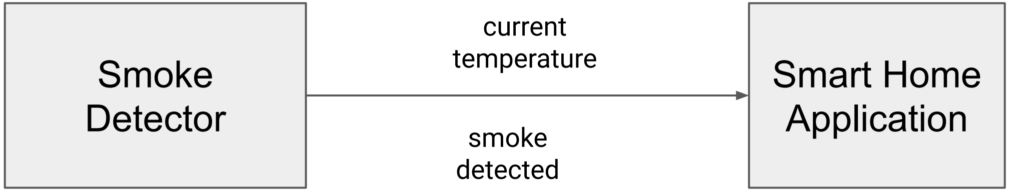
ECLIPSE 2021 CON

Rock solid Device Connectivity

Kai Hudalla, Bosch.IO GmbH

Smart Home Example



First rule of distributed systems:
Don’t distribute.

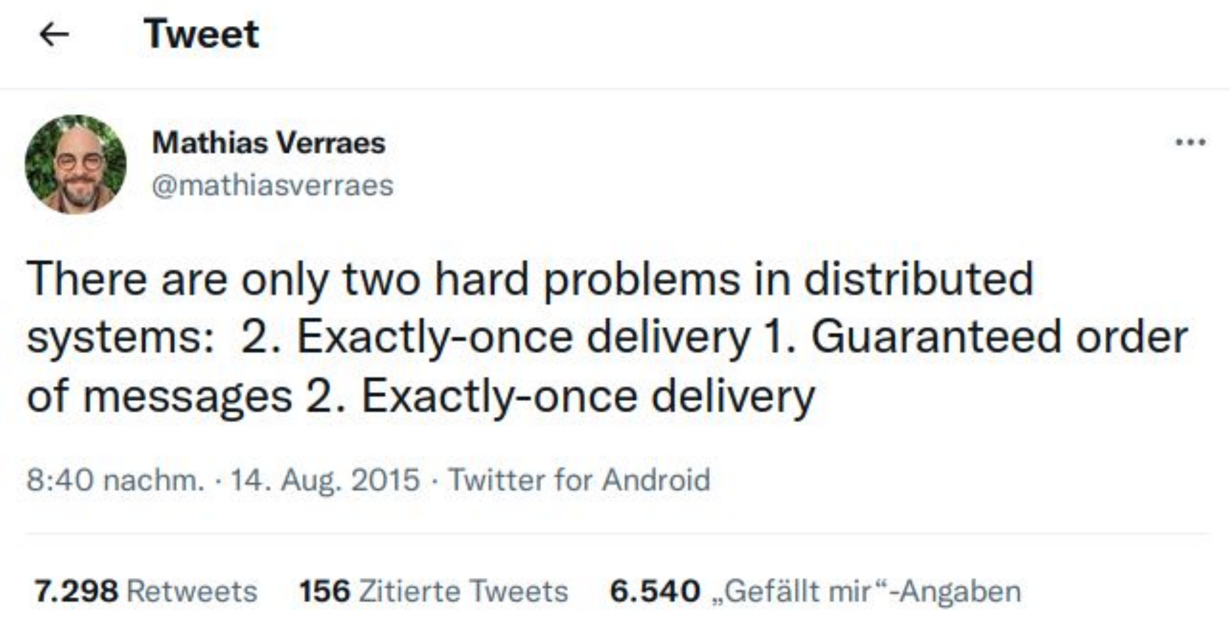
unknown origin

"The 8 Fallacies of Distributed Computing"

Peter Deutsch, James Gosling

1. The Network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. The network is secure.
5. Topology doesn't change.
6. There is one administrator.
7. Transport cost is zero.
8. The network is homogeneous.

https://web.archive.org/web/20171107014323/http://blog.fogcreek.com/eight-fallacies-of-distributed-computing-tech-talk/



Tweet

Mathias Verraes
@mathiasverraes

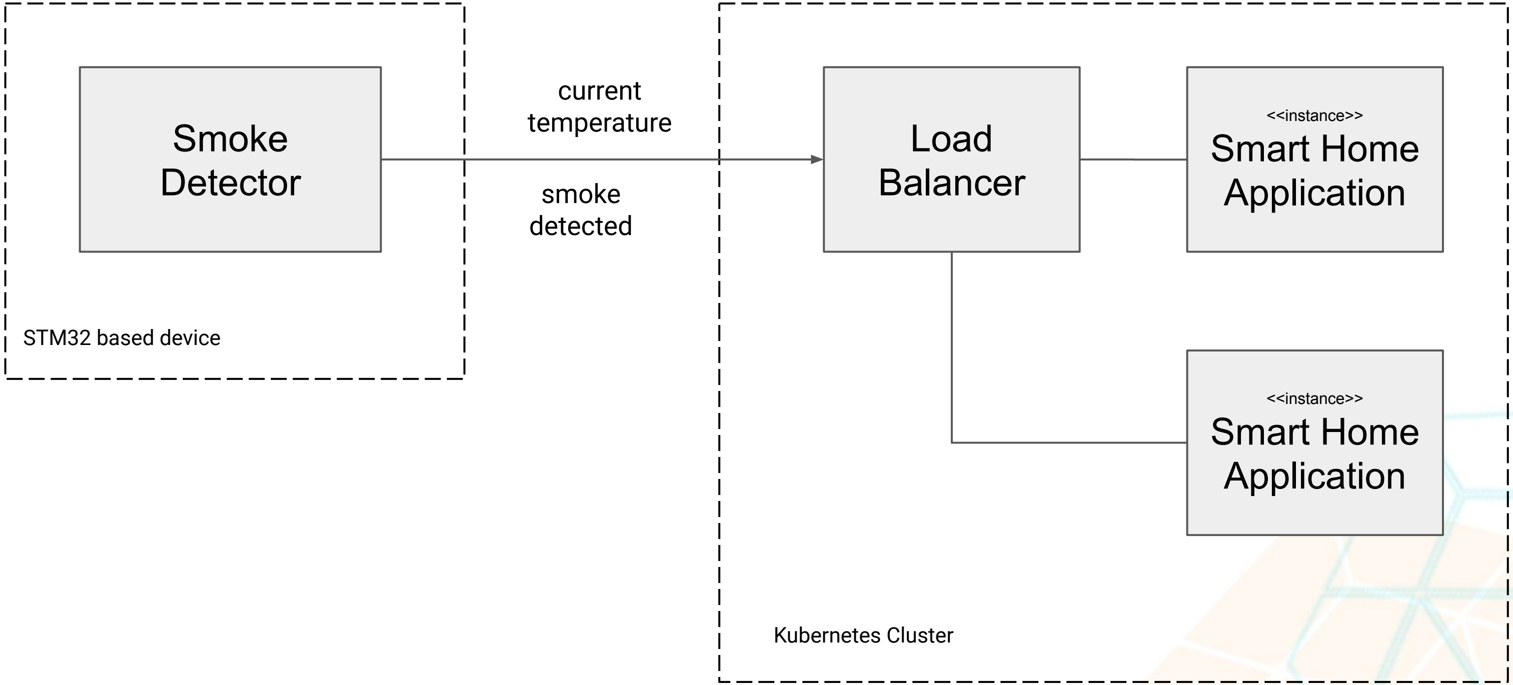
There are only two hard problems in distributed systems: 2. Exactly-once delivery 1. Guaranteed order of messages 2. Exactly-once delivery

8:40 nachm. · 14. Aug. 2015 · Twitter for Android

7.298 Retweets **156** Zitierte Tweets **6.540** „Gefällt mir“-Angaben

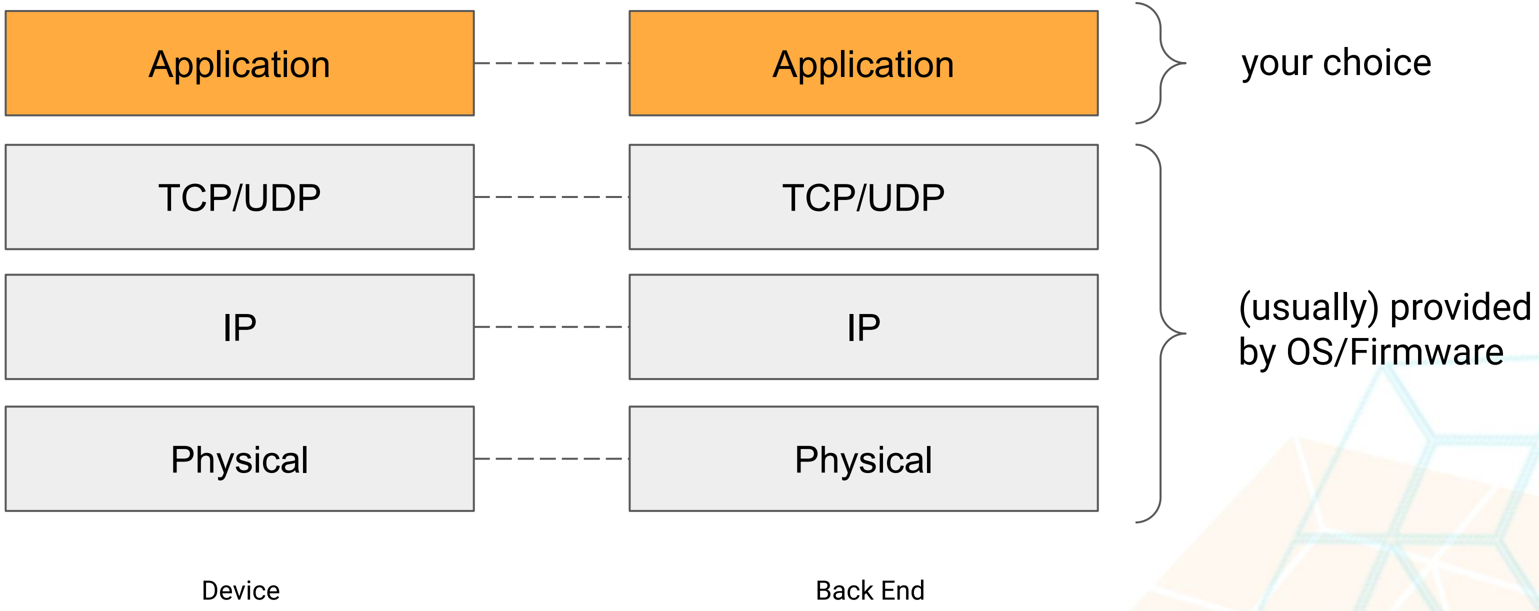
https://twitter.com/mathiasverraes/status/632260618599403520

Smart Home Example deployed



How to transmit the Data?

Use an existing application layer protocol and encode data in the payload



Application

Application

your choice

TCP/UDP

TCP/UDP

IP

IP

Physical

Physical

(usually) provided
by OS/Firmware

Device

Back End

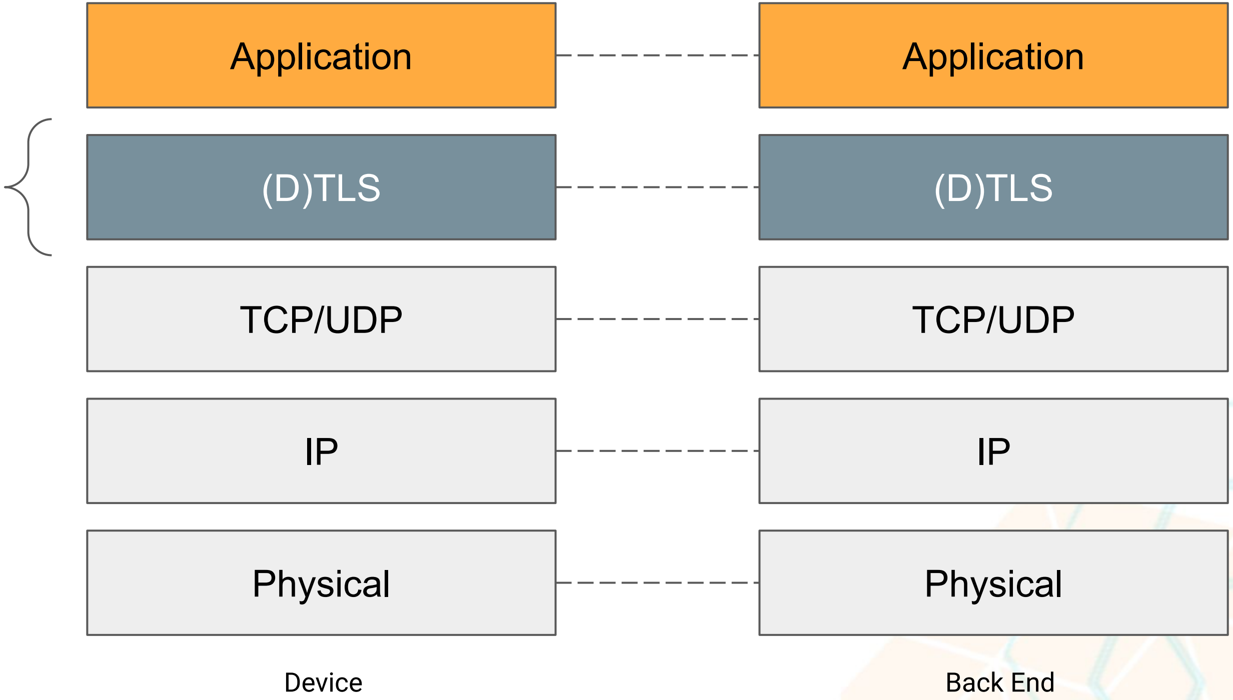
Application Layer Protocols of Choice

Protocol	Transport	Authentication	Meta Data	Pros/Cons
AMQP 1.0	TCP	SASL	connection, link, message	(+) feature rich, compact (+) flow control (-) few implementations
MQTT 5	TCP	TLS, (SASL), MQTT	message	(+) feature rich, compact (-) few implementations
CoAP	UDP	DTLS	message	(+) very compact (+) "binary HTTP"
HTTP	TCP	TLS, HTTP	message	(+) many implementations (-) verbose
MQTT 3.1.1	TCP	TLS, MQTT	no	(+) many implementations (-) no NACKS

How to keep Confidentiality?

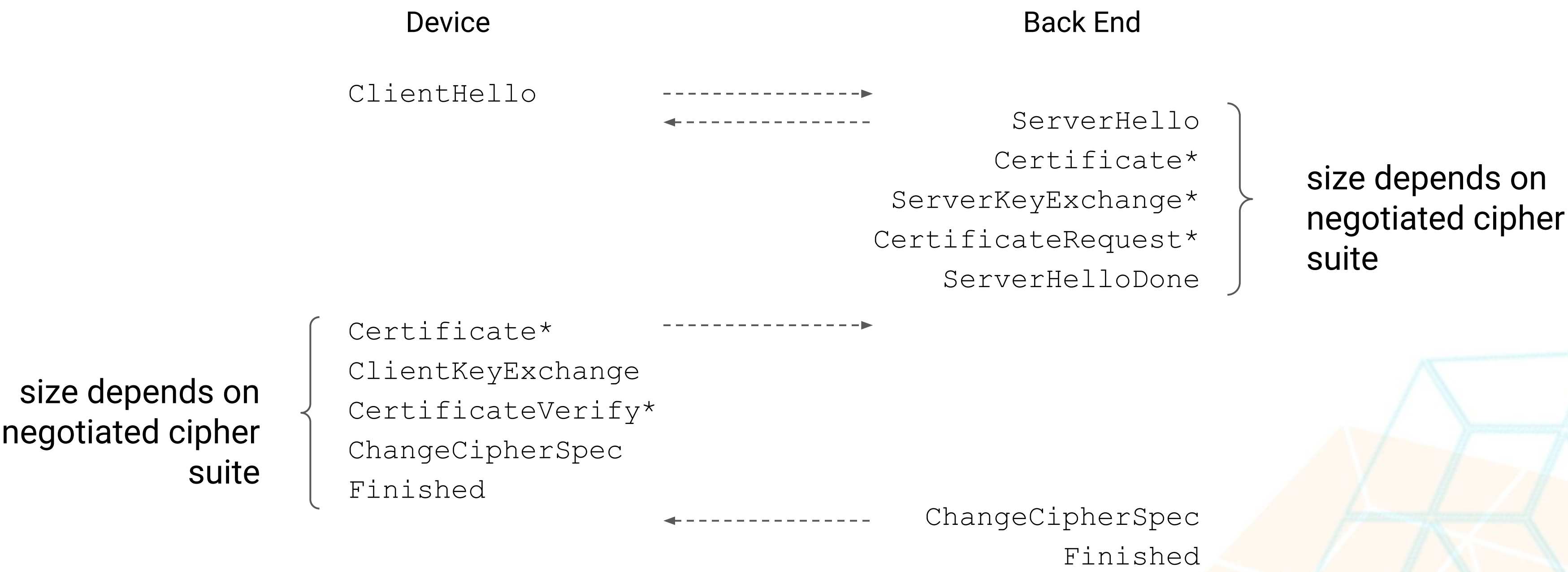
Use Transport Layer Security (TLS)

- OpenSSL
- OpenJDK
- BouncyCastle
- Eclipse tinydtls
- Eclipse Californium



How to keep Confidentiality?

Full TLS 1.2 Handshake Protocol



Device

Back End

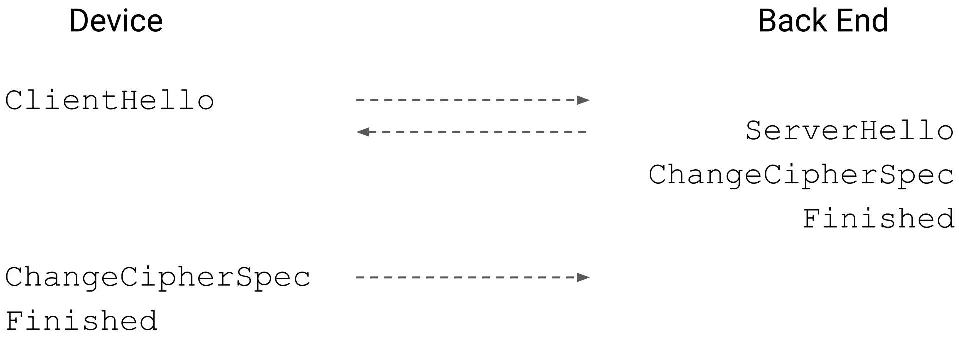
```
ClientHello                -------------->
                           <--------------      ServerHello
                                               Certificate*
                                         ServerKeyExchange*
                                        CertificateRequest*
                                             ServerHelloDone
Certificate*               -------------->
ClientKeyExchange
CertificateVerify*
ChangeCipherSpec
Finished
                           <--------------  ChangeCipherSpec
                                                    Finished
```

size depends on negotiated cipher suite

size depends on negotiated cipher suite

How to keep Confidentiality?

Abbreviated TLS 1.2 Handshake Protocol ("session resumption")



Device

Back End

ClientHello

ServerHello

ChangeCipherSpec

Finished

ChangeCipherSpec

Finished

How to keep Confidentiality?

Comparison of Cipher Suite Types

	RSA	ECDSA	PSK
Processing requirements	Cortex M4 class (HW Security Module?)	Cortex M3 class (e.g. Cortex M33 incl. HSM)	Cortex M0 class
Certificate size	1-2 kb	< 1 kb	0 b
Typical key length (bits)	2048	224-255	112

Example TLS 1.2 Suite	`TLS_DHE_RSA_WITH_AES_128_GCM_SHA256`	`TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`	`TLS_ECDHE_PSK_WITH_AES_128_GCM_SHA256,`
			`TLS_DHE_PSK_WITH_AES_128_GCM_SHA256`

Full handshake size	4-7 kb	2-4 kb	< 1 kb
Abbr. handshake size	< 500 b	< 500 b	< 500 b

https://sectigostore.com/blog/ecdsa-vs-rsa-everything-you-need-to-know/
https://danielpocock.com/rsa-key-sizes-2048-or-4096-bits/
https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-2.html

How to keep Confidentiality?

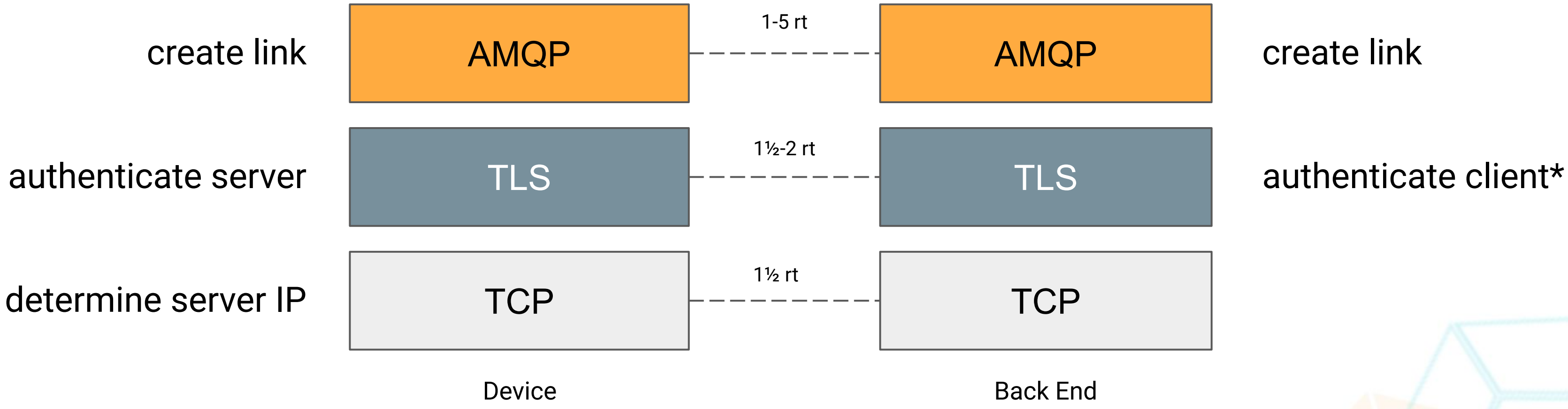
Best Practices

- **Do not** implement your own custom security layer!
- Use TLS 1.3 if possible, **never** use TLS < 1.2
- Configure an appropriate[1] set of cipher suites (on device AND back end)
- Include server certificate's issuer in device's trust store
- Use session resumption if possible
- Use DHE for perfect forward secrecy if appropriate
- Think about key/secret rotation as a **requirement of your system**

[1] according to your device's resources and the required level of security

How to make this robust?

Several (logical) connections need to be established



Before a device can send any data, all connections need to be established!

Reliable Transmission Cost

Hello, would you like to hear a TCP joke?

Yes, I'd like to hear a TCP joke.

OK, I'll tell you a TCP joke.

OK, I'll hear a TCP joke.

Are you ready to hear a TCP joke?

Yes, I am ready to hear a TCP joke.

OK, I'm about to send the TCP joke. It will last 10 seconds, it has two characters, it does not have a setting, it ends with a punchline.

OK, I'm ready to hear the TCP joke that will last 10 seconds, has two characters, does not have a setting and will end with a punchline.

I'm sorry, your connection has timed out... ...Hello, would you like to hear a TCP joke?

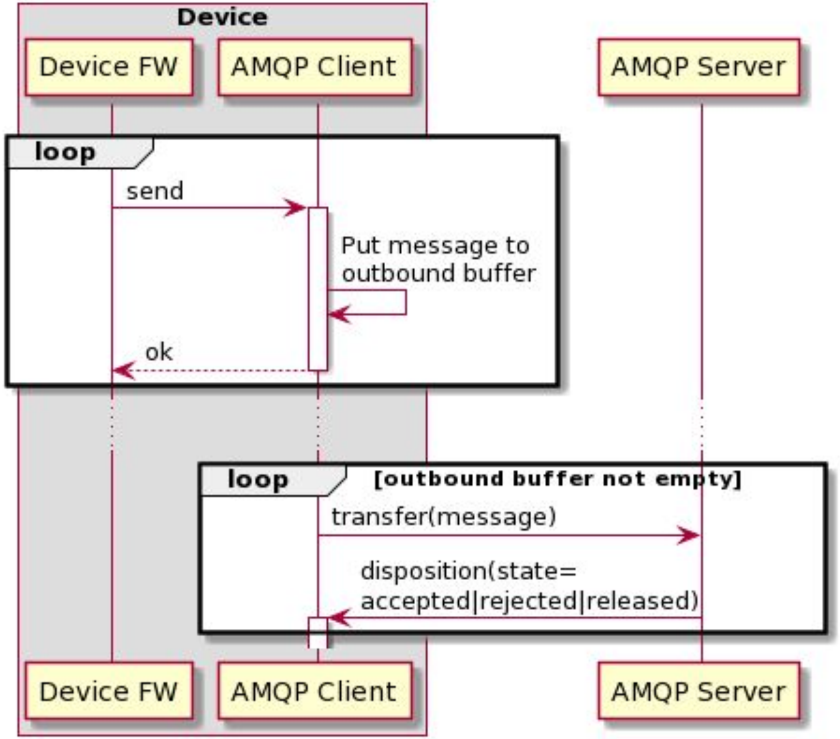
How to make this robust?

Best Practices for establishing a Connection

- Use DNS to resolve back end host's IP address
- Cache DNS data for short time only
- Register handler for connection establishment outcome
- Use an *aggressive* connection timeout
- Enable application layer protocol's *ping/keep alive/heartbeat* mechanism
- Use (exponential) backoff when retrying to connect

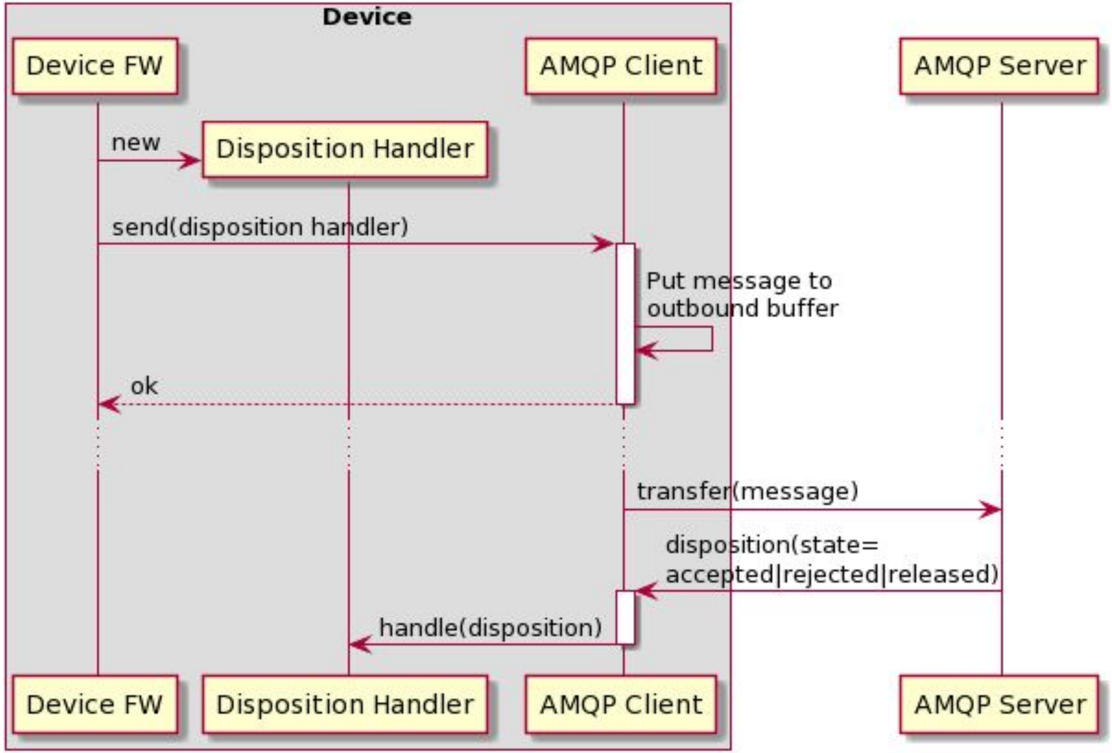
How to make this robust?

Sending Messages using AMQP 1.0


Device
Device FW
AMQP Client
AMQP Server
loop
send
Put message to outbound buffer
ok
loop
[outbound buffer not empty]
transfer(message)
disposition(state=accepted|rejected|released)
Device FW
AMQP Client
AMQP Server

How to make this robust?

Using a Disposition Handler



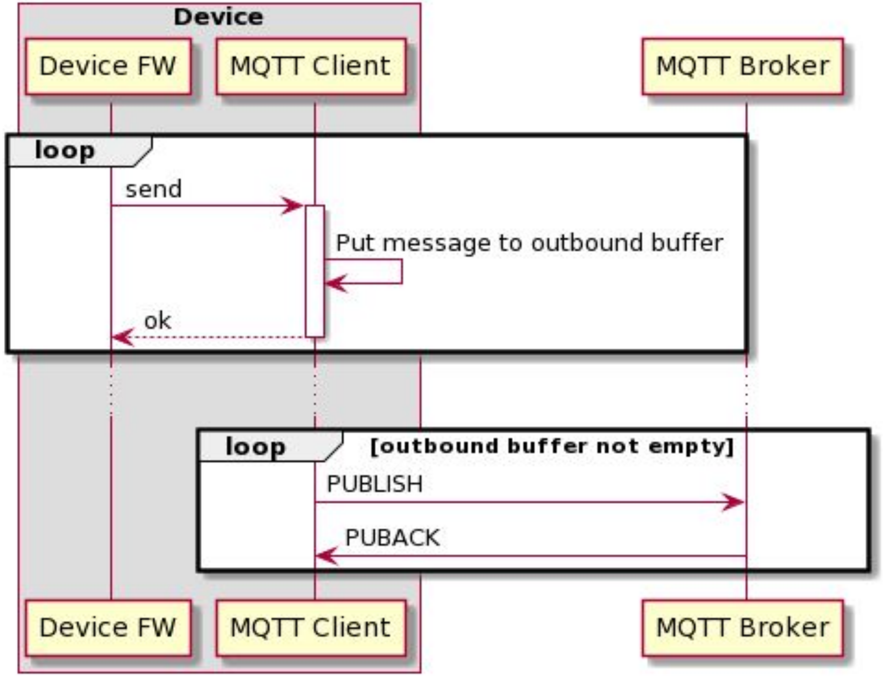
How to make this robust?

Best Practices for sending a Message

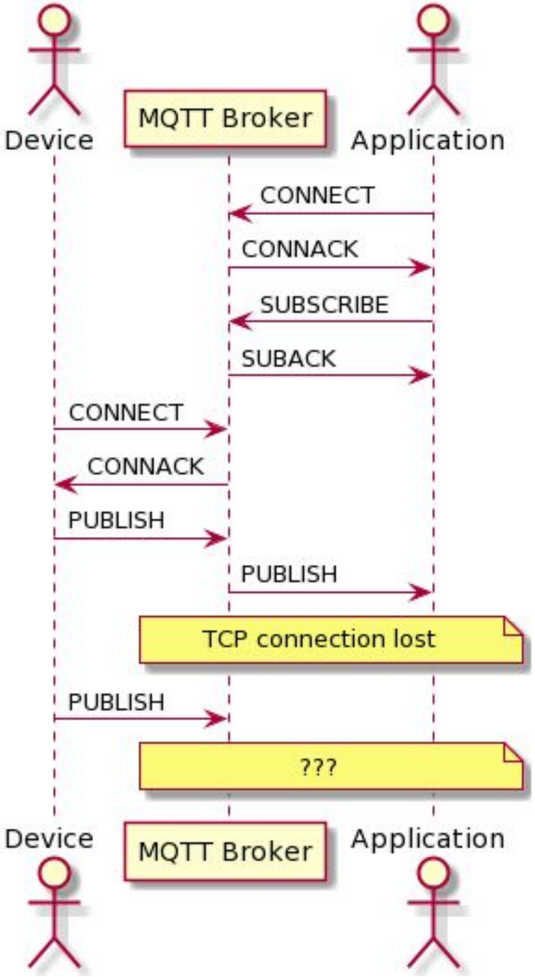
- Register handler for message transmission outcome
- Use an *aggressive* (N)ACK timeout
- Use (exponential) backoff when retrying failed attempt(s)

How to make this robust?

Sending Messages using MQTT 3.1.1



MQTT 3.1.1



Device

MQTT Broker

Application

CONNECT

CONNACK

SUBSCRIBE

SUBACK

CONNECT

CONNACK

PUBLISH

PUBLISH

TCP connection lost

PUBLISH

???

Device

MQTT Broker

Application

Evaluate the Sessions

- Please help by leaving feedback on the sessions you attend!
- To rate a session, you must be registered for it in Swapcard BEFORE the talk starts.
- Swapcard will prompt you to leave feedback after the end of each session.
- You may also rate a talk by locating the session from the “Agenda” or “My Event” buttons on the Event Home page. Click on the session and look for the “Give your feedback” box.

ECLIPSE 2021 CON

Thank you!

Join the conversation:

🐦 @EclipseCon | #EclipseCon

ECLIPSE 2021 CON

Rock solid Device Connectivity

ECLIPSE 2021 CON