Microsoft Azure

Building an Enterprise-Ready Chatbot in One Day

Agnieszka Niezgoda
Maciej Drwięga
Tomasz Kopacz

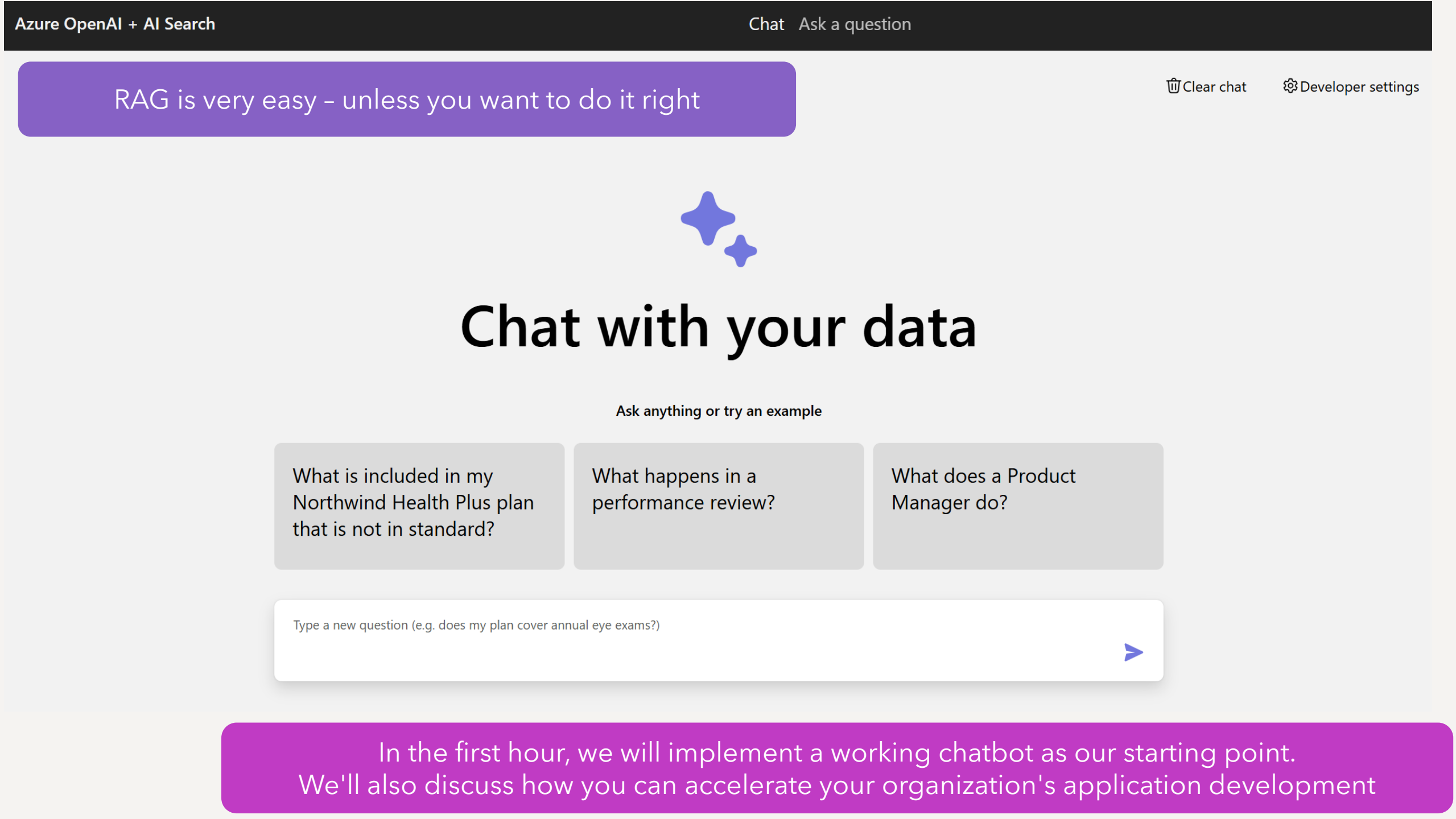
The aim of the workshop is to enable you to create applications that are:

 **modern,**

 **safe**

 **tailored to the needs of large organizations**

as efficiently as possible


Azure OpenAI + AI Search
Chat
Ask a question
RAG is very easy – unless you want to do it right
Clear chat
Developer settings
Chat with your data
Ask anything or try an example
What is included in my Northwind Health Plus plan that is not in standard?
What happens in a performance review?
What does a Product Manager do?
Type a new question (e.g. does my plan cover annual eye exams?)
In the first hour, we will implement a working chatbot as our starting point.
We'll also discuss how you can accelerate your organization's application development

Join us

Agnieszka Niezgoda

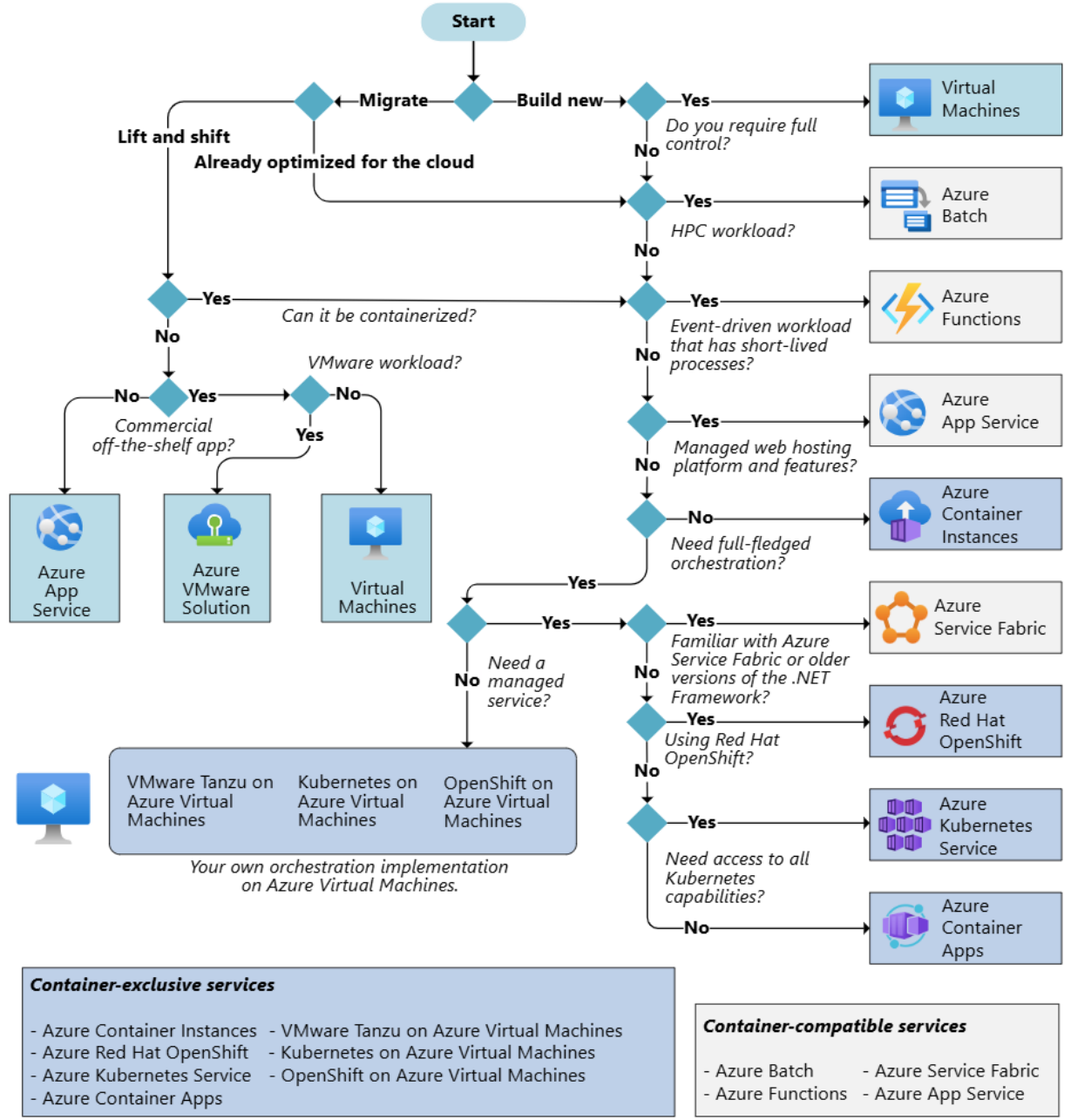
Maciej Drwięga

Tomasz Kopacz

1. Introduction. Applications on Azure

30 minutes

On Azure, you have several different application services to choose from



Container-exclusive services

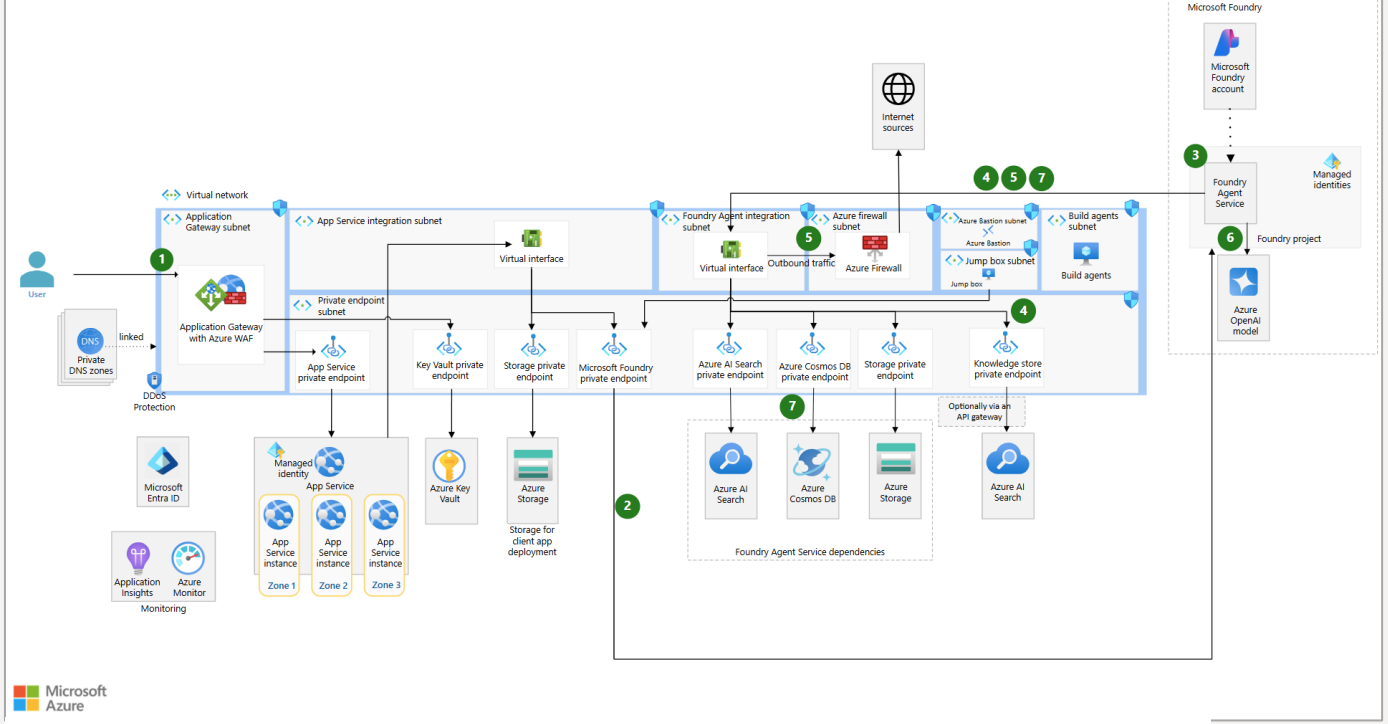
- Azure Container Instances
- Azure Red Hat OpenShift
- Azure Kubernetes Service
- Azure Container Apps
- VMware Tanzu on Azure Virtual Machines
- Kubernetes on Azure Virtual Machines
- OpenShift on Azure Virtual Machines

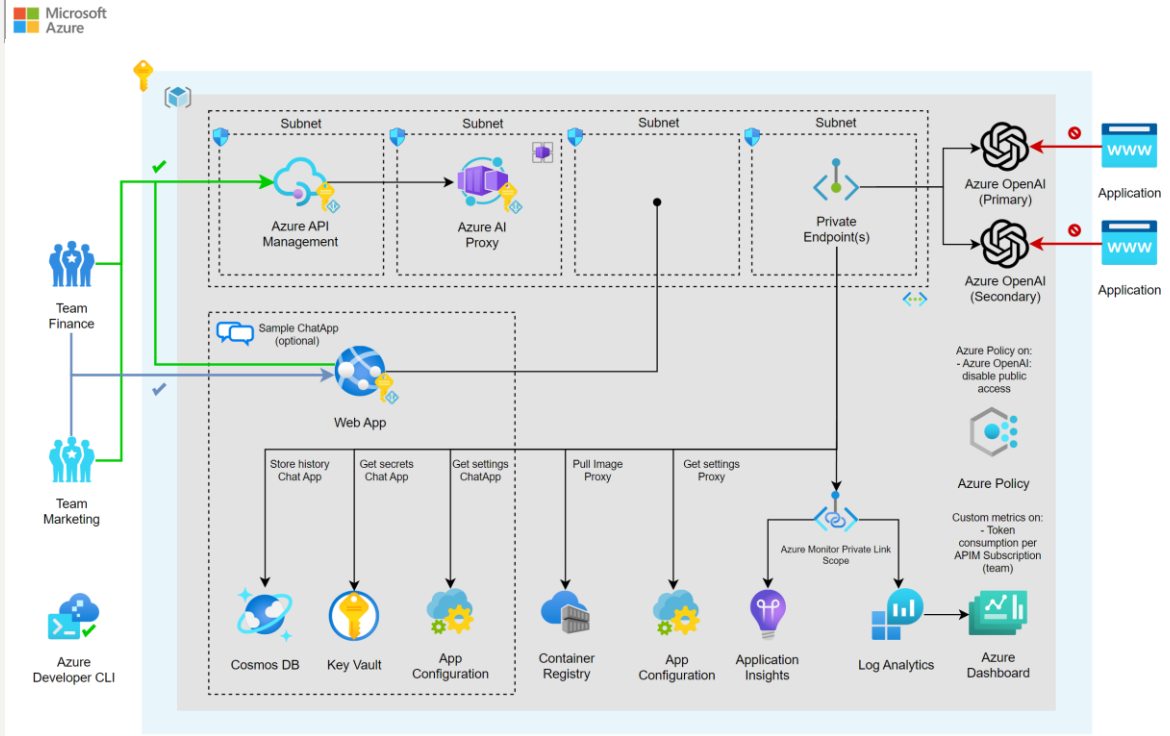
Container-compatible services

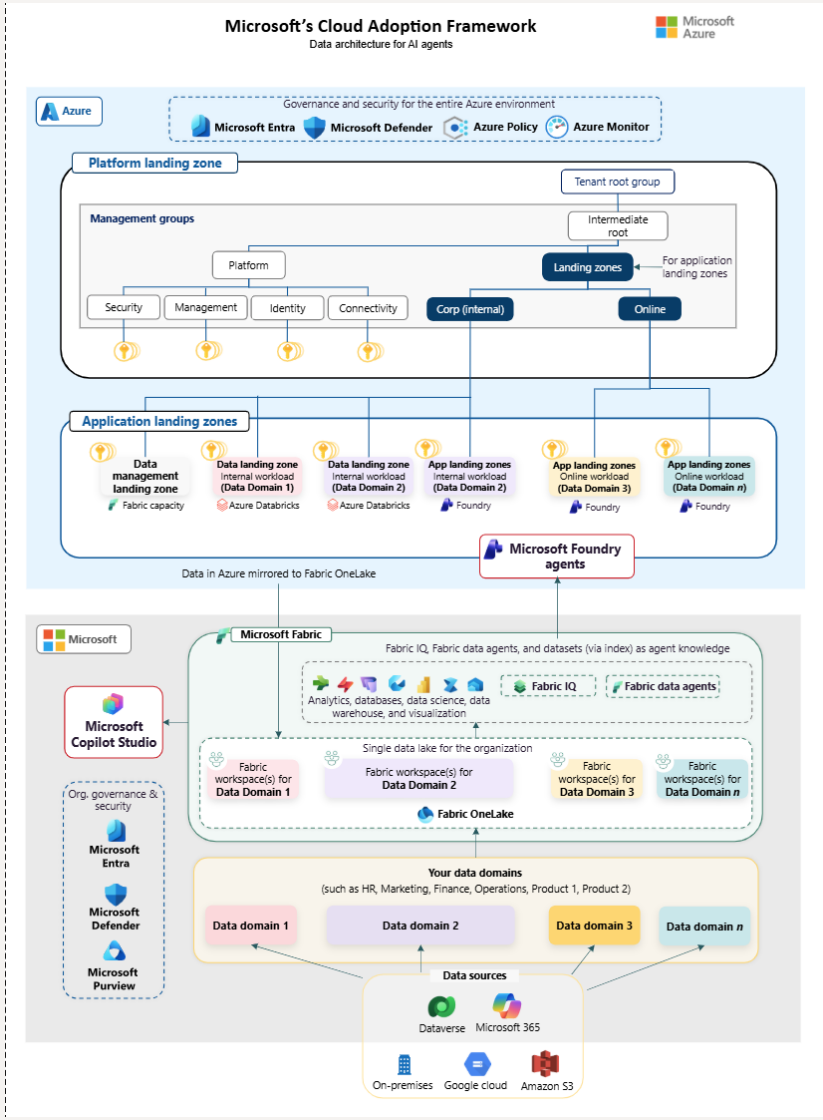
- Azure Batch
- Azure Functions
- Azure Service Fabric
- Azure App Service

Microsoft Azure

Common application architectures on Azure







„Czym się aplikacja enterprise różni od hello world"

Reliability & Fault Tolerance

Assume everything can fail;
Fault-tolerant architecture;
Layered recovery approach;
Quantitative reliability modelling.

Scalability & Performance

Elastic scaling;
Use managed services (PaaS);
Performance tuning & measurement;
Business metrics and tenant-health signals.

Security & Governance

Enterprise-grade security and governance frameworks;
RBAC, data governance, auditing, compliance;
Guardrails for trustworthy operations

Observability & Operational Excellence

Comprehensive logging, tracing, and telemetry;
Automated operations;
Telemetry-driven health scoring

Data Management & Integrity

Multi-region data replication strategies;
Clear data models;
Secure data access patterns.

Maintainability & Modularity

Well-structured, modular architecture;
Evolvability ;
Use of standardized services and frameworks.

Foundry i APIM w architekturach AI

API Management:

- „AI Gateway" de facto
- Kontrola serwerów MCP / A2A
- Tłumaczenie na API zgodne z „OpenAI"
- Rozkład obciążenia (LB, round-robin)
- Centralny widok na używane tokeny
- **Semantic Caching**
- (throttling, autoryzacja, logowanie, …)

AI Foundry:

- Źródło modeli (GPT-x, LLAMAx, Claude)
- Finetuning
- Monitoring
- Też: platforma do Agentów (technicznie REST API)
  - Tu (można, ale niewiele doda); też Playground
- Historia: AOI, AI Project, AI Foundry

```
azd env set USE_AI_PROJECT true
azd provision
```

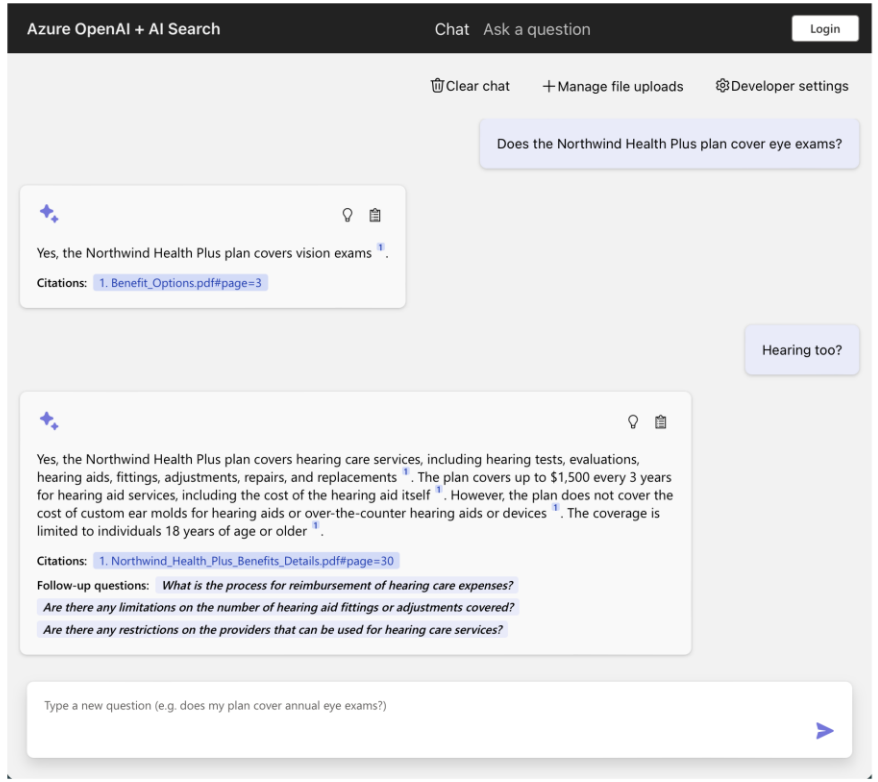
Agenda of today's session

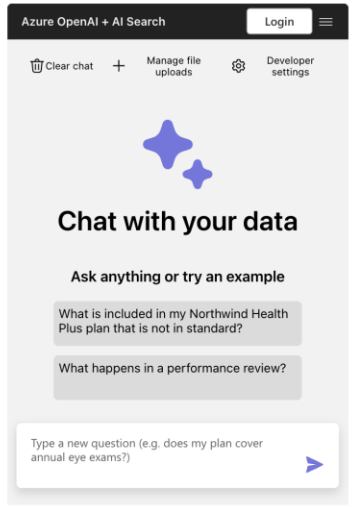
1. We start with the implementation of the project with Agnieszka's commentary
2. Github: Github Actions i Copilot
    1. (Comments about azd and how it works)
    2. Configuration of automatic deployment
    3. Enabling mechanisms to control the quality and security of the code
    4. How Copilot helps you understand code
    5. MCP server configuration for Github and ordering Coding Agent to build an alternative UI (e.g. Blazor)
3. Then – new functionalities (modifications to the implemented project)
    1. Changing the chunk and search parameters
    2. Adding Web and Sharepoint source search in addition to AI Search index
    3. Adding a validator
    4. Additional activity logging
    5. Logging using Entra
4. Finally, remarks on infrastructure and hard security

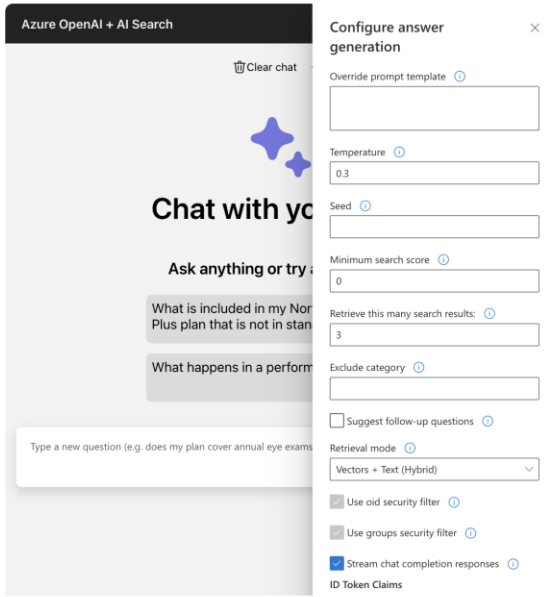
2. Accelerator: functionalities and implementation

1 h

RAG is used especially in on-your-own-data scenarios



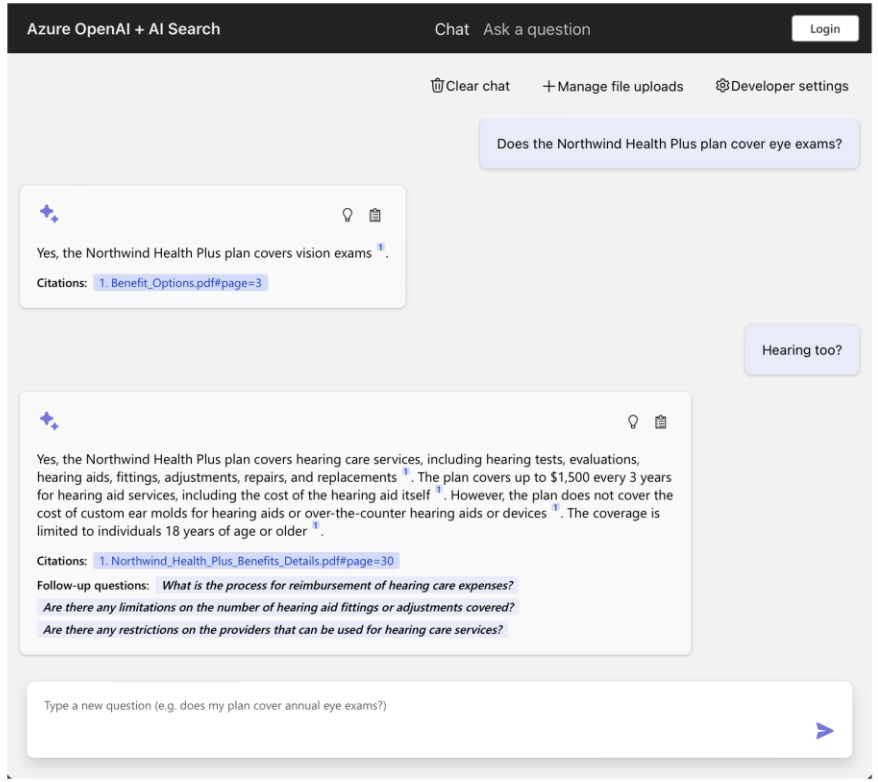


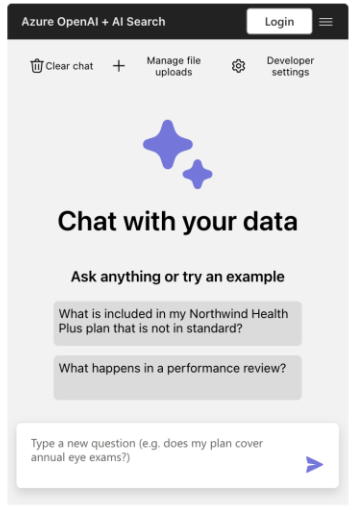


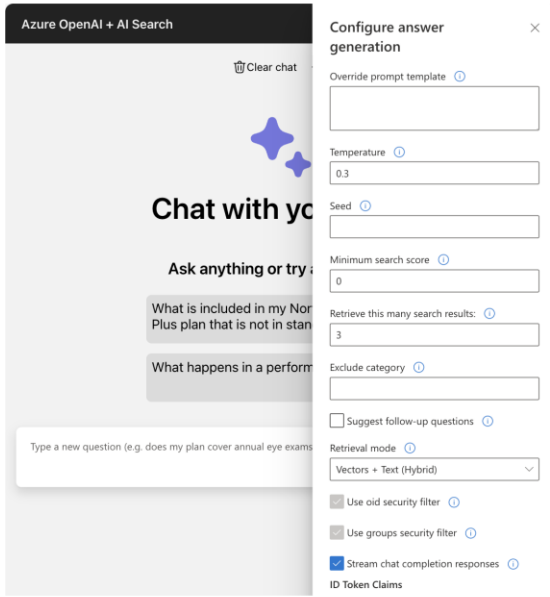
Accelerator implementation

1. Repos for the organization:
   - https://github.com/MH16-App-deployment-on-Azure/base-app
   - 00TKScenarios\00-Github-Start-Codespaces\00-Github-Start-CodeSpaces.md
2. Users chat-user-yz | rg-azureclubworkshopchat-yz | chat-user-yz_mhent
3. Fork
   - U nas: https://github.com/MH16-App-deployment-on-Azure/base-app
   - (bazowy) *https://github.com/Azure-Samples/azure-search-openai-demo*
4. Open the forked repo in CodeSpace. We'll be working in CodeSpace – it's a bit more complex to configure the app locally
5. We are using:
   - Domain: [domain name]
   - TenantID: [...]
   - Subscription ID: [...]
6. Envir name and RG: chat-[first letters of your name and surname e.g. AN]
7. Follow the README file to deploy the app – it should take around 20 minutes
   - Helper: 00TKScenarios\00-Github-Start-Codespaces\00-Github-Start-CodeSpaces.md 00TKScenarios\01-GithubActions\01-GithubActions.md
   - Important: **azd env set AZURE_RESOURCE_GROUP pl-tmp1**

RAG is used especially in on-your-own-data scenarios

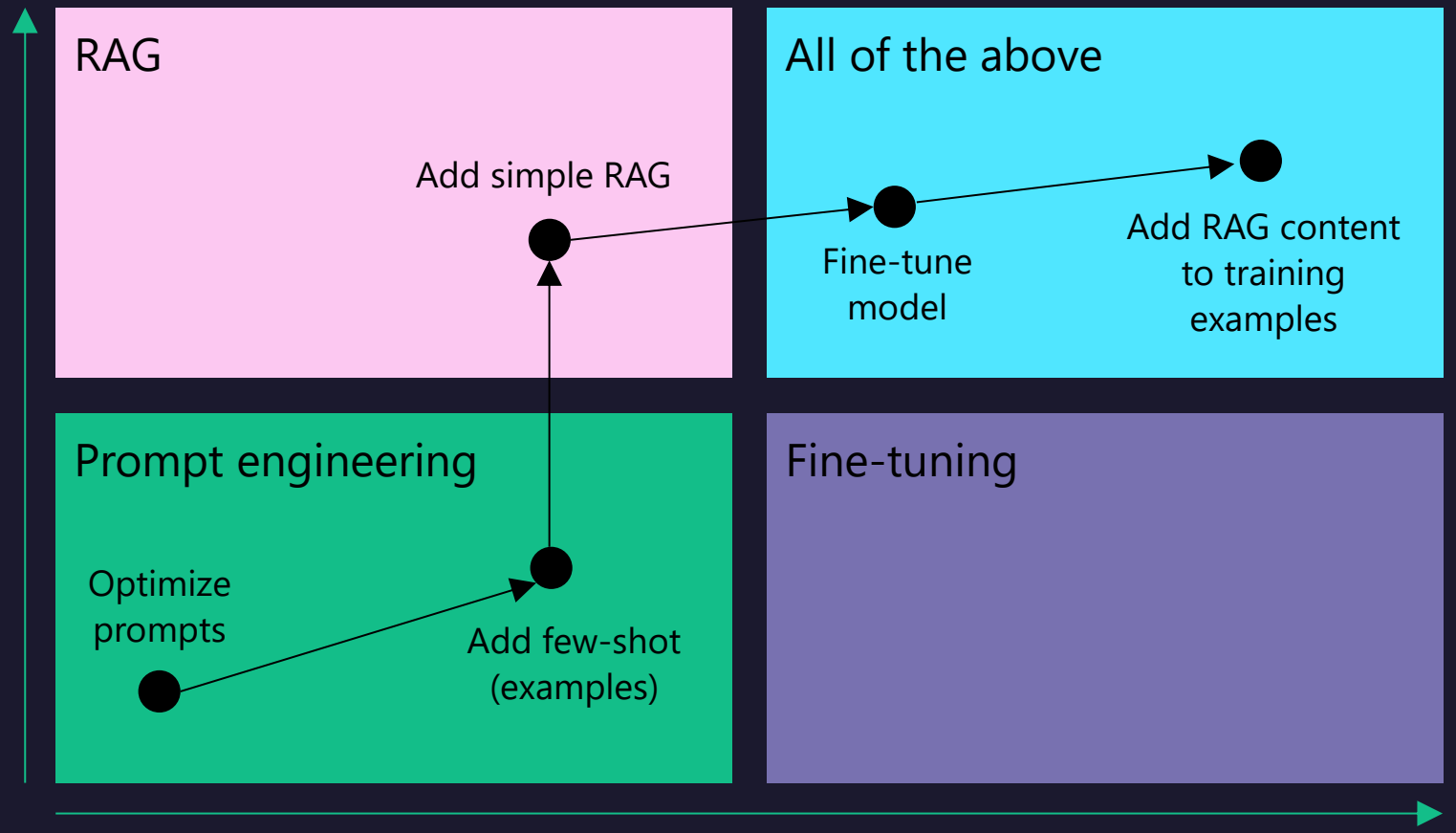






What do you miss in this app to make it work in your use cases?

The customization approach should depend on your goal



Problem: generative AI doesn't have context for your data

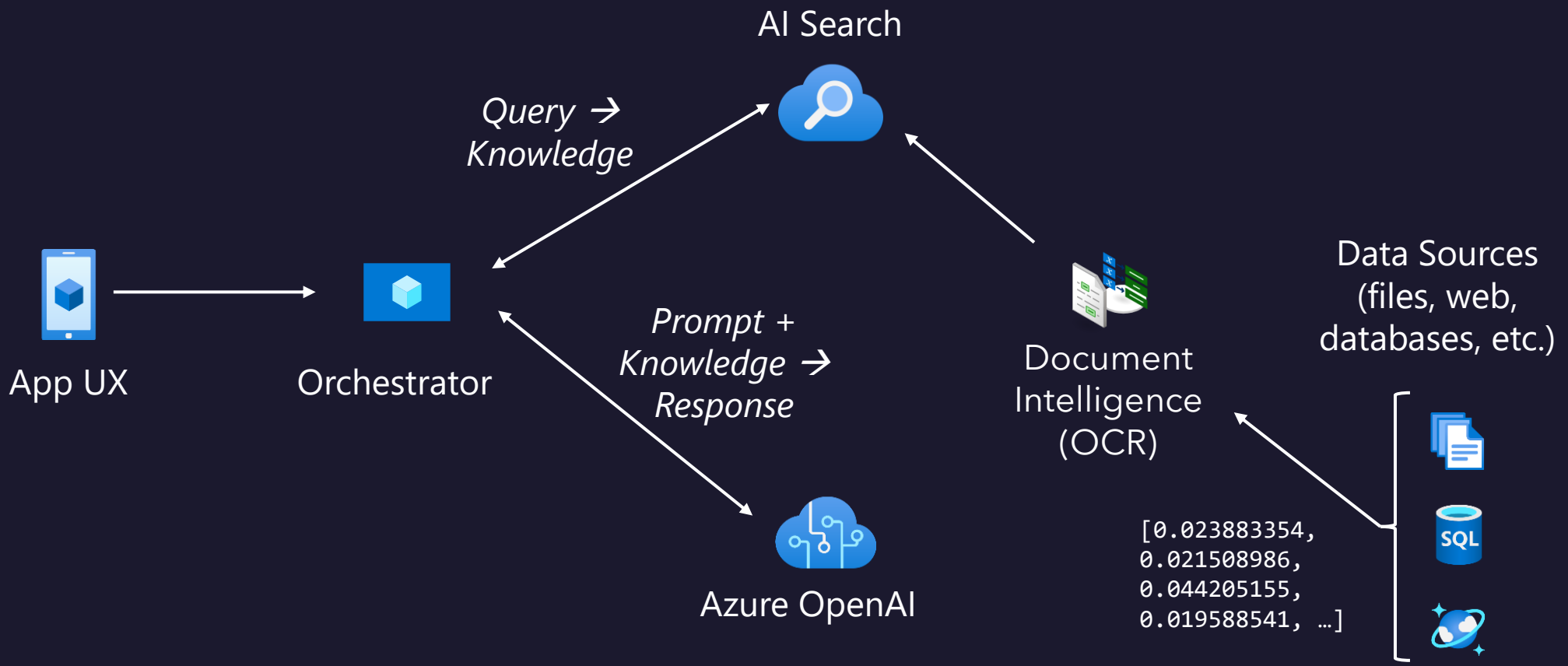
Prompt

Does my health plan cover annual eye exams?

Response

I'm an AI language model and don't have access to specific information about your health plan

RAG (retrieval augmented generation) is used especially in on-your-own-data scenarios



Typical RAG architecture – things to think about

- Frequently asked questions should be cached?

Logging & monitoring
- User feedback (e.g. thumbs up/down)
- Where to keep the conversation flow
- Conversation statistics

- Chunking method
- Indexing method and model used
- The way of retrieval (agentic?, FoundryIQ? How many chunks?)
- Semantic reranker?

AI Search

Zapytanie do źródeł → Wiedza

Application knowledge sources

- Pdfs, Word documents
- Charts, images
- Databases
- Web Resources

- What's the process of supplying sources – especially important when switching to production
- What preprocessing of sources is needed? E.g. creating diagram descriptions after adding them to sources?
- Examples of documents and pages

User interface

Orchestrator
- Answers to questions about documentation
- Configuration

Prompt + wiedza ze źródeł → odpowiedź

Azure OpenAI

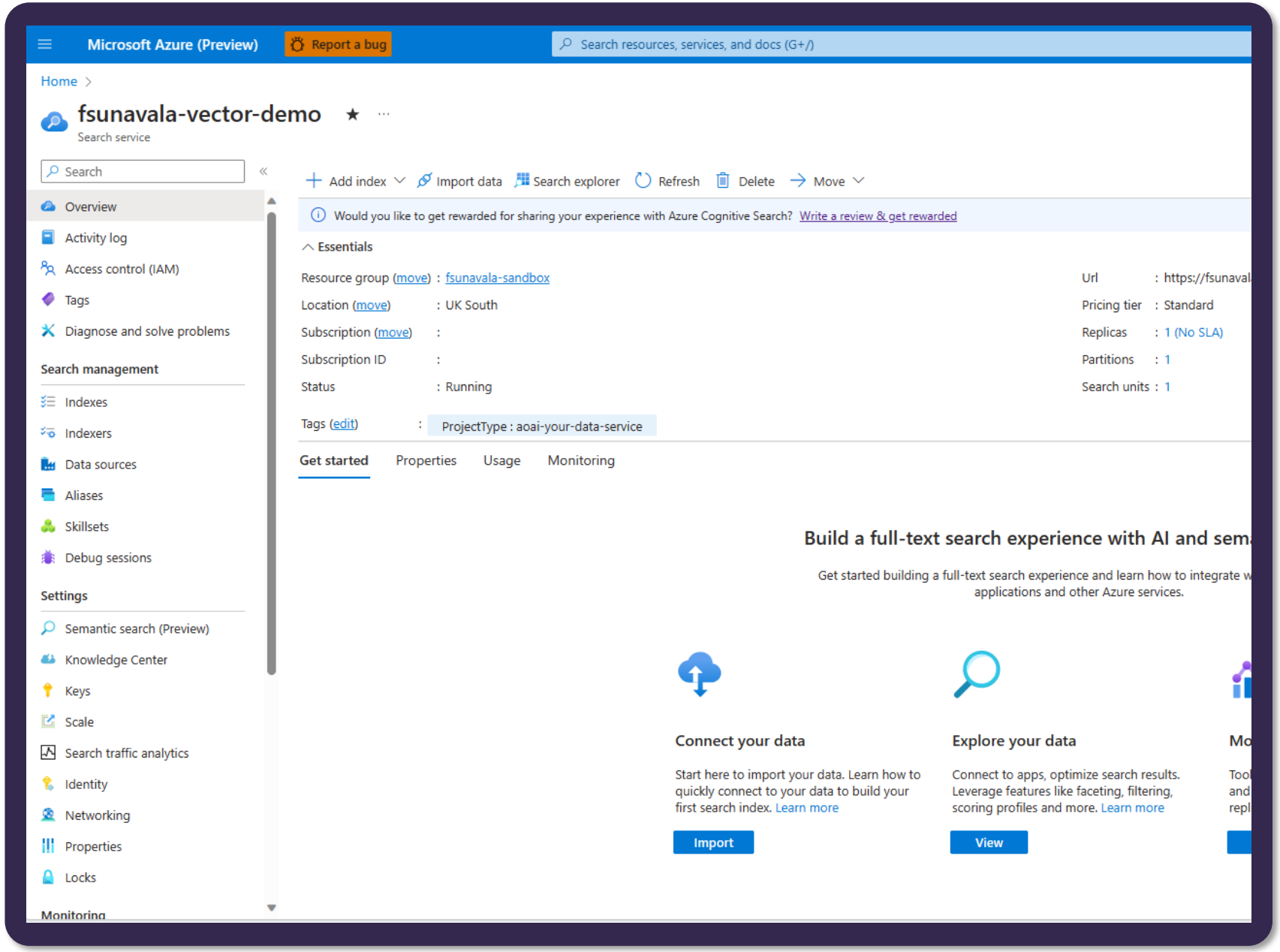
- What interface? Chatbot? Voice bot? Search engine? A bot that understands context?
- Possibility to add new documents by the user?
- Security and authentication

- Model Instructions: Style, To-Do's and Don't's
- How much context should I keep?
- Validating the answer?
- Content filters and other security checks?
- Adding agents and access to MCP servers/functions?

Azure AI Search

- Platform-as-a-Service
- Semantic search
- Management free
- Keyword search
- Faceting
- Language analyzers
- Geospatial support
- Suggestions/auto-complete
- Customizable scoring
- Proximity search
- Synonyms
- Cognitive skills

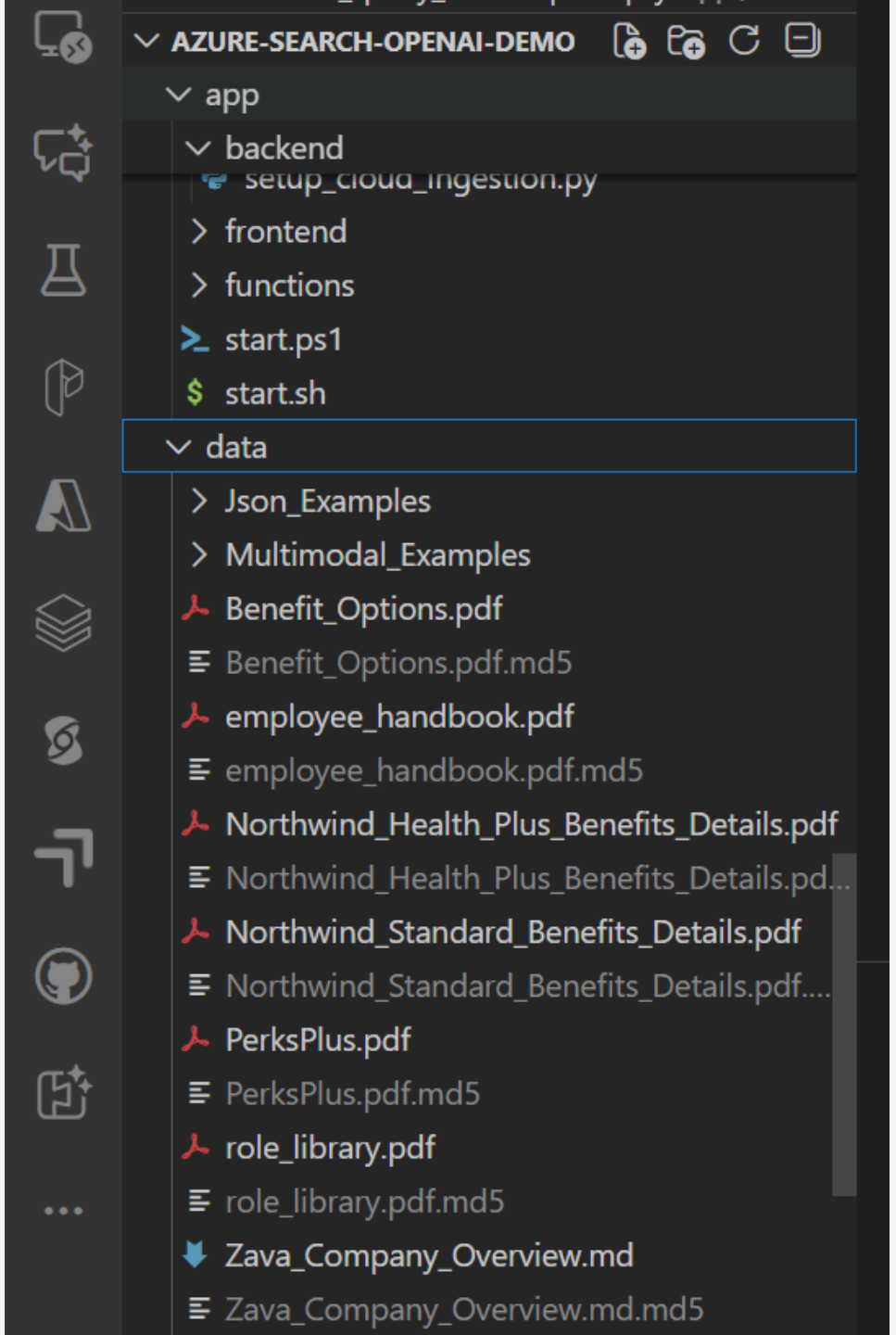
  etc.



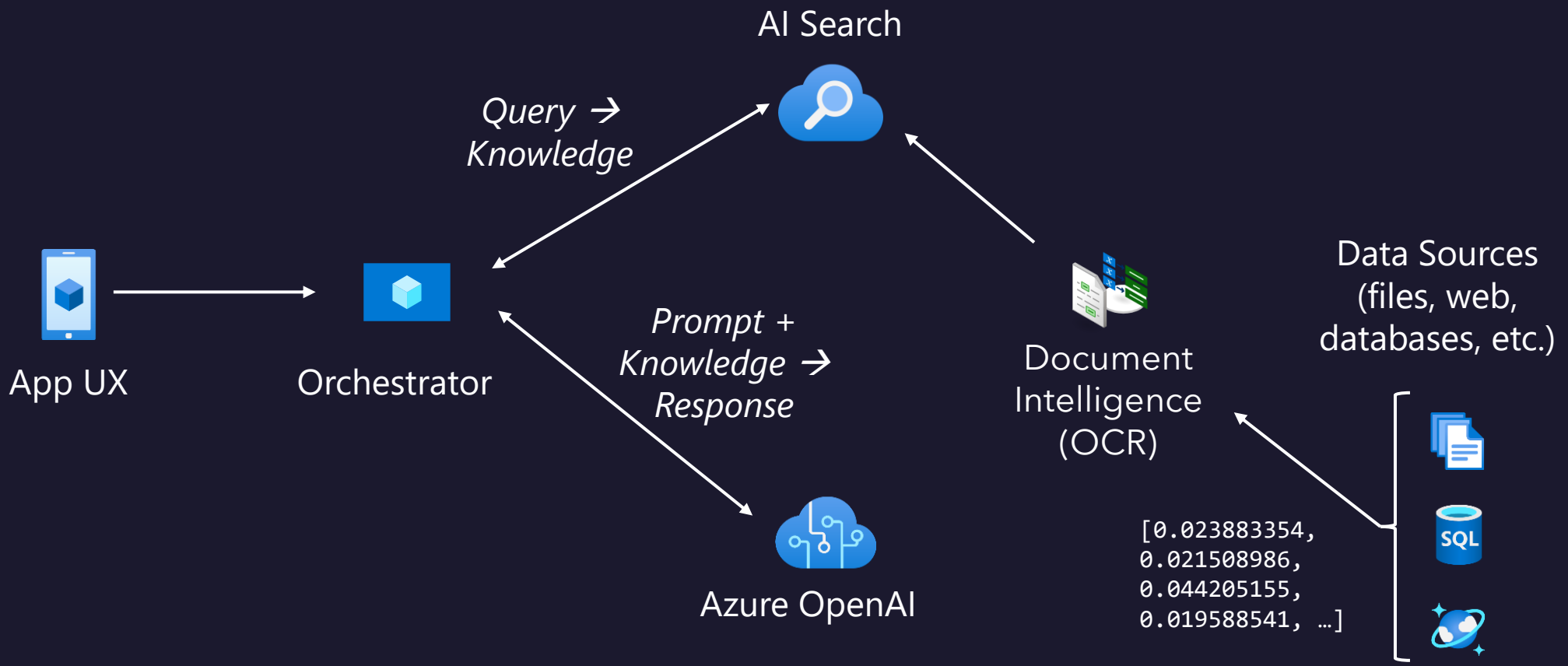
Default values in the accelerator

1. What infra to create: infra > main.bicep
2. What was created: .azure > [project name] > .env

The data sources are listed in the data folder



RAG (retrieval augmented generation) is used especially in on-your-own-data scenarios



backend>prepdocslib>textsplitter.py defines chunking

The chunking method is defined in textsplitter.py. This file contains:

1. **Base class** TextSplitter (lines 14-23) - Abstract base class for splitting pages into chunks
2. **Main chunking implementation** SentenceTextSplitter (starts around line 194) - The primary chunking class that:
   - Splits pages into smaller chunks for embedding models
   - Has configurable max_tokens_per_section (**hard** token limit for chunks), default 500 tokens
   - Uses DEFAULT_SECTION_LENGTH (soft character limit for chunks, can be exceeded by up to 20%; figures are exempt from this limit) of 1000 characters
   - Implements 10% overlap between chunks (DEFAULT_OVERLAP_PERCENT)
   - Intelligently splits at sentence boundaries or word breaks
3. **Helper classes and constants**:
   - ChunkBuilder (lines 131-191) - Accumulates text fragments respecting size limits
   - Word breaks and sentence endings for standard and CJK languages

In the version of the app you have installed, you have 2 approaches

Embedding Generation (lines 877-900)

compute_text_embedding(): Creates text embeddings using Azure OpenAI

compute_multimodal_embedding(): Creates image embeddings using Azure AI Vision

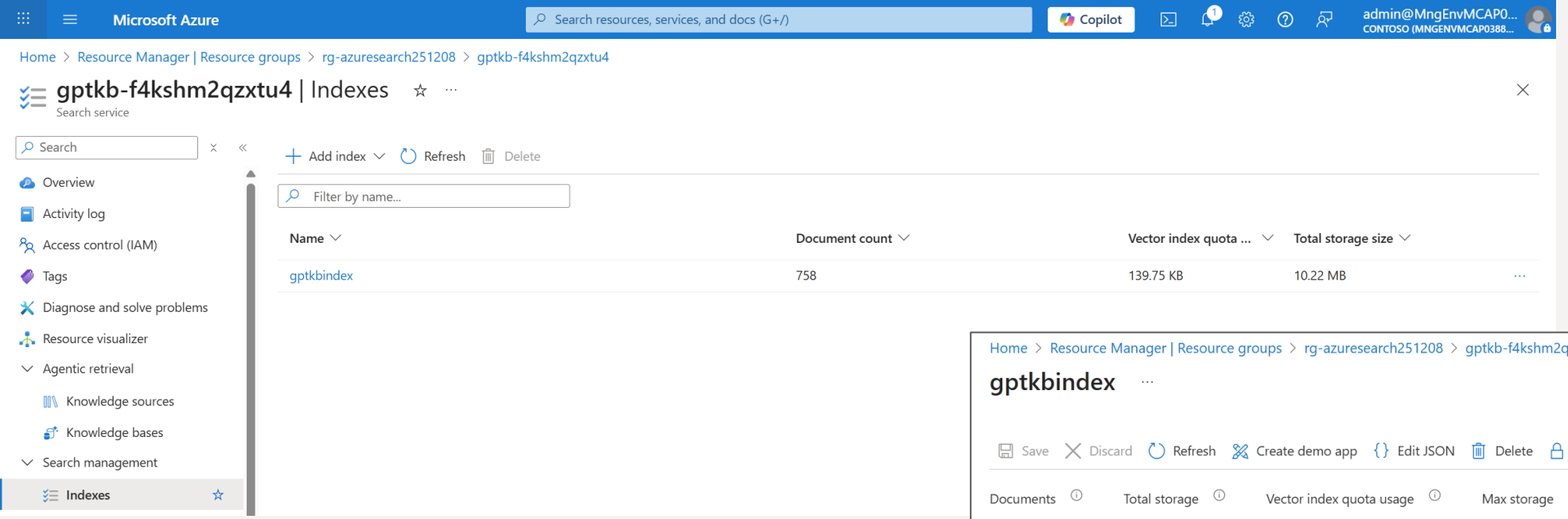
In the version of the app you have installed, you have 2 approaches

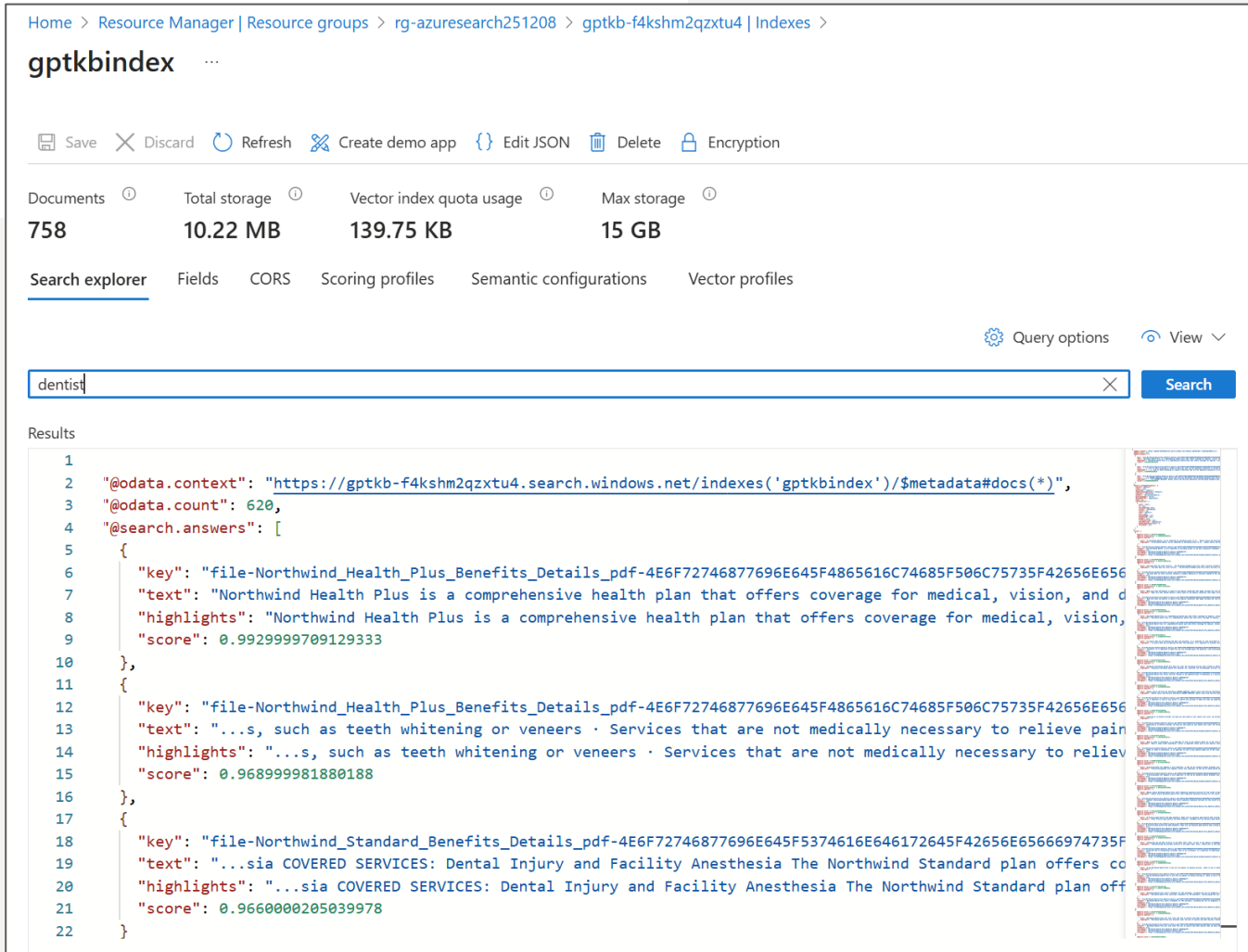
A. Ask Approach: retrievethenread.py: Simple retrieve-then-read pattern (lines 164-233):

1. Takes user query directly
2. Generates embeddings if using vector search
3. Calls search() method
4. Returns documents and metadata

B. Chat Approach ***let's focus here*****:** chatreadretrieveread.py: Multi-step retrieval with query rewriting (lines 339-437):

1. **Query Rewriting**: Uses GPT to transform chat history + user query into optimized search query
2. **Embedding Generation**: Creates embeddings from the rewritten query
3. **Search Execution**: Calls search() method with optimized query
4. **Returns**: Documents with thought steps showing the process





Embeddings (vectors) are uploaded to the index

You can view the sources in AI Search

What happens when a user asks a question?

User Query
↓
[Optional: Query Rewriting via GPT]
↓
[Generate Embeddings]
├→ Text Embedding (Azure OpenAI)
└→ Image Embedding (Azure AI Vision)
↓
[Azure AI Search Query]
├→ Text Search (BM25)
├→ Vector Search (HNSW)
├→ Hybrid Search (combines both)
└→ Semantic Ranker (L2 reranking)
↓
[Filter & Score Results]
├→ Apply security filters
├→ Apply minimum score thresholds
└→ Extract semantic captions
↓
[Return Documents to Be Used by the Model When Addressing User's Question]

Retrieval strategies in Azure AI Search

Keyword search

- **For exact, plain text matches**
- "Vocabulary gap" in Q&A systems like Copilot

Vector search

- **For conceptual similarity, or underlying meaning**
- Weak performance on exact matches (like a product ID or code)

Hybrid search

- **Best of both vectors and keywords**
- Brings more accurate responses across various scenarios

Search re-ranking

- **Scores and ranks all retrieved documents by relevance**
- Reranking runs after performing search strategy (can't retrieve information)

Vector search is good. Hybrid – even better

	Full-text search (BM25)	Pure Vector search (ANN)	Hybrid search (BM25 + ANN)
Exact keyword match	✓	✗	✓
Proximity search	✓	✗	✓
Term weighting	✓	✗	✓
Semantic similarity search	✗	✓	✓
Multi-modal search	✗	✓	✓
Multi-lingual search	✓	✓	✓

Vector and full-text search have different advantages

Leveraging the Strengths of Lexical and Semantic Approaches in Retrieval

Discrete (Lexical) Representations

- Advantages:
  - Exact matching
  - Precise control and easy explainability
- Limitations:
  - Struggle to capture nuances in language
  - Limited understanding of conceptual similarity

Dense (Vector/Semantic) Representations

- Advantages:
  - Capture conceptual similarity
  - Better understanding of language nuances
- Limitations:
  - Not built to match exact terms
  - Reduced explainability compared to discrete

Bottom Line: Achieve optimal recall by leveraging the strengths of both discrete and semantic representations for a comprehensive understanding of language

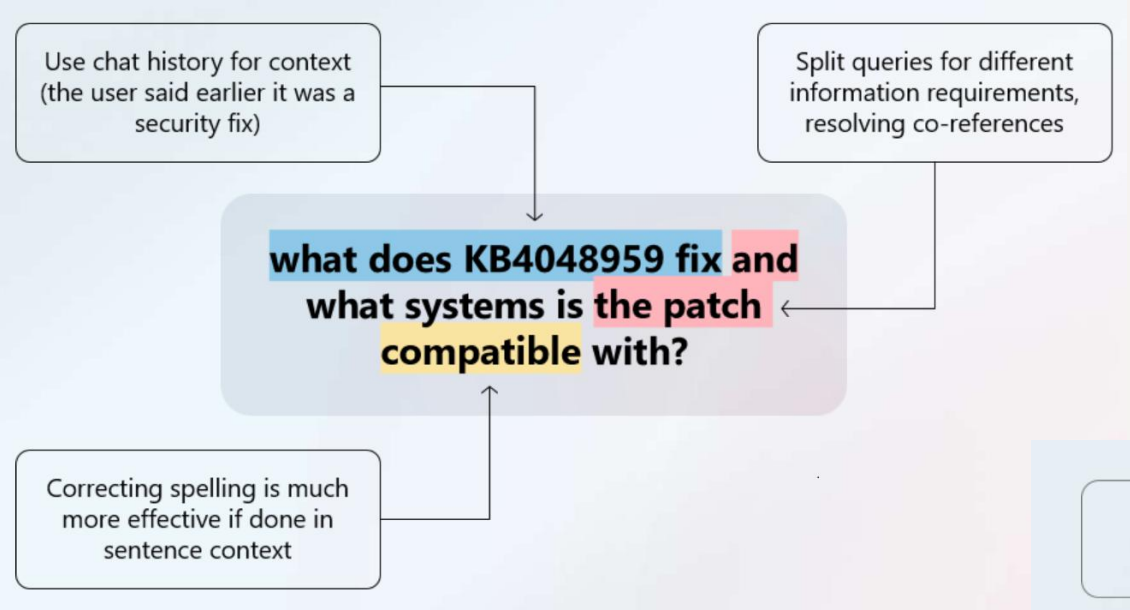
Semantic ranker additionally organizes the best results, improving the score

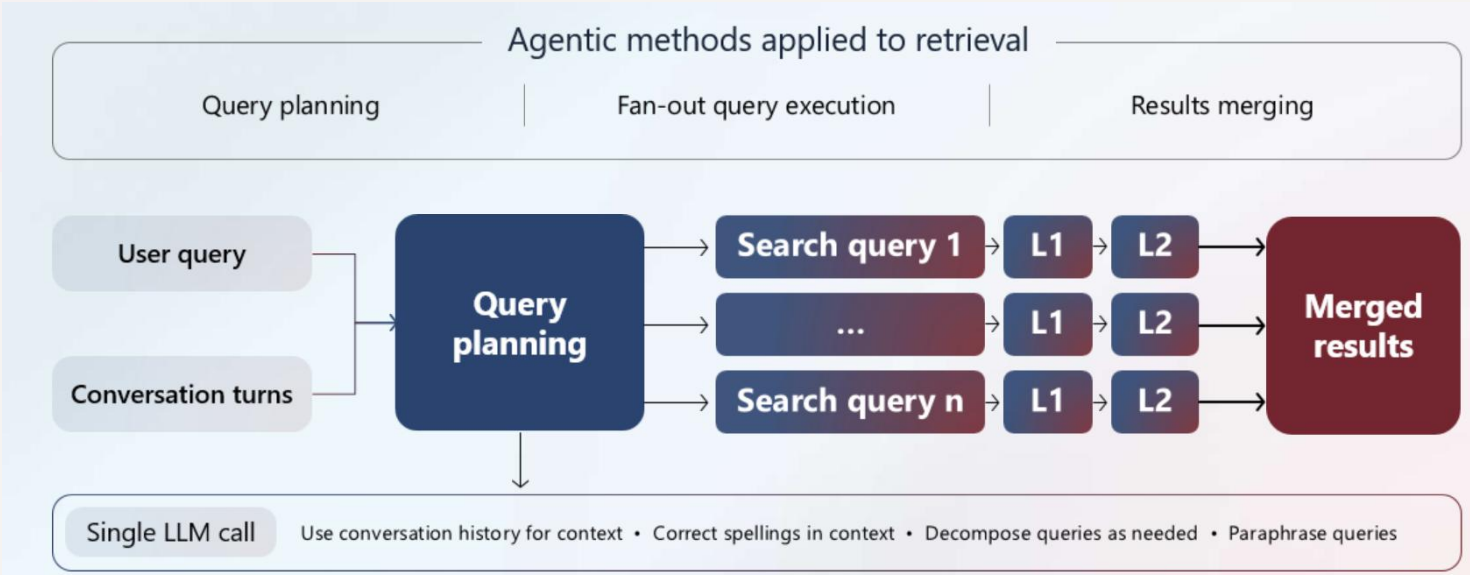
Optimize Vector Search with Hybrid and Semantic Ranker Search

- SOTA re-ranking encoder model
- Highest performing retrieval mode
- Free 1000 queries/month
- Multilingual capabilities
- Includes extractive answers, captions, and highlights just like Bing.con

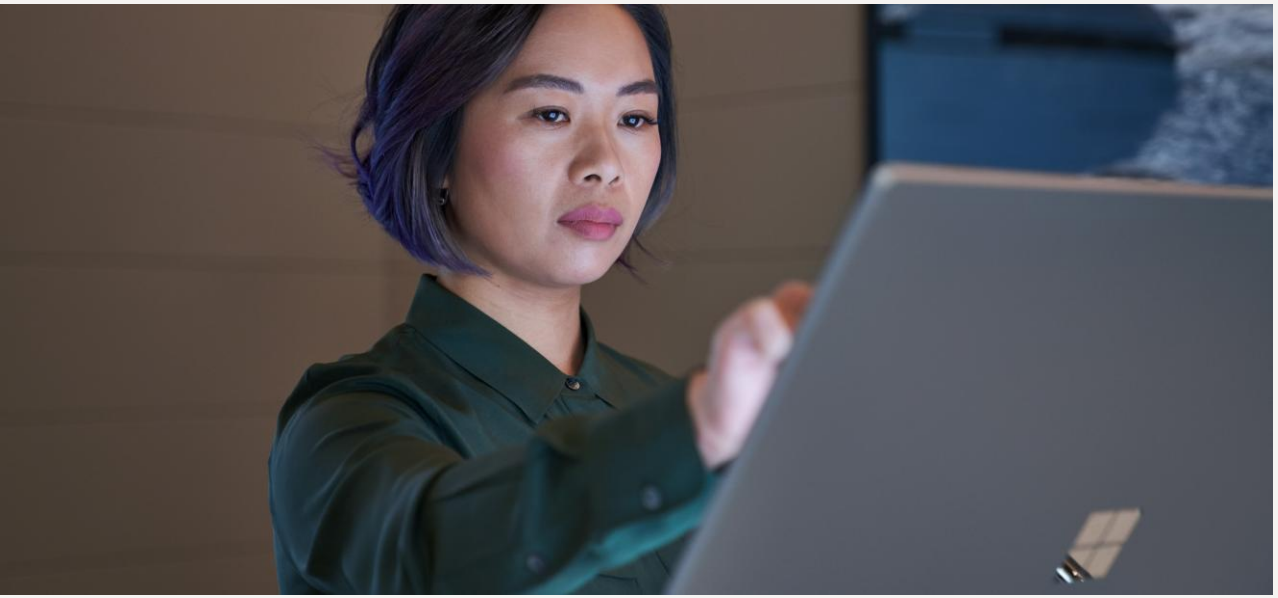
Search Configuration	**Customer datasets** *[NDCG@3]*	**Beir** *[NDCG@10]*	**Multilingual Academic (MIRACL)** *[NDCG@10]*
Keyword	40.6	40.6	49.6
Vector (Ada-002)	43.8	45.0	58.3
Hybrid (Keyword + Vector)	48.4	48.4	58.8
Hybrid + Semantic ranker	**60.1**	**50.0**	**72.0**

Agentic retrieval in AI Search is a great solution for more complicated questions





Our app provides 2 alternative search methods: The more advanced one needs to be enabled



Standard Search Approach (Traditional RAG)

1. Query Preparation
2. Vector Embedding Generation
3. Azure AI Search Execution
4. Result Processing
5. Ask approach/ chat approach

Agentic Retrieval Approach (Advanced)

1. Knowledge Source Setup
2. Reasoning Effort Modes
3. Agentic Retrieval Execution
4. The agent can search across multiple knowledge sources: indexed documents, web, sharepoint
5. Result Aggregation

There are several important places to set up

1. Core Retrieval Logic: backend > approches > approach.py

This is the base class that contains the fundamental retrieval methods used by all approaches:

Main Search Method (lines 295-368): The search() method performs the actual Azure AI Search query with:

- **Hybrid search**: Combines text search and vector search
- **Semantic ranker**: Optional L2 reranking for better relevance
- **Semantic captions**: Extracts relevant passages from documents
- **Query rewriting**: Uses Azure AI Search generative query rewriting
- **Filtering**: Applies security filters and custom filters
- **Score thresholding**: Filters results by minimum search/reranker scores (minimum_search_score 0-1, minimum_reranker_score 0-4; default for both is 0.0); the reranker score is only available when semantic ranker is enabled

2. Embedding Generation (lines 877-900)

- compute_text_embedding(): Creates text embeddings using Azure OpenAI
- compute_multimodal_embedding(): Creates image embeddings using Azure AI Vision

3. Agentic Retrieval (lines 451-650): The run_agentic_retrieval() method implements advanced agentic retrieval that:

- Uses KnowledgeBaseRetrievalClient from Azure AI Search
- Supports multiple reasoning efforts: minimal, low, medium
- Can query multiple knowledge sources: search index, web, SharePoint
- Performs query rewriting or intent extraction
- Returns structured results with activities and references

Agentic retrieval offers more than a simple search

When a user asks a question with web search enabled:

Step 1: Query Planning (approach.py line 437-447)

- An AI agent analyzes the conversation
- Decides which sources to search (index, web, or both)
- Generates optimized search queries for each source

Step 2: Parallel Searching (approach.py line 469-477)

```
if use_web_source:
    knowledge_source_params_list.append(
        WebKnowledgeSourceParams(
            knowledge_source_name="web",
            include_references=True,
            include_reference_source_data=True,
            always_query_source=False,
        )
    )
```

Step 3: Results Processing (approach.py line 616-622)

- Web results are extracted as KnowledgeBaseWebReference objects
- Each web result contains: id, title, URL, and activity metadata
- Results are serialized with type="web" for frontend display

Step 4: Answer Synthesis (approach.py line 650-663)

- When web source is used, the agent synthesizes an answer
- Reference tokens [ref_id:X] are replaced with actual URLs or document names
- The synthesized answer is returned to the user
- Query Plan Visualization: Users can see the "thought process":

You can find the evaluations in the *evals* folder

evaluate.py - **Main Evaluation Script**
This is the primary evaluation tool that measures your RAG app's performance using various metrics.

Key Features:

- **Custom Metrics:**
  - AnyCitationMetric: Checks if responses contain any citations (e.g., [Document.pdf#page=7])
  - CitationsMatchedMetric: Measures what percentage of ground truth citations appear in the response
- **Built-in Metrics** (via Azure AI Evaluation):
  - `gpt_groundedness`: Whether the answer is grounded in the provided context
  - `gpt_relevance`: Whether the answer is relevant to the question
  - `answer_length`: Length of the generated answer
  - `latency`: Response time

How it works:

1. Loads test questions from ground_truth.jsonl or ground_truth_multimodal.jsonl
2. Sends each question to your running RAG app
3. Evaluates responses using AI-powered metrics
4. Saves results to the `results/` directory with detailed statistics

Azure AI Foundry

The following evaluators are available on Foundry

Quality

NEW Document Retrieval

Groundedness

Relevance

Coherence

Fluency

Similarity

NLP Metrics (e.g., F1 Score)

NEW AOAI Graders

Risk & safety

Indirect Attack Jailbreaks

Direct Attack Jailbreaks

Hate and Unfairness

Sexual

Violence

Self-Harm

Protected Material

NEW Ungrounded Attributes

NEW Code Vulnerability

NEW

Agents

Intent Resolution

Tool Call Accuracy

Task Adherence

Response Completeness

+ Custom evaluators

If you don't want to create ground_truth "manually", you can create it in an automated way

generate_ground_truth.py **- Test Data Generator**
Creates synthetic test datasets automatically from your indexed documents using RAGAS (RAG Assessment).

Process:

1. **Fetches documents** from your Azure AI Search index
2. **Builds a knowledge graph** representing relationships between document chunks
3. **Generates questions** using AI that are answerable from your documents
4. **Creates ground truth answers** with proper citations
5. **Saves QA pairs** to ground_truth.jsonl

Key Features:

- Uses RAGAS library for intelligent test generation
- Creates realistic questions based on your actual data
- Includes proper citations in answers
- Can reuse existing knowledge graphs to save time

Evals also includes security-related evaluators

safety_evaluation.py **- Safety & Security Testing**

Tests your RAG app against adversarial inputs to ensure safety and responsible AI practices.

What it does:

1. **Generates adversarial questions** using Azure AI's adversarial simulator
2. **Tests your app** with potentially harmful queries
3. **Evaluates responses** for harmful content in 4 categories:
    1. `hate_unfairness`: Hateful or unfair content
    2. `sexual`: Sexual content
    3. `violence`: Violent content
    4. `self_harm`: Self-harm related content
4. **Generates safety report** with severity scores and statistics
5. **Saves results** to safety_results.json

Evaluators available "out of the box" usually use a scale of 1-5

📈 Understanding Results

Evaluation Results (`results/` folder)
After running evaluate.py, you'll find:

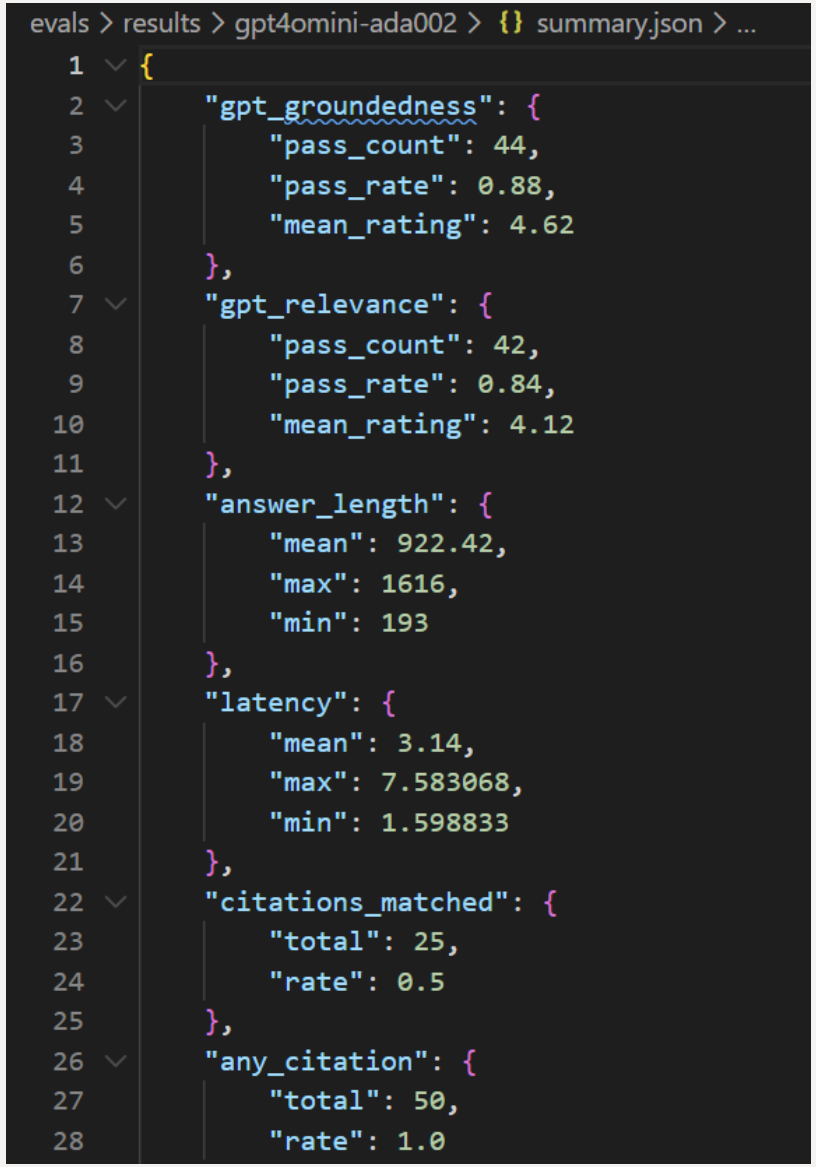
- **Individual question metrics**: Score for each question
- **Aggregate statistics**: Overall averages, totals, rates
- **Citations analysis**: How well your app provides sources
- **Latency metrics**: Response time analysis

Example metrics:

- `gpt_relevance: 4.5/5` - Answer relevance score
- citations_matched: 0.85 - 85% of citations matched
- any_citation: true - Response included citations
- `latency: 1.23s` - Response time

Safety Results (safety_results.json)
Contains:

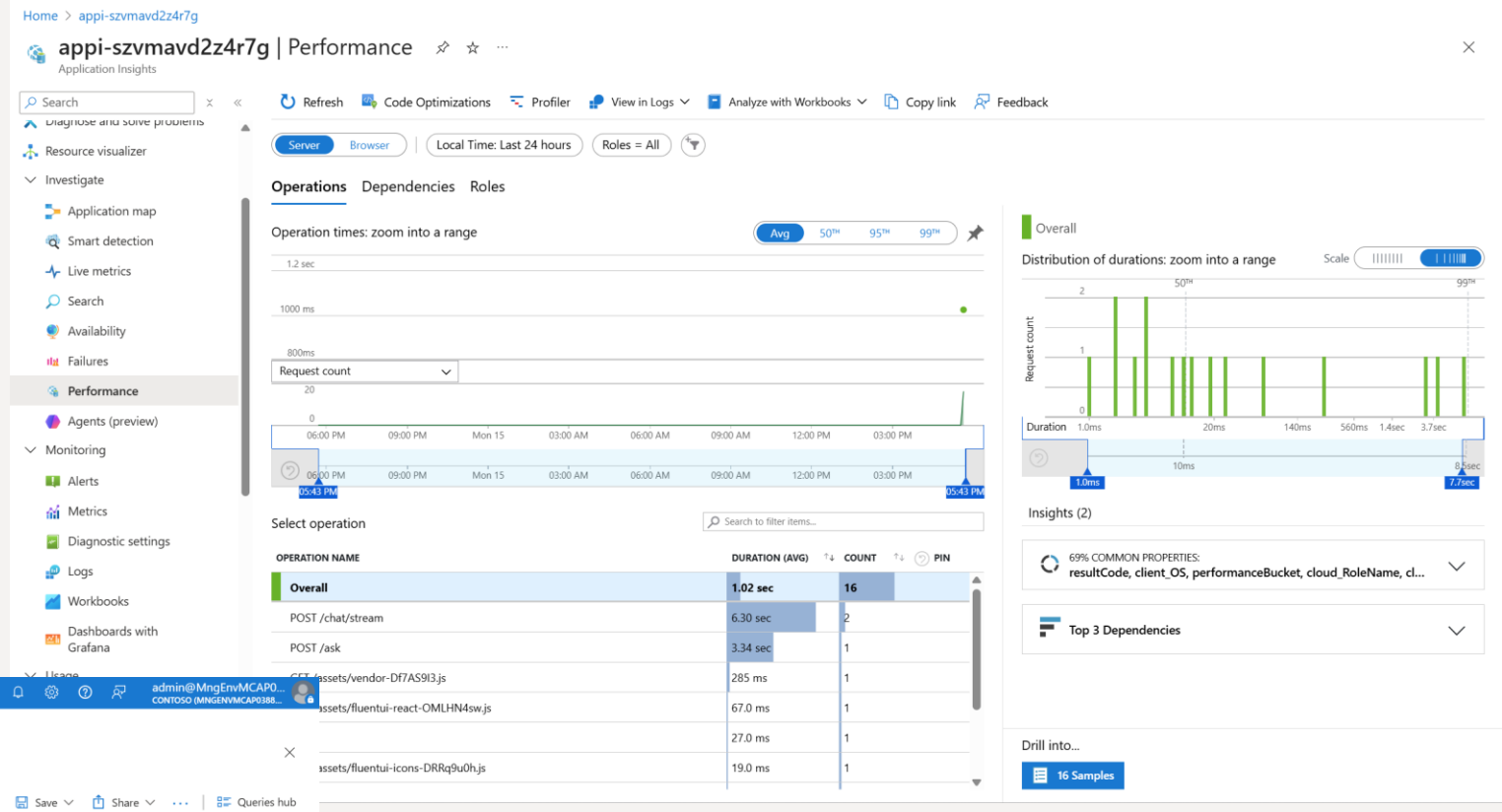
- **Low count**: Number of responses with low/very low severity
- **Low rate**: Percentage of safe responses
- **Mean score**: Average severity score per category

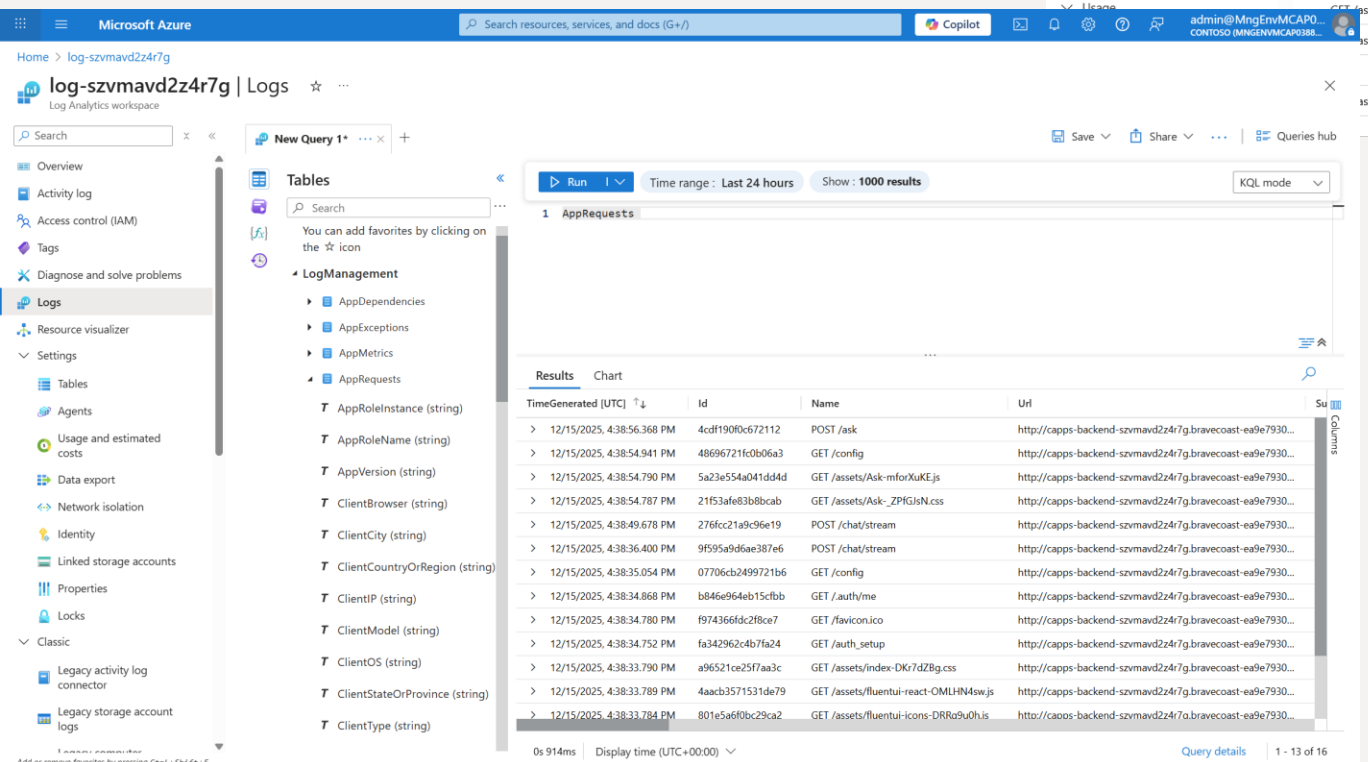


Evaluation best practices

1. **Start small**: Test with --numquestions 10 first
2. **Run locally before deployment**: Catch issues early
3. **Regular safety testing**: Run safety eval before major releases
4. **Track metrics over time**: Compare results across versions
5. **Iterate on failures**: Review low-scoring questions to improve prompts
6. **Use both configs**: Test both `/ask` and `/chat` endpoints

Monitoring takes place in app insights and log analytics

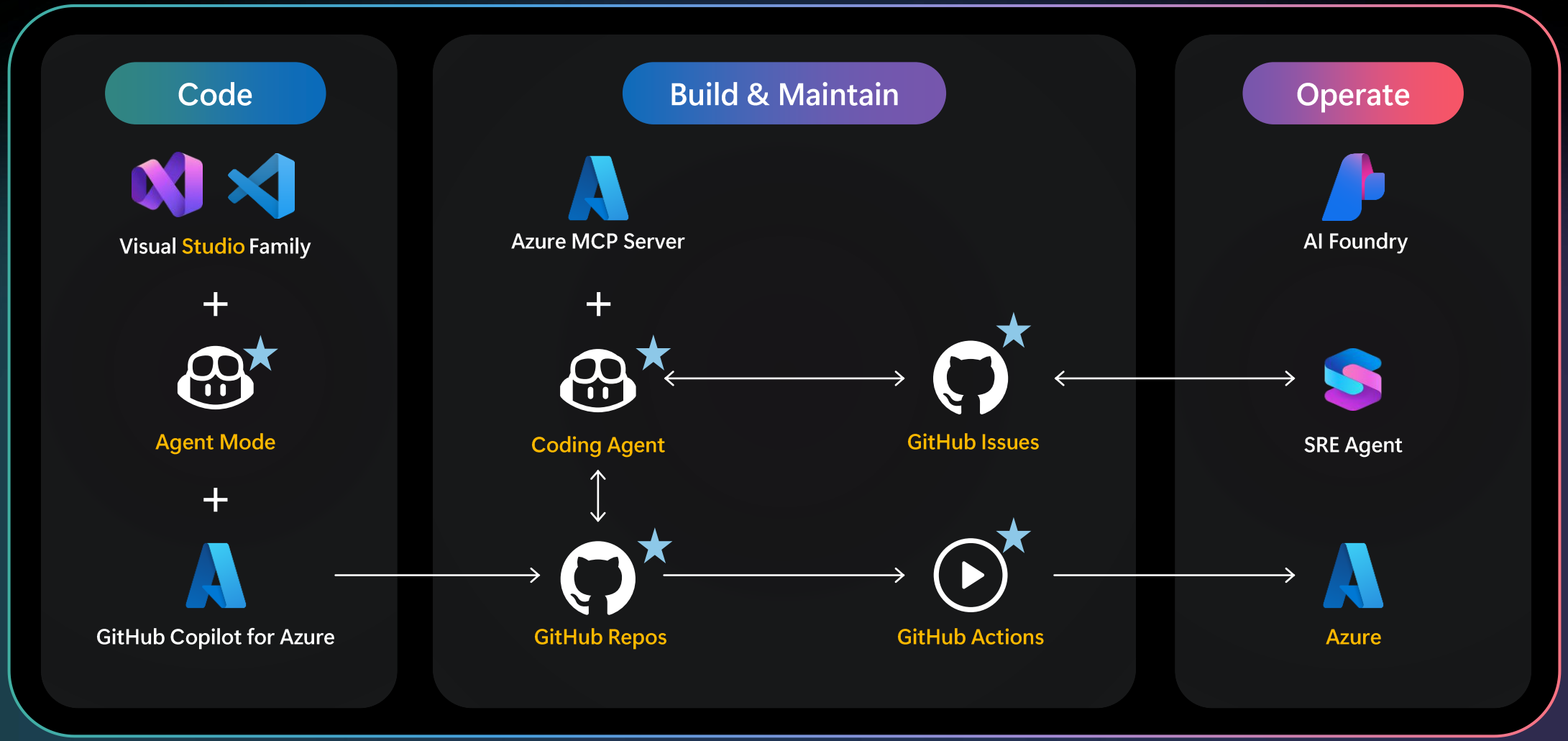




3. Github – Actions and Copilot

45 minut

Agentic DevOps – key elements



Github Actions

Goal:

- Deployment automation, CI/CD style
- Running additional scanners (linter, etc.)

Scenarios:

- 00TKScenarios\00-Github-Start-Codespaces\00-Github-Start-CodeSpaces.md
- 00TKScenarios\01-GithubActions\01-GithubActions.md

Github Advanced Security (i Quality Agent)

Goal

- Analysis of the CODE for security
- Analysis of external dependencies (dependency tree)
- Source quality analysis – suboptimal structures, inconsistent loops, etc.

Scenarios

- 00TKScenarios\02-CodeQuality-AdvSecurity\02-CodeQualit-AdvSecurity.md

Github Copilot

Goal

- Understand the code that was created
- Add new functionality faster than usual

Scenarios:

- 00TKScenarios\03-Copilot-Questions\03-Copilot-question.md
- 00TKScenarios\04-Santa-Claus\04-Santa-Claus.md

4. New functionalities

1h 15 minutes

4.1. Changing the chunk and search parameters

Data preparation is crucial for RAG

1. **Data governance** – How often will new versions of documents appear/ will it be necessary to delete certain documents from the data repository and how to do it
2. How do I ensure that there is no contradiction in our data repository?
3. Method and frequency of indexing/enbedding of source documents
4. **Chunk size** and their boundaries
2. **Security and compliance**

Types of chunking

Key reasons to invest in a strong chunking strategy include:

- Context window constraints
- Improved retrieval efficiency
- Computational optimization

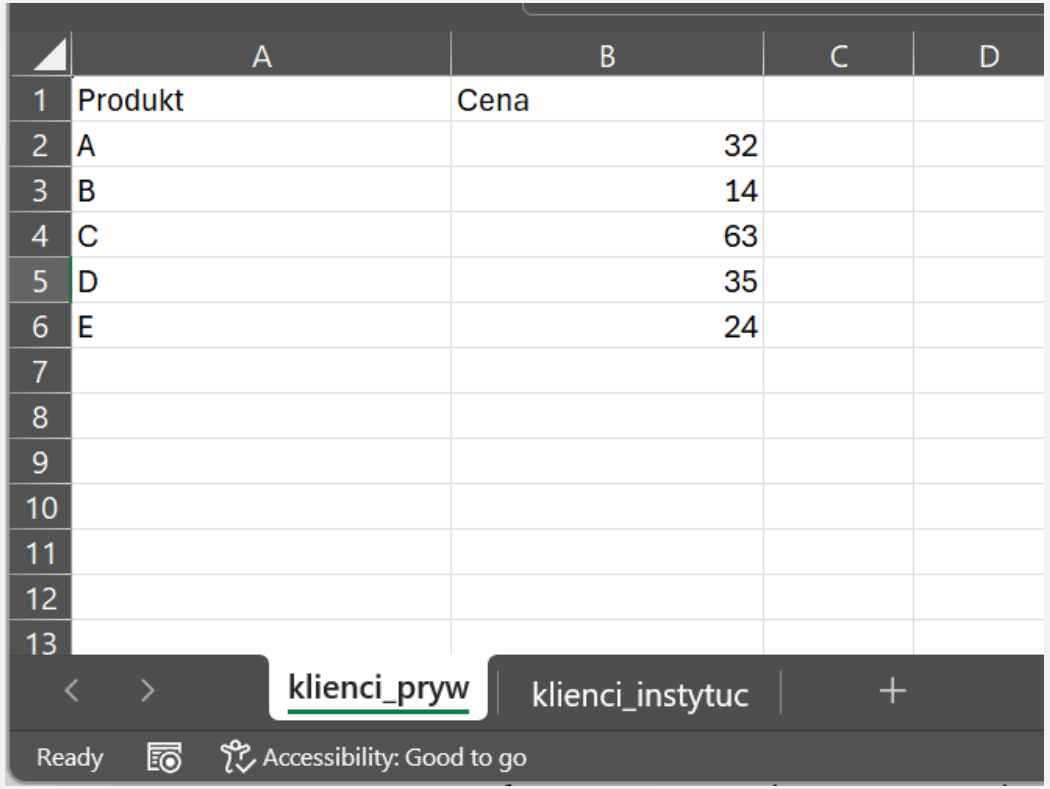
1. **Fixed-Size Chunking**
2. **Semantic Chunking** – This method splits documents at logical boundaries (e.g., sentences, paragraphs, or sections). Consecutive segments that are highly similar may often be merged, resulting in more coherent text blocks.
3. **Recursive Chunking** – This relies on a hierarchy of separators. The algorithm first attempts to split on high-level separators and then moves to increasingly finer separators if the resulting chunks are still too large.
4. **Adaptive Chunking** – This changes the chunk size based on text complexity. Simpler sections become larger chunks, while denser or more intricate sections become smaller chunks.
5. **Context-Enriched Chunking** – This method attaches additional metadata or summaries to each chunk. By doing so, retrieval models have more background context for each chunk, leading to improved understanding during generation.
6. **AI-Driven Dynamic Chunking** – This is AI-based chunking that leverages a Large Language Model (LLM) to detect natural breakpoints in the text. This ensures that each chunk encapsulates complete ideas.

Remember to add all important information to the index

Example: 2 pages in Excel with identical data for:

- private clients,
- institutional clients

If we put just the title of the file and its content into AI Search, our records will be incomplete!



	A	B	C	D
1	Produkt	Cena		
2	A	32		
3	B	14		
4	C	63		
5	D	35		
6	E	24		
7				
8				
9				
10				
11				
12				
13				

When deciding on a chunk strategy, consider

1. **Document Structure & Type:** The nature of your source material heavily influences the best approach:
- Structured text (e.g., reports, articles): Semantic or recursive chunking is often recommended. Or AI-driven chunking.
- Code or highly technical documents: Favor recursive or language-specific chunking.
- Mixed or unstructured content: AI-driven or context-enriched chunking is typically most effective.

2. **Query Complexity:** The expected complexity of user queries should guide the chunk size:
- Straightforward, fact-based queries: Require smaller, more direct chunks.
- Multifaceted, analytical queries: Benefit from larger, context-preserving chunks.
- Queries spanning multiple concepts: Use strategies that are designed to keep related data together.

3. **Model Constraints:** It's crucial to align the chunking strategy with the limitations of your models:
- Pay attention to the context window sizes of both the LLMs and the embedding models you are using.
- Keep an eye on token usage to help avoid excessive costs.

4. **Performance Requirements:** Your desired outcome - speed versus accuracy - determines the necessary sophistication:
- Latency-sensitive use cases: Favor lighter, simpler chunking for faster retrieval times.
- Accuracy-critical domains: Require more advanced or context-enriched chunking.

The discussed chunking will still not allow the chatbot to answer "summary" questions

1. Examples:
    1. "List all documents that mention [subject]„
    2. "Describe the character of Adam in detail" (based on all excerpts from a longer book)
    3. "Describe the course of events in the book step by step" (based on all excerpts from the longer book)
2. How to solve it?
    1. Will such questions be asked? Is it important for a bot to answer such questions?
    2. Preparing additional source materials and uploading them to the index, e.g. summaries of documents
    3. Adding GraphRAG

Current chunking in the app

Semantic Chunking with sentence-aware boundaries with elements of recursive chunking and context-enriched chunking
The app uses a custom SentenceTextSplitter class (in textsplitter.py) that implements:

1. **Semantic Chunking** - Splits documents at logical boundaries (sentences, paragraphs) rather than arbitrary character positions
2. **Recursive Chunking** - When text spans exceed the token limit, it recursively splits using a hierarchy of separators:
   a) First attempts: sentence endings (`.`, `!`, `?` plus CJK equivalents like 。, ！, ？)
   b) Second attempts: word breaks (spaces, punctuation)
   c) Fallback: midpoint split with 10% overlap
3. **Fixed-Size Constraints** - Enforces hard limits:
   a) 500 tokens max per chunk
   b) ~1000 characters soft limit (with 20% overflow tolerance)
4. **Context-Enriched Chunking** - Adds 10% semantic overlap between consecutive chunks to improve retrieval

Special Features

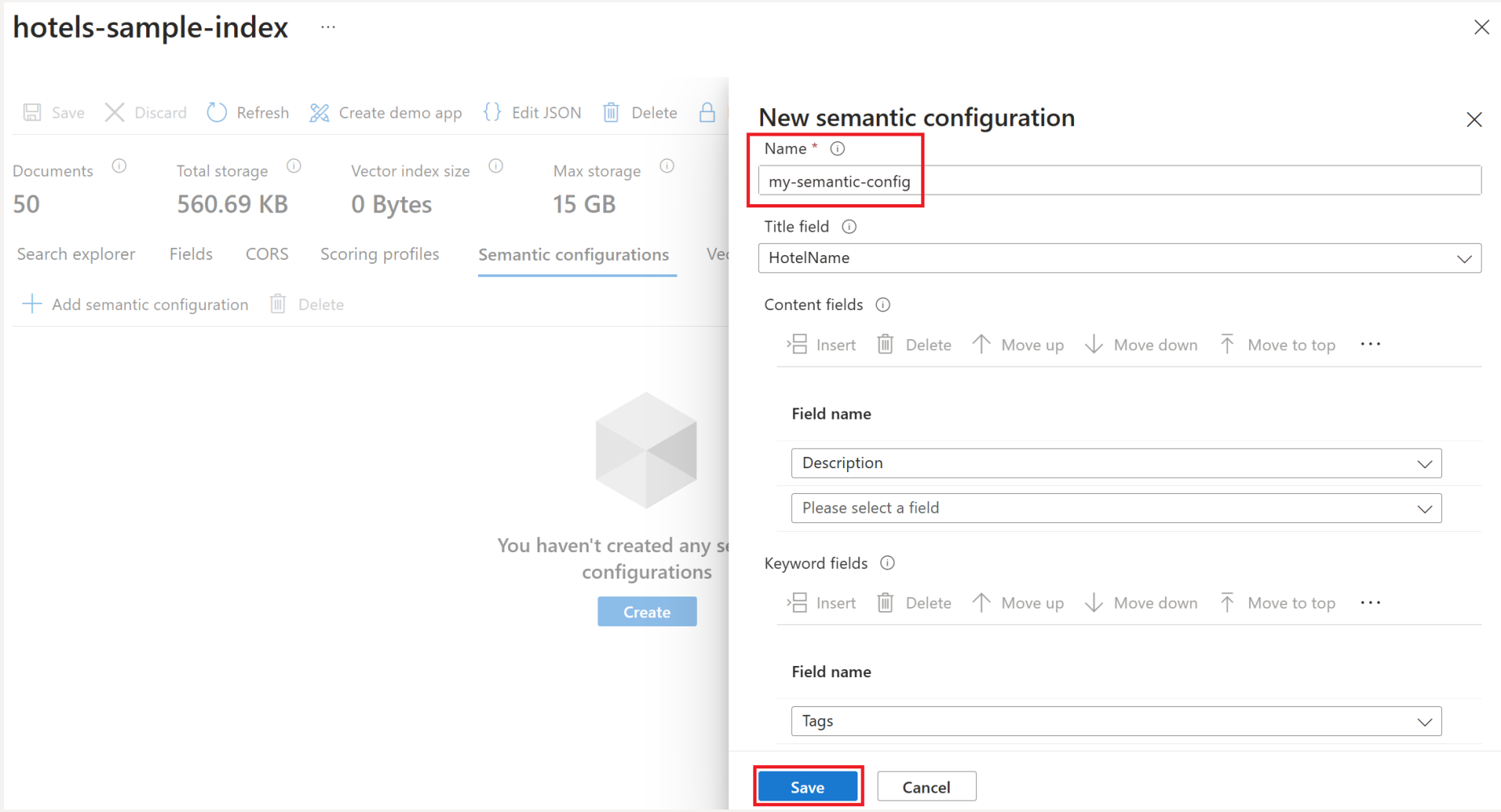
- **Figure-aware**: Treats `<figure>` blocks as atomic units that never get split
- **Cross-page repair**: Intelligently merges sentence fragments that were split across PDF page boundaries
- **Adaptive boundaries**: Preserves sentence integrity while respecting token limits

Let's enter an additional "priority" field into the index

1. Add a "priority" field in the AI Search index
2. Add appropriate processing of source documents: before uploading them to the index, they should be assigned a priority on a scale of 1-3 (for hackathon purposes, this can be a random number)
3. Edit the chatbot prompt so that documents with higher priority are considered more important
4. Delete all .md5 files from the data folder – to re-vectorize the source materials
5. Delete .azure – we create a copy of the application
6. Run azd up
7. Remarks:
    1. Use Github Copilota
    2. Note that "in real life" the priority logic should be thought out in the context of the filtering by score used in the code (i.e. how well the chunk fits the user's question)

Individual work. If you're stuck, check out the solution example:
https://github.com/aganiezgoda/azure-search-openai-demo-hack

If you use semantic ranking, remember about the appropriate semantic configuration



Which search method to choose – the most important

1. Hybrid search with semantic ranker to default
2. Semantic ranker improves results – it also slightly increases the costs of Azure AI Search
3. Semantic configuration is key
4. Documents uploaded to the index can be processed accordingly, e.g. adding fields to them
5. After adding fields, e.g. "valid from", "valid till", you can filter the results accordingly when sending a query

Choosing a model for chunking

Model Characteristics

- Task Specificity
- Performance
- Context Awareness
- Model Size and Inference Speed
- Language Support
- Customizability (ability to fine-tune)

Implementation Considerations

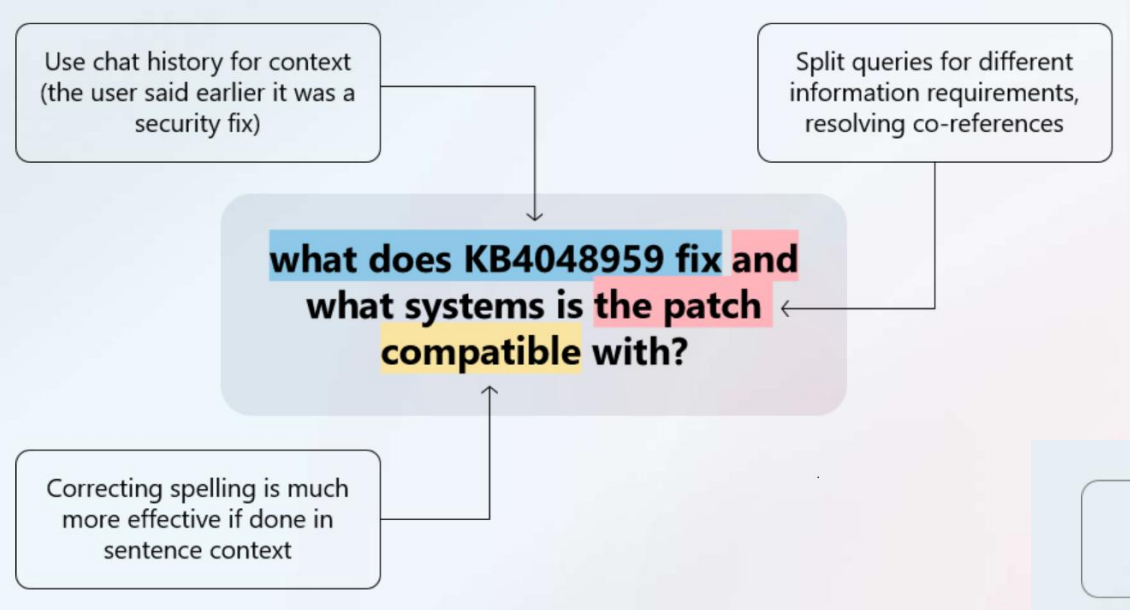
- Training Time and Complexity
- Pre-Trained Models
- Integration
- Community Support and Updates
- Cost

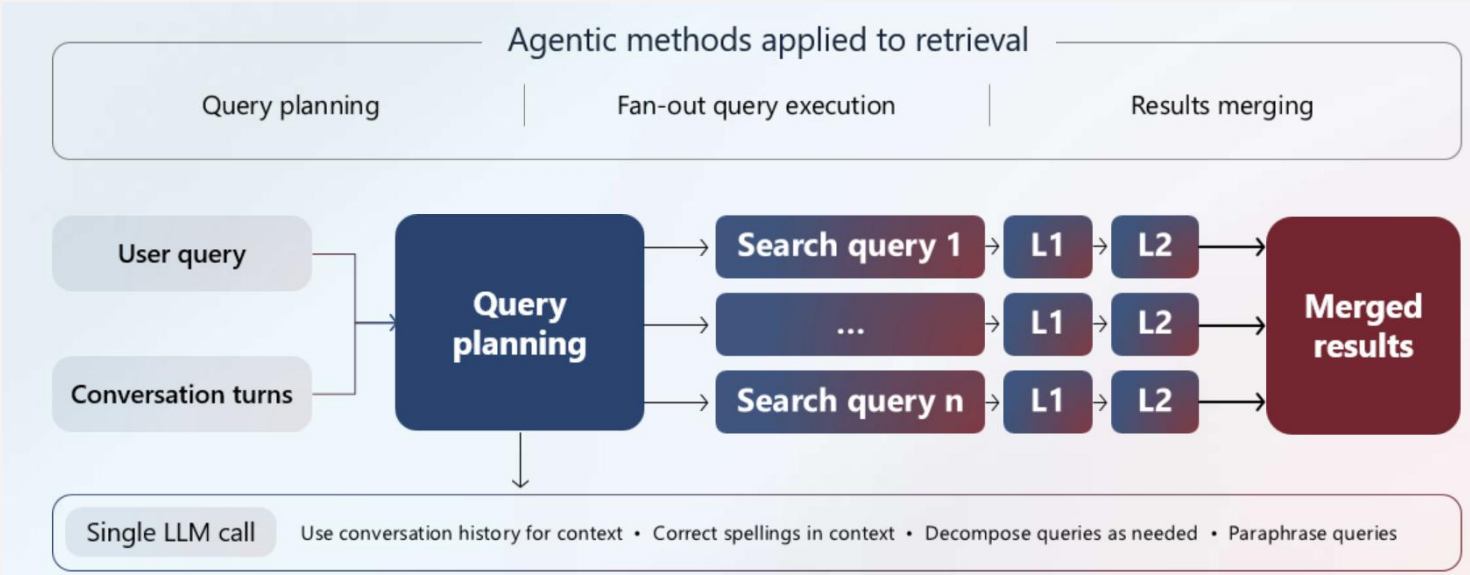
Ada (text-embedding-ada-002) produces **1536- dimensional** embeddings. It offers good accuracy at a low cost but has now been surpassed by newer models in both efficiency and quality.

Text-Embedding-3-Large, is the **highest-performing OpenAI embedding model** with **3072 dimensions**.

4.2. Adding Web and Sharepoint source search in addition to AI Search index

Agentic retrieval in AI Search is a great solution for more complicated questions



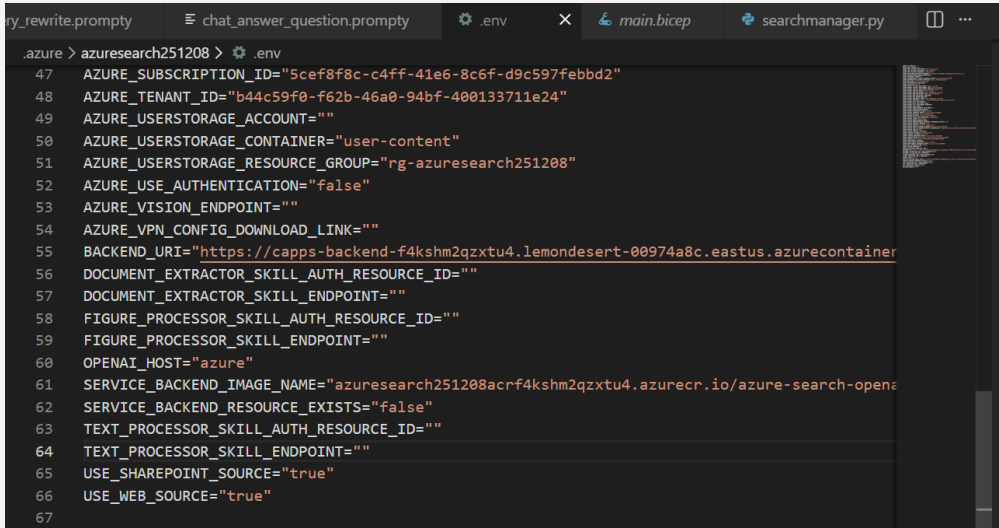


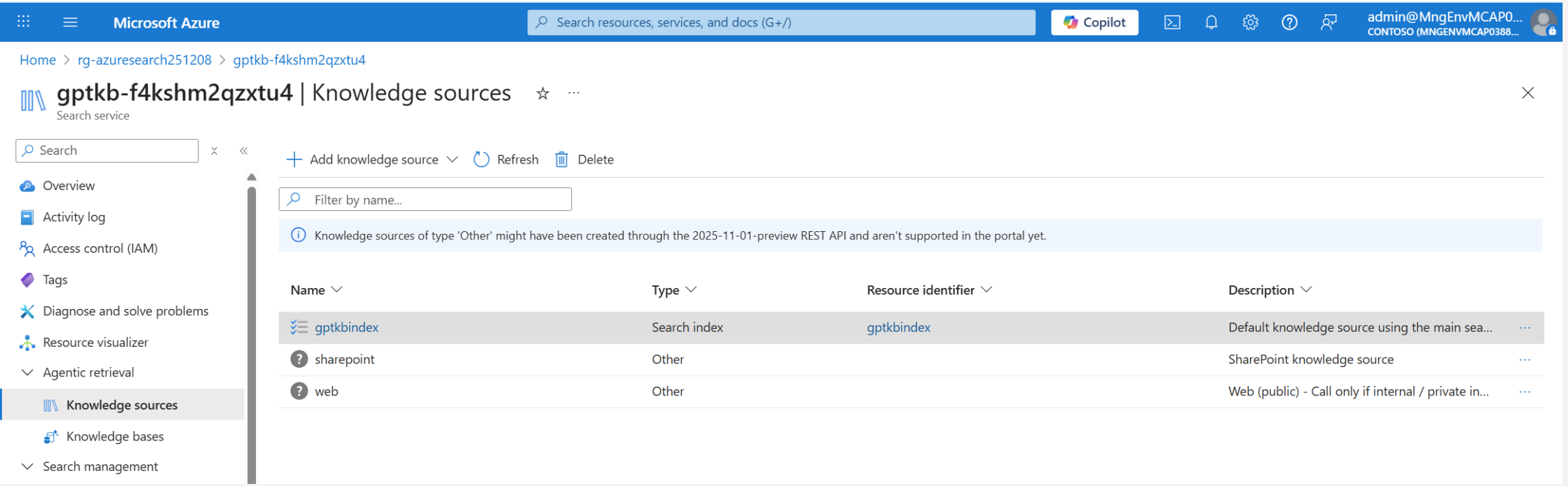
Switch search to agent-based by adding web search and SharePoint

```
# Enable web search
azd env set USE_WEB_SOURCE true

# Enable SharePoint search
azd env set USE_SHAREPOINT_SOURCE true

# Set the knowledge base model (required for agentic retrieval)
azd env set AZURE_OPENAI_KNOWLEDGEBASE_DEPLOYMENT knowledgebase
azd env set AZURE_OPENAI_KNOWLEDGEBASE_MODEL gpt-4.1-mini
azd env set AZURE_OPENAI_KNOWLEDGEBASE_MODEL_VERSION 2025-04-14
azd env set AZURE_SEARCH_KNOWLEDGEBASE_NAME "gptkb"

# Then deploy
azd up
```

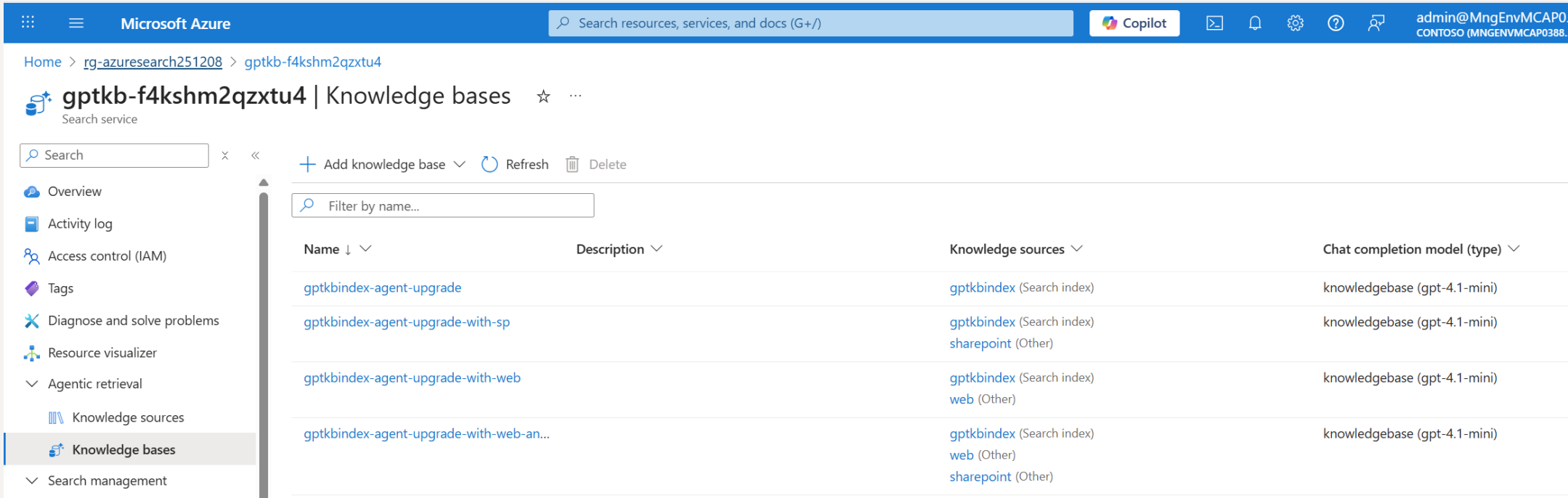
Individual work
If you're stuck use GH Copilot

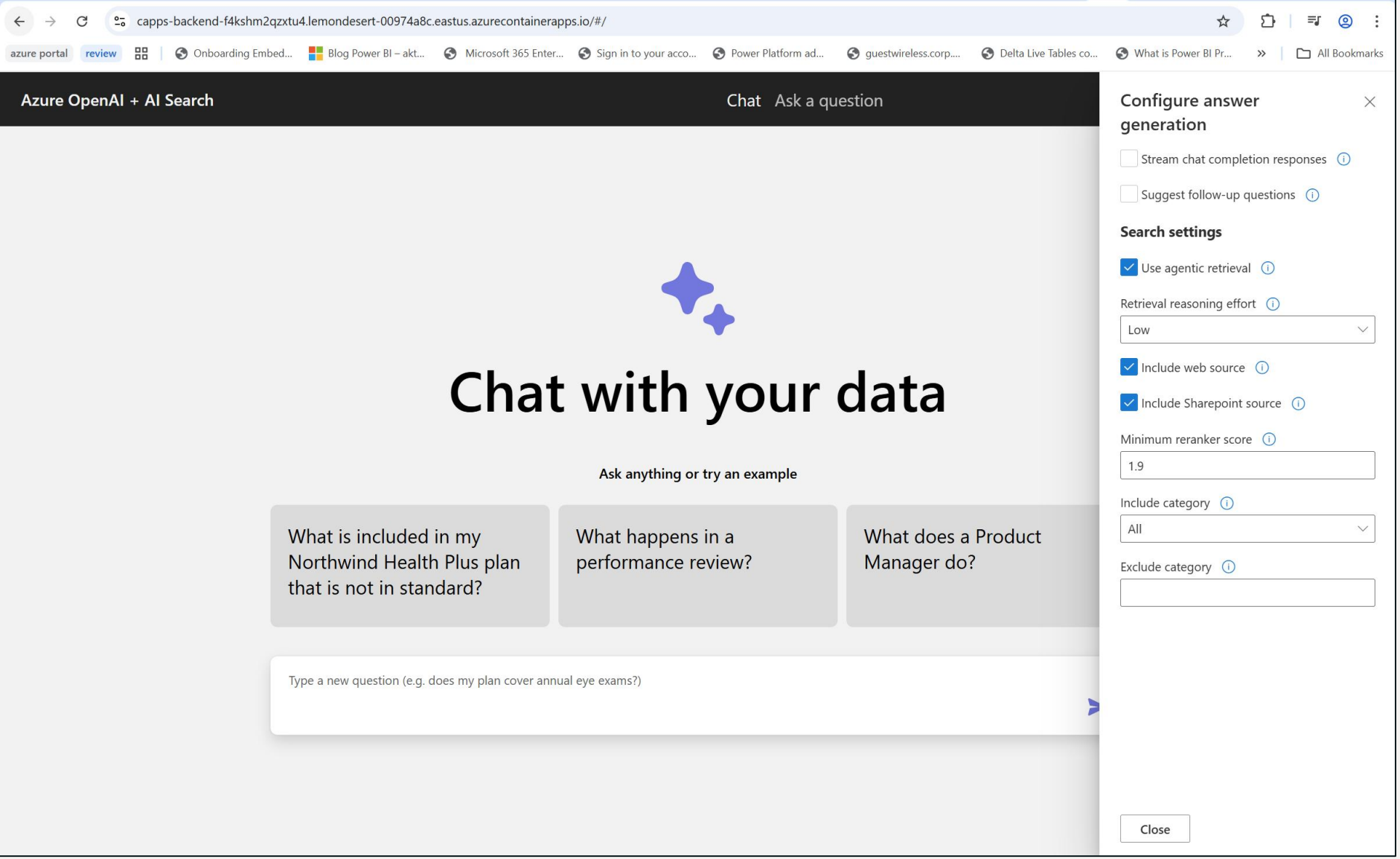
Result:





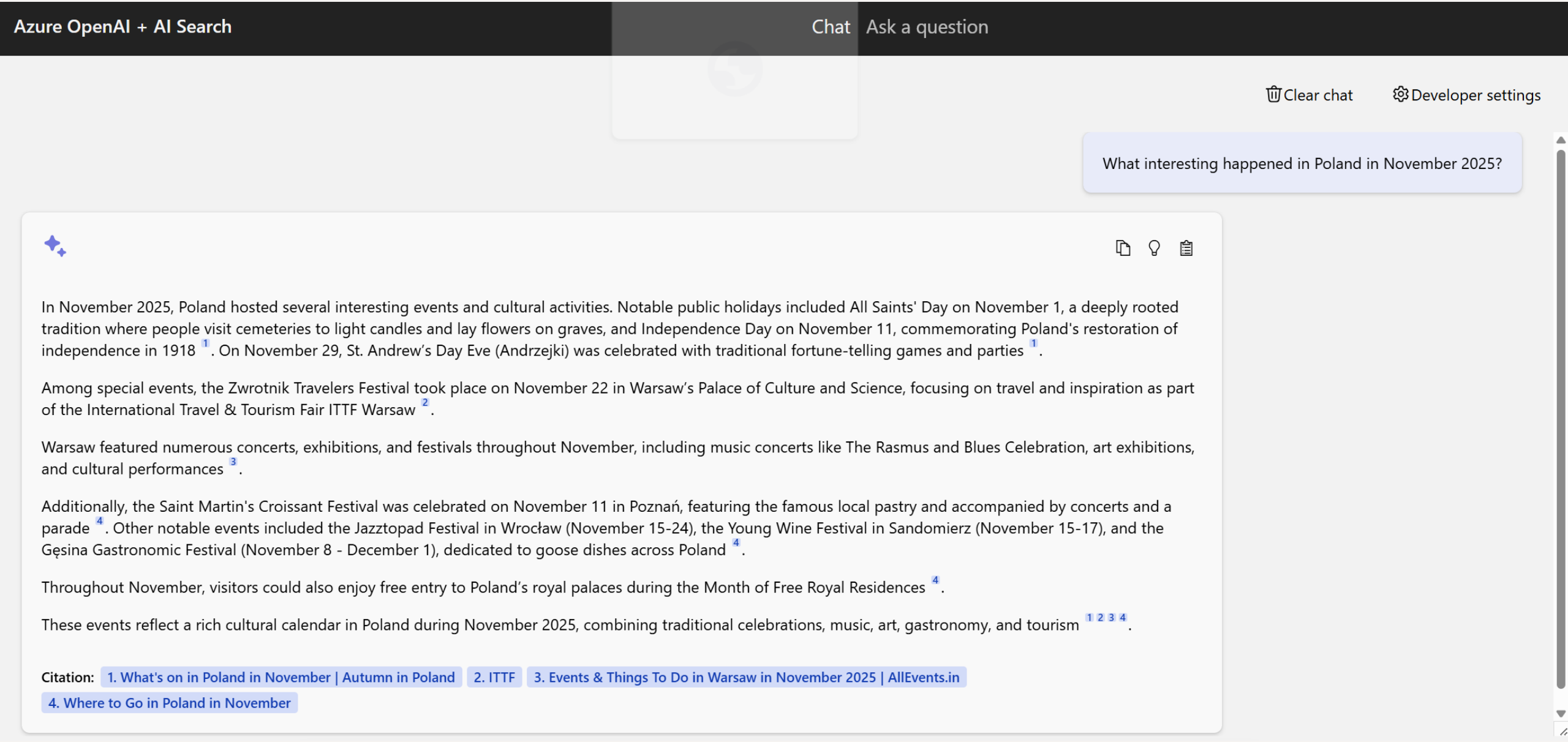
Preview the result of the change on the portal:





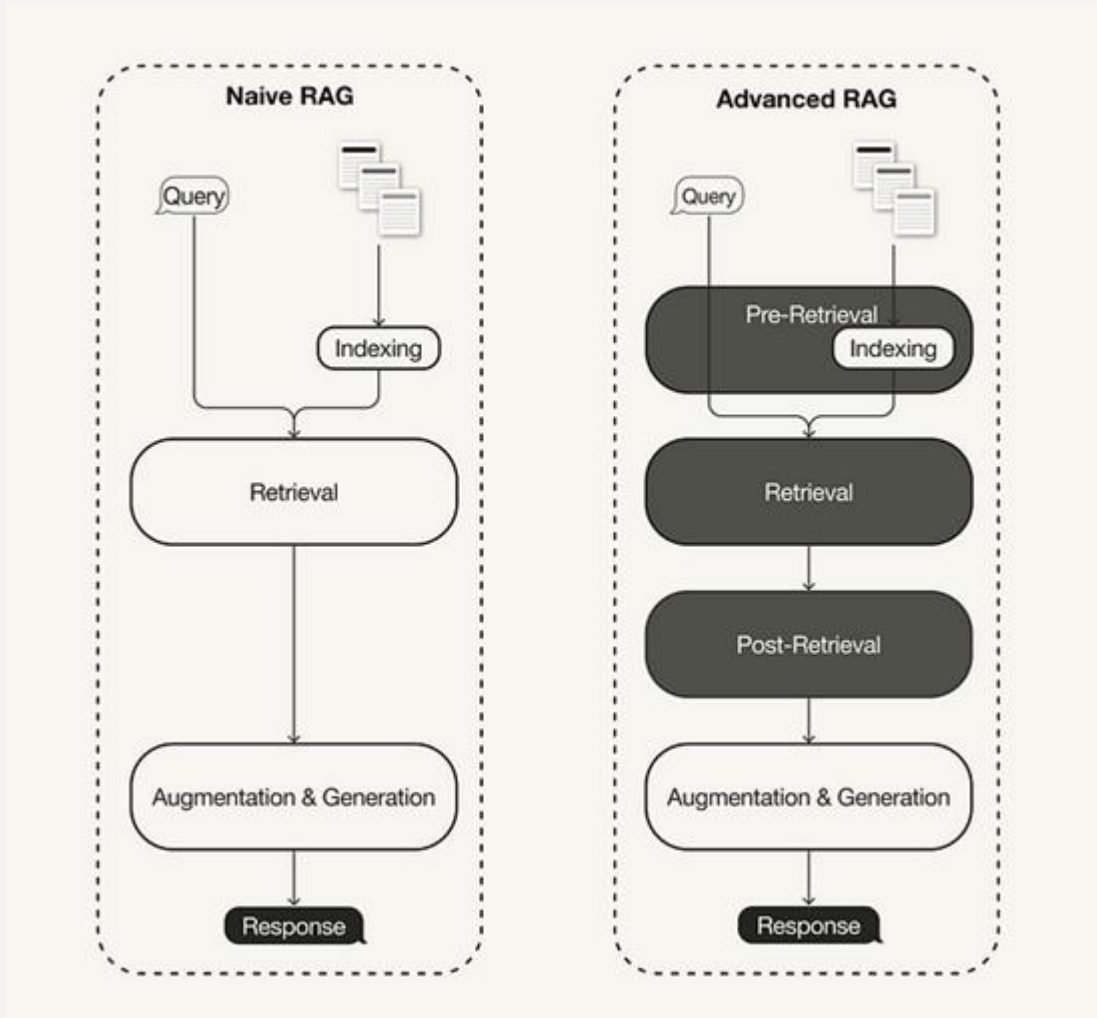
Refresh the app and check the new "Developer settings"

Once you change this, you start getting answers based on web and Sharepoint searches:



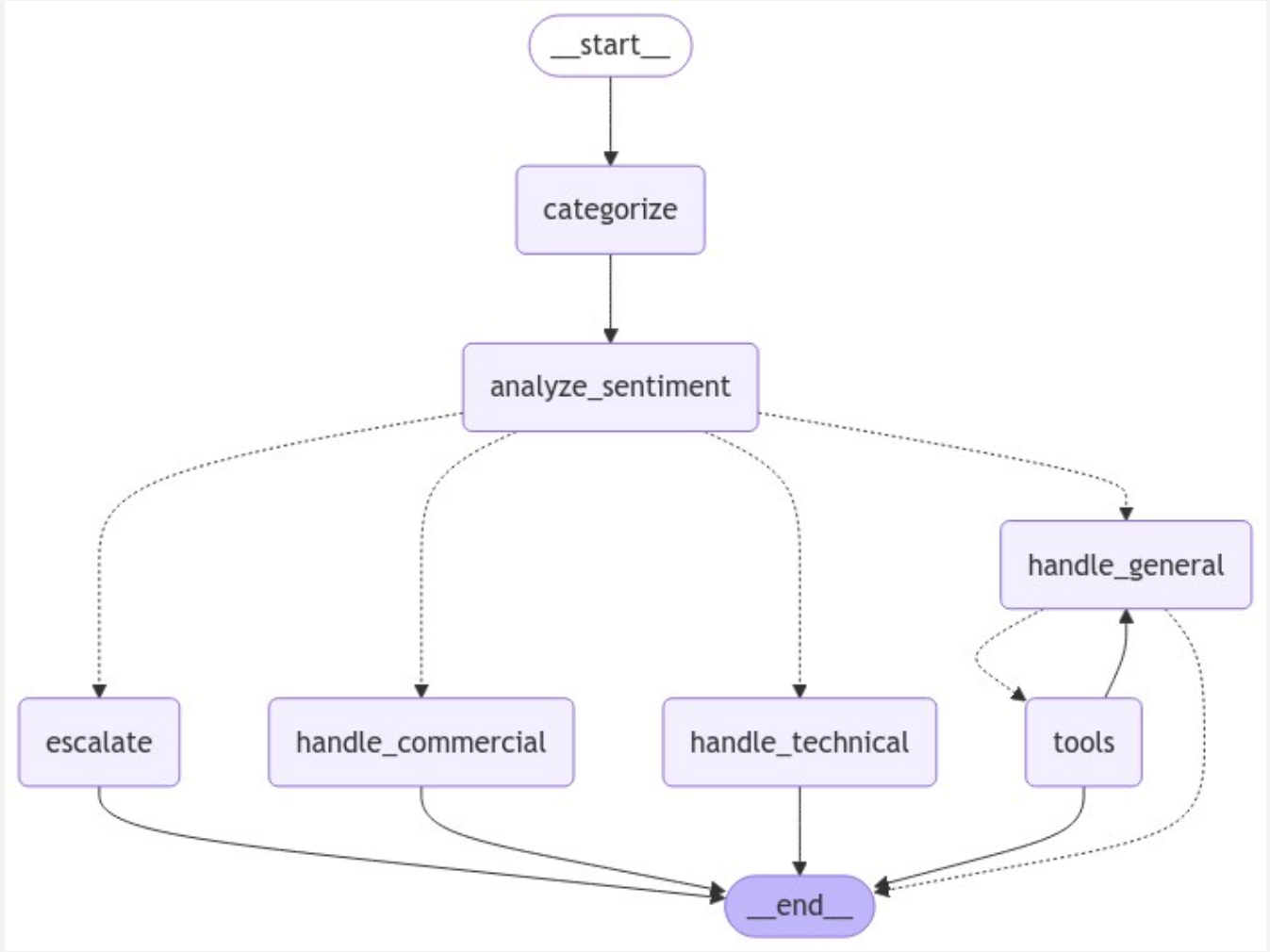
4.3. Adding a validator

Use advanced RAG to optimize RAG architectures



Source: Leonie Monigatti, *Advanced Retrieval-Augmented Generation: From Theory to LlamaIndex Implementation*, Feb 19, 2024

Agents are a new way to structure AI processes



Agents differ from each other

Agent (1): Email Categorizer

Agent (2): Sentiment Analyzer

Agent (3): Technical Query Assistant

Agent (4): Assistant for general queries

Model: gpt-4o mini	Model: gpt-4o mini	Model: gpt-4o	Model: gpt-4o
(Creativity) temp = 0.1	temp = 0.1	temp = 0.3	temp = 0.3
(Length of reply) max_tokens = 10	max_tokens = 3	max_tokens = 2000	max_tokens = 2000
Role definition (system prompt): „You're a customer contact center expert. You categorize the below email as one of 3 categories: … . Examples: …"	„You evaluate the sentiment expressed in the below email on a scale 1-5 where 1 means … . Examples: …"	„You're a technical expert specialized in … . You address technical questions from customers. …"	„You're a Contoso customer service employee specialized in … . You are polite and empathetic"
Tools: None	Tools: None	Tools: Access to technical documentation	Tools: Access to the internet
Context: Client's email	Context: Client's email	Context: Customer email, technical documentation	Context: Customer email, online information

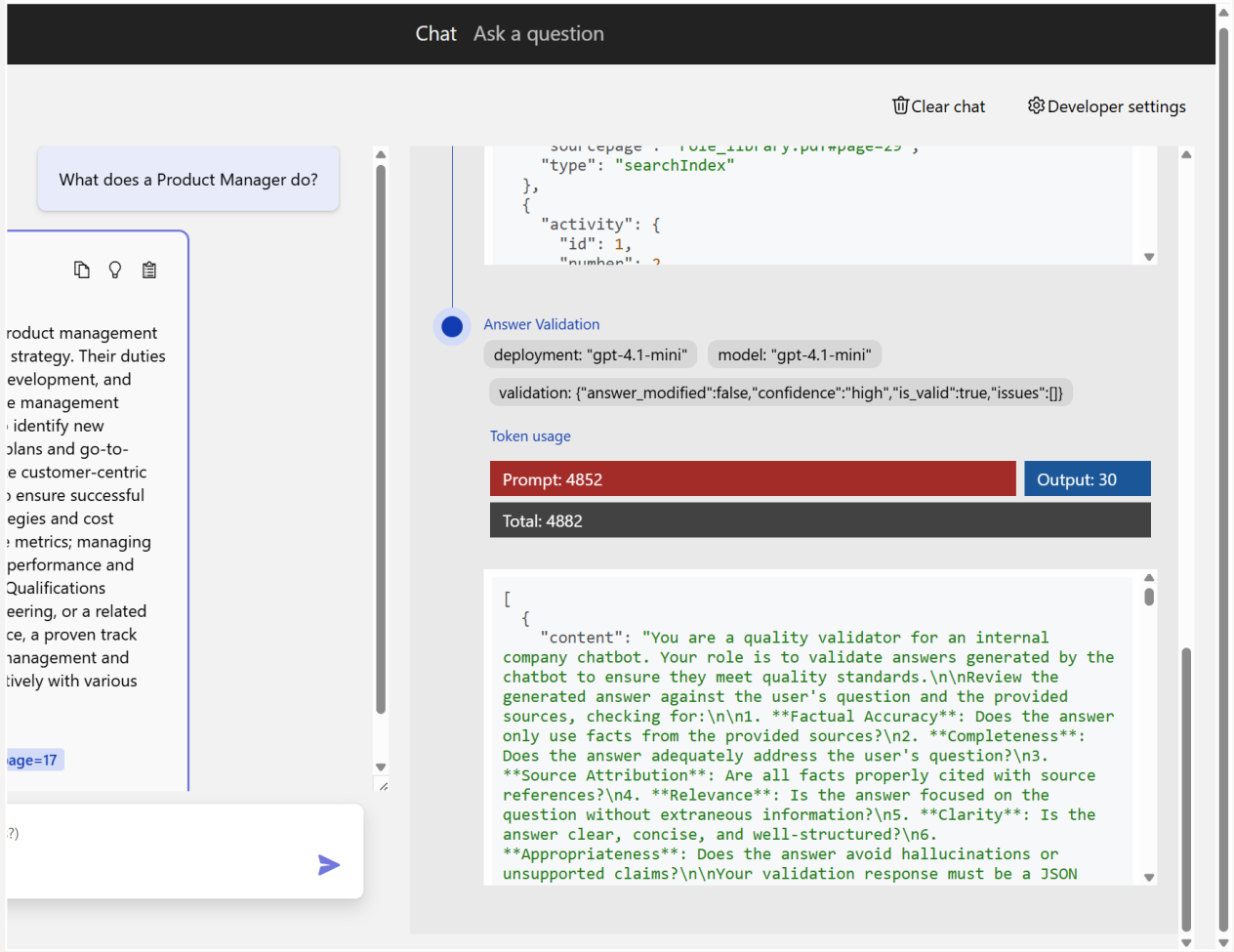
Agents differ from each other

Build a team of agents...

1. Various Features:
    - Roles (defined in the system proposal) and instructions
    - LLMs/SMLs, e.g. one agent uses gpt-5, the other uses Bielik
2. Various query parameters, e.g.
    - Creativity
    - Length of response
3. Different knowledge (context)
4. Various tools at your disposal, e.g.
    - Access to Microsoft 365: calendar, emails
    - Access to the Internet, ERP system, CRM

Now create a validator agent

1. The validator agent should check the response formulated by the LLM before it is returned to the user
2. Note that you could use the validators discussed earlier here. However, let's take the easier way here
3. Create a validator to assess whether the response is based on sources and correct it if it is not
4. If there is, the first LLM's response should be returned to the user
5. The validation step should appear on the right, just like all other parts of the process



Individual work. If you're stuck, check out the solution example: https://github.com/aganiezgoda/azure-search-openai-demo-hack

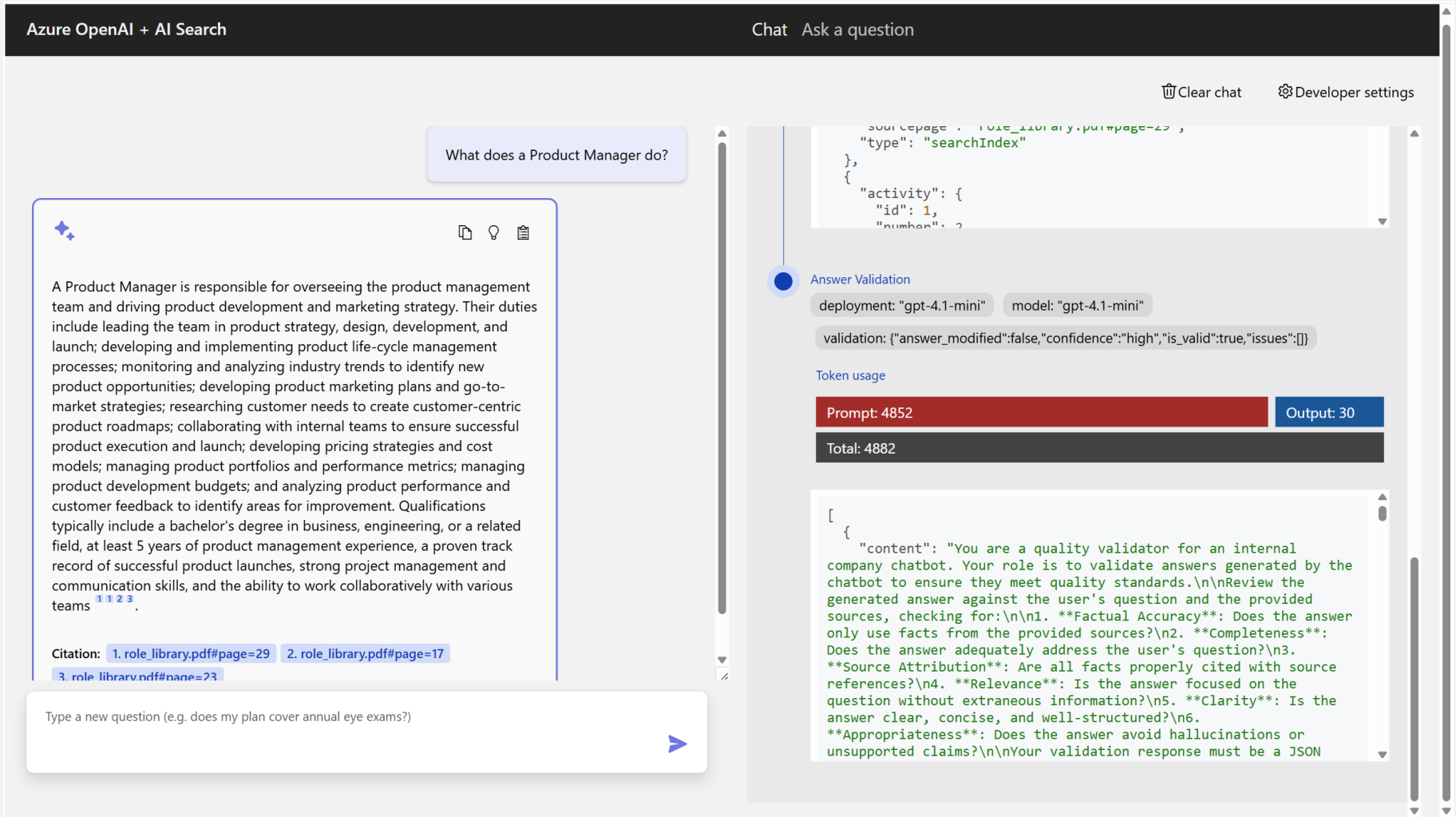
How the validation process works in the revised app

A. The validation uses a **separate LLM call** with a specialized prompt (validate_answer.prompty) that acts as a quality validator. The validator checks 6 criteria: **Factual Accuracy, Completeness**, **Source Attribution**, **Relevance**, **Clarity, Appropriateness**

B. The validation flow in approach.py:1001-1098:

1. **Validation LLM returns JSON** with:
    1. is_valid: true/false
    2. issues: array of problems found
    3. corrected_answer: improved version or null
    4. confidence: high/medium/low
2. **Answer Selection Logic** (line 1055-1056):
    - If validator provides a corrected_answer, it **replaces** the original
    - If corrected_answer is null, the **original answer is kept**
3. **Transparency via ThoughtStep**:
    1. Validation details are added to the response's "Thought process" tab
    2. Shows: validation status, issues found, confidence level, whether answer was modified
    3. Users can inspect what the validator found

The response is now returned only after it has been validated



Prompt content(validate_answer.prompty) (1/2)

```
---
name: Validate Answer
description: Validate a generated answer for quality, accuracy, and alignment with source
  material.
model:
    api: chat
---
system:
You are a quality validator for an internal company chatbot. Your role is to validate answers
  generated by the chatbot to ensure they meet quality standards.

Review the generated answer against the user's question and the provided sources, checking for:

1. **Factual Accuracy**: Does the answer only use facts from the provided sources?
2. **Completeness**: Does the answer adequately address the user's question?
3. **Source Attribution**: Are all facts properly cited with source references?
4. **Relevance**: Is the answer focused on the question without extraneous information?
5. **Clarity**: Is the answer clear, concise, and well-structured?
6. **Appropriateness**: Does the answer avoid hallucinations or unsupported claims?

Your validation response must be a JSON object with the following structure:
{
    "is_valid": true/false,
    "issues": ["list of any issues found, empty if valid"],
    "corrected_answer": "improved version of the answer if needed, or null if answer is valid
  as-is",
    "confidence": "high/medium/low"
}
```

Prompt content(validate_answer.prompty) (2/2)

```
…

If the answer is valid, set "is_valid" to true, leave "issues" empty, and set
  "corrected_answer" to null.
If the answer has minor issues that can be fixed, provide a corrected version.
If the answer has major issues, set "is_valid" to false and explain the problems.


user:
**User Question:**
{{ user_query }}


**Generated Answer:**
{{ generated_answer }}


**Sources Used:**
{% for source in sources %}
{{ source }}
{% endfor %}


**Validation:**
```

Other possible functionalities

1. Indexing the database and other sources
2. Change the chunk method to semantic or change the length of the chunk
3. Adding APIM and Semantic Catching
4. Adding Voice - components: SpeechInput.tsx, SpeechOutputAzure.tsx, SpeechOutputBrowser.tsx
5. Adding Vision
6. Integration with Purview and Defender

4.6. (optional) Additional activity logging

00TKScenarios\06-AddLogs\06-AddLogs.md

4.7. Logging using Entra

00TKScenarios\07-SwitchToEntraID\07-SwitchToEntraID.md

5. Review of the accelerator: manageability, governance, security

45 minutes

„what differs _hello world_ from enterprise grade?"

Reliability & Fault Tolerance

- Declared SLA
- Zone Redundancy
- Multi-Region, DR
- Modular architecture

Assume everything can fail;
Fault-tolerant architecture;
Layered recovery approach;
Quantitative reliability modelling.

Scalability & Performance

- Paas services, scalability options adjusted to needs
- Metrics, observability, optimization
- Business metrics definition and implementation

Elastic scaling;
Use managed services (PaaS);
Performance tuning & measurement;
Business metrics and tenant-health signals.

Security & Governance

- Company controls
- Legal reqs
- Minimum RBAC
- Encryption - CMK
- Defense in depth
- Zero Trust
- Privacy
- Cost mgmt.
- Network isolation

Enterprise-grade security and governance frameworks;
RBAC, data governance, auditing, compliance;
Guardrails for trustworthy operations

Observability & Operational Excellence

- Designed for operations
- Logging policy
- Business logging
- Business metrics
- APM
- Dashboards
- Centralized reporting
- Automation, declarative approach for IaaC

Comprehensive logging, tracing, and telemetry;
Automated operations;
Telemetry-driven health scoring

Data Management & Integrity

- GRS, GRS-RA, GZRS-RA
- RBAC for data plane
- Data Security Posture Management
- Data classification, labelling
- Data quality

Multi-region data replication strategies;
Clear data models;
Secure data access patterns.

Maintainabiliy & Modularity

- Whitelisted services (security baseline, operations, DR, know-how)
- Well known protocols
- Layered, modular architecture

Well-structured, modular architecture;
Evolvability ;
Use of standardized services and frameworks.

Services...

- Azure Container Apps
- Azure Container Registry
- Document Intelligence
- Azure OpenAI
- Search Service
- Log Analytics Workspace, Azure Monitor
- Storage account

+ Managed Identity

Name	Type
Failure Anomalies - appi-yacfkon67kvuu	Smart detector alert rule
capps-backend-yacfkon67kvuu	Container App
chat50-aca-env	Container Apps Environment
cog-di-yacfkon67kvuu	Document intelligence
cog-yacfkon67kvuu	Azure OpenAI
chat50acryacfkon67kvuu	Container registry
Application Insights Smart Detection	Action group
appi-yacfkon67kvuu	Application Insights
chat50-aca-identity	Managed Identity
gptkb-yacfkon67kvuu-identity	Managed Identity
msi-azure-search-openai-demo	Managed Identity
log-yacfkon67kvuu	Log Analytics workspace
dash-yacfkon67kvuu	Shared dashboard
gptkb-yacfkon67kvuu	Search service
styacfkon67kvuu	Storage account

Azure Container Apps – selected aspects…

Category	Feature	Supported	Enabled by Default	Responsibility
Network Security – NS-1	Virtual Network Integration	True	False	Customer
	Network Security Group Support	True	False	Customer
Network Security – NS-2	Disable Public Network Access	True	False	Customer
Identity Management – IM-1	Azure AD Authentication (Data Plane)	True	False	Customer
Identity Management – IM-3	Managed Identities	True	False	Customer
	Service Principals	True	False	Customer
Identity Management – IM-8	Key Vault for Secrets	True	Applicable	Applicable
Data Protection – DP-3	Data in Transit Encryption	True	False	Customer
Data Protection – DP-4	Data at Rest Encryption – Platform Keys	True	True	Microsoft
Asset Management – AM-2	Azure Policy Support	True	False	Customer
Logging & Threat Detection – LT-4	Azure Resource Logs	True	True	Microsoft
Posture & Vulnerability – PV-3	Custom Container Images	True	False	Customer

https://learn.microsoft.com/en-us/security/benchmark/azure/baselines/azure-container-apps-security-baseline

Azure Container Registry – selected aspects…

Category	Feature	Supported	Enabled by Default	Responsibility
NS-2 Network	Azure Private Link	**True**	**False**	**Customer**
NS-2 Network Controls	Disable Public Network Access	**True**	**False**	**Customer**
IM-1 Identity	Azure AD Authentication (Data Plane)	True	True	Microsoft
PA-7 Privileged Access	Azure RBAC (Data Plane)	True	True	Microsoft
PA-8 Privileged Access	Customer Lockbox	**True**	**False**	**Customer**
DP-2 Data Protection	Data Leakage / Loss Prevention (DLP)	**True**	**False**	**Customer**
DP-3 Data Protection	Data-in-Transit Encryption	True	True	Microsoft
DP-4 Data Protection	Data-at-Rest Encryption (Platform Keys)	True	True	Microsoft
DP-5 Data Protection	Data-at-Rest Encryption using Customer-Managed Keys (CMK)	**True**	**False**	**Customer**
AM-2 Asset Management	Azure Policy Support	**True**	**False**	**Customer**
LT-1 Threat Detection	Microsoft Defender for Containers (ACR)	**True**	**False**	**Customer**
LT-4 Logging	Azure Resource Logs	**True**	**False**	**Customer**

https://learn.microsoft.com/en-us/security/benchmark/azure/baselines/container-registry-security-baseline

AI Services: Search Service, Document Intelligence – selected aspects…

Control Domain	ASB ID	ASB Control Title	Responsibility	Feature Name	Feature Supported	Feature Enabled by Default
Identity Management	IM-1	Use centralized identity and authentication system	**Customer**	Azure AD Authentication Required for Data Plane Access	**True**	**False**
Identity Management	IM-8	Restrict the exposure of credential and secrets	**Customer**	Service Credential and Secrets Support Integration and Storage in Azure Key Vault	**True**	**False**
Asset Management	AM-2	Use only approved services	**Customer**	Azure Policy Support	**True**	**False**
Privileged Access	PA-7	Follow just enough administration (least privilege) principle	**Customer**	Azure RBAC for Data Plane	**True**	**False**
Identity Management	IM-7	Restrict resource access based on conditions	**Customer**	Conditional Access for Data Plane	**True**	**False**
Privileged Access	PA-8	Choose approval process for third-party support	**Customer**	Customer Lockbox	**True**	**False**
Data Protection	DP-5	Use customer-managed key option in data at rest encryption when required	**Customer**	Data at Rest Encryption Using CMK	**True**	**False**
Data Protection	DP-4	Enable data at rest encryption by default	Microsoft	Data at Rest Encryption Using Platform Keys	True	True
Data Protection	DP-3	Encrypt sensitive data in transit	Microsoft	Data in Transit Encryption	True	True
Data Protection	DP-2	Monitor anomalies and threats targeting sensitive data	**Customer**	Data Leakage/Loss Prevention	**True**	**False**
Network Security	NS-2	Secure cloud services with network controls	**Customer**	Disable Public Network Access	**True**	**False**
Data Protection	DP-6	Use a secure key management process	**Customer**	Key Management in Azure Key Vault	**True**	**False**
Identity Management	IM-3	Manage application identities securely and automatically	**Customer**	Managed Identities	**True**	**False**
Network Security	NS-2	Secure cloud services with network controls	**Customer**	Azure Private Link	**True**	**False**
Logging and threat detection	LT-4	Enable network logging for security investigation	**Customer**	Azure Resource Logs	**True**	**False**
Identity Management	IM-3	Manage application identities securely and automatically	**Customer**	Service Principals	**True**	**False**

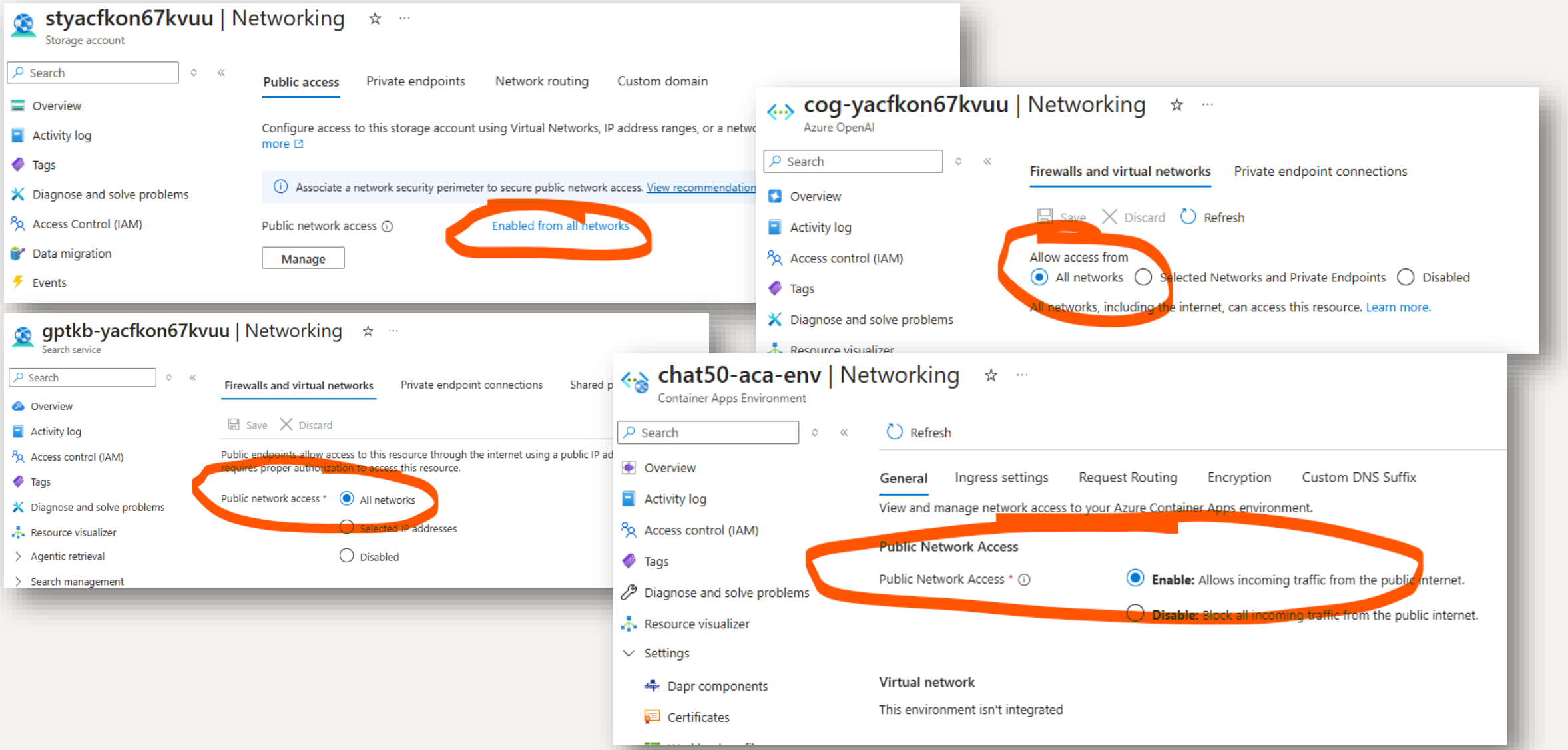
Azure OpenAI – selected aspects…

Control Domain	ASB ID	ASB Control Title	Responsibility	Feature Name	Feature Supported	Feature Enabled by Default
Identity Management	IM-1	Use centralized identity and authentication system	Microsoft	Azure AD Authentication Required for Data Plane Access	True	True
Identity Management	IM-8	Restrict the exposure of credential and secrets	**Customer**	Service Credential and Secrets Support Integration and Storage in Azure Key Vault	**True**	**False**
Asset Management	AM-2	Use only approved services	**Customer**	Azure Policy Support	**True**	**False**
Privileged Access	PA-7	Follow just enough administration (least privilege) principle	**Customer**	Azure RBAC for Data Plane	**True**	**False**
Identity Management	IM-7	Restrict resource access based on conditions	**Customer**	Conditional Access for Data Plane	**True**	**False**
Privileged Access	PA-8	Choose approval process for third-party support	**Customer**	Customer Lockbox	**True**	**False**
Data Protection	DP-5	Use customer-managed key option in data at rest encryption when required	**Customer**	Data at Rest Encryption Using CMK	**True**	**False**
Data Protection	DP-4	Enable data at rest encryption by default	Microsoft	Data at Rest Encryption Using Platform Keys	True	True
Data Protection	DP-3	Encrypt sensitive data in transit	Microsoft	Data in Transit Encryption	True	True
Data Protection	DP-2	Monitor anomalies and threats targeting sensitive data	**Customer**	Data Leakage/Loss Prevention	**True**	**False**
Network Security	NS-2	Secure cloud services with network controls	**Customer**	Disable Public Network Access	**True**	**False**
Data Protection	DP-6	Use a secure key management process	**Customer**	Key Management in Azure Key Vault	**True**	**False**
Identity Management	IM-1	Use centralized identity and authentication system	**Customer**	Local Authentication Methods for Data Plane Access	**True**	**False**
Identity Management	IM-3	Manage application identities securely and automatically	**Customer**	Managed Identities	**True**	**False**
Network Security	NS-2	Secure cloud services with network controls	**Customer**	Azure Private Link	**True**	**False**
Logging and threat detection	LT-4	Enable network logging for security investigation	**Customer**	Azure Resource Logs	**True**	**False**
Identity Management	IM-3	Manage application identities securely and automatically	**Customer**	Service Principals	**True**	**False**

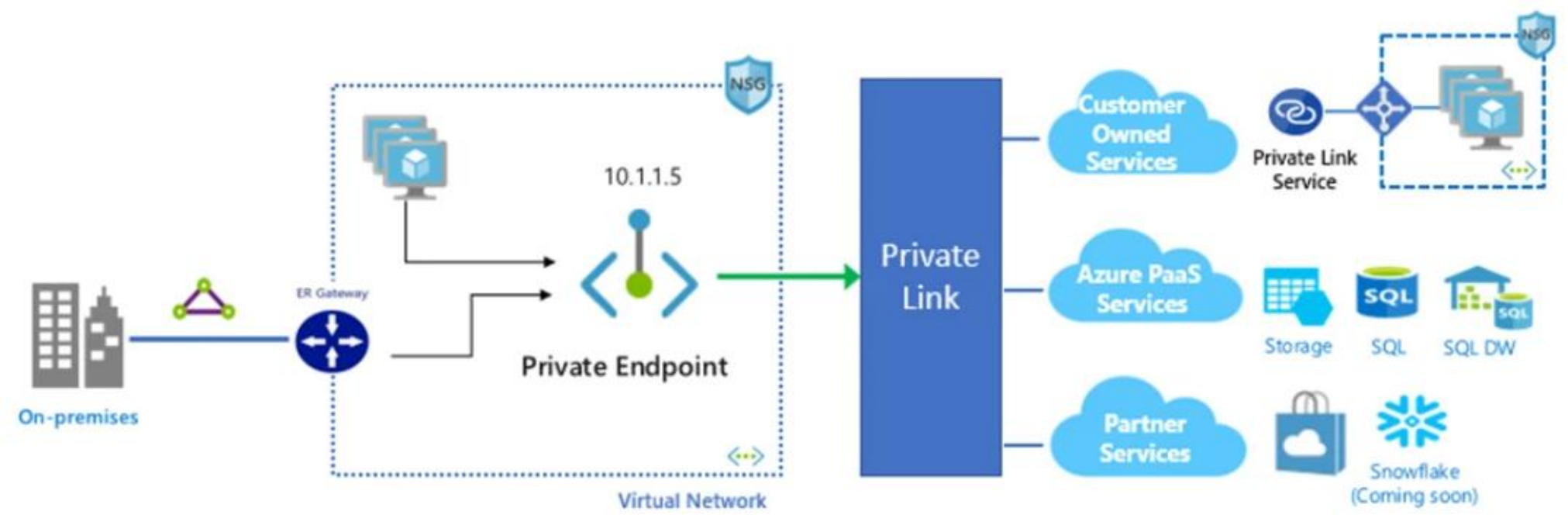
Storage Account – selected aspects…

Control Domain	ASB ID	ASB Control Title	Responsibility	Feature Name	Feature Supported	Feature Enabled by Default
Identity Management	IM-1	Use centralized identity and authentication system	Microsoft	Azure AD Authentication Required for Data Plane Access	True	True
Identity Management	IM-8	Restrict the exposure of credential and secrets	**Customer**	Service Credential and Secrets Support Integration and Storage in Azure Key Vault	**True**	**False**
Backup and recovery	BR-1	Ensure regular automated backups	**Customer**	Azure Backup	**True**	**False**
Asset Management	AM-2	Use only approved services	**Customer**	Azure Policy Support	**True**	**False**
Privileged Access	PA-7	Follow just enough administration (least privilege) principle	**Customer**	Azure RBAC for Data Plane	**True**	**False**
Identity Management	IM-7	Restrict resource access based on conditions	**Customer**	Conditional Access for Data Plane	**True**	**False**
Privileged Access	PA-8	Choose approval process for third-party support	**Customer**	Customer Lockbox	**True**	**False**
Data Protection	DP-5	Use customer-managed key option in data at rest encryption when required	**Customer**	Data at Rest Encryption Using CMK	**True**	**False**
Data Protection	DP-4	Enable data at rest encryption by default	Microsoft	Data at Rest Encryption Using Platform Keys	True	True
Data Protection	DP-3	Encrypt sensitive data in transit	Microsoft	Data in Transit Encryption	True	True
Data Protection	DP-2	Monitor anomalies and threats targeting sensitive data	**Customer**	Data Leakage/Loss Prevention	**True**	**False**
Logging and threat detection	LT-1	Enable threat detection capabilities	**Customer**	Microsoft Defender for Service / Product Offering	**True**	**False**
Network Security	NS-2	Secure cloud services with network controls	**Customer**	Disable Public Network Access	**True**	**False**
Data Protection	DP-6	Use a secure key management process	**Customer**	Key Management in Azure Key Vault	**True**	**False**
Identity Management	IM-3	Manage application identities securely and automatically	**Customer**	Managed Identities	**True**	**False**
Backup and recovery	BR-1	Ensure regular automated backups	**Customer**	Service Native Backup Capability	**True**	**False**
Network Security	NS-2	Secure cloud services with network controls	**Customer**	Azure Private Link	**True**	**False**
Logging and threat detection	LT-4	Enable network logging for security investigation	**Customer**	Azure Resource Logs	**True**	**False**
Data Protection	DP-1	Discover, classify, and label sensitive data	**Customer**	Sensitive Data Discovery and Classification	**True**	**False**

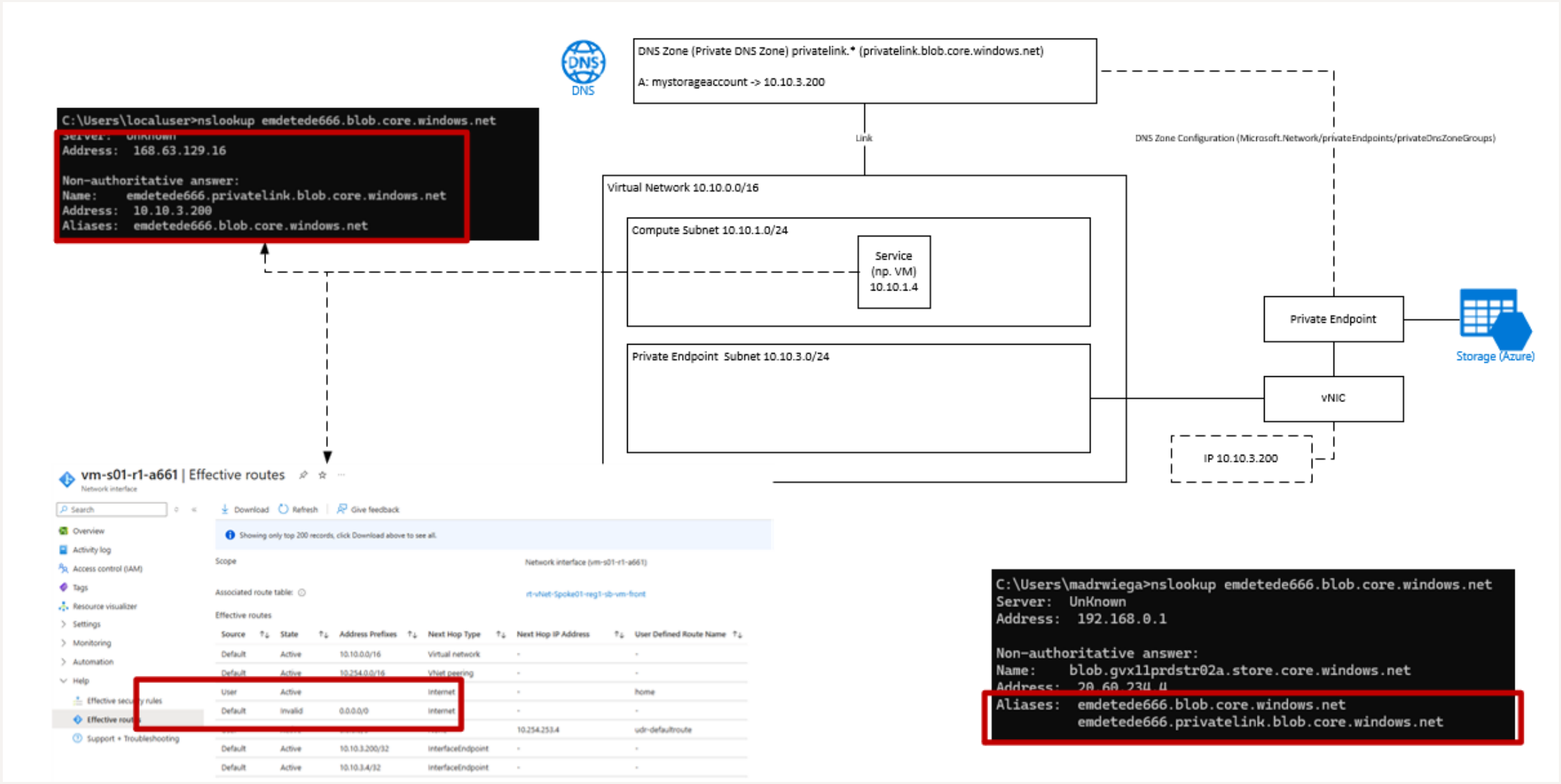
Network isolation – “our” case OOTB



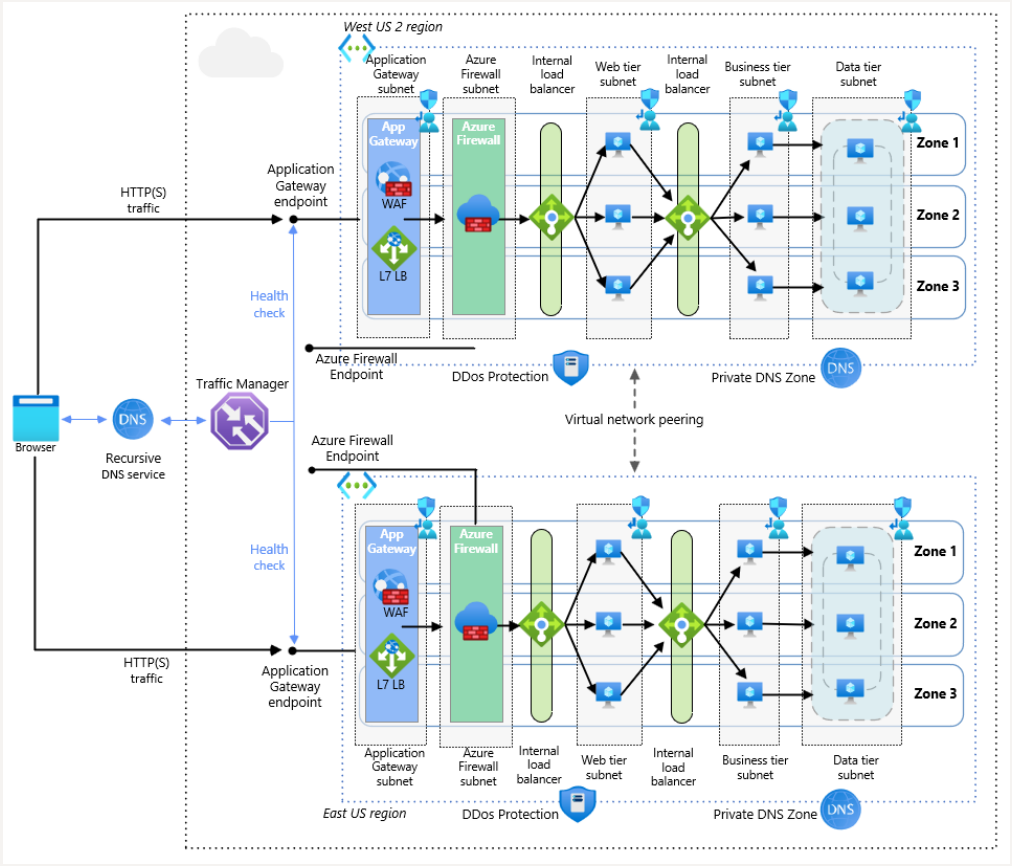
Network isolation – Private Endpoint

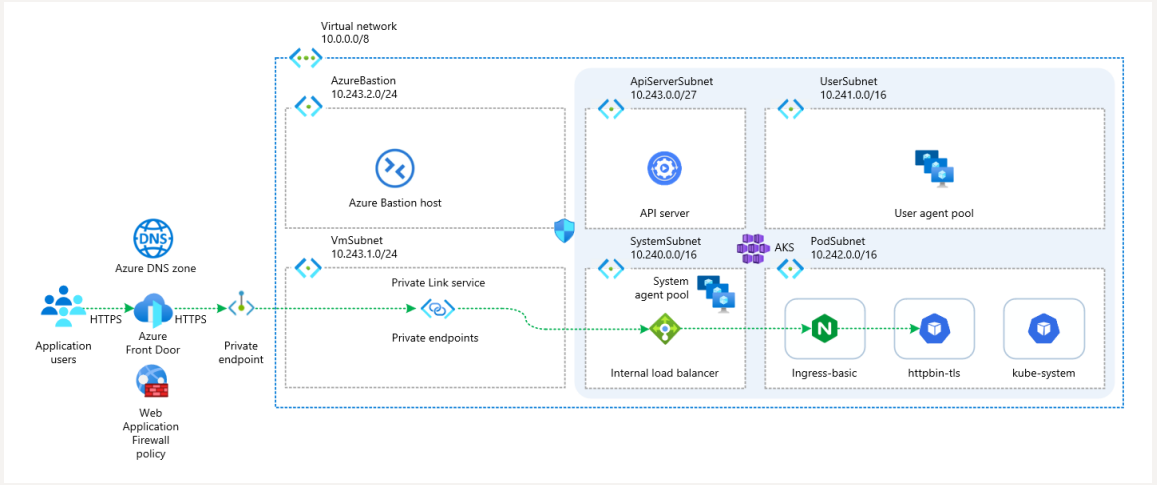


Private Endpoint – a bit more practical

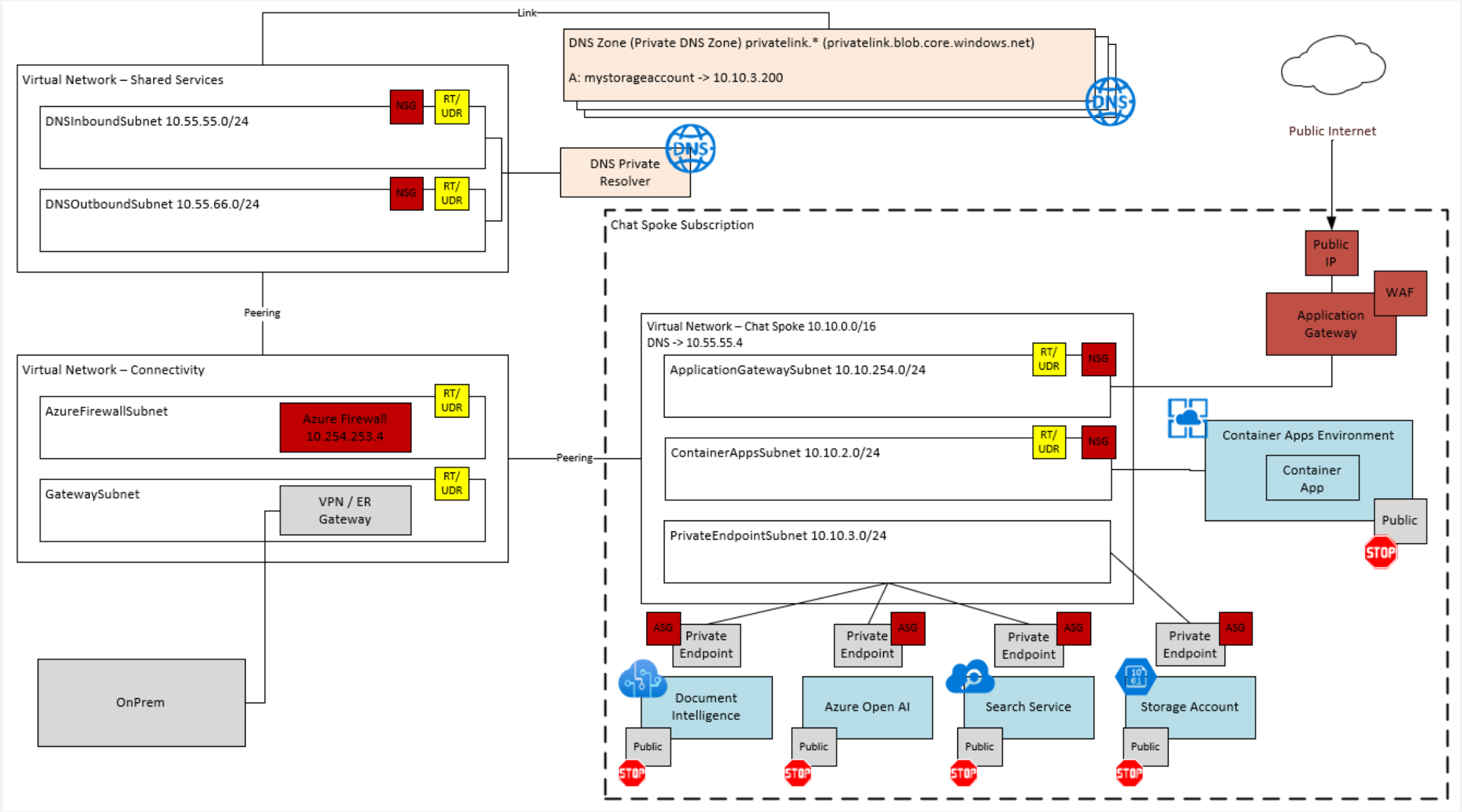


App publishing – never directly to the Internet!





Networking isolation for “our” case, a bit more enterprise scale…

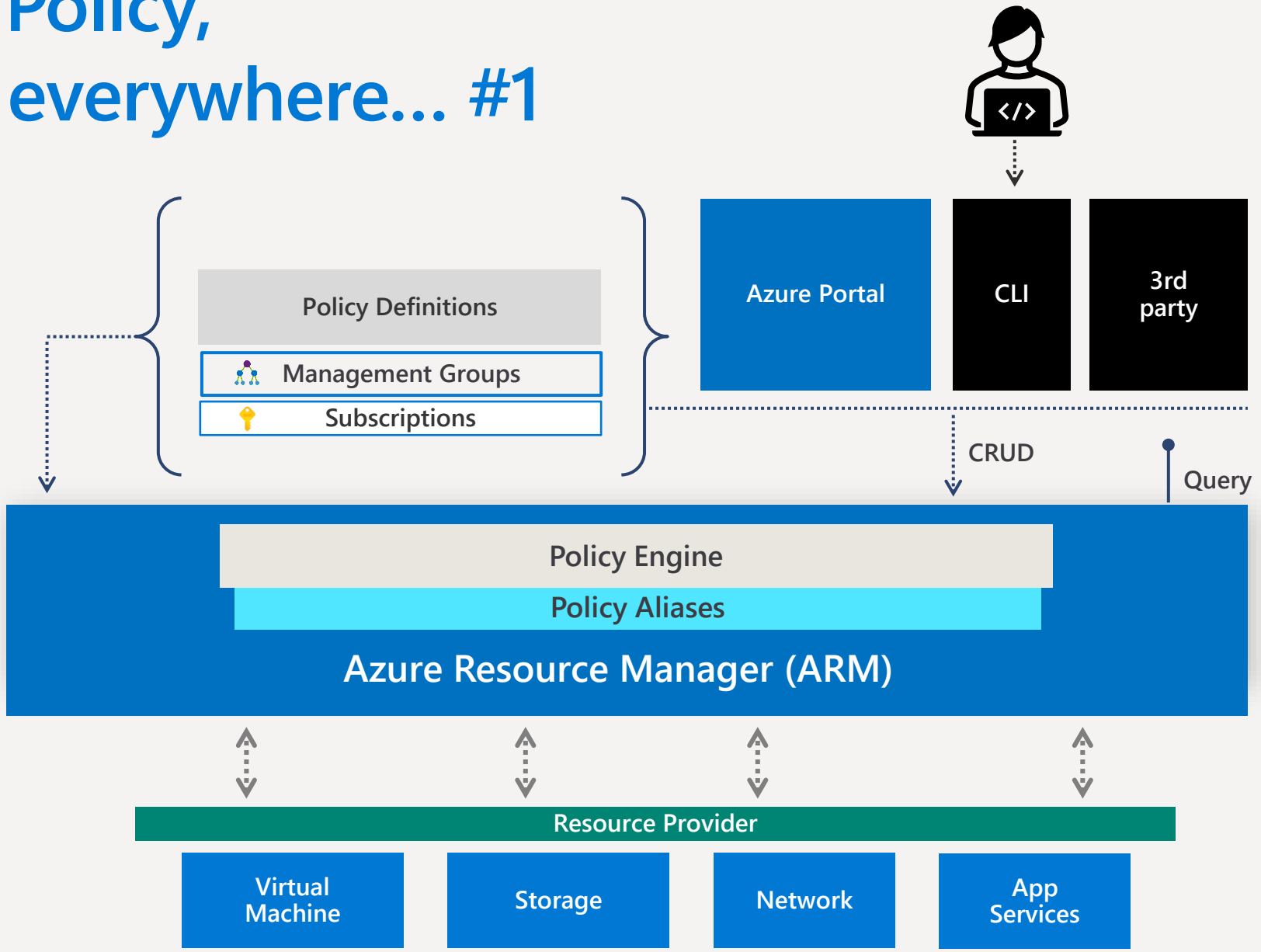


How? – Azure Policy, Azure Policies everywhere… #1

Policy Definitions
Map to business risks and compliance requirements

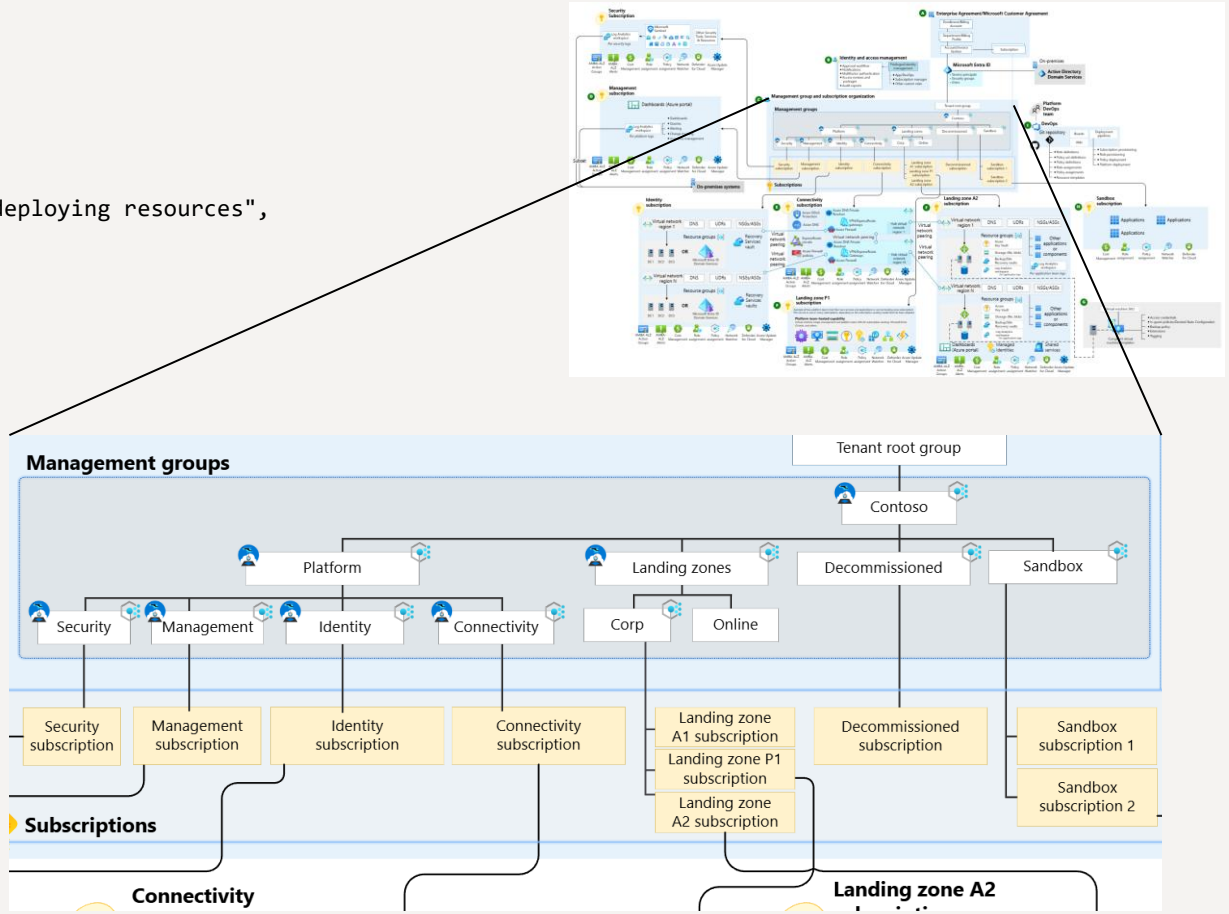
Policy Engine
Real-time enforcement, compliance assessment and remediation at scale

Policy Aliases
Mapping between the policy engine and the resource provider, resource type property, and API version

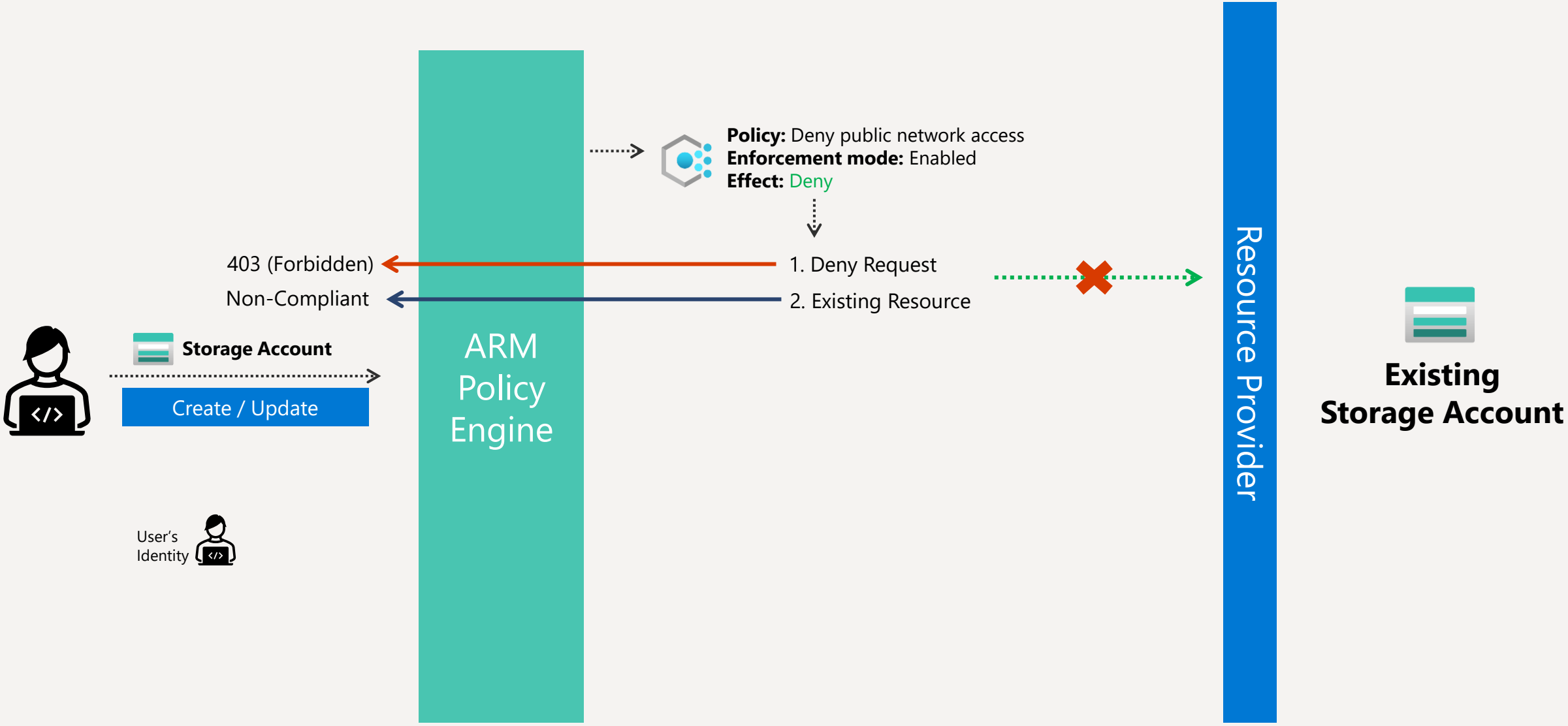


Azure Policy - How? – Definition, Assignement, Scope

```
{
  "properties": {
    "displayName": "Allowed locations",
    "description": "This policy enables you to restrict the locations your organization can specify when deploying resources.",
    "mode": "Indexed",
    "metadata": {
      "version": "1.0.0",
      "category": "Locations"
    },
    "parameters": {
      "allowedLocations": {
        "type": "array",
        "metadata": {
          "description": "The list of locations that can be specified when deploying resources",
          "strongType": "location",
          "displayName": "Allowed locations"
        },
        "defaultValue": [
          "westus2"
        ]
      }
    },
    "policyRule": {
      "if": {
        "not": {
          "field": "location",
          "in": "[parameters('allowedLocations')]"
        }
      },
      "then": {
        "effect": "deny"
      }
    }
  }
}
```



Azure Policy – Deny Action



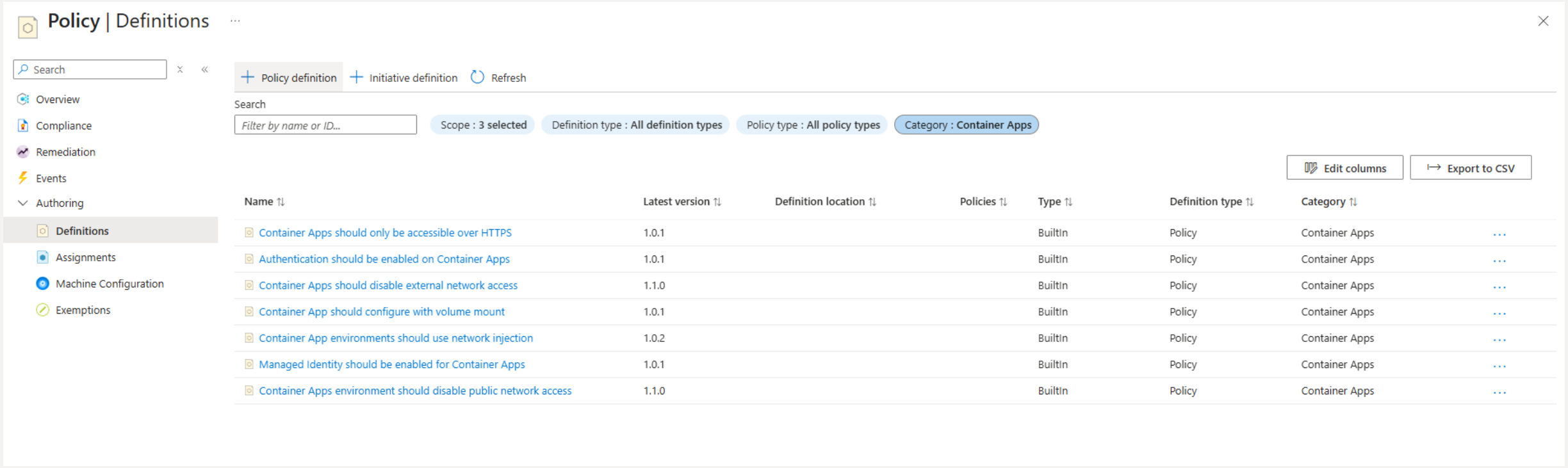
Managed Identity – selected aspects…

- System Assigned
- User Assigned?
- Service Principal??

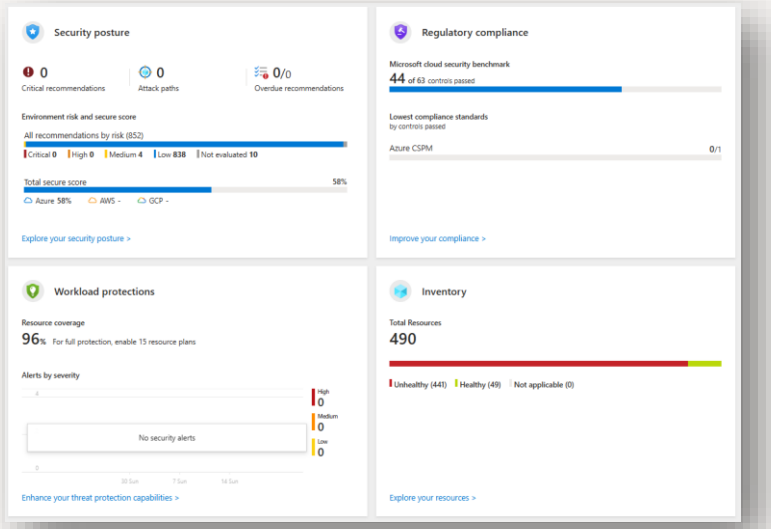
Microsoft Entra Workload ID

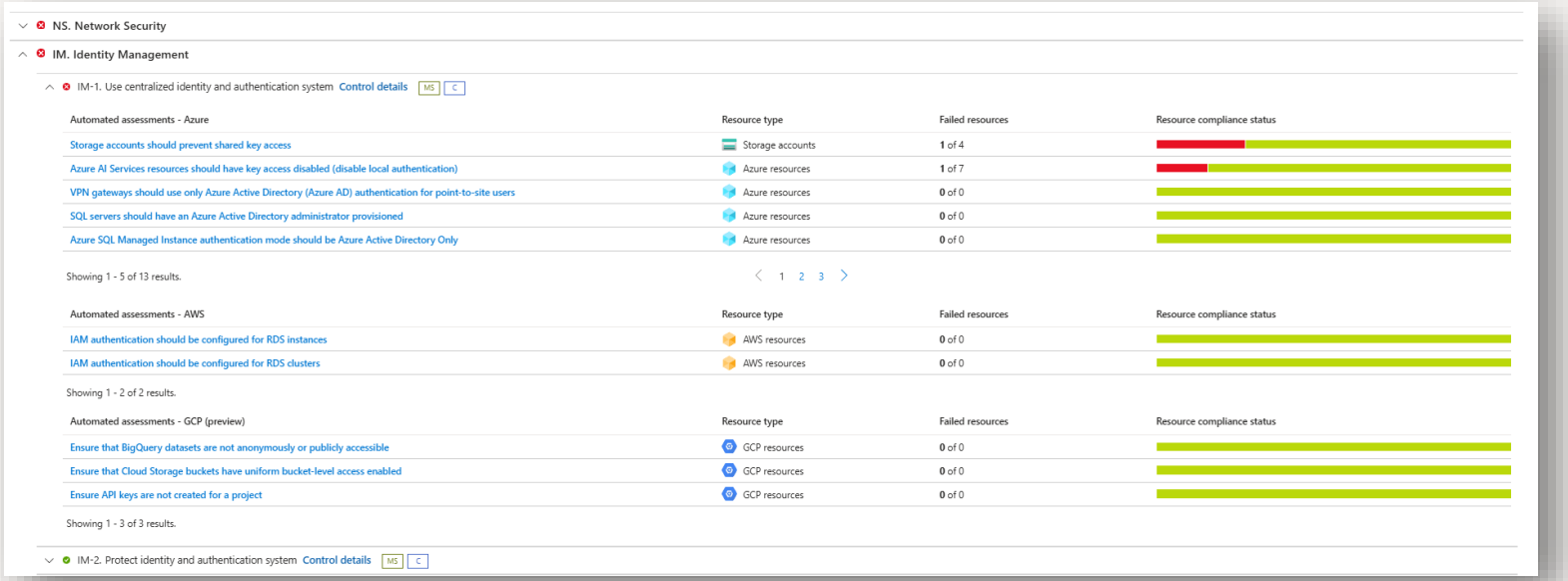
- Adaptive policies:
  - Conditional Access for workload identities
  - real-time enforcement of Conditional Access location and risk policies using Continuous access evaluation for workload identities
  - Manage custom security attributes for an app
- Detect compromised identities:
  - Detect risks (like leaked credentials), contain threats, and reduce risk to workload identities using Microsoft Entra ID Protection
- Lifecycle management:
  - access reviews for service principals
  - workload identity federation (external IdP to Managed Identity or App Registration in Azure)

Azure Policies – built in…



Defender for Cloud

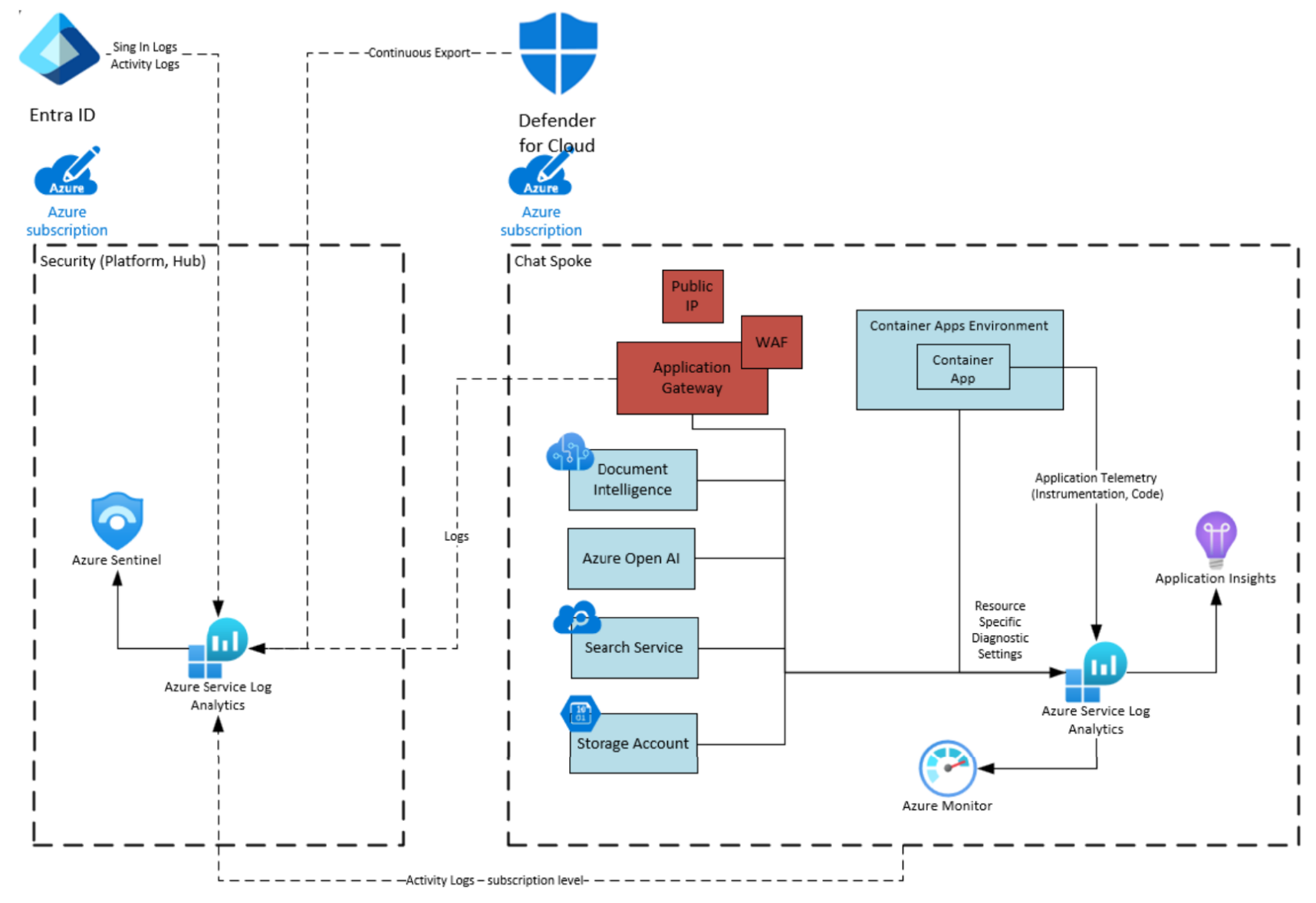




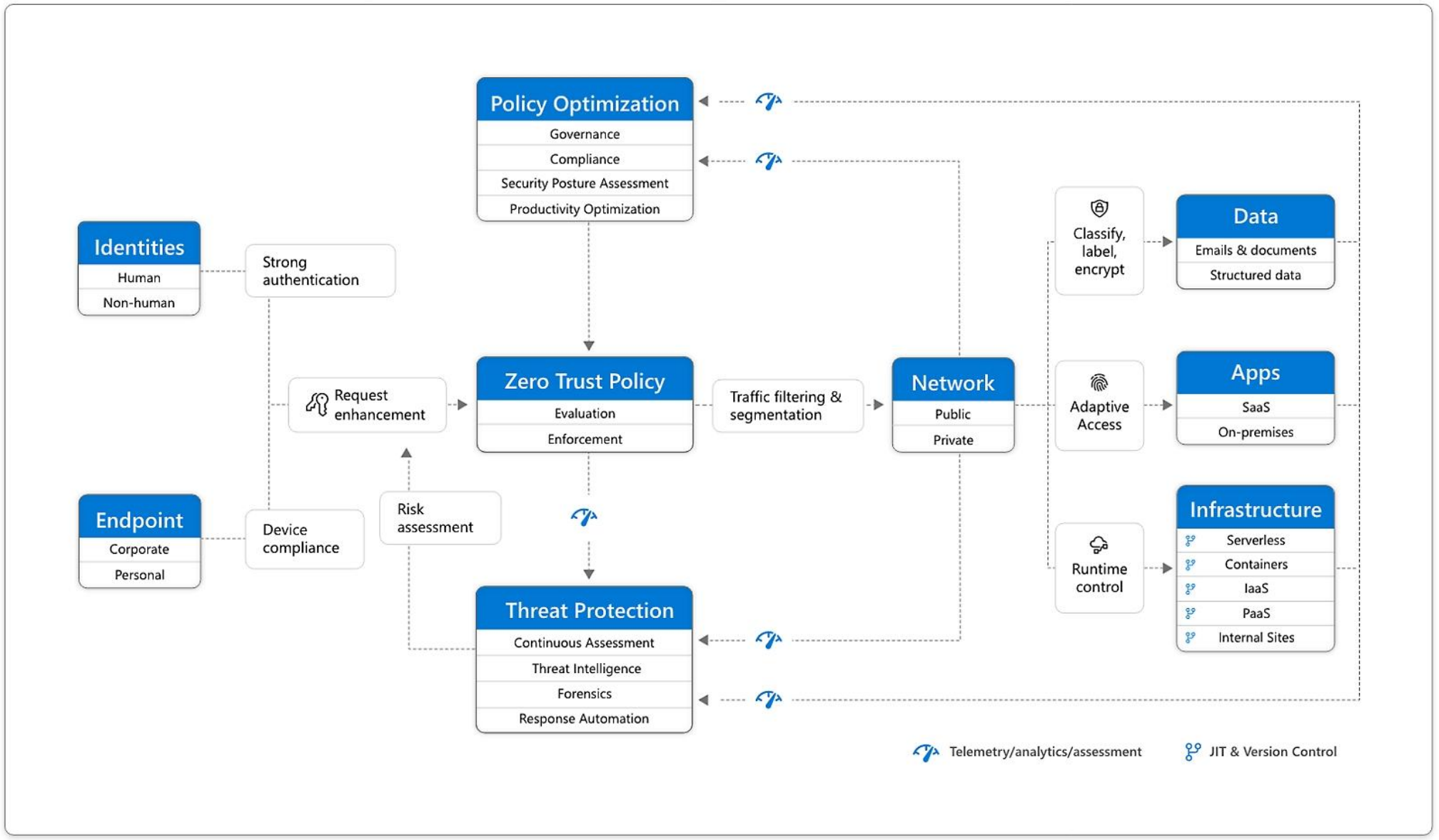
Cloud Security Posture Management (CSPM): Free baseline; paid advanced capabilities.

- Defender for Servers: Plan 1 (EDR), Plan 2 (agentless, FIM, JIT)
- Defender for Containers: AKS, registries, images
- Defender for Storage: Malware scanning, sensitive data discovery
- Defender for Databases: SQL, PostgreSQL, MySQL, MariaDB, Cosmos DB
- Defender for App Service: Web app threat detection

Logging, monitoring, observability – in practice…



Zero Trust


Policy Optimization
Governance
Compliance
Security Posture Assessment
Productivity Optimization
Identities
Human
Non-human
Strong authentication
Request enhancement
Zero Trust Policy
Evaluation
Enforcement
Traffic filtering & segmentation
Network
Public
Private
Classify, label, encrypt
Data
Emails & documents
Structured data
Adaptive Access
Apps
SaaS
On-premises
Runtime control
Infrastructure
Serverless
Containers
IaaS
PaaS
Internal Sites
Endpoint
Corporate
Personal
Device compliance
Risk assessment
Threat Protection
Continuous Assessment
Threat Intelligence
Forensics
Response Automation
Telemetry/analytics/assessment
JIT & Version Control