BUILD WITH mapbox

BUILD WITH mapbox

BUILD WITH mapbox

BUILD WITH mapbox

> I am going to literally build this presentation with mapbox!

Escaping the performance Bermuda Triangle

Optimizing integrations with the Mapbox Navigation SDK, Michael Gerasymenko, Delivery Hero

BUILD 2025 | Permanent link: https://gera.cx/mapbox/

Michael Gerasymenko

- Originally from **Ukraine**
- iOS Developer since 2009
- MSc Applied Mathematics
- I have experience working on **SDKs** myself developing Onfido iOS SDK

Where I work: Delivery Hero

- HQ in Berlin
- Part of Logistics, developing the Rider application
- Used in ~70 countries worldwide

Delivery Hero Countries

center

Rider App

- **Source of income** for hundreds of thousands of riders
- Must be extremely reliable, as any quality issue is keeping riders from delivering food and is impacting the company directly

What is performance?

- Different people would see it differently
- In general, **you would know that performance is bad when you see it**
- How to formally define it?

Application Quality Score (AQS)

- Compound metric developed by Delivery Hero in-house (0-100)
- Delivery Hero has 14 apps which are all reporting AQS: **Competition is strong** 🎉

How App Performance is reflected in AQS

- App start time to interactive (10 points)
- All screens time to interactive (10 points)
- Frozen frames (13 points)
- Slow rendering (5 points)
- Crash-free users — not really performance (35 points)
- App Hangs rate (25 points)
- App size (2 points)

Rider App performance challenges

The Rider app is used in extreme conditions:

- Frequently riders are using **old devices**
- The app is running for **8-10 hours** during the work shift
- Riders are working outside, devices are frequently experiencing **extreme heat**
- Computationally intensive map rendering, always-on display on high brightness and continuous network usage contribute to the **battery drain**
- Device has to be **charged** during the shift, which is adding up to the heat

How big is the usage of Mapbox?

- It's the map UI almost **always visible** on the screen
- In many countries we are using Mapbox **Navigation** SDK on top of it

How big is the impact of Mapbox on the AQS?

- App start time to interactive — SDK initialization
- Frozen frames, Slow rendering, App Hangs rate — SDK performance
- Crash-free users — SDK crashes
- App size — SDK size

Bermuda Triangle

Hangs are happening in the app, the **leadership is dissatisfied**, but we cannot do anything as the code is foreign for us.

center

What are the expectations from adding an SDK to the app?

> To be honest, I have yet to see an SDK that improves the performance

Let's talk about ownership: Locus of control

- Definition: **Ownership** means taking responsibility for the outcome, even if the code originates from a third-party SDK.
- Why it matters in SDK integration:
  - You control how the SDK is used, how it’s initialized, and how it interacts with your app.
  - Leadership sees results when your team takes initiative instead of waiting for the SDK provider to fix issues.
- Key mindset shift:
  - From “It’s Mapbox’s problem” → “How can we manage, mitigate, and improve its impact on our app?”

Let's talk about ownership: Learned helplessness

- Problem: Developers often assume “foreign code = out of my hands.”
- Effect: Delays in bug resolution, frozen frames, hangs, crashes — directly impacting AQS.
- Adopt proactive ownership:
    - Profile SDK behavior in your context.
    - Collect telemetry and report actionable insights.
    - Implement workarounds when SDK limitations affect your app.
- Outcome: Reduces blind spots, empowers the team, improves app reliability and performance.

Breaking out of the Bermuda Triangle

- Mapbox is not a **black box**
  - Learn best practices
  - Share detailed bug reports with Mapbox
- **Telemetry** for performance
- Be careful with **new SDK releases**
  - Use Beta releases, analyze the telemetry
- If Mapbox cannot fix it, maybe **you** can?
  - Profile the SDK and share learnings with Mapbox

War stories: Indefinite hang

Warning: This is technical!

As we implemented turn by turn navigation in the Rider App, there was obviously something off.

center

Indefinite hang: AQS

Coincidentally, at the same time, we introduced the Hang Rate to the AQS score... And our app was the worst in the whole Delivery Hero 😳

Indefinite hang: Investigation

We started investigating what could have caused this, and there was a clear correlation with the rollout of the turn by turn navigation.

However, on the developer machines, all is working perfectly well.

center

Investigation

- Telemetry
  - Using Sentry, we analyzed the logs of the hanging code
  - Stack traces highlighted issues with general UI performance
- Sharing issue with Mapbox
- Real-life testing (delivering the food)
  - We can see that in real-life testing the app is getting slow, but it's not possible to debug while delivering food
- Let's review the integration code!

Integration code

Using debugger and code review, we found the source of the issue, and it was not Mapbox (at least to a certain extent)

- iOS Navigation UI is styled with an iOS7-era technology called `UIAppearance`
- Something similar to CSS, where we are telling UIKit how to style elements
- We would like to change this style
- When the navigation header is re-created, we apply styles from Mapbox and our own styles
- Mapbox is propagating header updates every 1s

→ This is not ideal to re-create the view every 1s, but this would not bring the app to a choke

UIAppearance

I've had experience with UIAppearance, therefore, I knew it could be dangerous:

- `UIAppearance` is never clearing obsolete styles
- Hence, the more we are setting styles, the more of the styles are accumulated in the process memory
- However, the memory footprint is not a concern here
- When ANY new view is created, `UIKit` must check if any styles are applying for it
- Has to go through all registered styles
- If we are registering a bunch of styles every 1s, over time creating ANY new view would take more than 500ms

Key takeaways

- SDKs are not magic bullets: Integration matters more than the feature itself.
- Ownership matters: Take responsibility for the SDK’s impact on your app.
- Collaboration improves results
- Learn the SDK’s behavior
- Incremental & careful integration

Thank You!

> Happy to answer your questions, please share your thoughts!