# Secure and Authorized Client-to-Client Communication for LwM2M

Leandro Lanzieri [1], Peter Kietzmann [1], Thomas C. Schmidt [1], Matthias Wählisch [2]

leandro.lanzieri@haw-hamburg.de

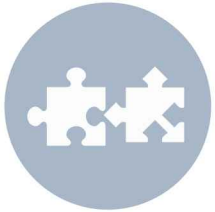
**RIPE 84**

18th May 2022 - Berlin, Germany

HAW HAMBURG

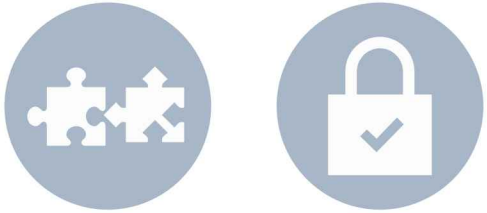
Freie Universität Berlin

# IoT Challenges



**Vendor Incompatibility**

# IoT Challenges



Vendor Incompatibility



Security Requirements

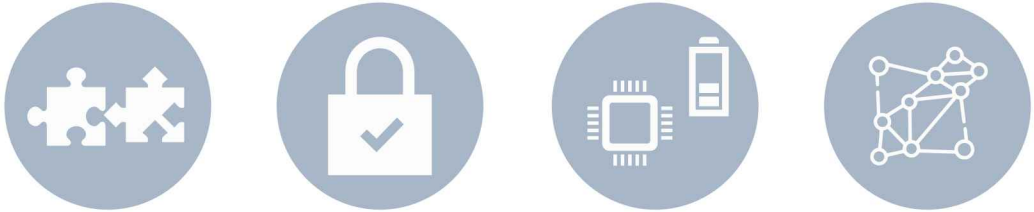
# IoT Challenges

Vendor Incompatibility

Security Requirements

Device Constraints

# IoT Challenges



Vendor Incompatibility

Security Requirements

Device Constraints

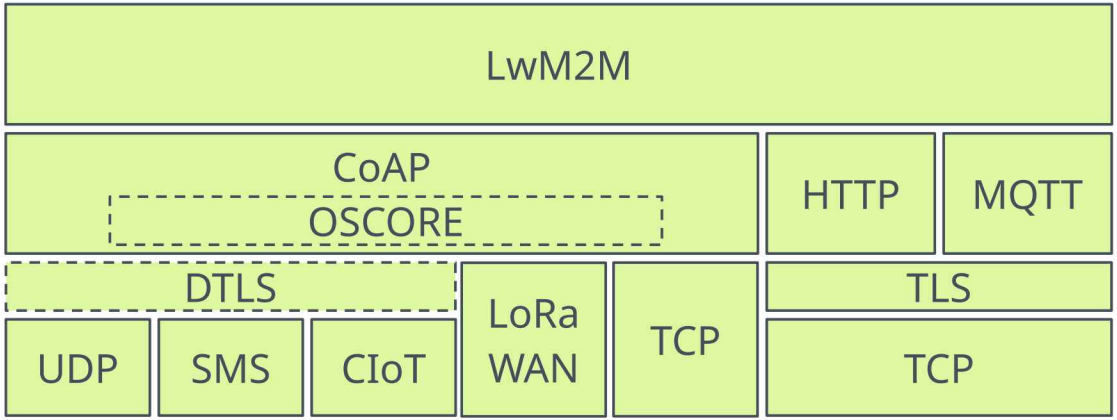
Edge Collaboration

# IoT Challenges

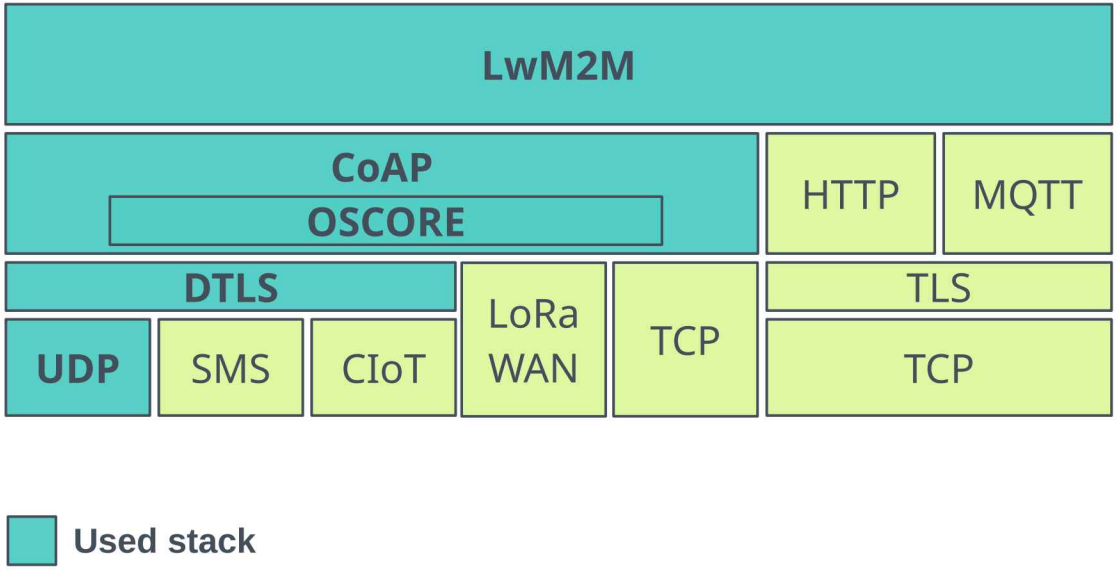
# IoT Challenges

# LwM2M Overview

# LwM2M Protocol Stack

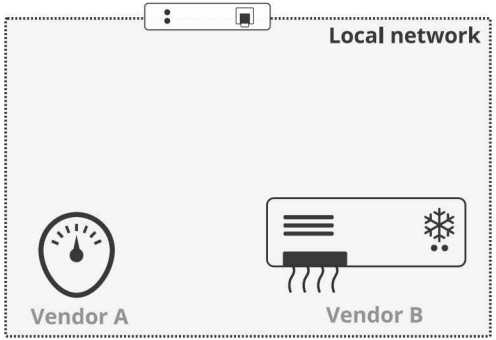


# LwM2M Protocol Stack



# LwM2M Background

- **IoT Device management**
  - Semantic interoperability across vendors
  - Resource access control
  - Bootstrapping and software updates
- **Two main entity types**
  - LwM2M Clients (IoT devices)
  - LwM2M Servers
- **Only Servers perform operations on Clients**
  - Using established secure communication
  - Require credentials and access rights
  - IoT applications interact with Clients only through Servers

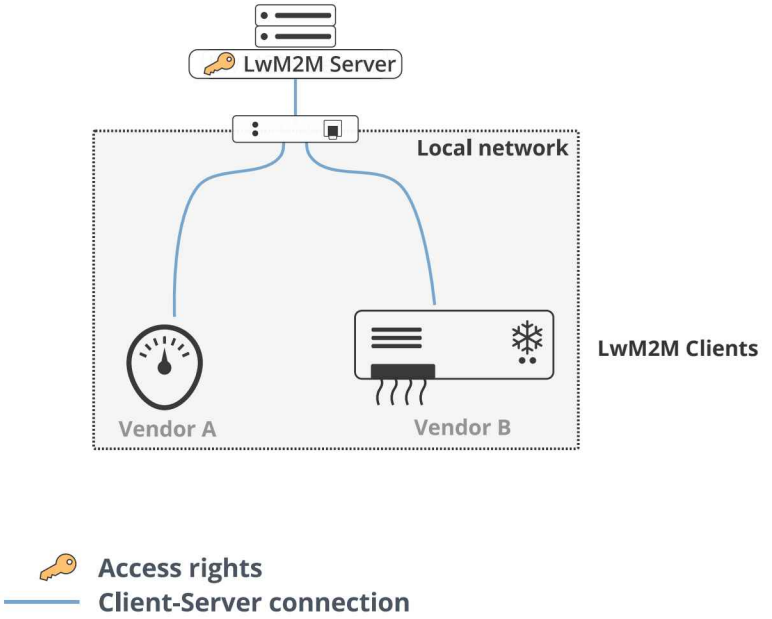


# LwM2M Background

- **IoT Device management**
  - Semantic interoperability across vendors
  - Resource access control
  - Bootstrapping and software updates

- **Two main entity types**
  - LwM2M Clients (IoT devices)
  - LwM2M Servers

- **Only Servers perform operations on Clients**
  - Using established secure communication
  - Require credentials and access rights
  - IoT applications interact with Clients only through Servers

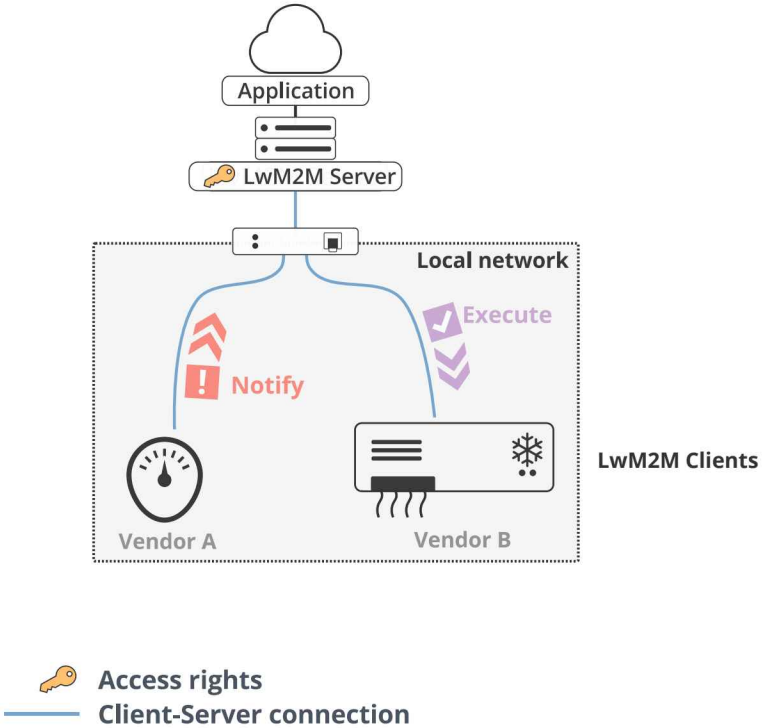


# LwM2M Background

- **IoT Device management**
  - Semantic interoperability across vendors
  - Resource access control
  - Bootstrapping and software updates

- **Two main entity types**
  - LwM2M Clients (IoT devices)
  - LwM2M Servers

- **Only Servers perform operations on Clients**
  - Using established secure communication
  - Credentials and access rights are required
  - IoT applications interact with Clients only through Servers



# LwM2M Background

- **IoT Device management**
  - Semantic interoperability across vendors

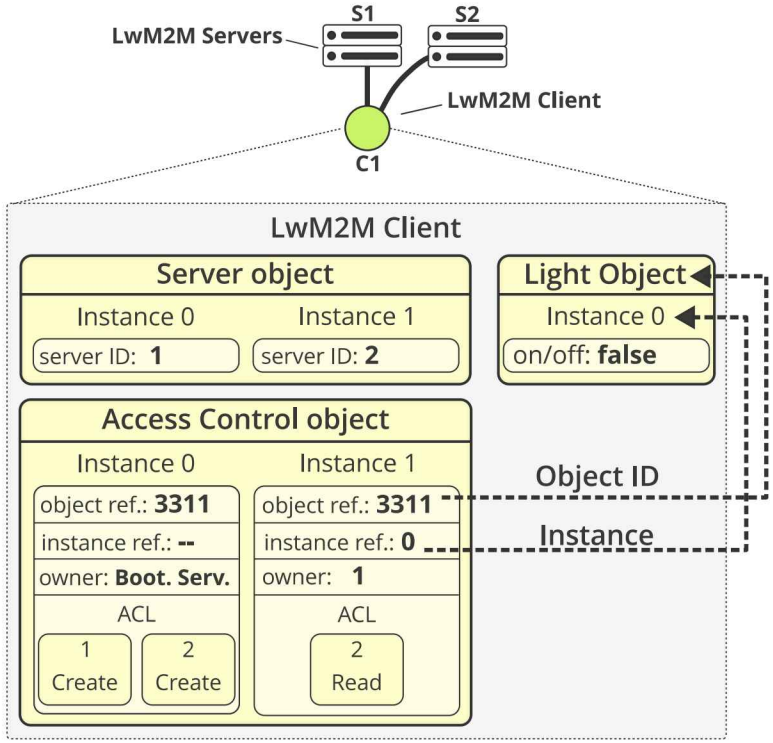

Server-Centric communication **prevents edge collaboration**.

- **Only Servers perform operations on Clients**
  - Using established secure communication
  - Credentials and access rights are required
  - IoT applications interact with Clients only through Servers

Access rights
Client-Server connection

# LwM2M Objects, Resources and Access Control

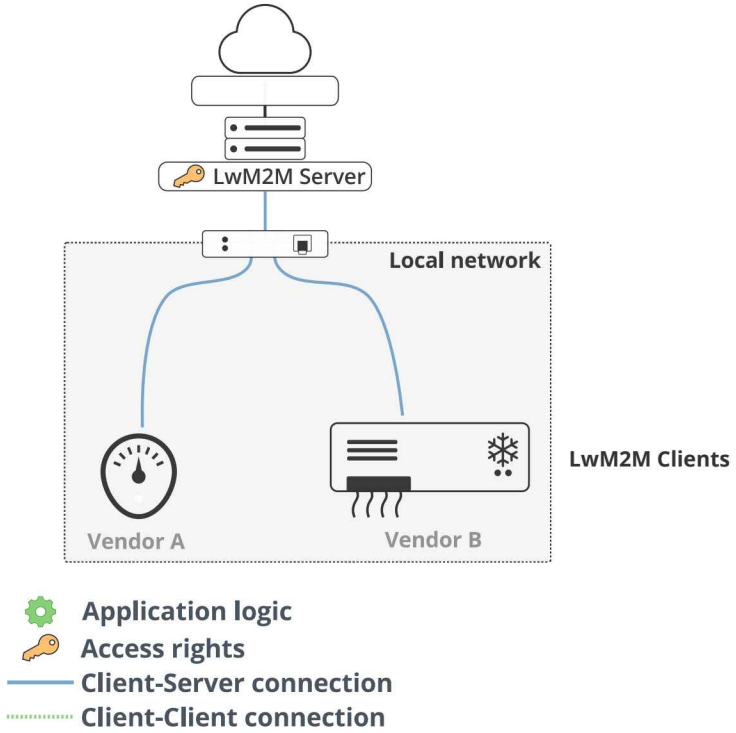
- **LwM2M Clients expose resources**
  - Resources are logically grouped into objects
- **Objects accept multiple operations**
  - Read, write, execute, create, etc.
- **Access control policies apply to objects**
  - Determine which operations a server may perform
  - Different servers may have different access



# LwM2M Client-to-Client (C2C) Communcation

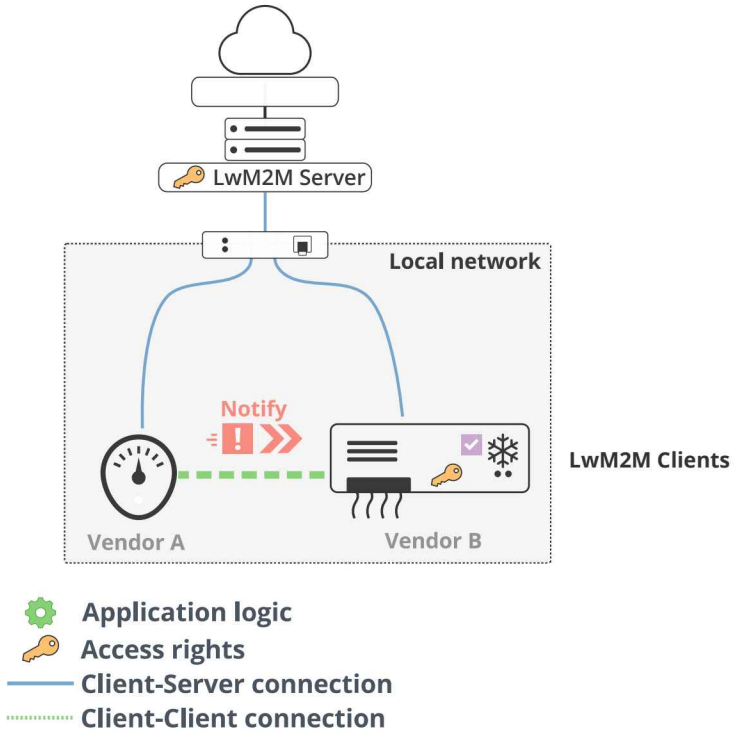
# Extended LwM2M Deployment

- Clients operate on other Clients resources
- Application logic is distributed on the edge
  - Reduces latency
  - Increases bandwidth
  - Local communication
- Servers monitor and manage Clients



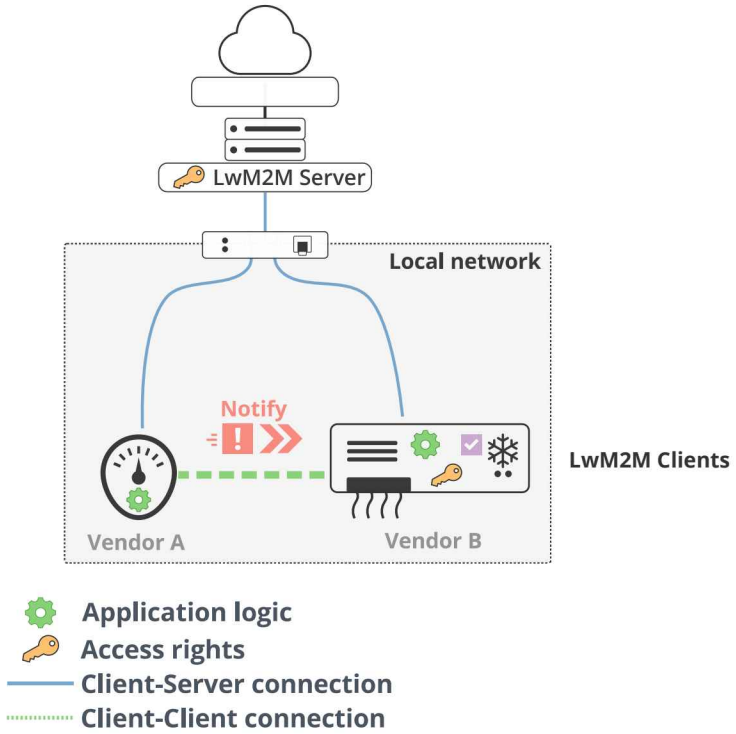
# Extended LwM2M Deployment

- Clients operate on other Clients resources
- Application logic is distributed on the edge
  - Reduces latency
  - Increases bandwidth
  - Local communication
- Servers monitor and manage Clients



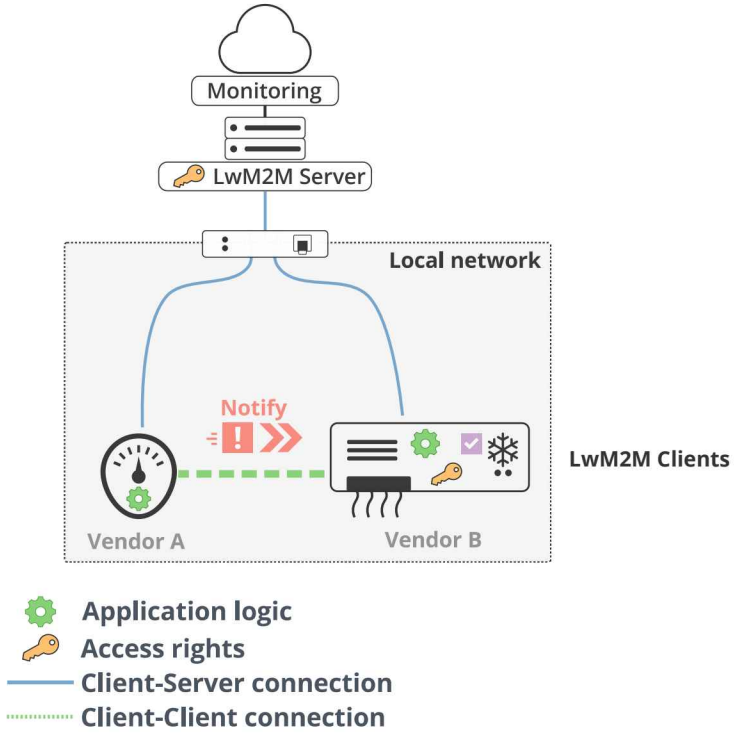
# Extended LwM2M Deployment

- Clients operate on other Clients resources
- **Application logic is distributed on the edge**
  - Reduces latency
  - Increases bandwidth
  - Local communication
- Servers monitor and manage Clients



# Extended LwM2M Deployment

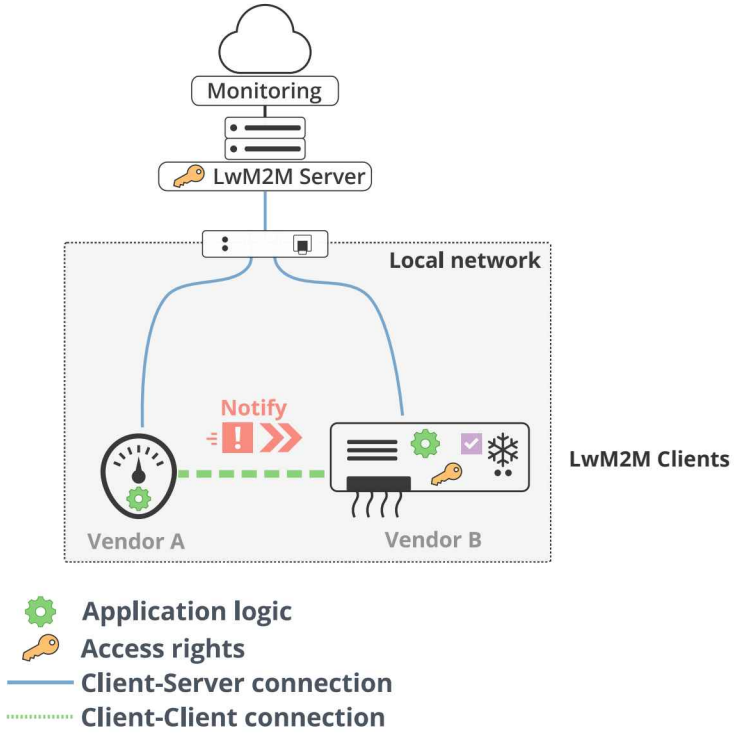
- Clients operate on other Clients resources
- Application logic is distributed on the edge
  - Reduces latency
  - Increases bandwidth
  - Local communication
- **Servers monitor and manage Clients**



# Extended LwM2M Deployment

- Clients operate on other Clients resources
- Application logic is distributed on the edge
  - Reduces latency
  - Increases bandwidth
  - Local communication
- Servers monitor and manage Clients

- **New LwM2M objects**
  - **Client** object: communication information
  - **Client Security** object: credentials for secure channel
  - **Client Access Control** object: remote clients access rights
- **Extended interfaces**
  - Allow client operation
  - Handle client access rights

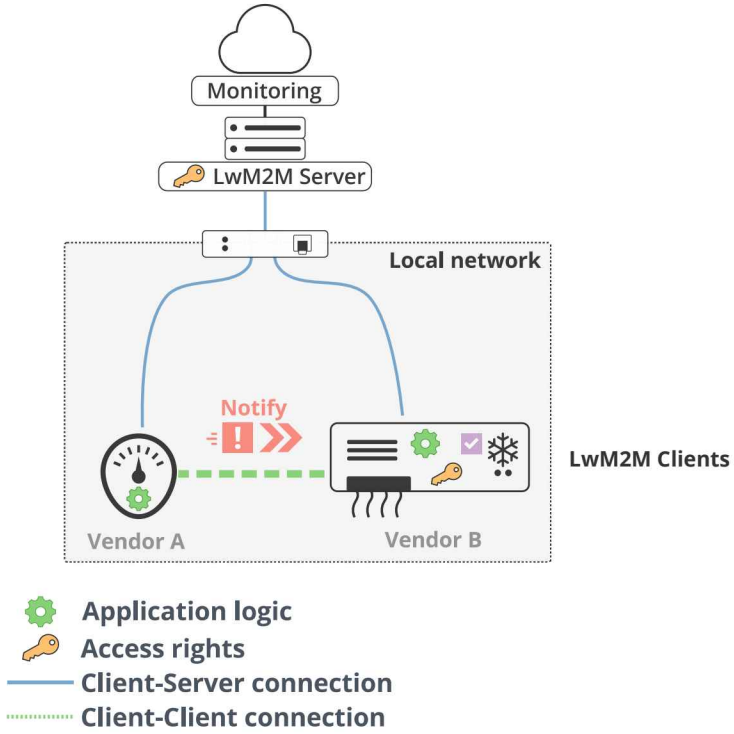


# Extended LwM2M Deployment

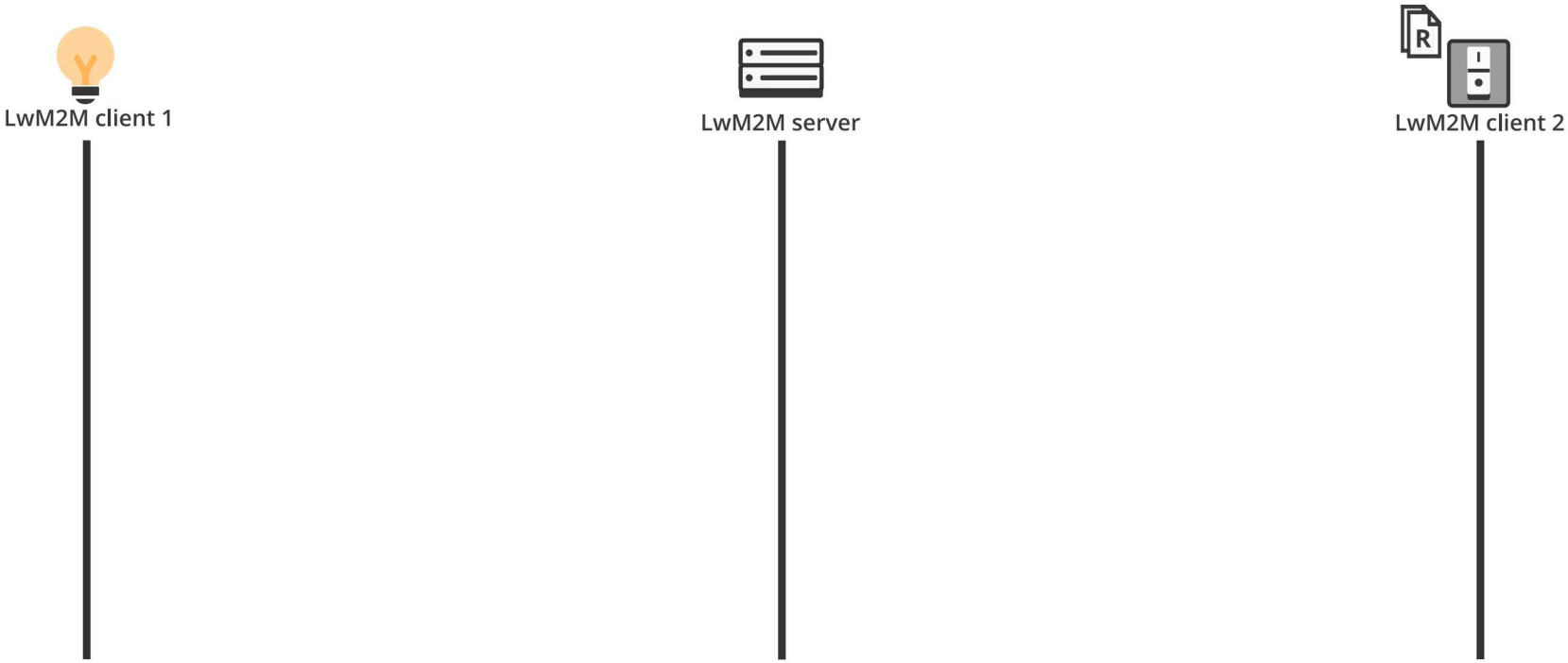
- Clients operate on other Clients resources
- Application logic is distributed on the edge
  - Reduces latency
  - Increases bandwidth
  - Local communication
- Servers monitor and manage Clients

- **New LwM2M objects**
  - **Client** object: communication information
  - **Client Security** object: credentials for secure channel
  - **Client Access Control** object: remote clients access rights
- **Extended interfaces**
  - Allow client operation
  - Handle client access rights

**Contribution 1**

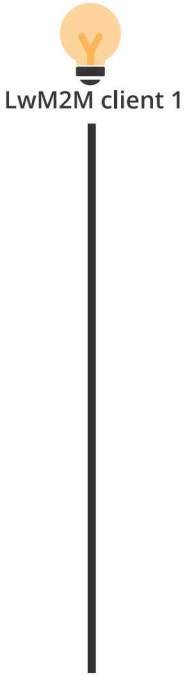


# Third Party Authorization of LwM2M Clients

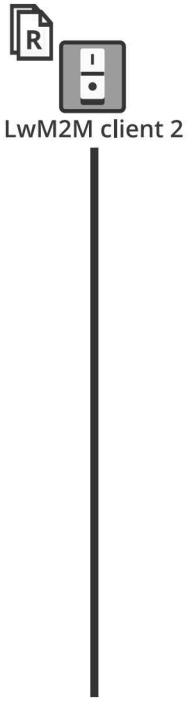


# Third Party Authorization of LwM2M Clients

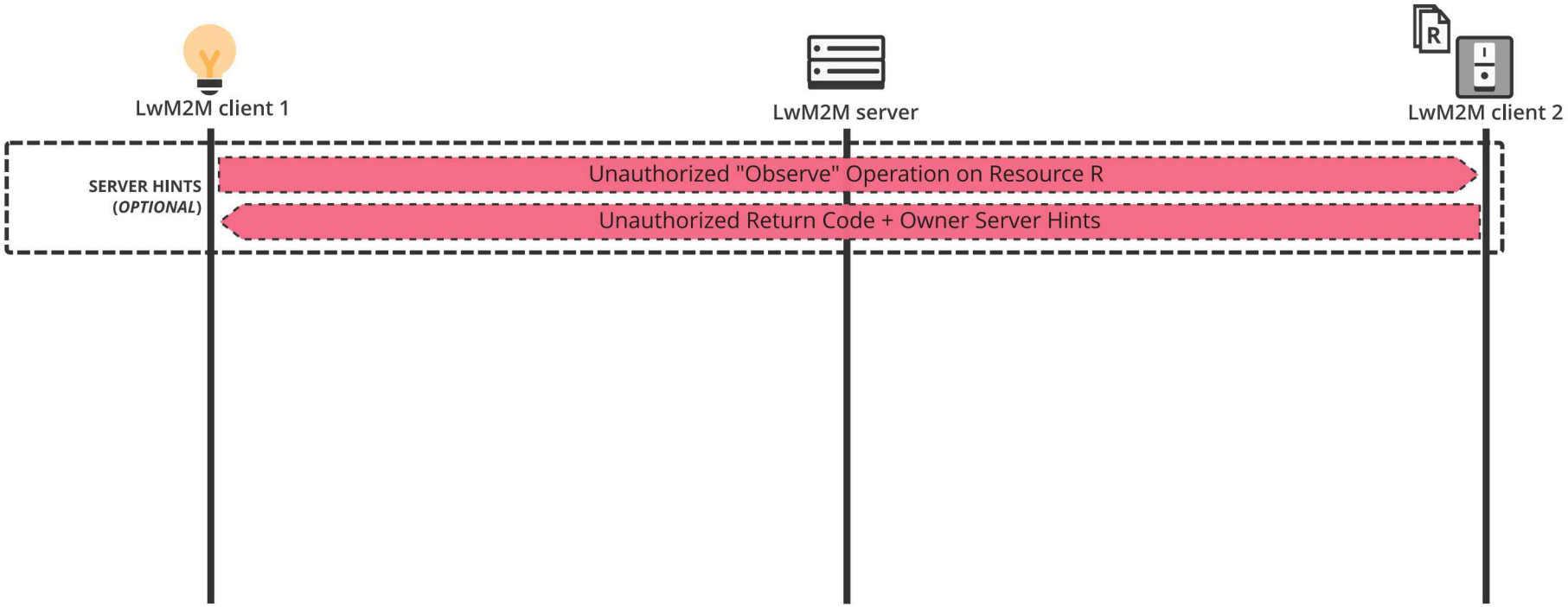
## Contribution 2



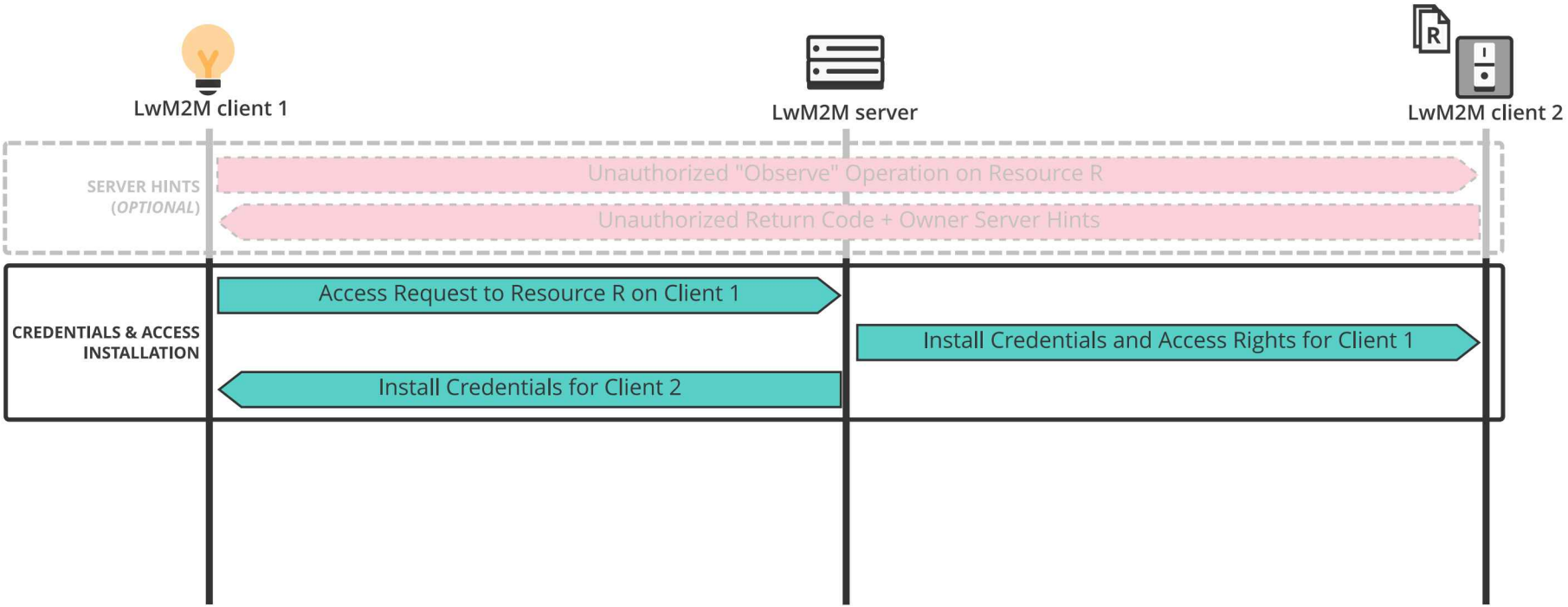




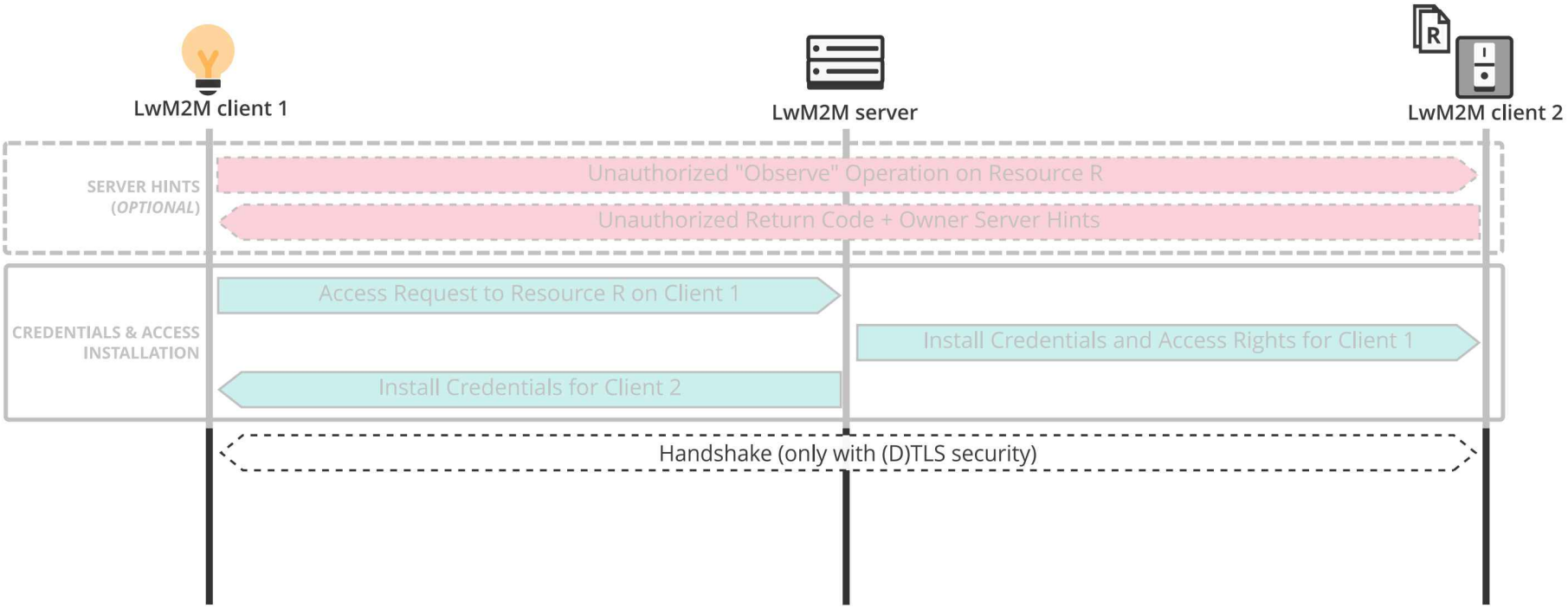
# Third Party Authorization of LwM2M Clients



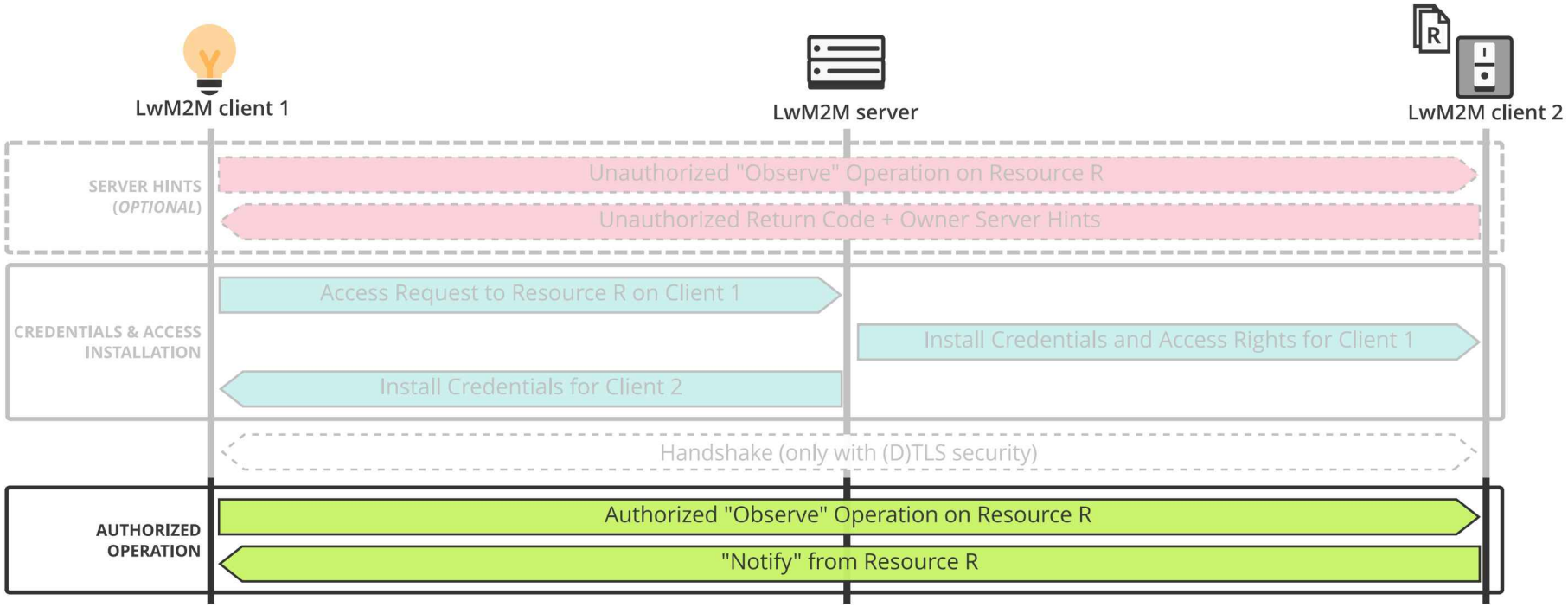
# Third Party Authorization of LwM2M Clients



# Third Party Authorization of LwM2M Clients



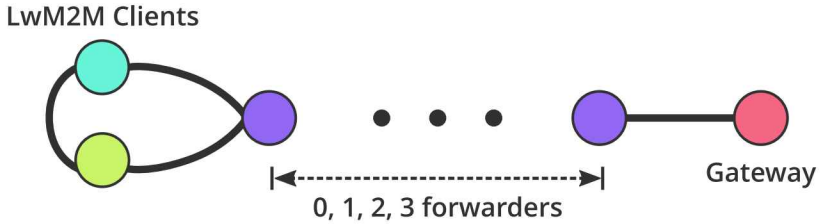
# Third Party Authorization of LwM2M Clients



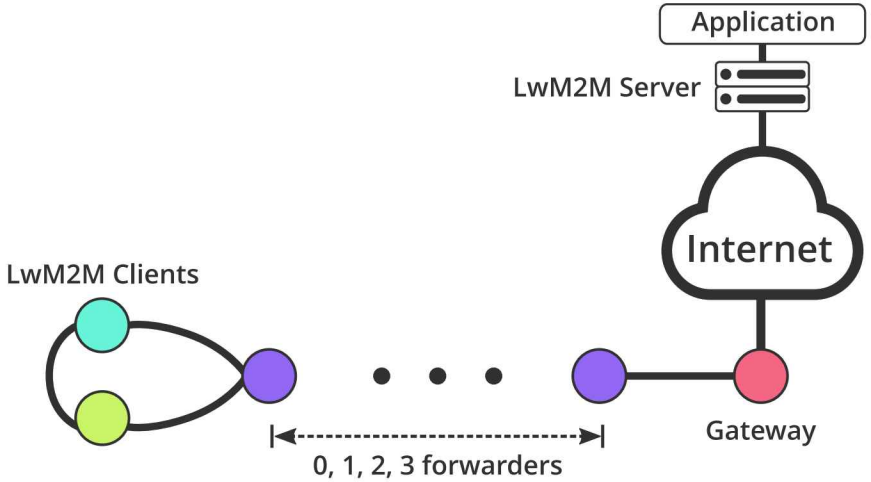
# Experimental Evaluation

# Experimental Setup

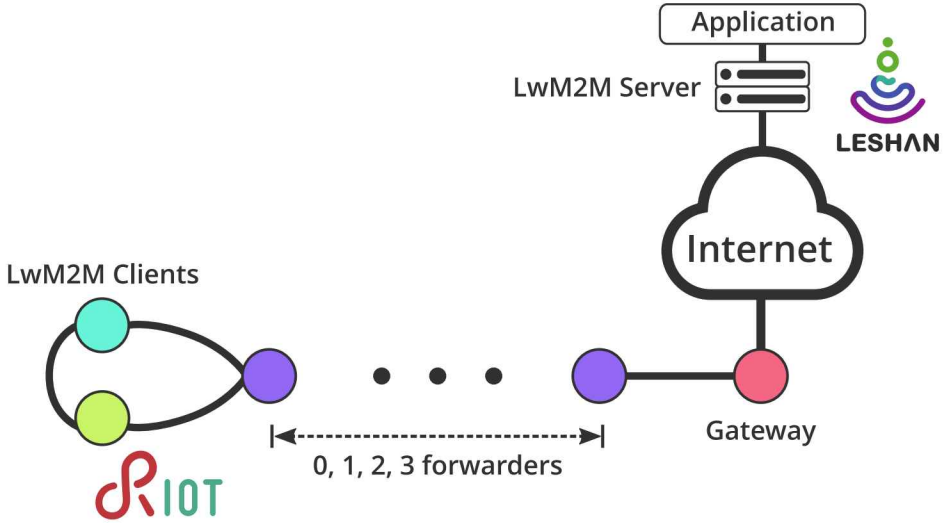
# Experimental Setup



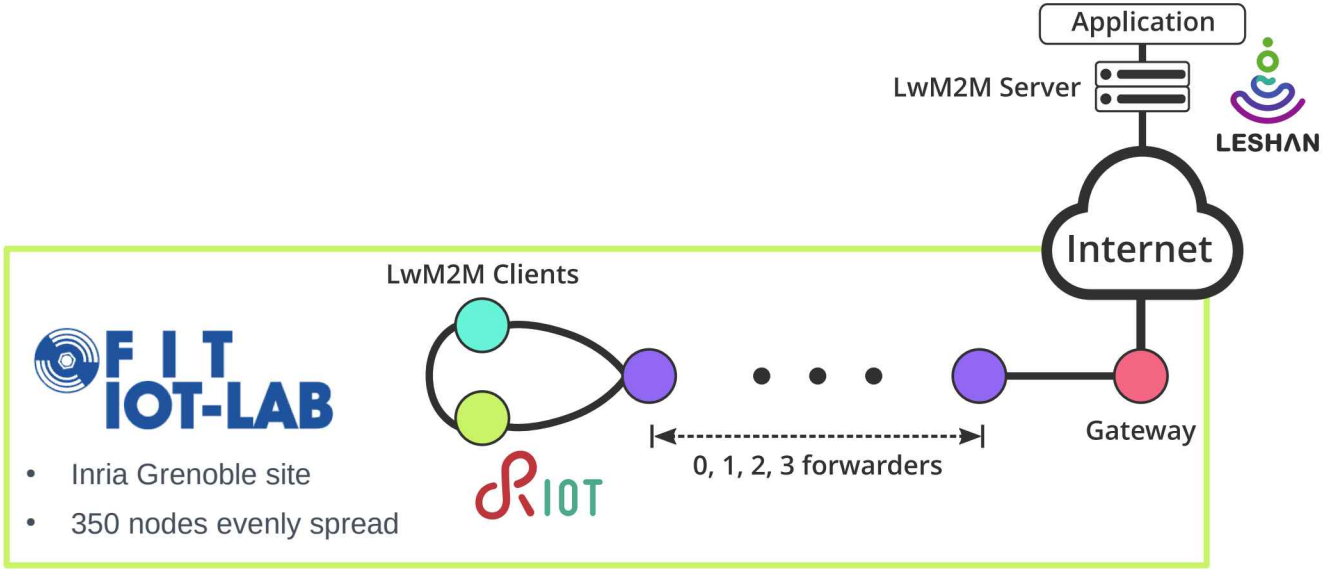
# Experimental Setup



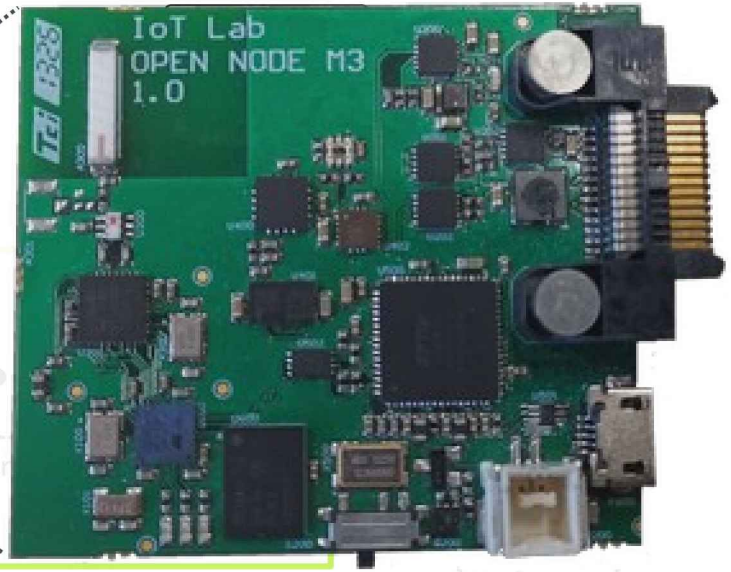
# Experimental Setup

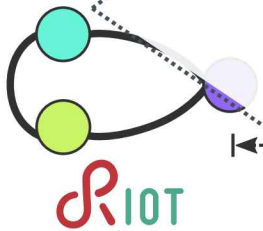


# Experimental Setup



# Experimental Setup

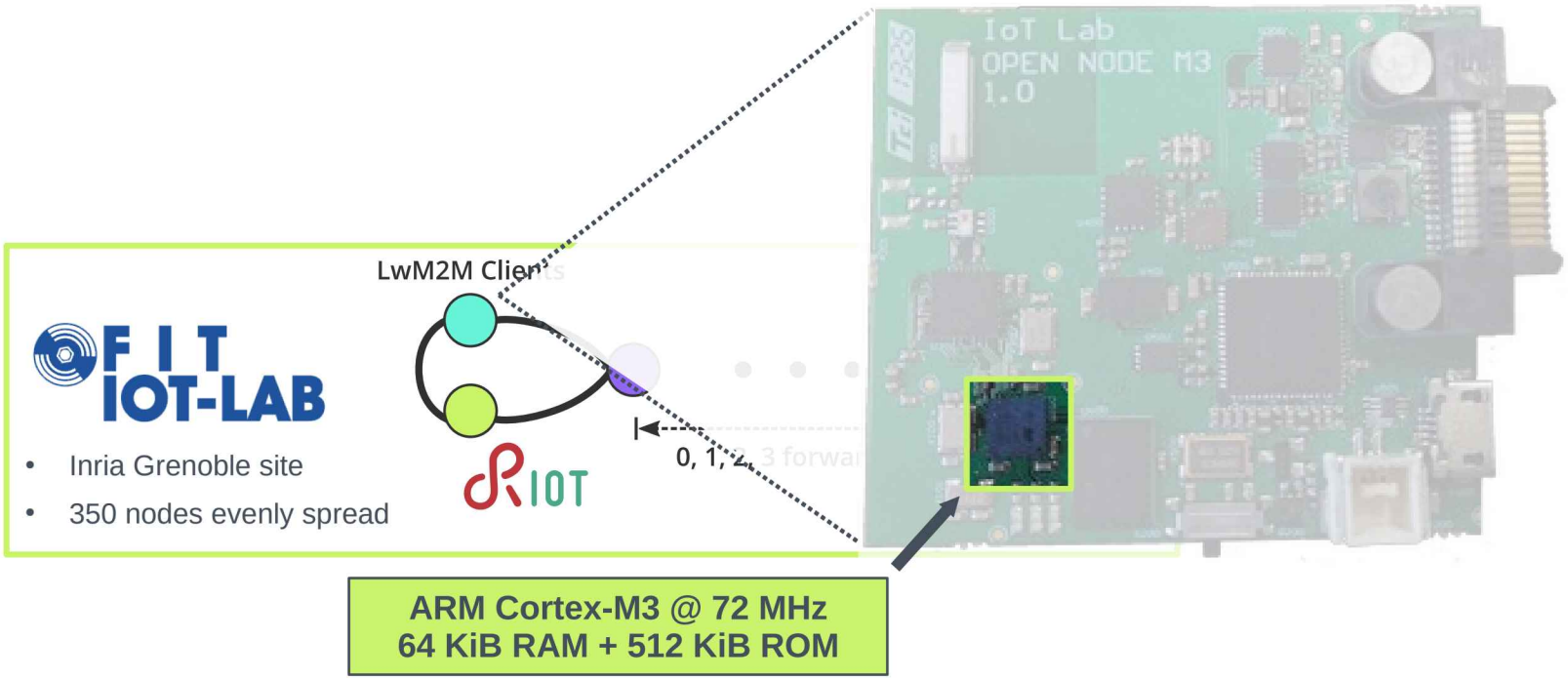


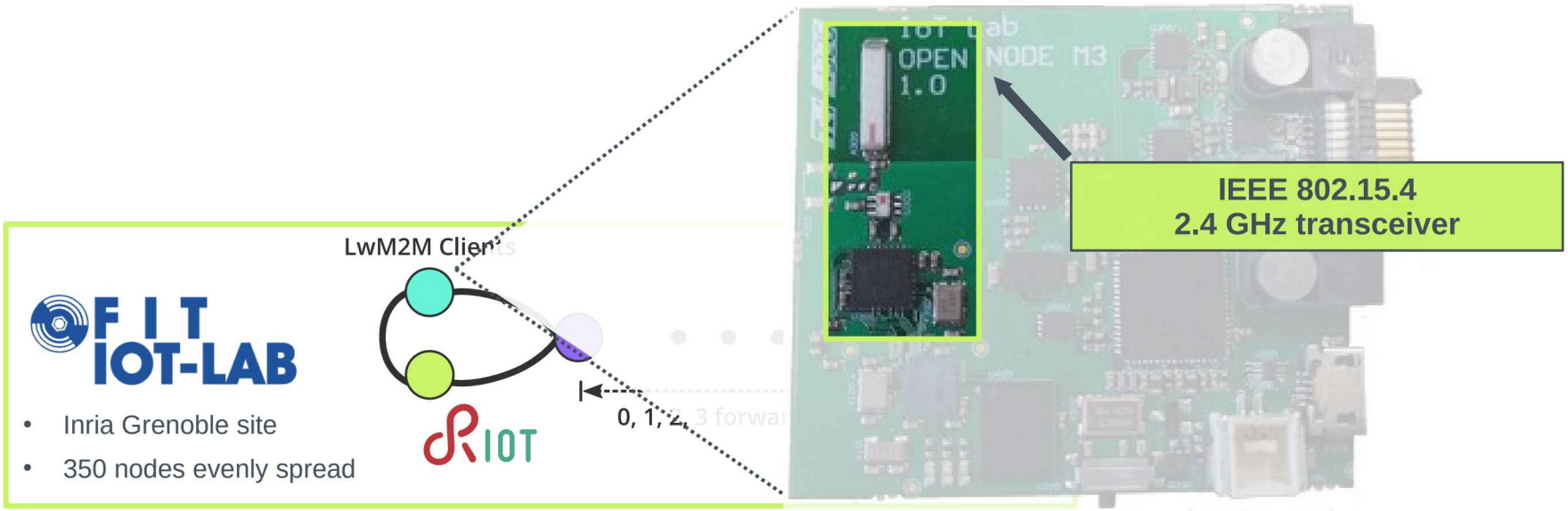

FIT IOT-LAB

- Inria Grenoble site
- 350 nodes evenly spread

# Experimental Setup



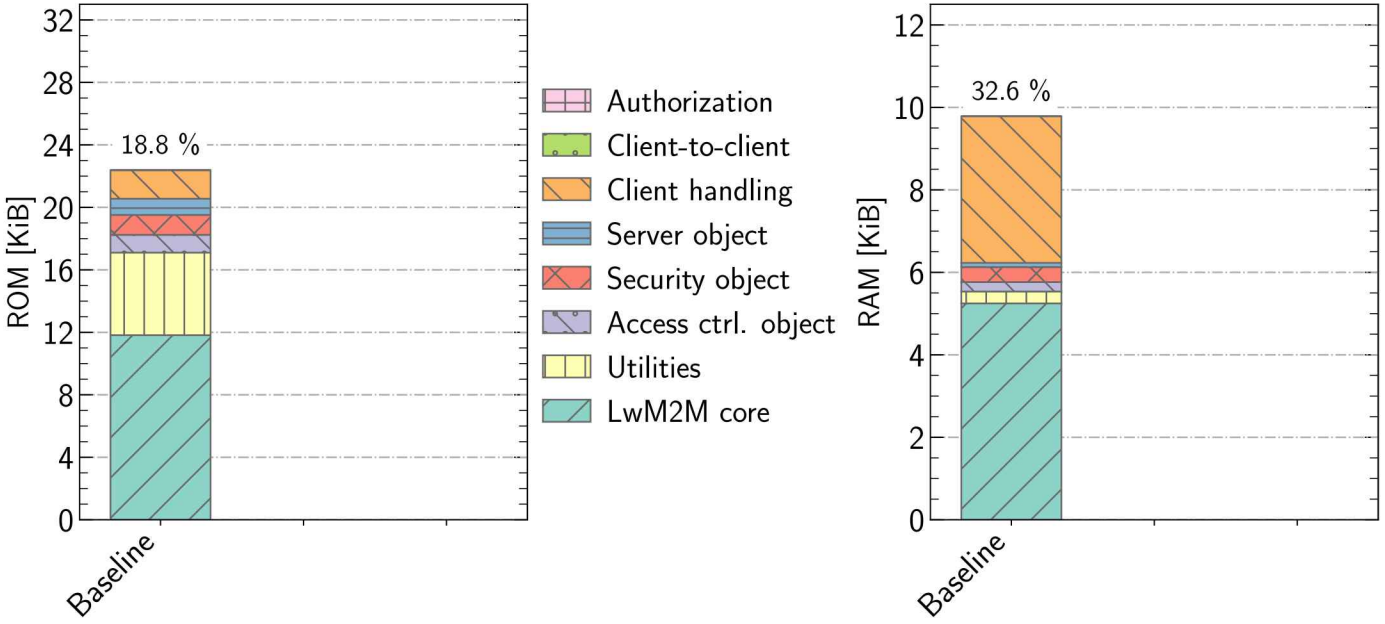
# Experimental Setup



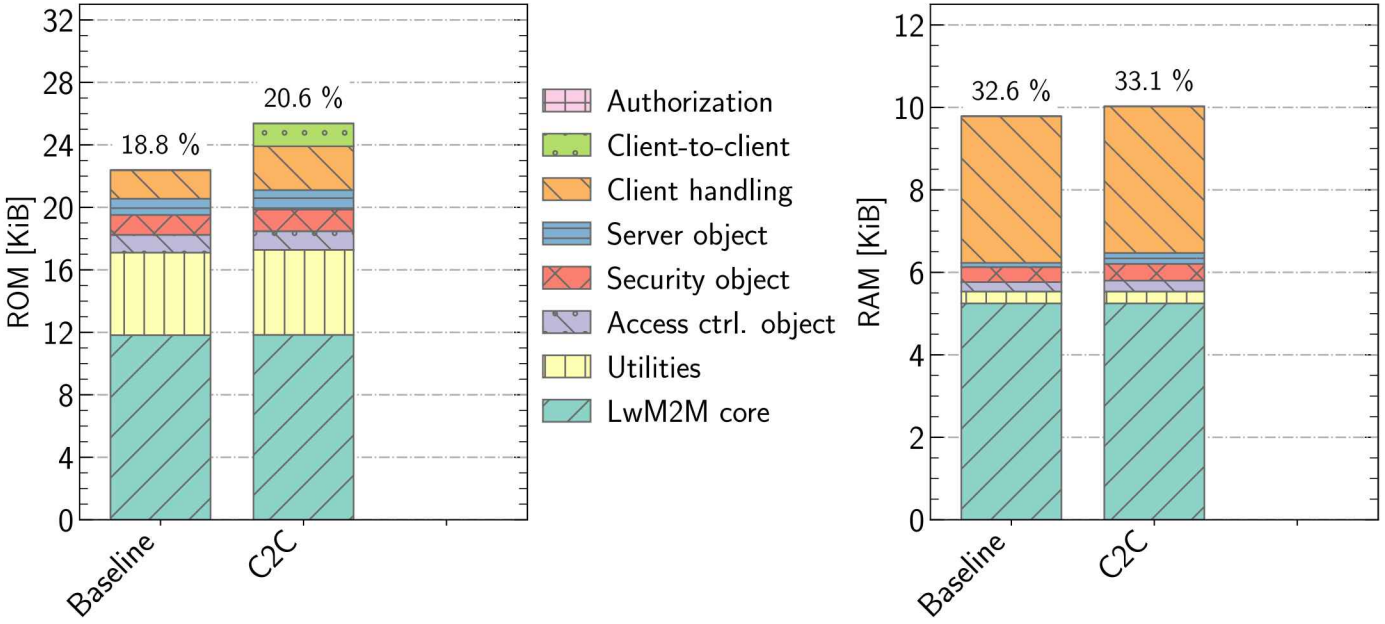

FIT IOT-LAB

- Inria Grenoble site
- 350 nodes evenly spread

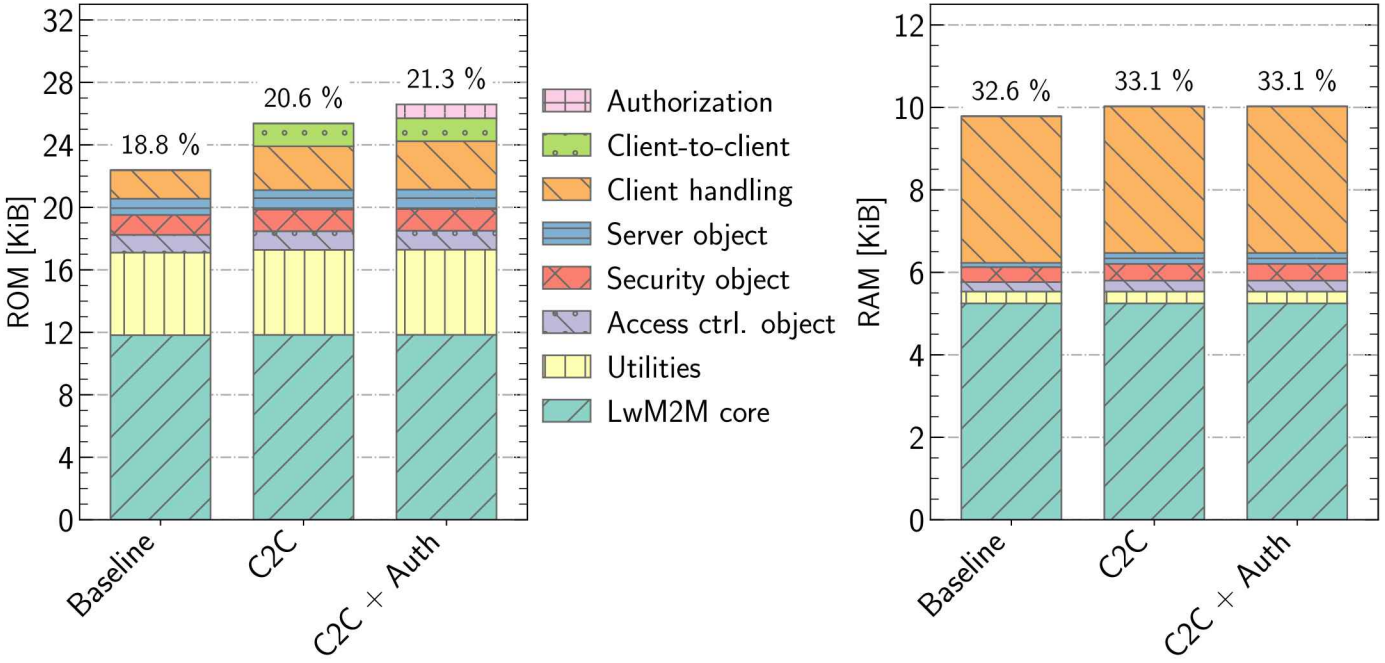
# Firmware Size



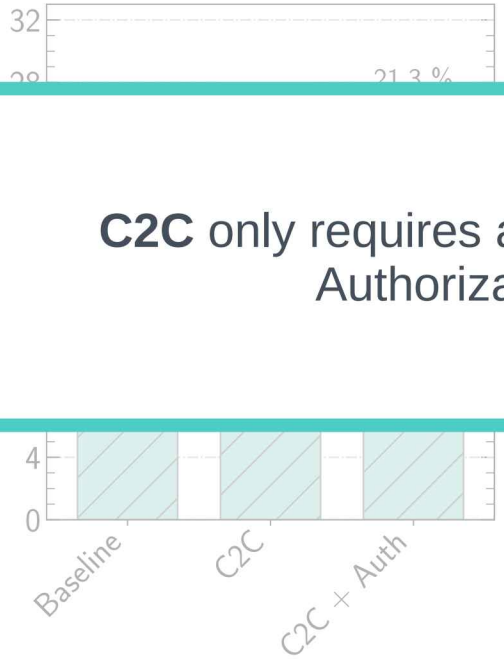
# Firmware Size

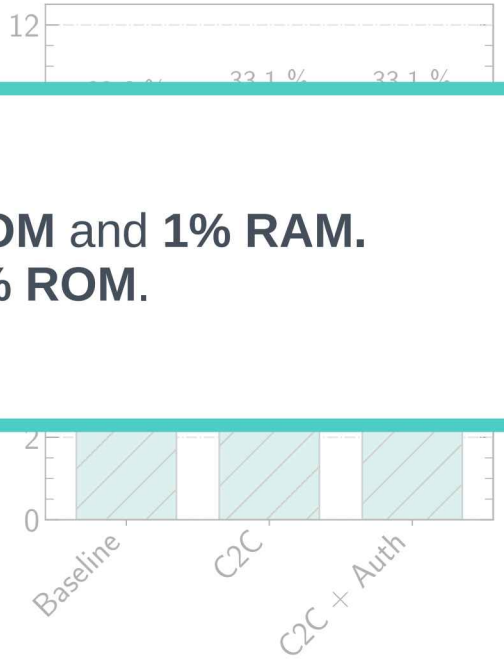


# Firmware Size



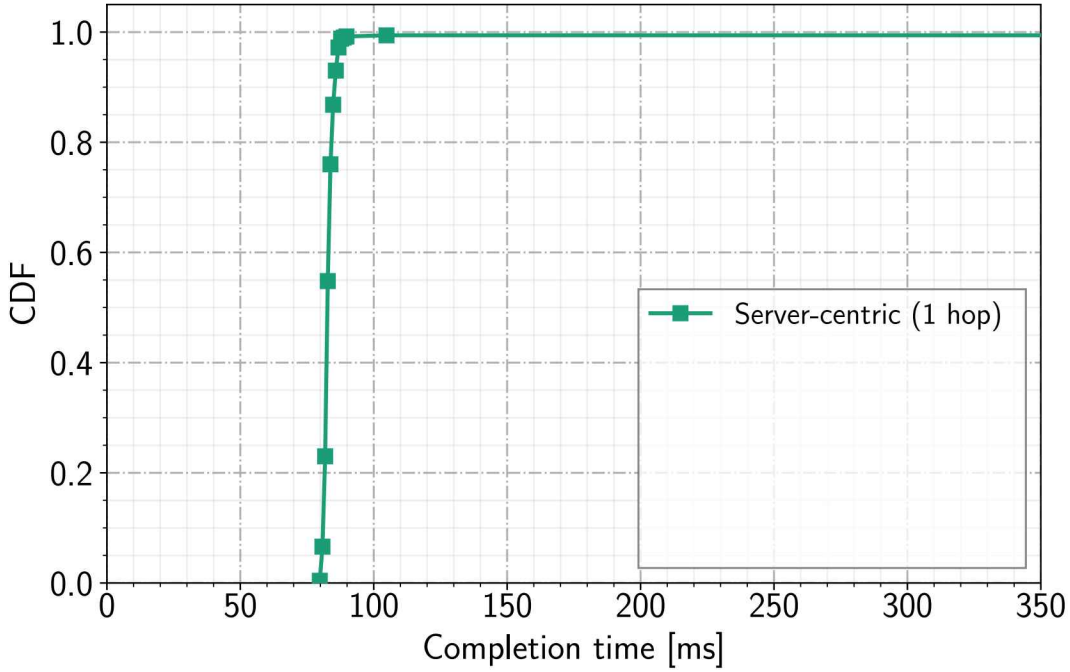
# Firmware Size



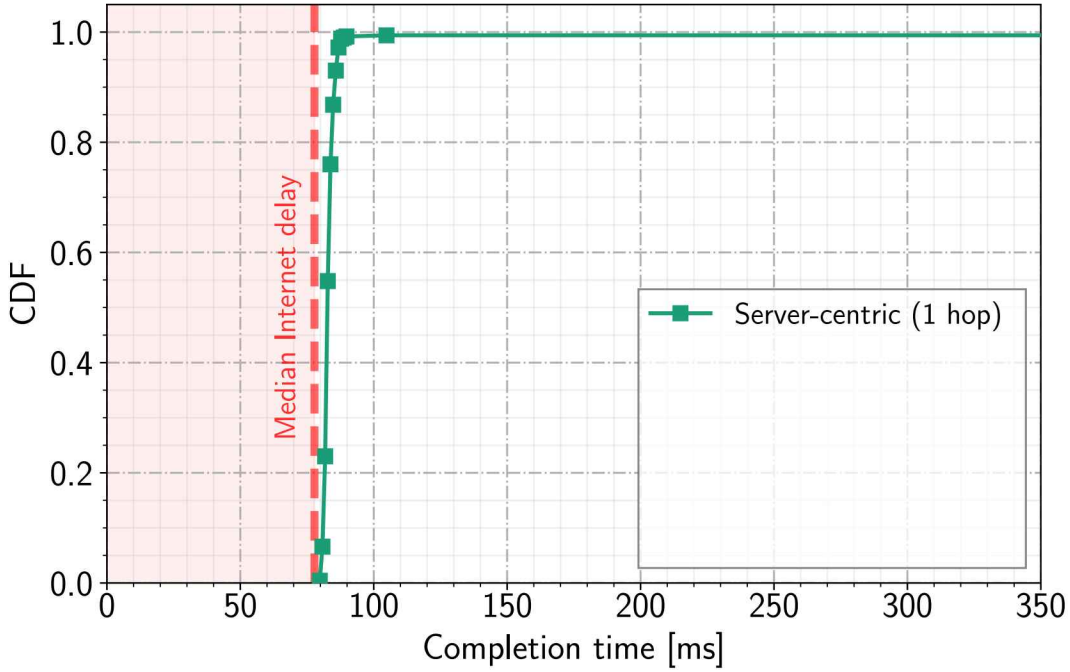




**C2C** only requires additional **3% ROM** and **1% RAM.**
Authorization an extra **5% ROM**.

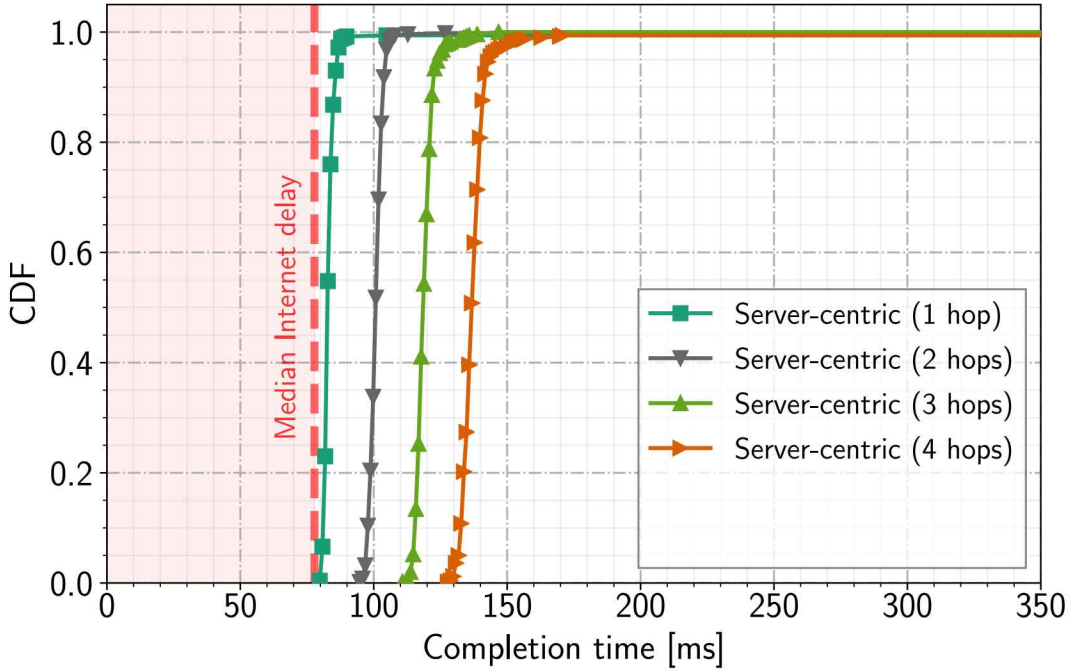
# Notification Arrival Time



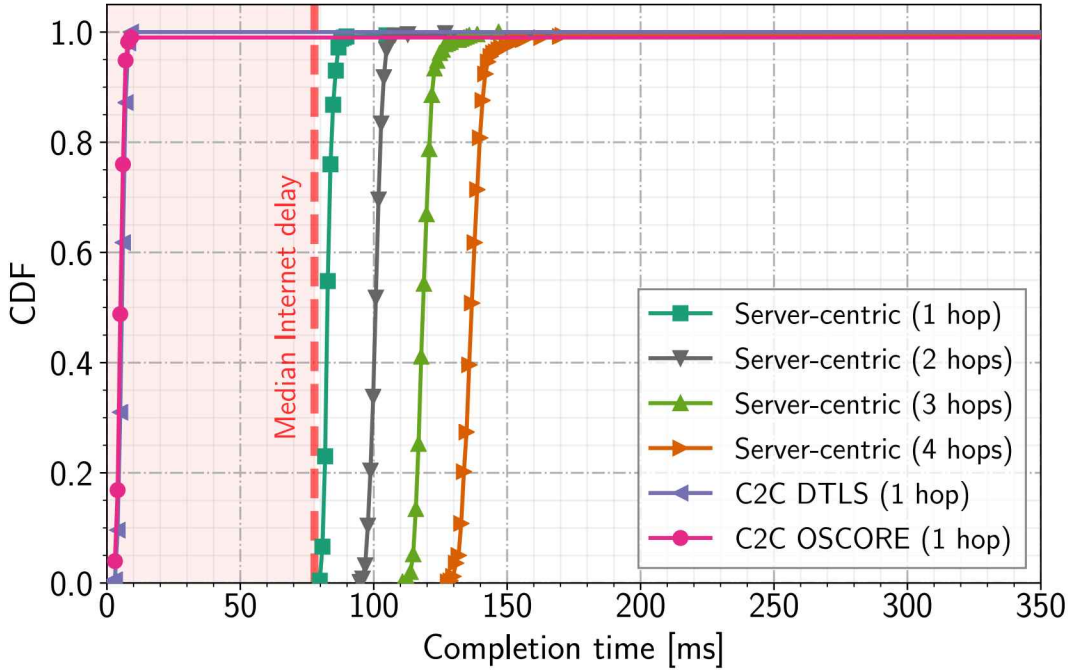
# Notification Arrival Time



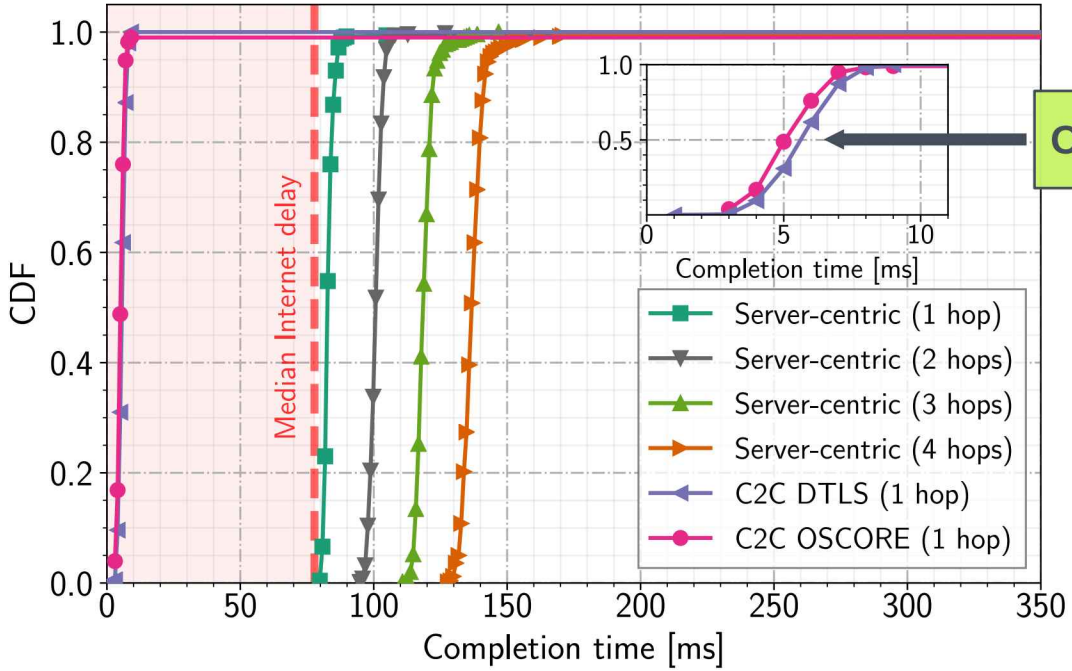
# Notification Arrival Time



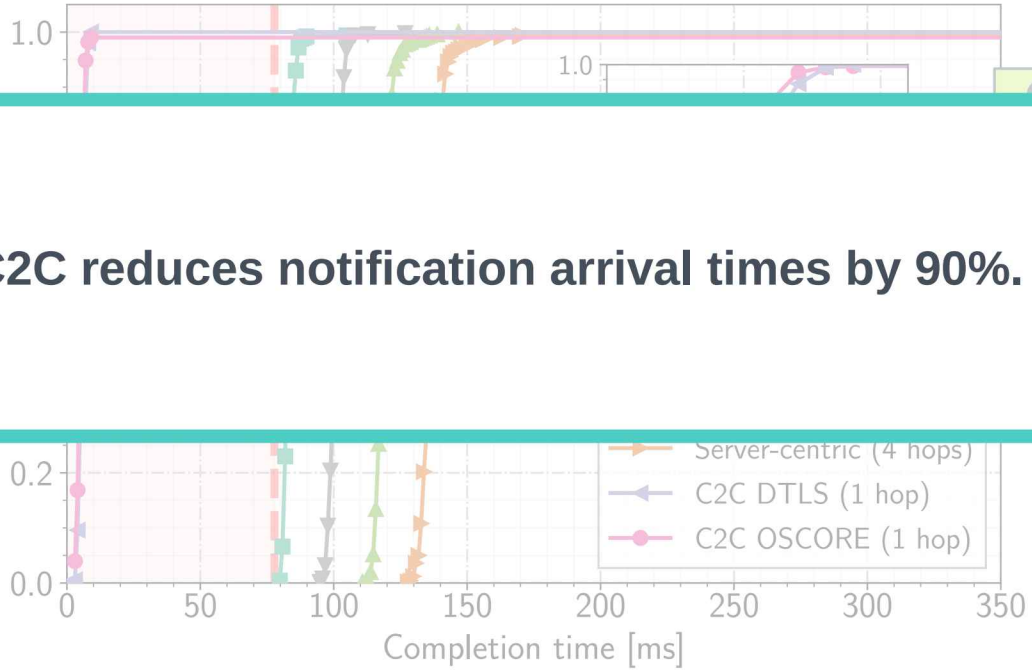
# Notification Arrival Time



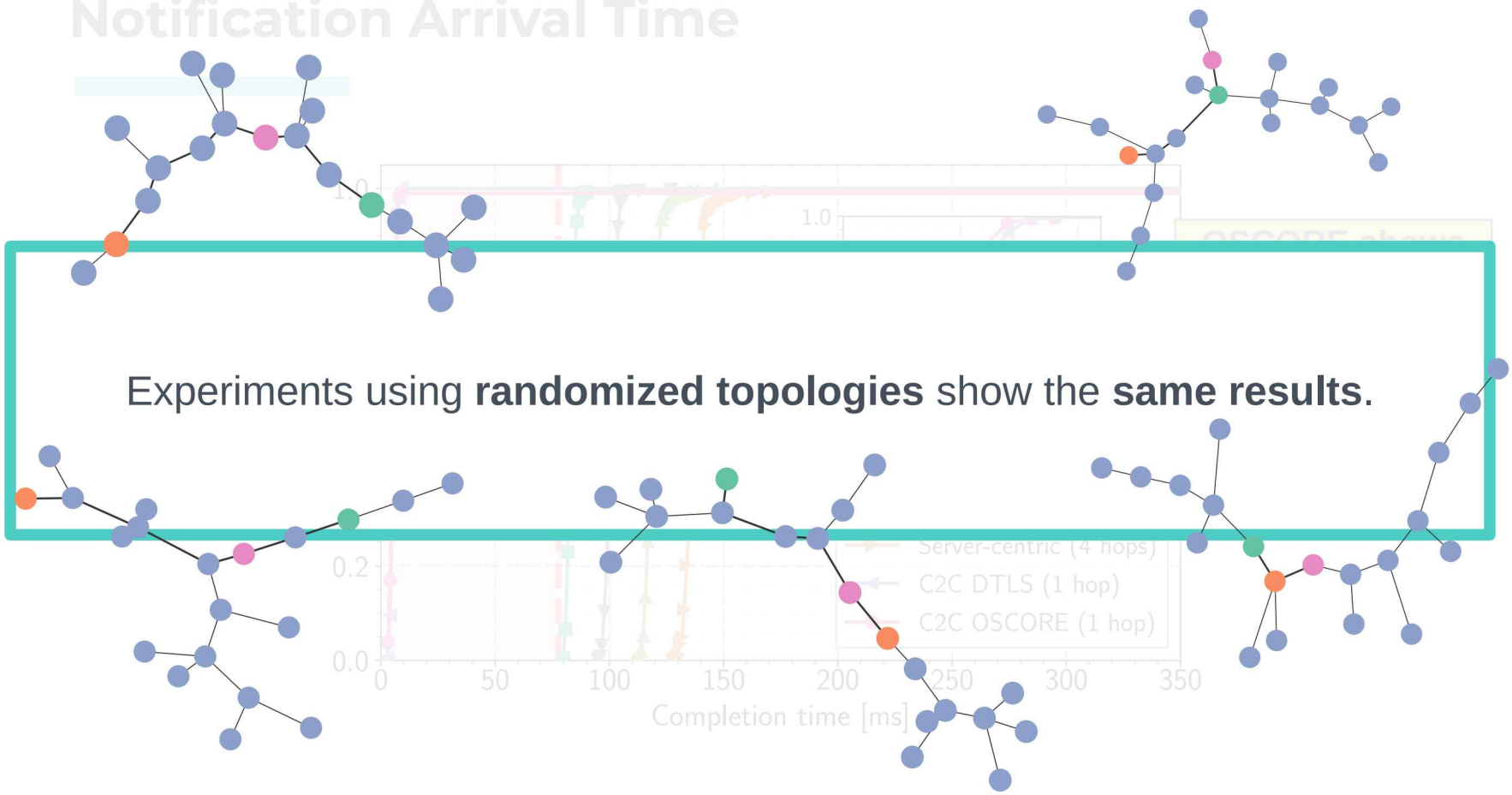
# Notification Arrival Time



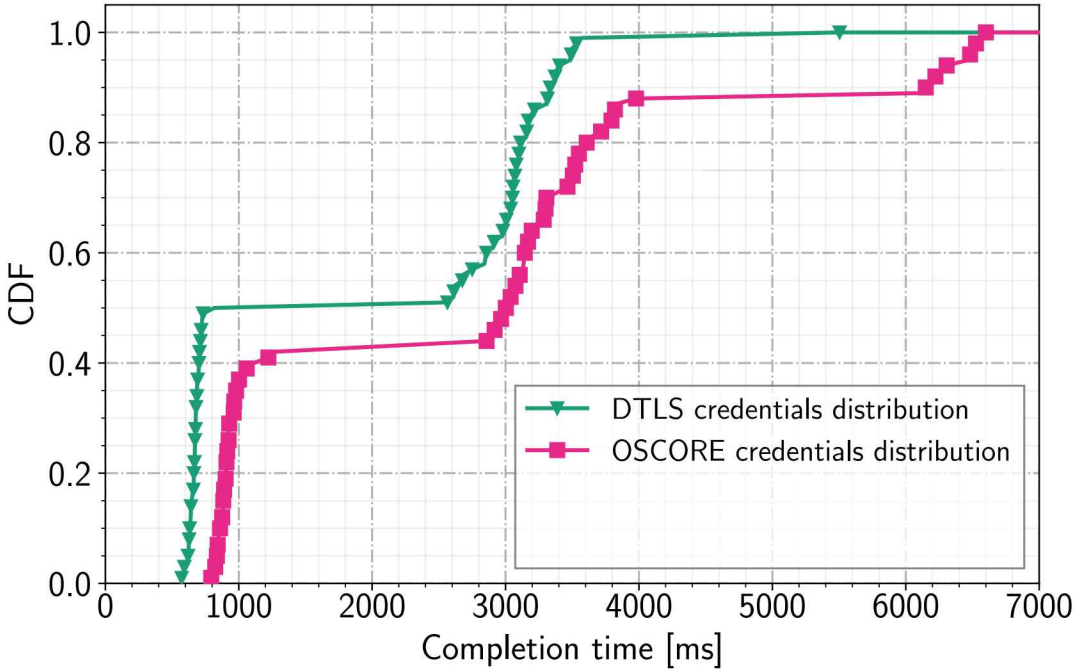
# Notification Arrival Time



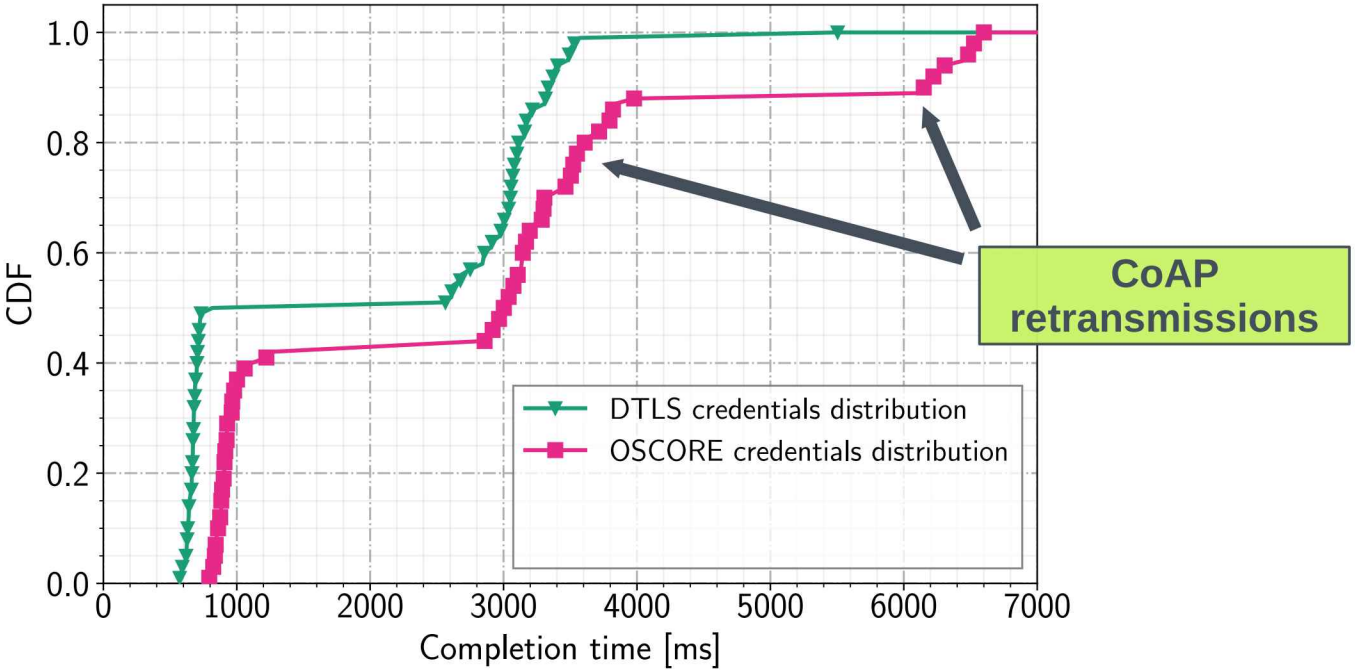

**C2C reduces notification arrival times by 90%.**


Notification Arrival Time
1.0
0.2
0.0
0
50
100
150
200
250
300
350
Completion time [ms]
Server-centric (4 hops)
C2C DTLS (1 hop)
C2C OSCORE (1 hop)
Experiments using **randomized topologies** show the **same results**.

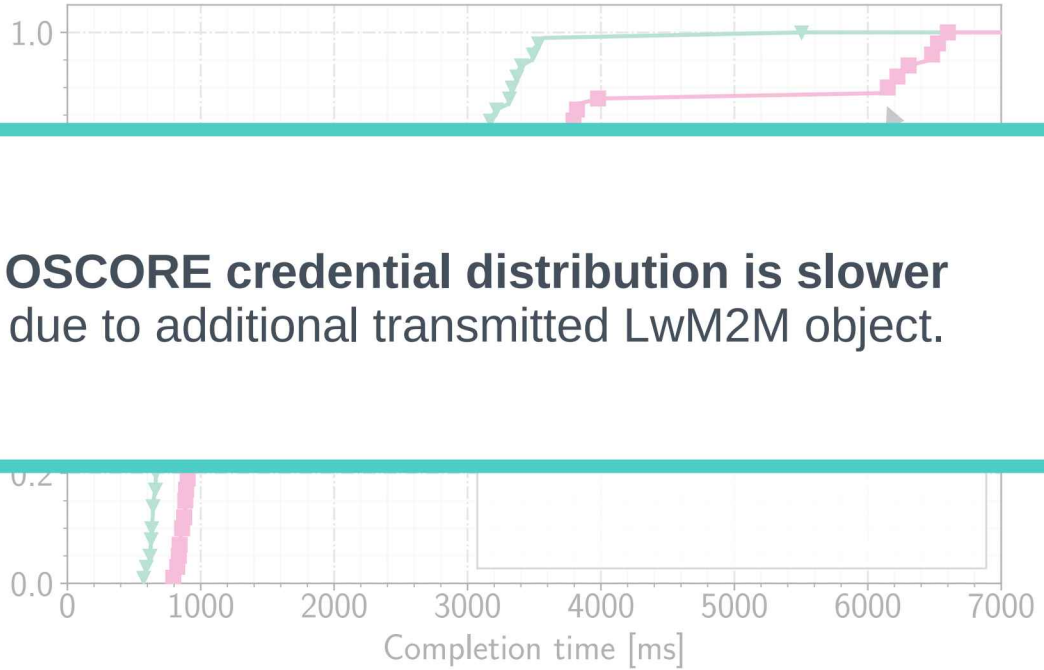
# Authorization Request & First C2C Operation



# Authorization Request & First C2C Operation

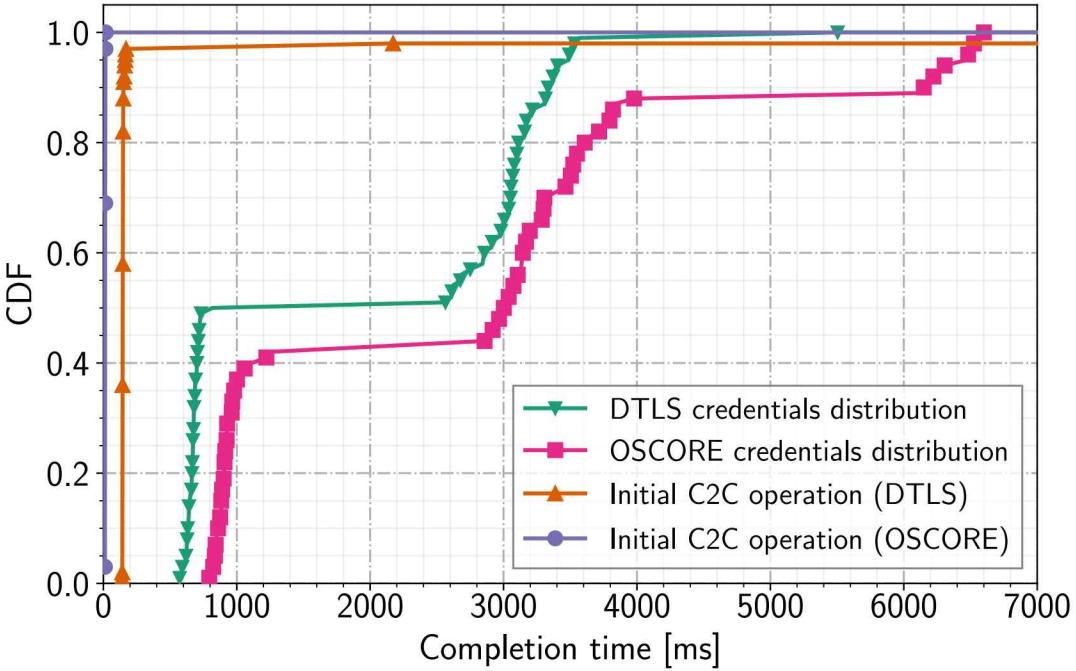


# Authorization Request & First C2C Operation

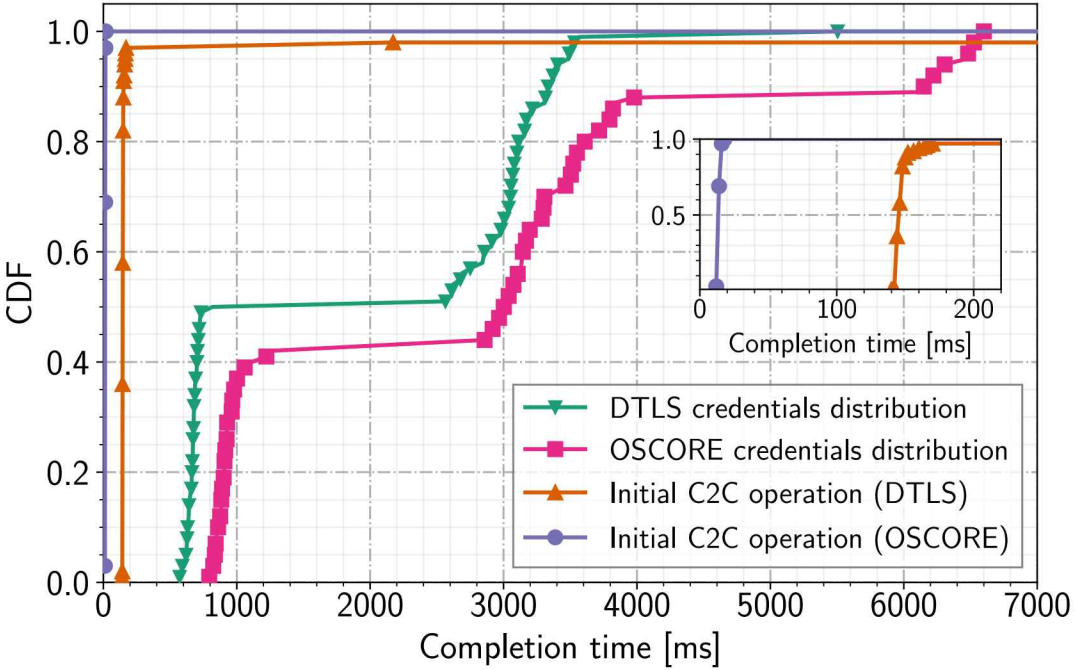



**OSCORE credential distribution is slower**
due to additional transmitted LwM2M object.

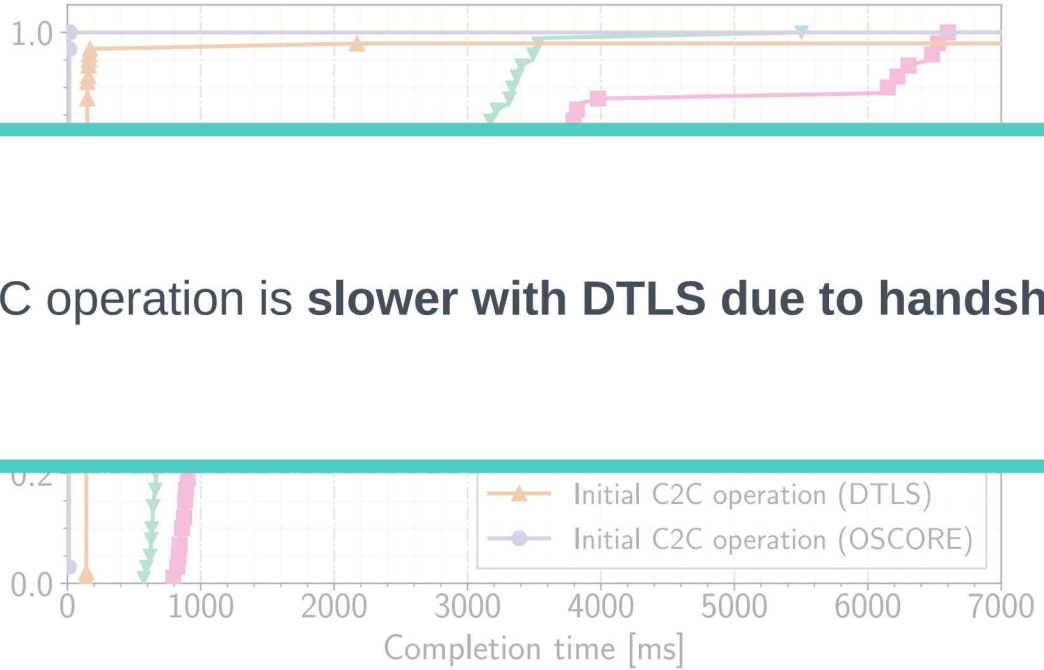
# Authorization Request & First C2C Operation



# Authorization Request & First C2C Operation

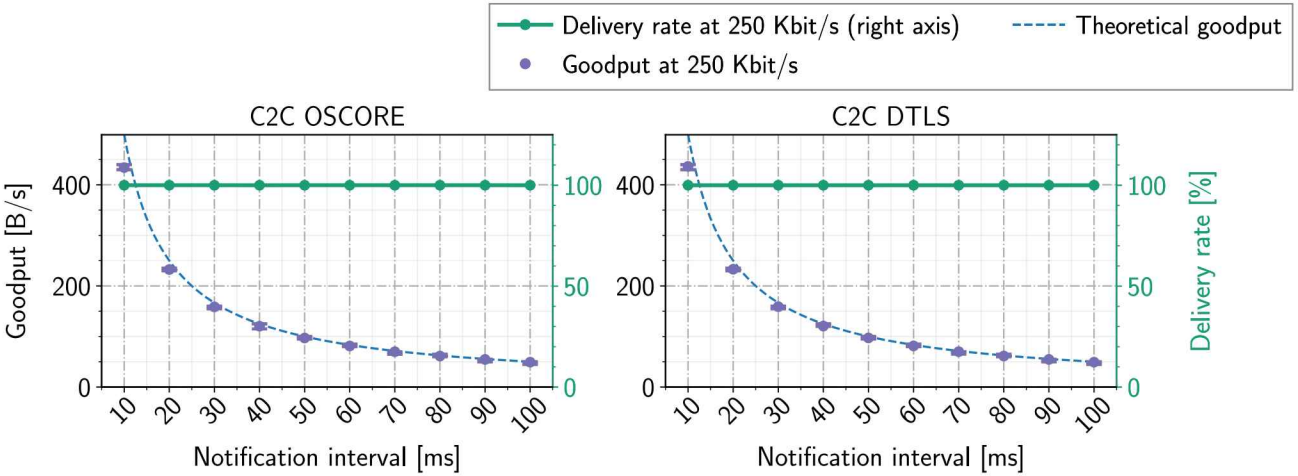


# Authorization Request & First C2C Operation

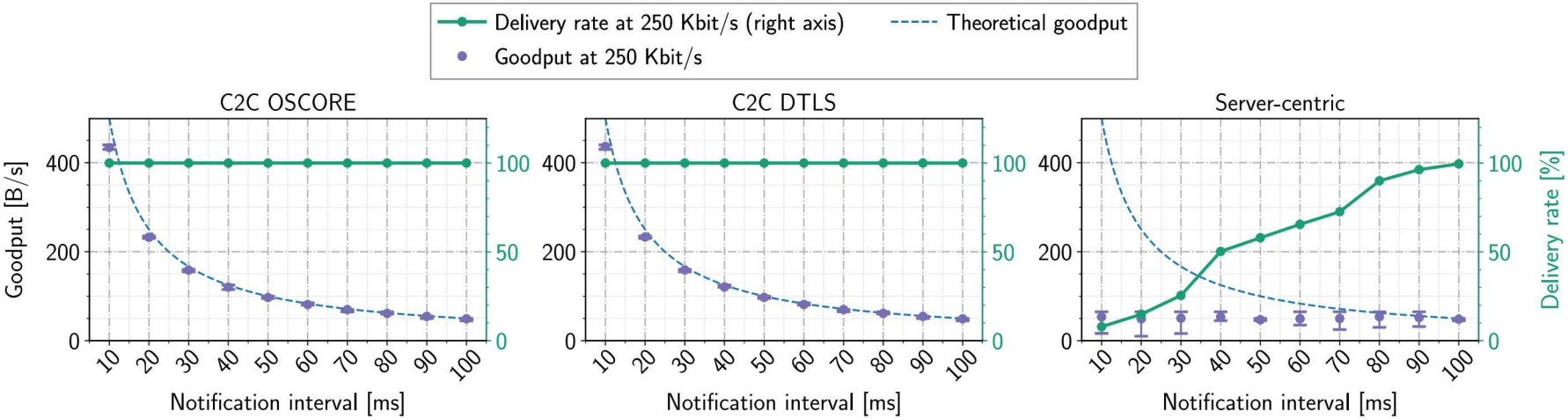



Initial C2C operation is **slower with DTLS due to handshake.**

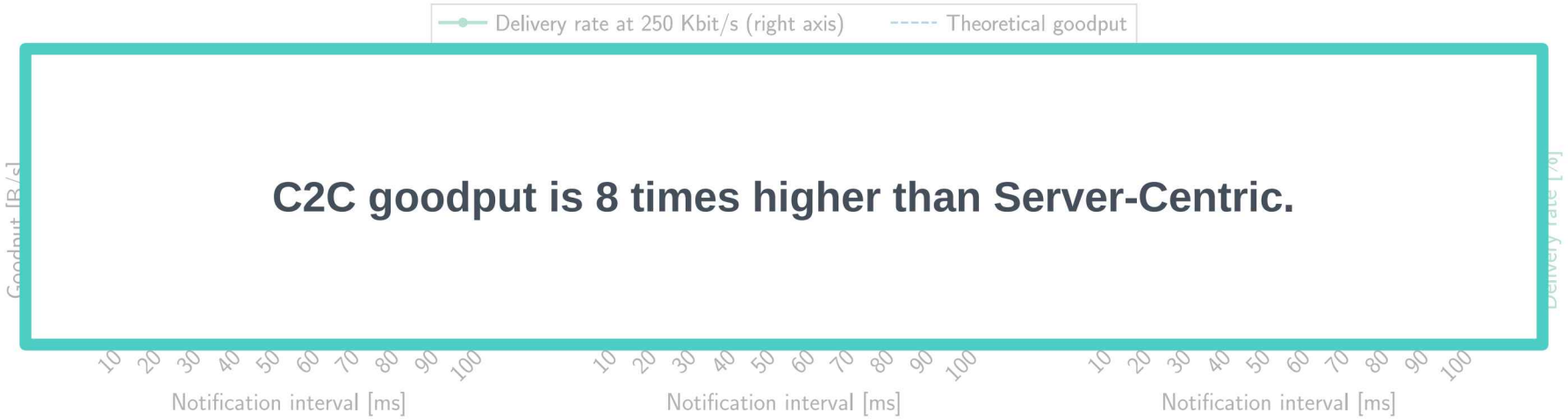
# Maximum Goodput with One Hop



# Maximum Goodput with One Hop

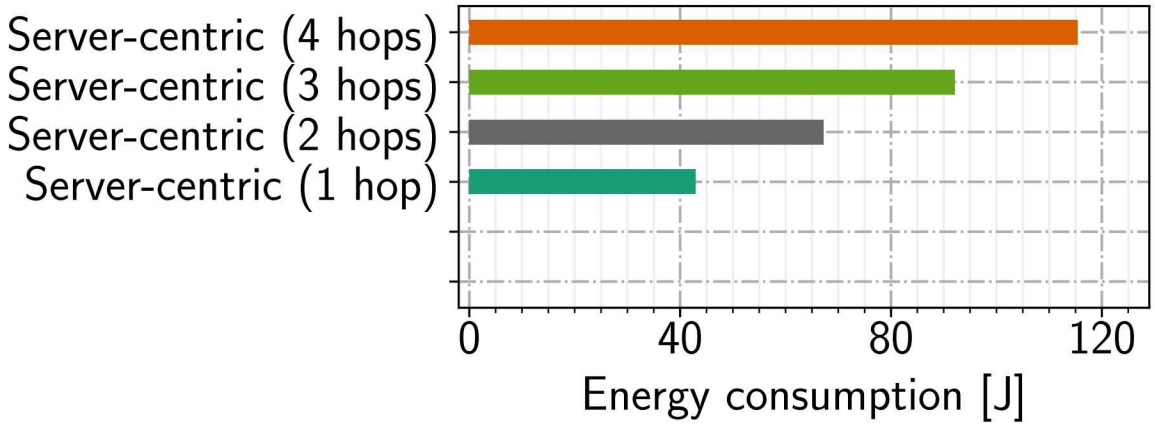


# Maximum Goodput with One Hop

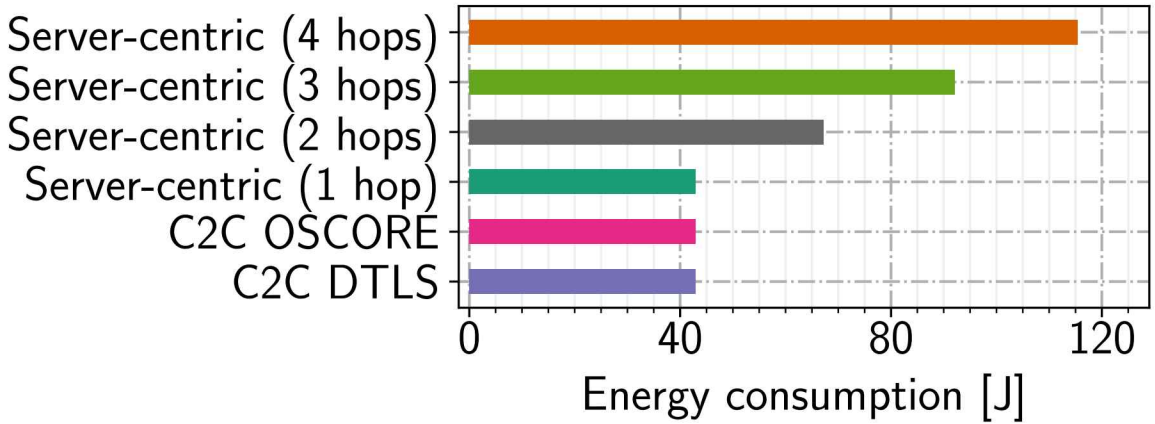



**C2C goodput is 8 times higher than Server-Centric.**

# Energy Consumption



# Energy Consumption



# Energy Consumption

C2C adds **no energy overhead.**
Using less hops reduces energy requirement.

# Conclusion & Outlook

- **We contributed**
  - A **third party authorization mechanism** for LwM2M Clients
  - New LwM2M objects and extended interfaces for **C2C communication**
  - An empirical **performance analysis** on real hardware
  - Public and **open source implementation** of the extensions

- **Our results show that**
  - C2C reduces data arrival times **by up to 90%**
  - C2C yields a more reliable and **8 times higher goodput**
  - Our extensions produce a relatively **small memory footprint**

- **In future work we will**
  - Analyse the applicability of ACE-OAuth framework to LwM2M
  - Explore the integration with Group OSCORE for multiple observations

# Conclusion & Outlook

- **We contributed**
  - A **third party authorization mechanism** for LwM2M Clients
  - New LwM2M objects and extended interfaces for **C2C communication**
  - An empirical **performance analysis** on real hardware
  - Public and **open source implementation** of the extensions

- **Our results show that**
  - C2C reduces data arrival times **by up to 90%**
  - C2C yields a more reliable and **8 times higher goodput**
  - Our extensions produce a relatively **small memory footprint**

- **In future work we will**
  - Analyse the applicability of ACE-OAuth framework to LwM2M
  - Explore the integration with Group OSCORE for multiple observations

# Conclusion & Outlook

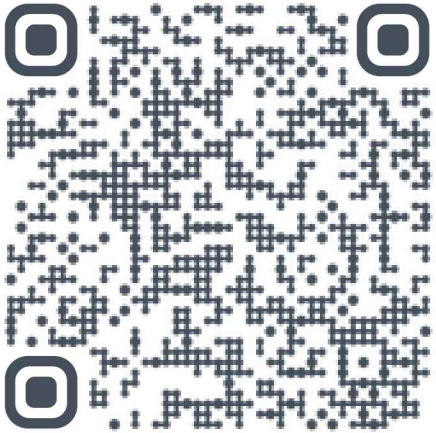
- **We contributed**
  - A **third party authorization mechanism** for LwM2M Clients
  - New LwM2M objects and extended interfaces for **C2C communication**
  - An empirical **performance analysis** on real hardware
  - Public and **open source implementation** of the extensions

- **Our results show that**
  - C2C reduces data arrival times **by up to 90%**
  - C2C yields a more reliable and **8 times higher goodput**
  - Our extensions produce a relatively **small memory footprint**

- **In future work we will**
  - Analyse the applicability of ACE-OAuth framework to LwM2M
  - Explore the integration with Group OSCORE for multiple observations

# Thank You!

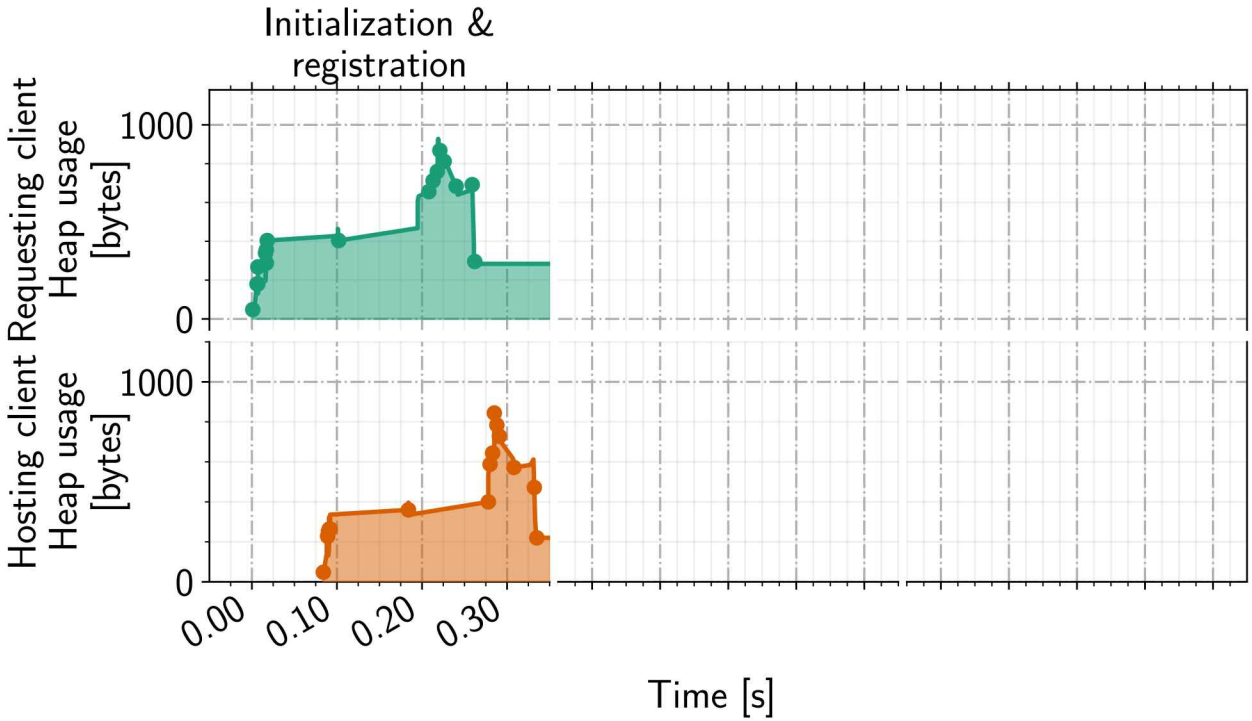
We support reproducible research.



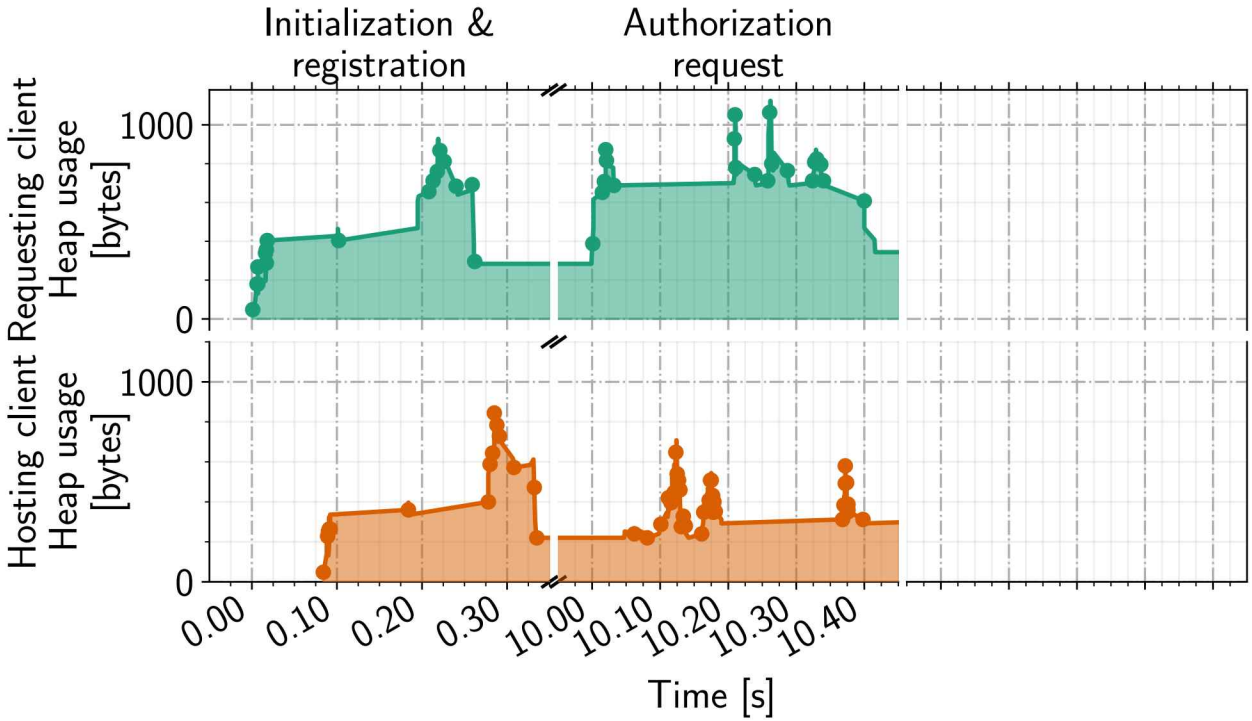
https://github.com/inetrg/ipsn-2022-lwm2mc2c

# Backup Slides

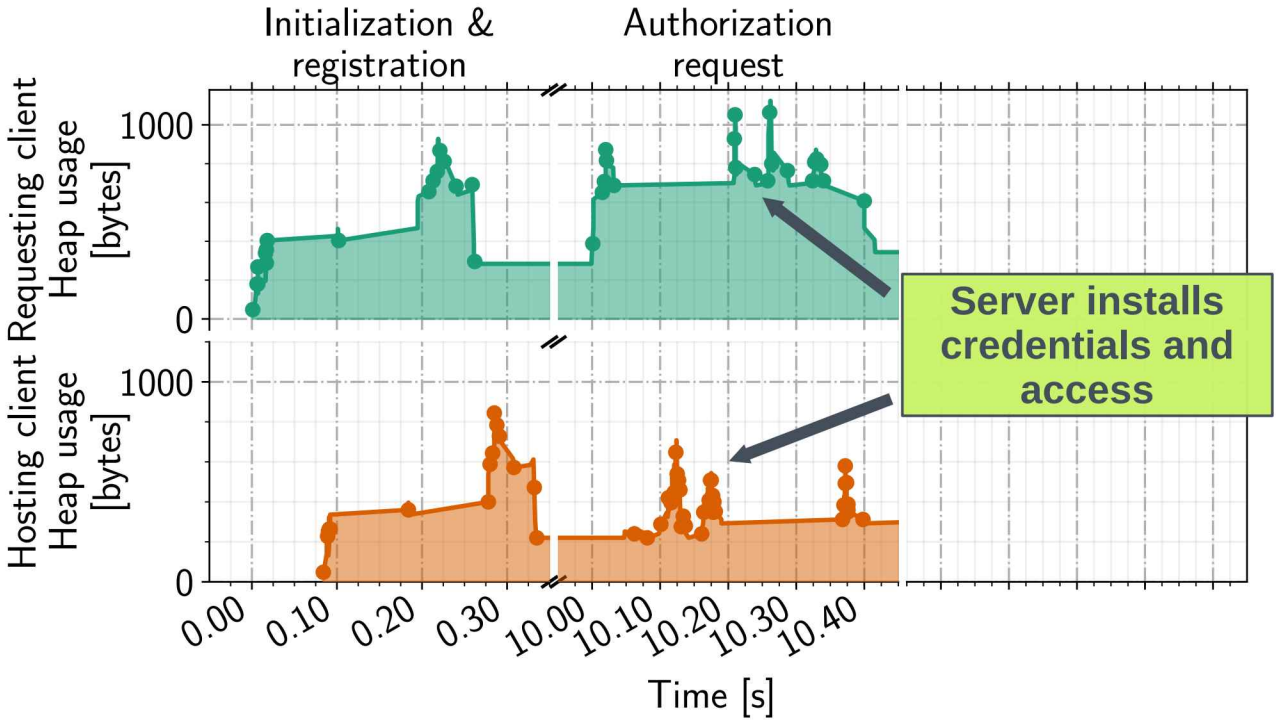
# Heap Usage


Initialization &
registration
Requesting client
Heap usage
[bytes]
1000
0
Hosting client
Heap usage
[bytes]
1000
0
0.00
0.10
0.20
0.30
Time [s]

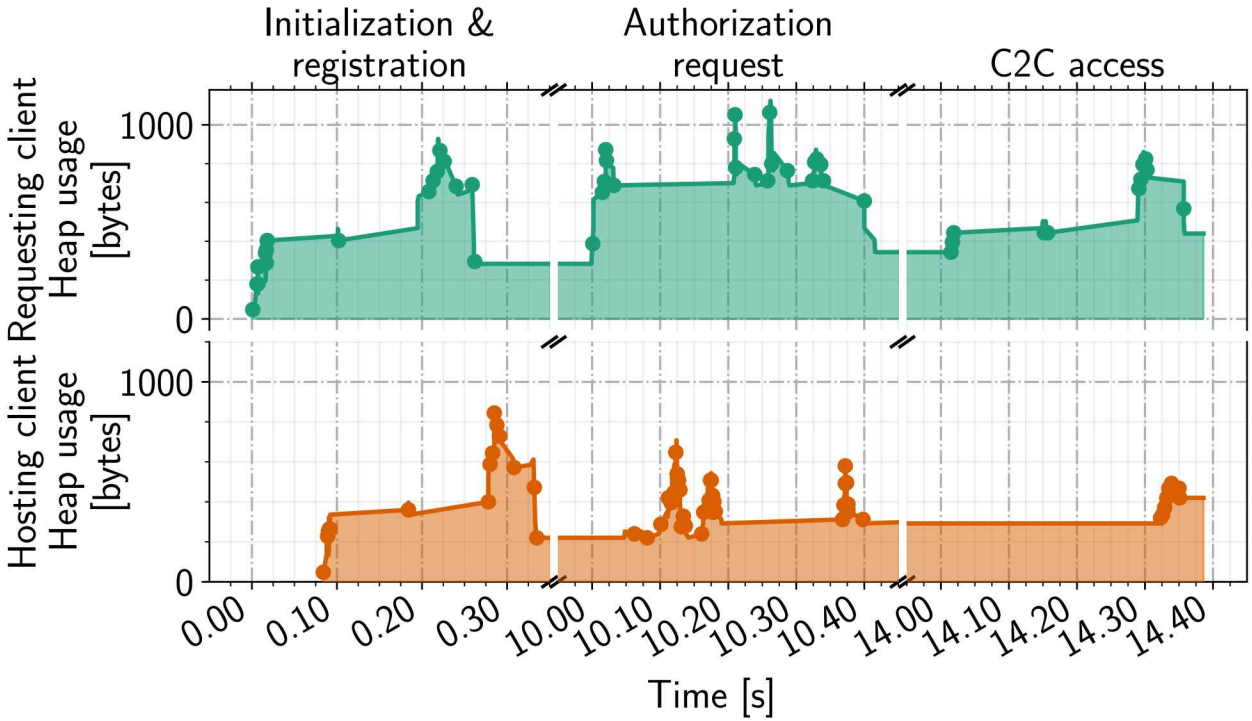
# Heap Usage


Initialization & registration
Authorization request
Requesting client Heap usage [bytes]
Hosting client Heap usage [bytes]
1000
0
0.00
0.10
0.20
0.30
10.00
10.10
10.20
10.30
10.40
Time [s]

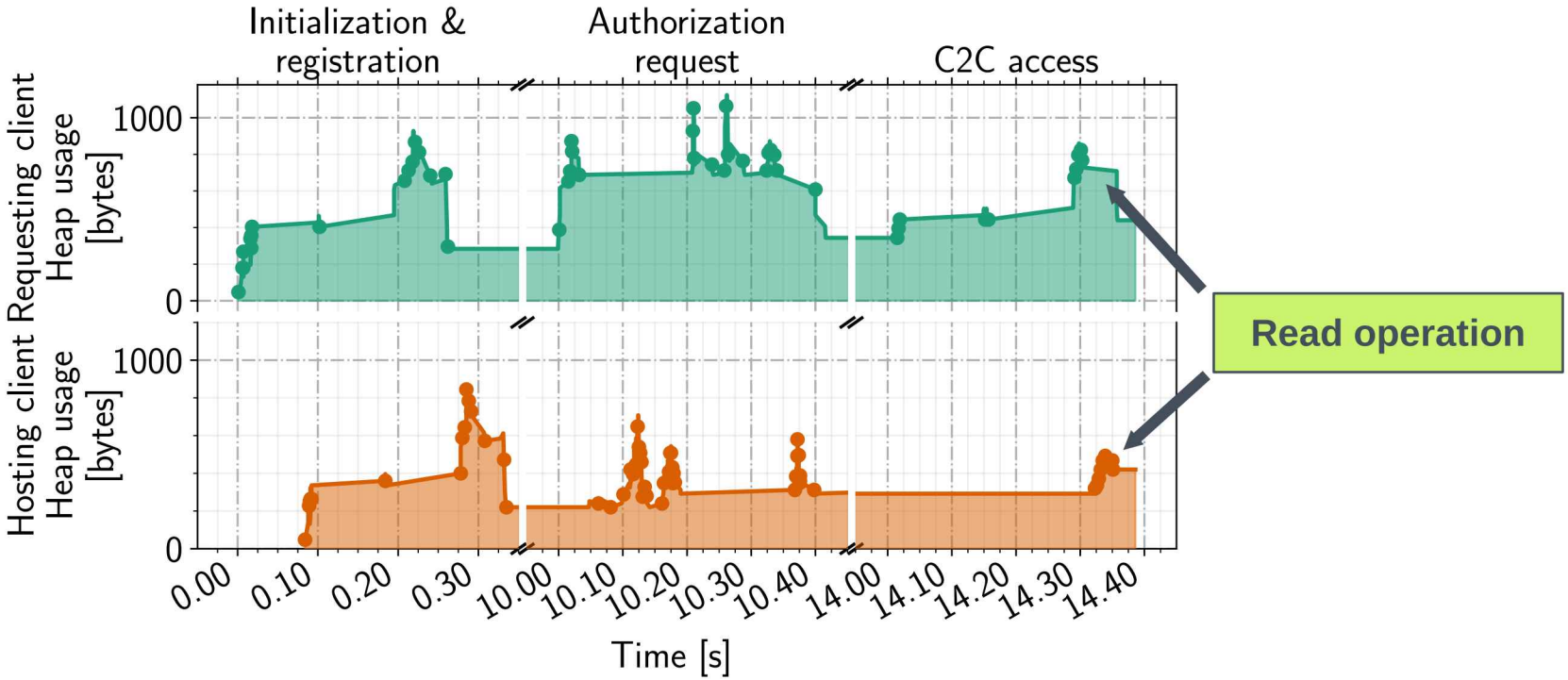
# Heap Usage


Initialization & registration
Authorization request
Requesting client Heap usage [bytes]
Hosting client Heap usage [bytes]
1000
0
1000
0
0.00
0.10
0.20
0.30
10.00
10.10
10.20
10.30
10.40
Time [s]
Server installs credentials and access

# Heap Usage


Initialization & registration
Authorization request
C2C access
Requesting client Heap usage [bytes]
Hosting client Heap usage [bytes]
1000
0
0.00
0.10
0.20
0.30
10.00
10.10
10.20
10.30
10.40
14.00
14.10
14.20
14.30
14.40
Time [s]

# Heap Usage


Initialization & registration
Authorization request
C2C access
Requesting client Heap usage [bytes]
Hosting client Heap usage [bytes]
1000
0
0.00
0.10
0.20
0.30
10.00
10.10
10.20
10.30
10.40
14.00
14.10
14.20
14.30
14.40
Time [s]
Read operation

# Packet Structure