

## スマートスピーカアプリケーション共通化プラットフォームの検討 Study on cross platform for smart speaker applications

望月 龍一<sup>†</sup> 降旗 佑季<sup>†</sup> 野澤 啓亮<sup>‡</sup> 大谷 瑛佑<sup>‡</sup> 横内 直樹<sup>‡</sup> 土屋 健<sup>†</sup>  
Ryuichi Mochizuki<sup>†</sup> Yuki Hurihata<sup>†</sup> Keisuke Nozawa<sup>‡</sup>  
Eisuke Otani<sup>‡</sup> Naoki Yokouchi<sup>‡</sup> Takeshi Tsuchiya<sup>†</sup>

### 1. はじめに

現在, Google により販売されている Google Home[1], Amazon の Alexa[2]などスマートスピーカが普及している. 各スマートスピーカはユーザの発話内容を認識する機能を持ち, 音声での対話を通してサービスを提供することが可能である. 利用可能な様々なサービスはアシスタントのスキルとして提供されている. アプリケーションの開発者は独自にスキルを実装することにより, ユーザとの会話を通したサービス提供が可能となっている.

スマートスピーカで提供可能なサービスは OS などの主要コンポーネントに依存している. そのため, サービスが独立して提供されている. このことから, 現在の Web において一般的に利用されているセッション, 動的ページへの適応などを考慮すると, サービスの利用性と開発性が低い点が課題となる.

本研究では, スマートスピーカを一般の Web API により稼働するアプリとの接続と, 共通化を行うためのプラットフォームを明らかにする.

### 2. Web アプリとの接続, 共通化手法

クラウドの機能の一部をネットワークのエッジ側, IoT デバイスの近傍に配置するコンピューティングモデルが提案されている[3]. フォグノードは計算資源, ストレージを持ち, ミドルウェアとしてデータ処理と制御, クラウドへの伝送を透過的に実現する. 本研究では, スマートスピーカと Web サービス間にプロキシに類似する形でフォグコンピューティングを適用し, ユーザのサービス利用状況に適応しデータ処理を実現する. フォグノードが収集したユーザデータに基づいてアプリケーションの提供を行う.

具体的に, ユーザの利用するサービスに基づき動的に Web 上からデータを取得し処理を行う手法, またフォグノード上に蓄積するためのプロトコル形式, ユーザのサービス利用状況に合わせ, Web 上のサービスとフォグノード上で行う処理内容を関連付けるサービス管理技術を検討する.

#### 2.1. Web サービスデータの取得

本検討では図 1 に示す形で, Web アプリケーションからの情報取得, データ処理を, フォグノードが実行するサービスとして管理を行う. データ処理サービスは Web サービスを提供しているサーバやクラウド上に任意に配置を行う. サービス実行のため, スマートスピーカアプリケーションの利用者に対して, フォグノードを割り当てる. このフォグノードがサービスをプラグインとして導入することで, 各 Web アプリケーションの情報収集を行う. Web サービスとフォグノード上で管理するサービスの関連付けはサービスの管理ノードを配置することにより行う.

フォグノードは複数のサービスを実行し, 処理結果を一つのデータベースに集約する. これにより複数サービスに対してのスマートスピーカアプリケーションの共通化を行う.

ユーザのサービス利用状況に適合する形で情報収集を行うため, ユーザの利用するサービス情報に加えて, セッション情報, 認証情報がフォグノードに記録される. 各サービスはフォグノード上からこれら情報を取得し通信, 処理を行う. サービスの信頼性は, Web サービスと同一のドメインを付与したノード上に処理を配置することにより検証を行うことが考えられる.

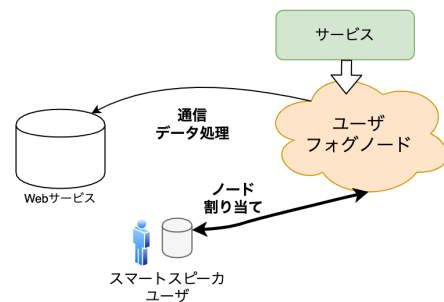


図 1 Web サービスデータの処理

#### 2.2. ユーザ, サービスの管理

ユーザの利用するサービスの特定, 管理を行うことを目的として, 図 2 に示すように Web ブラウザによるオプトイン形式のサービス管理手法を提案する. ユーザの Web ページ閲覧時, URL に基づいて利用可能なサービスを検索する. 該当するものが存在した場合, ユーザに提案する. サービスの検索は, サービス管理ノードにサービスと URL の関連付けの解決問い合わせを行うことにより行う.

ユーザがサービスの利用を行う場合, ユーザの情報の処理, 管理を行うフォグノード上に, サービスの登録を通知する.

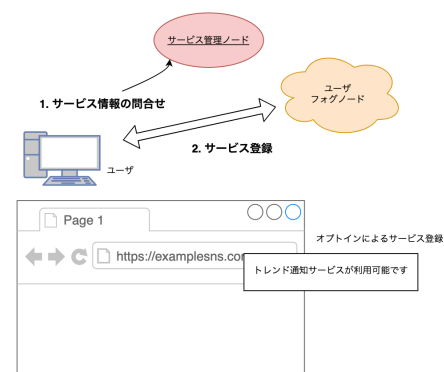


図 2 ユーザサービス管理

### 2.3. スマートスピーカとの連携

各サービスで処理を行なったデータはスマートスピーカでの発話、通知への利用のため、フォグノード上の共通データベース上で管理を行う。以下に保存形式の JSON データを例として示す。発話内容を示すデータを SSML 形式で記述する。またプッシュ通知有無やスマートスピーカとの相互対話を考慮したセッション情報など、発話のタイミングのデータを含める。現在スマートスピーカは音声メディアの利用のほか、画面表示機能を備えるものも存在するため、これら拡張のための情報を保存する。

```
{
  "service": "chat-notification",
  "ssml": "<speack>sample さんの新しい発言があります</speack>",
  "content-type": "notification",
  "media": {
    "image": "https://example.com/api/v1/..."
  }
}
```

現状、スマートスピーカに対するアプリケーションの提供は、各社のプラットフォームに依存する。そのため Google Home[1], Amazon Alexa[2]を例に、図 3 に示す形でそれぞれ対話用のアプリケーションを配置する。各社のクラウドサービスがアプリケーションを実行する。このとき、ユーザのフォグノードと接続するためユーザ（サービス）管理ノードに問い合わせを行う。発話したユーザのフォグノードを特定し通信、発話すべき情報が存在した場合などに発話する。

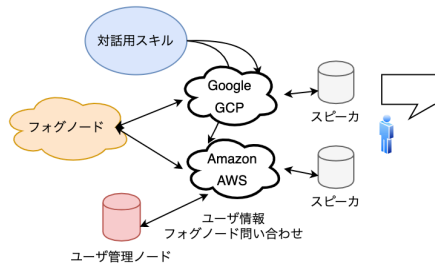


図 3 スマートスピーカとの連携

### 3. 提案プラットフォームの実装

現在の実装ではユーザデータの処理とデータベース管理を行うフォグノードを Node.js[4]により実装している。図 4 に示すように、ユーザからのサービス利用開始要求、データの追加と取得を Web API の提供により行っている。

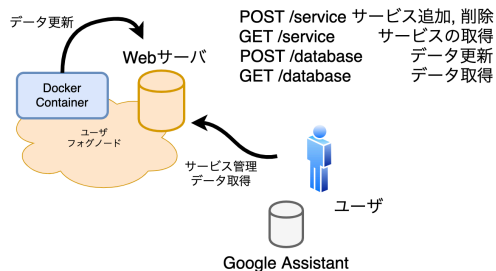


図 4 フォグノードの実装

サンプルのサービスとして現在では Open Weather Map Web API [5]より気象情報を取得するものを実装している。サービス利用開始時に住所情報をフォグノードに保存し、これを取得することで動的な Web API の参照を行なっている。サービスの実行環境は Docker[6]によるコンテナ型仮想環境を利用しており、Web サーバ上に実行可能なイメージを配置することで、Docker を導入している全てのユーザノードで共通のサービスを実行することが可能である。

ユーザはフォグノード上で提供される Web GUI からサービスの管理、データの確認が行える。実行中のノードの GUI を図 5 に示す。図中左側が実行中のサービスであり、右側にサービスにより収集されたデータを示している。

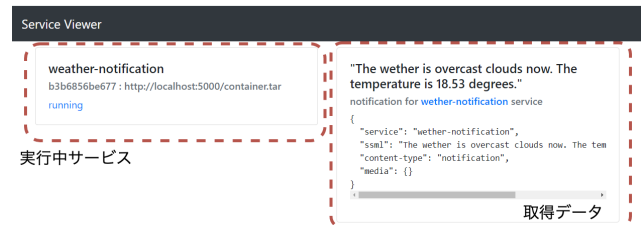


図 5 サービス管理画面

### 4. まとめ

スマートスピーカで提供可能なサービスは OS などの主要コンポーネントに依存している。そのため、サービスが独立して提供されておりサービスの利用性と開発性が低い点が課題となる。本研究では、スマートスピーカを一般の Web API により稼働するアプリとの接続と共通化を行うためのプラットフォームを検討した。提案手法ではスマートスピーカユーザにデータ処理と管理を行うフォグノードを配置する。フォグノードが動的にユーザのサービス利用状況を管理、Web API の操作や Web ページのクロールを行う。

今後の課題として、ユーザのサービス管理機能の実装。また、本提案手法がアプリケーションとして適用可能な Web サービスの検討を早急に行う必要がある。現状、SNS やチャットツールからの情報取得、通知機能について検討している。このため Cookie や Websocket を用いた動的ページからの情報取得を可能にするサービスの検討、アプリケーションの実装を行うことを予定している。

### 参考文献

- [1] Google Home, [https://store.google.com/jp/product/google\\_nest\\_hub](https://store.google.com/jp/product/google_nest_hub).
- [2] Amazon Alexa, <https://www.amazon.co.jp/meet-alexa/>.
- [3] A.Alrawais, A.Alhothaily, C.Hu, X.Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues", IEEE Internet Computing ( Volume: 21 , Issue: 2 , Mar.-Apr. 2017 ), PP. 34 - 42
- [4] Node.js, <https://nodejs.org/>.
- [5] Open Weather Map, <https://openweathermap.org/>.
- [6] Docker, <https://www.docker.com/>

† 公立諏訪東京理科大学 Suwa University of Science

‡ ビップシステムズ Bip systems