



KubeCon



CloudNativeCon

---

Europe 2022

---

WELCOME TO VALENCIA





KubeCon



CloudNativeCon

Europe 2022

# Making Sense of Chaos

*Implementing Chaos Engineering in a Fintech Company*

Iqbal Farabi & Giovanni Sakti  
Payments Platform (GoPay), GoToFinancial



# Making Sense of Chaos

*Implementing Chaos Engineering in a Fintech Company*

Hola!



**Iqbal Farabi**  
Engineering Manager  
*GoToFinancial*  
(@iqbal\_farabi)



**Giovanni Sakti**  
Sr. Engineering Manager  
*GoToFinancial*  
(@giosakti)





KubeCon



CloudNativeCon

Europe 2022

# Case Study

goto financial

goto



gojek

goto financial



tokopedia





KubeCon



CloudNativeCon

Europe 2022

# Case Study

# goto financial

 **gopay**

 **gopaylater**  
By Findaya

 **midtrans**

 **gobiz**

 **gobiz PLUS**

 **moka**

 **gostore**

 **gomodal**

 **selly**





KubeCon



CloudNativeCon

Europe 2022

# Case Study

# goto financial

 **gopay**

 **gopaylater**  
By Findaya

 **midtrans**

 **gobiz**

 **gobiz PLUS**

 **moka**

 **gostore**

 **gomodal**

 **selly**

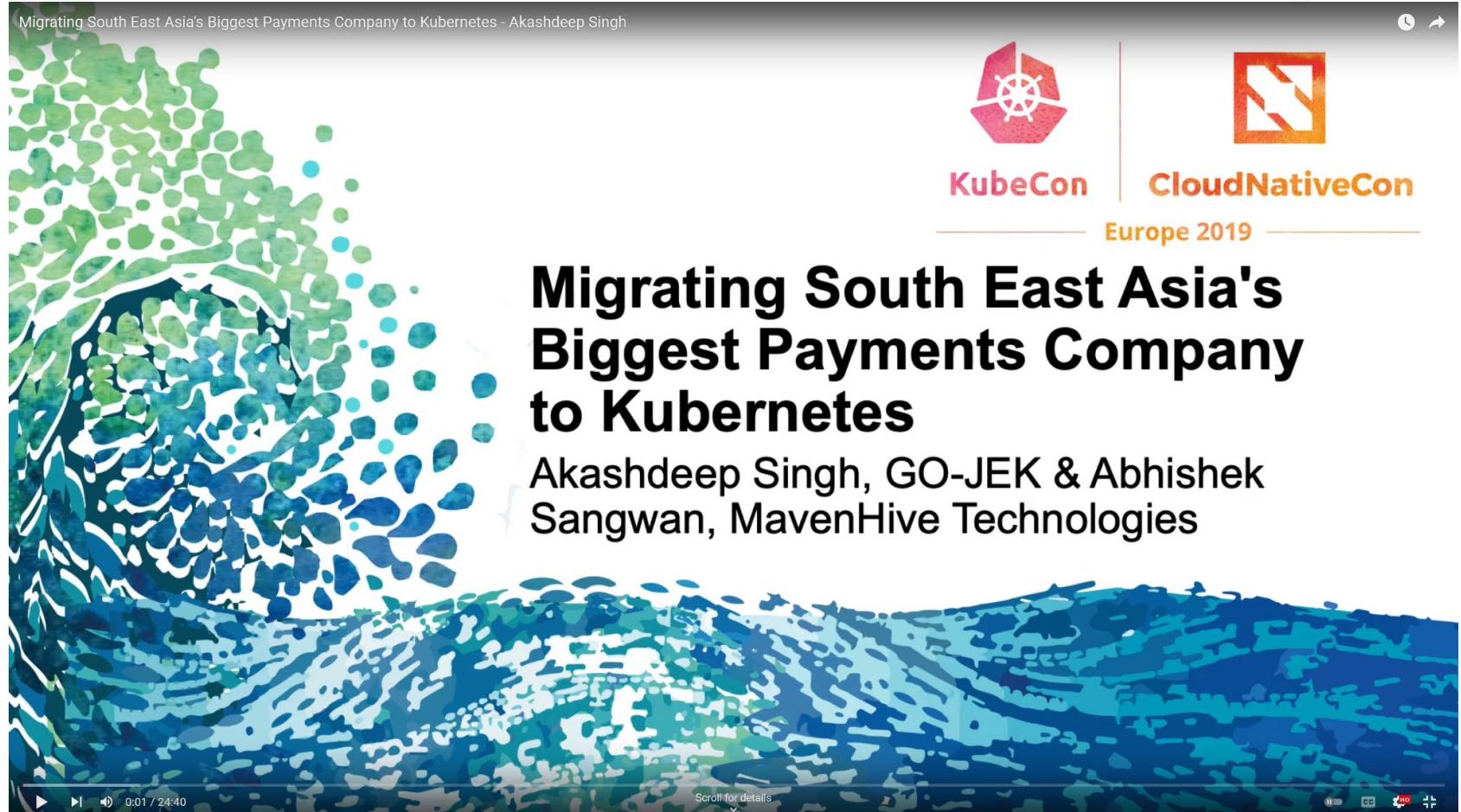


- **Payments** solution and **personal finance** management for individuals. First product launched *circa 2016*
- Serving **offline** (e.g. QR code) and **online** (e.g. e-commerce and on demand services) payments in the order of **tens of millions** active customers.
- Operating in **Southeast Asia**, primarily in **Indonesia**.

- **300+** Engineers (*and growing rapidly*)
- **500+** Services
- **30+** Kubernetes Clusters (*in multiple environments*)
- Multi-cloud & On-prem

# Kubernetes Migration

(2018-2020)





KubeCon



CloudNativeCon

Europe 2022

# Istio Migration (2020-2022)

Migrating Transactions Worth Billions of \$ to Service Mesh With N... Mahendra Kariya &amp; Shishir Joshi



KubeCon



CloudNativeCon

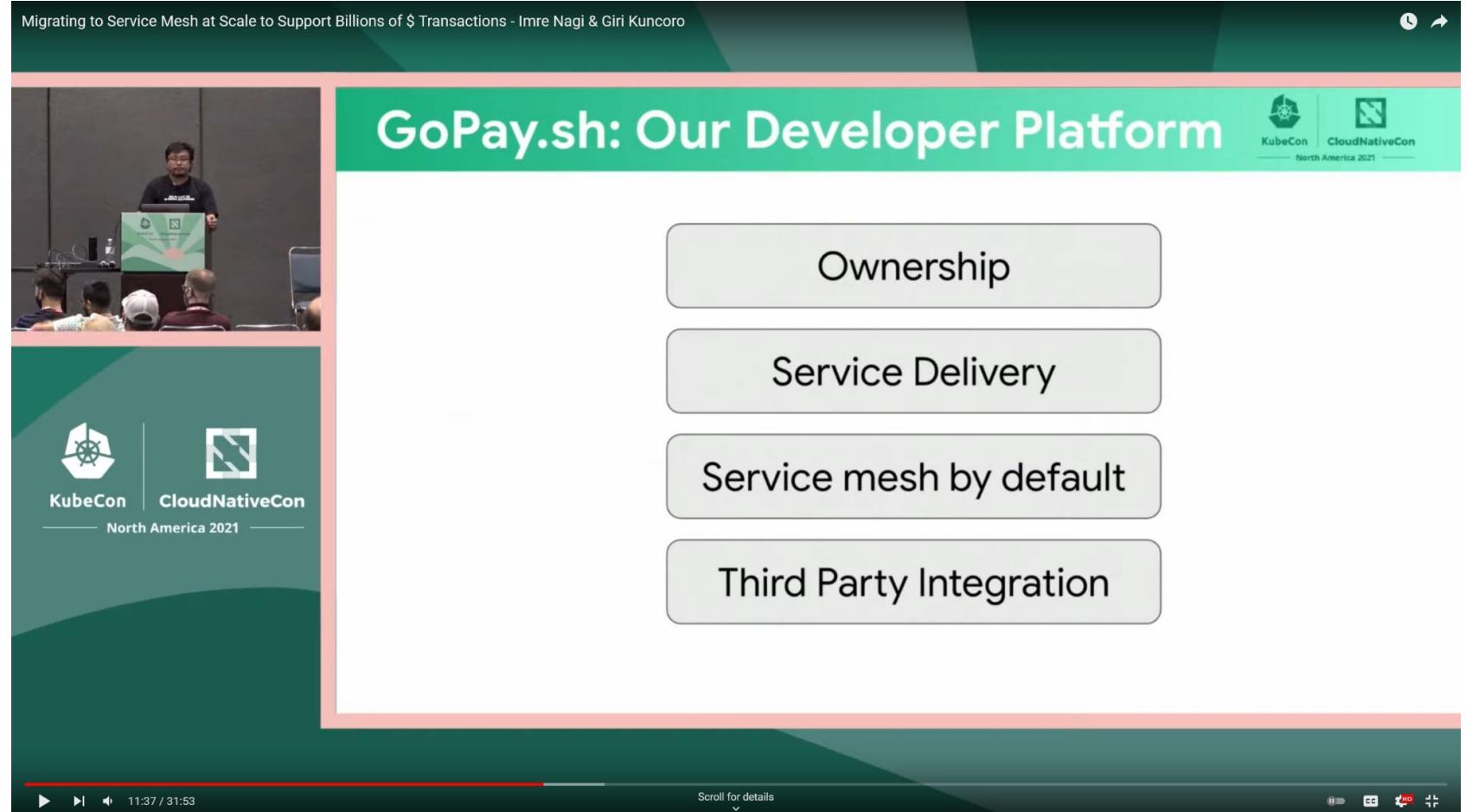
Europe 2020

*Virtual*

## Migrating Transactions Worth Billions of \$ to Service Mesh With No Downtime

*Mahendra Kariya & Shishir Joshi, Gojek*

# Developer Portal **gopay.sh** (2021-2022)



Migrating to Service Mesh at Scale to Support Billions of \$ Transactions - Imre Nagi & Giri Kuncoro

GoPay.sh: Our Developer Platform

- Ownership
- Service Delivery
- Service mesh by default
- Third Party Integration

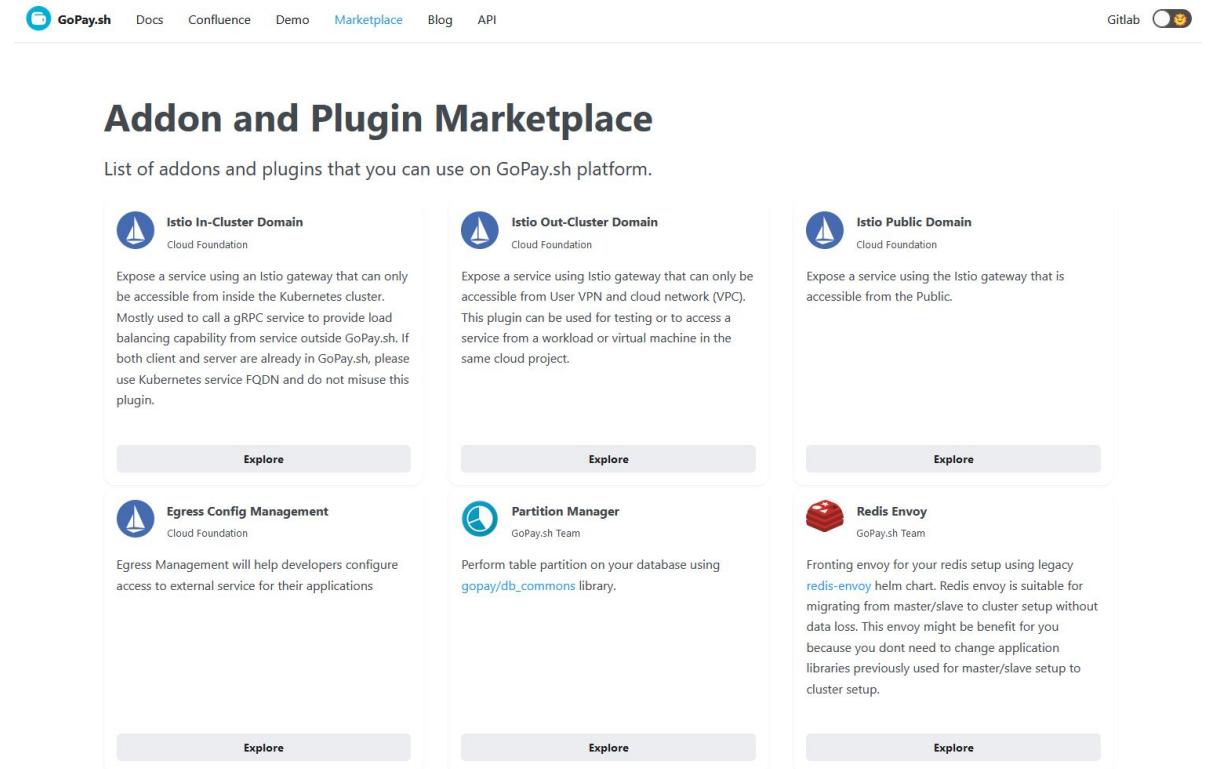
11:37 / 31:53

Scroll for details

# Developer Portal (gopay.sh) rollout

80% of services already onboarded

Adopted Open Service Broker API standard for addons integration



The screenshot shows the GoPay.sh developer portal with the following layout:

- Header:** GoPay.sh, Docs, Confluence, Demo, Marketplace, Blog, API, Gitlab, and a toggle switch.
- Section Title:** Addon and Plugin Marketplace
- Section Description:** List of addons and plugins that you can use on GoPay.sh platform.
- Grid of Marketplace Items:** Six items are displayed in two rows of three:
  - Istio In-Cluster Domain** (Cloud Foundation): Expose a service using an Istio gateway that can only be accessible from inside the Kubernetes cluster. Mostly used to call a gRPC service to provide load balancing capability from service outside GoPay.sh. If both client and server are already in GoPay.sh, please use Kubernetes service FQDN and do not misuse this plugin.  
[Explore](#)
  - Istio Out-Cluster Domain** (Cloud Foundation): Expose a service using Istio gateway that can only be accessible from User VPN and cloud network (VPC). This plugin can be used for testing or to access a service from a workload or virtual machine in the same cloud project.  
[Explore](#)
  - Istio Public Domain** (Cloud Foundation): Expose a service using the Istio gateway that is accessible from the Public.  
[Explore](#)
- Egress Config Management** (Cloud Foundation): Egress Management will help developers configure access to external service for their applications.  
[Explore](#)
- Partition Manager** (GoPay.sh Team): Perform table partition on your database using `gopay/db_commons` library.  
[Explore](#)
- Redis Envoy** (GoPay.sh Team): Fronting envoy for your redis setup using legacy `redis-envoy` helm chart. Redis envoy is suitable for migrating from master/slave to cluster setup without data loss. This envoy might be benefit for you because you dont need to change application libraries previously used for master/slave setup to cluster setup.  
[Explore](#)



KubeCon



CloudNativeCon

Europe 2022

# Current Challenges





KubeCon



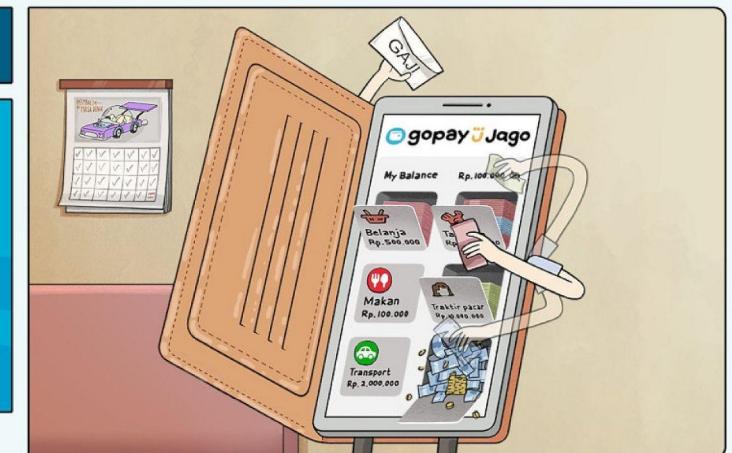
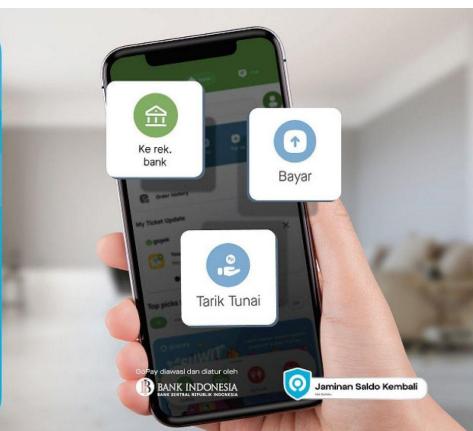
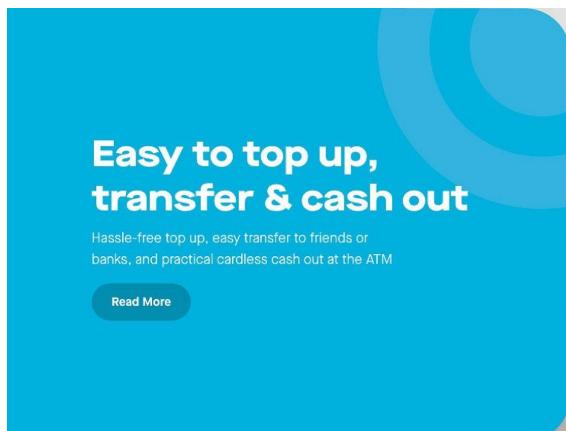
CloudNativeCon

Europe 2022

# Complexity

# Challenges - Complexity

## Payment System Lifeblood is use-cases



Making Sense of Chaos - KubeCon EU 2022 - Iqbal Farabi & Giovanni Sakti

# Challenges - Complexity

## More use-cases means



More features &  
integrations



More services &  
components

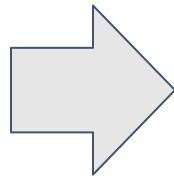


More people are  
involved

# Challenges - Complexity



**Complex System**



**Even More Complex**

- **Not a single person understand the entire system**
  - Each person only “understand” specific portion of system
  - Interaction between systems are not easily understood
- Numbers of novel incidents **numbers more than** incidents due to known issues



KubeCon



CloudNativeCon

Europe 2022

# Compliance

**Due to regulatory compliance with multiple government institutions,**

We have to adhere to the following guidelines:

- **Data sovereignty**

Several core services of our app and their data must reside in data centers located within their own country

- **Reliability**

Financial industry has **strict regulatory requirements** and **customer expectations** with regards to availability and reliability

## Reliability challenges when we started this...

- 
- 01 Reliability Metrics and Prioritization
  - 02 Incidents Handling
  - 03 Services and Components Ownership
  - 04 Accountability

## Reliability management process we have implemented so far...

01	<b>Reliability Metrics and Prioritization</b>	<ul style="list-style-type: none"><li>Well defined global and teams-based SLI/SLO</li><li>99.971%   99.9%</li><li>SLI/SLO adoption as a “common language”</li></ul>
02	<b>Incidents Handling</b>	<ul style="list-style-type: none"><li>Adopted <a href="#">incident command system</a>, significantly improved coordination, communication, and control over incident response.</li></ul>
03	<b>Services and Components Ownership</b>	<ul style="list-style-type: none"><li>Clear ownership of services and components via developer portal.</li><li>Rolling out of developer portal, covering 80% of services and components.</li></ul>
04	<b>Accountability</b>	<ul style="list-style-type: none"><li>Clear accountability for each items for both global and teams-based SLI/SLO.</li></ul>



KubeCon



CloudNativeCon

Europe 2022

**However,  
despite all of that...**

incidents keep happening  
due to the sheer complexity  
of our system

# Challenges - Getting Buy-ins

## We were “lucky”.

### Novel Incidents

For some rough few weeks where we faced several incidents that we haven't faced before.

### RCA with Stakeholders

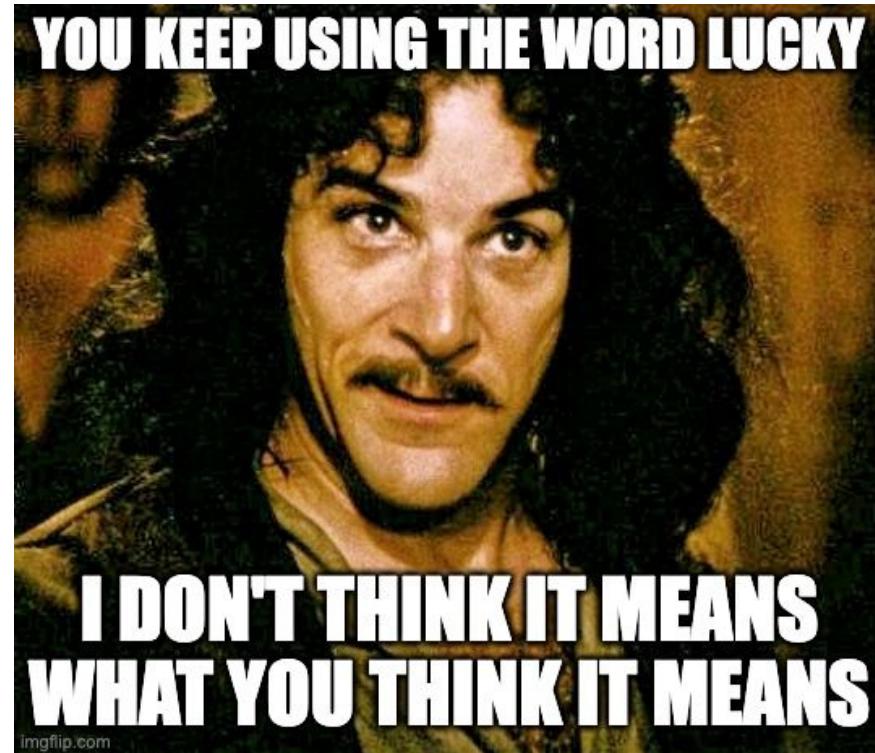
In our regular RCA call with stakeholders, we were asked why this happened.

### Realization

Both us and our stakeholders agreed that the complexity of our system has far exceeded any single person's understanding of it.

# Challenges - Getting Buy-ins

We were “lucky”.



We were “lucky”.

Our stakeholders arrived at the same conclusion.



KubeCon



CloudNativeCon

Europe 2022

# So it begins...





KubeCon



CloudNativeCon

Europe 2022

# Chaos Engineering



**It's a term that many like to use  
but only few understand what it actually is**

*(At least in our circle)*

## What we thought before...



**“Hey, let’s do Chaos Engineering!”**

**“Count me in! I love breaking stuff in production”**

*(True story, we’ve heard this a few times before)*

## Actually, when you think of it...

Can a fintech company, that operates in a heavily regulated industry, **break stuff intentionally** in production at all?

Then we did our research

This time for real.

Chaos Engineering is the discipline of **experimenting** on a system in order to **build confidence** in the system's capability to withstand turbulent conditions in production.

Principles of Chaos Engineering

## Customary Slide on Chaos Engineering Principles

- Build hypothesis around steady state
- Vary real world events
- Run experiments in production
- Automate experiments to run continuously
- Minimize blast radius

# Chaos Engineering - What We Found Out



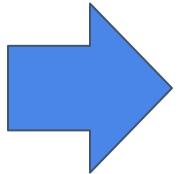
## We found out

Chaos Engineering    !=

Inducing Chaos

We found out

Chaos Engineering

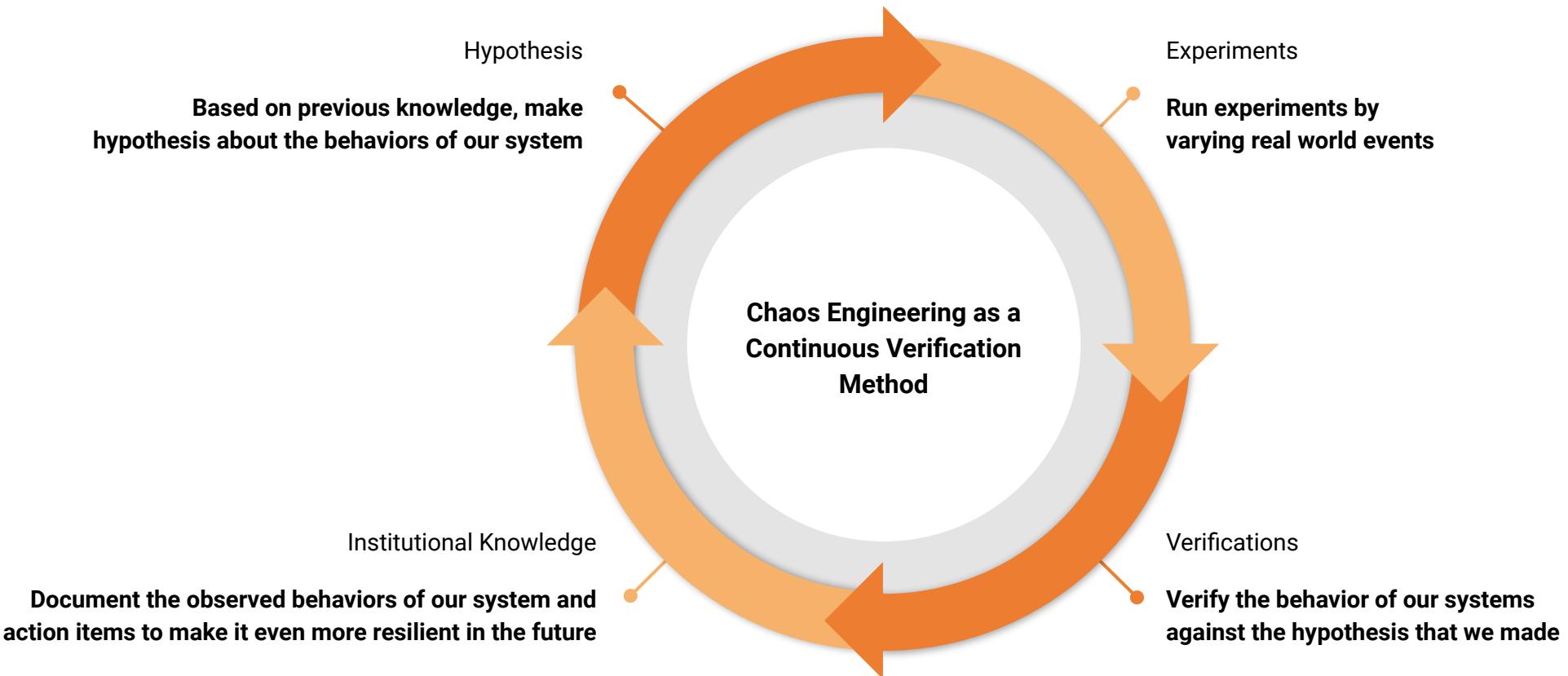


Verification

Continuous verification is a discipline of proactive experimentation in software, implemented as tooling that verifies system behaviors.

- Casey Rosenthal

# Chaos Engineering - What We Found Out





KubeCon



CloudNativeCon

Europe 2022

# Practices

# Practices - Our Journey Before (Recap)



Migration to  
Kubernetes

99.5% | 99.92%  
99.9% | 99.8%

SLI/SLO Org-wide  
Adoption



Migration to Istio



Incident Management  
Revamp



Rolling out  
developer portal

# Main Objective

Elevate our Institutional Knowledge  
and  
Improve our Systems



KubeCon



CloudNativeCon

Europe 2022

# Before we can implement continuous verification,

we need to ensure that  
we have a proper process to manage  
the reliability of our system



KubeCon



CloudNativeCon

Europe 2022

# Defining organization-wide SLI and SLO

is a prerequisite  
to manage our reliability  
systematically

## A little bit on SLI, SLO, and SLA

- Service Level Indicators

What are we measuring?

Example: latency of the system

- Service Level Objectives

How good should it be?

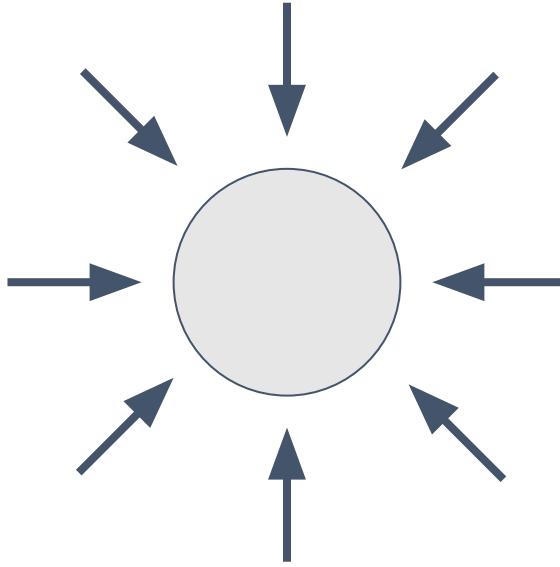
Example: 99% of the time, users should get response within 200ms

- Service Level Agreements

What do we promise to outside parties?

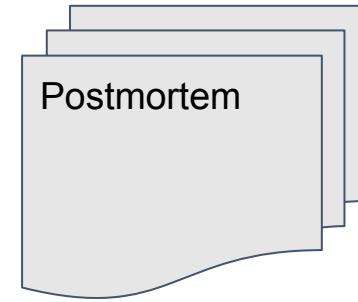
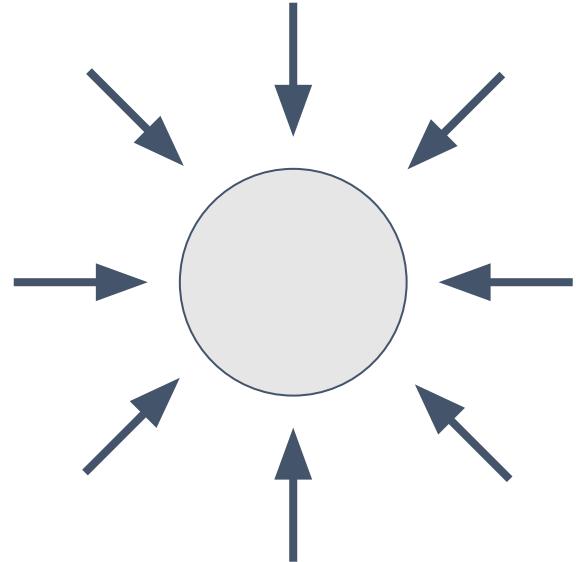
Example: if you use our payment method in your cashier, payment process by scanning QR code will finish in under 10s

# Practices - Adopting Chaos Engineering



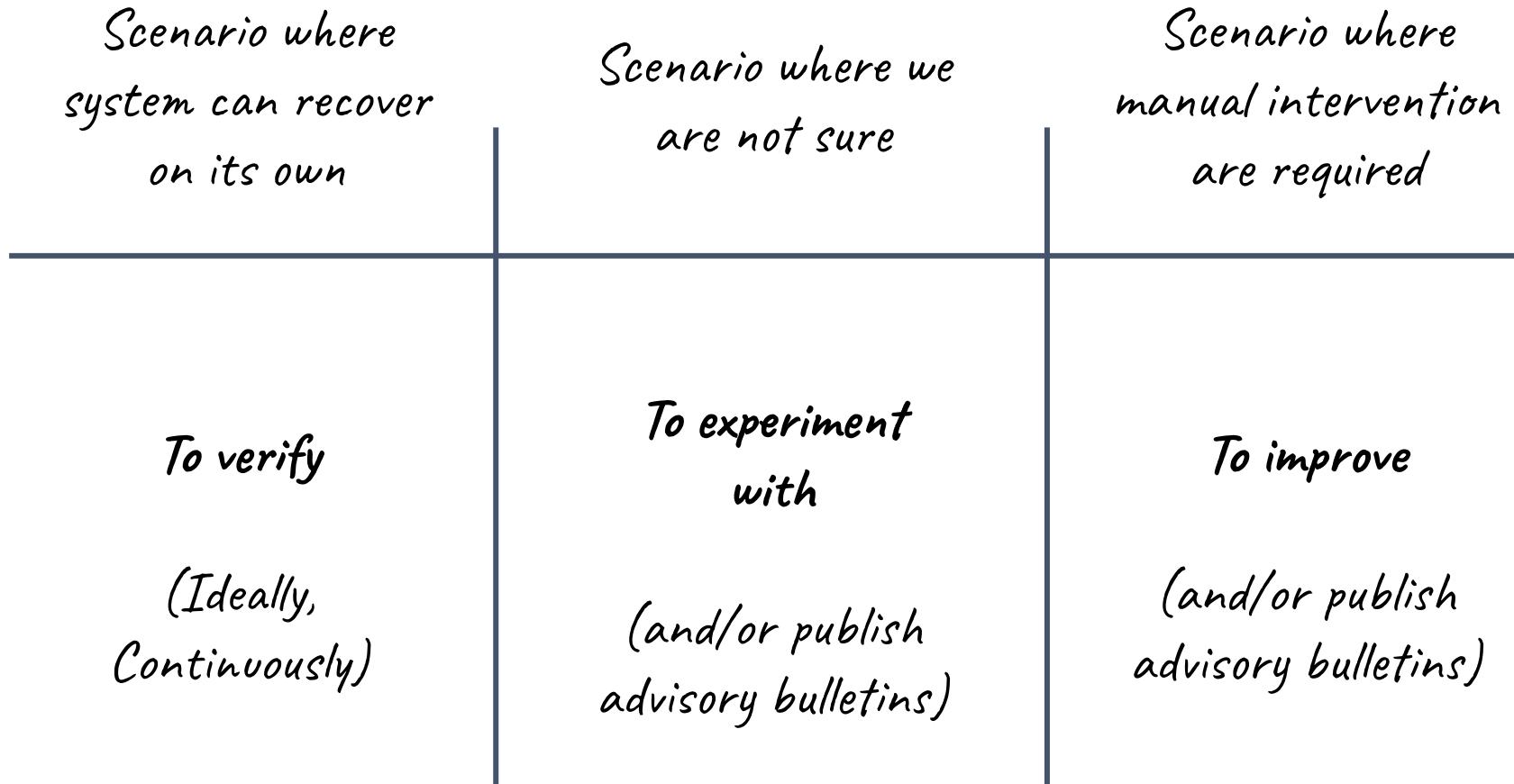
**Centralize knowledge on known limits  
(or capability)**

# Practices - Adopting Chaos Engineering

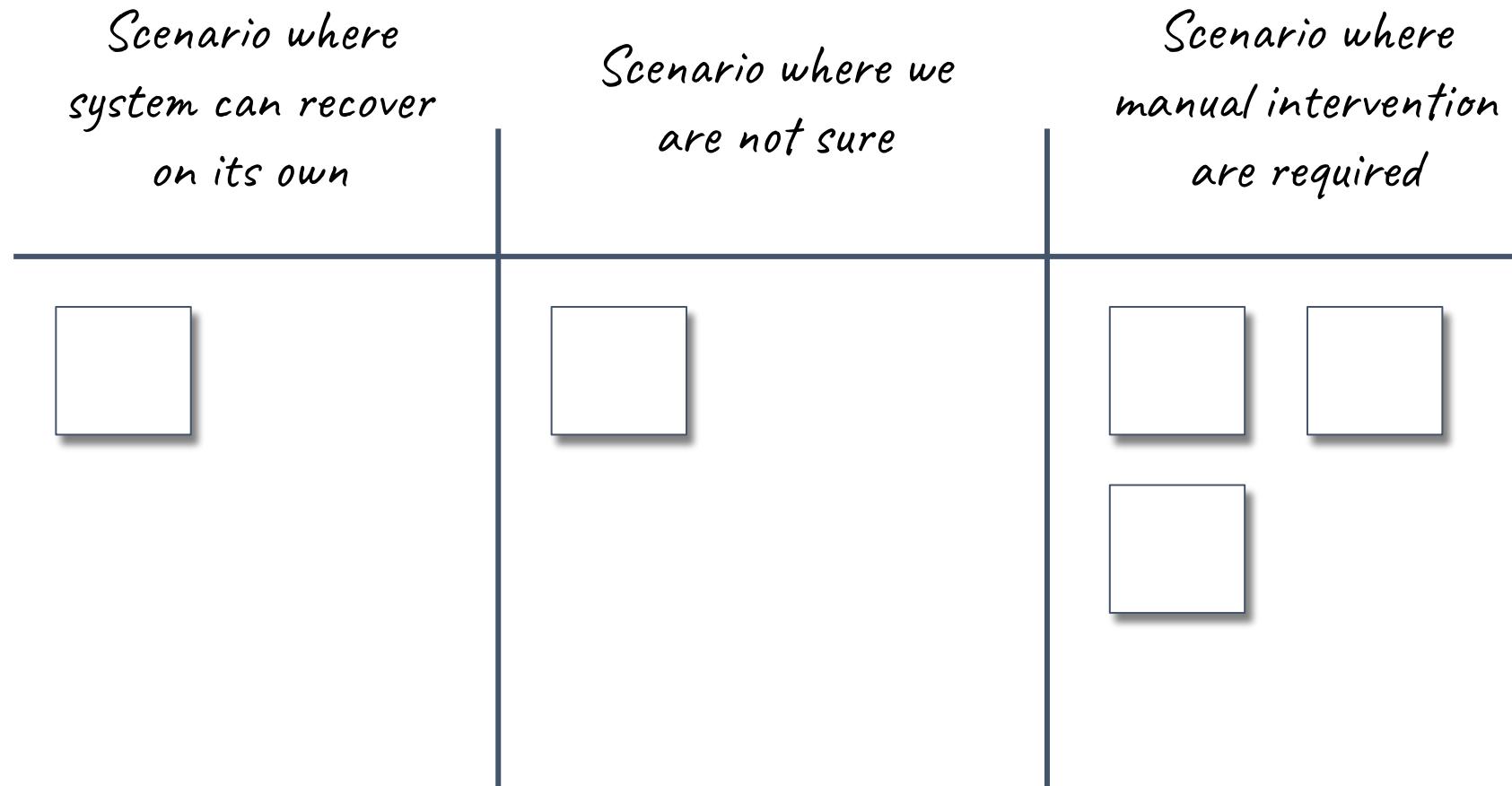


## Discovery from past postmortems

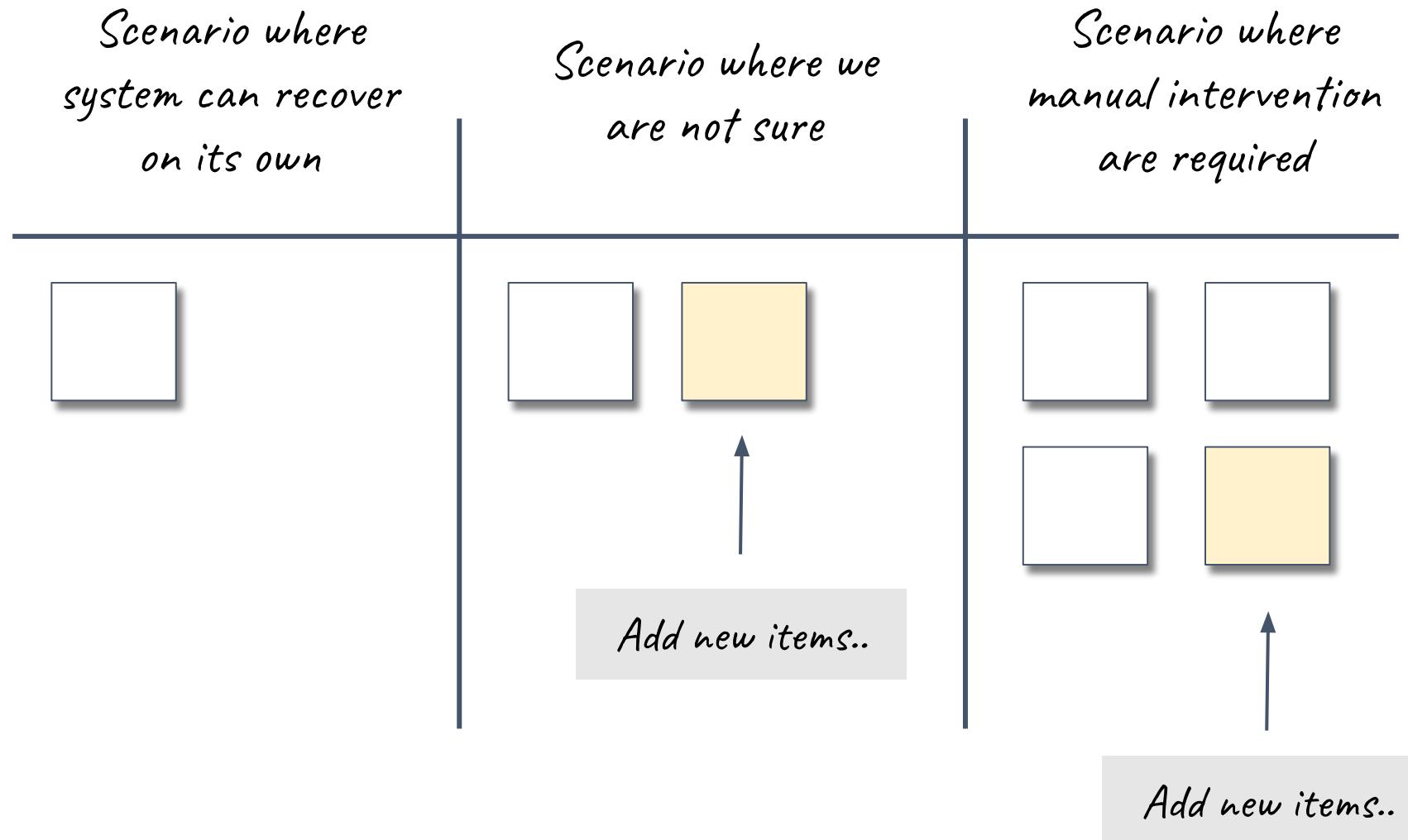
# Practices - Adopting Chaos Engineering



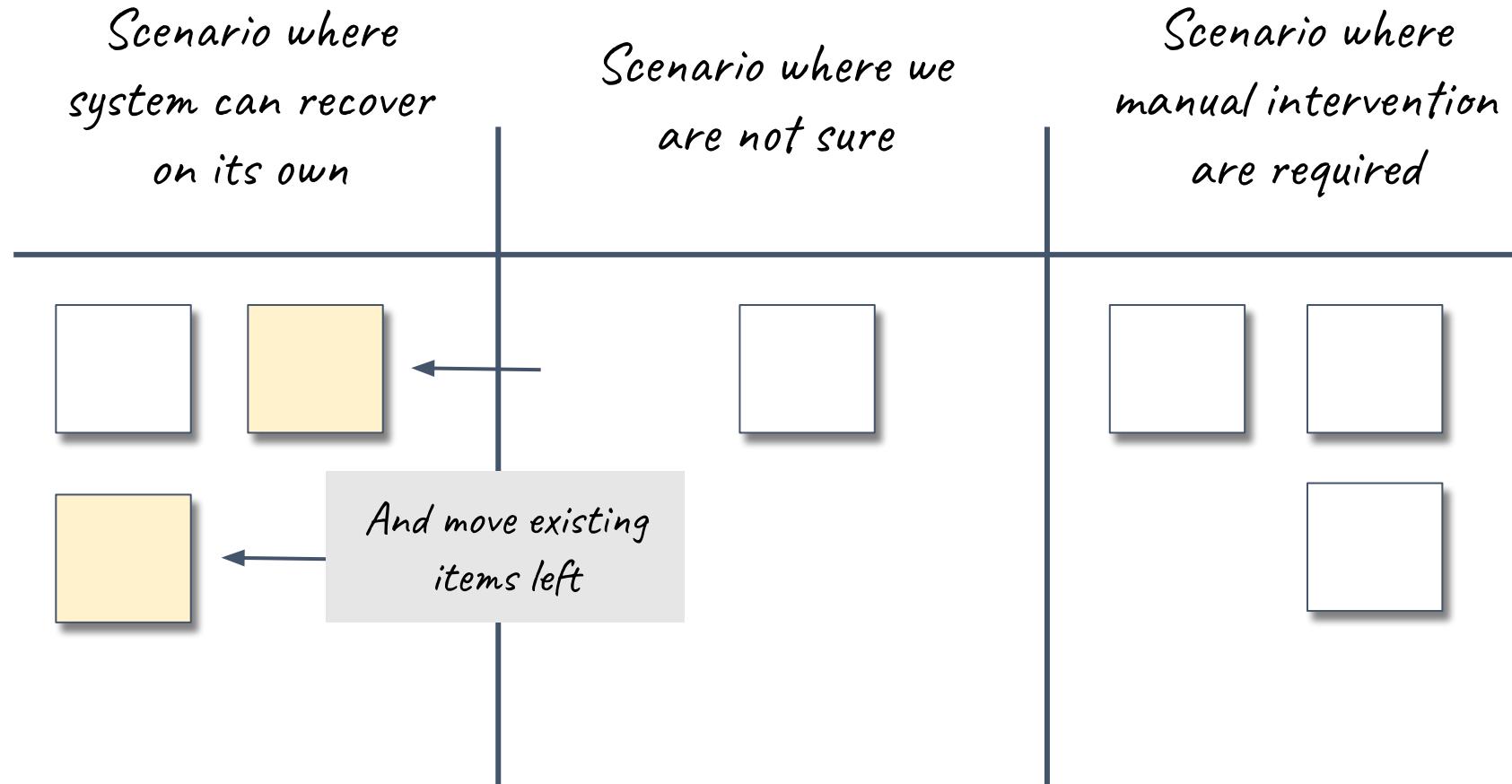
# Practices - Adopting Chaos Engineering



# Practices - Adopting Chaos Engineering



# Practices - Adopting Chaos Engineering



# Practices - Adopting Chaos Engineering

## YYYY.MM.DD - Known issues or limitations

*Short background on the known issues or limitations ...*

*When and where was this issue discovered for the first time ...*

*Recommendations or what can we do about it ...*

*Future (or ongoing) initiatives to address it ...*

# Advisory Bulletins

*Inspired from airworthiness directives*

# Practices - Adopting Chaos Engineering



## Interview People and Review Their Understanding of The System

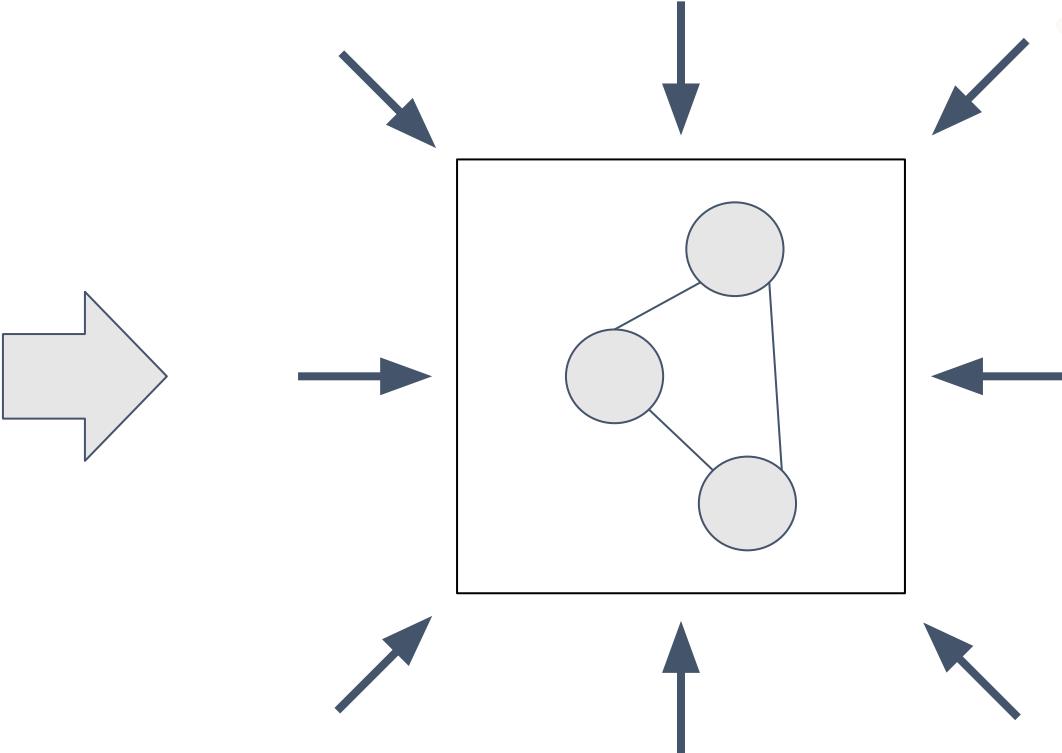
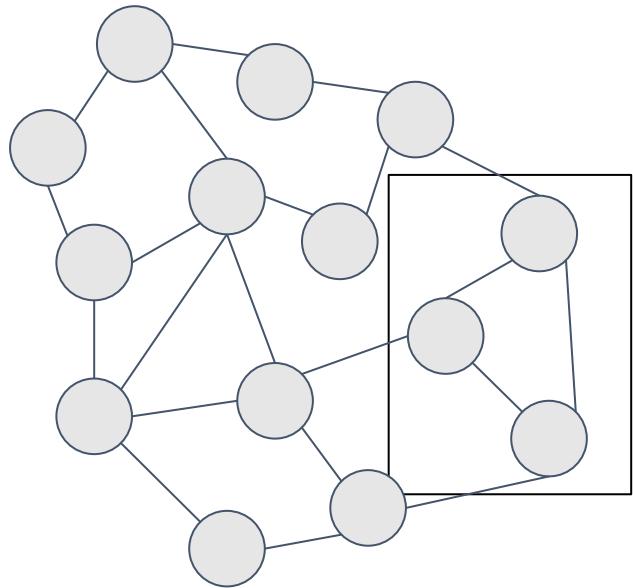
*Making Sense of Chaos - KubeCon EU 2022 - Iqbal Farabi & Giovanni Sakti*

# Practices - Adopting Chaos Engineering



## Verification by Experimentation

# Practices - Adopting Chaos Engineering



## Game Day

# Experiment by Piggybacking Operational Activities

# Practices - Adopting Chaos Engineering



## Experiment on Running System

## Do We Need Tools?

## We need help on:

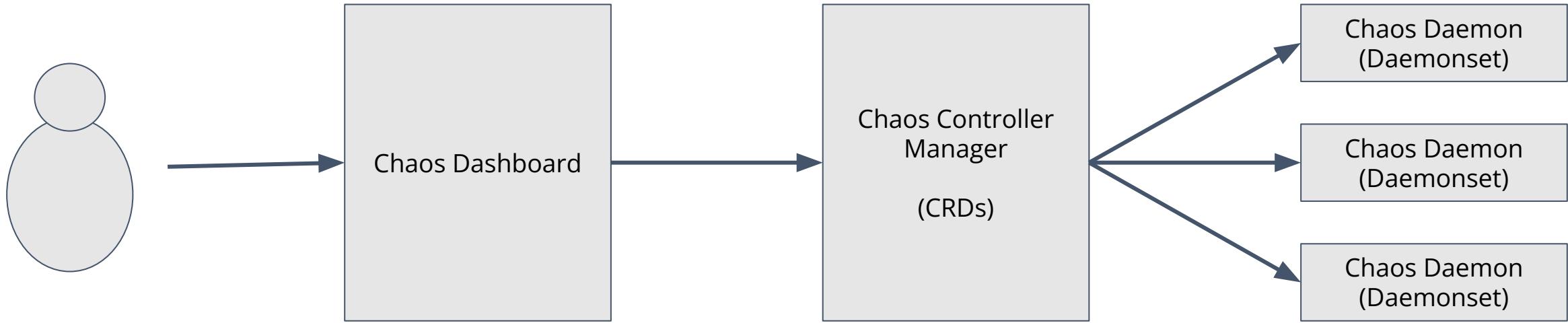
- A system that can help **inject various faults** into the systems
- A system that can help **gather various metrics** to help conclude the experiments
- A **scheduler or mechanism to integrate** fault injection with existing process
- A place where we can **store all experiments history & result**

We need help on:

- A system that can help **inject various faults** into the systems
- A system that can **analyze various metrics** to help conclude the experiments
- A **scheduler or mechanism to integrate** fault injection with existing process
- A place where we can **store all experiments history & result**

**Can Tools Help Us?**

# Practices - Adopting Chaos Engineering

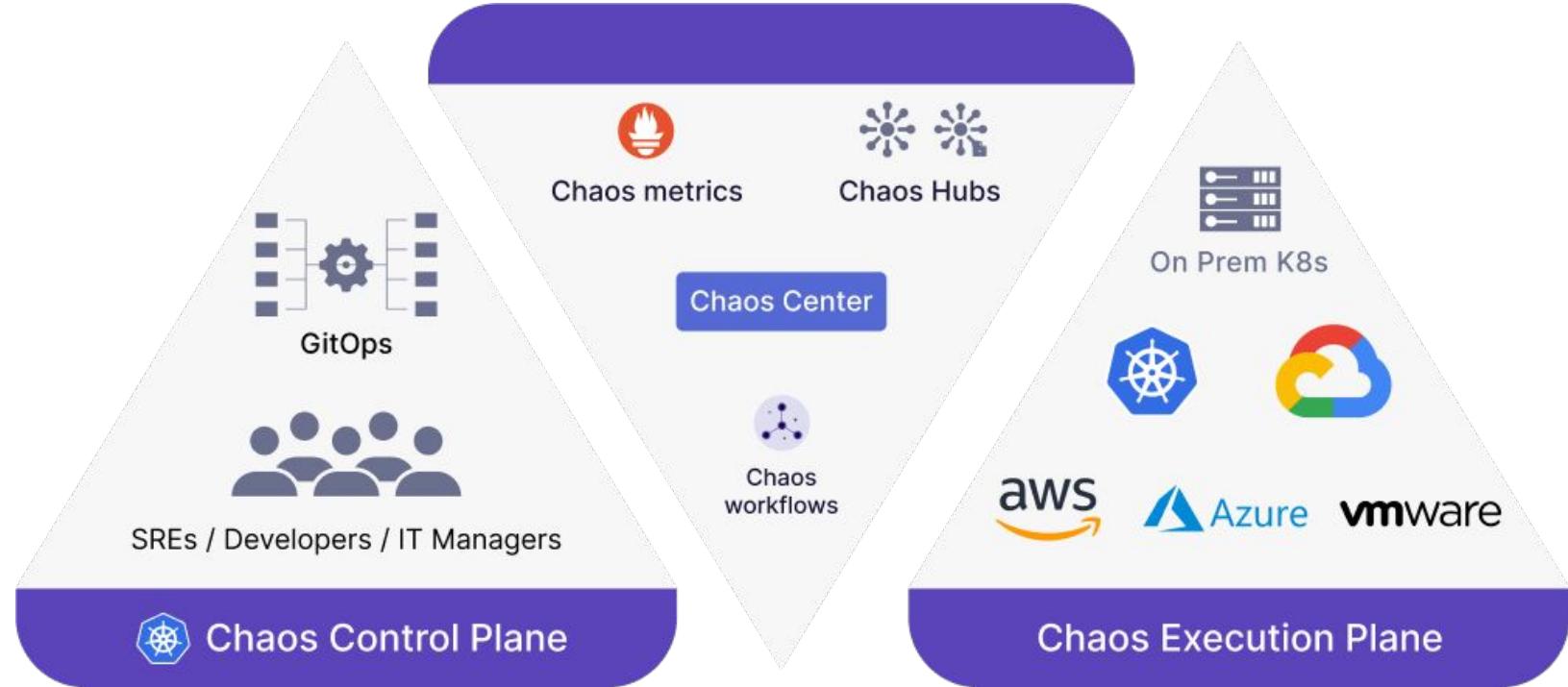


## Leveraging Tools - Chaos Mesh

# Practices - Adopting Chaos Engineering

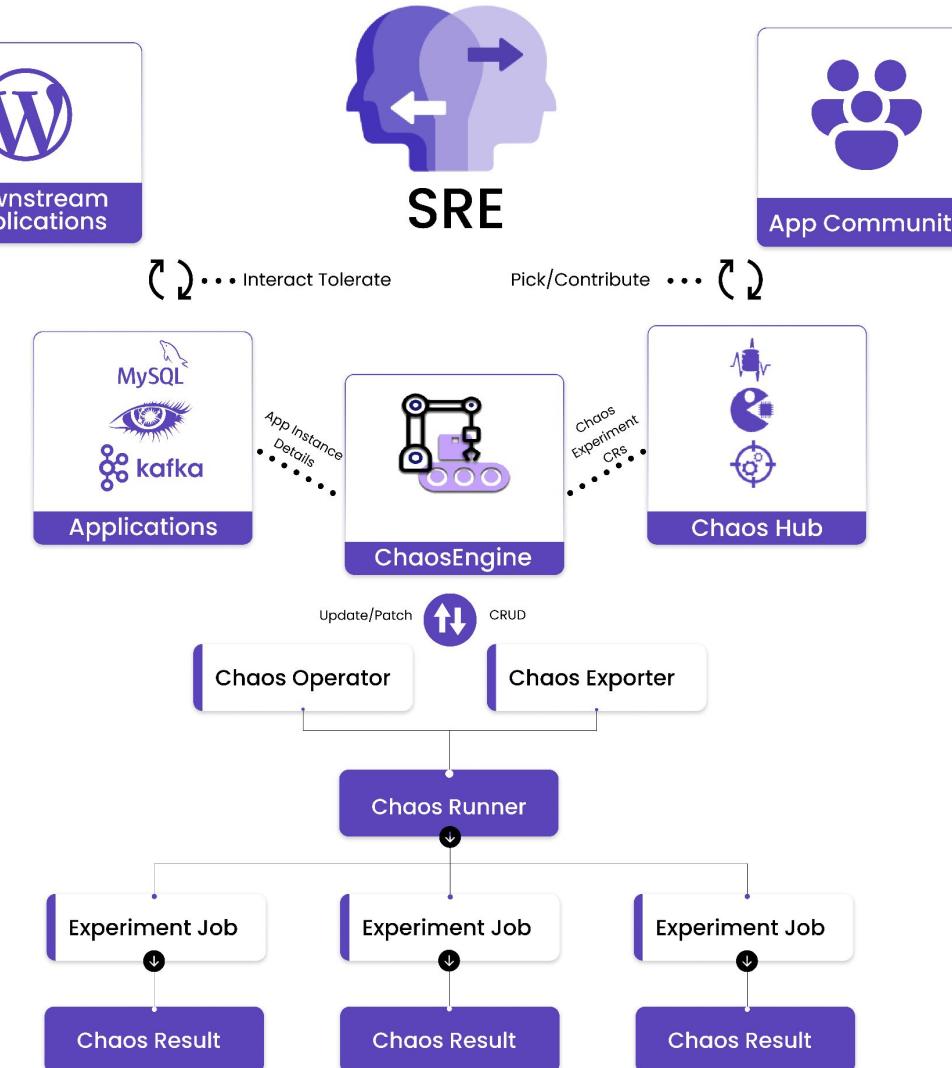
Needs	Chaos Mesh
A system that can help <b>inject various faults</b> into the systems	Supports basic resource, platform and application faults
A system that can help <b>gather various metrics</b> to help conclude the experiments	Metrics are exposed to <code>/metrics</code> HTTP endpoint which Prometheus can collect (Merged in <a href="#">#2397</a> ), Context-specific metrics can be collected to prometheus as well.
A <b>scheduler or mechanism to integrate</b> fault injection with existing process	Set of experiments can be grouped together under “chaos workflow”, which then can be ran in serial / parallel , scheduled, etc
A place where we can <b>store all experiments history &amp; result</b>	Past workflows and its results can be revisited early

# Practices - Adopting Chaos Engineering



## Leveraging Tools - Litmus Chaos

# Practices - Adopting Chaos Engineering



# Practices - Adopting Chaos Engineering

Needs	Litmus Chaos
A system that can help <b>inject various faults</b> into the systems	At the time we created this presentation, Litmus Chaos supports 51 experiments in <a href="https://hub.litmuschaos.io/">https://hub.litmuschaos.io/</a> , injected with Litmus Operator
A system that can help <b>gather various metrics</b> to help conclude the experiments	Experiments results stored as ChaosResult, read by ChaosExporter and can be exposed as Prometheus metrics
A <b>scheduler or mechanism to integrate</b> fault injection with existing process	ChaosScheduler enables us to schedule our experiments in a more granular manner
A place where we can <b>store all experiments history &amp; result</b>	Experiments and their results can be managed and viewed via a UI console called ChaosCenter



KubeCon



CloudNativeCon

Europe 2022

# Wrap-up



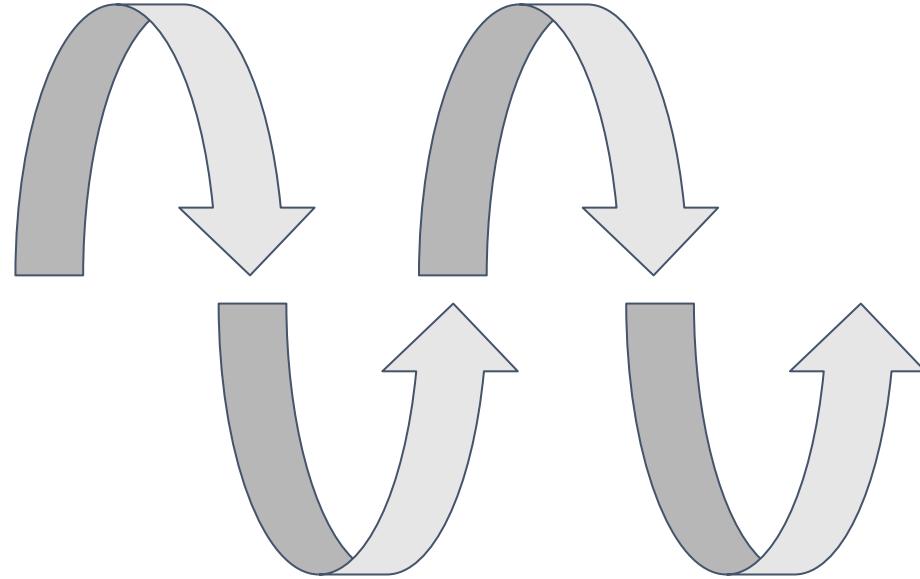
# Based on our journey so far, we learned that:

- Chaos engineering is a way to help us maintaining the **reliability** of a **complex system** sustainably
- By doing **continuous verification** to the system
- Defining an organization wide **SLI/SLO is a prerequisite** to do chaos engineering without sacrificing our reliability metrics

# Based on our journey so far, we learned that:

- Chaos engineering is more than just toolings, **it's the whole process** from implementing prerequisites, building hypothesis, running experiments, to running continuous verifications
- But **toolings** help us to conduct **experiments systematically**
- **Ultimately, as an organization, what we want is to continuously improve institutional knowledge about our complex system and take actions to improve its reliability**

# Wrap-up - What's Next



**Conduct verifications  
autonomously and continuously**

# Where we plan to take this practice in our org going forward:

- Establish processes and incorporate more toolings to **run experiments autonomously and continuously**
- **Roll out** practices to the greater organization to enable more teams to conduct their own experiments - integrate with our dev platforms
- **Learn from and share** what we learned to the chaos engineering community

# Wrap-up - References



- Chaos Engineering - Casey Rosenthal, Nora Jones
- [Principles of Chaos Engineering](#)
- [Chaos Mesh](#)
- [Litmus Chaos](#)

# Wrap-up - Credits

- Gopay Infrastructure Engineering Team
- Head of Payments Platform and engineering leadership
- Fellow product engineers in Payments Engineering Team



*Making Sense of Chaos - KubeCon EU 2022 - Iqbal Farabi & Giovanni Sakti*

# Making Sense of Chaos

*Implementing Chaos Engineering in a Fintech Company*



KubeCon



CloudNativeCon

Europe 2022

# Gracias!



Iqbal Farabi  
(@iqbal\_farabi)



Giovanni Sakti  
(@giosakti)

Iqbal Farabi & Giovanni Sakti  
Consumer Payment (GoPay), GoToFinancial