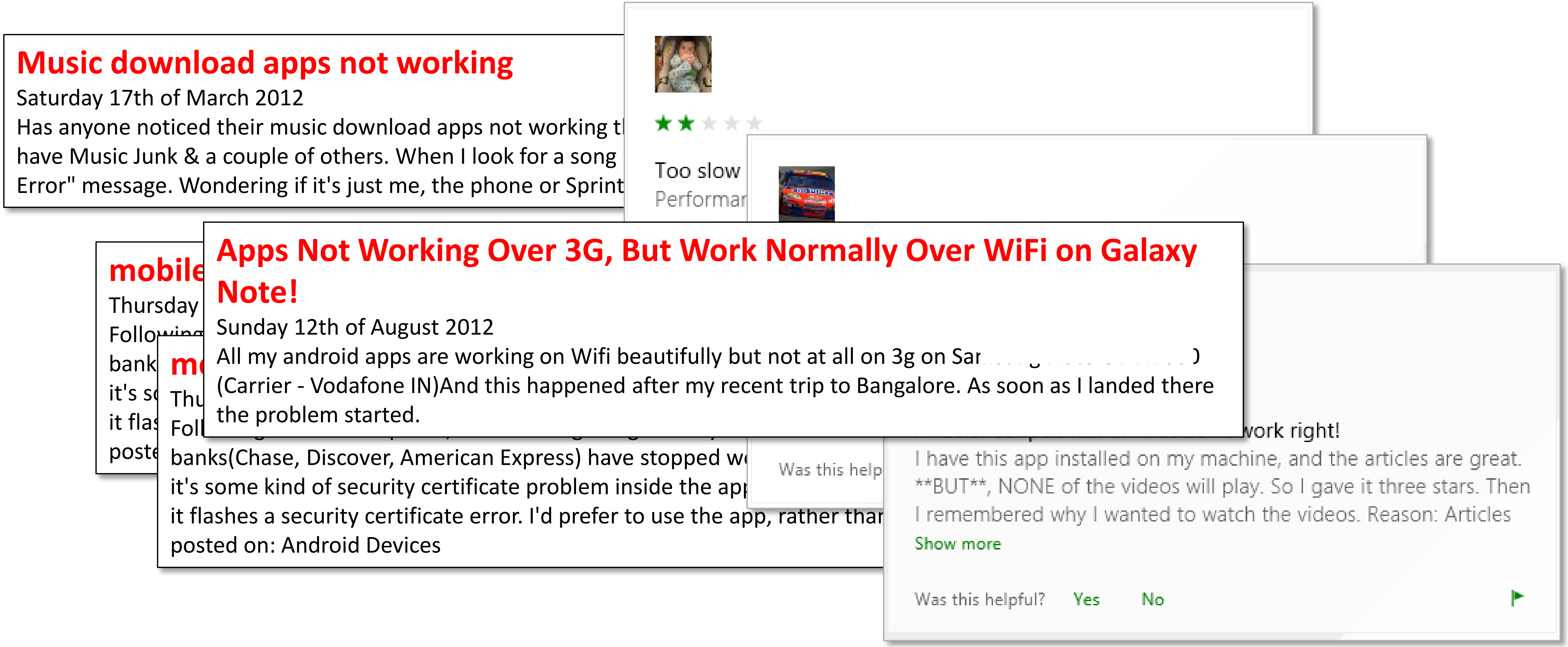
Caiipa: Automated Large-scale Mobile App Testing through Contextual Fuzzing

Chieh-Jan Mike Liang (MSR); Nicholas D. Lane (MSR);
Niels Brouwers (Delft); Li Zhang (USTC); Borje F. Karlsson (MSR);
Hao Liu (Tsinghua); Yan Liu (SJTU); Jun Tang (Harbin);
Xiang Shan (Harbin); Ranveer Chandra (MSR); Feng Zhao (MSR)

> Two Million Apps

> 500,000 Developers

The Health of the App Eco-system Drives the Mobile Platform Success… *But…*



Music download apps not working

Saturday 17th of March 2012

Has anyone noticed their music download apps not working t
have Music Junk & a couple of others. When I look for a song
Error" message. Wondering if it's just me, the phone or Sprint

★★☆☆☆

Too slow
Performar

mobile
Thursday
Following
bank
it's s
it flas
poste

Apps Not Working Over 3G, But Work Normally Over WiFi on Galaxy Note!

Sunday 12th of August 2012

All my android apps are working on Wifi beautifully but not at all on 3g on Sar
(Carrier - Vodafone IN)And this happened after my recent trip to Bangalore. As soon as I landed there the problem started.

m
Thu
Foll
banks(Chase, Discover, American Express) have stopped w
it's some kind of security certificate problem inside the app
it flashes a security certificate error. I'd prefer to use the app, rather than
posted on: Android Devices

Was this help

work right!

I have this app installed on my machine, and the articles are great. **BUT**, NONE of the videos will play. So I gave it three stars. Then I remembered why I wanted to watch the videos. Reason: Articles

Show more

Was this helpful? Yes No

Diverse Real-world Contexts to Consider

Various inputs

- User interac
- Senso

Environmental conditions

- Network conditions
- Geo-location
- Mobility trajectory

Contextual Fuzzing

De nfigurations

- CPU
- Memory
- OS

Cases of Bugs Found By Contextual Fuzzing

1. **Location bug**
   - An app by a magazine publisher is 50% more likely to crash outside of US
   - Confirmed by looking at user comments on MarketPlace
2. **Network transition**
   - A chat app can crash when the smartphone transits from Wi-Fi to 3G
   - Confirmed by user tests

Design Goals of Our Testing Service – *Caiipa*

1. **Comprehensive testing coverage (with Contextual Fuzzing)**
   - Fuzz real-world contexts that impact an app's behavior
   - **Result**: Up to 11× more crashes found when considering real-world contexts
2. **Detect unexpected problems**
   - In contrast to simply spot tests for specific failures
   - **Result**: 351 crashes found so far (but not yet reported by users)
3. **Timely and actionable feedback to users**
   - Deal with test state space explosion from considering real-world contexts
   - **Result**: Up to 30.90% more crashes found, under a fixed length of time

Talk Outline

Motivations behind Contextual Fuzzing

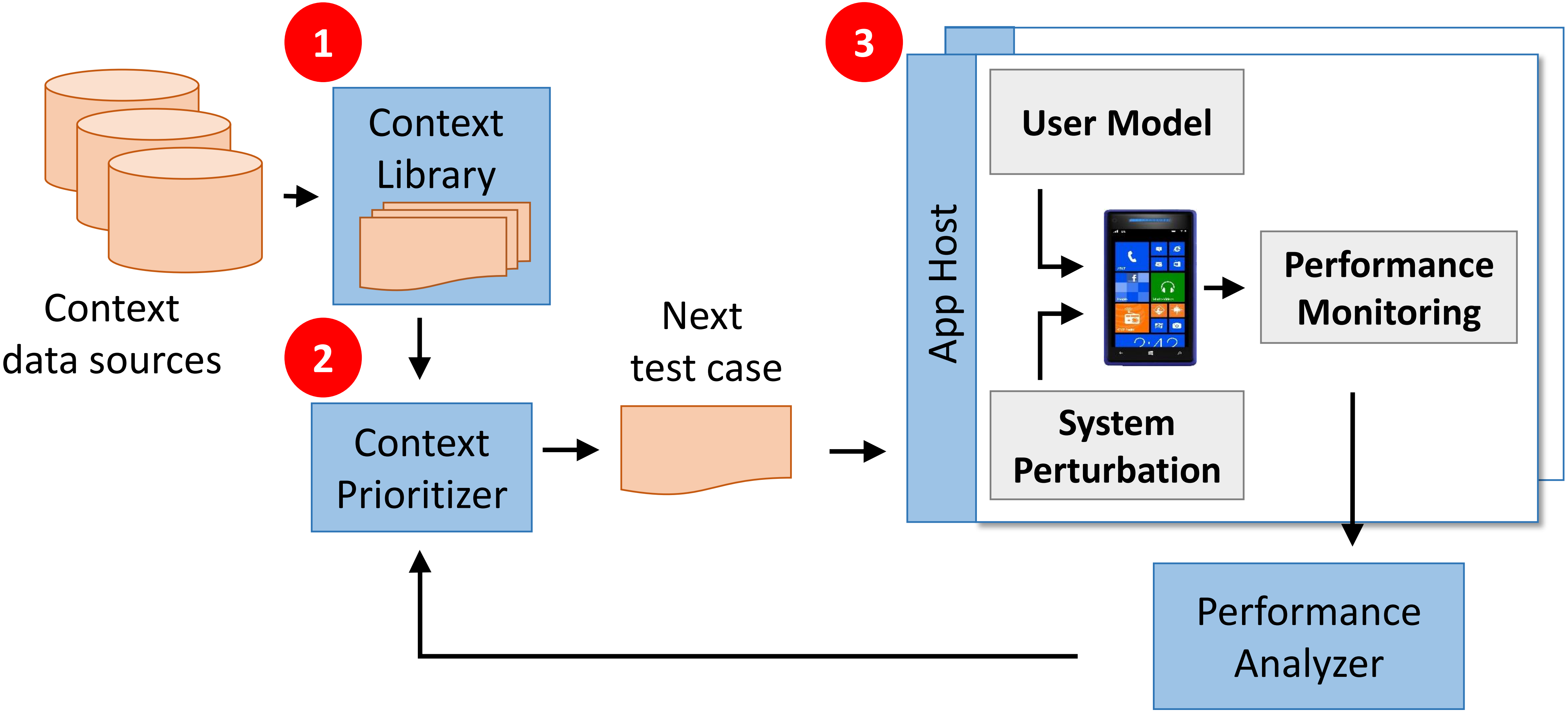
Challenges in realizing Contextual Fuzzing

- Hybrid of physical devices and emulators
- Test prioritization by leveraging app similarity

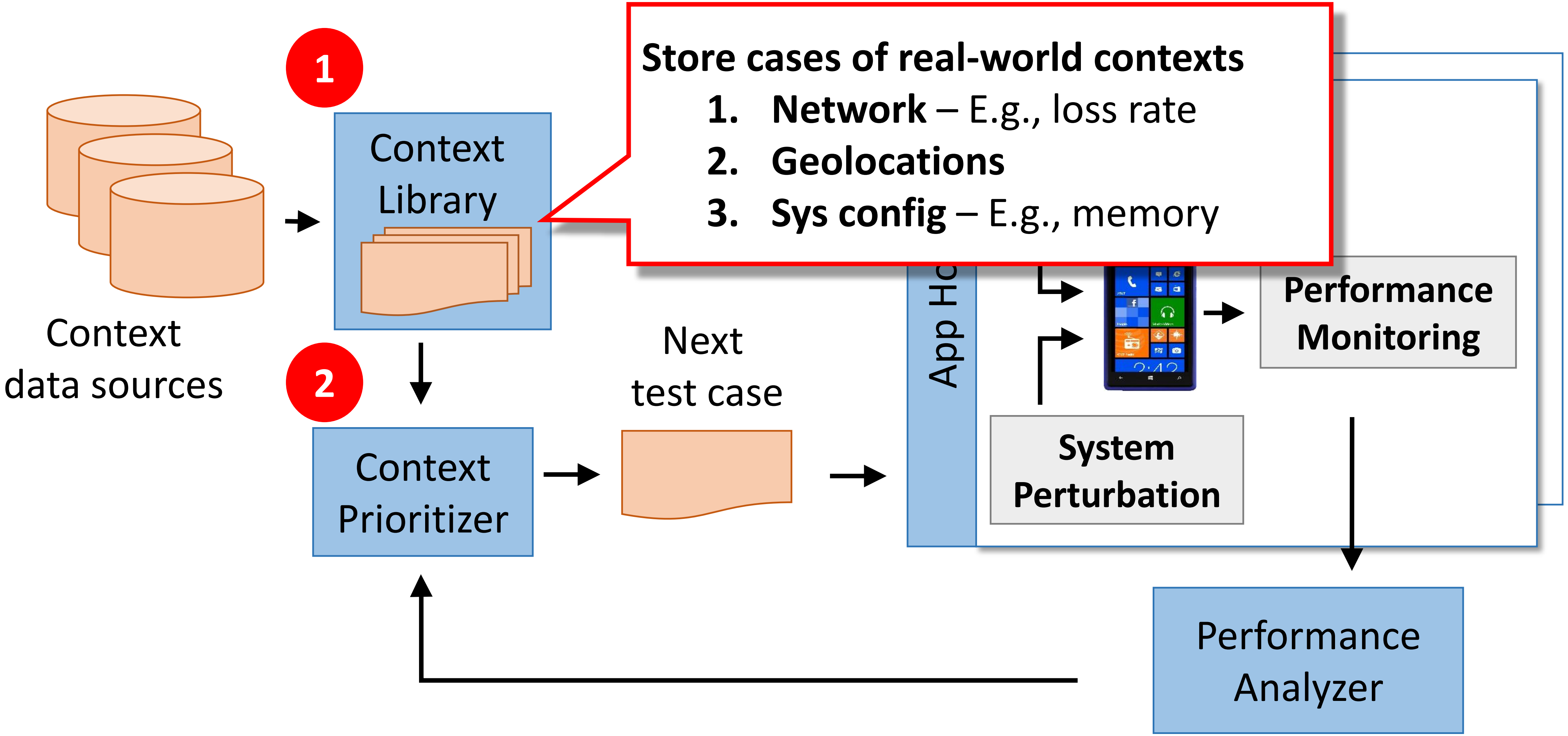
Performance of Caiipa

- System evaluation
- Case studies

Architecture of *Caiipa*



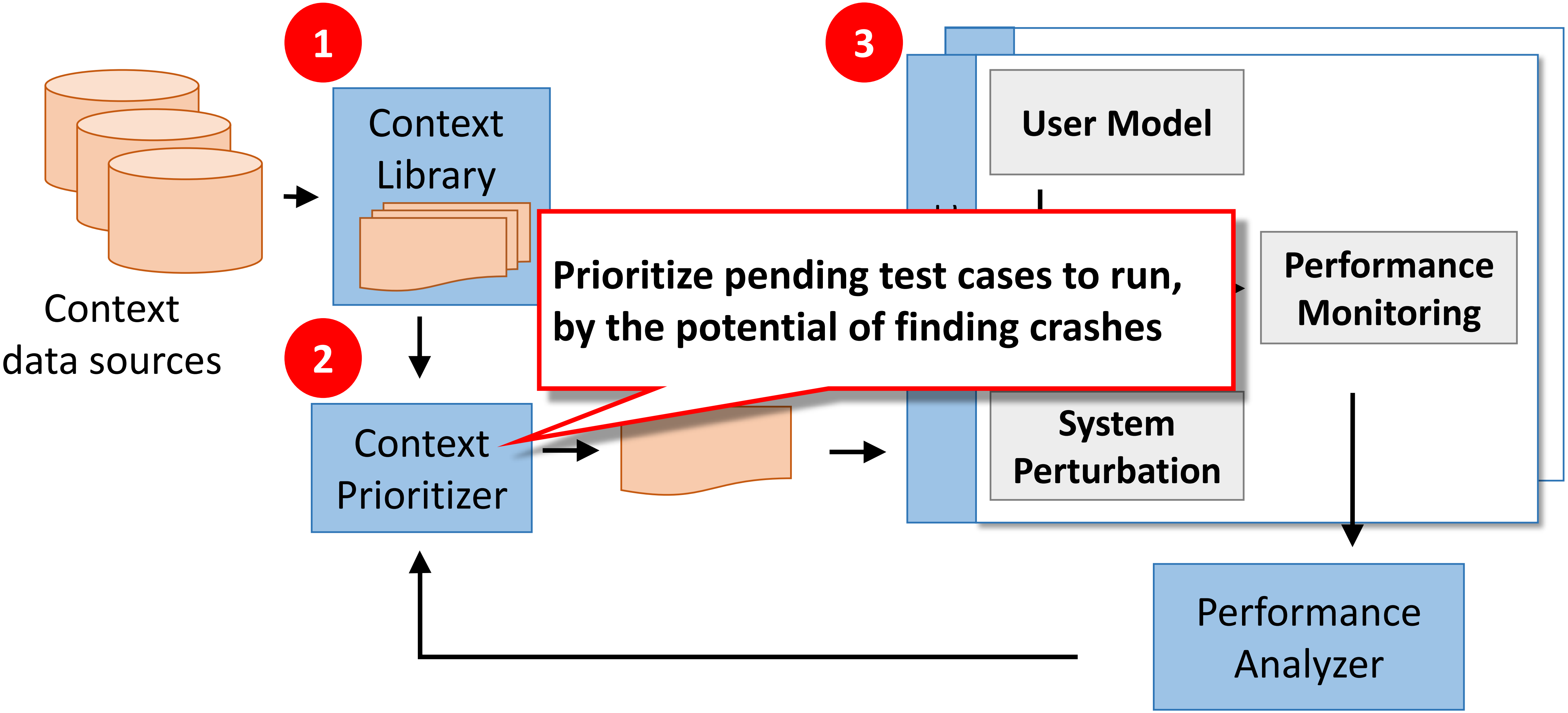
Architecture of *Caiipa*



Store cases of real-world contexts

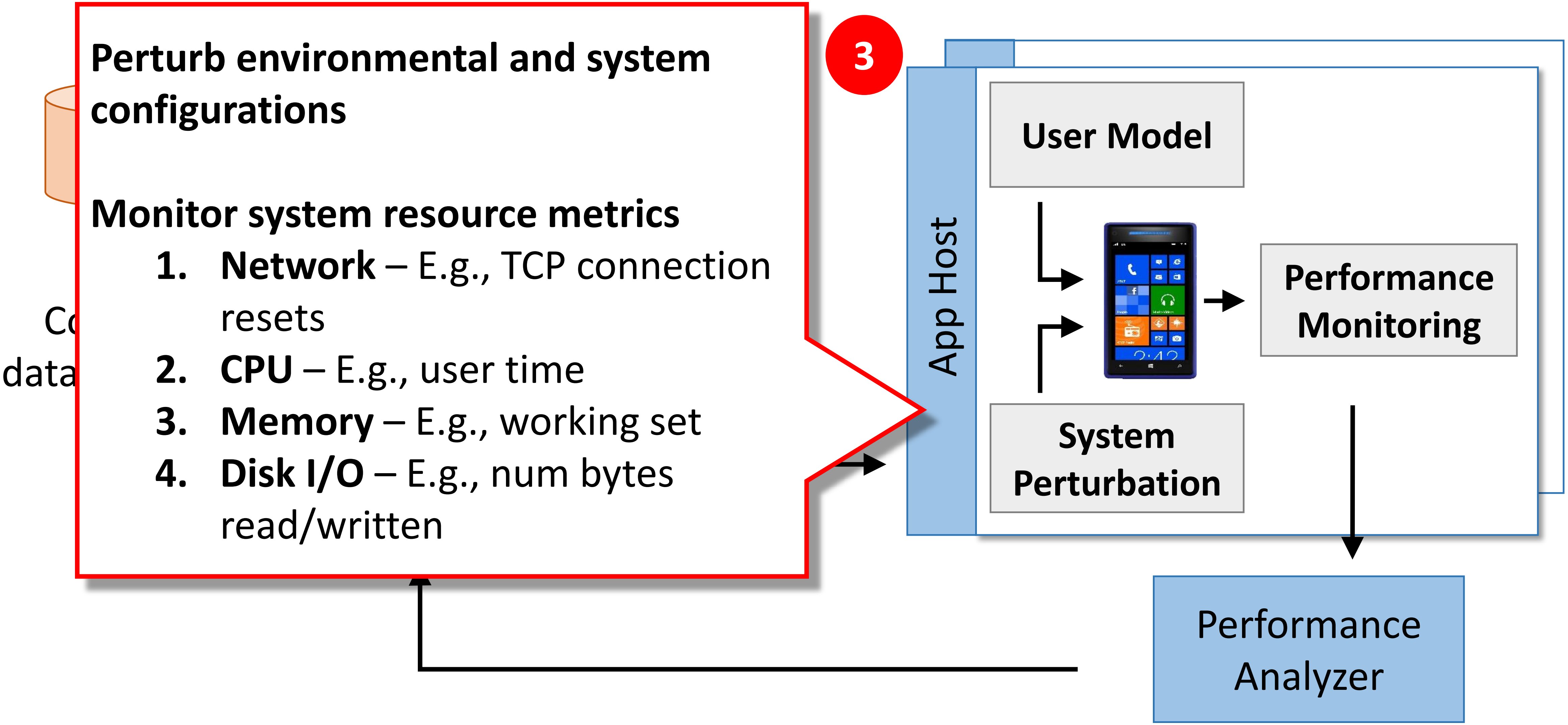
1. **Network** – E.g., loss rate
2. **Geolocations**
3. **Sys config** – E.g., memory

Architecture of *Caiipa*



Prioritize pending test cases to run, by the potential of finding crashes

Architecture of *Caiipa*



Perturb environmental and system configurations

Monitor system resource metrics

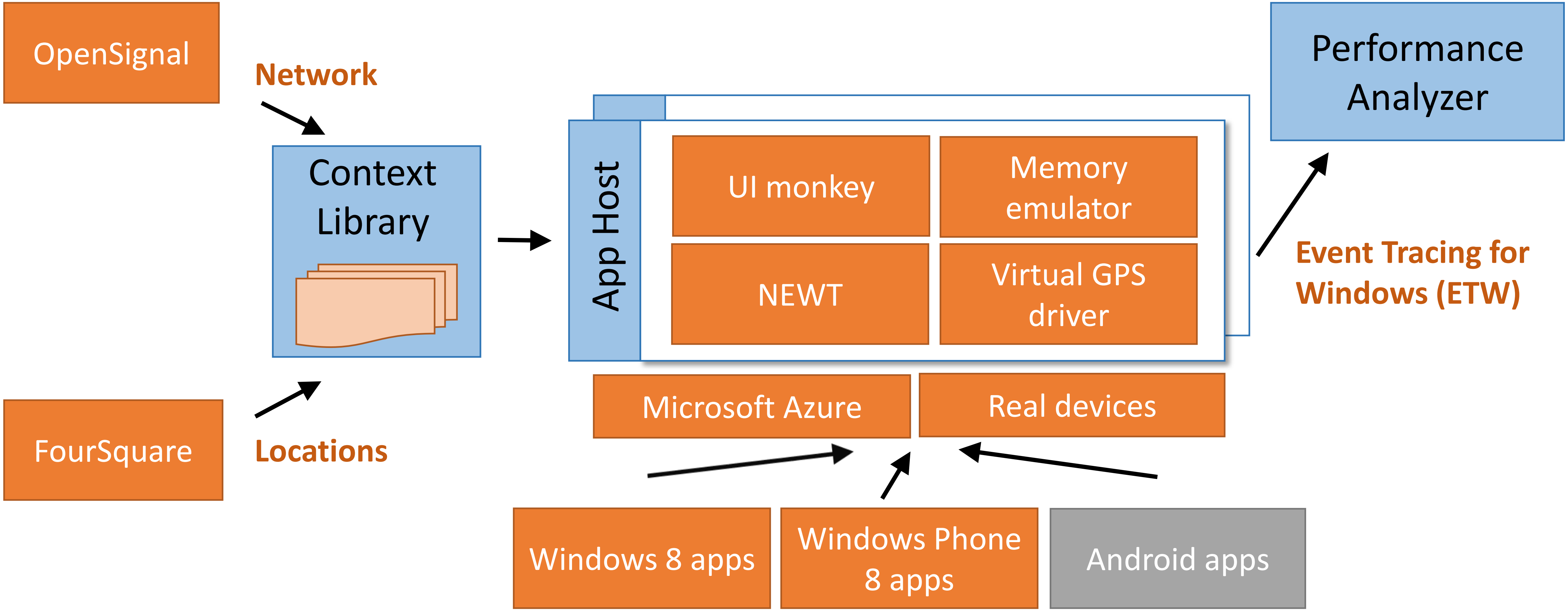
1. **Network** – E.g., TCP connection resets
2. **CPU** – E.g., user time
3. **Memory** – E.g., working set
4. **Disk I/O** – E.g., num bytes read/written

Challenges of Contextual Fuzzing: Scalability

1 **Testing with only physical devices does not scale up**

- Hindered by device quantity, and available real-world contexts
- **Solution**: Complement the system with emulation of real-world contexts
  1. User interactions
  2. Network conditions
  3. Available memory
  4. Geo-locations

Caiipa service in real life



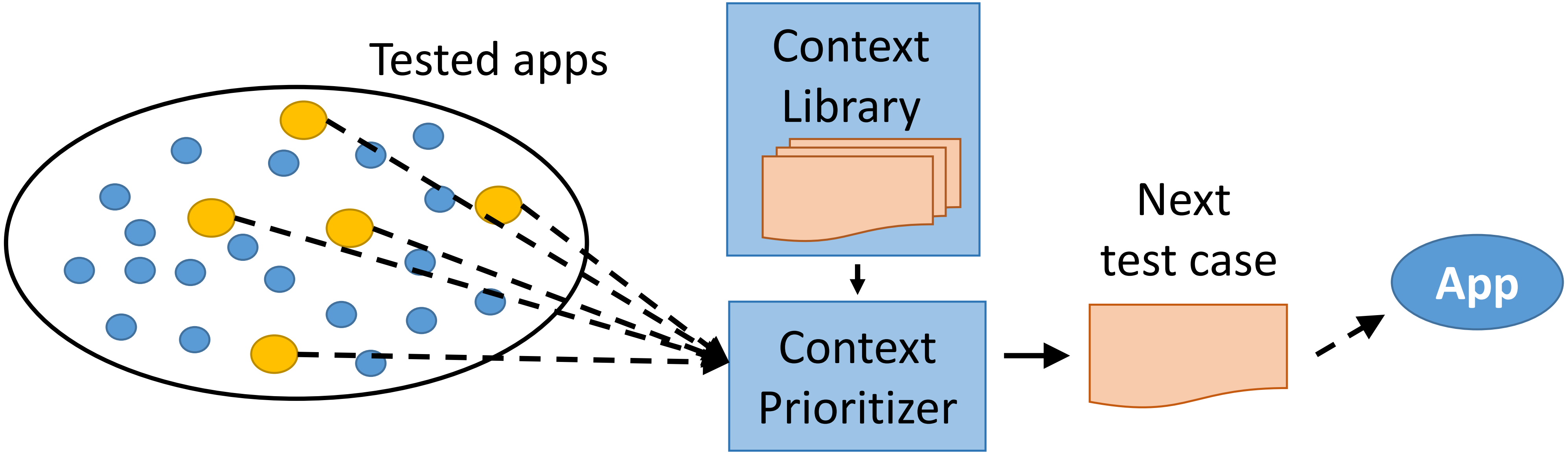
Challenges of Contextual Fuzzing: Scalability

2 State space explosion from numerous real-world contexts

- 10,504 contextual test cases currently in our library
- **Solution**: Test case prioritization with “app similar sets”

Test Case Prioritization

Idea

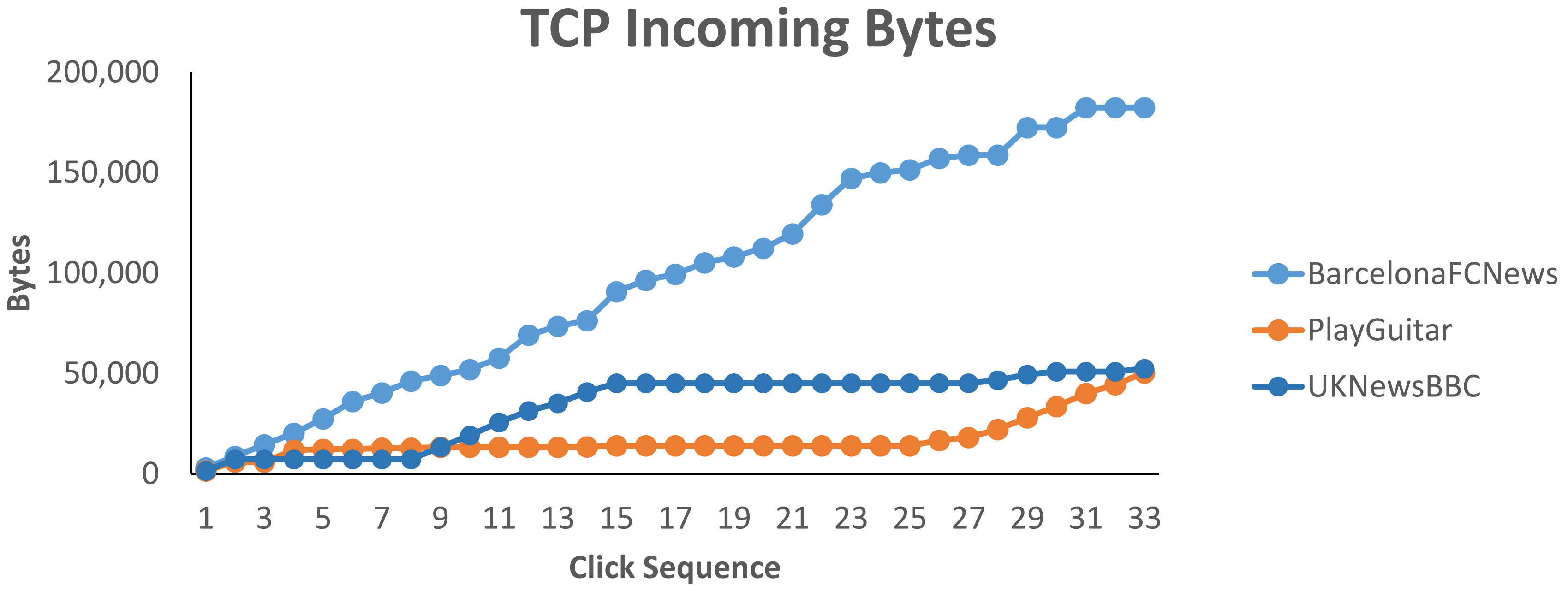


Overview of steps

1. Run the current app under *GPRS*, *802.11b*, and *4G* test cases
2. Find resource-based similarity set, $\mathrm{AppSimSet}$ (*explained next*)
3. Count crashes in pending test cases, as observed by $\mathrm{AppSimSet}$
4. Sort pending cases in descending order

Test Case Prioritization – App Similar Set

- **Functional categorization does not work well in identifying resource-based similarity sets**
    - E.g., not all news apps consume TCP traffic similarly

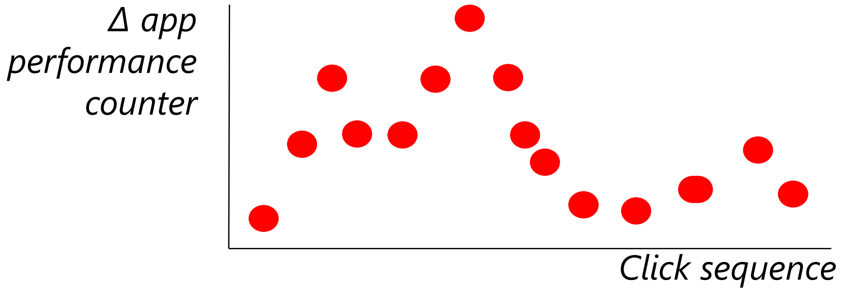


Test Case Prioritization – App Similar Set

Idea: “Similar” apps consume system/network resources in a similar way

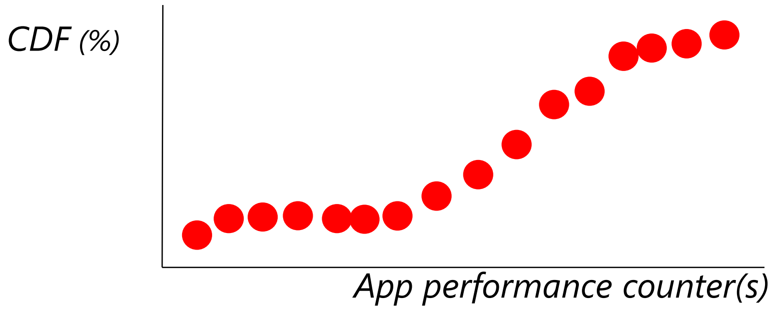
Step 1: *Extract* features

- Features: Changes in resource metrics after each UI click



Step 2: *Compare* features across apps

- Kolmogorov-Smirnov (KS) test: Comparing the CDF of two datasets

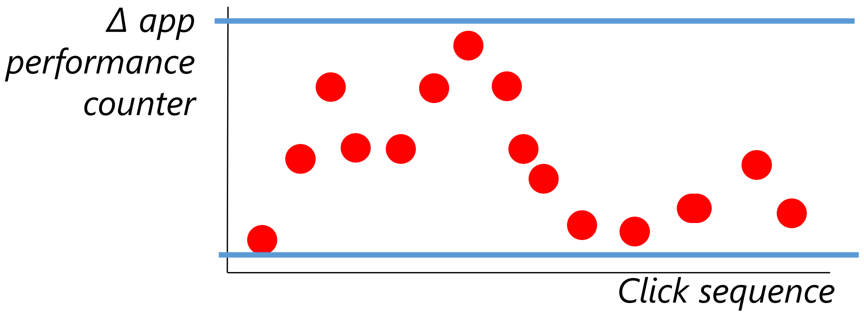


Outputs: One set of similar apps per resource metric

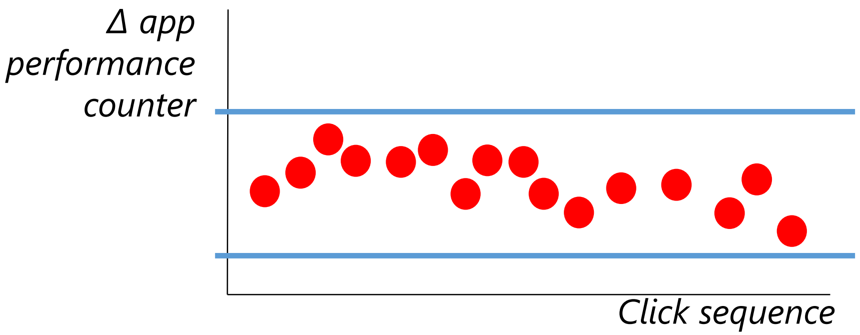
Test Case Prioritization – App Similar Set

Step 3: *Aggregate* per-metric similarity sets to get per-app similarity sets

- Weighted voting (on resource metrics)
- Higher weights are given to system metrics that observe higher fluctuations
  - More distinctive features for evaluating similarity



vs.



Talk Outline

Motivations behind Contextual Fuzzing

Challenges in realizing Contextual Fuzzing

- Hybrid of physical devices and emulators
- Test prioritization by leveraging app similarity

Performance of Caiipa

- System evaluation
- Case studies

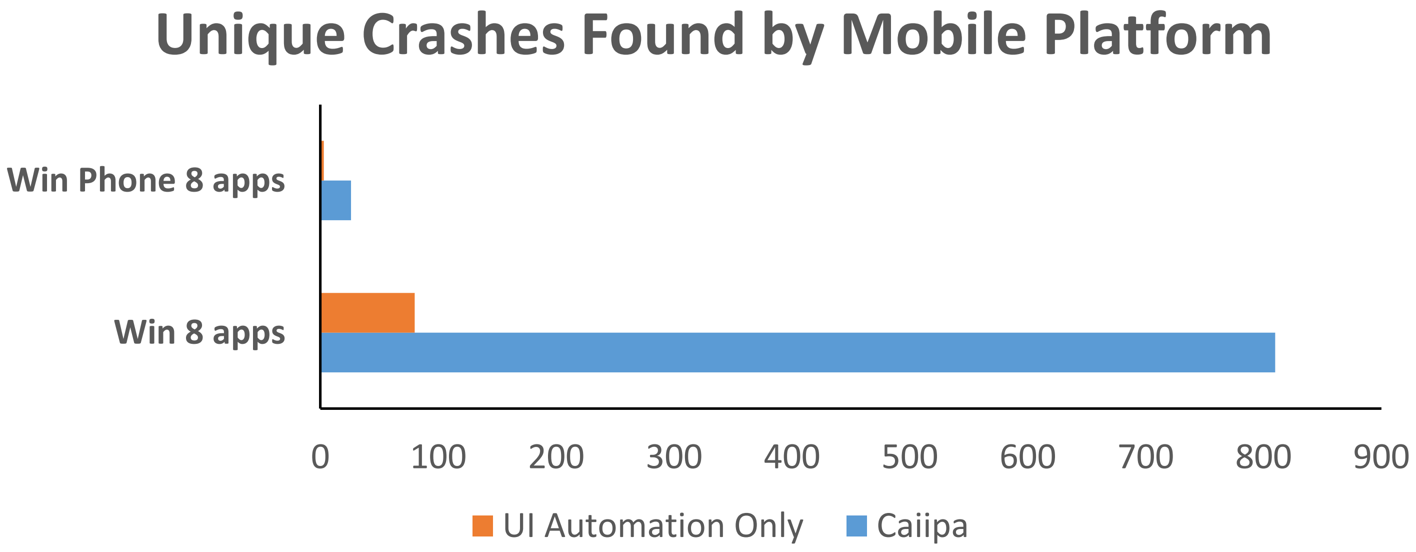
Evaluation Setup

- **Apps available on the market**
  - 235 Windows 8 store apps (targeting tablet devices)
  - 30 Windows Phone 8 apps (targeting smartphones)
- **Emulate three cities with top smartphone users**
  - Seattle, London, and Beijing
- **Emulate network conditions**
  - 350 most frequently observed ones on OpenSignal
  - 5 hard-coded ones: 802.11b, WCDMA, 4G, GPRS_OUT_OF_RANGE, GPRS_HANDOFF

Are Real-World Contexts Really Necessary For App Testing?

Observations

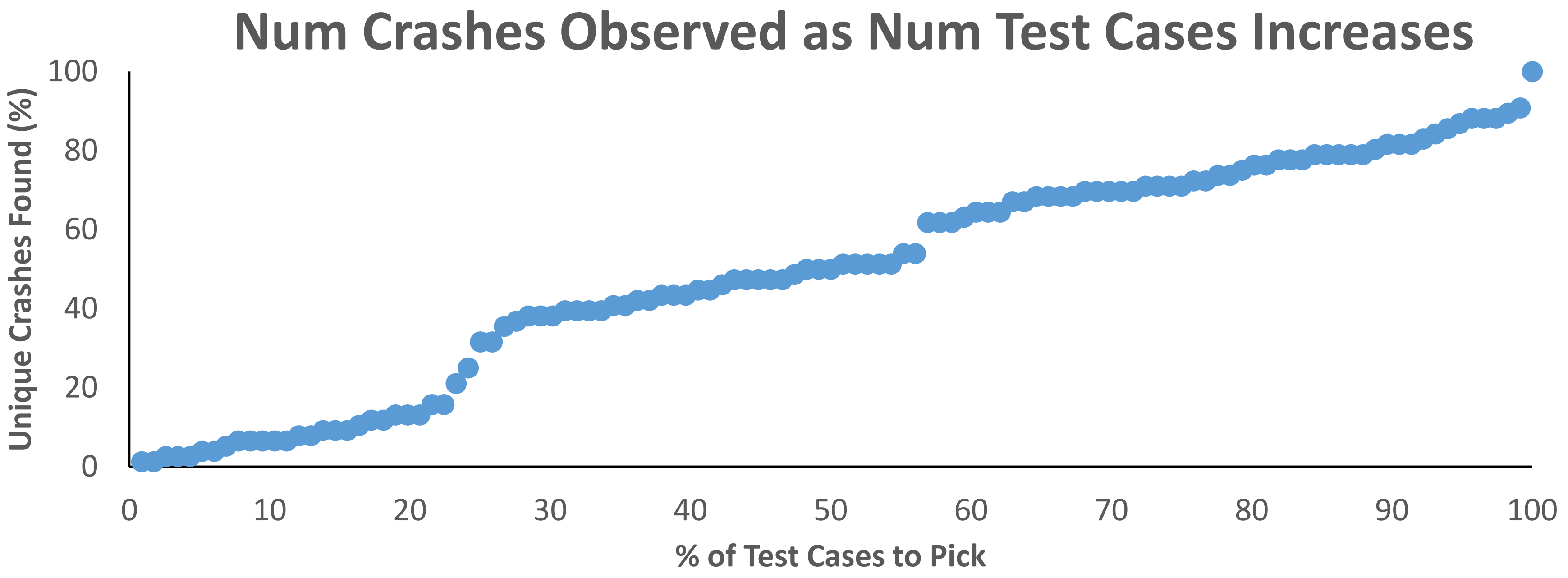
- *Caiipa* can find up to 11× more unique crashes



Do We Need So Many Contextual Test Cases?

Observations

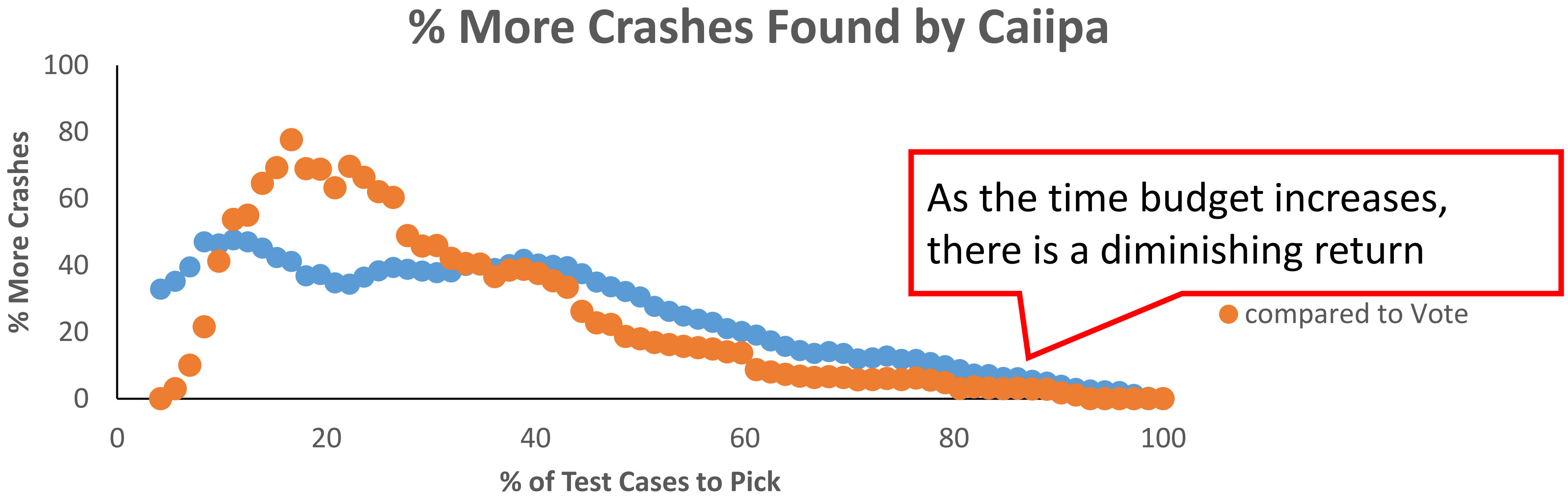
- Apps crash in different test cases
- Running each test case adds additional unique crash observations



How Does Context Prioritizer Perform?

Observations

- Given fixed time budget, *Caiipa*'s test prioritizer finds an average of 30.90% and 28.88% more crashes than *Random* and *Vote*



As the time budget increases, there is a diminishing return

Conclusion

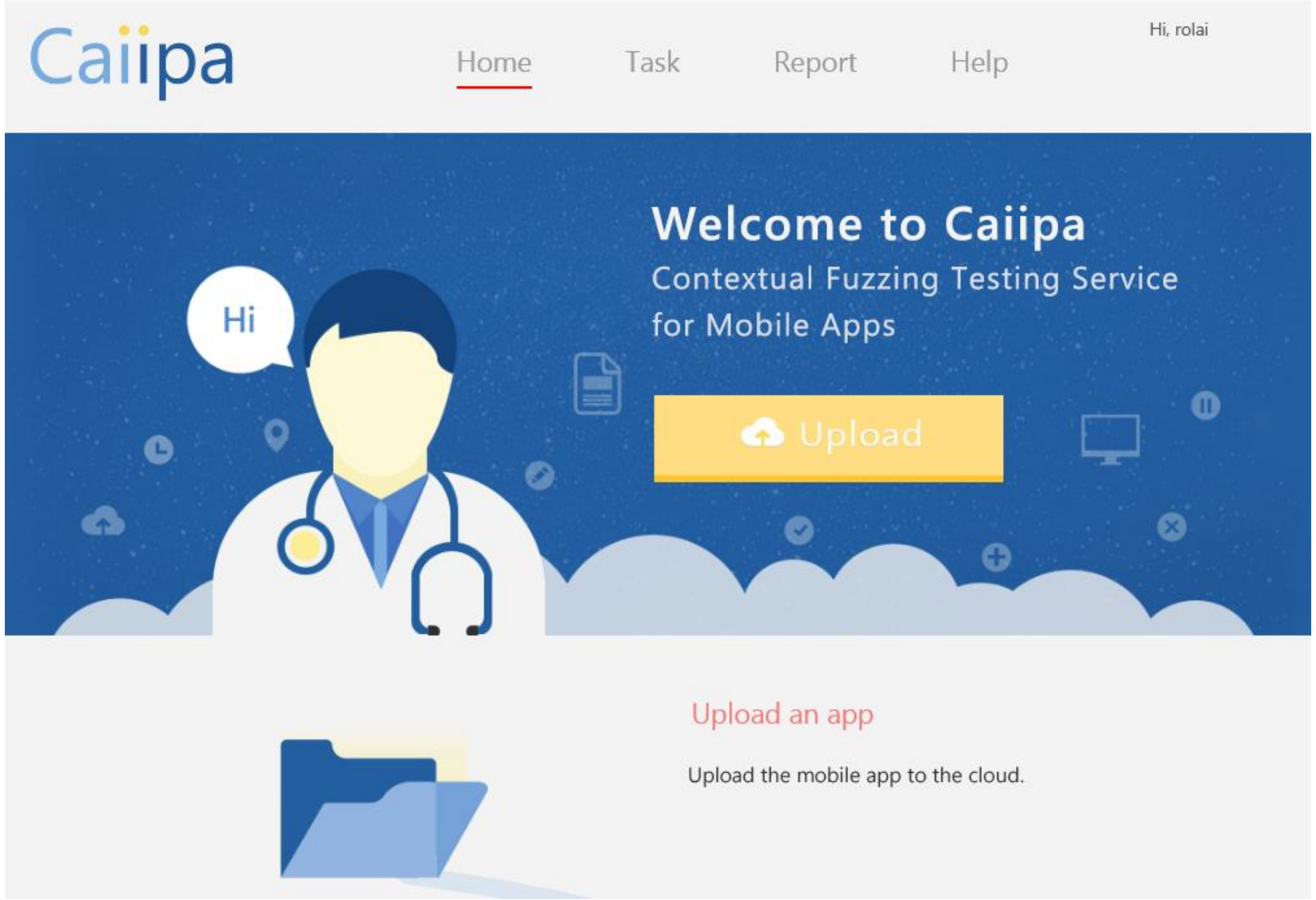
Summary

- *Caiipa* implements Contextual Fuzzing for better app testing coverage

Major results

- Find up to 11× more unique crashes, by considering real-world contexts
- Find up to 30.90% more crashes (under a fixed length of time), by prioritizing test cases with app similarity set

Thank You!



Welcome to Caiipa

Contextual Fuzzing Testing Service for Mobile Apps

Upload an app

Upload the mobile app to the cloud.