

Geeks Who Drink in Fukuoka (2025年を振り返る)

Dec 23 2024

2025年は**MapLibre Native** + iOSで
完全オフライン地図ができた👏👏らしいな!!

Satoshi Koda (Ocean Solution Technology Co.,Ltd.)

〔謝辞〕 本スライドの書体は全て

LINE Seed JP by LINEヤフー株式会社

です。オープンライセンスに感謝!!

はじめに：忙しいGeo開発者の皆様に、結論だけ

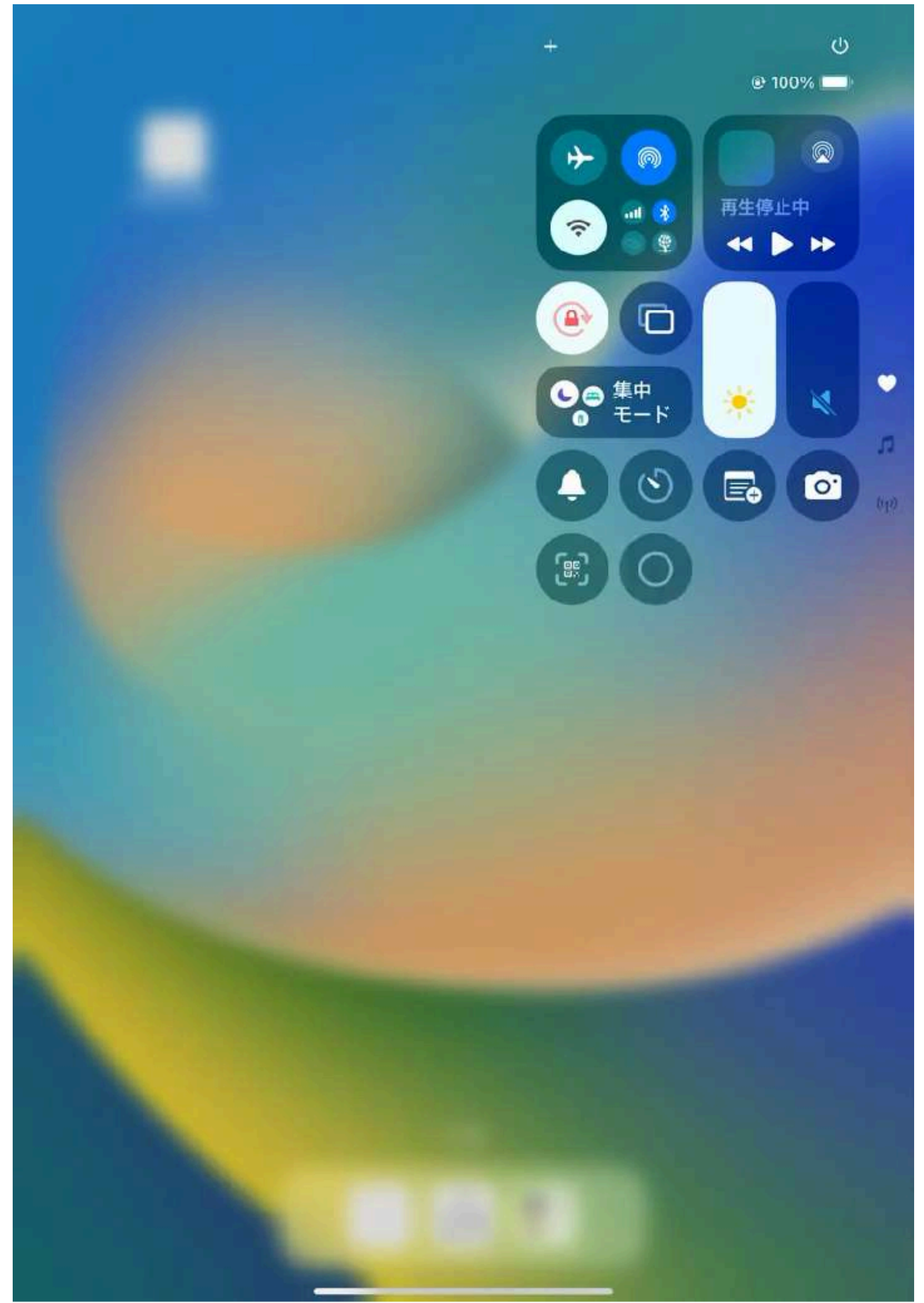
```
MLNVectorTileSource(  
    identifier : "hogehego",  
    tileURLTemplates : [ "mbtiles://ローカルFileURL" ]  
)
```

ファイルはアプリバンドル内に置き、これでいけました

- ・ アプリデータ内のファイルも同様にできるか、今後検証します
- ・ style.jsonで使えるかも今後検証します
- ・ MapLibreのオフライン機能も試しましたが、自分の要件ではこちらかなと

はじめに（続）

- ・ 今日作ったもののデモ動画（1分半）
- ・ ソースは後日GitHubに載せます
- ・ アカウントは ec22s です





国府田 諭

こうだ さとし

ec22s

- 四捨五入で60代の老体エンジニア
- 東京 → 埼玉 → 和歌山 → 鹿児島 → 長崎

現在：オーシャンソリューションテクノロジー(株)

- 趣味：車，音楽，写真（インスタ@ec22s）
- 好きな曲：Negicco “SNSをぶっとばせ”



弊社の漁業者向けiPadアプリ「トリトンの矛」





漁業者支援システム「トリトンの矛」IoT

仕事効率化

5件の評価
4.0
★★★★☆

年齢
4+
歳

カテゴリ
仕事効率化

デベロッパ
オーシャンソリ

言語
JA
その他1言語

サイズ
49.1
MB

アップデート >

いつも「トリトンの矛」をお使いいただきありがとうございます。
今回のアップデート内容は次のとおりです。[さらに表示](#)

プレビュー

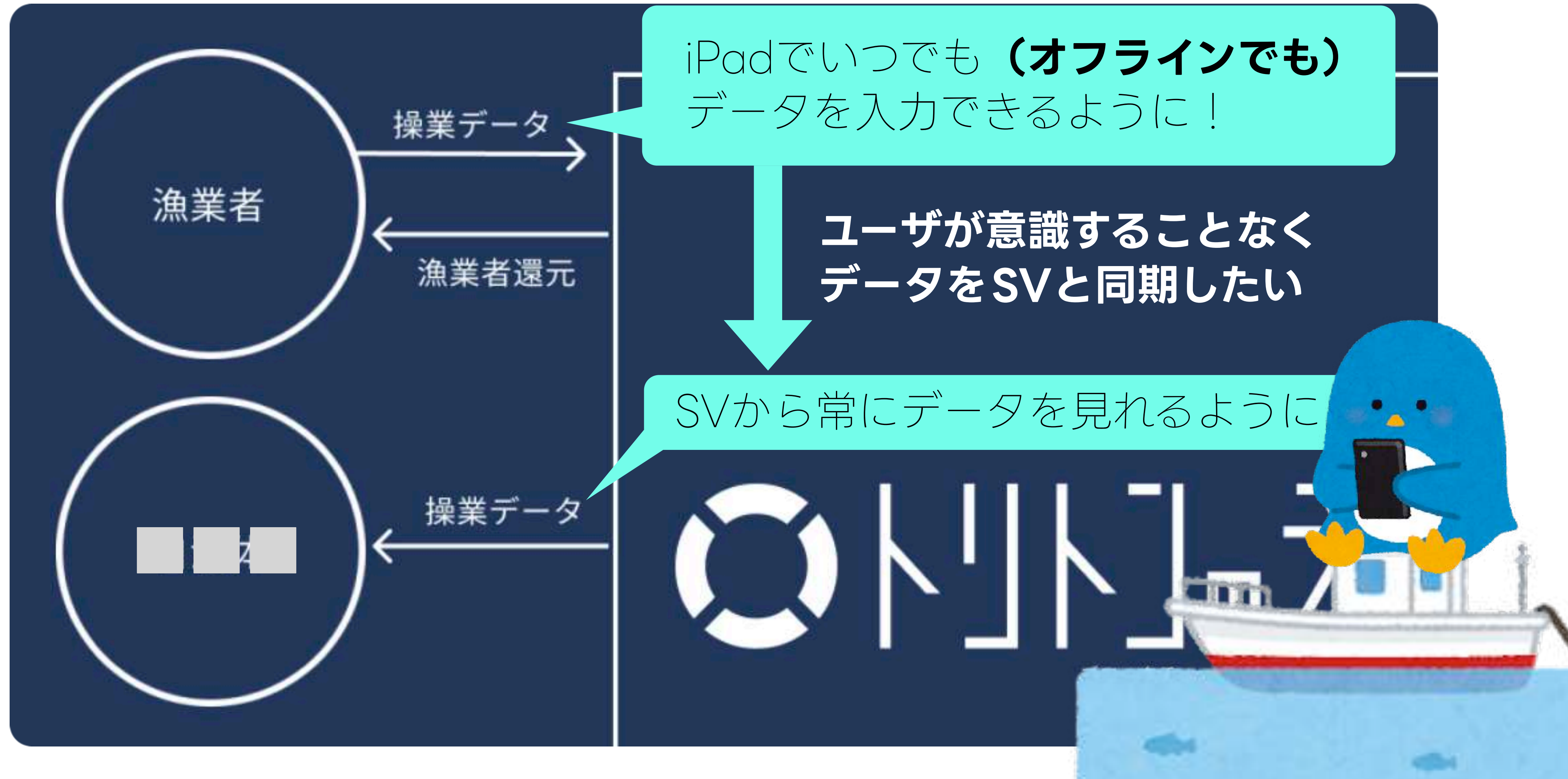


■アプリの概要

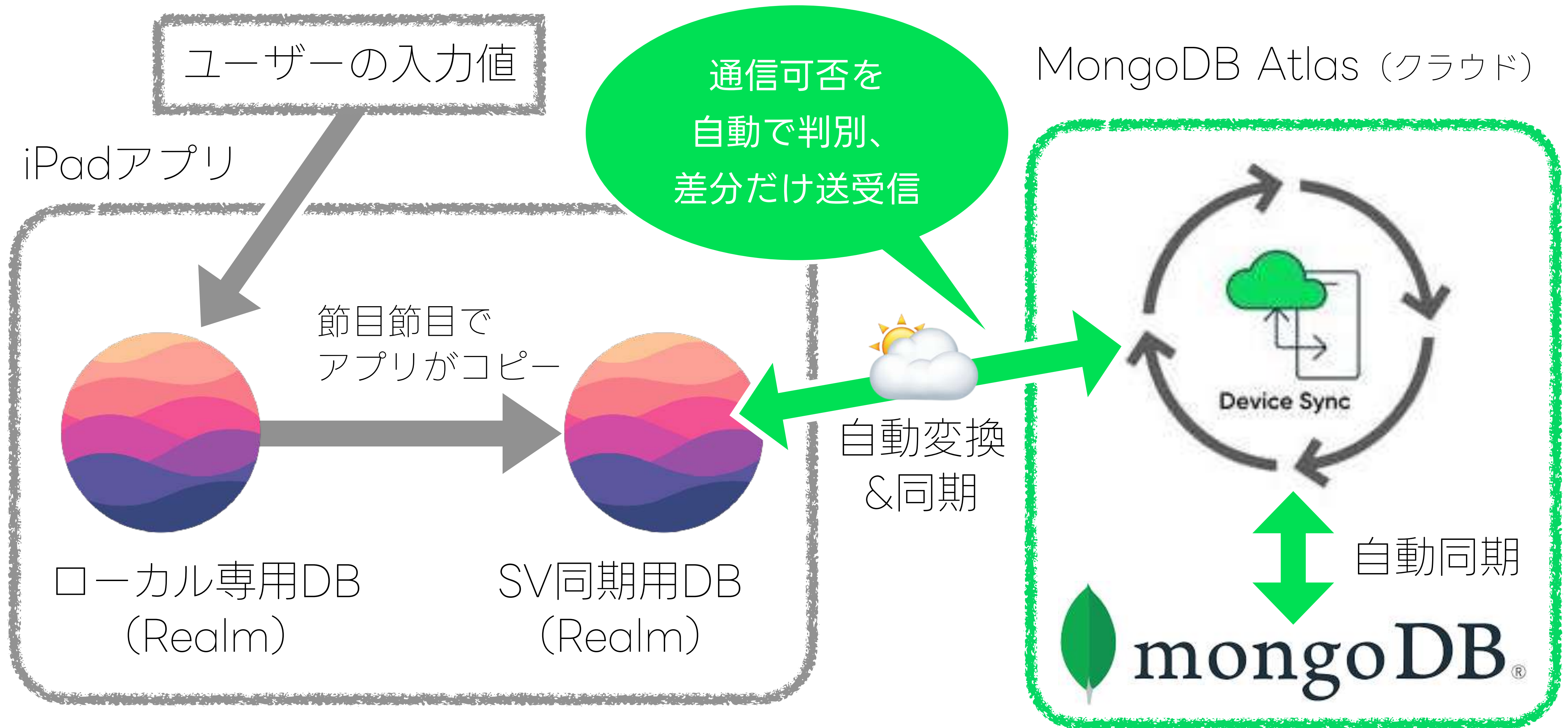
漁業者支援システム「トリトンの矛」iPad向けアプリケーションは、船舶に搭載されたIoT機器からの船舶の航跡情報をもとにした操業情報を参照するための電子操業日誌です。さらに、お役立ち機能（潮流や三枚潮の表示）で漁業者の出漁判断のサポートをします。

■操業情報機能は、リスト表示とカレンダー表示から選択可能で、地図上に出港から帰港までの船舶の航跡情報と操業ポイントを表示します。

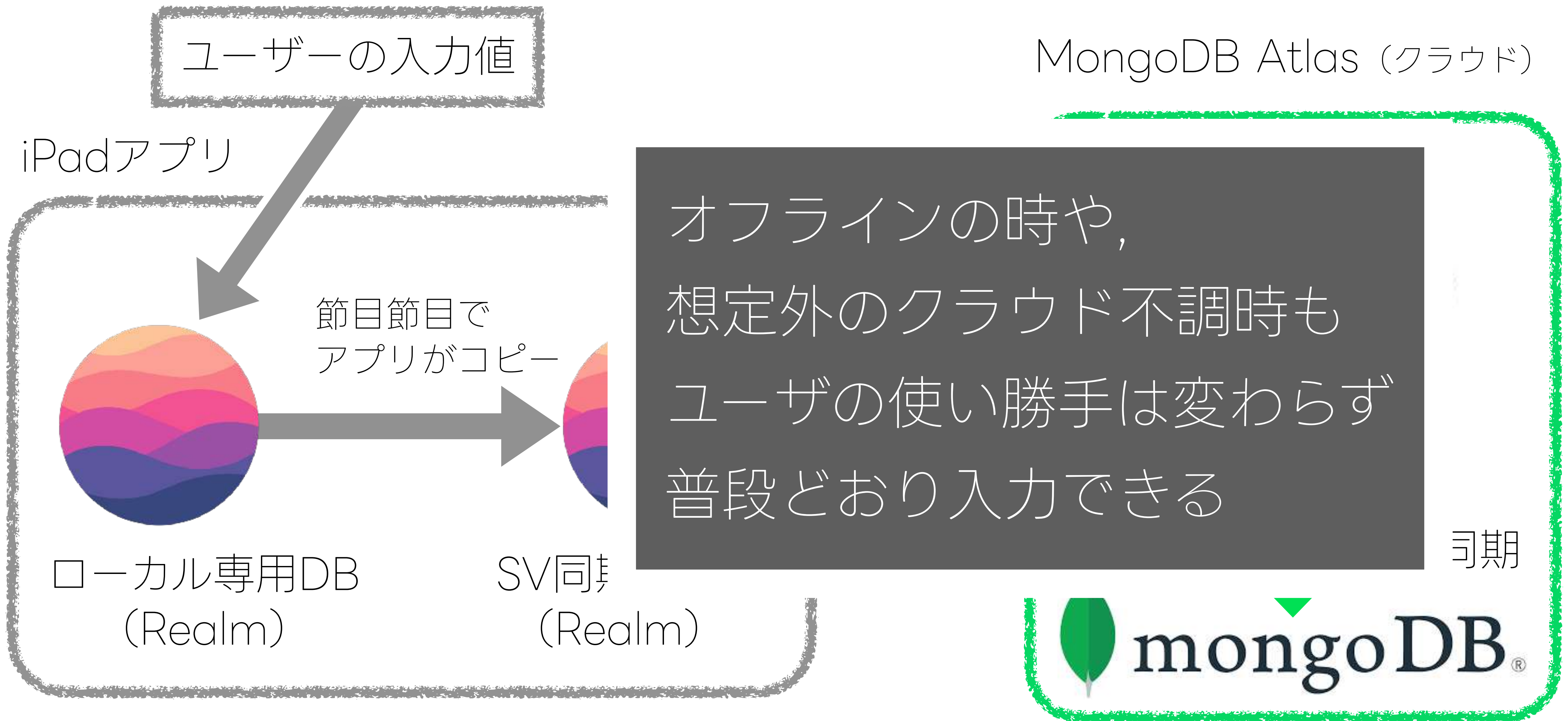
オフラインで使える機能の拡大に努力してきました



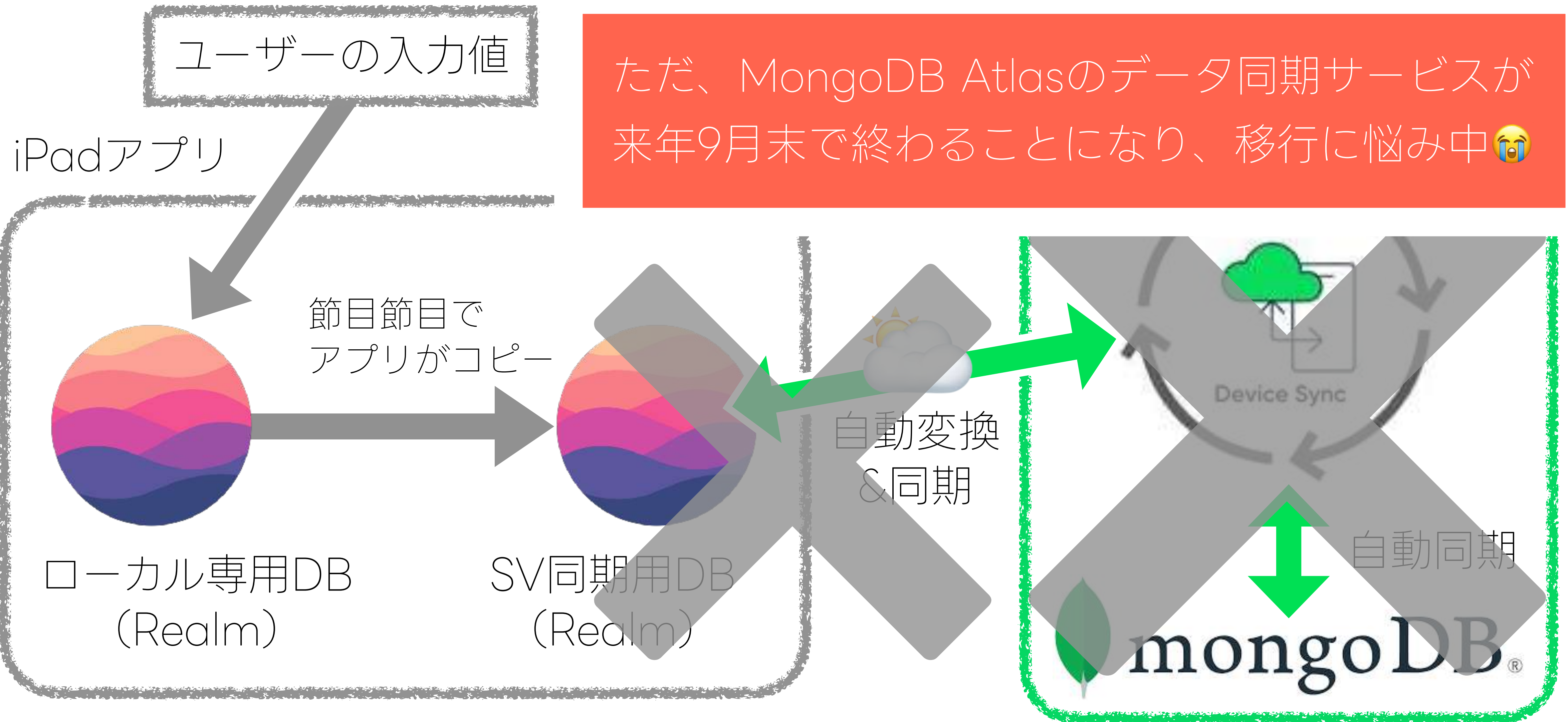
オフラインファーストな入力機能はリリース済 (一部ユーザー向け)



オフラインファーストな入力機能はリリース済（一部ユーザー向け）



オフラインファーストな入力機能はリリース済（一部ユーザー向け）

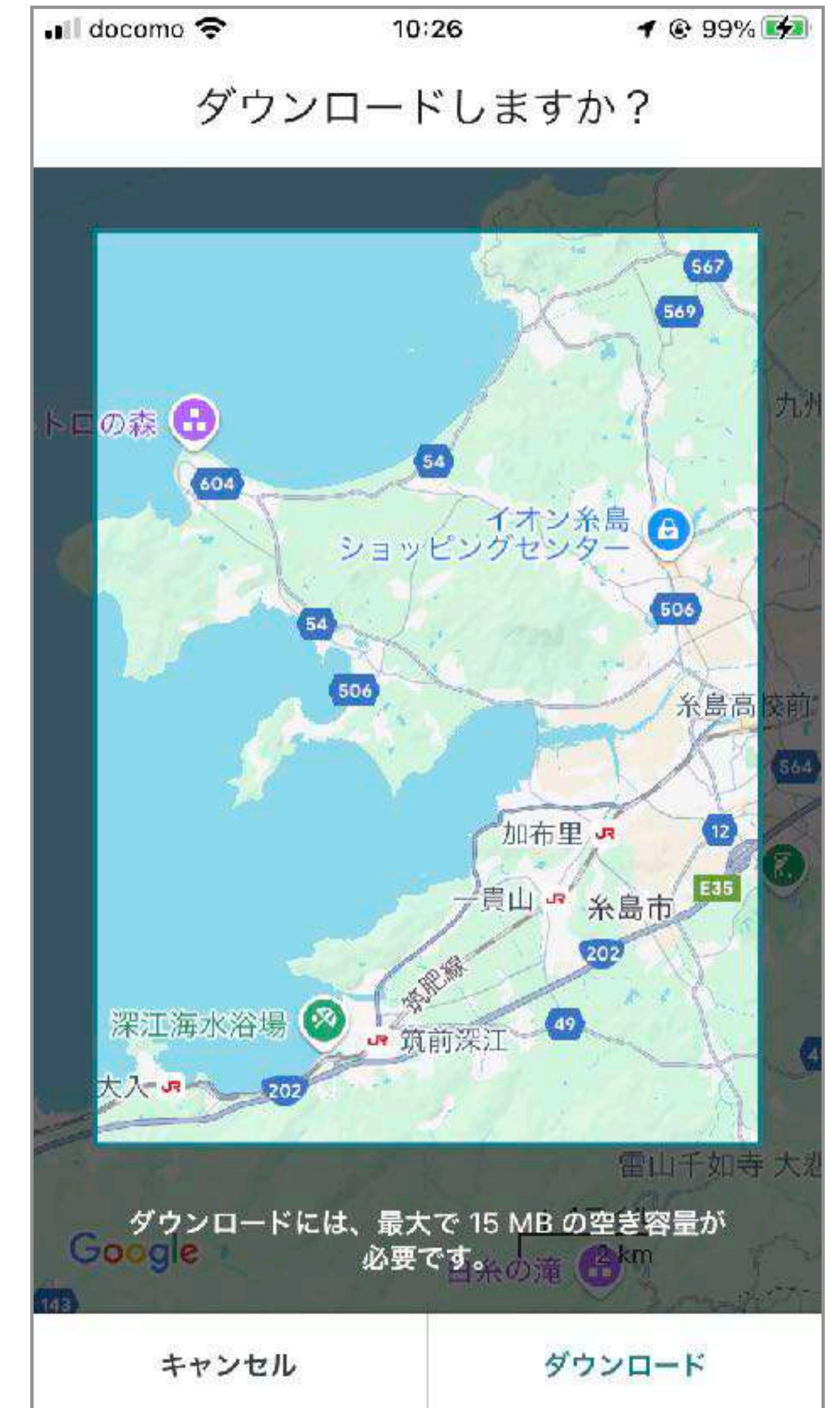


それはともかく、地図機能の向上も大事!!

- 海の上で詳細な「海底地形図」を見たいというニーズ
- データ量が多い
- 沿岸～海上の通信環境では、常にサーバから読み込む地図は厳しい
- 出漁する前、電波の良い所でダウンロードしてもらおう
- 一度ダウンロードしたらずっとオフラインで使えるべき

ところで、既存アプリの「オフラインマップ」使ってますか

Googleマップの場合



ところで、既存アプリの「オフラインマップ」使ってますか

Appleマップの場合



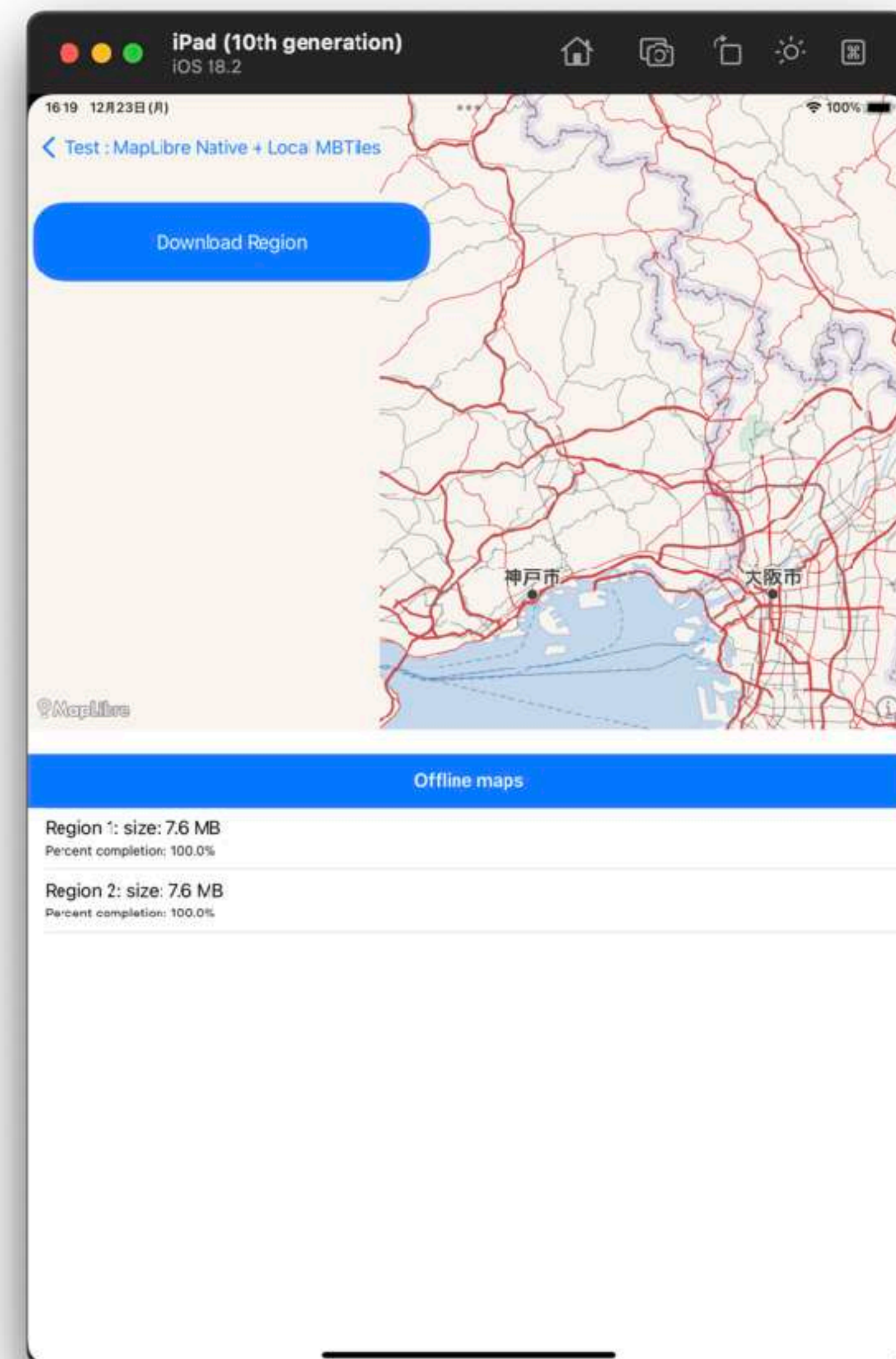
オフラインで見たい範囲を「四隅」で指定するより...

- 地域や主題ごとに分かれた「ファイル」を選ぶ方が良くない？
- 特に海上で、見たい地図を「四隅」で決めるのは無理がある

「オフライン機能」を備えた地図ライブラリもある

今回はMapLibreのサンプルアプリ ➡
をビルドしてみた けど・・・

- ダウンロードするのが「四角で囲われた範囲」になるので、微妙な場所で突然地図がなくなる
- オフラインで必要な範囲をうまく設定するのは難しそう💧・・・特に海上でかつ変更が出たら大変



そこで今回は、地域別ファイルを丸ごとオフライン化

GEOFABRIKの
OpenStreetMapデータ

Consider giving something back in the OSMF's [funding campaign!](#)

GEOFABRIK

Download OpenStreetMap data for this region:

Japan

[\[one level up\]](#)

The OpenStreetMap data files provided on this page are assumed to contain personal data and are subject to the regulations in the European Union.
[Extracts with full metadata](#) are available to

Commonly Used Formats

- [japan-latest.osm.pbf](#), suitable for Osmium, contains all OSM data up to 2024-12-22T20:00:00Z
- [japan-latest-free.shp.zip](#) is not available for

Other Formats and Auxiliary Files

- [japan-latest.osm.bz2](#), yields OSM XML which was last modified 124 days ago. File size: 3.3 GB
- [japan-internal.osm.pbf](#) The history file contains information.

Sub Regions

Click on the region name to see the overview page for that region, or select one of the

Sub Region	Quick Links	
	.osm.pbf	.shp.zip
Chūbu region	[.osm.pbf] (425 MB)	[.shp.zip]
Chūgoku region	[.osm.pbf] (159 MB)	[.shp.zip]
Hokkaidō	[.osm.pbf] (171 MB)	[.shp.zip]
Kansai region (a.k.a. Kinki region)	[.osm.pbf] (280 MB)	[.shp.zip]
Kantō region	[.osm.pbf] (395 MB)	[.shp.zip]
Kyūshū	[.osm.pbf] (234 MB)	[.shp.zip]
Shikoku	[.osm.pbf] (77 MB)	[.shp.zip]
Tōhoku region	[.osm.pbf] (275 MB)	[.shp.zip]



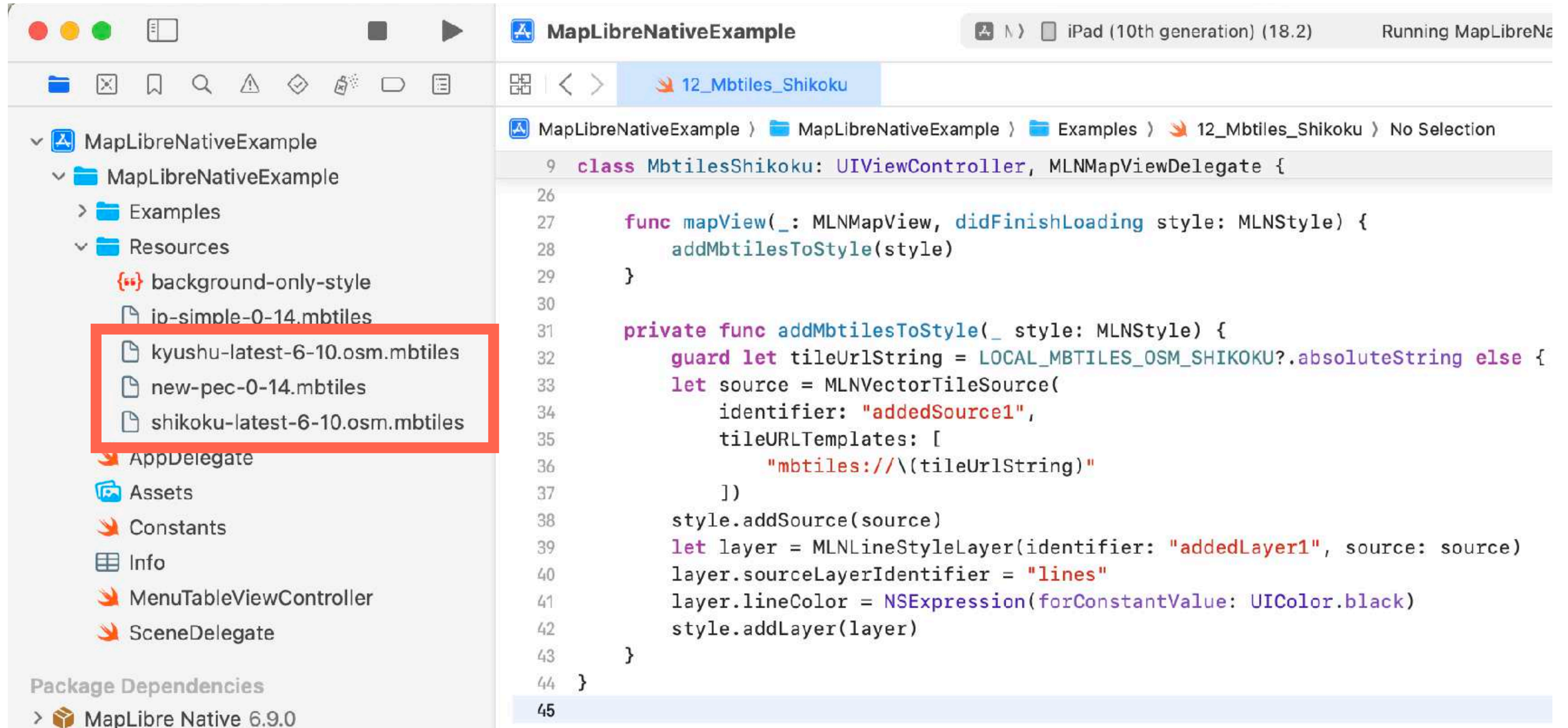
🇬🇧 Not what you were looking for? We're a consulting and software development company based in Karlsruhe, Germany specializing in GIS services. We're happy to provide custom processing, server setup and

ダウンロードしたデータの変換コマンド

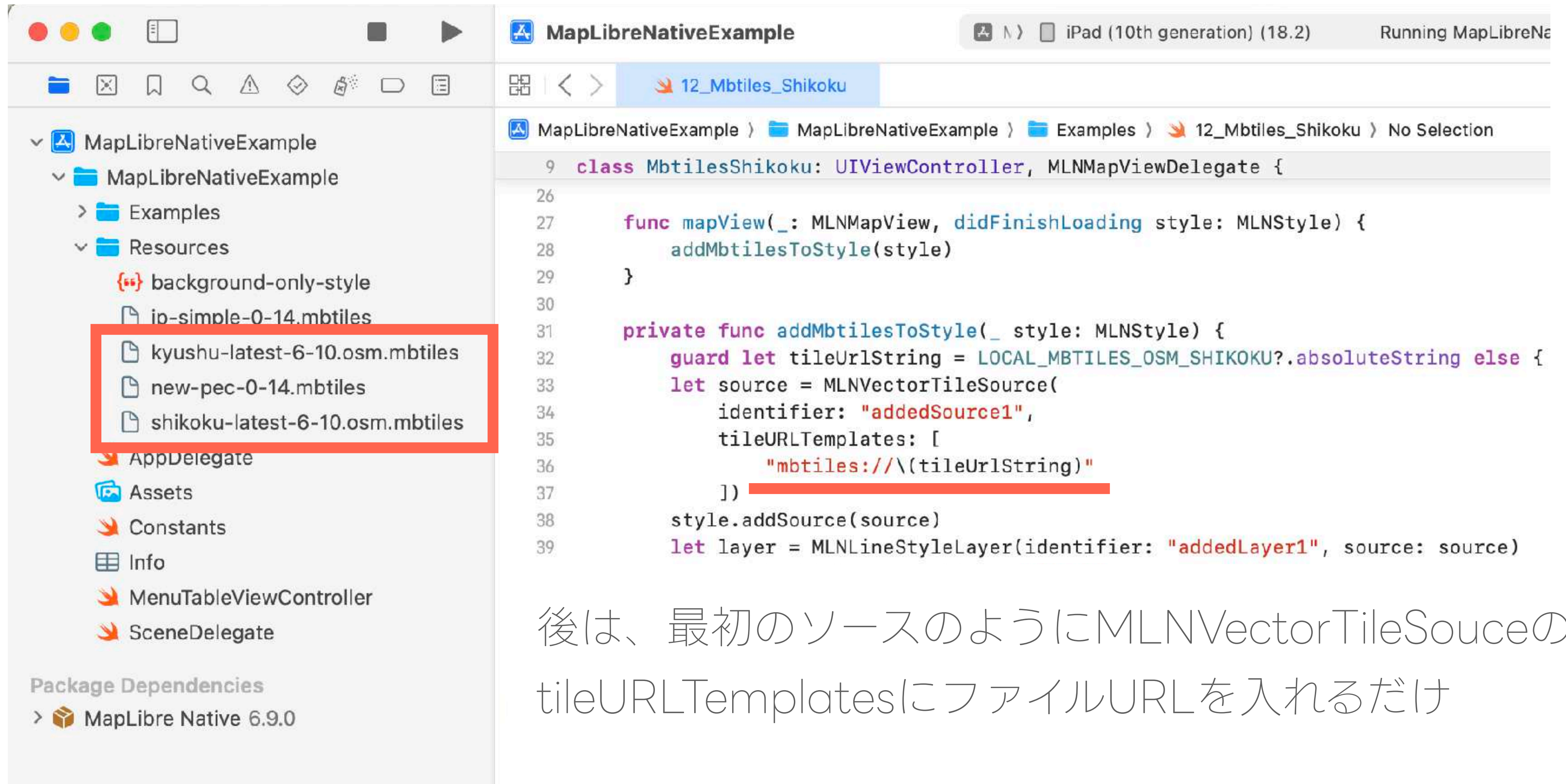
```
$ ogr2ogr -f MBTILES OUTPUT.mbtiles \  
-dsco MINZOOM=6 -dsco MAXZOOM=10 \  
"shikoku.osm.pbf"
```

- ・使うコマンドラインツール ogr2ogr のインストールは
macOSの場合 \$ brew install gdal
- ・ズームレベルの設定によって出力サイズ（と処理時間）が変わる
上記設定だと四国が22MB（元：77MB）、九州は59MB（元：234MB）

出力されたMbtilesファイルをアプリのソースに入れる



出力されたMbtilesファイルをアプリのソースに入れる

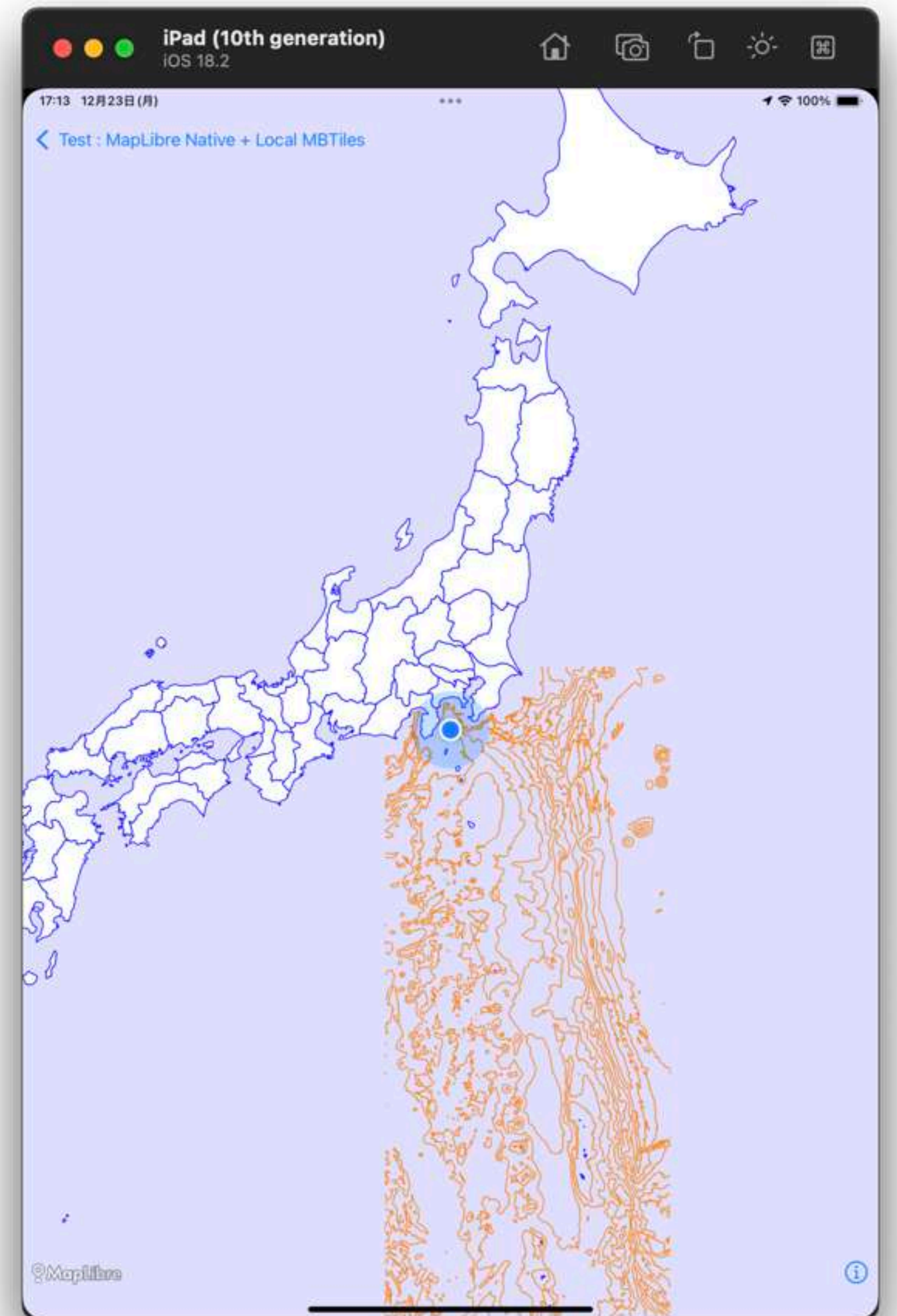


The screenshot displays the Xcode interface for the 'MapLibreNativeExample' project. On the left, the project navigator shows the 'Resources' folder containing several mbtiles files, with 'shikoku-latest-6-10.osm.mbtiles' highlighted. On the right, the code editor shows the implementation of the 'MbtilesShikoku' class, which inherits from 'UIViewController' and implements 'MLNMapViewDelegate'. The code defines a 'mapView' method that calls 'addMbtilesToStyle' and a private 'addMbtilesToStyle' method that adds a vector tile source to the map style. The tile source is configured with a URL template that points to the 'shikoku-latest-6-10.osm.mbtiles' file.

```
9 class MbtilesShikoku: UIViewController, MLNMapViewDelegate {
26
27     func mapView(_: MLNMapView, didFinishLoading style: MLNStyle) {
28         addMbtilesToStyle(style)
29     }
30
31     private func addMbtilesToStyle(_ style: MLNStyle) {
32         guard let tileUrlString = LOCAL_MBTILES_OSM_SHIKOKU?.absoluteString else {
33             return
34         }
35         let source = MLNVectorTileSource(
36             identifier: "addedSource1",
37             tileURLTemplates: [
38                 "mbtiles://\(tileUrlString)"
39             ])
40         style.addSource(source)
41         let layer = MLNLineStyleLayer(identifier: "addedLayer1", source: source)
```

後は、最初のソースのようにMLNVectorTileSouceのtileURLTemplatesにファイルURLを入れるだけ

- ・ 今後、document などアプリデータの場所でも同様にできるか検証します
(その方が柔軟性が高そう)
- ・ 今回は最初から地図をソースに入れましたが、ダウンロード用のUIでユーザが好きな時に好きな地図をGetできればと思っています



以上、2025年を振り返って

MapLibre Native + iOSで
完全オフライン地図ができた!!

と言えるよう頑張ろうと思います💪

終わり