CoAP over GATT

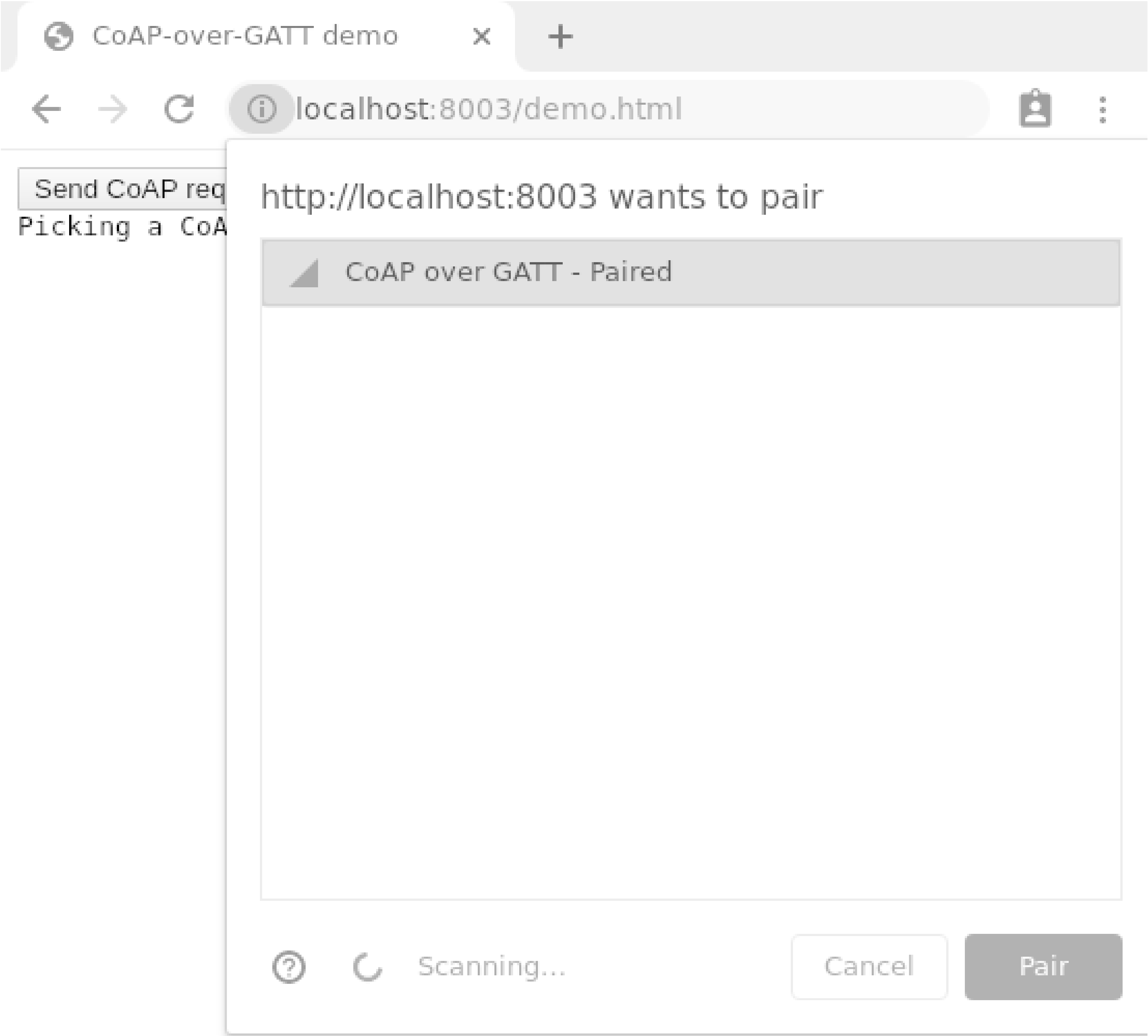
`draft-amsuess-core-coap-over-gatt`

Christian Amsüss

IETF117 San Francisco, CoRE, 2023-07-25

For −04: More about discovery

A slide from IETF116



Picking a CoA

http://localhost:8003 wants to pair

CoAP over GATT - Paired

What if we already know our peer...

...and there are many CoAP-over-GATT devices around?

“know”: by URI, by AS/audience, by advertised resources?

Changes to −04

- Use two separate characteristics for inbound and outbound traffic.
- Define semantics of service data field in advertisement field.
- `coap+gatt://001122334455.ble.arpa/`

  … `.arpa`?

Why go with `.arpa`?

- We already have two kinds of natural identifiers: BLE MACs (useful to some, unavailable toothers) and self-described names (transported in service data field)
  `coap+gatt://001122334455.ble.arpa/`
  `coap+gatt://34373131.ble-sd.arpa/` (“arbitrary” length, as unique as `.local`)
- While RFC 3986 leaves it open, practically there’s only one structure on “host names”.
  Just being in that name space doesn't mean name resolution is involved.
- `coap+gatt://host.example.com/`, `coap://001122334455.ble.arpa/`?
  Possible with ”BLE” records (no intention to pursue) and draft transport-indication draft (Proxy-Scheme but no Uri-Host, not anticipating much need for this on such links).

For further exploration, possibly in a later add-on

Based on a slide from IETF116

- Multicast by CoAP messages in beacons?
- SCHC retconning of non-CoAP attributes. (“Turn the temperature into a CoAP addressable resource”).

Postponed until there is need for it.

Roadmap

Unchanged since IETF116 – but now with a more concrete address proposal

- Update implementations, re-run tests.
- Keep using CoAP-over-GATT as driver for `transport-indication` (could also pick -over-WebSockets or `t2trg-slipmux`)
- Explore this in the WG?
  (Implementation report to arrive at one of the next interims)

Thanks

Comments?

Questions?