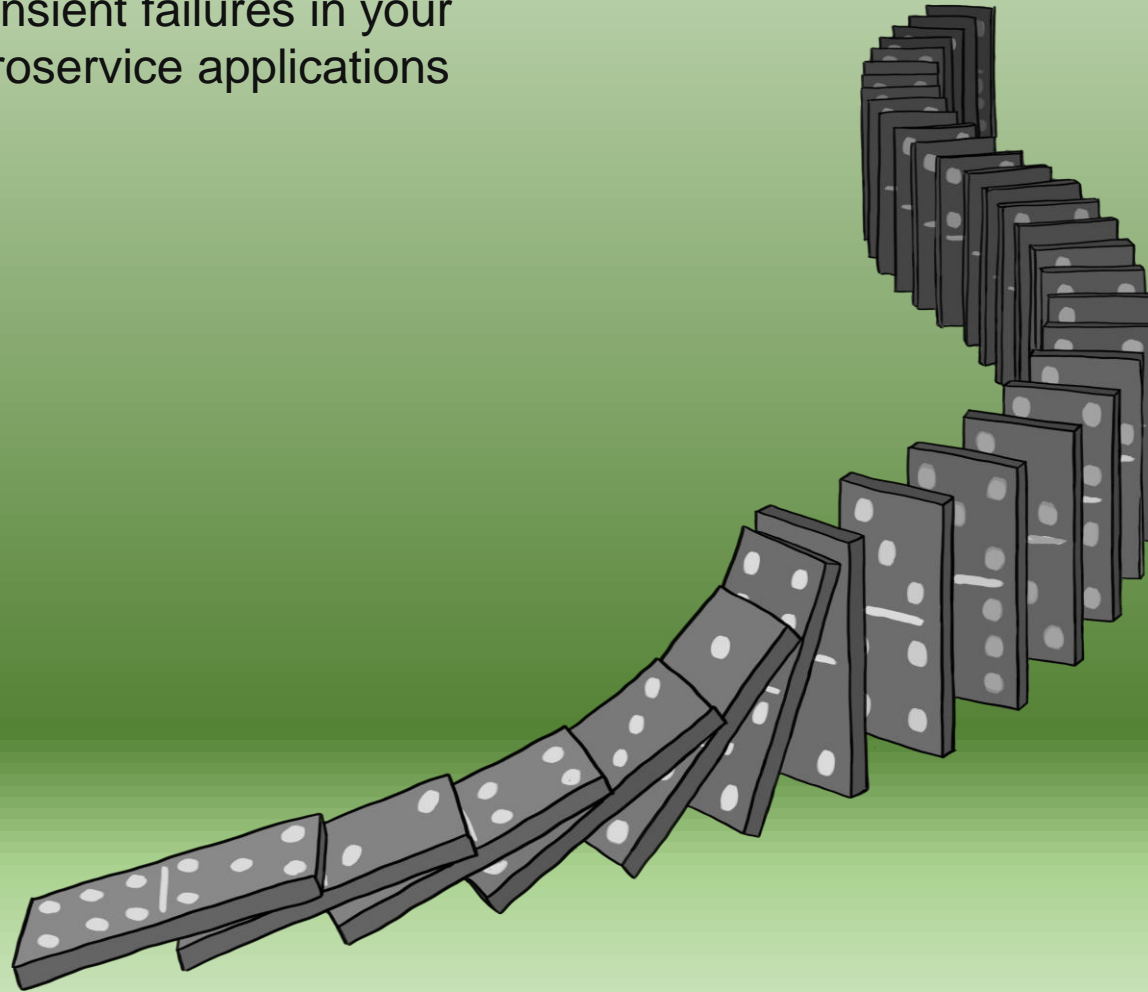
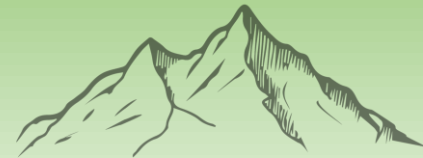


When Microservices Go Bad!

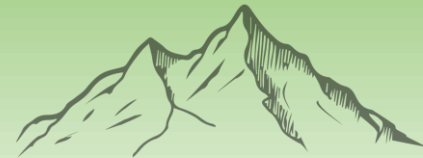
How to handle transient failures in your distributed or microservice applications



Andy Clark



Case Study – Union Castle Line



Recent Incidents

 The Independent

BT internet down: Outage takes down Amazon Alexa, Ring doorbells and more

BT internet down: Outage takes down Amazon Alexa, Ring doorbells and more ... the Ring video doorbell system that is also owned by Amazon,...

3 weeks ago



[https://www.cnn.com/2021/12/10/aws-explains-out...](https://www.cnn.com/2021/12/10/aws-explains-outage/index.html)

AWS explains outage and will make it easier to track future ones

10 Dec 2021 — A major **Amazon Web Services outage** on Tuesday started after network devices got overloaded, the company said on Friday.










<https://www.pwc.co.uk/issues/crisis-and-resilience>

Systems failure - PwC UK

A bank experiencing **systems failure** meant 600,000 customer payments and direct debits went missing. The failure was caused by the bank's IT infrastructure ...



Risk – Probability / Impact

Impact	Probability			
		Low	Medium	High
	High			
	Medium			
	Low			



Strategies for dealing with failures

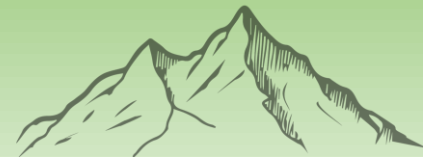
- Ignore them...
- Little and often
- Monitoring
- DIY resilience
- Policy based resilience



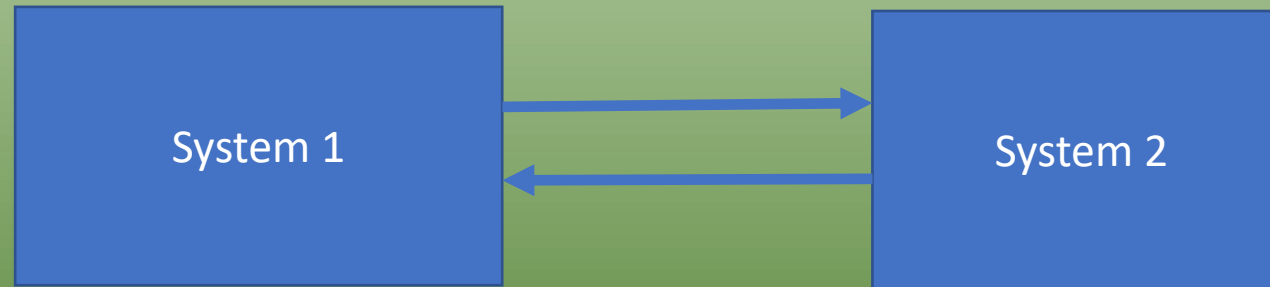
Policy Based Resilience



- Define a policy
 - What symptoms (return codes, exceptions)
 - What behaviour (retry, fallback, rate-limit)
- Call API using the policy

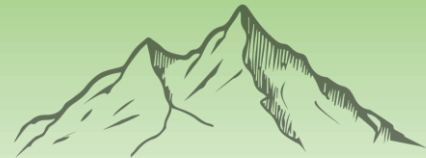


How many points of failure?



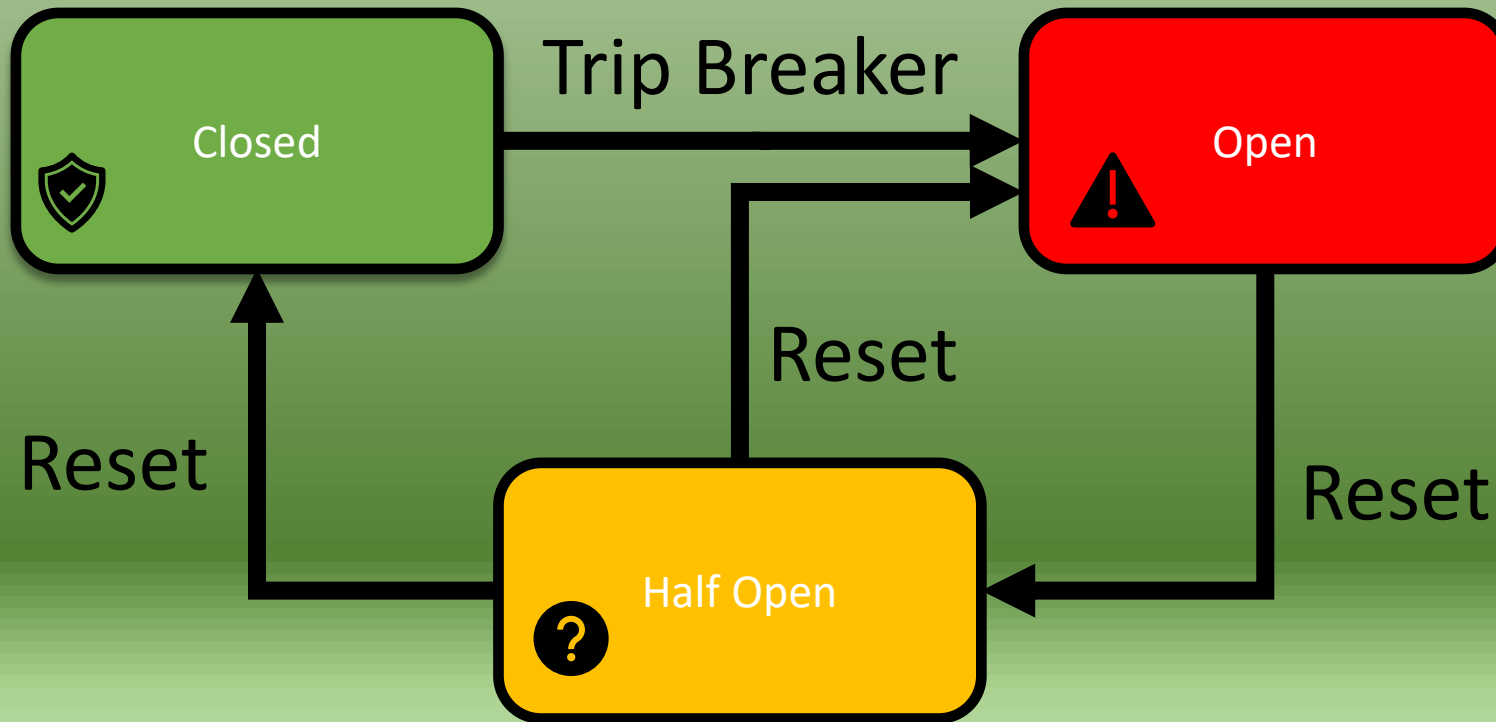
Strategies for dealing with failures

- Retry
 - Idempotency
 - Don't retry indefinitely
 - How long to wait?
 - Exponential Backoff
 - Jitter



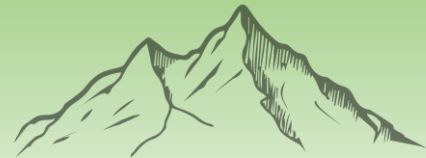
Strategies for dealing with failures

- Circuit Breakers



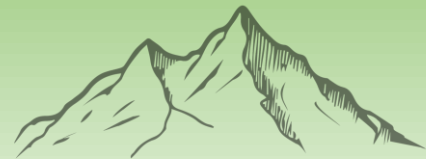
Strategies for dealing with failures

- Bulkheads
- Rate Limiting
- QoS



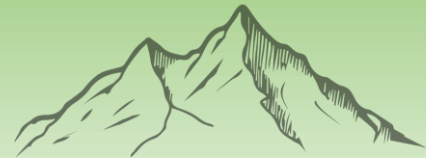
Strategies for dealing with failures

- Caching
- Fallback
- Queuing



Strategies for dealing with failures

- Timeouts
 - Many APIs have a default timeout
 - Don't wait forever



Other Polly Use Cases

- PolicyWrap – For complex policies
<https://github.com/App-vNext/Polly/wiki/PolicyWrap>
- Refreshing an access token
<https://www.jerriepelser.com/blog/refresh-google-access-token-with-polly/>
- NoOp – For testing or library
<https://github.com/App-vNext/Polly/wiki/NoOp>
- PolicyBuilder – For more creating our own policies
- Polly.Extensions.Http
<https://github.com/App-vNext/Polly.Extensions.Http>



Reference

Polly

- <http://www.thepollyproject.org/>
- <https://www.nuget.org/packages/Microsoft.Extensions.Http.Polly/>
- <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/implement-resilient-applications/implement-http-call-retries-exponential-backoff-polly>

Microservices

- <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/implement-resilient-applications/>
- Building Microservices by Sam Newman - <https://samnewman.io/>
- Cloud Design Patterns - <https://docs.microsoft.com/en-us/azure/architecture/patterns/>

Videos:

- <https://www.youtube.com/watch?v=F9r04PuaOcs>
- <https://www.youtube.com/watch?v=zSRFOi8bURw>
- <https://developertoarchitect.com/lessons-microservices.html>



Andy Clark

<https://www.linkedin.com/in/techychap/>

<https://hachyderm.io/@Techychap>

<https://www.mountainwarehouse.com/help/work-for-us/>

