

AUTOMATION · 2026-09

Connecting an agent starts with proving its key

The September Mesh enrollment work offers a practical lesson in identity, offline handoffs, and clear trust boundaries.



By Lyra

Published Oct 14, 2026 · 4 min read

September 2026 retrospective · Story context 2026-09-30



When a device introduces itself to a control application, a friendly name is useful for the person looking at the screen. The software needs a stronger way to decide which identity it is actually talking to.

September's Mesh work added individual signed gateway enrollment and an offline agent picker. Related CodeSeal work introduced identity issuance with proof of key possession. That gives us a practical way to talk about a common problem: two devices can use the same friendly name, so what tells the software which one is connecting?

The [public Mesh commit](#) shows the implementation. Here's the part to carry into your own connection workflow.

| Separate display names from identity

Two devices can both be named “workstation.” A person can rename an agent. Neither action should silently change the cryptographic identity associated with an enrollment.

Keep a stable identity identifier and its key information separate from the editable display label. Show enough context for the owner to select the intended device, especially when importing prepared identity material offline.

A picker is part of that boundary. It should help people distinguish the available identities before connecting. A silent default can be convenient until it chooses the wrong environment or account.

| Ask the key holder to prove possession

A system that accepts a public key needs confidence that the connecting party controls the corresponding private key. A challenge-and-response design can supply that evidence: the service issues a fresh challenge, and the client signs the agreed challenge payload.

The verifier checks the signature using the supplied public key and validates the surrounding protocol fields. The freshness, intended audience, expiration, and association with the enrollment attempt matter. A valid signature over an old or unrelated message doesn't establish approval for this new connection.

This is a protocol design sketch. Production enrollment should use a reviewed implementation with explicit replay protection and error handling. Cryptographic primitives are only one component of the workflow.

| Make the offline handoff understandable

Offline preparation can be useful when the local device and the hosted control surface are configured at different times. The handoff package should have a clear purpose and a limited set of fields.

Before import, tell the owner what identity is being added, which issuer is involved, and which claims will be trusted. After import, show whether the record is merely stored, cryptographically checked, or actually connected to a live device.

Those states can easily collapse into one green badge. Show them separately so a person can tell whether they have imported a record or actually connected a device.

For a small implementation, sketch the transition sequence first:

prepared -> imported -> verified -> approved -> connected

Some products combine steps, but the evidence behind each state still deserves a name. A later disconnect should change connection status without rewriting the historical enrollment record.

| Test confusion as well as failure

Create two synthetic identities with similar labels. Import both, rename one, and confirm the stable identifiers remain distinguishable. Try an expired challenge, a signature from the wrong key, and a valid record with the wrong intended audience.

Then test the human recovery path. If enrollment fails, can the owner inspect the reason and retry without deleting unrelated identities? Does the picker preserve a deliberate selection? Can a disconnected device be recognized without pretending it is online?

These tests exercise the boundary a user actually crosses. They are especially useful for local AI tooling, where a hosted interface and an on-device runtime can have independent lifecycles.

| Keep connection claims precise

A signed identity supports a particular trust decision. Task quality, tool availability, and local environment health each need their own checks.

Expose those capabilities separately and verify them through the workflows that need them. The September Mesh work is a good reminder that clear identity and clear status belong together. A person should be able to tell who is connecting, what has been checked, and what still needs to happen before work can begin.

| Sources & further reading

Checked 2026-10-14 · Story context 2026-09-30

1. [Mesh signed enrollment implementation](#) (2026-10-14)
2. [MDN signature verification](#) (2026-10-14)

[Read the current article online →](#)

Support the workshop

Explore software and learning offers at [StackSocial](#). Affiliate link: ProgreTech may earn a commission from purchases. Check current price, license and renewal terms before buying.

Find our projects at progretech.com.

ProgreTech LLC · Build / Learn / Improve / Together

This file contains its text, illustrations and fonts. No account, scripts or network connection required to read. External reference links need internet access.