

Policies & Contracts In Distributed Systems

Conf42 Site Reliability Engineering Conference
4th May 2023

Prathamesh Sonpatki

[Last9.io](https://last9.io)

[@prathamesh2_](https://twitter.com/prathamesh2_)

As a Developer

- I want to write more code
- I want to fix all the (existing*) bugs
- I want to write bug free code
- I want to use latest and greatest tools
- I want to integrate with best in class tools
- But

As a DevOps Engineer

- I want to make sure that infrastructure scales
- I want to make sure that application utilizes resources efficiently
- I want to make sure that infrastructure cost is not out of control
- I want to make sure
- But ...

As a TL/EM/DL

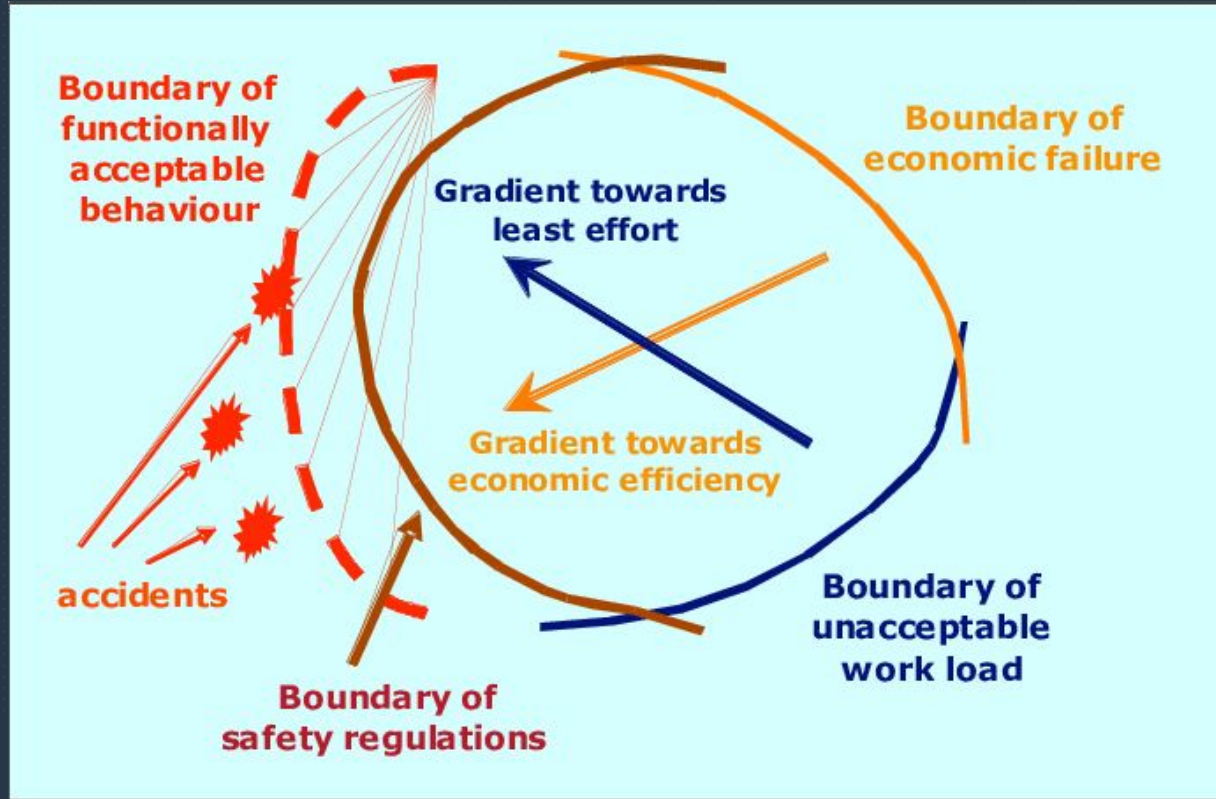
- I want to make sure that my team meets the deadlines
- I want to make sure that features work as expected
- I want to make sure that code/infra is performant enough
- I want to make sure that tech debt is not out of control
- I want to make sure that team is motivated enough
- I want to make sure
- But ...

As a Business/Product leader

- I want to make sure team velocity is not slow
- I want to make sure that external commitments are met
- I want to make sure that product is getting adopted
- I want to make sure that customer needs and expectations are considered by the product and engineering team
- But ...

But!

Rasmussen's model of how accidents happen



BOOM!

Failures will happen..

- Every stakeholder has a boundary and a limit
- One of the stakeholders pushes other's boundaries too much!

Eg.

- Business pushing for features rollout resulting into too much of tech debt for engineering
- Engineering chasing perfection slowing down delivery and velocity

Boundaries

Something that indicates or fixes a limit or extent

Boundaries...

- Team constraints
- Quality of work
- Time?
- Perfection?
- Cost?
- Pricing?
- Time to market?

Negotiation

- We will be able to release this but with few bugs
- We can do this but with increased AWS bill 💰
- We will be roll this out to new customers if team works overnight on weekend
- We will be able to fix those bugs causing that main customer to go away if we de-prioritize the feature pipeline
- We will be able to move faster if we add one more backend developer to the team

Negotiation → Contracts





A written or spoken enforceable agreement



API Interface

A written agreement about what the endpoint will return

Promises of POST /users endpoint

-  HTTP Status 201
-  HTTP Status 400
-  HTTP Status 401
-  HTTP Status 404

Runtime interface

A written agreement about how the endpoint will behave at runtime

Promises of POST /users

- Uptime 90%
- 10% requests are allowed to fail
- Every Weekend, 20% requests are allowed to fail
- During peak hours, the Latency can vary between 1000ms-5000ms

Runtime Interface

Service Level Objectives

Service Level Objectives

- Availability will be $> 99.99\%$ over 1 Day
- Latency will be < 4000 ms over 7 Days
- Uptime will be 98% over 7 Days

Service Level Objectives

- What is the error rate on this checkout flow?
- Can we promise 99.9% availability to this enterprise customer?
- Should we prioritize tech-debt over new features?
- Where should engineering focus for the next sprint?
- What's the success rate of this payment gateway?

Runtime ~~Interface~~

Promises



Service Level Objectives

Promises → **Policies**

Policies

- Set right expectations on what's possible
- Buy in from multiple stakeholders
- Framework for communication between stakeholders
- External client communication
- Helps in Build v/s Buy decisions

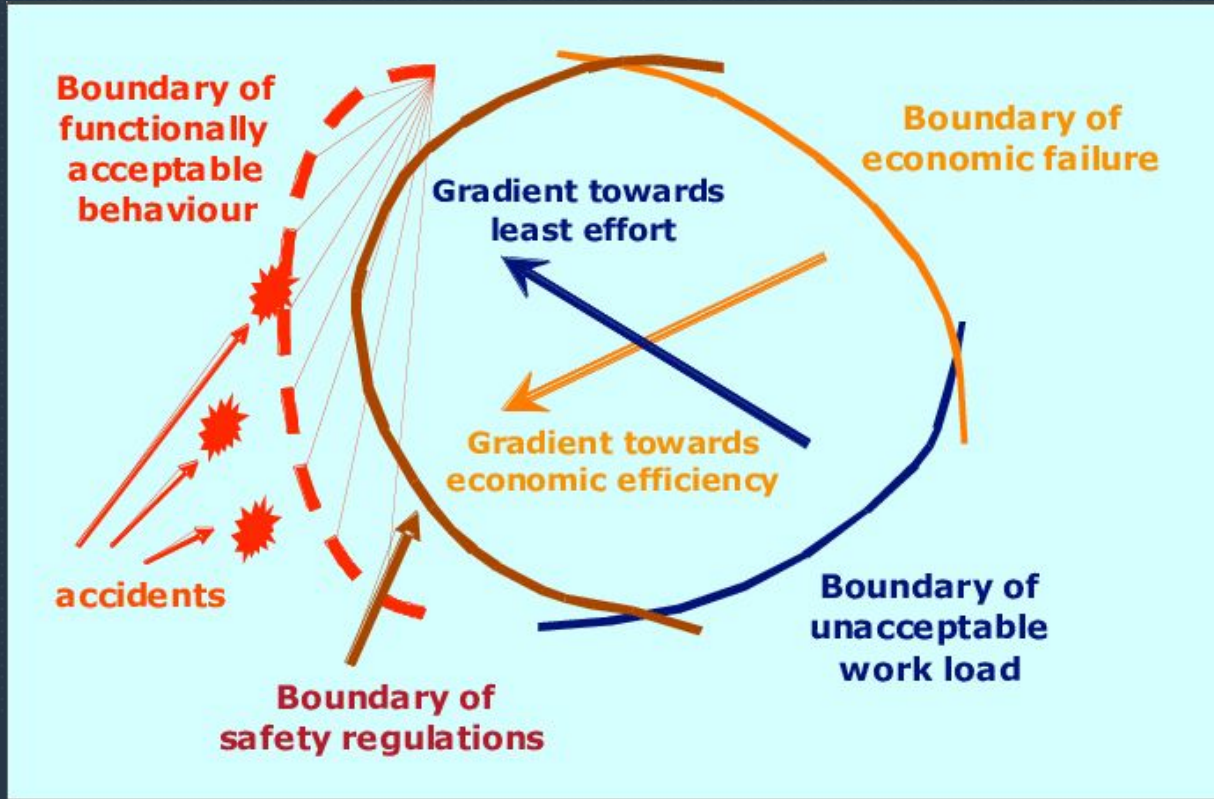
Tiered Services

- P0, P1, P2
- Different expectations from different tiers
- Not every service is priority!


Ladder of Reliability

- You can't improve what you can't measure
- First Baseline!
- Go one ladder at a time
- 90% → 95% → 99 % ✓
- 90% → 99.999% 😭

Rasmussen's model of how accidents happen



Boundaries (still) exist

But not broken! The tension(contracts and policies) keeps them in balance 



Thanks 🤝

Prathamesh Sonpatki

📁 [Last9.io](https://last9.io)

✍️ prathamesh.tech

🐧 twitter.com/prathamesh2_

🐘 hachyderm.io/@Prathamesh

[“Last9 of Reliability”](#) Discord

