Privaros: A Framework for Privacy-Compliant Delivery Drones

Rakesh Rajan Beck, Abhishek Vijeev, Vinod Ganapathy
Indian Institute of Science

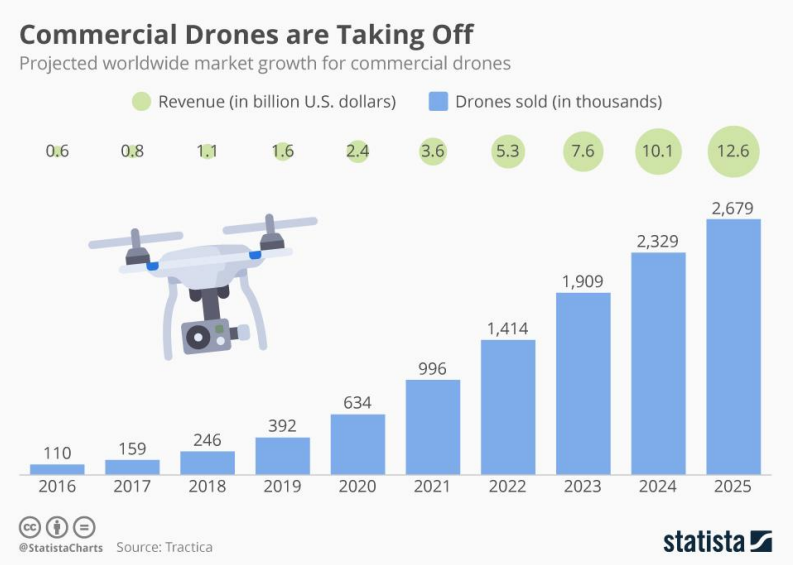
ACM CCS '20

GyeongHeon Jeong(ghjeong@mmlab.snu.ac.kr)

Index

- Introduction
- Background
  - ROS
  - MAC
- Details of Privaros
  - Policy Specifications
  - Countermeasure of ROS Shortcomings
  - Role of Hardware TEE
  - Integration with Platform
- Evaluation
  - Robustness of Policy Enforcement
  - Performance
- Conclusions & critique

Introduction



- Drones are now commonly available
- Drones are equipped with sensors(e.g., Camera, GPS)
- The more drones are used, the greater threat to individual privacy
- Researches to enforce privacy in drone are lacking

Introduction

“Privaros” - A framework that ensures guest drones are compliant with privacy-policies specified by host airspaces

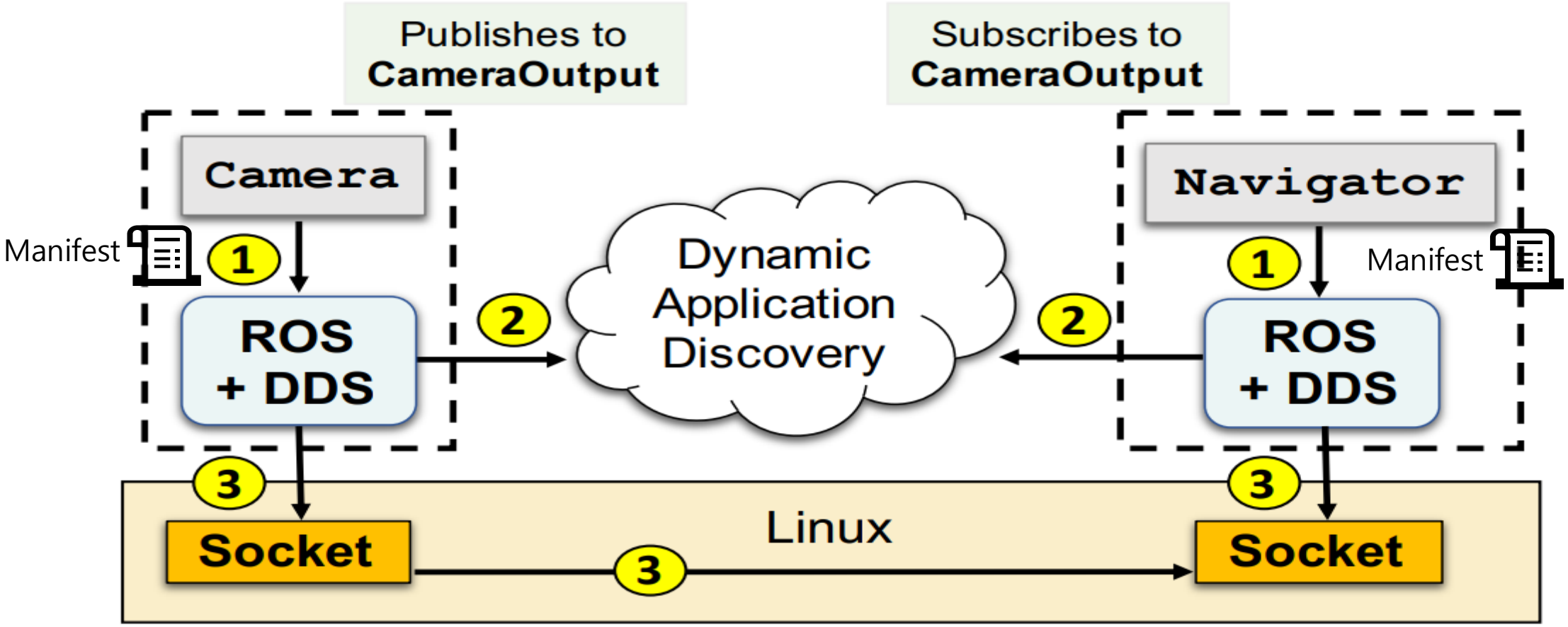
- Focuses on delivery drones(*exclude rogue drone cases*)
- Enforces policies with Mandatory Access Control (MAC) in the OS and Robot Operating System (ROS) layer
- Uses hardware-based attestations from Trusted Execution Environment (TEE)
- Can be integrated with the ‘Digital Sky portal’ interface

ROS (Robot OS)

- ROS applications communicate with Publish/Subscribe system
  - All applications publish or subscribe the topic
  - Each application registers manifest which contain their topic list

- Data Distribution Service (DDS) helps ROS`s Publish/Subscribe system
  - DDS implements Distributed protocol

ROS (Robot OS)



Publisher/subscriber communication in ROS

ROS (Robot OS)

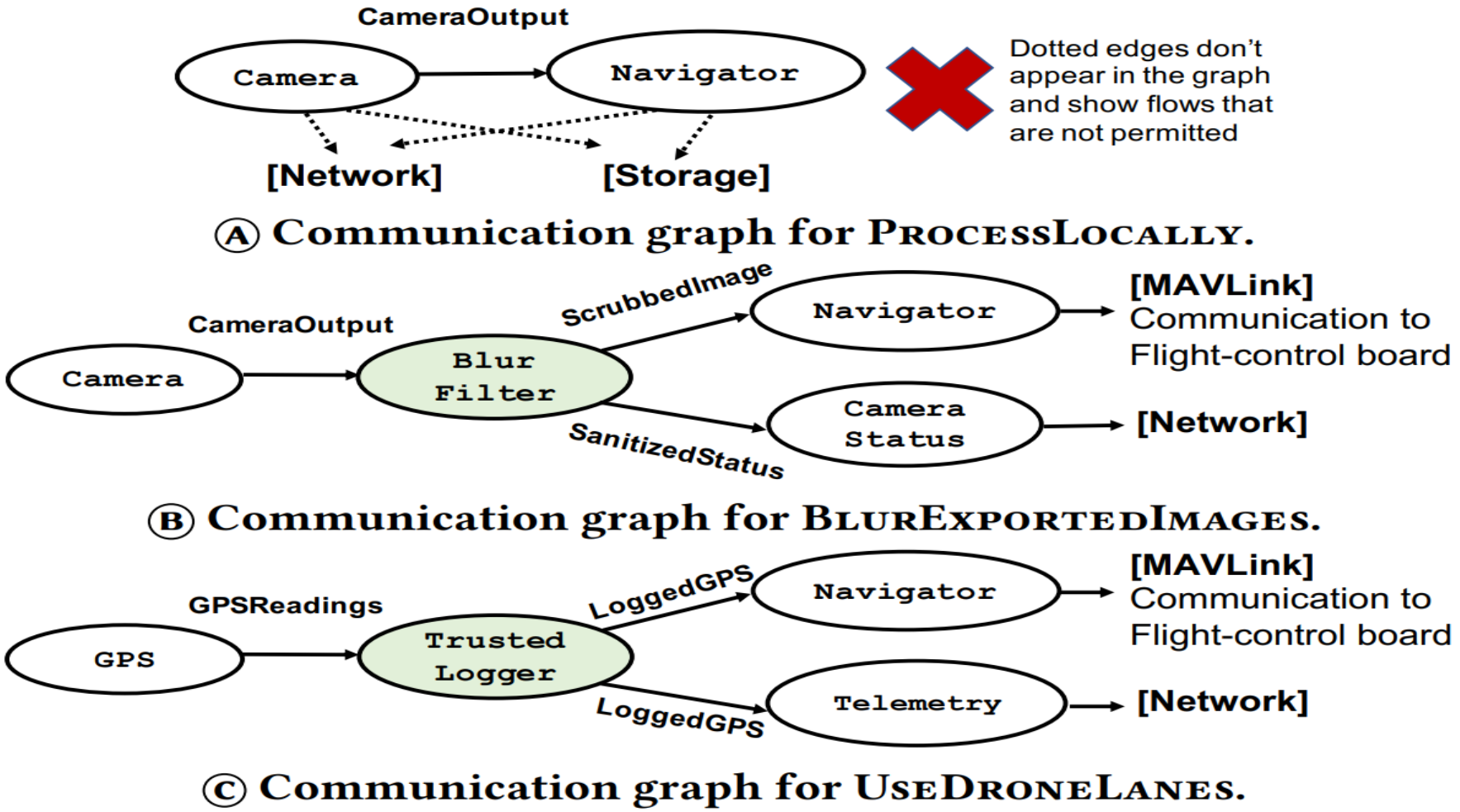
- SROS (Secure ROS) is ROS secure extension
  - Using cert. and TLS

- ROS Shortcomings with Privaros
  - ROS can be bypassed with direct OS-communication
  - Manifest contains only ‘Topic’, not ‘Type’
    - Topic : CamOutput
      - Type 1 : CamOutput::ImageType – Raw image
      - Type 2 : CamOutput::StatusType – State of camera

MAC (Mandatory Access Control)

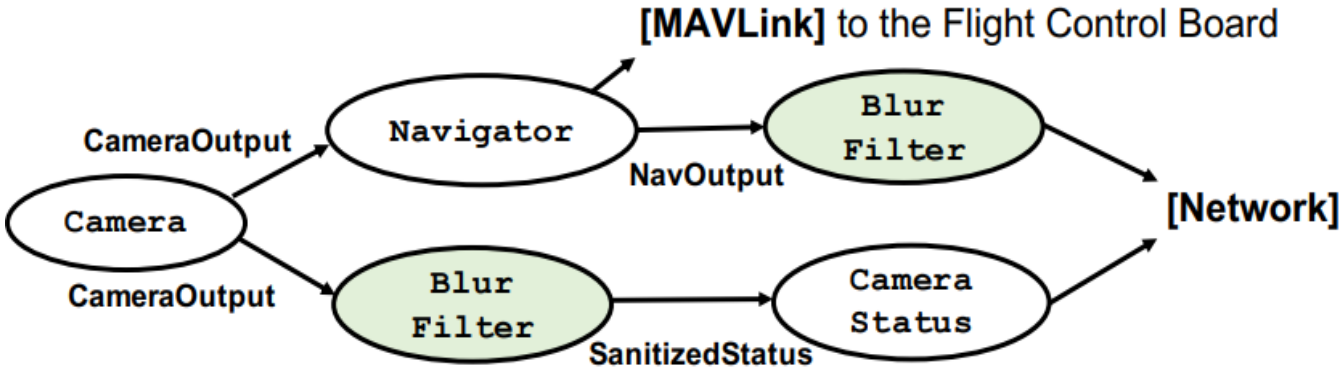
- MAC is type of access control which constrains to access objects
- MAC uses label to decide whether object is accessible or not
  - Label is determined only by the system's top administrator
- Privaros uses MAC to ensure communication should proceed by rules of Privaros

Policy Specifications



Policy Specifications

- Communication graphs are carefully designed by Privaros
  - Privaros makes all communication graph one by one
  - Suppose navigator application wants to send to network, and take raw image
  - Like upper case, It`s difficult to design graph without error
    - Thus, they made 'tool' which extract all current communication graph in drone
    - The tool is similar with *'audit2allow'*



Communication graph for BlurExportedImage

Countermeasure of ROS Shortcomings

- *Problem)* ROS can be bypassed with direct OS-communication
  - Those app cannot be applied policy, because they do not use publish/subscribe system

- *Solution)* MAC can block direct OS-communication
  - Privaros sets label not to access objects associated OS-communication
  - Label restricts all creation, reading, and writing of socket, shared memory, IPC, pipe, file systems related to OS-communication

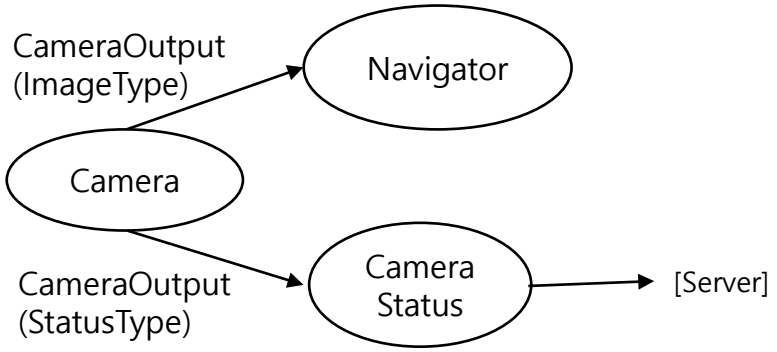
- *Result)* All app connection are composed of publish/subscribe system
  - Thus, MAC robustly ensures all apps are under policy

Countermeasure of ROS Shortcomings

- *Problem)* Manifest contains only ‘Topic’, not ‘Type’
  - Not being able to determine what ‘Type’ the app reads makes it difficult to implement

- *Solution)* Both ‘Topic’ and ‘Type’ is assigned for each connection edge in communication graph
  - Apps publish/subscribe its type along set edge

- *Result)* Difficulties to implement are resolved

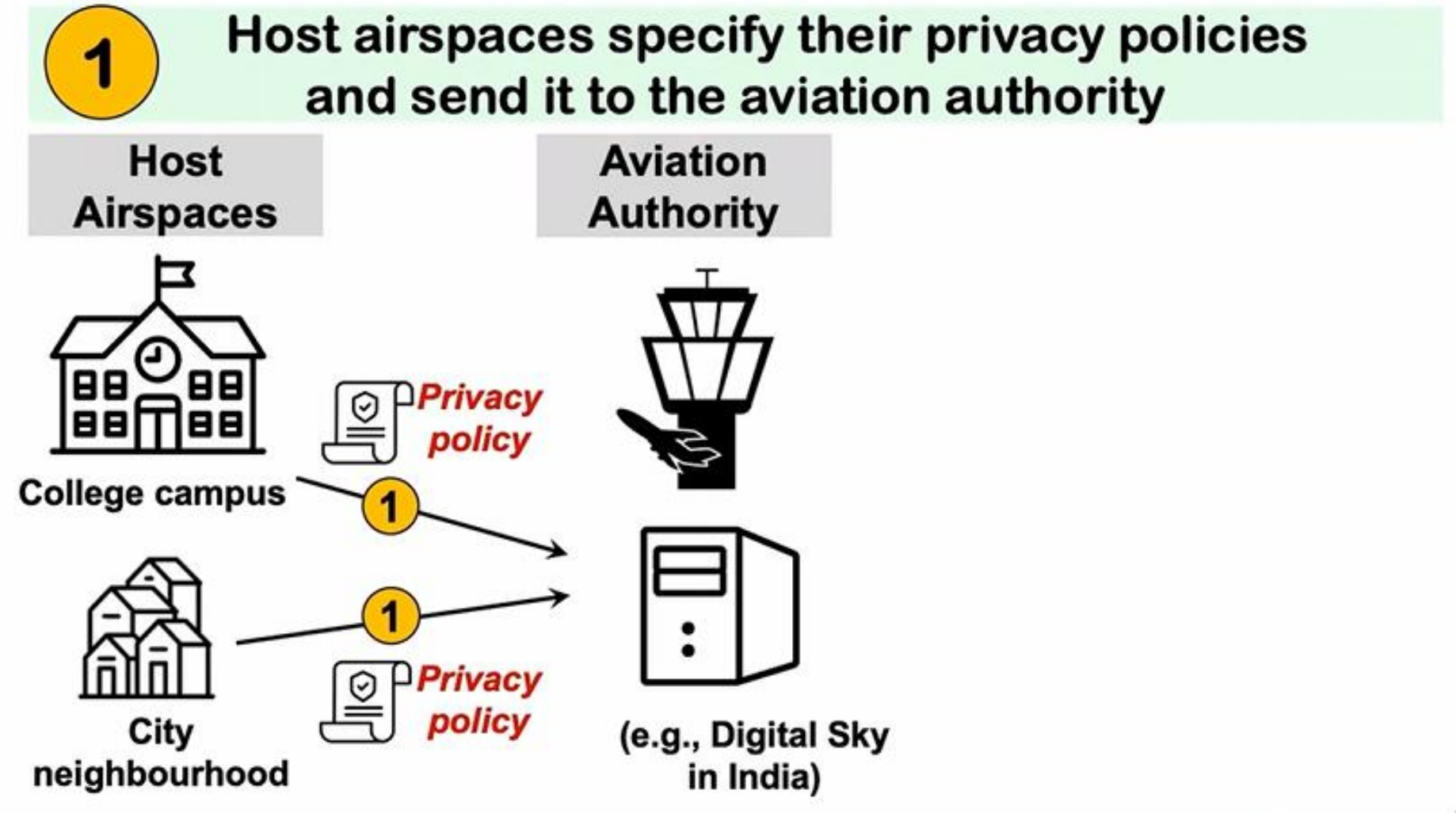


Communication graph with 'Type'

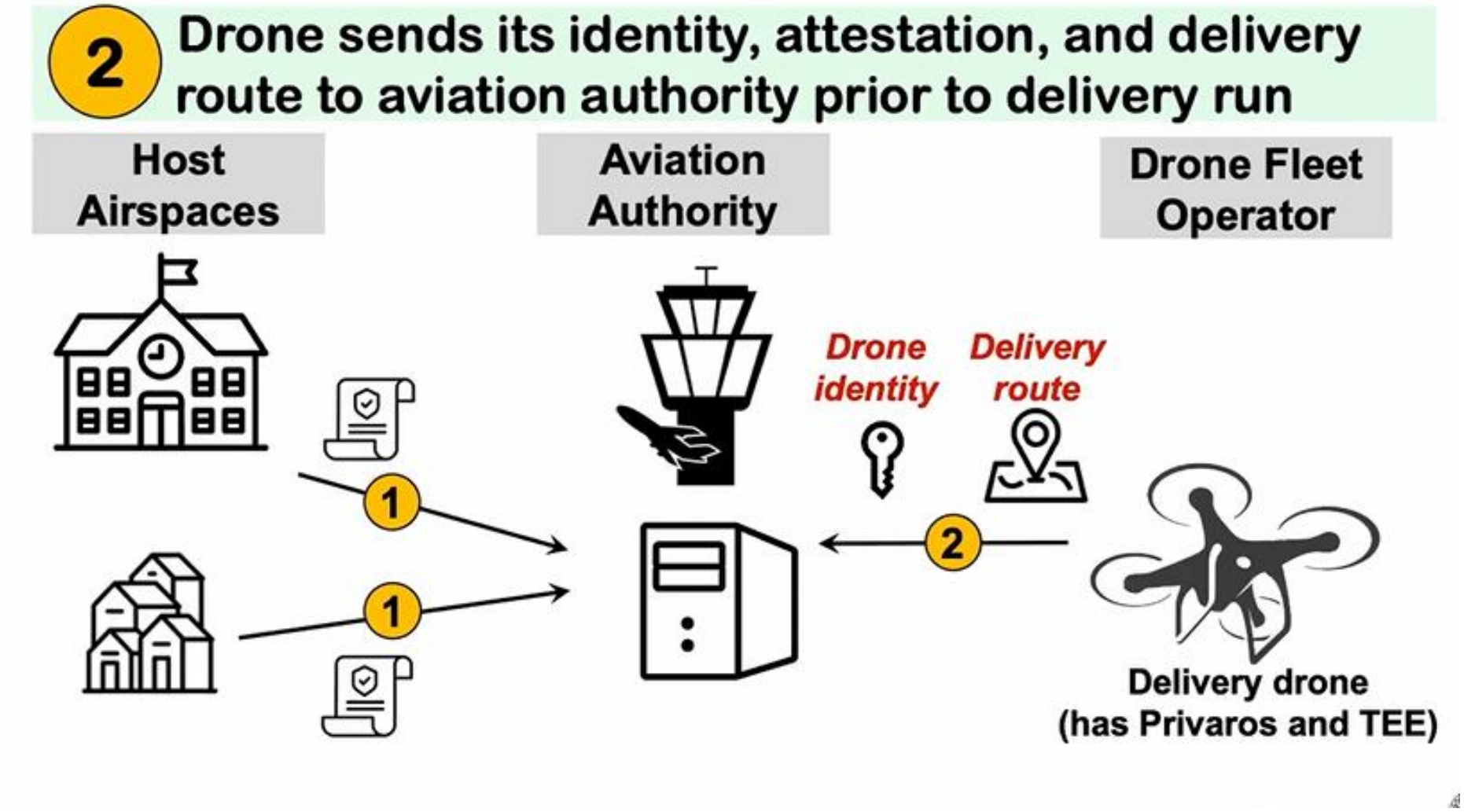
Role of the Hardware TEE

- In Privaros, they use Trusted Execution Environment (TEE) for attestation
  - Secure booting can obtain hash chain of normal booting software
  - Comparing hash chain to current hash chain verifies whether it modifies or not
- Log(e.g., GPS) can store in secure hardware not to be modify
  - Host are reliable to log in TEE
- With TEE, Privaros finally can prove they have robustly enforced policy

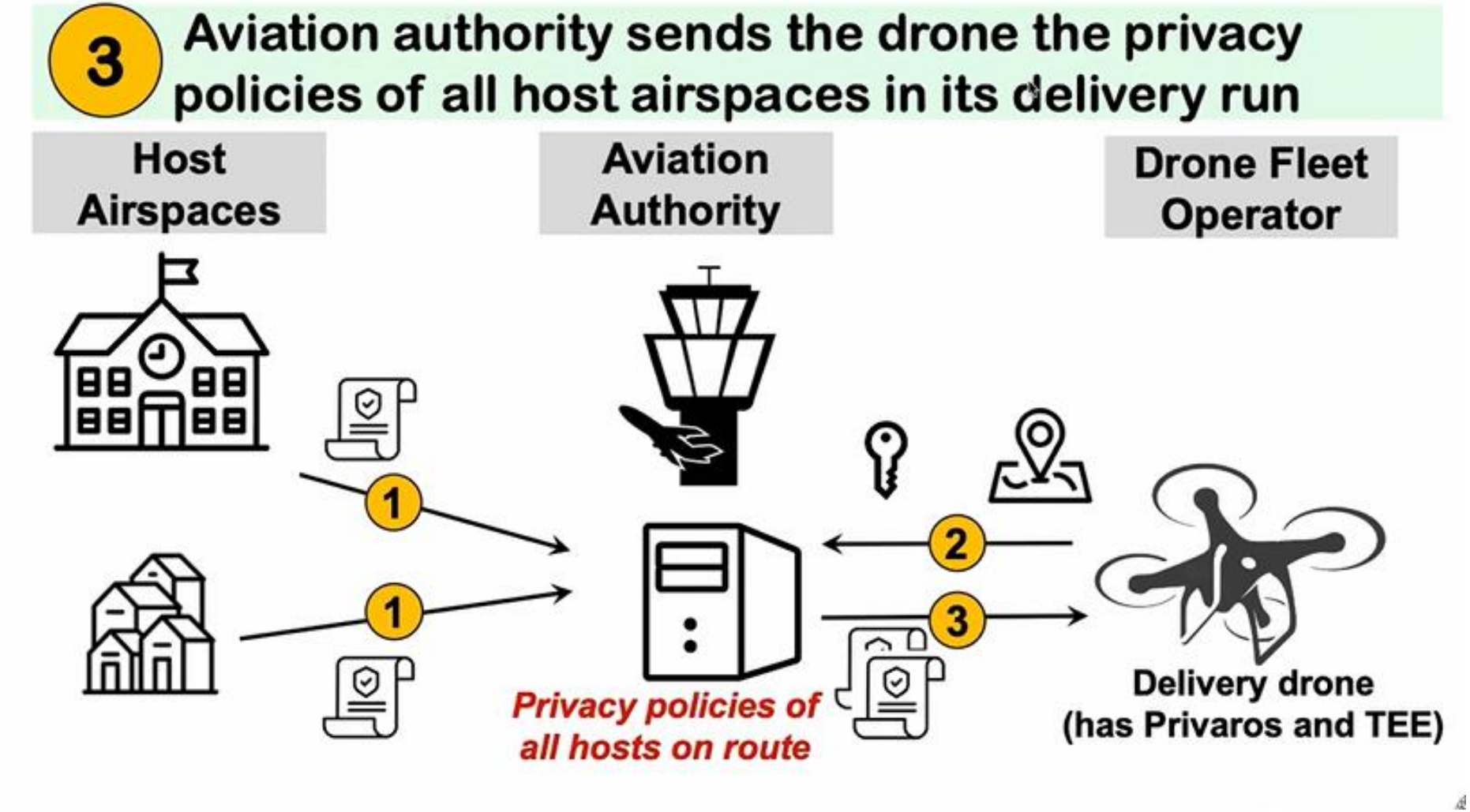
Integration with Platform



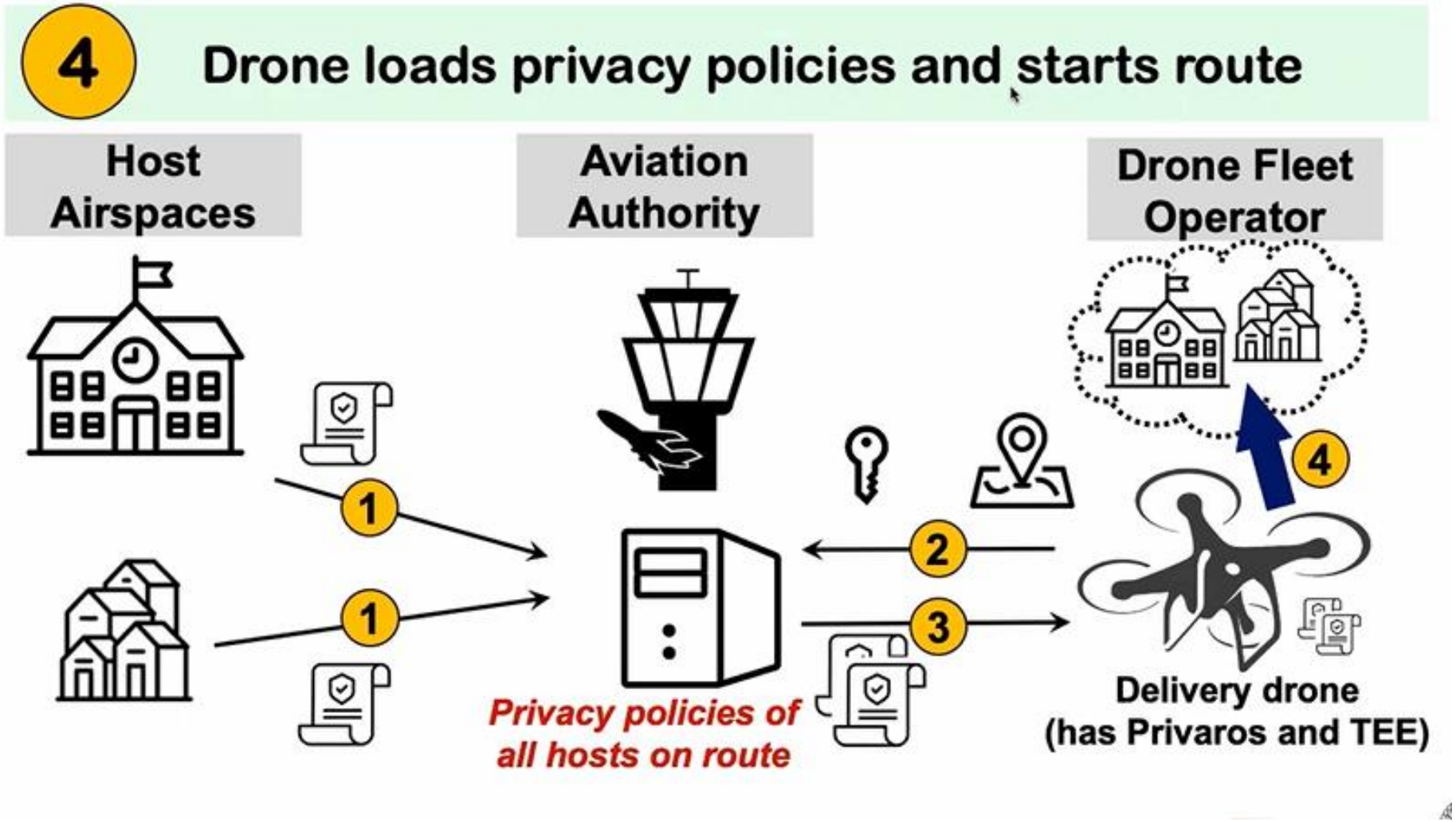
Integration with Platform



Integration with Platform

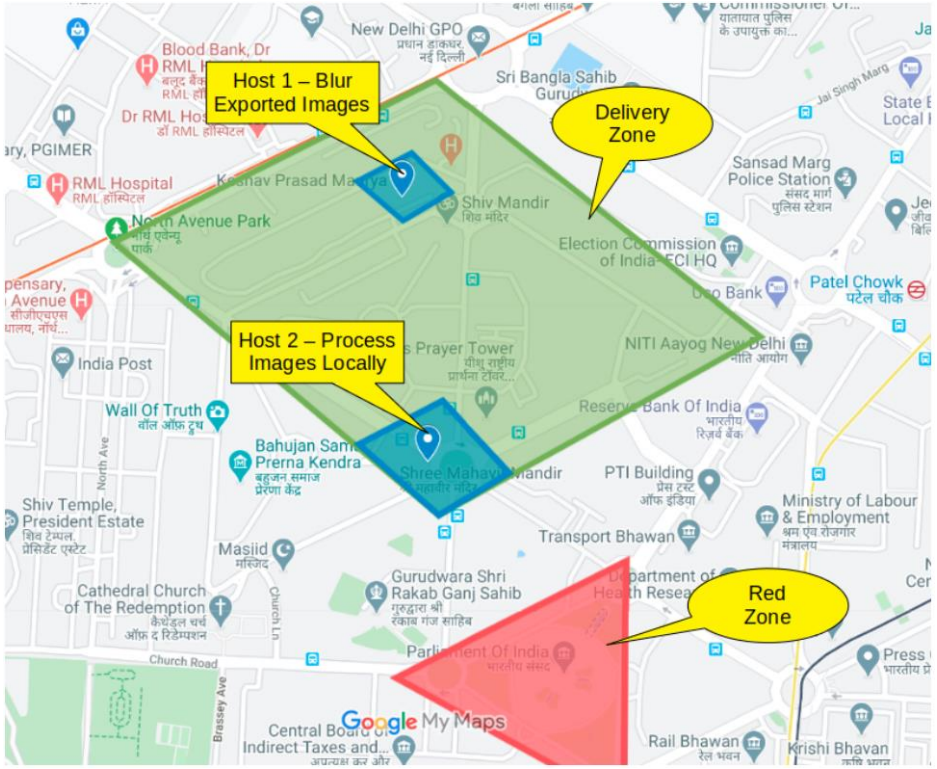


Integration with Platform



Integration with Platform

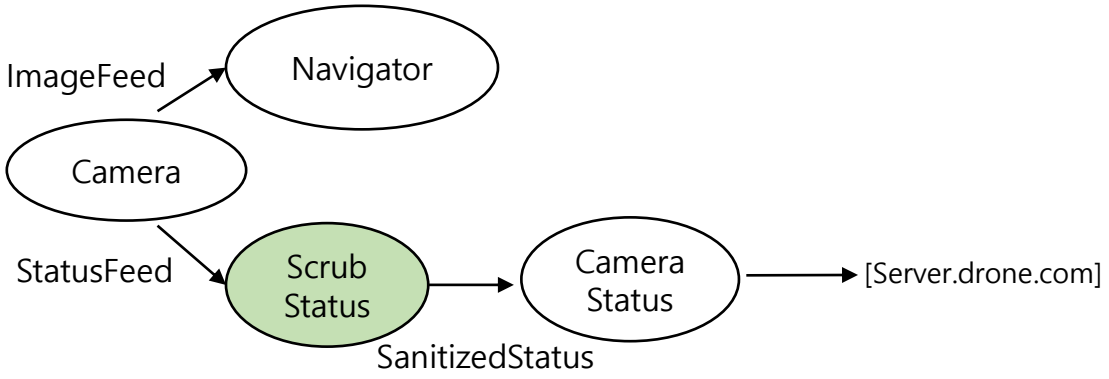
- India's Digital Sky portal
  - Portal uses visual map interface
  - Privaros can expend this interface
  - Benefit of integration
    - Host)
      - Simplification of policy setting UI
    - Privaros)
      - Exercise with all registered drone
      - Pre-computed communication graph



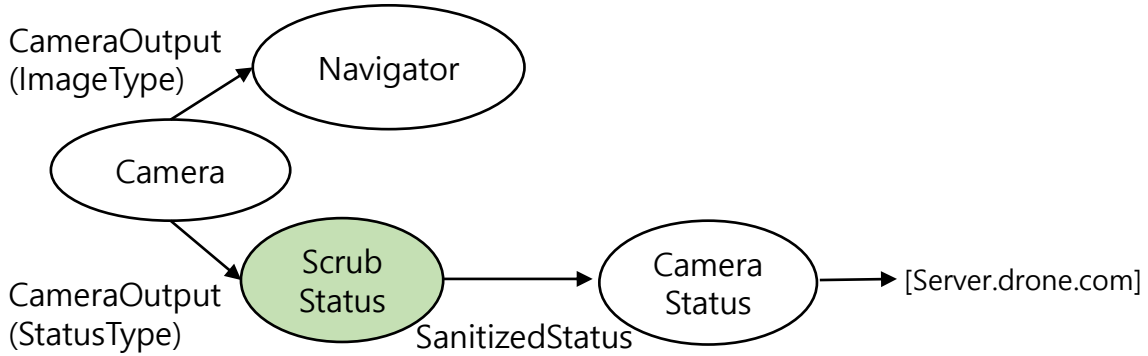
Screenshot of Digital sky portal

Robustness of Policy Enforcement

- Separated ‘Topic’
  - In ROS, 2 'Topics’ are needed
    - Manifest show only ‘Topic’, not 'Type’
  - In Privaros, only 1 ‘Topic’ is still needed
    - 'Type’ can be read



(A) With ROS



(B) With Privaros

Robustness of Policy Enforcement

- Direct communication via OS
  - ROS cannot control OS communication
  - Privaros restrict OS communication with MAC

Performance

Workload	Type of data published/subscribed
`Array`	Simple byte array
`PointCloud`	Set of N-dim. points *e.g.*, 2D images from camera depth sensors)
`Struct`	Structure holding a set of bytes (*e.g.*, 16 bytes in `Struct16`)
`NavSat`	Status of navigation satellite
`Range`	Single range reading obtained from a range sensor

Workloads from PerformanceTest[2]

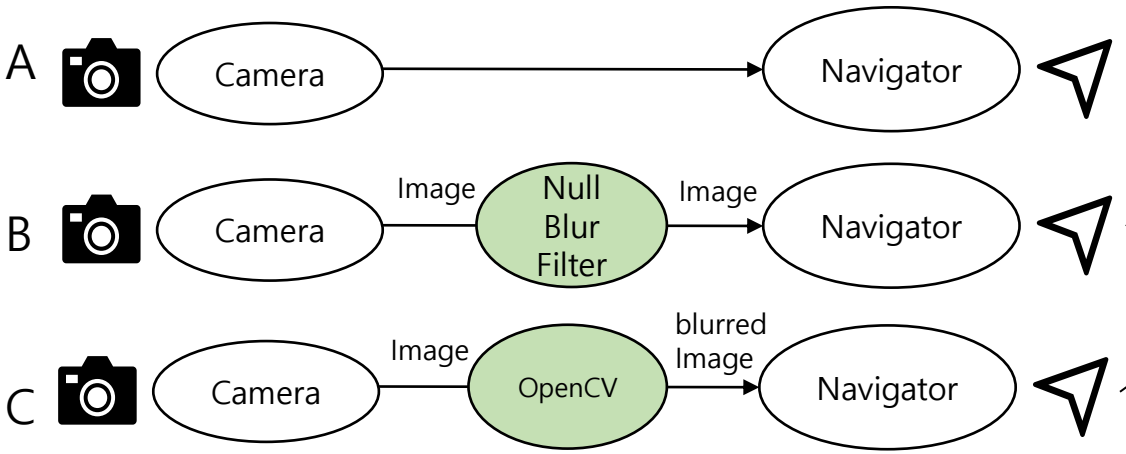
Workload	Latency (ms)	CPU (%)	Power (mW)
Baseline			
`Array1m`	16.255	6.728	2435.133
`PointCloud1m`	16.160	6.612	2441.062
`Struct32k`	6.494	2.526	2225.375
`NavSat`	1.543	1.381	2349.353
`Range`	1.433	1.378	2268.059
Privaros			
`Array1m`	17.225 (+5.9%)	7.050 (+4.8%)	2508.222 (+3.0%)
`PointCloud1m`	17.386 (+7.6%)	7.141 (+8.0%)	2437.294 (-0.2%)
`Struct32k`	7.109 (+9.5%)	2.665 (+5.5%)	2500.412 (+12.4%)
`NavSat`	1.922 (+24.6%)	1.506 (+9.1%)	2389.167 (+1.7%)
`Range`	1.928 (+34.5%)	1.501 (+8.9%)	2367.412 (+4.4%)

Microbenchmark performance

Performance

Workload	Baseline Latency (μs)	Privaros Latency (μs)
Pipe	15.471	15.640 (+1.093%)
UNIX domain sockets (TCP)	20.015	23.188 (+15.9%)
UDP (localhost)	35.039	35.374 (+1.0%)
TCP (localhost)	38.473	38.764 (+0.8%)
UDP (RPC)	51.549	52.335 (+1.5%)
TCP (RPC)	49.457	49.977 (+1.1%)

Experiments using lmbench



Scenario	Latency (ms)	Power (mW)
No redirection	8.124	4749.400
`BlurFilter/Null`	17.509 (+115.5%)	4836.200 (+1.8%)
`BlurFilter/OpenCV`	21.511 (+164.8%)	5132.400 (+8.1%)

Performance impact of flow redirection

Conclusion

- Privaros robustly enforce host-specified privacy policies
- There are low overheads on latency and power consumption
- Privaros is integrated with regulatory platforms(e.g., Digital Sky)

Critique

- Privaros focuses on only delivery drones
- Statically specified policy is complicating
- Difference between URAN?
  - Privacy preserving technique
  - UAV vs UAM+Passenger

Thank you for listening