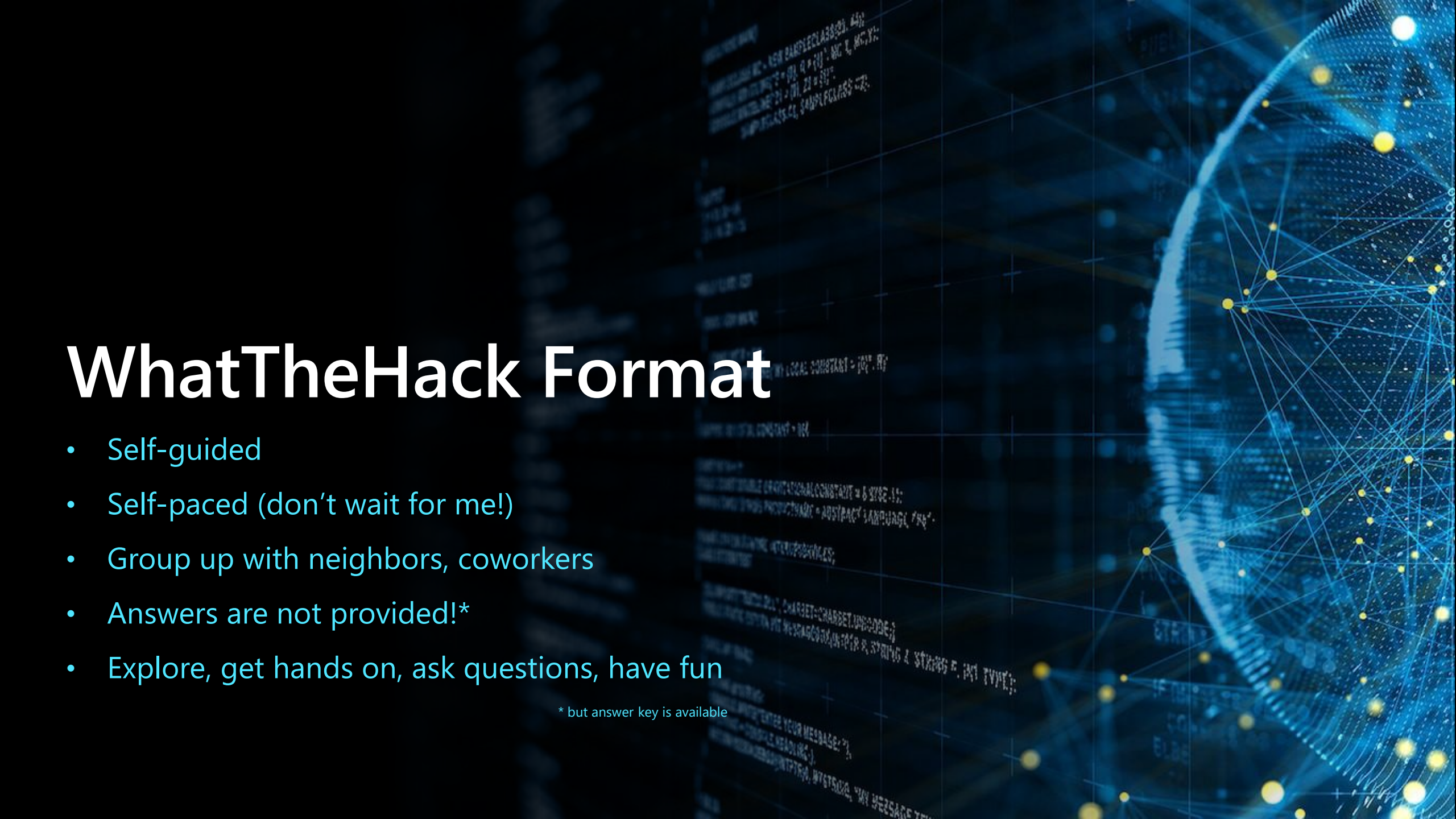
WhatTheHack

Azure OpenAI Apps

Agenda

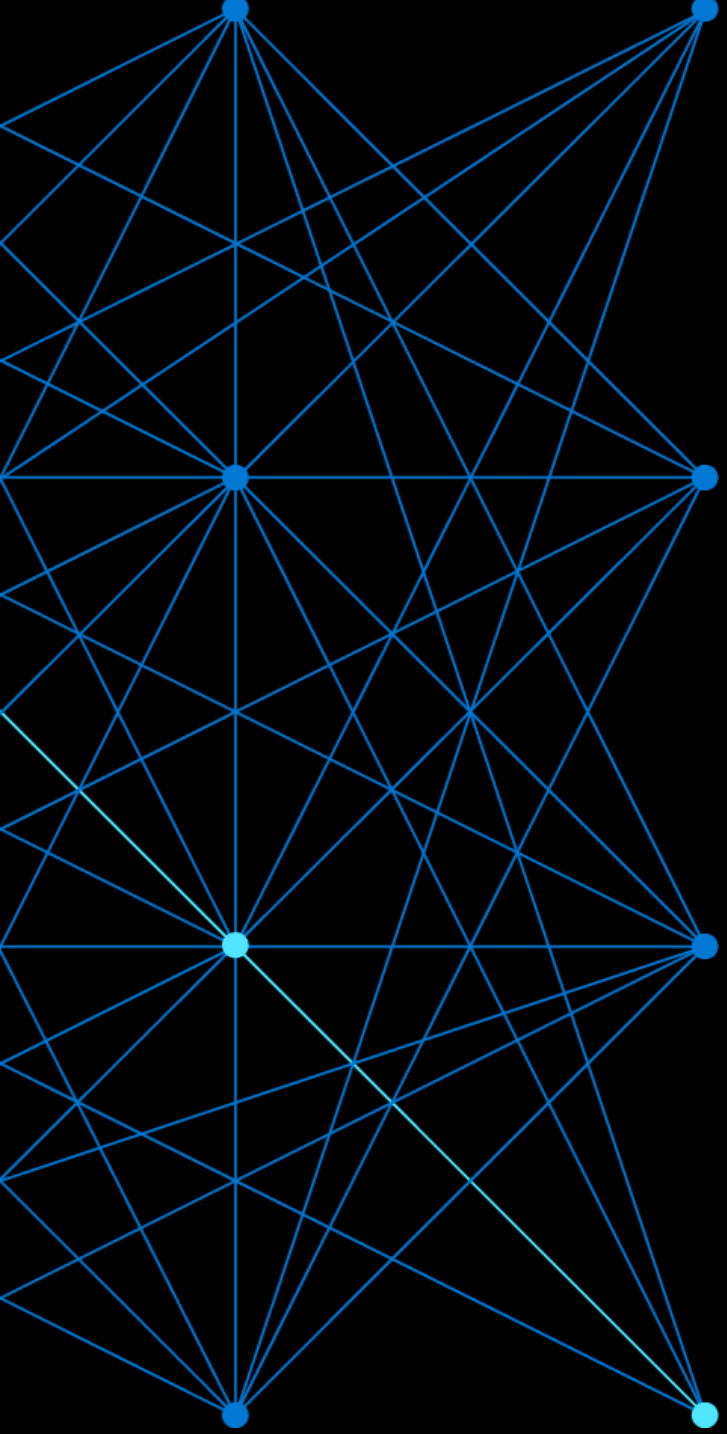
01	Introduction
02	Deploy Azure Prerequisite Infrastructure (30-45 minutes)
03	Challenge #1 (1 hour)
04	Challenge #2 (1 hour)
05	Lunch (1 hour)
06	Challenge #3 (1-2 hours)
07	Go Home
08	Come Back
09	Challenge #4 (1-2 hours)
10	Challenge #5 (1-2 hours)



WhatTheHack Format

- Self-guided
- Self-paced (don't wait for me!)
- Group up with neighbors, coworkers
- Answers are not provided!*
- Explore, get hands on, ask questions, have fun

* but answer key is available



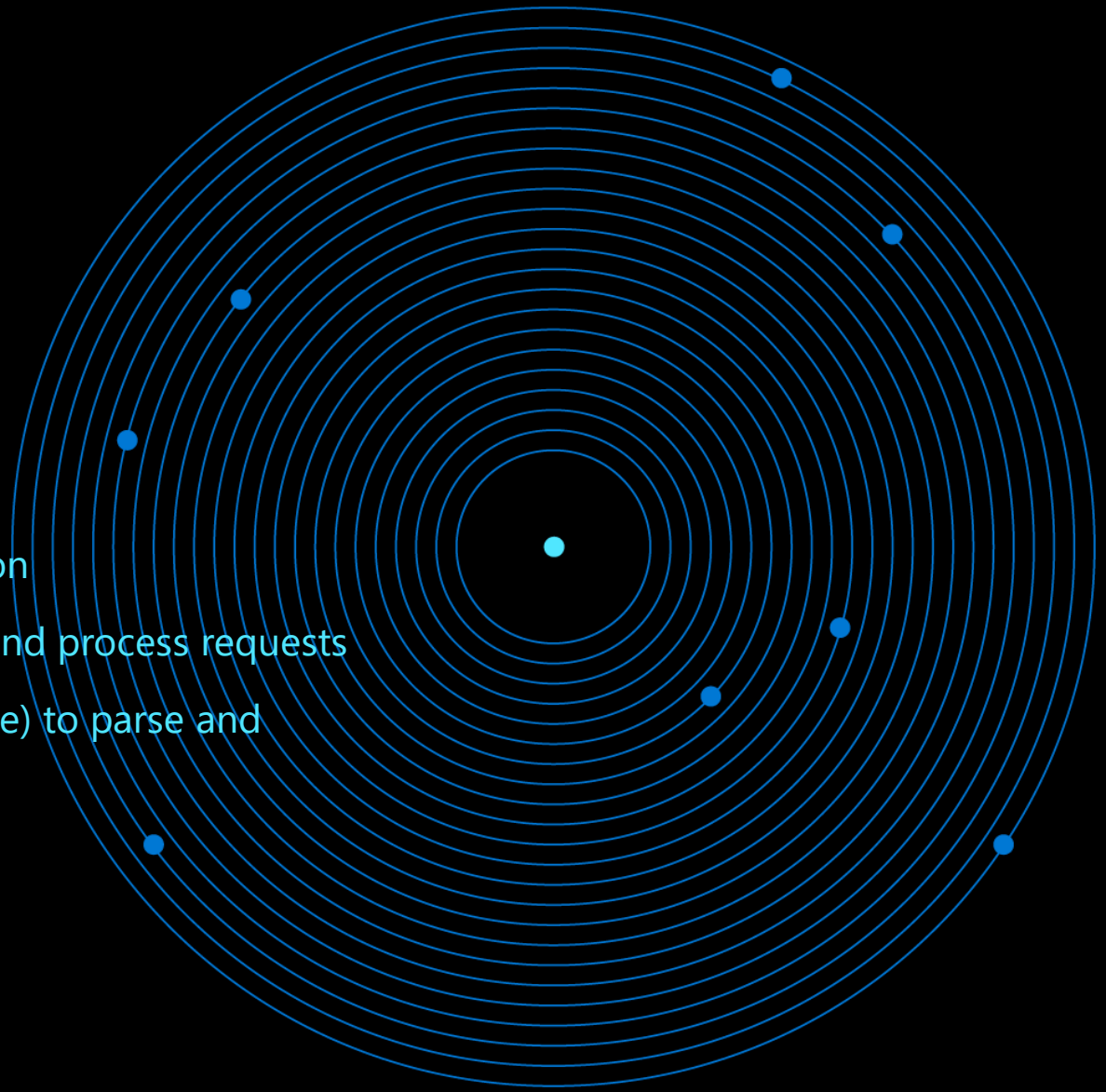
Learning Objectives

- Choose the right architecture and Azure services for different AI scenarios
- Provision, configure, and monitor Azure OpenAI resources
- Build intelligent apps and Q&A assistants using RAG architectures
- Implement solutions for both batch and real-time AI use cases

Microsoft Azure

Tasks

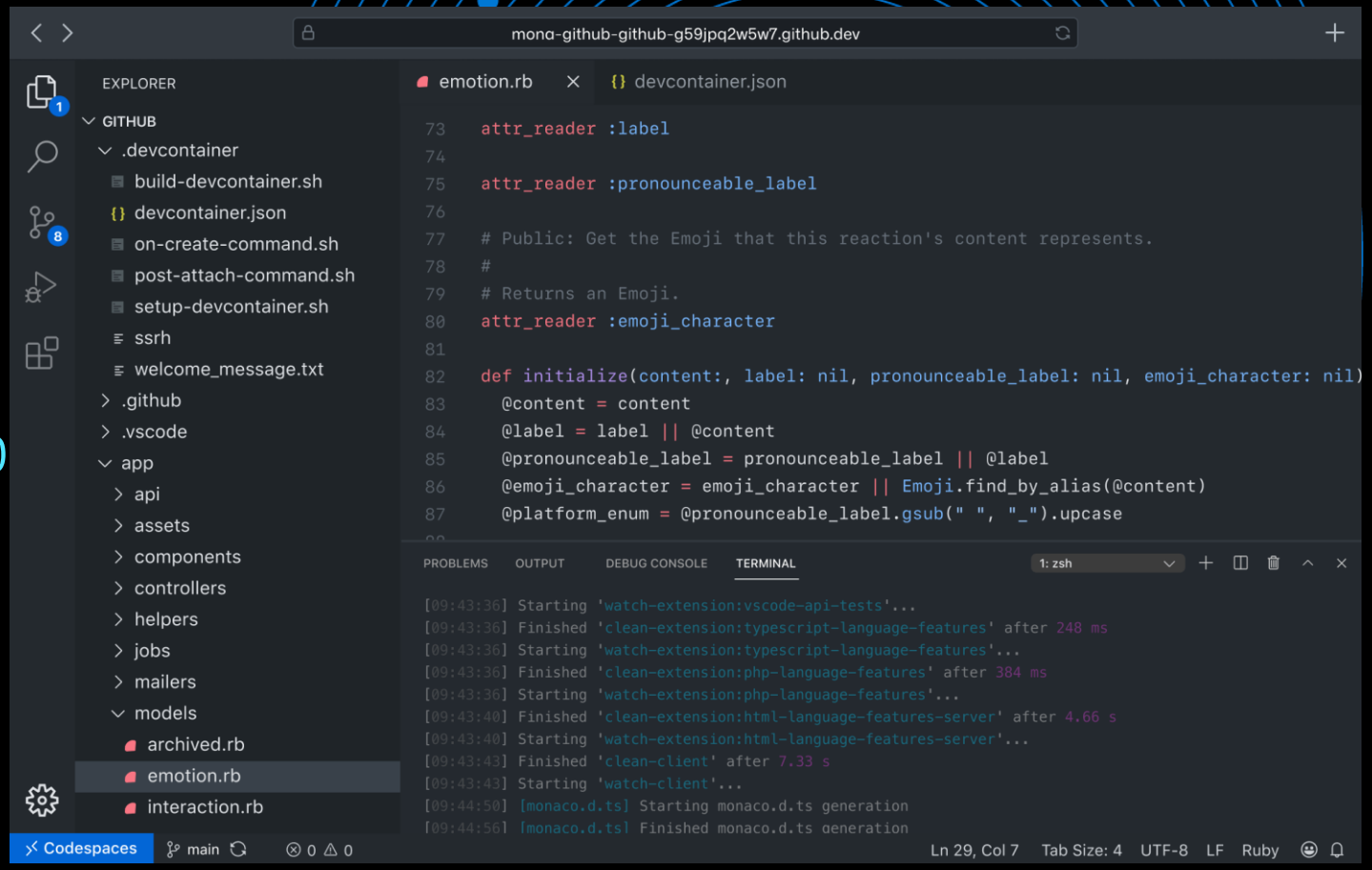
- Deploy Azure services using infrastructure as code
- Load source data into a retrieval system through automation
- Prompt engineering to enable agents to invoke functions and process requests
- Configure Azure AI platform services(Document Intelligence) to parse and prepare the data for the LLM
- Optimization of Azure OpenAI LLMs through caching
- Understand design concerns in a generative AI application

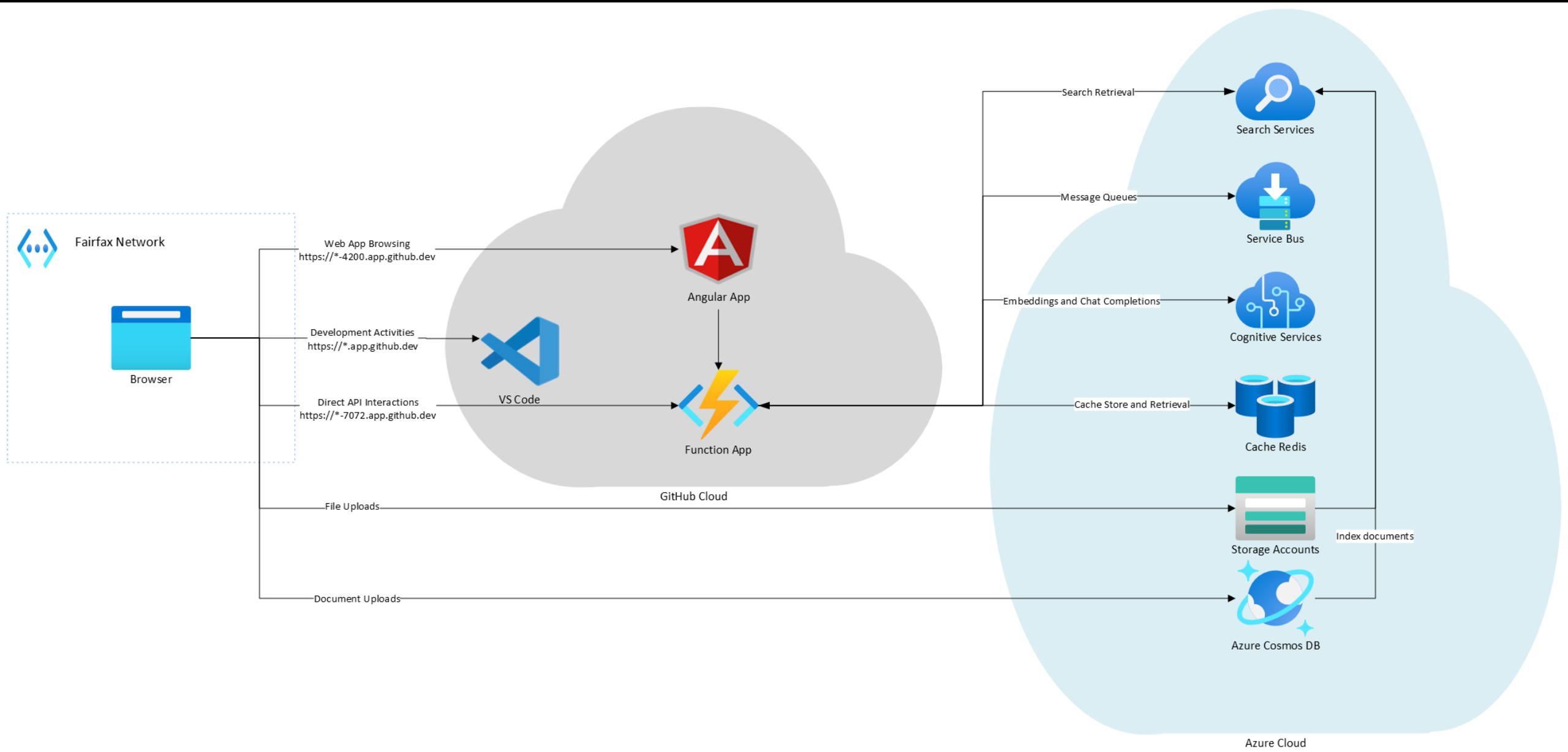


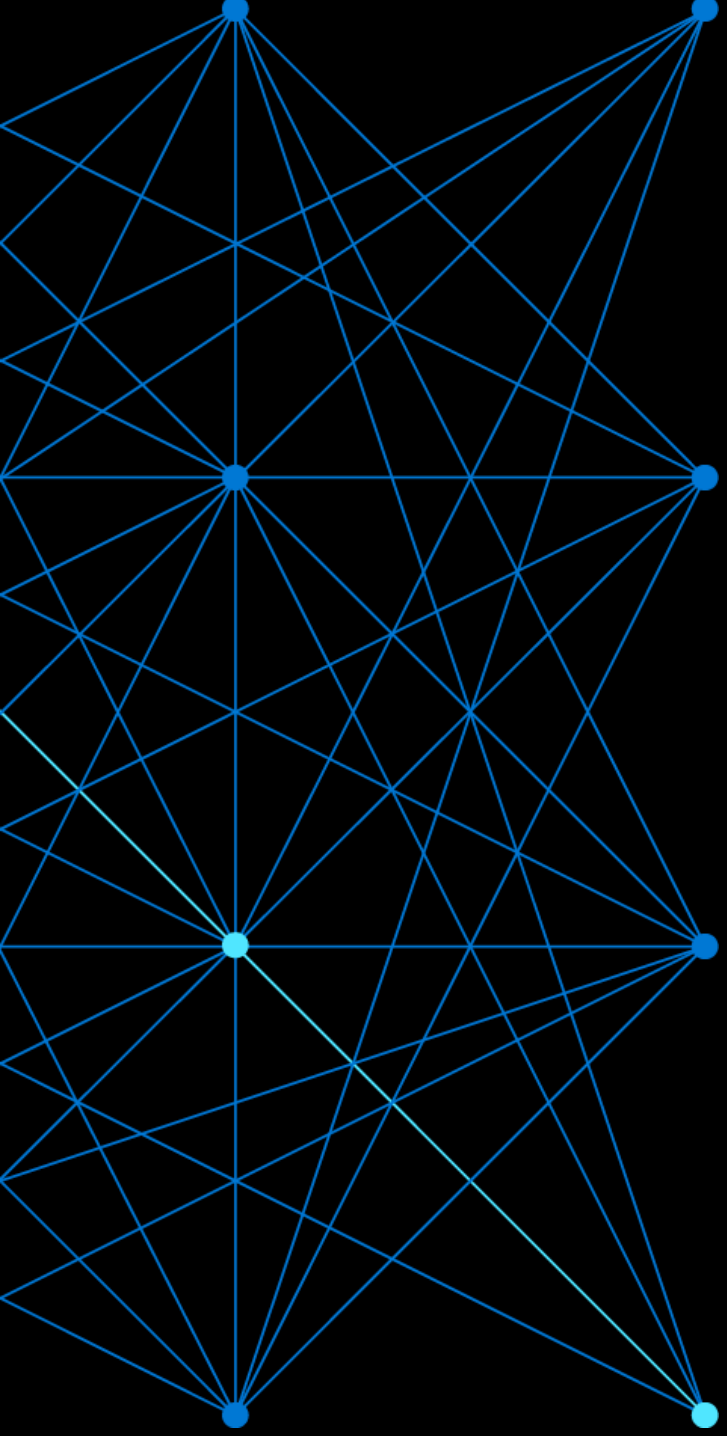
Microsoft Azure

GitHub Codespaces

- Browser-based VS Code IDE
- Execution, runtime happens on cloud compute
- Packages all development tools, prerequisites
- Free tier for personal accounts up to 80 core hours

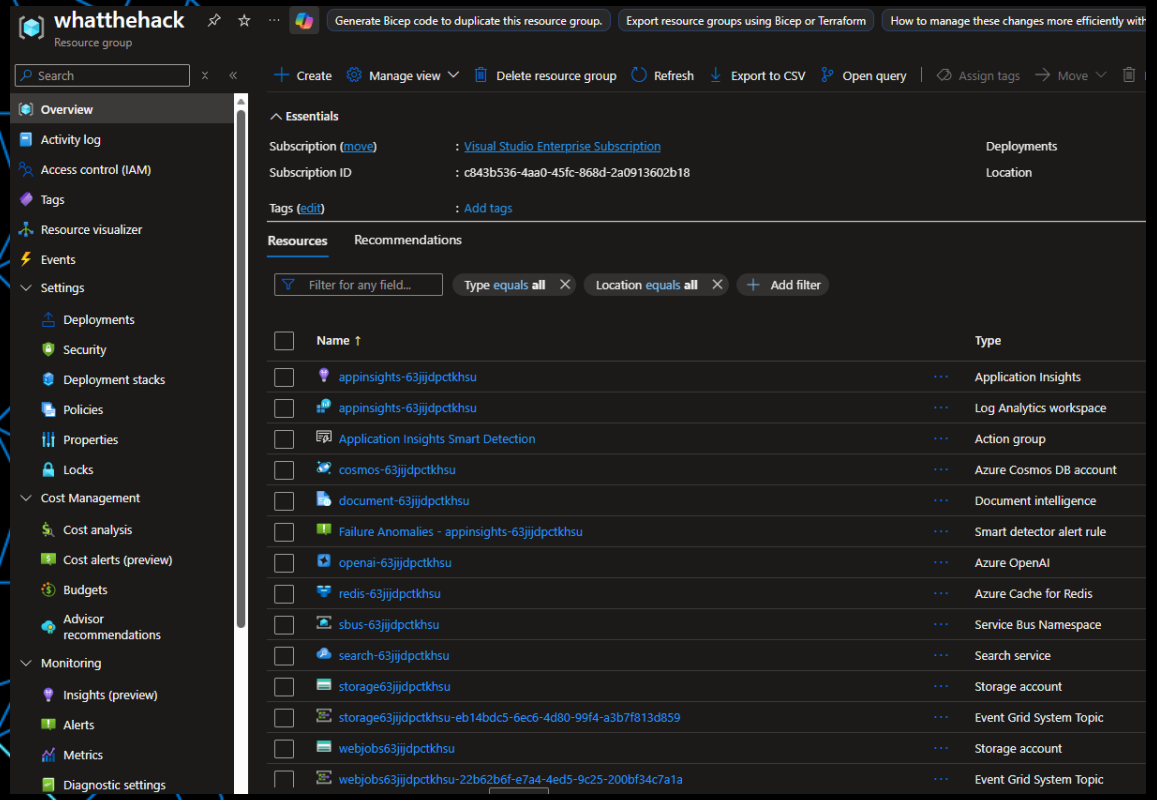



Fairfax Network
Browser
Web App Browsing
https://*-4200.app.github.dev
Development Activities
https://*.app.github.dev
Direct API Interactions
https://*-7072.app.github.dev
VS Code
Angular App
Function App
GitHub Cloud
File Uploads
Document Uploads
Search Retrieval
Message Queues
Embeddings and Chat Completions
Cache Store and Retrieval
Search Services
Service Bus
Cognitive Services
Cache Redis
Storage Accounts
Azure Cosmos DB
Index documents
Azure Cloud

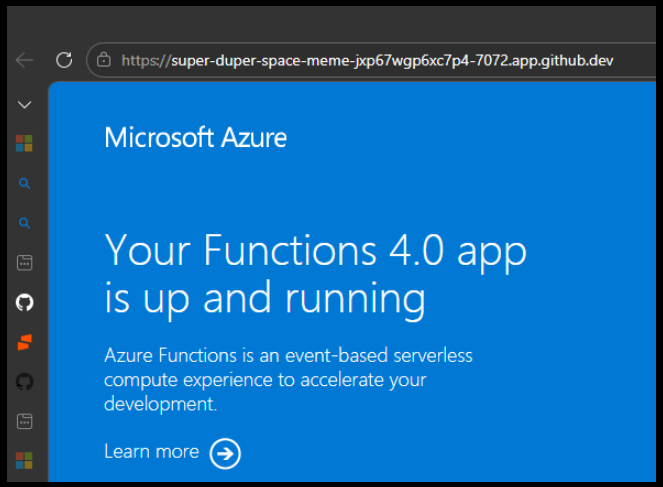


Challenge 00

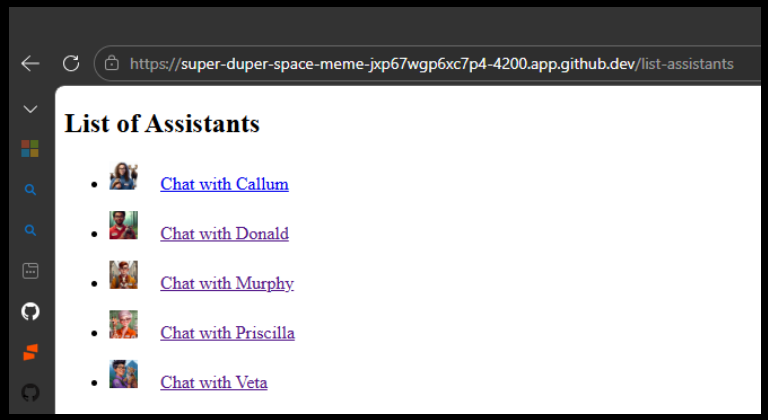
Challenge 00 Progress Check



Azure Portal with Deployed Infrastructure (14 items)



Function App

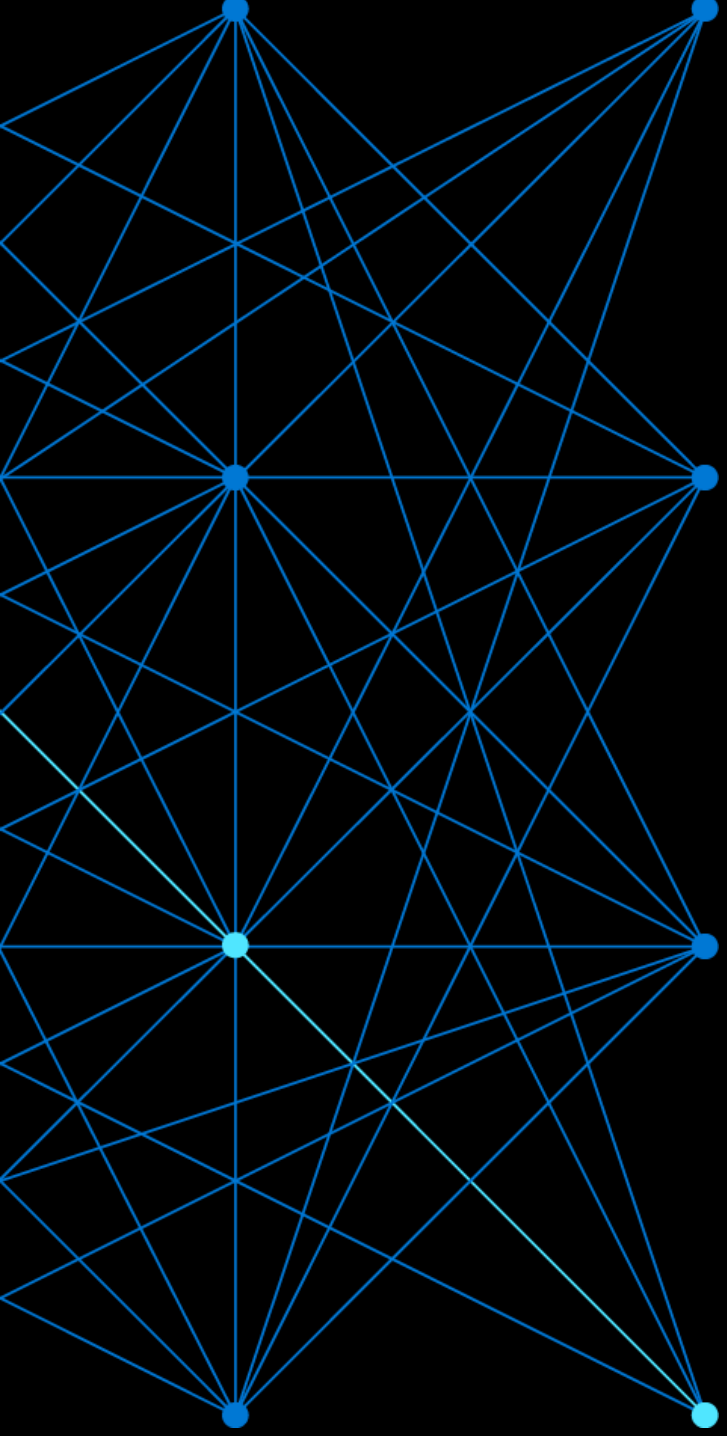


Angular App

Infrastructure as Code

- Declarative language
- Azure-specific (Terraform, AWS CloudFormation, Pulumi)
- Benefits
  - Repeatable
  - Disaster recovery
  - Cost savings
  - Documentation

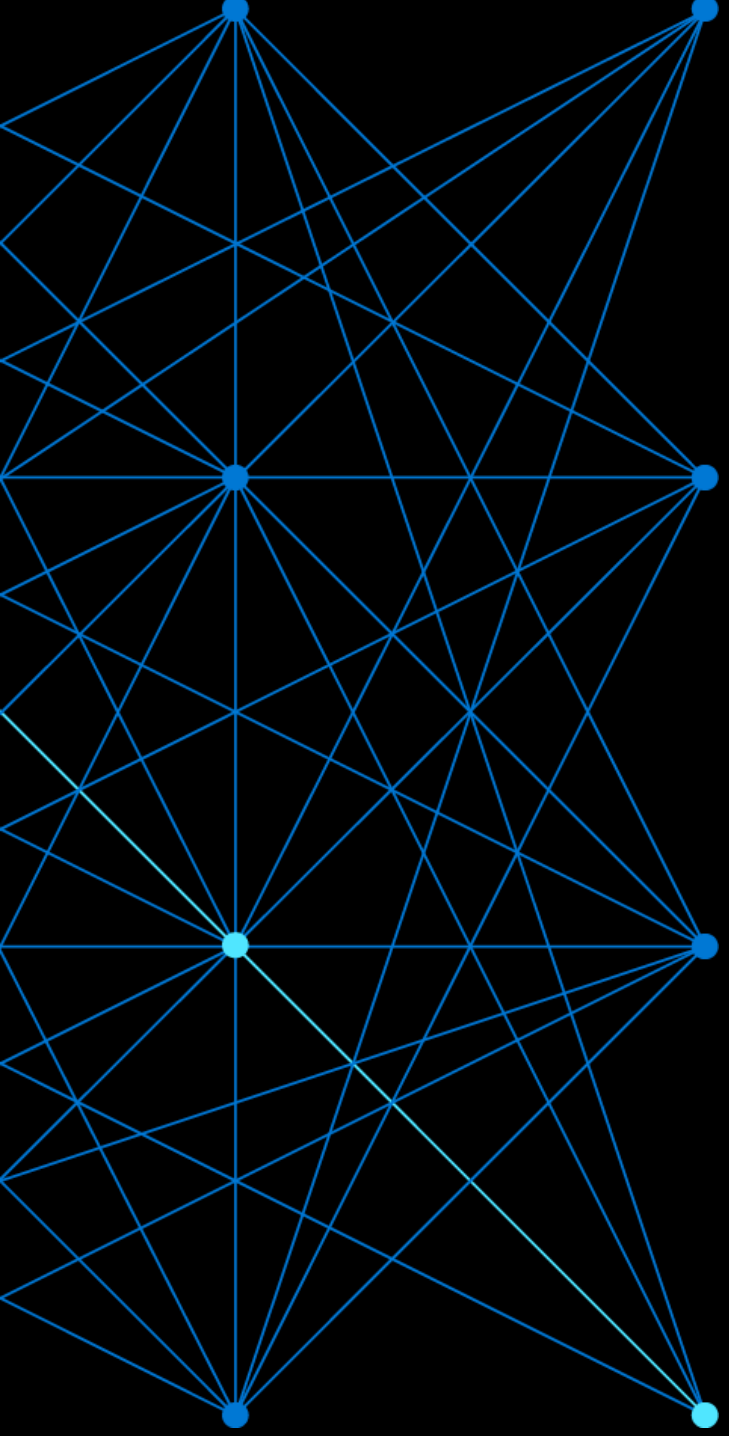
```
20
21      output redisPrimaryKey string = redis.outputs.primaryKey
22      output redisHostname string = redis.outputs.hostname
23
24      module serviceBus 'modules/servicebus.bicep' = {
25        name: 'serviceBusDeployment'
26        params: {
27          name: 'sbus-${suffix}'
28          location: location
29          queues: ['orange', 'lemon', 'grapefruit', 'tangerine']
30        }
31      }
32
33      output serviceBusConnectionString string = serviceBus.outputs.connectionString
34
35      module cosmos 'modules/cosmos.bicep' = {
36        name: 'cosmosDeployment'
37        params: {
38          name: 'cosmos-${suffix}'
39          databaseName: 'contoso'
40          location: location
41          containers: [
42            { name: 'yachts',          partitionKey: '/yachtId' }
43            { name: 'customers',       partitionKey: '/email' }
44            { name: 'reservations',    partitionKey: '/id' }
45            { name: 'examsubmissions', partitionKey: '/id' }
46          ]
47        }
48      }
```



Challenge 01

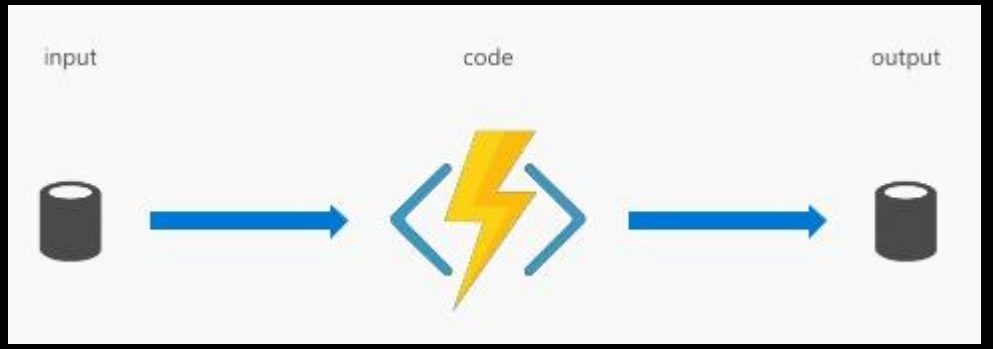
Discussion Points

- Azure CLI
- REST Client
- Function Bindings
- Azure CosmosDB, Storage, AI Search



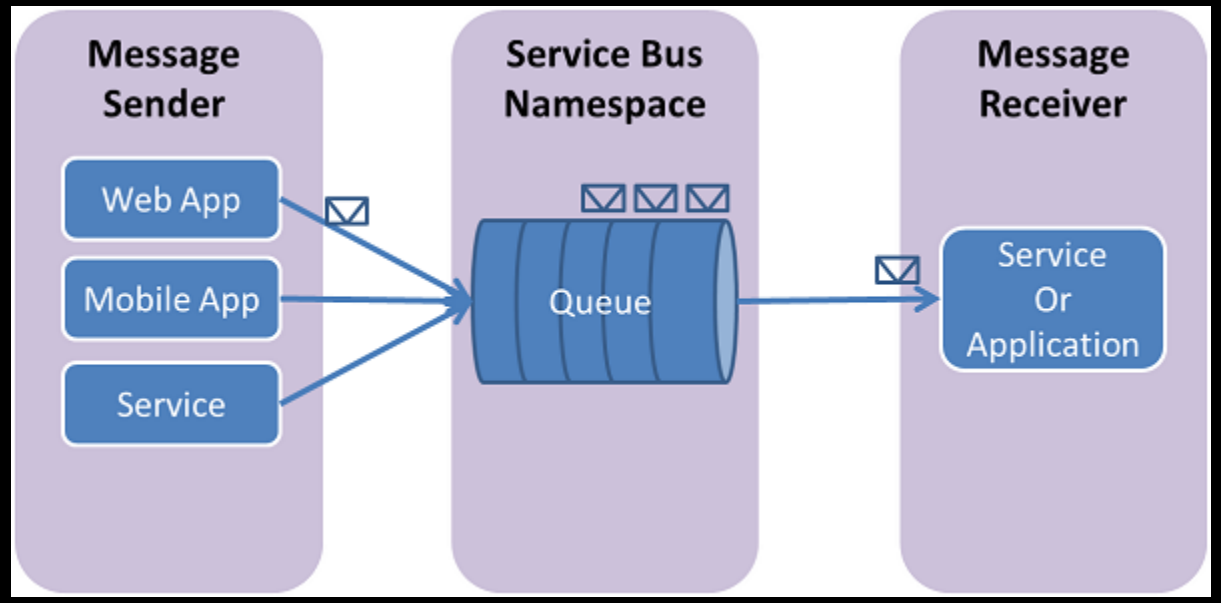
Azure Functions

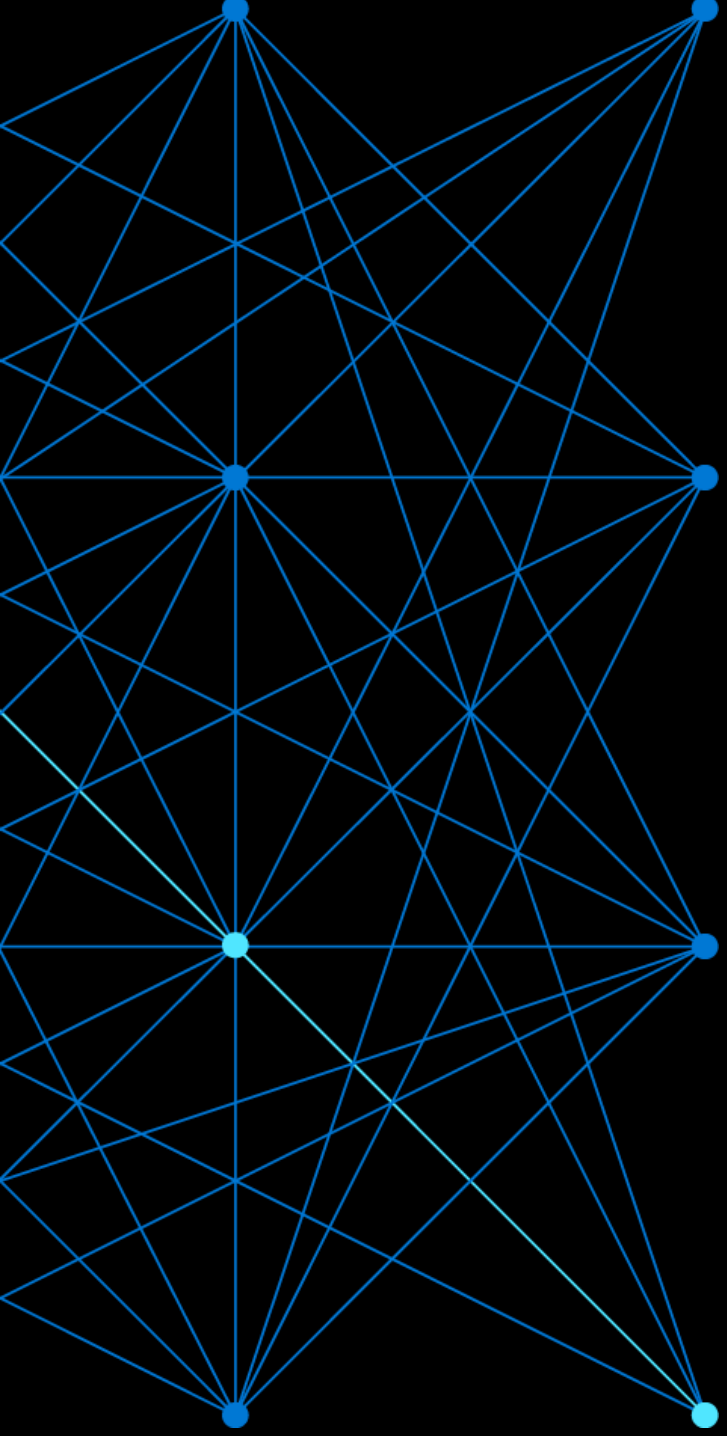
- Functions-as-a-Service
- .NET, Python, Javascript, Java
- Local emulator available
- Pay-per-execution or hosted plans
- Bindings for
  - HTTP
  - Queues
  - Databases
  - Timers
  - and more



Azure Service Bus

- Platform-as-a-Service
- Queues
- Topics and Subscriptions



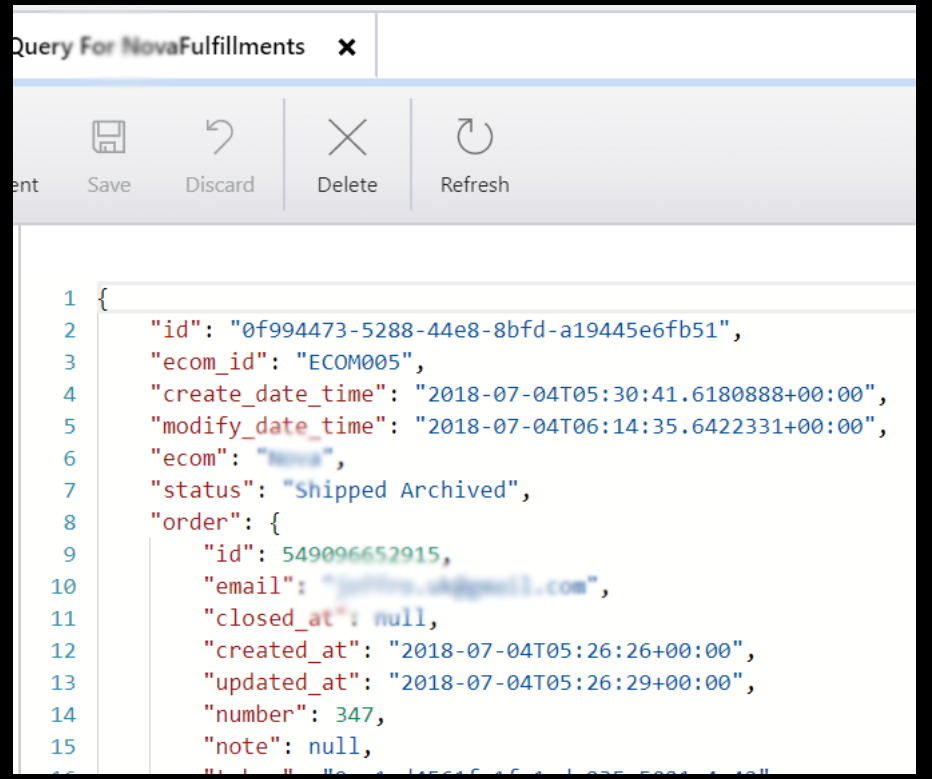


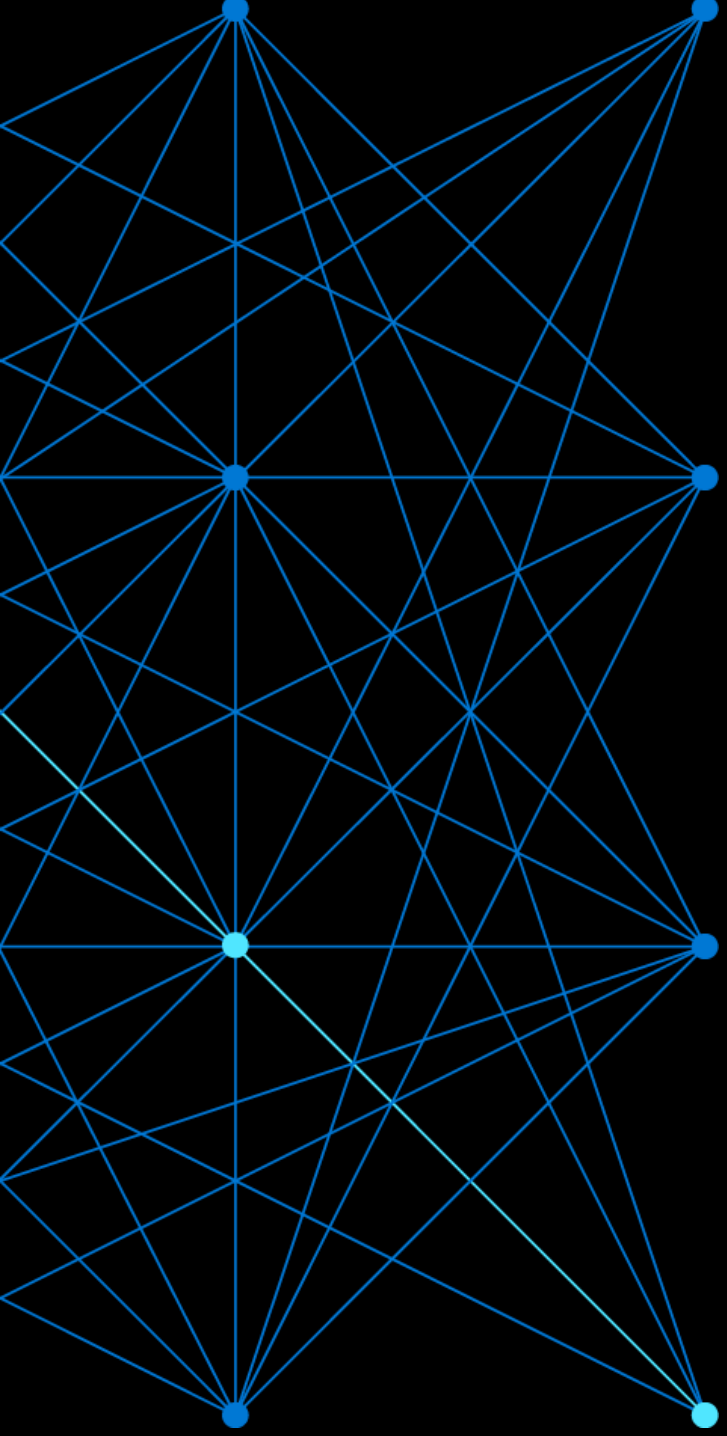
Azure Redis Cache

- Platform-as-a-Service
- In-memory database
- Ultra-fast performance
- Widely used for content caching

Azure Cosmos DB

- Platform-as-a-Service
- NoSQL / Document (JSON) database
- Queryable with SQL-like syntax
  - Mongo, Cassandra APIs
- Globally distributed
- Low latency for transactional workloads





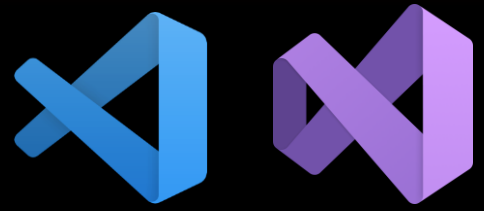
Challenge 02

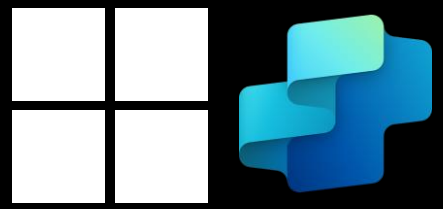
Discussion Points

- GitHub Copilot
  - Agent mode
  - Model selection
  - Context window
- Prompts
- MCP
  - Host, Client, Server

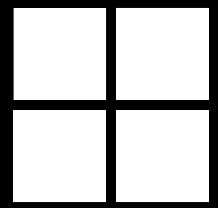
MCP is an open protocol that standardizes how applications provide context to LLMs.

Hosts





Clients



Servers

MCP Deep-Dive

Adapted from Mahesh Murag (Anthropic) AI Engineer session

MCP Client

Invokes Tools

Queries for Resources

Interpolates Prompts

MCP Server

Exposes Tools

Exposes Resources

Exposes Prompts

Tools

Model-controlled

Functions invoked by the model

- Retrieve / search
- Send a message
- Update DB records

Resources

Application-controlled

Data exposed to the application

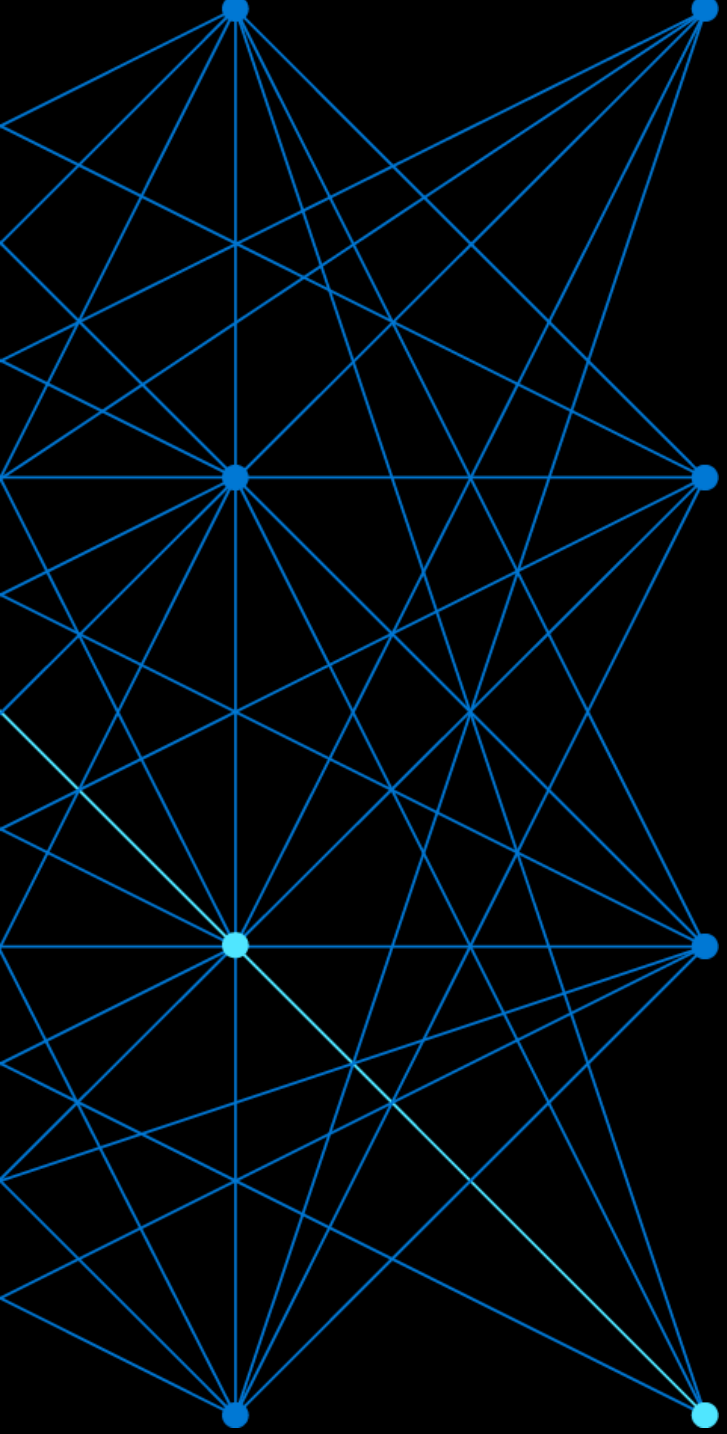
- Files
- Database records
- API Responses

Prompts

User-controlled

Pre-defined templates for AI interactions

- Document Q&A
- Transcript Summary
- Output as JSON



Challenge 03

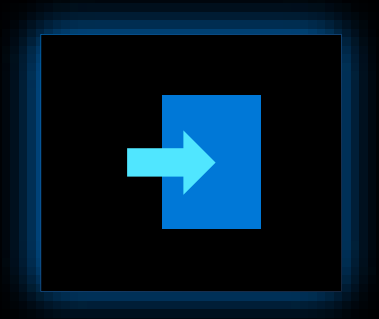
Discussion Points

- Document Intelligence
  - **Tip**: Extractor models take 10+ minutes each to train
- Service Bus and Messaging

How does document process automation work?

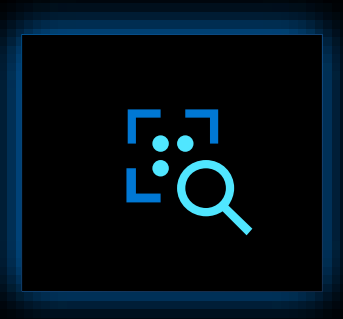
Azure Cognitive Services can help ingest, extract and enrich data

INGEST



Utilize Document Intelligence for digitizing printed or handwritten documents

EXTRACT

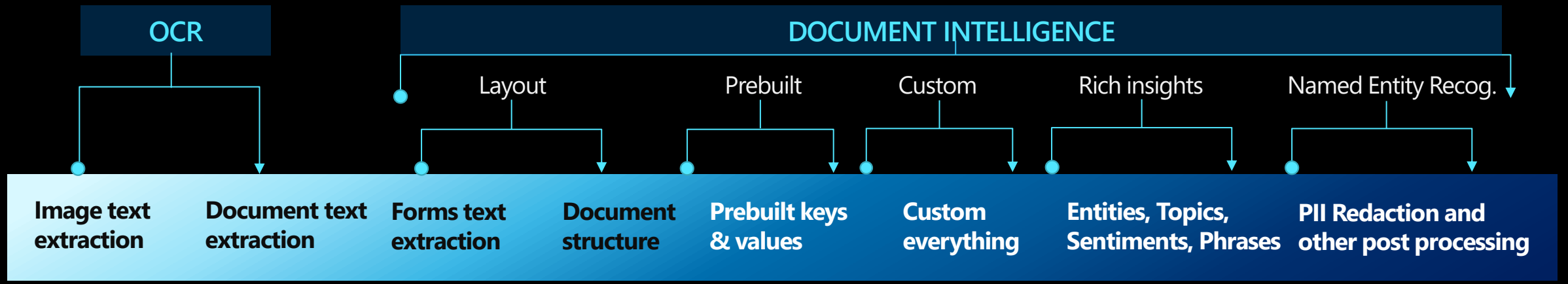


Use pre-trained or custom AI models for entity and key pair extraction with Document Intelligence

ENRICH

Use prebuilt AI services like Text Analytics or custom LUIS for documents models to enrich the data and make it relevant to your business needs

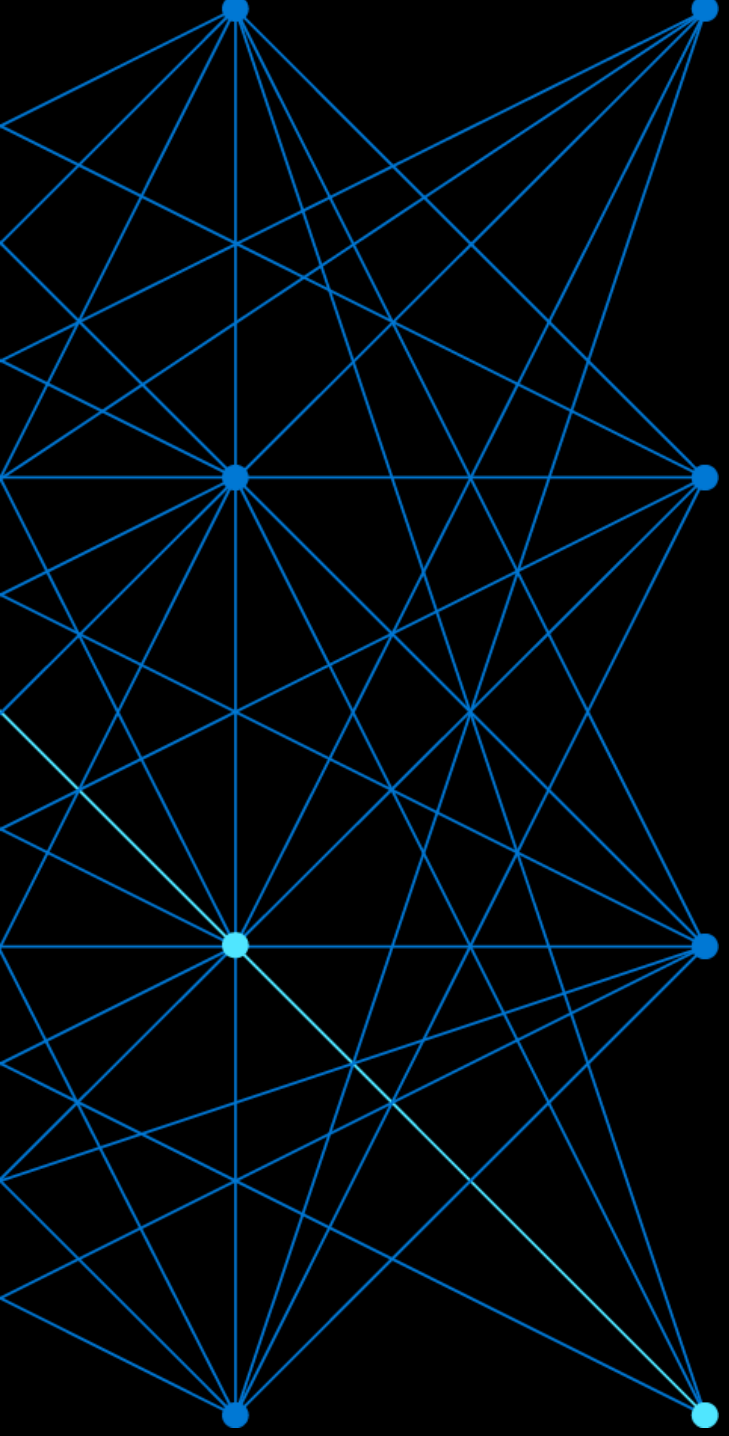
OCR and Document Intelligence



- ✓ Images
- ✓ Single pages
- ✓ Plain text

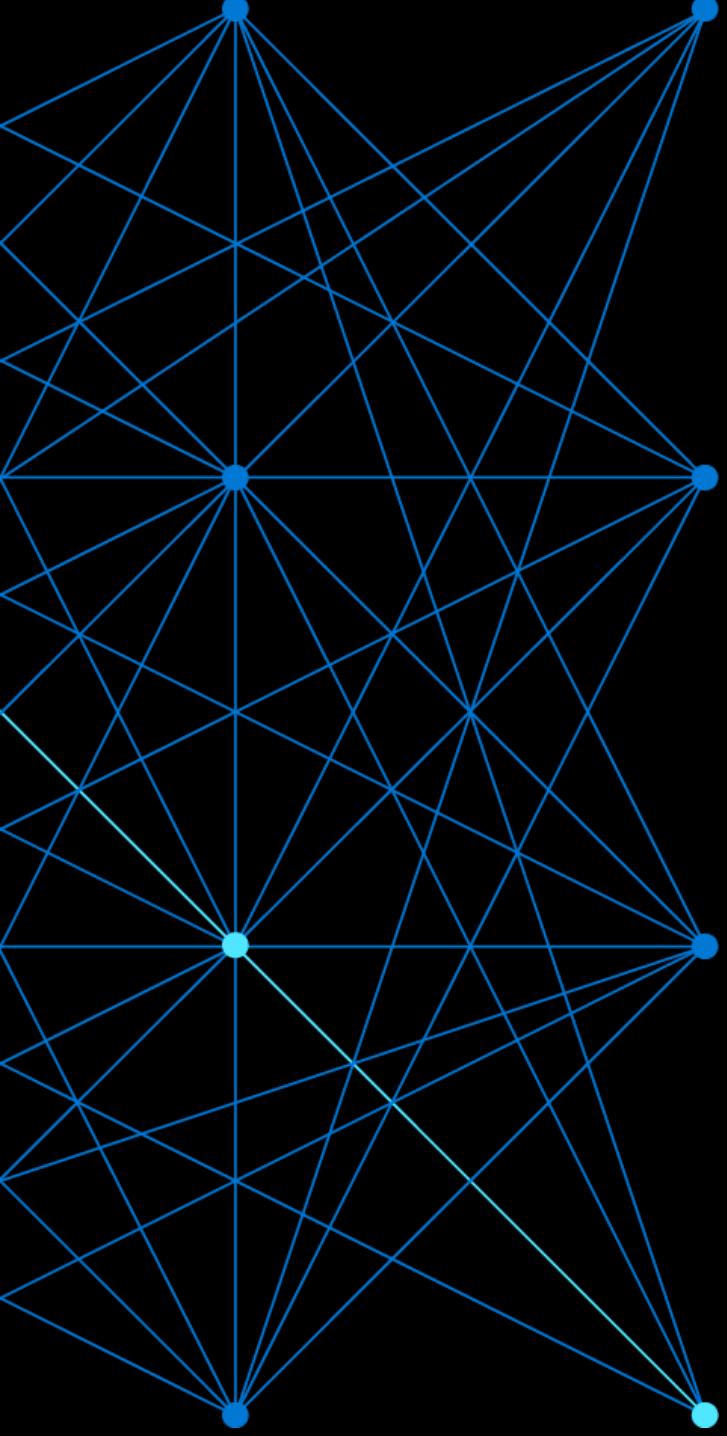
Discovery, Understanding, Evaluation, Adoption, Support

- ✓ Large documents
- ✓ Complex documents
- ✓ Rich insights
- ✓ Tables
- ✓ Checkboxes
- ✓ Pre-built capabilities
- ✓ Train your own model



Azure Service Bus

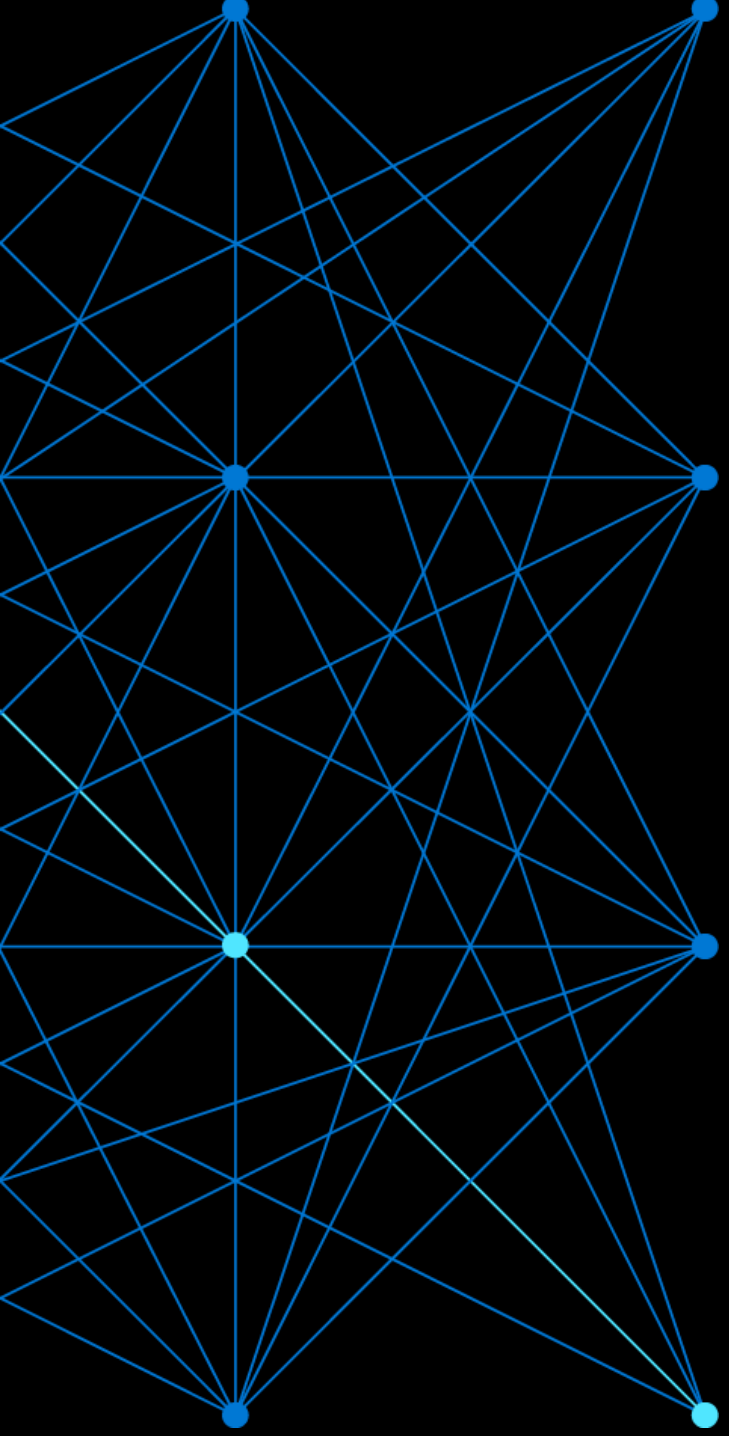
- Enables asynchronous communications
- Facilitates batching
- Helps with exception handling, DR, outages
- Can be used for scaling purposes



Challenge 04

Discussion Points

- Quota management
  - Via Azure OpenAI and external mechanisms



Challenge 05

Discussion Points

- Caching
- OpenAI pricing
- App Insights and distributed application monitoring