Case Study #2

Memi Lavi
www.memilavi.com

Muscar.auto

The Real Autonomous Car

Muscar.auto

- Manufactures autonomous systems for vehicles
- Has >10,000 vehicles on the roads right now
- Expects more than 200,000 vehicles by end of year
- Needs to reliably receive telemetry from cars and display data about them

Muscar.auto

Requirements

Functional

What the system should do

1. Web Based
2. Receive telemetry from cars (location, speed, breakdowns, etc)
3. Store telemetry in a persistent store
4. Display dashboards summarizing the data
5. Perform analysis on the data

Non-Functional

What the system should deal with

NFR – What We Know

1. Data intensive system
2. Not a lot of users
3. A lot of data
4. Performance is important

Muscar.auto

NFR - What We Ask

1. *"How many expected concurrent users?"* — 10
2. *"How many telemetry messages received per second?"* — 7,000
3. *"What is the average size of message?"* — 1KB
4. *"Is the message schema-less?"* — Yes

Muscar.auto

NFR - What We Ask

5. *"Can we tolerate some message loss?"* Sort of…

6. *"What is the desired SLA?"* Highest Possible

Muscar.auto

Data Volume

- 1 Message = 1KB
- 7,000 messages / sec = 7MB / sec

=> ~25GB / hr

=> ~605GB / day

=> ~221TB / year ← That's a lot!

Retention Period

Defines for how long records are kept in the database

What happens to them after the retention period?

- Deleted
- Moved to archive data store

Retention Period

Motivation:

- Keep database from exploding
- Improve query performance

AWS Config adds the ability to specify a data retention policy for your configuration items

Posted On: Aug 7, 2018

Muscar.auto

Retention Period

Muscar needs two types of data:

- Operational, near-real-time (location, speed, etc.)
- Aggregated and ready for analysis (BI – Business Intelligence)

Muscar.auto

Retention Period

Data Type	Used for…	Retention Period
Operational	Monitor real time data from cars. Performance is critical	
Aggregated	Reports, BI. Not real time, can be slower.	

Muscar.auto

Retention Period

Data Type	Used for…	Retention Period
Operational	Monitor real time data from cars. Performance is critical	1 week
Aggregated	Reports, BI. Not real time, can be slower.	Forever

Muscar.auto

Data Volume

- 1 Message = 1KB
- 7,000 messages / sec = 7MB / sec

    => ~25GB / hr

    => ~605GB / day

    => ~221TB / year

Muscar.auto

Data Volume

- 1 Message = 1KB
- 7,000 messages / sec = 7MB / sec

    => ~25GB / hr

    => ~605GB / day

    => ~4TB / week

Muscar.auto

Requirements

Functional

What the system should do

1. Web Based
2. Receive telemetry from cars (location, speed, breakdowns, etc)
3. Store telemetry in a persistent store
4. Display dashboards summarizing the data
5. Perform analysis on the data

Non-Functional

What the system should deal with

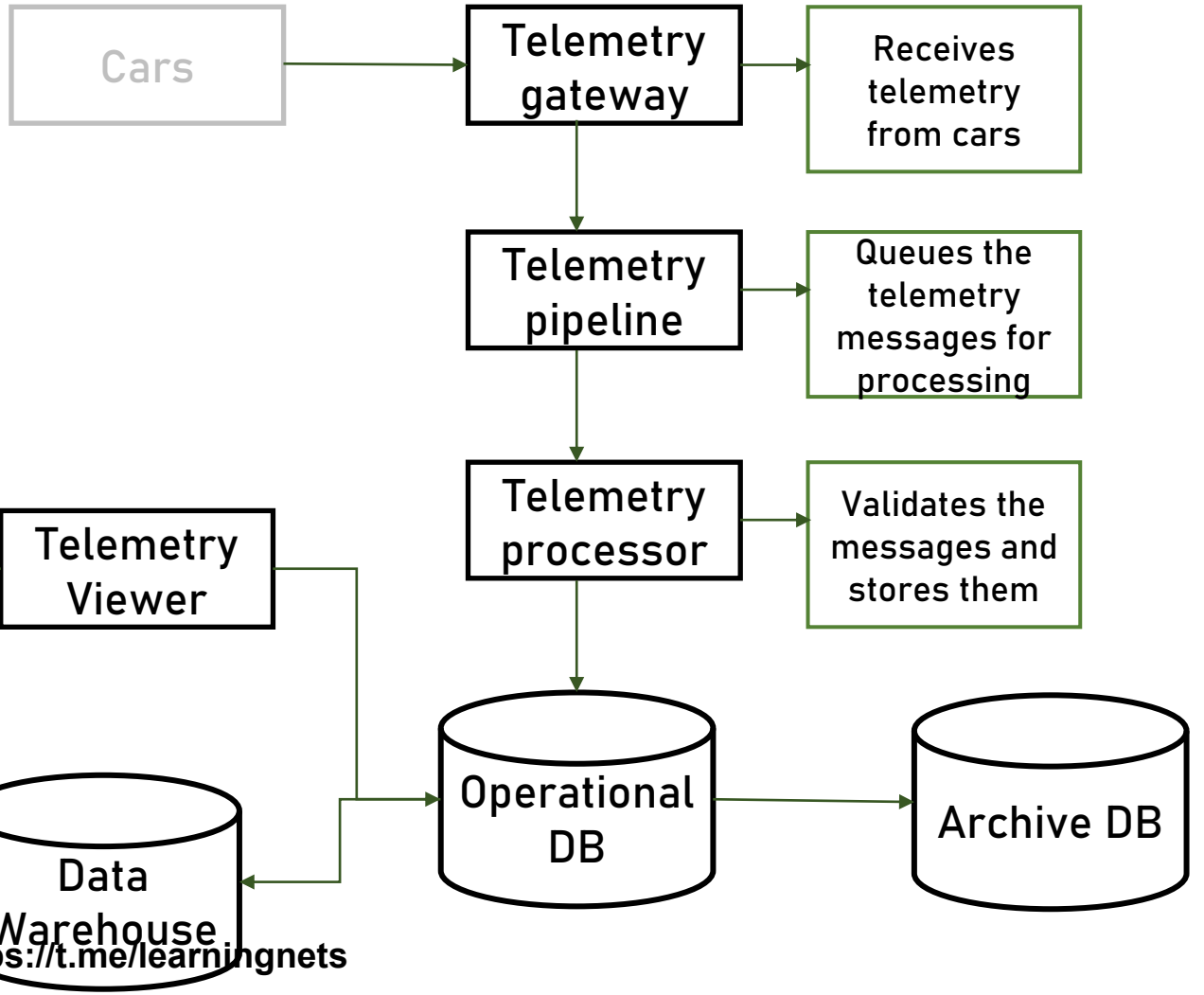
1. 10 Concurrent users
2. 7,000 msgs/sec
3. Max data in the operational DB: 4TB
4. Mission critical
5. Performance is critical

Muscar.auto

Components

Based on requirements:

1. Receive telemetry
2. Validate telemetry
3. Store telemetry
4. Query & analyze telemetry

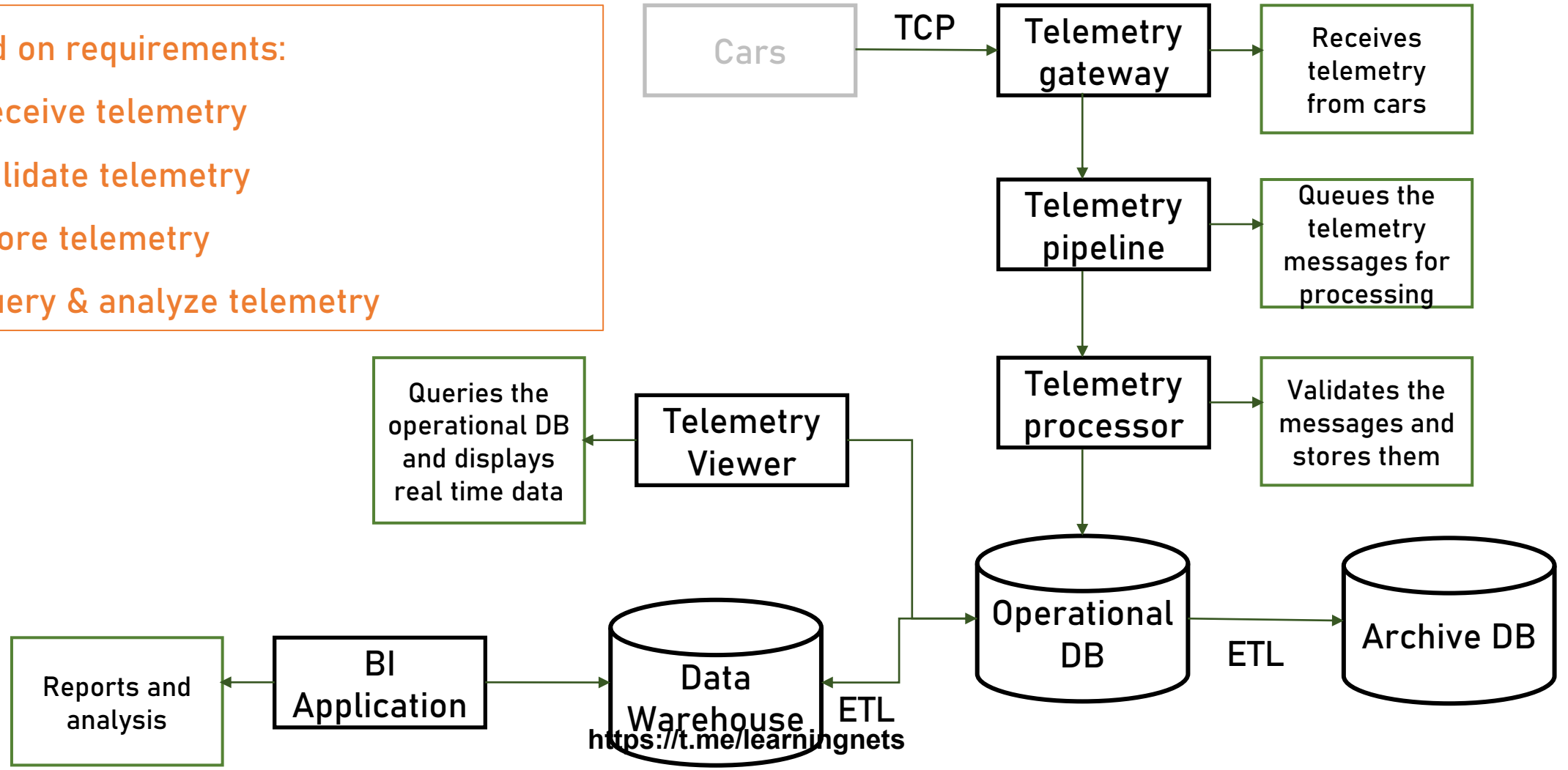


Muscar.auto

Messaging

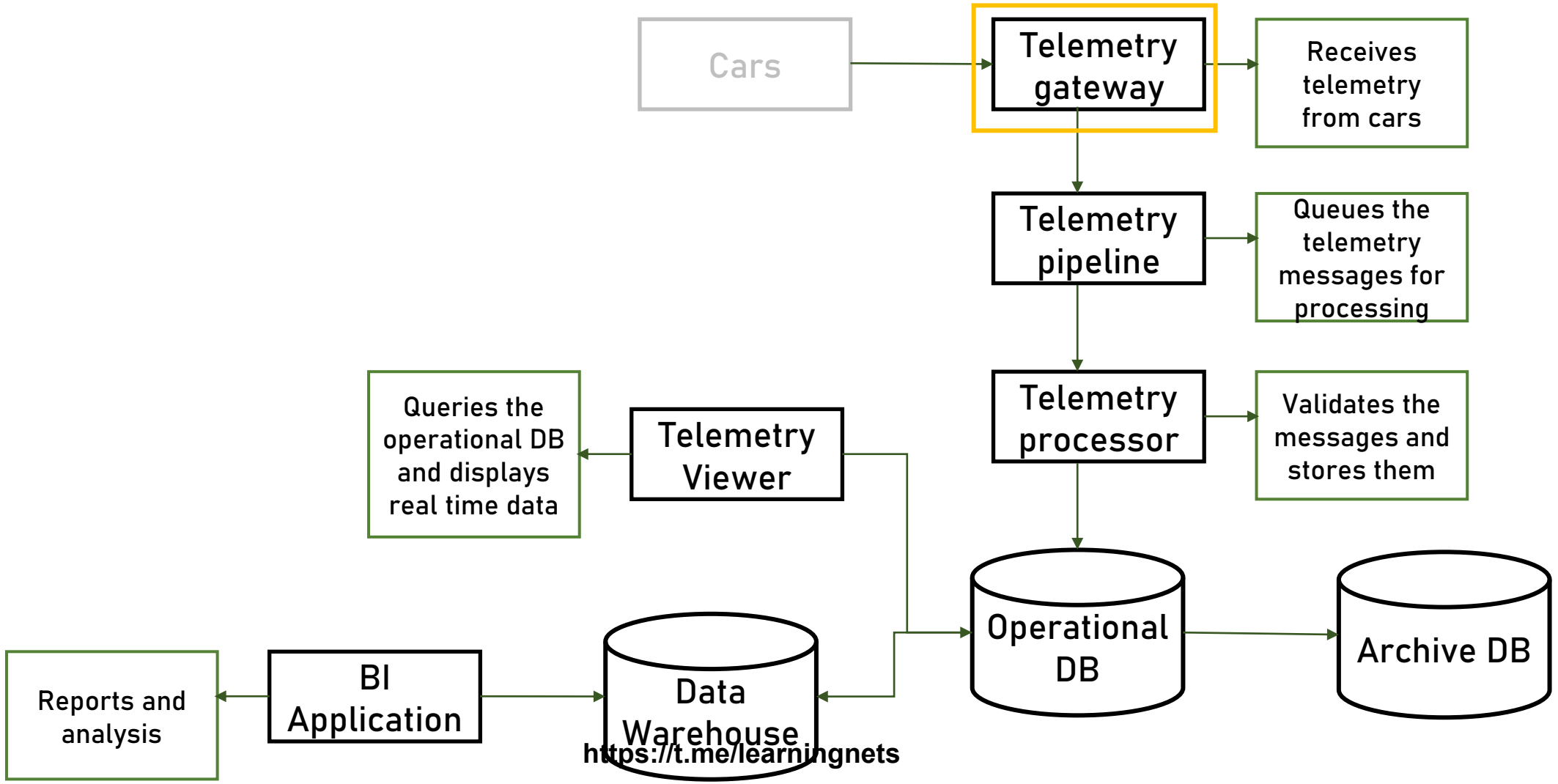
Based on requirements:

1. Receive telemetry
2. Validate telemetry
3. Store telemetry
4. Query & analyze telemetry



Muscar.auto

Components



Muscar.auto

Telemetry Gateway

What it does:

- Receives telemetry data from cars using TCP

- Pushes the telemetry data to the pipeline

Application Type

- Web App & Web API 
- Mobile App 
- Console 
- Service 
- Desktop App 

Muscar.auto

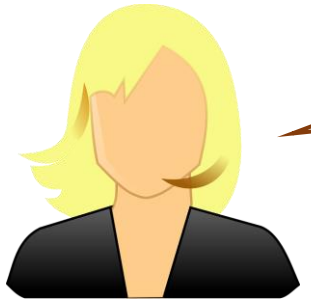
Technology Stack

Considerations:

- Load (7,000 msgs/sec)
- Performance
- Team's current knowledge
- Environment (OS, etc)

Muscar.auto

Technology Stack



Our developers are familiar with Python,
and are experts in JavaScript.
In addition, we use only Linux servers.

Python can't be used for the gateway

Too slow

We look for a language with great performance,

runs on linux, and leverages current skills

(Python & JavaScript)

Muscar.auto

Technology Stack

- Great performance
- Runs on Linux
- Leverages JS skills

Listeners in Azure

No raw TCP support

Listeners in Azure

- Not ideal
- Requires the most manual maintenance
- …but allows most flexibility

Muscar.auto

Scaling

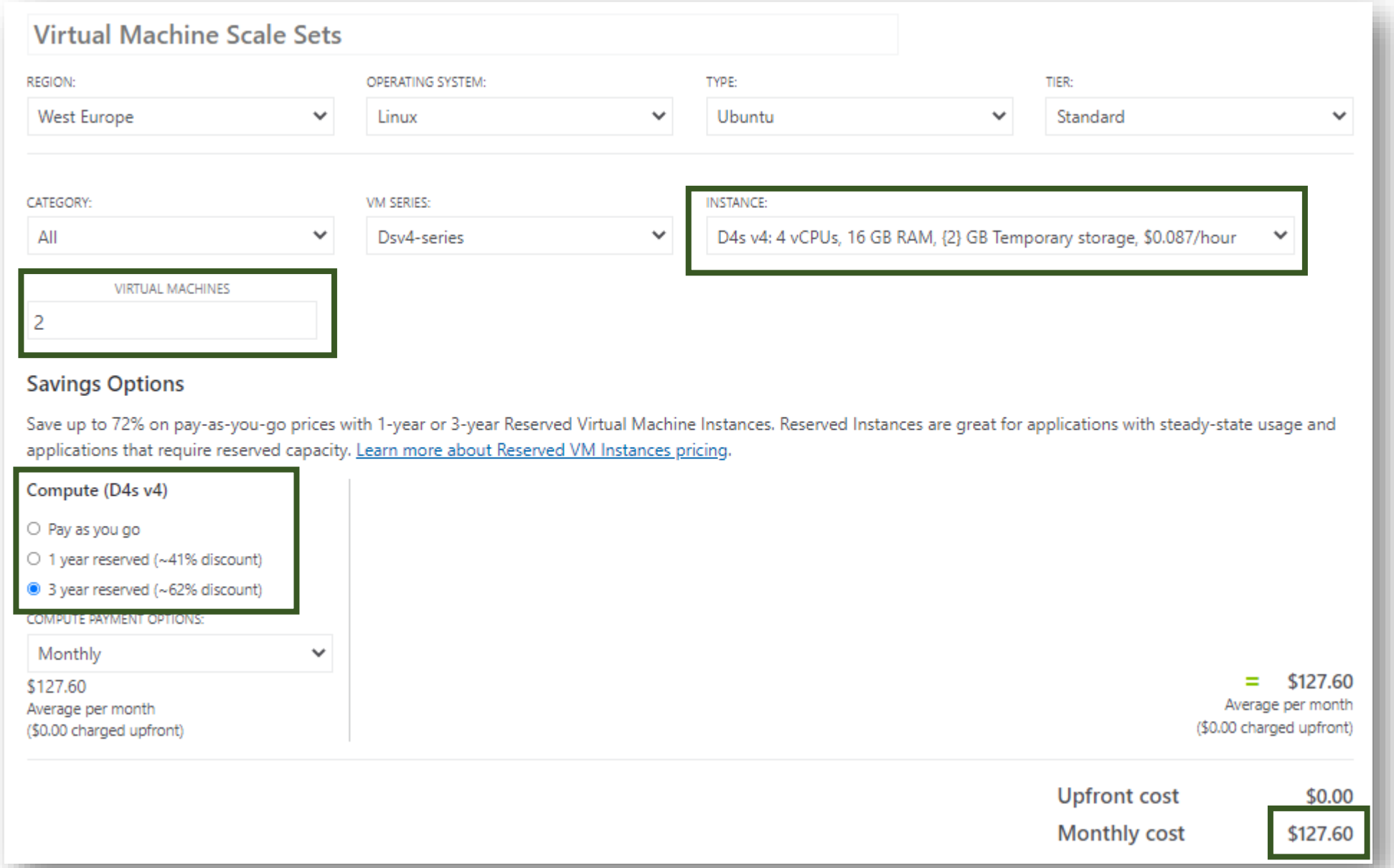
Remember:

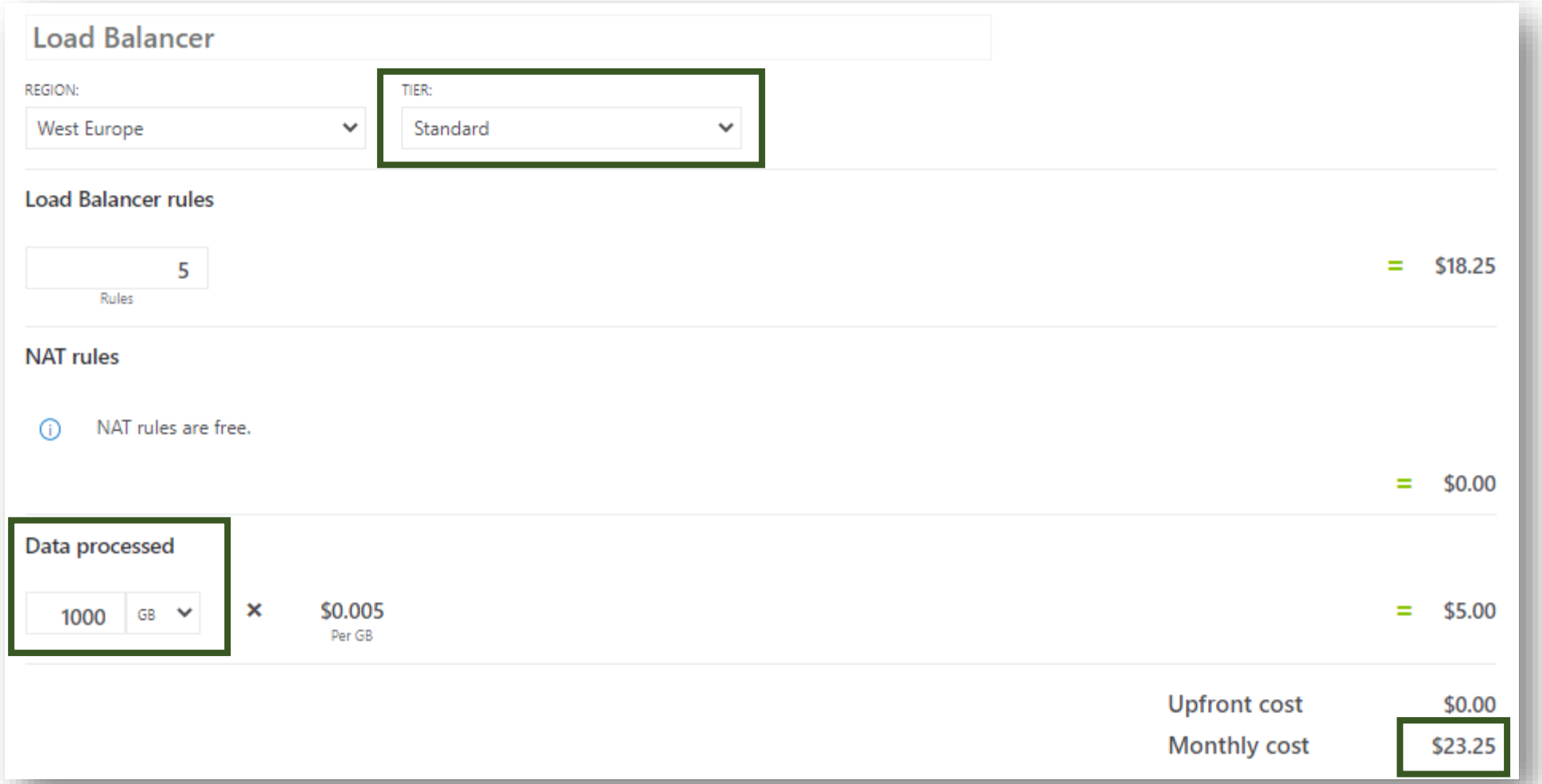
- Load (7,000 msgs/sec)
- Performance

Scaling

Scaling

 VM Scale Set +  Load Balancer

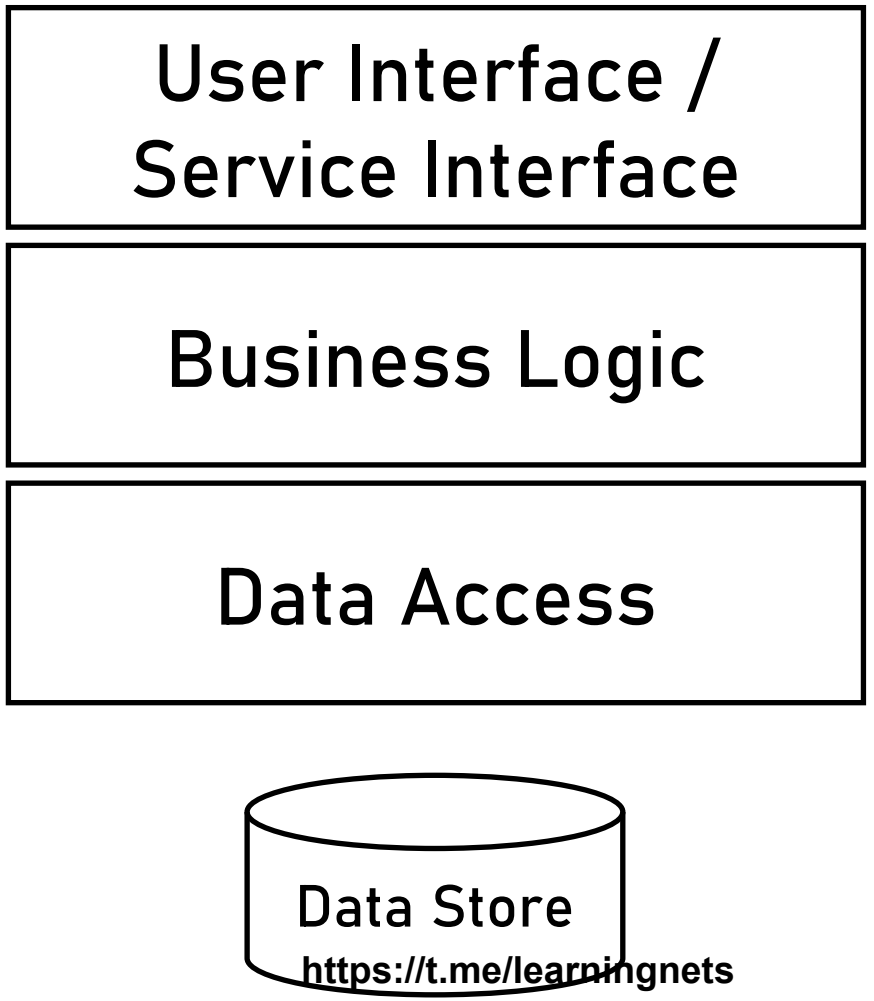




Muscar.auto

Architecture

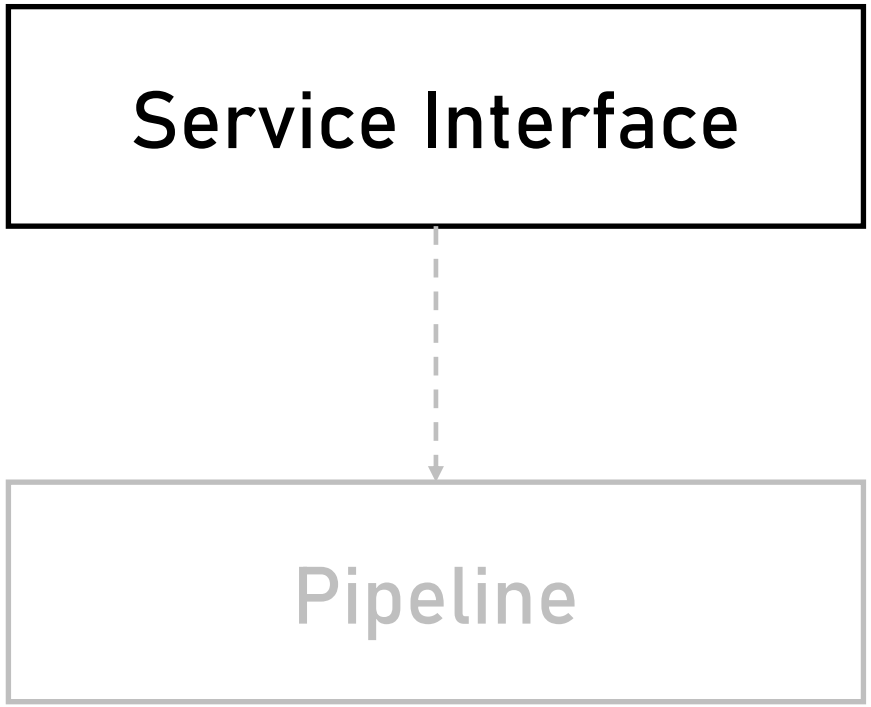
Traditional:



Muscar.auto

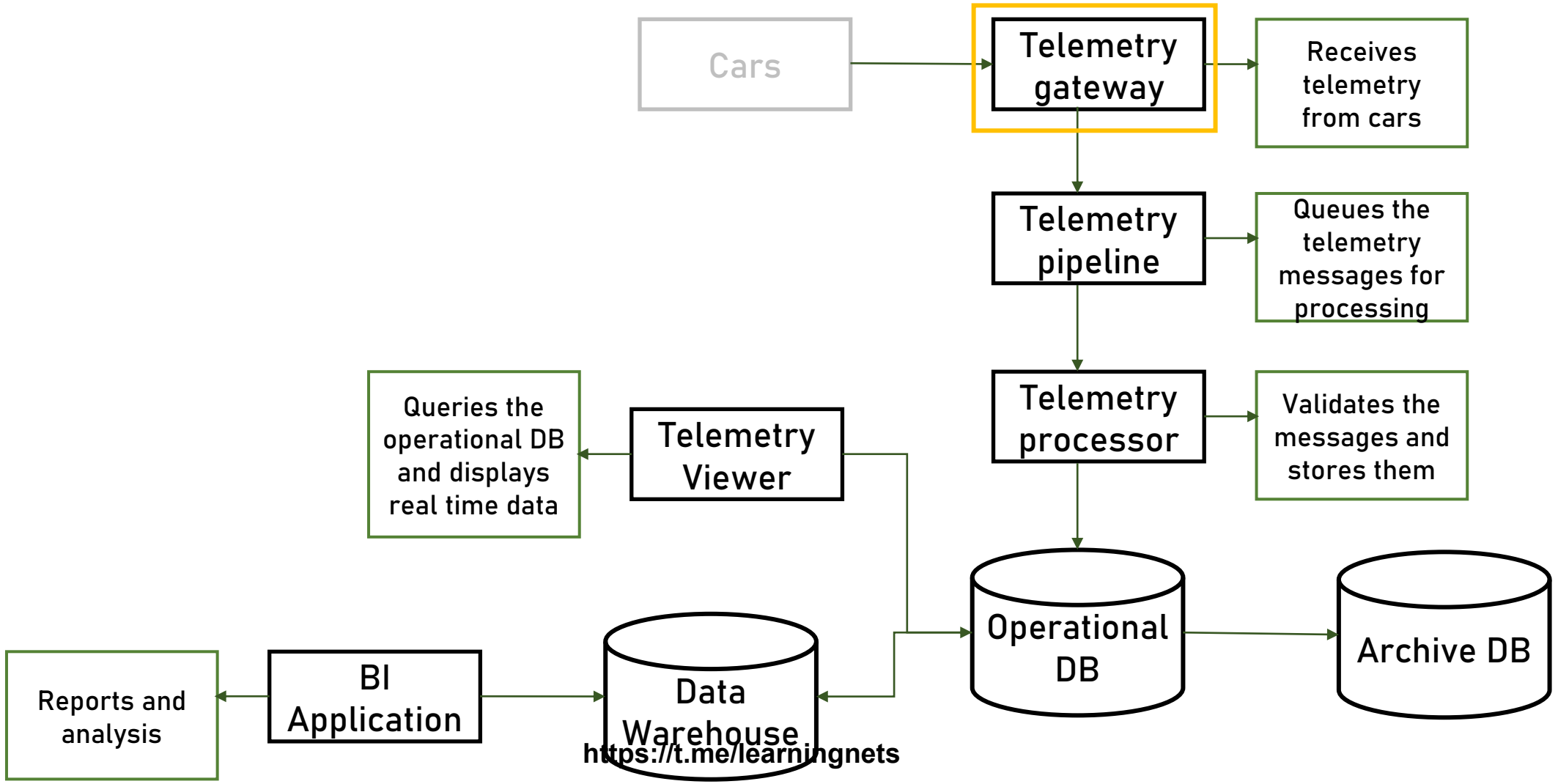
Architecture

In our case:



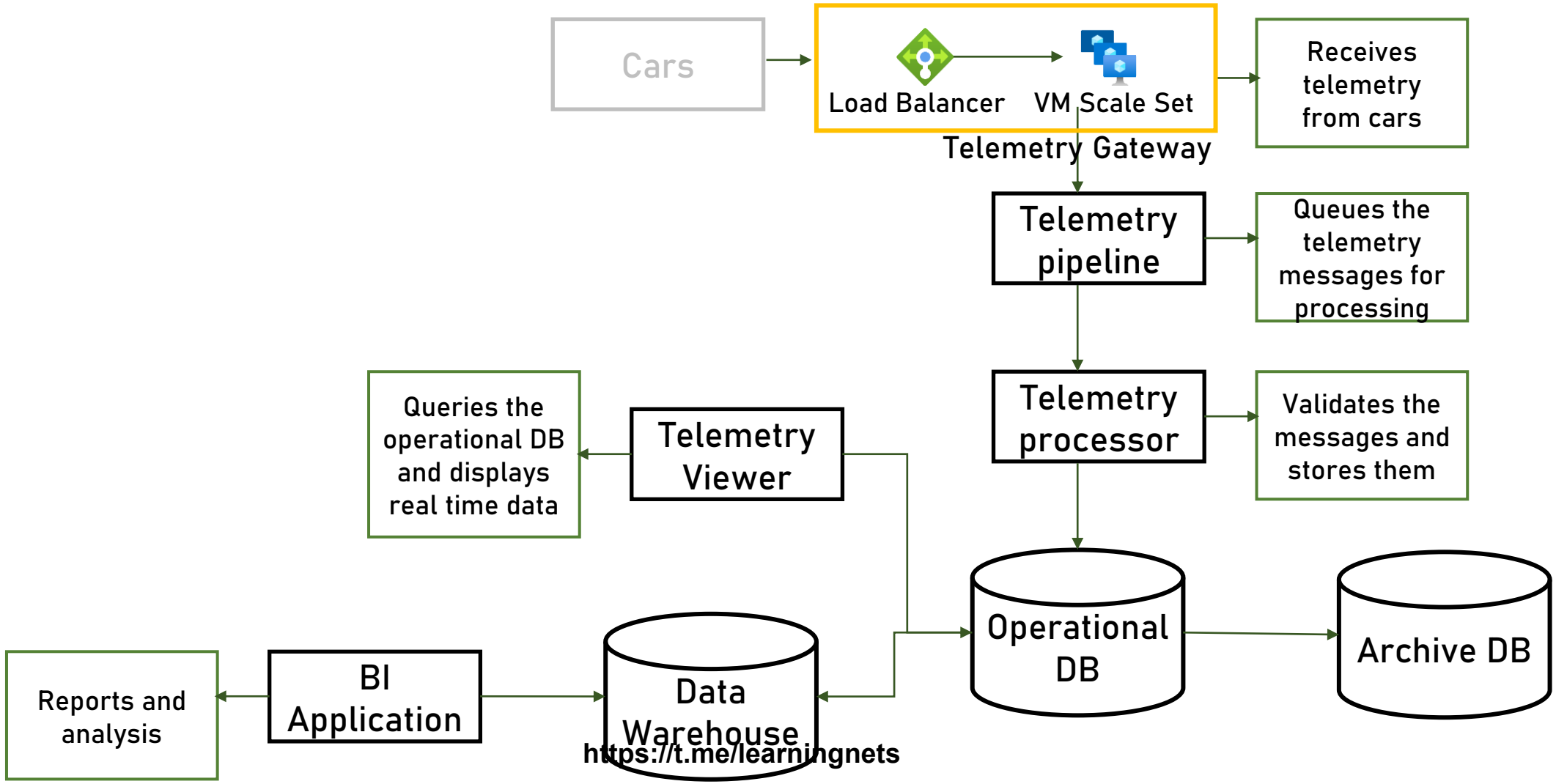
Muscar.auto

Components



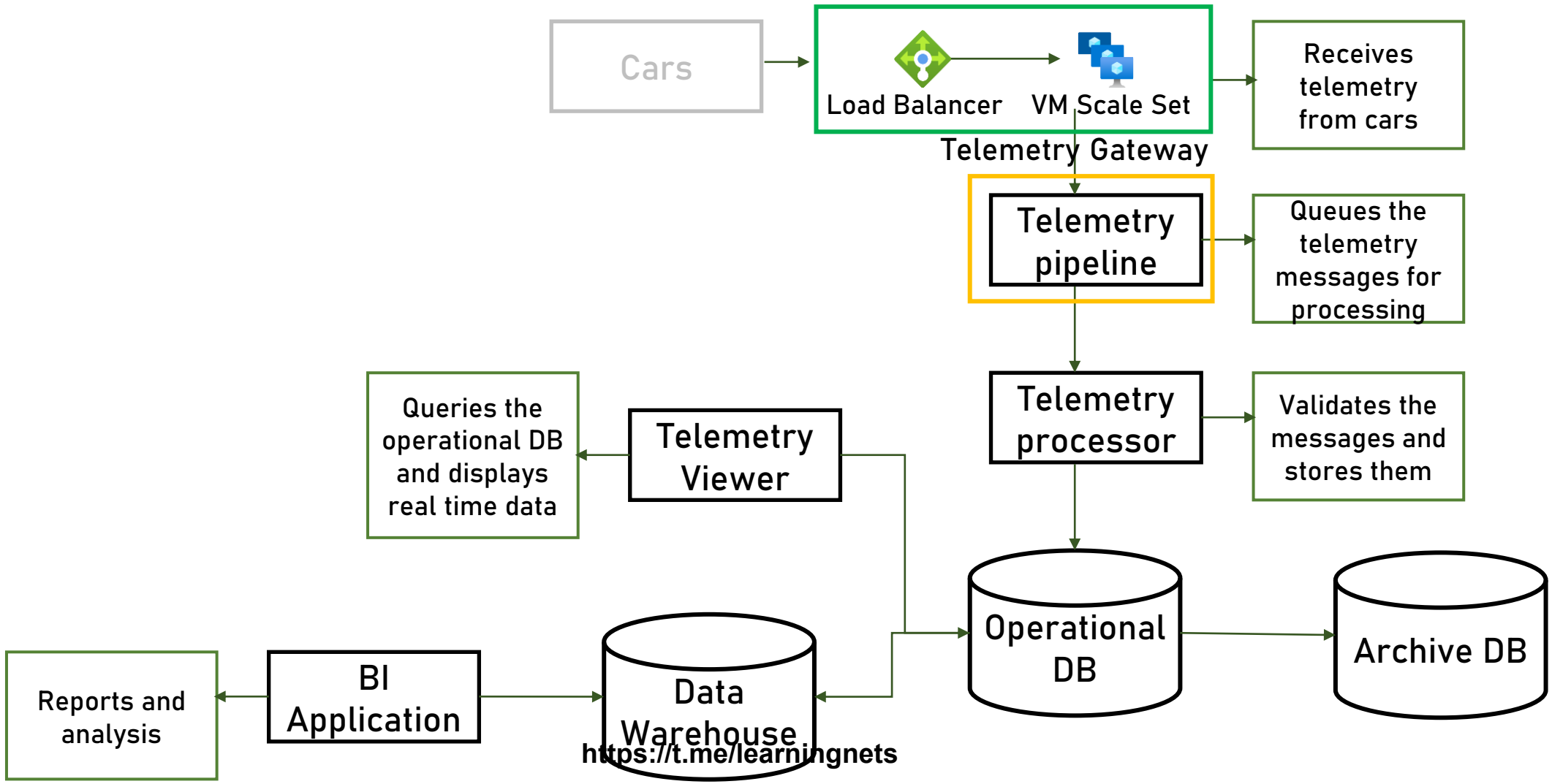
Muscar.auto

Components



Muscar.auto

Components



Muscar.auto

Telemetry Pipeline

What it does:

- Gets the telemetry messages from the gateway

- Queues the telemetry for further processing

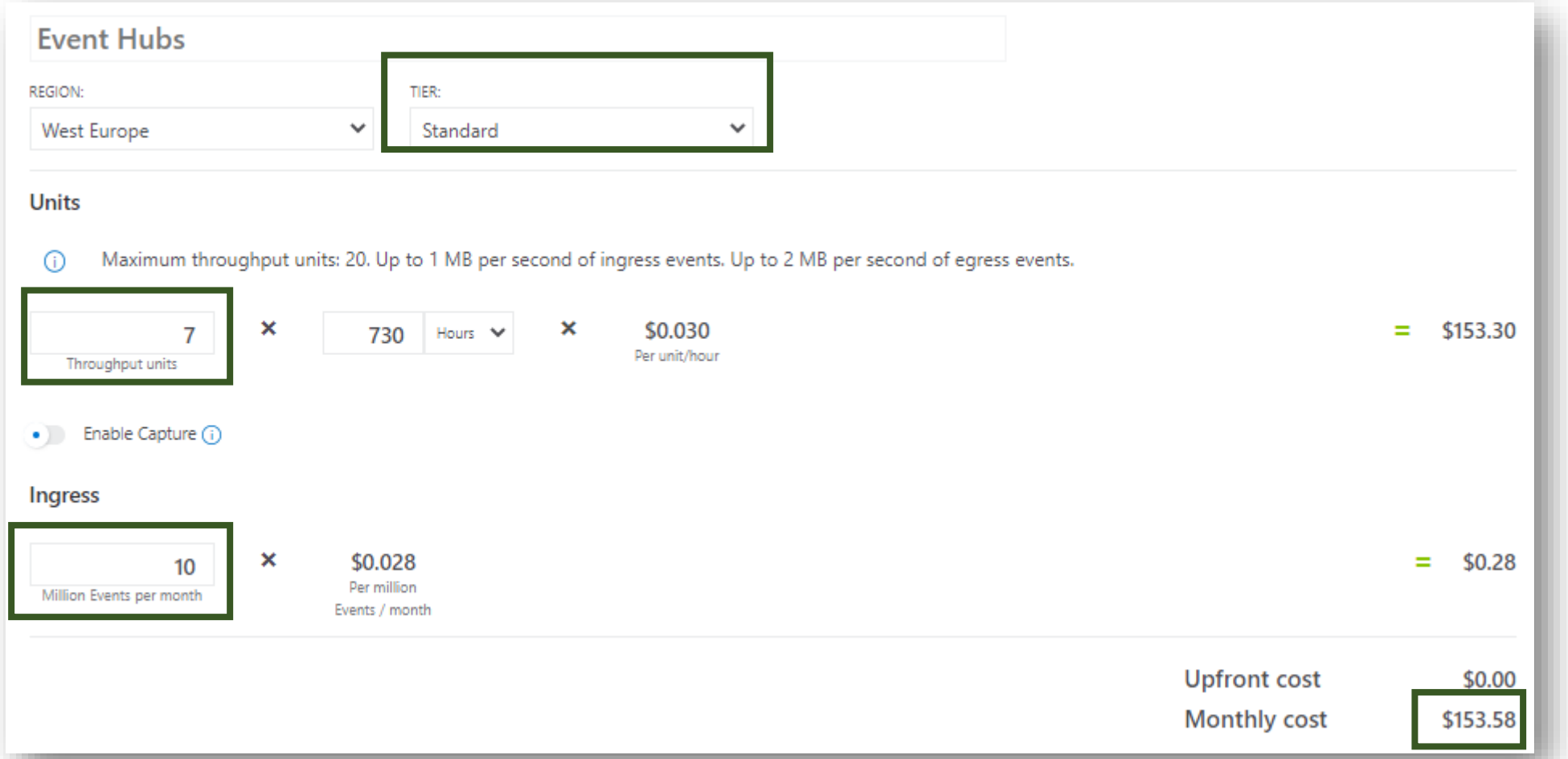
- Basically – a queue for streaming high volume data

Messaging in Azure

Service	Used For…	Guarantees Order	Max Msg Size	And also…
Storage Queue 	Dead simple queueing	Yes	64KB	Extremely simple, no additional cost
Event Grid 	Event driven architectures	No	1MB	Great integration with other services
Service Bus 	Advanced queueing solutions	Yes	256KB	Advanced messaging features, durable
Event Hubs 	Big data streaming	Yes	1MB	Low latency, designed for heavy load

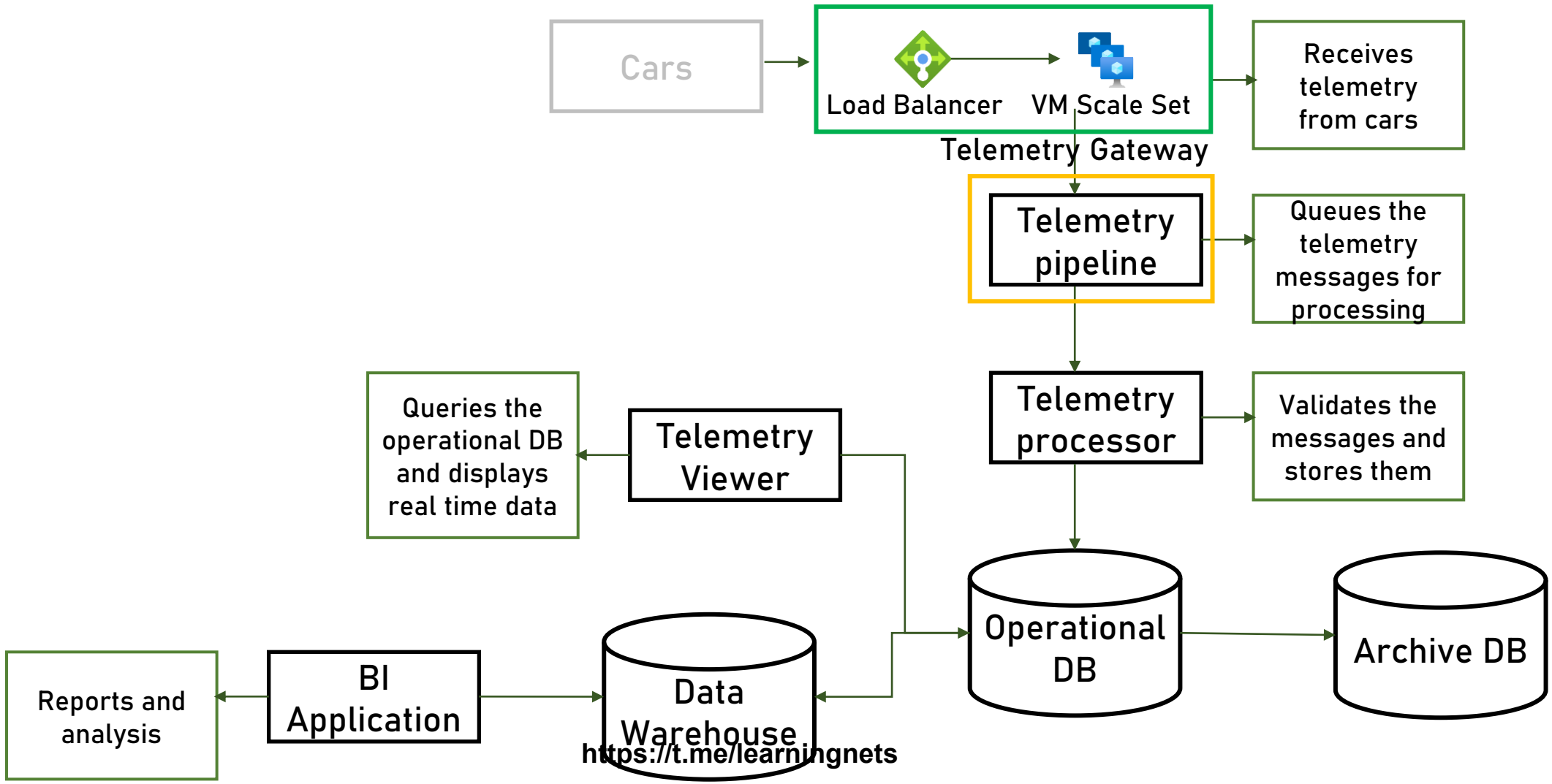
Muscar.auto

Each TU supports up to 1k msgs / sec



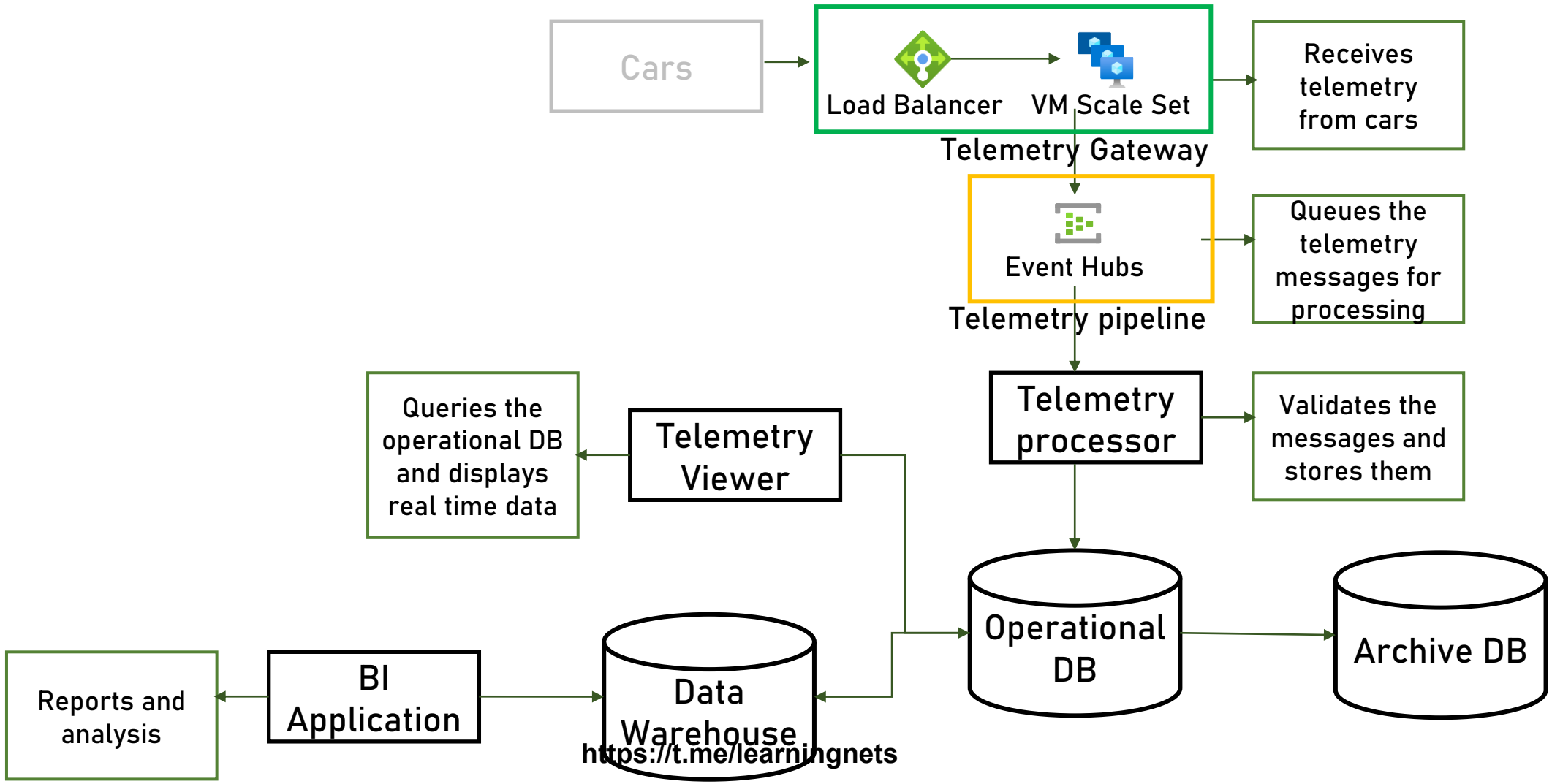
Muscar.auto

Components

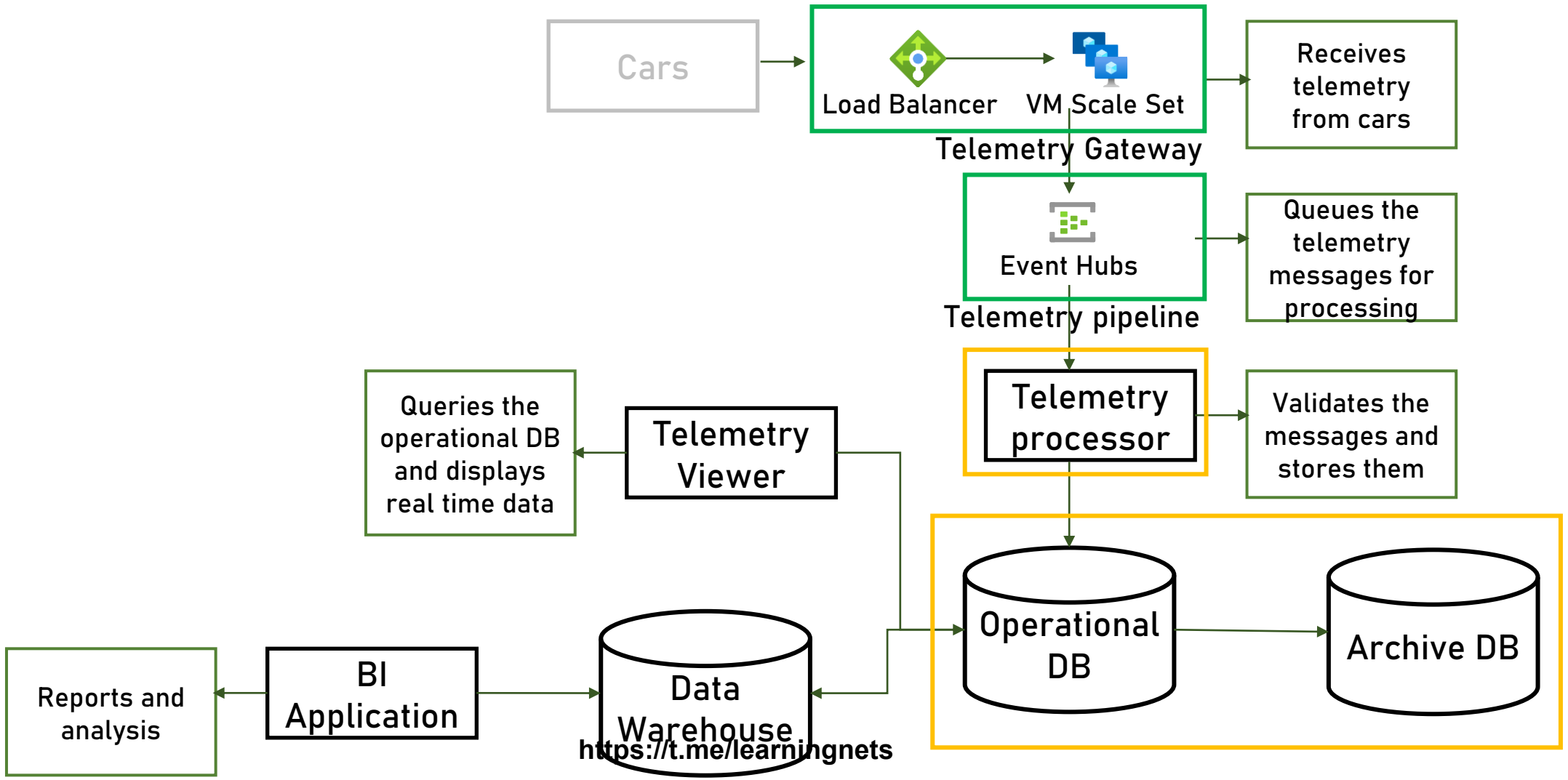


Muscar.auto

Components



Components



Muscar.auto

Telemetry Processor

What it does:

- Receives the messages from the pipeline

- Processes the messages (mainly validation)

- Stores the messages in a data store

Cloud Services

For:

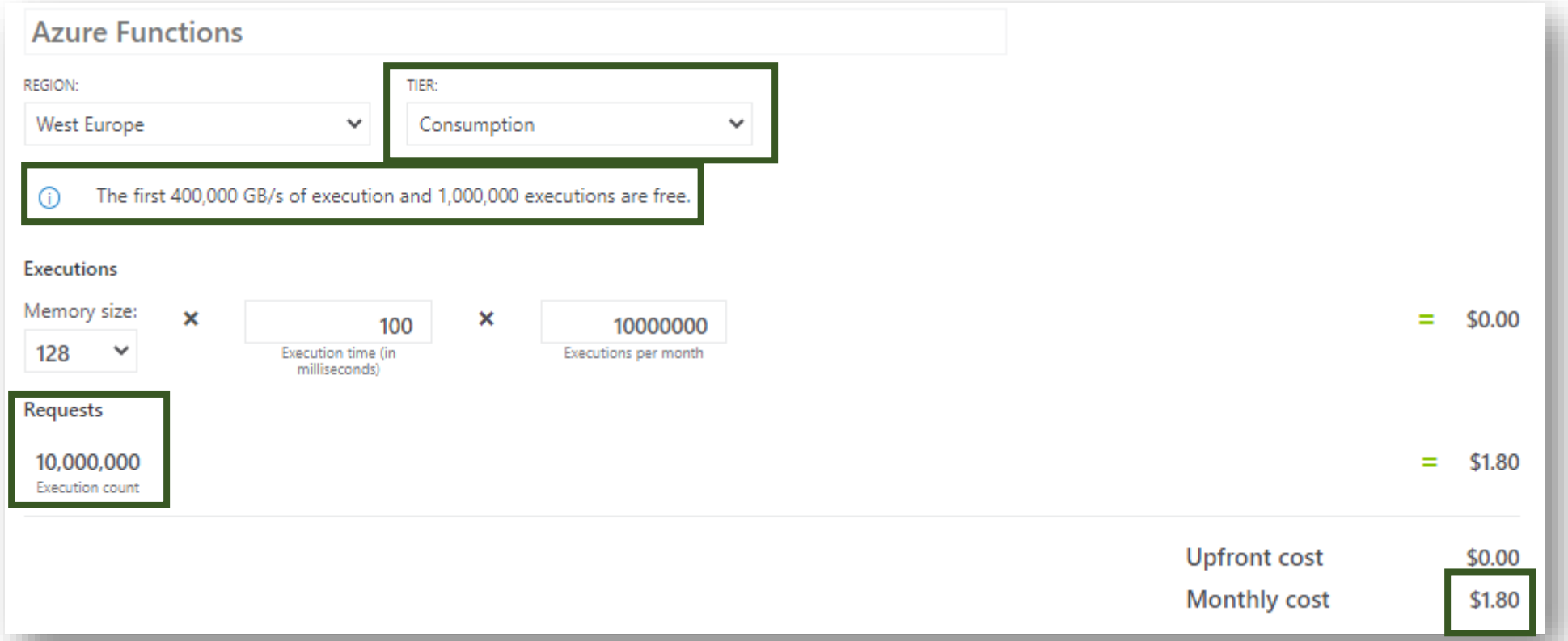
- The processor
- The datastore

Processor

Function App

- Designed for lightweight operations
- Great, built-in integration with Event Hubs
- Cost effective
- Autoscaling

Muscar.auto



Muscar.auto

Data store

What we're looking for:

- Schema-less message support
- Quick retrieval
- No complex queries

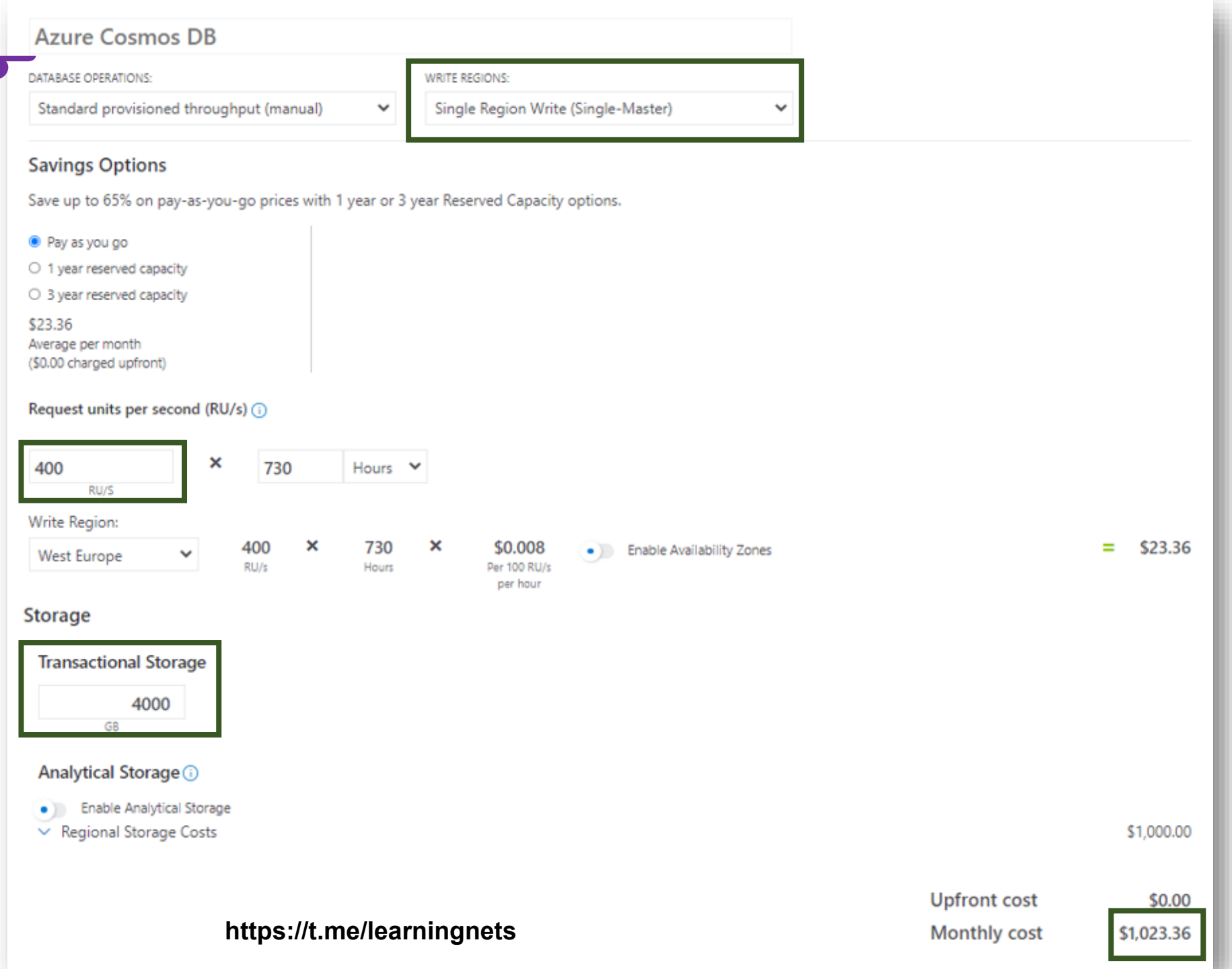
Muscar.auto

Technology Stack

Cosmos DB

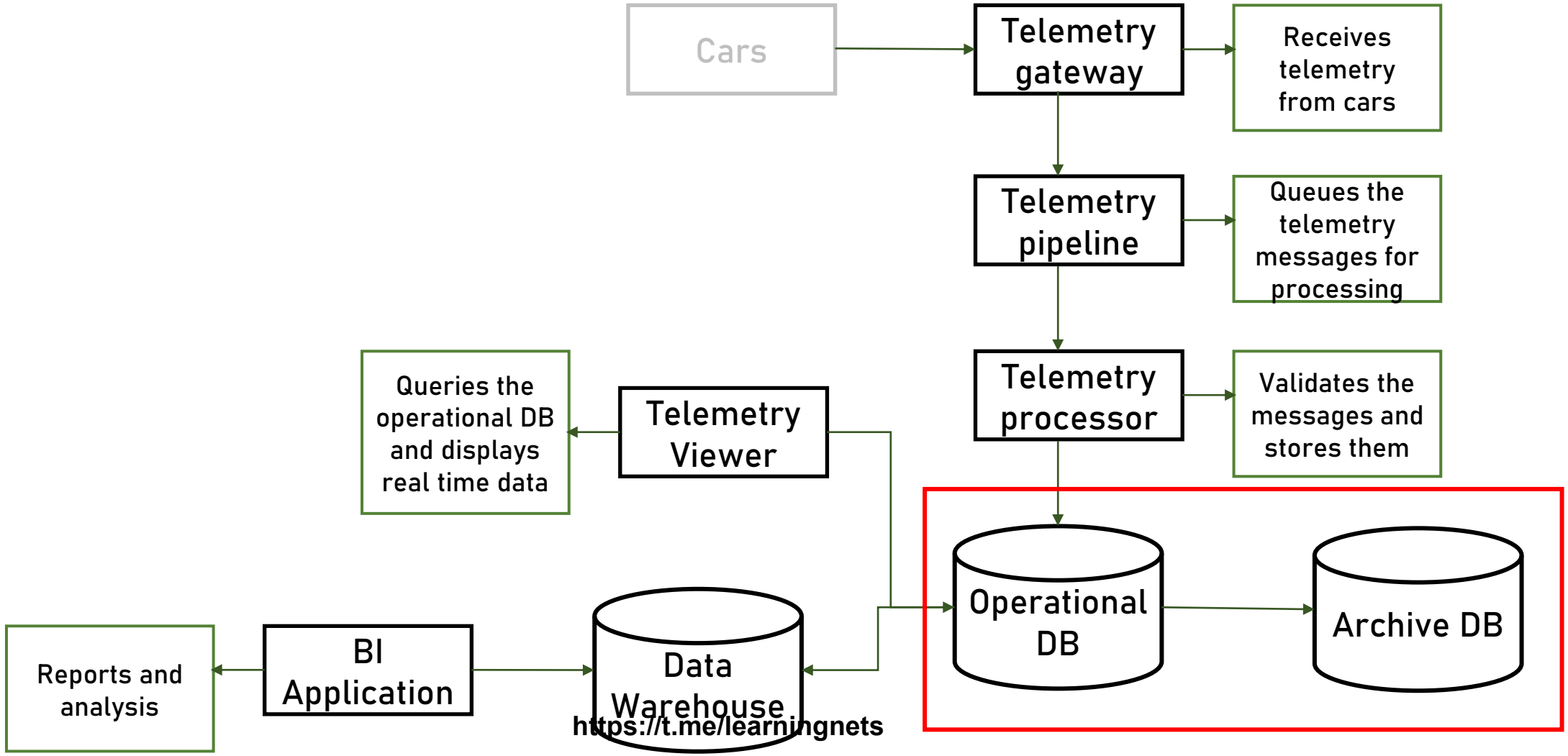
- Schema-less message support
- Quick retrieval
- No complex queries
- In addition:
  - Multi-region read / write
  - Multiple APIs
  - Great performance

No need for 7000 RU/s, Event Hubs balances load



Muscar.auto

Components



Muscar.auto

Archive Database

Archive– what we're looking for:

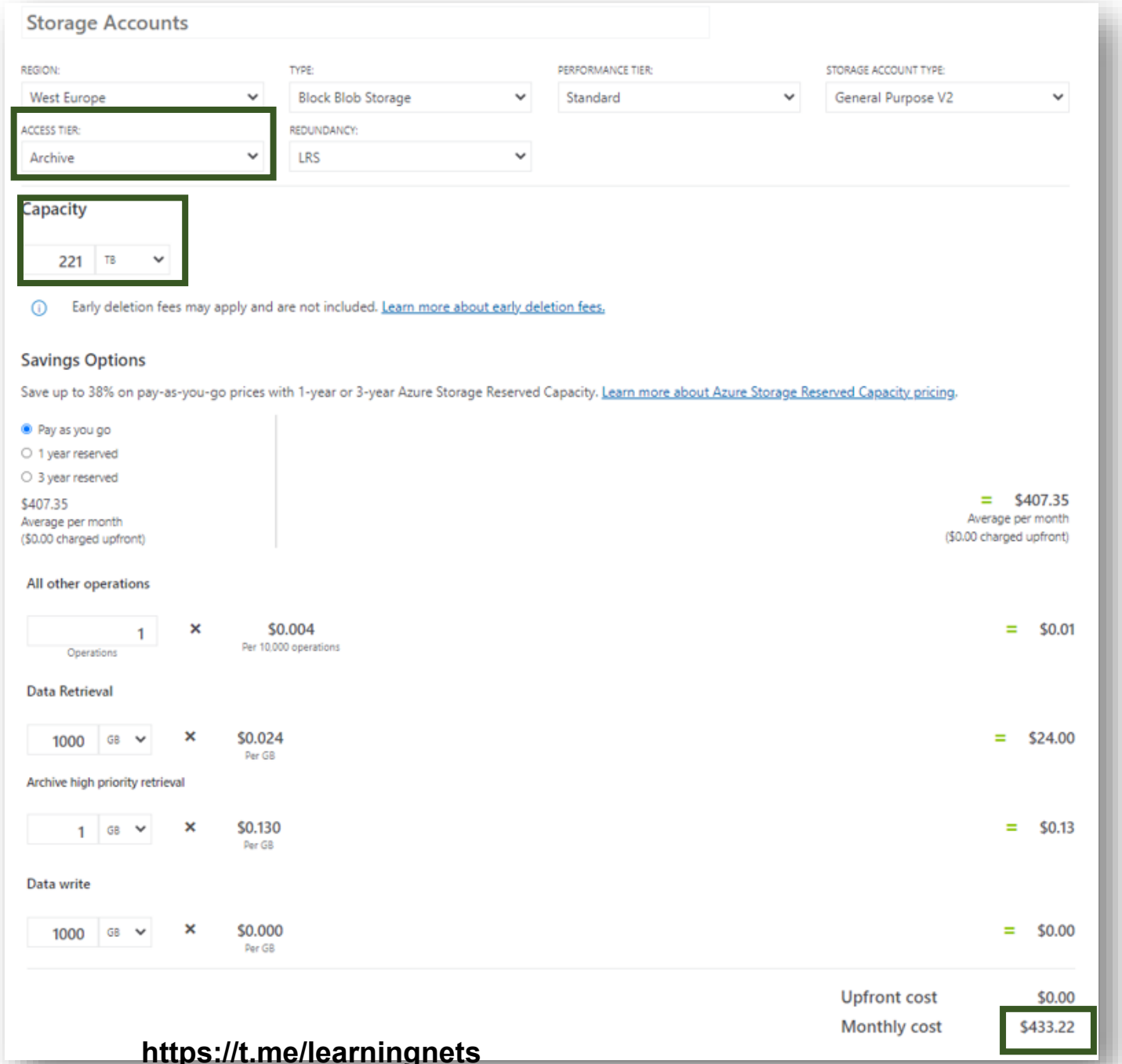
- Support for a huge amount of data (221TB / Year)
- Not accessed frequently
- No need for fast retrieval
- Save costs

Archive Database

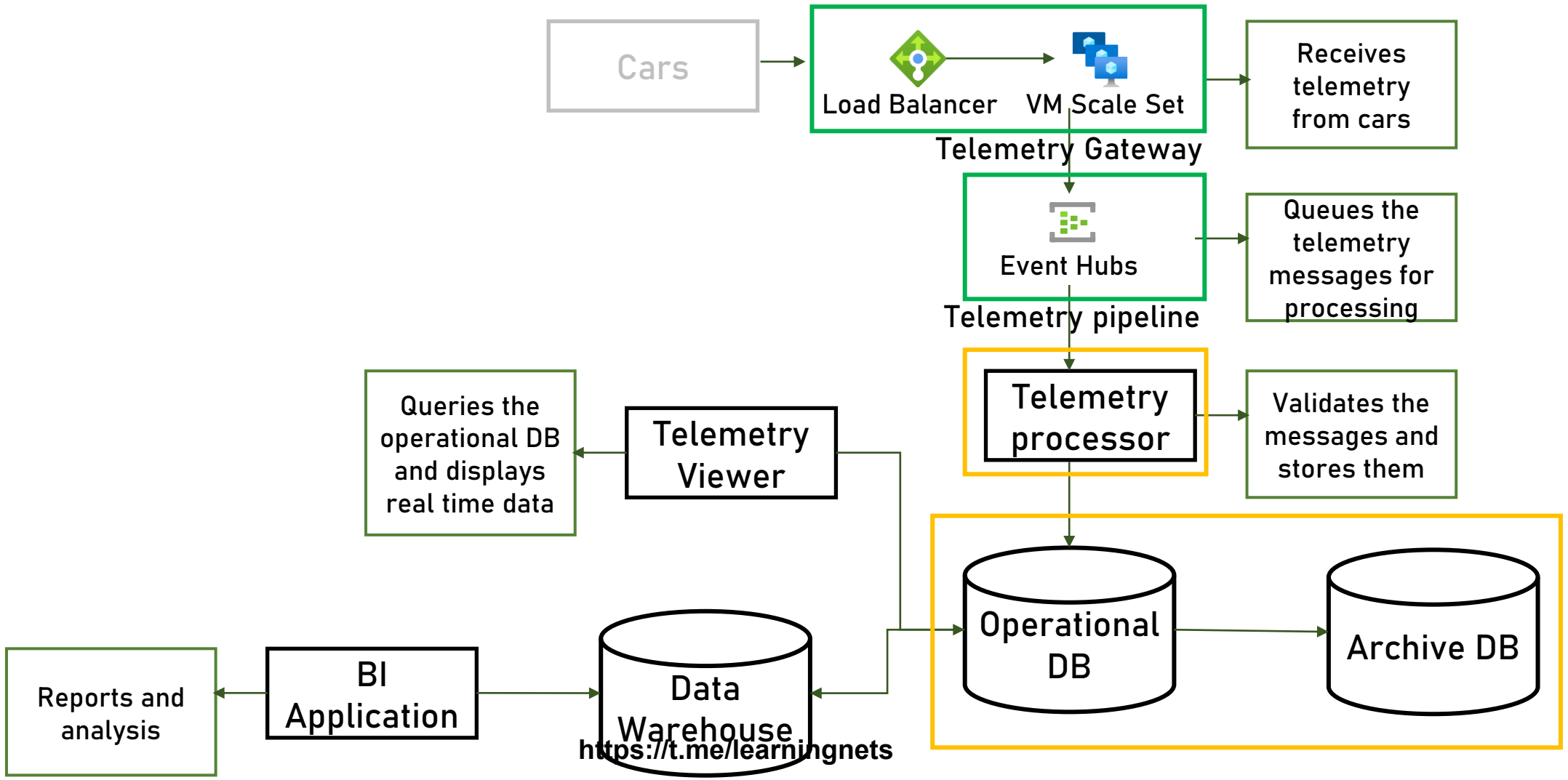
Storage Account

- Huge amounts of data (221TB / Year)
- Not accessed frequently
- No need for fast retrieval
- Save costs

Muscar.auto

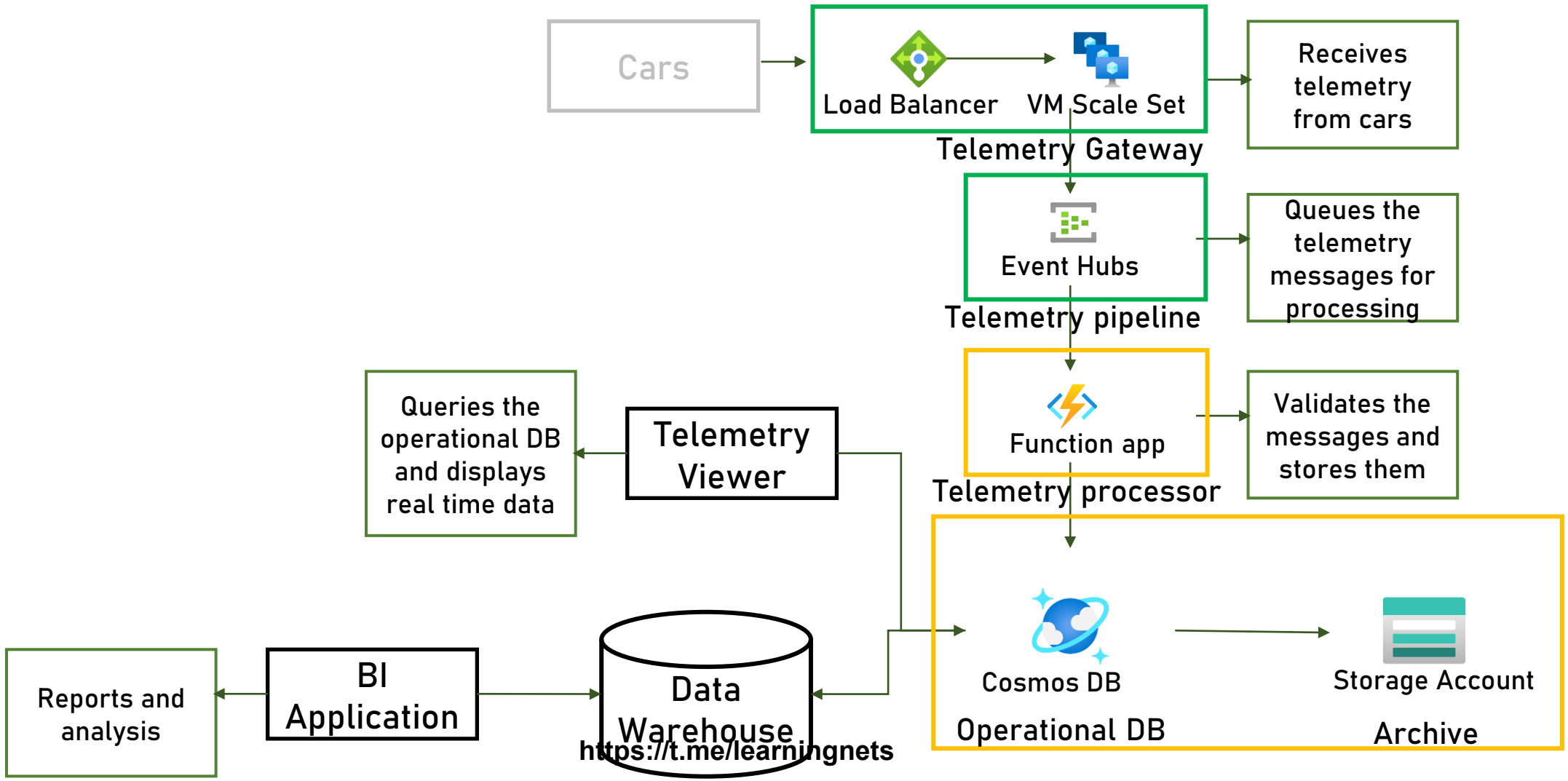


Components

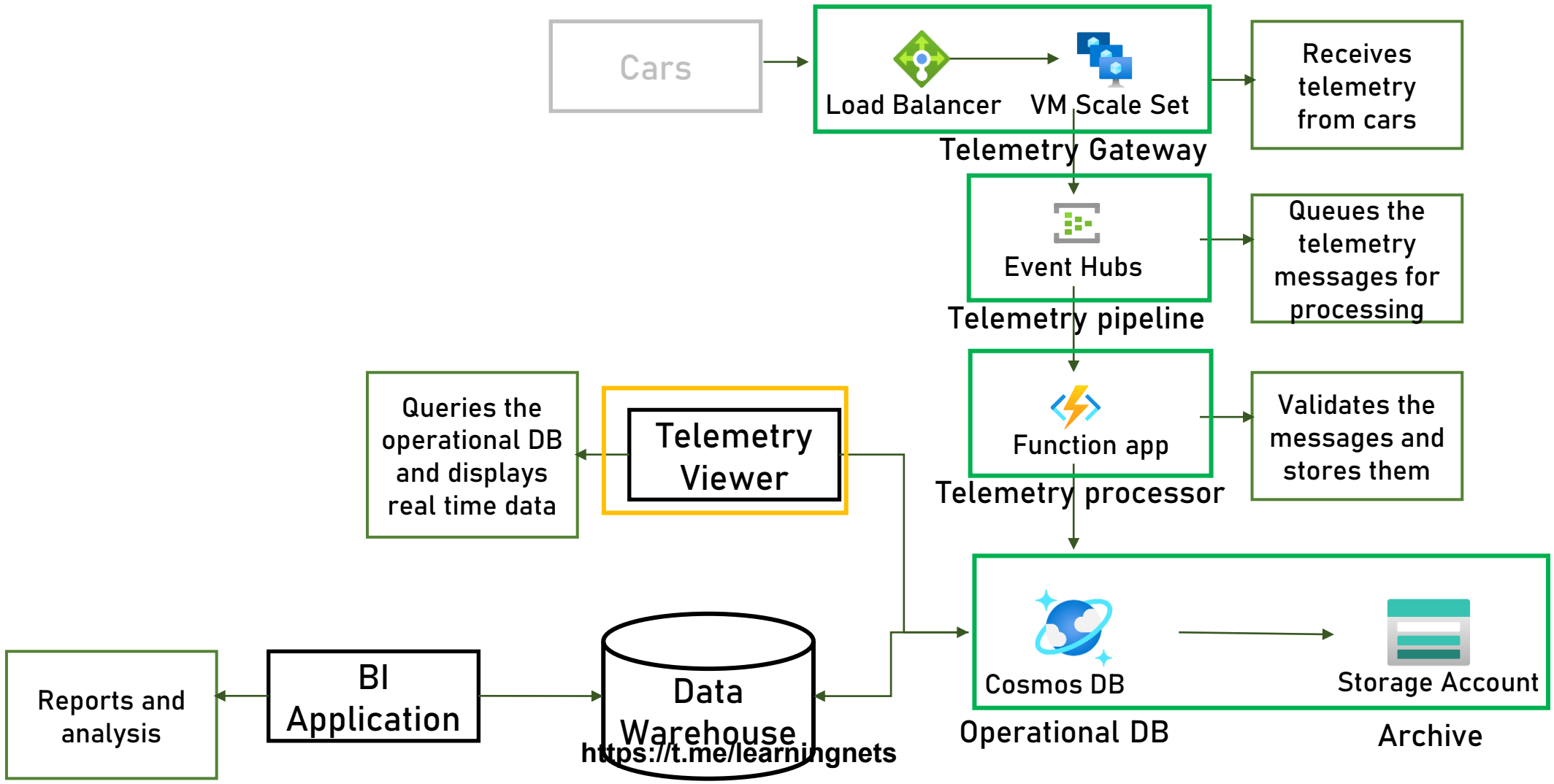


Muscar.auto

Components



Components



Muscar.auto

Telemetry Viewer

What it does:

- Allows end users to query telemetry data

- Displays real time data

What it doesn't:

- Analyzes the data

Application Type

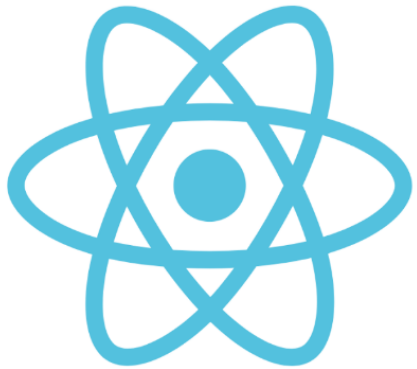
- Web App & Web API 
- Mobile App 
- Console 
- Service 
- Desktop App 

Muscar.auto

Technology Stack

Back End

Front End



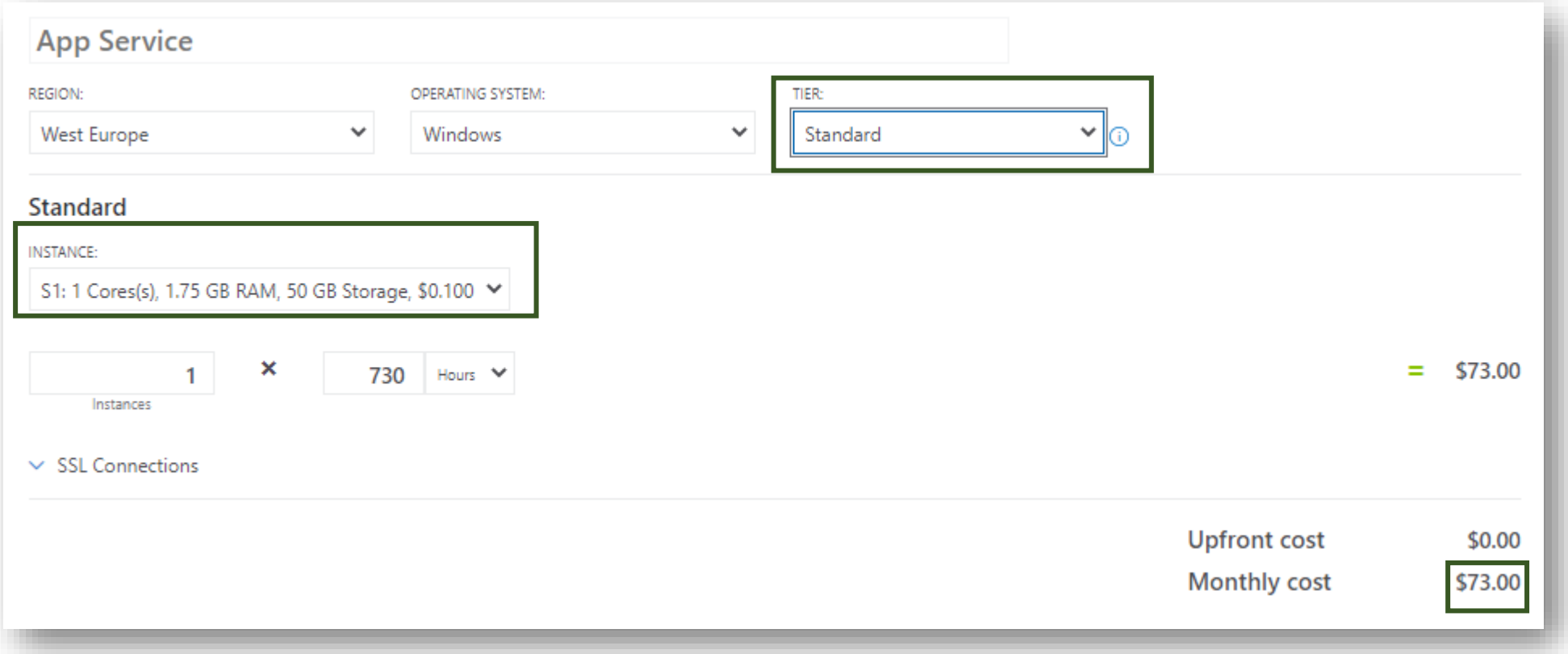
React JS

Azure Web API

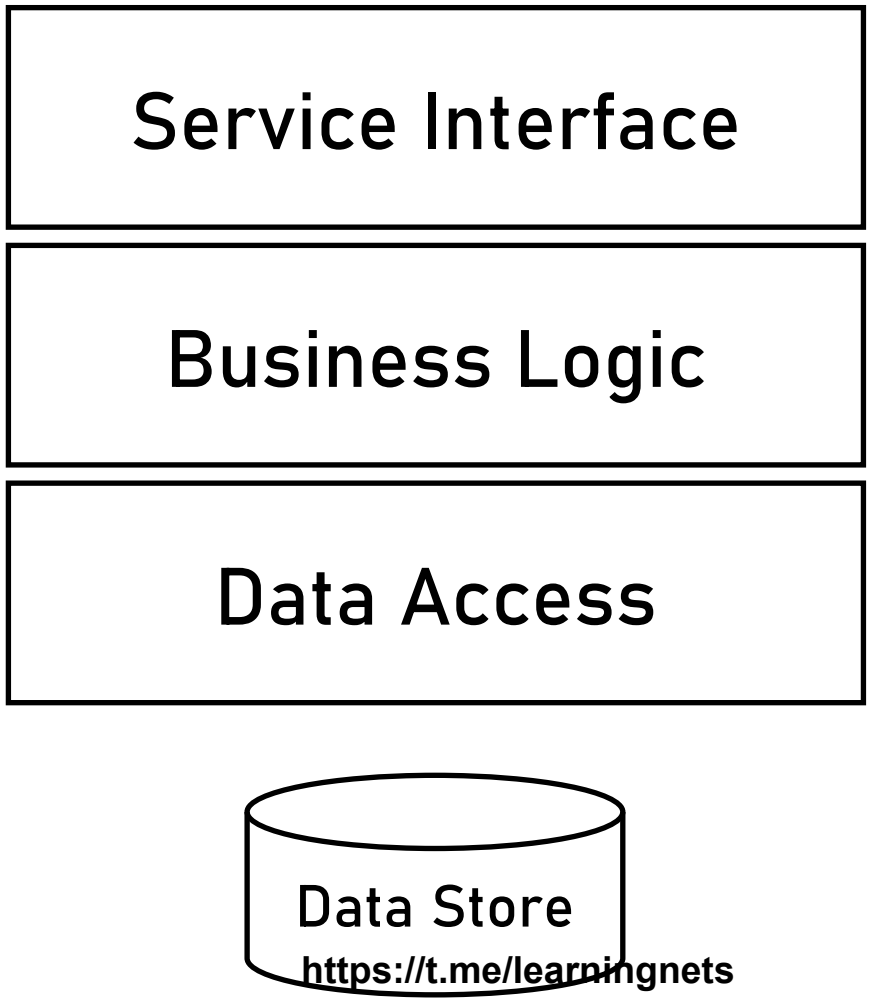
- Fully managed web app & API
- Supports many platforms
- Autoscale
- Support for WebJobs

Muscar.auto

Azure Web API



Architecture



Muscar.auto

API

- Get latest errors for all cars
- Get latest telemetry for specific car
- Get latest errors for specific car

Muscar.auto

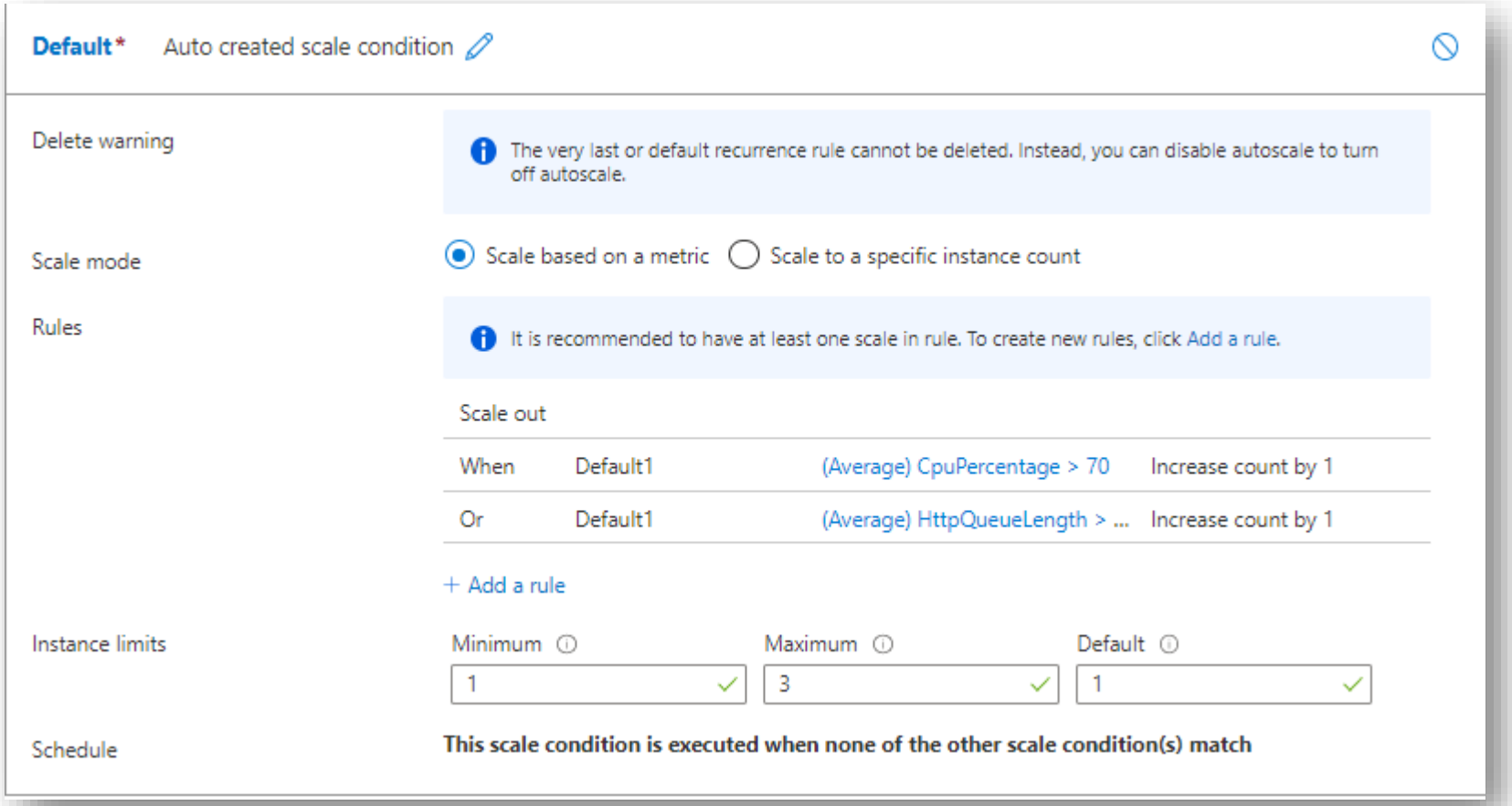
API

Functionality	Path	Return Codes
Get latest errors for all cars	`GET /api/v1/telemetry/errors`	`200 OK`
Get latest telemetry for specific car	`GET /api/v1/telemetry/{carId}`	`200 OK` `404 Not Found`
Get latest errors for specific car	`GET /api/v1/telemetry/errors/{carId}`	`200 Ok` `404 Not Found`

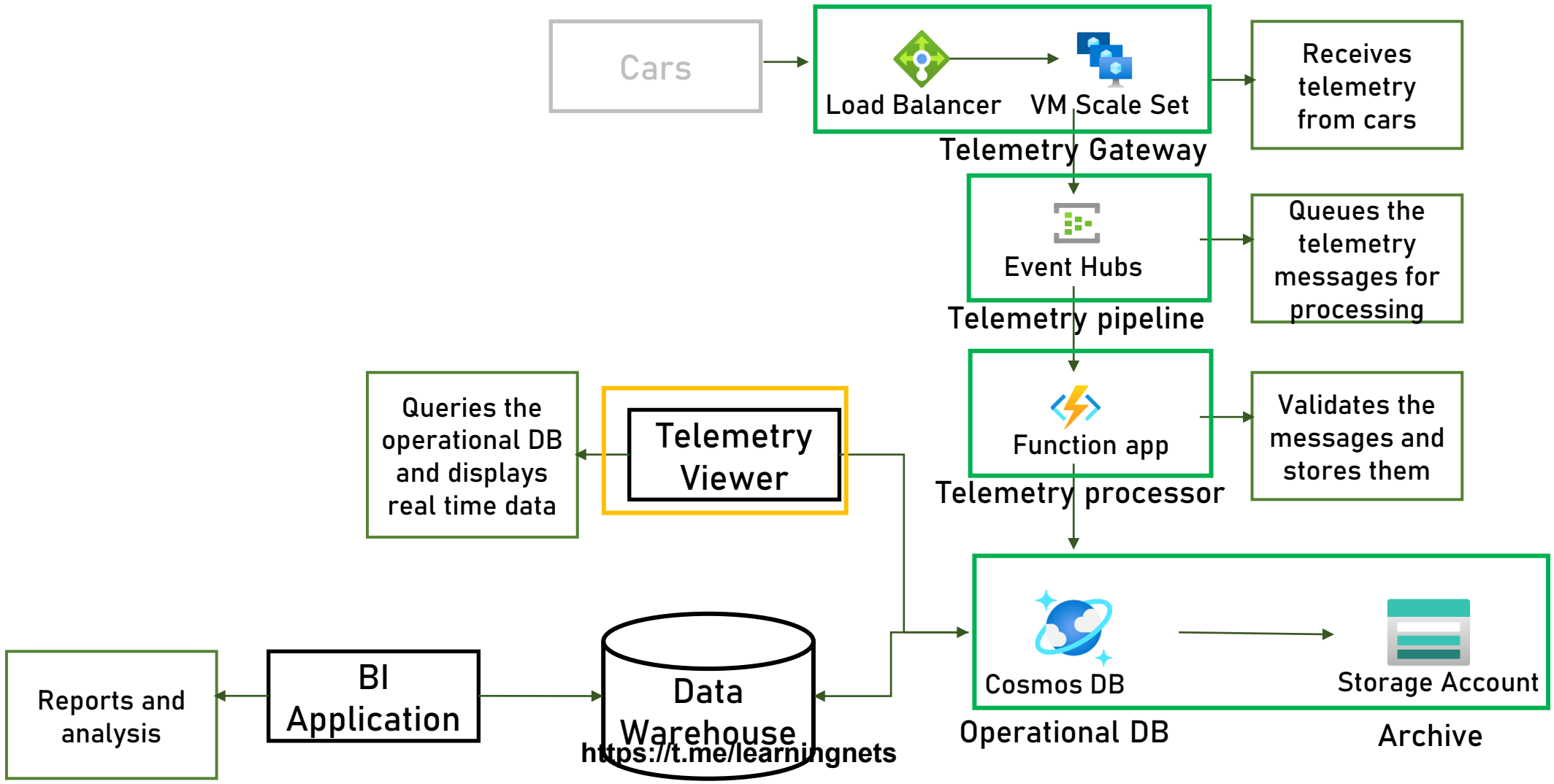
Muscar.auto

Telemetry Viewer Redundancy

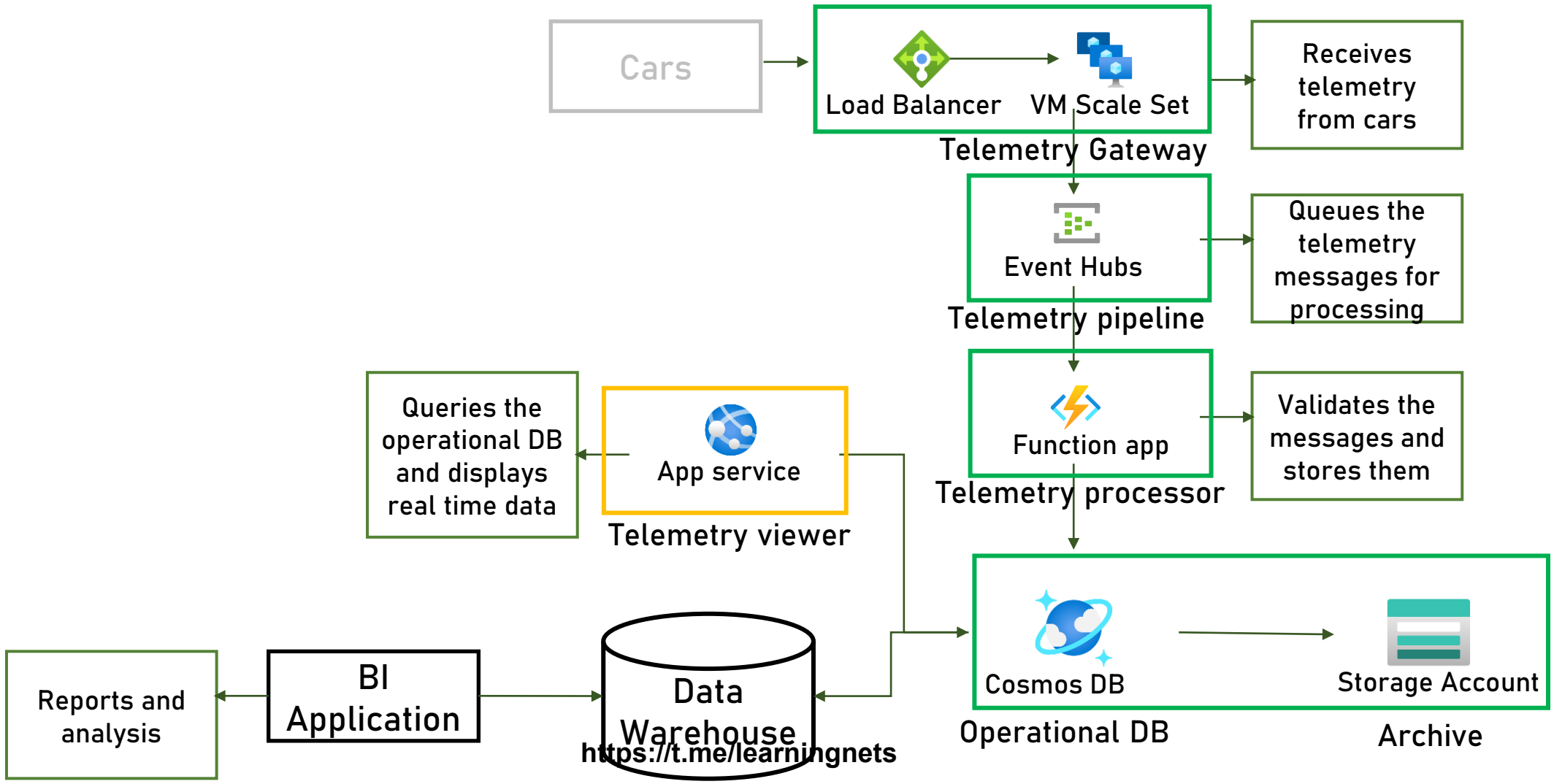
App service auto scale



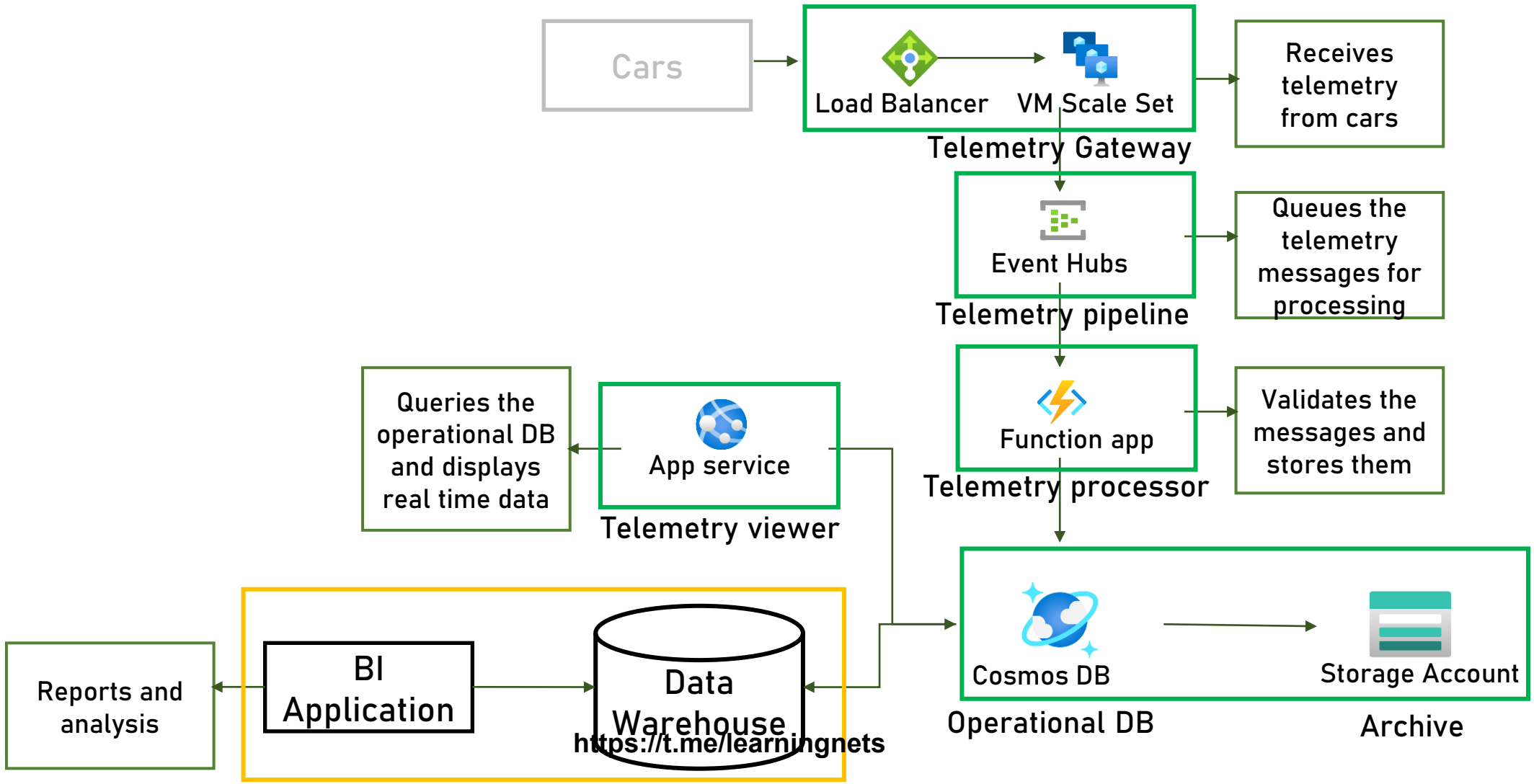
Components



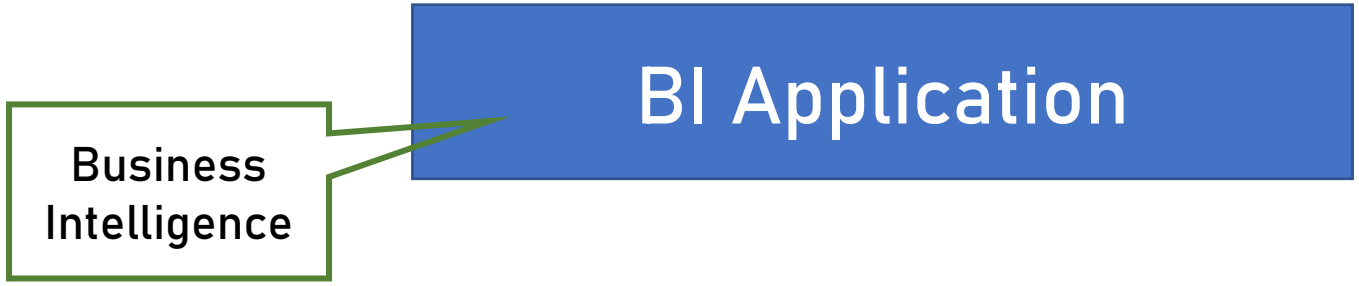
Components



Components



Muscar.auto



What it does:

- Analyzes telemetry data

- Displays custom reports about the data, trends, forecasts etc.

  - How many cars did break during the last month?

  - What is the total distance the cars drove?

Application Type

- Doesn't matter
- BI Application is ALWAYS based on an existing tool

Muscar.auto

BI Tools

BI Tools

Figure 1. Magic Quadrant for Analytics and Business Intelligence Platforms

Source: Gartner (February 2019)

BI Tools

- An important lesson:
  - Designing BI solution is NOT part of the architect's job
  - ALWAYS use BI expert for this task

Security

- Pay attention to:
  - Public accessible databases
  - Unprotected access to App Service

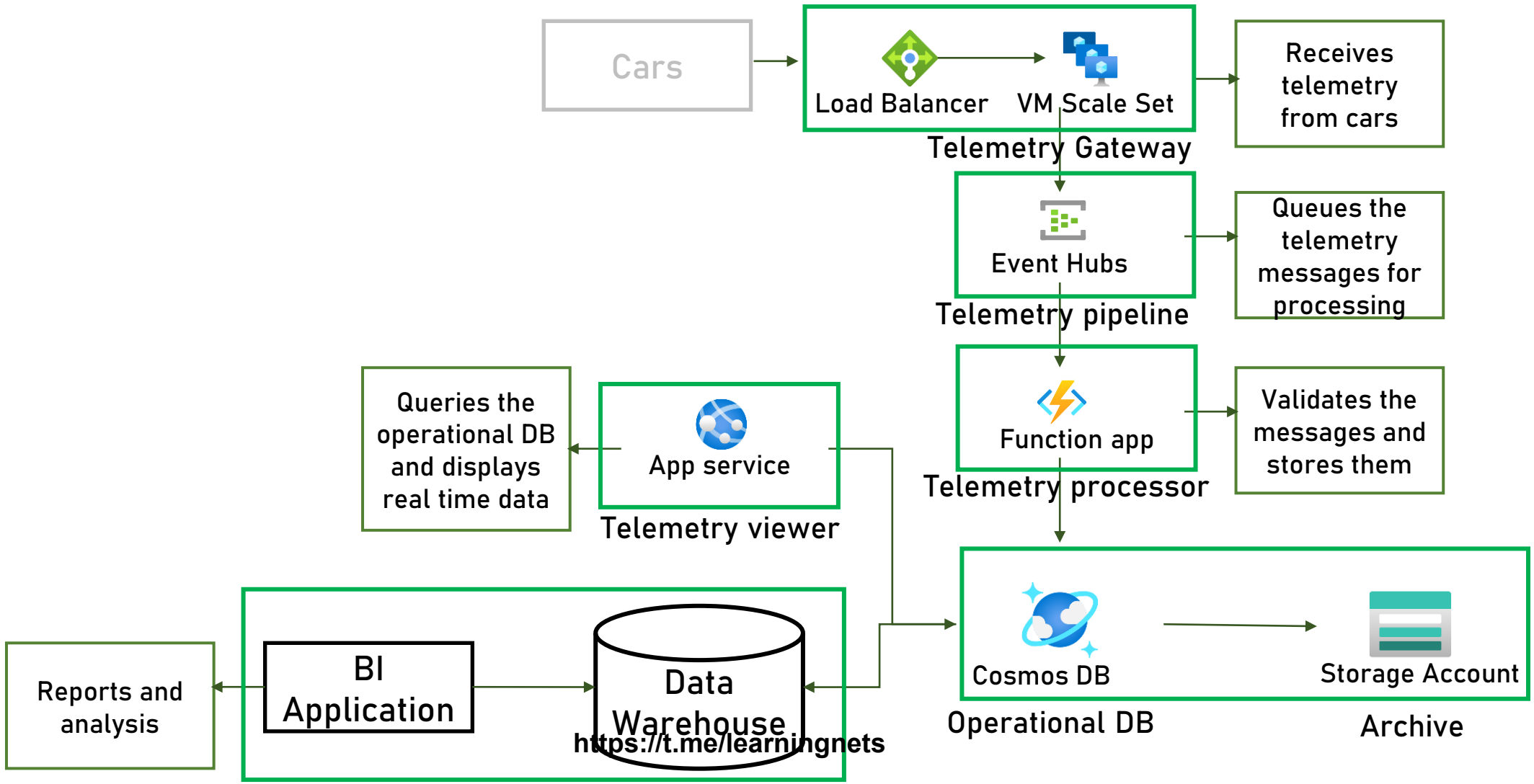
Security

- To-Do:
  - Block access to databases from unauthorized IP addresses

Security

- What about the App Service?
  - The client decided not to place WAF in front the App Service
    - Small service
    - Read-only operations
    - Save costs

Architecture Diagram



Muscar.auto

Cost

Estimated upfront cost	$0.00
Estimated monthly cost	$1,835.82

Download detailed cost estimation from the lecture's resources