Defined-trust Transport for Limited Domains

draft-nichols-tsv-defined-trust-transport-00

(A secure IP transport with link-local multicast capability)

K. Nichols nichols@pollere.net
V. Jacobson vanj@cs.ucla.edu
R. King randy.king@operantnetworks.com

IETF 114

RFC8520 (MUD) points out that requiring device **enrollment** is necessary but not sufficient to secure an IoT system.

Unfortunately, examples of real-world consequences abound:

Compromised light bulb takes over entire IoT network

Video surveillance company's 150,000 customer camera feeds hacked

LoRaWan encryption can be easily hacked

RFC8520 also notes that Things have restricted **roles** with rigid communication **constraints**. *Enforcing* these roles and constraints could have prevented all of the listed attacks. Almost everything needed for enforcement exists.

- A **topic-based** pub/sub application layer is common in IoT. Topic visibility at *transport-level* exposes a publication's **intent**.
- Securing pub/sub any-to-many transport **requires** per-publication **signing** for provenance: use **enrolled** identity as signing certificate.
- Chain-of-trust **identities** can attest to **role, capabilities**, **attributes**. Shared root-of-trust lets enrollees validate other members.

➡ Each transport instance knows who (identity) is saying what (topic); if it knows the constraint rules it can enforce them.

This is what DeftT does!

This approach yields benefits beyond security. Communication with *topics*, not endpoints:

- is inherently broadcast-friendly, moves efficiency from $O(n^2)$ to $O(n)$
- doesn’t need brokers/hubs (eliminates single point of failure/attack)
- doesn’t use *endpoint* identities (all addresses can be link-local self-assigned, no DNS, DHCP or ARP)

Obviates traditional security approaches of firewalls, air gaps, access control lists, and end-point authentication

Two “big ideas” in DeftT

1. Trust management **integrated** into the transport, using trust rules specific to each deployment

2. Transport communication model **embraces** a broadcast physical layer with topic-based collections rather than pipes connecting endpoints

DeftT uses trust schemas for (1) and set reconciliation for (2), both relatively recent advances.

Each can be employed separately from DeftT. In particular, (1) can be deployed with application pub/sub protocols like MQTT.

Defined-trust Communications Basics

- Configuration/bootstrap (using any reasonable approach) enrolls a Thing in a trust domain with a “bundle” containing:
  - Trust Anchor (TA) for the trust domain
  - trust schema cert (*signed* by the TA) containing **rules**
  - **identity** (public cert *signed* by TA + private signing key)
- **Locally generated** Trust Anchors are expected (but not required)
- Identities distributed as chains-of-trust
  - DeftT contains validation that checks this signing chain
  - chain contains roles, attributes, capabilities

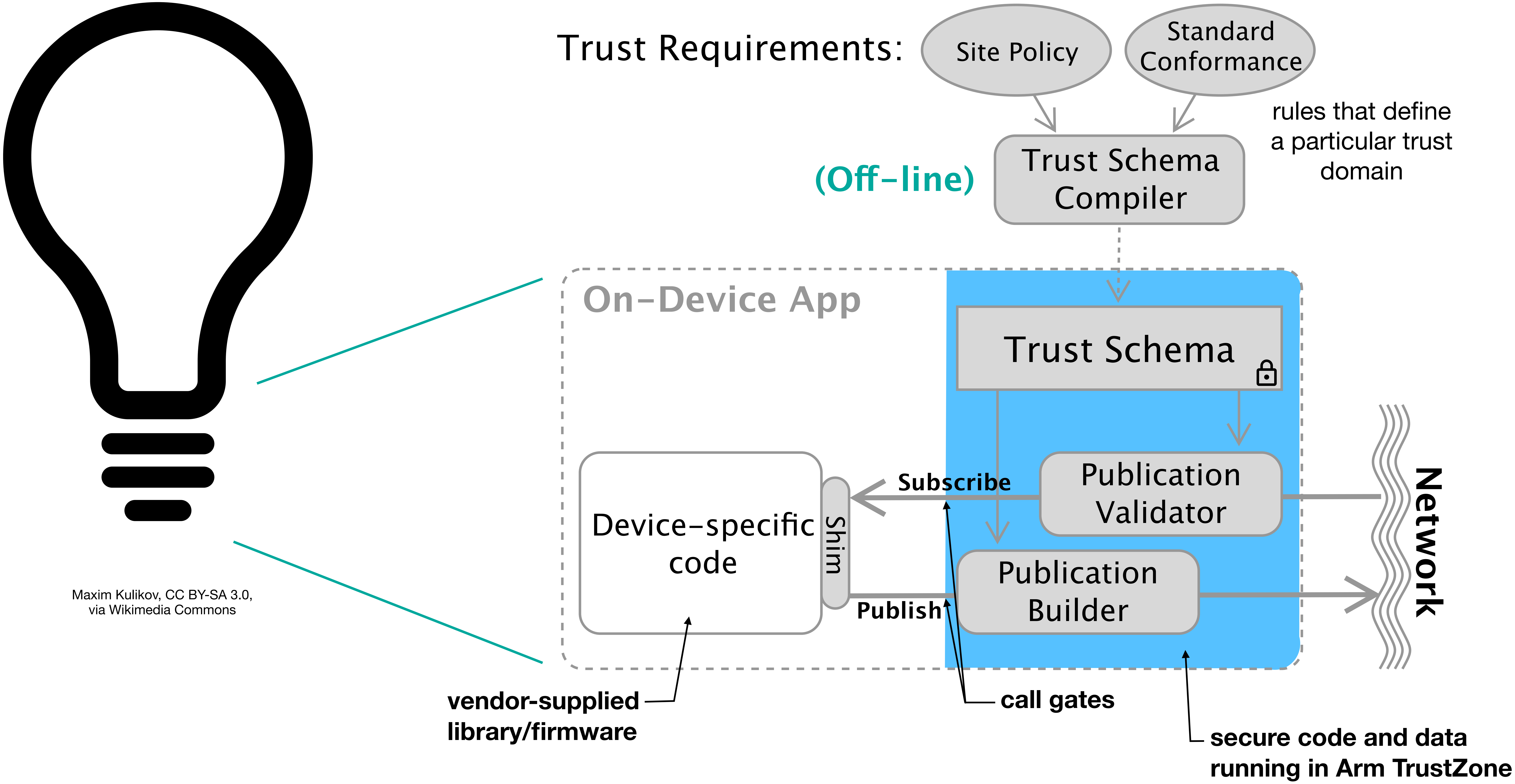
Schematized trust rules can be as simple as requiring that an enrolled identity signs all domain communications

- rules can evolve to finer-grained, role-, attribute-, capability-based
- *self-configuring* privacy via AEAD encryption with automatic, secure nonce key distribution via encryption using identity public keys

Trust schema updates can change run-time rules with ***no*** changes in code

- security for a domain can be increased; no code recompiles

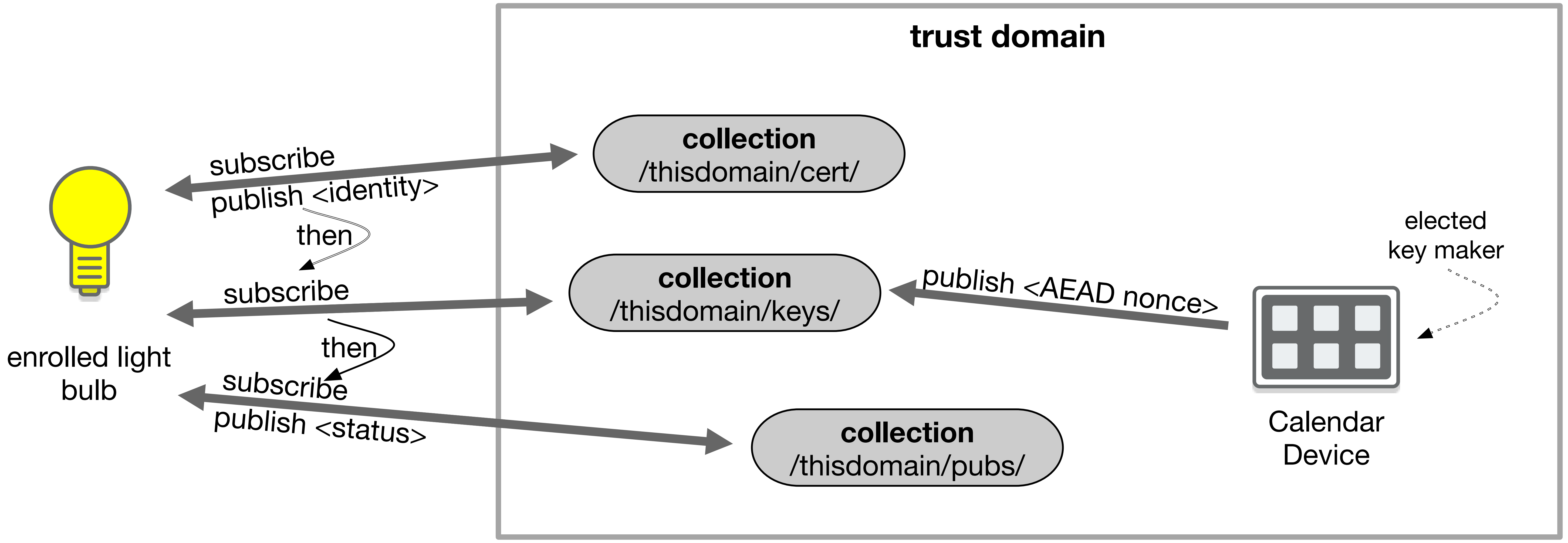
Trust schemas can be reused in different deployments by changing the TA



Maxim Kulikov, CC BY-SA 3.0,
via Wikimedia Commons

step1: Enroll light bulb by bootstrapping with its bundle

step 2: Enrolled light bulb joins trust domain with no use of external servers or routers



for privacy, key maker-capable devices elect a key maker

- light can **listen** for topics in /thisdomain/pubs with specific name, room, floor, or *all* lights.
- light can only **report** its status and *not* issue any commands

Notes

- Goal for this draft is an independent submission informational RFC
    - looking for feedback to improve the draft first
    - *possibly* collaborators on defined-trust communications
- Expose some ideas on transport and security that might be useful in other IETF work
- Pollere maintains an open source reference implementation on github, along with tools and examples. Bug reports are welcome (we may be slow to respond)