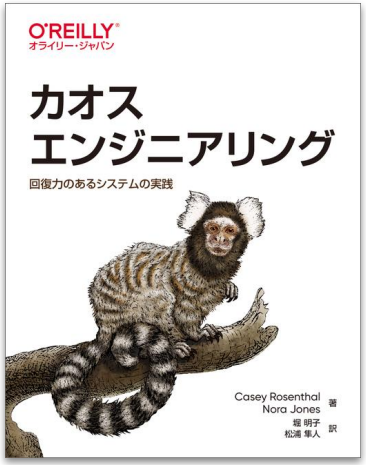
カオスエンジニアリングの裏話

JaSST'23 Tokyo

Meiko Hori & Hayato Matsuura

March 9th, 2023

自己紹介


O'REILLY
オライリー・ジャパン
カオス
エンジニアリング
回復力のあるシステムの実践
Casey Rosenthal
Nora Jones
著
堀 明子
松浦 隼人
訳

自己紹介

堀　明子

Customer Reliability Engineer @ Autify (Twitter: @m3iq)

- **経歴**
  - SIerに十数年所属。金融システムの開発 /保守に従事。
  - 2020年8月よりAutifyに参画
    - 自動テストプラットフォームのお客様支援
      - 対面でのサポート、利用データ基盤整備、ドキュメンテーション etc…
- **趣味**
  - 🎵 絵と音楽（作る/歌う/聴く）
  - 🏜️ 旅と食（北アフリカから中米まで）
  - 🎮 スプラトゥーンはスクイックリン

自己紹介

松浦 隼人

Engineering Manager @ Autify

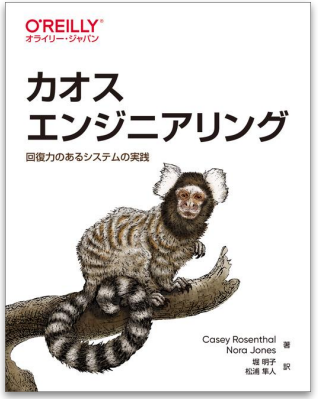
- Twitter : dblmkt
- **経歴**
  - インフラ何でも屋
  - 某ブログサービスのインフラ担当
  - テクニカルサポート
  - インフラ + Rails
  - Engineering manager
- **趣味**
  - 翻訳

本日のテーマ

カオスエンジニアリングの裏話

書籍「カオスエンジニアリング」

翻訳を通じて注目したワードを深掘り



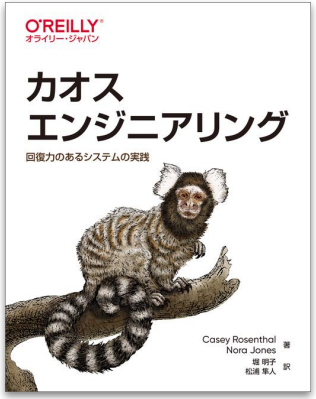
どんな本？

Casey Rosenthal氏・Nora Jones氏

および多くの寄稿者によって書かれた

カオスエンジニアリング の

原則・事例・応用が網羅された書籍



注目したワード

テスト vs 実験
Testing vs Experiments

定常状態
Steady state

妥当性確認 vs 検証
Validation vs Verification

影響範囲
Blast radius

注目したワード

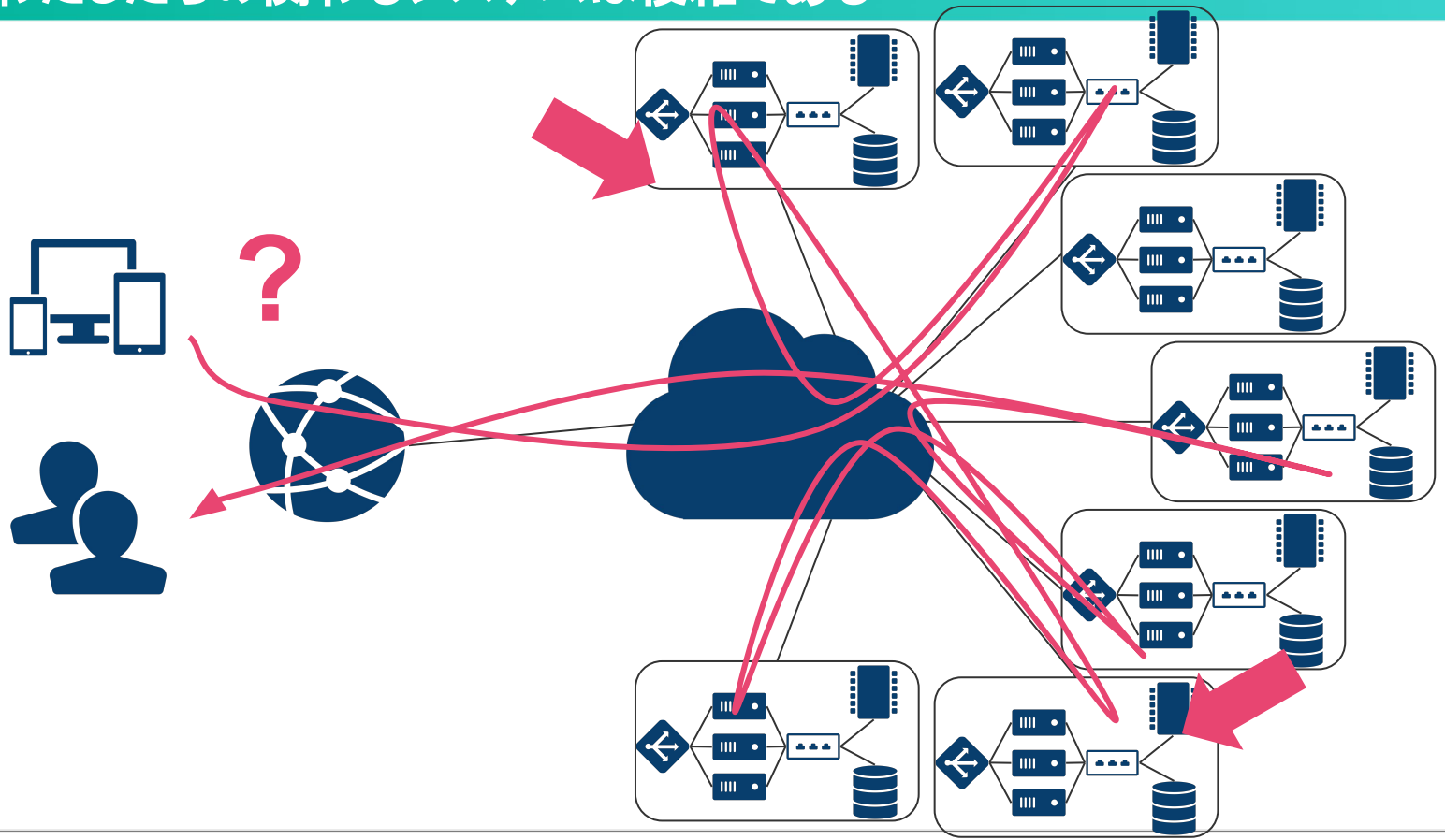
テスト vs 実験
Testing vs Experiments

定常状態
Steady state

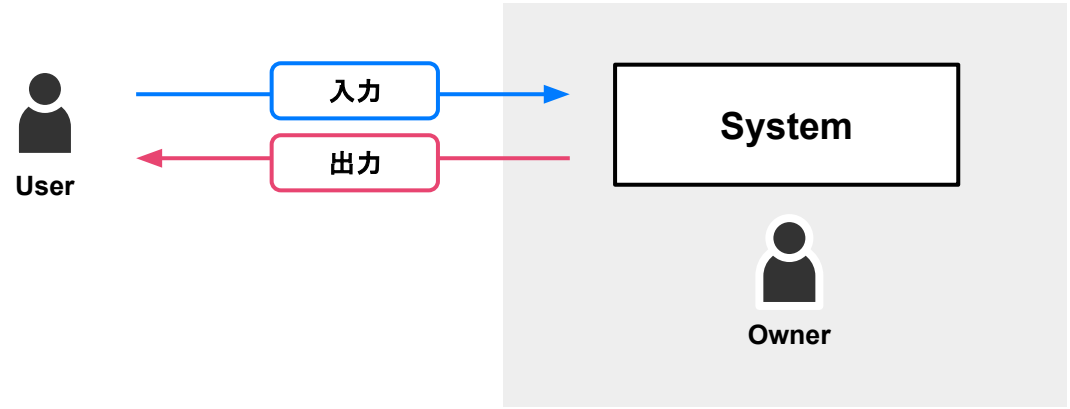
妥当性確認 vs 検証
Validation vs Verification

影響範囲
Blast radius

わたしたちの関わるシステムは複雑である



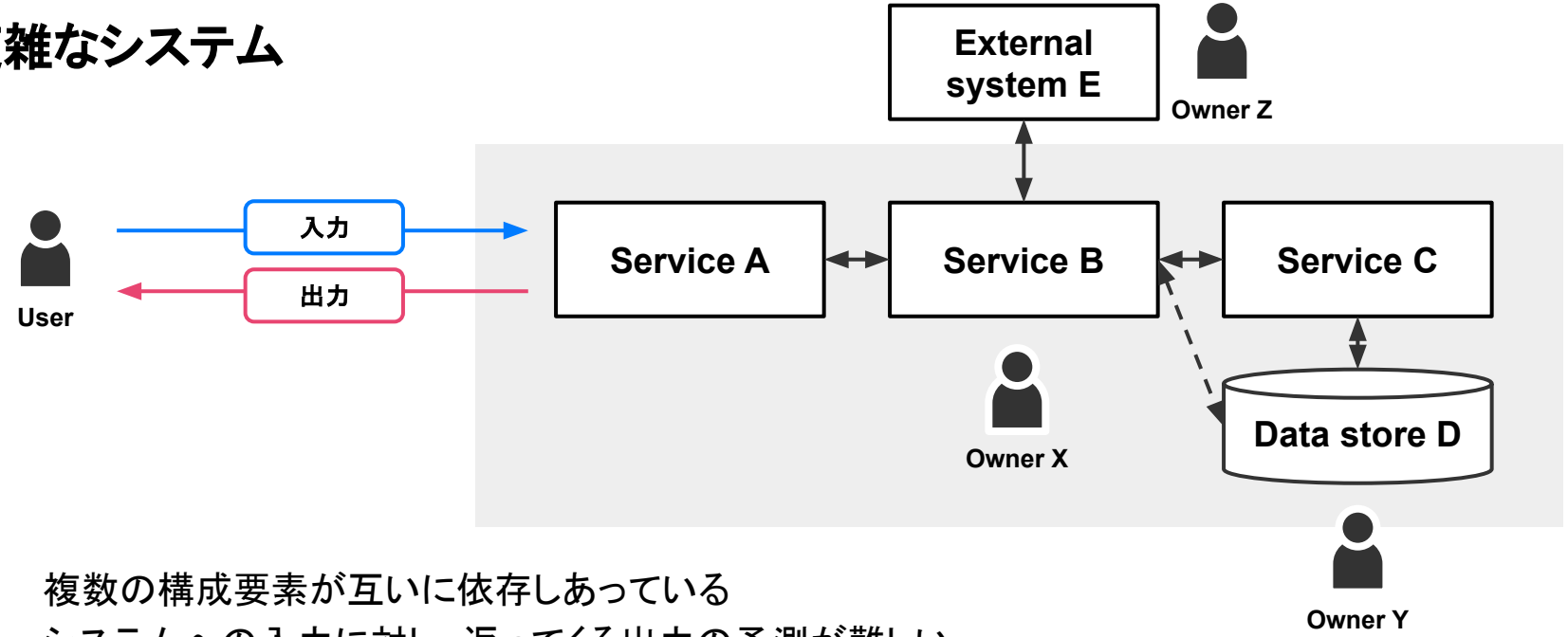
シンプルなシステム



- システムへの入力に対し、予測が容易な出力が返ってくる
- 構成要素に依存関係が少ない

複雑なシステム vs シンプルなシステム

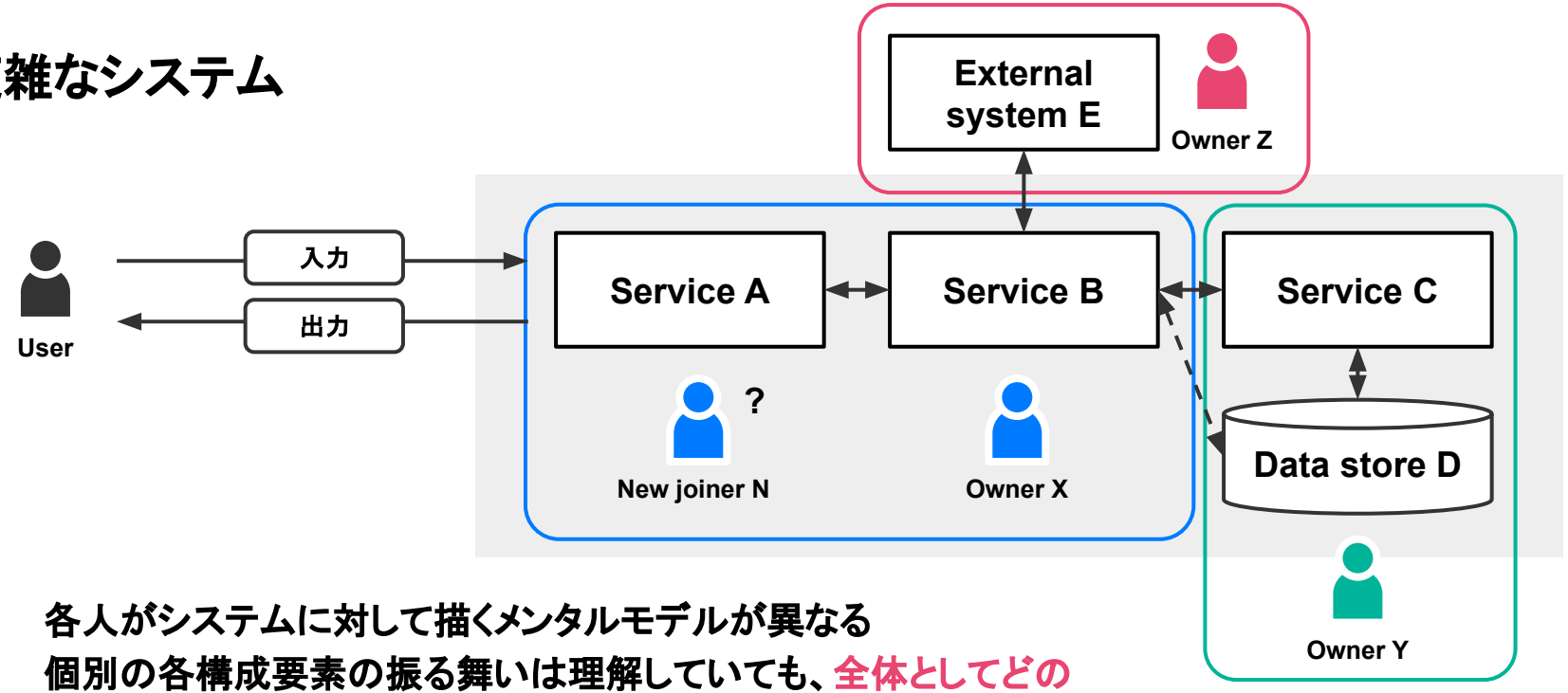
複雑なシステム



- 複数の構成要素が互いに依存しあっている
- システムへの入力に対し、返ってくる出力の予測が難しい

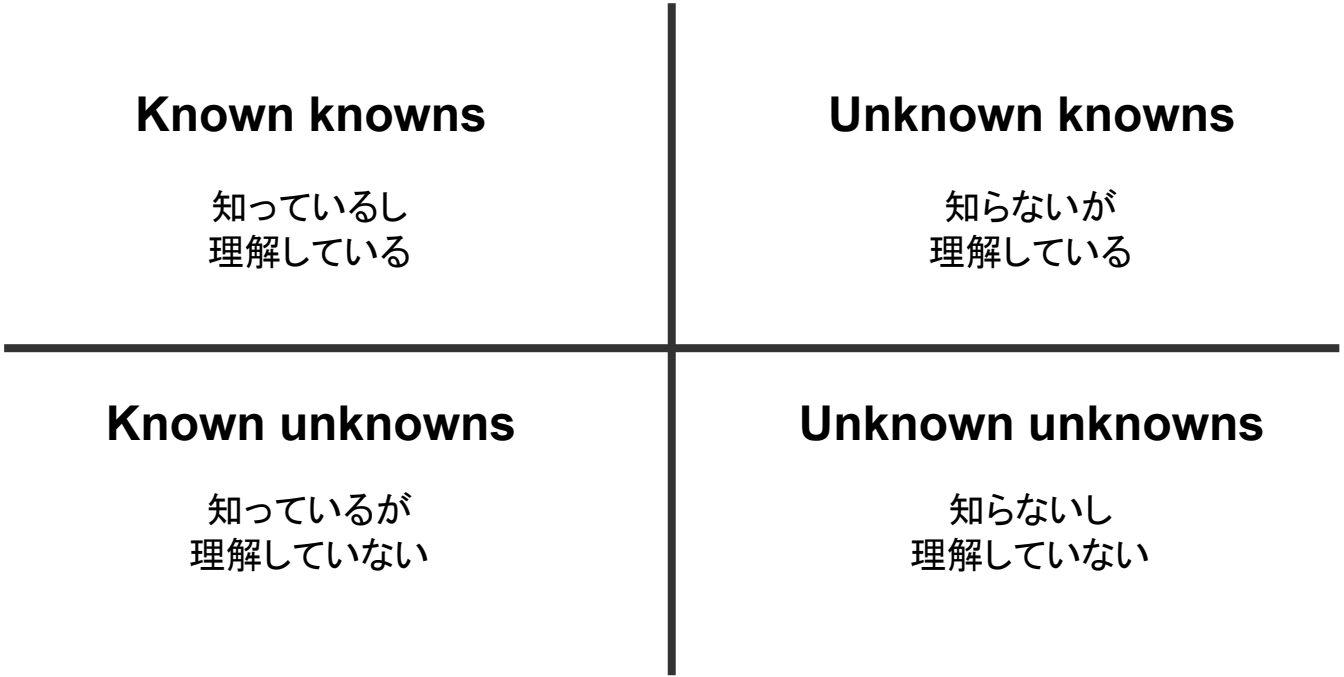
複雑なシステム vs シンプルなシステム

複雑なシステム



- 各人がシステムに対して描くメンタルモデルが異なる
- 個別の各構成要素の振る舞いは理解していても、全体としてどのような振る舞いをするかを誰も把握しきれない

Unknown unknownsに対処する



Unknown unknownsに対処する

テスト

直感

Known knowns

知っているし
理解している

Unknown knowns

知らないが
理解している

Known unknowns

知っているが
理解していない

監視

Unknown unknowns

知らないし
理解していない

カオスエンジニアリング

オブザーバビリティ

実験 (experiments) vs テスト(tests)

テスト (testing)

- システムについて**既に知られた**特性が、期待通りであることを確認する
- 主に、検証を行いたい対象が明らかになっているものに効果的

実験 (experiments)

- 仮説を立て、その仮説の反証を試みる
  - 反証されなかった場合は、仮説の信用度が上がる
  - 反証された場合は、新たなシステムの特性に関する学びが得られる
- **システムの 未知の特性 を見つけるのに効果的**

カオスエンジニアリングの
アプローチは主に実験寄り

妥当性確認 (validation) vs 検証 (verification)

妥当性確認 (validation)

- 個々の要素が期待通りに実装されていることの検証に重きを置く
- 水道水の**妥当性を確認**するには
  - 水道の配管や浄化、給水のシステムなど、水が届けられる過程にある全ての要素が期待通りの状態であるかを検査する

検証 (verification)

- **ビジネスケースと総合的な出力 に重きを置く**
- 水道水を**検証**するには
  - 蛇口から出てくる水（出力）が正常かどうか水質検査する

カオスエンジニアリングのアプローチは主に検証、ベリフィケーション寄り

カオスエンジニアリングの視点

総合的な出力 に注目し、**未知の特性** を発見する

注目したワード

テスト vs 実験
Testing vs Experiments

定常状態
Steady state

妥当性確認 vs 検証
Validation vs Verification

影響範囲
Blast radius

カオスエンジニアリングの発展原則

1. 定常状態に関する仮説を立てる
2. 実世界の事象を多様化させる
3. 本番環境で実験する
4. 継続的に実行できるよう実験を自動化する
5. 影響範囲を局所化する

カオスエンジニアリングの原則
https://principlesofchaos.org/ja/

書籍「カオスエンジニアリング」P43〜

カオスエンジニアリングの発展原則

1. 定常状態に関する仮説を立てる
2. 実世界の事象を多様化させる
3. 本番環境で実験する
4. 継続的に実行できるよう実験を自動化する
5. 影響範囲を局所化する

原則1

- **定常状態に関する仮説を立てる**
  (Build a Hypothesis around Steady State Behavior)
  - **Steady state 定常状態**
    - 自然科学一般の用語
      - 時間が変化しても変わらない状態
      - システムにおける期待される状態
    - コードが正しく動くか、ではなくシステムの状態が正しいか
      - 出力に注目する
      - 根本原因(どうしてそうなるか)に注目しすぎない
      - 明確なビジネス優先度を追跡するメトリクスに注目する

原則2

- 実世界の事象を多様化させる
  (Vary real-world events)
  - Vary：変わる、変える、多様化させる
  - 容易な変数・イベントを選ばない
  - よりハイレベル(= ユーザに近い)変数を変化させる

原則3, 4

- **本番環境で実験する**
  (Run experiments in production)

- **継続的に実行できるよう実験を自動化する**
  (Automate experiments to run continuously)

原則5

- 影響範囲を局所化する
  (Minimize blast radius)
  - Blast Radius 影響範囲
    - 直訳すると爆発半径
    - 何かが爆発した時のその影響が及ぶ範囲の半径
    - 実験が影響する範囲
      - ユーザ体験に注目

原則5

- 影響範囲を局所化する
  (Minimize blast radius)
  - 制御グループと可変グループを作って比較する
    - シャドートラフィック
    - 大口のトラフィックを除外する
    - リトライを実装して実験によって失敗したリクエストが影響しないようにする
    - サンプリング、スティッキーセッションなど
  - グループが小さくなる
    - グループ内のメトリクス変化が目立つようになる

注目したワード

テスト vs 実験
Testing vs Experiments

定常状態
Steady state

妥当性確認 vs 検証
Validation vs Verification

影響範囲
Blast radius

Q & A

ご清聴ありがとうございました！