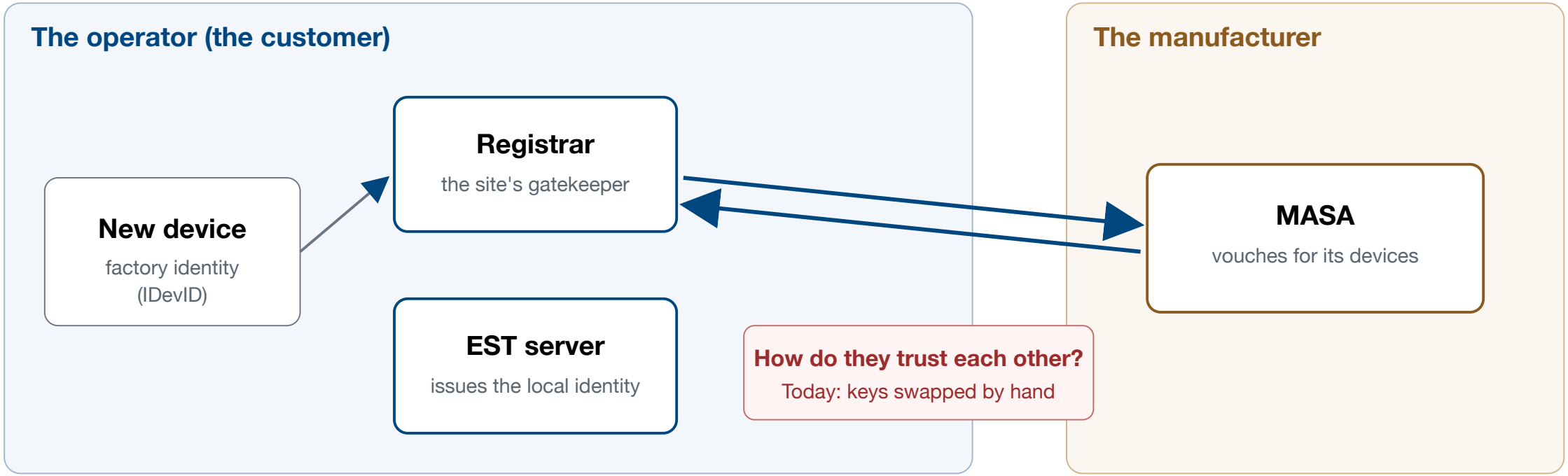
Checking each other's keys through DNS, for the named parts of 802.1AR enrollment

The same small, optional idea, in plain terms and pictures

Kaveh Ranjbar, Whisper Security (viaGraph B.V.)
kaveh@whisper.security
IEEE 802.1 Security TG electronic meeting, 2 November 2026

The situation today

A new device arrives at a site. Before it can join the network, three servers that all have names have to trust each other, and they often belong to different companies.

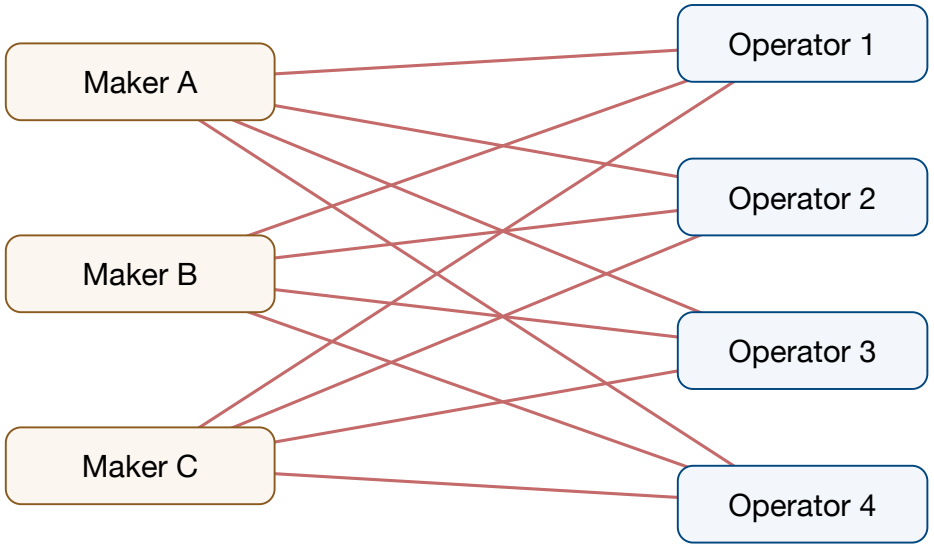


BRSKI (RFC 8995) leaves this to configuration: per-vendor trust anchors on the registrar (Section 5.4), customer certificates set up "as part of a sales ordering process" on the MASA (Section 5.4.1).

Why swapping keys by hand gets heavy

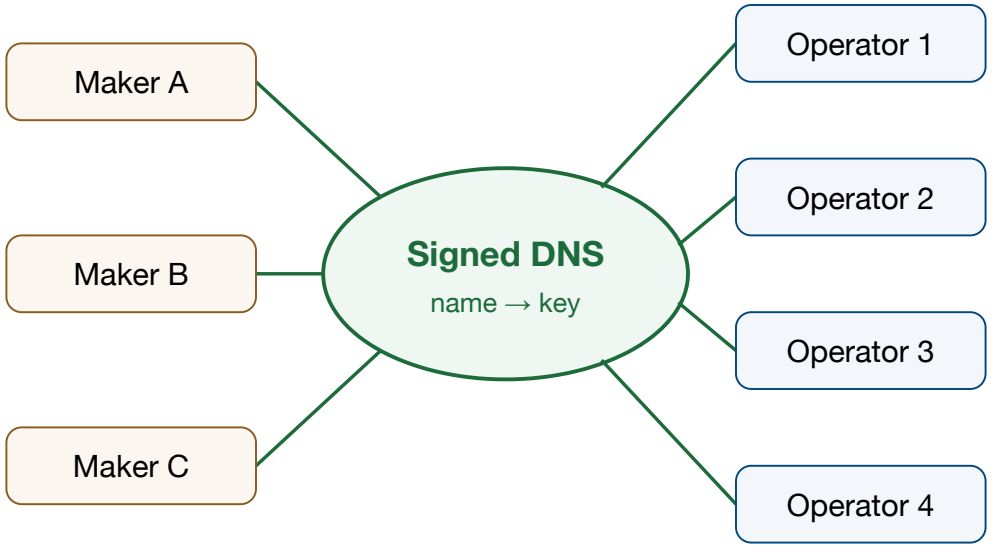
Take a simple example: three manufacturers and four operators who all work with each other.

By hand: every pair, in both directions



12 relationships, each set up and renewed by hand

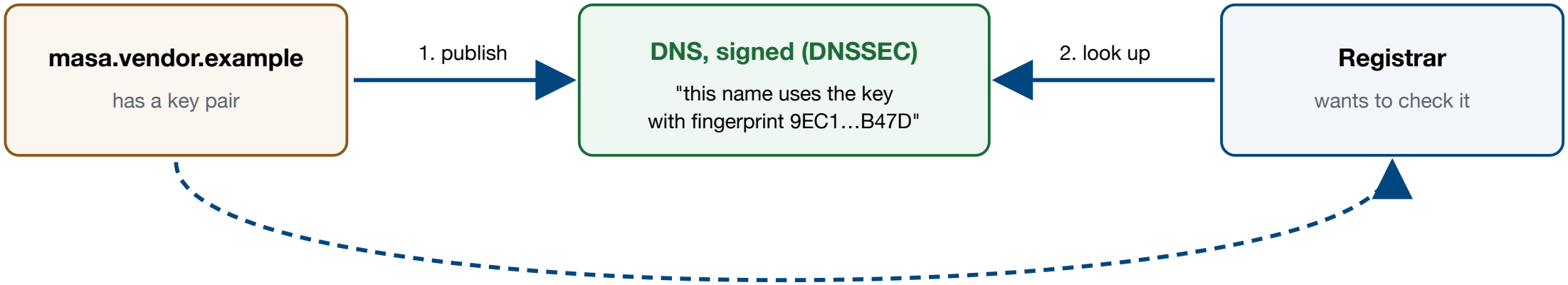
Through DNS: each one publishes once



7 records, each kept by its owner, looked up when needed

The idea in one sentence

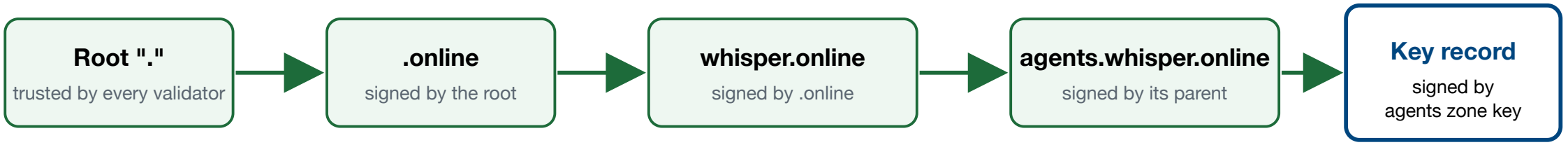
Each server publishes the fingerprint of its key under its own name in DNS, signs it with DNSSEC, and the other side looks it up and compares it during the TLS handshake.



Like a phone book where every entry carries a tamper-proof seal you can check yourself.

Why you can trust the lookup

DNSSEC is a chain of signatures from the root of DNS down to the name. If anyone changes the answer on the way, the check fails.



No certificate authority sits in the middle. The owner of the name is the one who vouches for the key.

This is the real chain behind the record on slide 9, from one of our production zones.

What the record looks like

One line in DNS. Only the three small numbers need explaining.

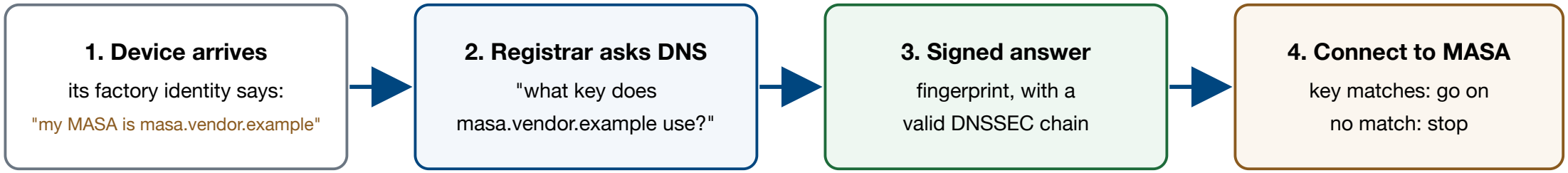
```
_443._tcp.masa.vendor.example.  IN  TLSA  3 1 1  <SHA-256 fingerprint of the key>
```

3	**Trust this key directly.** No certificate authority involved, and the certificate's own expiry date does not matter.
1	**Pin the key, not the certificate.** Renew the certificate with the same key and the record does not change.
1	**SHA-256 fingerprint.** Every DANE implementation must support it, and it looks the same whatever the key algorithm, ML-DSA included.

"3 1 1" is what RFC 7671 Section 5.1 recommends "at most sites".

Example A: the registrar checks the MASA

The device already carries its manufacturer's MASA address. The registrar just looks up that name.

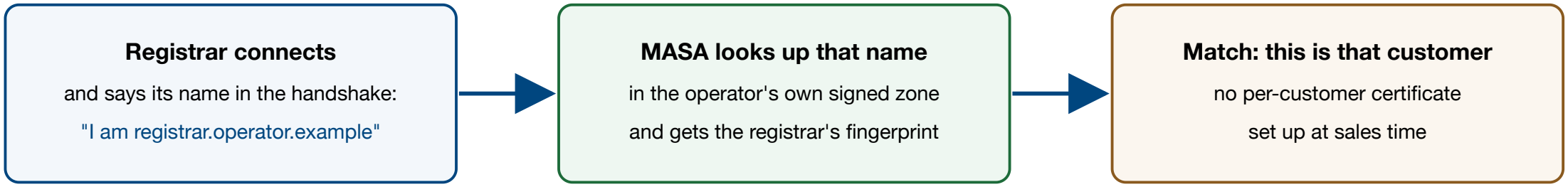


What is new: nothing on the device.

The MASA address is already in the IDevID (RFC 8995 Section 2.3.2). The DNS check sits next to the per-vendor anchors the registrar already has, it does not replace them.

Example B: the MASA checks the registrar

This is the direction that hurts today: a manufacturer has to know every customer's registrar. RFC 8995 already names DANE as one way to do it.



The missing piece was the client side, because the DANE RFCs cover servers. The IETF DANCE draft (draft-ietf-dance-client-auth) adds exactly that: one TLS extension, "dane_clientid", that carries the client's name. It is on the IESG telechat this Thursday, 5 November.

It works today: a real record

Not a BRSKI deployment, but the same record type, from one of our production zones. Anyone here can check it.

```
$ delv _443._tcp.ae3b051ff3bf7f478.tdc38e7c55bad3306a92b830f9bb1e4f9.agents.whisper.online
TLSA
; fully validated
... TLSA 3 1 1 9EC1EF18A1F15E5480A0FC3C8D6E9690B2CCF1A7FA7146201940F28C422BB47D
```

DNS side

"fully validated": the signature chain from the root down to this record checks out.

TLS side

OpenSSL connecting to the same name: "Verification: OK", "DANE TLSA 3 1 1 ... matched EE certificate".

Checked again on 2 November 2026. The certificate comes from our own private issuer, which "3" deliberately ignores.

What stays exactly as it is

Unchanged

- The device and its factory identity (IDevID)
- Offline validation, which is offline on purpose
- The DevID certificate format
- The vouchers and manufacturer anchors
- No new message for 802.1AR

Where it would sit

- Only between the named servers: registrar, MASA, EST
- As an extra, online check next to what they already do
- Optional, and only if people here find it useful

I am not asking for an amendment.

The costs, plainly

Needs a network	The checking side needs DNS with DNSSEC validation. That is why I keep it away from the device during bootstrap.
Not instant revocation	Removing a record works as caches expire, but an old signed record can be replayed until its signature runs out. Keep signatures short for important keys.
Your zone matters	Whoever controls the domain controls the keys, so registrar lock and care over DS records matter.
Privacy	Lookups can be seen; the draft recommends encrypted DNS and query name minimisation.

If the DNS check fails, the parties still have their configured anchors to fall back on.

What I would like to learn from you

- Does anyone run, or plan to run, registrars, MASAs or EST servers that must trust each other across companies?
- Would industrial, automotive, aerospace or power profiles find this useful?
- If it is worth anything, where does it belong: a short informative note in 802.1, or nowhere here, with the work staying in the IETF (ANIMA, DANCE)?
- And a further-out idea: an operator's own zone vouching for the local keys (LDevIDs) of the devices it enrolled. Worth a look, or out of scope?

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

Kaveh Ranjbar, kaveh@whisper.security. Disclosure: I reviewed the DANCE draft and am named in its acknowledgements, and Whisper Security builds identity products on DNSSEC and DANE. The detailed version of these slides, with every quote and reference, is available.