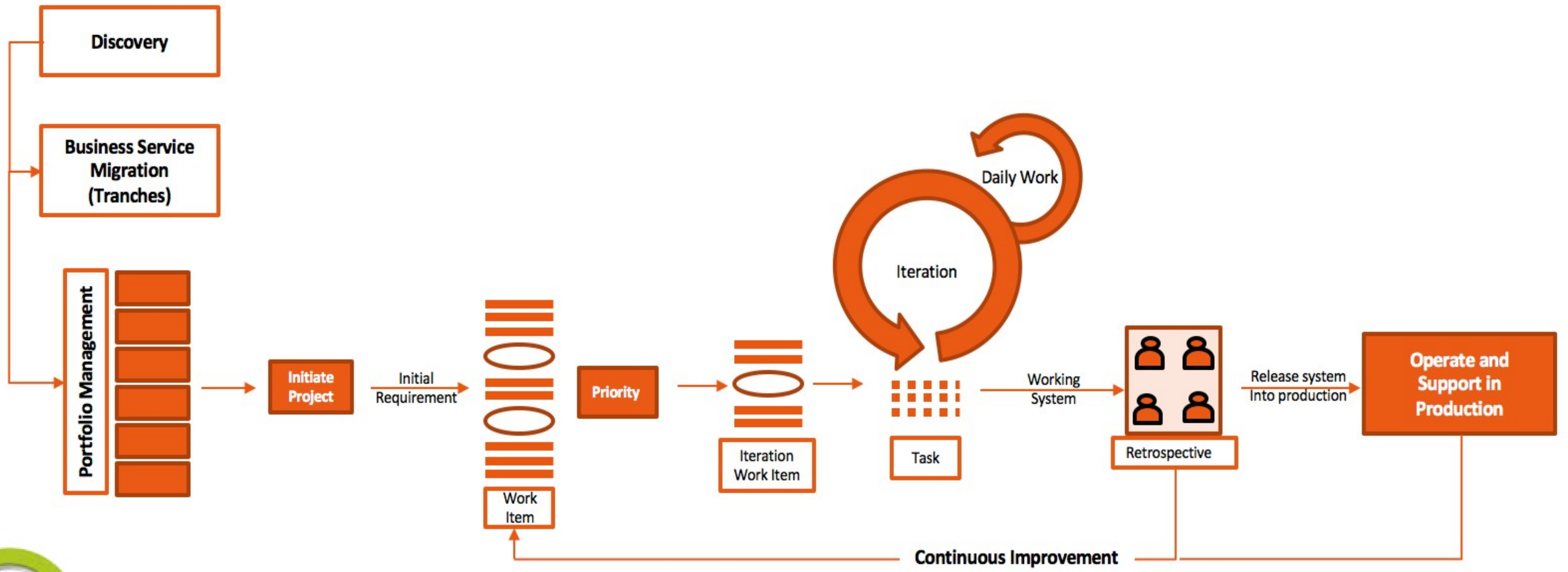
Large complex system deployment into AWS / AMS

Main concepts summarized

AMS = Amazon Managed Services a new 2017 available MSP model for AWS cloud

Contents

Process LifeCycle (PLF) Overview



Inception 0	Construction Iterations 1	Transition 2	Production 3	Decommission 4
• Project Plan • PIDD • E2E Design • Data Protection • Info & App Security • Test Strategy • Migration Approach	• Build & Test-Sprints • Transition Readiness • Implementation Change Request	• Release/Cutover • Go No Go • Service Overview	• Project Closure	• Decommission

Concepts

- Architecture that uses best practices (baked in)

- Has to be done within a framework High Level Design (HLD) but:
- Also by migration path and project (bottoms up)

- SSO built in from the beginning – need a ‘bottoms up’ view, not only a SWAG view (SME ‘I think that is right because…..well I think so)

- Scope is King (One Document to Rule Them All)

BP Owner for design……

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 deployments)
  - 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

Amazon RDS

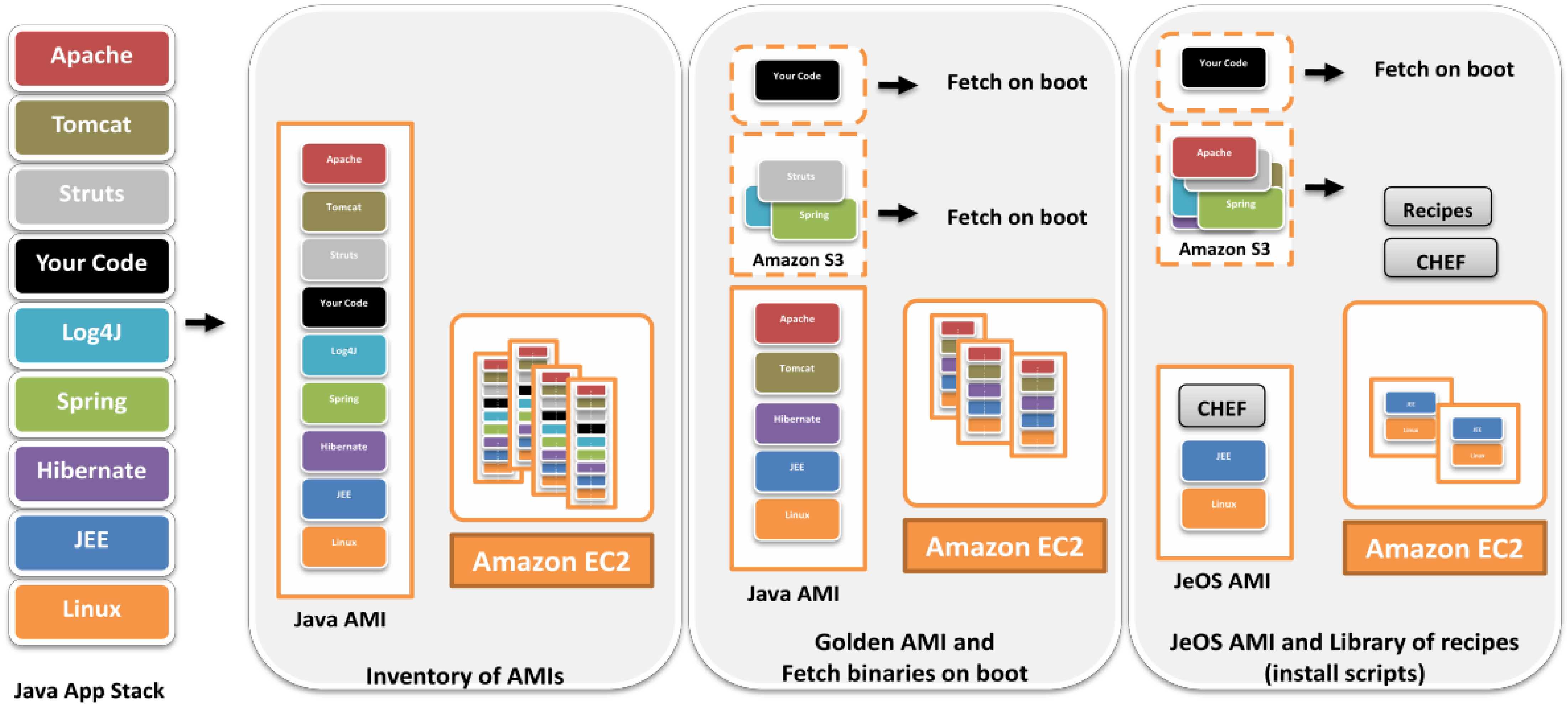
RDS DB instance

AMI

Elastic Load Balancing

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 – usually done post migration
  - Stateless Apps
  - Decoupled Apps and logic
  - Session Management of end user connection
  - Reduce app load through CDN or Caching

- DB Scaling
  - Read-Replicas
  - PIOPS (provisioned input output per second EBS volume)
  - 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)

Key Component #2: Security

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

Security at every level

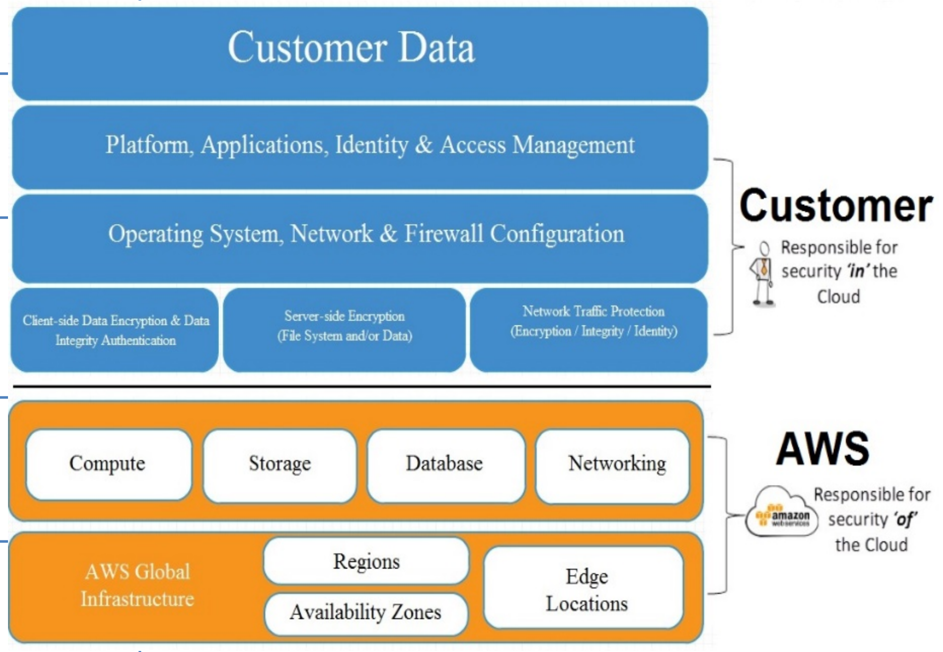
-3 Tiers: Web Server – Web Application Server - DB

Data Protection

-IAM, Policies, EC2 access, Private subnets

Backup and D-R

-DR is system recovery, Backup all data, Multi AZ/Regions, Block/File/Object/Database levels


Customer Data
Platform, Applications, Identity & Access Management
Operating System, Network & Firewall Configuration
Client-side Data Encryption & Data Integrity Authentication
Server-side Encryption (File System and/or Data)
Network Traffic Protection (Encryption / Integrity / Identity)
Customer
Responsible for security 'in' the Cloud
Compute
Storage
Database
Networking
AWS Global Infrastructure
Regions
Availability Zones
Edge Locations
AWS
Responsible for security 'of' the Cloud

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

AWS IAM

Internet gateway

Virtual private cloud

Data encryption key

Amazon S3

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
    - SSO as it is used today (assuming SSO into email, shared files etc)

- 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
    - Use the free tier to test, play
    - 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
    - Jenkins, Chef, OpsWorks – use these

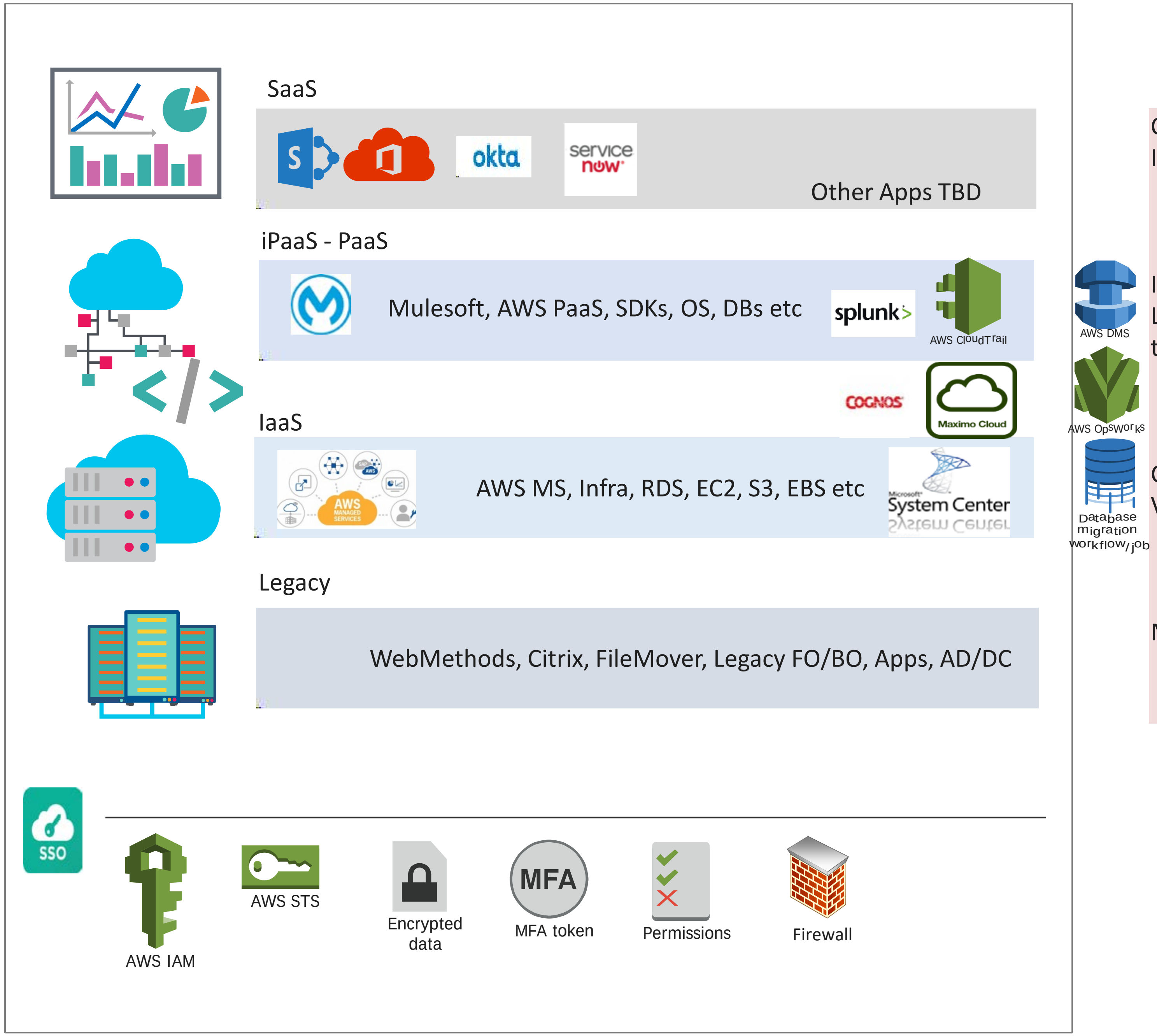
HLD and Assumptions

- Project Scope document TBC including migration processes end to end
- We need a Best Practices Owner (previous slides) to ensure BP inside the architecture + migrations
- Security TOM to be supported by migration, apps models
- End to End flows need to be worked out re **SSO** and role access eg O 365/Workspaces
- AWS MS vs Customer MS
    - (MS = Managed Services) (MSP = Managed Services Provider)

Example within a Tranche: Tranche 1

- Infrastructure first
- Then simple business apps second with the knowledge that;
- Each is a project with a defined scope and requirements

Platforms

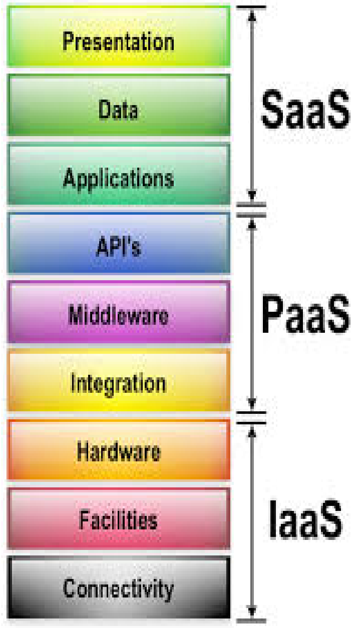


On Demand, Identify Mgmt ITSM

Integrate, Track, Monitor, Log, Provisioning, DevOps tools/automation

Compute, Storage, Network, VPC, Monitor

Migrate, Integrate, Hybrid



Platforms

1) AWS/AMS Account and VPC(s)

- Sandbox
- Dev-Test-Prod (with HA)
- DevOps tools + logging (Splunk w CT, CW etc)

2) AWS MS

- Dev-Test-Prod (with HA DR) + above
- Peering with 1)
- Use Workspaces within AMS to do DevOps

3) Mulesoft – Middleware, Data/Apps

4) OKTA – ID management with AD

5) ServiceNow – ITSM, Workflow Resource Approval

6) O 365 SP, MX, SCCM …………

SSO

Permissions

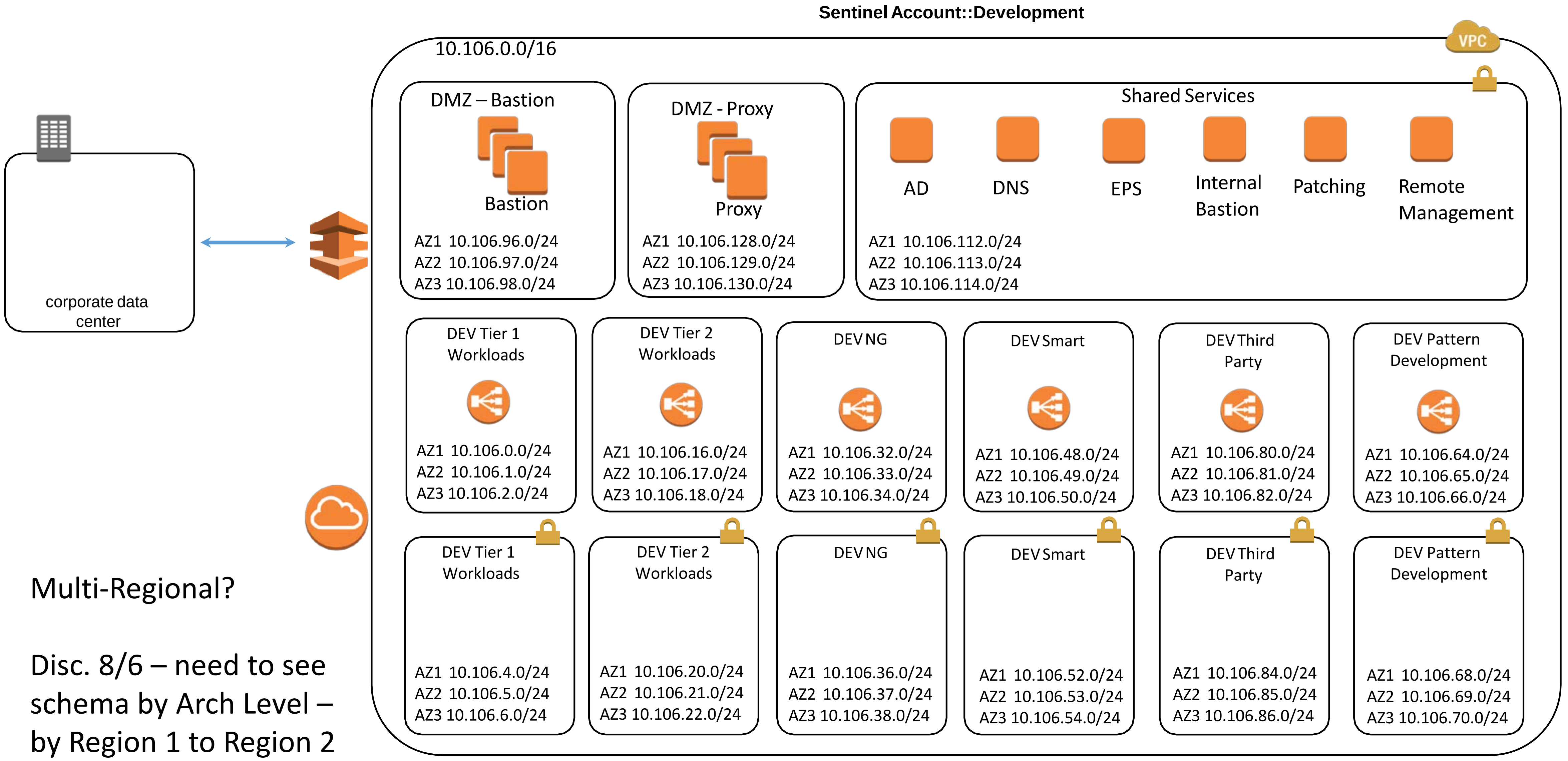
HBR.ORG
Harvard Business Review
SEPTEMBER 2011
46 The Big Idea
Three Myths About Health Care Exploded
Robert S. Kaplan and Michael E. Porter
123 Managing Yourself
How Great Bosses Engage Their Employees
Charalambos A. Vlachoutsicos
41 How I Did It
EBay's Founder on Innovative Social Change
Pierre Omidyar
EMBRACING COMPLEXITY
You can't avoid it, but your business can profit from it.

MFA

MFA token

AWS IAM

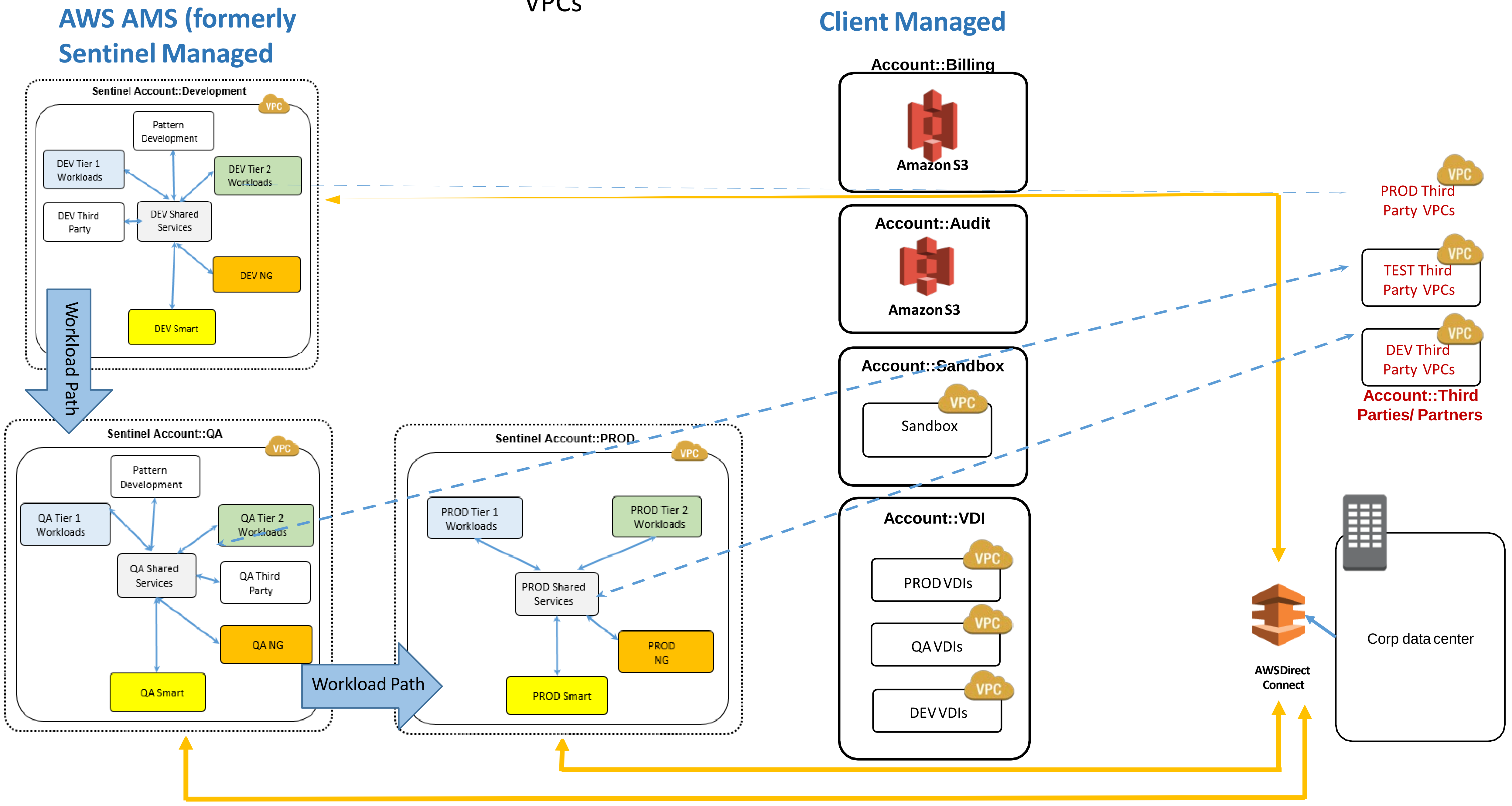
VPCs



Multi-Regional?

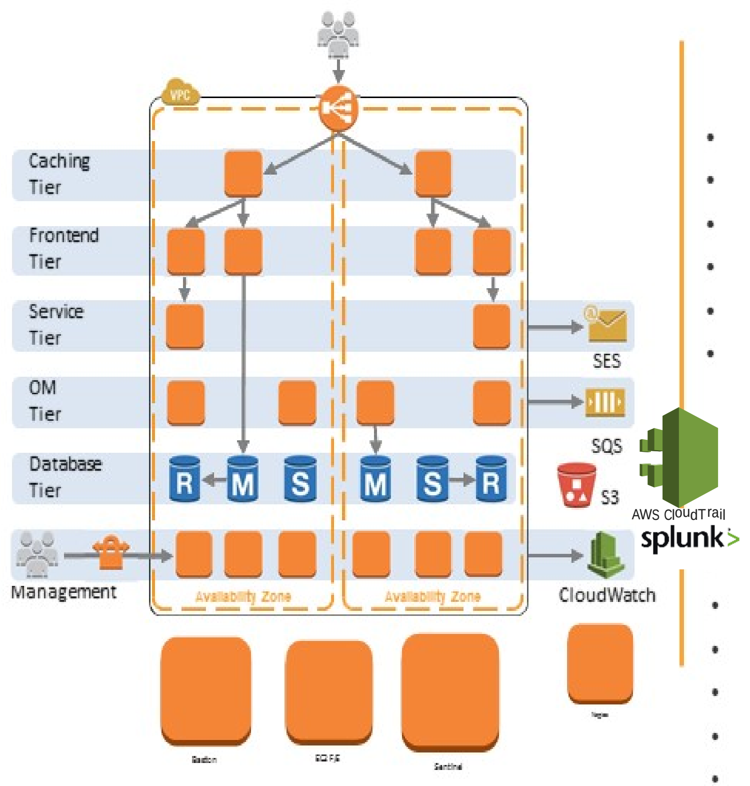
Disc. 8/6 – need to see schema by Arch Level – by Region 1 to Region 2 showing A-A; A-P; P-P

VPCs


AWS AMS (formerly Sentinel Managed
Client Managed
Sentinel Account::Development
VPC
Pattern Development
DEV Tier 1 Workloads
DEV Tier 2 Workloads
DEV Third Party
DEV Shared Services
DEV NG
DEV Smart
Workload Path
Sentinel Account::QA
VPC
Pattern Development
QA Tier 1 Workloads
QA Tier 2 Workloads
QA Shared Services
QA Third Party
QA NG
QA Smart
Workload Path
Sentinel Account::PROD
VPC
PROD Tier 1 Workloads
PROD Tier 2 Workloads
PROD Shared Services
PROD NG
PROD Smart
Account::Billing
Amazon S3
Account::Audit
Amazon S3
Account::Sandbox
VPC
Sandbox
Account::VDI
VPC
PROD VDIs
VPC
QA VDIs
VPC
DEV VDIs
VPC
PROD Third Party VPCs
VPC
TEST Third Party VPCs
VPC
DEV Third Party VPCs
Account::Third Parties/ Partners
AWS Direct Connect
Corp data center

3 Tiers+ within VPC MS

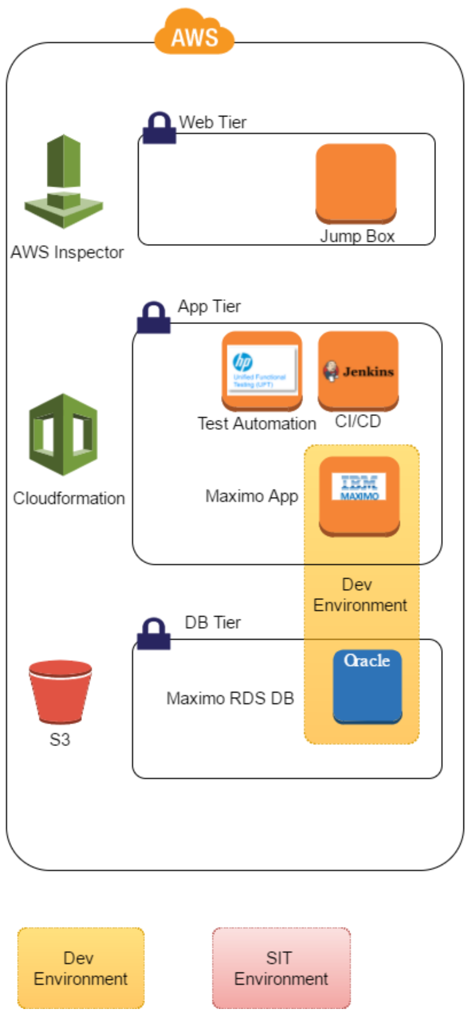
Layered

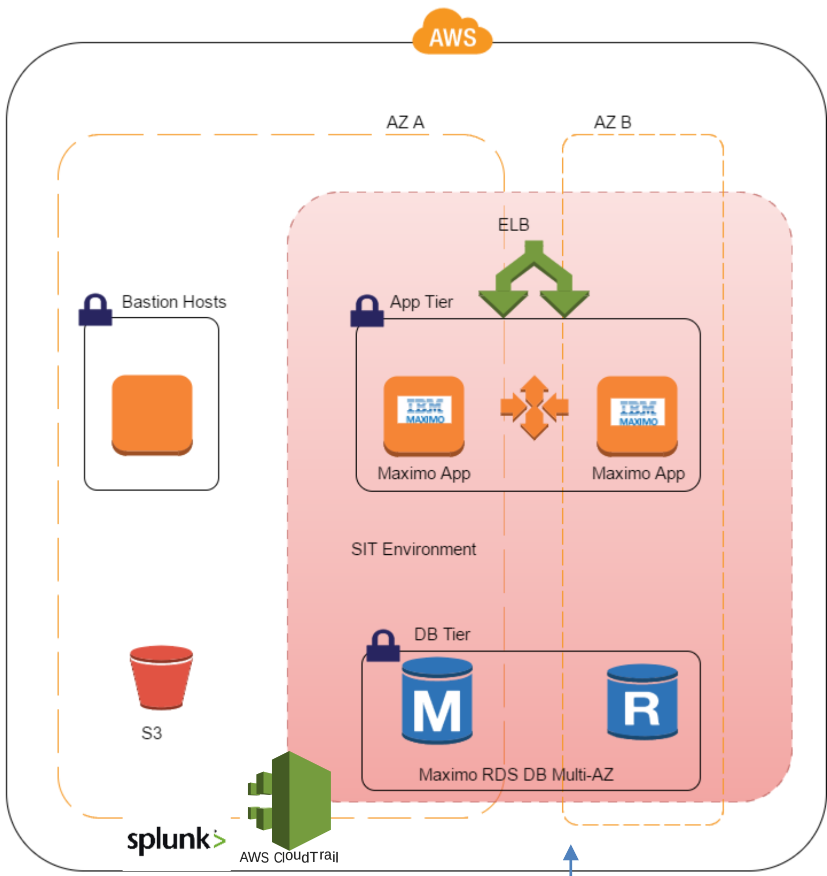


- Multi AZArchitecture
- X# - EC2
- X# - RDS
- X# - S3 Objects
- Chef – Jenkins deployment
- Middleware: MuleSoft

Management Servers

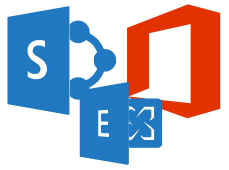
- Bastion
- Ec2 Front End Admin
- Ec2 Nagios+CloudWatch eg
- SFTP
- Sentinel



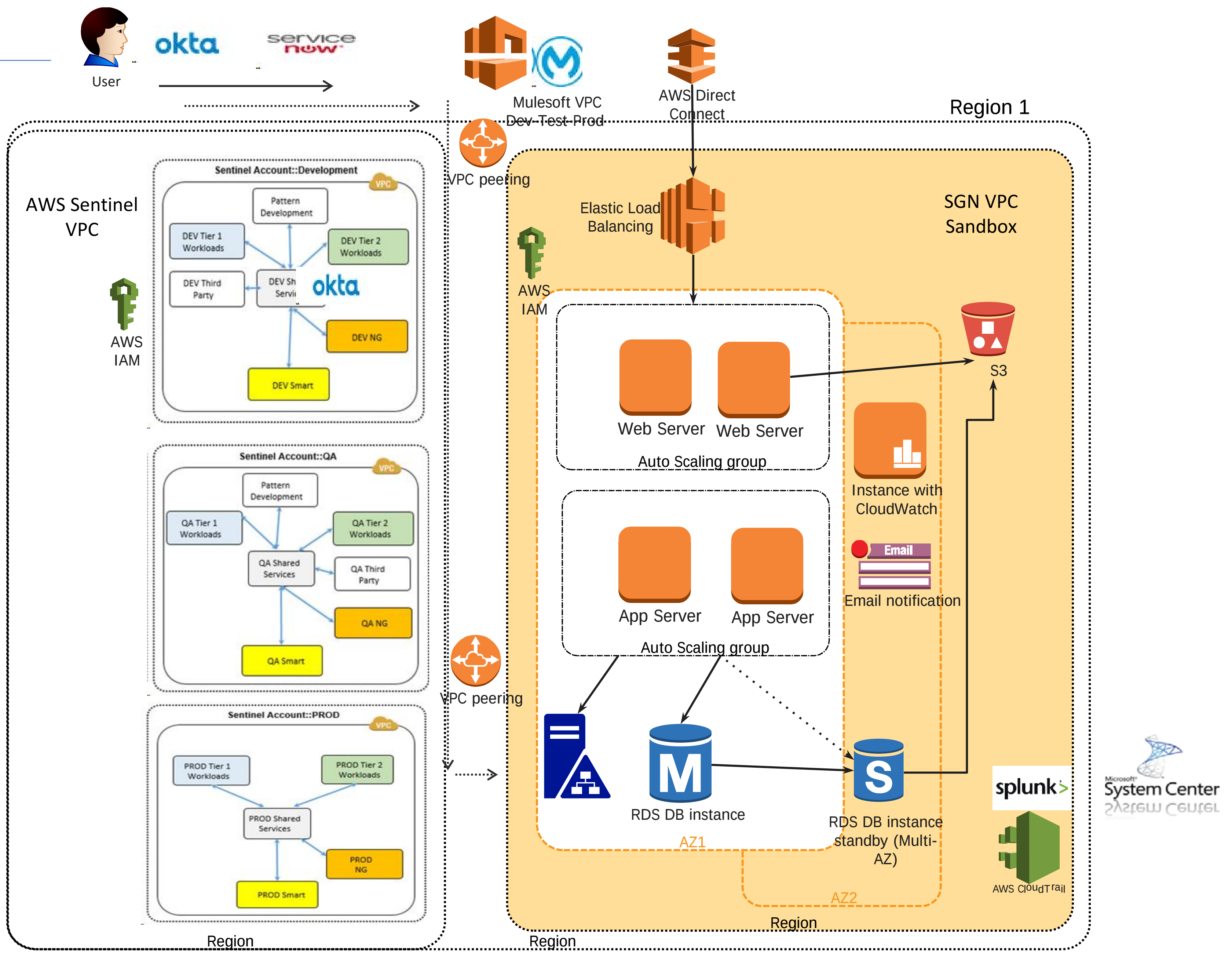


For each Project: R-POC
EC2 vs RDS
HA ,DR, Backup
Security
Sprawl, Cost control
Logs, Audits…….

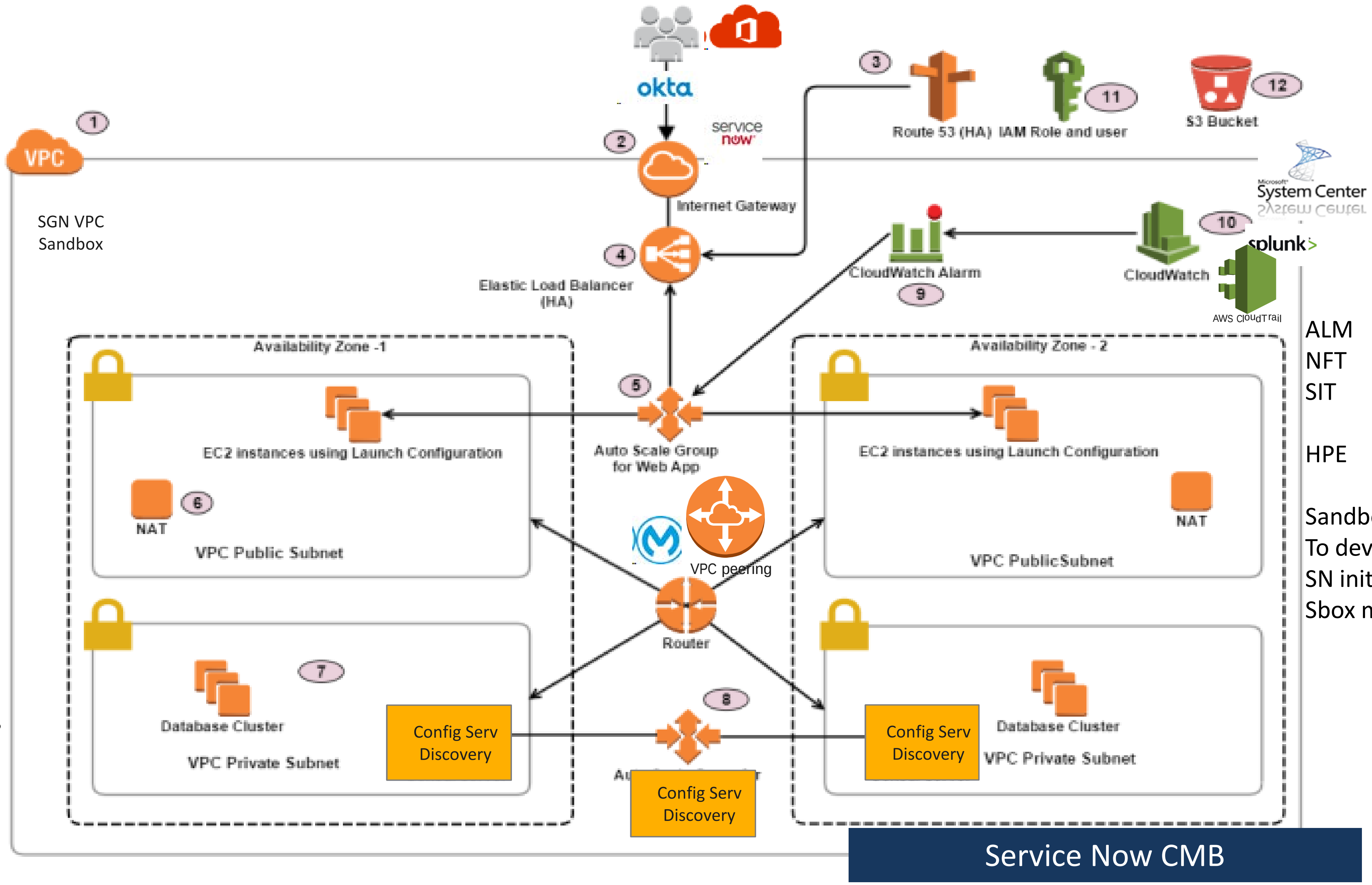
MS Complexity



IAM
Integrated Testing
VPC security
HA DR Backup
SSO
Sprawl
Cost control
Logs
Audits
.......SCCM



VPC flows



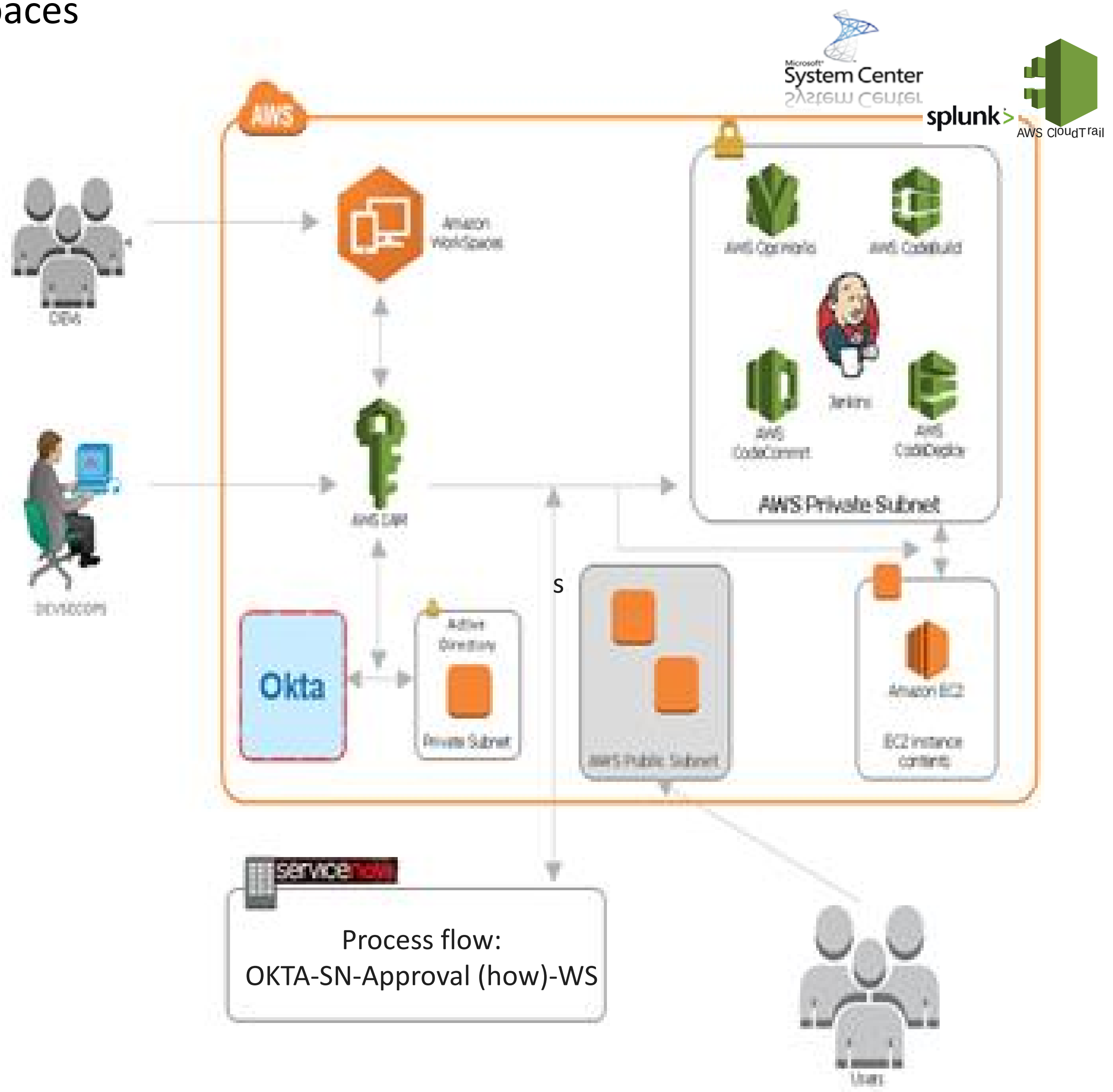
IAM
VPC security
HA DR
EC2 RDS
SSO
Sprawl
Cost control
Logs
Audits
Splunk+SCCM logs
(tbd)

ALM
NFT
SIT

HPE

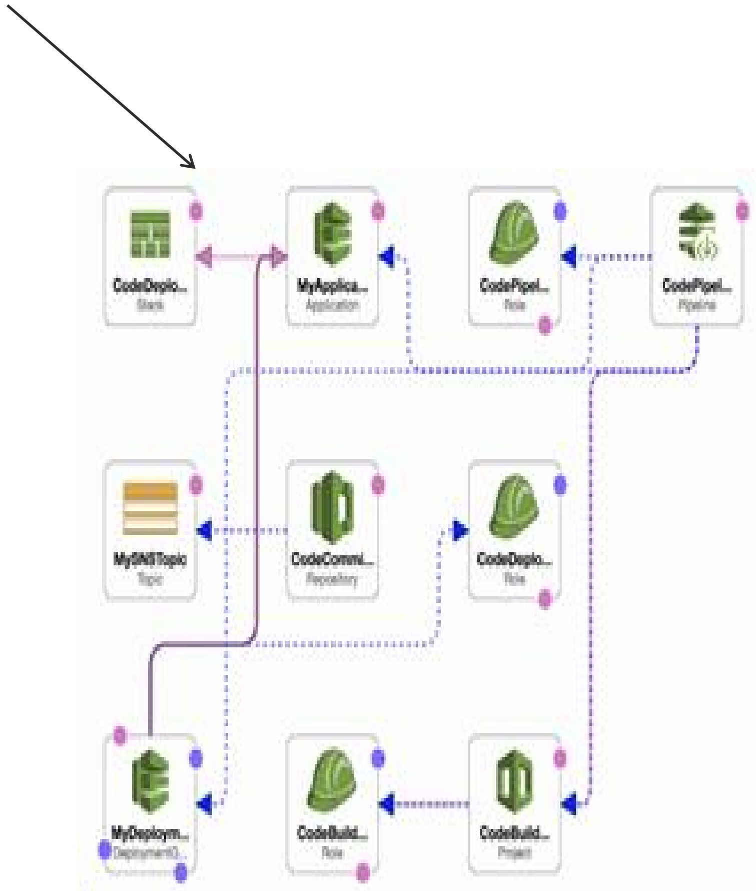
Sandbox
To dev/test
SN initiate after dep
Sbox migration

Workspaces

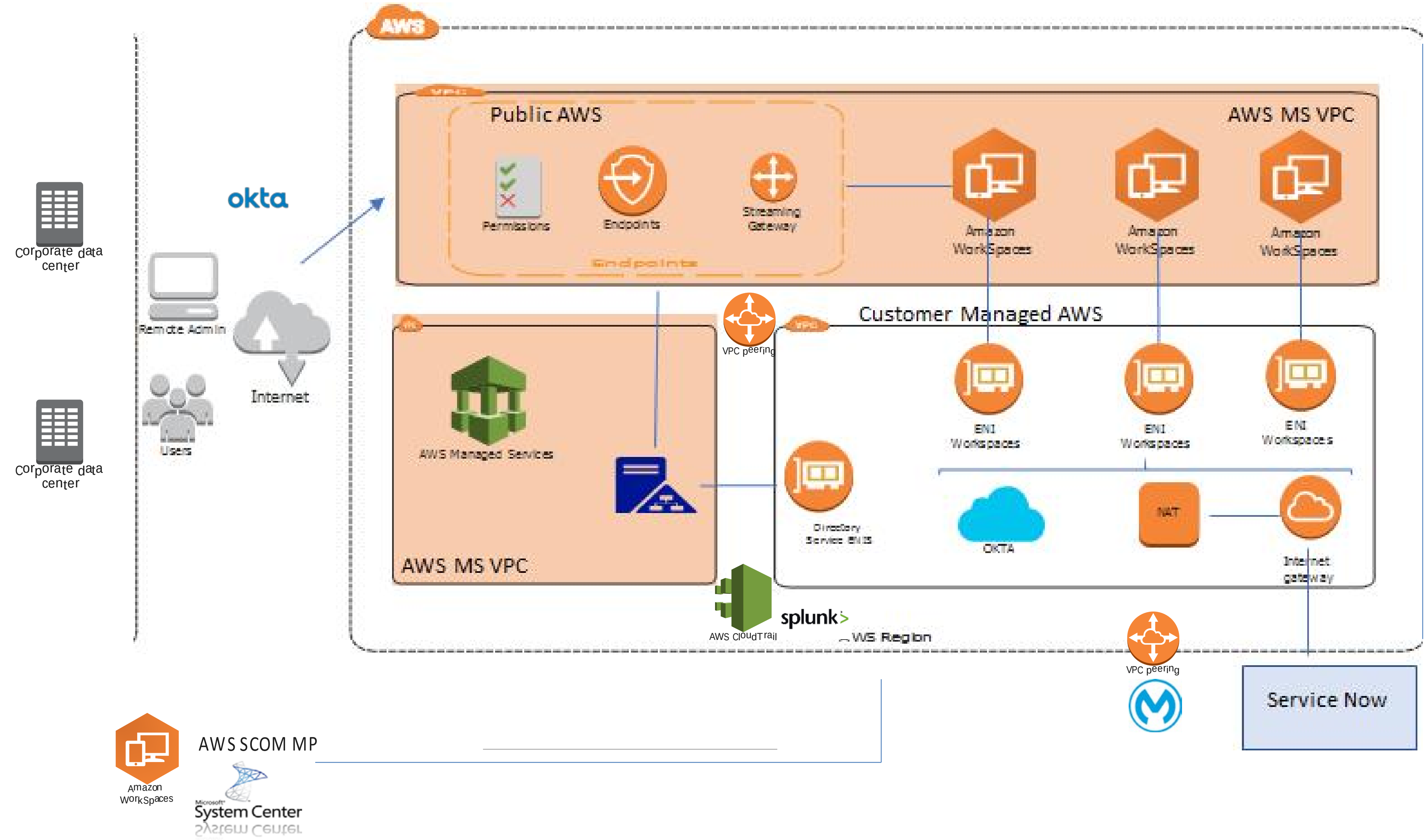


Process flow:
OKTA-SN-Approval (how)-WS

migration, patches same process;
SN-Jenkins-SR-New AMI-Code Deploy-Validate-RFC create-Encrypt-Into Catologue as latest AMI

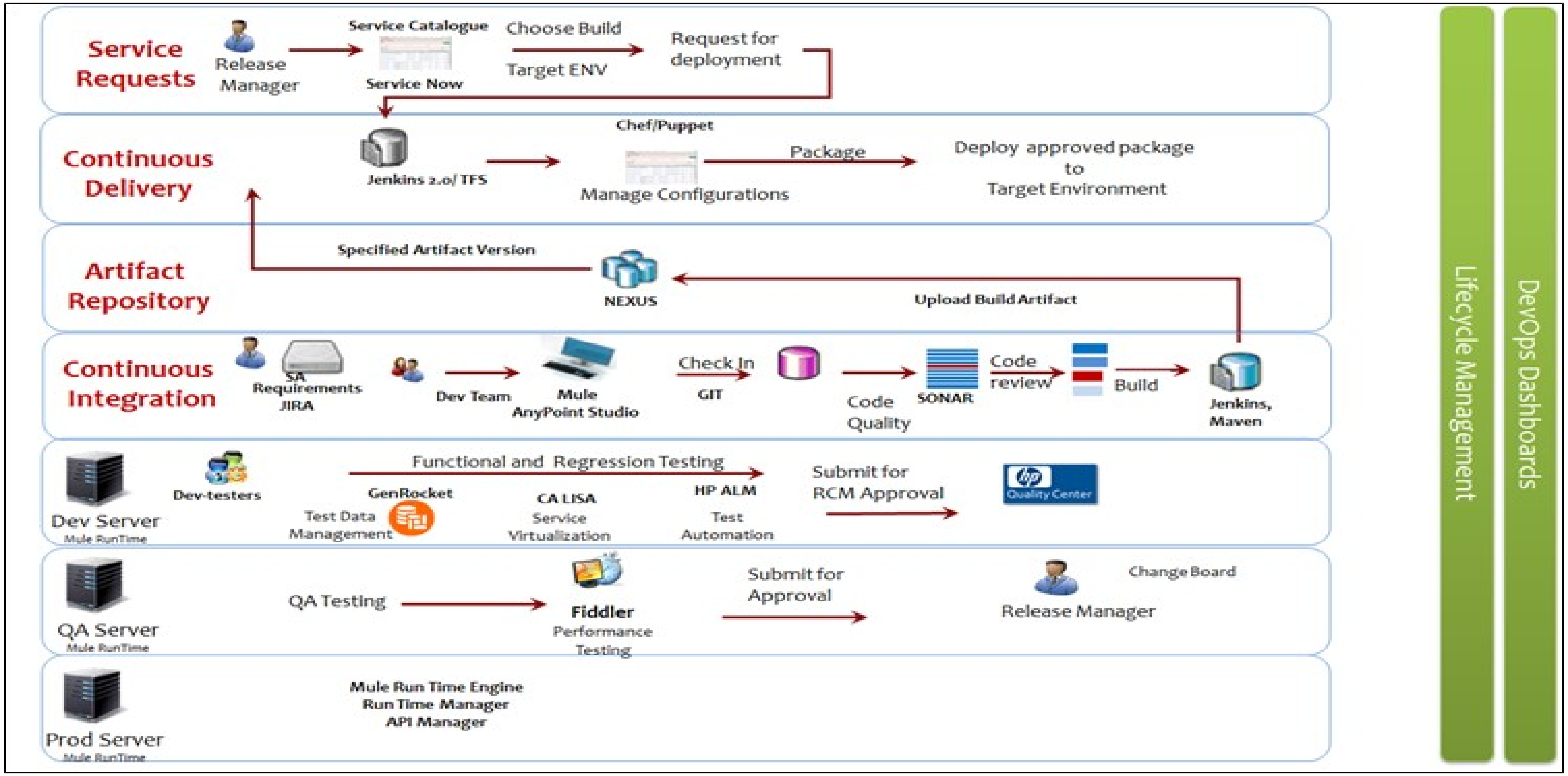


Workspaces



Nexus
To code deploy
Shared vs Specific Resources
OKTA
SN
MS

Swim lanes for CI/CD


Service Requests
Release Manager
Service Catalogue
Service Now
Choose Build
Target ENV
Request for deployment
Continuous Delivery
Jenkins 2.0/ TFS
Chef/Puppet
Manage Configurations
Package
Deploy approved package to Target Environment
Artifact Repository
Specified Artifact Version
NEXUS
Upload Build Artifact
Continuous Integration
SA Requirements JIRA
Dev Team
Mule AnyPoint Studio
Check In
GIT
Code Quality
SONAR
Code review
Build
Jenkins, Maven
Dev Server
Mule RunTime
Dev-testers
Functional and Regression Testing
GenRocket
Test Data Management
CA LISA
Service Virtualization
HP ALM
Test Automation
Submit for RCM Approval
Quality Center
QA Server
Mule RunTime
QA Testing
Fiddler
Performance Testing
Submit for Approval
Release Manager
Change Board
Prod Server
Mule RunTime
Mule Run Time Engine
Run Time Manager
API Manager
Lifecycle Management
DevOps Dashboards