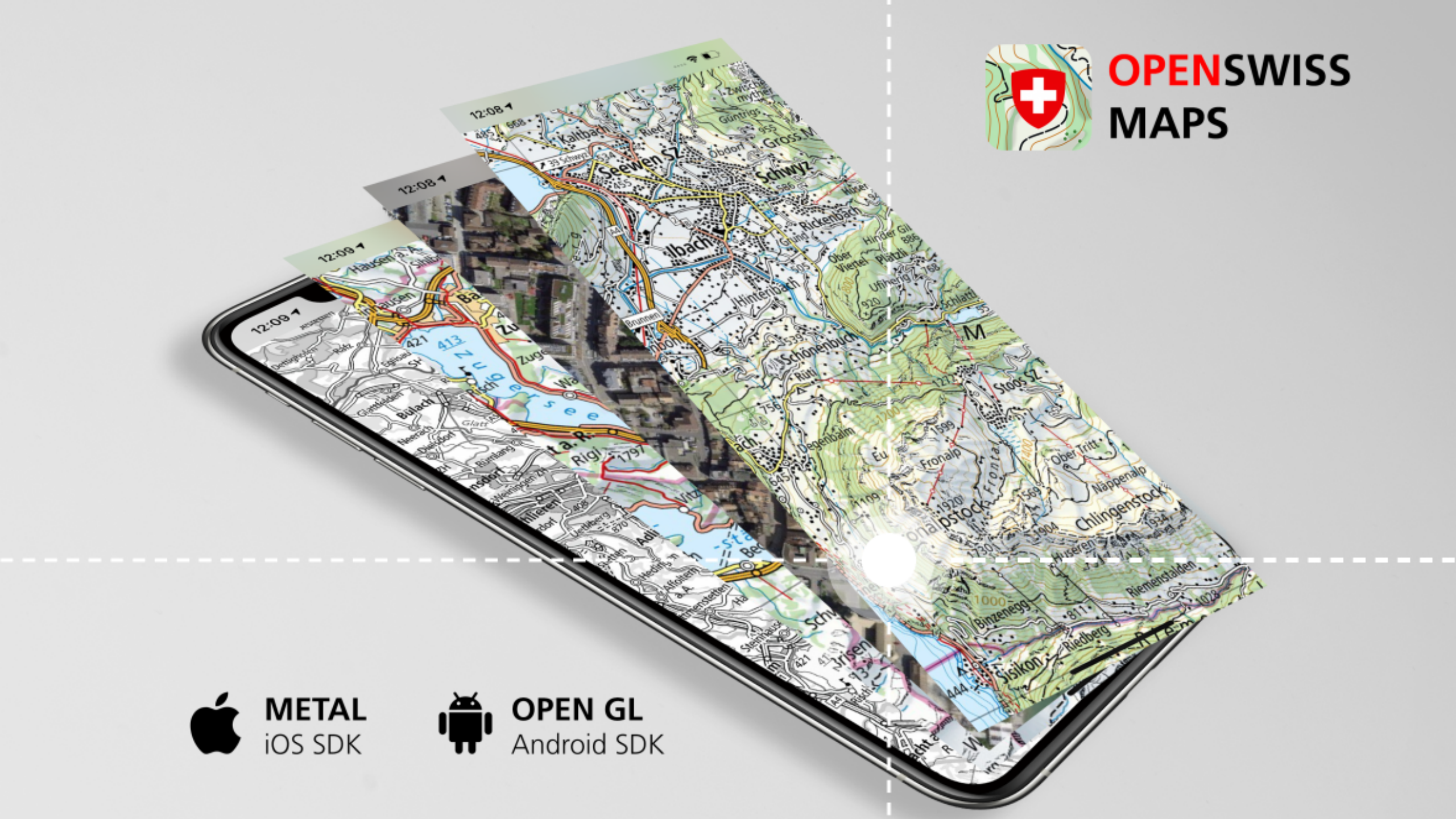
Open Swiss Maps SDK

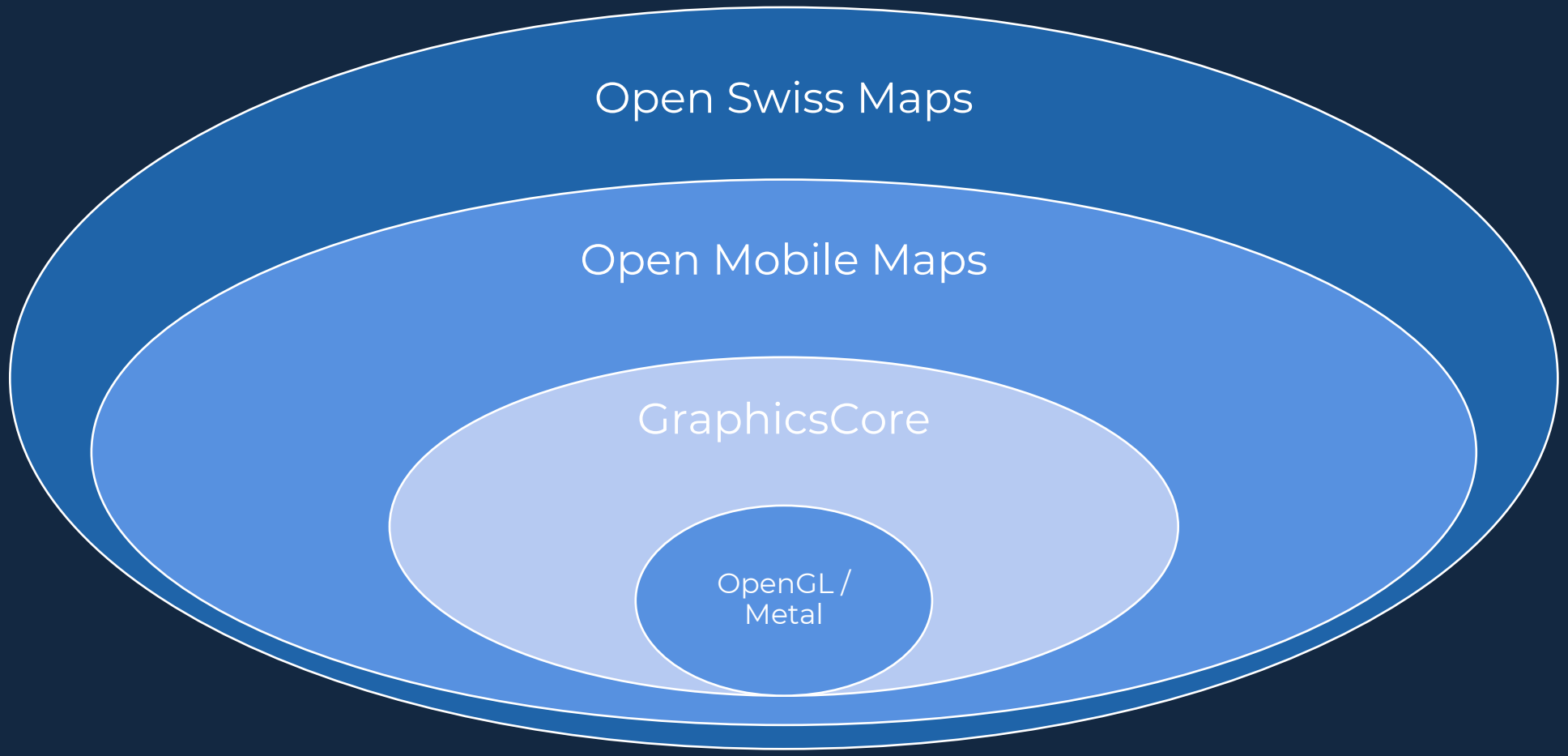
Simon Rösch

ubique


OPENSWISS
MAPS
METAL
iOS SDK
OPEN GL
Android SDK

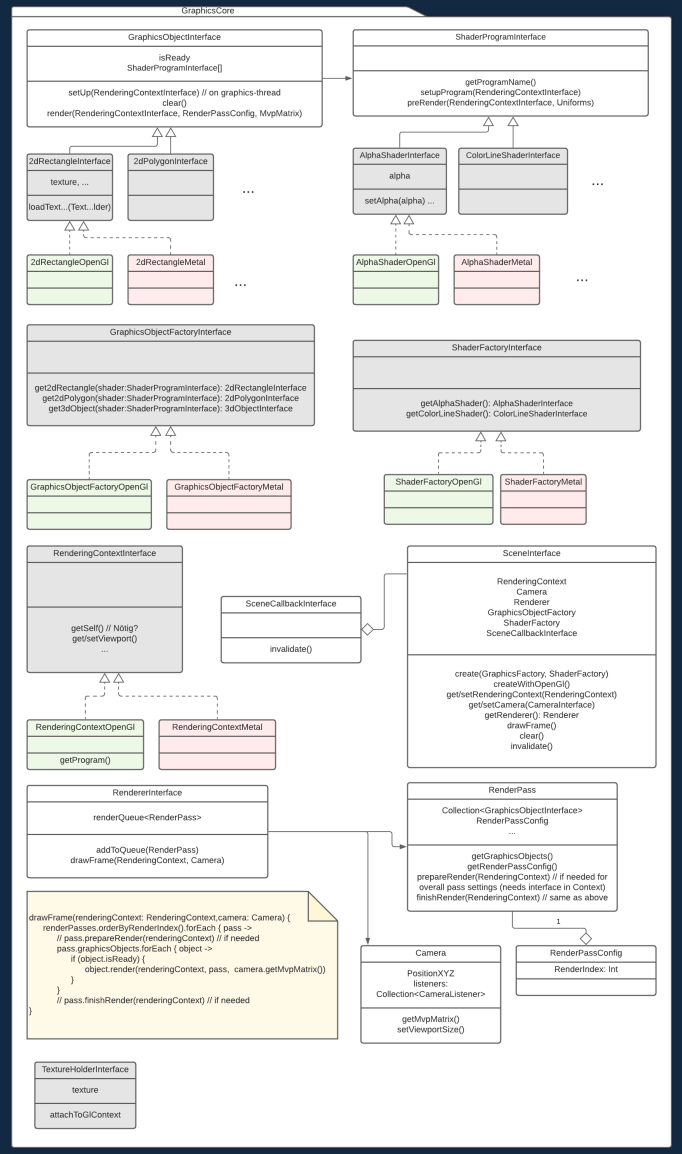
Open Swiss Maps SDK

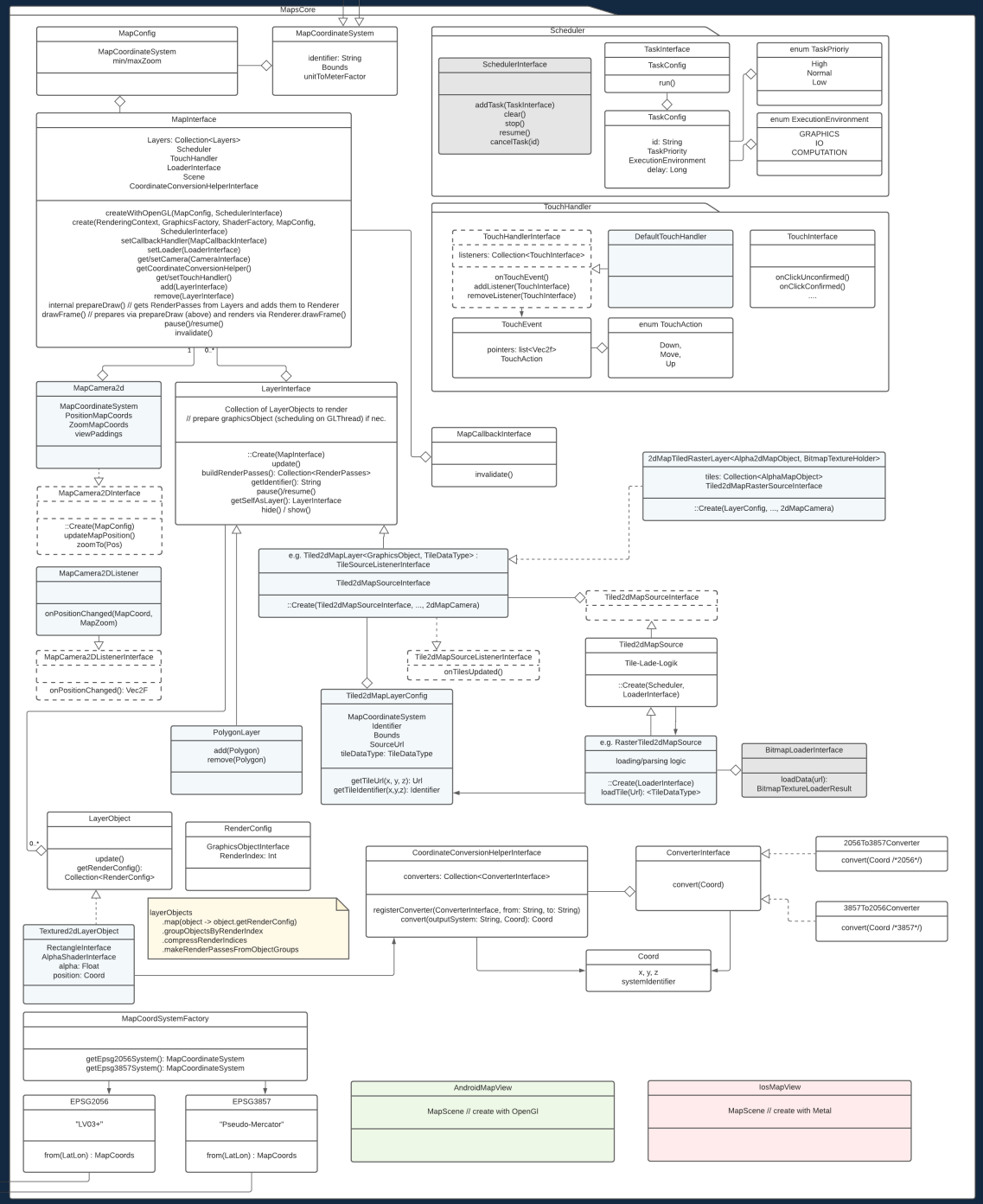
Architecture Overview



Open Swiss Maps SDK

Architecture In Detail





Open Swiss Maps SDK

Why build Open Mobile Maps?

Open Swiss Maps SDK

Why build Open Mobile Maps?

Open Swiss Maps SDK

Open Mobile Maps

- Optimized for mobile performance
- Extentable architecture
- Fully customizable & configurable

Open Mobile Maps: Technologies

- C++ for shared code (iOS/Android)
- OpenGL / Metal
- Djinni with Ubique extensions
- Kotlin / Swift for API

Open Mobile Maps: Features

- Touch Gesture Handling
- Coordinate conversions
- Tiled raster layers
- Polygon layers
- Icon layers
- WMTS capability parser

Open Swiss Maps SDK

How to use Open Swiss Maps

Open Swiss Maps SDK

It is easy to use!

Add Gradle dependency

```
dependencies {
  implementation 'ch.admin.geo.openswissmaps:openswissmaps-android:1.0.0'
}
```

Add SPM dependency

```
dependencies: [
    .package(url: "https://github.com/geoadmin/lib-open-swiss-maps-sdk.git", .upToNextMajor(from: "1.0.0"))
]
```

Initialize SwisstopoMapView

```
val mapView = findViewById<SwisstopoMapView>(R.id.map_view)
mapView.registerLifecycle(lifecycle)
```

Initialize SwisstopoMapView

```
class MapViewController: UIViewController {
  lazy var mapView = SwisstopoMapView()
  override func loadView() {
    view = mapView
```

Change Base Layer

```
mapView.setBaseLayerType(SwisstopoLayerType.SWISSIMAGE)
```

Change Base Layer

```
mapView.setBaseLayerType(type: .PIXELKARTE_GRAUSTUFEN)
```

Open Swiss Maps SDK

Adding more layers

Android

```
val layer = SwisstopoLayerFactory.createSwisstopoTiledRasterLayer(SwisstopoLayerType.DROHNEN, mapView.textureLoader)
layer.setAlpha(0.5)
mapView.addLayer(layer.asLayerInterface())
```

iOS

```
let drohnenLayer = mapView.addSwisstopoLayer(type: .DROHNEN)
drohnenLayer.setAlpha(0.25)
```

Open Swiss Maps SDK

Adjusting the Camera

Animate to position with zoom

```
mapView.getCamera()
        .moveToCenterPositionZoom(Coord(CoordinateSystemIdentifiers.EPSG4326(), 8.543912536386152, 47.37623511643675, 0.0),
        3000.0,
        true)
```

Restrict map bounds

```
mapView.getCamera().setBounds(RectCoord(Coord(CoordinateSystemIdentifiers.EPSG2056(), 2485071.58, 1299941.79, 0.0), Coord(
mapView.getCamera().setMinZoom(5000000.0)
mapView.getCamera().setMaxZoom(250.0)
```

Open Swiss Maps SDK

Adding your own data on top

```
val polygonLayer = PolygonLayerInterface.create()
polygonLayer.add(
    PolygonInfo(
        "Polygon",
        /* Coordinates */,
        /* holes */,
        false,
        Color(1.0f, 0.0f, 0.0f, 1.0f),
        Color(1.0f, 0.4f, 0.4f, 1.0f),
    )
)
polygonLayer.setCallbackHandler(object : PolygonLayerCallbackInterface(){
    override fun onClickConfirmed(polygon: PolygonInfo) {
        // React
    }
})
mapView.addLayer(polygonLayer.asLayerInterface())
```

Open Swiss Maps SDK

Adding your own data on top

```
val iconLayer = IconLayerInterface.create()
val texture = BitmapTextureHolder(/* drawable or bitmap */)
val iconSize = TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, 48f, resources.displayMetrics)
val icon = IconFactory.createIcon(
                                "Icon",
                                coordinate,
                                texture,
                                Vec2F(iconSize, iconSize),
                                IconType.INVARIANT)
iconLayer.add(icon)
iconLayer.setCallbackHandler(object : IconLayerCallbackInterface(){
        override fun onClickConfirmed(icons: ArrayList<IconInfoInterface>): Boolean {
                // React and return true if handled
                return true
        }
})
mapView.addLayer(iconLayer.asLayerInterface())
```

What’s next?

- Several optimizations
- Vector Maps
- Offline Maps
- 3D Maps
- Combination with AR / VR

Open Swiss Maps SDK

More information

swisstopo.ch/sdk

Open Swiss Maps SDK

Questions?

ubique

Thank you for your attention!

ubique

www.ubique.ch