



Chaos Engineering in Delphi

Embrace your inner chaos!



Marco Brevaglieri

Software Developer
Trainer & Consultant



Homepage & social links

👉 www.brevaglieri.it



Prometheus Client 4 Delphi

👉 <https://github.com/marcobrevaglieri/prometheus-client-delphi/>



Compila Quindi Va

👉 www.twitch.tv/compilaquindiva



Delphi Podcast

👉 www.delhipodcast.com



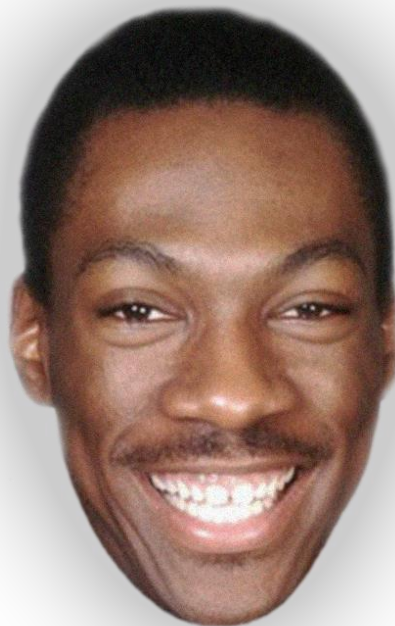
AGENDA

- Chaos Engineering: what is it?
- Principles of Chaos Engineering
- Getting started with Chaos Engineering
- How to code mitigating troubles
- Tools and libraries
- Recap and Q&A

begin

Murphy's Law

“Anything that
can go wrong
will go wrong,
and at the worst
possible time,,



Chaos Engineering?

«**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».

Principle Of Chaos Engineering (<https://principlesofchaos.org>)

Principles

- **Experimenting**: carefully applying failure
- **Build confidence**: verify if the hypothesis is correct
- **Withstand turbulent conditions**: find weaknesses and make systems more resilient

Experimenting



Steps

1. Set the baseline
2. Create a hypothesis
«When we stop service X, then Y will happen»
3. Test
4. Evaluate

Be careful!

1. Discuss the abort conditions
2. Prepare a rollback plan and a backup plan
3. Make sure you start with a small «blast radius»

Benefits

- Determining risk and cost and setting service-level indicators, objectives and agreements
- Testing your system as a whole (often complex and distributed)
- Deeper and shared knowledge about systems
- Find emergent issues you were unaware of

How do we start?

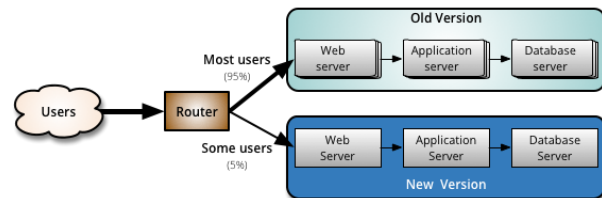


Start small

- Get everyone involved
- Start small (think about the blast radius!)
- Start locally (production is the holy grail!)

Convince your management

- Talk about resilience and reliability, not about chaos!
- Plan controlled experiments, during office hours
- Start small (adopt *Canary Releases**)
- Get them involved in the experiment design



* 👉 <https://martinfowler.com/bliki/CanaryRelease.html>

Common Scenarios

- Traffic spikes
- Unreliable network
- Unavailable dependencies
- Region evacuation
- Host failure
- DNS outage

More Scenarios

→ Software

- Expiring certificates
- Memory leaks
- Licenses
- Versioning

→ Applications

- Timeouts
- Retries
- Exception handling
- Load

→ Applications infrastructure

- Redundancy
- Self-healing
- Infrastructure as Code

→ Operations

- Monitoring and observability
- Incident response
- Measuring

→ People

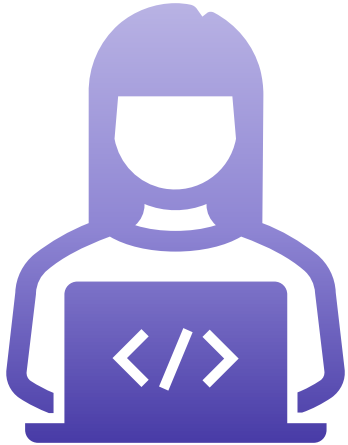
- People Chaos

Game Days

1. Plan a session of (at least) an hour
2. People from at least 2 teams, about 4
3. Decide on communication tools
4. Plan and design a few experiments
5. Execute and document results
6. Share the results within your organization

A good time is **at 10 in the morning**: everyone is awake, had some coffee and is... ready for action!

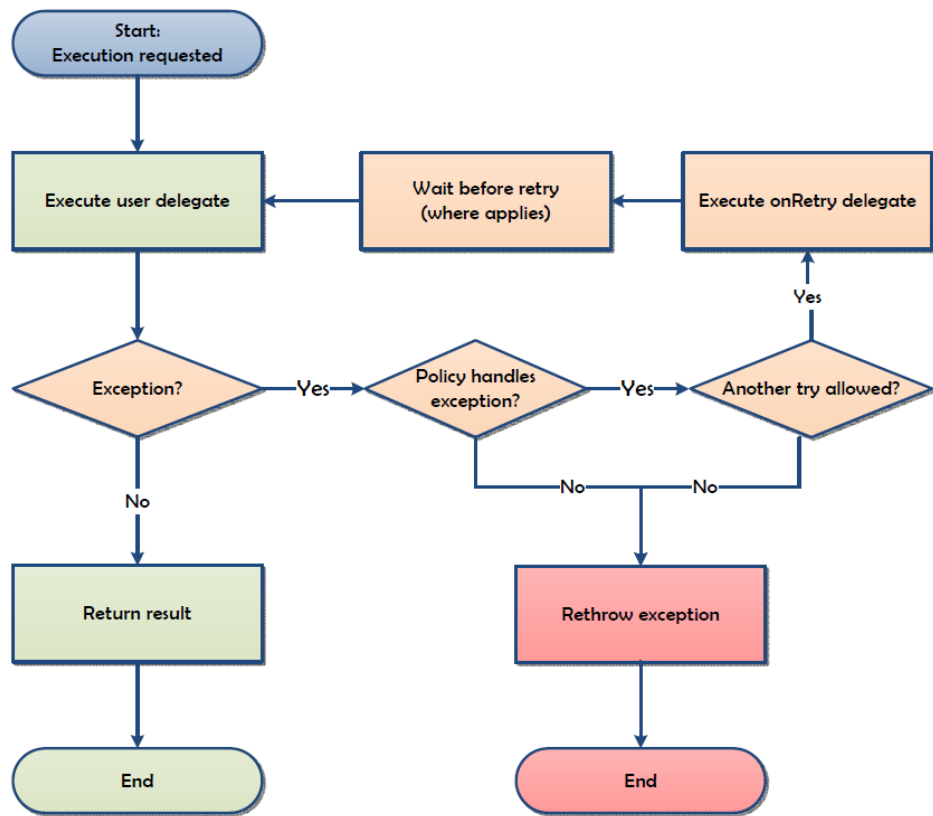
What about the code?



Architecture Patterns

- Retry Pattern
- Circuit Breaker
- Fallback (with default/cache)
- Timeout
- Bulkhead
- Rate Limiter
- Rate Gate

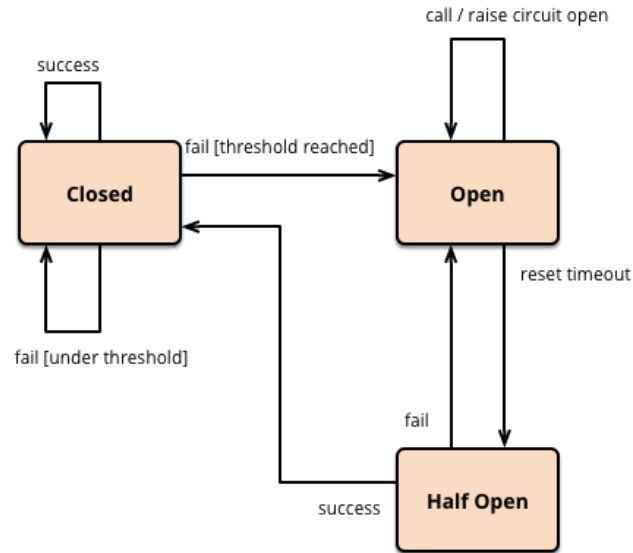
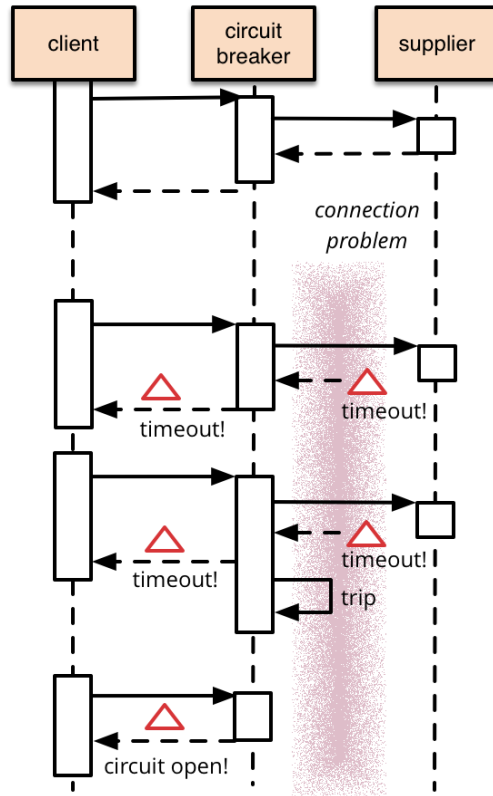
...and many others!



An example: implementing «Retry Pattern»

Let's write some code!





Another example: implementing «Circuit Breaker»

Tools & Libraries

- Chaos Monkey
(open source from Netflix!)

- Azure Chaos Studio

- Chaos Toolkit
(Python) 

- Polly (JS / .NET) 

- Simmy (.NET) 

...and Delphi? 

A little help is coming...



👉 <https://github.com/marcobreveglieri/murphy-delphi>

- Written in Delphi 11.3 Alexandria
- Provides a starting point for implementing resilient and fault-tolerant architecture patterns
- Takes advantage of latest Delphi language features to simplify code
- **Free and open-source!**

⚡ **Warning!** It is at a very early stage!

Let's ~~write~~ see some code!





Summary

Recap

- Chaos Engineering is a tough and wide topic, but as usual we can grasp and adopt the good things
- Start small, just on your local machine (and play!)
- Get everyone involved in team, including management
- Think about scenarios, plan experiments, define radius
- Discuss experiments with your team and collect ideas
- Test and then measure, measure, and document!

end.

Q & A





THANK YOU!

Have a nice conference! 🙌