ADDO

ALL DAY DEVOPS

NOVEMBER 12, 2020

Deepak Ramchandani, Contino

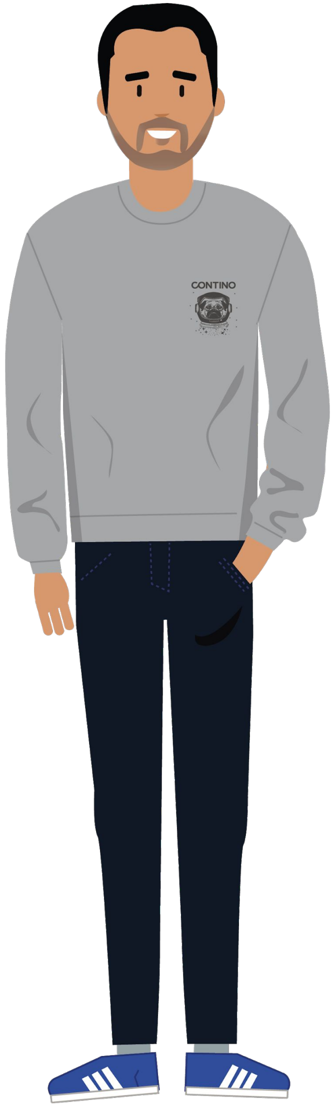
Driving Digital Transformation through CloudOps and SRE

Deepak Vensi

Account Principal @ Contino

(in) **linkedin.com/in/deepakrv/**

- Multi-Cloud adoption within Regulated Industries
- Advocate & build in-house mature Engineering practices with a focus on SRE
- Changing the Operating Model with a focus on FinOps, GitOps & DevSecOps
- Help develop Cloud Native Products and Services for Enterprises
- Build sustainable in-house digital capabilities for long term adoption



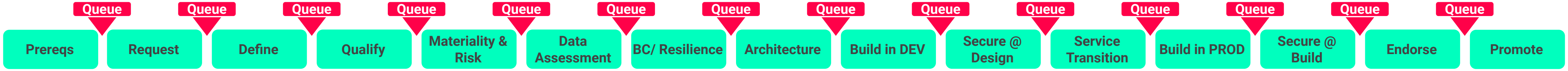
Agenda

- Core functions part of IT Operations
- What does Operations mean in Cloud
- Why SRE?
- How can you bring Reliability / Docs / Code / Controls together?
- How to upskills a team towards SRE

Core Functions of Traditional IT Operations

- Incident Management
- Change Management
- Release Management
- Configuration Management
- Capacity Management
- Business Continuity & Backup
- Asset Management
- Demand Management
- Knowledge Management
- Risk Management

What does a “Pipeline” look like?

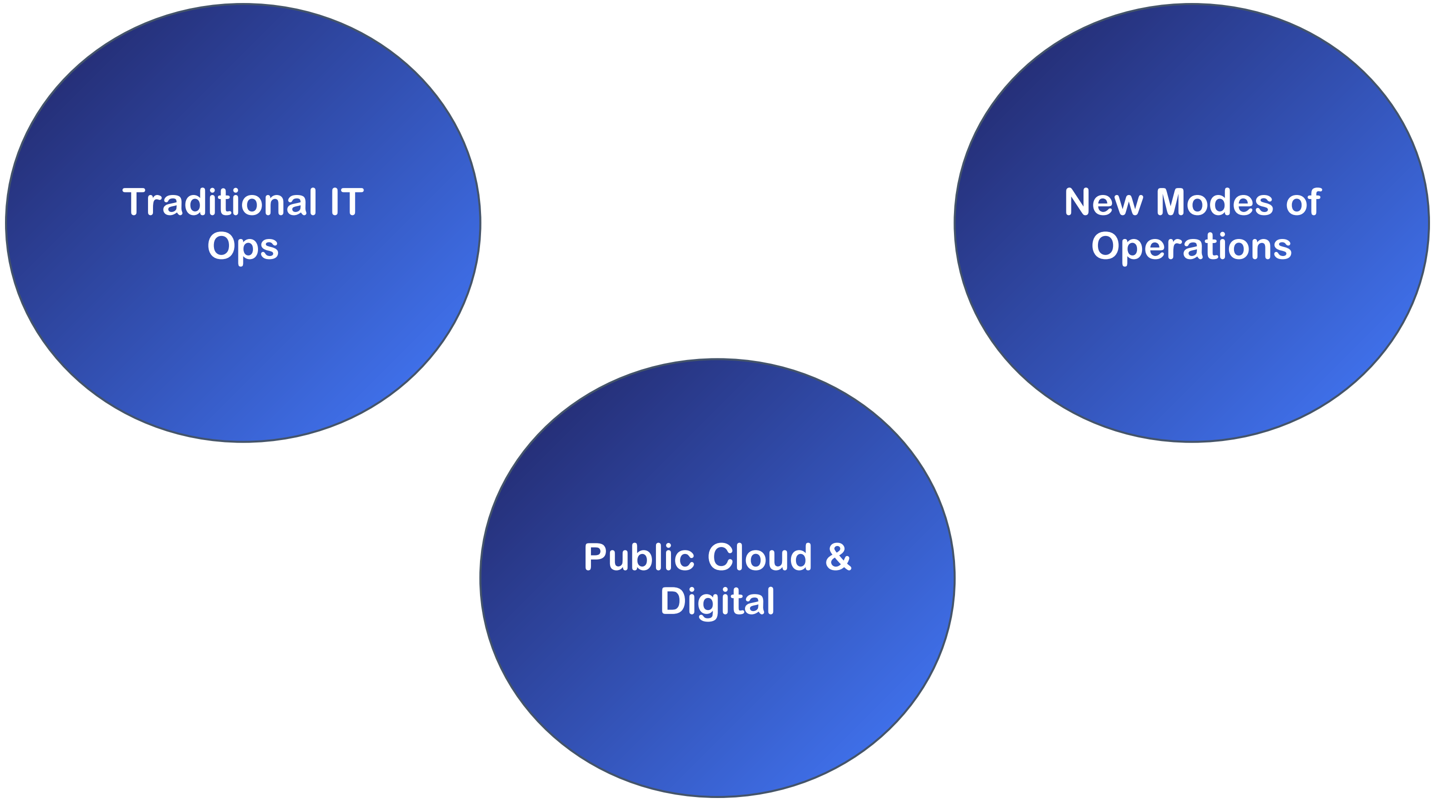


Prereqs
Queue
Request
Queue
Define
Queue
Qualify
Queue
Materiality & Risk
Queue
Data Assessment
Queue
BC/ Resilience
Queue
Architecture
Queue
Build in DEV
Queue
Secure @ Design
Queue
Service Transition
Queue
Build in PROD
Queue
Secure @ Build
Queue
Endorse
Queue
Promote

What do IT Ops really want?

- Reliability
- Security
- Operability
- Predictability
- “Control”

The Gap to Bridge



Traditional IT Ops

New Modes of Operations

Public Cloud & Digital

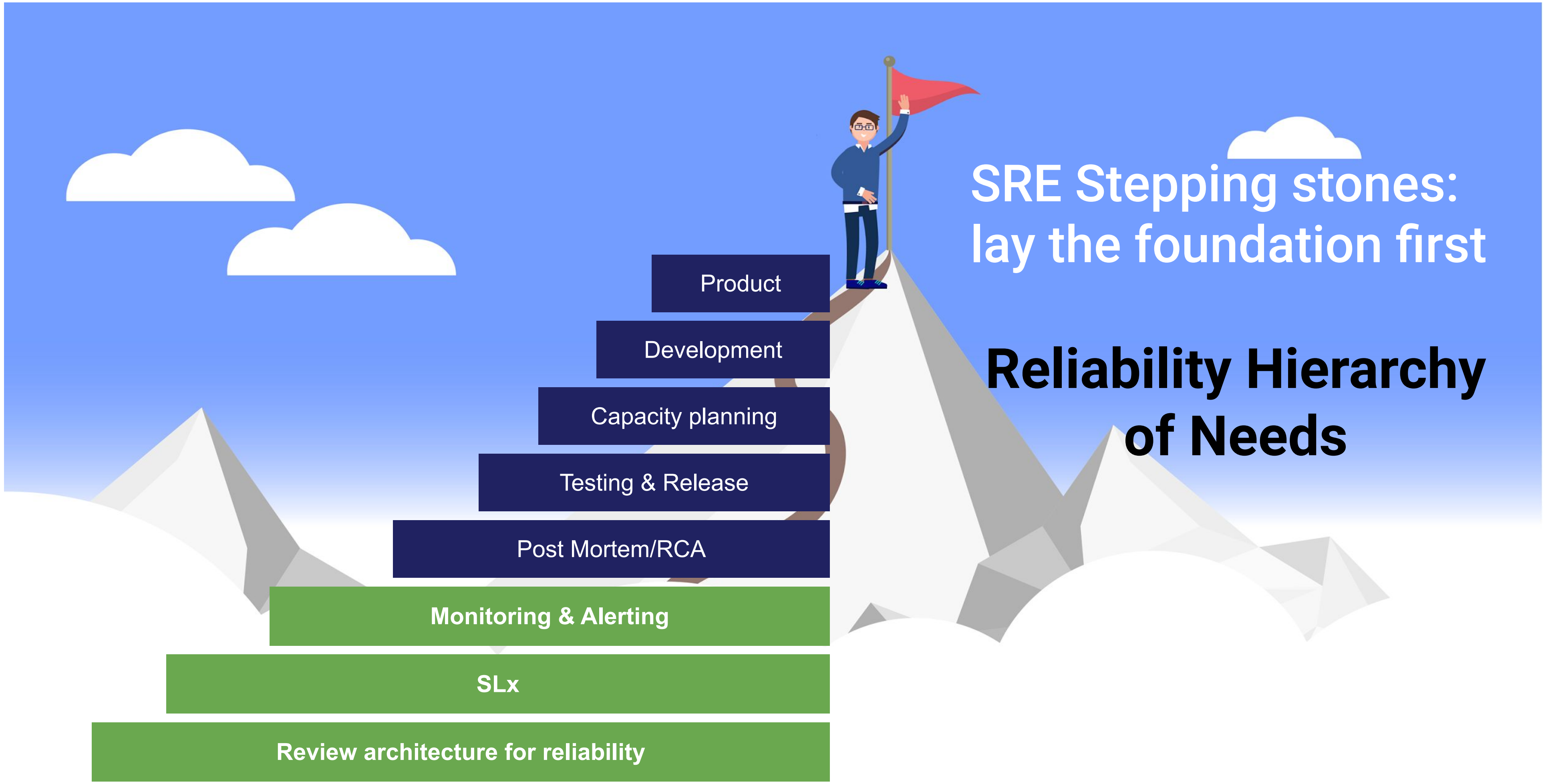
SRE to the Rescue!

Site reliability engineering (SRE) is a discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems.[1] The main goals are to create scalable and highly reliable software systems. According to Ben Treynor, founder of Google's Site Reliability Team, SRE is "what happens when a software engineer is tasked with what used to be called operations."

The four key pillars to CloudOps

- SRE
- FinOps
- GitOps
- DevSecOps

SRE to the Rescue!



Product

Development

Capacity planning

Testing & Release

Post Mortem/RCA

Monitoring & Alerting

SLx

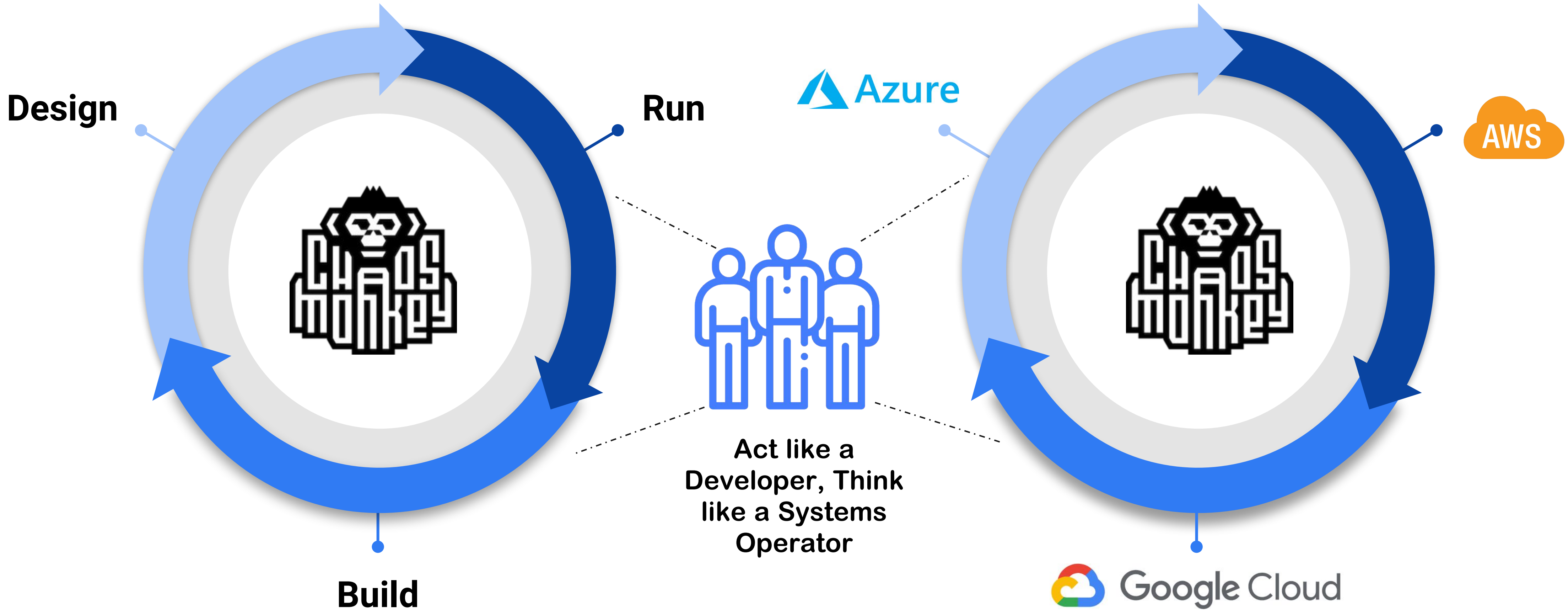
Review architecture for reliability

SRE Stepping stones:
lay the foundation first

Reliability Hierarchy of Needs

Fear not!

You Build it you Run it



Act like a Developer, Think like a Systems Operator

Roles and Responsibilities

1. Consult with Consumers

2. Create Service Level Indicators & Objectives

3. Design, Build & Run Platforms / Services / Apps

4. Eliminate Toil via Automation

9. Emergency Response & On-call Support

Site Reliability Engineer

"Act like a Developer, think like a Systems Operator".

5. Monitor Distributed Systems & Develop Products

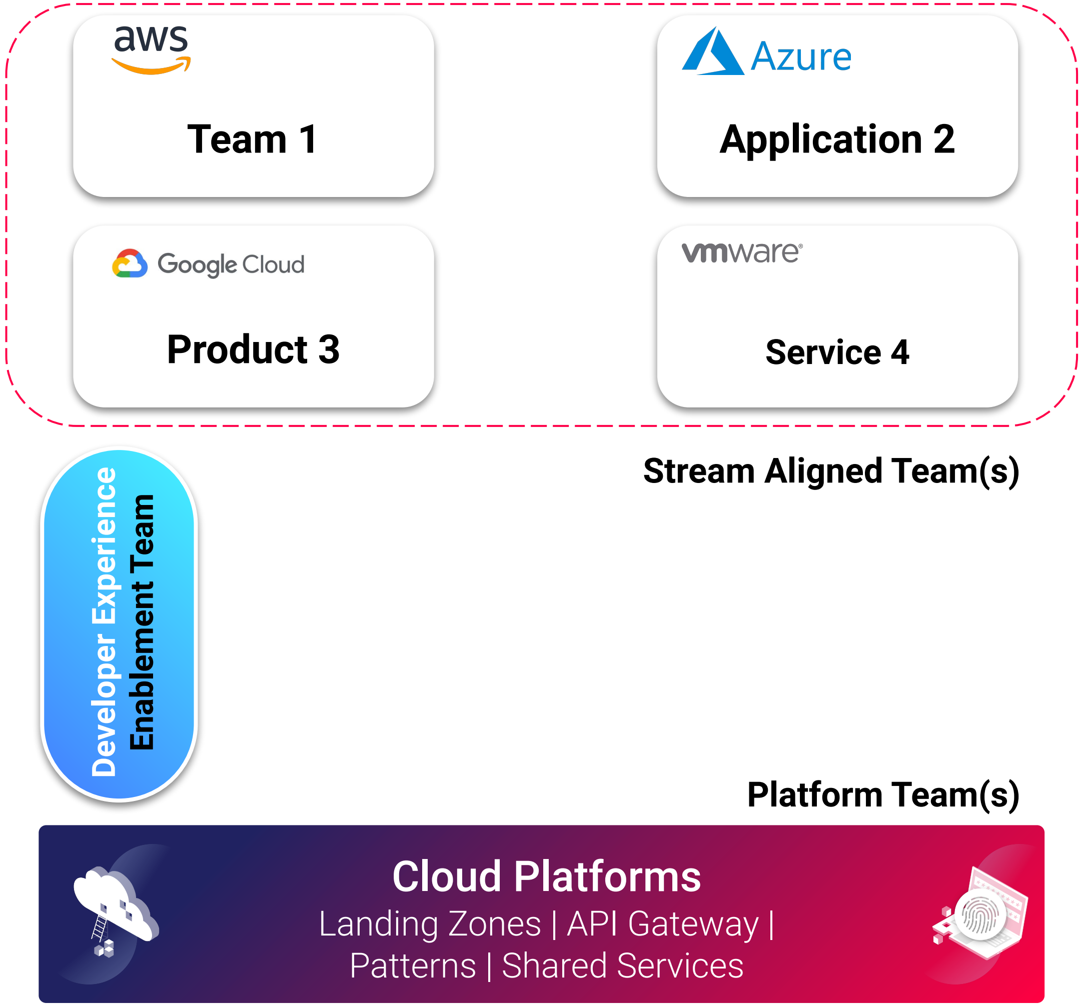
8. Track Platform Outages

7. Manage Platform & Consumer Incidents

6. Postmortems & Learn from Failure

How to get started?

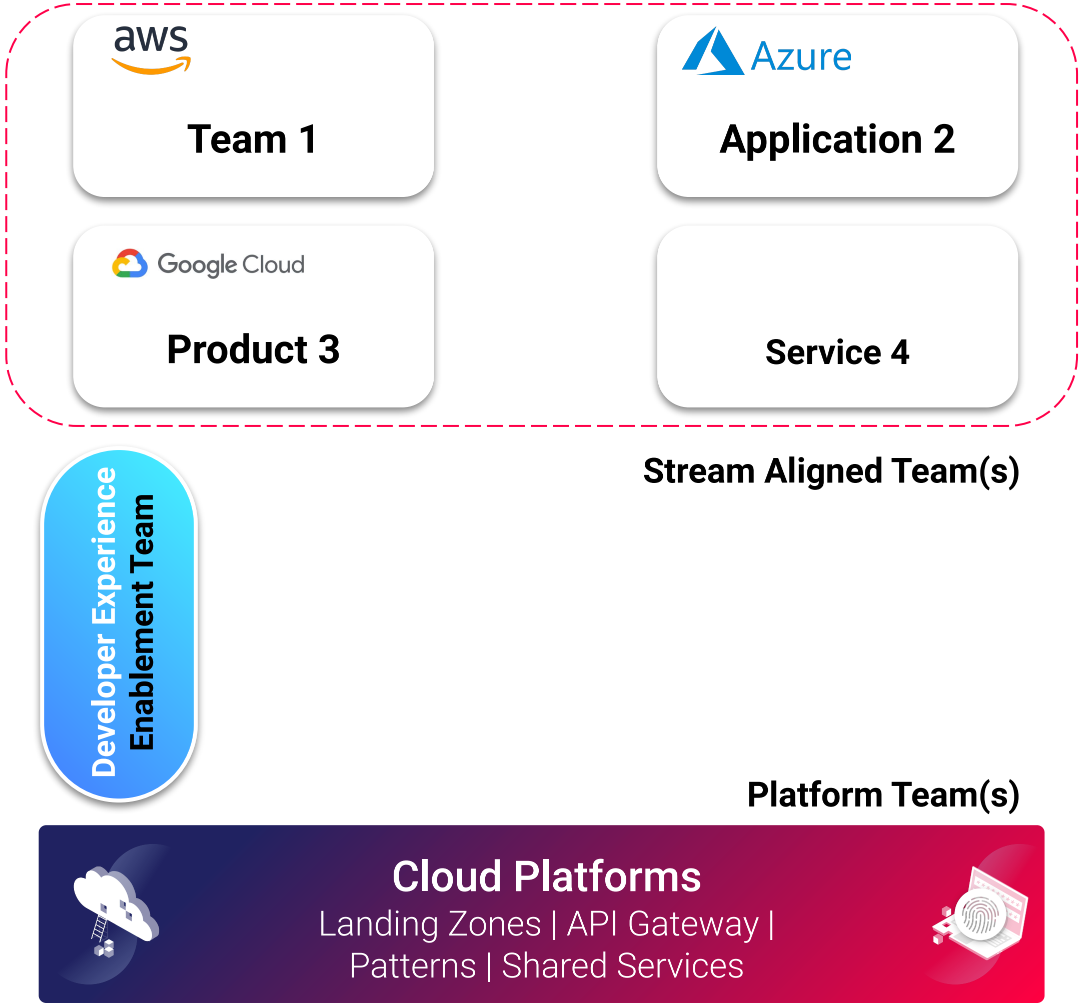
Team Topology



Customers: External facing services
Reliability: Service dependant

Customers: Internal Product/App teams
Reliability: Quite high!

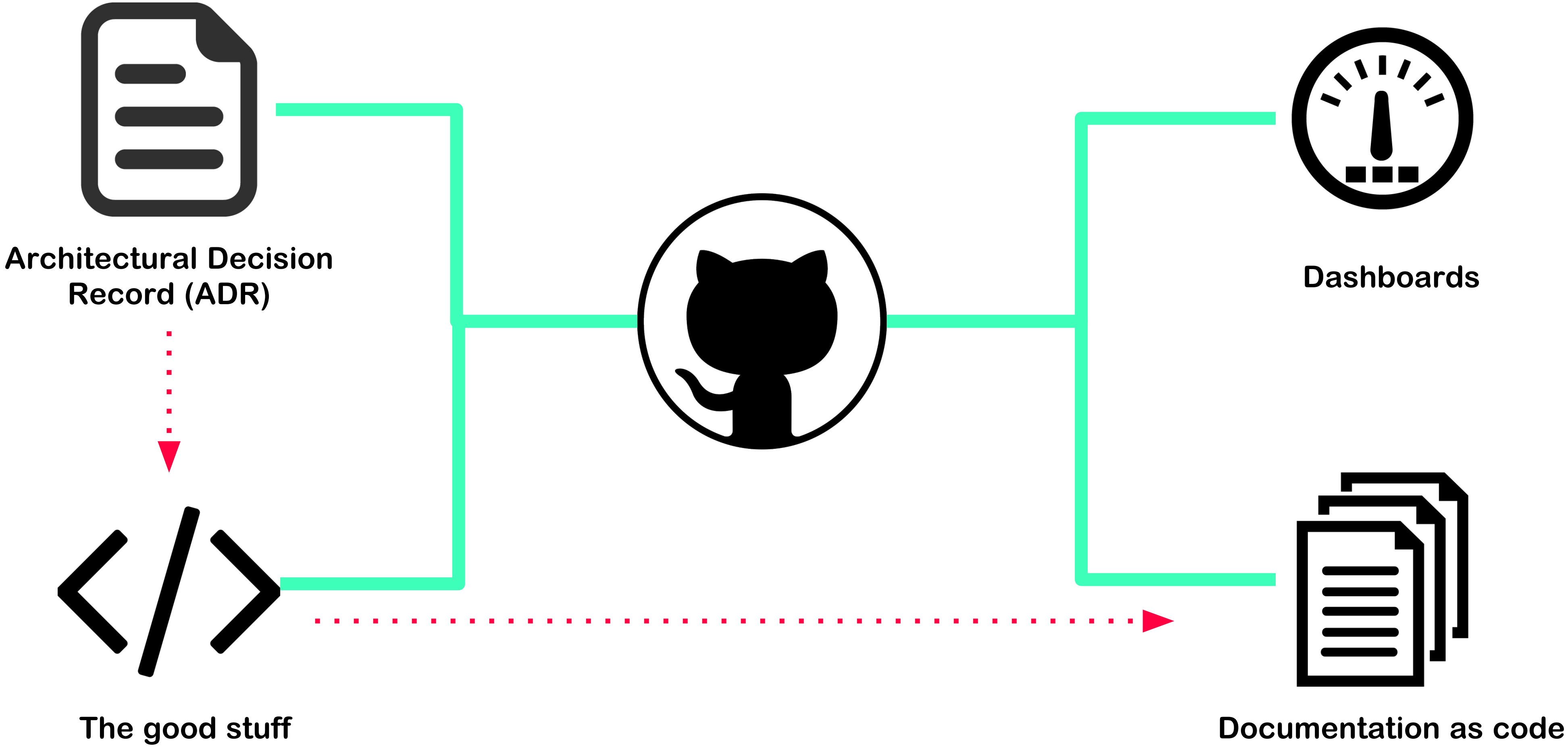
Team Topology



Mandates for Success

- Defining the "Team API" for Platform Teams
- Cloud Platforms enable the demand from the stream aligned teams, not dictate them
- Platforms Teams roadmap and priorities are public and open
- Platforms Teams have wiki based documentation available for consumption
- Team dependencies should be reduced to enable Flow
- Developer experience metrics should be tracked for Platform Teams, along with the 4 key metrics
- Technical Decisions have to be made based on Team Cognitive Load
- The Three key Interaction Models: Facilitate, Collaborate, X as a Service

Code-Docs-Controls-Reliability

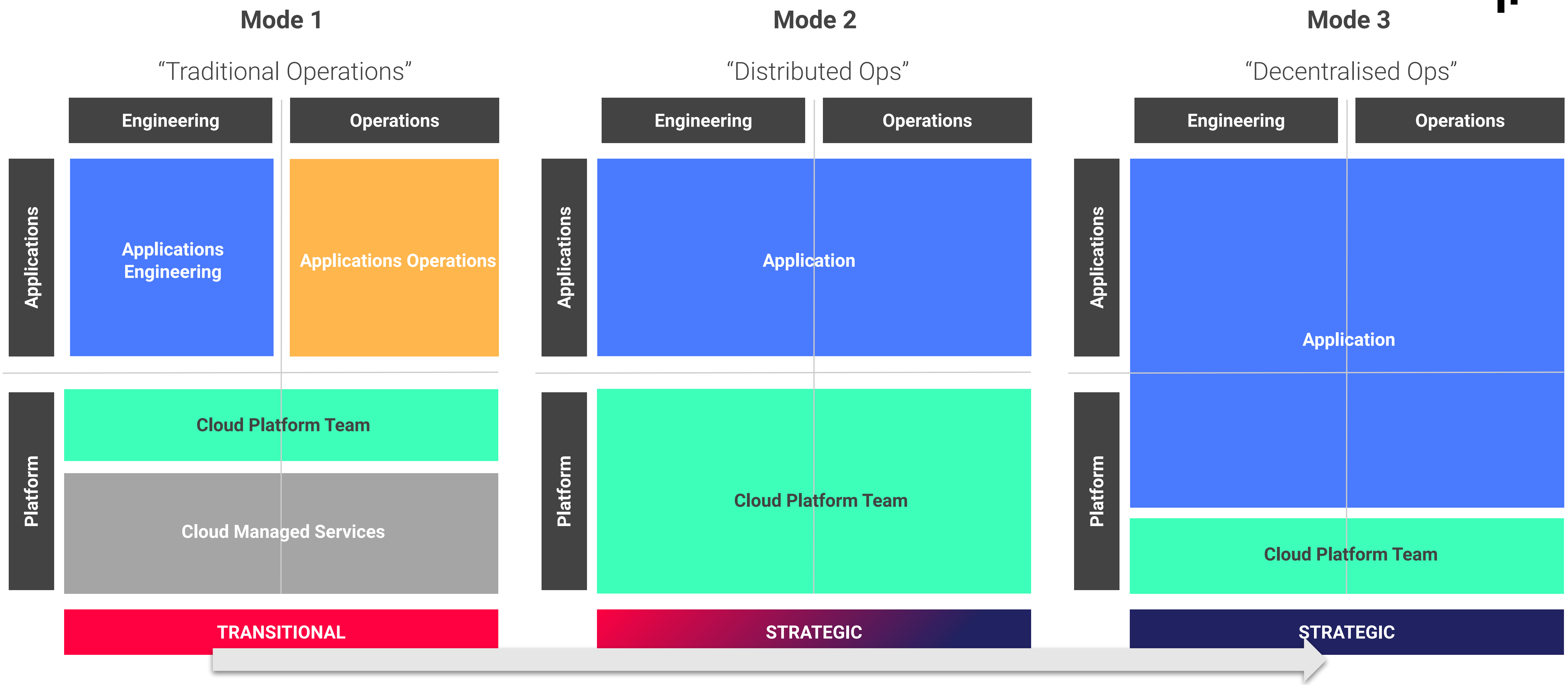


Architectural Decision Record (ADR)

Dashboards

The good stuff

Documentation as code



Mode 1

“Traditional Operations”

Engineering

Operations

Applications

Applications Engineering

Applications Operations

Platform

Cloud Platform Team

Cloud Managed Services

TRANSITIONAL

Mode 2

“Distributed Ops”

Engineering

Operations

Applications

Application

Platform

Cloud Platform Team

STRATEGIC

Mode 3

“Decentralised Ops”

Engineering

Operations

Applications

Application

Platform

Cloud Platform Team

STRATEGIC

How get organised and started?

- Kitchen Sink, a.k.a. “Everything SRE”
- Infrastructure / Platforms
- Tools
- Product/application
- Embedded
- Consulting

https://cloud.google.com/blog/products/devops-sre/how-sre-teams-are-organized-and-how-to-get-started

What to Measure

- Infrastructure / Platforms
  - Developer Experience / Developer Velocity
    - Function - Stability - Ease of Use - Clarity -
- Product/application
  - DORA Metrics:
    - Lead Time for Changes - Change Failure Rate - Time to Restore - Deployment Frequency Availability

How to upskills a team towards SRE

Get the basics right

- Find out what type of SRE team you are
- Start with a small product / platform
- Identify your customers
- Figure out what you need to measure
- Agree the metrics WITH your customers
- Mesure - Improve - Fail - Improve - Measure

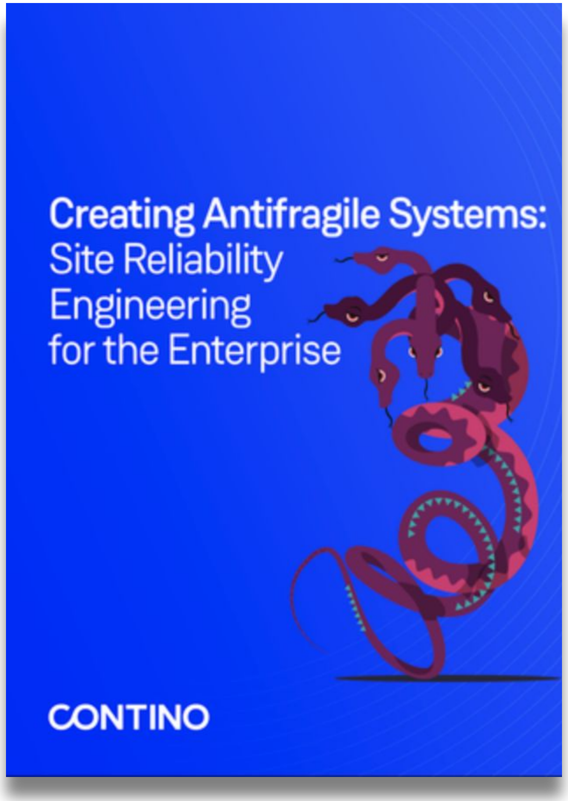
How to scale

- Align engineering OKRs to the organisation metrics
- Start running chaos engineering practices across teams for product improvements
- Start benchmarking your SRE maturity
- Use engineering mitosis to scale SRE practices

What is a Game Day?

> “ A game day simulates a failure or event to test systems, processes, and team responses. The purpose is to actually perform the actions the team would perform as if an exceptional event happened.

- To test whether or not the current systems are more or less resilient; with the adequate processes to support it
- To build the "muscle memory" of the team(s) on how to respond if an exceptional event happened.
- To evaluate a team's ability to design-build-run systems taking into consideration the well architected pillars of operations, security, reliability, performance, and cost.
- To test all aspects of one’s business for Reliability Readiness; specifically operations, test, development, security, business operations, and business leaders.
- Game Days are run in order to improve the availability of a system with the goal of increasing reliability by purposefully creating major failures on a regular basis.


Creating Antifragile Systems: Site Reliability Engineering for the Enterprise

CONTINO

- https://landing.google.com/sre/
- https://www.gremlin.com/blog
- Reliability Engineering at the Core of Continuous Innovation
- Boost Your Apps With An SRE Approach to Development

THANK YOU TO OUR SPONSORS

NORTHROP GRUMMAN

sonatype

StateFarm™

CloudBees®

Google Cloud

NowSecure™

D2IQ

ConnectALL
An Orasi Company

moogsoft

NS1.

Carnegie Mellon University
Software Engineering Institute

CONTINO

ATARC

MACQUARIE

DZone
A DEVADA MEDIA PROPERTY

MediaOps

TechBeacon

SD Times

Forcepoint