

クラウド連携型 地図 SDK 統合プラットフォームの開発

—MapConductor: 地図 SDK の共通言語の開発—

1. 背景

Android / iOS アプリケーションでスムーズな地図表示をするためには、Google Maps SDK や MapKit などの「ネイティブ地図 SDK」を用いる。これらにおいて数千・数万箇所以上のデータを表示するような本格的な地図アプリケーションを開発するには、それなりに専門的知識を要する。機能的な問題だけでなくメモリ管理やパフォーマンスなど、単に API を実行するだけでは解決できない問題が多く存在する。

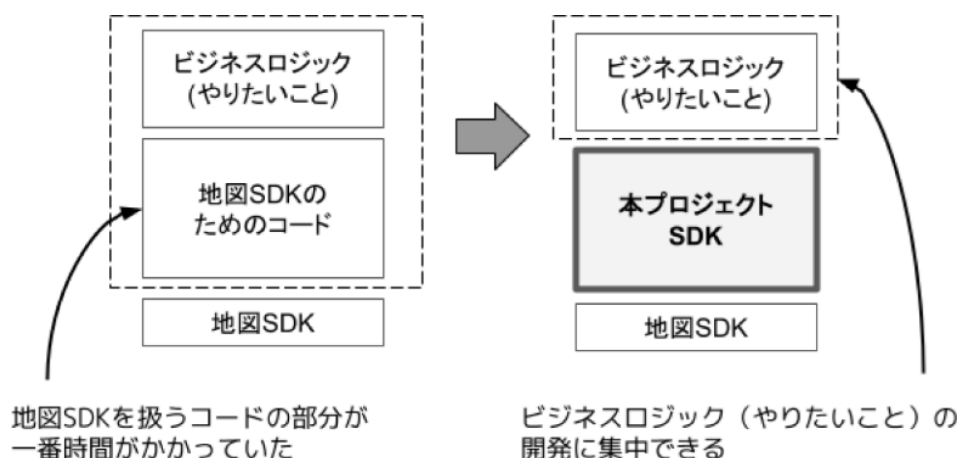
加えて、地図 SDK ごとに大きく異なるライセンス体系や料金体系も、開発・事業の自由度を制限する要因となっている。ライセンスの変更や料金の値上げなどにより別の地図 SDK に載せ替える場合、使用する地図 SDK に合わせた最適化をする必要があり、高いコストとリスクを伴っていた。

そこで我々は「地図の共通化レイヤー」を開発することで、この問題を解決することにした。サポートする全ての地図 SDK で“可能な限り”共通の API と共通の動作を提供することで、最小限のコストで地図 SDK を切り替えることができる仕組みの開発を目指すことにした。

2. 目的

本プロジェクトで目指したのは、モバイルアプリケーション開発者が「地図上に何を表現したいのか」という本質に集中できる環境の実現である。サポートする複数の地図 SDK をラッピングして共通の API を提供することで、開発の効率アップや地図 SDK の切り替えに掛かるコストの最小限化の実現を図っている。

またデータの管理にはクラウドを用いることが一般的だが、これも自由に切り替えができる仕組みを提供することで、開発者の生産性向上、地図データ提供者の市場拡大、地図 SDK ベンダー間の健全な競争を促進し、地図開発全体の民主化と持続的な発展に貢献することが本プロジェクトの目的である。



地図SDKを扱うコードの部分が
一番時間がかかっていた

ビジネスロジック (やりたいこと) の
開発に集中できる

図 1. 本プロジェクト SDK の位置付けと効果

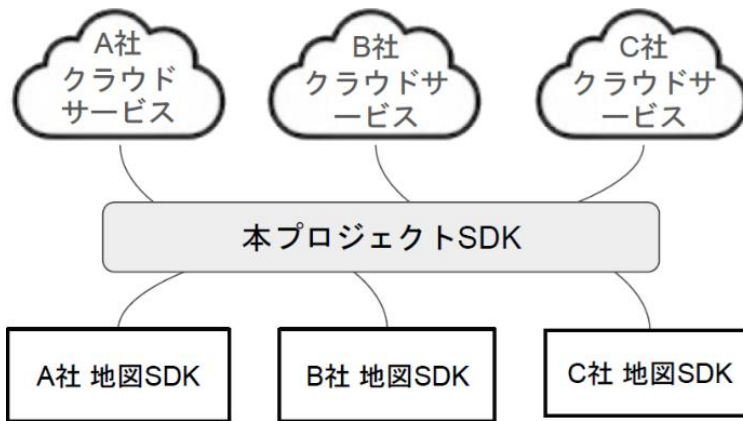


図 2. 各社地図クラウドも抽象化することで切り替えやすくする

3. 製品・サービスの内容

本プロジェクトにおいては「MapConductor(マップコンダクター)」というミドルウェア SDK を Android と iOS 向けに開発をした。どちらの SDK においても、同じ文法・プロパティ、同じ動作を行うように実装されている。

具体的には以下の機能を提供する。

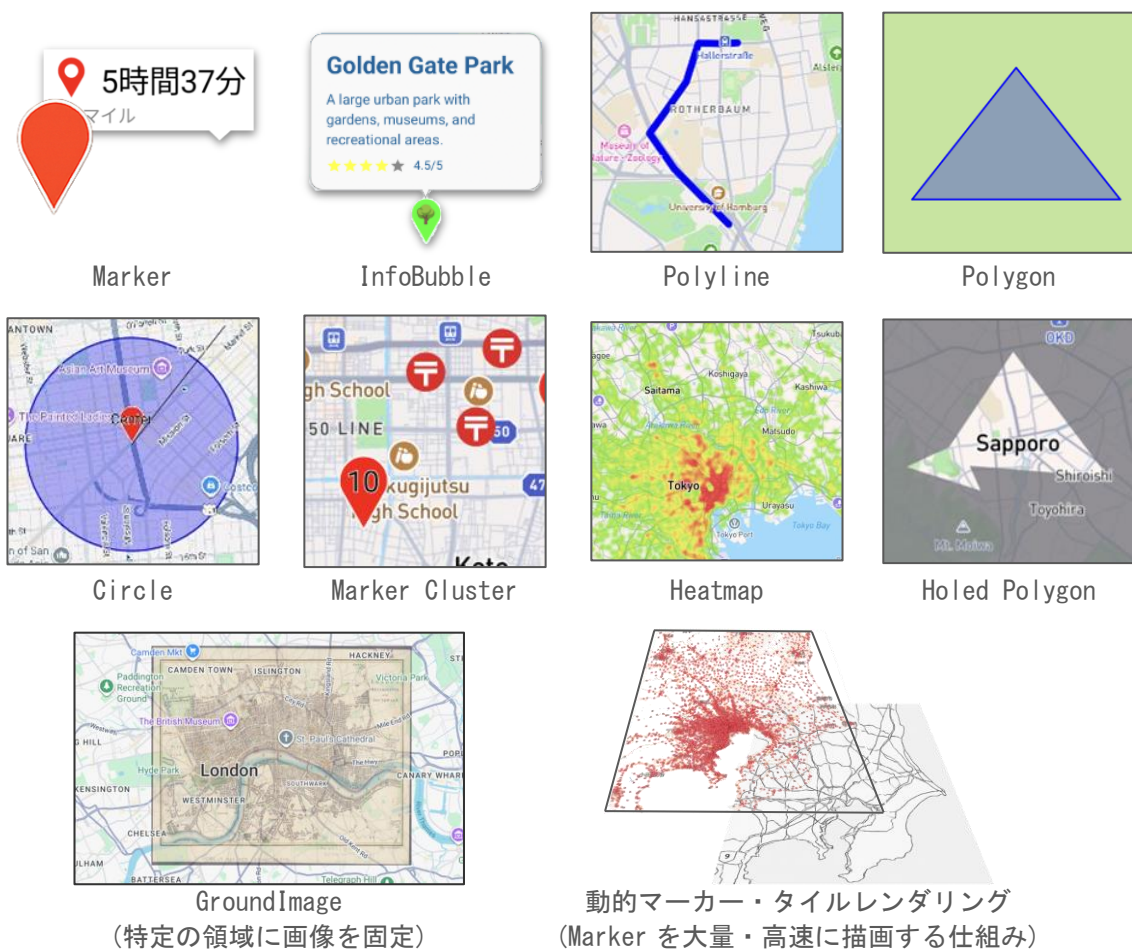


図 3. 本プロジェクト SDK が提供する機能例

近年、モバイルアプリケーション開発においては「宣言的 UI」という思想を用いることが主流となっている。これは画面の状態を「ステート」と捉え、画面描画に影響する値が変更されたことを検出すると、必要最低限な部分だけが再描画される仕組みである。これを実現しているのが、Android では Jetpack Compose であり、iOS では SwiftUI である。

宣言的 UI においては「そこに部品(コンポーネント)がある」と考えるだけでよく、位置関係やイベントリスナーなどは意識しなくて良い。これを地図に適用すると「マーカー(ピン)が指定の緯度経度の位置にある」だけを考えれば良く、地図が現在どの領域を表示しているかは考えなくて良い。つまり「存在する」という宣言だけをすればよいのである。

本プロジェクト SDK においては、Android・iOS の両方で同じ文法で記述できる API を開発した。このことによりアプリケーション開発者は、OS や地図 SDK の違いを意識することなく、地図機能を開発できるのである。

■ Android/Jetpack Composeのコード

```
@Composable
fun MyMap(markers: List<MarkerState>) {
    MapView(...) {
        markers.forEach { markerState ->
            Marker(markerState)
        }
    }
}
```

■ iOS/SwiftUI のコード

```
struct MyMap: View {
    ...
    var body: some View {
        MapView(...) {
            for markerState in self.markers {
                Marker(state: markerState)
            }
        }
    }
}
```

図 4. マーカーを表示する部分は同じ文法で記述ができる

クラウド連携において当初は、本プロジェクト SDK に特化した地図クラウドの構築を構想していたが、プロジェクトの中で得た市場調査の結果やユーザーフィードバックなどを勘案した結果、各地図 SDK ベンダーが既に提供している地図クラウドとの共生を目指すことにした。具体的には、各地図クラウドとの連携を簡単にする統一した API のアダプターの開発を目指している。現時点では、ArcGIS Online のみから、利用者が所有する非公開データの読み込みや交通情報のレイヤーを読み込むことができる。今後、機能拡張を行っていきたい。

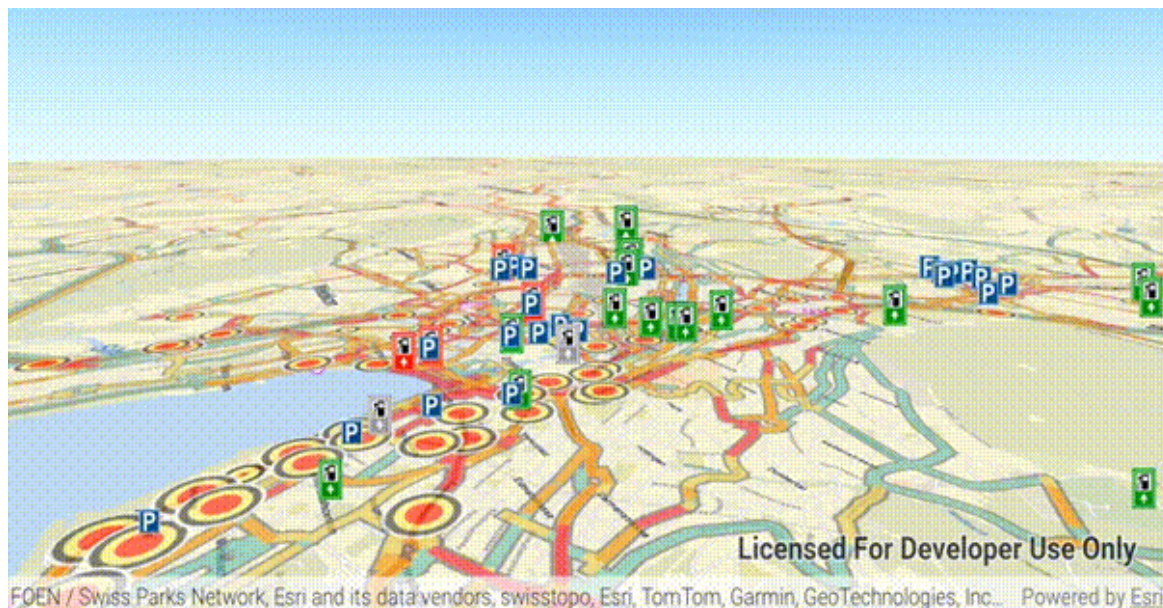


図 5. ArcGIS Online にある非公開な駐車場データと交通情報レイヤーを表示した様子

また、実験的な取り組みとして、GPS による位置測位と加速度センサーなどを用いた位置情報プロバイダーの開発を行った。GPS からの位置情報取得はバッテリー消費が激しい。そこで GPS にアクセスする秒数を伸ばし、その間に加速度センサーなどを用いて位置情報を推測することで、近似した位置測位を可能にしながら、バッテリー消費量を抑えることに成功した。

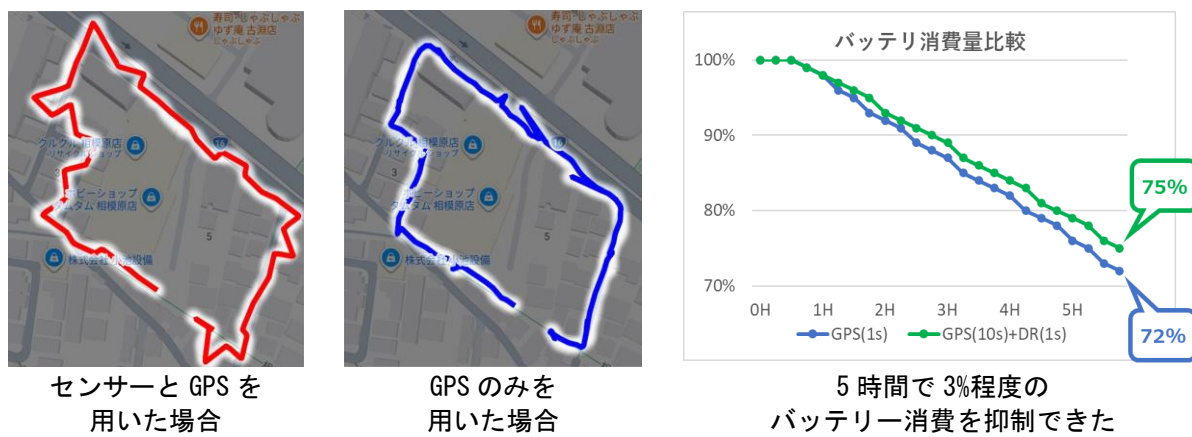


図 6. GPS とセンサー類を用いた位置情報プロバイダーの結果図

4. 新規性・優位性

地図業界において、各社地図 SDK を横断的にラッピングして共通的な API を提供するという地図の Kubernetes のようなプロジェクトは未だに確立されていない。ここに本プロジェクトの新規性がある。

そして本プロジェクトの優位性とは、「地図 SDK を利用するが、依存度を最低限にする」という実装をしている点である。例えば地図の上で、どのオーバーレイがクリックされたか、どの位置がクリックされたか、というのを、地図 SDK の判定機能は利用せずに、独自ロジックで判定をしている。

2025 年度未踏アドバンスト事業

この実装方法は他の API ラッパーライブラリとは一線を画す。限定的に複数の地図 SDK を抽象化して1つの API を提供するプロジェクトというのは、いくつか存在する。ただし多くの場合は1つの SDK において(主にFlutter や React Native などのクロスプラットフォーム向けに)Android / iOS だけをラッピングしていて、内部的には SDK の API を呼び出しているだけのことが多い。

過去に Mapstraction という同じコンセプトのプロジェクトが存在したが失敗した。原因は単なるラッパーライブラリであったことに尽きる。独自ロジックを持たないことで、地図 SDK 側の変更に振り回され、開発が受動的になり、コアメンバーが疲弊したことが記されている。

本プロジェクトにおいては、地図 SDK の機能をなるべく使わず、独自ロジックを実装することで、新しい地図 SDK を対応する場合も、対応するためのドライバーを書くコストは比較的少なく済む。結果として 8 ヶ月という短期間ながらも、数多くの地図 SDK をサポートすることに成功している。

【Android】

Google Maps、Mapbox、MapLibre、Here、ArcGIS

【iOS】

Google Maps、MapLibre、MapKit

5. 事業普及（または活用）の見通し

本プロジェクトの技術的な有用性や社会的な意義は開始当初から認められるものの、ビジネスとして事業展開は難しい点が指摘されていた。そこで本プロジェクトは OSS(オープンソースソフトウェア)として展開することとした。具体的には OSGeo 財団の Incubation 制度への採択を目指し、世界的な OSS コミュニティの構築を目指す。

6. 期待される波及効果

世界中の地図アプリケーション開発者において、選択肢の 1 つとして名前が挙がるくらいの波及効果を見込んでいる。これにより単なる便利なライブラリから、地図アプリケーション開発の中間プラットフォームとして昇華し、学術・産業問わず地図アプリケーション開発の分野に大きな影響を与えることを見込んでいる。

7. イノベータ名（所属）

大森 高明（所属：非公開）

勝又 雅史（所属：MK GeekLab LLC）

（参考）関連 URL MapConductor Web サイト

<https://mapconductor.com/ja/>