

mndwrk

+

mito.digital

Mobile Dev @ Scale

*LIFE-(AND BUDGET) SAVING TAKEAWAYS FROM
5 YEARS OF EXPERIENCE ON ENTERPRISE
MOBILE PROJECTS*



Gergely Békési

*Mobile Program
Manager*

Mito Digital



INTRO



mito.digital

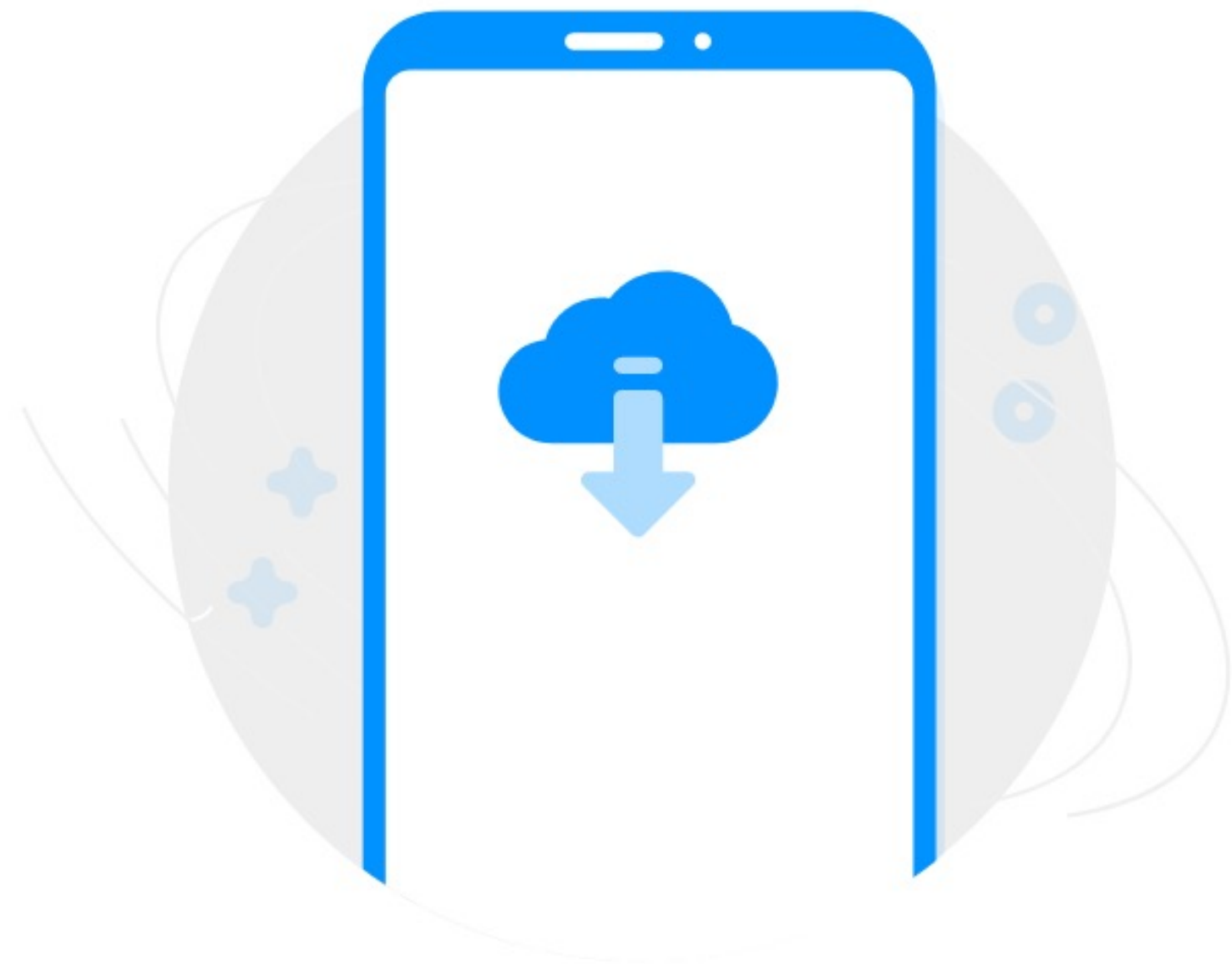
Mito Digital

We create human-centred
interfaces for enterprises of
the digital era.

experience
in
as:



TODAY IN A NUTSHELL
**BIG PROJECT.
BIG RISK.
CLEVER THINGS.**



FORCE UPDATE

Making sure your customers never miss an important app update again.

Force Update

Challenges

- Major API updates - how to manage?
 - Multiple API versions
 - Backwards compatible API structure
- Review of hotfix releases take time
- Delivery of update to clients could easily take multiple days

Force Update

Our approach

- App-level force update feature
- Backend-driven control over the update process
- Different platforms, different versions
- Recommended version vs. Required version
- Benefits
 - Ideal for crisis management (backend-controlled)
 - Fail-safe approach
 - Organisable solution



Source: [MVM Mobiliti Application](#)

Localisation

Don't hard-code your copy mistakes into your application.

Localisation

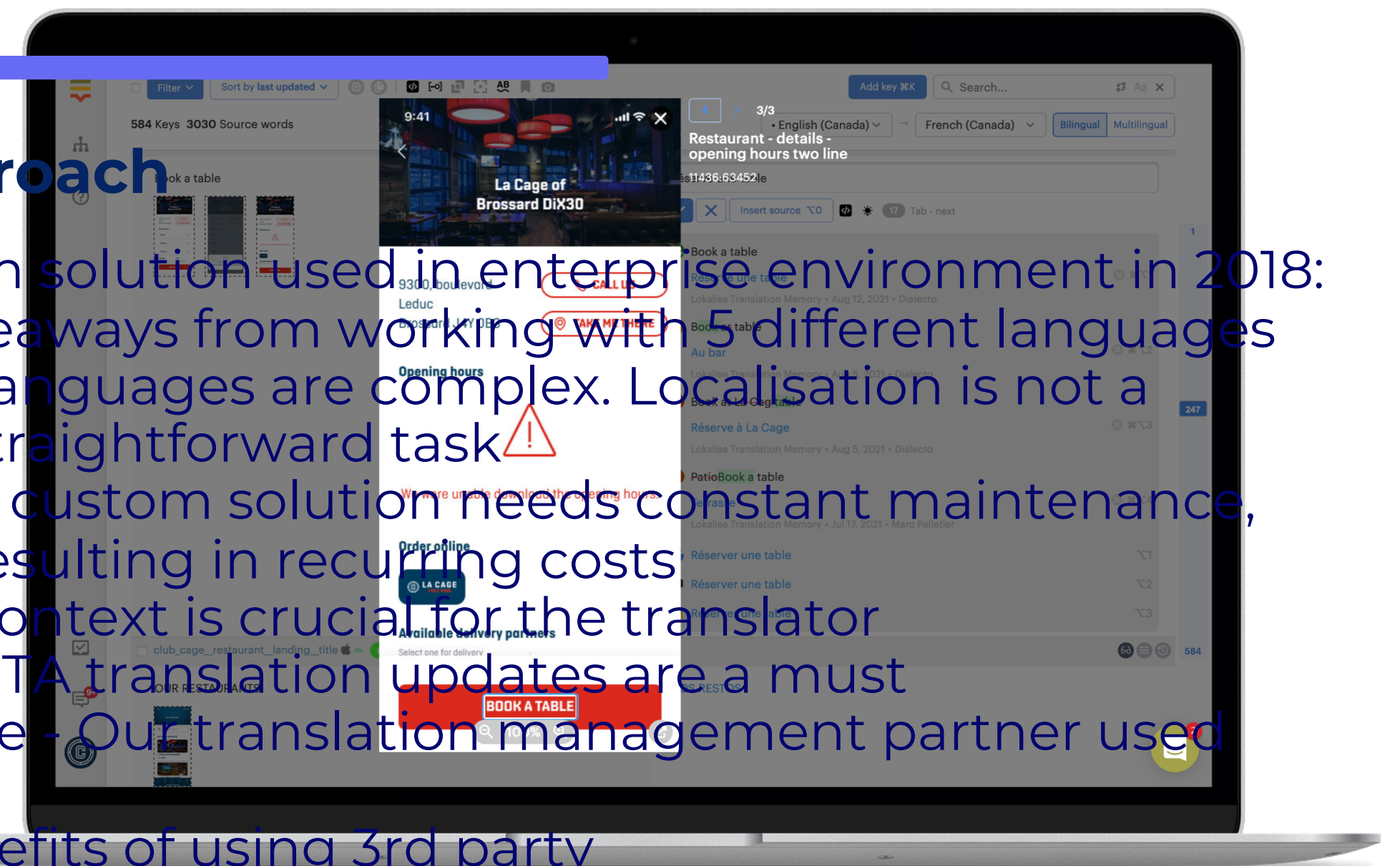
Challenges

- One's first idea is usually Excel, but trust us, it fails when you have 500+ translation terms in different grammatical contexts.
- Text maintenance:
 - What's being translated?
 - What happens if there's a typo?
 - What if the translation doesn't fit on the layout?
- Delivery of text changes: each app update requires development work and time

Localisation

Our approach

- Custom solution used in enterprise environment in 2018:
 - Takeaways from working with 5 different languages
 - Languages are complex. Localisation is not a straightforward task ⚠️
 - A custom solution needs constant maintenance, resulting in recurring costs
 - Context is crucial for the translator
 - OTA translation updates are a must
- Lokalise - Our translation management partner used today
 - Benefits of using 3rd party
 - Saving time and money with Figma integration



Source: [Lokalise's interface](#)

A photograph of three men sitting on a light-colored sofa in a casual indoor setting. The man on the left, with a beard and wearing a light blue hoodie and jeans, is looking down at a smartphone. The man in the center, wearing an orange beanie and a dark shirt, is also looking at a smartphone. The man on the right, wearing a grey hoodie and yellow pants, is looking towards the camera. A semi-transparent dark grey box with white text is overlaid on the center of the image.

Cache and Offline Behaviour

Don't let your users get stuck with a loader at critical moments.

Cache & Offline Behaviour

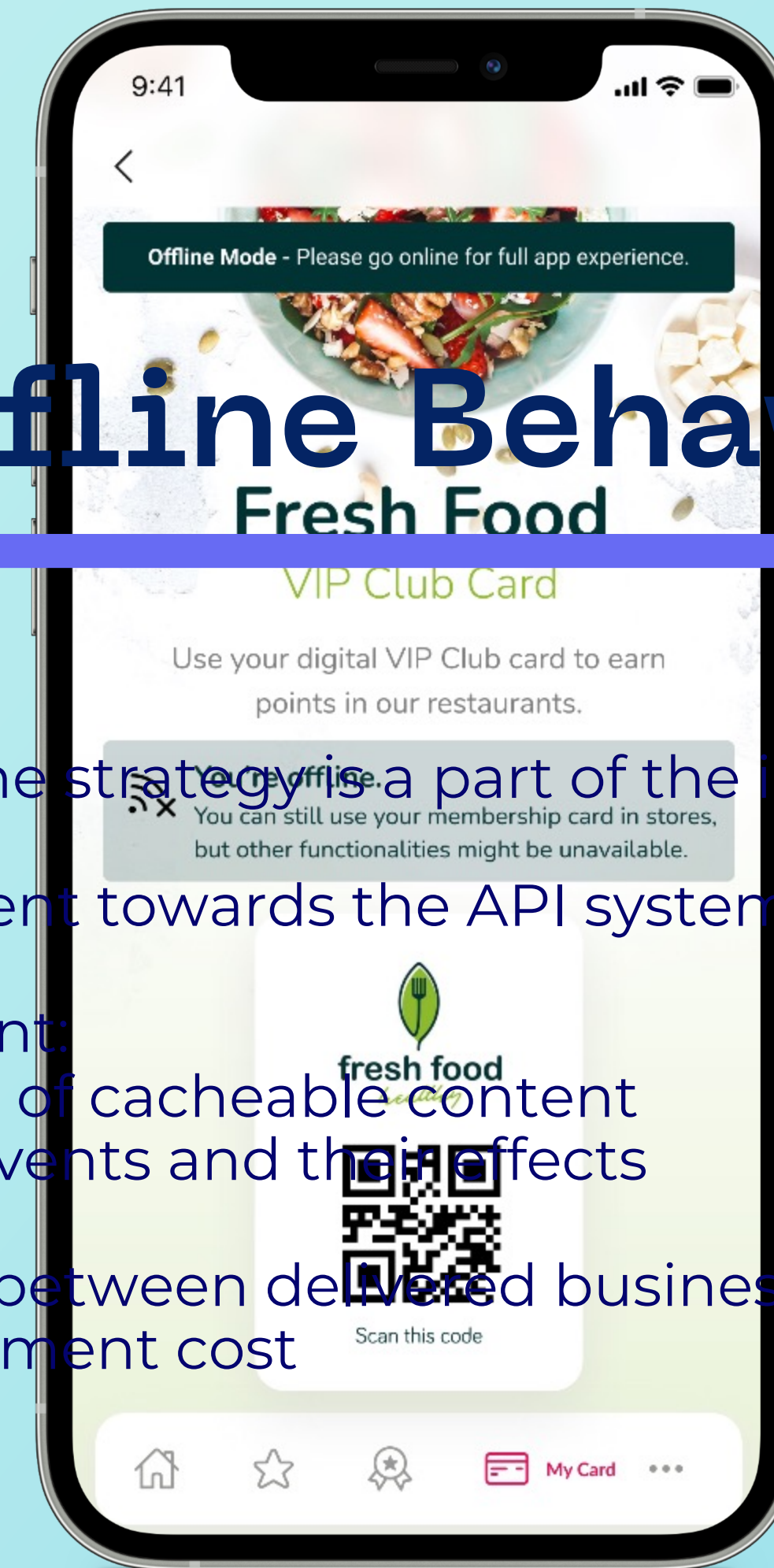
Challenges

- One of the benefits of a mobile application is that it can be used even when the device is offline.
- Caching dynamic data (e.g. API responses) is a non-trivial task. There is a handful of questions you need to consider:
 - What to cache?
 - For how long to cache?
 - What should trigger expiration of data?
- Offline requirements need to be assessed at a feature level to avoid:
 - User frustration originating from inferior customer experience
 - Unnecessary development efforts and costs

Cache & Offline Behaviour

Our approach

- Offline behaviour and cache strategy is a part of the initial design process
- Cacheability is a requirement towards the API system as well
- Caching of dynamic content:
 - Business side: definition of cacheable content
 - Technical side: trigger events and their effects
- Offline behaviour
- Finding the sweet spot between delivered business value and reasonable development cost



Application Size Matters

Respect your customers' storage.

Application Size Matters

Challenges

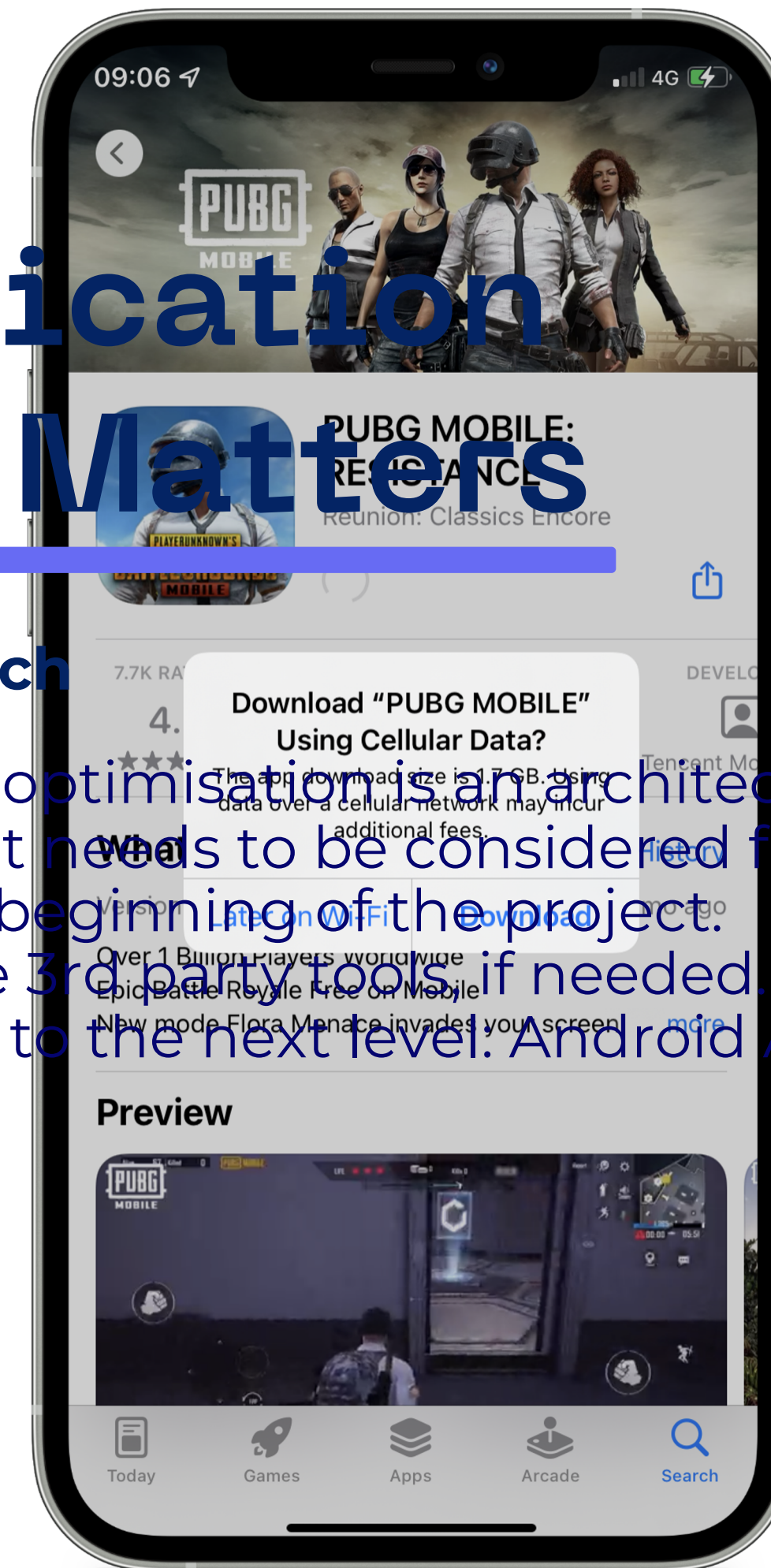
- Using and overusing 3rd party libraries in mobile development
- Finding the sweet spot: reasonable artifact size, without compromising quality and delivery



Application Size Matters

Our approach

- » App size optimisation is an architectural topic that needs to be considered from very the beginning of the project.
- Optimise 3rd party tools, if needed.
- Taking it to the next level: Android App Bundles.



Source: Apple App Store

Remote Control

Manage the features
of your live app.

Remote Control

Challenges

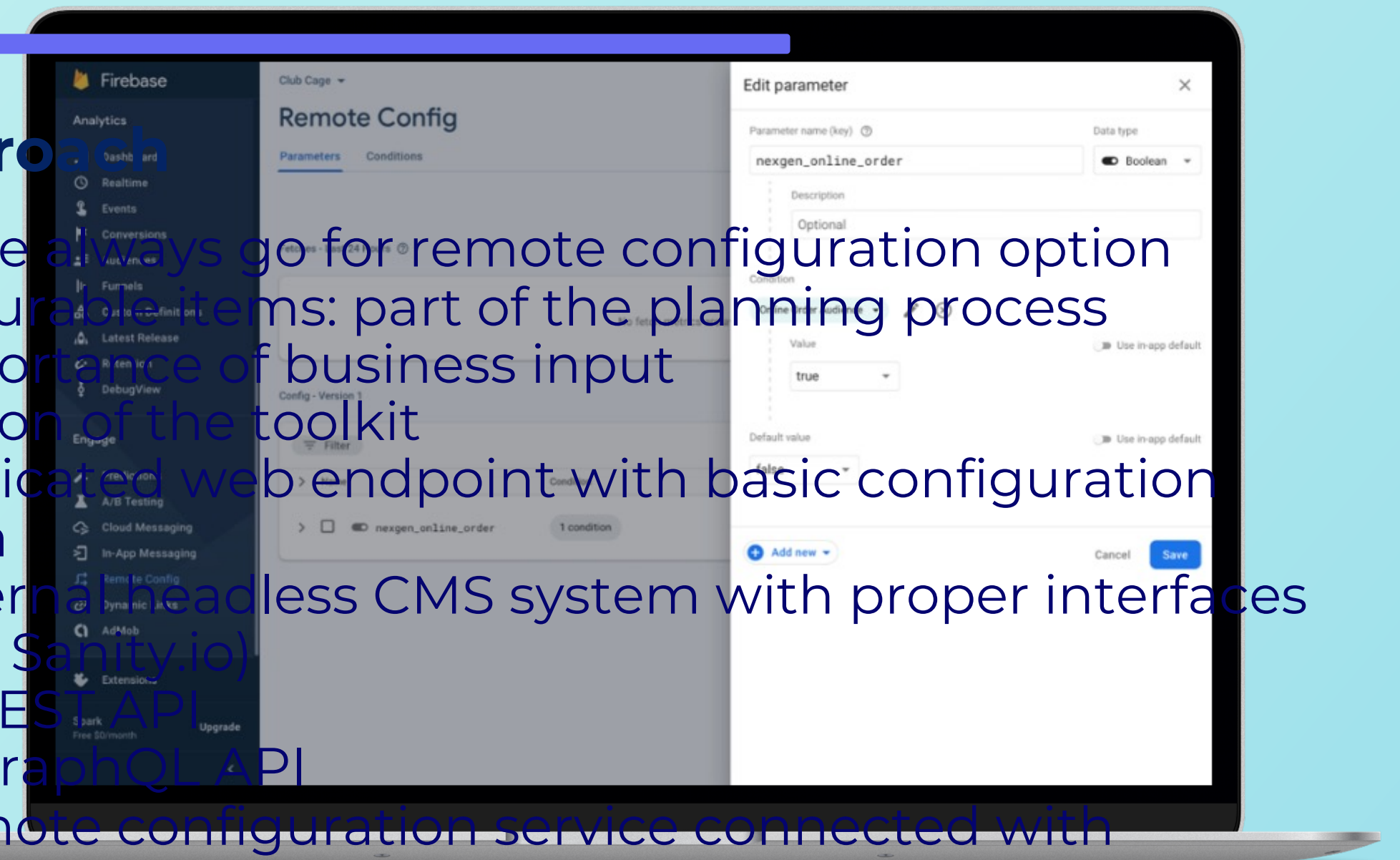
- How do you want to control the app's behaviour?
 - Hard-coded feature flags
 - Dedicated external system to manage remote config and provide OTA updates
- What do you want to control?
 - Too few configurable items: Lack of control
 - Too many configurable items: Cost and time
- Characteristics of the configuration
 - Non-critical updates: Background sync
 - Critical updates: Blocking sync



Remote Control

Our approach

- Why we always go for remote configuration option
- Configurable items: part of the planning process
 - Importance of business input
- Evolution of the toolkit
 - Dedicated web endpoint with basic configuration data
 - External headless CMS system with proper interfaces (e.g. Sanity.io)
 - REST API
 - GraphQL API
 - Remote configuration service connected with Analytics (e.g. Firebase Remote Config)
 - Conditional configuration options
 - A/B tests
 - Experiments, ML toolkit



Deeplink Management

Don't frustrate your users with
unreliable deeplinks.

DeepLink Management

Challenges

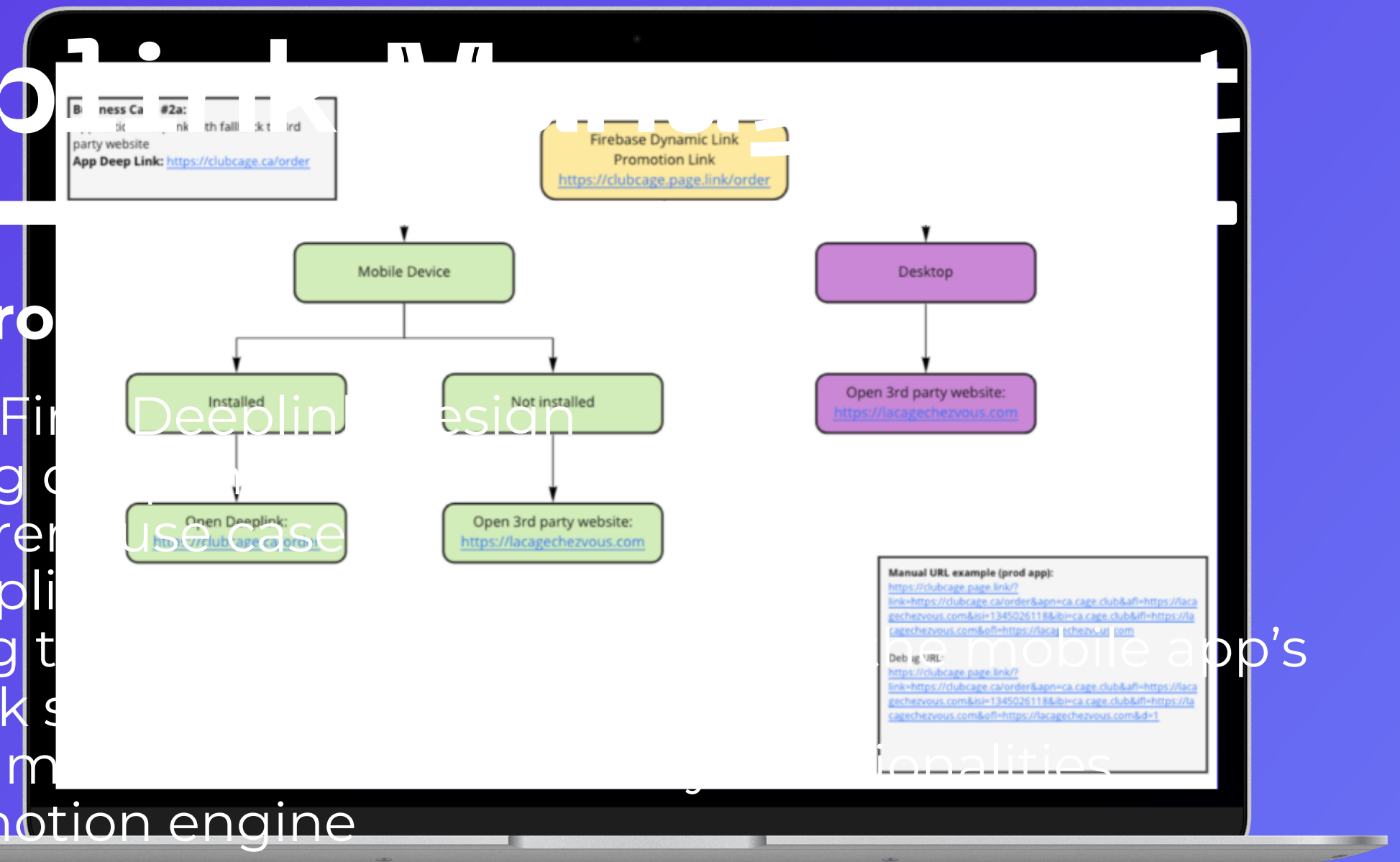
- Deeplinks work well when they seamlessly fit into your cross-channel experience
- Deeplinks fail if any of the below factors are missing:
 - Planning
 - Channel sync
 - Business side education
- Special edge cases
 - Mobile device, but app not installed
 - Mobile focused deeplink opened on desktop

DeepLink Mgmt

Our approach



- Mobile-Focused DeepLink design
- Creating different use cases
 - Different “DeepLink”
- Keeping track of deeplink success
- Dynamic promotion engine



Analytics

Never lose sight of how
customers use your application.
Keep adjusting.

Analytics

Challenges

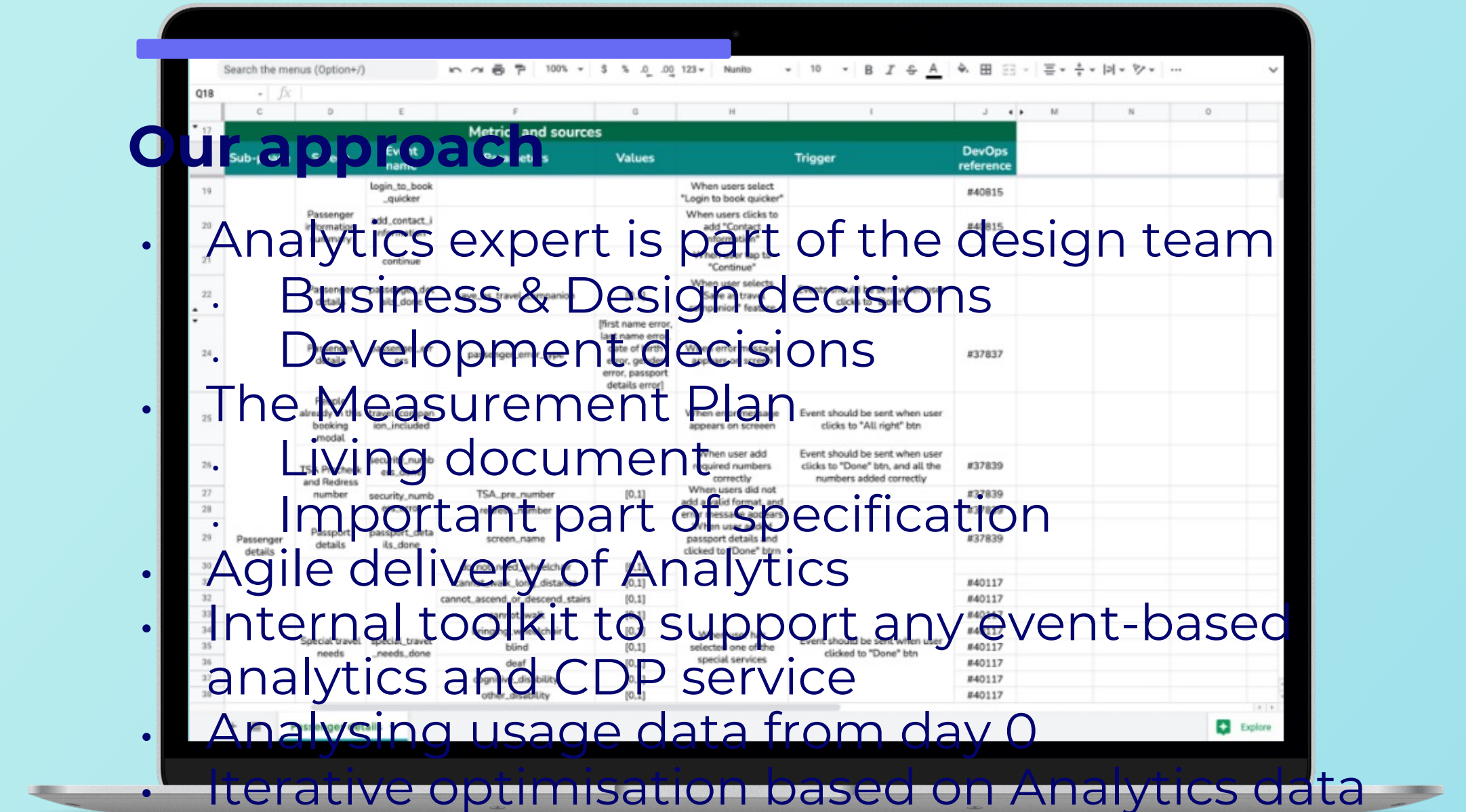
- An Analytics expert is not a developer, a developer is not an Analytics expert. Neither of them is designer.
- Caveats of handling Analytics at the end of the project
 - Last-minute surprises
 - Blocking issues
 - Technical debts
- Being flexible with the analytics provider / number of providers and CDP
- Going beyond Analytics
 - Performance Monitoring
 - Crash Management



Analytics

Our approach

- Analytics expert is part of the design team
- Business & Design decisions
- Development decisions
- The Measurement Plan
- Living document
- Important part of specification
- Agile delivery of Analytics
- Internal toolkit to support any event-based analytics and CDP service
- Analysing usage data from day 0
- Iterative optimisation based on Analytics data
- Beyond Analytics
 - App as front office
 - Performance Monitoring
 - Crash Management



Release Management

Manual releases are the past for everyone's sake.

Release Management

Challenges

- No one-size-fits-all solution for release automatisisation
- Risks of automatisisation without safeguards
- Identification and mitigation of corrupt release



Release Management

Our approach

- Starting from our tried and tested CI/CD workflows, always customizing to the actual project
- Support of major toolkits (Atlassian / Azure DevOps / GitLab CI)
- Safeguard system for automated releases
 - 100% unit test coverage
 - Static code analysis
 - Automated integration tests
 - All test processes are part of the automated process
- Scheduled releases
- Zero-touch approach
 - Staged rollout
 - Stability monitoring
 - Automatic release rollback based on release stability



Q&A

mndwrk



mito.digital

Thank you!

Gergely Békési

g.bekesi@mito.hu

Mobile Program

Manager

Mito Digital