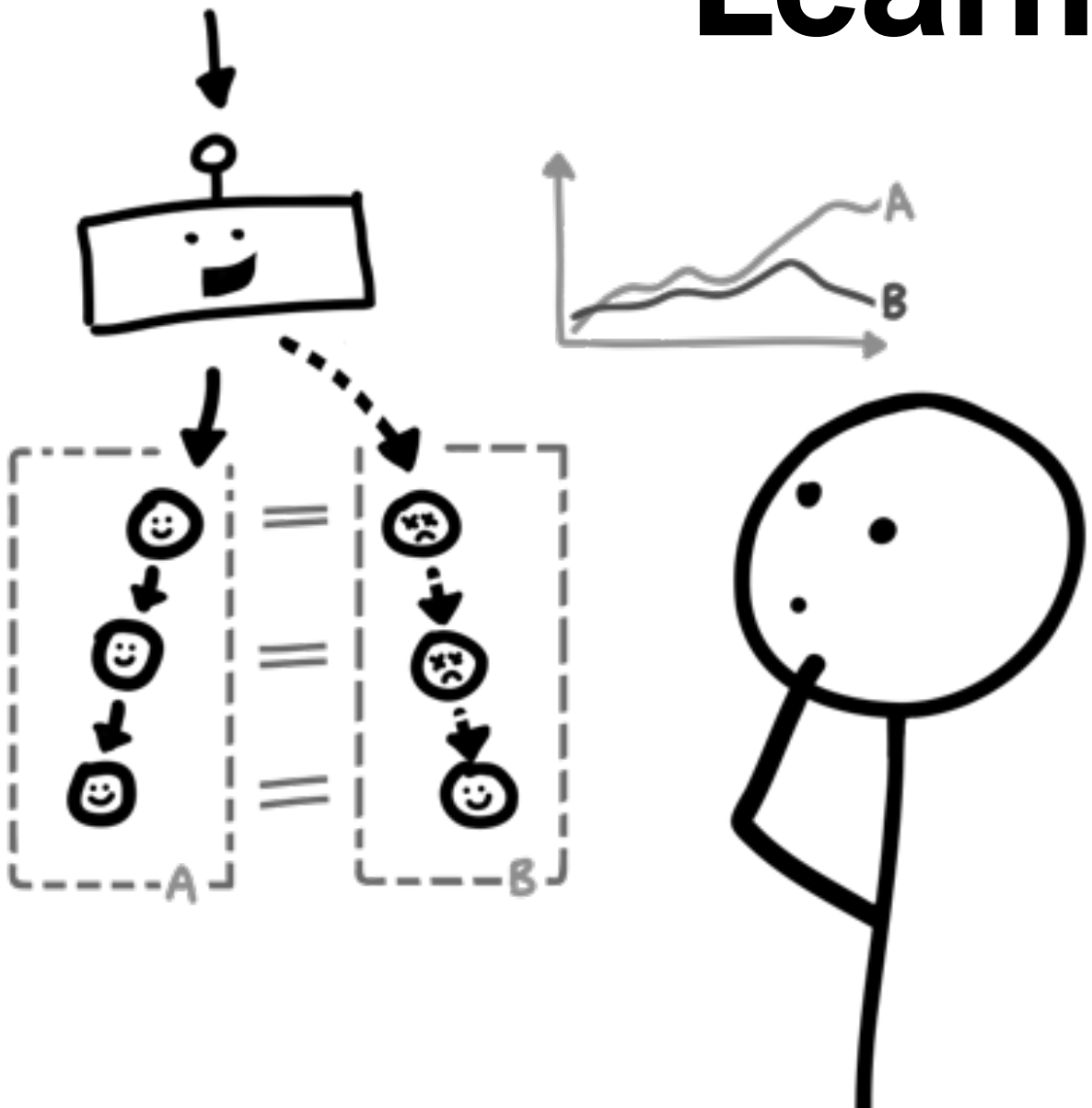
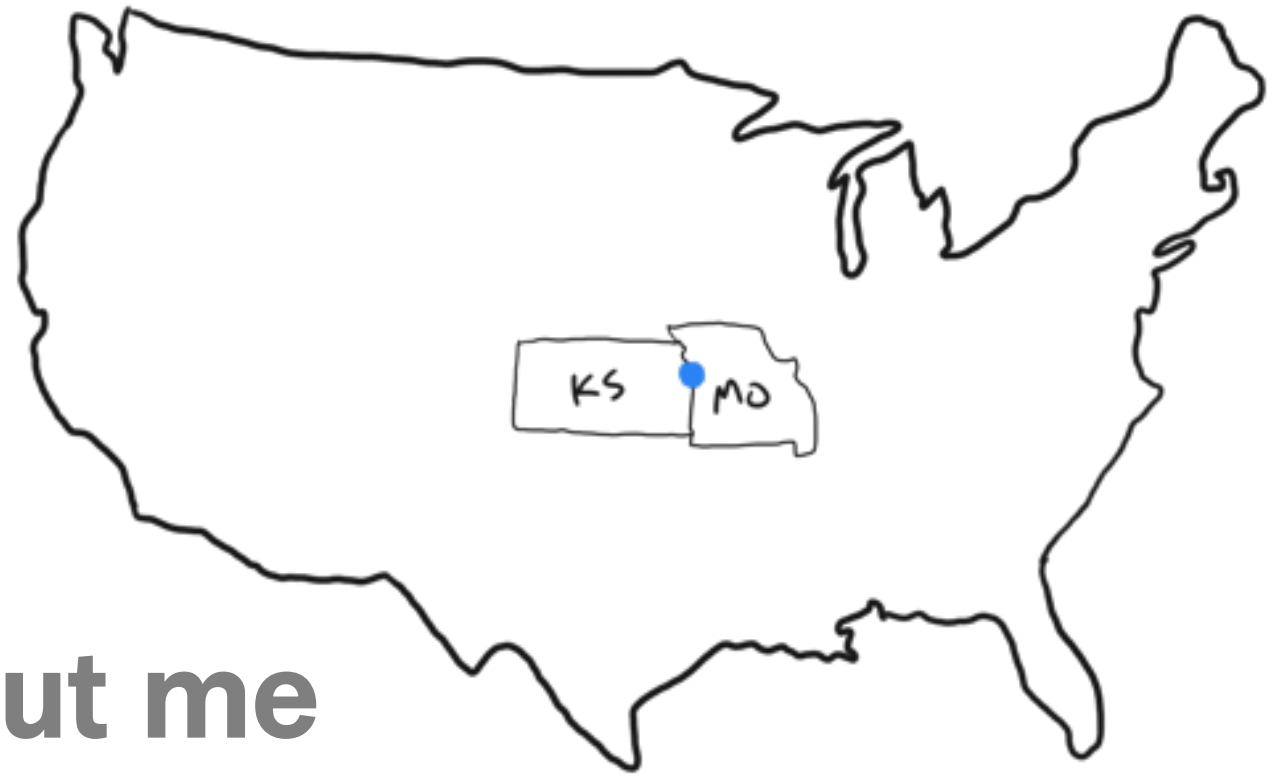
Learn in Production with Traffic Management



Carl Chesser

@che55er | che55er.io

Cerner

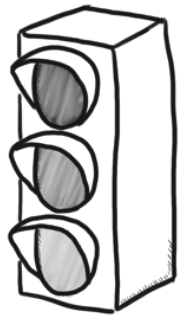


JOE'S KANSAS CITY BAR-B-QUE

About me

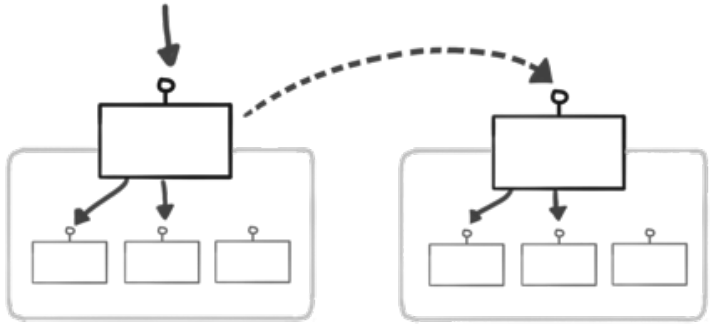
Background

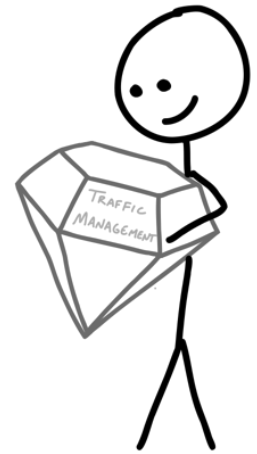
Traffic Management Patterns



Summary

Background





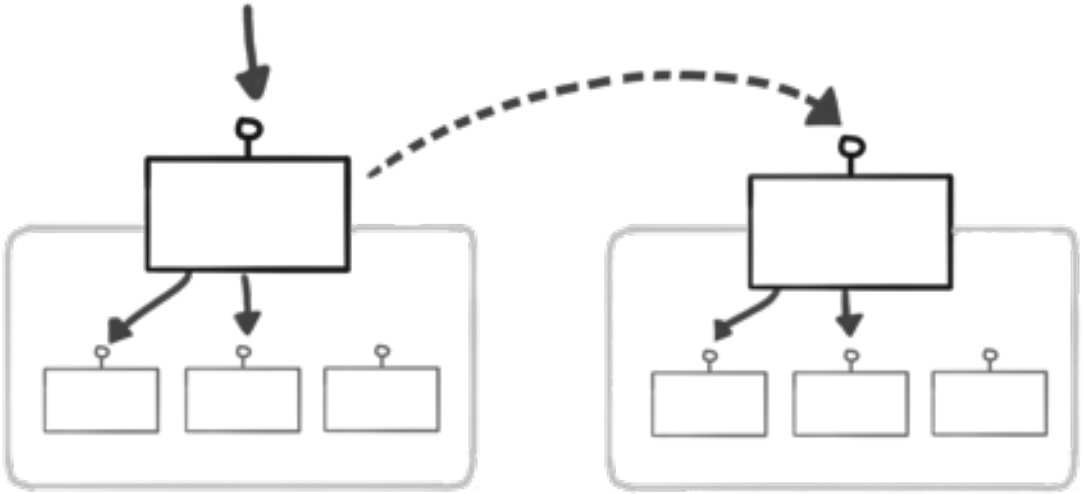
Avoid the Big Bang

We wanted to make the right thing, the easy thing to do.

Evolving platform dependencies can incur larger risks, and we wanted our evolutionary patterns to enable smaller gradual changes.

Build Patterns to Learn in Prod

Find ways to contain or control traffic flows so you can safely learn or experiment in production.



Consider ways to replicate or simulate traffic to ensure live traffic flows are not obstructed.

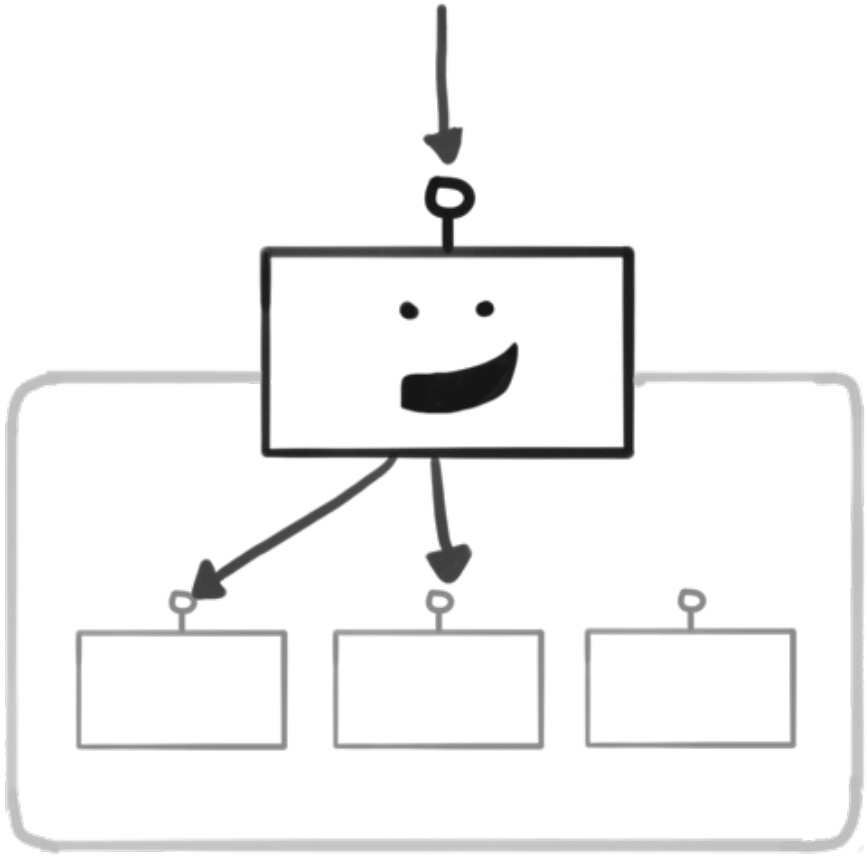
Traffic Management = Asset

We wanted to manage a common set of network patterns to serve as an asset, rather than a liability.

Resulted in a team focus and a set of first-class capabilities.

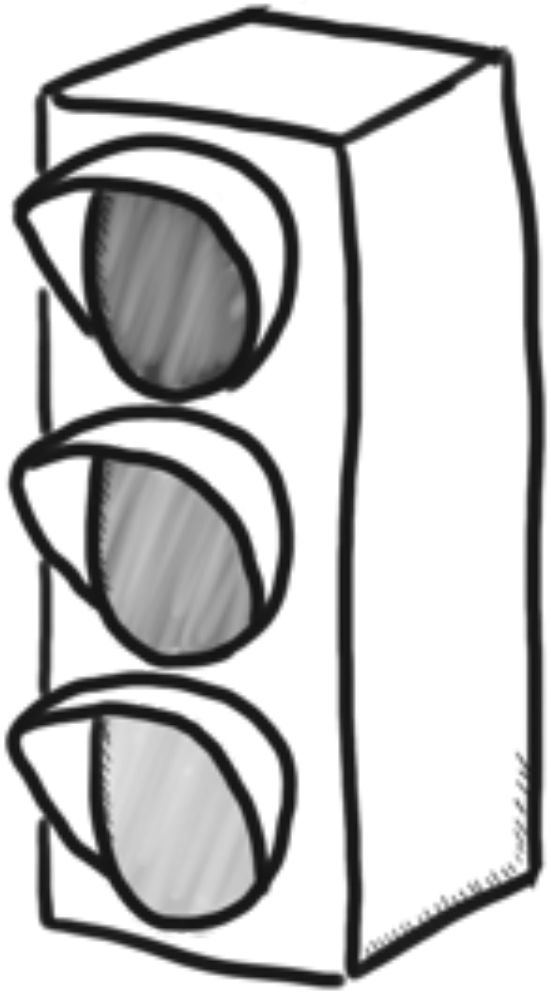
Gateway to Facilitate Change

We evolved our systems many times by leveraging a control gate into our system.



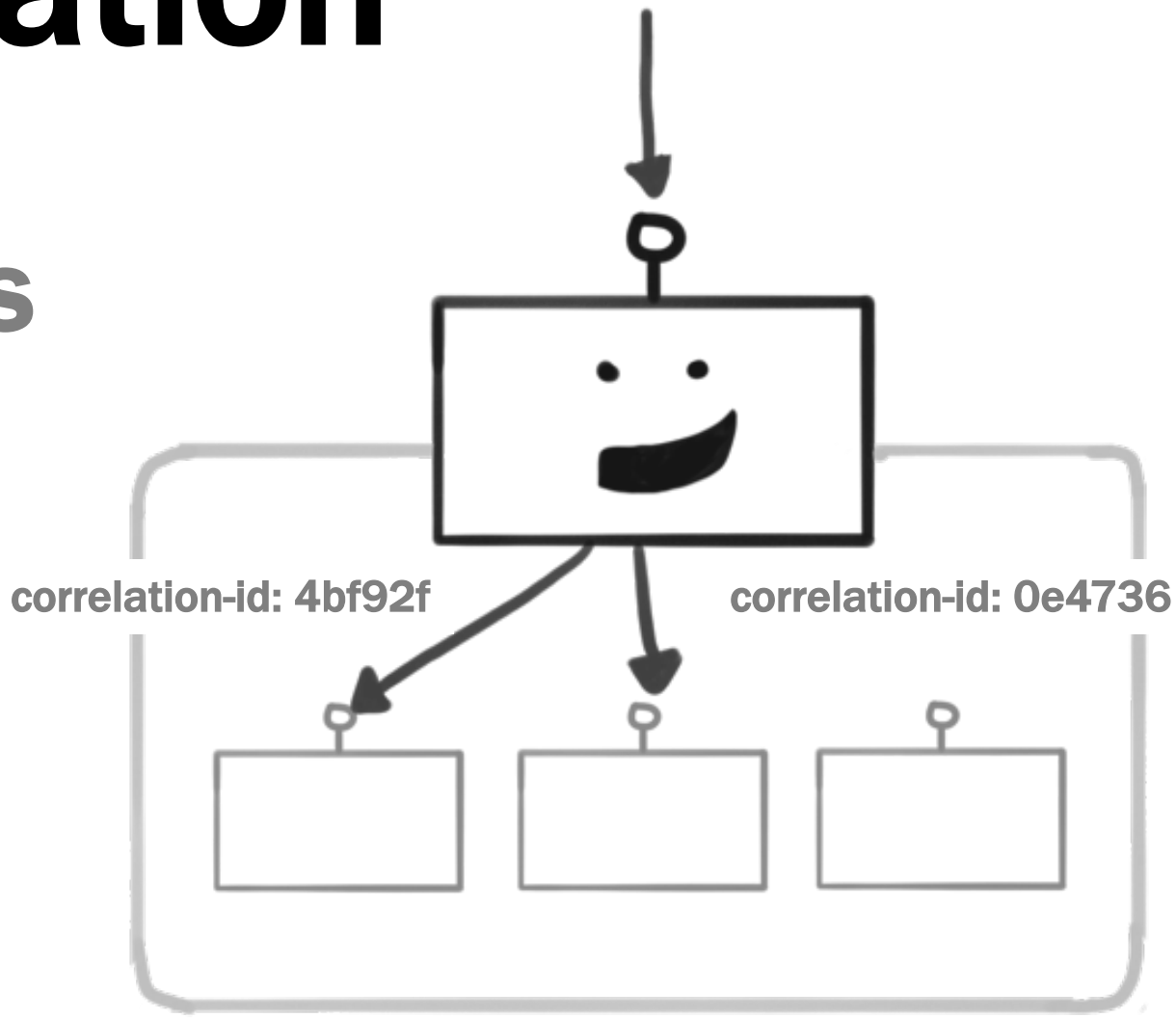
Used as an abstraction of the backing system.

Traffic Management Patterns



Enable Correlation

Establish guarantees that all ingress calls are annotated with correlation IDs.

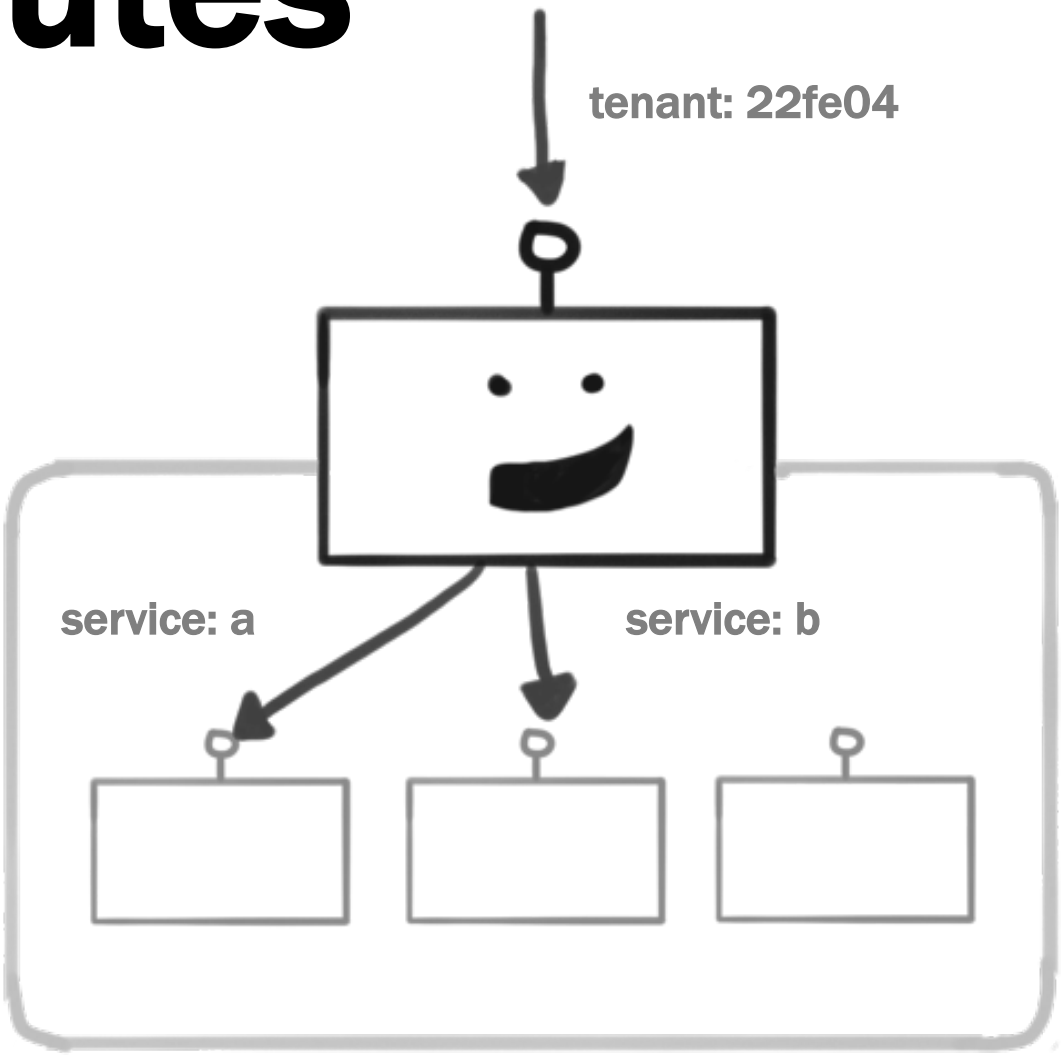


Check out: W3C Trace Context

Routable Attributes

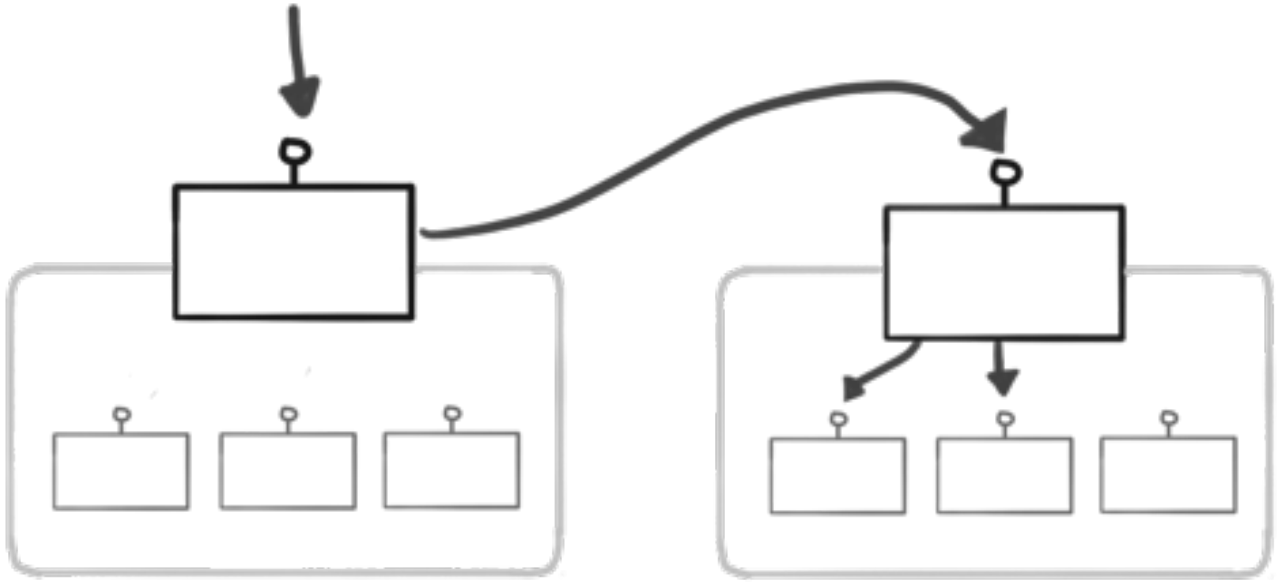
Determine valuable API attributes that will flex your traffic patterns.

Leverage lightweight request attributes that can easily be obtained without calling a remote dependency.



Chaining Traffic

Chaining enables newer use-cases by composing existing API gateway patterns.

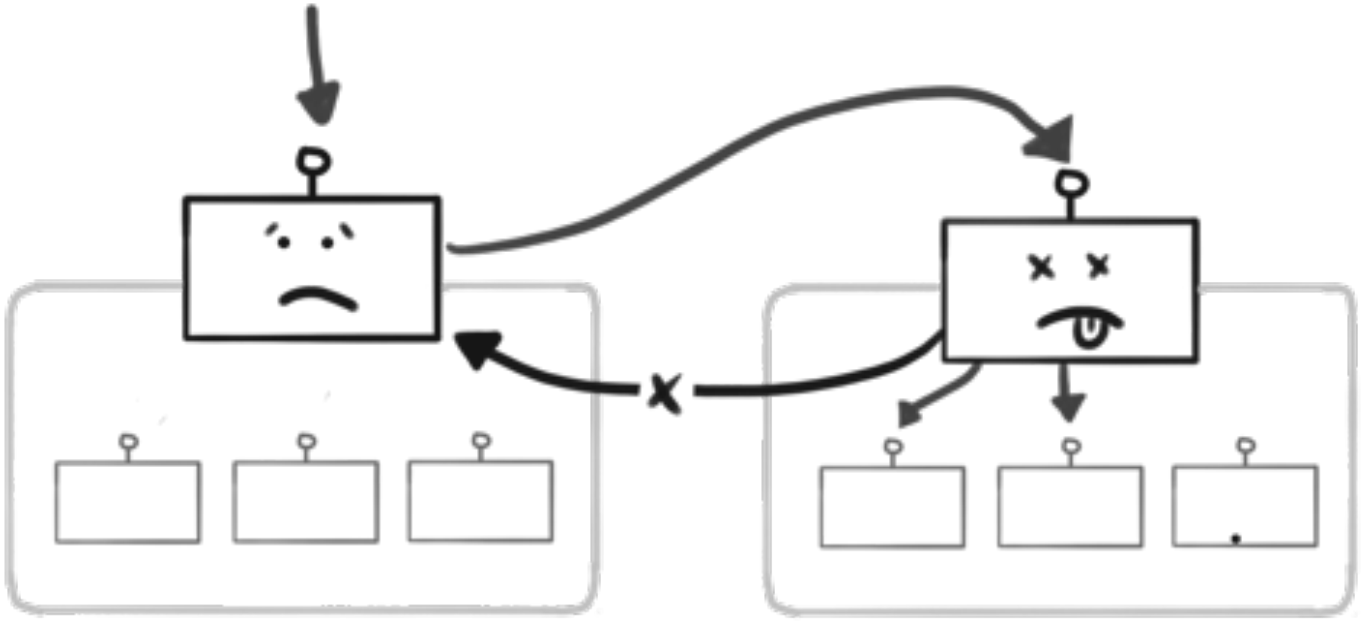


Gateways calling gateways

Bake in Safety Patterns

Encode safety patterns to minimize risks of larger system composition.

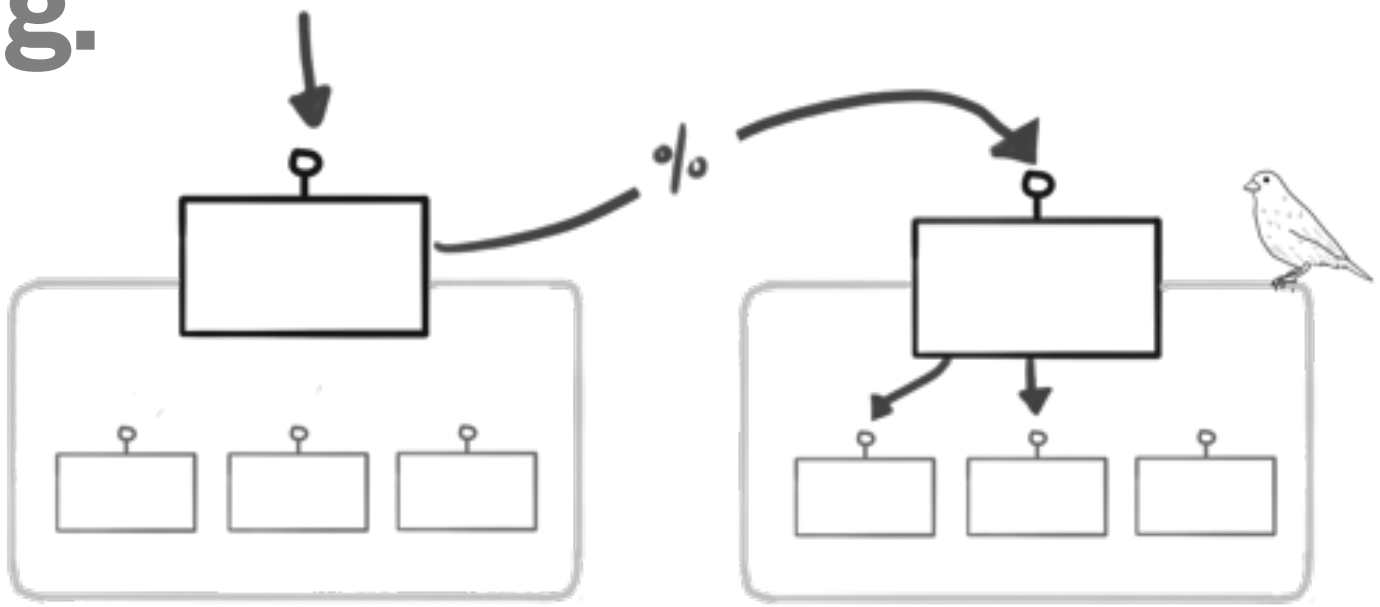
Fail fast when detecting a recursive call chain



Canary Traffic

Supports gradually transitioning a subset of traffic to a different backend by also leveraging chaining.

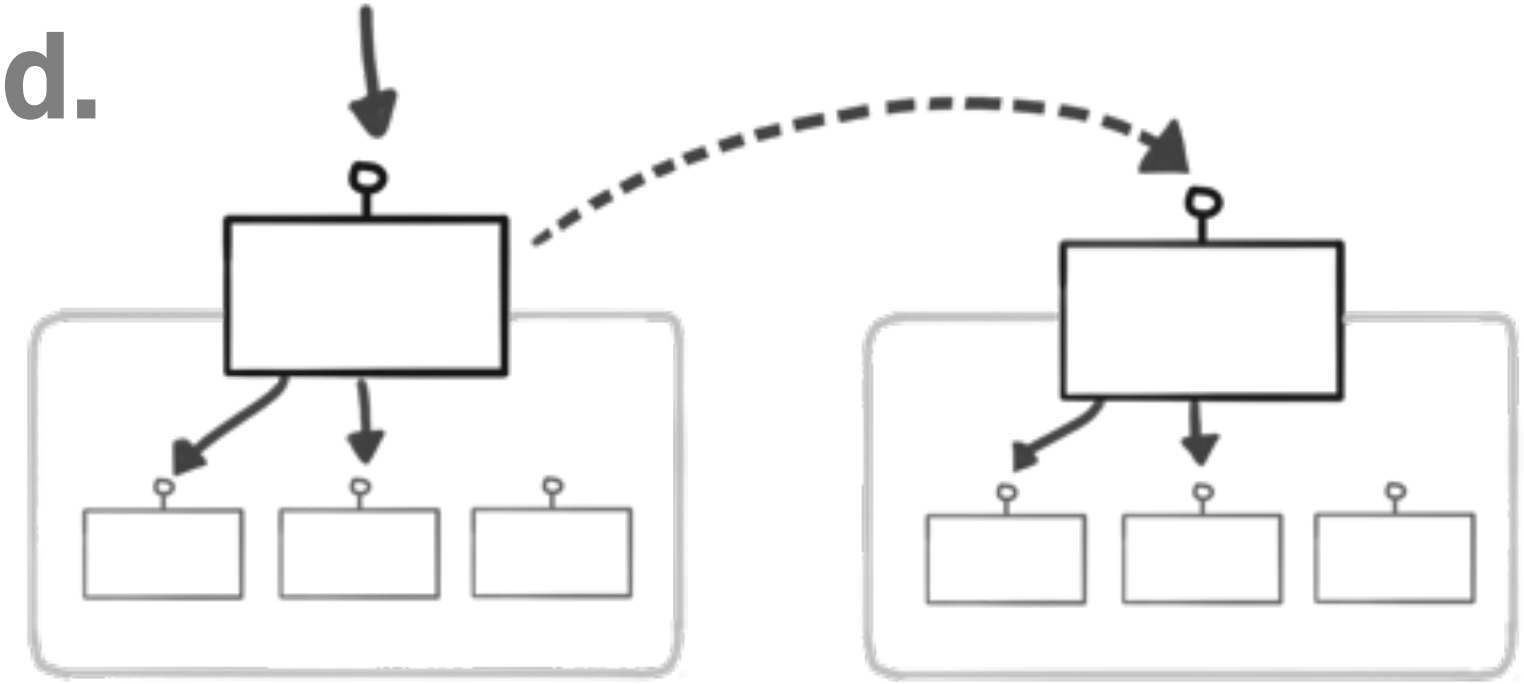
Avoid the Big Bang.



Shadowing Traffic

Replays a percentage of traffic to another backend.

Background replay of *safe* requests.
(read-only, HTTP GET)

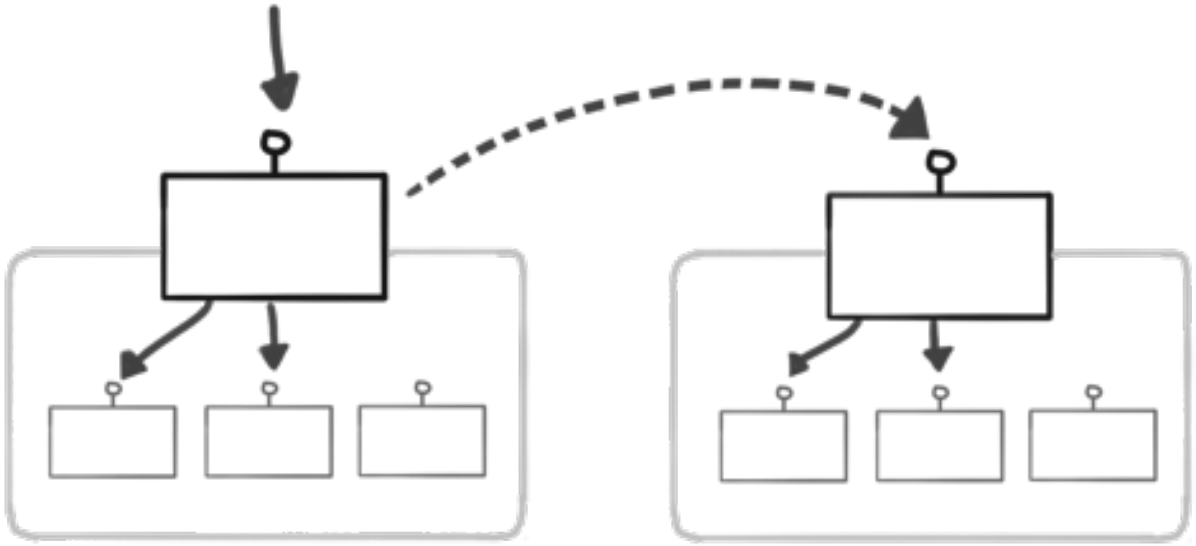


Build in a bulkhead for your resource pool supporting the replay of traffic to avoid unnecessary stress on your service at bursts of traffic.

Shadow Allows Early Testing

Rather than imposing a canary early with experiments, look to leverage a shadow of traffic.

Helps when a potential small percentage of failure still introduces undesirable risk.



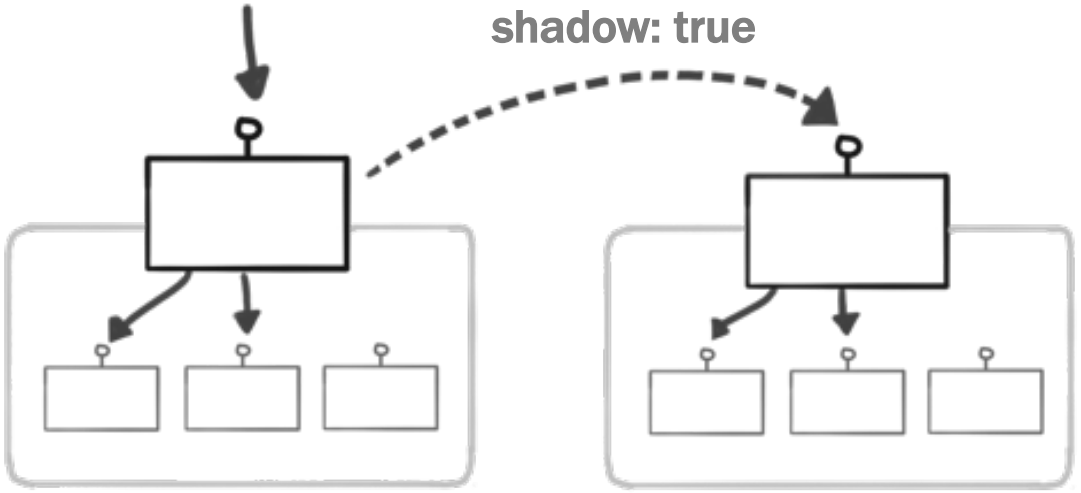
Comparing Systems

Leverage correlation to accurately compare subsets of traffic between two systems.

Canary or shadow would only be applied to a subset; therefore, you would only want to compare against this selected traffic.

Annotating Traffic

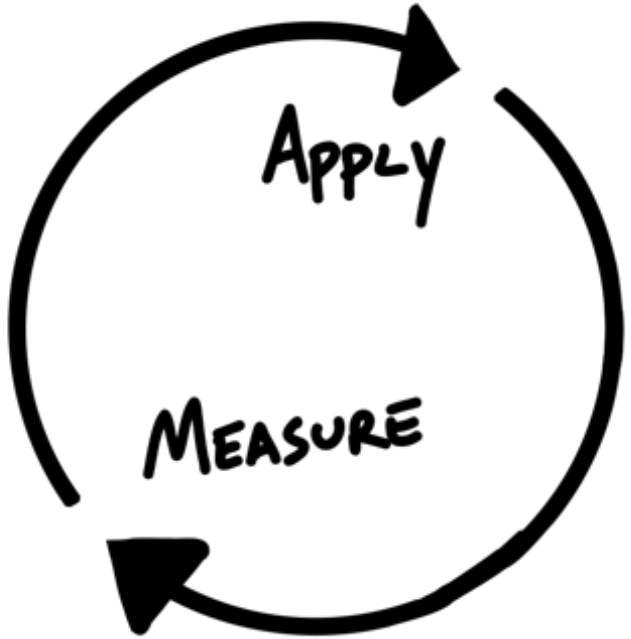
Make it easy to discern shadow requests from live requests in your telemetry data.



Simplifies when observing the live system of traffic and determining the impact of failures.

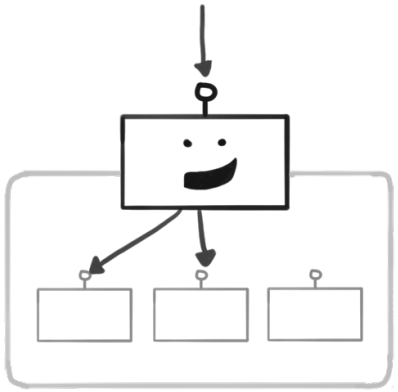
Learn and Iterate

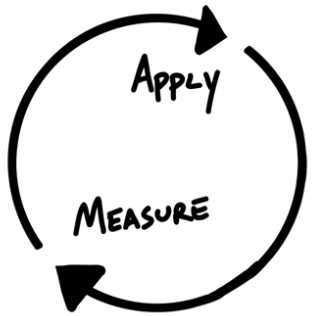
Establish a routine of gradually applying traffic patterns, measure, and repeat.



Recognize shadow traffic can be leveraged early in the process to learn, then applying gradual increments with canary.

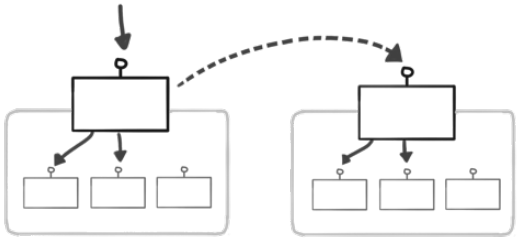
Summary



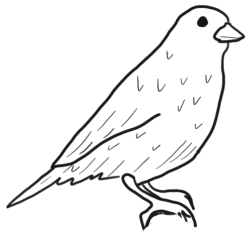


Identify ways to reduce how change is introduced.

Leverage shadow traffic early in your process for learning.

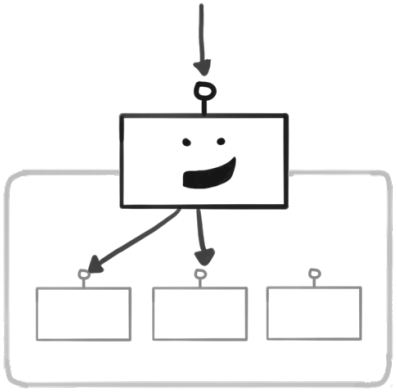


Consider traffic management as an asset.

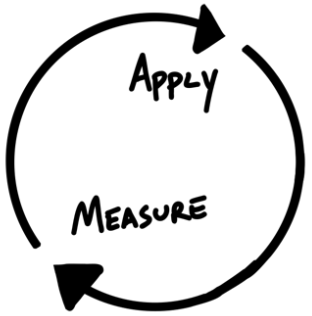


Apply canary traffic to gradually introduce changes (incremental % of increase).

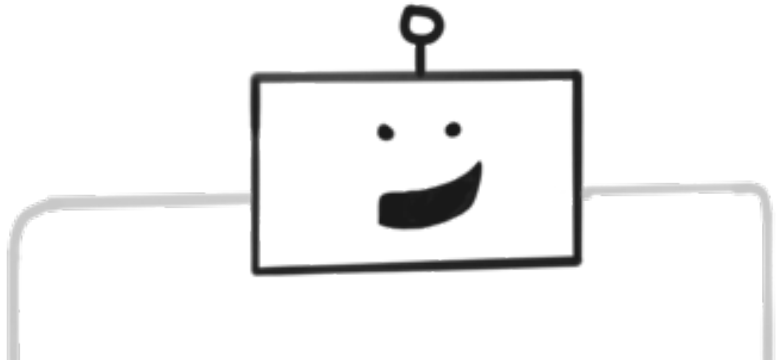
Guarantee common points of instrumentation and correlation.



Establish a feedback loop of applying traffic management patterns based on measurements.



Thank you!



Carl Chesser

@che55er | che55er.io