Resiliency Testing with Toxiproxy

Jake Pittis

usenix LISA17

October 29–November 3, 2017 | San Francisco, CA
www.usenix.org/lisa17 #lisa17

Resiliency Testing with Toxiproxy

Jake Pittis

usenix LISA17


shopify

Reasoning about failure is hard.

Too many kinds of failures.

Large complex systems.

Constantly changing.

Our intuition is often wrong.

Incident

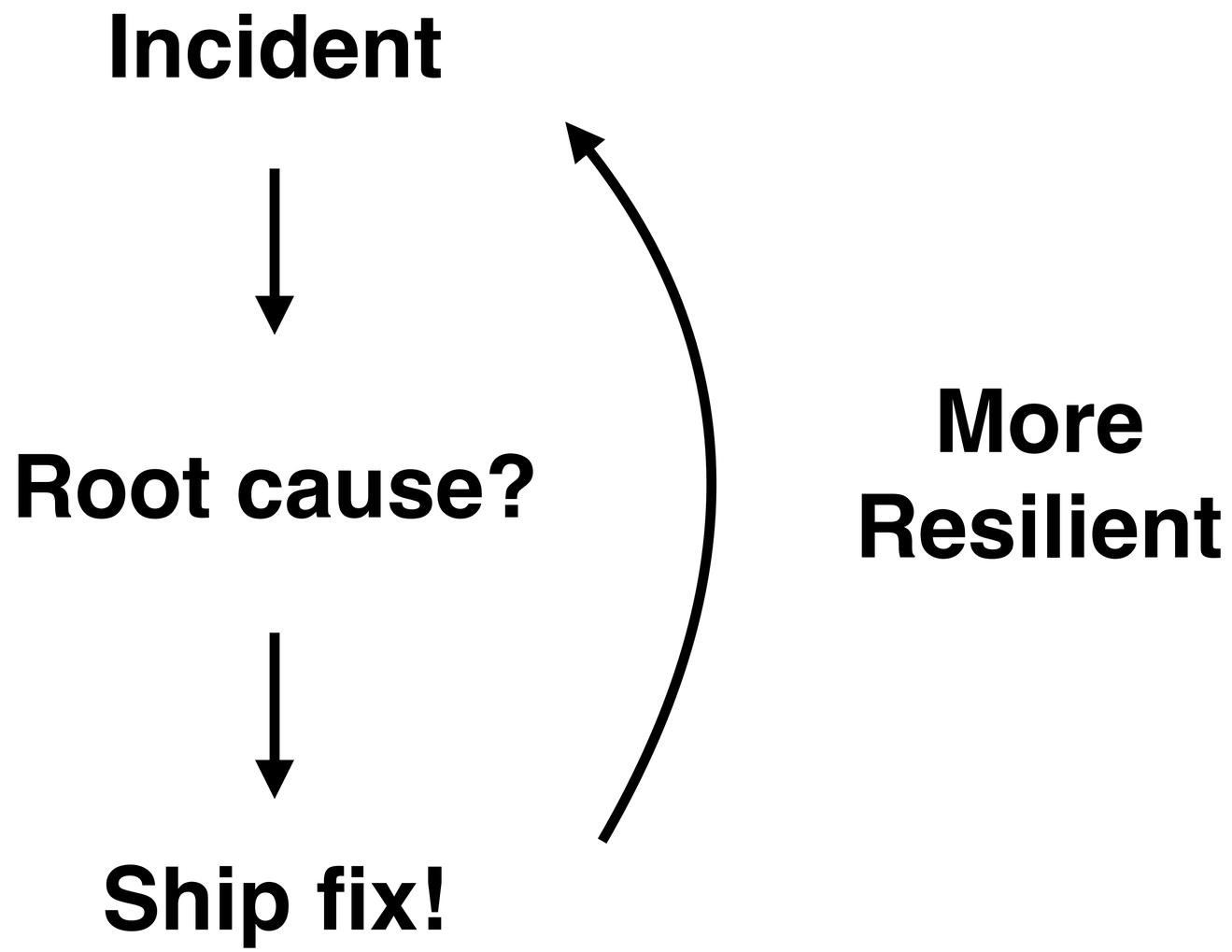
A natural failure in production.

A database writer goes down.

Incident

Root cause?

Ship fix!


Incident
Root cause?
Ship fix!
More
Resilient

Gameday

Artificially exercising a known failure scenario in production.

Flash sales.

Gameday

What broke?

Ship fix!

Incident

Root cause?

Ship fix!

Gameday

What broke?

Ship fix!

	Authenticity
Production Incident	100%
Gameday	Mostly

	Authenticity	Production Impact
Production Incident	100%	High
Gameday	Mostly	Controlled

	Authenticity	Production Impact	Automation
Production Incident	100%	High	Mixed
Gameday	Mostly	Controlled	Manual

	Authenticity	Production Impact	Automation	Accessibility
Production Incident	100%	High	Mixed	SRE
Gameday	Mostly	Controlled	Manual	SRE

Resiliency is a product concern.

Automatically Prevent Regression

Automatically Prevent Regression

Accessible to All Developers

Automatically Prevent Regression

Accessible to All Developers

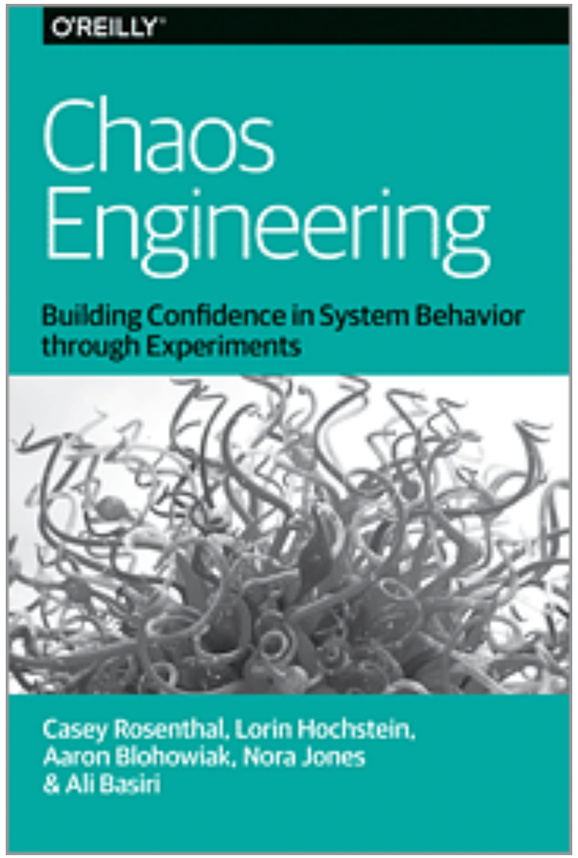
Lower Customer Impact

Automatically Prevent Regression

Accessible to All Developers

Lower Customer Impact

Maintain Authenticity

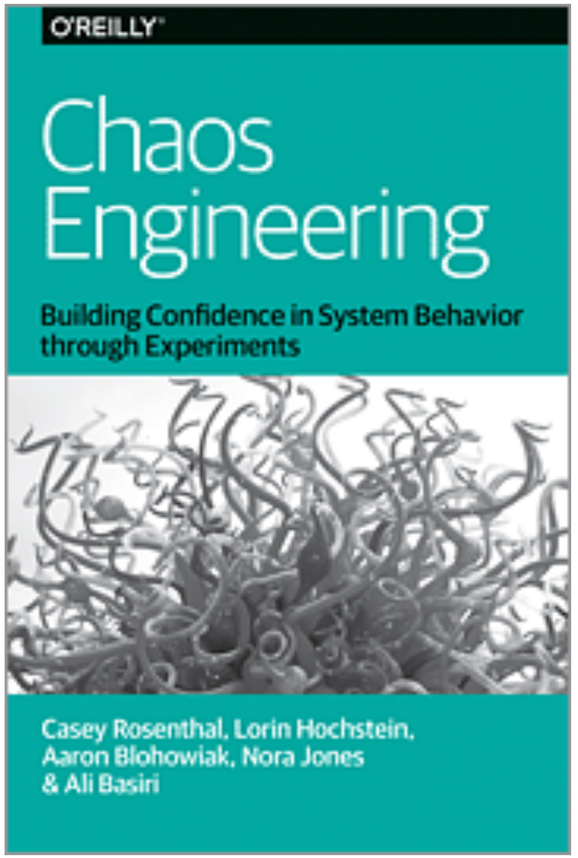

O'REILLY
Chaos Engineering
Building Confidence in System Behavior through Experiments
Casey Rosenthal, Lorin Hochstein, Aaron Blohowiak, Nora Jones & Ali Basiri

Chaos Engineering

Running experiments in production to cause and fix unknown failure scenarios.

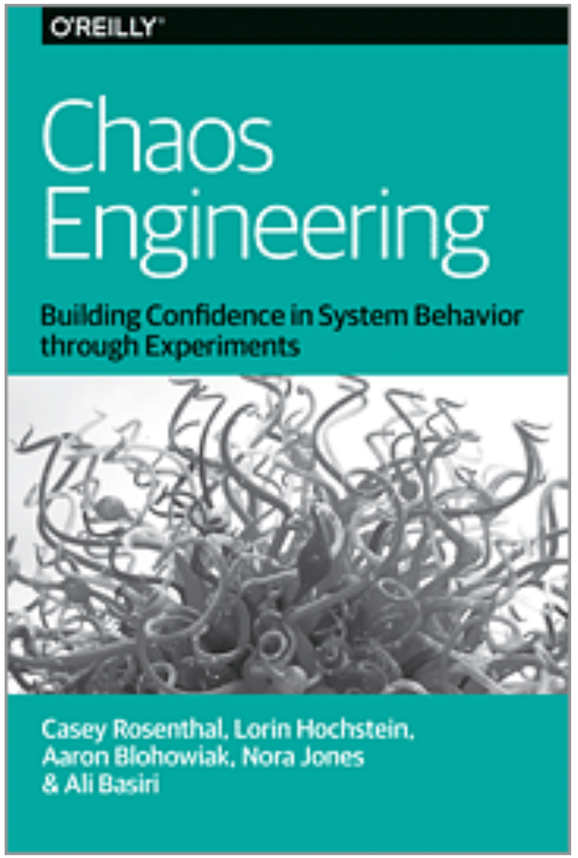
“Automate Experiments to Run Continuously”

Automatically Prevent Regression



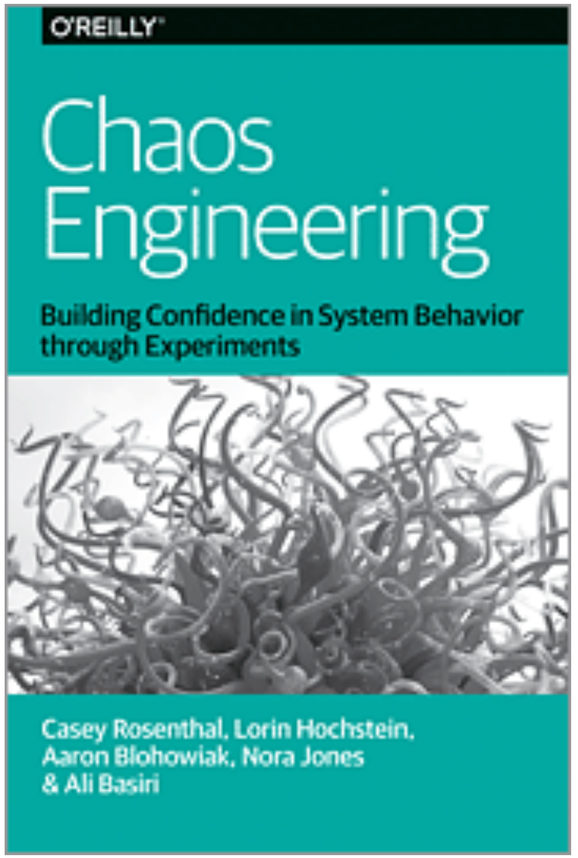
“Minimize Blast Radius”

Lower Customer Impact

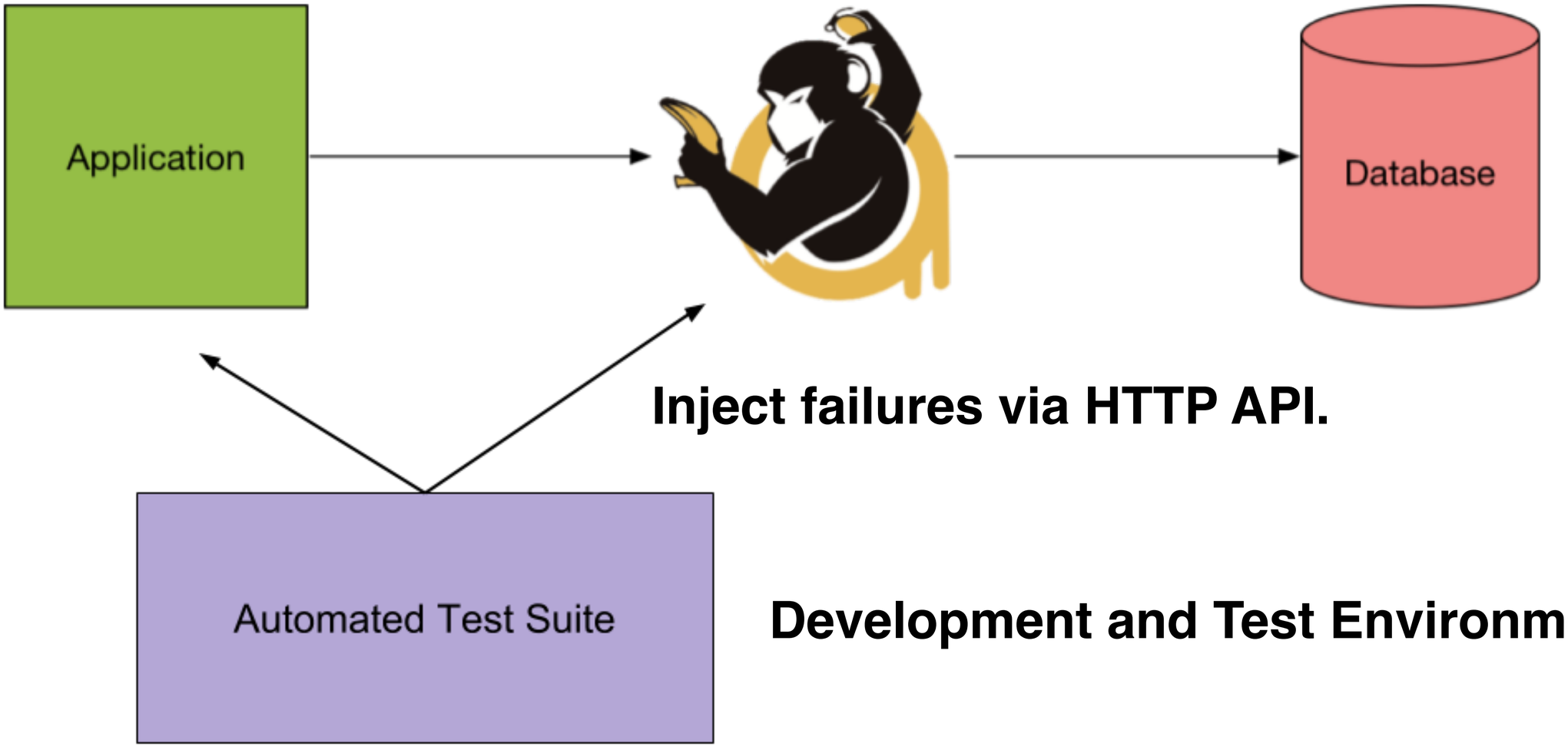


“Run Experiments in Production”

Maintain Authenticity

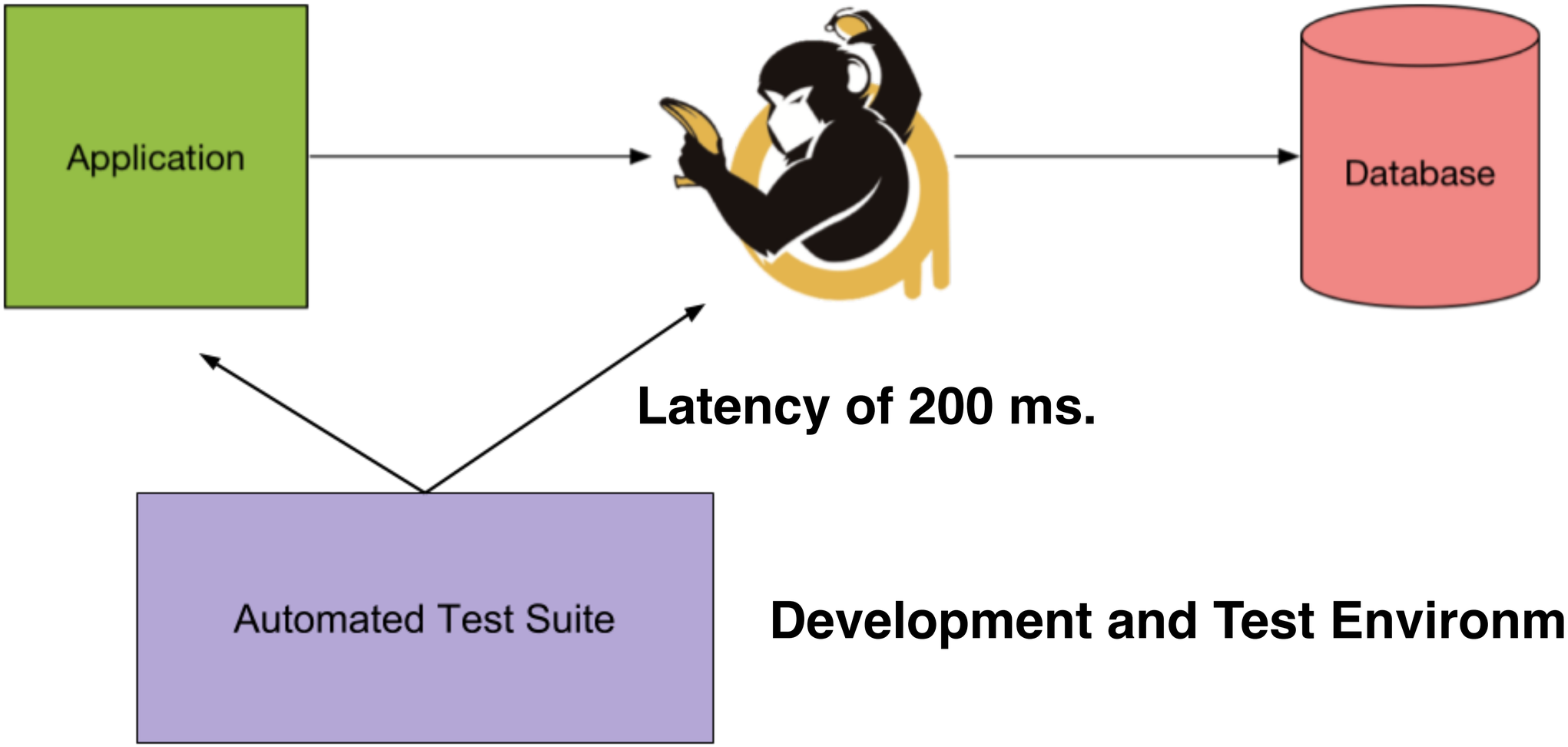


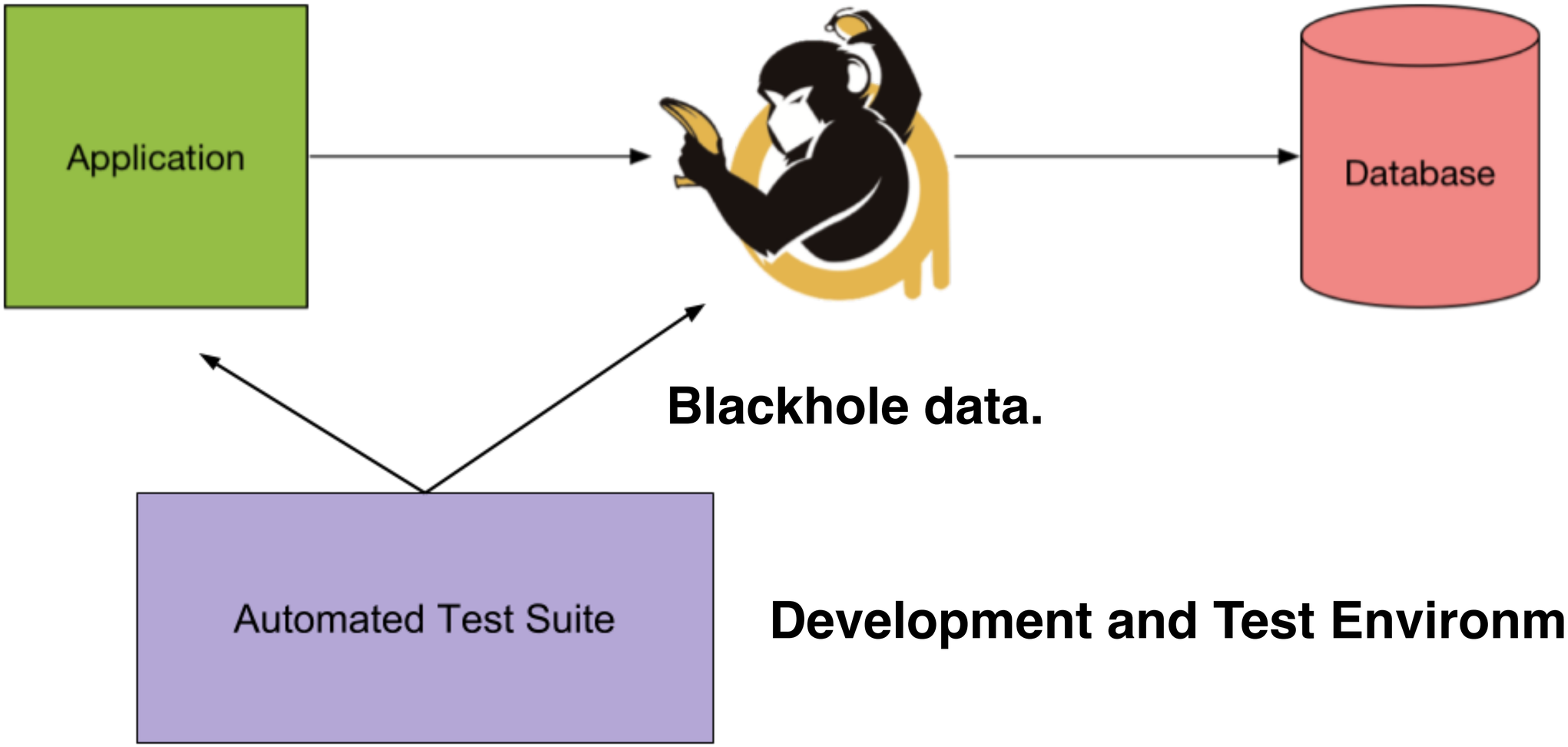
Toxiproxy



Inject failures via HTTP API.

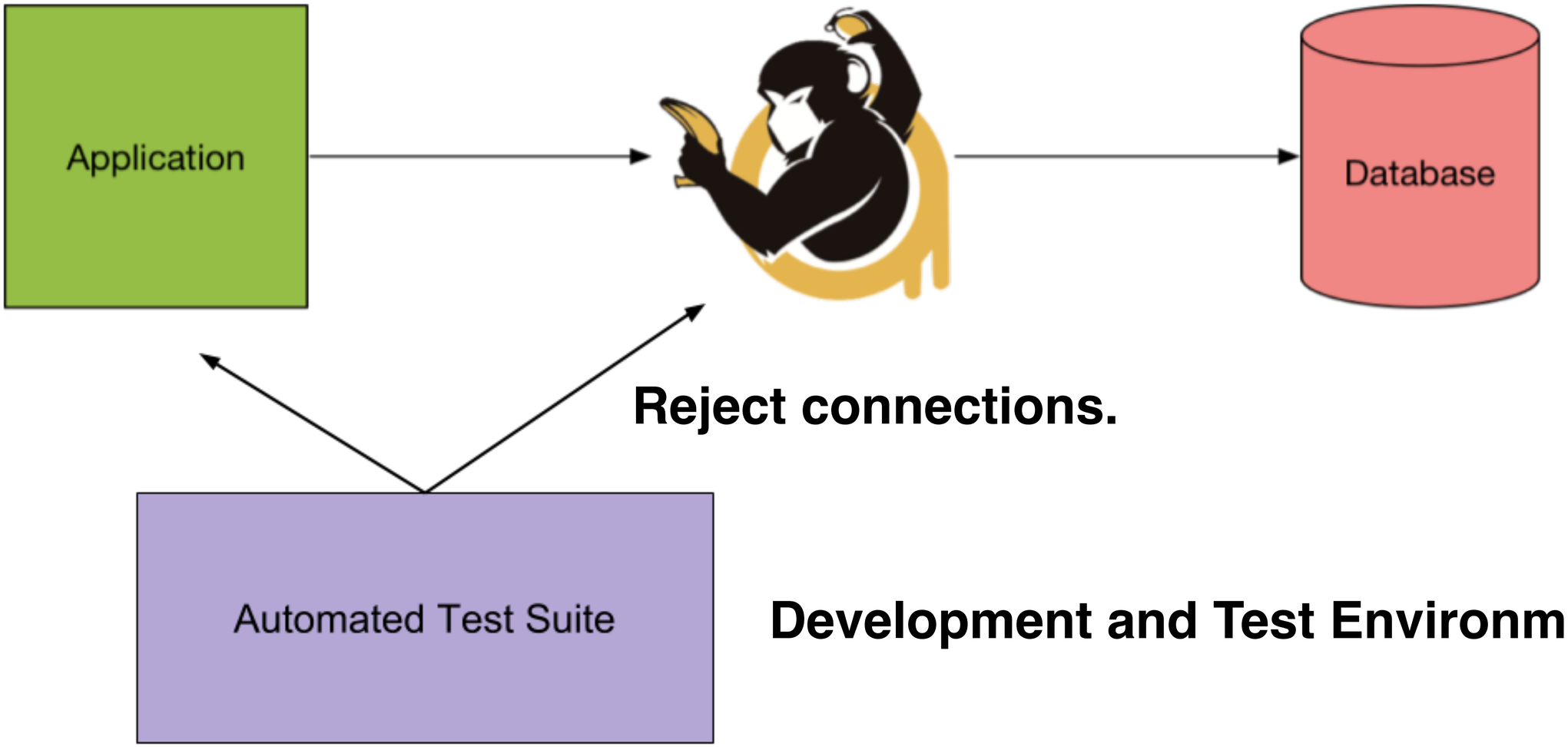
Development and Test Environment


Application
Database
Latency of 200 ms.
Automated Test Suite
Development and Test Environment



Blackhole data.

Development and Test Environment


Application
Database
Reject connections.
Automated Test Suite
Development and Test Environment

Reactive Testing

Incident

Root cause?

Ship fix!

Incident

Root cause?

Ship fix!

Does it work?

Regression?

A flash sale takes down redis while a deploy is going out.

A flash sale takes down redis while a deploy is going out.

Application boot relies on redis!?

A flash sale takes down redis while a deploy is going out.

Application boot relies on redis!?

Remove the dependency!

A flash sale takes down redis while a deploy is going out.

Application boot relies on redis!?

Remove the dependency!

Write a Toxiproxy test!

```
Toxiproxy[/.*/].down do
  assert application_can_boot
end
```

Accessible to All Developers ✅

Accessible to All Developers ✅

No Customer Impact ✅

Accessible to All Developers ✅

No Customer Impact ✅

Maintain Authenticity ✅

Accessible to All Developers ✅

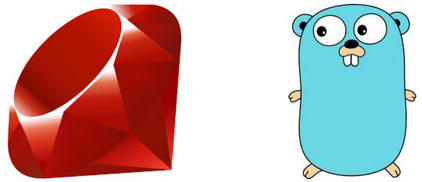
No Customer Impact ✅

Maintain Authenticity ✅

Automatically Prevent Regression ✅

A few hundred Toxiproxy Tests

All you need is a thin client library.



Java, Node, Python, PHP or write your own!

I used it reactively just last week.

Proactive Testing

Resiliency Matrix

Sections

Services

	Storefront	Checkout	Signups
MySQL	?	?	?
Redis	?	?	?
Memcached	?	?	?

Resiliency Matrix

Sections

Services

	Storefront	Checkout	Signups
MySQL	Degraded	Down	Down
Redis	Up	Down	Up
Memcached	Degraded	Up	Up

Resiliency Matrix

Sections

Services

	Storefront	Checkout	Signups
MySQL	Degraded	Down	Down
Redis	Up	Down	Up
Memcached	Degraded	Up	Up

```ruby
Toxiproxy['mysql_writer'].down do
  get '/'
  assert_response :ok
end
```

Incident

Root cause?

Ship fix and Toxiproxy test!

Gameday

What broke?

Ship fix and Toxiproxy test!

Create Resiliency Matrix

Test every intersection.

What’s next?

All our applications should have a resiliency matrices.

	Storefront	Checkout	Signups
MySQL	Degraded	Down	Down
Redis	Up	Down	Up
Memcached	Degraded	Up	Up

Integrate Toxiproxy into all our applications by default.

Gameday everything we can’t write Toxiproxy tests for.

Automate the gamedays.

Gamedays

	Storefront	Checkout	Signups
MySQL	Degraded	Down	Down
Redis	Up	Down	Up
Memcached	Degraded	Up	Up

Toxiproxy is open source.
(github.com/Shopify/toxiproxy)

Go read the readme for more information!

Thanks!