

愛知工業大学情報科学部情報科学科  
コンピュータシステム専攻

令和2年度 卒業論文

時空間フェンシングに基づく  
クラウドセンシングプラットフォームに  
関する研究

2021年2月

研究者 K17090 土本涼雅  
K17125 宮川信人

指導教員 梶克彦 准教授

# 目次

<b>第 1 章</b>	<b>はじめに</b>	<b>3</b>
1.1	研究背景 . . . . .	3
1.2	クラウドセンシングの課題 . . . . .	4
1.3	研究目的 . . . . .	5
1.4	論文構成 . . . . .	6
<b>第 2 章</b>	<b>関連研究</b>	<b>7</b>
2.1	スマートフォンによるセンシング . . . . .	7
2.2	クラウドセンシング . . . . .	7
2.3	クラウドセンシングプラットフォーム . . . . .	8
<b>第 3 章</b>	<b>時空間フェンシングに基づいたクラウドセンシングプラットフォーム</b>	<b>10</b>
3.1	時空間フェンシングの定義 . . . . .	11
3.2	ラヴラスの要求仕様 . . . . .	12
3.2.1	ラヴラスにおけるクラウドセンシングプラットフォームとしての設計基盤 . . . . .	12
3.2.2	プライバシーの保護 . . . . .	12
3.2.3	ラヴラスの利用規約方針 . . . . .	13
3.2.4	利用者への配慮 . . . . .	14
3.3	ラヴラスの実装 . . . . .	15
3.3.1	ラヴラスの操作手順 . . . . .	15
3.3.2	センシングデータの管理や各プロジェクトを管理するサーバの実装 . . . . .	17
3.3.3	依頼者用のプロジェクト管理 Web アプリの実装 . . . . .	22
3.3.4	協力者用のセンシングスマホアプリの実装 . . . . .	32
3.4	動作検証 . . . . .	35
<b>第 4 章</b>	<b>おわりに</b>	<b>37</b>
4.1	まとめ . . . . .	37
4.2	今後の課題 . . . . .	37
	<b>謝辞</b>	<b>39</b>
	<b>参考文献</b>	<b>40</b>

# 第1章 はじめに

## 1.1 研究背景

近年、高機能センサを備えたスマートフォンが増加している。内蔵されているセンサの例として、加速度や角速度、磁気、気圧、Wi-Fi、BLE、音などが挙げられる。これらのセンサは、正確で高精度な測定データの取得が可能のため、スマートフォンの3次元での位置や動作、周囲の環境のモニタリングが可能である。例えば、ユーザの現在位置や動き、ゲームでは、スマートフォンを振る・傾けるなどの複雑な操作や動きの測定が内蔵センサによって可能となる。追加モジュールによっては、花粉やPM2.5、風速や湿度などの計測も可能となる。

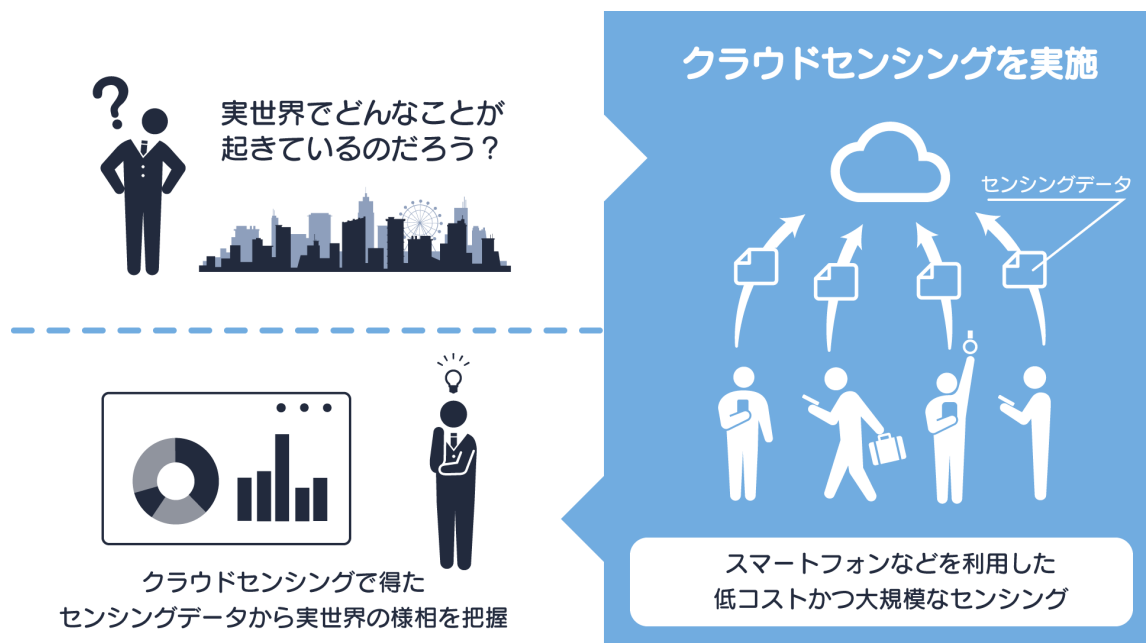


図 1.1: クラウドセンシングによるデータ収集の意義

このスマートフォンのセンシング能力を活かす試みとして、クラウドセンシングがある。クラウドセンシングの説明図を図 1.1 に示す。クラウドセンシングとは「群衆の持つスマートフォンなどの携帯端末に内蔵されたセンサを用いて低コストで大規模にセンサデータ取得し、そこから実世界の様相を把握するための方法論 [1]」である。クラウドソーシングとセンシングを掛け合わせたものである。参加型センシングとも言われる。スマートフォンの登場以前は、センサデータを多くの人から取得するのは現実的ではなかった。クラウドセンシング専用のデータ収集端末を開発しても、収集したデータの共有は困難であった。スマートフォンは文字入力、センシング、通信機能など様々な機能を備えているため、クラウドセンシングを行うための端末として非常に適している。スマートフォンの登場によって、クラウドセンシングという考え方が進んだと言っても過言ではない。クラウドセンシングの技術的背景としては、スマートフォンの内蔵センサの小型化・高性能化が挙げられる。センサの精度は初期のものに比べ、格段に良くなっている。また、どんどん小型化・スマート化し、電力消費量も少なくなっている。もう 1 つの技術的背景として、通信速度の高速化が挙げられる。現在は 4G（第 4 世代移動通信システム）が主流となつて

いるが、2020年には5G（第5世代移動通信システム）のサービスが開始された。5Gの通信速度は4Gの20倍、遅延は4Gの10分の1、4Gの10倍のデバイスに接続が可能となる[2]。クラウドセンシングの社会的背景としては、スマートフォンの急速な普及が挙げられる。2007年にApple社から「iPhone」、2008年にはAndroid端末が発売され、全世界に「スマートフォン」が爆発的に普及されていった。日本でも主流は「フィーチャーフォン」から「スマートフォン」に変化した。2010年はスマートフォンの世帯保有率は9.7% [3]で、まだ全体的な普及はしていなかった。この時点ではクラウドセンシングに協力するしないにかかわらず、まずクラウドセンシングに協力するためのスマートフォンを所持しているかしていないかで対象者が振り分けられる。つまり、対象者の母数が圧倒的に少なかった。クラウドセンシングは収集できるデータ数が多ければ多いほど価値が高くなるため、スマートフォンを所持していない2010年頃はクラウドセンシングを利用するには適していなかった。2019年にはスマートフォンの保有率は世帯では83.4%と2010年と比べ約9倍、個人では67.6% [3]となり、半数以上の人がスマートフォンを所持するようになっている。所持している人全員がクラウドセンシングに協力してくれるとは限らないが、明らかに母数は増えているため、現在はクラウドセンシングを利用するのに適している。

クラウドセンシングは幅広いデータ収集かつセンシングコストを削減できるため、様々な研究で採用されている。例えば、街頭の明るさ調査、観光スポット調査、騒音調査、温度調査などの研究が進められている[4]。クラウドセンシングは研究だけではなく、自治団体や地域施設の管理者など、様々な人にとって有効になりうる可能性がある。例えば先述の騒音調査や公害物質調査などは、自治団体にとって地域の住みやすさ改善のための重要なデータとなり、そのデータを基によりよい地域や町づくりが可能となる。また、公園管理者にとってはいつどのような場所に人が集まるのか、危険な箇所で子供が遊んでいないか、といった施設の安全確保に有効なデータを収集できる。

## 1.2 クラウドセンシングの課題

幅広いデータ収集かつセンシングコストを削減できるクラウドセンシングにはいくつかの課題がある。大きく分けて依頼者側・協力者側の課題がある。本研究では、クラウドセンシングを利用しデータ収集を行う人を依頼者、クラウドセンシングに協力しスマートフォンでデータ提供を行う人を協力者とする。

依頼者側の課題として、サーバやアプリなど専用システムの開発にかかるイニシャルコストやランニングコストが挙げられる。依頼者がクラウドセンシングを利用するためには、協力者のセンサデータを収集するための専用アプリや収集したセンサデータを管理するための専用サーバが必要となる。自治体や地域施設の管理者などシステムの開発知識がない人にとっては自ら開発するのは困難であり、業者に委託するにも費用や手間などが大きくかかる。また、知識のある研究者にとっても、システムの開発という研究の本質からずれた作業をしなければならないため、研究の速度は下がってしまう。そのため、依頼者のクラウドセンシングに対するハードルは高くなっている。

協力者側の課題として、物理的及び心理的コストといった課題が挙げられる。物理的コストとしては個別のクラウドセンシングアプリケーションのインストールや利用などにかかる手間や協力者の操作や通信といった負担が挙げられる。心理的コストとしては協力者のプライバシー障壁によるデータ提供への心配・不安、それらによるストレスなどが挙げられる。

まず協力者がクラウドセンシングに参加するためには、個別のクラウドセンシングアプリケーションをインストールしてもらい、継続的に利用してもらう必要がある。多くの人にとって、個別のアプリケーションのインストールはコストが高い。例えば、3人の依頼者がいた場合、専用のクラウドセンシングアプリケーションは3つある。そのため、3つのクラウドセンシングにすべて協力するとなった場合、3つもアプリケーションをインストールしなければならない。もっと協力するとなると、その分インストールしなければならないアプリケーションの数も増える。それにより、協力者の負担も増加するといえる。また、継続的に個別のアプリケーションを利用してもらうのも困難である。多くの人々は「一定期間利用しないアプリケーションは削除したい」といった欲求があるため、一定期間クラウドセンシングを行わないとアプリケーションを削除されてしまい、いざデータを収集したいとなったときには出来ない可能性が

ある。

また、協力者のアプリケーション内での操作やアプリケーションを使用する際のデータ通信量などの負担が多いと、協力者はアプリケーションを放置または削除してしまう。たとえ多くの協力者の獲得できたとしても、アプリケーションの操作や設定が複雑であったり、センシングデータの通信で協力者のデータ通信量を多く使ってしまったりする場合、協力者にとってクラウドセンシングが億劫に感じてしまうため、多くの収集データ量は見込めない。そのため、センシングやセンシングデータアップロードのために必要なスマートフォン操作や通信コストを最小限に抑える必要がある。

そして、第三者へのセンシングデータ提供に対する不安や個人情報悪用の心配などのプライバシー意識により協力者の獲得は容易ではない。もし、依頼者と協力者が信頼しあっており、データ提供に対する不安がなければ、センシングに対する協力は容易である。しかし、依頼者と協力者が赤の他人の関係であったら、協力者は自分自身の個人情報が悪用されるのではないかと疑ってしまい、データ提供の可能性は低い。また、協力者は依頼者を信頼していても、データ提供してもいい時とデータ提供したくない時がある。そのため、ある程度センシングする範囲や時間帯を絞って、協力者の承諾を得る必要がある。

また、クラウドセンシングではスマートフォンのセンサ情報や位置情報などの個人情報を多く取り扱うため、セキュリティやプライバシー保護の対策も万全である必要がある。依頼者と協力者の信頼関係が成り立っていても、データ管理のセキュリティが甘いと、第三者に悪用されてしまう場合がある。そのため、クラウドセンシングを利用するには、セキュリティやプライバシー保護への対策ができていなければならない。

## 1.3 研究目的

本研究ではクラウドセンシングの簡易的な利用と多様なデータ収集を行い、研究や調査におけるインシタル及びランニングコストの大幅な軽減を目的とする。依頼者側で課題となっていた専用システムを個別で開発する必要をなくし、センサの種類を選択するだけで簡易的にクラウドセンシングが利用可能となるシステムを実現する。また、加速度センサのみなどの単一のセンサデータだけでなく、“加速度センサと角速度センサ”や“加速度センサと気圧センサと Wi-Fi”など多様なセンサデータの収集により、どの研究や調査にも柔軟に対応する。

そして、協力者の発生し得る物理的及び心理的コストの軽減し、ユーザのセンシングの協力かつ継続を促進する。協力者側の物理的コスト面での課題となっていた個別アプリケーションのインストールをなくし、共通のアプリケーション1つでどのクラウドセンシングにも協力可能にする。それにより、協力者のインストールに対する負担を軽減し、クラウドセンシングの依頼が頻繁に行われていればアプリケーションが削除される心配もなくなる。また、スマートフォンの操作を最小限にし、センシングデータアップロードは Wi-Fi など携帯回線以外に接続されている場合のみに行う。心理的コスト面での課題となっていたセンシングデータ提供に対する不安は、依頼者の情報や目的や使用するセンサの種類などのセンシング内容の提示により払拭する。依頼者やセンシング内容に納得しセンシングの依頼を承諾した場合のみのセンシング開始により、勝手にセンシングされたり、センシングされたくないプライベートな時間にセンシングされるなどがなくなる。これにより、協力者のセンシングに対するハードルを下げられる。

我々はこの目的を実現するためにまず時空間フェンシングの概念を提案する。時空間フェンシングとは、地理的な場所を制限するジオフェンシングに時間要素を加えて拡張し、時間と場所で境界を区切る独自のフェンシング手法である。これにより、依頼者はシチュエーションを指定したデータ収集が可能となる。協力者はセンシングを承諾するか否かの判断がしやすくなる。

そして、この時空間フェンシングに基づいたクラウドセンシングプラットフォームを構築する。依頼者は Web アプリケーションでセンシング依頼の定義を行い、協力者はスマートフォンアプリケーションでセンシングを行う。これにより、依頼者は時間帯やエリア、目的、概要、センサの種類など必要な情報の入力のみでクラウドセンシングが可能となる。また、データ収集にかかるコスト削減かつ柔軟なセンシングが可能となる。協力者のセンシング協力には共通のスマートフォンアプリケーション1つでいいため、インストールのコストを軽減できる。さらに、依頼者が新規のクラウドセンシングを行う際に新規の協力

者を募らなくても、既に多くの協力者が本プラットフォームのスマートフォンアプリケーションを利用している状況になるため、協力者獲得が容易となる。

## 1.4 論文構成

本稿の構成は以下の通りである。2章では、クラウドセンシング及びクラウドセンシングプラットフォーム関連の既存研究を紹介し、その本研究との関連性を述べる。3章では、時空間フェンシングを定義し、それに基づいたクラウドセンシングプラットフォームの要求仕様・実装について述べ、動作検証を行う。最後に4章では、まとめと今後の課題について述べる。

## 第2章 関連研究

本章では本研究の関連研究について述べる。2.1 では、スマートフォンを用いたセンシングとジオフェンシングの関連研究について述べる。2.2 では、1 章で説明したクラウドセンシングの関連研究について述べる。2.3 では、クラウドセンシングの基盤となるプラットフォームの運用例や関連研究、それらの協力者に対するモチベーションの維持・向上方法について述べる。

### 2.1 スマートフォンによるセンシング

スマートフォンは、歩行動作や人々の振る舞いなど行動のセンシングなどに用いられている。例えば、スマートフォンの 3 軸加速度センサを用いて収集した移動データから特徴量を抽出し、歩行者を推定する研究 [5] などがある。また、歩行の推定は加速度センサだけではなく、ジャイロセンサを用いて曲がりの推定 [6] や気圧センサを用いて在階や高度の推定 [7][8] もされている。加速度センサとジャイロセンサを用いてデータ収集し、立ち止まる、振り向くなどユーザの興味行動 [9] や来た道を引き返す、うろうろするなどユーザの迷い行動 [10] を検出する研究などもある。スマートフォンは日常的に多くのユーザが利用しており、小型で持ち歩きやすいため、ユーザの普段の振る舞いをセンシングするには適している。また、高精度のセンサが搭載されているため、新規のセンシング専用端末の開発や購入といったコストを省ける。スマートフォンは通信機能も優れており、有線で繋がずとも収集したデータをそのままサーバにアップロードできるため、データの集積は容易である。

位置推定の高精度化や手軽にジオフェンスを構築できる BLE のようなデバイスの普及に伴って、様々なジオフェンシングを用いた研究が行われている。例として、危険エリアにジオフェンスを作成し、危険にさらされているユーザに対して災害情報やアドバイスを送ったり [11]、農場にジオフェンスを作成し、農場に入った人にそれぞれの農場にあった出荷予定日や天気情報などの情報提示を行う研究 [12] などがある。ジオフェンシングは研究だけではなく、スーパーや飲食店でも活用できるサービスである。例えば、店内に入ったらポイントやクーポンがもらえたり、特定のエリアに行くとお得な情報がゲットできるようになれば、買い手の増加も期待できる。ジオフェンシングにより、実施側はエリア内にいるスマートフォンユーザを把握でき、エリアにあった適切なサービスや情報の提供が可能となる。また、収集したデータを基にして、サービスの改善や情報の更新などにも役立つ。エリア内にいるスマートフォンユーザにとってもエリアにあったサービスや情報は非常に有益であるため、両者の相互利益をもたらす。

多様なデータ収集とクラウドセンシングの簡易利用が可能となるプラットフォームの利用により、データ収集するためのシステムやアプリケーションを開発する時間や手間、費用などを削減し、これらのような研究のスピードアップが期待できる。研究のスピードアップにより、より効率的に社会実装が進められたり、研究の深い追求が可能となる。センシングを基に処理を行う研究などには直接的にプラットフォームの活用はできないが、実装前のセンサの精度調査や推定などに有効である。

### 2.2 クラウドセンシング

クラウドセンシングを利用して多くの人々からデータ収集し、推定や分析をする研究はいくつかある。西村らの研究 [13] は、スマートフォンの加速度センサとマイクからスマートフォン保持者の歩行動作と周辺の雑踏音をそれぞれセンシングし、端末周辺の混雑状況の推定を行っている。混雑状況の推定の研究としては、バス内の混雑状況を加速度データと角速度データから推定する研究 [14] もある。これは混雑時

にバス利用者が他のバス利用者を避けるために、体を横に捻ったり、肩をすばめて移動したりする回避動作のデータを収集し、そのデータから混雑状況を推定するというものである。このような場合、収集データ量が多ければ多いほどより詳しく混雑状況を推定できる。朴らの研究 [15] では一般の自動車利用者から加速度センサなどのモーションセンサを用いてデータを収集し、凍結や舗装路などの路面状態や平坦やくぼみなどの路面形状の推定を行っている。

これらの研究ではクラウドセンシングシステムの開発などには大きなコストがかかると考えられる。依頼者がクラウドセンシングを実施するためには、協力者専用のセンシングスマートフォンアプリケーションや収集したデータを管理するサーバなどを開発する必要がある。協力者が依頼者の友人または研究仲間の場合、操作方法は口頭で補えばよいので、デザインや仕様にこだわる必要はない。しかし、協力者が赤の他人である場合、アプリケーションの操作方法を初めてでも理解しやすく簡易的にするなど工夫する必要がある。また、センサなどの種類やサンプリングレートをあらかじめ指定してアプリケーションを作成してしまうと、途中で変更したい場合に再インストールなどしなければならず、依頼者・協力者共に負担がかかってしまう。アプリケーション内でセンサなどの種類やサンプリングレートを変更する機能があっても、協力者の操作が必要であり、負担は増えてしまう。そのため、クラウドセンシングを利用してデータ収集したい場合は、プラットフォームがあると非常に便利である。依頼者としては、研究毎にセンシング専用のスマートフォンアプリケーションを作成・配布する必要がなくなるため、それらに費やしていた時間や手間の省略が可能となる。また、一定の仕様やデザインはプラットフォームとしてあらかじめ決まっているので、一から考える必要はない。協力者としては、研究別のスマートフォンアプリケーションをインストールする手間が省け、研究毎に使い分ける必要もなくなる。そして、協力者のスマートフォンアプリケーションの利用により、協力者を他のセンシングに誘導も可能であるため、多くの協力者獲得にも繋がる。

## 2.3 クラウドセンシングプラットフォーム

実際に運用を行っているクラウドセンシングプラットフォームとして、Ohmage[16][17]やAWARE[18][19]などがある。Ohmageはセンサデータ収集だけではなく、質問を作成し、協力者にテキストや選択肢、動画、画像、音声といった方法で回答し、収集が可能である。また、収集したデータは可視化・分析でき、独自の統計分析のためにデータのエクスポートも可能である。AWAREとは、依頼者向けにセンサー計測によるモバイルコンテキスト情報の計測、推測、ログ記録、共有に特化したプラットフォームである。AWAREはプラグインが可能であるため、AWAREで作成したアプリの拡張・機能の追加をし、依頼者独自のクラウドセンシングが実施できる。

クラウドセンシングは協力者の確保が非常に重要であるため、様々な方法でモチベーションを向上・維持させる必要がある。協力者のモチベーションを向上させる方法として金銭的インセンティブ、非金銭的インセンティブがある。金銭的インセンティブとは、お金などの報酬である。金銭的インセンティブは、協力者を獲得し、協力者のモチベーションを維持するには非常に有効である。金銭的インセンティブを用いたクラウドセンシングプラットフォームとしては、LiveLabs[20]が挙げられる。多くの協力者を獲得できるものの、報酬には限界があり、協力者が多くなればなるほど依頼者側の負担が大きくなる。また、多くなった協力者を保持するための費用も多くかかる。負担が多いからといって報酬を無くしてしまうと、協力者のモチベーションは下がってしまい、センシングに協力的ではなくなる可能性がある。一方で、非金銭的インセンティブとは、お金などの報酬の代わりとして楽しさや体験を報酬としている。非金銭的インセンティブを提供する手法として、ゲーミフィケーションが挙げられる [21]。ゲーミフィケーションとは、ゲームの構造や考え方をゲームとは異なった分野に組み込んでゲーム化するという意味である。ゲーミフィケーションを用いた例として、河中らの研究 [22] が挙げられる。河中らのプラットフォームは観光情報収集を目的としている。ゲーミフィケーションの内容は、特定のエリア内を歩き回る「エリアミッション」と特定の場所で写真撮影を行う「チェックインミッション」があり、ミッションをクリアすると付与されるポイントで協力者同士競い合うといったものである。ポイントに集めている間に加速度や



角速度などのセンサの値を収集し、5 秒ごとにサーバに送信している。「もっとポイントを獲得したい」という思いがセンシングのモチベーションの向上に繋がるため、協力者維持には有用である。モチベーションを生成するために、金銭的インセンティブと非金銭的インセンティブを柔軟に選択できる松田らのクラウドセンシングプラットフォーム [23] もある。これは、デフォルトとしてモチベーション生成方法が決まっている訳ではなく、依頼者が決める。ポイントやレベル、バッジの有無や付与条件、ランキングの付け方などのゲーミフィケーションや、金銭的インセンティブなどをセンシングに応じて設定できる。このプラットフォームは ParmoSense[24] として既に運用されている。しかし、モチベーションを保つために、必ずしもインセンティブが必要であるとは限らない。先述の Ohmage や AWARE は、クラウドセンシングの依頼者のシステム開発コストの削減を主に目的としたシンプルなプラットフォームである。そのため、協力者を多く集め維持するといった部分は標準機能としては備わっていない。その場合、協力者のモチベーションは「センシングに協力したい、研究に貢献したい」という気持ちだけである。例えば、坂村らのまちづくりのためのプラットフォーム [25] は、「町や地域の問題を解決したい」や「まちづくりを促進したい」といった地域の人々の思いやボランティア精神でモチベーションを保っている。クラウドセンシングの目的やどのように社会や暮らしに還元されるかが明確であれば、インセンティブなしでも協力獲得の可能性はある。

## 第3章 時空間フェンシングに基づいたクラウドセンシングプラットフォーム

本章ではまず時空間フェンシングの概念を定義し、時空間フェンシングに基づいたクラウドセンシングプラットフォーム「Lav.<sup>+</sup>(ラヴラス)」を構築する。3.1 節では時空間フェンシングの定義を行い、時空間フェンシングによるクラウドセンシングの利点を述べる。3.2 節では時空間フェンシングに基づいたクラウドセンシングプラットフォームの要求仕様を述べる。3.3 節では本クラウドセンシングプラットフォームの手順を述べ、クラウドセンシングプラットフォームのサーバ、依頼者専用の Web アプリケーション (以下, Web アプリ), 協力者専用のスマートフォンアプリケーション (以下, スマホアプリ) の実装について述べる。3.4 節では本プラットフォームの動作検証について述べる。

本クラウドセンシングプラットフォーム「Lav.<sup>+</sup>」(以下, ラヴラス) の命名は, "a view of Laplace's demon" 「ラプラスの魔の視界」から来ている。ラプラスの魔とは, ピエール=シモン・ラプラスによって提唱された概念で「自然界のあらゆる力と宇宙全体のある時点における状態を完全に把握することができ, かつ, これらの素材を完璧に解析する能力をもった仮想的な知的存在。このような魔 (demon) にとっては宇宙の中に何一つとして不確実なものはなく, 未来のことを完璧な形で予見することが可能となる。」[26] というものである。本プラットフォームは, ラプラスの魔のコンセプトにちなんで, クラウドセンシングを用いて実世界の様相を把握し分析しラプラスの魔の視界を垣間見られる様なプラットフォームを目指すという意味を込めてラヴラスと命名した。

### 3.1 時空間フェンシングの定義

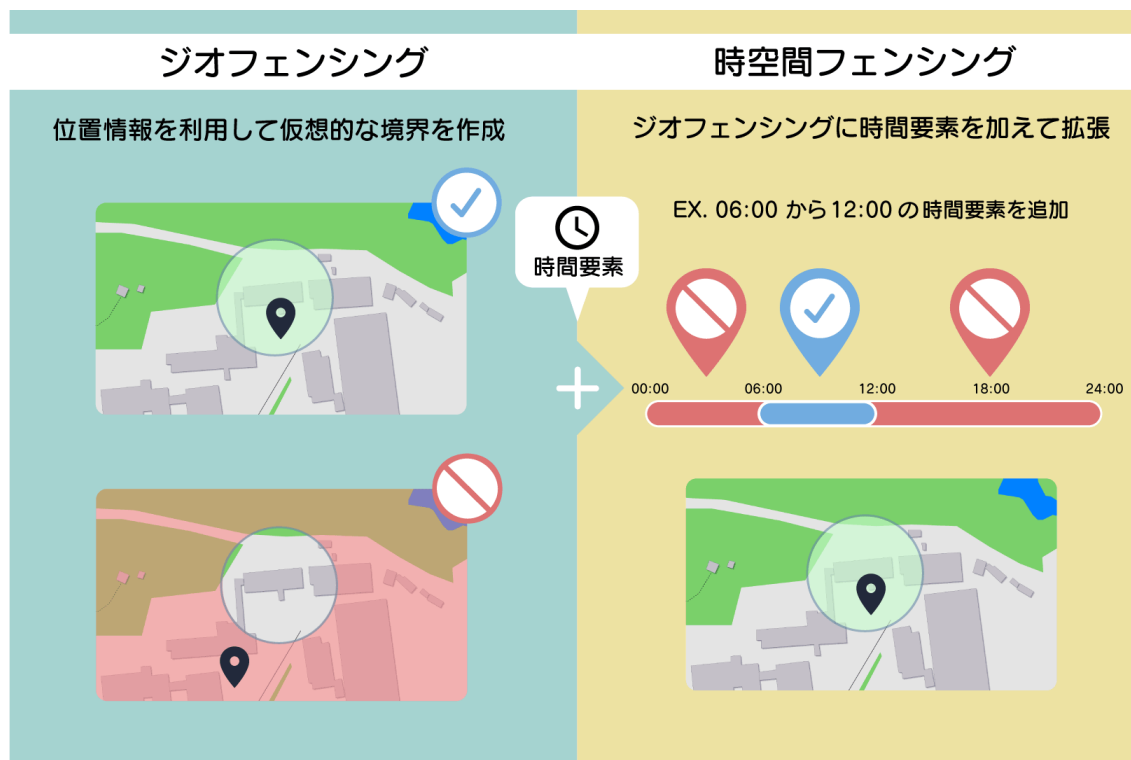


図 3.1: ジオフェンシングと時間要素を追加した時空間フェンシング

時空間フェンシングは「ジオフェンシングに時間要素を追加し拡張したフェンシング手法」として定義する。ジオフェンシングとは GPS や Wi-Fi, BLE ビーコンといった位置推定技術によって仮想的な境界を生成し、その境界に進入した、あるいは退出したときに特定のサービスを行うフェンシング手法である。位置推定の高精度化や手軽にジオフェンスを構築できる BLE のようなデバイスの普及に伴って、ジオフェンシングを用いた様々なアプリケーションが実現されている。つまり、時空間フェンシングとは、エリアと時間帯の指定によって仮想的な境界を生成し、その境界に進入した、あるいは退出したときに特定のサービスを行うフェンシング手法である。特定のサービスとは、本研究では「センシング」である。仮想的な境界内に進入したときにセンシングを開始し、境界外に退出したときにセンシングを終了する。

時空間フェンシングによって時間とエリアで境界を区切ると、依頼者は様々なシチュエーションを指定したクラウドセンシングが可能となる。様々なシチュエーションとは、例えば「午後3時から5時の公園」であったり、「3限の1号館401教室」や「昼間の食堂」などである。一方で、時空間フェンシングは時間やエリアに依存しないデータ収集には適さない。例えば、一日中センシングを行なったり、電車で移動したりなどである。このような長時間のセンシングやいつ終了するか定かではないセンシングは、協力者に消費電力など大きな負担をかけてしまう。また、エリアのみ及び時間帯のみを指定するデータ収集には適さない。エリアのみの場合、どこでセンシングされているかは分かっているが、いつセンシングされているかが明確ではない。時間帯のみの場合も同様である。依頼者と接点のない赤の他人が依頼者のセンシングに協力するとなった際、少しでもセンシング時間帯・エリアが曖昧であると、協力者は「協力しない」を選択する。エリアと時間帯両方の明確な設定により、赤の他人のセンシングへの協力判断がしやすくなる。そのため本プラットフォームでは、依頼者が想定し得るすべてのセンシングに対応している訳ではない。あくまでも時空間を指定したセンシングを想定している。

協力者のクラウドセンシングに対するプライバシー障壁は、時空間フェンシングによる時間とエリアの制限で軽減できる。時間とエリアの指定がない場合、協力者は「いつどこでセンシングが開始し、終了する

かわからない」という不安が生じるため、クラウドセンシングへの協力の判断がしにくい。時間とエリアの指定がある場合、協力者は自分自身がプライベートな活動をしているのか、ある程度他人にデータ提供しても良い活動をしているのかを区別・判断しやすくなるため、クラウドセンシングへの協力の判断がしやすくなる。時空間フェンシングにより、協力者は自分自身の可能な範囲内でセンシングに協力ができる。

## 3.2 ラヴラスの要求仕様

本節では、ラヴラスにおける非機能要件を提示する。3.2.1 項では、ラヴラスにおけるクラウドセンシングプラットフォームとしての設計基盤について述べる。3.2.2 項では、協力者のプライバシー保護について述べ、3.2.3 項では依頼者がラヴラスを利用する際の規約方針について述べる。3.2.4 項では、ラヴラスにおける利用者への配慮について述べる。

### 3.2.1 ラヴラスにおけるクラウドセンシングプラットフォームとしての設計基盤

クラウドセンシングプラットフォームを構築するためには、一定の制限・制約を設計基盤として設ける必要がある。この一定の制限・制約は、依頼者にとってクラウドセンシングの定義を簡略化でき、協力者にとっては受け入れやすいプラットフォームになるというメリットがある。また、クラウドセンシングの実施方法に決まった手法や手順はなく、依頼者によって多種多様であるため、制限や制約はプラットフォームとして成立させるために必要不可欠な要素である。関連研究として挙げた松田らの研究 [23] では、「ユーザ参加型モバイルセンシング基盤」という表現で我々のクラウドセンシングプラットフォームとしての設計基盤についての同様の記述がある。松田らの研究では、ゲーミフィケーションによって協力者の協力へのモチベーションを向上させ、クラウドセンシングを行うための「シナリオ」がモジュール化されて定義されている。それが我々のクラウドセンシングプラットフォームとしての設計基盤に相当する。松田らの研究における「シナリオ」は、新規モジュールを開発し新たな「シナリオ」を追加はできるものの、モジュール化されていない「シナリオ」でのセンシングは出来ないという制約がある。

我々の構築するクラウドセンシングプラットフォーム「ラヴラス」のセンシングプラットフォームとしての設計基盤は、時空間フェンシングである。我々が先ほど定義した時空間フェンシングを用いて、センシングを行う時間とエリアを制限するという制約を設けている。時空間フェンシングをクラウドセンシングプラットフォームとしての設計基盤とすると、依頼者としては主に時空間フェンシングの設定とセンシングの設定の2つを定義すればいいため、容易にクラウドセンシングの定義が可能となる。協力者としては、一定の時空間内でのみのセンシングであるため、プライバシーの観点からクラウドセンシングへの協力を受け入れやすいといえる。

ラヴラスで実現しないセンシングは、時空間フェンシングでは適さないセンシングである。例えば、一定の時間に依存しない1日の運動量といった長時間に渡るセンシングや時間に制限のない一定エリアでの活動のすべてといったセンシング、空間に依存しない車や電車等での移動中といったセンシングである。これらは時空間内でのみセンシングという制限により、ラヴラスでは行わない。また、時空間フェンシングの最大範囲を制限し、ラヴラスが定義する時間やエリアの範囲を超える時空間は定義できない。これはエリアが市や県単位であったり、時間が「0時から24時」であった場合、制限がないのと同じになってしまうからである。

### 3.2.2 プライバシーの保護

世界的なインターネットの一般化によって、インターネットサービスを利用する際の個人データの取り扱いが重要となっている。例えば、Facebook では個人情報の不正流出が問題となり、2018年4月10日にCEOであるマーク・ザッカーバーグ氏が謝罪を行っている [27]。このような背景から、EUは個人データの取り扱いに対して、2018年5月25日にデータ保護指令 (Data Protection Directive) に変わる

GDPR(General Data Protection Regulation：一般データ保護規則)[28] を施行した。GDPR は個人データやプライバシーの保護に関して、データ保護指令より厳格に規定し、制裁も強化している。

こういった国際的動向から、ラヴラスにおいてもプライバシーの保護は必須である。まず、設計基盤である時空間フェンシングという制限によって、時間やエリアの指定がないセンシングを実施しない。また、時間やエリアを指定されている場合でも、依頼者の承諾なしでのセンシングは行わない。これらにより、協力者の意図しないセンサのログデータ（以下、センシングデータ）の提供に伴うプライバシーの侵害というリスクを最小限に抑える。加えて、各協力者から提供されるセンシングデータやそれに結びつく端末情報などのセンシティブな情報を取り扱うため、依頼者のユーザ認証により、第三者による意図しないアクセスや情報の改ざんなどを防ぐ。

GDPR の個人データの定義（第 4 条）によると、協力者から提供される各センシングデータそのものは個人データには分類されないとされている。しかし、そのセンシングデータに付随される情報との組み合わせによって、個人データに分類される可能性がある。そのため、ラヴラスにはセンシングデータが個人情報となりえない設計が必要である。例えば、協力者からのセンシングデータ提供の際、センシングを行った環境情報として端末名やセンサの型番などの端末情報（以下、メタデータ）を取得するが、端末識別 ID である IMEI(International Mobile Equipment Identity) は個人データとなるため取得しない。加えて、メタデータには IMEI 以外に、センシングデータとメタデータの参照により個人が識別可能になる場合、メタデータはセンシングを行った環境が判明する情報のみ取得するものとする。

さらに、協力者が特定のセンシングを拒否できる設計や提供したデータの削除要請を行う機能が必要がある。上記で述べたように協力者から提供されるセンシングデータ及びメタデータは、個人情報に繋がらない特性情報のみに留めている。しかし、協力者がプライバシーを侵害されていると感じた場合、協力者の都合により特定のセンシングの拒否及び提供済みのセンシングデータ削除の要請などの要求に対応しなければならない。依頼者側は協力者の協力拒否や削除要請の拒絶はできない。これは、クラウドセンシングとは協力者ありきのデータ収集方法であり、協力者の信頼をなくしてしまっては成り立たないからである。もし拒絶ができた場合、クラウドセンシングに対する信頼はなくなってしまう、協力者は該当のセンシングだけではなく、本プラットフォームの利用をやめてしまう可能性がある。

### 3.2.3 ラヴラスの利用規約方針

依頼者はラヴラスを利用してクラウドセンシングを行う際、センシングの依頼（以下、センシングプロジェクト）を作成する必要がある。センシングプロジェクトの作成を行うためには、依頼者の利用資格とそのセンシングプロジェクトを立ち上げる目的や使用センサ等のセンシングプロジェクトの内容の明示が必要である。依頼者の利用資格を得るには、依頼者自身の情報が明示されている必要があり、匿名によるセンシングプロジェクトの作成はできない。この依頼者情報は、基本的に依頼者本人の責任で保証されるものとし、ラヴラスでは登録されるメールアドレスが大学や所属機関のものかどうかなどの簡易的な確認のみを行うものとする。センシングプロジェクトの内容の明示では、そのセンシングがどんな目的で行われるのか、センサはどんなものを使用するのか、実施期間や時空間フェンシングといった依頼内容を定義したものを明示する。これは、どんな依頼者がどんな目的でデータ収集したいのかを明確にし、その実施内容が妥当なものであるのかを協力者が判断ができるようにするために必要である。該当のセンシングプロジェクトの協力者にこれらの情報を明示し、協力者はその内容を了承した上でセンシングプロジェクトに協力する。

センシングデータの取り扱いの禁止事項として、「センシングデータの再利用禁止」と「保存期間以上のセンシングデータの保存禁止」を制定する。センシングデータの再利用禁止は、「センシングデータの利用は依頼者本人（組織の場合はその組織）のみであるものとし、依頼者は収集したセンシングデータをそのプロジェクトに明示した目的を超える利用や第三者への開示・譲渡等は禁止する」と示す。これは、協力者が依頼者のセンシングプロジェクトに同意した上で提供したセンシングデータが、協力者の意図と反して利用されてしまうからである。ラヴラスが指定する保存期間以上のセンシングデータの保存の禁止

は、「ラヴラスが定めるセンシングデータの保存期間を超える長期間の保存は禁止とする」と示す。これは、センシングデータの提供方法をダウンロードとしているため、3.2.2 項で述べた協力者による提供済みのセンシングデータの削除要請によって、削除されたセンシングデータが適切に利用できなくなるようにするためである。センシングデータの提供方法をダウンロードとしているのは、依頼者が独自の環境で柔軟にセンシングデータの解析が行えるようにサポートするためである。協力者による提供済みのセンシングデータの削除要請が行われたとき、直ちにサーバに保存された該当のセンシングデータの削除が行われるものとする。しかし、削除要請以前に該当のセンシングデータが依頼者によってダウンロードされ、保存期間の終了前であった場合は、依頼者に該当のセンシングデータが削除要請によってサーバから削除されたと通知し、直ちに該当のセンシングデータの利用中止・削除を要請する。このとき協力者の削除要請から依頼者が利用できなくなるまでにラグの発生が予想されるため、協力者に利用規約として提供済みのセンシングデータの削除要請が直ちに実行されるものではないと事前に通知する。

### 3.2.4 利用者への配慮

ラヴラスの利用過程において負担となる要素がある場合、利用者は獲得しにくくなる。そのため、利用者の視点から快適に利用できると思える配慮が必要である。こうした配慮は、依頼者・協力者共に受け入れやすいプラットフォームを実現するために必要であり、依頼者と協力者が相互に利用しやすいプラットフォームを目指す上で重要となる。本節では、心理的負担と物理的負担の2つの視点に分けて、その負担と配慮について述べる。

心理的負担としては、協力者がクラウドセンシングに協力する際に抱くプライバシーを侵害されているという不安が挙げられる。これは、協力者にとってスマートフォンから意図しないセンシングデータが提供されるかもしれないという不安ともいえる。これに対する配慮として、安心してクラウドセンシングに協力できる環境を提案する必要がある。まず、3.2.1 項で挙げた時空間フェンシングによる時間とエリアによる制限がある。協力者には、センシングプロジェクトに定義された時空間に進入した際にセンシングプロジェクトに協力するか否かを問う通知を送る。協力者は、依頼内容や設定された時空間を確認した上で協力するかを判断し承諾された場合のみセンシングを行い、拒否した場合はセンシングは行わない。このプロセスによって、協力者は意図したセンシングプロジェクトのみにセンシングデータの提供が可能である。さらに、3.2.2 項に挙げたように状況に応じてこのプロセスを通して承諾済みのセンシングプロジェクトの承諾取り消しや提供済みセンシングデータの削除が可能のため、誤って許可してしまったセンシングプロジェクトへの拒否や意図せず提供してしまったセンシングデータを削除もできる。

物理的な負担としては、1回のセンシング協力に伴う協力者の手間が挙げられる。1回のセンシング協力にかかる手間が大きいほど、協力者のセンシング協力に対するモチベーションは低下し、継続的なセンシング協力は得られにくい。これに対する配慮として、協力者の手間となるスマホアプリの操作は、基本的に上記に記した通知画面よりセンシングプロジェクトへの承諾及び拒否を行うのみで行えるように設計する。この操作は協力するセンシングプロジェクトごとに初めの1回のみであるため、再度同じセンシングプロジェクトに設定された時空間に進入した場合は一切操作は行う必要はなく、自動的にセンシングを開始する。また、センシングの終了や終了後のセンシングデータのアップロードは自動で行われるため、協力者がスマホアプリを操作する必要はない。さらに、センシングデータのアップロードは Wi-Fi 環境下のみで行うため、協力者のパケット通信量を最小限に抑える。この他に、依頼者・協力者に共通してラヴラスを利用する際に必要不可欠なユーザインターフェイスについても配慮を行う。ユーザインターフェイスは、利用者によって使いやすさに直結するものである。情報の表示や入力のしやすさが低ければ、利用者によって物理的負担となり利用されにくいプラットフォームとなるためである。

### 3.3 ラヴラスの実装

先述のプラットフォームの要求仕様に則って、クラウドセンシングプラットフォーム「ラヴラス」の実装を行う。本プラットフォームは、センシングデータの管理や各プロジェクトを管理するサーバ、依頼者専用のプロジェクトの作成・管理を行う Web アプリ、協力者専用のセンシングを行うスマホアプリによって構成する。3.3.1 項では、依頼者と協力者がラヴラスを操作する際の手順について述べる。3.3.2 項では、センシングデータの管理や各プロジェクトを管理するサーバの実装について述べる。3.3.3 項では、依頼者専用のセンシングプロジェクト管理やセンシングデータダウンロードを行う Web アプリの実装について述べる。3.3.4 項では、協力者専用のセンシングスマホアプリの実装について述べる。

#### 3.3.1 ラヴラスの操作手順

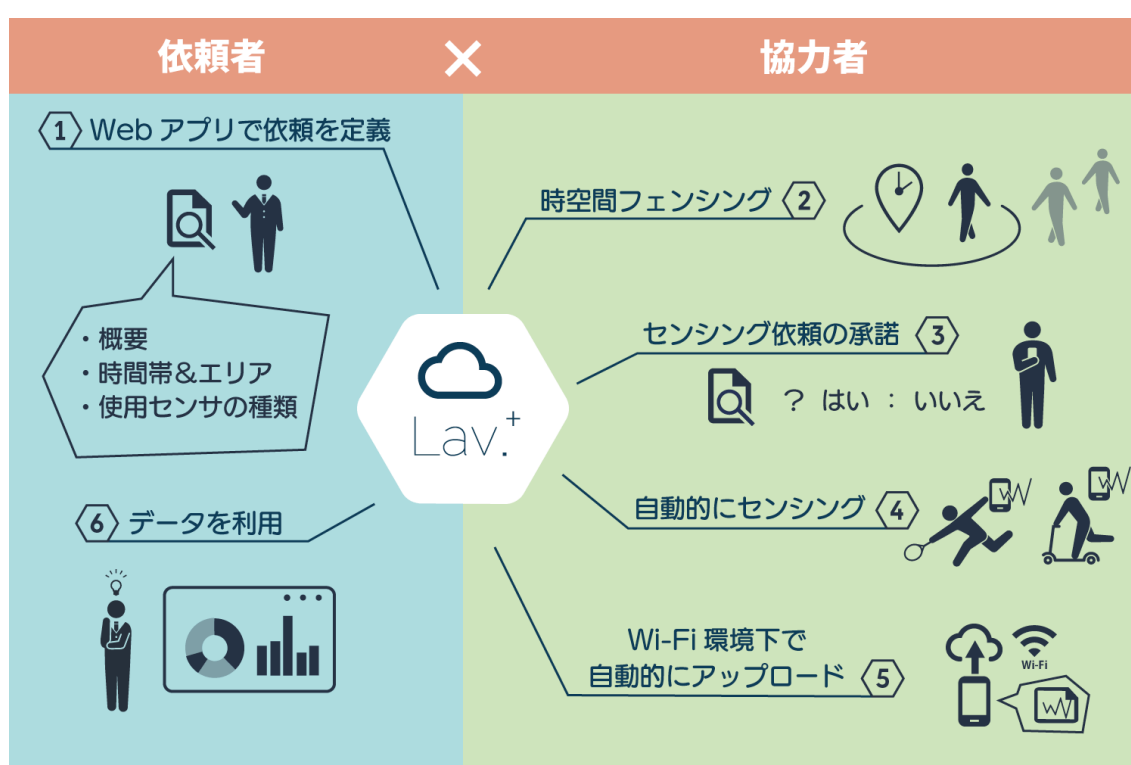


図 3.2: ラヴラスによるクラウドセンシングの操作手順

ラヴラスの一連の流れは「Web アプリでセンシングプロジェクトの定義」、「時空間フェンシング」、「センシング依頼の承諾」、「自動的にセンシング」、「Wi-Fi 環境下で自動的にアップロード」、「データ利用」の順で行う (図 3.2)。依頼者側では「Web アプリでセンシングプロジェクトの定義」と「データ利用」が行われる。一方で、協力者側では「時空間フェンシング」、「センシング依頼の承諾」、「自動的にセンシング」、「Wi-Fi 環境下で自動的にアップロード」が行われる。ラヴラスを利用してクラウドセンシングを行う際のユーザの使用デバイスとして、依頼者は PC を、協力者はスマートフォンを前提としている。依頼者がラヴラスを利用するためには、まず Web アプリでユーザ登録してもらう。協力者がラヴラスのクラウドセンシングに協力するためには、ラヴラス専用のスマホアプリをインストールしてもらう。

依頼者はプロジェクト管理 Web アプリにて、センシング依頼の内容を細かく定義し、センシングプロジェクトを作成する。センシングプロジェクトとはセンシングを行いたい時間帯やエリア、センシングを行う目的、概要、センシングに用いるセンサの種類やそのセンサのサンプリングレートなどである。これは協力者側のスマホアプリでセンシングや時空間フェンシングの設定が必要であるとともに、協力者に

センシングプロジェクトを提示し、協力するか否かの判断材料にしてもらう。作成されたセンシングプロジェクトはスマホアプリ側が適宜取得し、取得したプロジェクトに応じて時空間フェンシング・センシング依頼通知・センシングを行う。

スマホアプリ側ではセンシングプロジェクトに応じて、3.1 節の定義をもとに時空間フェンシングを行い、協力者が指定された時間帯かつエリアにいる場合のみ通知が送られる。そのため、時間帯・エリアのどちらか一方のみ当てはまる場合は通知は送られない。例えば、指定エリア内に進入していたとしても、指定時間帯に入っていない及び過ぎている場合は通知は送られない。指定時間帯に入っていて、指定エリア内に進入していない場合も同様である。また、時空間フェンシングはバックグラウンドで行うため、本スマホアプリを開いておく必要はない。そして、指定された開始時間から終了時間の間であれば、いつ指定されたエリアに進入しても通知は送られる。指定されたエリアを退出し、一定時間後再び進入した場合は、通知を送らずにセンシングを開始する。



図 3.3: センシング依頼のヘッドアップ通知と通知画面

協力者はヘッドアップ通知のクリックで本アプリケーションの通知画面を開き、依頼されているセンシングプロジェクトの内容を確認し、承諾・拒否の判断をする。通知画面で表示する情報としては、センシングを行うエリア、時間帯、センシングの目的、概要、依頼者の情報である。また、時空間内に進入した際の現在地と指定エリアを Google Maps で位置関係をより分かりやすく表示している。センシング依頼の内容に納得し、協力してもいい場合に承諾ボタン (通知画面では丸ボタン) を押す。そうすると、センシングが開始される。通知が届いた時間帯やエリアが協力者にとってセンシングされたくない時間またはエリアの場合、または届いたセンシング依頼に協力したくない場合は、拒否ボタン (通知画面ではバツボタン) を押す。そうすると、センシングは開始せずに終了する。ラヴラスのクラウドセンシングの手順において協力者がスマートフォンを操作するのは主にこの通知から承諾・拒否の部分である。承諾ボタンを押したら、もうスマホアプリは閉じて構わない。

センシングも常にバックグラウンドで行うため、アプリケーションを開いておく必要はない。バックグラウンドでのセンシングにより、協力者がセンシング終了などの操作をする必要がない。センシング終了後は Wi-Fi 接続時にのみ自動でサーバ側にアップロードされる。協力者のスマートフォンが Wi-Fi などに接続されている場合にセンシングデータをサーバにアップロードする。接続されていない場合は、アップロードせずディレクトリに格納しておき、Wi-Fi などに接続された際にまとめてアップロードする。依頼者は本 Web アプリにて、アップロードされたセンシングデータを閲覧可能である。



### 3.3.2 センシングデータの管理や各プロジェクトを管理するサーバの実装

センシングデータの管理や各プロジェクトを管理するサーバは、Web アプリ及びスマホアプリのどちらからの連携もスムーズ行えるようにそのどちらにも親和性が高い JSON ベースの REST API として設計した。REST API としてサーバを設計すると、MVC モデルにおける Model と Controller を View 部分にあたる Web アプリとスマホアプリを完全に分離し、実装の分業化及びオブジェクト指向設計の原則における単一責任の原則によって、各機能のカプセル化を実現し、メンテナンスや拡張実装をより柔軟にするという狙いもある。サーバは、構築する上で JSON 形式のファイルを取り扱いやすい言語を選択する必要があったため、JavaScript(Node.js) を選択した。また、JavaScript は JSON ファイルをよりネイティブに取り扱いやすい言語であり、REST API 専用のフレームワークの選択肢が他言語に比べて多く存在している。REST API のフレームワークは、Web サーバ系フレームワークとして有名な Express を、REST API 専用のフレームワークとして再設計された LoopBack4 を採用した。サーバで使用するデータベースは、今後のスマートフォンのさらなる高性能化による搭載センサの種類増加に伴うデータベースの変更やシステムの拡張性の向上を図るため、変更や操作が柔軟に可能なデータベースとして NoSQL データベースの MongoDB を採用した。

サーバのデータベースと REST API のエンドポイントを 3.2 節「ラヴラスの要求仕様」を基に設計を行った。NoSQL データベースである MongoDB にはリレーショナルデータベースにおけるテーブルの列の概念が存在しないため、各モデルの定義を LoopBack4 のモデルによって定義する。以下の各モデルの説明には、LoopBack4 で定義したモデルによって作成したコレクション (リレーショナルデータベースにおけるテーブルに相当) とそのコレクション内のドキュメント (リレーショナルデータベースにおけるレコードに相当) の一例を示すものとする。

```
{
  "_id": {
    "$oid": "5f9fcded7d55602698ccd9a9"
  },
  "username": "miyagawa",
  "realm": "宮川信人",
  "gender": "man",
  "birth": {
    "$date": "1998-06-13T00:00:00.000Z"
  },
  "introduction": "梶研究室B4",
  "icon": "https://○○○○○○○○",
  "url": "https://○○○○○○○○",
  "email": "k17125kk@aitech.ac.jp",
  "belongingTo": "愛知工業大学"
}
```

図 3.4: User コレクションのドキュメント一例

各依頼者を管理するためのモデルとして User モデル (図 3.4) を定義した。User モデルには、ドキュメント管理用の「id」、ユーザ名「username」、依頼者の本名「realm」、性別「gender」、生年月日「birth」、紹介文「introduction」、アイコン画像「icon」、e メールアドレス「email」、Web ページリンク「url」、所属機関「belongingTo」の 10 項目を定義した。要求仕様の 3.2.3 項「ラヴラスの利用及び利用規約の方針」に基づき、依頼者の利用資格の観点から匿名によるセンシングプロジェクトの作成はできないため、ユーザ名、依頼者の本名、性別、生年月日、e メールアドレス、所属機関を必須項目として設定を行った。REST API のエンドポイントは、ユーザ登録を行う「/signup(POST)」、ログイン用の「/users/login(POST)」、ログイン後自身の情報を取得する「/whoAmI(GET)」を作成した。

```

{
  "_id": {
    "$oid": "5fa3943872d4a2aae0096e14"
  },
  "title": "梶研究室の活動量調査",
  "image": "https://○○○○○○○○",
  "overview": "試験運用のため、梶研究室（4号館別館）での活動量を調査します",
  "purpose": "加速度センサを用いた、活動量調査を試験運用のため行います",
  "startDate": {
    "$date": "2020-11-03T00:00:00.000Z"
  },
  "endDate": {
    "$date": "2021-02-01T00:00:00.000Z"
  },
  "userId": {
    "$oid": "5f9fcdded7d55602698ccd9a9"
  }
}

```

図 3.5: Project コレクションのドキュメント一例

各センシングプロジェクトを管理するためのモデルとして Project モデル (図 3.5) を定義した。Project モデルには、ドキュメント管理用の「id」、プロジェクトタイトル「title」、サムネイル用画像「image」、概要説明「overview」、目的「purpose」、センシングプロジェクトの開始日「startDate」と終了日「endDate」、作成した User の id「userId」の 8 項目を定義した。REST API のエンドポイントは「/project」でそれぞれの操作に対応する HTTP メソッドにより CRUD 操作 (Create, Read, Update, Delete) が可能となる。これらの項目のうちサムネイル用画像以外の項目はすべて必須項目であり、要求仕様の 3.2.3 項「ラヴラスの利用及び利用規約の方針」に基づき、センシングプロジェクトの内容を明確にするために概要や目的の説明文の項目を定義している。また、同要求仕様を示されたセンシングに関するセンサの定義や時空間フェンシングの定義は、SensingSetting モデルで定義を行っている。センシングプロジェクト作成に必要な情報を Project モデルと SensingSetting モデルに分離するのは、スマホアプリが SensingSetting モデルのエンドポイントから各センシングプロジェクトのセンシングに関する設定を直接的な読み込みを可能にするためである。

```

{
  "_id": {
    "$oid": "5f9fd88fe78ded8b9cbb51bd"
  },
  "isProvidedProfile": false,
  "sensors": {
    "accelerometer": 100,
    "gyroscope": 100
  },
  "areaName": "4号館別館",
  "geo": {
    "type": "FeatureCollection",
    "features": [{
      "type": "Feature",
      "properties": {},
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [137.1152805536985, 35.18513377716786],
            [137.11568757891655, 35.18515241047897],
            [137.11567416787148, 35.18526804652008],
            [137.1152725070715, 35.18525379753812],
            [137.1152805536985, 35.18513377716786]
          ]
        ]
      }
    }]
  },
  "periodOfTimes": [{
    "interval": {
      "length": 1,
      "entity": "week",
      "dayOfWeek": ["tue", "fri"]
    },
    "period": {
      "from": "09:00:00",
      "to": "15:00:00"
    }
  }],
  "projectId": {
    "$oid": "5f9fcf377d55602698ccd9ab"
  }
}

```

図 3.6: SensingSetting コレクションのドキュメント一例

SensingSetting モデル (図 3.6) には、ドキュメント管理用の「id」、協力者に自身の基礎情報 (身長・体重・性別など) を求めるか否かの設定「isProvidedProfile」、使用センサの種類とそれぞれのセンサのサンプリングレート (Hz)「sensors」、時空間フェンシングのエリア名「areaName」、GeoJSON による時空間フェンシングのエリア設定「geo」、時空間フェンシングの時間帯設定「periodOfTimes」、紐づくセンシングプロジェクトの id「projectId」の 7 項目を定義した。REST API のエンドポイントは「/sensing-setting」でそれぞれの操作に対応する HTTP メソッドにより CRUD 操作が可能となる。協力者に自身の基礎情報を求めるか否かの設定「isProvidedProfile」は、依頼者がセンシング協力を行う協力者の体格や性別に応じた変化を含めて調査を行いたい場合に利用するための項目として定義した。これは、協力者に基礎情報の登録をスマホアプリで行ってもらい、その情報をセンシングデータと共に送信するものであるが、現在実装には至っていないため定義のみとする。時空間フェンシングの定義として、柔軟に依頼者の想定するシチュエーションに対応するためには、複数のエリアや異なる時間帯を定義できるフォーマットにする必要がある。時空間フェンシングのエリア設定「geo」は、現在のエリア判定方法が GPS 情報による判

定でのみであるため、地理的データを表現するフォーマットである GeoJSON 形式を用いてエリアの設定を行っている。エリアの定義に GeoJSON 形式を用いた理由は、2 つ以上のエリアの定義を可能にできる他、複雑なエリアの指定も可能にするためである。

```
"periodOfTimes": [{
  "interval": {
    "length": 1,
    "entity": "day"
  },
  "period": {
    "from": "16:00:00",
    "to": "18:00:00"
  }
}]
```

毎日の 16:00 から 18:00

```
"periodOfTimes": [{
  "interval": {
    "length": 2,
    "entity": "week",
    "dayOfWeek": ["tue", "fri"]
  },
  "period": {
    "from": "10:40:00",
    "to": "12:10:00"
  }
}]
```

隔週の火曜日と金曜日の 10:40 から 12:10

```
"periodOfTimes": [{
  "interval": {
    "length": 1,
    "entity": "day"
  },
  "period": {
    "from": "16:00:00",
    "to": "18:00:00"
  }
}, {
  "interval": {
    "length": 2,
    "entity": "week",
    "dayOfWeek": ["tue", "fri"]
  },
  "period": {
    "from": "10:40:00",
    "to": "12:10:00"
  }
}]
```

毎日の 16:00 から 18:00

と

隔週の火曜日と金曜日の 10:40 から 12:10

図 3.7: periodOfTimes の定義の例

時空間フェンシングの時間設定「periodOfTimes」も、複数の時間帯を定義可能なフォーマットを定義した (図 3.7)。このフォーマットは、interval と period からなるオブジェクトの配列として定義する。interval は、指定された時間帯を繰り返す間隔を定義するプロパティで length, entity, dayOfWeek からなるオブジェクトである。entity と dayOfWeek は間隔を設ける単位を表現し、entity には「day」もしくは「week」のどちらかのテキストが入り、「day」の場合は dayOfWeek プロパティが省略され、「week」の場合は dayOfWeek プロパティに指定する曜日を 3 文字の曜日の省略表記を用いて文字列の配列として定義する。length には、数字を指定し entity と dayOfWeek で指定した単位で間隔を設ける。period は、from プロパティにセンシング開始時間、to プロパティにセンシング終了時間を指定する。このフォーマットによって例えば、「毎日の 16:00 から 18:00 まで」や「隔週の火曜日と金曜日の 10:40 から 12:10」といった指定やその両方の指定が可能になる。

```

{
  "id": "3ff4fac4-9b95-43c6-8702-a197d8453da2",
  "filename": "3ff4fac4-9b95-43c6-8702-a197d8453da2.png",
  "originalname": "5f9fcf377d55602698ccd9ab_20210126-210844.log",
  "fileSize": 7340374,
  "uploadDate": "2021-01-31T11:32:50.794Z",
  "meta": {
    "sdkVersion": "29",
    "deviceModel": "Pixel4",
    "sensors": [
      {
        "Accelerometer": "LSM6DSR",
        "Gyroscope": "LSM6DSR"
      }
    ]
  },
  "projectId": "5f9fcf377d55602698ccd9ab"
}

```

図 3.8: SensingData コレクションのドキュメント一例

協力者によって提供される各センシングデータを管理するモデルとして SensingData モデル (図 3.8) を定義した。SensingData モデルには、ドキュメント管理用の「id」、サーバ内部で保存したファイル名「filename」、協力者から提供されたオリジナルのファイル名「originalname」、ファイルのサイズ (byte)「fileSize」、アップロードが行われた日時「uploadDate」、メタデータ「meta」、提供されたセンシングプロジェクトの id「projectId」の 7 項目を定義した。REST API のエンドポイントは「/sensing-data」でそれぞれの操作に対応する HTTP メソッドにより CRUD 操作が可能となる。オリジナルのファイル名を変更しサーバで保存を行っているのは、サーバ内で同名のファイルが保存され上書きされる可能性があるためである。また、メタデータは要求仕様の 3.2.2 項「プライバシーの保護」に基づき、個人データに該当しない OS の SDK バージョン、デバイスモデル、使用したセンサとその型番のみを取得する。センシングデータをアップロードするための専用のエンドポイントとして、「/sensing-data/upload(POST)」を作成する。このエンドポイントは、バイナリデータであるセンシングデータを取り扱うために JSON 形式によるエンドポイントではなく、「multipart/form-data」を使用し、スマホアプリがセンシングデータのアップロードと「/sensing-data」を通してセンシングデータの情報のアップロードを行う手間を削減するために、センシングデータがアップロードされるとサーバは自動的に SensingData コレクションにドキュメントを作成する。

```

{
  "_id": {
    "$oid": "5f9fcdded7d55602698ccd9aa"
  },
  "password": "$2a$10$gL5SPoDh0WEh7kYDf1o0a.sN3HoeVjLKhod1tby3vB4dP0CytAf1G",
  "userId": {
    "$oid": "5f9fcdded7d55602698ccd9a9"
  }
}

```

図 3.9: UserCredentials コレクションのドキュメント一例

最後に、要求仕様の 3.2.2 項「プライバシーの保護」に対応するためのセキュリティ対策として JWT トークン認証を導入した。JWT トークン認証を実現するために、User モデルに結びつき認証情報を保持する UserCredentials モデル (図 3.9) を定義した。UserCredentials モデルには、ドキュメント管理用の「id」、依頼者が設定したパスワードのハッシュ値「password」、結びつく User モデルの id「userId」の 3 項目で定義される。セキュリティ対策として依頼者の設定したパスワードは、そのままの値で保持せずハッシュ値を用いている。この JWT トークン認証によって各エンドポイントの不正なアクセスの防止を実現する。

例えば、提供済みのセンシングデータのダウンロードを行うエンドポイントは、センシングプロジェクトの作成を行った依頼者本人でなければアクセスができない。図 3.19 は、権限の無いユーザがセンシングデータのダウンロードを行うためのエンドポイントにアクセスした際に表示されるエラーメッセージである。

```
{
  "error": {
    "statusCode": 401,
    "name": "UnauthorizedError",
    "message": "Error verifying token : invalid signature"
  }
}
```

図 3.10: 権限の無いユーザがセンシングデータのダウンロードを試みる例

### 3.3.3 依頼者用のプロジェクト管理 Web アプリの実装

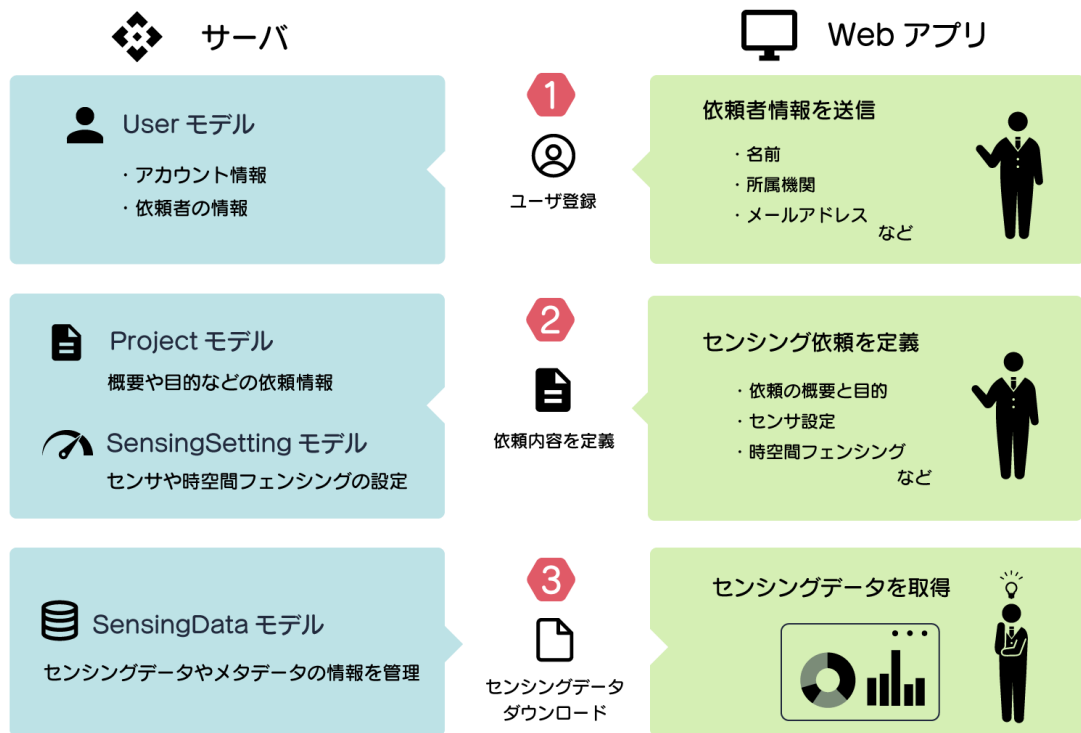


図 3.11: サーバと Web アプリの連携

依頼者のユーザ登録やセンシングプロジェクトの作成及び管理を行うための Web アプリには、Single Page Application(以下、SPA)を採用し実装を行った。SPA は、その名前にシングルページとあるように単一の HTML を JavaScript を用いた動的な変更により、画面を描画する Web アプリのアーキテクチャである。SPA を用いると JavaScript によってページを制御できるため、デスクトップアプリケーションのような操作を実現できる他、REST API との親和性も高くサーバとの連携に柔軟に対応できる。依頼者用のインターフェイスとして、Web アプリを採用する。これにより環境依存が少なく、SPA によってデスクトップアプリケーションに近い操作感を得られるため、要求仕様の 3.2.4 項「利用者への配慮」のユーザインターフェイスへの配慮を実現している。また、SPA のフレームワークにはコンポーネントベ-



スで SPA を作成できる React を採用した。コンポーネントベースとは、ヘッダやボタンといったパーツをコンポーネントという単位で作成し、それらを組み合わせによって SPA を作成する設計思想である。コンポーネントベースの React の採用により、作成したコンポーネントの再利用や修正・管理がしやすくなる。また、コンポーネント単位ですべてのパーツを管理できるため、拡張性やメンテナンス性の向上、開発工数の削減が期待できる。

The image shows a user registration form for a platform called 'Lav.+ クラウドセンシングプラットフォーム'. The form is displayed on a dark grey background with a blue geometric pattern on the left. It contains several input fields: 'ユーザー名' (Username), '名前 (実名)' (Real Name), '性別' (Gender) with radio buttons for '男' (Male), '女' (Female), and 'その他' (Other), '生年月日' (Date of Birth), 'メールアドレス' (Email Address), and '所属機関' (Affiliation). A blue '登録' (Register) button is at the bottom right.

図 3.12: ユーザ登録画面

Web アプリでは、まず依頼者がユーザ登録を行う。ユーザ登録では、サーバの User モデルに合わせたフォームを作成し要求仕様の 3.2.3 項「ラヴラスの利用及び利用規約の方針」に基づき本名、性別、生年月日、e メールアドレス、所属機関の 5 項目を必須項目とする。入力が完了し送信ボタンのタップにより、サーバの「/signup」エンドポイントに入力情報が送信され登録完了となる。次に、新規センシングプロジェクトを作成する。センシングプロジェクトの入力フォームは入力項目が多いため、要求仕様の 3.2.4 項「利用者への配慮」のユーザインターフェイスの配慮に対応し、「基本設定」「概要と目的」「有効期限」「センサ設定」「時空間フェンシング設定」の項目の 5 ページに分け、さらに入力の複雑な項目に対して入力操作を簡易化する仕組みで入力操作の向上を図っている。各ページの右上に前のページに戻るボタン、入力フォームの下部に次の入力項目に遷移するためのボタンを設置する。これらのページは、遷移を行っても内部的に入力されたフォームの情報を保持しており、入力を前のページに戻り修正が可能である。



図 3.13: 基本設定ページの入力画面と入力例

基本設定ページ (図 3.13) では、サムネイル用画像とセンシングプロジェクトの名称の入力を行う。この入力ページでは、プロジェクトの名称のみが必須項目であるため、サムネイル用画像は任意とする。この2つの項目は、サーバに定義された Project モデルの「title」と「image」に対応している。

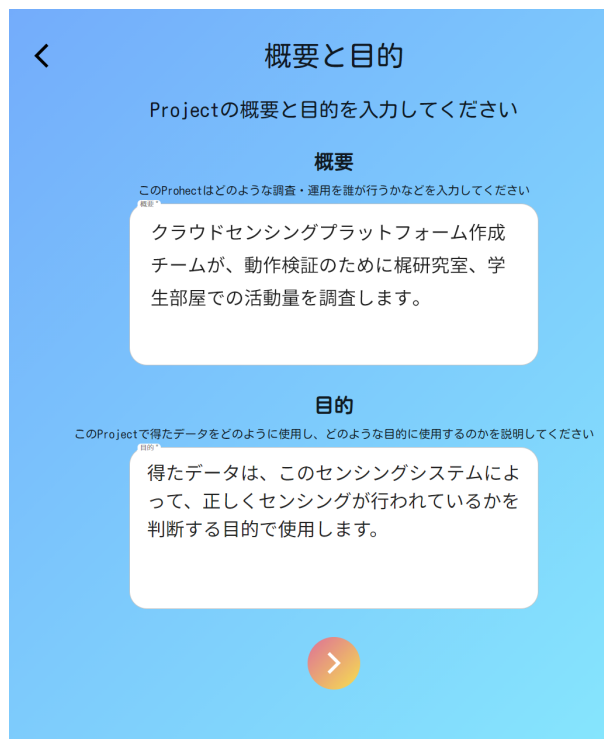


図 3.14: 概要と目的ページの入力画面と入力例

概要と目的ページ (図 3.14) では、概要と目的の入力を行う。この入力ページでは、どちらの入力項目も必須項目であり、最低文字数 20 文字以上の入力が必要となっている。この2つの項目は、サーバに定義された Project モデルの「overview」と「purpose」に対応している。



Projectの有効期限

このProjectの有効期限を設定してください

開始日

開始日  
2020/10/27

終了日

終了日  
2020/11/27

Calendar popup for November 2020:

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

図 3.15: 有効期限ページの入力画面と入力例

有効期限ページ (図 3.15) では、センシング開始日とセンシング終了日入力を行う。この入力ページでは、どちらの入力項目も必須項目である。入力フォームは、日付の入力を簡易化するためにカレンダーのポップアップからの入力を可能にする。この2つの項目は、サーバに定義された Project モデルの「startDate」と「endDate」に対応している。

センサの設定

必要なセンサを有効にして  
サンプルングレートを設定してください

プロフィール取得 (身長・体重・年齢など) を有効にしますか?  
☒ プロフィール取得

センサの種類	サンプルングレート (Hz)
加速度センサ	94
ジャイロセンサ	100
磁気センサ	100
気圧センサ	100
照度センサ	100
接近センサ	95
Wi-Fi	105
BLE	100

図 3.16: センサ設定ページの入力画面と入力例

センサ設定ページ (図 3.16) では、協力者に自身の基礎情報を求めるかの有無とセンサの設定を行う。協力者に協力者の基礎情報を求める設定は、3.3.2 項「センシングデータの管理や各プロジェクトを管理するサーバの実装」にも記述があるが、現在実装には至っていないため定義のみとなる。この入力ページでは、一つ以上のセンサの設定が必須である。このページのフォームの入力方法はトグルスイッチを採用しており、トグルスイッチにより有効になった設定と無効な設定を判別しやすいようにそれぞれの項目が明るくなるようになっている。また、各センサの設定は文字だけでなくアイコンを用いてそれぞれのセンサの設定を視覚的に判別しやすいようにしており、トグルスイッチによりサンプルングレートの入力が有効化されサンプルングレートの設定を行う。この 2 つの項目は、サーバに定義された SensingSetting モデルの「isProvidedProfile」と「sensors」に対応している。

←

時空間フェンジングの設定

時空間フェンジングの設定

エリア名\*

梶研究室 [学生部屋]

geoJson \*

```
{ "type": "FeatureCollection", "features":
[{"type": "Feature", "properties": {}, "geometry":
{"type": "Polygon", "coordinates":
```

```
  {
    "type": string "FeatureCollection"
    "features": [ 1 item
      0: { 3 items
        "type": string "Feature"
        "properties": {} 0 items
        "geometry": { 2 items
          "type": string "Polygon"
          "coordinates": [ 1 item
            0: [ 5 items
              0: [ 2 items
                0: float 137.1152798831463
                1: float 35.18513651736094
              ]
              1: [ 2 items
                0: float 137.11568355560303
                1: float 35.18515241047899
              ]
              2: [ 2 items
                0: float 137.11567349731922
                1: float 35.18526914259551
              ]
              3: [ 2 items
                0: float 137.11527653038502
                1: float 35.18525215342467
              ]
              4: [ 2 items
                0: float 137.1152798831463
                1: float 35.18513651736094
              ]
            ]
          ]
        ]
      ]
    ]
  }
}
```

図 3.17: 時空間フェンジング設定ページのエリア入力画面と入力例

時空間フェンジング設定ページ (図 3.17) では、時空間フェンジングに必要なエリアの設定と時間帯の設定を行う。この入力ページでは、エリア名の入力とエリア設定、時間帯を 1 つ以上の設定が必須である。エリアの入力では、GeoJSON 形式のテキストの入力を行う。この入力フォームは、[geojson.io](http://geojson.io)[29] 等のサイトを用いて GeoJSON 形式のテキストを取得・入力を前提としている。図 3.18 は、[geojson.io](http://geojson.io) を利用して GeoJSON 形式のテキストを取得する例である。

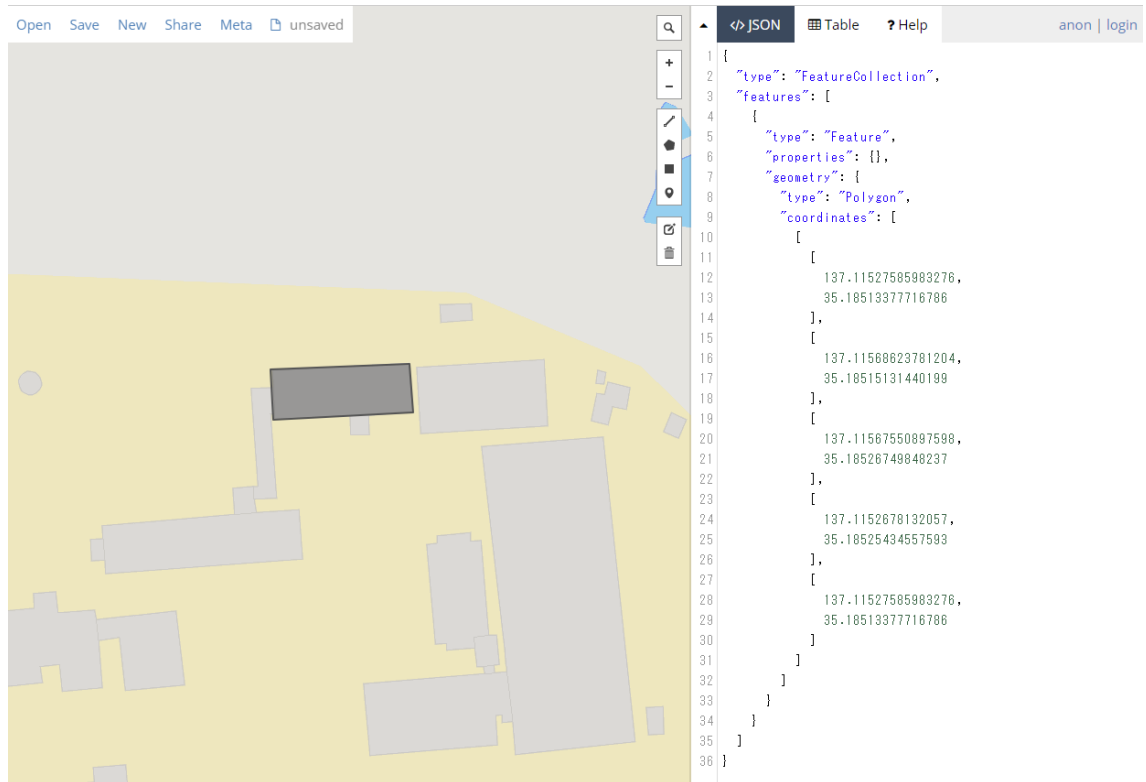


図 3.18: geojson.io を利用した GeoJSON 形式のテキストの取得と入力例



図 3.19: エリア設定入力に GeoJSON でない形式のテキストを入力する例

取得した GeoJSON 形式のテキストを入力すると、入力フォームの下部に整形された GeoJSON の情報の確認が可能である。図 3.19 のように GeoJSON 形式でない文字を入力すると、エラーが表示される様になっているため、入力が正常に行われているのかの確認が可能である。

時間帯

繰り返す間隔  
1 日

06:00 AM 07:00 PM

繰り返す間隔  
1 週間

日 月 火 水 木 金 土

10:00 AM 12:00 PM

繰り返す間隔  
1 週間

日 月 火 水 木 金 土

06:00 PM 07:00 PM

+

図 3.20: 時空間フェンシング設定ページの時間帯入力画面と入力例

10:00 AM PM

12 1 2 3 4 5 6 7 8 9 10 11

CANCEL OK

図 3.21: 時間の入力モーダル

時間帯の入力フォームでは、3.3.2 項「センシングデータの管理や各プロジェクトを管理するサーバの実

装」で定義した複数の時間帯を定義可能なフォーマットに対応した入力フォームを作成する。このフォームは、繰り返す間隔の数字入力とその右側に単位の選択を行うセレクトタがあり、セレクトタでは「日」と「週間」のどちらかを選択する。セレクトタで「週間」を選択した場合のみ、その下部に曜日を示すボタンが表示され、ボタンをクリックし曜日の指定を行う。最下部には開始時間と終了時間の入力項目があり、それぞれのフォームのクリックにより、図 3.21 のような時間を入力するモーダルが表示され時間の入力を行う。複数の時間帯を定義する場合は、フォーム下部の+ボタンをクリックにより複数の時間帯を定義できる。また、定義した時間帯の削除を行う場合はフォーム右上のゴミ箱のアイコンをクリックする。この3つの項目は、サーバに定義された SensingSetting モデルの「areaName」と「geo」と「periodOfTimes」に対応している。以上の項目をすべて入力後の送信ボタンクリックにより、サーバの「/project」及び「/sensing-setting」エンドポイントに入力情報が送信される。センシングプロジェクトの定義が完了となり開始日になると、センシングプロジェクトが開始される。

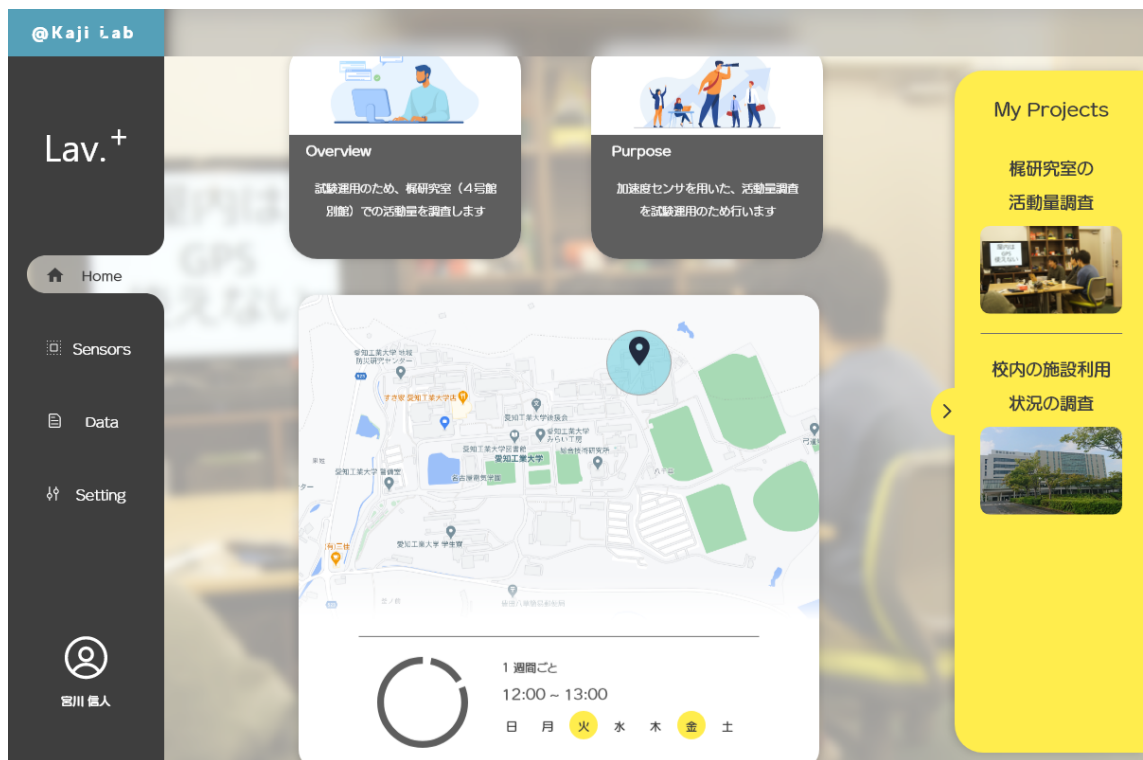


図 3.22: 管理画面ページ

センシングプロジェクトの定義を終了すると、管理画面にて定義したセンシングプロジェクトの情報閲覧が可能となる。協力者からセンシングデータを提供されると、その管理画面よりそのセンシングデータのダウンロードが可能となる。ラヴラスでは依頼者1人による複数のプロジェクトの作成を想定し、管理画面の右側に表示されたメニューより作成したセンシングプロジェクトの切り替え及び追加作成が可能となっている。

id	filename	url	file_size	upload_date	meta
0ec3d338-c8f7-4628-988e-2f294b71dfc4	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	2701567	2021-01-27T03:43:37.585Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
3a2fb4f5-1b47-47c2-8a75-45d8508c5fa7	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	2701567	2021-01-27T03:45:37.416Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
3ff4fac4-9b95-43c6-8702-a197d8453da2	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	1340374	2021-01-31T11:32:50.794Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
7daf8877-8973-46af-bbb4-2a8853769bca		<a href="#">↓</a>	2701567	2021-01-27T03:42:27.159Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
0ec3d338-c8f7-4628-988e-2f294b71dfc4	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	2701567	2021-01-27T03:43:37.585Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
3a2fb4f5-1b47-47c2-8a75-45d8508c5fa7	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	2701567	2021-01-27T03:45:37.416Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
3ff4fac4-9b95-43c6-8702-a197d8453da2	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	1340374	2021-01-31T11:32:50.794Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
7daf8877-8973-46af-bbb4-2a8853769bca		<a href="#">↓</a>	2701567	2021-01-27T03:42:27.159Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
0ec3d338-c8f7-4628-988e-2f294b71dfc4	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	2701567	2021-01-27T03:43:37.585Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
3a2fb4f5-1b47-47c2-8a75-45d8508c5fa7	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	2701567	2021-01-27T03:45:37.416Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
3ff4fac4-9b95-43c6-8702-a197d8453da2	5f9fcf377d55602698ccd9ab...	<a href="#">↓</a>	1340374	2021-01-31T11:32:50.794Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI
7daf8877-8973-46af-bbb4-2a8853769bca		<a href="#">↓</a>	2701567	2021-01-27T03:42:27.159Z	{"sdkVersion":"29","deviceM":["Accelerometer"],"LSM6DSI

図 3.23: 提供されたセンシングデータの一覧及びダウンロードを行うページ

提供されたセンシングデータのダウンロードは、Data メニューのクリックによりセンシングデータの一覧が表示され、ファイル名、ファイルサイズ、アップロード日、メタデータの情報を確認できる。センシングデータのダウンロードは、一覧の各列のダウンロードボタンのクリックにより可能である。

## 3.3.4 協力者用のセンシングスマホアプリの実装



図 3.24: サーバとスマホアプリの連携

ラヴラスでは、協力者がセンシングプロジェクト毎にスマホアプリをインストールする手間を省くため、協力者専用の Android アプリケーションを作成する。第1章でも述べたように、スマートフォンの利用者は使用頻度の低いアプリケーションを削除する傾向にある。理由としては、ストレージの容量確保のためであったり、整理整頓であったりと様々である。例えば、本プラットフォームのような基盤を用いず、個別の専用アプリケーションを作成し、クラウドセンシングを行うとする。この場合、専用アプリケーションの数は研究の数と比例する。クラウドセンシングを利用して行いたい人が増えれば増えるほど、個別の専用アプリケーションも増えていく。その都度協力者はアプリケーションをインストールする必要がある。インストールには時間や手間がかかり、ストレージを使用してしまう。1回ならまだしも、インストールの回数が2回3回と増えていくと、クラウドセンシング自体が面倒になり、非協力的になってしまう可能性がある。ストレージの使用容量が微量だったとしても、空き容量がなくなった場合に真っ先に削除される可能性もある。依頼者が別のクラウドセンシングを行う際、前回のシステムをそのまま使うとしても、アップデートや再インストールが必要となるため、協力者にとっては負担になる。また、一定期間クラウドセンシングを行わなかった場合、協力者は「もうこのアプリケーションでのセンシングは終わったのだ」と解釈し、削除する可能性がある。こうなると、いざもう一度センシングを行おうとなったときに協力者がいるとは限らない。個別でクラウドセンシングアプリケーションを作成する問題点として、仕様や操作法の違いも挙げられる。クラウドセンシングは一定の基準が定められている訳ではないため、依頼者独自の仕様や操作法である。アプリ毎に仕様を理解し操作するのは協力者にとって大きな負担になる。依頼者が所属している研究室の教授であったり、友人である場合、そのような負担も気にならないかもしれないが、赤の他人であった場合、協力者が本スマホアプリをインストールすれば、本プラットフォームで作成されたセンシングプロジェクトに協力が可能となる。これにより、依頼者はクラウドセンシングを利用するために、一からセンシング専用のスマホアプリを開発する必要がなくなり、協力者はクラウドセンシングに協力するために、何度もインストールする必要がなくなる。

本スマホアプリはサーバと連携し、センシングプロジェクトの受信やセンシングデータファイルとメ



タデータのアップロードを行う。センシングプロジェクトは適宜サーバから取得し、本スマホアプリ内のデータベースに登録する。新規のセンシングプロジェクトの場合は新しく登録し、既存のセンシングプロジェクトの場合は内容が変更している可能性を考慮して最新の方に更新する。データベースより現在時刻から一番近い指定開始時間を検索し、指定開始時間と指定終了時間を時空間フェンシングに設定する。また、設定した時間と同じセンシングプロジェクトのエリアとプロジェクト ID も設定しておく。プロジェクト ID は時空間内に進入した際に送られる通知画面に表示されるプロジェクト内容の検索に用いる。

時空間フェンシングは位置情報取得を最小限にするために、まず時間判定を行い、センシングプロジェクトで指定された開始時間にエリア判定を行う。時間判定とエリア判定を平行して行った場合でも、問題なく時空間フェンシングは動作する。しかし、我々のクラウドセンシングの中核として、協力者の負担軽減がある。位置情報の収集とスマートフォンの電池の消耗は直接的な関係がある。例えば、位置情報データの精度が高ければ高いほど電池の消耗は激しくなる。また、位置情報の計算頻度が高ければ高いほど電池の消費量は増加する。つまり、位置情報の精度と計算頻度が高いほど、より多くの電池量を消費するのである。そのため、スマートフォンの負担を軽減するには、低い精度と計算頻度で妥協する必要がある。しかし、時空間フェンシングを行う上で、位置情報は非常に重要な情報である。位置情報の精度が低いと、実際にはエリア内に進入しているのにスマートフォン上ではエリア外となってしまうたり、反対にエリア外にいるのにエリア内に進入してしまう可能性がある。それにより、途中でセンシングが終了してしまったり、ずっとエリア内にいるのにセンシング依頼通知が送られてないなどセンシングに支障をきたしてしまう。よって、位置情報の精度の低減もできない。指定開始時間以前に指定エリア内に入っているにもかかわらずセンシングは開始しないし、スマートフォンの充電を無駄に消耗させてしまう。そのため、指定開始時間後にエリア判定を行う。

協力者が指定センシングエリア内に進入したか否かは、点の多角形に対する内外判定 [30] で判定する。点の多角形に対する内外判定を図 3.25 に示す。点の内外判定とは、まず点 (赤丸) から多角形に対して水平線を引く。その線と多角形との交点 (青丸) の数が奇数個であれば多角形よりも内側にいる、偶数個であれば多角形よりも外側にいると判定できる。この場合、丁度線と多角形の辺が重なったり、線と多角形の頂点と重なると、誤った判定を行ってしまう。しかし、緯度経度は小数点以下が 7 桁もあり、位置情報は変化し続けるため、丁度重なる場合は今回考えないものとしている。実装当初は内外判定ではなく条件分岐を用いて、エリア判定を行っていた。現在位置の緯度が指定エリアの最北端よりも低く最南端よりも高い、また経度が指定エリアの最東端よりも低く最西端よりも高い場合、指定エリア内にいると判定した。しかし、この条件分岐を用いると、自ずと判定範囲が矩形になってしまう。そうなった場合、正常にエリア判定できるのは緯度経度と平行な矩形のみとなる。例えば、指定センシングエリアがひし形の場合を想定する。正常なエリア判定としては、図 3.26 の通り現在地 (赤丸) にいた場合はエリア内に進入していないため、通知は送られないはずである。しかし、条件分岐を用いたエリア判定の場合は、ひし形の外側にひし形の頂点を辺の中心とした矩形の判定エリアが形成されるため、図 3.27 のように現在地がエリア内にいると判定され、誤って通知が送られてしまう。一方で、点の内外判定は現在地から水平に引いた線とひし形のエリアの交点 (青丸) が 2 つと偶数個であるため、エリア外と判定できる (図 3.28)。これにより、依頼者が図 3.25 のようにどれだけ複雑な多角形のエリアを設定しても、内部か否かは判定可能である。今回は実装は行っていないが円のようなエリアでも判定は可能である。また、今回は GPS のみを用いたエリア判定のため、平面的なエリア判定しか行えない。1 階や 2 階などの立体的なエリア判定は今後の課題とする。

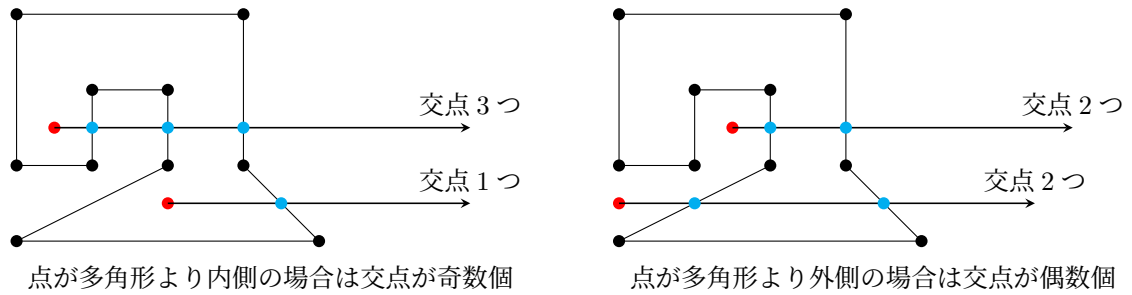


図 3.25: 点の多角形に対する内外判定

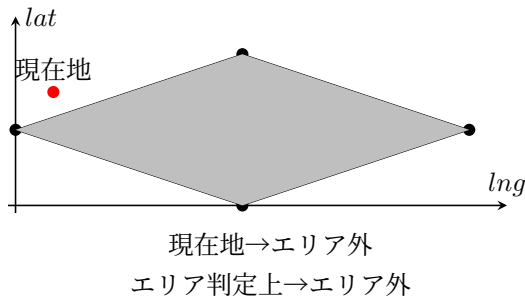


図 3.26: 正常なエリア判定

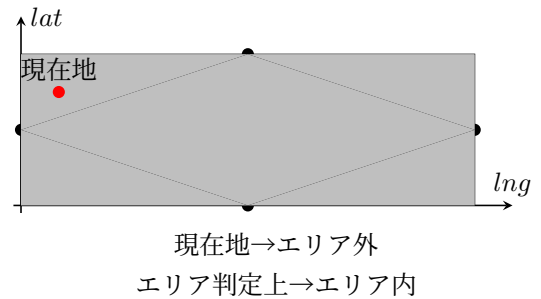


図 3.27: 条件分岐による矩形でのエリア判定

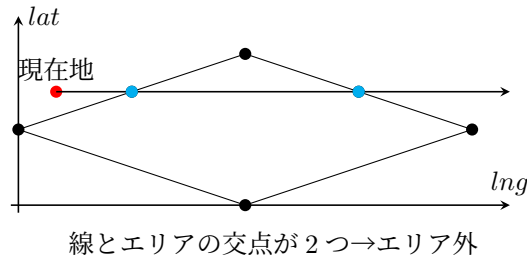


図 3.28: 点の内外判定でのエリア判定

協力者がセンシングプロジェクトで指定された時間帯とエリア内に進入したら、通知を送る。通知はヘッドアップ通知で、振動ありで送られる。ヘッドアップ通知により、スマートフォンを使用している際に目に留まりやすく、また振動もあるため、スマートフォンの画面を見ていない時やポケットに入れている時により気がつきやすくなる。通知のタップにより、アプリケーションを開き、センシング依頼の通知画面が表示される。通知画面では現在地と指定エリアを Google Maps で、時間帯をバーで、センシングプロジェクトの内容をスクロールで表示し、協力者に確認してもらう。指定エリアのマップ表示により、テキストよりも指定エリアの広さが解りやすく、現在地との位置関係も把握しやすい。

センシングデータファイルはディレクトリを監視し、ファイルへのセンシングデータ書き込み終了(書き込み用にファイルを開いて閉じた)を判定したとき、協力者のスマートフォンが Wi-Fi に接続されていればアップロードされる。まず協力者が指定エリアから退出した及び指定終了時間になった時、実行中のセンシングを終了する。センシング終了後、センシングデータを書き込んでいたファイルを閉じた際にイベント判定を行い、ファイルアップロードの段階に入る。ファイルアップロードでは、まずデータベースにセンシングファイル名とファイルを未アップロードという情報を紐づけて登録する。次に、Wi-Fi などのインターネット接続の有無を確認し、接続されている場合のみアップロードを行い、未アップロードの情報を更新する。携帯回線に接続された状態でセンシングデータをアップロードしてしまうと、協力者の

モバイル通信量を使用し負担をかけてしまう。手動でデータアップロードする場合は、携帯回線を好まない場合に Wi-Fi に接続できる場所へ移動してからアップロードするなど判断ができるため、アプリケーション側の配慮は必要としない。しかし、自動でアップロードする場合、判断する余地がないため、負担の少ない Wi-Fi に接続時のみにしなければならない。もし、Wi-Fi に接続されていない場合は、アップロードを見送り、次回ファイルアップロードの段階に入った際にまとめてアップロードする。

協力者の操作を最小限にするため、時空間フェンシングやセンシング、センシングデータのアップロードなどアプリの大部分はバックグラウンドで行う。そのため、スマホアプリを意識して開く必要はない。開くのは時空間内に進入した際に送られるヘッドアップ通知をタップした時のみである。それ以外は基本的に画面上のアイコンをタップしてアプリケーションを開くなどは不要である。通知画面を承諾した後も、「センシング中はアプリケーションを開いておかなければならない」という仕様ではないため、すぐに閉じて構わない。アプリケーション自体を終了させなければ、途中でセンシングは終了しない。バックグラウンドにより、協力者の操作などの負担の軽減に加え、センシングをしているという感覚を意識させないため、より普段の振る舞いをセンシングできる。

### 3.4 動作検証

本センシングプラットフォームを用いて、依頼者側のセンシングプロジェクトの作成やセンシングデータのダウンロード、協力者側の時空間フェンシングや通知、センシングプロジェクトに基づいたセンシング、センシングデータのアップロードなど一連の流れが正常に動作するか検証する。動作が正常か否かを確認するためにチェック項目を設けた。依頼者側のチェック項目としては、「センシングプロジェクトを作成できるか」、「アップロードされたセンシングデータはダウンロード可能か」を挙げた。協力者側のチェック項目としては、「サーバからセンシングプロジェクトを受信できるか」「依頼者によって作成されたセンシングプロジェクトに基づいて、時空間フェンシングができていないか」、「指定された時空間内に進入した際にヘッドアップ通知を送っているか」、「正しくセンシングできているか」、「センシング後、Wi-Fi 接続時にセンシングデータをアップロードできているか」を挙げた。



図 3.29: 動作検証エリア

今回の動作検証は、13時から14時までの愛知工業大学4号館別館(図3.29)での活動量を加速度センサで調査するといったシナリオに基づいて行った。スマートフォンはGoogle Pixel 4を使用した。なお、今回の動作検証は実装の初期段階であるため、複雑な状況は想定せず、単純な状況で行う。依頼者側は新規にWebアプリのユーザ登録を行い、協力者側は新規にスマホアプリをインストールする。そのため、サーバには1つもセンシングプロジェクトがない状態とする。

結果として、我々が期待した通りの動作が確認できた。まず、依頼者はWebアプリのユーザ登録を行う。次に、Webアプリにてセンシングプロジェクトを作成する。内容としては、時間帯を

協力者側のスマホアプリではサーバからセンシングプロジェクトを受信する。サーバに定義された1つのセンシングプロジェクトを問題なく受信できたため、正常であると判断する。次に、受信したセンシングプロジェクトに基づいて時空間フェンシングを行う。指定開始時間(13時)前に動作検証エリアに進入しても通知は送られてこなかったが、指定開始時間になると通知が送られてきた。エリアを退出するとセンシングは終了し、再度進入すると再度センシングが開始された。エリアに進入したまま指定終了時間を迎えると、センシングは終了した。これにより、時間判定・エリア判定ともに正常であると判断する。動作検証エリアが屋内であるため、GPSでエリア判定を行うと位置情報誤差はあったが、動作は確認できた。屋内でのエリア判定は今後の課題とする。また、通知も正常に送られていると確認できた。次に、Wi-Fi接続時のみセンシングデータをアップロードする。最初のセンシングはWi-Fiに接続していないモバイル通信の状態で行った。終了後にサーバを確認したところ、センシングデータは1件もなかった。指定時間内に再度エリアに進入し、今回はWi-Fiに接続した状態でセンシングを終了した。すると、サーバには今回のセンシングデータと最初のセンシング終了後にアップロードされなかったセンシングデータの2件が見受けられた。Wi-Fiに接続時のみセンシングデータのアップロードは正常に確認できた。動作検証エリアが屋内であるため、GPSでエリア判定を行うと位置情報誤差はあったが、動作は確認できた。屋内でのエリア判定は今後の課題とする。指定した時空間内に進入すると、ヘッドアップ通知が表示され、それをタップし、依頼を承認するとセンシングが開始され、正常な動作が確認できた。センシング後にサーバにアップロードされたセンシングデータを確認したところ、シナリオ通りに加速度が収集されていたため、動作検証は成功とする。

```
{
  "id": "3ff4fac4-9b95-43c6-8702-a197d8453da2",
  "filename": "3ff4fac4-9b95-43c6-8702-a197d8453da2.png",
  "originalname": "5f9fcf377d55602698ccd9ab_20210126-210844.log",
  "fileSize": 7340374,
  "uploadDate": "2021-01-31T11:32:50.794Z",
  "meta": {
    "sdkVersion": "29",
    "deviceModel": "Pixel4",
    "sensors": [
      {
        "Accelerometer": "LSM6DSR",
        "Gyroscope": "LSM6DSR"
      }
    ]
  },
  "projectId": "5f9fcf377d55602698ccd9ab"
}
```

図 3.30: 収集した生センシングデータ (一部)

## 第4章 おわりに

### 4.1 まとめ

本稿では初めに時空間フェンシングについて定義した。時空間フェンシングとは、地理的な場所を制限するジオフェンシングに時間要素を加えて拡張し、時間と場所で境界を区切る独自のフェンシング手法である。時空間フェンシングによって時間とエリアで境界を区切ると、依頼者は様々なシチュエーションを指定したクラウドセンシングが可能となる。一方で、時間やエリアに依存していないデータ収集には適していない。

次に、時空間フェンシングに基づいたクラウドセンシングプラットフォームの要求仕様を定義した。

依頼者がセンシングデータや各プロジェクトを管理するサーバを実装した。

依頼者がセンシングプロジェクトの作成やセンシングデータのダウンロードなどを行う Web アプリケーションを実装した。

協力者が時空間フェンシングに基づいたセンシングに協力するためのスマートフォンアプリケーションを実装した。

動作検証では、本プラットフォームの一連の流れが正常に動作するか検証した。

### 4.2 今後の課題

まず今後の課題としては、今回完全に実装できなかった協力者のプライバシー保護の徹底が挙げられる。

プラットフォームの運用を行うための課題として利用規約の作成をする必要がある。利用規約には、要求仕様の 3.2.3 「ラヴラスの利用及び利用規約の方針」に示したセンシングデータの保存期間以上の保存の禁止などの項目などに加え、利用に関しての前提条件の説明やプラットフォームが負う責任の詳細など制定し実際の運用に対応させるとともに、利用規約を前提として今後の実装を行うためである。

時空間フェンシングの課題として、時空間フェンシングの最大範囲の最適値の模索がある。要求仕様の 3.2.1 「クラウドセンシングプラットフォームとしての設計基盤」に時空間フェンシングの最大範囲を設定すると述べた。これは、設定する時空間を無制限に拡大してしまうと時空間フェンシングの意義がなくなってしまうからである。しかし、ラヴラスでは、現在、評価実験を行えていないため最大範囲をどこまでとするのかの指標がないため適切な時空間フェンシングの最大範囲の設定ができていない。この課題は、時空間フェンシングによって制限される時空間がどの範囲までフェンシングとして有効なのかを評価実験によって調査した上で最大範囲を設定する必要がある。

プラットフォーム全体の実装面における課題として、GPS 以外でのエリア判定を可能にするシステムを導入、参加済みのセンシングプロジェクトの辞退や削除要請の実装、センシングデータの軽量化、センシング設定のプリセットの作成がある。GPS 以外でのエリア判定を導入する理由は、現在採用している GPS によるエリア判定では、狭い範囲でのエリア範囲や GPS の不安定な屋内の判定、更には屋内における階数による判定や部屋単位のエリア判定ができないためである。これらの判定は、Wi-Fi や BLE といった電波を用いたエリア判定や建物から発生する磁気を用いたエリア判定の導入により実現する。GPS 以外でのエリア判定を実装するには、まずサーバに GPS 以外のエリア判定を定義できるモデルを作成した上で Web アプリの GPS 以外のエリア判定の設定項目の追加、そしてスマホアプリに GPS 以外の判定アルゴリズムを作成する必要がある。参加済みのセンシングプロジェクトの辞退や削除要請の実装は、要求仕様として 3.2.2 「プライバシーの保護」に定義を行ったものの現在実装に至っていない。この課題は、ラ

ヴラスの運用にあたって必須の実装であるため優先的に解決する必要がある。センシングデータの軽量化は、現在採用しているセンシングデータのフォーマットでのセンシングデータが肥大化する傾向にあるため軽量化によって要求仕様の3.2.4「利用者への配慮」に挙げた通信量に関する物理的負担を軽減するとともに、リアルタイムでのセンシングを実装を視野に入れる。現在採用されているセンシングデータのフォーマットは、すべての情報をテキストにて表現を行っているためセンシングデータの特性を考慮するとセンシングデータは数字による表現が多いためテキストでなく数値を表現するバイナリを使用した上で、センシングデータの圧縮によって大幅なセンシングデータの削減を実現できると考えられる。センシング設定のプリセットは、多くの需要が想定される加速度のセンシングを利用した歩数や消費カロリーなどのセンシングをプリセットとして定義し、3.2.4「利用者への配慮」に挙げたユーザインターフェイスの向上を目的とし、クラウドセンシングの定義をより簡略化が可能となる。

サーバの課題として、依頼者情報の信頼性の向上が挙げられる。これは、要求仕様の3.2.3「ラヴラスの利用及び利用規約の方針」に定義した依頼者情報の明示が依頼者によって保証されるためプラットフォームとして依頼者情報を保証していない。そのため、依頼者の設定したメールアドレスが組織や機関のものかどうかなどの確認を行うなどで依頼者情報の信頼性を向上させる目的させる必要がある。

Web アプリの課題として、Web アプリ上でのセンシングデータの可視化が挙げられる。ラヴラスは、将来的に自治団体や地域施設の管理者といった依頼者の利用を想定している。現在のセンシングデータの提供方法は、センシングデータのダウンロードという形でを行っているが、こういった依頼者によってセンシングデータのダウンロード・解析は困難であると想定される。そのため、Web アプリ上でのセンシングデータの可視化により、センシングデータの解析が困難な依頼者にとってもクラウドセンシングを実施できるプラットフォームを実現する必要がある。

スマホアプリの課題として、協力者の基礎情報登録が挙げられる。依頼者によっては性別や身長とセンサデータを結びつけて調査を行うケースも考えられる。そうした場合、協力者に性別や身長など自身の基礎情報をスマホアプリインストール時に登録してもらう必要がある。基礎情報の登録機能を実装する場合、自身の性別や身長を提供したくない協力者への登録強制はせず、任意での登録とする。

その他の課題として、マイクやカメラのセンシングデータの値の抽象化や行動認識モジュールによる収集、スマートウォッチへの対応などが挙げられる。マイクやカメラによって収集した音声・映像データは、プライバシーに関わる情報が非常に多く含まれてしまっている。依頼者にとっても、音声・映像データの取り扱いが困難である。そのため、プラットフォーム側で信号処理モジュールを通して発言数、騒音レベル、明るさレベル、といった値に抽象化した上で依頼者側にダウンロードしてもらう必要がある。加速度等、他のセンサ信号についても、歩数や消費カロリーといった多くの需要が想定される値に行動認識モジュールによって可能にする。スマートフォンだけではなくスマートウォッチにも対応すると、より柔軟で多様なデータ収集が可能となる。例えば、スマートウォッチのセンサからバイタルデータを収集して、運動負荷の調査などが可能となる。

これらの課題を解決した上で、依頼者・協力者ともにより柔軟に、より利用しやすくするため、評価実験の実施が必要がある。評価実験は今後このラヴラスを運用するにあたって、とても重要なものである。現在、基礎的な実装は動作検証にて確認できているものの、実際に依頼者と協力者に利用してもらったわけではない。そのため、今までの要求仕様を基にした実装が利用ユーザにとって有益なものかは定かではない。依頼者にとってクラウドセンシングがより簡易的に、より柔軟にできているか、協力者にとってクラウドセンシングに負担が少なく協力できているかを評価してもらい、再構築を行う必要がある。評価実験を行うにあたって、プライバシーの保護と個人情報の対策を万全にする必要がある。万全ではない状態で行うと、万が一不具合が起こった場合にユーザの個人情報を流出する恐れがあり、ユーザから信用を失う可能性があるため、評価実験は慎重に行う必要がある。評価実験を基に、依頼者側ではより多くのデータを集め、協力者側ではより安心してより簡易的にセンシングを可能にする必要がある。

## 謝辞

本研究を進めるにあたり，多くの御指導，御鞭撻を賜りました梶克彦准教授に深く感謝致します．  
最後に，日頃から熱心に討論，助言してくださいました梶研究室のみなさんに深く感謝致します．

## 参考文献

- [1] 下坂正倫. クラウドソーシングの現状と可能性: 3. クラウドセンシングの研究動向. 情報処理, Vol. 56, No. 9, pp. 891–894, aug 2015.
- [2] 5g と 4g で何が変わる? — 法人のお客さま — ソフトバンク. [https://www.softbank.jp/biz/5g/column3/#biz-5g-01\\_02](https://www.softbank.jp/biz/5g/column3/#biz-5g-01_02). (Accessed on 01/30/2021).
- [3] 総務省 | 令和元年版 情報通信白書 | 情報通信機器の保有状況. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r01/html/nd232110.html>. (Accessed on 01/23/2021).
- [4] Jinwei Liu, Haiying Shen, Husnu S. Narman, Wingyan Chung, and Zongfang Lin. A survey of mobile crowdsensing techniques: A critical component for the internet of things. *ACM Trans. Cyber-Phys. Syst.*, Vol. 2, No. 3, June 2018.
- [5] 岩本健嗣, 杉森大輔, 松本三千人. 3 軸加速度センサを用いた歩行者推定手法. 情報処理学会論文誌, Vol. 55, No. 2, pp. 739–749, feb 2014.
- [6] 鈴木惇也, 秋山征己, 田中博, 五百蔵重典. スマートフォン内蔵センサを用いた歩行位置推定に関する基本実験と評価. Technical Report 10, 神奈川工科大学情報学部情報工学科, 神奈川工科大学情報工学専攻, 神奈川工科大学情報工学専攻, 神奈川工科大学情報工学専攻, nov 2012.
- [7] 並木渉, 市野将嗣, 笠原弘樹, 吉井英樹, 吉浦裕. スマートフォンの気圧センサと気象情報を用いた高度推定手法. マルチメディア、分散協調とモバイルシンポジウム 2013 論文集, 第 2013 巻, pp. 1133–1139, jul 2013.
- [8] 米田圭佑, 望月祐洋, 西尾信彦. 気圧センシングを用いた行動認識手法. Technical Report 14, 立命館大学大学院情報理工学研究科, 立命館大学総合科学技術研究機構, 立命館大学情報理工学部, mar 2014.
- [9] 成元椋祐, 梶田宗吾, 山口弘純, 東野輝夫. スマートフォンを用いた興味行動のセンシング. Technical Report 49, 大阪大学大学院情報科学研究科, 大阪大学大学院情報科学研究科, 大阪大学大学院情報科学研究科, 大阪大学大学院情報科学研究科, feb 2018.
- [10] 高藤巧, 樋口雄大, 山口弘純, 東野輝夫. スマートフォン利用者の迷い行動の検出に関する一検討. Technical Report 1, 大阪大学大学院情報科学研究科, 大阪大学大学院情報科学研究科, 大阪大学大学院情報科学研究科, 大阪大学大学院情報科学研究科, feb 2016.
- [11] A. Suyama and U. Inoue. Using geofencing for a disaster information system. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1–5, 2016.
- [12] 吉田崇洋, 田口大悟, 井口信和. 位置情報に基づくウェアラブルデバイス向け農作業支援システムにおける農作業記録作成機能の開発. 2016 年度 情報処理学会関西支部 支部大会 講演論文集, 第 2016 巻, sep 2016.
- [13] 西村友洋, 樋口雄大, 山口弘純, 東野輝夫. スマートフォンを活用した屋内環境における混雑センシング. 情報処理学会論文誌, Vol. 55, No. 12, pp. 2511–2523, dec 2014.



- [14] 細川諒, 白石陽. 利用者の歩行特性と乗降車順を考慮したバス内混雑状況推定手法の提案. 第 82 回全国大会講演論文集, 第 2020 巻, pp. 221–222, feb 2020.
- [15] 朴斌, 相原健郎. クラウドソースによるモバイルセンシングを用いた路面状況推定. 情報処理学会論文誌, Vol. 59, No. 10, pp. 1784–1793, oct 2018.
- [16] H. Tangmunarunkit, C. K. Hsieh, B. Longstaff, S. Nolen, J. Jenkins, C. Ketcham, J. Selsky, F. Alquaddoomi, D. George, J. Kang, Z. Khalapyan, J. Ooms, N. Ramanathan, and D. Estrin. Ohmage: A general and extensible end-to-end participatory sensing platform. *ACM Trans. Intell. Syst. Technol.*, Vol. 6, No. 3, April 2015.
- [17] ohmage — open data collection. <http://ohmage.org/>. (Accessed on 01/23/2021).
- [18] Denzil Ferreira, Vassilis Kostakos, and Anind Dey. Aware: Mobile context instrumentation framework. *Frontiers in ICT*, Vol. 2, , 05 2015.
- [19] Aware – open-source context instrumentation framework for everyone. <https://awareframework.com/>. (Accessed on 01/23/2021).
- [20] Kasthuri Jayarajah, Rajesh Krishna Balan, Meera Radhakrishnan, Archan Misra, and Youngki Lee. Livelabs: Building in-situ mobile sensing & behavioural experimentation testbeds. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, p. 1–15, New York, NY, USA, 2016. Association for Computing Machinery.
- [21] Yutaka Arakawa and Yuki Matsuda. Gamification mechanism for enhancing a participatory urban sensing: Survey and practical results. *Journal of Information Processing*, Vol. 24, pp. 31–38, 01 2016.
- [22] 河中祥吾, 松田裕貴, 諏訪博彦, 藤本まなと, 荒川豊, 安本慶一. 観光客参加型センシングによる観光情報収集におけるゲーミフィケーションの有効性調査. マルチメディア, 分散協調とモバイルシンポジウム 2018 論文集, 第 2018 巻, pp. 145–151, jun 2018.
- [23] 松田裕貴, 荒川豊, 安本慶一. 多様なユースケースに対応可能なユーザ参加型モバイルセンシング基盤の実装と評価. マルチメディア, 分散協調とモバイルシンポジウム 2016 論文集, 第 2016 巻, pp. 1042–1050, jul 2016.
- [24] Parmosense — top. <https://ubi-naist.github.io/ParmoSense/>. (Accessed on 01/23/2021).
- [25] 坂村美奈, 米澤拓郎, 伊藤友隆, 中澤仁, 徳田英幸. Minaqn: 市民参加型まちづくりのための参加型センシング web プラットフォーム. 情報処理学会論文誌, Vol. 57, No. 10, pp. 2162–2174, oct 2016.
- [26] 山崎昶. 法則の辞典. 朝倉書店, 2006.
- [27] フェイスブック ceo「私の過ち」 米議会で謝罪: 日本経済新聞. <https://www.nikkei.com/article/DGXMZ029242870R10C18A4000000>. (Accessed on 01/23/2021).
- [28] Gdpr. <https://www.ppc.go.jp/enforcement/infoprovision/laws/GDPR/>. (Accessed on 01/23/2021).
- [29] geojson.io. <http://geojson.io/#map=2/20.0/0.0>. (Accessed on 01/28/2021).
- [30] 【第 2 回】点の多角形に対する内外判定 | 【技業 log】技術者が紹介する nttpc のテクノロジー | 【公式】nttpc. [https://www.nttpc.co.jp/technology/number\\_algorithm.html](https://www.nttpc.co.jp/technology/number_algorithm.html). (Accessed on 01/28/2021).