ANALYSIS OF THE ARCHITECTURE OF A CLOUD APPLICATION

Frunza Teodor-Octavian

VOLATILITY OF CLOUD COMPUTING

01 Cloud Computing Types

02 AWS Services

03 Cloud Architecture Study Case

04 Questions

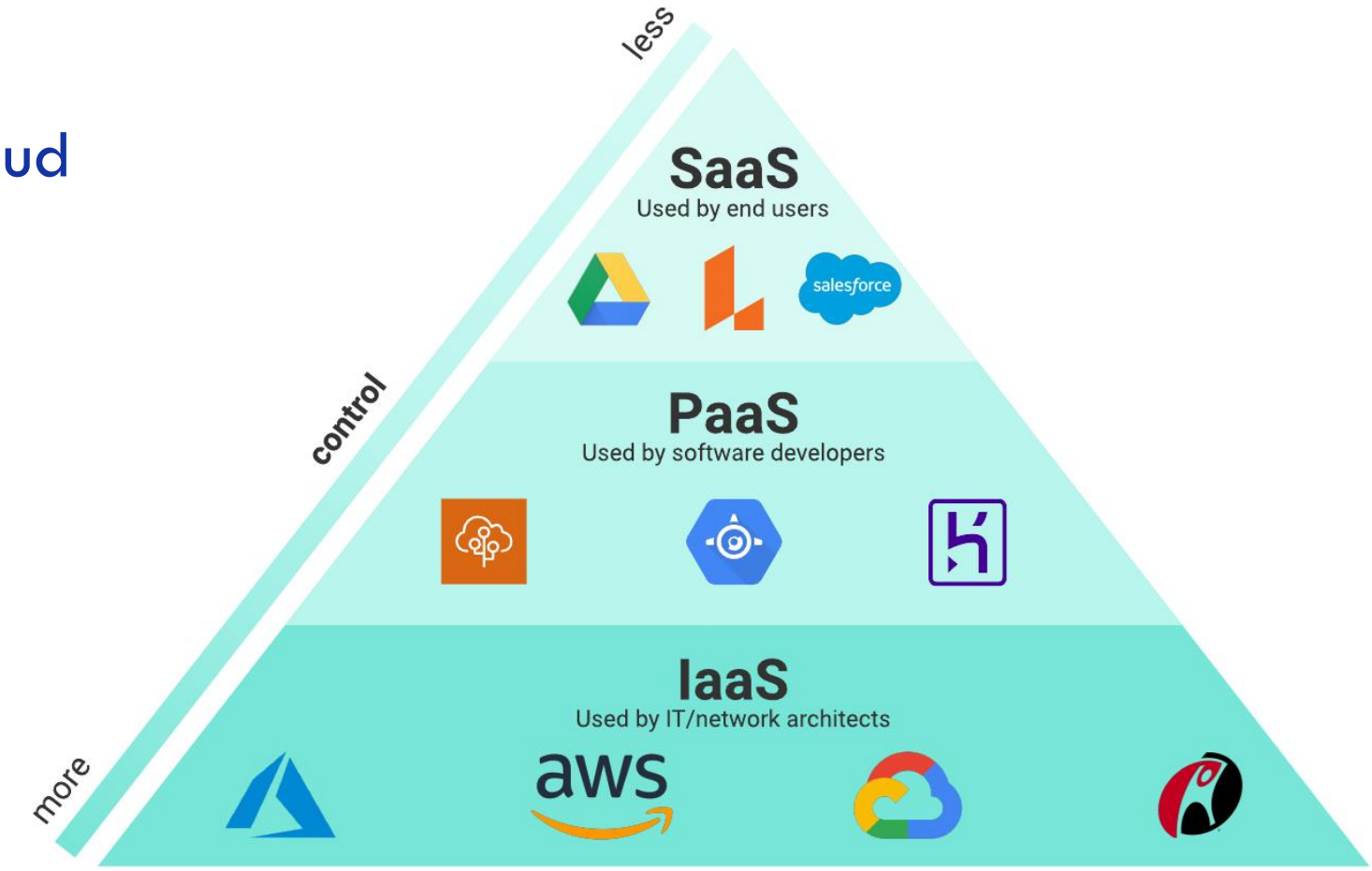
01

CLOUD COMPUTING TYPES

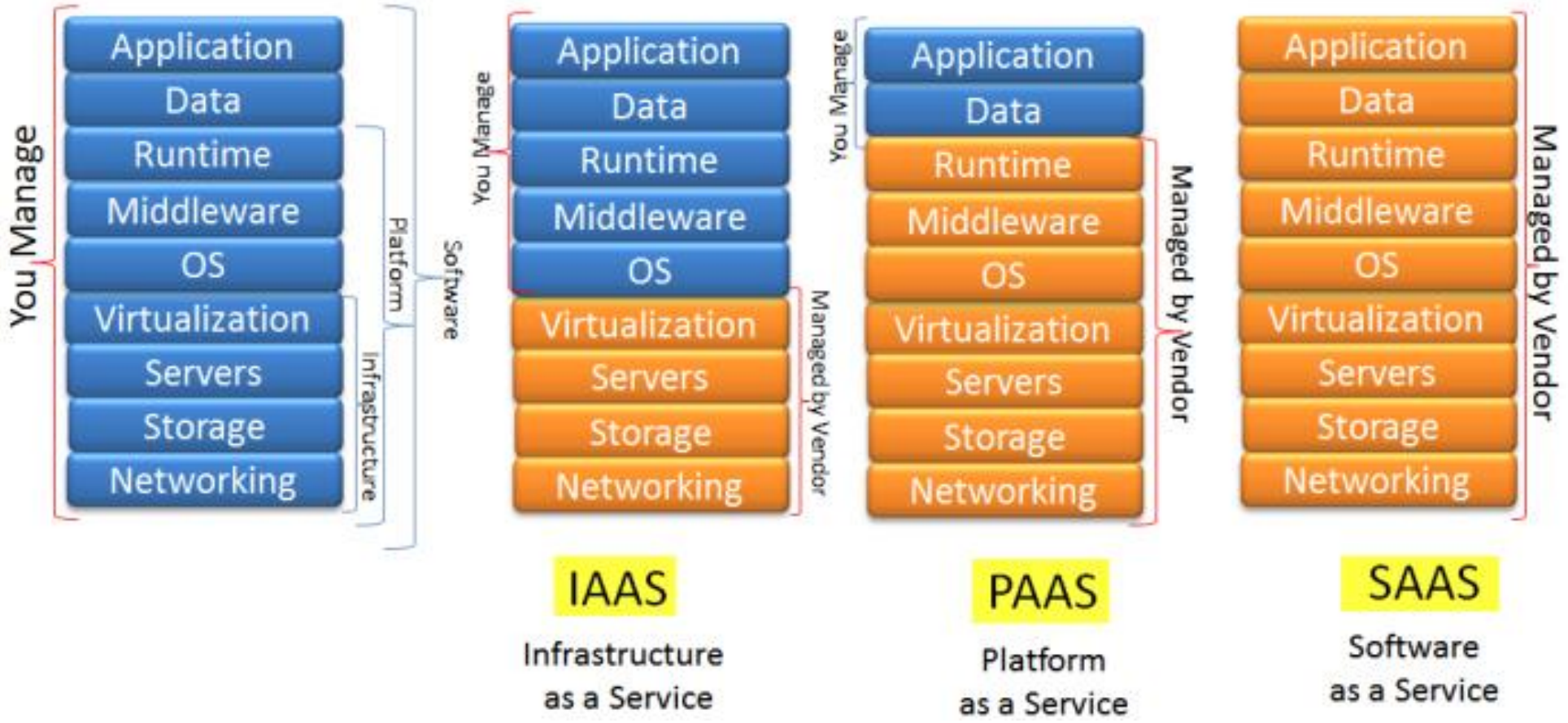
CLOUD COMPUTING TYPES

There are 3 types of cloud computing:

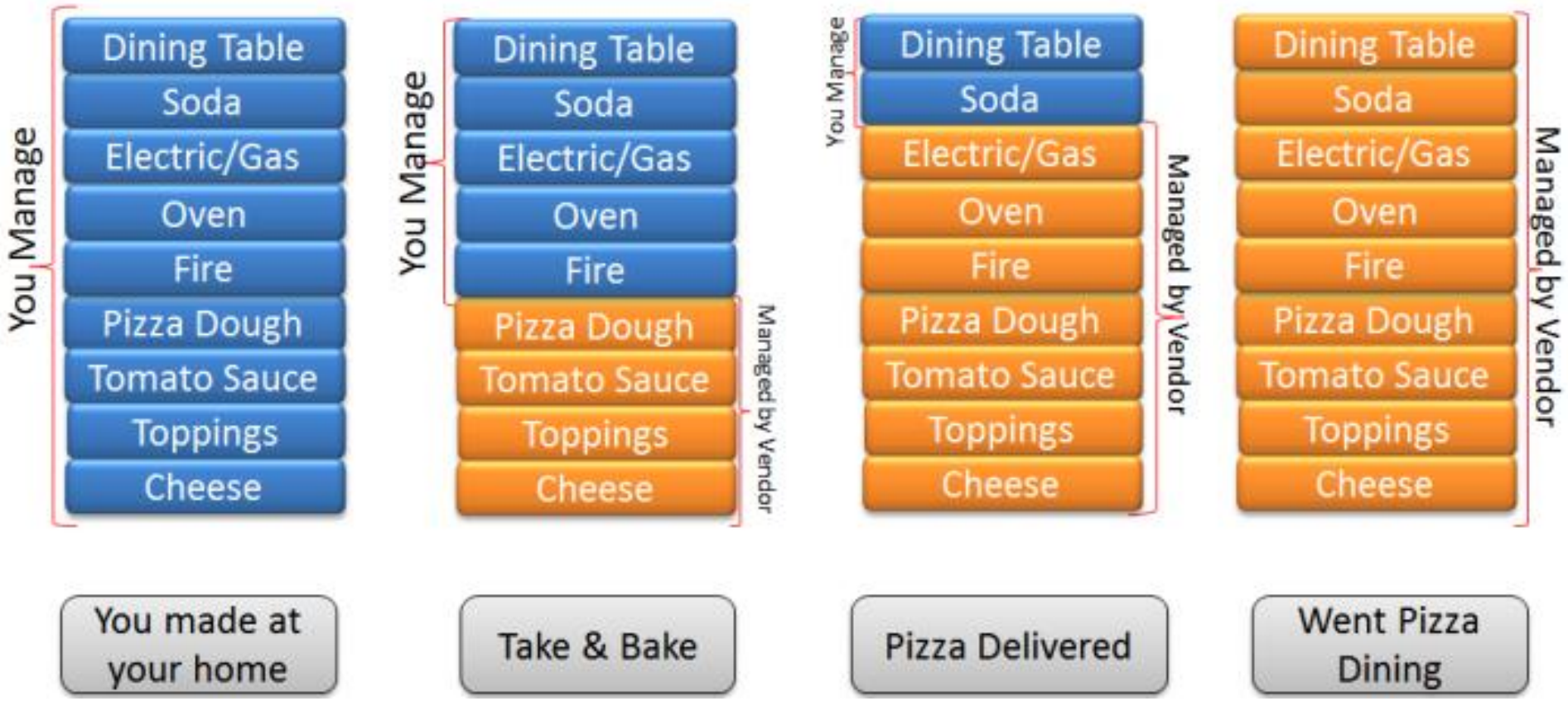
Infrastructure as a service (IaaS)
Platform as a service (PaaS)
Software as a service (SaaS)



CLOUD COMPUTING TYPES



CLOUD LIKE PIZZA



PLATFORM AS A SERVICE

Allows users to connect and use cloud-based services over secure Internet connections;

Complete software solution based on services integration;

Pay-as-you-go;

Manage only the applications and services you develop and cloud provider manages everything else;

Avoid expenses and complexity of managing licenses and hardware resources.

FUNCTION AS A SERVICE (FAAS)

Concept that aims to offer developers to create software function in a cloud environment without much overhead

Code is executed in stateless compute instances that are managed by cloud providers

Provides an event-driven computing architecture where functions are triggered by certain events

Pay-as-you-go

Primarily uses Lambda functions which are stateless and lightweight.

PROS & CONS

PROS	CONS
Cost-effective	Response latency (cold start)
Less overhead	Limitations
Effortless efficiency	Unmanaged Security
Increased scale	Debugging
More flexibility	Loss of fine-grained control
Faster time to market	Potentially expensive
Boosts productivity	

02

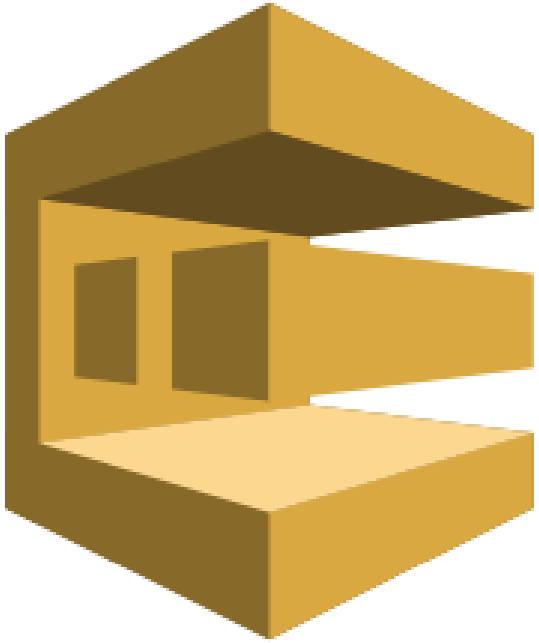
AWS SERVICES

AMAZON SQS

Amazon Simple Queue Service (SQS) is a fully managed message queueing service.

Allows the user to decouple and scale distributed systems , microservices and serverless applications.

SQS is used to send, store and receive messages between software components.



Amazon SQS

SQS FEATURES

There are 3 types of queues:

Standard Queue

FIFO Queue

Dead Letter Queue

SQS uses a deduplication ID in order to remove possible duplicated messages.

Default message retention period: 4 days. Can be customized.

Message delivery retries using message visibility (default 30 seconds)

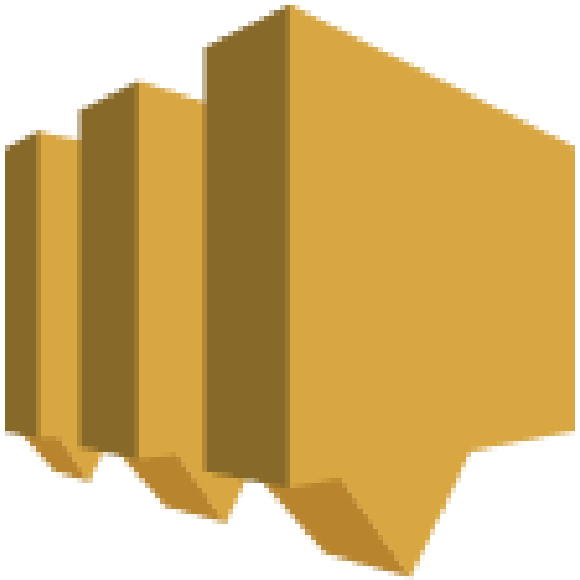
Used for communication at any time and level, without losing messages or requiring other services to be available.

AMAZON SNS

Amazon Simple Notification Service is a fully managed messaging service

Can be used to communicate between 2 services or between a service and a user

Provides topics for high-throughput, push-based, many-to-many messaging between microservices and event-driven serverless applications



SNS FEATURES

Push-based message broker, many-to-many

Ensure accuracy with message ordering and deduplication

Increase security with message encryption and privacy

Send notifications to users via SMS mobile push and email

Offers message filtering and batching

AMAZON ELASTICACHE

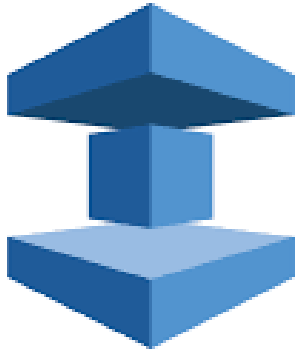
Distributed in-memory data store / cache environment

High-performance, highly scalable and available caching solution

Fully managed

Fault tolerant – has replication

Works with Redis and Memcached



Amazon
Elasticache

DYNAMODB

DynamoDB is a non-relation database also referred to as a document store

Can store any amount of data and supports any amount of traffic by automatically distributing the data and the traffic over multiple servers

Fully managed

Pay-as-you-go / dedicated

Amazon DynamoDB

DYNAMODB FEATURES

Highly scalable and flexible

High fault tolerance through replication

Fundamentally schema-less, but can have a schema

Very powerful querying capabilities

Has caching features

Uses read / write capacity units and storage for invoicing

Generous free tier with 25Gb of storage and 25 read + 25 write units / month

TERMINOLOGY

Dynamo DB is a NoSQL database and does not use the terms like tables and rows in the traditional way

Table – The main table in which data will be persisted (in most cases one should be enough)

Partition – A logical partition that hold documents that fall in a certain category

Partition key (HASH key) – An attribute that represents a group that will contain specific documents (can be correlated with SQL table)

Sort key (RANGE key) – An attribute that represent a sorting field (can be correlated with: SQL Id field)

TERMINOLOGY

Local Secondary Index

Acts as Composed Primary Key where we have: Partition Key + another field that is not already a sort key
A logical partition can have one primary Composed Primary Key (Partition key + sort key) and multiple Local Secondary Indices
Used in cases where we require multiple indexing with O(1) complexity
Consumes Read / Write units from the table's pool

Global Secondary Index

Similar to Local Secondary Indices
The Partition Key and Sort Key can differ from the main logical partition keys
Used to do optimised scan operations
Consumes dedicated Read / Write units that are added to the initial table pool

SHARDING

Sharding is a method for distributing a single logical dataset across multiple databases.

The databases can be stored on multiple machines.

Form of horizontal scaling.

Two types of sharding:

Horizontal sharding

- Same schema, unique rows.
- Increases processing speed and traffic load capacity

Vertical sharding

- Schema is a subset of the original schema
- Useful when queries return only a subset of the data

SHARDING - HORIZONTAL

Student ID	Name	Age	Major	Hometown
1	Amy	21	Economics	Austin
2	Jack	20	History	San Francisco
3	Matthew	22	Political Science	New York City
4	Priya	19	Biology	Gary
5	Ahmed	19	Philosophy	Boston

Student ID	Name	Age	Major	Hometown
1	Amy	21	Economics	Austin
2	Jack	20	History	San Francisco

Student ID	Name	Age	Major	Hometown
3	Matthew	22	Political Science	New York City
4	Priya	19	Biology	Gary
5	Ahmed	19	Philosophy	Boston

SHARDING - VERTICAL

Student ID	Name	Age	Major	Hometown
1	Amy	21	Economics	Austin
2	Jack	20	History	San Francisco
3	Matthew	22	Political Science	New York City
4	Priya	19	Biology	Gary
5	Ahmed	19	Philosophy	Boston

Student ID	Name	Age
1	Amy	21
2	Jack	20

Student ID	Major
1	Economics
2	History

Student ID	Hometown
1	Austin
2	San Francisco

SHARDING

Types of sharding:

Key-based
Range-based

Key-based (Hash-based)

Generates a hash based on input (from the same column)
The hash is the shard ID

Advantages	Disadvantages
Suitable for distributing data evenly to prevent hotspots	Difficult to dynamically add or remove additional servers to the database
Data is distributed algorithmically.	During migration, servers cannot write any new data, which may lead to application downtime

SHARDING

Range-based

Sharding data using a range of a given value
The hash is the shard ID
The range is based on a field

Advantages	Disadvantages
Suitable implementation and algorithm	May create database hotspots, since data could be unevenly distributed
Shards have identical schema	Inappropriate shard key selection may lead to unbalanced shards

SHARDING – PROS & CONS

PROS	CONS
Increased read/write throughput inside a shard	Query overhead - sharded database must have a separate machine or service which is capable of routing queries to the appropriate shard
Increased storage capacity – near-infinite scalability	Complexity of administration
High availability	Increased infrastructure costs

PARTITIONS

Partitions are generated based on the PARTITION KEY.

The logical partitions have SSD backing and automatic replication across different availability zones (defaults to 3 instances)

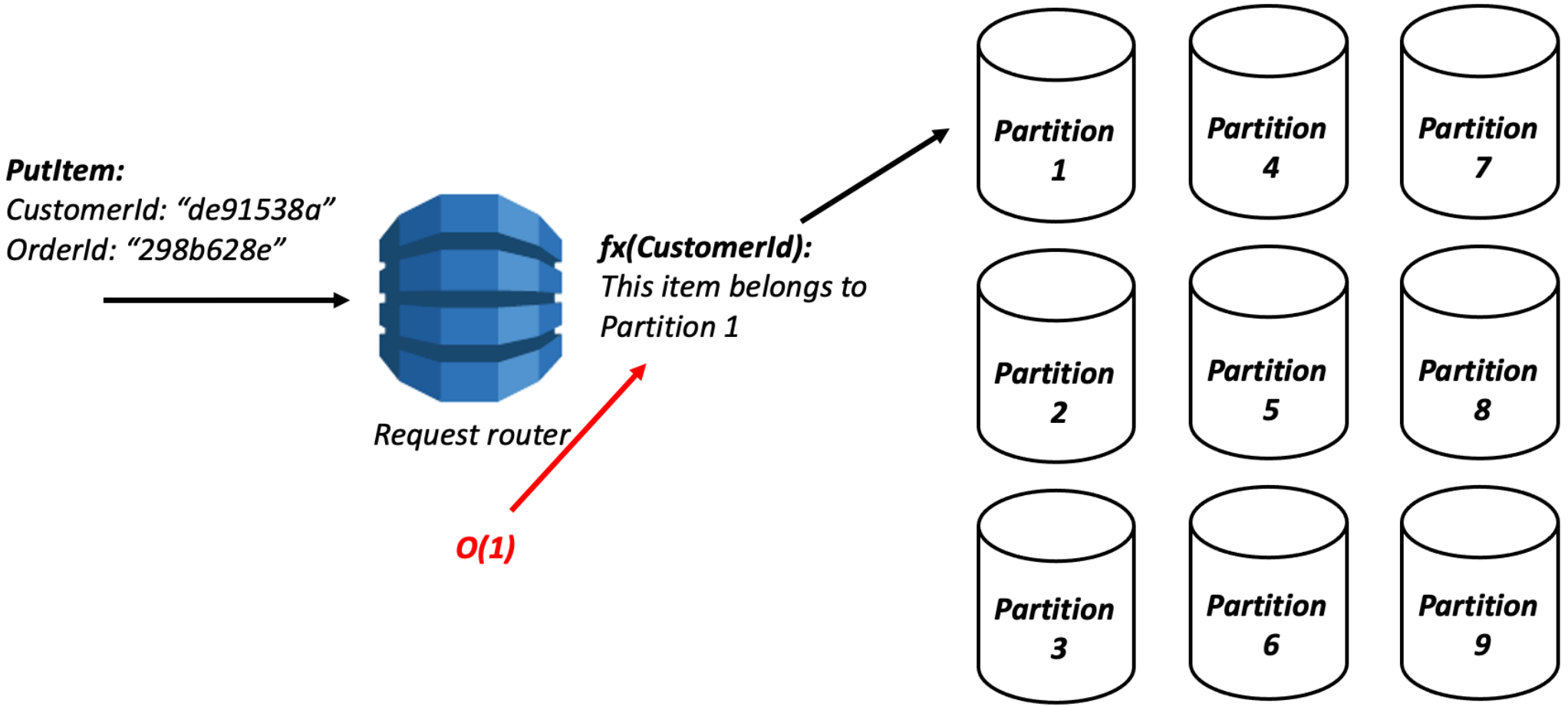
All the items that will have the same partition key will reside in the same logical partition

A document must always have a partition key. It can also have a sort key, but it is optional. If a sort key is present, it should be unique for every item in a partition (acts as an id)

A partition key and a sort key form a Composite Primary Key

Partition keys and sort keys will be used for indexing

HOW DOES IT WORK?



DATA REPLICATION

Replication refers to the process of copying data from one database server to another

Increases availability and fault-tolerance

Ensures consistency (eventual consistency and strong consistency)

Can be:

Synchronous – when writing new data, it propagates the changes immediately to the replicas

Asynchronous – the new data is first written to the main server and then propagates to the replicas

OPERATIONS

Basic **CRUD** operations.

Query – within a logical partition

Scan – withing the table, over all partitions

Write and update are merged in an **upsert** mechanism.

PROS & CONS

Pros	Cons
Scalable	Weak querying model
Cost effective	Lack of server-side scripts
Data replication	Table joins
Serverless	Hard to predict costs
Schema less DB	
Secure	

AWS LAMBDA

Small piece of code runs in an event-based, distributed system and can be triggered by various events

Integration with various AWS services

Part of PaaS and FaaS – no infrastructure to manage

Pay-as-you-go / dedicated

Stateless

PROPERTIES

Lambda functions run in a container that contains the runtime, libraries etc.

Types of functions:

Pay-as-you-go – has cold start, lower costs

Provisioned – no cold start, higher costs

Each function runs in an isolated environment

All dependencies, that are not already in the AWS directory, get installed in the initialization phase

Can share a common configuration (environment variables)

LIMITATIONS

Maximum run time: 15 minutes

Maximum concurrent executions: 1000

Storage for functions zips and layers 75GB

Memory allocation: 128 MB - 10 GB

Maximum lambda layers: 5

Maximum payload:

Synchronous: 6 MB

Asynchronous: 256 KB

AWS STEP FUNCTIONS

Serverless orchestrator service which provides a functionality for creating and managing multi-steps application workflows in the cloud

Designed to help organizations integrate multiple AWS services while allowing them to manage each microservice component independently

Integrates with Lambda and other AWS services (200+)

Built using Amazon States Language (ASL) – proprietary to Amazon

TERMINOLOGY

State machine – Type of computational device that is able to hold a state and update it based on various input. AWS Step functions are built on these state machines which are often referred as workflows.

State – Represents a step inside a workflow. Types of states:

Task state – performs work on a state machine
Choice state – choose between different paths (if statement)
Fail / Succeed state – stops the workflow in a failed / successful state
Pass state – passthrough state / pass output or some fixed data to another state
Wait state – pauses the state for some time (can be un to 1 year)
Parallel state – executes tasks in a parallel manner
Map state – executes tasks in an iterative manner

Transition – Action that represents a state change

TASK STATE

Used to complete a unit of work

Types of tasks:

Activity tasks

- Connect code / service that is running elsewhere (activity worker)
- The step function, completes the work and returns the result in an asynchronous manner
- Common in workflows that requires human interaction

Service tasks

- Connect steps to specific AWS services
- Step function invokes other services (Over 200 AWS service), waits for it to finish and resumes execution
- Automatizations

WORKFLOWS TYPES

Express workflow

High-volume, event-processing workloads such as IoT data ingestion, ETL, machine learning, microservices orchestration

By default runs by at-least-once model

Billed by number of executions, duration of executions and memory consumption

Standard workflow

Ideal for long-running, durable, auditable workflows – can run up to one year

By default runs only once (can be retried if a Retry behaviour is added)

Billed by the number of state transitions

Has free tier: 4000 state transitions

BENEFITS

Orchestrate complex workflows over multiple services with minimal overhead

Decouples the workflow logic from the business logic

Serverless – no infrastructure to manage

Reduced need for integration code

Has retry mechanism with exponential back-off

Easy to debug and understand because of the available intuitive UI

CHALLENGES

Proprietary language requirement (ASL)

Application code is harder to understand

Vendor lock-in

Monitoring limitations – high visibility inside the state machine, low visibility outside of it (input origin, output source)

LIMITATIONS

Maximum transition message size: 256Kb

Maximum request size: 1 MB

Maximum run time: 1 year

Maximum history log retention period: 90 days

Cannot resume state machine from any state

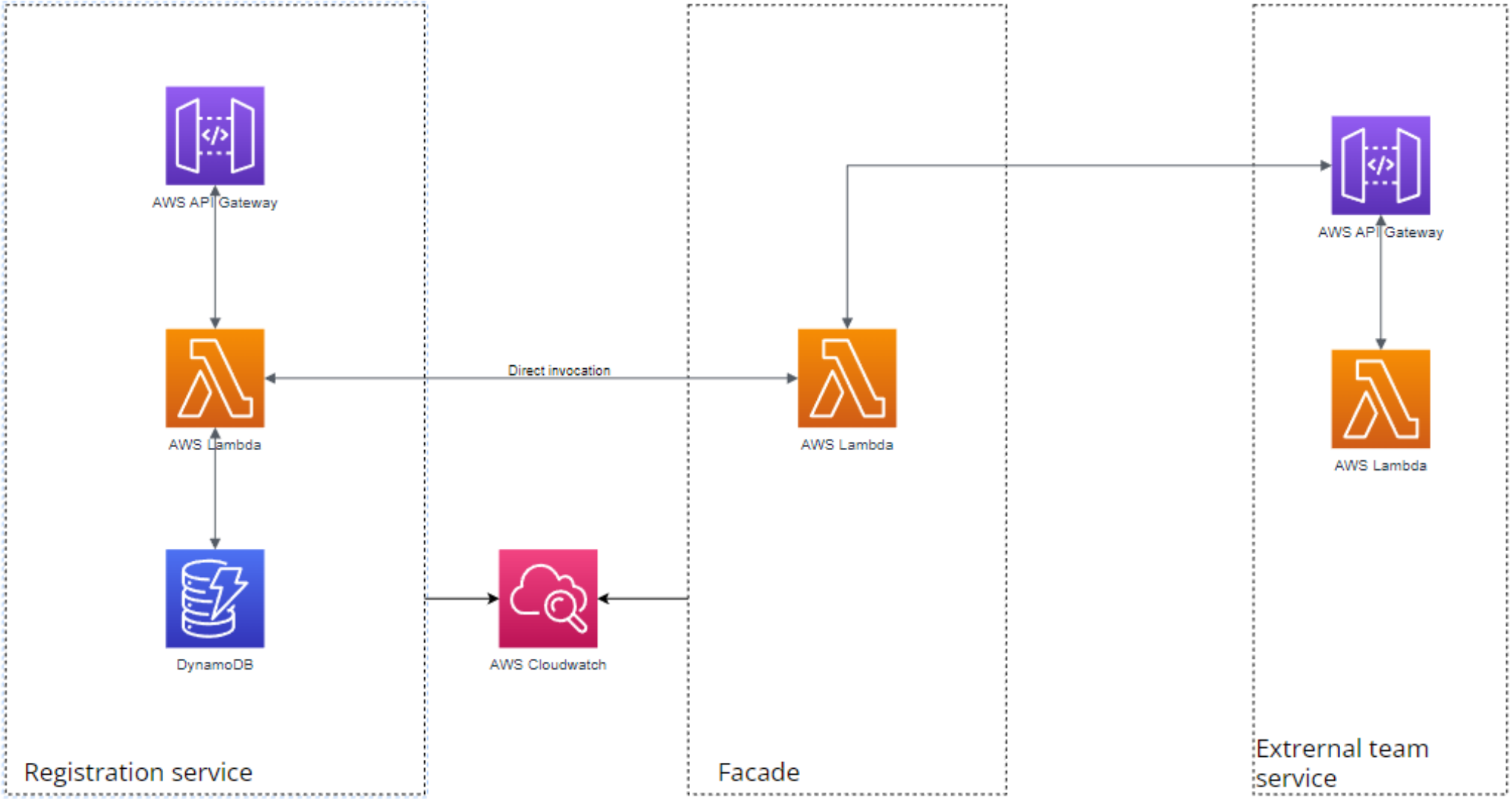
Maximum items execution history per workflow: 25.000

DEMO TIME

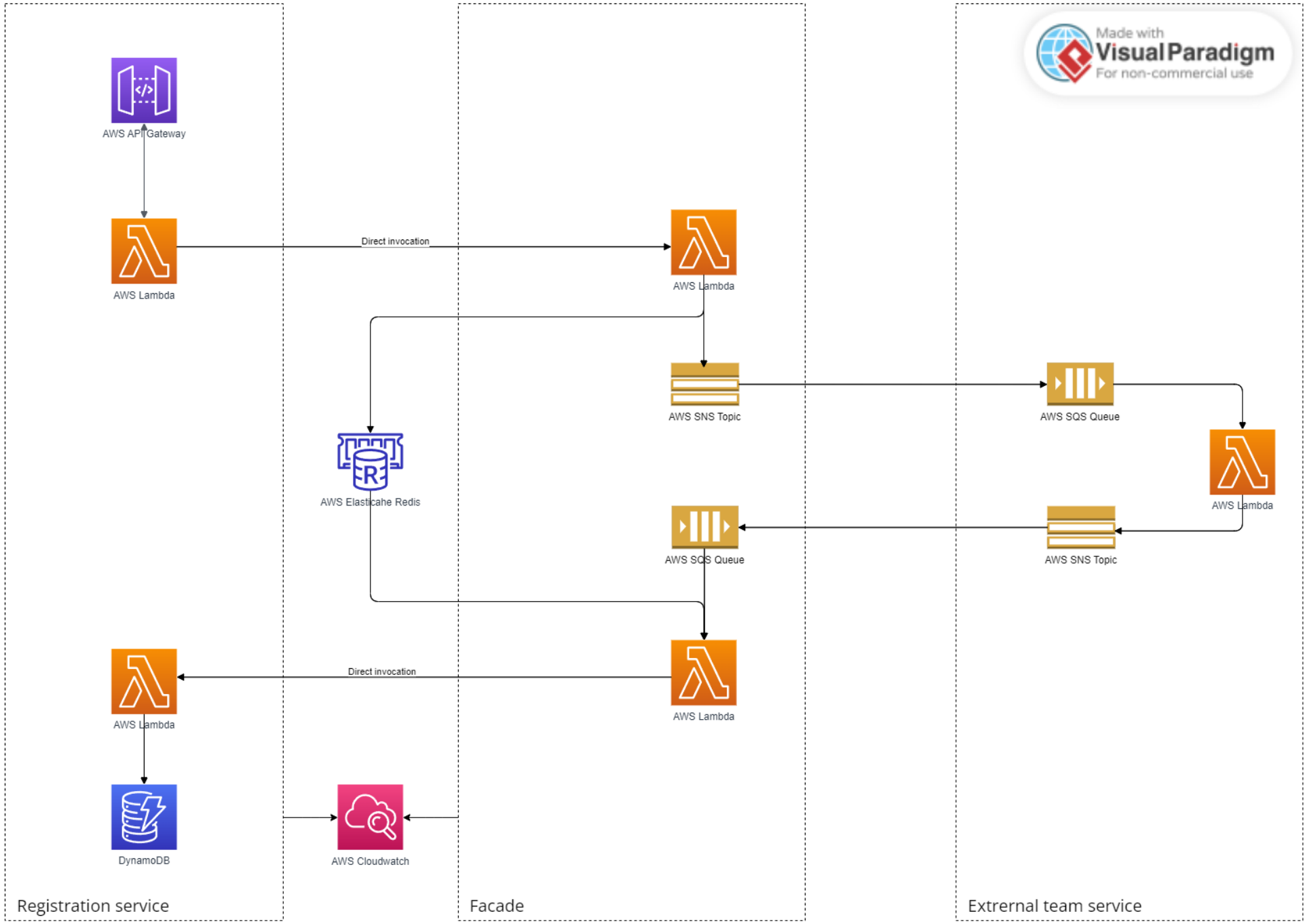
03

CLOUD ARCHITECTURE STUDY CASE

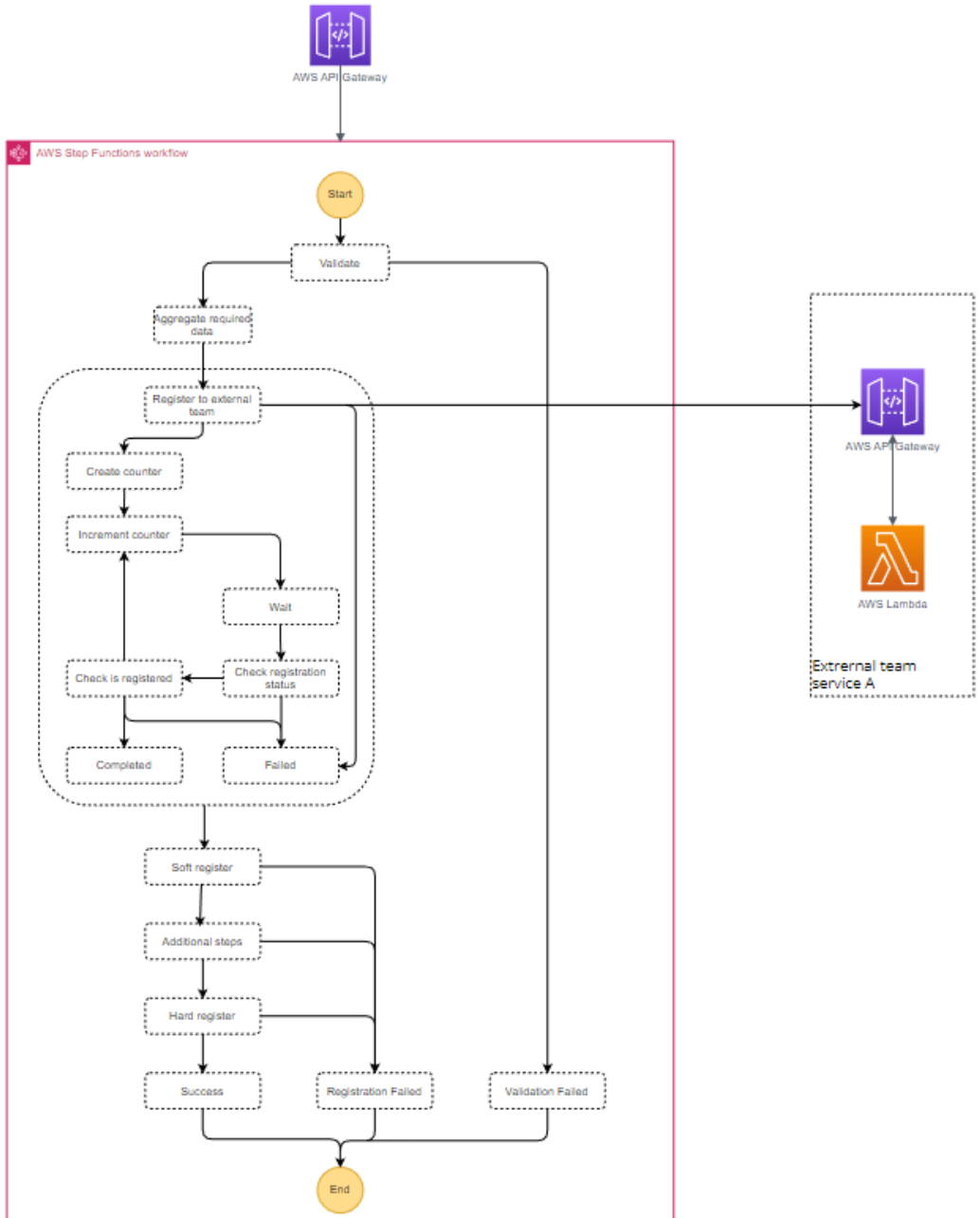
First iteration of the register BE



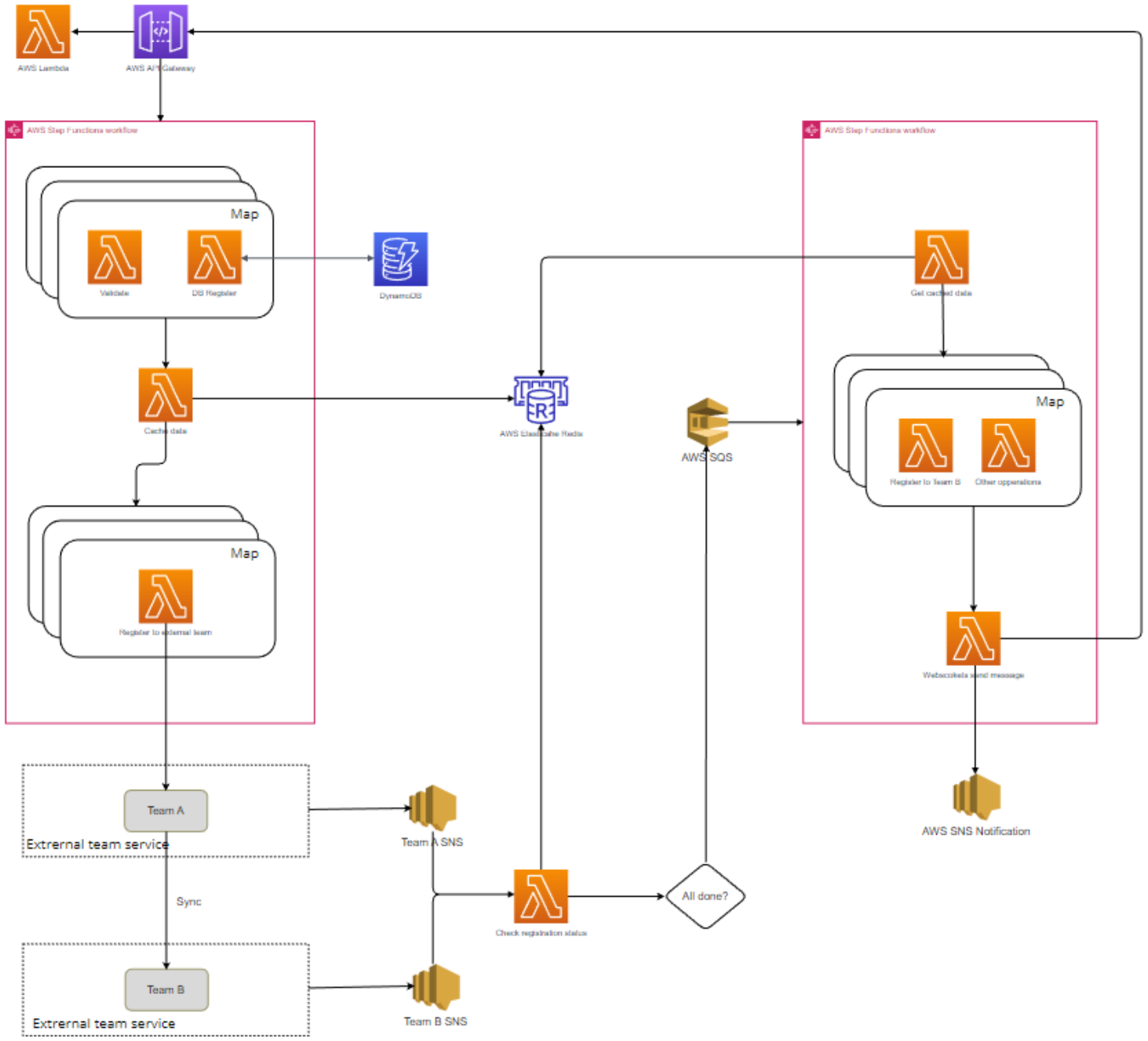
Second iteration of the register BE



Third iteration of the register BE



Fourth iteration of the register BE



04

QUESTIONS

THANK YOU!

levi nine

Technology Services