Internal vs External Certificates:
Strategy Guide

One of the most consequential early decisions in any certificate strategy is a deceptively simple one: which certificates should come from a public CA, and which should come from an internal private CA instead. Getting this wrong in either direction creates real problems, unnecessary cost and complexity on one side, or unnecessary risk and compatibility headaches on the other. This article lays out a practical framework for making that call correctly and consistently.

The Core Question Behind the Decision

The deciding factor is almost always who or what needs to trust the certificate. If the relying parties are entirely within an organization's own control, every client that will ever connect to this service is a system the organization itself manages, an internal CA is generally the better fit. If the relying parties include the general public, external partners, or any system outside the organization's direct control, a public CA is necessary, since there is no practical way to distribute a private root certificate to every browser and device on the internet.

When Public Certificates Are the Right Call

Any customer-facing website, public API, or service accessed by external partners without a pre-established trust relationship needs a publicly trusted certificate, full stop. This category also includes services where an organization wants the specific assurance and warranty protections that come with a public CA relationship, even for services that happen to have a limited audience today but could reasonably expand to a broader one later.

When Internal Certificates Are the Better Fit

Internal APIs, service-to-service communication within a private network, VPN client authentication, internal development and staging environments, and machine identities for containers and AI agents operating entirely within an organization's own infrastructure are all strong candidates for internal certificates. Using a public CA for these purposes adds unnecessary cost, unnecessary exposure through Certificate Transparency logs revealing internal naming conventions, discussed elsewhere in this series, and unnecessary dependency on external issuance infrastructure for traffic that never actually leaves the organization's own control.

The Hybrid Reality Most Organizations Actually Live In

Very few organizations of any meaningful size run purely public or purely internal certificate strategies; most operate a genuine hybrid, with public certificates covering external-facing services and an internal CA covering everything behind that boundary. Managing this hybrid well requires certificate inventory and monitoring tooling capable of tracking both categories consistently, discussed throughout this series, rather than treating internal and external certificates as entirely separate management problems handled by different teams with no shared visibility.

Edge Cases Worth Thinking Through Deliberately

Some situations sit less clearly on one side of the line: a service used only by a small number of trusted external partners might reasonably use either a public certificate or a carefully managed cross-trust arrangement with an internal CA, depending on how much control the organization has over those partners' own trust configurations. Similarly, a service that starts as purely internal but has a realistic near-term path to external exposure may be better served starting with a public certificate from the outset, avoiding a disruptive migration later once external exposure actually happens.

Documenting the Decision Framework, Not Just Making the Call Once

Organizations benefit considerably from documenting a clear, written policy for this decision, rather than relying on individual engineers making the internal-versus-external call inconsistently on a case-by-case basis. A documented framework, reviewed periodically as the organization's infrastructure and external relationships evolve, keeps this decision consistent across teams and prevents the kind of drift where similar services end up on opposite sides of the internal-external line simply due to whichever engineer happened to set each one up.

Applying This Framework to AI Infrastructure

AI infrastructure often blurs the internal-external line in new ways: a model-serving endpoint might start purely internal, then later be exposed to external partners or customers through an API product, and an AI agent operating internally today might eventually need to call external services directly on an organization's behalf. Applying the same deliberate, documented decision framework to AI infrastructure certificates from the start, rather than defaulting to whichever option was fastest to set up during initial development, positions an organization to handle these transitions cleanly as AI initiatives mature and their external exposure evolves.

The Countdown Is Already Running: 200 Days, 100 Days, 47 Days

Every certificate conversation in 2026 eventually arrives at the same clock, and it is worth closing on it here. The CA/Browser Forum's Ballot SC-081v3 is not a proposal under discussion; it is an approved, already-in-motion schedule. Maximum public TLS certificate lifetimes fall from 398 days to 200 days on March 15, 2026. They fall again to 100 days on March 15, 2027. By March 15, 2029, they drop to just 47 days, with domain validation itself needing to be re-proven roughly every 10 days.

01

Translate that into operational terms and the picture gets stark quickly. An organization currently renewing certificates a few times a year will be handling renewal events on the order of every couple of weeks by the end of this countdown, across every endpoint it operates. Manual tracking, calendar reminders, and a spreadsheet somebody checks once a month will not survive contact with that cadence. What has always been an occasional chore is becoming a continuous, automated operation, whether an organization plans for it or not.

02

Whichever side of this decision a given certificate falls on, public certificates remain fully subject to the shrinking lifetime schedule below, which is itself a meaningful factor in the decision: services that can reasonably stay internal avoid that specific pressure entirely, while anything genuinely public needs full automation in place well ahead of the 47-day milestone.

03

The 200-day, 100-day, and 47-day milestones are not distant hypotheticals; the first has already arrived. Organizations that build the automation loop now, generating keys, vaulting them securely, brokering issuance across Certificate Authorities through APIs, and rebinding certificates to live endpoints without manual intervention, will meet each deadline without disruption. Organizations that wait will be rebuilding their certificate operations under deadline pressure, with far less room for error and far less time to get it right. The countdown is the call to action. The only real decision left is whether to automate on your own schedule, or on the CA/Browser Forum's.