

# 業務システムとマイクロサービス

発表者：土岐 孝平

# 自己紹介

- 土岐孝平(とき こうへい)
- Springをよく使っている
  - 書籍「Spring3入門」の共著
- 業務システムの開発に携わっている
  - マイクロサービスの考えを取り入れたい

# マイクロサービスは良いもの？

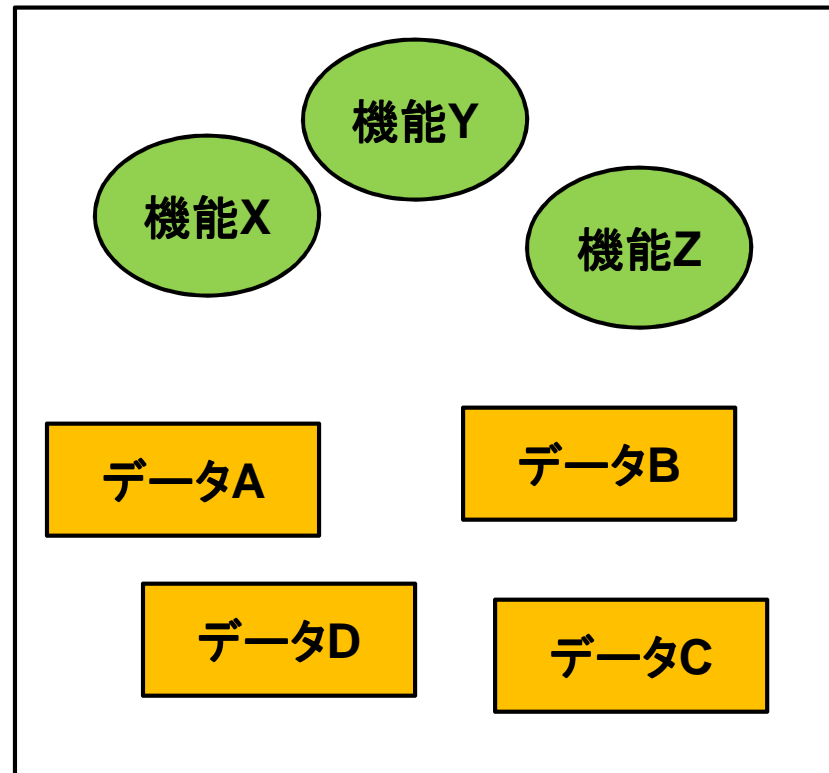
- 良いと思います
- システム(サービス)を小分けにする
  - 独立した仕組み
    - プログラミング言語、フレームワーク、データベース
  - 個別に開発、デプロイ
  - メリット
    - 改修しやすい、新しい技術に対応しやすい、システムを入替やすい
- サイロ化しないか？
  - 全体最適を考慮した上で小分けにしましょう

# データベースも分ける？

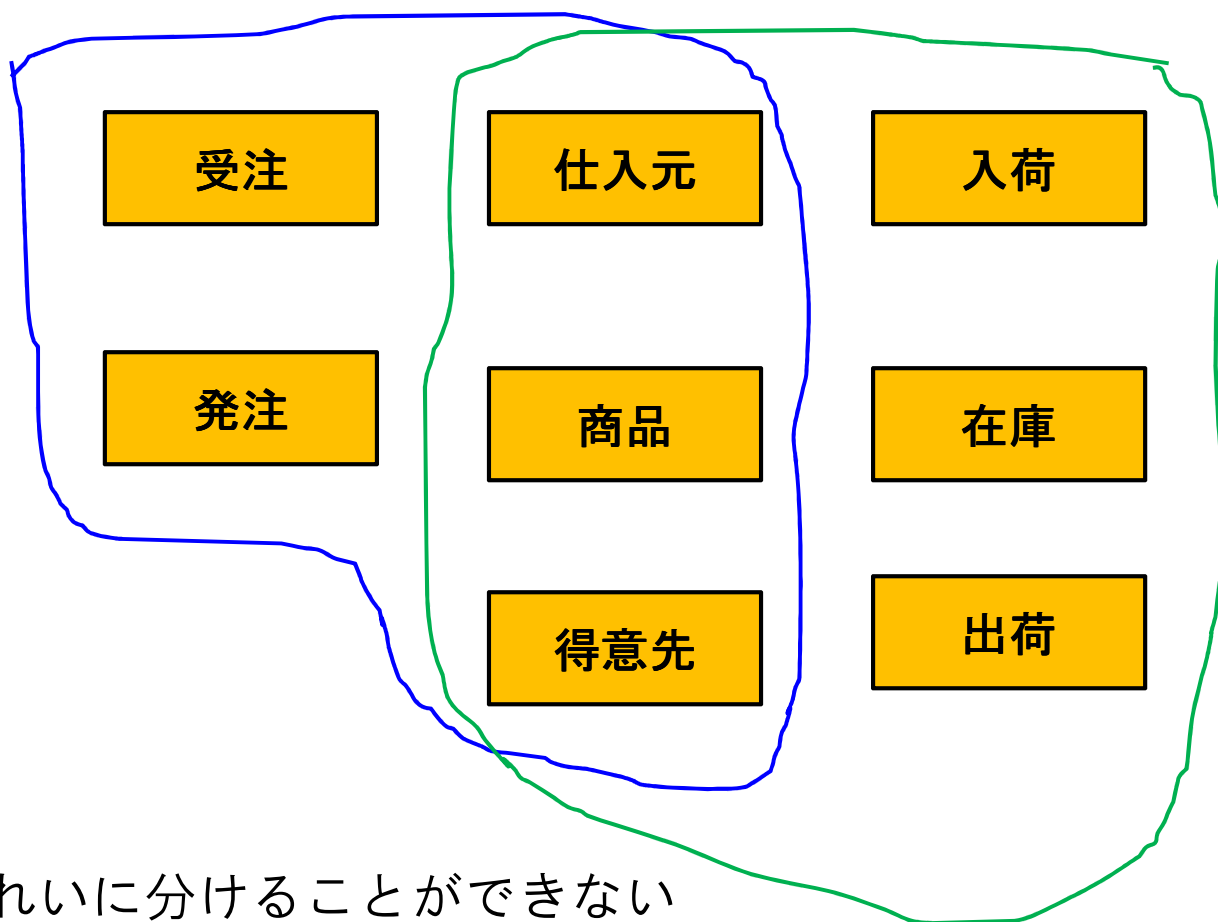
- Database per serviceパターン
  - By Chris Richardson
- メリット
  - システムに適した製品を選択できる
    - RDB、NoSQL、検索エンジン・・・
  - スキーマの変更が他のシステムに影響しない
- デメリット
  - システムを跨るトランザクション
  - システムを跨るデータの結合

# システム(サービス)の単位をどうするか？

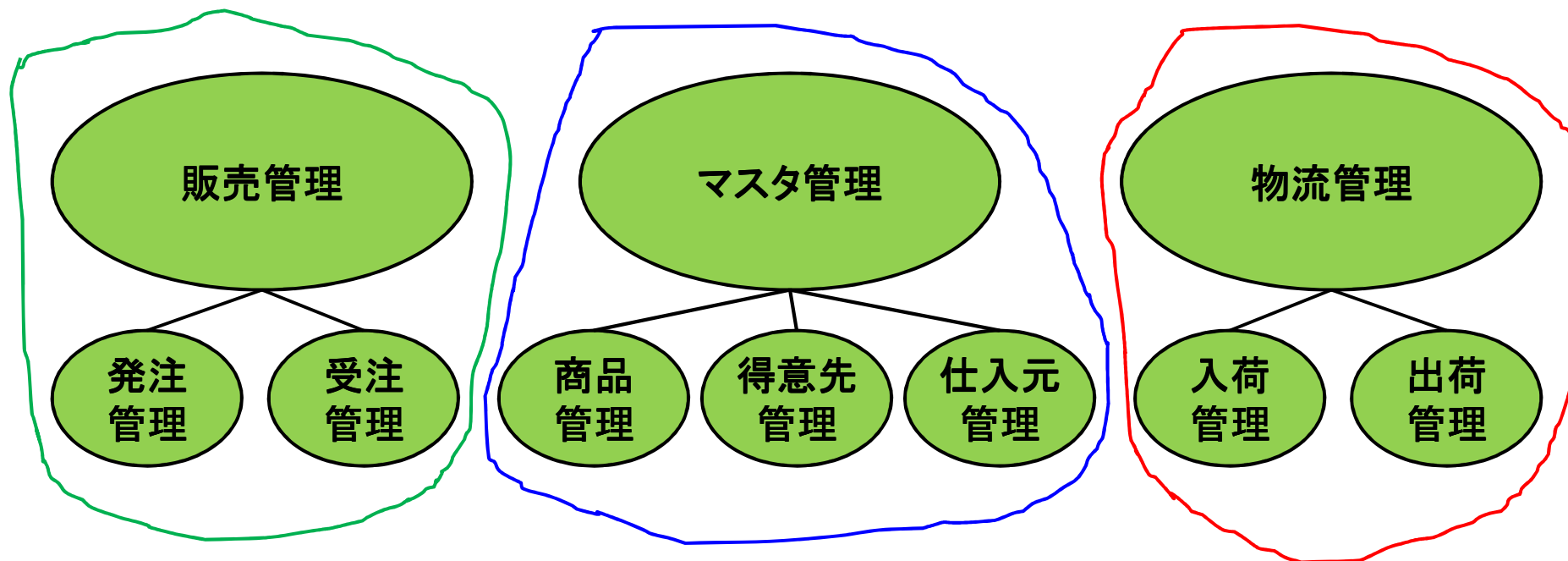
- 何を切り口にして分けるか？
  - データ？
  - 機能？



# データを切り口にした場合



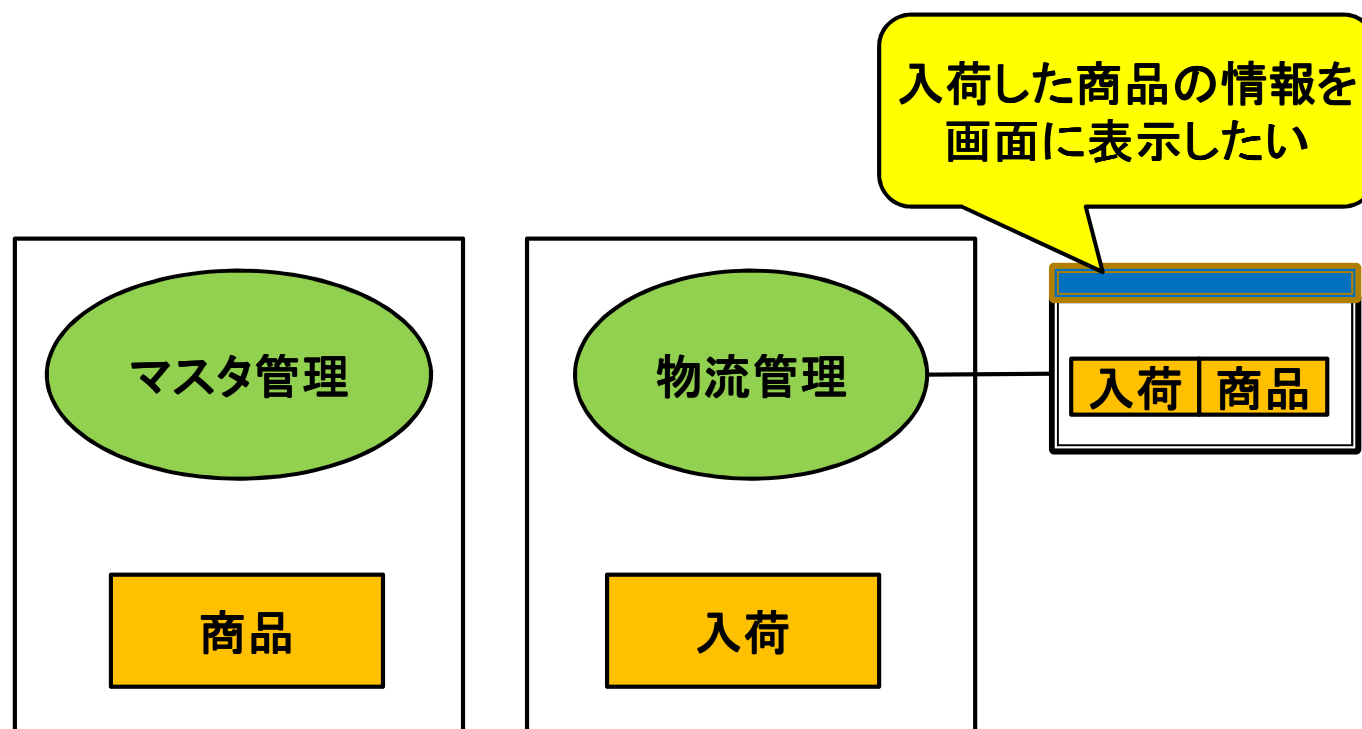
# 機能を切り口にした場合



- ・きれいに分けることができる
- ※ ただし、他のシステムのデータを参照する必要がある

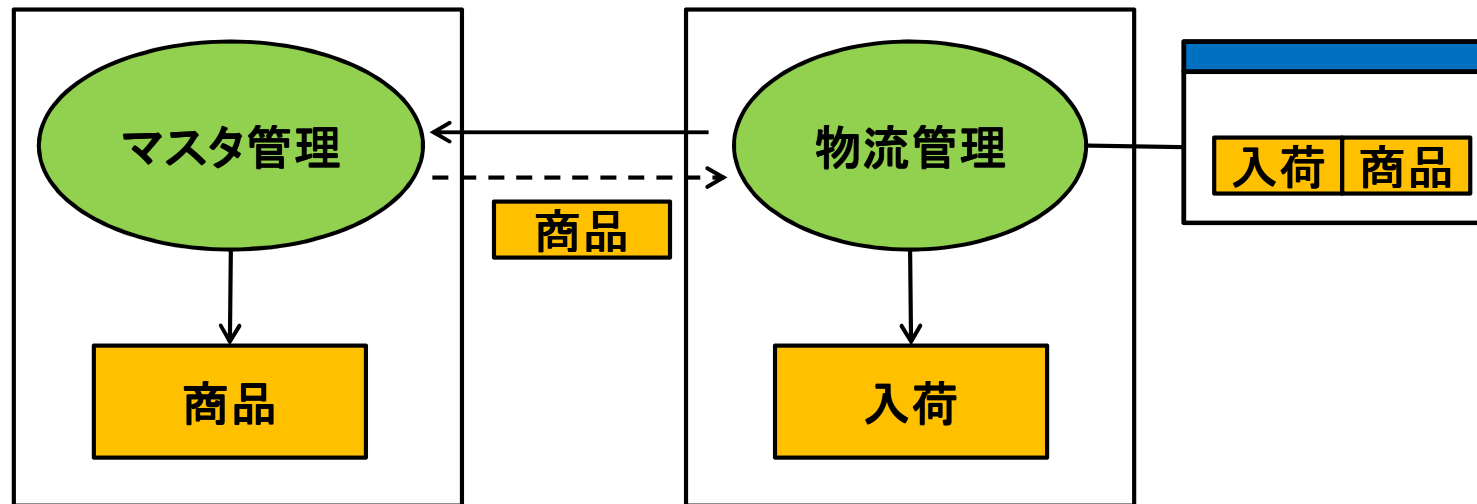
# 他のシステムのデータを参照

- 方法
  - リモートプロシージャコール(RESTなど)&アプリ側で結合
  - マテリアライズドビュー(実体を持ったビュー)として同期





# リモートプロシージャコール&アプリ側で結合



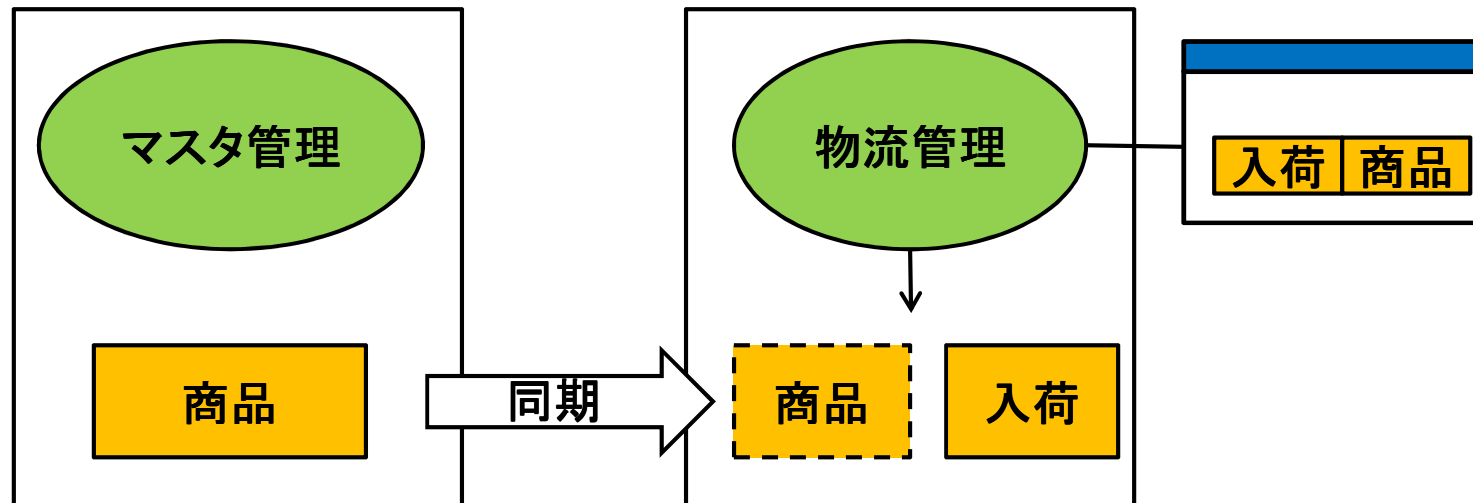
## 【メリット】

- ・参照する側でデータを持たなくてよい
- ・データの重複がない

## 【デメリット】

- ・パフォーマンス
- ・APIの設計が大変そう
- ・依存が強くなる
  - マスタ管理が停止すると物流管理が動かない
  - 開発時にマスタ管理のスタブが必要

# マテリアライズドビューとして同期



## 【メリット】

- ・ 柔軟に検索できる
- ・ 依存が弱くなる
  - マスタ管理が停止しても物流管理は動く
  - 開発時は商品のスキーマを用意するだけ

## 【デメリット】

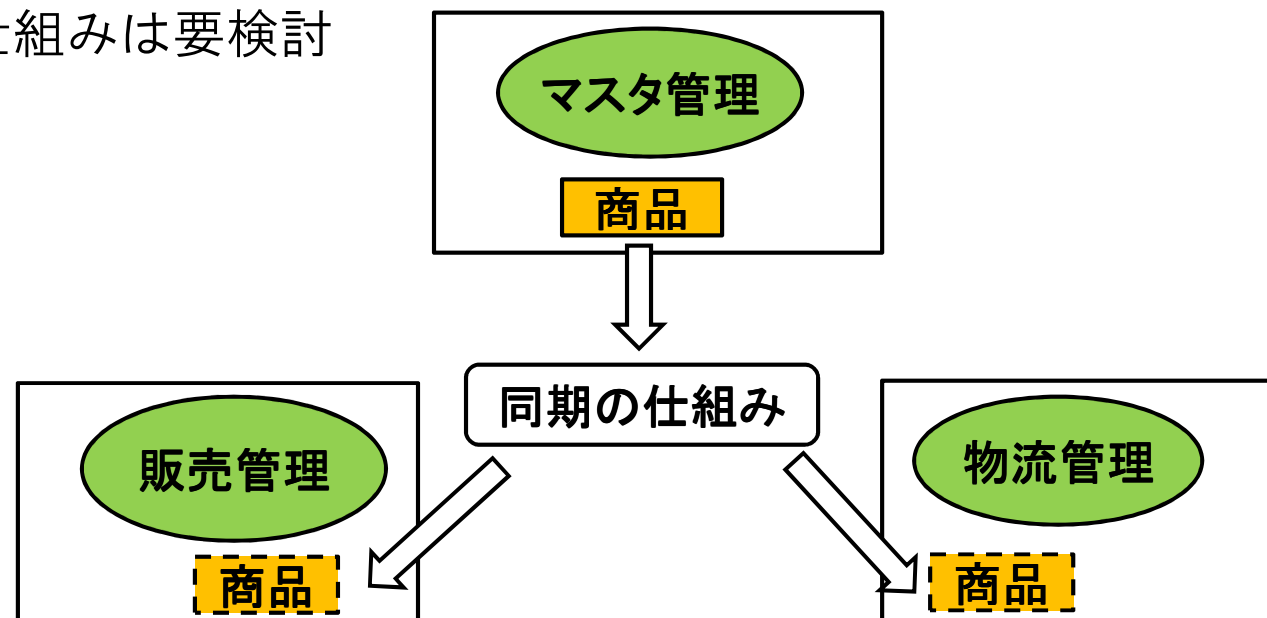
- ・ 仕組みが大変そう
  - ファイル？メッセージング？DBのマテリアビュー？
  - リアルタイム性、データの齟齬

# どっちの方法がよいか？

- ケースバイケース？
  - 指針を明確にするのが難しい
  - 2つの方法が無秩序に乱立しそう
- マテリアライズドビューとして同期した方が潰しが効く
  - マテリアライズドビューの方が柔軟な検索ができる
    - 検索エンジンを利用してもよい
- マテリアライズドビューで一本化したほうがよい気がする
  - ただし、同期の仕組みの作り込みは必要

# 私の今の方向性

- 機能を切り口にしてシステムを分ける
  - 開発が大変そうにならない程度の細かさ(細かくし過ぎない)
  - パッケージやSaaSで切り売りしてるくらいの粒度
- マテリアライズドビューとして参照データを同期
  - 同期の仕組みは要検討



ご清聴ありがとうございました