

synchronization

coordinating concepts

Daniel Jackson & Eagon Meng · MIT CSAIL/EECS · April 2026

synchronization
keeping concepts
independent

how to coordinate concepts?

suppose we want this behavior

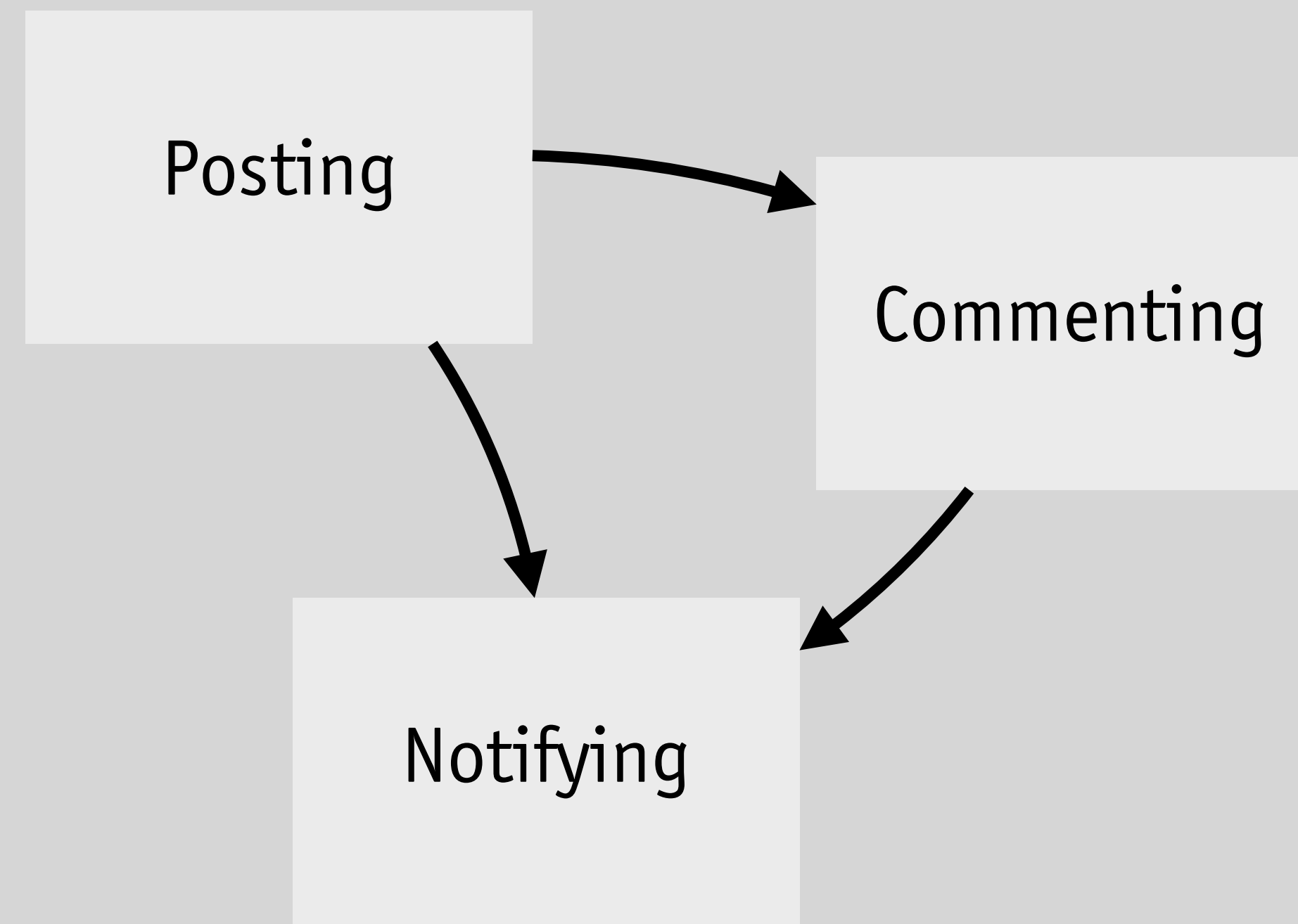
when a post is deleted,
(1) notify the author
(2) delete its comments
(3) notify comment authors

conventional solution

embed calls within modules

problems this causes

modules are now coupled
can't be defined independently
app-specific details entangled
behavior scattered across code



a better solution: synchronizations

express directly as **syncs**

when Posting.deleted (p)
where Posting.author (p, u)
then Notifying.notify (u)

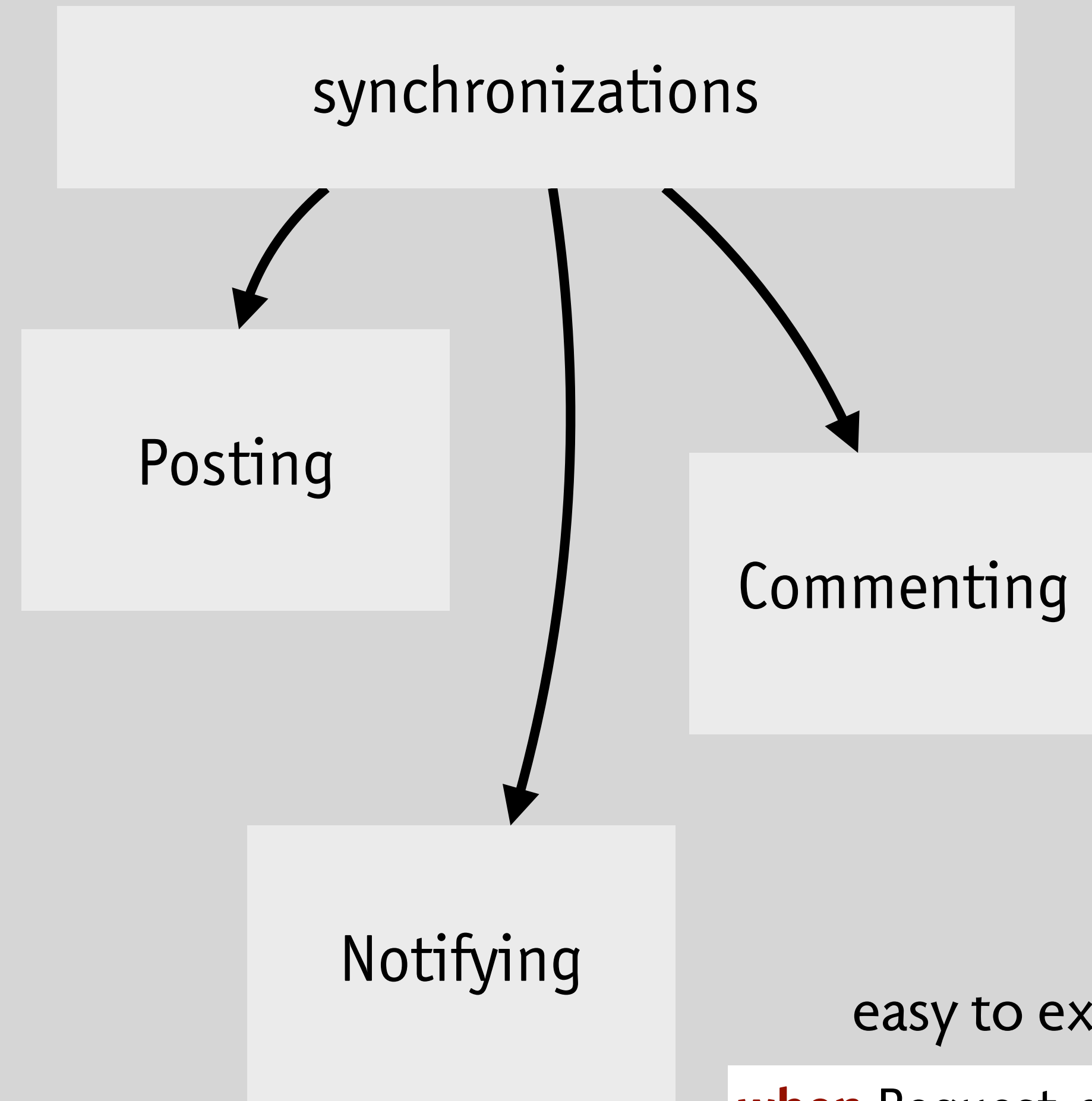
when Commenting.deleted (c)
where Commenting.author (c, u)
then Notifying.notify (u)

when Posting.deleted (p)
where Commenting.target (c, p)
then Commenting.delete (c)

happy consequences

concepts are independent!
app-specific details separated
interactions easily legible

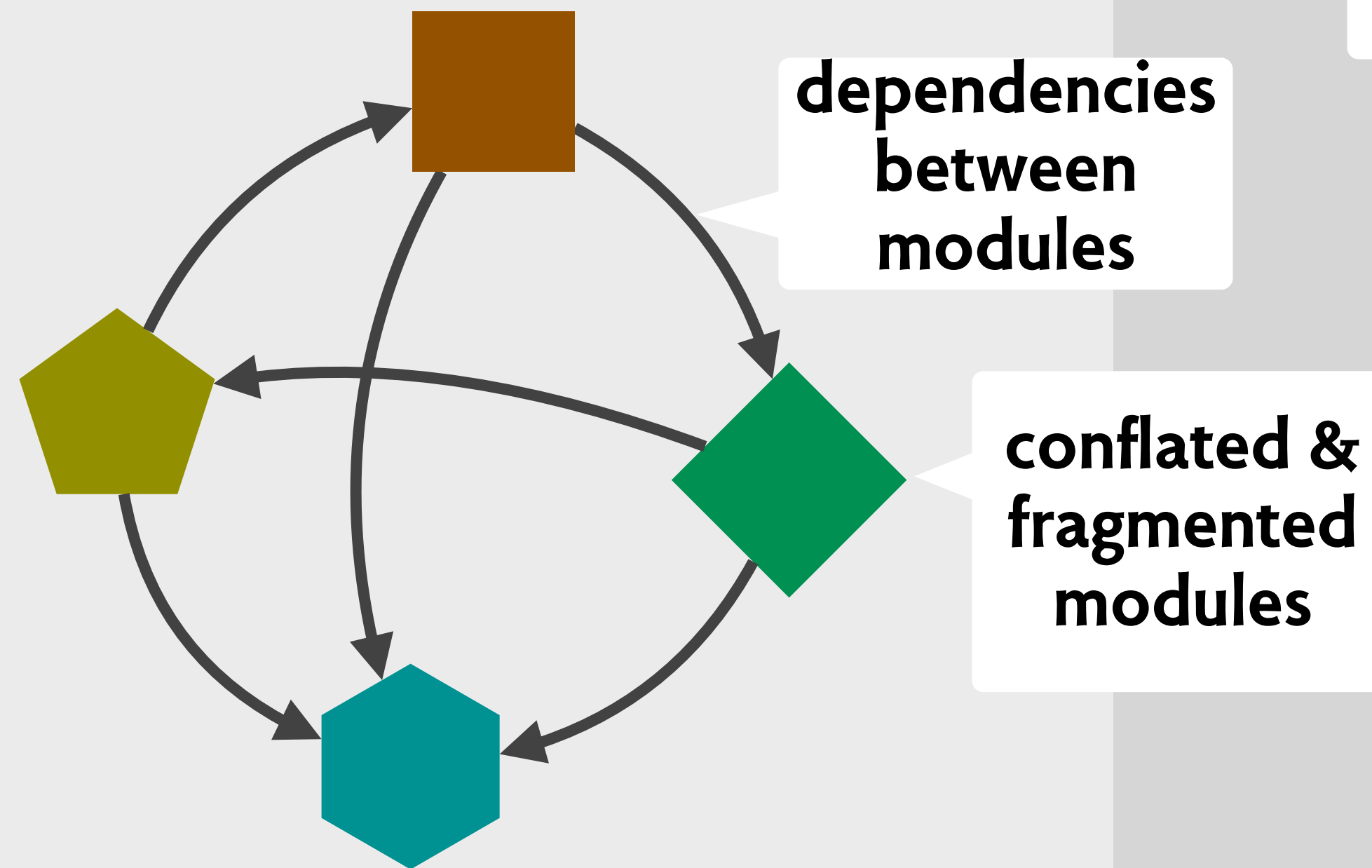
and easily log
actions and their
causal links



easy to express alternatives

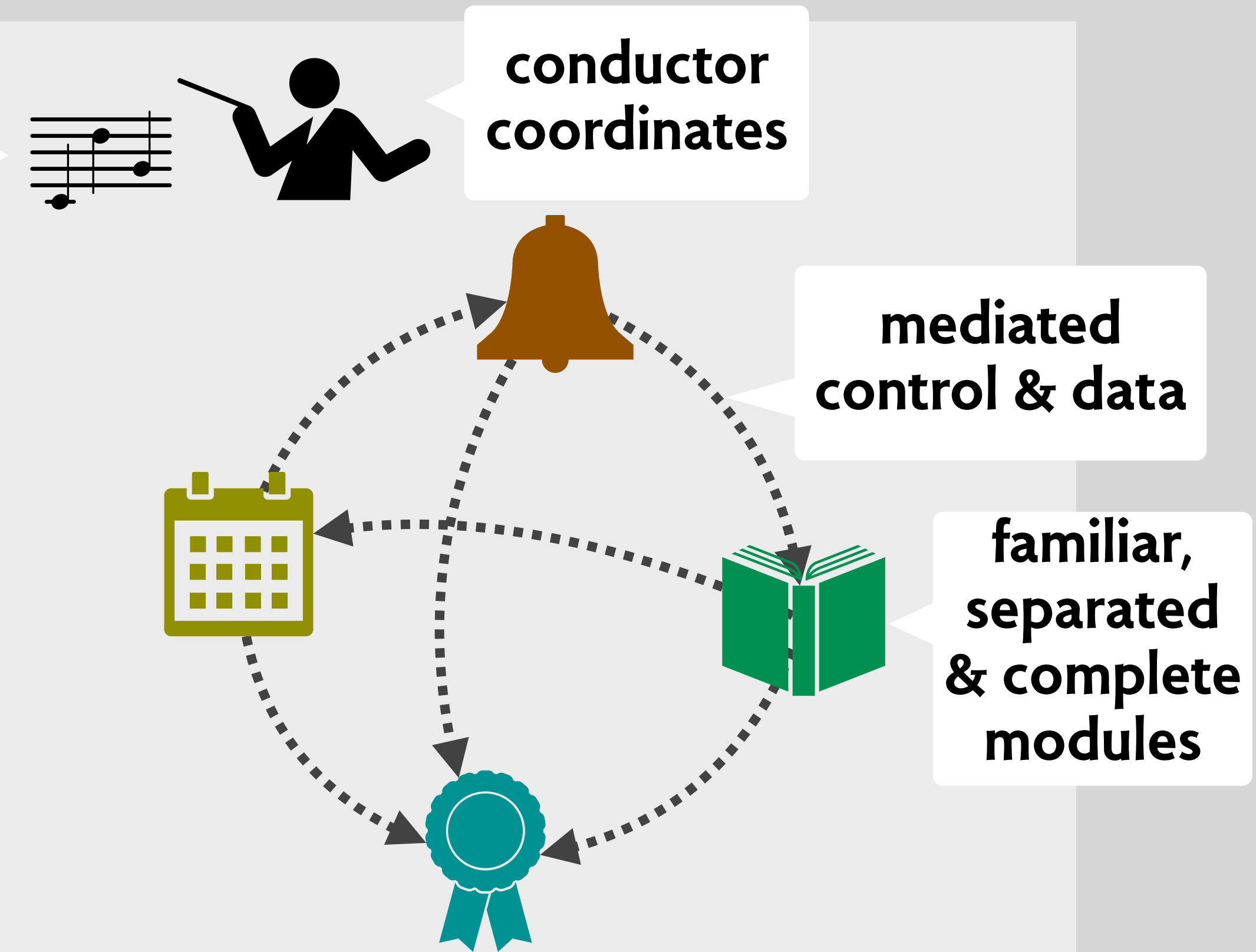
when Request.deletePost (p)
where Commenting: no comments on p
then Posting.delete (p)

an architectural view of concept composition



standard software development

app-specific details



concept-based software development

what are syncs used for?

maintaining invariants

no comments without post

getting inputs

author of new post is session user

authentication & access

only author can delete post

response hydration

get post with author image

profiling users

can't downvote until upvoted

achieve synergy

trash item when labeled deleted

logging & notification

notify on response to post

inferring intent

mark post as read on scroll

coordinating concepts

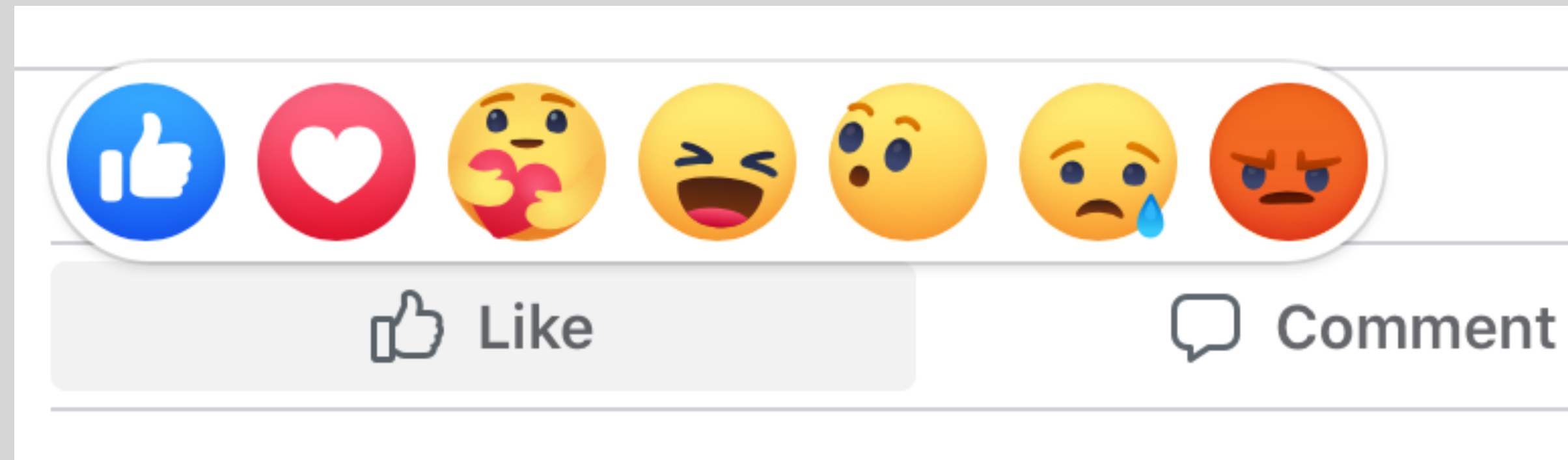
on joining, grant access

automation

follow user whose post liked

facebook
reactions

are angry reactions upvotes?



Facebook under fire

Five points for anger, one for a 'like': How Facebook's formula fostered rage and misinformation

Facebook engineers gave extra value to emoji reactions, including 'angry,' pushing more emotional and provocative content into users' news feeds

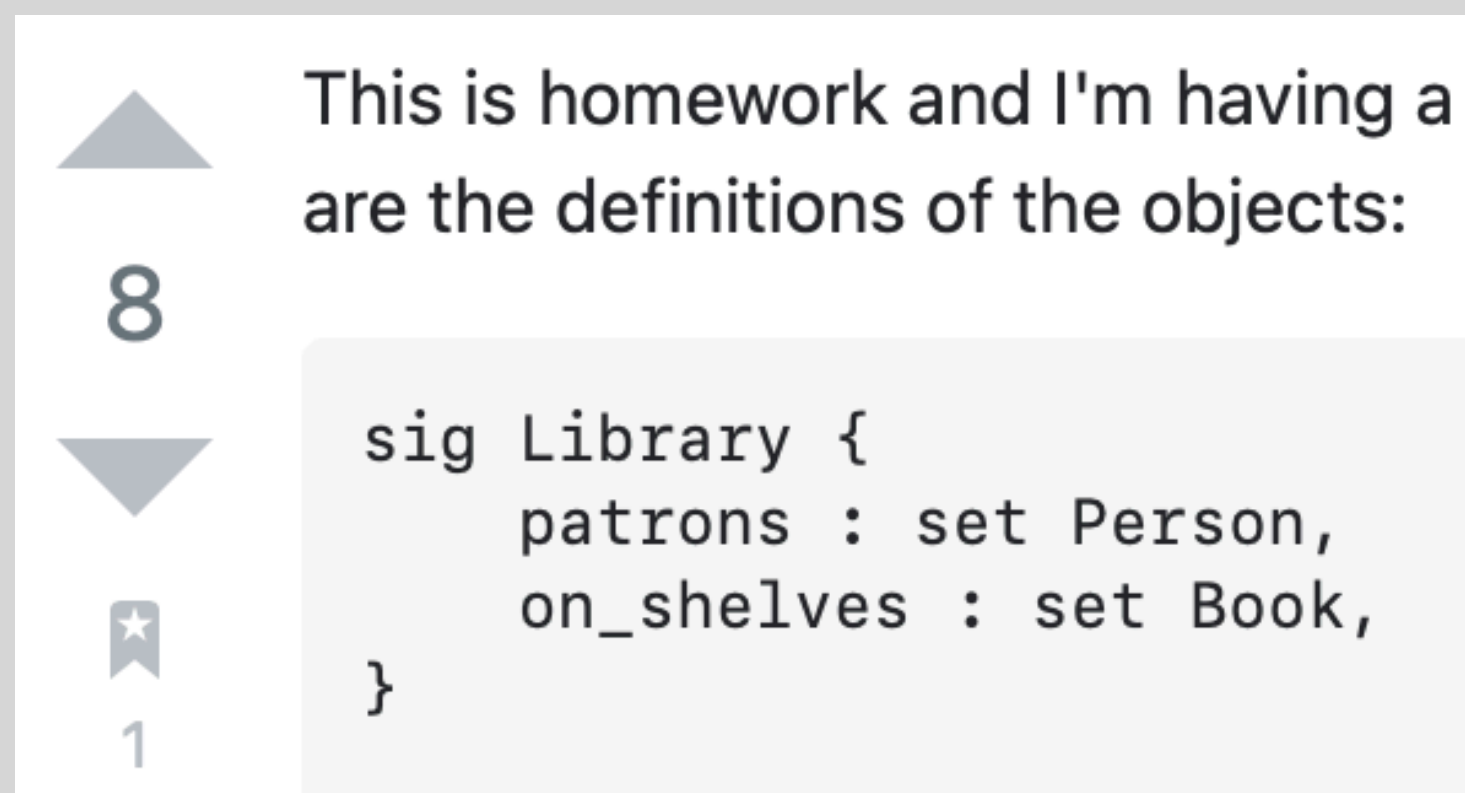
Updated October 26, 2021 More than 4 years ago

similar UIs, very different concepts

concept Upvote

purpose rank items by popularity

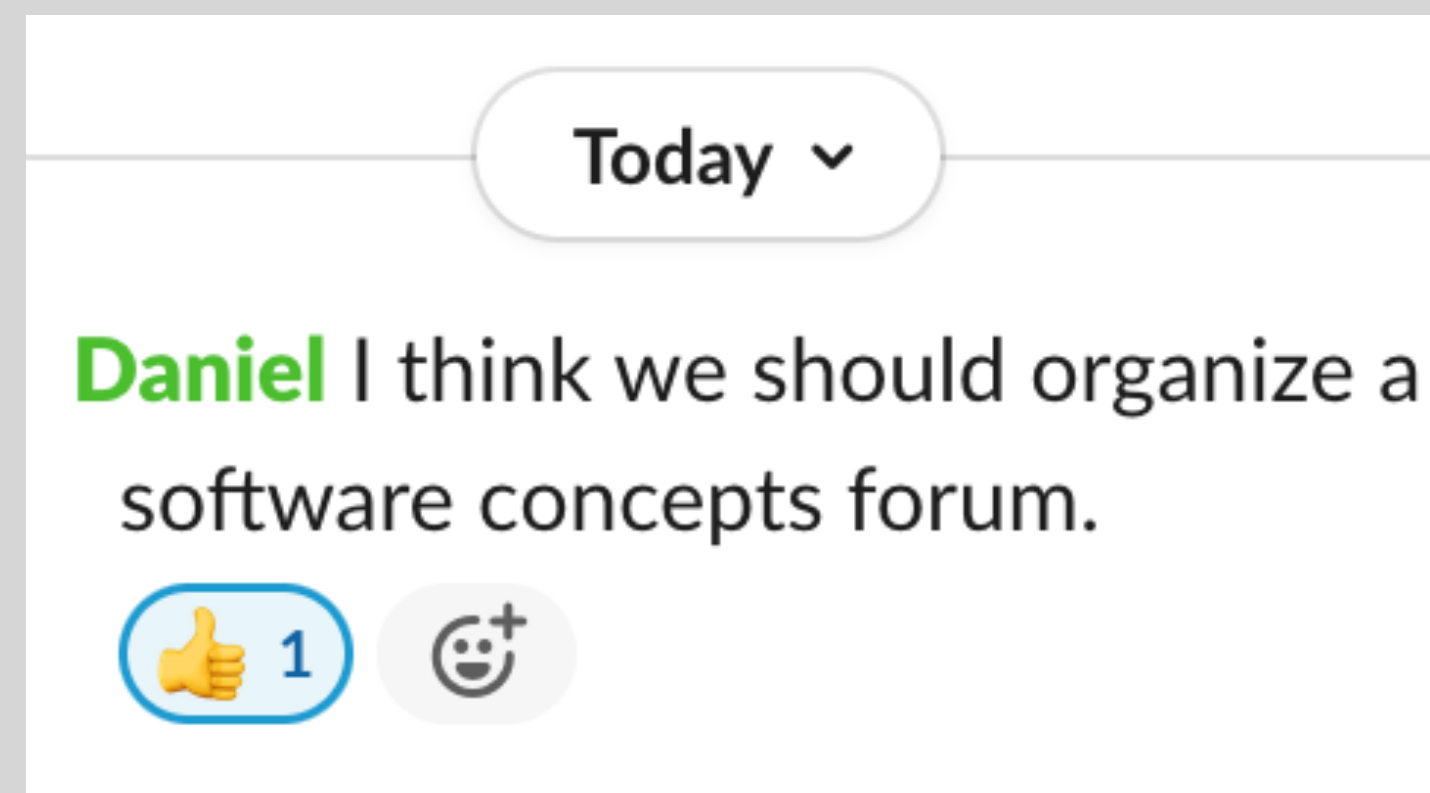
principle after series of upvotes of items, the items are ranked by their number of upvotes



concept Reaction

purpose send reactions to author

principle when user selects reaction, it's shown to the author (often in aggregated form)



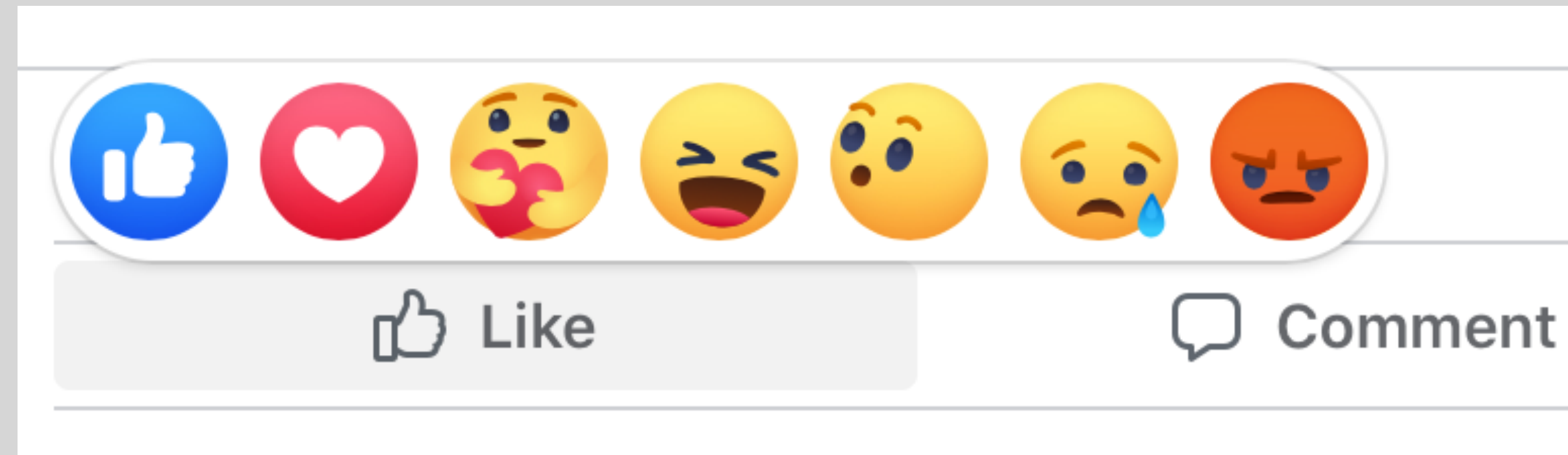
concept Recommendation

purpose use prior likes to recommend

principle user's likes lead to ranking of kinds of items, determining which items are recommended



a concept diagnosis



concept Upvote

purpose rank items by popularity

actions

upvote (u: User, i: Item)

...



concept Reaction

purpose convey emotion to author

actions

reactAngry (u: User, i: Item)

...

unwanted
sync?