International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Exploring Cloud-Adaptable Architectures for Public TLS Certificate Issuance Under Established Trust Constraints

Naresh Charugundla

Abstract: Public Transport Layer Security (TLS) certificate issuance systems form a critical component of the global internet trust ecosystem, enabling encrypted and authenticated communication for web services, cloud platforms, and distributed applications at internet scale. These systems operate within a well-defined compliance framework anchored by RFC 5280 and the CA/Browser Forum Baseline Requirements, which establish outcome-oriented expectations around private key protection, auditability, system integrity, and separation of duties. Traditionally, operators have satisfied these expectations through tightly controlled infrastructure environments that minimize ambiguity over administrative access and operational behavior. The increasing adoption of cloud-based infrastructure raises substantive questions about whether and how equivalent assurances can be established within deployment models that differ structurally from those in which existing compliance expectations were formed. This article presents an independent, standards-informed exploration of how public TLS certificate issuance systems might evolve toward cloud-adaptable architectures while remaining aligned with established trust constraints. The analysis identifies key compliance foundations, characterizes commonly observed deployment patterns, and examines the tensions that cloud adoption introduces. Five architectural directions — cryptographic isolation via externalized hardware security modules, verifiable execution environments, append-only audit log integrity, policy-driven control planes, and layered composable trust models — are evaluated against established expectations. The findings indicate that trust in certificate issuance systems is fundamentally a function of demonstrable assurance rather than deployment environment and that cloud-adaptable architectures can potentially satisfy established expectations provided controls are explicit, verifiable, and independently validated.

Keywords: *Public Key Infrastructure, TLS Certificate Issuance, Cloud Architecture, Hardware Security Modules, Certificate Authority Compliance, Trusted Execution Environments, Audit Integrity*

1. Introduction

Public TLS certificate issuance systems occupy a position of structural importance within the global internet trust ecosystem. By issuing X.509 certificates that bind domain identities to cryptographic keys, publicly trusted Certificate Authorities (CAs) enable browsers, operating systems, and cloud services to establish verified, encrypted connections across the open internet. The scale of this function is substantial: certificate issuance events underpin billions of TLS connections per day, spanning web services, application programming interface (API) infrastructure, cloud platform communications, and software distribution pipelines. The integrity of these systems—their ability to issue valid certificates only to legitimately authorized entities, to protect the private keys that underpin those certificates, and to maintain auditable records of all issuance activity—is therefore not merely an operational matter but a fundamental prerequisite for internet security at scale.

The compliance framework governing public TLS certificate issuance is well established. RFC 5280 defines the X.509 certificate profile that governs certificate structure and semantics [1]. The CA/Browser Forum Baseline Requirements define the operational, technical, and audit expectations that publicly trusted CAs must satisfy to maintain

Independent Researcher, USA

inclusion in browser and operating system root programs [2]. These frameworks are deliberately outcome-oriented: they specify what assurances must be demonstrated — strong key protection, comprehensive auditability, system integrity, and separation of duties — without prescribing specific infrastructure technologies or deployment models. In practice, however, the need to demonstrate these assurances with a high degree of confidence has historically favored deployment models that prioritize tight control over infrastructure, administrative access, and cryptographic operations.

Nevertheless, the trend for computing over the last decade was to move towards cloud-based infrastructure. Cloud platforms offer scalable elasticity, rapid deployment cycles, access to advanced security primitives, including Hardware Security Modules (HSMs) as managed services, and sophisticated identity and access management capabilities. These capabilities are increasingly relevant to the operational requirements of certificate issuance systems. Yet the structural characteristics of cloud environments—multi-tenancy, separation between infrastructure operators and system owners, abstraction of physical hardware, and reliance on provider-managed trust primitives—introduce questions about how the established compliance expectations for public TLS issuance can be evaluated and demonstrated within these new deployment models.

This tension between traditional deployment patterns and evolving infrastructure capabilities is the central problem this article examines. The question is not whether cloud environments are inherently more or less secure than traditional alternatives. It is whether and how the specific assurances required of publicly trusted certificate issuance systems can be credibly established within cloud-based architectures—and what architectural patterns are most likely to support that alignment. This is a question of growing practical significance as certificate issuance infrastructure evolves alongside the cloud platforms it increasingly depends on.

The primary contributions of this article are (1) a structured characterization of the compliance foundations and their interpretive logic as outcome-based requirements rather than infrastructure mandates; (2) an analysis of five architectural directions for cloud-adaptable TLS issuance systems and their alignment with established expectations; and (3) an identification of the trade-offs, open questions, and governance considerations that any transition toward cloud-adaptable architectures must address. The article proceeds as follows. Section 2 establishes the trust and compliance foundations. Section 3 characterizes observed deployment patterns. Trade-offs with cloud adoption and trust requirements are considered in section 4, while section 5 discusses architectural options. Section 6 considers compliance interpretations in the cloud. Section 7 covers trade-offs and open questions. Section 8 covers future work. Section 9 concludes.

2. Trust and Compliance Foundations in Public TLS Issuance Systems

The compliance framework governing publicly trusted TLS certificate issuance is built around a small number of foundational principles that define the assurances operators must demonstrate, regardless of the specific technologies or deployment architectures they employ. Understanding these principles in their interpretive form—as goal-based expectations rather than prescriptive infrastructure requirements—is a prerequisite for evaluating whether alternative deployment models can satisfy them.

The most important goal is to protect Certificate Authority private key material. The CA private keys for the root and intermediate CA hierarchies form the foundation of the entire trust model represented by the CA. If one of these was compromised, an attacker could issue bogus, but cryptographically valid, certificates to any domain, format, or protocol, thereby breaking security for all authenticated transport on the Internet. RFC 5280 and the CA/Browser Forum Baseline Requirements impose wide-ranging requirements on the generation, storage, and use of these keys [1][2]. In practice these requirements are usually met by specific-purpose hardware cryptographic computers (Hardware Security Modules). These devices generate and store cryptographic keys in a tamper-resistant hardware operating environment and perform signing operations in a secure execution environment such that plaintext private key material is never exposed outside the secured boundary of the device. FIPS 140-2 Level 3 validation is de facto

standard assurance for HSMs used in this application.

Second, thorough auditability. Certificate issuance systems must maintain an auditable record of all actions that affect the security of the certificate, including certificate requests, validation decisions, issuance actions, key usage, and administrative actions. Prominent requirements on audit logs include tamper-evidence (the ability to detect unauthorized modification or deletion of log entries) and auditability (the ability to associate any entry to a specific actor and timestamp). Certificate Transparency (CT) defined in RFC 6962 and RFC 9162 generalizes these requirements to the domain of public certificates issued by publicly trusted Certificate Authorities (CAs) that must be logged into append-only public logs whose integrity is provided by a Merkle tree [3][4]. This layered, auditable architecture, which combines internal operational audit trails with publicly available certificate transparency logs, allows for validation of internal compliance and accountability to ecosystem participants.

The third requirement is separation of duties: The key operations of a certificate-issuing system, e.g., key management, certificate issuance approval, and system administration, must be separated such that no one person can make key administrative decisions without the knowledge of other parties. This principle reflects industry best practice for high-assurance information security systems as well as the CA/Browser Forum Baseline Requirements, which require role-based access controls to enforce important separation between conflicting security roles [2]. The fourth principle, system integrity and operational determinism, requires issuance workflows that complete in a predictable and verifiable way, with clearly defined inputs, outputs, and validation steps. This determinism supports both operational reliability and audit validation, enabling external parties to verify system correctness based on observed evidence rather than on trust in operator claims.

A critical interpretive point about these four principles is that they are expressed as outcome expectations rather than as infrastructure requirements. The Baseline Requirements do not mandate on-premises deployment, specific hardware vendors, or particular network topologies. They require that operators demonstrate — through audit evidence acceptable to recognized auditors — that the required assurances are continuously maintained. This distinction between prescriptive infrastructure requirements and outcome-based trust expectations is foundational to the question of whether cloud-adaptable architectures can satisfy them, and it is the interpretive lens through which the remainder of this analysis proceeds.

Compliance Principle	Core Requirement	Typical Mechanism (Traditional)	Key Interpretive Question
Private key protection	Keys never exposed in plaintext; controlled usage	HSM — FIPS 140-2 Level 3	Can cloud HSM/KMS provide an equivalent controlled execution boundary?
Auditability	Tamper-evident, attributable records of all operations	Internal log + CT log (RFC 9162)	Can distributed cloud logs maintain completeness and integrity guarantees?
Separation of duties	No single actor can perform conflicting sensitive functions	Role-based admin segmentation	Can cloud IAM policies enforce equivalent functional separation?
System integrity	Predictable, verifiable issuance workflow behavior	Controlled infra + access restrictions	Can attestation mechanisms evidence equivalent execution integrity?

Table 1. Compliance principles for publicly trusted TLS certificate issuance systems, traditional implementation mechanisms, and key interpretive questions for cloud deployment evaluation.

3. Observed Deployment Patterns in Practice

CAs that issue publicly trusted TLS certificates to production environments often have one or more deployment patterns that are considered best practice. This is based upon the deployment patterns of CAs who have undergone regular compliance audits against standards such as WebTrust for CAs and ETSI EN 319 411. Such patterns are not explicitly mandated in the requirements, but rather

meet the outcome-based criteria in Section 2. An understanding of them is critical to interpreting and comparing other deployment models.

A common observation of CA systems is that they are contained in controlled environments where the CA private key material is only stored or consumed by a specific subset of infrastructure that is under strict control (where physical access, segmentation from general purpose corporate infrastructure, and sharing is minimized). The rationale is straightforward: demonstrating key protection and system integrity is substantially easier when the environment in which those properties must hold is small, well-defined, and subject to direct administrative control. Reducing environmental complexity reduces audit scope and decreases the number of variables that must be evidenced as controlled during compliance assessments. Luo et al.'s empirical analysis of certificate validation implementations confirmed that real-world certificate issuance and relying-party systems encode deep assumptions about environmental determinism in their operational logic [5].

A second common pattern is the implementation of restricted administrative domains with carefully defined role assignments. The roles usually defined are certificate requestor, validation officer, issuance authority, key custodian, system administrator, and auditor. Separation of duties is typically enforced through technical controls, such as multi-factor authentication or using a hardware token, in addition to procedural controls, such as requiring dual authorization for sensitive operations like CA key generation or root CA activation. The goal is to ensure that audit evidence of separation of duties can be independently verified, rather than relying solely on an operator's assertion of administrative practice.

A third characteristic involves the design of audit and logging infrastructure for determinism and completeness. Certificate issuance systems in compliant environments typically implement logging at multiple layers — application, system, network, and HSM — with log aggregation into tamper-evident stores whose integrity can be independently verified. The operational emphasis is on ensuring that no security-relevant event occurs outside the observable audit boundary and that the audit trail is sufficiently complete and correlated to reconstruct the full issuance workflow for any specific certificate. Public Certificate Transparency logging, now mandatory for publicly trusted TLS certificates under CA/Browser Forum policy, extends this audit infrastructure into an externally verifiable domain where misissuance can be detected by ecosystem monitors [3][4].

These observed patterns should be interpreted as responses to the practical demands of compliance demonstration rather than as standards-mandated architectural requirements. Different operators apply variations of these patterns based on their specific risk models, infrastructure capabilities, and interpretations of audit expectations. All of this has in common a desire to reduce the ambiguity in the evidence chain: the smaller and more controlled the operational environment, the greater the chance that compliance evidence can be related to compliance assertions that auditors can verify. This is certainly true for the question of cloud-adapted alternatives: the question is not whether cloud environments can be made secure, but whether they can produce compliance evidence as clearly as tightly controlled environments have historically produced.

4. Tension Between Cloud Adoption and Trust Requirements

The structural properties of cloud infrastructure may conflict with the patterns of showing compliance described in Section 3. These tensions do not reflect fundamental incompatibilities between cloud environments and the underlying compliance principles—the outcome-based nature of those principles means that different technical approaches can potentially satisfy them. They do, however, reflect real differences in how assurance must be established and evidenced within cloud deployment models compared to traditional, tightly controlled environments.

The most prominent tension concerns administrative trust boundaries. In traditional CA environments, the set of individuals with privileged access to CA systems is small, explicitly enumerated, and subject to background screening and contractual obligations. In cloud deployments, a distinction arises between the infrastructure provider — whose administrative staff have physical and logical access to the underlying hardware — and the system operator, who configures and manages the deployed workloads. This separation of administrative domains introduces a layer of trust that must be explicitly addressed in compliance evidence: how

does the operator demonstrate that the infrastructure provider's privileged access does not constitute an unauthorized pathway to sensitive cryptographic operations or audit data? This question is not unanswerable, but it requires a more complex evidence chain than traditional single-domain administrative models [6].

A second area of tension involves multi-tenancy and isolation assurance. Cloud platforms are architected to support multiple independent workloads on shared physical infrastructure, with isolation enforced through virtualization, containerization, and software-defined network controls. Modern cloud providers invest substantially in isolation mechanisms, and empirical security research has generally found cloud isolation to be robust against common attack classes. Nevertheless, the shared physical substrate means that isolation is enforced by software and firmware layers rather than by physical separation—a difference that some compliance frameworks and auditors have historically treated as a meaningful distinction for high-assurance systems. Trusted Execution Environments (TEEs) represent a hardware-based approach to addressing this concern by providing isolated execution boundaries whose integrity can be cryptographically attested [7].

Key custody in cloud environments introduces a third tension. Cloud providers offer managed HSM services, such as AWS CloudHSM, Azure Dedicated HSM, and Google Cloud HSM, that are FIPS 140-2 Level 3 validated and provide hardware-backed key generation and signing operations. These services provide strong technical controls analogous to on-premises HSMs. The compliance question concerns the trust relationship between the CA operator and the cloud provider with respect to key material: the operator must demonstrate that the provider cannot access the key material, that the key usage policy is enforced by the hardware boundary, and that the provider's operational procedures do not create exposure pathways that an on-premises HSM operator would not face. Attestation mechanisms for TEEs, as analyzed by Ménétrey et al., provide a formal basis for evidencing hardware-enforced execution boundaries—the question is whether current attestation frameworks are mature enough to satisfy CA audit expectations [8].

Auditability in distributed, multi-region cloud environments represents a fourth tension. Certificate issuance systems deployed in cloud environments may distribute components across multiple services, availability zones, and geographic regions. Log data may be generated across these distributed components and aggregated through cloud-native logging services. Ensuring that the resulting audit trail is complete, tamper-evident, and sufficiently correlated to support issuance workflow reconstruction requires careful architectural design. The use of append-only, cryptographically verifiable log structures — as demonstrated in the Certificate Transparency ecosystem via RFC 9162 — provides a technically sound basis for this assurance, but integrating such mechanisms into cloud-native logging architectures requires deliberate design choices that may not be present in default cloud logging configurations [4].

Tension Area	Cloud Characteristic	Compliance Concern	Potential Resolution Mechanism
Administrative trust boundary	Provider/operator separation	Provider admin access to sensitive operations	Cryptographic isolation of key operations from provider visibility
Multi-tenancy and isolation	Shared physical substrate	Software-enforced vs. physical isolation	Trusted Execution Environments with hardware attestation
Key custody	Managed HSM services	Provider relationship to key material	FIPS 140-2 L3 cloud HSM + attestation evidence
Auditability	Distributed, multi-service logging	Log completeness and tamper-evidence	Append-only cryptographic log structures (RFC 9162 model)
Control plane abstraction	Provider-managed infrastructure layer	Visibility into underlying system behavior	Immutable infrastructure + attestation-based integrity evidence

Table 2. Tensions between cloud adoption and public TLS certificate issuance compliance requirements, with potential resolution mechanisms.

5. Architectural Possibilities for Cloud-Adaptable TLS Issuance Systems

Given these compliance underpinnings and tensions, below we identify five cloud architectural directions that may help ease cloud deployments' alignment with the assurance expectations of publicly trusted certificate issuance systems. These architectural directions should be considered as areas for experimentation rather than prescriptions, as their applicability will depend on compliance interpretations, operational constraints, and infrastructure capability maturity. A common theme across all five is the movement from implicit environmental trust based on controlling the physical and administrative boundaries around an infrastructure environment, to explicit verifiable assurance based on cryptographic or attestation-based machinery to evidence the required properties, regardless of the infrastructure deployment environment.

5.1. Cryptographic Isolation via Externalised Hardware Security Modules

The most easily deployed architectural control for key protection in the cloud is an explicit cryptographic boundary around CA private key material, independent of the infrastructure surrounding it. In this model, all private key generation and signing operations are performed within purpose-built HSM hardware. These can either be HSMs sited in on-premises data centers and networked to cloud-hosted issuance workloads through secure, encrypted channels, or cloud-native HSM services with FIPS 140-2 Level 3 certification. The issuance workflow is controlled through the API exposed by the HSM, and access policies to the CA's private keys, their usage, and audit logging are enforced by the HSM. In effect, there is an architectural separation such that even if the issuance system's cloud-hosted components were compromised the CA private keys remain inside the HSM. Cloud HSM services from major providers have reached a level of operational maturity and compliance validation that makes this approach technically credible, though the compliance evidence chain must explicitly address the trust relationship between the operator and the provider with respect to hardware initialization and administrative access [6].

5.2. Verifiable Execution Environments

Trusted Execution Environments (TEEs) provide hardware-enforced isolated execution boundaries within which code can run with strong assurances that its execution has not been tampered with by the host operating system, hypervisor, or infrastructure operator. Technologies including Intel Software Guard Extensions (SGX), AMD Secure Encrypted Virtualization (SEV), and ARM TrustZone offer different operational models but share the fundamental capability of producing cryptographic attestation evidence — signed measurements of the executing code and its configuration that a remote verifier can use to confirm that the expected software is running in a genuine, unmodified TEE. In the context of certificate issuance, TEEs may support a credible evidence base for system integrity and execution determinism by enabling independent verification that the issuance workflow code has not been modified. The security limitations of current TEE implementations are well documented — side-channel vulnerabilities, attestation chain complexities, and firmware dependency risks have all been analyzed in the literature [7][8] — and any compliance use of TEE attestation would require auditors to develop evaluation frameworks for these limitations. The direction is nevertheless worth exploring as TEE technology matures and attestation frameworks stabilize.

5.3. Append-Only Cryptographically Verifiable Audit Logs

Certificate Transparency provides a proven, internet-scale precedent for the use of append-only, cryptographically verifiable logs in certificate issuance accountability. RFC 9162 specifies the CT v2.0 log format, which uses Merkle tree structures to produce inclusion proofs for individual log entries and consistency proofs that allow any observer to verify that the log has grown only by addition rather than by modification or deletion [4]. Applying this architectural pattern to internal operational audit logging — not only to public certificate transparency logs — could provide a technically sound basis for demonstrating audit integrity in cloud-deployed issuance systems, even where the underlying log storage is distributed across cloud-native services. An internally operated CT-style log that captures all issuance workflow events, signed by a key held in an HSM boundary, would produce audit records whose integrity is verifiable by

external auditors without requiring them to trust the integrity of the cloud logging infrastructure itself. This approach is technically feasible using existing open-source CT log implementations and requires no novel cryptographic assumptions.

5.4. Policy-Driven Control Planes and Identity-Based Separation of Duties

Cloud platforms offer sophisticated identity and access management (IAM) capabilities that may support the enforcement of separation of duties through policy-based controls rather than through physical or network-level segregation. The different permissions for example for certificate requestor, validation authority, issuance decision-maker, key custodian, audit reviewer, etc., are captured in explicit IAM policies associated with different identities. Cryptographic policy knows no privilege escalation, and effectively enforces separation of duties. Multi-party authorization requirements for critical operations (such as activating a root CA or managing keys) can be implemented using approval workflows that require multiple authenticated role-holders to consent to a sensitive operation before it takes place. Instead of building a trail of separation of duties compliance in physical access logs, one can do so in IAM policy settings and authorization audit trails, both of which are well supported. As Phiayura and Teerakanok demonstrated in their framework for zero trust architecture migration, systematic policy encoding of access boundaries can produce compliance evidence of equivalent quality to traditional administrative segregation, provided the policy framework is correctly designed and independently audited [9].

5.5. Layered Composable Trust Models

A broader architectural perspective recognizes that no single mechanism — HSM key isolation, TEE execution integrity, append-only audit logs, or policy-driven access control — addresses all four compliance principles simultaneously. A layered composable trust model addresses this by combining multiple independently verifiable controls, each targeting a specific compliance principle, into a coherent architecture in which trust emerges from the composition of components rather than from the integrity of a single environment. Under this model, key protection is addressed by HSM isolation, execution integrity is evidenced by TEE attestation where applicable, audit integrity is enforced by cryptographic log structures, and separation of duties is implemented through IAM policy. External auditors can evaluate each layer independently, and the composite evidence chain provides assurance that is more resilient to component-level failures than a single-mechanism approach. The operational complexity of this layered model is greater than that of a tightly controlled traditional environment, but the compliance evidence it produces is potentially more precise and more amenable to independent verification [10].

Architectural Direction	Primary Compliance Principle Addressed	Technical Mechanism	Maturity Assessment
Externalized HSM isolation	Private key protection	FIPS 140-2 L3 HSM; cloud HSM service	High—well-established; provider offerings audited
Verifiable execution environments	System integrity; execution determinism	Intel SGX / AMD SEV attestation	Medium—technology mature; audit frameworks emerging
Append-only cryptographic audit logs	Auditability, tamper-evidence	Merkle tree log structures (RFC 9162 model)	High—CT ecosystem proven at internet scale
Policy-driven IAM control planes	Separation of duties	Cloud IAM policies; multi-party authorization	High—IAM tooling mature; audit integration established
Layered composable trust model	All four principles (composite)	A combination of the above mechanisms	Medium — no established audit template; requires custom scoping

Table 3. Architectural directions for cloud-adaptable TLS certificate issuance systems, primary compliance principles addressed, technical mechanisms, and current maturity assessment.

6. Interpreting Compliance Expectations in Cloud Contexts

The architectural directions explored in Section 5 raise a second-order question that is as practically significant as the technical feasibility of the mechanisms themselves: how should existing compliance expectations for public TLS certificate issuance systems be interpreted when applied to cloud-based deployment models? This is not a purely technical question. This includes the interpretative frameworks used by auditors and root program operators (or certifying authorities) to review evidence of compliance. which were designed for customary deployment models but may need to be adapted to consider cloud-native types of evidence.

The main point is that compliance frameworks express results, not requirements. For private key protection, the result is that CA keys will not be accessed, changed, or exfiltrated by unauthorized personnel during normal operations of the CA. In native deployments, the HSM will leave physical evidence in the form of access logs and dual control. In cloud deployments, the same outcome can potentially be evidenced through FIPS 140-2 Level 3 validated cloud HSM service certifications, key usage audit trails generated by the HSM, and cryptographic proof that all signing operations occurred within the HSM boundary. The evidence type is different; the outcome being evidenced is the same. Whether a given auditor accepts cloud HSM evidence as equivalent to on-premises HSM evidence depends on their interpretive framework and risk tolerance—a variability that practitioners must navigate explicitly [2].

For auditability, the interpretive question is whether distributed cloud-native log infrastructure can produce records that satisfy the tamper-evidence and attributability expectations that compliance frameworks require. A cloud deployment that routes all operational events through a CT-style append-only log signed by an HSM-held key produces audit evidence that is, in a technical sense, more tamper-resistant than a traditional syslog-based audit trail stored in a filesystem that administrators have write access to. Auditors familiar with the CT model should be able to evaluate this evidence; auditors whose assessment frameworks are calibrated to traditional audit trail types may require guidance on how to interpret cryptographic log proofs [4].

For separation of duties, cloud IAM policies provide a technically precise and independently verifiable representation of access boundaries. A well-configured IAM policy that prevents a certificate requestor identity from performing issuance approval operations, enforced at the platform level rather than by procedural controls, arguably provides stronger separation of duties evidence than traditional role separation enforced through password policies and manual access reviews. Syed et al.'s survey of zero trust architectures noted that IAM-enforced access controls, when combined with continuous policy evaluation and audit logging, can provide security properties that meet or exceed those of traditional perimeter-based controls [10]. The compliance challenge is ensuring that auditors have frameworks to evaluate IAM policy configurations against separation of duties requirements.

The overall interpretive direction that emerges from this analysis is that compliance in cloud contexts is not less achievable than in traditional environments—it is differently evidenced. Cloud deployments produce compliance evidence that is often more precise, more cryptographically verifiable, and more continuously maintained than traditional evidence types. The challenge lies in the institutional translation layer: ensuring that auditors, root program operators, and standards bodies develop the interpretive frameworks needed to evaluate cloud-native evidence types against the outcome-based expectations that the underlying compliance principles express. This is primarily a governance and standardization challenge rather than a technical one, and it suggests that the evolution of cloud-adaptable TLS issuance architectures will require parallel evolution in compliance evaluation frameworks.

7. Trade-offs and Open Questions

These alternatives indicate that cloud-centric TLS certificate issuance solutions are technically feasible, may meet existing compliance requirements, and have trade-offs and meaningful questions that must be understood before deciding if such an approach is the right way to go.

The most significant trade-off concerns architectural complexity. A traditional, tightly controlled CA environment achieves compliance demonstration in part through simplicity: a small, well-defined infrastructure with limited dependencies and a clear

administrative boundary is easier to audit than a distributed, multi-layer system with components spanning cloud services, HSM integrations, TEE-enabled workloads, and policy-driven control planes. The layered composable trust model described in Section 5.5 requires auditors to evaluate multiple independent mechanisms, understand their interactions, and assess the composite evidence chain. This complexity increases audit cost, introduces more potential points of failure, and demands a level of auditor sophistication that is not uniformly available. Operators considering cloud-adaptable architectures must weigh the operational and scalability benefits against the compliance overhead that complexity introduces [7][8].

A second open question concerns the long-term trust relationship with cloud infrastructure providers. Cloud HSM services, TEE attestation chains, and IAM policy enforcement all depend on trust in the correctness and integrity of provider-managed hardware and software. For certificate issuance systems, this introduces a dependency that does not exist in operator-controlled environments: if a cloud provider's HSM firmware has an undisclosed vulnerability, or if a provider's TEE attestation service produces incorrect measurements, the security properties that the operator's compliance evidence relies upon may not hold. Muñoz et al.'s comprehensive survey of TEE security found that hardware-based trusted execution environments, while providing strong isolation guarantees, have been subject to side-channel attacks and implementation vulnerabilities that could affect the integrity of their security boundaries [7]. Operators must assess their tolerance for this provider dependency risk and consider architectural mitigations such as multi-provider redundancy or hybrid on-premises/cloud deployments for the most sensitive components.

Governance variability represents a third significant challenge. Different root program operators—Microsoft, Mozilla, Apple, and Google each maintain independent root programs—may apply different interpretive frameworks to cloud-native compliance evidence. A CA that successfully demonstrates compliance with one root program's audit expectations for a cloud-deployed architecture may face different requirements from another program, creating a fragmented compliance landscape that increases operational burden. Without industry-wide alignment — through the CA/Browser Forum, the Compliance Working Group, or equivalent bodies — on how cloud-native evidence types should be evaluated, operators face uncertainty about whether their architectural choices will achieve consistent acceptance across the programs on which their business depends [2].

Trade-off / Open Question	Description	Risk Level	Mitigation Direction
Architectural complexity	A multi-layer evidence chain is harder to audit than a simple controlled environment	Medium	Modular architecture with independently auditable components; clear audit scope documentation
Provider trust dependency	HSM/TEE integrity depends on provider hardware and firmware correctness	Medium–High	Multi-provider redundancy for key operations; hybrid on-premises/cloud for highest-sensitivity functions
Governance variability	Root programs may apply different standards to cloud-native compliance evidence.	High	CA/Browser Forum alignment; pre-audit engagement with root program operators
Audit framework maturity	Auditors lack established templates for evaluating TEE attestation and IAM-based controls.	Medium	Industry working groups; publication of reference audit frameworks for cloud-native CA evidence
Operational determinism	Cloud infrastructure abstraction reduces direct visibility into underlying system behavior	Low–Medium	Immutable infrastructure patterns; attestation-based state verification

Table 4. Trade-offs and open questions in cloud-adaptable TLS certificate issuance architectures, risk assessment, and mitigation directions.

8. Future Considerations

The evolution of cloud-adaptable certificate issuance architectures will depend not only on technical innovation but also on the parallel development of governance frameworks, audit methodologies, and industry consensus that can evaluate these architectures against established trust expectations. Several directions merit particular attention in this regard.

First, the formalization of guidance around verifiable infrastructure primitives is an important near-term need. TEE attestation, cloud HSM audit trails, and cryptographic log integrity mechanisms have each been developed and deployed in specific contexts — confidential computing, cloud key management, and certificate transparency, respectively — but have not been integrated into a unified framework for CA compliance evaluation. Developing such a framework, potentially through the CA/Browser Forum's existing working group structure, would reduce the interpretive variability that currently creates uncertainty for operators considering cloud-adaptable architectures. Analogous standardization efforts in adjacent domains—such as the National Institute of Standards and Technology (NIST)'s work on zero trust architecture in NIST SP 800-207—provide useful templates for how technical guidance can be formalized and adopted into compliance practice [9].

Second, the development of enhanced audit methodologies for distributed and layered architectures is necessary. Current WebTrust and ETSI audit frameworks were developed for relatively centralized CA environments and may not provide adequate coverage for the distributed, multi-component architectures that cloud deployments introduce. Audit methodologies that can assess the integrity of cryptographic log chains, evaluate TEE attestation evidence, and validate IAM policy configurations against separation of duties requirements would meaningfully reduce the compliance uncertainty that currently limits operator willingness to explore cloud-adaptable architectures. The broader convergence of the public PKI ecosystem toward shorter certificate lifetimes, automated issuance, and continuous validation — reflected in CA/Browser Forum ballot developments — creates a favorable context for updating audit frameworks to accommodate the operational characteristics of cloud-native systems [2].

Finally, continued ecosystem dialogue between operators, auditors, root program owners, and standards bodies will be essential for navigating the transition from traditional to cloud-adaptable deployment models. The trust ecosystem for public TLS certificates is fundamentally a collaborative governance system — its integrity depends on shared standards, mutual accountability, and continuous evolution in response to new operational realities. The questions raised in this article about how cloud-native evidence types map to outcome-based compliance expectations are ultimately questions for the ecosystem to resolve through collective interpretive work, informed by technical analysis of the kind this article attempts to contribute.

9. Conclusion

Public TLS certificate issuance systems occupy a position of foundational importance in internet security, and their compliance with established trust and audit expectations is a prerequisite for maintaining the integrity of the certificate ecosystem on which secure communications depend. This article has explored the question of whether and how these systems might evolve toward cloud-adaptable architectures without sacrificing the assurances that compliance frameworks require. The article rests on a foundational interpretive principle: that compliance requirements for public TLS issuance are expressed as outcome-based expectations rather than infrastructure mandates and that the question of whether a deployment model satisfies them depends on whether it can produce credible evidence of the required outcomes rather than on whether it resembles traditional deployment patterns.

Five architectural directions — cryptographic isolation via externalized HSMs, verifiable execution environments, append-only cryptographic audit logs, policy-driven IAM control planes, and layered composable trust models — emerge from this analysis as technically credible approaches to establishing the required assurances within cloud deployment contexts. Each direction addresses specific compliance principles, and their combination in a layered model provides the broadest coverage of the compliance framework. The trade-offs introduced by these approaches —

primarily increased architectural complexity, dependence on provider trust primitives, and governance variability across root programs — are real and must be addressed through careful design, pre-audit engagement, and industry alignment on cloud-native evidence evaluation frameworks.

The broader contribution of this analysis is a reframing of the cloud adoption question for certificate issuance systems: it is not a question of whether cloud environments are secure enough but of whether they can produce evidence of assurance that satisfies the institutional frameworks through which compliance is evaluated. Technical capabilities have advanced to the point where the mechanisms needed for cloud-native assurance exist and are deployable. The remaining work is largely institutional — developing the audit frameworks, governance guidelines, and industry consensus that enable those mechanisms to be evaluated with the confidence and consistency that the public trust ecosystem requires.

References

[1] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," Internet Engineering Task Force, RFC 5280, May 2008. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5280.html

[2] CA/Browser Forum, "Baseline Requirements for TLS Server Certificates." [Online]. Available: https://cabforum.org/working-groups/server/baseline-requirements/documents/

[3] B. Laurie, A. Langley, and E. Kasper, "Certificate transparency," Internet Engineering Task Force, RFC 6962, Jun. 2013. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6962.html

[4] B. Laurie, E. Messeri, and R. Stradling, "Certificate transparency version 2.0," Internet Engineering Task Force, RFC 9162, Dec. 2021. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9162.html

[5] M. Luo, B. Feng, L. Lu, E. Kirda, and K. Ren, "On the complexity of the web's PKI: Evaluating certificate validation of mobile browsers," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 6, pp. 4747–4762, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10066507

[6] H. Hadan, N. Serrano, and L. J. Camp, "A holistic analysis of web-based public key infrastructure failures: comparing experts' perceptions and real-world incidents," Journal of Cybersecurity, vol. 7, no. 1, p. tyab025, 2021. [Online]. Available: https://academic.oup.com/cybersecurity/article/7/1/tyab025/6470936

[7] A. Muñoz, R. Ríos, R. Román, and J. Lopez, "A survey on the (in)security of trusted execution environments," Computers & Security, vol. 129, p. 103180, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823000901

[8] J. Ménétrey, M. Pasin, P. Felber, and V. Schiavoni, "Attestation mechanisms for trusted execution environments demystified," in Distributed Applications and Interoperable Systems, Springer, Cham, 2022, pp. 95–113. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-16092-9_7

[9] P. Phiayura and S. Teerakanok, "A comprehensive framework for migrating to zero trust architecture," IEEE Access, vol. 11, pp. 19487–19511, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10052642

[10] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "Zero trust architecture (ZTA): A comprehensive survey," IEEE Access, vol. 10, pp. 57143–57179, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9773102

[11] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," National Institute of Standards and Technology, NIST Special Publication 800-207, Aug. 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf

[12] Hrishikesh Joshi, "Emerging technologies driving zero trust maturity across industries," IEEE Open Journal of the Computer Society, vol. 6, pp. 25–40, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10764723

[13] Yacine Felk, "Confidential computing," in Trends in Data Protection and Encryption Technologies, Springer, Cham, 2023, pp. 103–108.

[Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-33386-6_19

[14] M. Sommerhalder, "Trusted execution environment," in Trends in Data Protection and Encryption Technologies, Springer, Cham, 2023, pp. 97–102. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-33386-6_18

[15] R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten, "Automatic certificate management environment (ACME)," RFC 8555, Mar. 2019. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8555.html