Secure Drone Identification with Hyperledger Iroha

Andrei Gurtov

Yousef Hashem

Elmedin Zildzic

LINKÖPING UNIVERSITY

Introduction

Drone Remote Identification Protocol (DRIP)

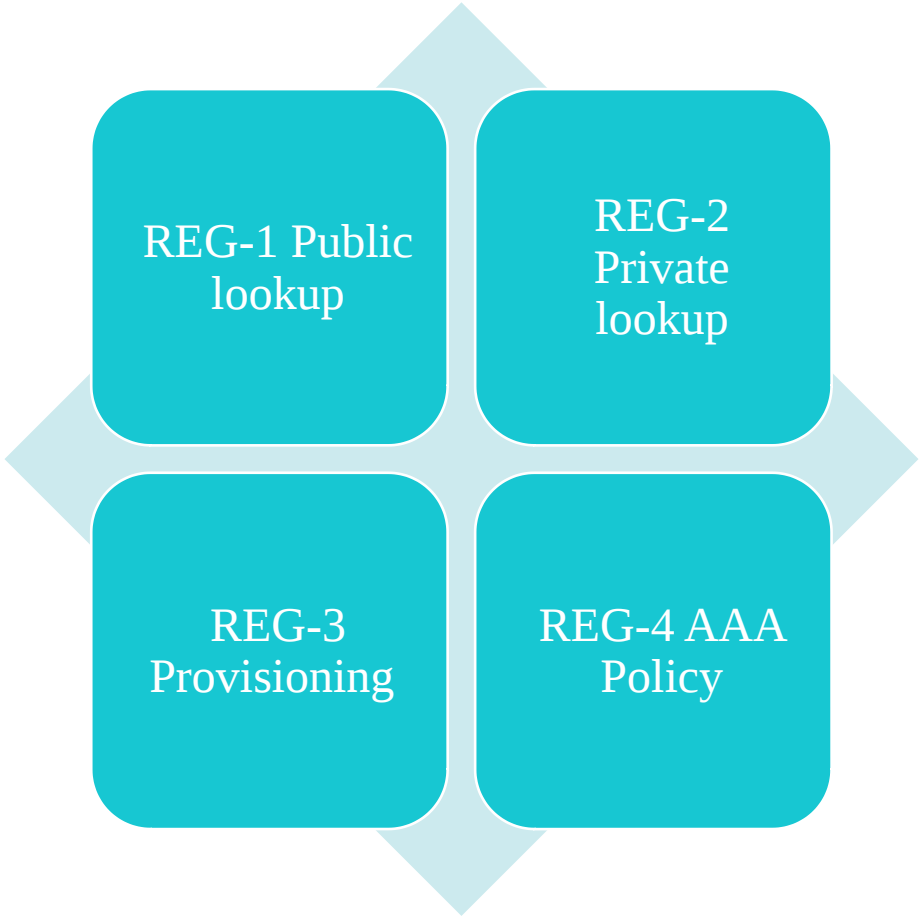
- Newly proposed protocol to incorporate authentication and trust mechanisms into drone communications
- DRIP Requirements submitted to IESG for publication
- DRIP RID & authentication standards proposed (IETF WG)

Yet to be determined DRIP solutions

Registry – Direct RID lookup

Network RID – Location updates

DRIP Registry Requirements



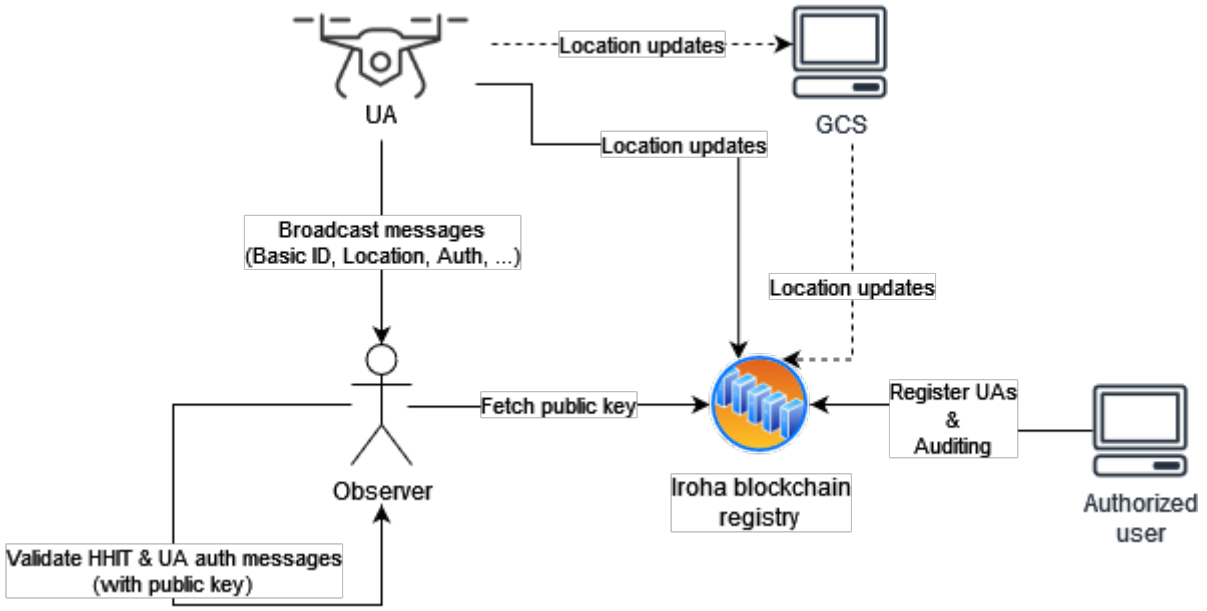
Main contributions

- Novel drone ID architecture based on Hyperledger Iroha and DRIP
- Informal security analysis of the proposed architecture
- Performance evaluation

Architecture

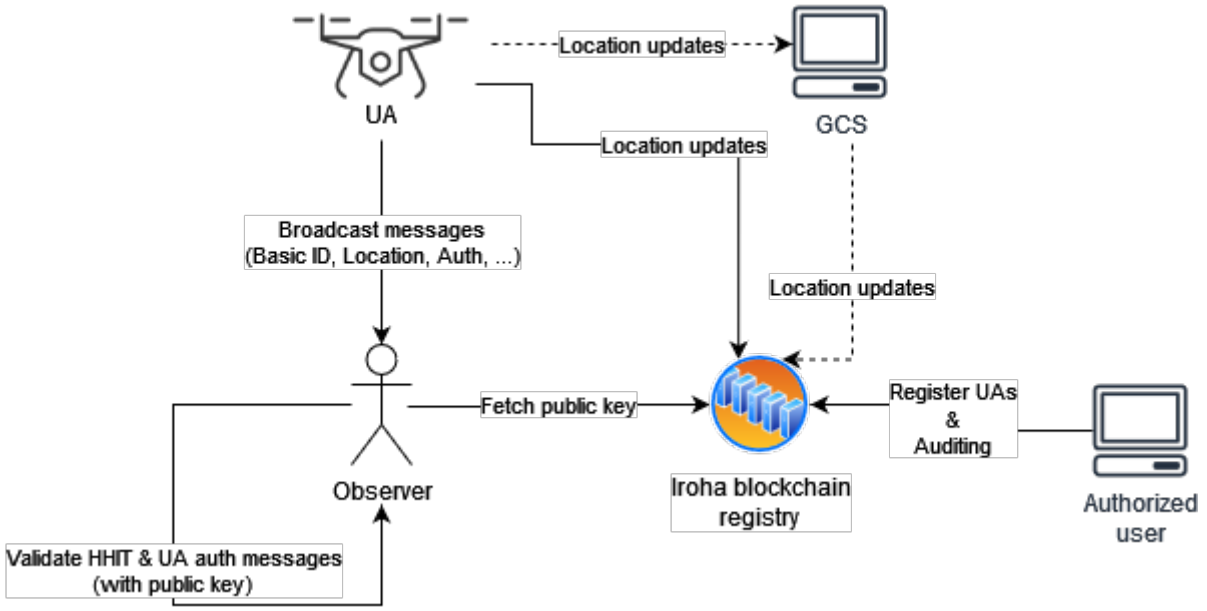
DRIP Registry interaction

- Observers receive Direct RID messages, and performs lookups on registry
- UAs and GCSs send location updates to registry
- Admin registers new accounts (drone/operators)
- UAs do not participate in the blockchain



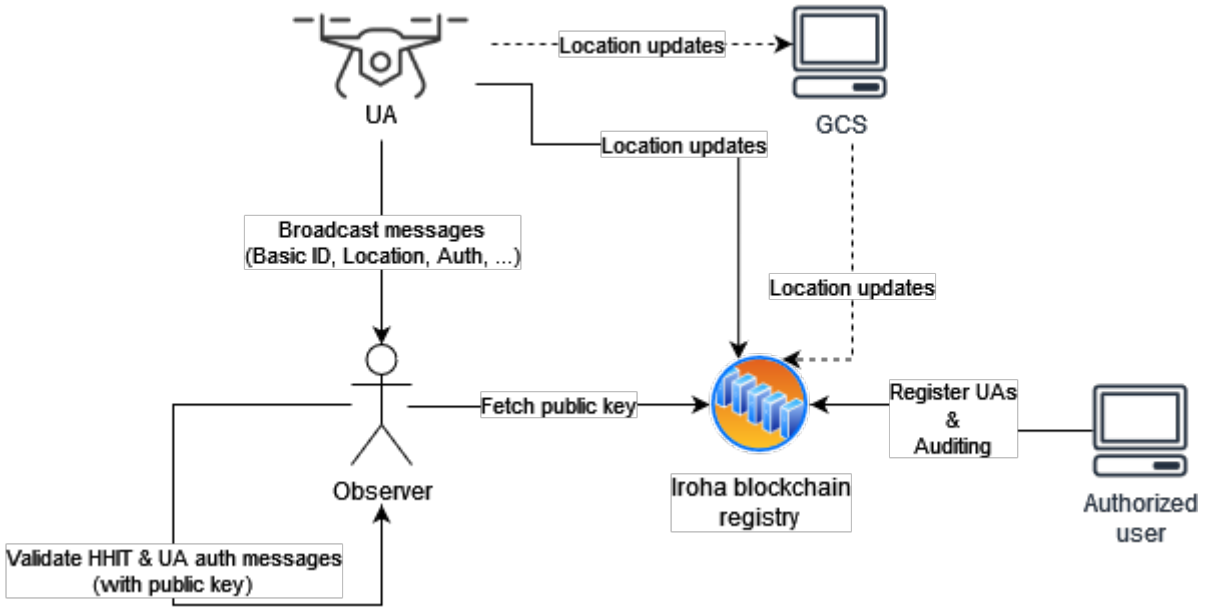
DRIP Registry Implementation

- OpenHIP hipv2_new_crypto branch
  - •Modified (by us) to support DRIP HIT Suite (EDDSA/cSHAKE)
  - • Include Hierarcial ID (HID) in HIT = HHIT
  - •Generate HHITs
- Our own scripts to generate DRIP public keys and certificates
  - •draft-ietf-drip-reqs-17
  - •draft-ietf-drip-rid-08



DRIP Registry Implementation

- DRIP Bluetooth Advertisements
  - ASTM F3411-19 and DRIP format
  - Works with OpenDroneID
- Web app to track drones via blockchain registry (Network RID)
- Android app to track and verify HHITs and auth messages (Registry lookup)



Hyperledger Iroha

Private permission-based blockchain framework	
Byzantine fault-tolerant consensus algorithm (YAC)	
Focus on performance and scalability	• Proposal (block) size should not affect consensus time
Configurable with other Hyperledger projects	• Hyperledger Burrow to support smart contracts
Supports domains	• Can be used to separate private and public lookups (REG-1 & -2)

Hyperledger Iroha Accounts

- Accounts can be created on multiple domains to separate PII from public data
    - One private domain
    - One public domain
- Any data set on the accounts is non-repudiable, however, values can still be modified
- One account cannot overwrite data set by another account
    - Any account that has set some data will be visible

```
[Account]:
-Account Id:- drone1@domain1
-Domain- domain1
-Roles-:
user
-Data-: {
    "admin@domain1": {"status": "grounded"},
    "drone1@domain1": {"status": "airborne"}
}
```

Security analysis

Hyperledger Iroha

Private-based

Prevent unauthorized access.

Impossible to inject (sybil) nodes without account with permissions to add them.

Permission-based

Transactions and queries require the correct permissions.

Byzantine Fault-tolerant

Uses a Byzantine Fault-tolerant algorithm - can tolerate up to n faulty nodes out of 3n+1.

Faulty nodes can be replaced.

Hyperledger Iroha

- Supports multisignature transactions
- Supports smart contracts w/ Hyperledger Burrow

Multisignature transactions

- Requiring multiple signatures for a single transaction can prevent single point-of-failures, such as when admin accounts are compromised.
- Make false data dissemination attacks harder – e.g. need to compromise both GCS and drone.

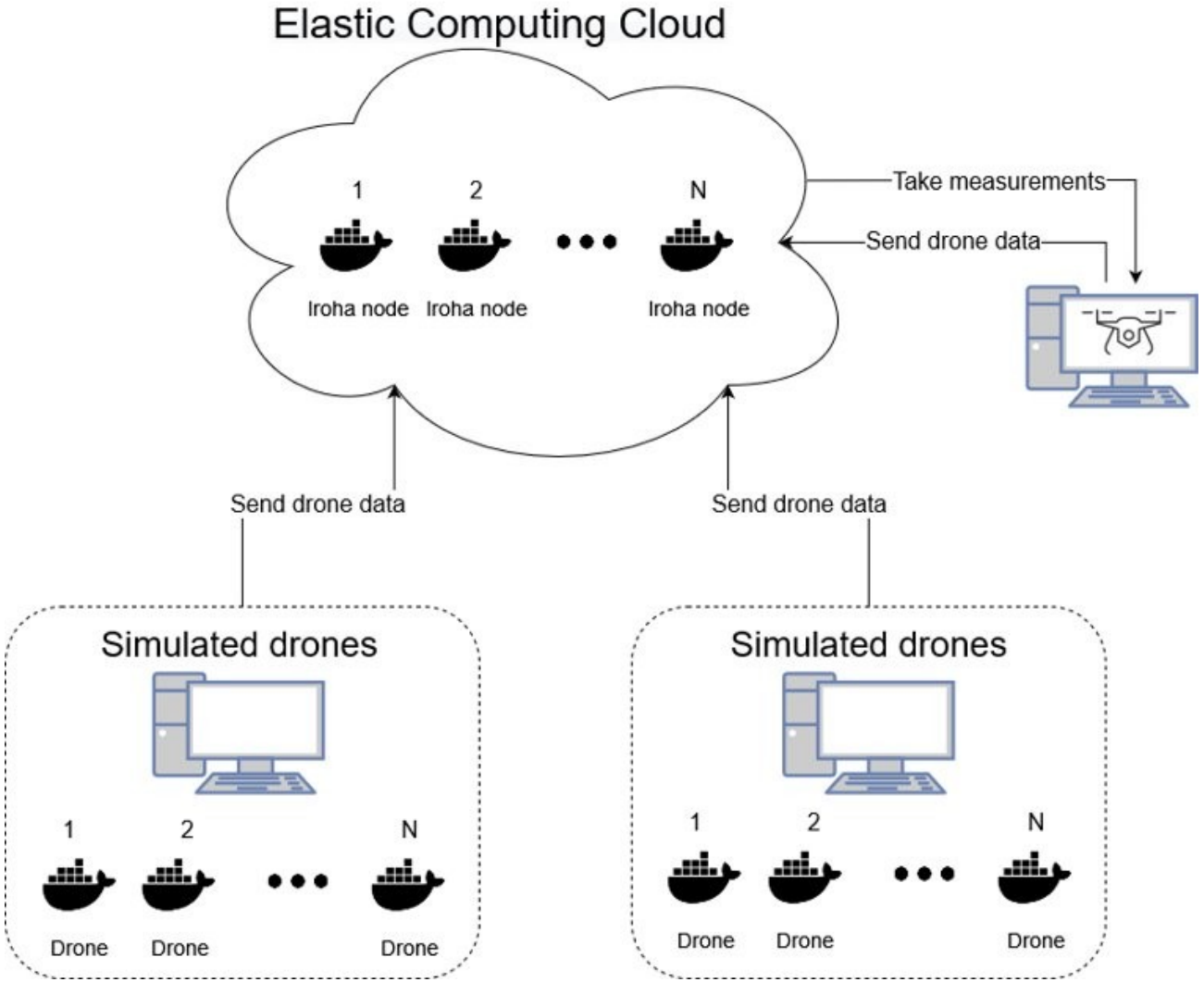
Smart contracts

- Make queries behave like transactions
  - Queries are stored on the blockchain
  - Auditability – Who has requested what information (REG-4)
  - Traceability

Performance evaluation

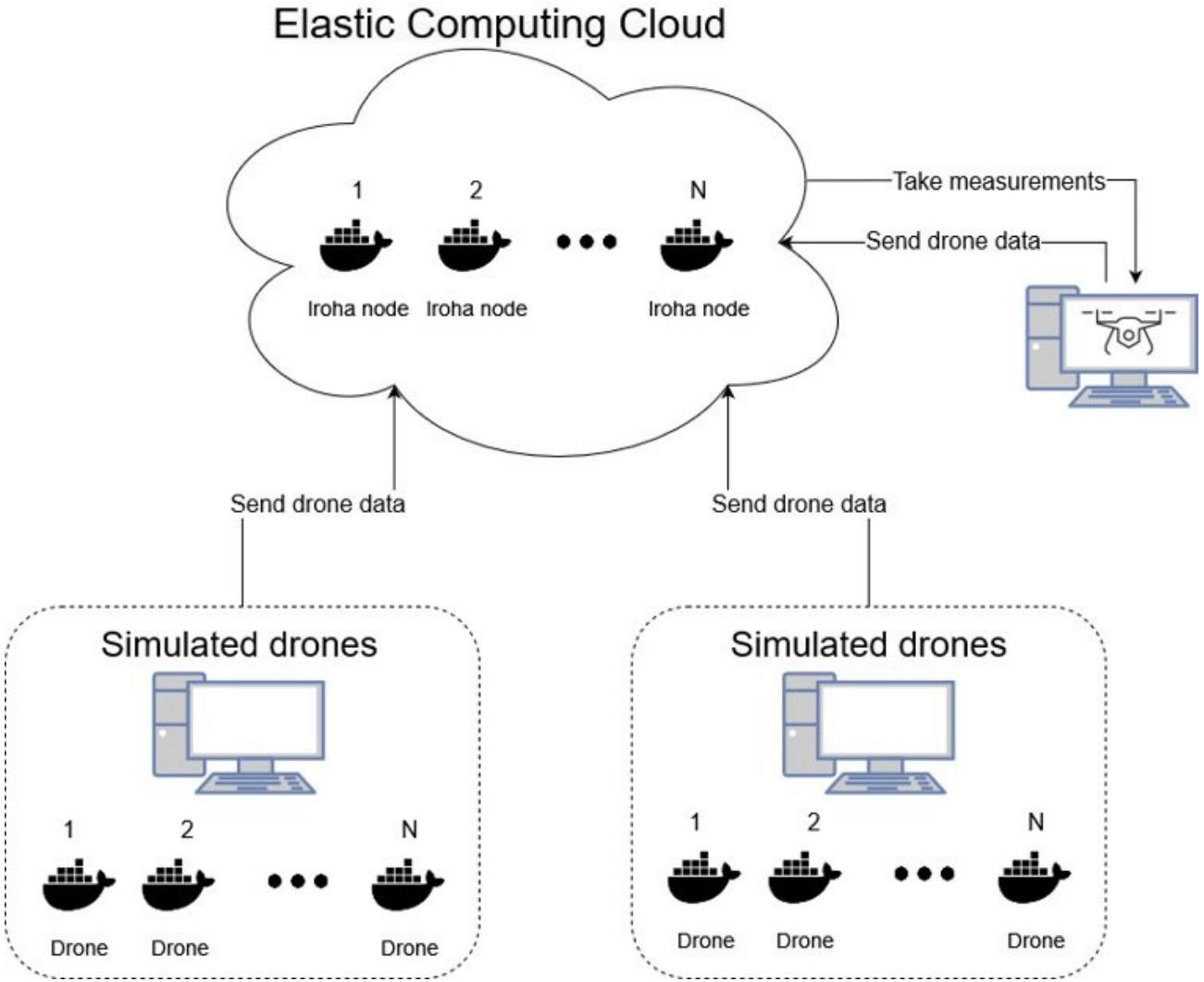
Methodology

- 16 and 30 Amazon EC2 instances used as Iroha nodes
  - Launched with Docker Swarm
- Per instance (free):
  - 2 vCPU Intel Scalable Processor @ 2.5 GHz with 6 CPU credits/hour
  - 1 GB RAM
  - 8GB EBS Storage
  - Up to 5 Gbps network speed
  - Region: eu-north-1



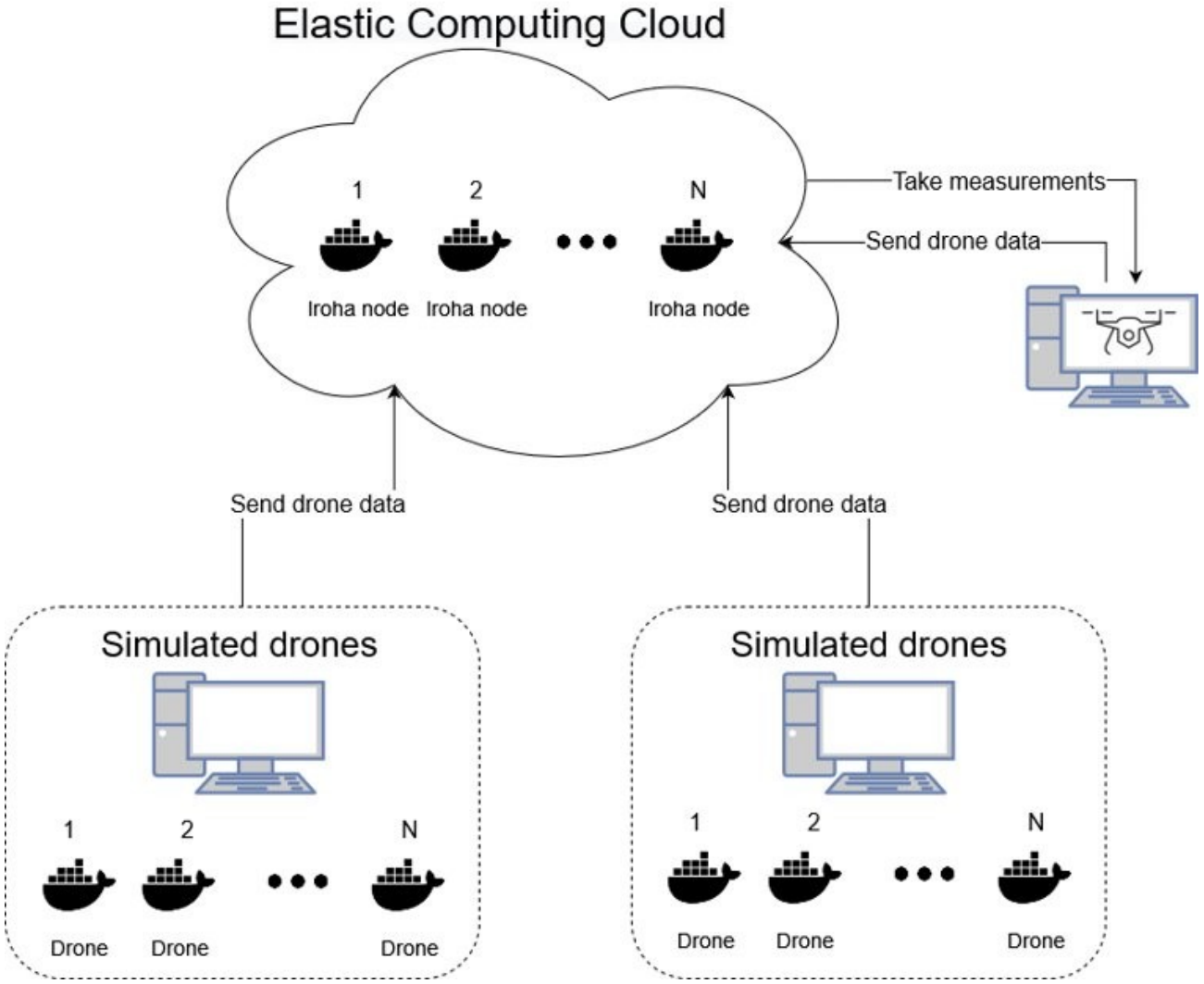
Methodology

- 100 and 200 simulated drones launched, sending a location update transaction each second
  - Simulated drones launched as workers with Locust
  - Workers randomly selected a node to send transactions to (load balancing)
- Workers launched on home networks, sending transactions to Iroha nodes through Internet
  - 10-15 ms RTT (ping)



Methodology

- Location updates used non-standard format
  - Latitude
  - Longitude
  - Altitude
  - Direction
  - Speed
  - Timestamp
  - Status
- 123 bytes payload
- Can be shortened by using standardized format (e.g. ASTM F3411-19, ~24 bytes)
  - Minimize blockchain storage requirements



Methodology

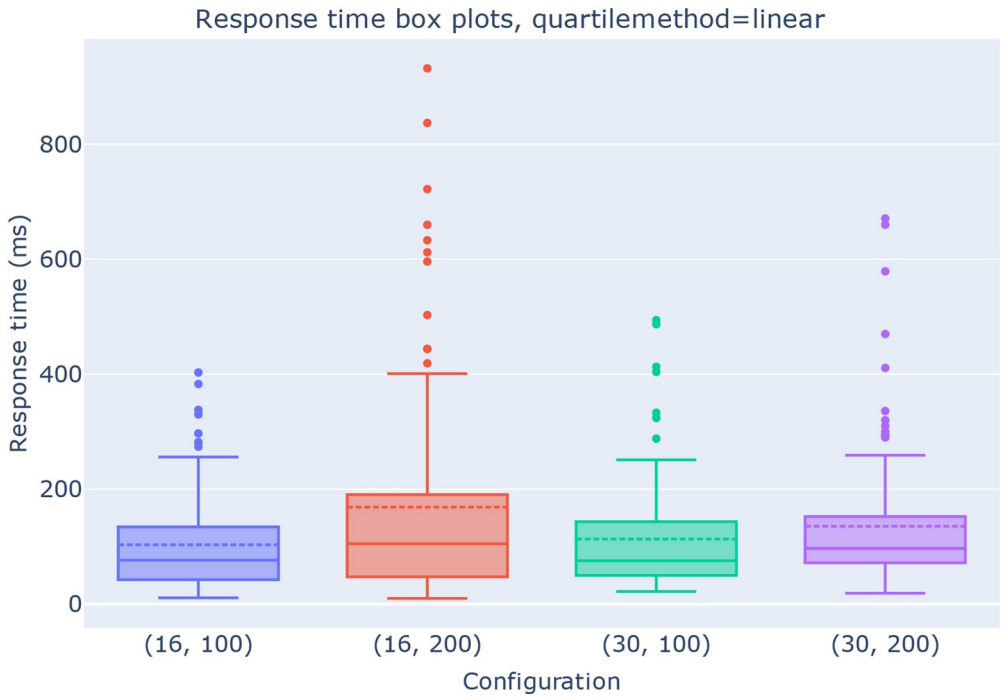
- Iroha performance parameters configurable
- Lower delays can yield better performance, but also introduce higher network loads (more vote messages exchanged between peers)

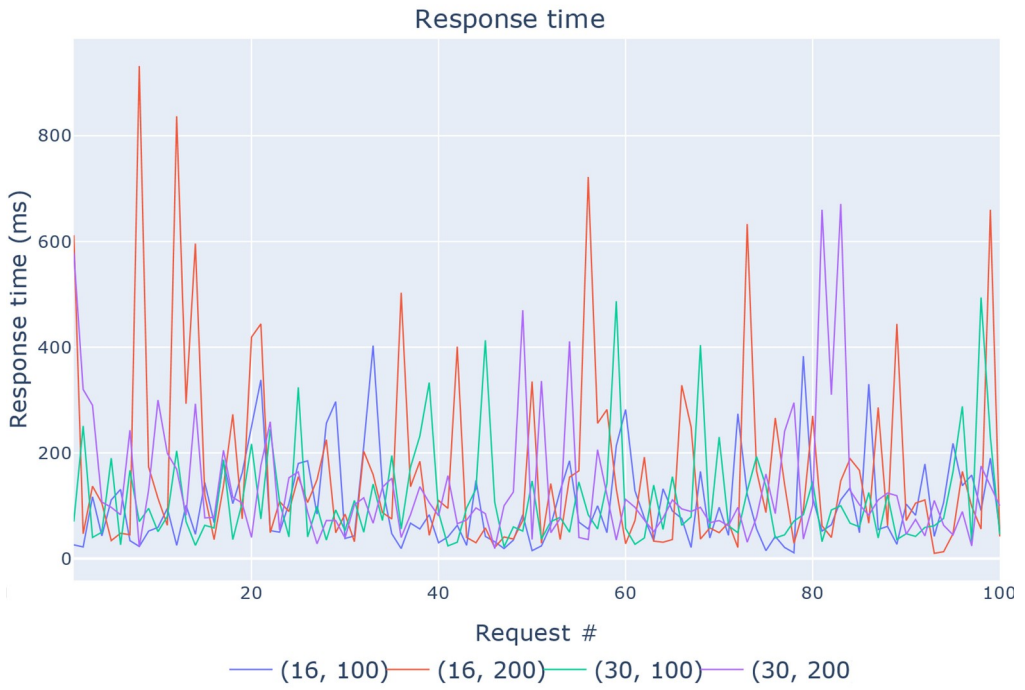
```
"max_proposal_size" : 1000,
"proposal_delay" : 100,
"vote_delay" : 200,
"mst_enable" : False,
"mst_expiration_time" : 200,
"max_rounds_delay": 100,
"stale_stream_max_rounds": 1000
```

Transaction response times

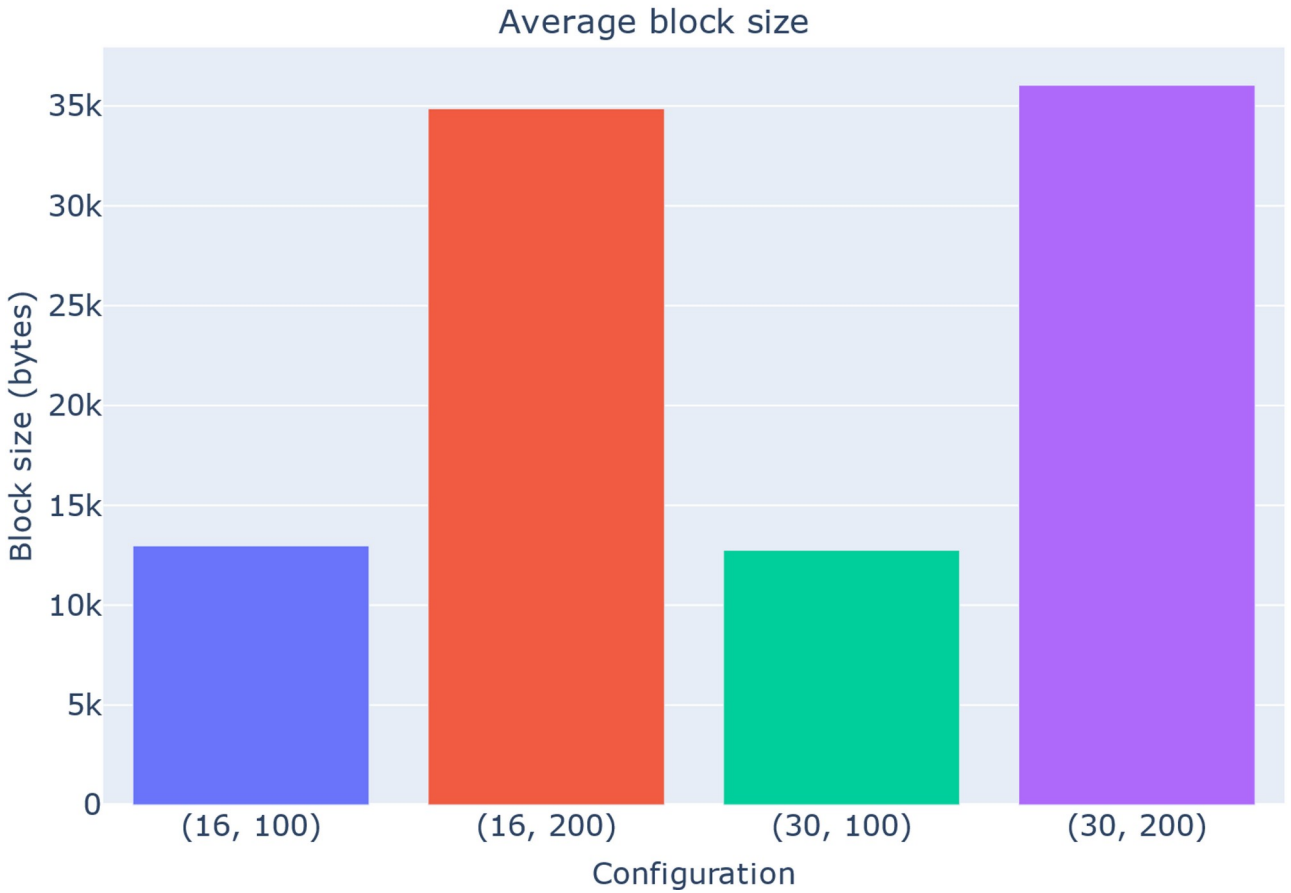
(=one-way trip time + block consensus delay)

	(16, 100)	(16, 200)	(30, 100)	(30, 200)
Min	11	10	22	19
Q1	42.5	47.5	50	72
Median	76.5	105	75.5	97
Mean	103.35	168.65	113.17	135.41
Q3	134.5	190.5	143.5	152.5
Upper fence	256	401	251	259
Max	403	932	494	671





Average block sizes



Conclusion

Conclusion

- Hyperledger Iroha can fulfill all DRIP registry requirements and has decent performance
- Frequent location updates places big requirement on storage
  - Might still be worthwhile to store on blockchain for "black box" purposes.
  - Less frequent location updates to lessen storage requirements