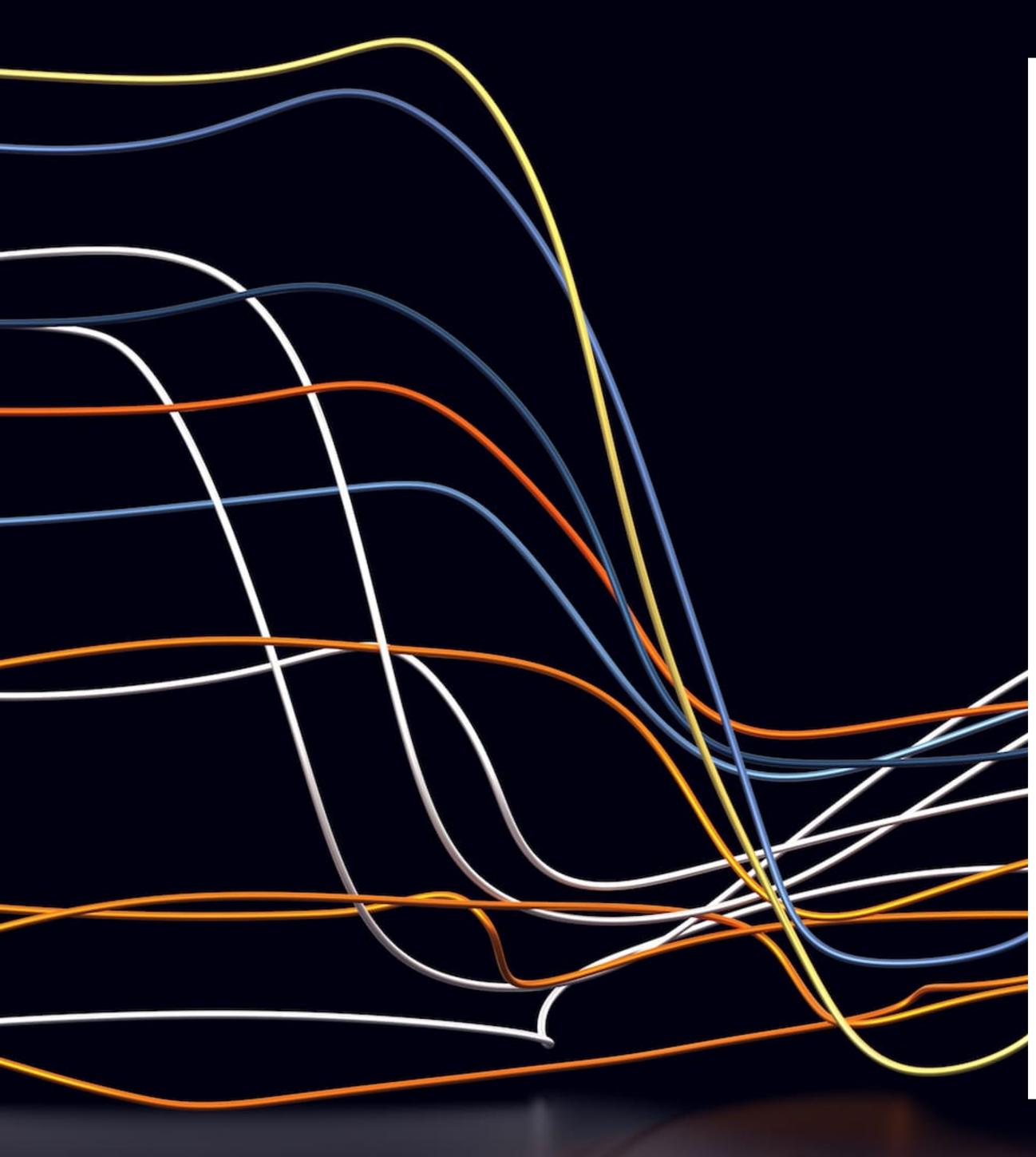


MunichJS Meetup 2024

Develop CI/CD-pipelines locally in TypeScript with dagger.io

INNOQ

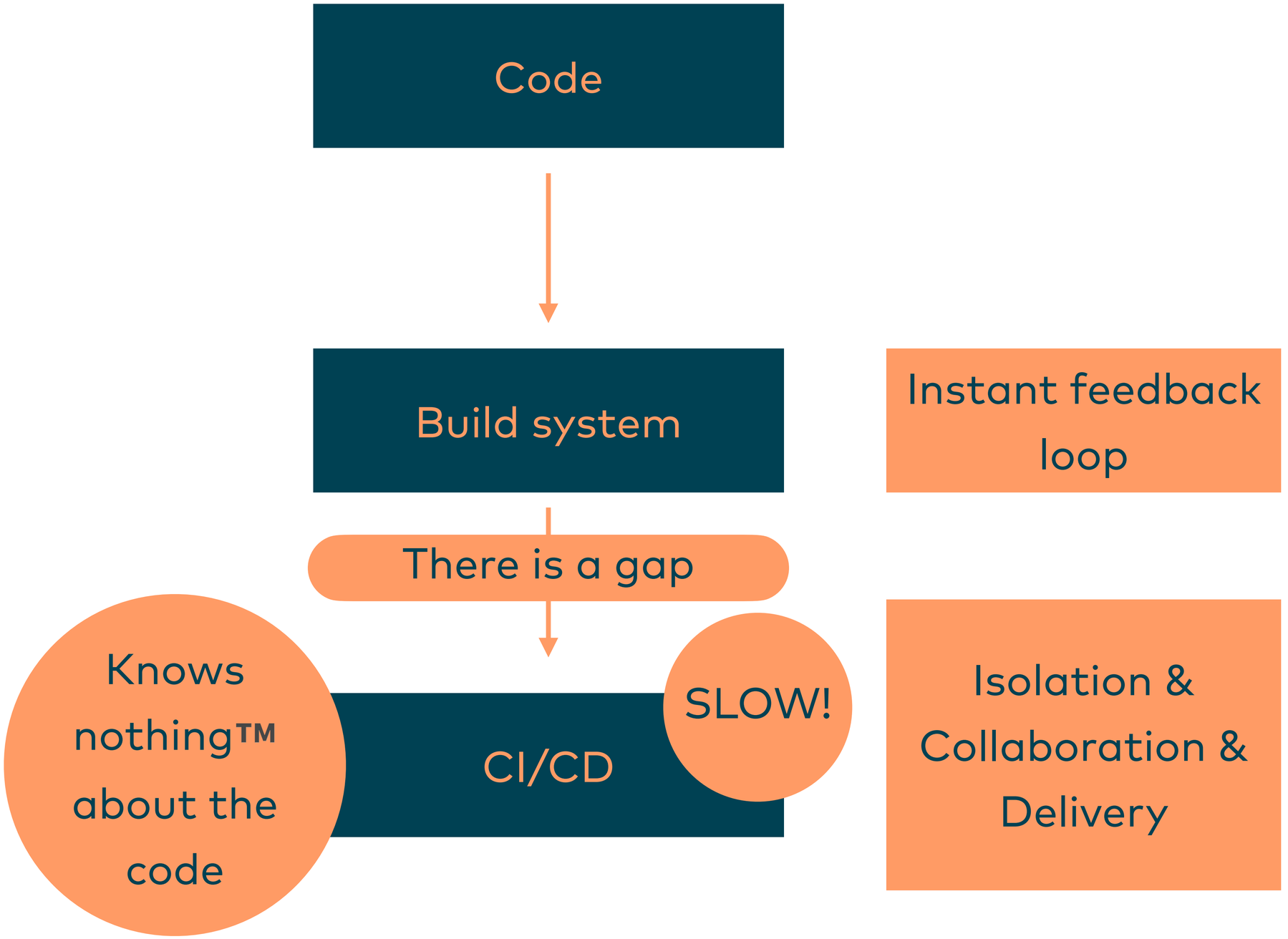
FABIAN KRETZER
INNOQ.SOCIAL/@FABIAN

Our journey

- Why?
- Origins
- Building blocks
- Concepts
- Example
- Future
- Opinion(s)

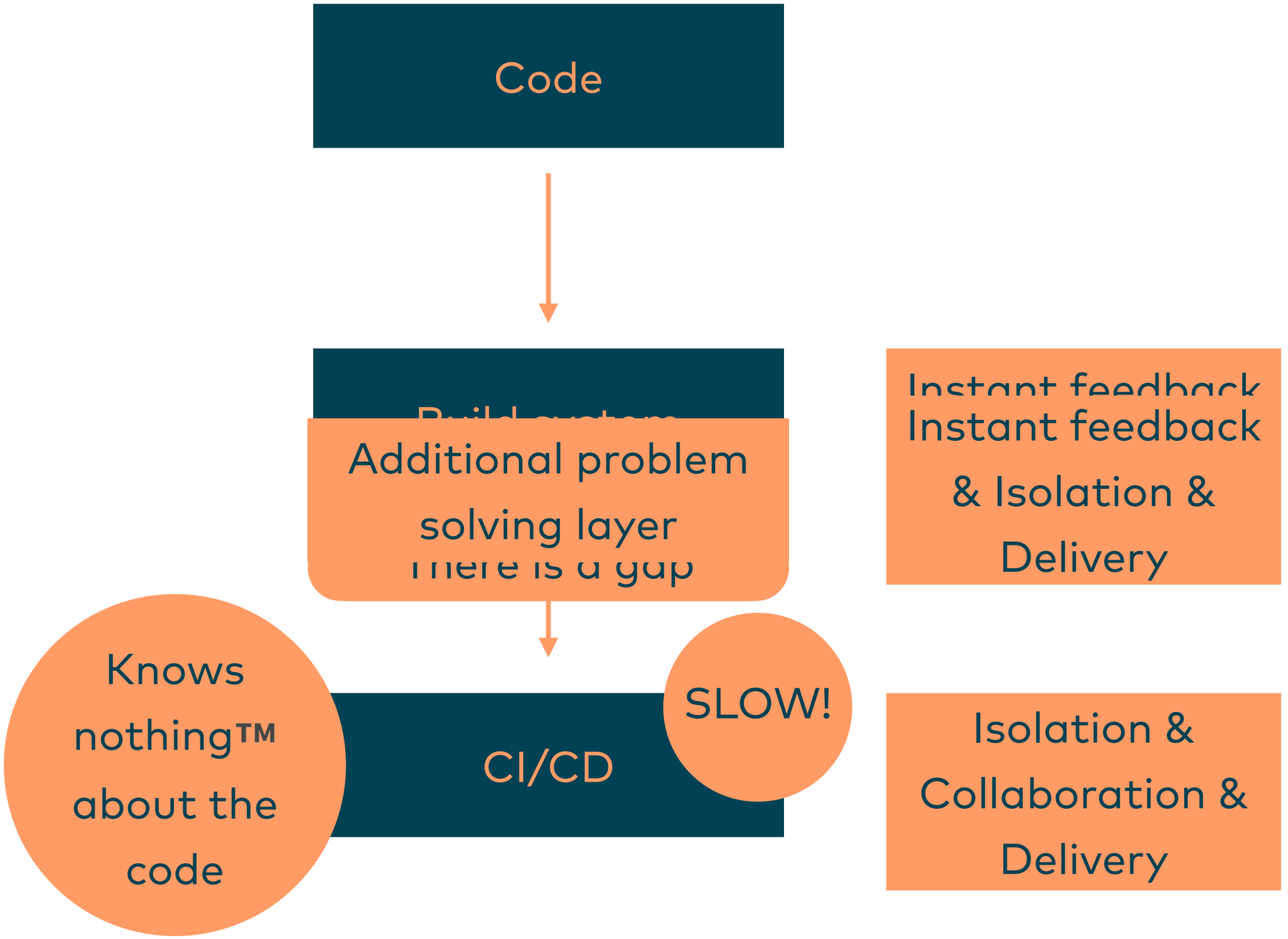
© dagger.io

But why?!


Code
Build system
Instant feedback loop
There is a gap
Knows nothing™ about the code
SLOW!
CI/CD
Isolation & Collaboration & Delivery

„Everything can be solved by an additional layer of indirection“

- *Unknown wise person*


Code
Build system
There is a gap
Additional problem solving layer
Instant feedback & Isolation & Delivery
Knows nothing™ about the code
CI/CD
SLOW!
Isolation & Collaboration & Delivery

Imperative vs. declarative

- Gradle vs. Maven vs. Jenkinsfile vs. .gitlab-ci.yml
- Its not a binary decision, but a continuum
- Reduce mental load -> Shift complexity to different layers
- Don’t hide complexity, but establish clear boundaries

Why – Summary

- Save interface between Build and CI
- Local development with…
- … Instant feedback loop

I don’t want to replace neither build nor CI/CD systems, but bridge nicely between them while solving some problems of both systems along the way.

The origin story

From the people that brought you docker

Containers

It' about the developer experience

"Engine lead“ Docker project

BuildKit

Low-Level Build definition format

LLB

„At the core of BuildKit is a Low-Level Build definition format. <...>

<LLB> defines a content-addressable **dependency graph** that can be used to put together very **complex build definitions**.

It also supports features not exposed in Dockerfiles, like direct data mounting and nested invocation. <...>

Everything about execution and caching of your builds is defined in LLB“

cuelang.org
Honorable mention

The first approach

- Built upon ~15 years of experience with Google GCL
- **Combine constraints** from different sources to produce a **deterministic output**
- Bonus: Comparing schemas for backwards compatibility
- Limited scripting: explicitly **constrained** -> converges to a **valid state in finite time**

https://cuelang.org/docs/about/#history

Pivot

Language specific SDKs

The second approach

- Arcane cuelang syntax -> adoption barrier
- „DevEx-First“ -> Let the people live where they feel at home -> let them use their day-to-day tools / languages
- SDKs generated from API schema
- Mental distinction between programming language syntax and dagger concepts easier to grasp in an environment you know well

Origins – Summary

- People with right™ mindset
- Mature foundational technologies
- Courage to do a pivot to get better DevEx
- Everything gets better if you throw container technology at it ;-)

Success of a technology is determined by its accessibility

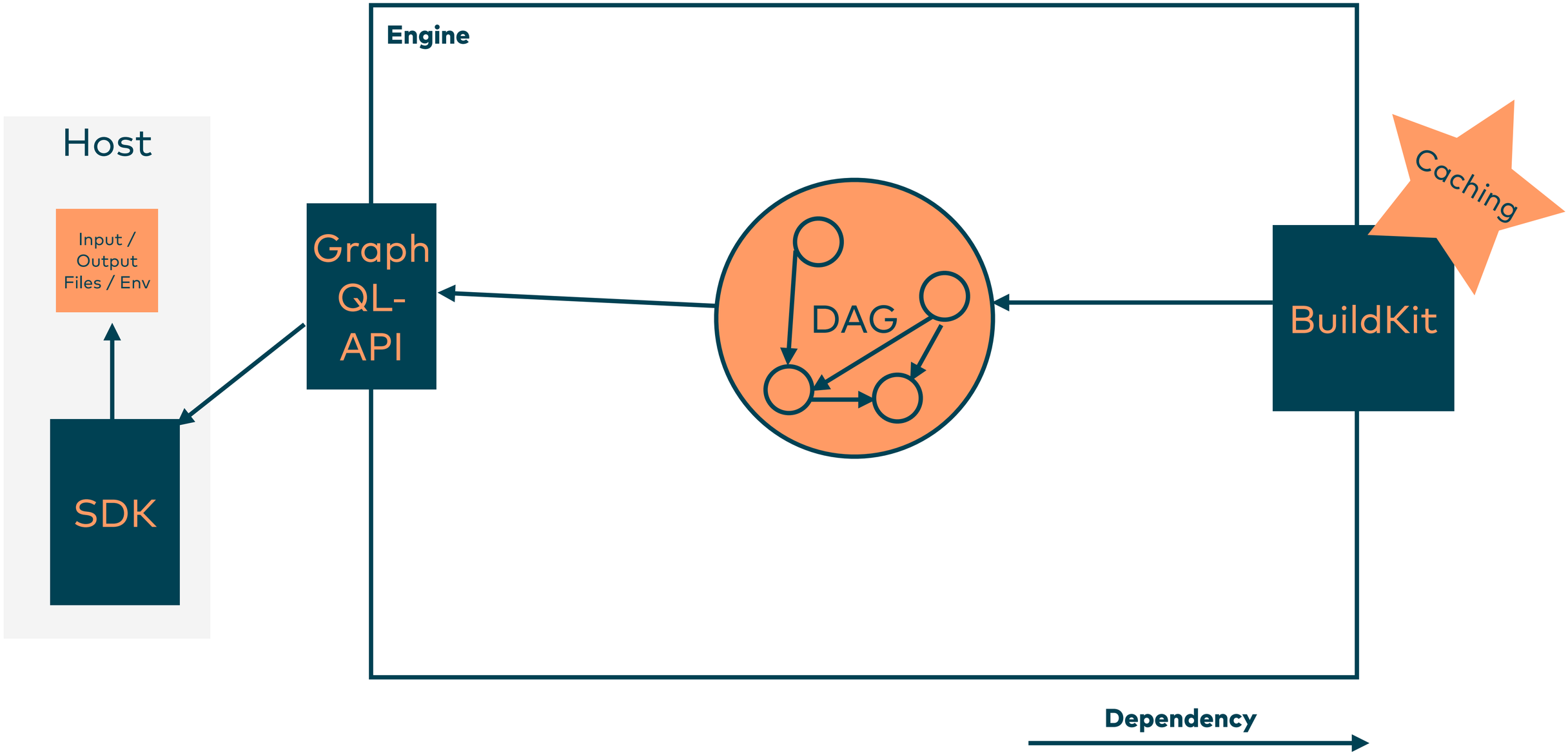
The building blocks

Concepts

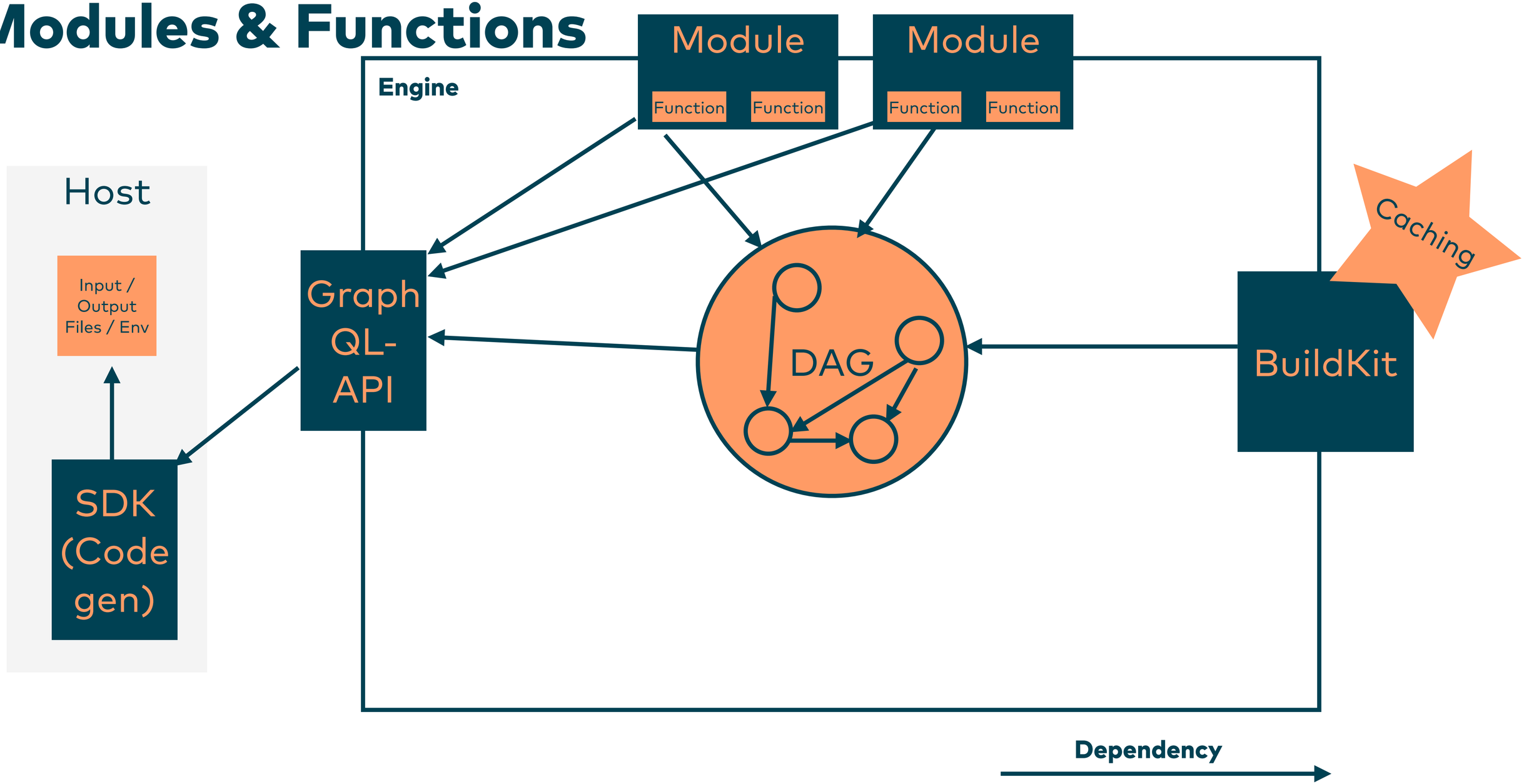
- SDKs
- GraphQL-**API**
- dagger engine with the DAG

dagger.io

Using a directed acyclic graph to our advantage


Engine
Host
Input / Output Files / Env
SDK
Graph QL-API
DAG
BuildKit
Caching
Dependency

Modules & Functions



Concepts – Summary

- SDKs
- GraphQL-**API**
- Docker engine & BuildKit:
  - concurrency & caching „for free“

Example - Lets blog!

- Build static site with goHugo
- Optimize images before deployment
- Build and (re)use dagger modules

dagger.io

The future (is now)

- More SDKs
- Cloud UI / Caching (business case)
- More Modules
- CLI-Tooling for Modules & Functions is finally here!
- „Cross-Language-Orchestration“ is now possible
- https://daggerverse.dev/

dagger.io

Opinions

- Good mixture of people, mindset, concepts and foundational technology
- Boundary between imperative and declarative layers is good (enough)
- Not revolutionary technology wise, but an evolution and very clever amalgamation of existing technologies

Getting Started!

- Getting started: https://docs.dagger.io/
- Discord: https://discord.gg/ufnyBtc8uY
- First steps: https://docs.dagger.io/quickstart/562821/hello
- Example: https://github.com/fkretzer/mujs24

Feedback?

- Used dagger.io?
- Can recommend similar / alternative tools?
- Declarative vs. imperative vs. mix of both?

- **Thanks for your attention!** ❤️

Feedback? Contact!

Fabian Kretzer

fabian.kretzer@innoq.com

innoq.social/@fabian

INNOQ

www.innoq.com

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstrasse 13
20095 Hamburg

Erftstr. 15-17
50672 Köln

Königstorgraben 11
90402 Nürnberg