

AIは「組織の鏡」だった。

～AIDDを挫折させないための情報整理とクラウドガバナンス～

クラウドエース株式会社
第四開発部 小林由暁



自己紹介



小林 由暁 / よっしー

クラウドエース株式会社

第四開発部 セルリーダー

#Google Cloud

#AIDD実践

#生成AI

#Google Cloud Partner Top Engineer 2025/2026

これまで作ってきた社内向けAIDDアプリの一例

プロジェクト管理アプリケーション

チームマップ

稼働管理シート

Google Cloud資格勉強アプリ

タスク積み重ね



この講演で伝えたいメッセージ

AIDD の壁は AI 性能差ではなく「社内データ」と「広げ方」にある

01

データを整える

所在・意味・正確性が見える化



02

小さく速く出す

Cloud Runで軽い展開を選ぶ

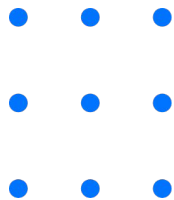


03

安全に広げる

認証・Secret・監査を共通化

運用者の役割はAIDDを止めることなく、続けられる条件を整えること。



データを整える

最初の壁は「どこに何のデータがあるか分からない」ことでした

AIの性能以前に、探索コストが開発速度を支配する

01 所在が分からない

どこに何の社内データがあるか分からないと初速が出ない

02 判断に時間がかかる

似た情報が複数の場所にあると、参照先の判断が止まる

03 人が迷う場所はAIも迷う

AIの性能以前に、探索コストが開発速度を支配する



次の壁は「そのデータがどれくらい正しいか分からない」ことでした

どの行も「不明」が並ぶと、AIに渡してよいか判断できない

「不明」が招く影響

入力前提の曖昧さ

出力の揺れにつながり、安心して使えない

属人化

「この人に聞かないと正しさが分からない」状態が残る

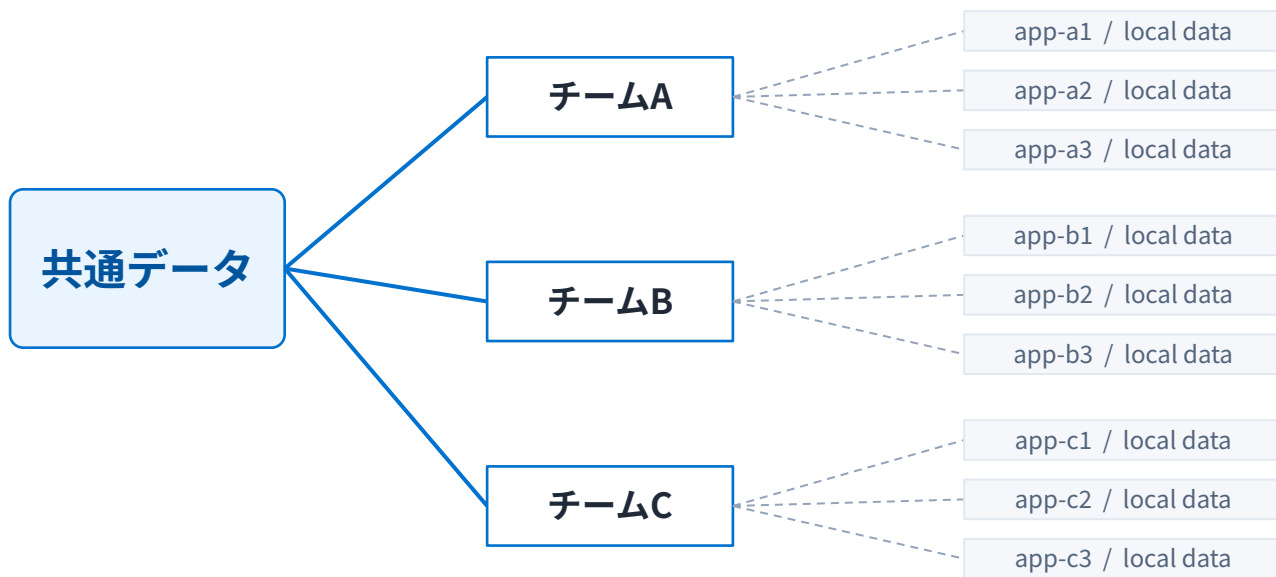
AIDDを進めるほど拡大

アプリ数に比例して、誤った前提が複製される

項目	鮮度	正確性	定義
顧客マスタ	不明	不明	不明
契約データ	不明	不明	不明
FAQナレッジ	不明	不明	不明
議事録/提案書	不明	不明	不明
社内規定	不明	不明	不明

各チームが個別に AIDDを進めるほど、データ分散は悪化しやすい

AIDDの成功体験そのものが、データの分岐を加速させる側面がある



本質的には全て同じデータ

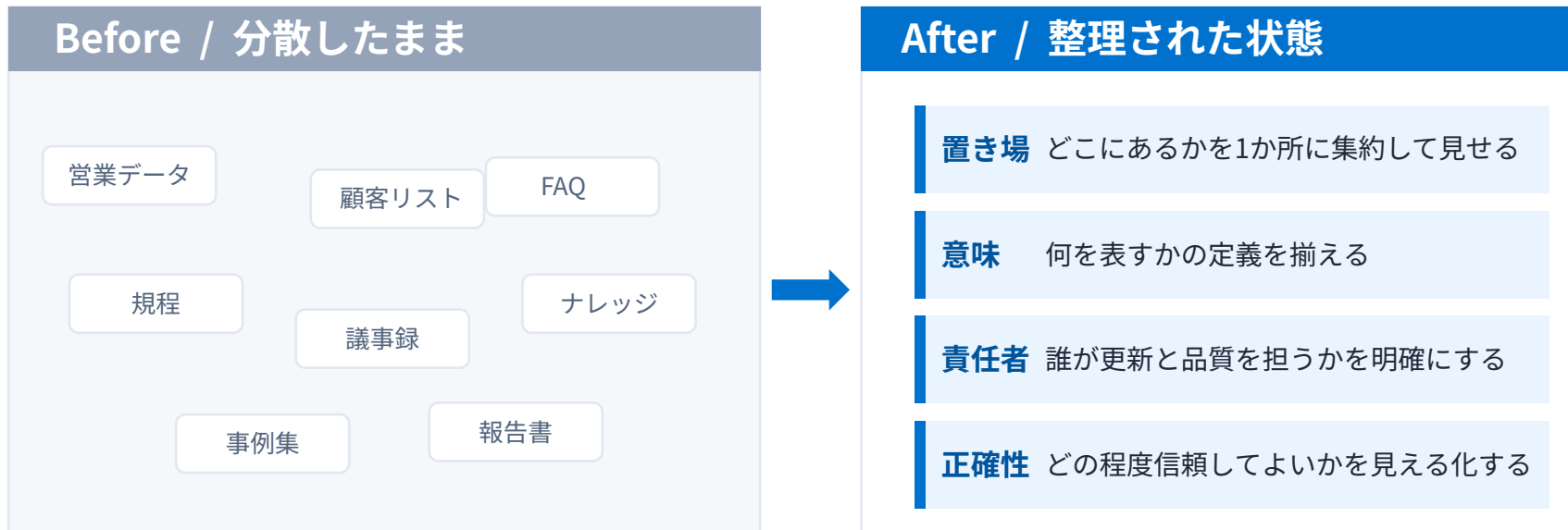
例：app-a1独自の給与データ

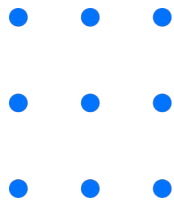
例：app-b1独自の給与データ

例：app-c1独自の給与データ

そのためデータ整理は、AI活用の後工程ではなく前提条件です

AIDDを継続可能にするには、データ整理を開発の前提として扱う





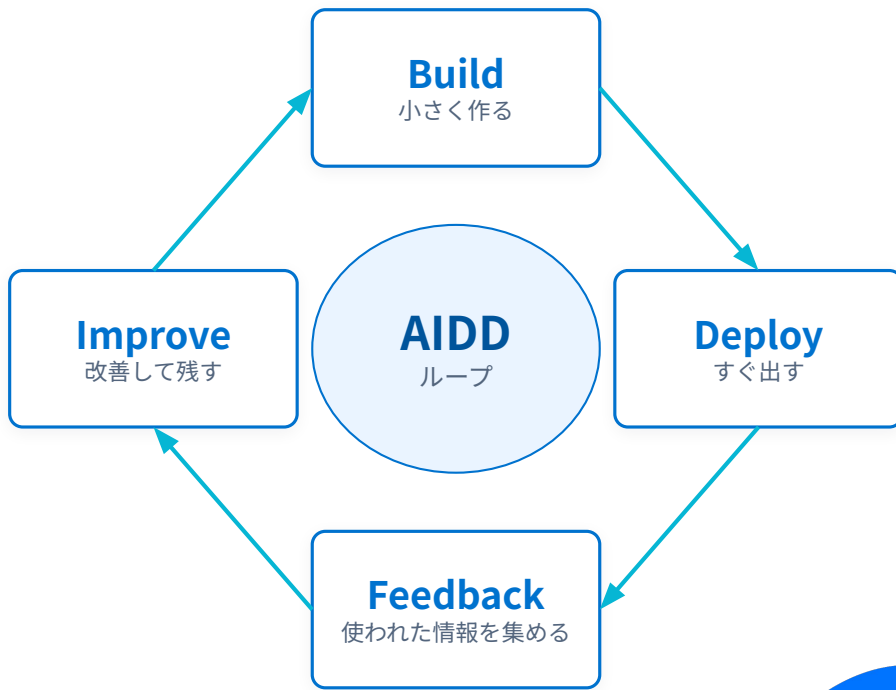
小さく速く出す

一方でAIDDアプリは、小さく速く展開できるほど価値が出やすい

完成度より先に「出して試せること」が価値になる

実行基盤が重いと

- ・デプロイで止まる
- ・出すこと自体に身構える
- ・改善サイクルが鈍る
- ・AIDDの良さが消える



回転数を上げると

- ・試行回数が増える
- ・現場の声が早く届く
- ・良いものだけ残せる
- ・撤退判断もしやすい

Cloud Run は、AIDDで作ったアプリを組織内に展開しやすくします

コンテナ化したアプリを、軽い手順で社内に届ける選択肢

載せやすい

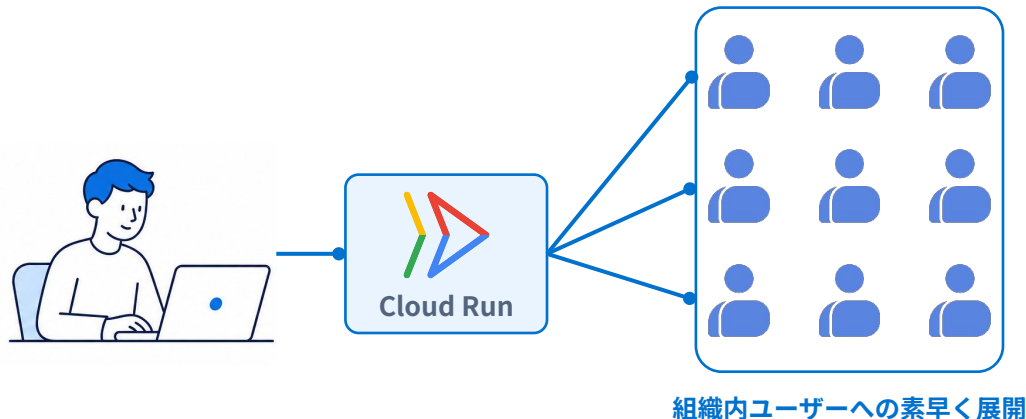
コンテナ化したアプリをそのまま動かせる

出しやすい

公開までの手順をシンプルに保ちやすい

回しやすい

短いサイクルで試行錯誤の回転数を上げられる

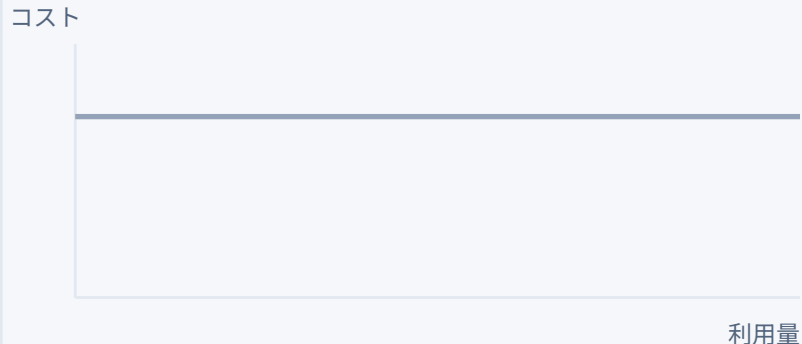


社内でAIDDアプリが増えた背景には、Cloud Run の手軽さが一助になっていた

リクエストベース課金は、社内向けAIDDアプリの展開モデルに合う

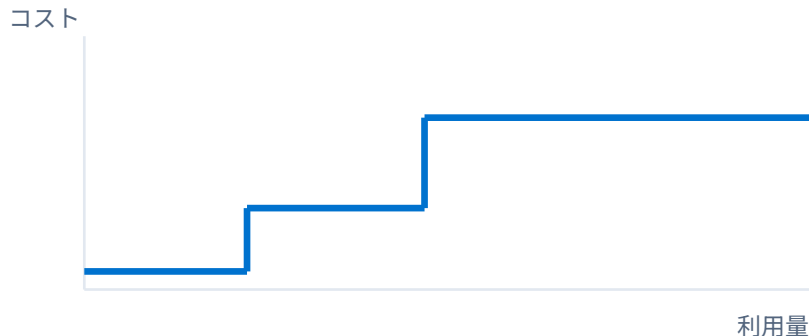
常時稼働を前提にしない社内利用に効く課金モデル

常時固定費モデル

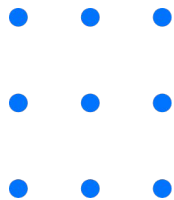


- 使われなくても費用が発生する
- 「試しに出す」のハードルが上がる
- 止める判断にもコストがかかる

利用量連動モデル



- 使われた分だけ費用が発生
- まず試す心理的ハードルを下げやすい
- 残す / 広げるの判断を後で行いやすい



安全に広げる

次の論点は「安全に広げ続けられるか」になります

Cloud Runで出せるようになると、運用設計の難所が現れる

運用ガードレールの内側で動かす

Cloud Run
app-1

Cloud Run
app-2

Cloud Run
app-3

Cloud Run
app-4

Cloud Run
app-5

認証・認可

誰が叩けるか / どう絞るか

Secret 管理

鍵や認証情報の取り扱いを統一

監査ログ

誰が何をしたかを後から追える

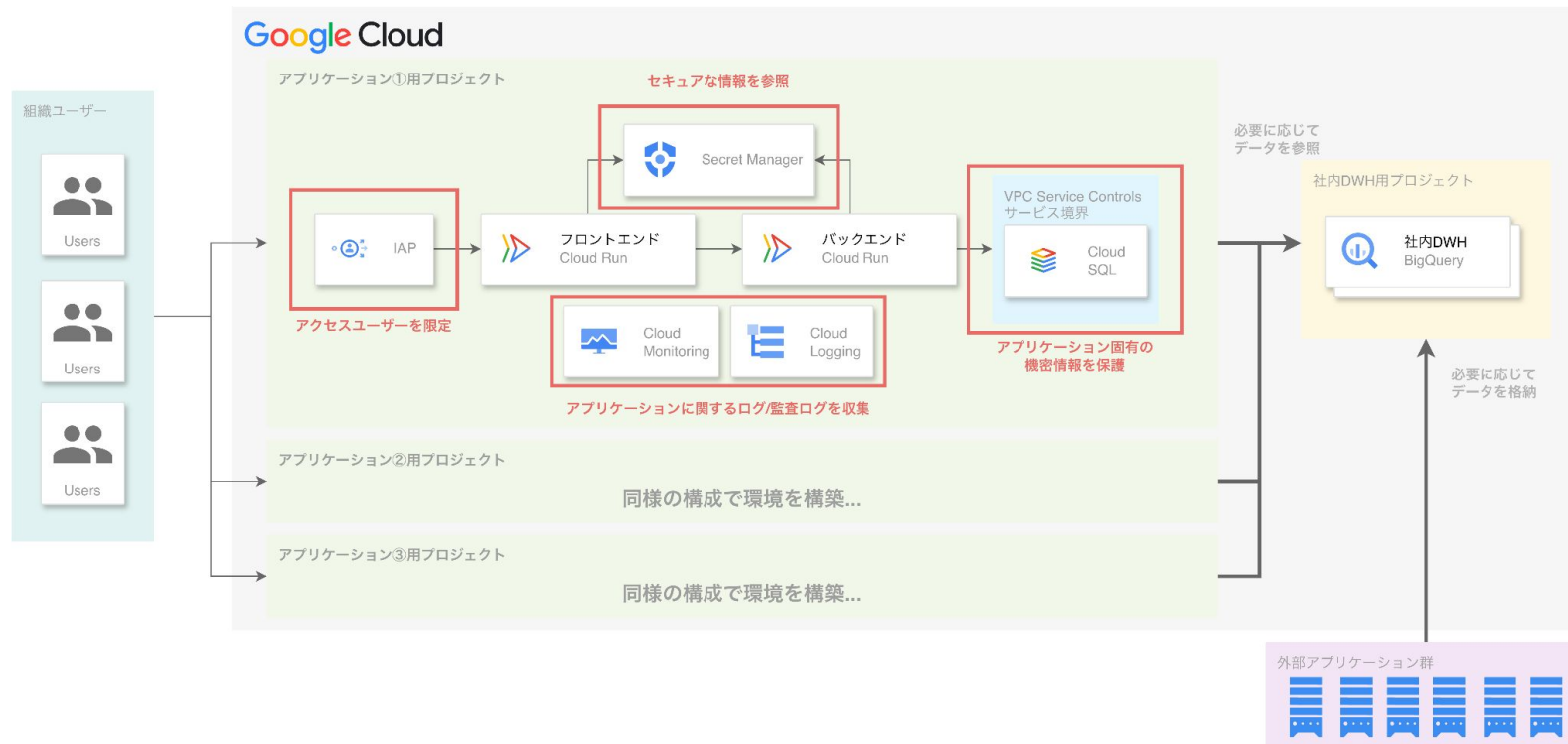
データアクセス境界

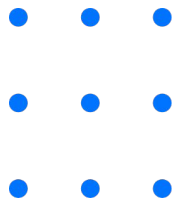
AIアプリから触ってよい範囲

ここを各アプリ個別対応にすると、運用負荷もリスクも増えやすい

最後に“型”を作成して展開するのも好ましいです

実行基盤と運用ルールはセットで整える





まとめ

運用者の役割は、3つの要素を両立させることです

信頼できるデータ × すぐ試せる基盤 × 安全に広げる仕組み

