Modularizing a 10-Year Monolith

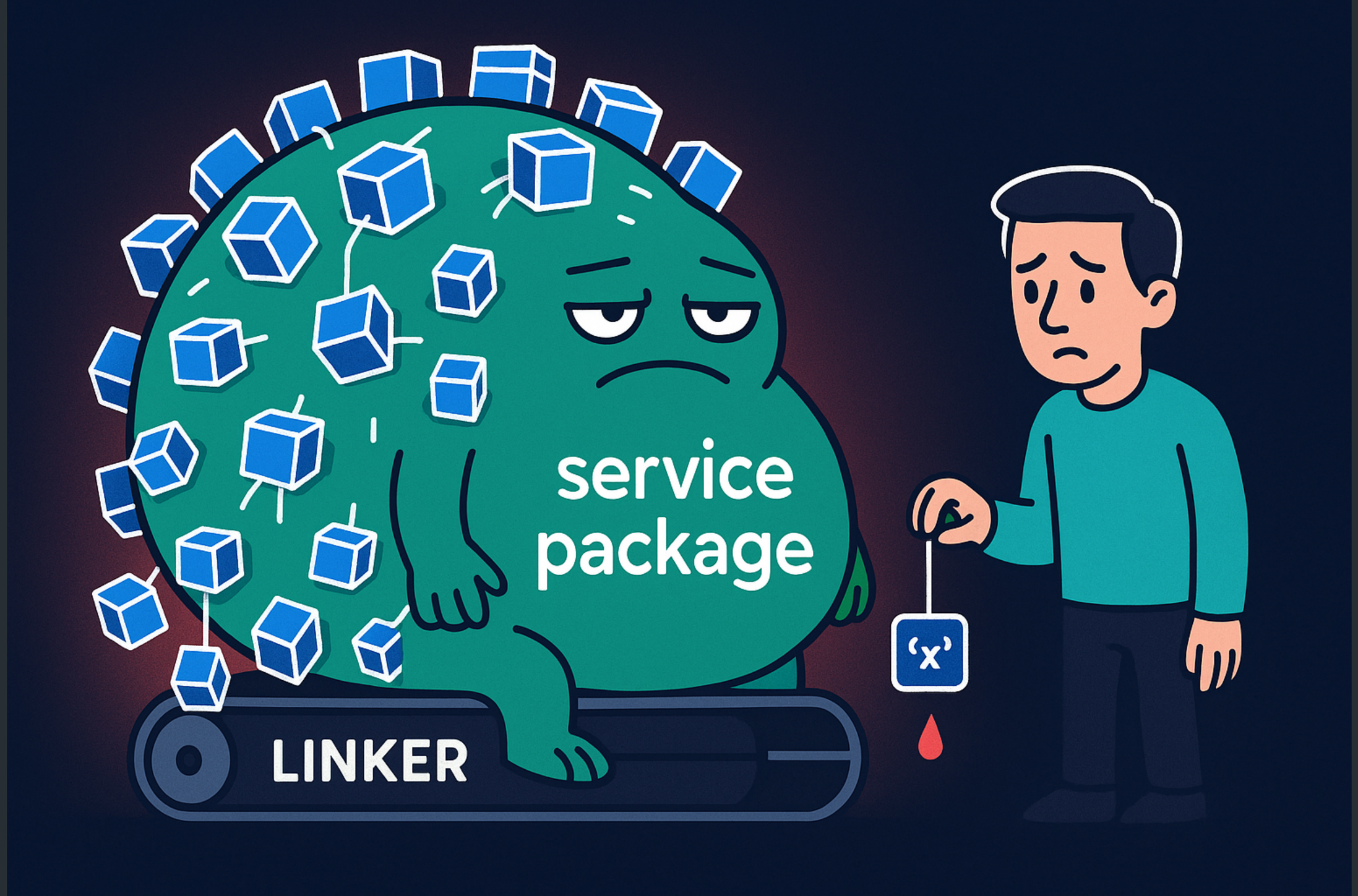
The Architecture, the People, and the Pain

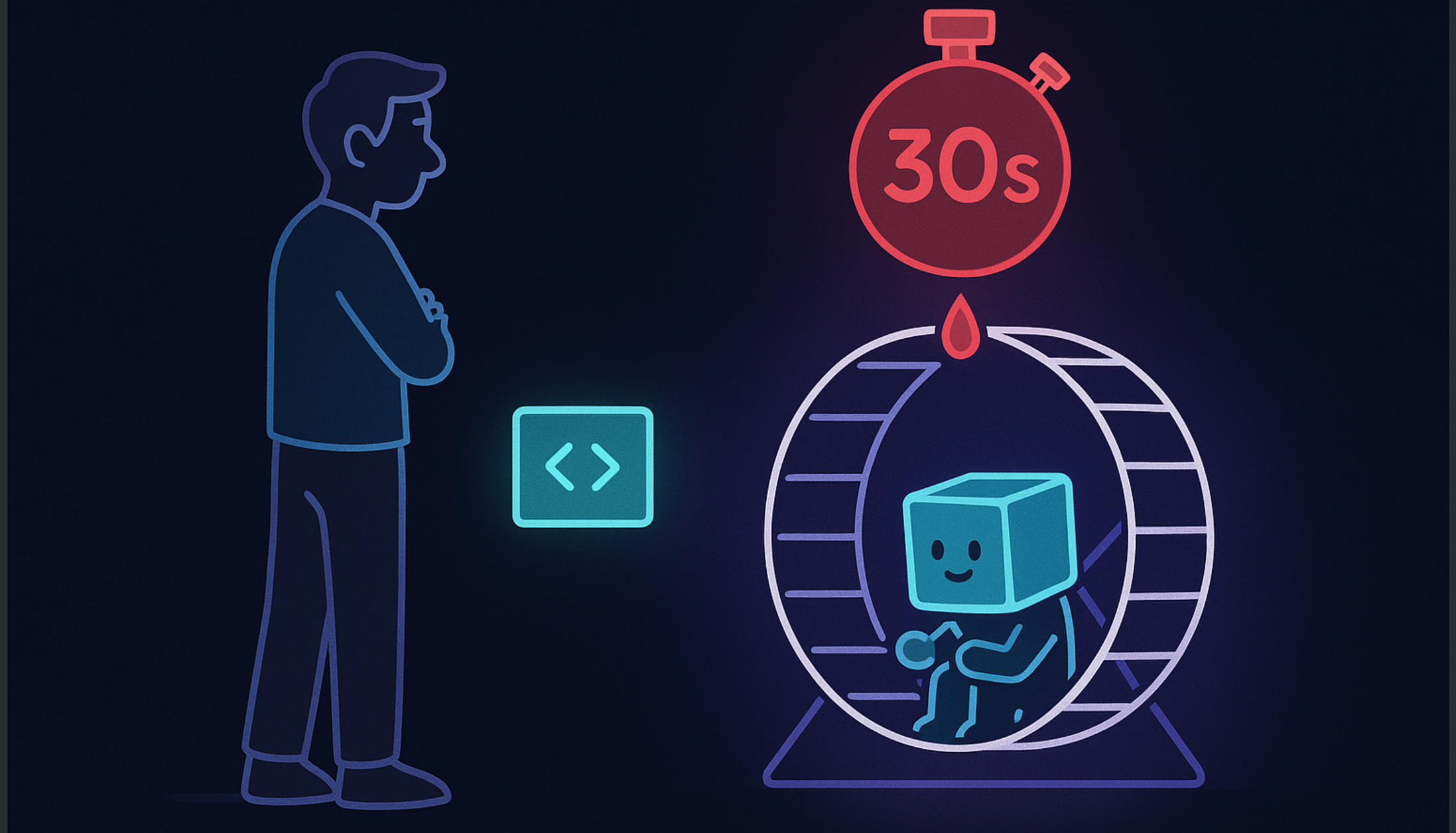
Victor Lyuboslavsky, CodeMash 2026

fleet

Bleeding out 30 seconds at a time

fleet


service
package
LINKER
'x'


30s

Small code moves fast.
Small code feels alive.

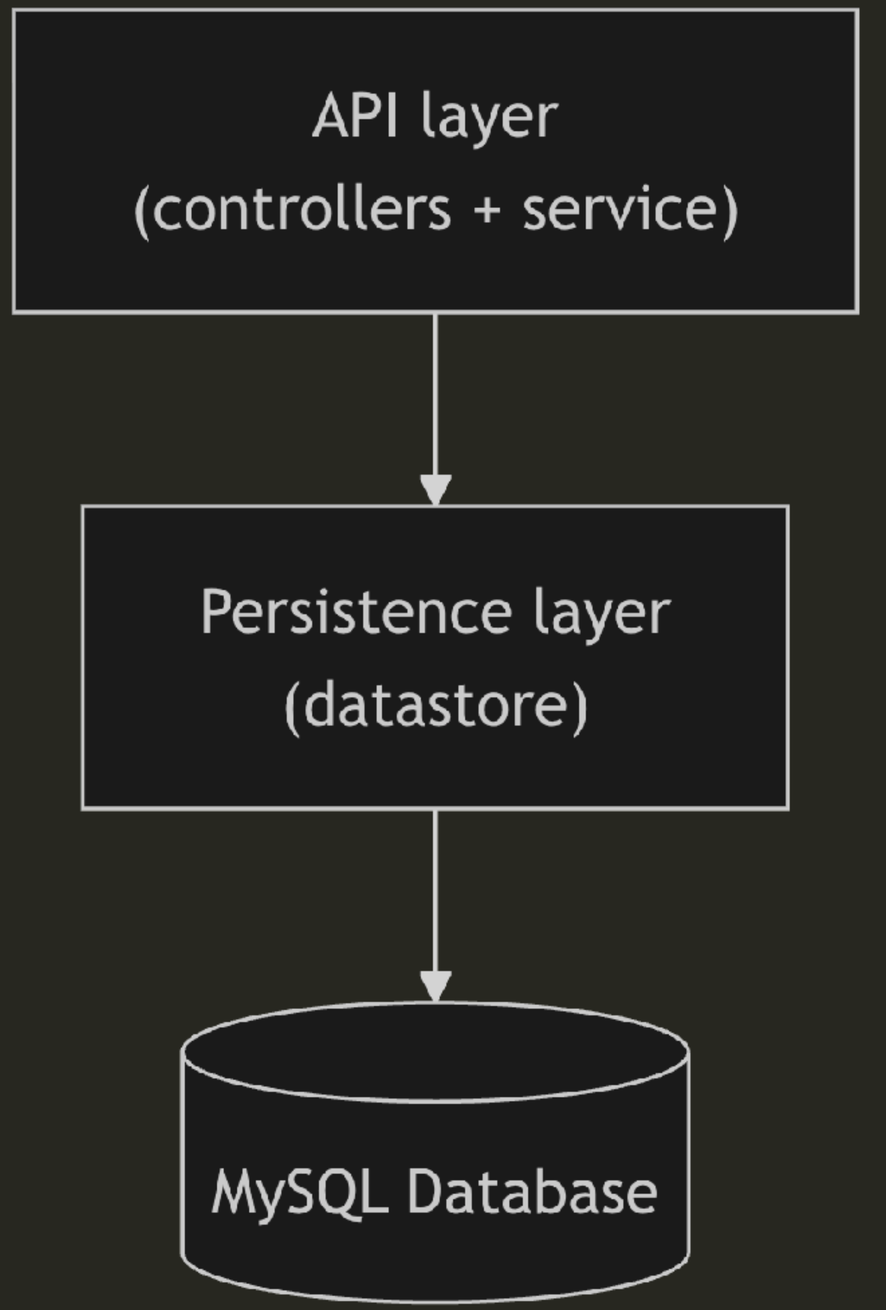
I had 25 years of experience.
Then I realized I actually had one year repeated 25 times.

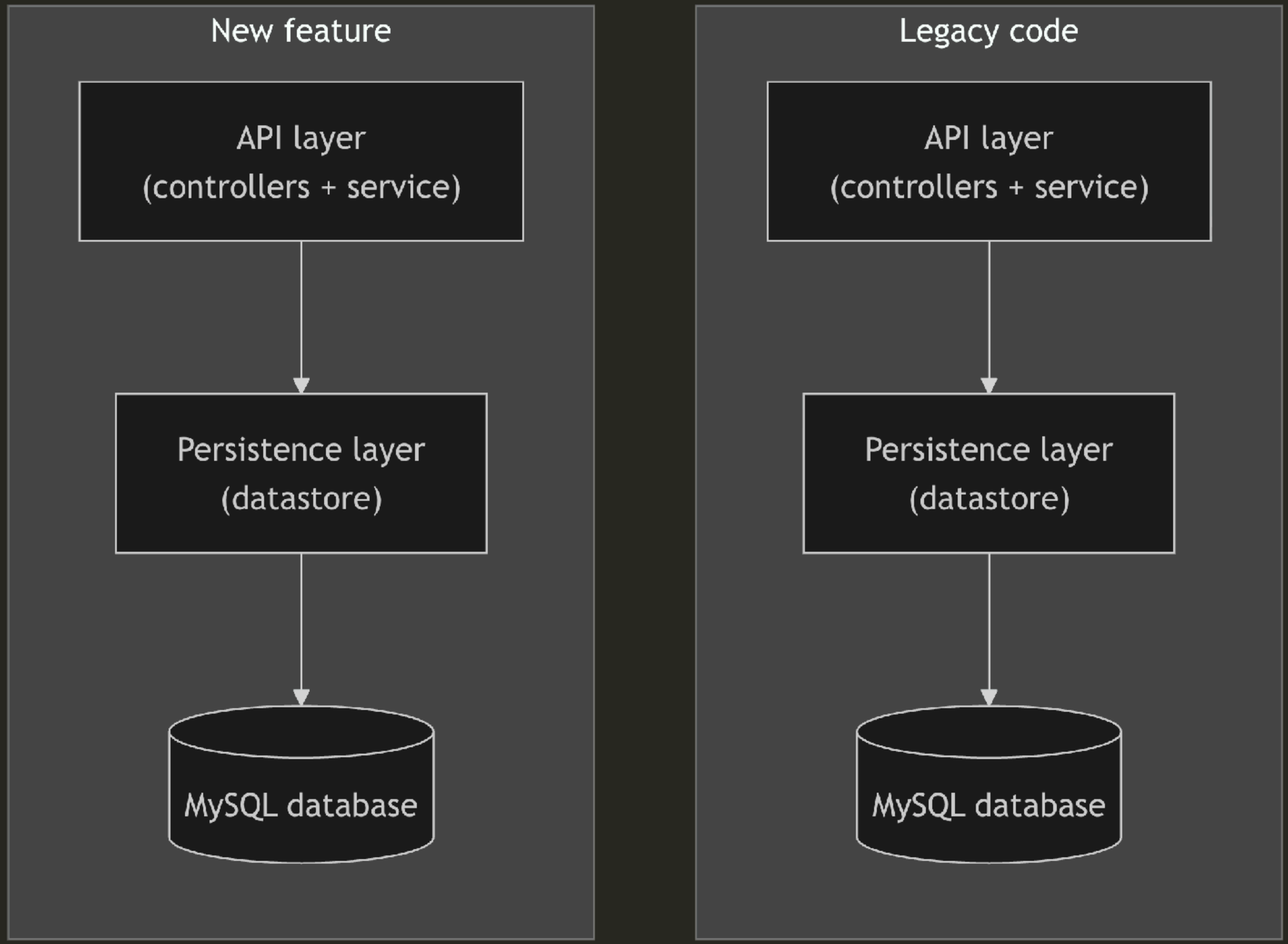
fleet

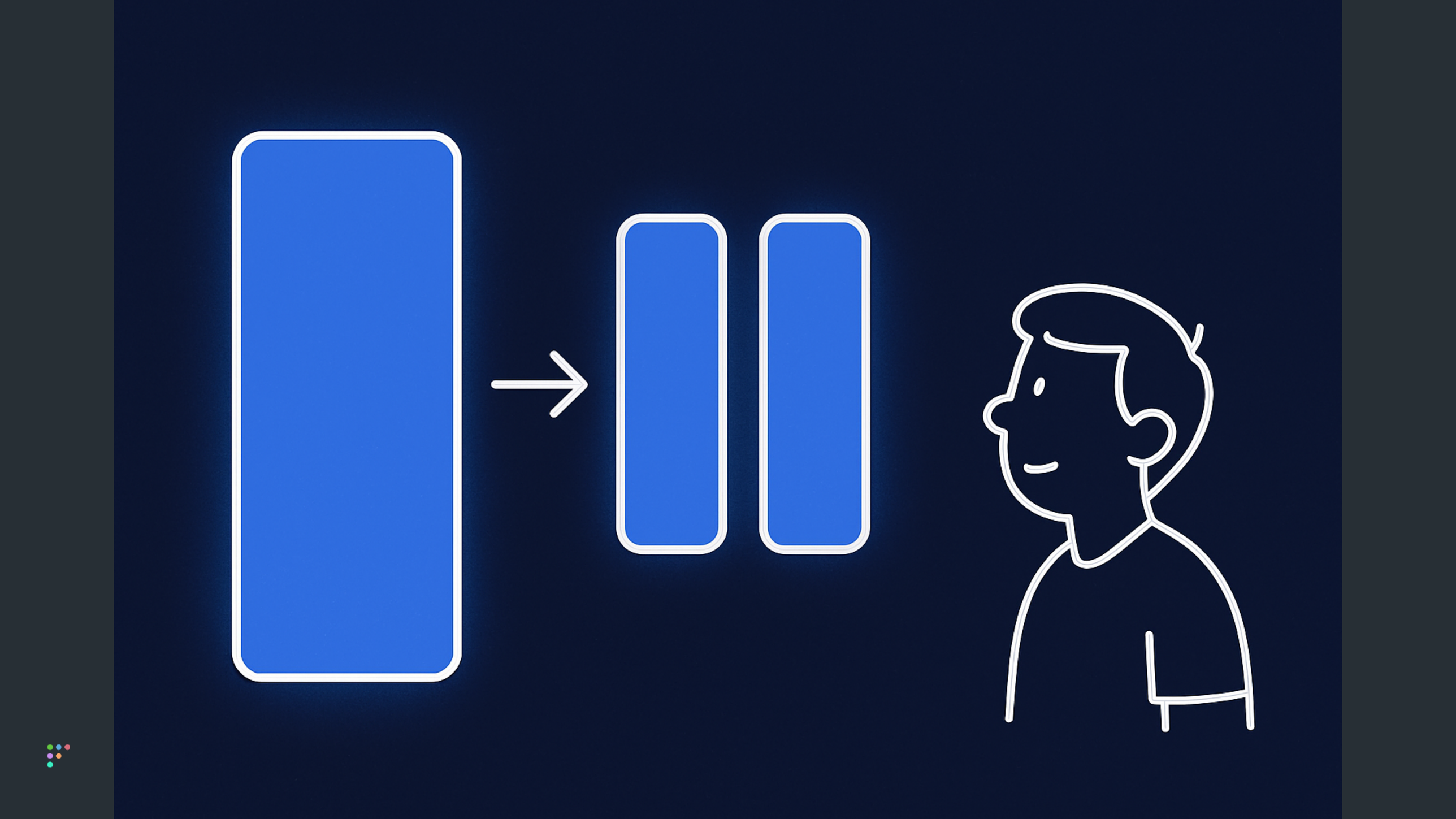
The purpose of software engineering is to control complexity, not to create it

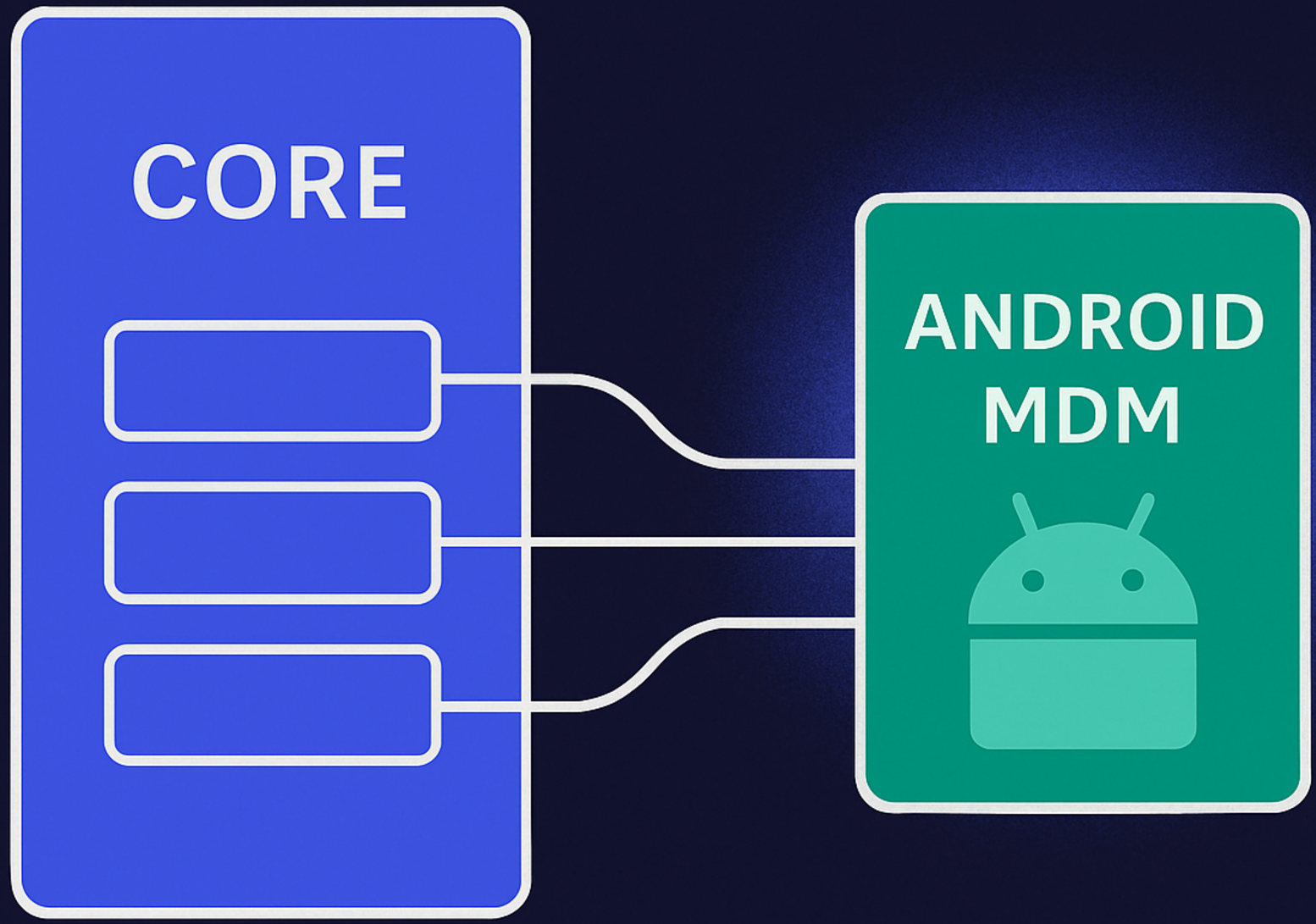
Pamela Zave

fleet


API layer
(controllers + service)
Persistence layer
(datastore)
MySQL Database


New feature
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database
Legacy code
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database

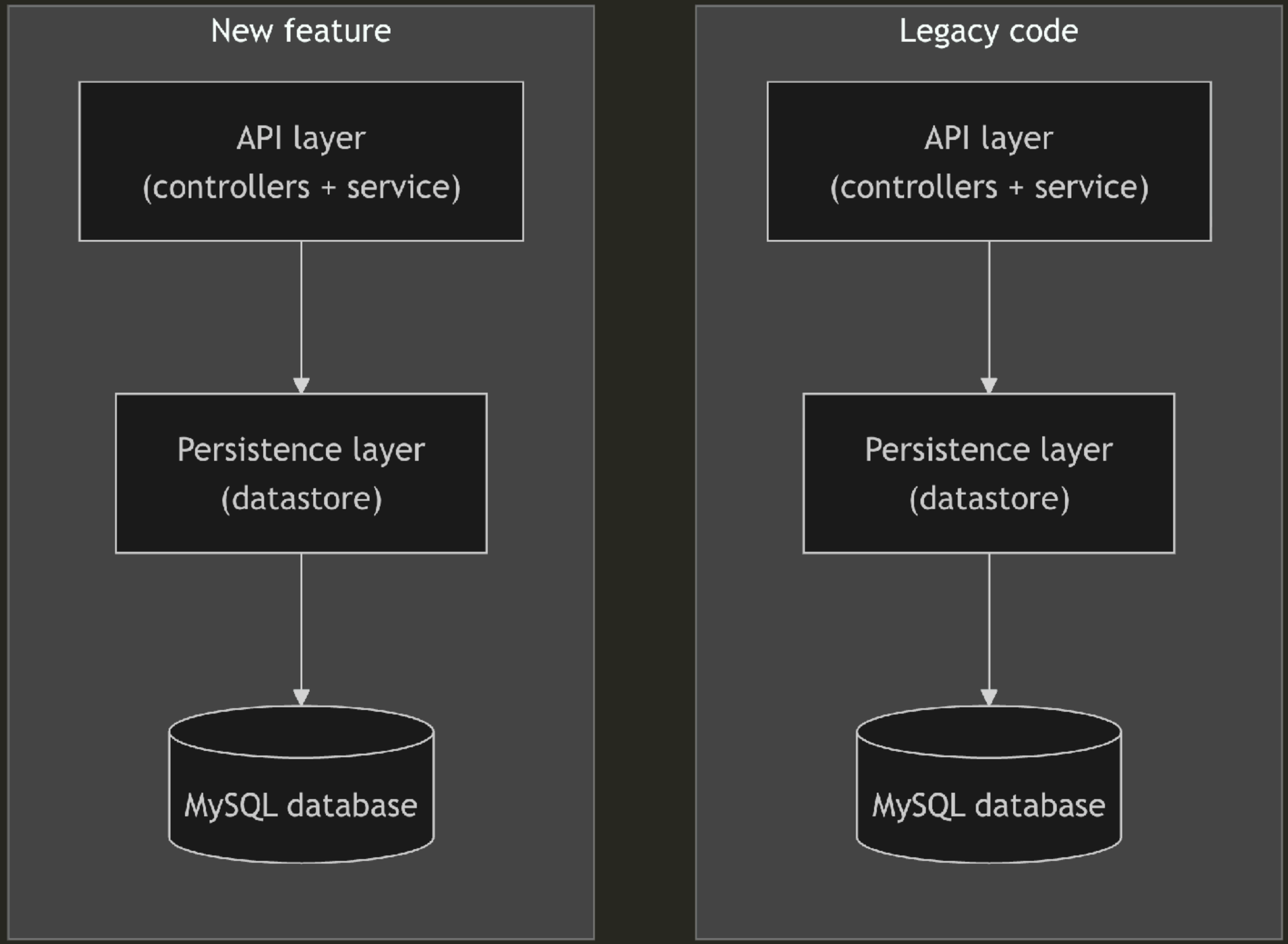


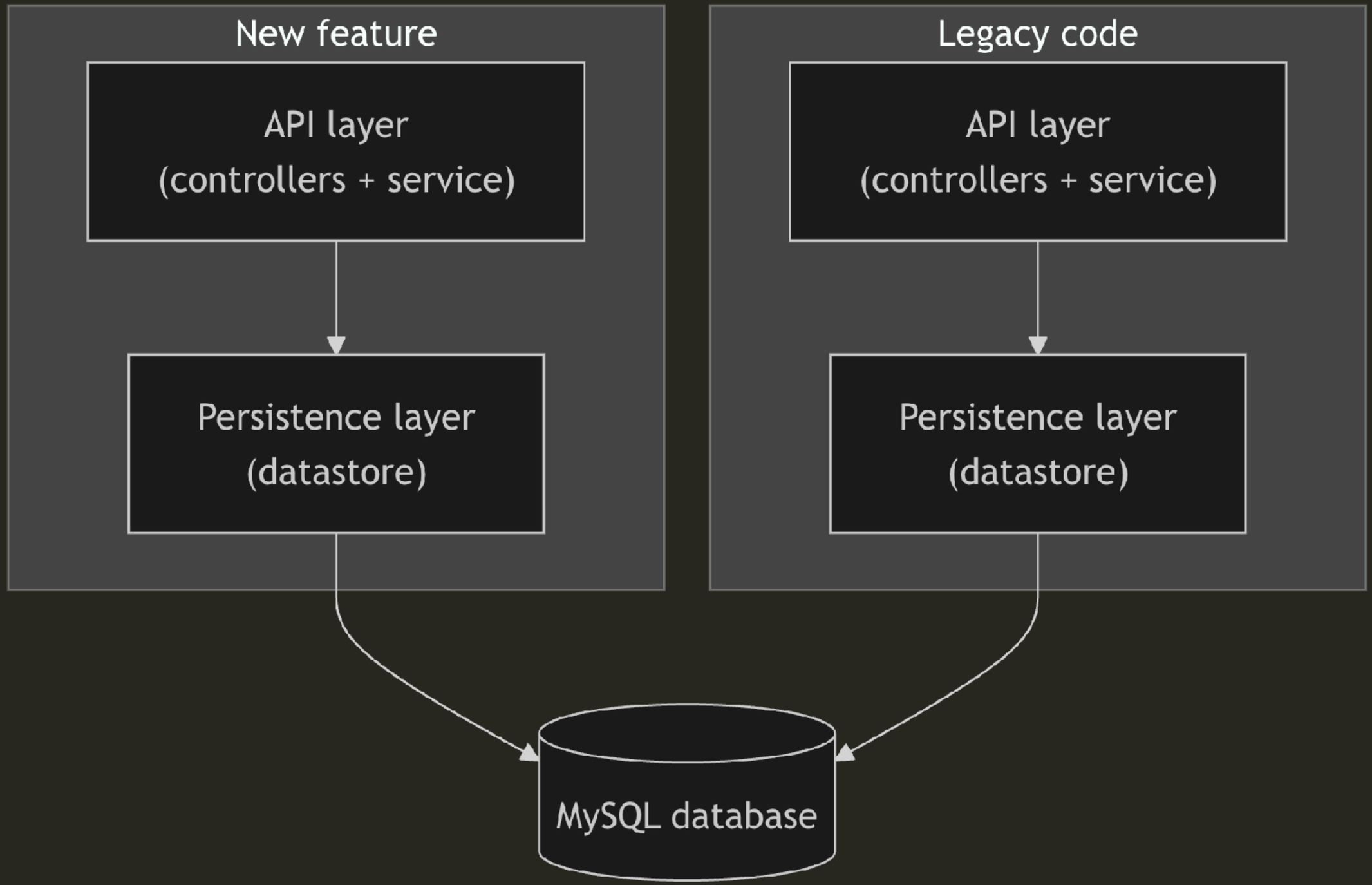

CORE
ANDROID
MDM

I didn’t write much new code.
I mostly relocated code that had been emotionally attached to the old package.

The code was ready.
The humans were not.

fleet

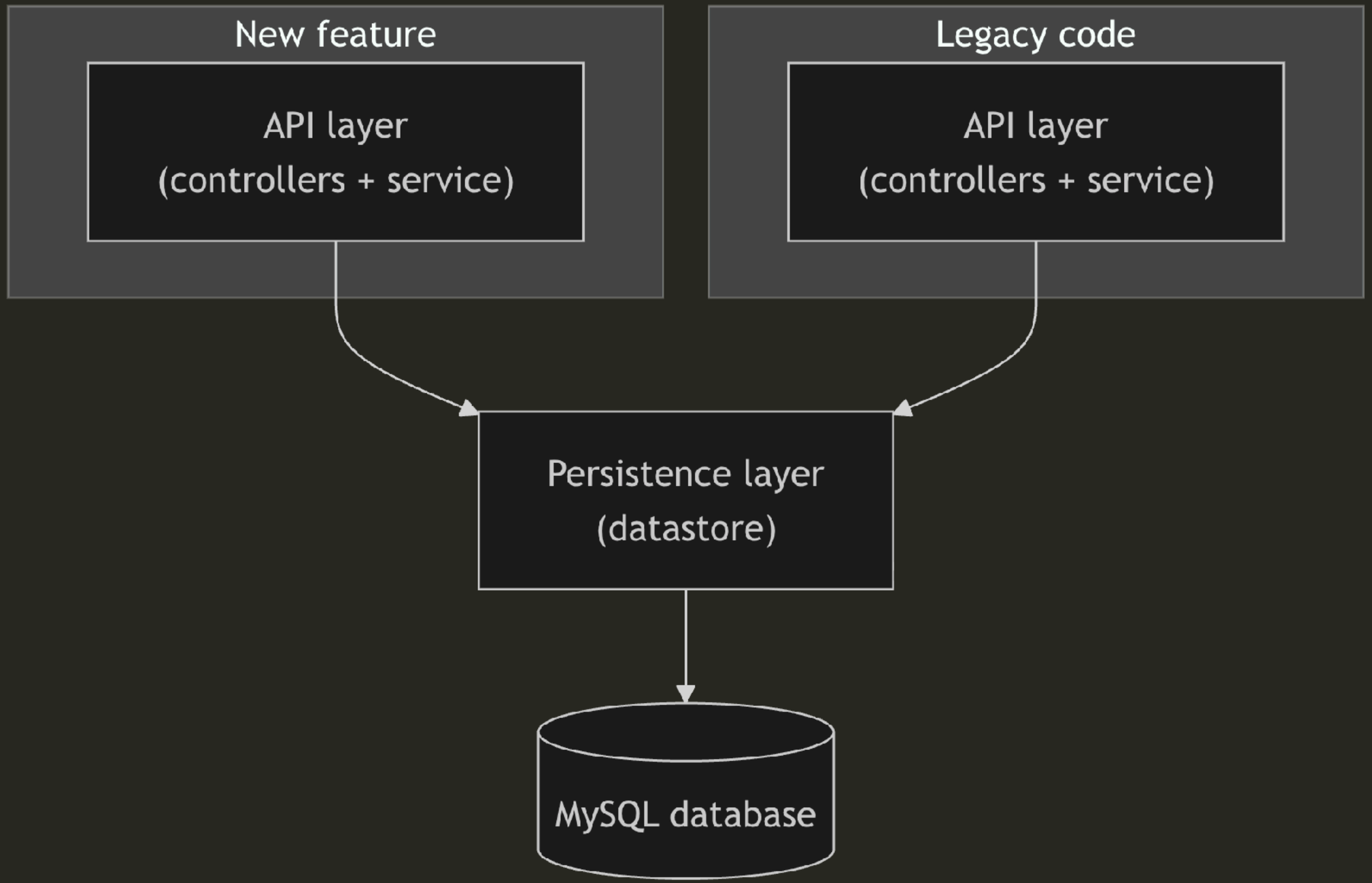

New feature
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database
Legacy code
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database


New feature
API layer
(controllers + service)
Persistence layer
(datastore)
Legacy code
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database

We agreed on the what.
We disagreed on the how deep.

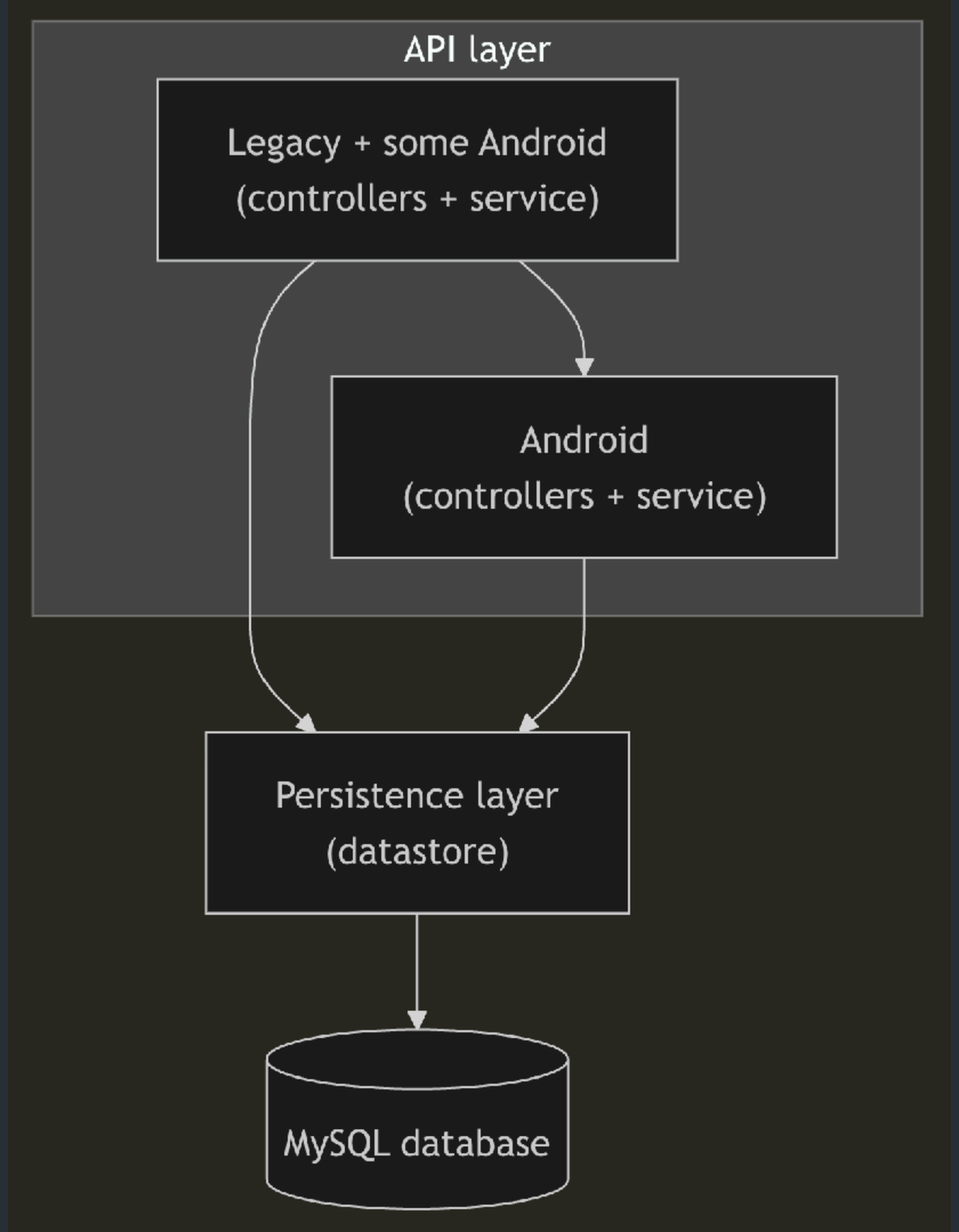
fleet

- We already know how the current architecture works
- How will we do DB transactions?
- This change is too big


New feature
API layer
(controllers + service)
Legacy code
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database

Architecture is like a diet:
it works great
until you stop paying attention

fleet


API layer
Legacy + some Android
(controllers + service)
Android
(controllers + service)
Persistence layer
(datastore)
MySQL database

The problem (so we thought)

fleet

Our biggest architectural dependency wasn't a package.
It was shared understanding.

fleet

Architecture without buy-in is just a suggestion.

fleet

Architecture is code + conversations

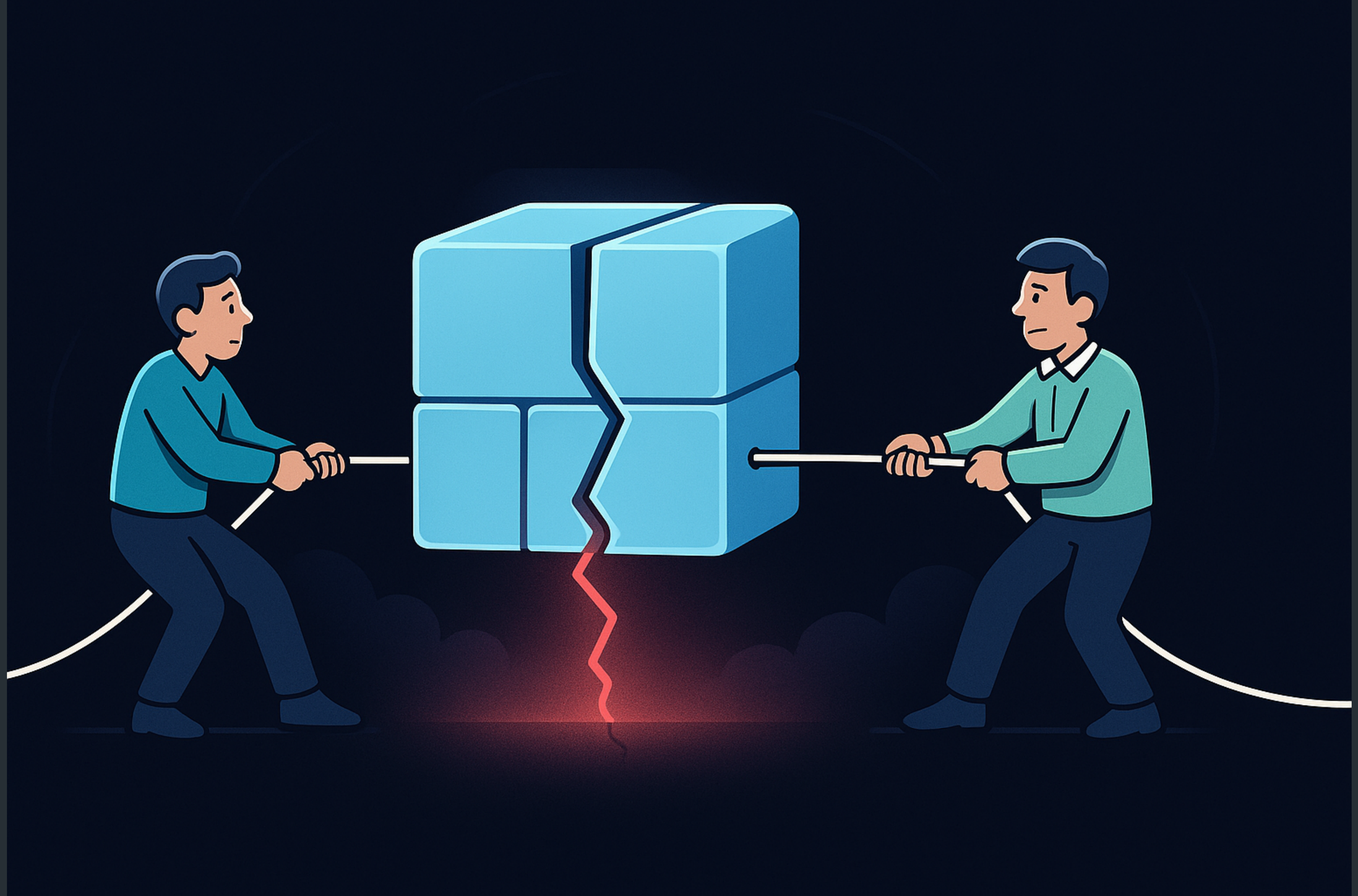
fleet

What is an architectural change, anyway?


FEATURE
ARCHITECTURE

PR reviews don't just check code. They check relationships.

fleet



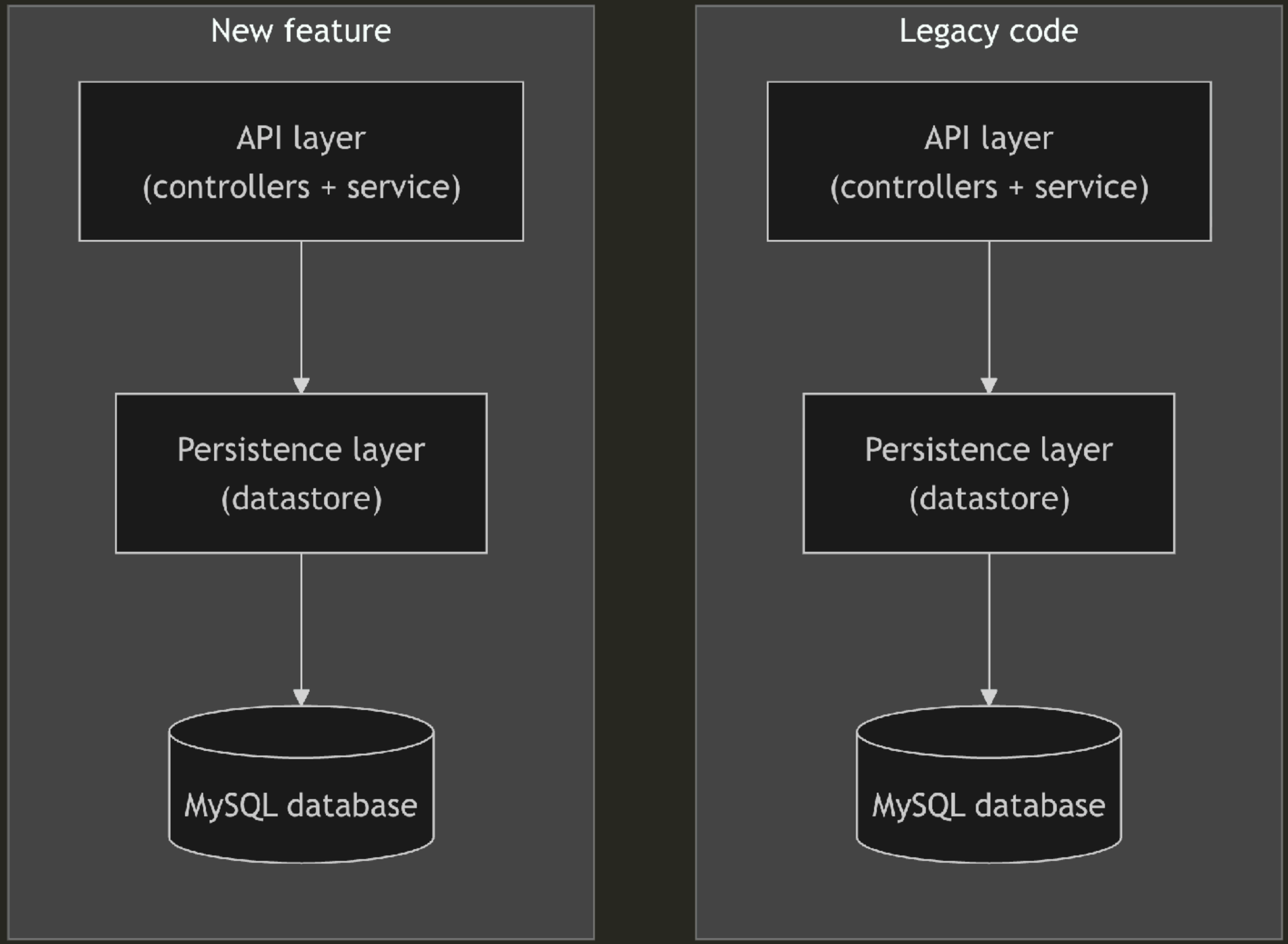
Architecture never stops changing.
It just changes where you're not looking.

fleet

Trust, but verify.

Ronald Reagan

fleet


New feature
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database
Legacy code
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database

Experience is the name everyone gives to their mistakes.

Oscar Wilde

fleet

I didn’t need better ideas.
I needed backup.

fleet

A bounded context is a polite way of saying: "please don’t touch my stuff"

fleet

- **Proposal 1: Platform-Centric (Vertical Slicing)**
- **Proposal 2: Capability-Centric (Domain-Driven Design)**
  - Agent Management (Foundation)
  - Device Enrollment & Lifecycle
  - Configuration Management
  - Software Lifecycle Management
  - Security & Compliance
  - Query & Reporting
  - Automation & Scripts
  - Identity & Access
  - Activity & Audit
  - Platform Core (Shared)
- **Proposal 3: Hybrid (pragmatic evolution)**

ADR PR: https://github.com/fleetdm/fleet/pull/35402

fleet

The hardest part of architecture is waiting for permission to write code.

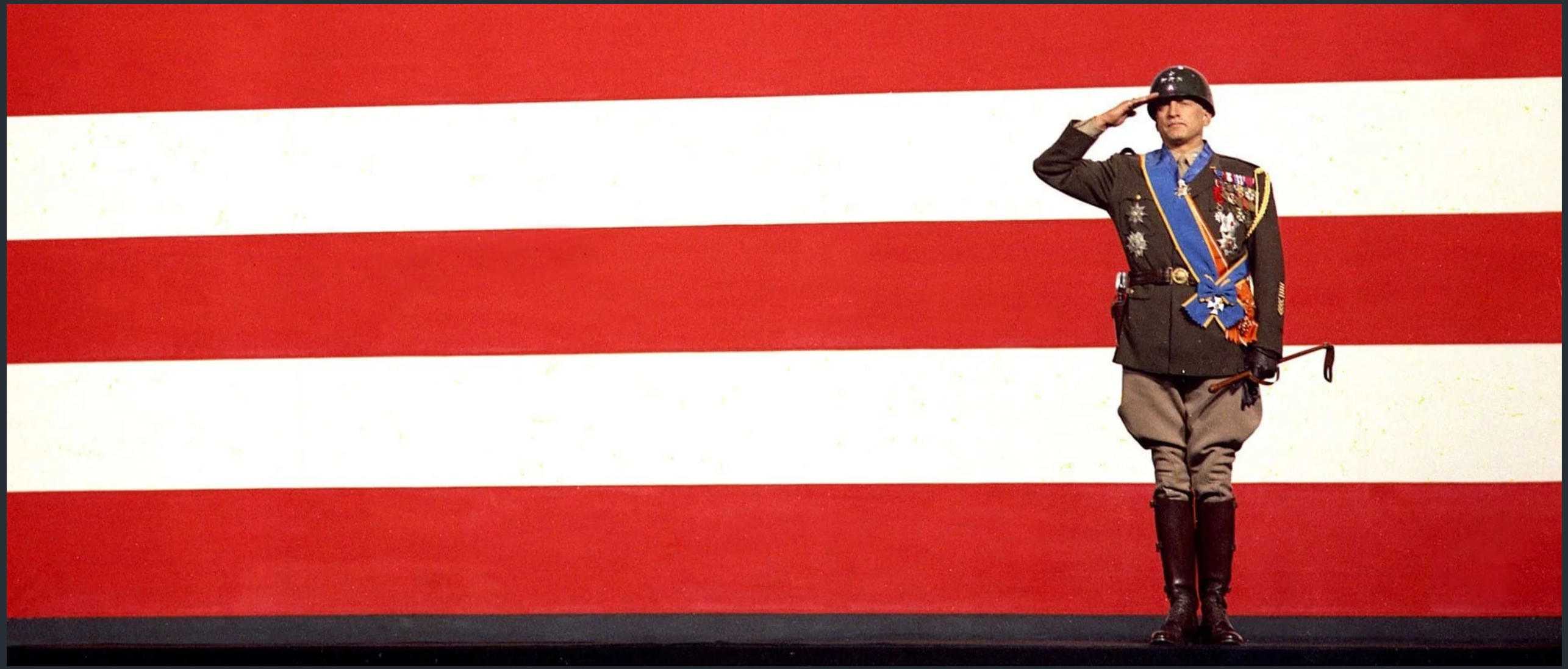
fleet

It felt like pushing a boulder uphill... and watching it roll back down overnight

fleet

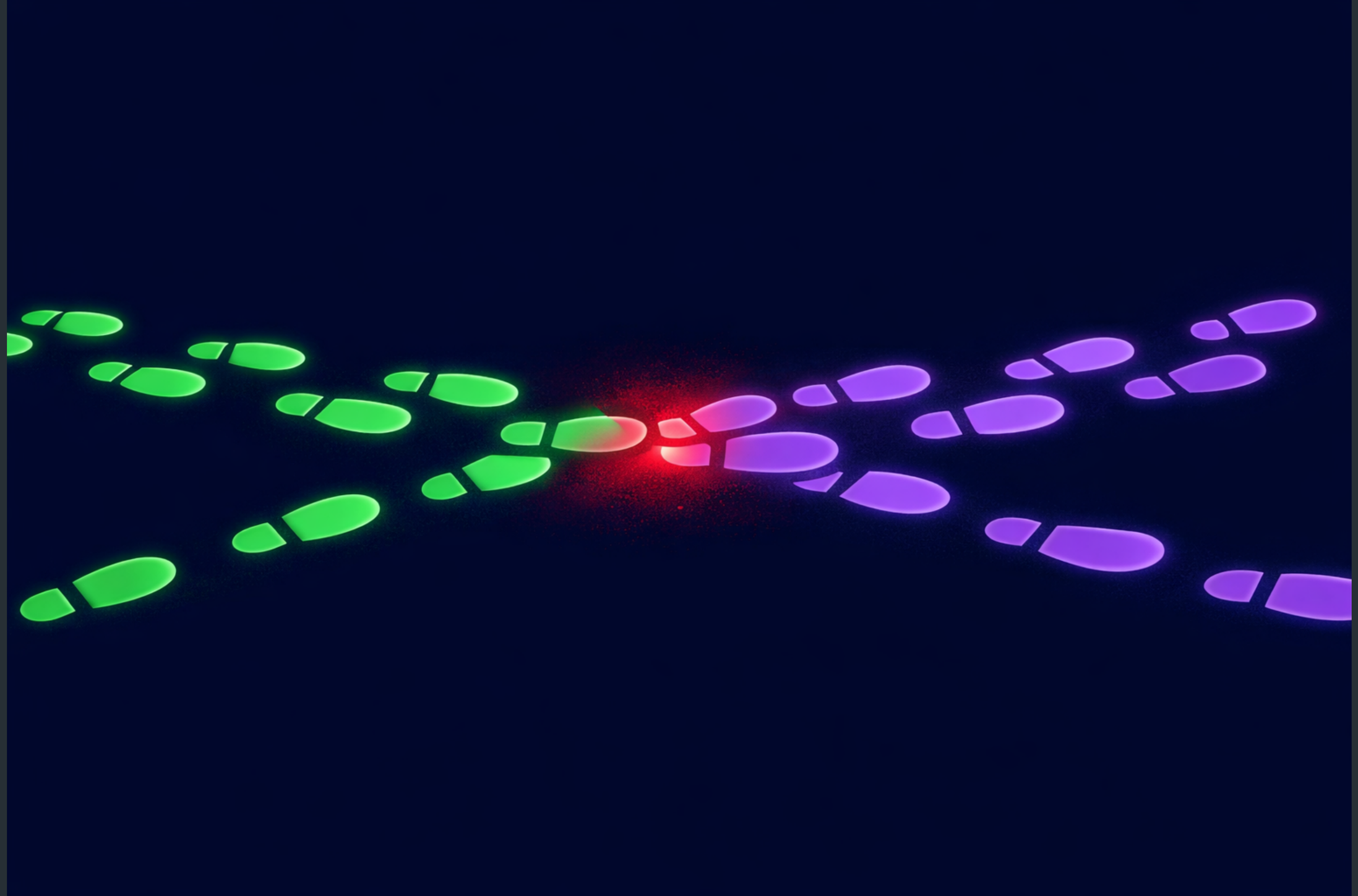
Understanding doesn’t spread by default

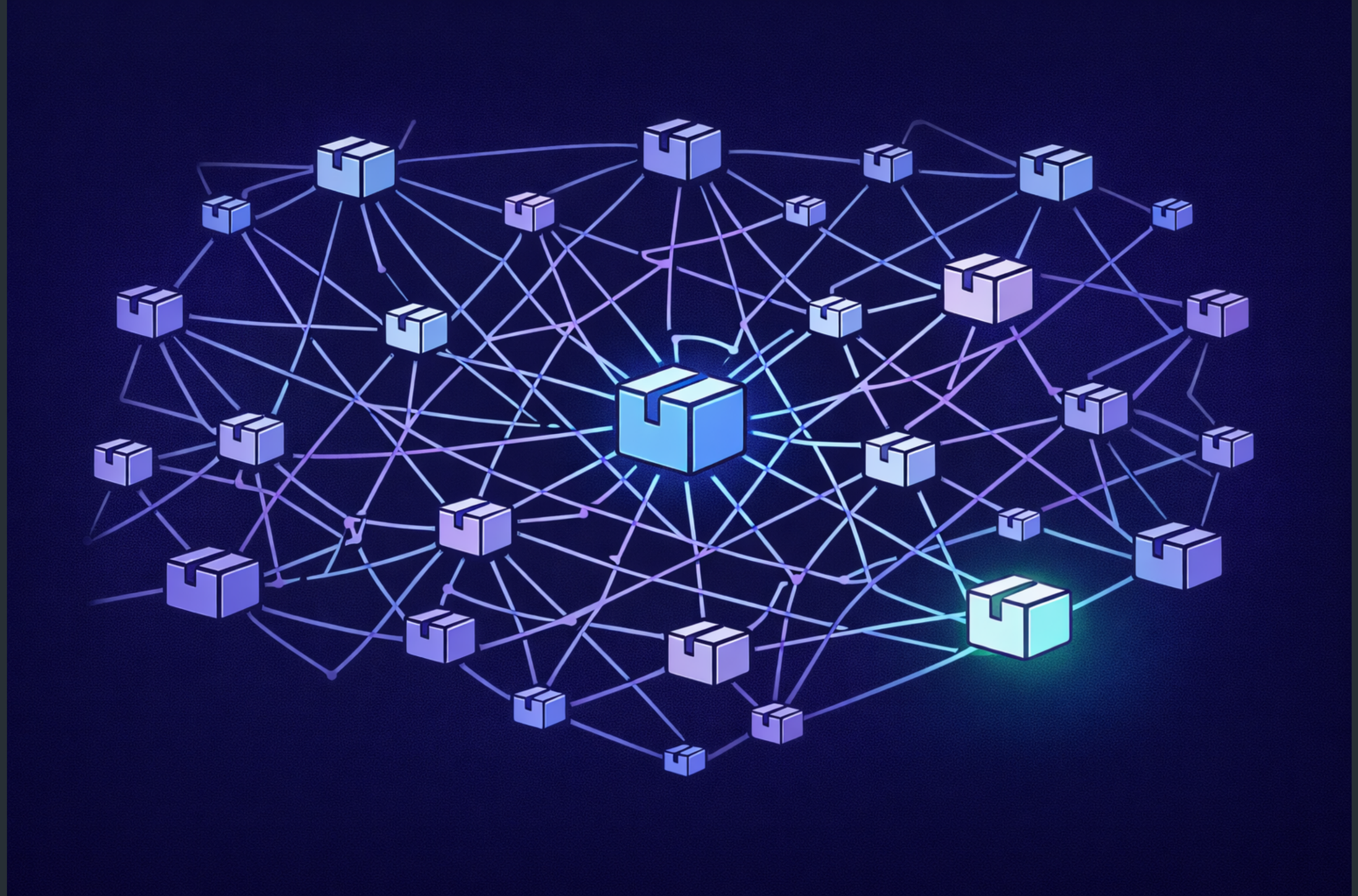
fleet

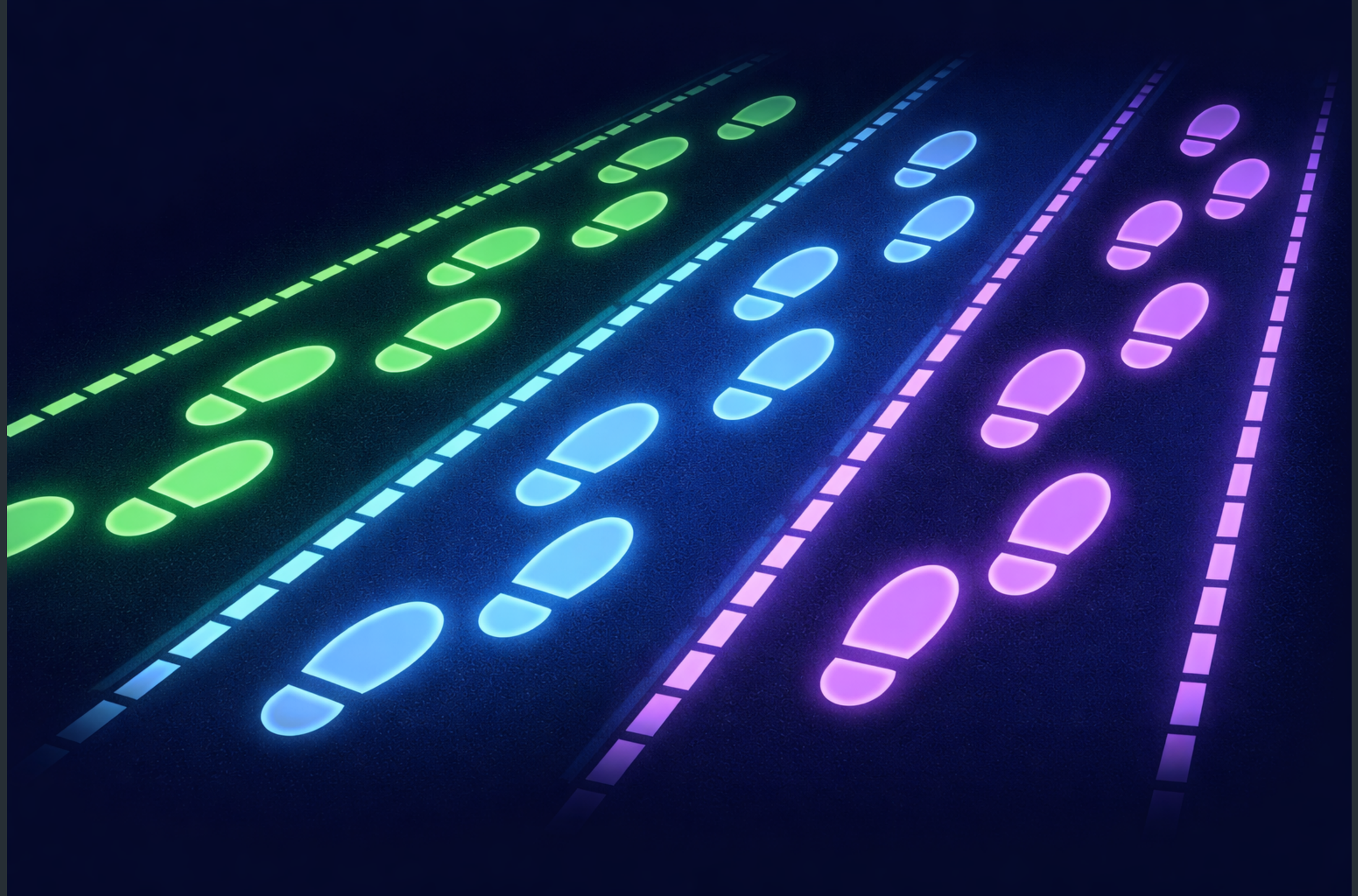


Why Modularization Matters

fleet

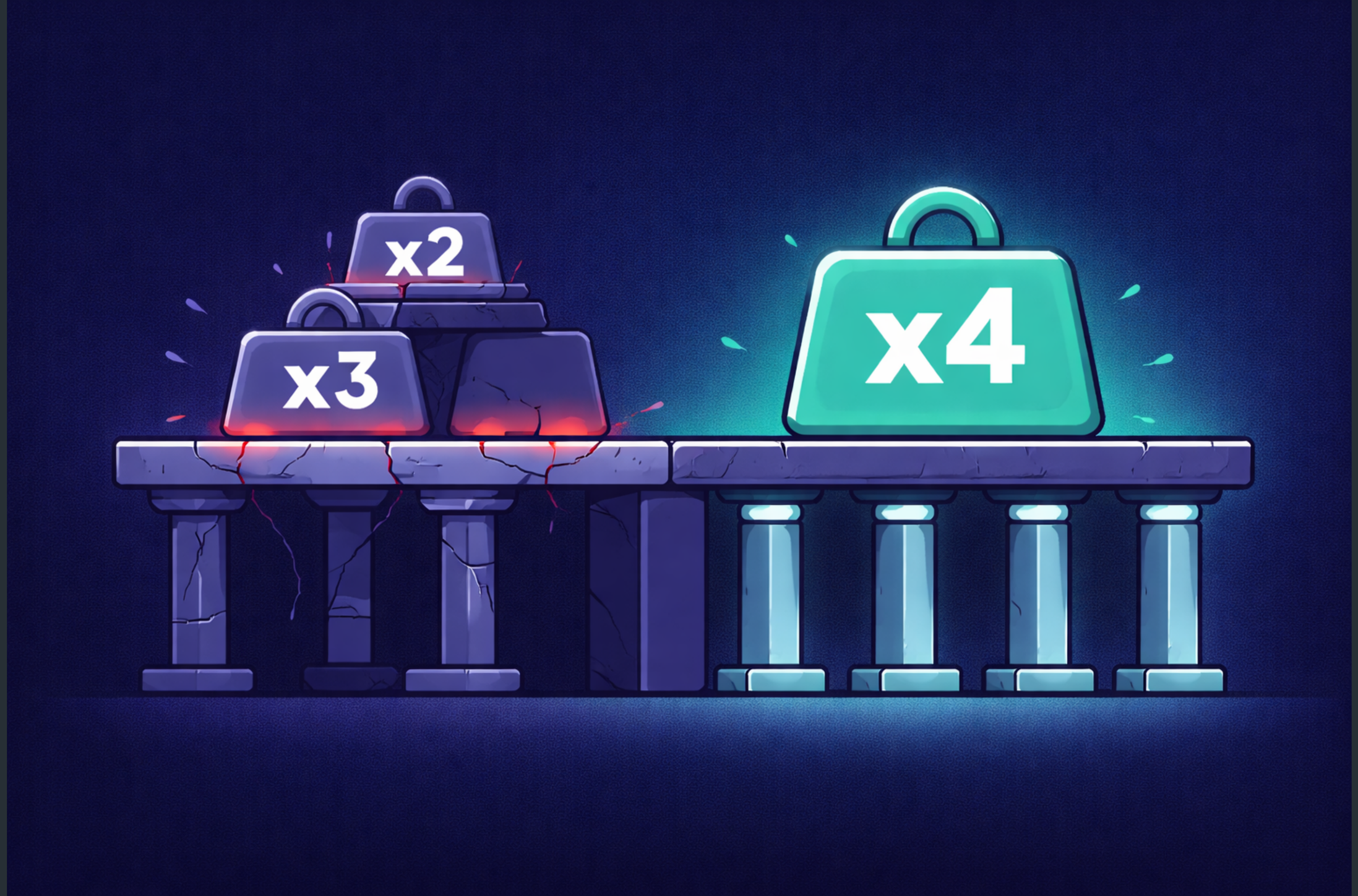


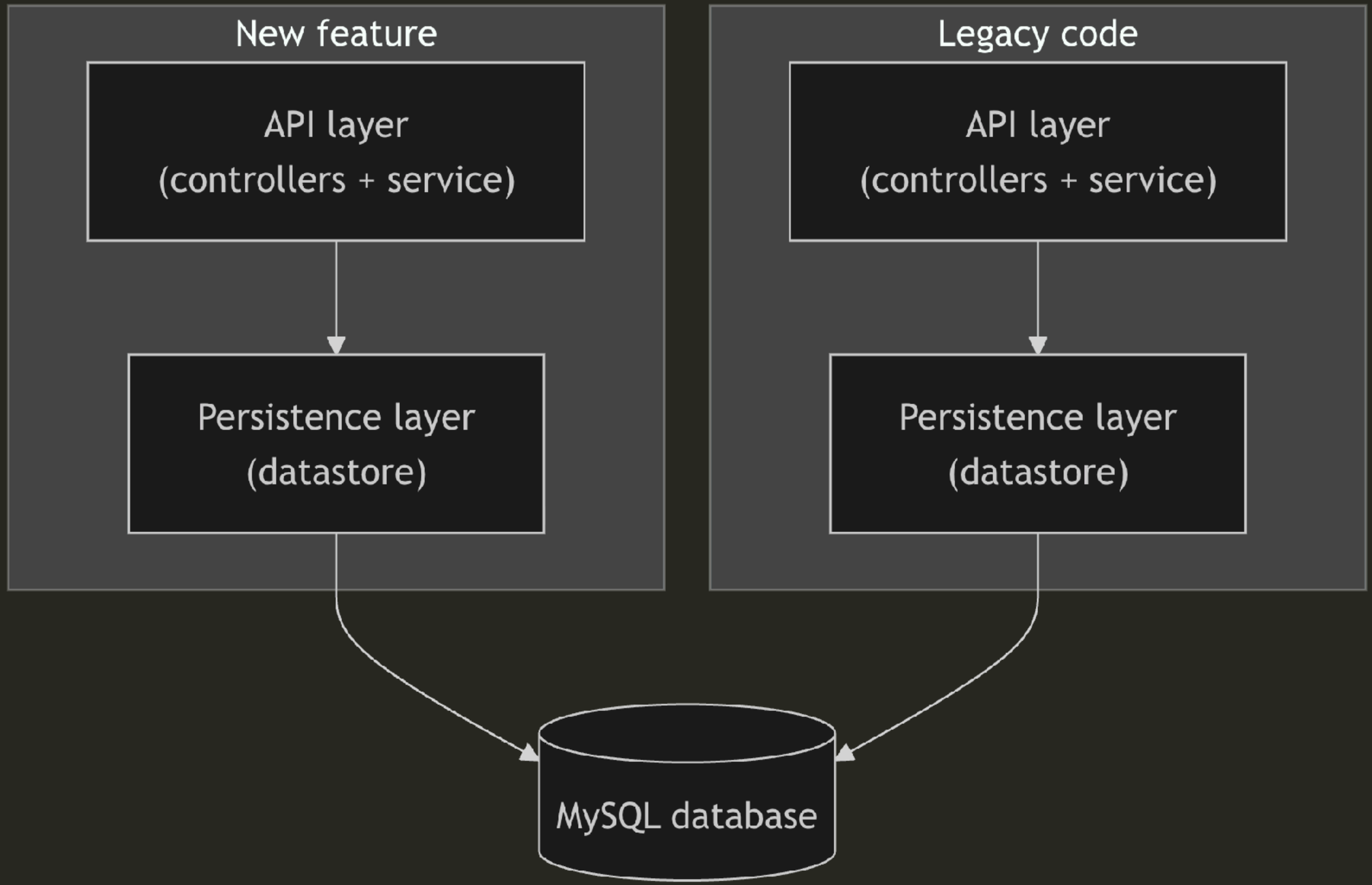




Modularity is what lets teams move fast without asking permission

fleet


x2
x3
x4

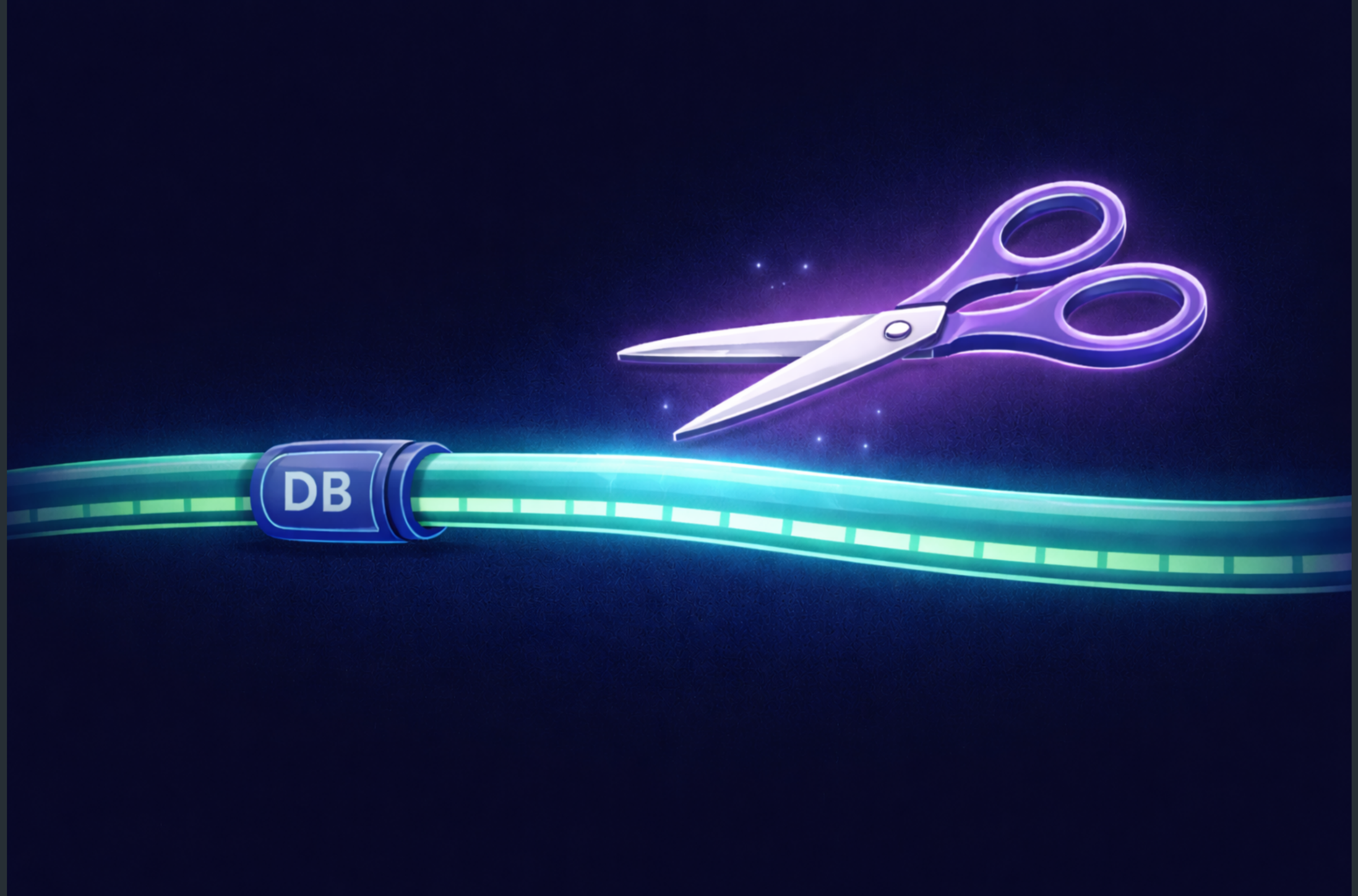

New feature
API layer
(controllers + service)
Persistence layer
(datastore)
Legacy code
API layer
(controllers + service)
Persistence layer
(datastore)
MySQL database

```
server/
└── activity/
    ├── activity.go                # Public interface and types
    ├── service/
    │   ├── handler.go             # HTTP handlers (controllers)
    │   ├── service.go             # Business logic implementation
    │   └── service_test.go        # Unit tests
    ├── mysql/
    │   ├── activity.go            # MySQL datastore implementation
    │   └── activity_test.go       # Unit tests
    └── tests/
        └── integration_test.go # Service + MySQL integration tests
```


DATABASE

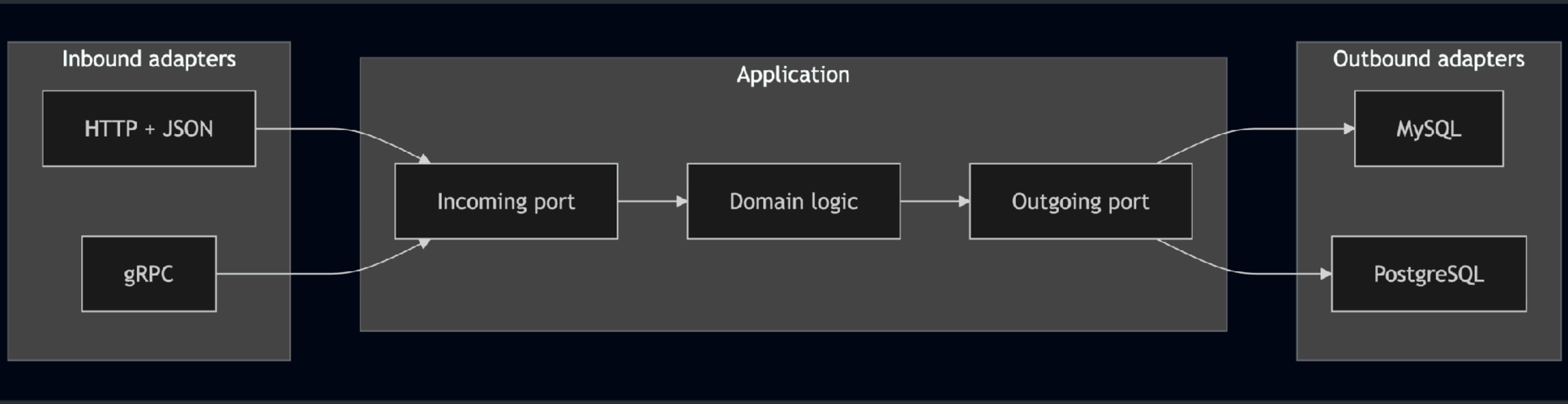
Databases scale data.
Modules scale teams.

fleet


DB

When enough smart people repeat the same advice, you assume you're the one missing something

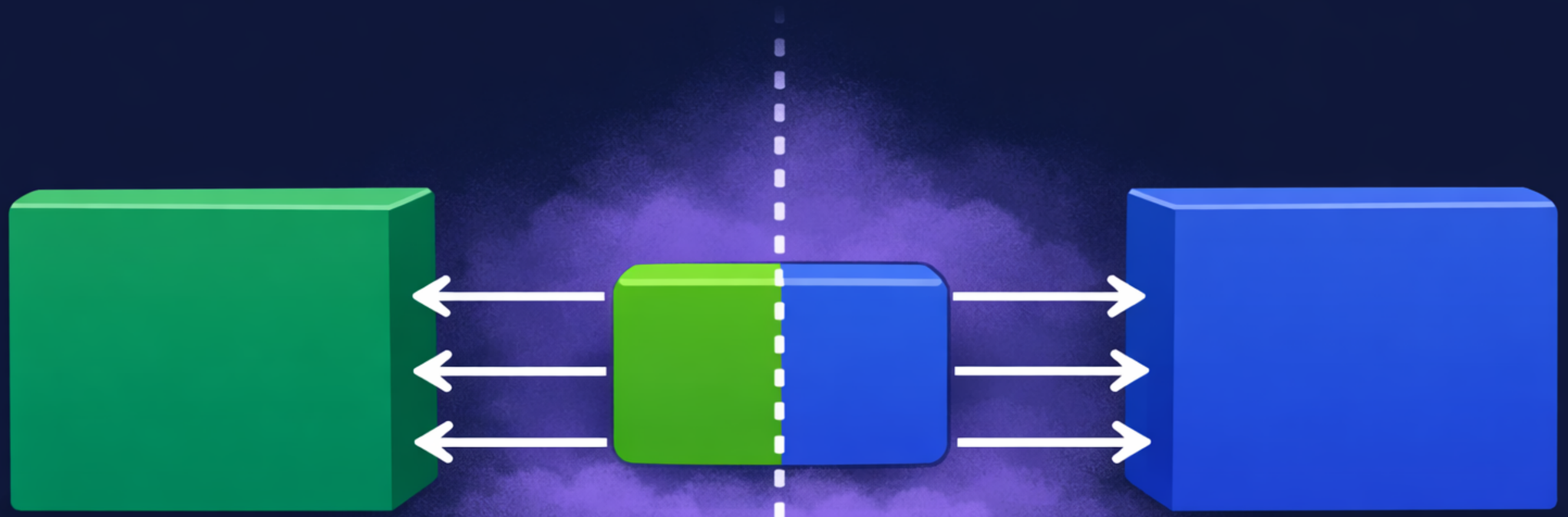
fleet

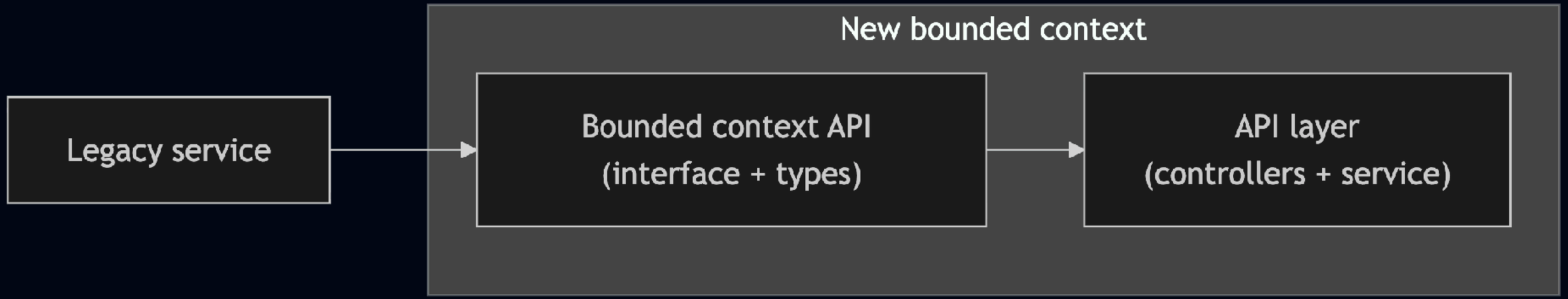

Inbound adapters
HTTP + JSON
gRPC
Application
Incoming port
Domain logic
Outgoing port
Outbound adapters
MySQL
PostgreSQL

```
exclude_any AS (
    SELECT
        software_installers.id AS installer_id,
        COUNT(software_installer_labels.label_id) AS count_installer_labels,
        COUNT(label_membership.label_id) AS count_host_labels,
        SUM(
            CASE
                WHEN labels.created_at IS NOT NULL AND (
                    labels.label_membership_type = 1 OR
                    (labels.label_membership_type = 0 AND :host_label_updated_at >= labels.created_at)
                ) THEN 1
                ELSE 0
            END
        ) AS count_host_updated_after_labels
    FROM
        software_installers
    INNER JOIN software_installer_labels
        ON software_installer_labels.software_installer_id = software_installers.id AND software_installer_labels.exclude = 1
    INNER JOIN labels
        ON labels.id = software_installer_labels.label_id
    LEFT JOIN label_membership
        ON label_membership.label_id = software_installer_labels.label_id
        AND label_membership.host_id = :host_id
    GROUP BY
        software_installers.id
    HAVING
        COUNT(*) > 0
        AND COUNT(*) = SUM(
            CASE
                WHEN labels.created_at IS NOT NULL AND (
                    labels.label_membership_type = 1 OR
                    (labels.label_membership_type = 0 AND :host_label_updated_at >= labels.created_at)
                ) THEN 1
                ELSE 0
            END
        )
        AND COUNT(label_membership.label_id) = 0
)
```


9:41
Ports and Adapters
Block This Contact?
You will no longer receive messages, calls, notifications from this person.
Block Contact
Cancel

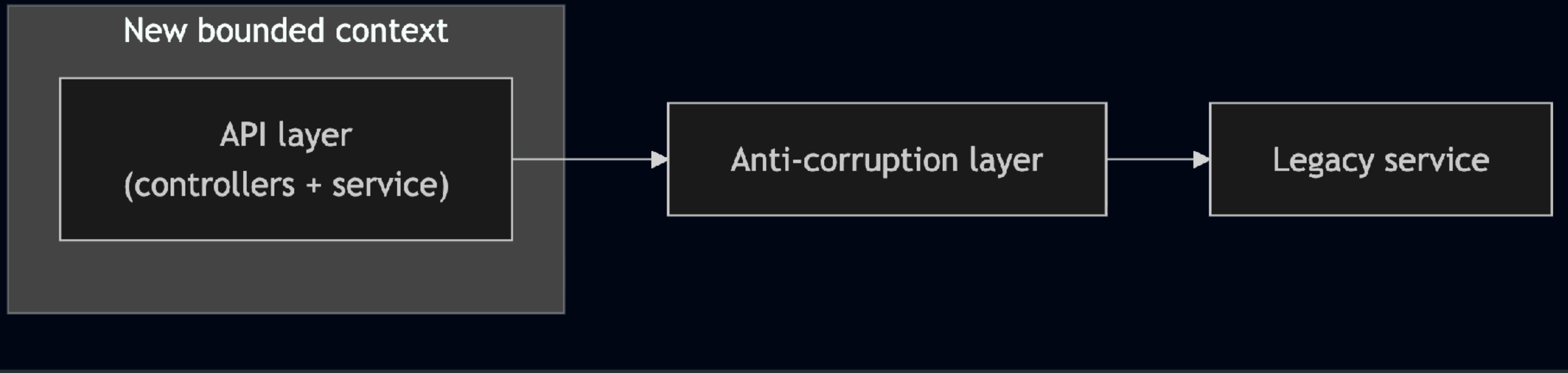
Explicit module boundaries




New bounded context
Legacy service
Bounded context API
(interface + types)
API layer
(controllers + service)

Reuse inside a boundary is efficiency.
Reuse across boundaries is coupling.

fleet


New bounded context
API layer
(controllers + service)
Anti-corruption layer
Legacy service

No matter how good the design is, "someone else's idea" never feels like the right one

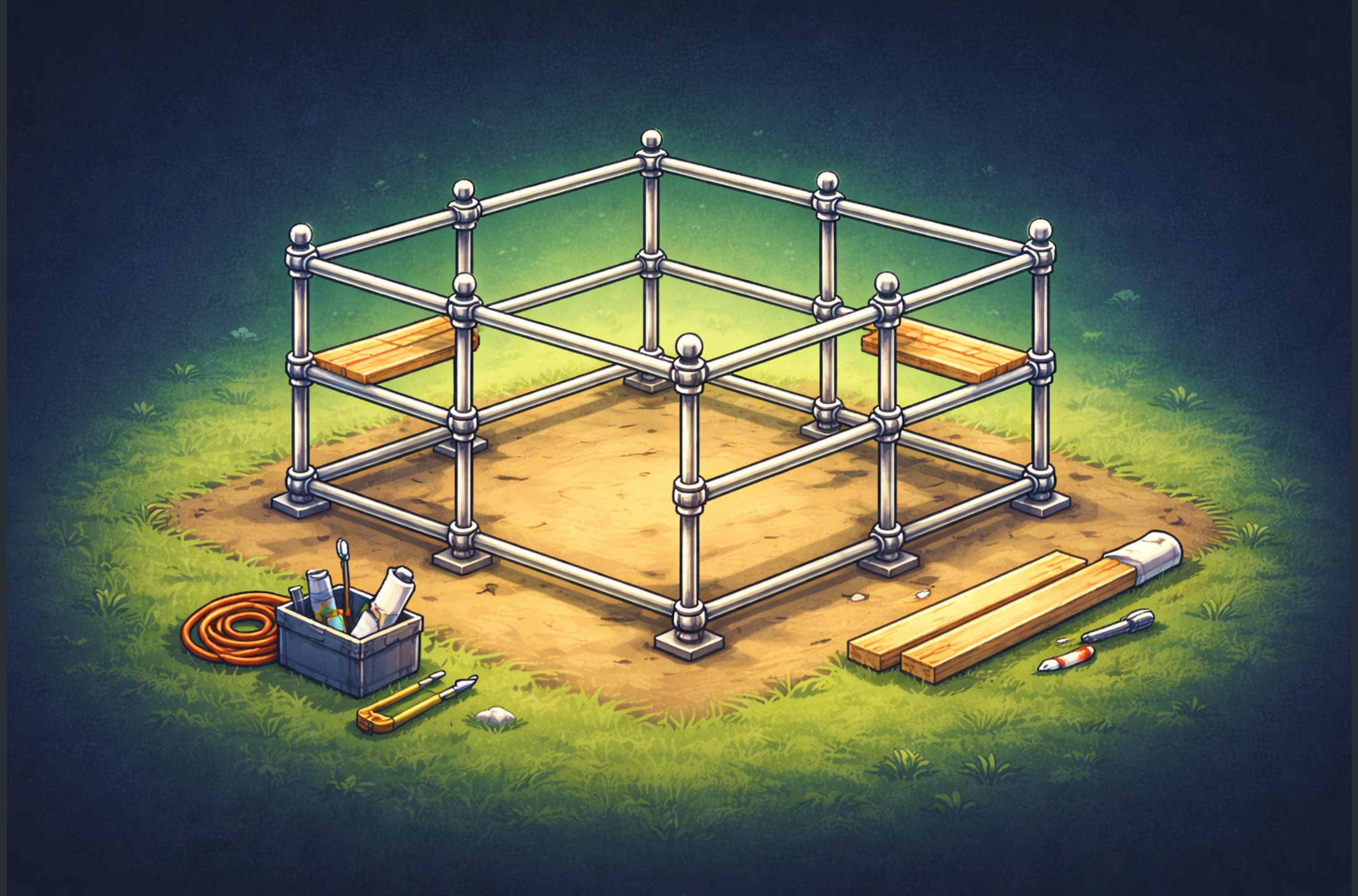
fleet

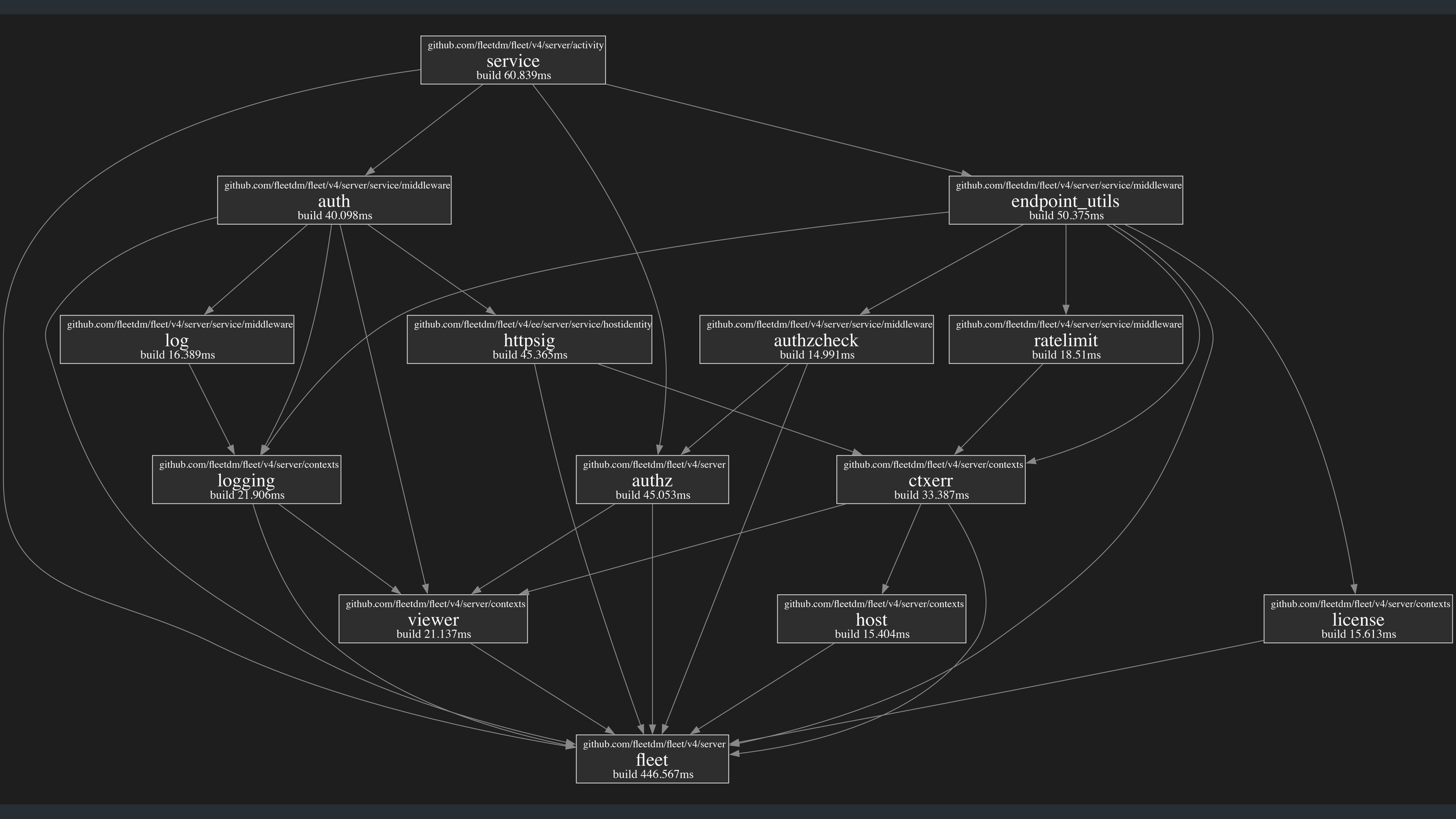
This was architecture by small, boring steps. On purpose.

fleet

You don’t discover coupling on a whiteboard

fleet




github.com/fleetdm/fleet/v4/server/activity
service
build 60.839ms
github.com/fleetdm/fleet/v4/server/service/middleware
auth
build 40.098ms
github.com/fleetdm/fleet/v4/server/service/middleware
endpoint_utils
build 50.375ms
github.com/fleetdm/fleet/v4/server/service/middleware
log
build 16.389ms
github.com/fleetdm/fleet/v4/ee/server/service/hostidentity
httpsig
build 45.365ms
github.com/fleetdm/fleet/v4/server/service/middleware
authzcheck
build 14.991ms
github.com/fleetdm/fleet/v4/server/service/middleware
ratelimit
build 18.51ms
github.com/fleetdm/fleet/v4/server/contexts
logging
build 21.906ms
github.com/fleetdm/fleet/v4/server
authz
build 45.053ms
github.com/fleetdm/fleet/v4/server/contexts
ctxerr
build 33.387ms
github.com/fleetdm/fleet/v4/server/contexts
viewer
build 21.137ms
github.com/fleetdm/fleet/v4/server/contexts
host
build 15.404ms
github.com/fleetdm/fleet/v4/server/contexts
license
build 15.613ms
github.com/fleetdm/fleet/v4/server
fleet
build 446.567ms

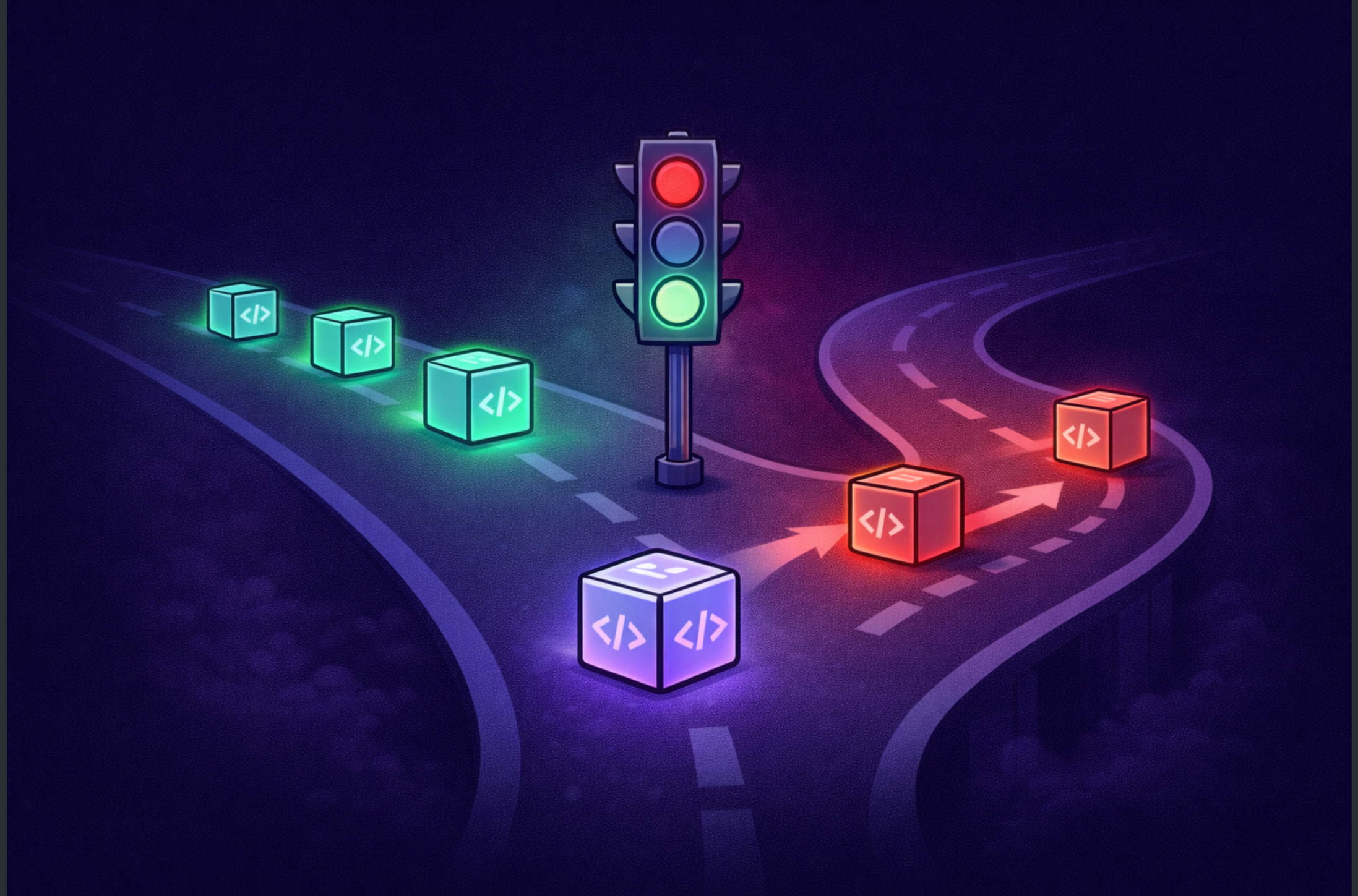
This package solved a lot of problems... by creating new ones

fleet

We didn’t stop reusing code.
We started being intentional about it.

If architecture only exists in people's heads, it exists on borrowed time

fleet

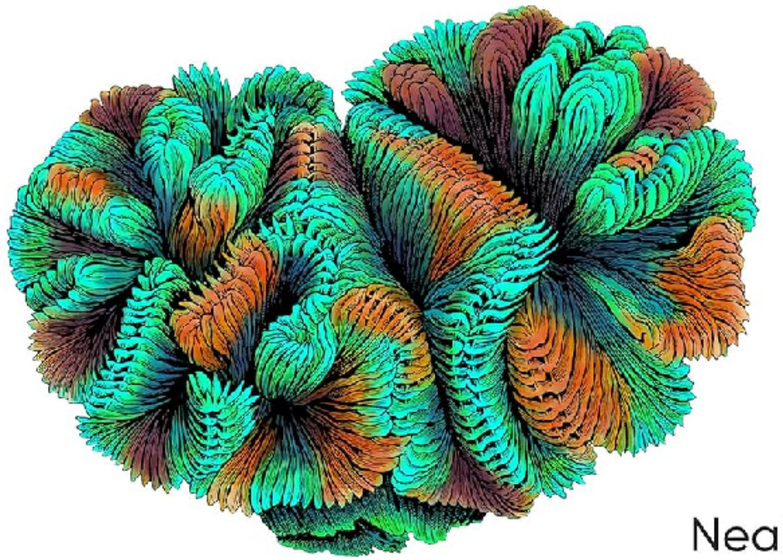


O'REILLY®

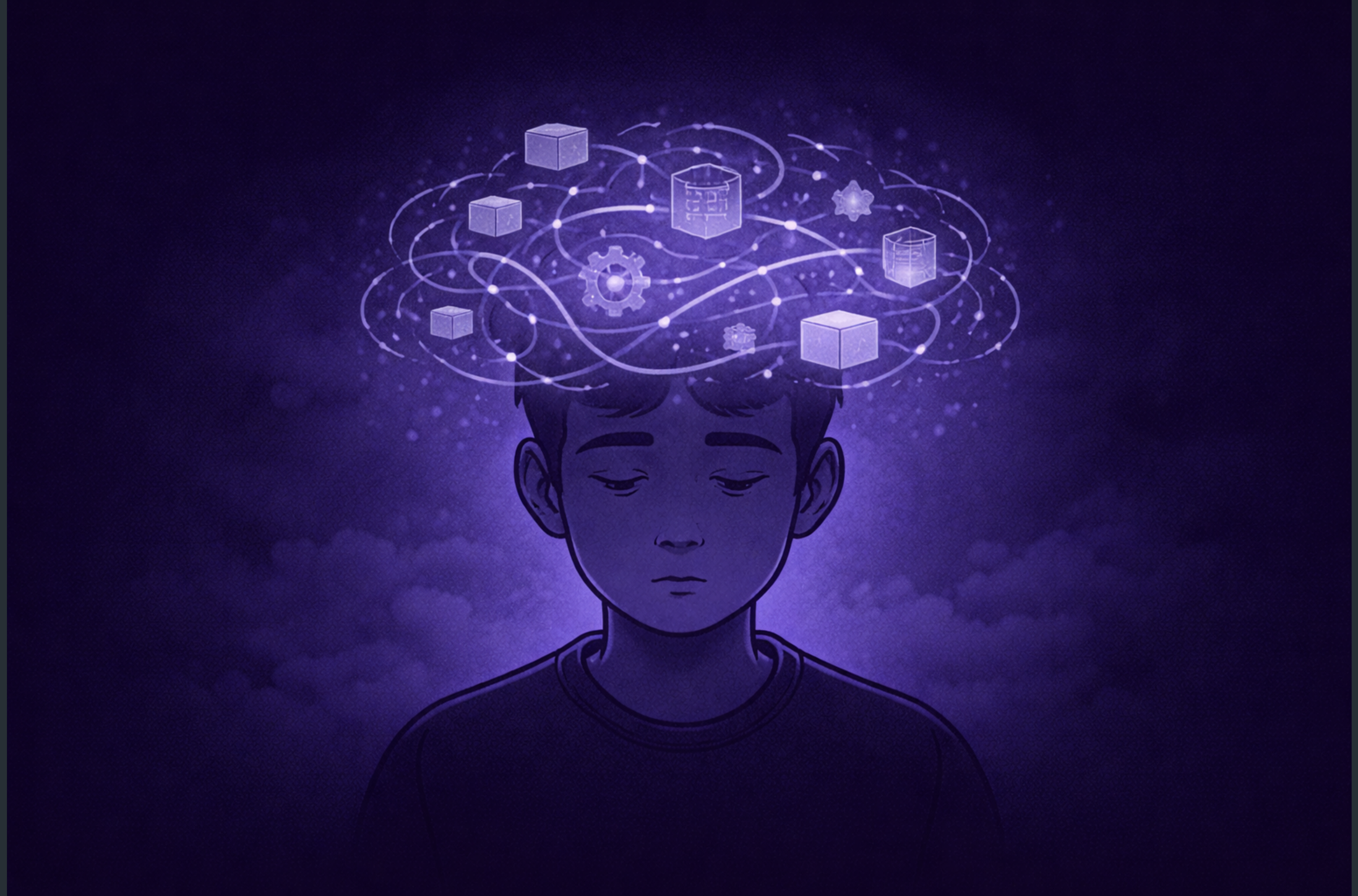
2nd Edition

Building Evolutionary Architectures

Automated Software Governance



Neal Ford,
Rebecca Parsons,
Patrick Kua & Pramod Sadalage
Forewords by Mark Richards & Martin Fowler

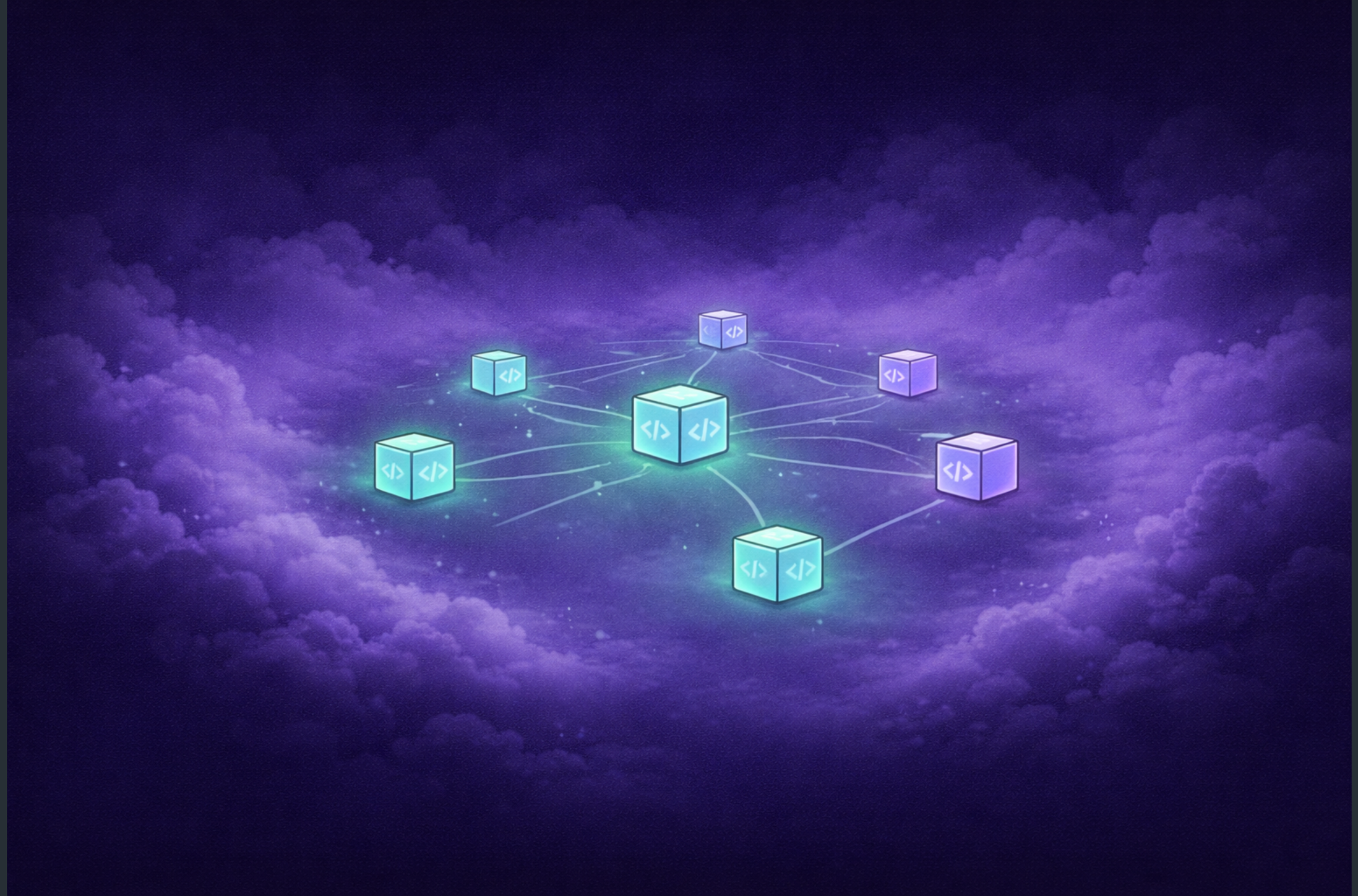


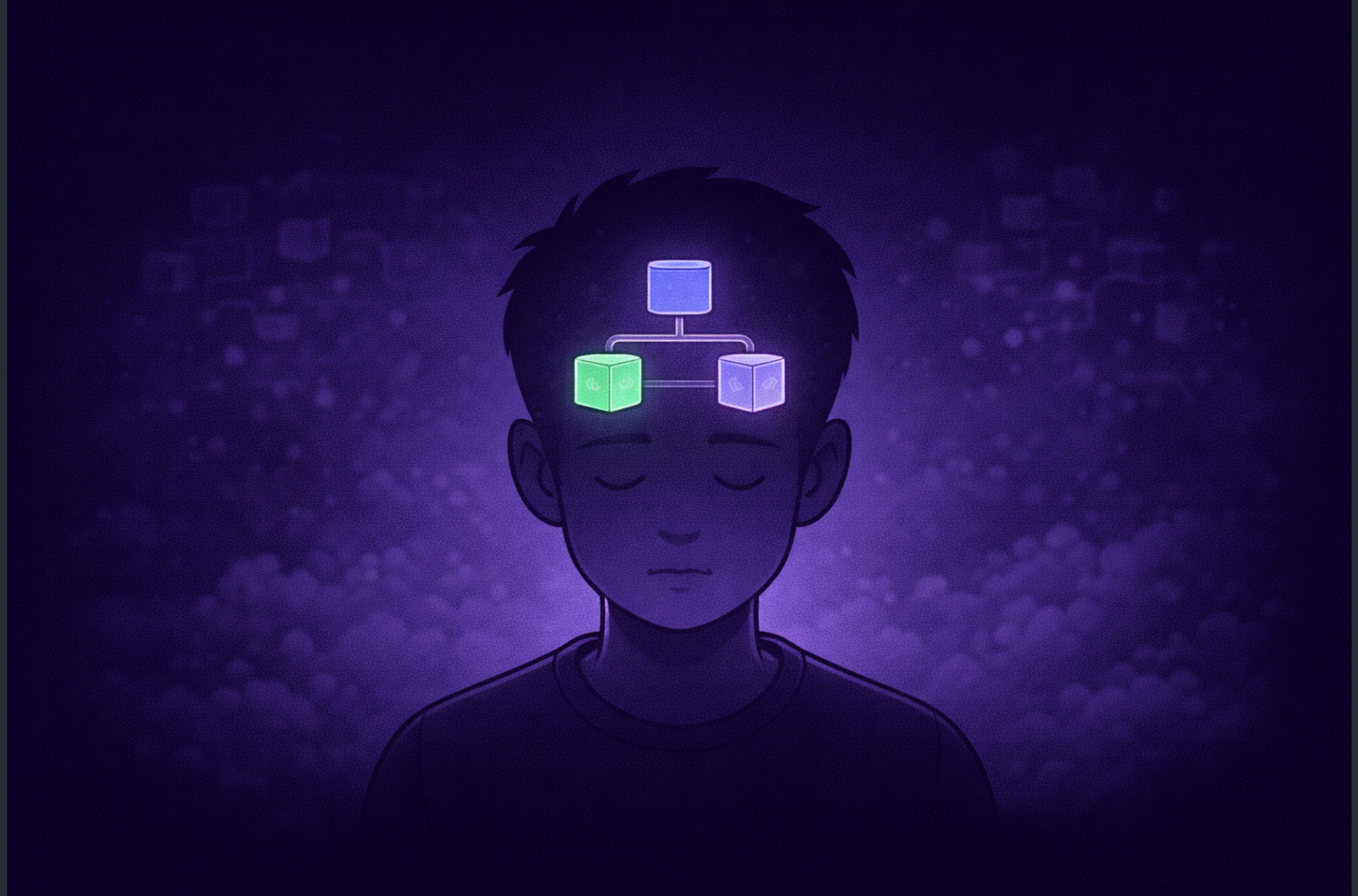
Architecture is an interface for humans

fleet

The system worked.
Changing it became the problem.

fleet





Links

- Fleet Device Management
  - fleetdm.com
  - github.com/fleetdm/fleet
  - We're hiring: fleetdm.com/jobs
- How to find me
  - LinkedIn: linkedin.com/in/lyuboslavsky
  - Blog: victoronsoftware.com
  - YouTube: youtube.com/@VictorOnSoftware