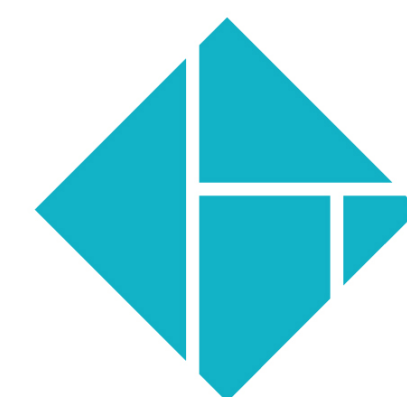


俺たちの
ドメイン駆動設計は
これからだ！



TIS
TIS INTEC Group

本谷

亮介

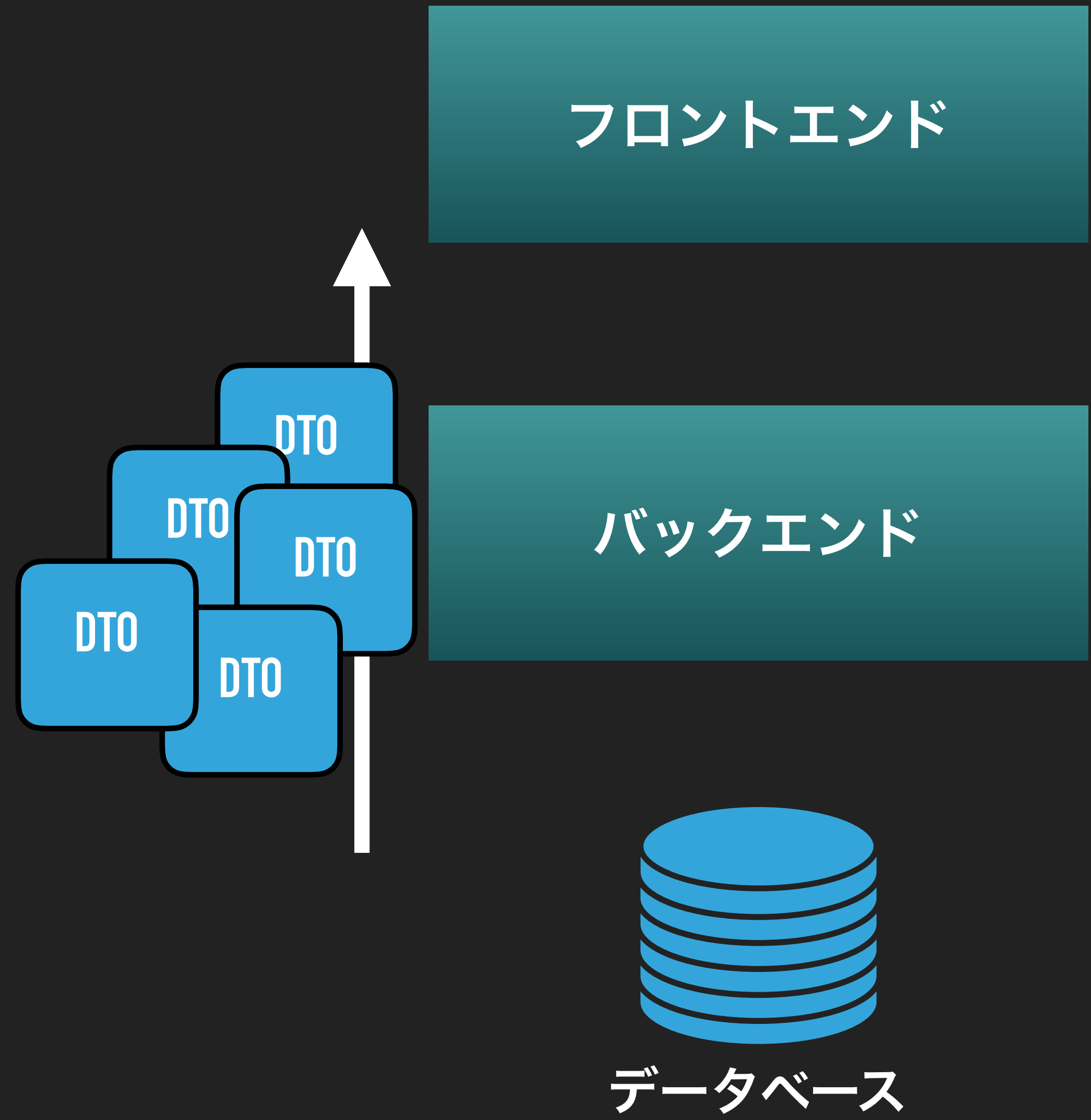
複雑性との戦い

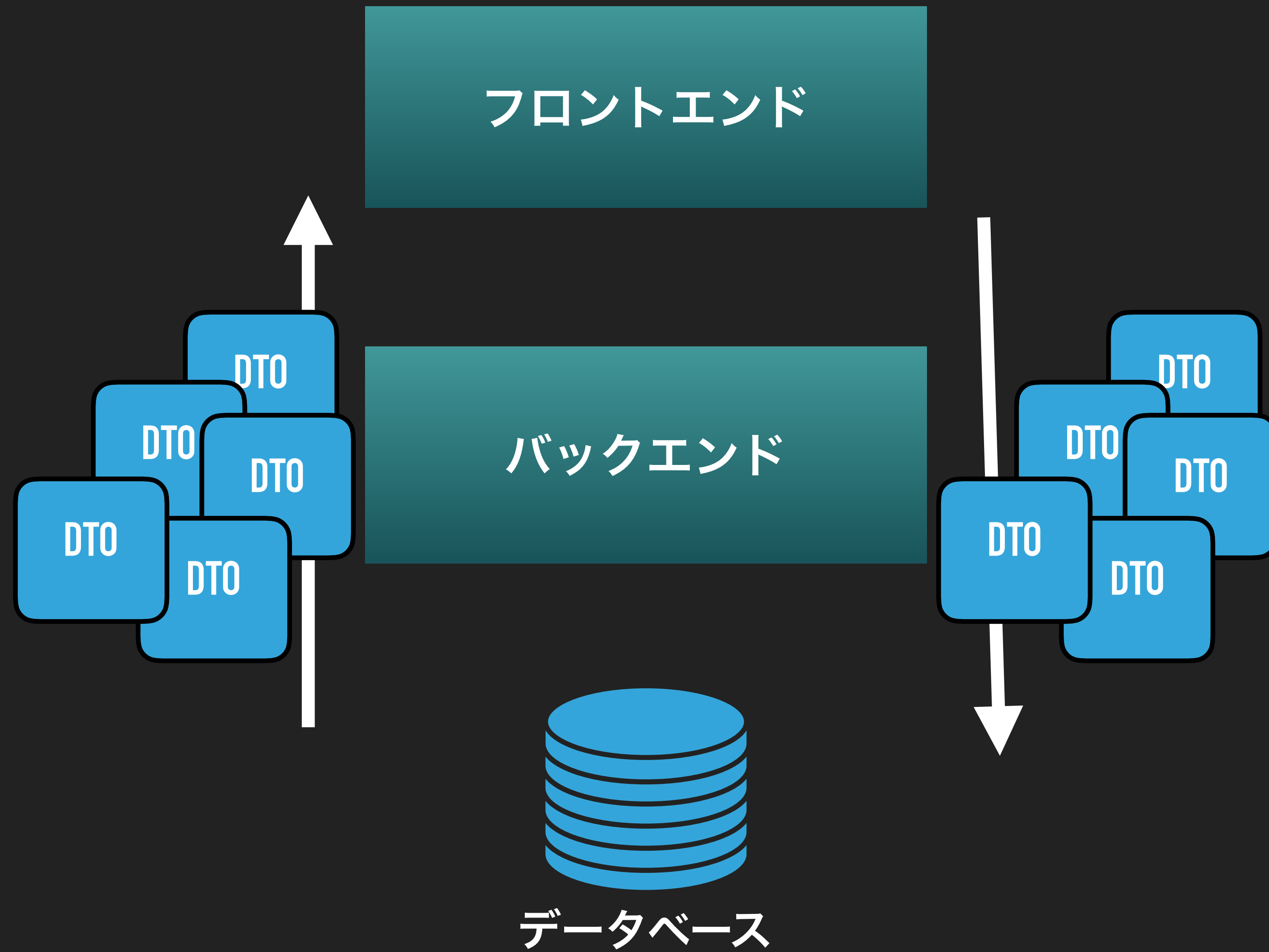
フロントエンド

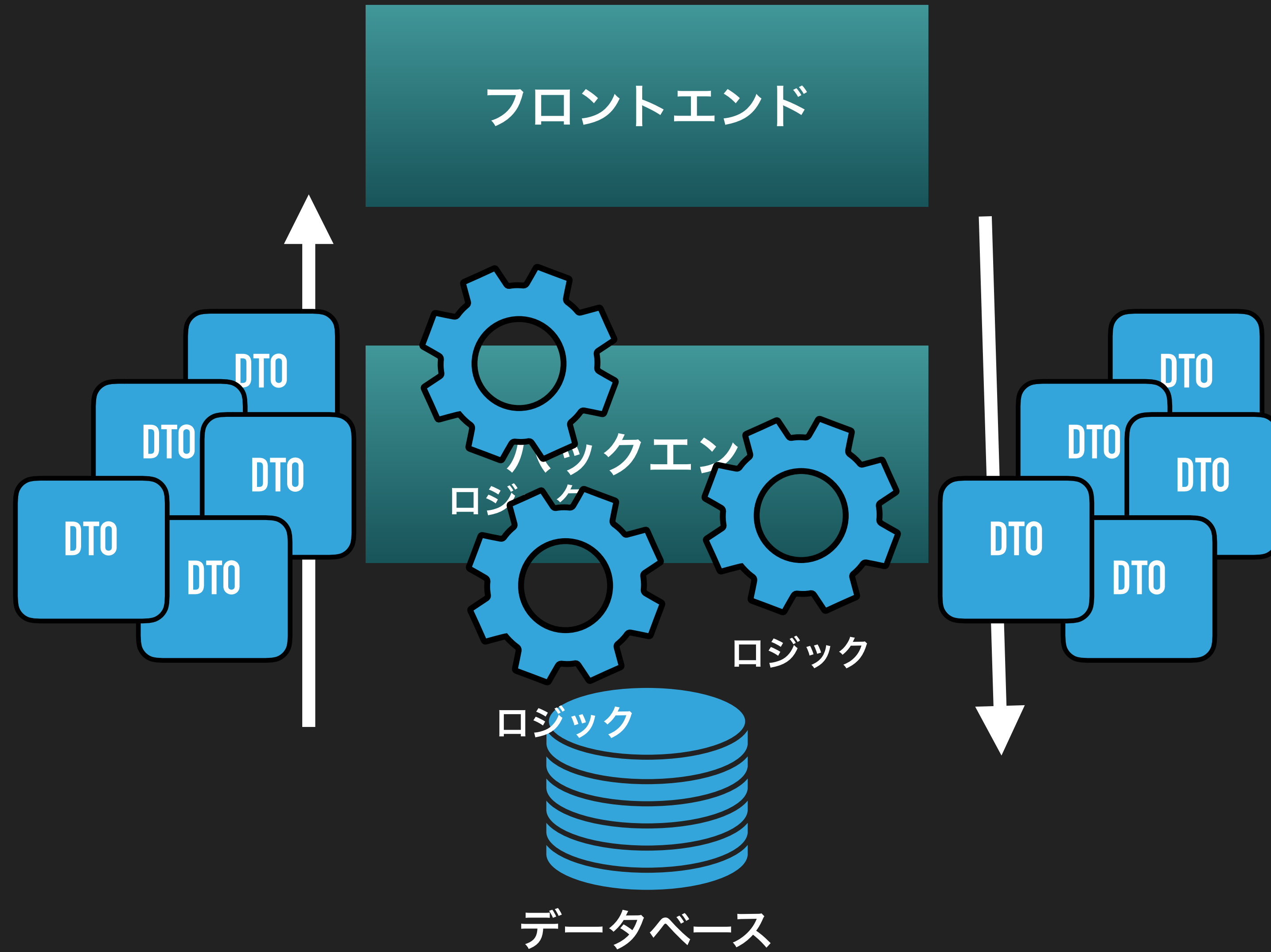
バックエンド



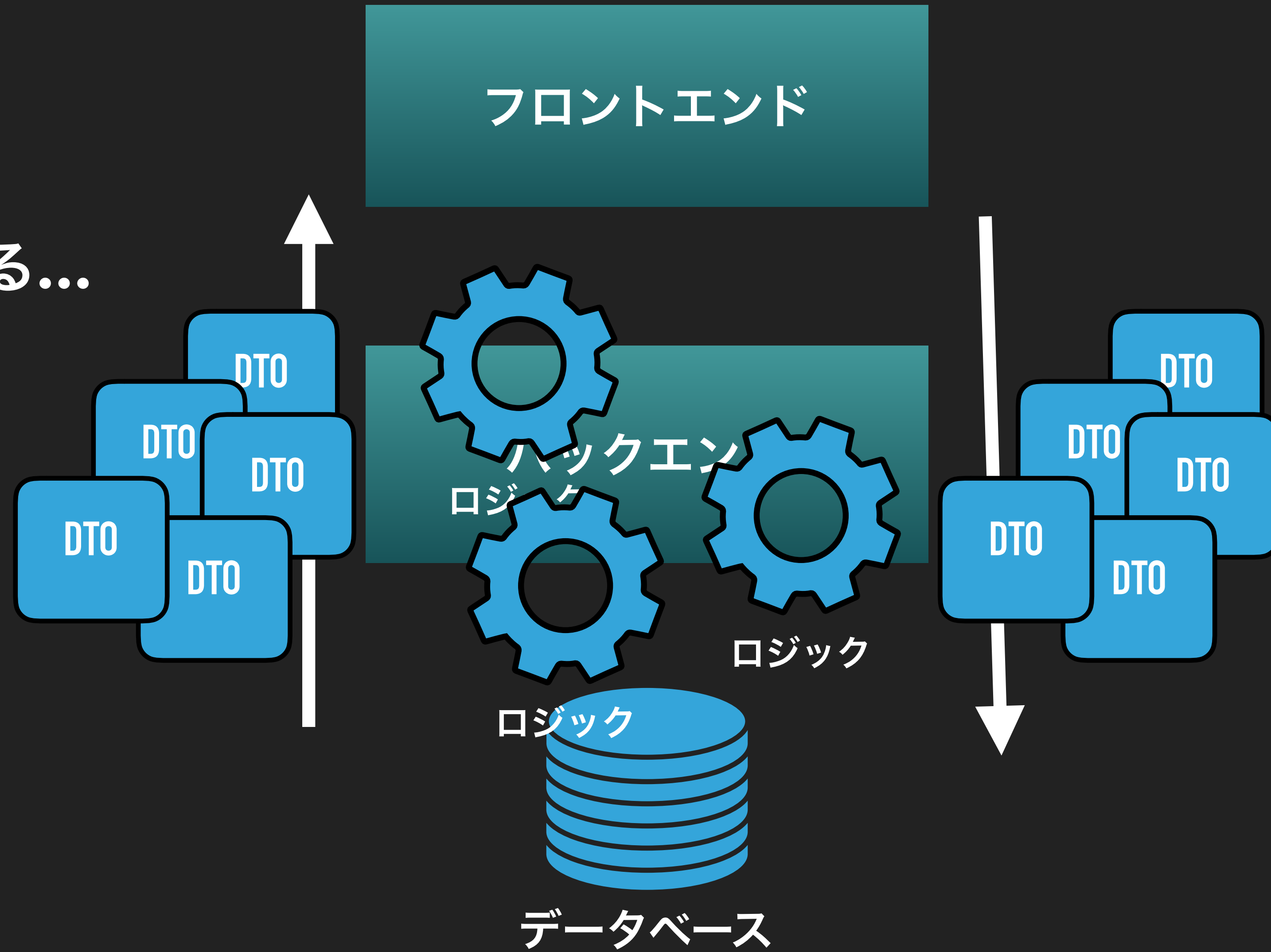
データベース



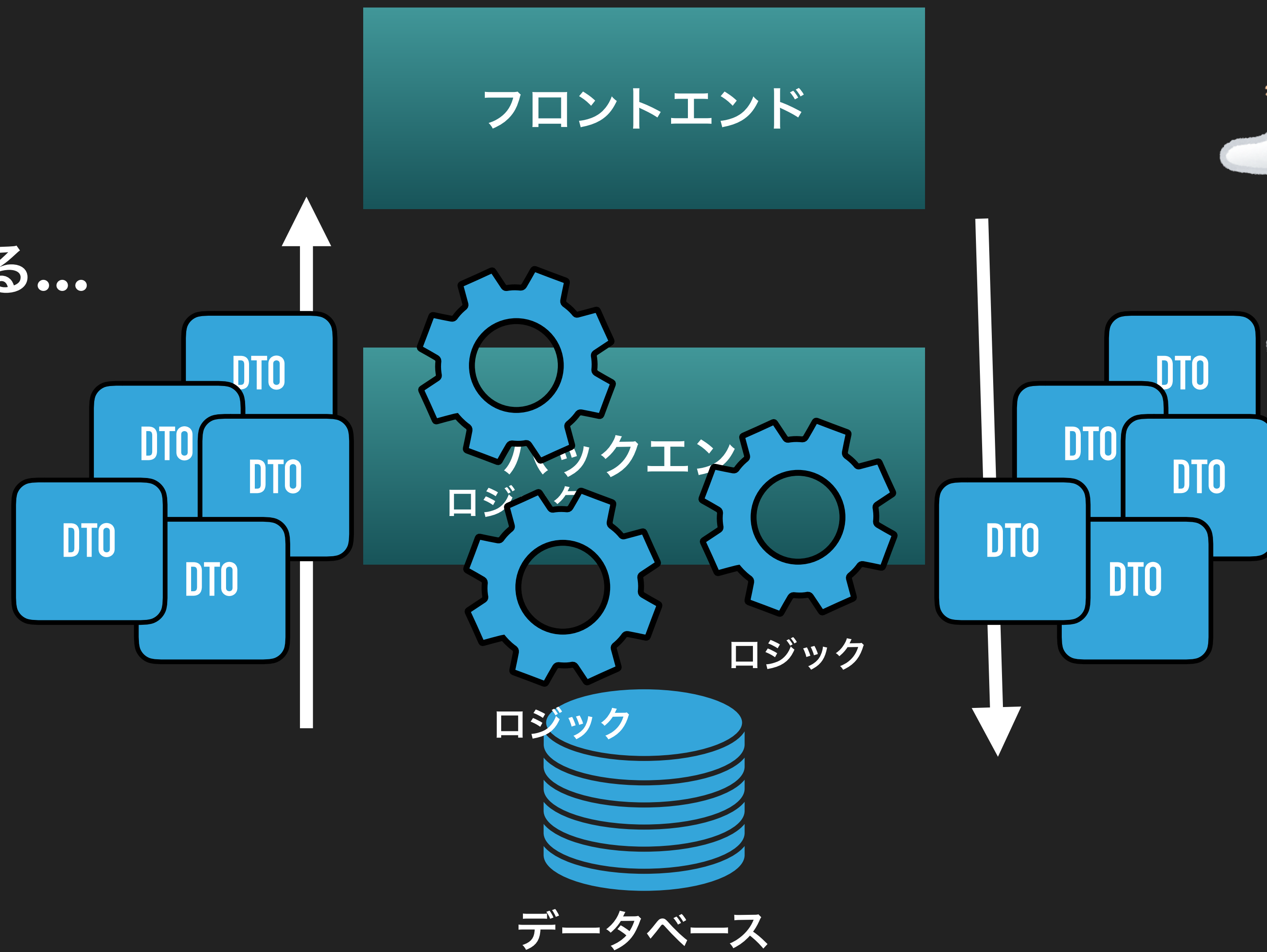




複雑になってくる...

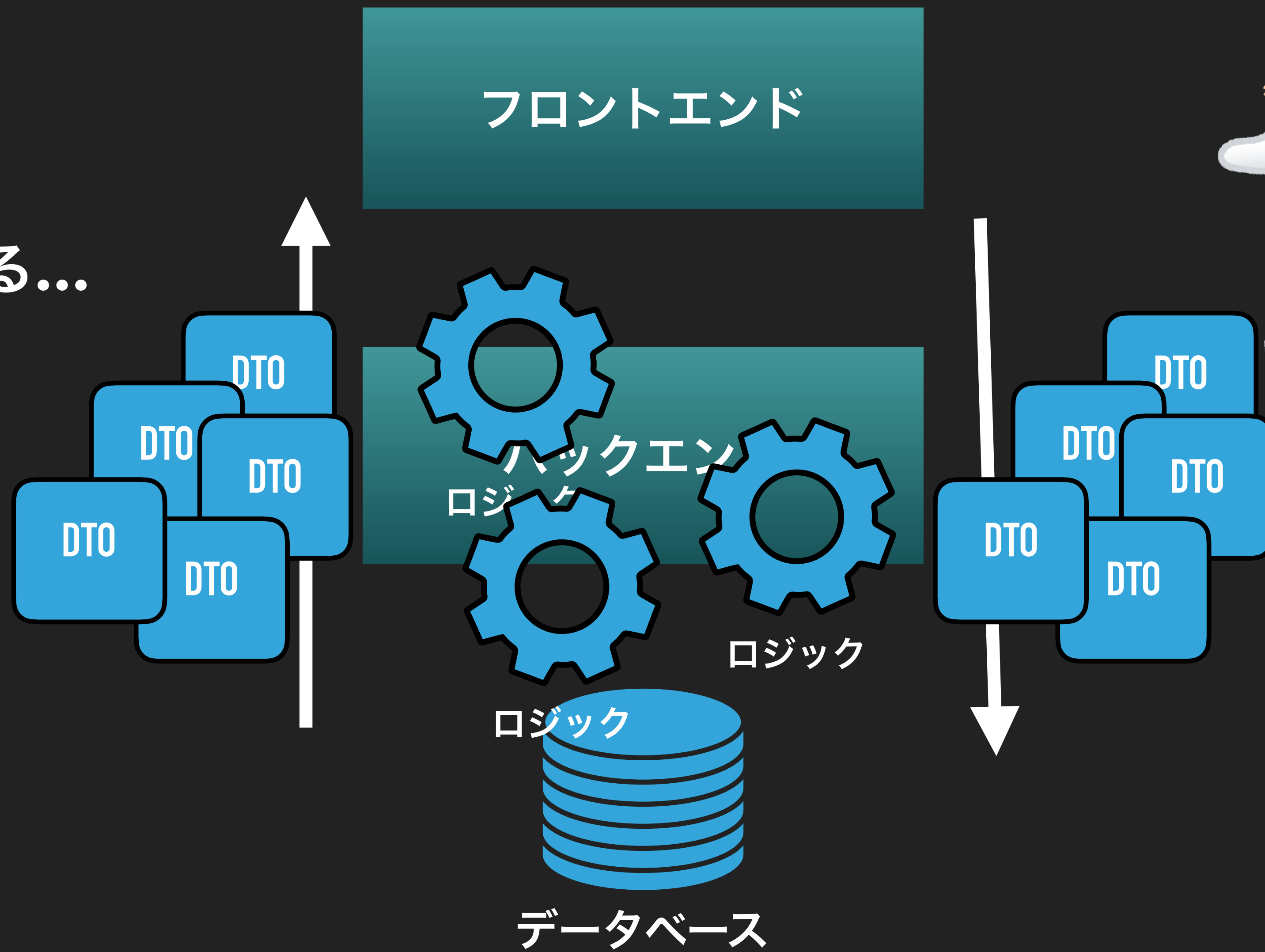


複雑になってくる...



仕様変更

複雑になってくる...



仕様変更

拡張性
変更容易性

...

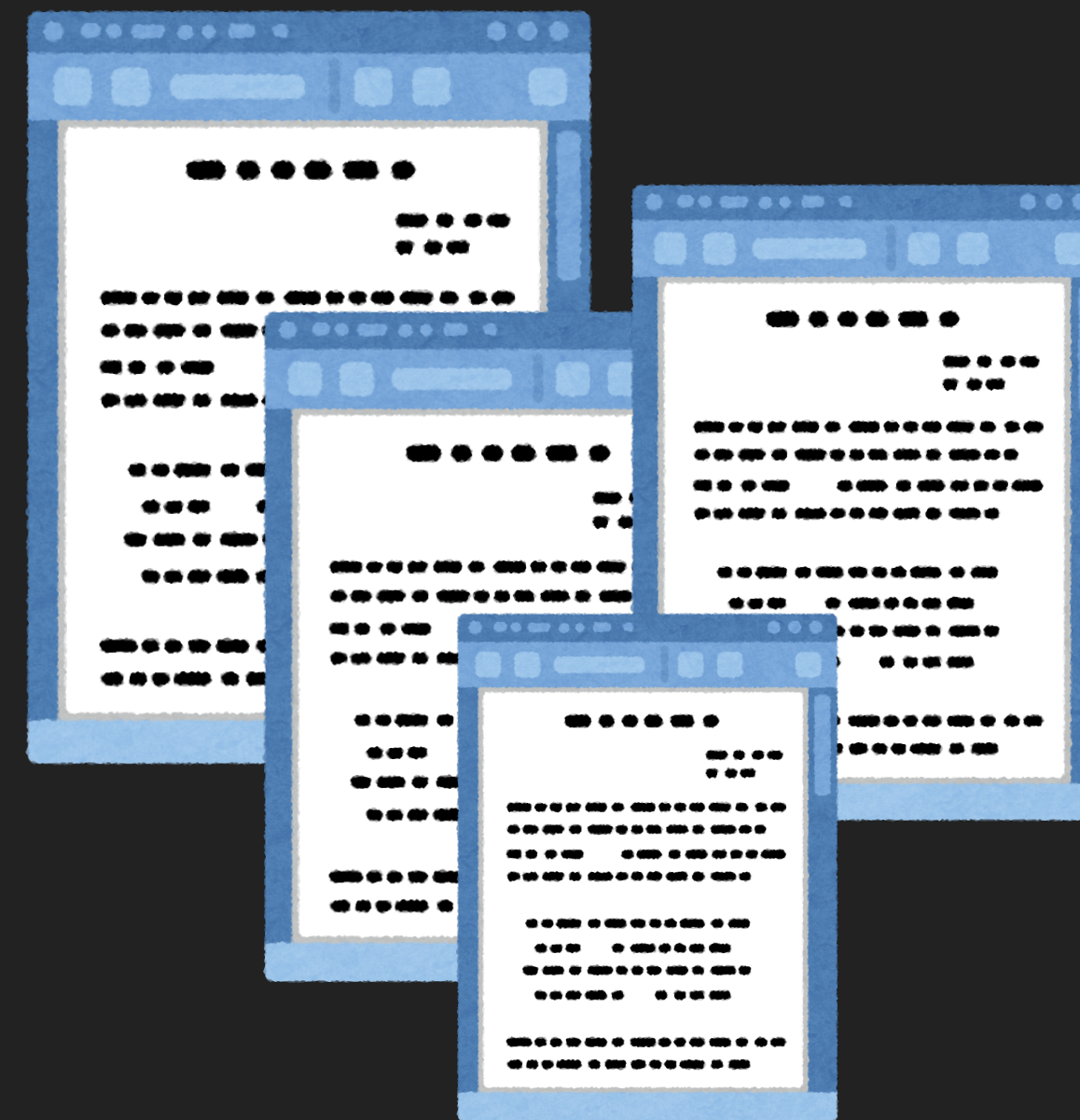
維持するのが難しい！

どうしてこうなる？🤔

コードを見ただけでは
意図が見えてこない！



業務知識
(なぜ)

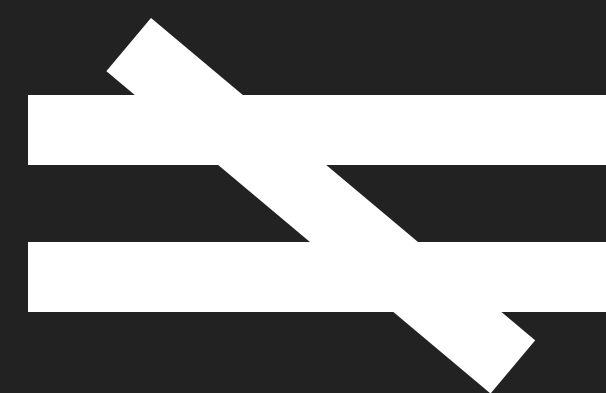


仕様書
(どのように)



対象業務（ドメイン）

の言葉



コード

- ResultDto
- Data
- doExecute

分かりずらいコードの完成

技術が複雑だから？

対象業務（ドメイン）
が複雑だから！

ドメイン駆動設計

多くのアプリケーションにおいて、最も重要な複雑さは、技術的なものではない。

Eric Evans

複雑なものはドメインそのものの、すなわち、ユーザの活動やビジネスなのである。

Eric Evans

ビジネスが持つ複雑性と 戦う道具

やったこと

ドメイン駆動設計（本）から
できそうなところをピックアップ

徐々にプロジェクトに適用

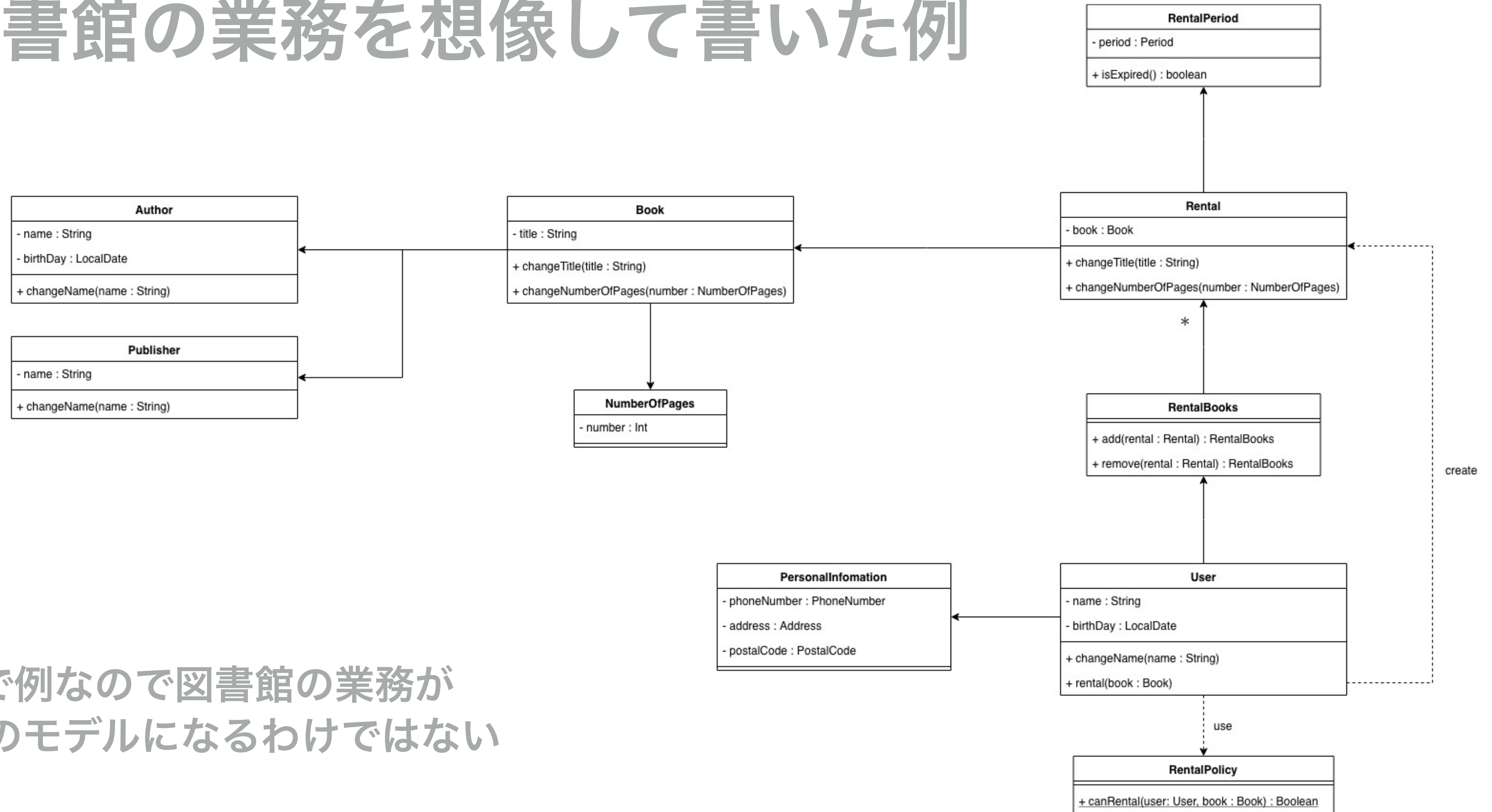
1. ドメインモデルの構築
2. 設計パターンの適用
3. ドメイン層の作成

① ドメインモデルを構築

ドメインモデルとは？

我々が対象ビジネスを
どのように捉えて、
どのようにシステム化するか
を表現するモデル

図書館の業務を想像して書いた例



※あくまで例なので図書館の業務がすべてこのモデルになるわけではない

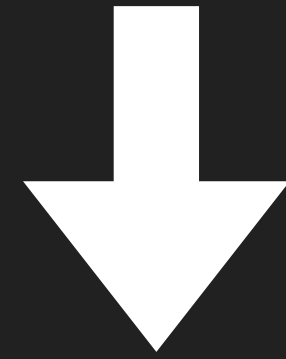
モデルを使う事で何が嬉しいのか？

ドメインモデルを元に実装する事で

コードが業務の

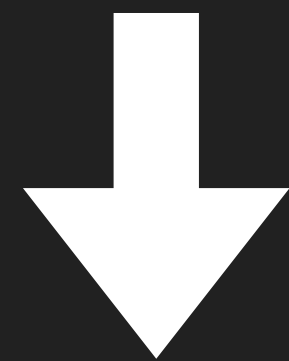
知識・概念と対応づく

ドメイン
(複雑なユーザーの活動・業務知識)

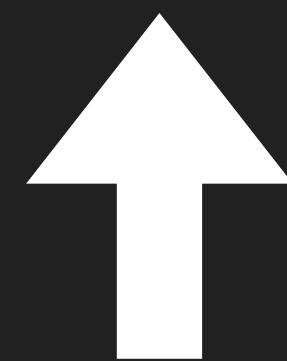


知識を選び抜き、抽象化

ドメインモデル



ドメインモデルを元に設計



変更のフィードバック

ソースコード

ドメイン
(複雑なユーザーの活動・業務知識)



対応する

ソースコード

注意！！

ドメインモデルとは特定の図ではなく、
図が伝えようとしている考え方である

Eric Evans

よくある誤解

- ドメインモデルはUMLクラス図
- 開発の最初に一気に作って終わり

超重要

1. お客様さんの業務の勉強

- システム化対象領域だけではなく、その周辺の知識も含める
- システムに置き換えたらどうなるか？は考えないようにする

2. システム化対象領域の概念の整理

- 名詞、動詞を中心に
- 「もの」だけでなく、プロセスにも着目する

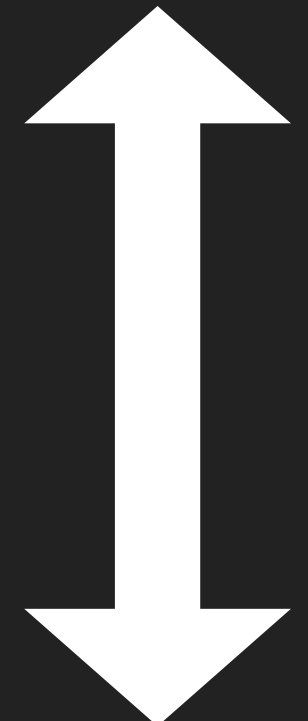
3. 重要な概念からモデルを作成

- いきなり完璧なモデルは出てこないと割り切ることが大切

②設計パターンの適用

設計パターンとは...

ドメインモデル



ソースコード

モデルの意図を
コードに反映させる

1. エンティティ
2. 値オブジェクト
3. サービス

エンティティ

同一性を持ったモデル要素に適用

対象の属性だけでは
比較できない性質

利用者

名前や年齢を比較しても
同一人物か判定できない


```
public class User {  
    private final Long id;  
    private String firstName;  
    private String lastName;  
  
    public Person(Long id, String firstName, String lastName) {  
        ...  
    }  
  
    public boolean isIdentifiedBy(Person otherPerson) {  
        return this.id == otherPerson.id;  
    }  
}
```

一意になるID

IDを用いて
同一性を比較する
メソッド

値オブジェクト

同値性を持ったモデル要素に適用

属性だけで同一なのか
比較できる性質

ページ数

表現している値のみで
同一なのか判定しても問題ない

10ページ == 10ページ true

10ページ == 20ページ false

```
public class NumberOfPages {
```

一意になるIDは持たない

```
    private final int number;
```

```
    public NumberOfPages(int value) {
```

```
        ...
```

```
    }
```

```
    public boolean isSameValueAs(NumberOfPages otherValueObject) {
```

```
        return this.number == otherValueObject.number;
```

```
    }
```

```
}
```

フィールドの値を使用し
同値性を比較するメソッド

サービス

ドメイン内の手続きを表現する

特定の概念に紐づかず、
プロセスのみに関心がある

貸し出し

特定の概念に紐づくかず
プロセスのみを表す

```
public class RentalService {
```

```
    // 送金
```

```
    public static Rental rentalBook(  
        User rentalUser,  
        Book rentalBook) {
```

```
        // 貸し出せるか確認
```

```
        if (!RentalPolicy.canRental(rentalUser, rentalBook))  
            throw new IllegalArgumentException();
```

```
        // 貸し出す
```

```
        Rental rental = rentalUser.rental(rentalBook);
```

```
        return rental;
```

```
    }
```

```
}
```

クラスメソッド
として実装し特定の概念から分離

③ ドメイン層の作成

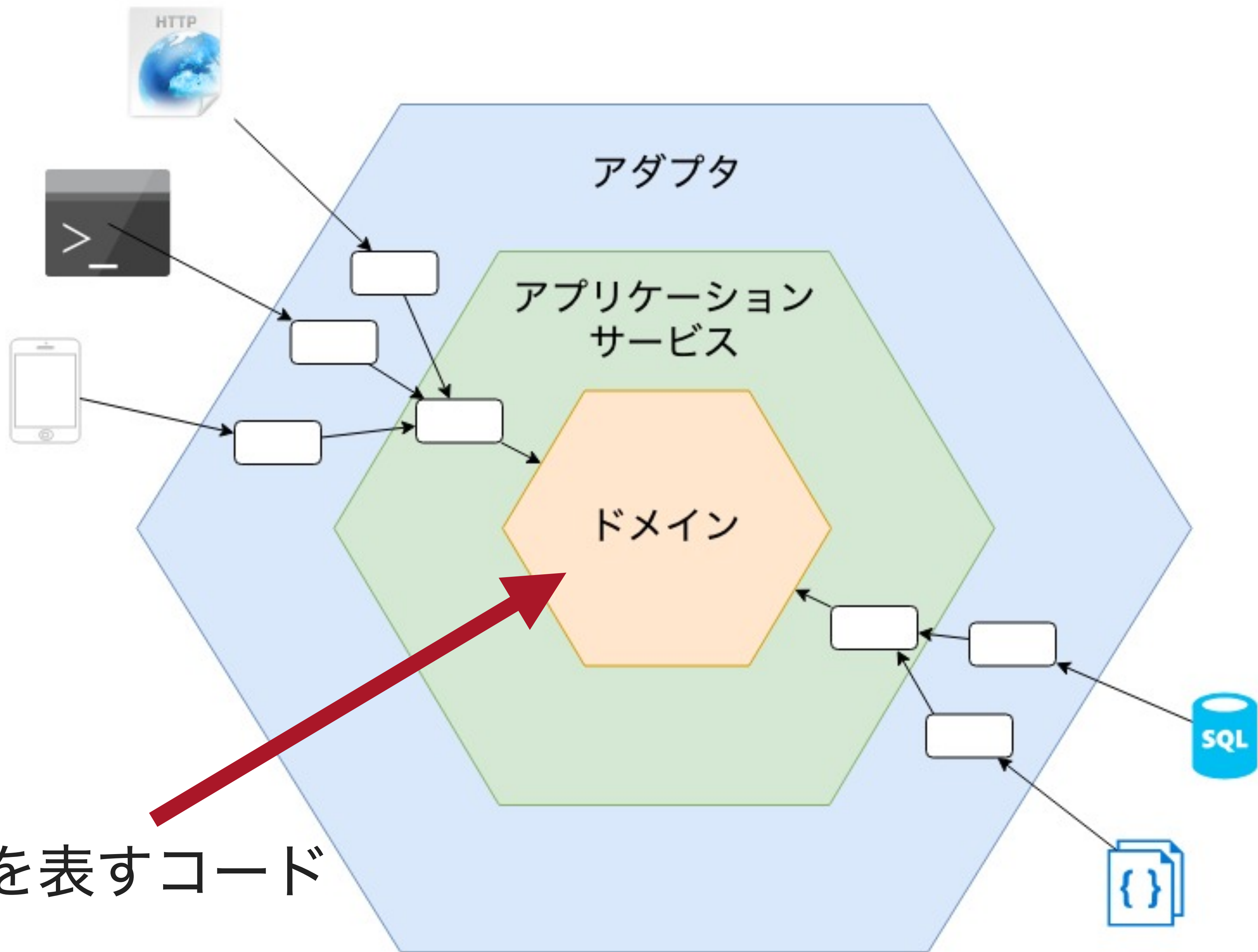
複雑なビジネスを表現するコードが
他の技術要素のコードと混ざる

わかりにくいコードの出来上がり

低凝集度、密結合...

ドメインモデルにより導き出される
コードを他の技術領域から隔離

へキサゴナルアーキテクチャ



業務を表すコード

Browser tabs: BigInteger (Java Platform SE 8), Google 翻訳, Fintan | アプリケーション開発ノウハウ・ツール

Navigation: ライセンス FAQ 関連用語; 🔍

Fintan

Fintanは、TIS株式会社がシステム開発プロジェクトで培ったノウハウを集約したサイトです。PJ推進のプラクティスや、要件定義/設計/プログラミング/テストといった作業のプラクティス、成果物のテンプレート/サンプル、各種開発ツールを提供します。

Fintanは、どなたでも無償でご利用いただけます。

🔍 SEARCH BY TAG タグから探す (AND検索)

- 要件定義 (3)
- テスト (6)
- Java (5)
- Nablarch (4)
- Spring (1)
- ウォーターフォール開発 (7)
- スクラム開発 (15)
- サービス開発 (4)

ソフトウェア設計

- アプリケーション方式設計書サンプル
- Nablarch アプリケーション開発標準
- Nablarch 設計書フォーマット&サンプル
- Nablarch ETLデザイナー
- SPA+REST APIのシステム構成例 (AWS)
- SPA+REST APIにおけるセッションを使った認証の実践例
- モニタリング、監視、ログ収集の実践例 (AWS)
- Reactを使ったフロントエンドのアーキテクチャ事例
- ヘキサゴナルアーキテクチャ導入事例

fintan.jp

結果どうだった？

① 理解しやすいコード

対象業務（ドメイン）

の言葉



コード

意図を表現するコード

- 開発速度向上
- 不具合発生を抑制



② コミュニケーションコストの低減

メンバ全員が
同じドメインモデルを元に
開発する

メンバ全員が
捉えている業務モデルが
同じ

言葉・考え方が
理解できる

マージリクエスト

- コードに何を加えたのか？
- コードをどう変更したのか？
- なぜそれをやったのか？

まるで自分が作業したのかのよう

分かる



③ 変更の影響範囲がすぐに分かる

今まで

要求の変化



コードにどんな影響が出るだろうか？

ドメイン駆動設計導入後

要求の変化



我々が捉えている**業務モデル**は
どう変化するのだろうか？

MENU



TIS キャリア採用サイト

エントリー >
募集職種一覧を見る

より高いスキルか。
より広い視野か。
あなたの求める可能性が
ここにある。

キャリア採用の お知らせ



ITパートナーから ビジネスパートナーへ

Leading & Professional

自ら考え自ら発信していく。リーダーシップを持って、果敢にチャレンジしてほしい。
みなさんにつたえたい、TISの想い。

人材採用メッセージ >



Fintanは、TIS株式会社がシステム開発プロジェクトで培ったノウハウを集約したサイトです。PJ推進のプラクティスや、要件定義/設計/プログラミング/テストといった作業のプラクティス、成果物のテンプレート/サンプル、各種開発ツールを提供します。

Fintanは、どなたでも無償でご利用いただけます。

タグから探す (AND検索)

- 要件定義 (3)
- テスト (6)
- Java (5)
- Nablarch (4)
- Spring (1)
- ウォーターフォール開発
- スクラム開発 (15)
- サービス開発 (4)

fintan.jp

ありがとう
ございました！