

# Modularizing a 10-Year Monolith

The Architecture, the People, and the Pain

Victor Lyuboslavsky, Conf42 Golang 2026





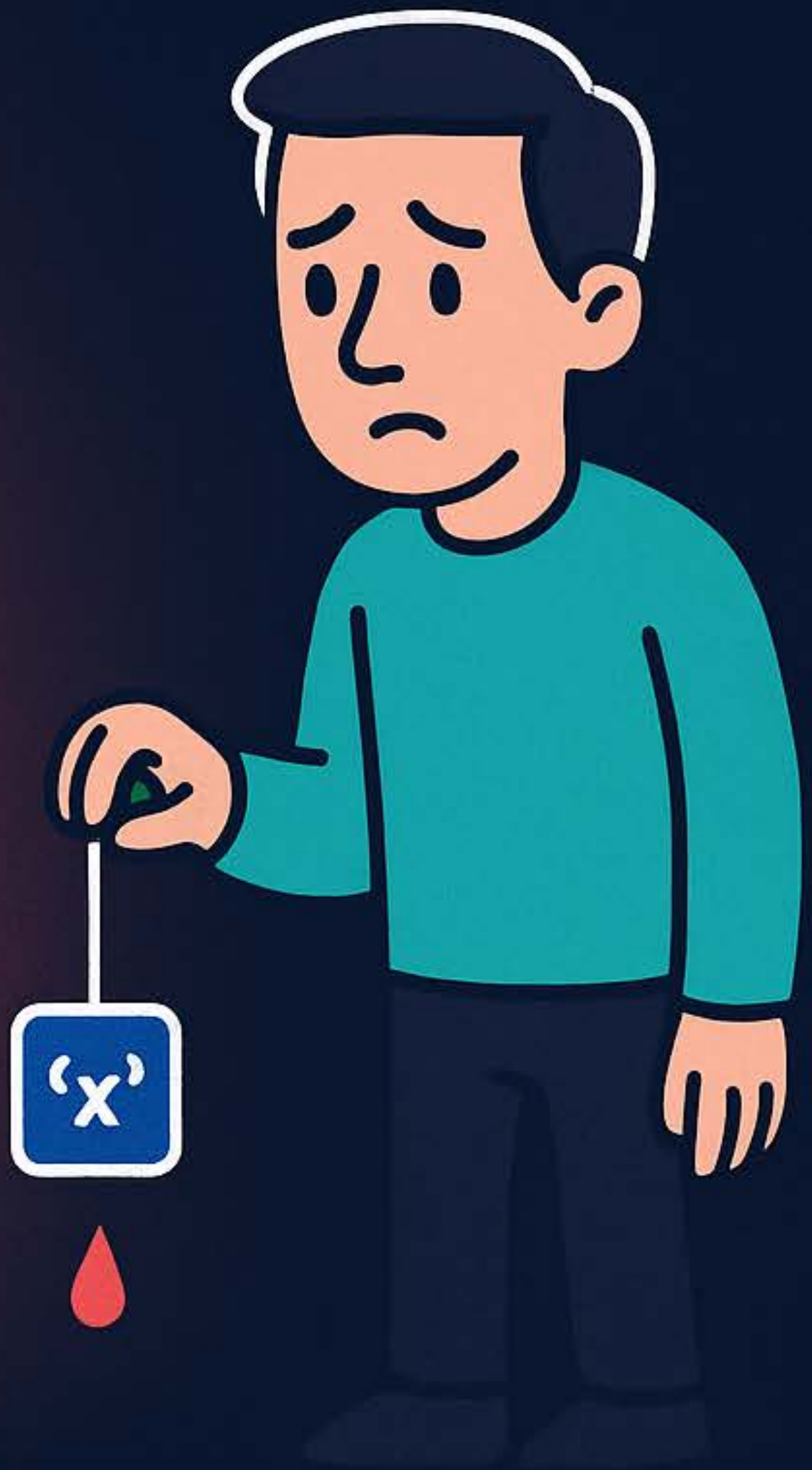
**Bleeding out 30 seconds at a time**





service  
package

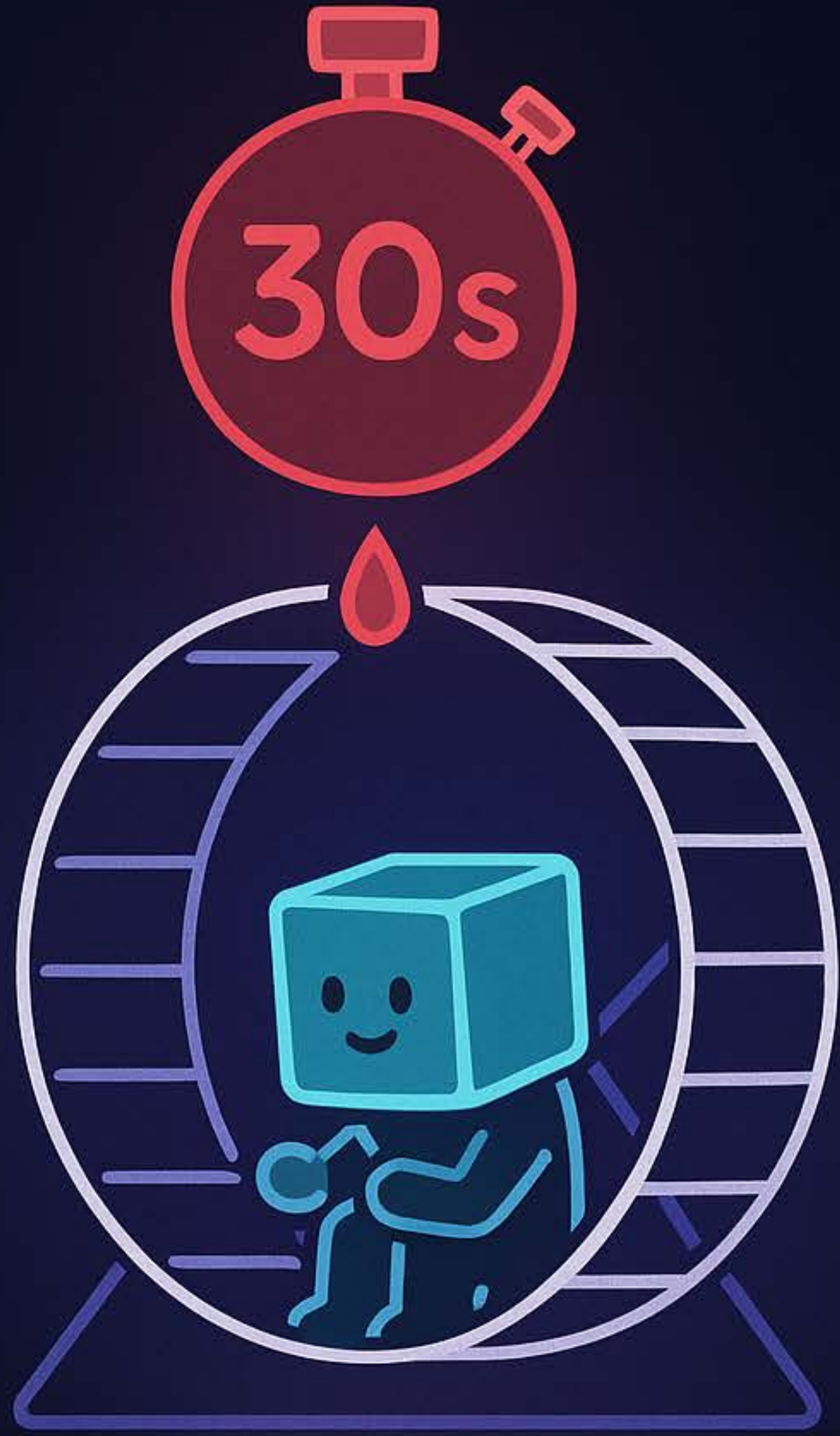
LINKER



'x'







**Small code moves fast.  
Small code feels alive.**



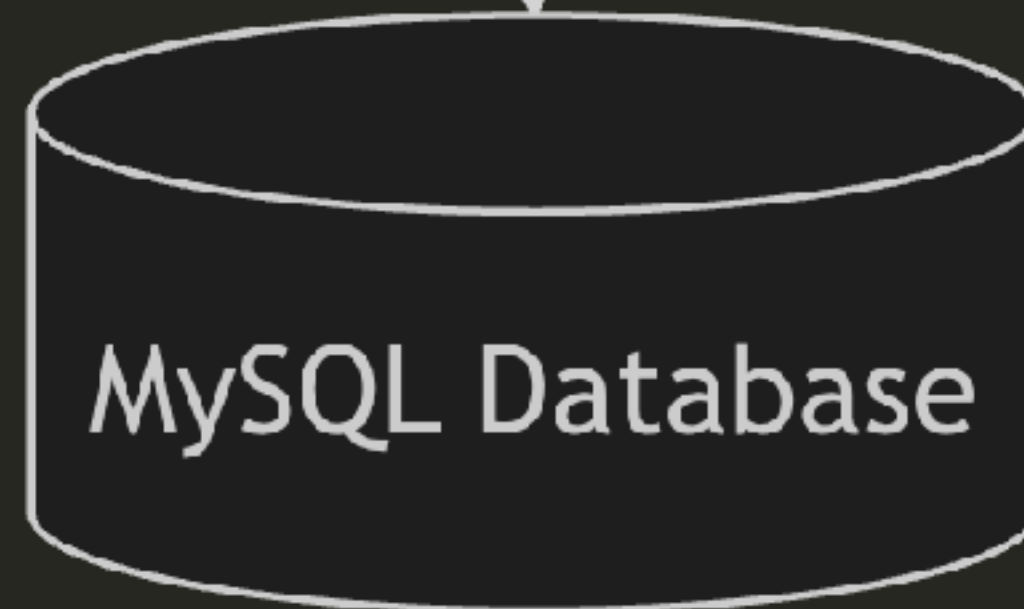
[github.com/fleetdm/fleet](https://github.com/fleetdm/fleet)



API layer  
(controllers + service)



Persistence layer  
(datastore)



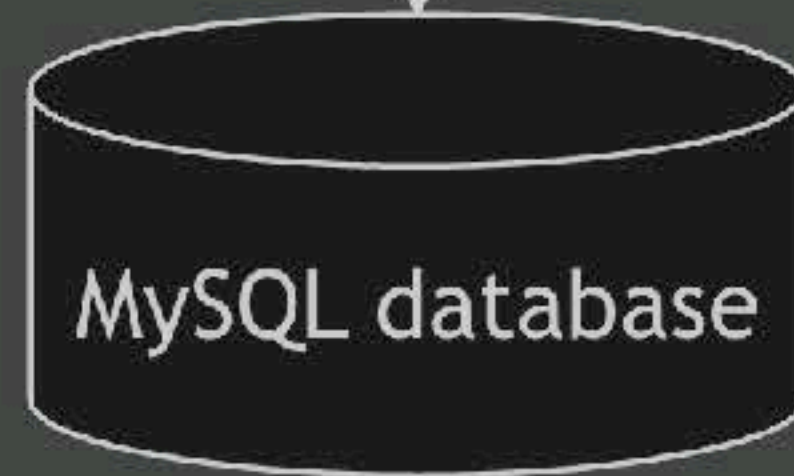


## New feature

API layer  
(controllers + service)



Persistence layer  
(datastore)

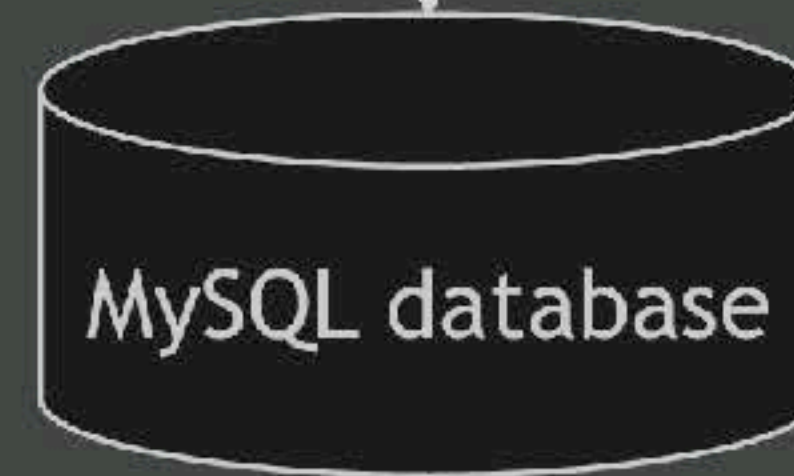


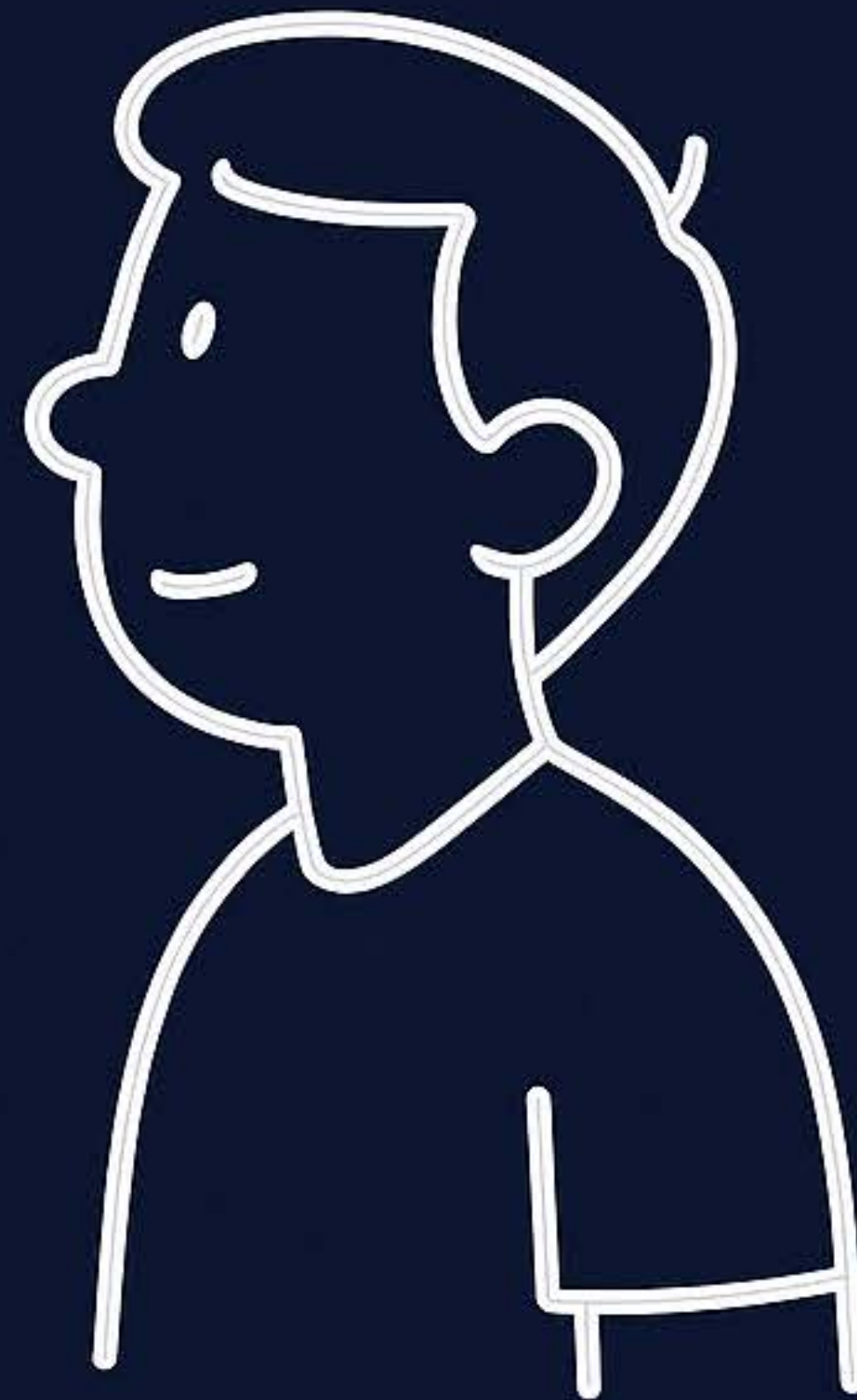
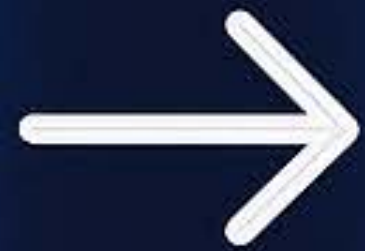
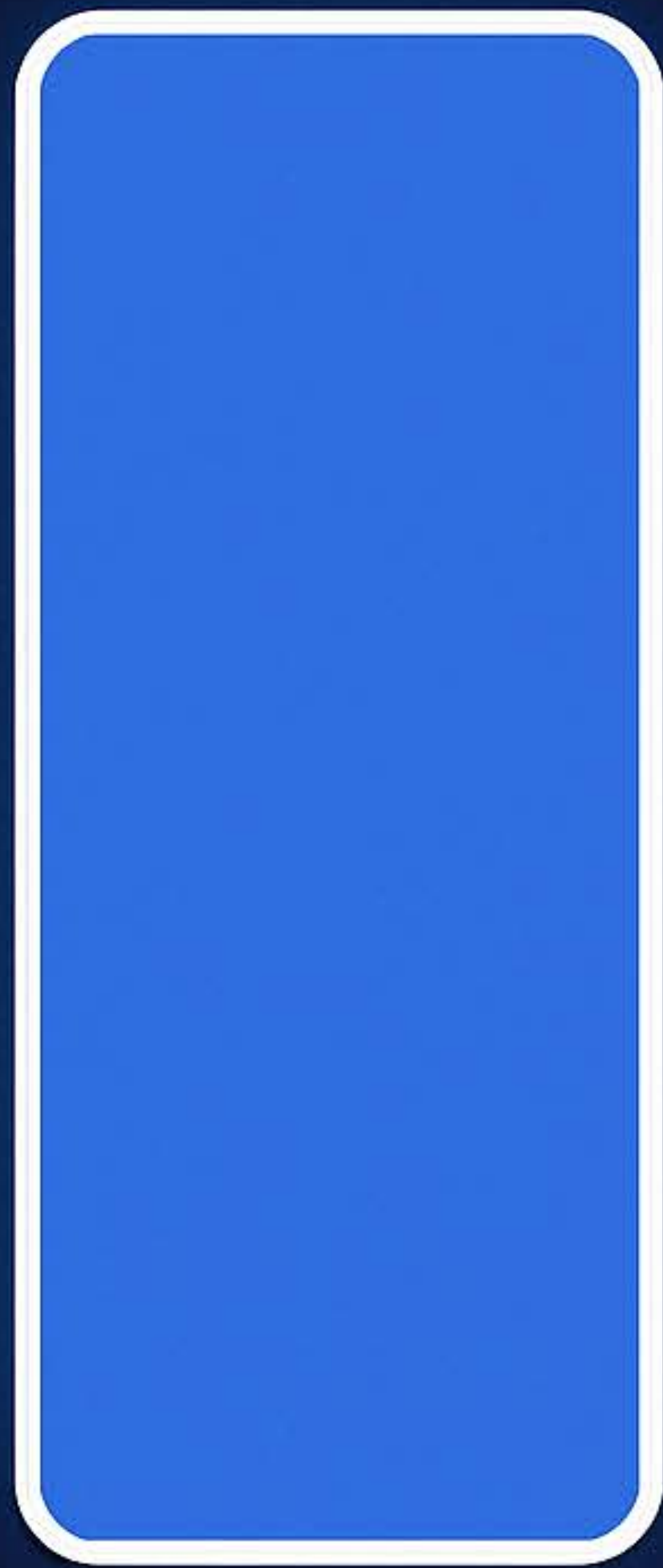
## Legacy code

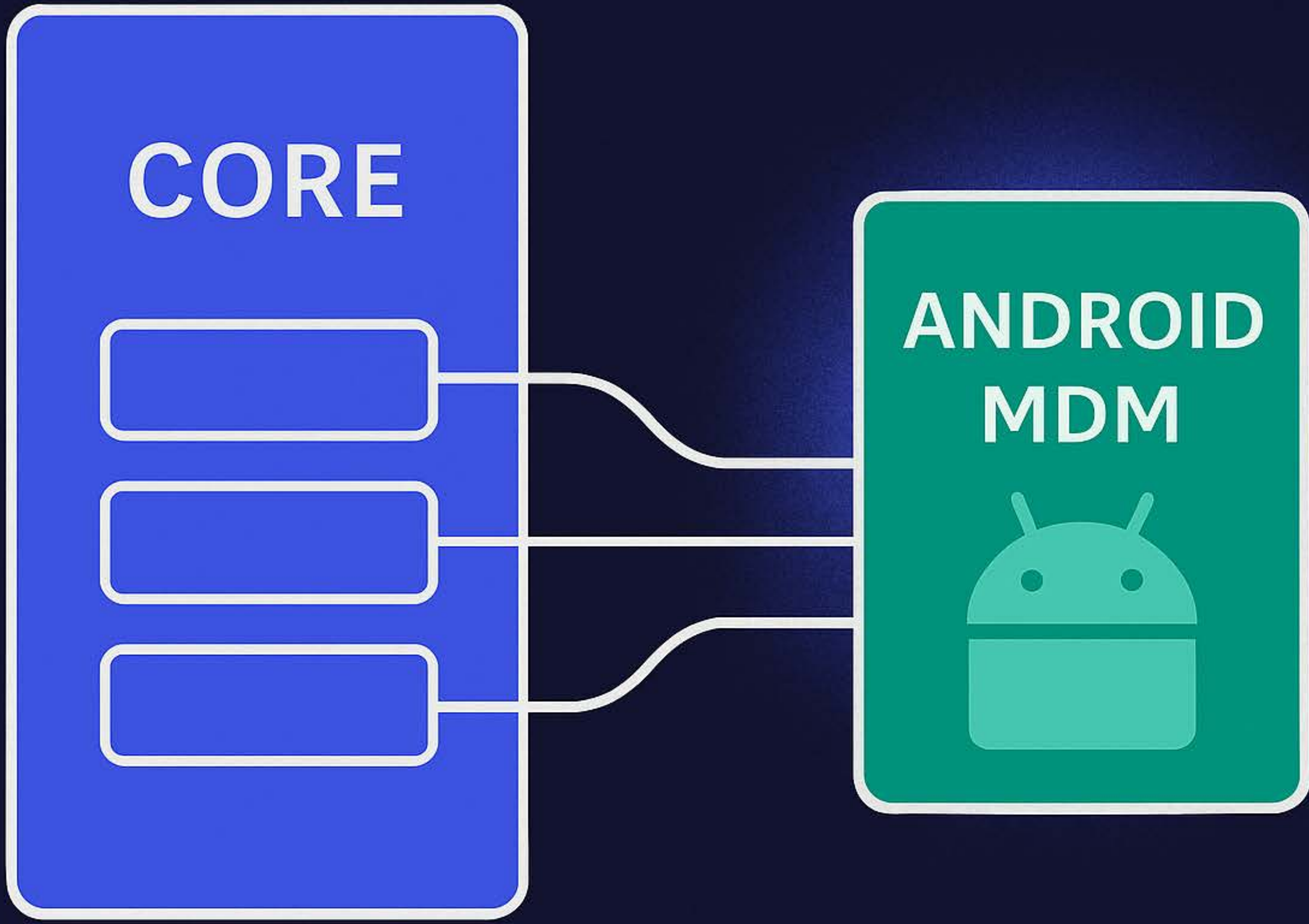
API layer  
(controllers + service)



Persistence layer  
(datastore)







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

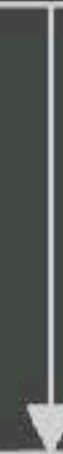
**The code was ready.  
The humans were not.**

## New feature

API layer  
(controllers + service)



Persistence layer  
(datastore)

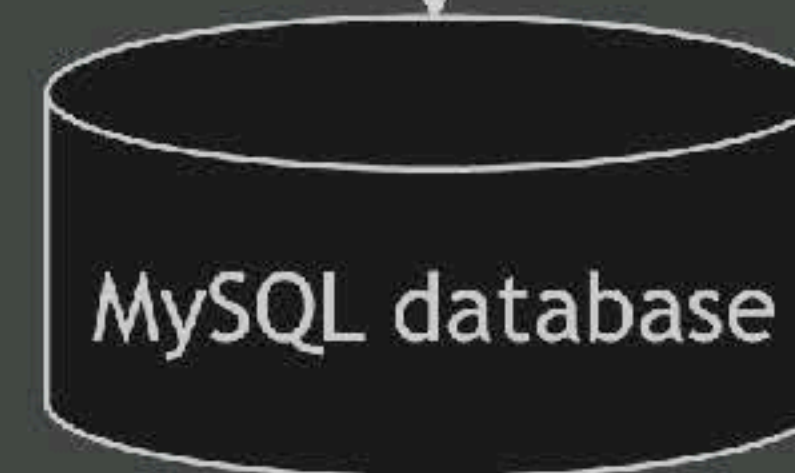


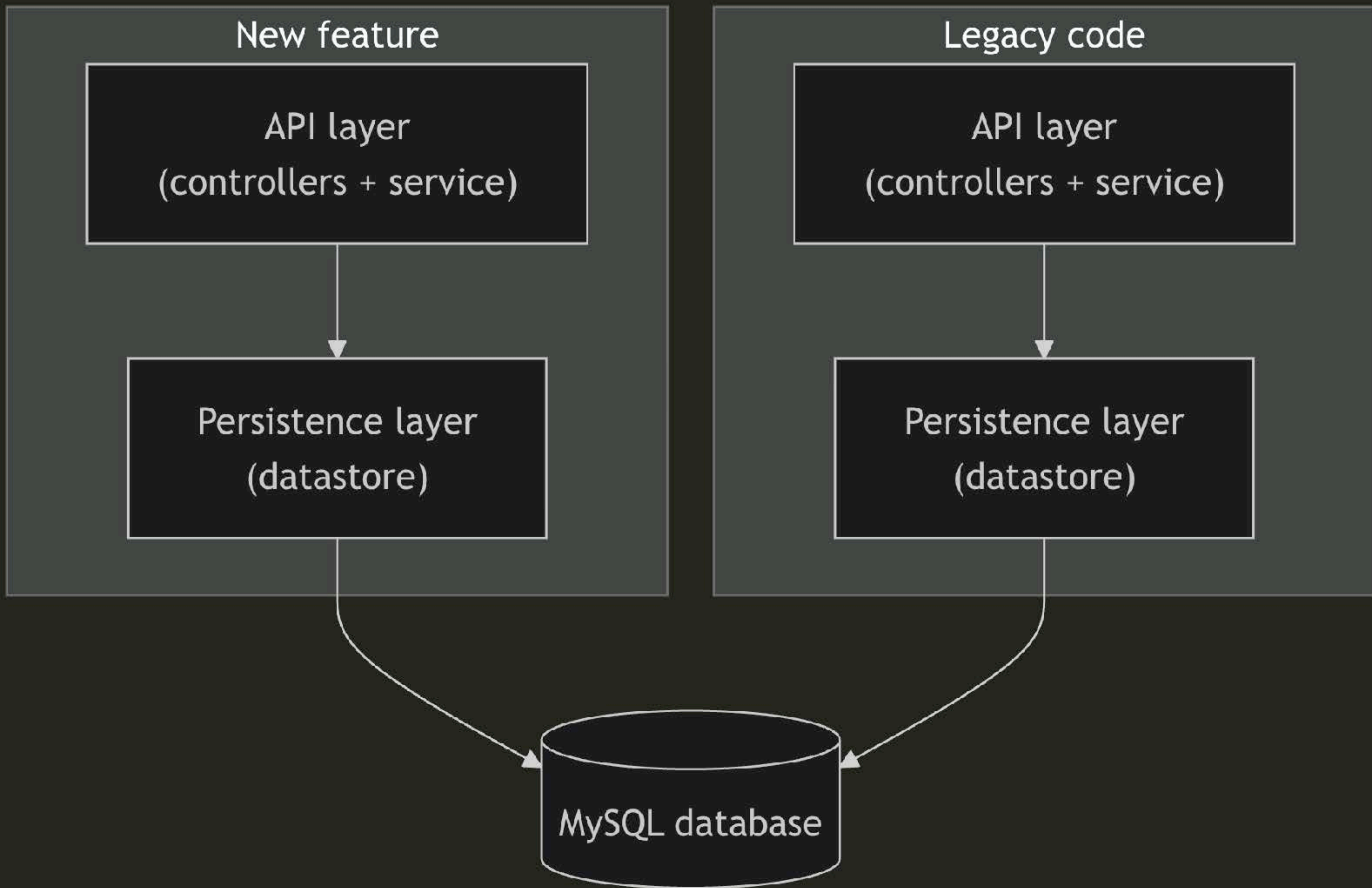
## Legacy code

API layer  
(controllers + service)



Persistence layer  
(datastore)

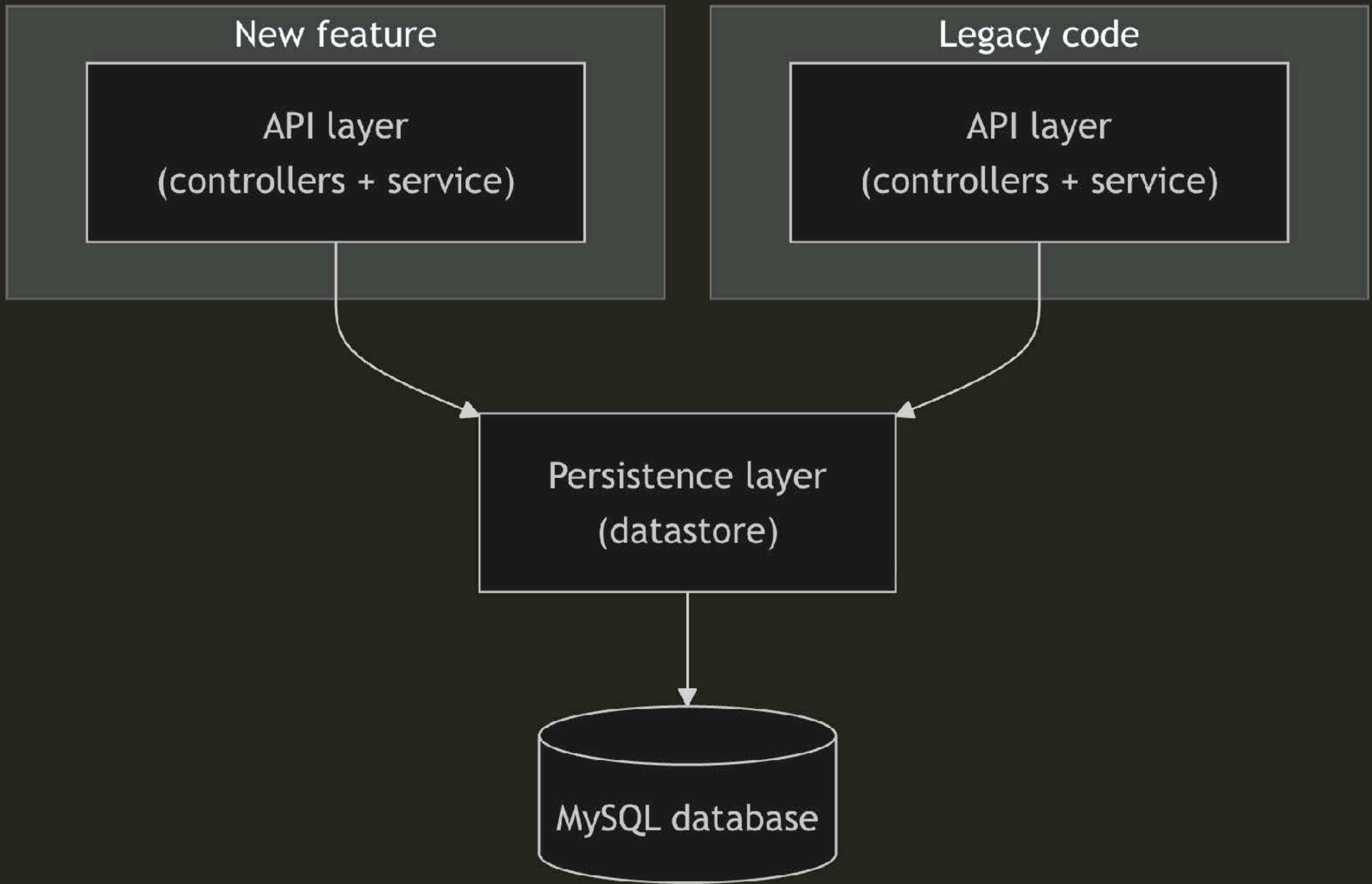




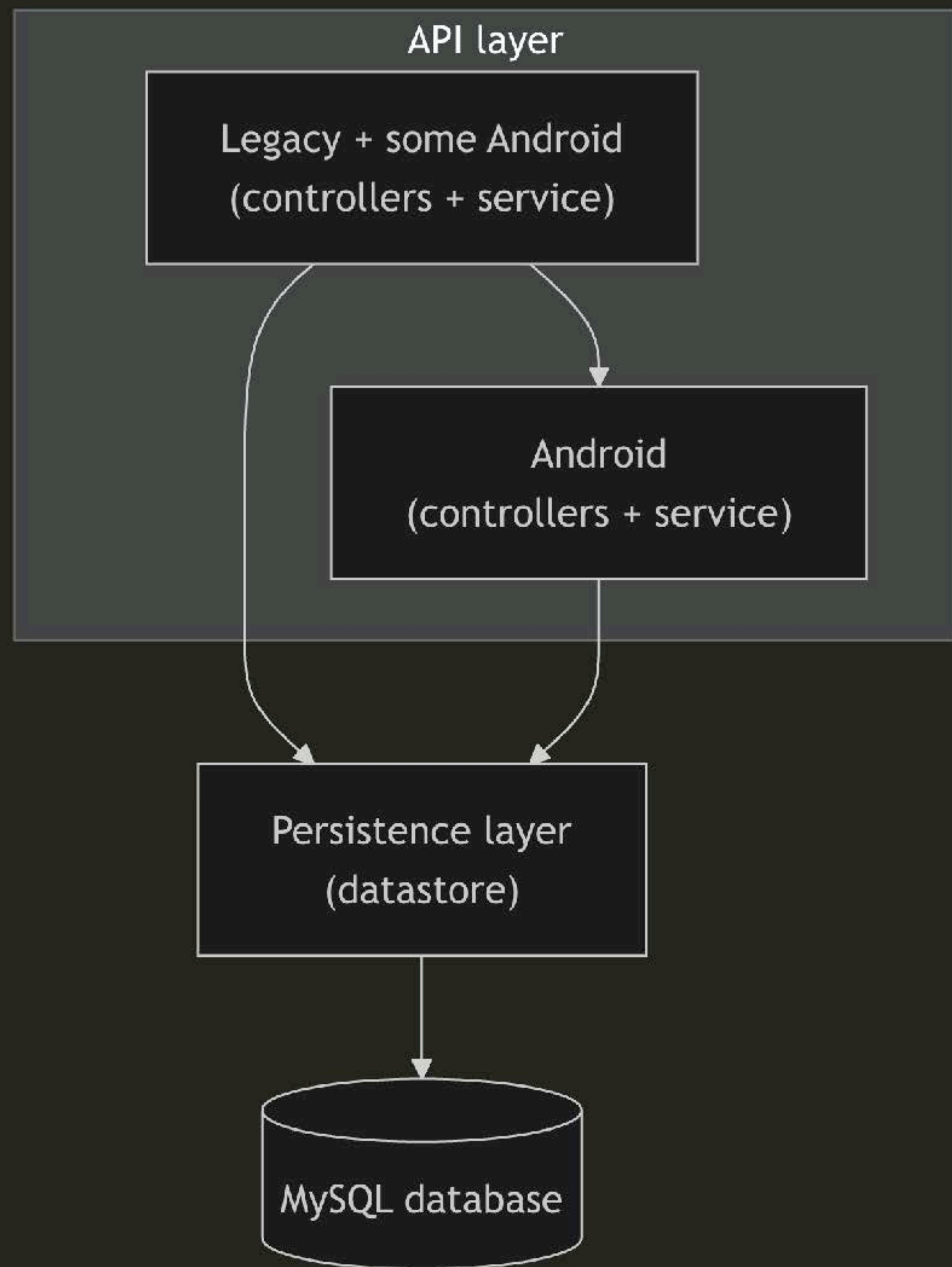
**We agreed on the what.  
We disagreed on the how deep.**

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





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





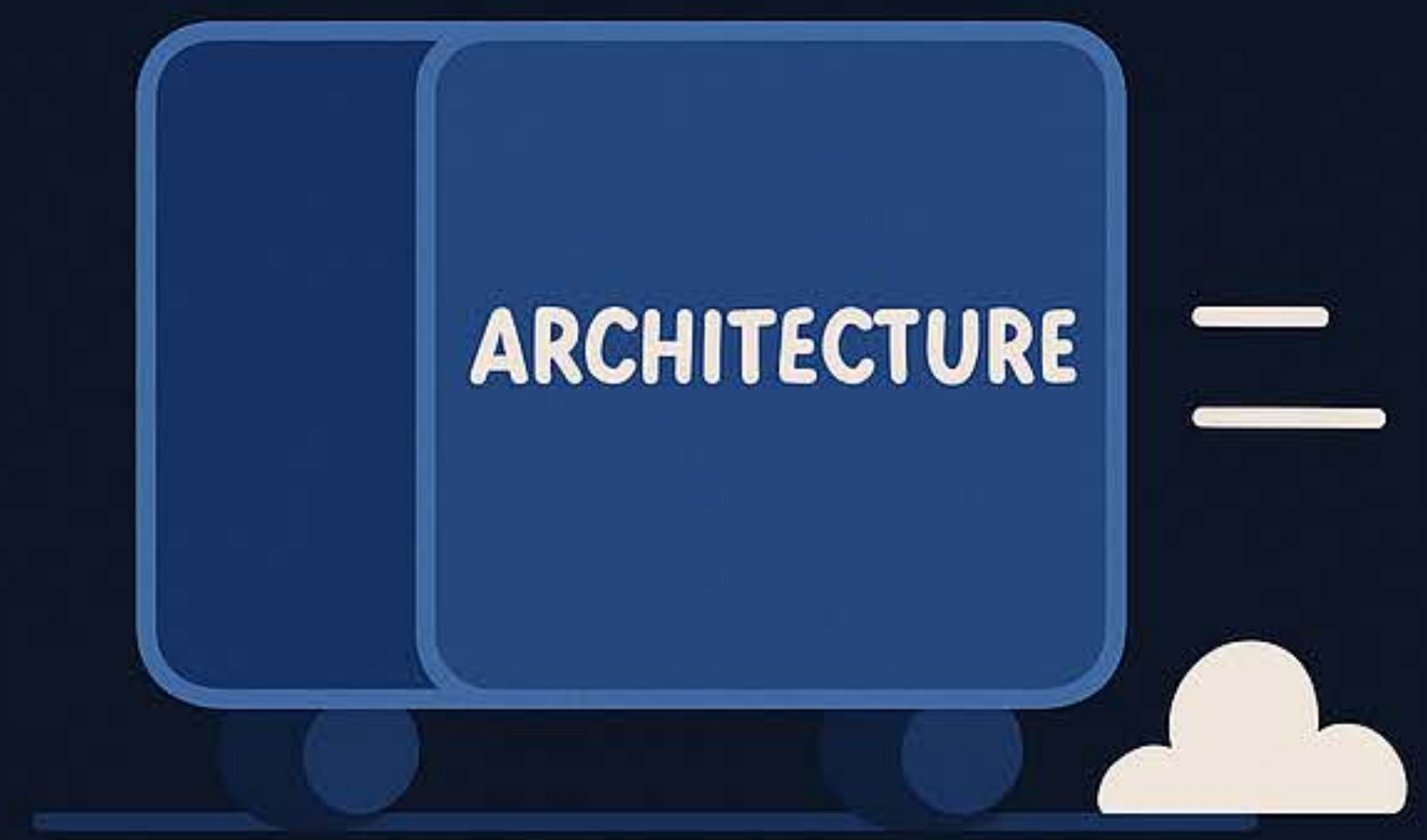
**Our biggest architectural dependency  
wasn't a package.  
It was shared understanding.**

**Architecture without buy-in is  
just a suggestion.**

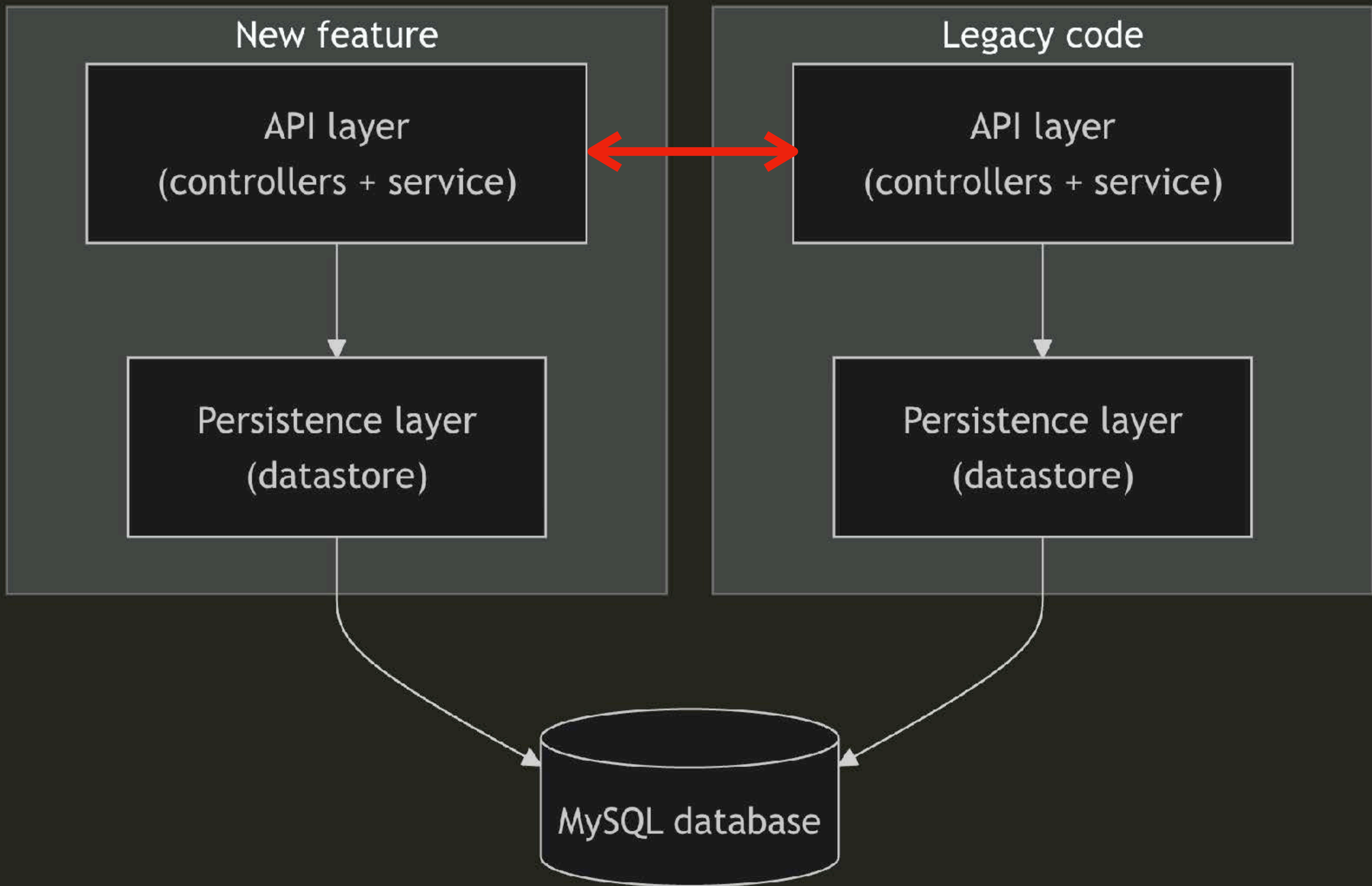


**Architecture is  
code + conversations**

**What is an architectural change,  
anyway?**



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



Experience is the name everyone gives to  
their mistakes.

Oscar Wilde

**I didn't need better ideas.  
I needed backup.**



**It's less risky when someone else  
shipped it**

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

- **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>**

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



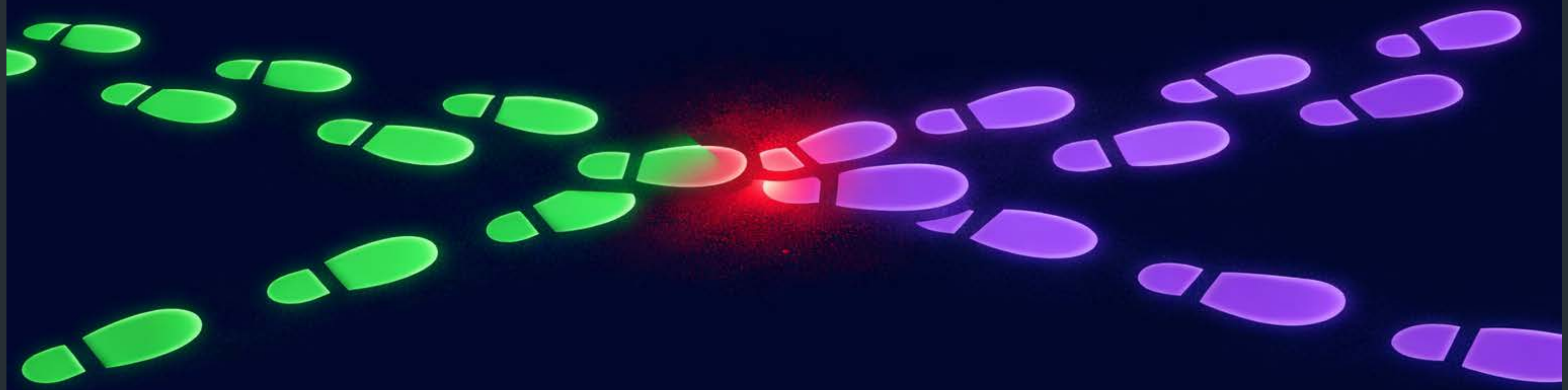
**It felt like pushing a boulder uphill...  
and watching it roll back down overnight**

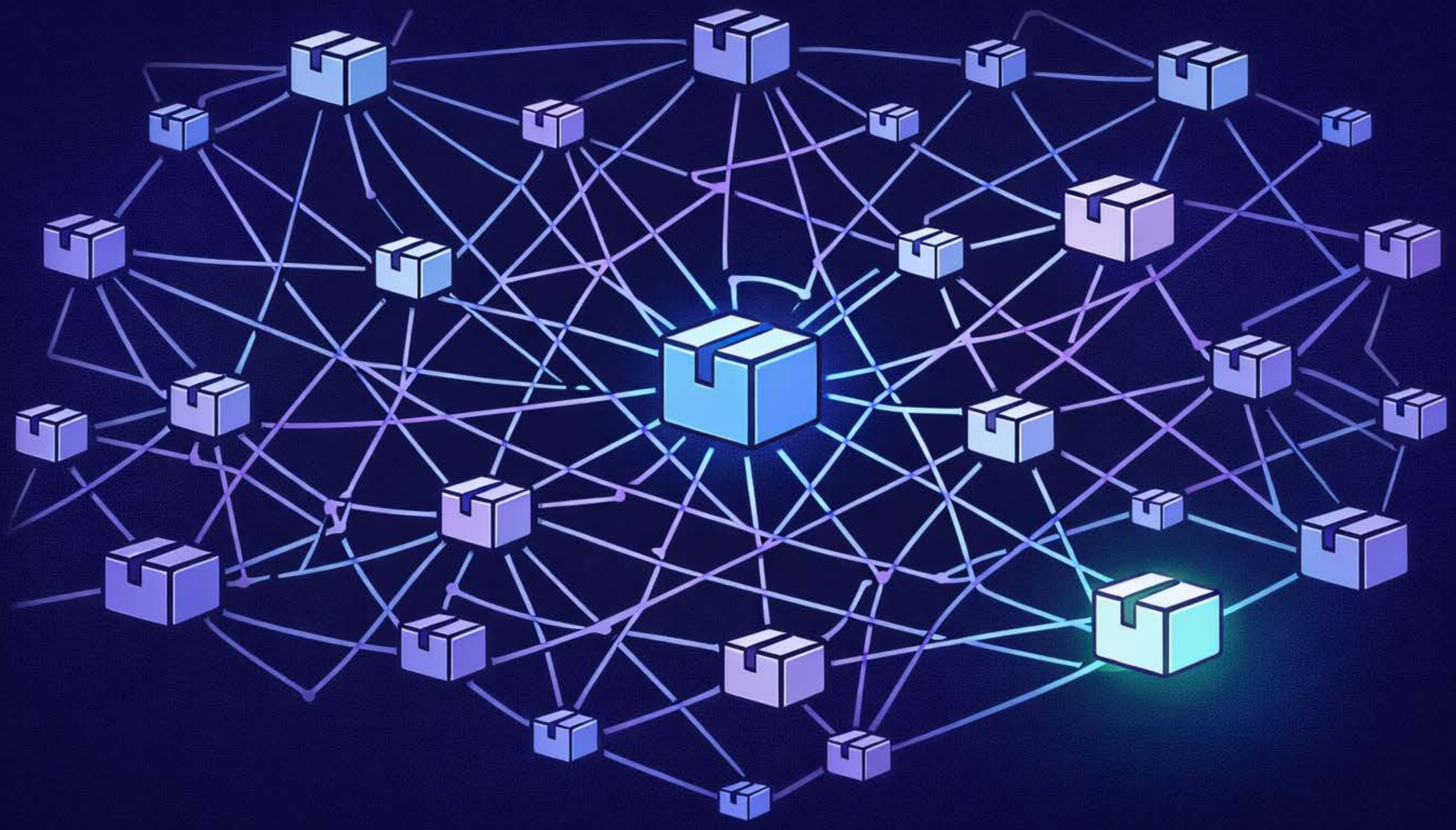
# Understanding doesn't spread by default

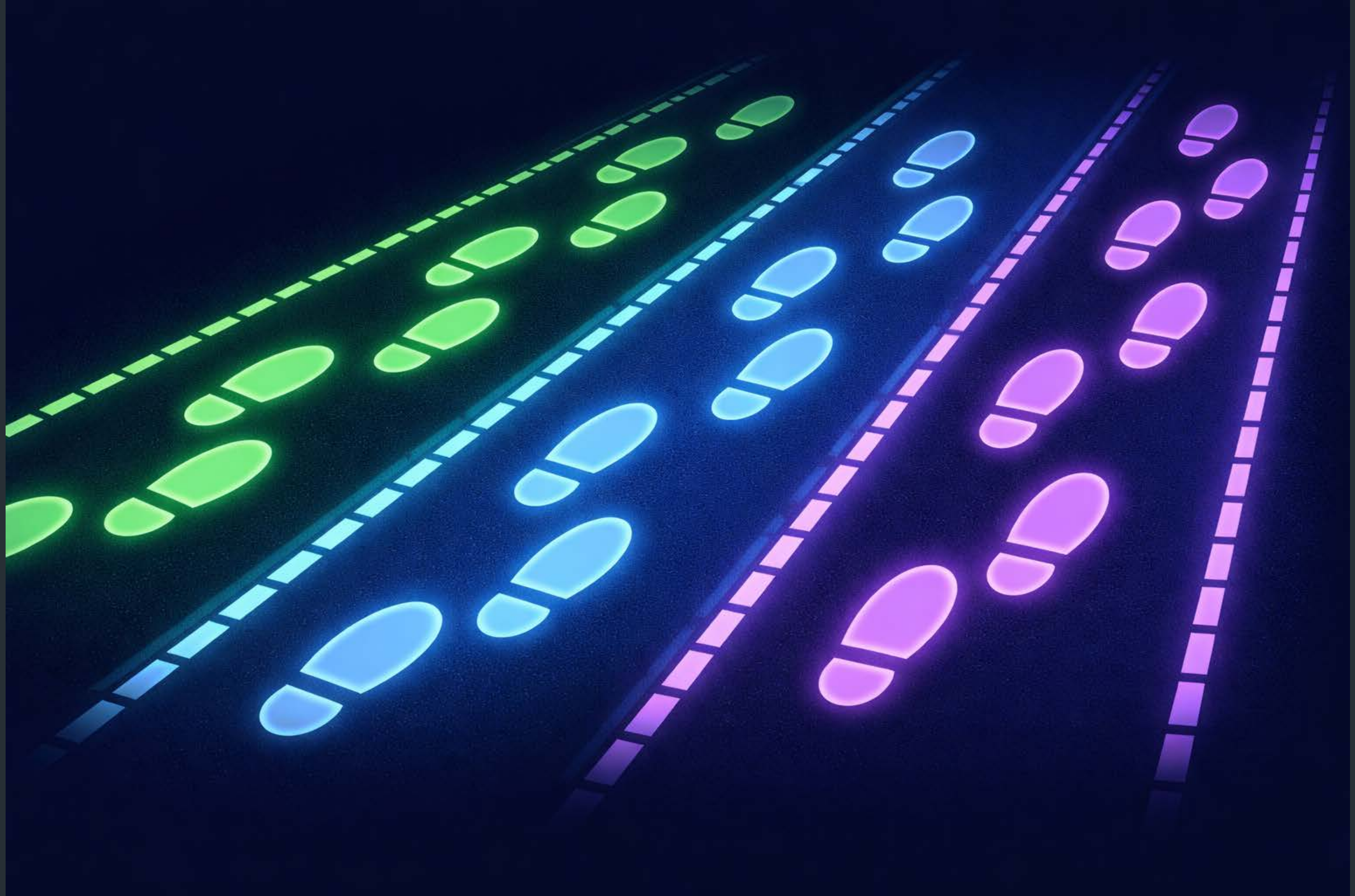


# Why Modularization Matters



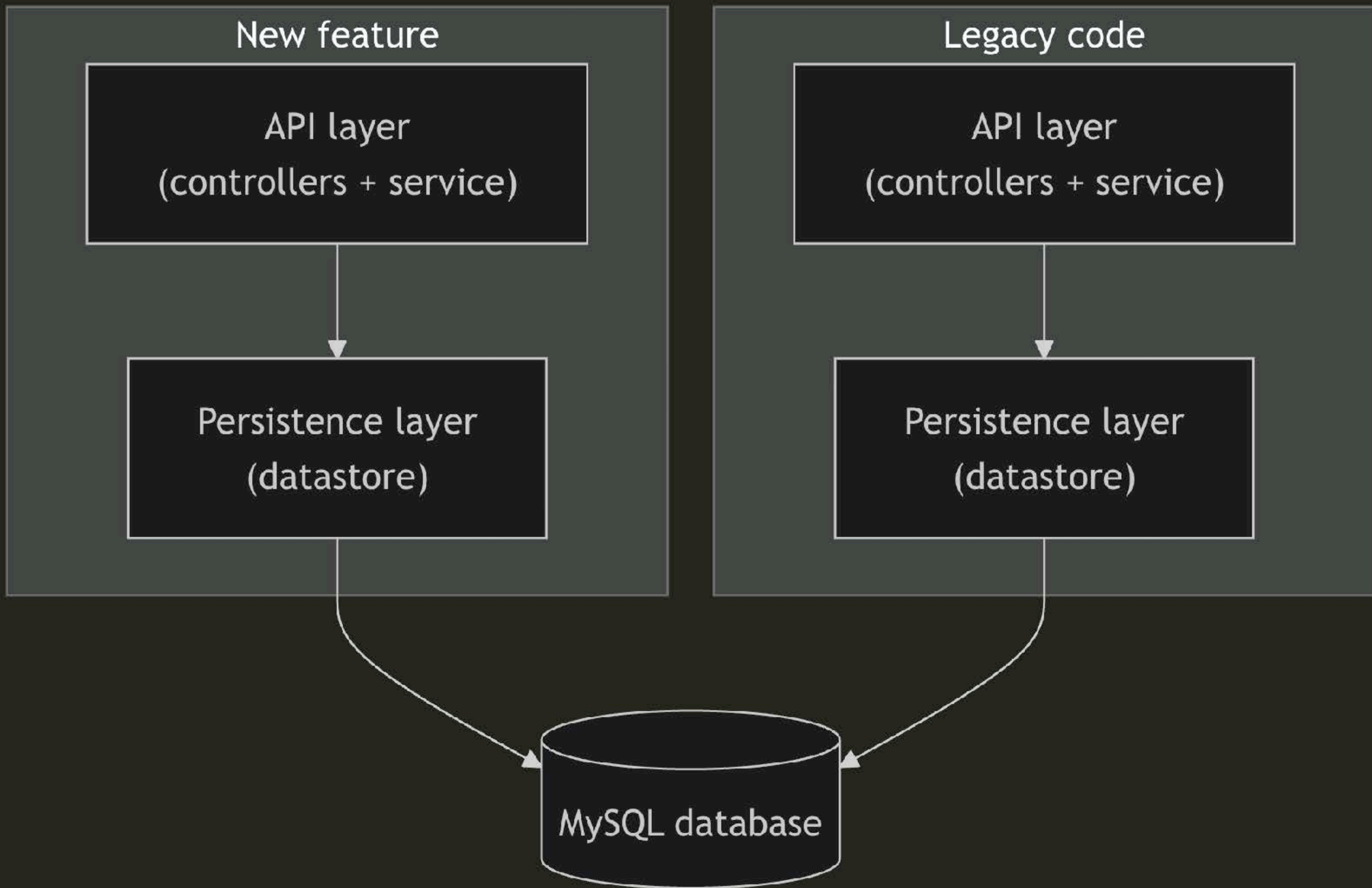






**Modularity is what lets teams move fast without asking permission**





**API calls preserve boundaries.  
Database shortcuts erase them.**

```
server/activity/
├── user.go           # Public types for ACL (User, UserProvider interface)
├── arch_test.go     # Architecture boundary tests
├── api/
│   └── list_activities.go # Public API types
├── bootstrap/
│   └── bootstrap.go     # Public bootstrap for wiring
├── internal/
│   ├── types/
│   │   └── activity.go  # Internal types
│   ├── mysql/
│   │   ├── activity.go  # MySQL datastore implementation
│   │   └── activity_test.go # Datastore unit tests
│   ├── service/
│   │   ├── service.go   # Service implementation
│   │   ├── service_test.go # Service unit tests
│   │   ├── handler.go   # HTTP handler registration
│   │   ├── handler_test.go # Handler unit tests
│   │   └── endpoint_utils.go # Endpoint utilities (decoder, encoder, middleware wiring)
│   └── tests/
│       └── integration_test.go # Full stack integration tests (HTTP -> Service -> MySQL)
server/acl/activityacl/
└── legacy_adapter.go # ACL adapter for legacy dependencies
```



**Just because it exists doesn't  
mean you own it**



**Databases scale data.  
Modules scale teams.**

9:41



## Ports and Adapters

### Block This Contact?

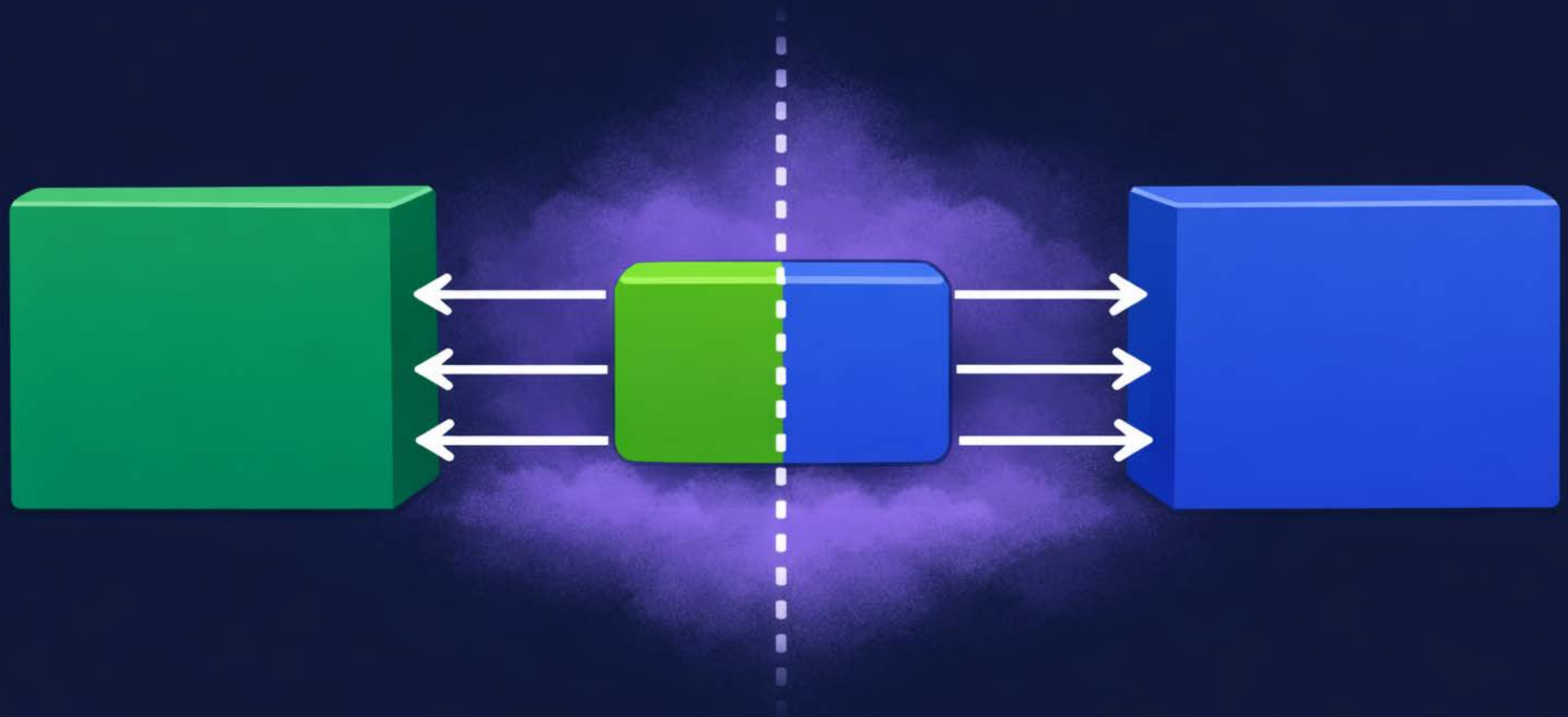
You will no longer receive messages, calls, notifications from this person.

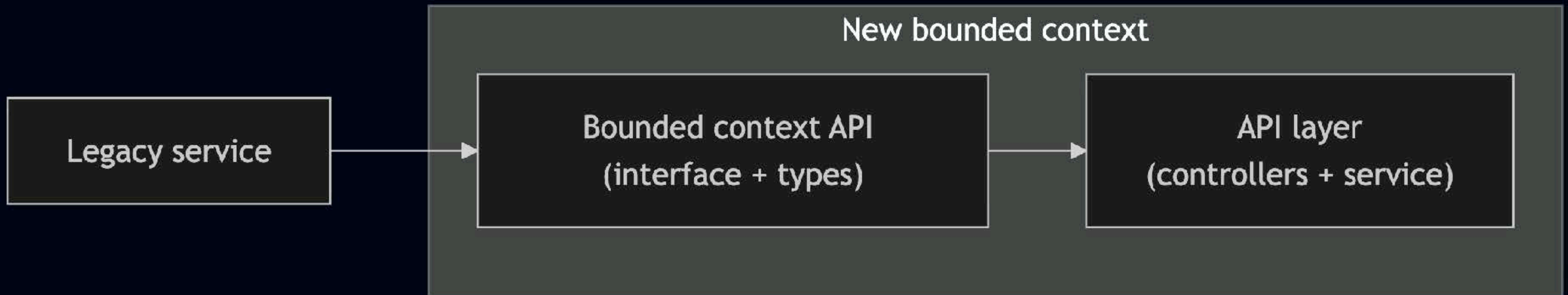
**Block Contact**

**Cancel**



# Explicit module boundaries

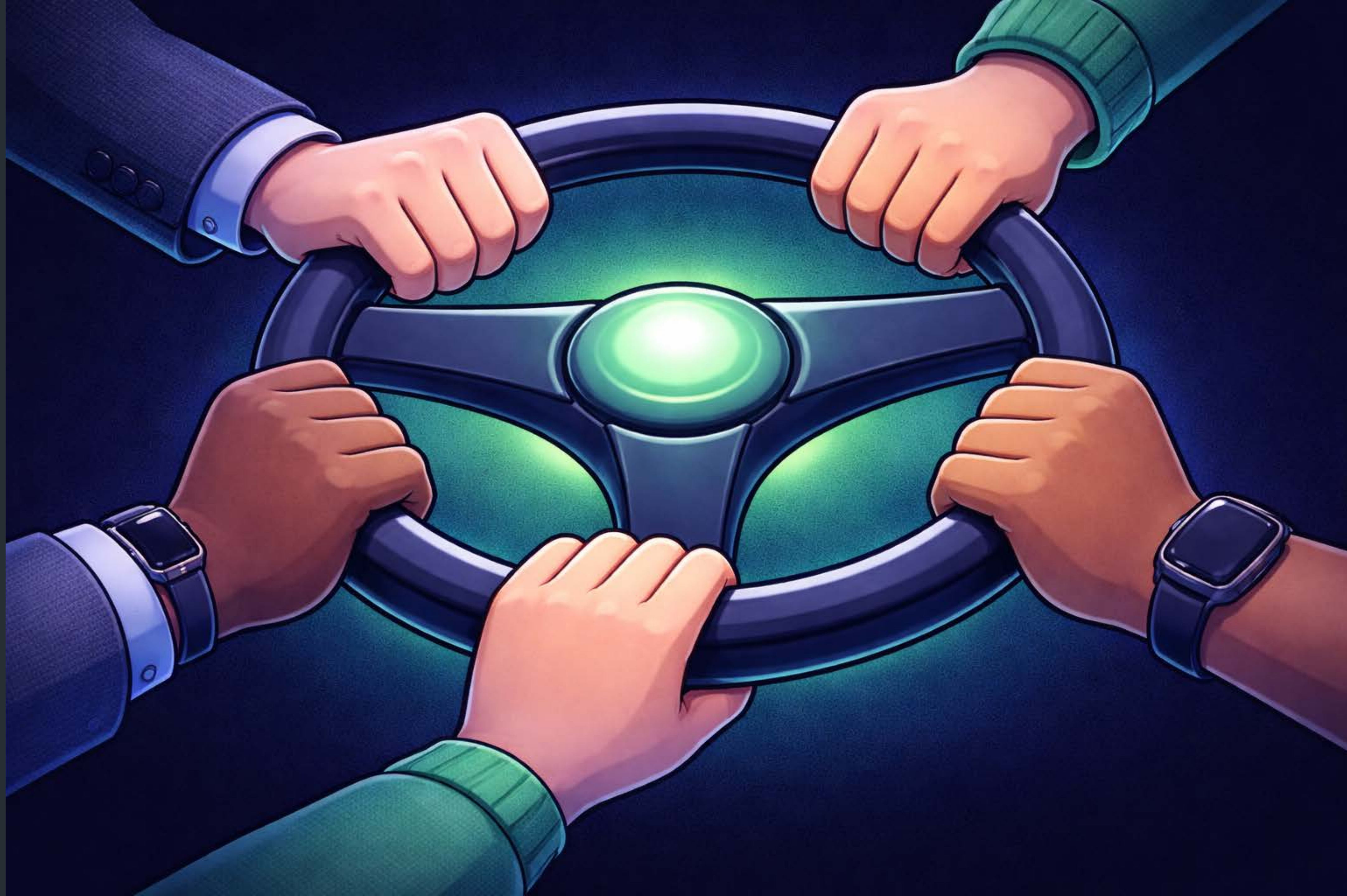






**No matter how good the design is,  
"someone else's idea" never feels  
like the right one**

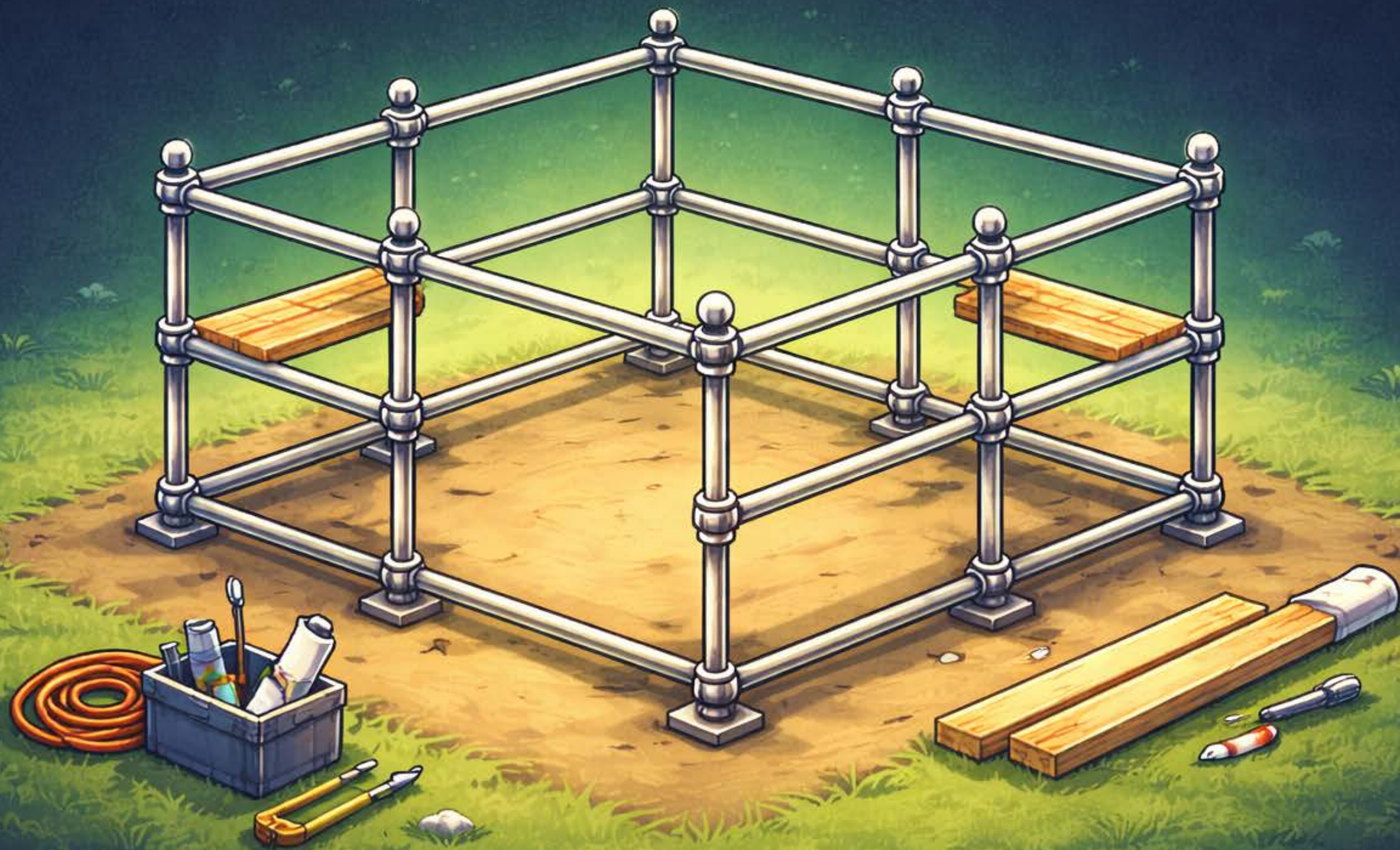


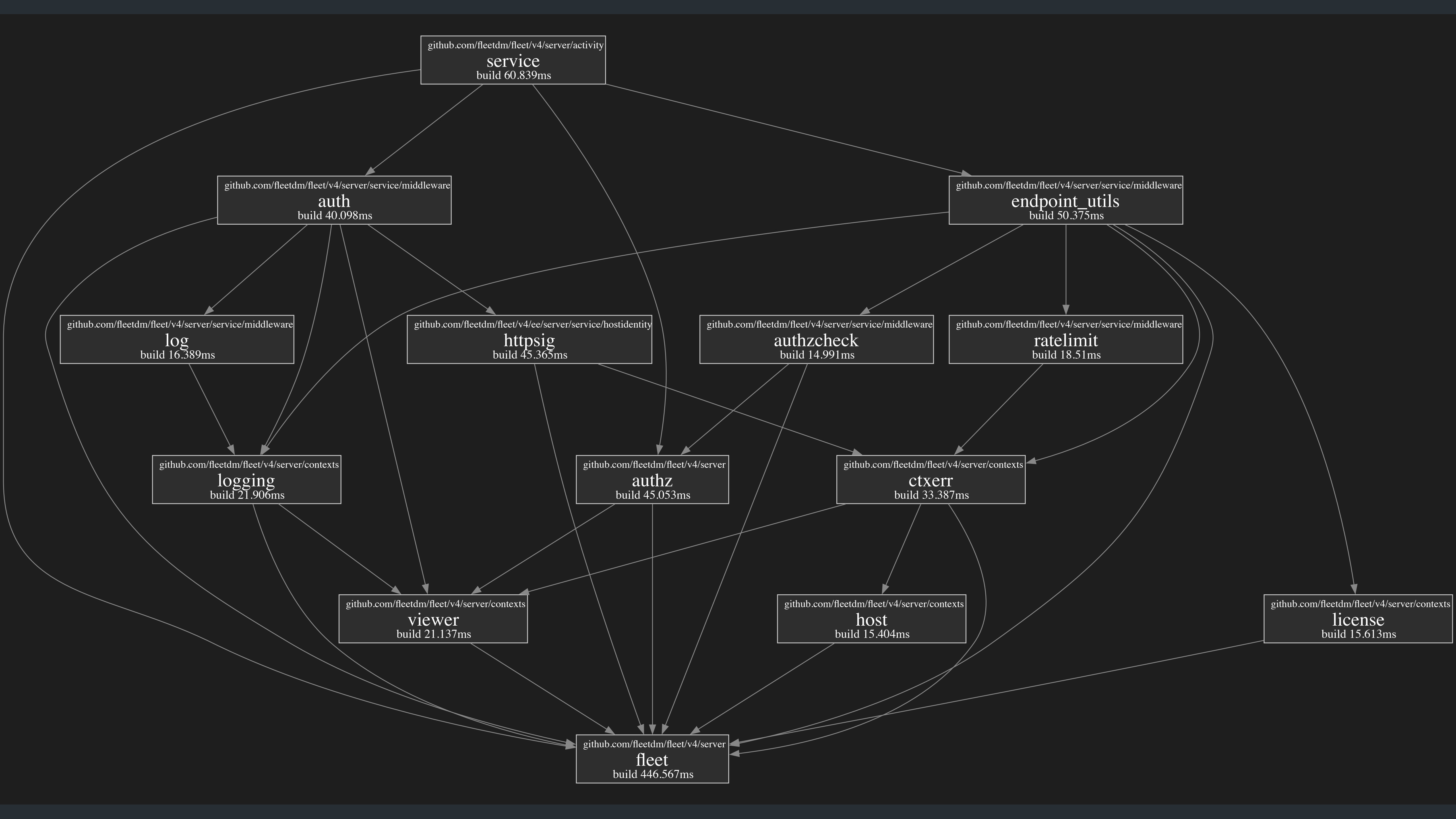


**This was architecture by  
small, boring steps.  
On purpose.**



**You don't discover coupling  
on a whiteboard**





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



**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**

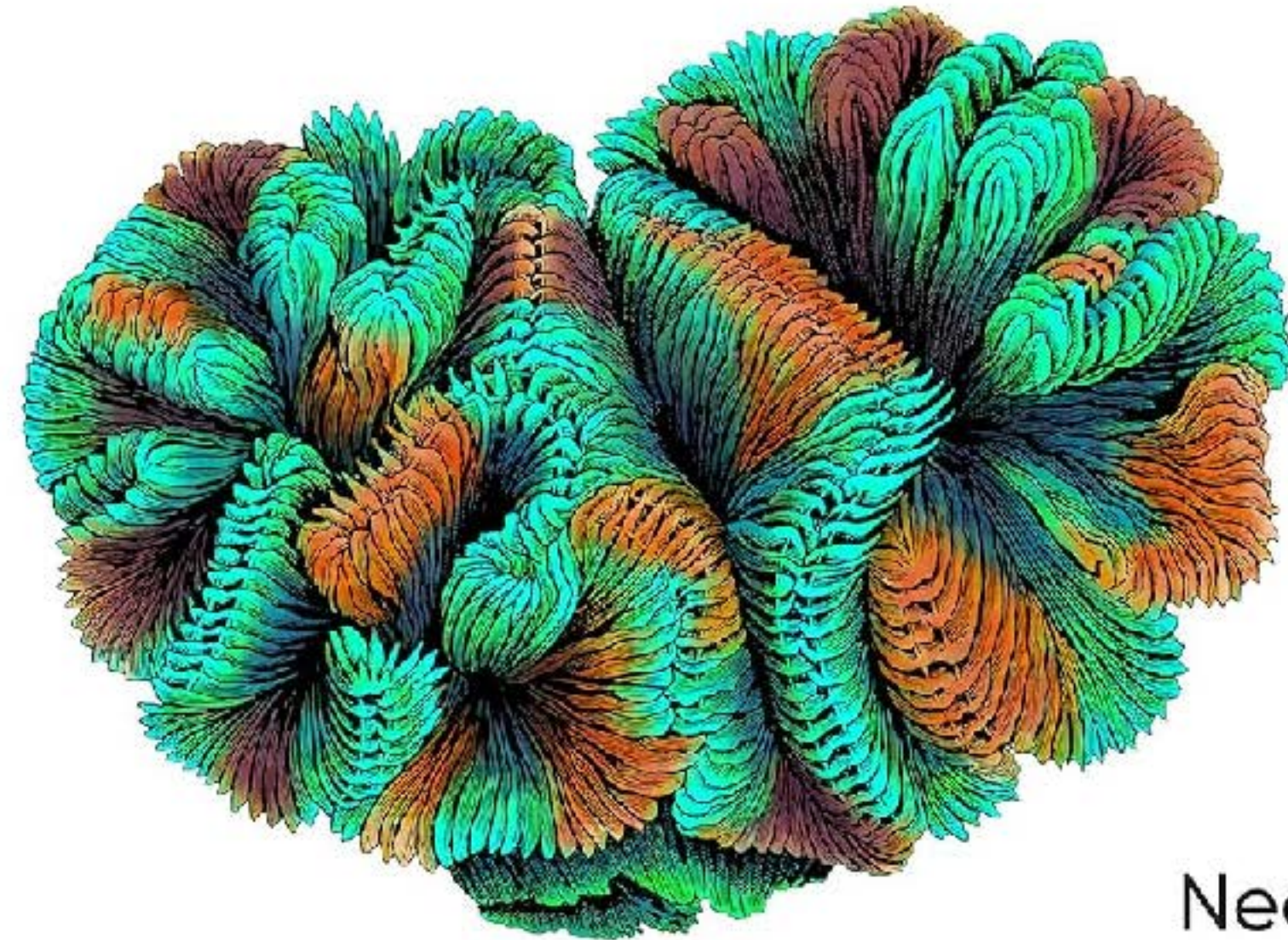


O'REILLY®

2nd Edition

# Building Evolutionary Architectures

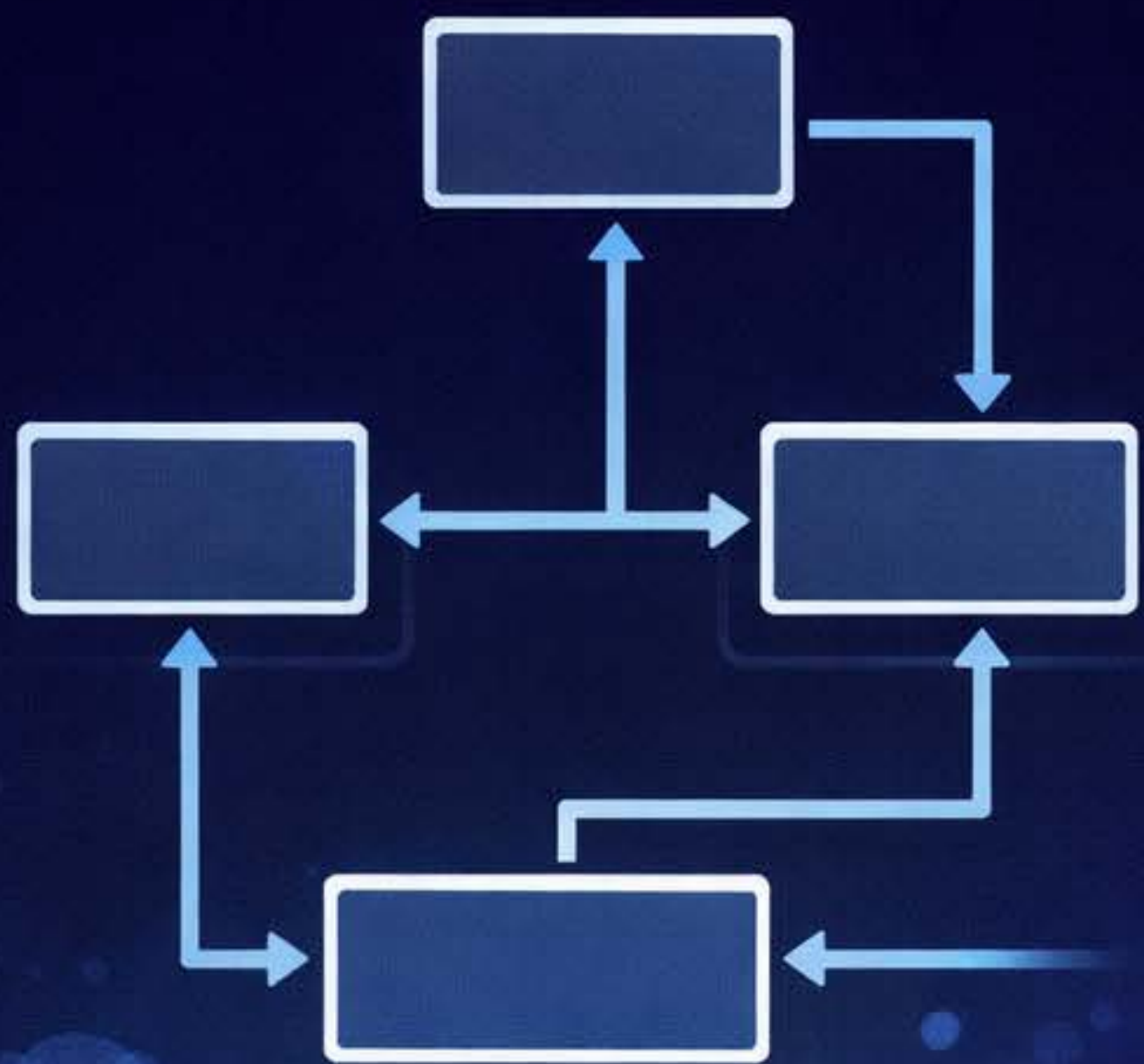
Automated Software Governance



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

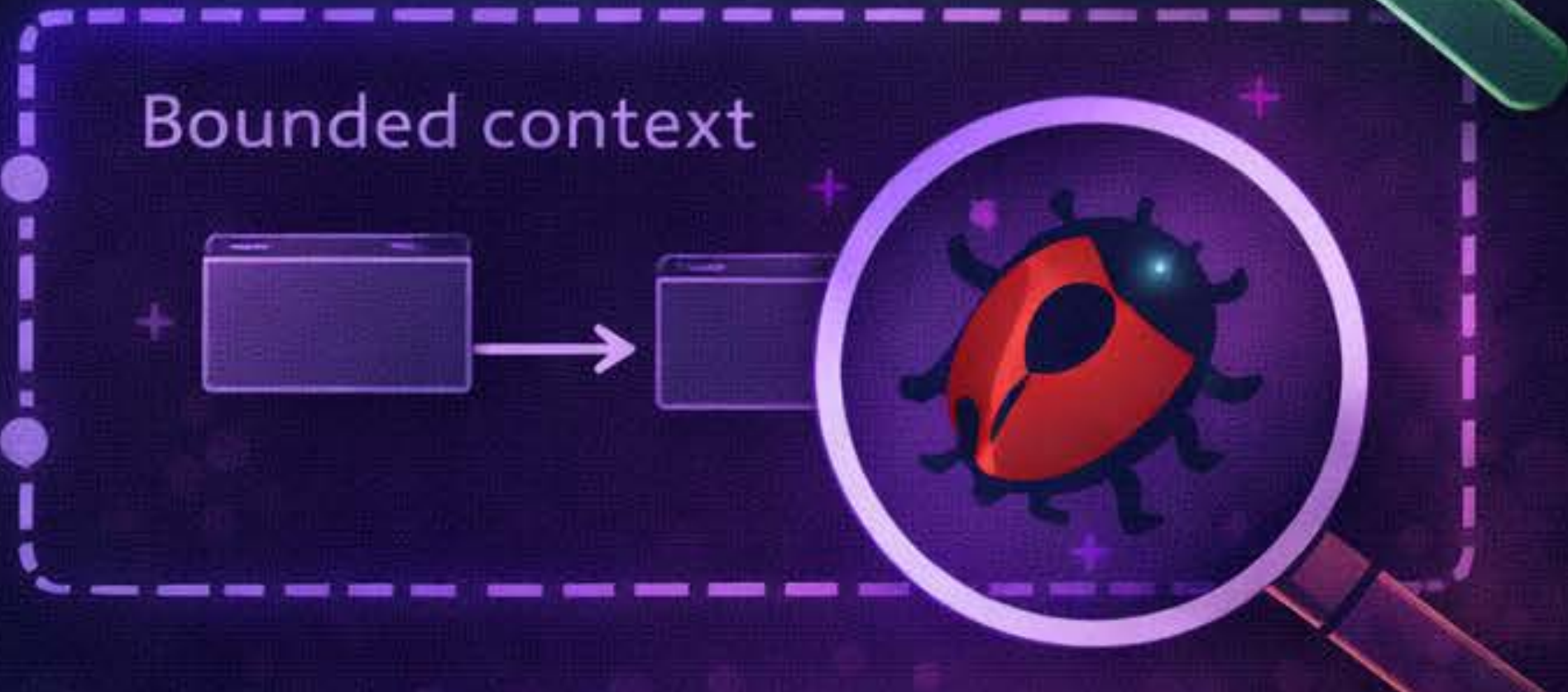
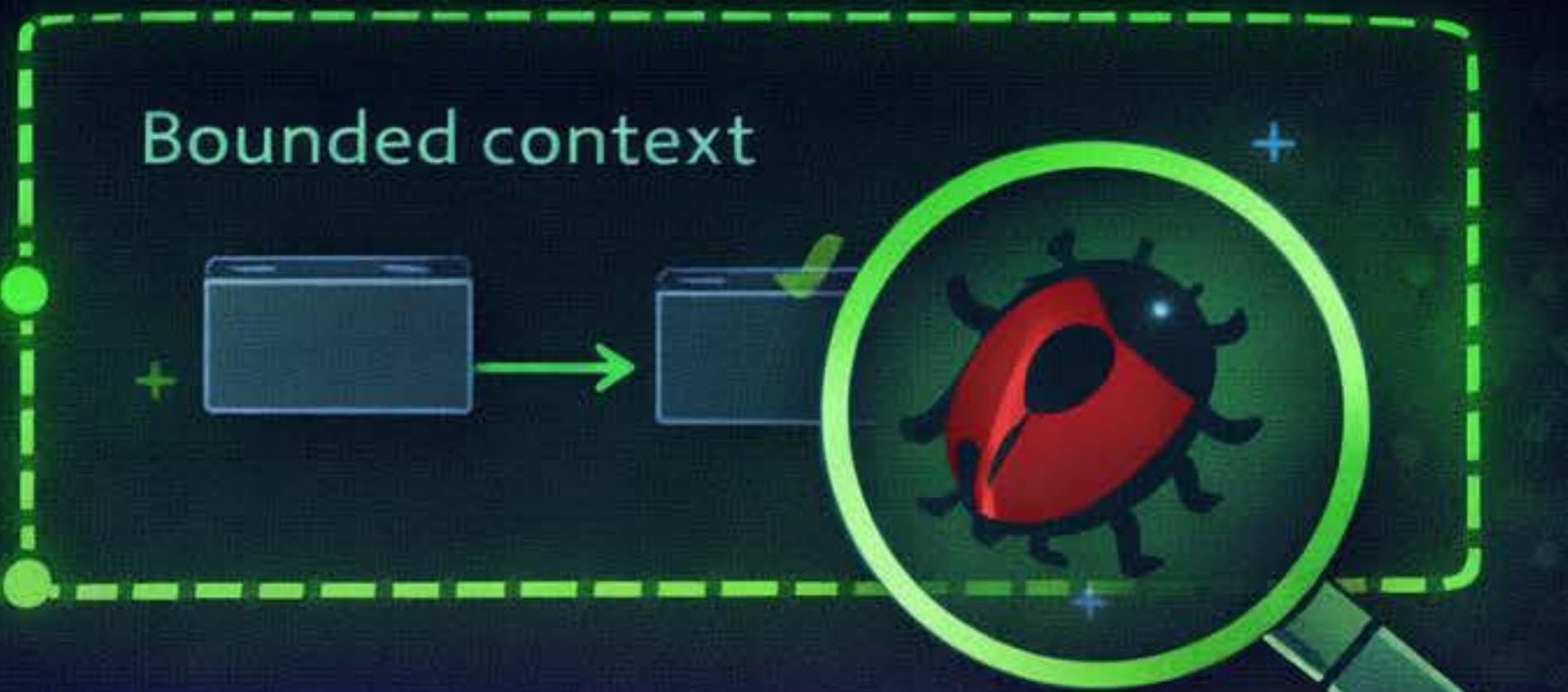


**Opinions are cheap.  
Running code isn't.**



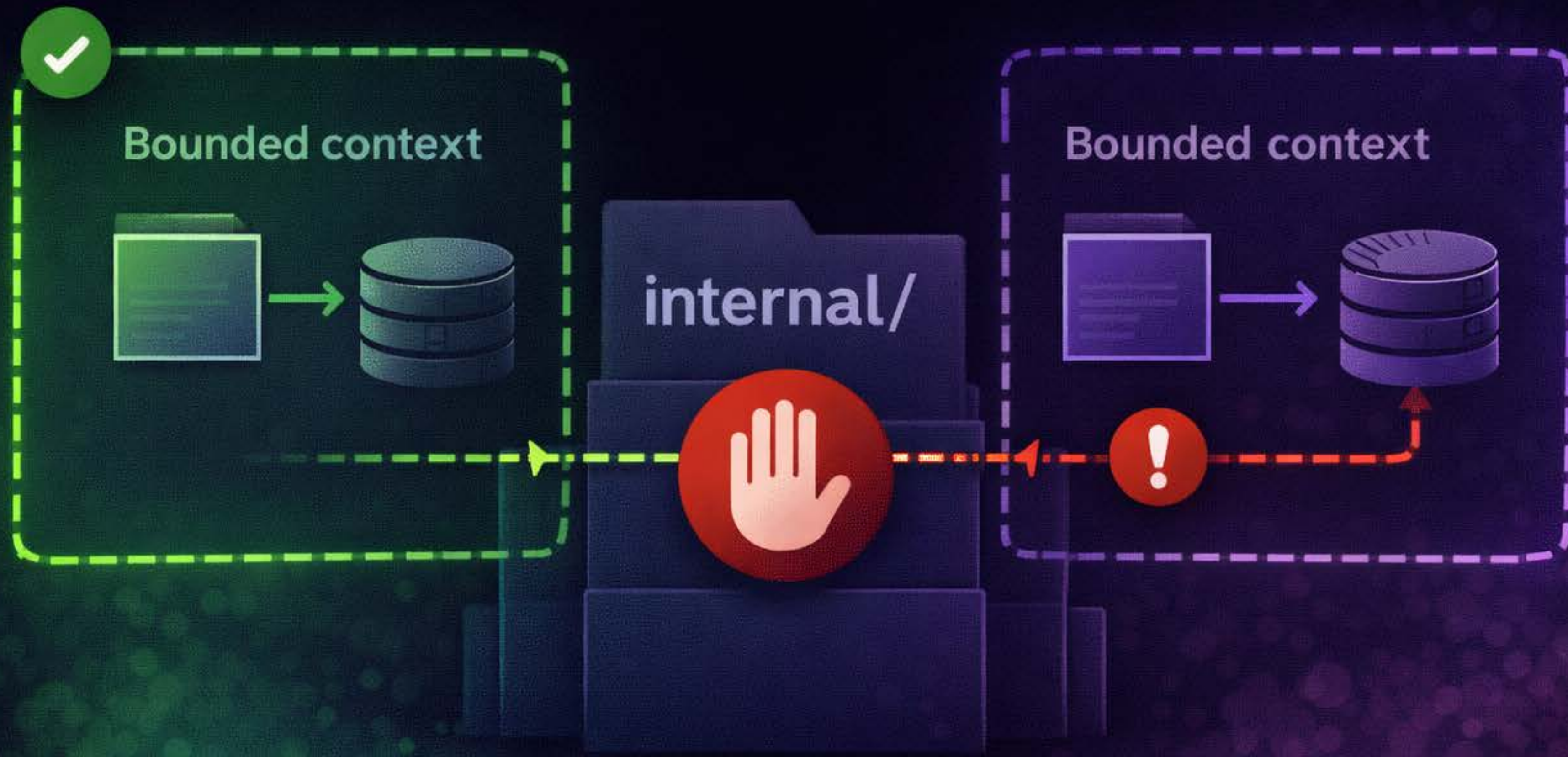
**Less philosophy.  
More broken builds.**

**Refactoring surfaced  
what we ignored**



```
server/activity/
├── user.go           # Public types for ACL (User, UserProvider interface)
├── arch_test.go     # Architecture boundary tests
├── api/
│   └── list_activities.go # Public API types
├── bootstrap/
│   └── bootstrap.go     # Public bootstrap for wiring
├── internal/
│   ├── types/
│   │   └── activity.go  # Internal types
│   ├── mysql/
│   │   ├── activity.go  # MySQL datastore implementation
│   │   └── activity_test.go # Datastore unit tests
│   ├── service/
│   │   ├── service.go   # Service implementation
│   │   ├── service_test.go # Service unit tests
│   │   ├── handler.go   # HTTP handler registration
│   │   ├── handler_test.go # Handler unit tests
│   │   └── endpoint_utils.go # Endpoint utilities (decoder, encoder, middleware wiring)
│   └── tests/
│       └── integration_test.go # Full stack integration tests (HTTP -> Service -> MySQL)
server/acl/activityacl/
└── legacy_adapter.go # ACL adapter for legacy dependencies
```



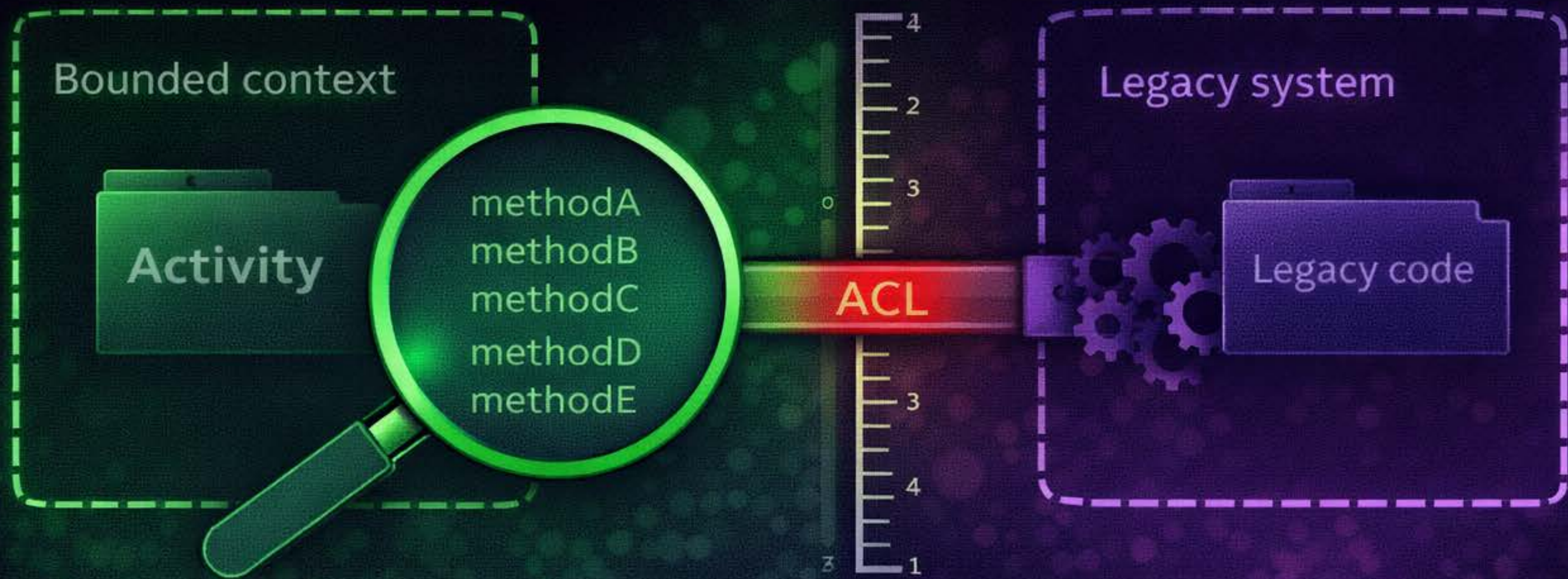


```
server/activity/
├── user.go           # Public types for ACL (User, UserProvider interface)
├── arch_test.go     # Architecture boundary tests
├── api/
│   └── list_activities.go # Public API types
├── bootstrap/
│   └── bootstrap.go    # Public bootstrap for wiring
├── internal/
│   ├── types/
│   │   └── activity.go # Internal types
│   ├── mysql/
│   │   ├── activity.go # MySQL datastore implementation
│   │   └── activity_test.go # Datastore unit tests
│   ├── service/
│   │   ├── service.go # Service implementation
│   │   ├── service_test.go # Service unit tests
│   │   ├── handler.go # HTTP handler registration
│   │   ├── handler_test.go # Handler unit tests
│   │   └── endpoint_utils.go # Endpoint utilities (decoder, encoder, middleware wiring)
│   └── tests/
│       └── integration_test.go # Full stack integration tests (HTTP -> Service -> MySQL)
server/acl/activityacl/
└── legacy_adapter.go # ACL adapter for legacy dependencies
```



**If wiring differs, behavior differs**





**CI got smaller.  
Signal got sharper.**

**Shared failures diluted ownership.  
Isolation fixed it.**

## Architecture tests



Dependency allowed



Dependency allowed



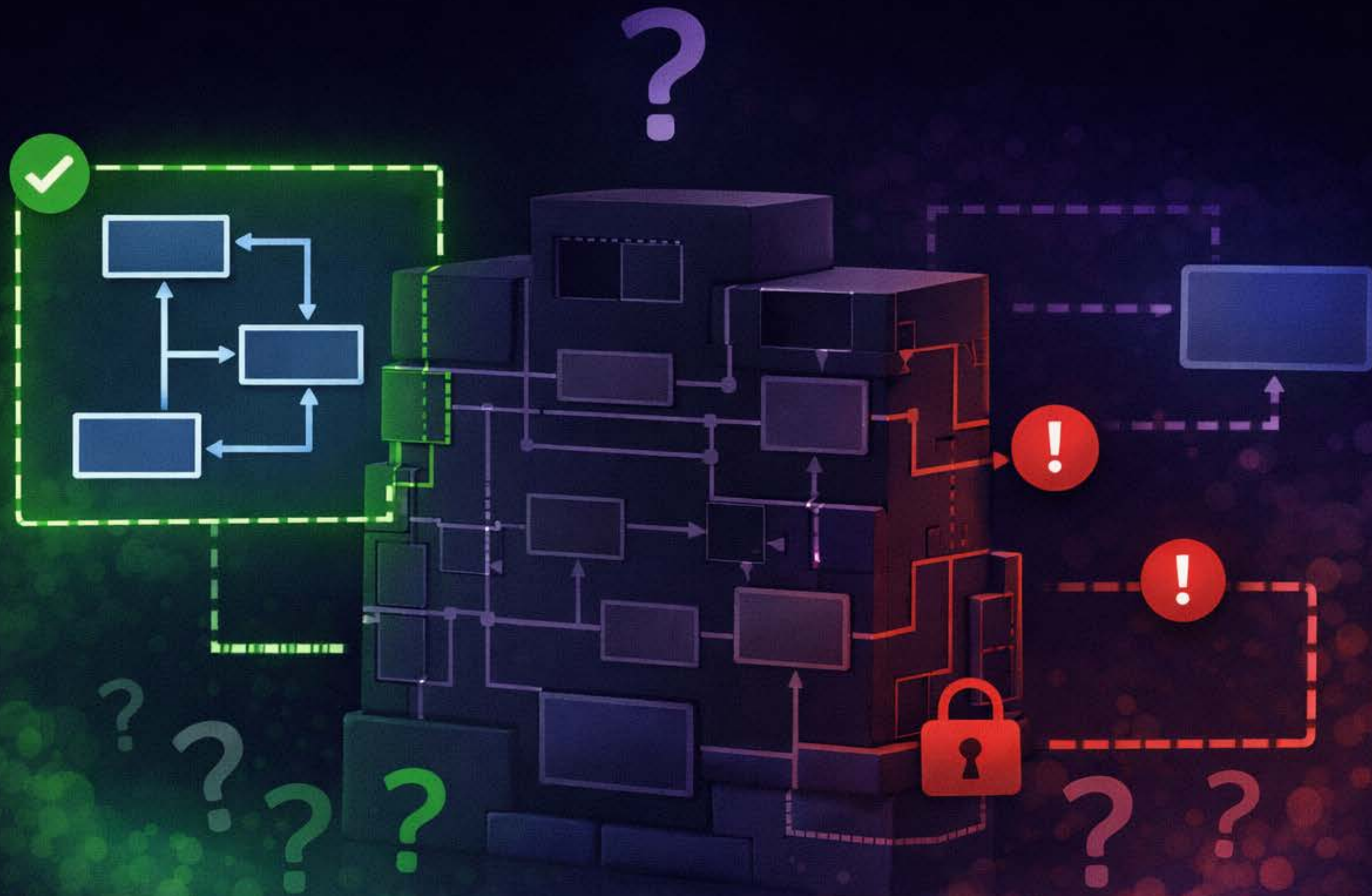
Dependency forbidden



000000101010010000  
00110000101001010101



**This isn't style.  
It's a constraint system.**

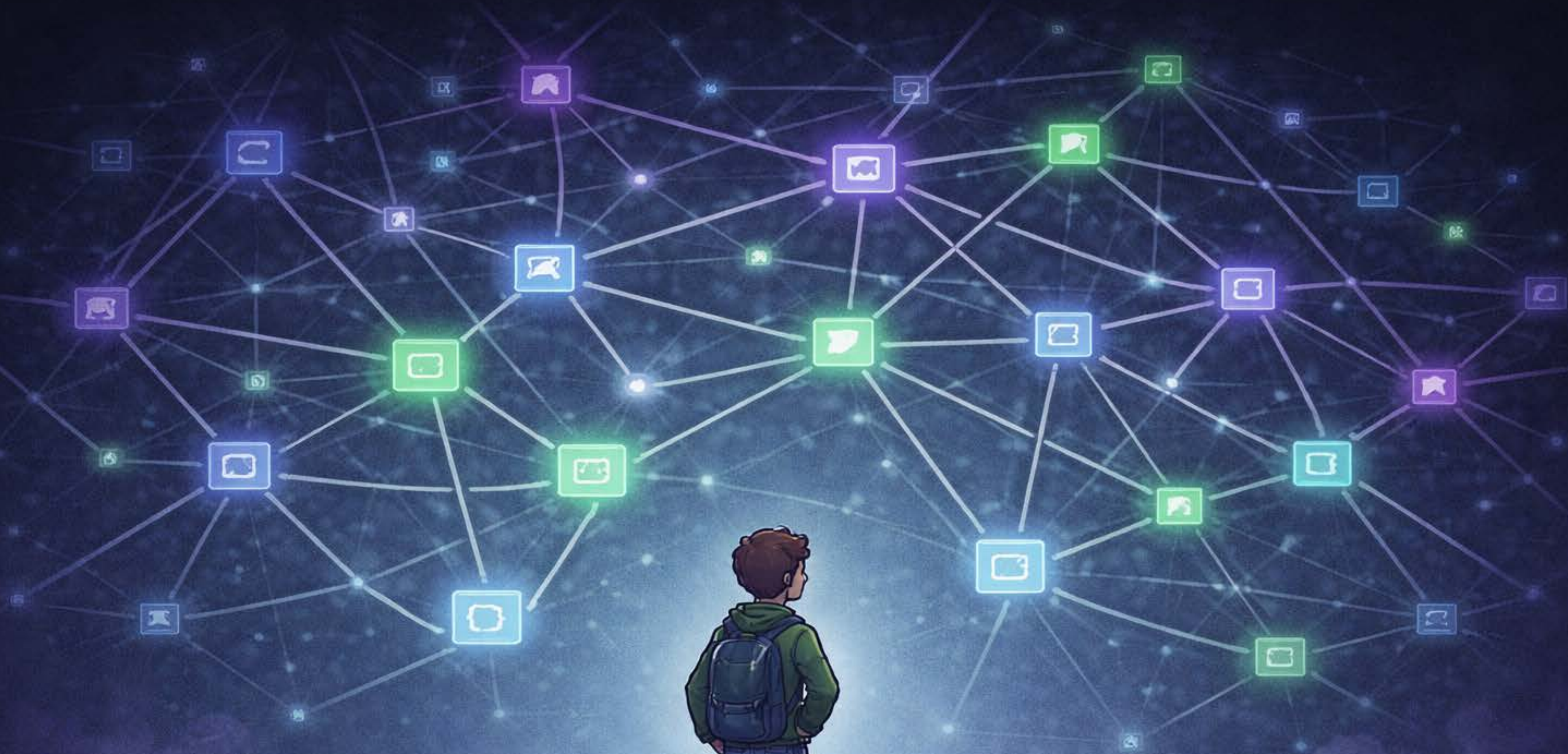


**Proven once.  
Now scale without breaking.**

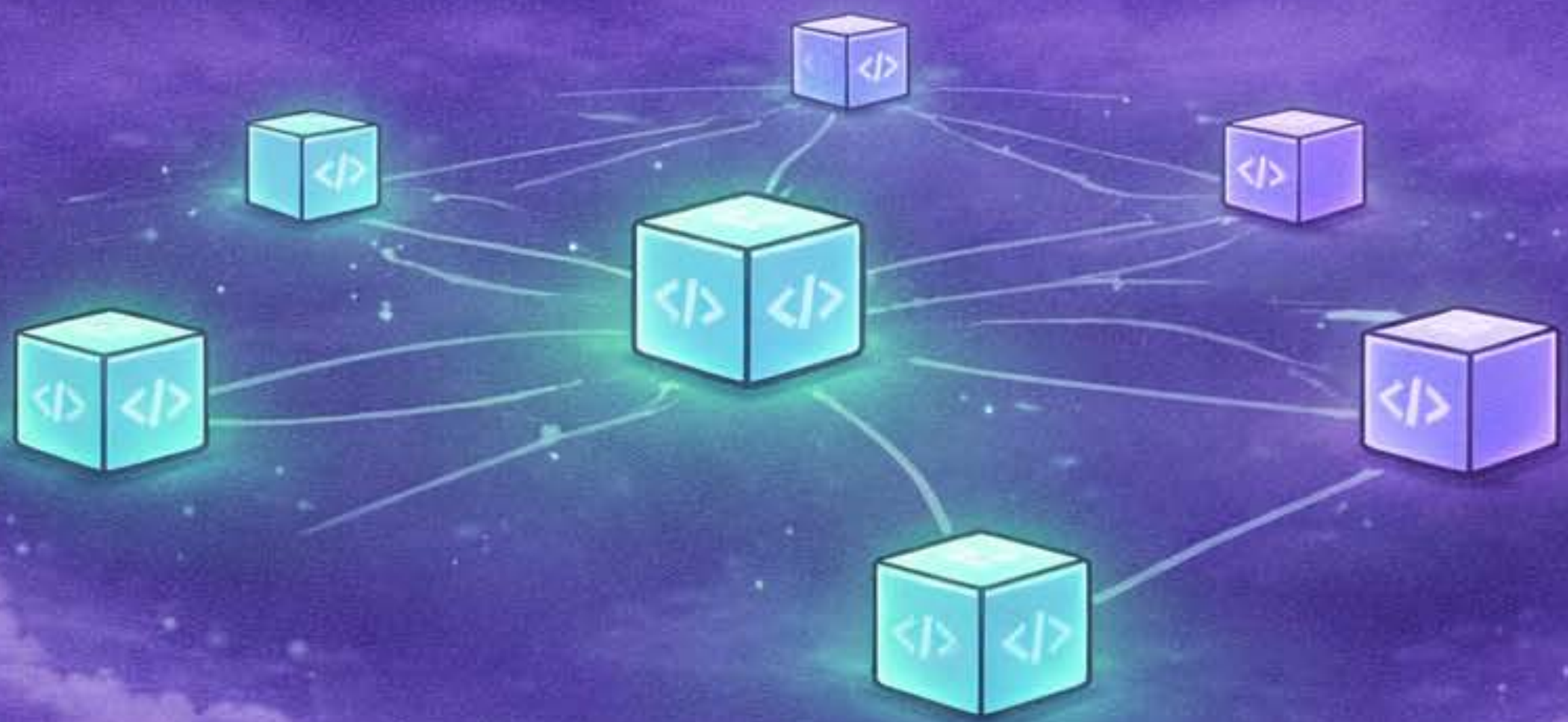


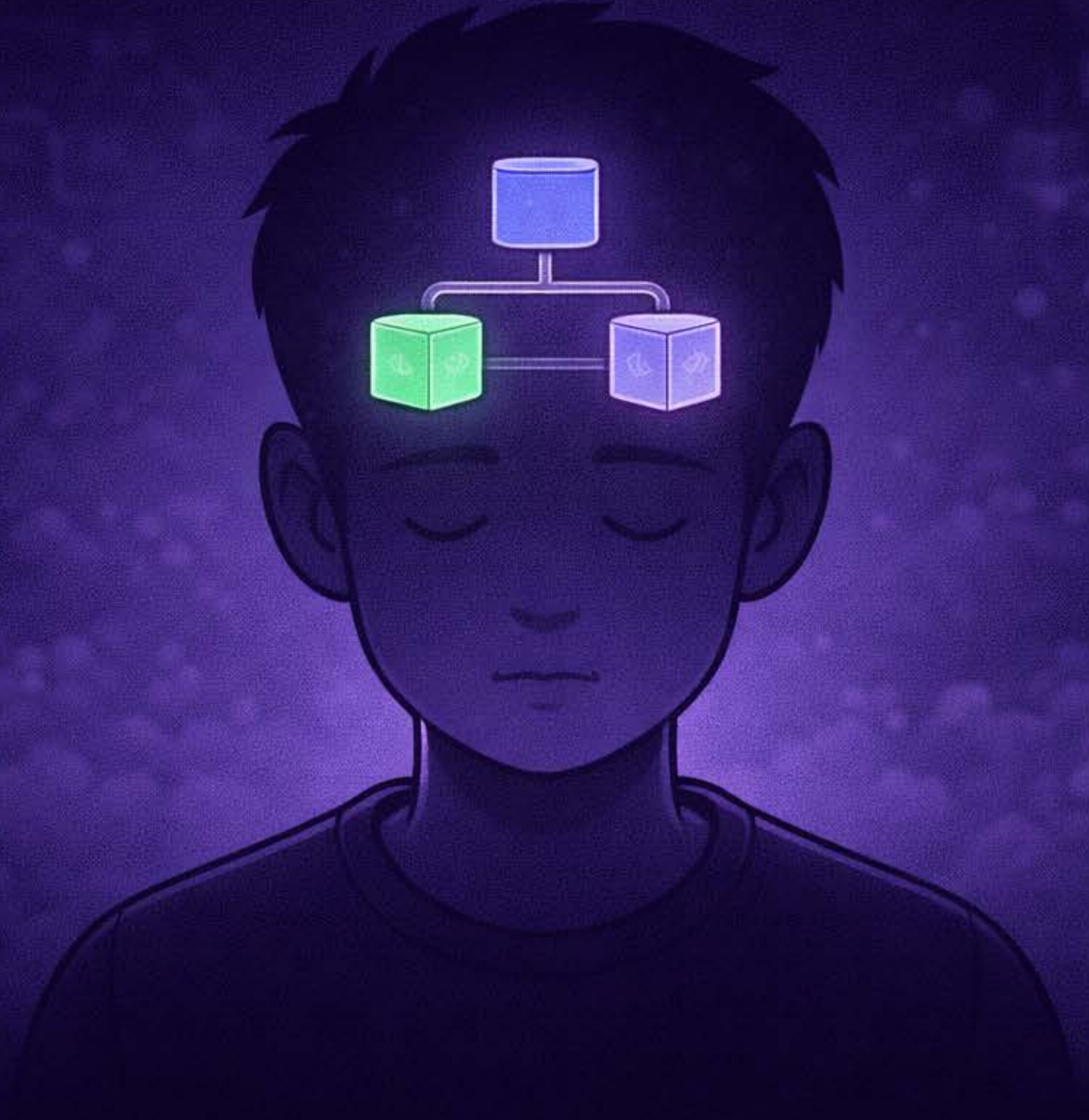
**Architecture is an interface for  
humans**

**The system worked.  
Changing it became the problem.**









# Let's stay in touch

## Fleet

- [fleetdm.com](https://fleetdm.com)
- [github.com/fleetdm/fleet](https://github.com/fleetdm/fleet)
- We're hiring: [fleetdm.com/jobs](https://fleetdm.com/jobs)

## Victor Lyuboslavsky, Principal Software Engineer

- LinkedIn: [linkedin.com/in/lyuboslavsky](https://www.linkedin.com/in/lyuboslavsky)
- Blog: [victoronsoftware.com](https://victoronsoftware.com)
- YouTube: [youtube.com/@VictorOnSoftware](https://www.youtube.com/@VictorOnSoftware)

