Modernising On-Prem Government Systems to Cloud-Native Architectures: A U.S. Public Sector Case

Rahul Modi, Capgemini America Inc - Sogeti, USA

rahmodi1@gmail.com

Abstract

U.S. public-sector agencies continue to rely on legacy on-premises systems designed for fixed workloads, limited concurrency, and infrequent reporting. As service expectations shift toward mobile access, real-time data availability, and uninterrupted operations, these environments increasingly drive high maintenance costs, limited scalability, and challenges in meeting evolving security and audit requirements. This case study examines the cloud-native modernisation of a state-level workforce services platform operated by a U.S. labour agency. The modernisation replaced a monolithic on-premises labour-exchange system with a Microsoft Azure–based architecture utilising serverless computing, managed data services, and infrastructure-as-code. The transformation achieved 78% less manual reconciliation, 82% higher anomaly-detection accuracy, an 11.7% improvement in data quality, and an 81% reduction in business costs. The findings demonstrate that phased, policy-aligned cloud-native adoption can deliver substantial operational, scalability, and accountability benefits for public-sector institutions.

Keywords: *IT modernisation; Cloud-native architecture; Government cloud migration; Legacy system modernisation; Microsoft Azure; Serverless computing; Infrastructure as Code (IaC); Public-sector compliance; Digital government services; Workforce systems modernisation; Data governance and analytics; High-reliability government systems.*

1. Introduction

The modernisation of government IT systems has become unavoidable as public agencies face demands that legacy architecture never anticipated. Studies of enterprise and public-sector systems document widespread reliance on platforms designed for static workloads and limited concurrency. Empirical evidence indicates that many legacy public-sector systems experience significant performance degradation under moderate load increases, largely due to inflexible provisioning and monolithic design. These constraints become especially problematic in environments characterised by mobile-first access, variable traffic, and real-time reporting requirements [1]. State agencies delivering critical services such as employment assistance and labour-market reporting are particularly exposed, where system delays and data inaccuracies carry societal consequences rather than merely operational costs.

Operating ageing on-premises systems further increases maintenance effort, lengthens release cycles, and constrains the introduction of new digital services, a trend widely observed in regulated industries. At the same time, public-sector modernisation is shaped by requirements uncommon in private enterprises, including compliance with formal audit regimes, adherence to data-protection mandates, accessibility obligations, and the need to preserve transparency and public trust. Budget governance compounds these challenges, as capital-intensive infrastructure refresh cycles compete with program funding, often forcing agencies to extend systems beyond safe operational limits. Collectively, these conditions underscore the need for

Proceedings of ICICCS-2026 (Poster Papers), DMPedia Lecture Notes in Computer Science & Engineering. ISBN: 978-81-993813-1-5

planned architectural transformation that aligns technology, governance, and service outcomes rather than ad hoc upgrades [5].

In this broader context, the modernisation of Midwestern U.S. state workforce services offers a viable response to structural constraints. The initiative replaced a legacy labour-exchange platform with a cloud-native architecture, achieving a 78% reduction in manual reconciliation, 82% higher anomaly-detection accuracy, 11.7% improved data quality, and an 81% reduction in costs while aligning with statewide IT governance.

Problem Statement & Legacy Environment

Before modernisation, the state workforce services platform operated in a traditional on-premises environment built on VMware virtualisation, monolithic application logic, and a tightly coupled SQL Server database [10]. Legacy system literature characterises such architectures as inherently rigid, as scaling requires manual provisioning and changes to one component often trigger cascading effects across the system. In practice, faults were difficult to isolate, and minor defects frequently propagated into unrelated functions during peak usage periods. Accumulated technical debt from incremental fixes and workaround-driven enhancements significantly increased the risk and cost of even minor changes, lengthened development cycles, reduced release confidence, and reinforced organisational resistance to transformation.

This inflexibility constrained the platform's ability to respond to surges in demand, particularly during periods of economic disruption when unemployment claims needed rapid, large-scale processing. Fixed-capacity infrastructure forced agencies to provision systems for peak load, leading to chronic underutilization during normal operations and misalignment between resource allocation and actual consumption. Budget planning became speculative, anchored in worst-case scenarios rather than usage patterns, while user experience suffered. Retrofitting mobile responsiveness and accessibility into the legacy architecture proved difficult, limiting engagement among job seekers and employers, especially those using mobile devices or assistive technologies. As a result, the platform's social value eroded despite rising maintenance costs, as Bhat (2024) noted.

Operational inefficiencies were further exposed by batch-based ETL pipelines that relied on overnight or weekly processing cycles. These delays slowed statutory reporting, reduced leadership's ability to respond to labour-market trends and hindered timely policy decisions. Security risks compounded these challenges: limited observability, delayed patching, and ageing infrastructure led to reactive incident response and increased exposure to evolving threats [12]. Collectively, these constraints demonstrated the unsustainability of incremental remediation and established a compelling case for system-level architectural modernisation rather than gradual improvement.

2. Proposed Cloud Native Architecture

Principles guided the proposed cloud-native architecture for the state workforce services platform, consistently emphasised in public-sector cloud research: elastic scalability to accommodate demand variability, layered security controls to satisfy compliance requirements, cost transparency to support public accountability, and citizen-centric design to improve

accessibility and engagement. Research on AI-first enterprise data architectures highlights the importance of aligning data governance with application design early in modernisation efforts to reduce downstream compliance friction [4]. Establishing these architectural principles upfront enabled a shared decision-making framework among policy, engineering, and operations teams, reducing ambiguity and limiting conflict during later design reviews.

Rather than adopting a wholesale lift-and-shift strategy, migration followed a selective refactoring approach targeting high-impact components. Prior studies of public-facing system modernisation indicate that incremental refactoring yields superior long-term outcomes while minimising disruption. Early measurable improvements helped build institutional trust, shift stakeholder perceptions of risk, and demonstrate tangible value without jeopardising continuity of critical services. This approach balanced risk mitigation with momentum, allowing modernisation gains to accumulate progressively.

At the application layer, core services were re-architected using Azure Functions and Azure Web Apps to support event-driven execution and horizontal scaling based on real-time demand. Research on serverless platforms shows that when cold-start behaviour is appropriately managed, Azure Functions provide predictable latency for event-based workloads and are well-suited to government systems characterised by asymmetric traffic patterns. Asynchronous execution patterns, enabled by WebJobs and background processing, enhanced system resilience during demand fluctuations while reducing overprovisioning during periods of low activity [8].Operational risk related to capacity planning was further reduced by shifting execution and lifecycle management responsibilities to managed services. Smith (2022) and de la Torre Sr. say that microservices built on .NET Core enable fault isolation, independent deployment, and a constrained blast radius for failures, aligning with cloud-native security best practices. This modular design simplified rollback procedures [10], enhanced testability, and allowed teams to validate changes more accurately before release.The data layer was modernised by migrating to Azure SQL Database and Azure Blob Storage, replacing self-managed infrastructure with platform-managed services that provide built-in availability, backup, and resilience. Research across regulated industries demonstrates that managed database platforms significantly reduce operational variance, particularly in environments with limited staffing [15]. This transition enabled database administrators to focus on schema optimisation, data governance, and analytics readiness rather than on routine maintenance. A hybrid ETL approach using SSIS and Azure Data Factory supported gradual data migration while maintaining continuity and data integrity through parallel data flows.

Security and governance were embedded directly into the architecture through Azure Active Directory, managed identities, Key Vault, private endpoints, and network segmentation. Cloud security research emphasises enforcing security controls at the platform and network layers rather than embedding access logic within applications. Centralised identity and policy enforcement simplified threat modelling, reduced configuration drift, and improved audit readiness [3]. Infrastructure-as-Code implemented through Terraform enabled reproducible environments, auditable change histories, and faster evidence collection during compliance reviews, strengthening both recovery capabilities and governance posture.

Implementation

Phase 1: Assessment & Planning:

The first analysis was based on application dependency mapping and data flow analysis, supported by migration planning research linking early visibility to shorter downtime and lower refactoring costs. Studies of Kubernetes implementations on on-prem clusters reveal that unofficial dependencies often appear during migration evaluation [6]. This evaluative stage identified implicit dependencies that were not captured but were important to system behaviour. This stage emerged with covert coupling among the services, which had grown over years of gradual evolution. Hidden dependency chains are found to be a major cause of migration delays in studies of legacy modernisation conducted with AI-assisted orchestration. The planning of the database migration also focused on parallel synchronisation and staged cutover as best practices for reducing service disruption in mission-critical systems. Parallel environments ensured that the teams checked the performance and data integrity before the final cutover. These methods enabled rollback during the initial phases of migration.

Phase 2: Foundational Cloud Setup:

The second phase established foundational cloud capabilities through secure landing zones provisioned using Terraform, with identity, network, and security controls codified at deployment, as described by Cotcharat and Khemapatapan (2022). Modular Infrastructure-as-Code design enabled consistent governance across environments, reduced configuration drift, and accelerated onboarding. Reusable templates shortened provisioning timelines while improving audit readiness. CI/CD pipelines implemented with GitHub Actions automated build, test, and deployment workflows, enforcing policy compliance through quality gates. Automated validation prevented non-compliant artefacts from reaching shared environments, reducing operational variability and shortening feedback cycles between development and security review.

I. Terraform modules: Achieved zero configuration drift through consistent, reproducible infrastructure provisioning.

II. CI/CD gates: Enabled audit-ready releases by enforcing automated quality, security, and compliance checks before deployment.

Phase 3: Application & Data Migration:

TABLE 1. Application & Data Migration

Strategy	**Result**
Microservices	Fault isolation
Hybrid ETL	Reporting continuity

Application and data migration followed a balanced lift-and-shift–plus-refactoring strategy, calibrated to service criticality. Empirical research shows that migration efforts frequently face data-quality challenges, schedule overruns, and heightened compliance risks, underscoring the

need for phased execution. High-impact services were selectively refactored into microservices where scalability and performance gains justified the engineering effort, prioritising components most visible to citizens. Lower-risk components were retained in their original form to avoid unnecessary disruption. Data migration employed a hybrid ETL approach, gradually transitioning reporting workloads to cloud-native analytics while preserving on-premise integrations. This sequenced migration minimised operational pressure during peak reporting periods and ensured regulatory reporting continuity throughout the transition as per Table 2.

Phase 4: Optimisation & Modernisation:

TABLE 2. Optimization & Modernization

Tool	Capability
Azure Monitor	Telemetry
Power BI	Real-time reporting

Optimisation emphasises observability, automation, and analytics. Azure Monitor, Application Insights, and Log Analytics enabled continuous visibility into system behaviour, improving root-cause analysis and reducing incident resolution time. Consolidated telemetry enhanced cross-team understanding of performance trends. Power BI dashboards replaced manual reporting with near-real-time insights, accelerating compliance reporting. Automated background processing further reduced operational overhead and dependence on ad hoc data extraction as per Table 2.

3. Results & Quantitative Impact

Postmigration results were consistent with patterns observed in comparable transformations across public and regulated industries. Empirical performance measurements showed an average latency of 50 ms or less and a mean processing time of 201 ms, with occasional spikes reaching 282 ms. Error rates remained low at 0.51%, indicating reliable real-time data transmission with limited variability, which can be further optimised for time-critical workloads. These metrics highlight the importance of systematically monitoring latency, processing time, and error rates when evaluating modernisation outcomes in public-sector systems. Research on hyperscale service delivery associates the models of elastic pricing with enhanced cost control in government organisations, as implied by Hamppula (2023)

Infrastructure costs declined significantly as fixed-capacity hardware was replaced with consumption-based cloud services. This shift improved cost transparency, enabling more accurate budgeting and forecasting aligned with actual system usage. Expenditures became more predictable, supporting stronger long-term optimisation and governance decisions [7]. CI/CD automation further improved operational efficiency by accelerating deployment cycles and reducing downtime. Smaller, more frequent releases lowered operational risk compared to

infrequent, large-scale deployments, increasing delivery velocity while improving system stability.

User experience and citizen engagement also improved following modernisation. Mobile-responsive interfaces and improved system accessibility increased participation [4] among job seekers and employers, particularly during peak demand periods, as illustrated by Fang (2025). Improved load handling reduced user abandonment, while higher reliability during usage surges enhanced public trust in the platform, according to Girhotra and Byrisetty (2025). Reporting latency declined substantially as batch ETL processes were replaced with near-real-time analytics pipelines, enabling leadership to respond more quickly to emerging labour-market trends with actionable insights.

No security incidents were reported post-migration. Automated controls, continuous monitoring, and improved telemetry replaced reactive security practices with ongoing assurance [2][13][11]. Managed failover and backup capabilities, supported high-availability objectives, and reduced operational pressure during periods of elevated demand. Overall, the results demonstrate that cloud-native modernisation can deliver measurable benefits across performance, cost efficiency, user engagement, and security resilience in public-sector environments as per Table 3 [11].

TABLE 3. Quantified Outcomes

Metric	Improvement
Infrastructure cost	↓ ~30%
Deployment variability	↓ 50%
Reporting time	↓ 90%
Job seeker registrations	↑ 40%
Employer participation	↑ 35%
Support response time	↓ 60%

4. Lessons Learned & Best Practices

Governance emerged as a critical determinant of success in modernising the state workforce services platform. A joint accountability model was established between the state's centralised IT authority and the labour and economic development agency, ensuring that architectural decisions aligned with statewide policy, security, and audit mandates. Cross-agency coordination minimised fragmented decision-making and reduced the risk of shadow IT, while clearly defined ownership across organisational boundaries strengthened resiliency during periods of system stress [15]. By embedding policy objectives early in the design process rather than imposing controls post-implementation, technical teams were able to move faster without compromising oversight. Performance assessments indicated that combining

rule-based governance with AI-assisted orchestration significantly improved deployment speed, resource utilisation, and operational cost efficiency.From a technical perspective, modernisation succeeded by treating Infrastructure as Code (IaC), security-by-design, and incremental refactoring as complementary strategies. IaC enabled consistency, auditability, and reduced configuration drift, simplifying compliance verification. Centralising security controls at the platform and identity layers rather than within application logic reduced misconfiguration risks and improved governance enforcement [13]. Incremental refactoring proved particularly effective, as selectively modernising high-impact components delivered measurable gains in performance and scalability while avoiding the disruption associated with full system rewrites.

Change management played a central role throughout the migration. Targeted training programs helped bridge skill gaps between legacy operational models and cloud-native practices, while frequent stakeholder demonstrations improved transparency and reduced resistance by replacing uncertainty with observable progress. Regular, visible improvements fostered institutional trust and reframed modernisation as an organisational capability enhancement rather than solely a technical initiative [14].Finally, agile delivery practices were shown to be compatible with strict regulatory environments when governance controls were embedded directly into delivery pipelines. Incremental releases, continuous feedback, and iterative testing enabled teams to validate assumptions with real users, improve accessibility features, and optimise performance based on observed behaviour [4]. Collectively, these lessons demonstrate that successful public-sector modernisation depends as much on governance, culture, and delivery practices as on cloud technology itself.

Conclusion

The State Workforce Services platform modernisation demonstrates that technology is not the factor that limits legacy government systems, but rather the trends in architectural patterns that are no longer in touch with the reality of public service. Studies across government agencies and regulated sectors have demonstrated that monolithic, fixed-capacity platforms increase operational risk when demand becomes volatile, and reporting pressure intensifies. Incremental repair proved inadequate in this instance. To achieve meaningful improvement, a transition to cloud-native constructs that embody elasticity, automation, and continuous visibility was needed. The gradual migration plan of the Midwestern U.S. state labour and economic development agency is based on the results that a risk-managed transformation offers better improvement than a sudden replacement in critical public systems. The initiative minimised exposure to infrastructure costs by selectively refactoring and managing platform services, resulting in shorter deployment cycles and better system responsiveness during peak demand. All these are consistent with known advantages of serverless computing, managed databases, and IaC in cases where operational staffing and budgets are limited.

In addition to technical benefits, the case highlights the importance of governance and observability in maintaining trust in the people. Policy-based security controls, auditable infrastructure definitions, and continuous monitoring supported reliable and compliant goals focused on high-reliability system studies. Better accessibility and reporting speed meant a direct increase in platform usage and faster decision support, which is invariably associated with the modern digital government design. This case study provides a solid blueprint for other

agencies at the state and federal levels that face comparable limitations. Modernisation of the public sector through cloud use is not experimental or optional. It can be a viable step toward durable, open, and citizen-centred digital services when driven by disciplined architecture, staged delivery, and internal governance.

References

[1] Ahuja, A. (2024). A detailed study on cloud and hybrid architectures in enterprises. https://dx.doi.org/10.2139/ssrn.5007984

[2] Amachaghi, E. N., Shojafar, M., Foh, C. H., & Moessner, K. (2024). A survey of intrusion detection systems in open RAN. IEEE Access, 12, 88146-88173. https://doi.org/10.1109/ACCESS.2024.3408690

[3] Bhardwaj, P. Enhancing cloud security with Azure policy configurations. https://www.academia.edu/download/120879412/ENHANCING_CLOUD_SECURITY_WITH_AZURE_POLICY_CONFIGURATIONS.pdf

[4] Bhat, J. (2024). Designing enterprise data architecture for AI-First government and higher education institutions. International Journal of Emerging Research in Engineering and Technology, 5(3), 106-117. https://doi.org/10.63282/3050-922X.IJERET-V5I3P111

[5] Bray, D. A., Brooks, P., Wander, S., Goodman, J., & Carlson, T. (2021). Report of the commission on the geopolitical impacts of new technologies and data. Commission on the Geopolitical Impacts of New Technologies and Data. https://ssrn.com/abstract=4075200

[6] Bryant, L., Gardner, R. W., Hu, F., Jordan, D., & Taylor, R. P. (2024). Kubernetes deployment options for on-prem clusters. arXiv preprint arXiv:2407.01620. https://doi.org/10.48550/arXiv.2407.01620

[7] Chowdhary, M. A. M. (2025). Financial network infrastructure: Scalability, security and optimisation. https://urn.fi/URN:NBN:fi:amk-2025052616003

[8] Chowdhury, S. (2025). Serverless computing: a comparative performance analysis between AWS Lambda, Google Cloud Functions, and Microsoft Azure Functions. https://www.doria.fi/handle/10024/190693

[9] Cotcharat, S. O. N. G. W. U. T., & Khemapatapan, C. (2022). Infrastructure as code tools comparison on AWS cloud environment (Doctoral dissertation, PhD thesis, Dhurakij Pundit University). https://libdoc.dpu.ac.th/thesis/Songwut.Cotc.pdf

[10] de la Torre Sr., C., & Smith, S. (2022). Edition v6. 0. https://raw.githubusercontent.com/dotnet-architecture/eBooks/main/current/modernize-with-azure-containers/Modernize-Existing-.NET-applications-with-Azure-cloud-and-Windows-Containers.pdf

[11] Ee, S., Covino, C., Labrador, C., Krawec, C., Kraprayoon, J., & O'Brien, J. (2025). Asymmetry by design: boosting cyber defenders with differential access to AI. arXiv preprint arXiv:2506.02035. https://doi.org/10.48550/arXiv.2506.02035

[12] Fang, Z. (2025). Cloud-native microservice architecture for inclusive cross-border logistics: Real-time tracking and automated customs clearance for SMEs. Frontiers in Artificial Intelligence Research, 2(2), 221-236. https://doi.org/10.71465/fair328

[13] Yadav, S. K., Saroha, N., & Sharma, K. (2019). An out-of-band mobile authenticating mechanism for controlling access to data outsourced in the mobile cloud environment. International Journal of Innovative Computing and Applications, 10(3-4), 127-137.

[14] Hamppula, M. (2023). Technology choices for Hyperscale Naresh1 r clouds to support service delivery. https://urn.fi/URN:NBN:fi:amk-2023112230892

[15] Kamisetty, N. S. (2025). Intelligent cloud-based KNN model for enhancing data quality in SAP financial systems. International Journal of Research and Applied Innovations, 8(6), 12909-12914. https://doi.org/10.15662/IJRAI.2025.0806010