WEBINAR

VERSION 1.1

Maximizing your cloud potential through Modernization

Today’s Agenda

- What is Modernization
- Modernization Inertia
- Risks with not modernizing
- Modernization methods and pathways
- Organization and Culture
- Worked example

WEBINAR

What is it?

Modernization looks at transforming existing applications and infrastructure to leverage **cloud-native services** to enable organizations to **deliver value sooner, safer and with higher quality.**

Modernization Inertia

Inertia (noun) - a tendency to do nothing or to remain unchanged

Perceived Risks

Resistance due to concerns around lock-in, security risks, increased costs, etc.

Complexity

Preference to keep things as-is due to the complexity of current systems.

Bureaucracy

Excessive corporate “red tape” making effecting change challenging.

Lack of expertise

Lack of knowledge of how to drive and execute modernization activities.

Complacency

Overconfidence due to current reputation or past success.

Failure to modernize brings risks

Legacy systems

Longer innovation cycles

Inability to compete

Loss of talent

Erosion of market share

Characteristics of Modern Applications

Cost savings

Reducing compliance risk

Business Agility

Increased developer productivity

Operational Resilience

Greater Observability

Characteristics of Modern Applications

Cost savings

Reducing compliance risk

Business Agility

Increased developer productivity

Operational Resilience

Greater Observability

Characteristics of Modern Applications

Cost savings

Reducing compliance risk

Business Agility

Increased developer productivity

Operational Resilience

Greater Observability

Characteristics of Modern Applications

Cost savings

Reducing compliance risk

Business Agility

Increased developer productivity

Operational Resilience

Greater Observability

Characteristics of Modern Applications

Cost savings

Reducing compliance risk

Business Agility

Increased developer productivity

Operational Resilience

Greater Observability

Characteristics of Modern Applications

Cost savings

Reducing compliance risk

Business Agility

Increased developer productivity

Operational Resilience

Greater Observability

Characteristics of Modern Applications

Cost savings

Reducing compliance risk

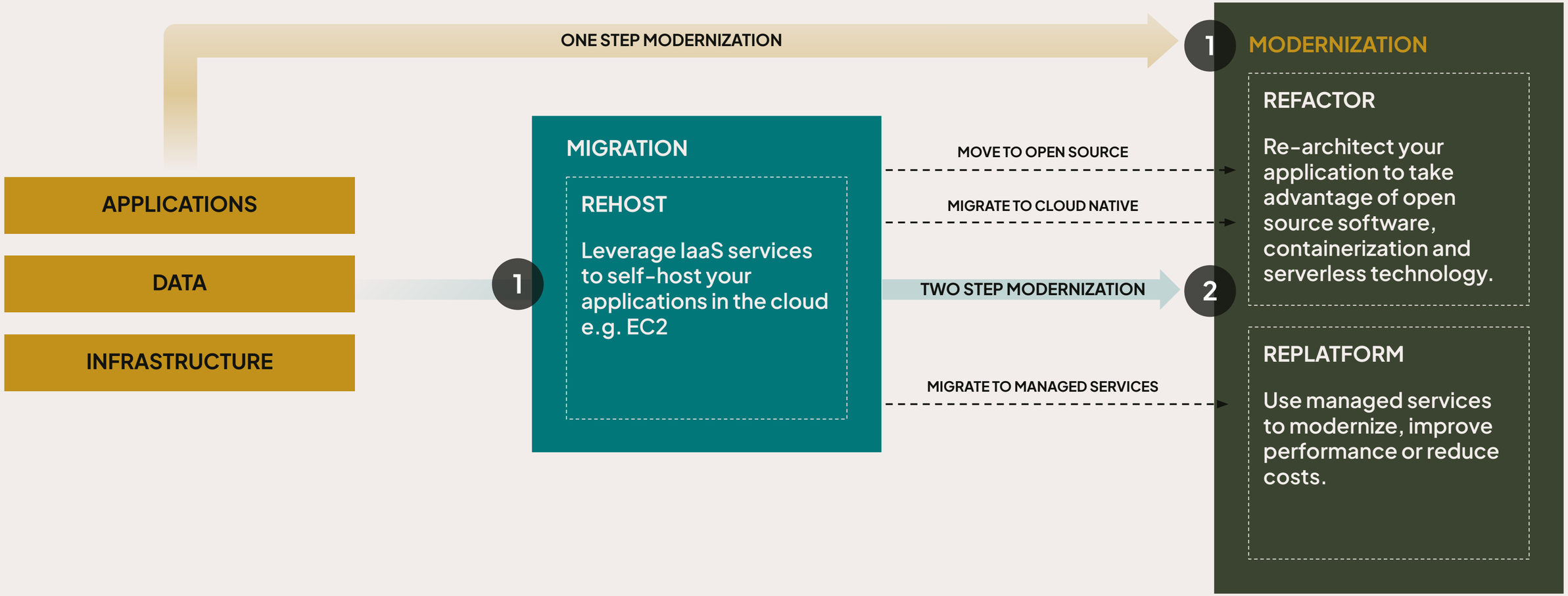
Business Agility

Increased developer productivity

Operational Resilience

Greater Observability

One-step vs Two-step modernization



Modernization Pathways

REPLATFORM

Containerize

Enable portability of your applications. Fully utilizing automated build and deployment tools

Managed Services

Leverage Cloud Managed Services to reduce operational heavy lifting.

REFACTOR

Open Source

Switch to open source software for cost savings and extensive community support

Serverless

Leverage serverless technologies to fully exploit the pay-as-you-go cost model

Re-architect

Redesigning your monolith applications to a loosely coupled software architecture to enable independent product teams to deliver faster

28%

Increase in organization revenue by moving from VMs to Containers

40%

Reduction in downtime by moving to containers

24%

Reduction in time to deploy features with Managed Databases

39%

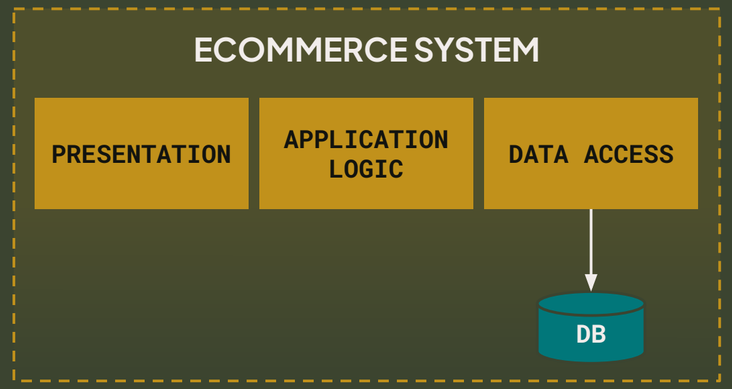
Reduction in IT infrastructure spend by using Serverless technology.

Source: Business Value of Cloud Modernization, Jan 2023, by Known Research (Commissioned by AWS), 505 Enterprises Surveyed - Whitepaper

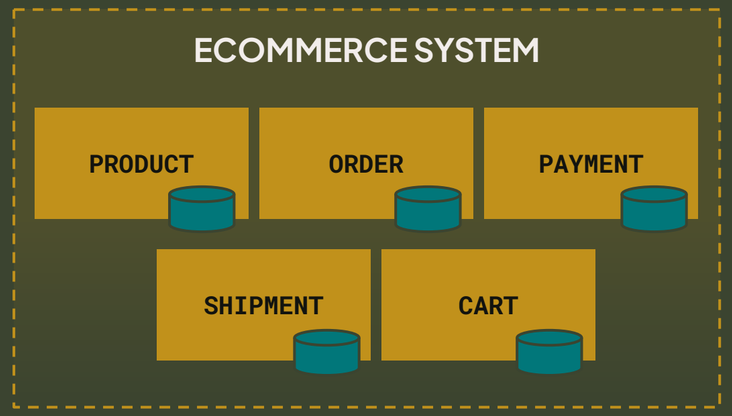
Re-architecting your software architecture to enable fast flow

Migrating a monolithic application where all modules are bundled together as a single deployable into to a microservices architecture by decomposing it into domains, subdomains and boundary context using **domain-driven design (DDD).**

A microservice architecture is **decentralized and loosely coupled**, allowing each service to scale and evolve independently based on user demands.



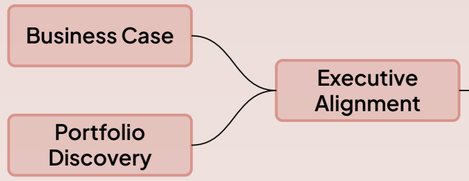
- Large teams for each technology function
- Software architecture may reflect that of an organization that structures teams based on their technology roles (Conway's Law) e.g. frontend engineer, database admin, etc.
- Higher coupling and dependencies and cognitive load



- Smaller, independent teams that are aligned to their subdomains within the larger business domain.
- Reduced coupling and cognitive load
- Improved flow and team autonomy

Modernization Approach

ACTIVATE AND ASSESS

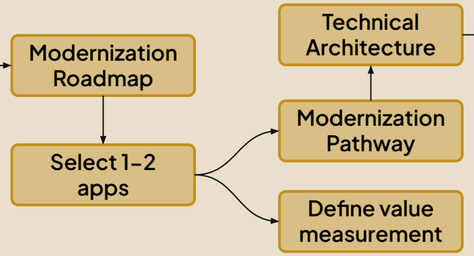


- Build a compelling business case for modernization which aligns with your strategic vision.
- Assessment of current application portfolio.
- Obtain executive alignment and buy-in
- Assemble modernization enablement team before kick-off

PLAN AND MOBILIZE

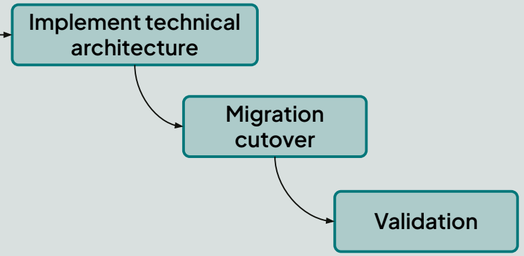
ENABLEMENT TEAM

Communicate Vision | Expert knowledge | Facilitate Design | Sustain momentum | Promote Learnings



- Establish modernization priorities and roadmap of major milestones.
- Select 1-2 strategic applications for pilot.
- Establish modernization pathway for each application, develop technical architecture.
- Define metrics to measure value that will be delivered by modernization activity e.g. DORA, revenue and growth, app performance, etc.

BUILD AND VALIDATE



- Execute modernization plan for selected pilot application
- Socialize learnings so other teams can learn.
- Validate delivered value against defined metrics.

ITERATE AND EVOLVE

Continuous Modernization

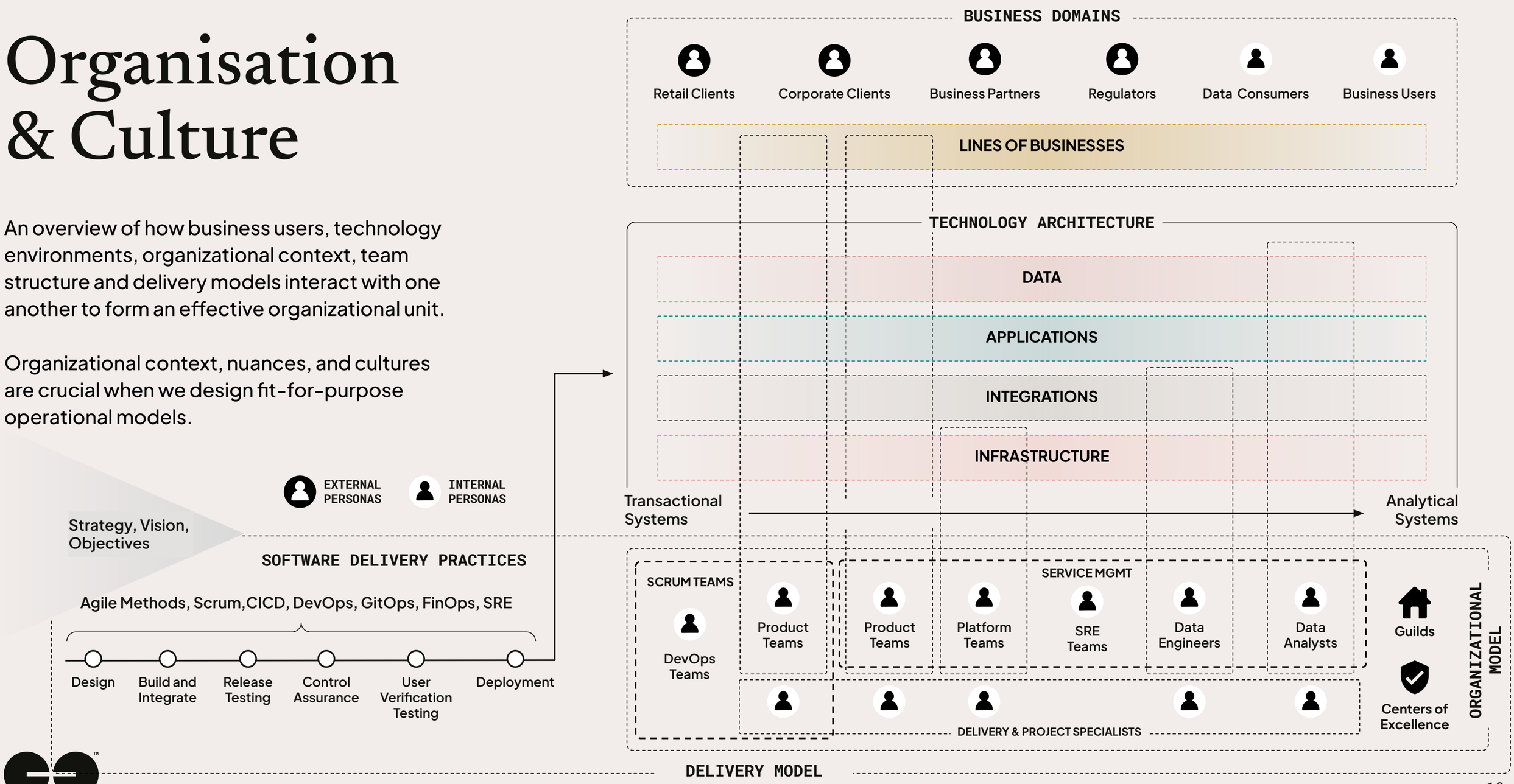
- Develop modernization playbooks and establish standard patterns.
- Review application portfolio and target remaining applications.
- Continuously evolve current and future portfolio of applications.

Organization and Culture

Organisation & Culture

An overview of how business users, technology environments, organizational context, team structure and delivery models interact with one another to form an effective organizational unit.

Organizational context, nuances, and cultures are crucial when we design fit-for-purpose operational models.



Operating Principles: Agile, DevSecOps and SRE

Function: Agile

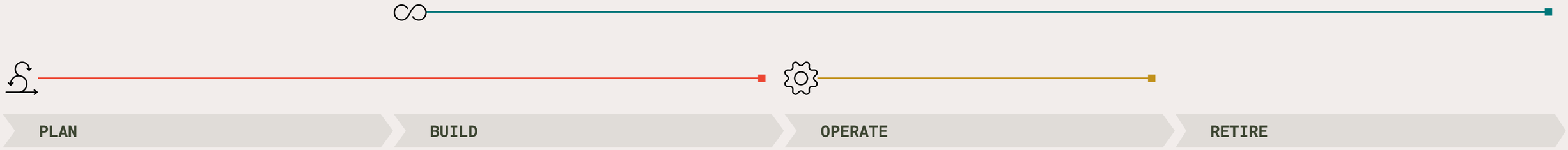
- Agile is an iterative approach to project management and software development emphasising collaboration, customer feedback, and rapid releases.
- Agile has four core values: prioritizing individuals and interactions, working software, customer collaboration, and responding to change

Form: DevSecOps

- DevOps is a software development and operations approach that integrates agile principles and security best practices to enable faster, more secure & more reliable software delivery.
- It involves enhanced automation & improved collaboration between development, operations and security teams.

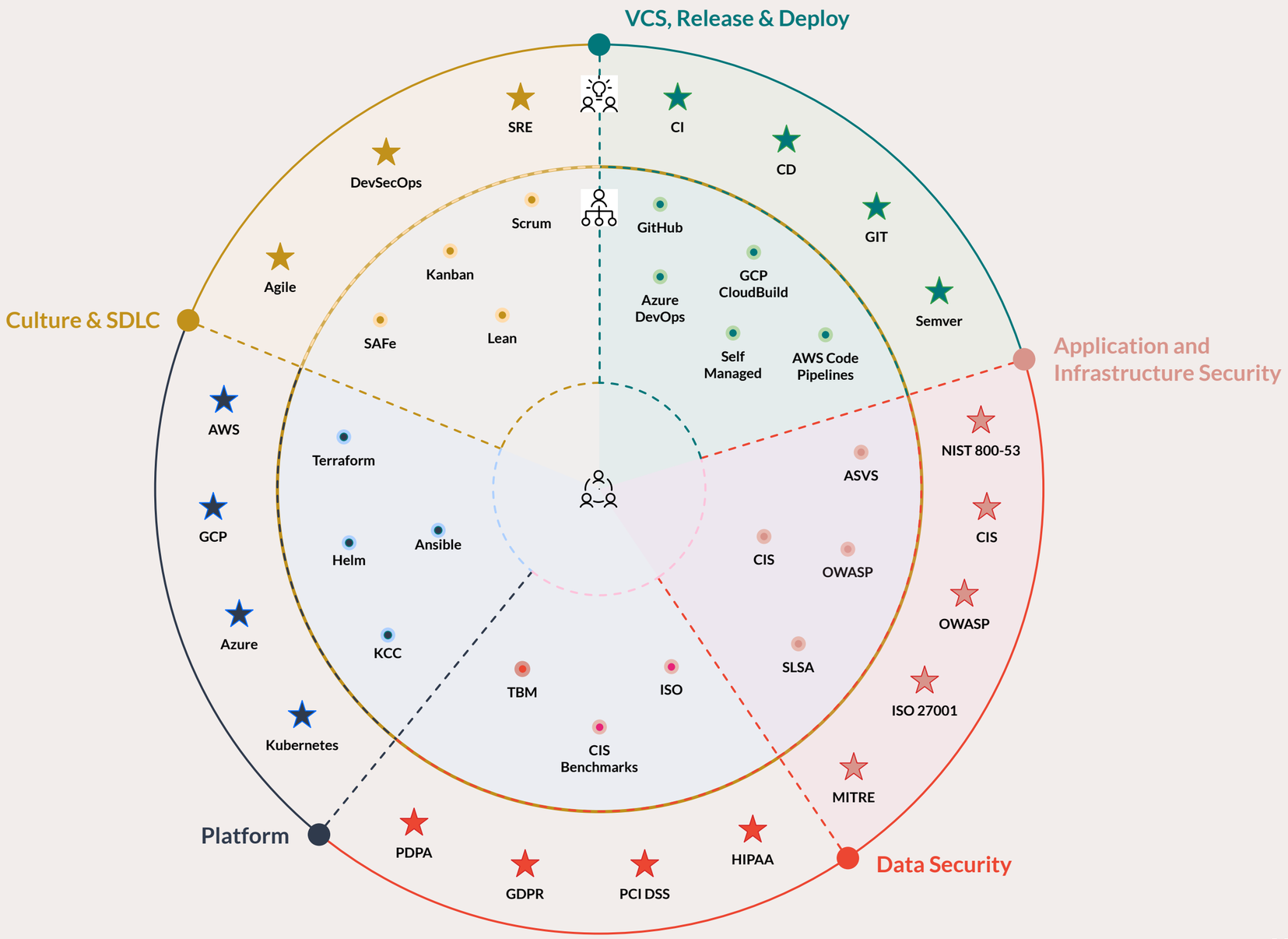
Sustain: SRE

- Site Reliability Engineering is an approach which treats operations as a software problem.
- It aims to monitor technology services through KPIs such as availability, latency, performance, and capacity, and automates actions from observability to produce self-healing systems.

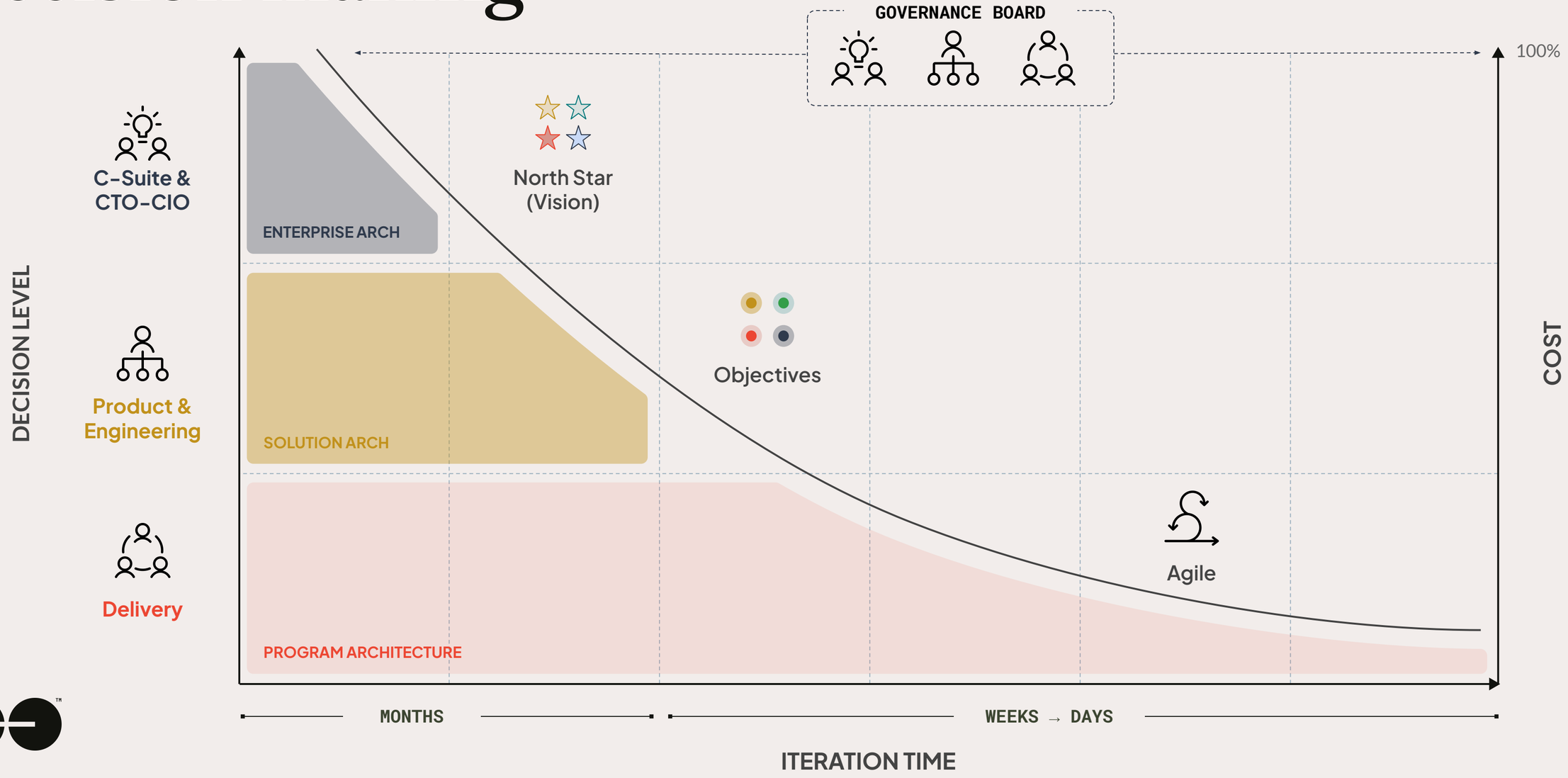


Clarity of the ecosystem

- Innovation is driven through diversity and this is especially true in the world of technology.
- Complexity is introduced through variation, which is inevitable when we want to enable developer and engineering choices at implementation time.
- To achieve value we must define **foundational principles** and minimum amount of **workflow** to achieve organisational outcome.
- Managing workflows and ensuring best practice across this complexity is achievable only by automating workflow at an orchestration level that drives both process and deployment.



Define the North Star to enable distributed decision making



A Comparison

A worked example: Context

Business: Growing e-Commerce platform
Architecture pattern: Three-tier web application
Software stack: React, Springboot, PostgreSQL, Kafka

NOTABLE REQUIREMENTS:

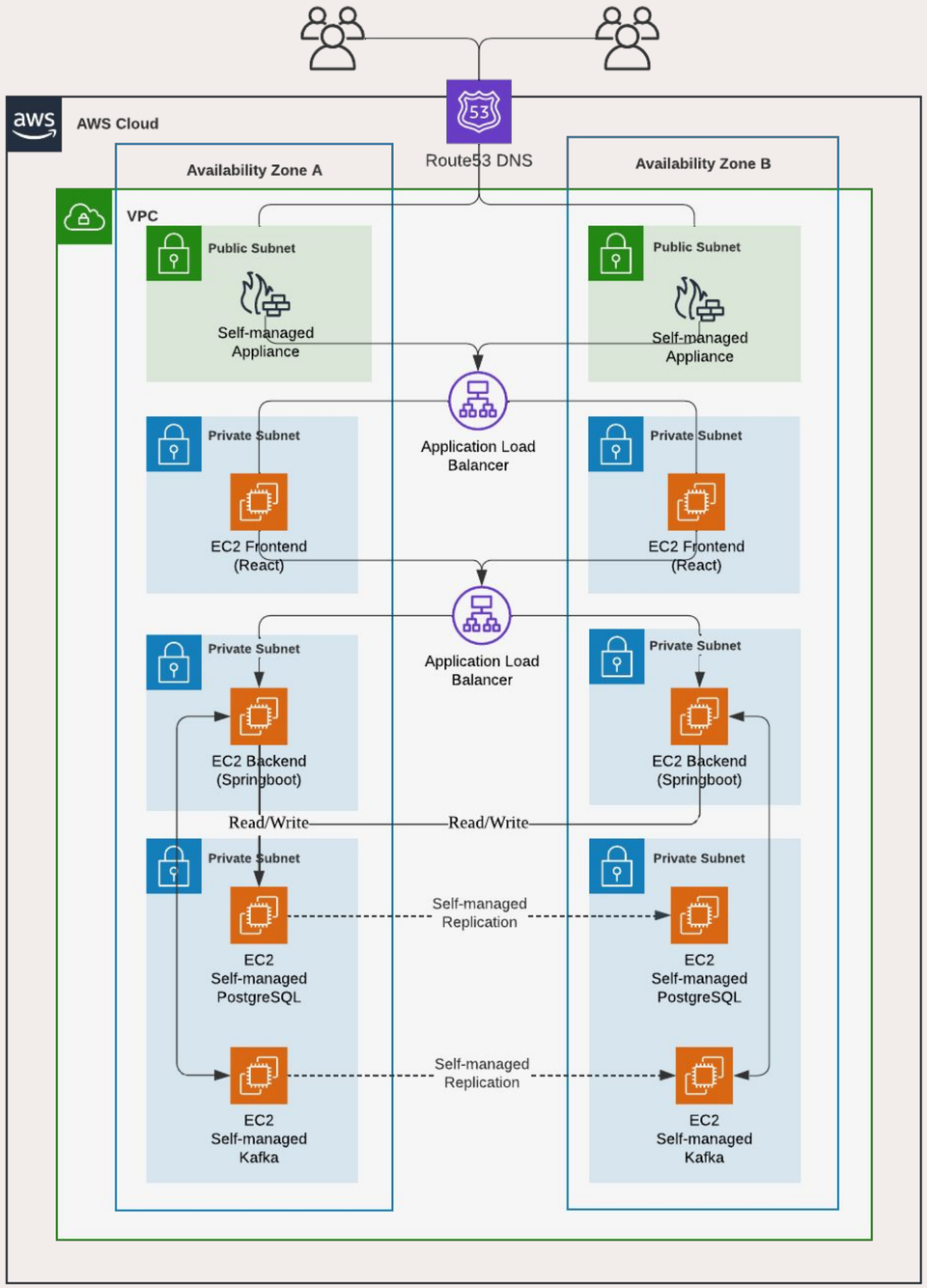
- Transaction orchestration
- Highly scalable
- Distributed development teams

A worked example: Context

Option 1

CHARACTERISTIC

- Self-managed technology stack.
- Software self-installed on standard EC2 instances.
- Patching and upgrades will need to be self-managed.
- Resiliency has to be built into design for the entire stack.
- Replication of data is self-configured.
- Baseline resources required to run perpetually.
- Scaling is complex and limited depending on application architecture.

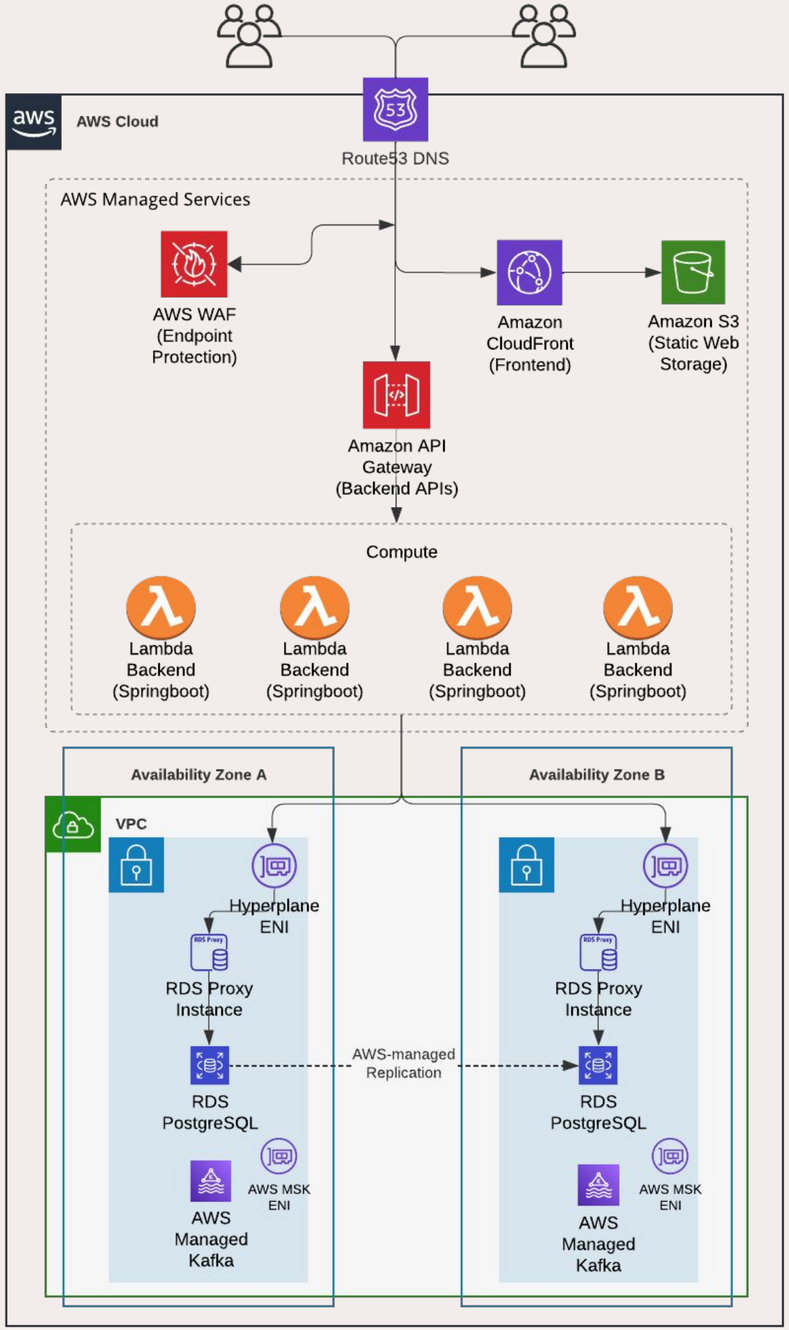


A worked example: Context

Option 2

CHARACTERISTIC

- Full to semi-managed native AWS services.
- Software is provisioned out-of-the-box and ready for use.
- Patching and upgrades on underlying infrastructure is automated and transparent.
- Resiliency is inherent for fully managed services.
- Replication of data on data layer is fully managed by AWS.
- Pay only for what you use.
- Scaling is automated, and from a users' perspective, is transparent and trivial.



A worked example: Cost

Workload patterns

- Regional customer base (Asia)
- Daily peak requests-per-second at 1000 RPS, typically between 8pm to 11pm.
- Average RPS is around 200–300 RPS.
- Monthly peak during seasonal and promotional periods is 3× of normal daily load patterns.
- Costs are based on a monthly basis.

Option 1 (Self-managed)

$7124/ month

Option 2 (AWS Native)

$6083/month

Thank you!