Whiteboard Notes

Materials

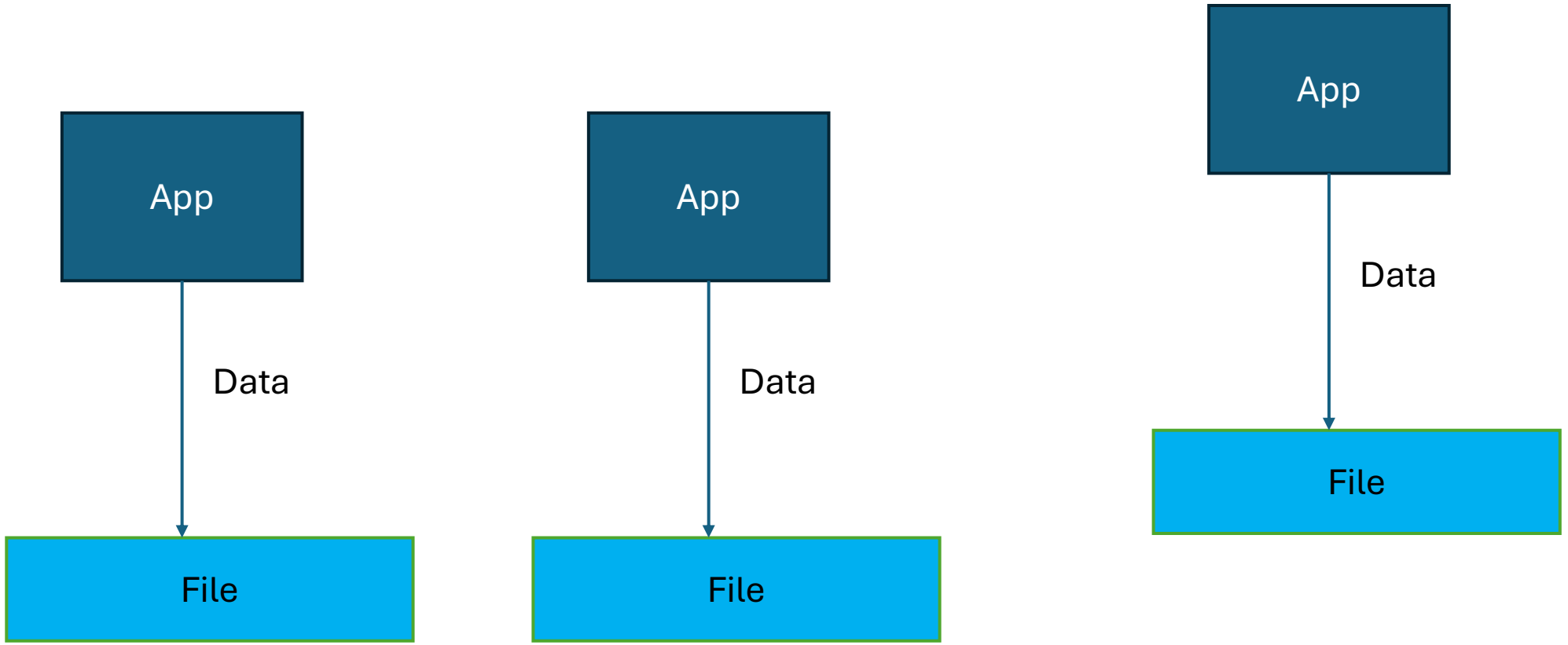
- https://github.com/nagabhushan1/eb

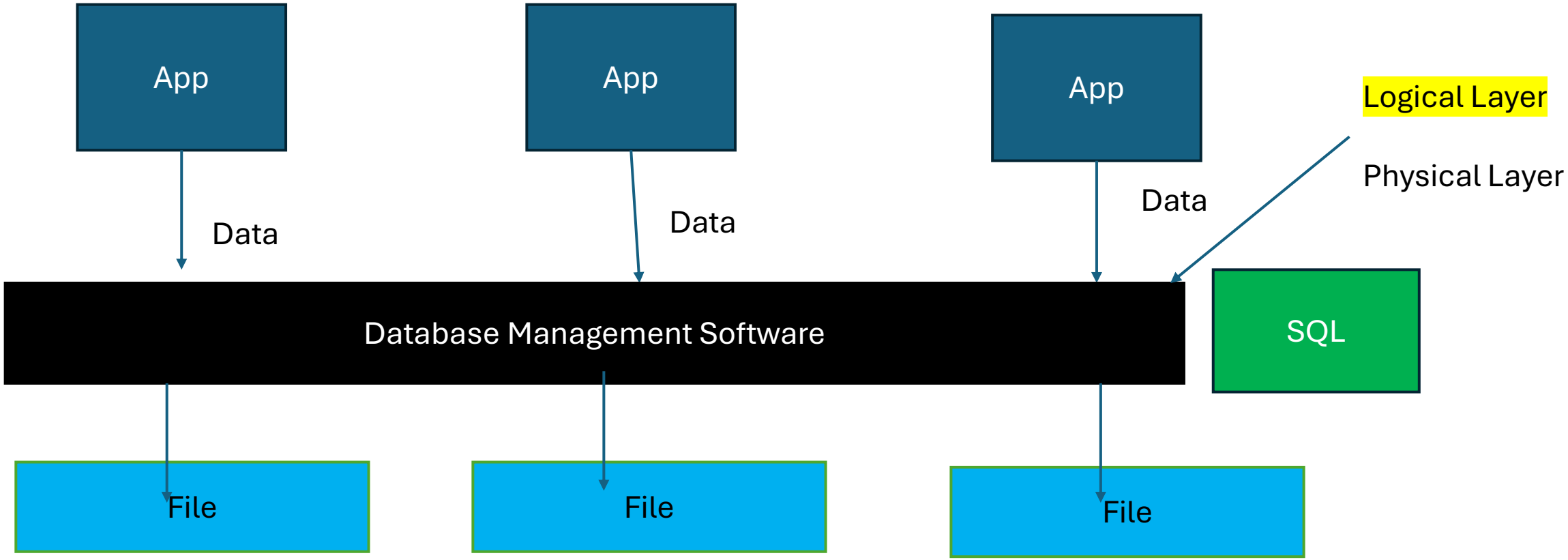
- https://codeshare.io/amLyeW

Data

- Data is the most important part of the application.
- Database is the common denominator for any application – No matter which language that was built with
- What is centric to any application is data.
- All data needs to be stored - from day 1
- Developer needs to honor 2 principles
  - Data Independence – Data needs to be independent of the application which created it
  - Data Persistence – Data should outlive the process / application which generated it

Good old days



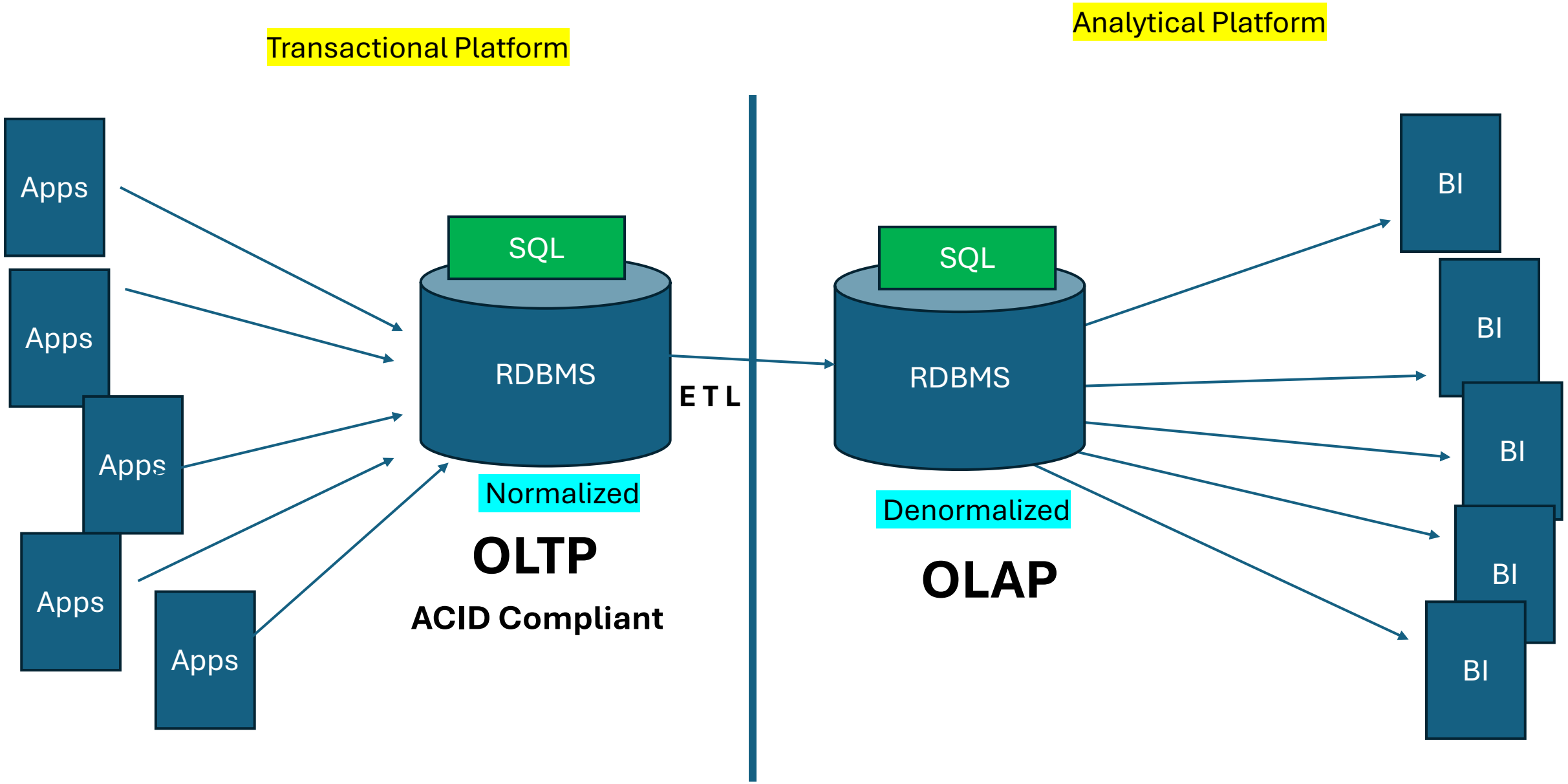


Oracle / MySQL / DB2 / Postgres / MSSQL / SQL Lite / Cloud SQL / RDS

3 main components in a Relational Data Model

- Collection of database objects (Tables, Views, Index, Procedures)
- Set of operators
- Set of integrity rules

Data Engineering World (Traditional)



Operators

- Operators are keywords / special symbols that will help us in performing operations on data. We can compare, combine and manipulate data using these operators
  - Arithmetic Operators
  - Relational Operators (Comparison Operators)
  - Logical Operators
  - Special Operators

Datatypes – Important points

- Infinity – Special constant for floating point numbers (binary float, binary double), represents values that are mathematically infinite (divide by zero)

- Nan – Not a Number → Not equal to anything → represents undefined or invalid mathematical operation

- Null → Absence of value (Unknown value) → Anything which is NULL is unknown

- Zero → A definite numerical value (It means nothing in arithmetic, however it is still a real number stored in a database)

```
SQL> select 5 + 0 as with_zero, 5 + NULL as with_null from dual;

 WITH_ZERO  WITH_NULL
---------- ----------
         5
```

```
SQL> select cast(0 as binary_double) / cast(0 as binary_double) from dual ;

CAST(0ASBINARY_DOUBLE)/CAST(0ASBINARY_DOUBLE)
---------------------------------------------
                                          Nan
```

```
SQL> select cast(1 as binary_double) / cast(0 as binary_double) from dual ;

CAST(1ASBINARY_DOUBLE)/CAST(0ASBINARY_DOUBLE)
---------------------------------------------
                                          Inf
```

Default date format in Oracle

DD-MON-YY

```
SQL> select sysdate from dual;

SYSDATE
---------
15-SEP-25

SQL>
```

Data Integrity

- Clean, correct and consistent data !
- Enforced using Constraints
- Constraints are used to prevent invalid data from being entered into your tables. Constraints are enforced on table columns!
  - Not Null
  - Unique
  - Primary Key
  - Foreign Key
  - Check

SQL Sub Languages (Sections)

- If a user has to perform operations on data, they use SQL.
- SQL has sub sections
  - DDL – Data Definition Language
    - CREATE, ALTER, DROP, TRUNCATE, RENAME (> ORACLE 9i)
  - DML – Data Manipulation Language
    - INSERT, UPDATE, DELETE, MERGE (ORACLE)
  - DQL – Data Query Language / DRL – Data Retrieval Language
    - **SELECT**
  - DCL – Data Control Language
    - GRANT, REVOKE
  - TCL – Transaction Control Language
    - COMMIT, ROLLBACK, SAVEPOINT

All DDLs are auto committed

Oracle Functions

- Functions are used to perform a specific task
- 2 types of Functions
  - Inbuilt Functions (Predefined Functions)
  - User Defined Functions

- 4 types of Predefined Functions
  - Number Functions
  - Date Functions
  - String Functions
  - Aggregate Functions

Joins

- Joins are used to retrieve data from multiple tables.

- Types of Joins
  - Equi Join / Inner Join → Matching rows only
  - Self Join → Joining a table to itself
  - Non Equi Join → Nonmatching rows
    - Left Outer Join → All rows from left table + matching rows
    - Right Outer Join → All rows from right table + matching rows
    - Full Outer Join → → All rows from left table + All rows from right table + matching rows

- *Note: In Oracle, we can also retrieve data from multiple tables without using join condition*

Subquery

- Single Row Subquery → Child query returns single value
- Multiple Row Subquery → Child query returns multiple values
  - We use “IN”, “ANY”, “ALL” operators in multiple row subqueries

Views

- View is a database object which is a virtual table and doesn’t store any data.
    - 2 types
        - Simple View
        - Complex View → Created by using multiple base tables
    - Read-Only Views
    - Materialized Views

Set Operators

- Also called Vertical Joins
  - UNION
  - UNION ALL
  - INTERSECT
  - MINUS

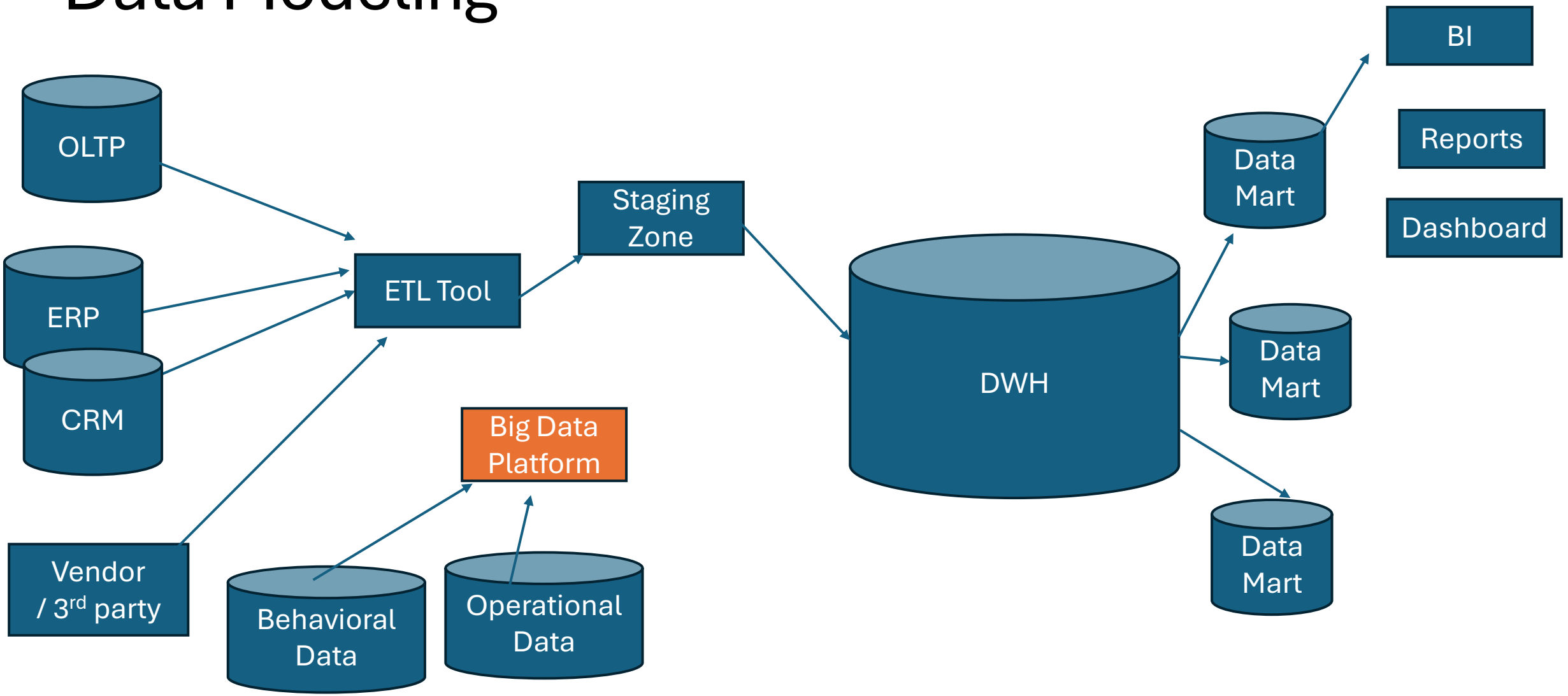
Datawarehouse (DWH)

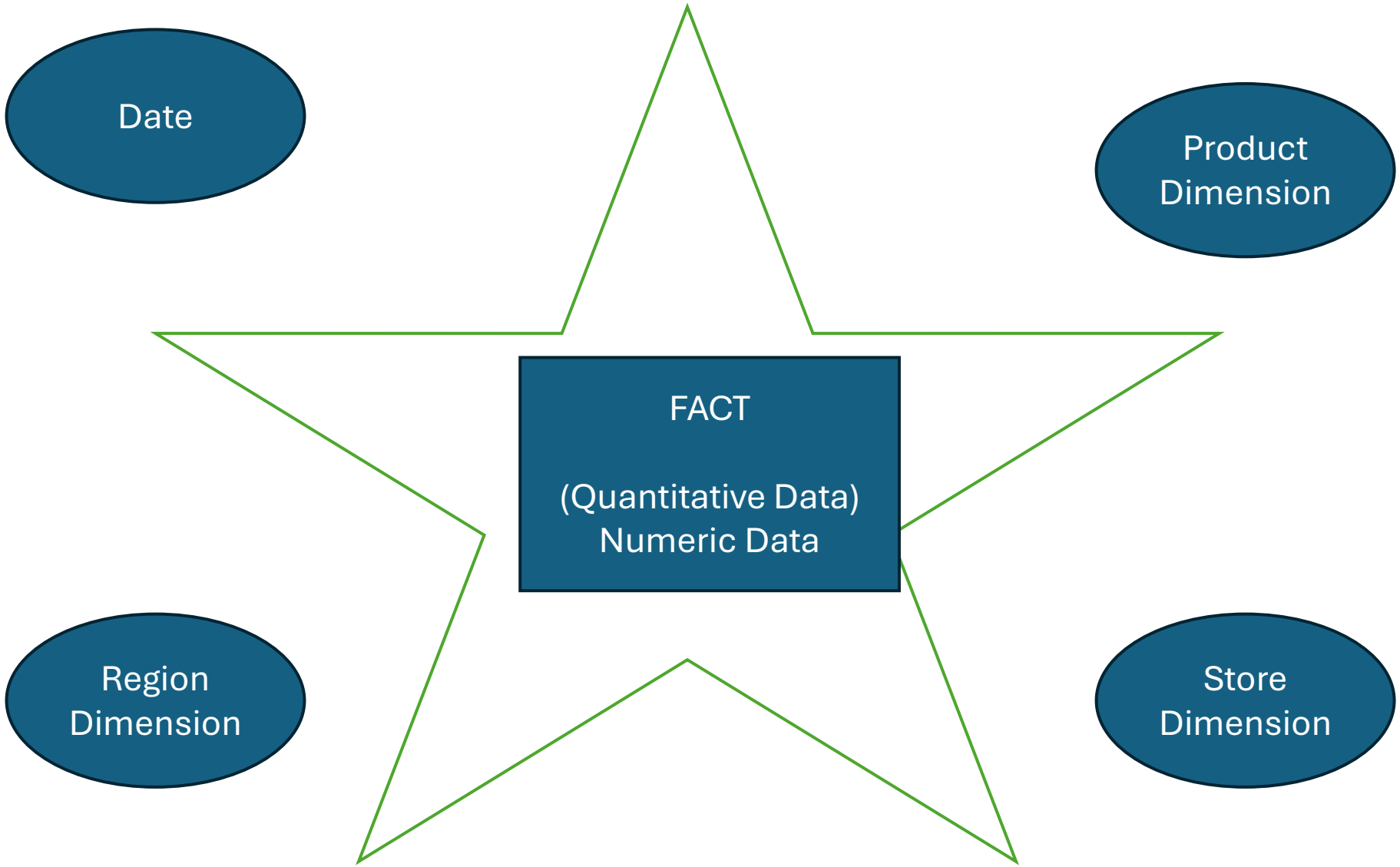
- Centralized repository designed to store and process large volume of data for analysis and decision making

- Database (OLTP) – Platform to store operational data
- Datawarehouse (OLAP) – Data analysis

- Applications (Web app, Mobile apps) generate data (ex. Online purchases, bank transactions). DWH do not generate transactional data. Instead, they collect and consolidate data from various transactional systems and operational systems

Datawarehouse Data Modeling




Date
Product Dimension
FACT
(Quantitative Data) Numeric Data
Region Dimension
Store Dimension

Python

Introduction to Python

- High level programming language
- Object Oriented
- General Purpose
- Interpreted

Python Evolution

- Python was developed in 1989 by ***Guido Van Rossam*** at National Research Institute (Netherlands)
- Officially made available in 1991 (Official DOB: 20-Feb-1991)

- The Complete Monty Python Circus – TV Show
  - Python V1 → 1994
  - Python V2 → 2000
  - Python V3 → 2008
  - ***Note: Python V3 won't provide backward compatibility***

Features

- Simple and Easy
- Open Source
- High Level
- Portable
- Dynamically Typed
- Functional Programming + Object Oriented Programming + Scripting
- Interpreted
- Extensible
- Extensive Libraries (Data Analysis, ML, AI..)

Identifiers

- Any name in a program – class name, function name, module name, variable name is called an Identifier
- Rules
  - Alphabets (lower case / upper case), digits (0-9) and underscore (_)
  - Identifier should not start with a digit
  - Identifiers are case sensitive
  - Keywords should not be used as identifiers

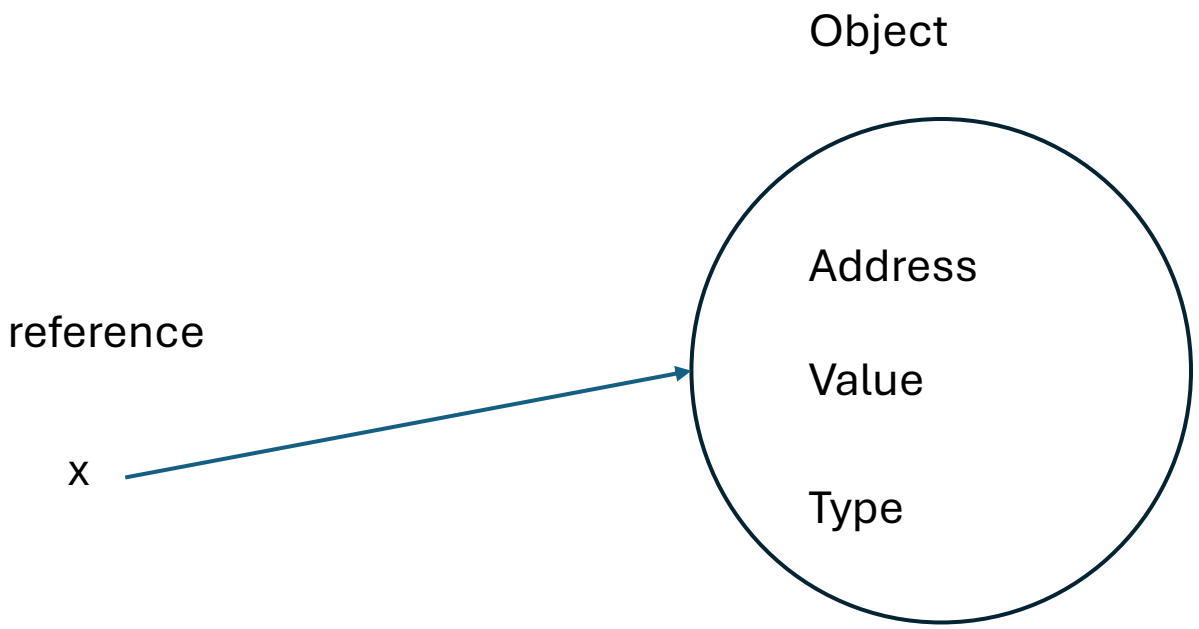
Reserved Keywords

- Some words are reserved by Python; they represent some functionality.

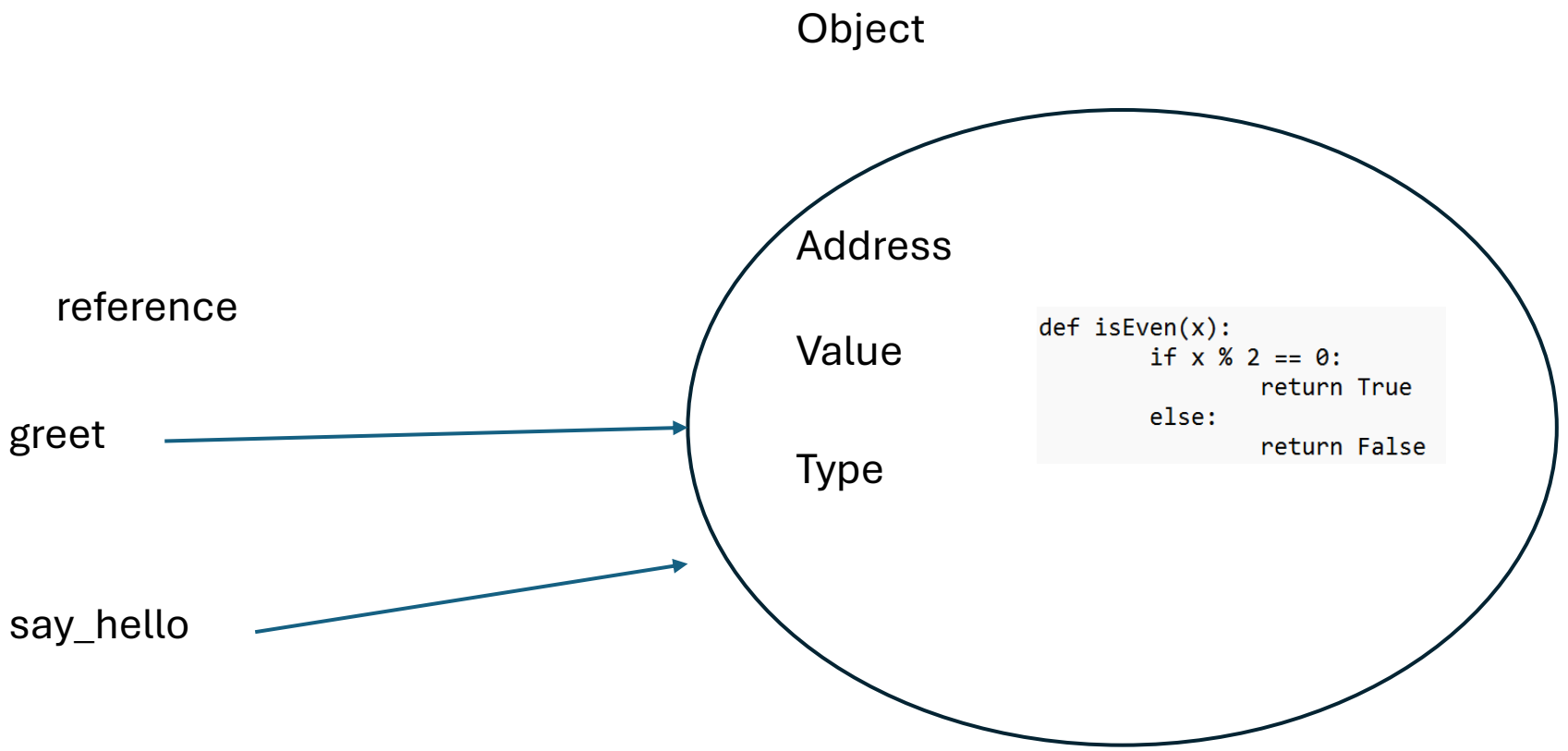
- 35 reserved keywords (Python 3.11.x)

- All keywords are alphabets
- Except the following 3 keywords, all keywords contain lower case alphabets
  - True
  - False
  - None

Everything is an object in Python



Function also is an object in Python



OOP

- Everything is an object in Python
- To create objects, we need a blue-print (model) / plan → Class
- We can create class to represent
  - Attributes (properties)
  - Behavior (actions)

  - Attributes are represented as variables
  - Actions are represented as methods

  - Noun → Classes
  - Adjectives → Attributes
  - Verbs → Methods

Cloud Computing

IT Services Company

Development Team

Operations / Ops Team

Data Scientist

Testing / QA Teams

SRE / Infra

Data Engineering

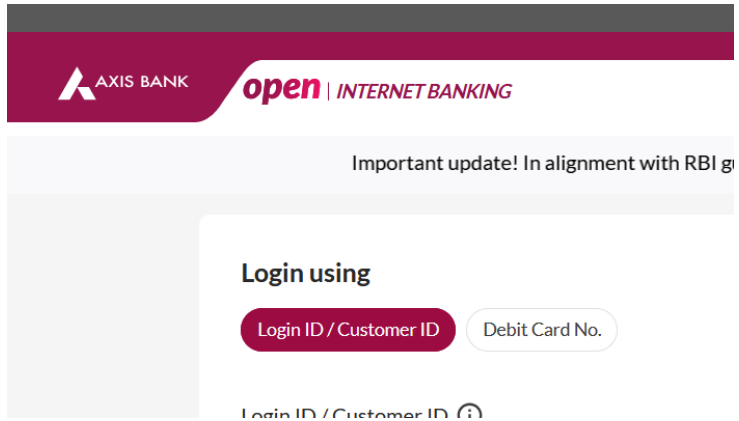
Admin

Data Analyst

BI

Non-IT

Bank (Customer / Client)



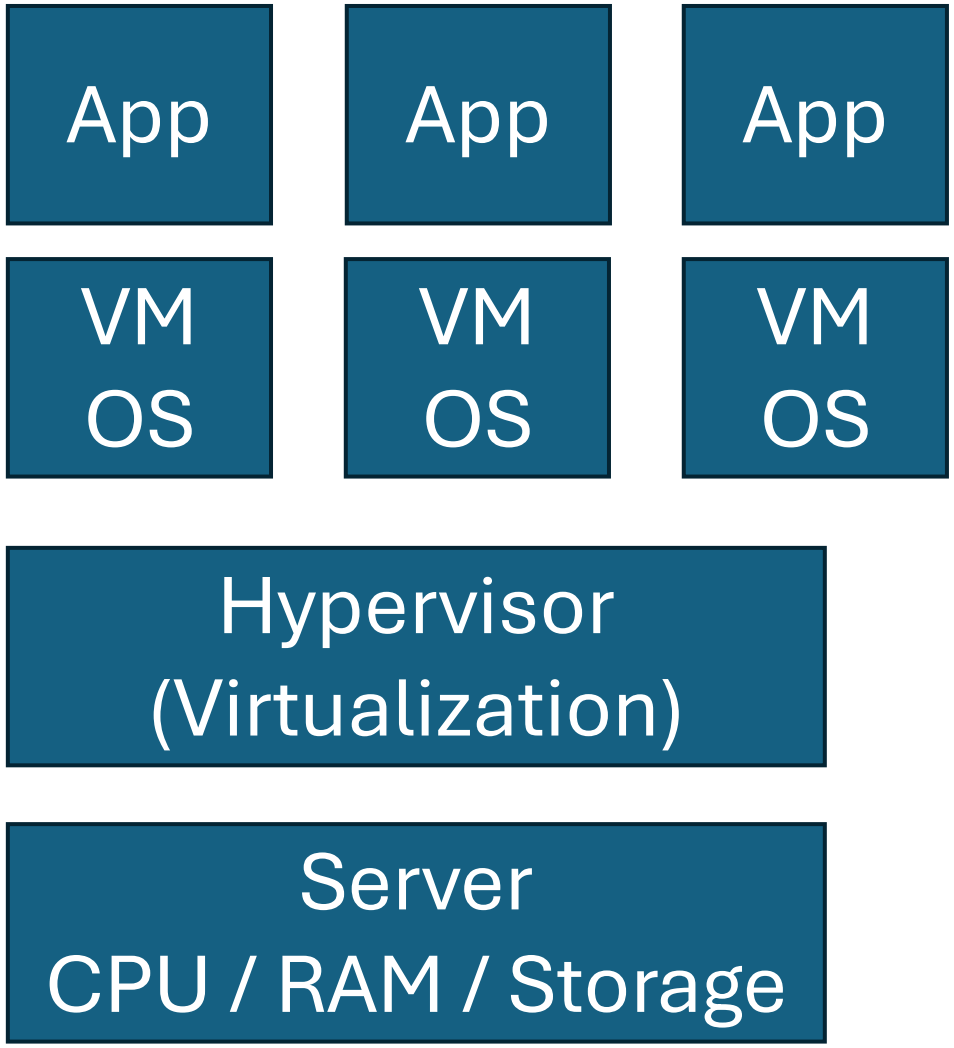
Infrastructure Requirements

- Physical Servers (Upfront Cost to procure the hardware)
- Maintain Data Centre
  - Networking
  - Firewall
  - Security
- Install OS → Linux / Windows Server
- Monitoring – App, Server

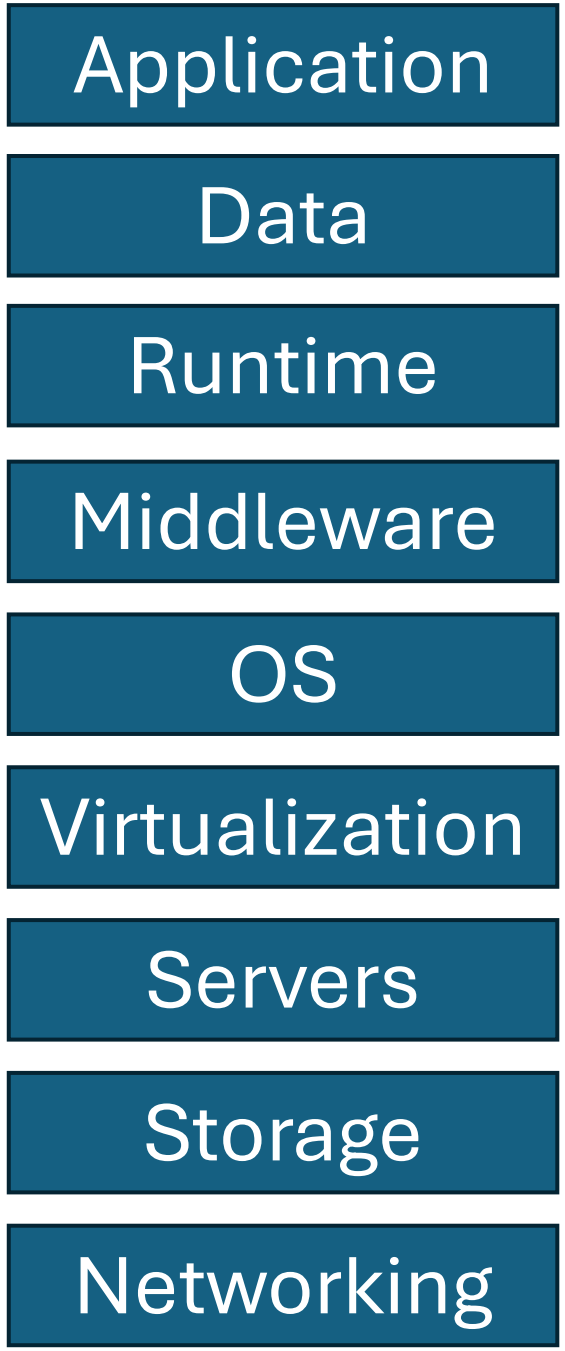
Infrastructure Requirements

- Servers
  - https://www.racksolutions.com/news/data-center-optimization/blade-server-vs-rack-server/?srsltid=AfmBOorUp2-n8a2IjBx2KCPxEFUU6EPPdkgwmng2rdm9Z-7IYCaYrUUf

Virtualization



Bare Metal
Hosted

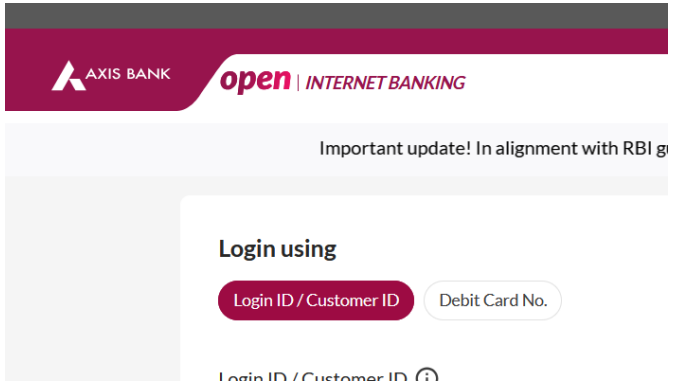


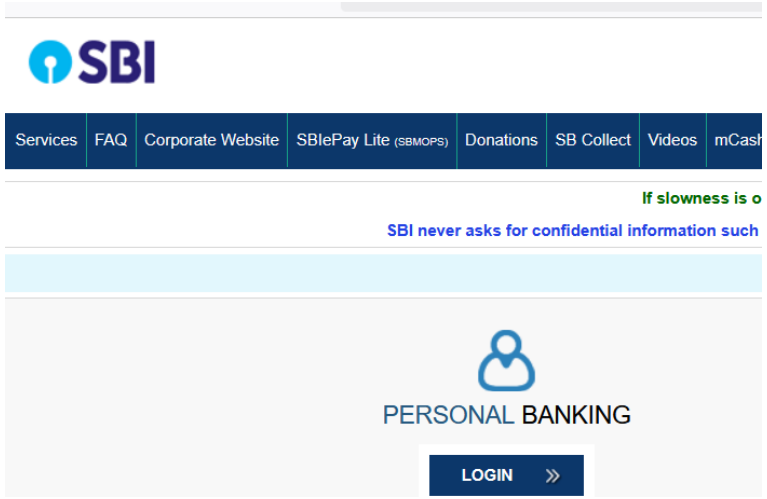
On-Premise Data Centre

If I decide to manage all this !

- Data Centre
- Manage all of these

- Capital Expenditure (Capex)
- Operational Expenditure (Opex)


AXIS BANK
open | INTERNET BANKING
Important update! In alignment with RBI g
Login using
Login ID / Customer ID
Debit Card No.


SBI
Services
FAQ
Corporate Website
SBIePay Lite (SBMOPS)
Donations
SB Collect
Videos
mCash
If slowness is o
SBI never asks for confidential information such
PERSONAL BANKING
LOGIN

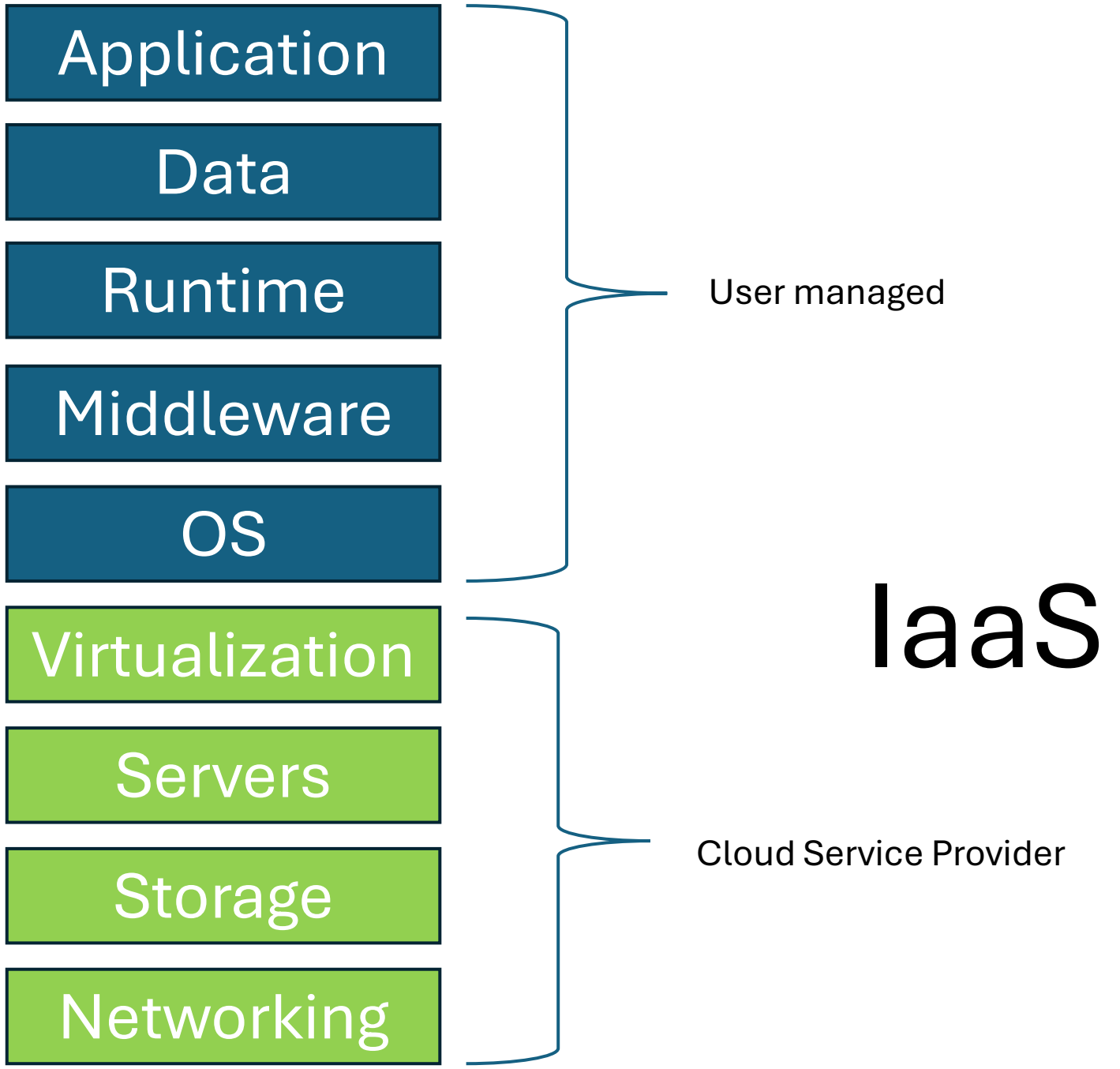
Internet


www

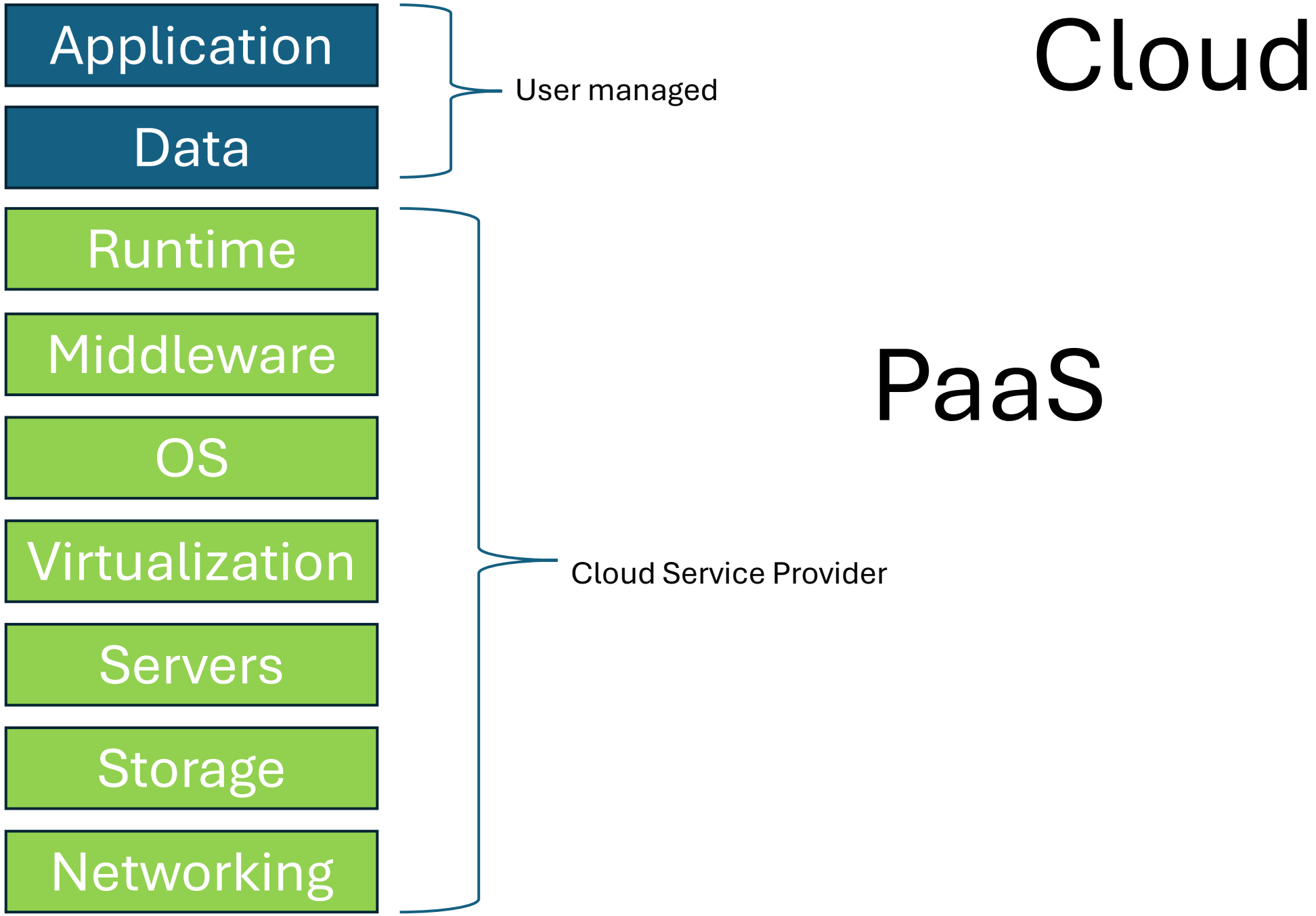
Cloud Service Providers

- AWS (Amazon Web Services)
- GCP (Google Cloud Platform)
- Azure (Microsoft)
- Alibaba, Digital Ocean....

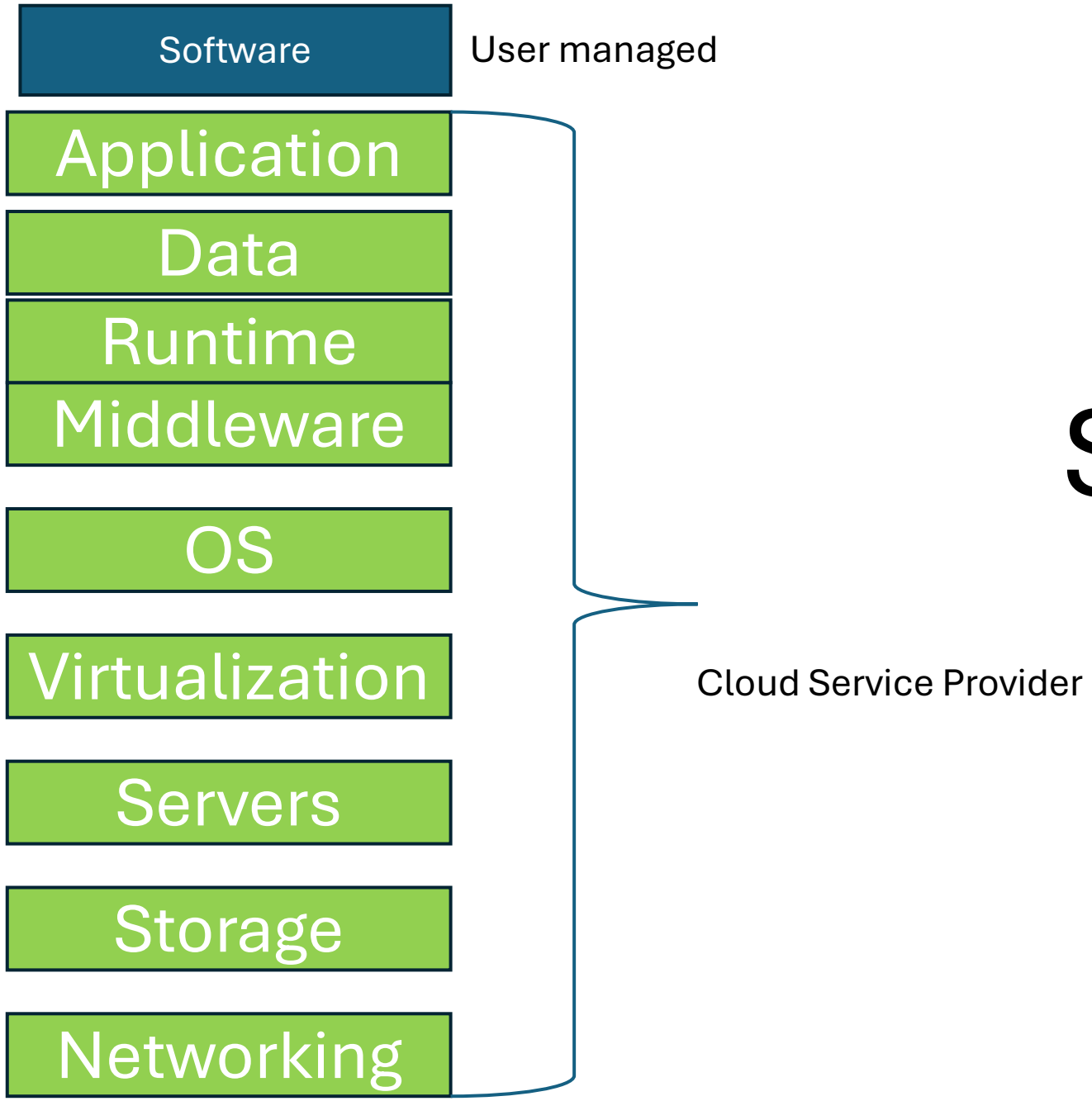
Cloud


Application
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking
User managed
Cloud Service Provider

IaaS


Application
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking
User managed
Cloud Service Provider
Cloud
PaaS

Cloud


Software
User managed
Application
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking
Cloud Service Provider

SaaS

IAM → Identity and Access Management

- IAM is a global service

- Root User Vs IAM User
  - Root User:
    - Whenever we create a new AWS account, a root user is setup
    - Root user has complete control on the account
    - Root user must be used only for initial setup, don’t use root user for daily work
  - IAM User:
    - Each IAM user represents one person in the organization / team
    - Users can be part of a group, so that it becomes easier to manage roles and policies

user01

adminAccess

AWS Acronyms

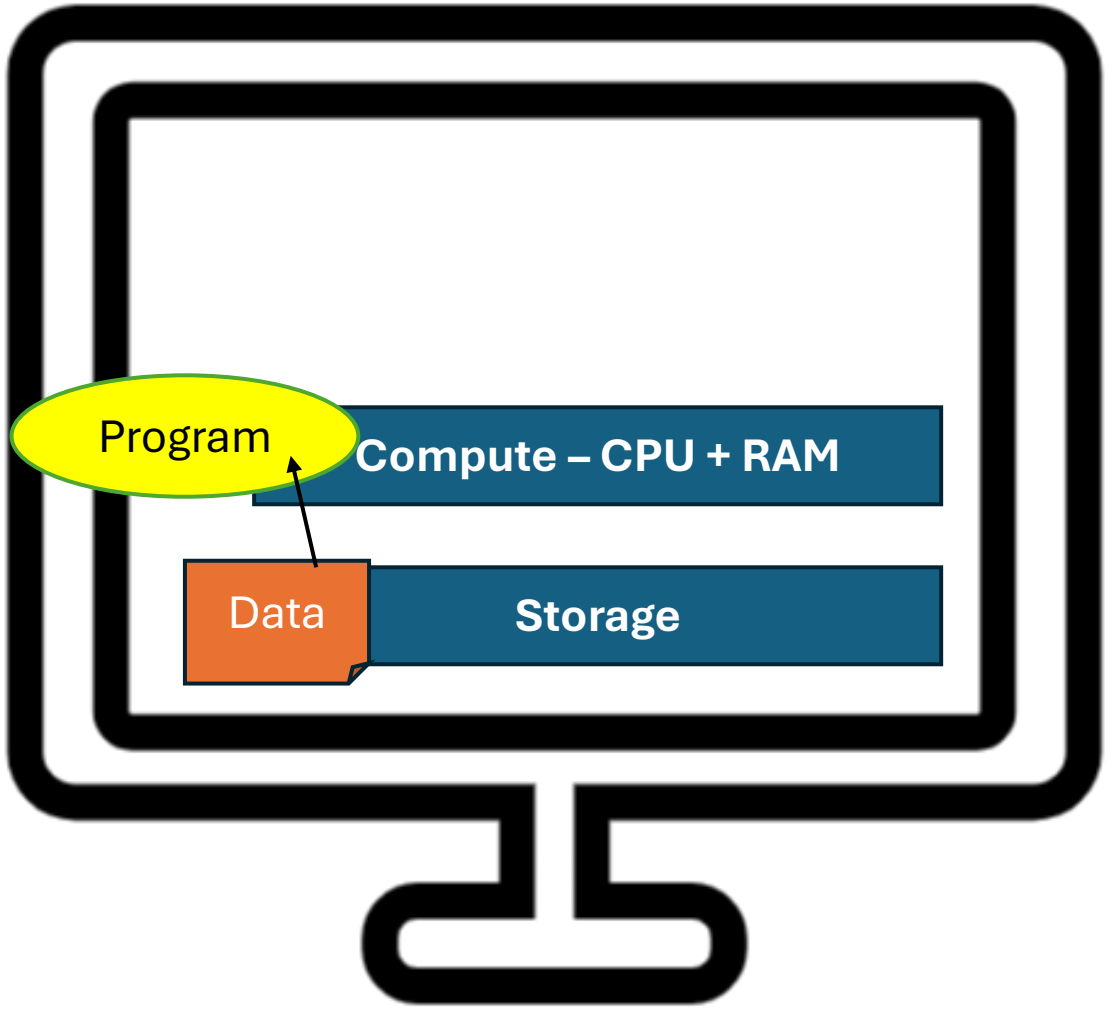
- AWS – Amazon Web Services
- EC2 – Elastic Compute Cloud
- S3 – Simple Storage Service
- IAM – Identity and Access Management

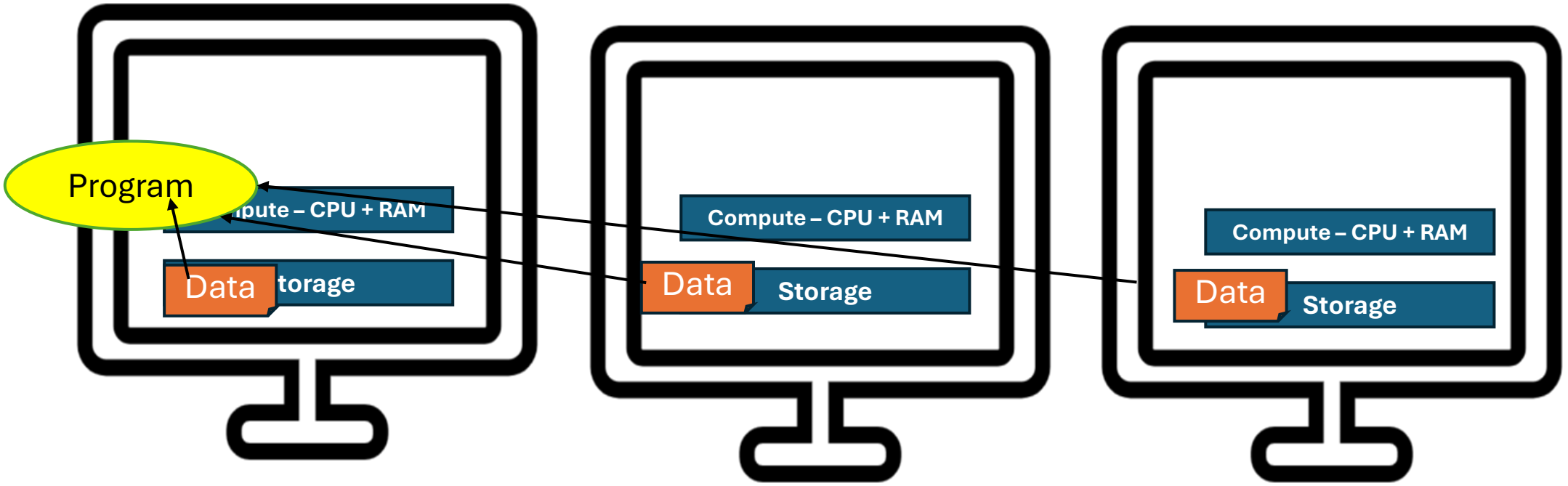
Spark Basics

Monolithic Computing

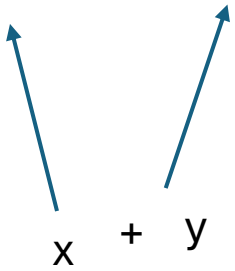
Big Data (Attributes)

- Volume
- Velocity
- Variety
- Veracity

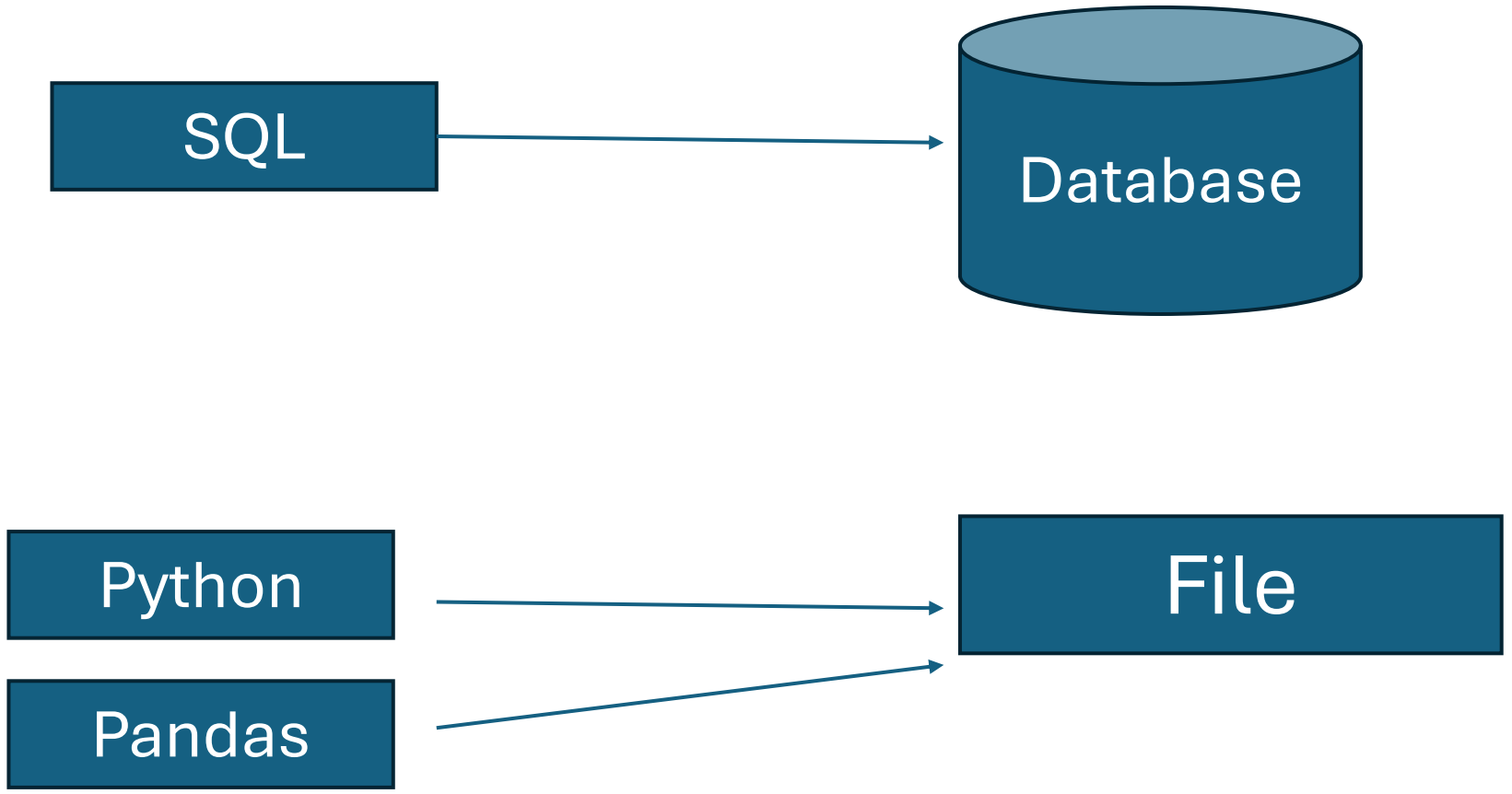



Program
Compute – CPU + RAM
Data
Storage
Compute – CPU + RAM
Data
Storage
Compute – CPU + RAM
Data
Storage

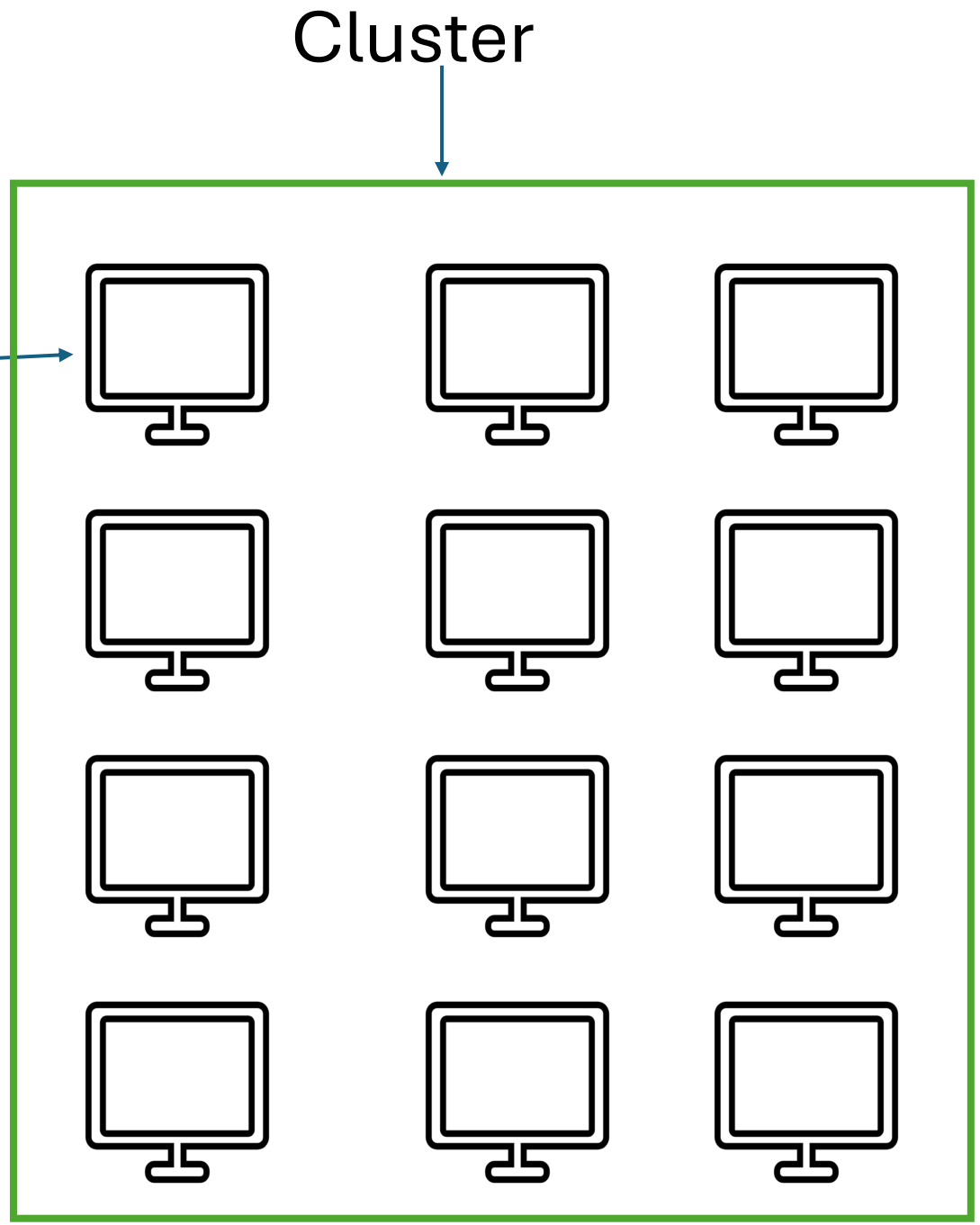
[10, 11, 22, 23, 34, 35, 40, 45, 60]



Traditional Systems (Monolithic Computing)



Distributed Systems



Node

Storage + Compute

Linear Scalability

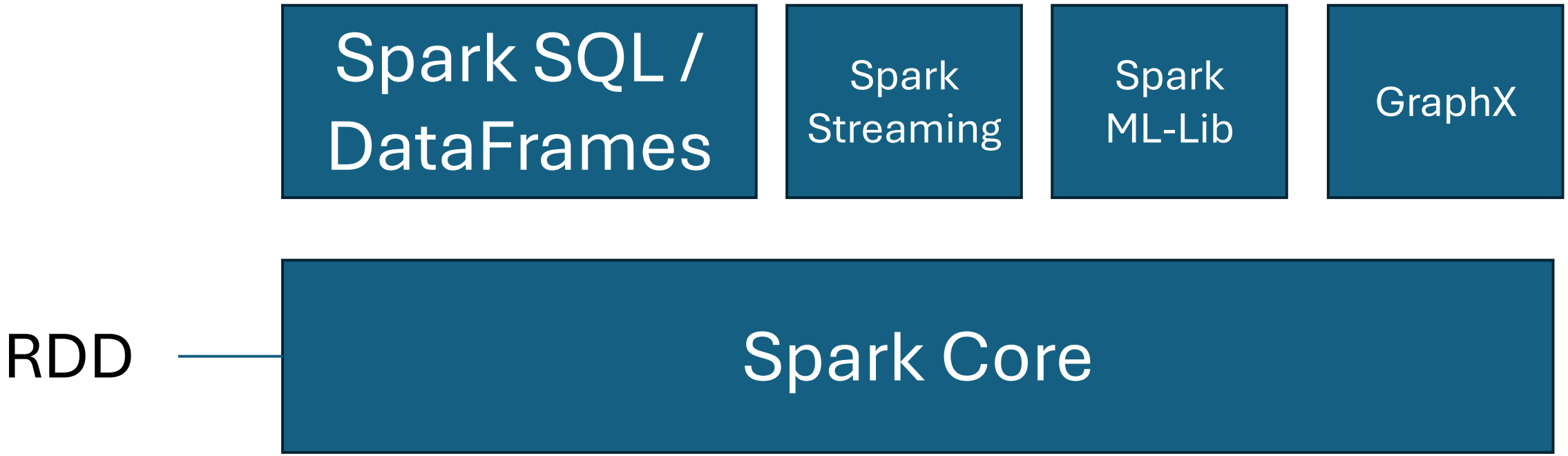
To get 2x storage, double the number of nodes

To get 2x compute, double the number of nodes

Apache Spark

- Spark is a unified engine for large scale data analytics
- Spark applications can be written in Scala, Python, Java and R
- Spark was built using Scala
- Spark is a distributed in-memory computation engine
- Spark was built to overcome limitations of Hadoop's MapReduce framework
- Spark core is RDD (Resilient Distributed Dataset) – primary programming abstractions in Spark
- RDDs in Spark are in-memory objects
- RDDs are immutable

Spark Components



Simple analogy

- 1TB Data processing → Traditional MySQL → Response ~60 mins

- Hadoop
  - 1TB Data processing → 10 node cluster → Response ~6 mins
  - 10x cheaper

- Spark (Faster than Hadoop)
  - 1TB Data processing → 10 node cluster → Response ~2 mins

- BigQuery (GCP)
  - 1TB Data processing → Few seconds

Spark RDD

https://spark.apache.org/docs/latest/rdd-programming-guide.html

- The main abstraction Spark provides is a *resilient distributed dataset* (RDD), which is a collection of elements partitioned across the nodes of the cluster that can be operated on in parallel.

- Spark revolves around the concept of a *resilient distributed dataset* (RDD), which is a fault-tolerant collection of elements that can be operated on in parallel.

Spark RDD

https://spark.apache.org/docs/latest/rdd-programming-guide.html

- There are two ways to create RDDs
  - *parallelizing* an existing collection
    - Parallelized collections are created by calling SparkContext's parallelize method on an existing iterable or collection
  - Referencing a dataset in an external storage system
    - PySpark can create distributed datasets from any storage source supported by Hadoop, including your **local file system**, HDFS, Cassandra, HBase, Amazon S3, etc.

https://spark.apache.org/docs/latest/rdd-programming-guide.html#rdd-operations

RDD Operations

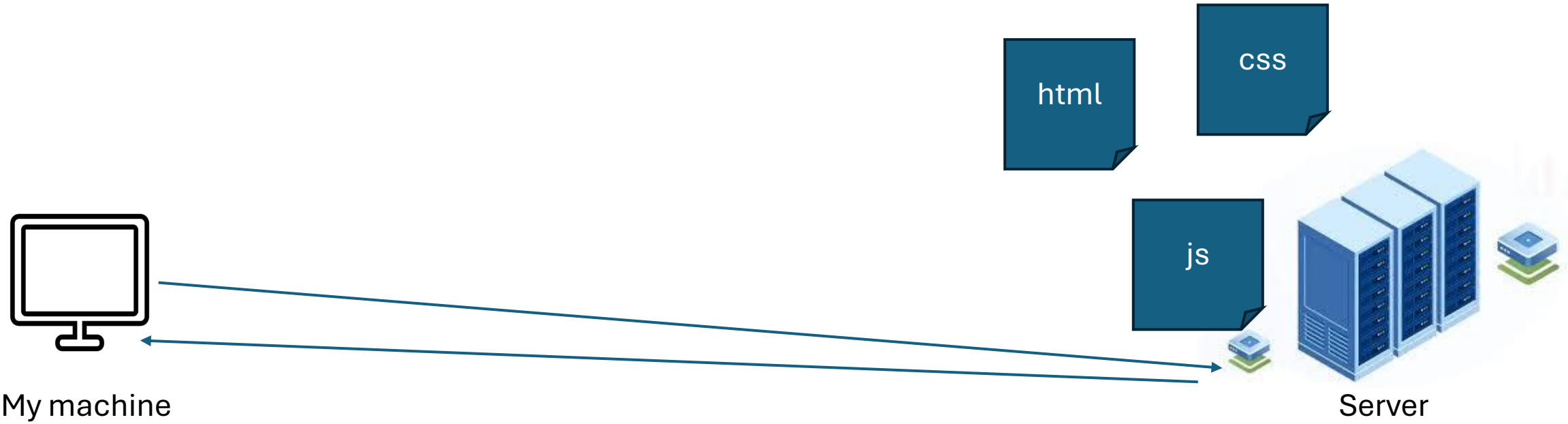
- RDDs support two types of operations
  - *Transformations*
    - create a new rdd from an existing one
    - For example, map is a transformation that passes each dataset element through a function and returns a new RDD representing the results
    - All transformations in Spark are *lazy*, in that they do not compute their results right away
  - *Actions*
    - return a value to the driver program after running a computation
    - reduce is an action that aggregates all the elements of the RDD using some function and returns the final result to the driver program

https://spark.apache.org/docs/latest/rdd-programming-guide.html#rdd-operations

Passing functions to Spark

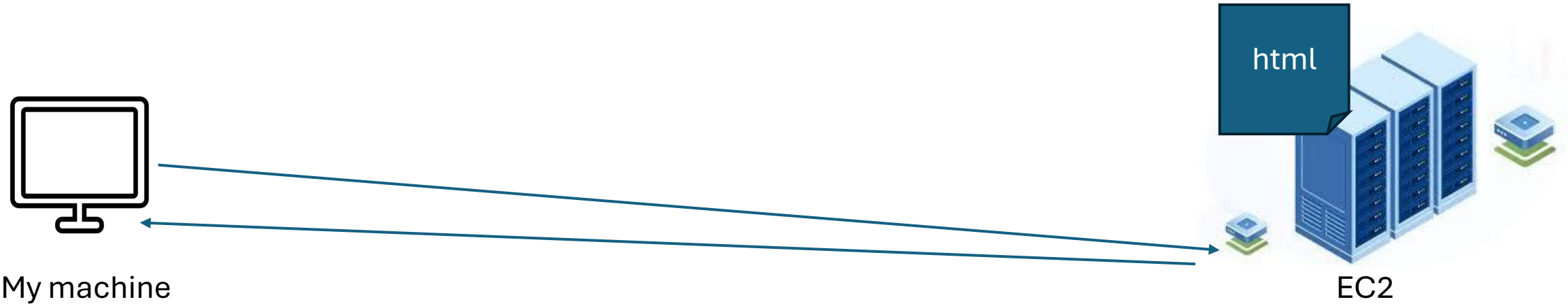
- Spark's API relies heavily on passing functions in the driver program to run on the cluster. There are three recommended ways to do this:
  - Lambda expressions, for simple functions that can be written as an expression. (Lambdas do not support multi-statement functions or statements that do not return a value.)
  - Local **defs** inside the function calling into Spark, for longer code.
  - Top-level functions in a module.

How do websites work?



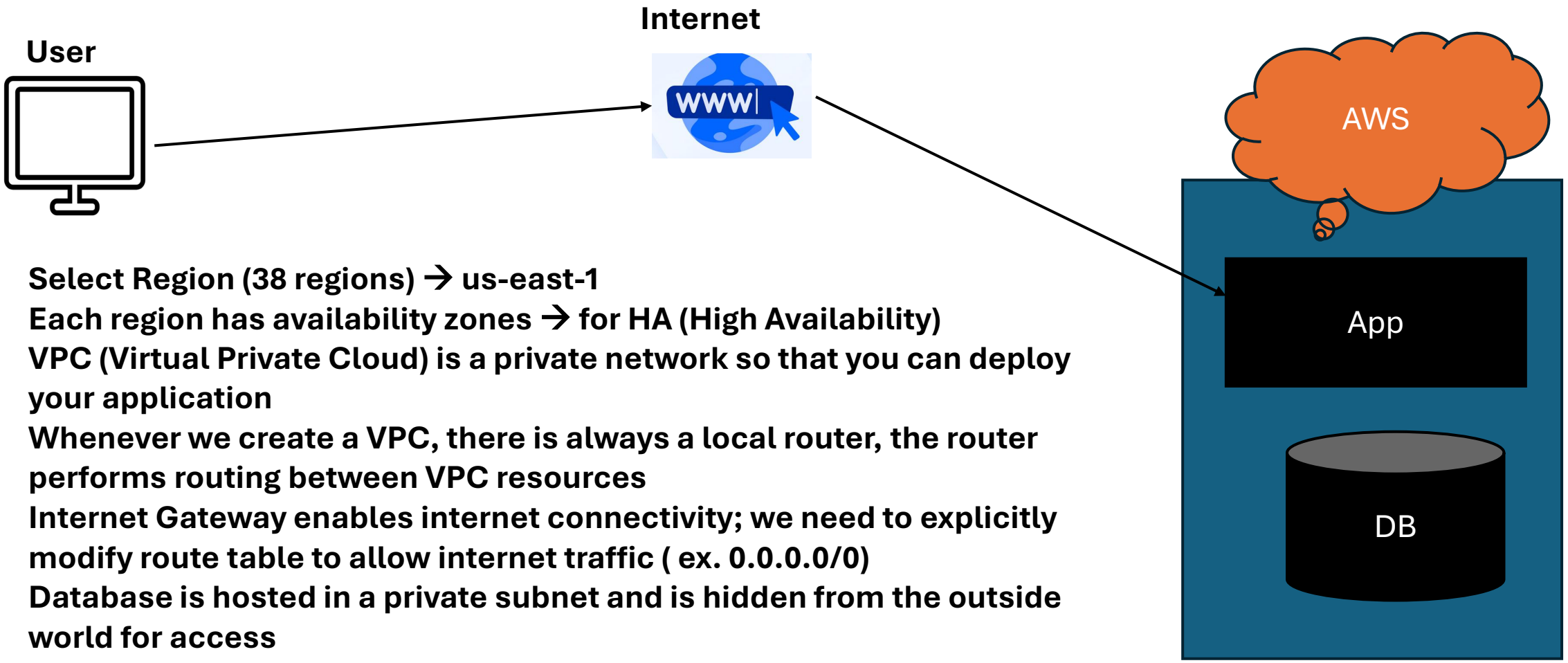
Public IP: 142.10.1.5

How do websites work?

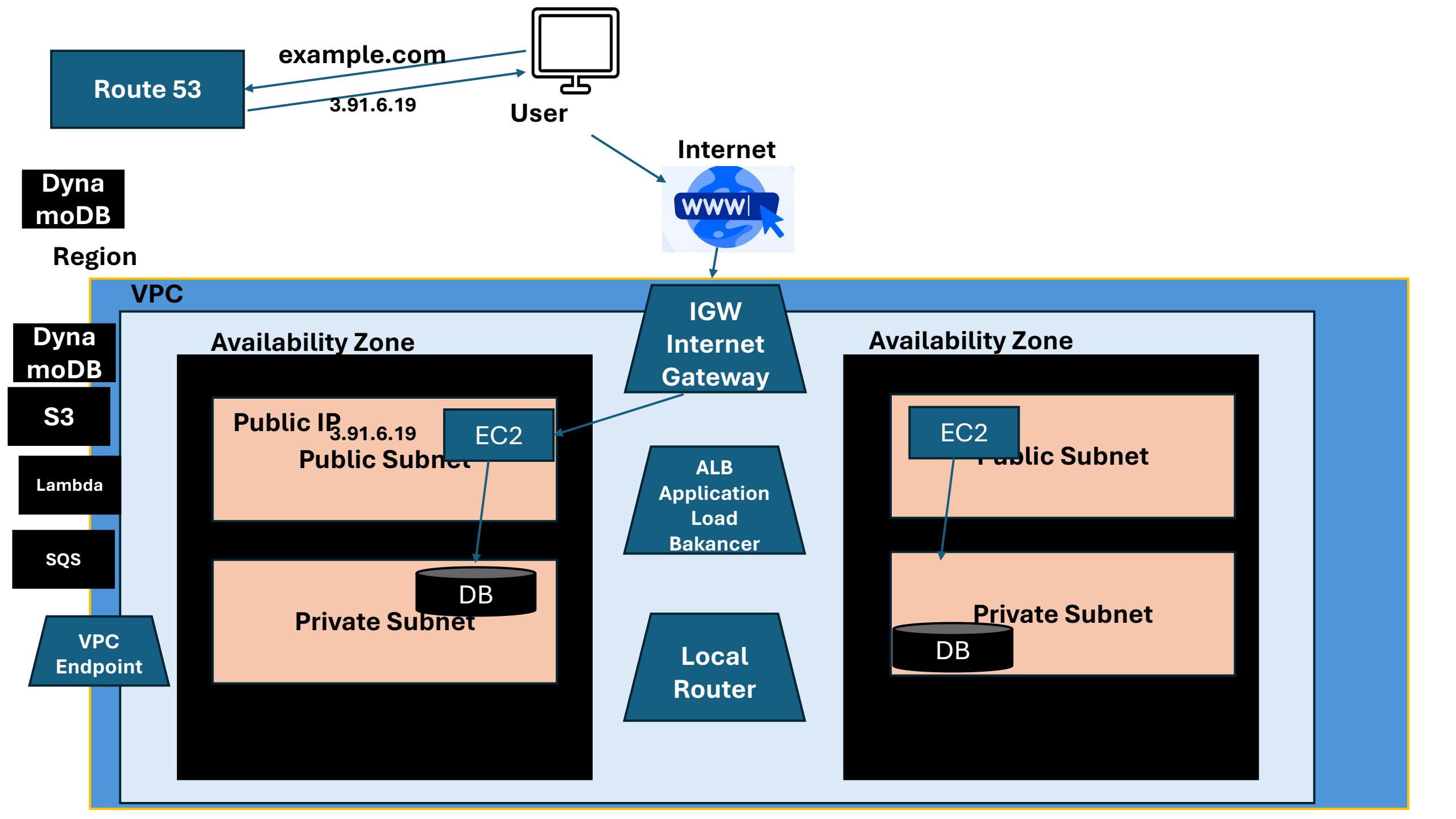


Public IP: 142.10.1.5

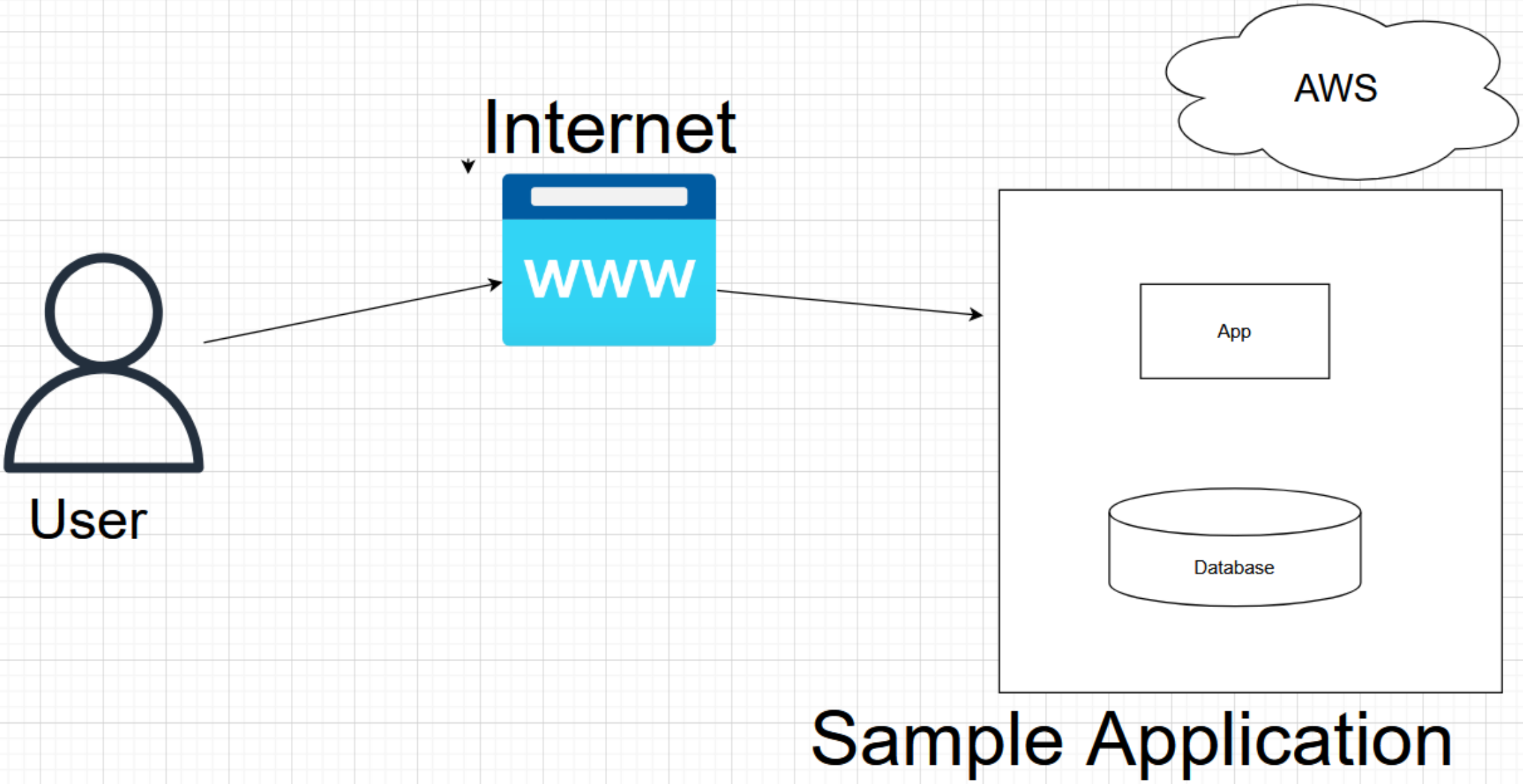
We want to deploy this application in AWS

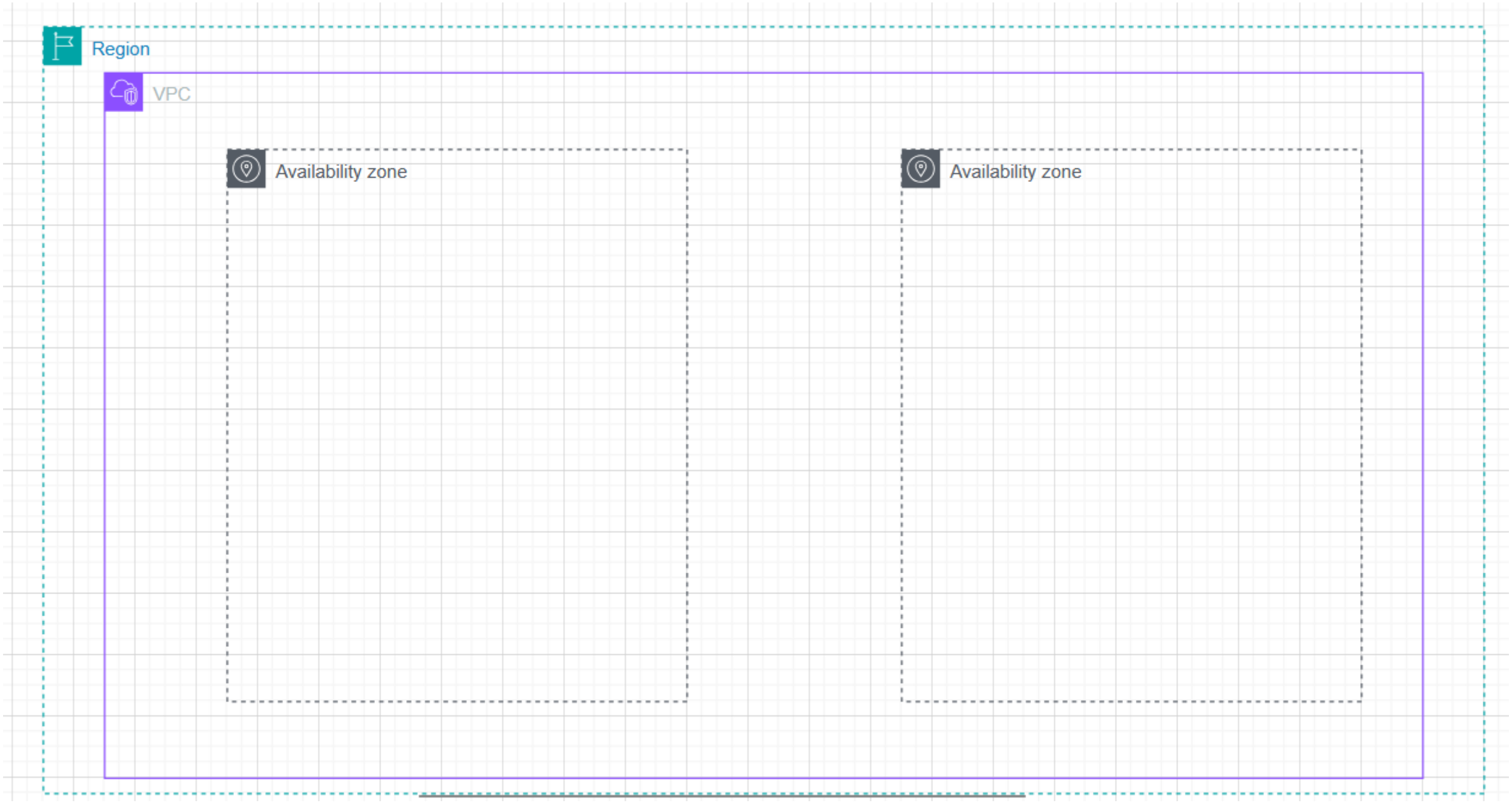


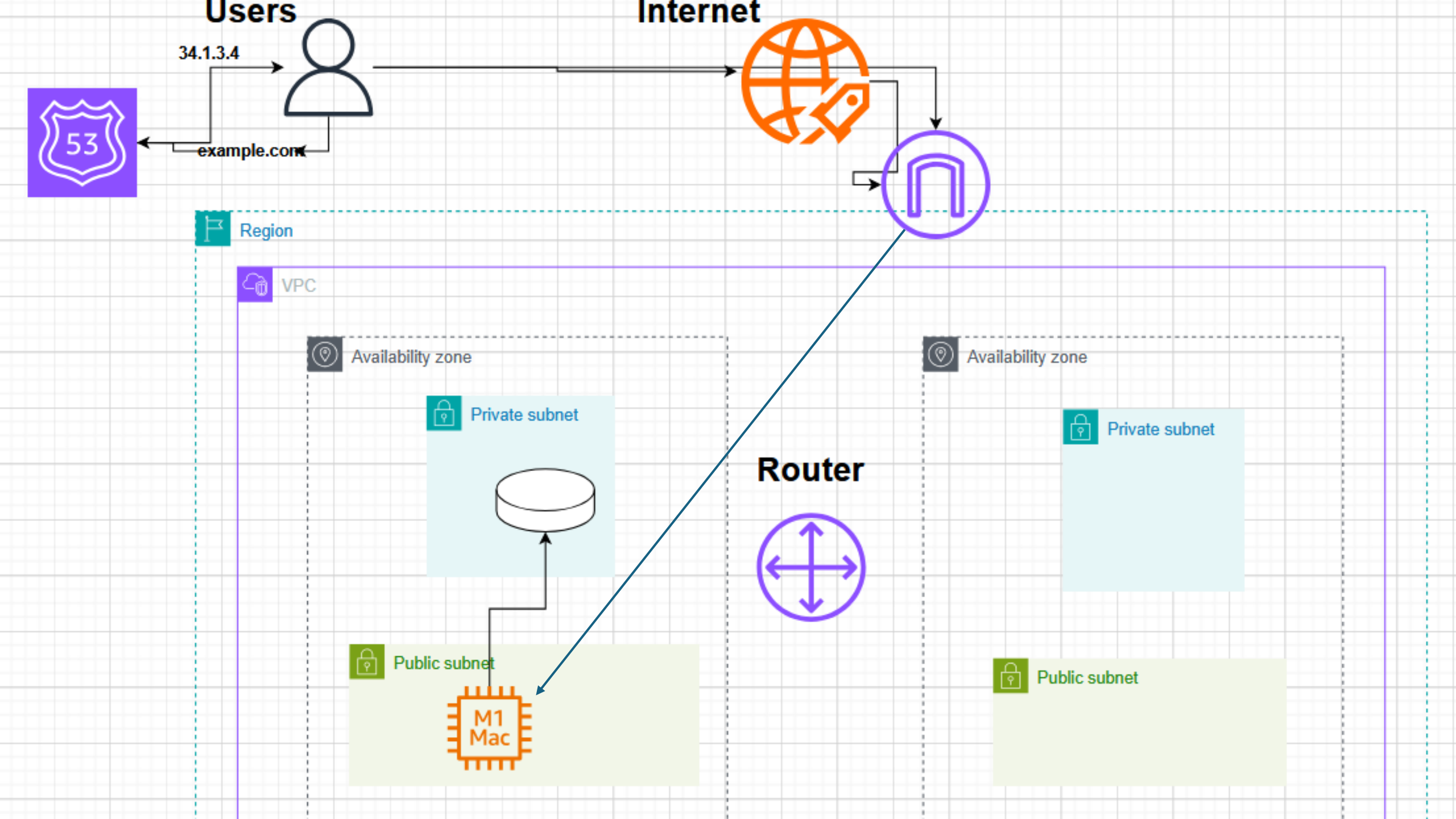
- Select Region (38 regions) → us-east-1
- Each region has availability zones → for HA (High Availability)
- VPC (Virtual Private Cloud) is a private network so that you can deploy your application
- Whenever we create a VPC, there is always a local router, the router performs routing between VPC resources
- Internet Gateway enables internet connectivity; we need to explicitly modify route table to allow internet traffic (ex. 0.0.0.0/0)
- Database is hosted in a private subnet and is hidden from the outside world for access
- ALB is balancing the load between different instances
- S3 buckets sit within AWS region, if the application server needs to access S3, use a VPC endpoint to communicate with S3


Route 53
example.com
3.91.6.19
User
Internet
Dyna moDB
Region
VPC
Dyna moDB
S3
Lambda
SQS
VPC Endpoint
Availability Zone
Public IP 3.91.6.19
EC2
Public Subnet
DB
Private Subnet
IGW Internet Gateway
ALB Application Load Bakancer
Local Router
Availability Zone
EC2
Public Subnet
Private Subnet
DB

We want to deploy this application in AWS




Region
VPC
Availability zone
Availability zone


Users
34.1.3.4
example.com
Internet
Region
VPC
Availability zone
Private subnet
Router
Public subnet
M1
Mac
Availability zone
Private subnet
Public subnet

IAM

- Identity and Access Management
- Allows you to control who can do what?

AWS Regions

https://aws.amazon.com/about-aws/global-infrastructure/regions_az/

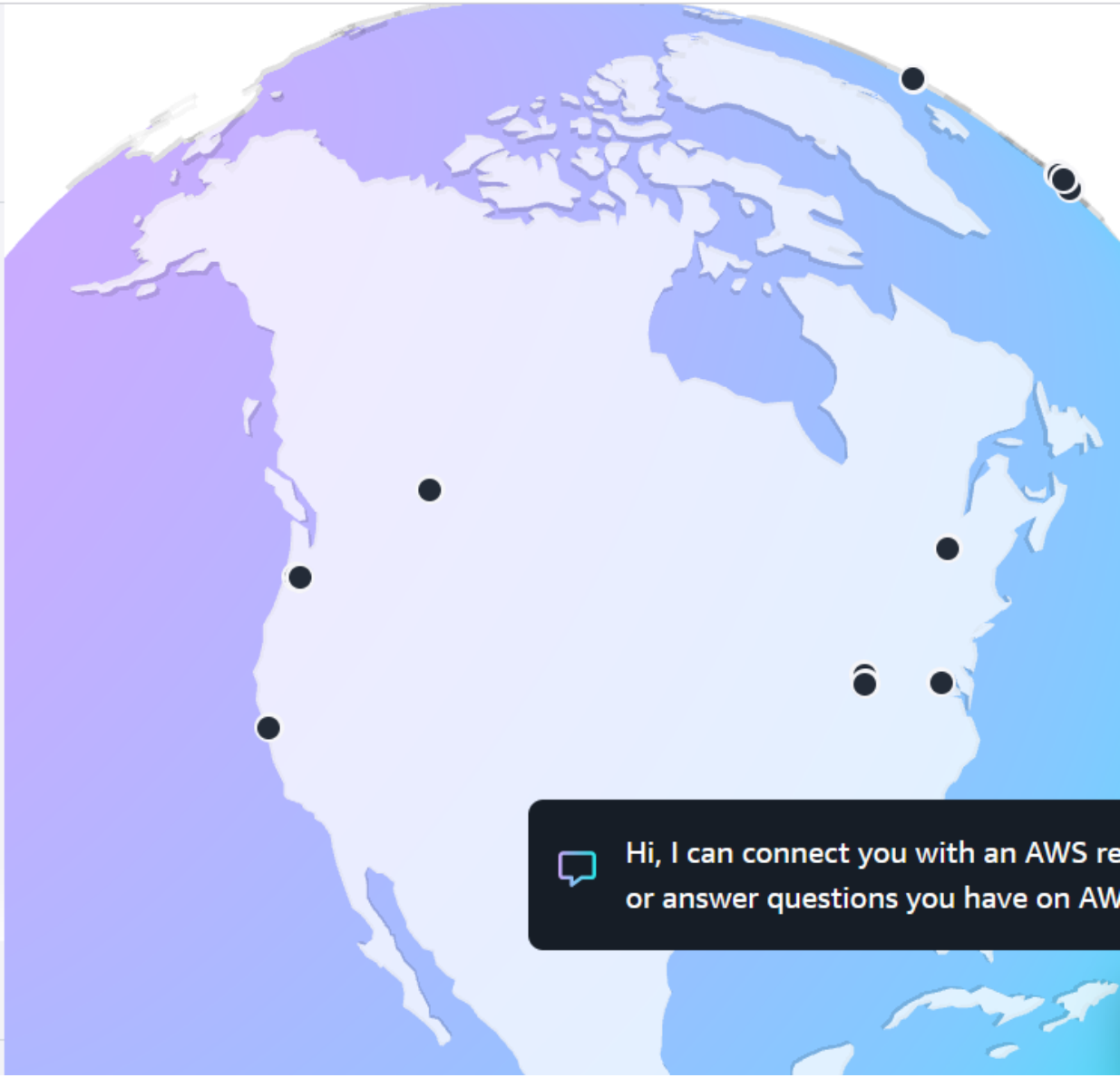
AWS Coverage Regions

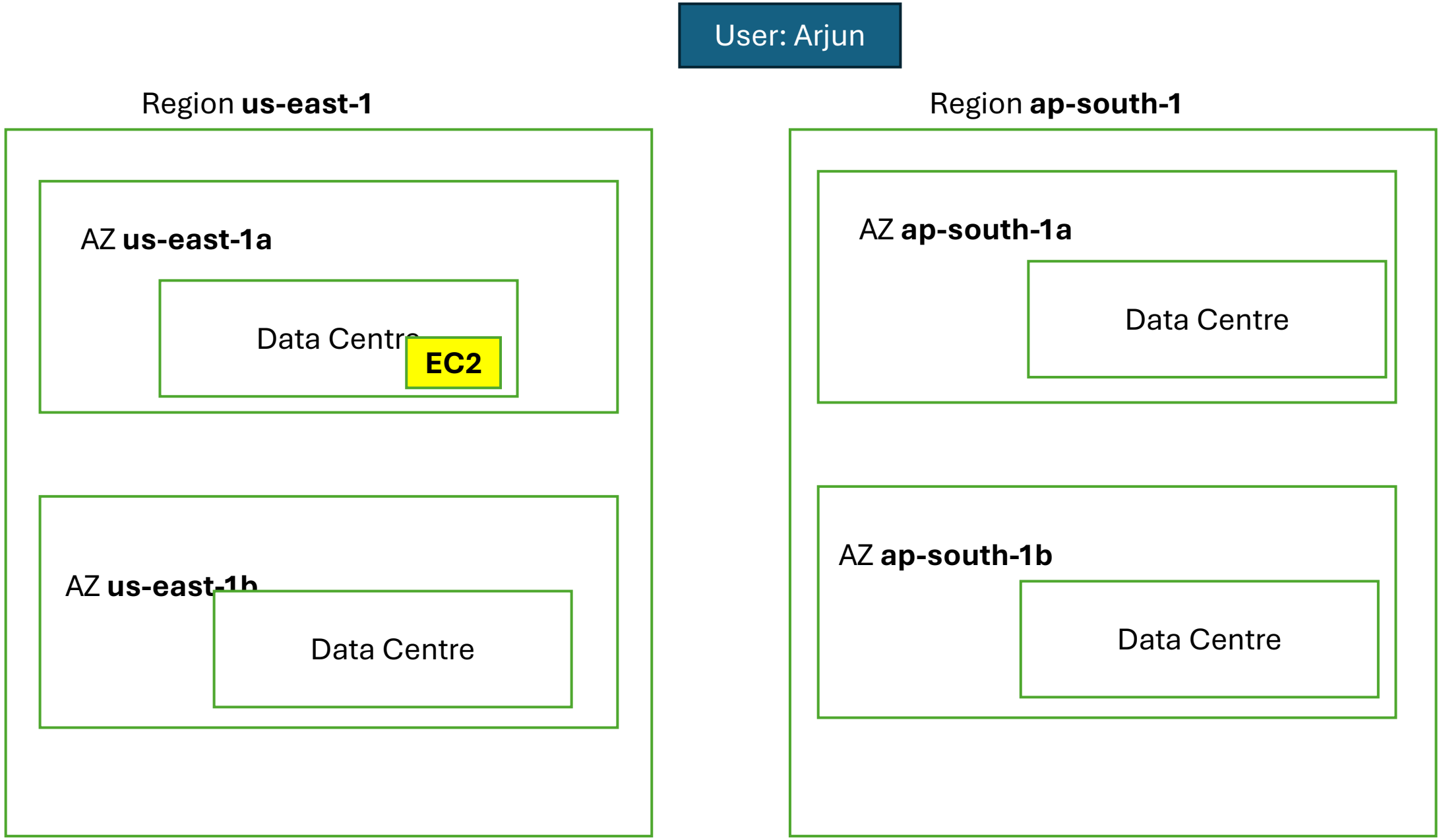
North America

Geographic Regions 9 —

- AWS GovCloud (US-East)
- AWS GovCloud (US-West)
- Canada (Central)
- Canada West (Calgary)
- Mexico (Central)
- US West (Northern California)
- US East (Northern Virginia)
- US East (Ohio)

● Available ○ Coming soon




User: Arjun
Region us-east-1
AZ us-east-1a
Data Centre
EC2
AZ us-east-1b
Data Centre
Region ap-south-1
AZ ap-south-1a
Data Centre
AZ ap-south-1b
Data Centre

OOP Python

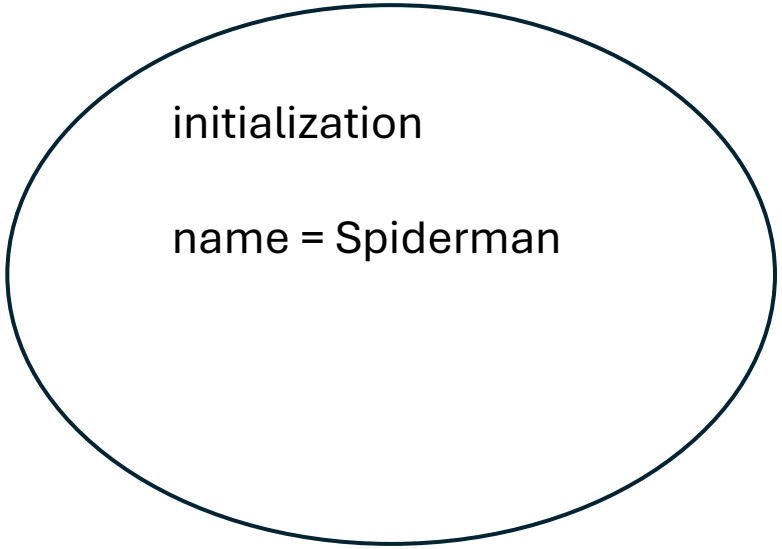
```python
class SuperHero:
    """
    This is a demo class. We will create an object later on.
    """
    def __init__(self):
        self.name = 'Spiderman'

    def printSuperHeroDetails(self):
        print(self.name)
```

Class

spiderman_boy

address reference

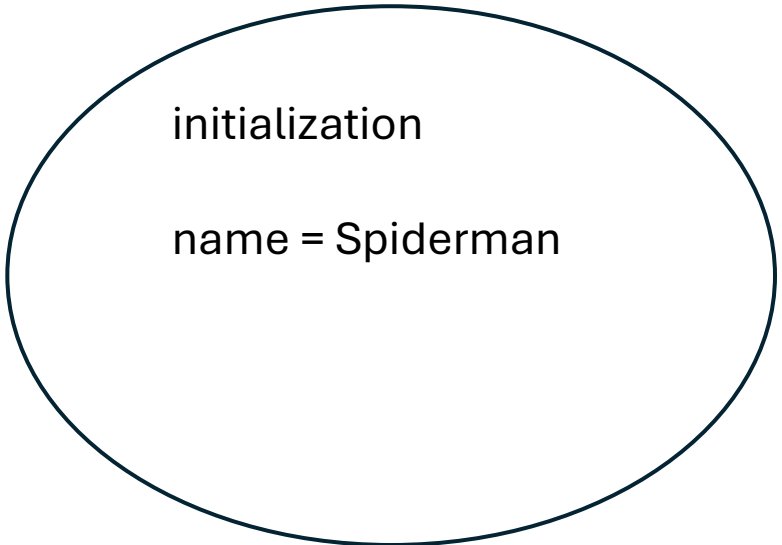
1234



spiderman_adult

address reference

4567



120

60

120 * 0.5

20

20

---

20 + 25

53.3333

60

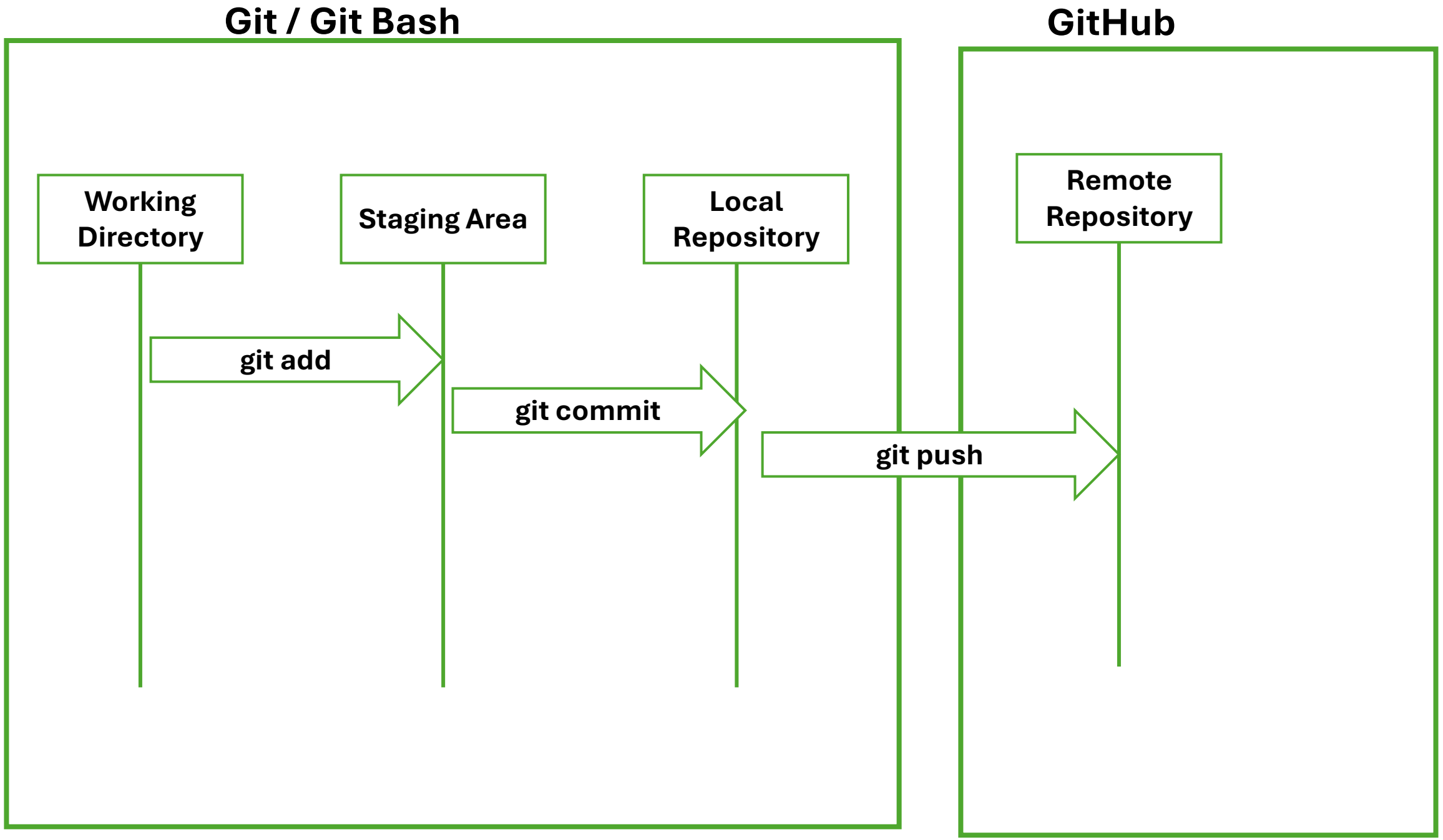
25

25

---

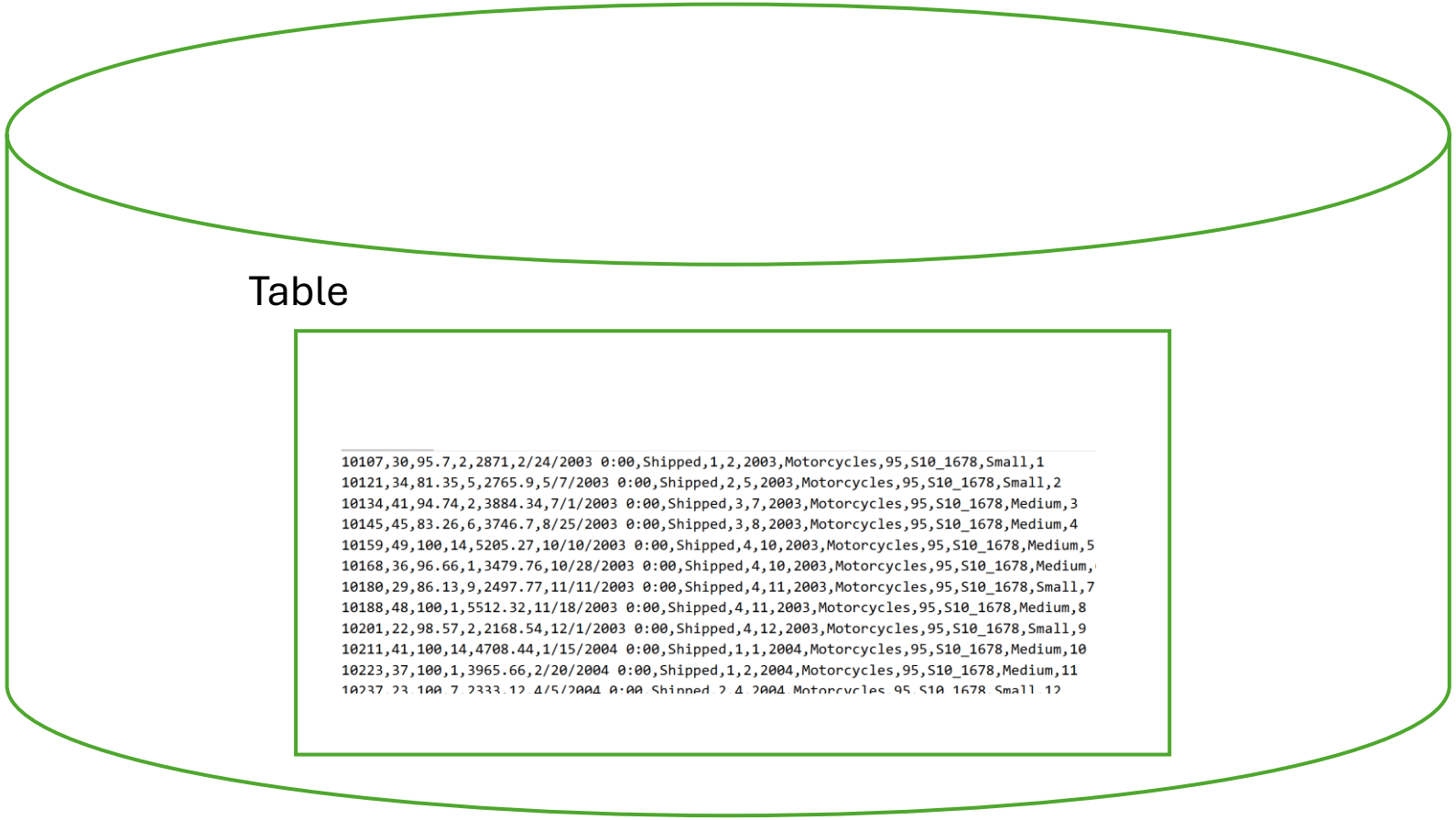
20 + 25

66.6666

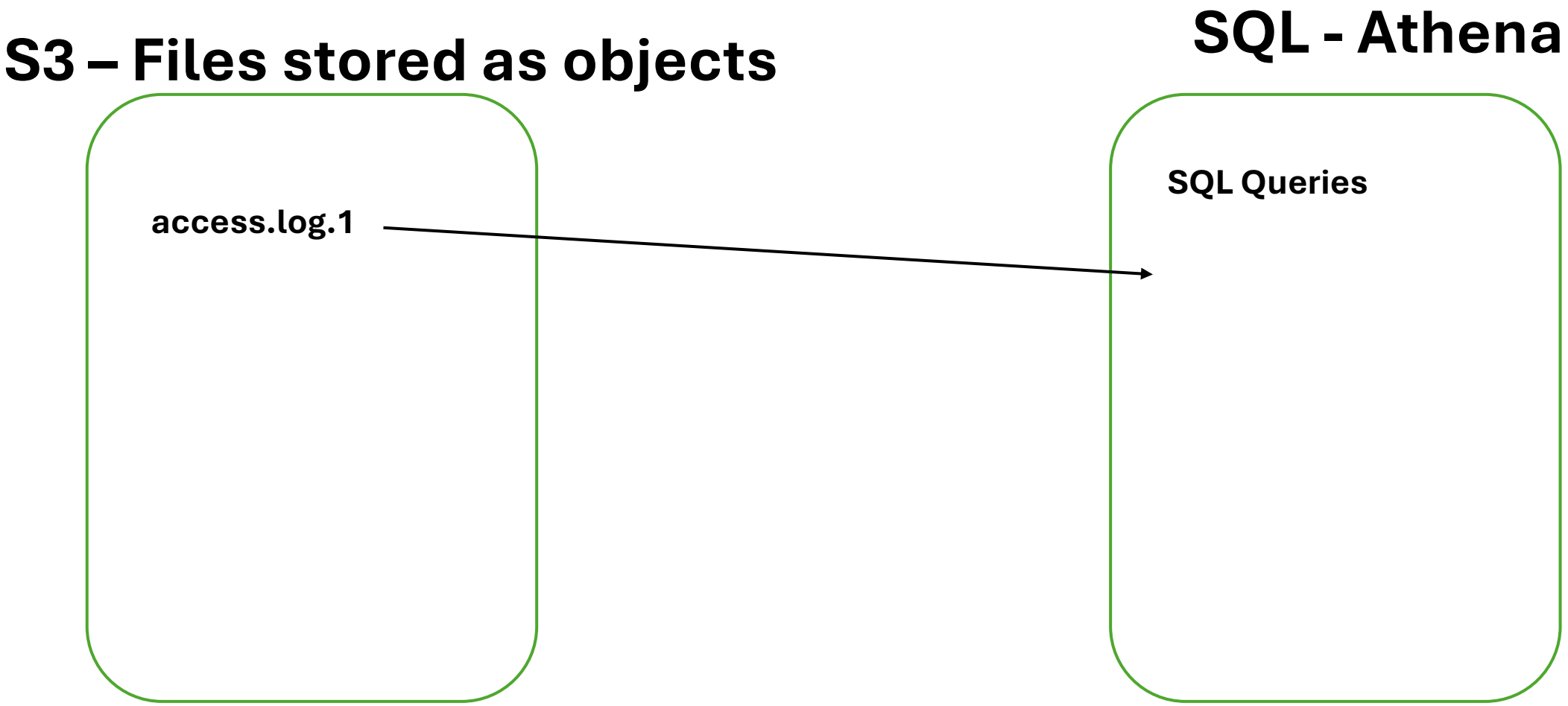

Git / Git Bash
GitHub
Working Directory
Staging Area
Local Repository
Remote Repository
git add
git commit
git push

Native Tables (Managed Tables)

Oracle
MySQL

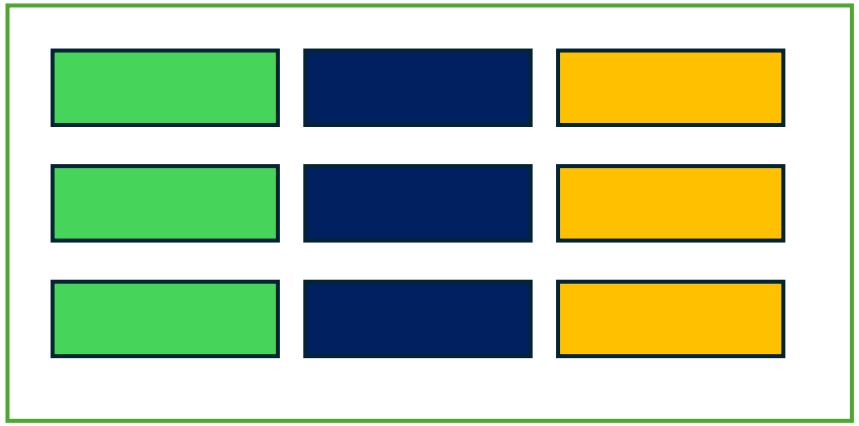


External Tables

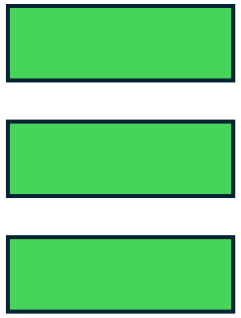


Row-oriented storage Vs Columnar storage

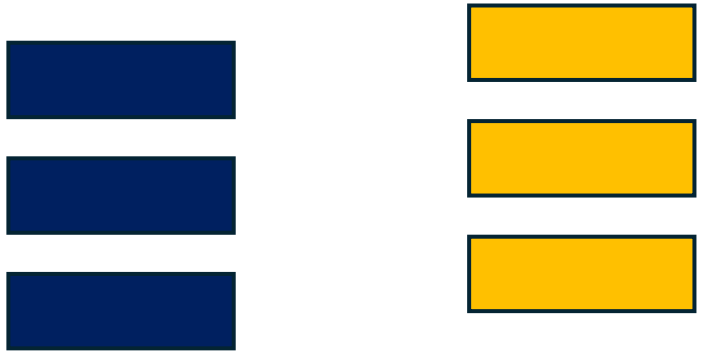
Row-oriented storage



csv



parquet



Weird Analogy

Developer

- No code → No bugs

Data Analytics

- No data processed → No data processing time → No data processing cost

- Less data processed → Less data processing time → Less data processing cost

Monolithic Computing

Traditional Data Analytics

SQL Clients
- SQL Plus
- Toad
- SQL-Developer

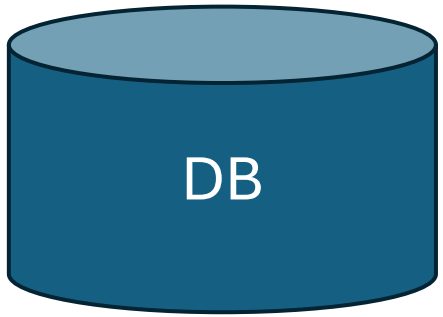
IDE
- PyCharm

SQL

Jupyter

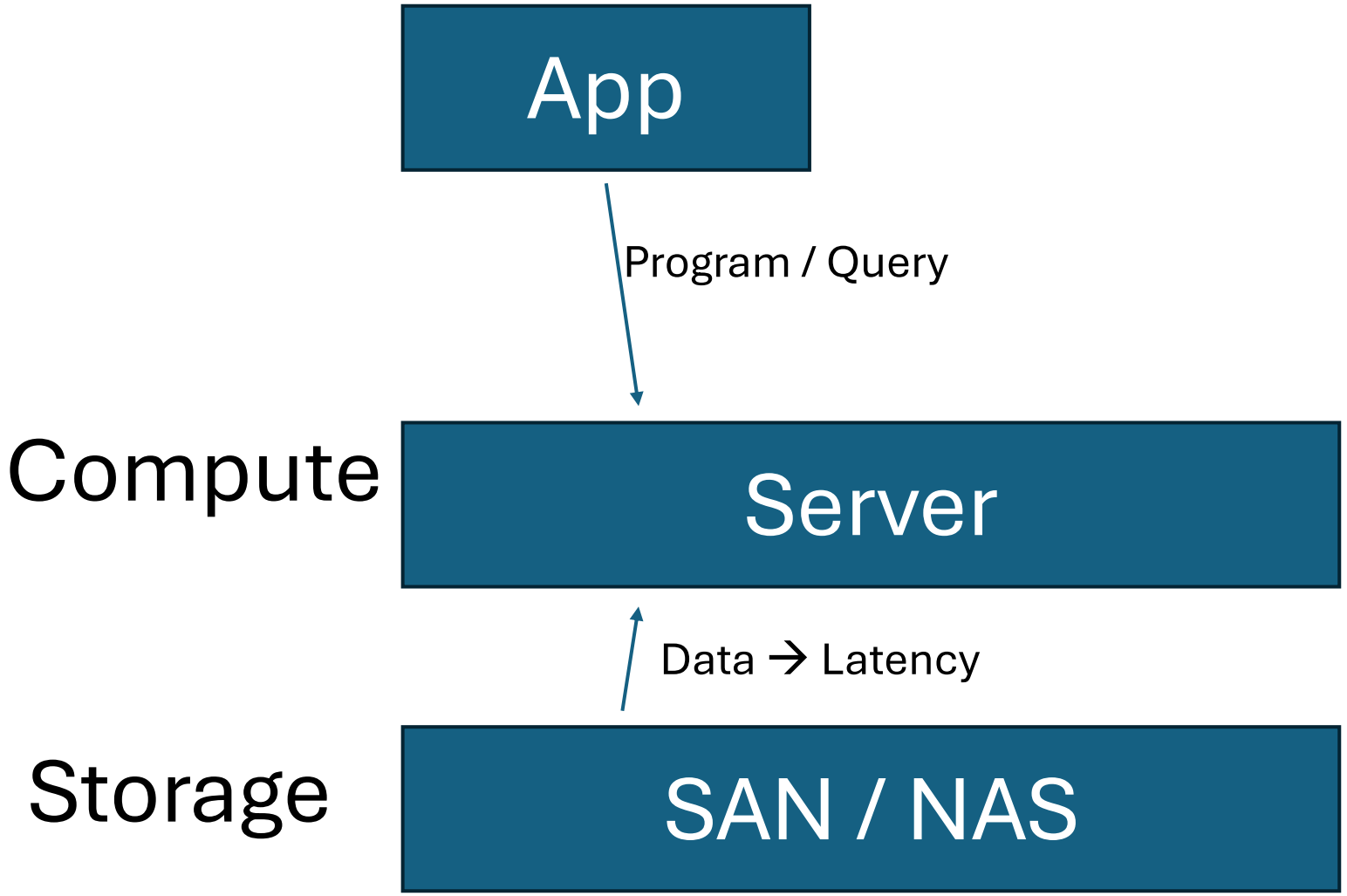
Marimo

Python



Excel

Monolithic Computing



Big Data

- Early 2000s
  - Social Media
  - IOT
  - Transactional Data, Behavioral Data, Operational Data, Log Data, Clickstream Data….
- Structured, Unstructured, Semi Structured

Data Storage

Developer

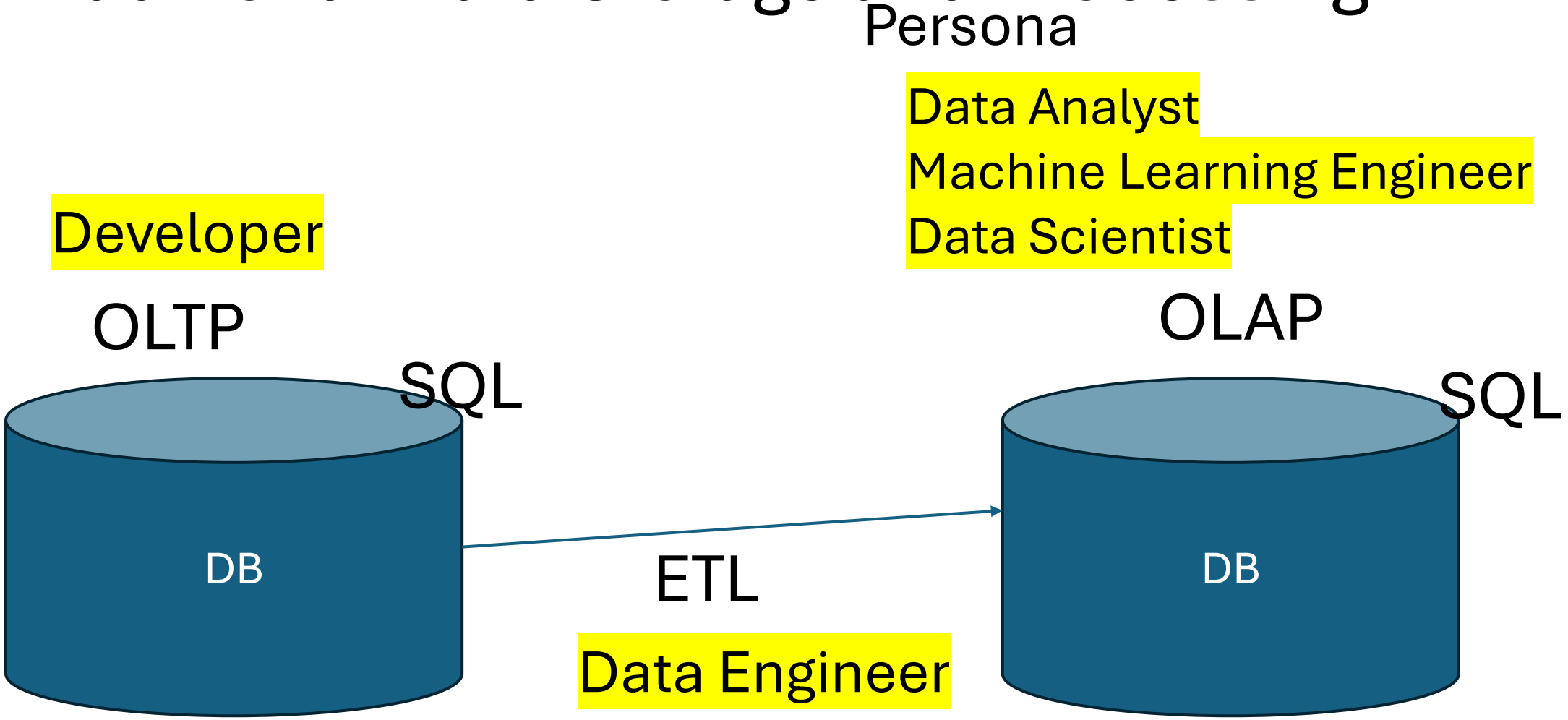
- OLTP (Database)
  - Oracle, MySQL, Postgres, SQL Server
  - ACID compliant
  - Normalized
  - Meant for transactional processing
  - Quick Inserts and Updates
  - Row Oriented

Persona

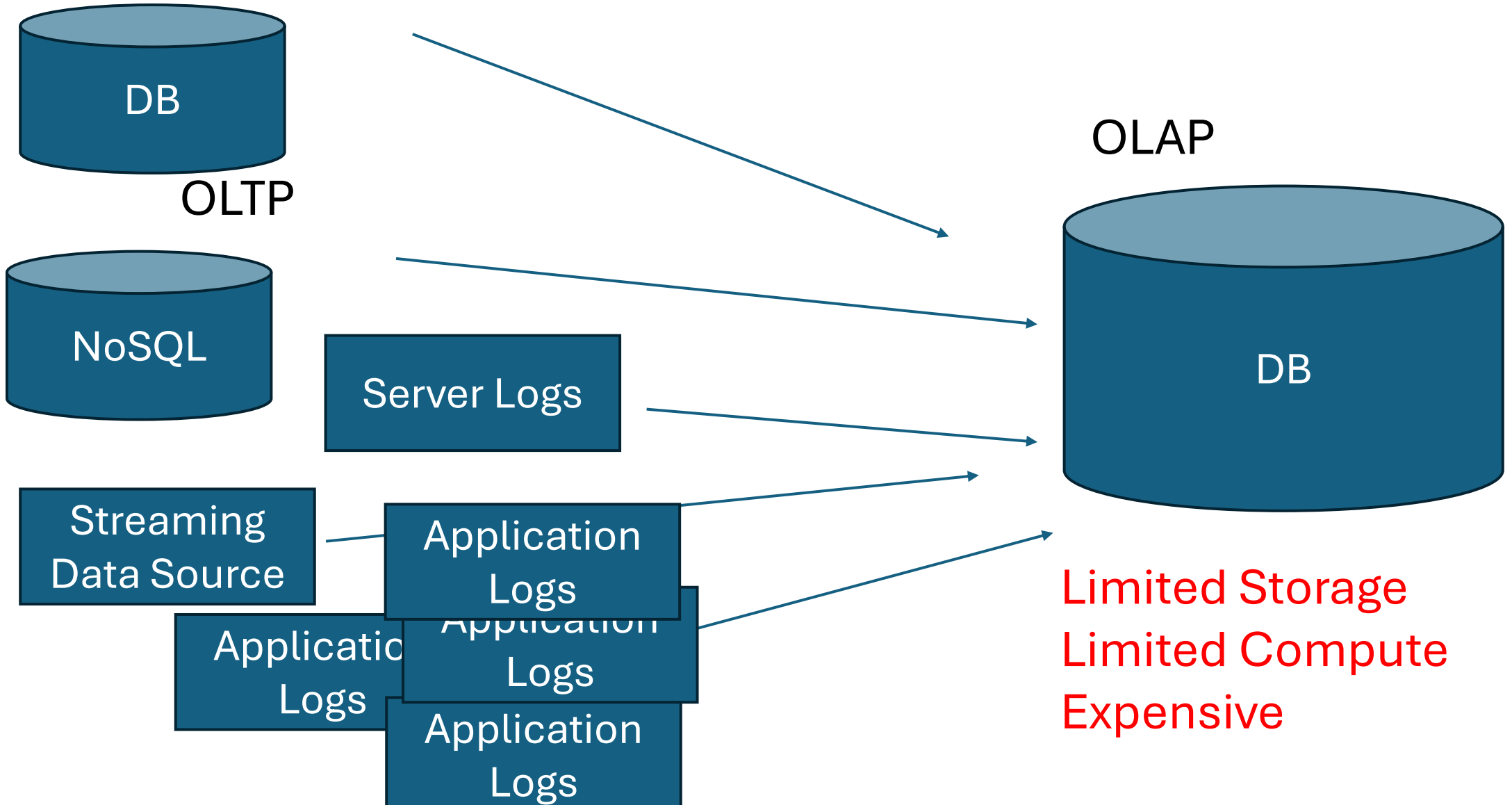
Data Analyst
Machine Learning Engineer
Data Scientist

- OLAP (Datawarehouse)
  - Teradata, Exadata, Vertica
  - Denormalized
    - Star Schema
    - Snowflake
    - Galaxy
  - Meant for aggregations
  - Columnar

Traditional Data Storage and Processing



Modern Applications



Modern Applications with Cloud (AWS)

OLTP

Source Systems

Any systems that generate data

Streaming Data

Kinesis

Spark Streaming

Kafka

ETL

Spark

Glue

S3

Redshift

EMR

DynamoDB

Spark is open-source

Data processing

- Batch Processing
  - Files / Databases
  - Offline Processing
  - Read huge data and process
  - Can be high latency jobs

  - Ex. Credit Card Statement Generation

- Stream Processing
  - Live Streams of Data
  - Real Time Processing
  - Read small chunks of data and process
  - Low latency jobs

  - Ex. Real time fraud detection