

Transitioning from Pro to Low Code with Azure Logic Apps

Leverage the knowledge, skills and patterns of a pro code developer in the low code environment of Azure Logic Apps

Wherever business takes you

MNPdigital.ca



Gold
Microsoft Partner



Agenda for today

- | | | |
|----|---|--------|
| 1. | Introduction & Goals | 3 min |
| 2. | Low code trend in software dev | 8 min |
| 3. | How do pro code developers fit into the new landscape | 5 min |
| 4. | Azure Logic Apps as our low code tool | 7 min |
| 5. | How do pro code developer skills translate to azure logic apps? | 30 min |
| 7. | Questions | 5 min |
| 8. | Final Thoughts | 2 min |

Introduction



Cameron McKay

Solution Architect

Cameron.McKay@mnp.ca



Cameron McKay is a Microsoft certified Cloud Developer/Architect with a background in web technology and business analysis; he has blended his expertise in these areas to deliver a variety of enterprise web and cloud applications using the Microsoft technology stack.

Cameron has proven experience leading teams and delivering web and cloud solutions throughout the full software development lifecycle.

Cameron is an advocate for defensive programming, Microsoft Azure development and integrating security into the software development lifecycle.

Cameron spends his off-hours as a tabletop board game enthusiast, cooking up delicious meals, watching anime and training his husky dogs. Cameron is always up for any tabletop game; he especially enjoys strategy games such as Settlers of Catan and Go (Baduk).

Goals for today

Leave you with two ideas and impressions...

01

Understand

You already have the skills to leverage low code as part of your problem-solving toolkit.

02

Implement

Provide you with a jumpstart to use what you already know about software development to build low code solutions.

What do these have in common?

Sewing Machine

Assembly Line

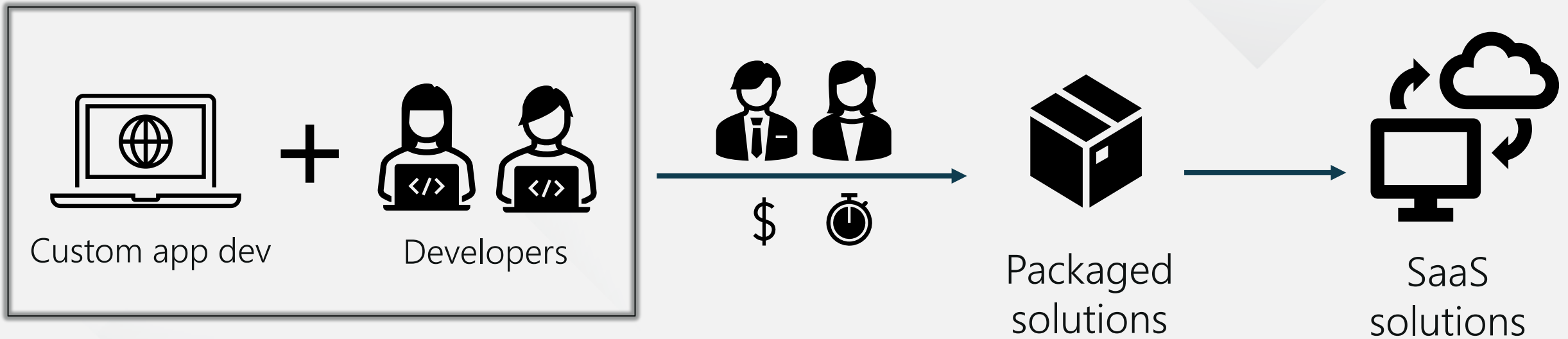
Printing Press



Our goals as pro code developers

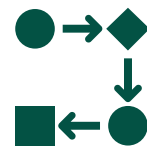
1. Solving complex problems
2. Architect and build reliable and maintainable code
3. Deliver as much value as possible to business and customers by shipping features

What is the trend in software development?





New problems to solve



Align with 80% of business workflow.
What about remaining 20%?

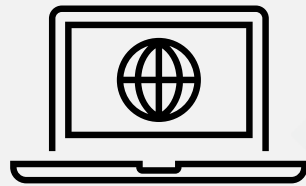


Automating interactions between
systems

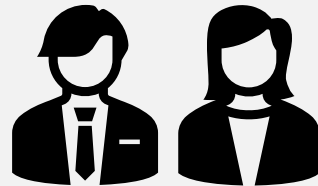


Need a single source of reporting for
decision making

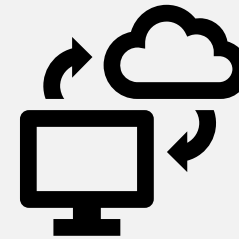
What is the trend in software development?



Low Code



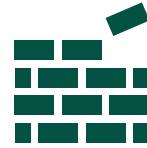
Citizen Developers



SaaS solutions



Citizen developers



Personal automations and share with colleagues



Low complexity automations



Gain knowledge of what it takes to build automations



There is no game of telephone



Pro code developers



Automations for entire departments and enterprise

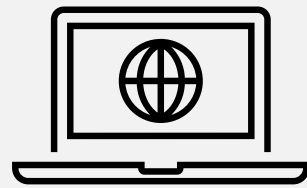


Complex automations with a lot of moving parts



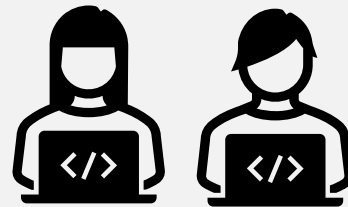
Consolidating data from many systems into a single source of reporting for decision making

What is the trend in software development?



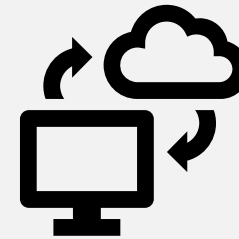
Pro code

+



Pro Code Developers

+



SaaS solutions



Let's solve these problems with only pro code



Cost of writing custom apps using only pro code is expensive



Time required to implement apps using only pro code is high



Our business will have difficulty keeping up with competitors who are leveraging low code



Customers will seek out companies that deliver the same quality for cheaper and less time

Implement an HTTP Client Wrapper

Pro Code	Low Code
Write HTTP Client Wrapper	Leverage out of the box HTTP action
Unit Test HTTP Client Wrapper	
Configure HTTP Client Wrapper	

Implement an SMTP Client Wrapper

Pro Code	Low Code
Write SMTP Client Wrapper	Leverage out of the box SMTP action
Unit Test SMTP Client Wrapper	
Configure SMTP Client Wrapper	

Implement an Azure Storage Wrapper

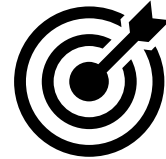
Pro Code	Low Code
Write Azure Storage Client Wrapper	Leverage out of the box Azure Storage action
Unit Test Azure Storage Client Wrapper	
Configure Azure Storage Client Wrapper	

Implement an Azure Key Vault Wrapper

Pro Code	Low Code
Write AKV Client Wrapper	Leverage out of the box AKV action
Unit Test AKV Client Wrapper	
Configure AKV Client Wrapper	



What if you could...



Leverage your skills as a pro code developer in a low code environment



Focus on design and architecture to integrate pro and low code together



Transfer saved time to testing and documentation

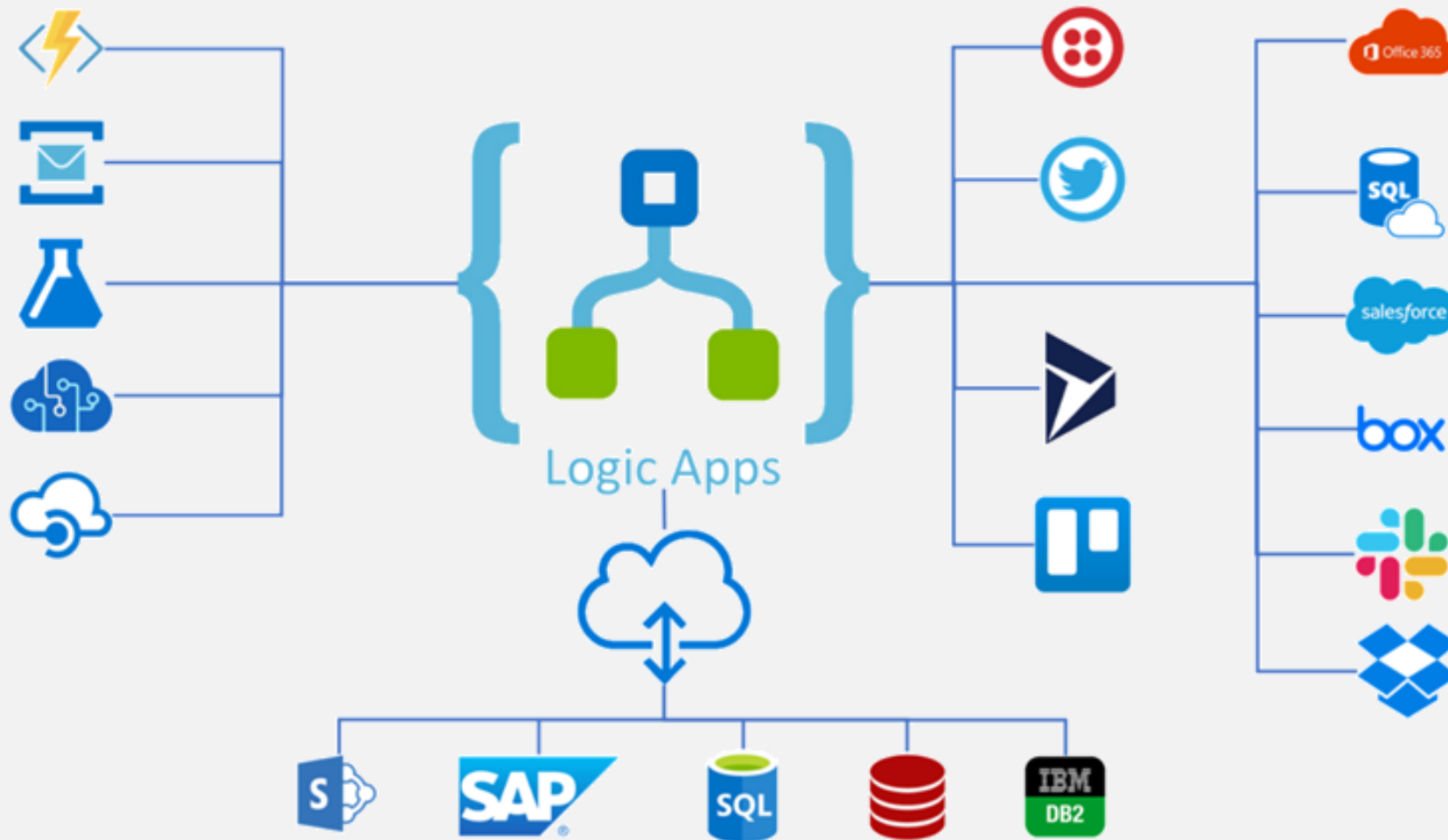


Deliver business value fast through pre-built connectors

Azure Logic Apps



Azure Logic Apps - Connectors { }



Connectors

Connectors in Logic Apps are categorized based on price for use



Built-in

The basics required to build an automation (for loops, if statements, variable declaration, etc.)



Managed - Standard

Pre-build connector for various SaaS solutions such as Business Central, Azure Storage, etc.



Managed - Enterprise

For well-defined industry protocols like EDI and sophisticated SaaS solutions like SAP



Resource Types



Consumption



Standard

Consumption

What do we get with consumption logic apps?



Multi-tenant

Logic apps across Microsoft Entra tenants share the same processing (compute), storage, network, etc.



Pricing

First 4,000 built-in executions are free. Connectors have a cost per execution.



One Workflow

One workflow per logic app.

Standard

What do we get with standard logic apps?



Single-tenant

Logic apps across Microsoft Entra tenants share the same processing (compute), storage, network, etc.



Pricing

Pay a set monthly cost for the underlying infrastructure to run built-in connectors.

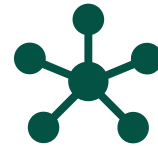


Many workflows

Many workflows deployed to a single logic app



Why does Standard hit the mark?



Virtual network Integration



Predictable pricing with ASP

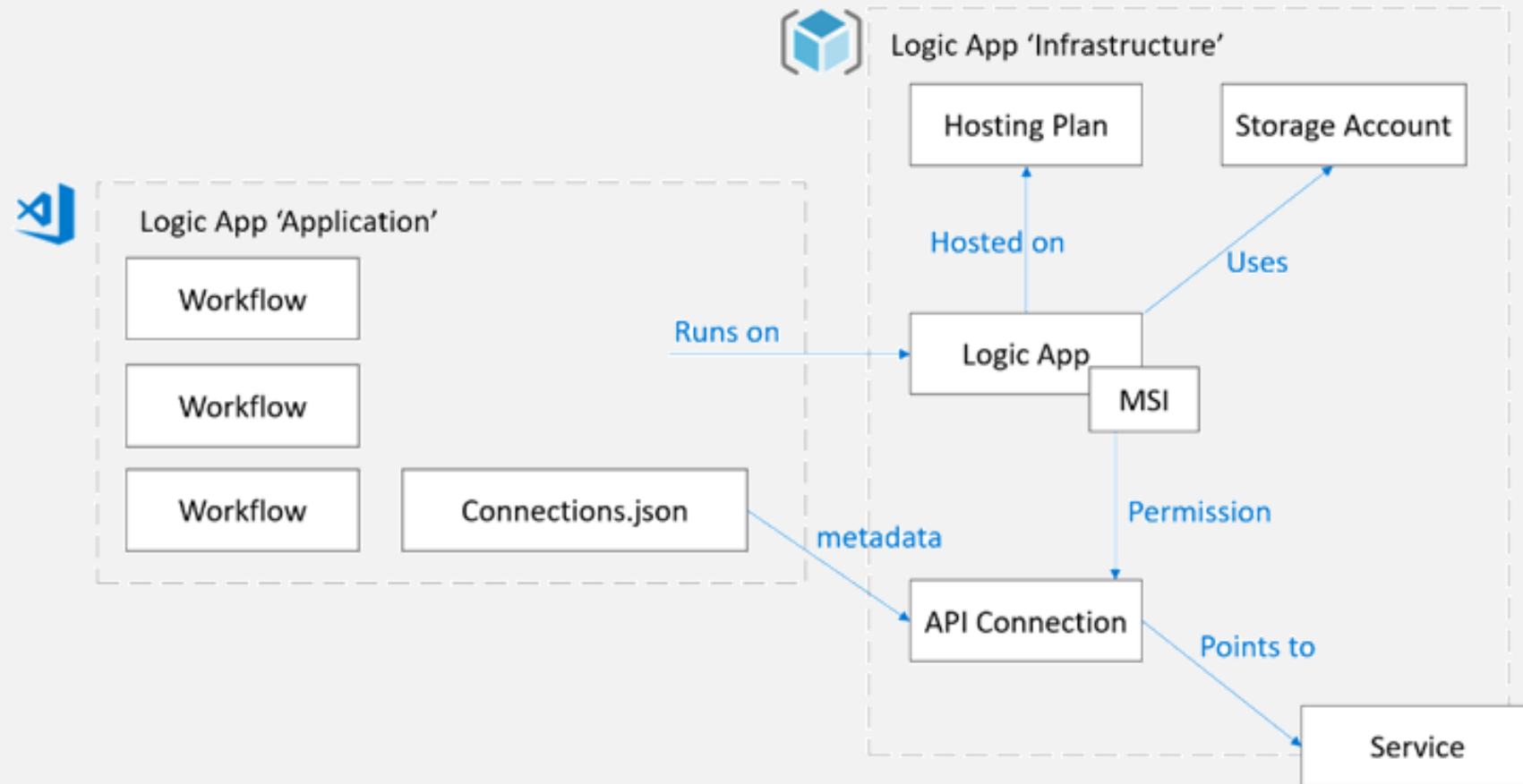


Multiple workflows

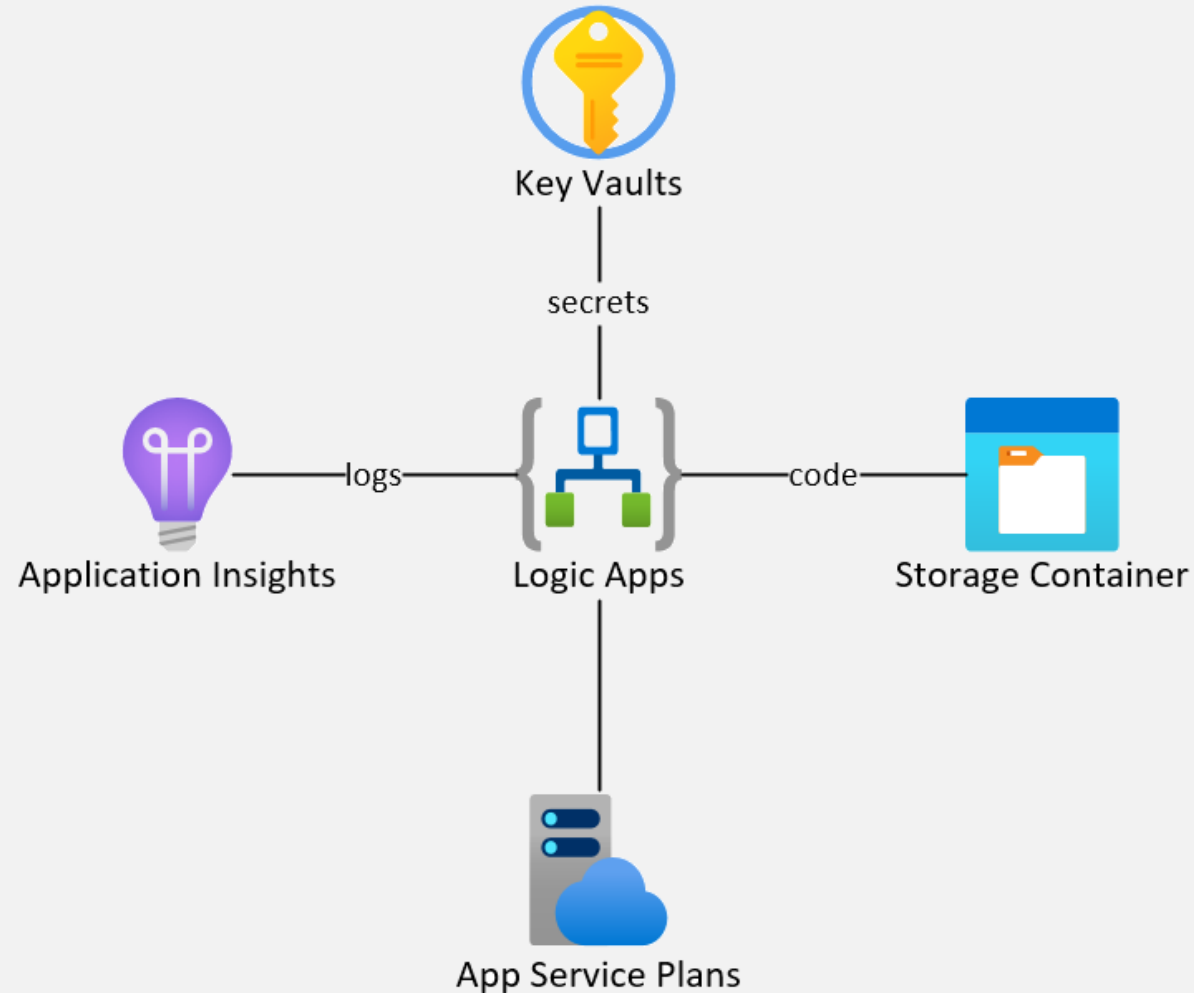


Hybrid pro and low code

Azure Logic Apps – Structure



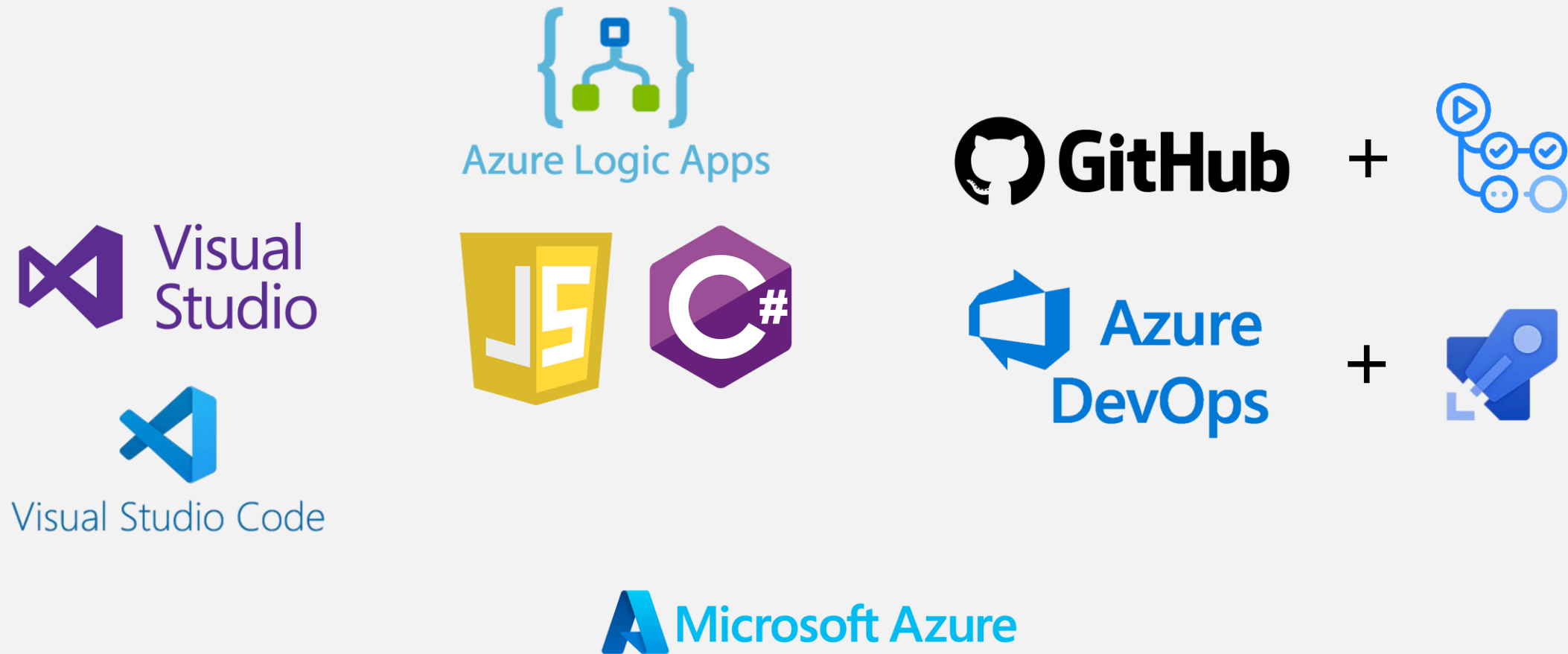
Azure Logic Apps – Infrastructure



Azure Logic Apps – Development Stack



Azure Logic Apps – Development Stack



Let's develop in low code!

Applying pro code patterns in low code

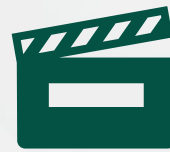
Azure Logic Apps - Basics

Let's get the lay of the land of writing azure logic apps



Where to start?!

Creating an Azure Logic App



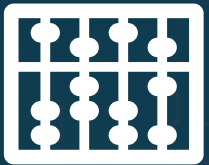
Triggers and actions

What's available to start working on a simple flow.



Responses

Responding to a request with an Azure Logic App



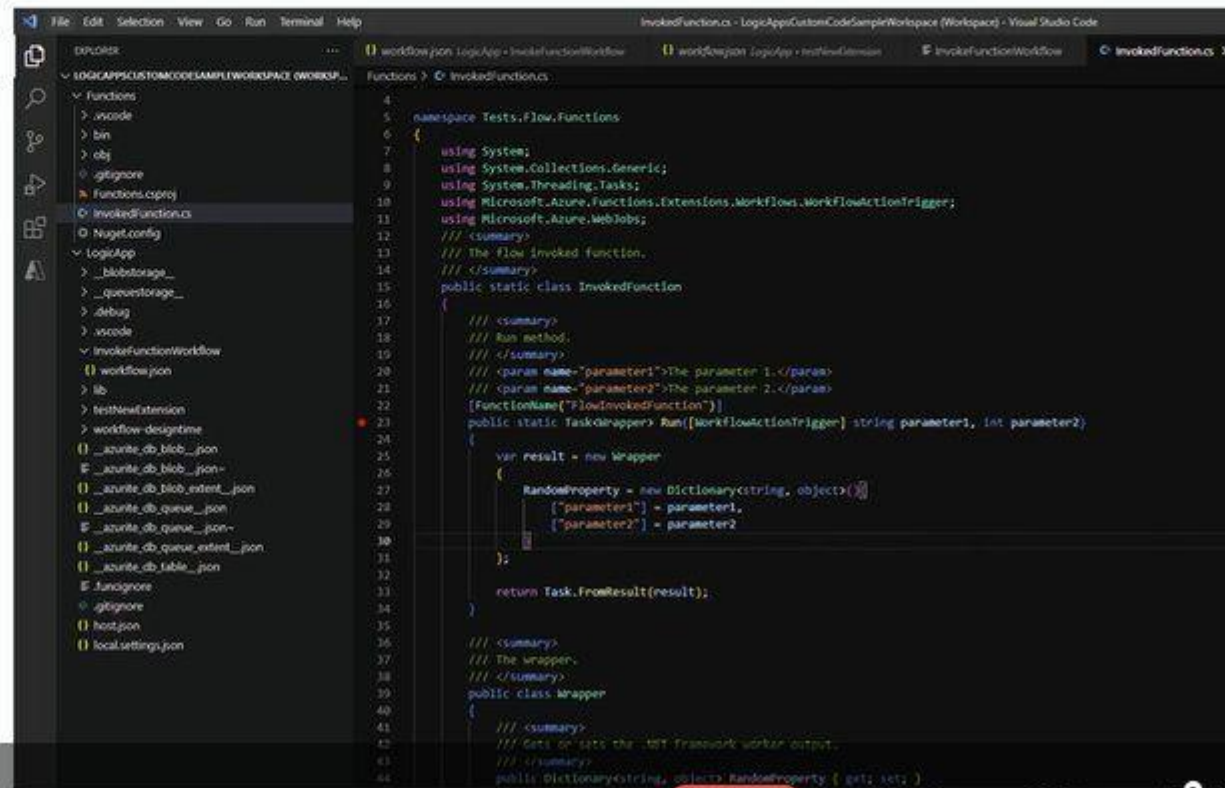
Calculations and Controls

- Compose action
- In-line code execution
- Looping
- Conditionals
- Parallelism



.NET Framework Custom Code

- Call .NET Framework code from Built-in action found in Azure Logic Apps (Standard)
- BizTalk migration scenarios unlocked
- New project structure including Workspace, Code and Workflow projects
- Local debugging for both workflows and code
- Full support for dynamic schema input/outputs including complex types



Azure Logic Apps - Scopes

They are great! We should always use them!



They are your best friend!

Be their best friend and you'll have a good time developing logic apps.



Organize your code

Group your actions into scopes to organize items. Think of #regions in C#.



SRP

Treat scopes like functions. Apply single responsibility principle to the actions within them.



Error Handling and Testing

- Try-Catch-Finally
- Terminate Action



Input Validation

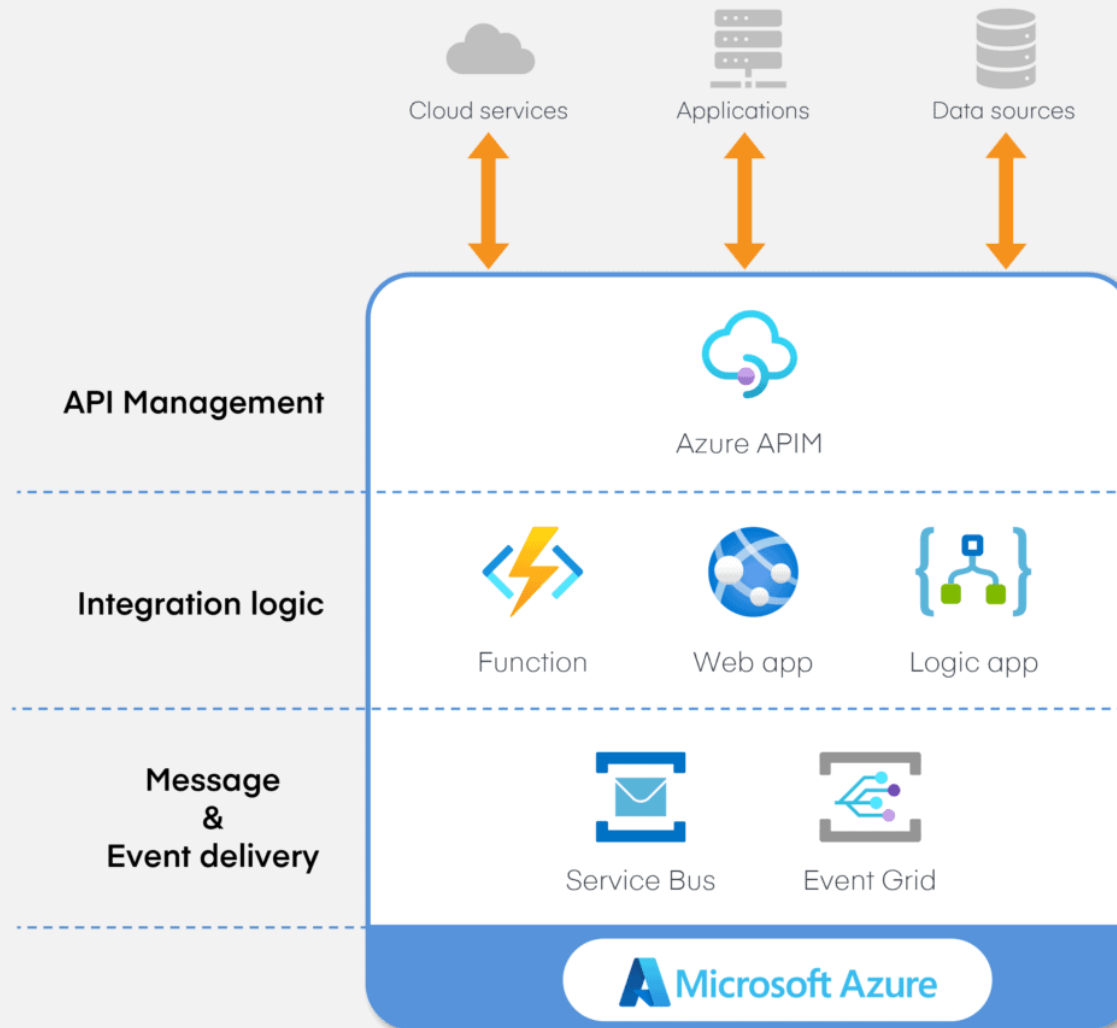
- Json Schema!



Code organization

- Leverage in-line code to do more complex processing (e.g. regex)
- Comments
- Naming conventions
- Child workflows
- Group actions using scopes

Azure Integration Services



Questions



Logic Apps, Aviators Community



[Newsletter - Azure Integration Services Blog](#)

Aviators Community – Michael Stephenson



[Azure Integration Services - When to use what](#)

[YouTube Channel](#)

[Logic Apps Testing YouTube Playlist](#)

[Logic App Testing Documentation](#)



Aviators Community – Kent Weare



[Azure Logic Apps Tips and Tricks](#)

[YouTube Channel](#)

[Getting Started with Azure Logic Apps Standard](#)

[Automated Testing for Azure Logic Apps](#)



Aviators Community – Kent Weare

1:00 - 2:00

Low code? Pro code? How about both using Azure Logic Apps

Kent Weare

📍 Umbria

Azure Logic Apps is a powerful platform for building automated workflows that can run anywhere. It offers a low code experience that enables you to create complex integrations using a graphical designer and a rich set of connectors. But what if you need to extend your workflows with custom logic that is not available out of the box? How can you leverage your existing .NET Framework skills and investments to enhance your Logic Apps solutions?

In this session, you will learn about the new .NET Framework custom code feature for Azure Logic Apps (Standard), which allows you to call compiled .NET Framework code from a built-in action in your workflow. You will see how this feature provides a no-cliffs extensibility capability that gives you the flexibility and control to solve the toughest integration problems.

By the end of this session, you will have a better understanding of how to use low code and pro code together to create powerful and scalable integration solutions using Azure Logic Apps. You will also learn how to take advantage of the latest features and updates in Logic Apps (Standard) to improve your productivity and performance.



Aviators Community – Sandro Perreira



[Logic Apps Best Practices](#)

[Professional Blog](#)

[YouTube Channel](#)

[Logic Apps Data Mapper](#)



Low code platforms reduce app development time by **50 %** - Redhat 2018

Low code can speed up application development by a factor of **10** - Forrester 2018

70% of new business applications will use low code tech by 2025 – Gartner 2021

Final Thoughts

Leave you with two ideas and impressions...

01

Understand

You already have the skills to leverage low code as part of your problem-solving toolkit.

02

Implement

Provide you with a jumpstart to use what you already know about software development to build low code solutions.

Contact

Cameron McKay

Solution Architect

Cameron.McKay@mnpc.ca

