DNSSEC-anchored discovery path for 802.1AR DevID named endpoints

A small, optional idea, with its limits up front

Kaveh Ranjbar, Whisper Security*
kaveh@whisper.security
IEEE 802.1 Security TG electronic meeting, 2 November 2026

* Whisper Security, viaGraph B.V.

Where I think this helps, and where I do not

Where it helps

- On the named side of enrollment, the registrar, the MASA and the EST server, which all have DNS names already.
- As an online complement, for operators who want to check each other's keys across organisations without exchanging trust anchors by hand.
- Optional and informative, and built only from work the IETF has published or has in final review, nothing new on the wire for 802.1AR.

Where it does not

- It does not replace offline IDevID validation, which is offline-capable on purpose.
- It does not change the DevID certificate, or the IDevID and LDevID model.
- The pledge keeps its manufacturer trust anchors, binding DANE to the device during bootstrap would be the wrong place.
- And I am not asking for an amendment, I would like to know first whether this is useful to anyone here.

Terms as in RFC 8995: IDevID "installed by the vendor on new equipment", LDevID "installed by the owner of the equipment", both "a term from 802.1AR".

What 802.1AR and BRSKI leave to each deployment

BRSKI (RFC 8995) builds enrollment on the 802.1AR IDevID, and how the named parties check one another is left to configuration:

> **Registrar to MASA.** "The mechanisms in [RFC6125] SHOULD be used in authentication of the MASA using a DNS-ID that matches that which is found in the IDevID." "Registrars SHOULD permit trust anchors to be preconfigured on a per-vendor (MASA) basis."
>
> RFC 8995, Section 5.4

> **MASA to registrar.** "It is expected that the setup and configuration of per-customer Client Certificates is done as part of a sales ordering process."
>
> RFC 8995, Section 5.4.1

> **EST.** "The client validates the TLS server certificate using the EST client Explicit and, if enabled, Implicit TA database(s)."
>
> RFC 7030, Section 3.3.1

So across organisations, each pair of parties ends up exchanging anchors or certificates by hand, which works, but does not scale very well.

BRSKI already points at DANE

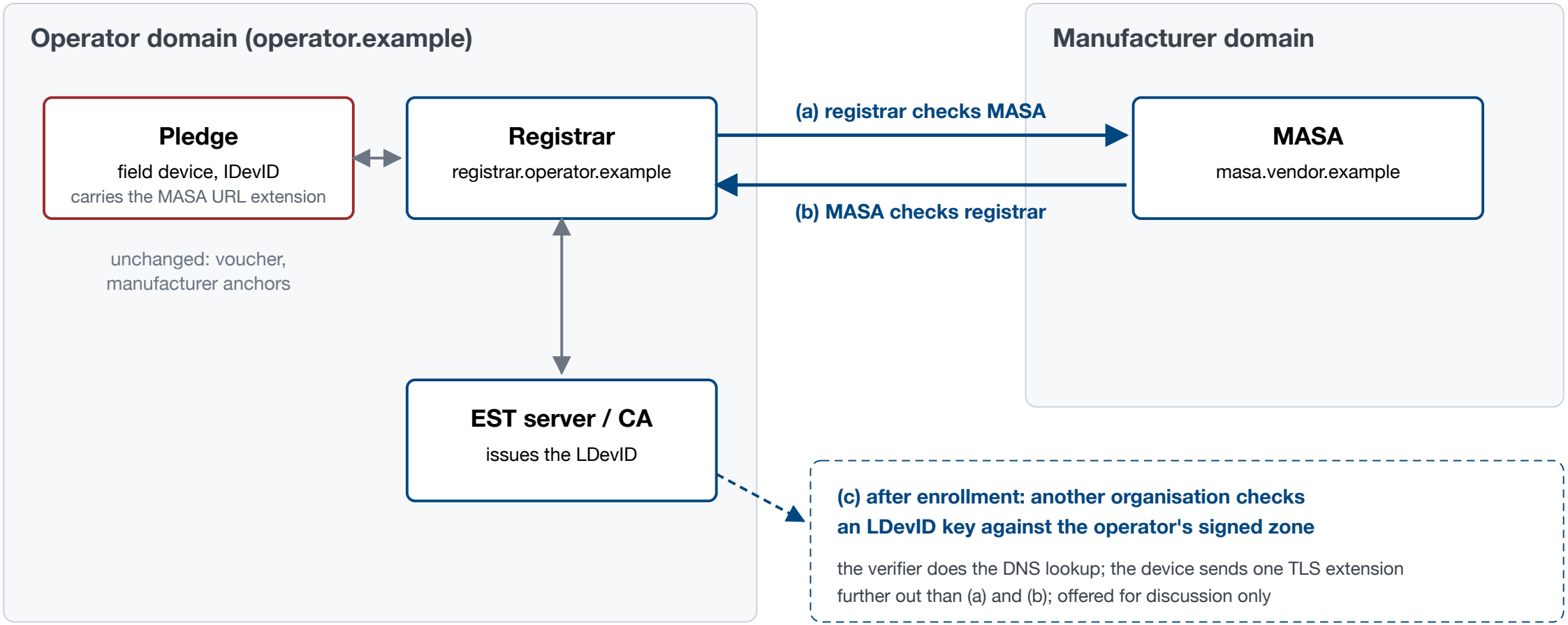
In Section 5.4.1, RFC 8995 lists the ways a MASA can identify a customer's registrar, and DANE is one of them:

- "The use of public PKI (i.e., WebPKI) end-entity certificates to identify the registrar is reasonable, and if done universally, this would permit a MASA to identify a customer's registrar simply by a Fully Qualified Domain Name (FQDN)."
- **"The use of DANE records in DNSSEC-signed zones would also permit use of a FQDN to identify customer registrars."**
- "A third (and simplest, but least flexible) mechanism would be for the MASA to simply store the registrar's certificate pinned in a database."

And: "This document does not make a specific recommendation on how the MASA authenticates the registrar as there are likely different tradeoffs in different environments and product values."

The piece that was missing is the client side, since the DANE RFCs cover servers and in BRSKI the registrar is the TLS client to the MASA. draft-ietf-dance-client-auth now fills that in: "This document describes how to use the TLSA record to publish client certificates or public keys", with a TLS extension, "dane_clientid", that carries the client's name.

Who checks whom



The record I have in mind, TLSA 3 1 1

```
_443._tcp.masa.vendor.example. IN TLSA 3 1 1 <SHA-256 of the SubjectPublicKeyInfo>
```

Field	What RFC 7671 says
3, DANE-EE	"the binding of the server public key to its name is based entirely on the TLSA record association" (Section 5.1)
1, SPKI	"the resulting TLSA record need not change across certificate renewals with the same key" (Section 5.1)
1, SHA2-256	"all DANE implementations are required to support SHA2-256" (Section 5.1)
Lifetime	with DANE-EE "the expiration date of the server certificate MUST be ignored. The validity period of the TLSA record key binding is determined by the validity period of the TLSA record DNSSEC signatures." (Section 5.1)
Recommendation	"it is RECOMMENDED that at most sites TLSA records published for DANE servers be "DANE-EE(3) SPKI(1) SHA2-256(1)" records." (Section 5.1)

It binds the key and not the certificate, so it survives renewal, and it is validated with DNSSEC from the root, with no certification authority in the path. Since selector 1 hashes the SubjectPublicKeyInfo as encoded, the record looks the same whatever the key algorithm, which may be of interest with P802.1AReg and ML-DSA, though each TLS stack still has to support the algorithm itself.

Three places it could help

(a) The registrar checks the MASA

The registrar already gets the MASA's name from the IDevID (the MASA URI extension, id-pe-masa-url, RFC 8995 Section 2.3.2), so it can look up a TLSA record for that name, next to the RFC 6125 name checks and per-vendor anchors that Section 5.4 already has.

```
_443._tcp.masa.vendor.example. IN TLSA 3 1 1 <SHA-256 of the MASA's SPKI>
```

(b) The MASA checks the registrar

Here the registrar is the TLS client, it sends its name in "dane_clientid" and the MASA looks the key up, which is the DANE route RFC 8995 Section 5.4.1 mentions. With DANE-EE "the client's name need not be present in the certificate at all" (draft Section 5).

```
registrar.operator.example. IN TLSA 3 1 1 <SHA-256 of the registrar's SPKI>     (draft Format 3, freeform)
```

(c) Cross-domain LDevID, a bit further out

An operator could publish the device keys it has enrolled under its own signed zone, and the draft's own example for this format, in Section 3.2, is titled "DevID". The verifier does the lookup, but the device's TLS client would need to send the extension, so I offer this one only for discussion.

```
sensor7._device.operator.example. IN TLSA 3 1 1 <SHA-256 of the LDevID SPKI>   (draft Format 2, device identity)
```

Something you can check with dig today

This is a per-agent identity record from one of our production zones, the same record type as on slide 6, though not a BRSKI deployment. I captured it on 30 October 2026 at 05:38 UTC.

```
$ delv _443._tcp.ae3b051ff3bf7f478.tdc38e7c55bad3306a92b830f9bb1e4f9.agents.whisper.online TLSA
; fully validated
_443._tcp.ae3b051ff3bf7f478.tdc38e7c55bad3306a92b830f9bb1e4f9.agents.whisper.online. 60 IN TLSA 3 1 1 9EC1EF18A1F15E5480A0FC3C8D6
E9690B2CCF1A7FA7146201940F28C 422BB47D
_443._tcp.ae3b051ff3bf7f478.tdc38e7c55bad3306a92b830f9bb1e4f9.agents.whisper.online. 60 IN RRSIG TLSA 13 7 60 20261031050000 2026
1029040000 54705 agents.whisper.online. RpBaCj9x...

$ dig +short agents.whisper.online DS
54705 13 2 99B88E7C1FD2A815F3DA0374D9FED2C4735F7EFDDB864CB1A79F9D55 43DFE06D
$ dig +short whisper.online DS
50855 13 2 3B33E9998425C1F43FCC3B8D4696DB587DB7CEA3F2A235E1556C4C4F 47E247D3
$ dig +short online DS
30961 13 2 9292C8B10406684B2E89209014BF4E14AC8D277EF5B6FC36122B0B58 78ACD47E
```

- The chain runs from the root to online, to whisper.online, to agents.whisper.online, whose key 54705 signs the TLSA record.
- The TTL is 60 seconds and the signature, algorithm 13 (ECDSA P-256 with SHA-256), runs from 29 October 04:00 to 31 October 05:00 UTC; we re-sign, so a later lookup will show newer dates.
- And the key itself, checked over TLS the same morning (05:47 UTC, OpenSSL 3.0.13): `openssl s_client -dane_tlsa_domain <name> -dane_tlsa_rrdata "3 1 1 9EC1...422BB47D"` returned "Verification: OK" and "DANE TLSA 3 1 1 ...fa7146201940f28c422bb47d matched EE certificate at depth 0". The certificate comes from our own private issuer, which DANE-EE does not look at.

What it costs

- **It needs to be online.** The verifier needs DNS and DNSSEC validation, done itself or through "a validating resolver to which it has a secure connection" (draft Section 8.2), which is exactly why I see it as a complement, and why I keep it away from the pledge.
- **If validation fails,** the draft's server "MUST either abort the connection with a handshake_failure TLS alert, or, if server policy allows, treat the client as unauthenticated" (Section 8.2), and for BRSKI parties the configured anchors are still there to fall back on.
- **Revocation has a limit.** Withdrawing a record takes effect as caches expire, 60 seconds in my example, but "DNSSEC-signed RRsets cannot be securely revoked before they expire" (RFC 7671 Section 11), so a replayed old record stays valid until its signature runs out, up to about 49 hours in my example. For high-value keys you want short signature lifetimes.
- **Key rollover** means publishing the new record beside the old one "at least two Times to Live (TTLs) or longer before the new chain is deployed" (RFC 7671 Section 8.1).
- **The zone becomes part of the trust,** so a registrar lock and care over who controls the DS record matter, as RFC 7671 Section 11 advises.
- **Privacy:** client names looked up in DNS can be seen, and the draft recommends encrypted DNS and query name minimisation (Section 12).

Where the pieces are today

Document	Status
RFC 6698, DANE TLSA	Proposed Standard, August 2012
RFC 7218, DANE acronyms	Proposed Standard, April 2014 (updates RFC 6698)
RFC 7671, DANE operations	Proposed Standard, October 2015 (updates RFC 6698)
RFC 8995, BRSKI	Proposed Standard, May 2021
draft-ietf-dance-client-auth-15	Published 27 October 2026. IESG Evaluation, on the agenda of the 5 November 2026 IESG telechat.

- OpenSSL's `s_client` verifies DANE with `-dane_tlsa_domain` and `-dane_tlsa_rrdata` (used on slide 8 with OpenSSL 3.0.13; also present in 3.5.9).
- We run DNSSEC-signed zones with per-agent DANE-EE 3 1 1 records in production, so I can bring implementation experience and not only a proposal.
- To be open about it, I reviewed draft-ietf-dance-client-auth and I am named in its acknowledgements, and Whisper Security builds identity products on DNSSEC and DANE.

What I would like to learn from you

- Does anyone here run, or plan to run, registrars, MASAs or EST servers that need to trust each other across organisations?
- Would the profile work, industrial, automotive, aerospace, and power outside 802, find a DNS-published key path useful for the named endpoints?
- If it is worth pursuing, where would it belong, a short informative note pointing at RFC 8995 Section 5.4.1 and the DANCE draft, or nowhere in 802.1, with the work staying in the IETF (ANIMA, DANCE)? You know that far better than I do.
- And is (c), an operator's own zone vouching for its LDevID keys, worth a second look, or out of scope?

I am glad to write this up properly if it is worth pursuing, and just as glad to let it rest if it is not. Thank you.

Kaveh Ranjbar, kaveh@whisper.security

References

- RFC 6698, The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA.
- RFC 7218, Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE).
- RFC 7671, The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance. Sections 5.1, 8.1, 11.
- RFC 8995, Bootstrapping Remote Secure Key Infrastructure (BRSKI). Sections 2.3.2, 5.4, 5.4.1.
- RFC 7030, Enrollment over Secure Transport. Section 3.3.1.
- RFC 8552, Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves.
- draft-ietf-dance-client-auth-15, TLS Client Authentication via DANE TLSA records (S. Huque, V. Dukhovni), 27 October 2026. Sections 2, 3.2, 5, 7, 8.2, 12.
- IEEE Std 802.1AR-2018, Secure Device Identity; P802.1AR-2018-Rev; P802.1AReg, Support for ML-DSA.