



PostgreSQL から **TiDB** へ 分散型 DB 移行の意思決定と実践

AI DevEx Conference 2026

- Future of Development Productivity -

ABOUT THIS SESSION

このセッションについて

WHO IS THIS FOR

分散型 DB への移行に関心のある方へ

— 現場視点のセッションです

急激なユーザー増加とトラフィックスパイクにより **DB のスケール限界** が見え始めていました。

PoC から本番運用までの実体験をもとに、**なぜ移行を決断したのか**、SQL 互換性や性能調整で直面した課題、AI を活用した開発プロセスをどう取り入れたのかを、パネルディスカッション形式で紹介します。

MODERATOR

波多野 謙介

株式会社コラボスタイル CIO

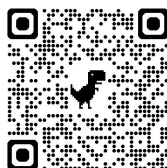
プロダクト開発の面白さに突き動かされ、営業、サポート、プロジェクト管理等、プロダクトのあらゆる側面に関わってきた元文系エンジニア。

限られた時間で自分の想いを形にできるAIコーディングにより、エンジニアとしての興奮が再到来。

眠れない日々を送っている。趣味は酒蔵巡り。



@hatanowf



SPEAKER 01

鬼頭 昌孝 (m.kito 🍡)

株式会社コラボスタイル 開発部 グループマネージャー

創業からのメンバーの一人で、プロダクト開発に従事。
スクラム開発やアプリケーションのアーキテクチャ設計、
AI-DLC を好んでいる。

- ・ 生粋の名古屋人、面白いことが好き
- ・ JAWS DAYS 2026 実行委員
- ・ 好きな漫画：キングダム & 蒼天航路🔥

📧 @mkitokitom



SPEAKER 02

後藤 佳汰

株式会社コラボスタイル 開発部 インフラエンジニア

工場勤務からエンジニアへ転身後、Web 制作から
インフラ・クラウド領域へと技術の幅を広げ、奥深さに
魅了される。

プロダクトを支えるインフラエンジニアとして、
TiDB への移行プロジェクトを牽引。

日々の業務で得た TiDB の知見やインフラ構築の
ノウハウを発信中。

✕ @kgto182123



THEME 1

なぜ **TiDB** への移行を 決断したのか？

01

単なるインフラ移行ではなく、

今後のサービス価値提供や成長も視野に入れた DB 移行戦略

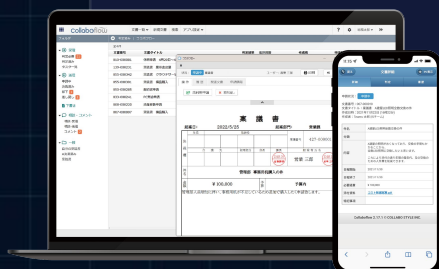
SOFTWARE AS A SERVICE

社内外の業務をつなぎ、情報収集から承認までをシームレスに支えるプラットフォーム

直感的に使える高機能ワークフローシステム



- 社内の承認・業務プロセスをデジタル化
- ノーコードで柔軟なワークフロー設計
- 現場主導で継続的に改善
- API・Webhook によるシステム連携



導入社数 **2,000** 社以上

ITreview ワークフローシステム部門
顧客満足度 **5年連続 No.1**

ワークフローとつながる申請文書受付システム



- 社内外の情報収集・受付をデジタル化
- ノーコードで柔軟なフォーム設計
- 社内・社外との接点をデジタル化
- コラボフロー・外部をシームレスに連携

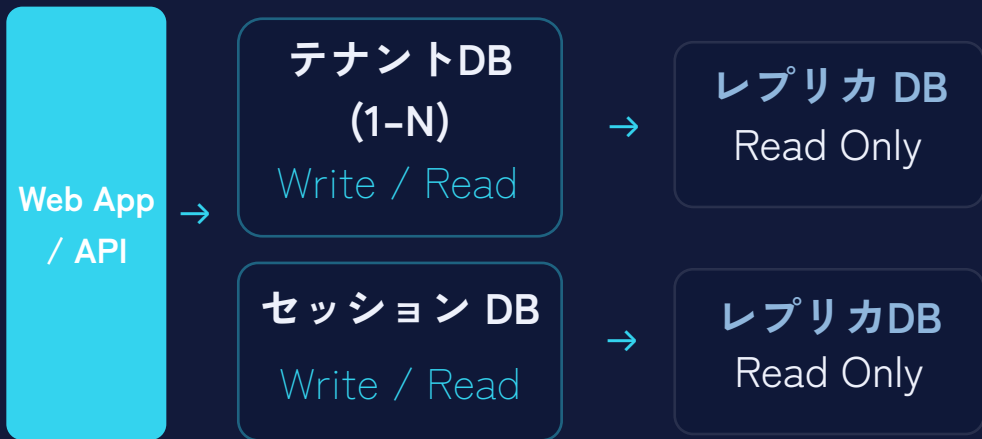


ARCHITECTURE

2つのプロダクトにおける DB 構成

コラボフロー：テナント分離型

- テナントごとの専用 DB
- セッション管理専用の DB を配置
- 高いテナント分離性とセキュリティ

 **COLLABO FLOW**

コラボフォーム：機能分離型

- サービス間の共通情報の管理 DB
- サービス固有の情報の管理 DB
- データの整合性と効率的なアクセス

 **COLLABO FORM**

- AWS Aurora PostgreSQL Serverless / 全データベースレプリケーション構成

PROBLEM → EXPECTATION

なぜ PostgreSQL から TiDB への移行したのか

直面していた課題

- 1 検索・集計パフォーマンスの劣化
- 2 書き込み負荷の垂直スケール限界
- 3 レプリケーション遅延による障害リスク

TiDBへの期待

- 1 水平スケーラビリティによる容量拡張
- 2 アプリ側の R/W 振り分けコード修正が不要
- 3 高可用性と安定運用の両立
- 4 インフラコストの削減

課題の解決・安定性向上・コスト削減のために

THE DECISIVE FACTOR

スケールの遅延改善、分析・集計、 コスト最適化、自動シャーディング etc.

→ アプリケーションの PG 修正の大幅削減も大きな決め手の1つに

TiDB には負荷を自動的に分散する仕組みがあり、
Read／Write を振り分けるコード修正が“**不要**”になる

従来

アプリ側で Read／Write を
明示するコードの修正が必要



TiDB

自動で負荷分散
膨大なコードの修正が不要に

THEME 2

機能開発とDB移行 並行してどう進めたのか 02

移行戦略と AI 時代の組織的アプローチ

STRANGLER FIG PATTERN

段階的な移行戦略

安全かつ確実に、プロダクトを成長させながら移行を完遂

STEP 1

影響の少ない領域を
局所的にTiDB化

セッション DB など
影響の小さい所から着手

**STEP 2**

スモールサクセスを
積み上げる

対象範囲を段階的に
拡大していく

**FULL MIGRATION**

全体への展開と
安定稼働の実現

旧構成を置き換え、
全面的に TiDB へ



TiDB化の範囲 ■ PostgreSQL (既存) ■ TiDB (移行済)

少しずつ確実に“strangle”していく

STRANGLER FIG PATTERN

影響の少ない領域から局所的に **TiDB** 化

コラボフロー：テナント分離型

コラボフォーム：機能分離型

セッション・共通情報をTiDB化により自動負荷分散。リードレプリカ不要によってコストも削減



 **COLLABO FLOW**



 **COLLABO FORM**

RESULTS

移行後の成果：スパイクの改善による 安定稼働と構成の最適化

10倍の高速化

比較	実行時間の 短縮率（改善度）	スピード 向上倍率
最速時	80.0% 改善	5.0倍
平均値	86.7% 改善	7.5倍
最遅時	90.0% 改善	10.0倍

インフラ構成の最適化



PJ体制とAI時代の組織的アプローチ（AI-DLC）

固定化されたチームの壁を越え、初手AIで横断的な移行PJチームにて推進！



THEME 3

異機種DBへの移行に おける課題と対処

03

手を動かして直面した壁と解決策

APP & CONFIG OPTIMIZATION

実践の中で見えた課題と対処

データ型とスキーマ

TypeORMとの型マッピングの相違やPG特有スキーマの除去

TiDB対応
の型へ

外部キーと性能

分散DBの特性を考慮し、性能影響の大きい外部キーの扱いを見直す

アプリ側で
制約を担保

RLS の見直し

DB機能に依存したアクセス制御から脱却し、高い移植性を確保する

アプリ側で
アクセス制御

動的テーブル作成

アプリケーション実行時に動的にテーブルを生成する仕組みがあり、DBごとにテーブル構成が異なっている

転送ツールで
対応

AI による変換

AIによる自動変換が期待通りとならない場合があり、AIと人間による確実な目視の確認も必要に

AI変換と
二重検証

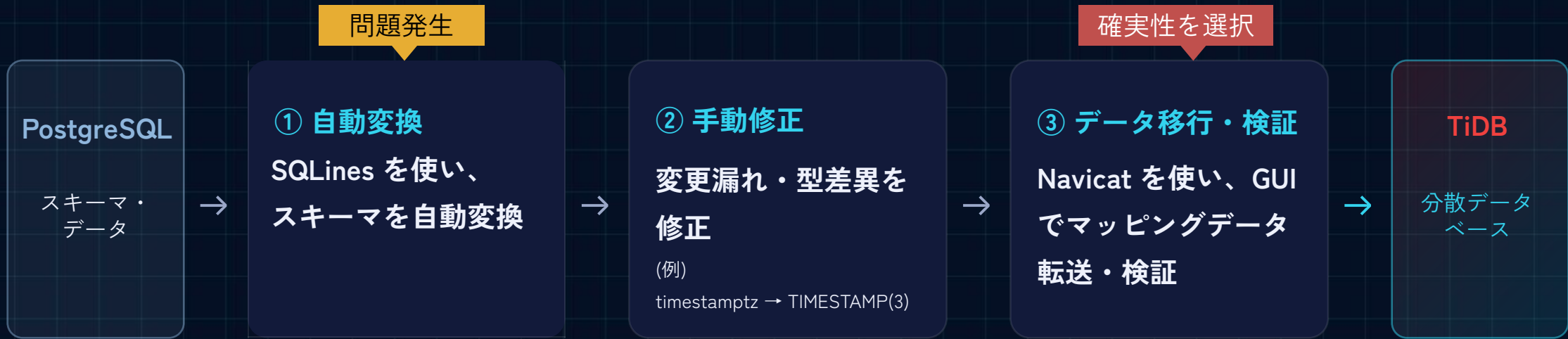
TRANSACTION

分離レベルは**悲観的トランザクション** + READ-COMMITTED を採用し、リトライ対応を実装

SCHEMA & DATA MIGRATION

確実性を重視した DB 移行プロセス

変換の処理不足を手動修正と目視確認する辛さも・・・

**課題**

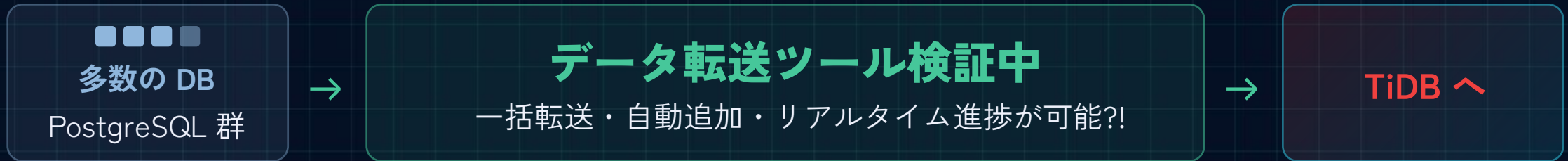
自動変換だけではスキーマや型を正しく
移行できず、手動補正と目視確認が不可欠だった。

- 確実な手法でスモールサクセスを積み上げる

Transport

大規模データ移行の課題に対するツールへの期待

手作業での移行を経験したからこそ辿り着いた、データレプリケーションツールを検証中



ツールに期待していること ⇨ 移行の課題

01

多数 DB の一括転送
・ 自動追加

02

ターゲットに TiDB
を指定可能

03

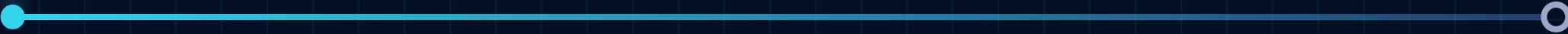
リアルタイムな
進捗管理

04

圧倒的な
転送速度

ROADMAP

本番はここから・・・多段階での移行戦略



▲ 現在：Stage 1 を検証中

次のフェーズ

STAGE 1

現在地

既存 DB 構成のまま
TiDB へデータ移行検証

構成は変えず、まず確実に移す（リフト&シフト）



STAGE 2

TiDB の特性に最適化した
DB 設計へリデザイン

サービスの保守性・拡張性を担保しながらも
分散DBの強みを引き出すテーブル構成へ再設計

多数のDB群を無停止・最小停止で移行するための検証も並行して推進中！

CLOSING

TiDB がもたらす価値

PostgreSQL からの移行は挑戦的ですが、適切なツール選定と柔軟な組織的アプローチと AI-DLC の活用で、未来のビジネス成長を支える確かな基盤作りをしていきます。

ご清聴ありがとうございました！