

# CoAP over GATT

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

with a side serving of

`draft-ietf-core-transport-indication`

Christian Amsüss

IETF114, CoRE, 2022-07-26

# Timeline

2020 Initial design, -00 and -01, prototypes

IETF109 Proposed for experimental

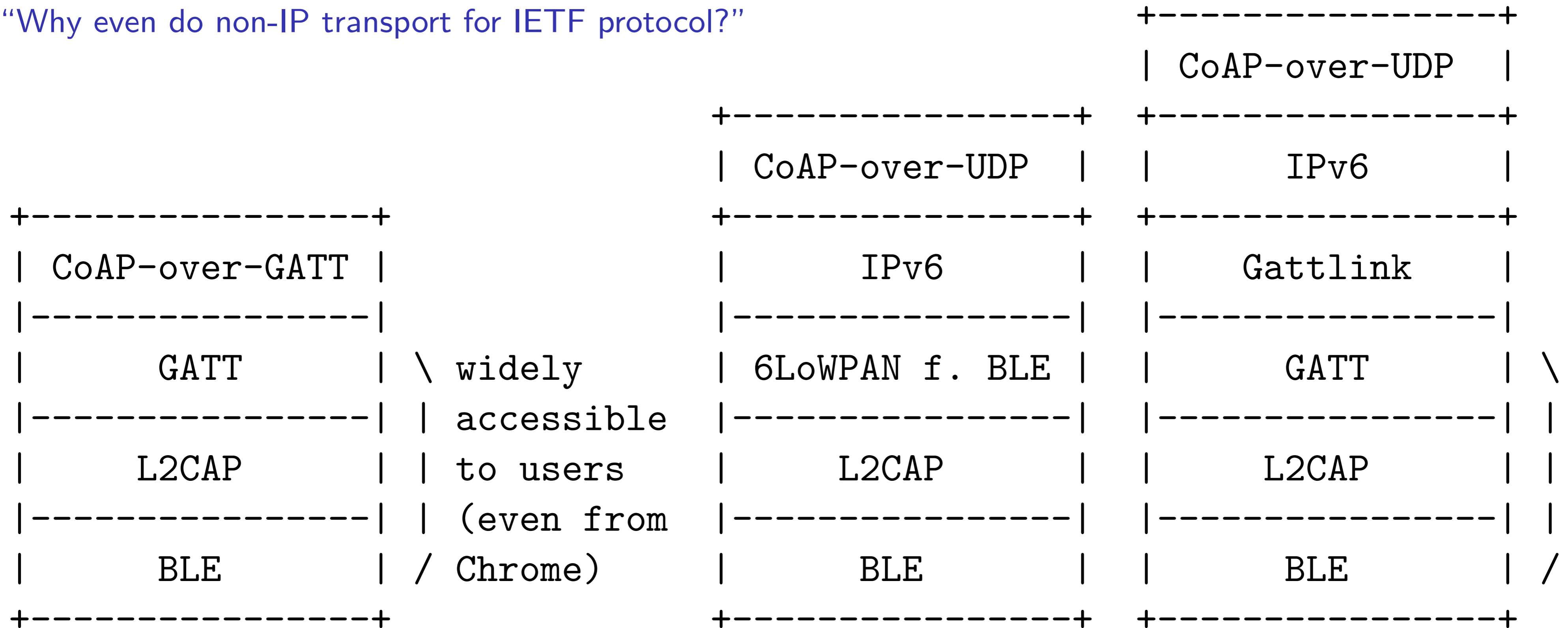
2001 transport-indication gaining traction

2022 Industry interest  
Ongoing filling of gaps

IETF114 ...

# Now explicit: Alternatives

“Why even do non-IP transport for IETF protocol?”



You Are Here

BLE IPSP

Goldengate

# Gaps being filled: Multiplexing, role reversal

Current text:

- Single direction per characteristic.
- Single pending request per characteristic.
- Create as many characteristics as you support observations.
- Message size limited by MTU.

# Gaps being filled: Multiplexing, role reversal

Current text:

- Single direction per characteristic.
- Single pending request per characteristic.
- Create as many characteristics as you support observations.
- Message size limited by MTU.

Reconsidering:

- Requests and responses on a single characteristic.
- Framed like CoAP-over-WebSockets.
- Use composite reads/writes for payloads exceeding MTU.

Shaped by GATT properties (ordered, unilateral reliability – highly unlike UDP), and experience of contributors.

# Addressing

It could be simple...

```
coap+gatt://00-11-22-33-44-55/.well-known/core
```

# Addressing

It could be simple...

```
coap+gatt://00-11-22-33-44-55/.well-known/core
```

...but it is not:

- iOS apps do not get to see MAC addresses, but hashes thereof that are salted with some host identifier.
- W3C Web Bluetooth suggests browsers use different identifiers than MAC addresses.

```
coap+gatt://whatever-the-client-resolves/.well-known/core
```

A bit like `coap://[fe80::815]%enp0s2/.well-known/core?`

# When would we even *want* to address by MAC?

Expected cases:

- Request from device: Likely forwarded through proxy in phone to `coap+tcp://...` address.
- Device registers at some RD-like service: That might pick a better name right away.  
`?ep=device1234&d=example.com`  
`<coap://device1234.example.com/temp>`,  
`</>;rel=has-proxy;anchor=<coap://device1234.example.com/temp>`<sup>1</sup>

This is why LwM2M could use NIDD for years and only recently think of the relevant scheme.

Is an explicitly selected own URI (that is possibly not even on the same transport) a direction `transport-indication` (and by extension `resource-directory-extensions`) should focus more on?

---

<sup>1</sup>Resolution rules might prevent that, `transport-indication` could specify reverse role to avoid the need to explicitly name `</>`, which is hard given the previous slide's caveats.



# Roadmap

- Develop capabilities built on GATT  
Input appreciated, but probably needs to come from BLE experts.
- Use CoAP-over-GATT as driver for transport-indication  
More for illustrative reasons; -over-WebSockets or t2trg-slipmux would have the very similar properties.
- WG feedback – potential for Standards Track?<sup>2</sup>

---

<sup>2</sup>IETF 109 slides said “not before Firefox can do it”; adoption showed that using from browser is not as much the center point as was previously assumed.

Thanks

Comments?

Questions?