

SERVERLESS OVERVIEW

Intro to Azure Functions





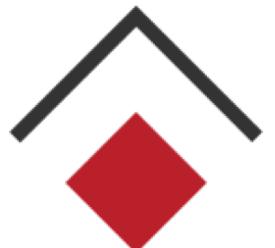
architectnow.net | @architectnow



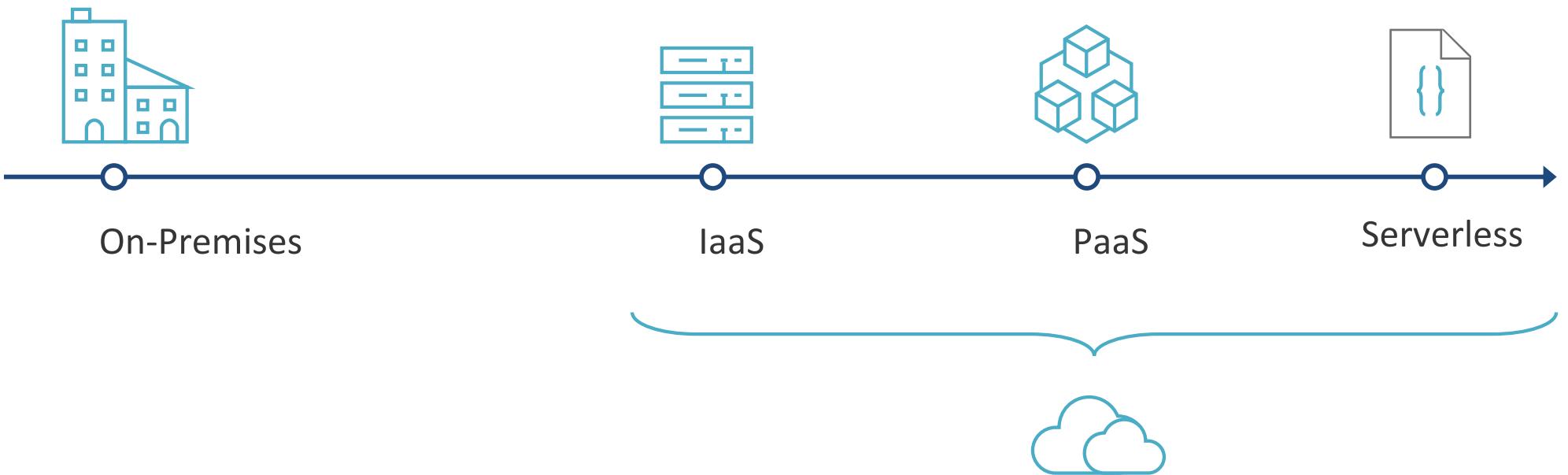
Chris Young

Director of Development

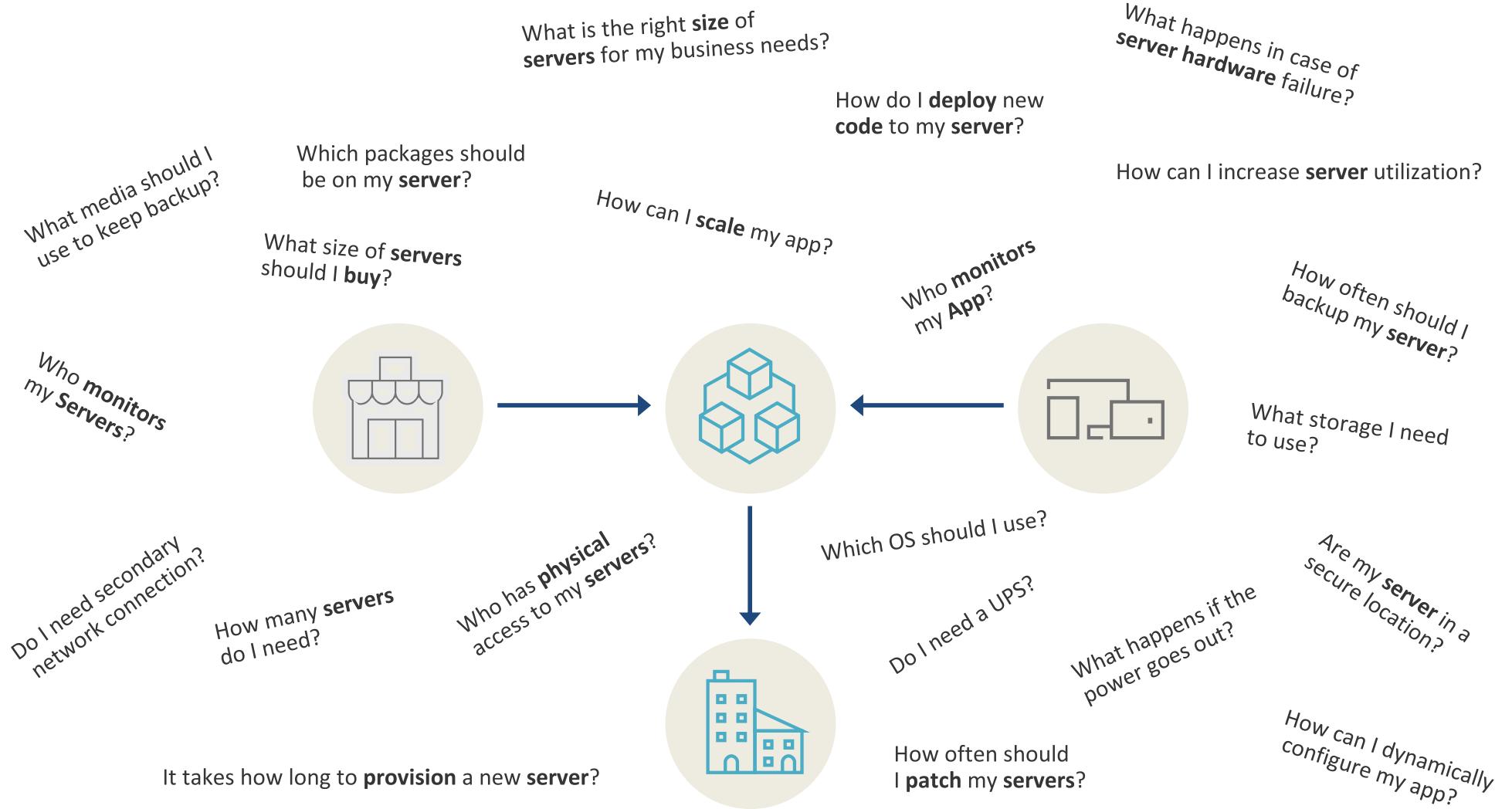
cyoung@architectnow.net | @chrisyoung



The “evolution” of application platforms



Before cloud



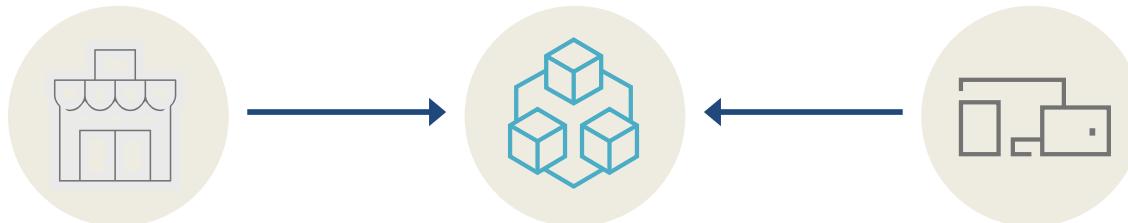
Then came IaaS—table stakes for digital business

What is the right **size of servers** for my business needs?

How can I increase **server utilization**?

How many **servers** do I need?

How can I **scale** my app?



How often should I **patch** my **servers**?

How often should I **backup** my **server**?

Which packages should be on my **server**?

How do I **deploy** new **code** to my **server**?

Which OS should I use?

Who **monitors** my App?



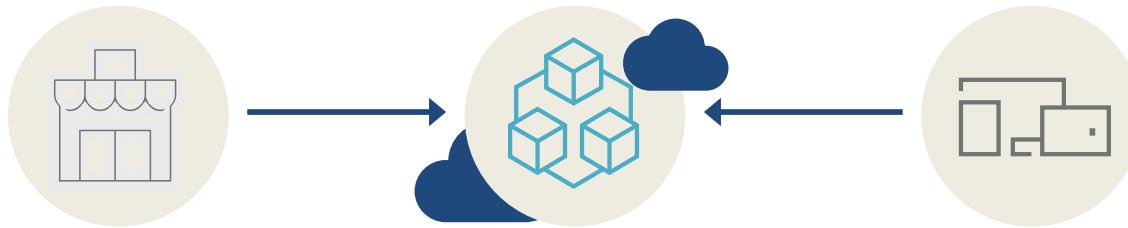
Then PaaS, critical for digital transformation

What is the right **size** of “**servers**” for my business needs?

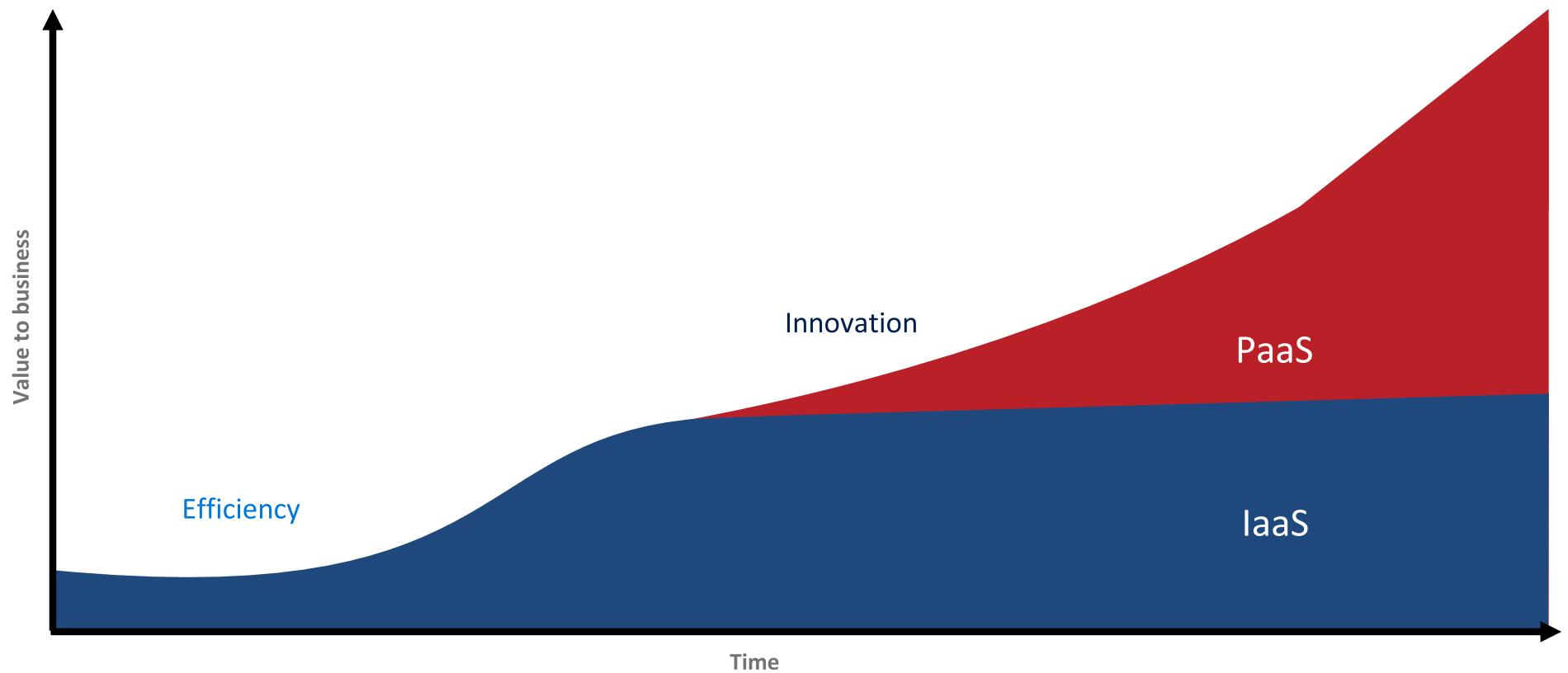
How can I increase “**server**” utilization?

How many “**servers**” do I need?

How can I **scale** my app?



Modernize your apps with PaaS

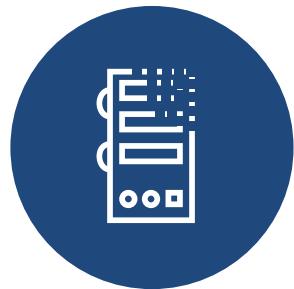


Serverless, the future platform for next gen apps

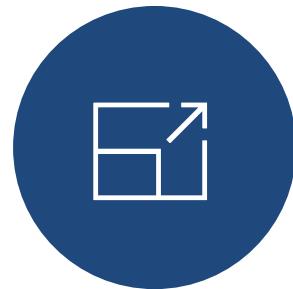
How do I **architect** my app to become Serverless?



What is Serverless?



Abstraction
of servers



Event-driven/
instant scale



Micro-billing

Benefits of Serverless



Reduced DevOps

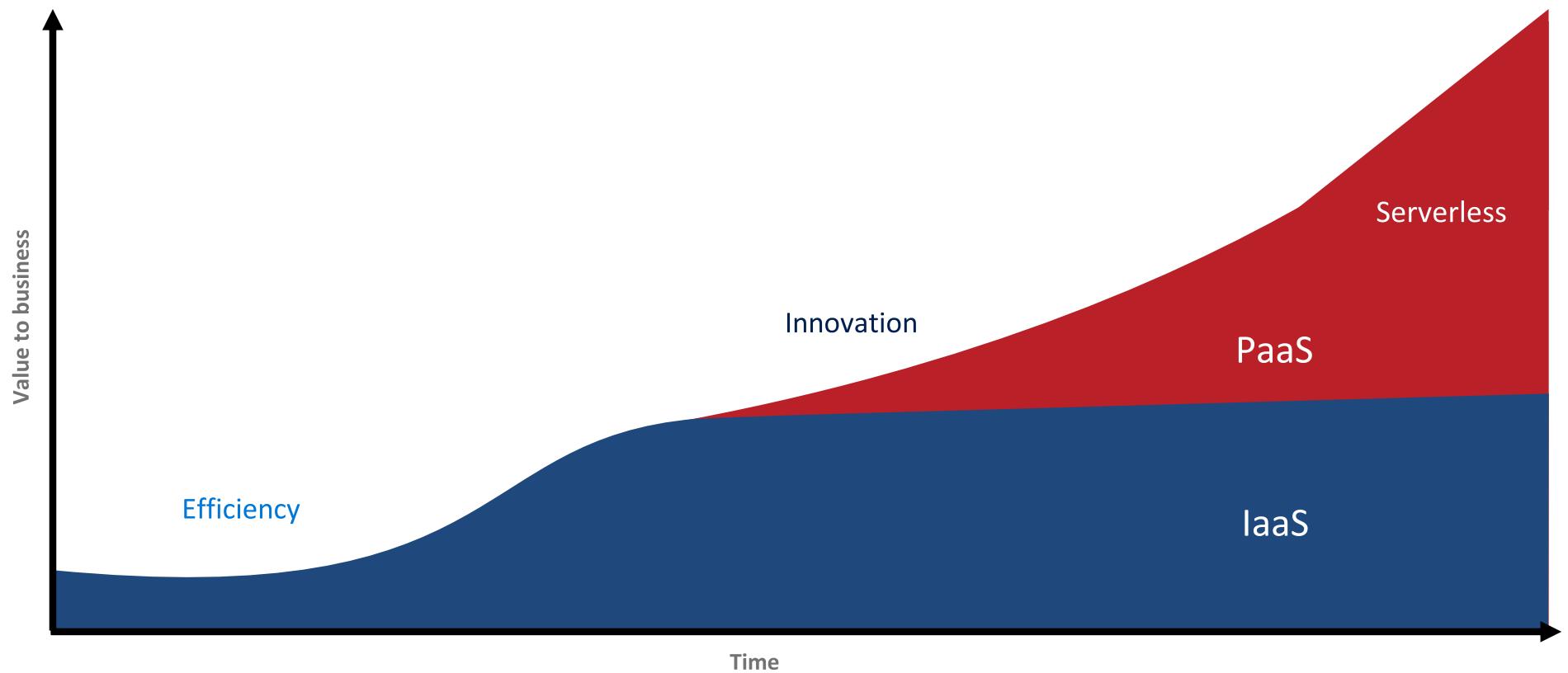


Focus on business
logic

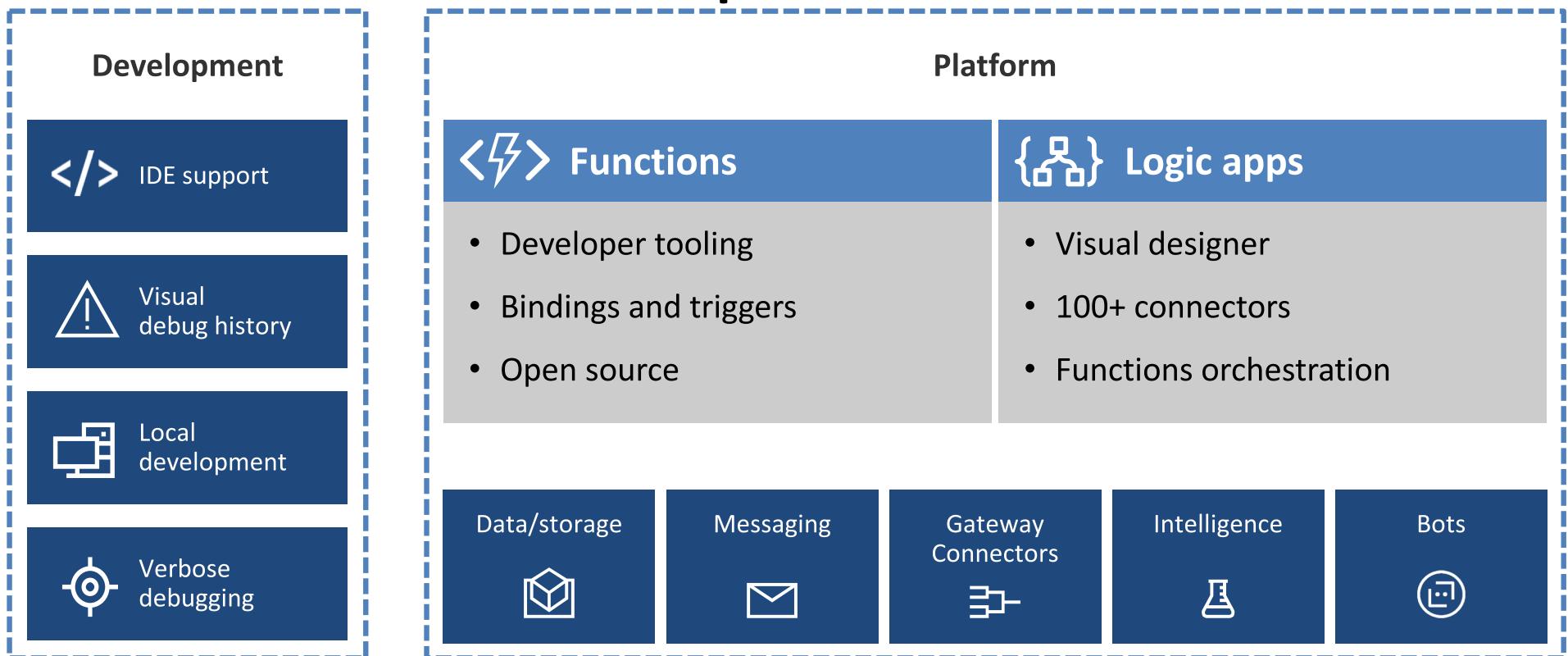


Faster time to
market

Build apps faster with Serverless

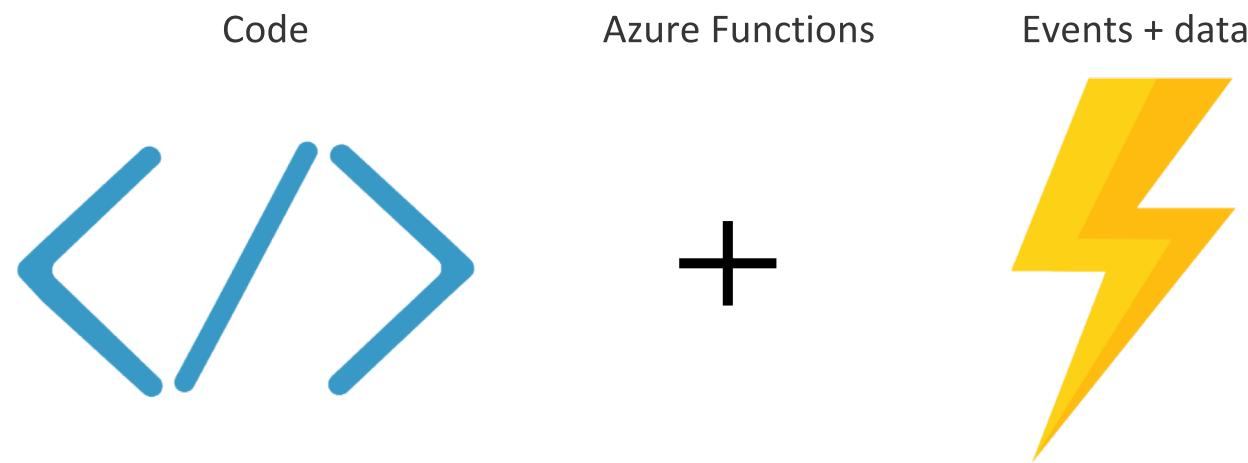


Serverless application platform components



Azure Functions

Introducing Functions



Use bindings in your code

run.csx

```
public static void Run(byte[] image, string filename,
                      Stream outputBlob, TraceWriter log)
{
    log.Info($"Processing image: {filename}");

    var imageBuilder = ImageResizer.ImageBuilder.Current;

    imageBuilder.Build(
        image, outputBlob,
        new ResizeSettings(640, 400, FitMode.Max, null), false);
}
```

function.json

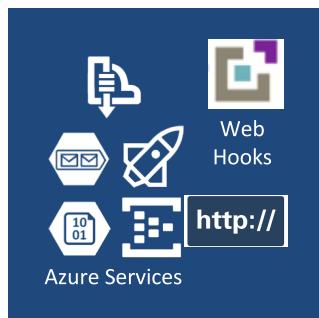
```
{
  "bindings": [
    {
      "name": "image",
      "type": "blobTrigger",
      "direction": "in",
      "path": "card-input/{filename}.jpg",
      "connection": "AzureWebJobsStorage"
    },
    {
      "type": "blob",
      "name": "outputBlob",
      "path": "card-output/{filename}.jpg",
      "connection": "AzureWebJobsStorage",
      "direction": "out"
    }
  ]
}
```

Triggers and Bindings

Type	Service	Trigger	Input	Output
Schedule	Azure Functions	✓		
HTTP (REST or webhook)	Azure Functions	✓		✓*
Blob Storage	Azure Storage	✓	✓	✓
Events	Azure Event Hubs	✓		✓
Queues	Azure Storage	✓		✓
Queues and topics	Azure Service Bus	✓		✓
Tables	Azure Storage		✓	✓
Tables	Azure Mobile Apps		✓	✓
No-SQL DB	Azure DocumentDB		✓	✓
Push Notifications	Azure Notification Hubs			✓
Twilio SMS Text	Twilio			✓

Seamless DevOps experience with Functions

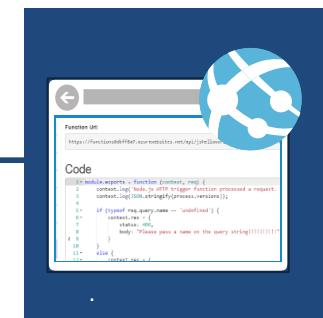
1) Trigger



2) Input binding



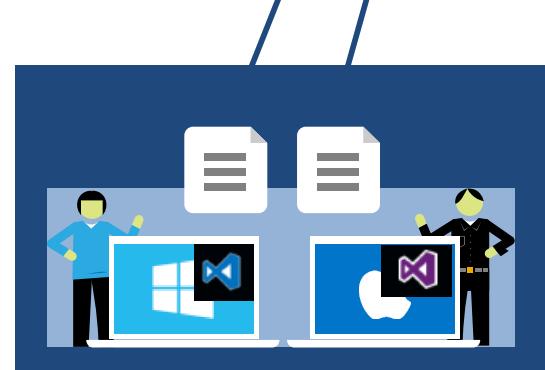
3) Develop



4) Execute



5) Output binding



7) Develop locally



6) Monitor and improve

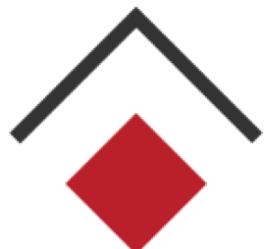
Azure Functions usage patterns

Great customer and community traction so far

“Brown field” – used as an integration, scripting tool (typically enterprises or larger ISVs)

“Green field” – used for backend creation (typically start-ups or smaller ISVs)

IoT – leveraged autonomously or as part of Azure IoT implementations



Logic Apps

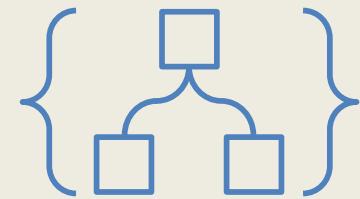
Introducing Azure Logic Apps

Powerful Integration and
Workflow Engine born in Azure

Fast integrations using a visual designer and workflow creation with triggers and actions

Connect applications, data and services

Connect and orchestrate Azure Functions



Logic apps
connectors—
Over 100 and
growing



facebook box twitter



....and more!

Logic Apps

Cloud APIs and platform functionality

Over 120 built-in connectors

Hosted and managed within the platform

Scales to meet your needs

First class designer experience

Rapid development

API connections

Authenticate once and reuse

Differentiate connection configuration

Simple to deploy

Portal experience for managing API Connections

SaaS

- appFigures
- Asana

- Azure API Management
- Azure App Services
- Azure Automation
- Azure Cognitive Face API
- Azure Cognitive LUIS
- Azure Cognitive Text Analytics
- Azure Cognitive Vision
- Azure Data Lake Store
- Azure Document DB
- Azure Event Hub
- Azure Functions
- Azure Machine Learning
- Azure Resource Manager
- Azure Service Bus
- Azure SQL
- Azure Storage Blob
- Azure Storage Queues

- Box
- Buffer
- Campfire

- Chatter
- Common Data Service
- Disqus
- DocuSign

- Dropbox
- Dynamics AX Online
- Dynamics CRM Online

- Dynamics CRM Service Bus

- Dynamics Financials
- Dynamics Operations

- Easy Redmine
- Eventbrite
- Facebook
- FreshBooks
- Freshdesk
- GitHub
- Gmail
- Google Calendar
- Google Contacts
- Google Drive
- Google Sheets
- Google Tasks
- GoTo Meeting
- GoTo Training
- GoTo Webinar
- Harvest
- HelloSign
- Infusionsoft
- JIRA
- Insightly
- Instagram
- Instapaper
- MailChimp
- Mandrill
- Medium
- Microsoft Project Online
- Microsoft Translator
- MSN Weather
- Muheimbi PDF
- Office 365
- Office 365 Users
- Office 365 Video
- OneDrive

- OneDrive for Business
- OneNote
- Outlook.com
- Outlook Tasks

- PagerDuty
- Pinterest
- Pipedrive
- Pivotal Tracker

- Power BI
- Project Online

- Redmine
- Salesforce
- Salesforce Chatter
- SendGrid

- SharePoint Online

- Slack
- SmartSheet
- SparkPost
- Stripe
- Survey Monkey
- Todoist
- Toodledo
- Trello
- Twilio
- Twitter
- Typeform

- UserVoice
- VS Team Services

- Webmerge

- wvorapress

- Wunderlist

- Yammer
- YouTube
- Zendesk

Protocols/Native

- HTTP, HTTPS
- HTTP Webhook
- FTP, SFTP
- SMTP
- RSS
- Compose, Query, Parse JSON
- Wait
- Terminate
- Workflow

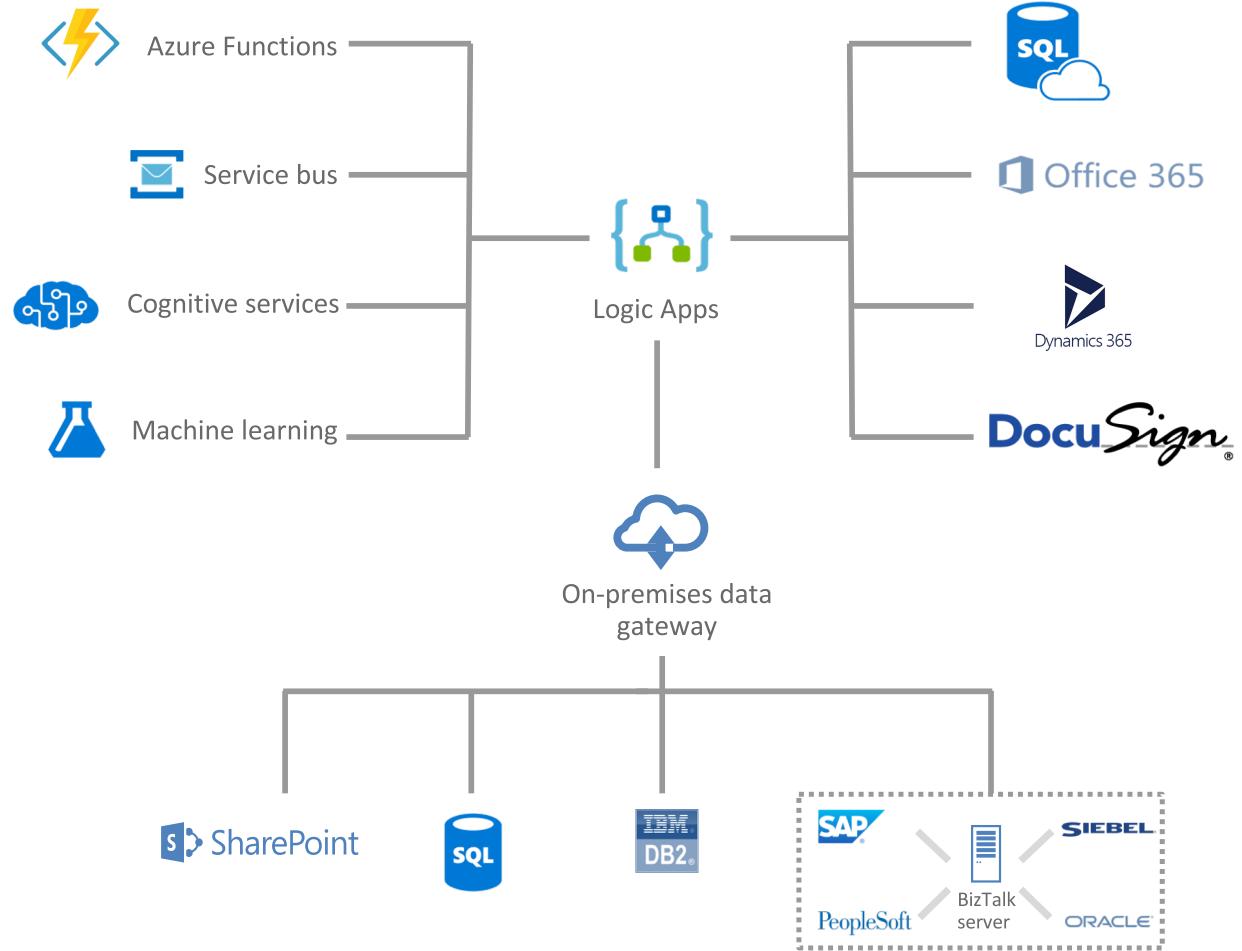
XML and EDI

- XML Validation
- Transform XML (+Mapper)
- Flat File Encode
- Flat File Decode
- X12
- EDIFACT
- AS2
- Integration Account Artifact Lookup

Hybrid

- BizTalk Server
- File System
- IBM DB2
- Informix
- Oracle DB
- SharePoint Server
- SQL Server
- SAP
- Websphere MQ

Logic Apps connects everything



Logic Apps Workflow Designer

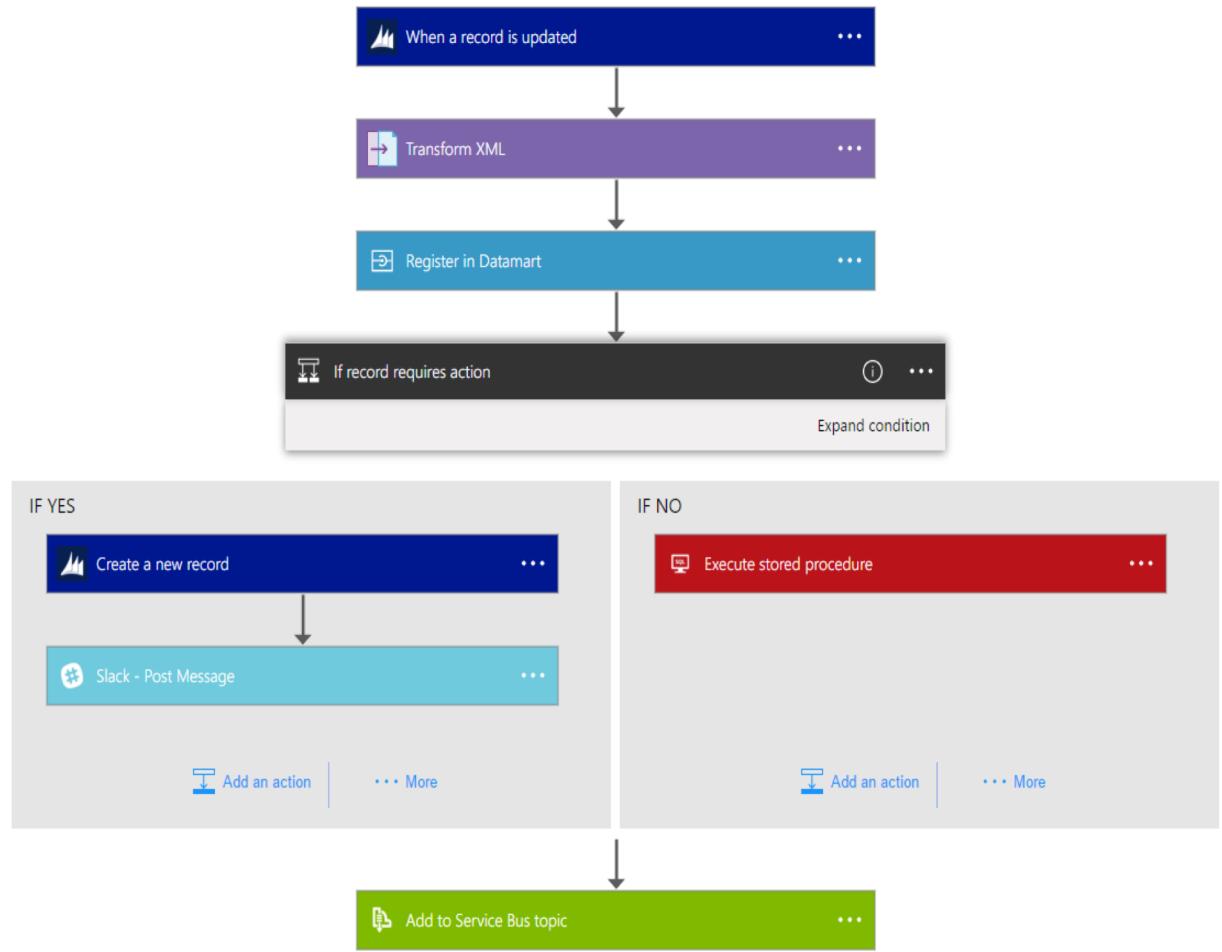
Workflow in the cloud

Powerful control flow

Connect disparate
applications

No code designer for
rapid creation

Also works within Visual
Studio for added CI/CD



Scenario 1

Real time Stream Analytics



Real-time stream analytics

Tasks/activities

- Replicate logs from one data center to another
- Analyze the logs
- Take action based on data



Before serverless

- Setup a VM/Container/WebJobs
- Build/patch/deploy (OS)
- Monitor infra (VM/Container)
- Manage FTP creds
- Use FTP library
- Use Azure SDK

Using serverless

- Setup a VM/Container/WebJobs
- Build/patch/deploy (OS)
- Monitor infra (VM/Container)
- Manage FTP with dynamic settings (automation)
- Use FTP library

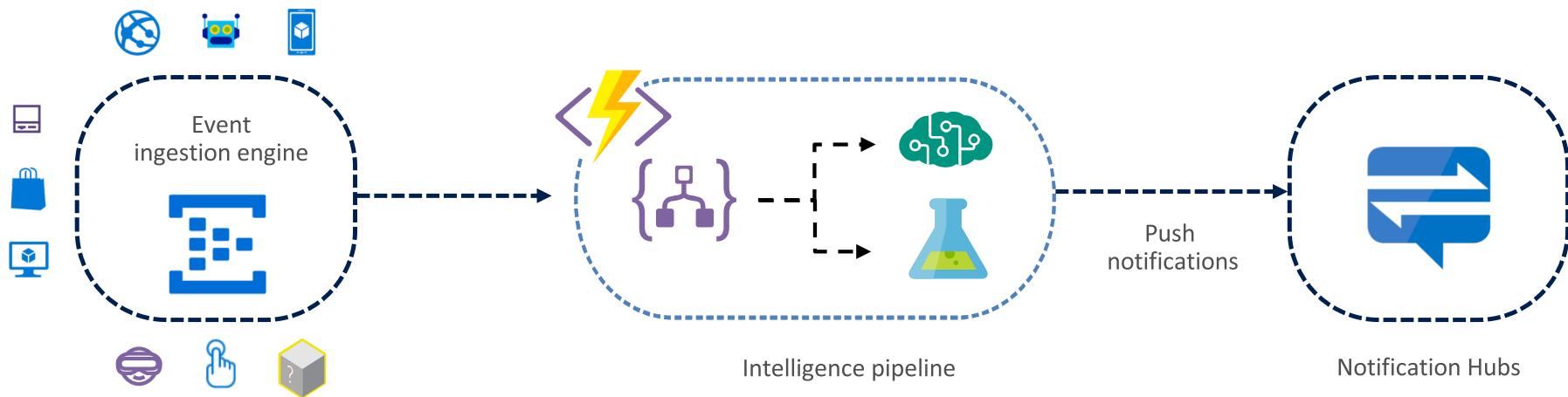
Scenario 2

Geo based marketing

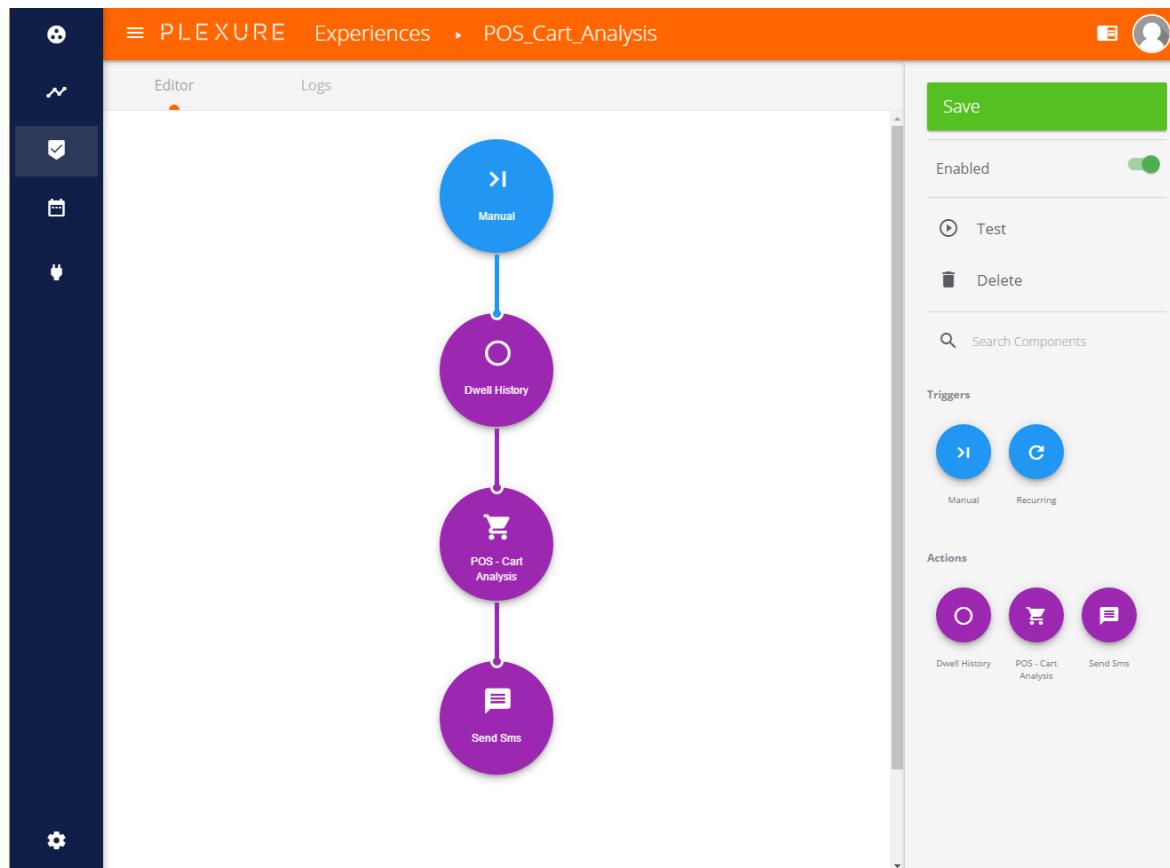
PLEXURE Geo-based marketing

Tasks/activities	Before serverless	Using serverless
<input checked="" type="checkbox"/> Stream logs to event ingestion engine	<input checked="" type="checkbox"/> Setup a VM/Container/WebJobs	<input checked="" type="checkbox"/> Setup a VM/Container/WebJobs
<input checked="" type="checkbox"/> Analyze the logs and live POS feed	<input checked="" type="checkbox"/> Build/patch/deploy (OS)	<input checked="" type="checkbox"/> Build/patch/deploy (OS)
<input checked="" type="checkbox"/> Push custom notifications based on data	<input checked="" type="checkbox"/> Monitor infra (VM/Container)	<input checked="" type="checkbox"/> Monitor infra (VM/Container)
	<input checked="" type="checkbox"/> Connect to partner integration system	<input checked="" type="checkbox"/> Connect to partner integration system
	<input checked="" type="checkbox"/> Ingest data logs	<input checked="" type="checkbox"/> Ingest data logs
	<input checked="" type="checkbox"/> Analyze logs and live POS feed	<input checked="" type="checkbox"/> Analyze logs and live POS feed
	<input checked="" type="checkbox"/> Push custom notifications	<input checked="" type="checkbox"/> Push custom notifications
	<input checked="" type="checkbox"/> Infrastructure customization	

Plexure architecture diagram



Plexure Workflow: Logic Apps as Backend



Build apps faster

Functions

- Consumption and provisioned options
- Local development and advanced DevOps with detailed monitoring
- Bindings to first and third-party services
- Open source runtime

Logic Apps

- Rich portal visual designer and Visual Studio integration
- Switch statements and loops
- 120+ connectors to SaaS and PaaS, with custom connector support
- ARM template-based for easy reuse

Increased developer productivity

Jump start your development

- Pre-built templates
- Automatic triggers
- Declarative bindings
- Use existing development skills

Orchestrate workflows easily

- Graphical visual designer
- Over 100+ connectors
- Edit, author, visualize workflows easily
- Quickly orchestrate multiple functions

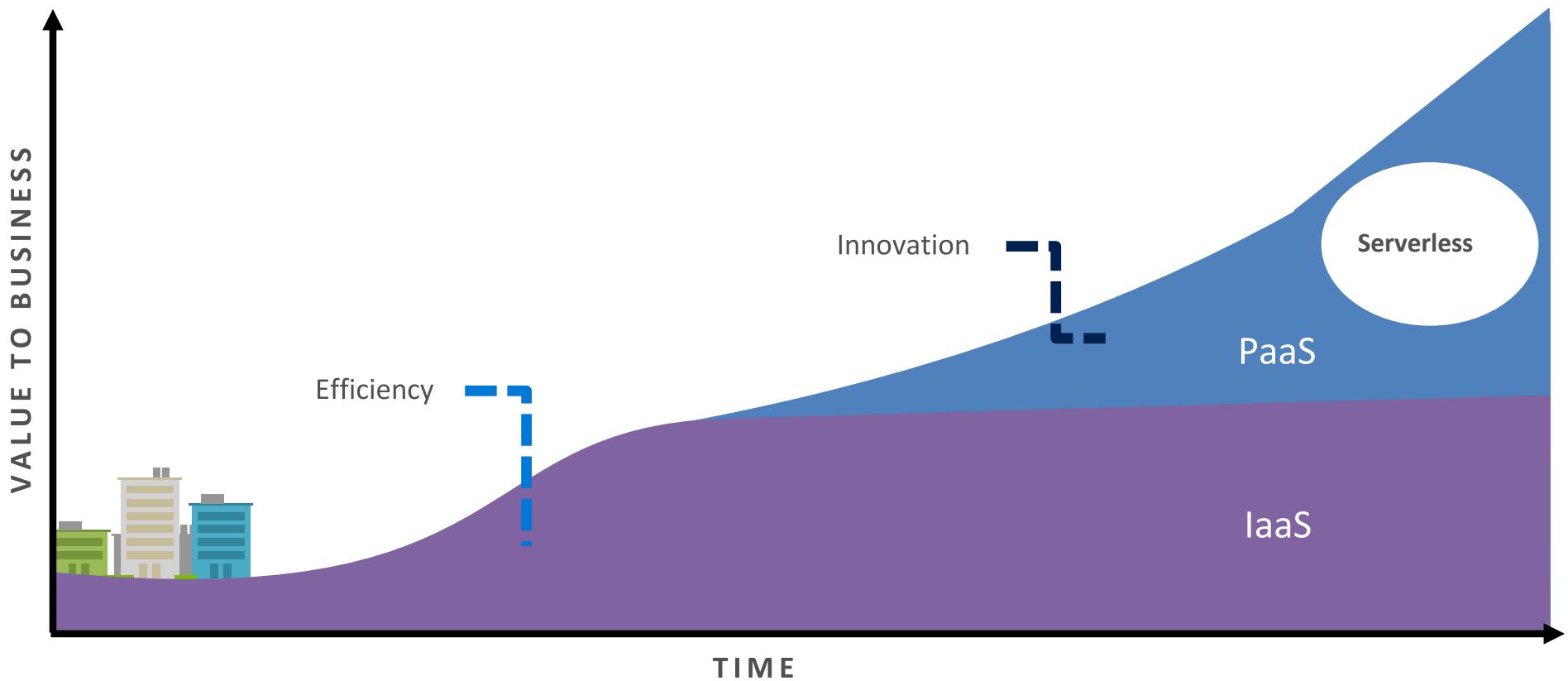
Build rich business apps

- Integrate with PowerApps and CDS*
- Integrate with Microsoft Flow*
- Git/GitHub integration

Rich development tools

- IDE support for visual studio
- F5 and visual debugging
- Verbose debugging
- Local SDK/runtime

Where does Serverless sit?



Serverless application platform

INNOVATE FASTER—PAY LESS

Build apps faster	Enable business agility	Reduce IT spend
<ul style="list-style-type: none">✓ Reduce development effort✓ Build rich business and LOB apps✓ Orchestrate a Serverless app in minutes	<ul style="list-style-type: none">✓ Continuous deployment✓ Leverage existing development skills✓ Build microservices	<ul style="list-style-type: none">✓ Continuous auto-scaling✓ Fully managed infrastructure✓ Micro-billing

Scenario 4

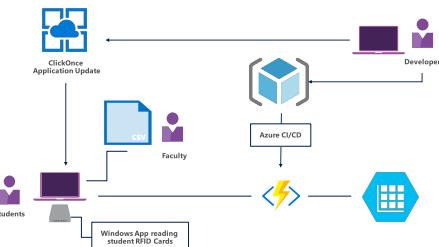
New application creation



New App creation

Tasks/activities

- Write a new web application
- Collect student data
- Co-relate student ID to student names
- Store student name and attendance information

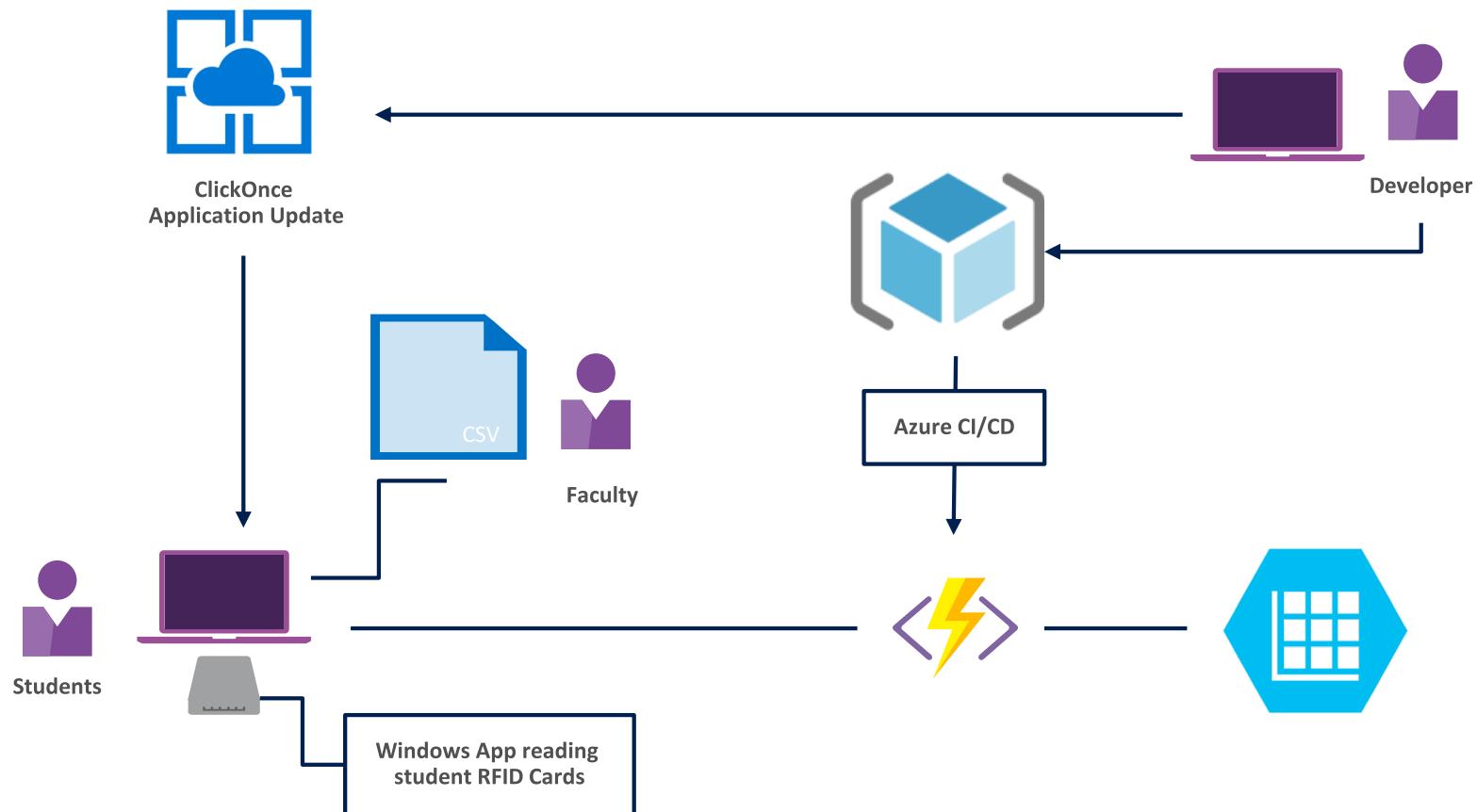


Before Serverless

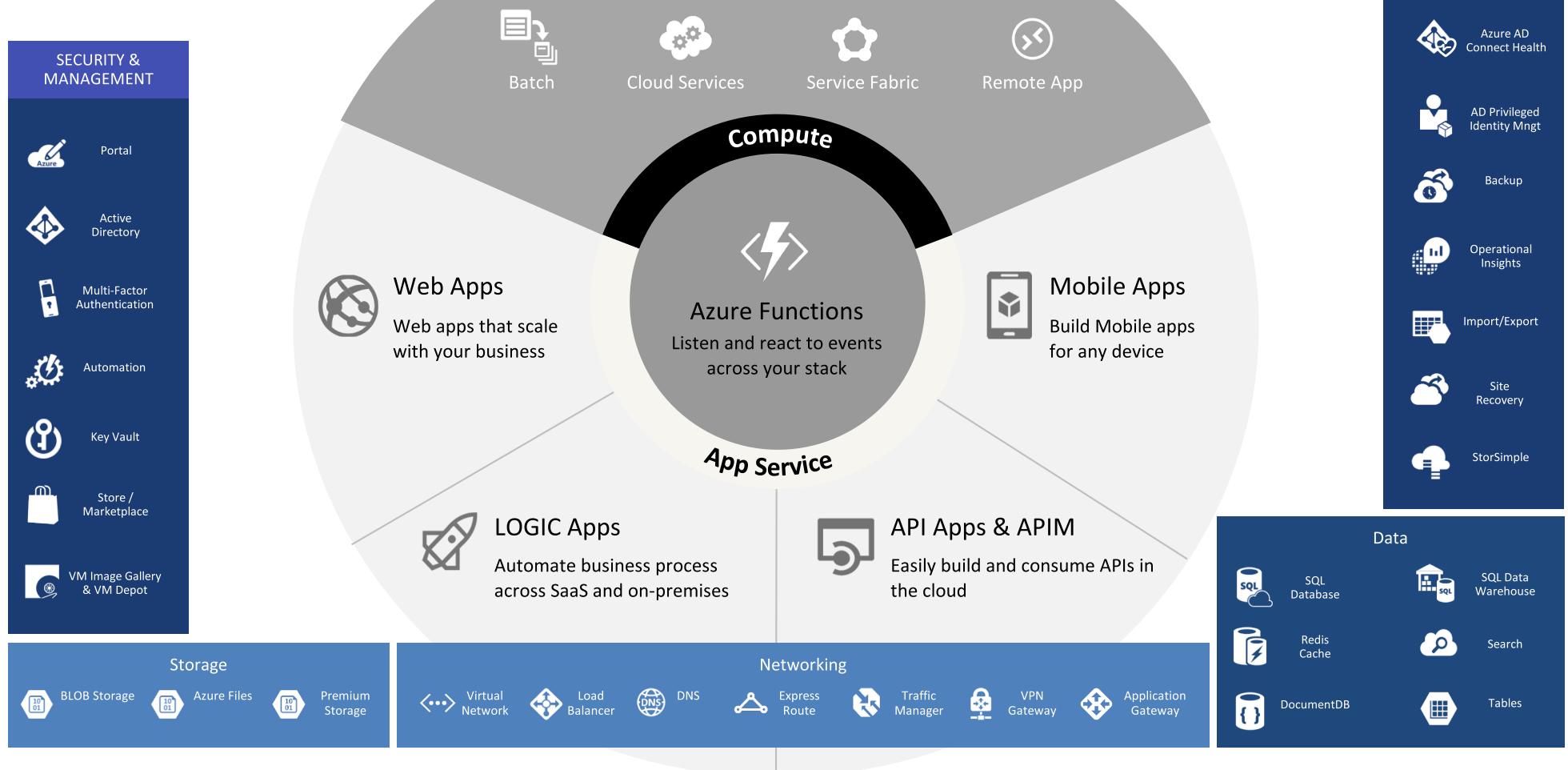
- Setup a VM/Container/WebJobs
- Build/patch/deploy (OS)
- Monitor infra
- Deploy WPF application
- Collect student ID from USB RFID reader
- Co-relate student ID to names and student records
- Store student and attendance information in table storage

Using Serverless

- Setup a VM/Container/WebJobs
- Build/patch/deploy (OS)
- Monitor infra
- Deploy WPF application
- Use RFID card reader to collect student IDs
- Co-relate student ID to names and student records
- Store student and attendance information in table storage



Functions: accelerate cloud development





architectnow.net | @architectnow



Chris Young

Director of Development

cyoung@architectnow.net | @chrisyoung