

# 複数リポジトリ環境下の バージョン管理システム 構築と運用

めぐろ LT #28  
株式会社カンリー 角谷維



## 👤 About Me

- **Name:** 角谷 練(すみや たもつ) / motsuo
- **Company:** 株式会社カンリー
- **Role:** web エンジニア / 一応フロント
- **Like:** AI 系 / 音楽 / ポーカー



# エンジニア本部について

## なぜ複数リポジトリが必要？

- ・ マイクロサービス化の進展
- ・ チーム分割によるリポジトリ分離
- ・ 言語・技術スタックの違い
- ・ デプロイサイクルの独立性

## よくある構成例

```
frontend/ → React App  
backend-api/ → REST API  
backend-job/ → Batch Jobs  
infra/ → Terraform
```



## Case 1: 依存関係の地獄

- API のバージョンアップ時にフロントエンドが追従できず
- 本番環境で互換性のないバージョンの組み合わせをデプロイ
- 結果: サービス停止 3 時間



## Case 2: リリースタイミングのズレ

- 複数チームの調整不足
- 機能追加の片方だけがリリースされる
- 結果: 不完全な機能が本番に露出

## 🎯 学んだこと

- コミュニケーションが最重要
- 自動化できる部分は徹底的に自動化
- モニタリングとロールバック戦略は必須
- 段階的リリースの仕組み作り

## 次回予告

具体的な解決策とツールの選定について詳しくお話しします！

# ご清聴ありがとうございました！

質問・相談はお気軽に



Twitter: @your\_twitter

GitHub: github.com/your\_github

Email: [your.email@example.com](mailto:your.email@example.com)