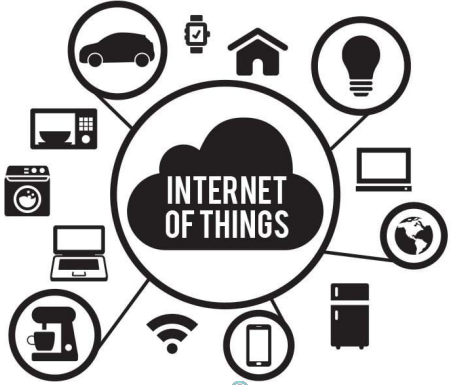
A Case Study of API Design for Interoperability and Security of the Internet of Things

Dongha Kim, Chanhee Lee, and Hokeun Kim

EAI SmartSP 2024 - 2nd EAI International Conference on Security and Privacy in Cyber-Physical Systems and Smart Vehicles
New Orleans, USA
Nov 8, 2024

1. Introduction

- Internet of Things (IoT) has been rising with the benefits of edge computing, such as low latency, privacy protection, and scalability [1, 2].
- **Heterogeneity** of devices -> Challenging to support **diversity** of communication models. (e.g. Smart City)
- **Interoperability**
  - Various communication protocols (e.g. Traffic management)
  - Different security requirements



	Communication Protocols	Security Requirements
Collecting Sensor Data	Publish-Subscribe	Low - Prioritize low power consumption
Controllilng Traffic Lights	Point-to-Point	High - Safety-critical

https://thenewstack.io/what-does-it-mean-to-be-on-the-internet-of-things/
https://www.challenge.org/knowledgeitems/the-smart-city-concept-through-digital-twins/

[1] Ning, H., Li, Y., Shi, F., Yang, L.T.: Heterogeneous edge computing open platforms and tools for Internet of things. Future Generation Computer Systems 106, 67–76 (2020)
[2] Yu, W., Liang, F., He, X., Hatcher, W.G., Lu, C., Lin, J., Yang, X.: A survey on the edge computing for the Internet of Things. IEEE access 6, 6900–6919 (2017)

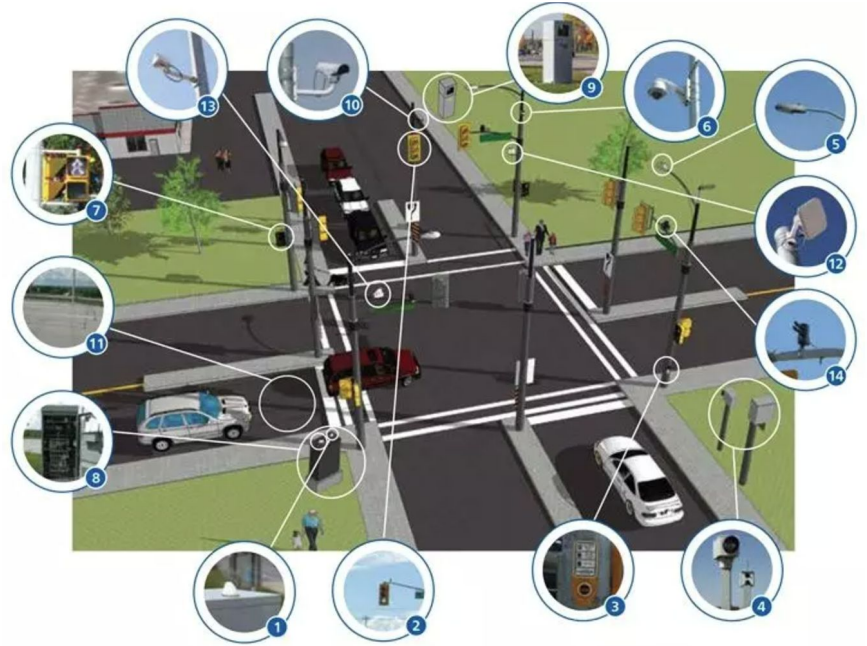
2. Research Goals

1. **Provide a common interface** that supports **multiple communication models**, for **seamless interaction** between heterogeneous subsystems.

2. **Incorporate** a flexible **security framework** that can be adaptively applied based on **various security requirements**.

3. **Implement** a working runtime system using open-source platforms, and **evaluate its performance** showing reasonably **small overhead** while simplifying software development and enhancing maintainability.

Smart City Traffic Monitoring System



1. **Sensors:**
   e.g. Vehicle detection cameras, speed sensors, pedestrian push buttons…

2. **Traffic Lights:**
   Should indicate the signal for vehicles and pedestrians to cross.

3. **Traffic Controllers:**
   Make decisions based on sensors, and send signals.

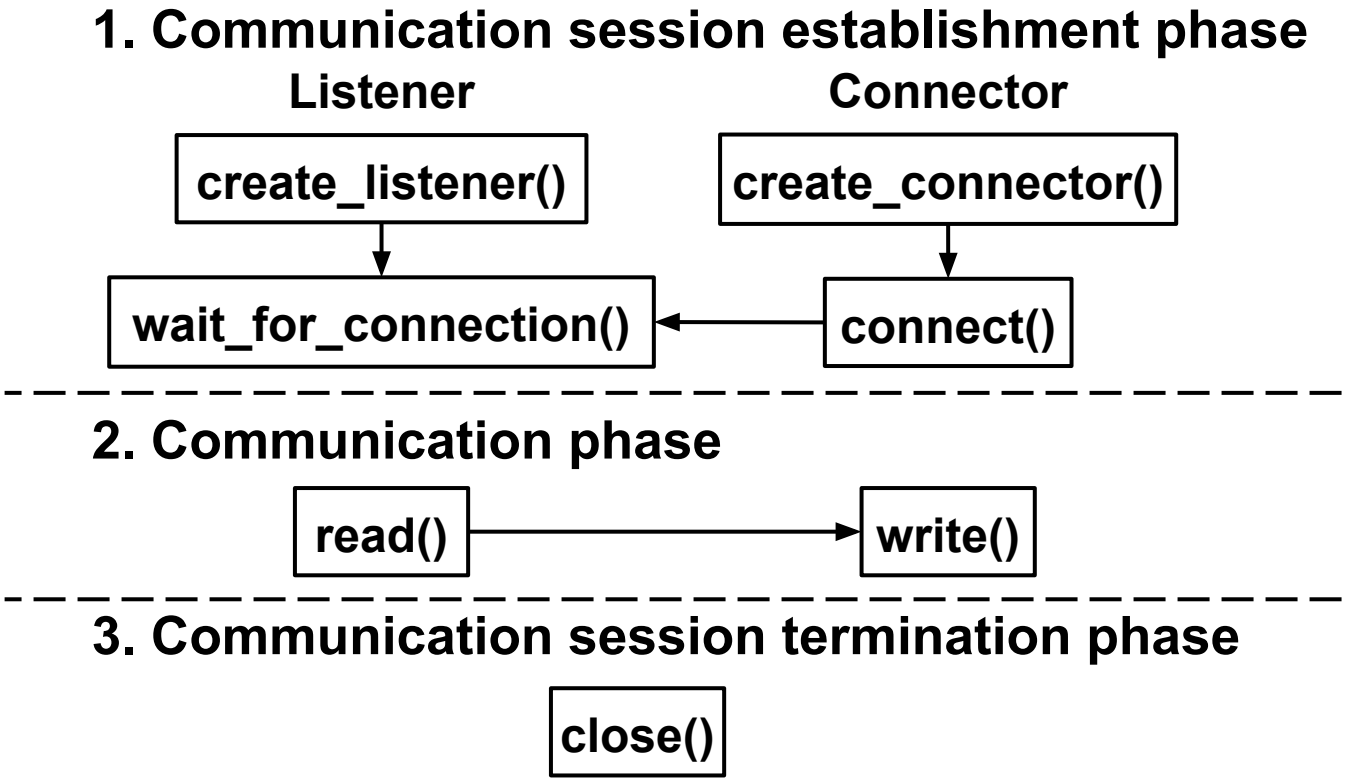
https://www.york.ca/newsroom/campaigns-projects/traffic-technology-intersections

How Client-Server and Publish-Subscribe Differ?

Aspect	Client-Server	Publish-Subscribe
Protocols	TCP	MQTT
Communication	Point-to-Point	Indirect via a broker (typically…)
Scalability	Limited by server capacity and direct connections.	Scales efficiently with multiple subscribers.
Coupling	Tightly coupled; clients need server details.	Loosely coupled; subscribers only need topic details.
L**atency**	Low, due to direct communication.	Slightly higher due to broker mediation.
Use Cases	Heavy, reliable data transfer (e.g., Traffic light controllers).	Lightweight (e.g., Sensor data).

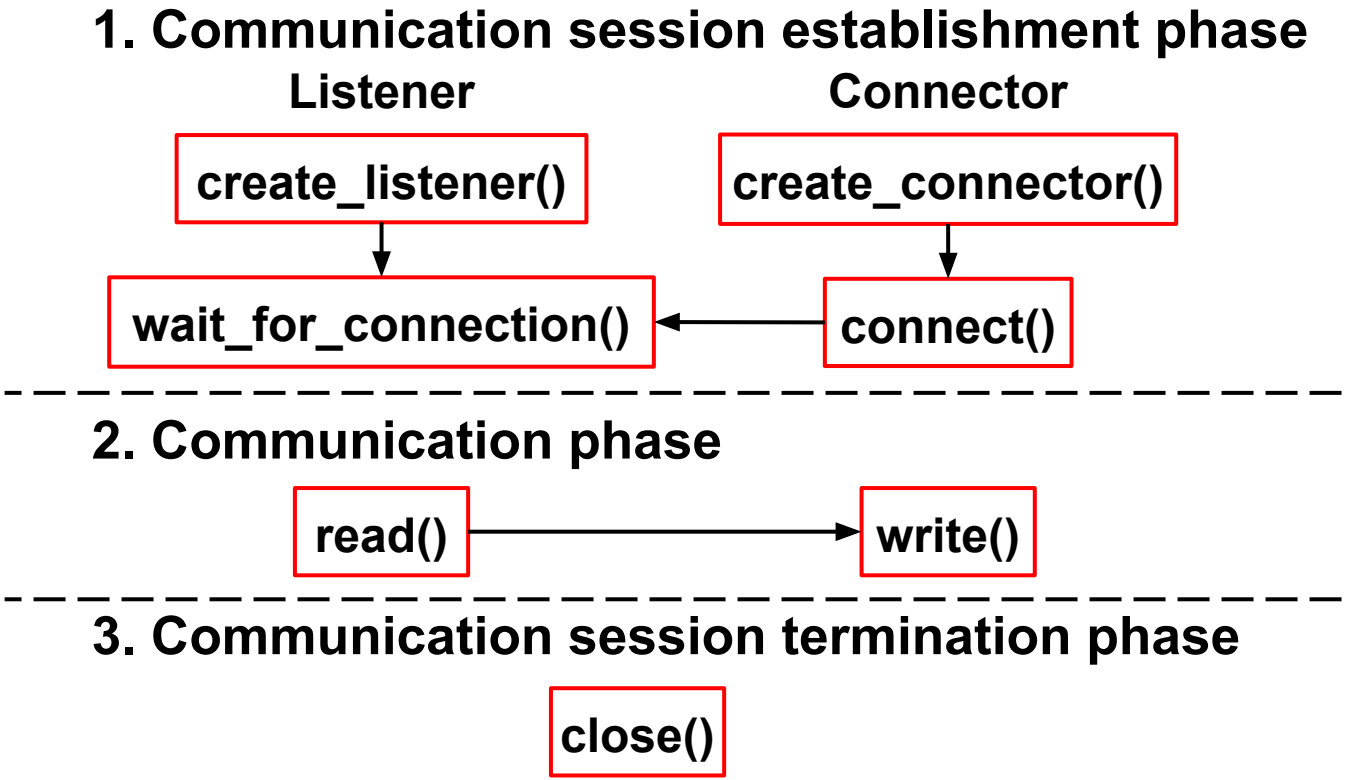
3. Proposed Approach Overview

- **Connector**: Requests connection.
- **Listener**: Accepts connection request.



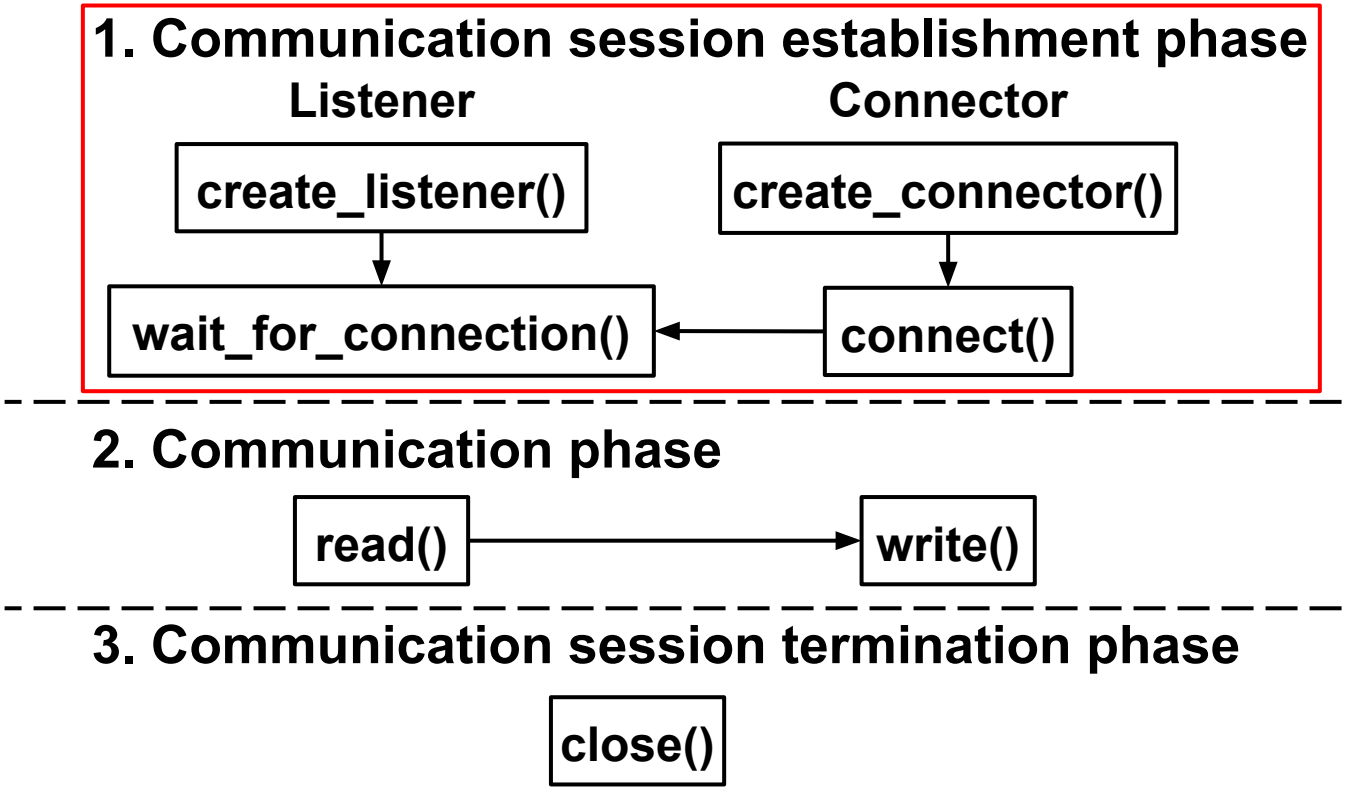
3. Proposed Approach Overview

- **Connector**: Requests connection.
- **Listener**: Accepts connection request.



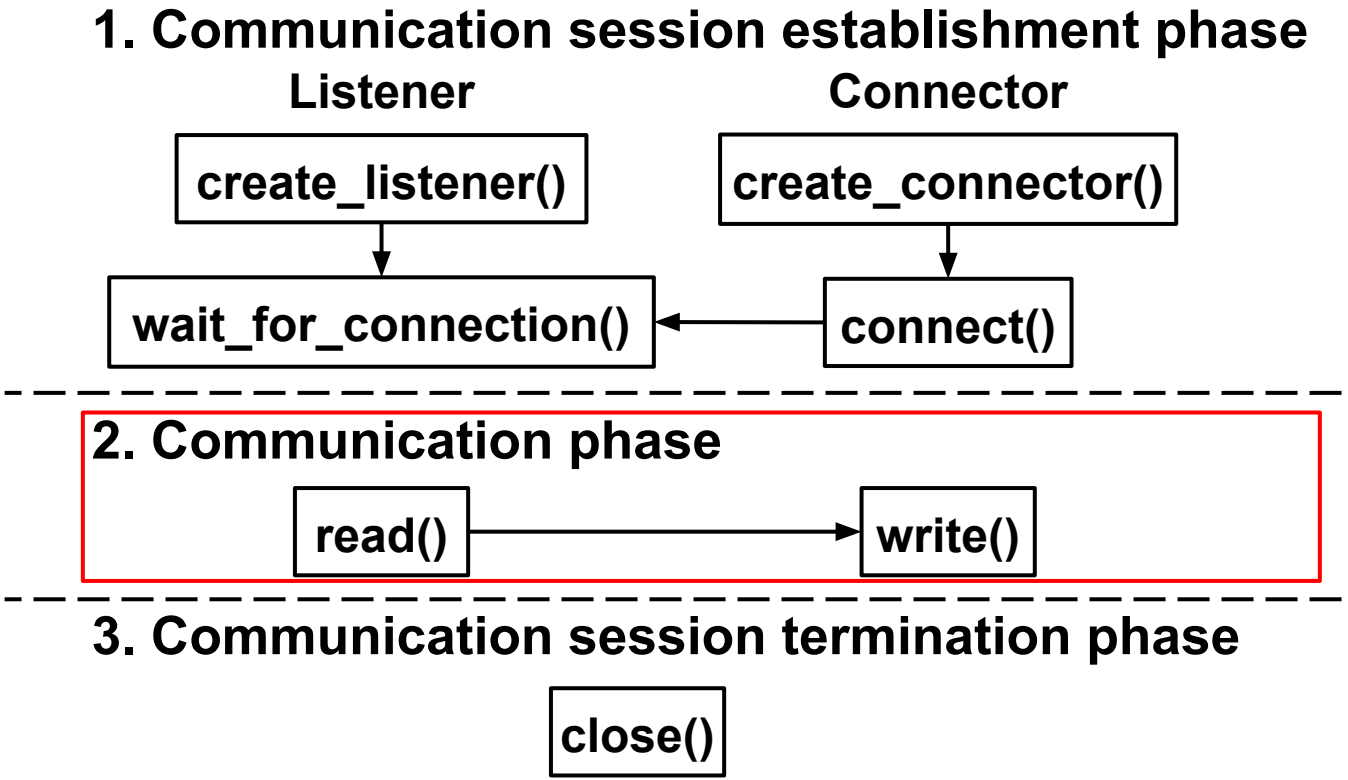
3. Proposed Approach Overview

- **Connector**: Requests connection.
- **Listener**: Accepts connection request.



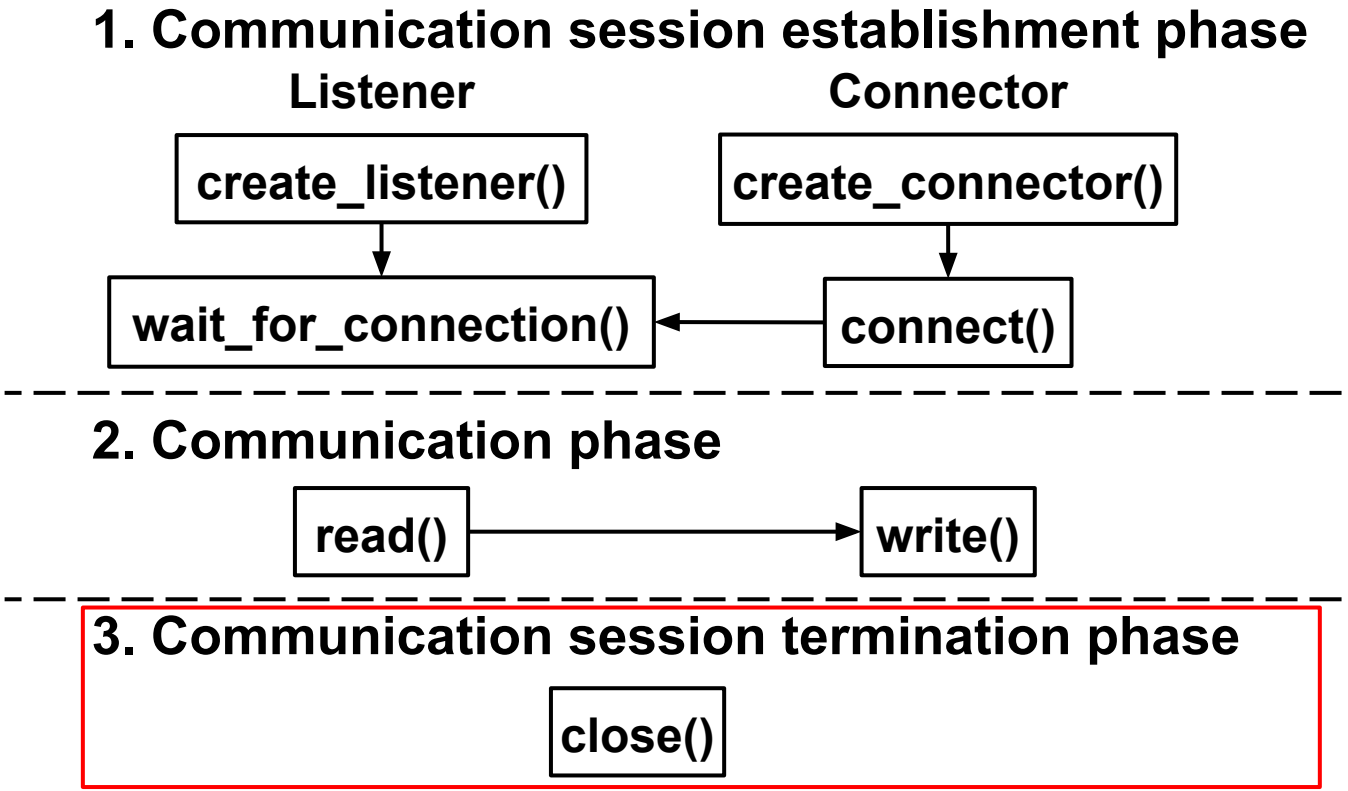
3. Proposed Approach Overview

- **Connector**: Requests connection.
- **Listener**: Accepts connection request.

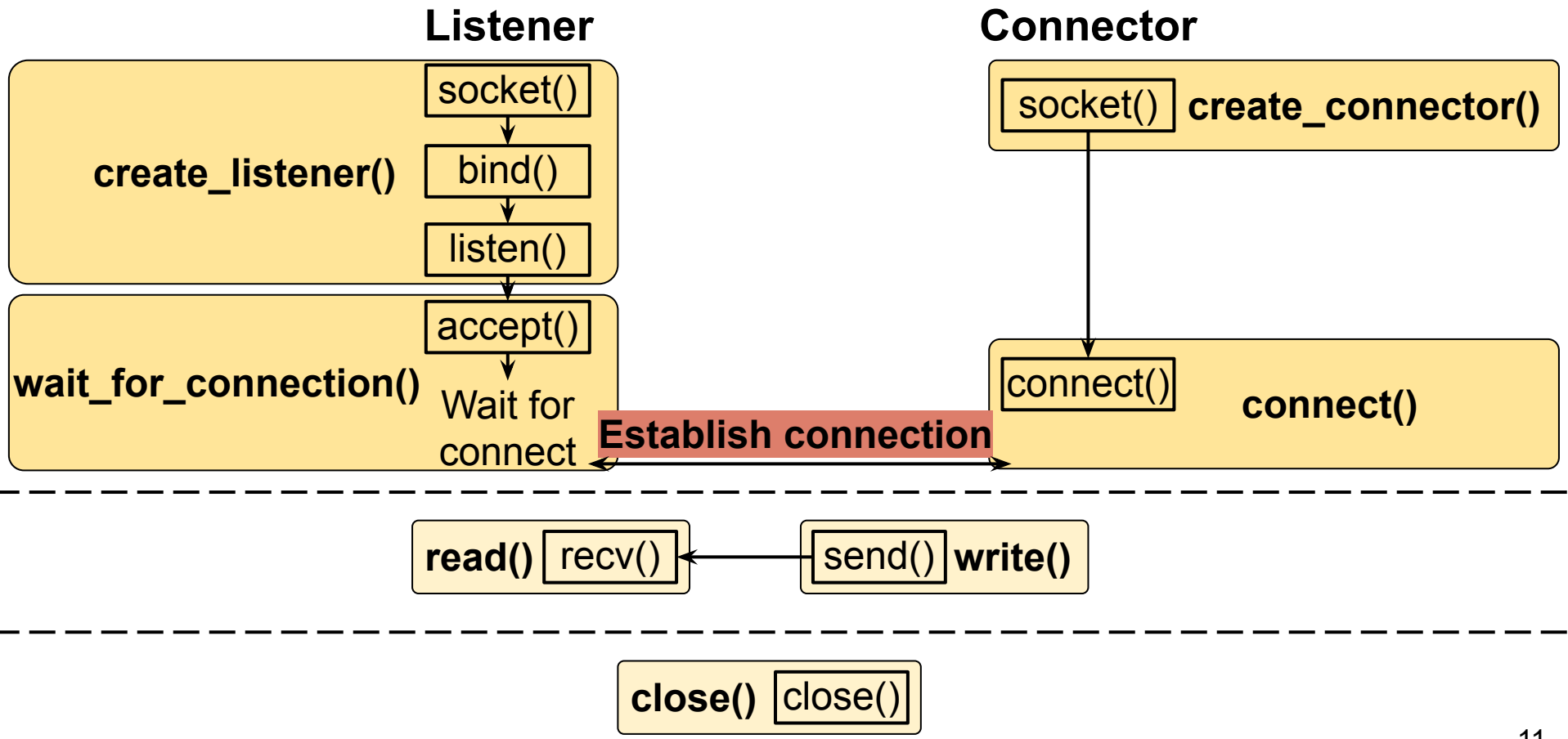


3. Proposed Approach Overview

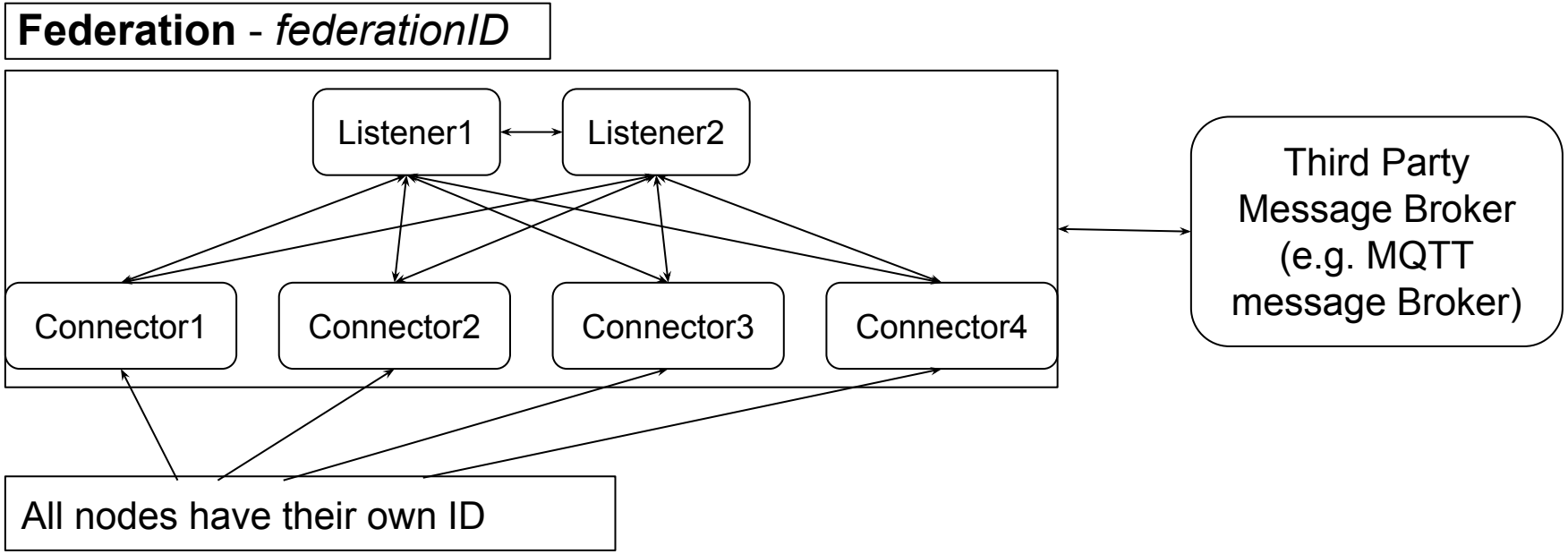
- **Connector**: Requests connection.
- **Listener**: Accepts connection request.



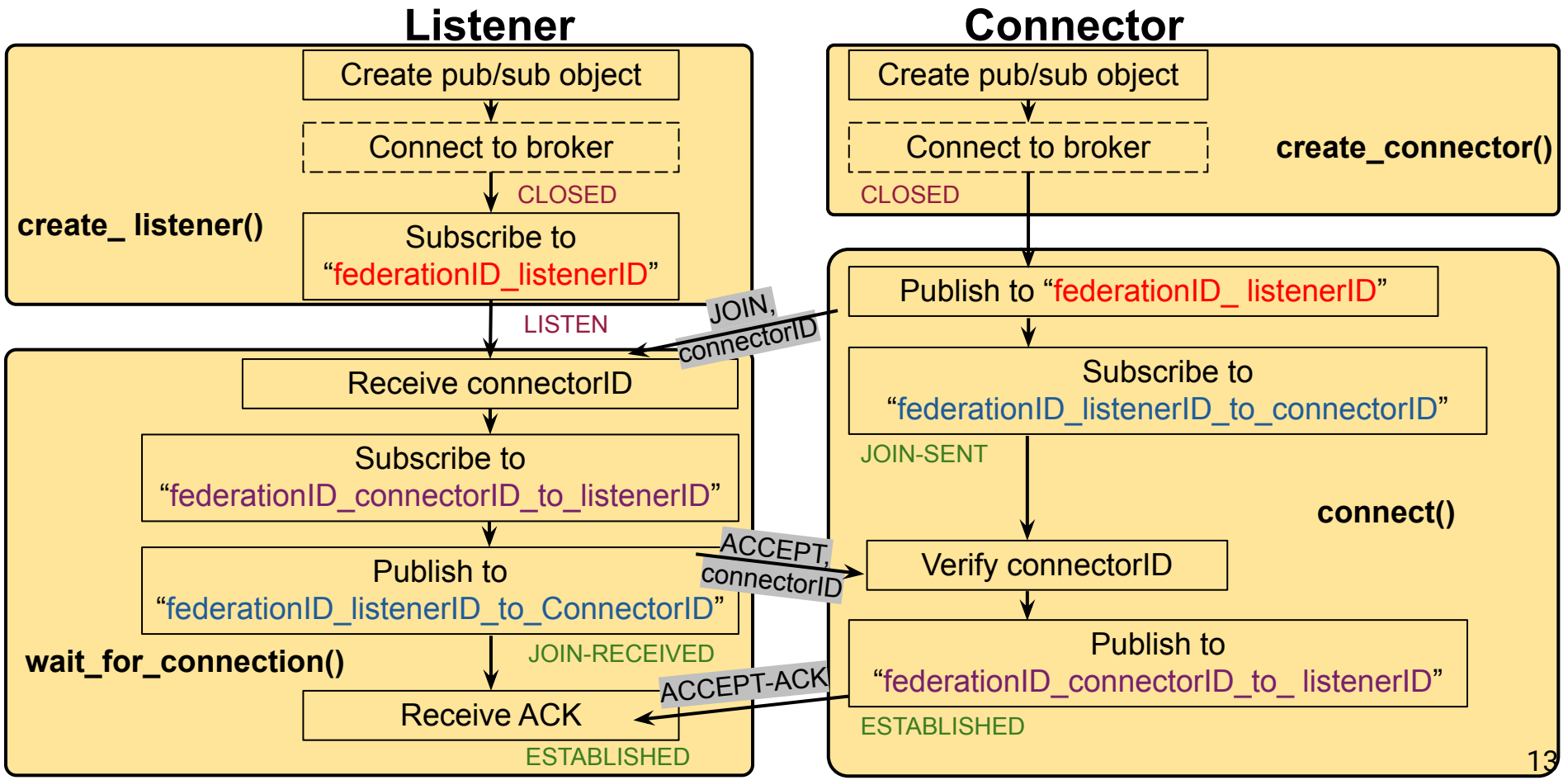
4. Point-to-Point Communication



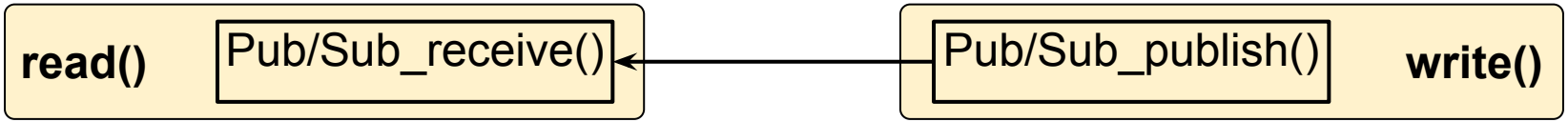
5. Federation

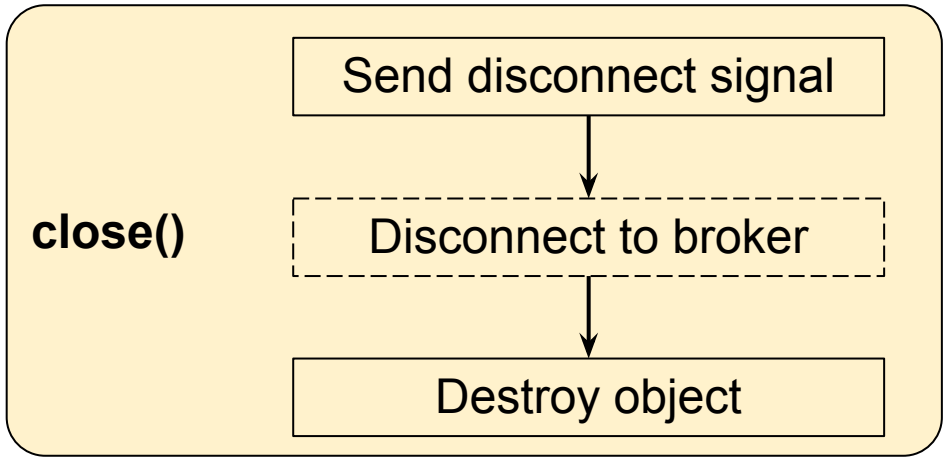


5. Publish-Subscribe Communication

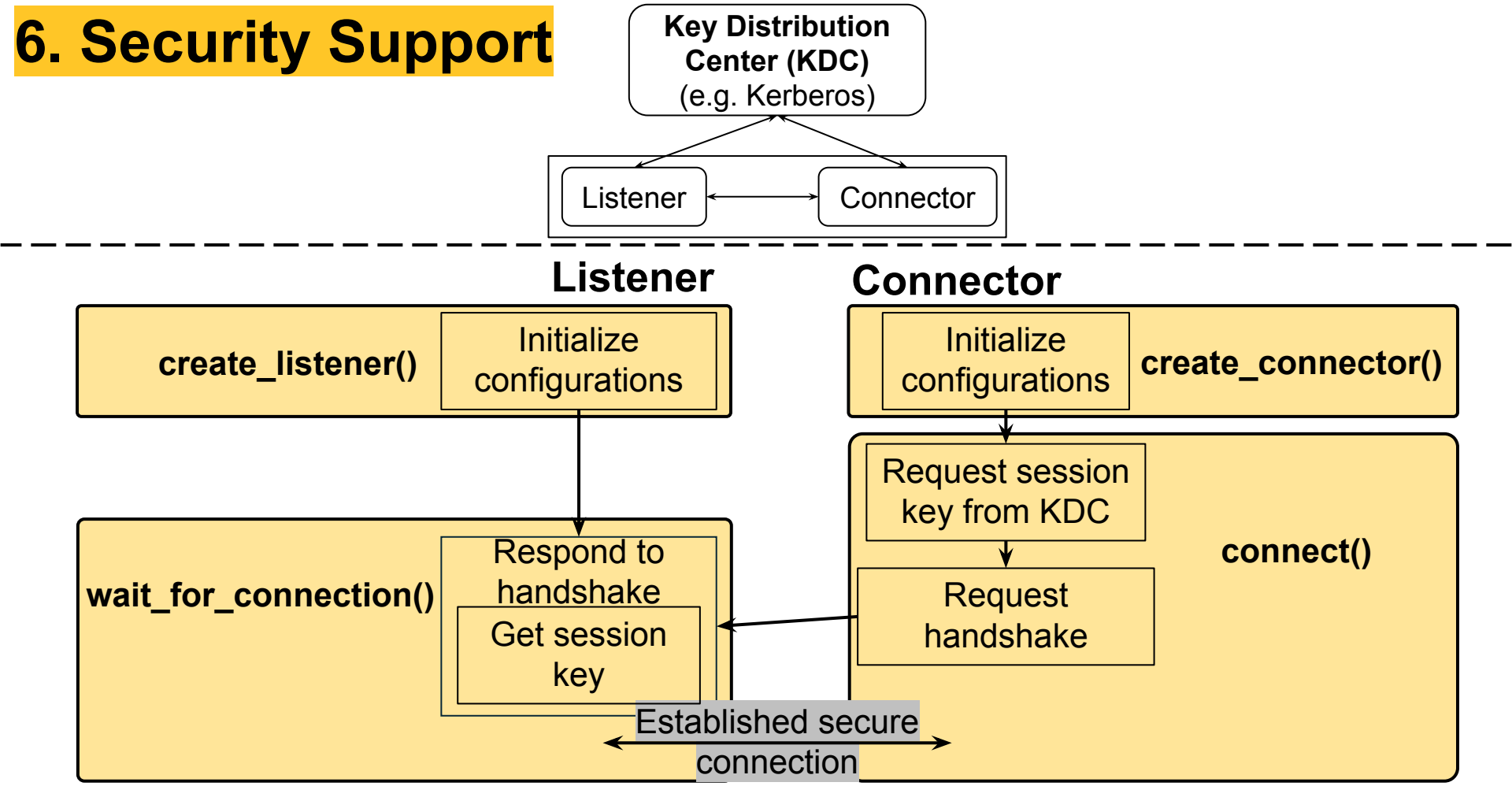


5. Publish-Subscribe Communication

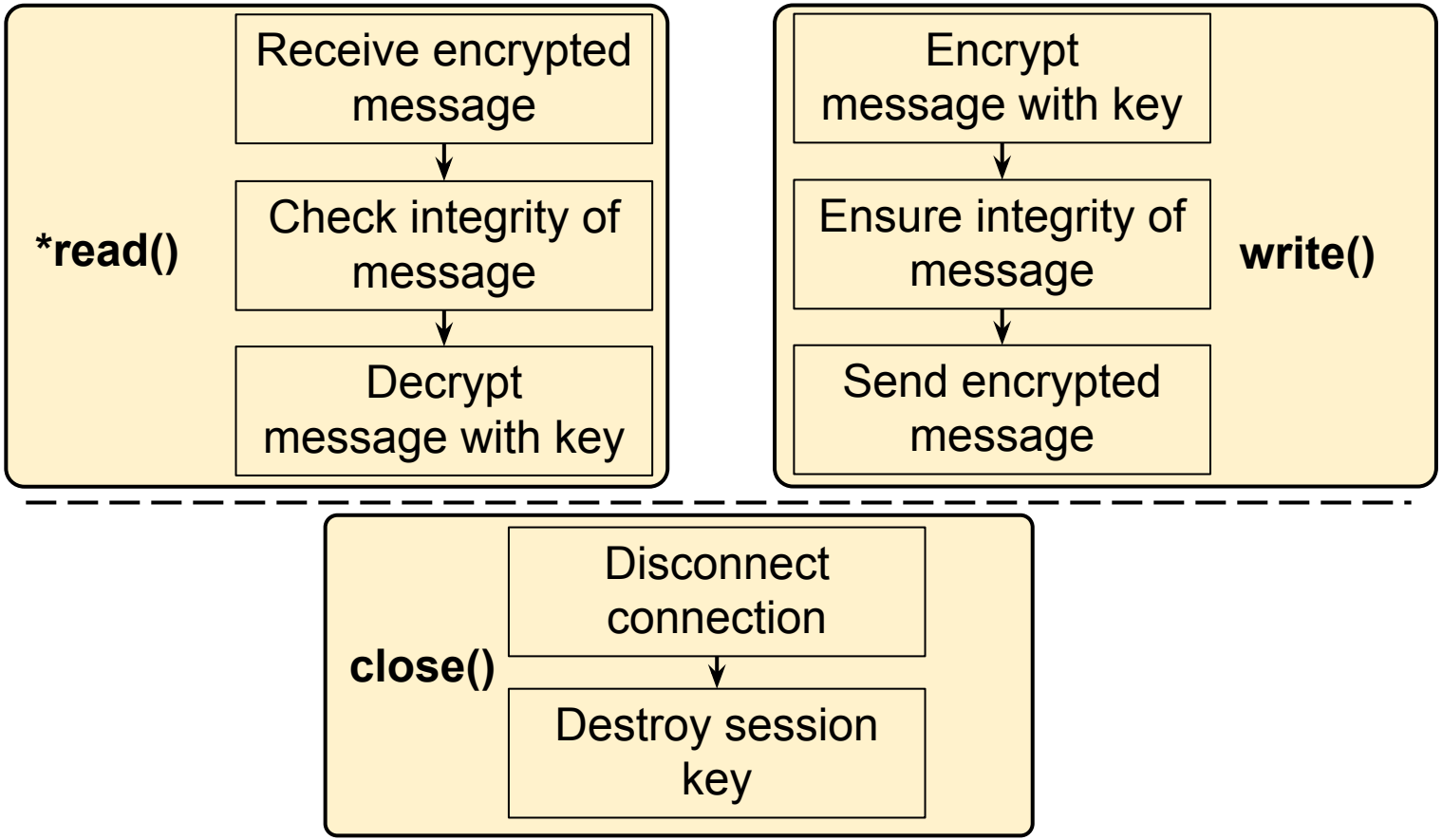




6. Security Support

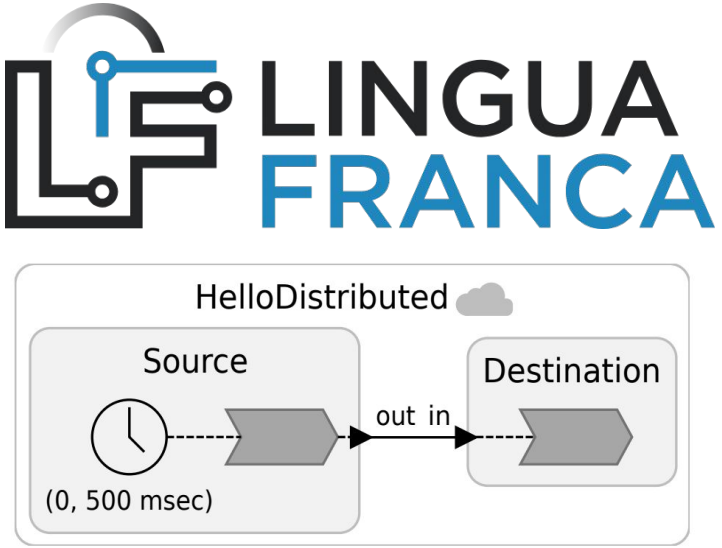


6. Security Support

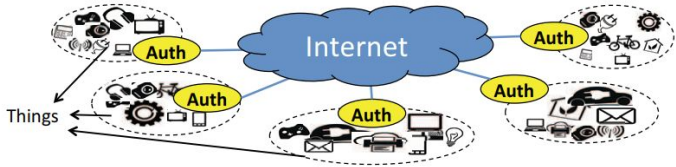


7. A Case Study: Design and Implementation

- **Lingua Franca:** https://github.com/lf-lang
  Coordination language designed to guarantee deterministic concurrency using *reactors*.
  The C runtime supports *federated execution* for distributed systems communicating over network.
  Compatible with embedded platforms including Arduino, Zephyr, and also bare metal devices.

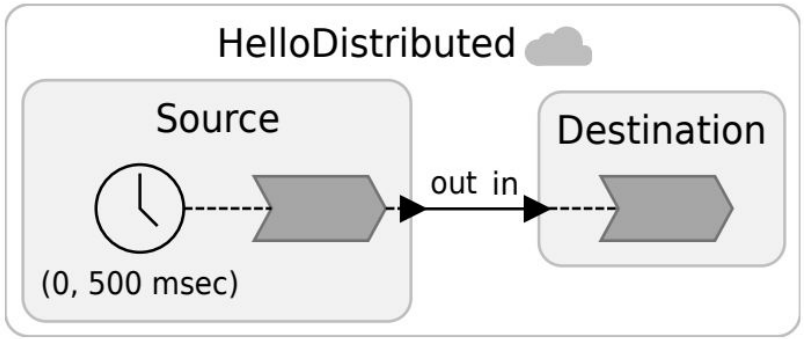


- **Secure Swarm Toolkit (SST):** https://github.com/iotauth/iotauth
  Provides authentication/authorization for its locally registered entities using local entity *Auth*.
  The C API supports resource-constrained devices [1].



[1] Kim, Dongha, et al. "SST v1. 0.0 with C API: Pluggable security solution for the Internet of Things." *SoftwareX* 22 (2023): 101390.

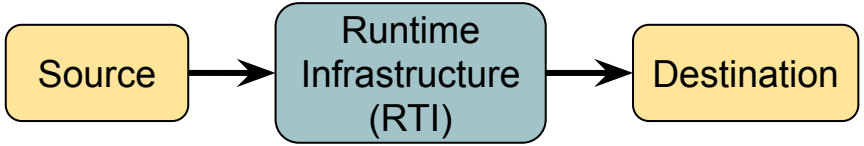
8. Example Program for Experiments



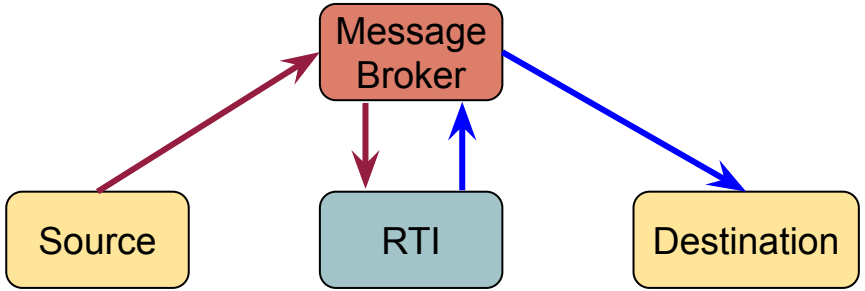

```
target C {
  coordination: centralized,
  comm-type: MQTT,
  timeout: 500 sec,
  auth: true
}

reactor Source {
  output out: int
  timer t(0, 500 msec)
  reaction(t) -> out {=
    lf_set(out, 0);
    =}
}

reactor Destination {
  input in: int
  reaction(in) {=
    lf_print("Dest received: %s", in->value);
    =}
}

federated reactor HelloDistributed{
  s = new Source()
  d = new Destination()
  s.out -> d.in
}
```

8. Experimental Scenarios

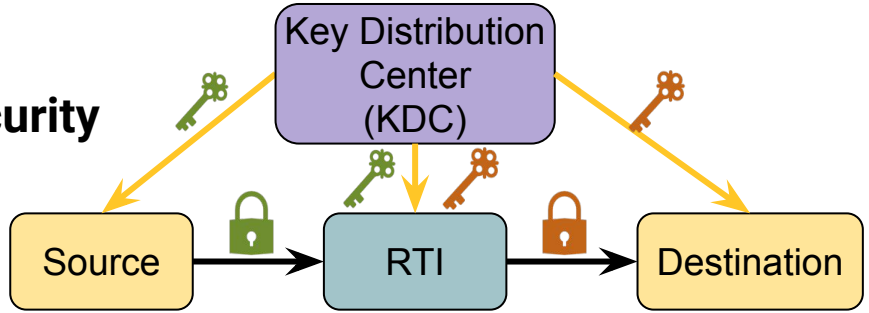
1. TCP



2. MQTT



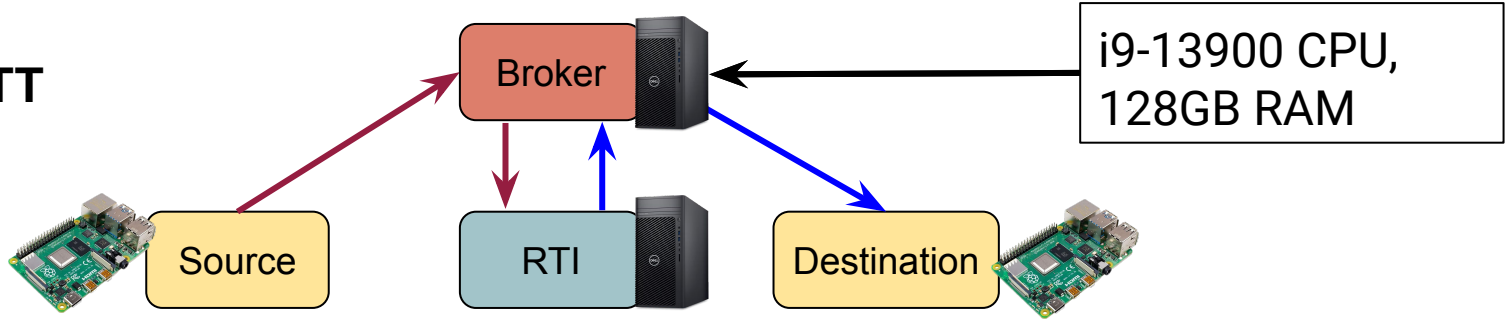
3. TCP + Security

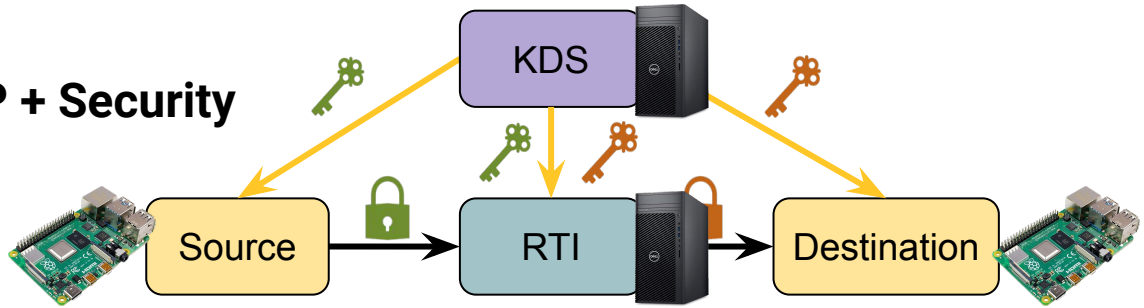


8. Experimental Setup

Wi-Fi end-to-end round-trip latency :
13.60 milliseconds.

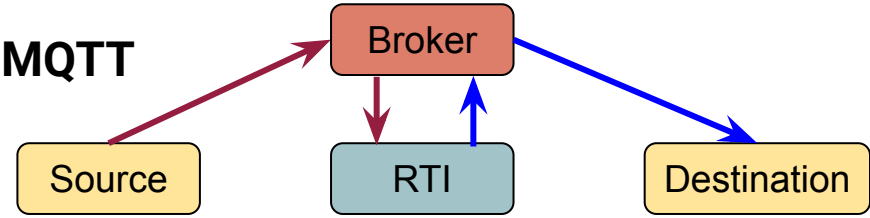
Raspberry Pi 4B
(4GB RAM)



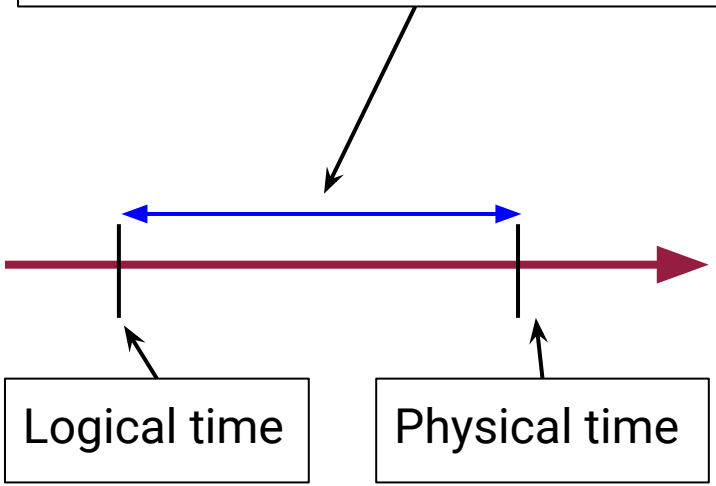


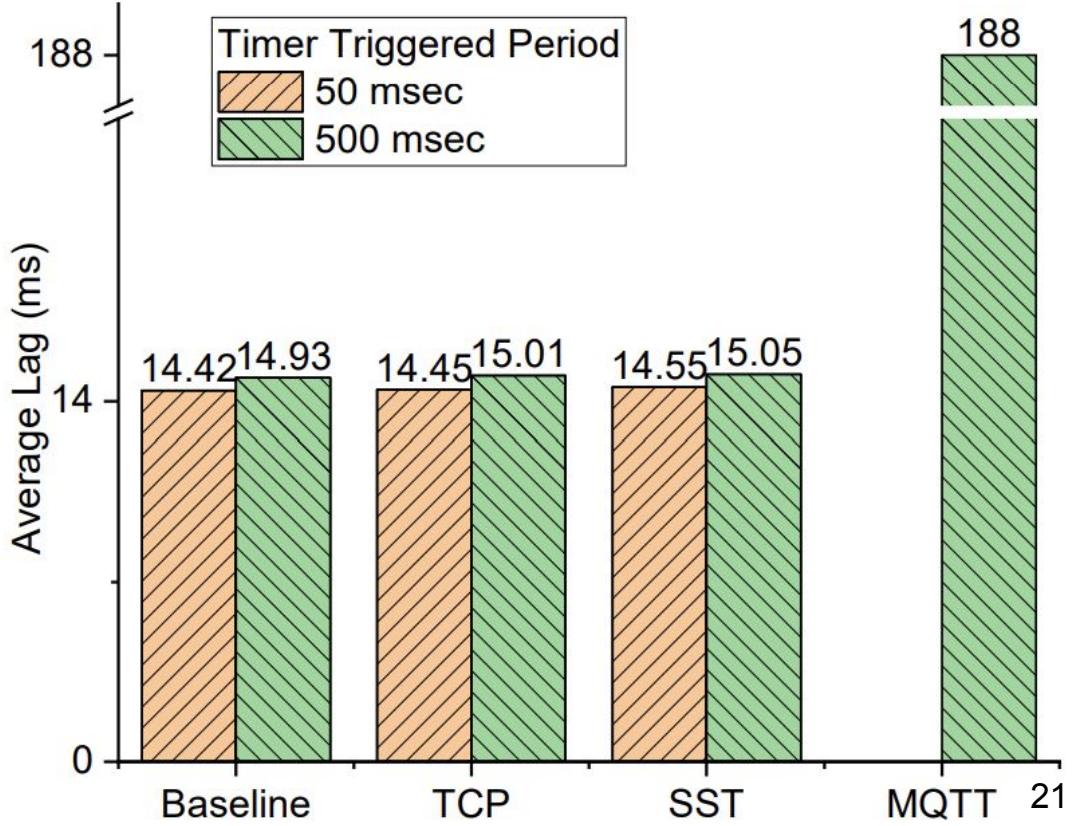
9. Average Lag

2. MQTT

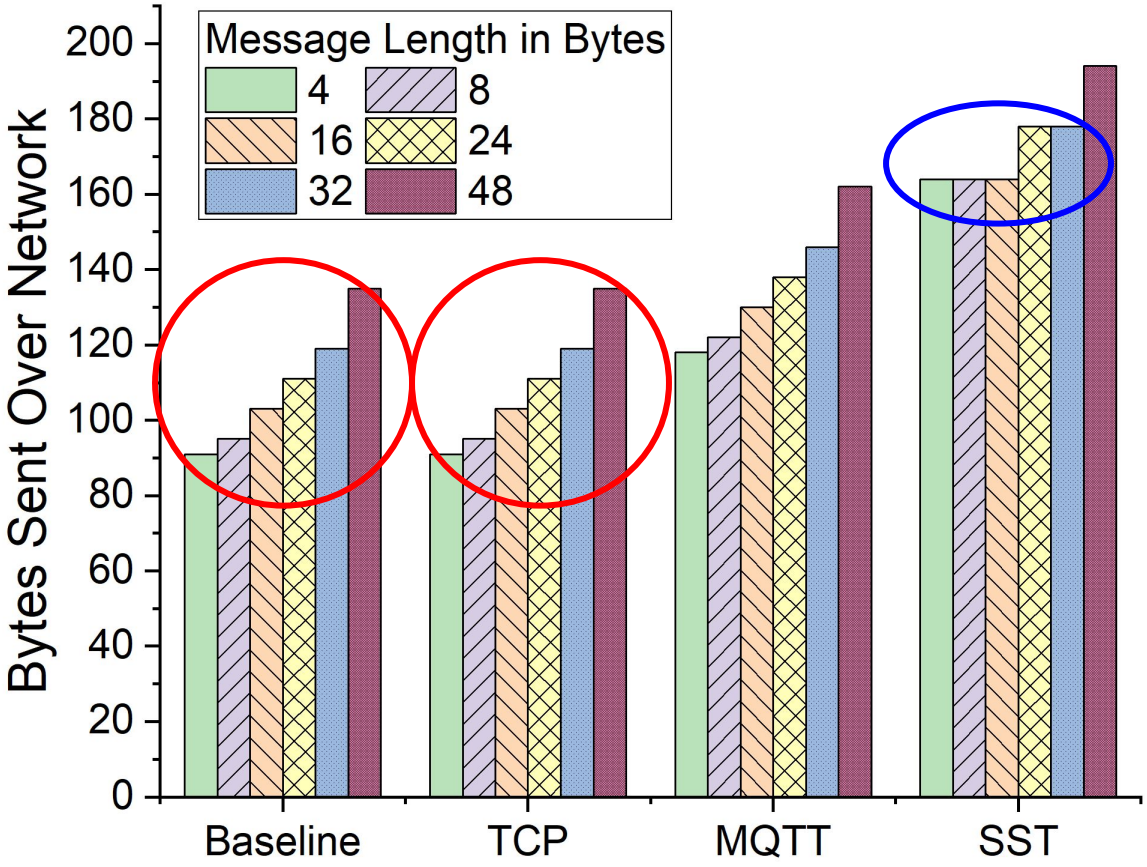


LAG:
Time lapse between system clock (physical clock) and agreed global time (logical clock)



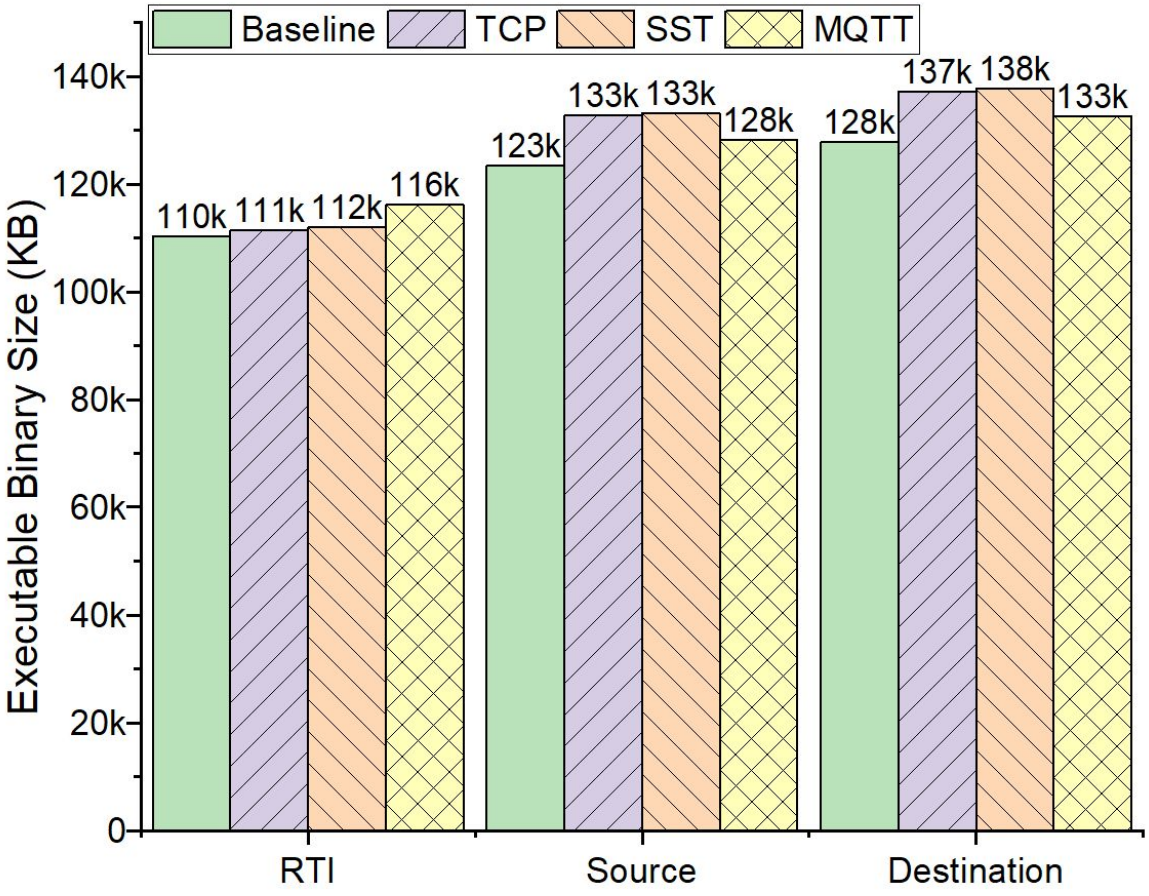


10. Message Length Sent In Bytes



- Does not add additional bytes excluding bytes added from the protocol itself.
- AES-CBC mode protects side channel attacks such as inferring message by the message length.

11. Binary Size Overhead



- **RTI:** 1%-5% Overhead
- **Nodes:** 3%-7% Overhead
- Overhead mostly comes from the compilation of the network abstract layer as a separate library.

Thank You For Your Attention!

Summary

- Proposes an API and runtime for interoperability and security in IoT and distributed CPS.
- Implements seven core API functions using open-source frameworks (Lingua Franca and SST) as a case study.
- Evaluates communication time overhead, message size, and binary size, showing minimal overhead.
- Plans future support for additional communication modes, federation of diverse nodes, and fine-grained security configurations.

Acknowledgement

Supported by NSF I/UCRC (IDEAS), NSF grant #2231620, and ATTO Research.

Contact Information: https://jakio815.github.io/, dongha@asu.edu, https://labs.engineering.asu.edu/kim/

Arizona State University