Breaking out of the cage: Move complex development to GitHub

Alexander Smirnov
Technology Expert, Nebius

A team of 500+ highly skilled professionals

Recognized as an NVIDIA® Preferred Partner

Mantsala

Amsterdam

Belgrade

Tel Aviv
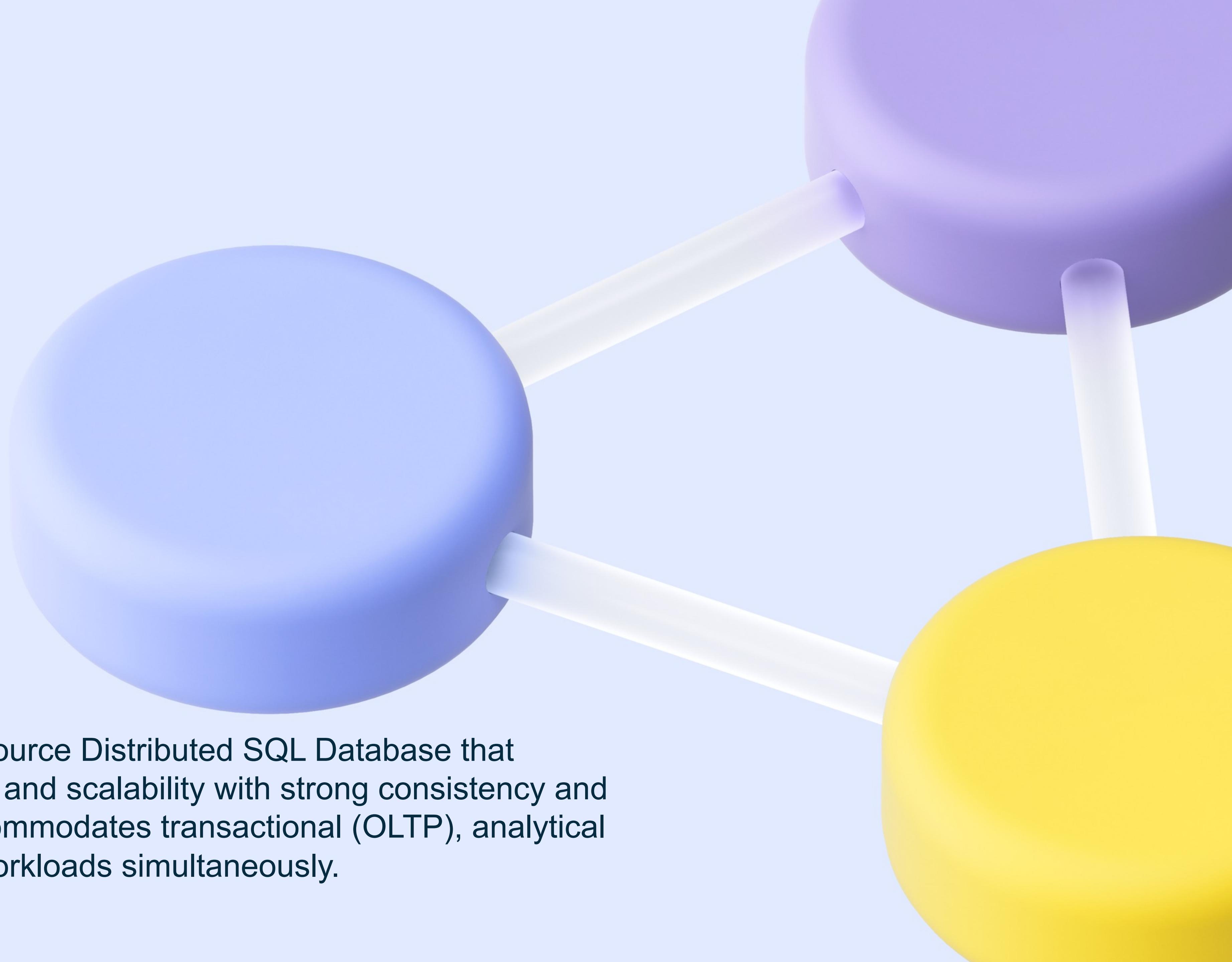
Green data center in Finland featuring modern solutions

16

Nebius AI Supercomputer ISEG Ranks 16th in the Top 500, exemplifying our cutting-edge HPC capabilities
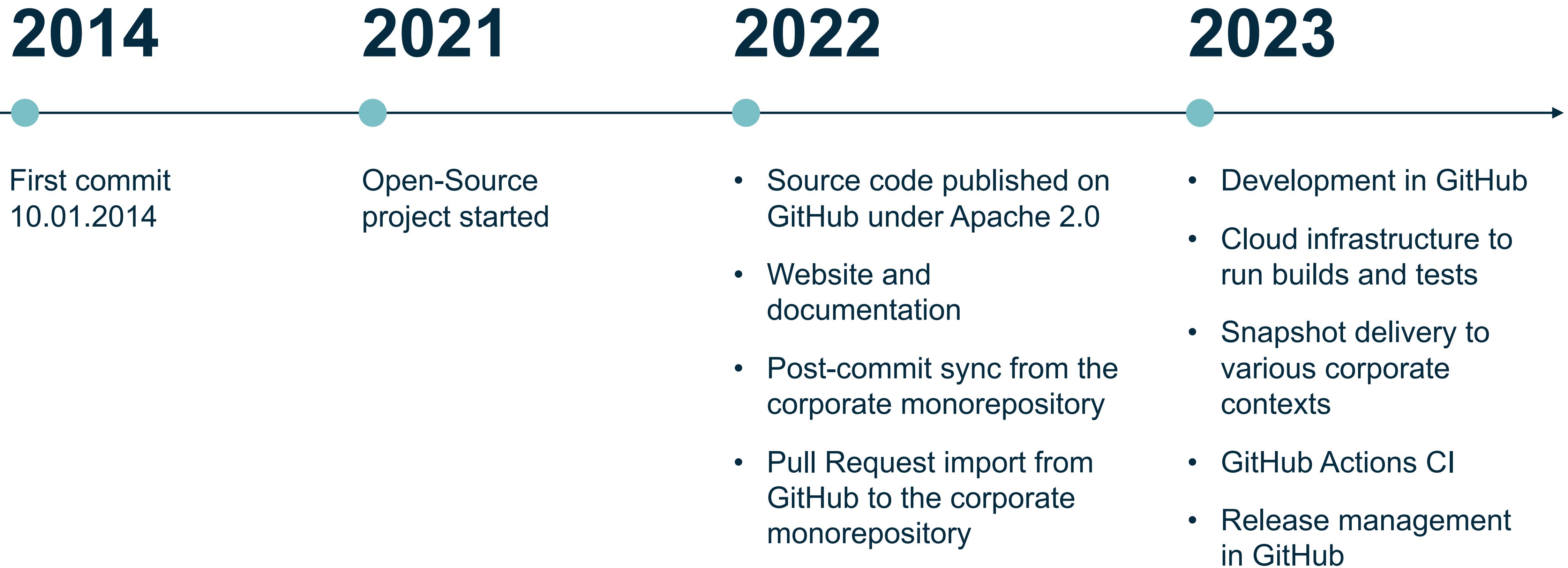
Nebius is a European public cloud with global expertise

YDB

YDB is a versatile open-source Distributed SQL Database that combines high availability and scalability with strong consistency and ACID transactions. It accommodates transactional (OLTP), analytical (OLAP), and streaming workloads simultaneously.
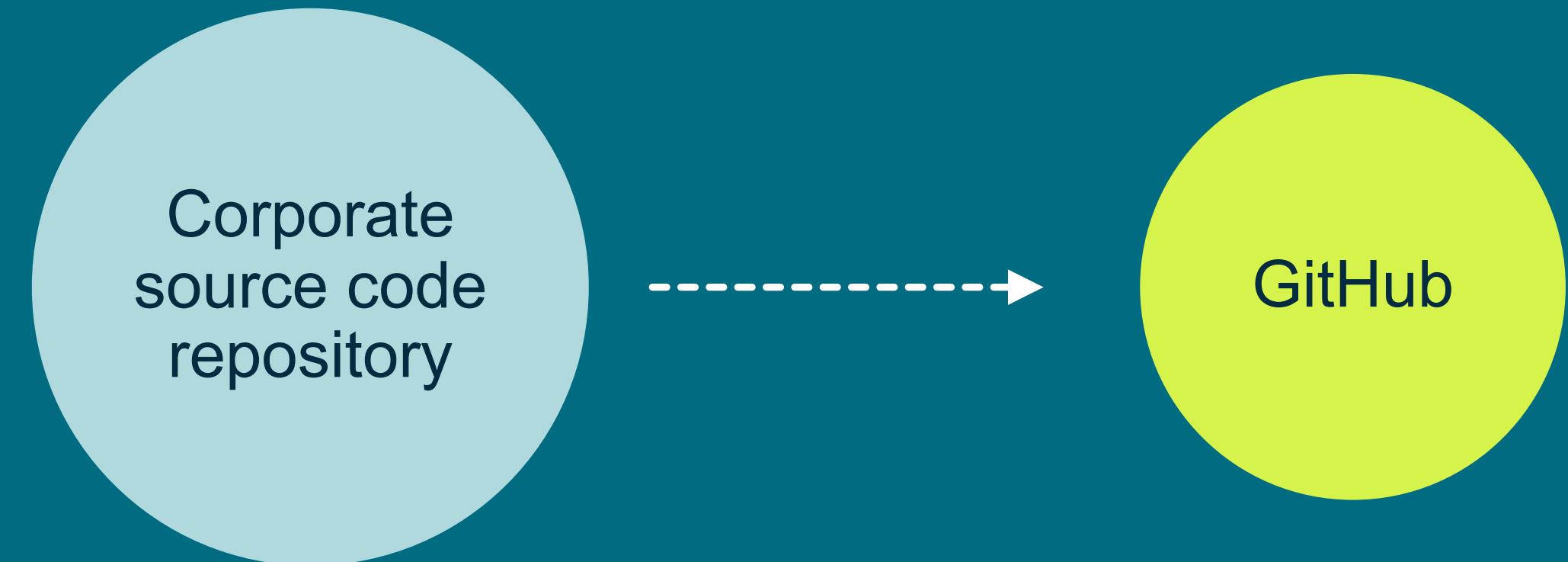
YDB Open-Source Timeline

2014

First commit
10.01.2014

2021

Open-Source
project started

2022

- Source code published on GitHub under Apache 2.0

- Website and documentation

- Post-commit sync from the corporate monorepository

- Pull Request import from GitHub to the corporate monorepository

2023

- Development in GitHub

- Cloud infrastructure to run builds and tests

- Snapshot delivery to various corporate contexts

- GitHub Actions CI

- Release management in GitHub

The Mirror

- Increase popularity

- Reduce vendor lock for corporate customers

Corporate source code repository

GitHub
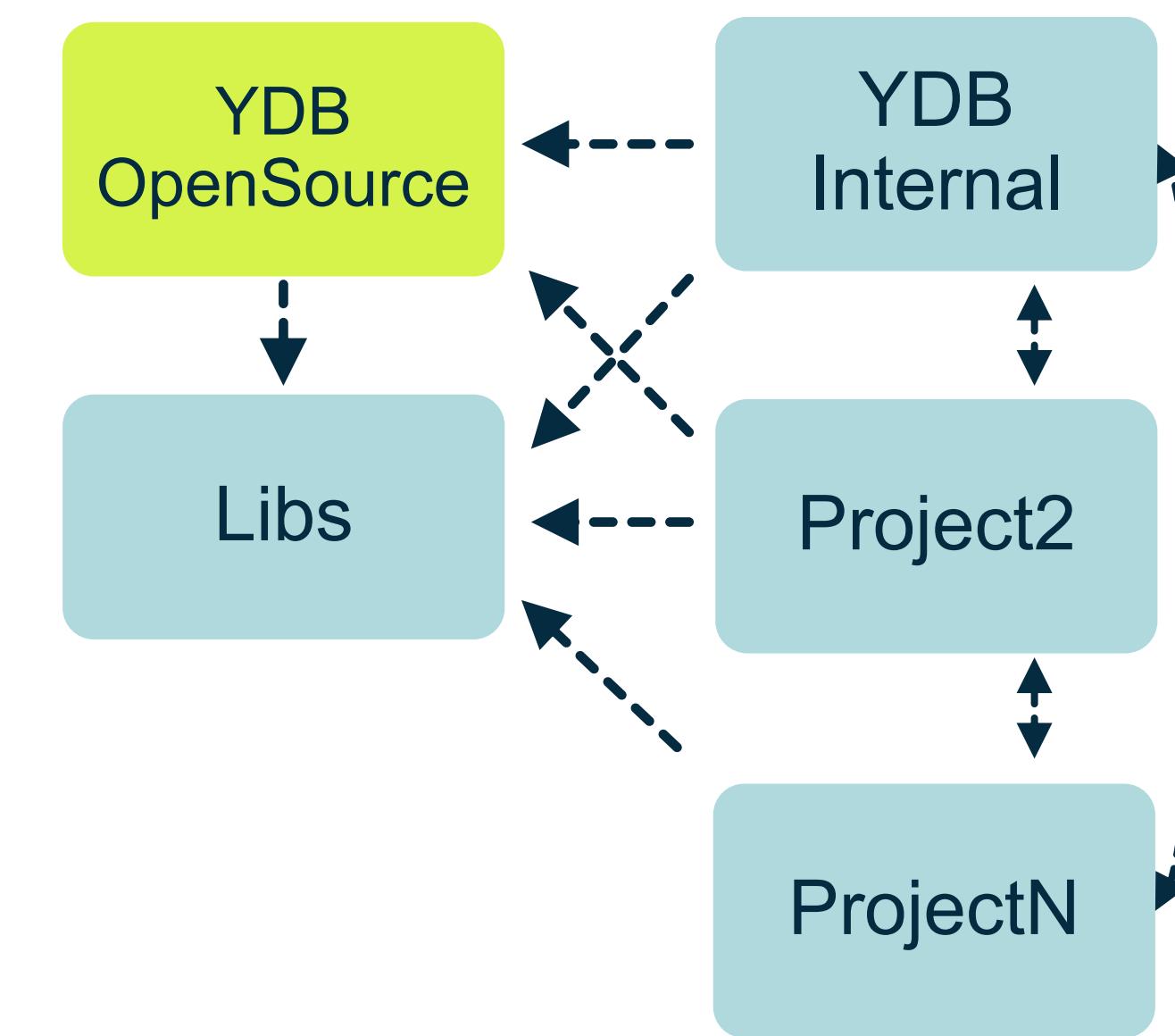
Open-source code isolation

Internal repository structure, arrows indicate dependencies:

BEFORE
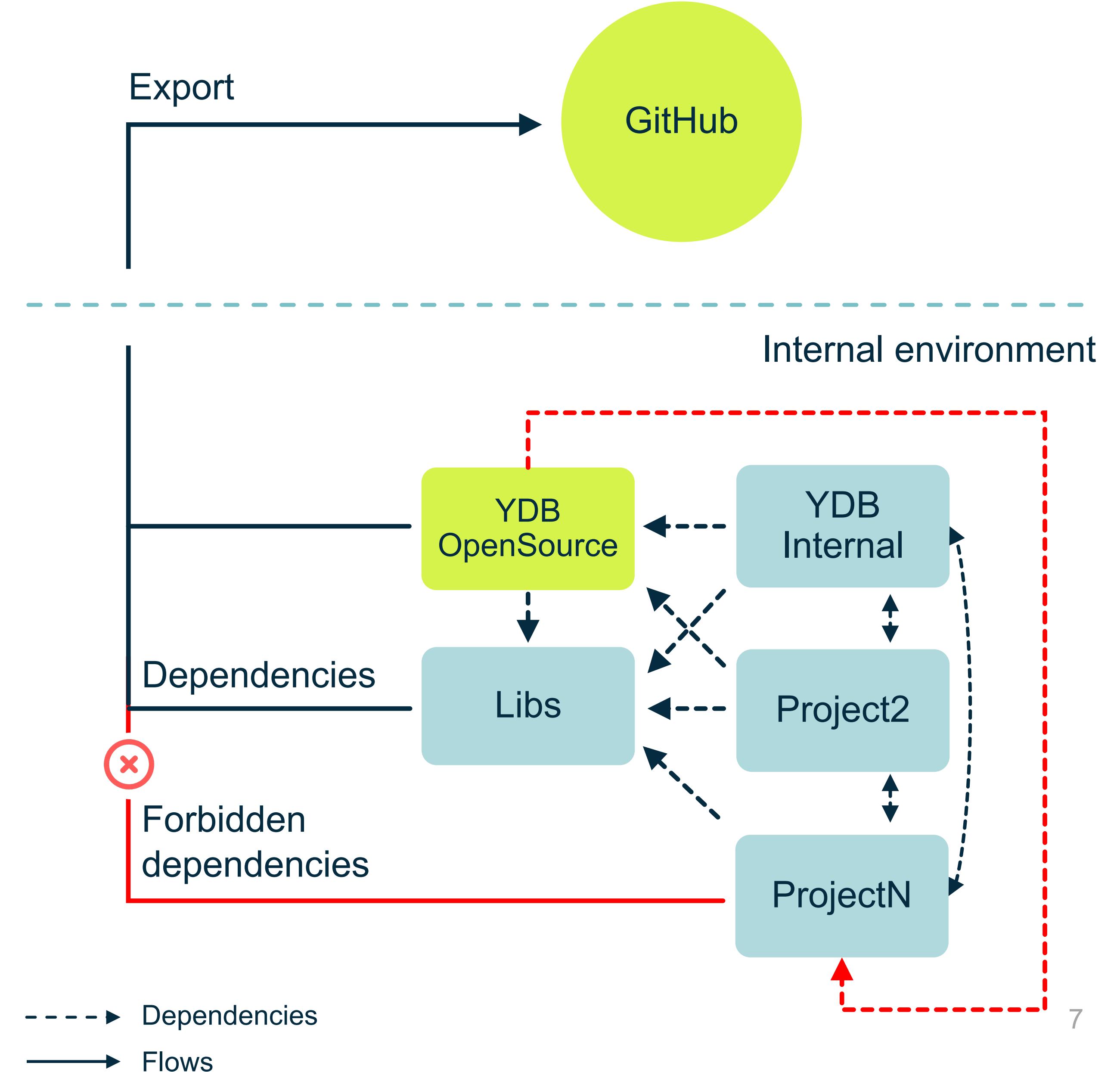
AFTER

Phase one

Export sync
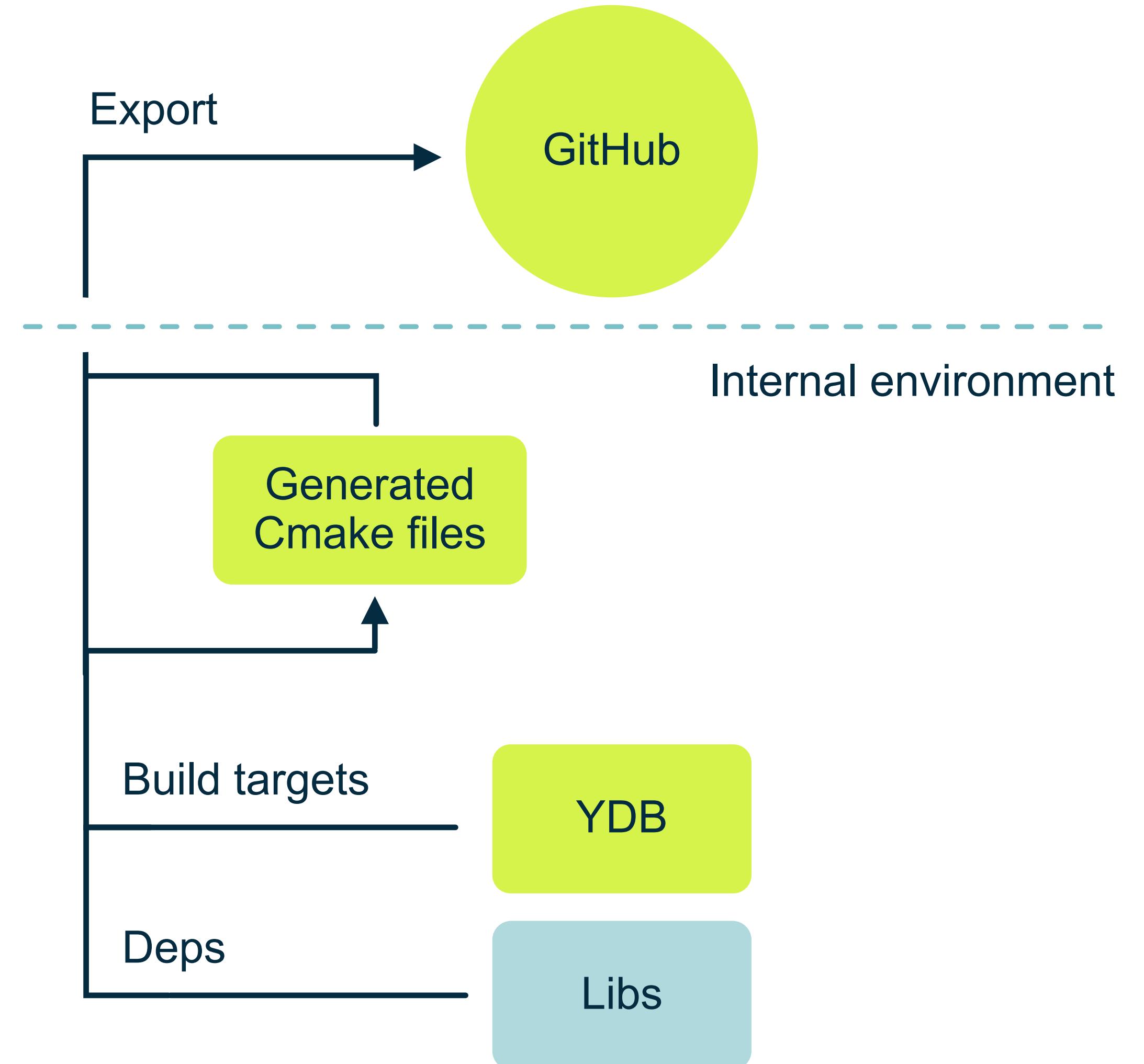
- Capture repository changes
- Build graph dependency discovery
- Filter for allowed directories

Export

GitHub

Internal environment

YDB OpenSource

YDB Internal

Dependencies

Libs

Project2

Forbidden dependencies

ProjectN

- - - → Dependencies
——→ Flows

Phase one

CMake build

- Internal build system is not available outside

- Generate CMakeLists.txt

- Unit tests supported

Export

GitHub

Internal environment

Generated Cmake files

Build targets

YDB

Deps

Libs
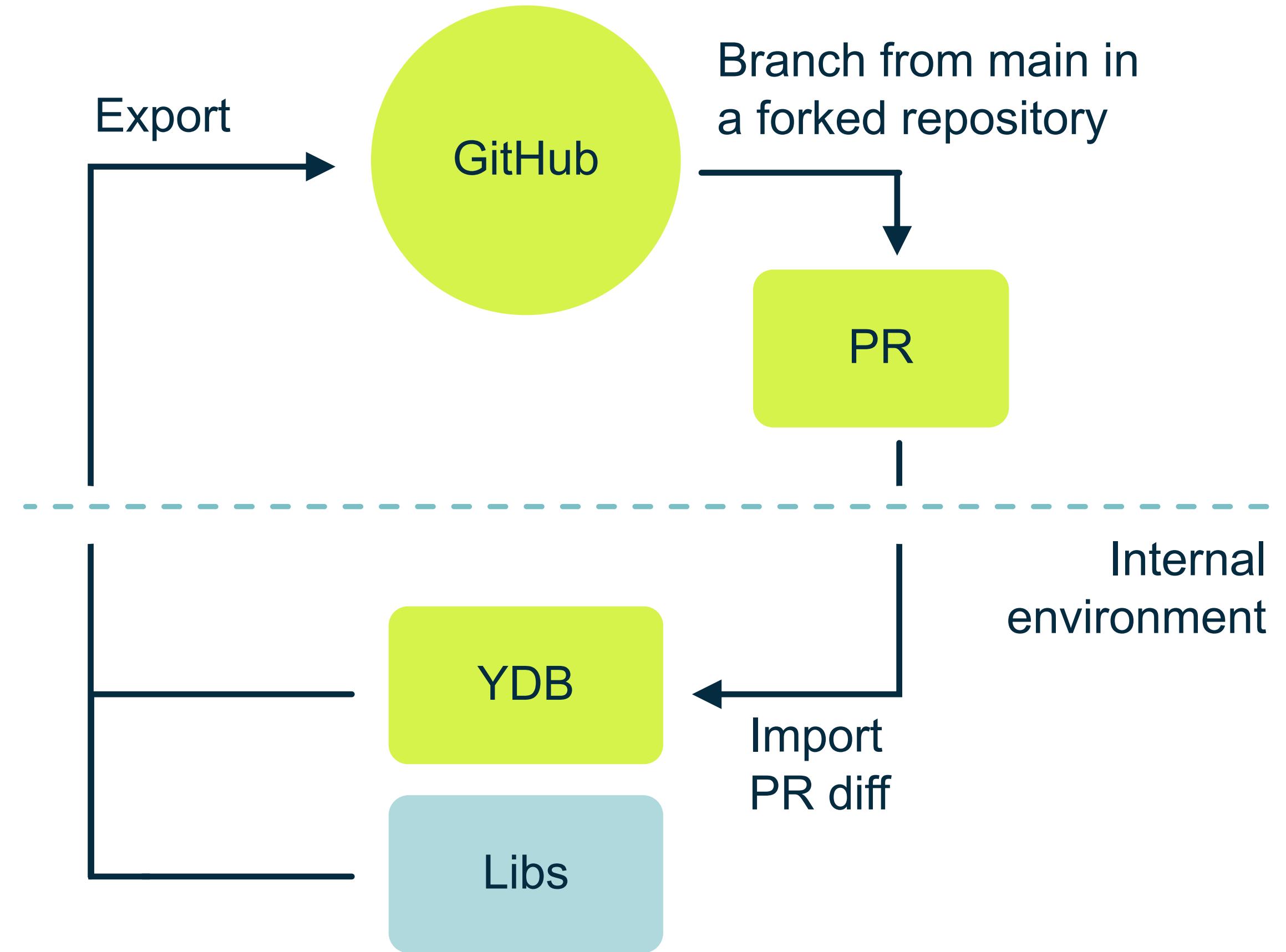
Pull Requests import

- Merge to main in GitHub is blocked

- Import PRs, merge internally

- Commit authorship is preserved

Export

Branch from main in
a forked repository

GitHub

PR

Internal
environment

YDB

Import
PR diff

Libs

Achievements

- Source code is publicly available
- Anyone can build and run YDB
- Some tests can be run too
- Increased interest
- Corporate clients attained
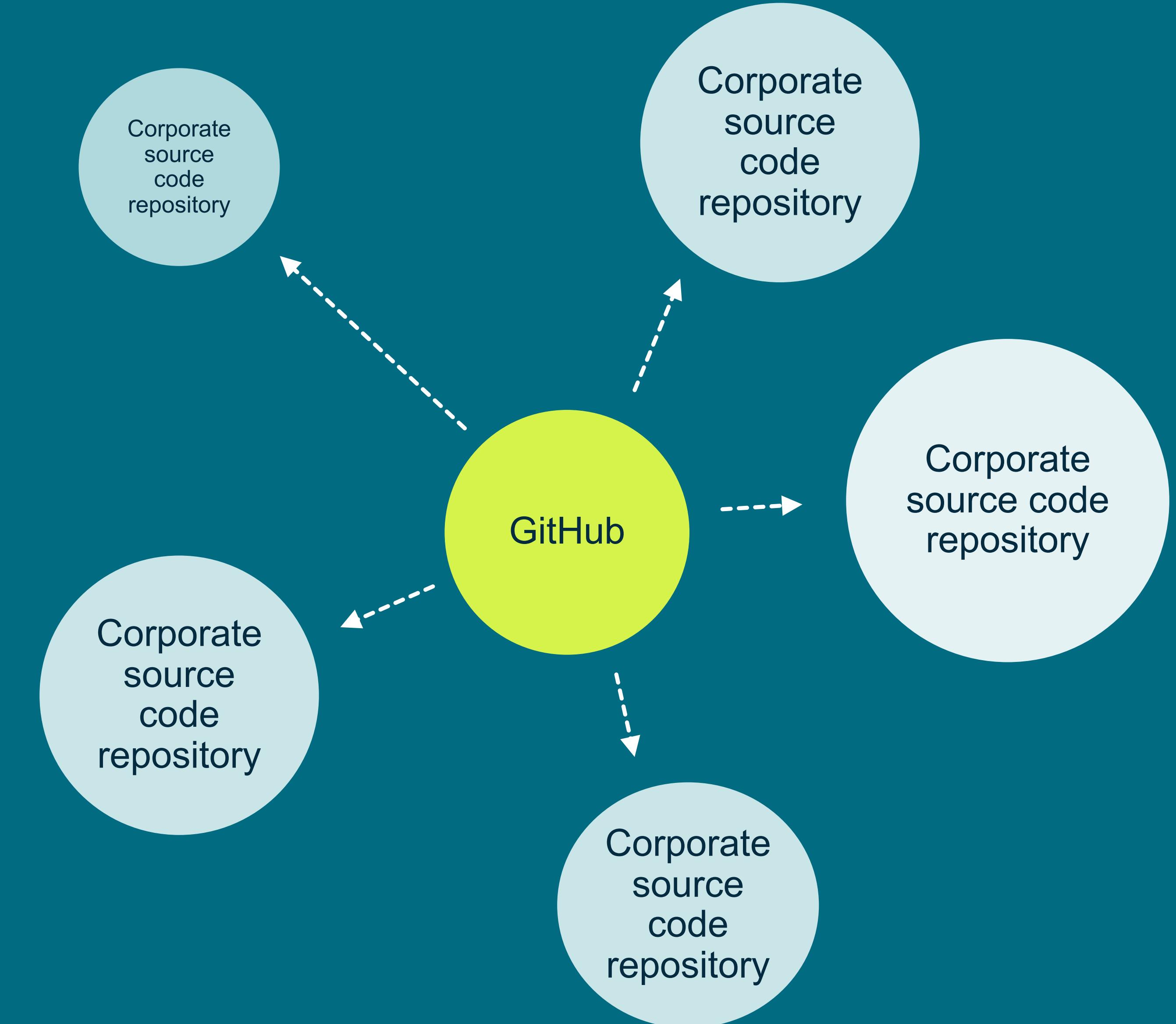
Impediments

- Contributors cannot run tests
- Contributors have no tools and infrastructure to work on a change
- Corporate contributions are almost impossible

The Star

- Enable contributions
- No private context for development
- One product for all users
- One process for all contributors

Corporate source code repository

Corporate source code repository

GitHub

Corporate source code repository

Corporate source code repository

Corporate source code repository

Public cloud

- On-demand virtual machines

- Permanently running VM for on-demand dispatcher

- Permanently running VM for build cache

- S3 object storage

Build cache

- Everyone can read

- Requires authentication to write

Continuous Integration

GitHub Actions:

Nightly checks

Pre-commit checks

Post-commit checks

Nightly checks, test results

- Run on schedule

- Every supported architecture, operating system, compiler

- Structured test information, JUnitXML

- Logs truncation

Nightly checks
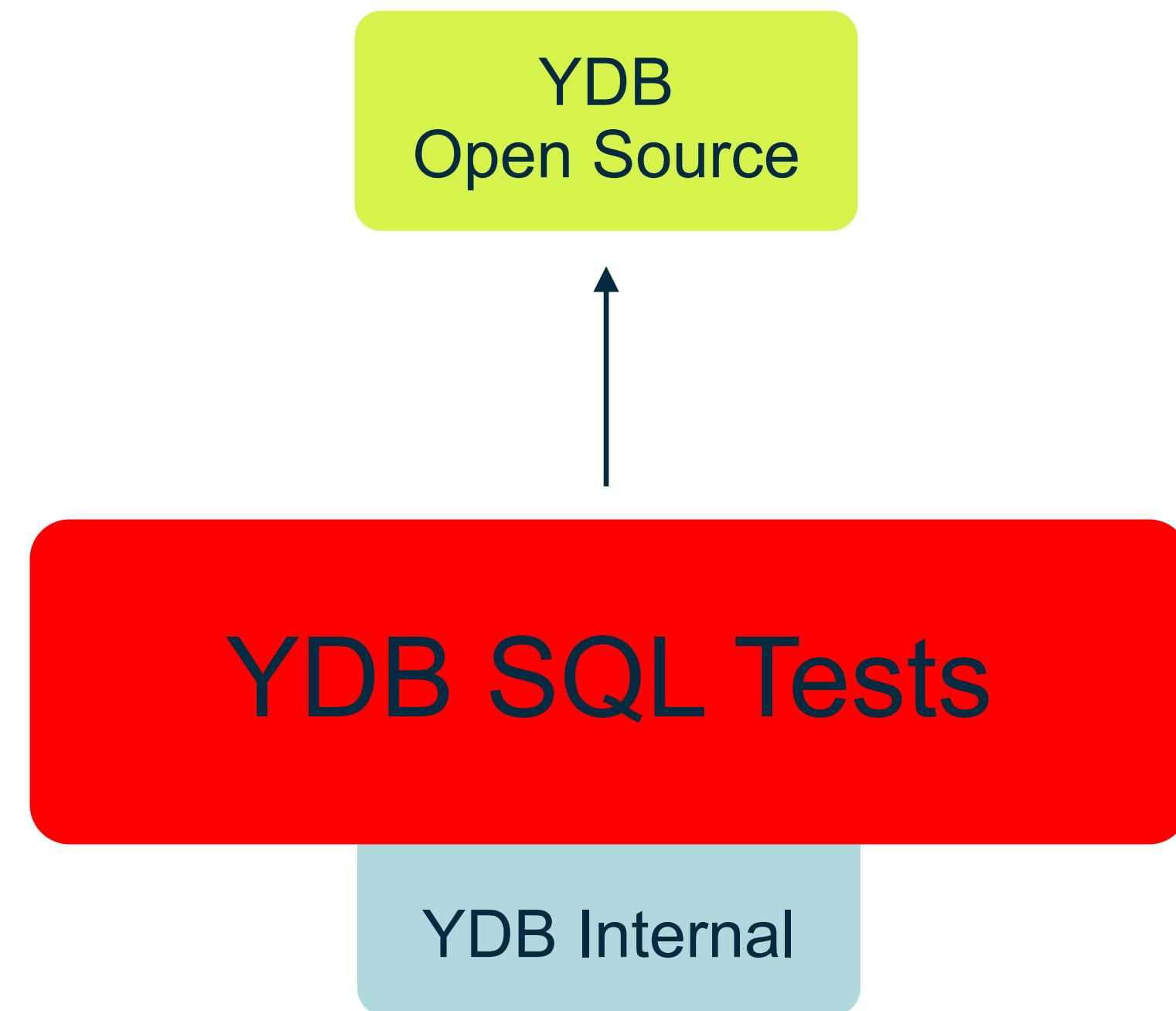
Pre-commit checks

Post-commit checks

Add tests to the YDB code

- You need more code to develop an application rather than to build it

- One of the teams brought a very complex test infrastructure

- Their tests could not be run with CTest

YDB
Open Source

YDB SQL Tests

YDB Internal

Back to internal build system

- Yatool goes Open-Source

- Supports existing development process

- Zero conversion for code import from GitHub

- Developers do not ~~want~~ need to relearn
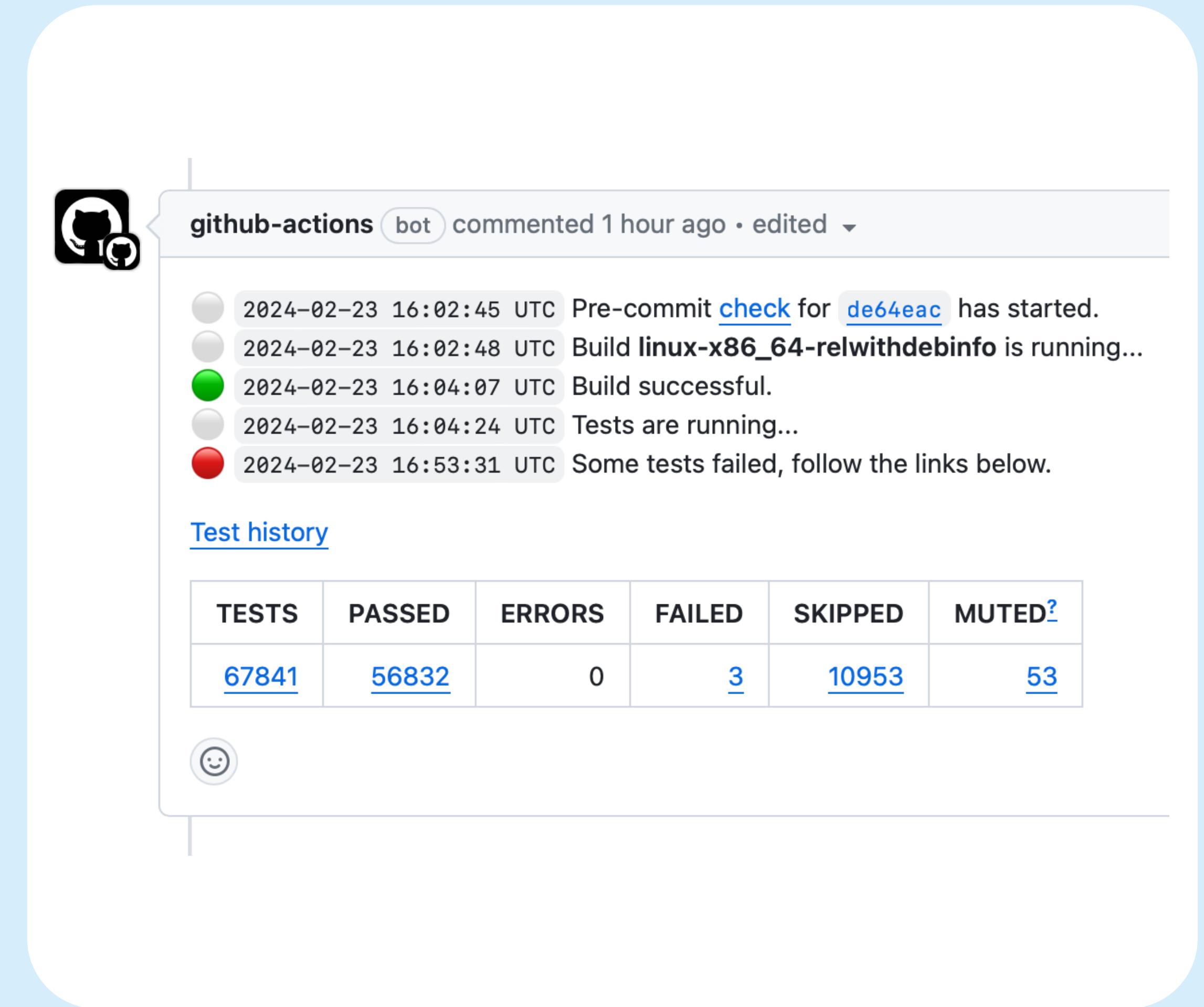
Pre-commit checks

- Green light to merge changes

- On a merge commit

- Dependency graph

- Summary in the Pull Request comments

Nightly checks

Pre-commit checks
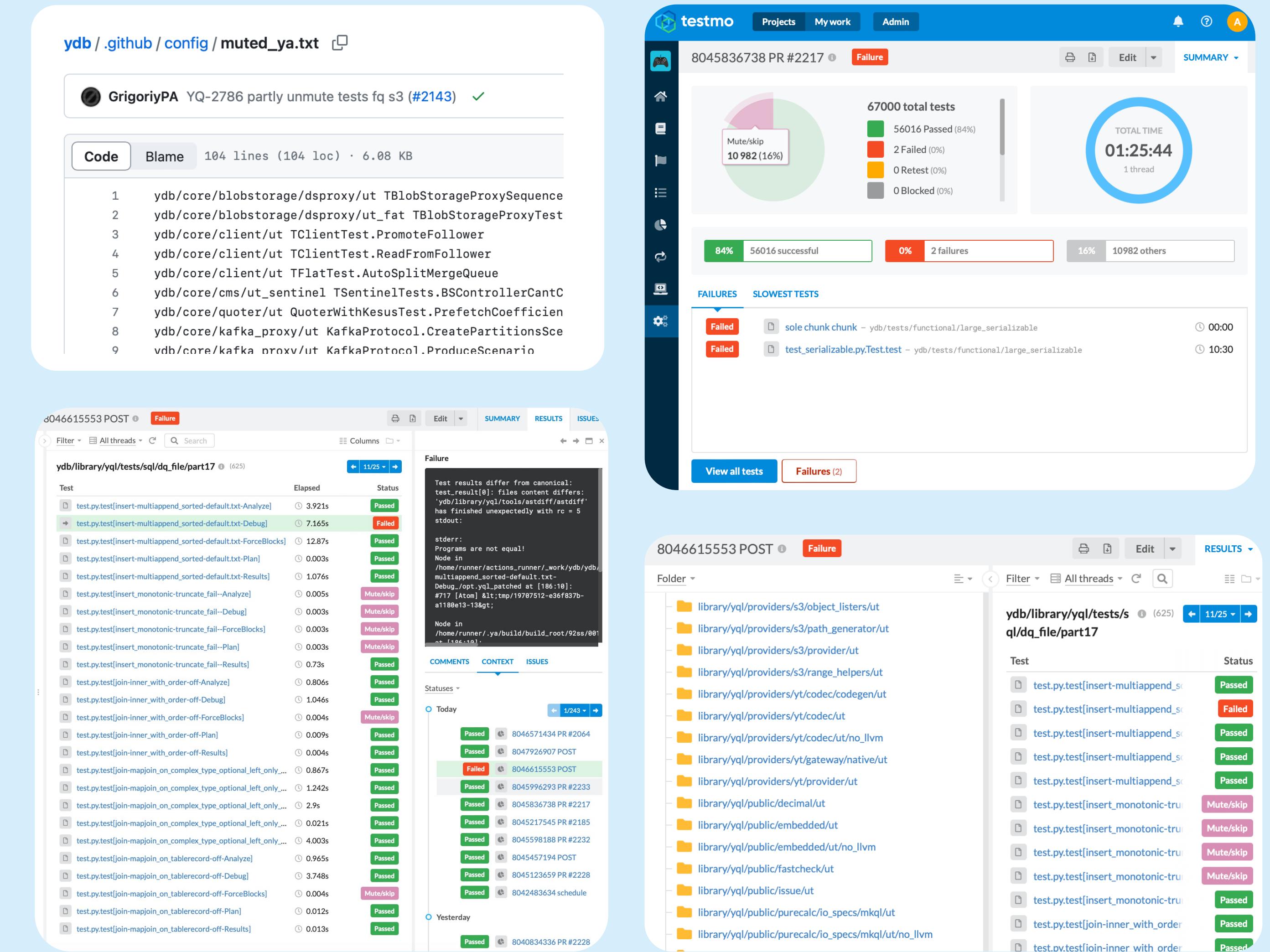
Post-commit checks

Test reporting

- Muted tests list

- Report to Testmo app

- Simple HTML report

Post-commit checks
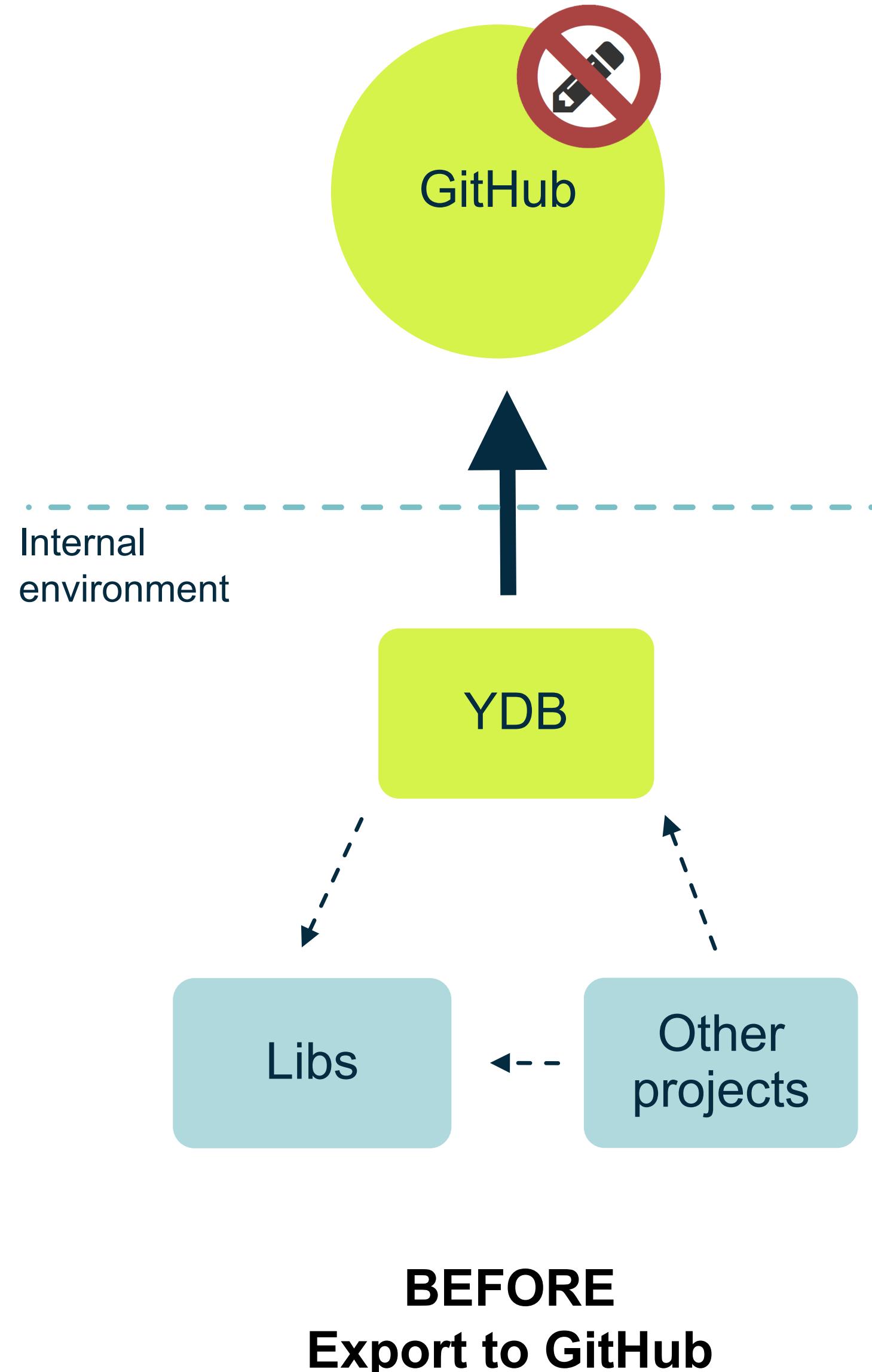
- To identify revision when a problem had appeared.

- Helped to estimate the compute resources consumption.

Nightly checks
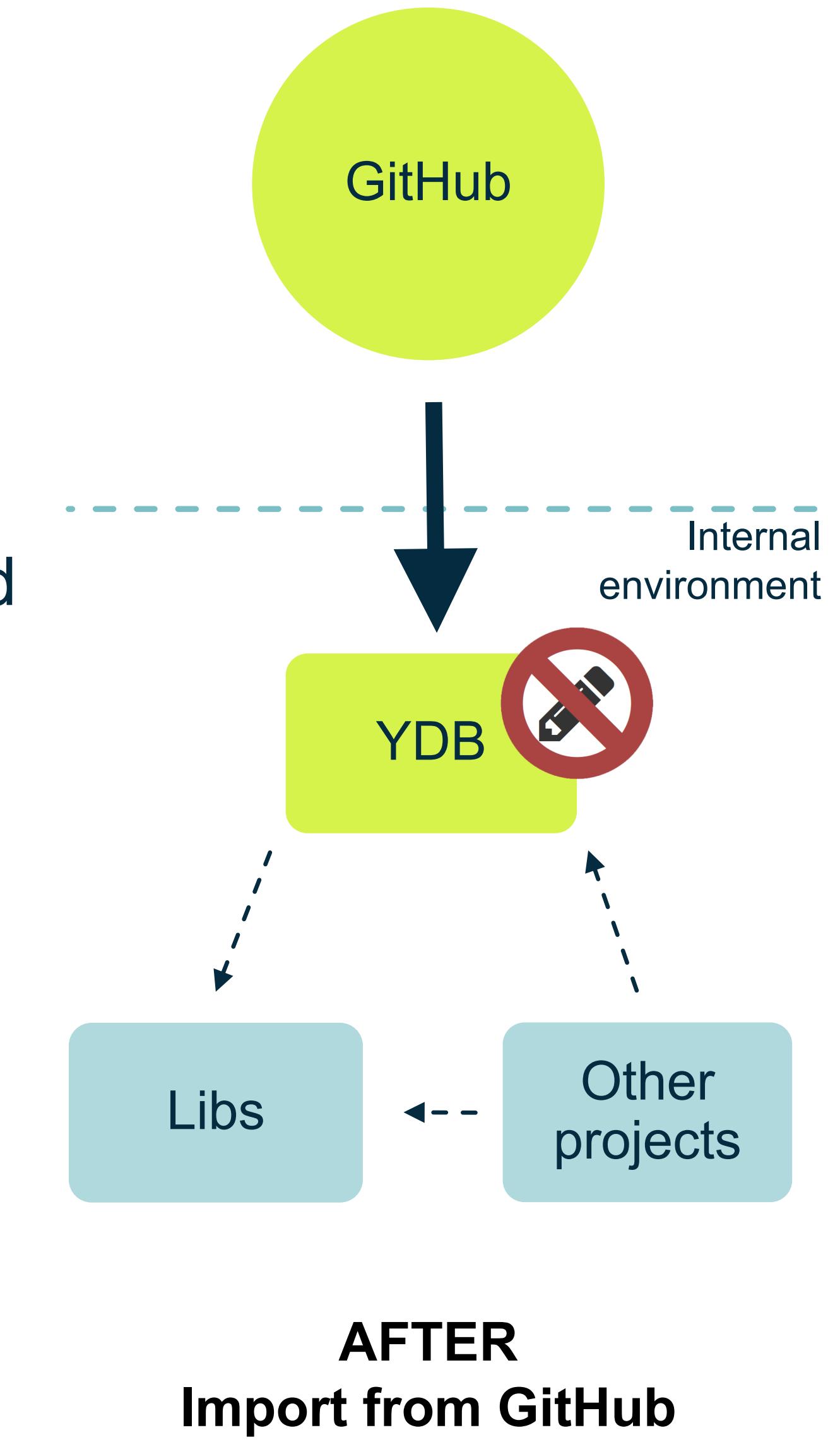
Pre-commit checks

Post-commit checks

Switch now?

- Import flow may fail as it is not tested

- Developers may fail to develop outside the monorepository

- We need to mitigate the risks above before actual switching

GitHub

Internal environment

YDB

Libs

Other projects

BEFORE
Export to GitHub

GitHub

Internal environment

YDB

Libs

Other projects

AFTER
Import from GitHub

- - → Dependencies

——→ Flows

20
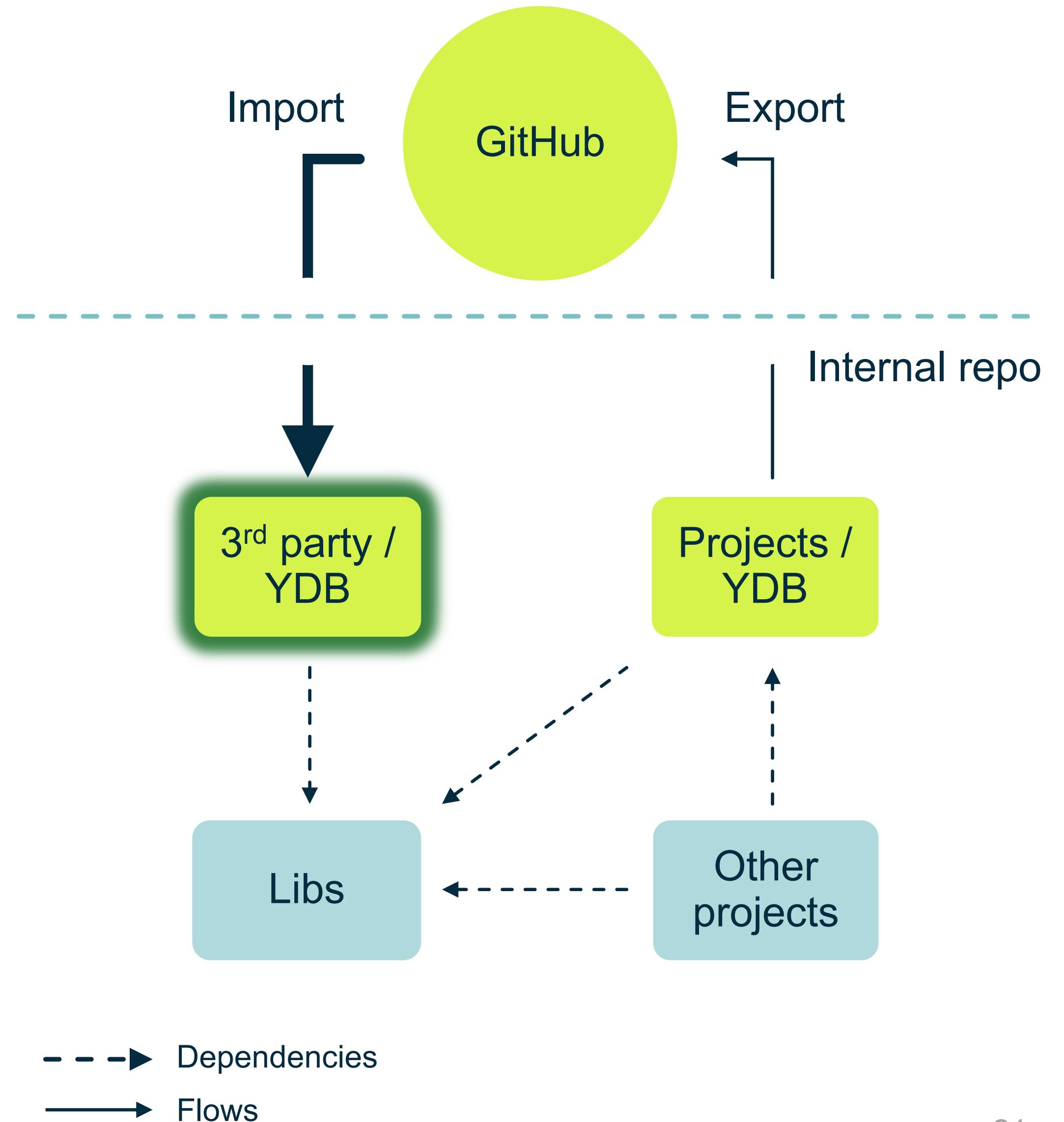
Prepare the switch: Step 1

Enable import to the 3rd-party/ydb

- Revision-based, 4 times a day

- Pull Request, required tests run

- Patching code to replace references

- Only YDB code, without libraries

- Stabilize import automation
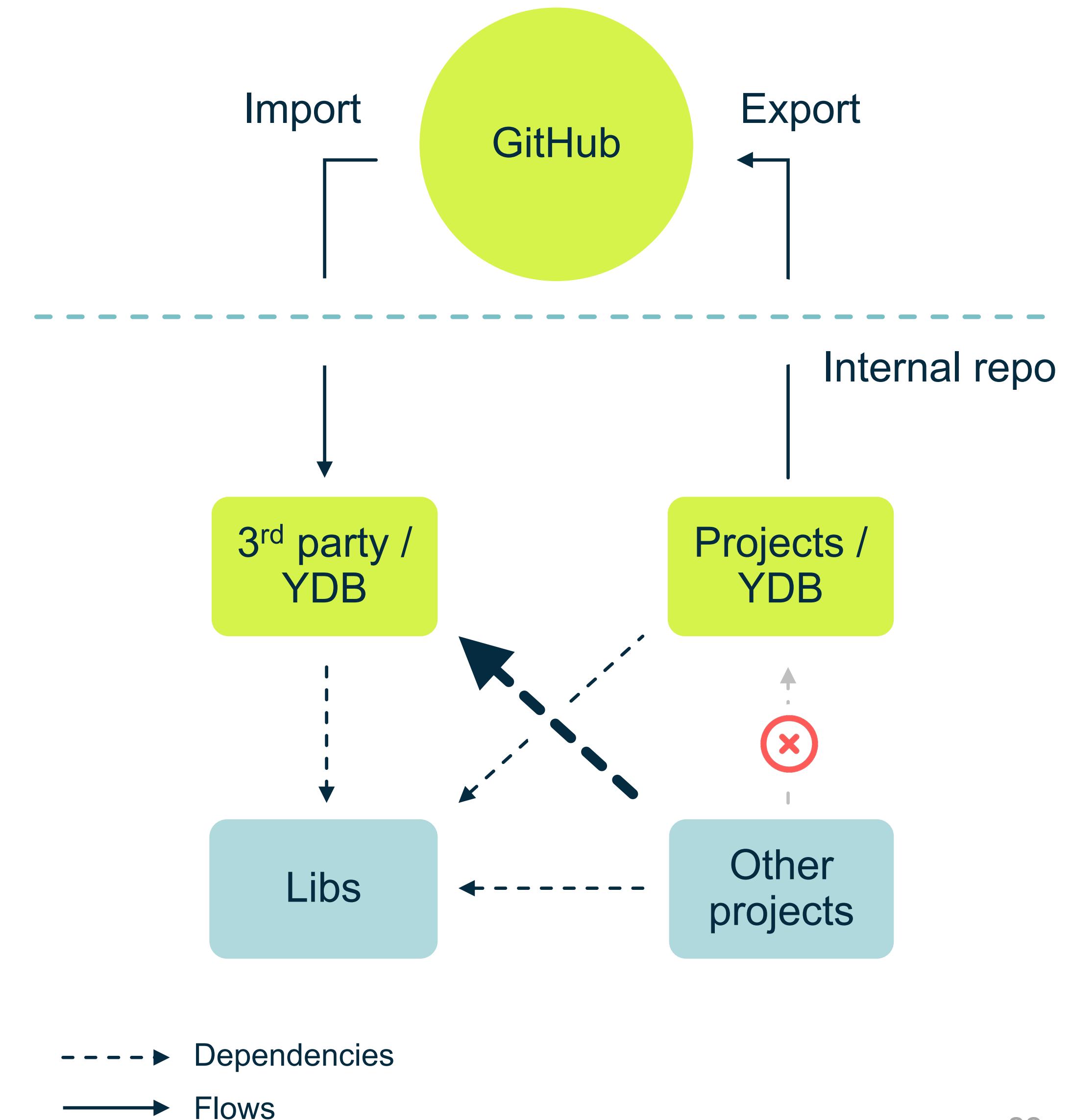
Prepare the switch: Step 2

Move all projects dependencies to YDB code as part of the 3rd party components
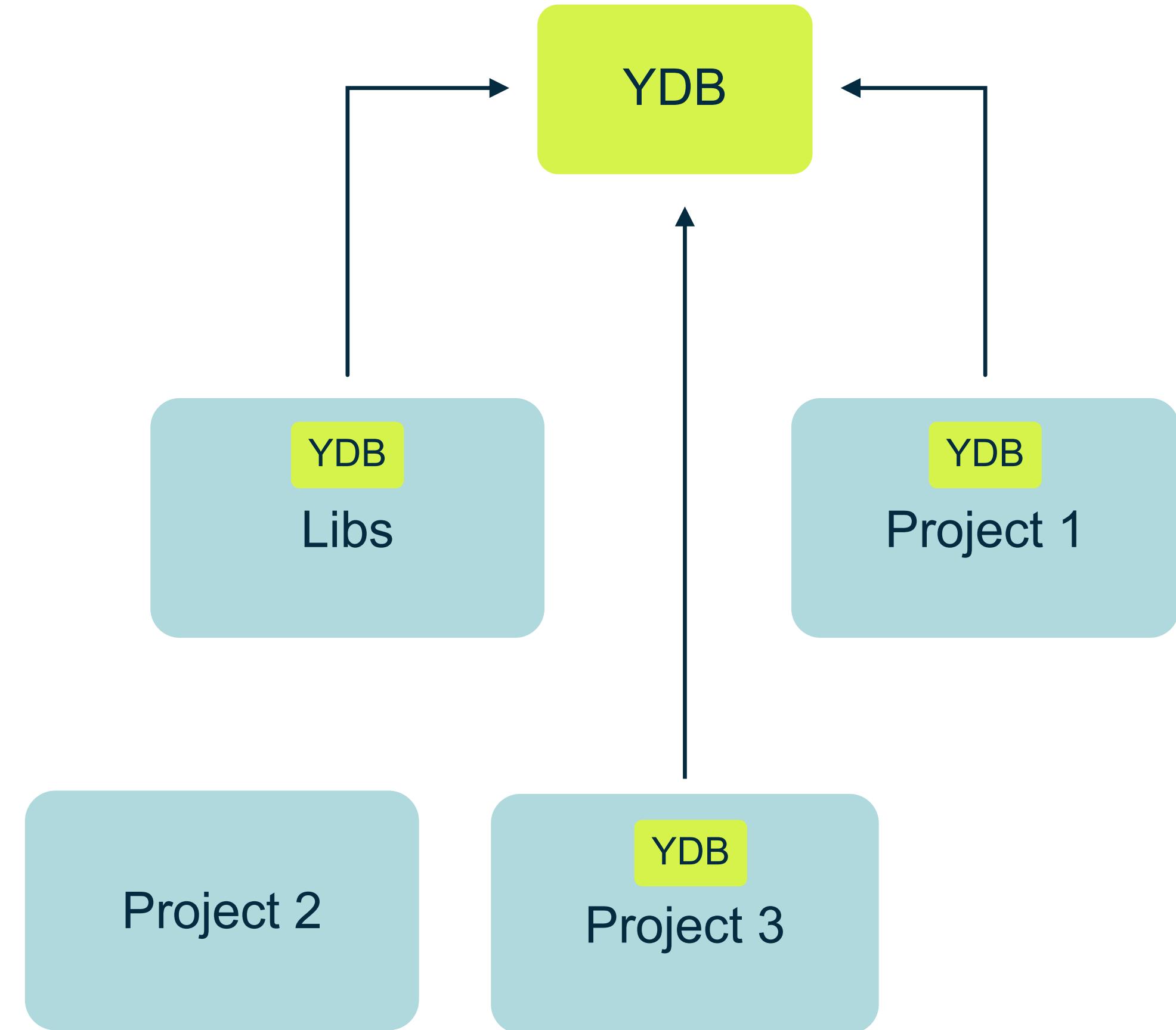
- Iterative, project by project

- Now running all tests of dependent projects on import iterations

- Cease running tests of dependent projects on YDB code changes

GitHub

Import

Export

Internal repo

3rd party / YDB

Projects / YDB

Libs

Other projects

- - - → Dependencies

──→ Flows

22

The real development perimeter unveiled
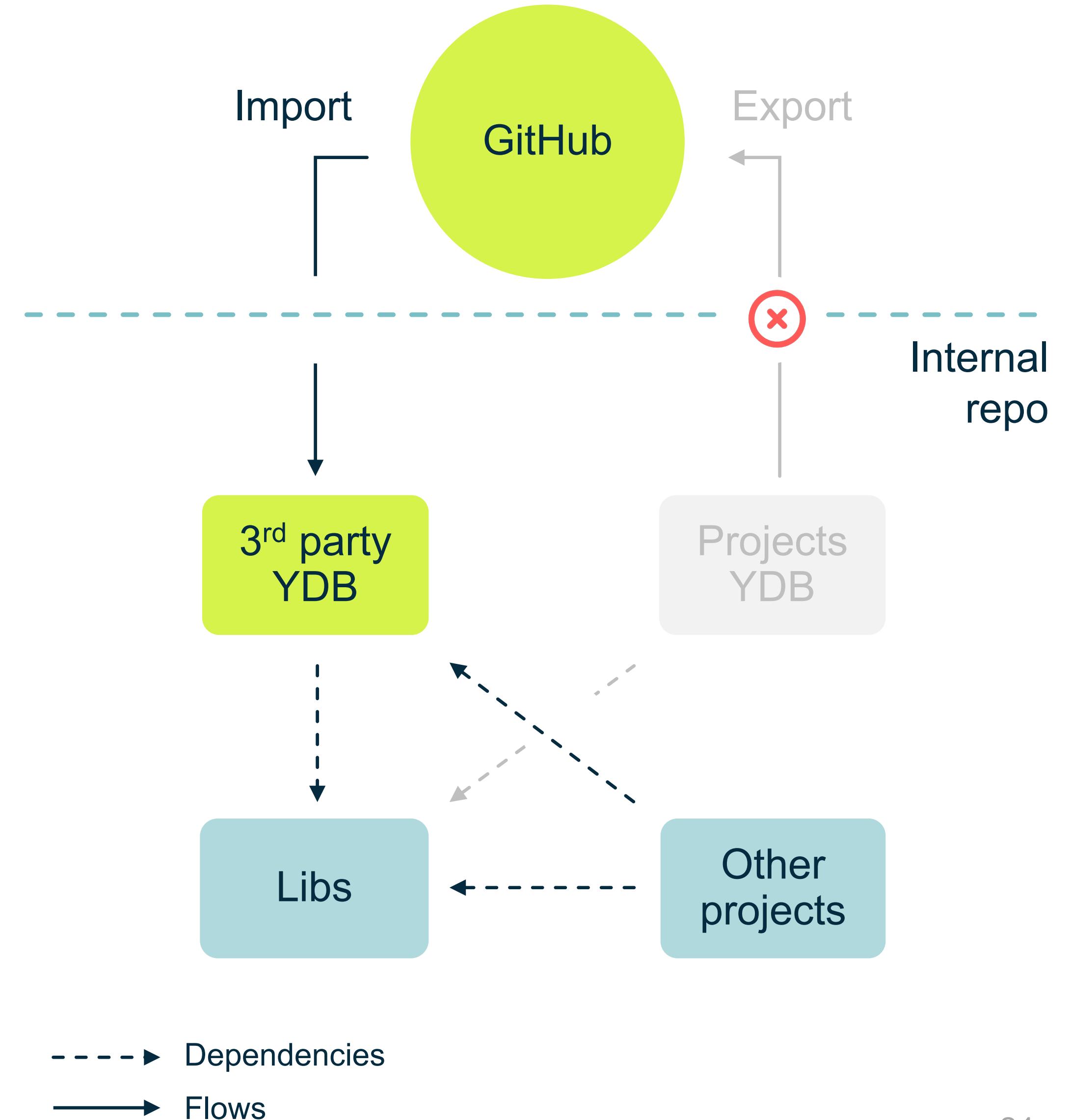
- Highly-cohesive code from common libraries

- More and more tests

- Auxiliary utilities

The switch
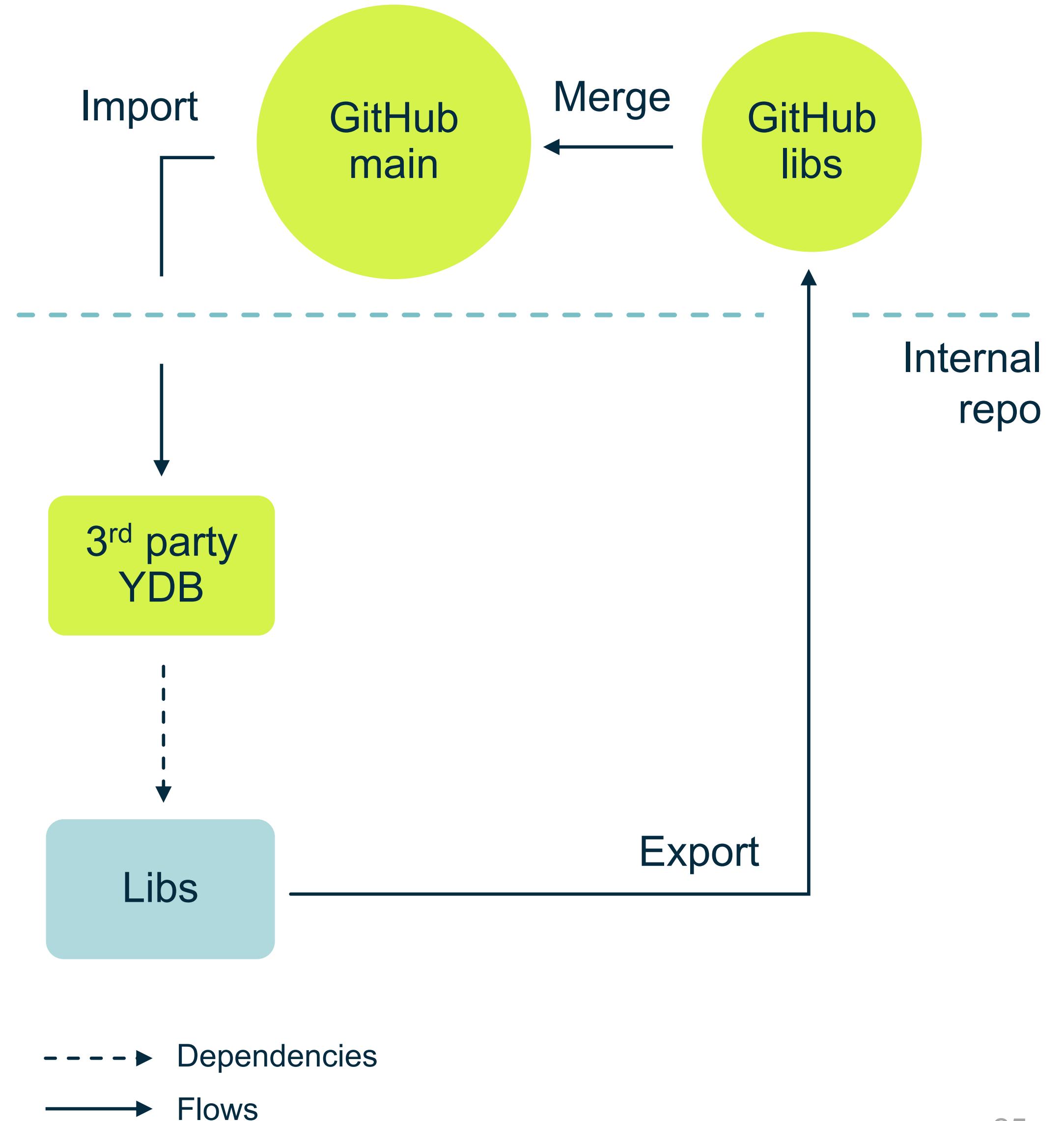
- Switch off YDB sync to GitHub, grant access to merge changes in GitHub

- Development process docs

- Test on a focus group

- A Telegram chat for onboarding

Import

Export

GitHub

Internal repo

3rd party YDB

Projects YDB

Libs

Other projects

- - - - ▶ Dependencies

──────▶ Flows

Libraries

- We depend on libraries from the parent repository.

- Libraries are exported to a special branch on GitHub.

- Libraries are merged to main once a week.

Import

Merge

GitHub main

GitHub libs

Internal repo

3rd party YDB

Libs

Export

- - - → Dependencies

——→ Flows

Achievements

- Full development cycle on GitHub

- One point of truth, one product for many users

- Changes can be verified before submitting

- Infrastructure providers can be replaced

Challenges

- Build for Windows is still CMake-only

- Server logs, core dumps to be collected

- Too many tests run every check because of non-optimal dependencies

- Resolve dependency on libraries

And many more…

Product vision

Sets the goals

Smooth changes

Keep morale high
Mitigate risks one by one

YDB in Nebius

We use it extensively
We contribute to it

YDB - Welcome to contribute
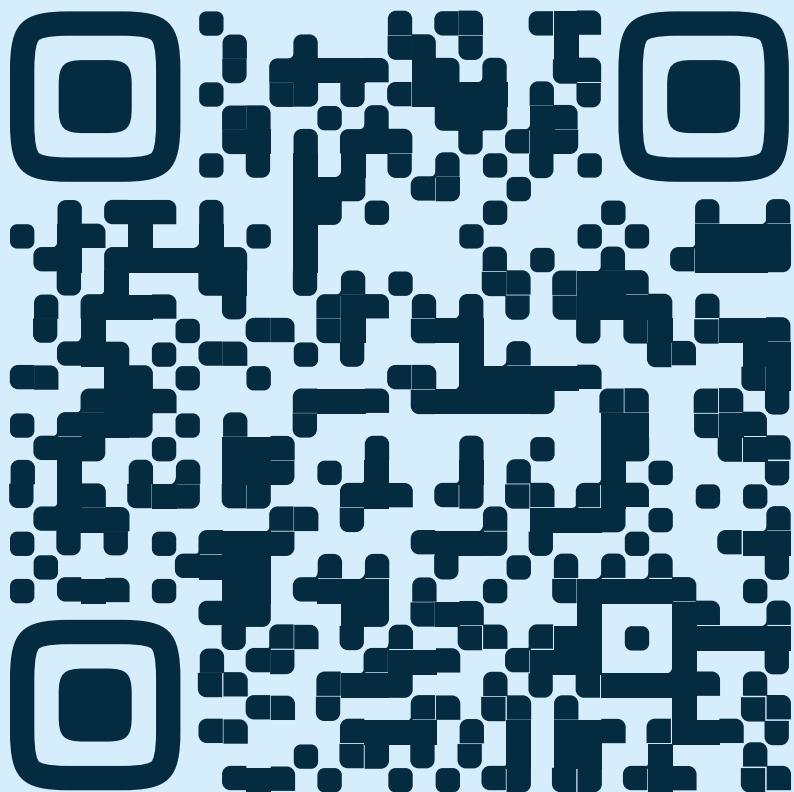
ydb.tech

github.com/ydb-platform/ydb

NEBIUS

Thank you!

Alexander Smirnov
Technology Expert, Nebius
alexv-smirnov@nebius.com

YDB - Welcome to contribute

ydb.tech

github.com/ydb-
platform/ydb