教育を中心にやりたいこと。（面白そうなことをしよう）

- Let’s try something exciting!

1. 何をするのか。

- GitHub Copilotを使用したプログラミング支援の解説
  生成AIによるプログラミング支援が様々な分野で注目されています。
  今回はGitHub Copilotを使用したプログラミング支援の解説を行います。
- Markdownを使用したドキュメントについて
- GitHubの新しい使い方
- K8sとPowerAppsを使用したアプリケーション開発

秀雄さんのように、PowerAppsをJava/MySQLやKubernetesと組み合わせて高度に運用する場合と、**通常のPowerApps運用（DataverseやSharePoint中心）**との違いを比較すると、以下のようなメリット・デメリットが見えてきます。

⚖️ 比較：通常のPowerApps運用 vs. 外部バックエンド＋K8s構成

観点	通常のPowerApps運用（Dataverse等）	外部バックエンド＋K8s構成（Java/MySQL）
🔧 開発の容易さ	GUI中心でノーコード/ローコード	API設計・K8s構成・セキュリティ設計が必要
🧠 学習コスト	Power Platformの知識で完結	Java/K8s/CI/CD/セキュリティの知識が必要
📦 データ管理	DataverseやSharePointで一元管理	MySQLや外部DBで自由に設計可能
🔌 拡張性	Power Platform内での拡張が中心	任意の言語・フレームワークで拡張可能
🔐 セキュリティ	Azure ADと統合、RBACが容易	独自認証設計が必要（OAuth2, JWTなど）
📱 モバイル対応	自動対応（PowerAppsアプリ）	API連携次第で柔軟に対応可能
🚀 スケーラビリティ	Dataverseの制限あり	K8sで自由にスケール可能
🔍 ロギング・監視	Power Platform内で限定的	Prometheus/Grafanaなどで詳細監視可能
💰 コスト	ライセンス体系に依存	Azureリソース＋K8s運用コストが発生

☑️ 外部構成のメリット（Java/K8s/MySQL）

- **既存資産の活用**：社内のJavaコードやMySQLデータをそのまま活かせる
- **柔軟な設計**：複雑なロジックや独自認証を自由に実装可能

- **DevOps対応**：CI/CDやGitOpsで運用効率化
- **スケーラブル**：K8sで負荷に応じた自動スケーリングが可能

⚠ 外部構成のデメリット

- **初期構築が複雑**：API設計、セキュリティ、K8s構成に時間がかかる
- **運用負荷**：クラスタ管理、障害対応、セキュリティ更新が必要
- **PowerAppsとの接続設計**：Custom Connectorの設計と保守が必要

🎯 どちらを選ぶべきか？

- **業務部門主導の簡易アプリ** → 通常のPowerApps運用（Dataverse中心）
- **エンジニア主導の業務システム連携** → Java/K8s/MySQL構成
- **既存システムとの統合が重要** → 外部構成が有利
- **スピード重視・PoC段階** → PowerApps単体で構築し、後からAPI連携

もし、Zennでこの比較を記事化するなら「PowerAppsで始める業務アプリ開発：クラウドネイティブ vs. フルスタック連携」みたいなタイトルが映えそうですね。テンプレートや構成案もお手伝いできますよ。どう展開しましょう？