

アプリ連携を実現する協調アーキテクチャの研究

鈴木鷹[†] 菅原溪[†] 松田来央[†] 森田純恵[†]

秋田県立大学システム科学技術学部情報工学科[†]

1. はじめに

Society4.0では知識や情報の共有が不足し、分野横断的な連携が不十分であった。しかし、Society5.0が実現する社会では、IoTを中心にして、すべての人とモノをつなぎ、様々な知識や膨大なデータが共有されることで、新たな価値が生み出されると期待されている。Society5.0を実現するには、IoTデバイスを通じて蓄積されたデータの効果的な活用が不可欠であることから異分野融合、横連携、協調領域の考え方が重要である。その実装において、エンタープライズアーキテクチャは、エンタープライズアーキテクチャを構成する各アーキテクチャ層の横連携、協調領域が重要とされている。この考え方はSociety5.0のビジョンに共通している。本研究では、Society5.0を実現するためには、どのようなアーキテクチャが最適であるかという評価指標を探索する。

2. 先行研究と本研究の目的

アプリ連携を行っている先行事例として、急激な進化を遂げているメルカリ・メルペイを選び、調査を行った結果、複数のアーキテクチャを組み合わせ採用していることが明らかとなった。また、どのアーキテクチャを採用するかについては専門家の経験や勘をもとに行われているということが判った。

先行研究では、アーキテクチャの評価方法のサーベイ論文をレビューした[1]。そこでは、評価モデルは提案されているが、実際のケーススタディにおいて一部の測定項目は、専門家によって評価が行われており、評価結果は明らかにされていない。

これらのことからアーキテクチャの決定や評価は専門家によって異なるため、一般化する必要があると考える。これが本研究の目的である。

3. 代表的なアーキテクチャの調査

本研究の調査対象とする代表的な3つのアーキテクチャを比較評価したものを図1に示す。この3つのアーキテクチャは、先行事例であげたメルカリ・メルペイとの関係性も深く、このアーキテクチャに提案する評価指標をあてはめることで本研究を進める。それぞれの概要を以下に示す。

1) クリーンアーキテクチャ

アプリケーションを複数のレイヤー(層)に分割す

ることで、それぞれの役割と責任を明確にする。

2) マイクロサービスアーキテクチャ

1つのアプリケーションをビジネス機能に沿った複数の小さいサービスの疎に結合された集合体として構成する。

3) エンタープライズアーキテクチャ

企業全体のシステムを統一的手法でモデル化し、業務とシステムの最適化を図る手法。

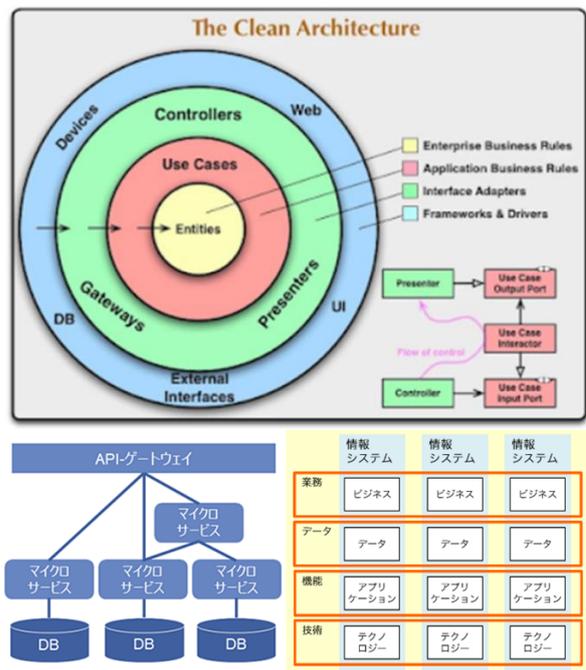


図1. 調査対象とする3つのアーキテクチャ

4. アーキテクチャ評価指標としての品質指標

アーキテクチャ評価モデルと品質項目は、「Decision Making for the Software Architecture Structure Based on the Criteria Importance Theory」[2]を参照した。図2は、品質項目と測定関数の算出方法とその階層化を示す。

図2を以下に説明する。まず、最下位層であるレベル4には、任意のアーキテクチャxがある。

得られた個々のメトリクスは、階層構造の一部として使用される。レベル3に11個の個別メトリクスがあり、 f_1, \dots, f_{11} と表記する。次のステップでは、個々のメトリクスをグループにまとめ、論理的なグループ化によって集約されたメトリクスが得られるようにする。したがって、レベル2では個々のメトリクスを不連続集合にグループ化し、新しい基準ベクトルを形成する。同様の操作がレベル1でも行われる。

Reserch on cooperative architecture for application coordination
[†] You Suzuki, Kei Sugawara, Kio Matsuda, Sumie Morita
 Akita Prefectural University[†], Faculty of Systems Science and
 Technology, Department of Computer Science and Engineering

最後に最上位レベル(レベル 0)で、単一のベクトル基準 $f_{1234567891011}^0$ が得られる。

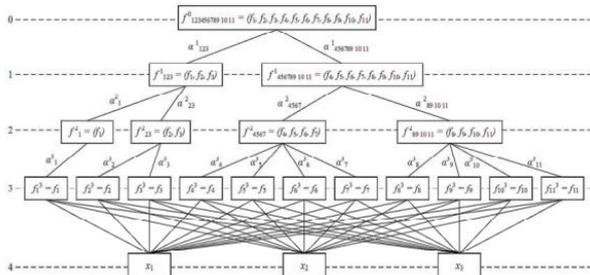


図 2. 5 段階の階層とそれに対応する重要度係数 α

図 3 は、図 2 で使用するメトリクスを階層構造で表したものであり、レベル 3 以上の基準ベクトルとリンクする。

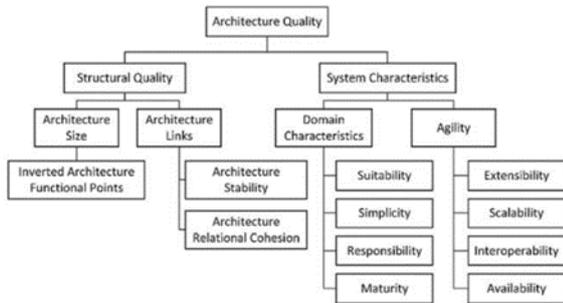


図 3. ソフトウェアアーキテクチャのメトリクス階層

5. メルカリ・メルペイのアーキテクチャの測定

先行事例であげたメルカリ・メルペイでの調査からメルペイ PaymentPlatform の機能は、図 4 のように構成されている。メルペイ PaymentPlatform のシステムはマイクロサービスを採用しており[3]、機能においてはクリーンアーキテクチャを採用している[4]。

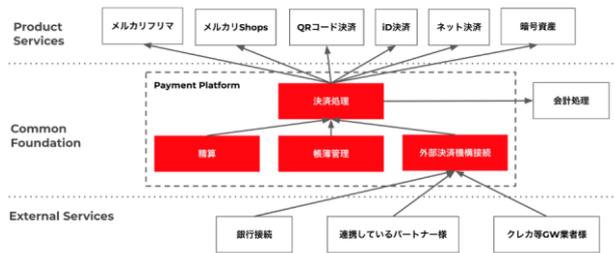


図 4. メルペイ PaymentPlatform 機能構成図

4 章で紹介したモデルにあわせたメルペイ PaymentPlatform のアーキテクチャにおける品質は、以下のように測定することができる。

- 1) Level14: 図 4 に示すそれぞれのコンポーネント
 - 2) Level13: 11 個の個々のメトリクスの算出
 - 3) Level12: 図 3 に示す 4 分類に集約して算出
 - 4) Level11: システムの特徴とその構造を算出
- しかし、メルカリのソースコードは非公開である

こと、外部からの動作検証では 11 個のメトリクスの全てが測定できるものではない。これからの測定活動は、メルカリのエンジニアリングチームが実施していると想定し、今後公開される可能性はある。

6. アーキテクチャ品質測定の実践

上記よりアーキテクチャの品質測定は、オープンソースか自作のソフトウェアで、かつ、手元で実行できることがやはり求められる。都市 OS である FIWARE 等も候補の一つであるが、現状、本学研究室で設計している FIWARE に近い動的情報基盤システムに本モデルを適用、アーキテクチャの測定を行う。図 5 にその動的情報基盤のアーキテクチャを示す。

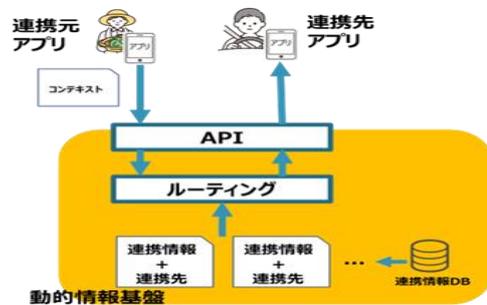


図 5. 動的情報基盤のアーキテクチャ

7. まとめと今後の課題

本稿では、先行研究で提案されたモデルを活用してアーキテクチャの評価を行った。対象となるアーキテクチャの評価は順調に進んだが、品質測定においては、重み付け係数を設定する際に専門家の経験や勘が不可欠であるという課題が残った。今後、オープンソース化されているシステムにおいて、数値に基づく客観的な重み付け方法の導入が必要である。

参考文献

[1] Samira Silva, Adiel Tuyishime, et al, 「Quality Metrics in Software Architecture」, 2023 IEEE 20th International Conference on Software Architecture (ICSA), 13-17 March 2023

[2] Sergey Orlov, Andrei Vishnyakov, 「Decision Making for the Software Architecture Structure Based on the Criteria Importance Theory」, ICTE 2016, December 2016

[3] 「メルカリ Shops を支えるメルペイ PaymentPlatform の仕組み, メルカリエンジニアリング」, <https://engineering.mercari.com/blog/entry/2021217-c38d98fbf6/>

[4] 「マイクロサービスにクリーンアーキテクチャを採用する上で考えたこと, メルカリエンジニアリング」, <https://engineering.mercari.com/blog/entry/2019-12-19-100000/>