# Zero-Config Fuzzing for Microservices

ASE'23

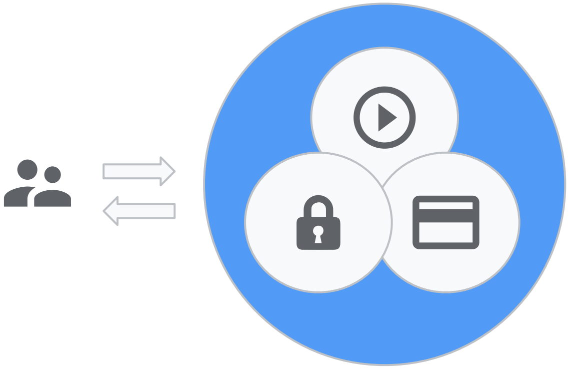
Wei Wang

Andrei Benea

Franjo Ivančić

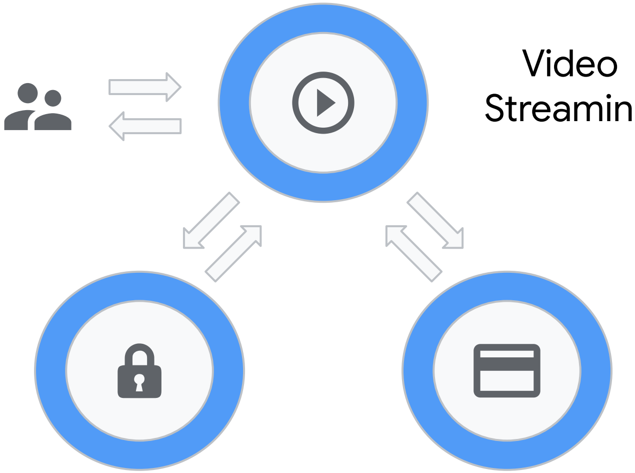
# Microservice

## Monolithic Architecture



Video Streaming Service

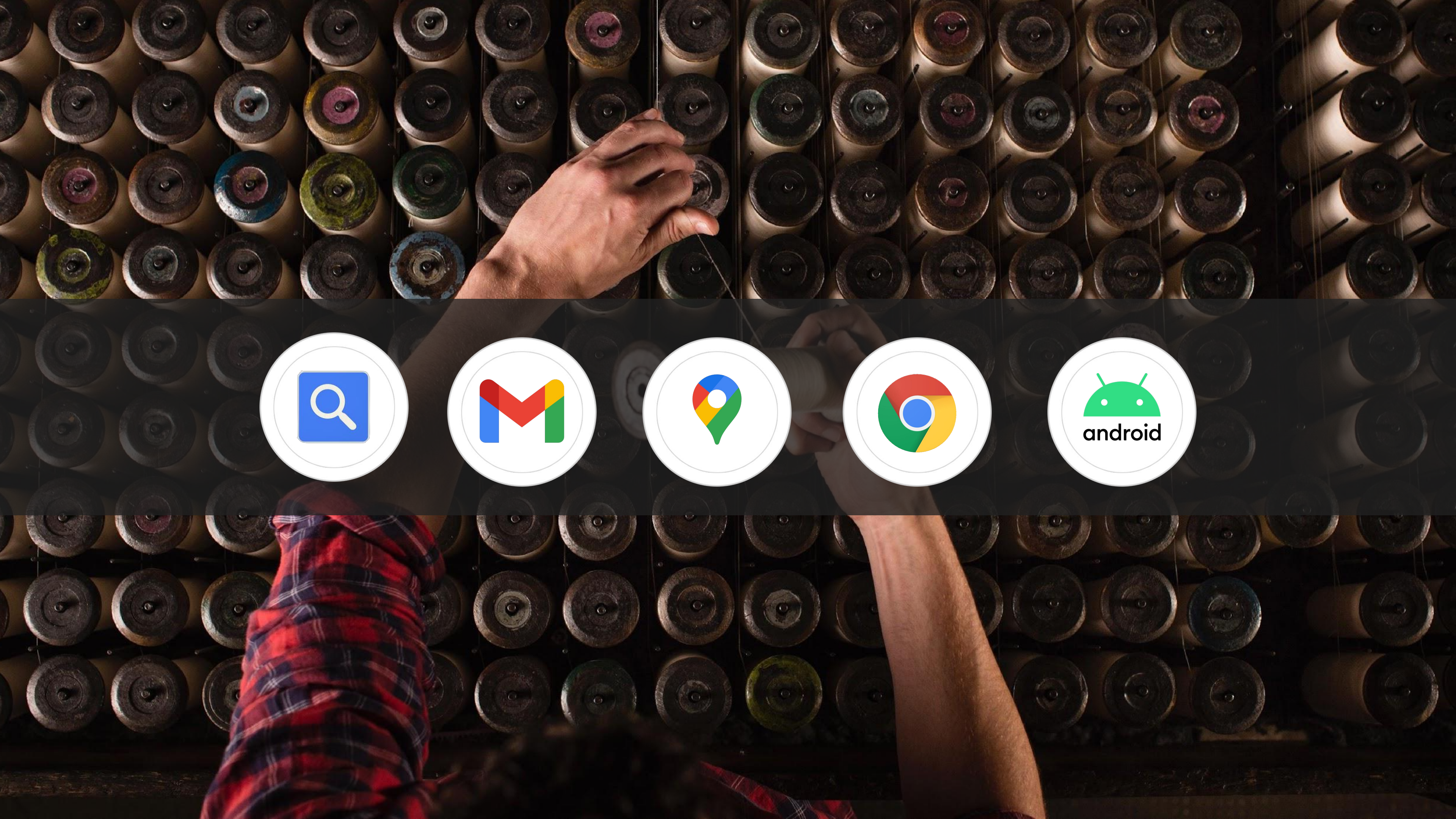
## Microservice Architecture



Video Streaming

Authentication

Payment


android

Development Velocity

Build faster

Encapsulation

# Why Microservices?

Easier to debug

Scalability

Flexibility

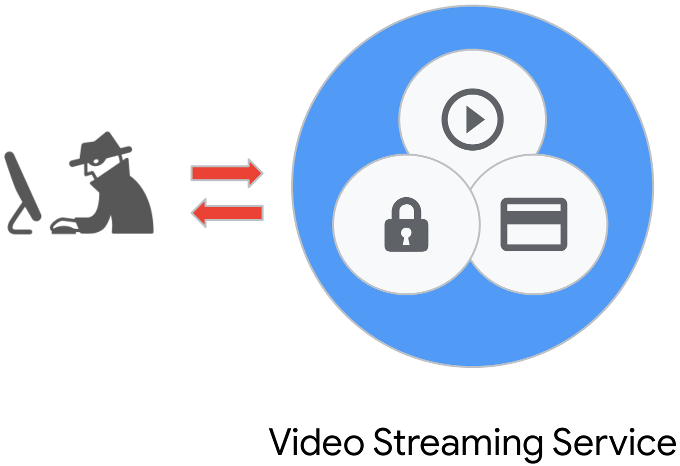
Agility

# Any New Challenges?

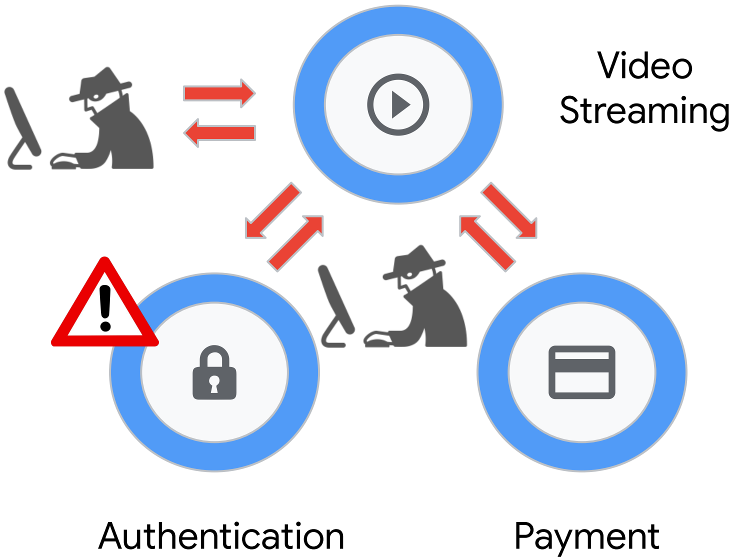
> “The decomposition of an application into a set of distributed and collaborating microservices ...., increases an application’s attack surface.”

P. Nkomo and M. Coetzee, "Software development activities for secure microservices", *Computational Science and Its Applications – ICCSA 2019*, pp. 573-585, 2019.

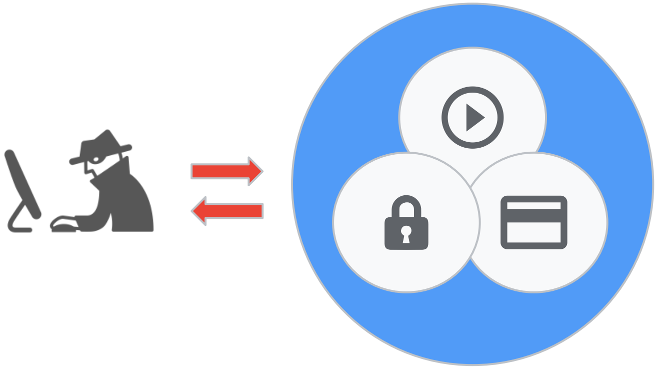
## Monolithic Architecture



## Microservice Architecture

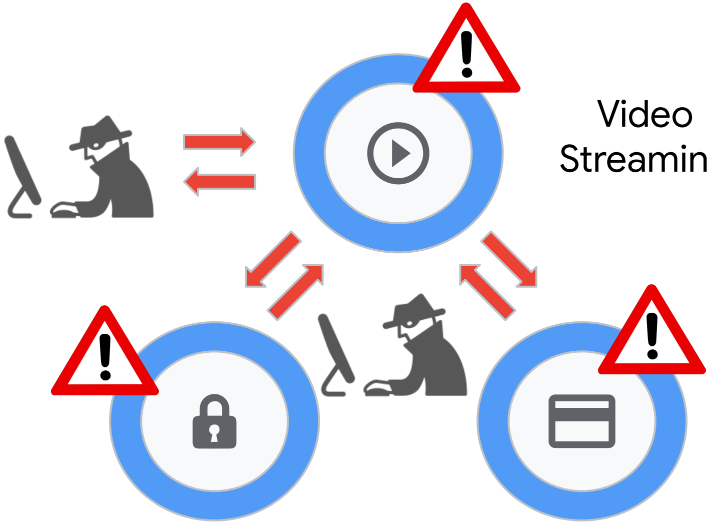


## Monolithic Architecture



Video Streaming Service

## Microservice Architecture



Video Streaming

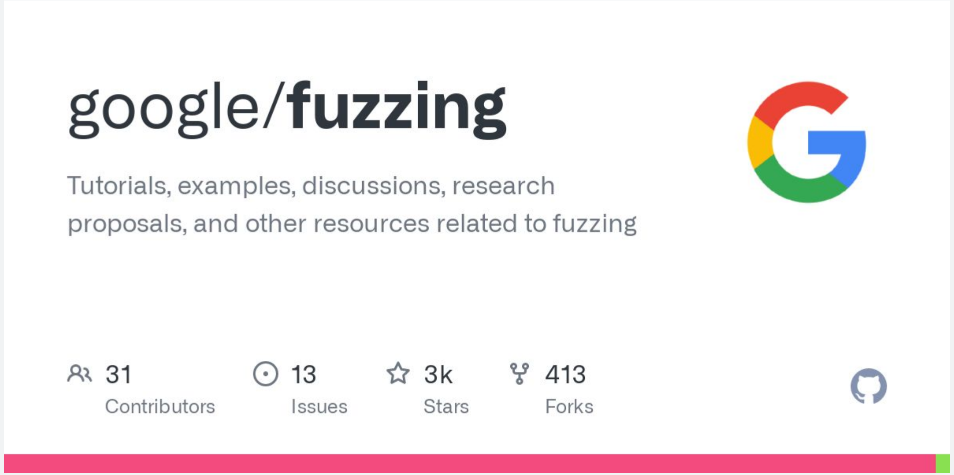
Authentication

Payment

# Fuzzing Microservice

> “Fuzzing is an automated method for detecting bugs in software that works by feeding unexpected inputs to a target program.”

Google Open Source Blog: Open sourcing ClusterFuzz, Feb. 7, 2019



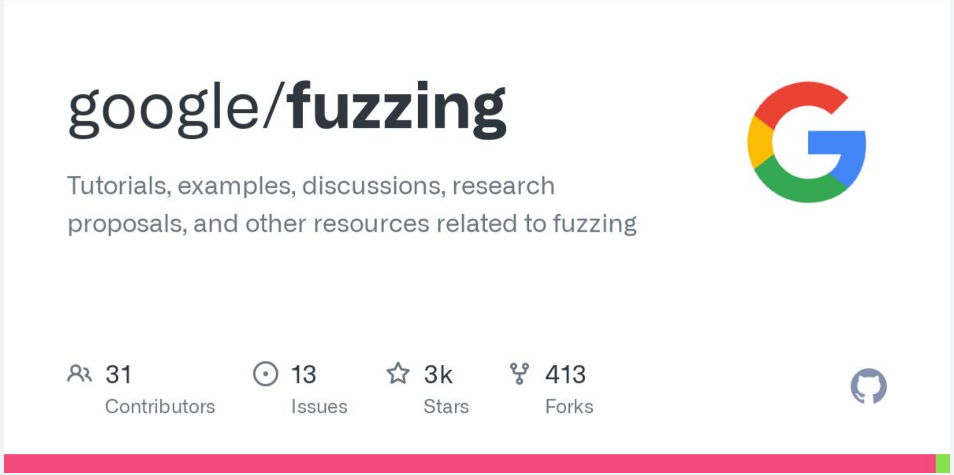
# google/**fuzzing**

Tutorials, examples, discussions, research proposals, and other resources related to fuzzing

| 31 | 13 | 3k | 413 |
|---|---|---|---|
| Contributors | Issues | Stars | Forks |

> Since launching in 2016, Google's free OSS-Fuzz code testing service has helped get over 8800 vulnerabilities and 28,000 bugs fixed across 850 projects.

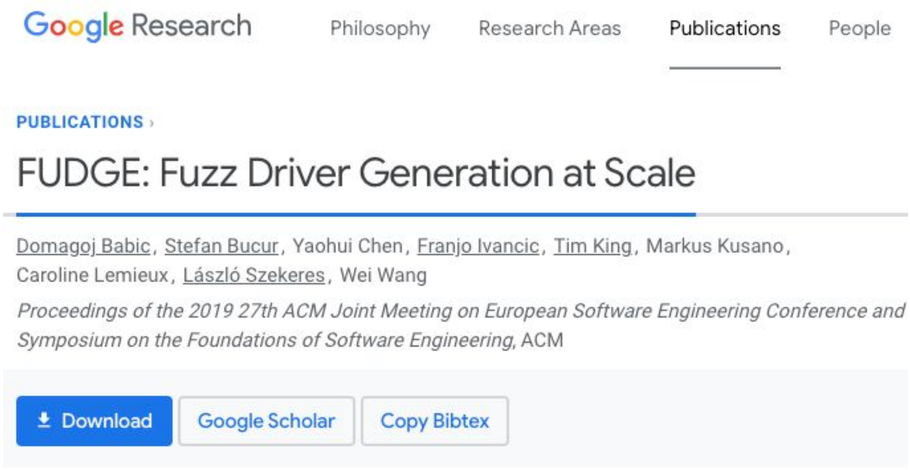
[Google Security Blog: Taking the next step: OSS-Fuzz in 2023](), Feb. 1, 2023



google/**fuzzing**

Tutorials, examples, discussions, research proposals, and other resources related to fuzzing

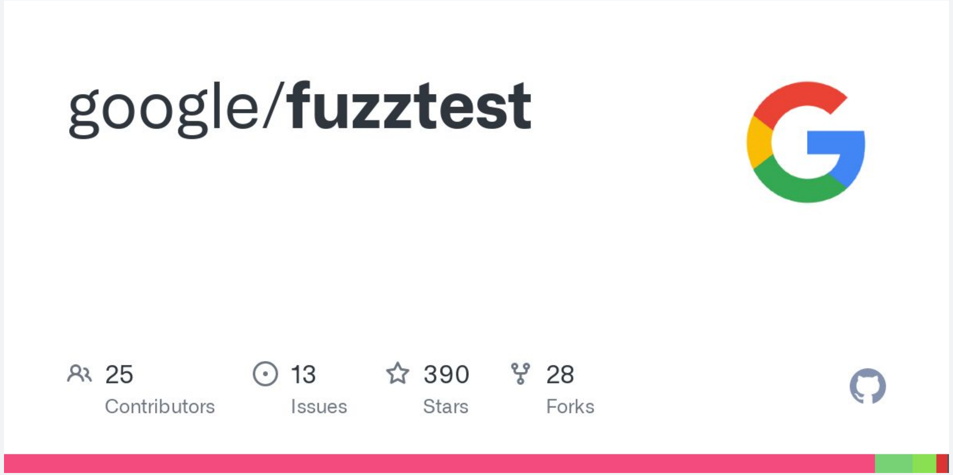
31 Contributors · 13 Issues · 3k Stars · 413 Forks



FUDGE: Fuzz Driver Generation at Scale

Domagoj Babic, Stefan Bucur, Yaohui Chen, Franjo Ivancic, Tim King, Markus Kusano, Caroline Lemieux, László Szekeres, Wei Wang

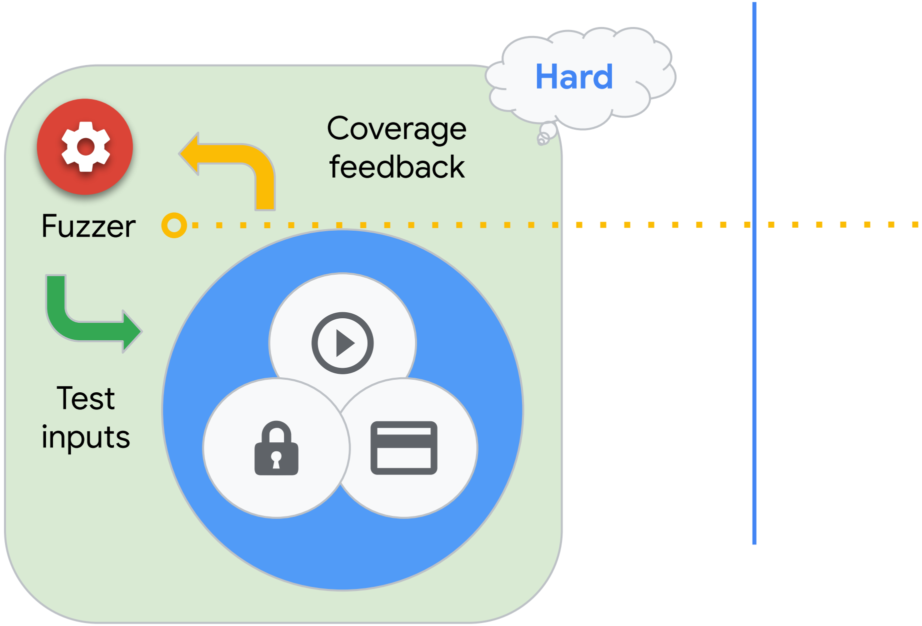
*Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ACM



google/**fuzztest**

| 25 | 13 | 390 | 28 |
|---|---|---|---|
| Contributors | Issues | Stars | Forks |

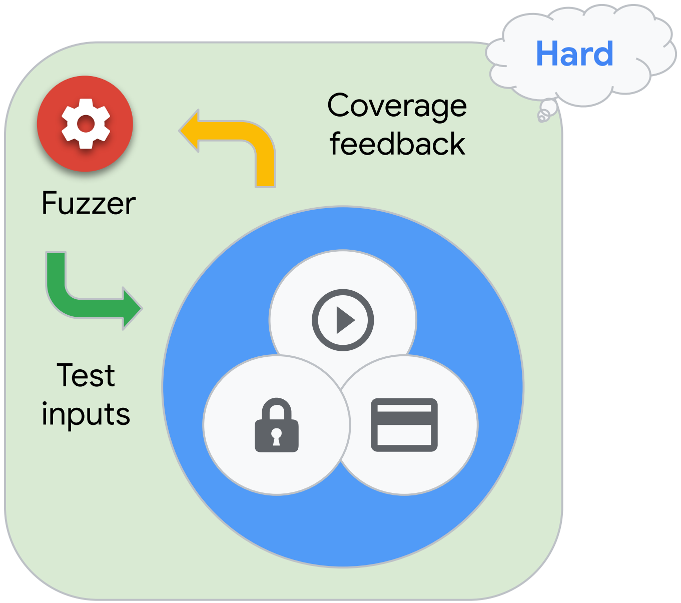
# Fuzzing Monolithic Service



Video Streaming Service

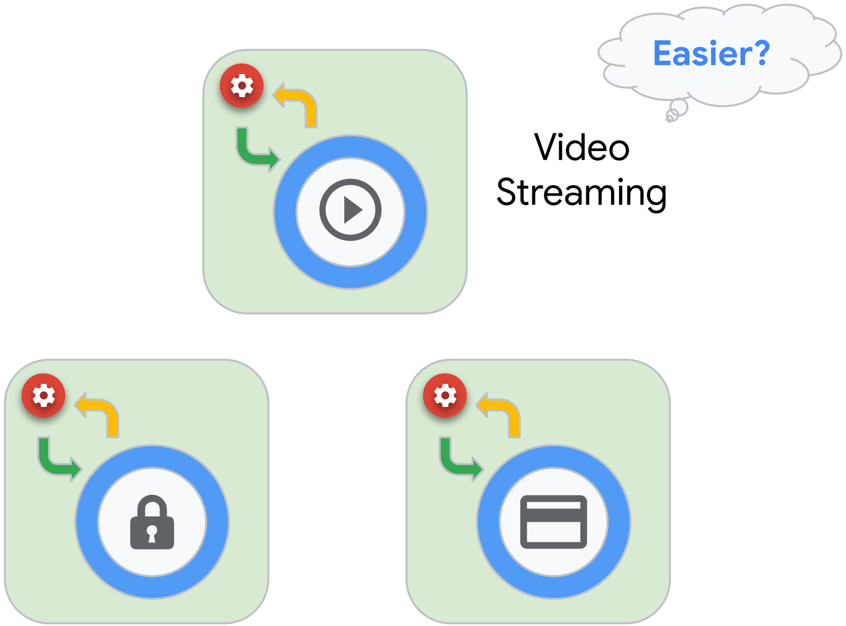
Coverage-guided in-process fuzzing

## Fuzzing Monolithic Service



Video Streaming Service

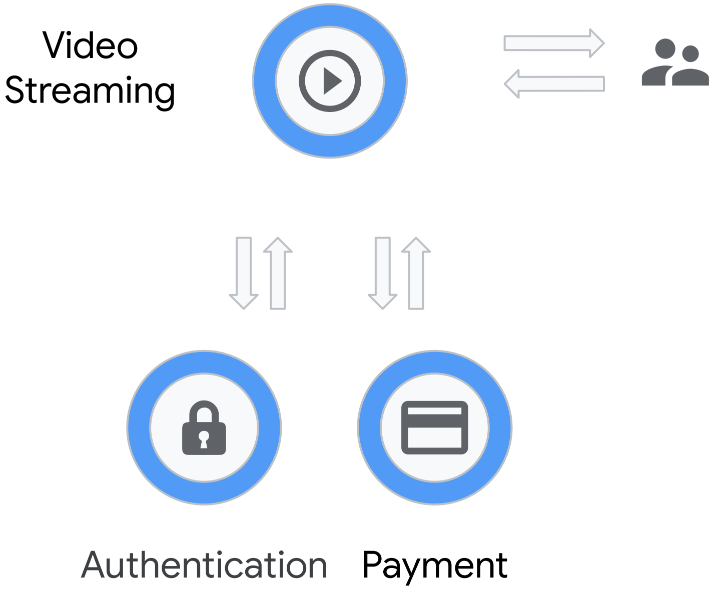
## Fuzzing Microservice



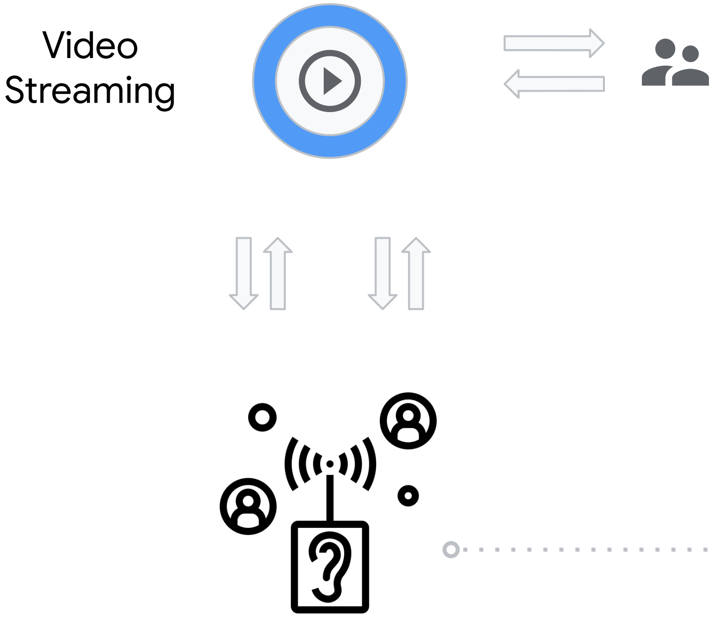
Authentication

Payment

# Fuzzing Microservice

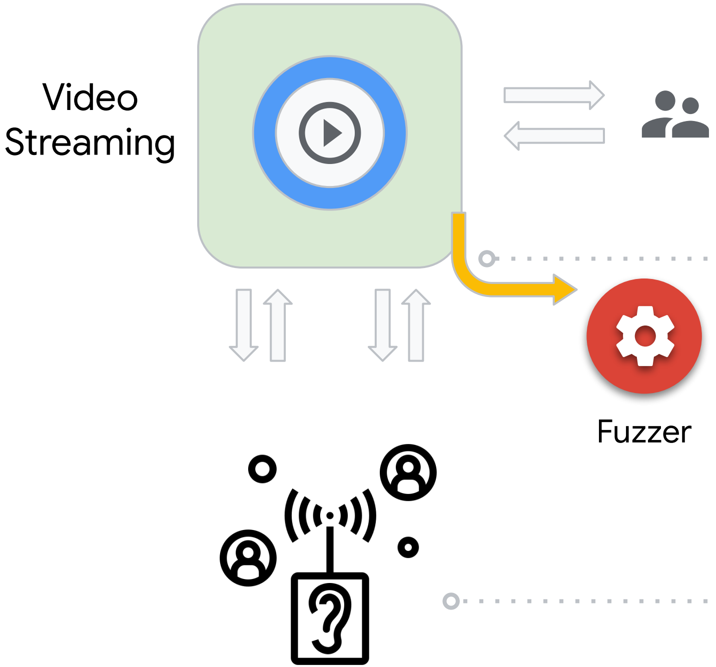


# Fuzzing Microservice



Intercept and mock out communication to backend services.

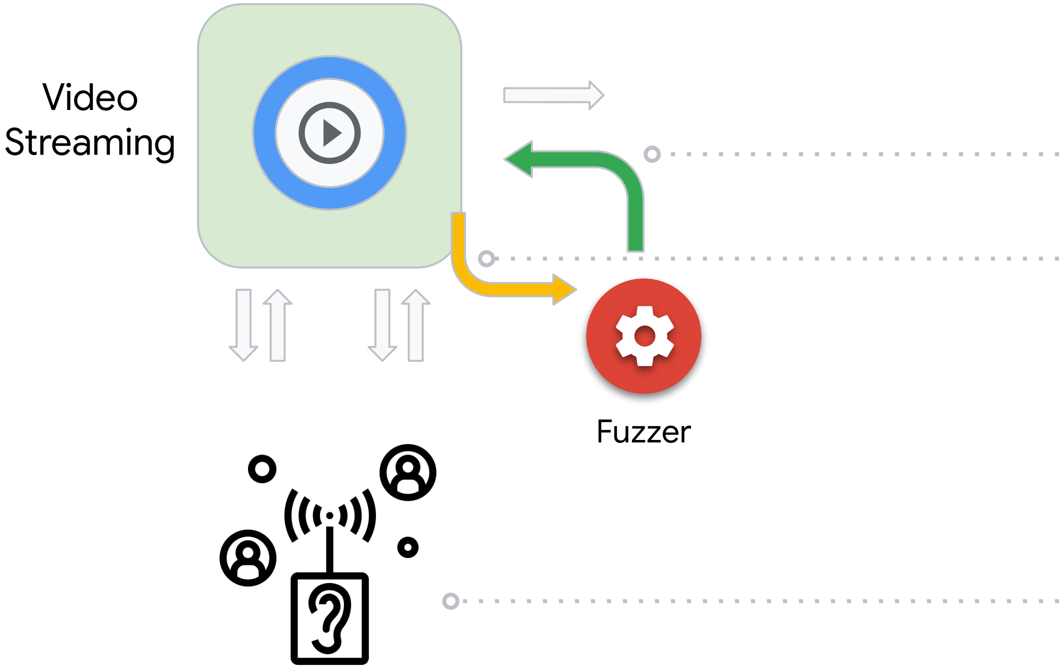
# Fuzzing Microservice



Instrument the service to get coverage feedback for the fuzzer.

Intercept and mock out communication to backend services.

# Fuzzing Microservice

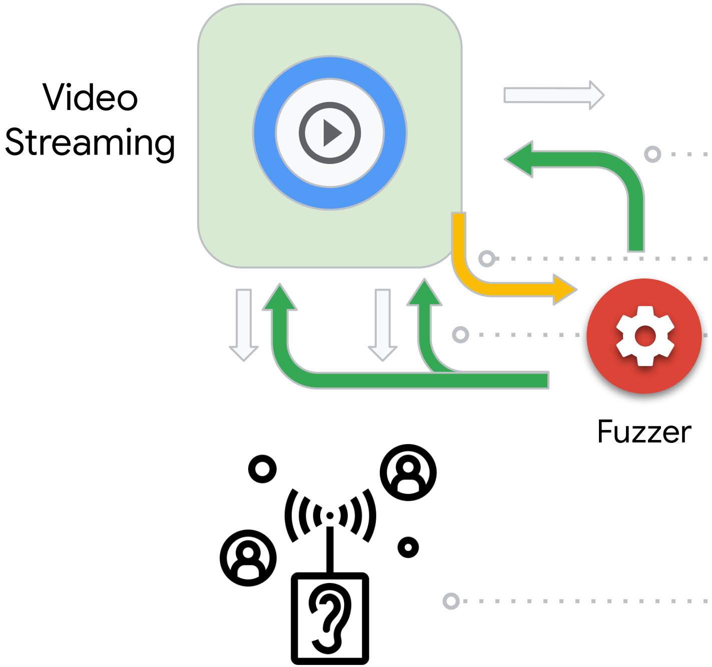


Mutate requests from users.

Instrument the service to get coverage feedback for the fuzzer.

Intercept and mock out communication to backend services.

# Fuzzing Microservice



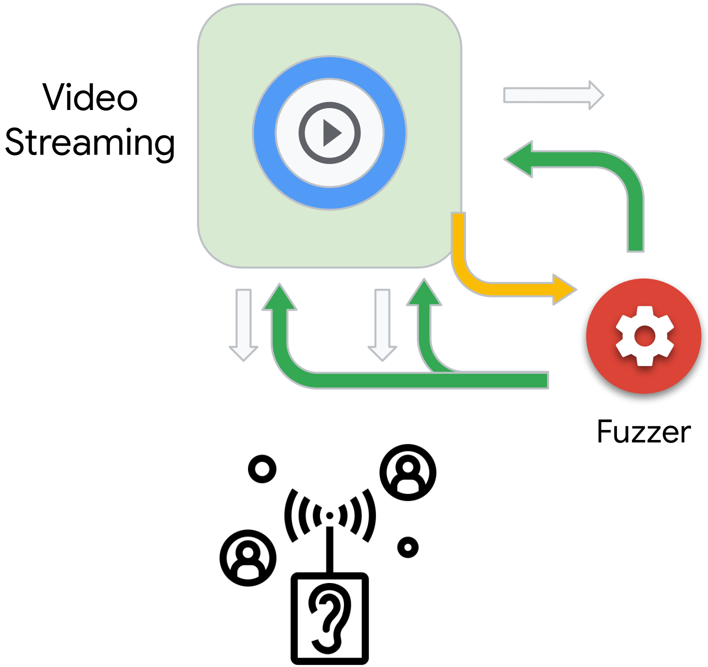
Mutate requests from users.

Instrument the service to get coverage feedback for the fuzzer.

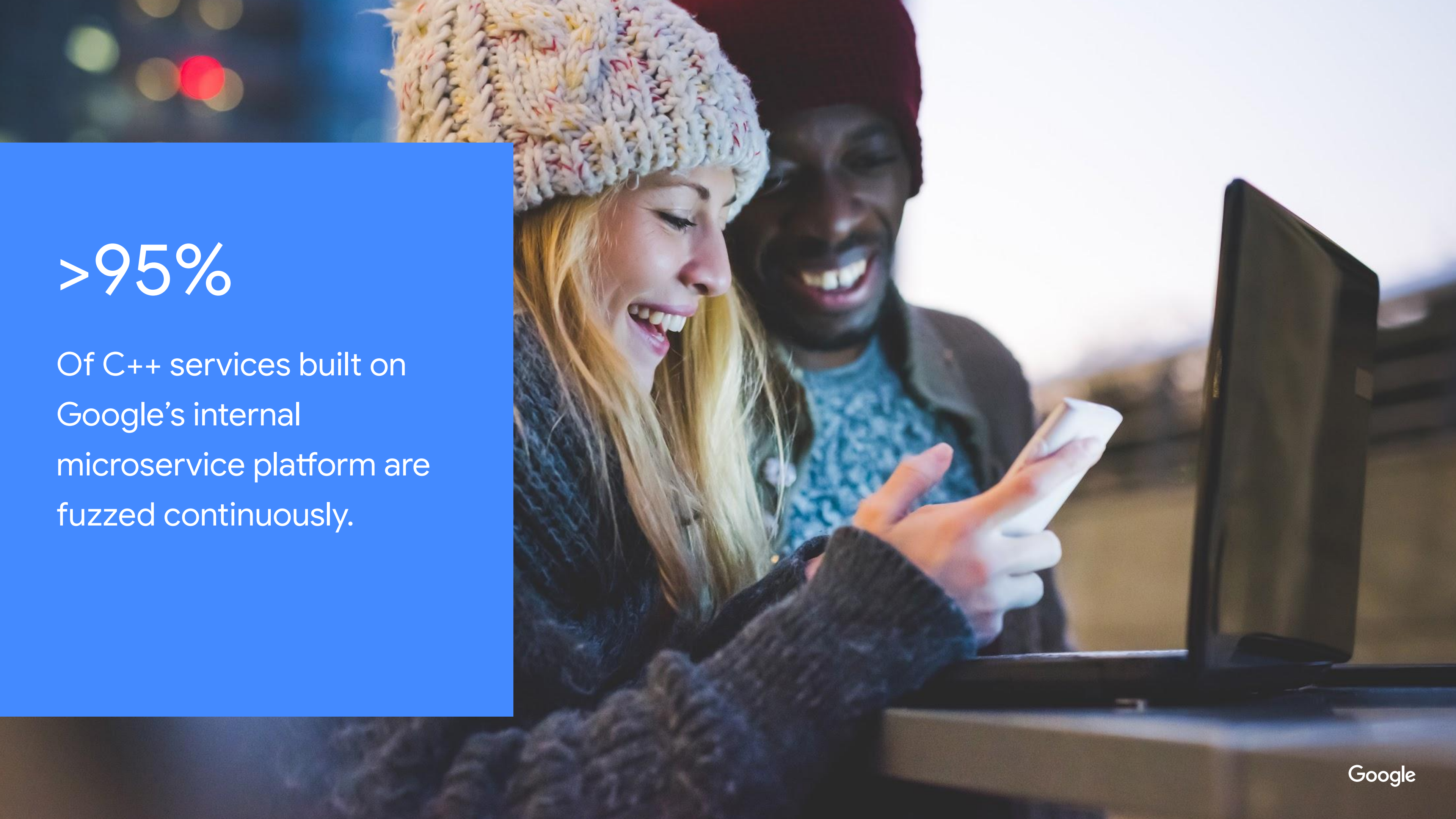
Mutate responses from backend services.

Intercept and mock out communication to backend services.

# Fuzzing Microservice



| | |
|---|---|
| Zero-Config | No human intervention is required. |
| Hermetic | No side-effect to backend services. |
| Efficient | Comprehensive testing coverage. |



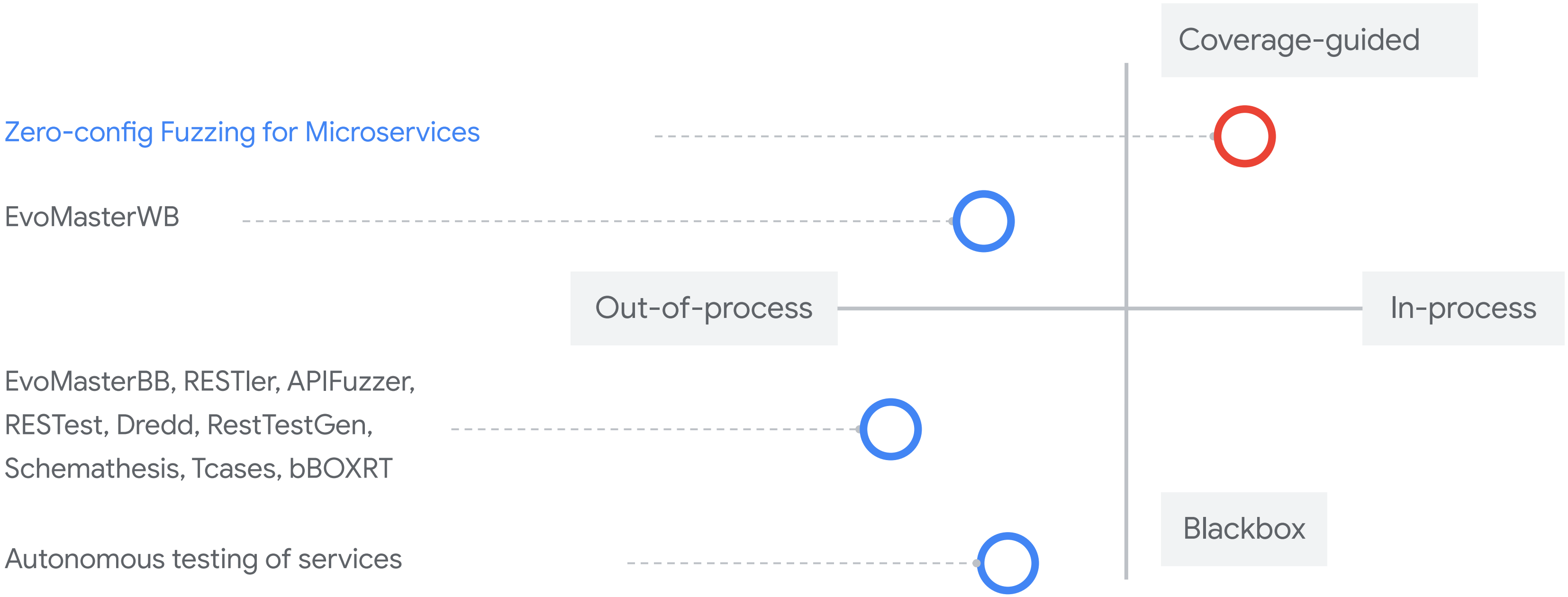
# >95%

Of C++ services built on Google’s internal microservice platform are fuzzed continuously.

# Thousands

Of errors in real-world microservice applications have been reported and got fixed.

# Related Work: Fuzzing Services



Coverage-guided

Zero-config Fuzzing for Microservices

EvoMasterWB

Out-of-process

In-process

EvoMasterBB, RESTler, APIFuzzer, RESTest, Dredd, RestTestGen, Schemathesis, Tcases, bBOXRT

Blackbox

Autonomous testing of services

1. M. Kim, Q. Xin, S. Sinha, and A. Orso, “Automated test generation for REST APIs: no time to rest yet,” in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*. ACM, jul 2022
2. P.I Marinescu, “Autonomous testing of services at scale.” in Engineering at Meta, 2021.

# Lessons Learned: Bugs & Developers

1 Developers fix bugs found by auto-generated tests just as fast as bugs found by human written tests.

2 Memory bugs caused by race conditions can be challenging to reproduce.

3 There are developers complain that fuzzing generates unrealistic inputs that won't happen in real life.

4 Overall, we trust developers on deciding which bugs to prioritize fixing.

# Zero-Config Fuzzing for Microservices

ASE'23

Wei Wang

Andrei Benea

Franjo Ivančić