

Spring Cloud 2024 Fundamentals

Introduction to Spring Cloud Fundamentals



Mohamed Echout

Senior Technical Trainer

@MEchout

Overview of Spring Cloud Fundamentals



Meet Anna – The Founder of Caffeine Power



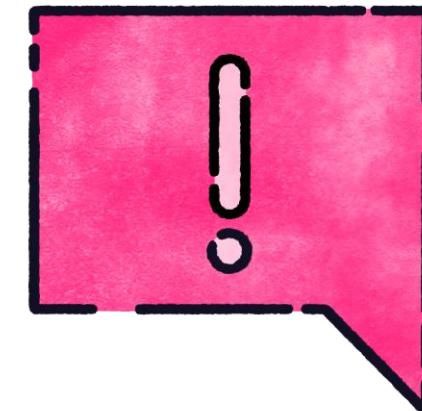
Anna

**Passionate about
coffee and technology**



Caffeine Power

**Tech-driven coffee
shop chain**

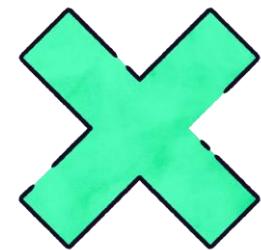


Problem

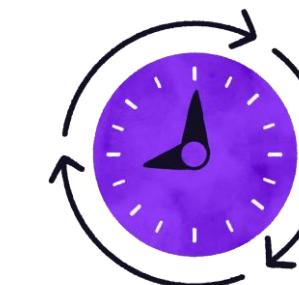
**Struggle to handle high
demand**



The Growing Pains of Caffeine Power



Orders getting lost



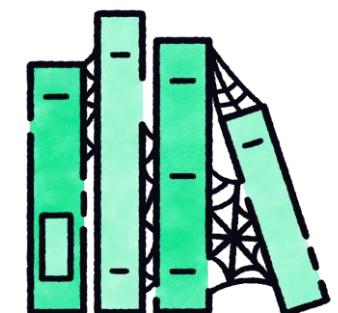
Store takes weeks to integrate



Crashes during busy hours



Failed payments

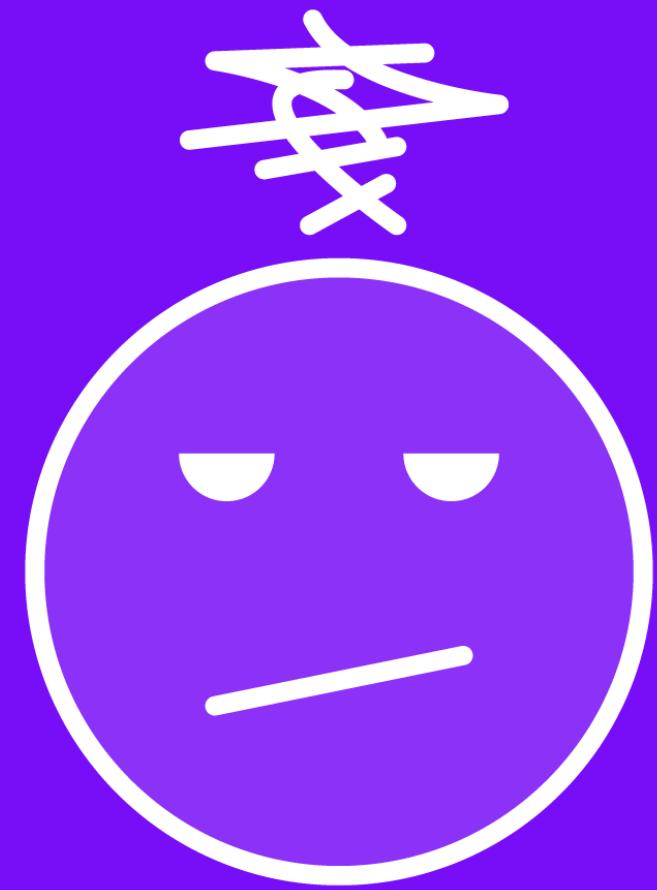


Inaccurate inventory data



Manually confirming orders



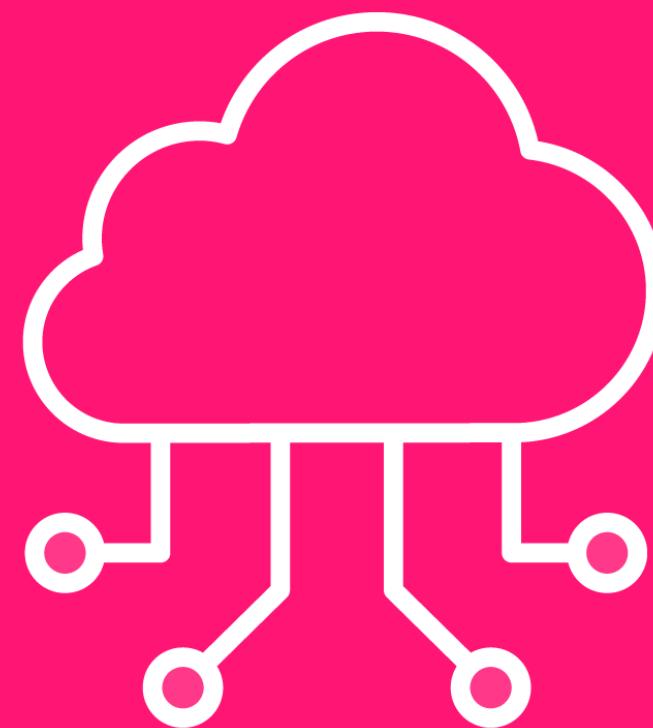


Anna is trapped in tech problems!



**This is where Spring Cloud
comes to the rescue!**



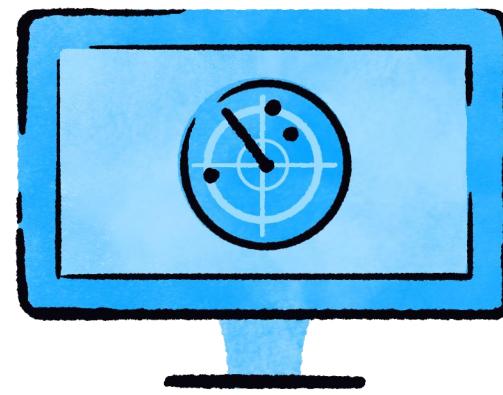


Spring Cloud

A tool that helps big apps work smoothly by making different parts talk to each other, stay updated, and keep running even when lots of people use them.



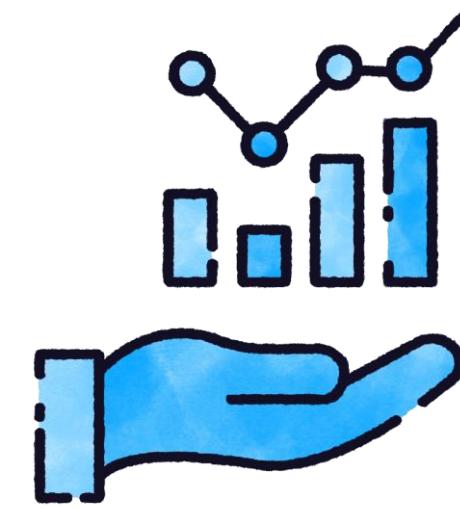
Spring Cloud Key Components



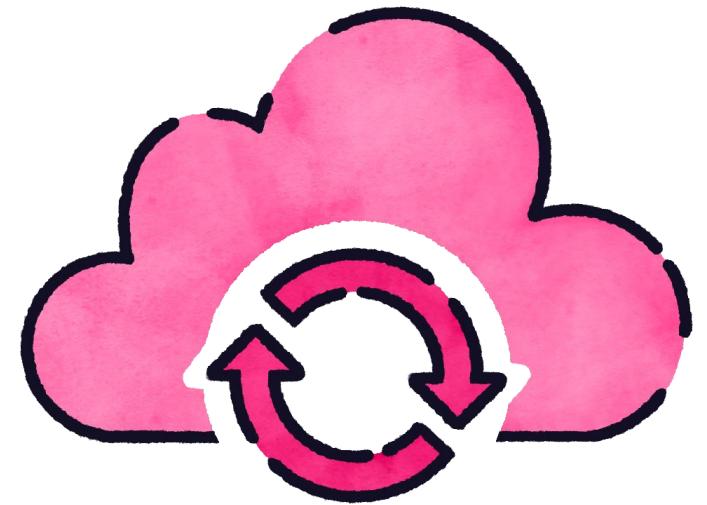
Eureka



Spring Cloud
Config



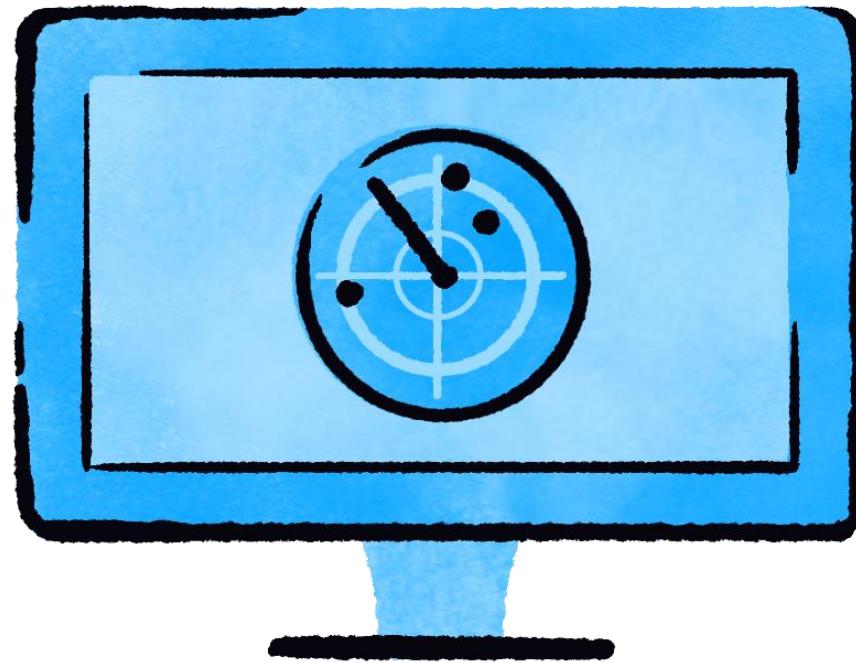
Resilience4j



Spring Cloud
Gateway



Service Discovery Eureka



No more lost orders!
The system always knows where to send them



Config Management Spring Cloud Config

**Every store gets real-time updates on prices,
stock, and promotions**



Crash Prevention Resilience4j

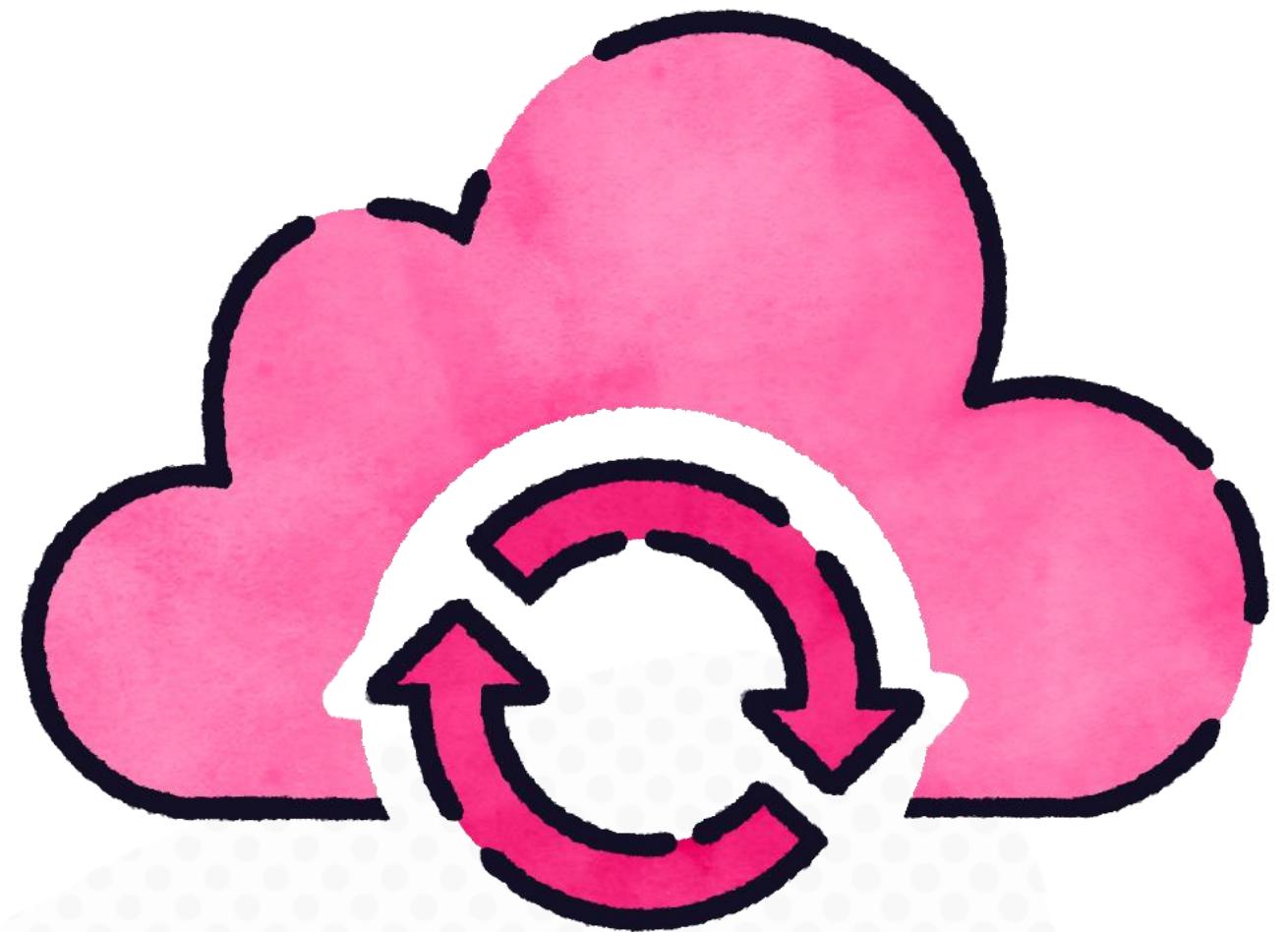


If a part of the system fails, Spring Cloud steps in to keep Caffeine Power running



Traffic Control Gateway

No more app crashes; the system balances the load and keeps things running smoothly





Anna is free! What's next?





Service Discovery and Resilience

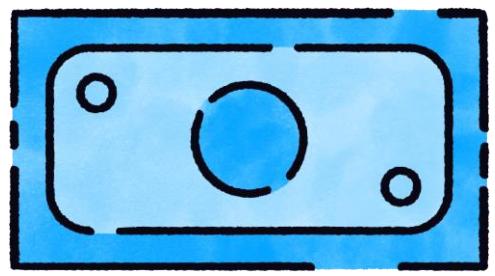




Why service discovery matters?



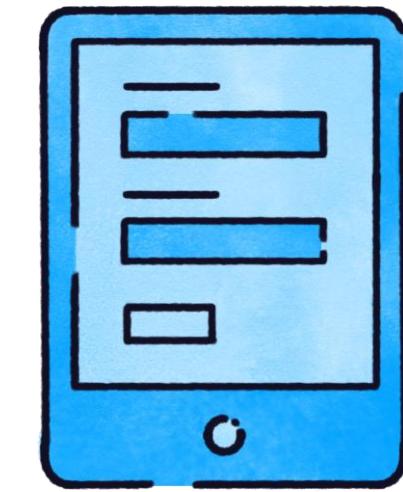
The Problem Without It



Payment



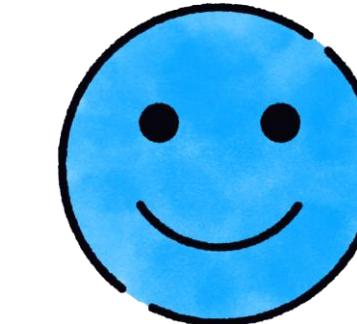
Inventory



Authentication



Order



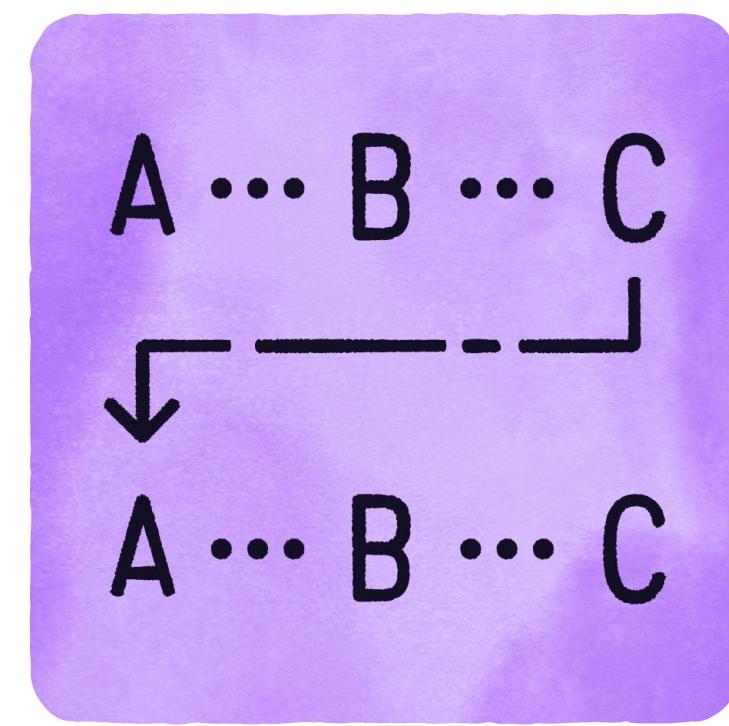
Reviews



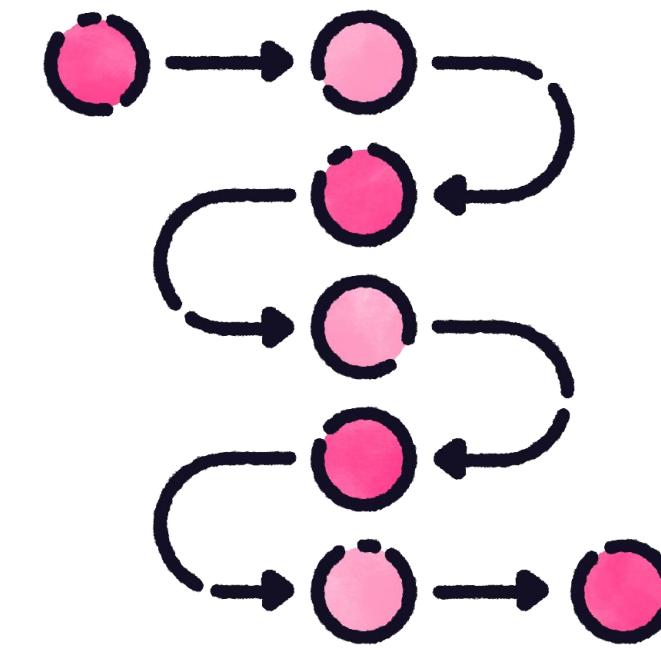
Solution: Service Discovery!



Automatically
registers all services



Keeps track of where
they are

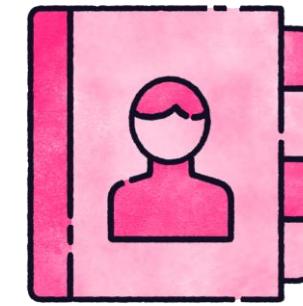


Ensures communication
between them

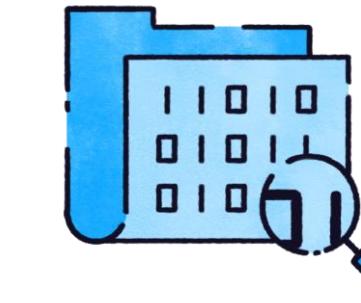


Meet Eureka – The Address Book for Services!

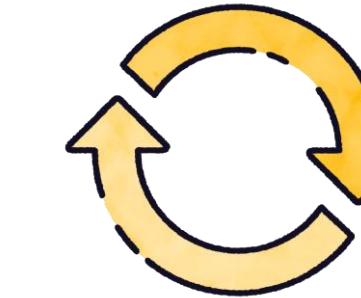
Eureka



Find services easily



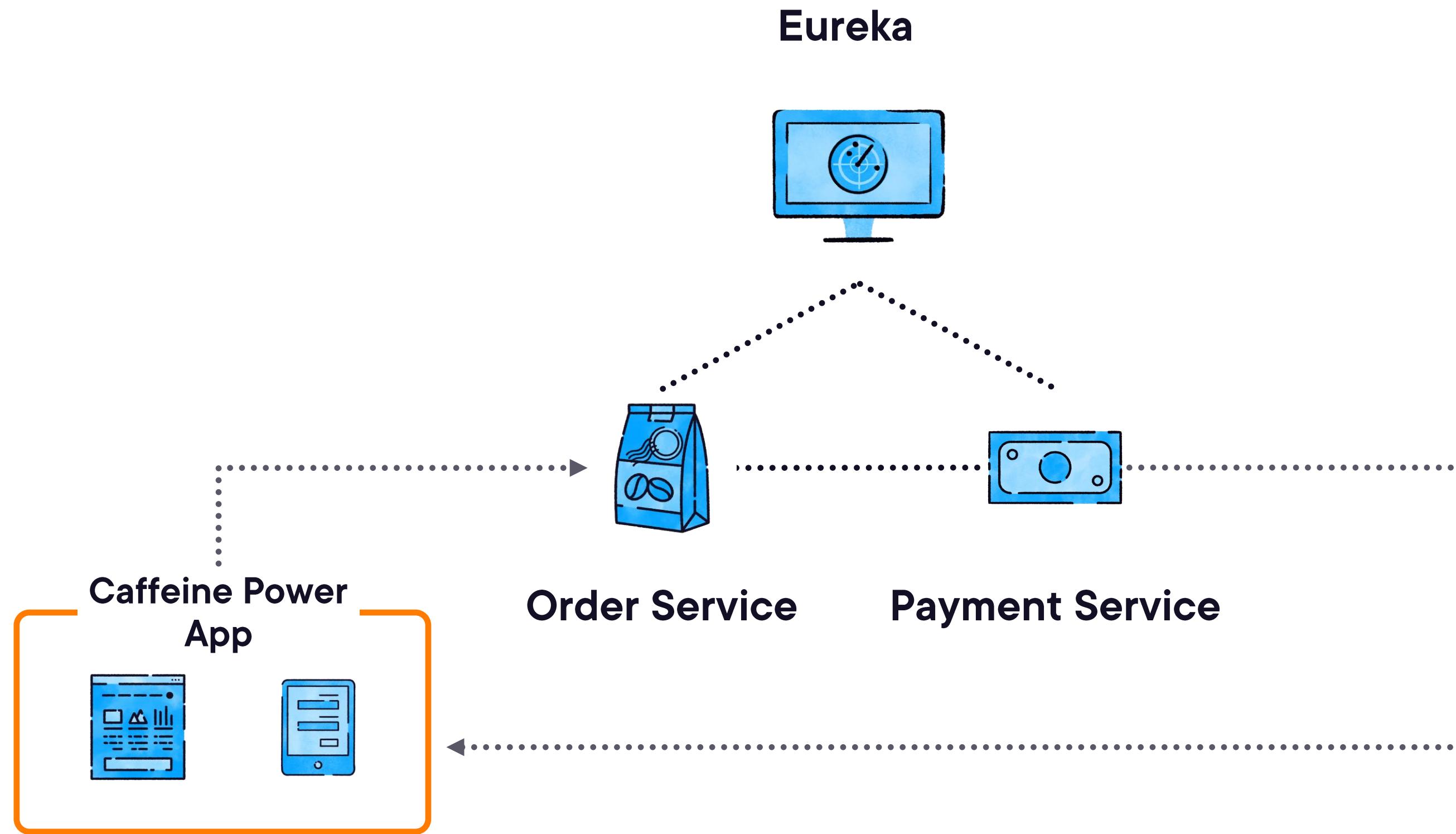
Automated updates



Automated tracking



How Eureka Keeps Caffeine Power Running

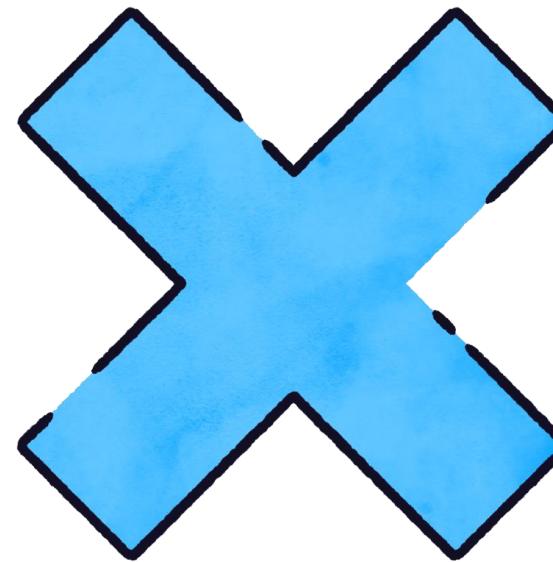




What happens when a service fails?

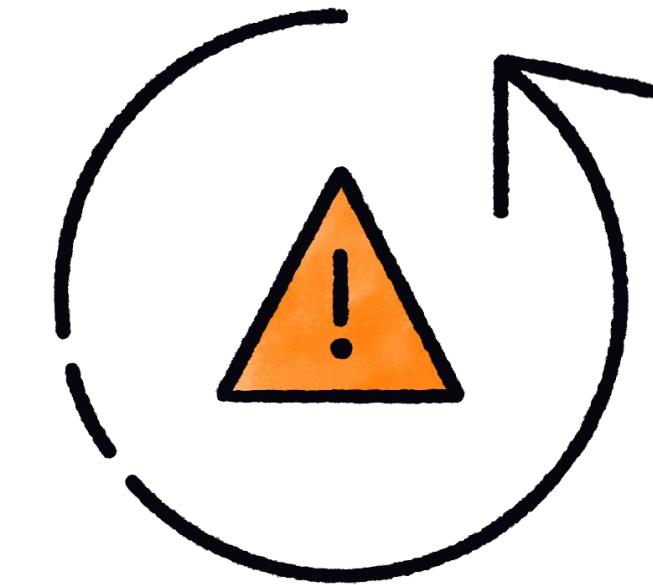


The Reality of Microservices



Service goes down

**Network issues, bugs, or traffic
overload**

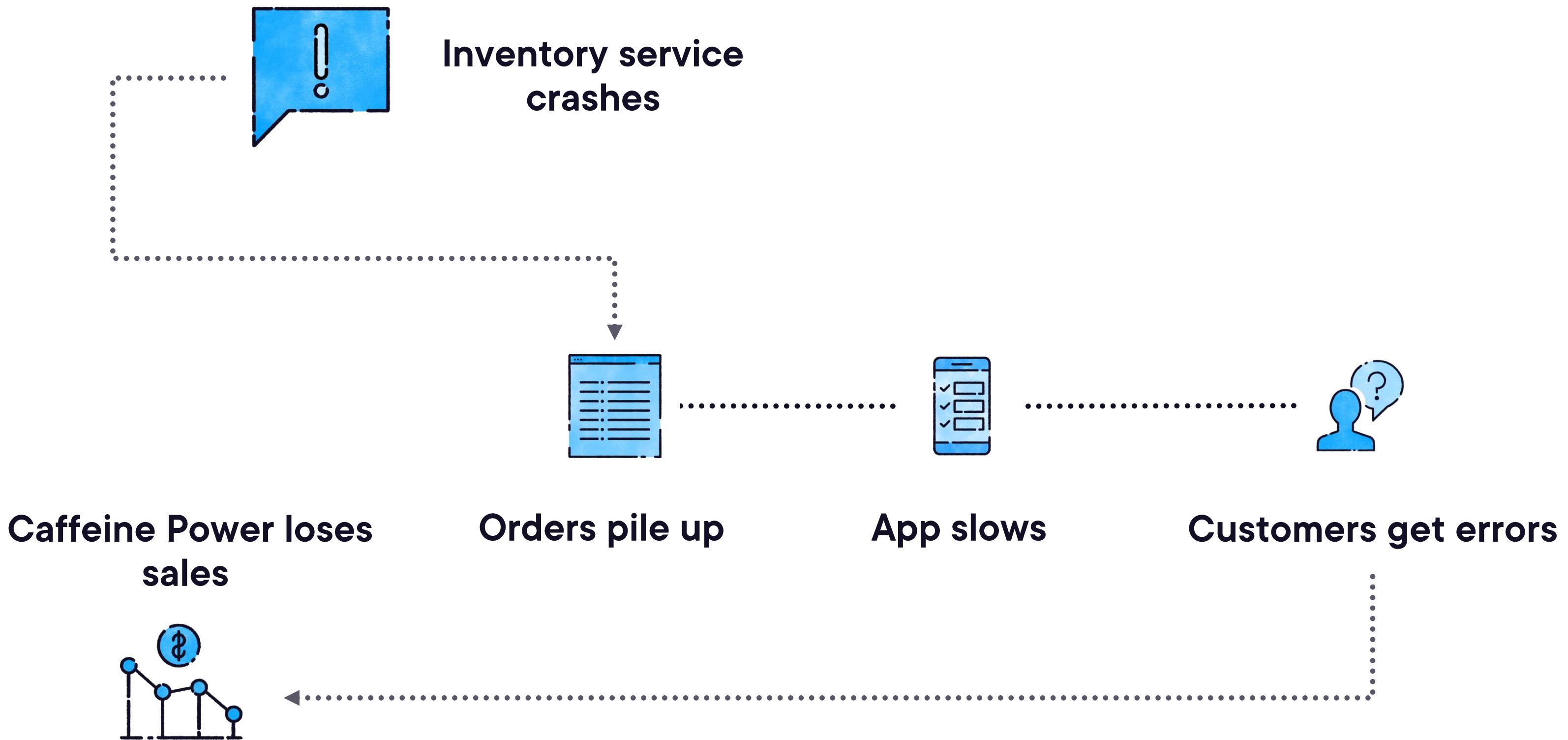


Without a fallback

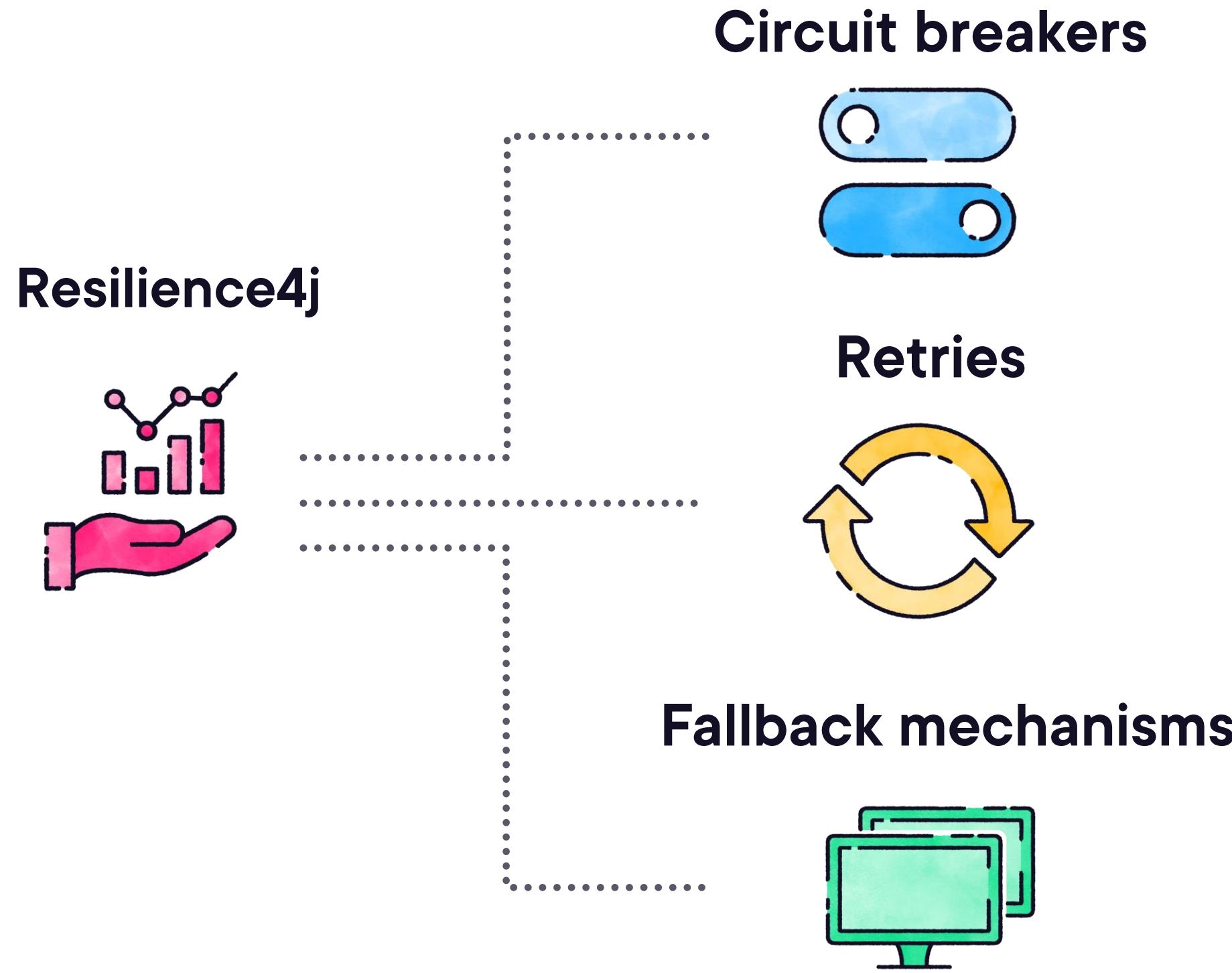
**Other services keep trying to reach
it, causing delays and failures**



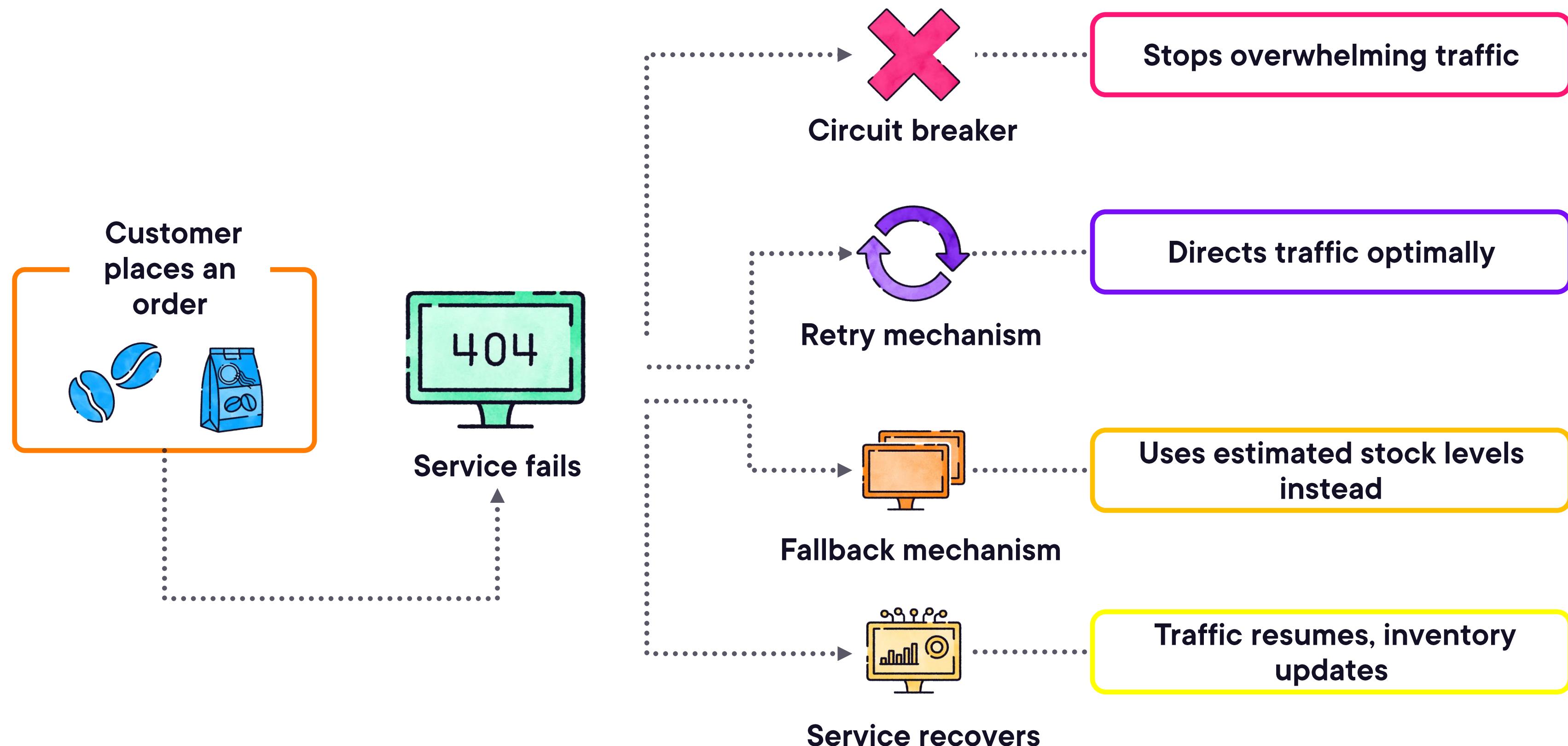
What Happens When a Service Fails in Caffeine Power?



Solution: Resilience with Resilience4j!



How Resilience4j Keeps Caffeine Power Running



Before vs. After: Service Discovery and Resilience Fixed Caffeine Power

Before

vs.

After

Services couldn't find each other

Orders get lost

App crashes during rush

Broken recommendations

Scaling takes weeks

Eureka connects services dynamically

Orders always reach the right service

Resilience4j prevents overload

Fallbacks suggest alternatives

New stores integrate instantly



Demo: Setting Up Spring Cloud Dependencies

