Resilient Software Design

Pt 1: Introduction

Bugsquashing Seminar
Dominique Cheray, Janek Gröhl
Welcome to the DKFZ!
• Distributed Systems

• Service Oriented Architectures

• Microservices

• Resilient Software Design
Context

- Distributed Systems
- Service Oriented Architectures
- Microservices
- Resilient Software Design
“We will prepare for the armies of illogical users who do crazy, unpredictable things”

- Michael Nygard
1. the power or ability to return to the original form, position, etc., after being bent, compressed, or stretched; elasticity.

2. ability to recover readily from illness, depression, adversity, or the like; buoyancy.
Main Concept

- $\text{Availability} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$

- MTTF = Mean Time To Failure
- MTTR = Mean Time To Recover
Resilience:

- Ability of a system to handle unexpected situations
- Best case: without the user noticing it
- Worst case: with a graceful degradation of service
Example

Example

- Simian Army
- Fault Injection Testing
Example

- Chaos Monkey
- Arbitrarily shuts down
  - Services
- At peak usage times

System needs to handle this
- Chaos Gorilla
- Arbitrarily shuts down
  - Servers
Example

• Chaos Kong

• Arbitrarily shuts down
  • Server Cluster Regions
Thank you for your attention!

Questions?