TechLead Conf · June 30, 2021

Rebuilding mobile bank development around the business

From a CTO who owns not just the design but its implementation

TechLead Conf 2021



From «how it was» to «when it is your turn»

What we'll cover

- How it was before 2020
- What changed and how we accounted for it
- Rolling it out: teams, architecture, quality
- Triggers, outcomes and further reading

Part 1

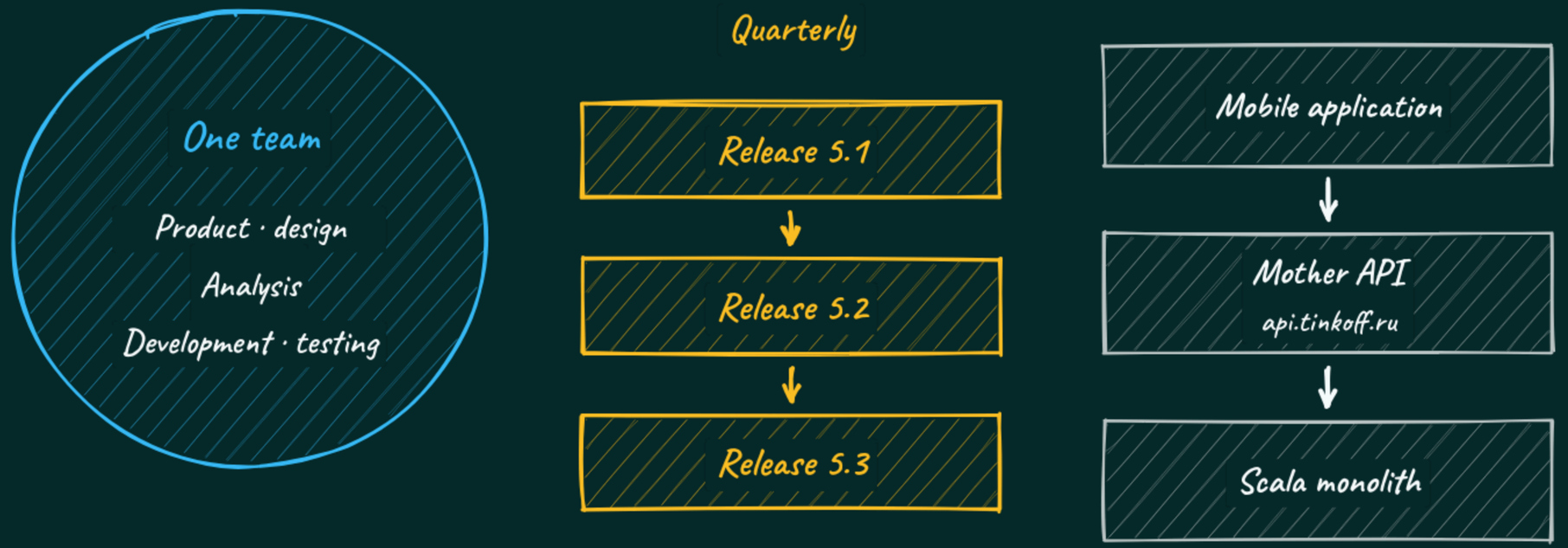
01

How it was before 2020

End of 2019

One team, quarterly releases, a monolith

End of 2019 How it was



This worked — and here is why

The design matched the business it served

- There were a few main products
- The team was small — around 50 people
- The focus was on adding functionality
- Large irregular releases suited the business

Part 2

02

What changed and how we took it in

The business grew, the IT structure changed

Business What changed

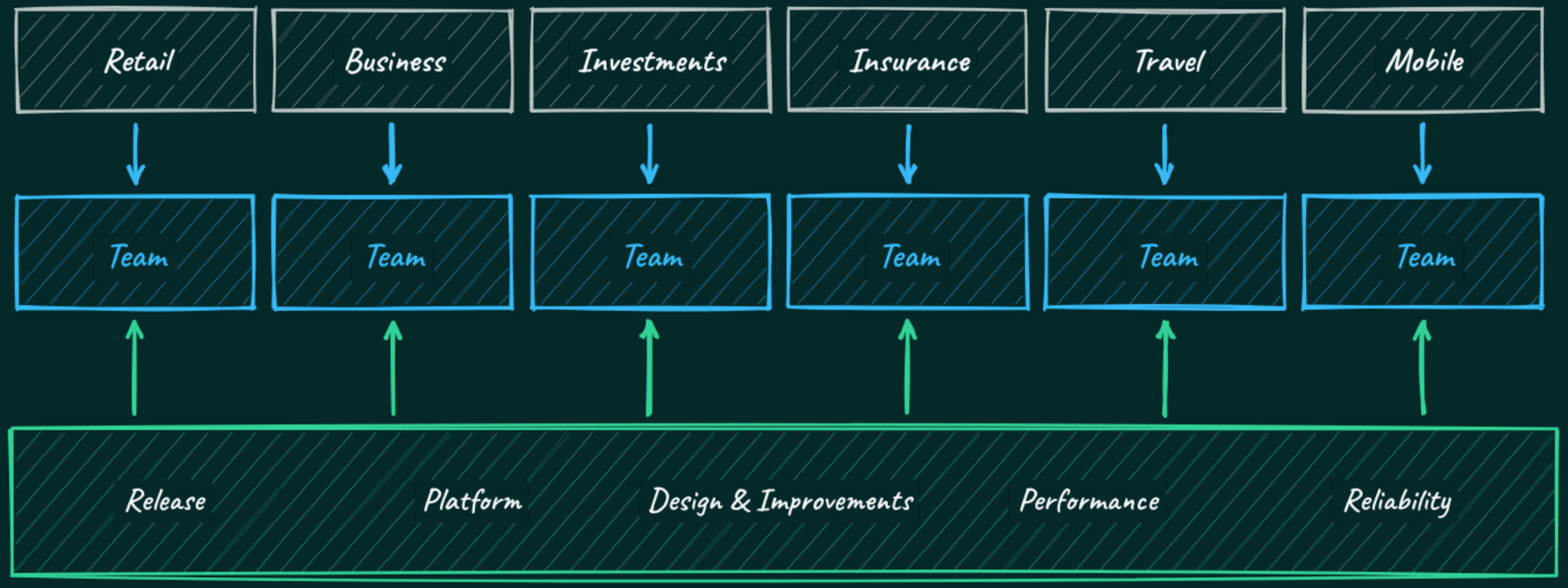
A monoline bank grew into a group of companies

The Tinkoff group and the IT structure

- 2007–2020: from Tinkoff Black to the SuperApp
- Six business verticals now
- Platforms grew above the infrastructure
- Each vertical wants its own pace

Each customer gets a team; everyone shares platforms

Solution Platform



Releases are tied to dates, not to scope

The release train

- Teams must be able to work independently
- The release leaves on its date
- Miss it and you take the next one
- The contents stopped being a promise

Train Release train

Change the teams and you must change the system

Conway Conway's law

Conway’s law, and brownfield rather than greenfield

- Organizations design systems that copy their communication
- The desired team structure changed
- So the system has to be redesigned
- The business cannot pause — so we redesign evolutionarily

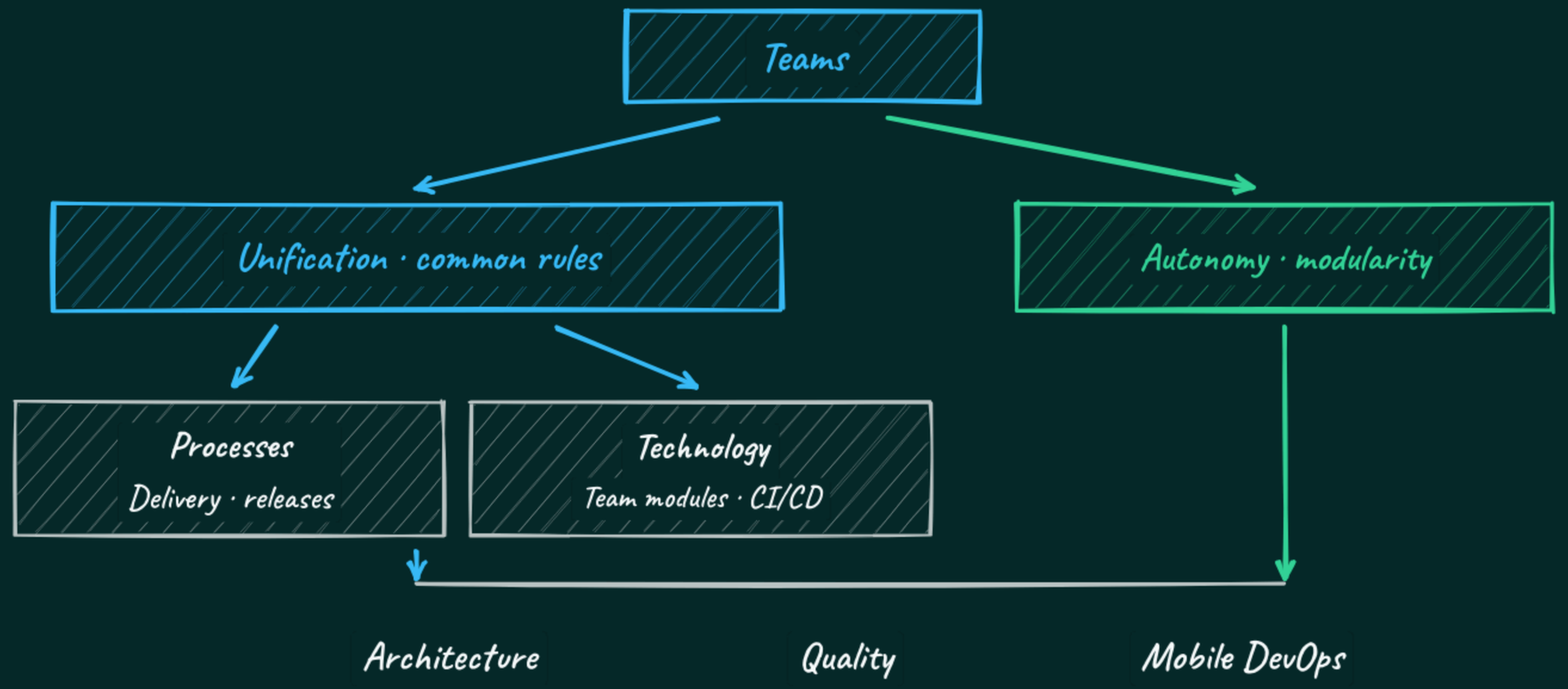
Part 3

03

How we rolled the changes out

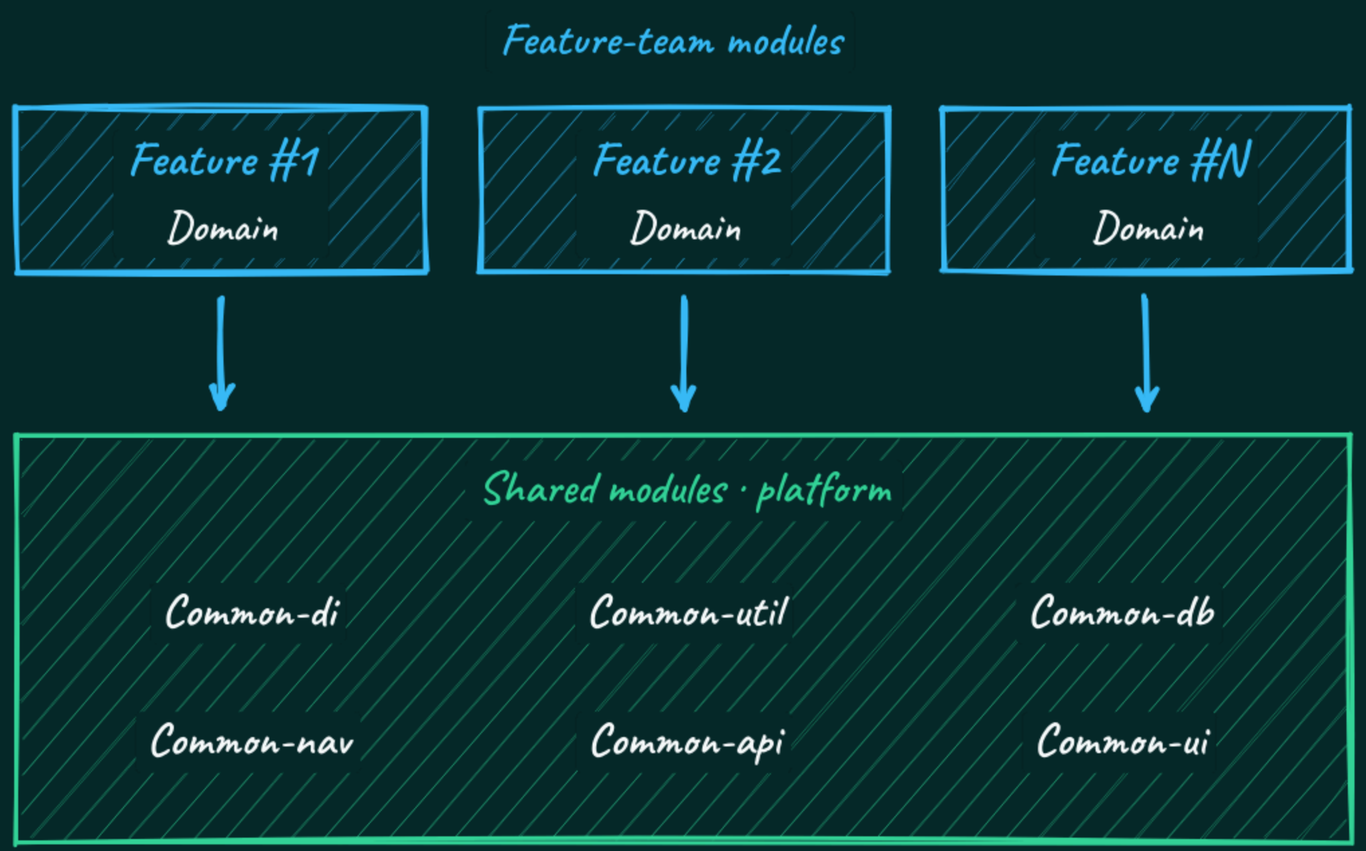
Teams, architecture, quality, Mobile DevOps

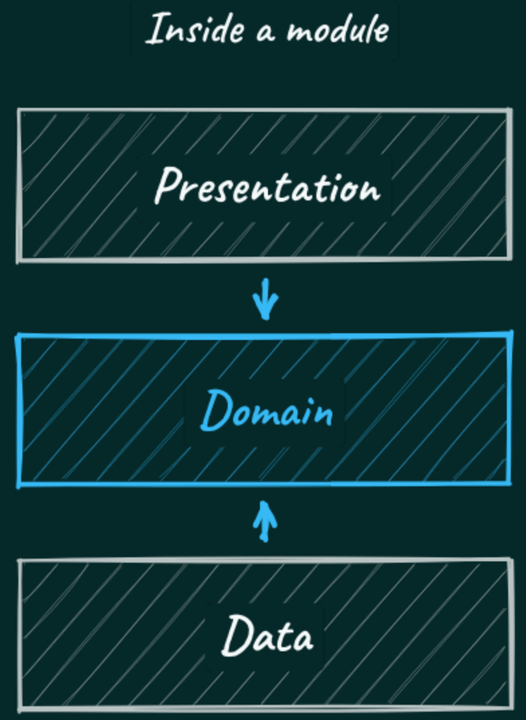
Common rules support team autonomy



Feature teams own their modules

Architecture | Modules





From checklists to test cases and a testing pyramid

Quality Testing

Quality assurance

- Test model: checklists → test cases
- We wrote down the test debt on existing functionality
- The pyramid: unit, integration, UI, end-to-end
- Plus quality metrics, crowdtesting and contract testing

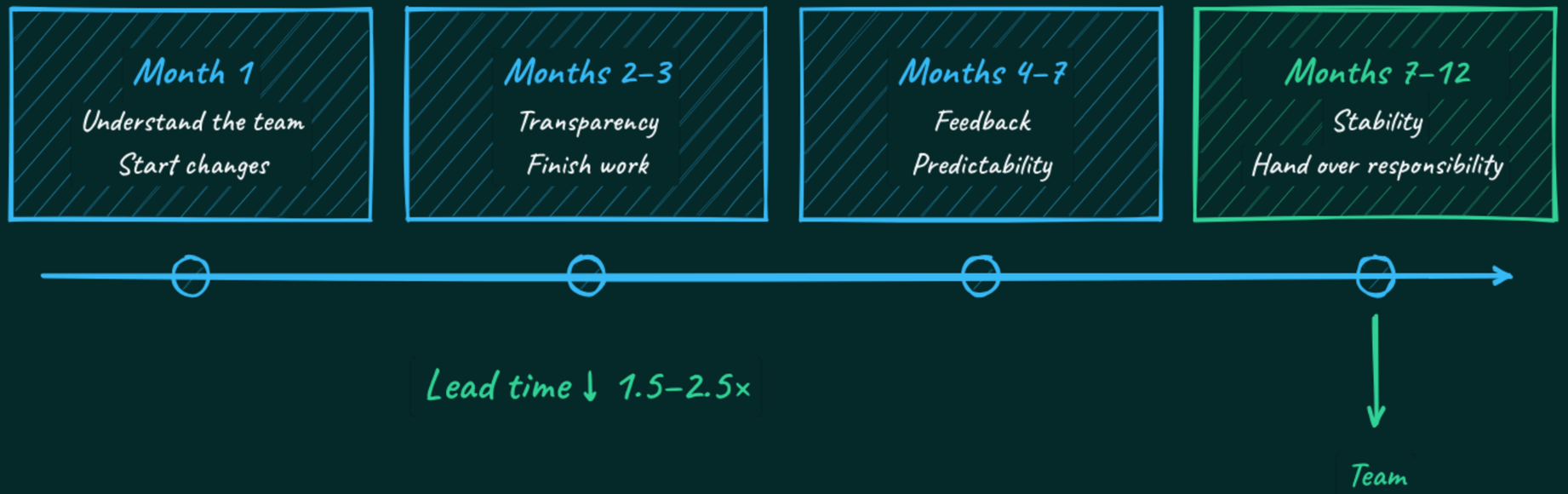
The added load demanded its own improvements

Mobile DevOps

- More builds and autotests — we reworked the CI/CD infra
- Code, style and architecture control became a requirement
- We automated the checks: Danger for iOS, Sonar for Android
- Client monitoring and feature toggles were added

Responsibility for change passes to the team

Process | Delivery manager



Part 4

04

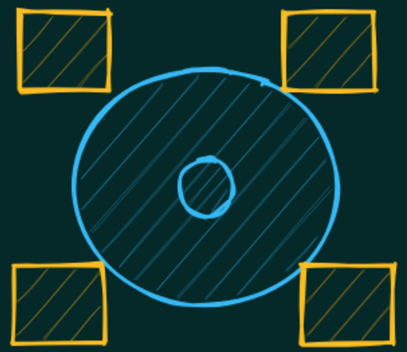
Triggers and outcomes

When it is time to change your own teams

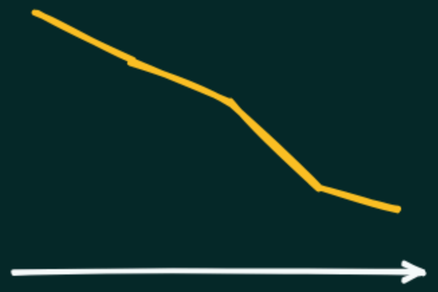
Three signals that teams need to evolve

Triggers When it's time

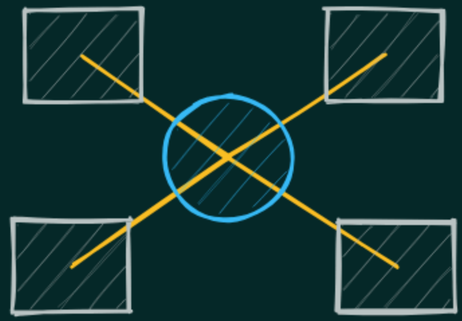
Size
Software outgrows the team



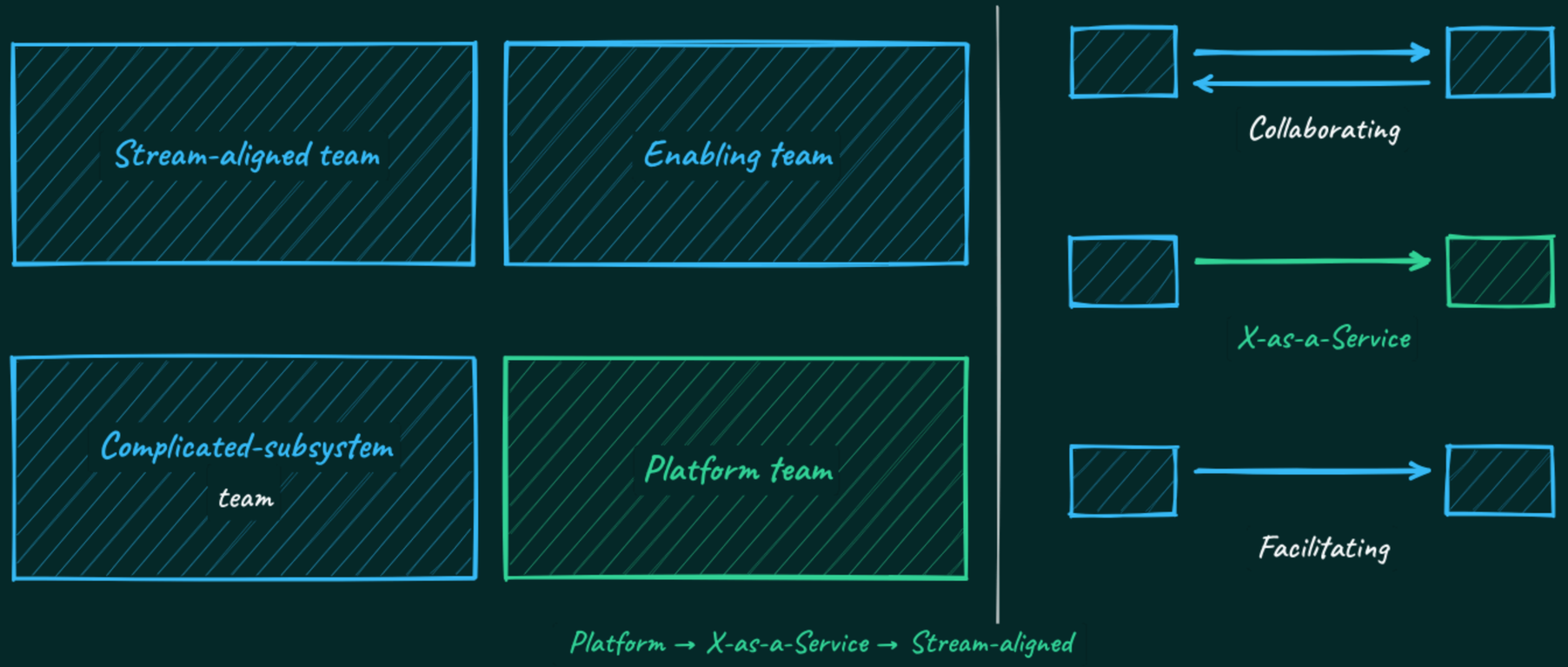
Delivery
Sustained slowdown



Dependencies
Fragmented services



Teams use three interaction modes



Everything changed at once, and it was all connected

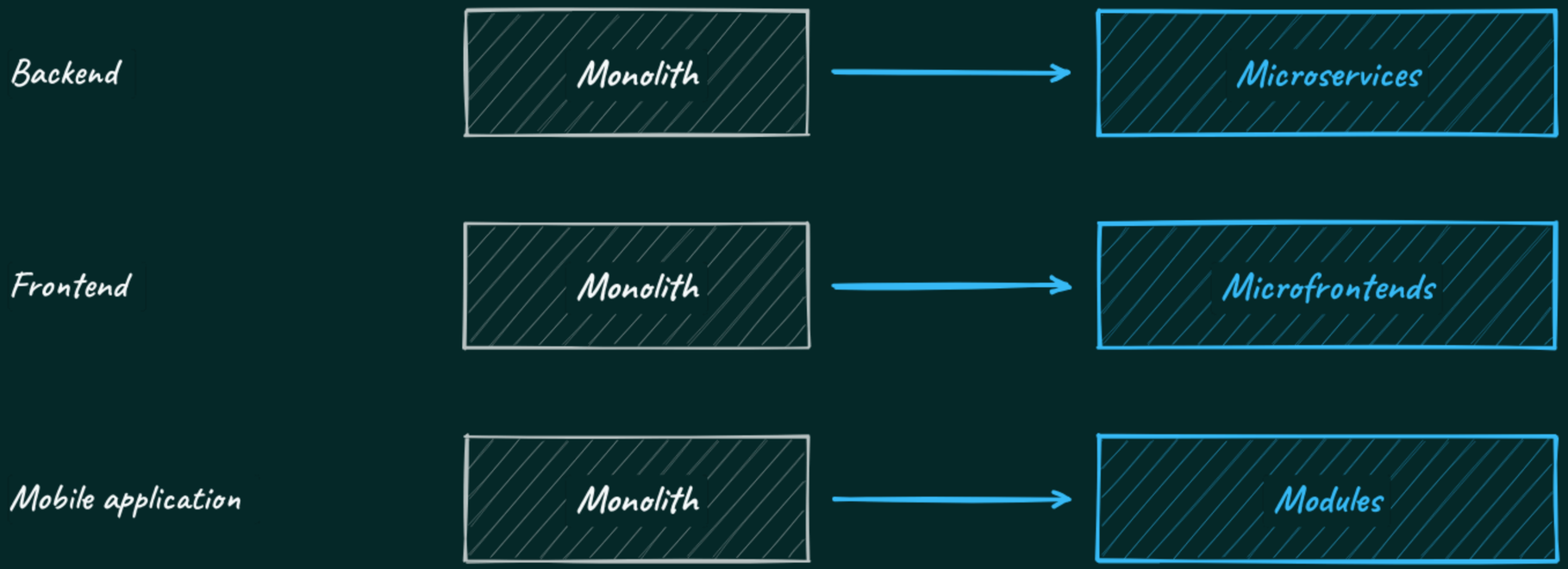
What came out of it

- The business requirements changed
- The team evolution triggers fired
- We had to redesign the team structure
- And with it the process, architecture, quality and CI/CD
- All of these changes are connected and run in parallel

- That is exactly what makes it hard and interesting

Decomposition connects backend, frontend and mobile

P.S. The same evolution



Further reading Sources

What to read next

The links from the talk’s closing slide

- The «Team Topologies» book and its three-part summary
- A review of «SRE practices in mobile development»
- Articles on platform teams and client-friendly APIs
- Earlier talks: Teamlead Conf 2018 and ArchDays 2019

polomodov.tech

Thank you!

All slides and links are in the Telegram channel

Alexander Polomodov

Technical Director & Fellow,
T-Technologies



@book_cube