

Securing a Smooth Landing with AI

November 28th 2024



AJ Bajada

Azure, DevOps and automation enthusiast

And of course... Star Wars!

GitHub handle: tw3lveparsecs



As ChatGPT sees my life



Agenda



- Azure Landing Zones
- Deployment Types
- Quotas and Limits
- Models
- Tokens
- Rate Limits
- Authentication Methods
- Architecture

Azure Landing Zones



An Azure Landing Zone is a pre-configured environment



Designed to facilitate the onboarding of workloads (specifically OpenAI solutions in this case)



Provides a scalable and secure foundation for your AI workloads

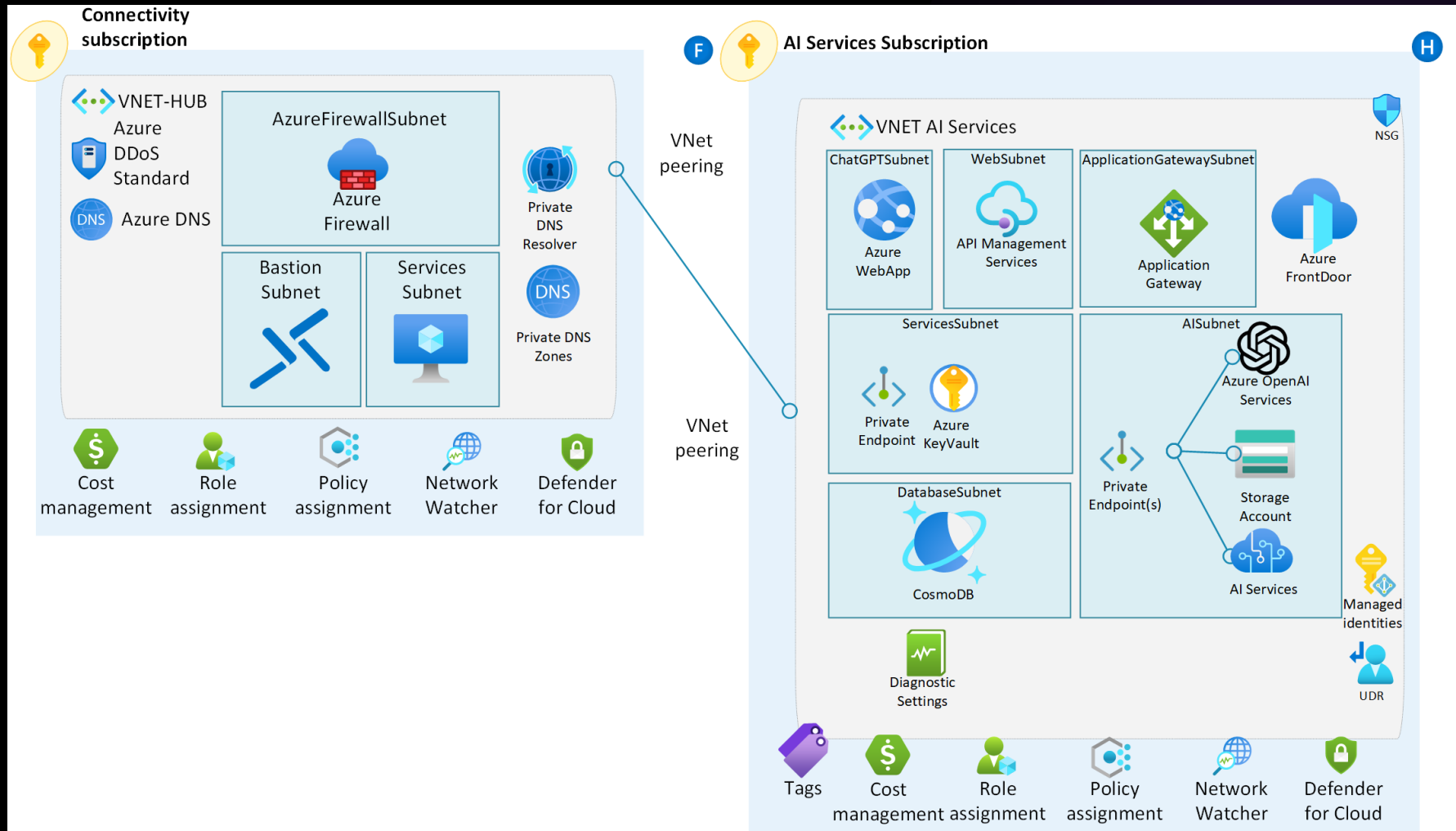


Ensures compliance and governance guardrails are in place



Enables rapid deployment and operations

Azure Landing Zones



Deployment Types

Azure OpenAI offers three types of deployments.

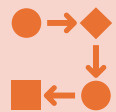
1. Global
2. Standard
3. Provisioned

These provide a varied level of capabilities that provide trade-offs on: throughput, SLAs, and price.

Deployment Types - Standard



Standard deployments provide a pay-per-call billing model on the chosen model.



Models available in each region as well as throughput may be limited.



Standard deployments are optimised for low to medium volume workloads with high burstiness.

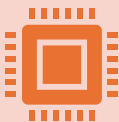
Deployment Types - Provisioned



Provisioned deployments allow you to specify the amount of throughput you require in a deployment



The service then allocates the necessary model processing capacity and ensures it's ready for you



Throughput is defined in terms of provisioned throughput units (PTU) which is a way of representing the throughput for your deployment

Deployment Types - Global



Enables you to leverage Azure's global infrastructure to dynamically route traffic to the data center with best availability for each request.



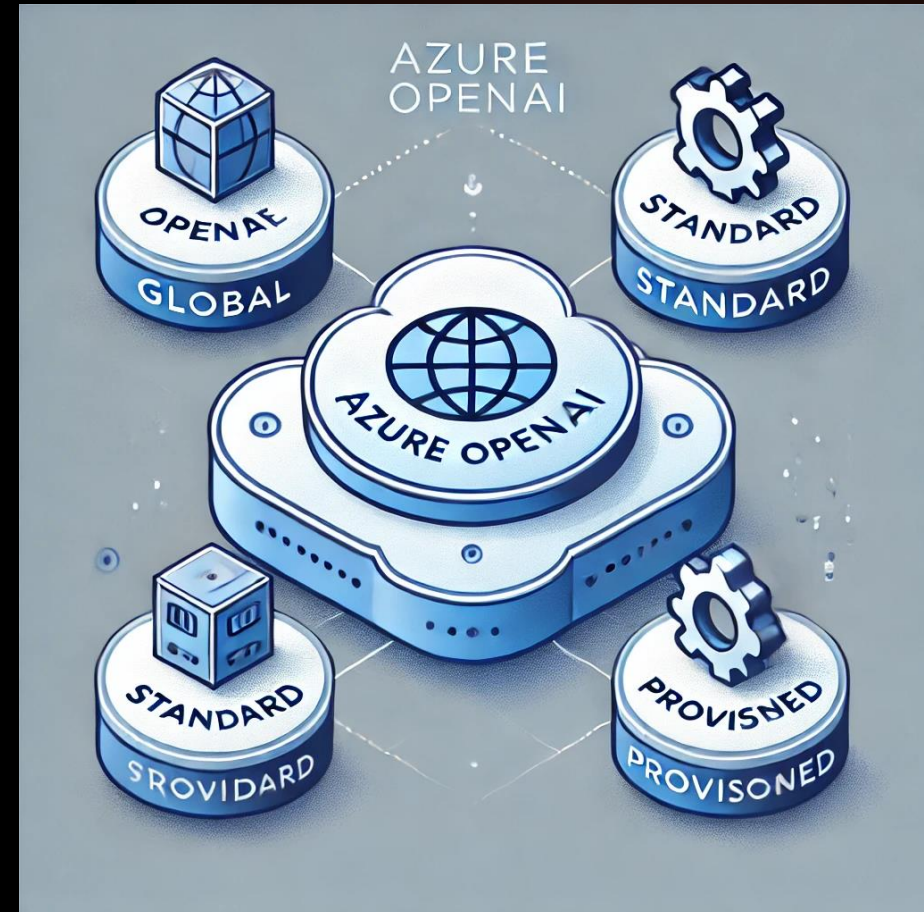
Split into Global Standard, Provisioned and Batch deployment types each with their own trade offs



Important to review each deployment type to ensure its fit for purpose

Deployment Types

- Why should I care?
 - Deployment types will change the overall architecture that gets deployed within your landing zone
 - Data residency and regional requirements
 - Availability of services and models
 - Latency considerations



Models

Models	Description
o1-preview and o1-mini	Limited access models, specifically designed to tackle reasoning and problem-solving tasks with increased focus and capability.
GPT-4o & GPT-4o mini & GPT-4 Turbo	The latest most capable Azure OpenAI models with multimodal versions, which can accept both text and images as input.
GPT-4o-Realtime-Preview	A GPT-4o model that supports low-latency, "speech in, speech out" conversational interactions.
GPT-4	A set of models that improve on GPT-3.5 and can understand and generate natural language and code.

- Azure OpenAI Service is powered by a diverse set of models with different capabilities and price points
- Model availability varies by region and cloud

Models

Models	Description
GPT-3.5	A set of models that improve on GPT-3 and can understand and generate natural language and code.
Embeddings	A set of models that can convert text into numerical vector form to facilitate text similarity.
DALL-E	A series of models that can generate original images from natural language.
Whisper	A series of models in preview that can transcribe and translate speech to text.
Text to speech (Preview)	A series of models in preview that can synthesize text to speech.

Quotas and Limits

There are
default quotas
and limits that
apply to
OpenAI

- 30 per region per subscription
- 32 max standard deployments per resource
- 6 new connections per minute
- Etc

These are well documented by Microsoft

Quotas and Limits

- There are also quotas and limits that apply depending on the Azure region and model chosen
- Example Australia East rate limits

GPT-4	GPT-4 32K	GPT-4 Turbo	GPT-4 Turbo-V	GPT-35 Turbo	GPT-4o Global Standard	GPT-4o-mini Global Standard	GPT-4-Turbo Global Standard	Text-Embedding-Ada-002
40 K	80 K	80 K	30 K	300 K	30 M	50 M	2 M	350 K

Quotas and Limits

- Why are models and quotas so important?
 - The quotas and the models required by your application determines the which region(s) are required for your deployment
 - This means you need supporting infrastructure in each of those regions (networking, logging, etc.)



Tokens

AI tokens are units of computational resources used to measure and control the consumption of AI services

In the context of OpenAI deployments, tokens represent the cost associated with processing requests and executing AI models.

Tokens



Usage Measurement: Tokens quantify the amount of computational effort required for different AI tasks, providing a standardised way to track resource consumption



Rate Limiting: Tokens help implement rate limiting to manage and control the number of requests processed within a given timeframe



Cost Control: By tracking token usage, organisations can monitor and manage costs effectively, ensuring budget adherence and financial predictability



Traceability: Token usage metrics enable detailed reporting and analysis, supporting accountability and optimisation of AI resource allocation

Tokens

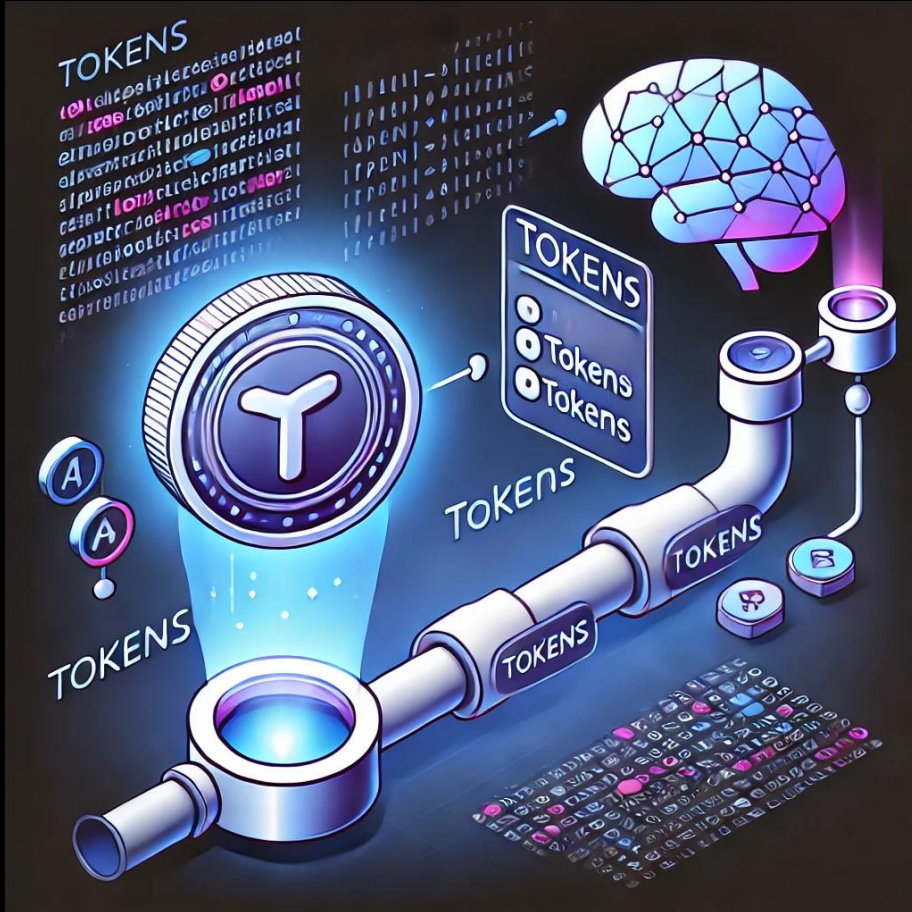
Determine a foundational token baseline for your application

In cases where its unknown, utilise a formula to determine the token baseline

Example:

$\text{<number_of_tokens>} * \text{<messages_per_minute>} * \text{<number_of_applications>}$

Tokens



- Is it essential for me to know about applications tokens?
 - The number of tokens required by your application determines the model(s) chosen
 - Tracking tokens requires additional infrastructure such as log analytics workspaces for monitoring and alerting

Rate Limits

Rate limits are mechanisms used to control the number of requests or operations that can be performed within a specific timeframe

They are essential for ensuring the stability, security, and fair usage of services

Rate Limits

Traffic Management: Prevents system overload by controlling the influx of requests

Security: Protects against abuse and malicious attacks by limiting the rate of incoming requests

Fair Usage: Ensures equitable access to resources by preventing any single user from monopolising the service

Cost Control: Helps manage costs by restricting excessive use and providing predictable resource consumption patterns

Performance Optimisation: Maintains optimal service performance and reliability by preventing bottlenecks and ensuring resources are used efficiently

Rate Limits



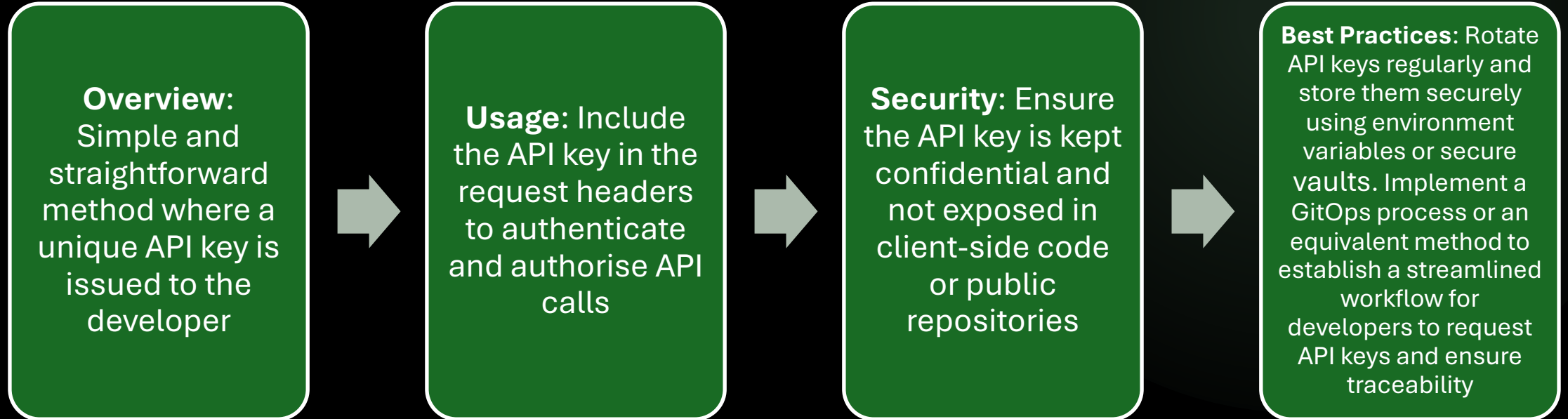
- Rates limits apply to the application, so I don't need worry, right?
 - OpenAI has built in rate limiting - this applies to all applications consuming the OpenAI instance
 - APIM or equivalent infrastructure is required to rate limit per application
 - Shared or distributed model considerations

Authentication Methods

When integrating OpenAI services, developers can use various authentication methods to ensure secure access and proper usage

1. API Key Authentication
2. Entra ID Authentication
3. Managed Identity
4. OAuth

Authentication Methods – API Key



Authentication Methods – Entra ID



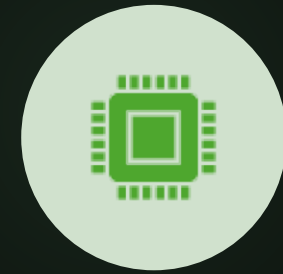
Overview: Uses Azure's identity and access management service to secure access to OpenAI services



Usage: Obtain an access token from Entra ID and include it in the request headers



Security: Provides robust security features like multi-factor authentication (MFA), conditional access policies, and role-based access control (RBAC)



Best Practices: Integrate with organisational identity management systems for seamless user authentication and authorisation

Authentication Methods – Managed Identity

Overview: Simplifies authentication by leveraging managed identities provided by Azure for Azure resources

Usage: Azure services (like VMs, App Services) use their managed identity to obtain tokens for OpenAI services without needing to manage credentials

Security: Eliminates the need to store credentials, reducing the risk of credential exposure

Best Practices: Use managed identities whenever possible for enhanced security and ease of management

Authentication Methods – OAuth


Overview: Standard authorisation framework that provides access delegation via tokens



Usage: Implement OAuth flows (like client credentials, authorisation code) to obtain access tokens for API calls



Security: Supports scopes and granular permissions, enabling fine-grained access control



Best Practices: Use secure storage for client secrets and tokens and ensure proper handling of refresh tokens

Authentication Methods

- Authentication methods, should I have some input?
- Yes! Enforce managed identities – no passwords, enhanced security
- Implement Azure policies to put guardrails in place and limit API key usage in production
 - API keys are an easy flight path for developers
 - Quickly make it into production and in some cases hard coded in application code



Architecture

When it comes to the overall architecture of OpenAI deployments, there are several key considerations

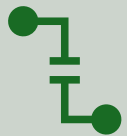


Shared Model: Deploy OpenAI in a dedicated landing zone to support shared consumption across multiple services and ensure scalability for future growth



Private Networking: Implement private networking to ensure no public access, enhancing security and compliance

Architecture



Ingress Flows with WAF: Support ingress flows through a Web Application Firewall (WAF) to protect against threats and manage traffic securely



Resiliency: Ensure high availability and resiliency by deploying two OpenAI instances in an active-passive setup, allowing for seamless failover when quotas are hit, or instances are unavailable

Architecture

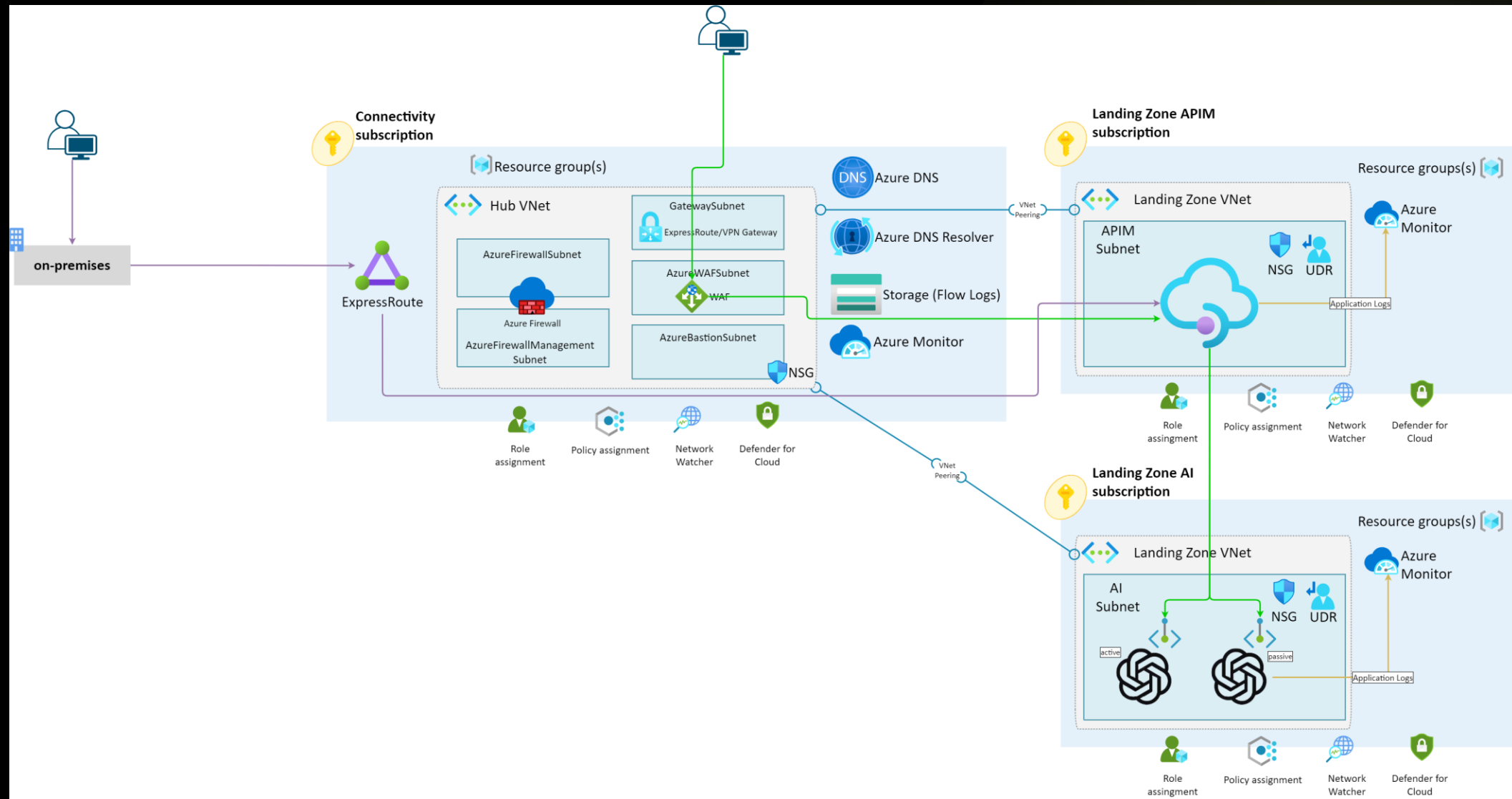


Rate Limiting and Metrics Reporting: Implement rate limiting to control the usage and integrate logging into Azure Monitor and Log Analytics for detailed reporting on token usage and other critical metrics



Cost Control and Traceability: Monitor and trace consumption effectively to maintain cost control and ensure transparent tracking of resource usage

Architecture



Demo



Up for a demo on deploying an OpenAI landing zone?

Questions

