

# マイクロサービスによるシステム設計

キヤノン株式会社

株式会社東芝

古谷浩平

常木健介

株式会社デンソー

日本電子計算株式会社

稲川拓

原島優太

## 開発における問題点

近年のシステムにおいては変動する顧客ニーズに対応していくため、柔軟なシステム設計が必要とされている。このような要望に応えるためのアーキテクチャの一つとしてマイクロサービスアーキテクチャがある。しかしながら設計初心者にとっては設計の難易度が高く、躓きが発生してしまいうる。

## 解決のためのアプローチ

現実的なシステムを実際に設計初心者でマイクロサービスアーキテクチャでの設計を実施し、どのような場面で躓きが発生するかを検証した。また、そのような際にどのような手法により躓きから脱却することが出来るかを模索した。

# アプローチ

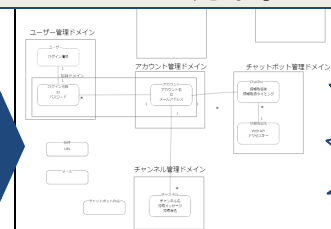
## ユースケース図

ユースケース記述: アカウントを作成する	
<u>ユースケース記述: チャンネルへ投稿する</u>	
ユースケース	チャンネルへ投稿する
アクター	チャットユーザ
前提条件	チャットサービス利用権限(表示) / Chatサービス利用ユーザ—
実行条件	<ul style="list-style-type: none"> <li>Chatサービス利用権限(表示) / Chatサービス利用ユーザとしてログインしていること</li> <li>投稿用のチャンネルが存在していること</li> </ul>
事後条件	投稿したメッセージがチャンネルに表示されていること
代替	<ul style="list-style-type: none"> <li>1. チャンネルのメッセージ投稿権限を表示する</li> <li>2. メッセージを入力する</li> <li>3. メッセージを投稿を実行する</li> <li>4. メッセージ投稿時のメッセージを表示する。メッセージ投稿が完了する</li> </ul>
例外	代替条件
	例外条件
	備考

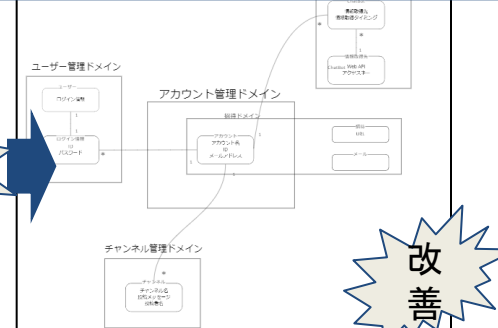
## ユビキタス言語一覧

[illegible]

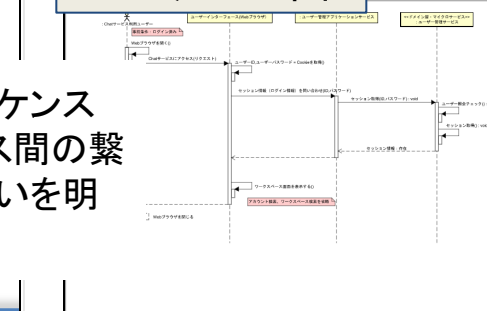
## 概念クラス・サービス分割



## 概念クラス・サービス分割



## シーケンス図



ドメインエキスパートの不足  
→サービスへの理解不十分  
に起因する躓きが発生

設計経験者の不在  
→サービス振舞いへの  
理解不十分に起因する  
躓きが発生

名詞によるグループ化を実施し、データに責任を持つサービスを明確化

取り組みやすいシーケンス  
図を作成し、サービス間の繋  
がりや必要な振る舞いを明  
確化

# 設計変更の影響

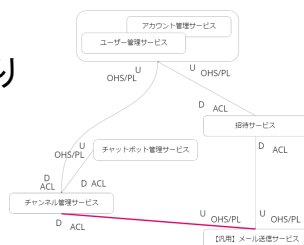
## 設計したシステムに仕様変更を適用したところ

□ユビキタス言語に基づく設計により、  
変更箇所の特定制が容易であった。

□各サービス間連携に適した設計により設計変更が軽微であった。

□サービスごとの変更の影響が他サービスに波及しにくかった。

マイクロサービスアーキテクチャに求められる  
変更容易性のあるシステム設計を実施可能であった。



## 評価・考察

名詞によるグループ化やシーケンスからの検討といった手法によりマイクロサービスアーキテクチャ設計が出来た。

保守性	仕様変更があっても変更するサービス・箇所 が漏れなく特定しやすかった。
信頼性	サービス間を非同期接続することで一部の 障害の全体への波及を避けることが可能
移植性	サーバーやDBのリプレースがあった場合 でも移植しやすい。