DTU Compute
Department of Applied Mathematics and Computer Science

DTU

Fault-Tolerant Cloud Computing Architectures

A comparative study

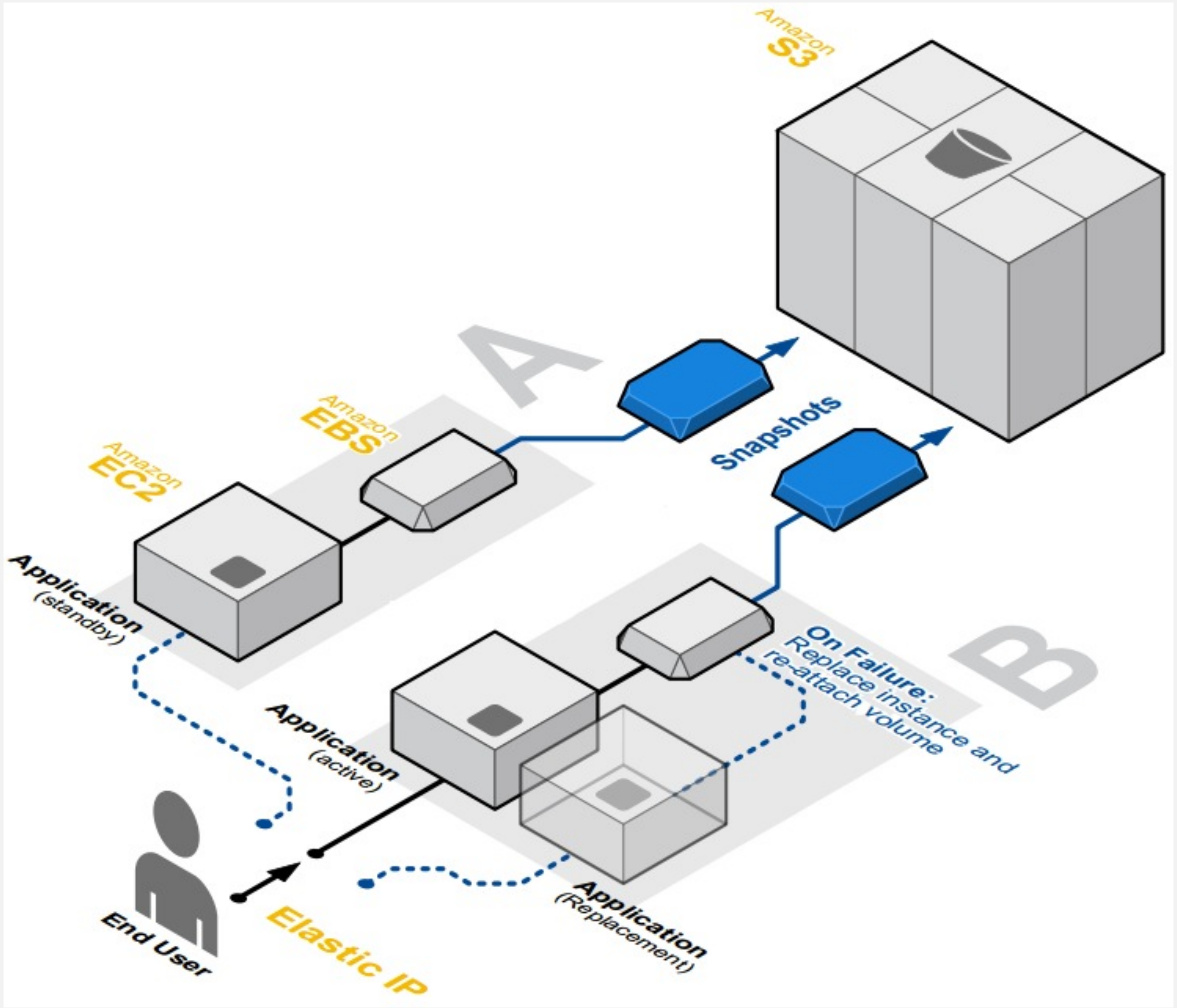
Authors: Andreas Kjeldsen (s092638), Morten Eskesen (s133334)

amazon webservices™

Google Cloud Platform

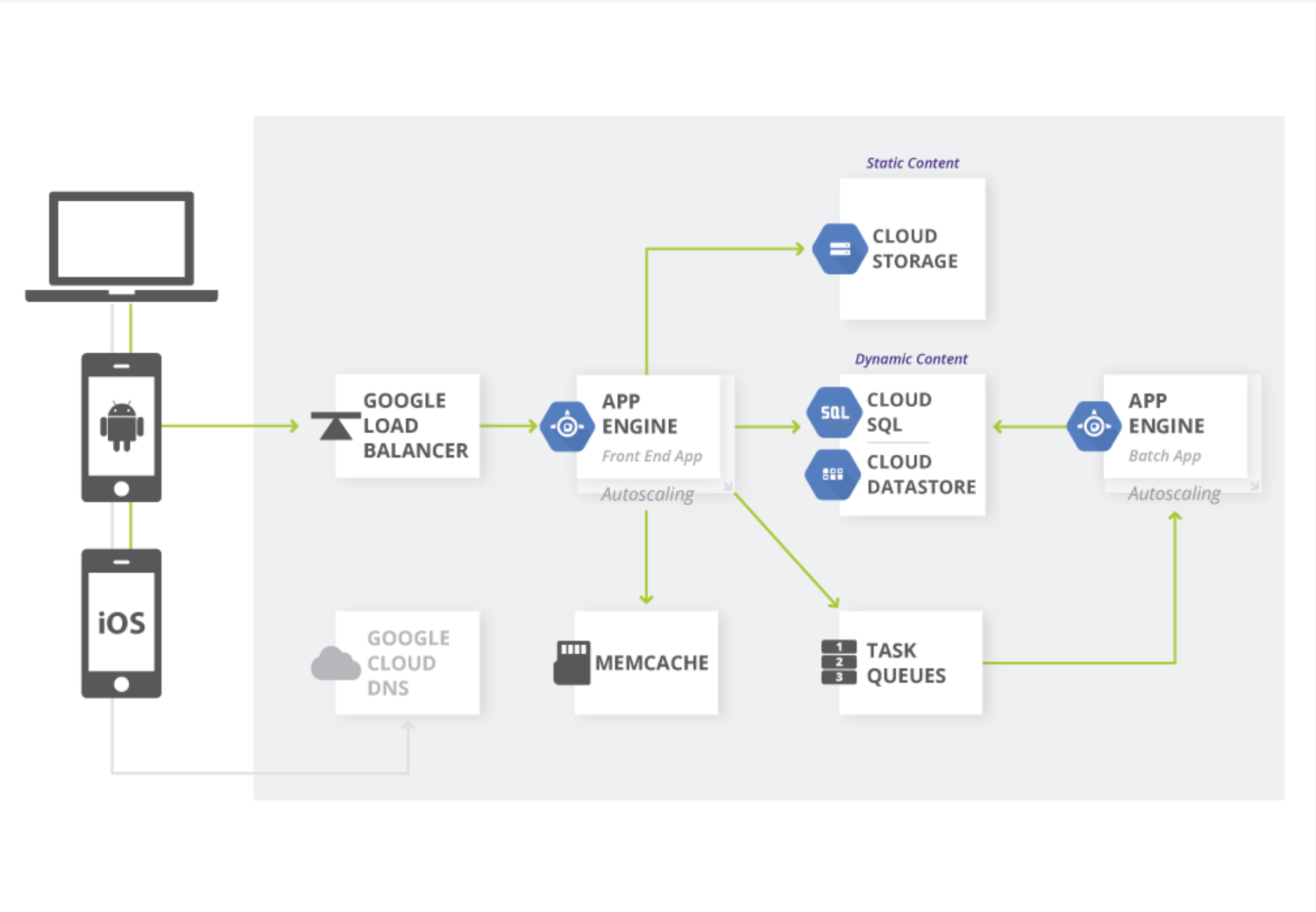
Architecture

Many services offered. Services have different functions.
Some services are fault-tolerant by design and can be connected to other services, thereby making a complete fault-tolerant system.



Example of fault-tolerant application architecture hosted on AWS

Many services offered by Google Cloud Platform with different functions.
Systems built with the Google Cloud Platform are fault-tolerant as the services are designed to be fault-tolerant.



Google Web Application Architecture on Google App Engine

Redundant Storage

Four different storage services: Simple Storage Service (S3), Elastic Block Storage (EBS), Databases (RDS, DynamoDB & SimpleDB), EC2 instances with SSDs.

S3 is highly available and redundant storage. EBS and the databases have replication and can be backed up to S3. The SSDs on the EC2 instances should push data to persist to EBS.

Three different storage services: Cloud Storage (CS), Cloud Datastore (CD) and Cloud SQL (CSQL).

Data in CS is protected through redundant storage at multiple physical locations and highly available. CD replicates data across many data centers using an algorithm to solve consensus problem.
CSQL is highly available, replication and backups are handled by Google.

Availability

Multiple regions available throughout the world.
Each region has multiple availability zones.
Resources are replicated to various availability zones, thus the resources are highly available and failure independent.

Multiple regions throughout the world.
Each region has multiple availability zones.
Deploying applications in different regions achieve a high degree of failure independence.

Load Balancing

Distributes load between multiple computing instances.
Can automatically detect failed computing instances and make sure traffic is only routed to running computing instances.
Can automatically launch new computing instances if resources are becoming limited.

Can be done with either network load balancing or HTTP load balancing.
Distributes the load between multiple computing instances.
Can detect faulty computing instances and replace it and/or distribute the load between running instances.
Allocates computing instances as the traffic requires it.

DTU Compute
Department of Applied Mathematics and Computer Science