

Chasing Behavioral Parity

Java, Swift and JS interop in Flutter





Who knows MapLibre Flutter?



Hands up!

Gabriel Palmisano

Mobile Developer at EBWorld

MapLibre Flutter maintainer

○ Bari

Where I live

449 km north

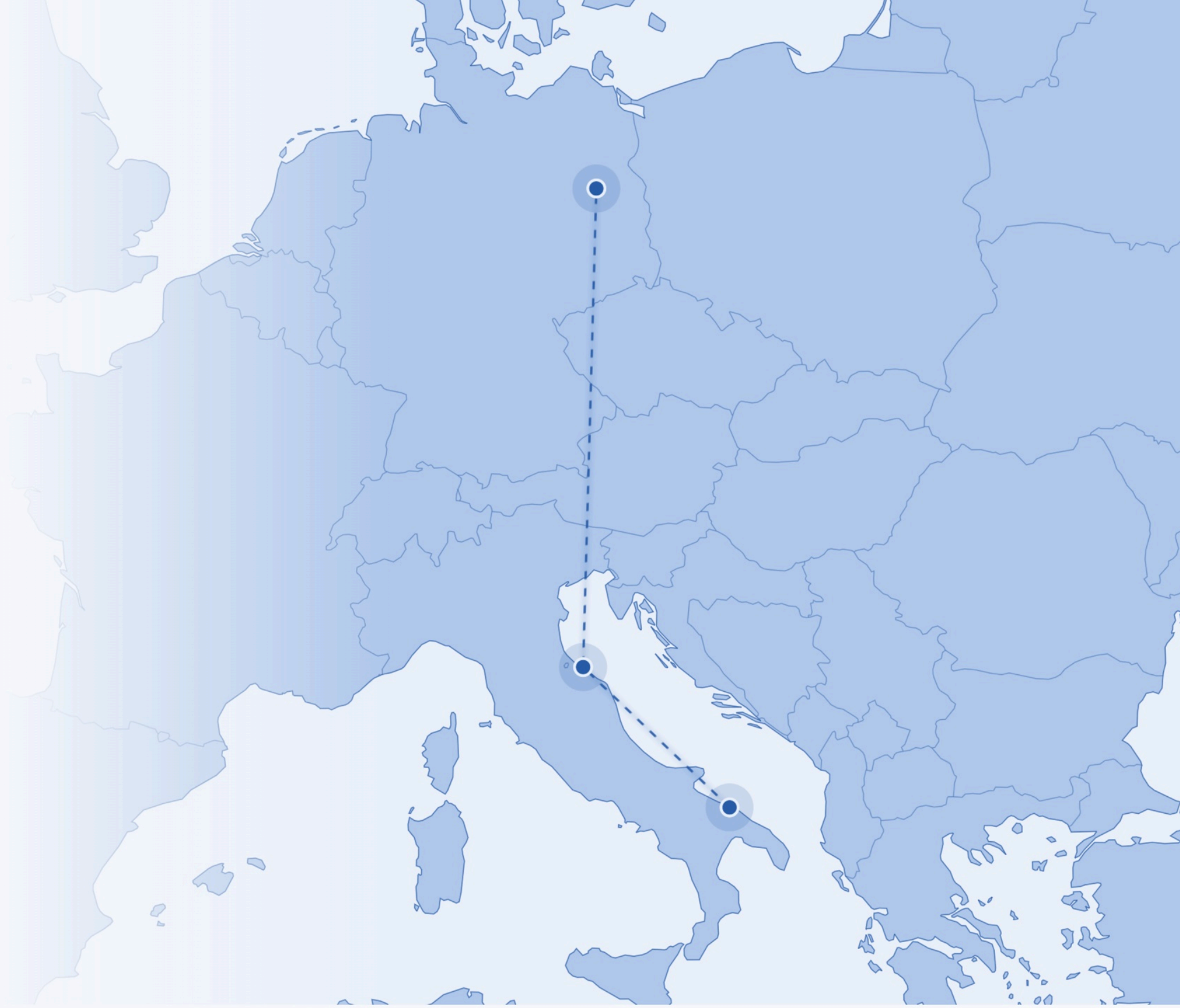
○ Pesaro

EBWorld, where I work

956 km north

● Berlin

flutterCon, today



Same job, drawn differently

`flutter_map`



Drawn by Flutter

Pure Dart, on the canvas.
No native code.

`google_maps_flutter`



Google Maps SDK

Needs a Google API key.
The SDK is closed source.

`mapbox_maps_flutter`



Mapbox engine

Vector maps on the GPU.
Closed, needs a token.

`maplibre_gl`



MapLibre engine

Vector maps on the GPU.
Open source, no key.

A very short story

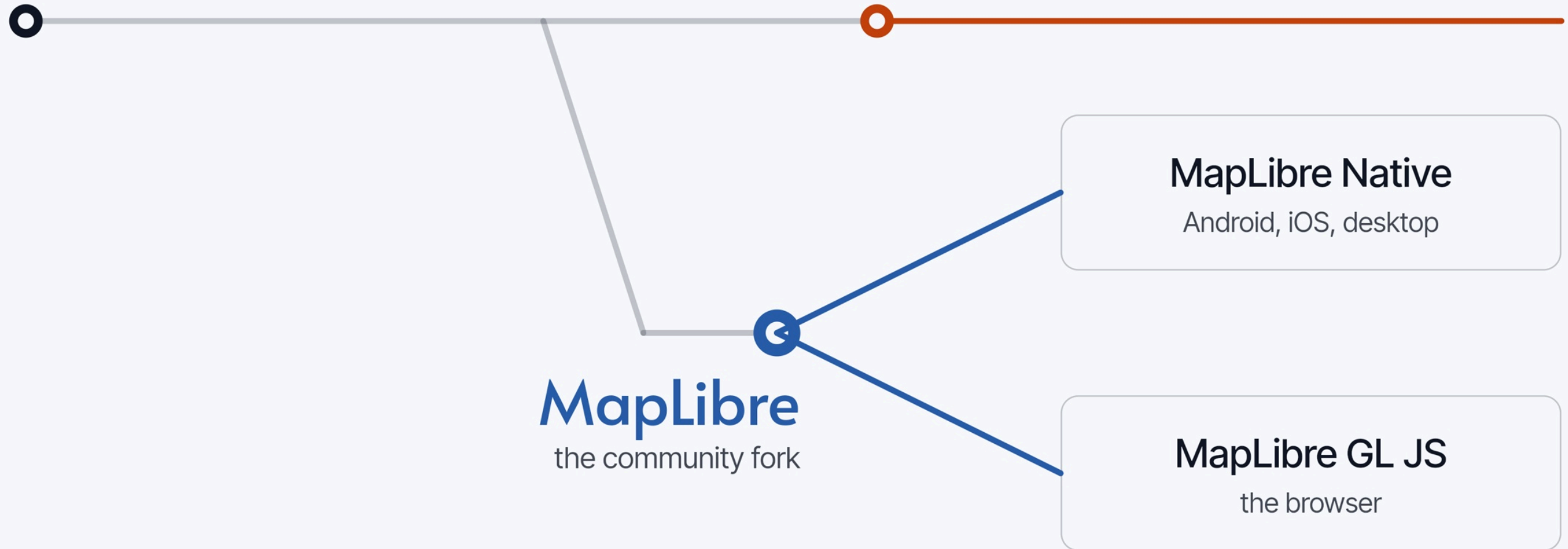
Mapbox
open source

December 2020
new license, closed

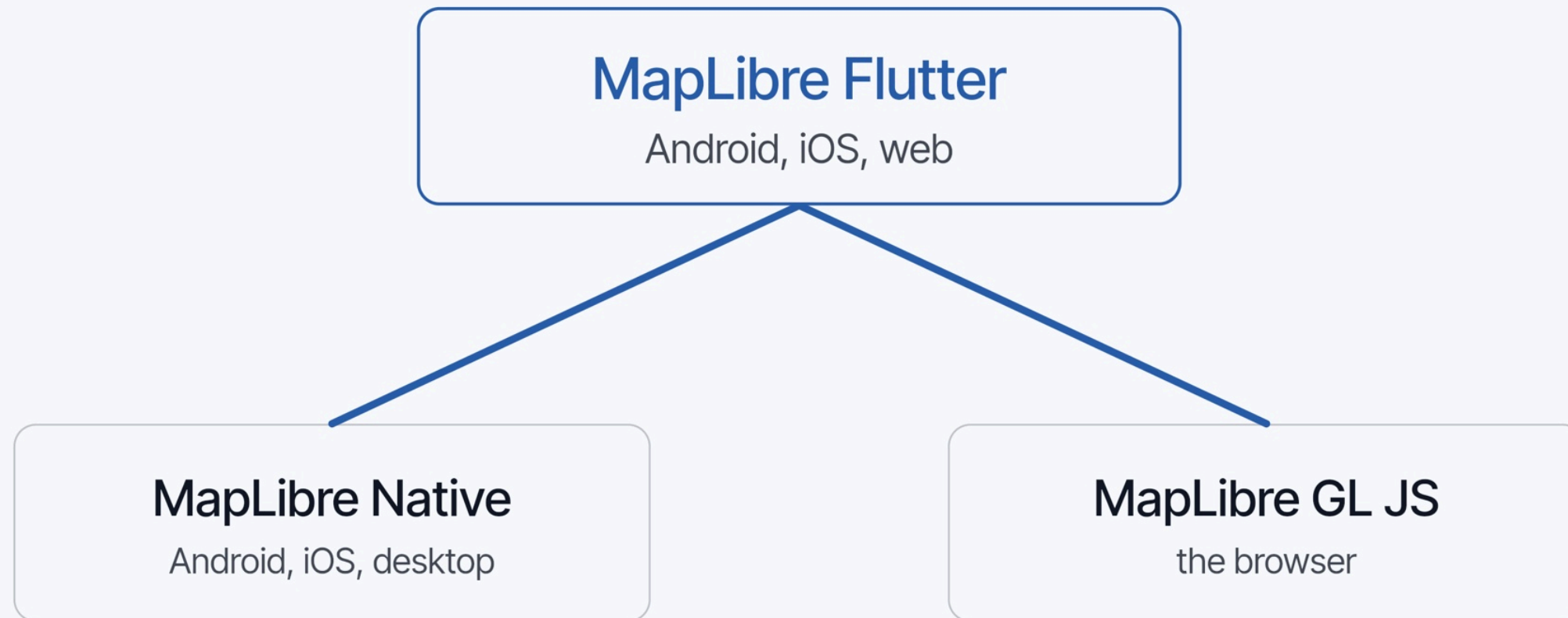
MapLibre
the community fork

MapLibre Native
Android, iOS, desktop

MapLibre GL JS
the browser



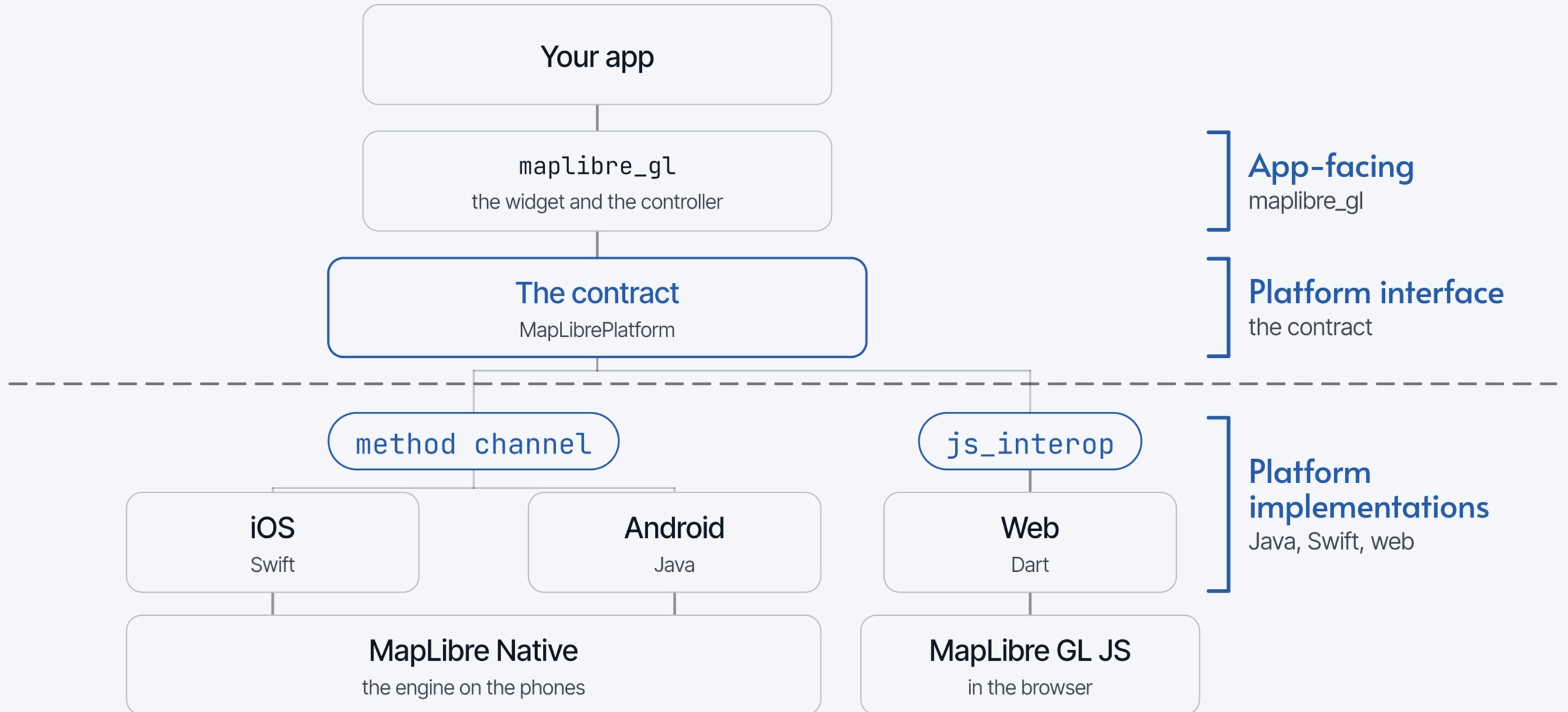
A very short story

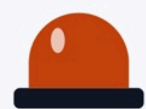


How does it work?

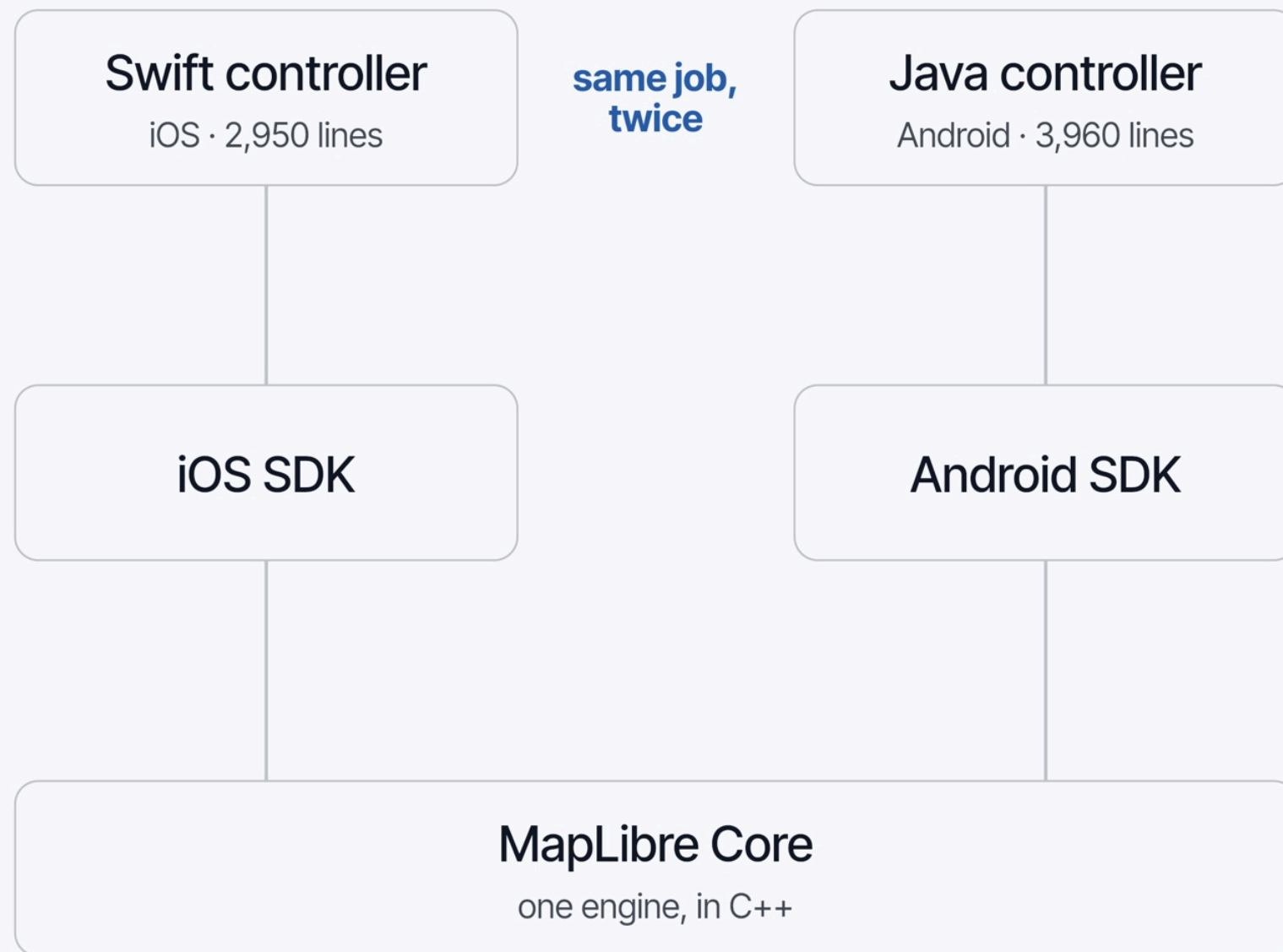
Let's discover it!

A federated architecture





First warning!



Feature state on Android ✓

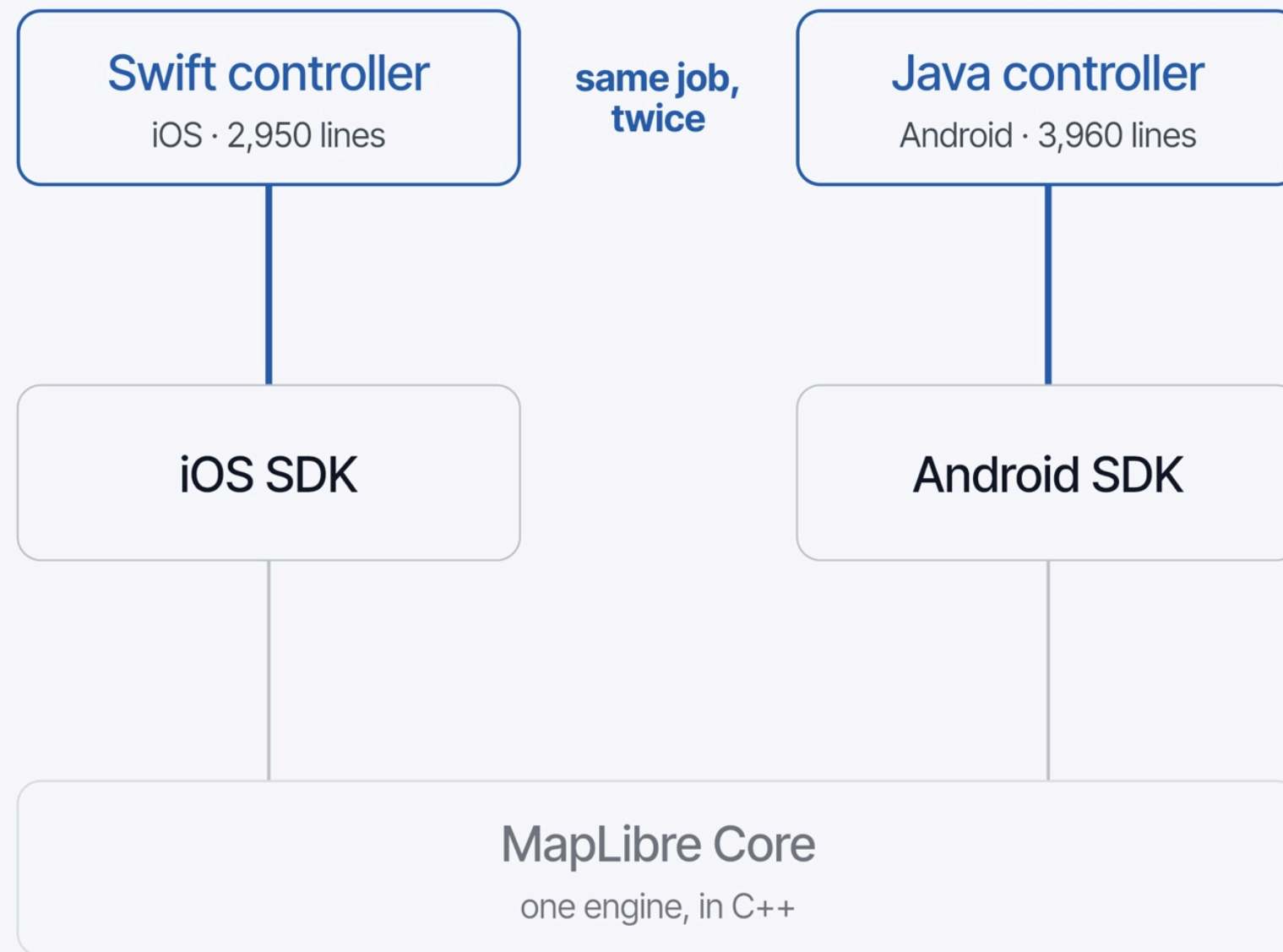


Feature state on iOS ✗

`UnsupportedError`, on iOS

`setFeatureState` is not available on iOS because the MapLibre iOS SDK does not expose the feature state API yet. Feature state is supported on Android and web.

First warning!



MapLibre Flutter never talks to the core. Yet.



Second warning!

MapLibre Native

its maintainers, its roadmap



MapLibre GL JS

other maintainers, other roadmap



Globe on the web ✓

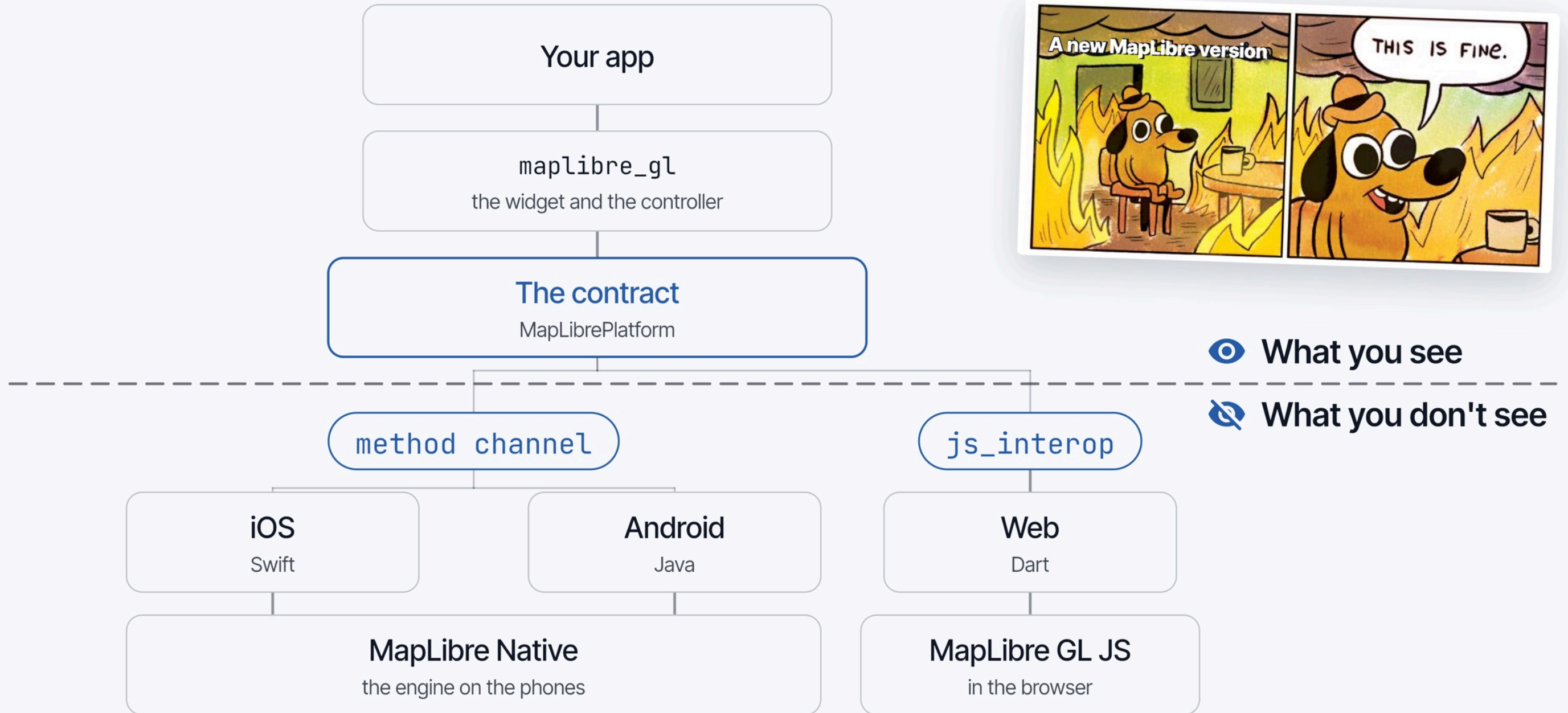


Flat map on the phones ✗

`UnsupportedError`, on Android and iOS

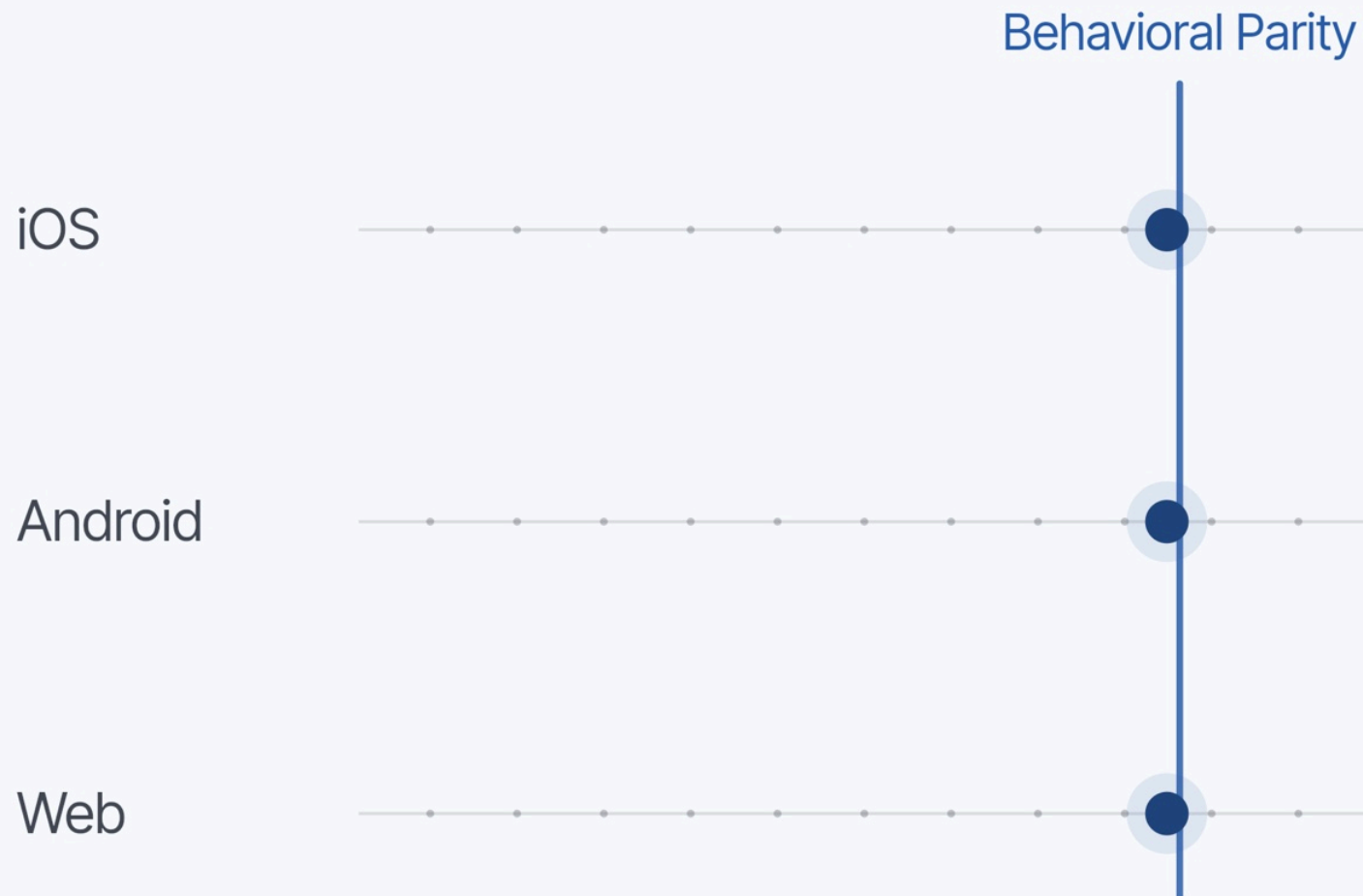
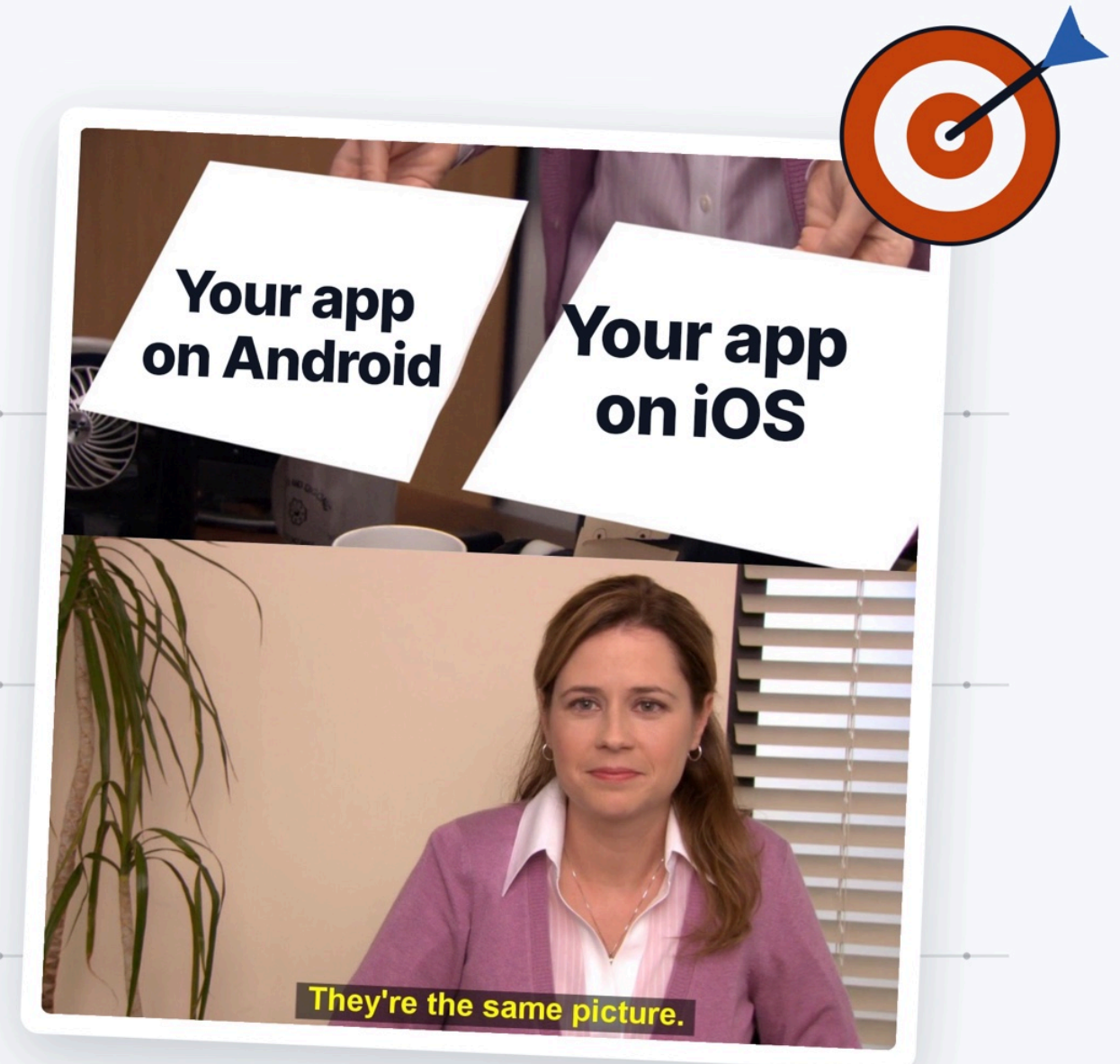
`setProjection` is not available on Android and iOS because MapLibre Native only renders the mercator projection. Projections are supported on web.

On every update



We **chase** the behavioral parity

Upstream moves. We follow. Upstream moves again.



So, how can we reconcile
all of this?

 Keep the three implementations updated

 Bump the MapLibre versions

 Every release.

So, where is the **hard part**?

 No breaking changes

Test. Test. Test.

Every feature, on every platform.

Our CI

 Unit, widget and web tests ✓

 Builds Android, iOS and web ✓

 A real map on a phone ✗

 That one is by hand, today.

Suggestions and help are **very welcome!**

github.com/maplibre/flutter-maplibre-gl

Your API stays the same

MapLibreMapController



```
controller.addGeoJsonSource(...)  
controller.addCircleLayer(...)  
controller.setLayerProperties(...)
```

Okay,
almost never.

Your API stays the same

0.27.0

Actions needed

Web: MapLibre GL JS 6 requires WebGL2, so Safari and iOS before 15 show no map. [Point MapLibreMap.webLibrarySource at a version 5 build to keep those browsers working.](#)

What about a feature that works on native and **not on web**?

map_screen.dart

```
1 controller.setLayerProperties(  
2   'stops',  
3   SymbolLayerProperties(  
4     iconOverlap: 'always',  
5   ),  
6 );
```

`iconOverlap`

Allows for control over whether to show an icon when it overlaps other symbols on the map.

Sdk Support:

basic functionality with js (not on android, ios)



setProjection

UnsupportedError on Android and iOS



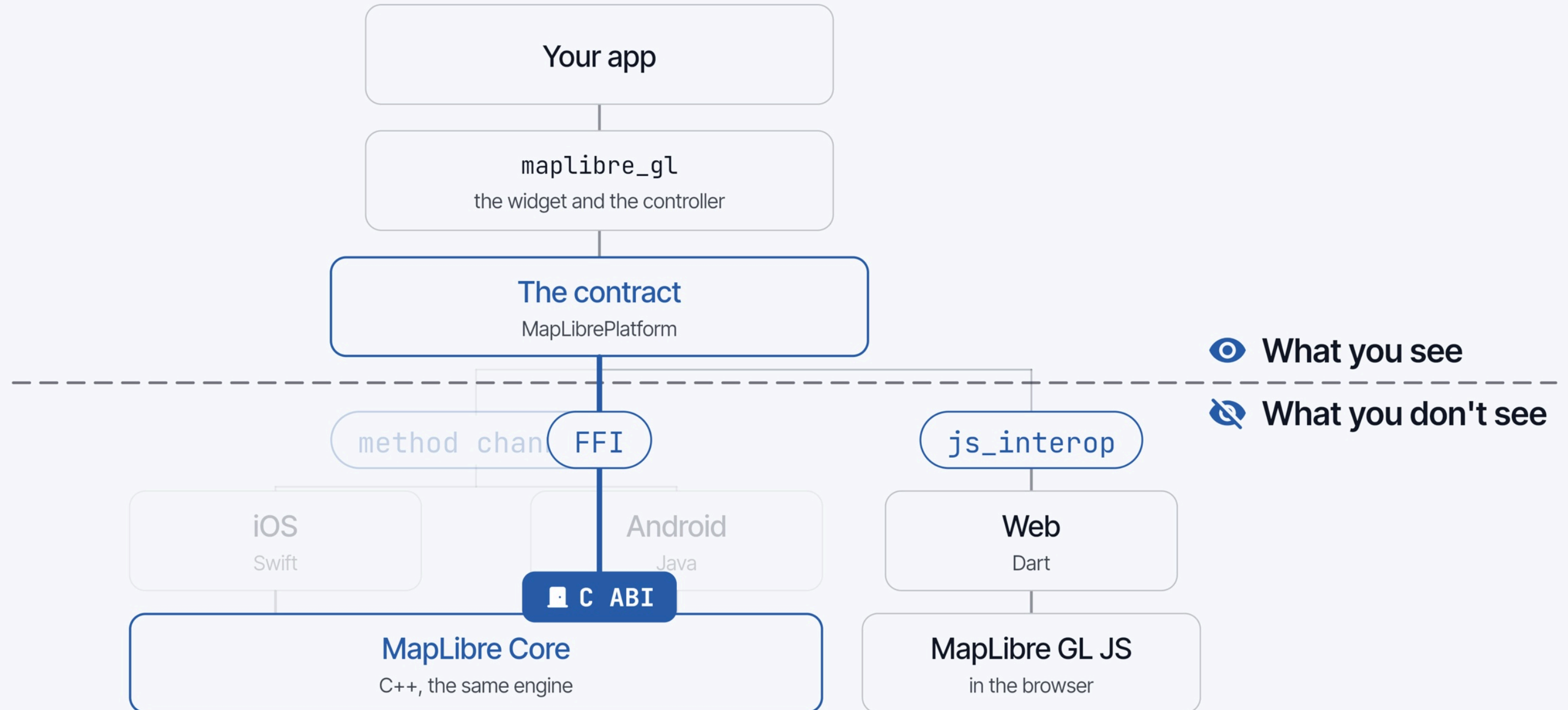
setFeatureState

UnsupportedError on iOS

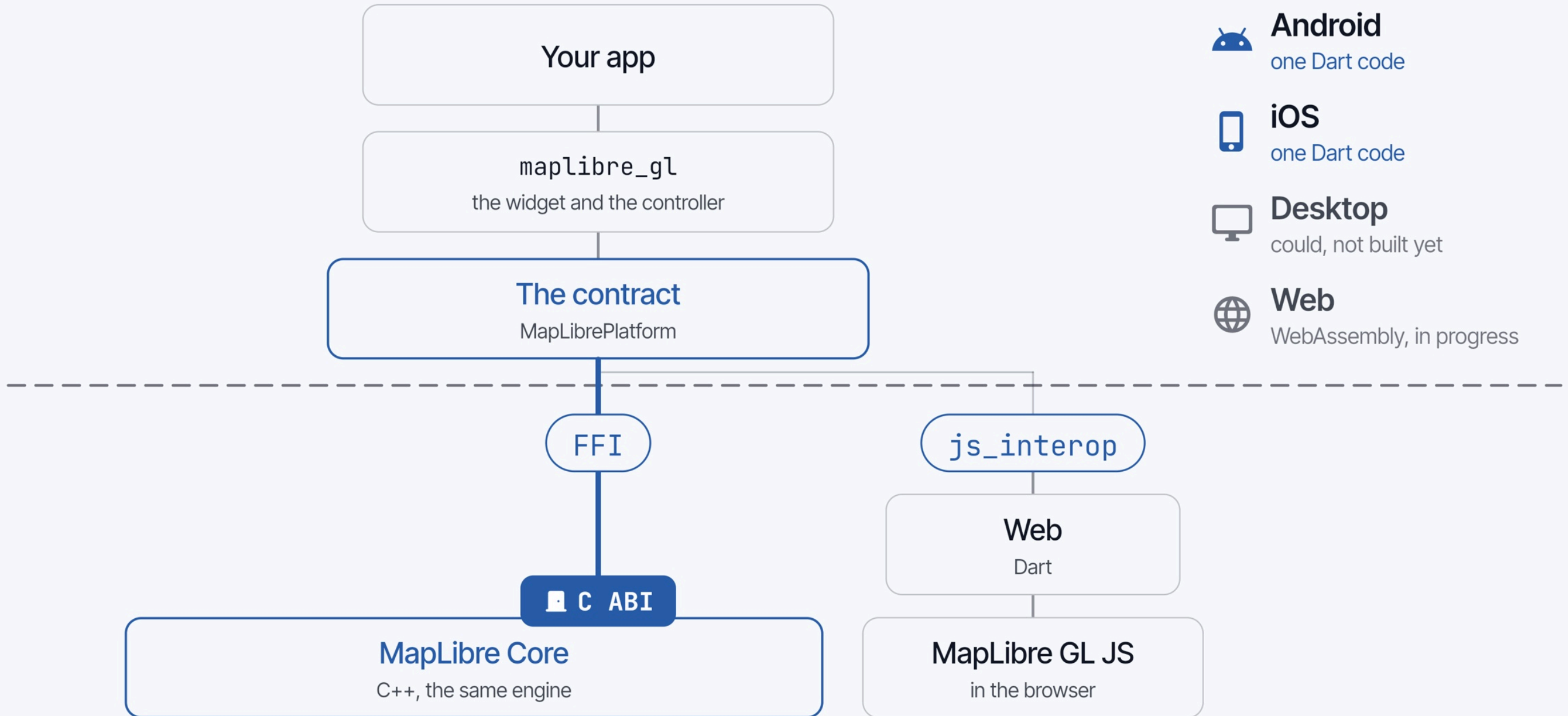
BUT, WAIT!!

We didn't give up!

Straight to the engine



Straight to the engine



How is that possible?

The power of Dart.
It runs everywhere :)

Does it help?

The FFI experiment against the current plugin

10_x

GeoJSON, 20,000 points

1.46 s → 0.15 s

2_x

Loading a style

330 ms → 157 ms

30/s

2,000 live points

9 → 30 updates a second

① One Android phone, a profile build, experimental code. Part of the gain is a different renderer. iOS not measured yet.

So, what did we learn?

- 1 One API is one contract.**
Don't destroy your users' dreams with breaking changes.
- 2 Parity breaks when anything moves.**
Plan for the chase.
- 3 Write down every difference.**
Where your users can see it.
- 4 Keep using the latest Dart interop.**
jnigen, ffigen, build hooks: YouTube and the Dart docs.

Thank you!

Questions?



Try it on [pub.dev](#)



About me