Automating Chaos Engineering with the Chaos Toolkit

Proactive Reliability Validation

Sylvain Hellegouarch • ChaosIQ CTO • Lead Developer of the Chaos Toolkit • KubeCon 2018

A reminder

Chaos Engineering starts with the **rather high complexity**
of “knowing” our systems end to end.

It is one **tool/practice/discipline**
which collects data to remain knowledgeable.

Practicing edge cases **in a controlled manner**
is at the heart of Chaos Engineering

A Chaos Engineer mindset?

The Chaos Engineer is not a rockstar
but loves supporting the team.

Be open minded

Practice the art of experience without prejudice.

Nurture empathy

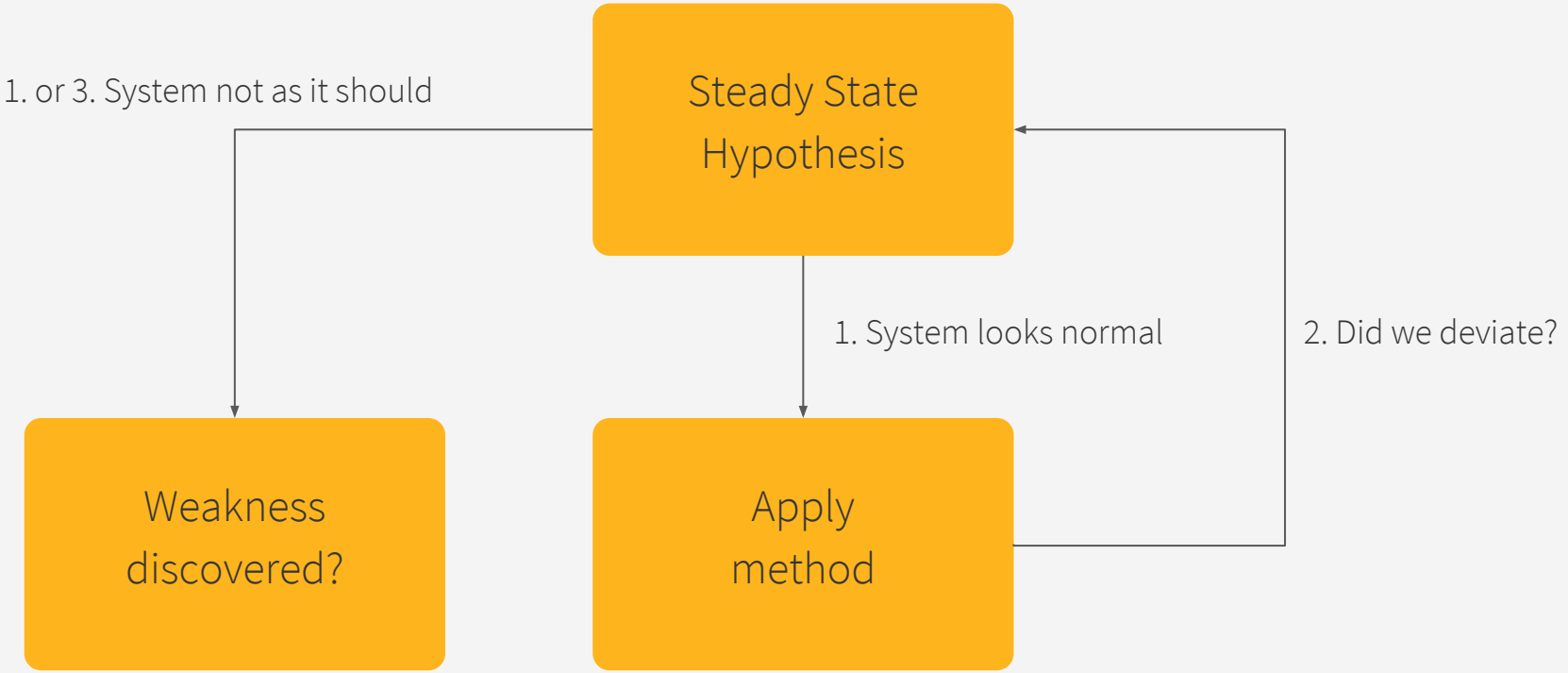
Don’t blame or be snarky. Assertive but not arrogant.

Love an experimental approach

Don’t be scared of not knowing the outcome.
It’s the fun of the game.

The Chaos Engineering flow

The Chaos Engineering flow



I wish I were better at diagrams. Screenshots allowed, though :D

Steady state

The steady state refers to the normality in your systems.

For instance

“Customers spend $X at 8am on average”

It is your baseline.

It comes from objective observations.

Let’s hold back a minute

Forming a hypothesis is not always straightforward.

Usually, you first ask questions based on intuition.

Even before that, you should observe the system to collect data leading you to good questions.

Hypothesis

The hypothesis makes a statement
that should be proved or disproved

Usually to detect weaknesses,
but can also be used to predict an outcome.

For instance

“Reducing by half our payment service instances does not decrease the average spending.”

The hypothesis doesn’t have to be business oriented

but decide which metrics are really critical for you.

Experimental method

Things we vary in the system to prove/disprove the hypothesis.

But should also probe the system for relevant data
to support the analysis.

For instance

“Scale down the payment service by half and query Prometheus for the number of payment and their amount during that period.”

Don’t vary too many things
or analysis will be fruitless.

In summary

A Chaos Engineer demonstrates **great empathy** for users, teams and the system.

By observing the system, the Chaos Engineer asks questions leading to **hypothesis and an experiment**.

A chaos experiment starts with a steady-state hypothesis.

A chaos experiment varies conditions in the system while **probing for relevant data**.

Did the system deviate?

We proved our hypothesis and can start learning.

ChaosToolkit

ChaosToolkit

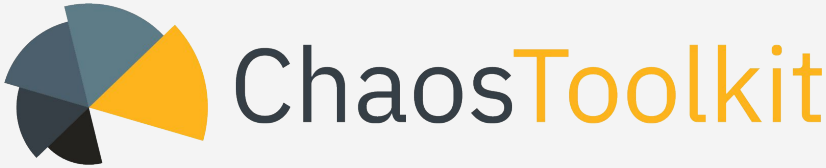
implements the flow of the experiment we just described.

ChaosToolkit

is an open source tool, born in September 2017.

ChaosToolkit

gathers a small, nicely growing, community
with contributions from its members.



is implemented in Python 3.5+ with a functional approach.

ChaosToolkit

Core objectives

Be the entry points for Chaos Engineering experiments.

Simplicity is the key for durability in a software.

Interface with **platforms** and **systems** natively.

Prefer extending over re-writing.

Be declarative.

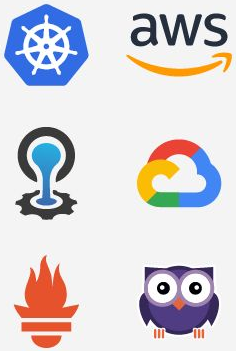
Automation first class citizen.

Be a friendly project community :)

You

ChaosToolkit

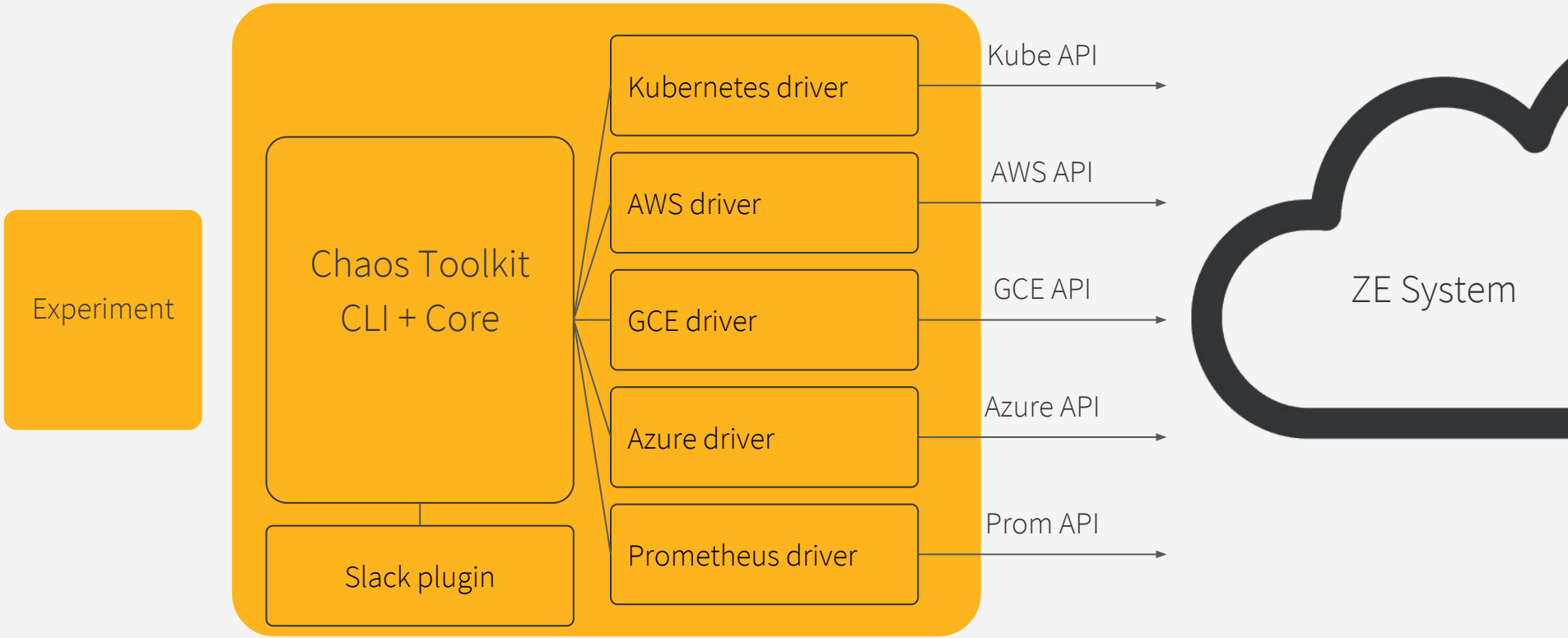
Drivers



Plugins

Your System

General architecture



Experiment

Open API for a declarative description
of the experiment end-to-end.

Specified at http://chaostoolkit.org/reference/api/experiment

Can be described in **JSON** and **YAML**.

Drivers

Drivers are merely extensions that know how to **talk to a specific system** through its API.

Drivers can be in **Python** or even **simple processes** executed by the toolkit.

Drivers don’t try to be too clever,
but offer the right level of operations
you need to shake and probe your system.

Existing drivers

Kubernetes GCE AWS Azure

Humio Cloud Foundry Prometheus

Plugins

Plugins are extensions that **add functionalities** to the toolkit itself.

Reporting

Notifications (Slack…)

Demo(s)

What’s next for the Chaos Toolkit?

Kubernetes Job/CRD for scheduling

Newer Documentation

Drivers in other languages (Go, Rust…)

More on discoverable experiments

A 1.0 in 2018

Any questions?

@lawouach