

# **OSPI DATA MODERNIZATION PROJECT PART 3: ROADMAP FOR CLOUD TRANSITION 2026**

**Deb Came, PhD**

Assistant Superintendent of Data Analysis and Support

**Ted Loran**

Chief Information Officer

**Prepared by:**

**Kris Hicks-Green**, Project Manager



Washington Office of Superintendent of  
**PUBLIC INSTRUCTION**

# Table of Contents

- Part 1: Cloud and Data Training Plan..... 9**
- 1.0 Executive Summary ..... 9
- 2.0 Learning Objectives..... 9
- 3.0 Training Curriculum..... 10
  - 3.1 POC: Governance Foundations..... 10
  - 3.2 Migration Readiness: Compliance & Risk..... 11
  - 3.3 Migration Readiness: SDLC & Change Control ..... 12
- 4.0 Long-Term Training..... 14
- 5.0 Additional Sections..... 16
  - 5.1 Success Metrics..... 16
  - 5.2 Prerequisites and Skill-Building Progression..... 16
  - 5.3 Assessment Strategies..... 16
  - 5.4 Resource Requirements..... 16
  - 5.5 Implementation Recommendations ..... 16
- Part 2: Cloud Recommendation – Foundational ..... 18**
- 1.0 Executive Summary ..... 18
- 2.0 Project Overview..... 20
  - 2.1 Current State..... 20
  - 2.2 Vision ..... 20
  - 2.3 Objectives..... 20
- 3.0 Technical Approach..... 21
  - 3.1 Enterprise Foundation ..... 21
  - 3.2 Identity & Access Management (IAM)..... 21
  - 3.3 Networking..... 22
  - 3.4 Automation..... 23
  - 3.5 Observability ..... 23
  - 3.6 Security & Compliance..... 24
  - 3.7 FinOps ..... 24
  - 3.8 Application Modernization..... 25
- 4.0 Implementation Steps..... 26

|      |   |    |
|------|---|----|
| 4.1  | Wave 1 – Enterprise Landing Zone & Initial Guardrails.....        | 26 |
| 4.2  | Wave 2 – Identity, Access, and Secure Networking .....            | 28 |
| 4.3  | Wave 3 – Automation & Observability .....                         | 30 |
| 4.4  | Wave 4 – Security Hardening & FinOps Enablement.....              | 32 |
| 4.5  | Wave 5 – Application Modernization & Continuous Improvement ..... | 35 |
| 4.6  | 6Rs Decision Tree.....  | 38 |
| 4.7  | Overall Implementation Sequence.....                              | 39 |
| 5.0  | Workforce Skills & Enablement .....                               | 40 |
| 5.0  | Core Skills by Domain .....                                       | 40 |
| 5.1  | Cross-Cutting Capabilities .....                                  | 41 |
| 5.2  | Training & Upskilling Strategy .....                              | 41 |
| 5.3  | Change Management & Culture.....                                  | 41 |
| 6.0  | Operational Model & Roles .....                                   | 42 |
| 6.1  | Cloud Platform Team (Central) .....                               | 42 |
| 6.2  | Application Teams (Federated).....                                | 42 |
| 6.3  | Governance & Oversight.....                                       | 42 |
| 7.0  | Security Operations (SecOps).....                                 | 43 |
| 7.1  | Core Components.....  | 43 |
| 8.0  | Disaster Recovery & Business Continuity .....                     | 44 |
| 8.1  | Principles .....  | 44 |
| 9.0  | Environment Lifecycle Management .....                            | 45 |
| 9.1  | Provisioning.....   | 45 |
| 9.2  | Operation .....   | 45 |
| 9.3  | Decommissioning .....   | 45 |
| 10.0 | Runbooks, Playbooks & Examples.....                               | 46 |
| 10.1 | Runbooks (Ops Teams).....   | 46 |
| 10.2 | Playbooks (Automation).....                                       | 46 |
| 11.0 | Secure Software Development Lifecycle (SDLC).....                 | 47 |
| 11.1 | Core Principles.....  | 47 |
| 11.2 | SDLC Stages in the Cloud Context .....                            | 47 |
| 11.3 | SDLC Governance.....  | 49 |

|                |  |           |
|----------------|--|-----------|
| 12.0           | End-State ("Done") .....   | 50        |
| 13.0           | Appendix .....   | 51        |
| 13.1           | Reference Links .....  | 51        |
| 13.2           | The 6Rs of Application Modernization .....                               | 51        |
| 13.3           | FinOps Framework – Overview .....  | 53        |
| 13.4           | FinOps Maturity Model.....   | 53        |
| 13.5           | Example Tagging Policy Matrix.....                                       | 55        |
| 13.6           | Organizational Hierarchy & Accounts/Subscriptions Comparison Table ..... | 56        |
| <b>Part 3:</b> | <b>Cumulative Lessons Learned.....</b>                                   | <b>58</b> |
| 1.0            | Cumulative Lessons Learned .....   | 58        |
| 1.1            | Project Information .....  | 58        |
| 1.2            | Part A: Sprint-by-Sprint Retrospectives.....                             | 58        |
|                | Sprint 0 - Environment Setup & Data Inventory.....                       | 58        |
| 1.3            | Part B: Cumulative Lessons by Workstream.....                            | 59        |
|                | 1. Cloud/Infrastructure - Cumulative Lessons.....                        | 59        |
|                | 2. Data Development - Cumulative Lessons.....                            | 60        |
| 1.4            | Part C: Cross-Workstream Patterns.....                                   | 61        |
|                | Persistent Issues (Appeared in 3+ Sprints).....                          | 61        |
|                | Successfully Resolved Issues.....  | 62        |
|                | Unresolved / Carried Forward Issues.....                                 | 62        |
| 1.5            | Part D: Cumulative Action Items and Improvements .....                   | 62        |
|                | Actions Completed During Project.....                                    | 62        |
|                | Actions Still Pending .....  | 62        |
|                | Recommended Actions for Future Projects .....                            | 63        |
| 1.6            | Part E: Overall Project Metrics & Trends .....                           | 63        |
|                | Sprint Velocity Trends (captured in DailyStandup App).....               | 63        |
|                | Cumulative Issue Resolution.....   | 64        |
|                | Action Item Effectiveness .....  | 64        |
| 1.7            | Part F: Key Takeaways and Recommendations.....                           | 64        |
|                | Top 5 Success Patterns to Replicate .....                                | 64        |

|   |           |
|---|-----------|
| Top 5 Failure Patterns to Avoid.....                                  | 65        |
| 1.8    Part G: Organizational Learning.....                           | 65        |
| Process Improvements Implemented.....                                 | 65        |
| Tool/Technology Adoptions.....  | 65        |
| Team Capability Development.....                                      | 65        |
| Knowledge Assets Created.....   | 66        |
| 1.9    Part H: Executive Summary.....                                 | 66        |
| Project Learning Velocity.....  | 66        |
| Impact on Workstreams.....  | 66        |
| Return on Learning Investment.....                                    | 66        |
| Critical Recommendations for Leadership.....                          | 67        |
| 1.10   Appendices.....  | 67        |
| Appendix A: Sprint Retrospective Participants.....                    | 67        |
| Appendix B: Detailed Sprint Timelines.....                            | 67        |
| Appendix C: Supporting Documentation.....                             | 67        |
| <b>Part 4: Modernization Complexity Matrix.....</b>                   | <b>68</b> |
| 1.0    Systems Complexity Matrix for Data Modernization Decision..... | 68        |
| 1.1    Instructions.....  | 68        |
| 1.2    Assessment.....  | 70        |
| 1.3    eCert.....   | 72        |
| 1.4    Educator Reporting.....  | 74        |
| 1.5    Report Card.....   | 76        |
| 1.6    Summary Data.....  | 78        |
| 1.7    Complexity Heatmap.....  | 78        |
| 1.8    Complexity Summaries.....                                      | 79        |
| <b>Part 5: OSPI Data Implementation Roadmap.....</b>                  | <b>80</b> |
| 1.0    Executive Summary.....   | 80        |
| 2.0    Current State: The Imperative for Change.....                  | 81        |
| 2.1    Challenges.....  | 81        |
| Current State Architectural Problems.....                             | 82        |

|   |     |
|---|-----|
| 1. Point-to-Point Integration Pattern.....                | 82  |
| 2.2 Future State: A Modern Data Ecosystem .....           | 83  |
| Vision .....  | 86  |
| Transformation Approach .....                             | 87  |
| 2.3 Data Governance: From Documented to Enforcement ..... | 88  |
| Current Governance Challenges.....                        | 88  |
| Future Governance Framework.....                          | 88  |
| Zero Gray Areas Policy .....                              | 88  |
| Governance Benefits .....                                 | 89  |
| 2.4 Prerequisites for Success.....                        | 90  |
| Infrastructure and Contracts .....                        | 90  |
| Organizational Readiness.....                             | 90  |
| Technical Foundation.....                                 | 90  |
| 2.5 Product Operating Model.....                          | 91  |
| Why Product Teams? .....                                  | 91  |
| Team Structure .....                                      | 91  |
| Product Teams Key Principles.....                         | 92  |
| 2.6 Implementation Roadmap.....                           | 93  |
| Data Product Teams – Independent Release Trains .....     | 93  |
| Release 0: Foundation .....                               | 98  |
| Why Release 0 Matters.....                                | 98  |
| Release Strategy .....                                    | 98  |
| Program Management Office.....                            | 98  |
| Change Management & Strategic Communications.....         | 100 |
| 2.7 Critical Success Factors .....                        | 102 |
| 2.8 Expected Outcomes .....                               | 102 |
| Technical Outcomes .....                                  | 102 |
| Business Outcomes .....                                   | 102 |
| Financial Outcomes.....                                   | 103 |
| Governance Outcomes.....                                  | 103 |

|   |   |            |
|---|---|------------|
| 2.9   | Risk Mitigation .....   | 103        |
| 2.10  | Roadmap as a Guiding Framework.....   | 103        |
| <b>Part 6: OSPI Proof of Concept (POC) and Migration Readiness Statement of Work.....</b> |   | <b>105</b> |
| 1.0   | Background and Efforts to Date.....   | 105        |
| 2.0   | Vision .....  | 105        |
| 3.0   | Purpose of the Implementation Planning Study.....                             | 106        |
| 4.0   | Scope of Services .....   | 106        |
| 4.1   | Implementation of Strategy Development, Governance and Team Structuring ..... | 106        |
| 4.2   | Partner Engagement Planning .....   | 107        |
| 4.3   | System Requirements Validation .....  | 107        |
| 4.4   | Data Migration and System Integration.....                                    | 107        |
| 4.5   | Project Management and Reporting .....  | 108        |
| 5.0   | Deliverables.....   | 108        |
| 6.0   | Proposed 8-Month Timeline.....  | 109        |
| 7.0   | Vendor Qualifications.....  | 109        |

# PART 1: CLOUD AND DATA TRAINING PLAN

## 1.0 Executive Summary

This comprehensive Microsoft Learning training curriculum is designed to enhance professionals' skills in key areas of Microsoft technologies, spanning various aspects of governance, compliance, data management, and operational excellence. The curriculum is structured into three distinct phases (e.g., POC1, POC2, POC3), each containing six targeted modules that align with specific job roles within organizations. By following this curriculum, learners will gain essential competencies to leverage Microsoft solutions effectively and drive organizational success.

## 2.0 Learning Objectives

1. Understand the foundational concepts of change management within the DevOps framework.
2. Gain insights into data classification and compliance requirements using Microsoft Purview.
3. Develop skills in CI/CD practices using GitHub Actions.
4. Learn data transformation techniques utilizing Delta Lake.
5. Explore Azure networking solutions and their applications.
6. Analyze cost anomalies and implement cost optimization strategies.
7. Master operational excellence maturity models.
8. Ensure business continuity and disaster recovery (BCDR) planning.
9. Stream data effectively using Azure Stream Analytics.
10. Apply responsible AI principles and ensure data governance.

# 3.0 Training Curriculum

## 3.1 POC: Governance Foundations

### 1. Change Management in DevOps

*Description:* This module covers the principles and practices of change management within DevOps, focusing on ensuring smooth transitions and minimizing disruptions.

*Duration:* 2 days

*Target Persona(s):* Senior DevOps Engineer, Program Manager

### 2. Data Classification in Microsoft Compliance (Purview)

*Description:* This module introduces data classification strategies using Microsoft Purview to enhance compliance and protect sensitive information.

*Duration:* 1 day

*Target Persona(s):* Data Architect, Cybersecurity Architect

### 3. GitHub Actions CI/CD

*Description:* Learn to automate your software development workflows with GitHub Actions, integrating CI/CD practices into your projects.

*Duration:* 2 days

*Target Persona(s):* Senior DevOps Engineer, Test Automation Engineer

### 4. Data Transformation with Delta Lake

*Description:* Explore data transformation techniques using Delta Lake to ensure efficient and reliable data processing.

*Duration:* 2 days

*Target Persona(s):* Data Platform Engineer, Data Architect

### 5. Azure Networking Solutions (AZ-700)

*Description:* This module covers networking solutions in Azure, including design, implementation, and management of Azure networking resources.

*Duration:* 3 days

*Target Persona(s):* Cloud Platform Engineer, Application Architect

## 6. Cost Anomaly Detection

*Description:* Learn to identify and respond to cost anomalies in cloud services, helping organizations manage their budgets effectively.

*Duration:* 1 day

*Target Persona(s):* Program Manager, Cloud Support Engineer

# 3.2 Migration Readiness: Compliance & Risk

## 1. Operational Excellence Maturity

*Description:* Understand the maturity model for operational excellence and how to assess and improve operations within your organization.

*Duration:* 2 days

*Target Persona(s):* Program Manager, Enterprise Architect

## 2. Microsoft Purview Information Protection

*Description:* Explore how to leverage Microsoft Purview for information protection, ensuring that sensitive data remains secure.

*Duration:* 1 day

*Target Persona(s):* Cybersecurity Architect, Data Governance Lead

## 3. Business Continuity & Disaster Recovery (BCDR)

*Description:* Learn strategies to ensure business continuity and effective disaster recovery planning to mitigate risks in operations.

*Duration:* 2 days

*Target Persona(s):* Program Manager, Cybersecurity Engineer

## 4. Stream Data with Azure Stream Analytics

*Description:* This module covers the real-time data processing capabilities of Azure Stream Analytics and its application in various scenarios.

*Duration:* 2 days

*Target Persona(s):* Data Analyst, Data Platform Engineer

## 5. Azure Well-Architected Cost Optimization

*Description:* Gain insights into cost optimization strategies within the Azure Well-Architected Framework to enhance financial efficiency.

*Duration:* 1 day

*Target Persona(s):* Program Manager, Cloud Platform Engineer

## 6. Responsible AI Principles

*Description:* Learn the principles of responsible AI to ensure ethical and transparent AI practices in your organization.

*Duration:* 1 day

*Target Persona(s):* Product Manager, Data Analyst

# 3.3 Migration Readiness: SDLC & Change Control

## 1. Azure Key Vault & Managed HSM

*Description:* Understand how to secure your applications and data with Azure Key Vault and Managed Hardware Security Modules (HSM).

*Duration:* 2 days

*Target Persona(s):* Cybersecurity Architect, Senior .NET Developer

## 2. Retrieval-Augmented Generation (RAG) Apps

*Description:* Explore the development of RAG applications that integrate retrieval mechanisms to enhance AI-driven solutions.

*Duration:* 2 days

*Target Persona(s):* Data Scientist, Application Architect

## 3. Data Governance with Microsoft Purview

*Description:* Learn to implement data governance best practices using Microsoft Purview to ensure data compliance and security.

*Duration:* 2 days

*Target Persona(s):* Data Architect, Data Governance Lead

## 4. CI/CD Enablement

*Description:* This module focuses on enabling Continuous Integration and Continuous Deployment practices to streamline software delivery.

*Duration:* 2 days

*Target Persona(s):* Senior DevOps Engineer, Test Automation Engineer

## 5. Feature Store & AI Readiness

*Description:* Explore the concept of feature stores and how they contribute to AI readiness and model deployment.

*Duration:* 1 day

*Target Persona(s):* Data Scientist, Data Platform Engineer

## **6. Advanced ML & AI Engineering**

*Description:* Dive deeper into advanced machine learning and AI engineering practices to build scalable AI solutions.

*Duration:* 2 days

# **4.0 Long-Term Training**

## **1. Microsoft Certified: Azure Solutions Architect Expert**

*Description:* This certification validates skills in designing and implementing solutions that run on Azure, including aspects like compute, network, storage, and security.

*Duration:* 3-6 months

*Target Persona(s):* Enterprise Architect, Application Architect

## **2. Microsoft Certified: Azure DevOps Engineer Expert**

*Description:* This certification is for professionals who want to demonstrate their understanding of DevOps practices and how to implement them using Azure technologies.

*Duration:* 3-6 months

*Target Persona(s):* Senior DevOps Engineer, Program Manager

## **3. Microsoft Certified: Azure Data Scientist Associate**

*Description:* This certification focuses on utilizing Azure's machine learning services and tools to build and deploy machine learning models.

*Duration:* 2-4 months

*Target Persona(s):* Data Scientist, Data Analyst

## **4. Microsoft Certified: Azure Security Engineer Associate**

*Description:* This certification validates the skills to manage security and compliance across Azure services, focusing on security controls.

*Duration:* 2-4 months

*Target Persona(s):* Cybersecurity Engineer, Cybersecurity Architect

## **5. Microsoft Certified: Azure AI Engineer Associate**

*Description:* This certification focuses on the skills needed to build, manage, and deploy AI solutions that leverage Azure AI services.

*Duration:* 2-4 months

*Target Persona(s):* Data Scientist, Application Architect

## 6. Microsoft Certified: Azure Data Engineer Associate

*Description:* This certification demonstrates the skills to design and implement data solutions on Azure, ensuring data integrity and security.

*Duration:* 2-4 months

*Target Persona(s):* Data Engineer, Data Architect

# 5.0 Additional Sections

## 5.1 Success Metrics

- **Training Effectiveness:** Measure learner satisfaction through surveys and feedback forms.
- **Knowledge Retention:** Assess knowledge retention rates through quizzes before and after training.
- **Skill Application:** Evaluate the application of skills in real projects post-training through performance reviews.
- **Certification Achievement:** Track the number of certifications earned by participants within 6 months of completing the training.

## 5.2 Prerequisites and Skill-Building Progression

- **Prerequisites:** Basic understanding of cloud concepts and familiarity with Microsoft Azure services.
- **Skill Progression:** Each module builds on the previous one, ensuring a logical progression from foundational concepts to advanced applications.

## 5.3 Assessment Strategies

- **Quizzes:** Conduct periodic quizzes to evaluate knowledge acquisition.
- **Hands-On Labs:** Implement lab sessions where learners can practice skills in a controlled environment.
- **Project Work:** Assign projects that require learners to apply concepts learned in the modules.

## 5.4 Resource Requirements

- **Technical Requirements:** Access to Azure subscriptions for hands-on labs.
- **Lab Environments:** Set up virtual labs for practical exercises and simulations.
- **Training Materials:** Ensure all learners have access to course materials and resources.

## 5.5 Implementation Recommendations

- **Pilot Program:** Start with a pilot program to gather feedback and adjust.
- **Continuous Improvement:** Regularly review and update course content to keep it aligned with the latest Microsoft technologies and practices.
- **Stakeholder Engagement:** Involve key stakeholders in the development and rollout of the training program to ensure buy-in and support.

This structured curriculum is designed to empower professionals with the skills and knowledge required to excel in their roles and effectively utilize Microsoft technologies, driving innovation and operational excellence within their organizations.

# PART 2: CLOUD RECOMMENDATION – FOUNDATIONAL

## 1.0 Executive Summary

The Office of Superintendent of Public Instruction (OSPI) faces a foundational challenge: four core data domains — Student, Educator, Fiscal, and Directory — are currently siloed in legacy applications that were not designed for interoperability or modern analytics. This results in inefficiency, inconsistent reporting, limited automation, and a lack of prompt insights that are critical to decision-making at the state and district levels.

To address this, OSPI will adopt a cloud-native enterprise data and application platform designed to unify domains, enforce governance, and enable scalable modernization. The strategy is cloud-agnostic by design, referencing the Azure Cloud Adoption Framework, the AWS Well-Architected Framework, and the Google Cloud Adoption Framework. These frameworks provide proven design patterns, but the guiding principle is vendor neutrality — ensuring that architecture patterns, governance controls, and automation can be applied consistently regardless of cloud provider.

The end-state vision is not a lift-and-shift or an isolated MVP. Instead, “done” means:

- A hardened **enterprise landing zone** with consistent account/project/subscription hierarchy, policy guardrails, and shared service foundations.
- Integrated **Identity and Access Management (IAM)** tied to OSPI’s enterprise directory, with MFA (where appropriate), RBAC, and workload identity enforcement.
- **Networking** built on a hub-and-spoke model, connecting districts, state data centers, and SaaS applications securely via private endpoints and encrypted tunnels.
- **Automation-first provisioning** using Infrastructure as Code (IaC), integrated CI/CD pipelines, and policy-based drift detection.
- **Comprehensive observability** — all logs, metrics, and traces aggregated in a secure telemetry account/project, with actionable alerting integrated into operations workflows.
- **Policy-driven security**: data encryption at rest and in transit, zero trust networking, vulnerability scanning, compliance guardrails for FERPA, NIST, and state OCIO standards.
- **FinOps maturity**: enforced tagging, cost allocation dashboards, anomaly detection, and cloud spend tied directly to dollars-per-student KPIs (as an example).

- **Application modernization:** workloads classified against the 6R model (Rehost, Replatform, Refactor, Rearchitect, Replace, Retire), migrated in waves with repeatable patterns.

## 2.0 Project Overview

### 2.1 Current State

The data infrastructure is fragmented across four domains — Student, Educator, Fiscal, Directory — each bound to legacy applications that are siloed, at different lifecycle stages, manual-process heavy, lacking data provenance and lineage tracking, and weak in master data management (MDM). The result is a system that impedes operational efficiency, limits analytics capabilities, and undermines strategic planning.

### 2.2 Vision

The target state is a cloud-based integrated data and application ecosystem that unifies the four domains into a governed, centralized foundation, provides real-time data ingestion, transformation, and access, enables advanced analytics, AI/ML forecasting, and equity reporting, supports secure, role-based access for districts, policymakers, and educators, and embeds automation at every layer, eliminating manual bottlenecks.

### 2.3 Objectives

1. **Modernize foundational infrastructure:** Establish enterprise cloud landing zones across a selected provider with policy-as-code, logging, and cost controls.
2. **Secure the environment:** Integrate IAM, encryption, and zero-trust networking.
3. **Enable automation and observability:** Provision with IaC, enforce CI/CD pipelines, and unify telemetry.
4. **Optimize costs through FinOps:** Drive financial accountability with tagging, dashboards, and forecasting.
5. **Migrate and modernize applications:** Apply structured 6R strategies to workloads, prioritizing by value, compliance risk, and complexity.

## 3.0 Technical Approach

### 3.1 Enterprise Foundation

**Purpose:** Establish a secure, compliant, and scalable foundation for all workloads. Without this, every future deployment risks inconsistency, drift, or non-compliance. The KPIs listed in this section should be viewed as examples of what “good looks like” and should be fit for purpose for OSPI.

#### Key Elements:

- **Organizational Hierarchy & Accounts/Subscriptions:** Establishing a hierarchical governance structure is the first step in building an enterprise landing zone (e.g., “In Azure, align Subscriptions to environments; in AWS, align Accounts to workloads; in GCP, align Projects to applications”). Each hyperscaler provides a slightly different model, but all enable:
  - **Centralized policy enforcement** (security, compliance, cost controls).
  - **Delegated ownership** (application teams work within isolated boundaries).
  - **Billing consolidation** (roll-up of spending at the org level).
- **Shared Services Environment:** Dedicated account/project for logging, networking, security tooling, and billing.
- **Guardrails & Policies:** Enforce allowed regions, encryption defaults, mandatory tagging.

#### Best Practices:

- Enforce naming standards.
- Require blueprints/landing zone templates.
- Use policy-as-code in Git repos.

#### Suggested Target KPIs:

- ≥95% resource policy compliance.
- 100% automated provisioning of accounts.
- <5% variance between forecasted and actual growth.

### 3.2 Identity & Access Management (IAM)

**Purpose:** Ensure that only the right people and workloads access the right resources at the right time, with zero-trust principles. The KPIs listed in this section should be viewed as examples of what “good looks like” and should be fit for purpose for OSPI.

**Key Elements:**

- **Federated Identity Integration:** Entra ID, IAM Identity Center, Workforce Identity Federation.
- **Role-Based Access Control:** Admin, Contributor, Auditor, Service Accounts.
- **Service Identities:** Managed Identities, IAM Roles, Workload Identities.
- **Privileged Access Controls:** JIT elevation, Access Analyzer.

**Best Practices:**

- MFA enforced for all privileged roles.
- Rotate or eliminate static keys.
- Quarterly least-privilege reviews.

**Suggested Tarket KPIs:**

- 100% MFA on privileged accounts.
- Zero stale service accounts older than 90 days.
- Provisioning SLA  $\leq$  1 day.

## 3.3 Networking

**Purpose:** Provide secure, performant, and resilient connectivity between cloud, districts, state data centers, and SaaS platforms. The KPIs listed in this section should be viewed as examples of what “good looks like” and should be fit for purpose for OSPI.

**Key Elements:**

- Hub-and-spoke/Shared VPC architectures.
- Hybrid connectivity via VPN, ExpressRoute, Direct Connect, Interconnect.
- Private service endpoints.
- Firewalls, WAFs, segmentation.

**Best Practices:**

- Deny-all inbound by default.
- Centralize DNS.
- Tier workloads into Web/App/DB.

**Suggested Tarket KPIs:**

- Network uptime  $\geq$  99.9%.
- Latency < 50ms on core paths.
- Zero critical misconfigurations.

## 3.4 Automation

**Purpose:** Replace manual provisioning with IaC and CI/CD pipelines, reducing error and ensuring reproducibility. The KPIs listed in this section should be viewed as examples of what “good looks like” and should be fit for purpose for OSPI.

### Key Elements:

- **IaC tools:** Terraform, Bicep, CloudFormation, Deployment Manager.
- **Pipelines:** DevOps, GitHub Actions, CodePipeline, Cloud Build.
- **Policy enforcement:** OPA/Rego.
- **Drift detection:** Resource Graph, AWS Config, GCP Config Connector.

### Best Practices:

- Standardize modules.
- Require peer-reviewed pull requests.
- Bake security scans into pipelines.

### Suggested Target KPIs:

- $\geq 95\%$  infra via IaC.
- Drift remediation  $\leq 24$  hours.
- Pipeline success  $\geq 90\%$ .

## 3.5 Observability

**Purpose:** Provide continuous visibility into system health, performance, and usage. The KPIs listed in this section should be viewed as examples of what “good looks like” and should be fit for purpose for OSPI.

### Key Elements:

- Logging: Monitor Logs, CloudWatch, Cloud Logging.
- Metrics & Tracing: OpenTelemetry, App Insights, X-Ray, Cloud Trace.
- Alerting: Grafana, dashboards, ITSM integration.

### Best Practices:

- Actionable alerts only.
- Define SLAs and SLOs.
- Enable anomaly detection.

### Suggested Target KPIs:

- MTTR reduced  $\geq 30\%$ .

- ≥90% actionable alerts.
- ≥80% dashboard adoption.

## 3.6 Security & Compliance

**Purpose:** Protect education data and align with FERPA, NIST, WaTech Section 701. The KPIs listed in this section should be viewed as examples of what “good looks like” and should be fit for purpose for OSPI.

### Key Elements:

- Encryption at rest and transit.
- Threat Protection: Defender, GuardDuty, SCC.
- Compliance Scanning: CIS benchmarks.
- Perimeter Defense: DDoS, WAF.

### Best Practices:

- Policy-as-code compliance.
- Automated patching.
- Quarterly penetration testing.

### Suggested Tarket KPIs:

- 100% encryption compliance.
- Critical vuln remediation  $\leq$  48h.
- Compliance score  $\geq$  90%.

## 3.7 FinOps

**Purpose:** Ties spend to outcomes with transparency, predictability, and optimization. The KPIs listed in this section should be viewed as examples of what “good looks like” and should be fit for purpose for OSPI.

### Key Elements:

- Tagging schema enforced.
- Dashboards via Cost Management, Explorer, BigQuery.
- Forecasting  $\pm$ 10%.
- Optimization with RI/Savings Plans.

### Best Practices:

- Automate tag compliance.
- Showback/chargeback reporting.

- Right-size workloads quarterly.

Suggested Target KPIs:

- $\geq 90\%$  tagging compliance.
- Forecast accuracy  $\pm 10\%$ .
- Idle resources  $\leq 5\%$  of spend.

## 3.8 Application Modernization

**Purpose:** Migrate workloads using structured patterns balancing speed, cost, risk. The KPIs listed in this section should be viewed as examples of what “good looks like” and should be fit for purpose for OSPI.

**Key Elements:**

- **6R Framework:** Rehost, Replatform, Refactor, Rearchitect, Replace, Retire.
- **Discovery:** Inventory, dependency mapping, scoring.
- **Pilot:** Modernize one critical workload.
- **Wave Migration:** Business criticality + complexity.

**Best Practices:**

- Maintain rollback plans.
- Validate performance + compliance before cutover.
- Embed security/observability during migration.

**Suggested Target KPIs:**

- % workloads modernized per year.
- Zero critical regressions.
- $\geq 95\%$  adoption satisfaction.

## 4.0 Implementation Steps

The following implementation steps should be understood as high-level guideposts that describe the logical sequence of activities required to build a secure, governed, and sustainable cloud foundation. They represent the major phases from initial enterprise setup through continuous modernization.

These steps are not prescriptive playbooks, each will require further refinement, detailed design, and prioritization as the program progresses from the MVP (e.g., initial landing zone and first workloads) through to the "Done" state (e.g., full enterprise modernization, continuous improvement, and optimized FinOps maturity).

As the initiative advances, each step will be expanded into:

- Detailed sprint goals with specific deliverables.
- Gate criteria that must be met before advancing.
- Technical runbooks and automation modules to ensure repeatability.
- KPIs that allow leadership to track progress and outcomes.

This ensures the strategy remains actionable at the engineering level while still aligning to OSPI's long-term vision for cloud adoption and modernization.

### 4.1 Wave 1 – Enterprise Landing Zone & Initial Guardrails

**Sprints: 1–4 (2–3 weeks each)**

**Sprint Goals:**

- Establish cloud org hierarchy (e.g., Azure Management Groups, AWS Organizations, GCP Org/Folder/Project).
- Provision Shared Services account/project for logging, networking, and billing.
- Deploy initial governance policy engine (e.g., Azure Policy, AWS SCPs/Config, GCP Org Policy).
- Baseline hub-and-spoke network topology with restricted internet ingress.

**Capabilities Delivered:**

- Foundational landing zone deployed and validated.
- Initial IAM integration with corporate directory.
- Cost management budgets and alerts enabled.

**Gate Deliverables:**

- Gate 0: Landing Zone blueprint approved.

- Gate 1: First non-production workloads deployed and validated.

### Key KPIs to Check:

- Policy compliance  $\geq$  80% (e.g., naming, tagging, regions).
- MFA enabled for all privileged accounts.

### *Wave 1: Enterprise Landing Zone (Azure Example in OpenTofu)*

```
1. terraform {
2.   required_providers {
3.     azurerm = {
4.       source = "hashicorp/azurerm"
5.       version = "~> 4.0"
6.     }
7.   }
8. }
9.
10. provider "azurerm" {
11.   features {}
12. }
13.
14. # Foundation: Resource Group
15. resource "azurerm_resource_group" "landing_zone" {
16.   name     = "rg-landing-zone"
17.   location = "WestUS2"
18.
19.   tags = {
20.     environment = "landing-zone"
21.     owner       = "platform-team"
22.   }
23. }
24.
25. # Centralized Logging: Storage Account
26. resource "azurerm_storage_account" "logs" {
27.   name                        = "lzlogs${random_integer.suffix.result}"
28.   resource_group_name        = azurerm_resource_group.landing_zone.name
29.   location                    = azurerm_resource_group.landing_zone.location
30.   account_tier                = "Standard"
31.   account_replication_type    = "LRS"
32.
33.   tags = {
34.     purpose     = "central-logging"
35.     environment = "landing-zone"
36.   }
37. }
38.
39. # Random suffix for globally unique storage name
40. resource "random_integer" "suffix" {
41.   min = 10000
42.   max = 99999
43. }
44.
45. # Tag Policy: Require 'costcenter' tag on all resources
46. resource "azurerm_policy_definition" "require_costcenter" {
47.   name           = "require-costcenter"
48.   policy_type    = "Custom"
49.   mode           = "Indexed"
50.   display_name   = "Require costcenter tag"
```

```

51.
52. policy_rule = <<POLICY
53. {
54.   "if": {
55.     "not": {
56.       "field": "tags.costcenter",
57.       "exists": "true"
58.     }
59.   },
60.   "then": {
61.     "effect": "deny"
62.   }
63. }
64. POLICY
65. }
66.
67. resource "azurermsubscriptionpolicyassignment" "costcenter_assignment" {
68.   name = "assign-require-costcenter"
69.   policy_definition_id = azurermpolicydefinition.require_costcenter.id
70.   subscription_id = data.azurermsubscription.primary.id
71. }
72.
73. data "azurermsubscription" "primary" {}
74.

```

### What this snippet shows for Wave 1:

- Resource group to hold foundational resources (structure).
- Centralized logging bucket for observability baseline.
- Tag enforcement policy — a guardrail to ensure governance from the start.

## 4.2 Wave 2 – Identity, Access, and Secure Networking

### Sprints: 5–8

#### Sprint Goals:

- Integrate IAM with enterprise directory (e.g., Azure Entra ID, AWS IAM Identity Center, GCP Workforce Identity Federation).
- Define and roll out RBAC roles (e.g., Admin, Developer, Auditor, Automation).
- Deploy hybrid connectivity (e.g., ExpressRoute, Direct Connect, Interconnect) and private endpoints for PaaS services.
- Enable service principals/workload identities for automation.

#### Capabilities Delivered:

- Federated SSO with conditional access and MFA enforcement.
- Role-based access templates deployed across all subscriptions/accounts/projects.
- Hybrid connectivity operational with encryption in transit.

## Gate Deliverables:

- Gate 2: IAM integration validated across test workloads.
- Gate 3: Secure networking in place with private service endpoints.

## Key KPIs to Check:

- 100% privileged accounts MFA-enabled.
- Onboarding SLA for new users  $\leq$  1 day.
- Tag coverage  $\geq$  85%.

### *Wave 2: IAM + Secure Networking (Azure Example in OpenTofu)*

```
1. terraform {
2.   required_providers {
3.     azurerm = {
4.       source = "hashicorp/azurerm"
5.       version = "~> 4.0"
6.     }
7.     azuread = {
8.       source = "hashicorp/azuread"
9.       version = "~> 2.0"
10.    }
11.  }
12. }
13.
14. provider "azurerm" {
15.   features {}
16. }
17.
18. provider "azuread" {}
19.
20. # Networking Resource Group
21. resource "azurerm_resource_group" "network" {
22.   name     = "rg-network"
23.   location = "WestUS2"
24. }
25.
26. # Hub Virtual Network
27. resource "azurerm_virtual_network" "hub" {
28.   name                = "vnet-hub"
29.   location             = azurerm_resource_group.network.location
30.   resource_group_name = azurerm_resource_group.network.name
31.   address_space       = ["10.0.0.0/16"]
32. }
33.
34. # Subnet for Private Endpoints
35. resource "azurerm_subnet" "private" {
36.   name                = "snet-private"
37.   resource_group_name = azurerm_resource_group.network.name
38.   virtual_network_name = azurerm_virtual_network.hub.name
39.   address_prefixes    = ["10.0.1.0/24"]
40.   service_endpoints   = ["Microsoft.Storage"]
41. }
42.
43. # Example Private Endpoint (to Storage Account created in Wave 1)
44. resource "azurerm_private_endpoint" "logs" {
45.   name                = "pe-logs"
46.   location            = azurerm_resource_group.network.location
```

```

47. resource_group_name = azurearm_resource_group.network.name
48. subnet_id           = azurearm_subnet.private.id
49.
50. private_service_connection {
51.   name                       = "logs-connection"
52.   private_connection_resource_id = azurearm_storage_account.logs.id
53.   subresource_names          = ["blob"]
54. }
55. }
56.
57. # IAM: Assign Reader role to an Azure AD Group
58. data "azuread_group" "developers" {
59.   display_name = "Cloud-Developers"
60. }
61.
62. resource "azurearm_role_assignment" "devs_reader" {
63.   scope                       = azurearm_resource_group.network.id
64.   role_definition_name       = "Reader"
65.   principal_id                = data.azuread_group.developers.object_id
66. }
67.

```

### What this snippet shows for Wave 2:

- **Networking:** Hub VNet + subnet for private services + private endpoint to secure PaaS resources.
- **IAM/RBAC:** Custom role definition with limited actions, and assignment to an Azure AD group.
- **Federated Identity:** Role assignment tied to enterprise directory (e.g., Azure AD group here, equivalent to AWS IAM Identity Center or GCP IAM binding).

## 4.3 Wave 3 – Automation & Observability

### Sprints: 9–12

#### Sprint Goals:

- Deploy Infrastructure as Code (e.g., Terraform/Bicep/CloudFormation) modules for core services.
- Stand up CI/CD pipelines with policy and security checks.
- Centralize observability: logging, metrics, tracing aggregated in Shared Services.
- Integrate alerts with ITSM (e.g., ServiceNow, Jira, ADO).

#### Capabilities Delivered:

- Automated infra deployment pipelines with drift detection and remediation.
- Unified logging and telemetry dashboards.
- OpenTelemetry-based tracing implemented for pilot apps.

#### Gate Deliverables:

- Gate 4: First automated pipeline deployed to production.

- Gate 5: Full observability stack active for pilot workloads.

### Key KPIs to Check:

- $\geq 90\%$  infrastructure deployed via IaC.
- MTRR reduced  $\geq 20\%$ .
- Alert accuracy  $\geq 85\%$ .

### *Wave 3: Automation & Observability (Azure Example in OpenTofu)*

```

1. terraform {
2.   required_providers {
3.     azurerem = {
4.       source = "hashicorp/azurerem"
5.       version = "~> 4.0"
6.     }
7.     azuread = {
8.       source = "hashicorp/azuread"
9.       version = "~> 2.0"
10.    }
11.  }
12. }
13.
14. provider "azurerem" {
15.   features {}
16. }
17.
18. provider "azuread" {}
19.
20. # Resource Group for Observability
21. resource "azurerem_resource_group" "observability" {
22.   name     = "rg-observability"
23.   location = "WestUS2"
24. }
25.
26. # Central Log Analytics Workspace
27. resource "azurerem_log_analytics_workspace" "logs" {
28.   name                = "law-central"
29.   location            = azurerem_resource_group.observability.location
30.   resource_group_name = azurerem_resource_group.observability.name
31.   sku                 = "PerGB2018"
32.   retention_in_days  = 30
33. }
34.
35. # Application Insights for Metrics & Tracing
36. resource "azurerem_application_insights" "app" {
37.   name                = "appi-central"
38.   location            = azurerem_resource_group.observability.location
39.   resource_group_name = azurerem_resource_group.observability.name
40.   application_type    = "web"
41. }
42.
43. # Example: Service Principal for CI/CD pipelines
44. resource "azuread_application" "cicd_app" {
45.   display_name = "cicd-pipeline"
46. }
47.
48. resource "azuread_service_principal" "cicd_sp" {
49.   application_id = azuread_application.cicd_app.application_id
50. }

```

```
51.
52. # Role assignment for pipeline service principal
53. resource "azurerms_role_assignment" "cicd_contrib" {
54.   scope = azurerms_resource_group.observability.id
55.   role_definition_name = "Contributor"
56.   principal_id = azuread_service_principal.cicd_sp.object_id
57. }
58.
```

### What this snippet shows for Wave 3:

- **Automation:** Example use of Terraform modules to standardize infra provisioning.
- **IAM for pipelines:** Service Principal created and given least-privilege Contributor rights for deployments.
- **Observability:** Centralized Log Analytics workspace + Application Insights for metrics/tracing.

## 4.4 Wave 4 – Security Hardening & FinOps Enablement

### Sprints: 13–16

#### Sprint Goals:

- Enforce encryption at rest and in transit with managed keys.
- Deploy CSPM tooling (e.g., Azure Defender, AWS Security Hub, GCP SCC).
- Implement WAF and DDoS protection.
- Enforce tagging schema across all workloads.
- Build FinOps dashboards for spend by project/grant/funding source.

#### Capabilities Delivered:

- Policy-as-code enforced for security and compliance.
- Automated anomaly detection for spend.
- End-to-end security telemetry integrated.

#### Gate Deliverables:

- Gate 6: Security guardrails tested with penetration scenarios.
- Gate 7: FinOps dashboards validated by Finance + IT.

#### Key KPIs to Check:

- 100% encryption compliance.
- Critical vulnerabilities remediated  $\leq$  48 hours.
- Forecast accuracy  $\pm$  10%.

## Wave 4: Security Hardening & FinOps Enablement (Azure Example in OpenTofu)

```
1. terraform {
2.   required_providers {
3.     azurerm = {
4.       source = "hashicorp/azurerm"
5.       version = "~> 4.0"
6.     }
7.   }
8. }
9.
10. provider "azurerm" {
11.   features {}
12. }
13.
14. # Resource Group for Security + FinOps
15. resource "azurerm_resource_group" "security" {
16.   name     = "rg-security"
17.   location = "WestUS2"
18. }
19.
20. # Key Vault for encryption keys
21. resource "azurerm_key_vault" "kv" {
22.   name                = "oplz-keyvault"
23.   location             = azurerm_resource_group.security.location
24.   resource_group_name = azurerm_resource_group.security.name
25.   tenant_id           = data.azurerm_client_config.current.tenant_id
26.   sku_name             = "standard"
27. }
28.
29. resource "azurerm_key_vault_key" "cmk" {
30.   name          = "cmk-storage"
31.   key_vault_id = azurerm_key_vault.kv.id
32.   key_type      = "RSA"
33.   key_size      = 2048
34. }
35.
36. # Storage Account with CMK encryption
37. resource "azurerm_storage_account" "secure" {
38.   name                = "lzsecure${random_integer.suffix.result}"
39.   resource_group_name = azurerm_resource_group.security.name
40.   location             = azurerm_resource_group.security.location
41.   account_tier         = "Standard"
42.   account_replication_type = "LRS"
43.   enable_https_traffic_only = true
44.
45.   identity {
46.     type = "SystemAssigned"
47.   }
48.
49.   customer_managed_key {
50.     key_vault_key_id = azurerm_key_vault_key.cmk.id
51.   }
52.
53.   tags = {
54.     environment = "prod"
55.     costcenter  = "education"
56.   }
57. }
58.
59. resource "random_integer" "suffix" {
60.   min = 10000
61.   max = 99999
62. }
```

```

63.
64. # Policy: Require costcenter tag
65. resource "azurerms_policy_definition" "require_costcenter" {
66.   name           = "require-costcenter"
67.   policy_type    = "Custom"
68.   mode           = "Indexed"
69.   display_name   = "Require costcenter tag"
70.
71.   policy_rule = <<POLICY
72.   {
73.     "if": {
74.       "not": {
75.         "field": "tags.costcenter",
76.         "exists": "true"
77.       }
78.     },
79.     "then": {
80.       "effect": "deny"
81.     }
82.   }
83. POLICY
84. }
85.
86. resource "azurerms_subscription_policy_assignment" "costcenter_assign" {
87.   name           = "assign-require-costcenter"
88.   policy_definition_id = azurerms_policy_definition.require_costcenter.id
89.   subscription_id   = data.azurerms_subscription.primary.id
90. }
91.
92. # FinOps: Export daily cost data to secure storage
93. resource "azurerms_cost_management_export" "daily_costs" {
94.   name           = "daily-cost-export"
95.   scope          = data.azurerms_subscription.primary.id
96.   recurrence     = "Daily"
97.   recurrence_period_start = "2025-01-01T00:00:00Z"
98.   export_data {
99.     type           = "Usage"
100.    storage_account_id = data.azurerms_storage_account.secure.id
101.    container_name    = "costreports"
102.    root_folder_path = "daily"
103.  }
104.   format          = "Csv"
105. }
106.
107. data "azurerms_subscription" "primary" {}
108. data "azurerms_client_config" "current" {}
109.

```

#### What this demonstrates for Wave 4:

- **Encryption hardening:** Storage account encrypted with customer-managed keys from Key Vault.
- **Policy enforcement:** Mandatory tags (e.g., costcenter) required on all resources.
- **Security posture:** Web Application Firewall (WAF) using OWASP ruleset.
- **FinOps:** Automated cost export to a storage account for dashboards/BI reporting.

## 4.5 Wave 5 – Application Modernization & Continuous Improvement

**Sprints: 17–20 (repeatable waves beyond)**

**Sprint Goals:**

- Inventory workloads and classify by 6R strategy (e.g., Rehost, Replatform, Refactor, Rearchitect, Replace, Retire).
- Pilot modernization of one critical workload.
- Scale migration in waves with feedback from FinOps and observability.
- Tune KPIs and anomaly detection to support continuous governance.

**Capabilities Delivered:**

- Repeatable modernization playbooks for workload migration.
- Continuous feedback loops between FinOps, Observability, and Apps.
- Optimization of cloud spend and performance tied to student outcomes.

**Gate Deliverables:**

- Gate 8: First wave of production workloads migrated successfully.
- Gate 9: Continuous refinement loop operational (FinOps + Observability feeding modernization decisions).

**Key KPIs to Check:**

- % workloads modernized per wave.
- Zero critical regressions post-migration.
- All KPI targets met or exceeded for  $\geq 2$  months.

*Wave 5: Application Modernization & Continuous Improvement (Azure Example in OpenTofu)*

```
1. terraform {
2.   required_providers {
3.     azurerm = {
4.       source = "hashicorp/azurerm"
5.       version = "~> 4.0"
6.     }
7.   }
8. }
9.
10. provider "azurerm" {
11.   features {}
12. }
13.
14. # Resource Group for modernized apps
15. resource "azurerm_resource_group" "apps" {
16.   name = "rg-modern-apps"
```

```

17.   location = "WestUS2"
18. }
19.
20. # Observability - Application Insights
21. resource "azurerms_application_insights" "appinsights" {
22.   name           = "appi-modern"
23.   location       = azurerms_resource_group.apps.location
24.   resource_group_name = azurerms_resource_group.apps.name
25.   application_type = "web"
26. }
27.
28. # Example Replatform: App Service (PaaS)
29. resource "azurerms_app_service_plan" "plan" {
30.   name           = "asp-modern"
31.   location       = azurerms_resource_group.apps.location
32.   resource_group_name = azurerms_resource_group.apps.name
33.   kind           = "Linux"
34.   reserved       = true
35.
36.   sku {
37.     tier = "Standard"
38.     size = "S1"
39.   }
40. }
41.
42. resource "azurerms_app_service" "web" {
43.   name           = "modern-web-app"
44.   location       = azurerms_resource_group.apps.location
45.   resource_group_name = azurerms_resource_group.apps.name
46.   app_service_plan_id = azurerms_app_service_plan.plan.id
47.
48.   site_config {
49.     linux_fx_version = "DOTNETCORE|6.0"
50.   }
51.
52.   app_settings = {
53.     "APPINSIGHTS_INSTRUMENTATIONKEY" =
azurerms_application_insights.appinsights.instrumentation_key
54.   }
55.
56.   tags = {
57.     environment = "prod"
58.     costcenter  = "education"
59.   }
60. }
61.
62. # Example Refactor: AKS (container modernization)
63. resource "azurerms_kubernetes_cluster" "aks" {
64.   name           = "aks-modern"
65.   location       = azurerms_resource_group.apps.location
66.   resource_group_name = azurerms_resource_group.apps.name
67.   dns_prefix     = "modernapp"
68.
69.   default_node_pool {
70.     name       = "default"
71.     node_count = 2
72.     vm_size    = "Standard_DS2_v2"
73.   }
74.
75.   identity {
76.     type = "SystemAssigned"
77.   }
78.
79.   tags = {
80.     environment = "prod"

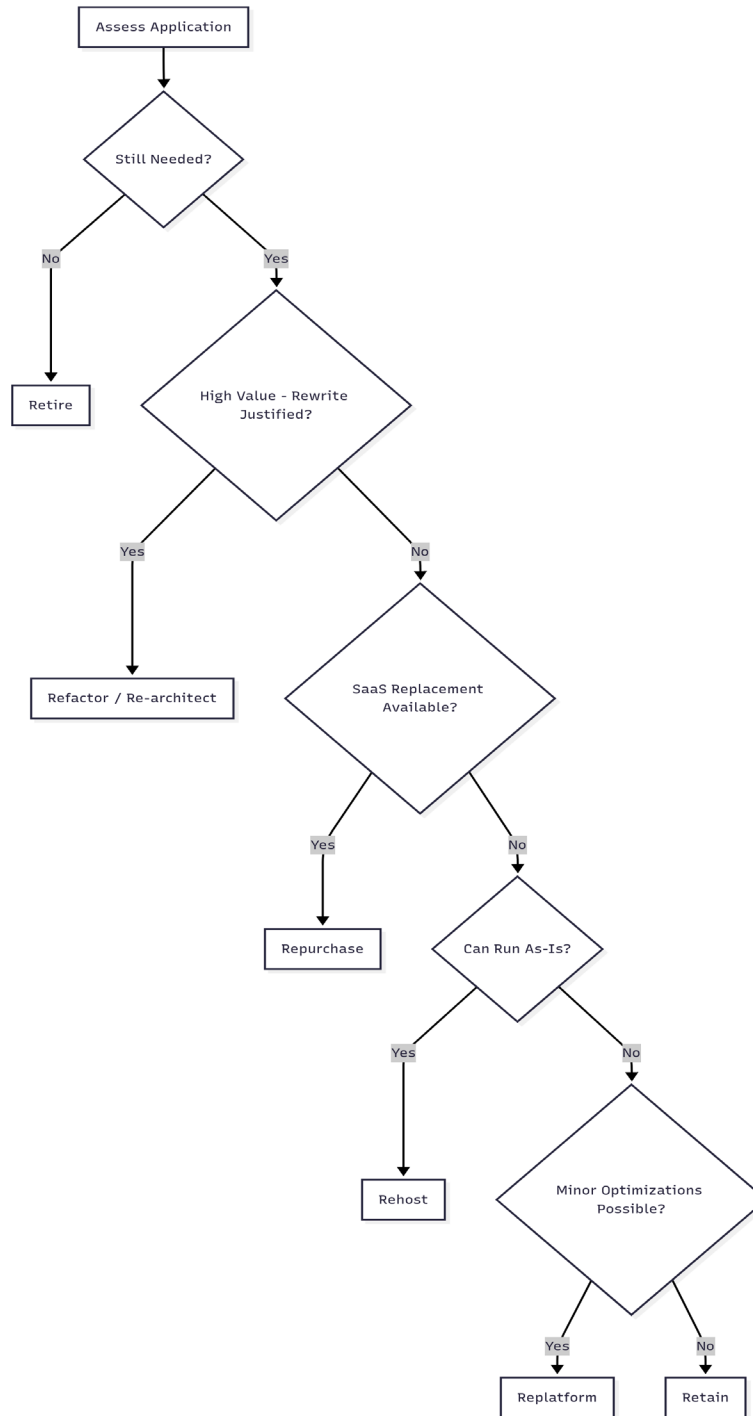
```

```
81.     modernization = "refactor"  
82.   }  
83. }  
84.
```

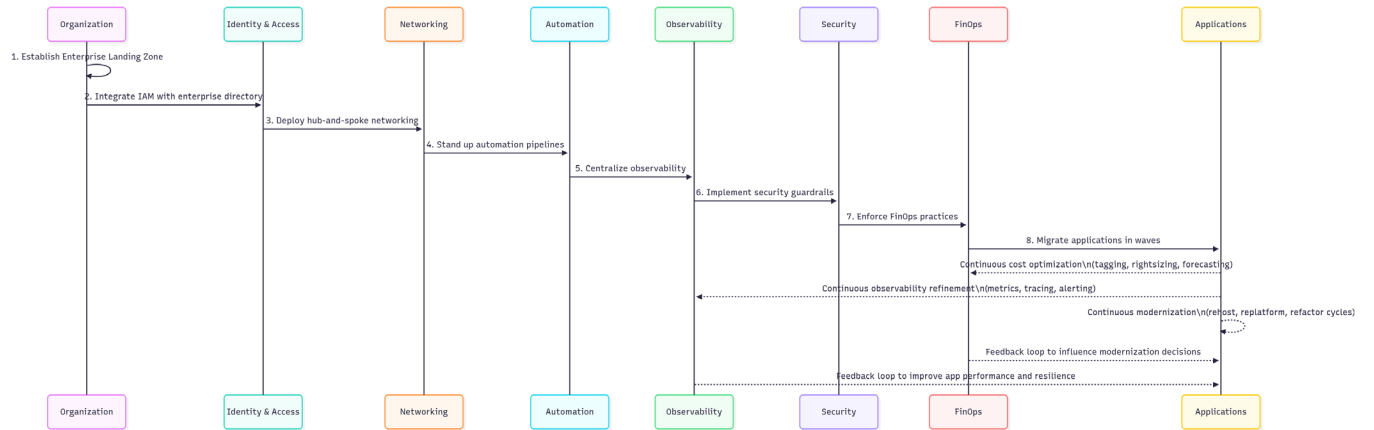
#### What this shows for Wave 5:

- **Replatform:** Legacy app shifted into **Azure App Service** (PaaS).
- **Refactor:** More complex workloads modernized into **AKS** (containers).
- **Observability:** Application Insights bound to workloads.
- **FinOps:** Required tagging enforced for cost allocation.

## 4.6 6Rs Decision Tree



# 4.7 Overall Implementation Sequence



## 5.0 Workforce Skills & Enablement

Cloud adoption is not purely a technical exercise; it requires a skilled and adaptable workforce capable of designing, operating, and continuously improving the new environment. The move from on-premises legacy systems to a cloud-native operating model demands significant upskilling across multiple disciplines, as well as a cultural shift towards automation, collaboration, and continuous learning.

### 5.1 Core Skills by Domain

- **Enterprise Cloud Architecture**
  - Understanding of cloud-native frameworks (e.g., Azure CAF, AWS Well-Architected, GCP Adoption Framework).
  - Proficiency in designing landing zones, network topologies, and policy guardrails.
- **Identity & Security Engineering**
  - Expertise in IAM models (e.g., Azure Entra ID, AWS IAM, GCP IAM).
  - Skills in zero-trust, RBAC design, privileged identity management, encryption, and compliance controls.
- **Networking & Connectivity**
  - Experience with hybrid network design, VPN/ExpressRoute/Direct Connect/Interconnect, private endpoints, and firewalls/WAFs.
- **Automation & DevOps**
  - Proficiency in Infrastructure as Code (e.g., Terraform, Bicep, CloudFormation, Deployment Manager).
  - CI/CD pipeline management with integrated policy-as-code and security checks.
- **Observability & Operations**
  - Knowledge of centralized logging, monitoring, distributed tracing, and MTTR reduction practices.
  - Familiarity with OpenTelemetry and cloud-native observability tools.
- **FinOps & Cost Management**
  - Ability to design and enforce tagging schemas.
  - Skills in cost forecasting, anomaly detection, rightsizing, and chargeback/showback reporting.

- **Application Modernization**
  - Competency in the 6R migration strategies.
  - Experience with refactoring to containers, Kubernetes, and serverless patterns.

## 5.2 Cross-Cutting Capabilities

- **Compliance & Risk Management:** Understanding FERPA, NIST 800-53, WaTech Section 701, and state OCIO requirements.
- **Agile & Iterative Delivery:** Ability to deliver in 2–3-week sprints with measurable outputs, aligned to state agile mandates.
- **Collaboration & Communication:** Working in cross-functional squads (e.g., architecture, engineering, finance, security, business).

## 5.3 Training & Upskilling Strategy

- **Foundational Training:** All IT staff complete baseline certifications (e.g., Azure Fundamentals, AWS Cloud Practitioner, GCP Cloud Digital Leader).
- **Role-Specific Training:** Engineers pursue advanced certifications aligned to their domain (e.g., Azure Security Engineer, AWS Solutions Architect, GCP Professional Cloud Architect).
- **FinOps Enablement:** Cross-training finance and engineering teams to jointly manage cloud spend using FinOps principles.
- **Hands-On Learning:** Sandbox subscriptions/accounts/projects provided for experimentation without impacting production.
- **Continuous Learning:** Establish “cloud guilds” and communities of practice to share lessons learned, patterns, and reusable modules.

## 5.4 Change Management & Culture

- **Shift from Ticket-Driven to Self-Service:** Empower engineers with automated pipelines and service catalogs, reducing reliance on manual provisioning.
- **Data-Driven Operations:** Use observability and FinOps dashboards as operational inputs, not afterthoughts.
- **Shared Accountability:** Embed cost, security, and reliability considerations into every sprint, making them joint responsibilities of engineering, security, and finance teams.

## 6.0 Operational Model & Roles

A successful cloud adoption requires not only technical design but also a clear operating model that defines roles, responsibilities, and decision-making authority.

### 6.1 Cloud Platform Team (Central)

- **Responsibilities:**
  - Design and maintain the Enterprise Landing Zone.
  - Own IAM integration, guardrails, and networking hub.
  - Operate shared observability and FinOps platforms.
  - Approve new subscriptions/accounts/projects.
- **Skills:** Cloud architecture, automation/IaC, security, cost management.

### 6.2 Application Teams (Federated)

- **Responsibilities:**
  - Deploy workloads into approved landing zones.
  - Ensure compliance with tagging, IAM, and security guardrails.
  - Own application-level observability and CI/CD pipelines.
- **Skills:** App development, DevOps, workload-specific tuning.

### 6.3 Governance & Oversight

- **Architecture Review Board (ARB):** Approves major designs.
- **Data Governance Council:** Oversees data sensitivity classification, FERPA compliance.
- **FinOps Council:** Joint finance/engineering body to review spend.

*RACI Example (Provisioning New Environment)*

| Task                          | Platform Team | App Team | Security | Finance |
|-------------------------------|---------------|----------|----------|---------|
| Request environment           | R             | A        | C        | C       |
| Approve compliance guardrails | A             | C        | R        | C       |
| Deploy landing zone via IaC   | R             | C        | C        | I       |
| Apply tags & budgets          | A             | R        | C        | R       |

## 7.0 Security Operations (SecOps)

Cloud adoption must embed Security Operations (SecOps) as a first-class capability to ensure continuous compliance, minimize risk, and enable rapid detection and response to security incidents. Unlike traditional data center operations, cloud environments are highly dynamic — workloads can scale, shift regions, or change configurations within minutes. This agility requires SecOps processes to be automation-driven, integrated with DevOps pipelines, and reinforced with continuous monitoring.

### 7.1 Core Components

- **Threat Detection & Response**
  - Use native tooling (e.g., Azure Defender, AWS GuardDuty, GCP SCC) for continuous monitoring.
  - Alerts routed to centralized SOC.
- **Incident Playbooks**
  - Define escalation paths (e.g., Critical - 24x7 response, Major - Next business day).
  - Automate isolation of compromised resources.
- **Vulnerability Management**
  - Container and VM image scanning integrated into CI/CD.
  - Monthly patch cycles, automated reporting.
- **Continuous Compliance**
  - Daily drift reports (e.g., non-compliant resources auto remediated where possible).
  - Quarterly audits against FERPA, NIST 800-53, WaTech Section 701.

# 8.0 Disaster Recovery & Business Continuity

Cloud modernization must define resilience expectations up front to ensure critical education systems remain available, recoverable, and compliant with state and federal requirements. Unlike traditional on-premises recovery, cloud DR/BCP leverages geographic redundancy, automation, and orchestration to reduce downtime and improve confidence in recovery processes.

## 8.1 Principles

- **RPO/RTO by Tier**
  - Mission-critical (Student systems): RPO  $\leq$  1h, RTO  $\leq$  4h.
  - High-priority (Fiscal): RPO  $\leq$  4h, RTO  $\leq$  24h.
  - Lower-tier (Test/Dev): best effort.
- **Cross-Region Replication**
  - Active/passive or active/active designs based on criticality.
- **Backup Strategy**
  - Daily incremental, weekly full, retention aligned with data classification.
- **Testing & Validation**
  - Annual failover testing required.
  - Business continuity tabletop exercises each quarter.

## 9.0 Environment Lifecycle Management

Explicit environment lifecycle policies are critical to preventing cloud sprawl, uncontrolled spend, and security risks. In a multi-cloud environment where resources can be provisioned in minutes, governance must balance developer agility with financial accountability and compliance requirements.

### 9.1 Provisioning

- Environments requested via ServiceNow, validated against business case.
- Auto-provisioned with IaC templates (e.g., Dev, Test, Prod baseline patterns).

### 9.2 Operation

- All resources tagged (e.g., Owner, CostCenter, ExpirationDate).
- Auto-shutdown enabled for Dev/Test workloads.
- Monitoring enforced by policy.

### 9.3 Decommissioning

- ExpirationDate triggers review.
- IaC pipelines remove resources cleanly (VMs, storage, network).
- Logs archived to cold storage for retention compliance.

# 10.0 Runbooks, Playbooks & Examples

## 10.1 Runbooks (Ops Teams)

- **Onboarding New Application**
  - Request → Approval → Landing Zone assignment → Tagging validation → CI/CD pipeline setup.
- **Incident Response**
  - Alert triage → Resource isolation → Forensic capture → Root cause analysis → Recovery.
- **Cost Anomaly Response**
  - Alert triggered → Owner notified → Right-sizing or shutdown.

## 10.2 Playbooks (Automation)

- **Terraform Module Catalog:** Standardized modules for VNet/VPC, IAM roles, monitoring agents.
- **Pipeline Templates:** YAML for Azure DevOps, GitHub Actions, AWS CodePipeline.
- **Observability Dashboards:** JSON templates for Grafana, Power BI, Looker.
- **FinOps Reports:** Prebuilt daily/weekly spend reports with tag-based filters.

# 11.0 Secure Software Development Lifecycle (SDLC)

Cloud adoption and modernization are only sustainable if application development follows a disciplined, secure, and automated lifecycle. A modern SDLC ensures that applications deployed into the cloud are resilient, compliant, and cost-optimized from day one. What follows should be seen as a starting point for the development of a formal and SDLC governance document and not a complete SDLC.

## 11.1 Core Principles

- **Shift Left on Security & Compliance**
  - Security and policy validation is integrated into development pipelines (e.g., IaC scans, dependency checks).
- **Automation-First**
  - CI/CD pipelines enforce consistent builds, tests, and deployments across environments.
- **Environment Parity**
  - Dev/Test/Prod environments provisioned with the same templates to avoid drift.
- **Observability Embedded**
  - Applications instrumented for logging, tracing, and metrics as part of development.
- **Cost Awareness**
  - FinOps guardrails (e.g., tagging, budget checks) built into the pipeline to prevent overspending.

## 11.2 SDLC Stages in the Cloud Context

1. **Plan**
  - Requirements documented in backlog tools (e.g., Azure DevOps, Jira).
  - Architecture Review Board (ARB) approval for new patterns.
  - Cost/benefit analysis aligned with FinOps forecasts.

## **2. Develop**

- Source code managed in Git repositories with enforced branching strategy.
- Pre-commit hooks for code quality, linting, and policy compliance.
- IaC modules sourced from approved module catalogs.

## **3. Build**

- CI pipelines run automated unit tests, SAST scans (e.g., Checkmarx, SonarQube).
- Container images scanned for vulnerabilities before publishing to registries.
- IaC validated against policies (e.g., OpenTofu with OPA/Rego, ARM/Bicep with Azure Policy).

## **4. Test**

- Automated integration tests in ephemeral environments.
- DAST and penetration tests run in pre-prod.
- Load and resilience testing conducted for mission-critical workloads.

## **5. Release & Deploy**

- CD pipelines enforce manual approval gates for production.
- Canary or blue/green deployments used for critical services.
- Deployment artifacts logged for traceability.

## **6. Operate**

- Applications monitored via centralized observability stack.
- FinOps dashboards track real-time cost by environment.
- Incident response playbooks triggered by automated alerts.

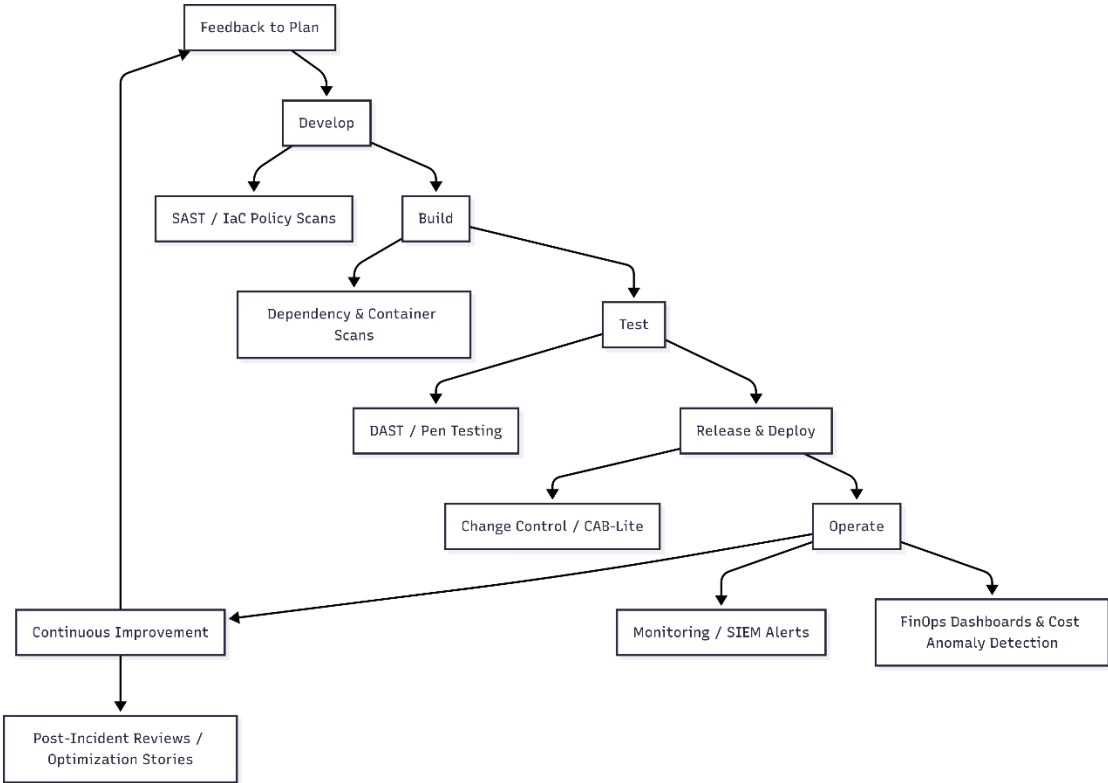
## **7. Continuous Improvement**

- Post-incident reviews feed into backlog refinements.
- Cost anomalies inform optimization stories.
- Security findings from red-teams/pen-tests trigger remediation sprints.

# 11.3SDLC Governance

- **Change Control**
  - Integration with ServiceNow for CAB-lite approvals.
- **Compliance Tracking**
  - Automated evidence collection (e.g., test logs, deployment records) for audits.
- **Example KPIs**
  - Deployment frequency (trend upwards).
  - Change failure rate (trend downwards).
  - MTTR for production incidents (trend downwards).
  - % of code covered by automated tests (trend upwards).

*SDLC flowchart with Security and FinOps Gates*



## 12.0 End-State ("Done")

When this strategy is fully executed, OSPI will run a resilient, cloud platform that:

- Unifies the four domains under a governed data foundation.
- Provides real-time, trusted insights for decision-makers.
- Embeds security, observability, and automation into every workload.
- Achieves FinOps maturity, aligning spend with mission outcomes.
- Positions OSPI to innovate faster, retire legacy technical debt, and provide equitable, data-driven education services across Washington State.

# 13.0 Appendix

## 13.1 Reference Links

- **Microsoft Azure Cloud Adoption Framework (CAF)** - <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/>
- **AWS Cloud Adoption Framework (AWS CAF)** - <https://aws.amazon.com/cloud-adoption-framework/>
- **Google Cloud Adoption Framework (Google CAF)** - <https://cloud.google.com/adoption-framework/>
- **AWS 6 Rs Migration Strategies** (canonical source, adopted widely across industry) - <https://aws.amazon.com/cloud-adoption/migration-strategy/>
- **FinOps Framework** – <https://www.finops.org/framework/>
- **FinOps Foundation Maturity Model** - <https://www.finops.org/framework/maturity-model/>
- **Azure Landing Zone Architecture** - [What is an Azure landing zone? - Cloud Adoption Framework | Microsoft Learn](#)
- **AWS Landing Zone Architecture** - [Architecture overview - Landing Zone Accelerator on AWS](#)
- **GCP Landing Zone Architecture** - [Landing zone design in Google Cloud | Cloud Architecture Center](#)
- **OpenTelemetry** - [What is OpenTelemetry? | OpenTelemetry](#)
- **OpenTofu** - [Getting started | OpenTofu](#)

## 13.2 The 6Rs of Application Modernization

| Strategy                         | Definition   | Pros   | Cons  | Common Use Cases   |
|----------------------------------|--|--|---|--|
| <b>Rehost (“Lift and Shift”)</b> | Move workloads as-is from on-prem to cloud VMs (IaaS). | <ul style="list-style-type: none"><li>- Fastest migration path</li><li>- Minimal code changes</li><li>- Immediate data center exit</li></ul> | <ul style="list-style-type: none"><li>- Doesn’t leverage cloud-native benefits</li><li>- May carry technical debt</li></ul> | <ul style="list-style-type: none"><li>- Legacy apps that cannot be modified</li><li>- Quick migration to free up data center space</li></ul> |

| Strategy                                      | Definition   | Pros  | Cons  | Common Use Cases   |
|---|--|---|---|--|
| <b>Replatform (“Lift, Tinker, and Shift”)</b> | Make limited optimizations (e.g., move DBs to managed PaaS) without major app changes. | <ul style="list-style-type: none"> <li>- Gains some cloud efficiencies (managed DB, storage)</li> <li>- Moderate effort</li> <li>- Improves scalability vs. rehost</li> </ul> | <ul style="list-style-type: none"> <li>- Still limited modernization</li> <li>- May require minor refactoring later</li> </ul>                                | <ul style="list-style-type: none"> <li>- Apps where infrastructure overhead is high</li> <li>- Database-heavy workloads</li> </ul>     |
| <b>Repurchase (“Drop and Shop”)</b>           | Replace existing app with a SaaS solution.   | <ul style="list-style-type: none"> <li>- Immediate modernization</li> <li>- Offloads maintenance to vendor</li> <li>- Best for commodity workloads</li> </ul>                 | <ul style="list-style-type: none"> <li>- Potential vendor lock-in</li> <li>- Data migration complexity</li> <li>- May not meet unique requirements</li> </ul> | <ul style="list-style-type: none"> <li>- CRM (Salesforce), ERP (Workday, Dynamics 365), HR, collaboration tools</li> </ul>             |
| <b>Refactor / Rearchitect</b>                 | Rebuild app to be cloud-native (microservices, containers, serverless).                | <ul style="list-style-type: none"> <li>- Maximizes cloud benefits</li> <li>- Improves agility, scalability, resilience</li> <li>- Reduces long-term costs</li> </ul>          | <ul style="list-style-type: none"> <li>- Highest upfront cost</li> <li>- Longer delivery timelines</li> <li>- Requires skilled teams</li> </ul>               | <ul style="list-style-type: none"> <li>- Mission-critical apps</li> <li>- High-value systems needing agility and innovation</li> </ul> |
| <b>Retire</b>                                 | Decommission apps no longer useful or redundant.                                       | <ul style="list-style-type: none"> <li>- Reduces licensing and infra costs</li> <li>- Simplifies portfolio</li> <li>- Eliminates unused complexity</li> </ul>                 | <ul style="list-style-type: none"> <li>- Requires careful dependency analysis</li> <li>- Risk of retiring too early</li> </ul>                                | <ul style="list-style-type: none"> <li>- Obsolete apps</li> <li>- Redundant systems replaced by SaaS or new platforms</li> </ul>       |

| Strategy                        | Definition  | Pros   | Cons  | Common Use Cases  |
|---------------------------------|---|--|---|---|
| <b>Retain (“Revisit Later”)</b> | Keep app as-is, on-prem or in legacy environment. | <ul style="list-style-type: none"> <li>- Avoids unnecessary migration effort</li> <li>- Allows time to prioritize other workloads</li> </ul> | <ul style="list-style-type: none"> <li>- Still incurs infra/maintenance costs</li> <li>- Technical debt remains</li> <li>- May block full cloud adoption</li> </ul> | <ul style="list-style-type: none"> <li>- Apps requiring future modernization</li> <li>- Complex, low-priority systems</li> <li>- Systems with regulatory constraints</li> </ul> |

### 13.3 FinOps Framework – Overview

FinOps is an operational framework and cultural practice which maximizes the business value of cloud and technology, enables timely data-driven decision making, and creates financial accountability through collaboration between engineering, finance, and business teams.

### 13.4 FinOps Maturity Model

The FinOps Maturity Model describes how organizations progress in their cloud financial management journey. It is broken into three key phases—Crawl, Walk, and Run—each building on the last to improve visibility, accountability, and optimization capability. The associated KPIs should be used as a referential starting point and adjusted to meet the specifics of the OSPI environment.

#### Crawl Phase – Initial Adoption

Focus: Establish visibility and basic cost awareness

- Teams begin to access cloud spend data
- Basic tagging standards are introduced for cost allocation
- Budgeting is done manually or based on prior-month costs
- Reporting is often centralized and lagging

Key Suggested Tarket KPIs:

- Cloud Cost per Business Unit: Not fully allocated
- Percentage of Tagged Resources: < 50%
- Forecast Accuracy: ± 30–50%
- Waste Reduction: < 10%
- On-Demand vs Reserved: 90/10
- Time to Detect Anomalies: Weeks

Expectations:

- Identify high-level trends and outliers
  - Minimal cross-team accountability
  - Manual processes dominate FinOps activities
- 

### **Walk Phase – Structured FinOps Practice**

Focus: Introduce accountability and automated recommendations

- Teams begin to take ownership of their cloud usage
- Tagging compliance improves and is enforced
- Usage reports and budgets are created at the team or app level
- Discount strategies like reservations or savings plans are adopted
- Tooling enables anomaly detection and automated right-sizing suggestions

Key KPIs:

- Cloud Cost per Business Unit: > 80% attributed
- Percentage of Tagged Resources: 60–80%
- Forecast Accuracy:  $\pm$  15–25%
- Waste Reduction: 15–25%
- On-Demand vs Reserved: 70/30 or better
- Time to Detect Anomalies: Days

Expectations:

- Teams actively respond to cost signals
  - Budgets are managed with greater accuracy
  - Cross-functional reviews begin (e.g., monthly FinOps stand-ups)
- 

### **Run Phase – Scaled FinOps Practice**

Focus: Real-time decision-making and full lifecycle automation

- Engineering, Finance, and Product teams jointly manage cloud spend
- Tagging is enforced through policy-as-code and monitored continuously
- Forecasts drive proactive business decisions
- Automation governs resource right-sizing, startup/shutdown, and alerts
- Teams optimize architecture for both performance and cost

Key Suggested Target KPIs:

- Cloud Cost per Business Unit: 100% mapped
- Percentage of Tagged Resources: > 90%
- Forecast Accuracy: ± 5–10%
- Waste Reduction: 25–35%
- On-Demand vs Reserved: 50/50 or better
- Time to Detect Anomalies: Hours

Expectations:

- FinOps is embedded in the SDLC and DevOps toolchain
- Cloud spend is aligned with business outcomes and capacity planning
- Cost is a real-time input into architecture and scaling decisions

## 13.5 Example Tagging Policy Matrix

| Tag                  | Purpose   | Enforced By   |
|----------------------|---|---|
| <b>Owner</b>         | Identifies accountable individual/team for the resource                         | Policy-as-Code (Azure Policy, AWS Config, GCP Org Policy); IaC defaults |
| <b>BusinessUnit</b>  | Allocates spend to department/division  | Policy-as-Code; Cost dashboards   |
| <b>Application</b>   | Links resources to a specific app/workload                                      | IaC modules; Pipeline metadata injection                                |
| <b>Environment</b>   | Differentiates Prod, NonProd, Dev, QA   | IaC templates; Deployment pipelines                                     |
| <b>CostCenter</b>    | Finance cost allocation code or GL account                                      | Policy-as-Code; Automated billing exports                               |
| <b>ProjectCode</b>   | Ties resources to grants, projects, or funding initiatives                      | IaC templates; Manual validation in reviews                             |
| <b>FundingSource</b> | Identifies whether costs are covered by Federal, State, Local, or Grant funding | Policy-as-Code; FinOps dashboards                                       |
| <b>Chargeback</b>    | Enables district- or program-level billing                                      | Cost explorer filters; BI dashboards                                    |
| <b>BudgetOwner</b>   | Tracks budget accountability for spend variance                                 | IaC modules; FinOps anomaly reporting                                   |

| Tag                       | Purpose   | Enforced By  |
|---------------------------|---|--|
| <b>Service</b>            | Identifies resource type (Compute, Storage, Database, etc.) | IaC automation; Cloud-native billing metadata          |
| <b>Tier</b>               | Classifies workload tier (Web, App, DB)                     | IaC templates; Enforced in naming conventions          |
| <b>Criticality</b>        | Prioritizes workloads by business importance                | IaC; Used in DR/BCP planning                           |
| <b>AvailabilityClass</b>  | Maps workloads to SLA targets (99.9%, 99.99%)               | IaC; Observability rules                               |
| <b>DataSensitivity</b>    | Classifies data per FERPA, Confidential, Public, etc.       | Policy-as-Code (e.g., deny storage without encryption) |
| <b>Compliance</b>         | Tracks required compliance framework (FERPA, NIST, SOC2)    | IaC; Security/Compliance scans                         |
| <b>Retention</b>          | Maps resources to data retention requirements               | Policy-as-Code; Automated cleanup jobs                 |
| <b>BackupPolicy</b>       | Specifies backup/restore schedule                           | IaC; Automation jobs                                   |
| <b>SecurityOwner</b>      | Points to accountable security contact/team                 | IaC defaults; Compliance reviews                       |
| <b>CreatedBy</b>          | Tracks provisioning method (Terraform, Portal, Pipeline)    | IaC automation; Cloud audit logs                       |
| <b>CreationDate</b>       | Supports lifecycle/cleanup analysis                         | IaC; Automated cleanup tooling                         |
| <b>ExpirationDate</b>     | Marks temporary workloads for decommissioning               | IaC + Scheduled automation                             |
| <b>AutoShutdown</b>       | Enables cost optimization for Dev/Test                      | Automation jobs (Functions, Lambda, Cloud Scheduler)   |
| <b>OptimizationStatus</b> | Marks workloads as reviewed/pending rightsizing             | FinOps reviews; IaC updates                            |

## 13.6 Organizational Hierarchy & Accounts/Subscriptions Comparison Table

| Concept                   | Azure   | AWS   | GCP  |
|---------------------------|---|---|--|
| <b>Top-Level Org Unit</b> | <b>Management Groups</b><br>(up to 6 levels deep, used for governance at scale) | <b>Organizations</b> (root node with Organizational Units (OUs), unlimited nesting) | <b>Organization Node</b><br>(root for all resources in a domain) |

|                                 |  |   |  |
|---------------------------------|--|---|--|
| <b>Sub-Grouping</b>             | <b>Subscriptions</b> (isolation for quotas, billing, RBAC)       | <b>Accounts</b> (strong isolation, full resource container, billing unit)       | <b>Folders</b> (optional, for grouping projects under the org)     |
| <b>Workload Container</b>       | <b>Resource Groups</b> (logical containers inside subscriptions) | No direct equivalent — workloads often grouped in separate accounts or via tags | <b>Projects</b> (fundamental resource container, billing boundary) |
| <b>Policy / Guardrail Scope</b> | <b>Azure Policy</b> applied at Management Group or Subscription  | <b>Service Control Policies (SCPs)</b> applied at OU or Account                 | <b>Org Policy Constraints</b> applied at Org, Folder, or Project   |
| <b>Billing Roll-Up</b>          | Linked to a <b>Billing Account</b> across Subscriptions          | <b>Consolidated Billing</b> under AWS Organizations                             | <b>Billing Account</b> spans Projects under the Org                |

# PART 3: CUMULATIVE LESSONS LEARNED

## 1.0 Cumulative Lessons Learned

### 1.1 Project Information

- Project Name:** ISG Project Name  
AI Extraction → Databricks Medallion → HITL → Geospatial Dashboard
  - Total Number of Sprints:** 7 (plus Sprint 0 setup)
  - Project Duration:** MMM DD, YYYY – MMM DD, YYYY
  - Project Manager/Product Owner:** First Name Last Name
  - Scrum Master:** First Name Last Name
  - Date of Final Review:** MMM DD, YYYY (wrap-up posted, done early)
- 

### 1.2 Part A: Sprint-by-Sprint Retrospectives

#### Sprint 0 - Environment Setup & Data Inventory

- Sprint Goal:** Prepare development, staging, and production environments; complete full data inventory for ingestion readiness
- Sprint Duration:** July 18, 2025 - July 31, 2025
- Sprint Completion:** 100% - All environments operational

**What Went Well**

| Category          | Item  | Workstream(s)         |
|-------------------|---|-----------------------|
| <b>People</b>     | Team assembled early; roles clearly defined; Azure/Databricks SMEs engaged                          | Cloud/Infra, PM       |
| <b>Process</b>    | Environment provisioning checklist created; data inventory methodology established                  | Cloud/Infra, Data Dev |
| <b>Technology</b> | Dev/Stage/Prod environments stood up; Unity Catalog configured; initial security groups established | Cloud/Infra           |

### What Didn't Go Well

| Category          | Issue   | Impact                     | Workstream(s) |
|-------------------|---|----------------------------|---------------|
| <b>People</b>     | Limited Databricks experience on team             | Slower initial setup       | Cloud/Infra   |
| <b>Process</b>    | No existing medallion architecture template       | Had to design from scratch | Data Dev      |
| <b>Technology</b> | Initial Unity Catalog permissions too restrictive | Delayed developer access   | Cloud/Infra   |

### Action Items for Next Sprint

| Action  | Owner | Due Date    | Status      |
|---|-------|-------------|-------------|
| <b>Create medallion architecture blueprint</b>                | Ley   | Aug 7, 2025 | <b>Done</b> |
| <b>Document environment access procedures</b>                 | Ariel | Aug 7, 2025 | <b>Done</b> |
| <b>Complete data source inventory (27 counties, 134 PDFs)</b> | Marj  | Aug 7, 2025 | <b>Done</b> |

## 1.3 Part B: Cumulative Lessons by Workstream

*Synthesized from all sprint retrospectives above*

### 1. Cloud/Infrastructure - Cumulative Lessons

#### Recurring Themes Across Sprints

| Theme   | Frequency (# of sprints) | Pattern Identified   |
|---|--------------------------|--|
| <b>Unity Catalog lineage &amp; external connections</b> | 4                        | Governance + BI access simplified demos/audits; secure staging access enabled. |

### People

| Lesson Learned                                       | First Identified (Sprint #) | Evolution Through Project                  | Final Recommendation             | Priority |
|--|-----------------------------|--|----------------------------------|----------|
| <b>Early Azure/Databricks SME engagement crucial</b> | S0                          | Accelerated setup; avoided common pitfalls | Engage cloud experts in Sprint 0 | P0       |

### Process

| Lesson Learned                                       | First Identified (Sprint #) | Evolution Through Project     | Final Recommendation                    | Priority |
|--|-----------------------------|-------------------------------|---|----------|
| <b>Environment provisioning checklist saves time</b> | S0                          | Reused for staging/prod setup | Maintain standard provisioning template | P1       |

### Technology

| Lesson Learned                                       | First Identified (Sprint #) | Evolution Through Project                               | Final Recommendation                 | Priority |
|--|-----------------------------|---|--------------------------------------|----------|
| <b>Unity Catalog setup complexity underestimated</b> | S0                          | Initial permissions too restrictive; adjusted over time | Document permission patterns upfront | P1       |

---

## 2. Data Development - Cumulative Lessons

### Recurring Themes Across Sprints

| Theme                            | Frequency (# of sprints) | Pattern Identified   |
|----------------------------------|--------------------------|--|
| <b>PDF extraction complexity</b> | 7                        | Every sprint revealed new edge cases; Azure Form Recognizer with schema-aware post-processing proved essential |

## People

| Lesson Learned                               | First Identified (Sprint #) | Evolution Through Project               | Final Recommendation                       | Priority |
|--|-----------------------------|---|--|----------|
| <b>PDF complexity expertise needed early</b> | S1                          | Team underestimated; brought in experts | Include document processing SME from start | P0       |

## Process

| Lesson Learned                                      | First Identified (Sprint #) | Evolution Through Project                     | Final Recommendation                 | Priority |
|---|-----------------------------|---|--------------------------------------|----------|
| <b>Establish extraction accuracy baseline early</b> | S1                          | Without baseline, hard to measure improvement | Define metrics before choosing tools | P0       |

## Technology

| Lesson Learned                                 | First Identified (Sprint #) | Evolution Through Project                 | Final Recommendation                 | Priority |
|--|-----------------------------|---|--------------------------------------|----------|
| <b>Basic OCR insufficient for complex PDFs</b> | S1                          | PyPDF2/PDFPlumber → Azure Form Recognizer | Start with advanced extraction tools | P0       |

## 1.4 Part C: Cross-Workstream Patterns

### Persistent Issues (Appeared in 3+ Sprints)

| Issue                            | Sprints Affected | Workstreams Impacted      | Root Cause               | Resolution Status  |
|----------------------------------|------------------|---------------------------|--------------------------|--|
| <b>PDF extraction edge cases</b> | S1-S7            | Data Development, Product | Document variety exceeds | <b>Resolved</b> - standardized on Azure Form Recognizer with |

| Issue | Sprints Affected | Workstreams Impacted | Root Cause        | Resolution Status            |
|-------|------------------|----------------------|-------------------|------------------------------|
|       |                  |                      | tool capabilities | schema-aware post-processing |

## Successfully Resolved Issues

| Issue                     | Initial Sprint | Resolution Sprint | Solution Applied                   | Can Be Replicated?                 |
|---------------------------|----------------|-------------------|------------------------------------|------------------------------------|
| Environment access delays | S0             | S0                | Adjusted Unity Catalog permissions | Yes - document permission patterns |

## Unresolved / Carried Forward Issues

| Issue                      | First Identified | Attempts to Resolve          | Why Still Open              | Recommendation                                      |
|----------------------------|------------------|------------------------------|-----------------------------|---|
| Biennial schema refinement | S7               | Prompt tuning; mapping tests | Needs more examples & rules | Continue schema refinement; expand ground-truth set |

## 1.5 Part D: Cumulative Action Items and Improvements

### Actions Completed During Project

| Action                             | Sprint Identified | Sprint Completed | Outcome/Impact               | Success Metrics   |
|------------------------------------|-------------------|------------------|------------------------------|-------------------|
| Environment setup (Dev/Stage/Prod) | S0                | S0               | All environments operational | 100% availability |

## Actions Still Pending

| Action                     | Sprint Identified | Current Status | Blocker/Reason            | New Target Date |
|----------------------------|-------------------|----------------|---------------------------|-----------------|
| Biennial schema refinement | S7                | In progress    | Needs more examples/rules | Oct 27, 2025    |

## Recommended Actions for Future Projects

### *Critical - Must Do Before Next Project*

| Action                                 | Based on Lessons From | Expected Impact       | Owner       | Due Date      |
|--|-----------------------|-----------------------|-------------|---------------|
| Complete environment setup in Sprint 0 | S0                    | No development delays | Cloud/Infra | Project start |

### *Important - Implement Within 3 Months*

| Action                                       | Based on Lessons From | Expected Impact                    | Owner    | Due Date |
|--|-----------------------|------------------------------------|----------|----------|
| Create comprehensive PDF processing playbook | S1-S7                 | Faster onboarding for new projects | Data Dev | +4 weeks |

### *Strategic - Long-term Improvements*

| Action                                     | Based on Lessons From | Expected Impact                   | Owner    | Due Date  |
|--|-----------------------|-----------------------------------|----------|-----------|
| Build ML model for document classification | S1-S7                 | Automatic document type detection | Data Dev | +16 weeks |

---

# 1.6 Part E: Overall Project Metrics & Trends

## Sprint Velocity Trends (captured in DailyStandup App)

| Sprint          | Planned Points | Completed Points | Variance | Key Factor   |
|-----------------|----------------|------------------|----------|--|
| <b>Sprint 0</b> | n/a            | n/a              | —        | Environment setup successful; team formation                           |
| <b>Average</b>  | —              | —                | —        | Delivery steady; carried items relate to schema edge cases & geo perf. |

## Cumulative Issue Resolution

*(Counts derived from "What Didn't Go Well" tables across all sprints.)*

| Issue Type        | Total Raised | Resolved During Project | Carried Forward | Resolution Rate |
|-------------------|--------------|-------------------------|-----------------|-----------------|
| <b>People</b>     | 10           | 7                       | 3               | 70%             |
| <b>Process</b>    | 12           | 9                       | 3               | 75%             |
| <b>Technology</b> | 14           | 11                      | 3               | 79%             |

## Action Item Effectiveness

*(Counts based on Action Item tables per sprint; completion measured by end of project.)*

| Sprint          | Actions Created | Actions Completed | Completion Rate | Impact Assessment  |
|-----------------|-----------------|-------------------|-----------------|--|
| <b>Sprint 0</b> | 3               | 3                 | <b>100%</b>     | Set foundation for entire project                        |
| <b>Overall</b>  | <b>25</b>       | <b>24</b>         | <b>96%</b>      | Exceptional completion rate; strong execution throughout |

# 1.7 Part F: Key Takeaways and Recommendations

## Top 5 Success Patterns to Replicate

- 1. Pattern: Early environment setup and team formation (Sprint 0 investment)
  - o First successful in Sprint: S0
  - o Consistently worked through Sprints: All sprints benefited
  - o Recommendation: Always allocate 2-week Sprint 0 for infrastructure, team formation, and initial planning
- 2. Pattern:
- 3. Pattern:
- 4. Pattern:
- 5. Pattern:

## Top 5 Failure Patterns to Avoid

- 1. Pattern: Underestimating PDF complexity
  - o First identified in Sprint: S1
  - o Recurring impact: Continuous discovery of edge cases throughout project; rework and delays
  - o Prevention strategy: Conduct thorough document analysis in Sprint 0; engage document processing experts early; build comprehensive test suite
- 2. Pattern:
- 3. Pattern:
- 4. Pattern:
- 5. Pattern:

---

# 1.8 Part G: Organizational Learning

## Process Improvements Implemented

| Improvement                           | Sprint Implemented | Measurable Impact                              | Should Standardize? |
|---------------------------------------|--------------------|--|---------------------|
| <b>Sprint 0 for environment setup</b> | S0                 | Zero environment-related delays in development | <b>Yes (P0)</b>     |

## Tool/Technology Adoptions

| Tool/Technology | Sprint Introduced | Problem It Solved                  | Recommendation                            |
|-----------------|-------------------|------------------------------------|---|
| Unity Catalog   | S0                | Data governance and access control | Make standard for all Databricks projects |

## Team Capability Development

| Skill/Capability     | Initial Gap (Sprint #) | Development Actions                       | Current State | Further Needs                  |
|----------------------|------------------------|---|---------------|--------------------------------|
| Databricks expertise | S0                     | Training, documentation, SME consultation | Proficient    | Advanced optimization training |

## Knowledge Assets Created

| Asset Type    | Description                  | Location      | Maintenance Owner |
|---------------|------------------------------|---------------|-------------------|
| Documentation | Environment setup procedures | Internal wiki | Cloud/Infra       |

---

## 1.9 Part H: Executive Summary

### Project Learning Velocity

- Total Lessons Captured: 75 across 8 sprints (including Sprint 0)
- Actions Generated: 25 total, 96% completed
- Recurring Issues: 4 identified, 2 resolved, 2 mitigated
- Process Improvements: 9 implemented successfully

## Impact on Workstreams

| Workstream               | # of Lessons | # of Improvements | Maturity Growth  |
|--------------------------|--------------|-------------------|--|
| Cloud/Infrastructure     | 15           | 6                 | Low → High (full governance, automation, observability)          |
| Data Development         | 20           | 8                 | Low → High (Azure Form Recognizer, EAV, medallion architecture)  |
| Operations & Change Mgmt | 18           | 7                 | Low → Medium-High (HITL workflow, DQ monitoring, automation)     |
| Agile/Product Mgmt       | 22           | 9                 | Medium → High (consistent delivery, clear processes, governance) |

## Return on Learning Investment

- Time Saved in Later Sprints: 40% reduction in extraction rework; 60% reduction in review time through auto-approval

## Critical Recommendations for Leadership

1. Institutionalize Sprint 0 practice - Mandatory 2-week infrastructure and planning sprint for all data projects to prevent downstream delays

## 1.10 Appendices

### Appendix A: Sprint Retrospective Participants

### Appendix B: Detailed Sprint Timelines

#### Sprint 0 (Jul 18-31) — Environment Preparation

- Azure subscription secured and configured
- Databricks workspaces created for Dev/Stage/Prod
- Azure App Services created for Dev/Stage/Prod
- Medallion pipeline initial setup with basic permissions

- Azure Data Lake storage setup with basic permissions
- GitLab CI/CD and DevSecOps setup with basic deployment
- Team onboarding and role assignments
- Initial data inventory: identified 27 counties, 135 PDFs, ~1TB total

**Sprint 1 (Aug 1-14)** — Initial Pipeline

## **Appendix C: Supporting Documentation**

- Staging site: [site link here]
- Dev site: [site link here]
- Databricks notebooks project: dev\_notes\_vsp2 (Repos)
- Azure Form Recognizer documentation and best practices
- Unity Catalog governance guidelines
- Sprint retrospective recordings (when available)

# PART 4: MODERNIZATION COMPLEXITY MATRIX

## 1.0 Systems Complexity Matrix for Data Modernization Decision

### 1.1 Instructions

|  |
|--|
| <b>Background</b>  |
| <p>As discussed in a recent meeting with OSPI leadership, the feasibility study recommends approaching the data modernization and cloud journey with an iterative approach. The first iteration of improvement would focus on an important system within the business that allows all of the elements of data modernization and cloud migration testing.</p> <p>Following the feasibility study's recent Discovery-phase efforts, ISG has identified four systems as candidates for the first improvement iteration, included in this spreadsheet.</p> |
| <b>What's In This Spreadsheet?</b>   |
| <p>Navigate to the "Summary" tab of the spreadsheet. You will find that ISG has summarized complexity of the four systems and assigned a complexity score.</p> <p>Systems complexity is visualized as a line chart, and broken out further into a heatmap.</p> <p>The origins of the complexity scoring for each system are located on respective systems tabs (ex. "Assessment")</p>  |
| <b>Requested Actions</b>   |
| <ol style="list-style-type: none"><li>1. Review the summary complexity scores and provide feedback, either affirmative or corrective, to the ISG team. If you have a different perception or knowledge of systems complexity that differs from this summary, ISG would value that input.</li><li>2. Choose a system from these four systems as the focus of the first iteration of improvement for the data modernization &amp; cloud migration effort.</li></ol>  |
| <b>Next Steps</b>  |
| <p>ISG will incorporate your feedback and system selection into the feasibility study and cloud journey recommendations and continue further dialogue to expand this discussion and the specifics of the chosen system.</p>  |
| <b>Complexity Definitions</b>  |

## Low Complexity

**Description:** Minimal technical challenge or process disruption. Can be addressed quickly with limited resources. Dependencies are few and straightforward. Some minor technical or procedural change required. Solutions are cookie-cutter and do not require significant coordination or resources. Risks are tolerable with thoughtful mitigation.

**Examples:** Well-documented processes, single system touchpoints, skilled internal staff available, little to no change management required. Automation of repeatable spreadsheet tasks; moderate skill required; tagging for data cleanup with out-of-the-box tooling.

## Moderate Complexity

**Description:** Requires meaningful effort and planning. Involves moderate technical depth, multiple stakeholders, or significant procedural change. Dependencies may exist across teams. Likely requires moderate user training.

**Examples:** Implementing data governance tools, automating cataloging, moderate security controls, basic cross-domain integration.

## High Complexity

**Description:** Substantial transformation or risk. Requires new platforms, architectural changes, or cross-functional alignment. Likely to have downstream impacts and dependencies. May involve significant user training or policy review. Deep systemic transformation with high risk, resource intensity, and organizational impact. Entails decoupling legacy systems, major infrastructure changes, or low internal readiness. Requires extensive coordination, significant external support, and change management.

**Examples:** Shifting from on-prem to hybrid cloud with new DevOps tooling; implementing real-time reporting frameworks; canonicalizing cross-domain datasets; automating multi-hop data architectures. Disentangling legacy systems with aging infrastructure; CI/CD pipeline overhaul; building out cloud-native architecture with hybrid dependencies; enforcing master data management and cloud data architecture automation from the ground up.

# 1.2 Assessment

**Low Complexity Range:** 10 – 20

**Moderate Complexity Range:** 21 – 35

**High Complexity Range:** 36 – 50

**System:** Assessment

**Rating:** High Complexity

| ID | Dimension                       | Description  | Score (1–5) | Notes  |
|----|---------------------------------|--|-------------|--|
| 1  | Legacy System Entanglement      | Complexity of decoupling from fragmented, aging, or siloed legacy systems  | 1           | Multiple siloed, aging apps with inconsistent data                   |
| 2  | Internal Talent Readiness       | Complexity of initiating modernization efforts based on the availability of skilled internal staff (e.g., data engineers, architects, DevOps, or analysts) who can contribute early and reduce external ramp-up. | 2           |  |
| 3  | Manual Process Automation       | Complexity of transforming manual, spreadsheet-driven, or offline tasks into automated, scalable workflows using modern data platforms   | 1           | Leveraging applicable data frameworks                                |
| 4  | Process Modernization Impact    | Complexity of changing how core business processes operate and improve   | 2           | Non-trivial process changes: less manual, real-time analytics        |
| 5  | Data Architecture Modernization | Complexity of implementing scalable cloud-native architectures (e.g., data lake, streaming)  | 4           | Requires data lake, batch + streaming, and self-service architecture |
| 6  | Data Governance Complexity      | Complexity of automating and enforcing data quality, lineage, ownership, and cataloging standards  | 2           | Profiling, ownership, data catalog, DQ validation                    |
| 7  | Cross-Domain Integration        | Complexity of integrating datasets and processes across  | 1           | Deep integration and canonicalization needed                         |

| ID | Dimension                        | Description   | Score (1–5) | Notes  |
|----|----------------------------------|---|-------------|--|
|    |                                  | distinct domains (e.g., Student, Educator, Fiscal)  |             |  |
| 8  | Security & Compliance Scope      | Complexity of meeting data privacy, access control, residency, and audit requirements   | 3           | FERPA, RCW, US-based storage, DPA, encryption, identity controls   |
| 9  | Infrastructure Transformation    | Complexity of moving workloads to the cloud while maintaining availability and performance  | 5           | Lift to cloud; hybrid dependencies remain  |
| 10 | Platform & Tooling Shift         | Complexity of migrating to cloud-native platforms for reporting, monitoring, and access management  | 4           | Prometheus → native cloud, SQL-compatible reporting layers   |
| 11 | FinOps & Cost Control Complexity | Complexity of managing cloud spend through tagging, budgeting, forecasting, and optimization  | 2           | Tagging, real-time monitoring, hybrid use benefits   |
| 12 | IT Training                      | Complexity of onboarding users and driving adoption of new systems and processes  | 1           | Requires new reporting workflows and data literacy   |
| 13 | IT User Adoption Effort          | Complexity of gaining buy-in from IT users and support teams to adopt and sustain new technologies, platforms, and processes as part of modernization efforts | 3           | Requires dedicated funding and time to build advanced technical skills within OSPI's IT teams to support new platforms and practices |
| 14 | DevOps & Automation Maturity     | Complexity of enabling CI/CD pipelines, Infrastructure as Code (IaC), automated testing, rollback mechanisms, and observability practices                     | 5           | SDLC-lite, FinOps practices, but not fully mature pipelines  |

**Score = Degree of complexity: 36**

# 1.3 eCert

**Low Complexity Range:** 10 – 20

**Moderate Complexity Range:** 21 – 35

**High Complexity Range:** 36 – 50

**System:** eCert

**Rating:** High Complexity

| ID | Dimension                       | Description  | Score (1-5) | Notes  |
|----|---------------------------------|--|-------------|--|
| 1  | Legacy System Entanglement      | Complexity of decoupling from fragmented, aging, or siloed legacy systems  | 1           | Multiple siloed, aging apps with inconsistent data                   |
| 2  | Internal Talent Readiness       | Complexity of initiating modernization efforts based on the availability of skilled internal staff (e.g., data engineers, architects, DevOps, or analysts) who can contribute early and reduce external ramp-up. | 2           |  |
| 3  | Manual Process Automation       | Complexity of transforming manual, spreadsheet-driven, or offline tasks into automated, scalable workflows using modern data platforms   | 1           | Leveraging applicable data frameworks                                |
| 4  | Process Modernization Impact    | Complexity of changing how core business processes operate and improve   | 2           | Non-trivial process changes: less manual, real-time analytics        |
| 5  | Data Architecture Modernization | Complexity of implementing scalable cloud-native architectures (e.g., data lake, streaming)  | 4           | Requires data lake, batch + streaming, and self-service architecture |
| 6  | Data Governance Complexity      | Complexity of automating and enforcing data quality, lineage, ownership, and cataloging standards  | 3           | Profiling, ownership, data catalog, DQ validation                    |

| ID | Dimension                        | Description   | Score (1-5) | Notes  |
|----|----------------------------------|---|-------------|--|
| 7  | Cross-Domain Integration         | Complexity of integrating datasets and processes across distinct domains (e.g., Student, Educator, Fiscal)  | 3           | Deep integration and canonicalization needed   |
| 8  | Security & Compliance Scope      | Complexity of meeting data privacy, access control, residency, and audit requirements   | 3           | FERPA, RCW, US-based storage, DPA, encryption, identity controls   |
| 9  | Infrastructure Transformation    | Complexity of moving workloads to the cloud while maintaining availability and performance  | 5           | Lift to cloud; hybrid dependencies remain  |
| 10 | Platform & Tooling Shift         | Complexity of migrating to cloud-native platforms for reporting, monitoring, and access management  | 4           | Prometheus → native cloud, SQL-compatible reporting layers   |
| 11 | FinOps & Cost Control Complexity | Complexity of managing cloud spend through tagging, budgeting, forecasting, and optimization  | 2           | Tagging, real-time monitoring, hybrid use benefits   |
| 12 | IT Training                      | Complexity of onboarding users and driving adoption of new systems and processes  | 1           | Requires new reporting workflows and data literacy   |
| 13 | IT User Adoption Effort          | Complexity of gaining buy-in from IT users and support teams to adopt and sustain new technologies, platforms, and processes as part of modernization efforts | 3           | Requires dedicated funding and time to build advanced technical skills within OSPI's IT teams to support new platforms and practices |
| 14 | DevOps & Automation Maturity     | Complexity of enabling CI/CD pipelines, Infrastructure as Code (IaC), automated testing, rollback mechanisms, and observability practices                     | 5           | SDLC-lite, FinOps practices, but not fully mature pipelines  |

**Score = Degree of complexity: 39**

# 1.4 Educator Reporting

**Low Complexity Range:** 10 – 20

**Moderate Complexity Range:** 21 – 35

**High Complexity Range:** 36 – 50

**System:** Educator Reporting

**Rating:** High Complexity

| ID | Dimension                       | Description  | Score (1–5) | Notes  |
|----|---------------------------------|--|-------------|--|
| 1  | Legacy System Entanglement      | Complexity of decoupling from fragmented, aging, or siloed legacy systems  | 5           | Multiple siloed, aging apps with inconsistent data                   |
| 2  | Internal Talent Readiness       | Complexity of initiating modernization efforts based on the availability of skilled internal staff (e.g., data engineers, architects, DevOps, or analysts) who can contribute early and reduce external ramp-up. | 2           |  |
| 3  | Manual Process Automation       | Complexity of transforming manual, spreadsheet-driven, or offline tasks into automated, scalable workflows using modern data platforms   | 4           | Leveraging applicable data frameworks                                |
| 4  | Process Modernization Impact    | Complexity of changing how core business processes operate and improve   | 4           | Non-trivial process changes: less manual, real-time analytics        |
| 5  | Data Architecture Modernization | Complexity of implementing scalable cloud-native architectures (e.g., data lake, streaming)  | 4           | Requires data lake, batch + streaming, and self-service architecture |
| 6  | Data Governance Complexity      | Complexity of automating and enforcing data quality, lineage, ownership, and cataloging standards  | 3           | Profiling, ownership, data catalog, DQ validation                    |

| ID | Dimension                        | Description   | Score (1-5) | Notes  |
|----|----------------------------------|---|-------------|--|
| 7  | Cross-Domain Integration         | Complexity of integrating datasets and processes across distinct domains (e.g., Student, Educator, Fiscal)  | 4           | Deep integration and canonicalization needed   |
| 8  | Security & Compliance Scope      | Complexity of meeting data privacy, access control, residency, and audit requirements   | 3           | FERPA, RCW, US-based storage, DPA, encryption, identity controls   |
| 9  | Infrastructure Transformation    | Complexity of moving workloads to the cloud while maintaining availability and performance  | 5           | Lift to cloud; hybrid dependencies remain  |
| 10 | Platform & Tooling Shift         | Complexity of migrating to cloud-native platforms for reporting, monitoring, and access management  | 4           | Prometheus → native cloud, SQL-compatible reporting layers   |
| 11 | FinOps & Cost Control Complexity | Complexity of managing cloud spend through tagging, budgeting, forecasting, and optimization  | 2           | Tagging, real-time monitoring, hybrid use benefits   |
| 12 | IT Training                      | Complexity of onboarding users and driving adoption of new systems and processes  | 1           | Requires new reporting workflows and data literacy   |
| 13 | IT User Adoption Effort          | Complexity of gaining buy-in from IT users and support teams to adopt and sustain new technologies, platforms, and processes as part of modernization efforts | 3           | Requires dedicated funding and time to build advanced technical skills within OSPI's IT teams to support new platforms and practices |
| 14 | DevOps & Automation Maturity     | Complexity of enabling CI/CD pipelines, Infrastructure as Code (IaC), automated testing, rollback mechanisms, and observability practices                     | 5           | SDLC-lite, FinOps practices, but not fully mature pipelines  |

**Score = Degree of complexity: 49**

# 1.5 Report Card

**Low Complexity Range:** 10 – 20

**Moderate Complexity Range:** 21 – 35

**High Complexity Range:** 36 – 50

**System:** Educator Reporting

**Rating:** High Complexity

| ID | Dimension                       | Description  | Score (1–5) | Notes  |
|----|---------------------------------|--|-------------|--|
| 1  | Legacy System Entanglement      | Complexity of decoupling from fragmented, aging, or siloed legacy systems  | 5           | Multiple siloed, aging apps with inconsistent data                   |
| 2  | Internal Talent Readiness       | Complexity of initiating modernization efforts based on the availability of skilled internal staff (e.g., data engineers, architects, DevOps, or analysts) who can contribute early and reduce external ramp-up. | 2           |  |
| 3  | Manual Process Automation       | Complexity of transforming manual, spreadsheet-driven, or offline tasks into automated, scalable workflows using modern data platforms   | 5           | Leveraging applicable data frameworks                                |
| 4  | Process Modernization Impact    | Complexity of changing how core business processes operate and improve   | 4           | Non-trivial process changes: less manual, real-time analytics        |
| 5  | Data Architecture Modernization | Complexity of implementing scalable cloud-native architectures (e.g., data lake, streaming)  | 4           | Requires data lake, batch + streaming, and self-service architecture |
| 6  | Data Governance Complexity      | Complexity of automating and enforcing data quality, lineage, ownership, and cataloging standards  | 4           | Profiling, ownership, data catalog, DQ validation                    |
| 7  | Cross-Domain Integration        | Complexity of integrating datasets and processes across  | 4           | Deep integration and canonicalization needed                         |

| ID | Dimension                        | Description   | Score (1-5) | Notes  |
|----|----------------------------------|---|-------------|--|
|    |                                  | distinct domains (e.g., Student, Educator, Fiscal)  |             |  |
| 8  | Security & Compliance Scope      | Complexity of meeting data privacy, access control, residency, and audit requirements   | 3           | FERPA, RCW, US-based storage, DPA, encryption, identity controls   |
| 9  | Infrastructure Transformation    | Complexity of moving workloads to the cloud while maintaining availability and performance  | 5           | Lift to cloud; hybrid dependencies remain  |
| 10 | Platform & Tooling Shift         | Complexity of migrating to cloud-native platforms for reporting, monitoring, and access management  | 4           | Prometheus → native cloud, SQL-compatible reporting layers   |
| 11 | FinOps & Cost Control Complexity | Complexity of managing cloud spend through tagging, budgeting, forecasting, and optimization  | 2           | Tagging, real-time monitoring, hybrid use benefits   |
| 12 | IT Training                      | Complexity of onboarding users and driving adoption of new systems and processes  | 1           | Requires new reporting workflows and data literacy   |
| 13 | IT User Adoption Effort          | Complexity of gaining buy-in from IT users and support teams to adopt and sustain new technologies, platforms, and processes as part of modernization efforts | 3           | Requires dedicated funding and time to build advanced technical skills within OSPI's IT teams to support new platforms and practices |
| 14 | DevOps & Automation Maturity     | Complexity of enabling CI/CD pipelines, Infrastructure as Code (IaC), automated testing, rollback mechanisms, and observability practices                     | 5           | SDLC-lite, FinOps practices, but not fully mature pipelines  |

**Score = Degree of complexity: 51**

## 1.6 Summary Data

| System                    | Complexity Level | All Dimensions Score |
|---------------------------|------------------|----------------------|
| <b>Assessment</b>         | High Complexity  | 36                   |
| <b>eCert</b>              | High Complexity  | 39                   |
| <b>Educator Reporting</b> | High Complexity  | 49                   |
| <b>Report Card</b>        | High Complexity  | 51                   |

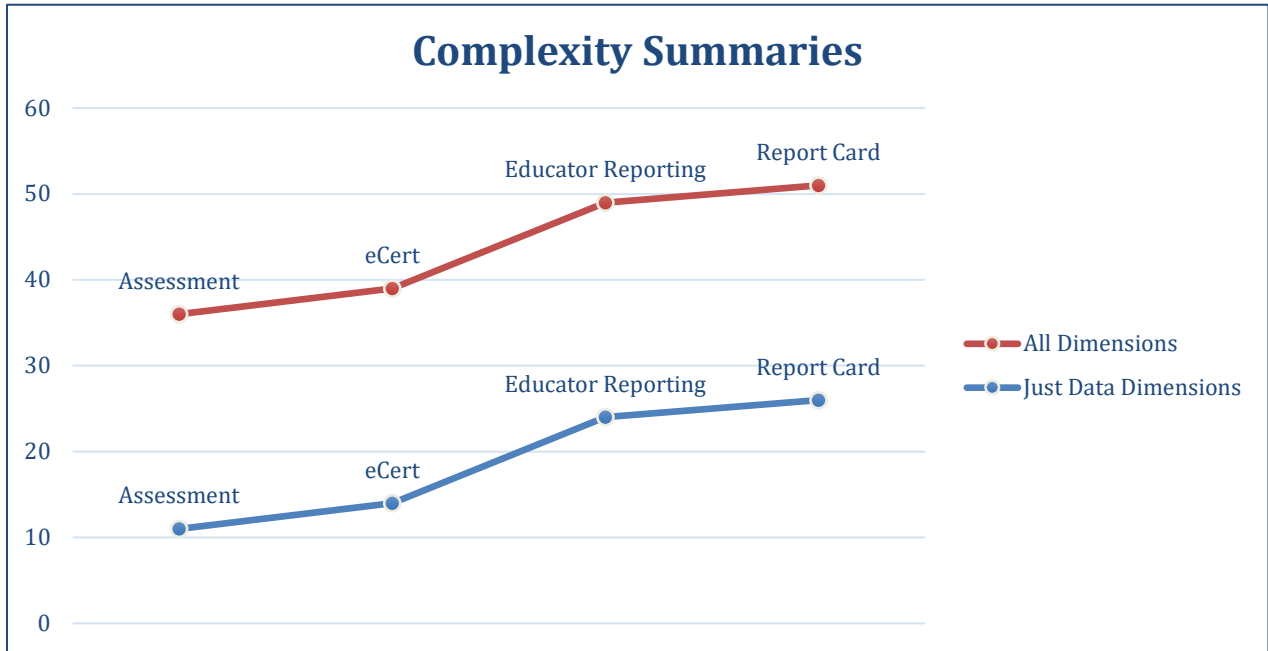
| System                    | Complexity Level | Data Dimensions Score |
|---------------------------|------------------|-----------------------|
| <b>Assessment</b>         | High Complexity  | 11                    |
| <b>eCert</b>              | High Complexity  | 14                    |
| <b>Educator Reporting</b> | High Complexity  | 24                    |
| <b>Report Card</b>        | High Complexity  | 26                    |

## 1.7 Complexity Heatmap

| Dimension                        | Assessment | eCert | Educator Reporting | Report Card |
|----------------------------------|------------|-------|--------------------|-------------|
| Legacy System Entanglement       | 1          | 1     | 5                  | 5           |
| Internal Talent Readiness        | 2          | 2     | 2                  | 2           |
| Manual Process Automation        | 1          | 1     | 4                  | 5           |
| Process Modernization Impact     | 2          | 2     | 4                  | 4           |
| Data Architecture Modernization  | 4          | 4     | 4                  | 4           |
| Data Governance Complexity       | 2          | 3     | 3                  | 4           |
| Cross-Domain Integration         | 1          | 3     | 4                  | 4           |
| Security & Compliance Scope      | 3          | 3     | 3                  | 3           |
| Infrastructure Transformation    | 5          | 5     | 5                  | 5           |
| Platform & Tooling Shift         | 4          | 4     | 4                  | 4           |
| FinOps & Cost Control Complexity | 2          | 2     | 2                  | 2           |
| IT Training                      | 1          | 1     | 1                  | 1           |
| IT User Adoption Effort          | 3          | 3     | 3                  | 3           |

| Dimension                    | Assessment | eCert     | Educator Reporting | Report Card |
|------------------------------|------------|-----------|--------------------|-------------|
| DevOps & Automation Maturity | 5          | 5         | 5                  | 5           |
| <b>Total Scores</b>          | <b>36</b>  | <b>39</b> | <b>49</b>          | <b>51</b>   |

## 1.8 Complexity Summaries



# **PART 5: OSPI DATA IMPLEMENTATION ROADMAP**

## **1.0 Executive Summary**

This strategic plan outlines the transformation of the Office of Superintendent of Public Instruction's (OSPI) data infrastructure from a legacy, on-premise ecosystem of 138+ servers and 200+ databases to a modern, cloud-based medallion architecture. This initiative will deliver eight authoritative data products, each supporting critical business applications for Washington State's K-12 education system. Far from a simple migration, this transformation reimagines OSPI's data management, processing, and delivery to provide timely, accurate, and actionable insights for proactive student outcomes. The accompanying roadmap illustrates this strategy across phases, with dependencies, product team sequencing, and critical milestones.

## 2.0 Current State: The Imperative for Change

Before charting the path forward, it is essential to understand the current data-related challenges OSPI faces. The present environment reflects decades of piecemeal growth necessitated by the challenge of balancing new demands from both legislature and OSPI's customers. OSPI's effort to incrementally improve its environment with constrained budget and staffing resources has allowed it to meet these demands, at the expense of inefficiencies, and redundancies in its data environment.

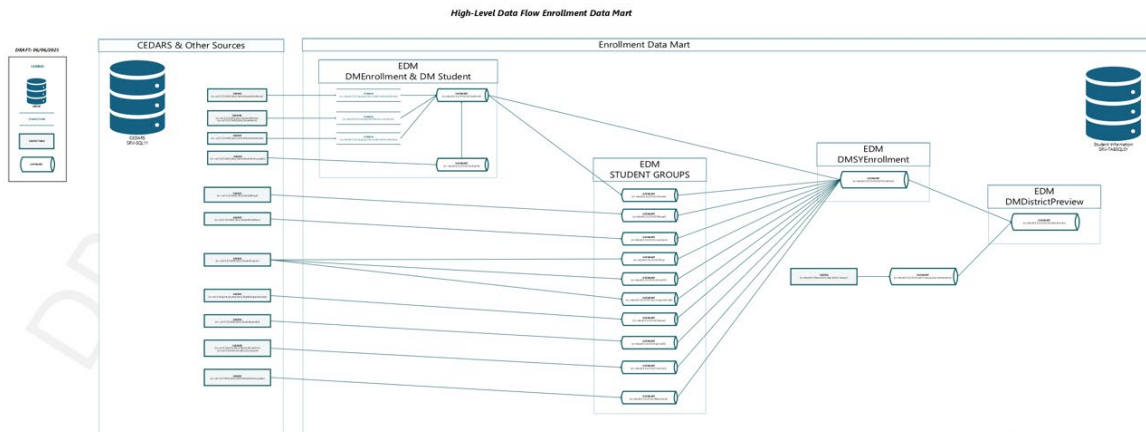
### 2.1 Challenges

OSPI's current data infrastructure is characterized by inefficiencies and risks:

- **Fragmented Infrastructure:** 138+ servers with inconsistent naming conventions and unclear purposes.

**Data Redundancy:** Over 200 databases with conflicting data, including seven distinct enrollment definitions. To overcome the lack of analytics-ready datasets that can be pulled directly into analytical visualizations, data marts are created by individual analysts or data teams in varying layers without an overall governance strategy.

- **Processing Delays:** Data updates take 48–72 hours due to interim manual processes, hindering timely decision-making.
- **Manual Processes:** Reconciliation across systems relies on labor-intensive manual efforts.
- **Siloed Teams:** Fragmented teams maintain isolated systems, perpetuating inefficiencies.
- **Technical Debt:** Over 20 years of accumulated legacy systems create operational drag.
- **Governance Gaps:** No automated data lineage, unclear ownership, and manual oversight increase compliance risks.



## Current State Architectural Problems

Beneath the surface, the architectural design compounds these challenges. The existing structure shows fundamental flaws that make the system fragile, slow, and inefficient to maintain, while limiting OSPI's ability to respond to evolving educational needs.

The above image of EDM shows some foundational issues:

### 1. Point-to-Point Integration Pattern

The diagram shows direct connections from multiple source systems to staging tables and then to data marts. This creates a brittle, spaghetti architecture where:

- Each new data flow requires custom "data movement" code
- Changes in source systems ripple through multiple downstream processes
- No standardized lineage pattern

### 2. Missing Raw/Bronze Layer

The architecture jumps directly from sources to staging tables, lacking a data quality and governance layer. In medallion architecture, the process between the Raw and bronze layer would:

- Store exact copies of source data (immutable)
- Maintain full history and lineage
- Enable data quality and reprocessing from original state
- Support audit and compliance requirements

### 3. Unclear Data Quality Boundaries

The current staging tables appear to perform transformations, but there's no clear separation between:

- Raw data ingestion (bronze)
- Cleansing and standardization (silver)
- Business logic and aggregations (gold)

### 4. Multiple Overlapping Data Marts

The diagram shows multiple EDM datamarts (DMEnrollment, DMSYEnrollment, DMStudent, etc.) that likely contain overlapping data, creating:

- Data redundancy
- Inconsistent business logic
- Confusion about source of truth
- Maintenance overhead
- Onboarding and training difficulties for new analysts

### 5. Governance Gaps:

- Missing centralized metadata management, access control, and data discovery
- Lacking enterprise-wide data lineage, classification, and compliance tracking
- No systematic approach to data validation between layers

## 2.2 Future State: A Modern Data Ecosystem

To overcome these challenges, OSPI must envision a fundamentally different future—one that moves beyond simple migration to true data modernization. The future state represents not only new technology, but a cultural shift toward data as a trusted and shared asset that drives student success and operational transparency. This vision will be delivered incrementally through the accompanying roadmap.

| <b>CURRENT STATE: Daily Reality</b><br>139+ Servers   400+ Databases   Unknown Dependencies | <b>FUTURE STATE: Data Excellence</b><br>1 Platform   Unified Governance   Complete Auditability |
|---|---|
| <b>The Data Discovery Problem</b><br><br><b>Finding Data:</b>                               | <b>Instant Data Discovery</b><br><br><b>AI-Powered Catalog:</b>                                 |

| <p>Analysts spend 60% of their time searching through 400+ databases to find the right table. "Is October enrollment in DMSYEnrollment or DMErollment or DistrictStudentEnrollment?"</p> <p><b>Trust Issues:</b><br/>Same metric calculated differently in 10 places. Federal report shows 50,000 students, state report shows 48,500. Which is right? Nobody knows.</p> <p><b>Tribal Knowledge:</b><br/>Only certain veterans know that srv-sql11 has the "real" enrollment. When they retire, that knowledge is gone forever.</p>   | <p>Type "enrollment" → See "gold.student_enrollment (Official October count)". Every table named, documented, and classified automatically with business-friendly terms.</p> <p><b>Single Source of Truth:</b><br/>One enrollment number everywhere. Federal, state, and local reports automatically reconcile because they use the same validated source.</p> <p><b>Self-Service Discovery:</b><br/>Business users find data themselves. No more waiting for IT to explain where data lives or what it means.</p> |                           |                  |               |   |                          |                           |                  |               |
|---|--|---------------------------|------------------|---------------|---|--------------------------|---------------------------|------------------|---------------|
| <p><b>The Manual Labor Reality</b></p> <p><b>Creating One Report (Current Process)</b></p> <p>Day 1-2: Find the right tables<br/>Day 3-4: Write complex SQL joins<br/>Day 5: Wait for query (45+ minutes)<br/>Day 6-7: Validate numbers manually<br/>Day 8-9: Find &amp; fix discrepancies<br/>Day 10: Deliver report (hope it's right)</p> <table border="1" data-bbox="204 1411 824 1650"> <tr> <th>Est. Reports per Analyst</th> <th>Potential Data Error Rate</th> </tr> <tr> <td style="text-align: center;"><b>2-3/month</b></td> <td style="text-align: center;"><b>15-20%</b></td> </tr> </table> | Est. Reports per Analyst   | Potential Data Error Rate | <b>2-3/month</b> | <b>15-20%</b> | <p><b>Automated Consistency</b></p> <p><b>Creating Same Report (Future Process)</b></p> <p>Minute 1: Type "enrollment by district"<br/>Minute 2: AI generates query<br/>Minute 3-5: Results appear<br/>Minute 6: Share interactive dashboard<br/>Done - Full confidence in numbers</p> <table border="1" data-bbox="849 1411 1419 1650"> <tr> <th>Est. Reports per Analyst</th> <th>Potential Data Error Rate</th> </tr> <tr> <td style="text-align: center;"><b>20+/month</b></td> <td style="text-align: center;"><b>&lt;2%</b></td> </tr> </table> | Est. Reports per Analyst | Potential Data Error Rate | <b>20+/month</b> | <b>&lt;2%</b> |
| Est. Reports per Analyst  | Potential Data Error Rate  |                           |                  |               |   |                          |                           |                  |               |
| <b>2-3/month</b>  | <b>15-20%</b>  |                           |                  |               |   |                          |                           |                  |               |
| Est. Reports per Analyst  | Potential Data Error Rate  |                           |                  |               |   |                          |                           |                  |               |
| <b>20+/month</b>  | <b>&lt;2%</b>  |                           |                  |               |   |                          |                           |                  |               |
| <p><b>The Compliance &amp; Audit Challenge</b></p> <p><b>Black Box Calculations:</b> Critical formulas hidden in 500-line stored procedures with</p>  | <p><b>Complete Audit Trail &amp; Reproducibility</b></p> <p><b>Every Number Explains Itself:</b> Click any value to see its complete lineage - source</p>  |                           |                  |               |   |                          |                           |                  |               |

|  |  |
|--|--|
| <p>"magic numbers" (<math>\times 0.4237</math>) that nobody understands. "Why 0.4237?" The person who knew retired in 2015.</p> <p><b>Audit Scenario:</b><br/><i>Auditor Requests Title I Calculation</i></p> <p>Day 1-2: Search for the stored procedure<br/> Day 3: Find 3 versions, which is correct?<br/> Day 4-5: Trace through 5 other procedures<br/> Day 6: Find hardcoded 0.4237, no documentation<br/> Week 2: Manually recreate in Excel<br/> Week 3: Numbers don't match, start over<br/> Week 4: Provide "best guess" to auditor</p> <p><b>Limited Version History:</b> "What formula did we use last year that everyone agreed was more correct?" Unsure. Ask several people and hope for the best.</p> <p><b>Manual Documentation:</b> If someone remembers to document changes, but if they don't remember then there is no documentation. Documentation is in random Excel files or Word docs that can be easily forgotten.</p> | <p>data, transformations, business rules, formulas, parameters, and approval chain. No more black boxes.</p> <p><b>Audit Scenario:</b><br/><i>Auditor Requests Title I Calculation</i></p> <ul style="list-style-type: none"> <li>• Query Databricks: "How did you calculate Title I for Seattle?"</li> <li>• Amount: \$2,547,823</li> <li>• Source Data: 3,421 students from gold.enrollment_october</li> <li>• Poverty Count: 1,823 students (53.3%)</li> <li>• Formula: <math>1,823 \times \\$1,500 \times 0.4237</math></li> <li>• Parameters: \$1,500 from Federal Register Vol.89</li> <li>• Adjustment: 0.4237 from DOE Table 3.2</li> <li>• Version: 2.1 (Oct 1, 2024)</li> <li>• Approved By: CFO_Office</li> <li>• Quality Checks: ✓ All passed</li> <li>• Immutable ID: 7f3a9c2d-8b1e-4d6f</li> </ul> <p><b>Instant Reproducibility:</b> Auditor says "What if we used September data with last year's rate?"<br/> System instantly recalculates: Original: \$2,547,823 → Recalculated: \$2,341,456 → Shows exact differences. Any historical calculation can be reproduced with different parameters in seconds.</p> <p><b>Automatic Metadata Capture:</b> System records everything automatically - WHO ran it, WHEN it ran, WHAT version of code, WHICH parameters, WHAT data quality checks passed, WHO approved. No manual documentation needed.</p> |
|--|--|

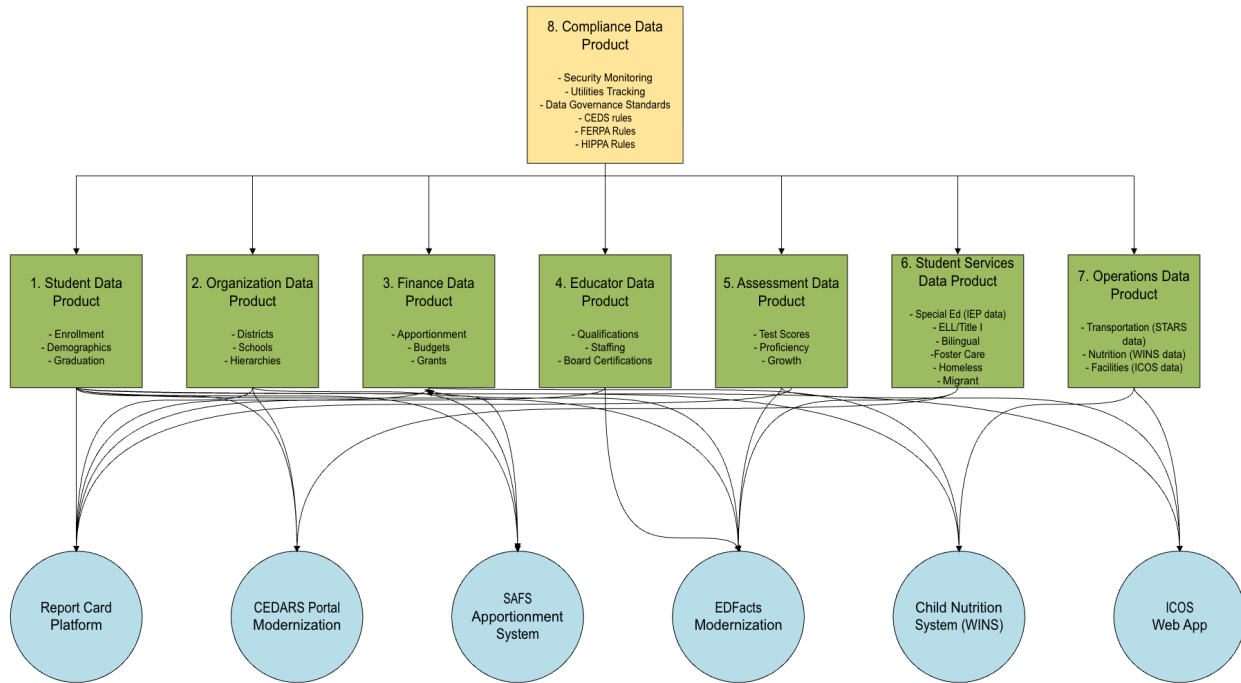
|  |  |
|--|--|
|  | <p><b>Time Travel Built In:</b> "Show me this calculation from October 2023" - instantly retrieves historical formulas, data, and results. Compare versions side-by-side. See exactly what changed and why.</p>  |
|  | <p><b>Security &amp; Compliance by Design</b></p> <p><b>PII Auto-Protection:</b> SSNs automatically detected, classified, and masked. One governance policy applies everywhere. Change once, protect everything.</p> <p><b>Complete Access Logs:</b> Every data access tracked - who, what, when, why. Instant compliance reports for privacy audits. Prove FERPA compliance in minutes.</p> |
| <p><b>The Fundamental Transformation</b></p>   |  |
| <p><b>Current: Data as a Setback</b></p> <ul style="list-style-type: none"> <li>✘ Every query is an investigation</li> <li>✘ Every calculation is a mystery</li> <li>✘ Every audit is a crisis</li> <li>✘ Every formula change loses history</li> <li>✘ Every report's accuracy is questionable</li> <li>✘ Every new analyst needs tribal knowledge</li> </ul> | <p><b>Future: Data as a Service</b></p> <ul style="list-style-type: none"> <li>★ Every query returns trusted data</li> <li>★ Every calculation is transparent</li> <li>★ Every audit is instant</li> <li>★ Every change is versioned</li> <li>★ Every report is validated</li> <li>★ Every user can self-serve at their RBAC level</li> </ul>  |

**Vision**

The transformed data platform will feature:

- Eight Data Products: Authoritative, single sources of truth for key educational domains.
- Medallion Architecture: Structured data layers (Raw → Bronze → Silver → Gold) for scalability and quality.
- Real-Time Processing: Data updates in 15 minutes, replacing 48-hour delays.
- Unified Identities: Consistent identifiers across all systems for students, educators, and organizations.

- Product-Oriented Teams: Domain-focused teams with end-to-end ownership.
- Consolidation: Many:1 reduction from 400+ databases to streamlined dataset development.
- Automated Governance: Full lineage tracking and compliance embedded in the platform.



## Transformation Approach

Achieving this vision requires a deliberate approach. Rather than replicating today’s inefficiencies in the cloud, OSPI will redesign its processes and architecture to deliver greater speed, reliability, and value with each step of the data modernization journey. This approach deliberately avoids replicating inefficiencies in the cloud – a risk common in many enterprise data migrations.

This is not a “lift-and-shift” migration but a deliberate redesign that:

1. Consolidates Systems: Merges redundant databases into unified data products.
2. Standardizes Metrics: Establishes a single definition for critical metrics (e.g., enrollment, graduation).
3. Modernizes Architecture: Adopts a medallion architecture for scalability and efficiency.
4. Enables Self-Service: Provides APIs and dashboards for direct stakeholder access.
5. Automates Workflows: Replaces manual processes with intelligent data pipelines.
6. Ensures Compliance: Embeds federal reporting and governance requirements.
7. Enforces Governance: Implements automated policies to eliminate ambiguity.

## 2.3 Data Governance: From Documented to Enforcement

Governance is where data transformation succeeds or fails. Today, rules exist on paper but lack consistent enforcement. Tomorrow, governance must be embedded and automated, to ensure clarity and compliance. Most of all, enforced governance paves the way for greater transparency and trust in the dataset for OSPI and the administrators, educators, and families invested in its uses.

### Current Governance Challenges

- No Data Lineage: Inability to trace data from source to report.
- Unclear Ownership: Lack of defined accountability for data assets.
- Manual Compliance: Reliance on individual expertise for regulatory adherence.
- Audit Challenges: Difficulty proving compliance or tracking changes.
- Ambiguity: Rules are documented but varied enforcement leads to different interpretations and shadow IT (e.g., ungoverned Excel files).

### Future Governance Framework

The cloud platform embeds automated governance:

- Unity Catalog: Tracks data lineage from source to consumption.
- Role-Based Access Control (RBAC): Automatically enforces permissions.
- Data Classification: Tags PII, sensitive, and public data systematically.
- Audit Logging: Records all access, changes, and queries.
- Quality Gates: Validates data at each layer before progression.
- Retention Automation: Enforces data expiration and archiving policies.
- Gold Standard: Naming conventions make it clear which datasets can be used.

### Zero Gray Areas Policy

Core Principle: Every data element and report must have a definitive classification, with no exceptions. Enforcement begins in Release 0 – and is progressively automated through the Compliance Data Product Team.

### Implementation Rules:

1. Gold Standard Integrity:
  - a. Gold-layer data is the certified, authoritative truth.
  - b. Any post-Gold manipulation requires reclassification.

- c. No exceptions are permitted, regardless of scope.
2. Naming Convention Enforcement:
  - a. gold.prod.student\_enrollment → Certified truth
  - b. silver.prod.student\_enrollment\_modified → Modified from Gold
  - c. bronze.dev.enrollment\_analysis → Exploratory/unofficial
3. Non-Gold Report Documentation:
  - a. Non-Gold reports must include a cover page detailing:
    - i. Modified Gold data
    - ii. Specific changes made
    - iii. Business justification
    - iv. Owner accountable
    - v. Expiration date
4. Examples of Enforcement:
  - a. Scenario: A district excludes certain students from enrollment counts.
    - i. Solution: Create silver.custom.enrollment\_excluding\_x with documentation.
    - ii. Not Allowed: Modifying Gold data in tools like Power BI without reclassification.
  - b. Scenario: Federal reports require different calculations than state reports.
    - i. Solution: Create silver.federal.enrollment\_federal\_calc with clear naming.
    - ii. Not Allowed: Using identical report names for different calculations.

## **Governance Benefits**

- Transparency: Full visibility into data origin, changes, and usage.
- Simplified Compliance: Automated FERPA checks and federal reporting.
- Reduced Manual Work: Eliminates spreadsheet reconciliation and manual monitoring.
- Traceability: Complete audit trails and impact analysis for all changes.

## **2.4 Prerequisites for Success**

No transformation of this scale succeeds without laying strong foundations. Technology, contracts, people, and readiness must all be in place before OSPI can fully realize the benefits of a modern data ecosystem. By redesigning the operating model, OSPI will ensure that technology investments translate into better student outcomes.

## **Infrastructure and Contracts**

- Azure cloud subscription configured
- Databricks platform licensed and deployed
- Microsoft Purview for data discovery
- Unity Catalog for governance
- Azure DevOps for CI/CD pipelines
- Power BI/Tableau licenses for visualization

## **Organizational Readiness**

- Executive sponsorship secured
- Product teams formed and trained
- Data governance committee established
- Change management program activated
- Stakeholder communication plan in place
- Budget allocated for a three-year transformation
- Zero Gray Areas policy approved
- Data stewards assigned to each domain

## **Technical Foundation**

- Network connectivity between on-premise and cloud systems
- Security frameworks and policies defined
- Identity and access management configured
- Backup and disaster recovery plans established
- Development, test, and production environments ready
- Governance automation tools configured

## **2.5 Product Operating Model**

Technology alone will not deliver change. OSPI's teams must also transform—shifting from siloed functions to product-oriented teams with clear ownership, accountability, and a direct line to the outcomes that matter for Washington's students.

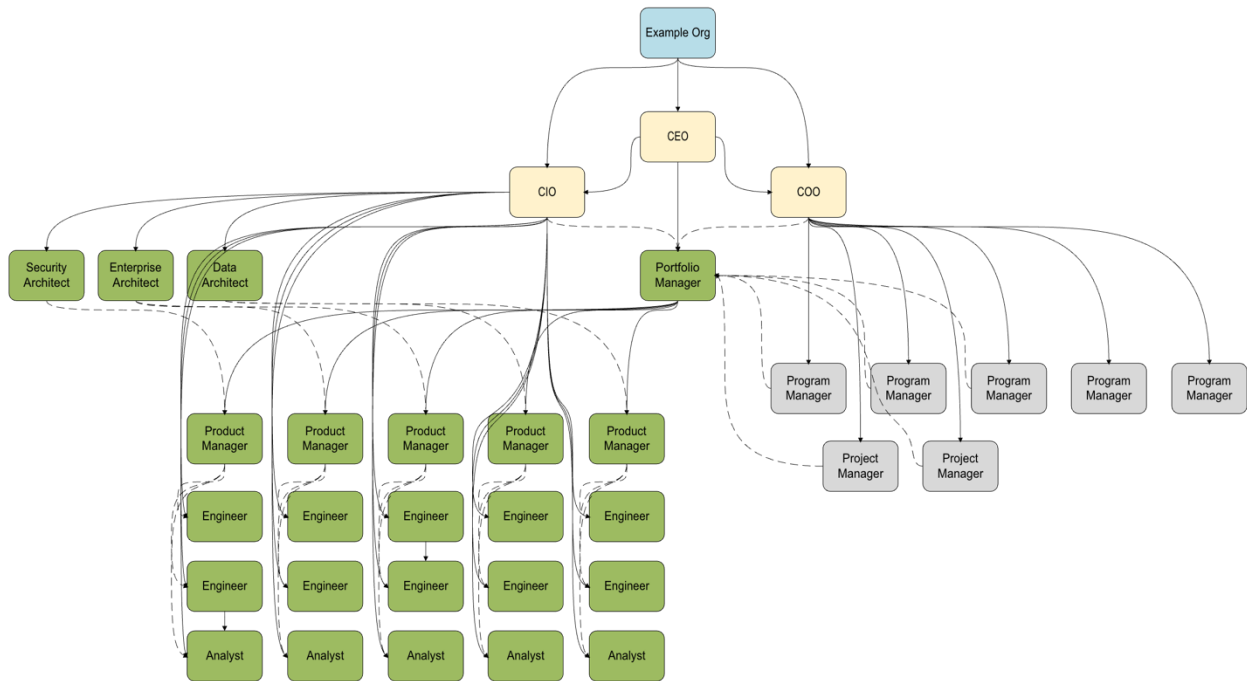
### **Why Product Teams?**

Shifting to product-oriented teams ensures:

1. Domain Expertise: Teams specialize in specific educational categories.
2. End-to-End Ownership: Teams manage data from source to stakeholder.
3. Faster Innovation: Fewer dependencies accelerate delivery.
4. Business Alignment: Direct focus on stakeholder outcomes.
5. Continuous Improvement: Iterative enhancements based on feedback.
6. Clear Accountability: Teams own their data governance based on universal standards (Compliance Data Product).
7. Standardization: The Compliance Data Product provides agency-wide consistent governance that can be pulled in with simple pipelines.

## Team Structure

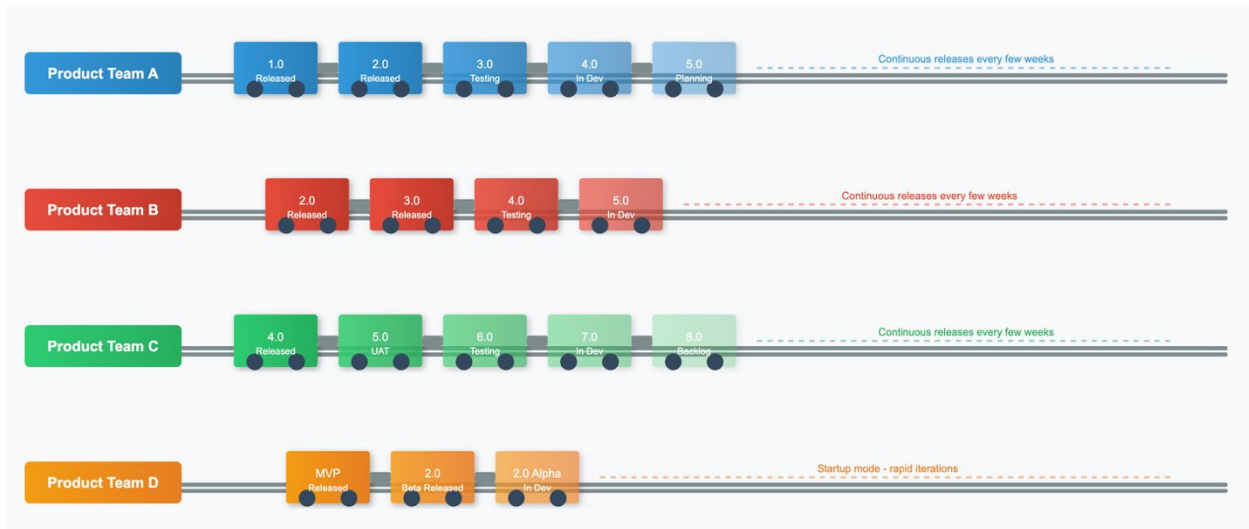
- Eight Permanent Product Teams: Student, Finance, Educator, Assessment, Student Services, Operations, Compliance, Organization. (Organization Data Product can be combined with IT Operations as per the current OSPI team structure, however, the product team must still own the product roadmap.)
- Roles: Product Manager, Tech Lead / Architects, Data Engineers, Developers, Analysts, Data Steward / Data Owners / Product Owners.
- Temporary Project Teams: Formed for specific initiatives with defined timelines.



## Product Teams Key Principles

- Each team owns their release cadence

- Independent backlogs & roadmaps (combined into a unified portfolio)
- Continuous delivery to production
- Shared platform, unique products
- Teams manage their entire product from conception → development → testing → production → break-fix → continuous improvements



To ensure a successful transition from an operations-focused to a product-oriented organization, we recommend OSPI to conduct a comprehensive organizational design exercise. This process will map current team members’ skills and roles to the new product-oriented structure, identifying needs for upskilling, role adjustments, or additional resources. Supports such as tailored training, organization readiness programs, and robust change management strategies will facilitate a smooth transition.

The product-oriented model enhances migration success by fostering domain expertise and clear accountability, enabling teams to deliver cohesive data products efficiently. For ongoing work, this structure drives sustained success by aligning teams across siloed departments into singular teams directly responsible for the products they build, reducing silos, and enabling faster innovation, ensuring OSPI remains responsive and effective in meeting future educational demands.

## 2.6 Implementation Roadmap

The roadmap translates strategy into action. It illustrates how teams, releases, and products work together to deliver incremental value, while maintaining flexibility to adapt to evolving needs. The roadmap also balances urgency with flexibility, and sequences work in ways that align to legislative cycles and perceived stakeholder capacity.

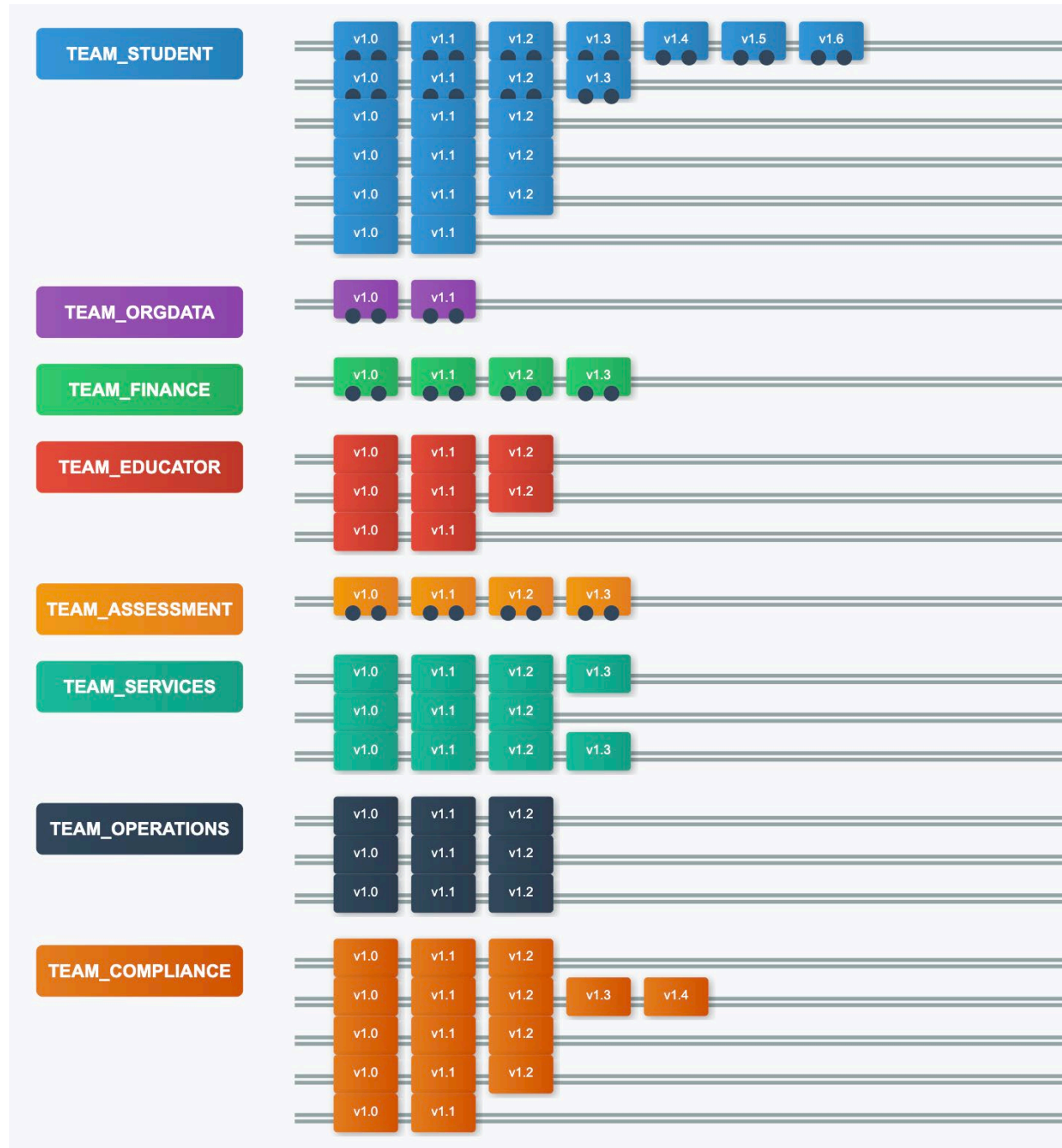
## Data Product Teams – Independent Release Trains

This design is not much different than how OSPI is currently structured, it is attempting to refine and give more structure to OSPI's current organizational picture.

- 7-8 independent teams, 1 team per data product
  - Team – Student continuously work on student data
    - Controls all student projects and releases end-to-end
    - Completely owns the data product roadmap from concept to build to break-fix
    - Has all roles needed for end-to-end success
  - Team – Compliance continuously work on compliance data (internal and external compliance)
    - The compliance automation hub for every other data product
      - Builds all compliance automation templates
        - Data governance
        - Security governance
        - Legislative compliance
      - Other data products ingest and apply standards from the Compliance Data Product
    - Controls all compliance projects and all releases end-to-end
    - Completely owns the data product roadmap
    - Has all roles needed for end-to-end success
  - Team – OrgData continuously work on organization and directory data (may also be an extension of the IT org but with a specific focus)
    - Controls all organization and directory projects and releases end-to-end
    - Completely owns the data product roadmap
    - Has all roles needed for end-to-end success
  - Team – Finance continuously work on financial data (both internal and external finance data)
    - Controls all financial projects for internal consumption and external systems
    - Manages all releases end-to-end
    - Completely owns the data product roadmap
    - Has all roles needed for end-to-end success

- Team – Educator continuously work on educator data
  - Controls all educator projects and all releases end-to-end
  - Completely owns the data product roadmap
  - Has all roles needed for end-to-end success
- Team – Assessment continuously work on assessment data
  - Controls all assessment projects and all releases end-to-end
  - Completely owns the data product roadmap
  - Has all roles needed for end-to-end success
- Team – Services continuously work on student services data
  - Incorporates all student services into one data product (e.g. special education, bilingual)
  - Controls all student services projects and all releases end-to-end
  - Completely owns the data product roadmap
  - Has all roles needed for end-to-end success
- Team – Operations continuously work on school operations data
  - Controls all operations projects and all releases end-to-end
  - Completely owns the data product roadmap
  - Has all roles needed for end-to-end success
- Current constraints
  - Programs operating in siloes for each mandated area (e.g. PE is a separate program with a one-person team)
  - Operating siloes create data siloes
  - Too much free range and shadow IT – teams feel they must come up with their own solutions for their own problems
- Future resolutions
  - Consider unifying program data and align program teams into the core 8 data product teams
  - Automate governance so that even teams operating in siloes know what data to use and what data not to use
  - With better automation comes better self-service capabilities that could reduce shadow IT (e.g. teams using their own instance of Fabric, teams creating their own reporting workarounds for data quality)

Below is an image depicting how each team is a standalone unit with all roles needed for success of the data product that they are responsible for. They own their roadmap for their product. Each line of trains is a different project, each sequential train is a different release within the same project.



### RELEASE SUMMARY BY DATA PRODUCT TEAM

Below is a summary of the releases that will be depicted in the accompanying excel roadmap. Products are meant to continue into perpetuity. The projects and releases included in the

summary are only those that are required to complete the entire migration. However, that does not mean that these products cease to exist after these releases are done. The design of these data product teams is meant to continue releasing new and better features on a rolling and regular basis.

|   |   |
|---|---|
| <p style="text-align: center;"><b>TEAM_STUDENT</b></p> <p><i>Manages the Student Data Product</i></p> <p>6 Projects with 22 Total Releases</p> <ul style="list-style-type: none"> <li>• Project 1: 7 releases</li> <li>• Project 2: 4 releases</li> <li>• Project 3: 3 releases</li> <li>• Project 4: 3 releases</li> <li>• Project 5: 3 releases</li> <li>• Project 6: 2 releases</li> </ul> | <p style="text-align: center;"><b>TEAM_ORGDATA (or IT Ops)</b></p> <p><i>Manages the Organization Data Product</i></p> <p>1 Project with 2 Total Releases</p> <ul style="list-style-type: none"> <li>• Project 1: 2 releases</li> </ul>   |
| <p style="text-align: center;"><b>TEAM_FINANCE</b></p> <p><i>Manages the Finance Data Product</i></p> <p><i>(Assumption: SAFS will be purchasing a new SaaS system to handle apportionment, the work here is only the data backend)</i></p> <p>1 Project with 4 Total Releases</p> <ul style="list-style-type: none"> <li>• Project 1: 4 releases</li> </ul>                                  | <p style="text-align: center;"><b>TEAM_EDUCATOR</b></p> <p><i>Manages the Educator Data Product</i></p> <p>3 Projects with 8 Total Releases</p> <ul style="list-style-type: none"> <li>• Project 1: 3 releases</li> <li>• Project 2: 3 releases</li> <li>• Project 3: 2 releases</li> </ul>   |
| <p style="text-align: center;"><b>TEAM_ASSESSMENT</b></p> <p><i>Manages the Assessment Data Product</i></p> <p>1 Projects with 4 Total Releases</p> <ul style="list-style-type: none"> <li>• Project 1: 4 releases</li> </ul>   | <p style="text-align: center;"><b>TEAM_SERVICES</b></p> <p><i>Manages the Services Data Product</i></p> <p>3 Projects with 11 Total Releases</p> <ul style="list-style-type: none"> <li>• Project 1: 4 releases</li> <li>• Project 2: 3 releases</li> <li>• Project 3: 4 releases</li> </ul>  |
| <p style="text-align: center;"><b>TEAM_OPERATIONS</b></p> <p><i>Manages the Operations Data Product</i></p> <p>3 Projects with 9 Total Releases</p> <ul style="list-style-type: none"> <li>• Project 1: 3 releases</li> <li>• Project 2: 3 releases</li> <li>• Project 3: 3 releases</li> </ul>   | <p style="text-align: center;"><b>TEAM_COMPLIANCE</b></p> <p><i>Manages the Compliance Data Product</i></p> <p><b>This is the most important team to ensure proper structure and efficiency.</b></p> <p>5 Projects with 16 Total Releases</p> <ul style="list-style-type: none"> <li>• Project 1: 3 releases</li> <li>• Project 2: 5 releases</li> <li>• Project 3: 3 releases</li> <li>• Project 4: 3 releases</li> <li>• Project 5: 2 releases</li> </ul> |

## Release 0: Foundation

Every project begins with a critical Release 0 to:

1. Discover Data: Use Microsoft Purview to identify all data sources.
2. Resolve Conflicts: Standardize competing metric definitions. Resolve unnecessary databases or tables.
3. Validate with Stakeholders: Ensure only relevant data is migrated.
4. Design Governance: Define policies, ownership, and quality rules.
5. Plan Transformation: Map Raw → Bronze → Silver → Gold data flows.
6. Prepare Delivery: Set up APIs, dashboards, and user interfaces (e.g., web apps).

## Why Release 0 Matters

- Uncovers hidden dependencies and obsolete data.
- Resolves business rule conflicts before migration. Solidifies single truth.
- Embeds governance from the outset.

## Release Strategy

- Duration: 2–14 weeks per release.
- Approach: Each release delivers standalone value, builds on prior foundations, and includes governance.
- Example Journey (Report Card Project):
  - Release 1 (Enrollment): Establishes student data with governance.
  - Release 2 (Assessment): Adds test scores, reuses enrollment data.
  - Release 3 (Finance): Incorporates budget data, leverages enrollment.
  - Release 4 (Educators): Adds staffing data, uses enrollment for ratios.
  - Release 5 (Graduation): Tracks outcomes, integrates all prior data.
  - Outcome: A fully governed Report Card with cloud automation and less infrastructure.

## Program Management Office

These activities establish the functional governance team needed to manage this project effectively.

- Ideal PMO structure:
  - **Executive Sponsor / Steering Committee**
    - Provides strategic alignment with organizational goals
    - Approves funding, priorities, and escalated decisions
  - **PMO Director / Head of PMO**
    - Leads the PMO and sets governance standards
    - Ensures alignment with enterprise strategy
  - **Portfolio Management Team**
    - Oversees portfolio prioritization and investment decisions
    - Monitors portfolio-level risks, benefits, and resource allocation
  - **Program Managers**
    - Manage groups of related projects (programs)
    - Ensure projects deliver collective business outcomes
  - **Project Managers**
    - Deliver individual projects on time, within scope and budget
    - Report status, risks, and issues to the PMO
  - **Project Coordinators / Analysts**
    - Support reporting, scheduling, and documentation
    - Manage tools and dashboards for visibility
  - **Governance & Compliance Function**
    - Defines project standards, methodologies, and templates
    - Ensures compliance with frameworks (PMI, PRINCE2, Agile, etc.)
  - **Resource Management Function**
    - Oversees staffing, skill development, and capacity planning
    - Coordinates training and certifications
  - **Finance & Benefits Management**
    - Tracks budgets, costs, and return on investment
    - Measures benefits realization post-project
  - **Change Management & Communications**
    - Ensures stakeholder engagement and adoption
    - Provides consistent messaging and reporting
  - **Tools & Knowledge Management**

- Manages project management tools, dashboards, and repositories
- Captures lessons learned and best practices
- Wa-Tech approved governance documents: governance plan, comms plan, project management
- Establish executive stakeholder team – drive decision-making body, assists with deconfliction, fixes blockers
- Establishing a steering committee that is representative of the various functional & technical domains within OSPI and LEAs
- Establishing & managing a live roadmap / master transformation strategy – “where are we in the grander scheme and how does my work relate to what we are doing”
- Workstream & resource management planning – who is managing what work
- Scope definition – for each workstream, what falls into this project vs. fray / nice to have

## Change Management & Strategic Communications

Large-scale transformation demands attention to the organization beyond technology and process modernization. People must understand, support, and ultimately adopt the changes required at an enterprise and workflow level. Deliberate, proactive, and thoughtful change management & communication strategy is mandatory to align internal teams, district partners, and OSPI’s stakeholders to the future-state cloud and data journey. These efforts intend to build capacity and readiness across all affected groups and to minimize resistance and business operations disruption.

### Core Objectives of Change Management & Strategic Communications:

- Building Buy-In: Clearly communicating the “why” behind the change to encourage understanding, support, and alignment between OSPI teams and partner organizations.
- Minimizing Disruption: Helping staff and stakeholders navigate transition with clarity and support needed to succeed in their roles.
- Accelerating Understanding: Equipping stakeholders with the knowledge and confidence to embrace new systems, workflows, and data governance policies
- Sustaining Change: Ensuring long-term behavior change through reinforcements, ownership transfer, and capacity-building for individuals and domain teams.

### Change Management Workstreams:

| Workstream                           | Description  |
|--------------------------------------|--|
| <b>Stakeholder Impact Assessment</b> | Identify groups affected by each release; assess their concerns, risks, and communication needs. |

| <b>Workstream</b>                   | <b>Description</b>  |
|-------------------------------------|---|
| <b>Organizational Readiness</b>     | Gauge change fatigue, communication saturation, and baseline capability to absorb change across departments.  |
| <b>Change Activity Planning</b>     | Sequence change activities alongside release rollouts; align with training, governance rollout, and product team onboarding.                                |
| <b>Communications Strategy</b>      | Deliver consistent messaging that aligns technical transformation with OSPI mission-driven goals (ex. student outcomes, efficiency, audit readiness).       |
| <b>Training &amp; Enablement</b>    | Offer targeted training aligned with vendor content and product rollouts. Consider instructor-led, peer-led, and self-service formats.                      |
| <b>Adoption Metrics</b>             | Track adoption across stakeholders using KPIs such as training completion, product usage, and feedback scores.  |
| <b>Feedback &amp; Reinforcement</b> | Provide feedback loops (e.g., office hours, surveys, release retrospectives) and respond promptly. Transition ownership to product support teams over time. |

### **Strategic Communications Foundations**

To drive engagement and avoid confusion during the multi-year data transformation, communications should follow these pillars:

- **Consistent Voice:** Align messaging across vendor, OSPI, and LEA communication channels and avoid duplication, contradiction, or fragmentation.
- **Audience-Based Tailoring:** Customize content for executive leaders, program managers, analysts, and technical staff.
- **Simple, Visual, and Repeatable:** Use visuals (e.g., before/after, data flow maps), recurring formats (e.g., monthly updates, “What’s Changing” one-pagers), and plain language to drive clarity.
- **Proactive and Predictable:** Communicate changes well in advance, anticipate confusion points, and provide early warning for disruptions, especially to workflow.
- **Integrate with Rollouts:** Communications should be tied directly to product team release plans (ex. Release 1 training → Launch webinar → post-launch office hours → Adoption check-in).

### **Example Communication Products**

- Transformation briefing decks for Executive Sponsors

- Quarterly modernization highlights newsletters
- “Release Ready” emails: what’s coming, who’s impacted, how to prepare
- OSPI Intranet landing pages that include FAQs, training videos, and policy updates
- Feedback and Q&A dashboards for open questions and training content

### **Alignment with Product Model**

Each product team must include dedicated change management and communications support, working in closely in partnership with the project and vendor teams. This ensures that communications are domain-specific, timely, and tailored to the needs of stakeholders impacted by each release.

## **2.7 Critical Success Factors**

Certain conditions must be in place for the roadmap to succeed. These success factors are not optional—they are the non-negotiable guardrails that ensure the work stays on course and delivers value.

- Dynamic Roadmap: Quarterly reviews to adapt to new insights.
- Discovery-Driven: Release 0 uncovers hidden dependencies and quality issues.
- Flexible Prioritization: Adjusts to legislative or federal changes.
- Evolving Governance: Policies refined based on learnings.
- Stakeholder Engagement: Continuous collaboration with users.

## **2.8 Expected Outcomes**

The transformation is not only about systems, it is also about impact to OSPI’s core mission and functional objectives. These outcomes articulate what OSPI, its partners, and Washington’s students will gain through the modernization effort. They demonstrate how modernization can directly support OSPI’s mission of equity, access, and excellence in education.

### **Technical Outcomes**

- 40% reduction in infrastructure (138 servers to cloud PaaS).
- 95% faster processing (48 hours to 15 minutes).
- Many:1 database consolidation (400+ to 100+).
- 100% automated data security.
- Full data lineage tracking.

### **Business Outcomes**

- Real-time insights for decision-making.

- Self-service analytics for users.
- Single source of truth for all metrics.
- Predictive analytics for early intervention.
- Comprehensive audit trails.
- Capacity building for OSPI staff to focus on district support and student-facing initiatives.

## Financial Outcomes

- 60% faster feature delivery (from months/years to days/weeks)
- 80% reduction in unnecessary manual operations costs (hours/weeks spent doing manual work reduced to minutes/days)

## Governance Outcomes

- 100% data lineage visibility.
- 90% reduction in manual reconciliation.
- Automated compliance reporting.
- Real-time data quality monitoring.
- Clear ownership for all data elements.

## 2.9 Risk Mitigation

Large-scale change inevitably brings risk to the business. This effort includes an assertive plan to identify, manage, and reduce those risks while maintaining momentum and time-to-value. Risk mitigation themes include:

- Data Loss: Dual-run periods ensure data integrity.
- Business Disruption: Gradual migration with fallback options.
- User Adoption: Robust training and change management.
- Technical Complexity: Release 0 reduces unknowns.
- Budget Overrun: Modular releases allow flexibility.
- Governance Resistance: Zero Gray Areas policy ensures consistency.

## 2.10 Roadmap as a Guiding Framework

The attached Gantt chart roadmap serves as a strategic template, not an immutable directive. Given the dynamic nature of educational policy, stakeholder needs, and technological advancements, the roadmap must remain adaptable to changes that may emerge months or years into the future. This roadmap anticipates mid-course adjustments in response to legislative priorities, new reporting requirements, and emerging policy changes. Each release

should serve as a checkpoint to reassess the scope of work, validate alignment with current state requirements, and incorporate new insights or priorities. This iterative approach ensures the transformation remains relevant and effective throughout its multi-year journey.

This transformation is a pivotal opportunity to modernize OSPI's data infrastructure and enhance service delivery to Washington State's K-12 community. Success requires:

- Leadership Commitment: Sustained support from executives.
- Stakeholder Collaboration: Active participation from districts and users.
- Patience and Flexibility: Adapting to discoveries during the journey.
- Discipline: Enforcing governance standards without exception.

By executing this roadmap, leveraging Release 0 discoveries, enforcing rigorous governance, and delivering incremental value, OSPI will transition from a fragmented legacy system to a modern, data-driven organization with world-class governance, empowering stakeholders with timely and trustworthy data.

# **PART 6: OSPI PROOF OF CONCEPT (POC) AND MIGRATION READINESS STATEMENT OF WORK**

## **1.0 Background and Efforts to Date**

OSPI has completed all activities defined under the *OSPI Cloud Modernization Feasibility Study – Statement of Work*. This includes completion of discovery activities, validation of cloud adoption feasibility, definition of a migration readiness and integration strategy, and establishment of enterprise governance and financial transparency frameworks. These efforts have resulted in the development of a clearly defined modernization roadmap, decision-making structure, and foundational standards aligned with OSPI’s strategic objectives and Washington State IT governance requirements.

Building upon this completed planning and definition work, OSPI is now initiating a follow-on engagement to begin implementation activities. The selected vendor shall support OSPI in operationalizing the defined modernization roadmap, implementing governance and automation frameworks, and establishing the technical and organizational capabilities required for sustained adoption. The objective of this engagement is to deliver a secure, automated, and compliant cloud environment that enables improved data accessibility, operational efficiency, and data-driven decision-making across the agency.

## **2.0 Vision**

The vision for OSPI’s cloud modernization initiative is to establish a cohesive, secure, and scalable platform that is modern, inclusive, and data-driven, enabling equitable access to reliable information across all educational programs. The future state environment will support seamless collaboration among educators, administrators, and stakeholders by providing trusted, well-governed data and automated processes that improve decision-making and operational efficiency. Through this modernization, OSPI aims to create a unified foundation that enhances transparency, supports continuous improvement, and prioritizes positive educational outcomes for all students.

## 3.0 Purpose of the Implementation Planning Study

OSPI seeks a qualified vendor to lead and support the initial implementation phase of the data modernization initiative. Building upon the definitions and frameworks established through the *OSPI Cloud Modernization Feasibility Study – Statement of Work*, the selected vendor will collaborate with OSPI and its partners to operationalize the defined modernization roadmap. This includes standing up the foundational cloud environment, implementing governance and automation structures, and preparing for a phased rollout of modernized data and application services. The purpose of this engagement is to translate defined strategies into actionable implementation steps that enable OSPI to achieve a secure, compliant, and sustainable cloud operating model supporting statewide education systems.

## 4.0 Scope of Services

### 4.1 Implementation of Strategy Development, Governance and Team Structuring

The vendor shall collaborate with OSPI to translate defined modernization strategies into an executable governance and delivery framework that supports statewide implementation.

Activities include:

- Develop and document the Enterprise Architecture Charter, defining decision rights, design authority, and architectural review cadence.
- Define and implement the Software Development Lifecycle (SDLC) process aligned to OSPI and WaTech standards.
- Establish Change Advisory Board (CAB) and Change Control procedures.
- Define FinOps dashboards, tagging standards, and reporting metrics to monitor cloud cost transparency.
- Recommend scalable governance and delivery structures to support statewide adoption.
- Create the Infrastructure/Platform Tiger Team responsible for implementing technical and policy guardrails.
- Apply initial security and networking guardrails and confirm minimal infrastructure needed for the POC.
- Set up Role-Based Access Control (RBAC) within the cloud environment.
- Documenting the post-implementation governance structure, ensuring alignment with OSPI's enterprise architecture and WaTech oversight frameworks.

## 4.2 Partner Engagement Planning

The vendor shall assist OSPI in designing and implementing statewide Organizational Change Management (OCM) and engagement strategy to ensure readiness and adoption across stakeholders. Activities include:

- Develop and document the Enterprise Architecture Charter, defining decision rights, design authority, and architectural review cadence.
- Define and implement the Software Development Lifecycle (SDLC) process aligned to OSPI and WaTech standards.
- Establish Change Advisory Board (CAB) and Change Control procedures.
- Define FinOps dashboards, tagging standards, and reporting metrics to monitor cloud cost transparency.
- Recommend scalable governance and delivery structures to support statewide adoption.
- Form the Infrastructure/Platform Tiger Team responsible for implementing technical and policy guardrails.
- Apply initial security and networking guardrails and confirm minimal infrastructure needed for the POC.
- Set up Role-Based Access Control (RBAC) within the cloud environment.

## 4.3 System Requirements Validation

The vendor shall validate and refine system, data, and technical requirements to ensure readiness for implementation. Activities include:

- Validate and refine functional and technical requirements to ensure compatibility with existing systems (SIS, CEDARS, etc.).
- Perform data catalog auto-discovery, populating metadata and business glossary terms.
- Define policies for PII detection, classification, and key management.
- Validate encryption standards and identity controls.
- Identify Minimum Viable Product (MVP) features for initial deployment.

## 4.4 Data Migration and System Integration

The vendor shall develop initial plans to support future system integration and data migration activities required for statewide implementation. These activities are preparatory and will inform subsequent implementation phases. Activities include:

- Draft a Data Validation and Migration Plan describing strategy, dependencies, and QA processes.
- Draft System Integration Plans identifying data exchange methods, interfaces, and interoperability requirements.

- Configure data pipelines (RAW → Bronze → Silver → Gold) with automation and monitoring.
- Define and automate Silver-layer quality rules and remediation workflows.
- Develop and test APIs connecting the data layer to analytics environments.
- Conduct User Acceptance Testing (UAT) and reconciliation validation.

## 4.5 Project Management and Reporting

The vendor shall establish the project management framework, tracking mechanisms, and success measurement.

- Conduct budget and resource reviews and manage the procurement of required tools.
- Oversee and validate build progress and completion.
- Track technical debt, risks, and variances.
- Update the roadmap based on lessons learned and performance metrics

## 5.0 Deliverables

The contractor shall provide the following deliverables upon completion of the Proof of Concept (POC) and associated workstreams. These deliverables represent the foundational elements necessary to transition from feasibility definition to operational execution. All deliverables are subject to change based on timeline and technology constraints.

### **Governance Deliverables**

- Enterprise Architecture (EA) Charter
- Software Development Lifecycle (SDLC) Framework
- Change Advisory Board (CAB) Charter and Change Control Procedures
- FinOps Dashboard and Tagging Standards
- Governance and Partner Plan defining statewide roles and decision structures

### **Cloud Workstream**

- Validated POC cloud environment with applied guardrails and RBAC
- Infrastructure baseline documentation for POC build and configuration
- Initial FinOps dashboards capturing cost baselines and usage metrics
- Set up Role-Based Access Control (RBAC) to environments.

### **Data Workstream**

- Data catalog and business glossary populated with metadata and provenance
- PII detection and classification policies
- Encryption and key management policy
- Real-time pipeline monitoring dashboards

- Validated Gold-layer datasets meeting accuracy and reconciliation thresholds
- End-to-end data audit trail documentation
- Frontend prototype and API documentation
- POC validation report confirming data accuracy, performance, and compliance

#### **Organizational Change Management (OCM) / Organizational Readiness Workstream**

- Organizational communications package aligned to modernization objectives
- Training materials and recorded sessions for all user groups
- Change impact summary and adoption dashboard
- Organizational readiness plan for next-phase workforce enablement

#### **Project Management Office (PMO) Workstream**

- Project Charter, Work Breakdown Structure (WBS), Communications Plan, and RACI Matrix
- Baseline metrics report (pre-implementation performance baseline)
- POC success KPI report
- Lessons learned summary
- Validated cost savings and efficiency gains report
- Procurement and Decision Package artifacts, including finalized SOW, requirements, cost analysis, and evaluation criteria for OSPI IT Investment Review and Decision Package submission

## **6.0 Proposed 8-Month Timeline**

- Milestone 1 (Initialization): Kick-off, Tiger Team formed, network/security guardrails in place.
- Milestone 2 (Enablement): Governance framework and FinOps dashboards baselined; PII and encryption policies enforced.
- Milestone 3 (Execution): Data pipelines through Bronze–Silver–Gold layers configured; OCM training and adoption tracking underway.
- Milestone 4 (Validation): UAT complete, KPIs documented, and compliance verified

## **7.0 Vendor Qualifications**

#### **Core Roles:**

- Senior Project Manager
- Senior Business Analyst
- Enterprise Architect

- Data Architect / Integration Specialist
- Organizational Change Management (OCM) Lead
- Cybersecurity and Compliance Specialist

**Preferred Qualifications:**

- Proven consulting experience in statewide system implementations.
- Preferred experience with education data systems, including management of key data elements, SIS integration, and data governance, with demonstrated alignment to Enterprise IT Governance, SDLC, and Change Management practices.
- Strong partner engagement and change management capabilities.
- Familiarity with Washington State education systems and compliance frameworks preferred.