

DR in Azure

Memilavi
www.memilavi.com



DR

- Disaster Recovery
- A plan to recover from a complete shutdown of a Region
 - Usually as a result of a disaster (earthquake, flood, etc)
- Some apps require it, some don't
- Might have substantial cost aspects
- Remember: A complete shutdown of a Region is extremely rare

How DR Works?

- In order to set up DR, we need to do the following:
 - Select a DR site
 - A secondary Region that will function as our primary in case of a disaster
 - Configure it to be ready for activation when necessary

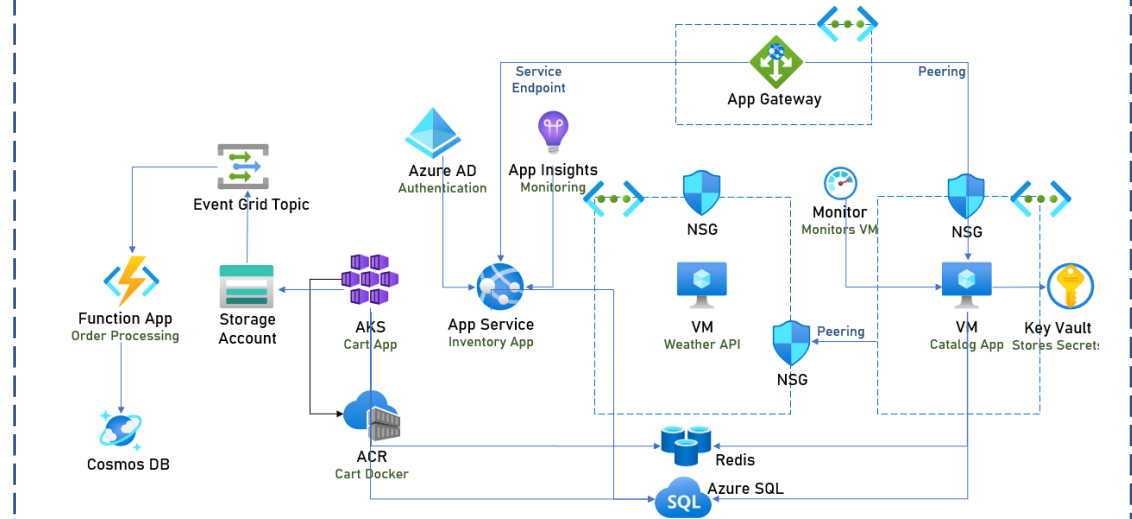
How DR Works?

West Europe Primary

GONE



North Europe Secondary – not used



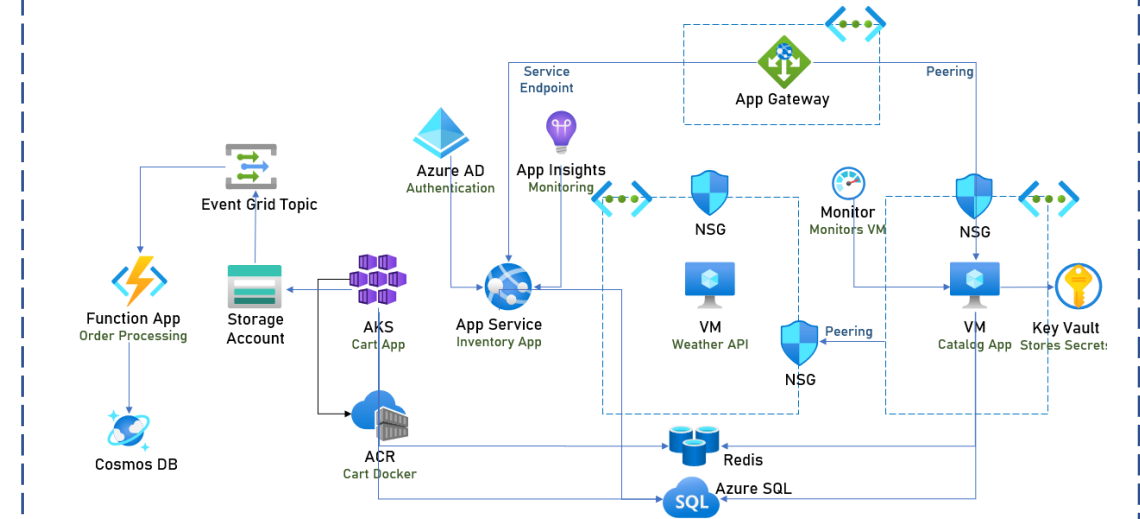
How DR Works?

West Europe Primary

GONE



North Europe Primary



DR Concepts

- Hot / Cold

Hot



- Failover to secondary site happens automatically with no downtime
- No data loss
- Requires duplicate infrastructure
- The most expensive method

Cold



- Failover to secondary site takes some time
- Might be manual
- Some data might be lost
- Less expensive

DR Concepts

- Hot or Cold – how to decide?
- Depends on the system's requirements
- A global ecommerce website, serving million of customers –
probably Hot
- An HR app for the organization – definitely Cold (if at all...)

DR Concepts

- RPO / RTO

RPO



- Recovery Point Objective
- How much data we allow ourselves to lose in case of a disaster
- Usually measured in minutes
- In other words – what's the frequency of data sync to the secondary region
- Example: We have an RPO of 5 minutes

RTO

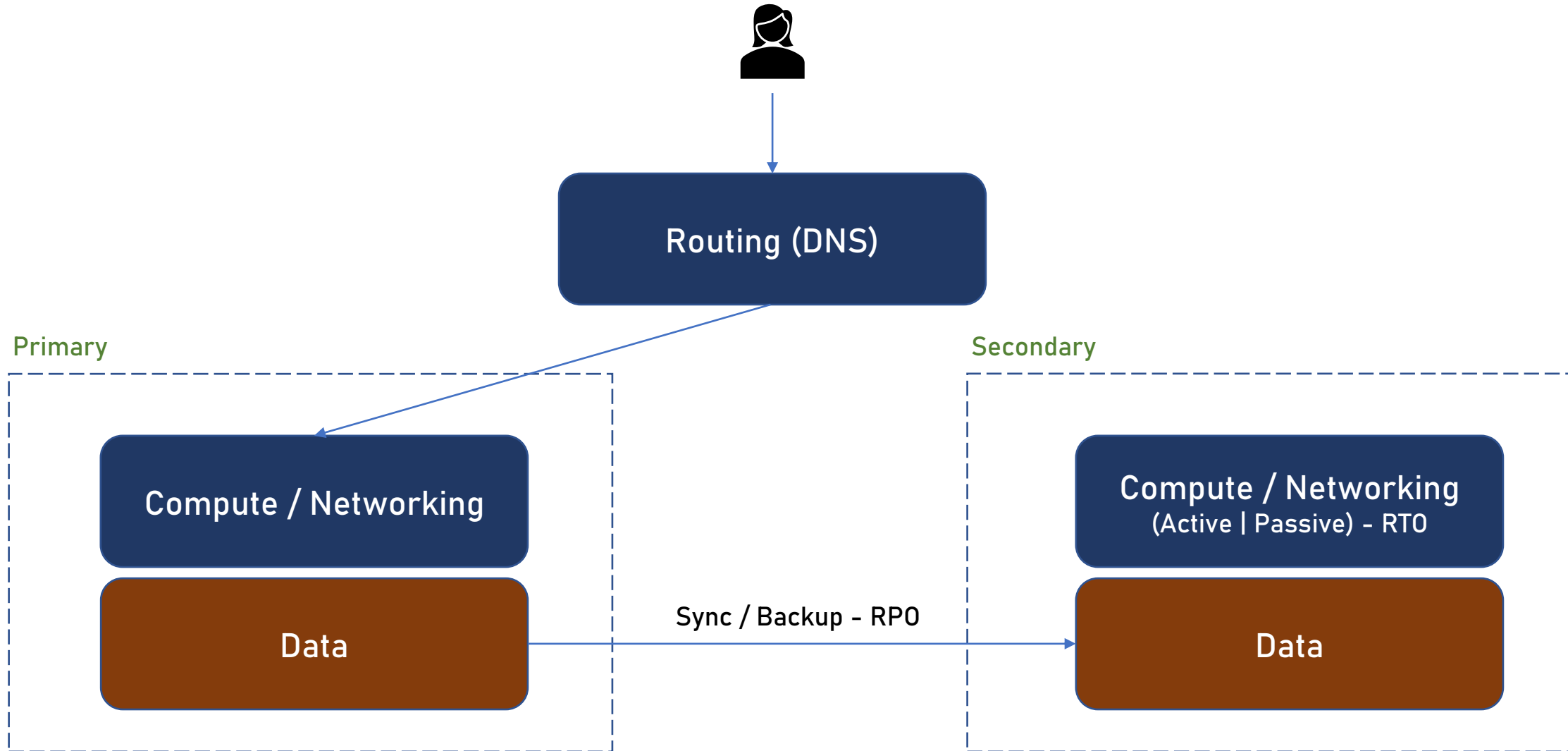


- Recovery Time Objective
- How much downtime we can tolerate in case of a disaster
- Usually measured in minutes
- In other words – how long it should take before the system is up again
- Not necessarily with the most up to date data, depends on the RPO

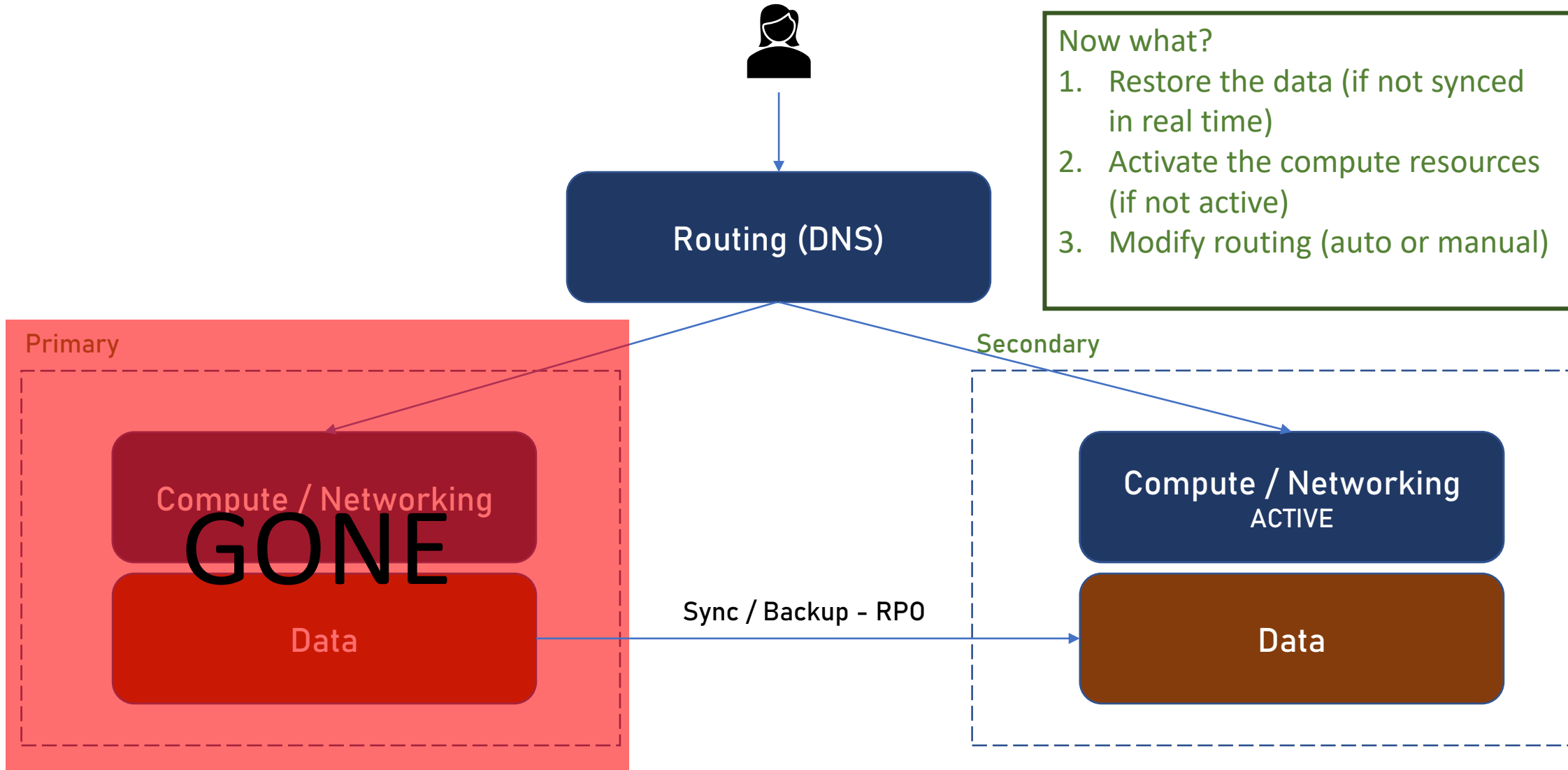
DR Concepts

- RPO and RTO– how to decide?
- Depends on the system's requirements
- A massive reporting system will probably go for low RPO, but can compromise on the RTO
- A global chat will focus on RTO

Basics of DR Implementation



Basics of DR Implementation



DR of Data in Azure

- Main question when designing the DR of data is:

What is the RPO?

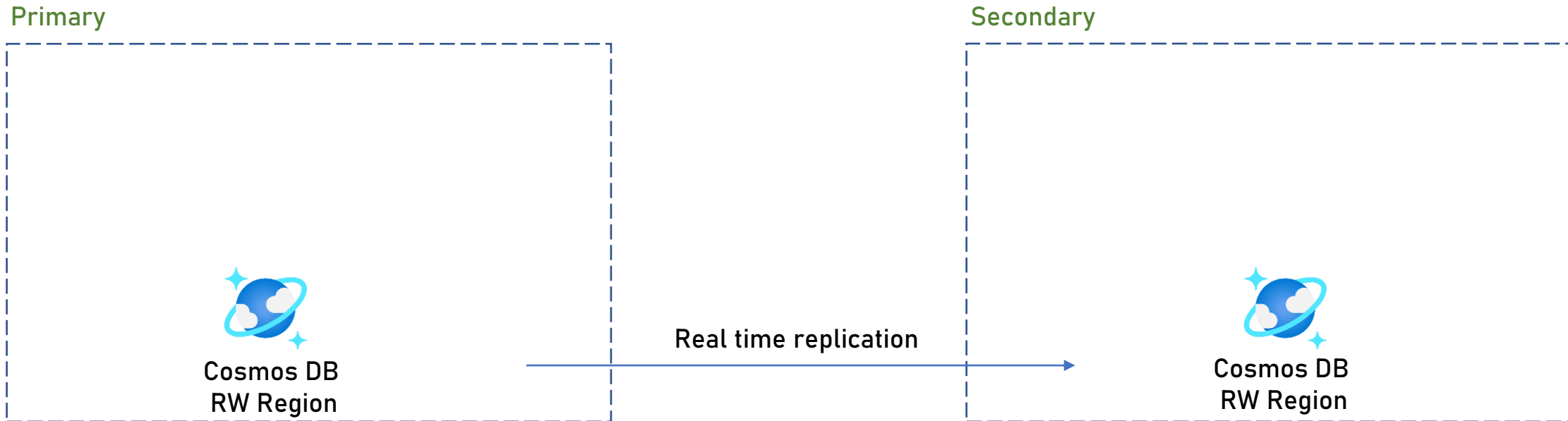
(Or – how much data loss do we tolerate?)

DR of Data in Azure

- If RPO = 0 (no data loss in case of disaster):
 - We need database that always syncs with the secondary region
 - Currently – three such databases in Azure:
 - Azure SQL (with Geo-Replication and Failover Group)
 - Cosmos DB (with multi-region account)
 - Azure Storage (with GRS redundancy)

DR of Data in Azure

RPO = 0

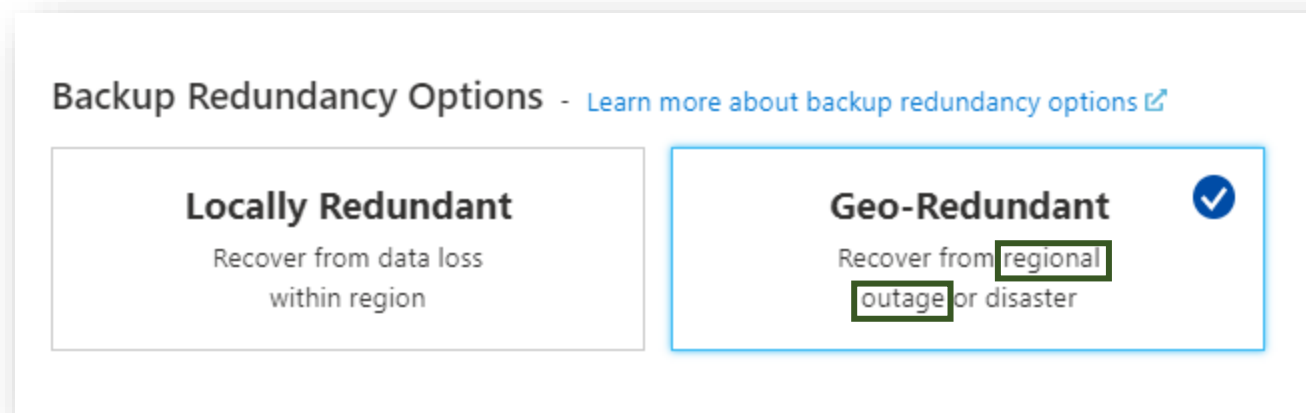


DR of Data in Azure

- If $RPO > 0$ (some data can be lost):
 - Ensure DB's backup frequency is compliant with the RPO
 - The backup storage should be GRS

DR of Data in Azure

- Example – Azure MySQL:



The General purpose storage is the backend storage supporting [General Purpose](#) and [Memory Optimized tier](#) server. For servers with general purpose storage up to 4 TB, full backups occur once every week. Differential backups occur twice a day. Transaction log backups occur every five minutes. The backups on general purpose storage up to 4-TB storage are not snapshot-based and consumes IO bandwidth at the time of backup. For large databases (> 1 TB) on 4-TB storage, we recommend you consider

DR of Data in Azure

Create MySQL server

Microsoft

Server details

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

Server name * ⓘ

secondary-db2 ✓

Data source * ⓘ

None

Backup

This option allows you to restore from the most recent geo-redundant backup of any server in this subscription. The storage capacity of the server will be determined by the backup. Select a backup to continue. [Learn more](#) ⓘ

Backup * ⓘ

Select a backup

Location * ⓘ

(Europe) West Europe

Primary

GONE



Azure MySQL

Backups to...



Storage Account

Replicates to... (GRS)



Storage Account

Restores from...



Azure MySQL

DR of Data in Azure

- Note that:
 - The RPO in the previous example is minimum 5 minutes (the backup frequency)
 - The second example is much cheaper, no secondary active database is needed when primary is active

DR of Compute in Azure

- Main question when designing the DR of compute is:

What is the RTO?

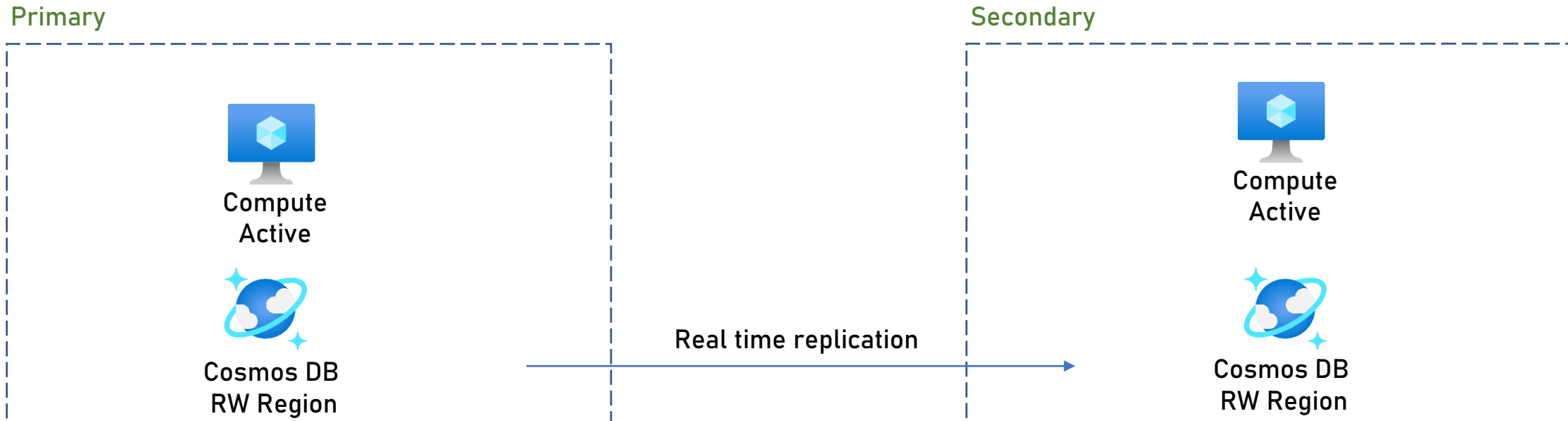
(Or – how much downtime can we tolerate?)

DR of Compute in Azure

- If RTO = 0 (no downtime in case of disaster):
 - Compute in secondary region should always be up and running

DR of Compute in Azure

RT0 = 0

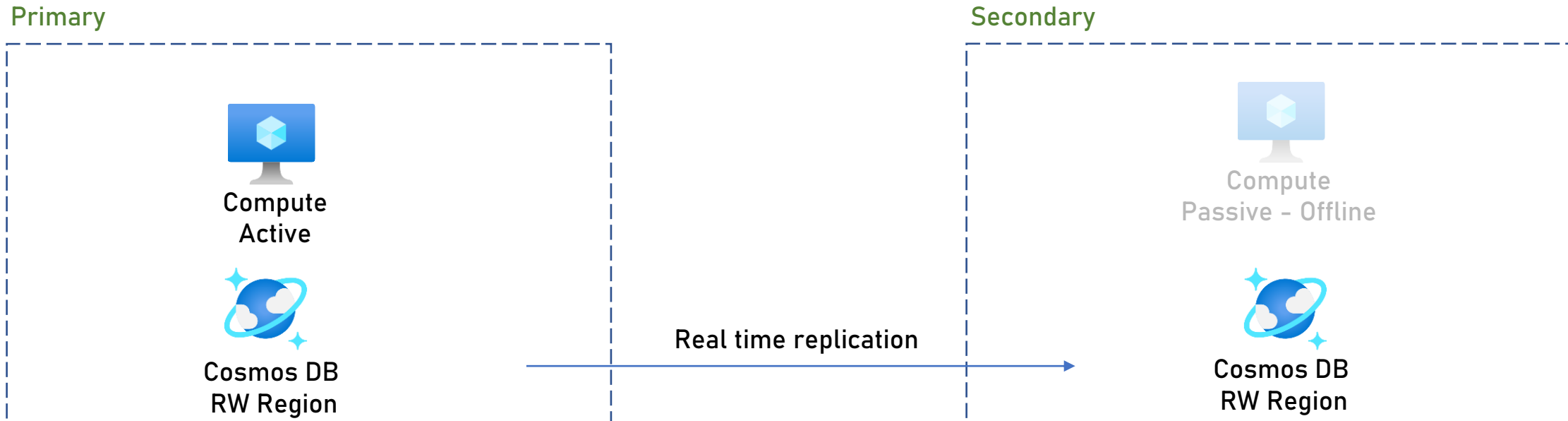


DR of Compute in Azure

- If $RTO > 0$ (some downtime is tolerated):
 - Either:
 - Have non-active (passive) compute on standby in secondary region
 - Create the compute when disaster occurs in secondary region

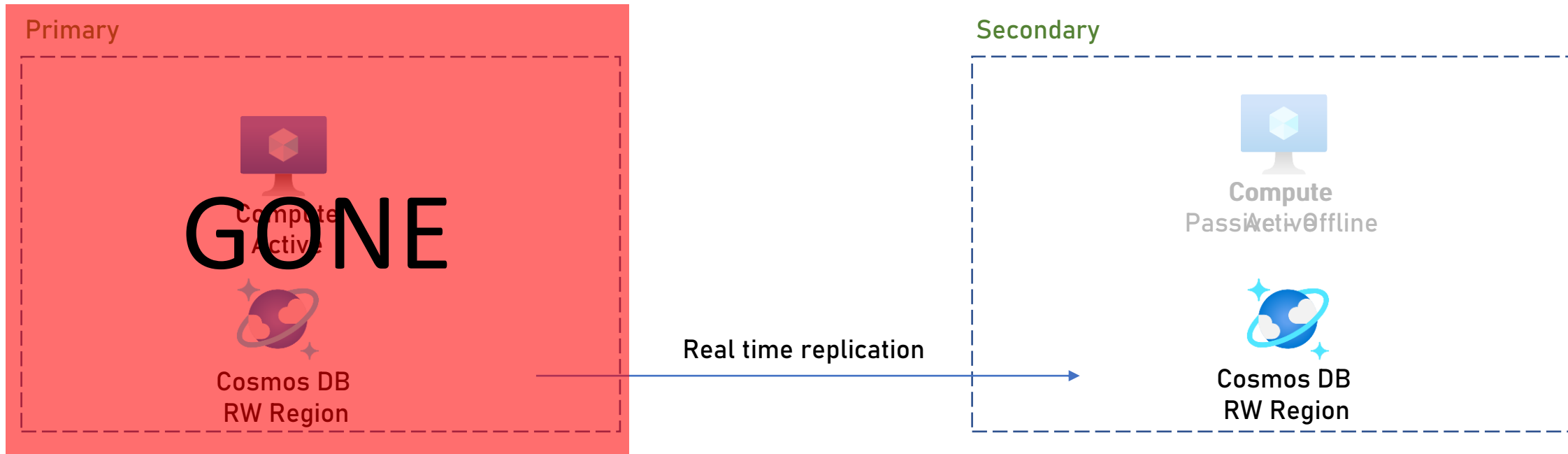
DR of Compute in Azure

RT0 > 0



DR of Compute in Azure

RT0 > 0



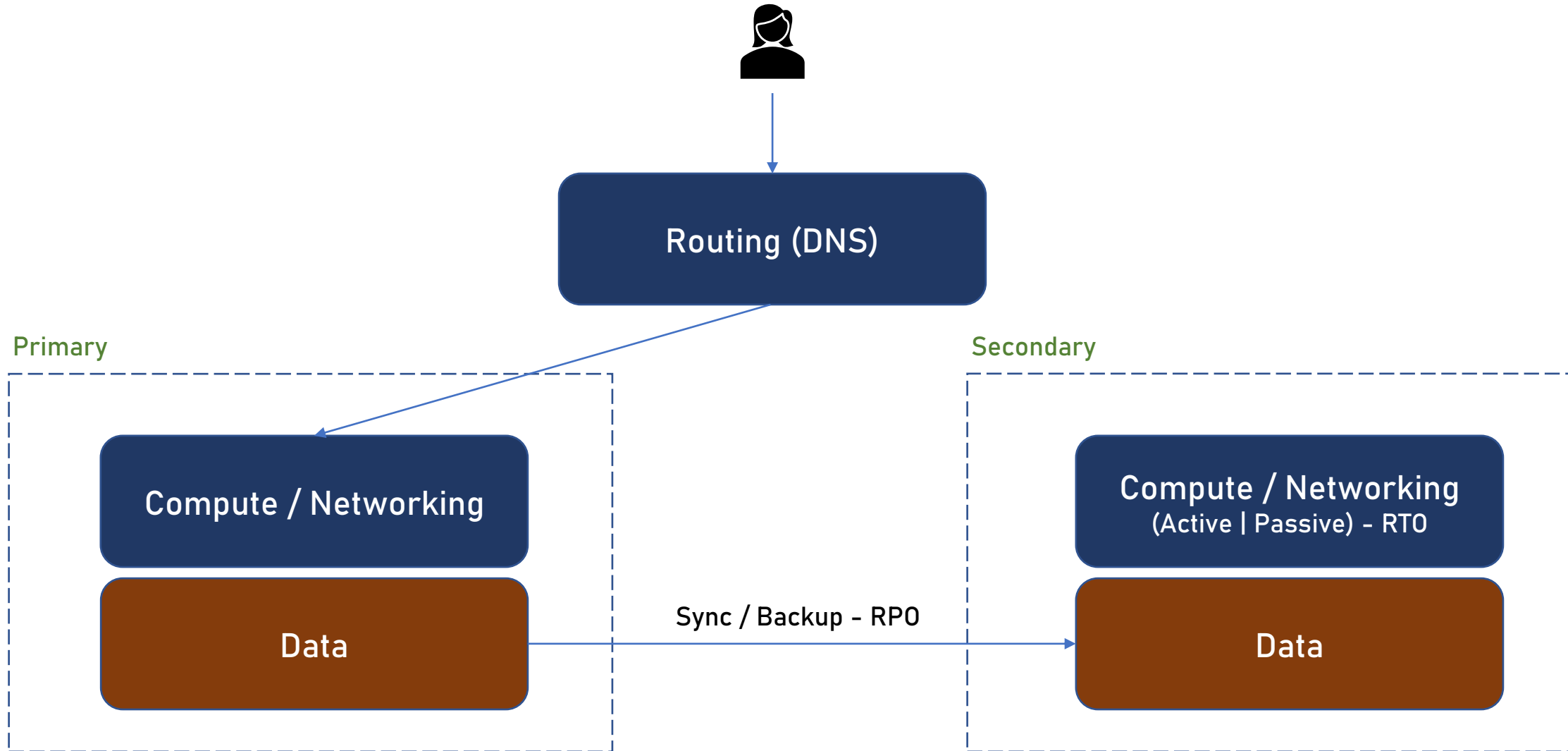
DR of Compute in Azure

- Note that:
 - The second example is much cheaper, no secondary active compute is needed when primary is active

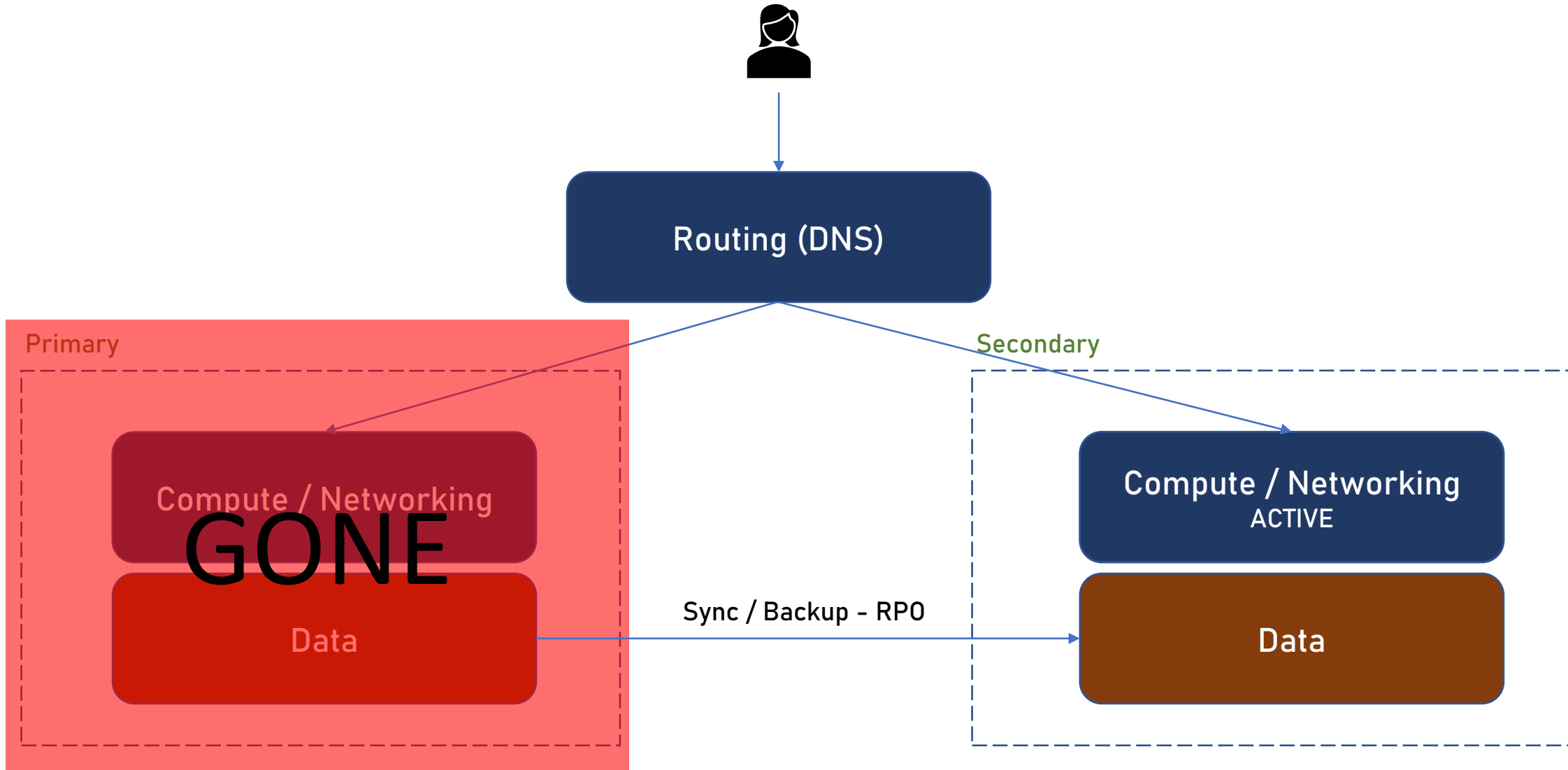
Routing in DR

- During DR users should be routed to the secondary region

Basics of DR Implementation



Basics of DR Implementation



Routing in DR

- Major three methods:
 - Inform the users about the new address of the app (in the secondary region)
 - Manually change DNS record to point to the secondary region
 - Use automatic routing

Routing in DR

- Azure has two automatic routing services

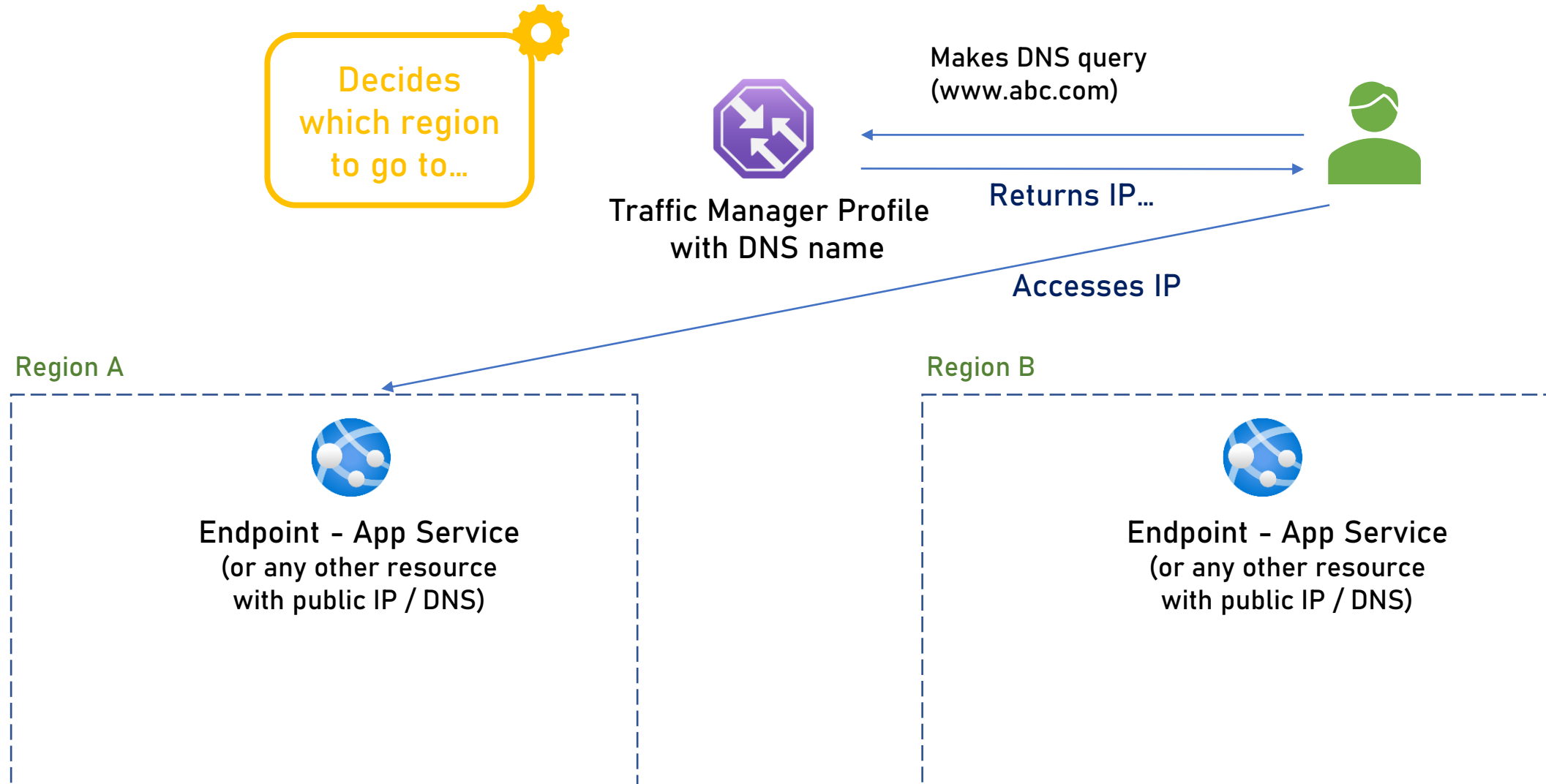
Traffic Manager

Front Door

Azure Traffic Manager

- DNS-based traffic load balancer
- Enables traffic distribution across global Azure regions
- Provides high availability and responsiveness

How Does Azure Traffic Manager Work?



Routing Algorithms

Priority

→ Always use primary service, when it's down – using backup endpoints (That's the DR...)

Weighted

→ Distribute traffic across endpoints according to weights defined by you

Performance

→ Use the closest region to improve latency

Geographic

→ Direct traffic to specific geography based on the location of the DNS query

Multivalue

→ Returns list of healthy endpoints so the client can choose what to do with them

Subnet

→ Map source IP to endpoint

Traffic Manager Pricing

Traffic Manager

REGION:
West Europe

DNS Queries

10

Million/month

= \$5.40

Health Checks

Azure

2

Endpoints

×

\$0.36

Per month

= \$0.72

Fast Interval Health Checks Add-on (Azure)

0

Endpoints

×

\$1.00

Per month

= \$0.00

i

Fast endpoint health checks need to be purchased as an add-on to basic endpoint health checks.

External

0

Endpoints

×

\$0.54

Per month

= \$0.00

Fast Interval Health Checks Add-on (External)

0

Endpoints

×

\$2.00

Per month

= \$0.00

i

Fast endpoint health checks need to be purchased as an add-on to basic endpoint health checks.

Real User Measurements ⓘ

0

Million measurements

×

\$2.00

Per month

= \$0.00

Azure Front Door

- Global entry point for web apps
- Works on Layer 7 (HTTP/HTTPS)
- Multiple routing methods
- Similar to Application Gateway but in global scale

Azure Front Door Features

- URL-path based routing
- Session affinity
- SSL Offloading
- Web Application Firewall (WAF) integration
- URL Rewrites
- HTTP/2 support

Azure Front Door Pricing

Azure Front Door

REGION:
West Europe

Outbound Data Transfer
Zone 1: North America, Europe and Africa
100 GB = \$17.00

Inbound Data Transfer
20 GB = \$0.20

Routing rules
1 Rules 730 Hours = \$21.90

Web Application Firewall (WAF)
Policy
0 Policies \$5.00 Per policy per month = \$0.00

Custom Rules
0 Rules \$1.00 Per rule per month = \$0.00
2 Requests processed (in millions) \$0.60 Per million requests = \$1.20

Managed Ruleset
0 Default Rulesets \$20.00 Per ruleset per month = \$0.00
1 Requests processed (in millions) \$1.00 Per million requests = \$1.00

Upfront cost \$0.00
Monthly cost \$41.30

Traffic Manager vs Front Door

- Which one to choose?
- Generally – if you need HTTP-related capabilities go with Front Door
- Examples:
 - URL-path based routing
 - SSL Offloading
 - Web Application Firewall
- Otherwise – go with Traffic Manager, usually cheaper

ReadIt!

Cloud Architecture

