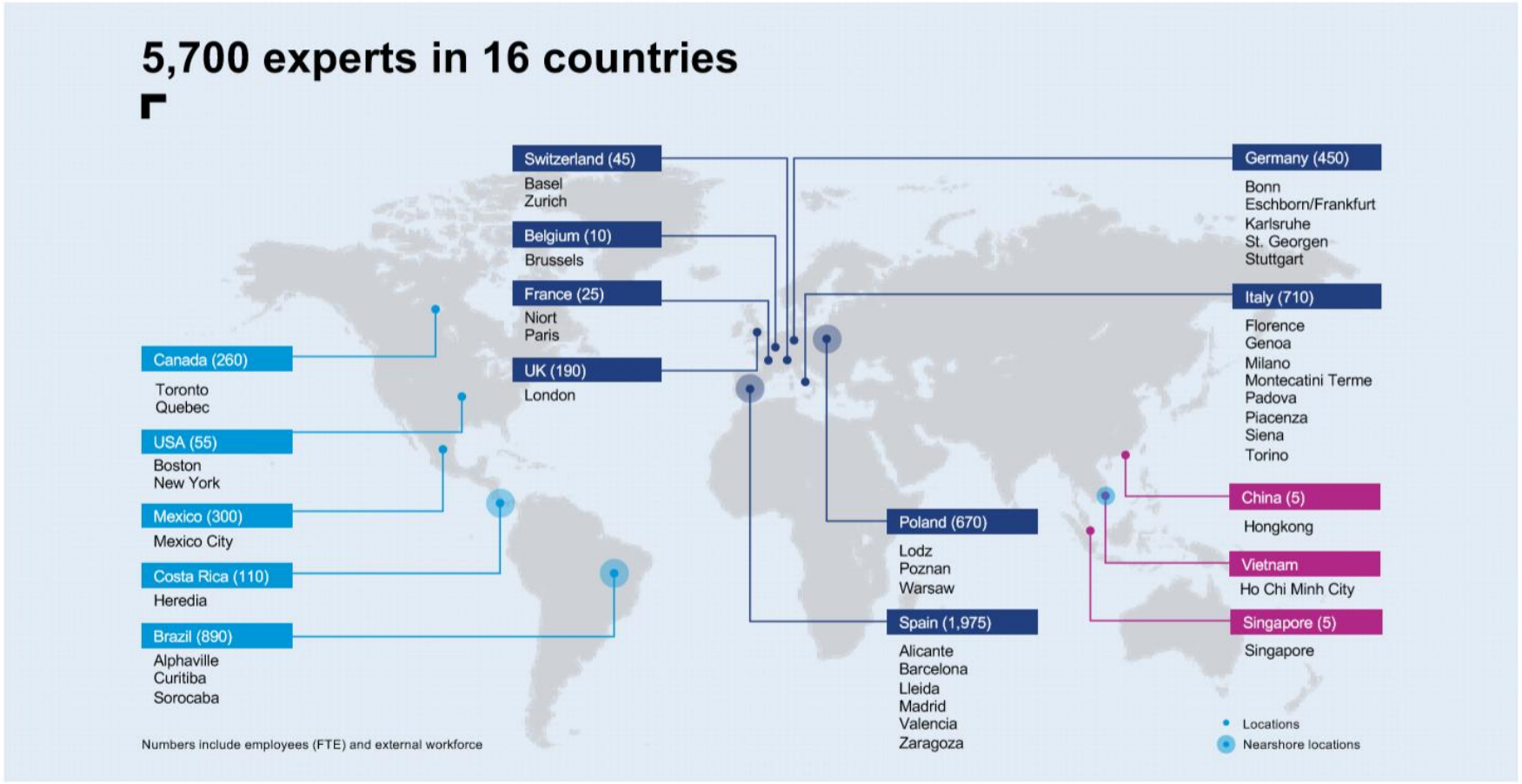
CLOUD COMPUTING

INTRODUCTION TO
CLOUD DATA PROCESSING

Jakub Kasprzak
GFT Poland, UAM, Big Data 2021

THE REVOLUTION

ABOUT **GFT** & ME



Jakub Kasprzak
Principal Consultant
AWS Data Architect

jakub.kasprzak@sqlpedia.pl

GFT

sqlpedia

INTRODUCTION TO CLOUD COMPUTING
AGENDA

- How clouds are formed
- What the cloud computing is
- Evolution of cloud computing
- Solution architectures, design principles
- What kind of services we can find on the cloud – quick overview
- Examples of (big) data processing using cloud services
- Migration to the clouds

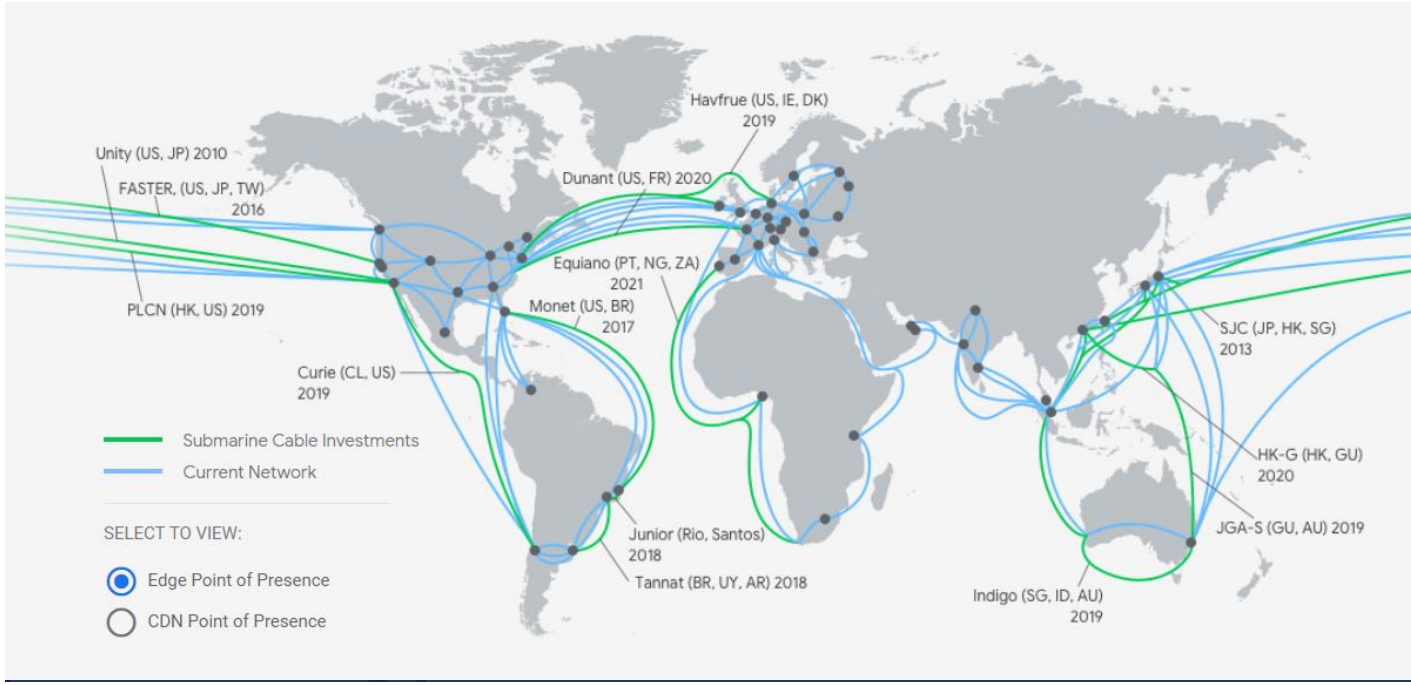
HOW CLOUDS ARE FORMED

HOW CLOUDS ARE FORMED
DRIVERS AND FACTORS

- **Resource sharing** and usage optimization, hard to predict demand for computing power
- **Fast time to market**, infrastructure and environments (DEV/STG/UAT) on demand
- Building new solutions based on existing components (**focus on your business**!)
  Go global in minutes, short *time-to-market (6 months as a base-line for projects)*
- Testing and experimenting with new ideas
  low entry-cost for new projects, testing with different components, infra
- High availability and **scalability**
- Advanced technologies used as services
  Building skills to DIY vs consuming the services

WHAT DOES A CLOUD LOOK LIKE
INFRASTRUCTURE

- Geographic Regions (EU, America, ASIA etc.)
- **Regions** consists of multiple AZ (at least two) it is a physical location around the world with cluster data centers.
- **Availability Zone** (AZ) is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region
- High througput **network**, multiple parallel 100Gbs links
- Edge points / Content Delivery Network
- Dedicated connections (np. AWS Direct Connect, Azure ExpressRoute)
- **Azure Stack Arc, AWS Outposts, GCP Anthos** (Compliance and Data Residency)



CLOUD MIGRATION
PUBLIC CLOUD PROVIDERS

- Is it right for my business ? What technologies do I need ? What are my motivations ?
- Compliance with regulators
- **Which cloud is the best?**
- Service availability in your region
- Skills of my IT team ?
- Cloud migration - multi-cloud providers
- Vendor lock-in. Cloud agnostic, does it work ?

aws

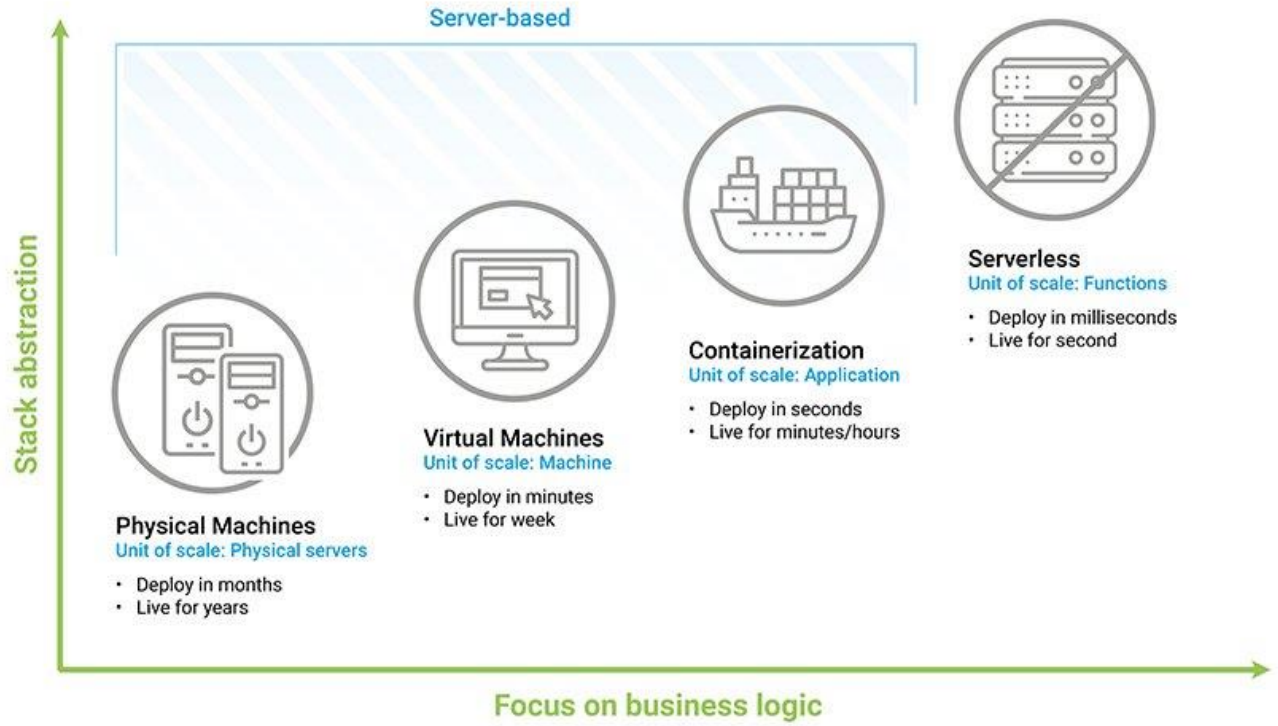
Microsoft Azure

Google Cloud

Figure 1. Magic Quadrant for Cloud Infrastructure and Platform Services

EVOLUTION OF CLOUD COMPUTING
WAY TO SERVERLESS

- Physical Machines
  The old world.
- Virtual Instances
  virtualized servers, you can change their capabilities with the click of a button
- Containerization
  Run your application and its dependencies in resource-isolated processes
- Serverless (functions)
  abstract the execution environment from the code you want to execute



Source https://blogs.itemis.com/en/how-serverless-architectures-change-development-process-and-project-business

IAAS, PAAS, SAAS
RESPONSIBILITY MODEL

On-Prem

- **Customer's responsibility (CR) :** FULL

IaaS

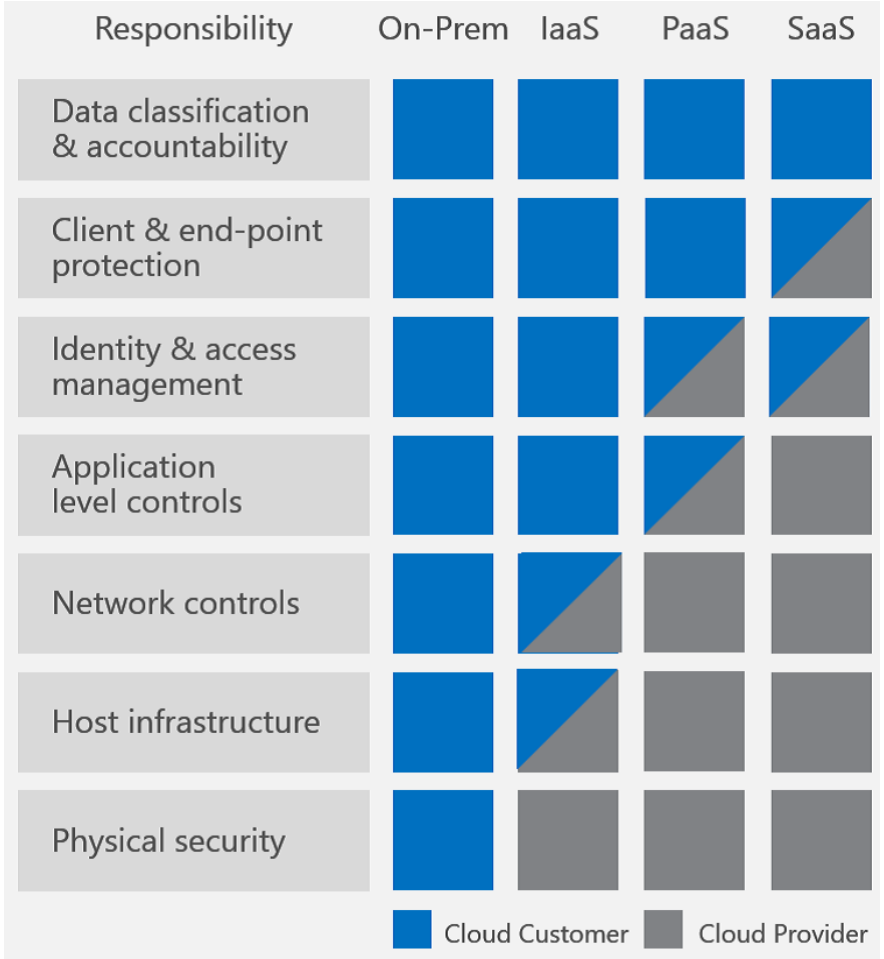
- **CR :** People, Data, Applications, Runtime, Middleware, Operating System, Virtual Network
- **CSP responsibility:** Hypervisor, Servers, Storage, Physical Network

PaaS

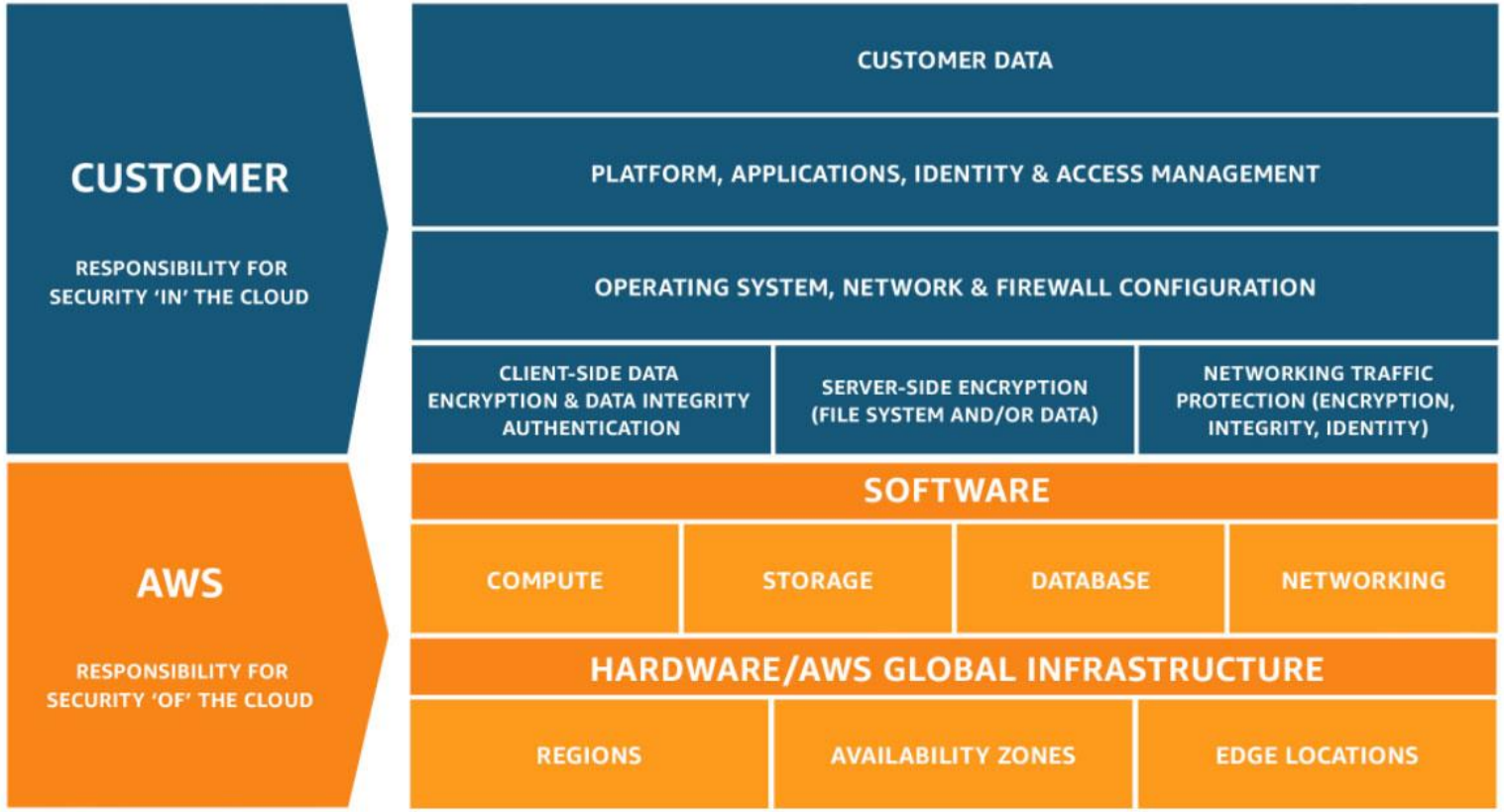
- **CR:** People, Data, Applications
- CSP: Runtime, Middleware, Operating System, Virtual Network, Hypervisor, Servers, Storage, Physical Network

SaaS

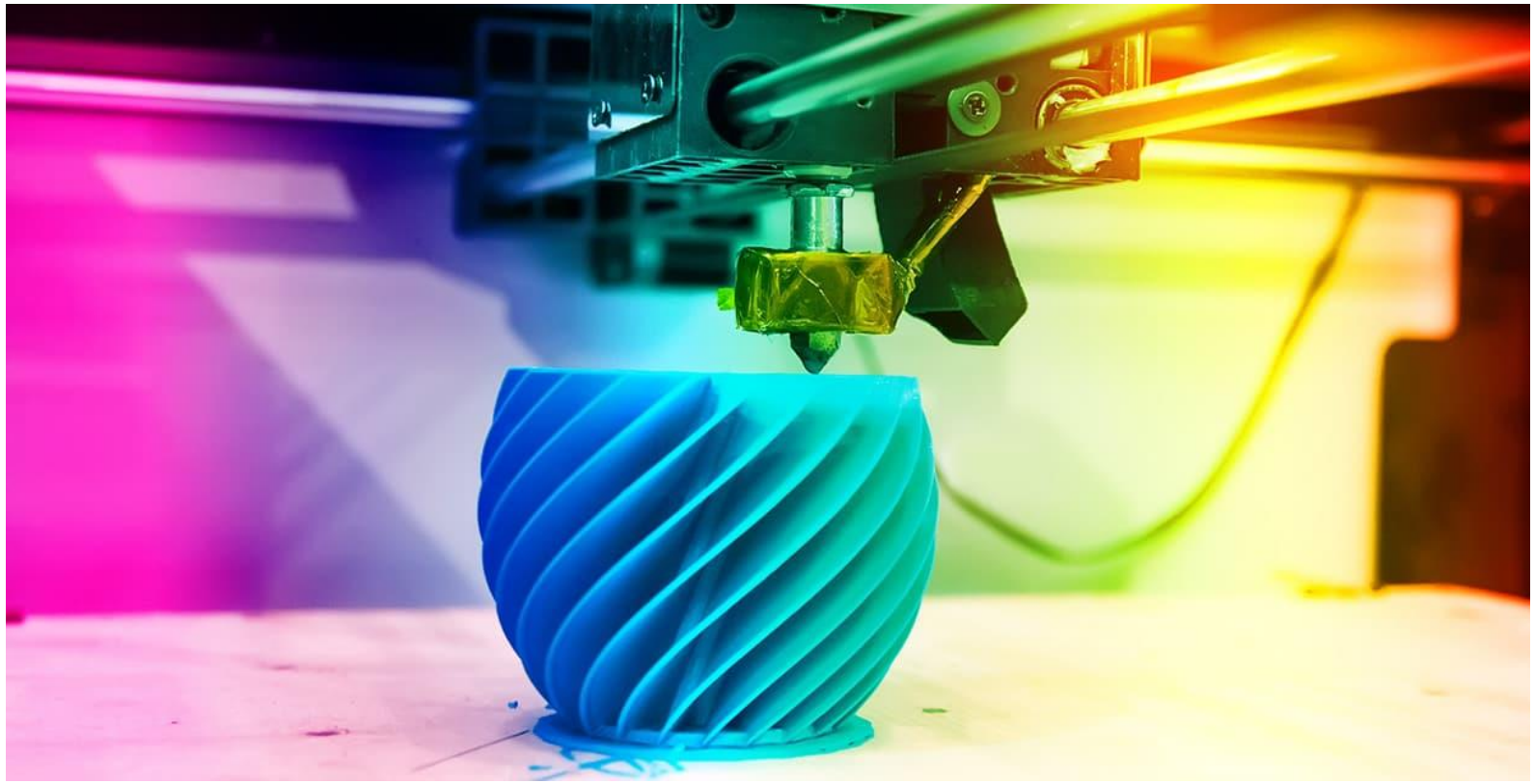
- **CR : People, Data**
- **CSP :** Applications, Runtime, Middleware, Operating System, Virtual Network, Hypervisor, Servers, Storage, Physical Network



EXAMPLE
AWS SHARED RESPONSIBILITY MODEL



HOW WOULD YOU DESCRIBE AN OBJECT?



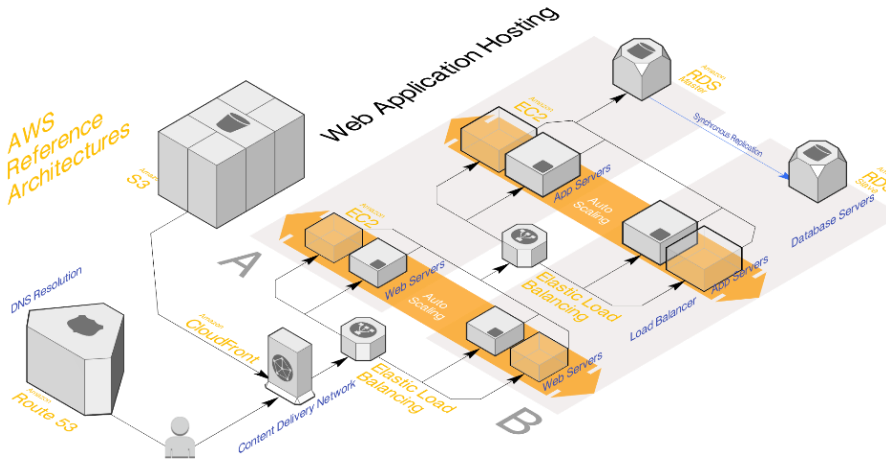
INFRASTRUCTURE AS A CODE
IAAC

- Cloud native approaches AWS Cloud Formation, GCP Deployment Manager, Azure Resource Manager
- It enables you to **quickly** set up your complete infrastructure by running a script (for all ENVs !)
- you can go through the same **version control**, automated testing and **CI/CD** processes
- It's the single source of truth for your configuration (**consistency**). You guarantee the same configurations will be deployed over and over, without discrepancies
- Lowering the **costs** of infrastructure management

HashiCorp Terraform

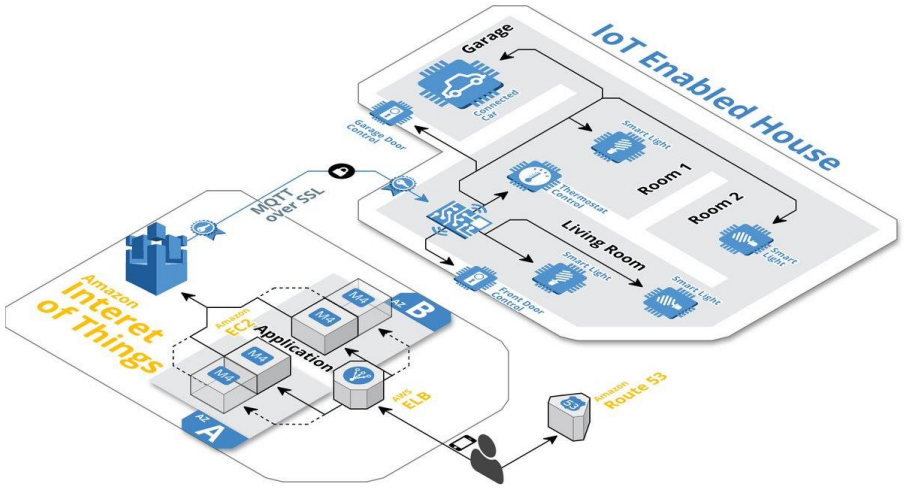
CLOUD SOLUTIONS ARCHITECTURE
DESIGN PRINCIPLES

- Every design must be justified by the **business**
- Preapre yourself for **failuers**, they happen
  Failures detection (metrics, alarms), automatically recover, logging, **self-healing** (AWS Fault Injection Sim), idempotency, retries (transient failures), fail over, isolation
- Stop guessing your **capacity** needs
  Idle resources, under/over provisioning, autoscaling (up/down)
  test at production scale, partitioning (horizontal and functional)
- **High availability** - redundancy
  SPOF, Load Balancers, replication
- **Serverless** vs Managed Services vs Virtualization
  When possible use Serverless / PaaS rather than IaaS (Presto/Athena, RabbitMQ/SQS, Hadoop/EMR, Kafka/Kinesis)



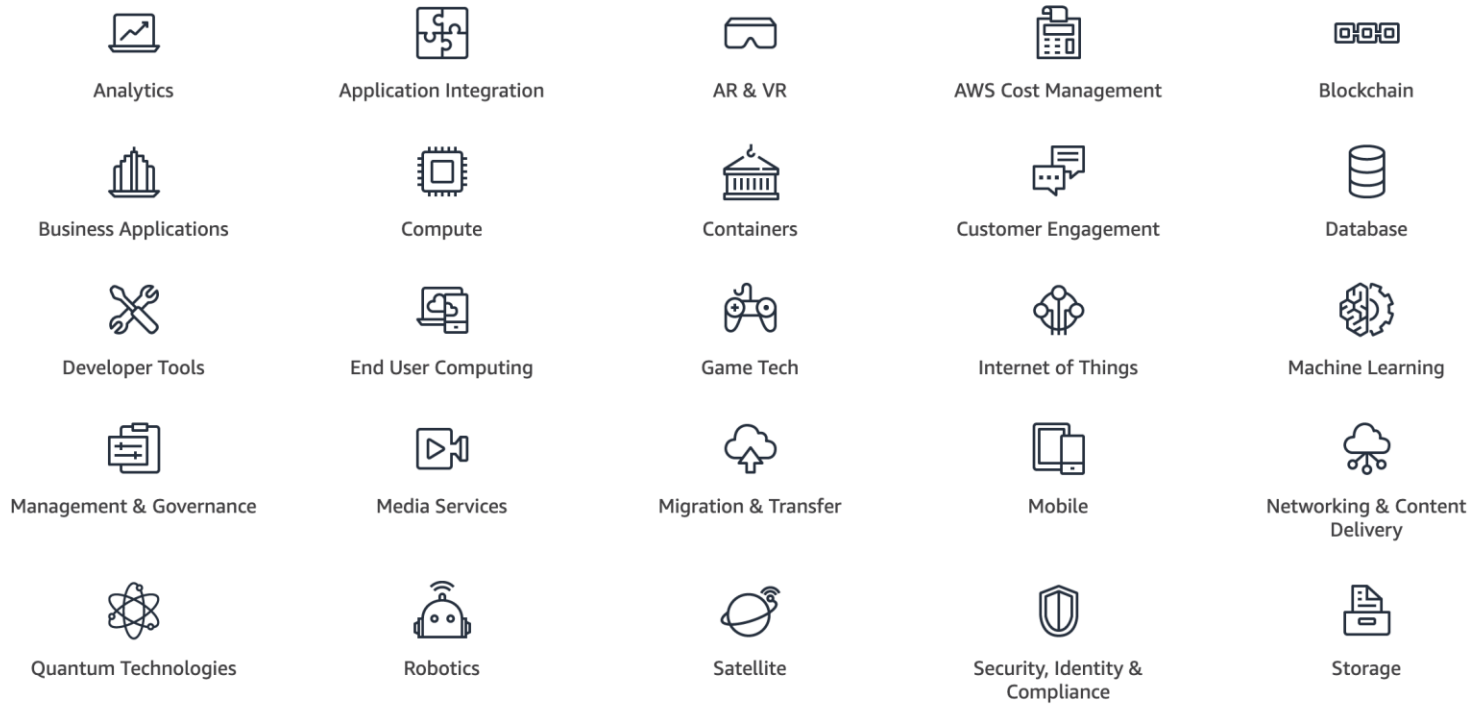
CLOUD SOLUTIONS ARCHITECTURE
DESIGN PRINCIPLES

- **Infrastructure as a Code** / configuration as Code
  Automated deployments CI/CD – frequent small deployements, limiting human errors in the process
- Don't use one solution for everything
  There is no silver bullet, RDBMS/NoSQL/Hadoop are great but there is no one answer for all needs/questions
- **Boudned-context** approach (Domain Driven Design)
  A bounded context maps to a subdomain of the business domain
  Partition data based on processing logic
- Preapre for evolution – **loosely coupled architecture**
  Async messaging



CLOUD SERVICES
AWS/ AZURE/ GCP

- Compute
- Storage
- Data Analytics
- Data Collection
- Databases
- Networking
- Security
- AI / ML
- IoT



CLOUD COMPUTE
EC2, EKS, ECS, FARGATE

- EC2 instances don't show the full potential of cloud services !
- **Types** of the instances adjusted to the needs
  Dedicated (bare metal), general purpose, memory, compute, storage optimized, accelerated computing (GPU)
- EC2 **purchasing** options
  On-demand, reserved (marketplace), spot instances, dedicated hosts (single tenant hardware)
- EC2 **Autoscaling** groups
- Compute engine for **containers**
  choice between serverless (Fargate) and managed service (ECS/EKS based on EC2 instances)
- Farget Spot instances (up to 70% discount)
  Ideal for fault-tolerant use cases such as big data, CI/CD, and batch processing

EC2

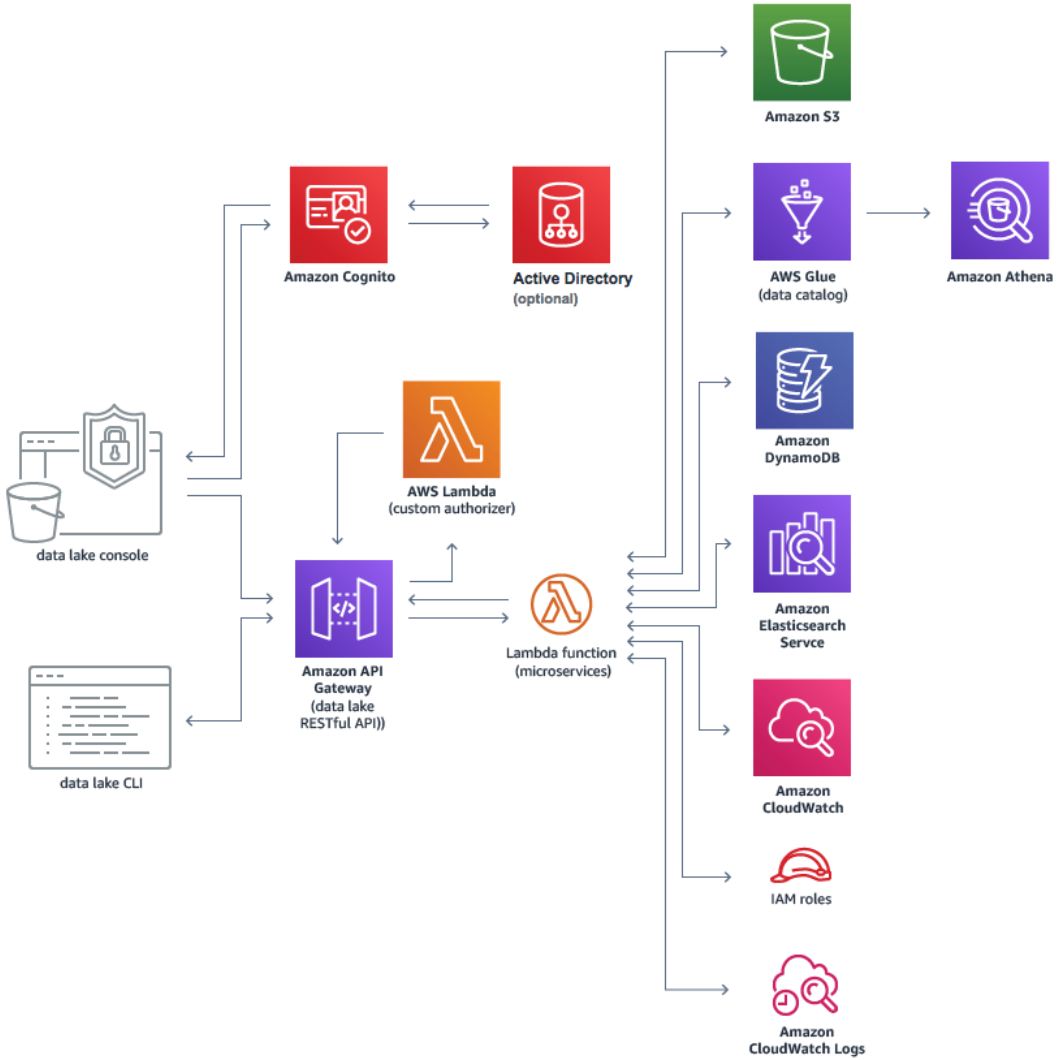
ECS

EKS

Fargate

CLOUD SERVERLESS COMPUTE
AWS LAMBDA

- Serverless **event-driven data processing**
  You can run code without provisioning or managing servers. You pay only for the compute time you consume.
- Great integration with other services
  Could be triggered by S3 events, DynamoDB, Kinesis, SQS, Step functions and many others
- Great for small, short tasks
  Limits : max 10GB memory, 900s, 1000 concurent executions (soft limit), layers 5/250MB
- Use cases
  Continuous ETL's, IoT backends, Web applications, API endpoints, support for your own docker images (up to 10GB in size)



Source https://aws.amazon.com/solutions/data-lake-solution/

CLOUD ANALYTICS
EMR, GLUE

- **Elastic Map Reduce**
  Managed cluster platform that simplifies running big data frameworks
- The modern way how to use **Hadoop**
  Launch your clusters in minutes. Don't worry about node provisioning, infrastructure setup, Hadoop configuration, versions compatibility
  Scale your cluster if needs be
- A broad range of Big Data technologies in one place
  Spark, Flink, Presto, Pig, Scoop, Hadoop, HBase, Hue, JupyterHub, Zeppelin, Mahout, Mxnet, TensorFlow and many others
- **Glue** is a fully managed ETL service
  Supports Python/Scala code to run ETL tasks (Spark or Python jobs)
- Glue **Data Catalogue** as a metadata store (for jobs/Athena)
- Glue **DataBrew** (lineage)

EMR

Glue

DATABASES
RDS, REDSHIFT, DYNAMODB, DOCUMENTDB …

- **RDS** managed relational database service
  Supports the most popular database engines (MySQL, Postgre, Oracle, SQL Server, MariaDB, AWS Aurora), scalable, highly available (multi AZ deployments, automatic backups snapshots)
- **Redshift** a modern data warehouse
  based on Postgre, petabyte-scale data warehousing & data lake analytics
- **DynamoDB,** NoSQL database
  key-value database that delivers single-digit millisecond performance at any scale (pricing pay on demand / provisioned cap)
- **DocumentDB (Mongo 3.6), Cassandra, QLDB, Neptune** and others
  Specialized database engines for different types of use cases: documents, timeseries, blockchain (banking transactions), graphs.


Amazon RDS


Amazon Redshift


Amazon DynamoDB

OBJECT STORE
SIMPLE STORAGE SERVICE

- The optimal storage solution is based on the kind of access method
- General characteristic
  Durability 99.999999999%, low latency, high throughput performance
- Strong consistency model (new 2020 Nov)
  Replaced old Read after write consistency (for new objects) and eventual consistency for DELETES and PUTS of the existing objects
- Storage tiers
  Standard, IA, One-zone, Glacier, Deep Glacier, Intelligent tiering
- Versioning
- Security (KMS data encryption in transit & at rest),

DATA SECURITY
SERVICES

- Identity and Access Management (IAM**)**
  It's the most important security service for managing access to AWS services and resources
- Key Managements Service (**KMS**)
  Encryption at rest and in transit at every single step of data journey
- Amazon **Macie**
  Data protection & classification. Macie is a security service that uses machine learning to automatically discover, classify, and protect sensitive data (only two US regions at the moment)
- Virtual Private Cloud (**VPC**)
  a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define
- Be prepared for security incidents!

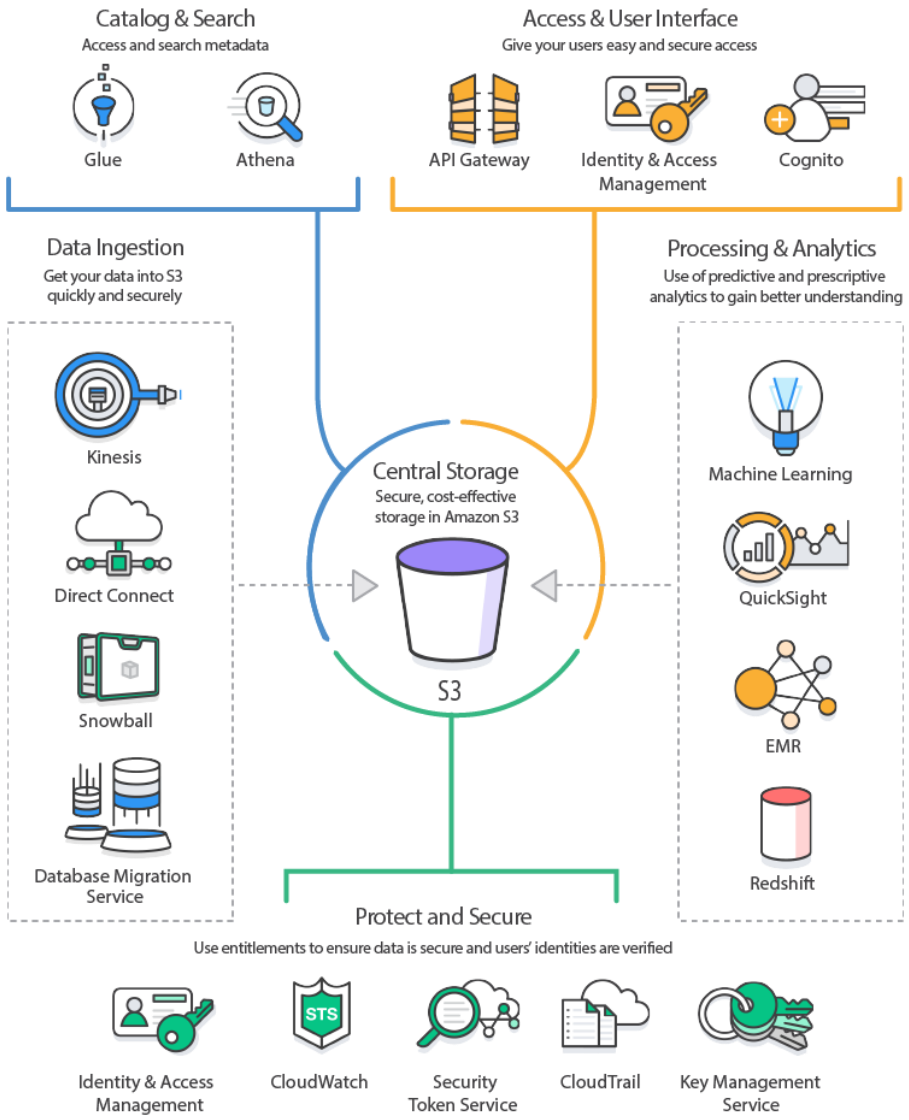

AWS Identity and Access Management (IAM)


AWS Key Management Service


Amazon Macie

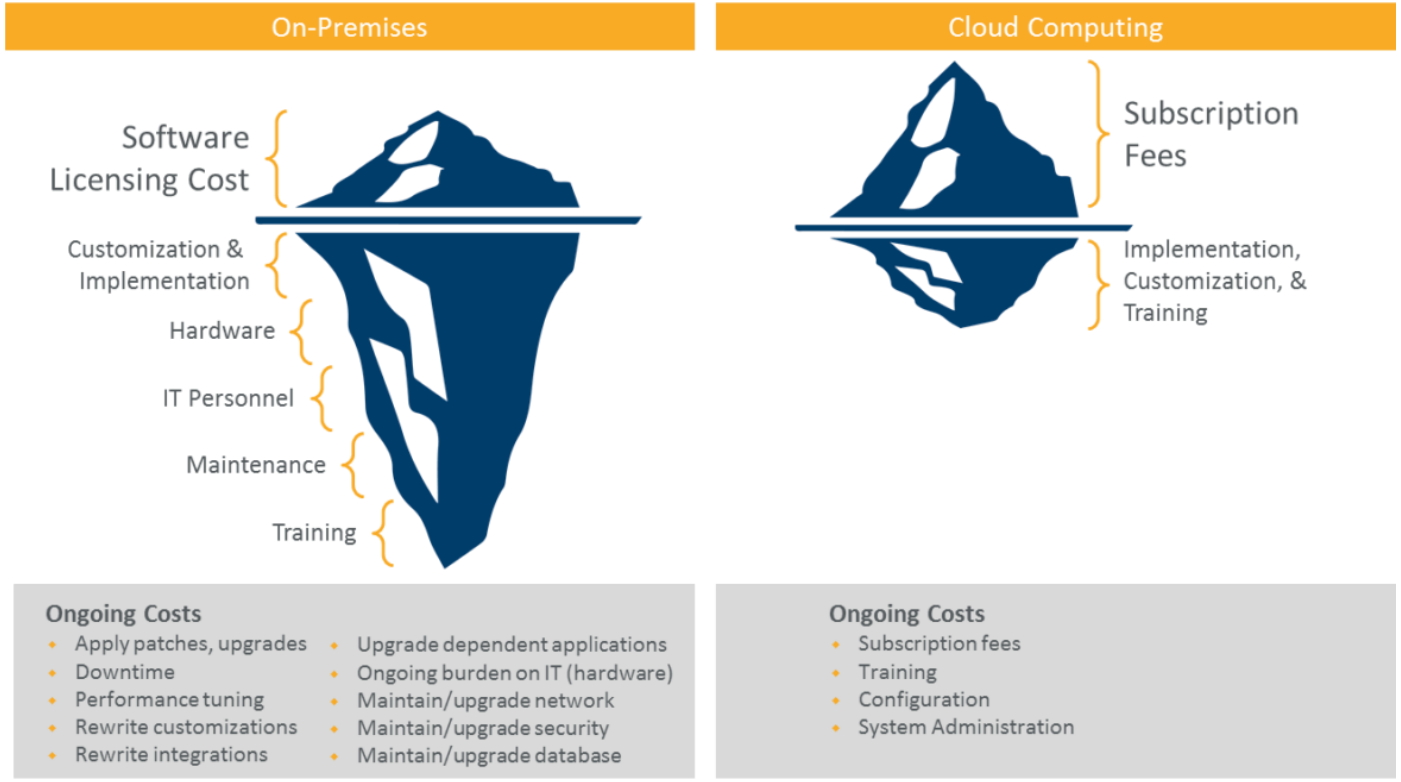
DATA IN THE CLOUDS
BIG DATA USE CASES

- Data Lake
  Infinite (almost) storage, linear scalability, constant delays, HA, good integration with other services
- IoT data processing
  Can support billions of devices and trillions of messages, and can process and route those messages to endpoints and to other devices reliably and securely
- Sagemaker (ML/AI)
  Fully integrated development environment (IDE) for machine learning. Train, tune your ML models, hyperparametrization



MIGRATION TO THE CLOUD
COSTS

- Migration to clouds it's not about moving your infrastructure to the cloud provider 1:1
- Cloud services are new paradigms, new products, new types of services, support, constant development, patching, versioning
- Initial costs are usually higher when you compare them to the classical on-premise data centers



Questions ?

LINKS AND RESOURCES

- GFT project example MOX Bank https://mox.com/features/
- AWS Outpost https://aws.amazon.com/outposts/
- AWS compliance programs https://aws.amazon.com/compliance/programs/
- Shared responsibility model https://aws.amazon.com/compliance/shared-responsibility-model/
- Netflix Simian Army https://netflixtechblog.com/the-netflix-simian-army-16e57fbab116
- Design principles https://aws.amazon.com/architecture/
- Azure design principles https://docs.microsoft.com/en-us/azure/architecture/guide/design-principles/