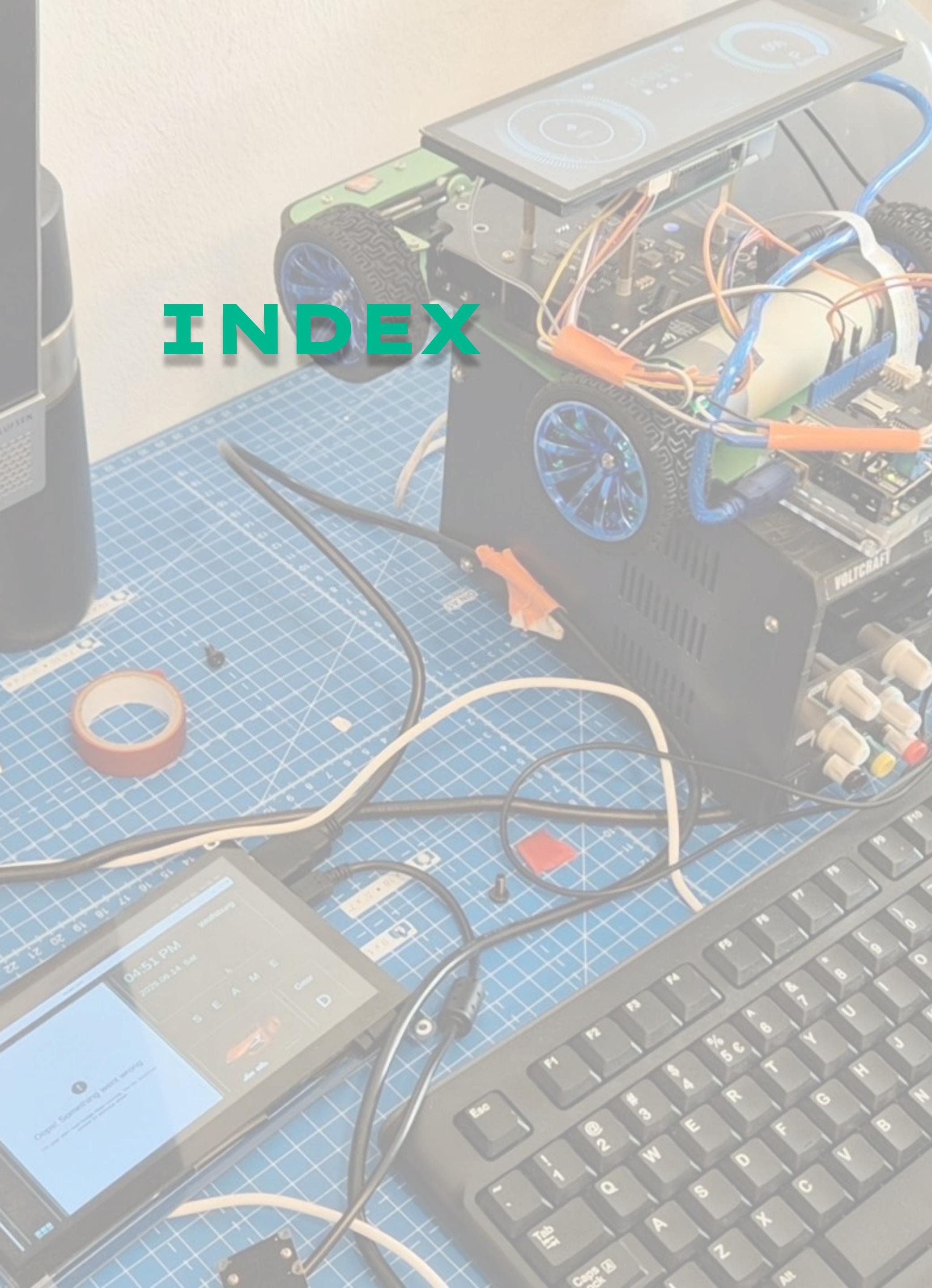


The background of the slide is a collage of various images related to mobile device security and hacking. It includes a close-up of a smartphone screen displaying a root exploit menu; a laptop keyboard and mouse on a desk; a small robot or hobby vehicle with its internal electronics exposed, connected to a computer via cables; and a Bang & Olufsen speaker. A green circular overlay is centered on the robot image, containing the title text.

**SECURE  
OTA**

2024 - 2025 SEA:ME 3rd  
**Mobility Cyber Security**

# INDEX



## 1. Introduction

- a. Members
- b. Project Overview

## 2. Over-The-Air

- a. Update beyond smartphone
- b. Our OTA system

## 3. Attack Scenarios

- a. Fake Server Redirection
- b. Replay Attack



# Members



**Hanbin Yeo**

Architecture  
Network  
Security  
UX/UI Design



**Jangwoon Park**

Architecture  
Network  
Yocto

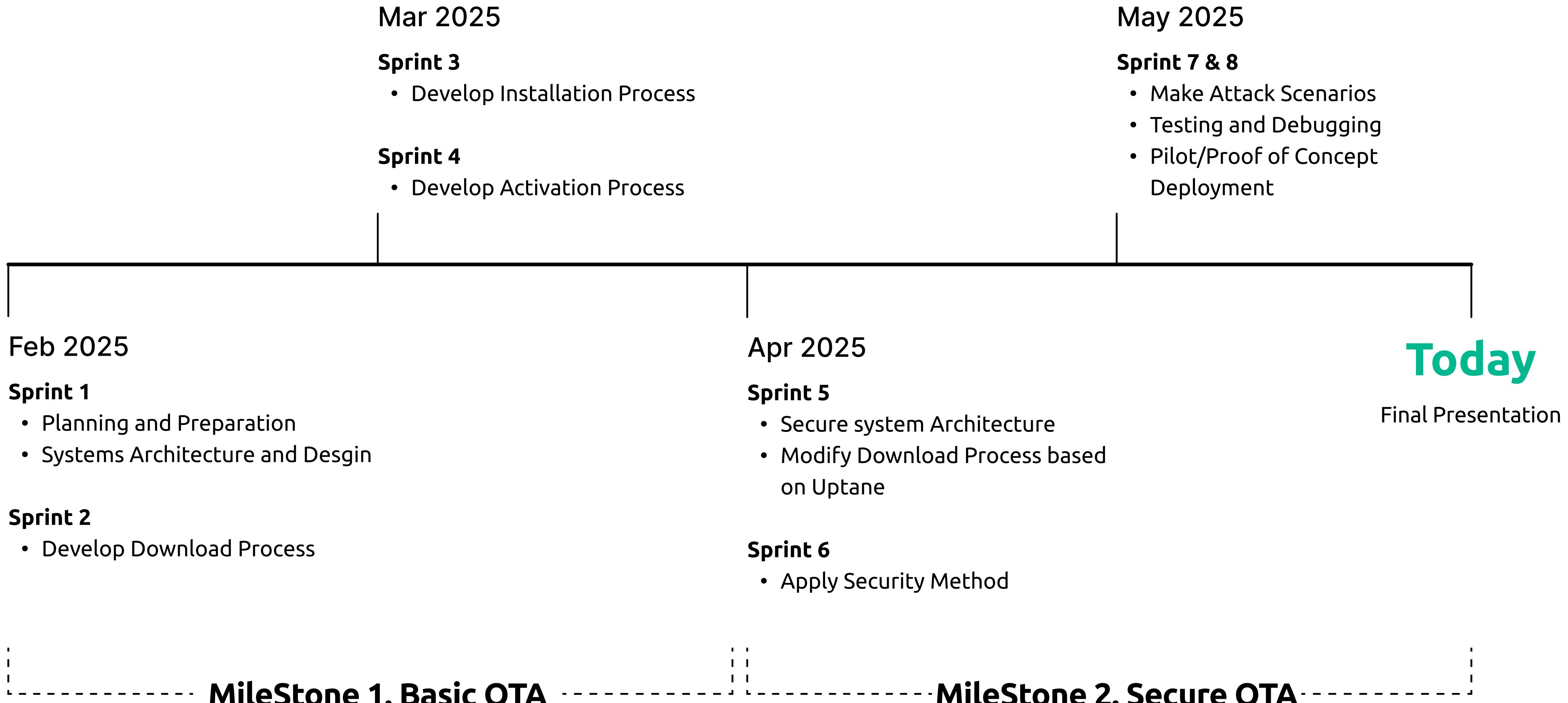


**Kunho Park**

Yocto  
security  
docker



# Timeline



# Over-The-Air(OTA): Beyond Smartphone

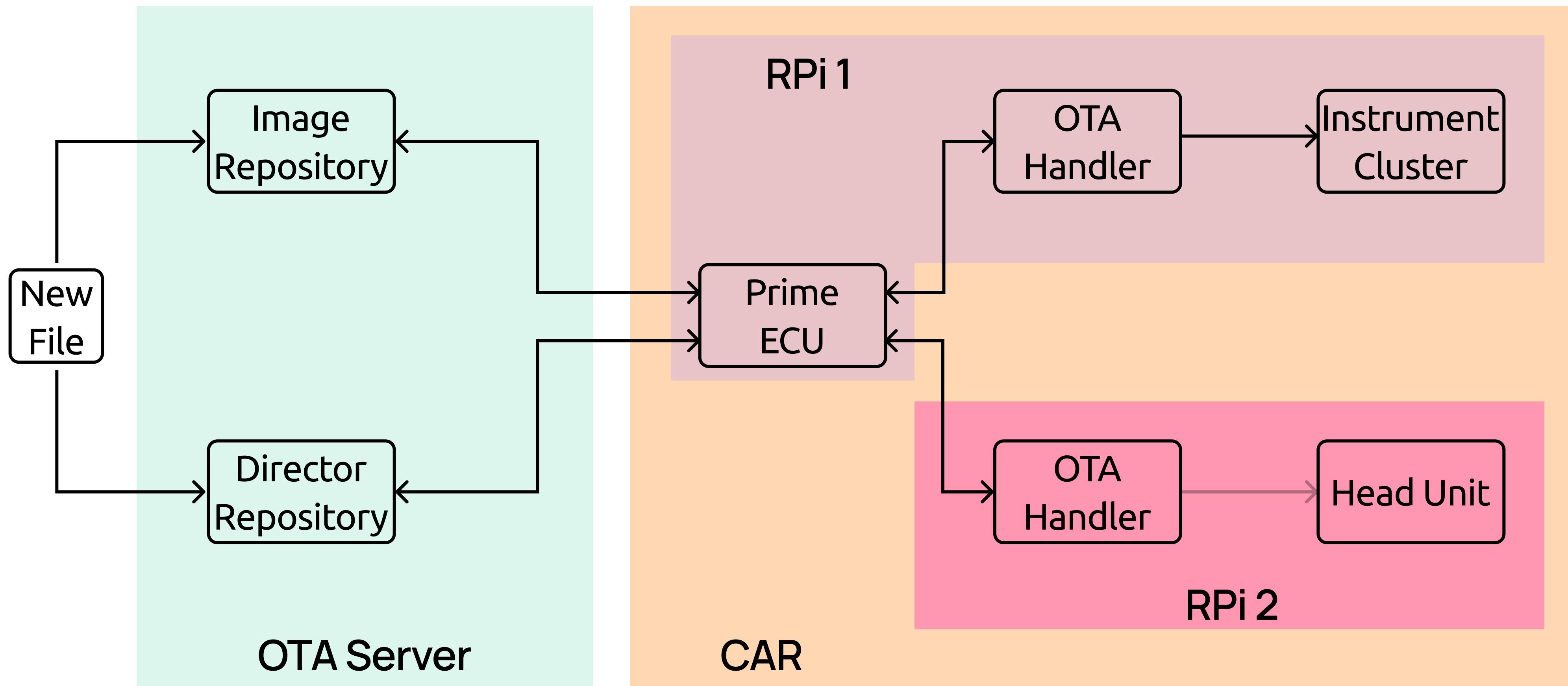
What is Vehicle OTA?



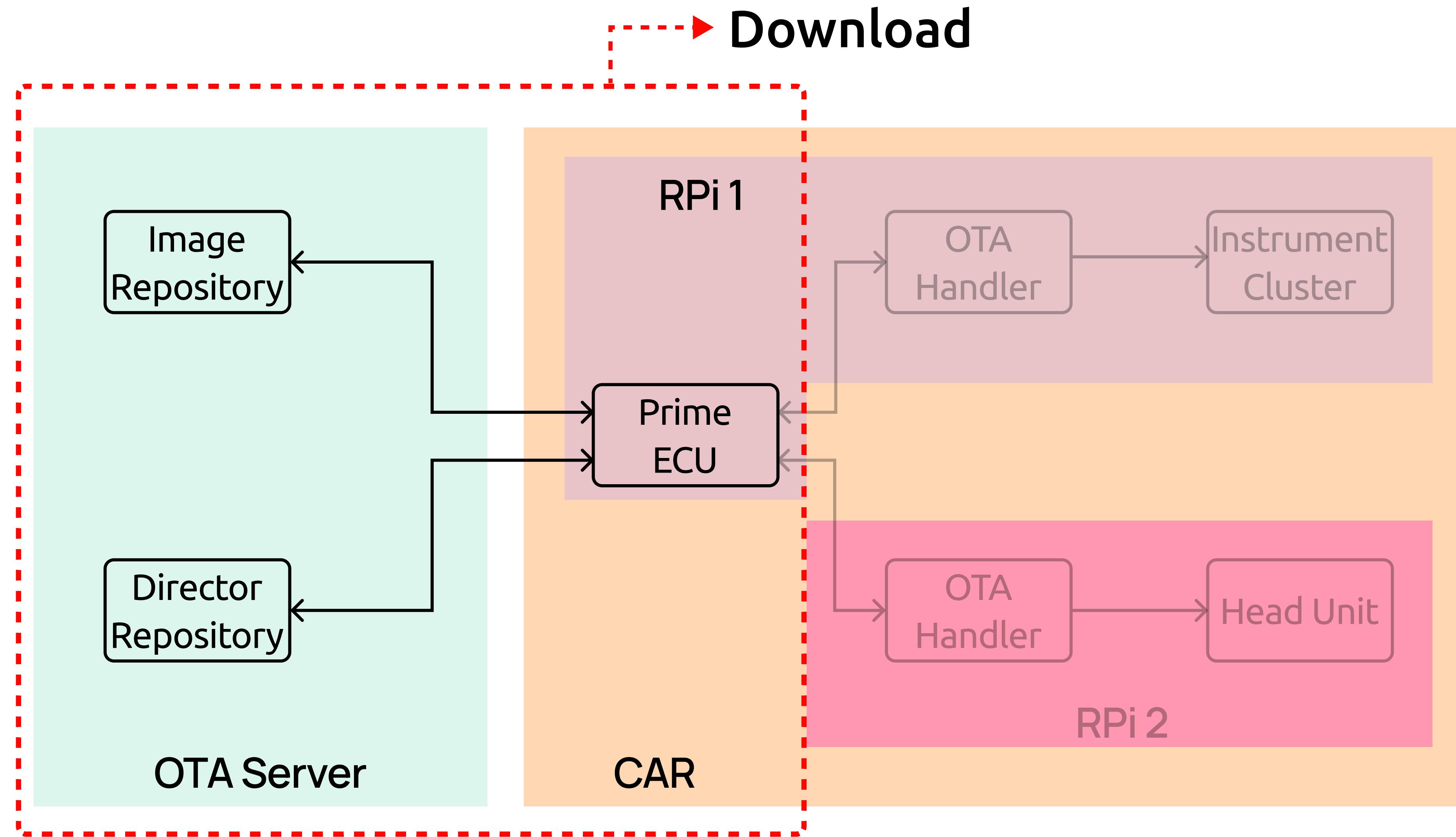
Why is Vehicle OTA  
Different from Phone Software?

-  Safety Criticality
-  System Complexity
-  Cybersecurity Imperative
-  Rollback & Redundancy
-  Regulatory Compliance

# Architecture



# Download



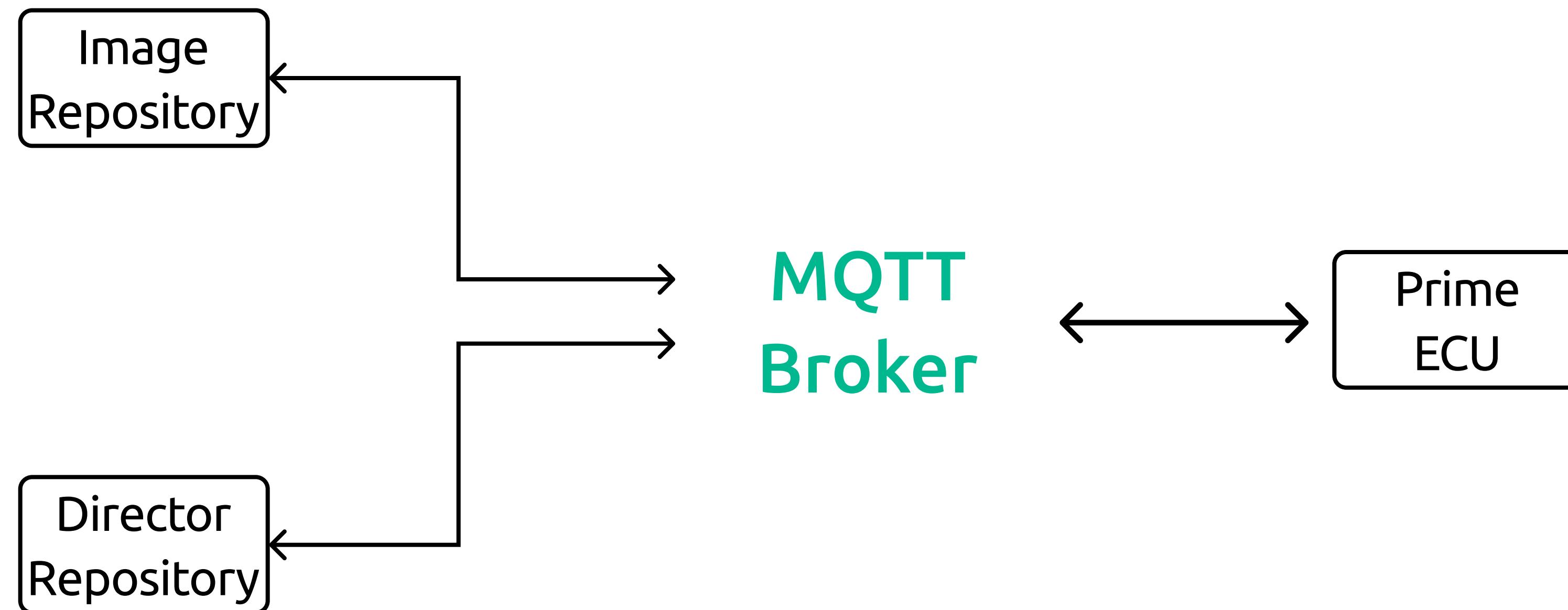
# Uptane Framework

Uptane

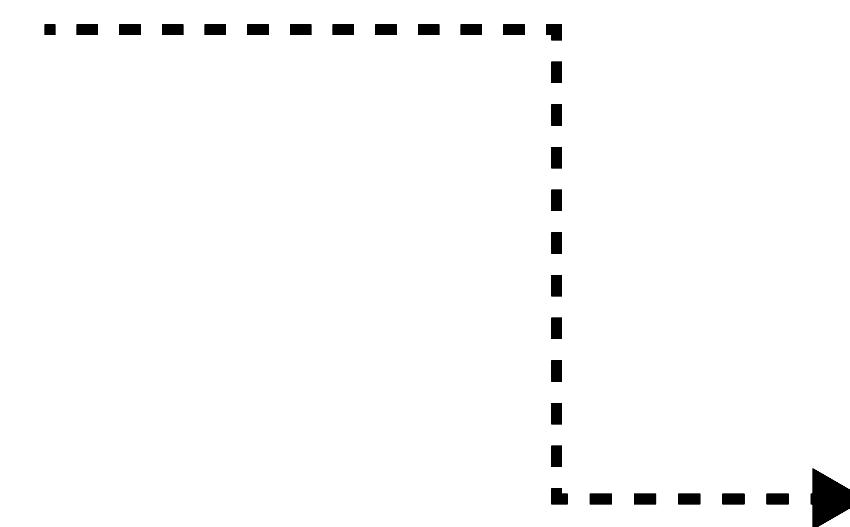
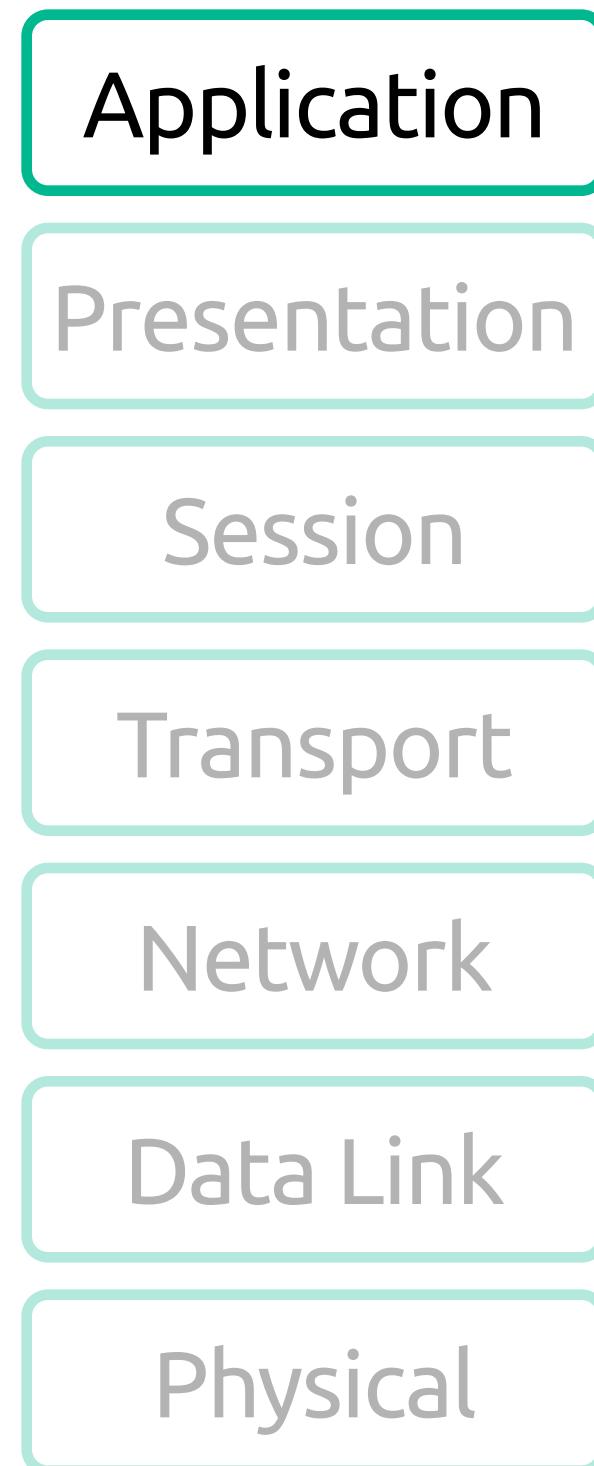


- Car OTA Update Framework
- Separation of trust
- Minimizes the risk

# Update via MQTT



# Signature & Timestamp

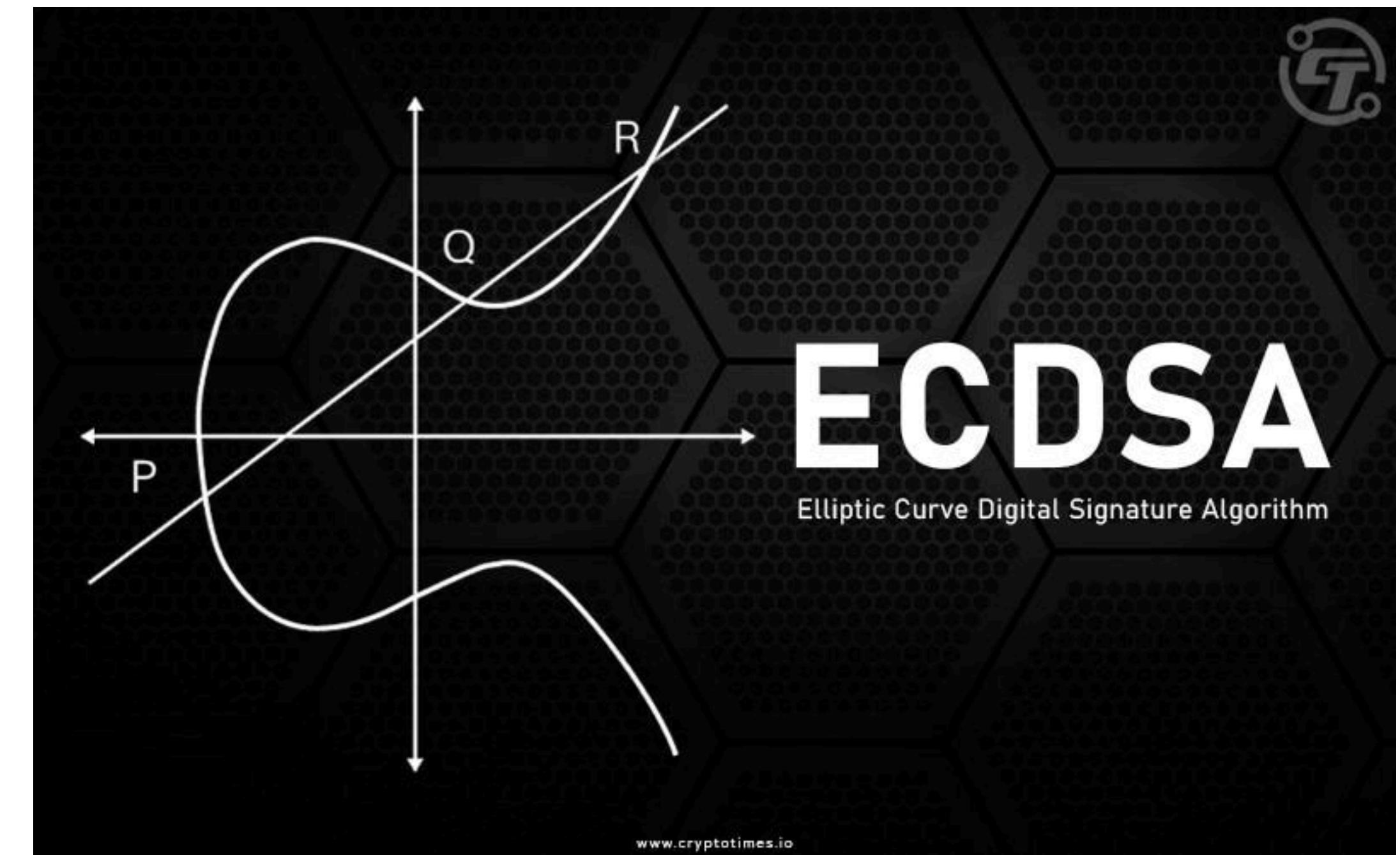


Topic : xxxx/xxxx  
Msg : xxxxxxxxxxxx

**Signature : xxxxxxxxxxxxxxxxxxxx**  
**Timestamp : 2025/06/16/15:52**

# Signature - ECDSA

**ECDSA**  
(Elliptic Curve Digital Signature Algorithm)



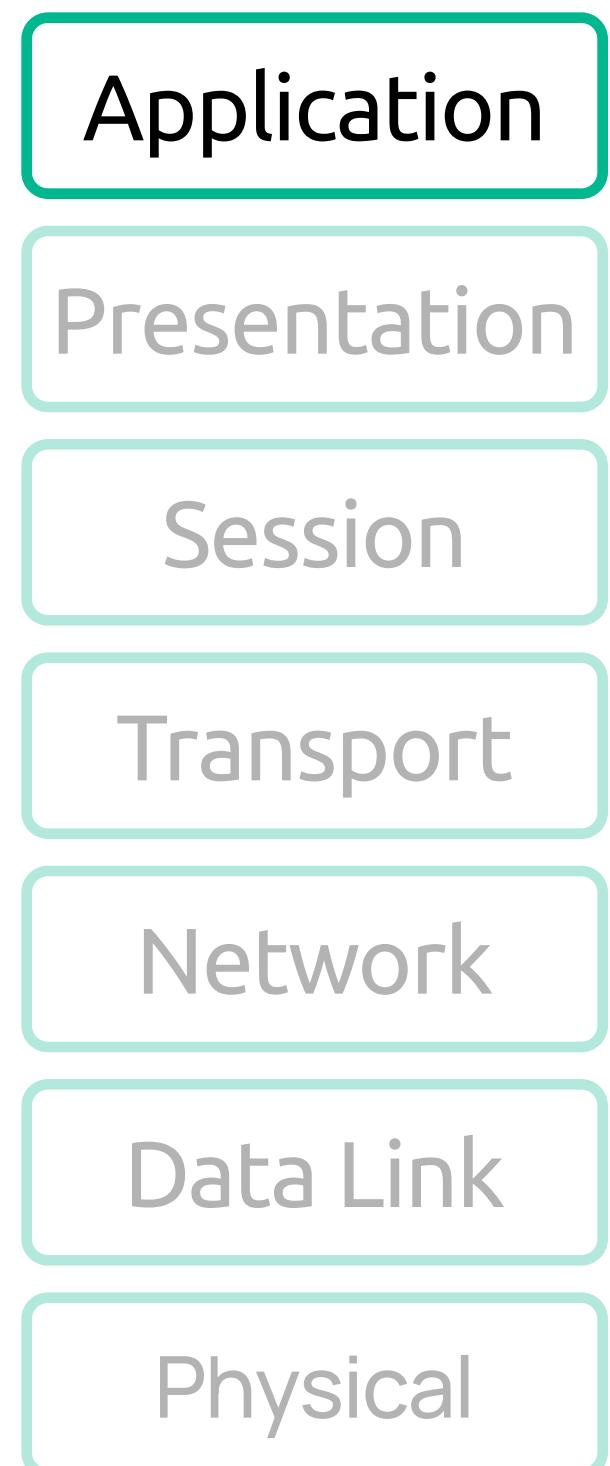
# ECDSA vs RSA



# RSA

**384****Key Length****7680****936****Key Generation****0.4****904****Sign****12****491****Verify****1770**

# Timestamp



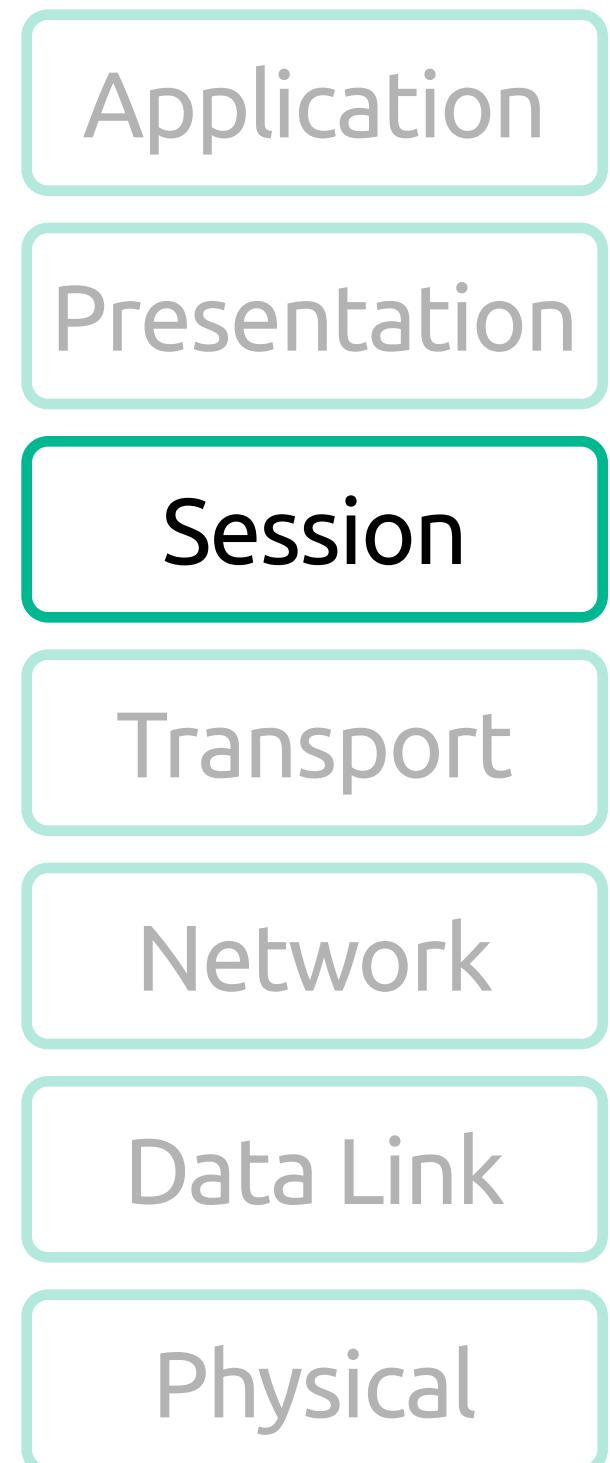
Topic : xxxx/xxxx

Msg : xxxxxxxxxxxx

Signature : xxxxxxxxxxxxxxxxxxxx

**Timestamp : 2025/06/16/15:52**

# TLS



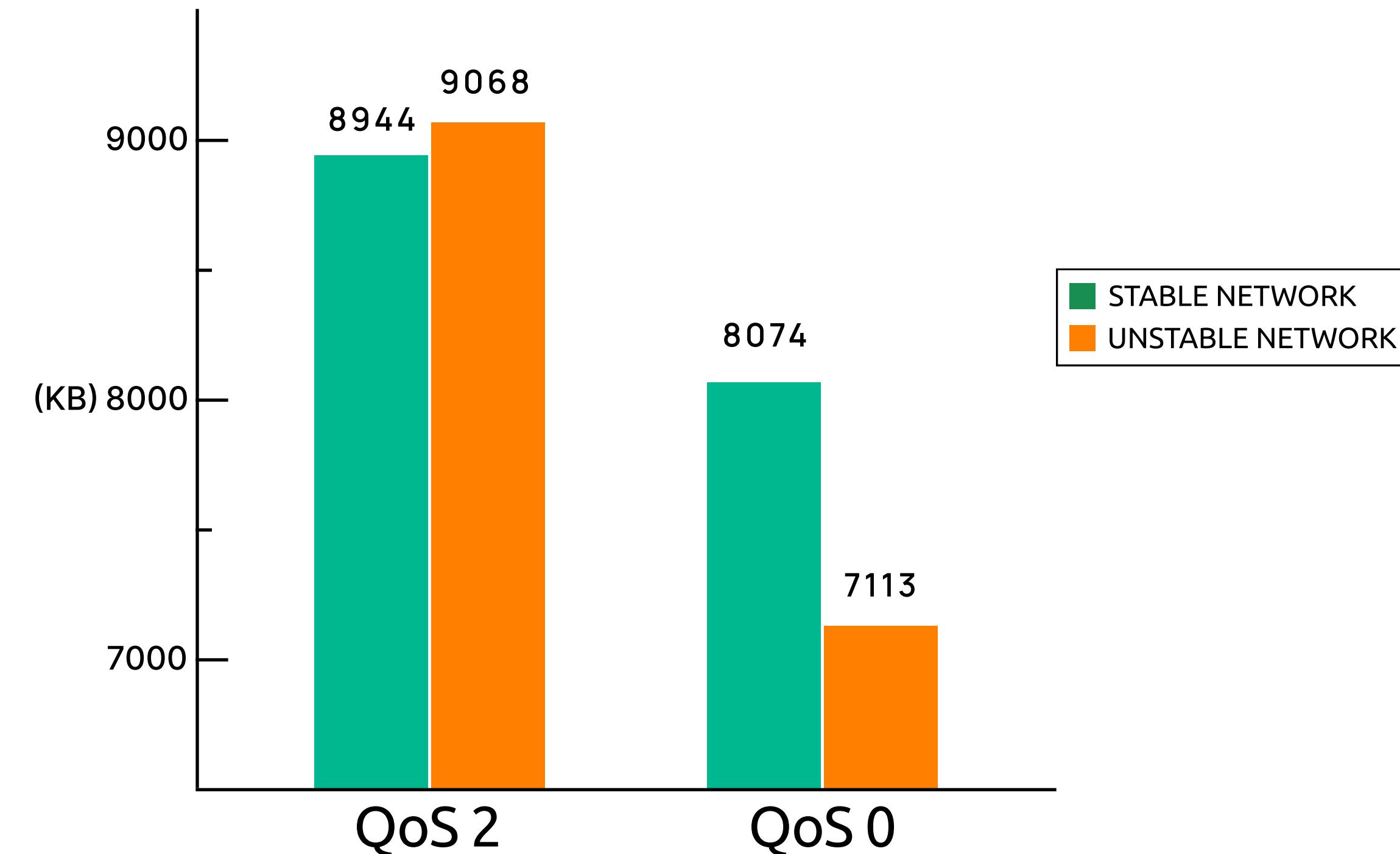
## TLS

Topic : XXXXX/XXXX  
Msg : XXXXXXXXXXXX  
signature : XXXXXXXXXXXXXXXXXX  
Timestamp : 2025/06/14/15:52

# MQTT QoS Setting

- QoS 0 - At Most Once
- QoS 1 - At Least Once
- QoS 2 - Exactly Once

## Network Traffic Throughput



# MQTT QoS Setting

## QoS 0 - At Most Once

- Current version state

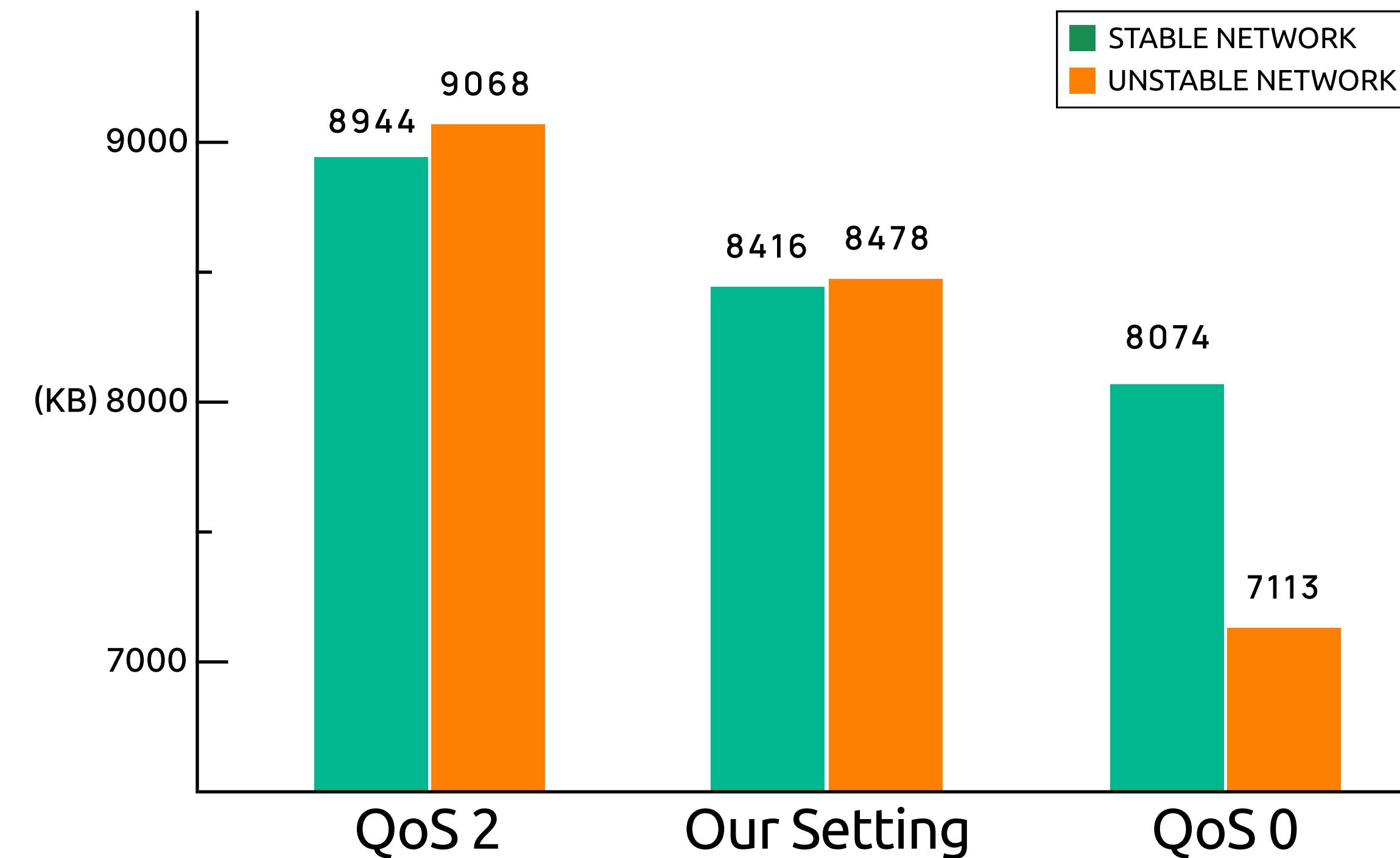
## QoS 1 - At Least Once

- Update meta data
- Request meta data

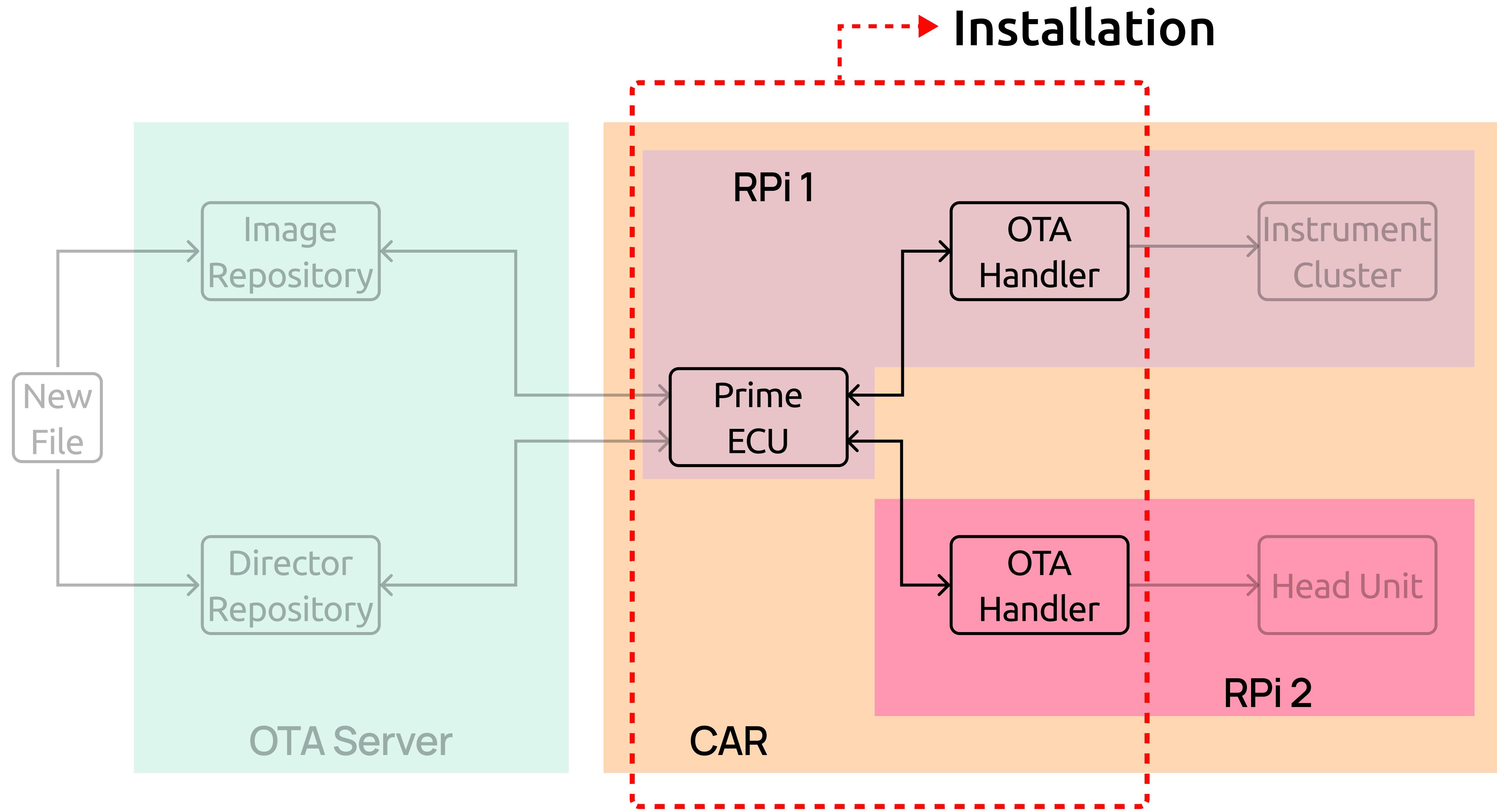
## QoS 2 - Exactly Once

- Download URL

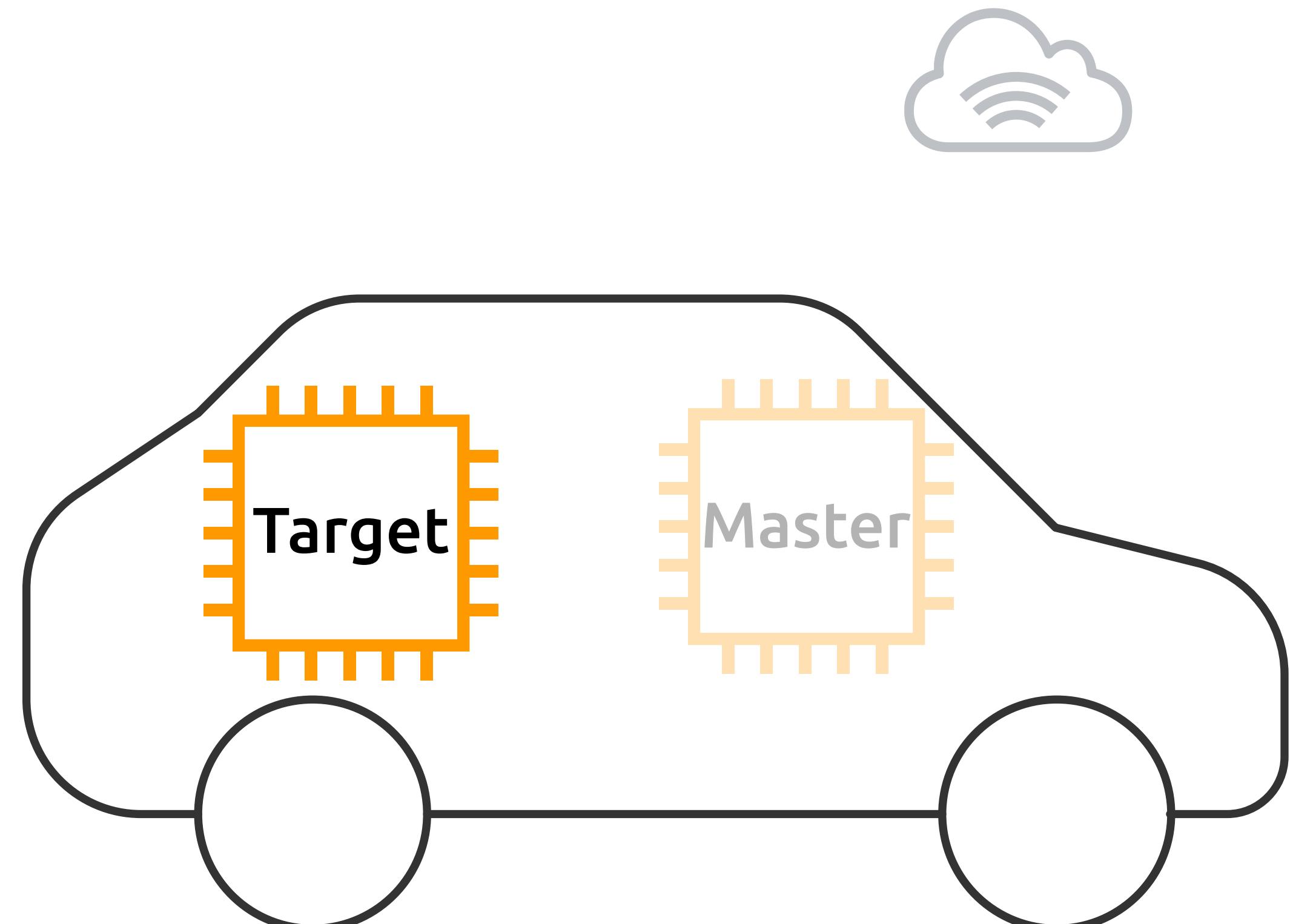
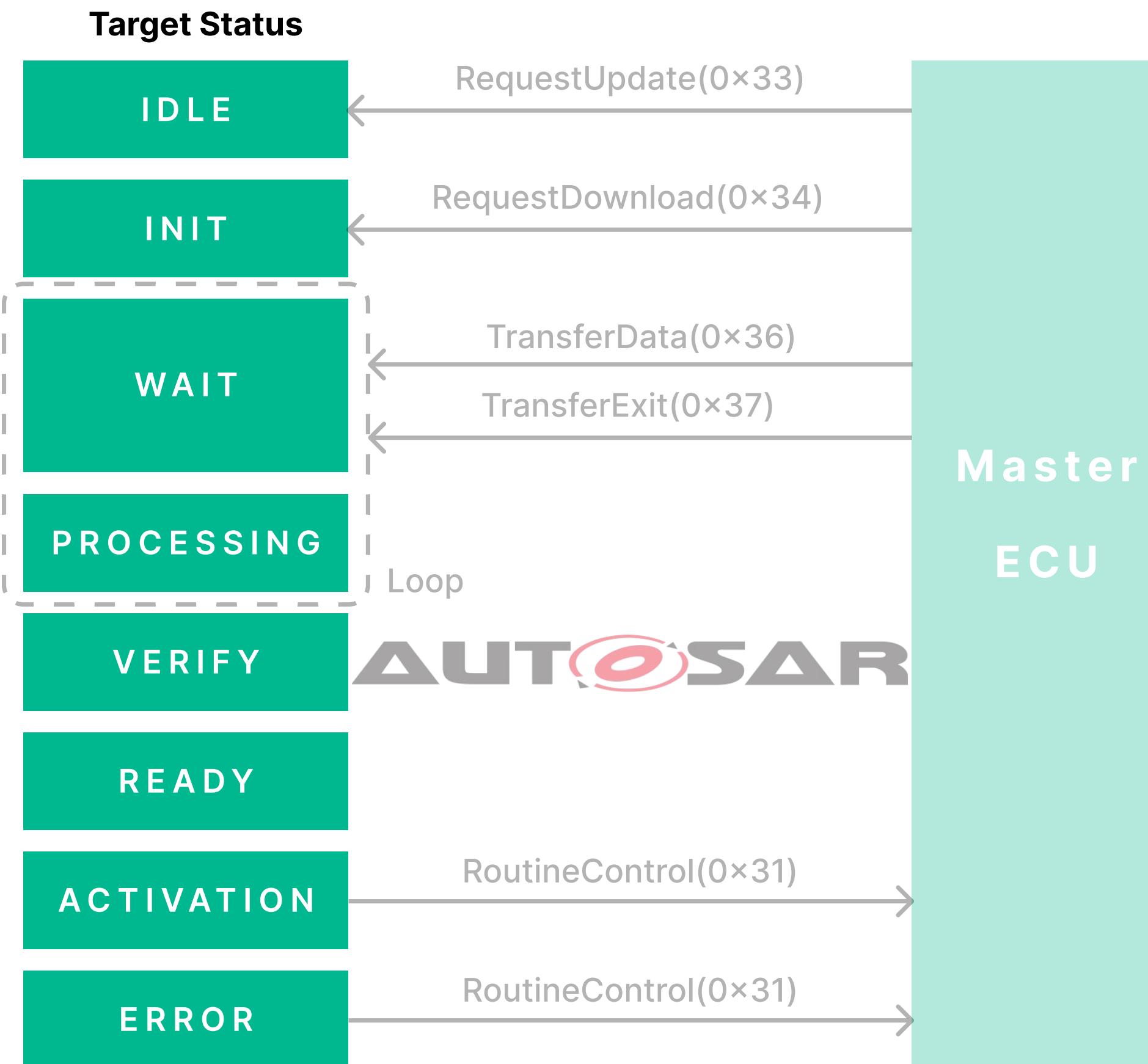
### Network Traffic Throughput



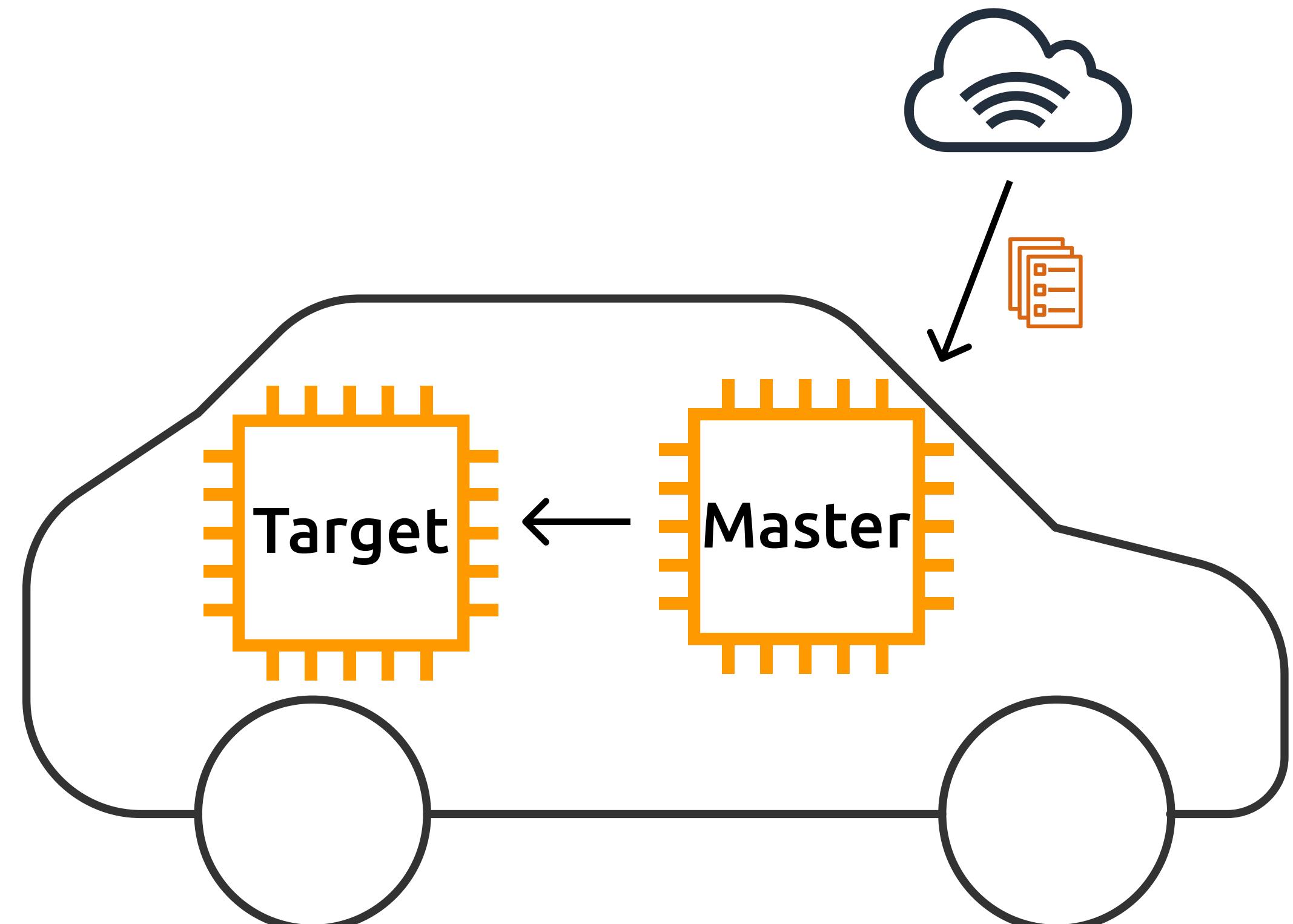
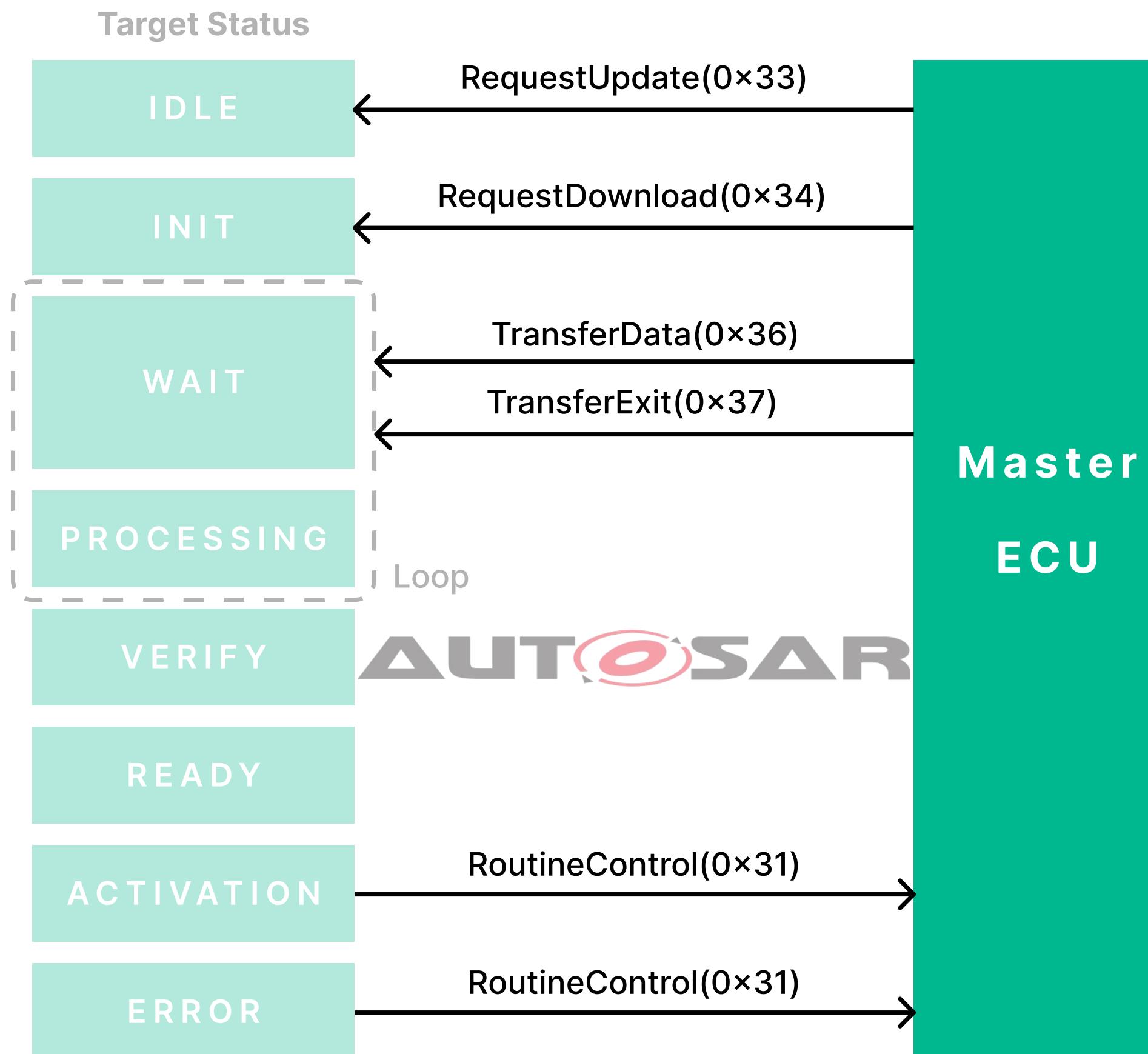
# Installation



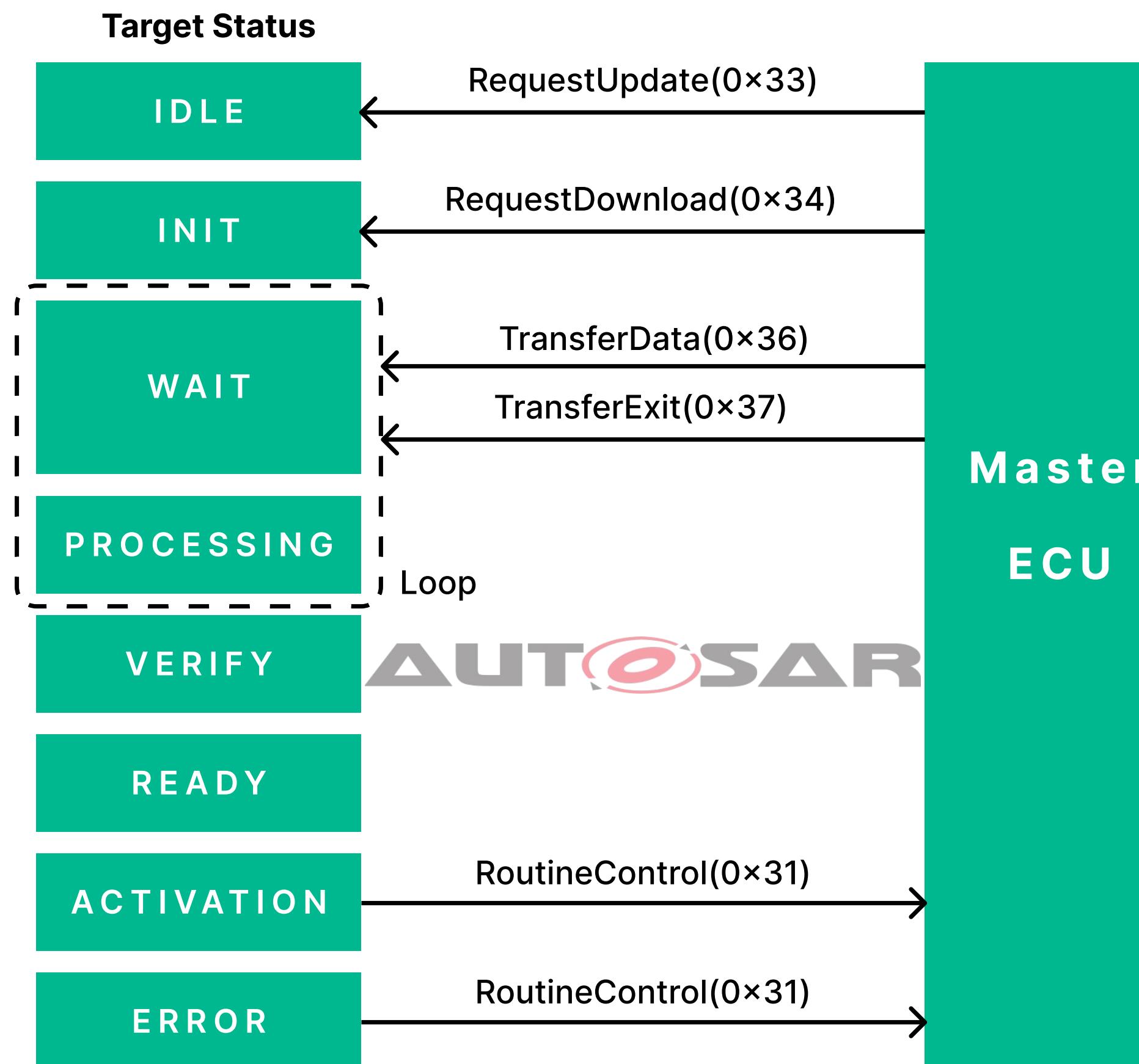
# Installation - Process



# Installation - Process



# Installation - Process



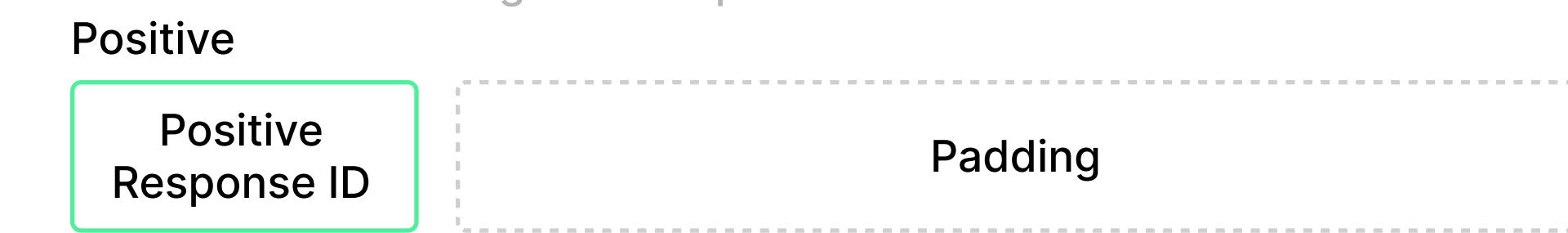
## Payload Frame based on UDS protocol

### Request



### Response

Positive Response ID = Service Identifier + 0x30  
 Negative Response ID = 0x7F



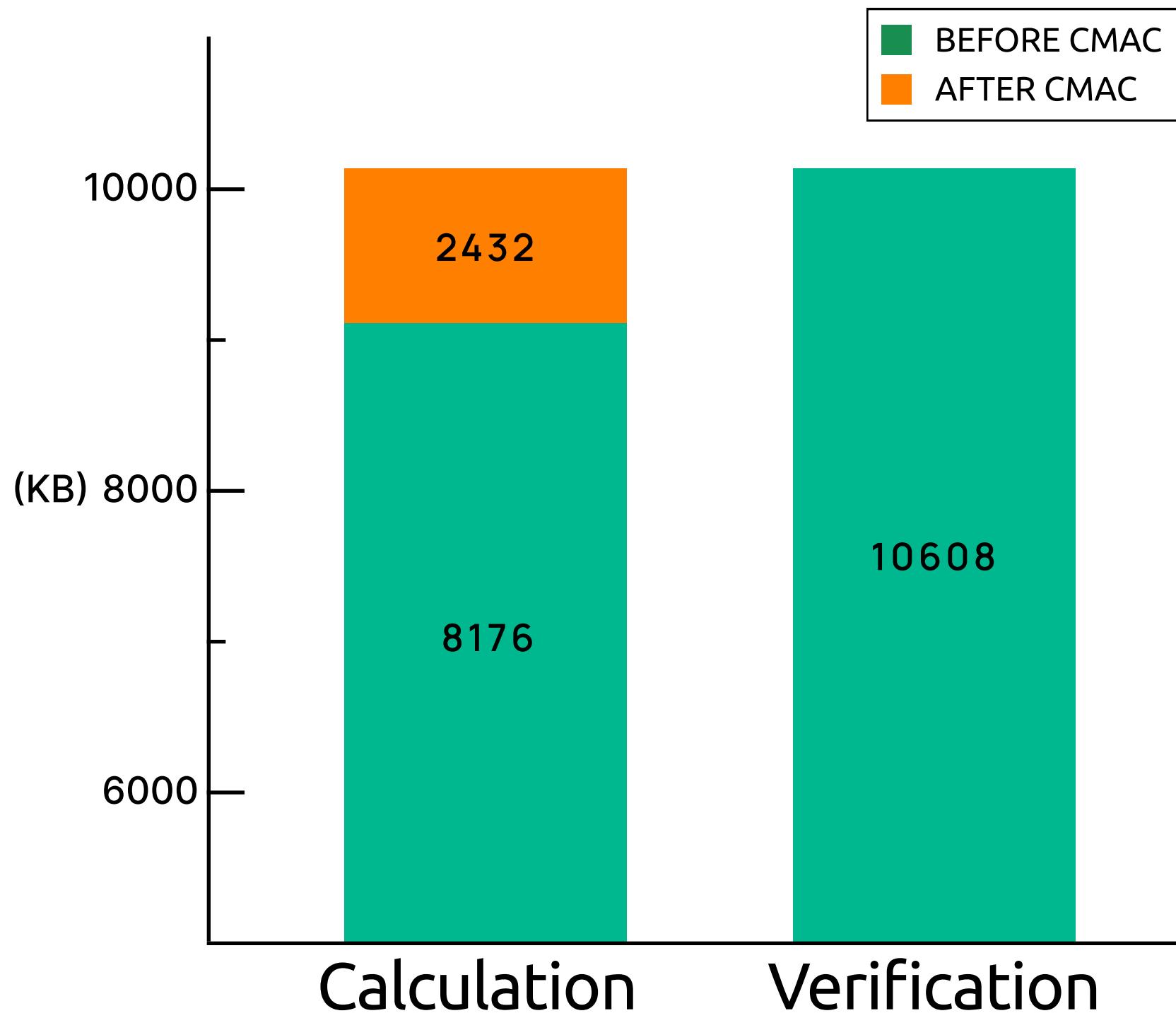
### Negative



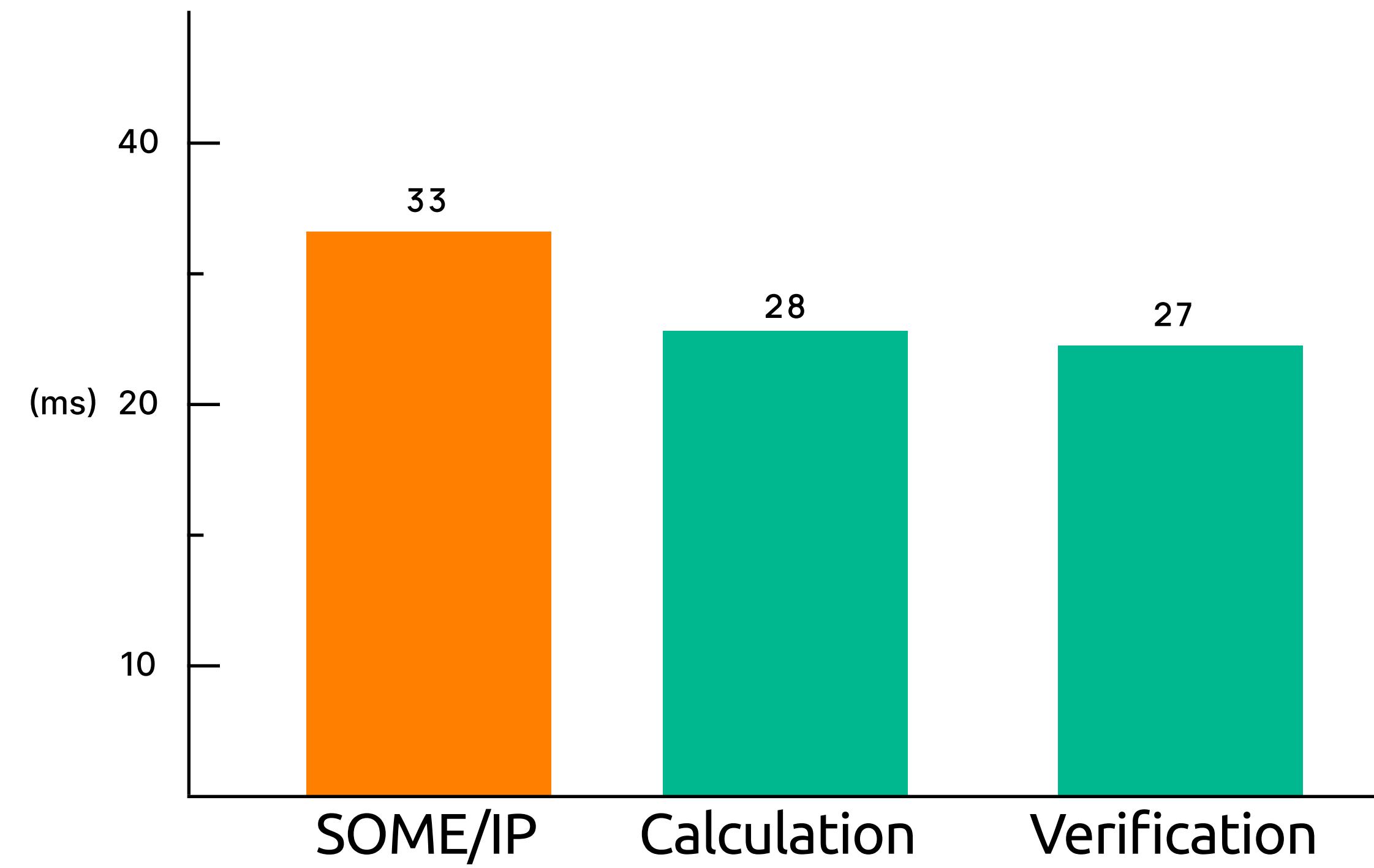
# Installation - Verification

## AES-CMAC

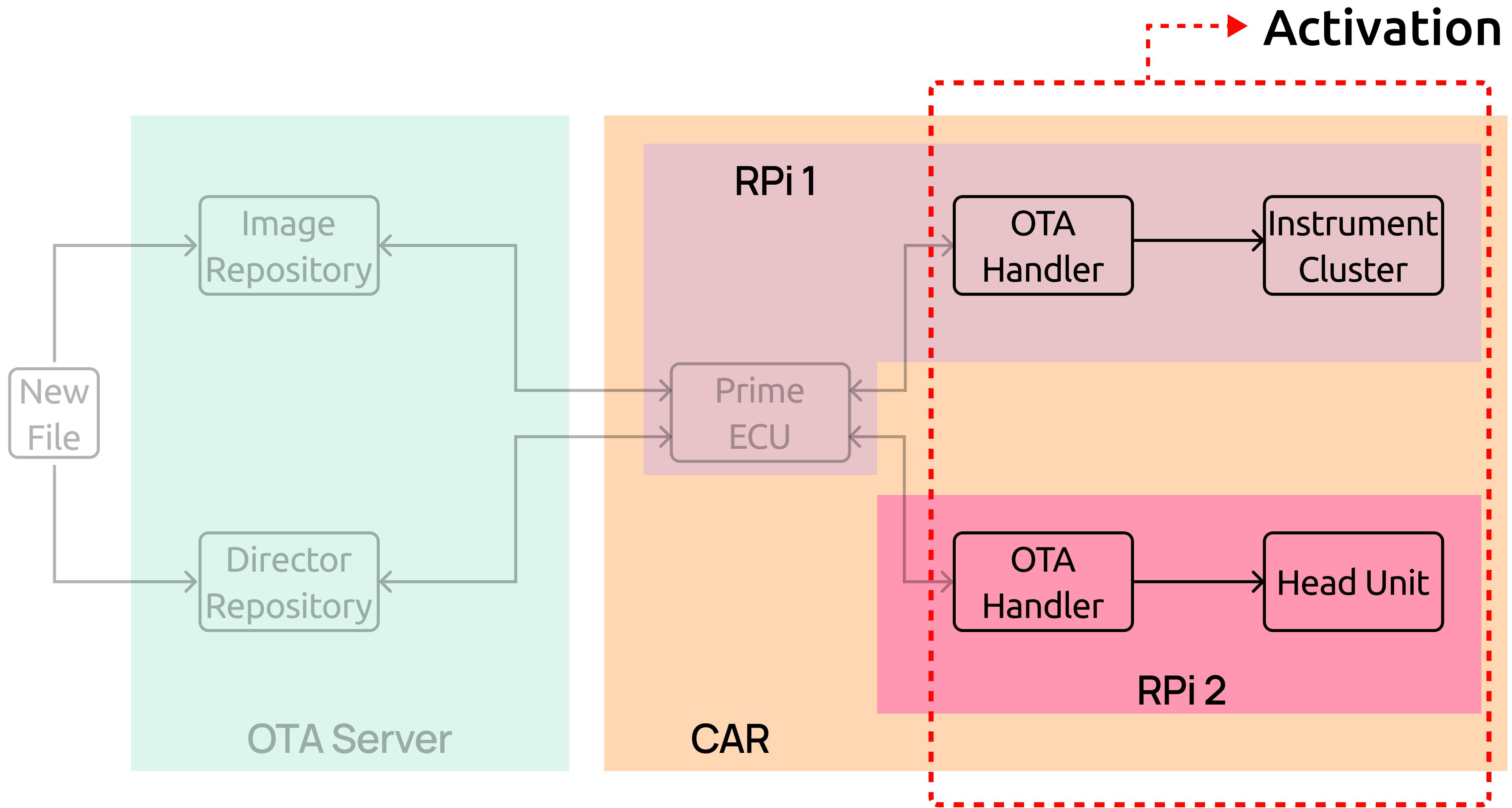
### Memory Usage



### Duration



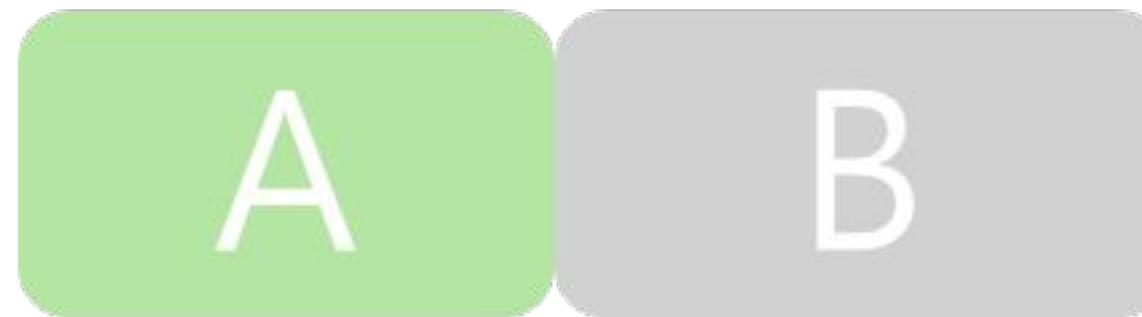
# Architecture



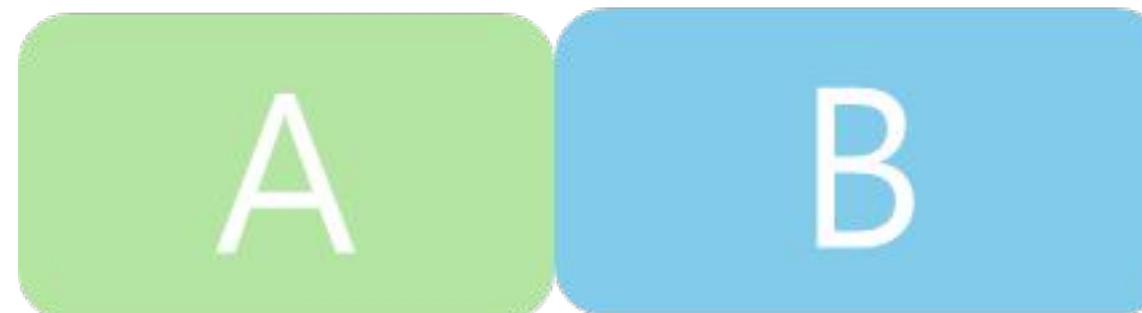
# Activation - A/B Updates

## Definition & Process

- Two identical system partitions (A and B)



- New software installs on inactive partition



- Partitions switch roles on success



### Enhanced Reliability

Seamless revert to previous version if update fails



### Improved User Experience

Minimal interruption, vehicle stays operational

# Attack Scenario 1 - Vulnerability



# Attack Scenario 1 - Vulnerability

Figure 1: Network traffic capture showing MQTT Publish messages from a client to a server. A red dashed arrow highlights the sequence of frames 196, 197, and 198.

The screenshot shows Wireshark displaying network traffic from a file named "qos2\_network\_off.pcapng". The packet list pane shows 333 total packets, with the last three highlighted by a red dashed arrow. The details pane shows the structure of a MQTT Publish message, and the bytes pane shows the raw hex and ASCII data of the selected frames.

No.	Time	Source	Destination	Protocol	Length	Info
187	19.660043076	192.168.86.227	192.168.86.255	UDP	43	59916 → 8765 Len=1
188	19.660049390	192.168.86.227	192.168.86.255	UDP	43	38807 → 8765 Len=1
189	20.099780656	3.160.39.26	192.168.86.22	TLSv1.2	248	Application Data
190	20.099813460	192.168.86.22	3.160.39.26	TCP	66	46662 → 443 [ACK] Seq=1 Ack=183 Win=14266 Len=0
191	20.284077491	192.168.86.43	192.168.86.22	MQTT	300	Publish Message (id=6) [permission/server]
192	20.284302639	192.168.86.22	192.168.86.43	MQTT	70	Publish Received (id=6)
193	20.297747919	192.168.86.43	192.168.86.22	TCP	66	59037 → 1883 [ACK] Seq=492 Ack=1071 Win=495 Len=0
194	20.297748440	192.168.86.43	192.168.86.22	MQTT	70	Publish Release (id=6)
195	20.297986341	192.168.86.22	192.168.86.43	MQTT	70	Publish Complete (id=6)
196	20.298032829	192.168.86.22	192.168.86.115	MQTT	298	Publish Message [permission/server]
197	20.354960435	192.168.86.43	192.168.86.22	TCP	66	59037 → 1883 [ACK] Seq=496 Ack=1075 Win=495 Len=0
198	20.532628133	192.168.86.22	192.168.86.115	TCP	298	[TCP Retransmission] 1883 → 33719 [PSH, ACK] Seq=1 Ack=1075 Win=495 Len=0
199	20.768618026	192.168.86.22	192.168.86.115	TCP	298	[TCP Retransmission] 1883 → 33719 [PSH, ACK] Seq=1 Ack=1075 Win=495 Len=0
200	21.236610719	192.168.86.22	192.168.86.115	TCP	298	[TCP Retransmission] 1883 → 33719 [PSH, ACK] Seq=1 Ack=1075 Win=495 Len=0
201	21.410785607	192.168.86.22	192.168.86.1	DNS	98	Standard query 0x2c9c A safebrowsing.googleapis.com
202	21.427081450	192.168.86.1	192.168.86.22	DNS	114	Standard query response 0x2c9c A safebrowsing.googleapis.com
203	21.427801917	192.168.86.22	192.168.86.1	DNS	98	Standard query 0xbe56 AAAA safebrowsing.googleapis.com
204	21.428129299	192.168.86.22	216.58.206.74	TCP	74	36658 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
205	21.441436705	216.58.206.74	192.168.86.22	TCP	74	443 → 36658 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
206	21.441477838	192.168.86.22	216.58.206.74	TCP	66	36658 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TS=1400000000000000000
207	21.442675369	192.168.86.22	216.58.206.74	TLSv1.3	2549	Client Hello
208	21.444658460	192.168.86.22	216.58.206.74	TLSv1.3	775	Change Cipher Spec, Application Data
209	21.446190313	192.168.86.1	192.168.86.22	DNS	126	Standard query response 0xbe56 AAAA safebrowsing.googleapis.com
210	21.454586300	216.58.206.74	192.168.86.22	TCP	66	443 → 36658 [ACK] Seq=1 Ack=1401 Win=267520 Len=0
211	21.459464864	216.58.206.74	192.168.86.22	TCP	66	443 → 36658 [ACK] Seq=1 Ack=2484 Win=266496 Len=0
212	21.459465267	216.58.206.74	192.168.86.22	TCP	66	443 → 36658 [ACK] Seq=1 Ack=3193 Win=265984 Len=0
213	21.465549062	216.58.206.74	192.168.86.22	TLSv1.3	1924	Server Hello, Change Cipher Spec, Application Data

Frame 196: 298 bytes on wire (2384 bits), 298 bytes captured (2384 bits) on interface wlp0s20f3, id 0

Ethernet II, Src: 64:d6:9a:9b:e4:2a (64:d6:9a:9b:e4:2a), Dst: c4:75:ab:83:69:e5 (c4:75:ab:83:69:e5)

Internet Protocol Version 4, Src: 192.168.86.22, Dst: 192.168.86.115

Transmission Control Protocol, Src Port: 1883, Dst Port: 33719, Seq: 1051, Ack: 1651, Len: 232

MQ Telemetry Transport Protocol, Publish Message

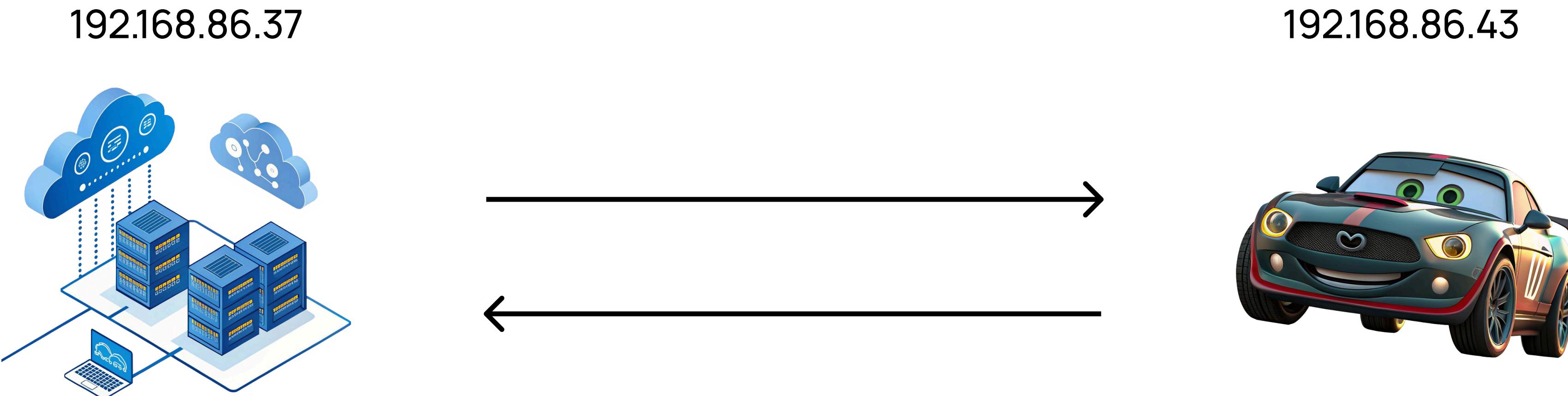
```

0000  c4 75 ab 83 69 e5 64 d6  9a 9b e4 2a 08 00 45 00  u.i.d.*.E.
0010  01 1c 77 b3 40 00 40 06  94 4e c0 a8 56 16 c0 a8  .w@.N.V...
0020  56 73 07 5b 83 b7 96 5e  2b f6 de f1 69 66 80 18  Vs[...+if...
0030  01 f5 2e e9 00 00 01 01  08 0a 85 97 e2 0f c6 0a  ...
0040  22 30 30 e5 01 00 11 70  65 72 6d 69 73 73 69 6f  "0...p ermissio
0050  6e 2f 73 65 72 76 65 72  7b 22 75 70 64 61 74 65  n/server {"update
0060  22 3a 20 74 72 75 65 2c  20 22 74 69 6d 65 73 74  ": true, "timest
0070  61 6d 70 22 3a 20 22 32  30 32 35 2d 30 36 2d 31  amp": "2 025-06-1
0080  32 54 31 31 3a 33 33 3a  33 30 2e 33 36 33 30 30  2T11:33: 30.36306

```

```
        "u·id": "1·*·E·  
        "w@@": "N·V·  
Vs": ["A·+·if·  
      ],  
      "00": permission/  
server: {"update": true,  
"timest  
amp": "2025-06-1  
2T11:33:30.36306
```

# ATT 1 - ARP Spoofing



# ATT 1 - ARP Spoofing

192.168.86.37



192.168.86.22



192.168.86.43



# ATT 1 - DNS Redirection

192.168.86.37



192.168.86.22



REQUEST

`http://192.168.86.37`

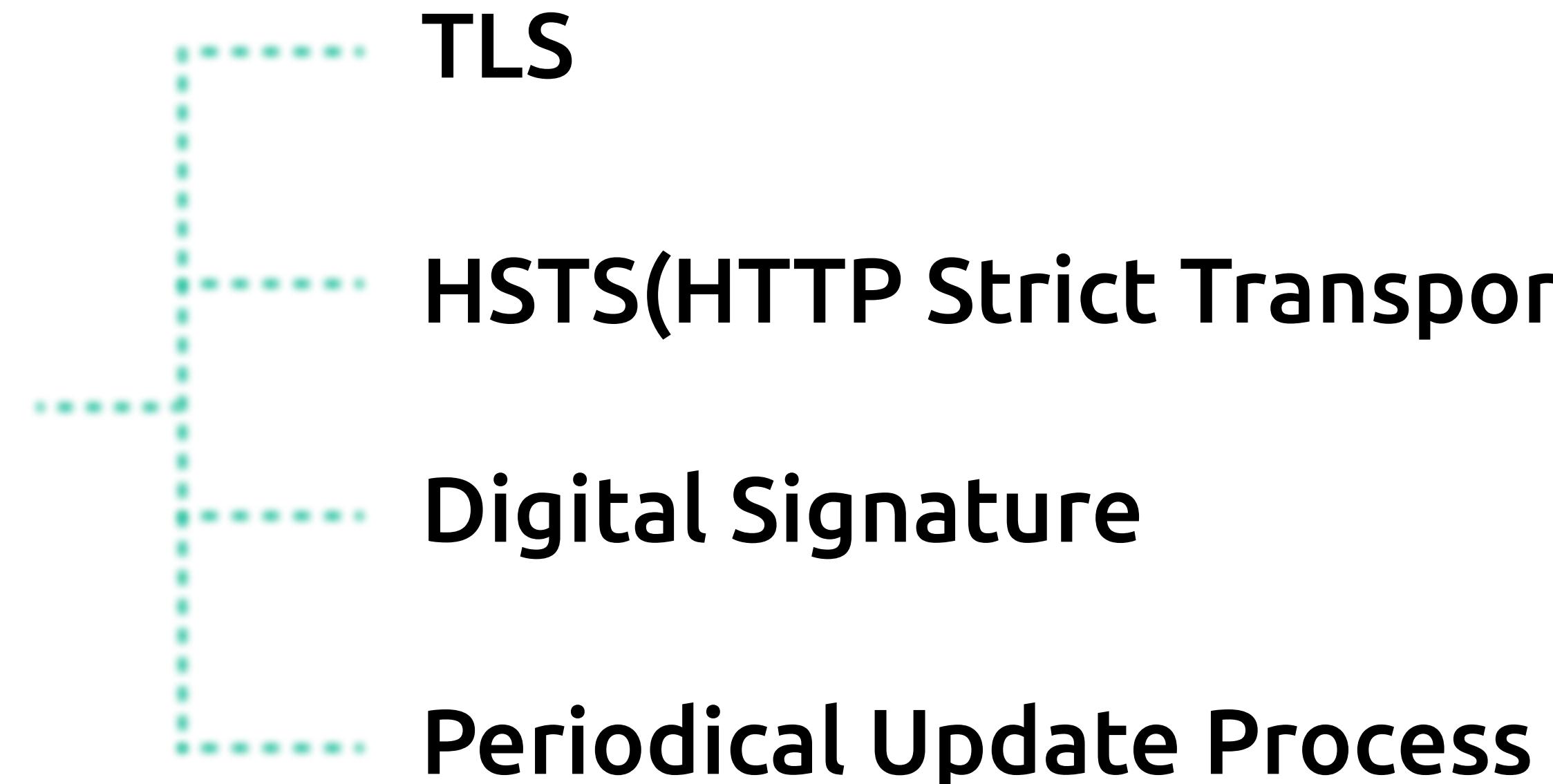
GET

`http://192.168.86.22`

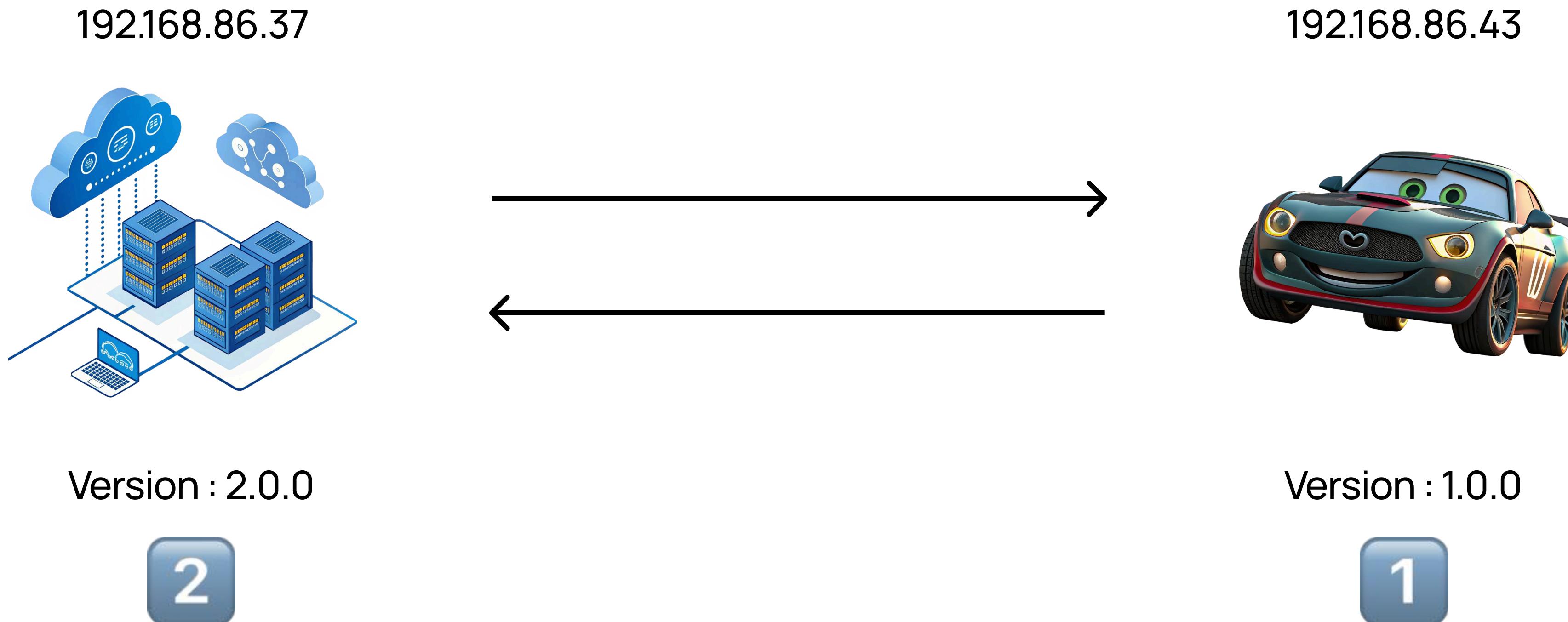
192.168.86.43



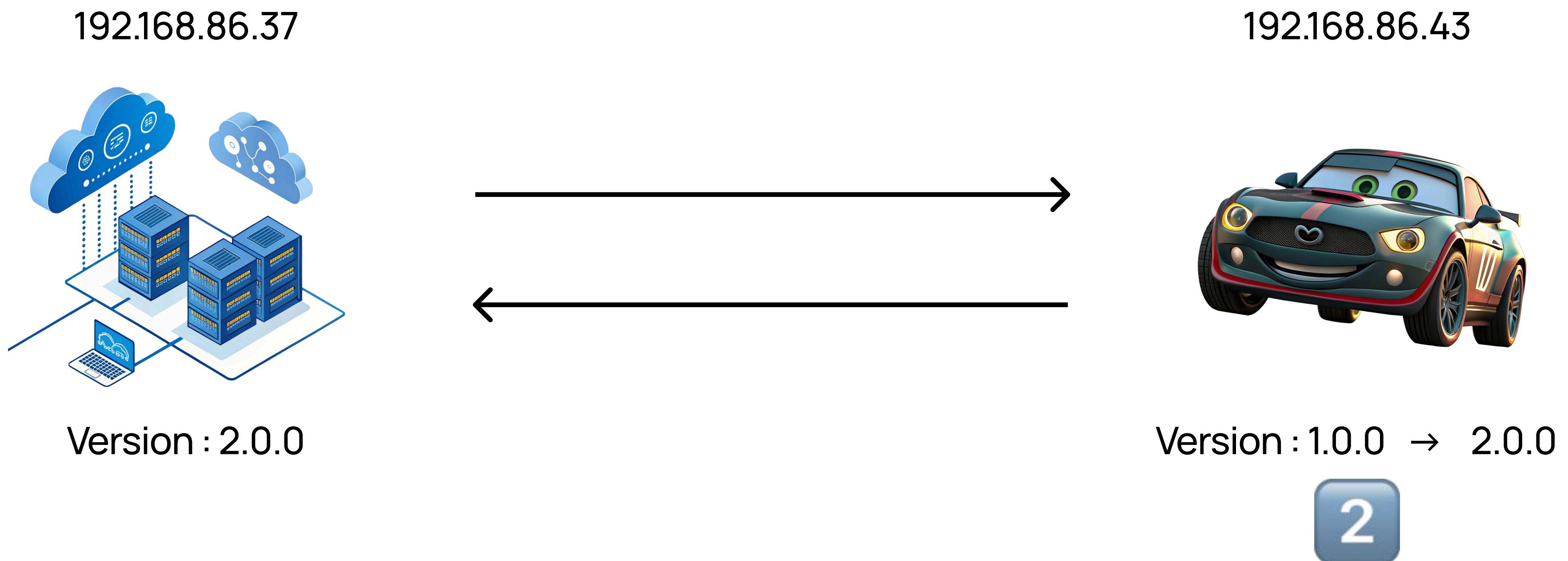
# ATT 1 - How to protect



# ATT 2 - Original Update Scenario



# ATT 2 - Original Update Scenario



# ATT 2 - Precondition



If attacker knows...

- Broker & Client IP address
- TLS certification
- Update Sequence

# ATT 2 - Capture Message

192.168.86.37



Version : 2.0.0

192.168.86.22



Version : 2.0.0



192.168.86.43



Version : 1.0.0 → 2.0.0

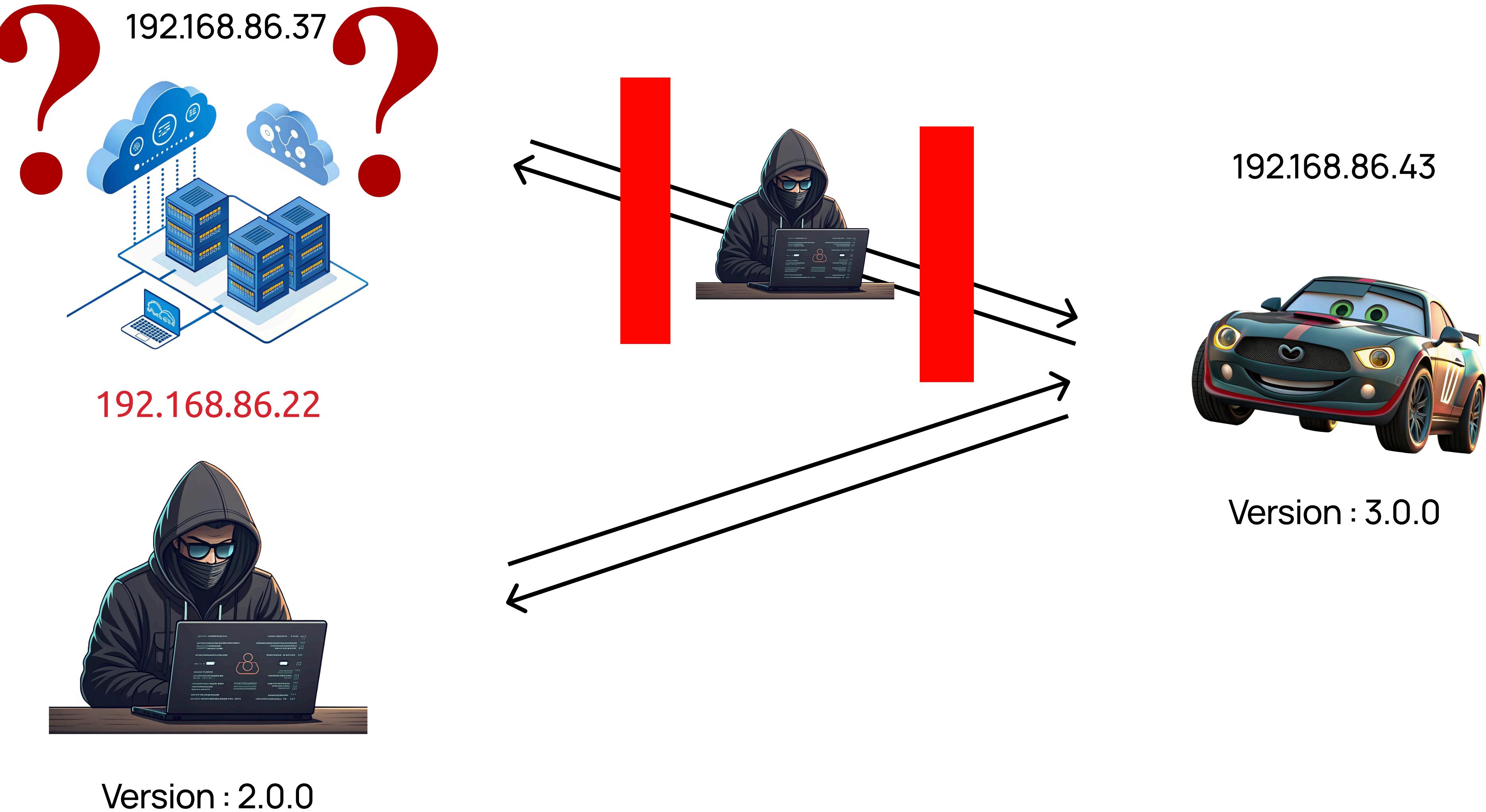


# ATT 2 - Replay attack

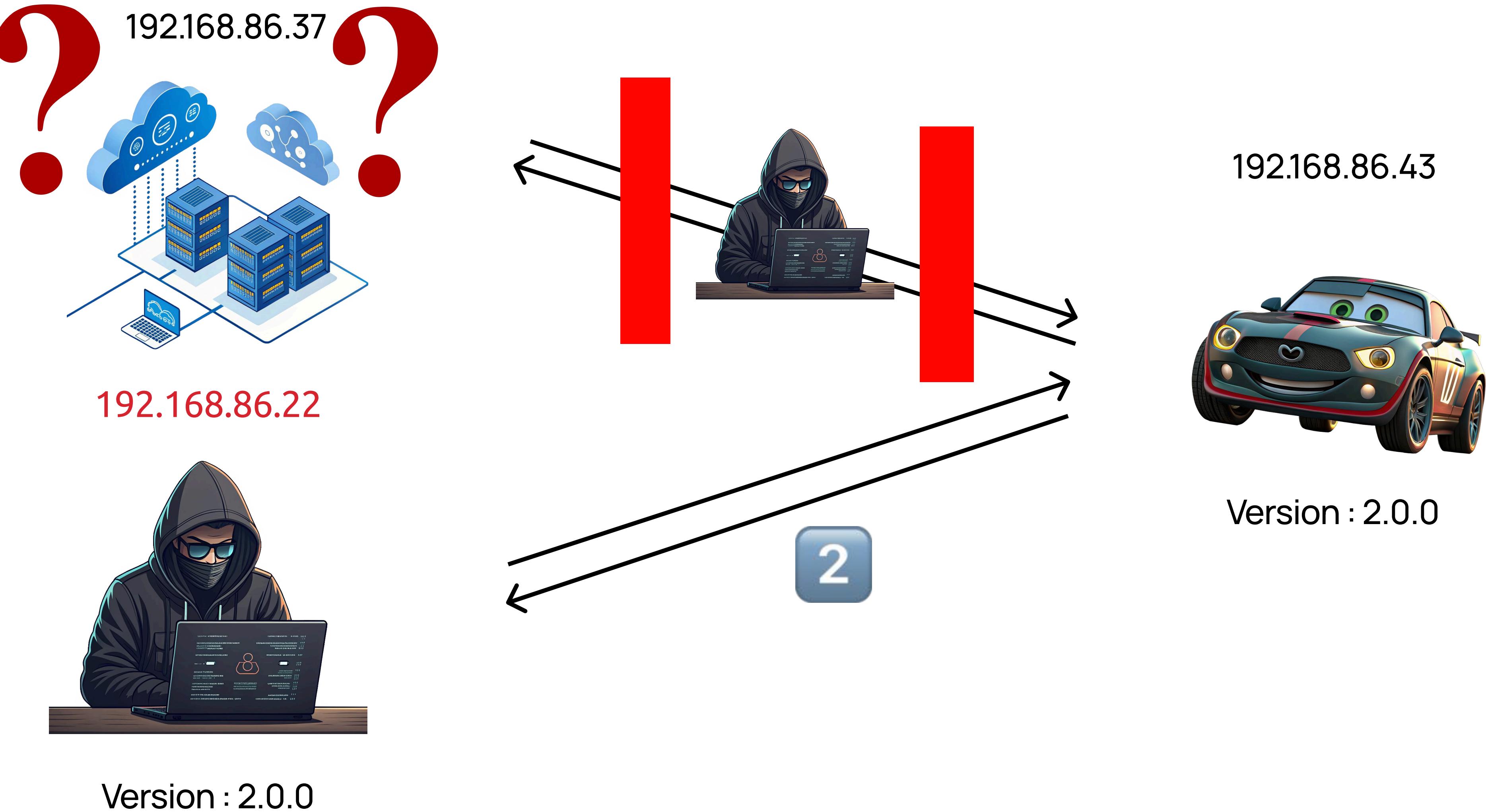


Find some vulnerability in version 2.0.0

## ATT 2 - Replay attack



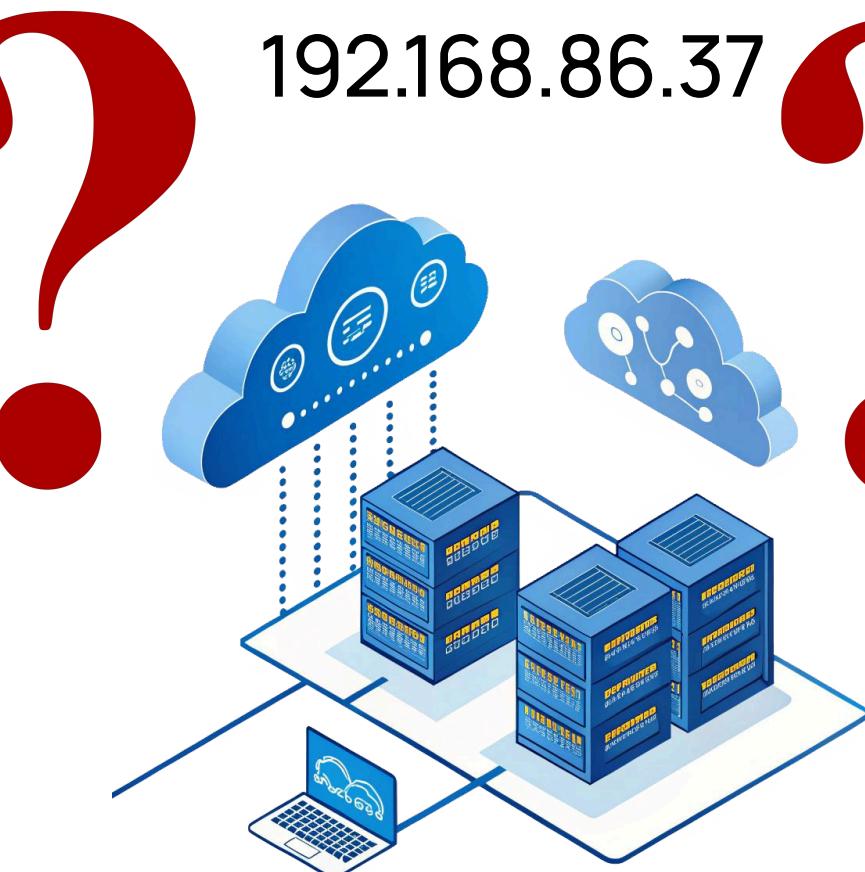
## ATT 2 - Replay attack



# ATT 2 - Replay attack

demo video

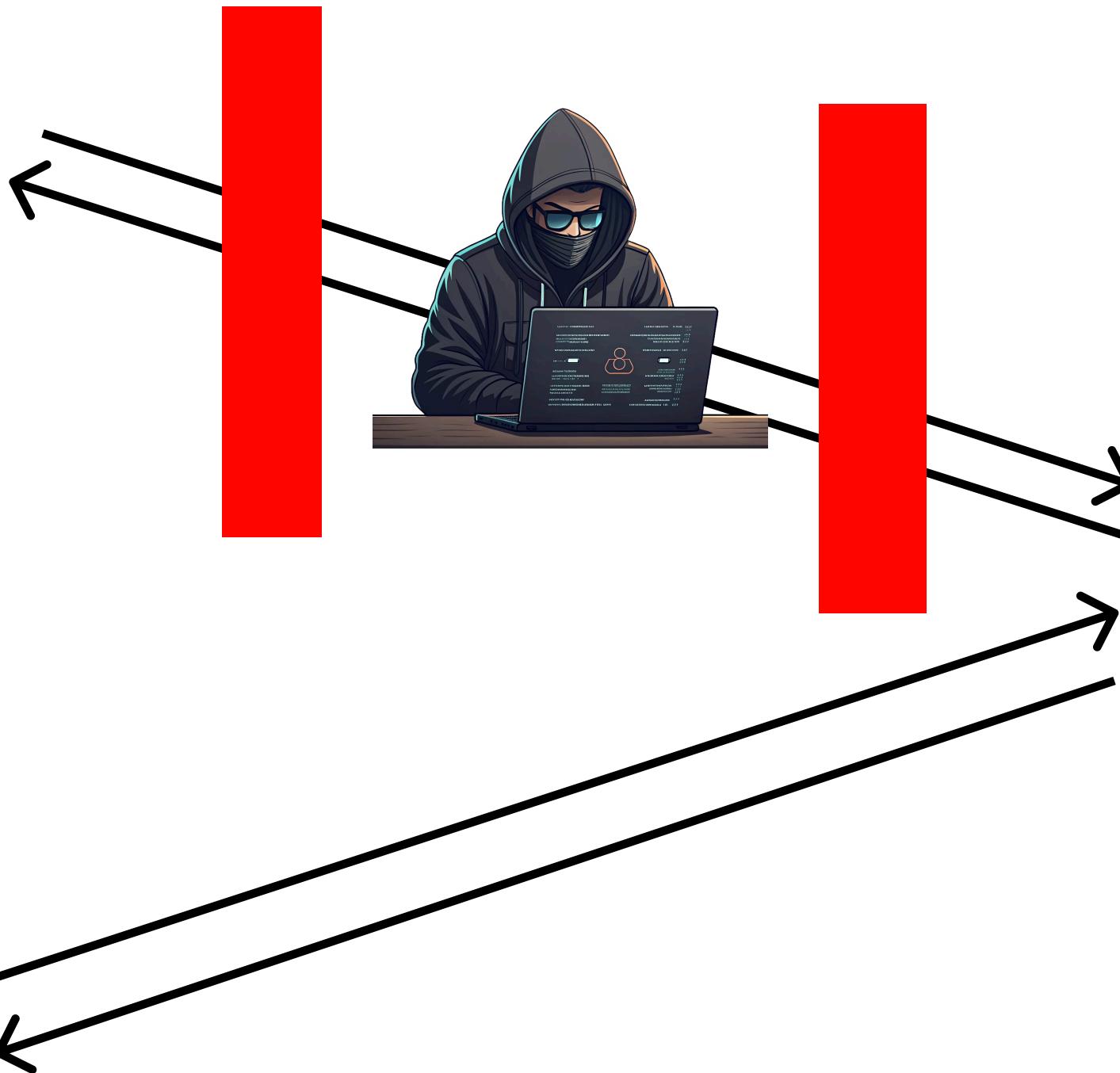
## ATT 2 - How to protect



192.168.86.22



Version : 2.0.0



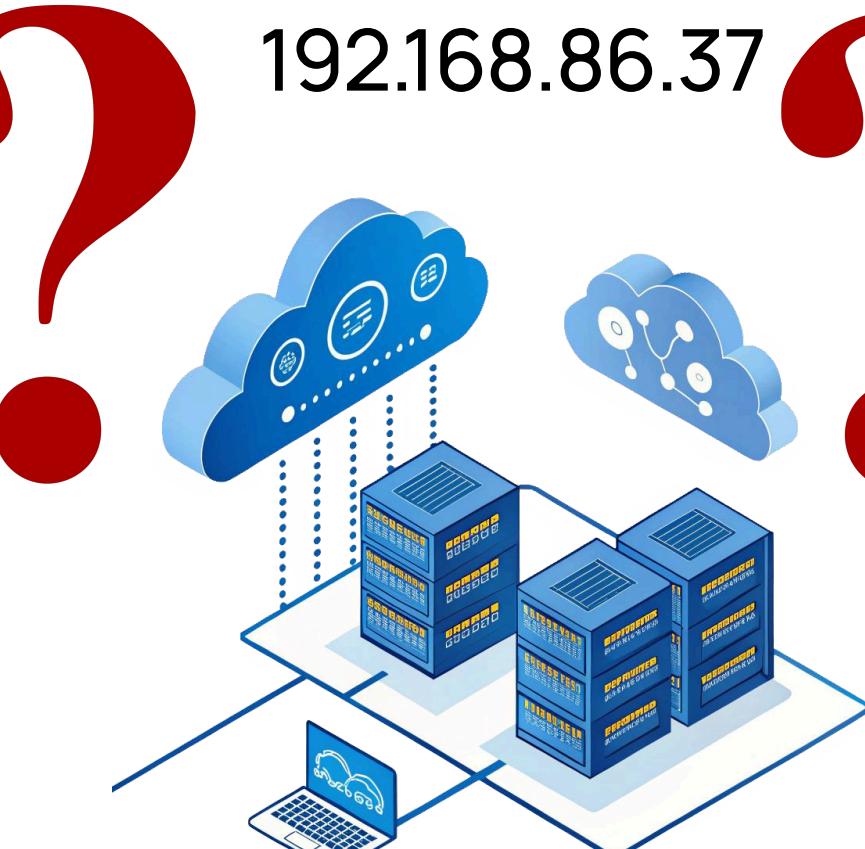
192.168.86.43



Version : 3.0.0

Timestamp

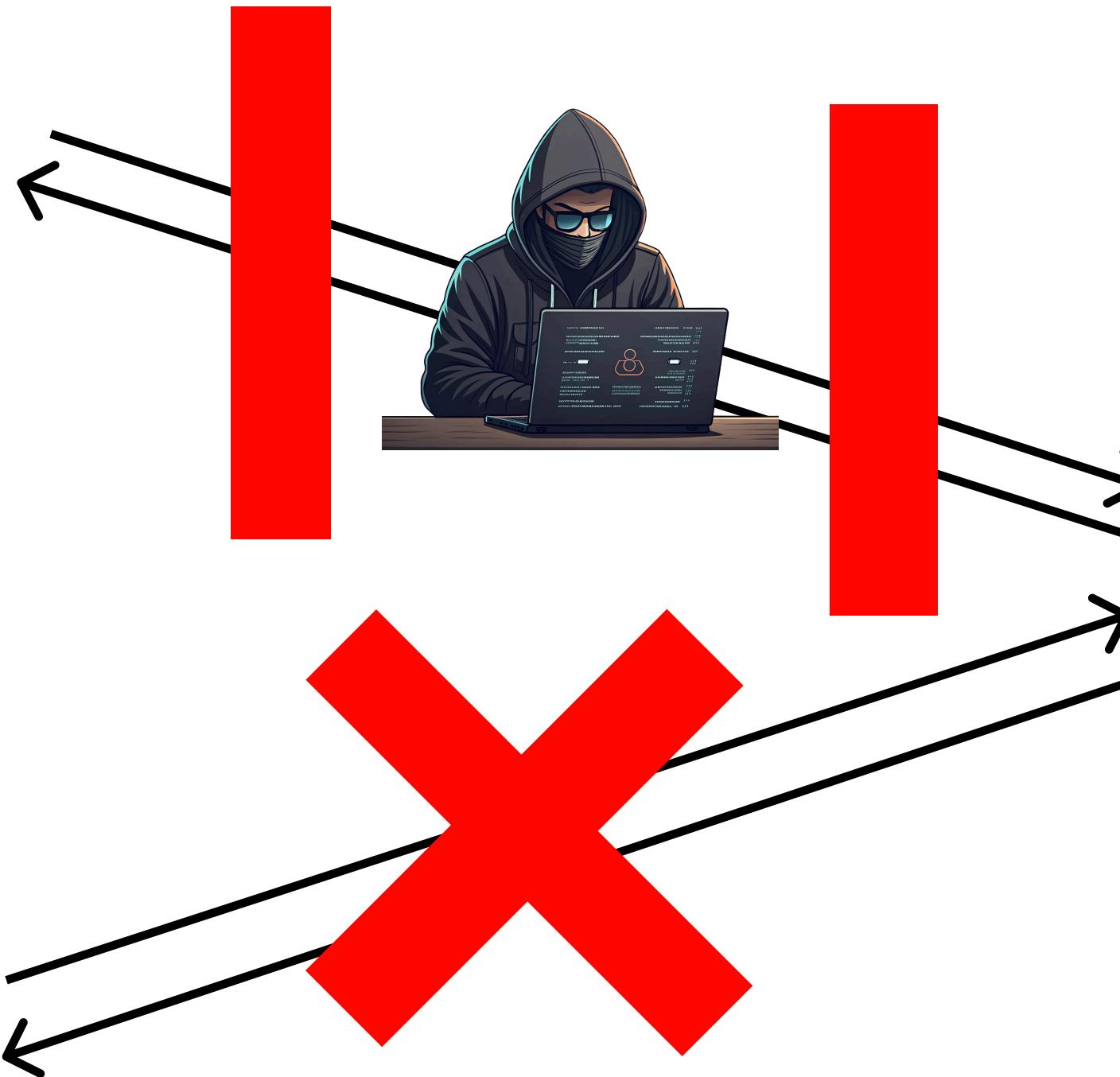
## ATT 2 - How to protect



192.168.86.22



Version : 2.0.0



192.168.86.43



Version : 3.0.0

**Timestamp**

# THANK YOU

# seahme