

back to modularity

Daniel Jackson · MIT CSAIL · HYTRADBOI · 2025

how good are LLMs at generating code?

single functions

46%

of code is autogenerated
when Copilot enabled
(June 2023)

92%

accuracy of coding
GPT-4 + AgentCoder on MBPP
(Jan 2024)

whole apps

If you ask GPT-4 to write whole apps for you, it usually can't get there. But if you architect the app in your head, you can ask it to build it out one function at a time pretty effectively.

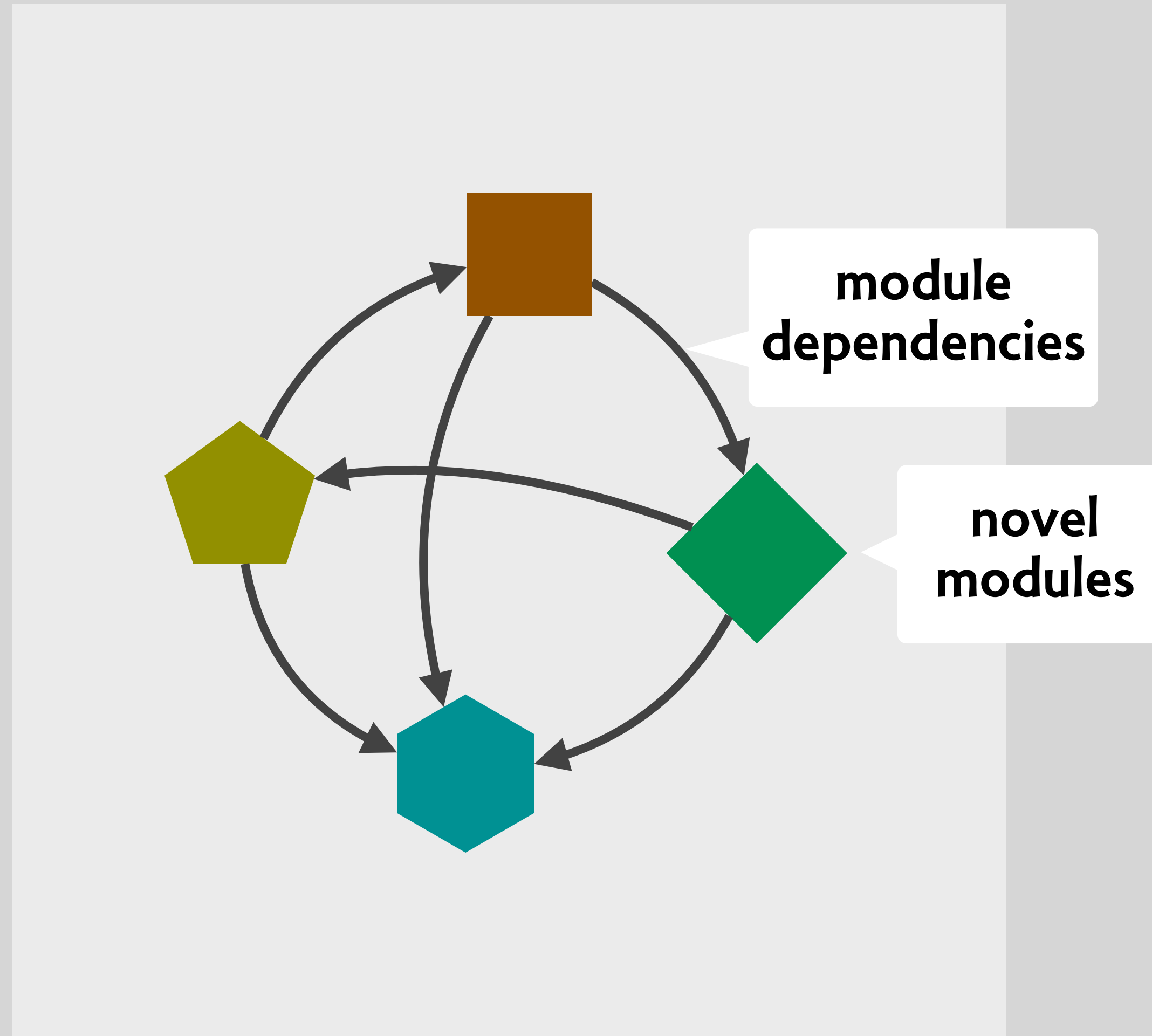
HackerNews, May 2024

The skills of the developer will be to figure out, "How small do I have to go until I can leverage AI to synthesize that code for me?"

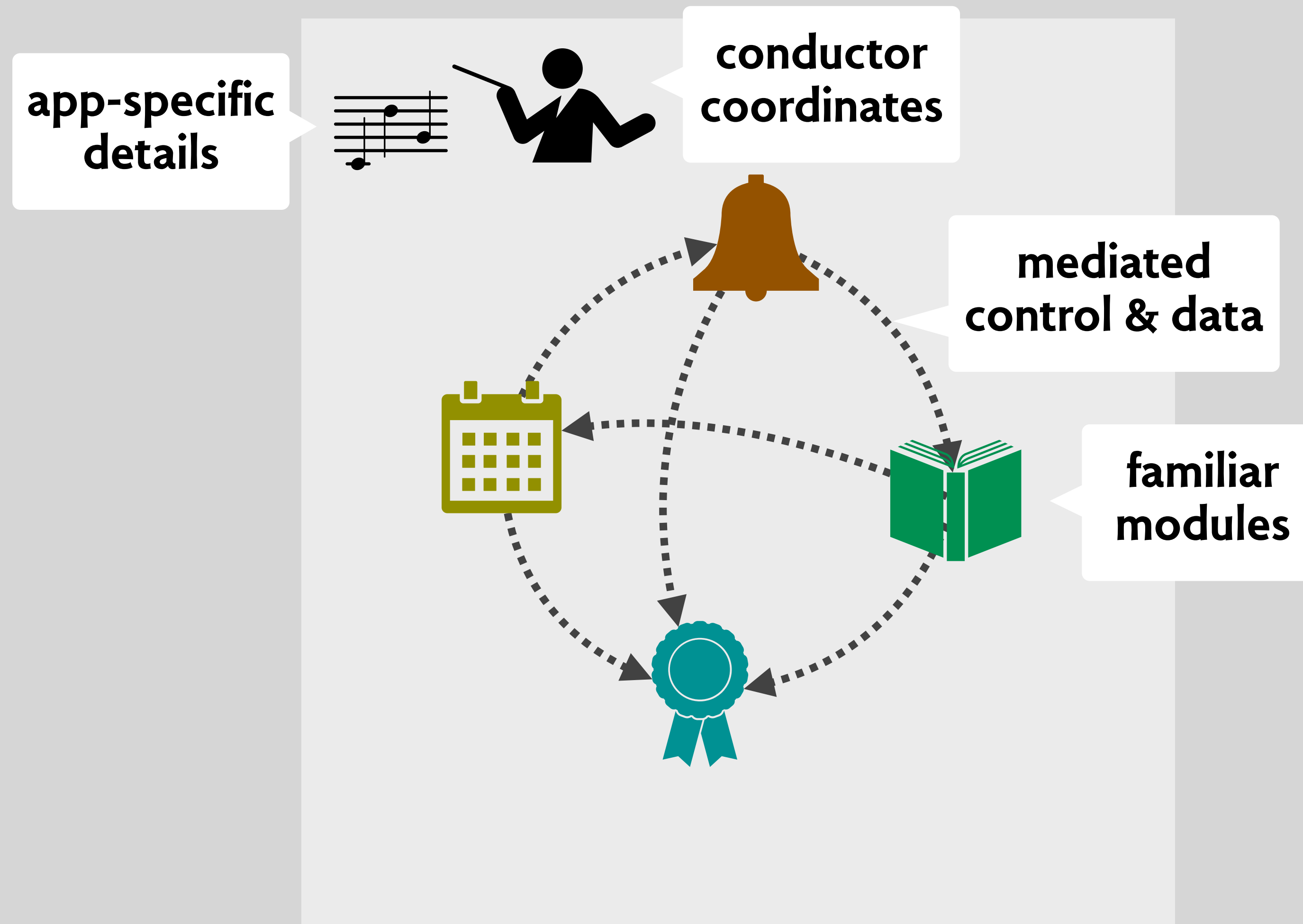
Thomas Dohmke, GitHub CEO, June 2023

it's time to
rethink how software
is **structured**

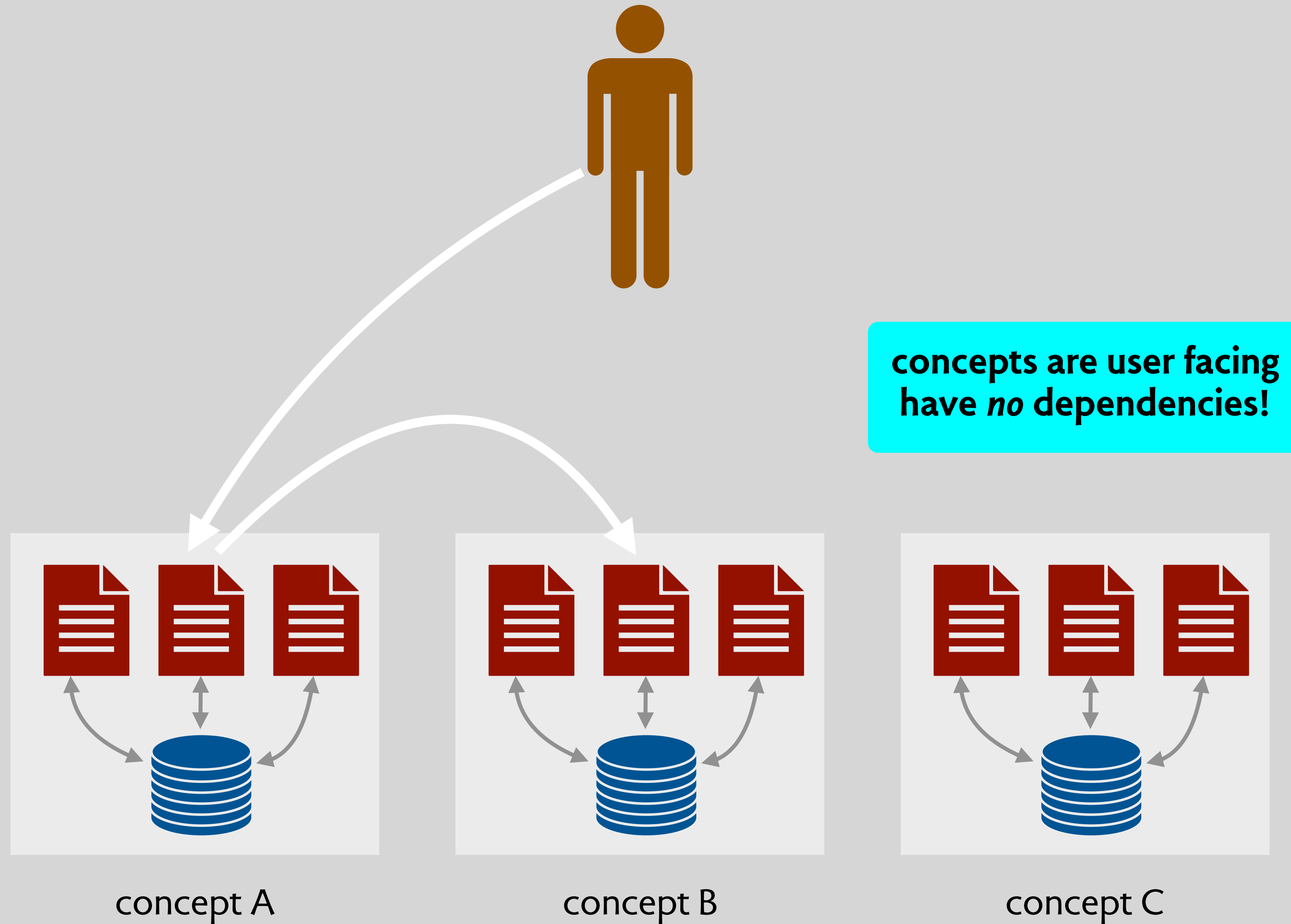
building an app one module at a time



a different approach: true modularity



viewing a system in terms of concepts



designing
hacker news

most apps are made from familiar functions

Y

Hacker News

new | past | comments | ask | show | jobs | submit

login

▲ Jackson structured programming (wikipedia.org)

106 points by haakonhr 63 days ago | hide | past | favorite | 69 comments

Post

Session

Upvote

Favorite

user: danielnicholas

created: 63 days ago

karma: 11

Profile

Comment

you might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift

I'd point to these ideas as worth knowing:

...ing problem that involves traversing ... structures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized it.

- The archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them
- Coroutines (or code transformation) let you structure code more cleanly when you need to read or write more than one structure. It's why real iterators (with yield), which offer a limited form of this, are (in my view) better than Java-style iterators with a next method.
- The idea of viewing a system as a collection of asynchronous processes (Ch. 11 in the JSP book, which later became JSD) with a long-running process for each real-world entity. This was a notable contrast to OOP, and led to a strategy (seeing a resurgence with event storming for DDD) that began with events rather than objects.

[0] <https://groups.csail.mit.edu/sdg/pubs/2009/hoare-jsp-3-29-09...>

▲ ob-nix 63 days ago [-]

... this brings back memories! In the late eighties I, as a teenager, found a Jackson Struct. Pr. book at the town library. I remember I was amazed at the text and wondered why I hadn't heard about the method before.

If I remember correctly did the book clearly point out backtracking as a standard method, while mentioning that most languages lacked that, so it had to be implemented manually.

... with some creative variation

▲ Jackson structured programming (wikipedia.org)

106 points by haakonhr 63 days ago | hide | past | favorite | 69 comments

▲ danielnicholas 63 days ago [-]

If you want an intro to JSP, you can find it in 2009.

For those who don't know JSP, I

- There's a class of programming languages that are statically typed but bases code structure only on types.

- There are some archetypal programming languages that help with them helps.

- Coroutines (or code transformation) and iterators (with yield), which offer a way to structure code.

- The idea of viewing a system as a collection of events for each real-world entity. This is a way of thinking about events rather than objects.

[0] <https://groups.csail.mit.edu/robotics-1/www/FALL/robotics1/lectures/01/01-02.html>

▲ ob-nix 63 days ago [-]

... this brings back memories! In the late eighties I, as a teenager, found a Jackson Struct. Pr. book at the town library. I remember I was amazed at the text and wondered why I hadn't heard about the method before.

If I remember correctly did the book clearly point out backtracking as a standard method, while mentioning that most languages lacked that, so it had to be implemented manually.

“combinational creativity” [Boden]

familiar elements combined in new ways

for HackerNews, things like

a post has a title and either just a link, or just a question
no comments on a post after 2 weeks, no edits after 2 hours
can't downvote a comment until your own post upvoted

...

a sample concept: upvote

concept Upvote

purpose rank items by popularity

state

by: Vote -> **one** User

for: Vote -> **one** Item

Upvote, Downvote: **set** Vote

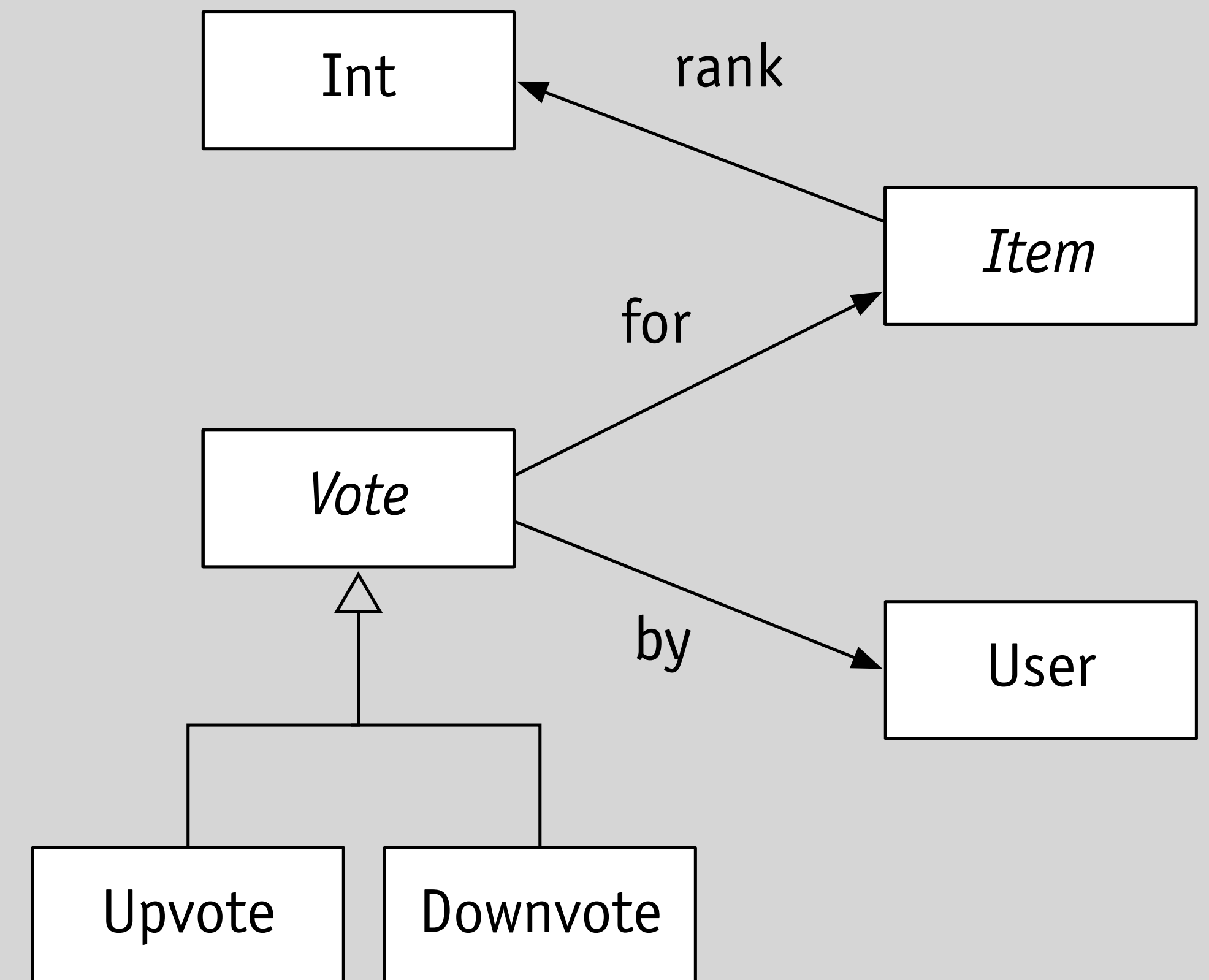
rank: Item -> **one** Int

actions

upvote (u: User, i: Item)

downvote (u: User, i: Item)

unvote (u: User, i: Item)



how to add app-specific functionality?

concept Upvote

purpose rank items by popularity

actions

upvote (u: User, i: Item)

downvote (u: User, i: Item)

unvote (u: User, i: Item)

suppose I want this behavior:

you can't downvote an item
until you've received
an upvote on your own post

define a new concept!

a hint: not just used by Upvote

concept Karma

purpose privilege good users

state

karma: User -> one Int

actions

reward (u: User, r: Int)

permit (u: User, r: Int)

concept Post

purpose share content

state

author: Post -> one User

body: Post -> one Text

actions

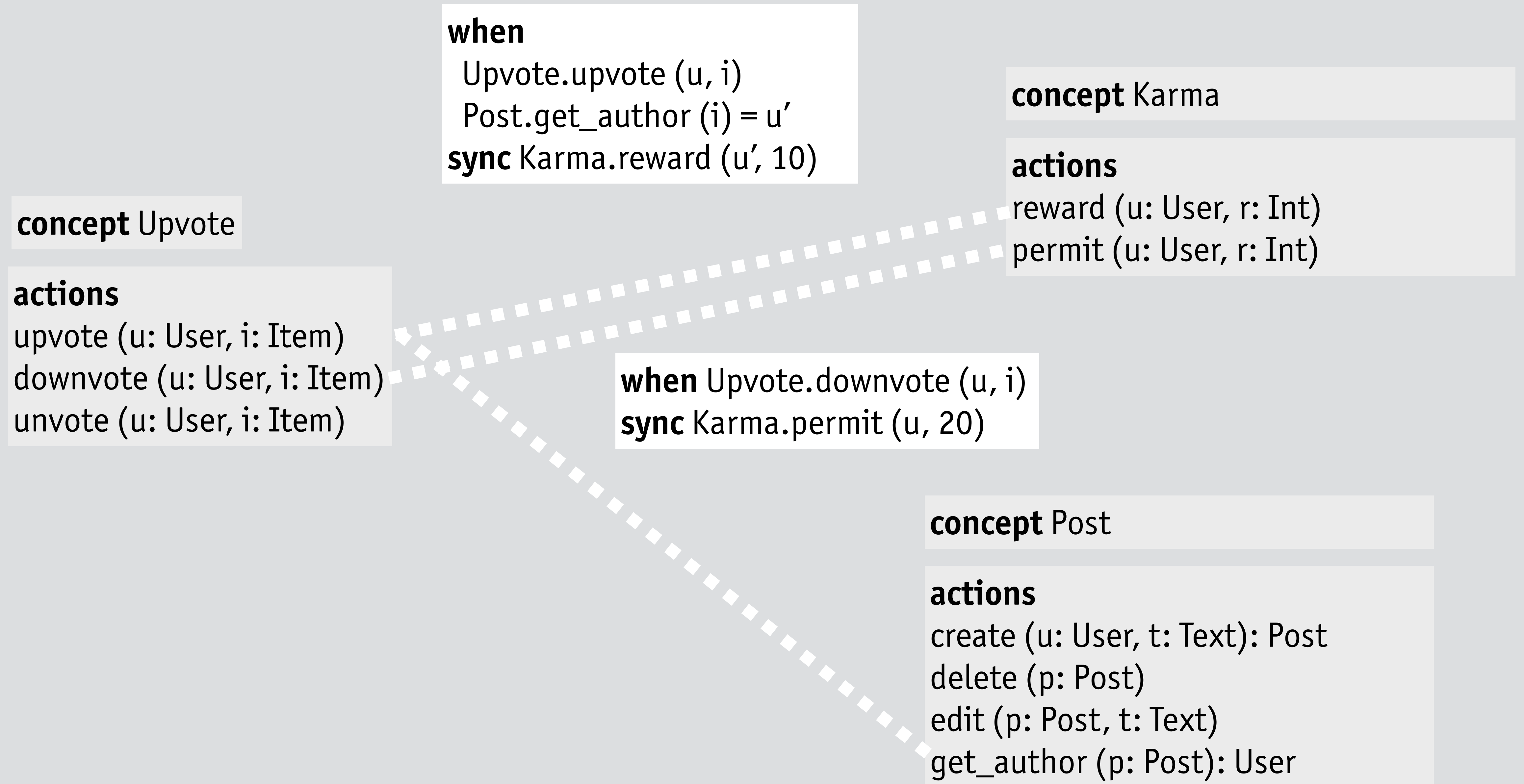
create (u: User, t: Text): Post

delete (p: Post)

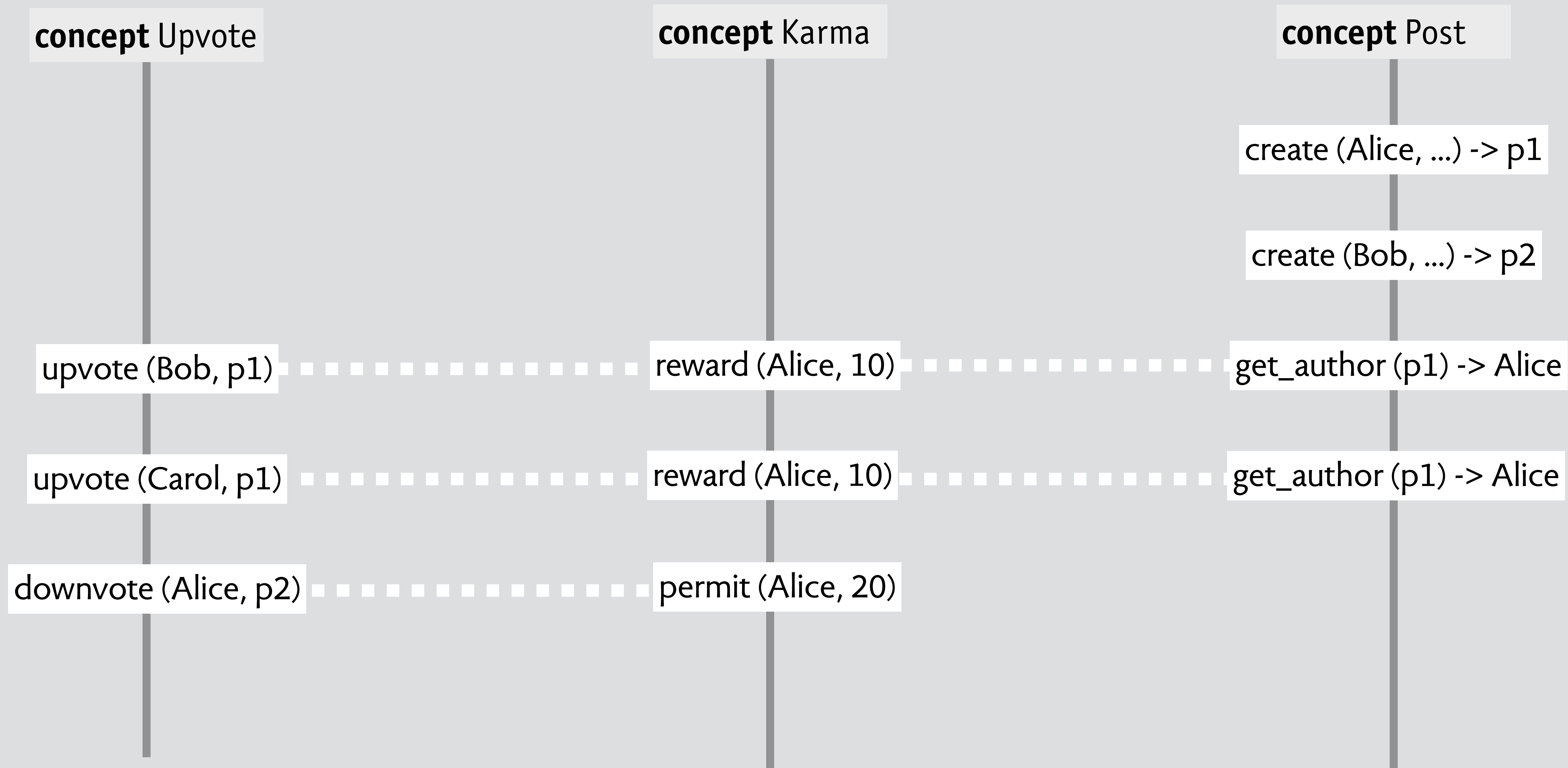
edit (p: Post, t: Text)

get_author (p: Post): User

compose concepts by action synchronization



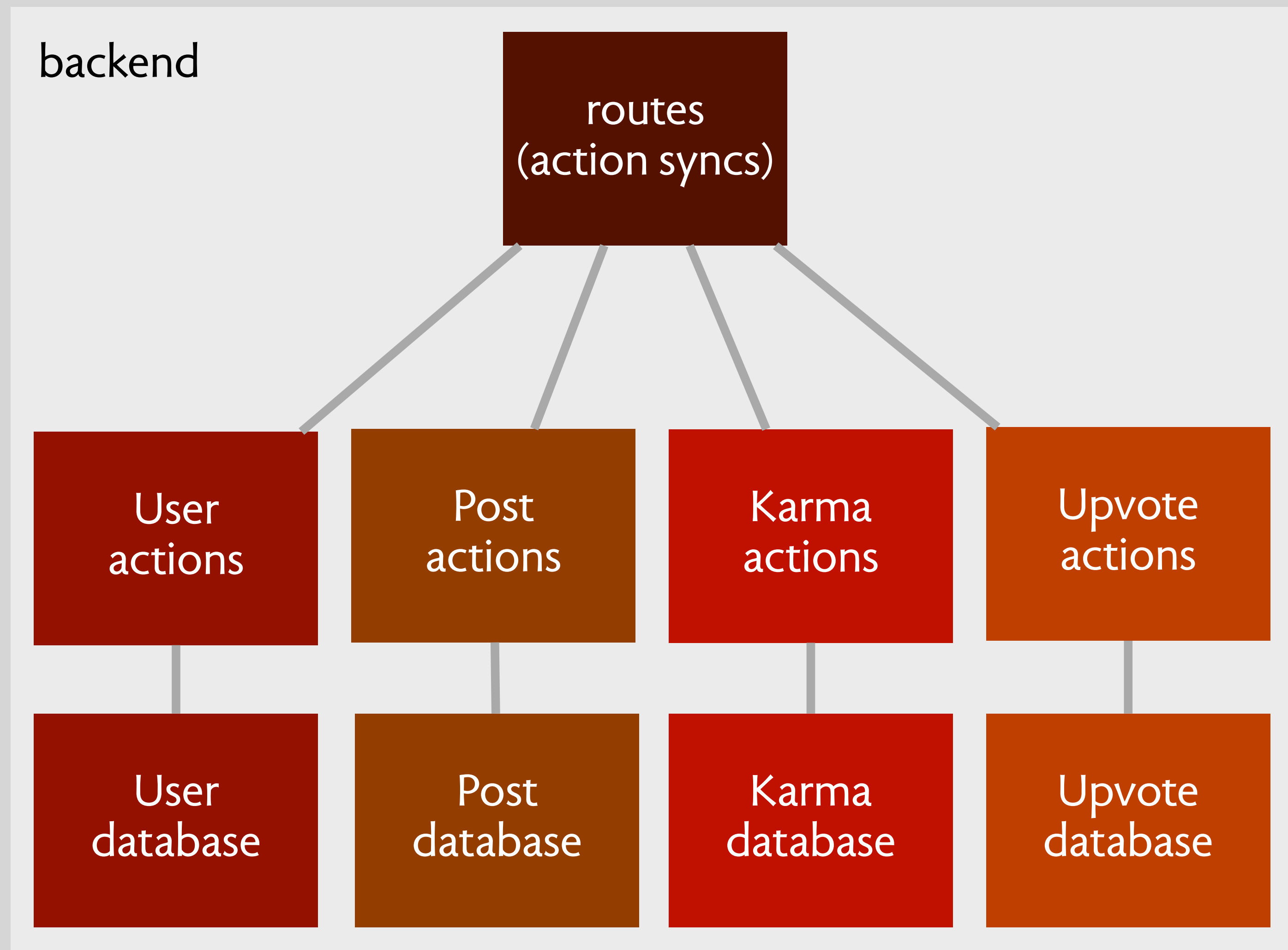
synchronizing concepts



coding hacker news

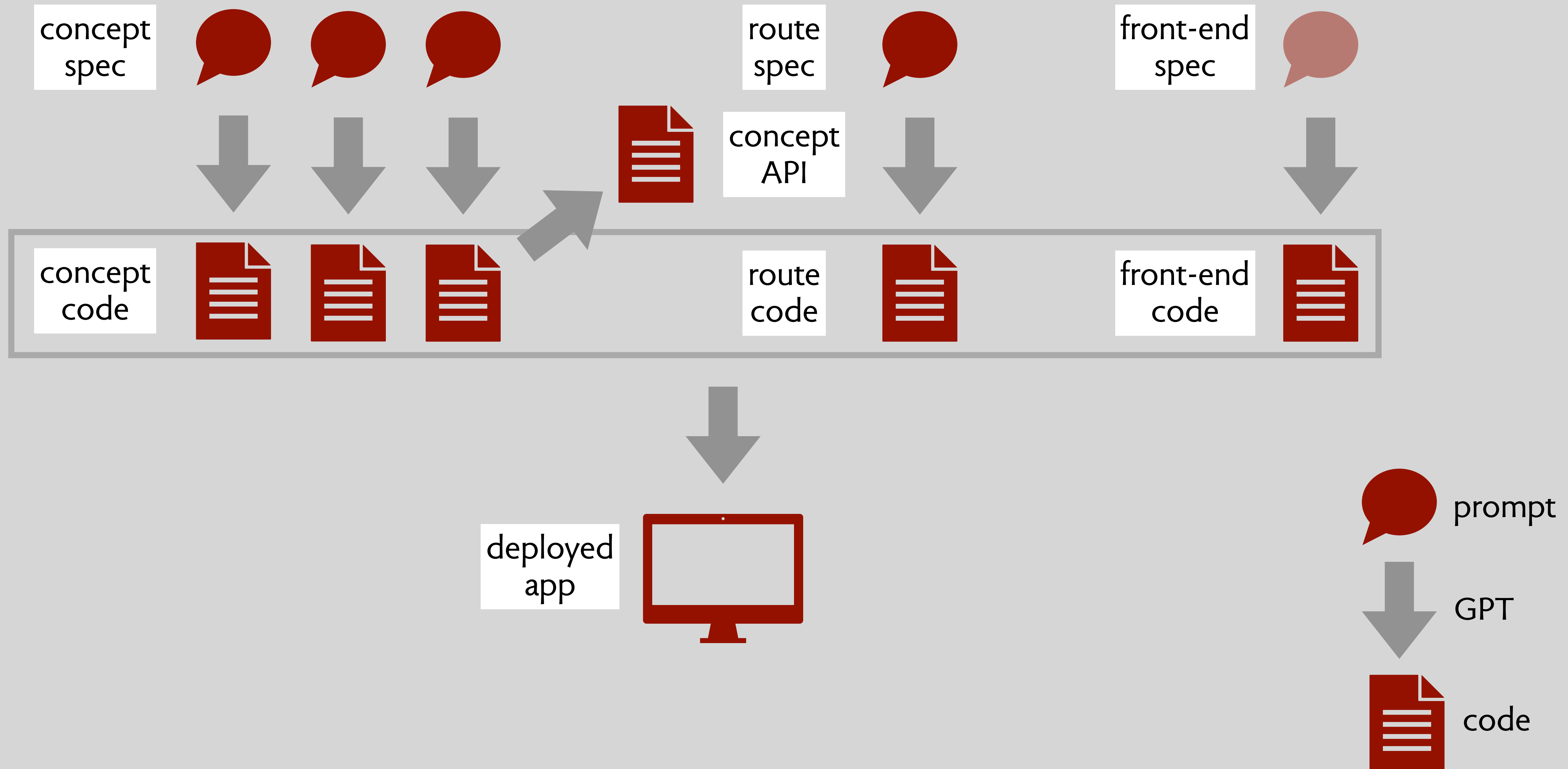
joint work with Abutalib Namazov

a new way to structure a web app



no
dependencies
between
concepts!

now we can generate code!



creating a concept



Create a new concept

☐ Bulk create

Concept name

Karma

Description

Each karma should have user (generic) and points (number). Actions should be: increase(user, x), decrease(user, x), get(user), isAllowed(user, threshold) (throws error if user doesn't have at least threshold)

Create

**brief &
application
independent**

first kodless generates the concept code

Source code:

```
1 import { ObjectId } from "mongodb";
2 import DocCollection, { BaseDoc } from "../framework/doc";
3 import { NotAllowedError } from "../framework/errors";
4
5 interface KarmaDoc extends BaseDoc {
6   user: ObjectId;
7   points: number;
8 }
9
10 export default class KarmaConcept {
11   public readonly karmas: DocCollection<KarmaDoc>;
12
13   constructor(collectionName: string) {
14     this.karmas = new DocCollection<KarmaDoc>(collectionName);
15   }
16
17   async increase(user: ObjectId, x: number) {
18     const karma = await this.karmas.readOne({ user });
19     if (karma) {
20       await this.karmas.collection.updateOne({ user }, { $inc: { points: x } });
21     } else {
22       await this.karmas.createOne({ user, points: x });
23     }
24   }
25 }
```

**database
schema
etc**

**typical
coding
details**

then it generates an API spec from the code

Generated spec:

```
KarmaDoc = {  
  user: ObjectId;  
  points: number;  
}  
  
KarmaConcept = {  
  constructor: (collectionName: string) => KarmaConcept; // instantiates a new KarmaConcept with the  
given collection name  
  async increase: (user: ObjectId, x: number) => void; // increases the karma points for a user; cre  
ates a new karma record if not exists  
  async decrease: (user: ObjectId, x: number) => void; // decreases the karma points for a user; thr  
ows NotAllowedError if resulting in negative karma  
  async get: (user: ObjectId) => number; // returns the karma points for a user; returns 0 if the us  
er has no karma record  
  async isAllowed: (user: ObjectId, threshold: number) => void; // throws NotAllowedError if user's  
karma points do not meet the threshold  
}
```

**spec is more
detailed
than prompt**

creating a route



Create a new route

☐ Bulk create

Route description

POST /register – register and login. New users start with 1 karma.

Create

**no explicit mention
of concepts or their
actions**

kodless generates the route code

Delete

```
1 @Router.post("/register")
2   async createUser(session: WebSessionDoc, username: string, password: string) {
3     WebSession.assertLoggedOut(session);
4     const { msg, user } = await User.create(username, password);
5     await Karma.increase(user._id, 1);
6     WebSession.start(session, user._id);
7     return { msg };
8   }
```

**Kodless
uses concept APIs
to select actions**

hacked news: home page



Hacked News

[new](#) | [past](#) | [comments](#) | [ask](#) | [show](#) | [jobs](#) | [submit](#)

[login](#)

1. ▲ [Show HN: Kodless -- make websites with AI \(github.com\)](#)
1 point by dnj a day ago | [hide](#) | [discuss](#)
2. ▲ [Monolith -- CLI tool for saving web pages as a single HTML file \(crates.io\)](#)
1 point by rust 9 days ago | [hide](#) | [discuss](#)
3. ▲ [Show HN: Kodless -- make websites with AI \(github.com\)](#)
3 points by kodless 9 days ago | [hide](#) | [2 comments](#)
4. ▲ [Software = concepts \(essenceofsoftware.com\)](#)
6 points by dnj 9 days ago | [hide](#) | [2 comments](#)
5. ▲ [Jobs HN: I am hiring a wine expert](#)
1 point by recruiter 10 days ago | [hide](#) | [discuss](#)
6. ▲ [Show HN: Fuiz -- free, open-source and privacy-friendly alternative to Kahoot \(fuiz.us\)](#)
3 points by best_dev 10 days ago | [hide](#) | [7 comments](#)
7. ▲ [Ask HN: How many bugs do you have per line of code?](#)
1 point by big_asker 10 days ago | [hide](#) | [discuss](#)
8. ▲ [Github -- Use this website to share your code \(github.com\)](#)
2 points by barish2 10 days ago | [hide](#) | [discuss](#)

[Guidelines](#) | [FAQ](#) | [Lists](#) | [API](#) | [Security](#) | [Legal](#) | [Apply to YC](#) | [Contact](#)

Search:

hacked news: thread

 **Hacked News** [new](#) | [threads](#) | [past](#) | [comments](#) | [ask](#) | [show](#) | [jobs](#) | [submit](#)

[kodless \(3\)](#) | [logout](#)

Software = concepts (essenceofsoftware.com)

6 points by dnj 9 days ago | [unvote](#) | [hide](#) | [favorite](#) | [2 comments](#)

add comment

* [kodless](#) 9 days ago | [next \[-\]](#)

dnj, you should check out my platform Kodless -- it helps you generate software with concepts without writing any code.

[reply](#)

dnj 9 days ago | [next \[-\]](#) [unvote](#)

That's great to see! Have you seen GPT-powered concept tutor? <https://essenceofsoftware.com/studies/larger/tutor/>

[reply](#)

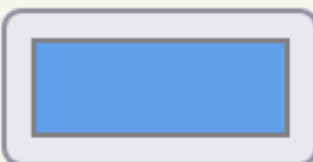
[Guidelines](#) | [FAQ](#) | [Lists](#) | [API](#) | [Security](#) | [Legal](#) | [Apply to YC](#) | [Contact](#)

Search:

hacked news: user profile

user: kodless
created: 2024-03-25T19:14:36.316Z
karma: 3
about:

- [submissions](#)
- [threads](#)
- [favorites](#)
- [hidden](#)



change top bar color

Search:

hacked news: posts by date

[previous day](#) | [next day](#) | 2024-03-25

- 1. [Software = concepts](#) (essenceofsoftware.com)
6 points by dnj 9 days ago | [unvote](#) | [hide](#) | 2 comments
- 2. [★ Show HN: Kodless -- make websites with AI](#) (github.com)
3 points by kodless 9 days ago | [hide](#) | 2 comments
- 3. [Show HN: Fuiz -- free, open-source and privacy-friendly alternative to Kahoot](#) (fuiz.us)
3 points by best_dev 10 days ago | [unvote](#) | [hide](#) | 7 comments
- 4. [Github -- Use this website to share your code](#) (github.com)
2 points by barish2 10 days ago | [unvote](#) | [hide](#) | [discuss](#)
- 5. [▲ Ask HN: How many bugs do you have per line of code?](#)
1 point by big_asker 10 days ago | [hide](#) | [discuss](#)

Search:

successes & challenges

coverage of all major functionality

flagging, hiding, karma, etc
nested comments, complicated ranking rules
many application-specific details (eg, karma)

minimal prompting required

< 1,000 words in total in prompts

no code editing needed

some prompt revisions to tweak behavior

frontend code not yet fully automated

but have designed HTMx like DSL

need actions to query concept state

eg, `Post.getByIds` to simplify route code

implementation details in a few route specs

eg, prompt for GET/session says "return currently logged in user with karma in same object"

conclusions

where we are

a DSL for syncs (Eagon Meng)

removes all the magic in our current framework

clean separation of core logic from presentation

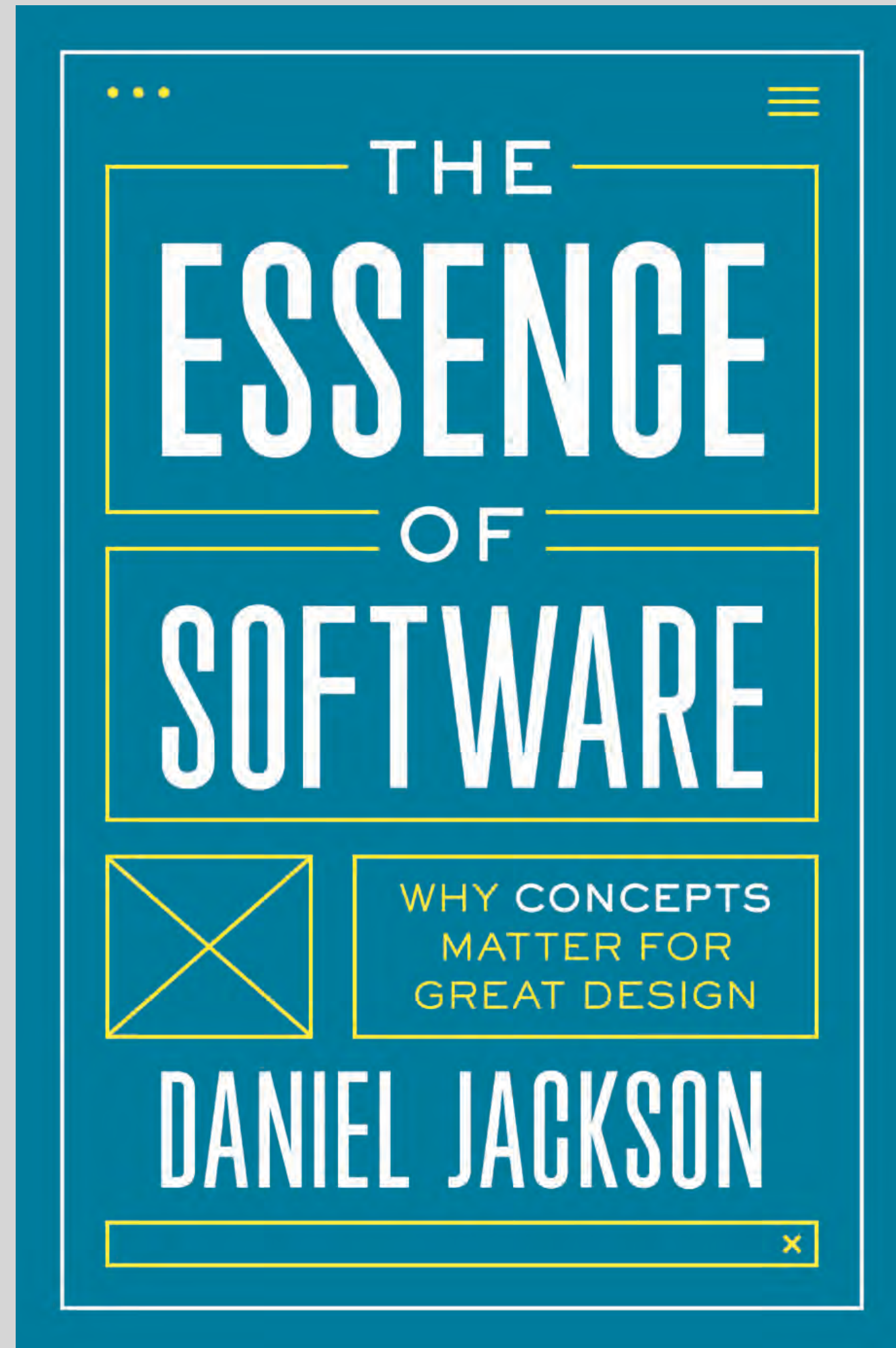
factor our persistence layer, write concepts over plain JS

database challenges

concepts are polymorphic, but can't express in schemas

syncing doesn't allow joins across concept schemas

for more about concept design



<https://essenceofsoftware.com>

<https://essenceofsoftware.com>

blog posts

tutorials

forum

newsletter