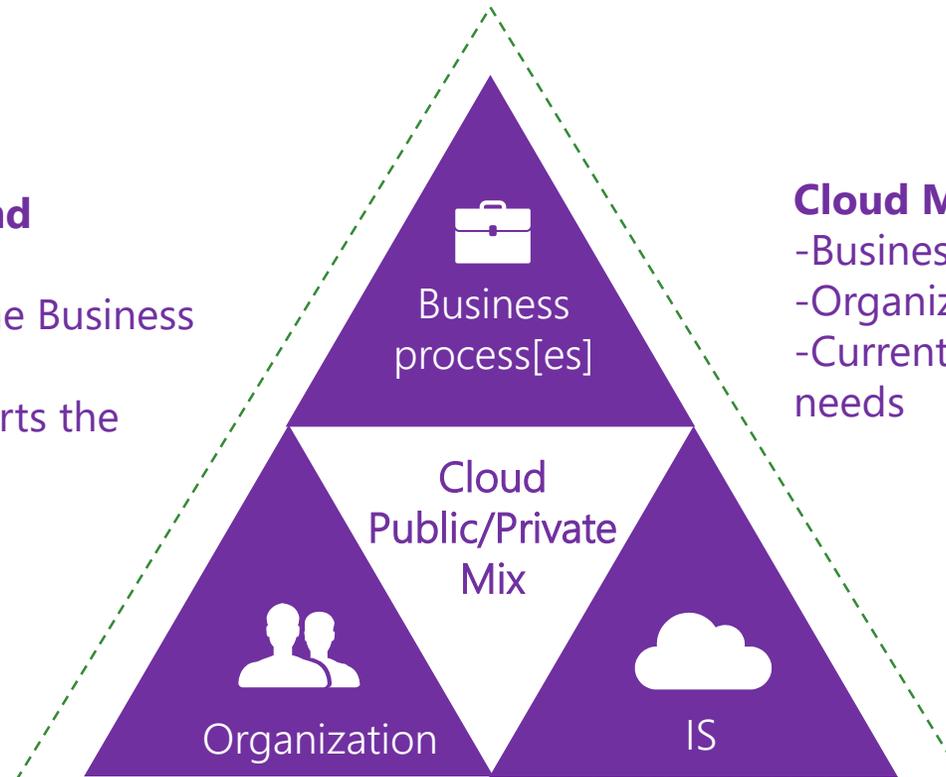


Cloud Systems - Aligning IS to the Business

What is the Strategy and Purpose?

- Align Cloud and IS to the Business Strategy
- Make sure Cloud supports the Business Processes



Cloud Models must support

- Business Requirements
- Organizational Objectives and Culture
- Current IS/IT requirements and future needs

Aligning IS to the Business via Cloud and in particular:

- Data Understanding and Transformation
- Customized Software and Application Renewal, Transformation

THE CLOUD JOURNEY - USE CASES



AWS cloud

Co-Location
Worries about Security
VDI, ROBO
Consistent Workloads
Applications are poorly written too expensive to rebuild or maintain
HA OLTP
Lack of Public Cloud skills

PRIVATE CLOUD

E-Commerce, DevOps
In-consistent Workloads
Web Tier of an app is mission critical, no latency
Applications re-written for Web
Data Storage esp. of critical objects
Mobility
Reduce HW



Google Cloud Platform

PUBLIC CLOUD



We use a methodology and pricing Grid to calculate a budget and solution for our clients and prospects based on their use case requirements

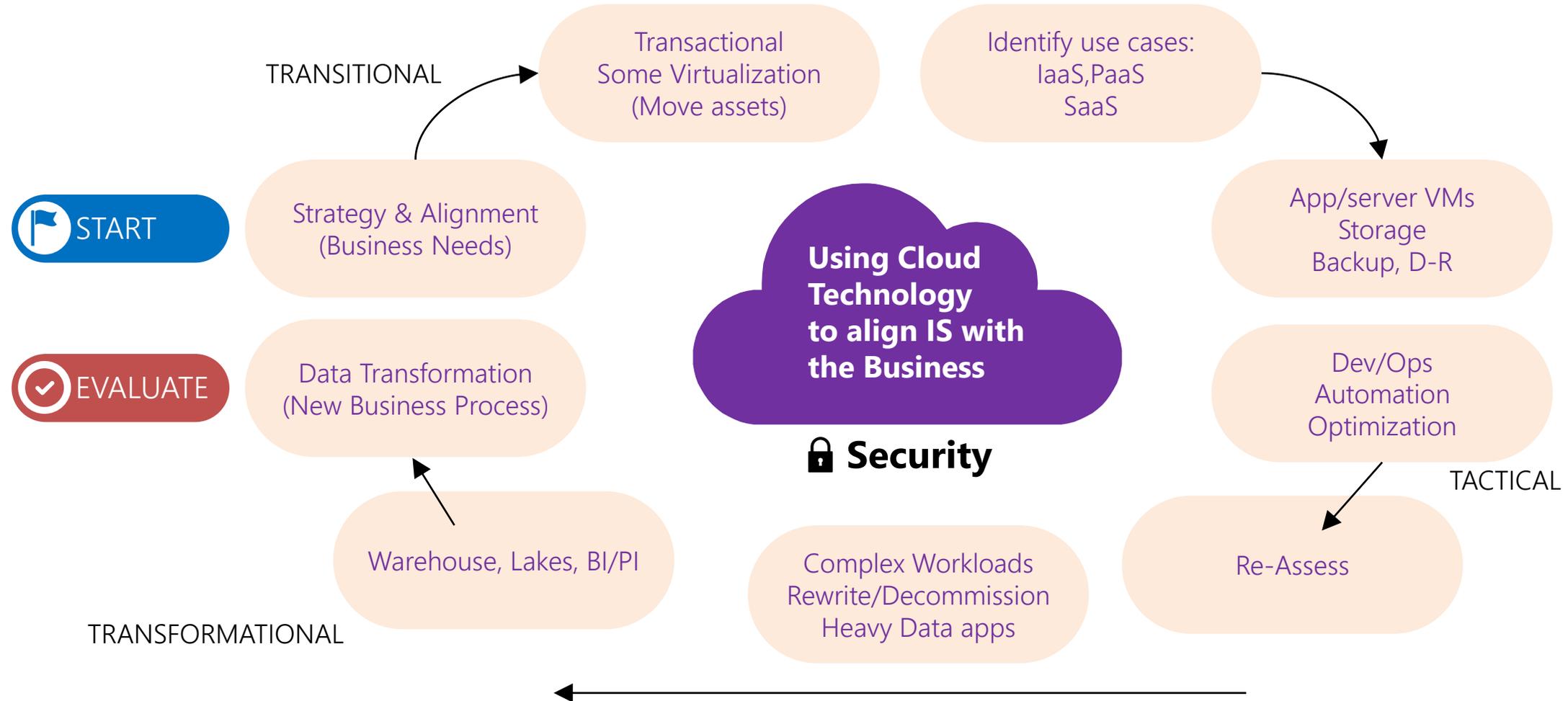
Packaged Applications to VMs (Elasticity)
Budget constraints
Backup and D-R
Co-Location HW reduction
Dev-Ops
Mobile Applications
Mobility and Access

HYBRID CLOUD

THE CLOUD JOURNEY – THE VIRTUOUS CIRCLE



Aligning IS to Business using Cloud Technology



Key Components:

- Scaling & High Availability or HA
- Security
- Cost Optimisation
- Design Practices to be considered

Key Component #1: Scaling

A main benefit of using IaaS and a Managed PaaS is scaling both vertically (IaaS) and horizontally (IaaS & PaaS)

Scaling Services vertical and horizontal

-ELB, Auto Scaling, EC2 tools, Resizing EBS or storage

DB Scaling in real time or during maintenance window, Managed services patches, OS etc

Automated Deployments

-Cloud Formation, AMIs (golden, hybrid), JSON



Key Component #1: Scaling

A main benefit of using IaaS and a Managed PaaS is scaling both vertically (IaaS) and Horizontally (IaaS & PaaS)

- ❖ Design for failure – the Netflix approach (largest user of S3 in the world)
 - ❖ Each component should be built for failover
 - ❖ Design to scale with an increase in load – this means a stateless back-end
- ❖ Automated Deployments
 - ❖ CloudFormation /OpsWorks
 - ❖ Beanstalk for PaaS (smaller websites)
 - ❖ AMIs customized and hybrid
 - ❖ Backup AMIs in case of a hack
- ❖ Multi-zone and Region set up
 - ❖ ELB and Auto-scaling
 - ❖ RDS with Multi-AZ
- ❖ Use Scalable Services
 - ❖ ELB, Auto Scaling, Cloudwatch
 - ❖ Part of HA and Resiliency



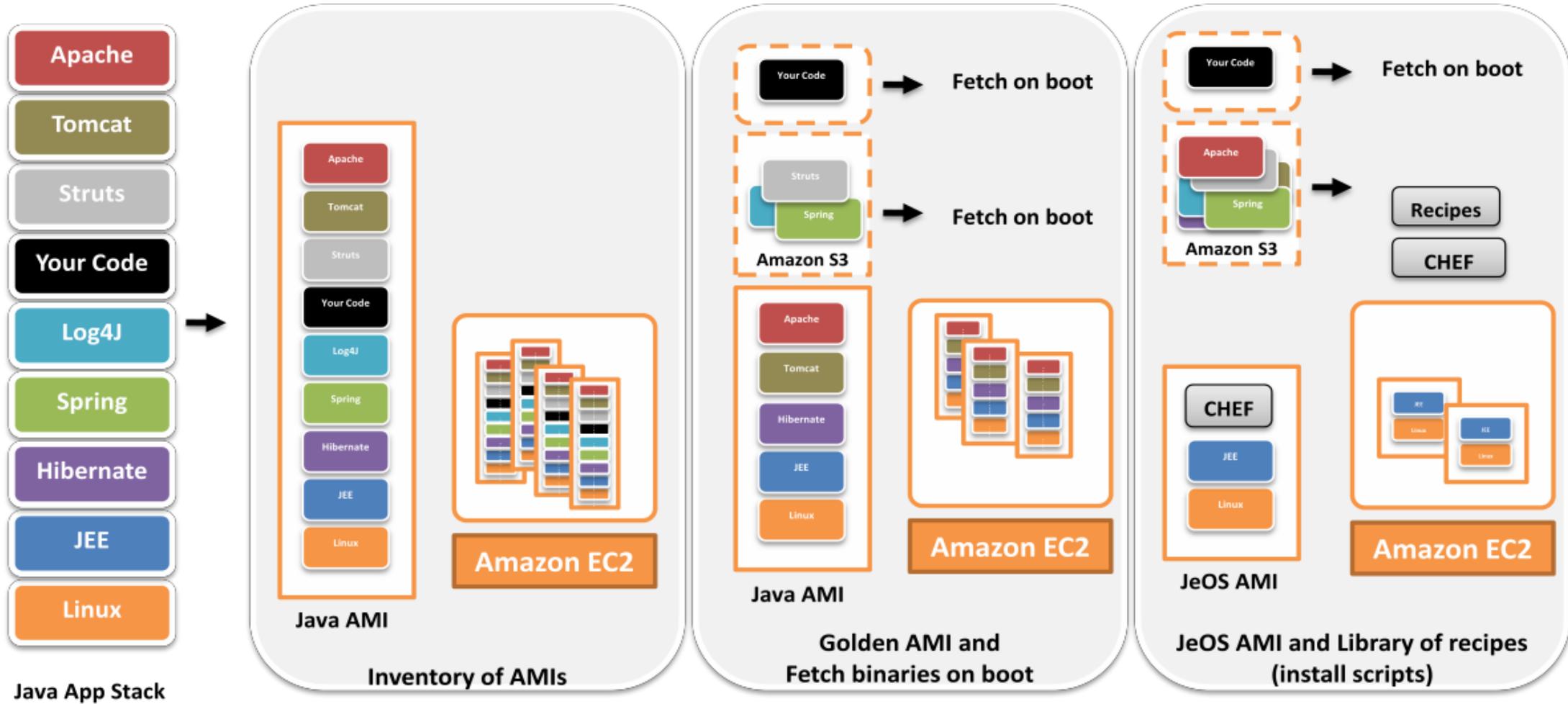
AWS Architectural Practices



AWS cloud

Key Component #1: Scaling

AMIs



Key Component #1: Scaling

A main benefit of using IaaS and a Managed PaaS is scaling both vertically (IaaS) and horizontally (IaaS & PaaS)

❖ Application Design

- ❖ Stateless Apps
- ❖ Decoupled Apps and logic
- ❖ Session Management of end user connection
- ❖ Reduce app load through CDN or Caching

❖ DB Scaling

- ❖ Read-Replicas
- ❖ PIOPS
- ❖ Aurora or Dynamo

❖ Capacity management

- ❖ Use alerts on capacity usage (memory, storage, CPU) manual or automatic re-configuration of scale
- ❖ Pay for services used on demand

❖ EC2 Performance

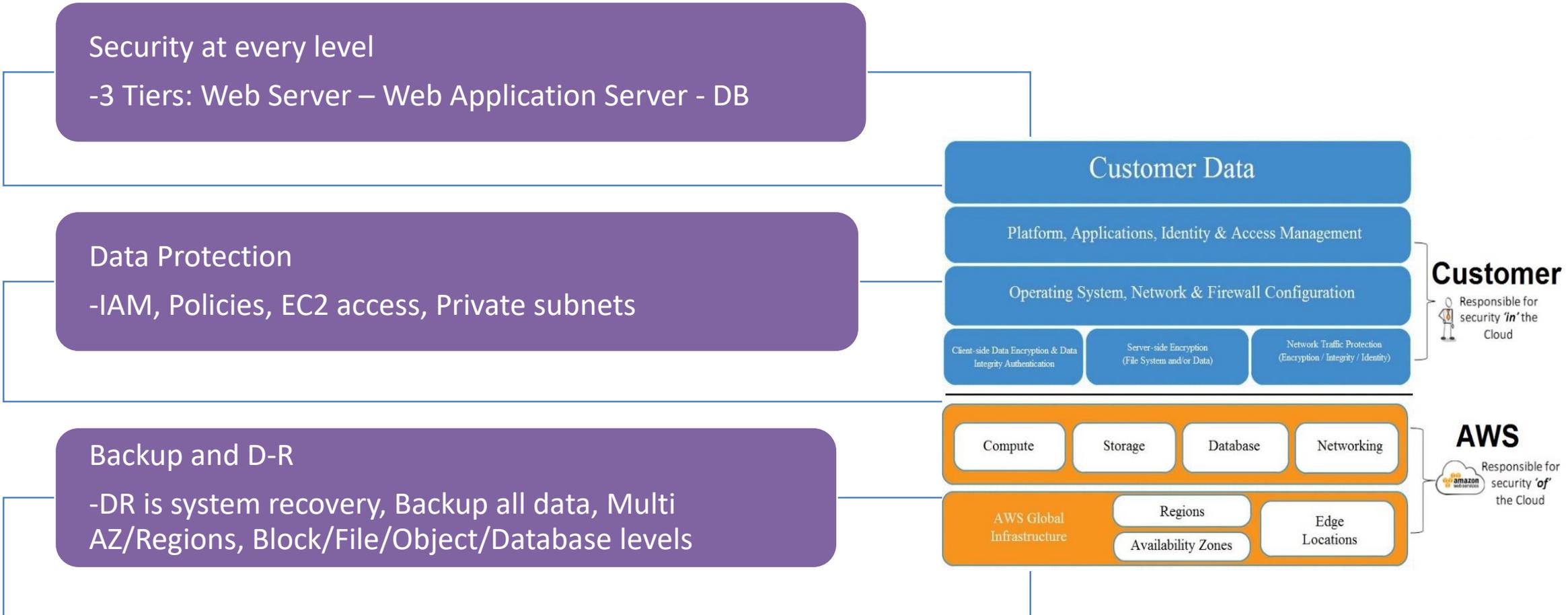
- ❖ Instance size & IOPS
- ❖ EBS Optimized instances
- ❖ Use EC2 Optimisation tools (new)

AWS Architectural Practices



Key Component #2: Security

AWS is a shared security model, you are responsible for your data.



Key Component #2: Security

❖ IAM

- ❖ Policies for users, groups and roles
- ❖ Fundamental to AWS usage



❖ Every Tier or Level is secured

- ❖ VPC: Security group, private subnet, ACL, route tables
- ❖ EC2: Key-pair, security group
- ❖ S3: Policy-bucket/object based
- ❖ MFA



❖ Data

- ❖ Encryption at rest – S3, EBS, EBS snapshots, RDS, RDS snapshots
- ❖ SSL encryption in transit



❖ Backup and D-R

- ❖ Snapshots, Multi-zone/Region, Copy to another account, Backup each service

❖ Version control

- ❖ AMIs, S3



❖ Traceability

- ❖ Cloudtrail (api level), Logs for S3 & ELB, OS level logs

Key Component #3: Cost Optimisation

Pay as you go, service instantiated is charged (unless free)

Free Tier

-AMIs, RDS, EC2 micro

Use Managed Services

-IaaS and PaaS automatically managed for you

Scale on demand

-Reduces costs, use monitors, alarms and trusted advisor

Key Component #3: Cost Optimisation

- ❖ Pay as you go
 - ❖ Opex
 - ❖ Provision capacity as needed, scale some services as needed
- ❖ Use the Free Tier
 - ❖ Learn AWS with Free Tier services
 - ❖ Use them to build a Demo and POC
- ❖ Transparent pricing
 - ❖ AWS billing updated every few hours
 - ❖ Tag your resources to organize the billing
- ❖ Automate
 - ❖ Inspector will offer advice on security, instance size, provisioning, cost reductions
 - ❖ Set billing alarms with Cloud Watch
 - ❖ Look at Reserved or spot pricing
 - ❖ Use Auto-Scaling if load variations (scaling down as well to save money)



Amazon Inspector

Design Best Practices

- ❖ Always Design to scale Horizontally not Vertically
 - ❖ Saves money
 - ❖ Provision capacity as needed, scale some services as needed
 - ❖ Introduce Redundancy (N+1)
- ❖ Application Architecture
 - ❖ Stateless and Loosely Coupled
 - ❖ REST/SOA, Webservices, Gateway
- ❖ Automate Deployment
 - ❖ Configure Bootstrapping, use Cloud Formation
 - ❖ Use an AMI instead of configuring software in production systems
- ❖ Dev Ops
 - ❖ Keep Production static ie don't change with patches on the go
 - ❖ Everything goes through DevOps testing
 - ❖ Automate this with Chef, Puppet, Cloud Formation
- ❖ Use DBaaS to scale
 - ❖ Select RDS, NoSQL as needed
 - ❖ Migrate your data and application 1:1 using DMS

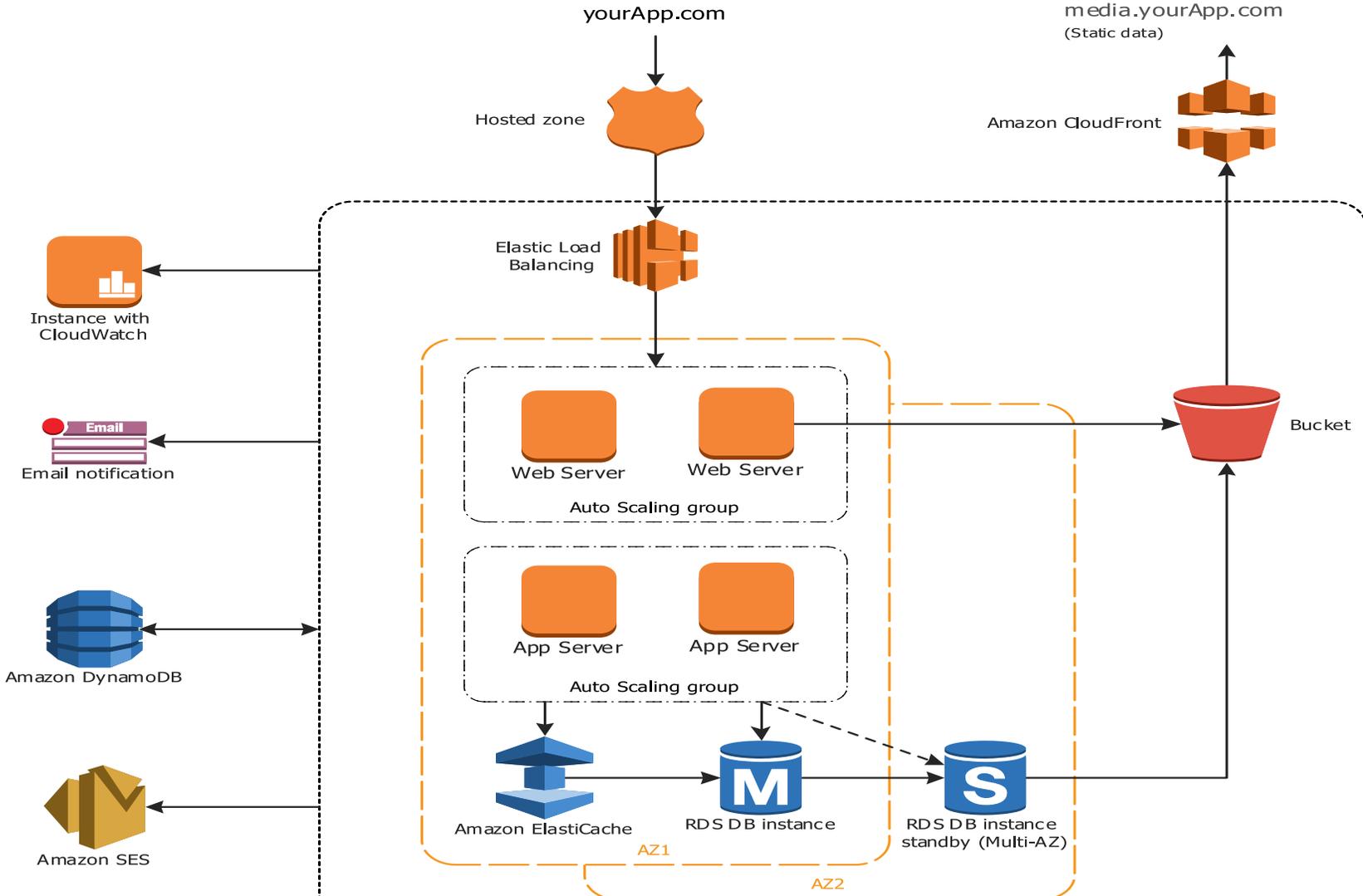
Design Best Practices

- ❖ Reduce Database Load
 - ❖ Read Replicas to read data only
 - ❖ Can use Web A.S. caching techniques
- ❖ Security
 - ❖ All levels, must be built into design from the beginning
- ❖ Manage the Costs
 - ❖ You likely won't save that much in year 1 vs on premise or co-location
 - ❖ As you become more familiar with AWS, each year should see more usage, better productivity, and many benefits vs the legacy infrastructure
 - ❖ Vigilance is mandatory as you use the system and understand the cost metrics
- ❖ Dev Ops
 - ❖ Professionalizes software, IT deployment
 - ❖ Training costs can be substantial, need to be budgeted
 - ❖ It is culture + automation

AWS Architectural Practices



Design Best Practices – Example of HA



AWS Architectural Practices



Design Best Practices – Example of HA

