E2E Scenarios



Visual Studio

Azure

Power Platform

Developer
creates APIs and
code first
functions/UX in
Visual Studio

Developer
stores APIs
and data in
Azure

Makers leverages
data & APIs to
create and share
apps, flows &
bots with the
Power Platform

Professional
developers

Scale through
**Co-authoring &
Collaboration**

IT
professional

Business
analysts



What makes the Power Platform different from low code tools of the past?



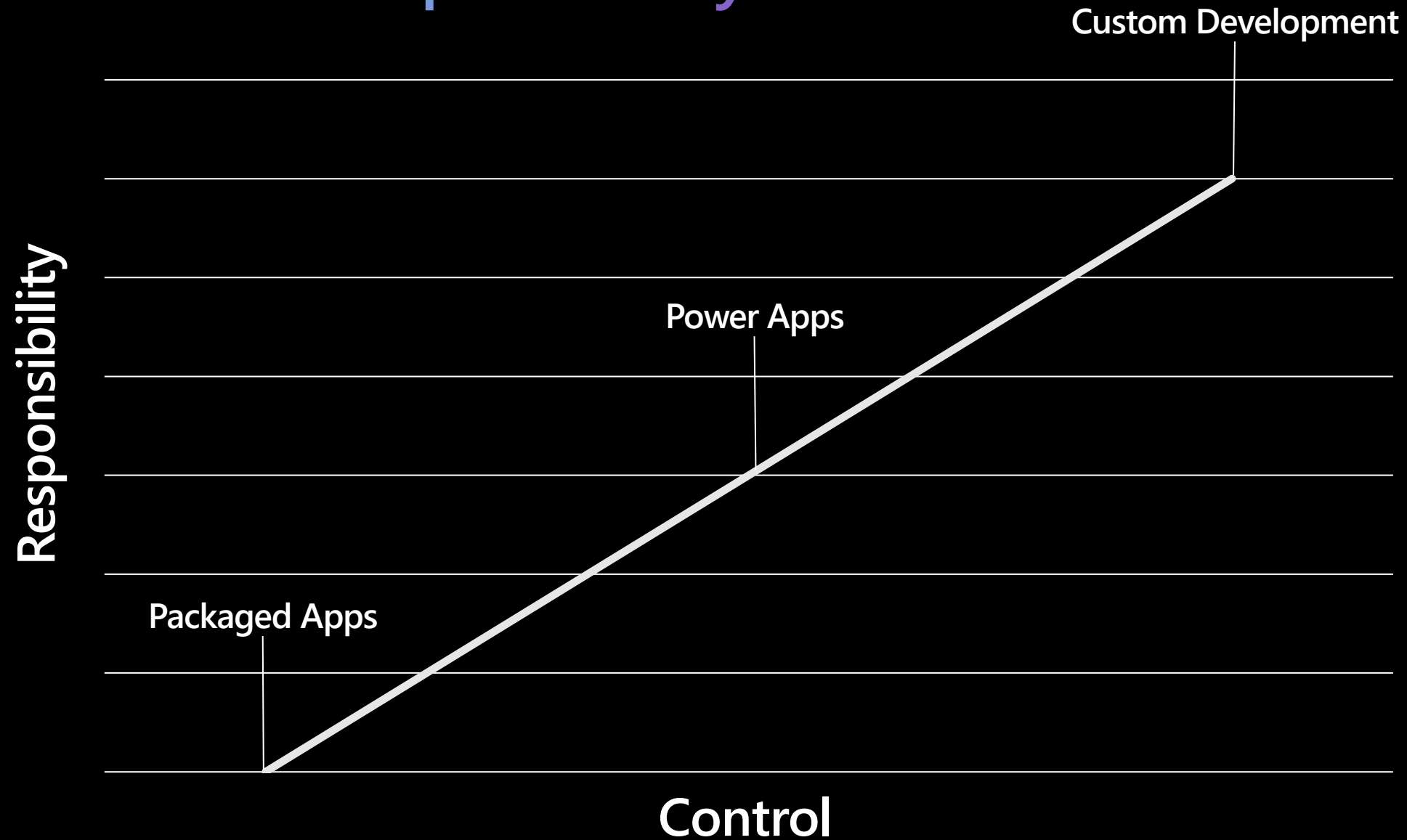
Pro-dev + Power Platform

1. Low-code isn't always low-complexity – but it will help reduce complexity.
2. We've seen jumps in abstraction before – but this time it's with increased focus on:
 - Connectivity
 - Security, Administration, Governance, & Monitoring
 - Application Life Cycle Management & Versioning
 - Modern Workplace Integration
 - Component based collaboration & co-authoring (often between pro-coders and low-code makers)

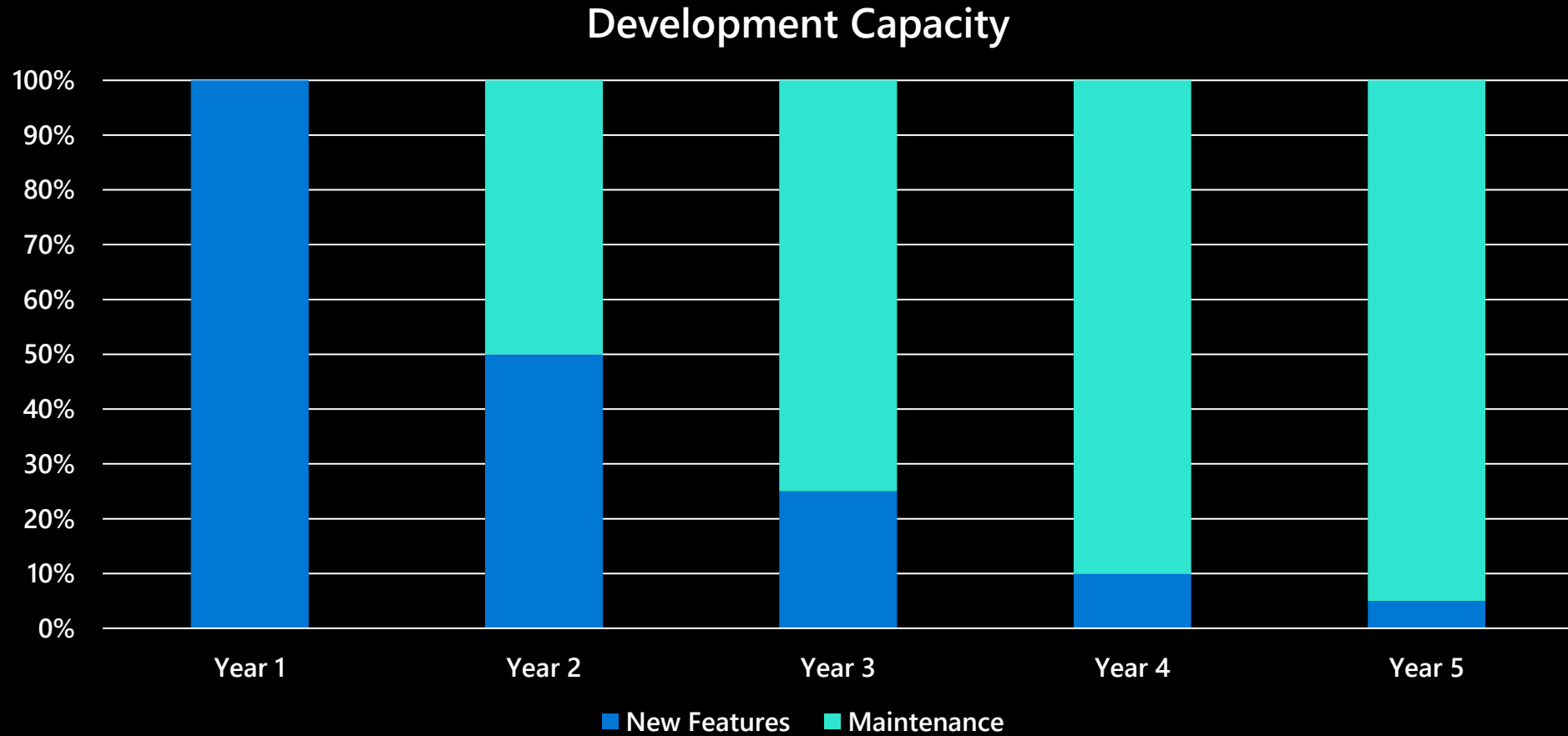
How will the Power Platform make our
development faster?



Control vs. Responsibility



The Technical Debt Problem



How will the Power Platform make development faster?

1. Less code to maintain

- Less libraries to constantly update with security patches and updates
- Less database scripts to manage
- Less plumbing – security, network calls, asynchronous tasks, offline, mobile apps & more

2. Reduced framework breaking changes

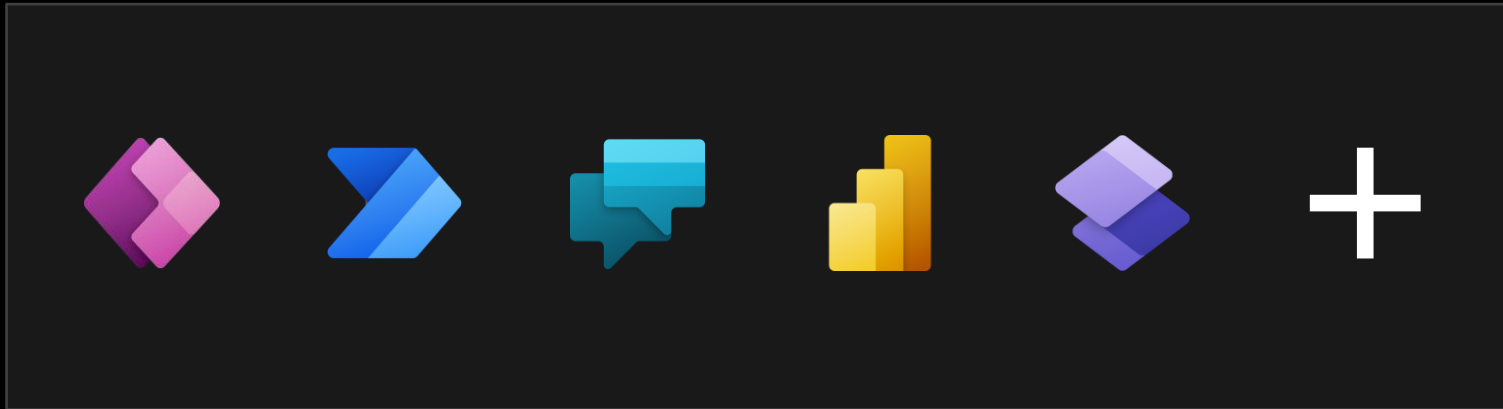
3. Component based and metadata driven development

4. Authentication, Authorization & User Management built-in

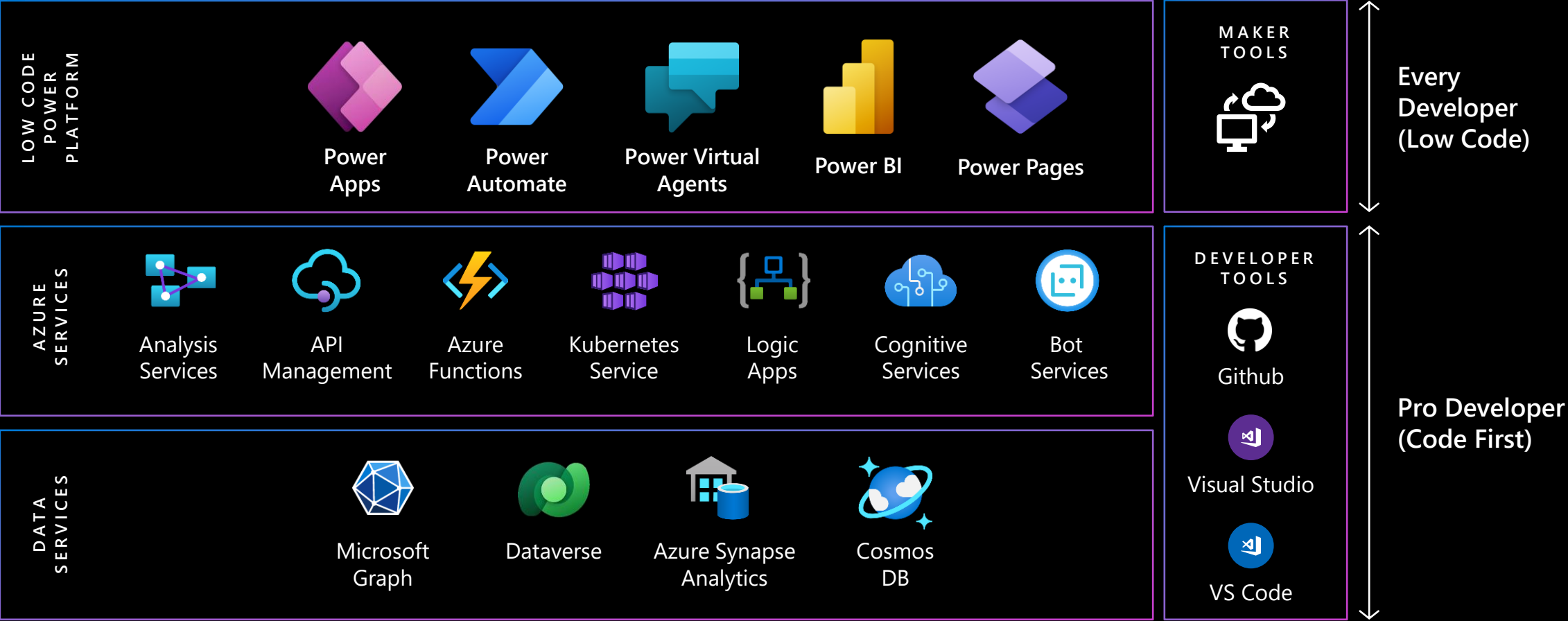
5. Simplified deployment & versioning (especially database schema)

Does the Power Platform limit my control?

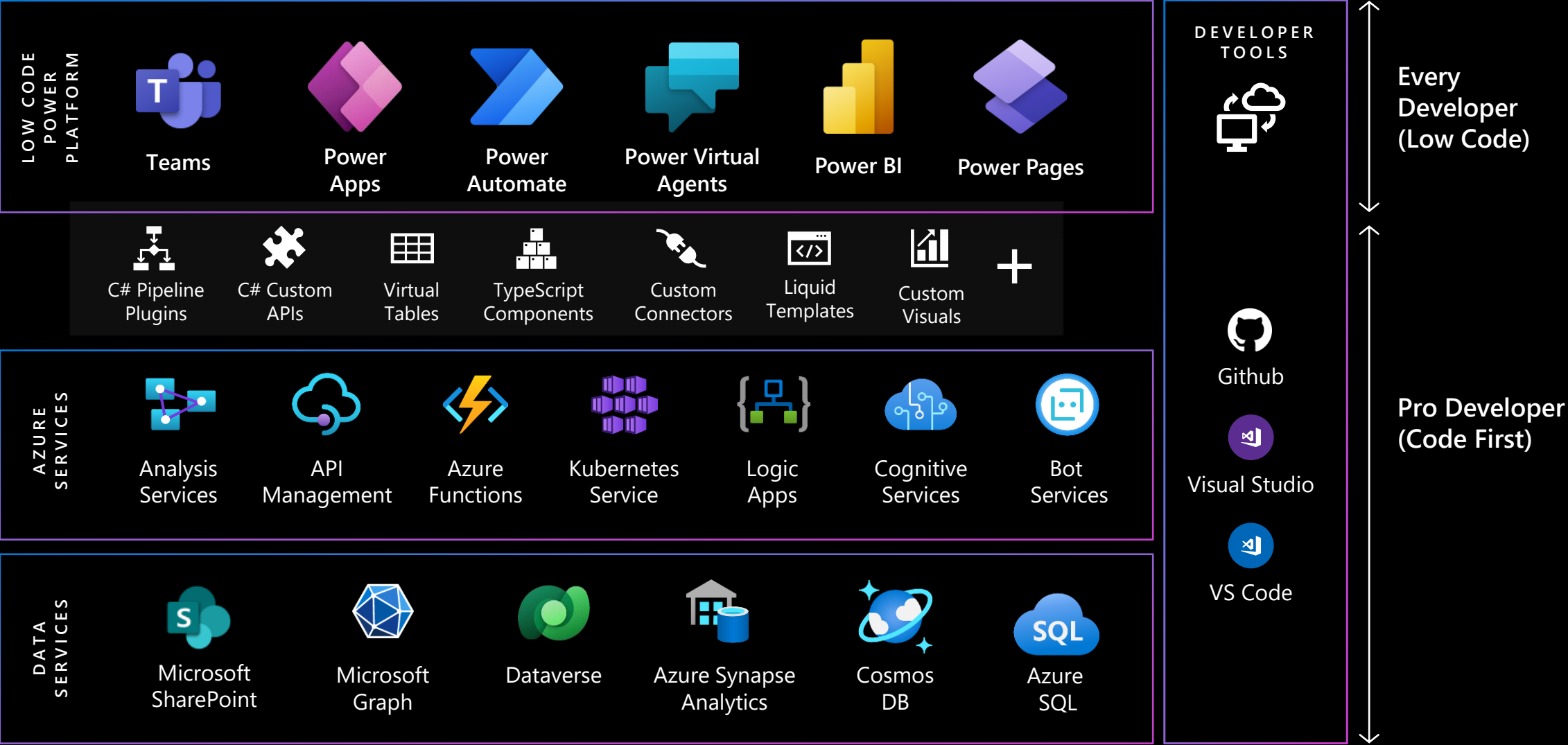
How do I extend it?



Extending the Power Platform

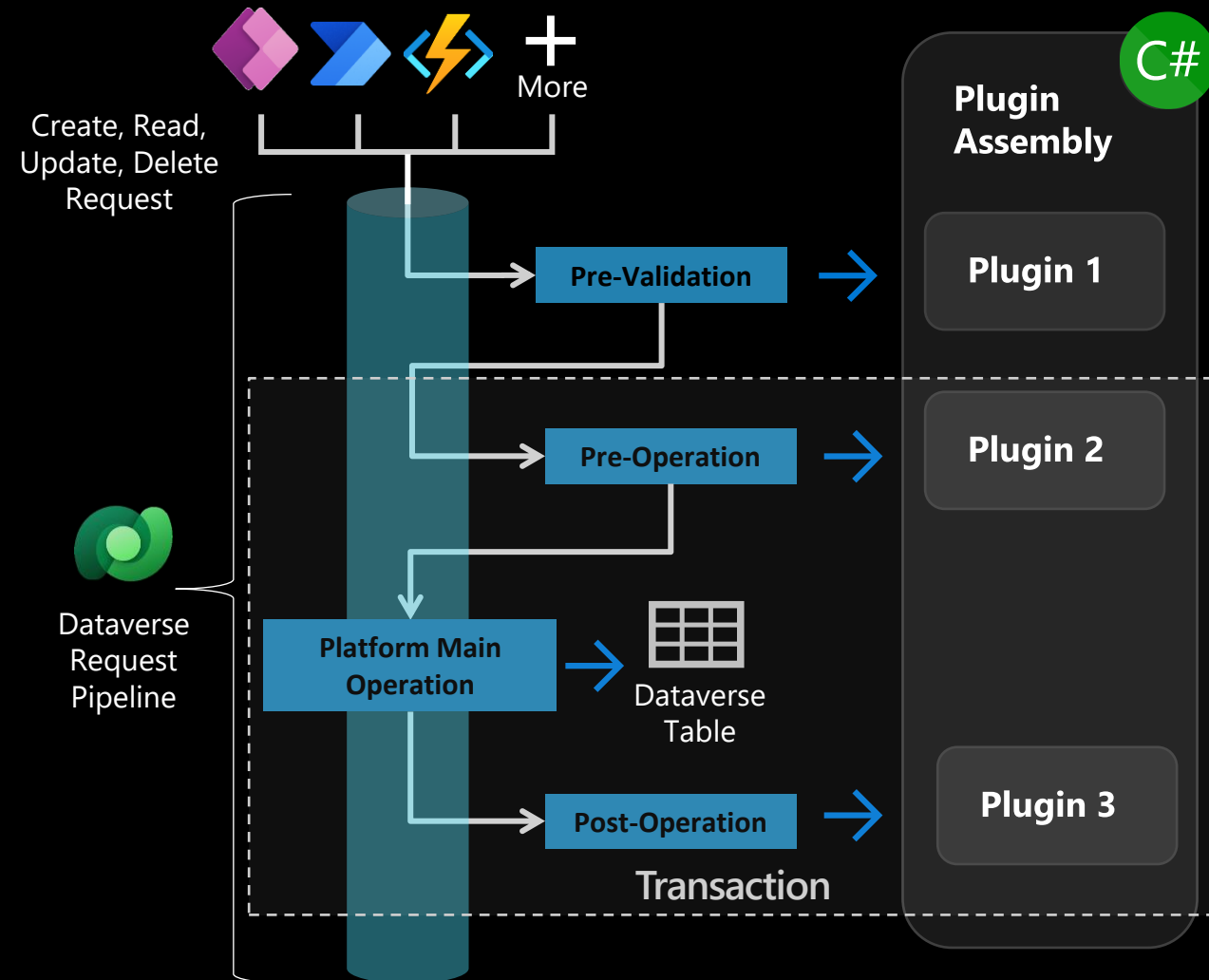


Extending the Power Platform

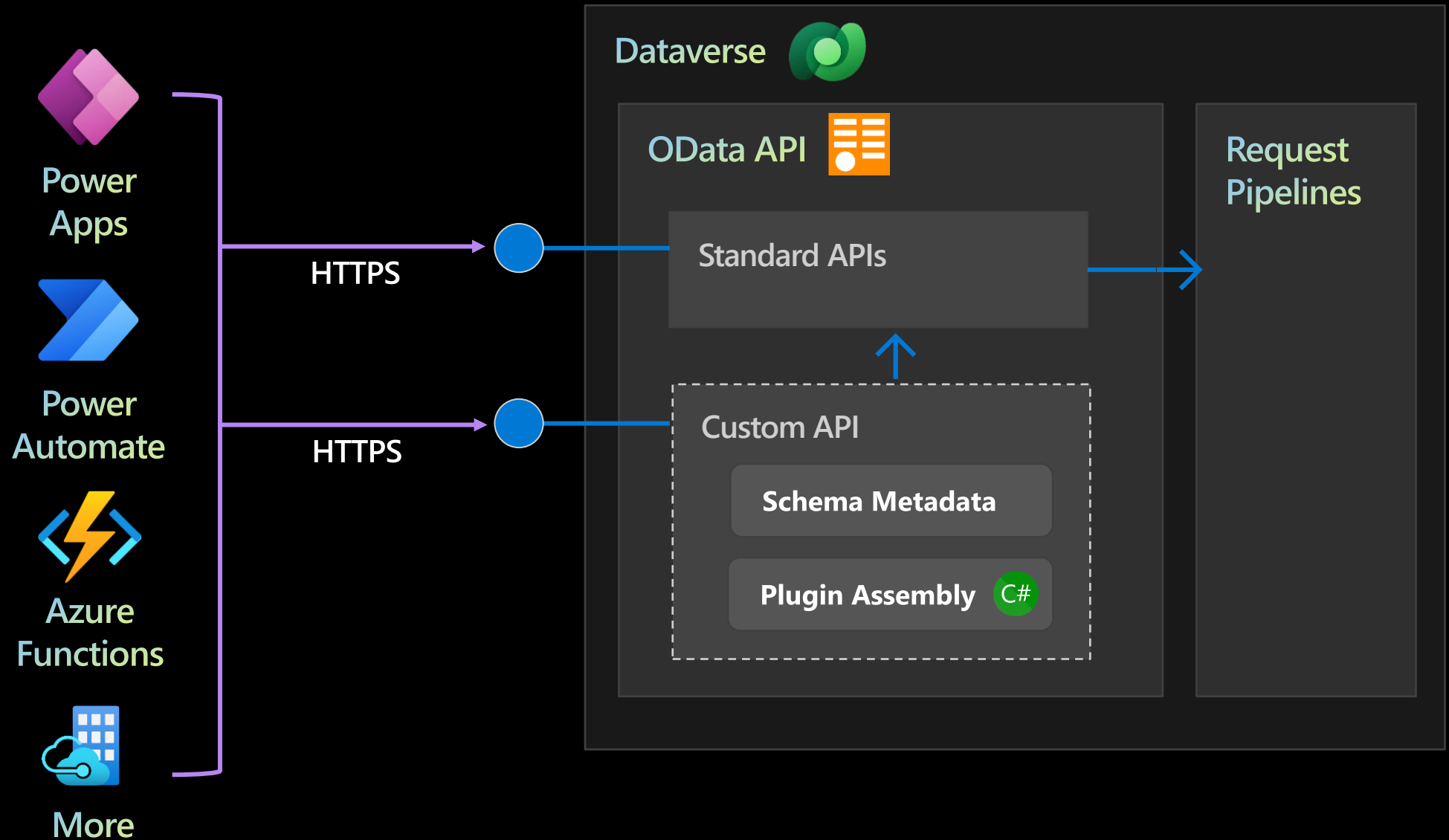


C# Dataverse Pipeline Plugins

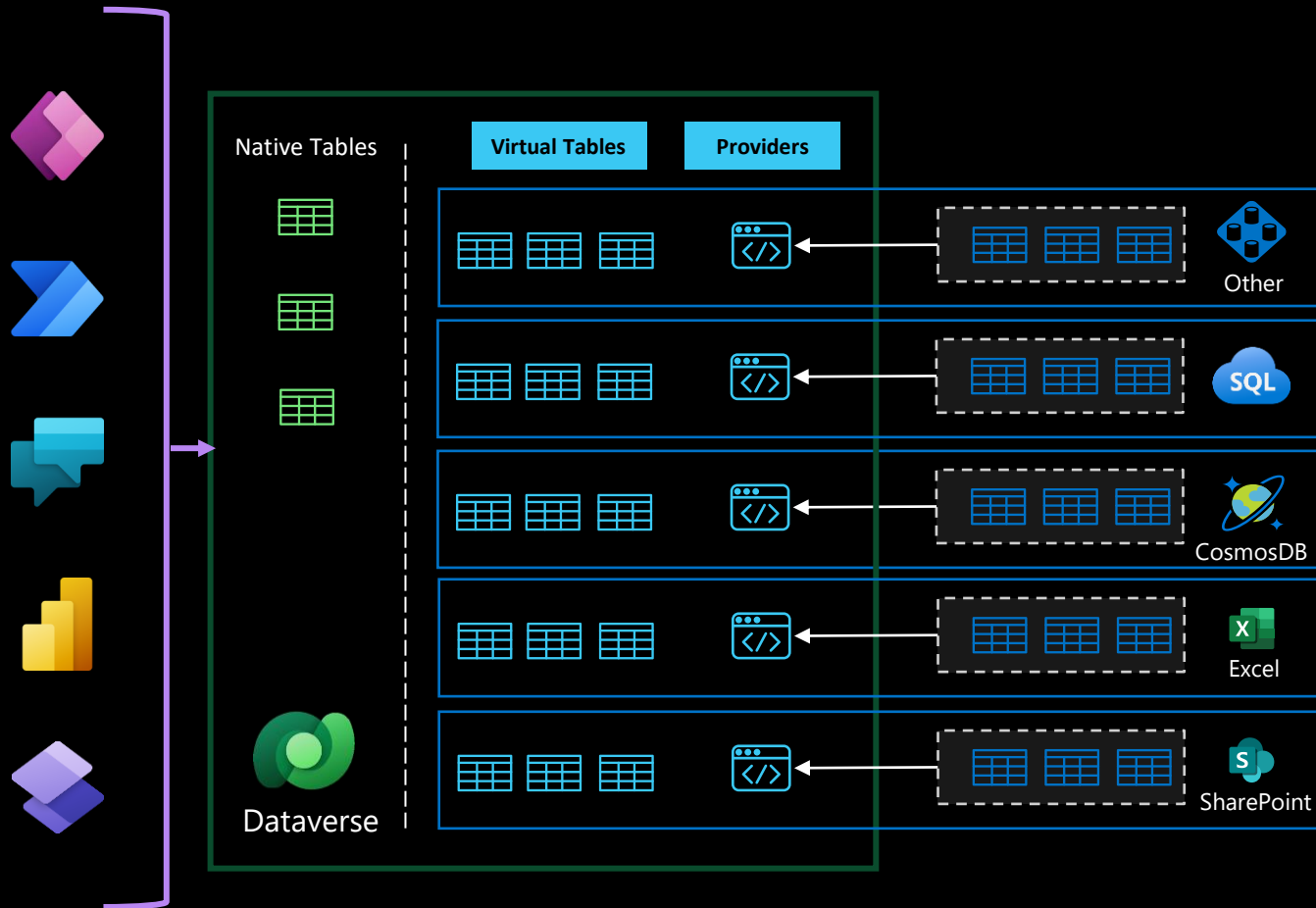
- Adds logic before/during/after a Dataverse transaction
- Logic will execute irrespective of where the Dataverse API is called from
- Best for Fast and Synchronous logic that benefits from executing inside the Dataverse context



C# Dataverse Custom APIs



Virtual Tables



- Virtual tables “attach” external tables or collections to Dataverse
- Exposes *consistent* APIs for working with local and remote stores
- Supports cross-data-source relationships
- Create/Read/Update/Delete
- C# Custom data providers

New table from external data wizard (Preview)

The screenshot displays the Microsoft Power Apps home page. The top navigation bar includes the 'Power Apps' logo, a search bar, and user information for 'FrankTip2'. The left sidebar contains navigation links: Home, Learn, Apps, Create, Datasource, Flows, Chatbots, AI Builder, Cards (preview), Solutions, DevTools, and GraphQLTools.

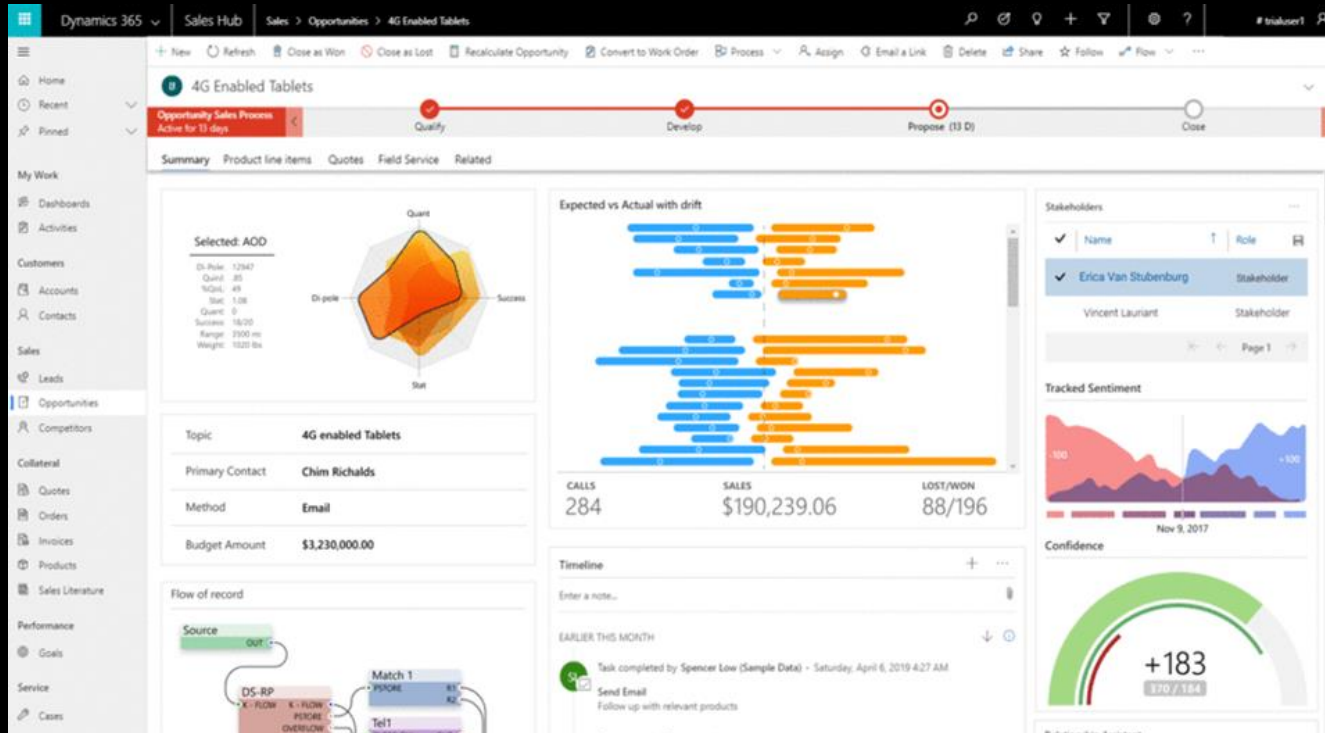
The main content area is titled 'Build business apps, fast' and includes a link to 'Learn about Power Apps'. Below this, the 'Start from' section offers five templates: Blank app, Datasource, SharePoint, Excel, and SQL, each with a 'Watch video' link. A 'More create options' link is also present.

The 'Learning for every level' section features four video tutorials: 'Get started with Power Apps' (Beginner, 51 mins), 'Author a basic formula to change properties in a...' (Beginner, 42 mins), 'Work with external data in a Power Apps canvas app' (Intermediate, 43 mins), and 'Manage and share apps in Power Apps' (Beginner, 42 mins).

The 'Your apps' section shows a table of existing apps. A yellow banner at the top of the table indicates that '2 connection references need to be updated.'

Name	Modified	Owner	Type
Asset Checkout	1 wk. ago	Frank Lan	Model-driven
Innovation Challenge	1 wk. ago	Frank Lan	Model-driven

Power Apps Component Framework (PCF)



Developers can build compelling visual components for Power Apps



Custom components are responsive, reusable and support multiple form factors



Reuse your current IP and skills– framework is based on standard web technologies TypeScript/JS, CSS and HTML5

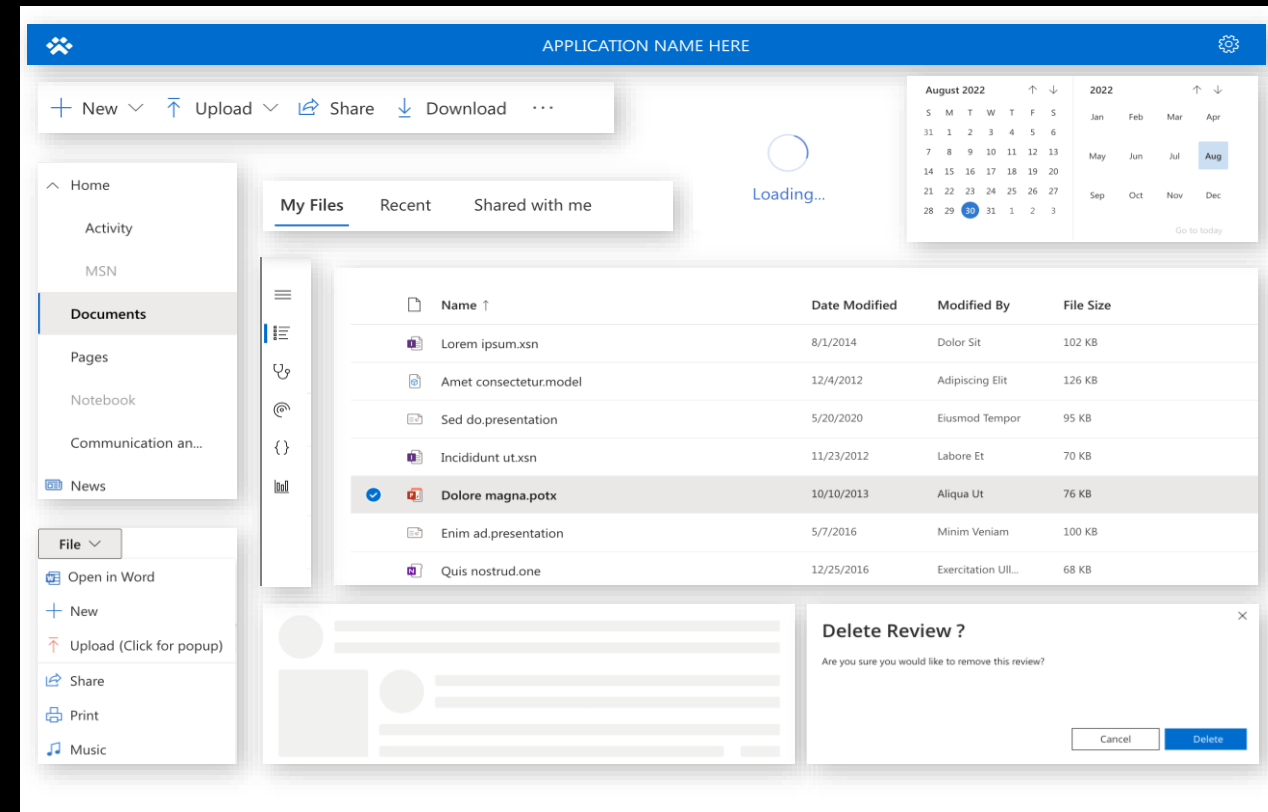


React and Fluent UI support

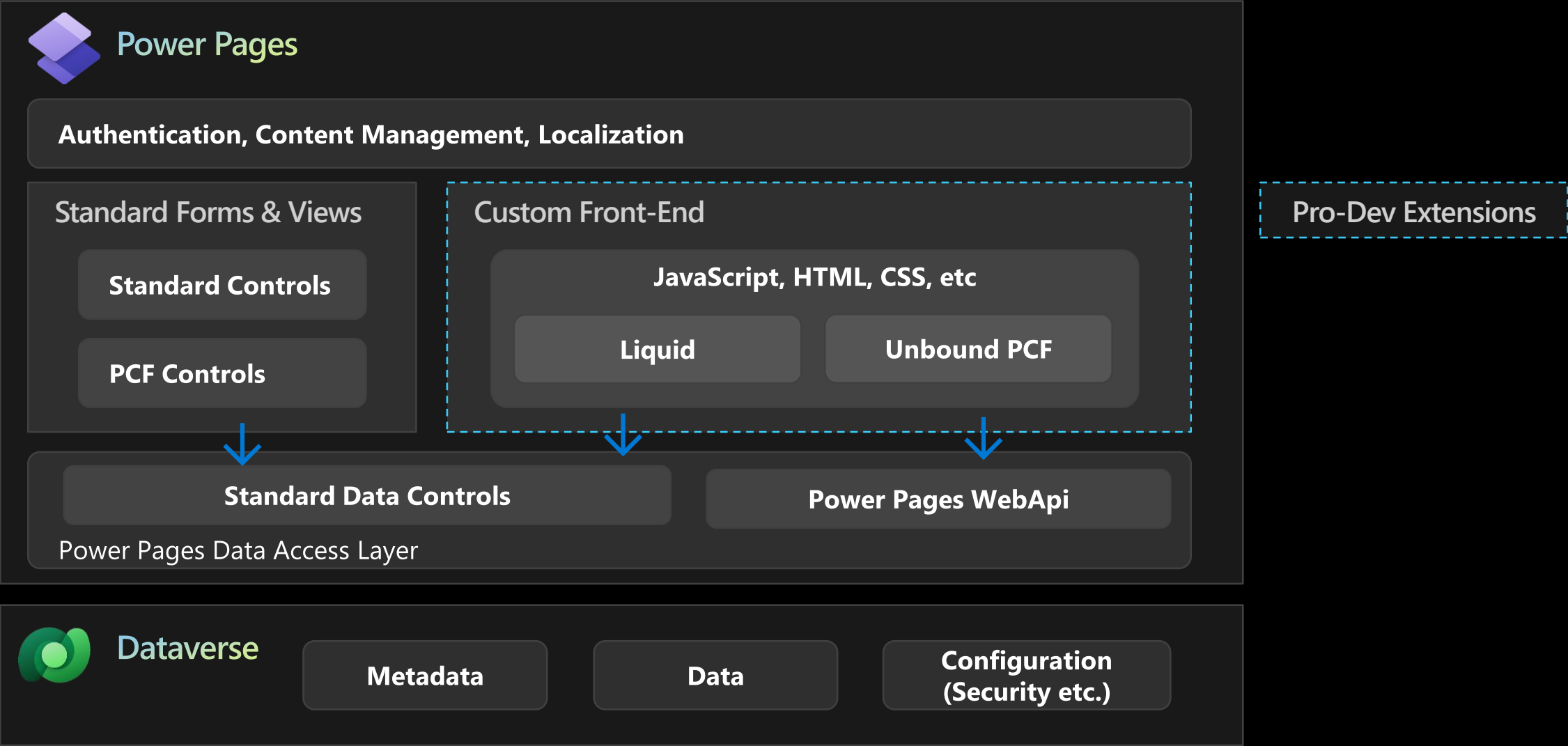
Creator Kit (PCF)



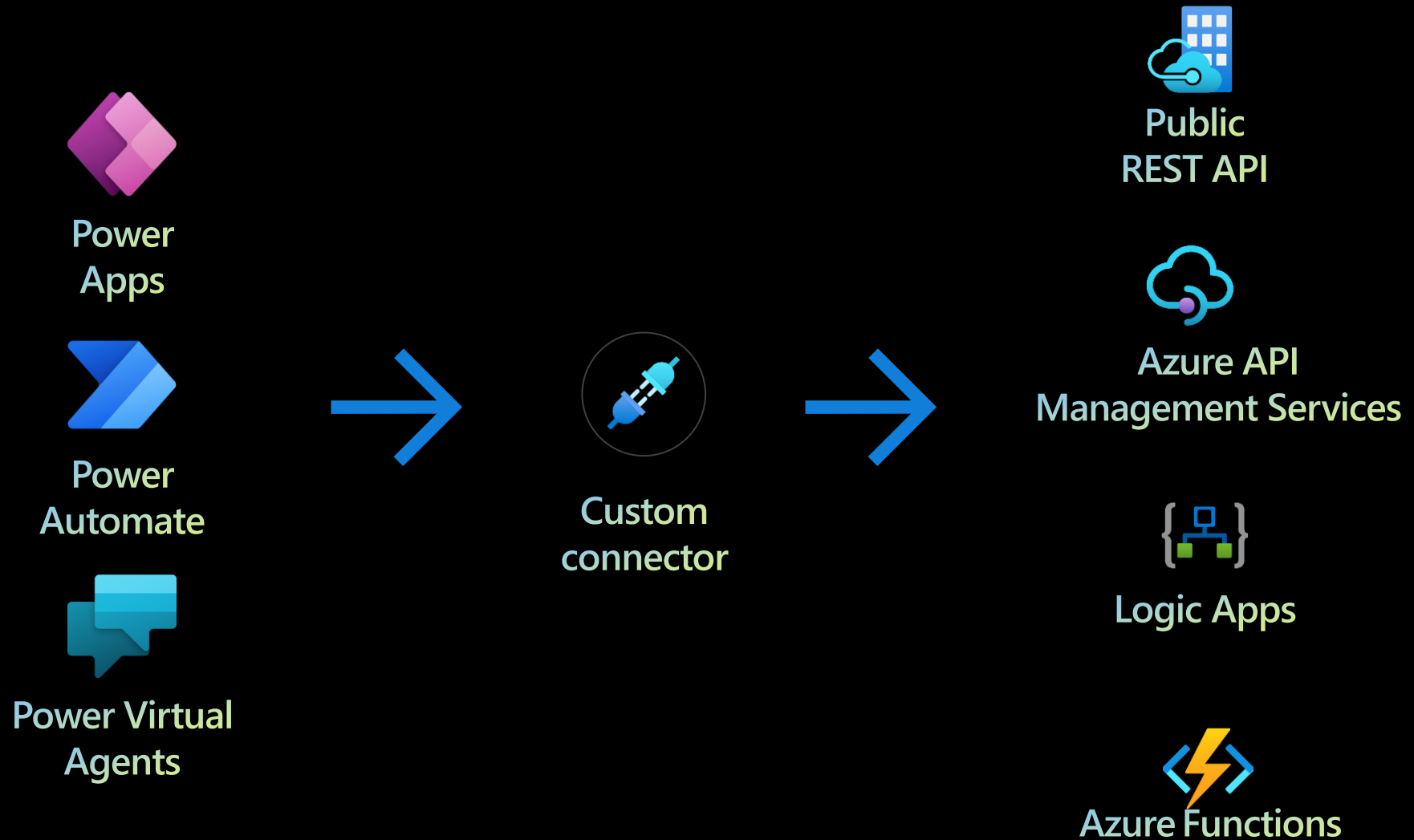
- ✓ Build performant apps that look and feel like modern model-driven apps
- ✓ Bridge until all native Fluent controls available
- ✓ Accessible & Responsive
- ✓ Open Source - Demonstrates Pro-Code Best practices



Power Pages Extensions



Custom Connectors



Demo

Visual Studio Custom Connector Authoring and Debugging

Where can I go to learn more?

Where can I go to learn more?

- aka.ms/pp/dev – Microsoft Power Platform Developer Docs
- aka.ms/pac/admin – Power Platform CLI
- aka.ms/connectors – Power Platform Custom Connector Docs
- aka.ms/lowcode-february/challenge – Cloud Skills Challenge
- aka.ms/lowcode-february – Low Code February
- aka.ms/virtualtables – Virtual Tables
- aka.ms/creatorkit – Examples of Fluent UI PCF components

Get started for free!

aka.ms/lowcode-february/devplan