

Dr Aled Sage  
aled@cloudsoft.io

cloudsoft



# Application Resilience: Challenges & Good Practice

City Business Club - ISITC Europe | July 2019

VP Engineering @ Cloudsoft

PhD and 20 years experience with distributed systems

Expertise in Cloud, Devops and Automation

Across different verticals, including financial services



@aledsage



aled@cloudsoft.io



linkedin.com/in/aledsage



github.com/aledsage

1. What is resiliency?
2. Changing landscape
3. Case study
4. Problems & solutions



- Business applications
- Service level objectives (SLOs)  
Service level agreements (SLAs)
  - Aim is *not* 100% availability
  - “Availability” is not binary
- Design for failure
  - Proactive and reactive

CRM, payroll, analysis, trading systems, settlement systems, ...

Google Site Reliability Engineers (SRE)

“Striving for Imperfection: Using an **error budget** to move fast without compromising high reliability”

Proactive: High availability (HA)  
Avoid problems

Reactive: Know when something is wrong  
Know how to respond  
Repair  
Disaster recovery (DR)

**Gartner**

# **Why Business Leaders Don't Care About the Cost of Downtime**

9 April 2019



- Business wants agility
- Security and reliability still vital
- All while driving down costs



- Private and public cloud
- Kubernetes and Openshift (using containers)
- Bare-metal and mainframes



TODO: read up on Metro Bank

arling

<b>Incumbents</b>	<b>Disruptors</b>
Hybrid environment	Everything is cloud-native
Chaos engineering? “No way!”	Kill servers in production
Wall Street bank: decrease VM provisioning to 9 hours	New container in seconds; New Virtual Machines in minute(s)
3-6 month releases	Small changes several times a day
100s of applications Many lines of business	Small product set
Legacy high-value apps	No legacy apps

## Improve the basics:

- Improve resiliency and agility of existing apps
- Fit with old-world and the future

## *Self-healing applications!*

*Your application can automatically detect and fix problems*

### APPLICATION-CENTRIC APPROACH

- ✓ Empower and involve application architects and developers
- ✓ Codify best practices and use of the bank's strategic tools
- ✓ Consistent way to automate resiliency



### ACHIEVEMENTS UNLOCKED

- ✓ Application-level Resiliency at scale
- ✓ Meets business & regulatory needs
- ✓ Across multiple evolving platforms







## Manual

Manual procedures and tribal knowledge.  
Responses to degradation or failure not automated.



## Bespoke

Wide variation in techniques and quality;  
bespoke processes.



## Operations fragmented

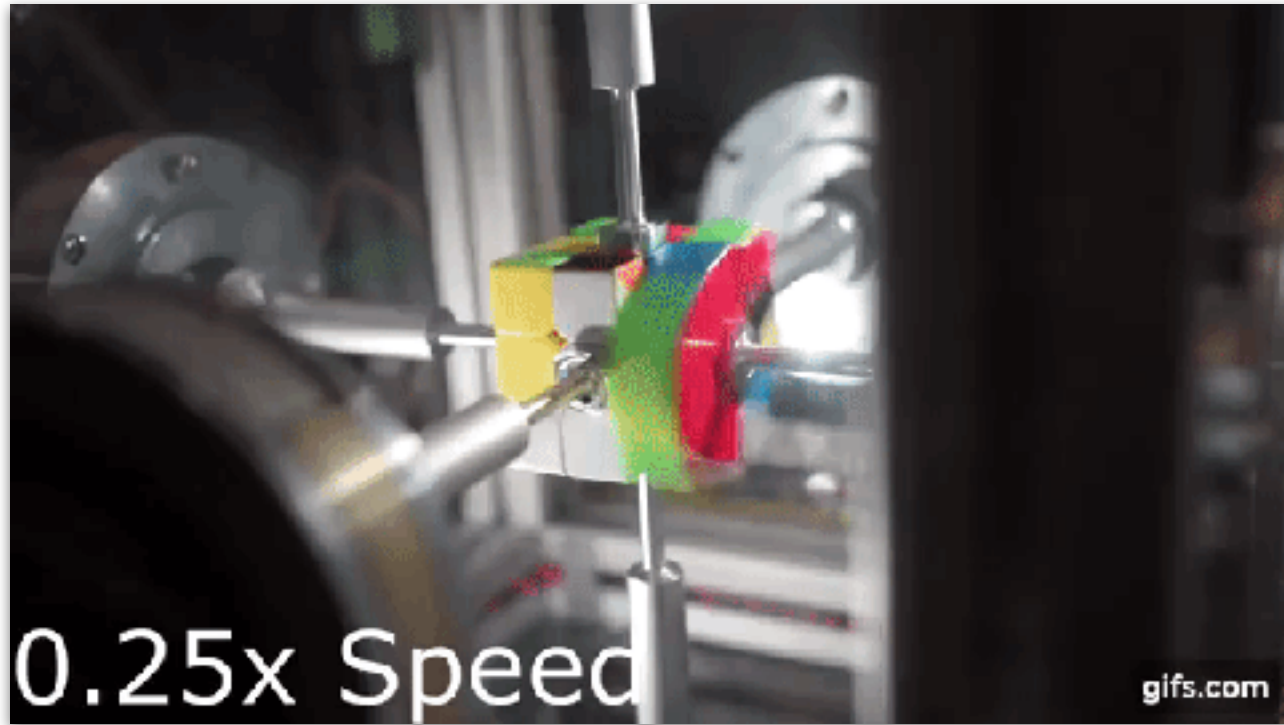
Involves many systems.  
Fiefdoms & politics.  
Resiliency is a cross-cutting concern.



## Rarely tested

Sufficient to meet auditors' requirements.  
But little practice or variation in scenarios;  
manual testing is expensive.

<b>Problems</b>	<b>Solutions</b>
Tribal knowledge, rarely tested	Gamedays
Processes, environments and applications are complicated and ever-changing	Post mortems & continual improvement
High availability and DR needs understanding of application, but infrastructure-focused	Application-centric approach
Bespoke / inconsistent approaches	Empower the application architects; Codify and share best practices.



<b>Problems</b>	<b>Solutions</b>
Manual runbooks	Automated recovery
Involves many systems	Encourage APIs and automation; “glue code” to codify their use; focus on common application architectures.
Runbooks and CMDB out-of-date	“Living model” of the application; Deployment and in-life management codified in this model.

- **Restart** the web-servers (e.g. on an out-of-memory)
- **Repave** servers or clusters (e.g. when a restart fails, or an update is required)
- **Auto-scale** based on latency, load or time-of-day
- **Manage DR** within a region and/or across regions



## Self configuring

- Automated configuration of components and systems follows high-level policies. Rest of system adjusts automatically and seamlessly

## Self healing

- System automatically detects, diagnoses, and repairs software and hardware problems

## Self optimizing

- Components and systems continually seek opportunities to improve their own performance and efficiency

## Self protecting

- System automatically defends against malicious attacks or cascading failures. It uses early warning to anticipate and prevent system wide failures



## AMP streamlines development, operations and governance for any application on any cloud



### Rapidly Scalable

Autonomic policy-based management



### Cloud Enabling

Liberate applications from infrastructure



### Powerful Tools

Consistent in-life management tooling and features



### Improve Visibility

Unified management view of applications

## This week

- ❑ Get the Gartner report: “Why Business Leaders Don’t Care About the Cost of Downtime” free of charge
- ❑ Talk to person(s) in charge of application resiliency
  - ❑ Example post mortem from the last major incident; how was this knowledge shared and used?
  - ❑ How are application authors involved in planning for and implementing resiliency?
  - ❑ Is there consistent monitoring and recovery strategies (focusing 80:20 rule of existing apps)?
  - ❑ How often is it tested? Are there gamedays?

## Next 90 days

- ❑ Identify example app(s)
- ❑ Get a review from an expert of an example workload (“well-architected review”)
- ❑ Organise gamedays
- ❑ Introduce post mortems

## Next 6 months

- ❑ Consider automation software at the application level for ‘low-hanging fruit’
- ❑ Involve and empower your application authors







Thank You!  
Any Questions?

+aled@cloudsoft.io let's finalise before  
the event an include a link to  
Slideshare to get a copy of these