Applying Well Architected Framework to Machine Learning Ops

Vikas S. Rajput

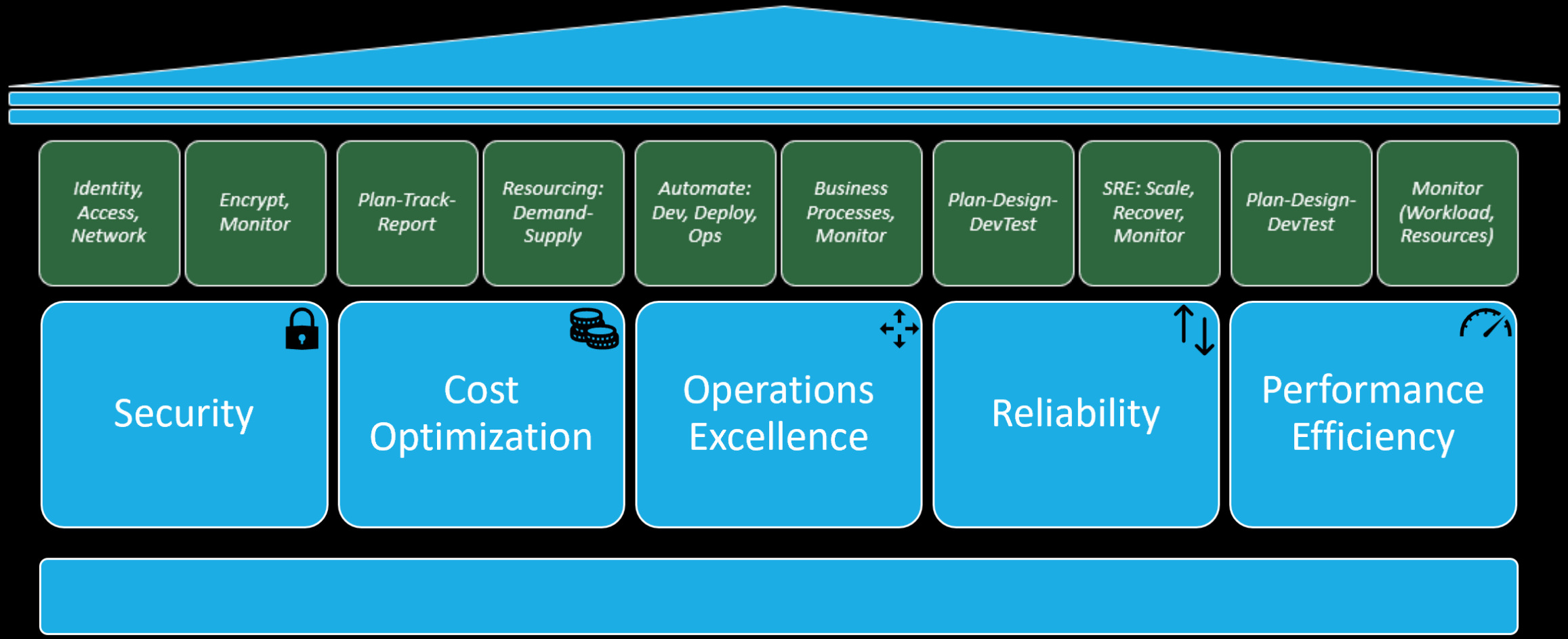
Agenda

1. What’s WAF?
2. What’s MLOps?
3. WAF + MLOps

Well Architected Framework

WAF Pillars

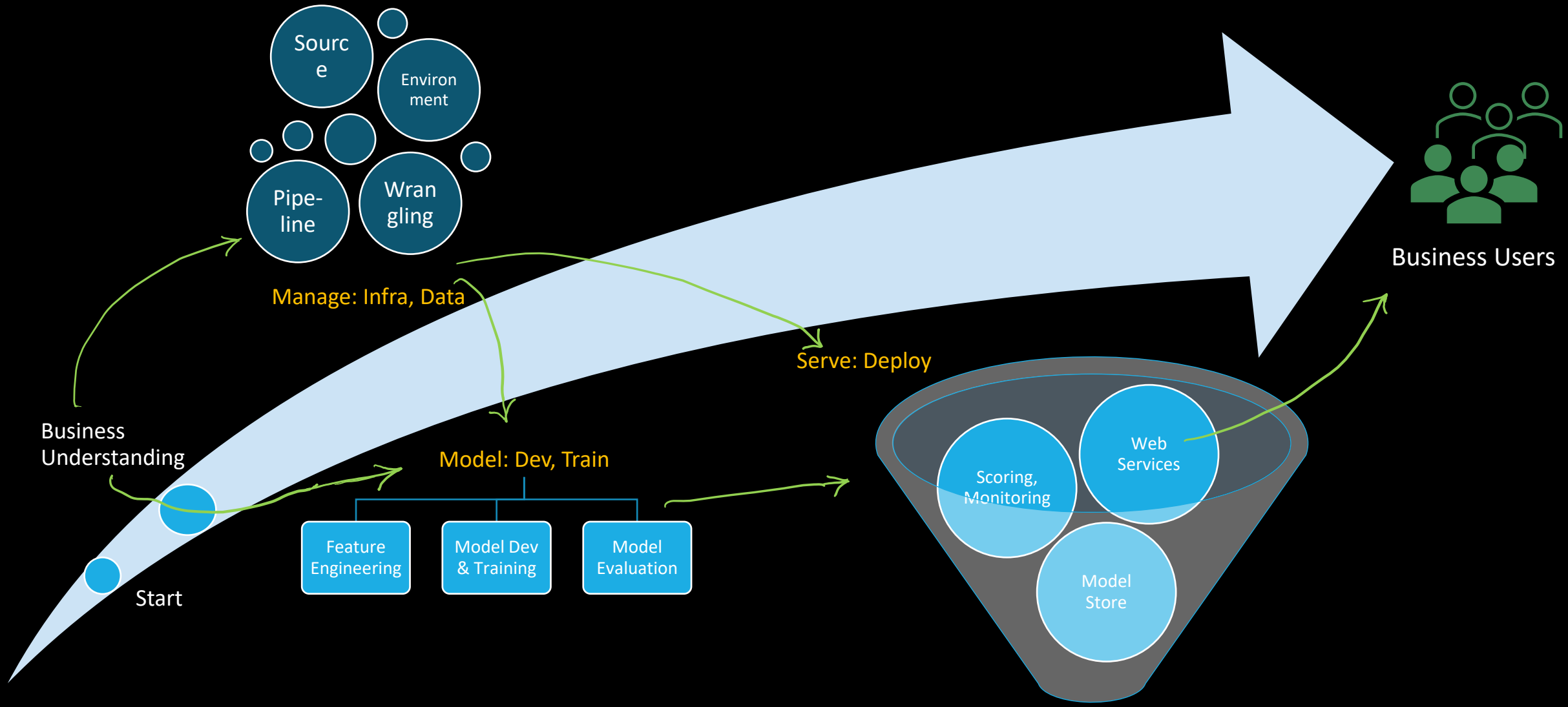
WAF Pillars

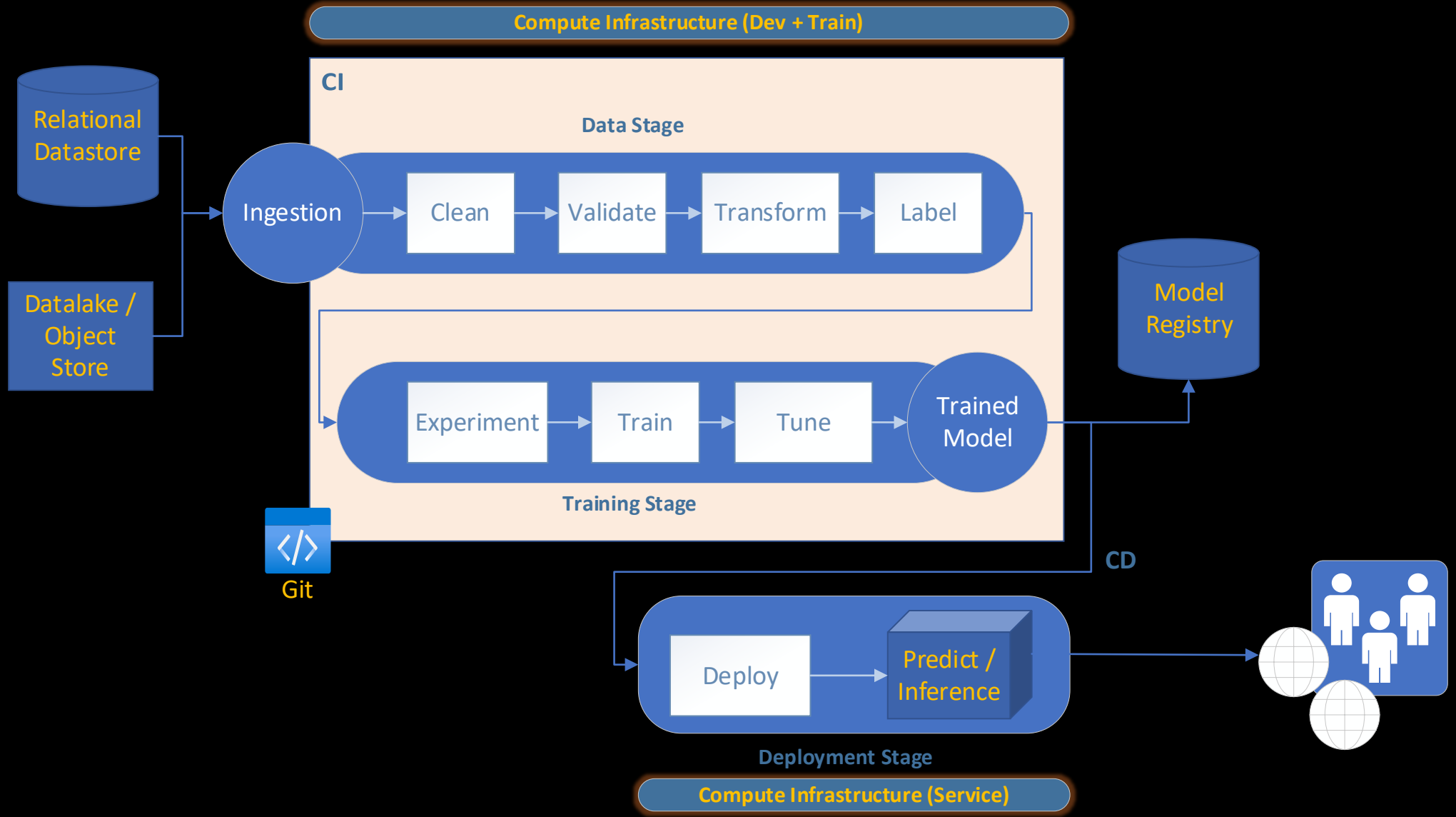

Identity, Access, Network
Encrypt, Monitor
Plan-Track-Report
Resourcing: Demand-Supply
Automate: Dev, Deploy, Ops
Business Processes, Monitor
Plan-Design-DevTest
SRE: Scale, Recover, Monitor
Plan-Design-DevTest
Monitor (Workload, Resources)
Security
Cost Optimization
Operations Excellence
Reliability
Performance Efficiency

Machine Learning

MLOps

ML: Delivery Life Cycle



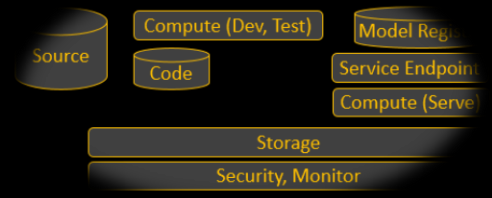


Ref: AI Infrastructure Alliance
Https://ai-infrastructure.org

MLOps + WAF

Through WAF Lens

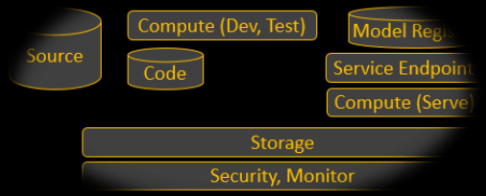
MLOps through WAF Lens



Security

Identity	Access	Network	Protect	Monitor
Centralize IAM	Limit privileged Access	Protect Source, Integration and Consumption tiers	Data Governance	Scan Credentials (code, data, config)
Manage Keys & Identities	Review & Revoke	Plan for cross-network access	Encrypt – rest, transit	Monitor for Unauthorized Access
Leverage SSO, Multi-factor	Least Privilege	Leverage Security Rules, Appliances or Services	Plan for Failure Scenario	Centralize Log Mgmt. and Analysis

MLOps through WAF Lens



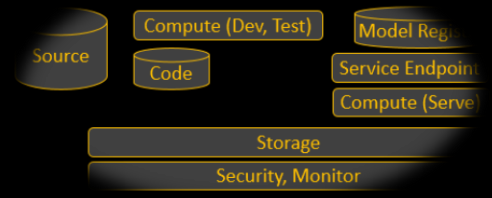
Cost Optimization

Plan-Track-Report	Resourcing
Plan / set limits on experiments, resources	Use managed services
Track outgoings	Experiment with small data volume
Consider ALL contributors, not just ML Systems	Right size training & model hosts

Operations Excellence

Automation	Monitor
Code everything. Modularize iteratively.	Resources. And Workload. Think Throughput.
Decide versioning, iteration strategy early.	Monitoring as Code. $$$ - Embed vs Platform.
Tear down every time*. Persistence is a Luxury.	Patterns are great.. ..for eyes and Ops

MLOps through WAF Lens



Reliability

Plan-Dev-Test	SRE
Know your numbers: SLA, SLO, MTTR, MTBF etc.	Target Immutable Infrastructure
Design: Prevent Failures. Code: Expect Failures.	Release Management: Controls, Deployment, Rollback
Change Management. One at a time. Monitor.	Automate everything. Including testing.

Performance Efficiency

Plan-Dev-Test	Monitor
Efficient Infrastructure - Compute, Memory	Throughput, not just resources
Efficient Code / Libraries	Benchmark early – Start, End, Check
Distribute or Parallelize or Offload jobs	Know workloads. Scale as profiles.

Questions

Thank You!