Serverless Data Processing (CSCI 5410)

Dr. Saurabh Dey

Outline

1. Why Containerization and Serverless?

2. N-Tier Architecture

Why Containerization?

Easy Development

- Reduces development time
- Enables efficient testing
- Easy module management

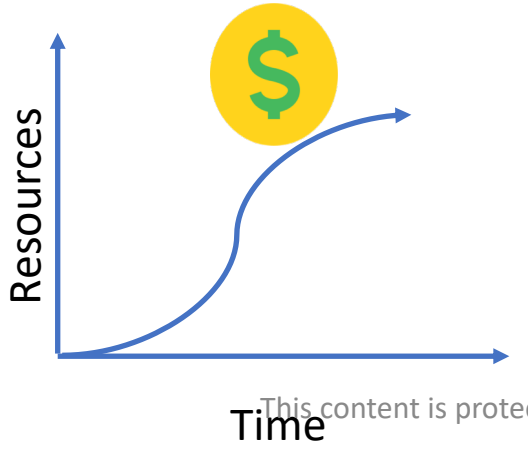
Easy Deployment

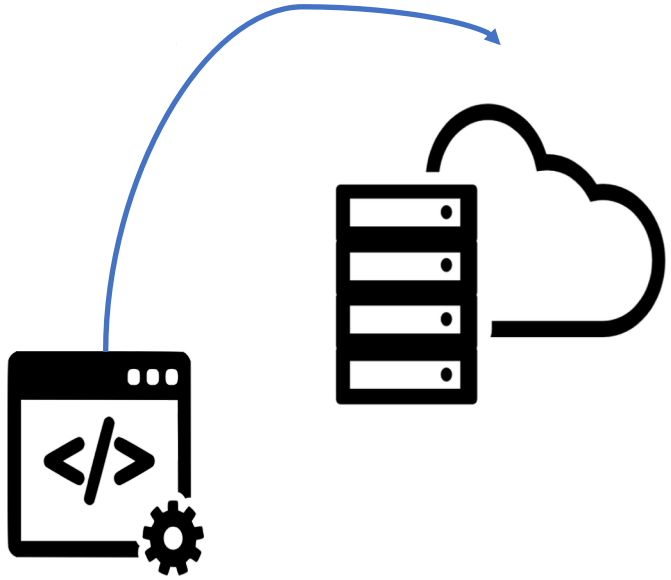
- Integration
- Selection of best in breeds

If Containerization, then why serverless?

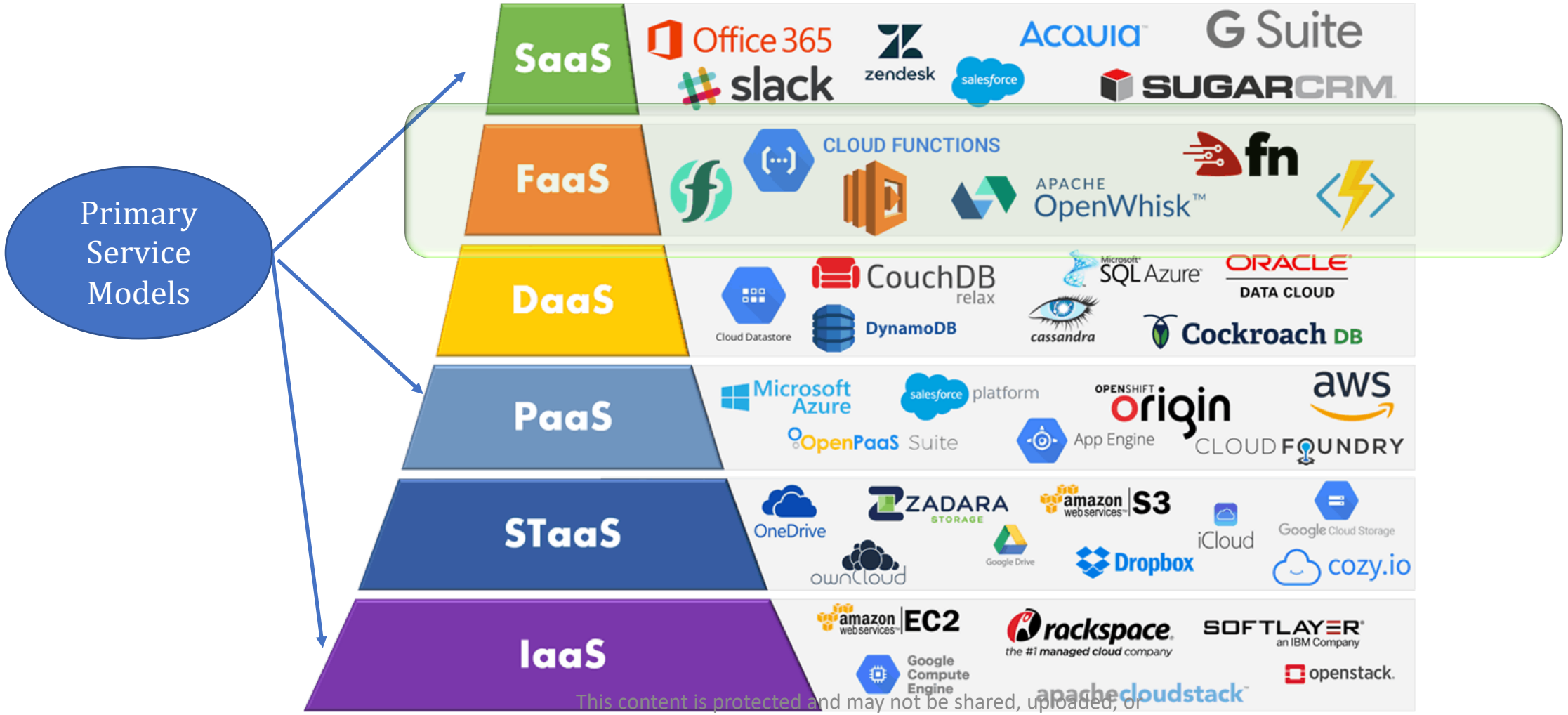
- We are working on the cloud server, and managing the server

- Cloud server is there even when we do not require.





Cloud Service Delivery Models



https://imelgrat.me/cloud/cloud-services-models-help-business/

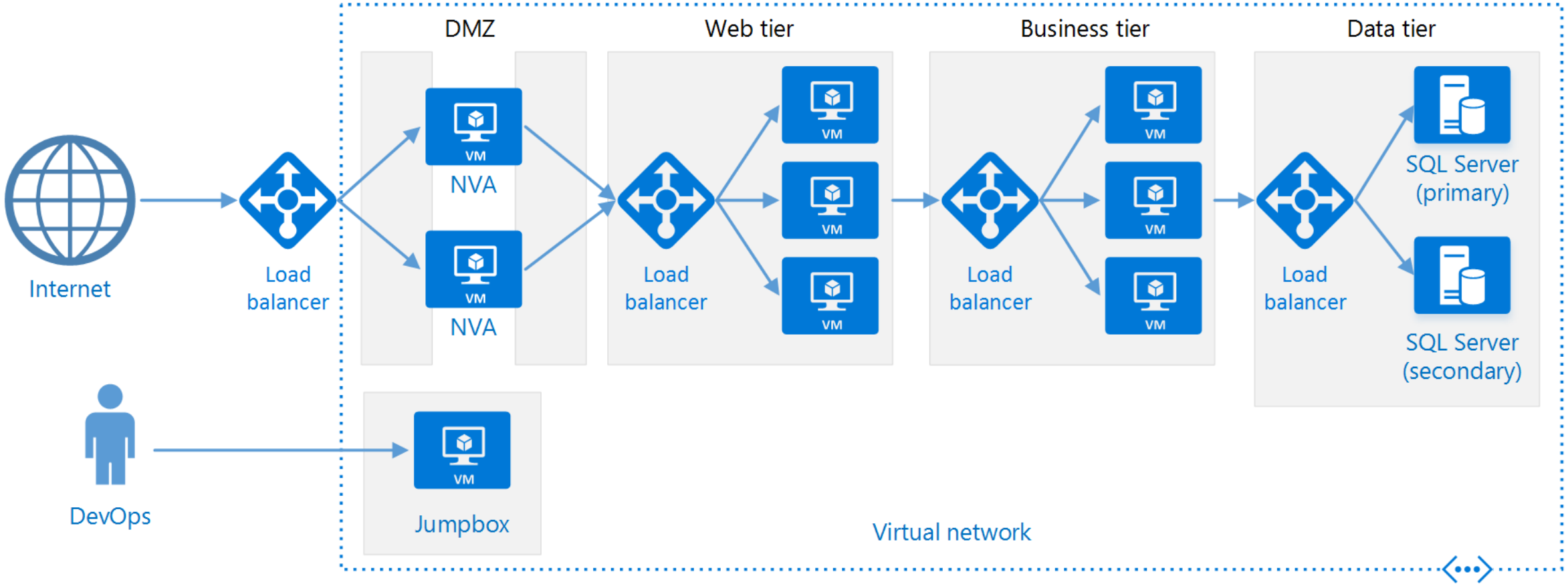
What is Serverless Computing?

Does function-as-a-service indicate an absence of a server?

- No

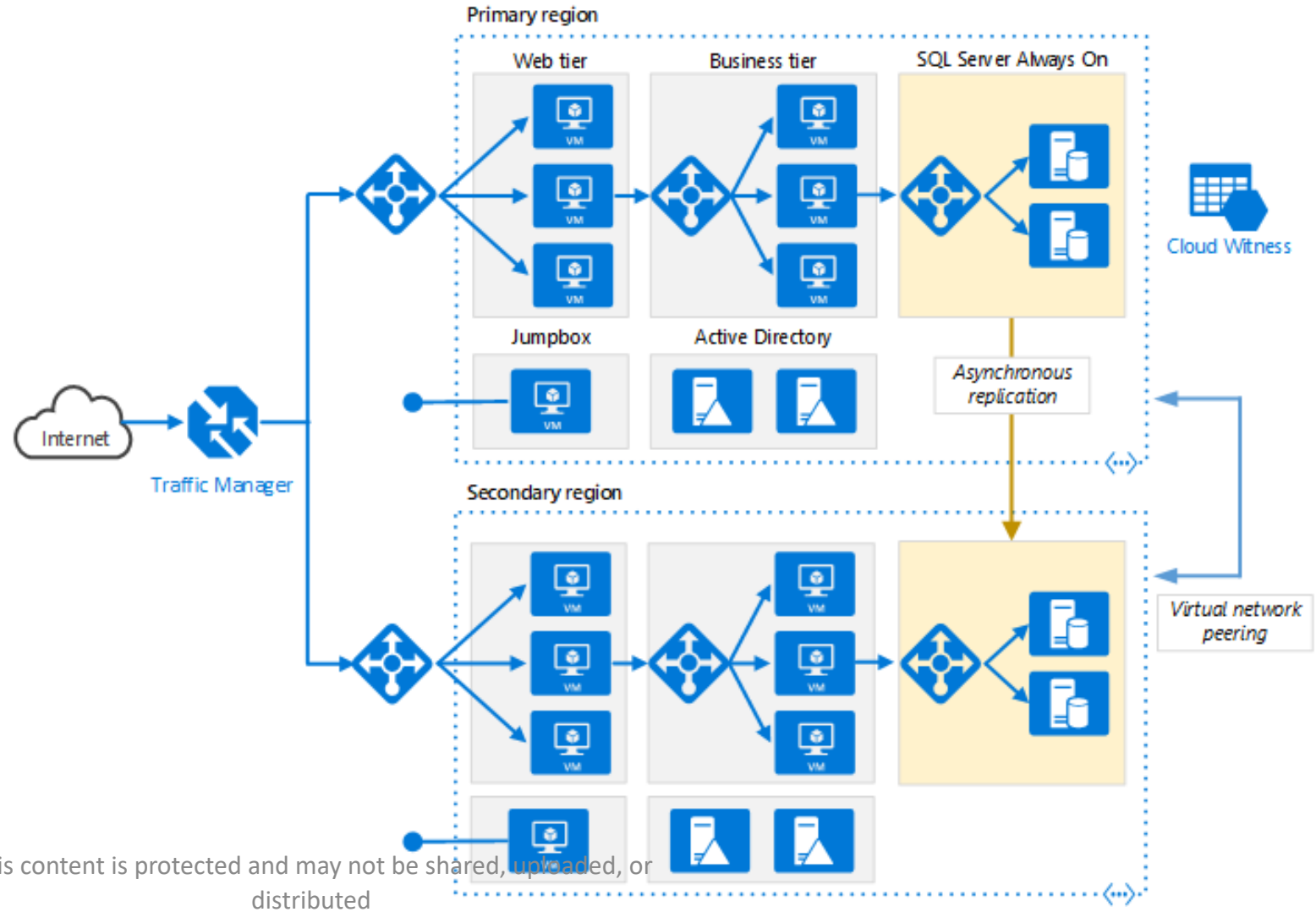
Servers are still needed, and those run in the background.

N-Tier Architecture Example



Multi-Region N-Tier

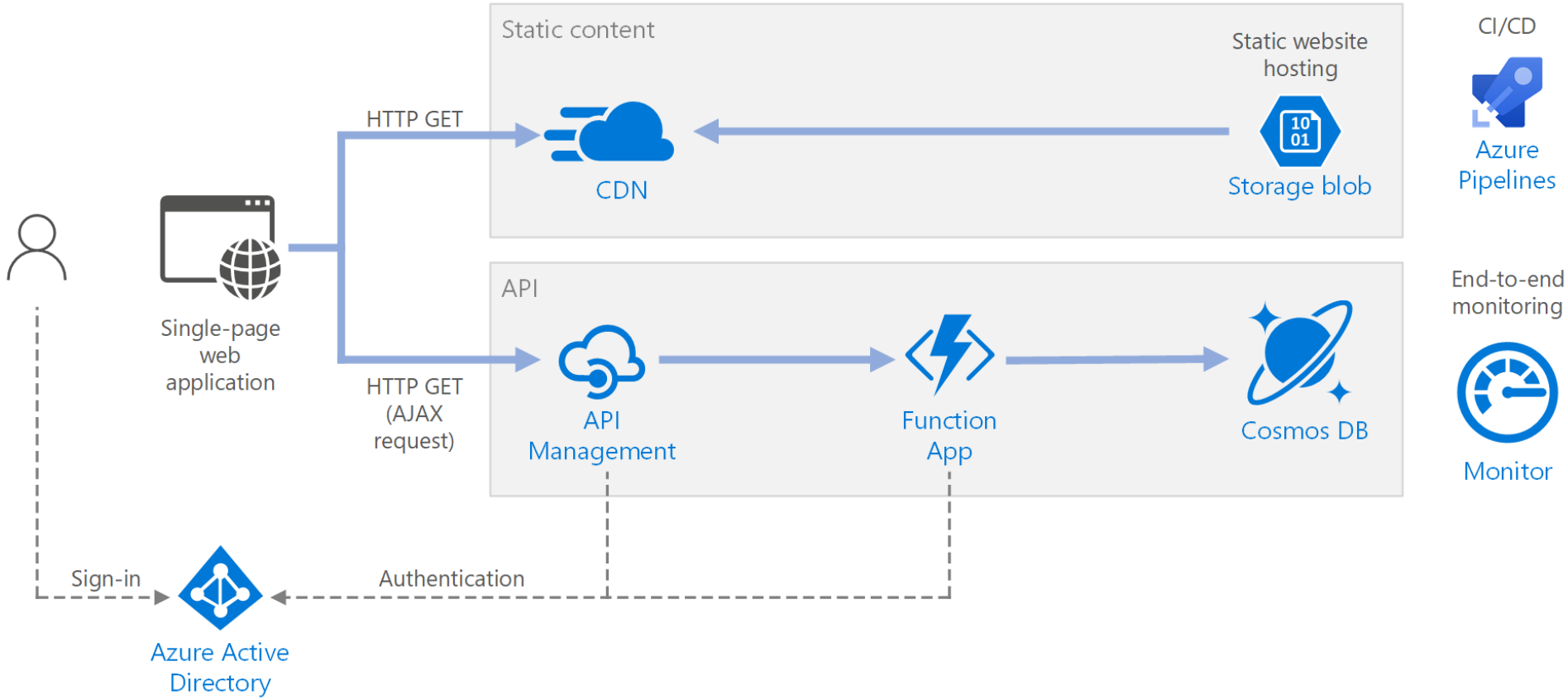
- To increase availability
- To get data replication



https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/n-tier/multi-region-sql-server

Serverless Architecture

- Abstraction of backend
- Scale automatically
- No need to configure



https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/serverless/web-app

Pros and Cons of Serverless Computing

Advantages:

- No Server Management Needed
- Pay what you use
- Inherently Scalable
- Quick Deployments and Updates

Disadvantages:

- Debugging code is challenging
- Security Issues
- Not for long-running process
- Performance Issues
- Vendor Lock-in

https://www.cloudflare.com/learning/serverless/why-use-serverless/

Questions to Consider

- Where will you implement security in the N-Tier architecture?
- Will you consider serverless data processing for continuous monitoring and processing of water /power usage data?