Controller Validity as a Renewable Security Property

A Personal Technical Perspective on gTLD Integrations with Alternative Naming Systems

Nguyen Ngoc Duy | Individual capacity | 16 August 2026

Position in brief: I support the TSG's string+controller model, but I believe controller evidence should never be treated as permanently valid. For every integrated name, the operator should be able to demonstrate that the evidence proving the same controller is still current. I recommend a time-bounded, renewable controller attestation, supported by event-triggered revalidation and fail-closed behavior when the evidence becomes stale or cannot be revalidated.

1. My Position

The TSG's core requirement is compelling: the same string should remain under the control of the same party across the global DNS and any integrated alternative naming system. The Initial Report states that this condition must be reliably satisfied at all times and that a proposed operator must demonstrate that the integration cannot be broken under deployment conditions without the controller's instruction or approval.[1] I agree with that security objective.

My concern is specifically about time. A controller relationship can be correct when an integration is created and become incorrect later. Domain names expire, are re-registered, change registrant, move between registrars, enter administrative states, and recover from operational failures. The technical question is therefore not only whether the integration can prove control once, but whether the proof used to authorize continued integration remains sufficiently current throughout the life of the name.

2. Why I Believe Controller Proof Needs an Explicit Lifetime

I do not think synchronization alone fully answers this question. Two systems can be synchronized with each other and still be synchronized to an obsolete controller assertion. If both systems continue to rely on a token, identifier, snapshot, or mapping that was valid yesterday but no longer reflects the authoritative control relationship today, the systems may appear internally consistent while representing the wrong controller.

This is not merely hypothetical. The current IETF DNSOP integration draft treats domain-name lifecycle and domain-control validation as distinct considerations. It notes that expiration or other lifecycle changes can alter control and require action to remain synchronized. In its ENS case study, it also describes how an old positive proof can persist until a newer proof replaces it, illustrating the broader danger of stale assertions in an integrated environment.[2]

Concrete failure scenario: Registrant A establishes example.tld in the DNS and an alternative naming system, and the controller linkage is verified. Later, the DNS name expires and is legitimately re-registered by B. If the alternative system continues to accept the original proof, token, or controller identifier without revalidation, A may still be represented as the controller outside the DNS even though B now controls the DNS name. The integration has not necessarily "crashed"; rather, its trust statement has aged out of reality.

3. My Proposal: Time-Bounded Controller Attestation

I recommend that the final framework treat proof of controller as a renewable security assertion. I use the term time-bounded controller attestation, or integration lease, to describe the requirement. The term is not intended to prescribe a particular protocol or data structure. It describes an operational property: evidence that authorizes a string+controller linkage should have a defined validity period and should stop authorizing continued operation when its validity can no longer be demonstrated.

The practical purpose of a lease is simple. It forces the system to answer, repeatedly and in an auditable way, a question that matters to users: is the party represented as the controller still entitled to be represented as the controller now?

4. Minimum Operational Requirements

- **Maximum attestation age.** Each integration should state the maximum period for which controller evidence may be relied upon without successful revalidation. The period should be measurable and auditable. I would not prescribe one universal duration at this stage; the appropriate interval may depend on the proof mechanism and how quickly authoritative control can change.
- **Event-triggered revalidation.** Revalidation should occur immediately after events that can change or materially affect the control relationship, including registrant change, re-registration after expiry, a controller-identifier or key rotation, relevant transfer workflows, and recovery from a partition or outage where control could not be checked.
- **Authoritative proof lineage.** The operator should record what source established control, when it was checked, what evidence was accepted, and which authoritative state the evidence represented. This makes a later audit capable of distinguishing a current proof from a merely consistent copy of an old one.
- **Fail-closed behavior.** If the attestation expires and revalidation cannot complete, the alternative-system representation should not continue indefinitely as if control were still verified. The affected alt-name should be withheld, disabled, or moved to another non-authorizing state until fresh proof is available.
- **Fresh recovery.** Restoration after a failed or expired attestation should require new evidence. Returning connectivity should not, by itself, reactivate the previous controller binding.
- **Operational metrics.** Operators should be able to report expired attestations, failed revalidations, the duration of stale windows, and incidents in which a change of controller was detected after an existing proof had become outdated. These metrics would make the requirement testable after launch, not only at approval time.

5. Why This Matters for Security and Stability

The security value of string+controller integration comes from the reliability of the controller relationship, not only from string equality. If the proof of that relationship has no meaningful lifetime, a system can preserve the appearance of consistency while the underlying authority has changed. That creates a path to user confusion, impersonation, misdirected transactions, and conflicting identity claims even if the technical components continue to operate normally.

Recent measurement research on DNS integrations provides a useful warning. Valapu et al. studied what they call zombie linkages: integrations that retain an old linkage after DNS ownership has changed or lapsed. Their measurements cover Web PKI, ENS, and Maven Central rather than the TSG's proposed registry-service model, so I do not treat their prevalence estimates as directly transferable. I do, however, consider the failure class relevant: designs that validate control once and then allow the linkage to persist can accumulate stale authority, whereas designs with expiration or repeated validation limit that risk.[3]

6. Architecture-Neutral Implementation

I do not believe this requirement should force ICANN to select one technical architecture. An SRS-centered design could derive the attestation from registry state; a distributed USoT design could use signed or cryptographic evidence; another design might use a short-lived proof tied to an authoritative controller identifier. The implementation may differ, but the security outcome should be the same: no active integration should rely indefinitely on controller evidence whose current validity cannot be demonstrated.

A risk-sensitive approach is also possible. Integrations with strong revocation signals and authoritative real-time state may support longer or continuously refreshed attestations, while systems with delayed settlement, weaker revocation, or external proof dependencies may need shorter validity windows. I would prefer measurable assurance requirements over a single fixed timeout chosen without operational evidence.

7. Proposed Evaluation Question

For every active integrated name, can the operator demonstrate not only that the controller is the same across systems, but that the evidence supporting that conclusion is current, auditable, and guaranteed to stop authorizing the integration if it becomes stale?

8. Conclusion

I support continued development of the TSG framework and the string+controller approach. My recommendation is to make controller validity explicitly temporal. A name should not remain integrated merely because control was established correctly at some point in the past. The integration should periodically or event-by-event renew the evidence on which that trust depends.

In my view, a time-bounded controller attestation would strengthen the TSG's model without making it unnecessarily prescriptive. It turns 'same controller' from a historical fact into a continuously renewable security property, while giving evaluators an objective way to test whether the integration remains trustworthy throughout the lifecycle of a name.

References

[1] ICANN, Report of the Technical Study Group: gTLD Integrations with Alternative Naming Systems, Initial Report, draft dated 10 August 2026. Relevant sections include 4 (Basic Elements of String+Controller Integration), 6.3 (Proving Control), 8 (Operational Scenarios), and 9 (Technical and Practical Feasibility).

[2] S. Sheth, A. Kaizer, B. Newbold, N. Johnson, Integration of DNS Domain Names into Application Environments: Motivations and Considerations, draft-ietf-dnsop-integration-04, IETF DNSOP, 24 July 2026. Work in progress. Relevant sections include Domain Name Lifecycle, Domain Control Validation, Synchronization Mechanisms, and the ENS case study.

[3] S. T. Valapu, J. Heidemann, M. Jonker, R. Sommese, Zombies in Alternate Realities: The Afterlife of Domain Names in DNS Integrations, arXiv:2605.06880, 2026. The study measures stale linkages in Web PKI, ENS, and Maven Central and compares validate-once, expiring, and repeated-validation designs.

Research note: The empirical study in Reference [3] evaluates broader DNS integrations, not ICANN-approved gTLD/alternative-naming registry services. It is used here to support the general stale-authorization failure model, not to estimate the expected frequency of failures under the TSG framework.