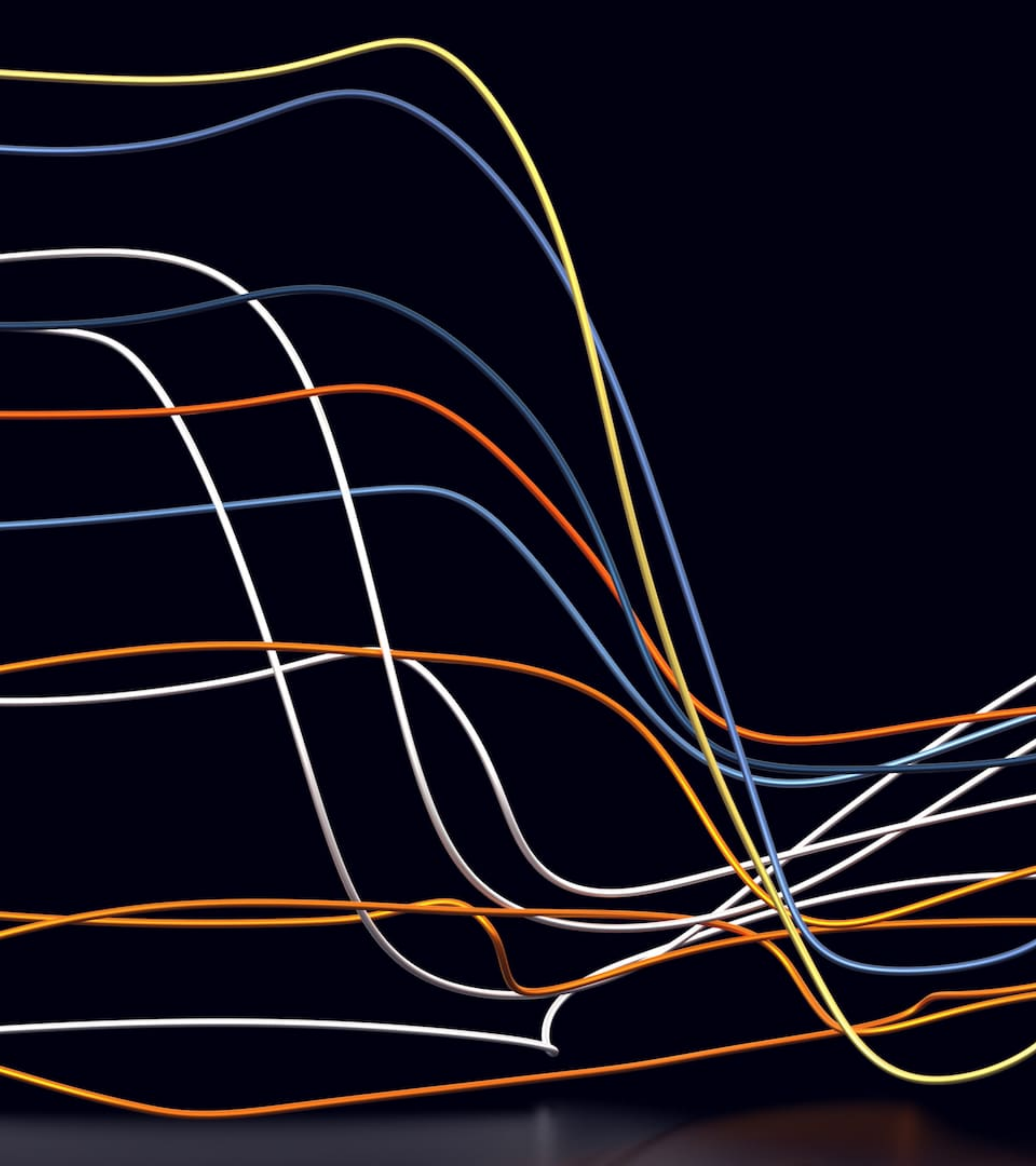




Code Days 2024

Programmierbare CI/CD-Pipelines lokal entwickeln

— mit dagger.io

INNOQ



FABIAN KRETZER
INNOQ.SOCIAL/@FABIAN

Worüber sprechen wir?

- Warum?
- Wo kommt das her?
- Grundpfeiler
- Konzepte
- Beispiel
- Zukunft
- Meinung(en)



© dagger.io

Warum?!

Code



Build system

Schnelles
feedback

Hier ist eine Lücke



Versteht™
den Code nicht

Langsam

CI/CD

Isolation &
Zusammenarbeit
& Delivery

„Alle Probleme können durch eine weitere Ebene von Indirektion gelöst werden.“

- Unbekannte schlaue Person

Code



Build system

„Problemlösungslayer“

hier ist eine Lücke



Versteht™
den Code nicht

CI/CD

SLOW!

Schnelles
Feedback

Isolation &
Zusammenarbeit
& Delivery

Imperativ vs. deklarativ

- Gradle vs. Maven vs. Jenkinsfile vs. .gitlab-ci.yml
- Keine binäre Frage, sondern ein Kontinuum
- Mentale Last verringern, Komplexität auf verschiedene Ebenen verteilen
- Komplexität nicht verstecken, sondern klare konzeptionelle Grenzen definieren
- Mein Ursprünglicher Zugang zum Thema: Cuelang

Warum – Zusammenfassung

- Zugängliches Interface zu und zwischen Build and CI mit klaren Verantwortlichkeiten
- Lokale Entwicklung mit...
- ... schnellem feedback

Ich möchte Build- und / oder CI/CD-Systeme nicht ersetzen, sondern eine Brücke zwischen ihnen bauen. Und nebenbei noch ein paar andere Probleme lösen.

Die Ursprünge

From the* people that brought you docker

*some

Container

It' all about the
developer
experience



Arnaud Porterie @arnaudporterie · 10. Juli 2019



Maybe the real treasure was the developer experience we made along the way.

"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"

<https://docs.docker.com/build/buildkit/#llb>

cuelang.org

Honorable mention

1. Ansatz: so deklarativ wie möglich, so „skriptbar“ wie nötig

- Aus Google GCL entstanden
- Schemata aus unterschiedlichen Quellen kombinieren und deterministisches Ergebnis erzeugen
- Bonus: Schemata auf Rückwärtskompatibilität prüfen
- Bewusst beschränkte Skriptingmöglichkeiten: Gültiger Zustand in endlicher Zeit
- <https://cuelang.org/docs/about/#history>

Shift

Zu

Sprachspezifischen SDKs

2. Ansatz: SDKs durch Codegeneration

- cuelang zu große Hürde
- DevEx-First -> Nutzende abholen wo sie stehen -> direkt in den jeweiligen Sprachökosystemen
- SDKs generiert aus API-Schema
- Mentale Trennung von Syntax und dagger-Konzepten für Nutzende einfacher in „Ihren“ Sprachen

Ursprünge – Zusammenfassung

- Leute mit dem richtigenTM mindset
- Mut zum Pivot im Sinne der DevEx
- Ausgereifte Basistechnologien
- Alles wird besser, wenn wir
Containertechnologie auf unsere Probleme
werfen 😊

**Erfolg einer Technologie wird durch ihre Zugänglichkeit
bestimmt**

Grundpfeiler

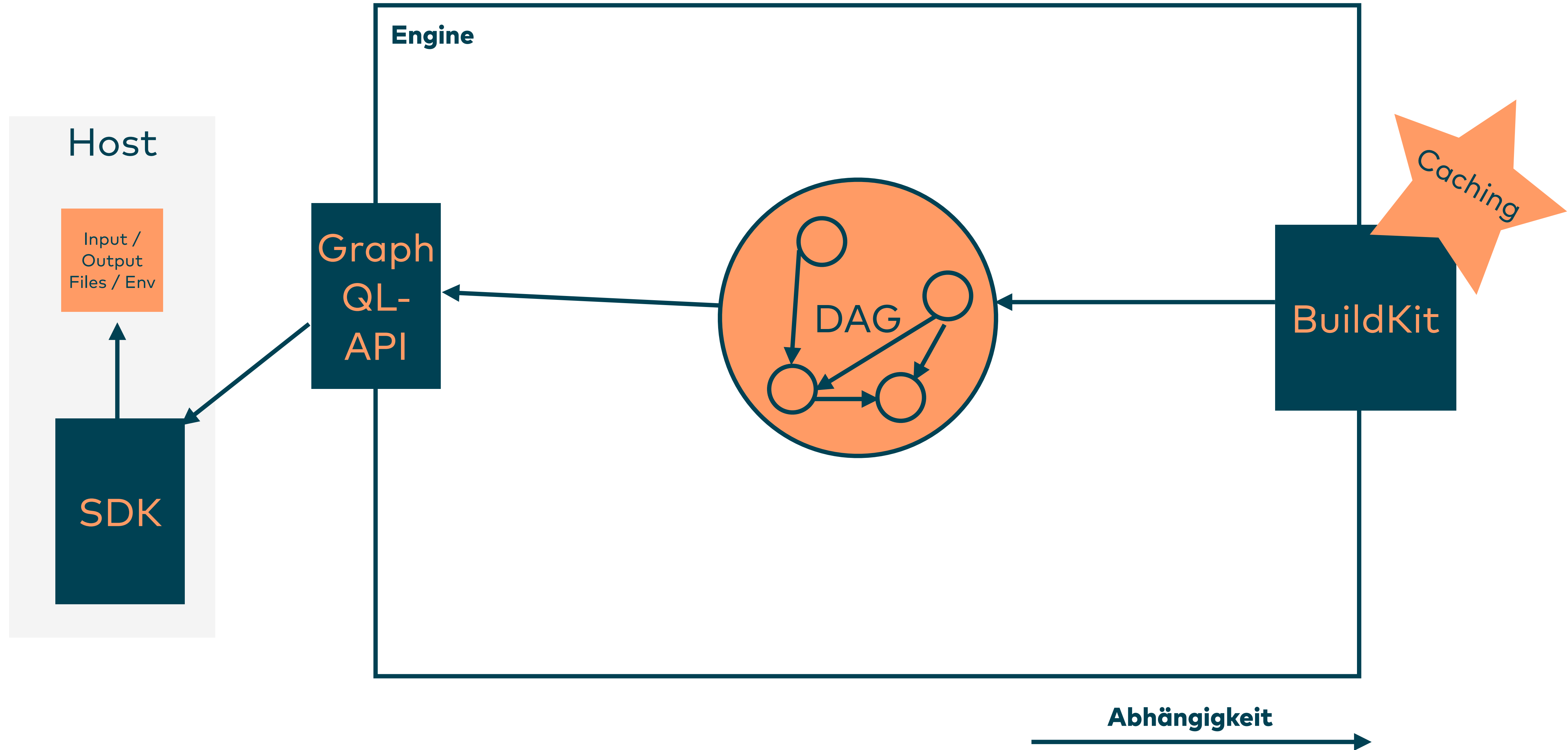
Konzepte

- SDKs
- GraphQL-API
- **Die Engine mit dem DAG**



dagger.io

directed acyclic graph



Konzepte – Zusammenfassung

- Sprachspezifische SDKs
- (GraphQL)-API
- „Gratis“ caching durch Container Engine & BuildKit

Beispiel - Lets blog!

- Statische Website mit goHugo
- Bilder durch pngquant optimieren



dagger.io

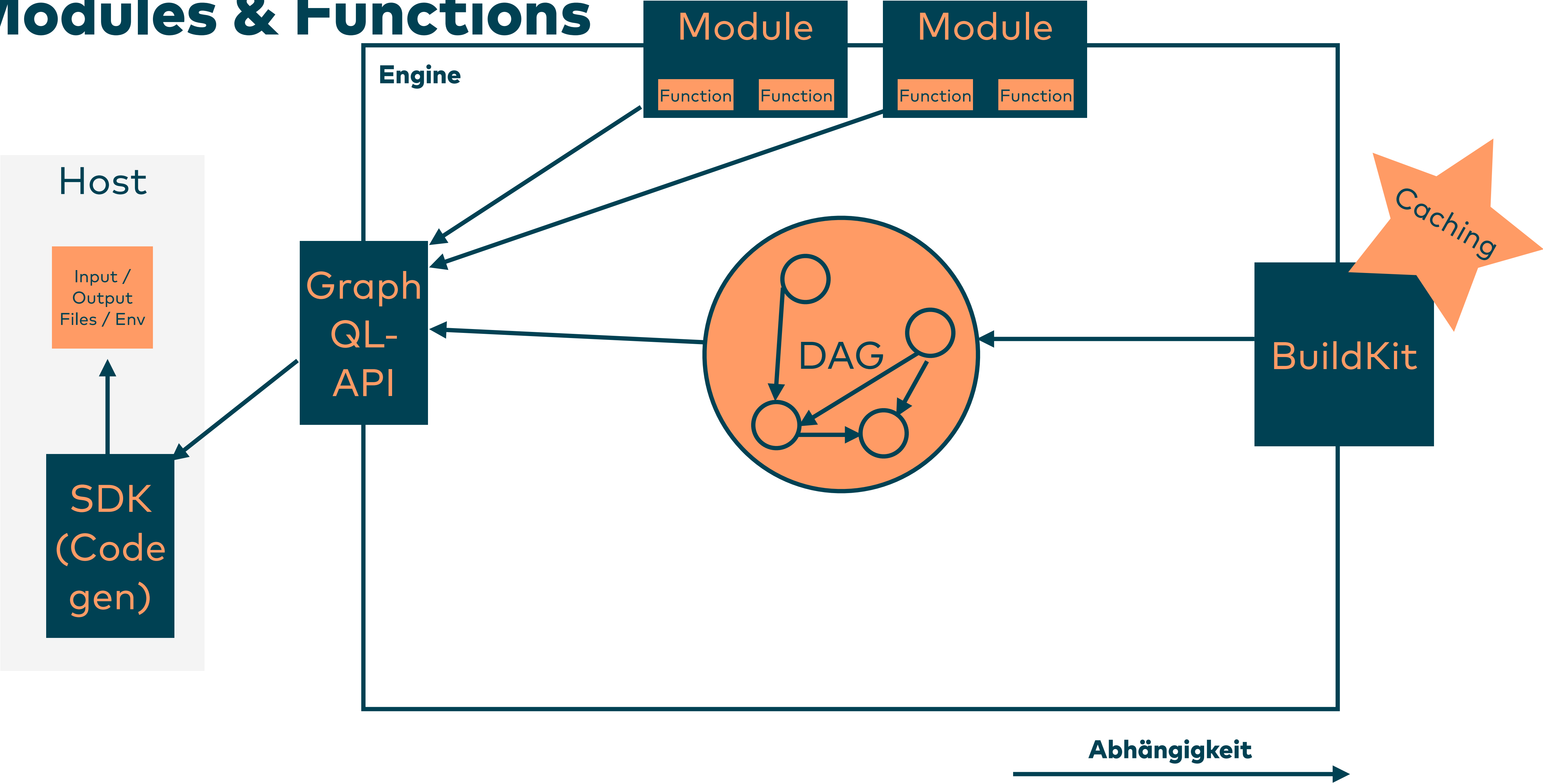
Zukunft

- (Mehr) SDKs
- Cloud UI / Caching
(Kommerzialisierungsstrategie)
- CLI-Tooling mit Modules & Functions
- Echte „Cross-Language-Orchestration“
- Nächste Abstraktionsebene



dagger.io

Modules & Functions



Meinung(en)

- Guter Mix von mindset, Konzepten und Basistechnologien
- Konzeptionelle Grenze zwischen Deklaration der Abhängigkeiten über die GraphQL-API und Ausführung in der Engine
- Keine Revolution, aber eine gute Evolution und Vermengung existierender Technologien



Loslegen!

- Getting started: <https://docs.dagger.io/>
- Discord: <https://discord.gg/ufnyBtc8uY>
- Start mit CLI: <https://docs.dagger.io/cli/389936/run-pipelines-cli>
- Beispiel aus dem Talk: <https://github.com/fkretzer/daggercd2024>



Feedback?

- dagger.io schon benutzt?
- Ähnliche Tools mit ähnlichem Fokus?
- Deklarativ, Imperativ, Beste Mischung?
- **Vielen Dank für die Aufmerksamkeit!** ❤️

Feedback? Contact!



Fabian Kretzer

fabian.kretzer@innoq.com

innoq.social/@fabian

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