



Introduction to Azure Integration Services

Harold Campos

Principal Product Manager
Azure Logic Apps, BizTalk and HIS



@hcampous



hcampous@microsoft.com



<https://www.linkedin.com/in/hcampous>



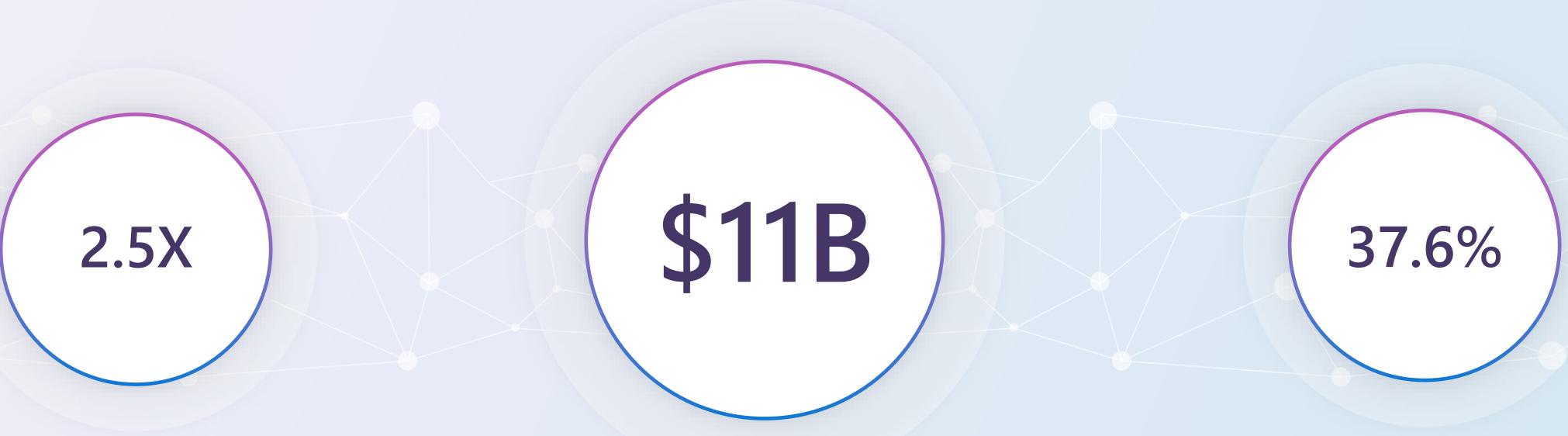
<https://www.youtube.com/@hcampous>



EXCELLENCE IN RECRUITMENT



Integration is the fast-growing market



2.5X

\$11B

37.6%

iPaaS growth compared to
Enterprise Software

iPaaS market forecast for
2026

Year-over-year growth

1: Gartner, Market Share: Application Infrastructure and Middleware, Worldwide, 2022 (May 2023)

2: Gartner, Software Market View, 2021-2022 (Sept 2022)

3: Gartner, Critical Capabilities for Enterprise Integration Platform as a Service (Sept 2023)

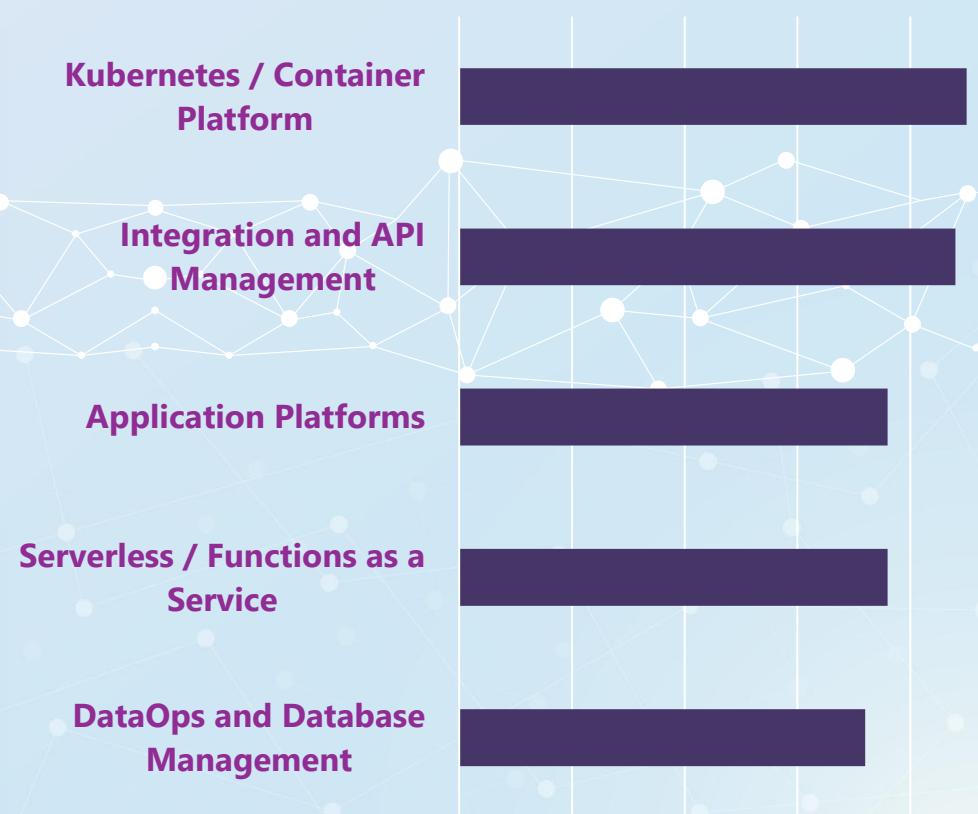
Organizations significantly invest into Integration and API Management



Enterprises will leverage at least two different integration tools¹



Of time and cost of building a digital platform will be towards integration²



1. Choose the Best Integration Tool for Your Needs Based on the Three Basic Patterns of Integration, Gartner
2. Use a Hybrid Integration Approach to empower digital transformation, Gartner
3. PaaS Spending Trend Survey, IDC



Of global respondents say investments in APIs will stay the same or increase over the next 12 months, up from 89% last year¹

Of respondents say most of their organization's development effort was spent working with APIs¹

1. 2023 State of the API Report, Postman, 2023

Explosive app growth

Diverse set of applications packaged, SaaS, custom-built to meet the customer needs



Number of apps expected to be created by 2025 is more than the total in the last 40 years

IDC: Updates Its Future of Digital Innovation Framework. Oct 2022

App proliferation creates challenges

The difficulty at SPAR was that we didn't have an overview of where the data was heading to or where the errors were popping up. So, we didn't have insights into our system. It was a **black box of different data streams**.



– Joran van den Brink
IT Integration Manager
SPAR NL

We saw the potential for collaboration with traditional fund managers like us, and so it became **essential to have APIs that would allow us to tap into new distribution channels** and form those partnerships easily. This made API adoption almost a no-brainer.



– Allen Woo Kai Yeen
Chief Technology Officer
AHAM Capital

We had a fairly **large number of integration technologies grown organically**. With the increased demand for integrations, especially the **growing number of SaaS** solutions, we wanted to **standardize and remove complexity** in order to reduce costs and improve ease of use for future development.

– Carel Nederveen
Enterprise Technology Architect
KPMG Netherlands

The company's success may have started with lab equipment, but in compliance with all data protection guidelines and transparency towards our customers, it is important to BÜCHI that we can create **new business models that use all our data and aggregate best cases for our customers**.



– Reto Hossmann
Global Head of ICT
BÜCHI Labortechnik AG

The lumber and building materials industry tends to be **heavily invested in on-premises legacy**. We wanted improvements in line with our go-forward strategy on architecture, which is cloud first and allows for subscription-based licensing to scale with US LBM's quickly growing business. It was important that the new middleware solution aligned with these requirements.



– Greg Bossert
Vice President of Enterprise Applications
US LBM

The essentials of getting everything to play together

1

Integration



Integration with Legacy and New Applications, Vendors, Partners

2

Composition



Composite services or APIs, which allow to build new functionalities by composing existing services, internal and external

3

Automation



Enables multi-step processes, where independent apps collaborate to execute a business process smoothly

Deriving Intelligence

With these foundational elements in place, businesses can leverage their data to derive intelligence, driving informed decision-making and innovation.

Integration and API Management are Essential for AI Success

Importance of Integration

Facilitate the seamless flow of data between systems, applications, and AI models.



Role of API Management

Acts as the bridge between AI models and external applications, allowing for easy access to AI capabilities.



Azure Integration Services



Comprehensive flexible platform
enables the creation of tailored
solutions for unique
requirements



Logic Apps

Cloud-based workflow tool



API Management

Full API lifecycle management



Event Grid

Event routing service



API Center

Centralized API discovery hub



Service Bus

Reliable cloud message delivery



Functions

Event-driven serverless code



Data Factory

Data movement & ETL tool

Azure Integration Environment

Unified experience for managing and monitoring Azure Integration Services resources

The Azure Integration Services Advantage



Unlock Seamless Data Connectivity

800+ out-of-the-box connectors

Industry protocols and connectors X12, EDIFACT, HL7, SWIFT MX, MQTT



Build Composable Enterprise

API lifecycle management

Unified API management across hybrid and multi-cloud

Composition through pro-code and low-code



Enable all Developers

Low-code visual workflow designer

Pro Developer Experience

Custom connectors capabilities

Custom code extensibility

Built-in Security

- Logic Apps
- API Management
- Service Bus
- More...



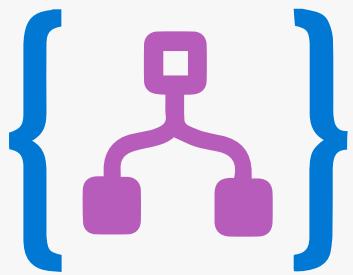
Fortify Security and Compliance

Integration w/ Microsoft, Azure Security Products



Compliance

- PCI DSS
- SOC 1, 2, 3
- HIPAA BAA
- More...



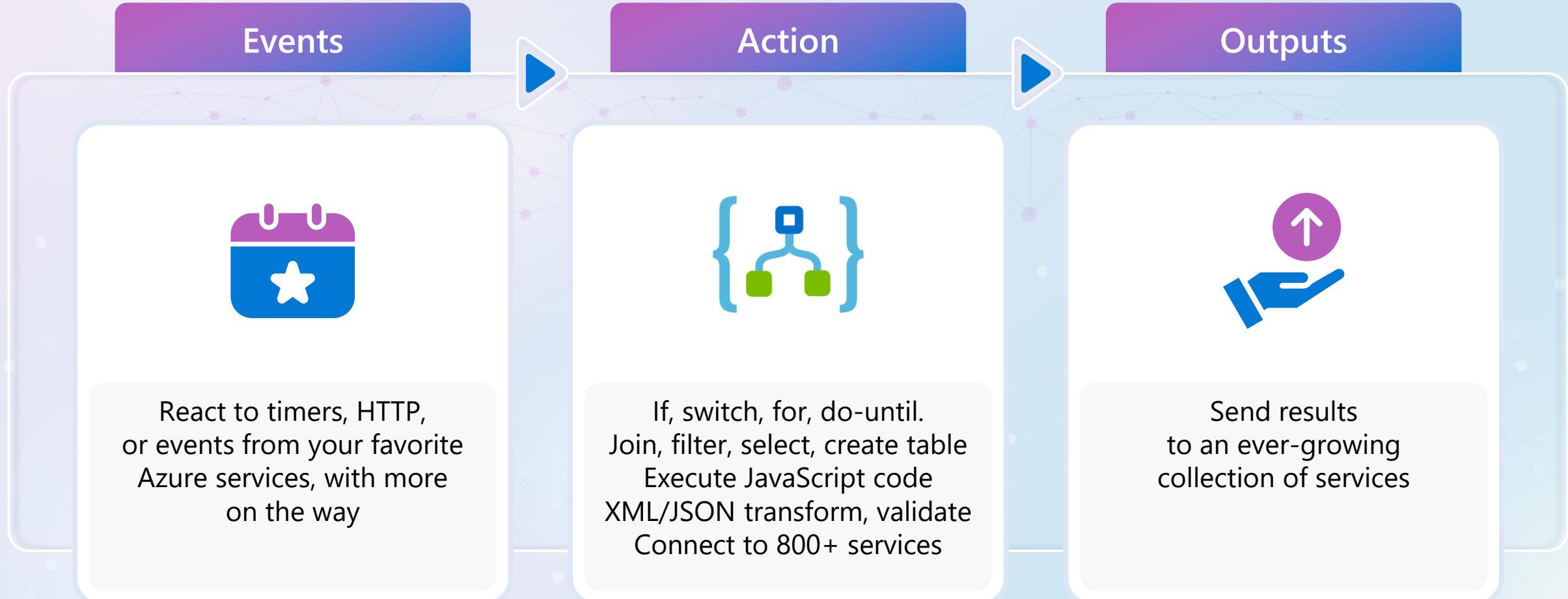
Azure Logic Apps

Cloud-based service for automating workflows, integrating business processes, applications and data across hybrid environments in a visual manner.



- 1 Develop/test   
- 2 Source control integration with broad set of popular platforms   
- 3 Set breakpoints for each action & use debugger to review local variables
- 4 Push to cloud directly or CI/CD for end-to-end validation

Azure Logic Apps Programming Model



Runtime Options Flexibility

Runs on
Tenancy
Connectivity
Billing
Extensibility
DevOps
Scale & Performance

 Consumption

Azure
Multi-tenant
On-premises data gateway
Per action, pay-as-you-go
Codeless connectors
ARM

300,000 executions per 5-minute.
Single instance can run as long as 365 days.
Almost 20-billion executions per day.

 Standard

Azure, local, Arc, Docker, and Kubernetes
Single-tenant
VNet, private endpoints
Per hosting plan + storage
Codeless and codeful connectors + custom code + visual data mapper
Infrastructure as code

Stateless workflow for low latency, high throughput workloads.
Built-in connectors run alongside runtime with low latency.
Scale between 210 (1x of WS1) to 84,000 (100x of WS3) ACUs.

Stateful workflow

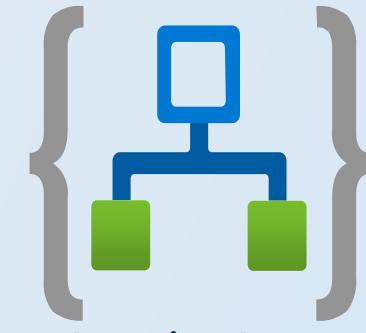
- As the workflow executes, input & output (aka. state, therefore 'stateful') from each steps are preserved to storage
- High resiliency
- In the event of infrastructure failure during workflow execution, the run can be reconstructed once restored, and continue execution to completion
- Slower than stateless workflow due to storage operations
- Run history is available
- When to use: Optimized for high reliability, ideal for process business transitional data
- Stateful workflow is the default behavior for consumption Logic Apps

Stateless workflow

- As the workflow executes, input & output (aka. state, therefore 'stateful') from each steps are stored in memory
- High throughput
- Workflows are treated as atomic transaction, so it will execute or fail as a unit
- Faster than stateful workflow as state is kept in memory
- No run history by default, but can enable debug option when needed
- When to use: Optimized for low latency, ideal for request-response and processing IoT events
- Selected operations not available: Webhook

Demo

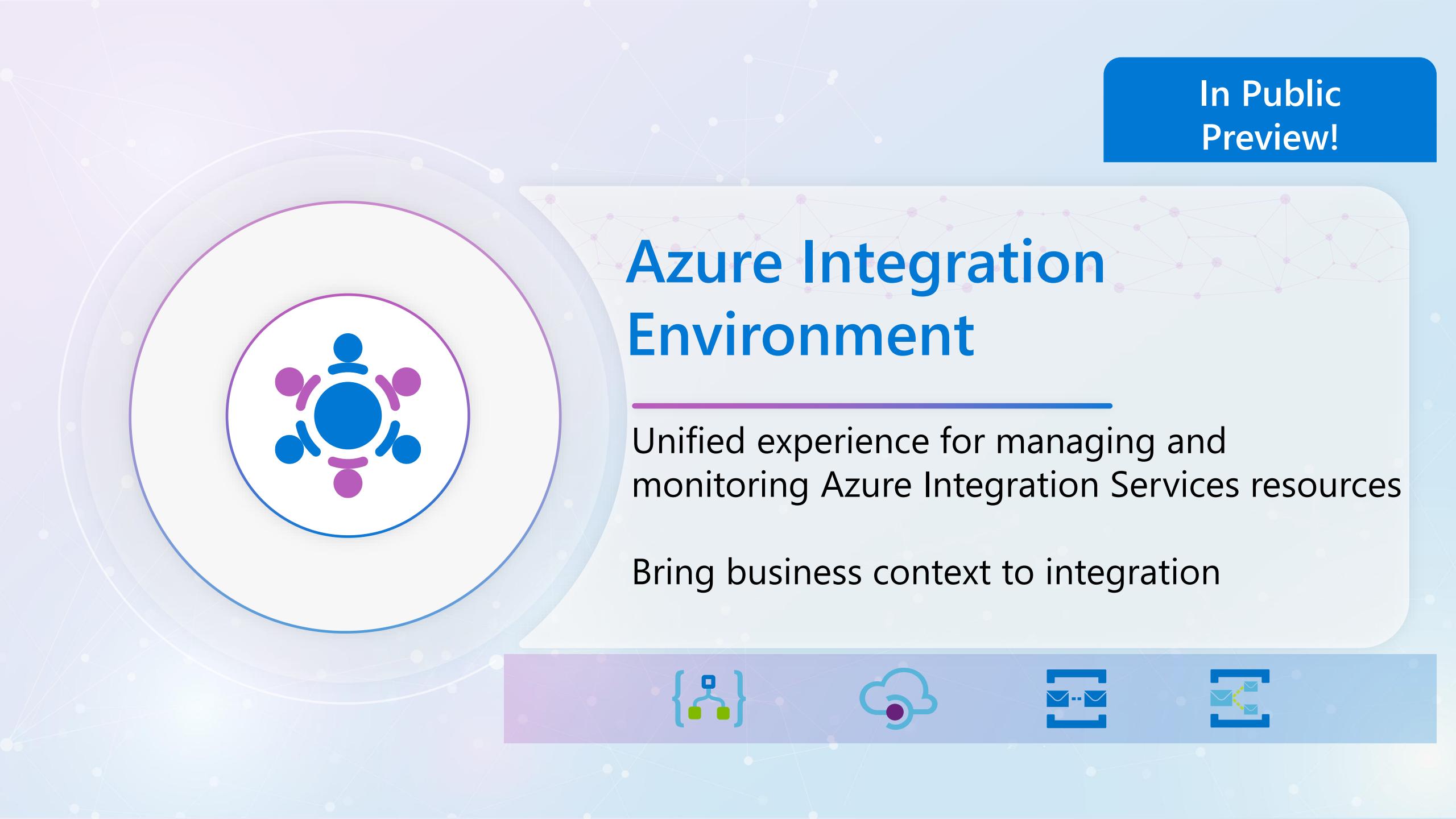
Creating and configuring
an Azure Logic App and a
simple workflow.



Logic App
Workflow

What's New





In Public
Preview!

Azure Integration Environment

Unified experience for managing and monitoring Azure Integration Services resources

Bring business context to integration



Streamline operations using Integration Environment



Flexibility in grouping resources based on organizational standards and principles

Group by landscapes

- Development
- Test
- Production

Group by business units

- Finance
- Marketing
- Operations



Further simplify management by breaking down environments into logical groupings

- Payroll
- Order processing
- Employee onboarding
- ...and more!

End-to-end AIS Monitoring Public Preview

Analyze application workloads, troubleshoot failures, identify anomalies without context switching

Built on Azure App Insights for telemetry data and Workbooks for visualizations

The screenshots illustrate the end-to-end monitoring capabilities of Application Insights across different Azure services:

- Top Screenshot (APIM Tab):** Shows the main Insights blade for the 'customer-service-poweroutage' application. It includes navigation links for Home, customer-service-poweroutage, and Insights. The Insights section displays a search bar, refresh controls, and a dropdown for ApplicationInsights (set to 'Appinsights-Integrate2023'). Below this are sections for Resources, Application monitoring (with sub-options for Insights, Business process tracking, Logic Apps, Service Bus, and APIM), and Business processes.
- Middle Screenshot (Service Bus Tab):** Shows the Service Bus blade for the same application. It includes navigation links for Home, customer-service-poweroutage, and Insights. The Insights section displays a search bar, refresh controls, and a dropdown for ApplicationInsights (set to 'Appinsights-Integrate2023'). Below this are sections for Service Bus (set to 'apseth-ns'), Topics and queues (All), and tabs for Overview, Requests, and Messages. The Requests tab is currently selected.
- Bottom Screenshot (Operations Tab):** Shows the APIM blade for the same application. It includes navigation links for Home, customer-service-poweroutage, and Insights. The Insights section displays a search bar, refresh controls, and a dropdown for ApplicationInsights (set to 'Appinsights-Integrate2023'). Below this are sections for Timeline, APIs, and Operations. The Operations tab is currently selected. The main content area displays a table of API usage statistics:

API	OperationName	Successful Requests	Failed Requests	Unauthorized Requests	Other Requests	Total Requests	AverageDuration
CPL-APIManagement/apis/work-management-mean-tim...	None	6	0	0	186	192	0.33465416666666664
CPL-APIManagement/apis/work-management-mean-tim...	powerrestore	18	0	0	0	18	964.85165555555556

Business Process Tracking New Capabilities

Business process stage status

Enhanced Token Picker support

The screenshot shows the Microsoft Azure Business Process Tracking interface. The main view displays two transactions: CASE-1173 and CASE-1172. CASE-1173 has 9 stages succeeded and 1 stage failed, starting at 6/6/2024, 10:10:47 PM. CASE-1172 has 10 stages succeeded and 0 stages failed, starting at 6/6/2024, 10:08:28 PM. The interface includes a sidebar with options like Overview, Access control (IAM), Resource visualizer, Settings, Locks, Business process tracking, Editor, and Transactions (which is selected).

This screenshot shows the same Azure Business Process Tracking interface as above, but with a detailed process flow diagram overlaid on the right side. The diagram illustrates a workflow starting with 'Ticket-Created' (green), followed by 'Work-Order-Created' (green), 'Work-Order-Dispatched' (green), 'Work-Order-Onsite' (green), and 'Work-Order-Complete' (green). From 'Work-Order-Complete', three parallel paths emerge: 'CRM-Ticket-Closed' (green), 'Inventory-Update' (green), and 'Safety-Check' (red). 'CRM-Ticket-Closed' leads to 'Customer-Communication' (green). 'Inventory-Update' and 'Safety-Check' converge before reaching 'Customer-Communication'. 'Customer-Communication' leads to 'Publish-To-LLM' (green). Timings are indicated between steps: 3s, 25s, 31s, 18s, <1s, 1s, <1s, and 3s.

Logic Apps Standard Hybrid Deployment Model

Deploy standard Logic Apps to customer managed infrastructure

Semi-connected scenarios

Control plane is hosted in Azure and accesses data plane through Azure ARC

Early Access Preview – June 2024

Public Preview – Autumn 2024

The image displays three separate windows of the Microsoft Azure Logic Apps interface:

- Connected Environment:** Shows the configuration for a Logic App named "cplsaftyincidents". It includes details like Resource group (ishank-acा-preview-rg), Status (Running), Location (North Central US (Stage)), Subscription (BTS4), and Application URL (https://cplsaftyincidents.ishank-acा-preview-rg.azurewebsites.net/api/cplsaftyincidents).
- Workflows:** Lists various workflows associated with the logic app, such as "cplprocessincidents", "createsql", "sample1", "sqlsample", "stateful1", "test1", "triggersql", and "xmlesample". All workflows are currently enabled.
- Run History:** Displays a history of recent runs for a workflow named "createsql". The table shows three successful runs on 6/6/2024, each taking between 2.19 and 2.47 seconds.

Logic Apps Testing Framework

The screenshot displays a development environment for testing Logic Apps, showing both the code editor and the logic app designer.

Code Editor (VS Code):

- File Explorer:** Shows the project structure under "LA-UNITTEST-DEMO (Workspace)" including "LogicApp", "Tests", and "Connections".
- Editor:** Displays "unitTestSample.cs" code, which includes imports for System, Newtonsoft.Json, and Microsoft.Azure.WebJobs, and defines a class with methods for handling workflow actions and assertions.
- Outline:** Shows no symbols found in the document.
- Timeline:** Shows the history of changes made to the file.

Logic App Designer:

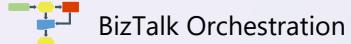
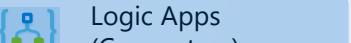
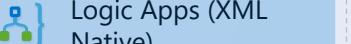
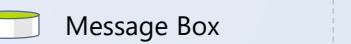
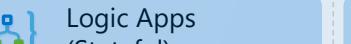
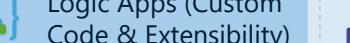
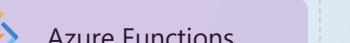
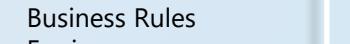
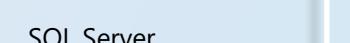
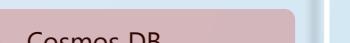
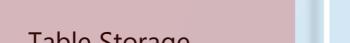
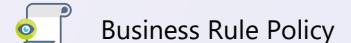
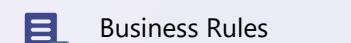
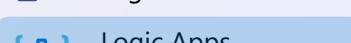
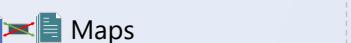
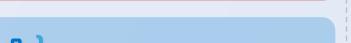
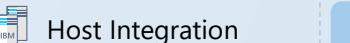
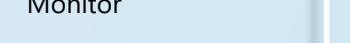
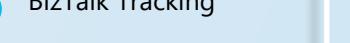
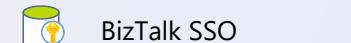
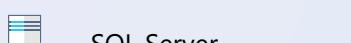
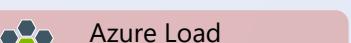
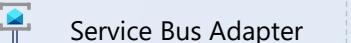
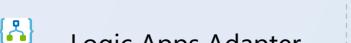
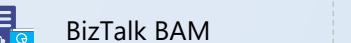
- Workflow:** A logic app workflow titled "When a HTTP request is received" is shown. It starts with an "Initialize variable" action, followed by an "HTTP" action (method: GET). This is followed by a "Parse JSON" action, which feeds into a "Switch" control. The "Switch" control has two cases: "Case received" (leading to an "Allow receive" action) and "Case processing" (leading to an "Allow processing" action). Both cases lead to a "Compose allowProcessing" action, which then leads to a "Compose response" action and finally a "Response" action.
- Mocked Results:** On the right, a "Mocked Results" panel shows the expected response for an HTTP request. It includes fields for OrderId (1234), OrderDate (2024-06-01), and OrderItems, which contains items like ItemId - 1 (12), Itemname - 1 (myitem1), Quantity - 1 (10), and Price - 1 (20.00). There is also a "TotalPrice" field set to 200.00.
- Services:** At the bottom right, there are links to "[Azure Table Service]", "[Azure Queue Service]", and "[Azure Blob Service]".

BizTalk Migration is gaining a momentum

- Business Rules Engine – [Public Preview](#)
- Native XML Support – [Private Preview](#)
- Architectural guide – GA
- BizTalk Maps Support – GA
- .Net Framework Custom Code – GA
- SWIFT – [Public Preview](#)
- Patterns Library (Templates) – [In Development](#)
- WCF/SOAP Support – [Analysis](#)
- JMS Support – [Analysis](#)
- MLLP/HL7 – [In Development](#)

BizTalk to Azure Logic Apps



Orchestrations	EAI	API/XML/SOAP	Durable Messaging	Helper Functions	Configurations
 BizTalk Orchestration 	 BizTalk Adapters   	 Logic Apps (XML Native) 	 Message Box 	 Custom Code  	    
Business Rules	B2B Integration	Data Transformation	Asynch Messaging	Mainframes and Midranges	Management & Monitoring
  	  	  	 	 	   
Security & Governance	Disaster Recovery	Hybrid	Business Users		
 	  	  	 		

Azure Messaging Platform [Integration Brokers](#)

Message Queuing

Service Bus - Reliable Cloud
Messaging and Hybrid Integration

Discrete Event Routing & PubSub

Event Grid - Event Handling and
Delivery at Cloud Scale

Device Messaging

Event Grid – Cloud MQTT broker
for D2C, C2D, D2D messaging

Event Streaming

Event Hubs - Receive millions of
events from distributed clients

Stream Processing

Stream Analytics – Process and
Route Events in real-time

Message Catalog

Event Hubs Schema Registry –
governance for message processing

Azure Messaging Continuous Innovation

- Kafka
- Geo Replication
- Emulators
- MQTT
- PubSub eventing
- SaaS streaming integration

Centralized API inventory with Azure API Center!



Azure API Center: Centralized API Inventory for Seamless Discovery, Consumption, and Governance



Centralized API Inventory Management

Build and maintain an up-to-date inventory of all organizational APIs

Provides visibility into the entire API landscape



Streamlined API Governance

Govern org-wide APIs effectively, ensuring compliance with standards and policies

Centralized control over API lifecycle management, security, and compliance.

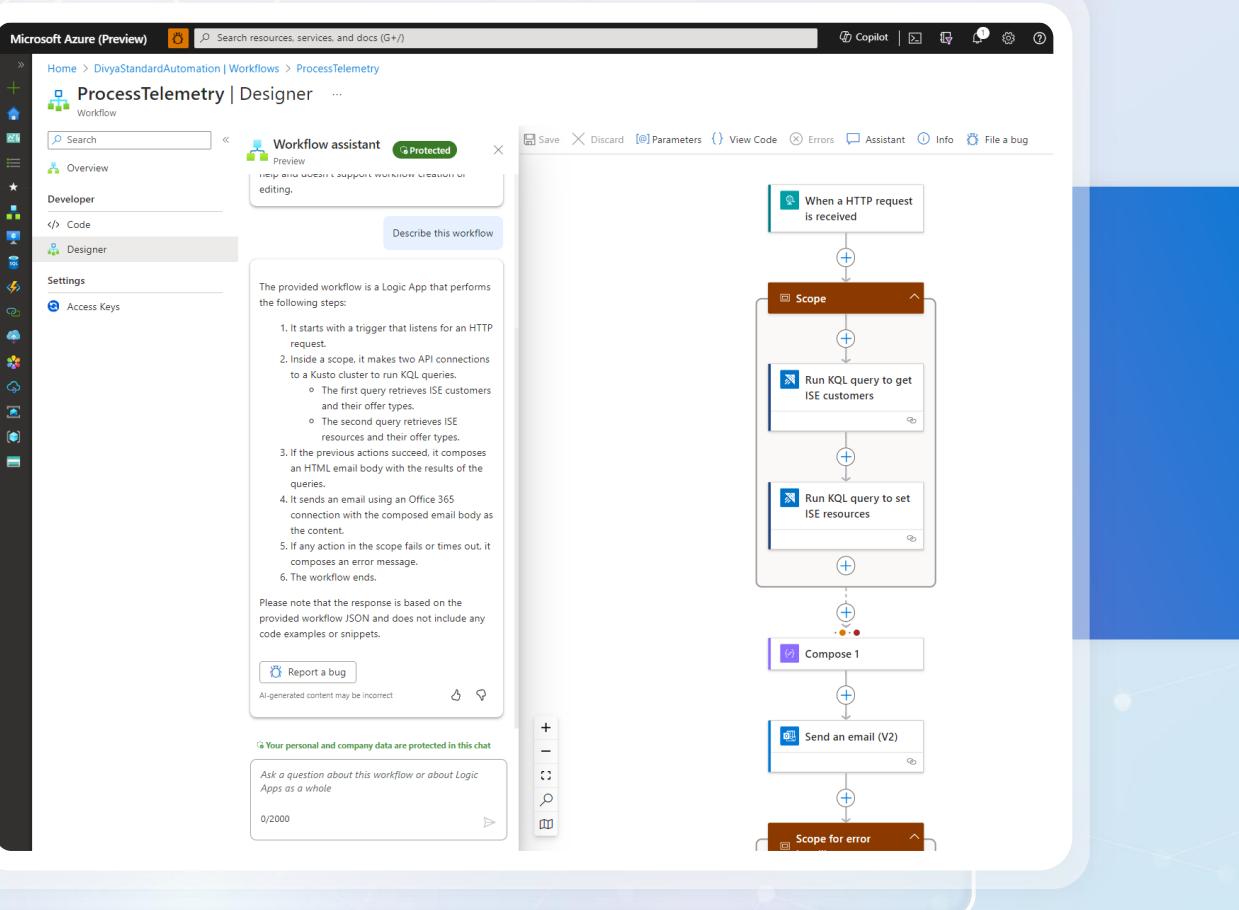


Simplified API Discovery and Consumption

Easy discovery of APIs within the organization's inventory

Accelerates API consumption

Streamline integration development with AI



Chat Interface

Seamlessly integrated within the Logic Apps designer

Powered by AI

Leveraging Azure OpenAI and ChatGPT technologies

Answer anything about



Describing Workflows

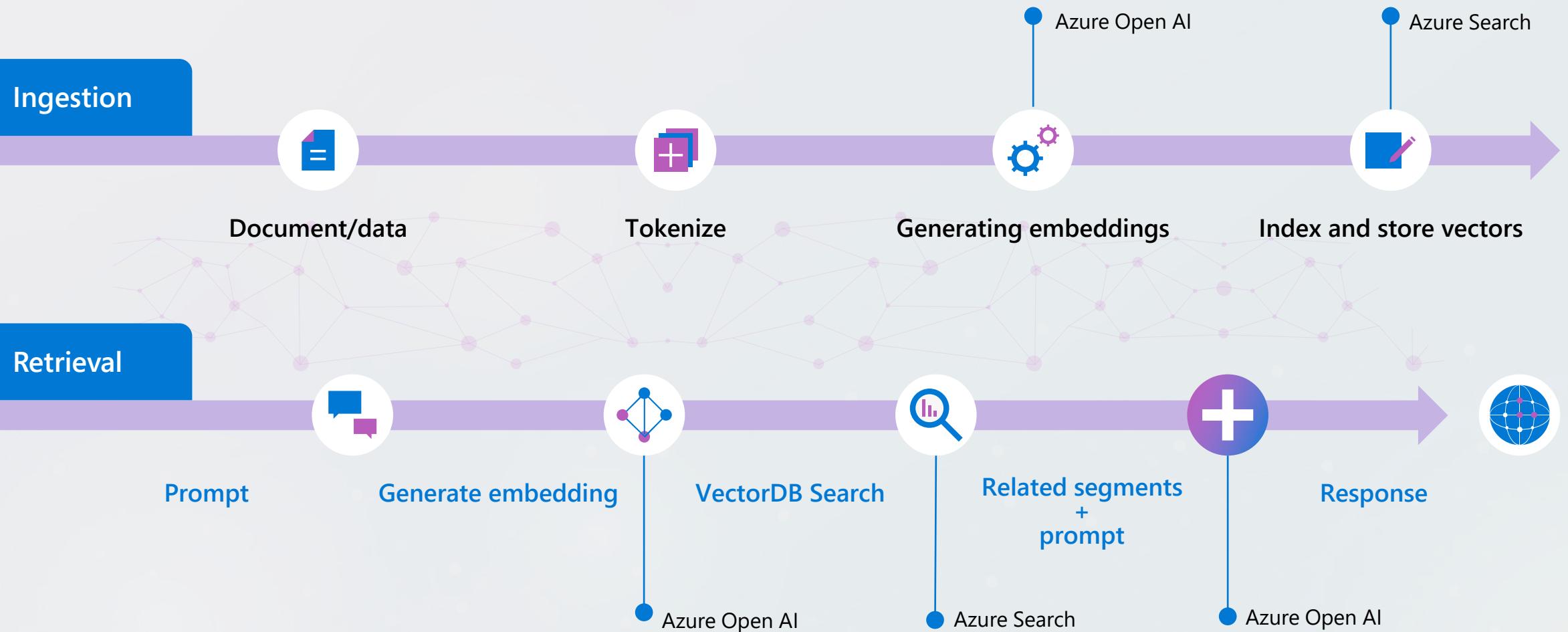


Logic Apps Capabilities



Integration best practices

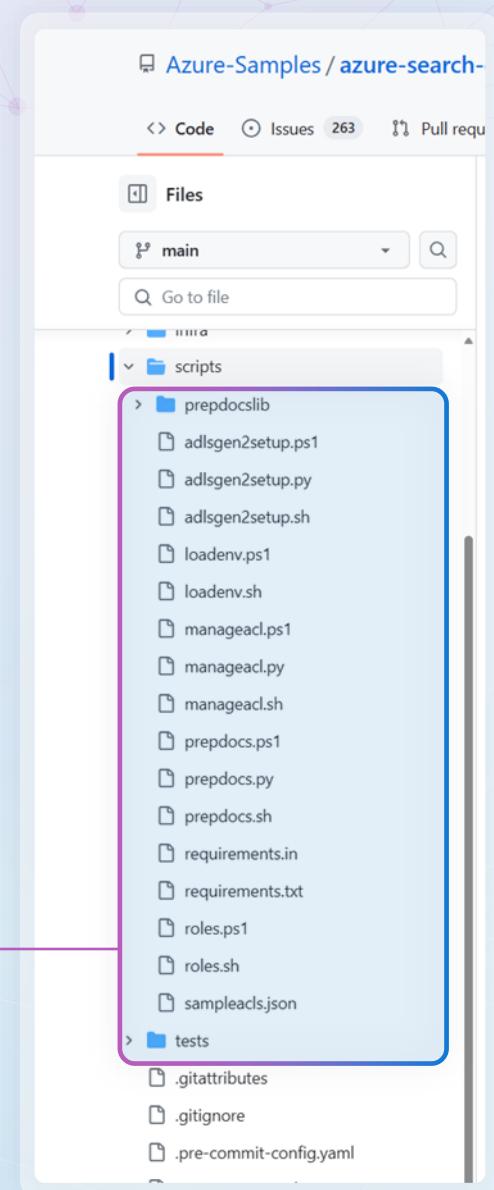
Ingestion and Retrieval using RAG pattern



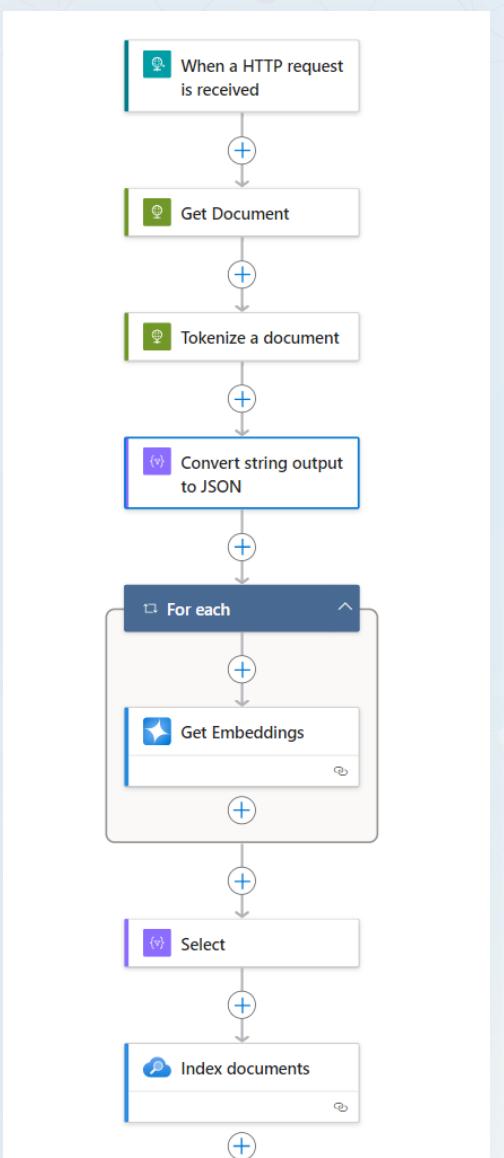
RAG applications using Logic Apps

Manual steps that will
be saved by using Logic
Apps connector

Ingestion flow
with code



Ingestion flow with
Logic Apps connector



RAG applications using Logic Apps

Manual steps that will
be saved by using Logic
Apps connector

Retrieval flow
with code

Azure-Samples / azure-search

Code Issues 263 Pull requests

Files

main Go to file

.github .vscode app backend approaches core app.py gunicorn.conf.py main.py requirements.in requirements.txt text.py frontend mypy.ini start.ps1 start.sh

Retrieval flow with
Logic Apps

When a HTTP request
is received

system message

generate search query

get search query

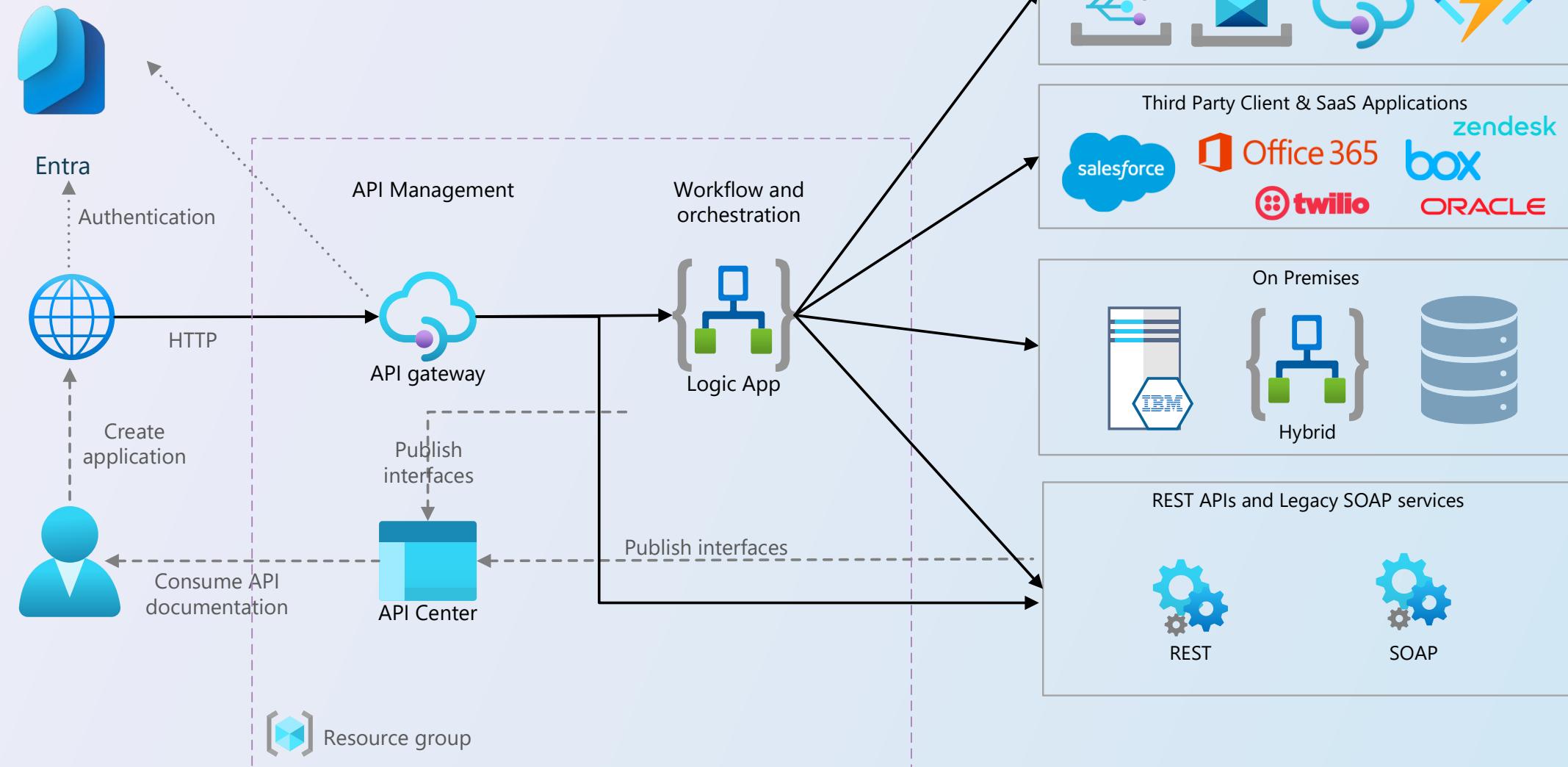
Gets a single
embedding

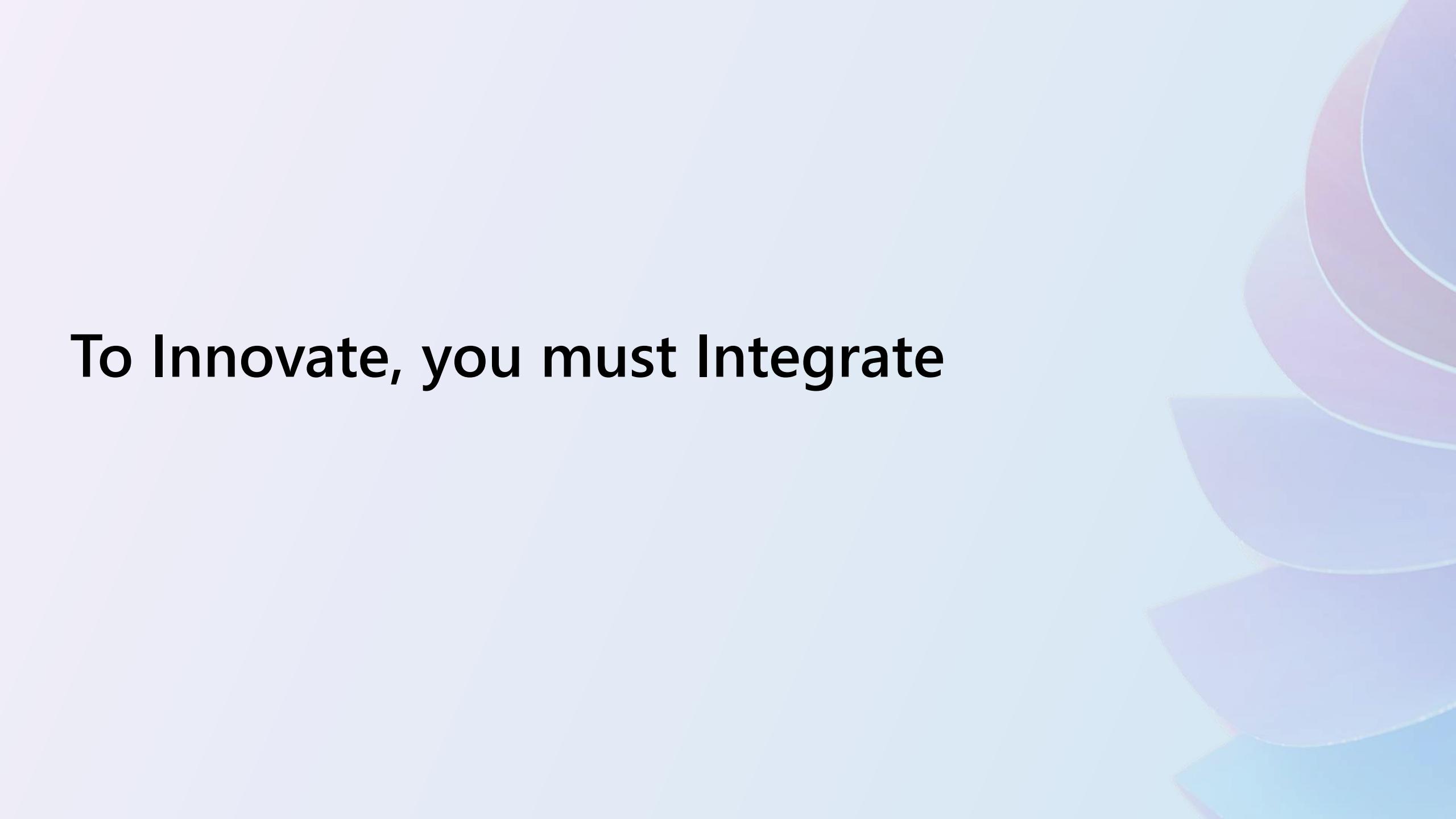
Vector search

create prompt

Get chat completions

Reference Architecture

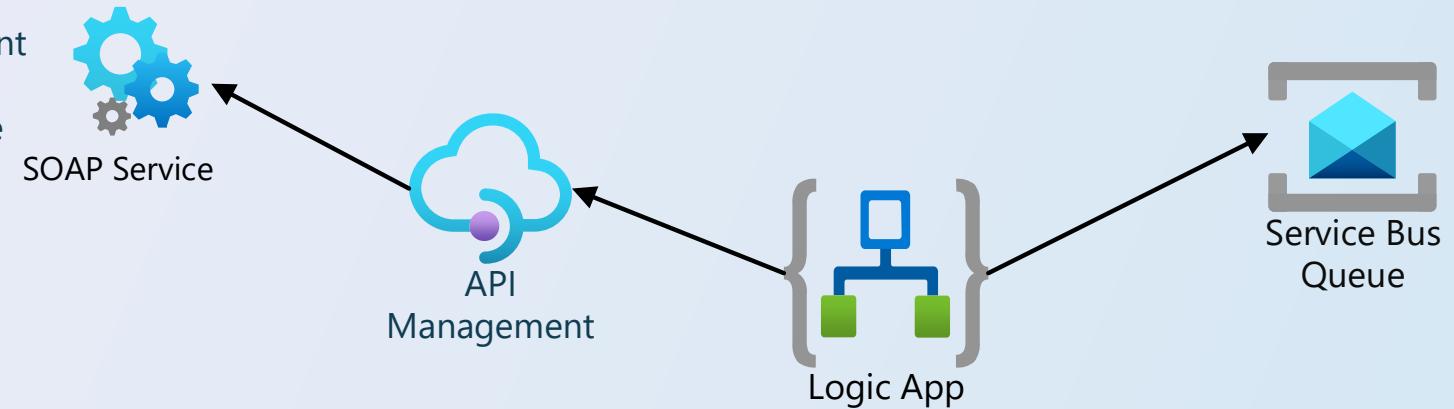




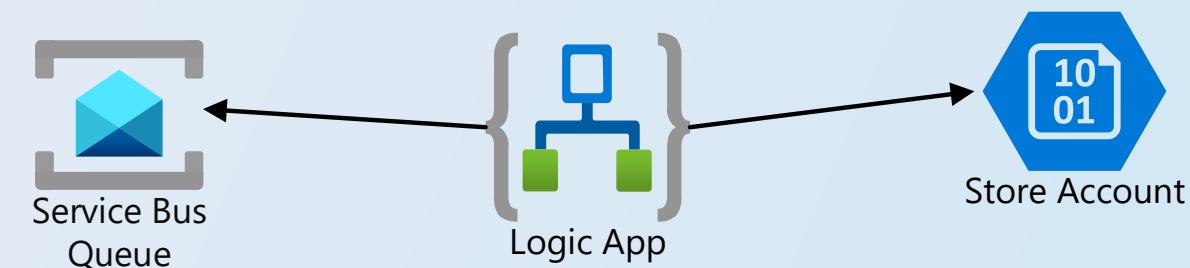
To Innovate, you must Integrate

Demo

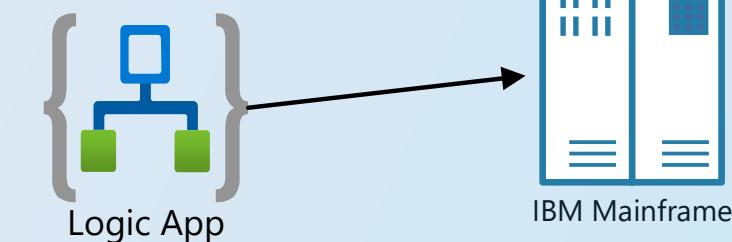
Workflow calls SOAP Service to get Orders. Logic Apps uses API Management connector to get the orders. Then the orders are sent to a Service Bus Queue Q1.



Workflow waits for Service Bus event to get Orders from Queue Q1. Logic Apps Then the orders are sent to an storage account.



BONUS: Workflow is triggered via HTTP to call different systems running on an IBM mainframe (via secure Expressroute circuit)



Thank you!



@hcamosu



hcamosu@microsoft.com



<https://www.linkedin.com/in/hcamosu>



<https://www.youtube.com/@hcamosu>