Chaos Engineering to Continuous Verification

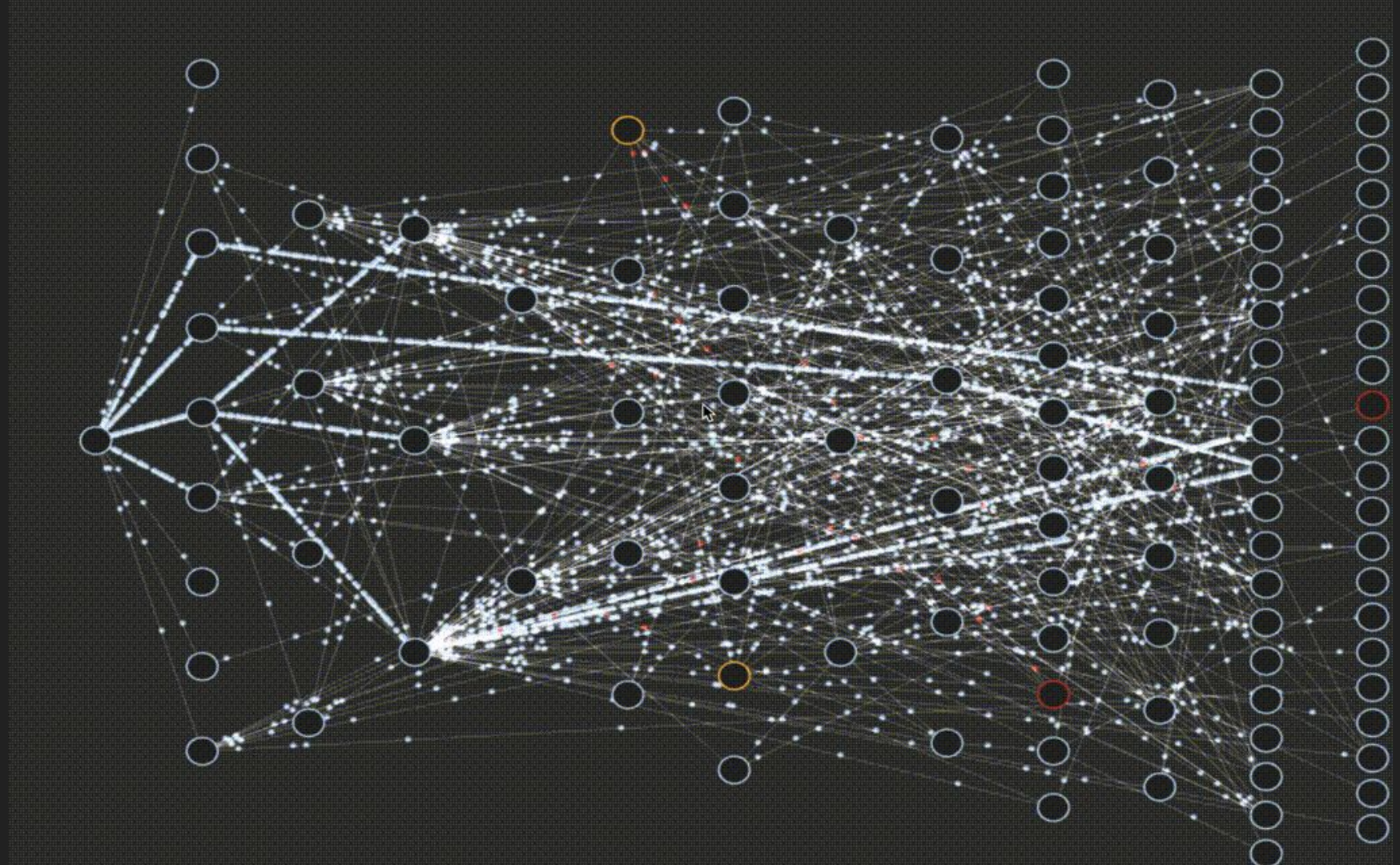
@CaseyRosenthal

CONTINUOUS VERIFICATION

@CaseyRosenthal

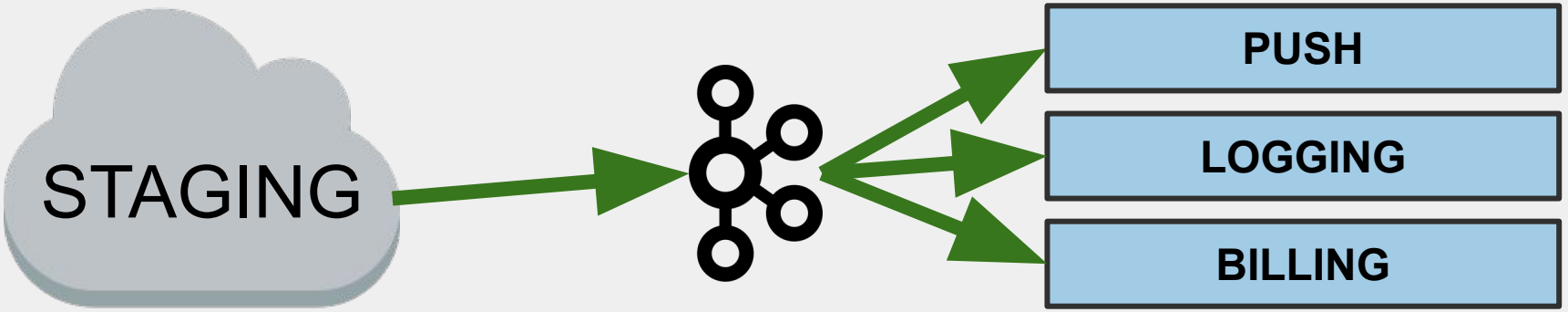

Chaos
Monkey

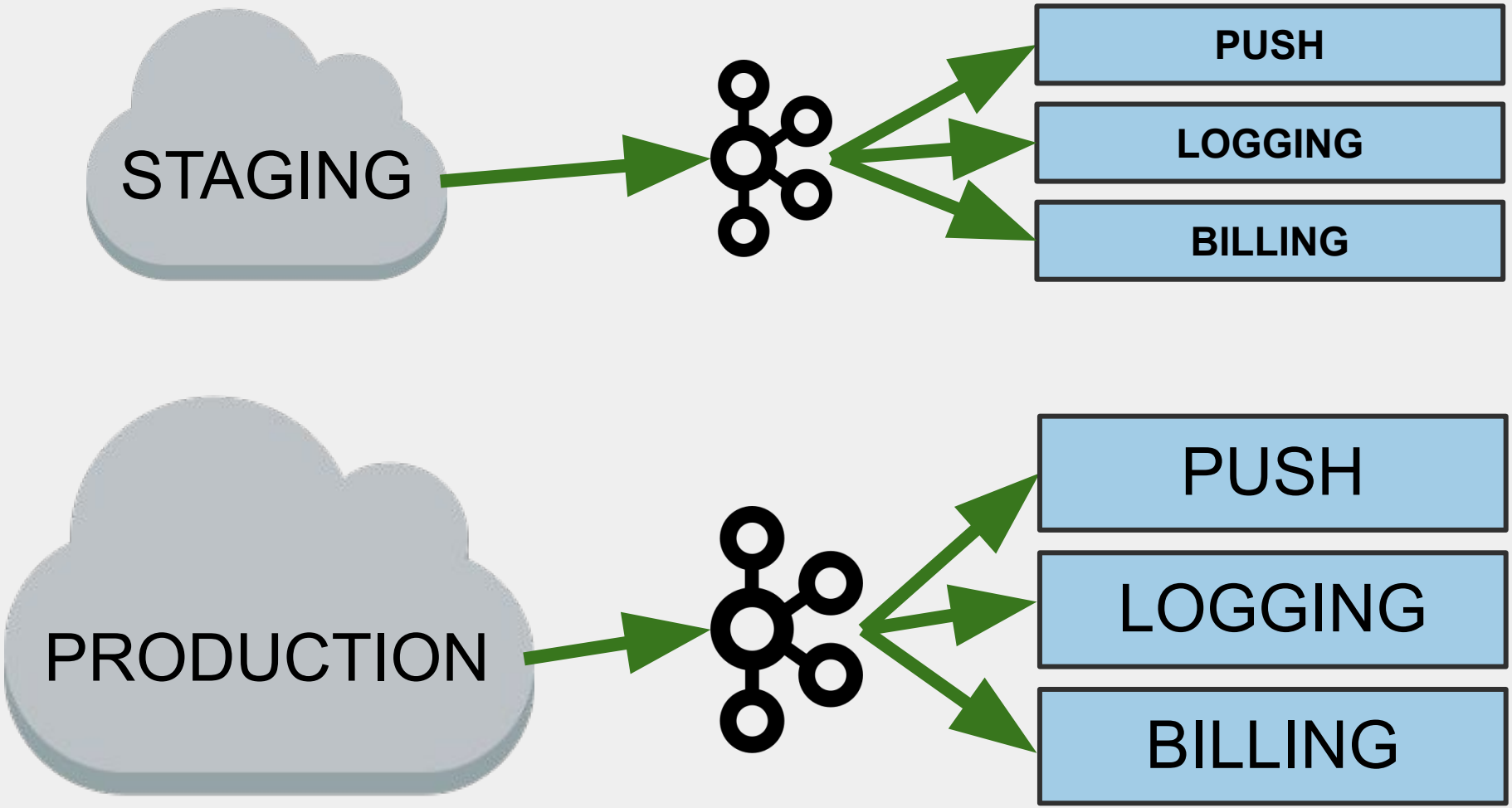

ChAP

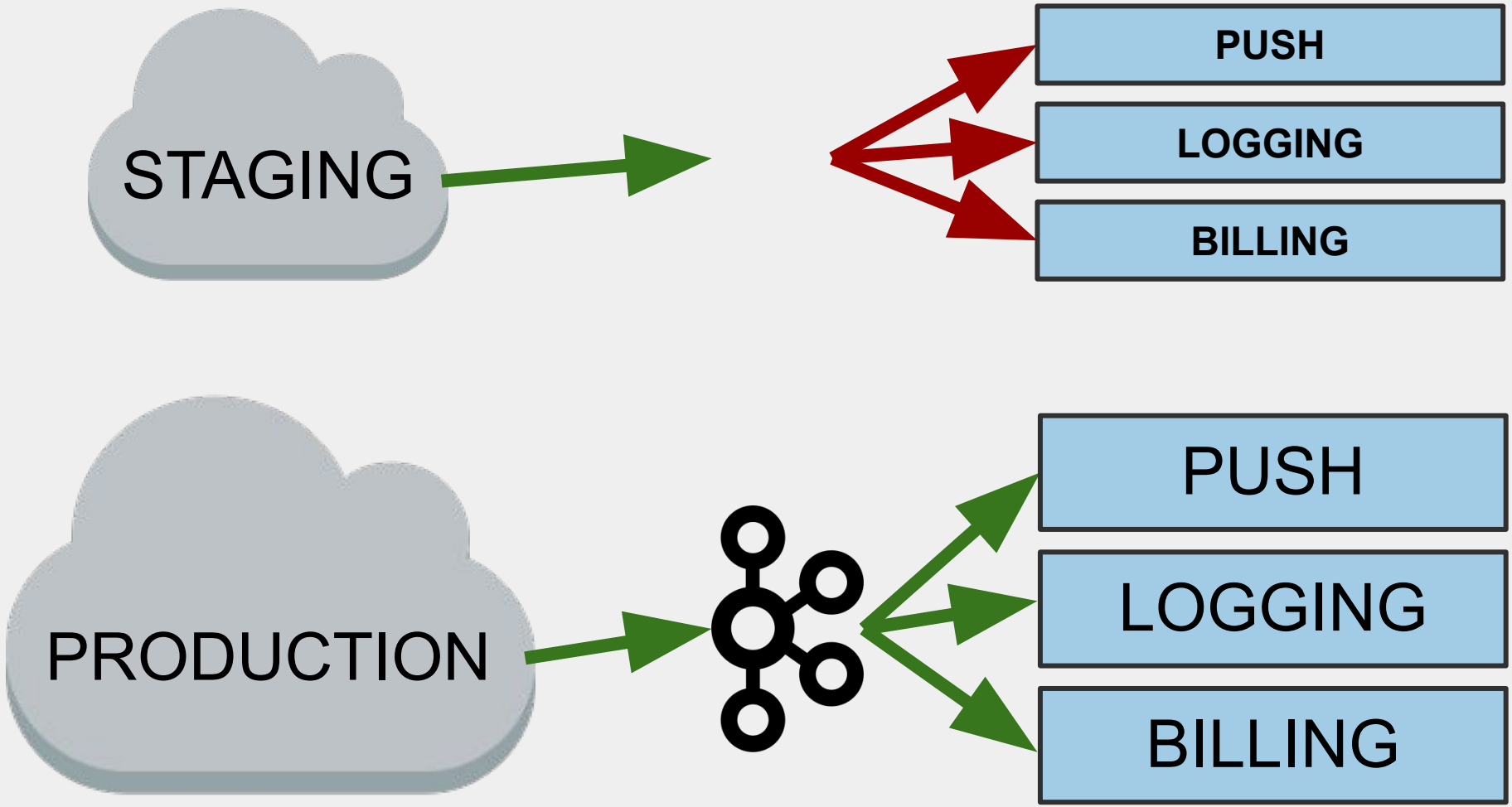


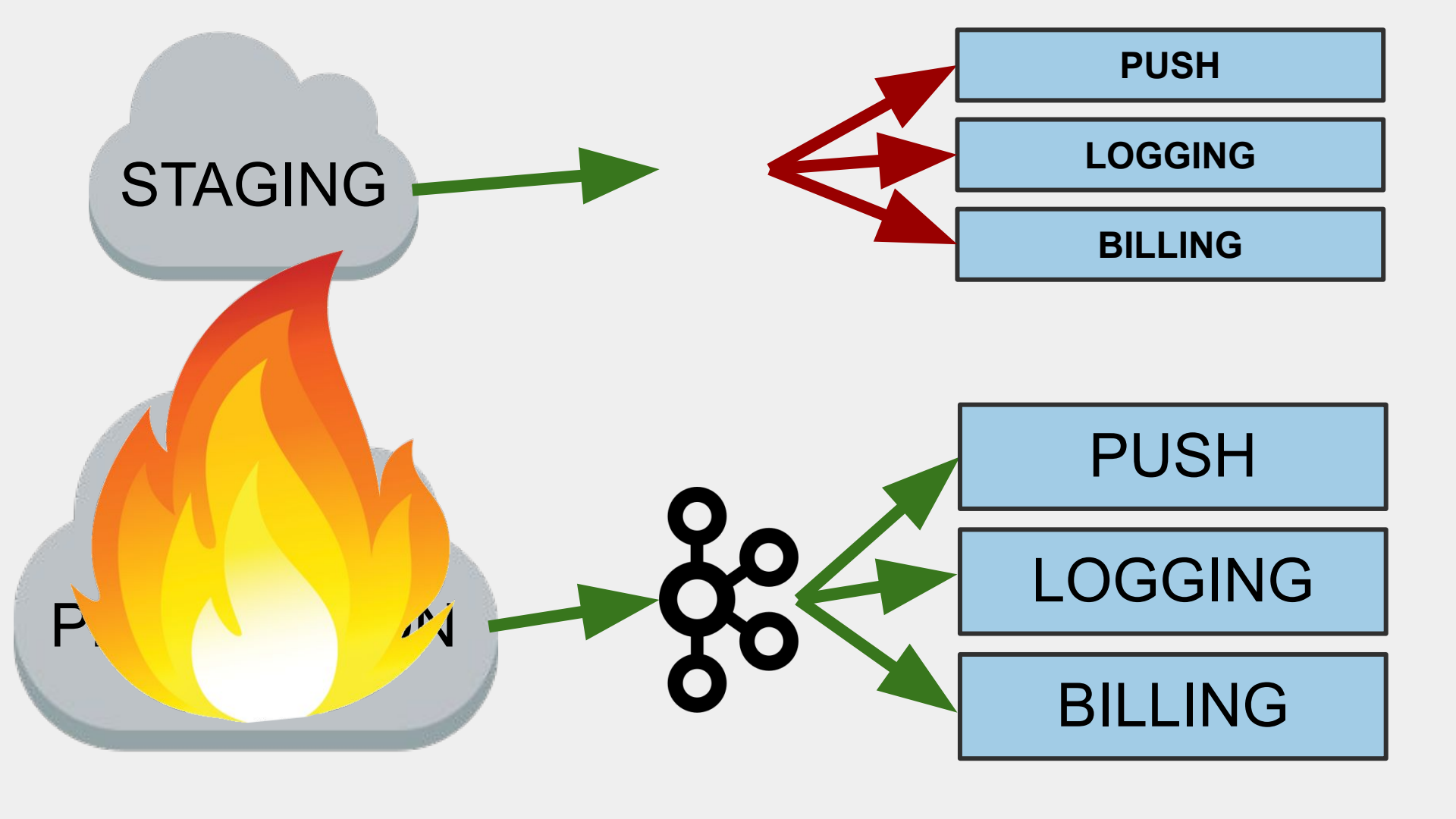
What could go wrong in a complex system?

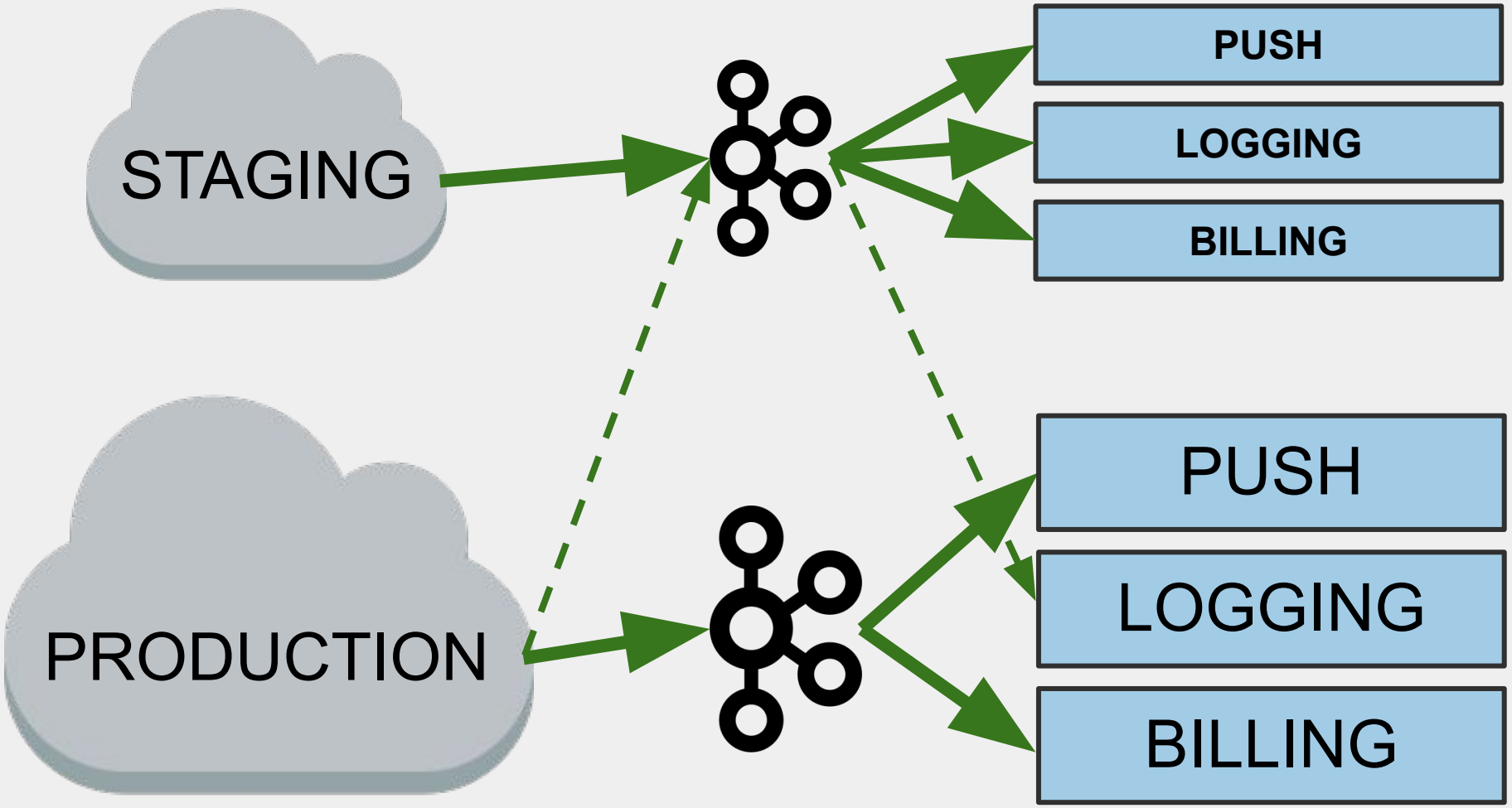
@CaseyRosenthal


STAGING
PUSH
LOGGING
BILLING

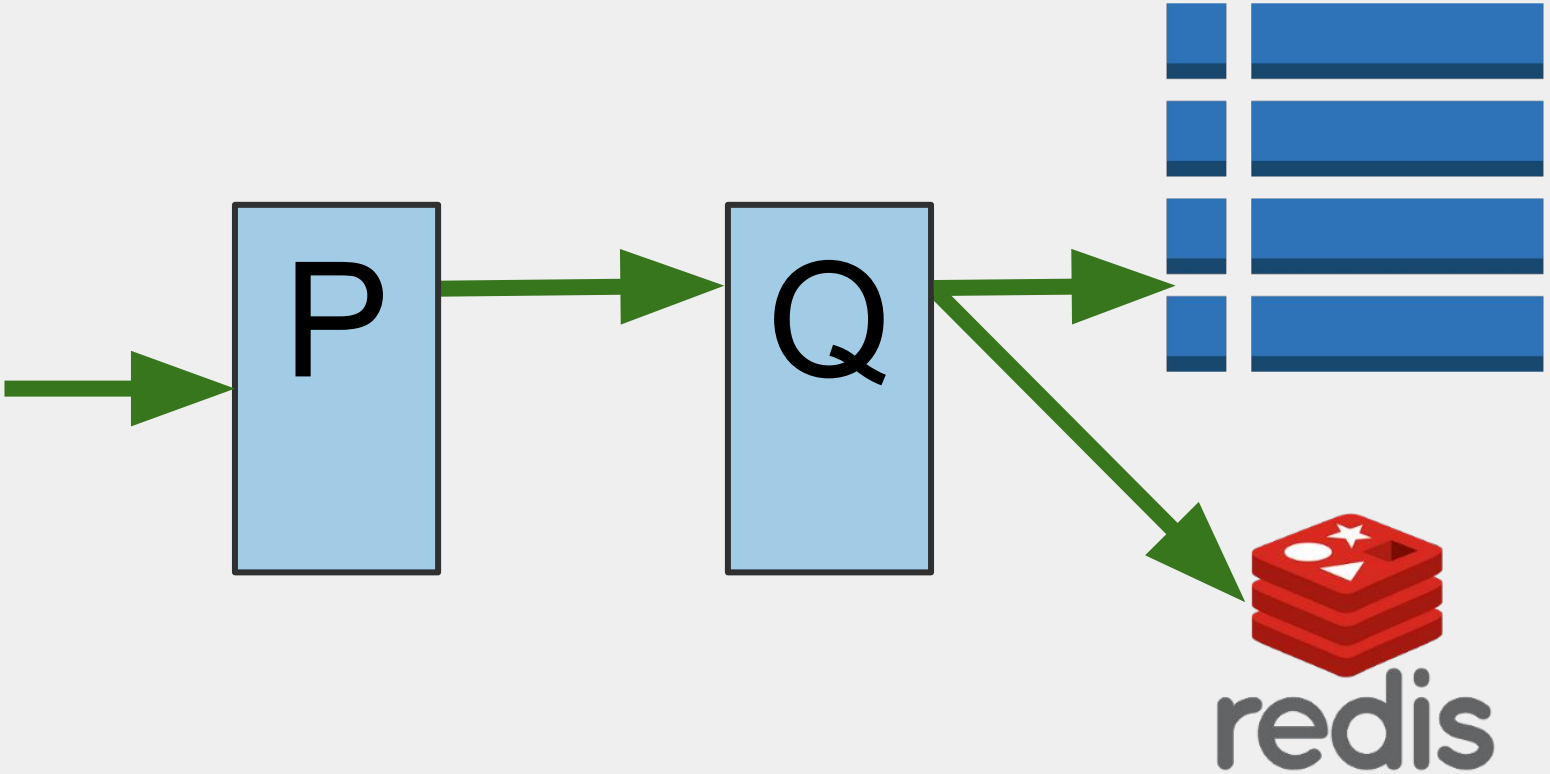

STAGING
PUSH
LOGGING
BILLING
PRODUCTION
PUSH
LOGGING
BILLING

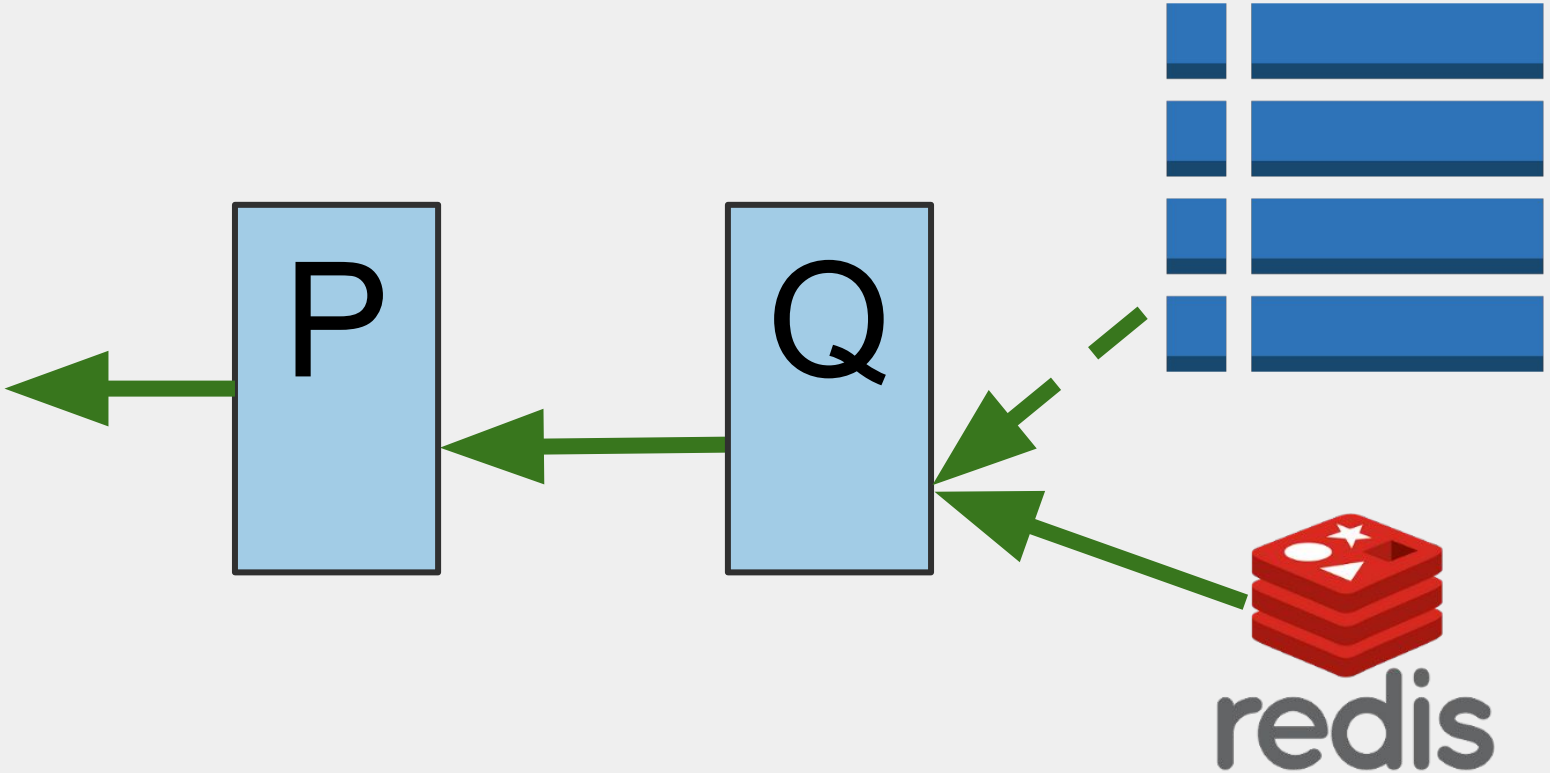

STAGING
PUSH
LOGGING
BILLING
PRODUCTION
PUSH
LOGGING
BILLING

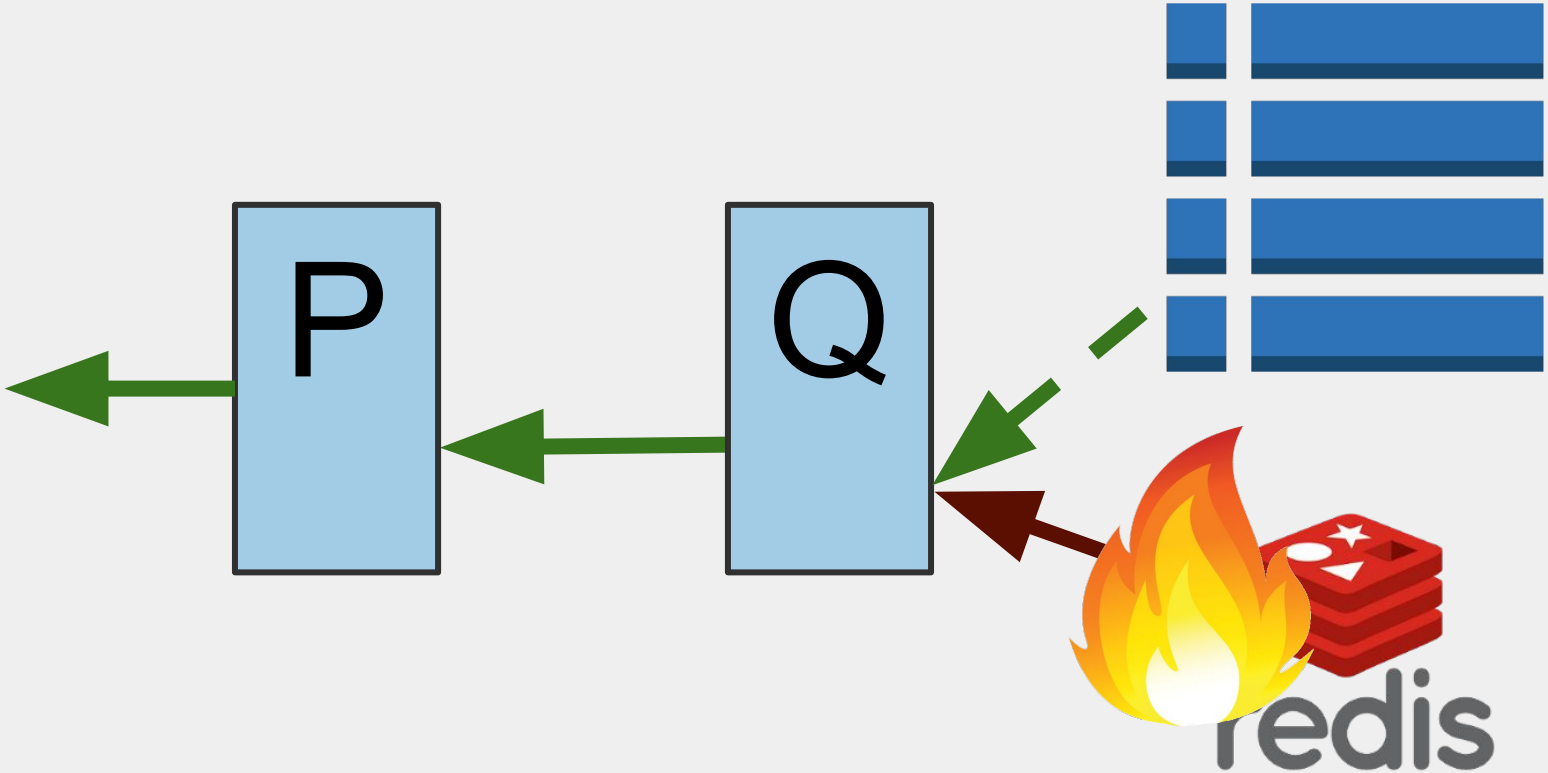

STAGING
PUSH
LOGGING
BILLING
PUSH
LOGGING
BILLING

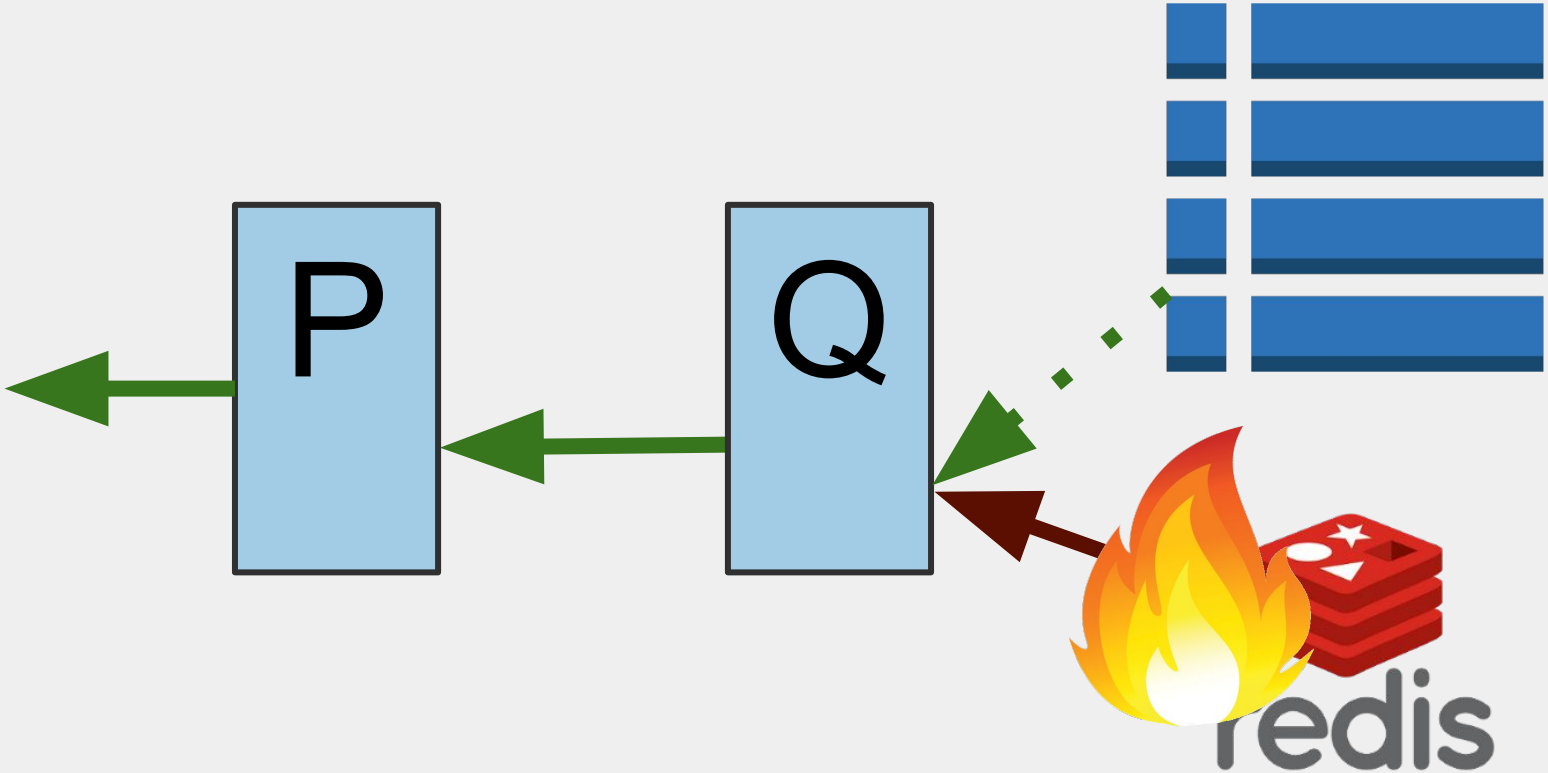

STAGING
PUSH
LOGGING
BILLING
PRODUCTION
PUSH
LOGGING
BILLING

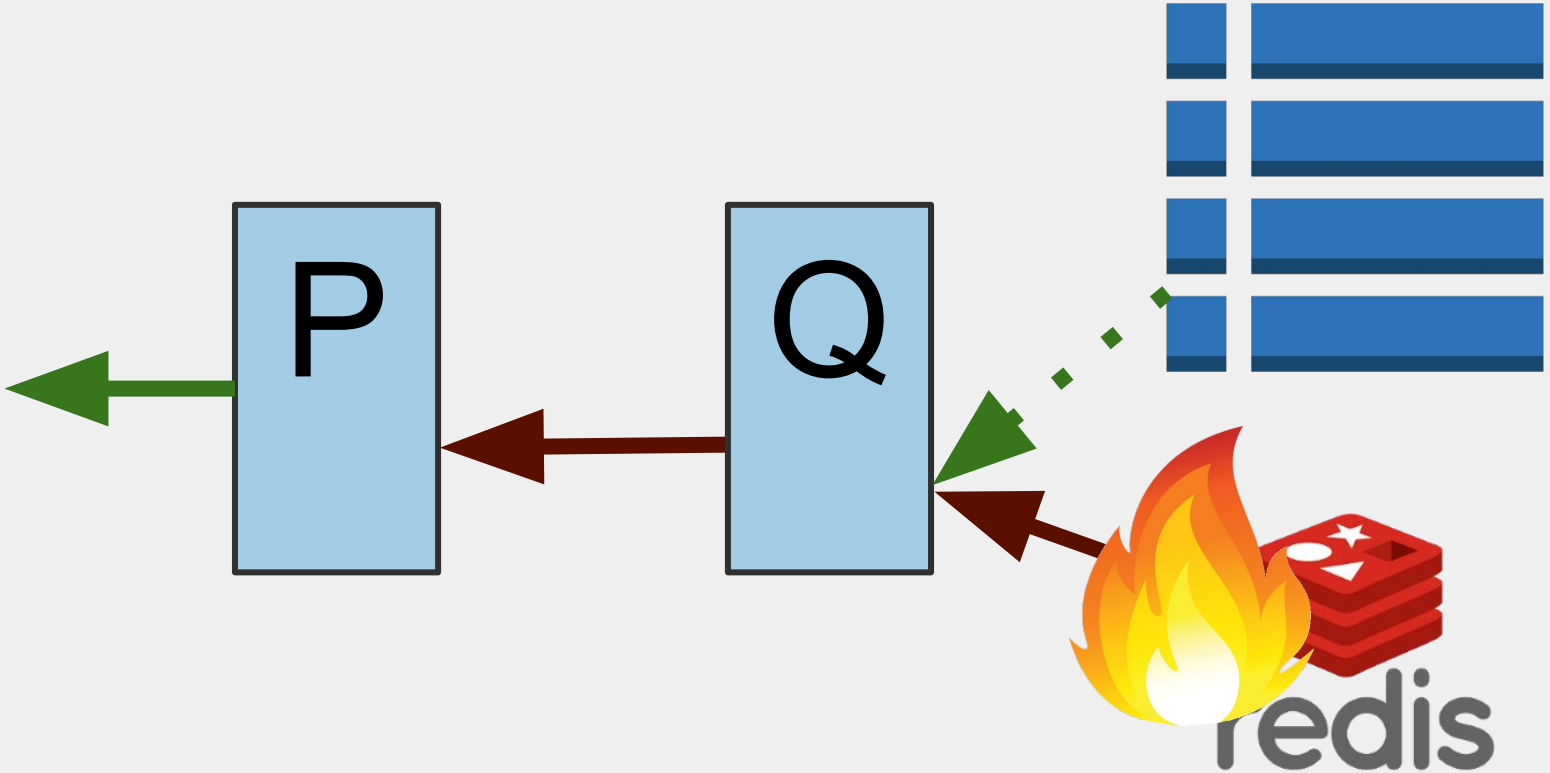
@CaseyRosenthal

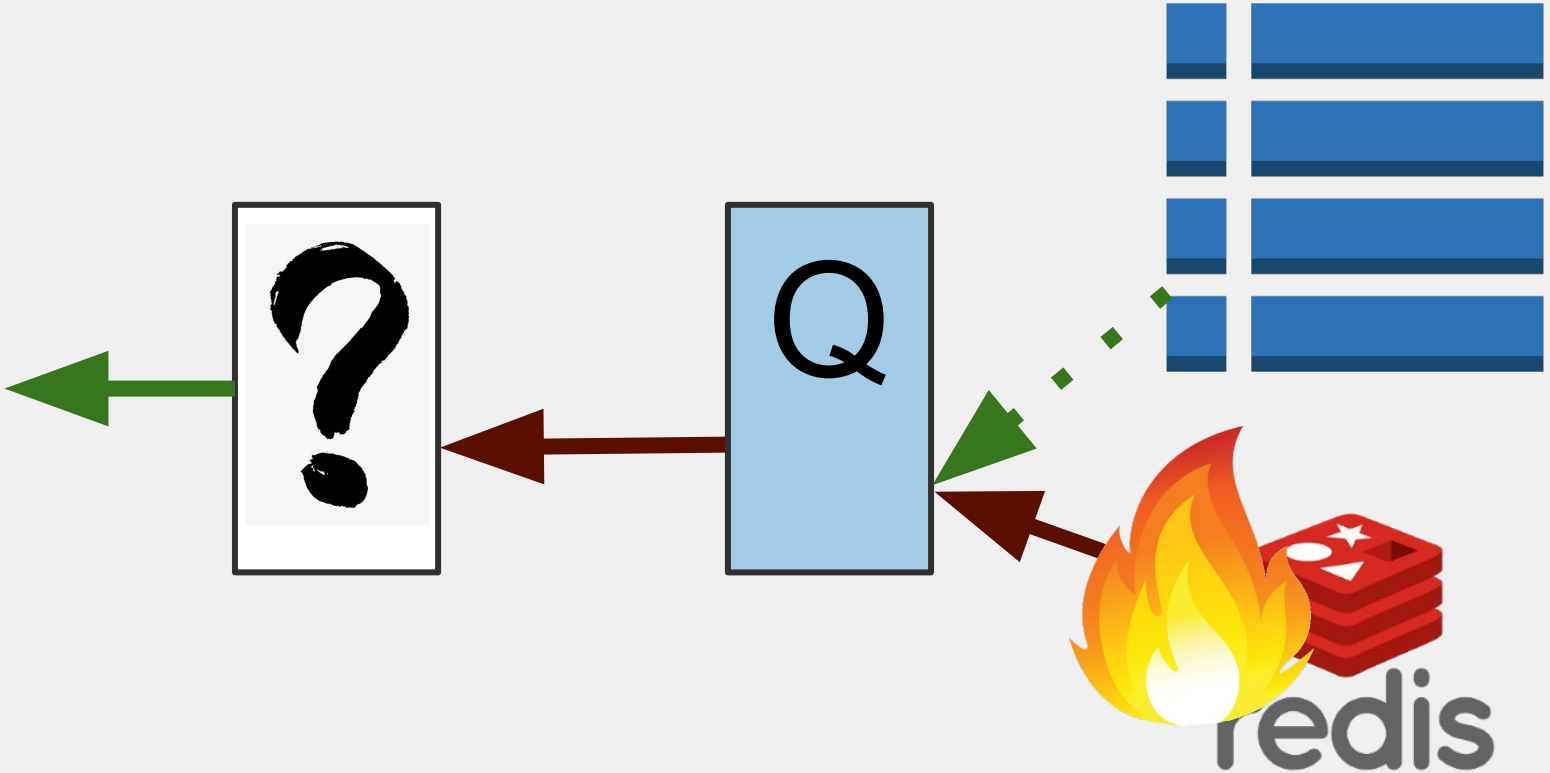

P
Q
redis

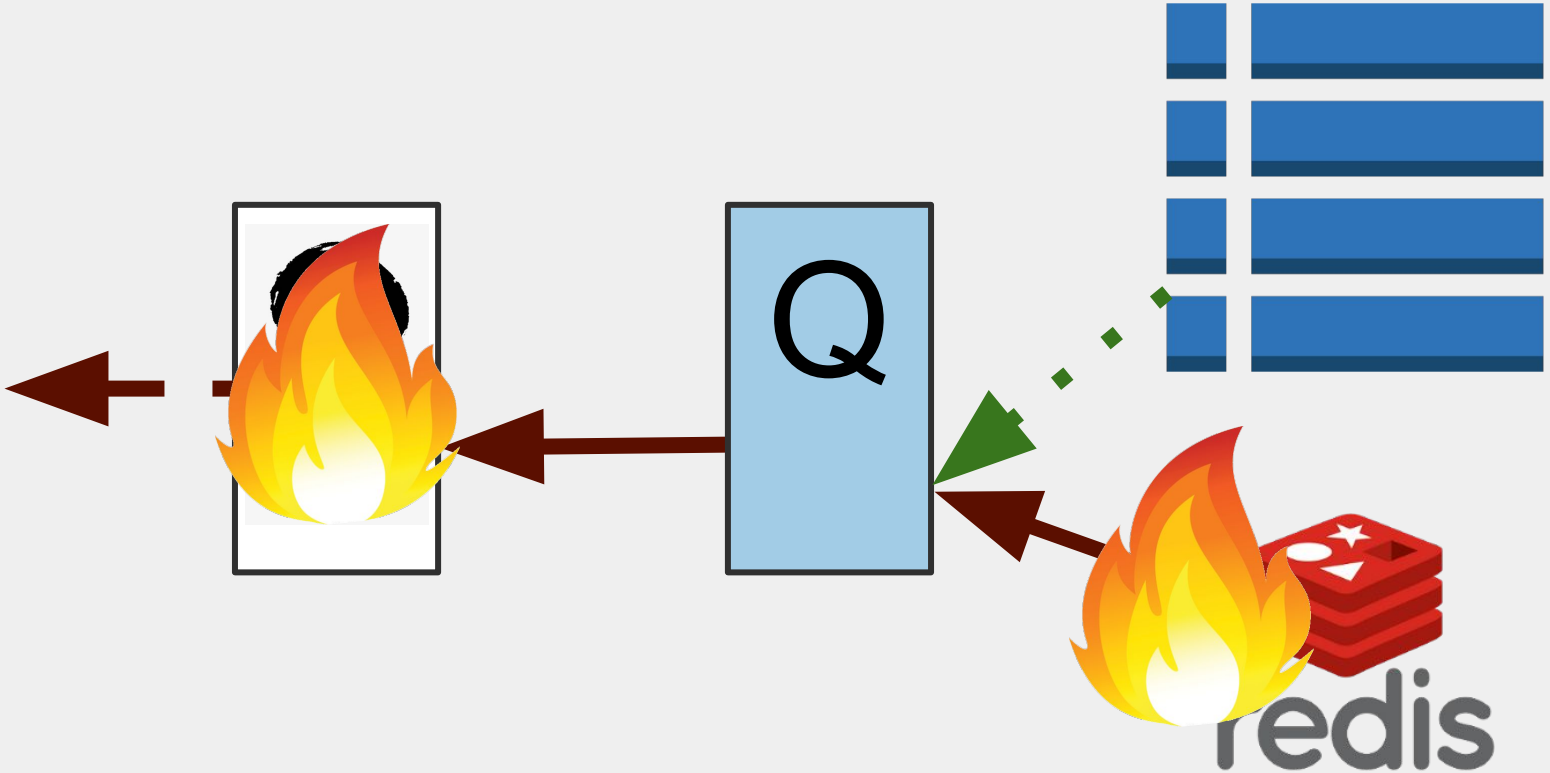

P
Q
redis


P
Q
redis

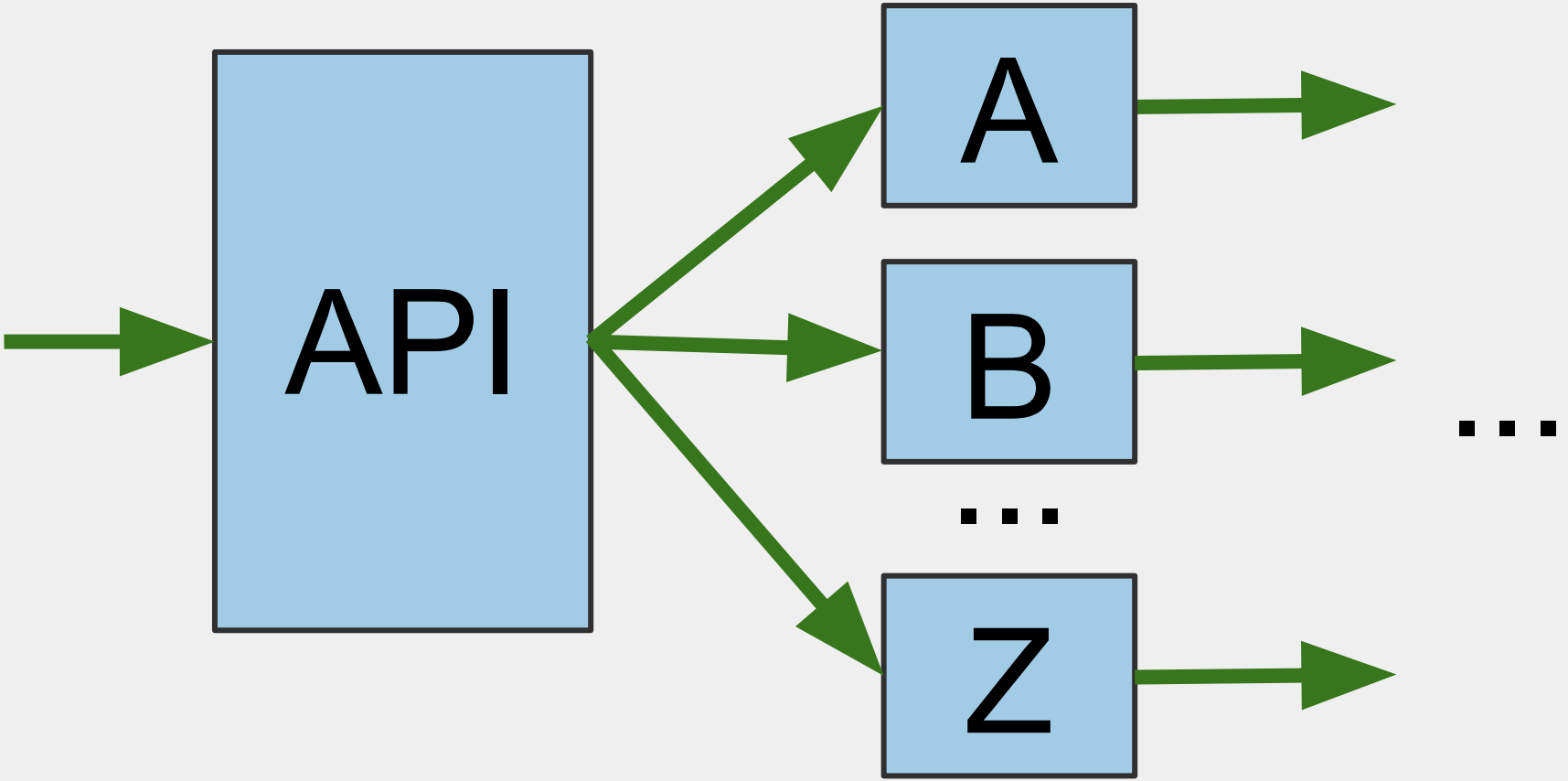

P
Q
redis

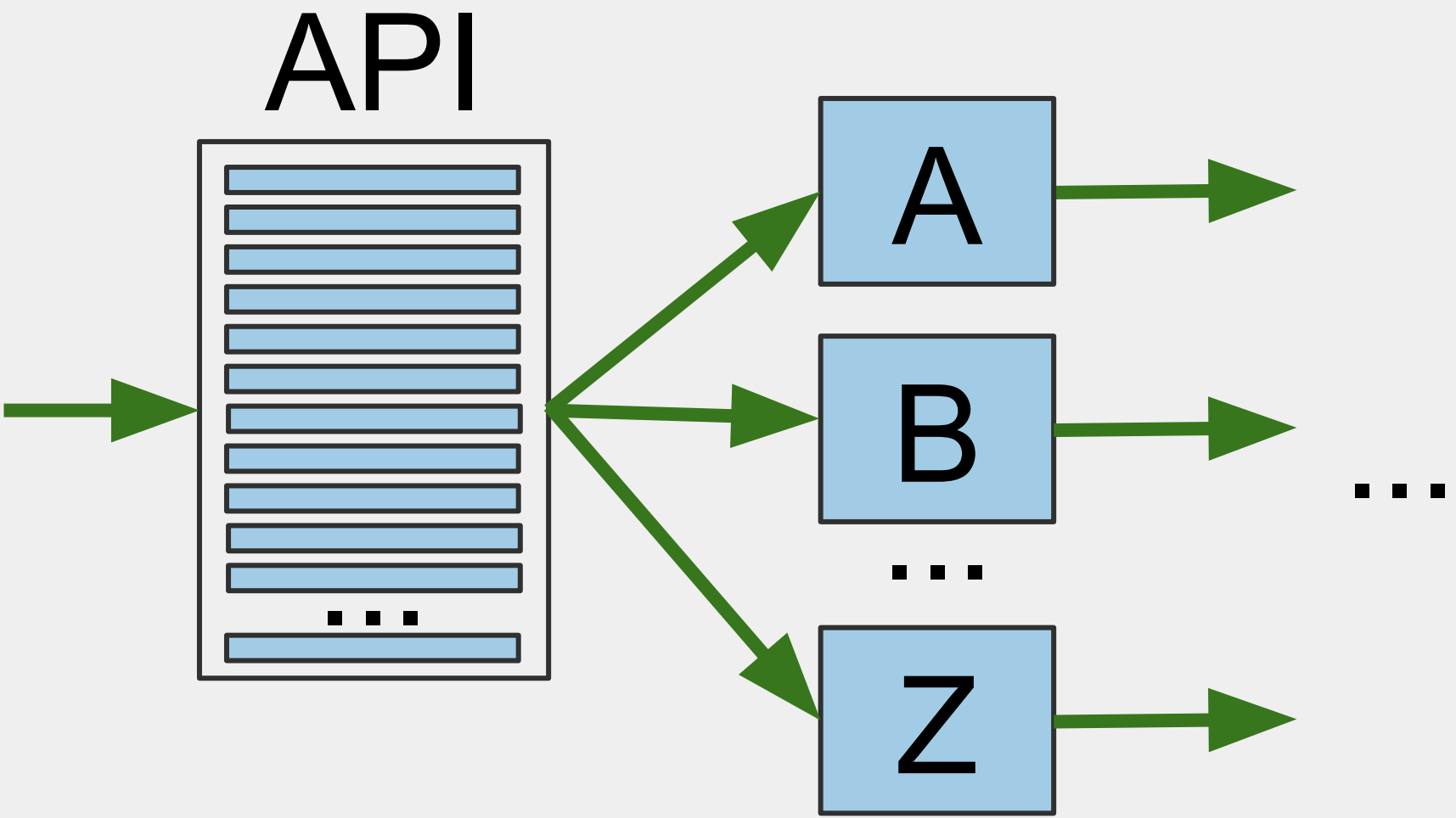

P
Q
redis

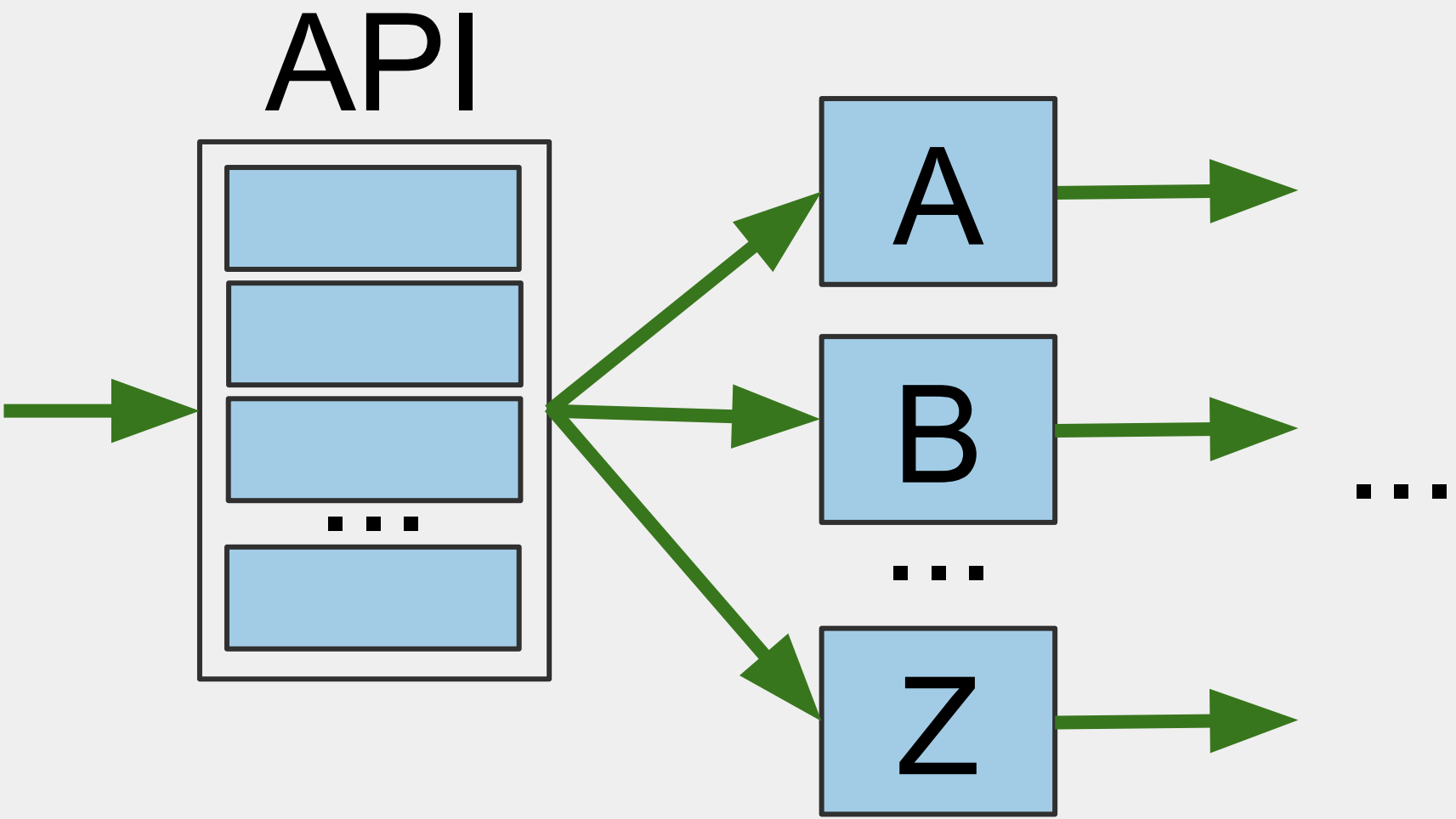

?
Q
redis

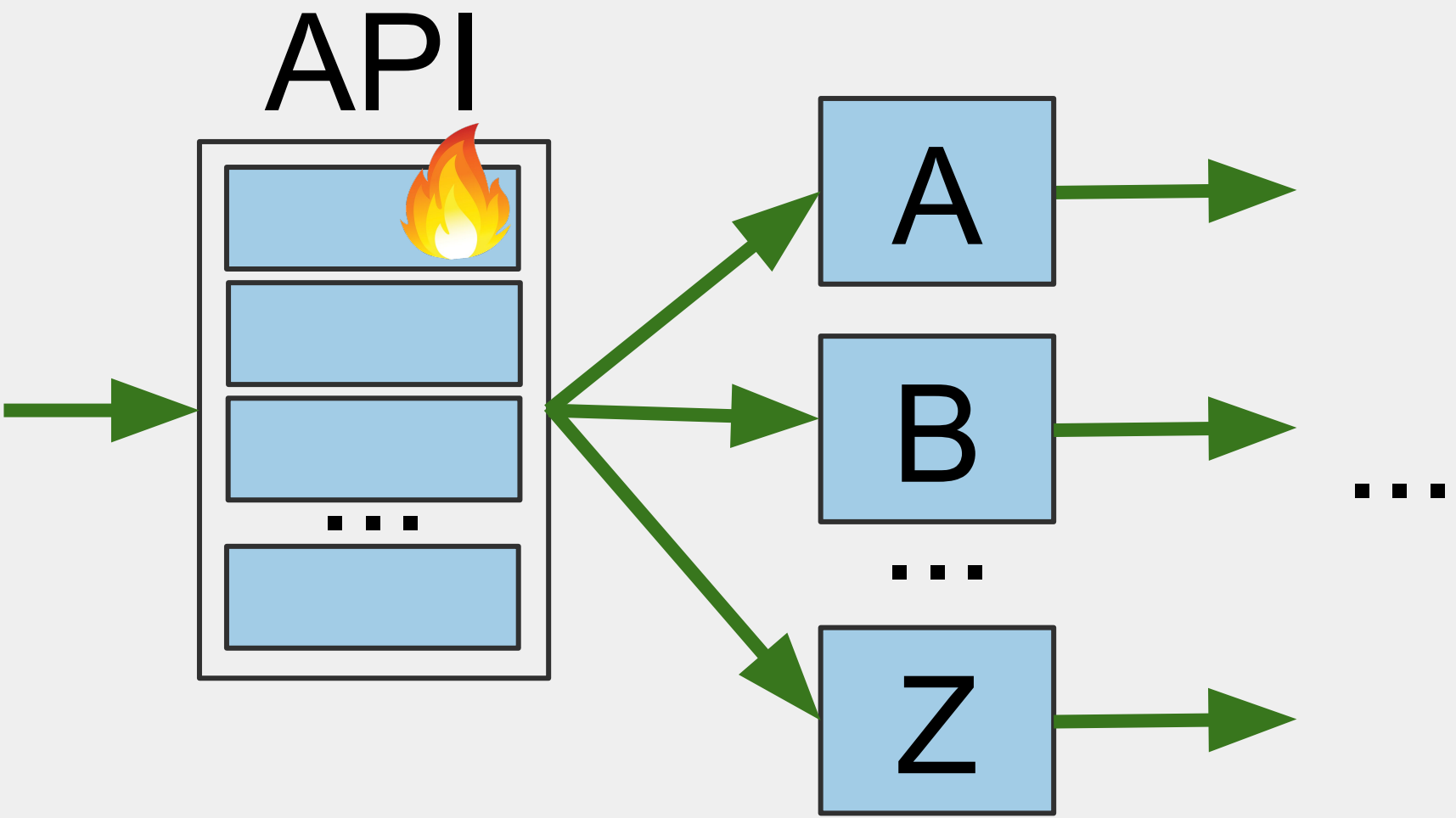

Q
redis

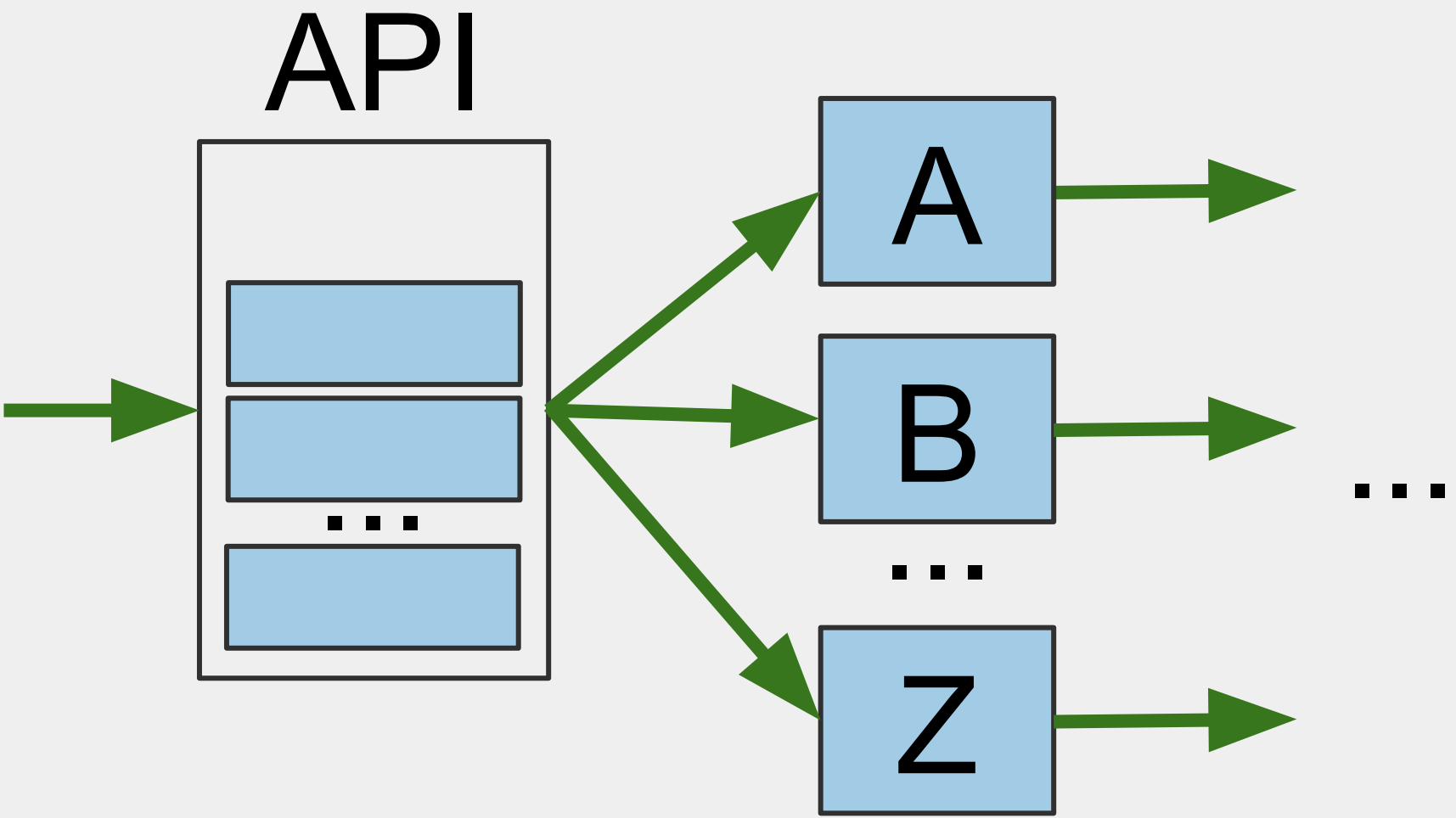
@CaseyRosenthal

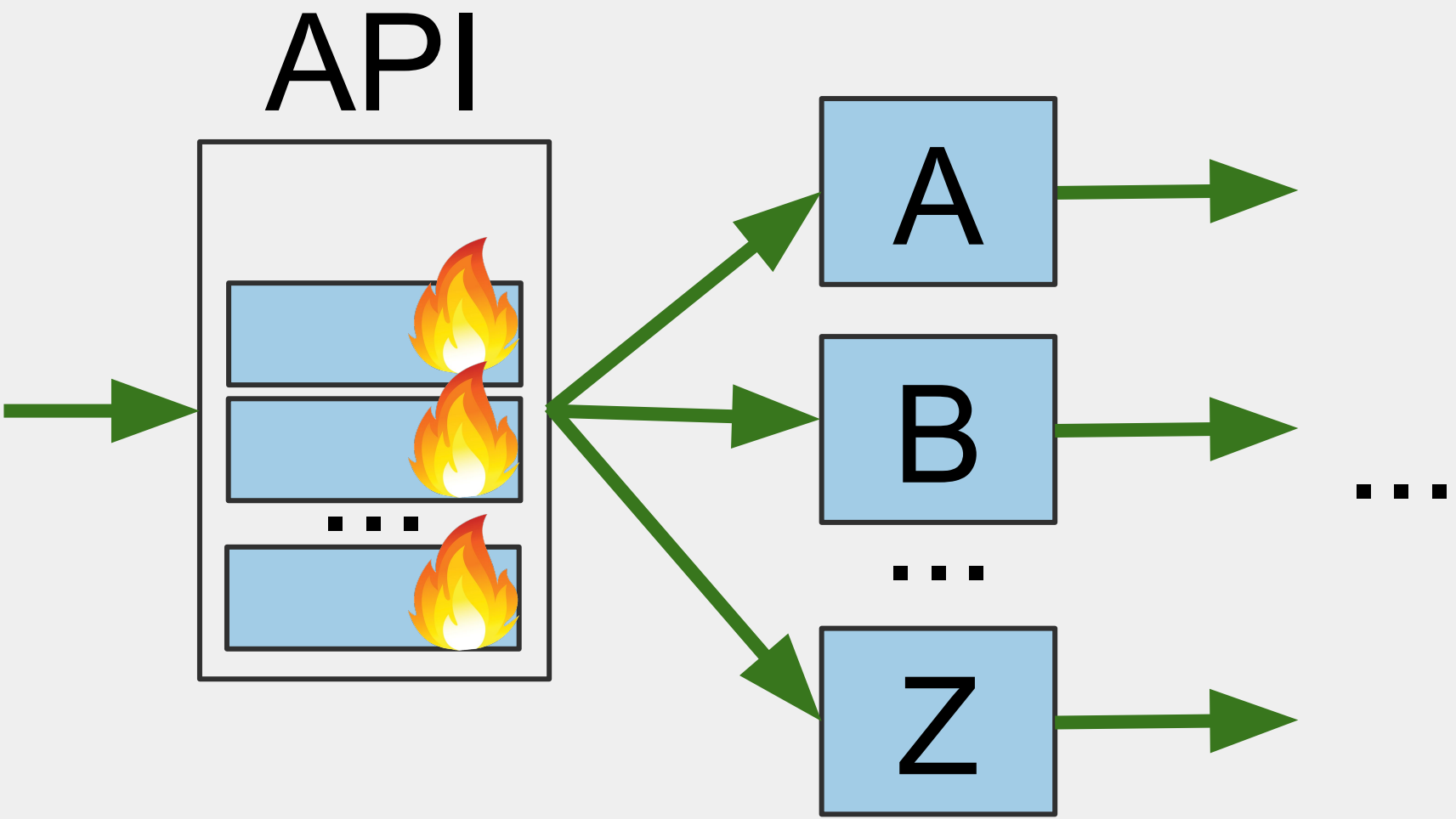

API
A
B
...
...
Z

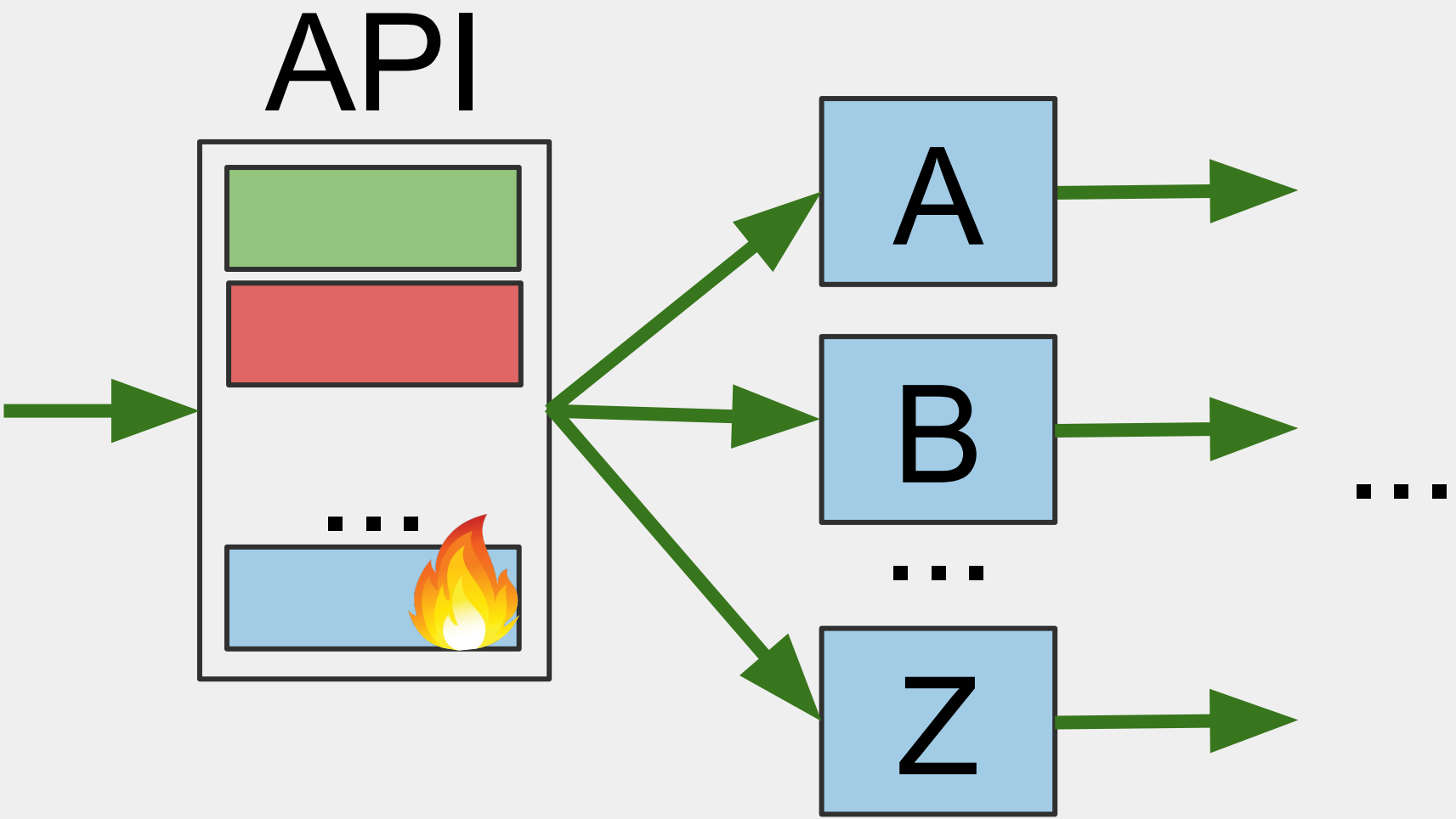

API
A
B
...
Z
...

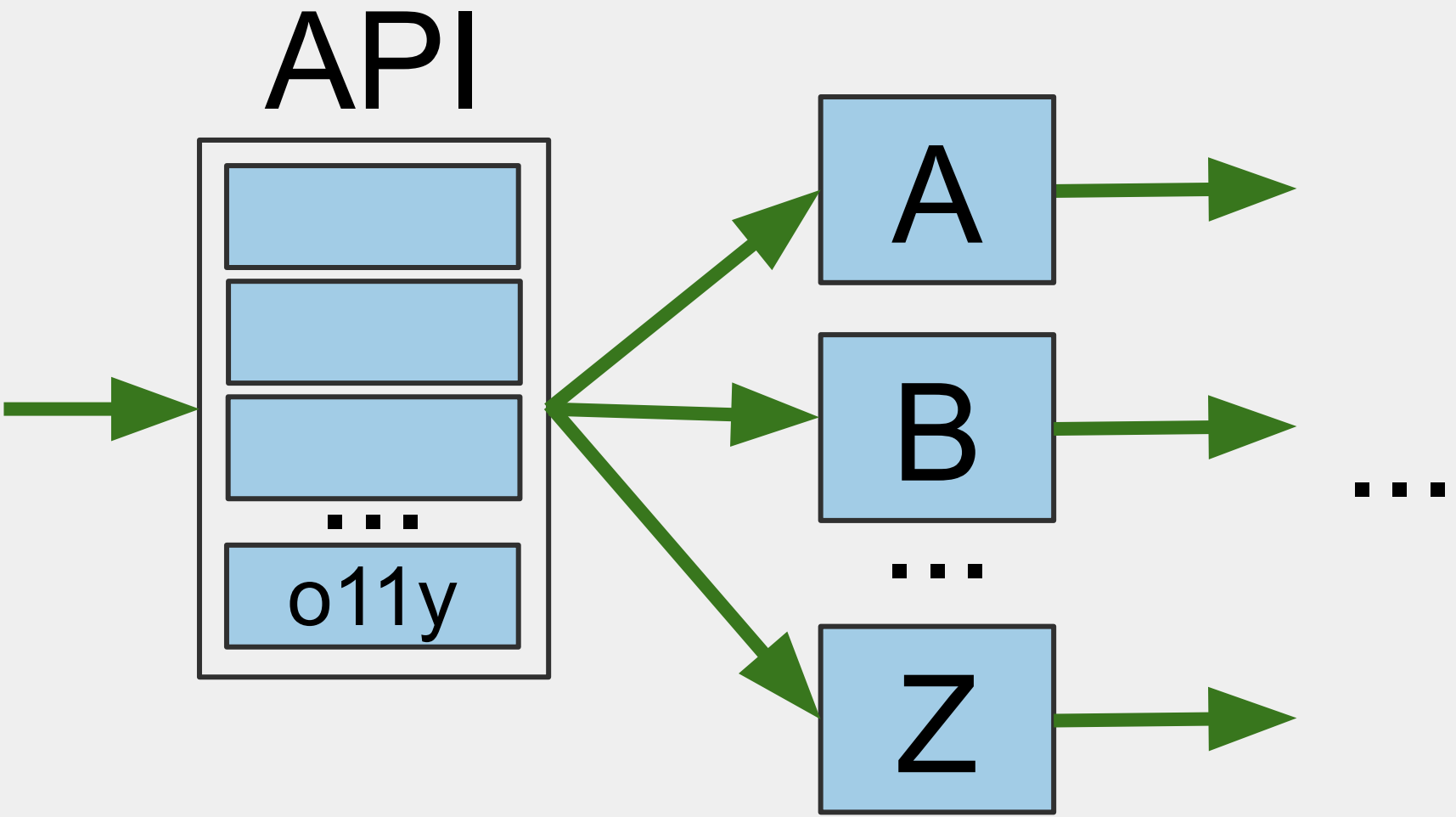

API
A
B
Z

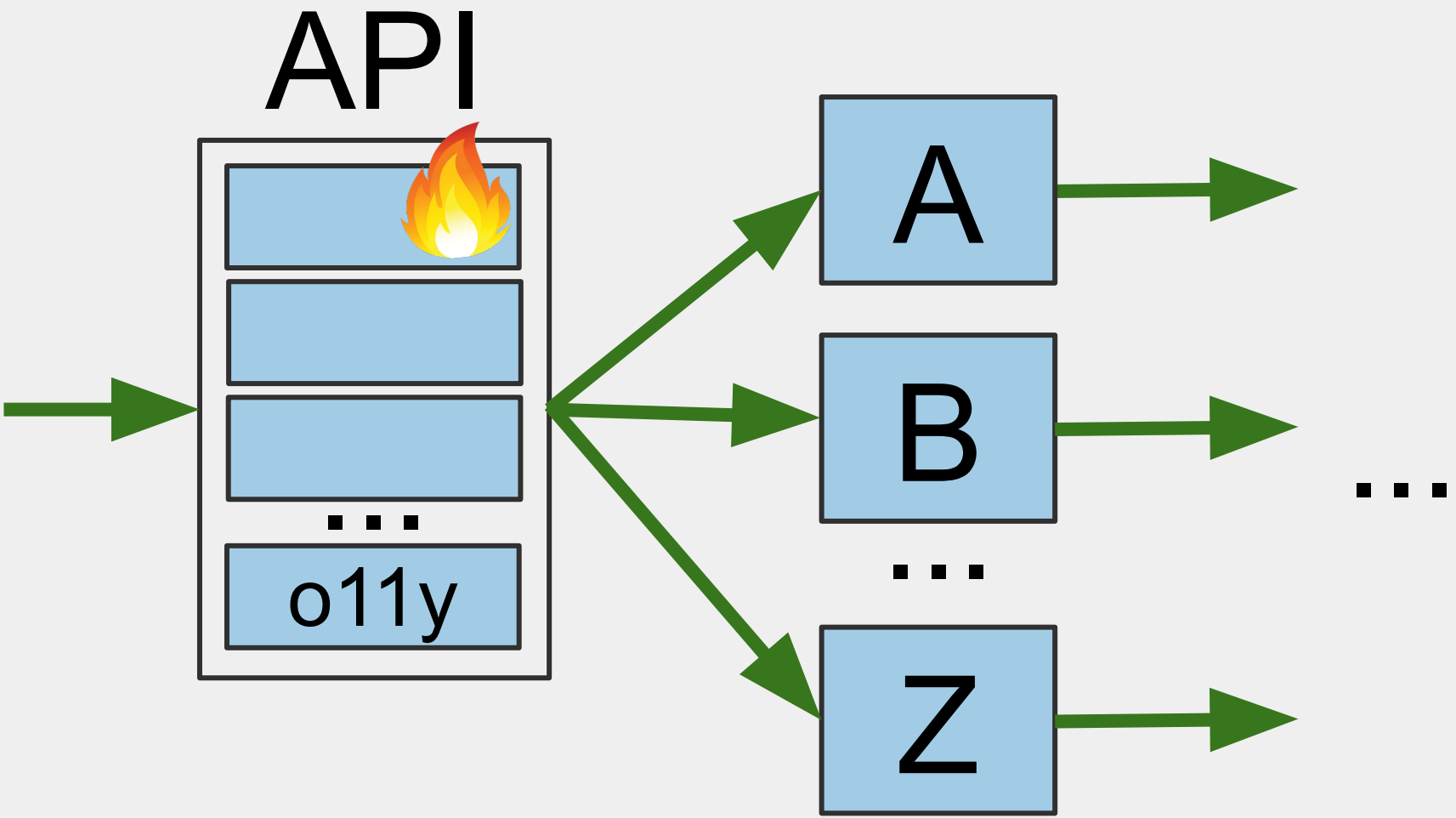

API
A
B
Z

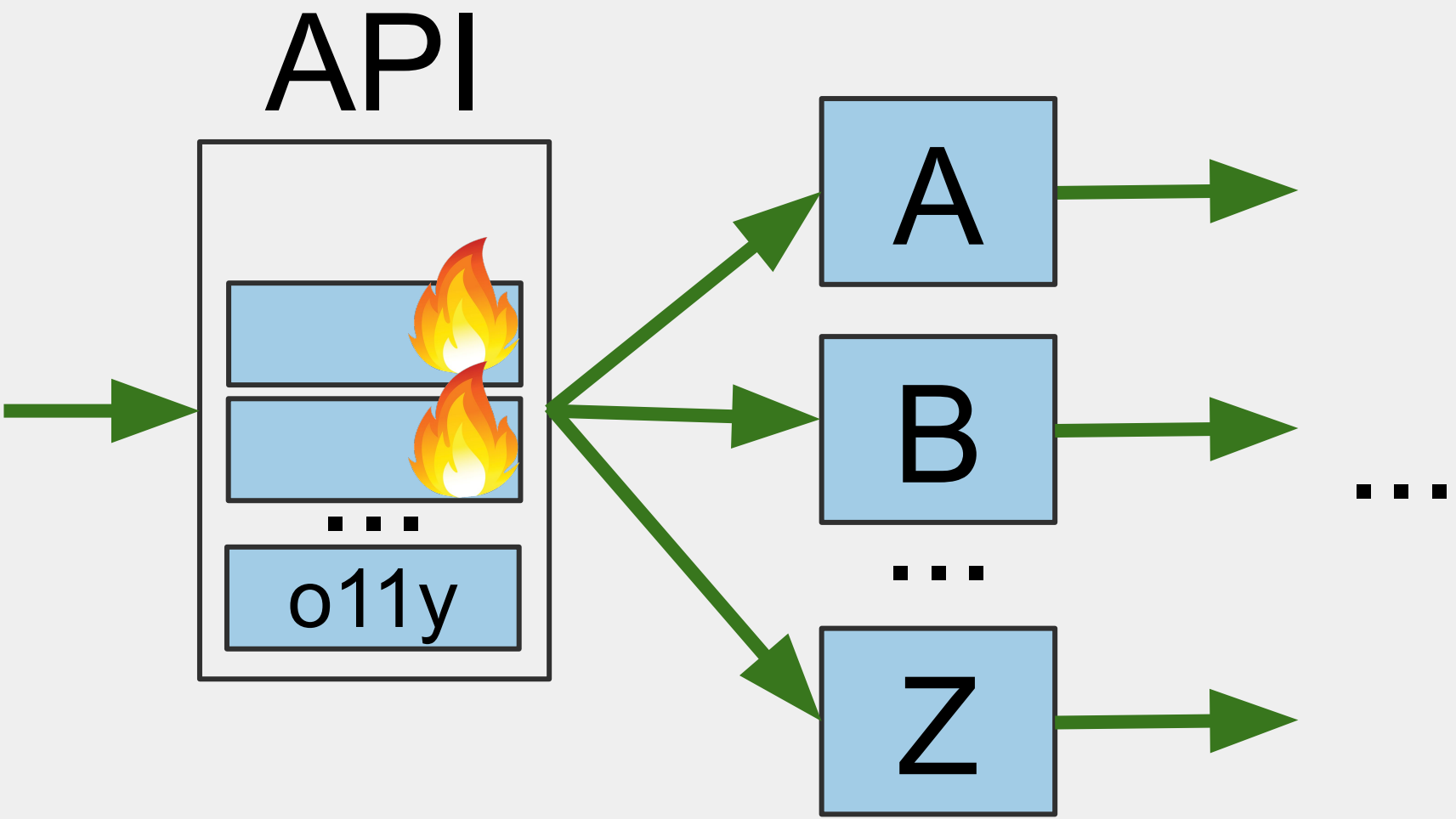

API
A
B
Z

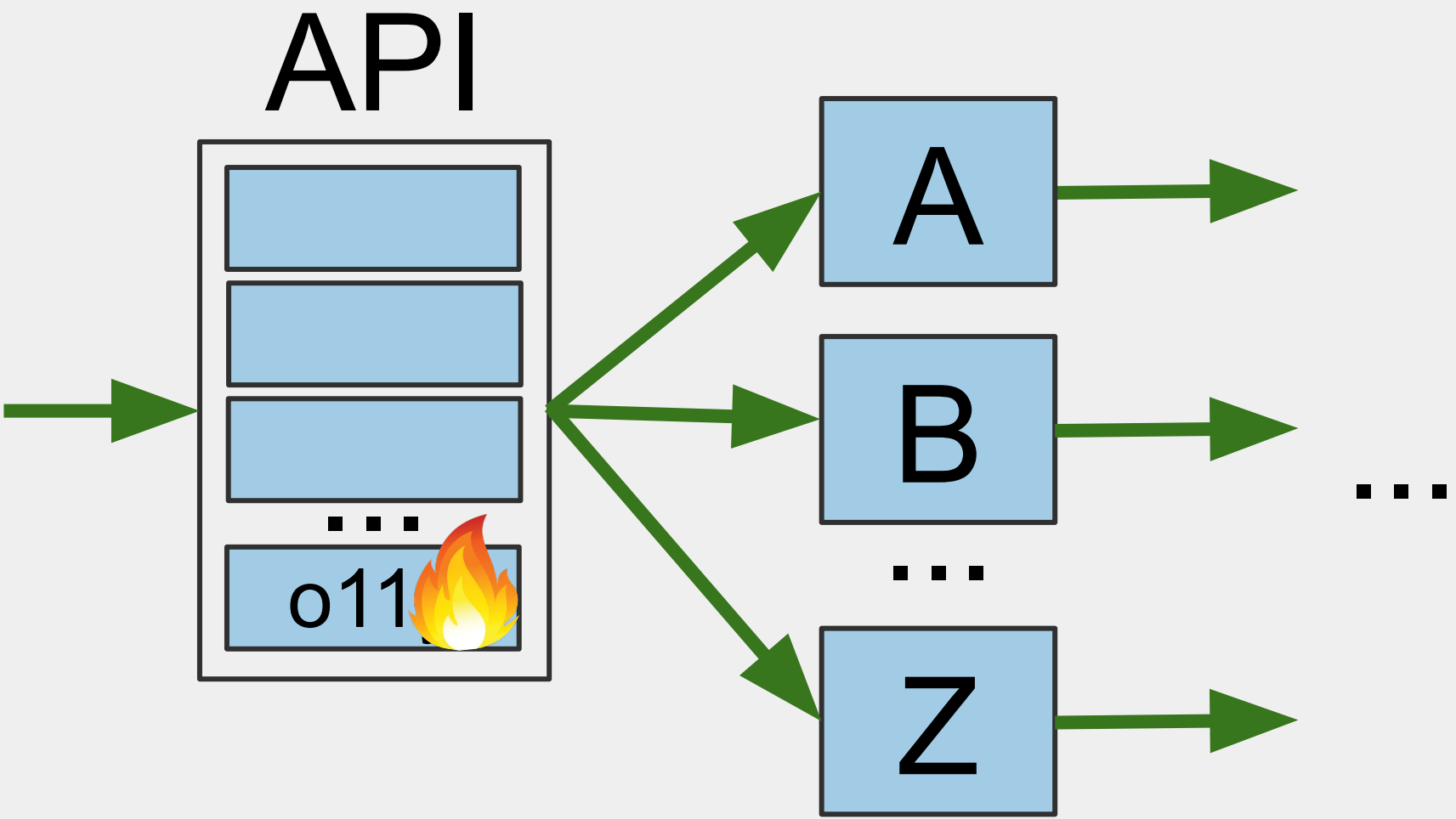

API
A
B
Z


API
A
B
Z


API
...
o11y
A
B
...
Z
...


API
A
B
...
Z
...
...
o11y


API
A
B
...
Z
...
...
o11y

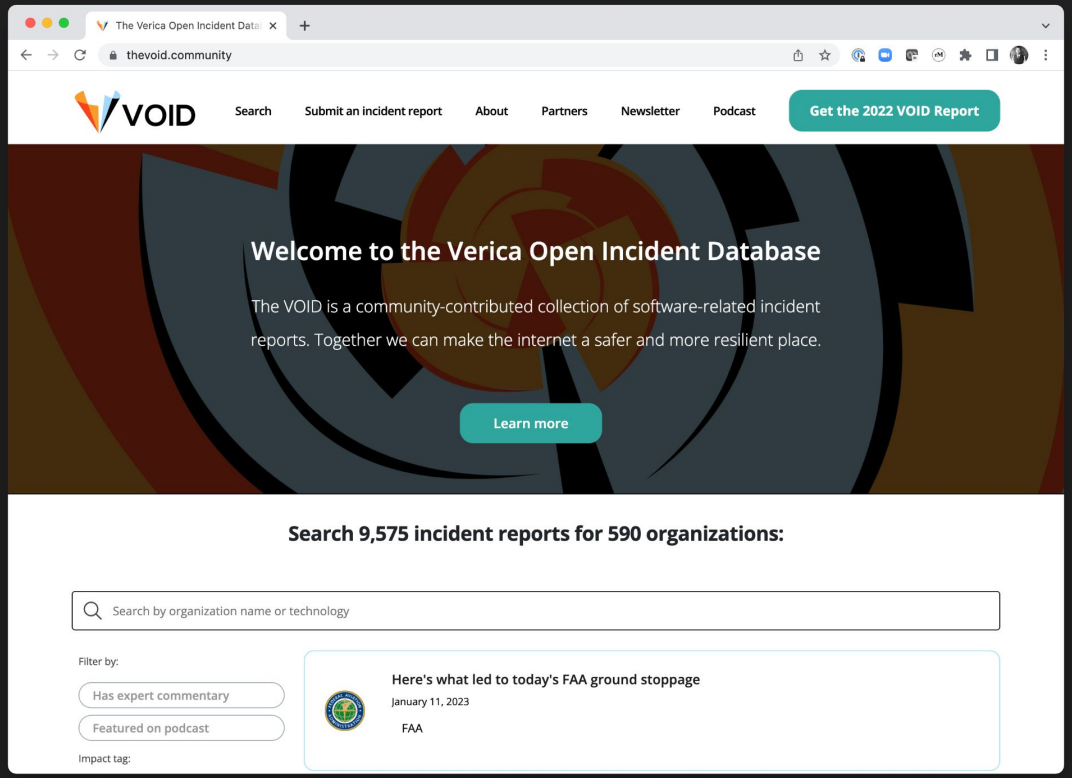

API
o11
A
B
Z

@CaseyRosenthal

All components could be 100% correct, and yet the system exhibits undesirable behavior.

@CaseyRosenthal

@CaseyRosenthal



@CaseyRosenthal

About the VOID

The Verica Open Incident Database (VOID) makes public software-related incident reports available to everyone, raising awareness and increasing understanding of software-based failures in order to make the internet a more resilient and safe place.

https://www.thevoid.community

VOID

@CaseyRosenthal

@CaseyRosenthal

- Myth 1: Remove the People Who Cause Accidents

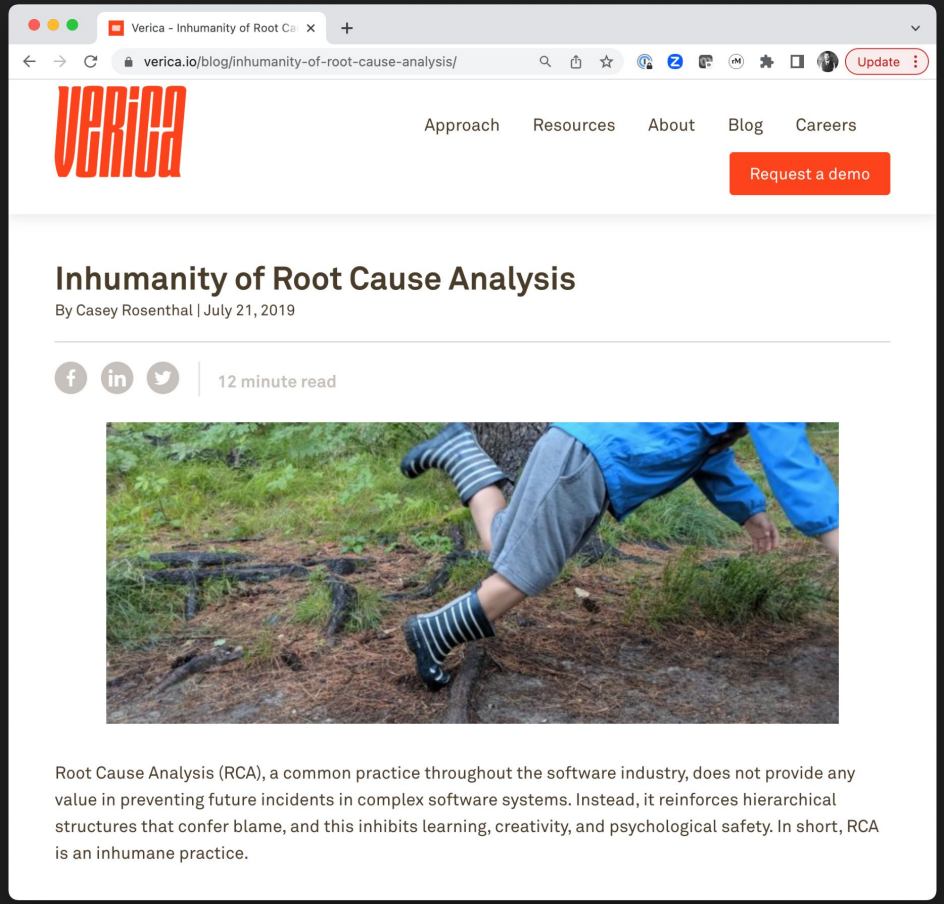
@CaseyRosenthal

- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks

@CaseyRosenthal

- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes

@CaseyRosenthal



@CaseyRosenthal

- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes

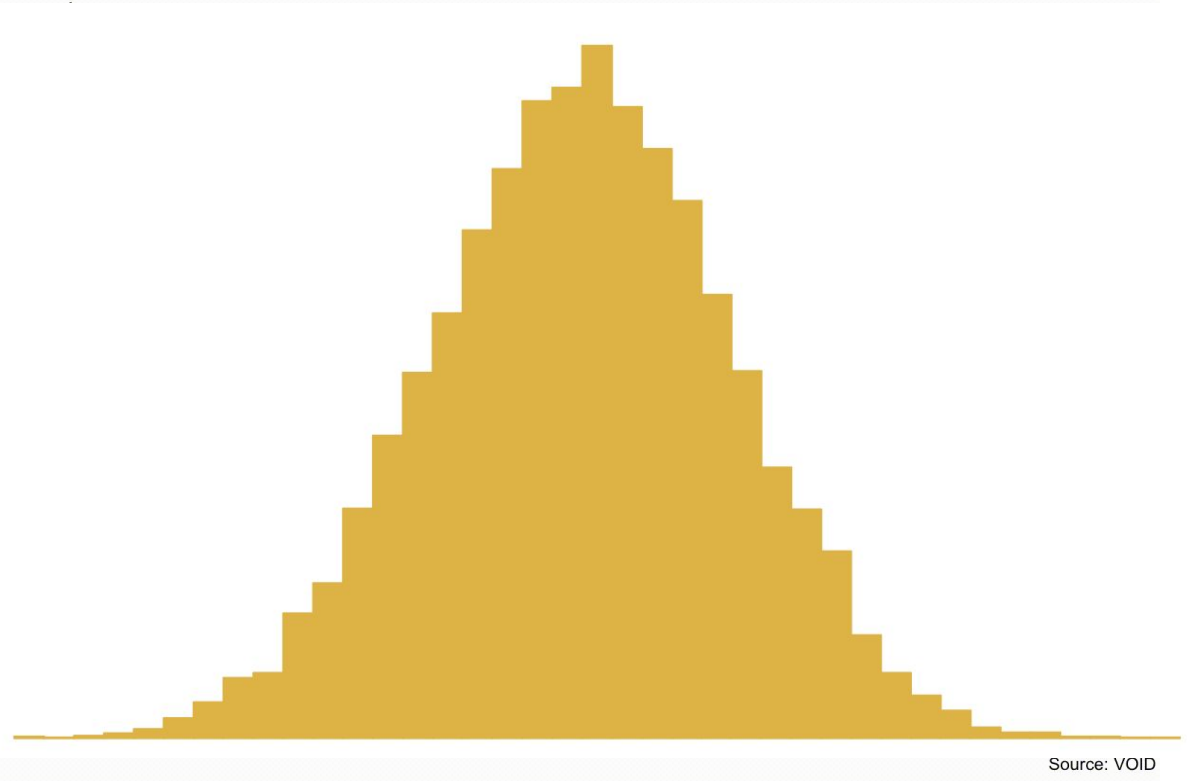
@CaseyRosenthal

- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes
- Myth 4: Measure Reliability Quantitatively

@CaseyRosenthal

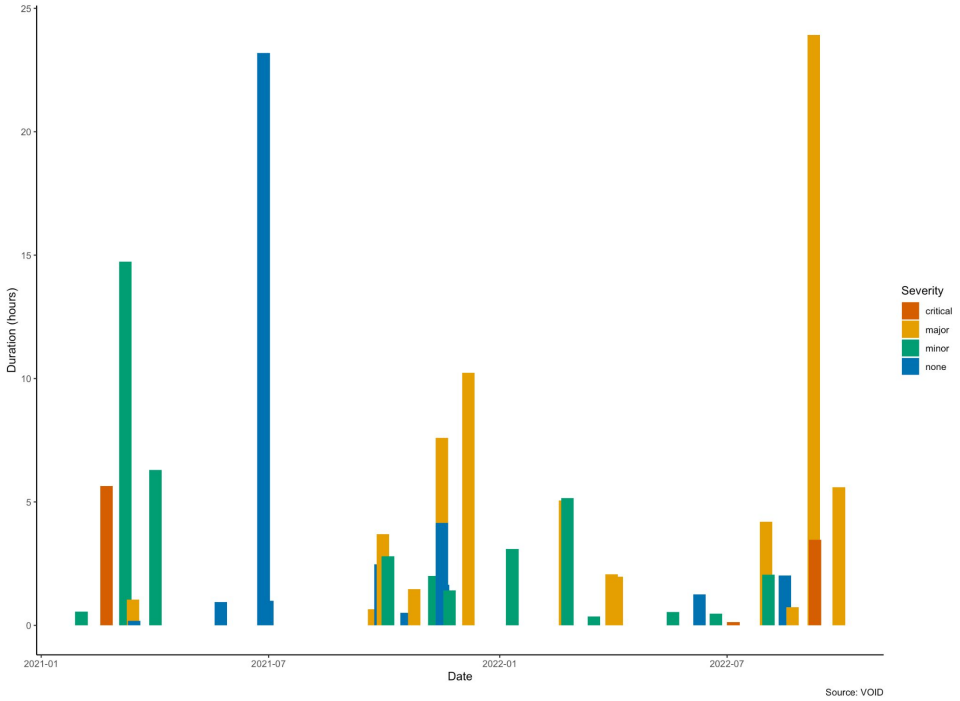
Mean Time to Resolution MTTR

The Distribution Matters



Source: VOID

VOID



THEY'RE NOT RELATED!

We analyzed status page data across almost 7K incidents from 10 different companies.

- Only 2 of them showed *very* weak correlations between duration and severity.
- R = -.18 and -.17, respectively ($p < .05$)

VOID

MTTR Fatal Flaws

- Provably wrong statistic
- Statistically insignificant data sample
- Duration doesn’t correlate with severity
- Measurement errors
- Unactionable analysis

MTTR Can’t Tell You

- How reliable your software or systems are
- How agile/effective your team or organization is
- If you’re getting better at responding to incidents
- Whether the next one will be longer or shorter
- How “bad” any given incident is

VOID

- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes
- Myth 4: Measure Reliability Quantitatively

@CaseyRosenthal

- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes
- Myth 4: Measure Reliability Quantitatively
- Myth 5: Avoid Risk

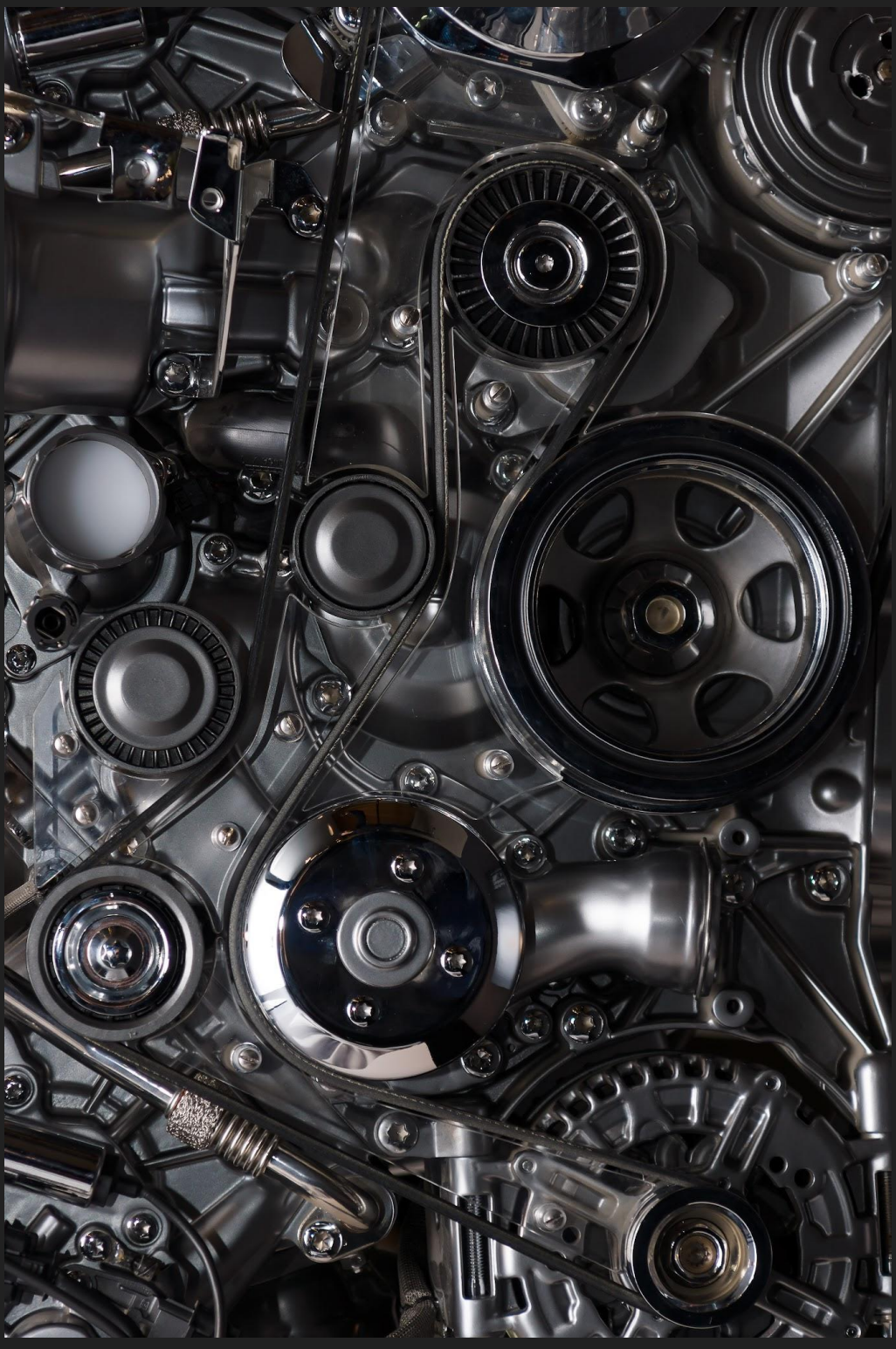
@CaseyRosenthal

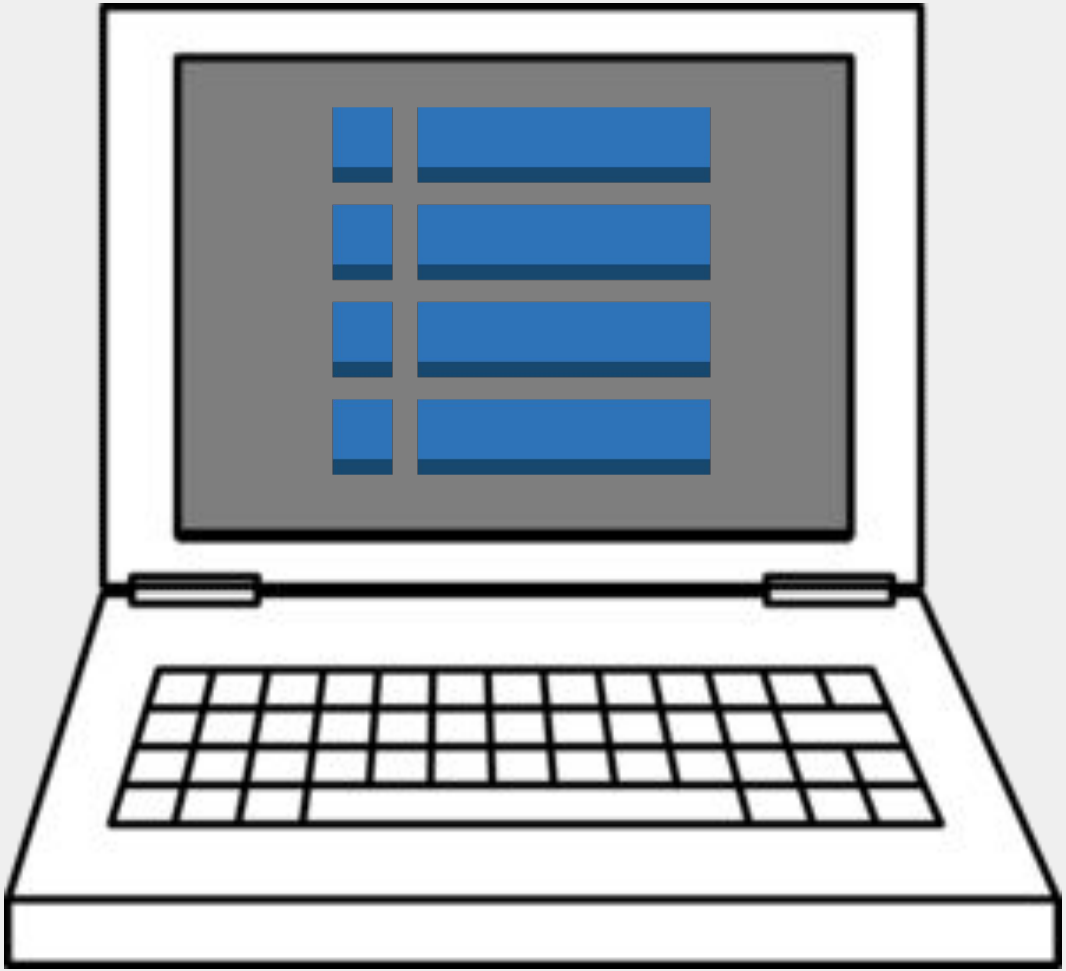
- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes
- Myth 4: Measure Reliability Quantitatively
- Myth 5: Avoid Risk
- Myth 6: Simplify

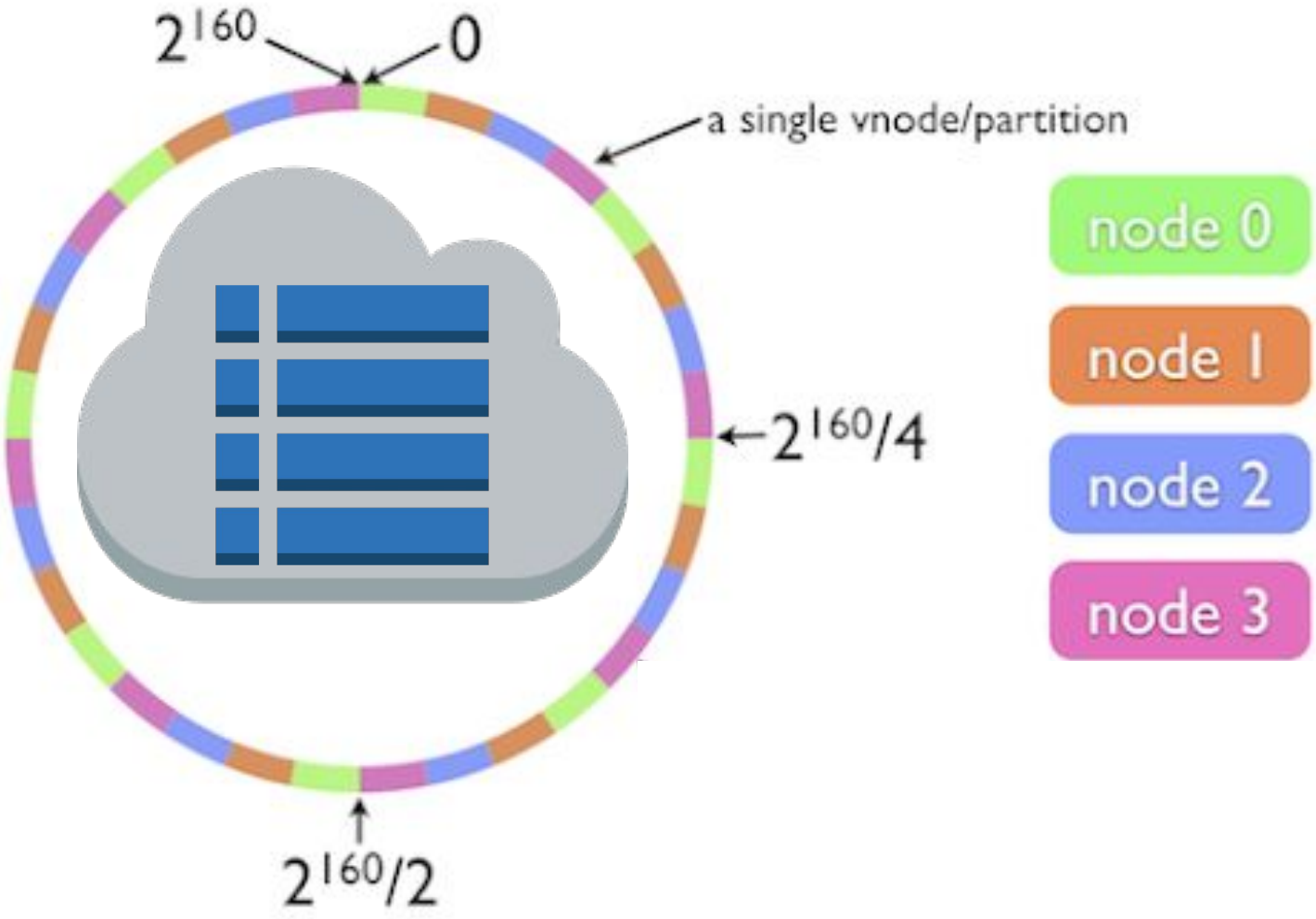
@CaseyRosenthal

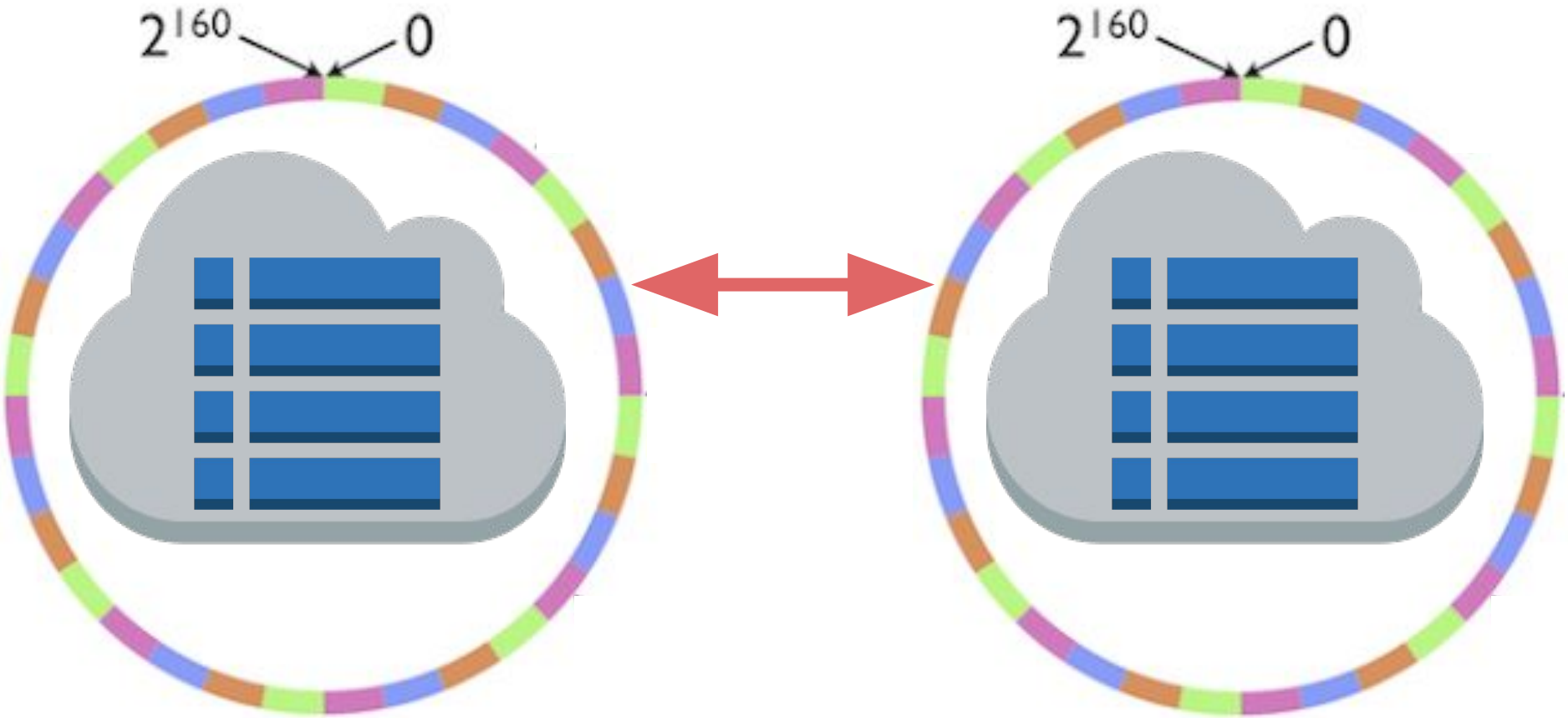
ACCIDENTAL

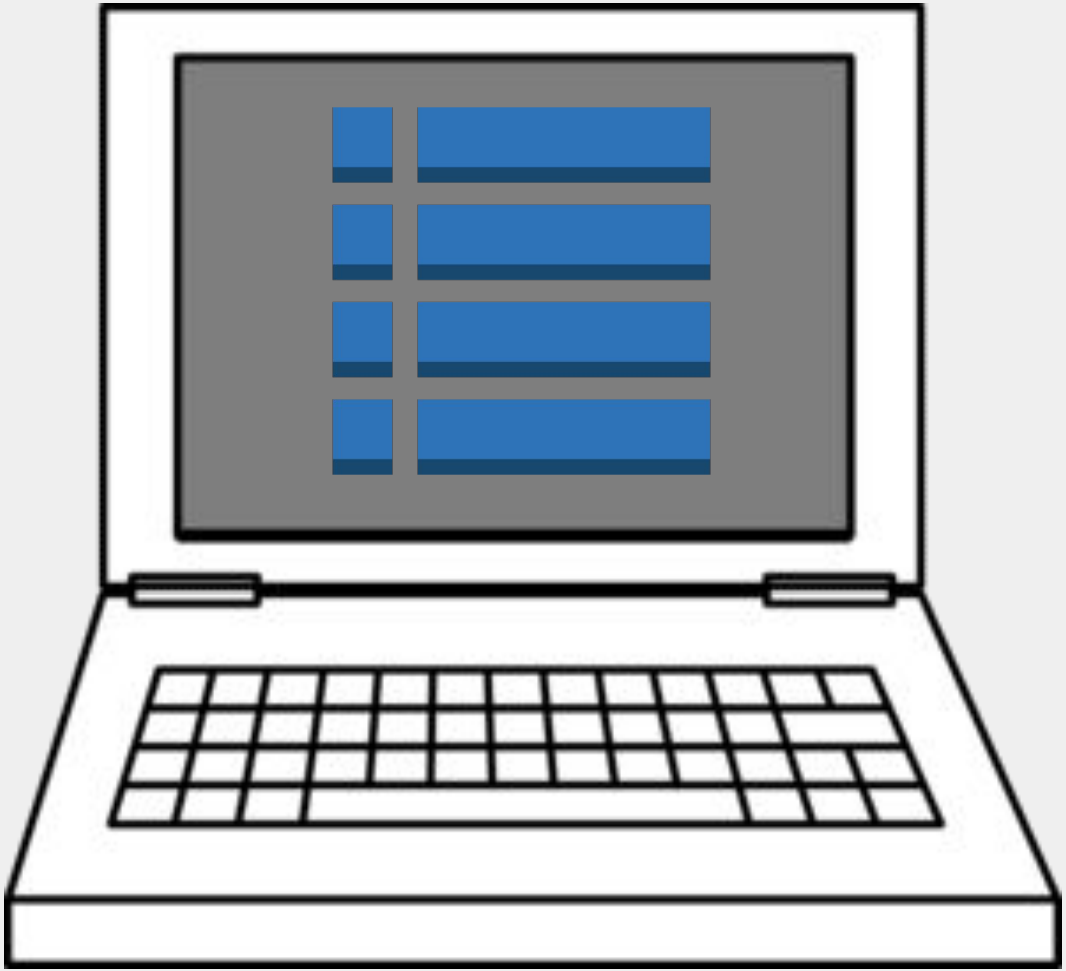
ESSENTIAL





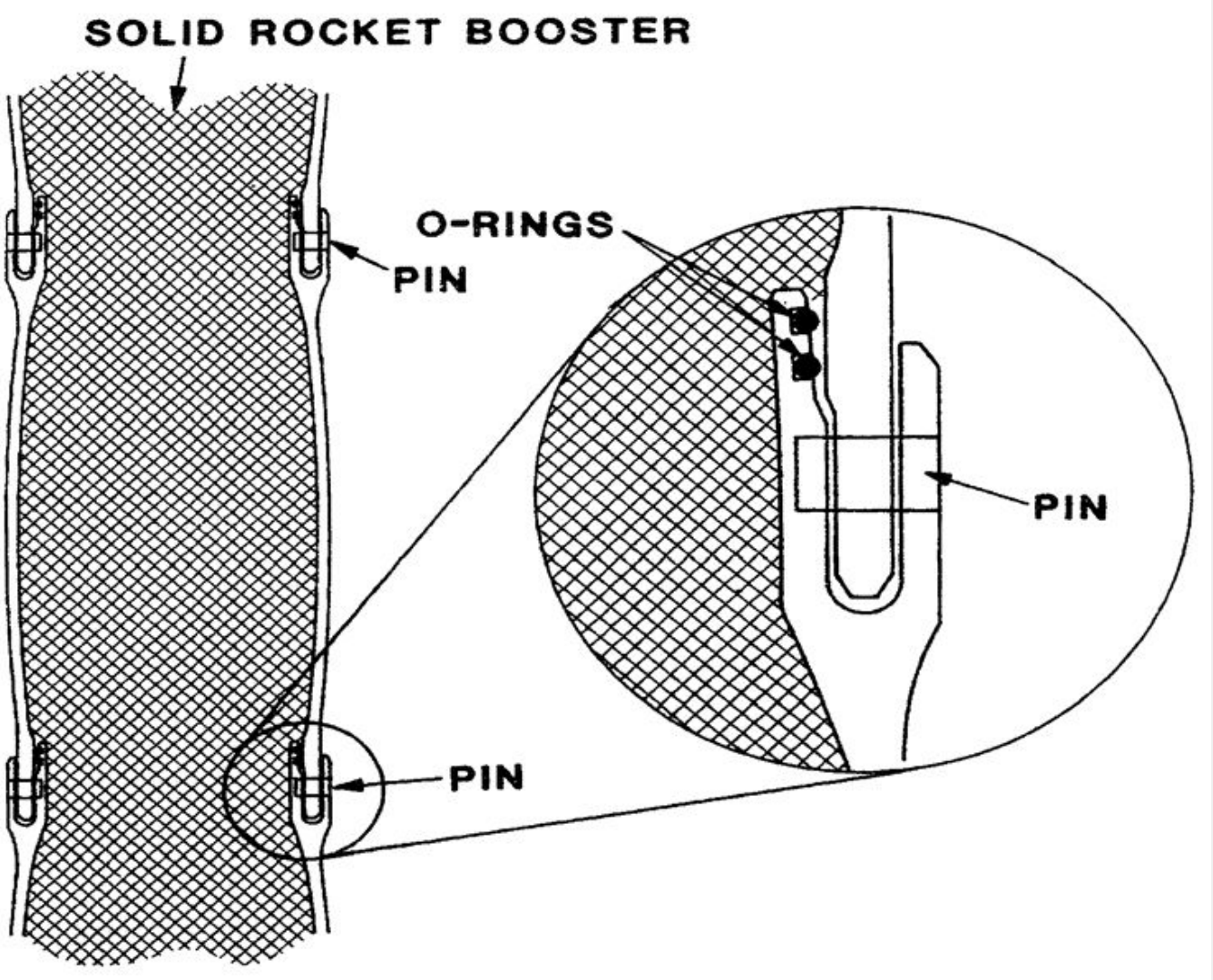

2^{160}
0
a single vnode/partition
$2^{160}/4$
$2^{160}/2$
node 0
node 1
node 2
node 3


2^{160}
0
2^{160}
0



- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes
- Myth 4: Measure Reliability Quantitatively
- Myth 5: Avoid Risk
- Myth 6: Simplify
- Myth 7: Add Redundancy

@CaseyRosenthal


SOLID ROCKET BOOSTER
O-RINGS
PIN
PIN
PIN

- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes
- Myth 4: Measure Reliability Quantitatively
- Myth 5: Avoid Risk
- Myth 6: Simplify
- Myth 7: Add Redundancy

@CaseyRosenthal

How do we survive the undesirable effects of complex systems?

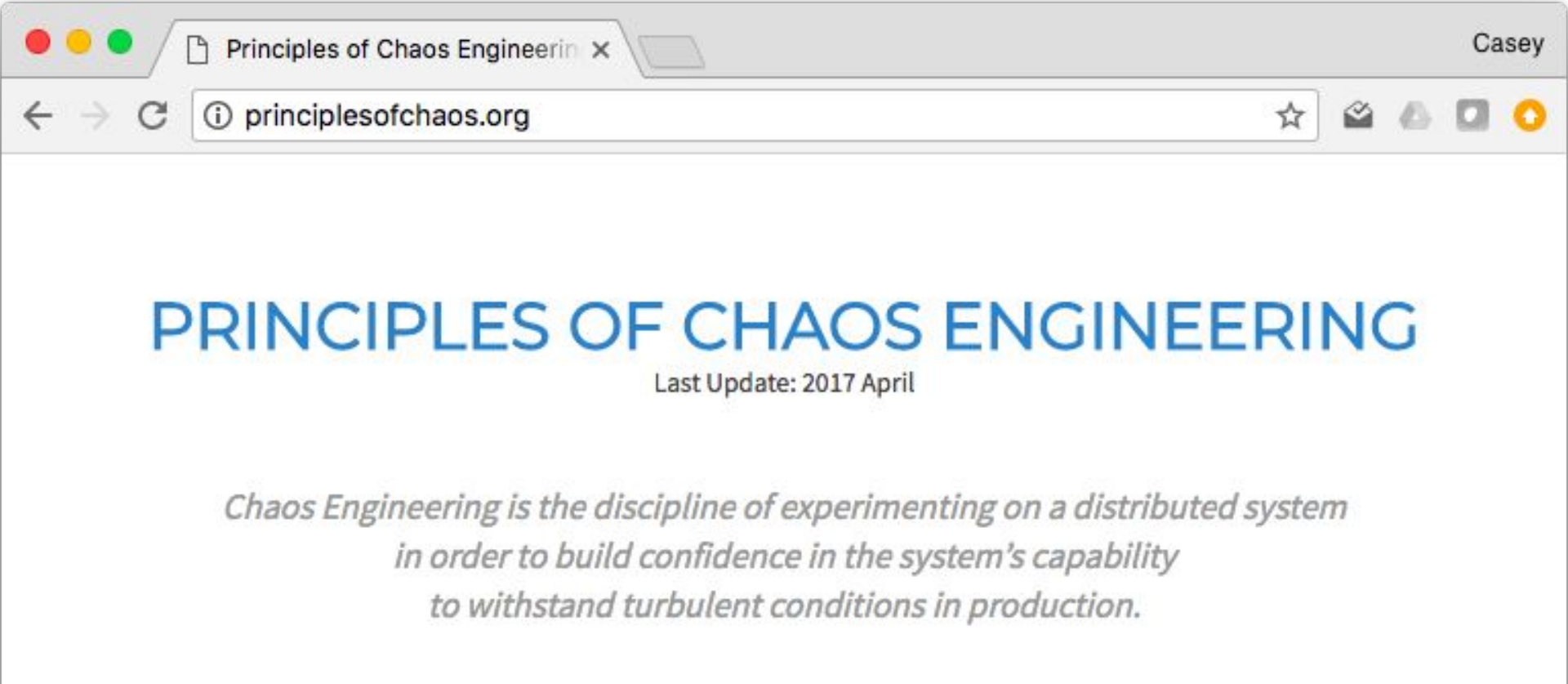
@CaseyRosenthal

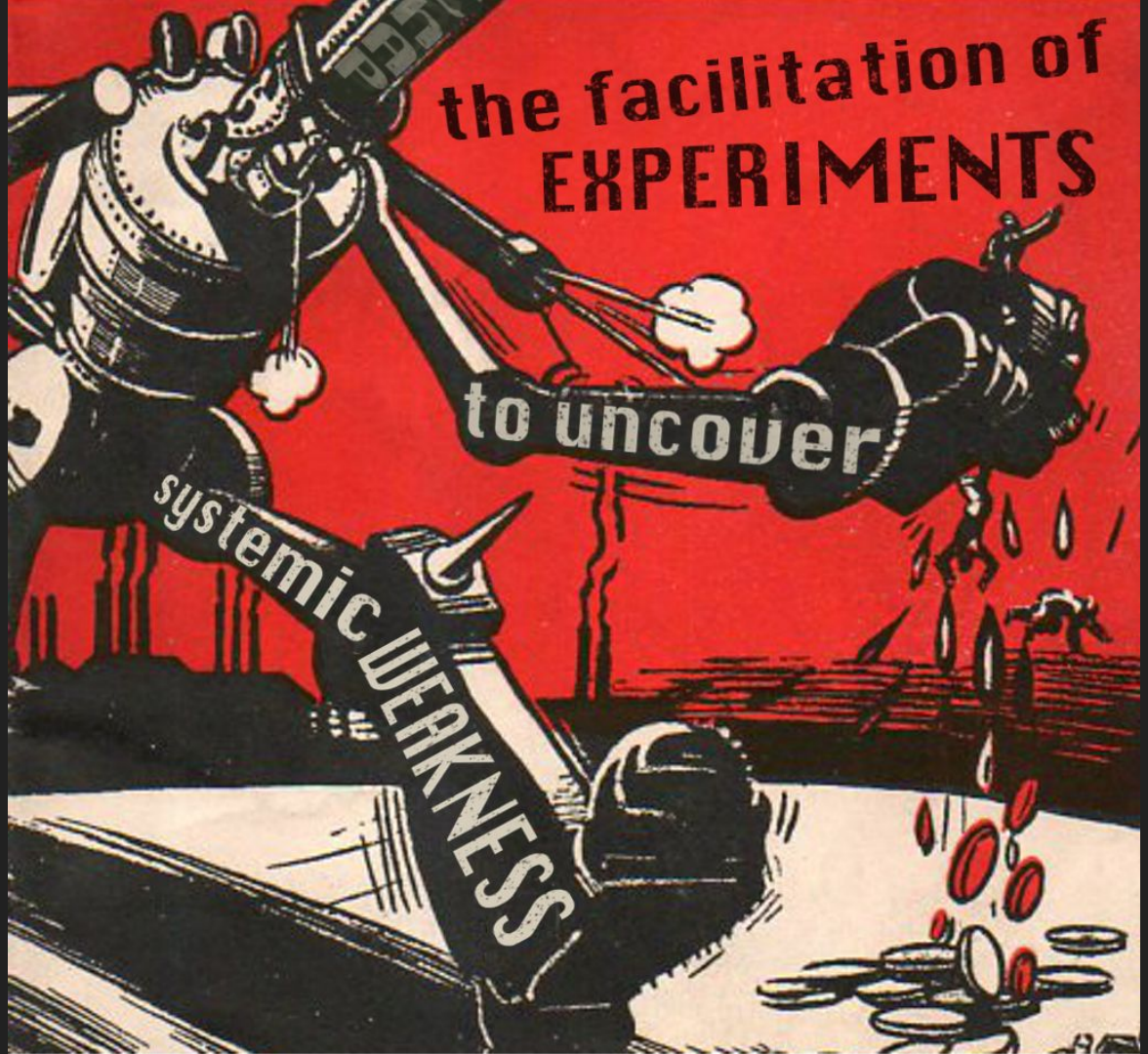
How do we
make systems reliable?

@CaseyRosenthal

@CaseyRosenthal

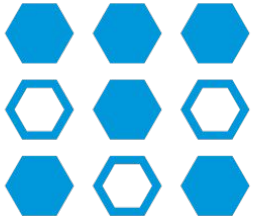

ECONOMICS
WORKLOAD
SAFETY


Principles of Chaos Engineerin
Casey
principlesofchaos.org
PRINCIPLES OF CHAOS ENGINEERING
Last Update: 2017 April
Chaos Engineering is the discipline of experimenting on a distributed system
in order to build confidence in the system's capability
to withstand turbulent conditions in production.

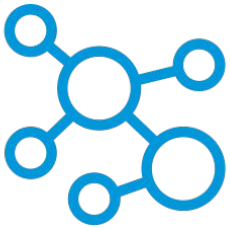

the facilitation of
EXPERIMENTS
to uncover
systemic WEAKNESS

@CaseyRosenthal

Economic Pillars of Complexity

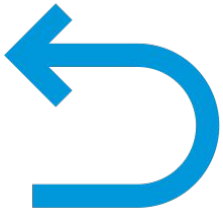


STATES



RELATIONSHIPS

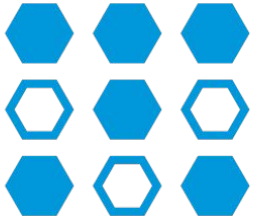
ENVIRONMENT



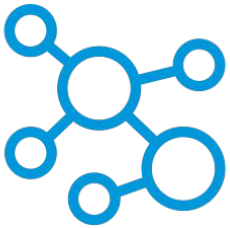
REVERSIBILITY

@CaseyRosenthal

Economic Pillars of Complexity

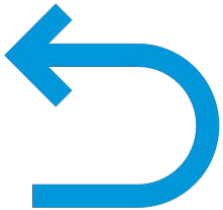


STATES



RELATIONSHIPS

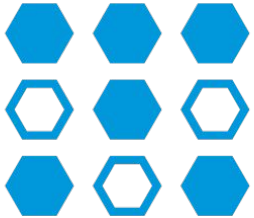
ENVIRONMENT



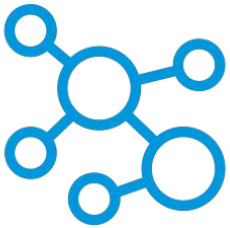
REVERSIBILITY

@CaseyRosenthal

Economic Pillars of Complexity

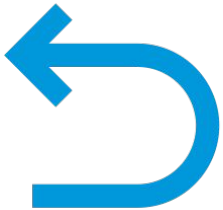


STATES



RELATIONSHIPS

ENVIRONMENT

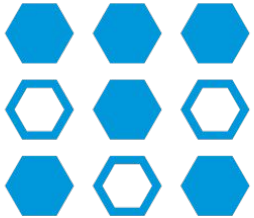


REVERSIBILITY

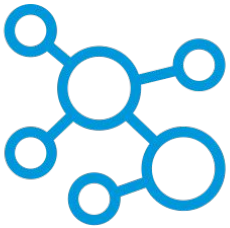
@CaseyRosenthal

THE BOSSES OF THE SENATE.

Economic Pillars of Complexity

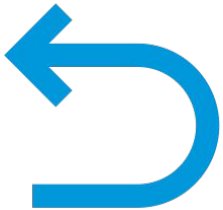


STATES



RELATIONSHIPS

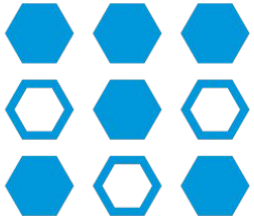
ENVIRONMENT



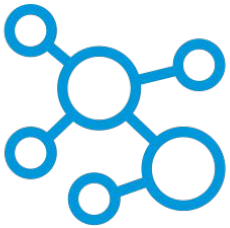
REVERSIBILITY

@CaseyRosenthal

Economic Pillars of Complexity

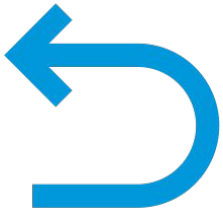


STATES



RELATIONSHIPS

ENVIRONMENT



REVERSIBILITY

@CaseyRosenthal

Software Engineering: the Reversibility Profession

@CaseyRosenthal

“The chief merit of [software engineering] is its technical efficiency, with a premium placed on precision, speed, expert control, continuity, discretion, and optimal returns on input.”

-Merton

s/bureaucracy/software engineering/

@CaseyRosenthal

“The chief merit of bureaucracy is its technical efficiency, with a premium placed on precision, speed, expert control, continuity, discretion, and optimal returns on input.”

-Merton

Software Engineering: the Bureaucratic Profession

@CaseyRosenthal

Software Engineering: doing it WRONG since 1913

@CaseyRosenthal

@CaseyRosenthal

Evolution of complex system operations.

CI

One of the most efficient methods for uncovering misalignments in software is to put the code together and run it. **Continuous Integration** was promoted heavily as part of XP methodology as a way to achieve this and is now a common industry norm.

CD

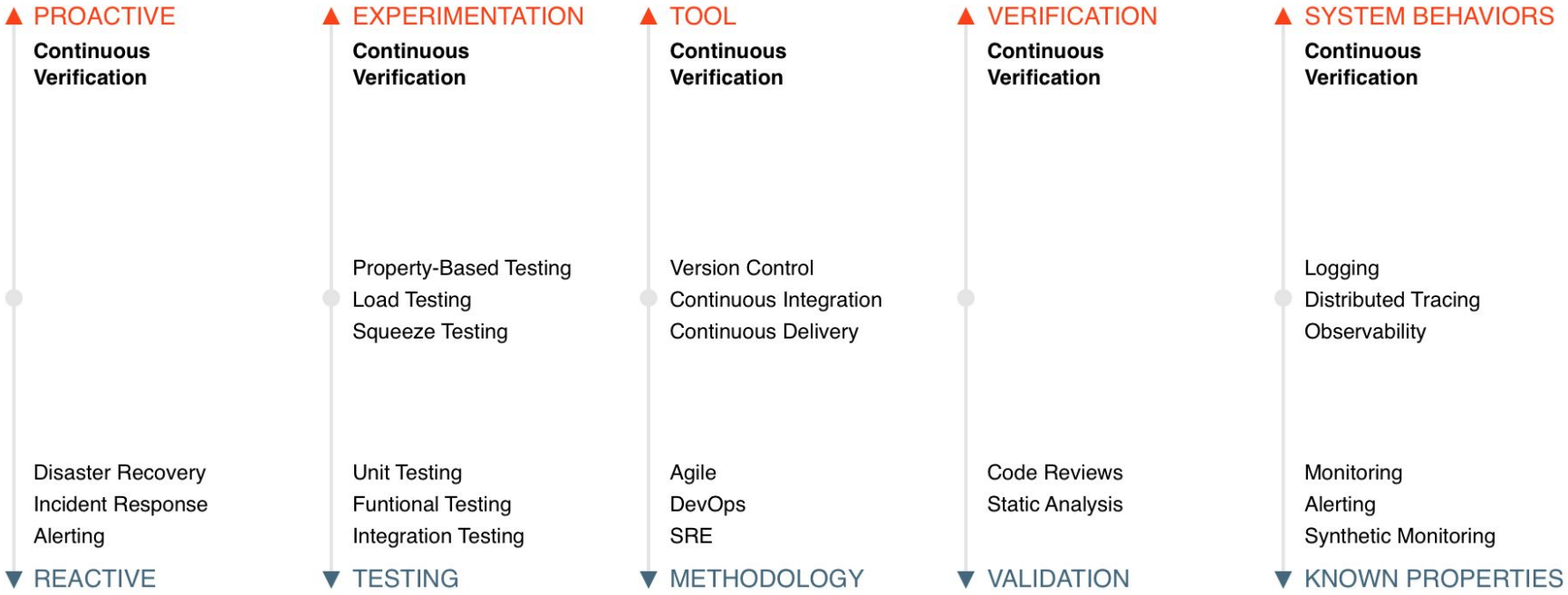
Continuous Delivery builds on the success of CI by automated the steps of preparing code and deploying it to an environment. CD tools allow engineers to choose a build that passed the CI stage and promote that through the pipeline to run in production.

CV

Like CI/CD, **Continuous Verification** is born out of a need to navigate increasingly complex systems. Modern organizations can't validate that the internal machinations of the system work as intended, so instead they verify that the output of the system matches expectations.

Continuous Verification is a proactive, experimentation tool for verifying system behavior.

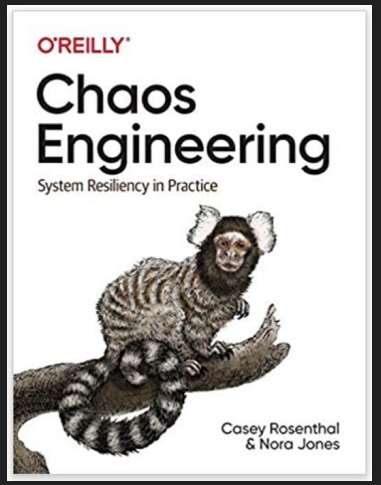
How do other things compare?



Don’t fight complexity.
Navigate it.

@CaseyRosenthal

chaos.community



verica.io

@CaseyRosenthal

- Myth 1: Remove the People Who Cause Accidents
- Myth 2: Document Best Practices and Runbooks
- Myth 3: Defend against Prior Root Causes
- Myth 4: Measure Reliability Quantitatively
- Myth 5: Avoid Risk
- Myth 6: Simplify
- Myth 7: Add Redundancy

@CaseyRosenthal

@caseyrosenthal

@CaseyRosenthal