Everywhere All at Once: Co-Location Attacks on Public Cloud FaaS

Zirui Neil Zhao, Adam Morrison, Christopher W. Fletcher, Josep Torrellas

University of Illinois Tel Aviv University

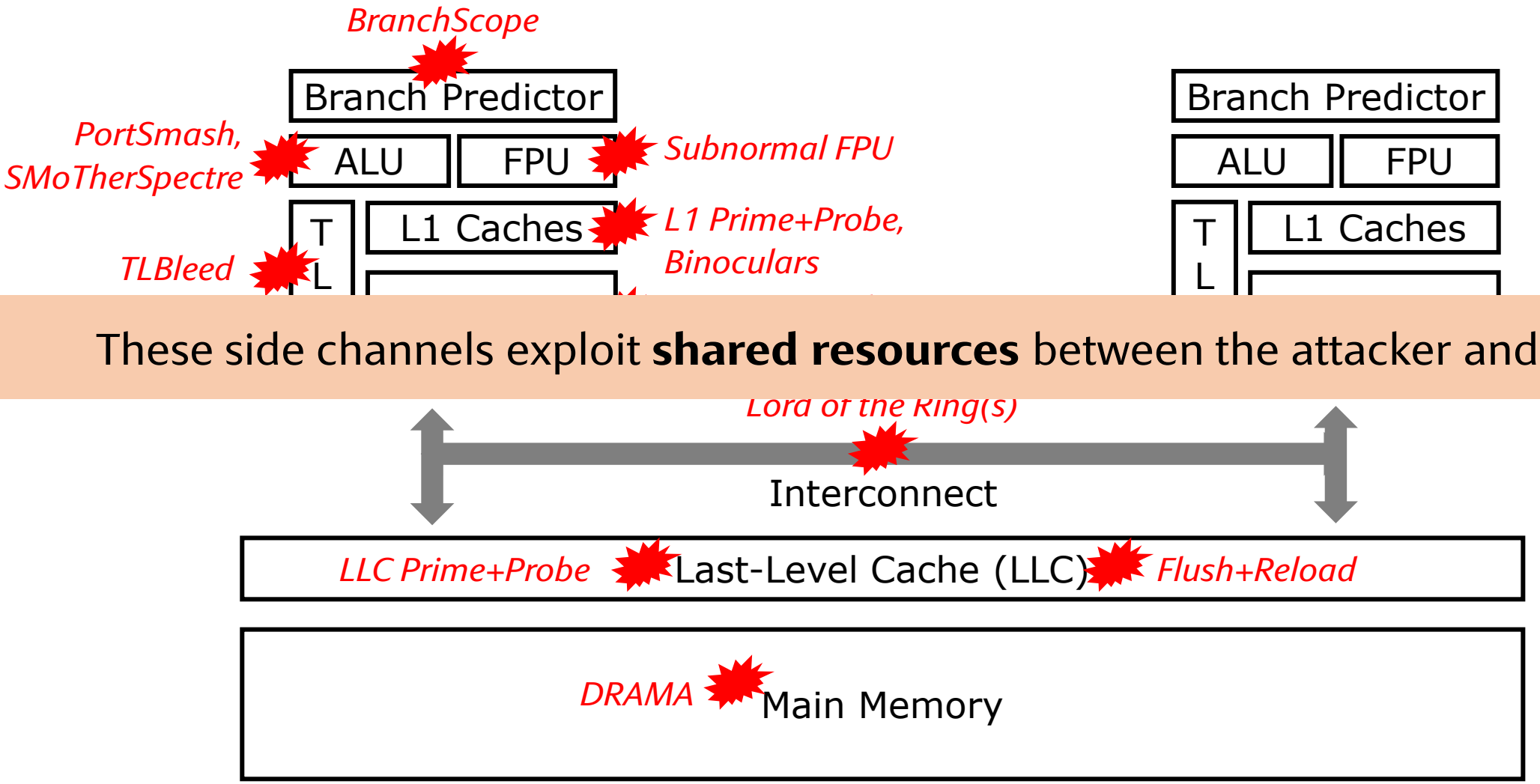
ASPLOS '24 – Session 1D: Attacks and Mitigations

ILLINOIS

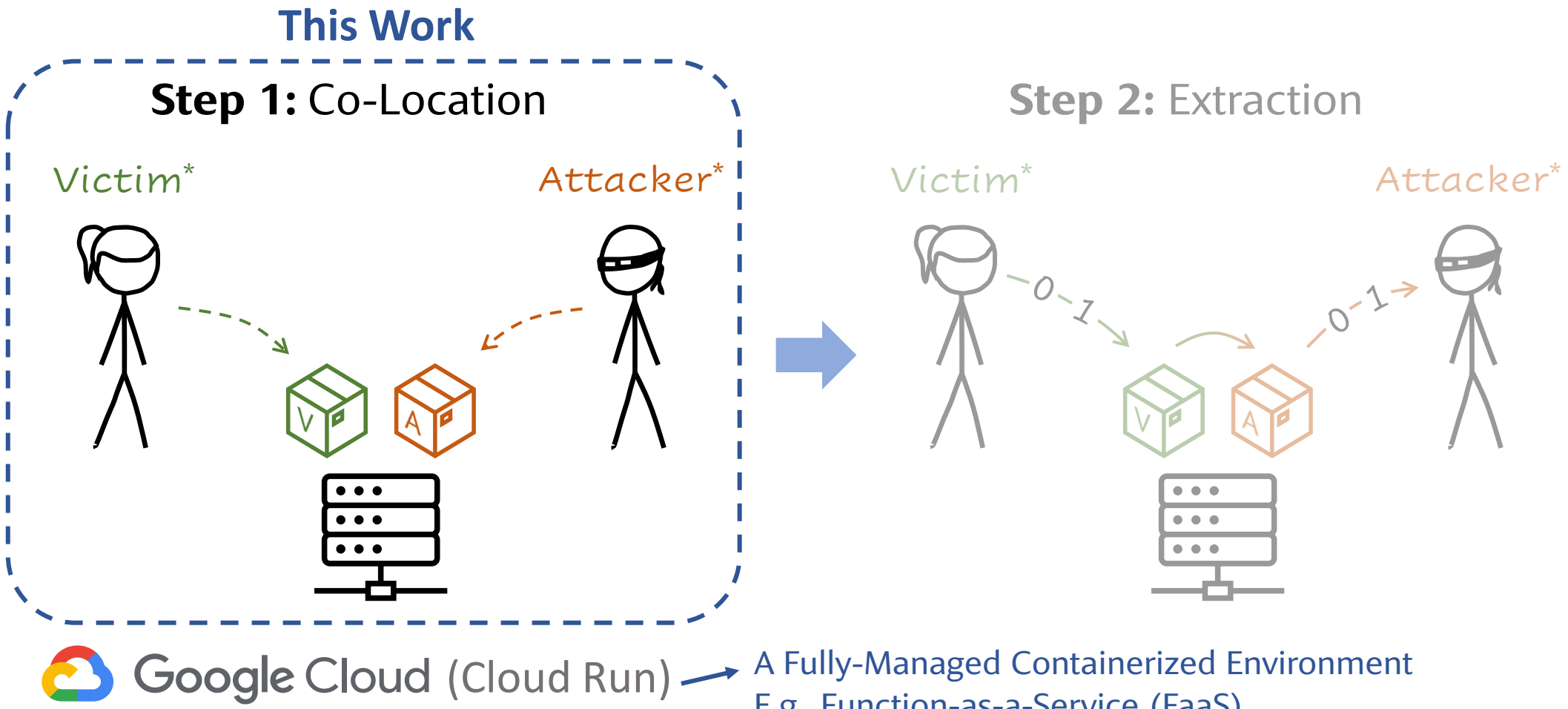
TEL AVIV UNIVERSITY

i-acoma group

Many Microarchitectural Side-Channel Attacks

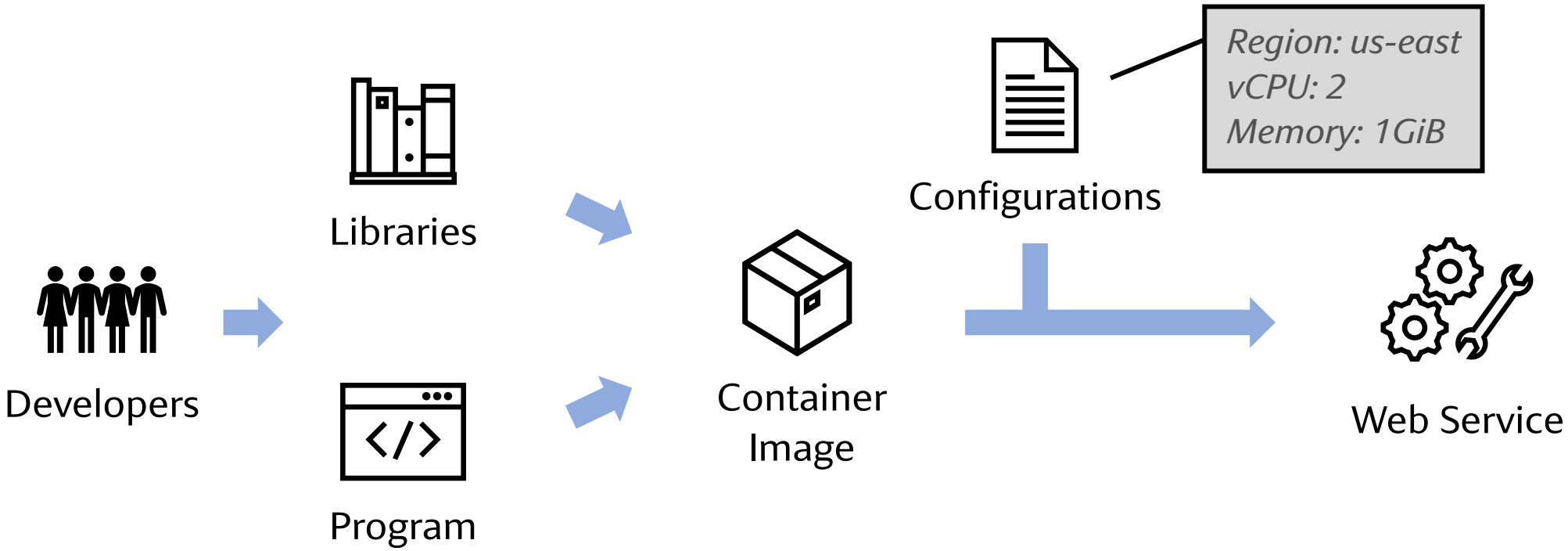


Steps of Side-Channel Attacks in Public Cloud

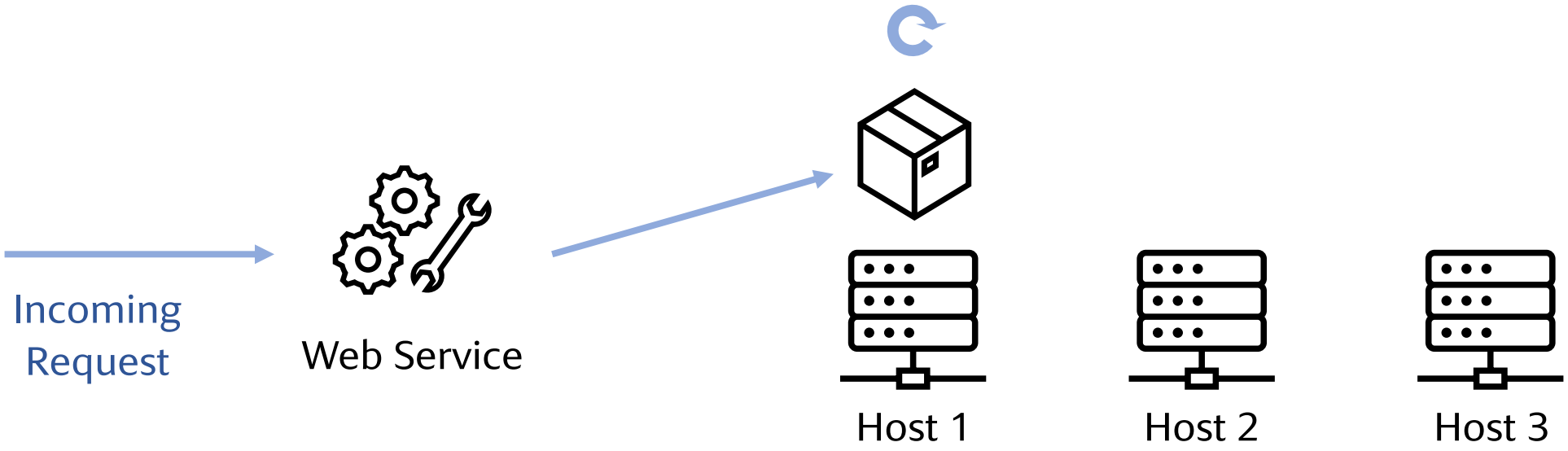


*Characters are based on https://xkcd.com/2176 and https://xkcd.com/1808 (under a CC Attribution-NonCommercial 2.5 License)

Background: Fully-Managed Containerized Environment



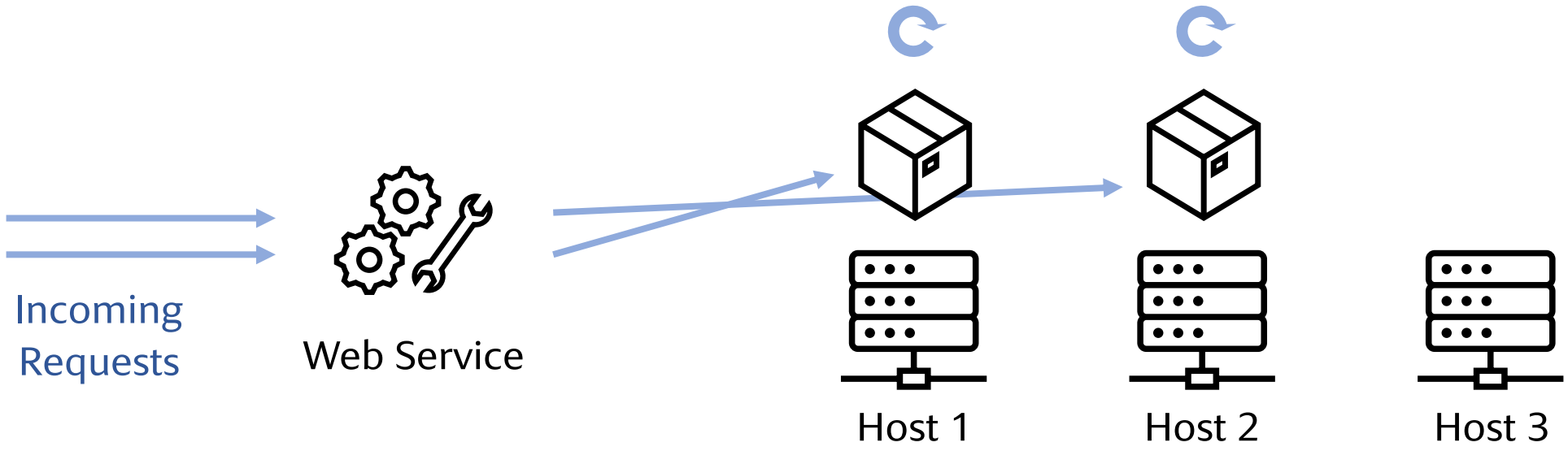
Background: Fully-Managed Containerized Environment



Cloud vendor automatically launches a container instance

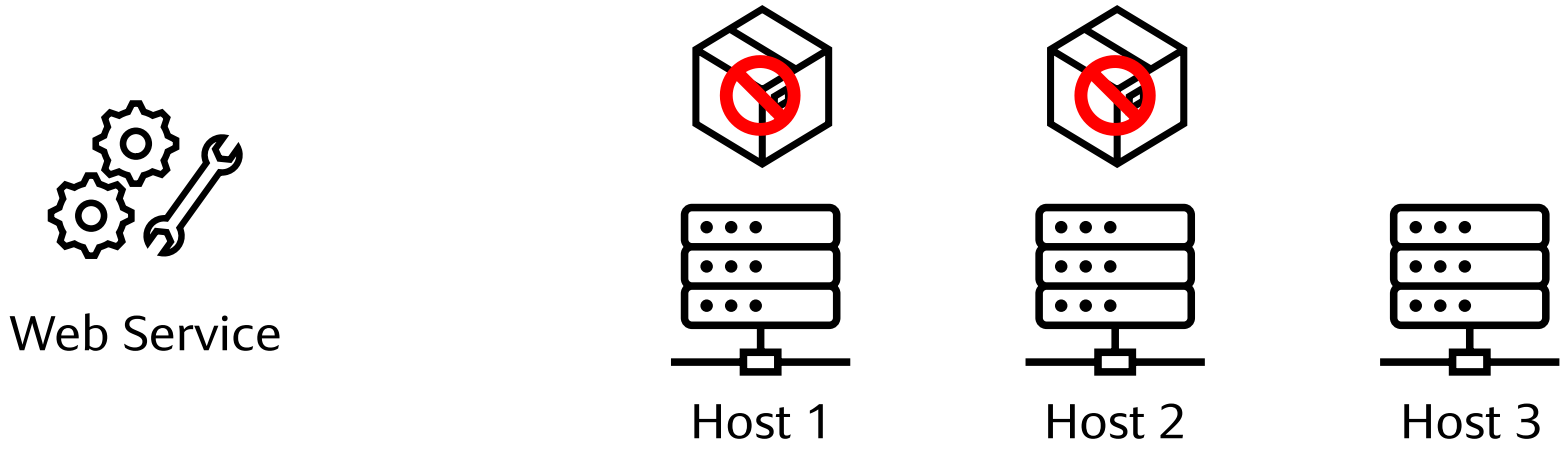
(The instance placement is managed by the vendor)

Background: Fully-Managed Containerized Environment



Cloud vendor launches more instances to handle traffic increases

Background: Fully-Managed Containerized Environment



Takeaways:

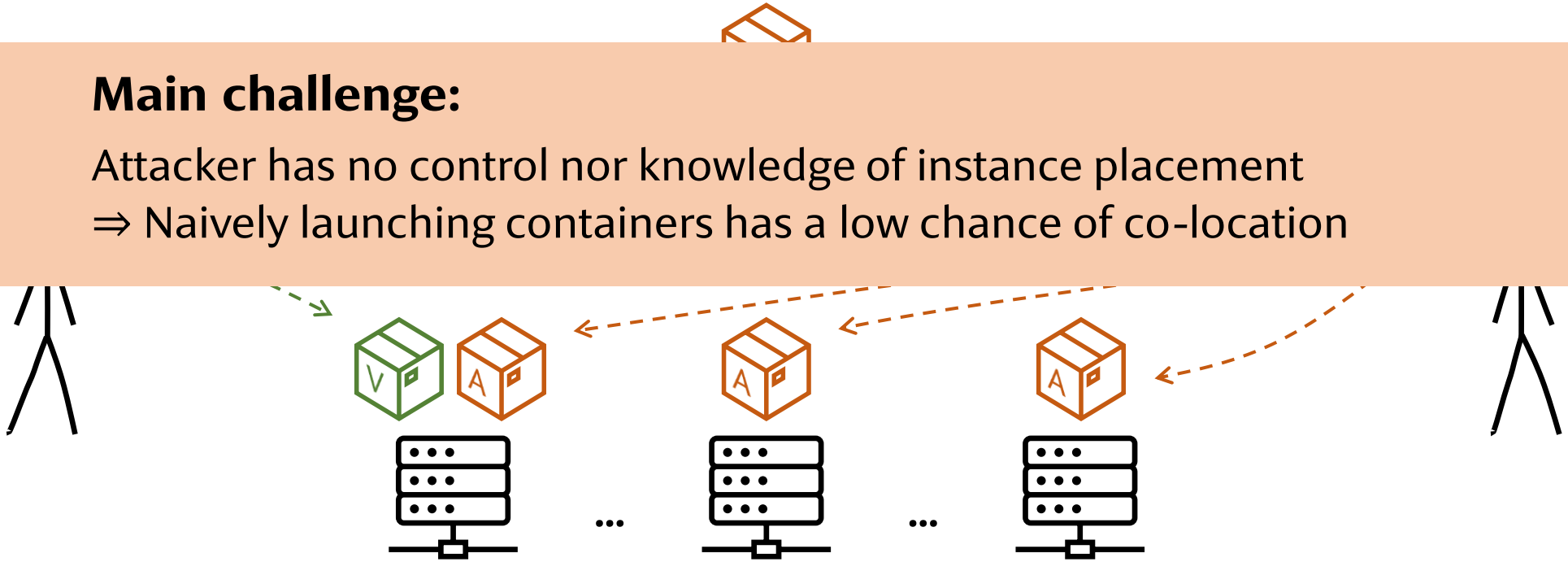
- Container instance placement is fully managed by the cloud vendor
- Container creation and destruction are automatically adapted to service's demand

Challenge of Co-Locating with the Target Victim

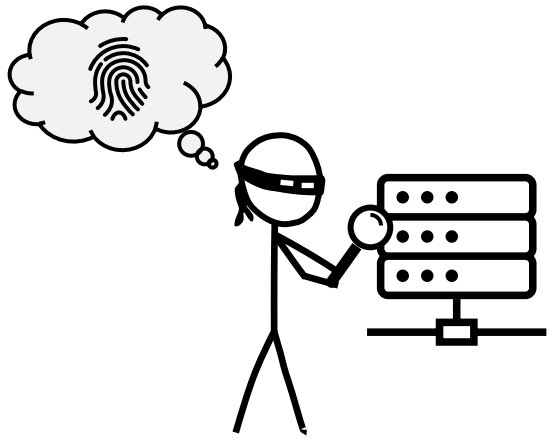
Attacker's goal: Spread attacker containers across many hosts
⇒ Increase the chance of co-location

Main challenge:

Attacker has no control nor knowledge of instance placement
⇒ Naively launching containers has a low chance of co-location

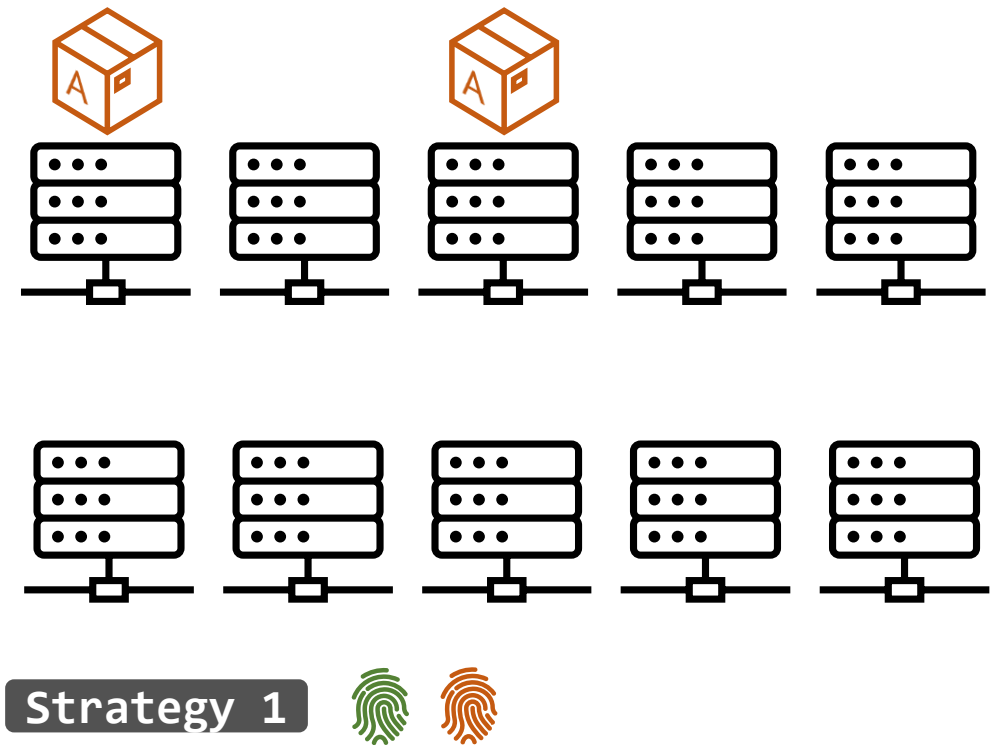


Idea: Fingerprint Host → Reverse Engineer Placement Behavior

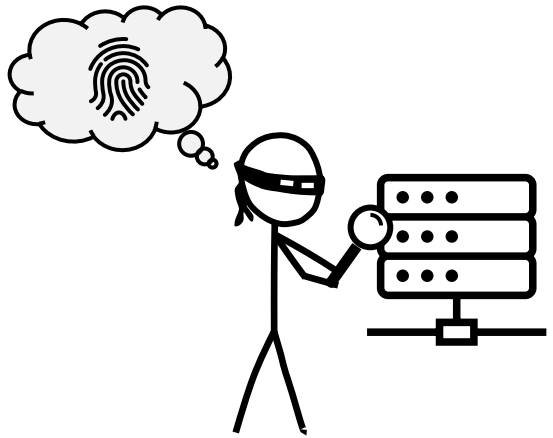


Accurate host fingerprinting

Understand container placement

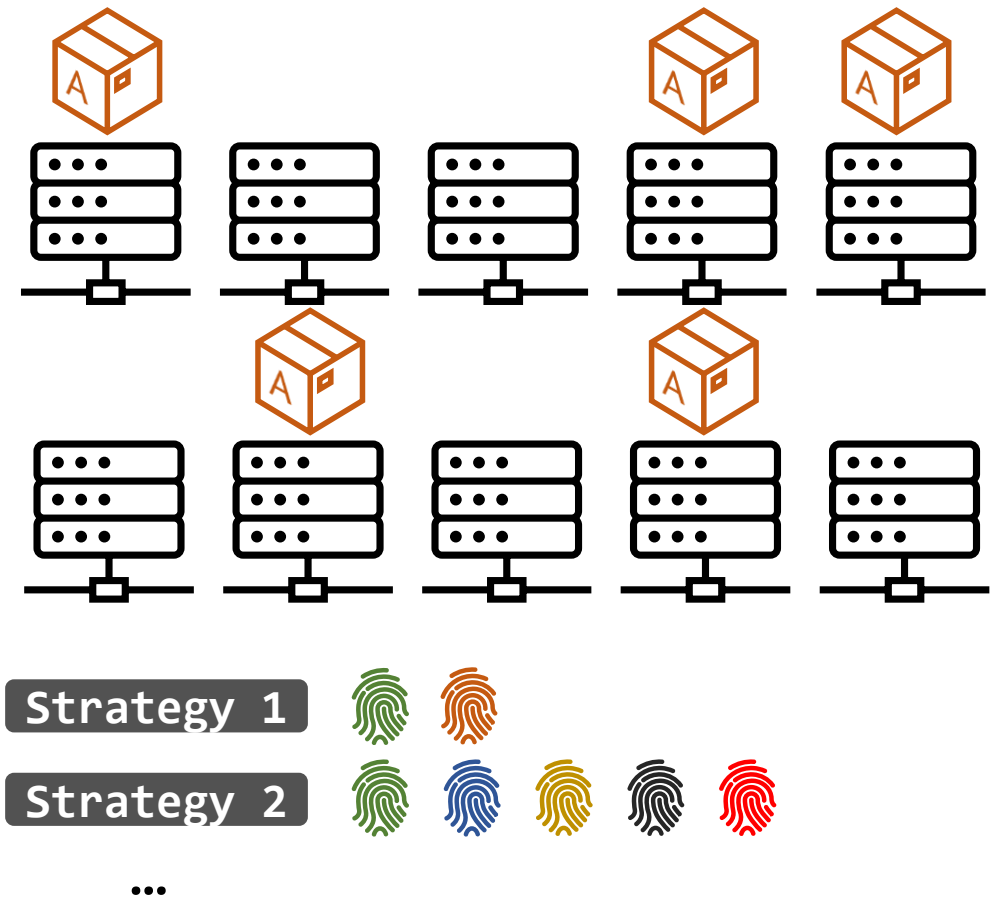


Idea: Fingerprint Host → Reverse Engineer Placement Behavior

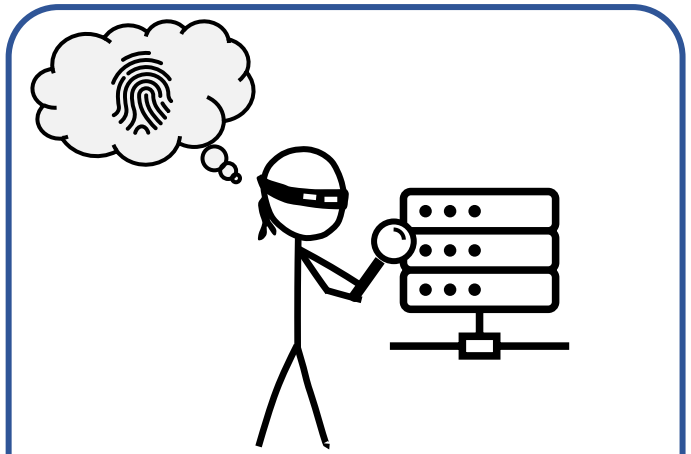


Accurate host fingerprinting

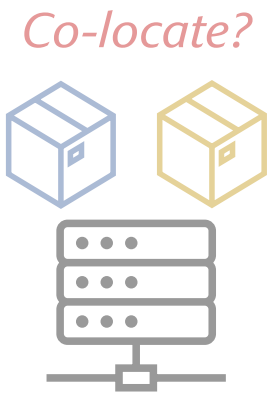
Understand container placement



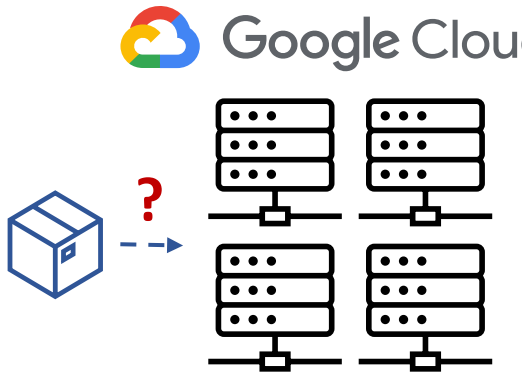
Main Contributions & Results



1. Accurate
Host Fingerprinting



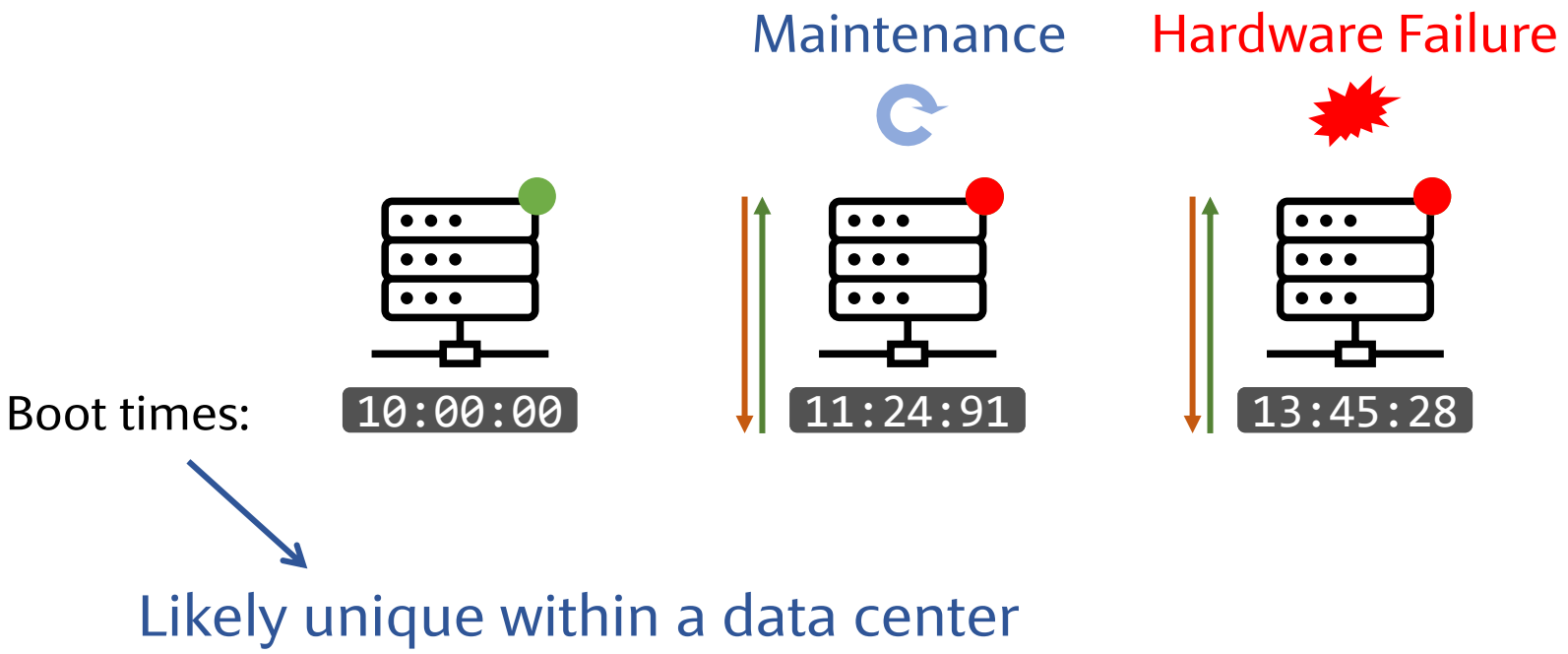
2. Inexpensive
Co-Location Test



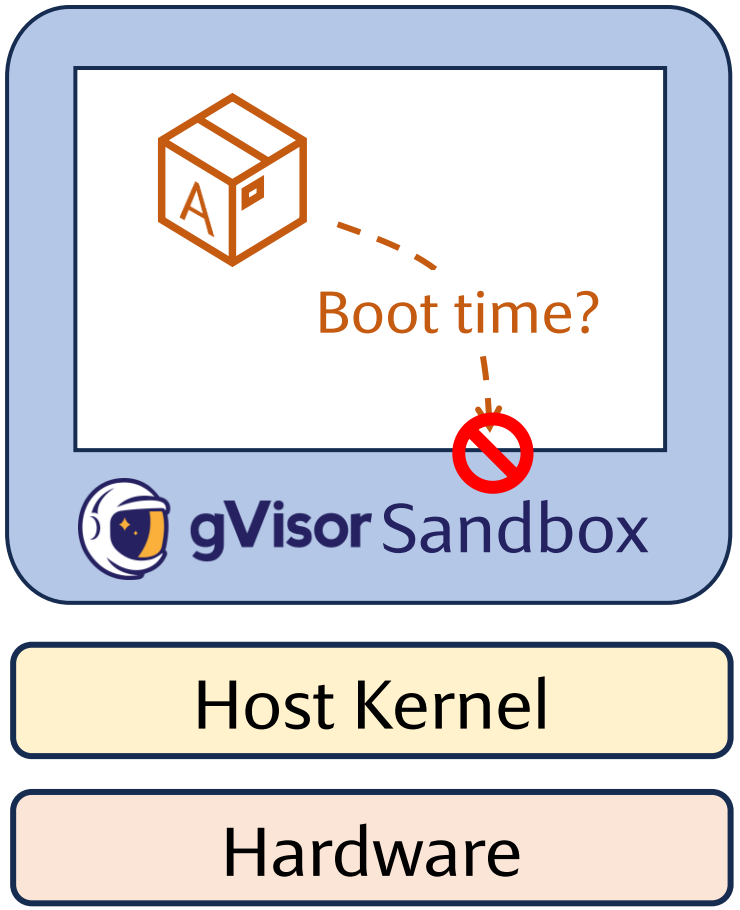
3. Exploitable
Placement Behavior

Exploitable behavior of Google Cloud ⇒ High chance of co-location

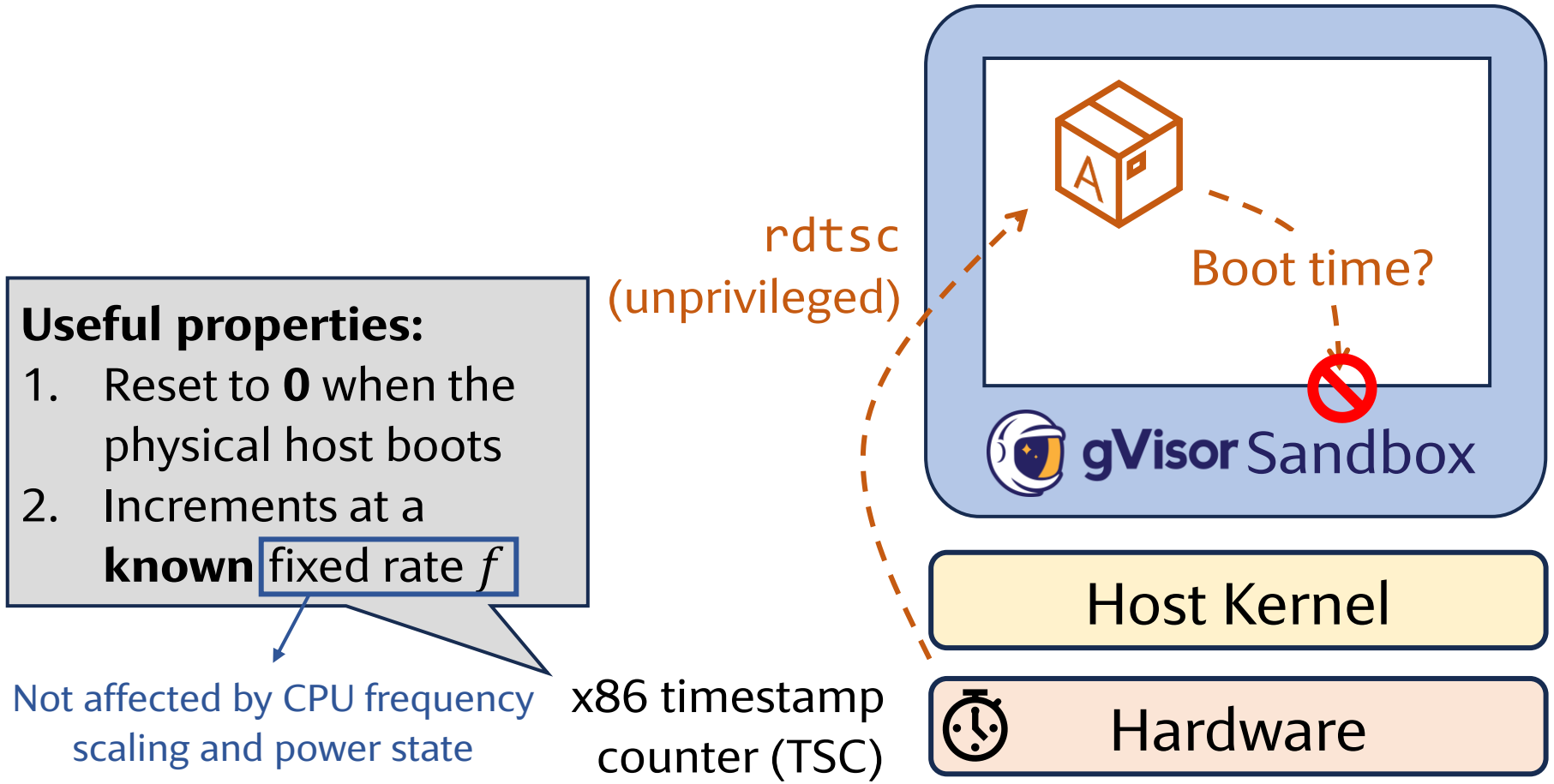
Insight 1: Physical Host's Boot Time as Fingerprint



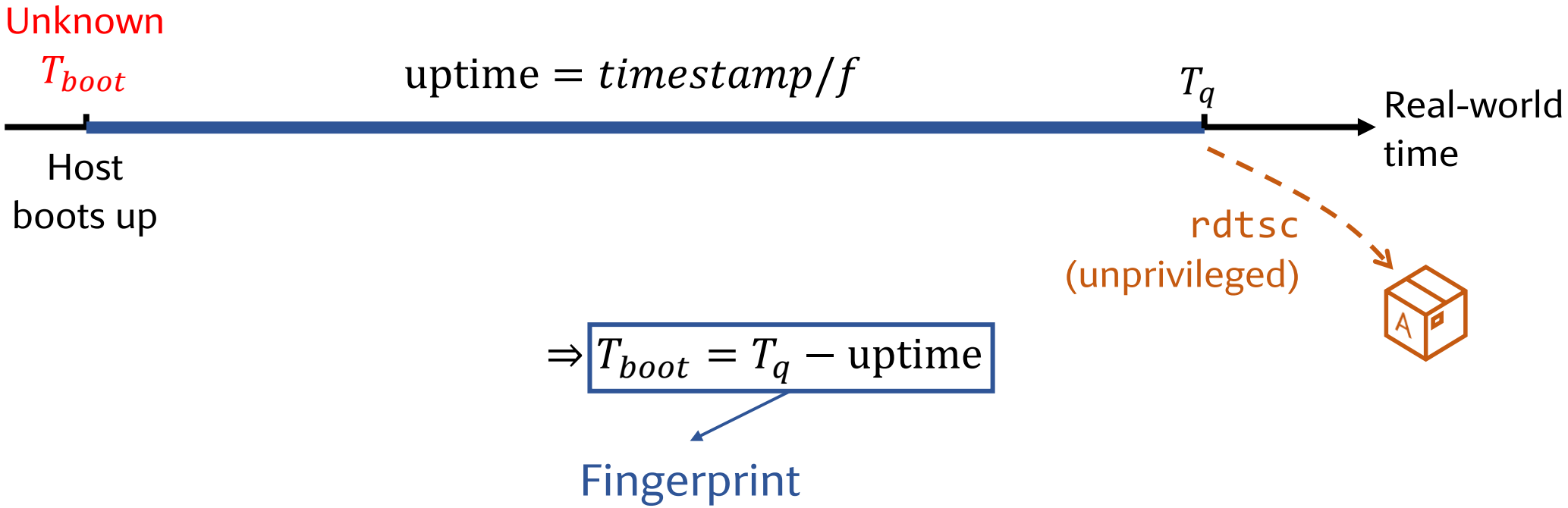
Challenge: Host Information is Hidden Due to Sandboxing



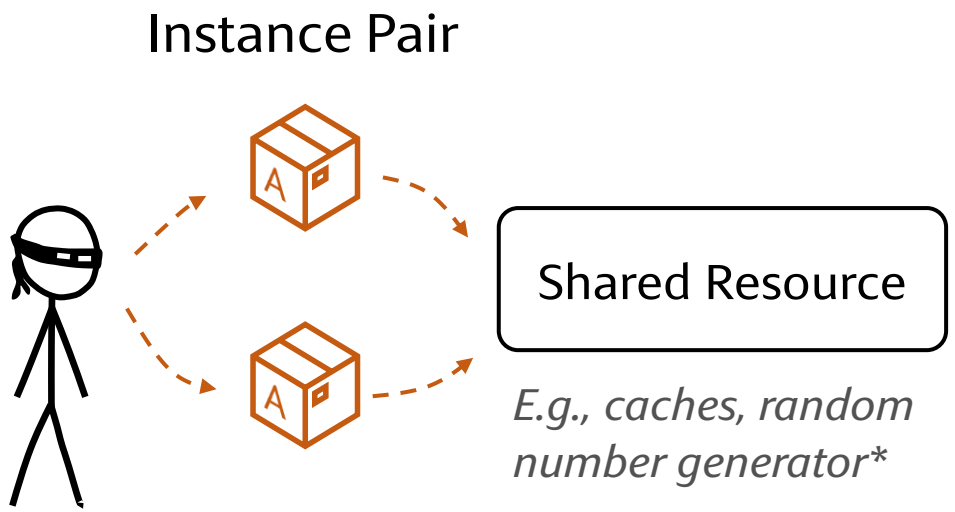
Insight 2: Bypassing Software Protection by Asking the Hardware

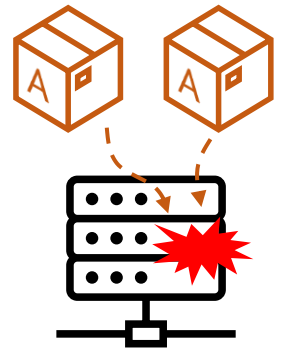


Derive Boot Time From Timestamp Counter

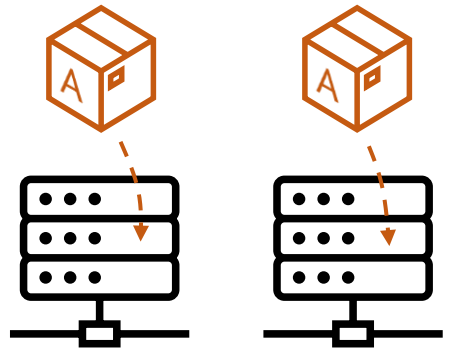


Verifying Co-Location





Contention!
⇒ Co-Located



No Contention
⇒ Not Co-Located

Scalability issue: it requires $O(N^2)$ pairwise tests to verify N containers

The paper discusses a scalable, fingerprint-assisted method for verifying co-location

* Evtyushkin et al., Covert Channels through Random Number Generator: Mechanisms, Capacity Estimation and Mitigations (CCS '16)

Host Fingerprints are Highly Accurate

For each pair of container instances

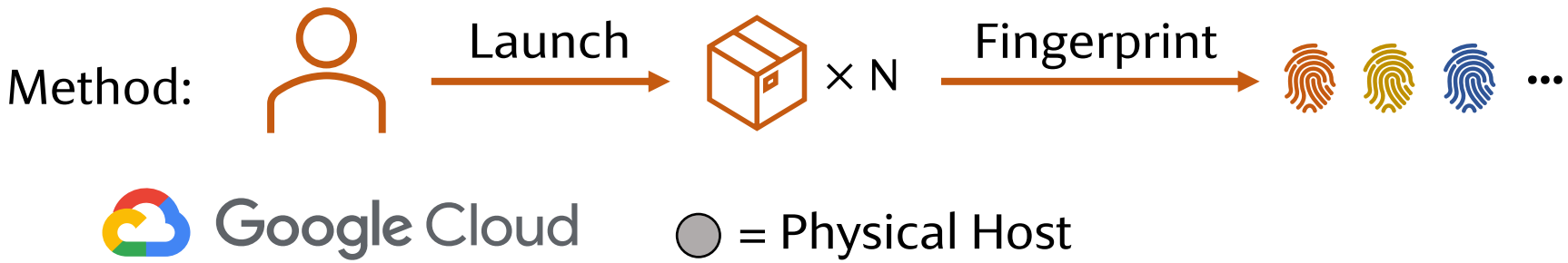
- **False positive (FP):** same fingerprints but not co-located
- **False negative (FN):** different fingerprints but co-located

- Measure accuracy in three data center regions (us-central1/east1/west1)
- Repeat measurements five times in each data center region

Average FN rate: 0.00%
Average FP rate: 0.02%

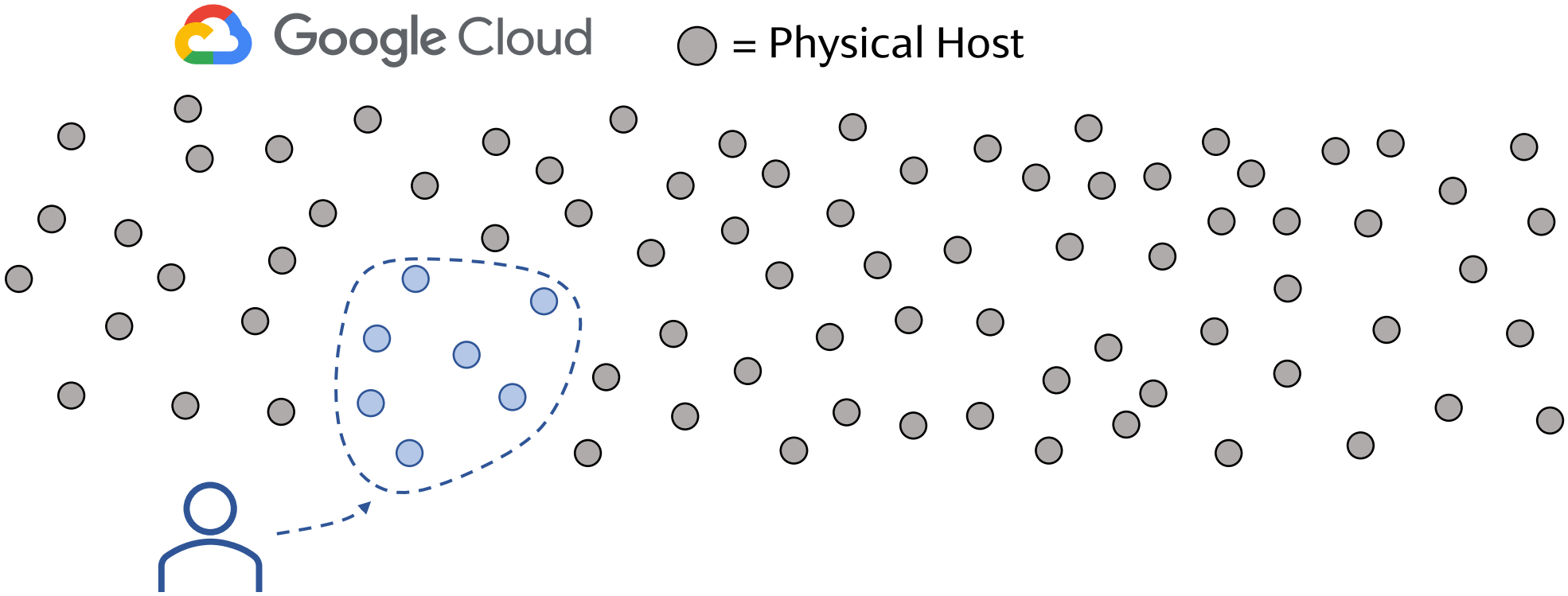
☺ 14 out of 15 measurements generate perfect fingerprints (no FP nor FN)

Understanding Instance Placement Policy



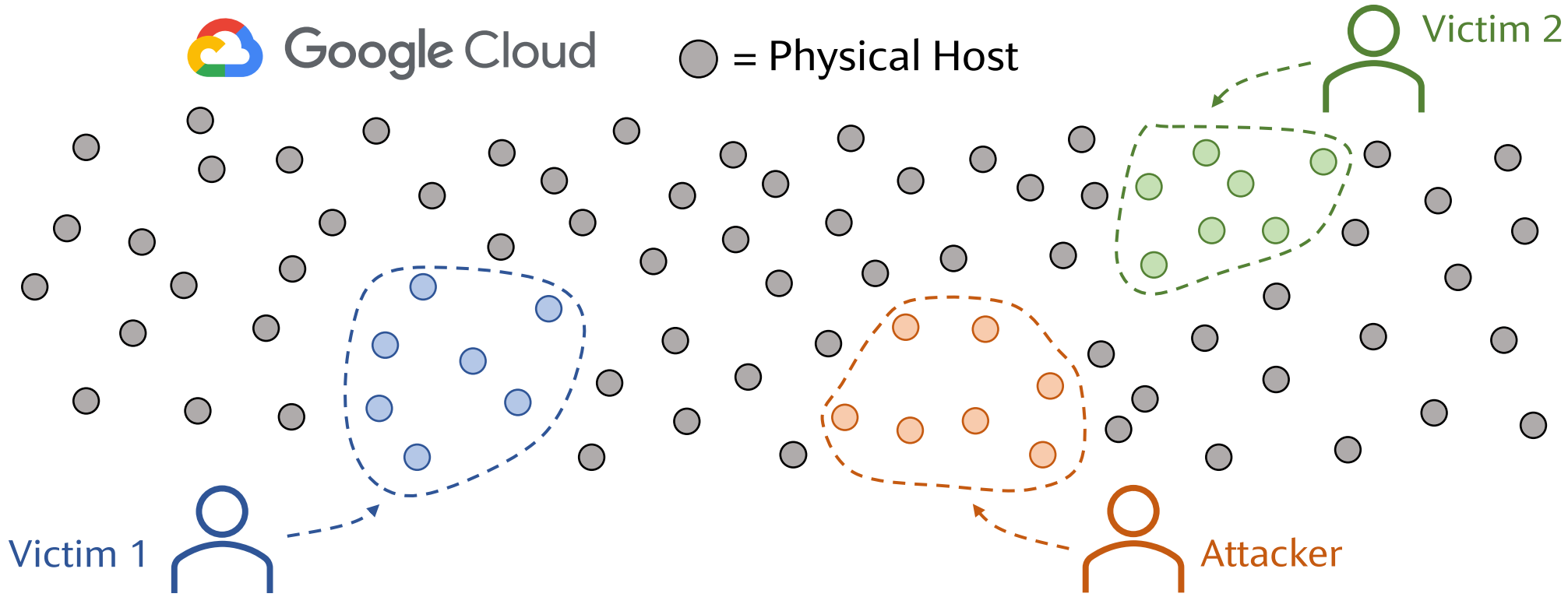
Observation 1: An Account Has a Preferred Set of Hosts

Why: Affinity scheduling to reduce communication overhead

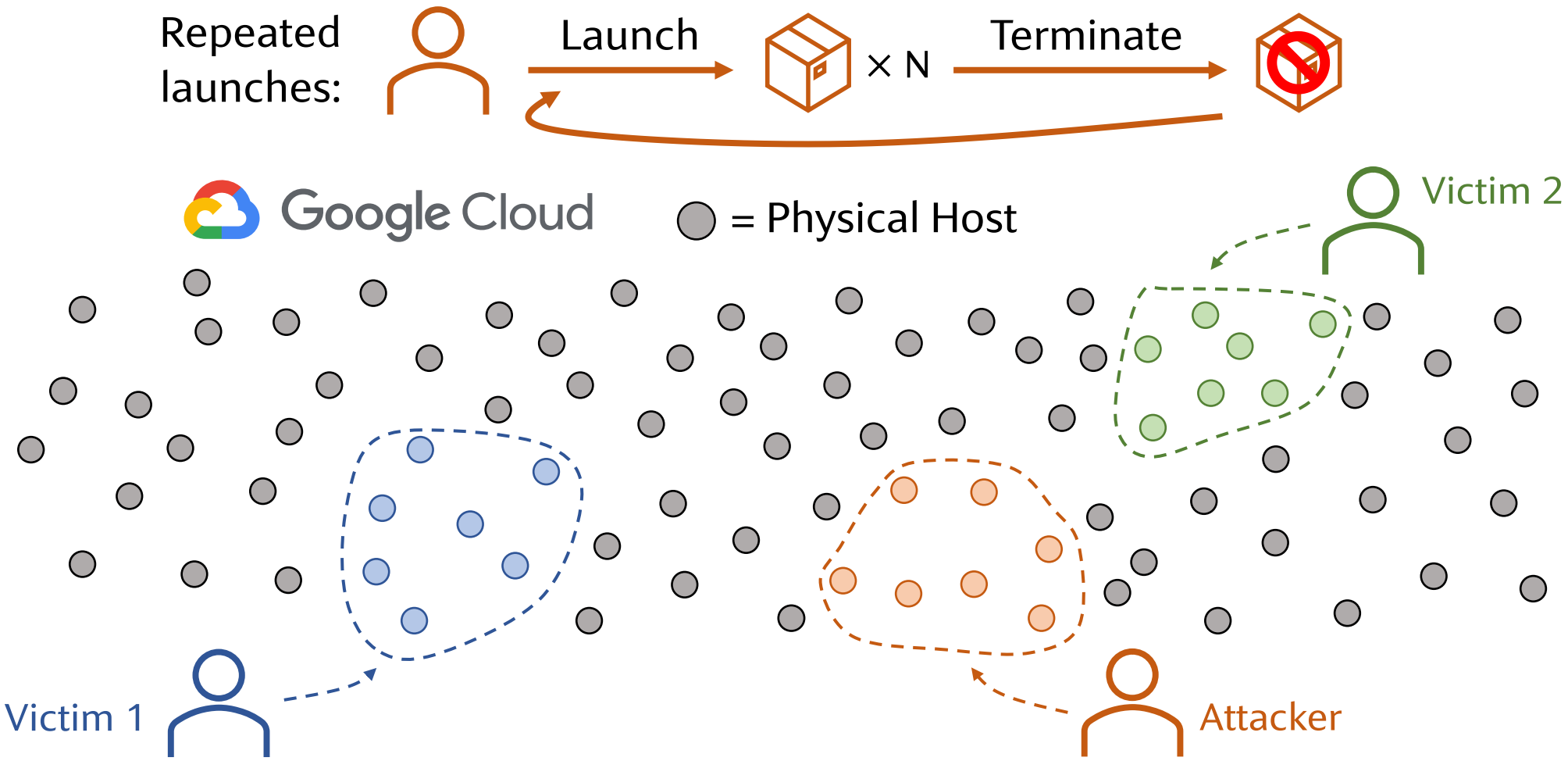


Observation 2: Different Accounts Have Different Preferred Hosts

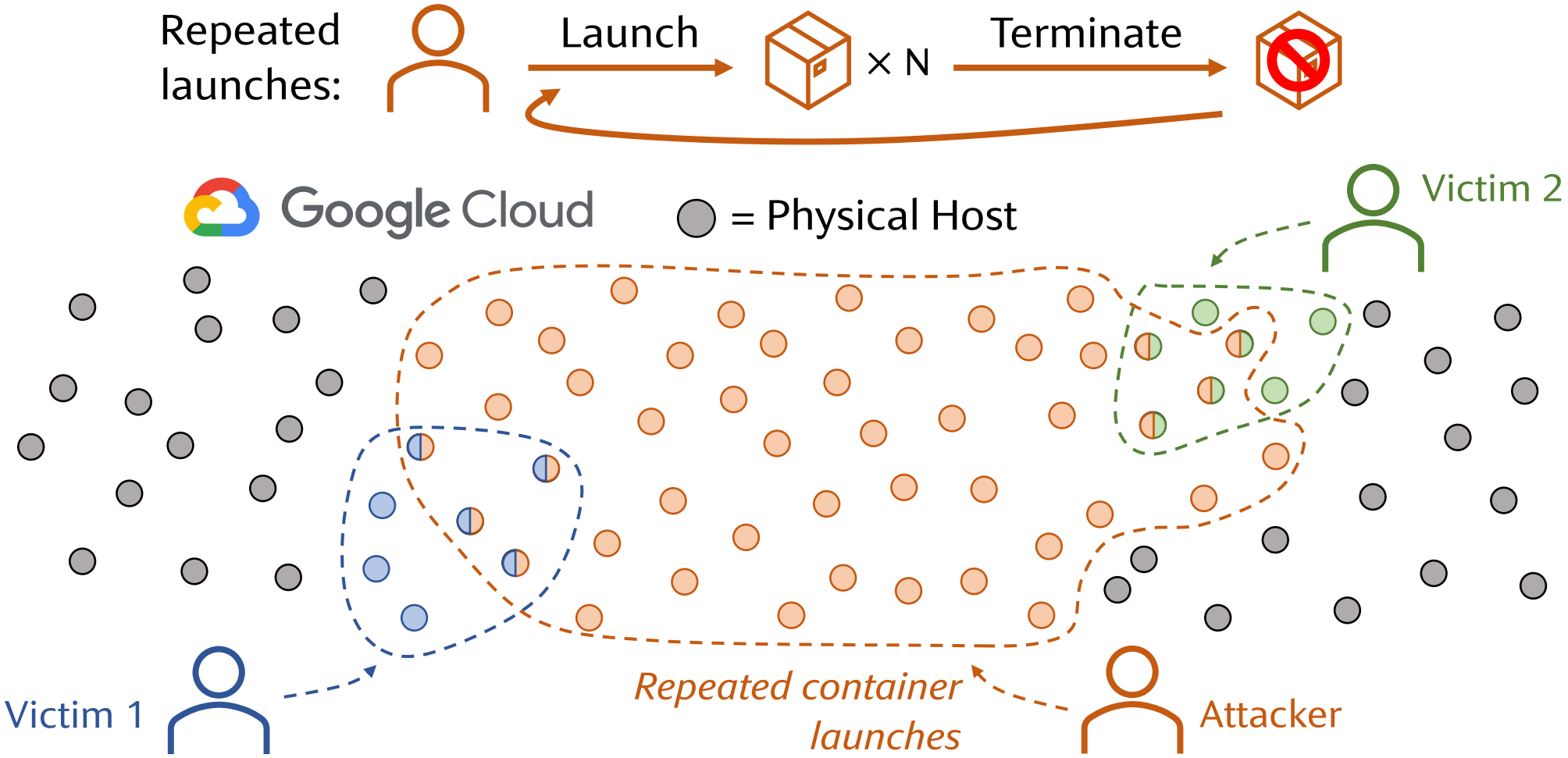
Implication: Low chance of co-location with a target user



Observation 3: Repeated Launches Spread Instances

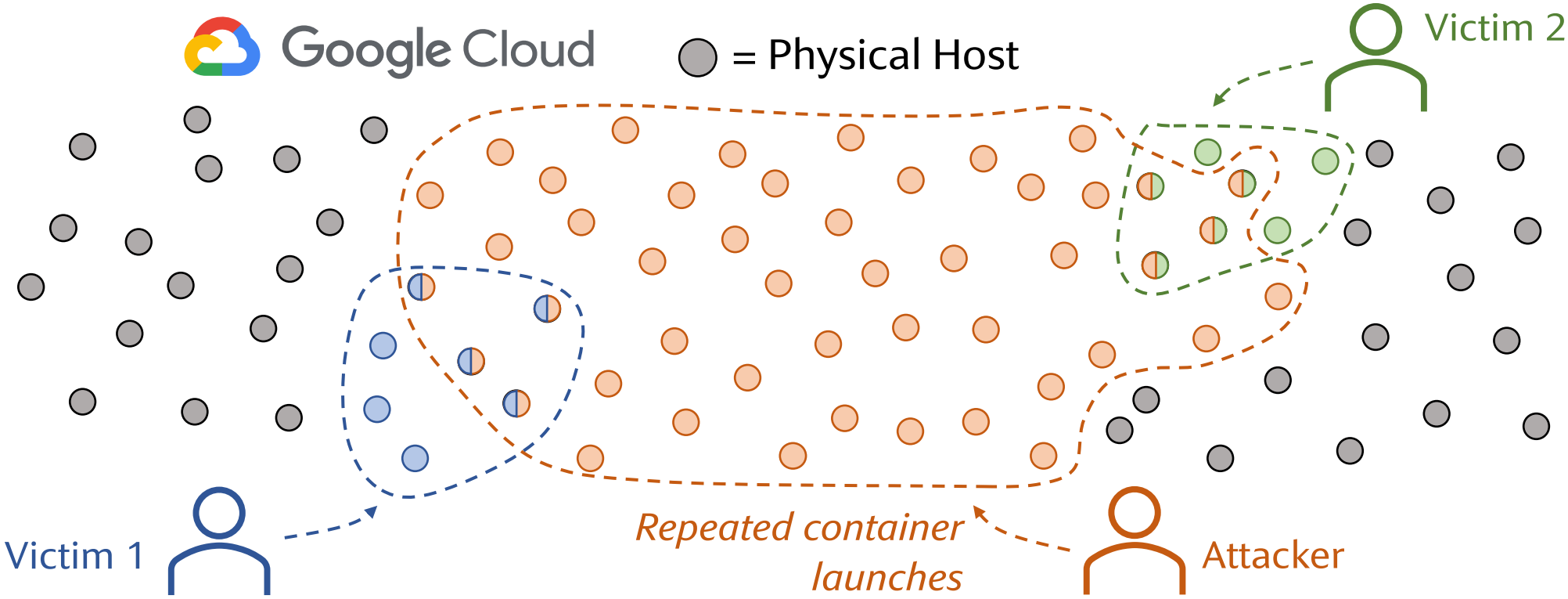


Observation 3: Repeated Launches Spread Instances



Observation 3: Repeated Launches Spread Instances

Why: Repeated launches ⇒ User has high demand ⇒ Load balance

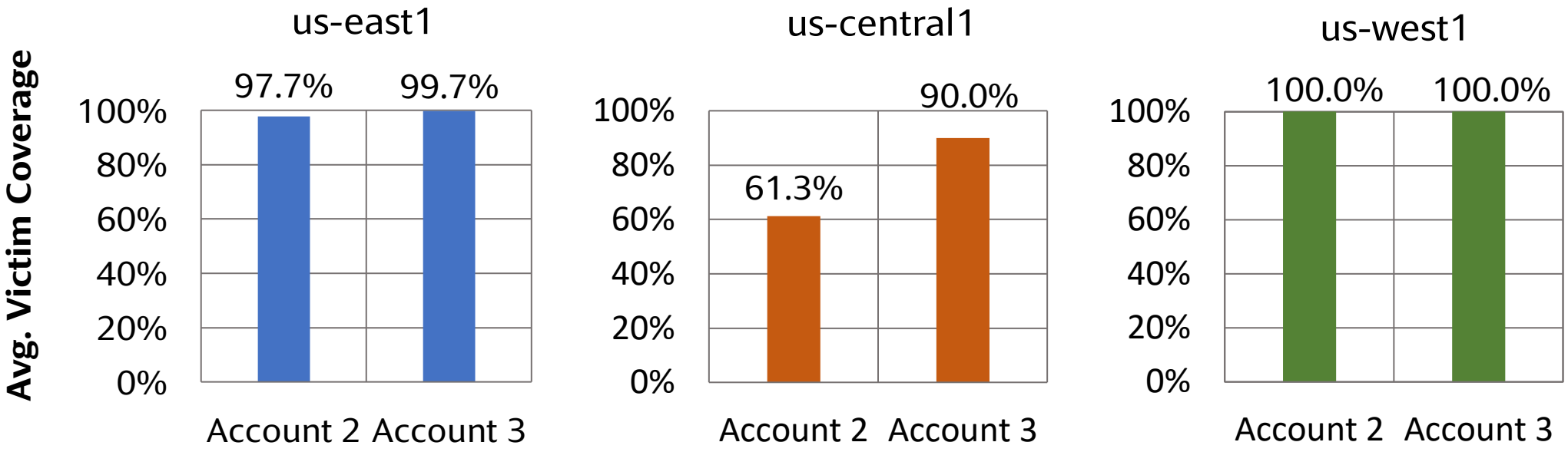


Evaluation: Co-Location with Victims

Victim coverage: Percentage of victim instances that are co-located with the attacker

High Victim Coverage and Low Attack Cost

Average Victim Instance Coverage (3 repetitions in each region)

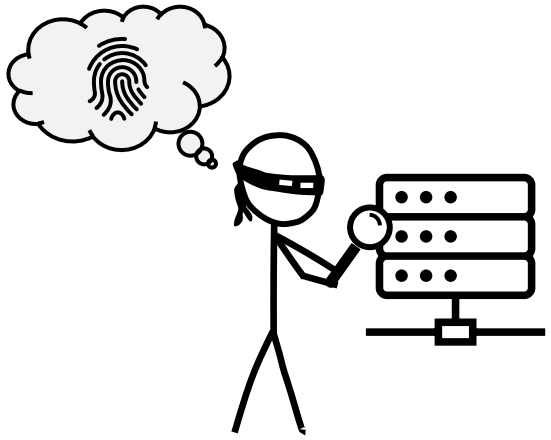


Avg. Attack Cost: 24 USD 23 USD 27 USD

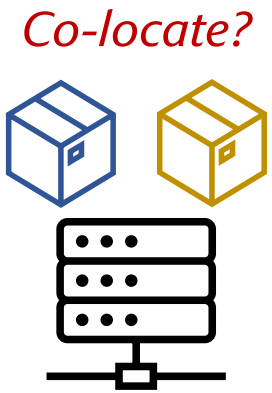
Takeaway: High victim coverage and low attack cost

Conclusions

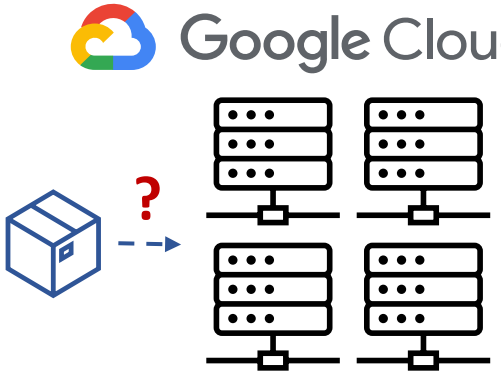
Insight: Fingerprint Host → Reverse Engineer Placement Behavior



1. Accurate
Host Fingerprinting



2. Inexpensive
Co-Location Test

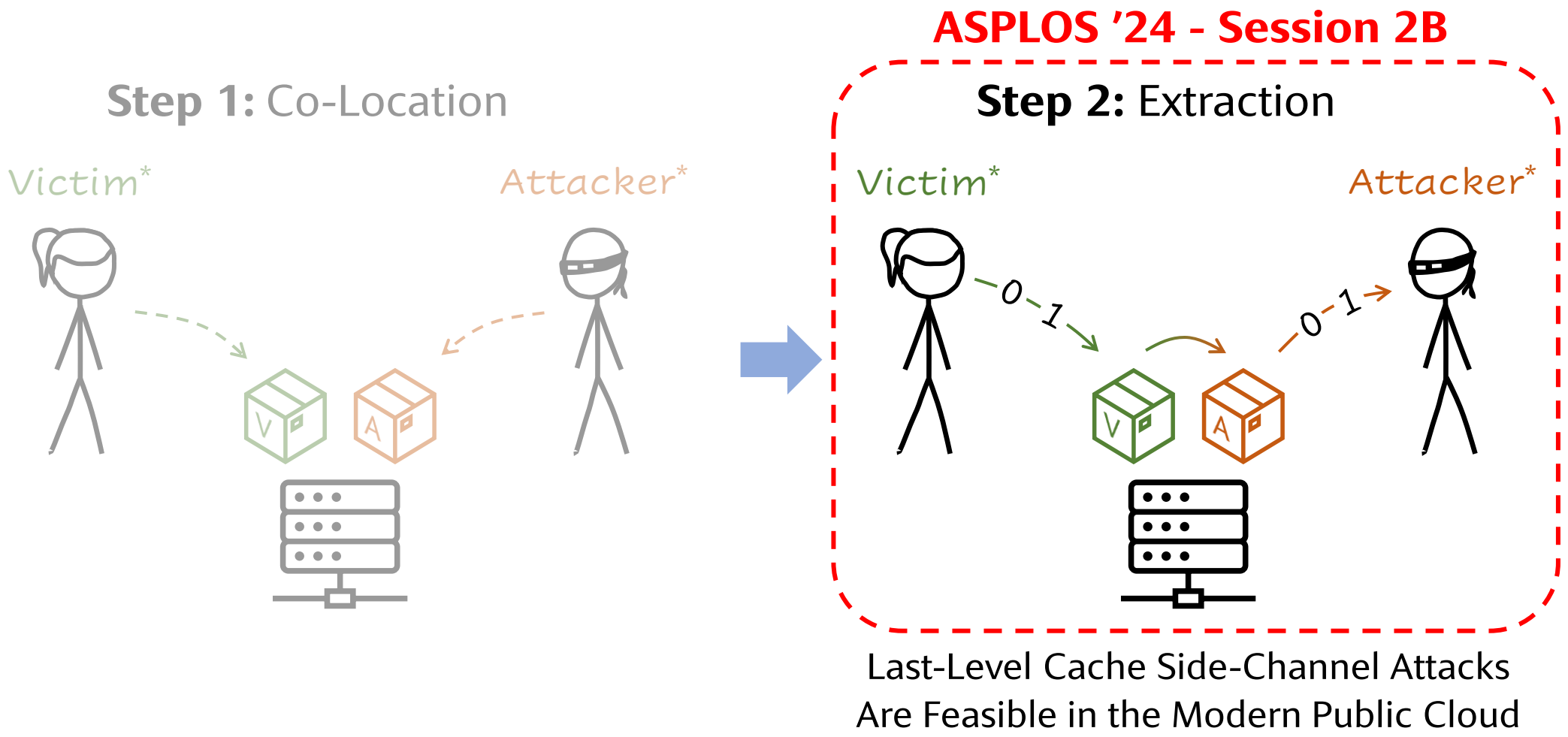


3. Exploitable
Placement Behavior

⇒ High victim coverage and low attack cost

GitHub Repo: https://github.com/zzrcxb/EAAO

Steps of Side-Channel Attacks in Public Cloud



Last-Level Cache Side-Channel Attacks
Are Feasible in the Modern Public Cloud

*Characters are based on https://xkcd.com/2176 and https://xkcd.com/1808 (under a CC Attribution-NonCommercial 2.5 License)