

how to become a great software designer

Daniel Jackson, MIT CSAIL · Boston ACM/IEEE · March 24, 2022



BRENDAN HOFFMAN FOR THE NEW YORK TIMES

...buried at Lychakiv Cemetery in Lviv alongside a fellow soldier, Ivan Koverznev.

na's Role: amplifying emlin Lies

By EDWARD WONG
 GTON — One of Rus-
 cendary disinforma-
 ns ramped up days
 defense and foreign
 issued statements
 g the Pentagon

Putin Threatens Takeover of Western Cor

By ANTON TROIANOVSKI

Besieged by an onslaught of sanctions that have largely un- done 30 years of economic inte- gration with the West in the space of two weeks, President Vladimir V. Putin on Thursday opened the door to nationalizing the assets of Western companies pulling out of Russia and exhorted senior offi- cials to "act decisively" to pre- serve jobs.

With Russia in danger of de- faulting on its sovereign debt and facing a sharp contraction in its economy, the West is betting that the looming, generation-defining economic crisis could make Rus- sian turn on their president. It is possible, however, that the crisis would end up strengthening Putin, validating his narrative that the West is determined to de-

Russian Leader Facing Generation-Defining Economic Crisis

"I have no doubt that these sanctions would have been imple- mented no matter what," Mr. Putin said in televised remarks on Thursday, arguing that his inter- vention in Ukraine served merely as a pretext for the West to try to wreck Russia's economy. "Just as we overcame these difficulties in years past, we will overcome them now, too."

But the sanctions imposed in the two weeks since the invasion — combined with multinational companies that employ tens of thousands of Russians voluntarily deciding to withdraw amid the

global outrage — economic pressu- has faced under 1 With the rub nearly half its month, prices risen sharply, cau ing at supermark bank, which has k stock exchange ci war began, has is capital controls, pr companies from withe than \$5,000 in cash months.

"This will be a gi- formational down- Ruben Enikolopov, New Economic Scho The Institute of Finance, a Washingta- sation of financial- dicted that Russia- percent decline in it

Continued on Po



GRIEF IN KYIV V-1

Germany to aid the likely not follow suit because they S&P 500 dropped 0.7 percent

how to build software?
two problems to solve





functional structures





functional structures



implementation structures



is it usable? effective? fit for purpose?

functional structures

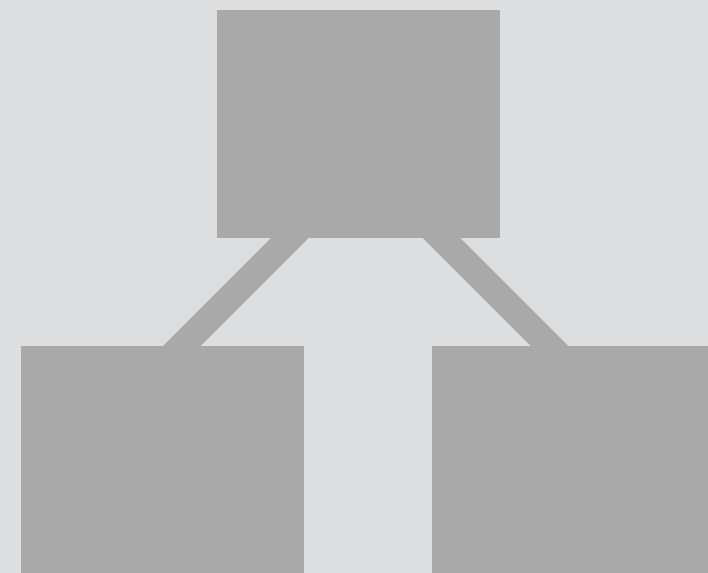


implementation structures



functional structures

is it usable? effective? fit for purpose?



implementation structures

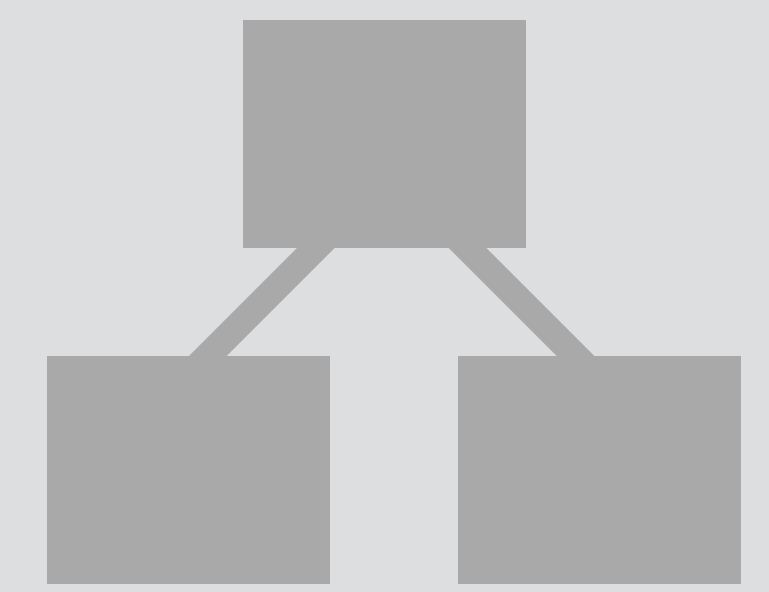
is it robust? performant? maintainable?



functional structures

is it usable? effective? fit for purpose?

software design



implementation structures

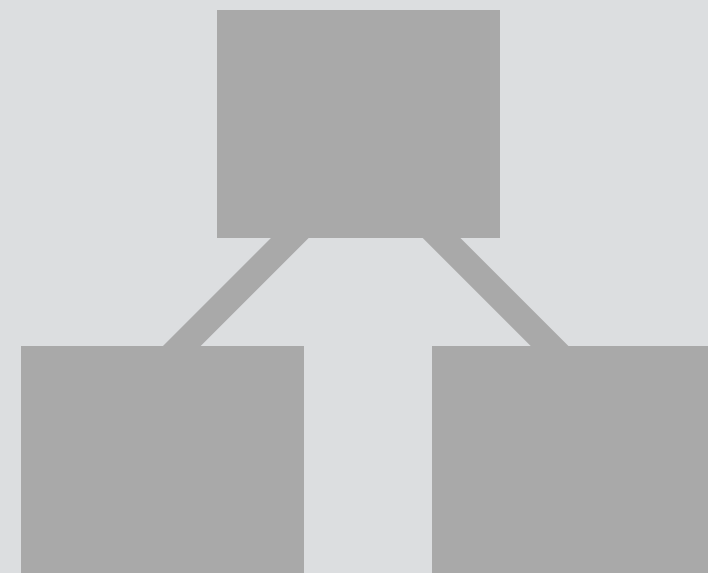
is it robust? performant? maintainable?



functional structures

is it usable? effective? fit for purpose?

software design



implementation structures

is it robust? performant? maintainable?

software engineering



The rôle of a formal functional specification is simply to act as a logical firewall between two completely different concerns... The **pleasantness problem** concerns the question whether a system meeting such-and-such a formal functional specification would satisfy our needs, meet our expectations and fulfill our hopes. The **correctness problem** concerns the question whether a given design meets such-and-such a formal functional specification.

Dijkstra, EWD952 (1986)

ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS



ESSAYS ON SOFTWARE ENGINEERING

THE
MYTHICAL
MAN-MONTH

FREDERICK P. BROOKS, JR.

ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS



ESSAYS ON SOFTWARE ENGINEERING

THE MYTHICAL MAN-MONTH

FREDERICK P. BROOKS, JR.

Conceptual integrity is the most important consideration in system design.

Fred Brooks, *Mythical Man Month* (1975)

ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS



ESSAYS ON SOFTWARE ENGINEERING

THE MYTHICAL MAN-MONTH

FREDERICK P. BROOKS, JR.

Conceptual integrity is the most important consideration in system design.

Fred Brooks, *Mythical Man Month* (1975)

I am more convinced than ever. Conceptual integrity is central to product quality.

Mythical Man Month Anniversary Edition (1995)

ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS



ESSAYS ON SOFTWARE ENGINEERING

THE MYTHICAL MAN-MONTH

FREDERICK P. BROOKS, JR.

Conceptual integrity is the most important consideration in system design.

Fred Brooks, *Mythical Man Month* (1975)

I am more convinced than ever. Conceptual integrity is central to product quality.

Mythical Man Month Anniversary Edition (1995)

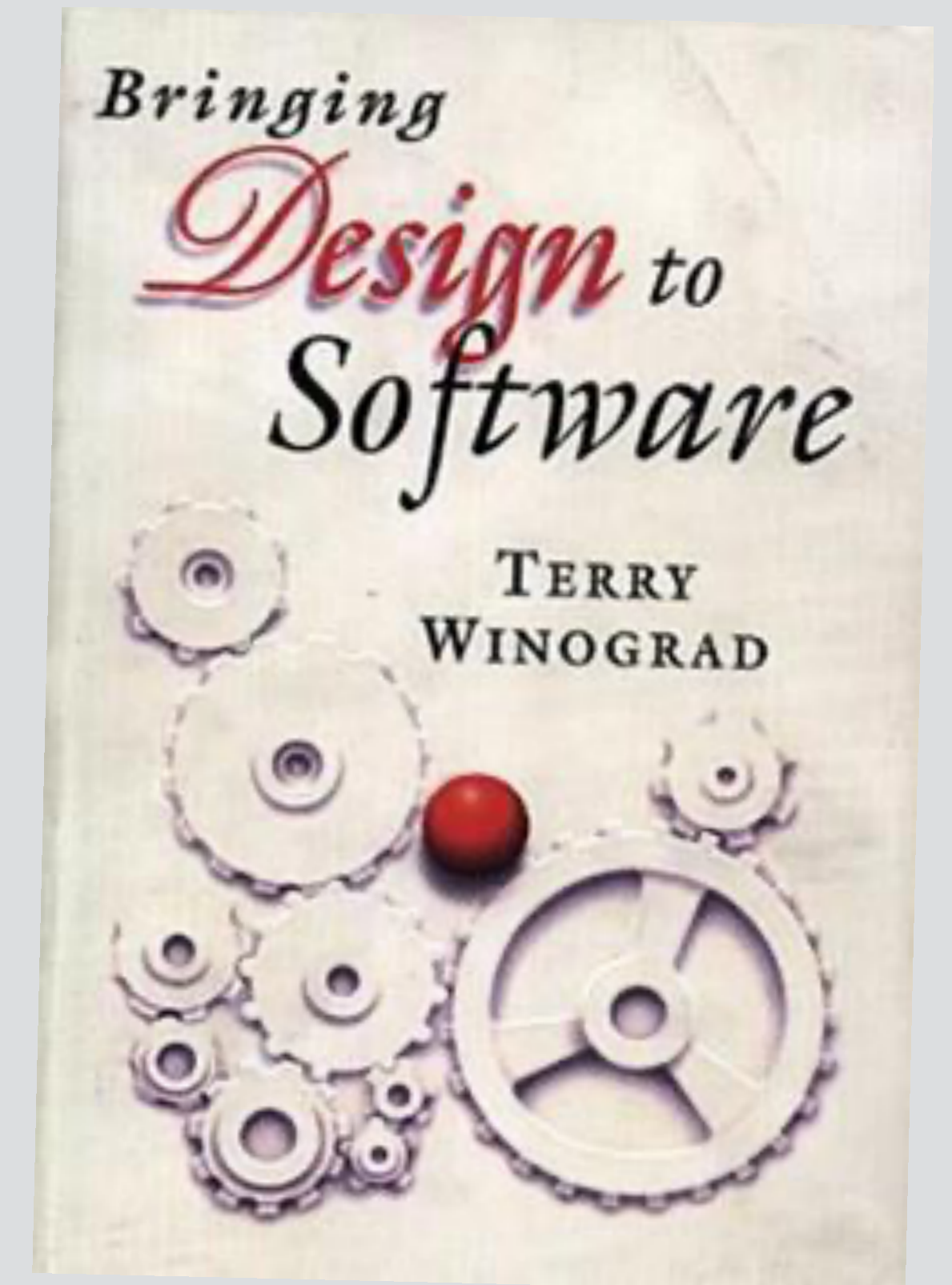
The essence of a software entity is a construct of interlocking concepts... I believe the hard part of building software to be the specification, design, and testing of this conceptual construct, not the labor of representing it...

No Silver Bullet (1986)



When you go to design a house
you talk to an architect first, not
an engineer. Why is this?

A Software Design Manifesto
Mitchell Kapor, 1996
paraphrasing slightly

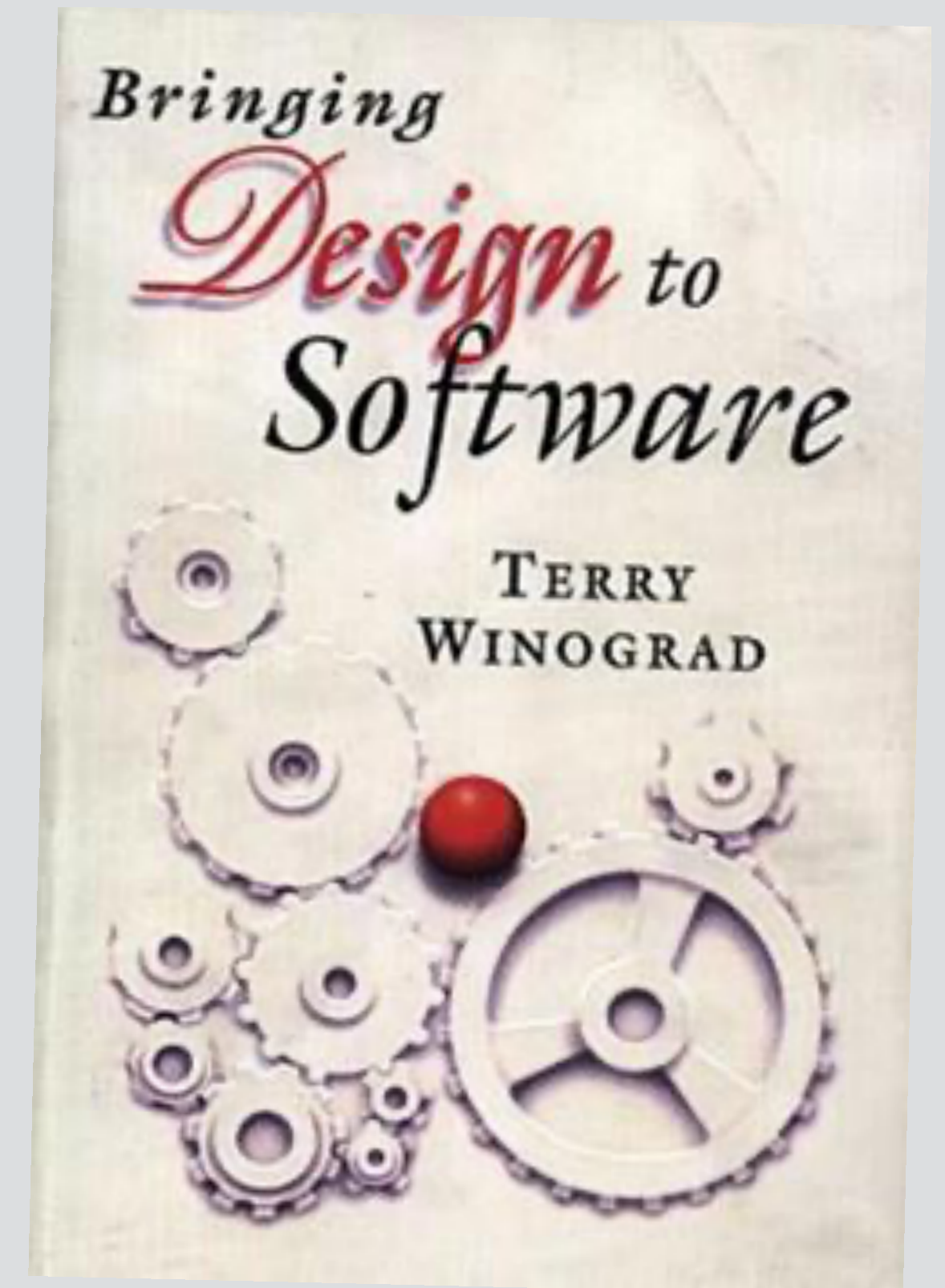




When you go to design a house **you talk to an architect first**, not an engineer. Why is this?

Because the criteria for what makes a good building fall outside the domain of engineering.

A Software Design Manifesto
Mitchell Kapor, 1996
paraphrasing slightly



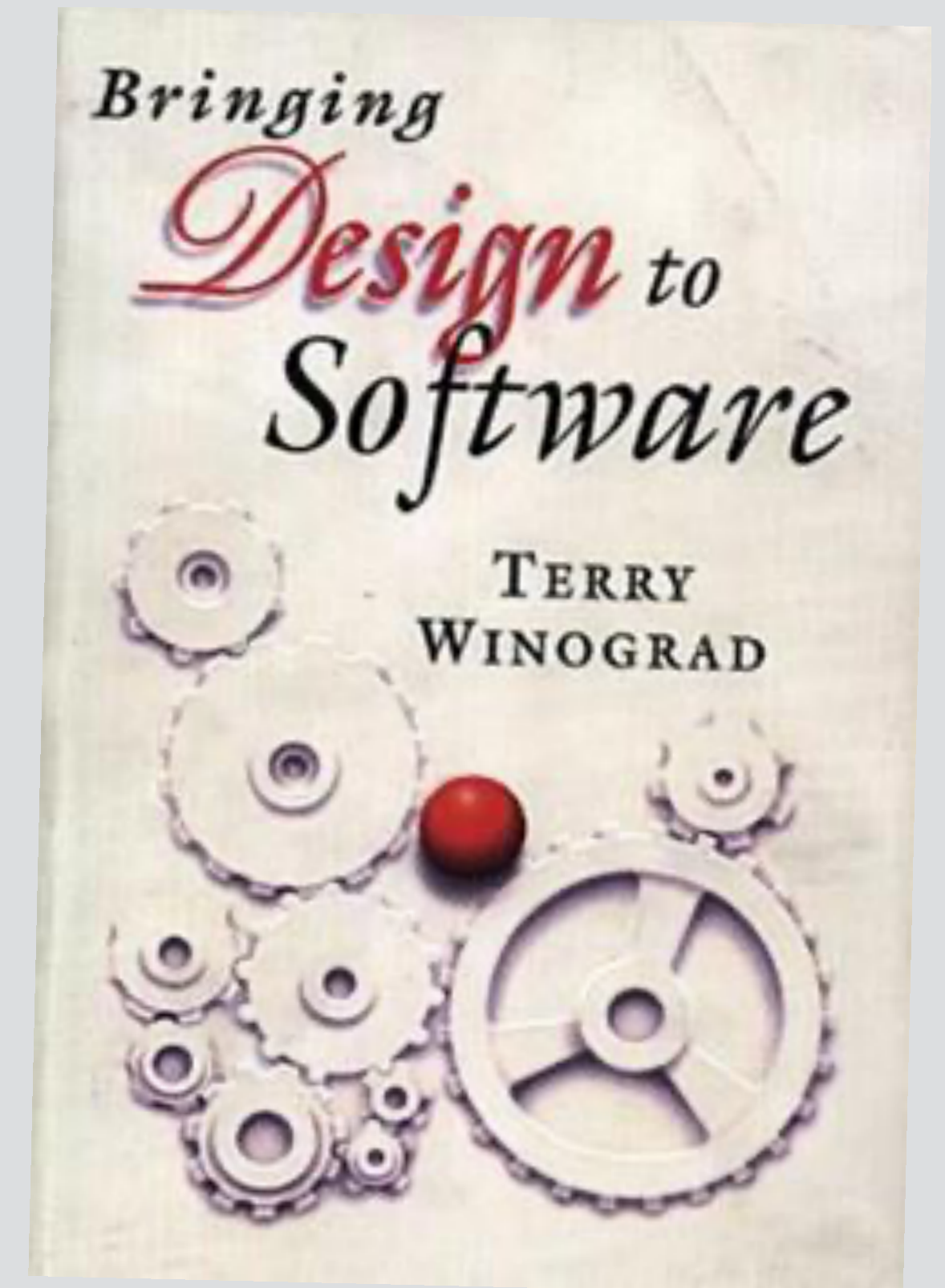


When you go to design a house **you talk to an architect first**, not an engineer. Why is this?

Because the criteria for what makes a good building fall outside the domain of engineering.

Similarly, in computer programs, the **selection of the various components** and elements of the application must be driven by the conditions of use. **How is this to be done?** By software designers.

A Software Design Manifesto
Mitchell Kapor, 1996
paraphrasing slightly



**what makes
great designers?**

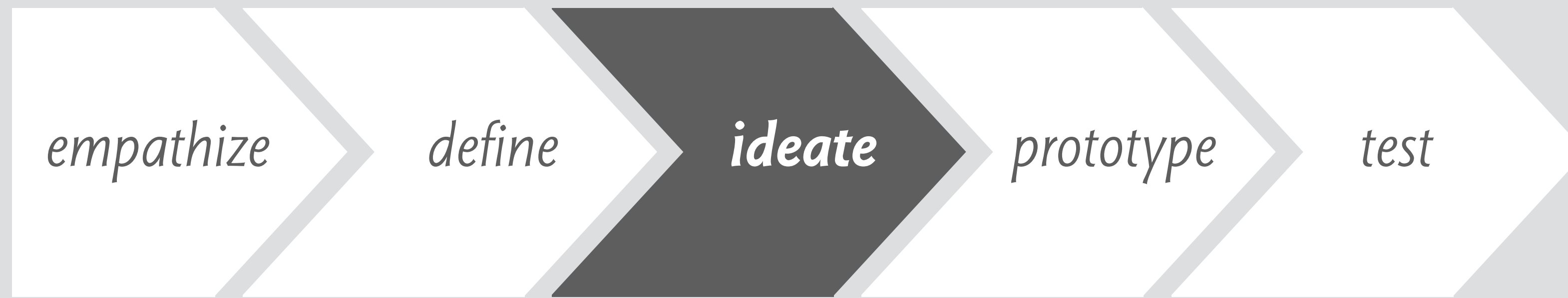
design by inspiration

inspiration = talent + process

design by inspiration

inspiration = talent + process

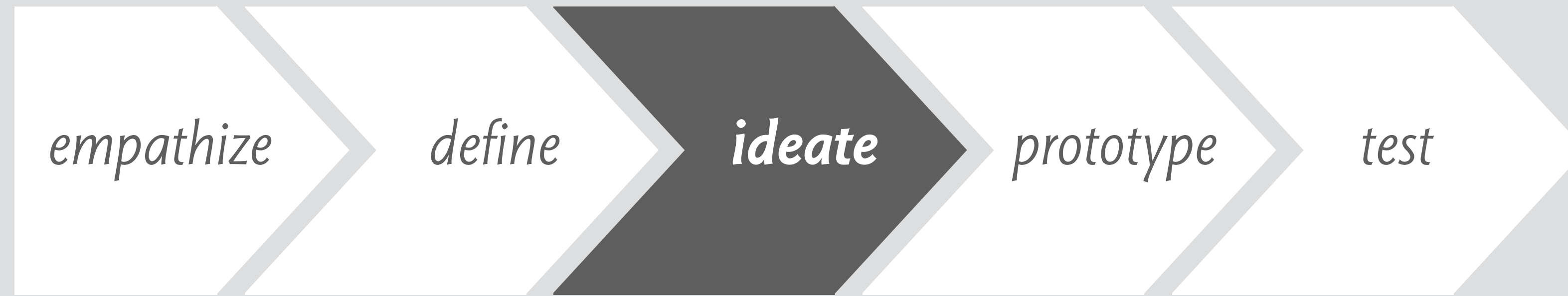
the design thinking process



design by inspiration

inspiration = talent + process

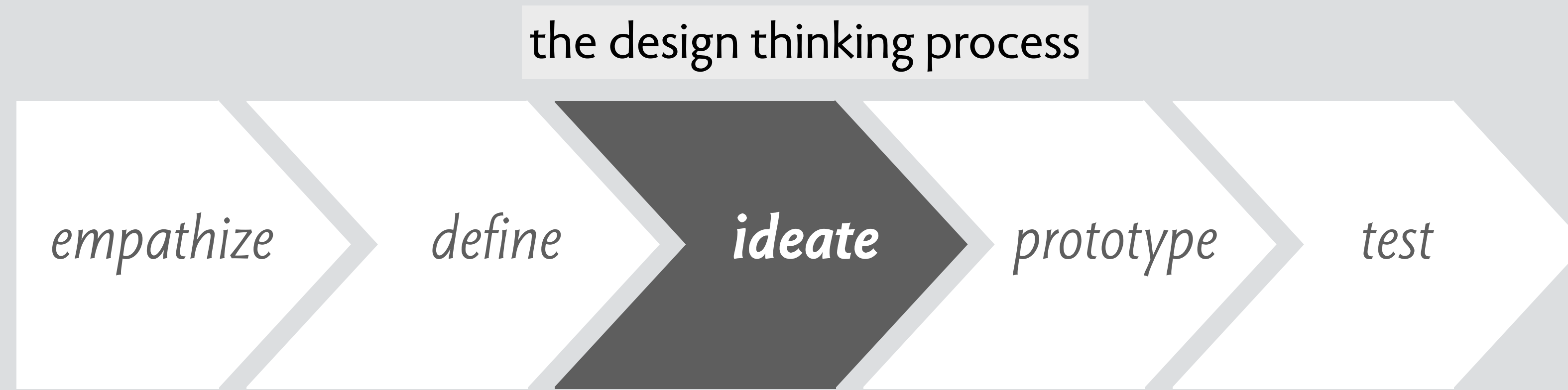
the design thinking process



problem at hand defines context

design by inspiration

inspiration = talent + process



problem at hand defines context

is there any great design that is the result of a great design process?

design by expertise

expertise = patterns + design moves

familiar structures

familiar steps

design by expertise

expertise = patterns + design moves

familiar structures

familiar steps

The well-kept secret behind the most successful designs is that they are less new than they appear. Invariably, it turns out that, in some form or other, the designer has done the same thing before. —AFD

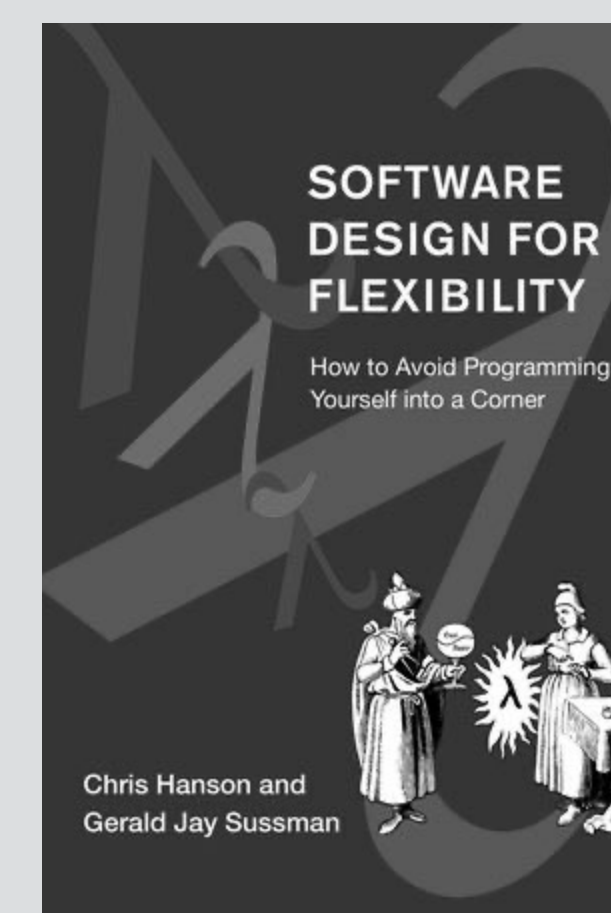
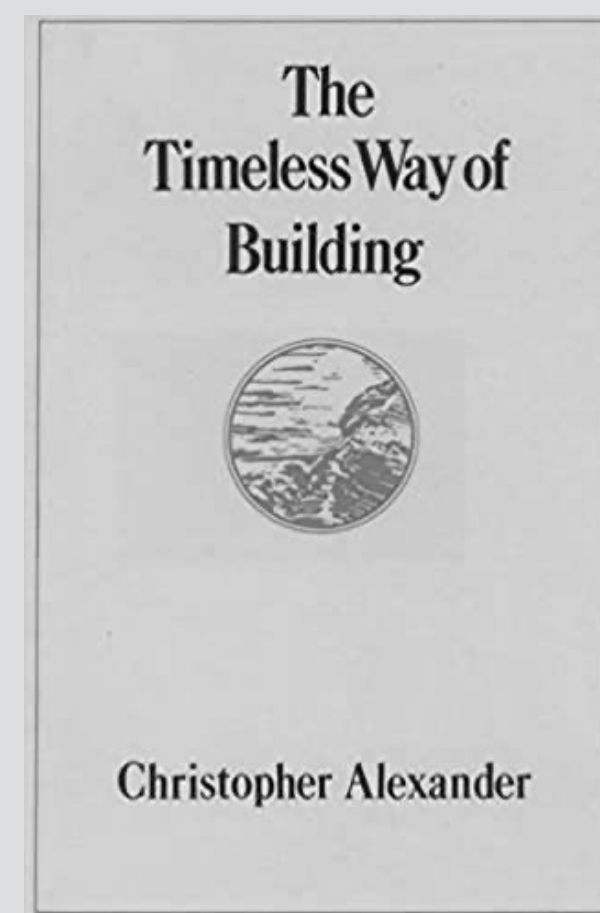
design by expertise

expertise = patterns + design moves

familiar structures

familiar steps

The well-kept secret behind the most successful designs is that they are less new than they appear. Invariably, it turns out that, in some form or other, the designer has done the same thing before. —AFD



**what are the elements
of software design?**

▲ 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 might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift in 2009.

For those who don't know JSP, I'd point to these ideas as worth knowing:

- There's a class of programming problem that involves traversing context-free structures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized input and output.
- There are some archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them helps.
- 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.

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

▲ 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 might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift in 2009.

For those who don't know JSP, I'd point to these ideas as worth knowing:

- There's a class of programming problem that involves traversing context-free structures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized input and output.
- There are some archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them helps.
- 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.

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

▲ Jackson structured programming (wikipedia.org)

post

session

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

▲ danielnicholas 63 days ago [-]

If you want an intro to JSP, you might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift in 2009.

For those who don't know JSP, I'd point to these ideas as worth knowing:

- There's a class of programming problem that involves traversing context-free structures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized input and output.
- There are some archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them helps.
- 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.

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

▲ Jackson structured programming (wikipedia.org)

post

session

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

▲ danielnicholas 63 days ago [-]

If you want an intro to JSP, you might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift in 2009.

For those who don't know JSP, I'd point to these ideas as worth knowing:

- There's a class of programming problem that involves traversing (or generating) tree structures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized it.
- There are some archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them helps.
- 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.

comment

[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.

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

▲ Jackson structured programming (wikipedia.org)

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

post

session

▲ danielnicholas 63 days ago [-]

If you want an intro to JSP, you might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift in 2009.

For those who don't know JSP, I'd point to these ideas as worth knowing:

- There's a class of programming problem that involves traversing (or generating) tree structures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized it.
- There are some archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them helps.
- 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.

comment

[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 backtrack standard method, while mentioning that most languages lacked that, so it had to be implemented manually.

reply

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

▲ Jackson structured programming (wikipedia.org)

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

post

session

upvote

▲ danielmicolas 63 days ago [-]

If you want an intro to JSP, you might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift in 2009.

For those who don't know JSP, I'd point to these ideas as worth knowing:

- There's a class of programming problem that involves traversing (or generating) tree structures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized it.
- There are some archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them helps.
- 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.

comment

[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 backtrack standard method, while mentioning that most languages lacked that, so it had to be implemented manually.

reply

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

▲ Jackson structured programming (wikipedia.org)

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

post

session

upvote

favorite

▲ danielmicolas 63 days ago [-]

If you want an intro to JSP, you might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift in 2009.

For those who don't know JSP, I'd point to these ideas as worth knowing:

- There's a class of programming problem that involves traversing (or generating) tree structures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized it.
- There are some archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them helps.
- 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.

comment

[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 backtrack standard method, while mentioning that most languages lacked that, so it had to be implemented manually.

reply

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

▲ Jackson structured programming (wikipedia.org)

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

post

session

upvote

favorite

▲ danielnicholas 63 days ago [-]

user: danielnicholas

created: 63 days ago

karma: 11

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:

comment

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.

- There are some archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them helps.
- 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 backtrack standard method, while mentioning that most languages lacked that, so it had to be implemented manually.

reply

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

▲ Jackson structured programming (wikipedia.org)

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

post

session

upvote

favorite

▲ danielnicholas 63 days ago [-]

user: danielnicholas

created: 63 days ago

karma: 11

comment

karma

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 backtrack standard method, while mentioning that most languages lacked that, so it had to be implemented manually.

reply

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

a concept has a name

concept Upvote

a concept has a name

concept Upvote

same concept in HackerNews,
NYTimes comment section,
StackOverflow, etc

a concept has a name

concept Upvote

same concept in HackerNews,
NYTimes comment section,
StackOverflow, etc

Reader Picks

All



John

Boston | Oct. 27

To protect children? Seems far more likely it's yet one more way to extract personal information to feed the insatiable advertising machines.

[1 Reply](#) [143 Recommend](#) [Share](#)

[Flag](#)

a concept has a purpose

concept Upvote

purpose rank items by popularity



This is homework and I'm having a
are the definitions of the objects:

8



```
sig Library {  
  patrons : set Person,  
  on_shelves : set Book,  
}
```




1

a concept has a purpose


concept Upvote

purpose rank items by popularity

 This is homework and I'm having a
are the definitions of the objects:

8

```
sig Library {  
  patrons : set Person,  
  on_shelves : set Book,  
}
```



 1

concept Reaction

purpose send reactions to author

Today ▾

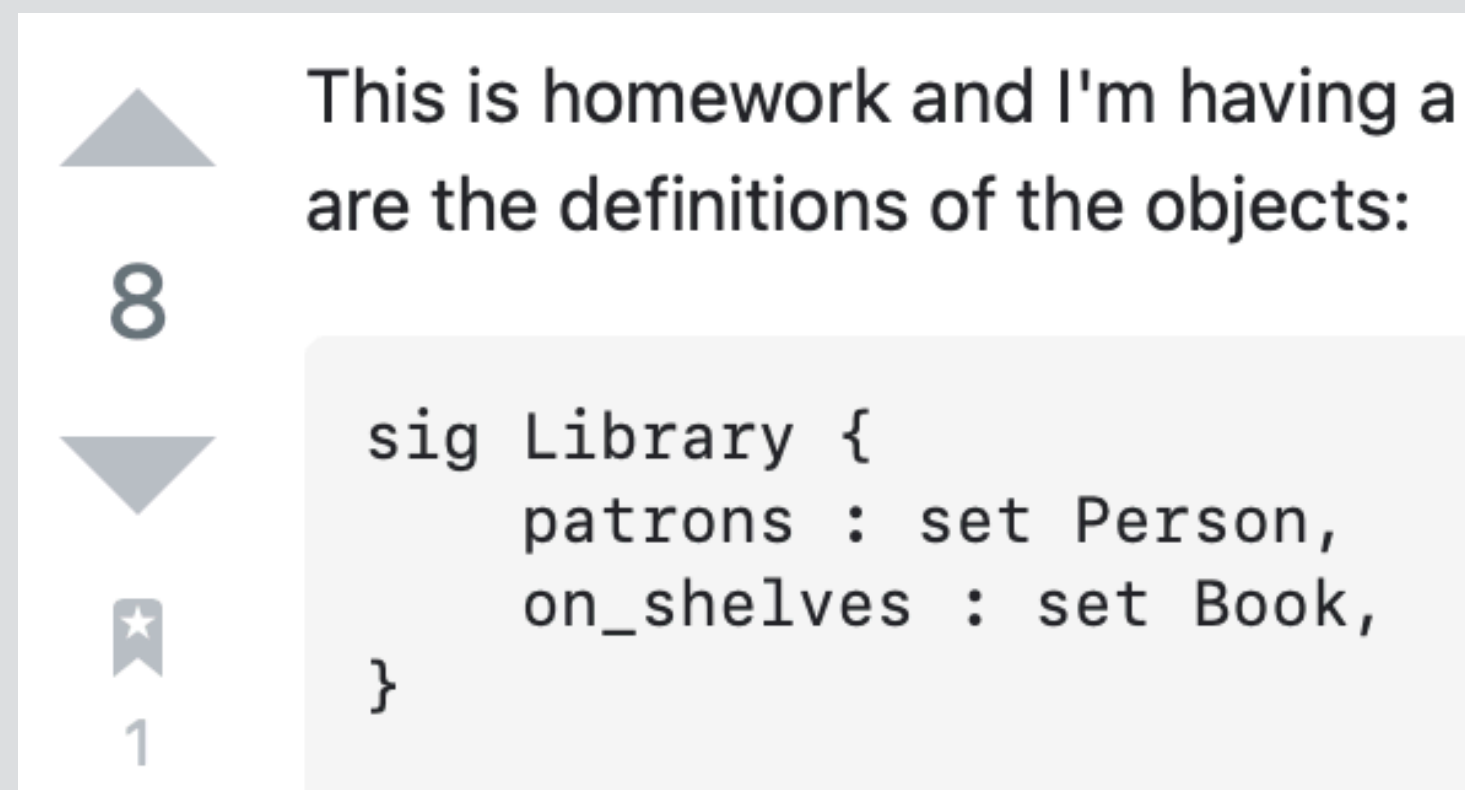
Daniel I think we should organize a
software concepts forum.

 1 

a concept has a purpose

concept Upvote

purpose rank items by popularity



This is homework and I'm having a
are the definitions of the objects:

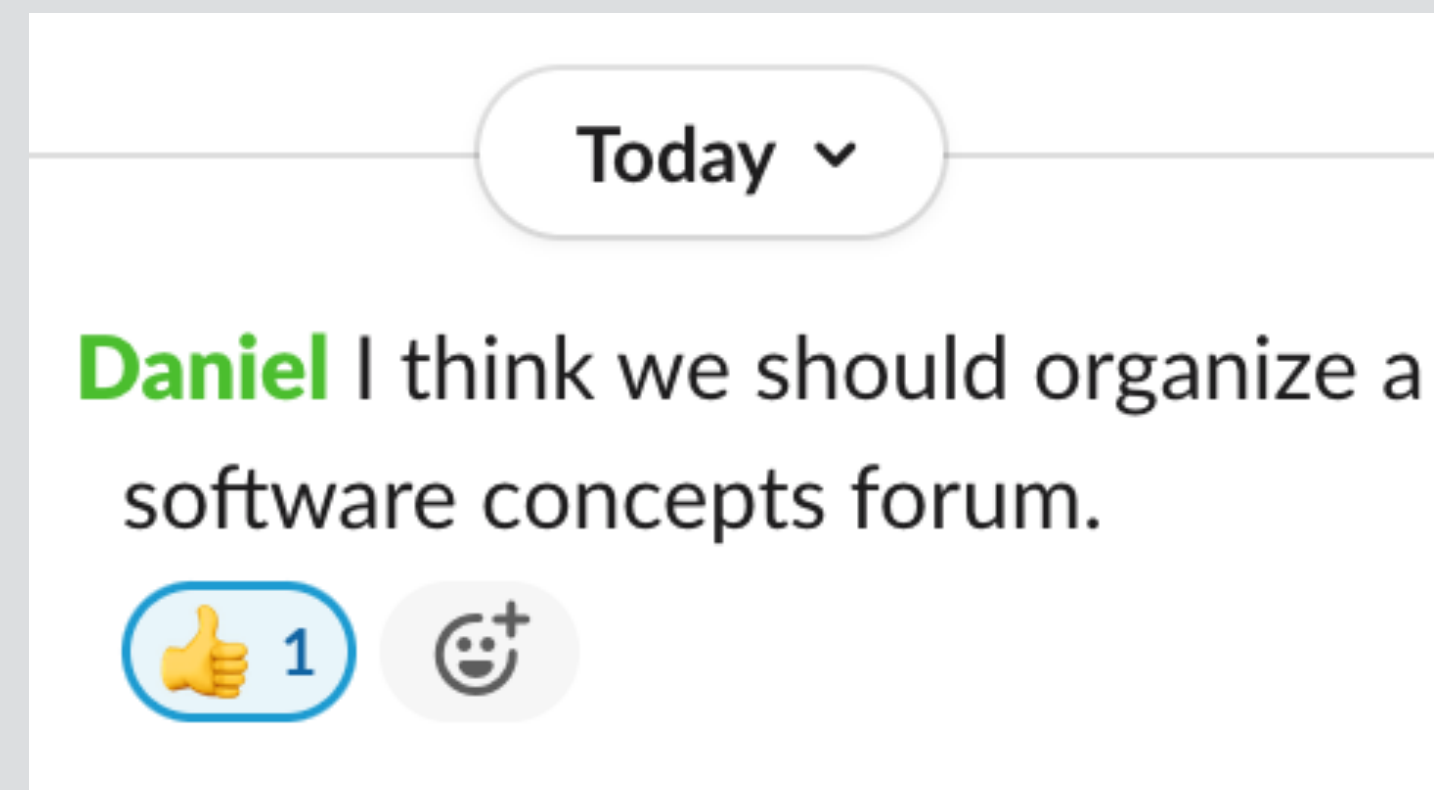
8

```
sig Library {  
  patrons : set Person,  
  on_shelves : set Book,  
}
```

1

concept Reaction

purpose send reactions to author



Today ▾

Daniel I think we should organize a
software concepts forum.

👍 1 🗨️

concept Recommendation

purpose use prior likes to recommend



SHTISEL

🔍 + 👍 🗨️ ▾

a concept has a state

concept Upvote

purpose rank items by popularity

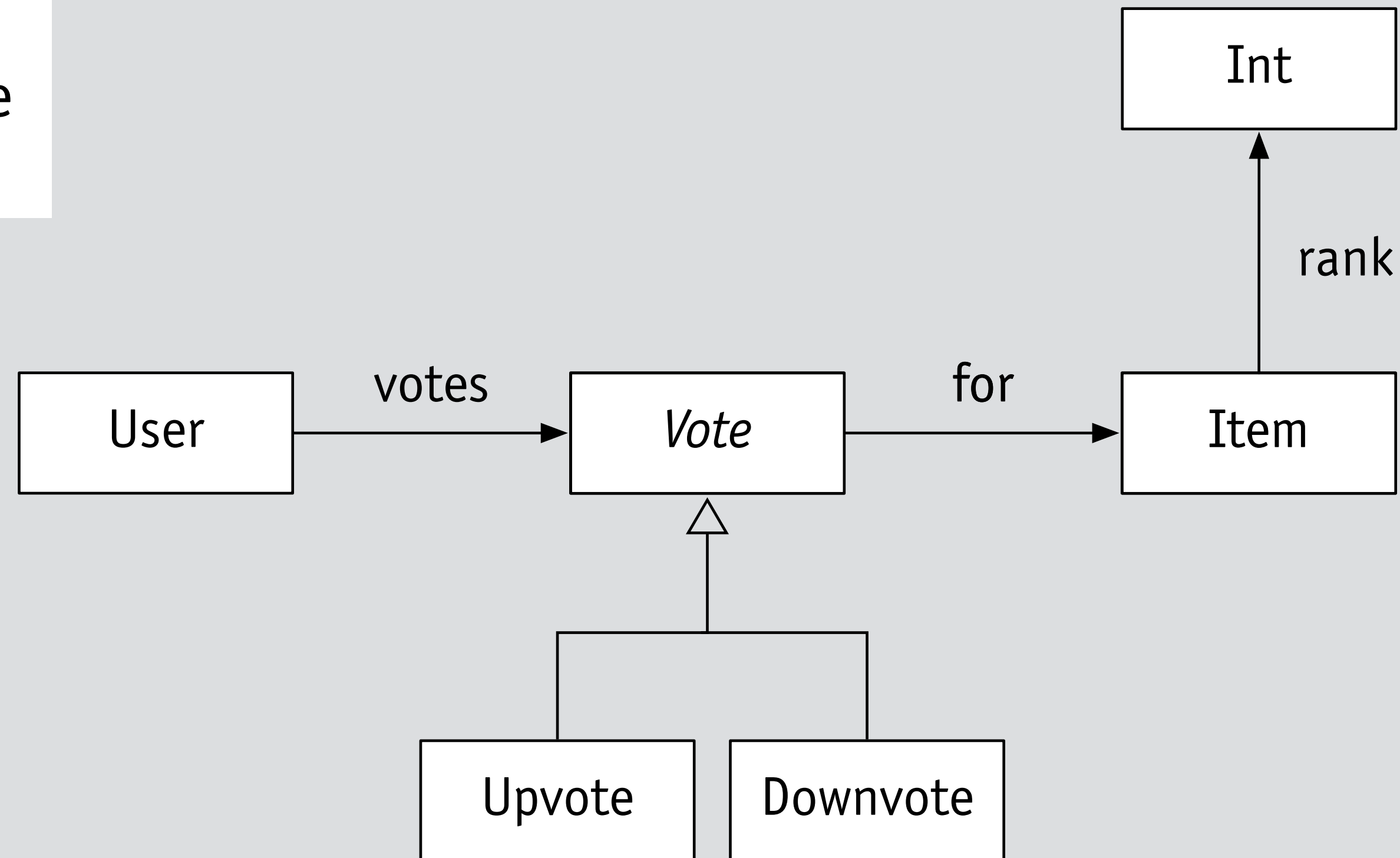
state

votes: User -> set Vote

for: Vote -> one Item

Upvote, Downvote: set Vote

rank: Item -> one Int



a concept has a state

concept Upvote

purpose rank items by popularity

state

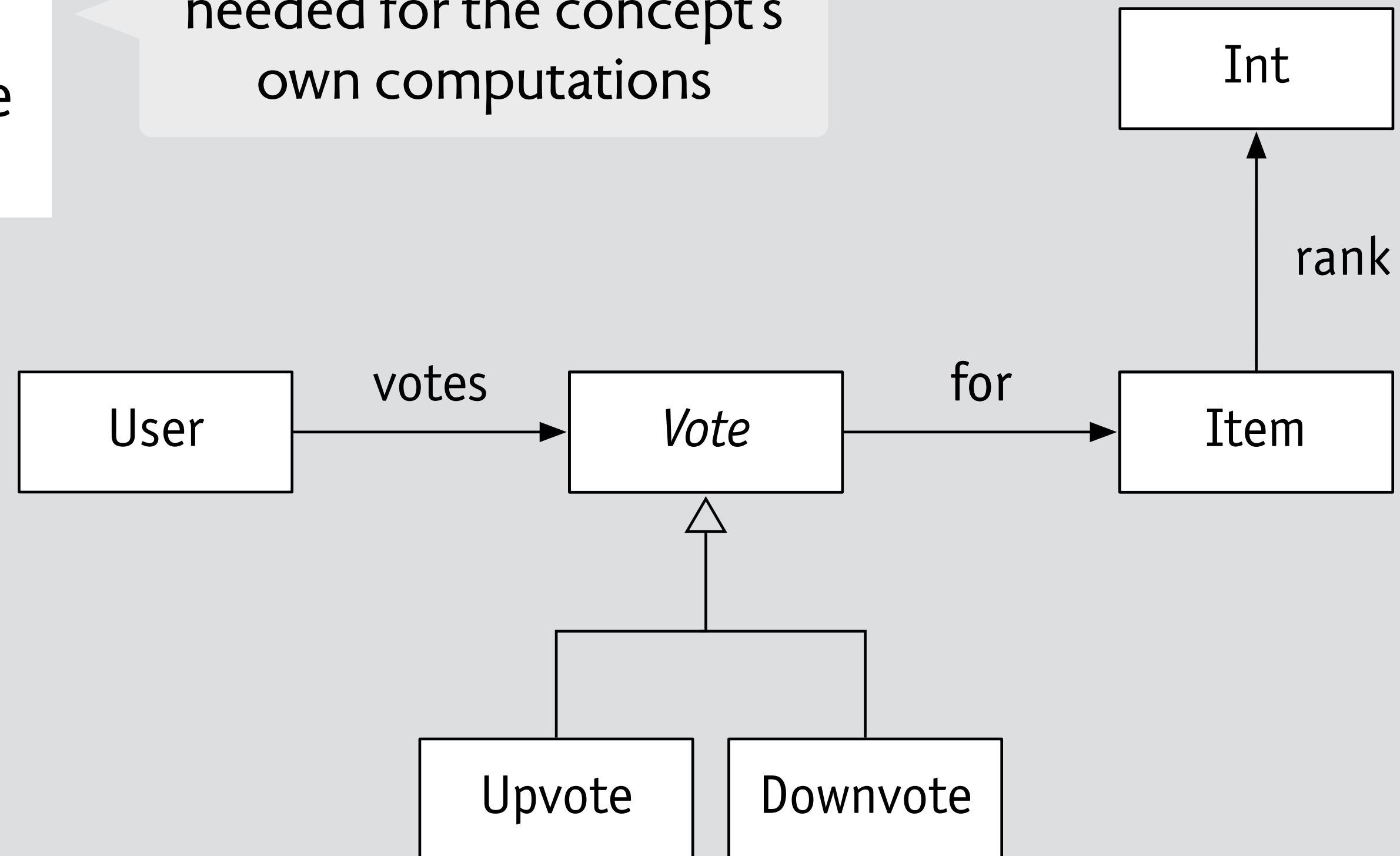
votes: User -> set Vote

for: Vote -> one Item

Upvote, Downvote: set Vote

rank: Item -> one Int

include in state **only** what's needed for the concept's own computations



a concept has a state

concept Upvote

purpose rank items by popularity

state

votes: User -> set Vote

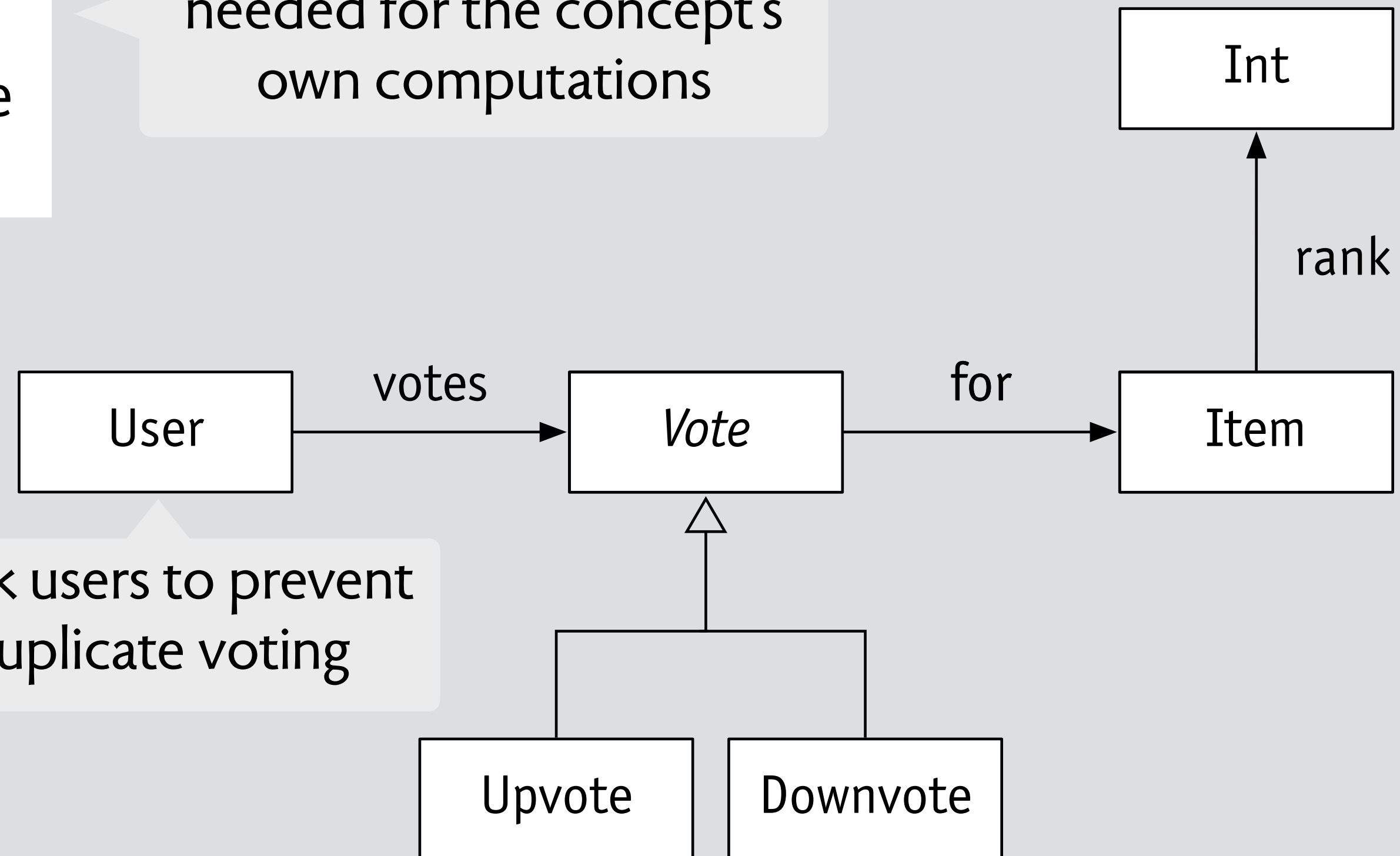
for: Vote -> one Item

Upvote, Downvote: set Vote

rank: Item -> one Int

include in state **only** what's needed for the concept's own computations

track users to prevent duplicate voting



a concept has a state

concept Upvote

purpose rank items by popularity

state

votes: User -> set Vote

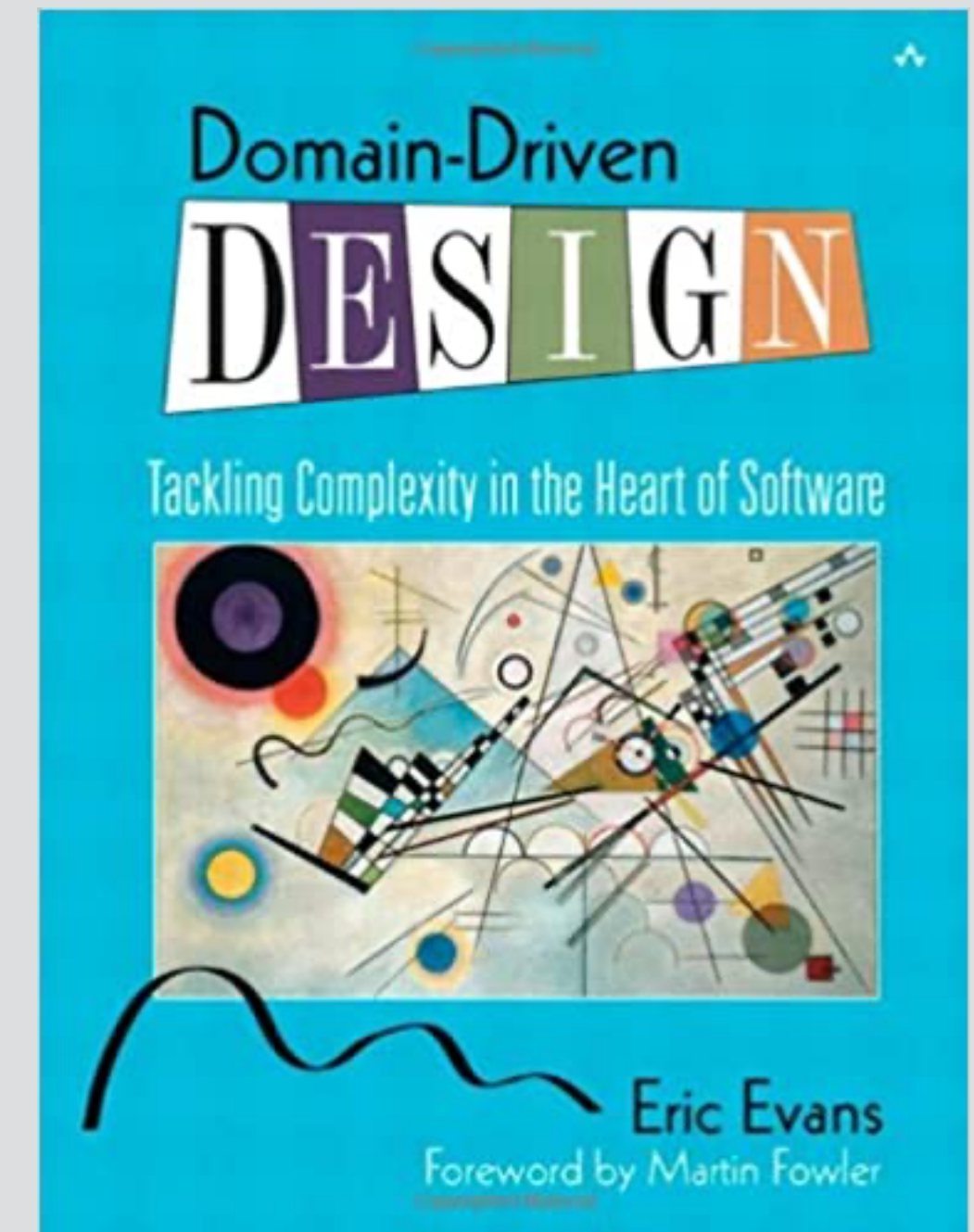
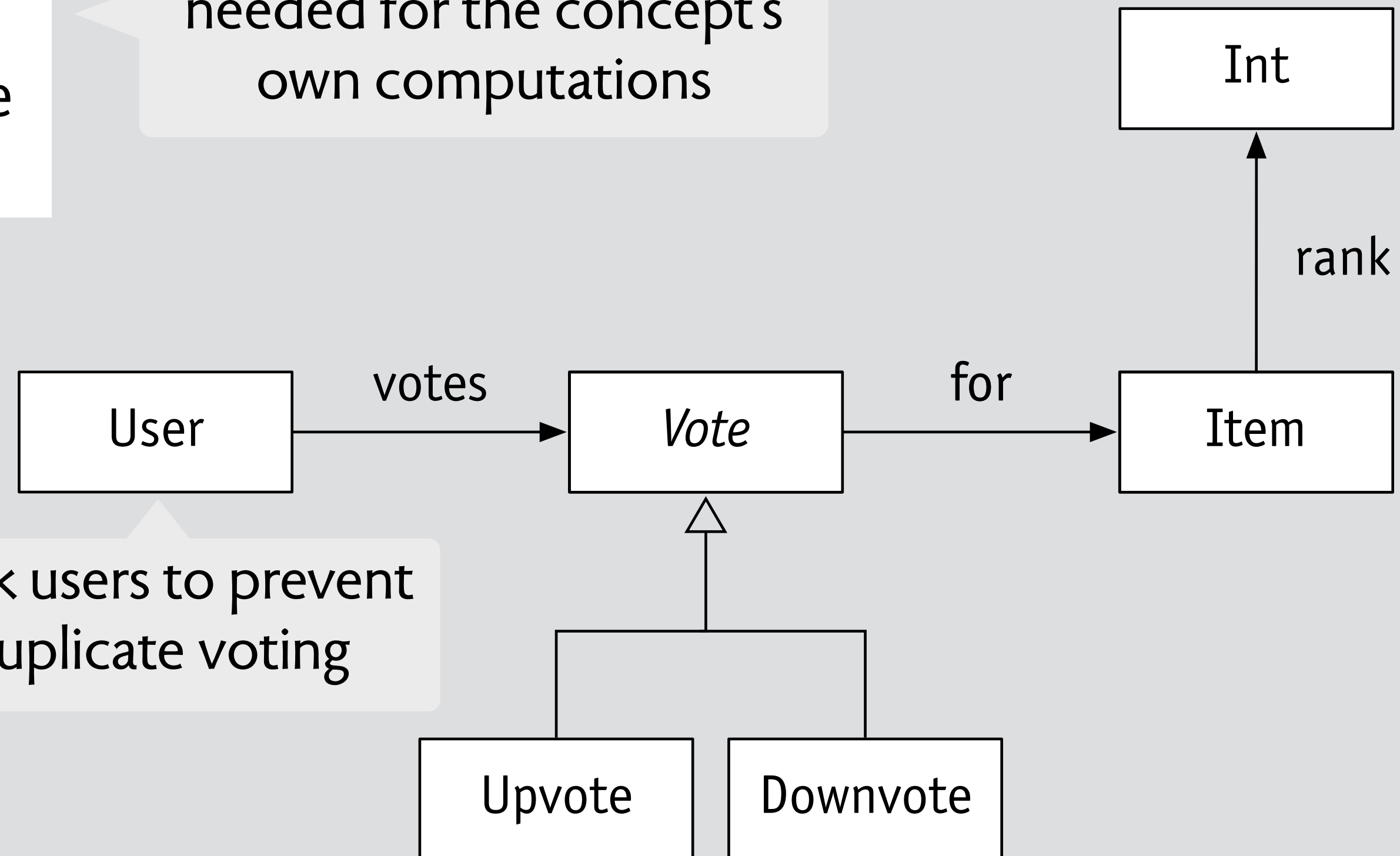
for: Vote -> one Item

Upvote, Downvote: set Vote

rank: Item -> one Int

include in state **only** what's needed for the concept's own computations

track users to prevent duplicate voting



like bounded context in DDD, but even more localized

a concept has actions

concept Upvote

purpose rank items by popularity

state

votes: User -> set Vote

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)

a concept has actions

concept Upvote

purpose rank items by popularity

state

votes: User -> set Vote

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)

actions capture the concept
behavior in full

a concept has actions

concept Upvote

purpose rank items by popularity

state

votes: User -> set Vote

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)

actions capture the concept
behavior in full

downvote (i: Item, u: User)

// no existing Downvote for i in u.votes

// remove any Upvote for i from u.votes

// add a Downvote for i in u.votes

// update i.rank ...

a concept has actions

concept Upvote

purpose rank items by popularity

state

votes: User -> set Vote

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)

actions capture the concept
behavior in full

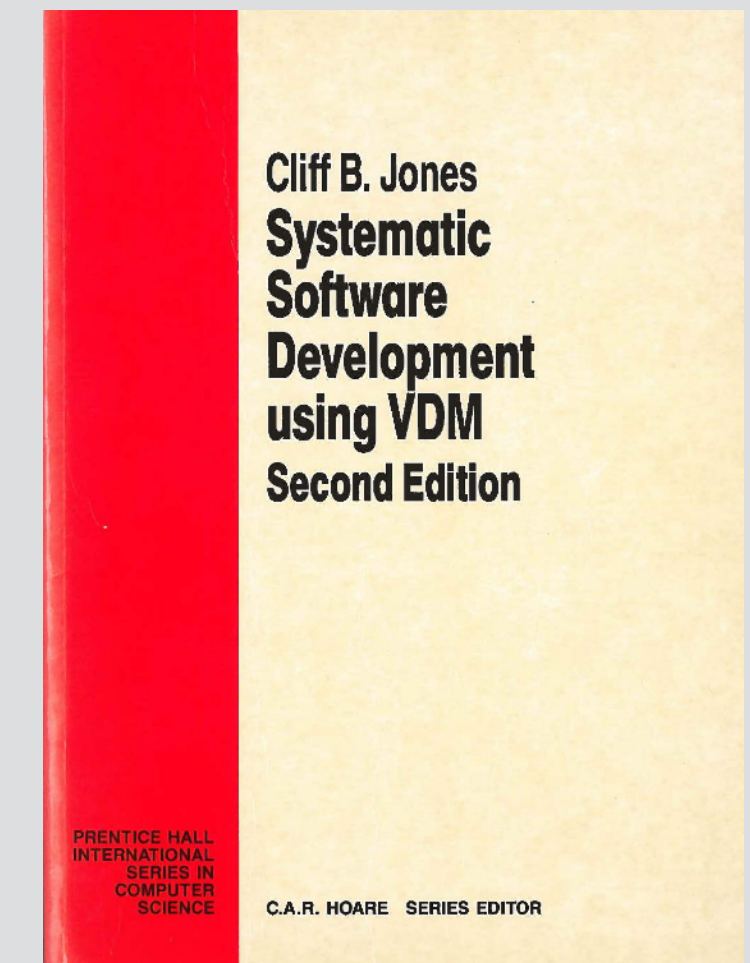
downvote (i: Item, u: User)

// no existing Downvote for i in u.votes

// remove any Upvote for i from u.votes

// add a Downvote for i in u.votes

// update i.rank ...



succinct specification
as actions on states

VDM (1986)

Z (1992)

Larch (1993)

Event-B (2006)

Alloy (2006)

a concept catalog entry

concept Upvote

related concepts

Recommendation, Reaction, ...

design variants

downvote as unvote
use age in ranking
weigh downvotes more

typical uses

social media posts
comments on articles
Q&A responses



known issues

preventing double votes
(require login, use IP address, save cookie)
saving storage space
(freeze old posts and from user info)

often used with

Karma, Session, ...

**composing concepts
by synchronization**

how to extend behavior?

concept Upvote

actions

upvote (u: User, i: Item)

downvote (u: User, i: Item)

unvote (u: User, i: Item)

how to extend behavior?

concept Upvote

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

N upvotes on your own items

how to extend behavior?

concept Upvote

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

N upvotes on your own items

define a new concept!

a hint: not just used by Upvote

how to extend behavior?

concept Upvote

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

N upvotes on your own items

define a new concept!

a hint: not just used by Upvote

concept Karma

purpose privilege good users

state

karma: User -> one Int

contribs: User -> set Item

actions

contribute (u: User, i: Item)

reward (u: User, r: Int)

permit (u: User, r: Int)

concept Upvote

actions

upvote (u: User, i: Item)

downvote (u: User, i: Item)

unvote (u: User, i: Item)

concept Karma

actions

contribute (u: User, i: Item)

reward (i: Item, r: Int)

permit (u: User, r: Int)

concept Upvote

actions

upvote (u: User, i: Item)
downvote (u: User, i: Item)
unvote (u: User, i: Item)

when upvote (u, i)
and i in u'.contribs
also reward (u', 10)

concept Karma

actions

contribute (u: User, i: Item)
reward (i: Item, r: Int)
permit (u: User, r: Int)

concept Upvote

actions

upvote (u: User, i: Item)

downvote (u: User, i: Item)

unvote (u: User, i: Item)

when upvote (u, i)
and i in u'.contribs
also reward (u', 10)

when downvote (u, i)
also permit (u, 20)

concept Karma

actions

contribute (u: User, i: Item)

reward (i: Item, r: Int)

permit (u: User, r: Int)

concept Upvote



concept Karma



concept Upvote



concept Karma

contrib (Alice, post1)



concept Upvote



concept Karma

contrib (Alice, post1)

contrib (Bob, post2)



concept Upvote

upvote (Bob, post1)

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)

concept Upvote

concept Karma

upvote (Bob, post1)

when upvote (u, i)
also reward (u, 10)

contrib (Alice, post1)

contrib (Bob, post2)

reward (Alice, 10)

concept Upvote

concept Karma

upvote (Bob, post1)

upvote (Carol, post1)

when upvote (u, i)
also reward (u, 10)

contrib (Alice, post1)

contrib (Bob, post2)

reward (Alice, 10)

reward (Alice, 10)

concept Upvote

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)

upvote (Bob, post1)

when upvote (u, i)
also reward (u, 10)

reward (Alice, 10)

upvote (Carol, post1)

reward (Alice, 10)

downvote (Alice, post2)

concept Upvote

concept Karma

upvote (Bob, post1)

upvote (Carol, post1)

downvote (Alice, post2)

when upvote (u, i)
also reward (u, 10)

when downvote (u, i)
also permit (u, 20)

contrib (Alice, post1)

contrib (Bob, post2)

reward (Alice, 10)

reward (Alice, 10)

permit (Alice, 20)

concept Upvote

concept Karma

upvote (Bob, post1)

when upvote (u, i)
also reward (u, 10)

reward (Alice, 10)

upvote (Carol, post1)

reward (Alice, 10)

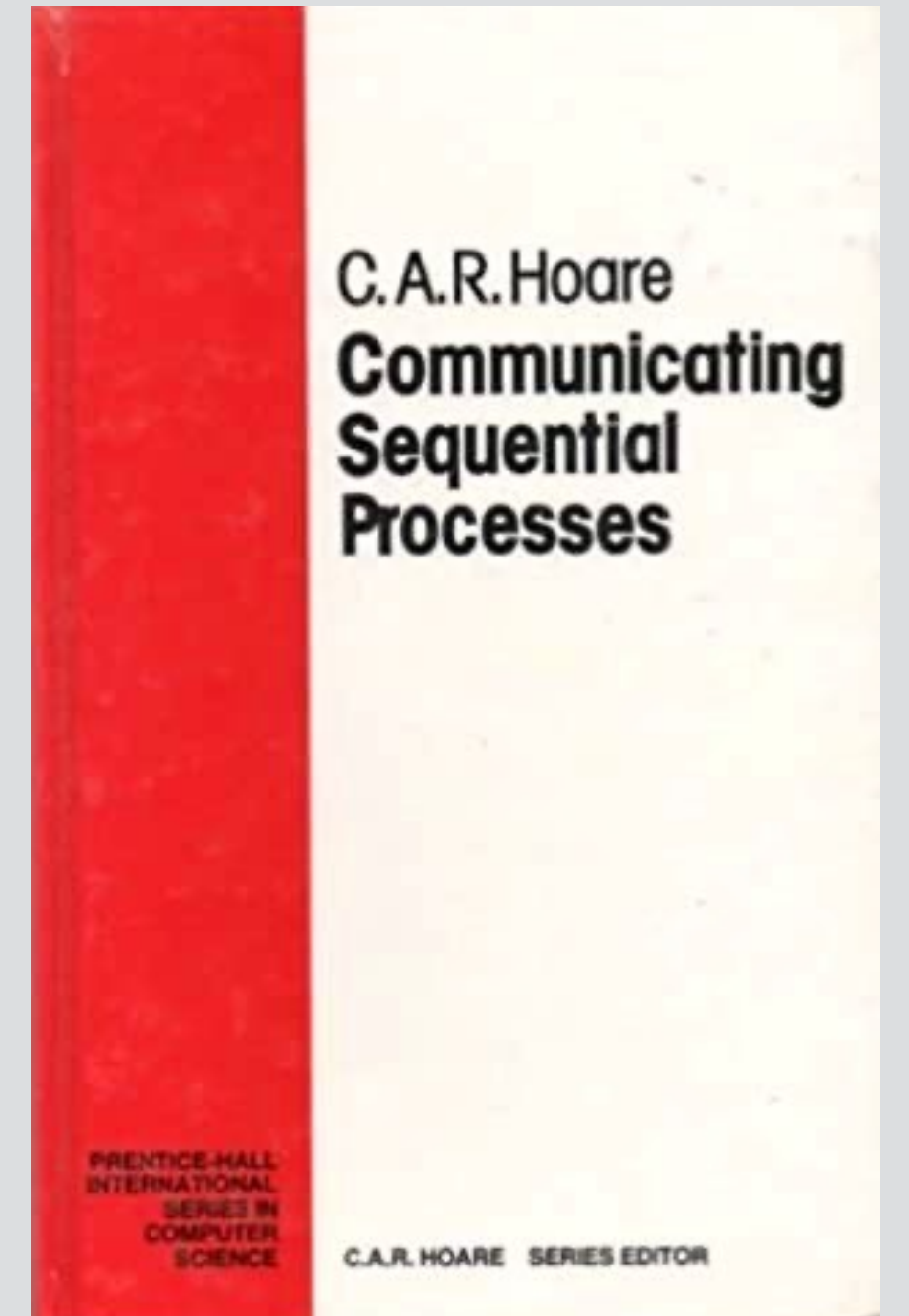
downvote (Alice, post2)

when downvote (u, i)
also permit (u, 20)

permit (Alice, 20)

contrib (Alice, post1)

contrib (Bob, post2)



composition uses
event sync from
Hoare's CSP

concept Upvote

concept Karma

upvote (Bob, post1)

when upvote (u, i)
also reward (u, 10)

reward (Alice, 10)

upvote (Carol, post1)

reward (Alice, 10)

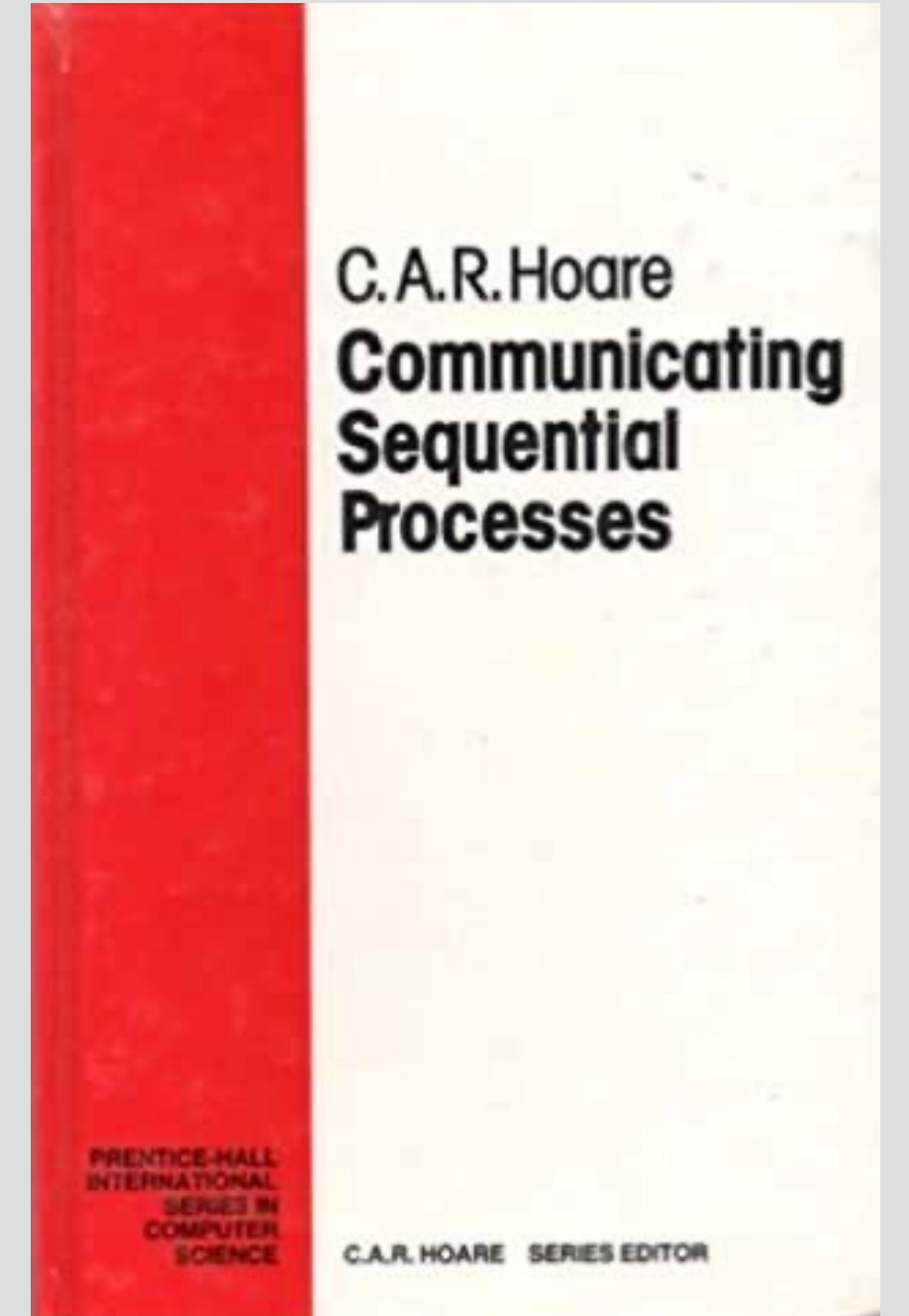
downvote (Alice, post2)

when downvote (u, i)
also permit (u, 20)

permit (Alice, 20)

contrib (Alice, post1)

contrib (Bob, post2)



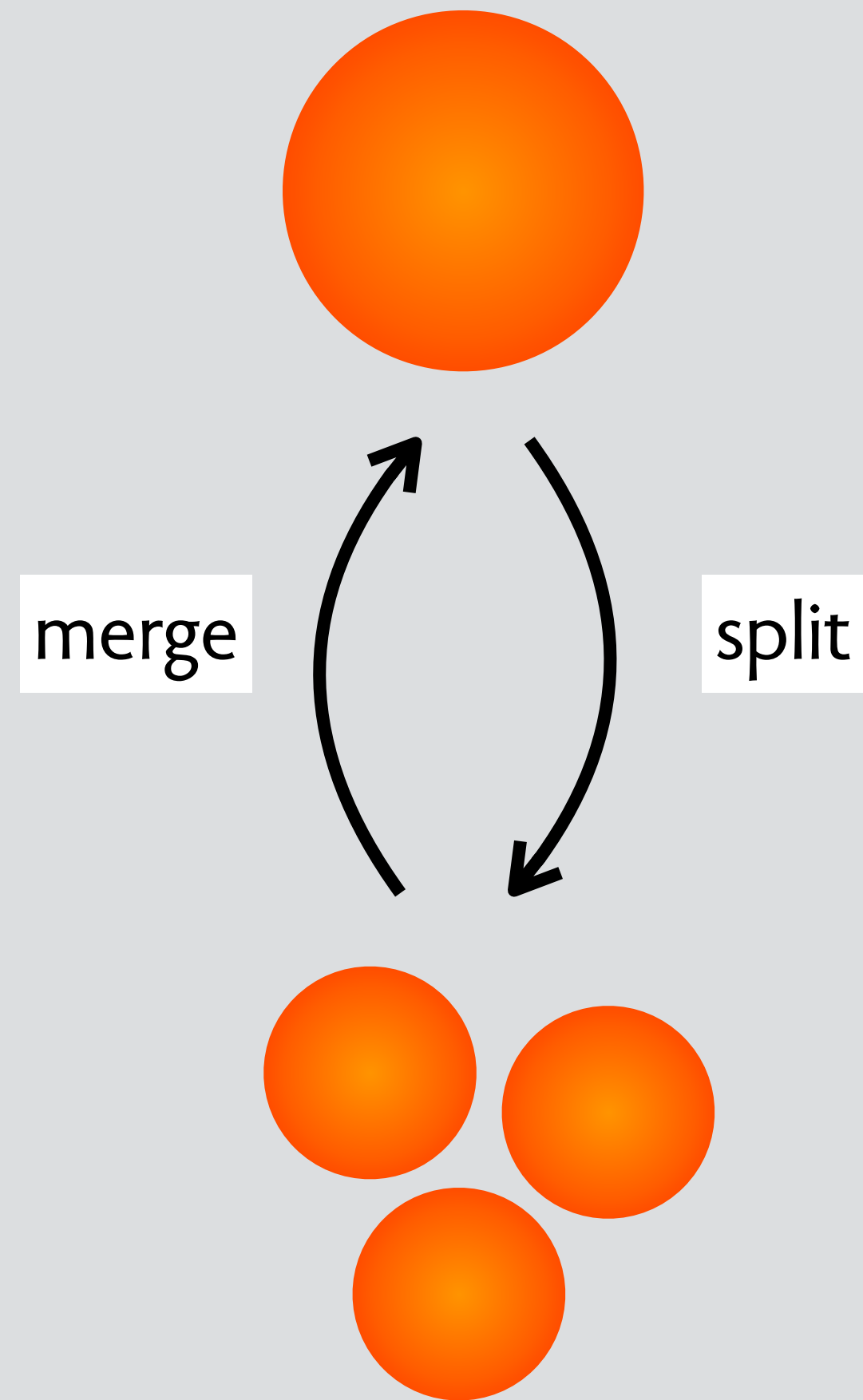
composition uses
event sync from
Hoare's CSP

**no concept coupling
concepts preserve properties**

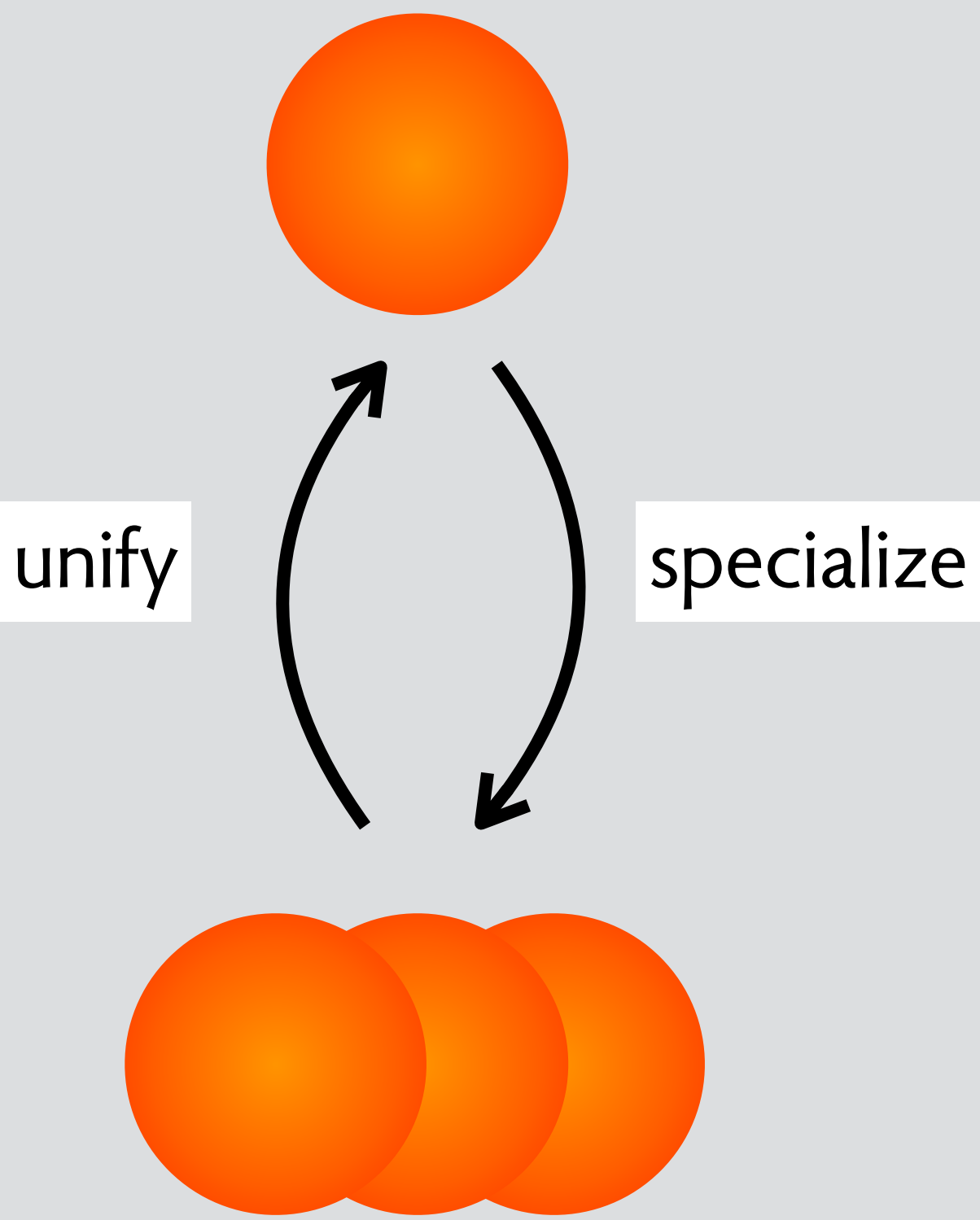
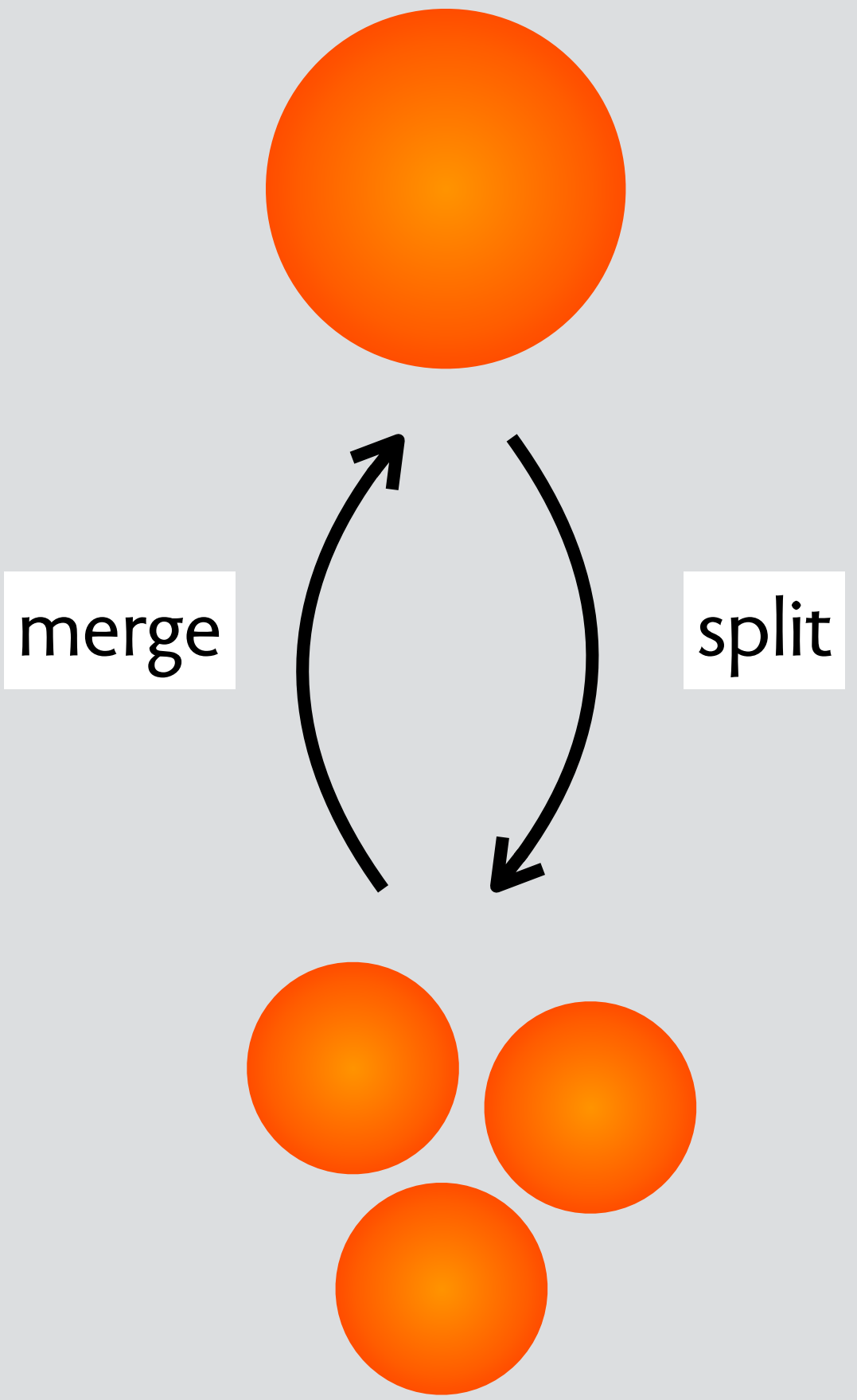
design moves
mechanical analogs

three pairs of design moves

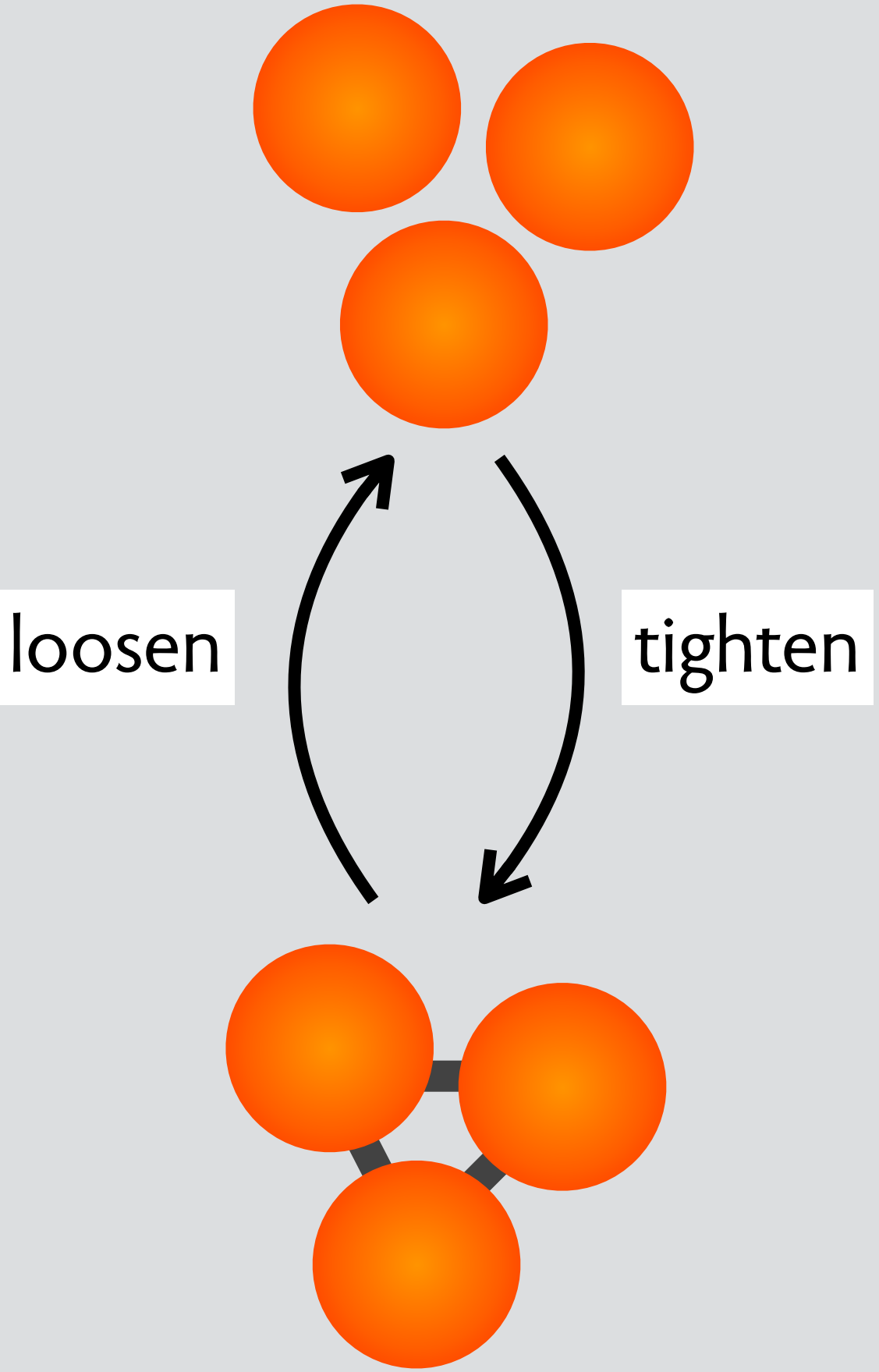
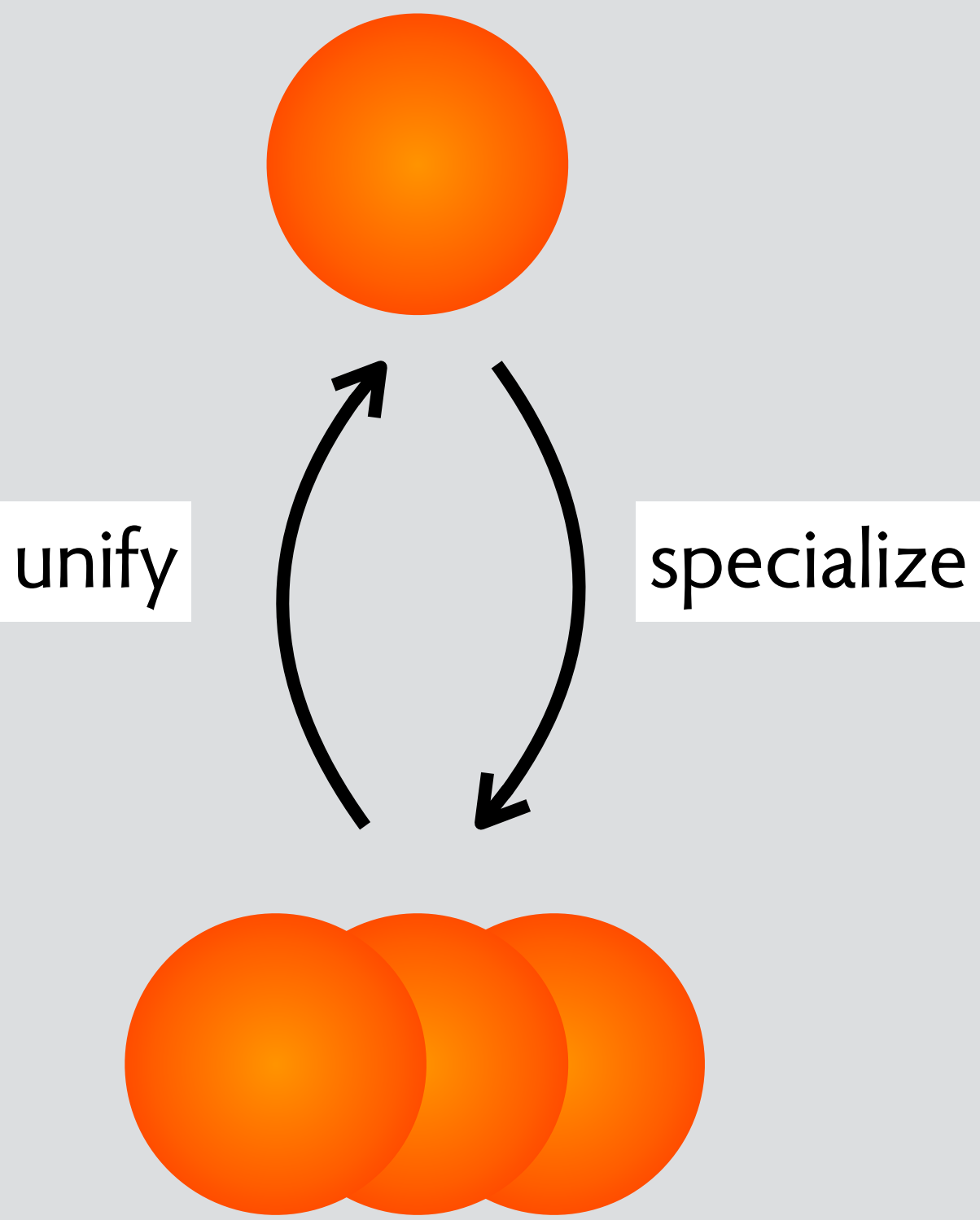
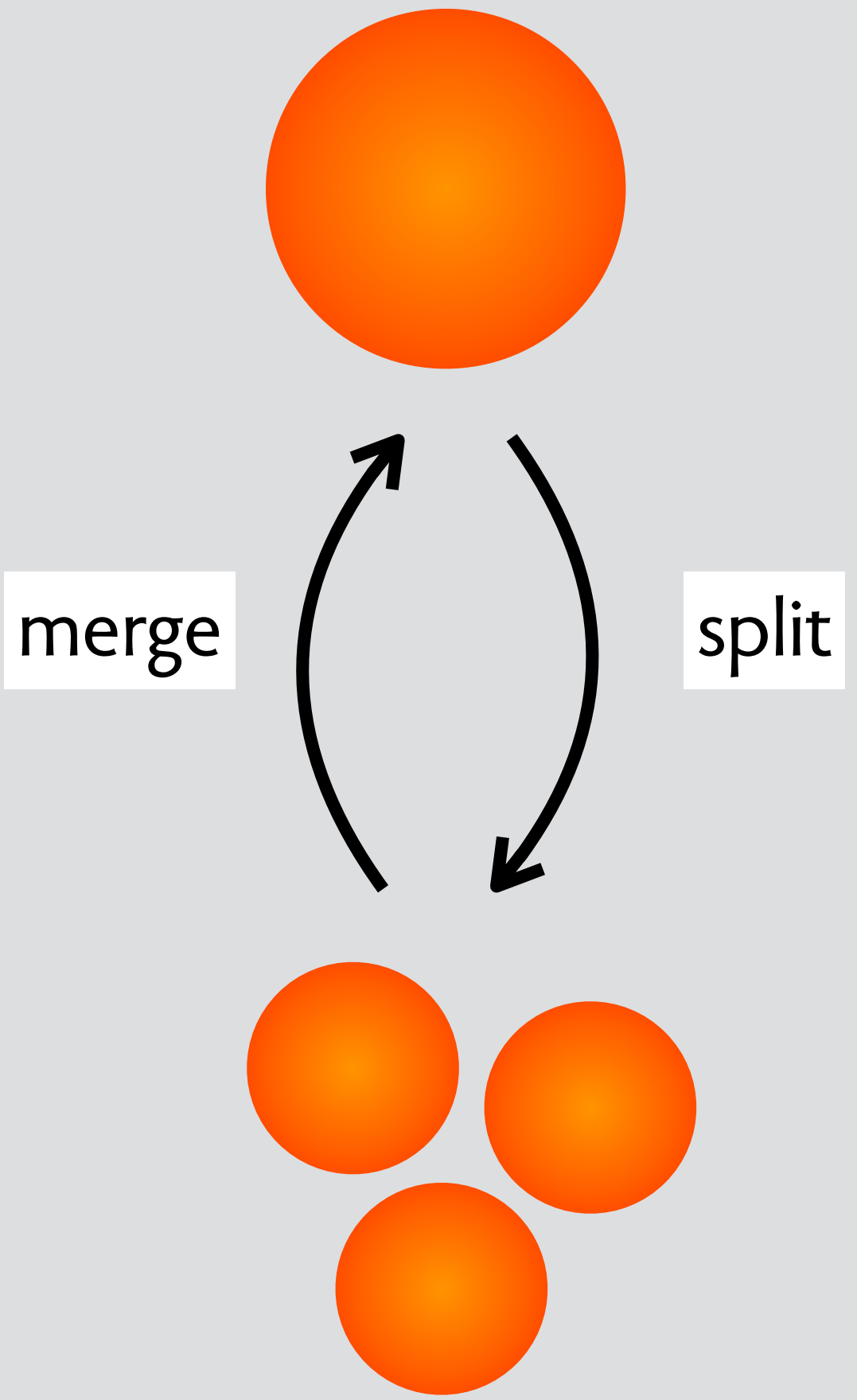
three pairs of design moves



three pairs of design moves



three pairs of design moves



split-merge: tradeoff simplicity/flexibility

split-merge: tradeoff simplicity/flexibility



photocopier

split



printer + scanner

split-merge: tradeoff simplicity/flexibility



photocopier

split



printer + scanner



emergency flashlight

merge



flashlight + battery + charger

unify-specialize: tradeoff simplicity/specificity

unify-specialize: tradeoff simplicity/specificity



set of wrenches

unify



adjustable wrench

unify-specialize: tradeoff simplicity/specificity



set of wrenches

unify



adjustable wrench

specialize



macro lens



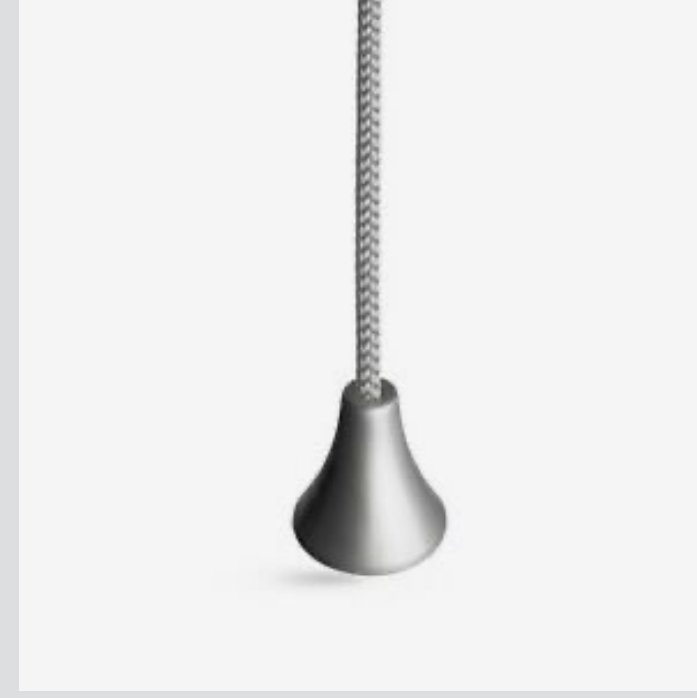
general-purpose lens

tighten-loosen: tradeoff automation/flexibility

tighten-loosen: tradeoff automation/flexibility



light pull / door lock

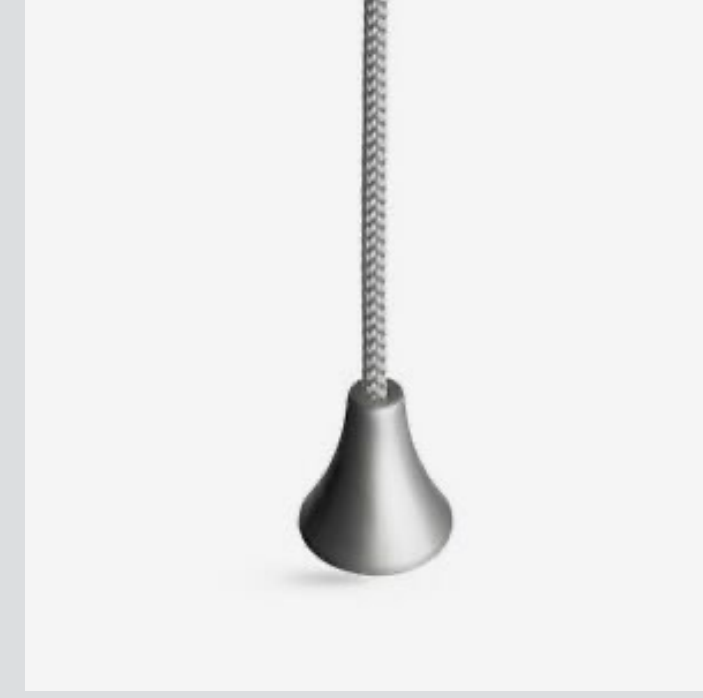


tighten



airplane toilet lock

tighten-loosen: tradeoff automation/flexibility

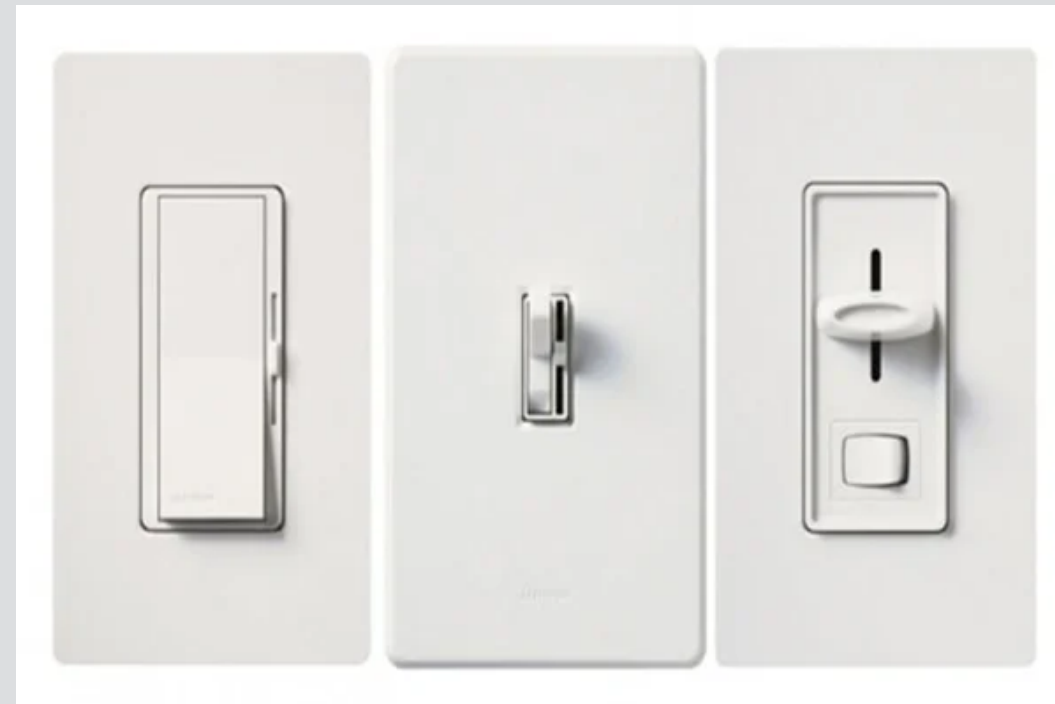


tighten



light pull / door lock

airplane toilet lock



loosen



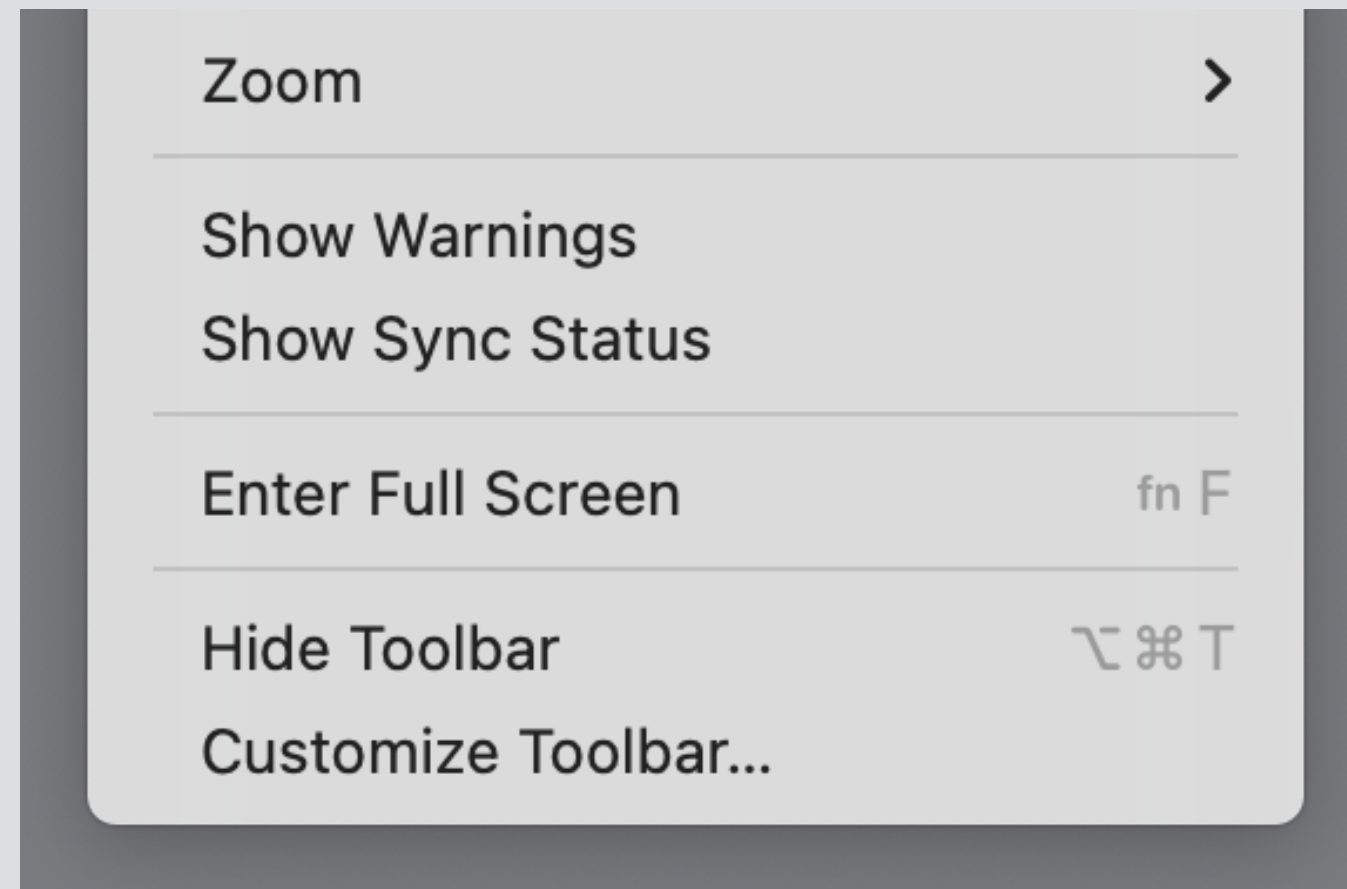
dimmers with separate controls

rotary dimmer switch

successful
design moves
in software

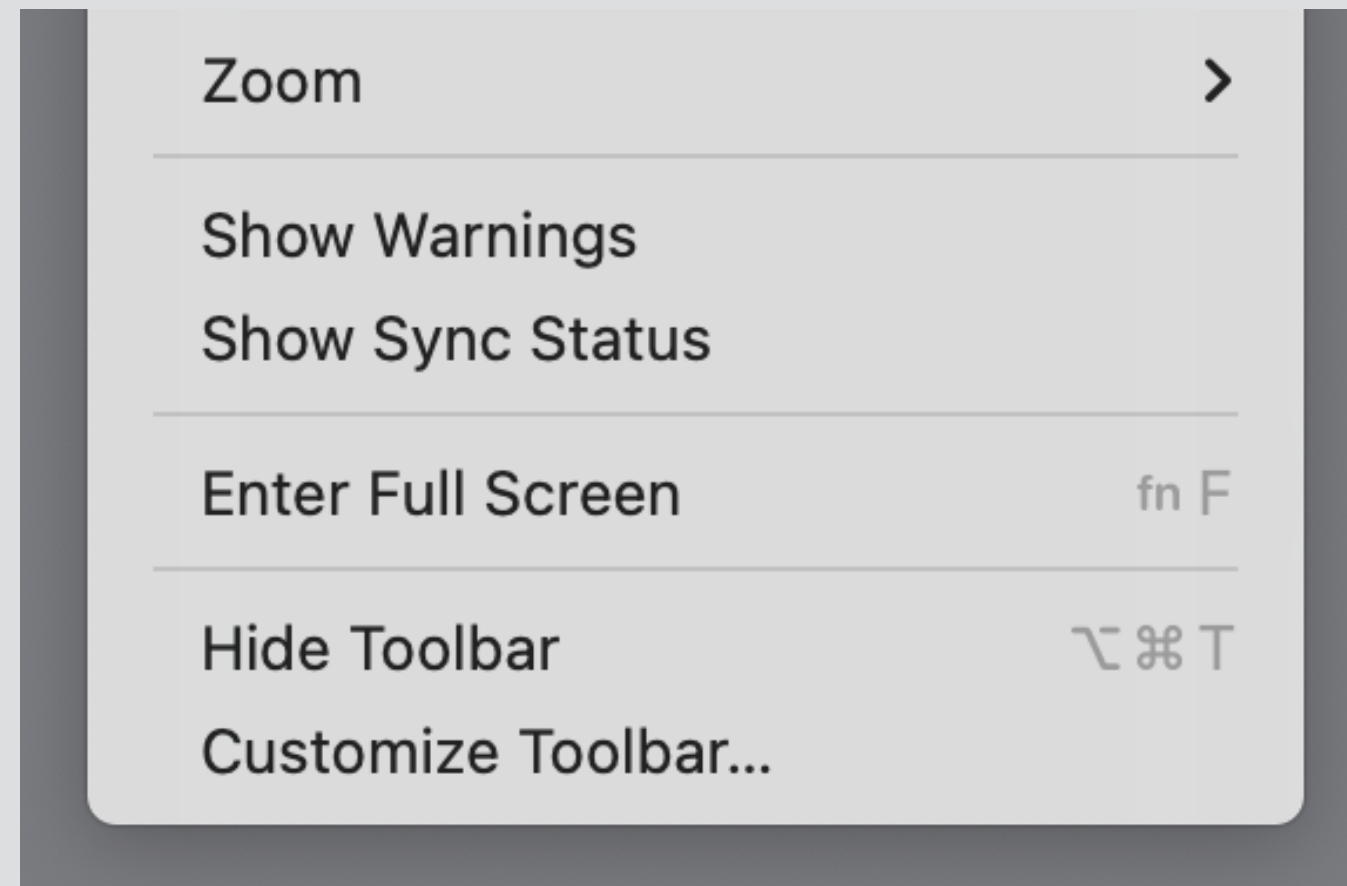
split: emergence of a concept in Keynote

split: emergence of a concept in Keynote



full screen toggle
emerges as partial concept
(c. 2010?)

split: emergence of a concept in Keynote

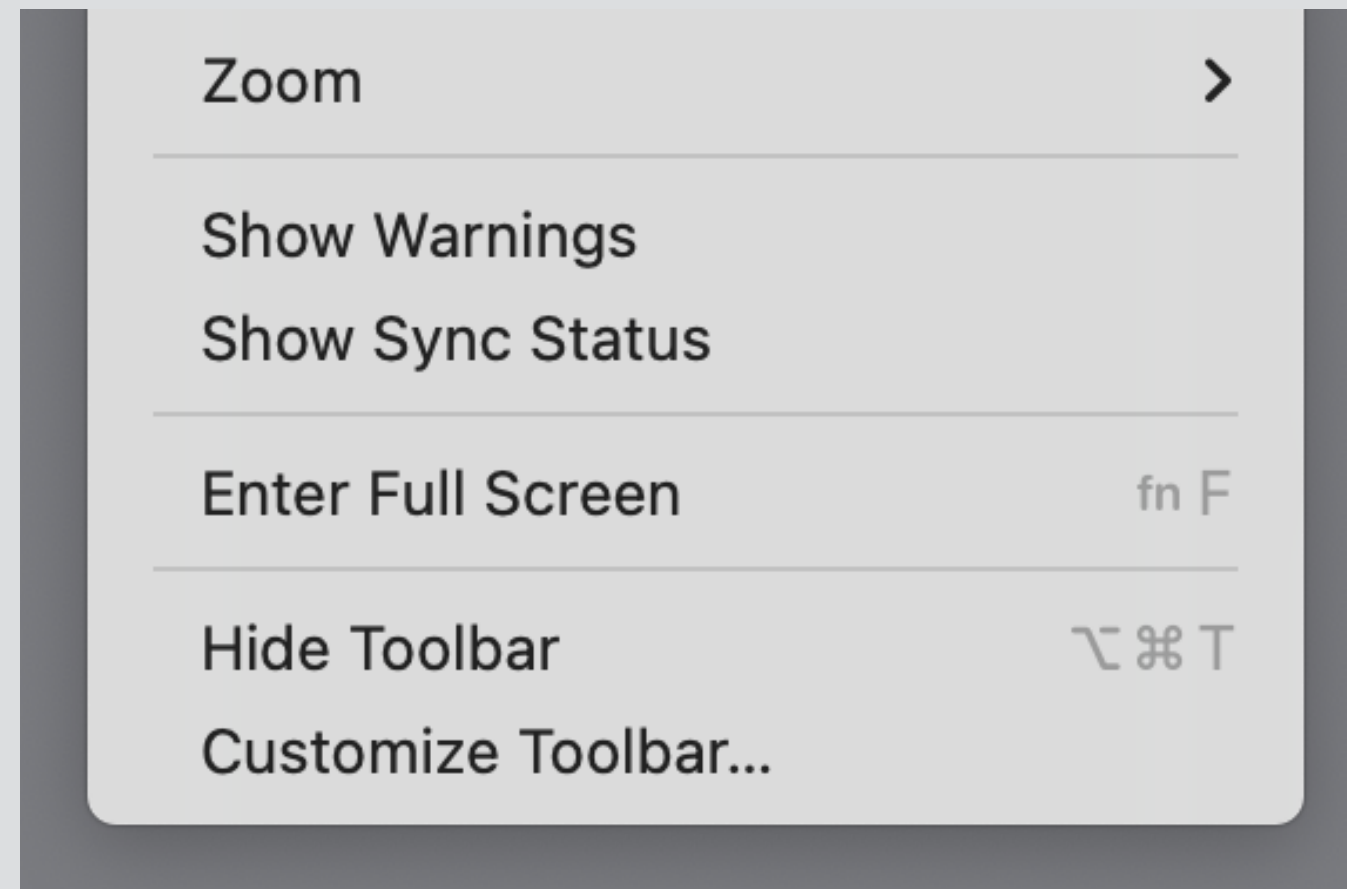


full screen toggle
emerges as partial concept
(c. 2010?)

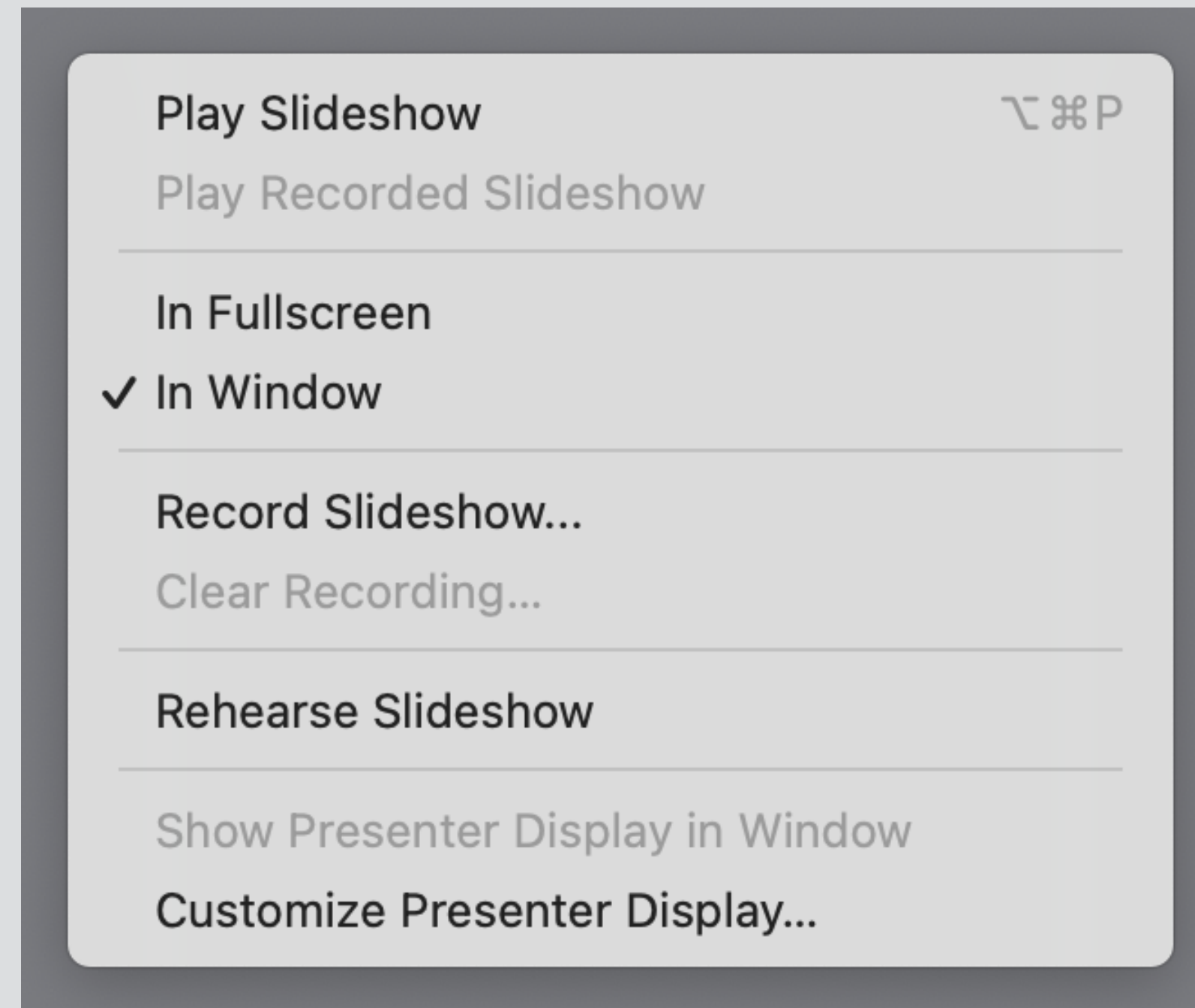


play-in-window option
now an independent concept
(2021)

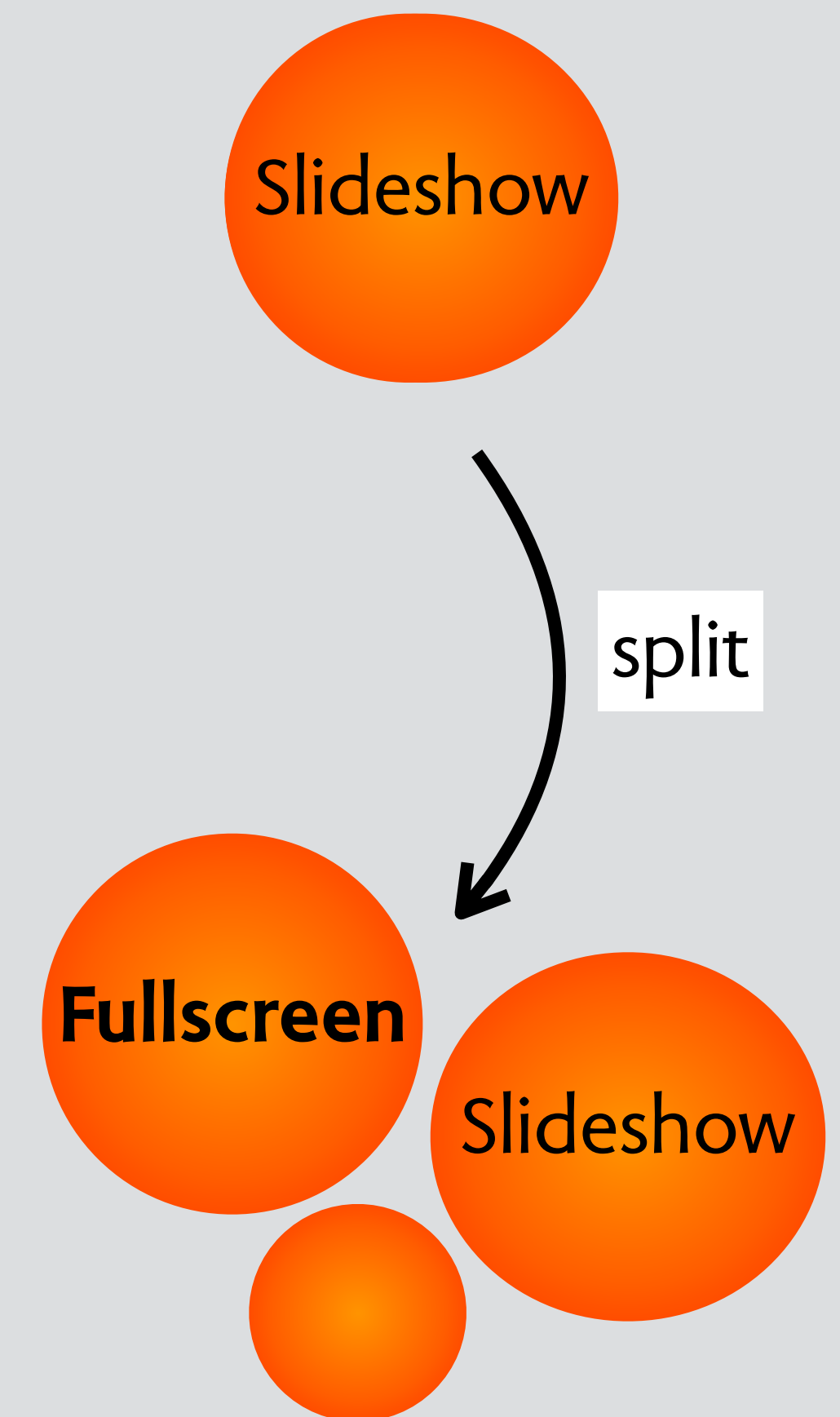
split: emergence of a concept in Keynote



full screen toggle
emerges as partial concept
(c. 2010?)



play-in-window option
now an independent concept
(2021)



merge: a simple & clever new shortening concept

yellkey

enter url and length of time for key to exist.

generate yellkey

your key is: **move.**

go to www.yellkey.com/move to use.

one concept, allows familiar words as short URLs

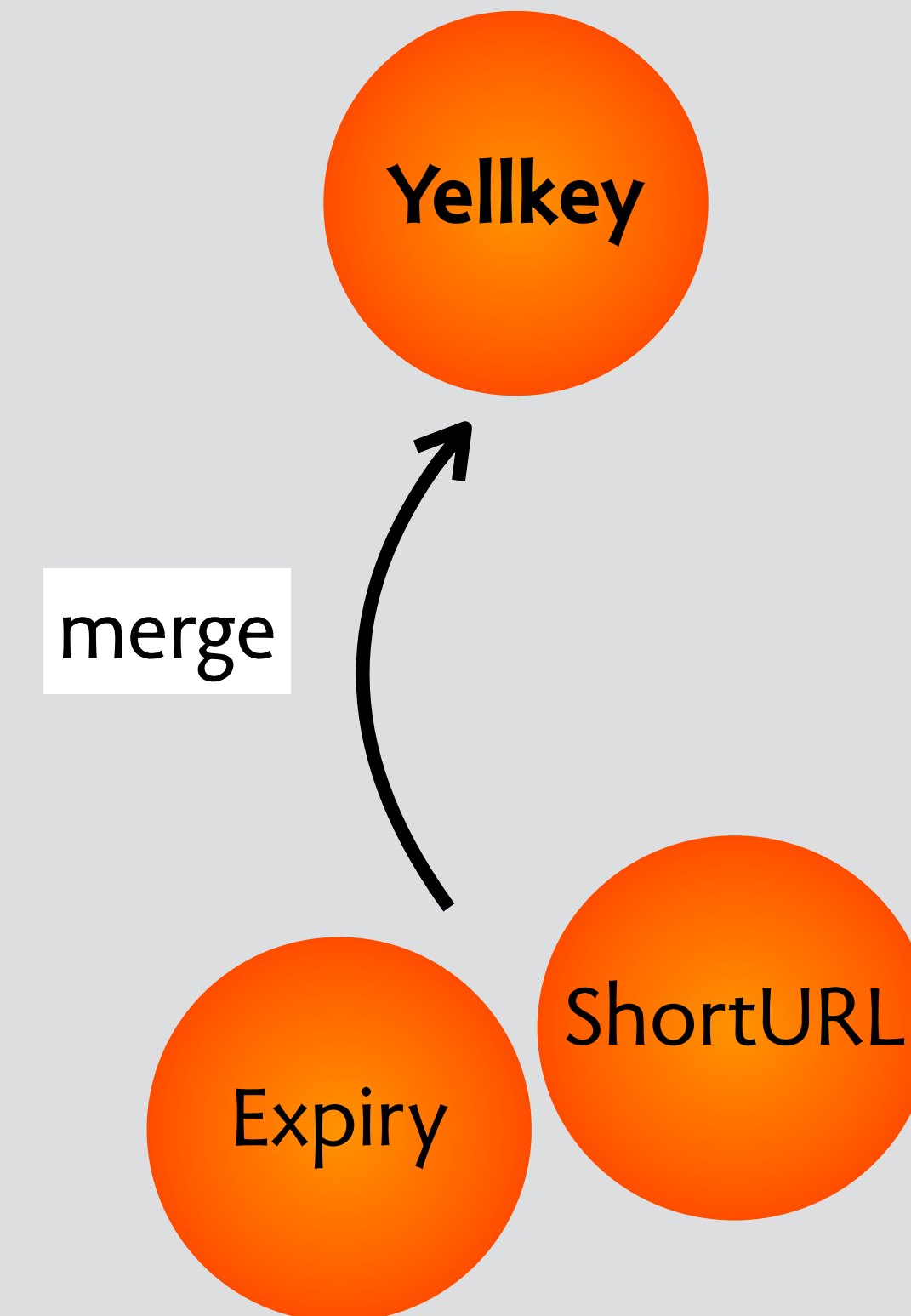
merge: a simple & clever new shortening concept

yellkey

enter url and length of time for key to exist.

your key is: **move**.

go to www.yellkey.com/move to use.



one concept, allows familiar words as short URLs

unify: subsuming access control in MIT's Moira

WebMoira List Manager : Daniel Jackson

[Help](#) | [My Lists](#) | [Undo Log \(1\)](#)

List Name: dnj-play1

Description: none

Attributes: active, moira mailing list

Permissions: private, visible

Last Modified: by dnj with moiraws on 22-mar-2022 09:39:00

[Edit](#)

Members

Add Member: [Add](#)

Leave List: [Remove Me](#)

MIT Users

Daniel Jackson (dnj) [remove](#)

Email Addresses

daniel@dnj.photo [remove](#)

Administrators

Owner: [dnj-play2 \(List\)](#)

Change Owner: [Change](#)

Add Administrator: [Add](#)

Leave Owner List: [Remove Me](#)

MIT Users

Daniel Jackson (dnj) [remove](#)

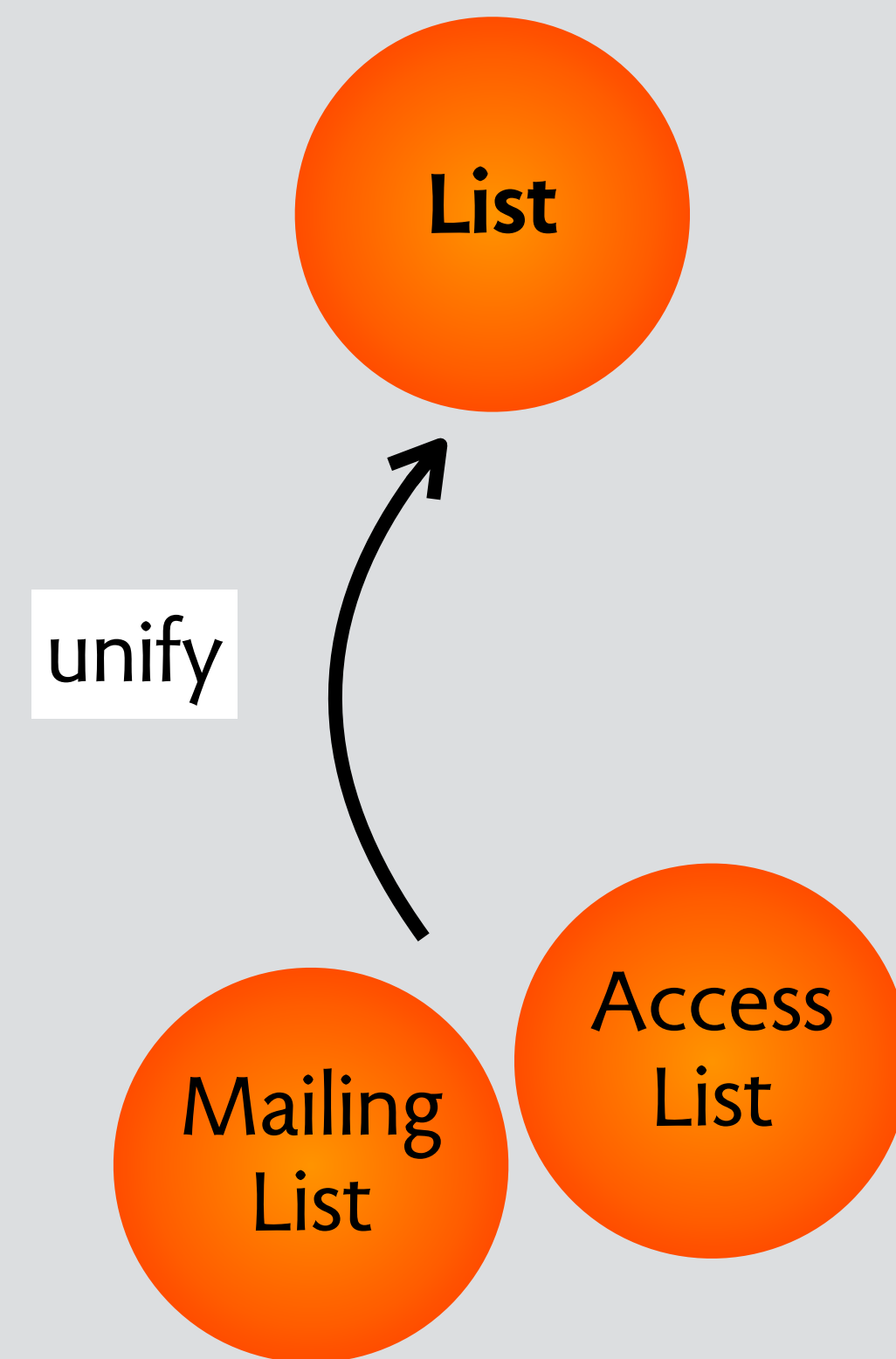
unify: subsuming access control in MIT's Moira

WebMoira List Manager : Daniel Jackson [Help](#) | [My Lists](#) | [Undo Log \(1\)](#)

List Name: dnj-play1
Description: none
Attributes: active, moira mailing list
Permissions: private, visible
Last Modified: by dnj with moiraws on 22-mar-2022 09:39:00

[Edit](#)

Members	Administrators
Add Member: <input type="text"/> Add	Owner: dnj-play2 (List)
Leave List: Remove Me	Change Owner: <input type="text"/> Change
Add Administrator: <input type="text"/> Add	Leave Owner List: Remove Me
MIT Users	MIT Users
Daniel Jackson (dnj) remove	Daniel Jackson (dnj) remove
Email Addresses	
daniel@dnj.photo remove	



unify: subsuming access control in MIT's Moira

can toggle mailing list attribute

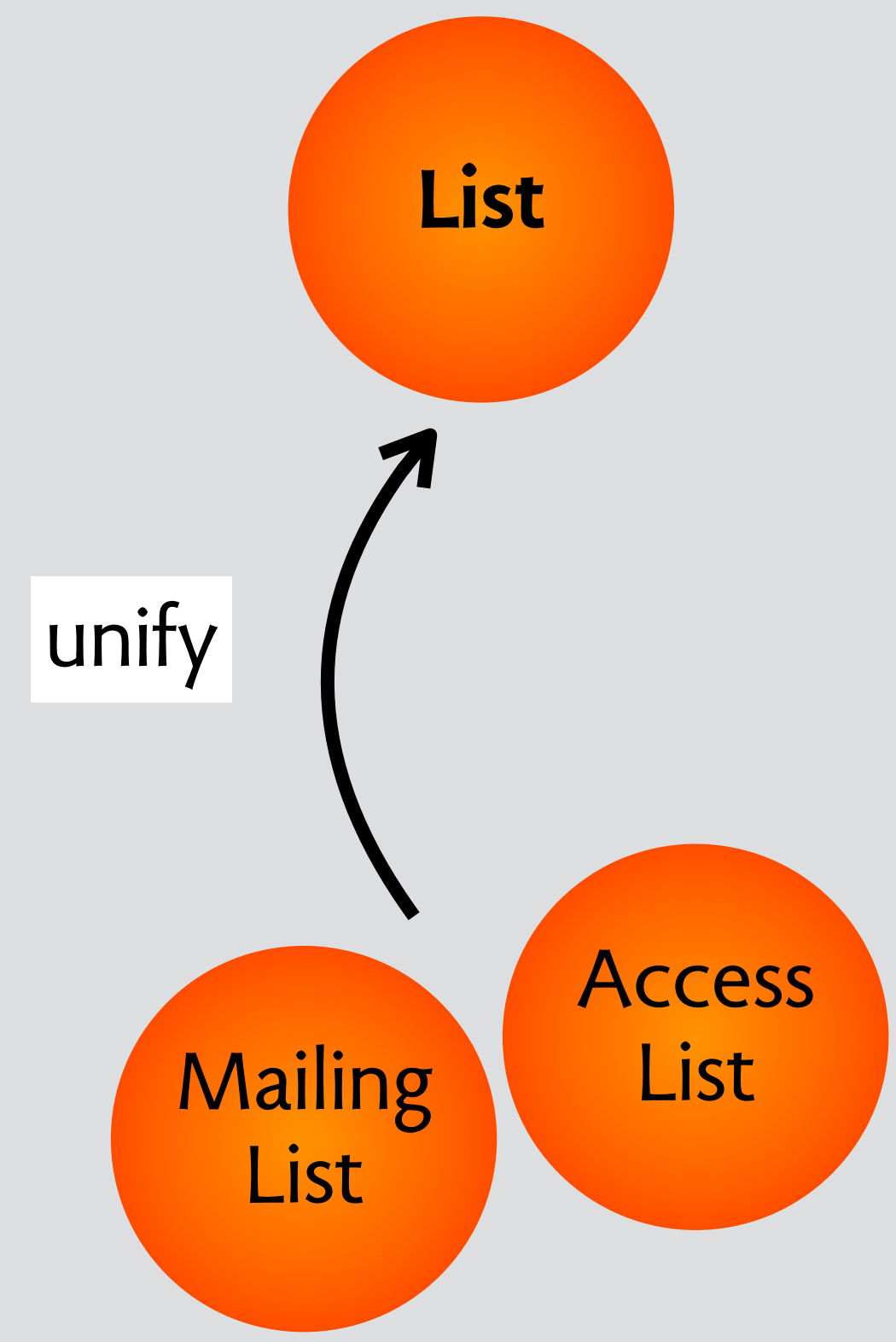
WebMoira List Manager : Daniel Jackson [Help](#) | [My Lists](#) | [Undo Log \(1\)](#)

List Name: dnj-play1
Description: none
Attributes: active, moira mailing list
Permissions: private, visible
Last Modified: by dnj with moiraws on 22-mar-2022 09:39:00

[Edit](#)

Members
Add Member: <input type="text"/> Add
Leave List: Remove Me
MIT Users
Daniel Jackson (dnj) remove
Email Addresses
daniel@dnj.photo remove

Administrators
Owner: dnj-play2 (List)
Change Owner: <input type="text"/> Change
Add Administrator: <input type="text"/> Add
Leave Owner List: Remove Me
MIT Users
Daniel Jackson (dnj) remove



unify: subsuming access control in MIT's Moira

can toggle mailing list attribute

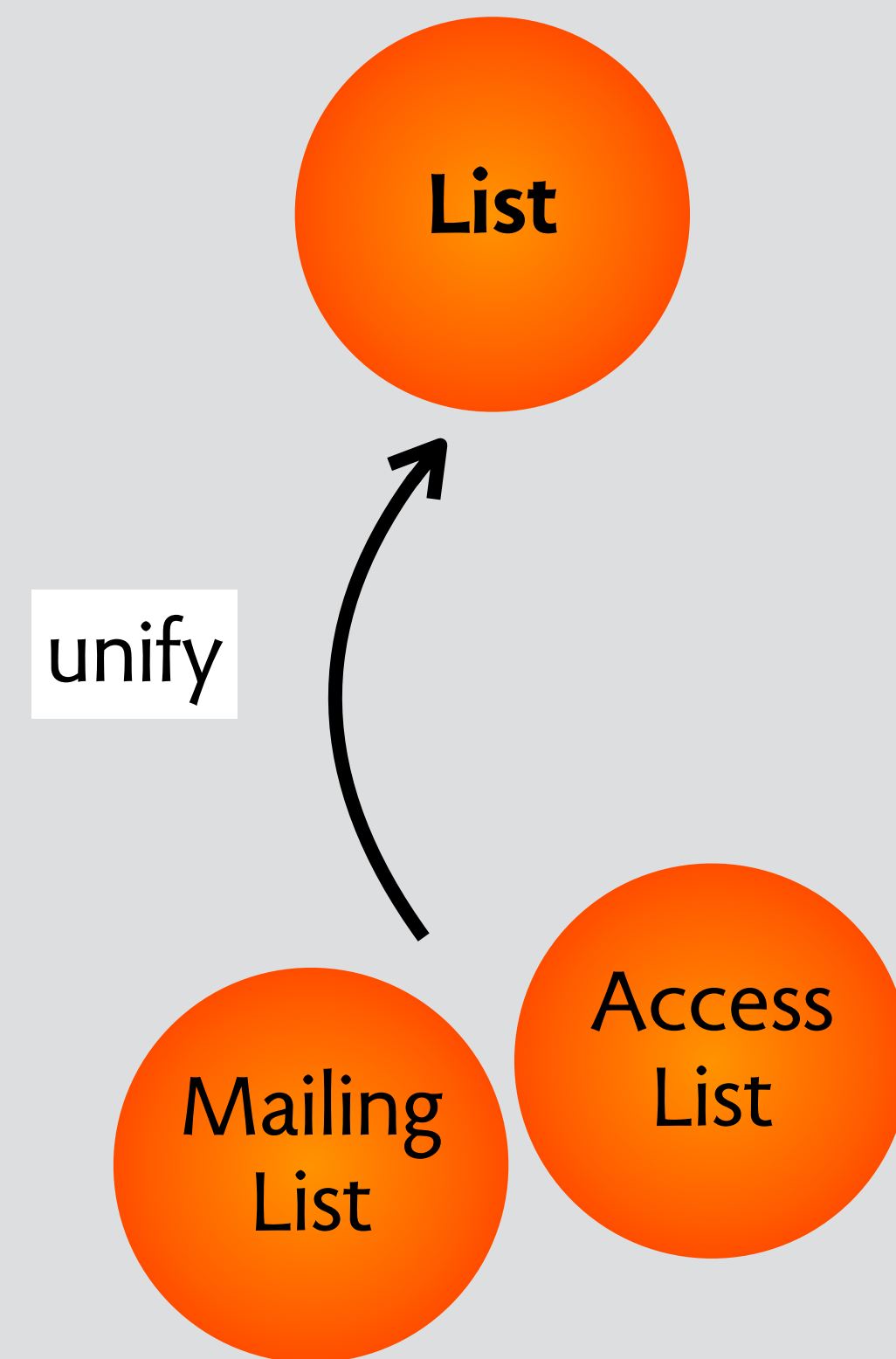
can create admin list with no login users!

WebMoira List Manager : Daniel Jackson [Help](#) | [My Lists](#) | [Undo Log \(1\)](#)

List Name: dnj-play1
Description: none
Attributes: active, moira mailing list
Permissions: private, visible
Last Modified: by dnj with moiraws on 22-mar-2022 09:39:00

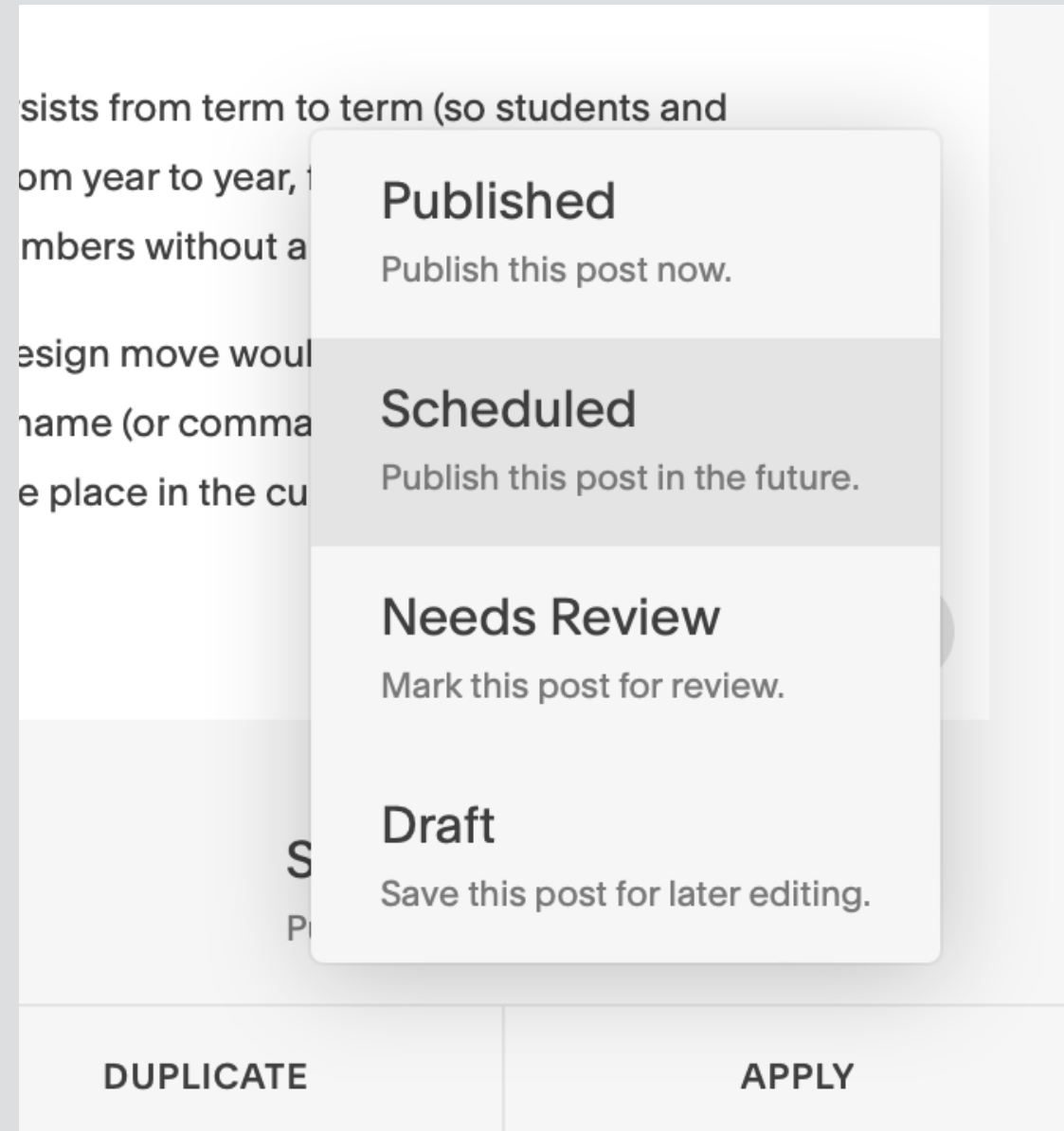
[Edit](#)

Members	Administrators
Add Member: <input type="text"/> Add	Owner: dnj-play2 (List)
Leave List: Remove Me	Change Owner: <input type="text"/> Change
MIT Users	Add Administrator: <input type="text"/> Add
Daniel Jackson (dnj) remove	Leave Owner List: Remove Me
Email Addresses	MIT Users
daniel@dnj.photo remove	Daniel Jackson (dnj) remove



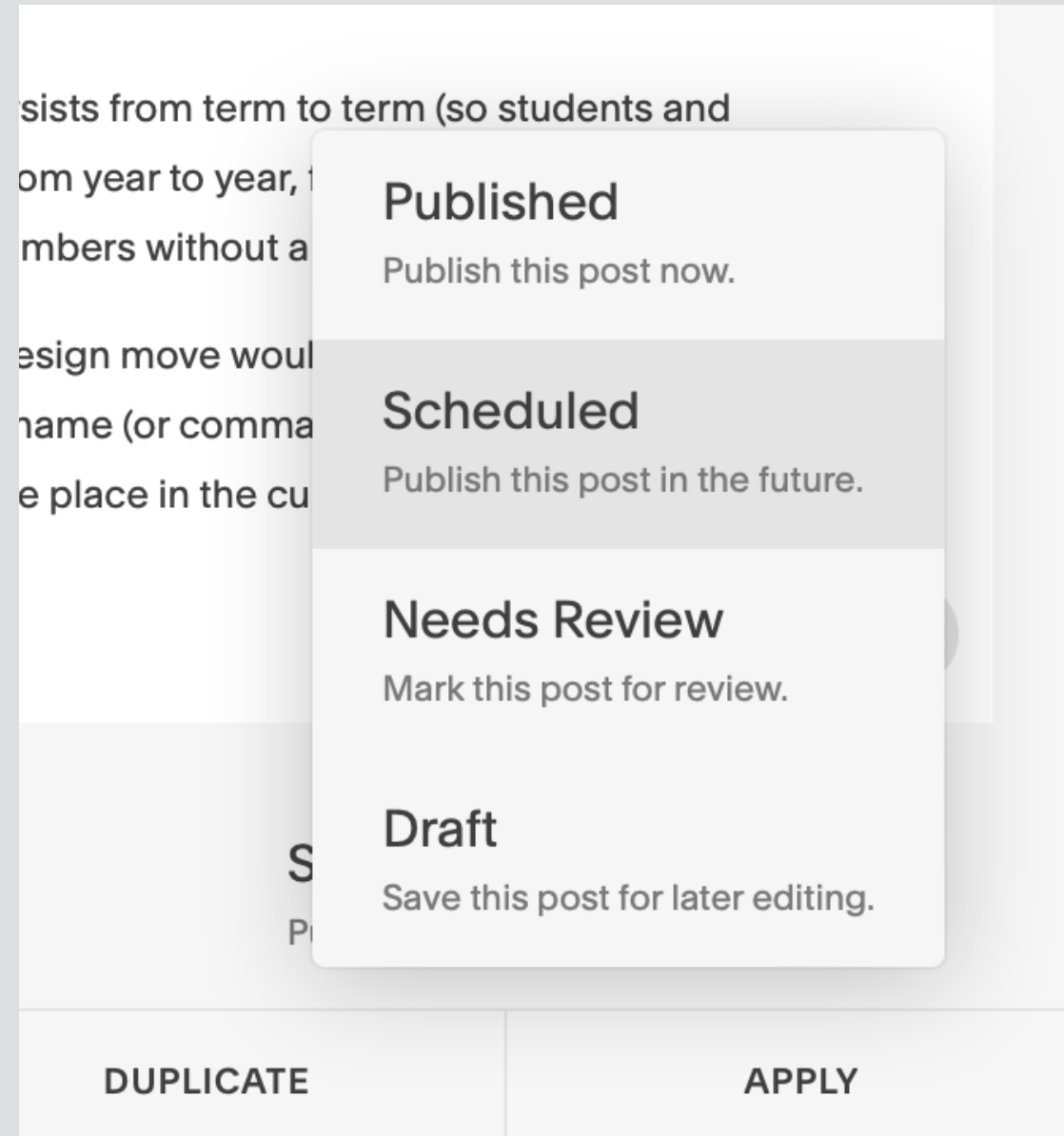
tighten: metadata and publication status in Hugo

tighten: metadata and publication status in Hugo

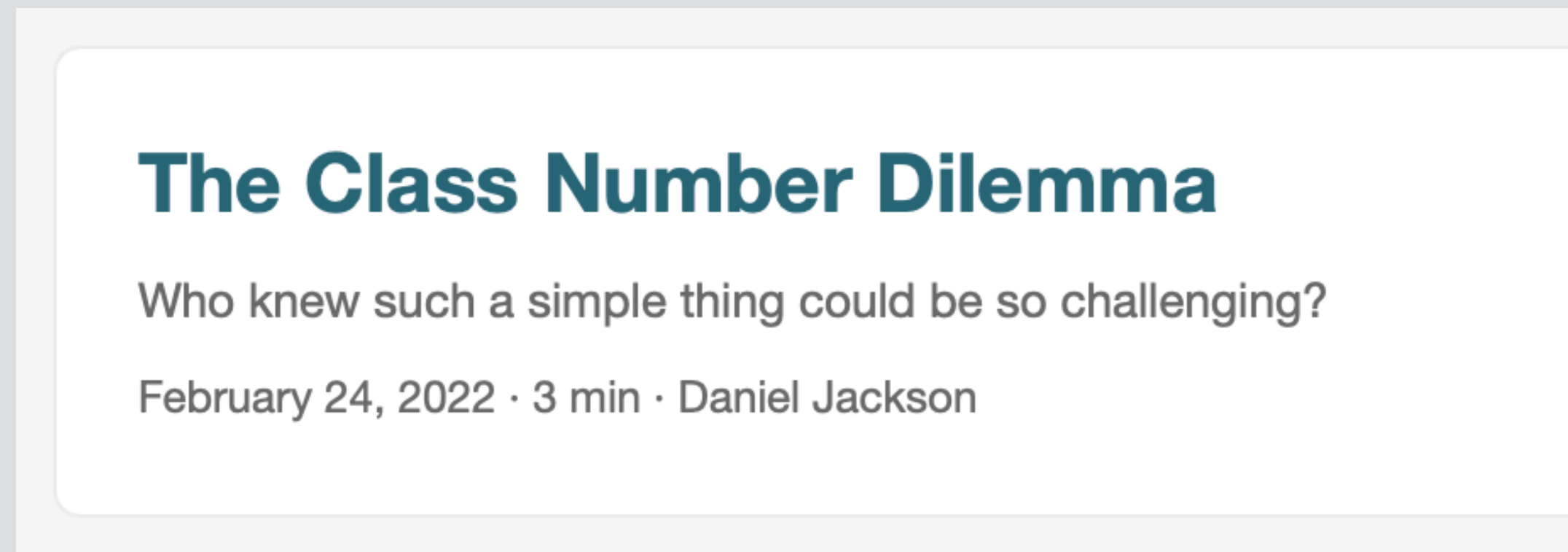


Squarespace: can schedule blog posts but not other pages, and can change pub date independently (only affects order)

tighten: metadata and publication status in Hugo



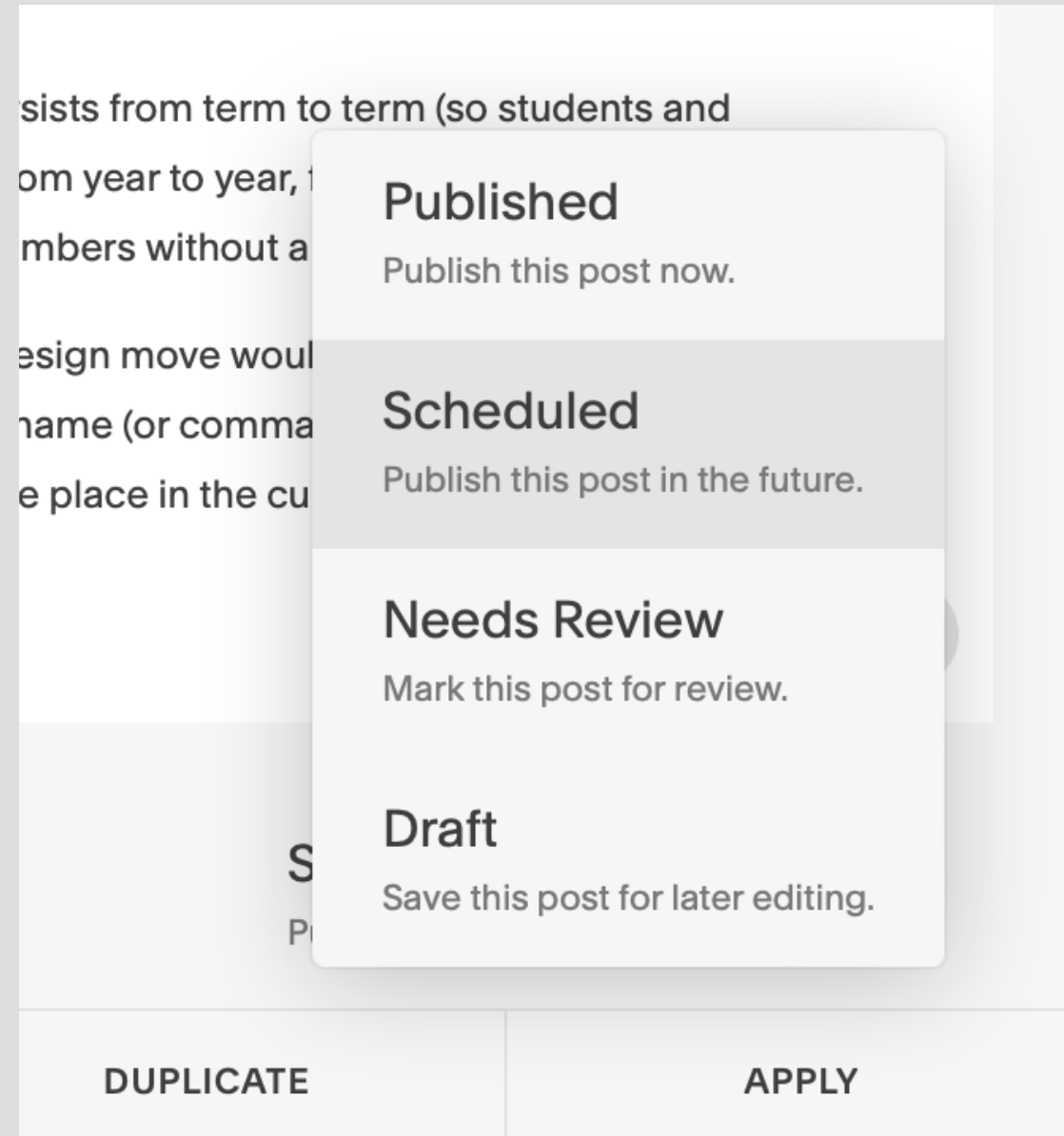
```
---
author: "Daniel Jackson"
title: "The Class Number Dilemma"
date: "2022-02-24"
description: "A Concept Example in Everyday Life"
summary: "Who knew such a simple thing could be so challenging?"
---
```



Squarespace: can schedule blog posts but not other pages, and can change pub date independently (only affects order)

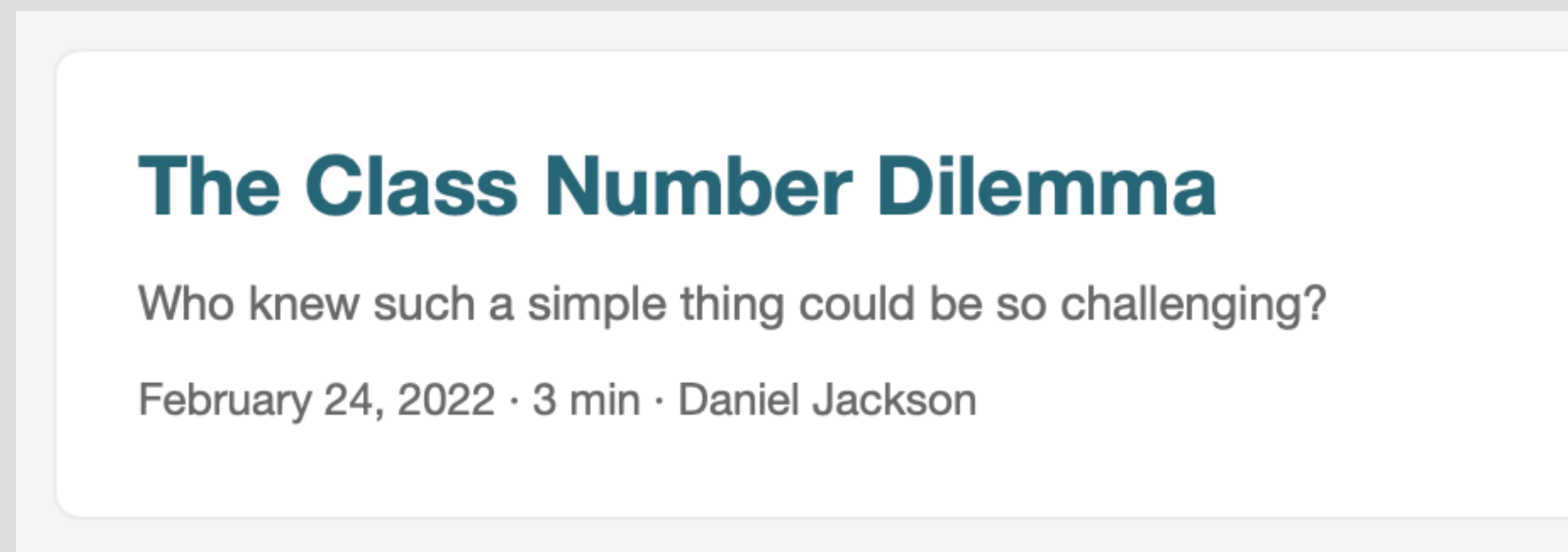
Hugo: any page can have date field to schedule, just set date in future

tighten: metadata and publication status in Hugo

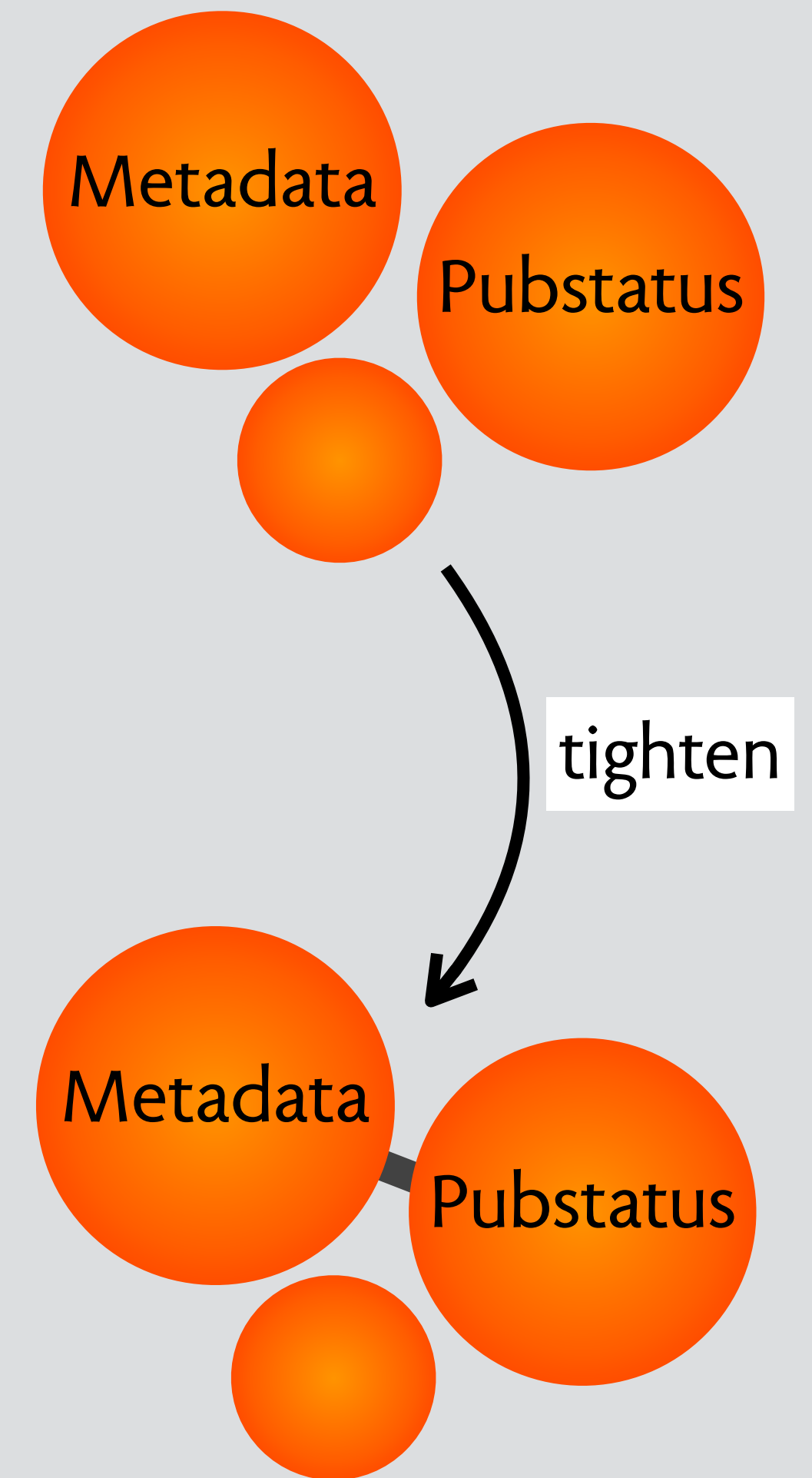


Squarespace: can schedule blog posts but not other pages, and can change pub date independently (only affects order)

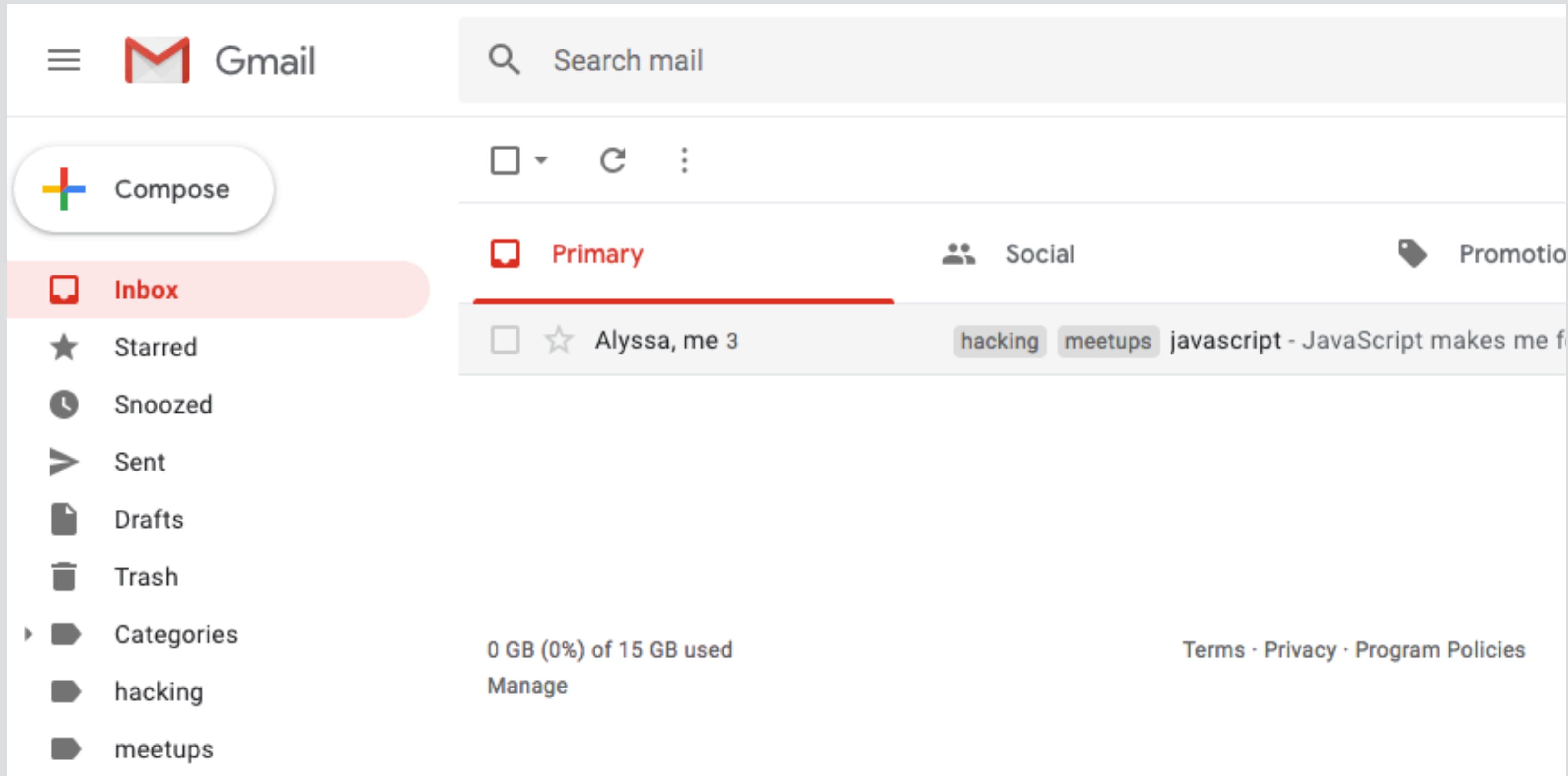
```
---
author: "Daniel Jackson"
title: "The Class Number Dilemma"
date: "2022-02-24"
description: "A Concept Example in Everyday Life"
summary: "Who knew such a simple thing could be so challenging?"
---
```



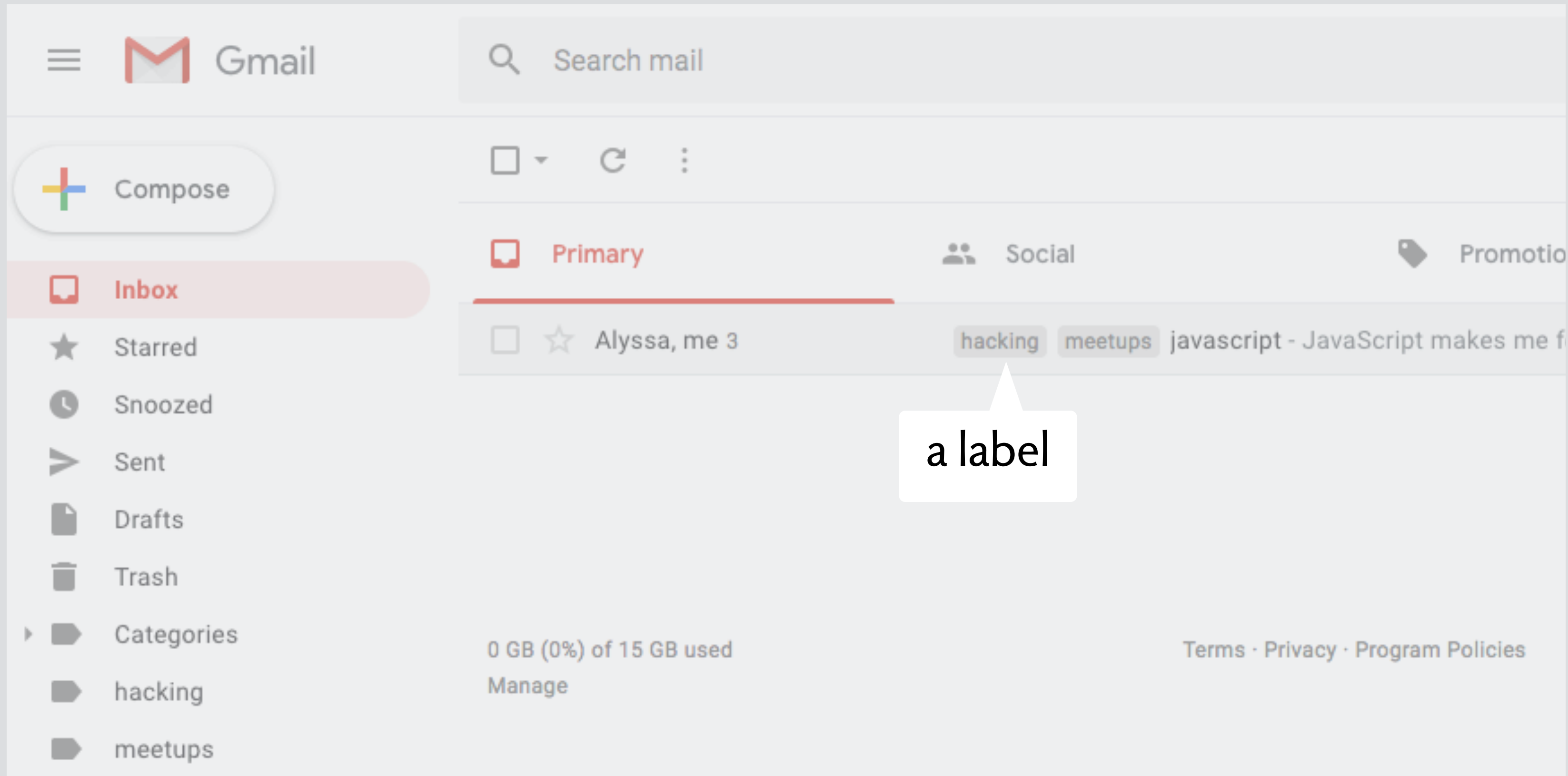
Hugo: any page can have date field to schedule, just set date in future



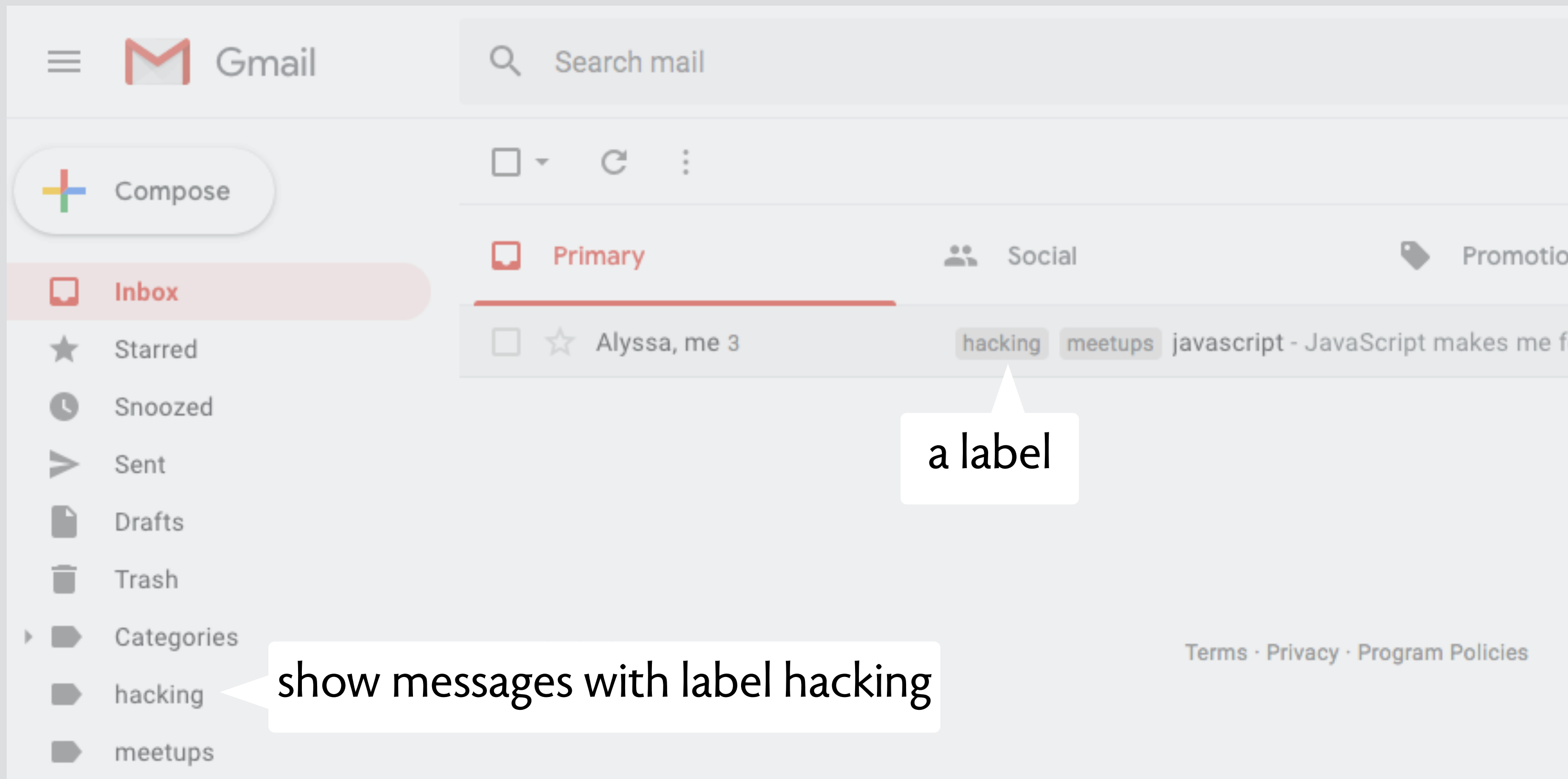
tighten: label and trash concepts in Gmail



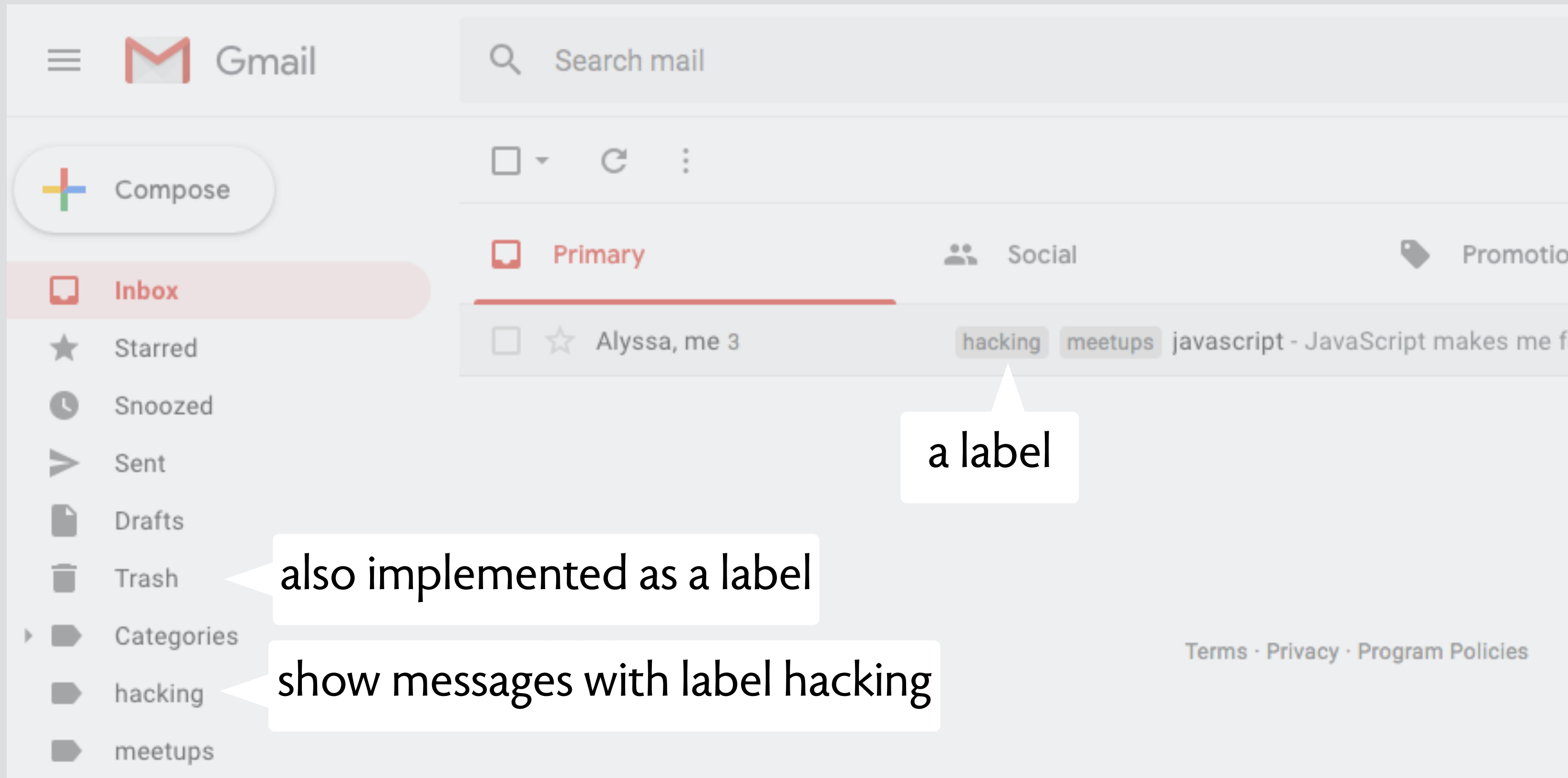
tighten: label and trash concepts in Gmail



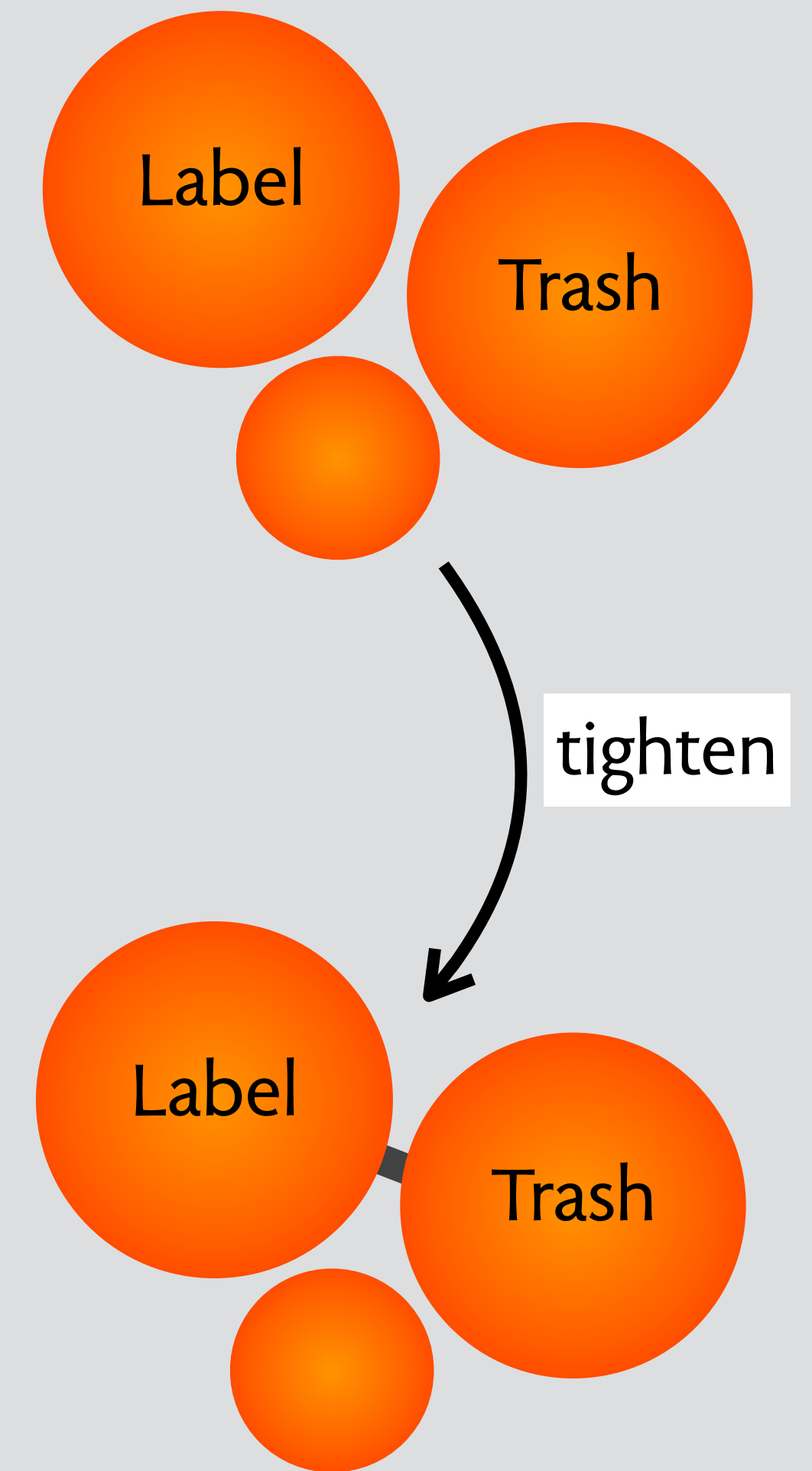
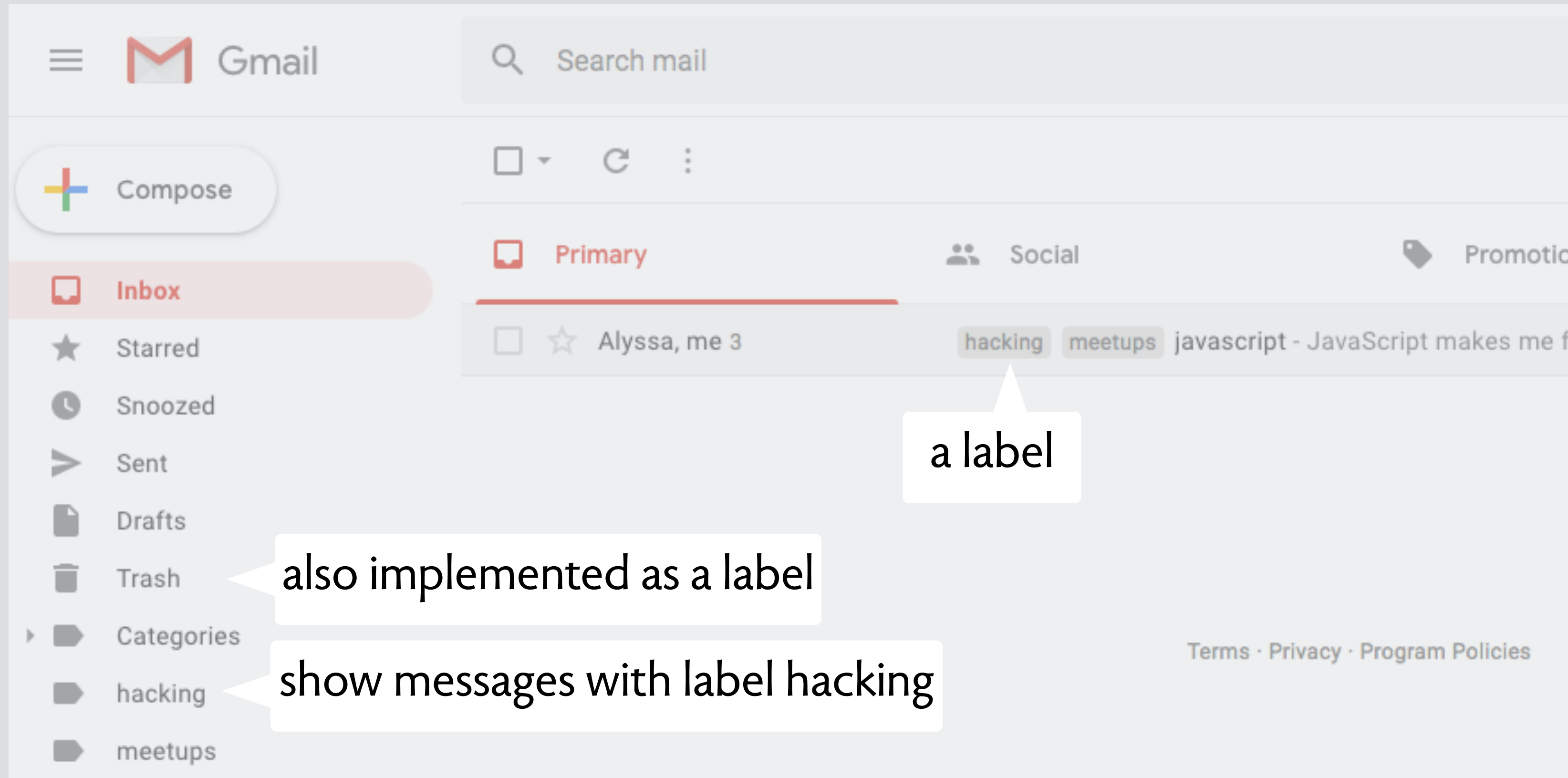
tighten: label and trash concepts in Gmail



tighten: label and trash concepts in Gmail



tighten: label and trash concepts in Gmail





concept trash

purpose undo deletion

structure

trash: **set** Item

actions

delete (i: Item)

restore (i: Item)

empty ()



concept trash

purpose undo deletion

structure

trash: **set** Item

actions

delete (i: Item)

restore (i: Item)

empty ()



concept label

purpose organize with overlapping

structure

labels: Item -> **set** Label

actions

add (i: Item, l: Label)

remove (i: Item, l: Label)

find (ls: **set** Label, **out** is: **set** Item)



concept trash

purpose undo deletion

structure

trash: **set** Item

actions

delete (i: Item)

restore (i: Item)

empty ()



concept label

purpose organize with overlapping

structure

labels: Item -> **set** Label

actions

add (i: Item, l: Label)

remove (i: Item, l: Label)

find (ls: **set** Label, **out** is: **set** Item)

when delete (i)
also add (i, 'trash')



integrating these concepts is tricky

click on
trash

The screenshot shows the Gmail Trash interface. At the top, there are several utility buttons: a checkbox with a dropdown arrow, a refresh icon, a 'More' dropdown, a page indicator '1-2 of 2', left and right navigation arrows, a keyboard icon with a dropdown arrow, and a settings gear icon with a dropdown arrow. Below these buttons is a blue link that says 'Empty Trash now' followed by a note in parentheses: '(messages that have been in Trash more than 30 days will be automatically deleted)'. The main content area contains two email entries, each with a checkbox, a trash icon, the sender's name, tags, the subject, and the time.

checkbox	trash icon	sender	tags	subject	time
<input type="checkbox"/>		me, Alyssa (13)	hacking meetups todo	javascript - Hello a	11:48 am
<input type="checkbox"/>		Andy from Google	Updates	Ben, welcome to your new Googl	9:01 am

integrating these concepts is tricky

click on
trash

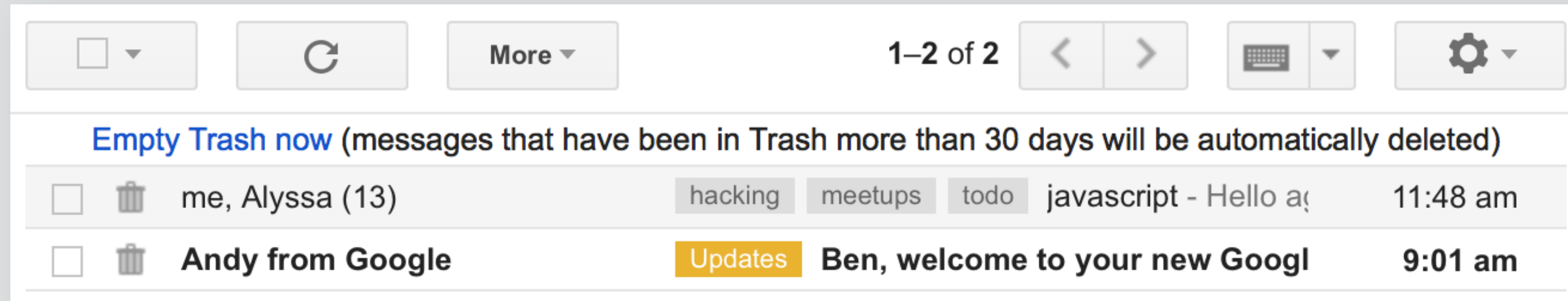
filter on
todo label

The screenshot shows a Gmail interface for the Trash folder. At the top, there are several controls: a checkbox with a dropdown arrow, a refresh button, a 'More' dropdown, a page indicator '1-2 of 2', left and right navigation arrows, a keyboard shortcut icon, and a settings gear icon. Below these controls is a blue link that says 'Empty Trash now' followed by a note in parentheses: '(messages that have been in Trash more than 30 days will be automatically deleted)'. The main area contains two email entries, each with a checkbox, a trash icon, the sender's name, a list of labels, the subject line, and the time. The first entry is from 'me, Alyssa (13)' with labels 'hacking', 'meetups', and 'todo', and subject 'javascript - Hello a', sent at '11:48 am'. The second entry is from 'Andy from Google' with a yellow 'Updates' label, subject 'Ben, welcome to your new Googl', and sent at '9:01 am'.

Checkbox	Trash Icon	Sender	Labels	Subject	Time
<input type="checkbox"/>		me, Alyssa (13)	hacking, meetups, todo	javascript - Hello a	11:48 am
<input type="checkbox"/>		Andy from Google	Updates	Ben, welcome to your new Googl	9:01 am

integrating these concepts is tricky

click on
trash

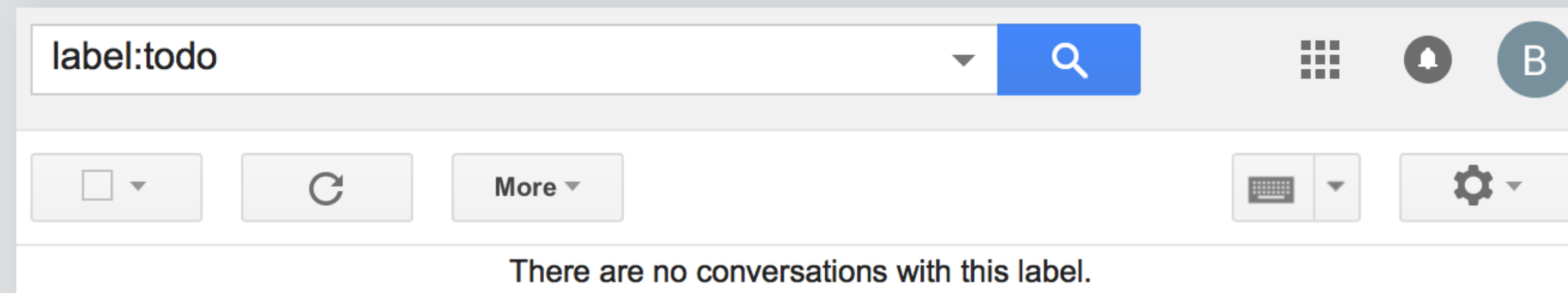


1-2 of 2

[Empty Trash now](#) (messages that have been in Trash more than 30 days will be automatically deleted)

<input type="checkbox"/>		me, Alyssa (13)	hacking	meetups	todo	javascript - Hello a	11:48 am
<input type="checkbox"/>		Andy from Google	Updates	Ben, welcome to your new Googl			9:01 am

filter on
todo label

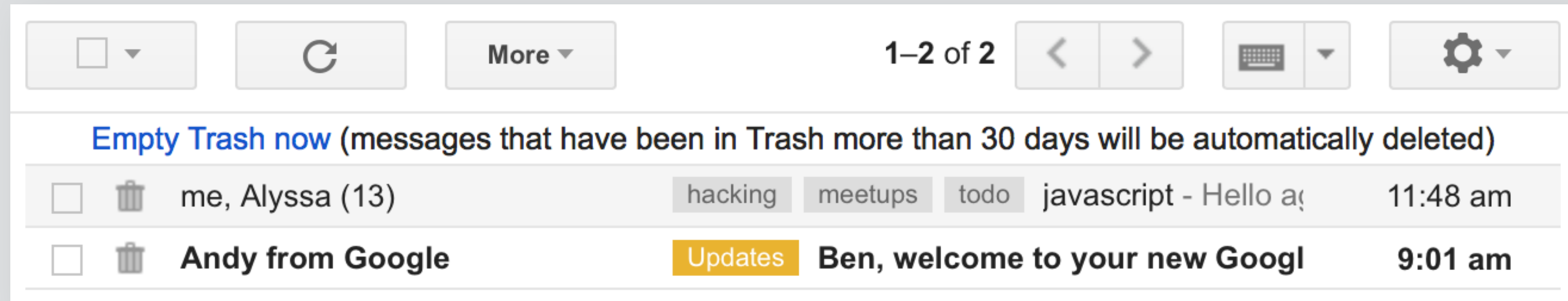


label:todo

There are no conversations with this label.

integrating these concepts is tricky

click on
trash

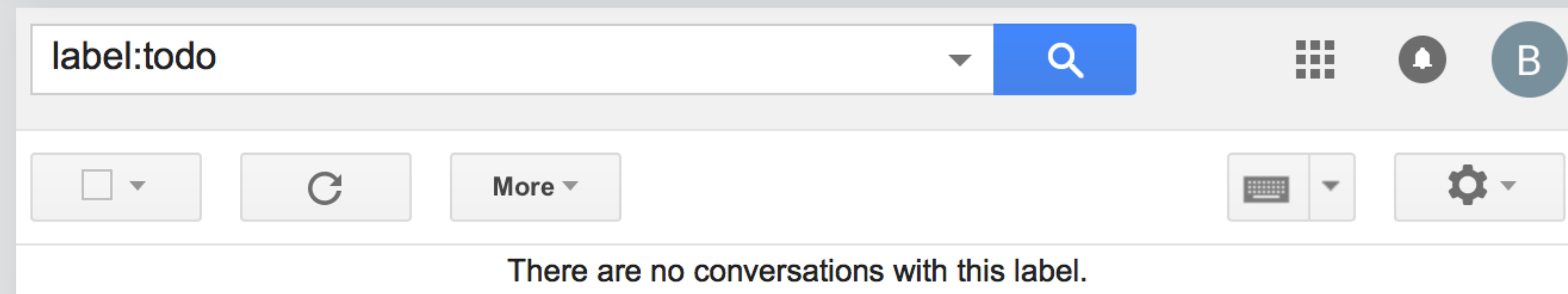


1-2 of 2

[Empty Trash now](#) (messages that have been in Trash more than 30 days will be automatically deleted)

<input type="checkbox"/>		me, Alyssa (13)	hacking	meetups	todo	javascript - Hello a	11:48 am
<input type="checkbox"/>		Andy from Google	Updates	Ben, welcome to your new Googl			9:01 am

filter on
todo label



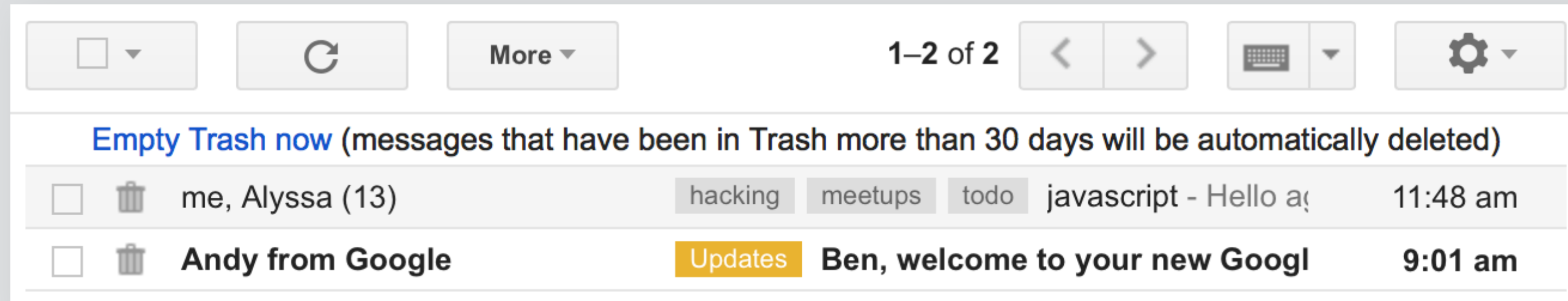
label:todo

There are no conversations with this label.

filter on
todo
and trash

integrating these concepts is tricky

click on trash



1-2 of 2

[Empty Trash now](#) (messages that have been in Trash more than 30 days will be automatically deleted)

<input type="checkbox"/>		me, Alyssa (13)	hacking	meetups	todo	javascript - Hello a	11:48 am
<input type="checkbox"/>		Andy from Google	Updates	Ben, welcome to your new Googl			9:01 am

filter on todo label

label:todo

There are no conversations with this label.

filter on todo and trash

label:todo label:trash

1-1 of 1

<input type="checkbox"/>		me, Alyssa	Trash	hacking	meetups	todo	javascript -	10:11 am
--------------------------	--	------------	-------	---------	---------	------	--------------	----------

integrating these concepts is tricky

click on trash

1-2 of 2

[Empty Trash now](#) (messages that have been in Trash more than 30 days will be automatically deleted)

<input type="checkbox"/>		me, Alyssa (13)	hacking	meetups	todo	javascript - Hello a	11:48 am
<input type="checkbox"/>		Andy from Google	Updates	Ben, welcome to your new Googl			9:01 am

filter on todo label

label:todo

There are no conversations with this label.

filter on todo and trash

label:todo label:trash

1-1 of 1

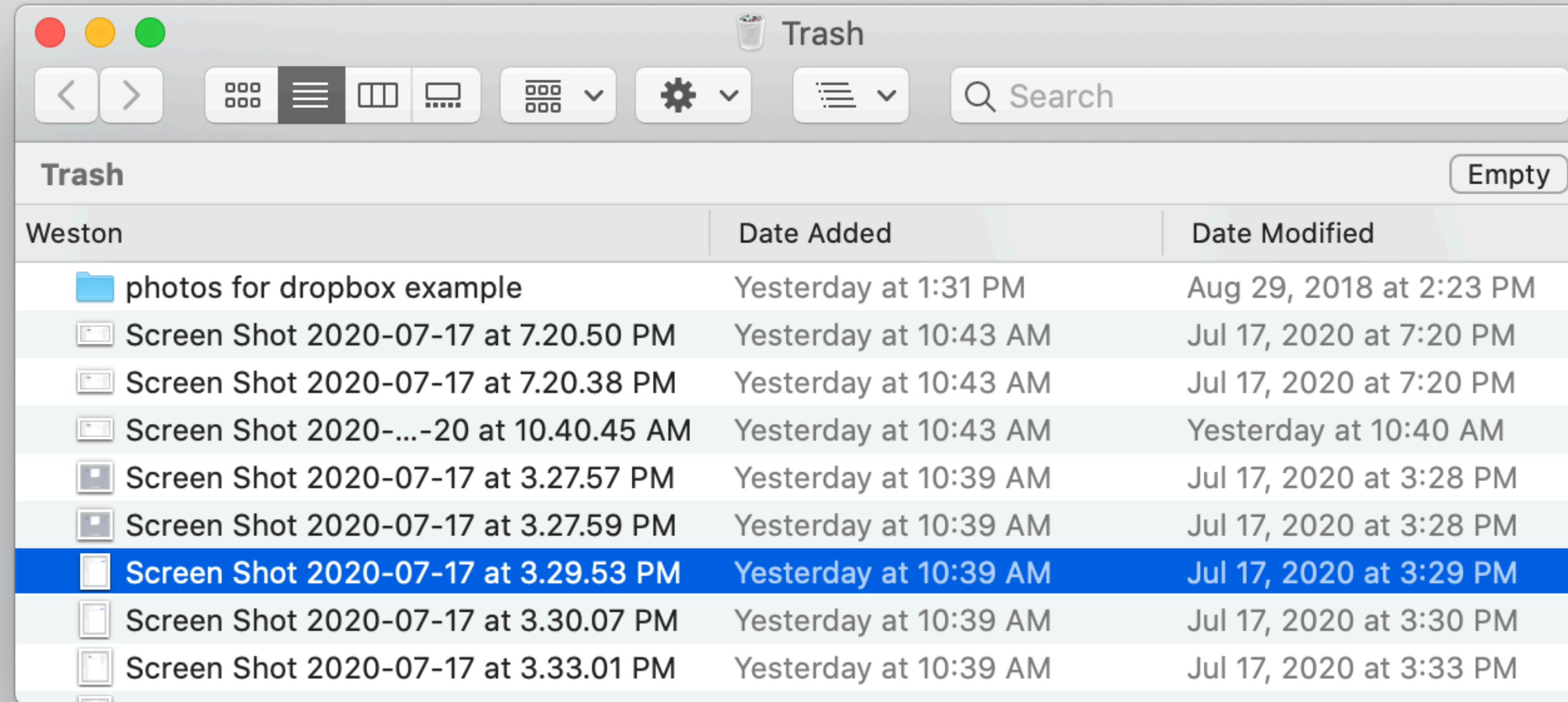
<input type="checkbox"/>		me, Alyssa	Trash	hacking	meetups	todo	javascript -	10:11 am
--------------------------	--	------------	-------	---------	---------	------	--------------	----------

filter on something else

label:todo OR label:meetup

Some messages in Trash or Spam match your search. [View messages.](#)

a beautiful (but tricky) synergy



a beautiful (but tricky) synergy

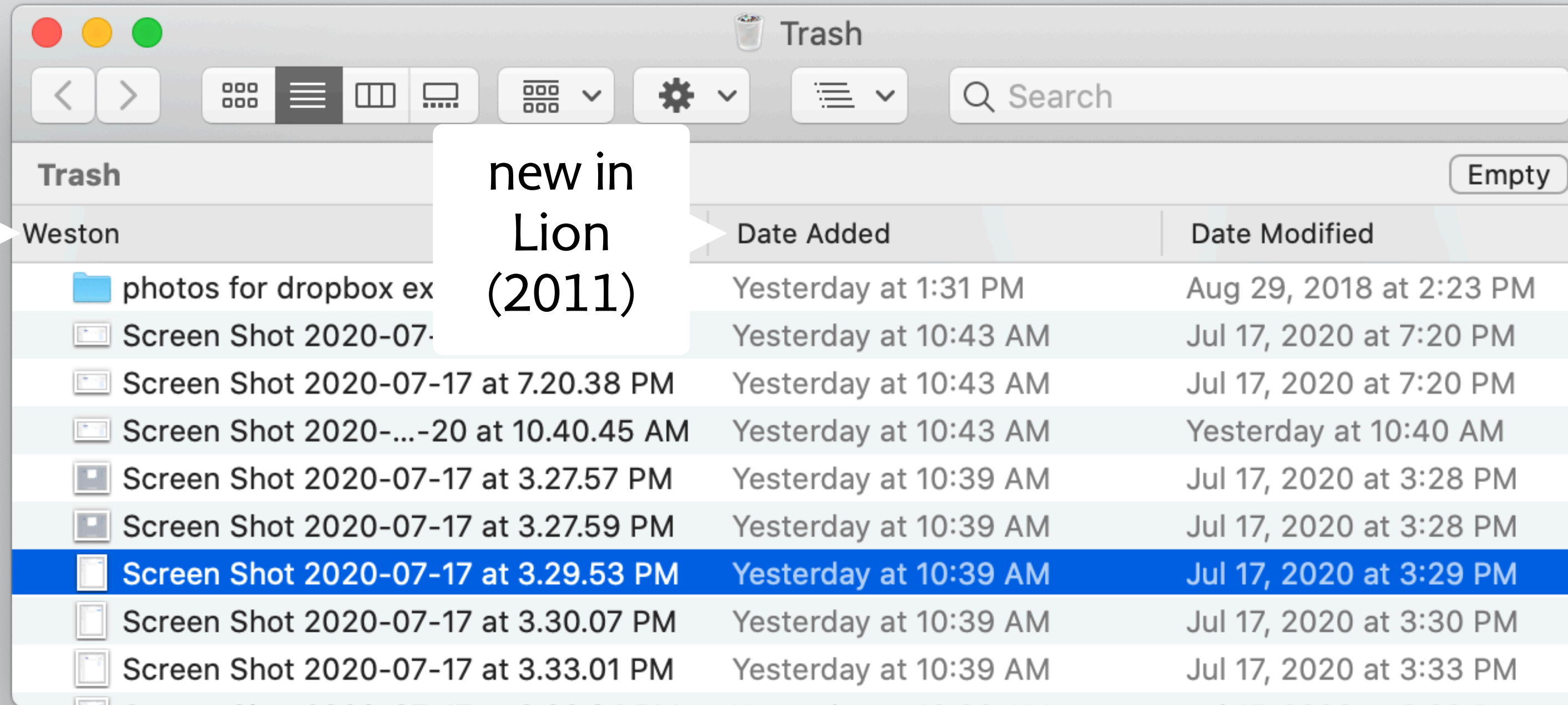
folder
sortable by
volume!

Weston	Date Added	Date Modified
photos for dropbox example	Yesterday at 1:31 PM	Aug 29, 2018 at 2:23 PM
Screen Shot 2020-07-17 at 7.20.50 PM	Yesterday at 10:43 AM	Jul 17, 2020 at 7:20 PM
Screen Shot 2020-07-17 at 7.20.38 PM	Yesterday at 10:43 AM	Jul 17, 2020 at 7:20 PM
Screen Shot 2020-...-20 at 10.40.45 AM	Yesterday at 10:43 AM	Yesterday at 10:40 AM
Screen Shot 2020-07-17 at 3.27.57 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:28 PM
Screen Shot 2020-07-17 at 3.27.59 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:28 PM
Screen Shot 2020-07-17 at 3.29.53 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:29 PM
Screen Shot 2020-07-17 at 3.30.07 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:30 PM
Screen Shot 2020-07-17 at 3.33.01 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:33 PM

a beautiful (but tricky) synergy

folder
sortable by
volume!

new in
Lion
(2011)



a beautiful (but tricky) synergy

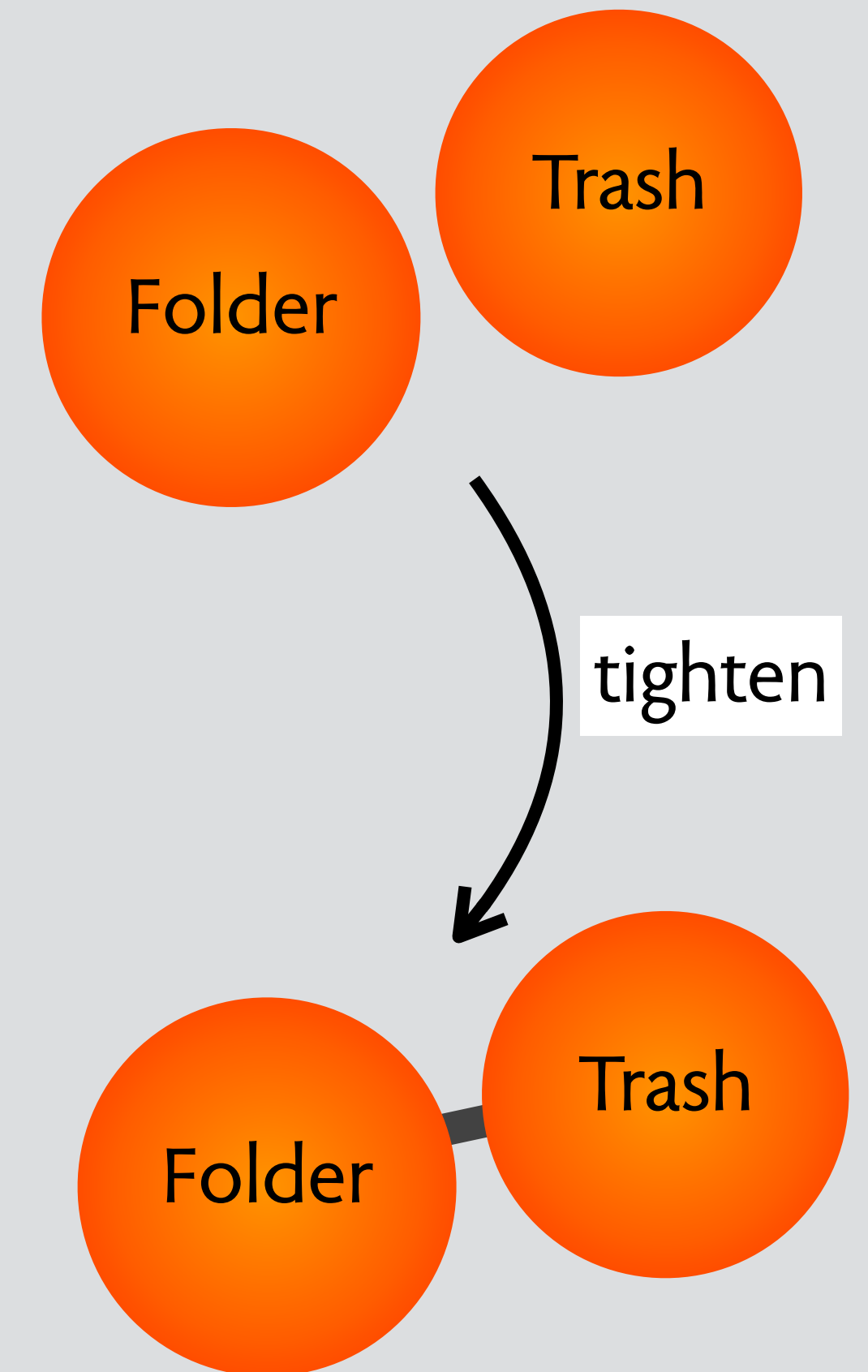
folder
sortable by
volume!

Trash Empty

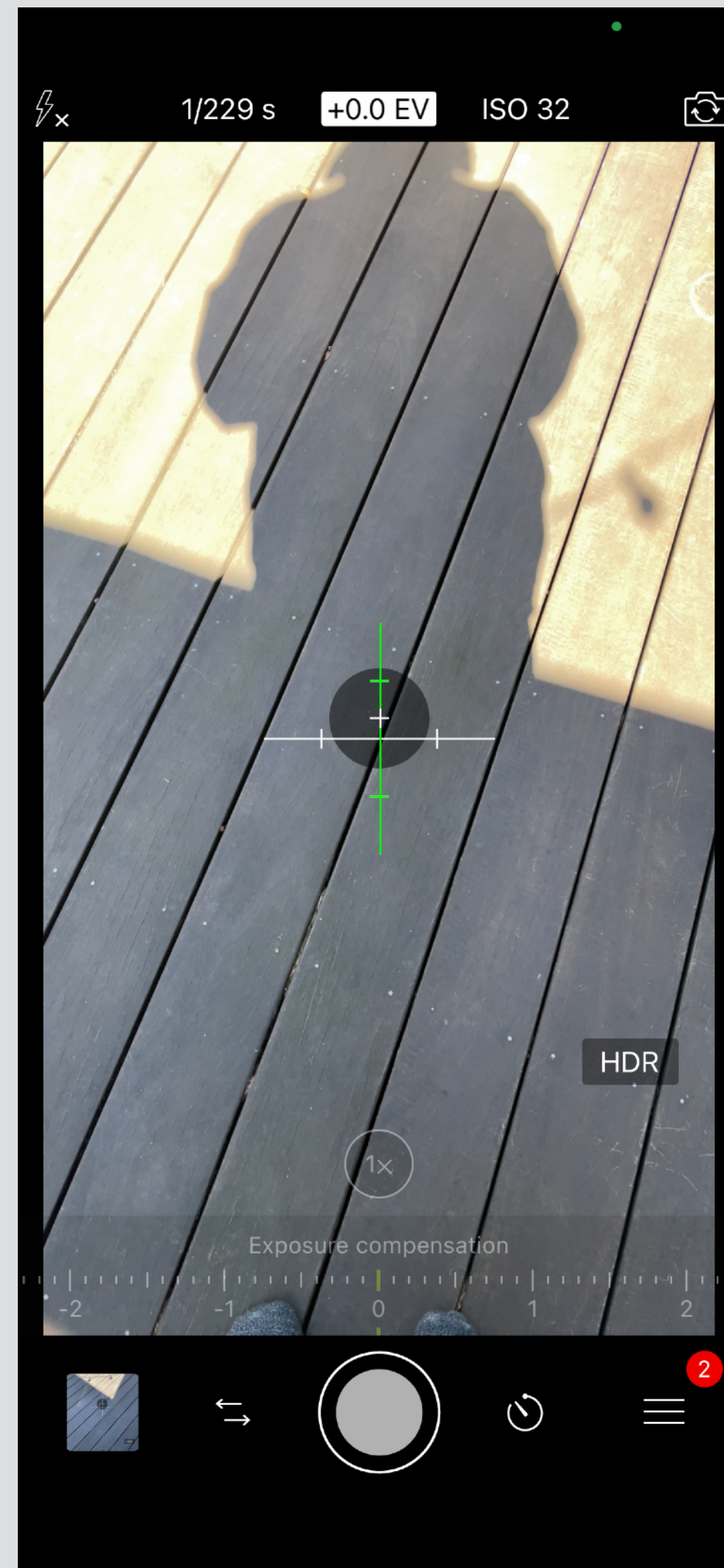
Weston

	Date Added	Date Modified
photos for dropbox ex	Yesterday at 1:31 PM	Aug 29, 2018 at 2:23 PM
Screen Shot 2020-07-	Yesterday at 10:43 AM	Jul 17, 2020 at 7:20 PM
Screen Shot 2020-07-17 at 7.20.38 PM	Yesterday at 10:43 AM	Jul 17, 2020 at 7:20 PM
Screen Shot 2020-...-20 at 10.40.45 AM	Yesterday at 10:43 AM	Yesterday at 10:40 AM
Screen Shot 2020-07-17 at 3.27.57 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:28 PM
Screen Shot 2020-07-17 at 3.27.59 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:28 PM
Screen Shot 2020-07-17 at 3.29.53 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:29 PM
Screen Shot 2020-07-17 at 3.30.07 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:30 PM
Screen Shot 2020-07-17 at 3.33.01 PM	Yesterday at 10:39 AM	Jul 17, 2020 at 3:33 PM

new in
Lion
(2011)

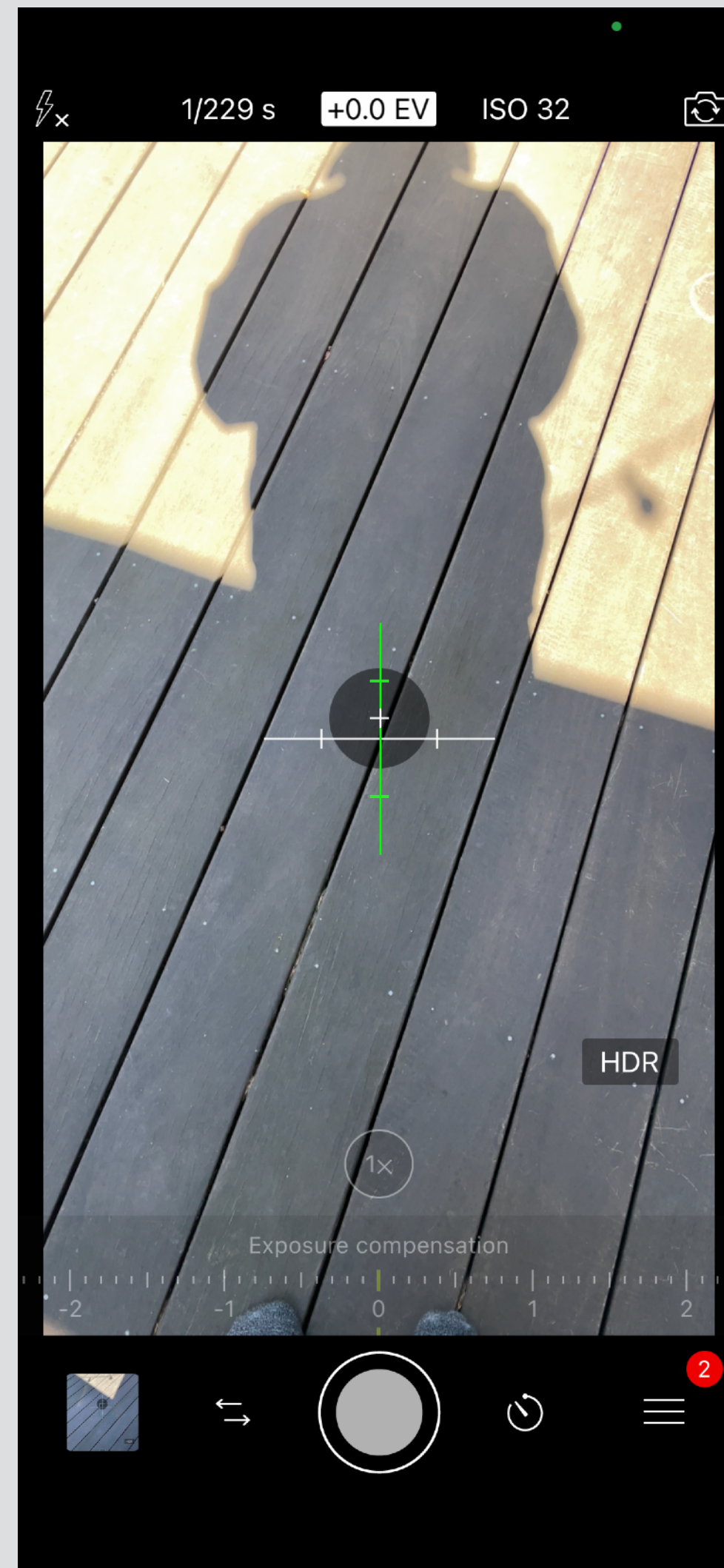


loosen: expert control in ProCamera

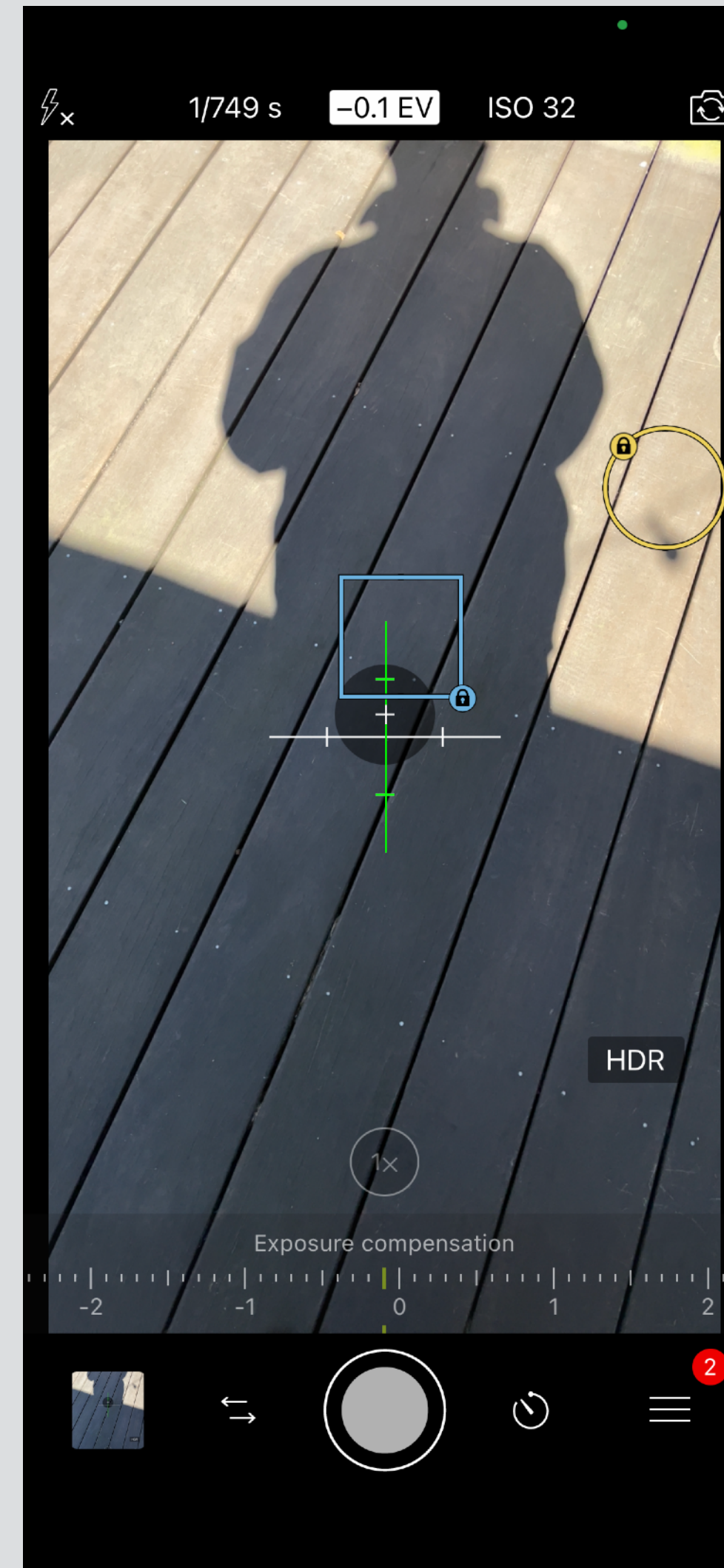


focus and exposure
points aligned

loosen: expert control in ProCamera

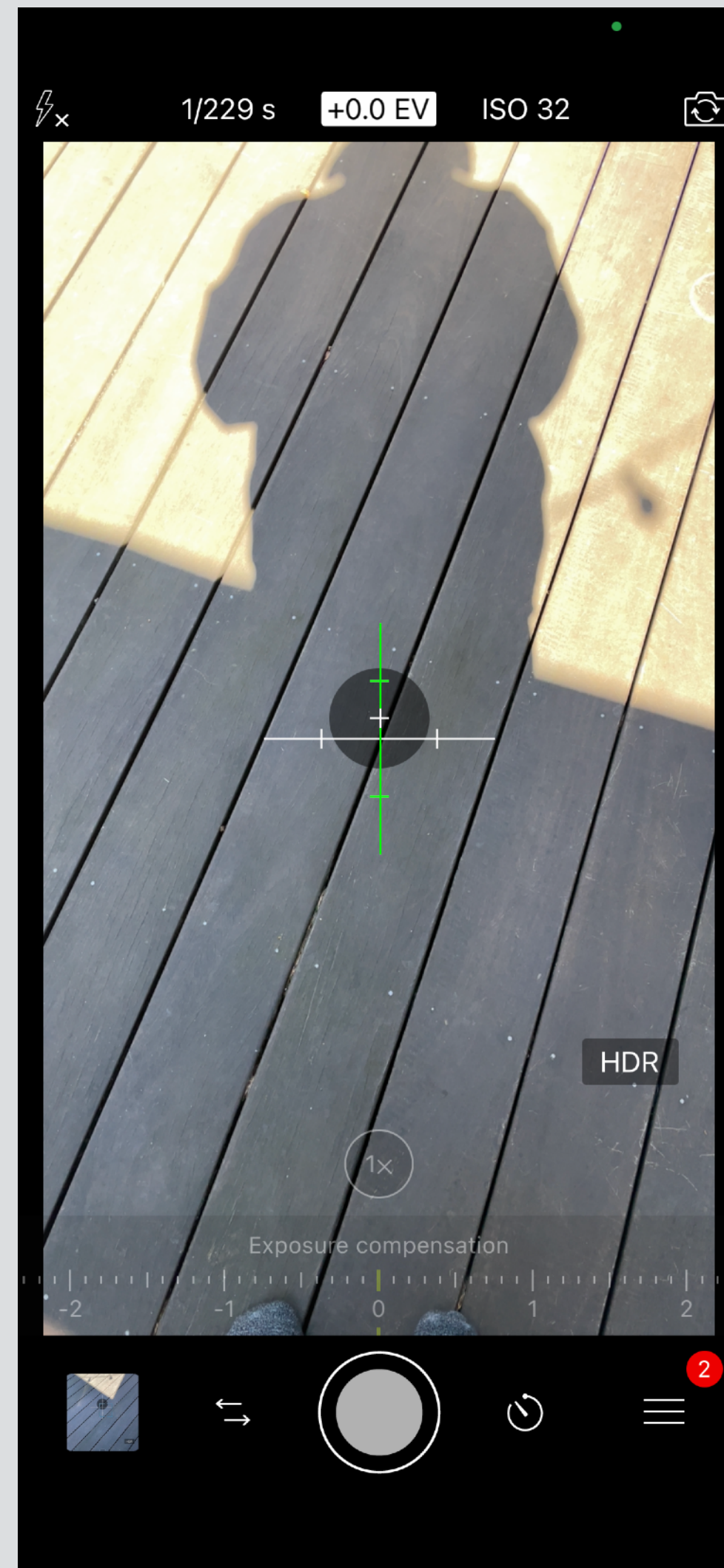


focus and exposure
points aligned

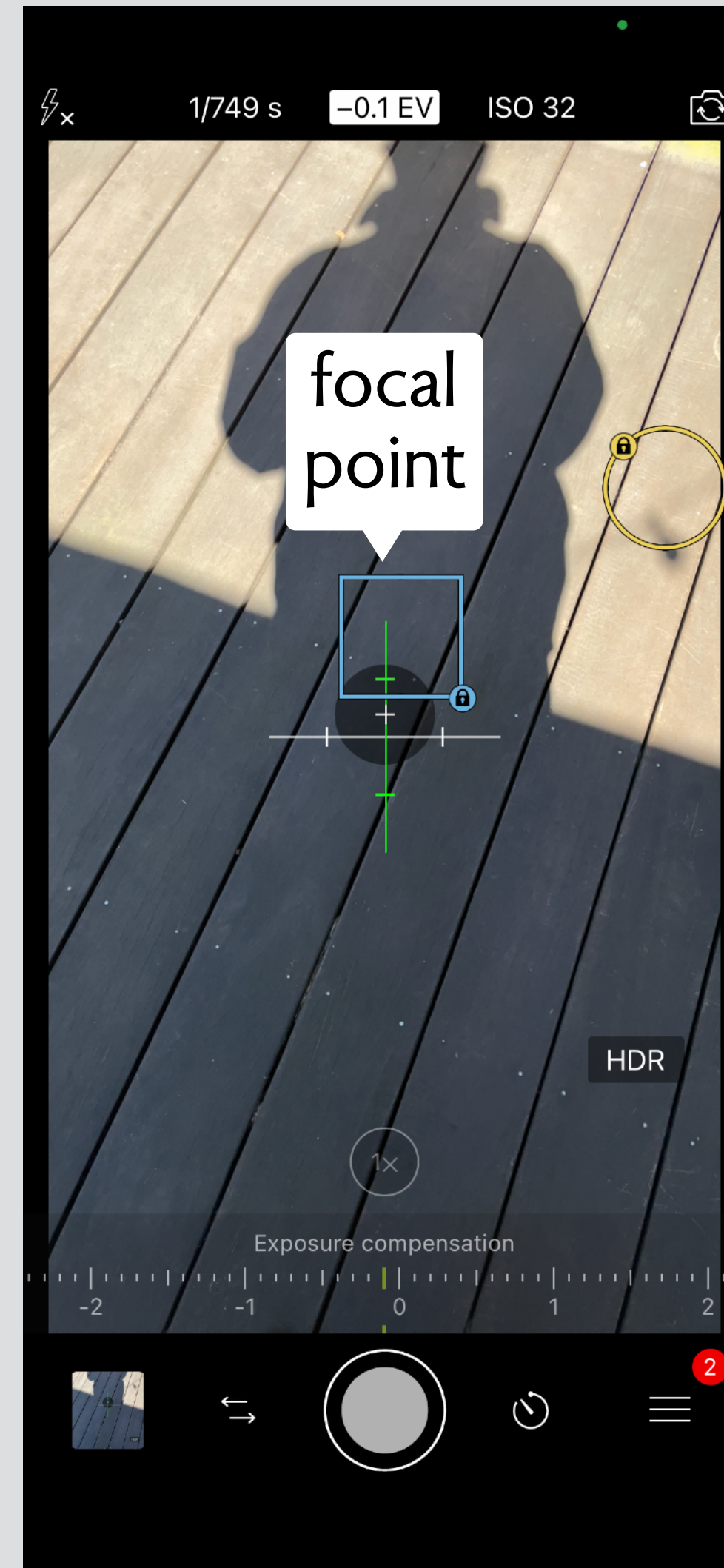


focus and exposure
points separated

loosen: expert control in ProCamera

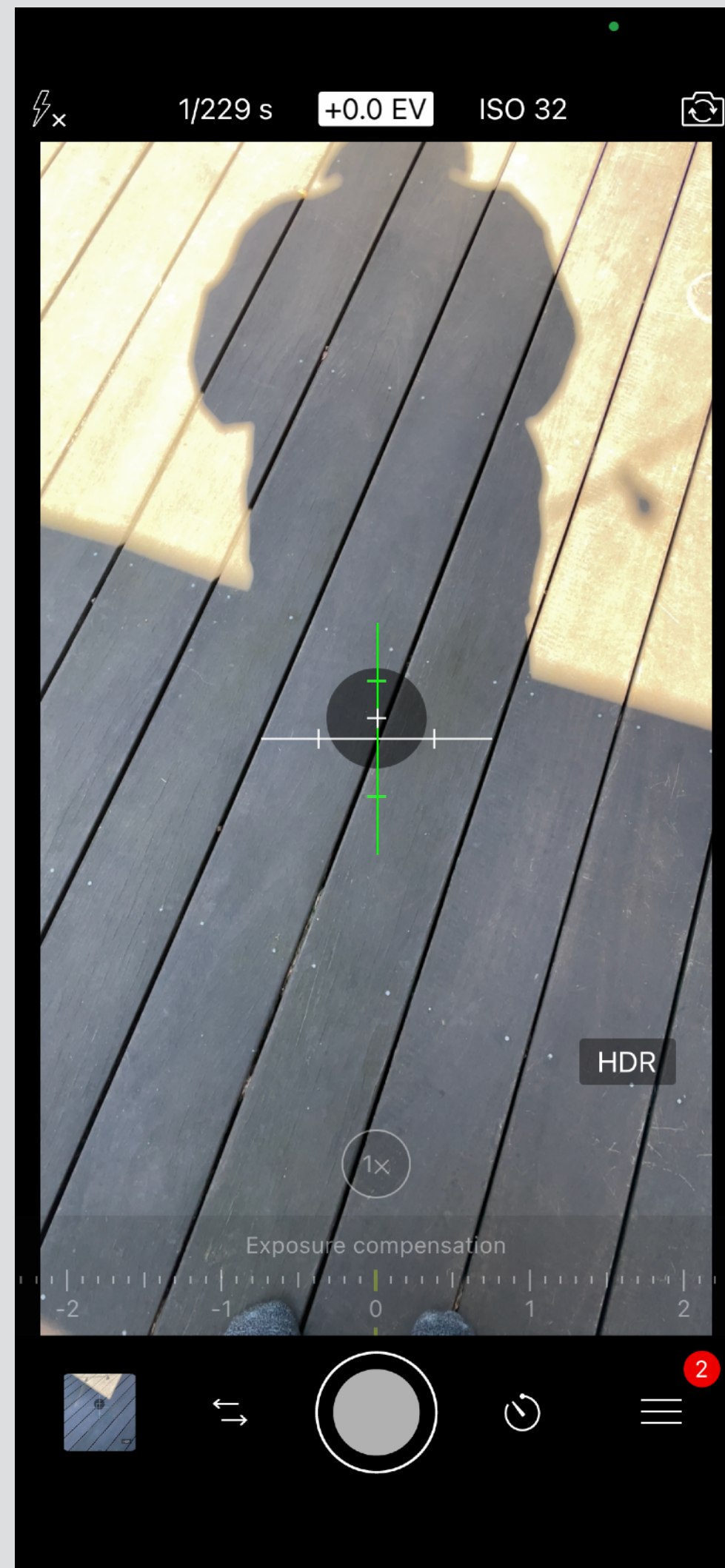


focus and exposure points aligned

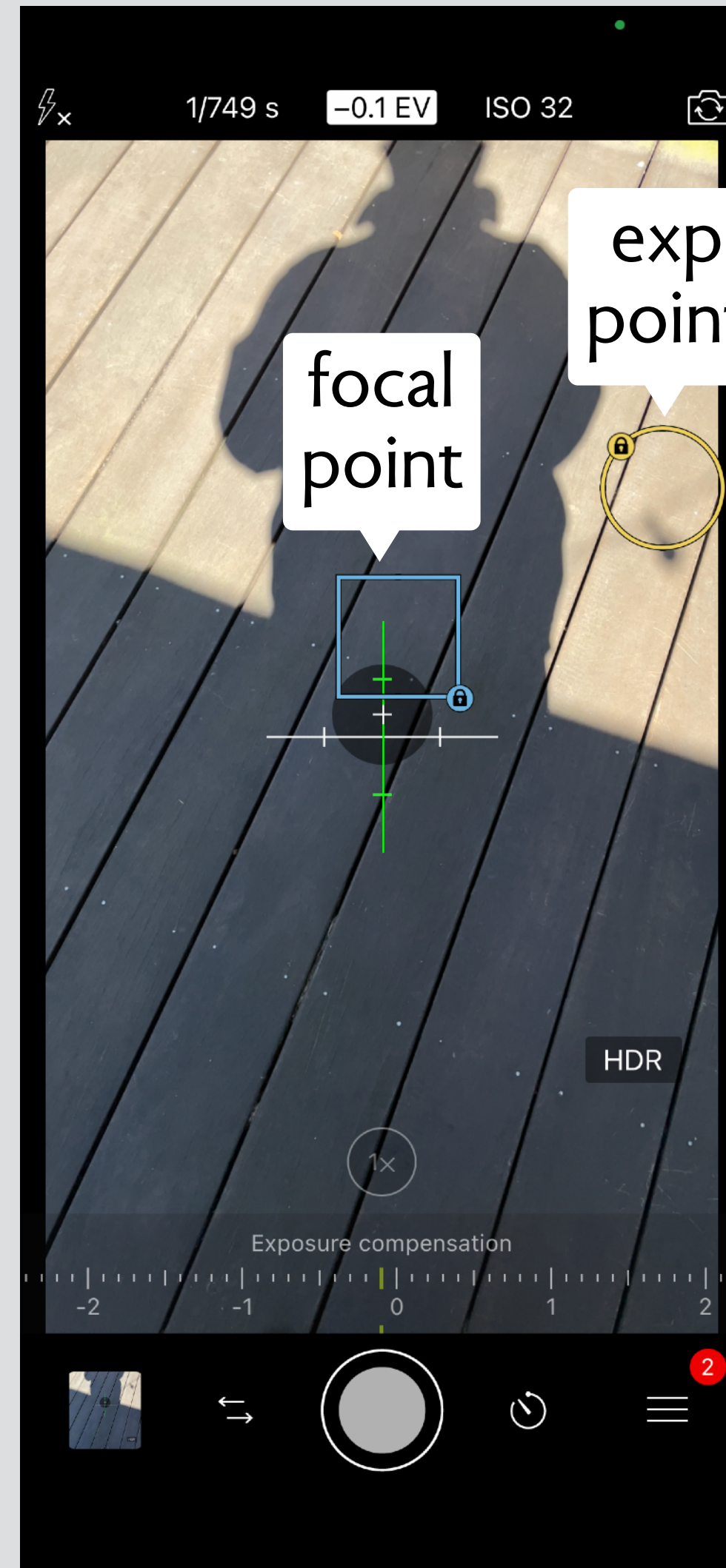


focus and exposure points separated

loosen: expert control in ProCamera

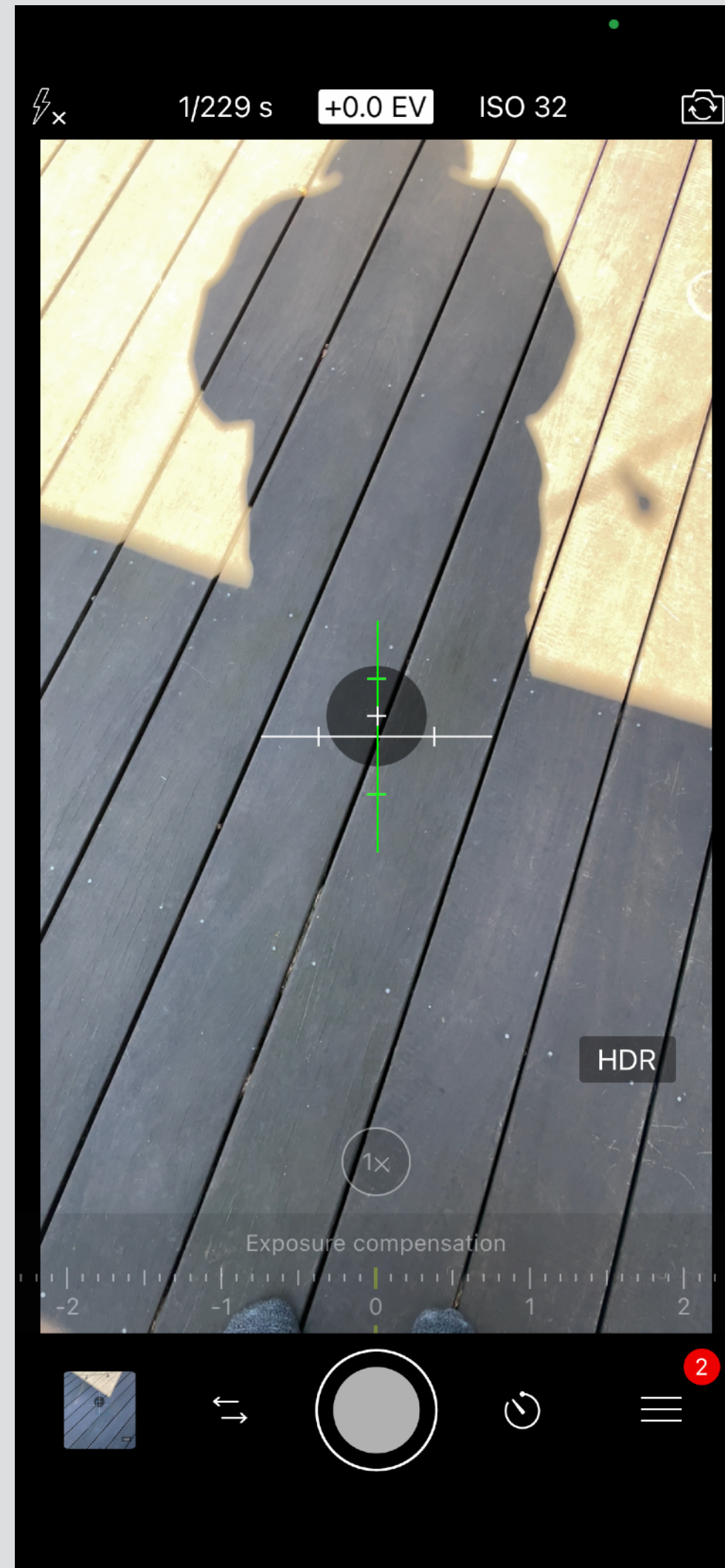


focus and exposure points aligned

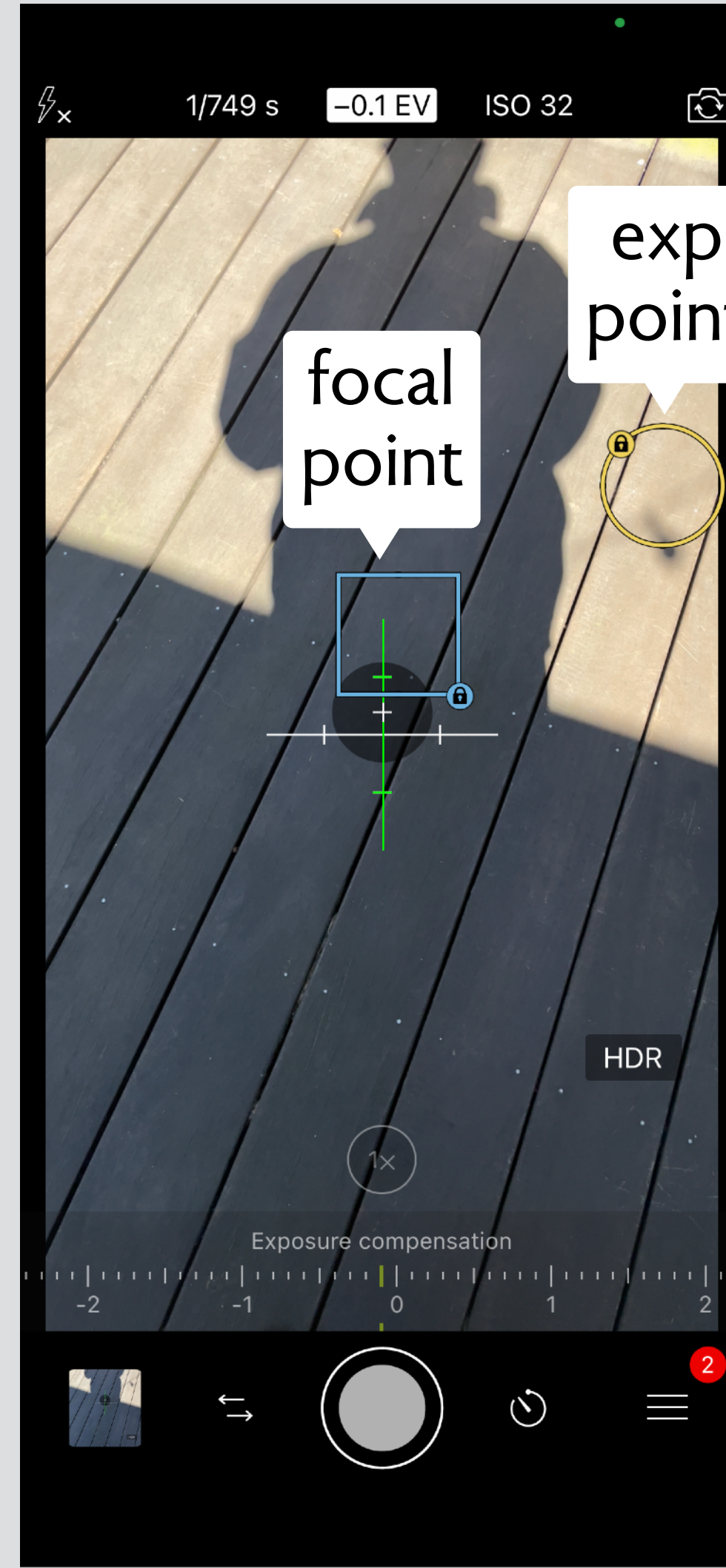


focus and exposure points separated

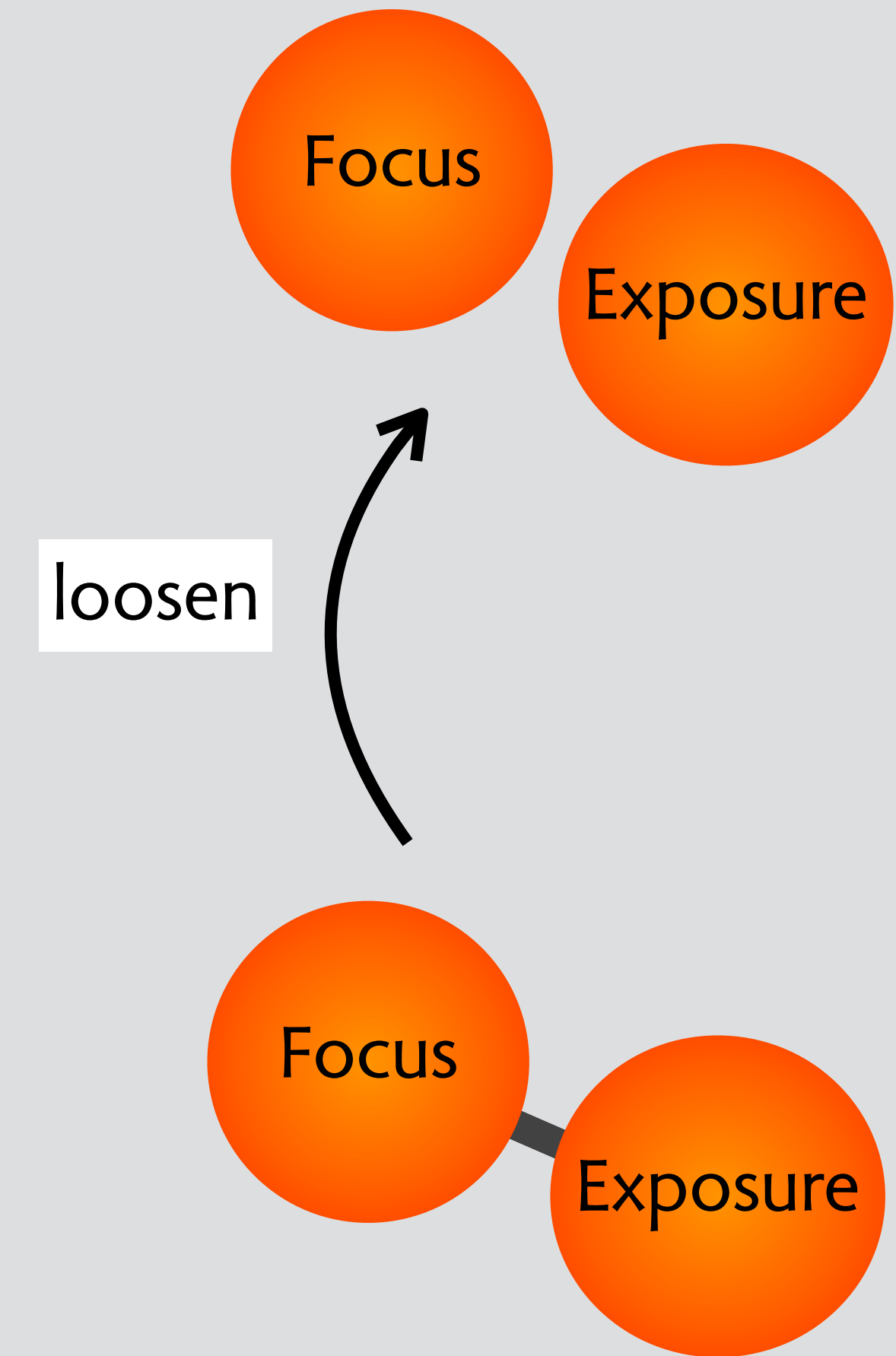
loosen: expert control in ProCamera



focus and exposure points aligned



focus and exposure points separated



design moves
in response to
problems

aspect ratio
in fujifilm cameras

a lovely camera fuji x100



complex menu system: image quality setting



complex menu system: image quality setting



complex menu system: image quality setting



aspect ratio



aspect ratio



image size setting



image size setting



image size setting

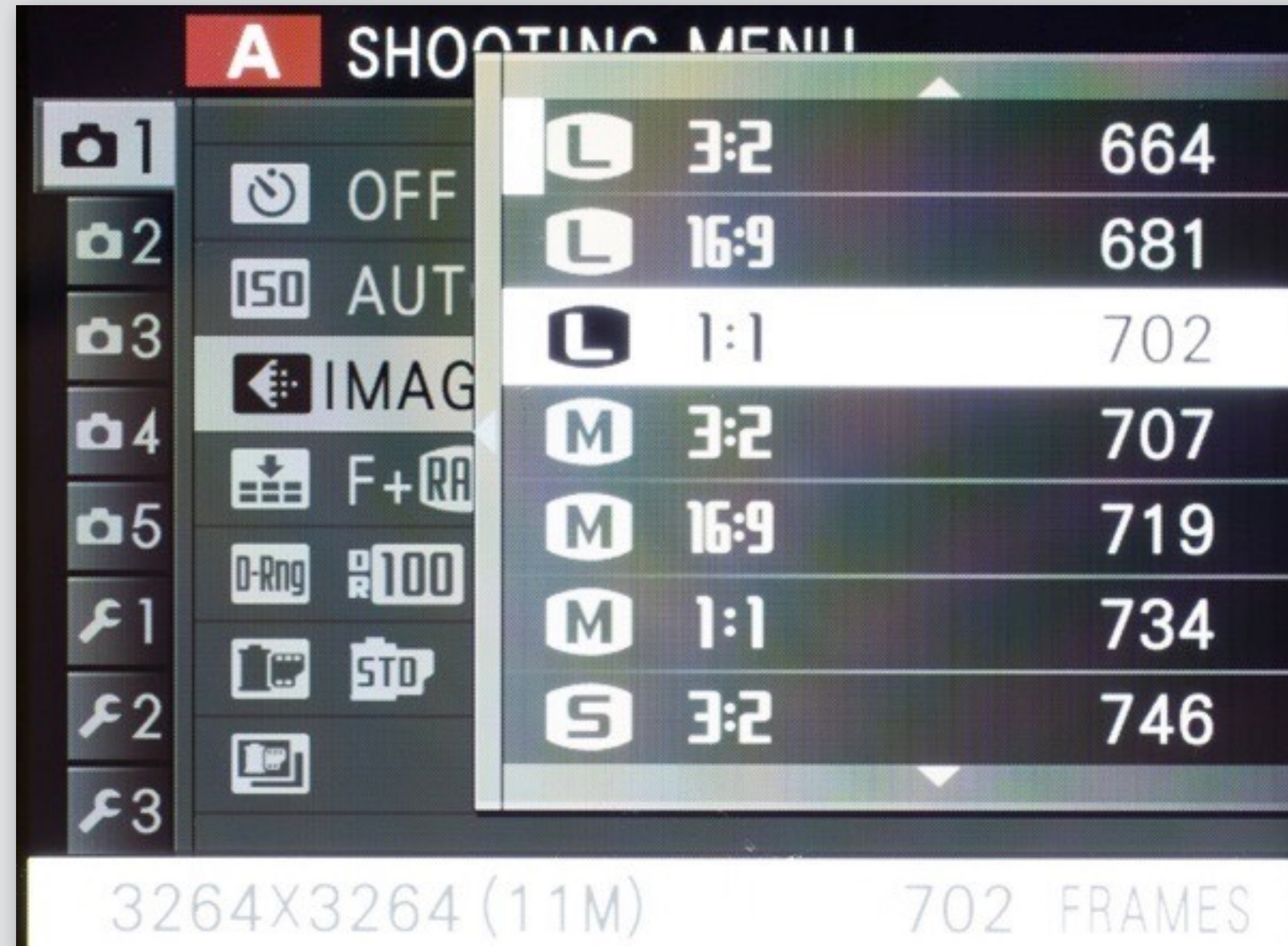


image size setting



non-standard ratio + raw?



problem #1: no non-standard ratio unless also save JPG!



raw image showing non-destructive aspect ratio crop

problem #2: very few ratio options

change.org



Petition details

Comments

Updates



Fuji, give us 4:3, 5:4, and 6:7 aspect ratios on X-series cameras

636 have signed. Let's get to 1,000!



At 1,000 signatures, this petition is more likely to be featured in recommendations!

problem #2: very few ratio options

change.org

Petition details Comments Updates



Fuji, give us 4:3, 5:4, and 6:7 aspect ratios on X-series cameras

636 have signed. Let's get to 1,000!



At 1,000 signatures, this petition is more likely to be featured in recommendations!

L	3:2	664
L	16:9	681
L	1:1	702
M	3:2	707
M	16:9	719
M	1:1	734
S	3:2	746

diagnosis?

aspect ratio is not a concept

merged into JPEG image size concept
so cannot be controlled independently
I call this "overloading by piggybacking"

solution: split concepts

would allow ratio change to raws without JPEGs
would avoid combinatoric explosion of options

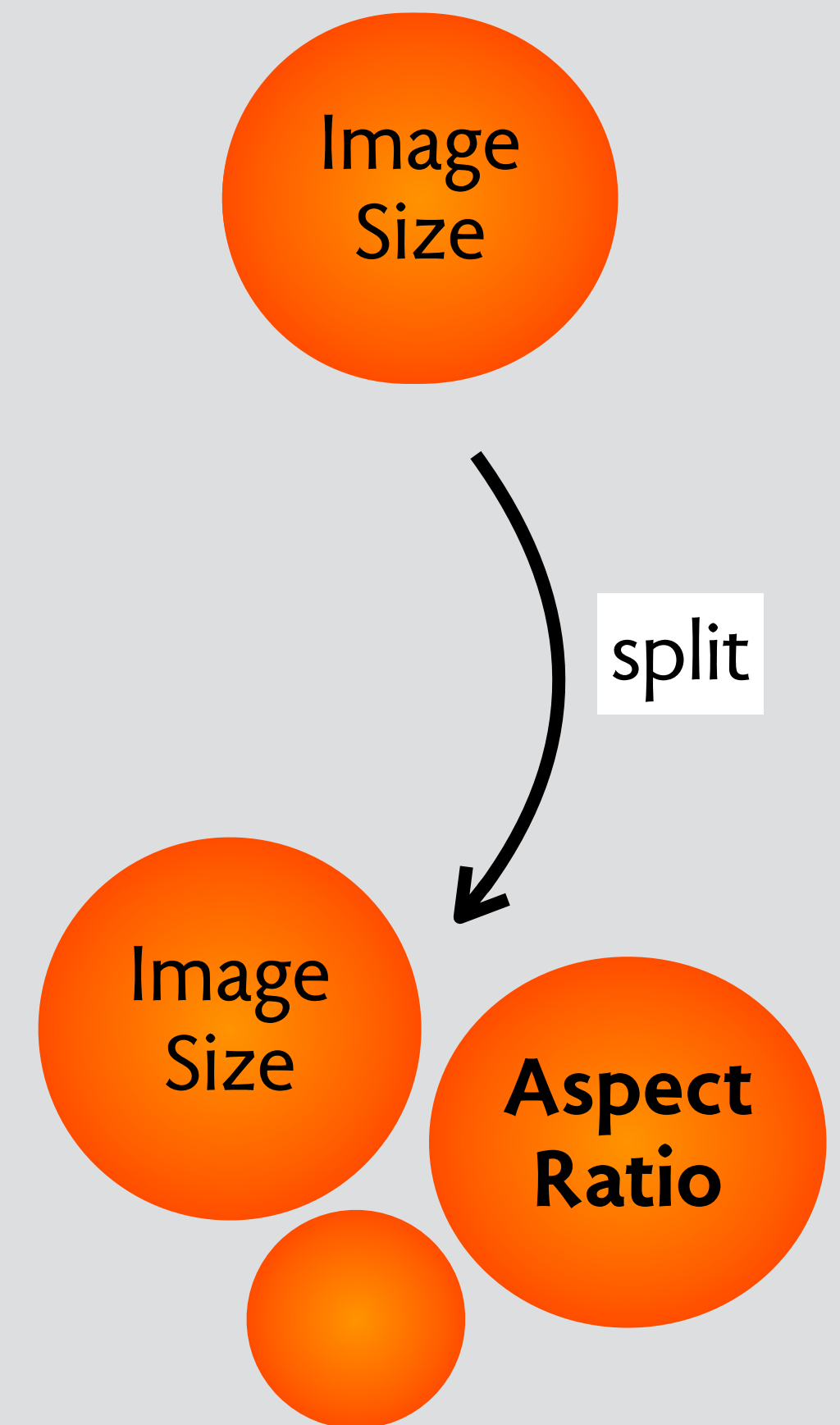
diagnosis?

aspect ratio is not a concept

merged into JPEG image size concept
so cannot be controlled independently
I call this "overloading by piggybacking"

solution: split concepts

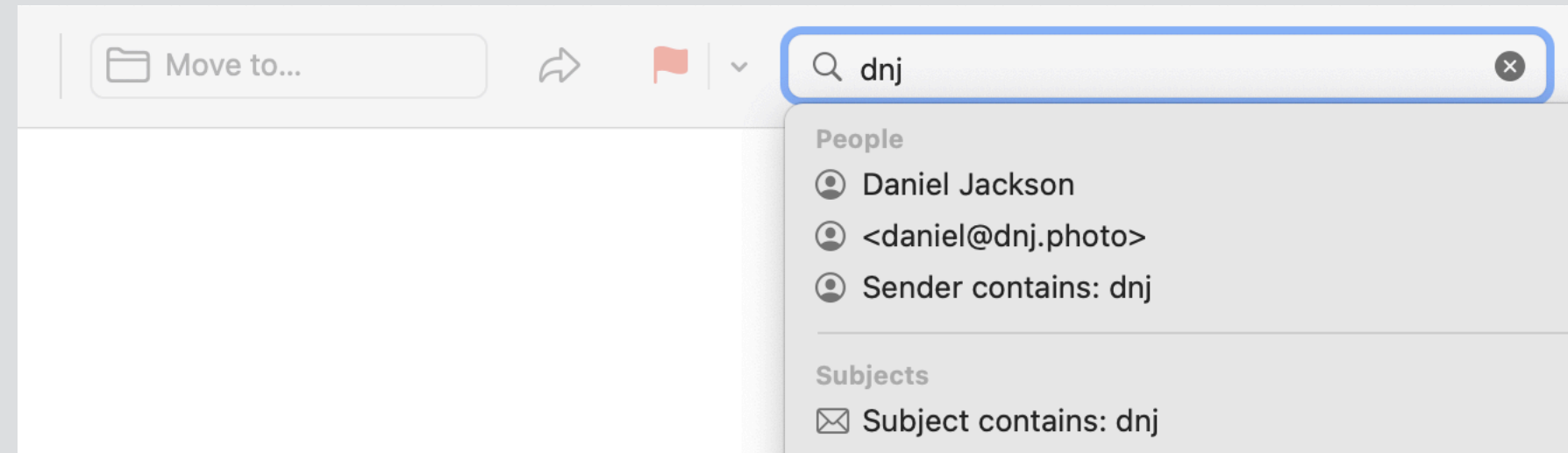
would allow ratio change to raws without JPEGs
would avoid combinatoric explosion of options



message filters
in apple mail

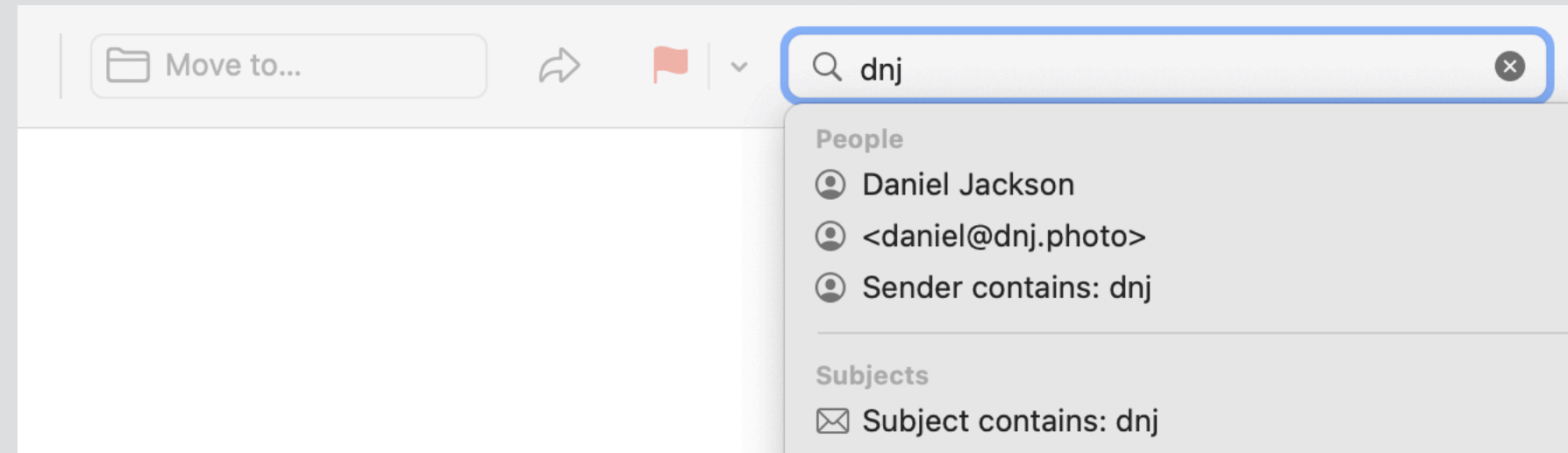
how many ways to filter messages?

how many ways to filter messages?

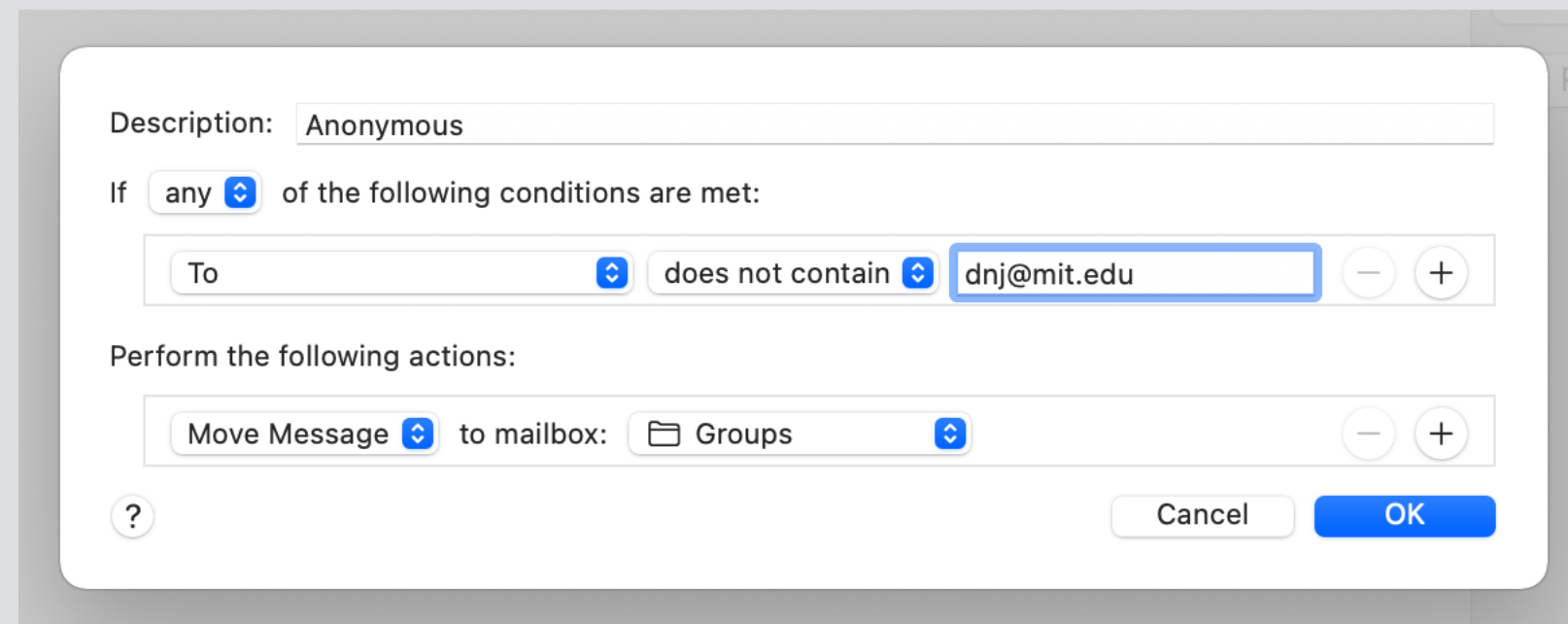


search for a message

how many ways to filter messages?

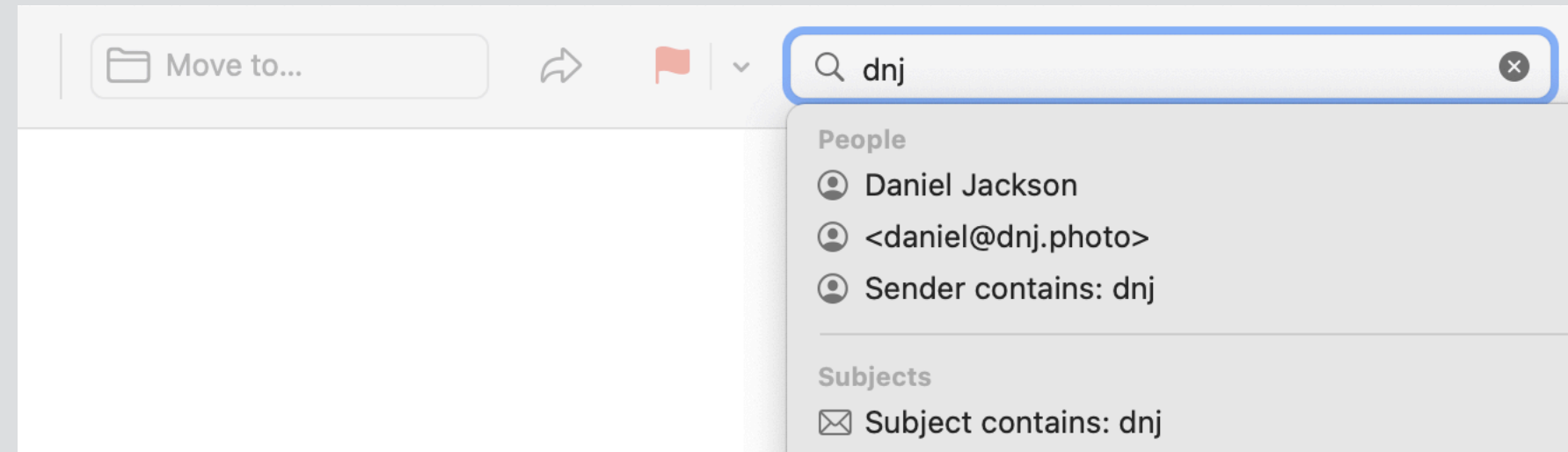


search for a message

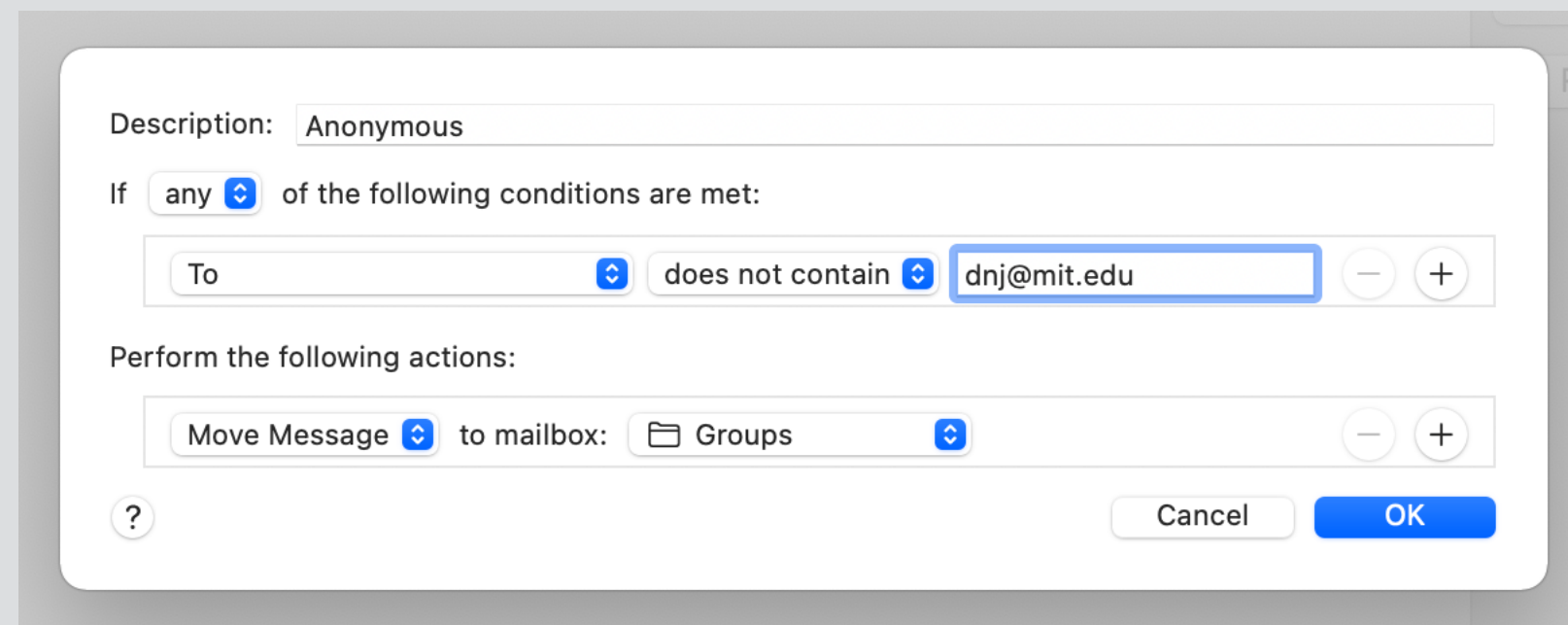


create a rule

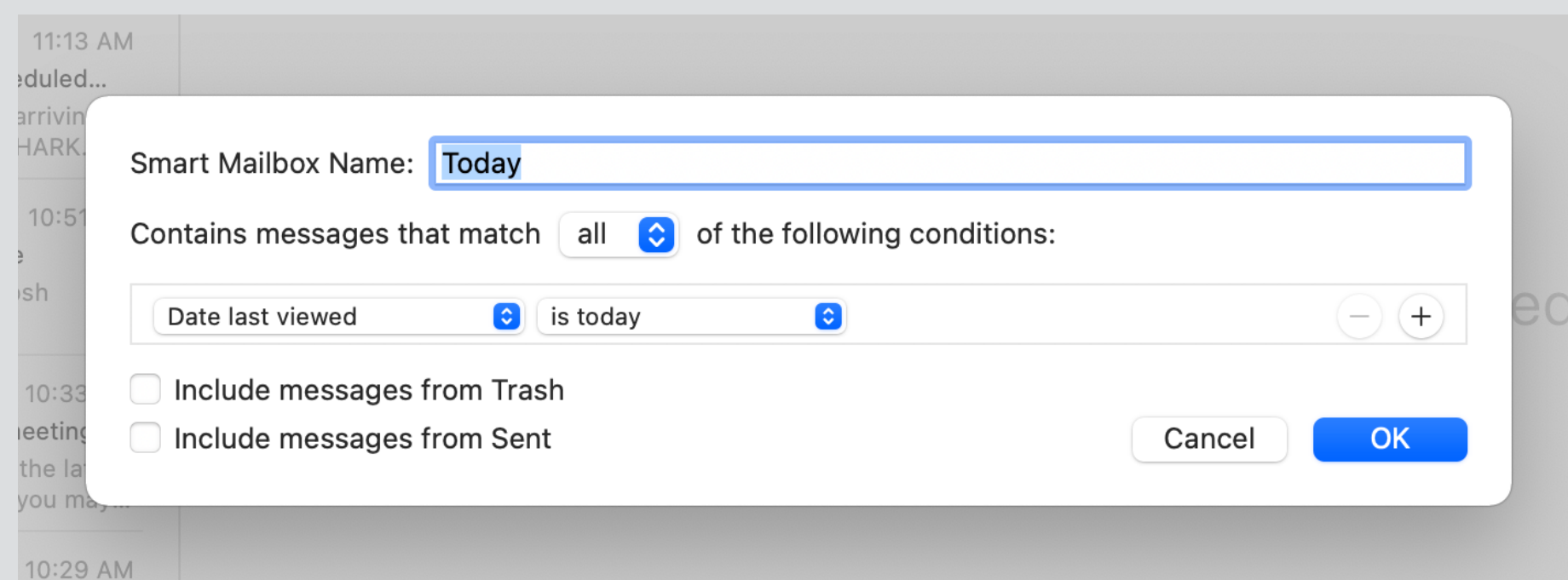
how many ways to filter messages?



search for a message

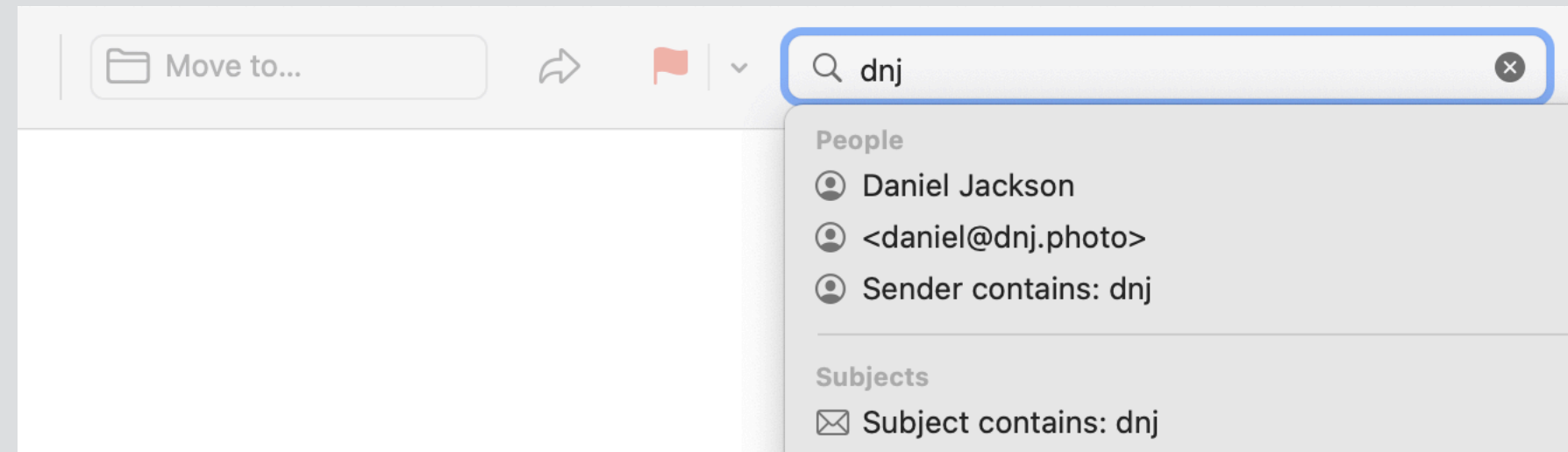


create a rule

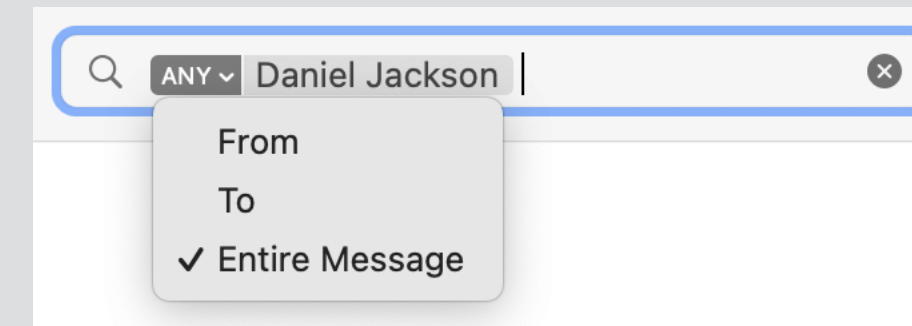


define a smart folder

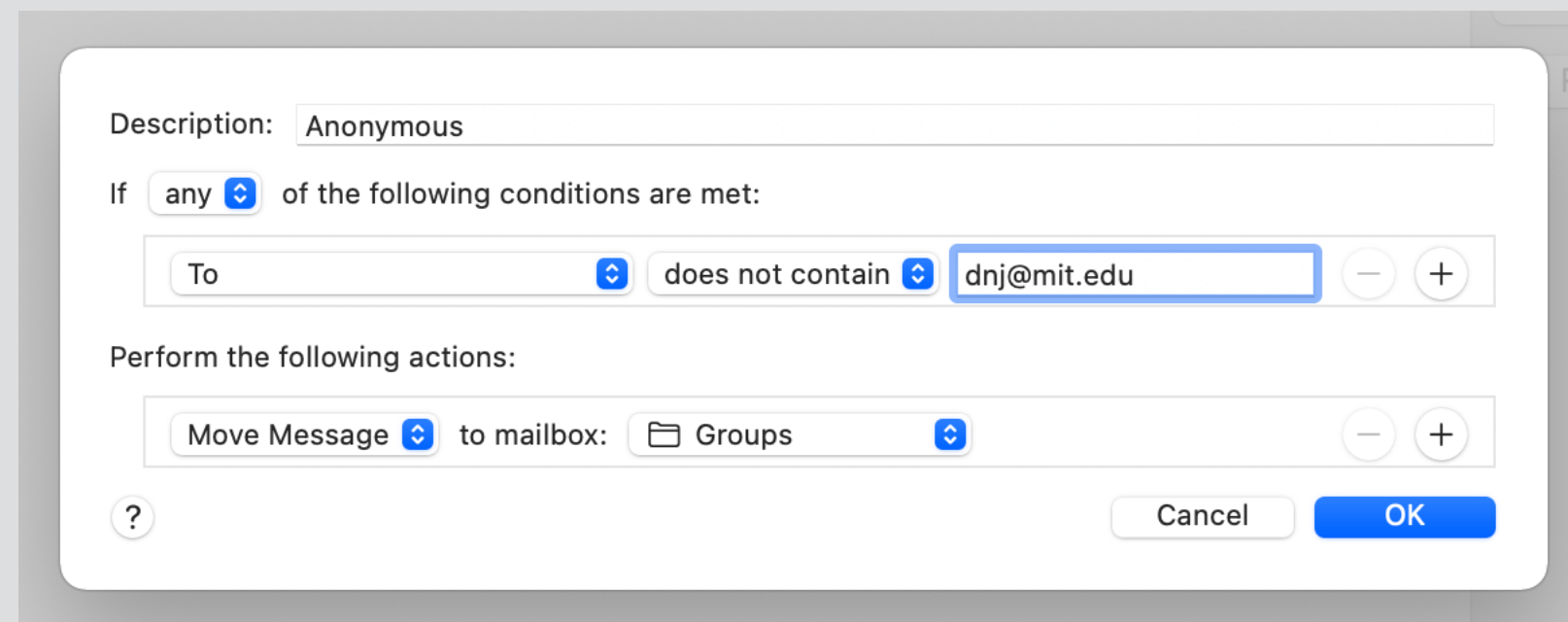
how many ways to filter messages?



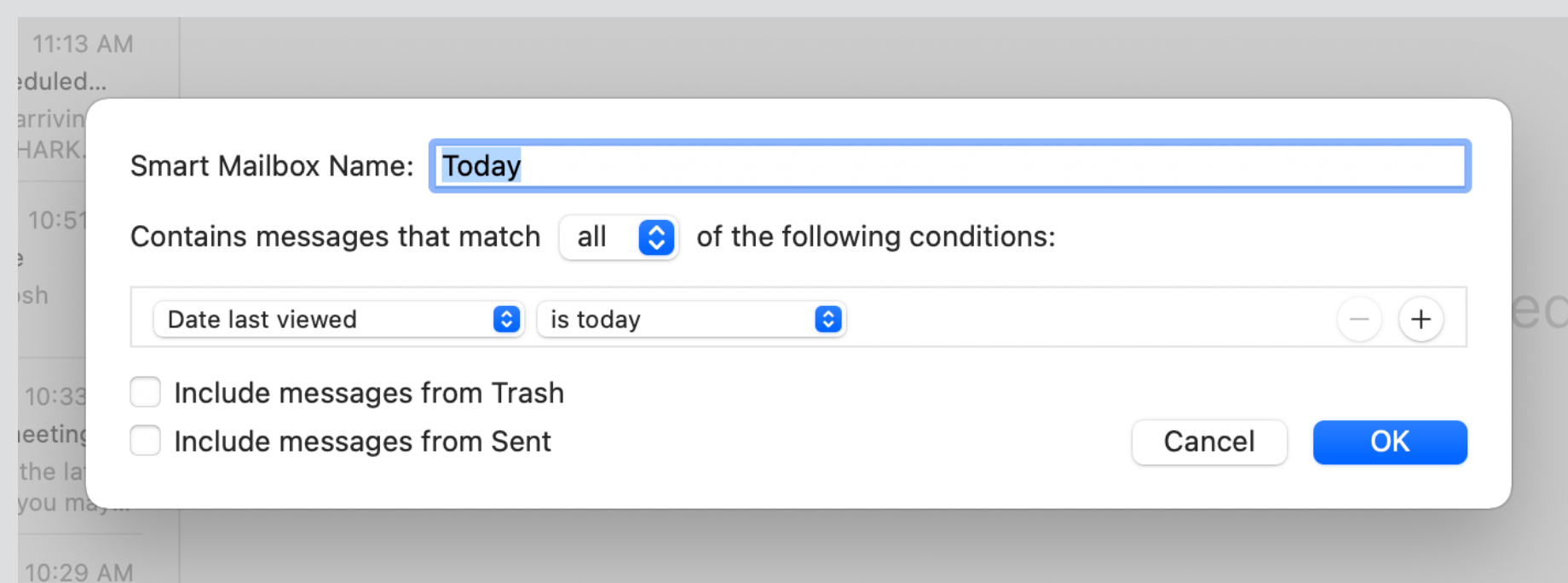
search for a message



search options

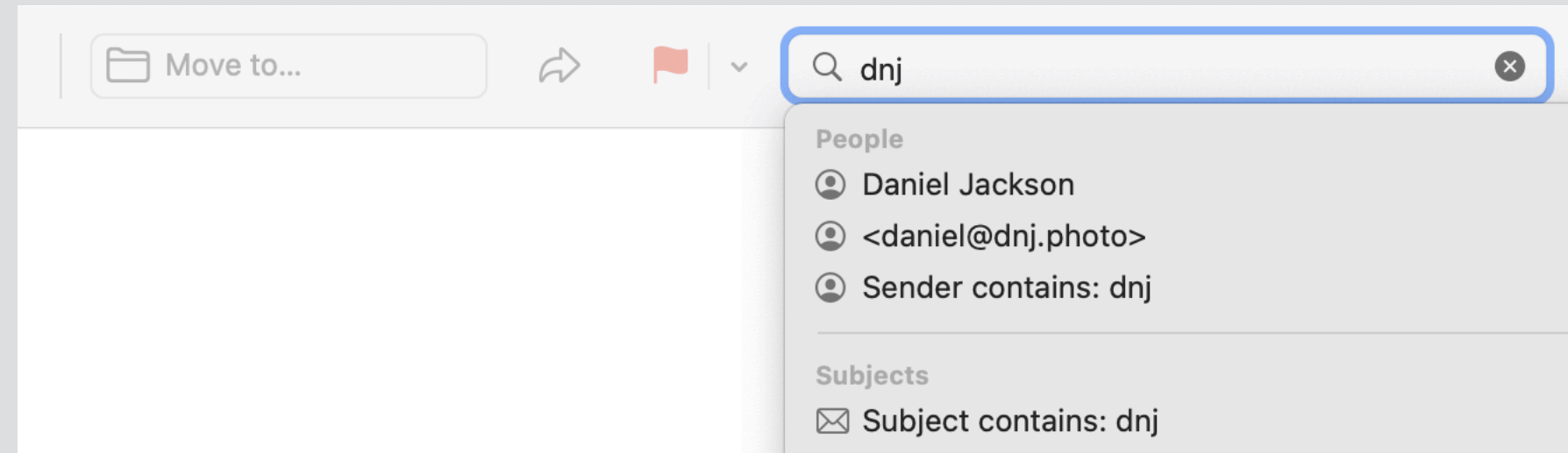


create a rule

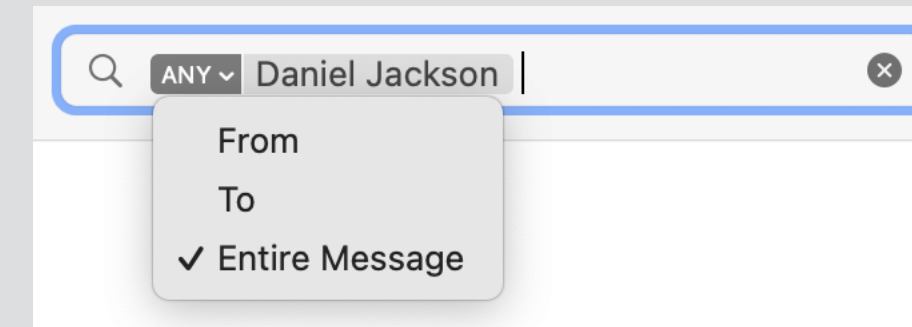


define a smart folder

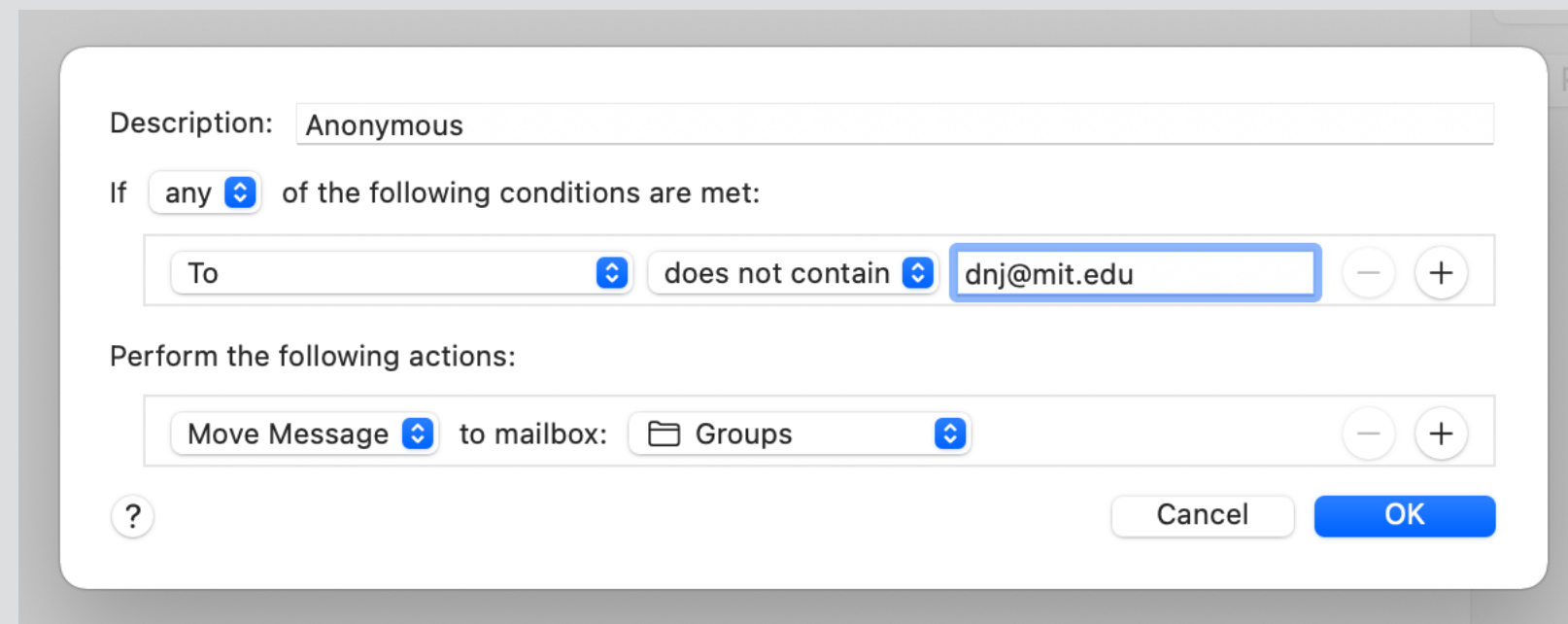
how many ways to filter messages?



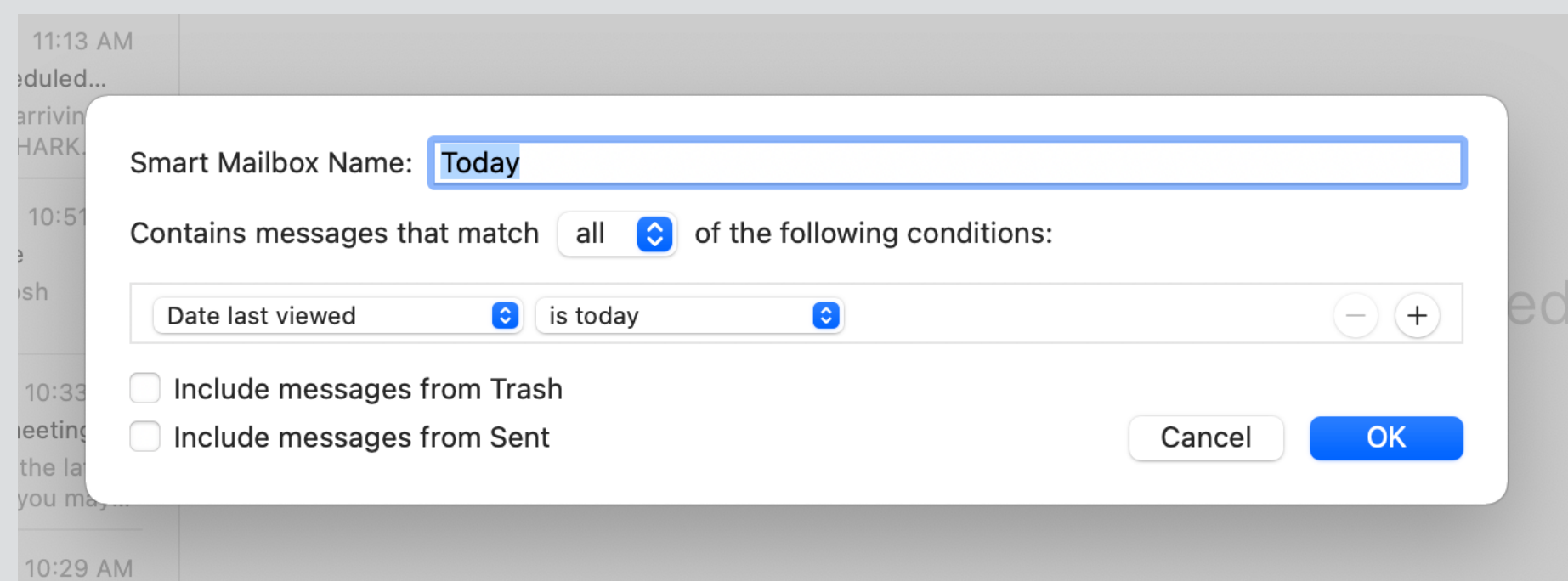
search for a message



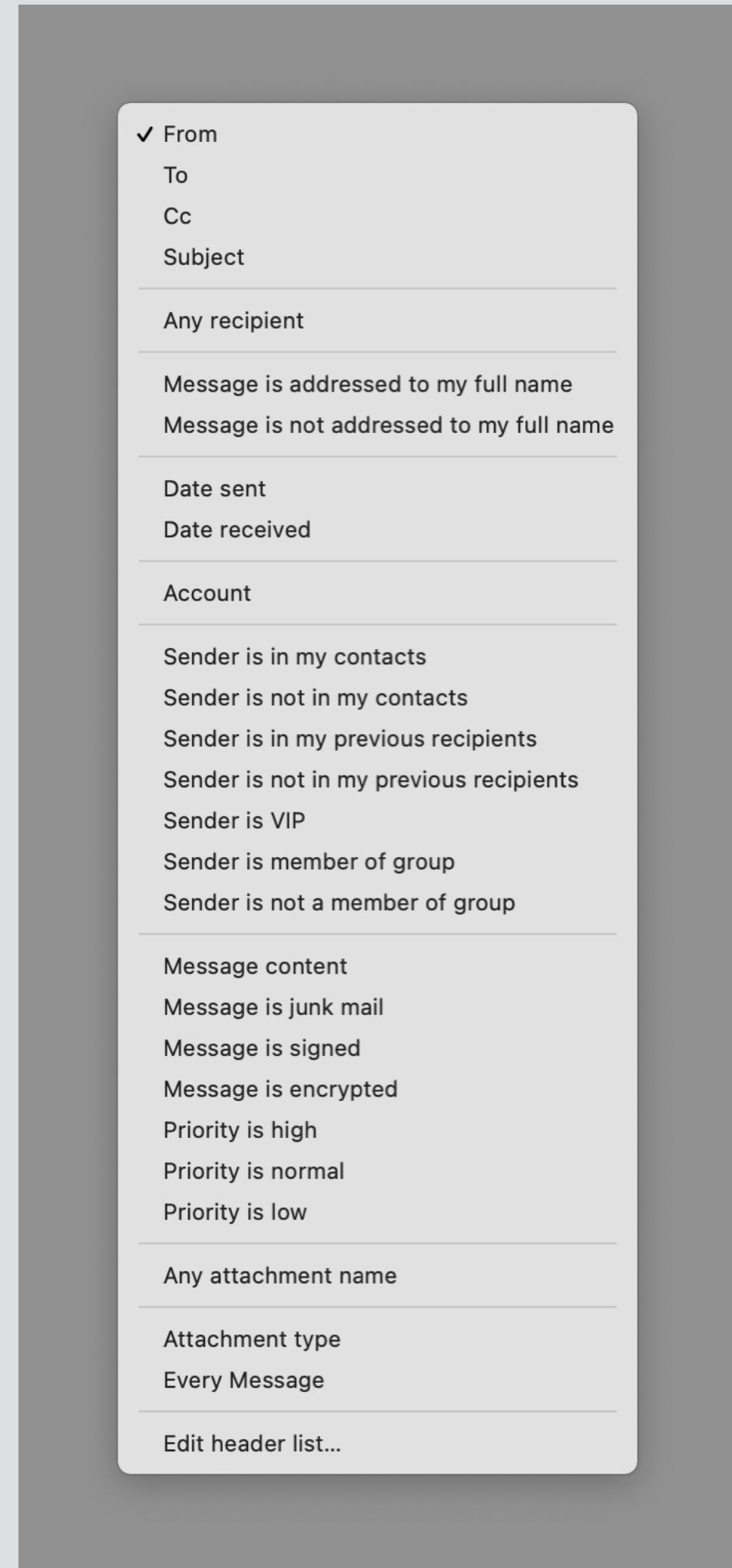
search options



create a rule

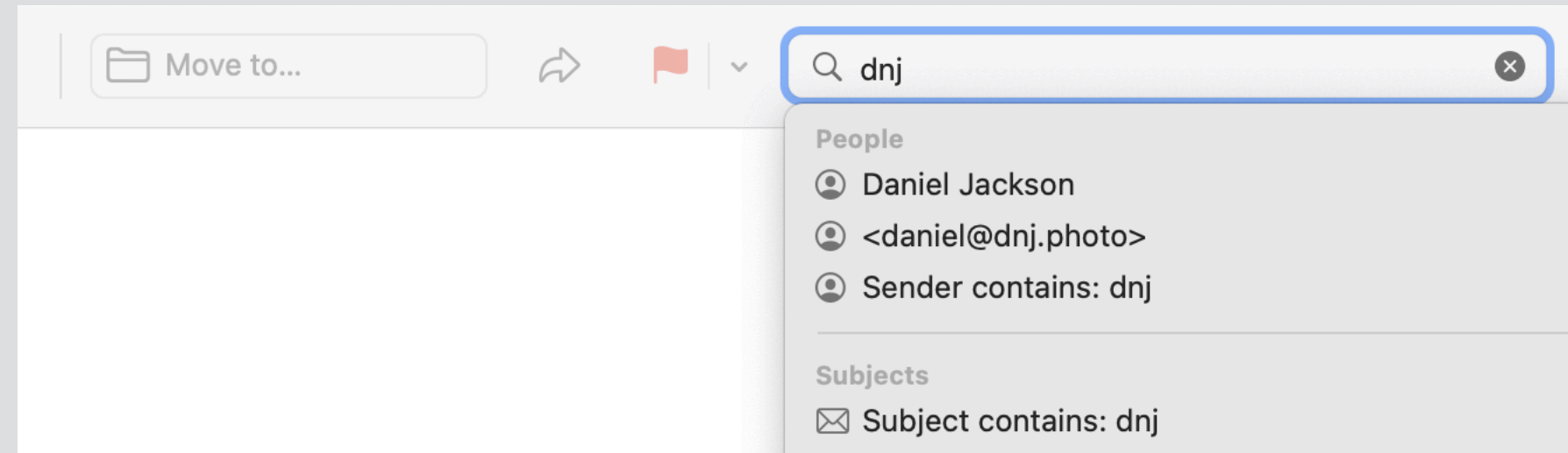


define a smart folder

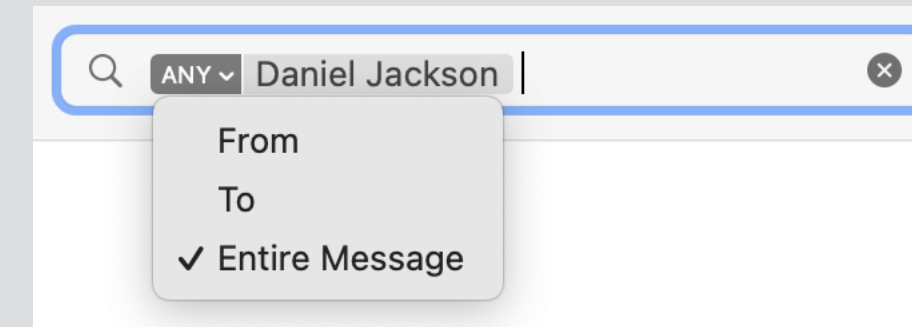


rule options

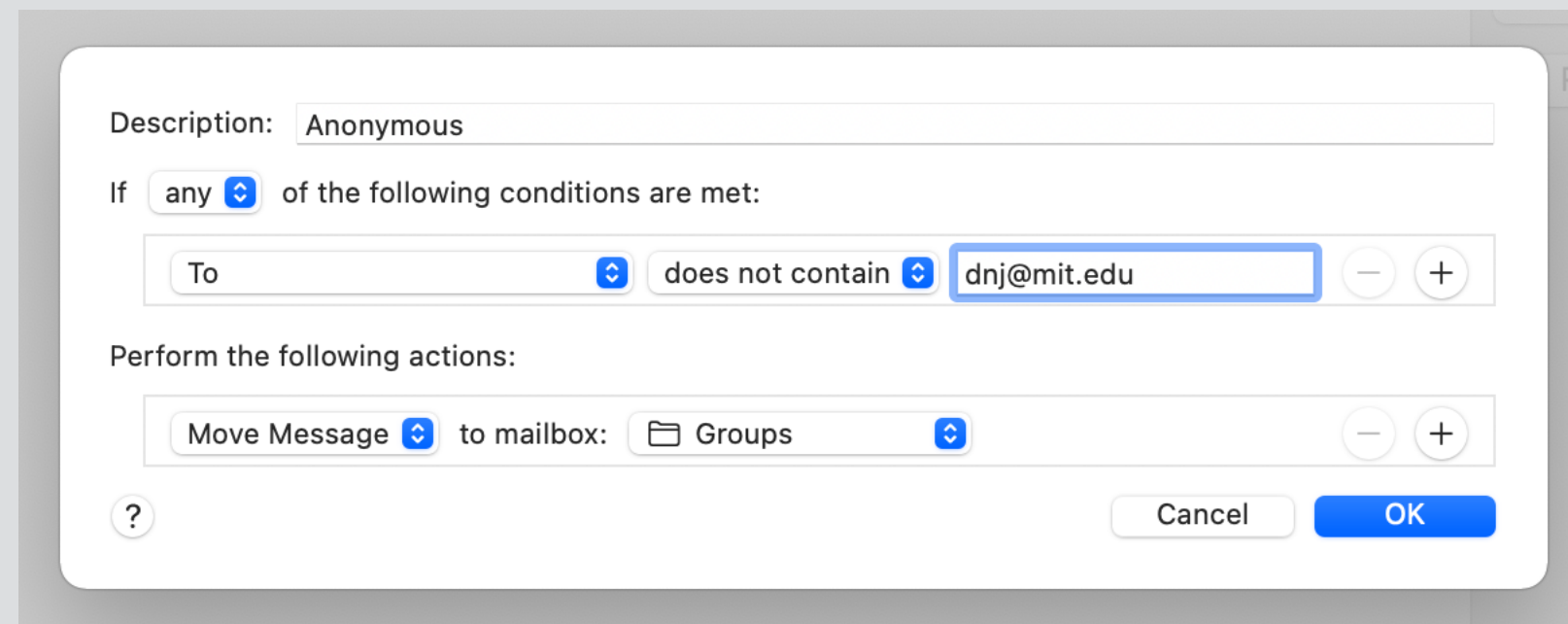
how many ways to filter messages?



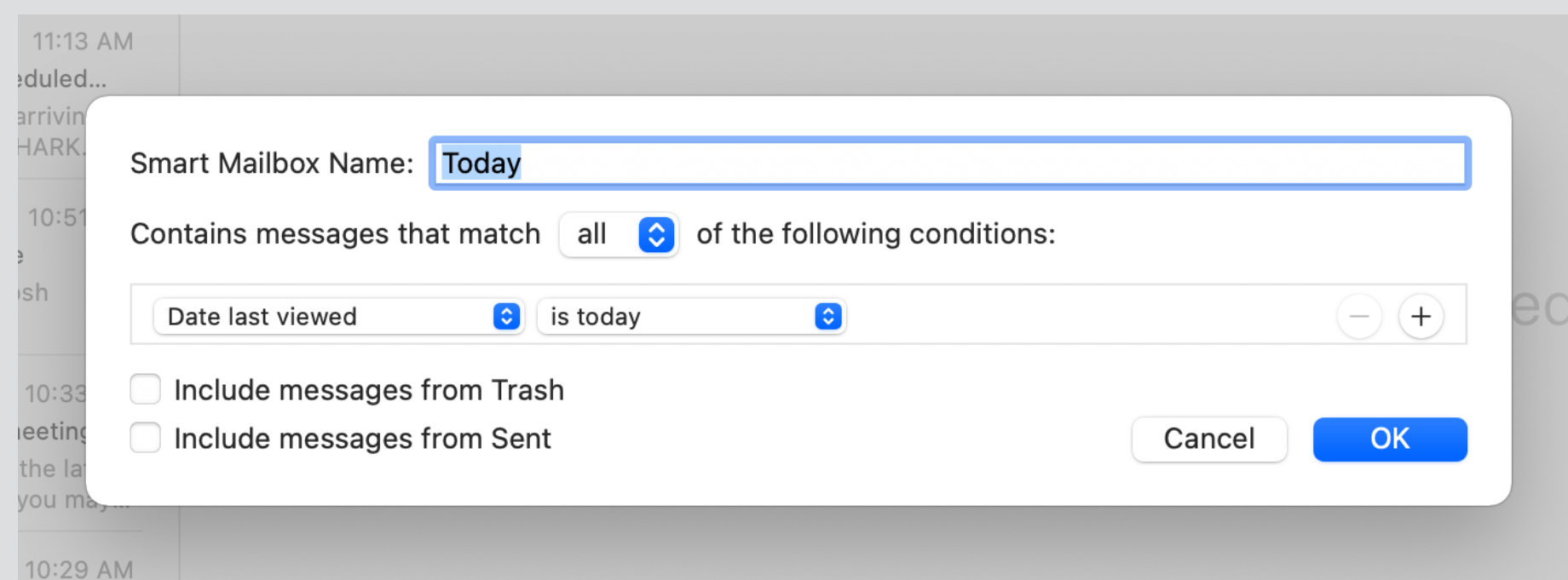
search for a message



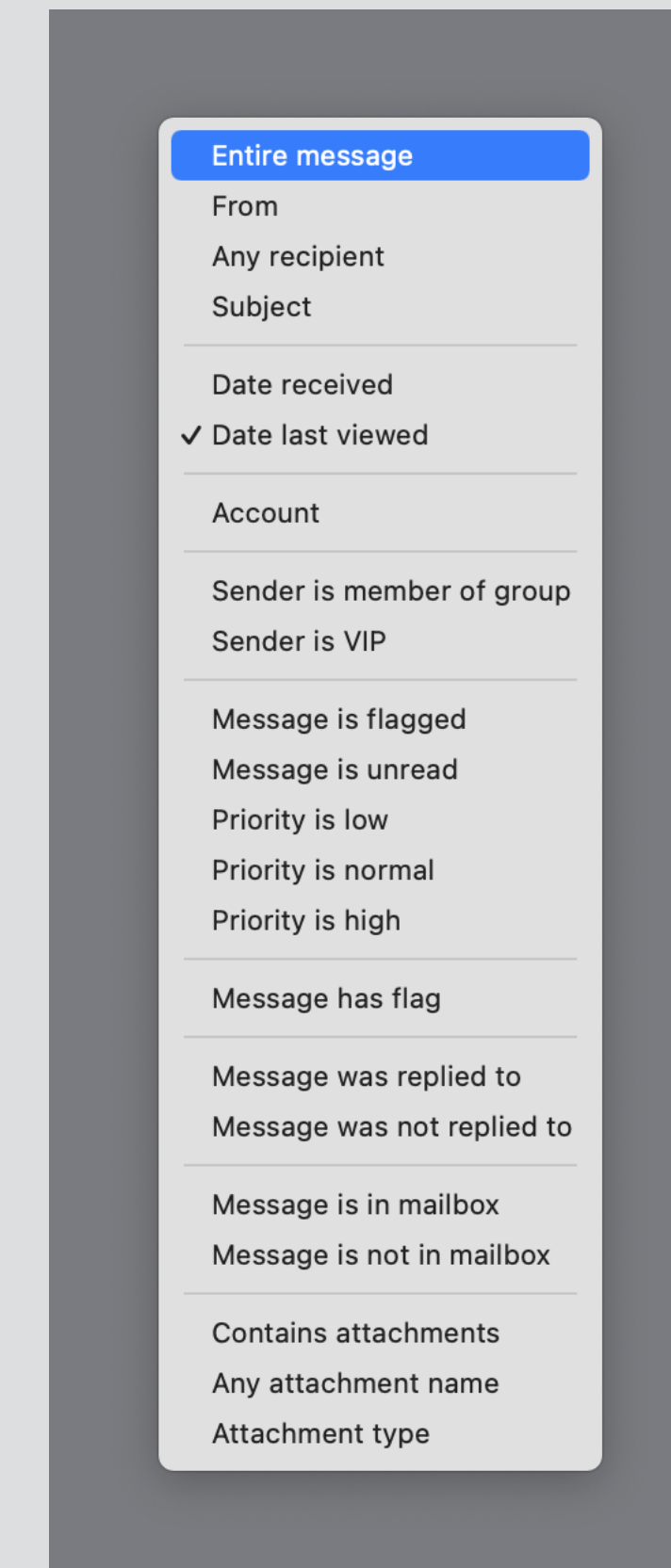
search options



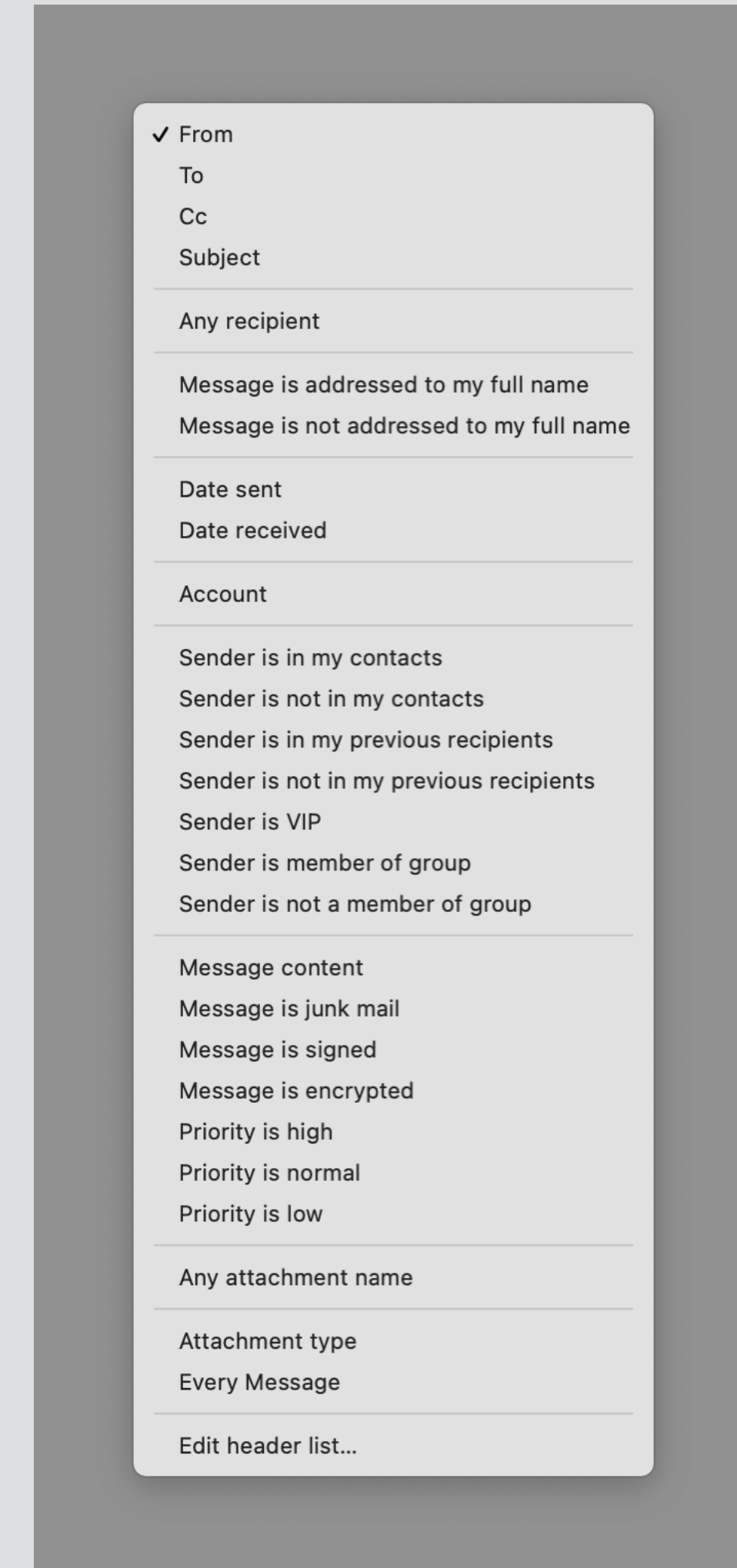
create a rule



define a smart folder



smart folder options



rule options

diagnosis?

search, rule and smart folder

all include their own specialized concepts
incomparable features, different UIs

unify in a single message filter concept

include "create folder from search", eg

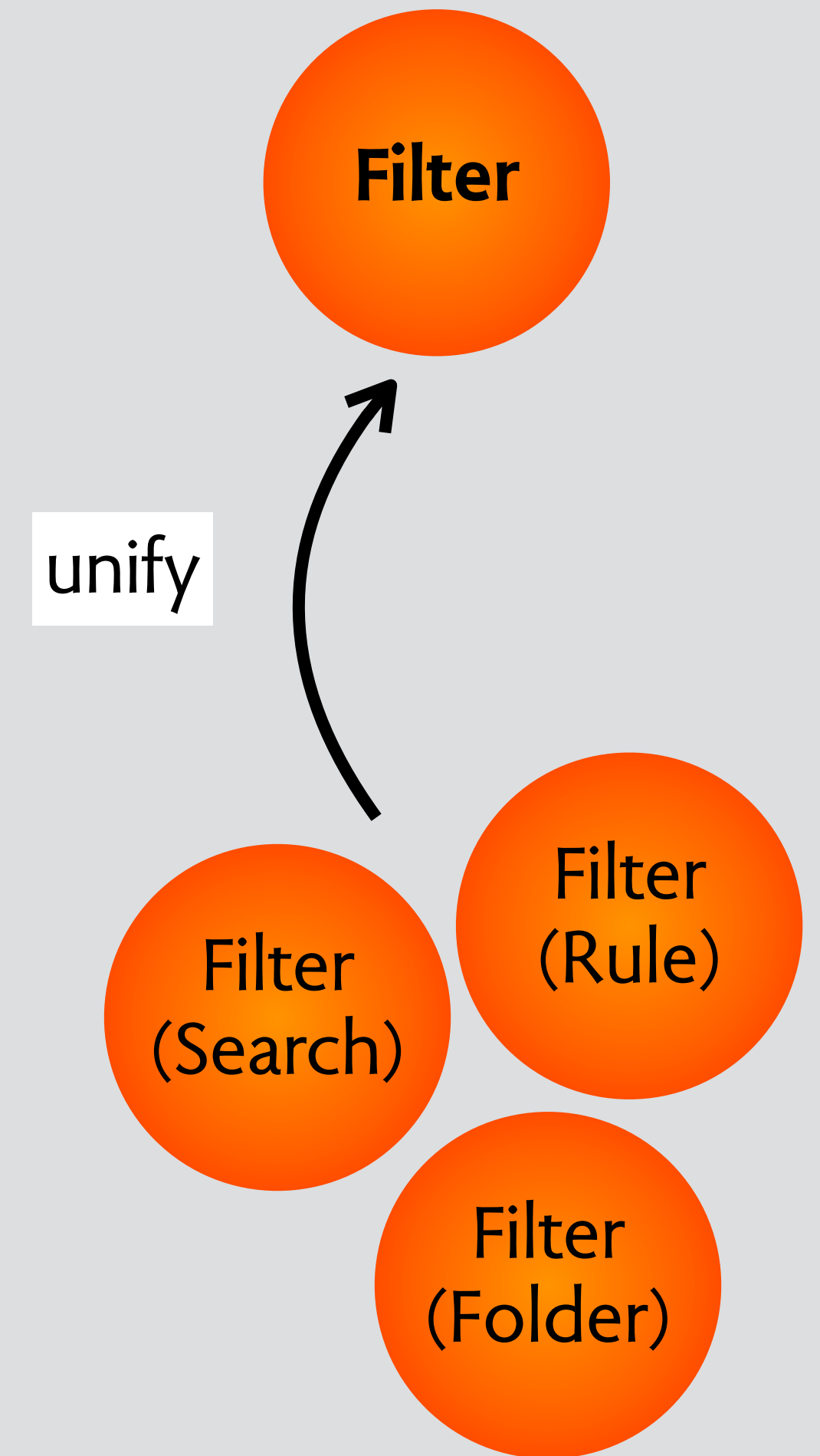
diagnosis?

search, rule and smart folder

all include their own specialized concepts
incomparable features, different UIs

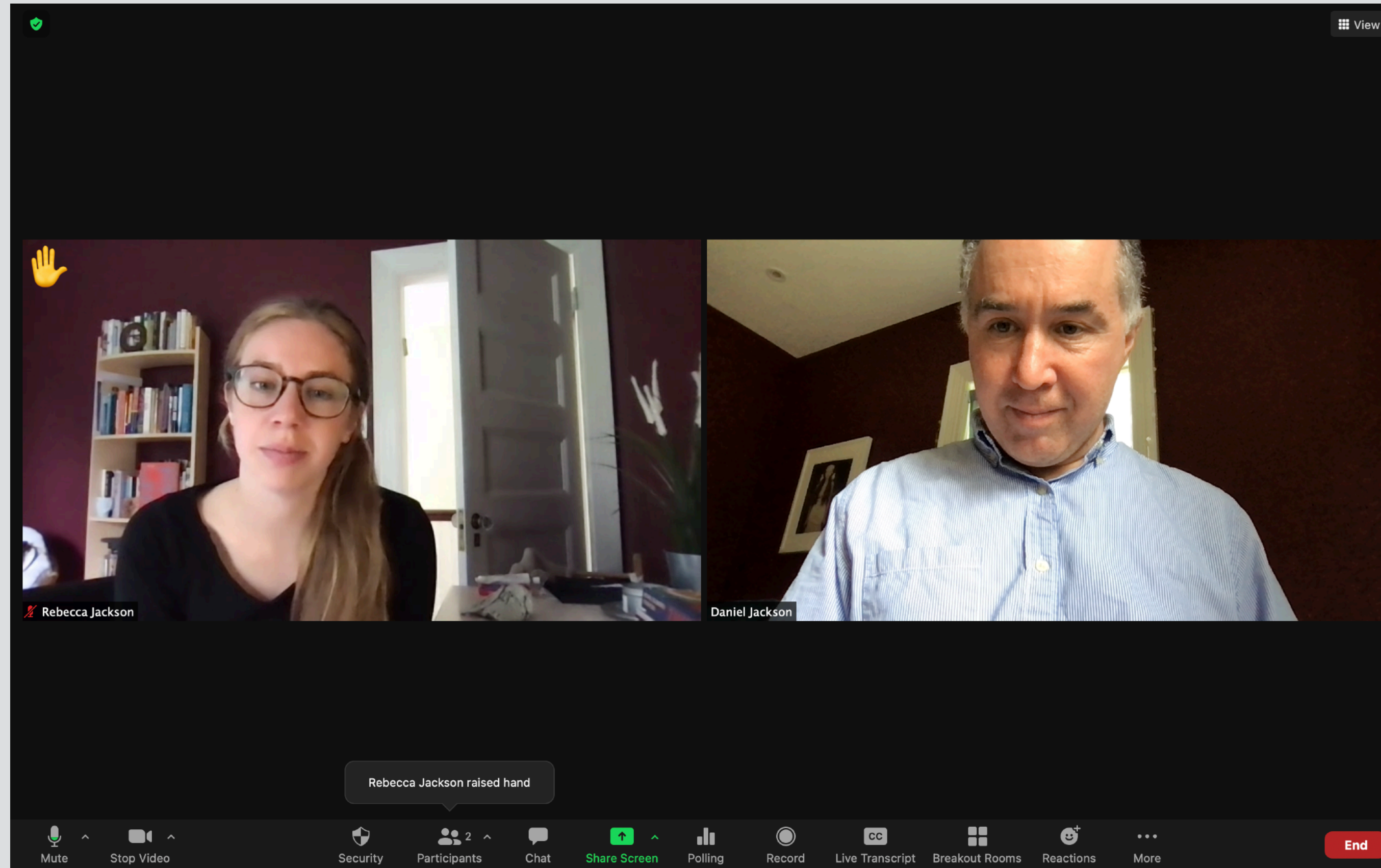
unify in a single message filter concept

include "create folder from search", eg

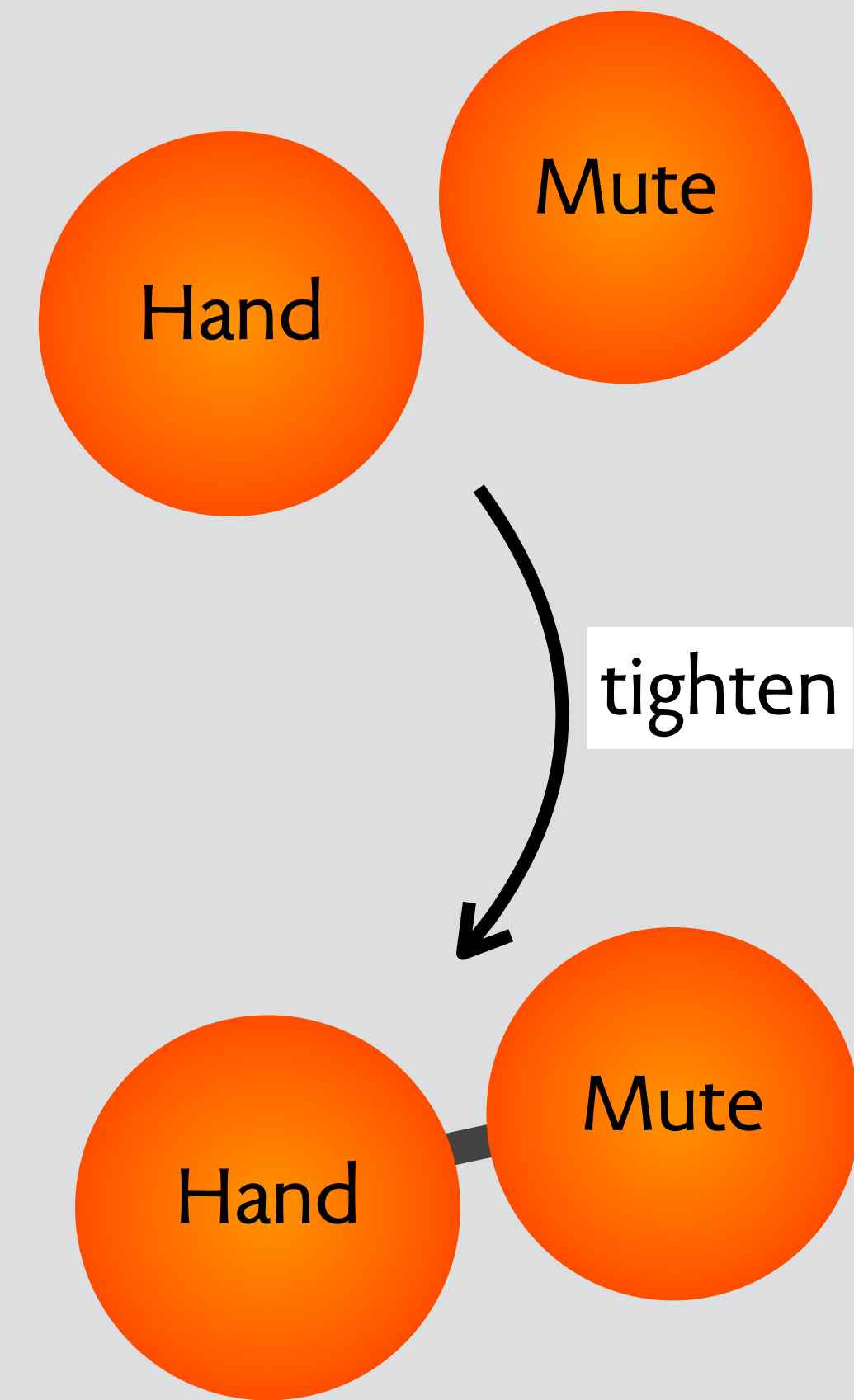
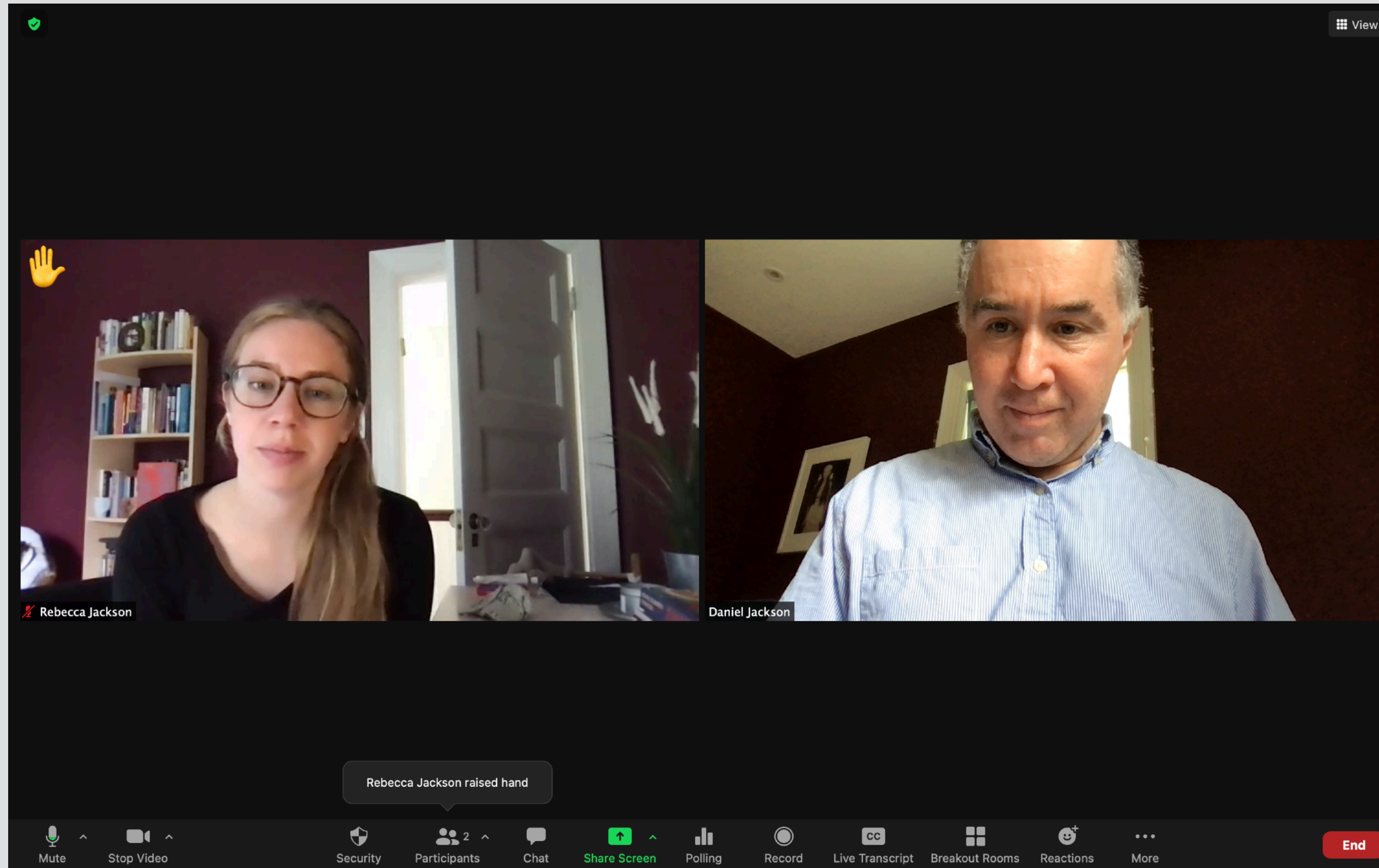


**sticky hands
in zoom**

zoom

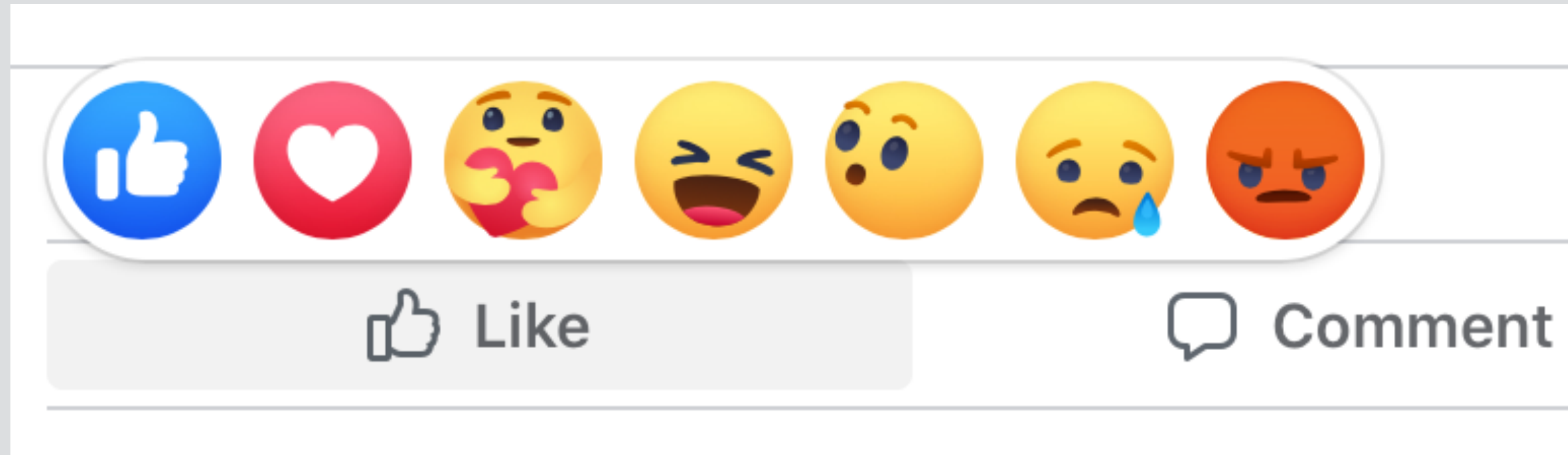


zoom

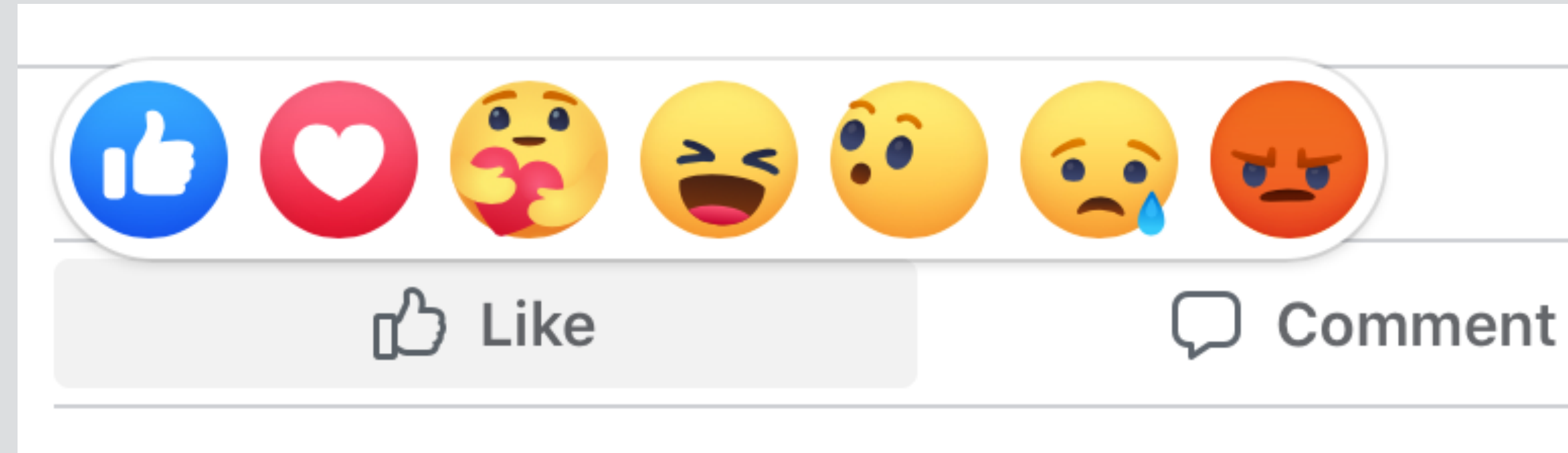


likes
in facebook

are angry reactions upvotes?



are angry reactions upvotes?



concept Upvote

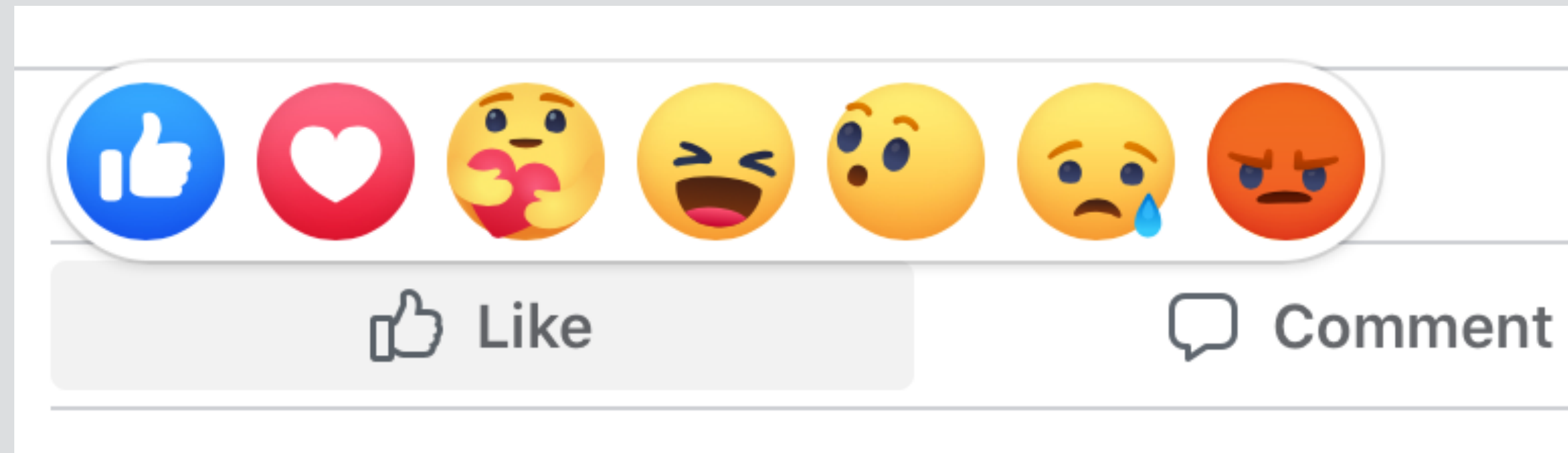
purpose rank items by popularity

actions

upvote (u: User, i: Item)

...

are angry reactions upvotes?



concept Upvote

purpose rank items by popularity

actions

upvote (u: User, i: Item)

...



concept Reaction

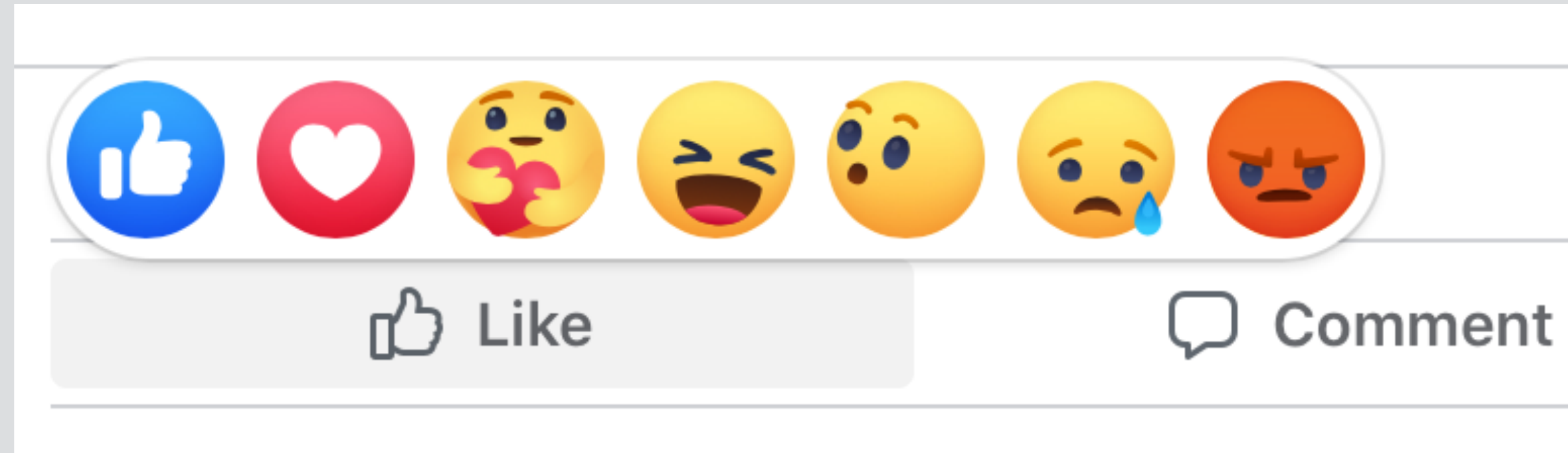
purpose convey emotion to author

actions

reactAngry (u: User, i: Item)

...

are angry reactions upvotes?



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?

a facebook split



Like

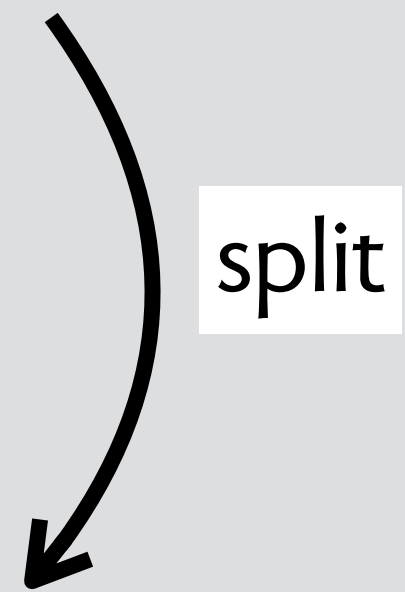


Comment

a facebook split

 Like

 Comment

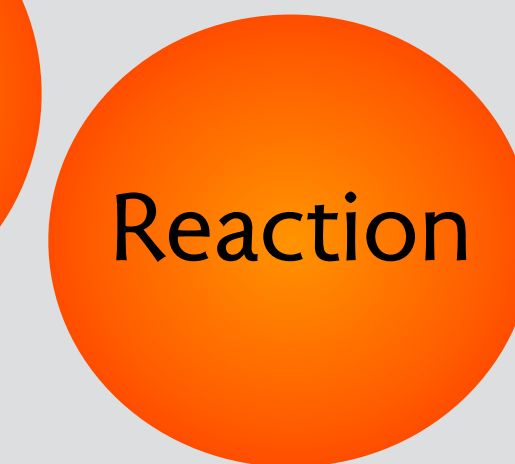
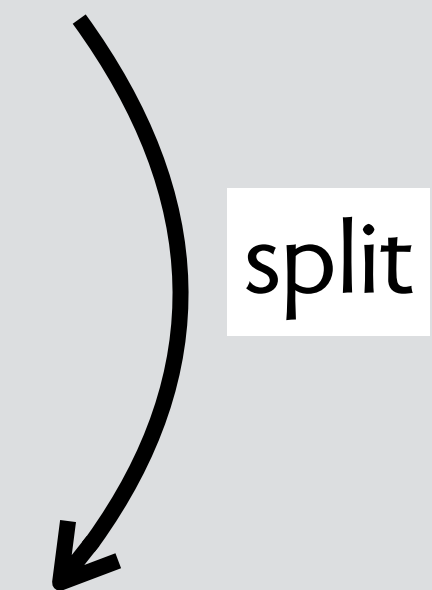
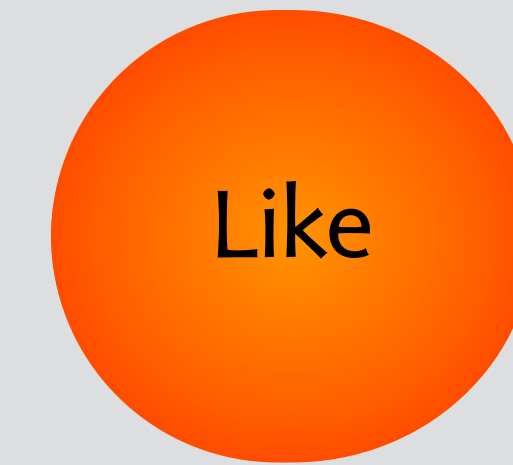
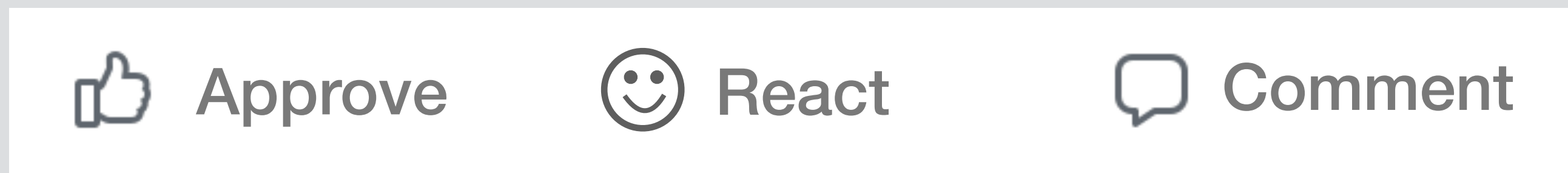
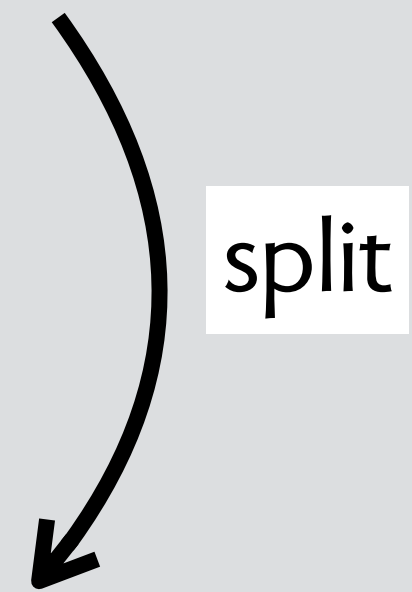
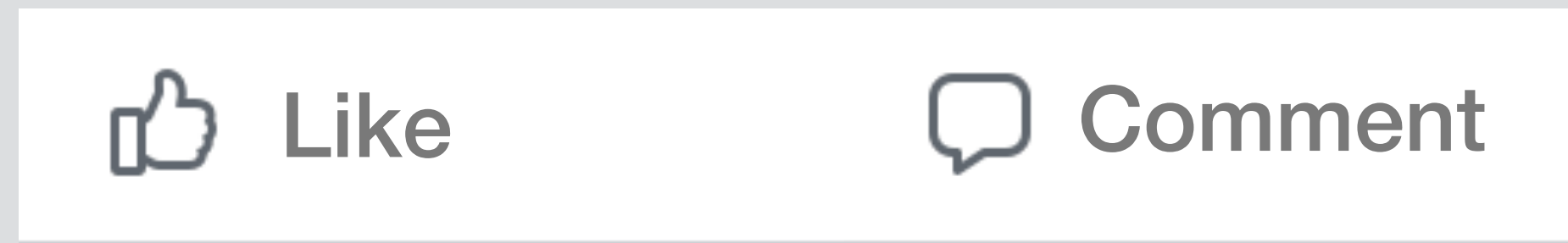


 Approve

 React

 Comment

a facebook split



**event deletion
in google calendar**



Arvind Satyanarayan

November 15, 2018 at 2:04 PM

Re: TALK: Monday 11-19-2018 Kanit (Ham) Wongsuphasawat: No...

[Details](#)

Cc: seminars@csail.mit.edu, HCI-Seminar@lists.csail.mit.edu



This message is from a mailing list.

[Unsubscribe](#) 

Despite some erroneous messages sent to this list accidentally, Kanit's talk is happening!
Please join us on Monday.

2:14 58'

🕒 🔇 📶 🔋 64%



HCI Semina

Wednesday, Dec



Daniel's Calendar

Delete

Duplicate

Copy to...

Help & feedback

2:14 58'

64%



HCI Semina
Wednesday, Dec



Daniel's Calendar

Delete

Duplicate

Copy to...

Help & feedback

seminar
announced as
email to listserv
with attached
calendar event

2:14 58'

64%



HCI Semina
Wednesday, Dec



Daniel's Calendar

Delete

Duplicate

Copy to...

Help & feedback

seminar
announced as
email to listserv
with attached
calendar event



event **installed**
automatically in
user's calendar

2:14 58'

64%



HCI Semina
Wednesday, Dec



Daniel's Calendar

Delete

Duplicate

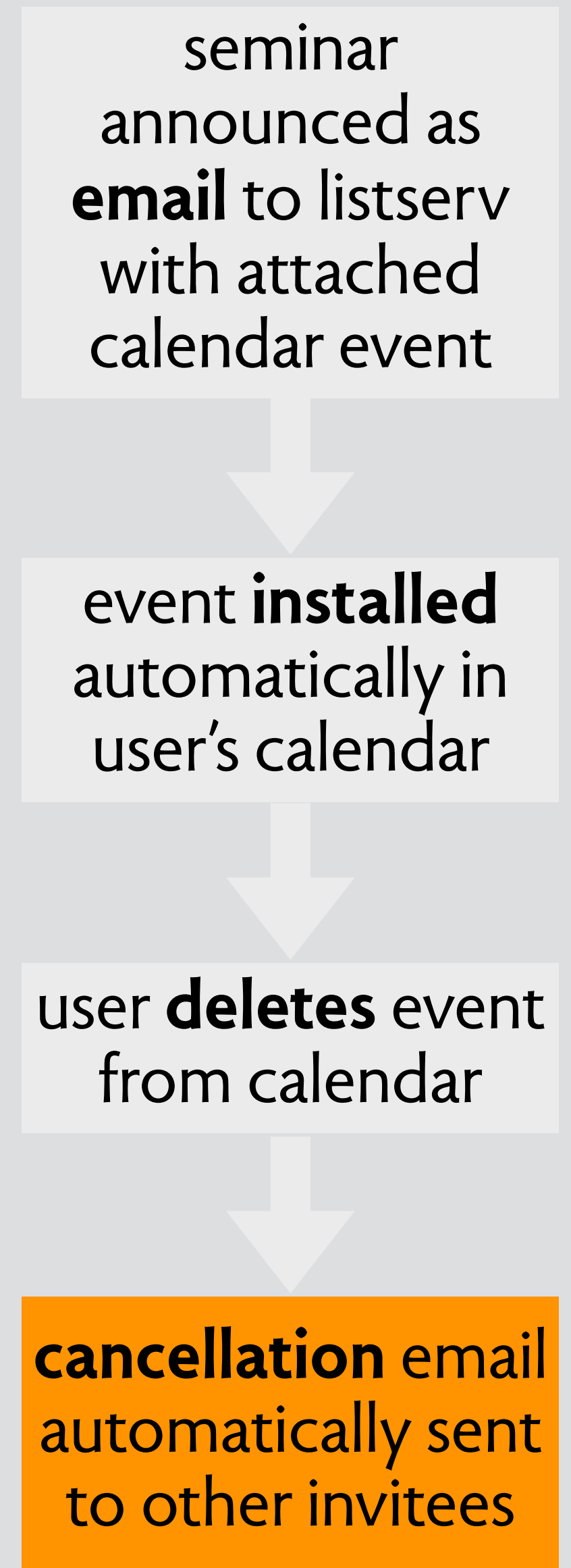
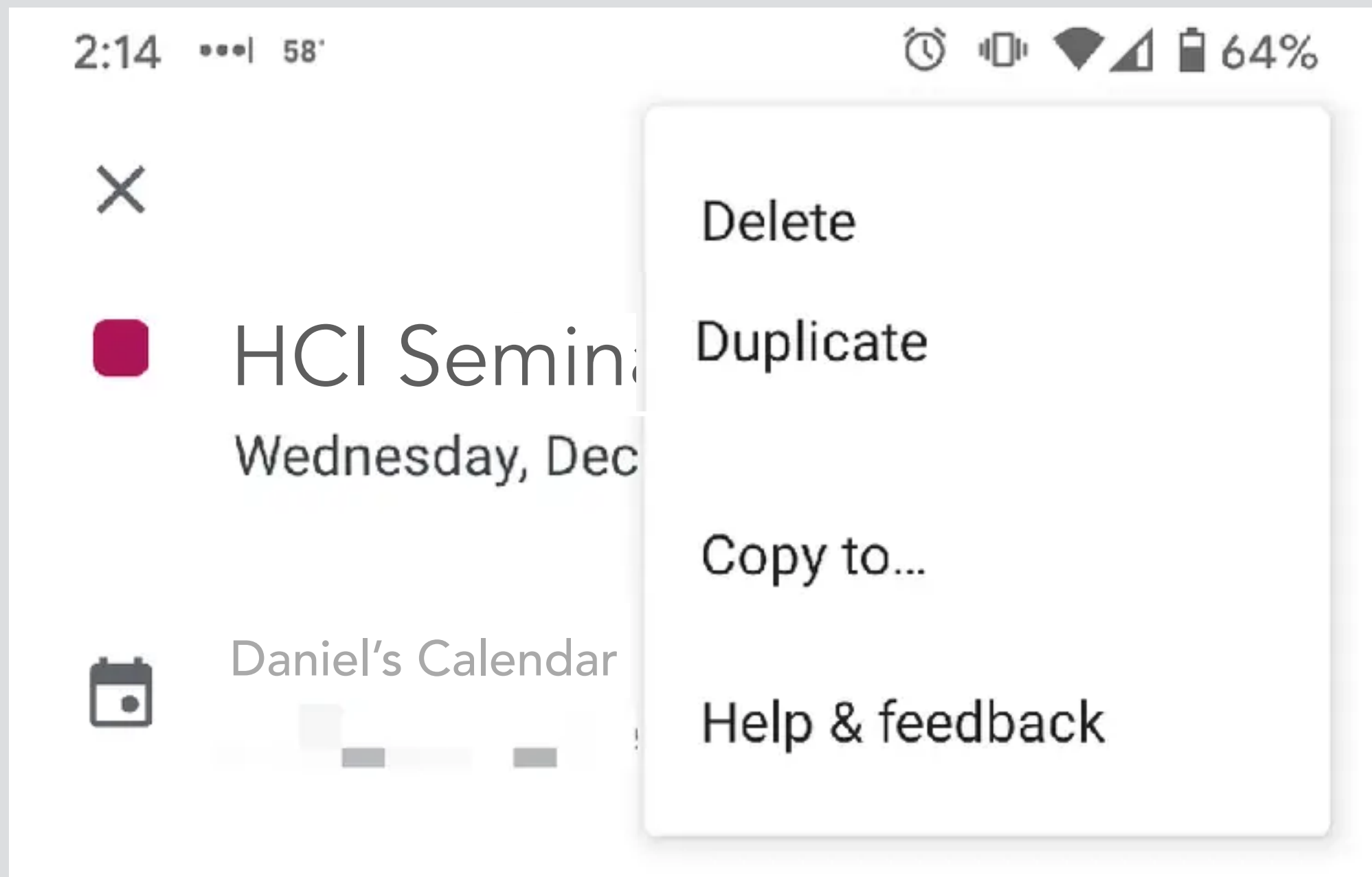
Copy to...

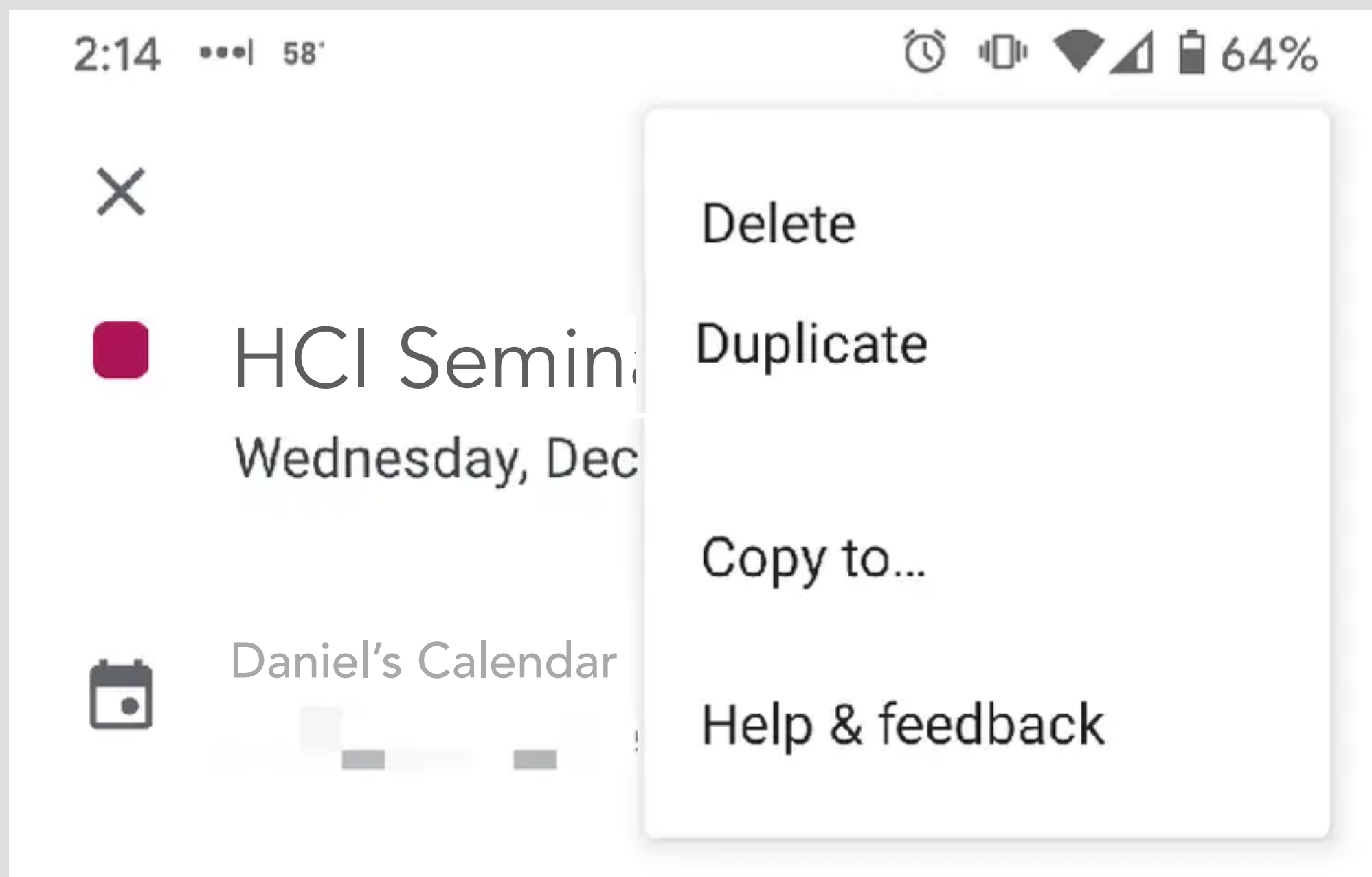
Help & feedback

seminar
announced as
email to listserv
with attached
calendar event

event **installed**
automatically in
user's calendar

user **deletes** event
from calendar





Canceling and deleting events in the Google Calendar mobile app is similar to on a desktop.

1. First, open Google Calendar.
2. Tap on the event you wish to cancel.
3. Press on the three dots in the top right corner of the event window.
4. Select Delete.
5. Tap Delete event. Google Calendar will send a cancellation email to the guests.

Mar 22, 2021

<https://wpamelia.com> › Blog

[How to Cancel an Event in Google Calendar - Amelia booking ...](#)





Are you sure you want to delete this event?

Deleting this meeting will remove it from your calendar and notify the invitees that this event has been deleted. You can't undo this action.

Cancel

Delete

a long time problem in iCal too
how to delete spam calendar events?

diagnosis?

diagnosis?



concept calendar

purpose record engagements

actions

createEvent (...): Event

deleteEvent (e: Event)

...

diagnosis?



concept calendar

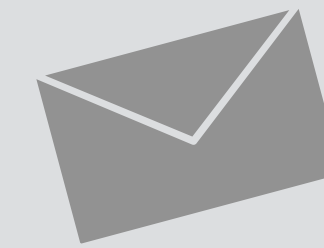
purpose record engagements

actions

createEvent (...): Event

deleteEvent (e: Event)

...



concept invitation

purpose coordinate events

actions

accept (e: Event)

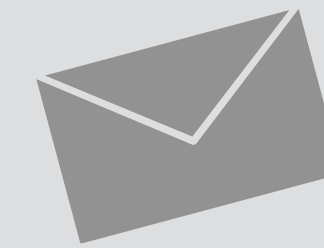
decline (e: Event)

...

diagnosis?



concept calendar
purpose record engagements
actions
createEvent (...): Event
deleteEvent (e: Event)
...

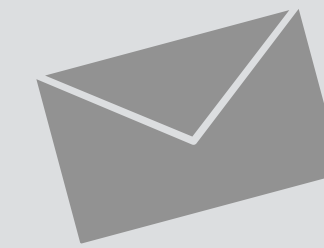


concept invitation
purpose coordinate events
actions
accept (e: Event)
decline (e: Event)
...

diagnosis?



concept calendar
purpose record engagements
actions
createEvent (...): Event
deleteEvent (e: Event)
...



concept invitation
purpose coordinate events
actions
accept (e: Event)
decline (e: Event)
...

unwanted
sync!

apple's solution



Are you sure you want to delete this event?

Deleting this event will notify the organizer that you're declining the event and deleting it from your calendar. You can't undo this action.

Cancel

Delete and Don't Notify

Delete and Notify

resolution to design problem
make sync optional

apple's solution



Are you sure you want to delete this event?

Deleting this event will notify the organizer that you're declining the event and deleting it from your calendar. You can't undo this action.

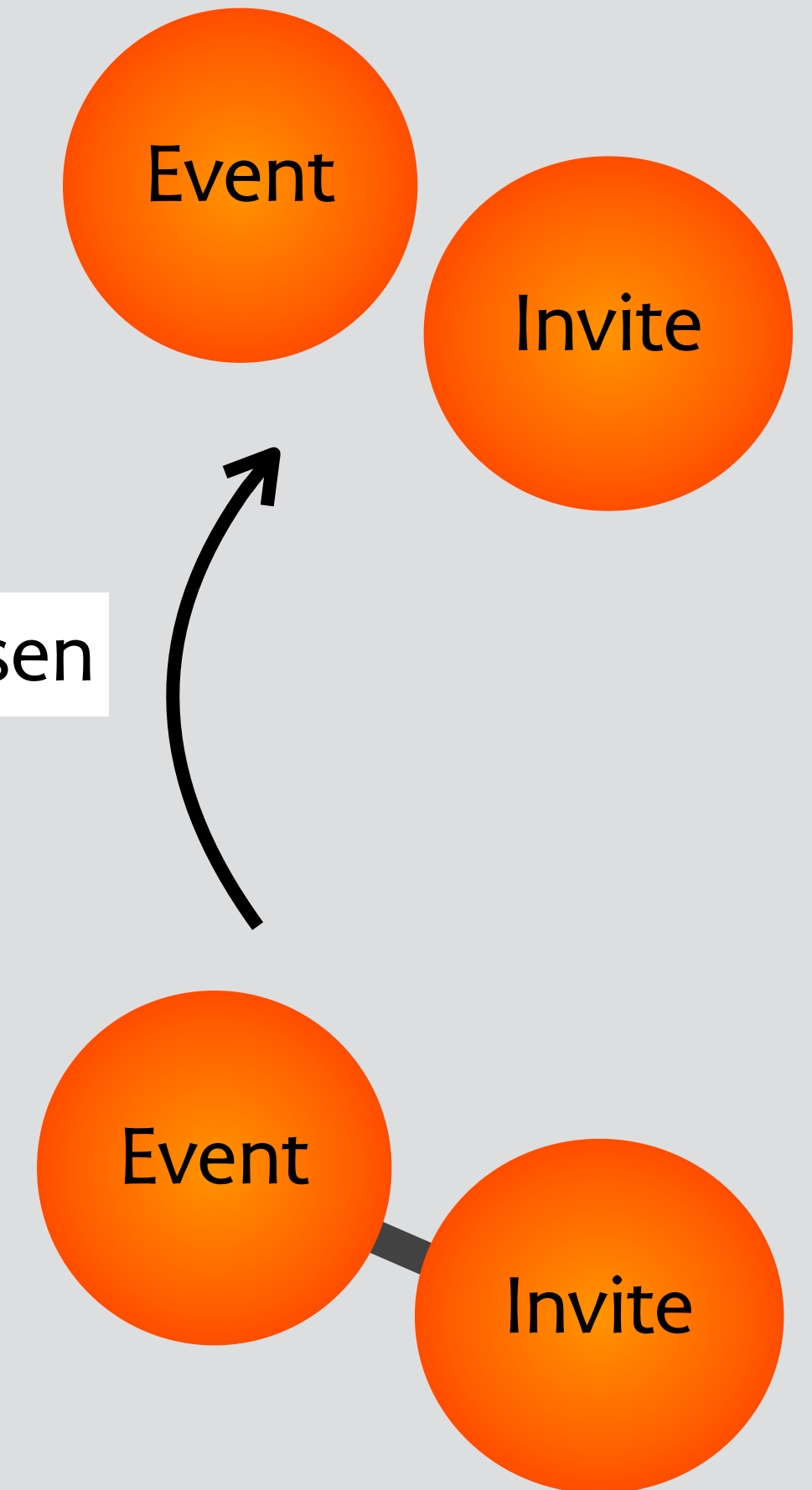
Cancel

Delete and Don't Notify

Delete and Notify

resolution to design problem
make sync optional

loosen



takeaways

takeaways

structure your software design with concepts

inventory the concepts, identify the critical ones
see if you can describe them fully independently
then formulate interactions as synchronizations

takeaways

structure your software design with concepts

inventory the concepts, identify the critical ones
see if you can describe them fully independently
then formulate interactions as synchronizations

apply design moves to explore new options

never a panacea, always a tradeoff

takeaways

structure your software design with concepts

inventory the concepts, identify the critical ones
see if you can describe them fully independently
then formulate interactions as synchronizations

apply design moves to explore new options

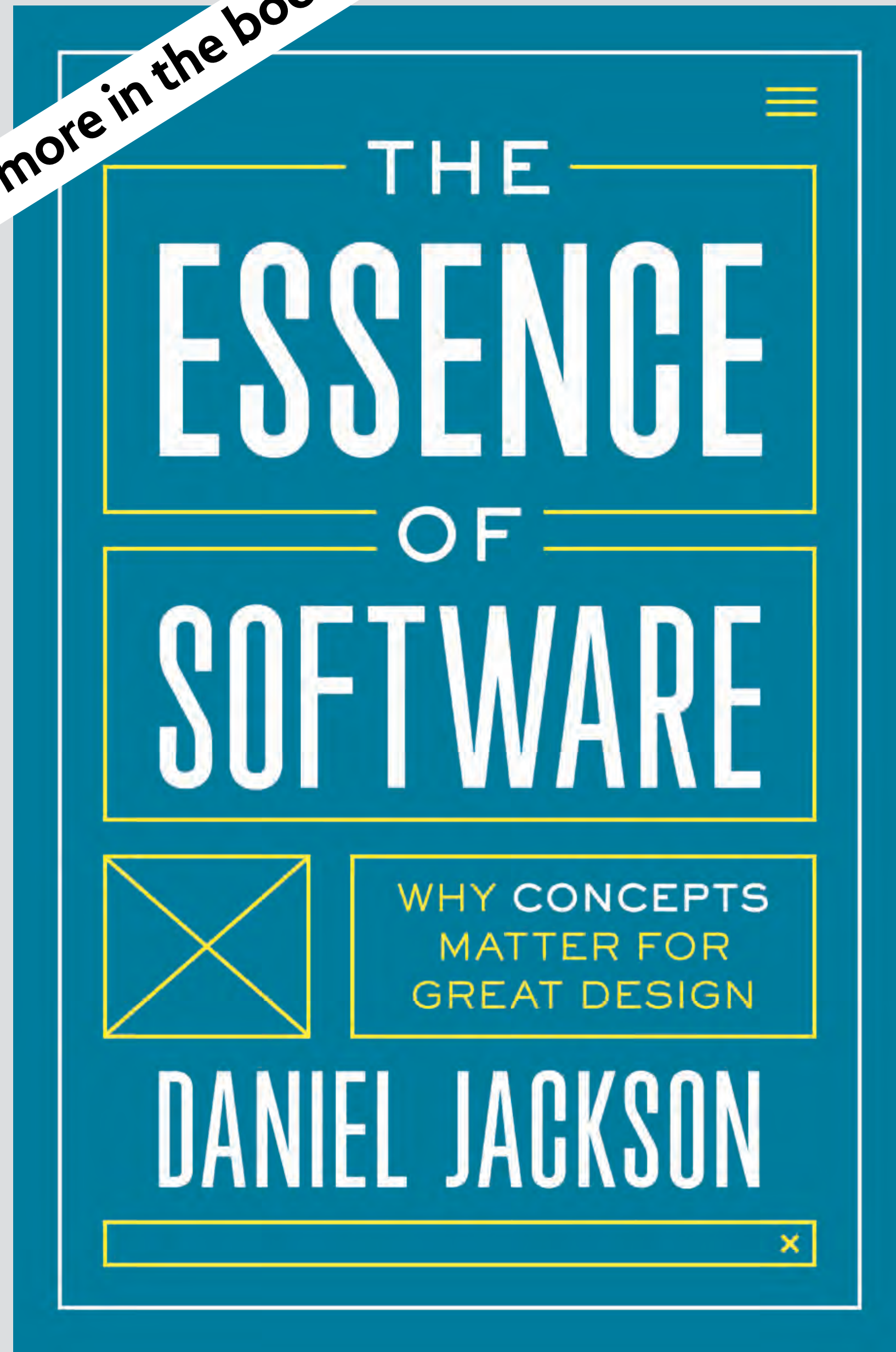
never a panacea, always a tradeoff

software concepts as patterns

only hinted at this, but equally important
don't reinvent the wheel!

express your design as sync of familiar concepts?

much more in the book



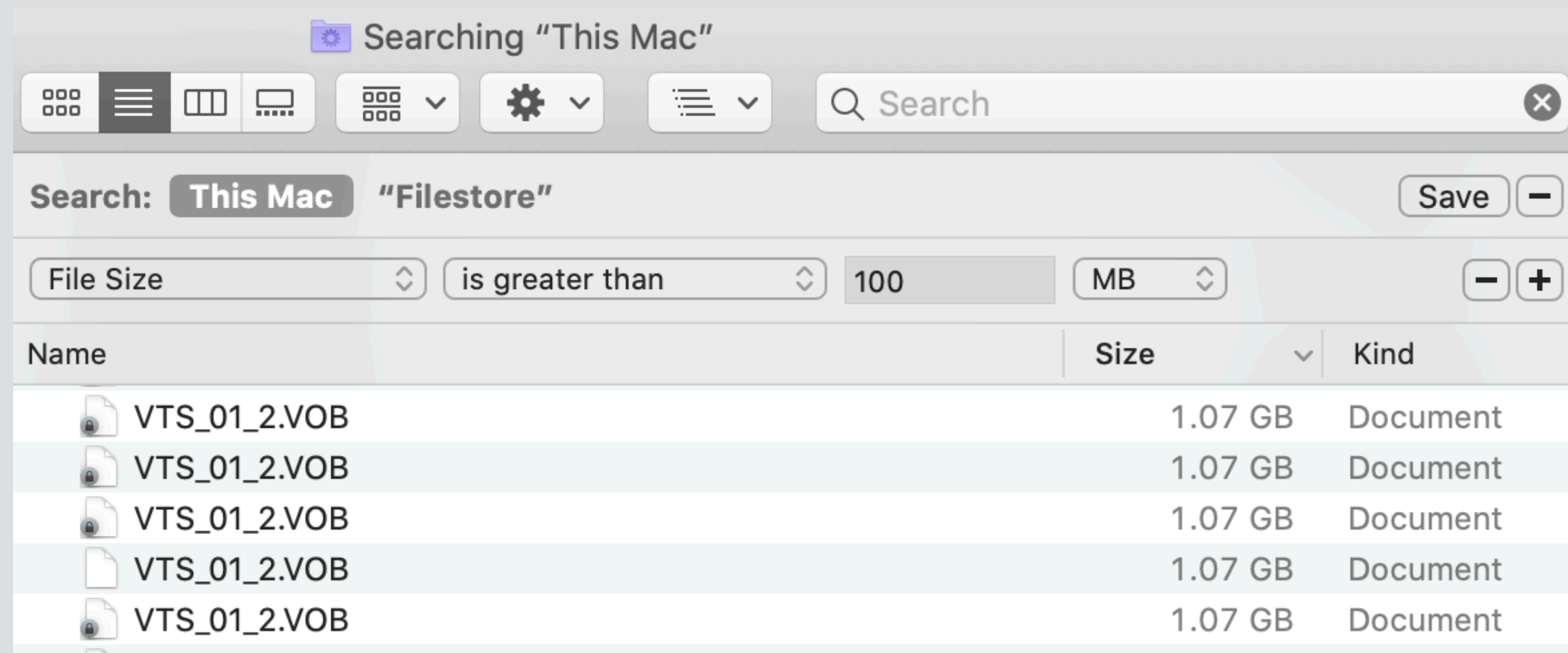
essenceofsoftware.com

newsletter
essenceofsoftware.com/subscribe

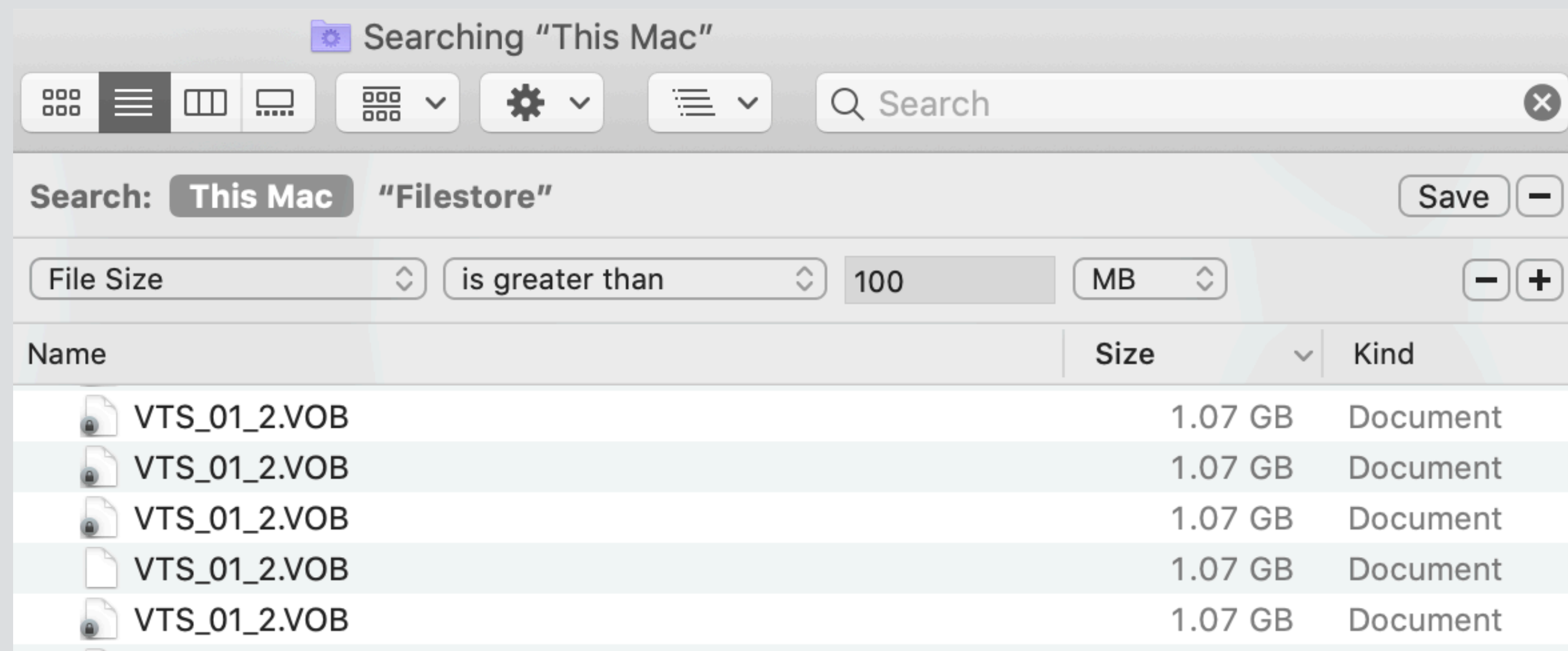
join the discussion
about concept design!
forum.softwareconcepts.io

**unused
slides**

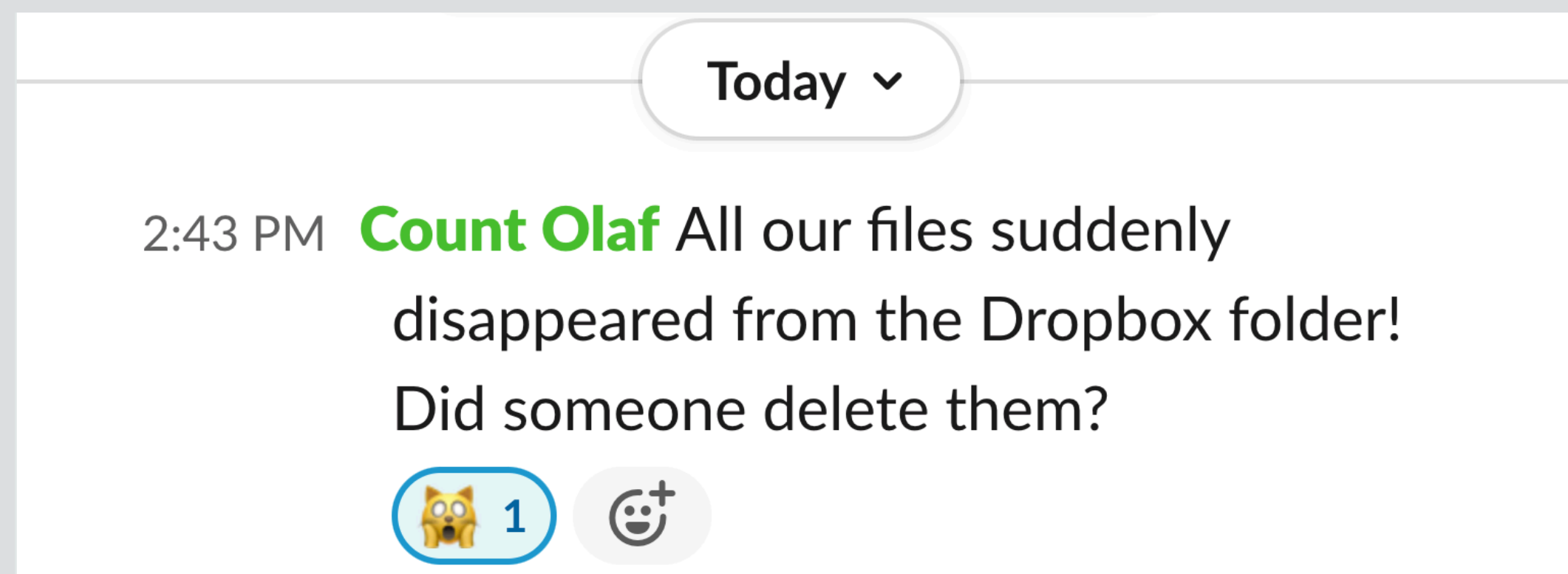
Dropbox & the case of the disappearing files



making space: list big files and delete the ones you don't recognize



making space: list big files and delete the ones you don't recognize



very unhappy colleagues

Quora

🔍 Search



Dropbox: [Edit](#)

Someone accidentally deleted thousands of files in my company Dropbox: how can I quickly undelete them? [Edit](#)

[Add Question Details](#)

[Comment](#) · [Share](#) · [Report](#) · [Options](#)



Ava is a party planner



Ava is a party planner



Bella is having a party




Ava is a party planner



Bella is having a party

The screenshot shows the Dropbox web interface. On the left is a sidebar with the following navigation items: Home, Files, All files, Shared, File requests, and Deleted files. The main content area features a search bar at the top, a notification bell, and a user profile icon labeled 'AA'. Below this is the 'Dropbox Overview' section, which includes a 'Show' button and a table of shared folders. The table has columns for selection, name, members, and actions.

<input type="checkbox"/>	Name ↑	Members ↓	⋮ ↓
<input type="checkbox"/>	 Bella Plan ☆	2 members	⋮



Ava is a party planner

Dropbox interface for user Ava (AA). The left sidebar contains navigation options: Home, Files, All files, Shared, File requests, and Deleted files. The main content area shows a search bar, a notification bell, and the user initials 'AA'. Below the search bar is the 'Dropbox Overview' section with a 'Show' button and a three-dot menu. A table lists a folder named 'Bella Plan' with a star icon, 2 members, and a three-dot menu.

<input type="checkbox"/>	Name ↑	Members ▾	⋮ ▾
<input type="checkbox"/>	Bella Plan ☆	2 members	⋮



Bella is having a party

Dropbox interface for user Bella (BB). The left sidebar contains navigation options: Home, Files, All files, Shared, File requests, and Deleted files. The main content area shows a search bar, a notification bell, and the user initials 'BB'. Below the search bar is the 'Dropbox Overview' section with a 'Show' button and a three-dot menu. A table lists a folder named 'Bella Plan' with a star icon, 2 members, and a three-dot menu.

<input type="checkbox"/>	Name ↑	Members ▾	⋮ ▾
<input type="checkbox"/>	Bella Plan ☆	2 members	⋮



Ava is a party planner

Dropbox

Overview [Show](#) ⋮

<input type="checkbox"/>	Name ↑	Members ▾	☰ ▾
<input type="checkbox"/>	Bella Plan ☆	2 members	⋮



Bella is having a party

Dropbox

Overview [Show](#) ⋮

<input type="checkbox"/>	Name ↑	Members ▾	☰ ▾
<input type="checkbox"/>	Bella Plan ☆	2 members	⋮

- Share
- Download
- Send with Transfer
- Request files
- Star
- Rewind
- Rename**
- Move
- Copy
- Delete
- Events
- Pin to Dropbox



Ava is a party planner

Dropbox interface for user Ava (AA). The left sidebar contains navigation options: Home, Files, All files, Shared, File requests, and Deleted files. The main content area shows a search bar, a notification bell, and the user's initials 'AA'. Below the search bar is the 'Dropbox Overview' section with a 'Show' button and a three-dot menu. A table lists shared folders:

<input type="checkbox"/>	Name ↑	Members ▾	⋮ ▾
<input type="checkbox"/>	Bella Plan ☆	2 members	⋮



Bella is having a party

Dropbox interface for user Bella (BB). The left sidebar contains navigation options: Home, Files, All files, Shared, File requests, and Deleted files. The main content area shows a search bar, a notification bell, and the user's initials 'BB'. Below the search bar is the 'Dropbox Overview' section with a 'Show' button and a three-dot menu. A table lists shared folders:

<input type="checkbox"/>	Name ↑	Members ▾	⋮ ▾
<input type="checkbox"/>	My Party Plan ☆	2 members	⋮



Ava is a party planner

Dropbox

Overview [Show](#) ⋮

<input type="checkbox"/>	Name ↑	Members ▾	⋮
<input type="checkbox"/>	Bella Plan ☆	2 members	⋮

does the name change for Ava too?



Bella is having a party

Dropbox

Overview [Show](#) ⋮

<input type="checkbox"/>	Name ↑	Members ▾	⋮
<input type="checkbox"/>	My Party Plan ☆	2 members	⋮



Ava is a party planner

The screenshot shows Ava's Dropbox interface. The left sidebar contains navigation options: Home, Files, All files, Shared, File requests, and Deleted files. The main content area has a search bar, a notification bell, and a profile icon labeled 'AA'. Below the search bar is the 'Dropbox Overview' section with a 'Show' button and a three-dot menu. A table lists folders with columns for checkboxes, 'Name', 'Members', and a three-dot menu. One folder is visible: 'Bella Plan' with a star icon and '2 members'. A red callout box with a pointer to the folder name contains the text 'does the name change for Ava too?'.



Bella is having a party

The screenshot shows Bella's Dropbox interface. The left sidebar contains navigation options: Home, Files, All files, Shared, File requests, and Deleted files. The main content area has a search bar, a notification bell, and a profile icon labeled 'BB'. Below the search bar is the 'Dropbox Overview' section with a 'Show' button and a three-dot menu. A table lists folders with columns for checkboxes, 'Name', 'Members', and a three-dot menu. One folder is visible: 'My Party Plan' with a star icon and '2 members'.

answer: it depends

if Ava just shares Bella Plan with Bella

and Bella renamed the folder, Ava sees no change

if Ava shared a folder containing Bella Plan

then Ava does see the change

Bella deletes
Bella Plan from
shared folder
Bella Party

Delete folder?



Are you sure you want to delete **Bella Plan** from the shared folder 'Bella Party'?

Cancel

Delete

Bella deletes
Bella Plan from
shared folder
Bella Party

Delete folder?



Are you sure you want to delete **Bella Plan** from the shared folder 'Bella Party'?

Cancel

Delete

Bella deletes
shared folder
Bella Party

Remove shared folder?

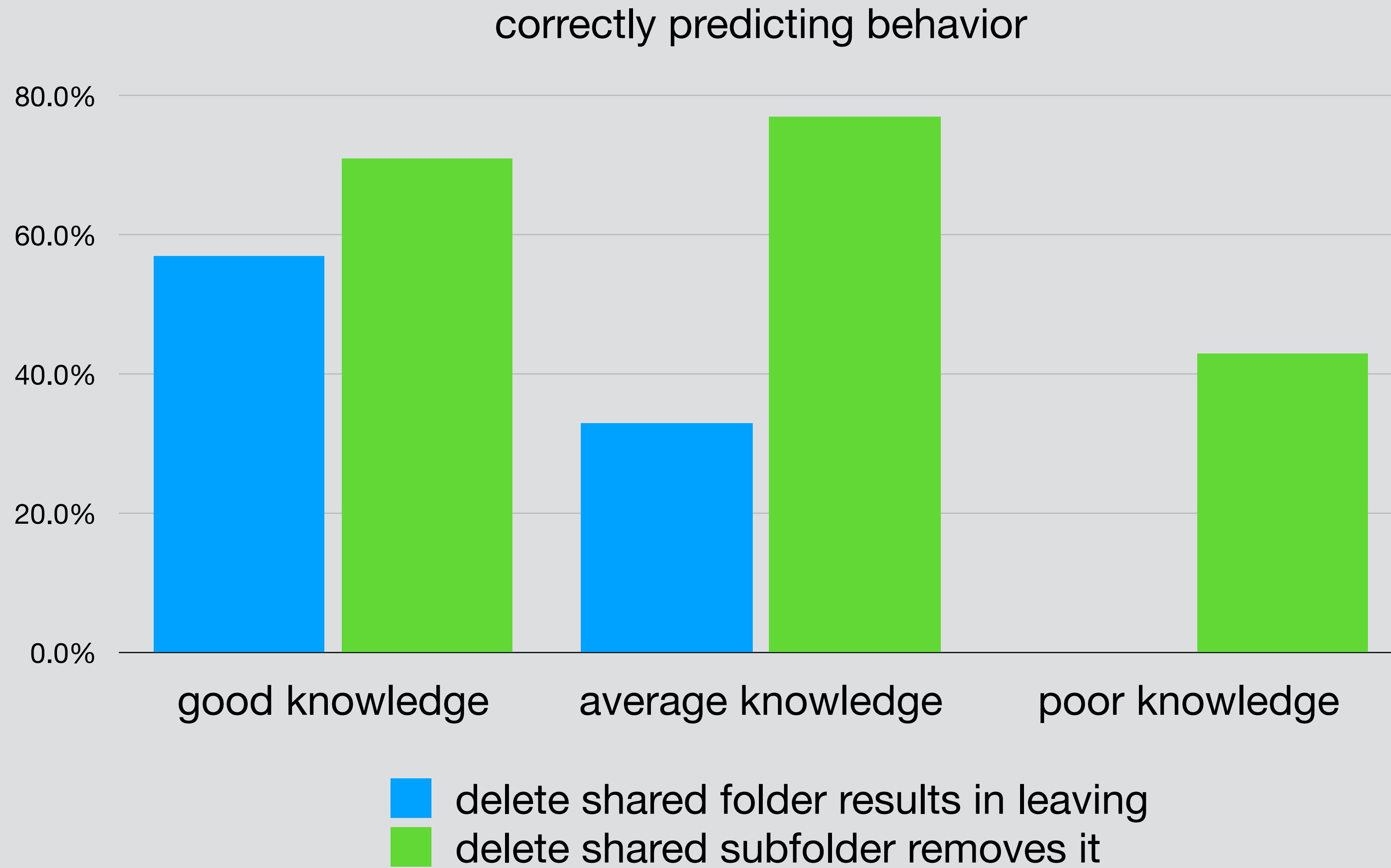


Are you sure you want to remove the shared folder **Bella Party** from your Dropbox? This folder will stay shared with any existing members. You can re-add it later.

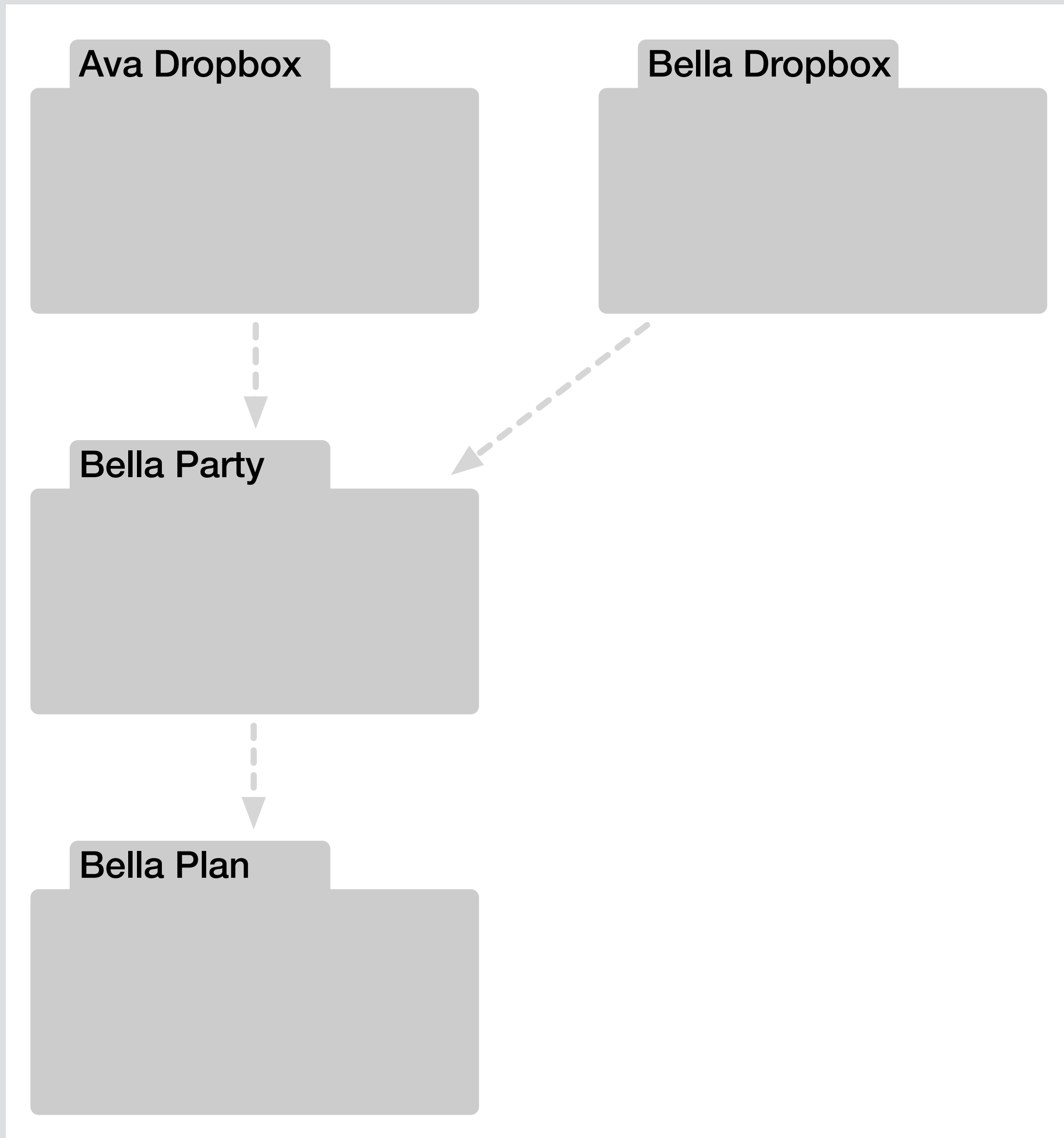
Cancel

Remove

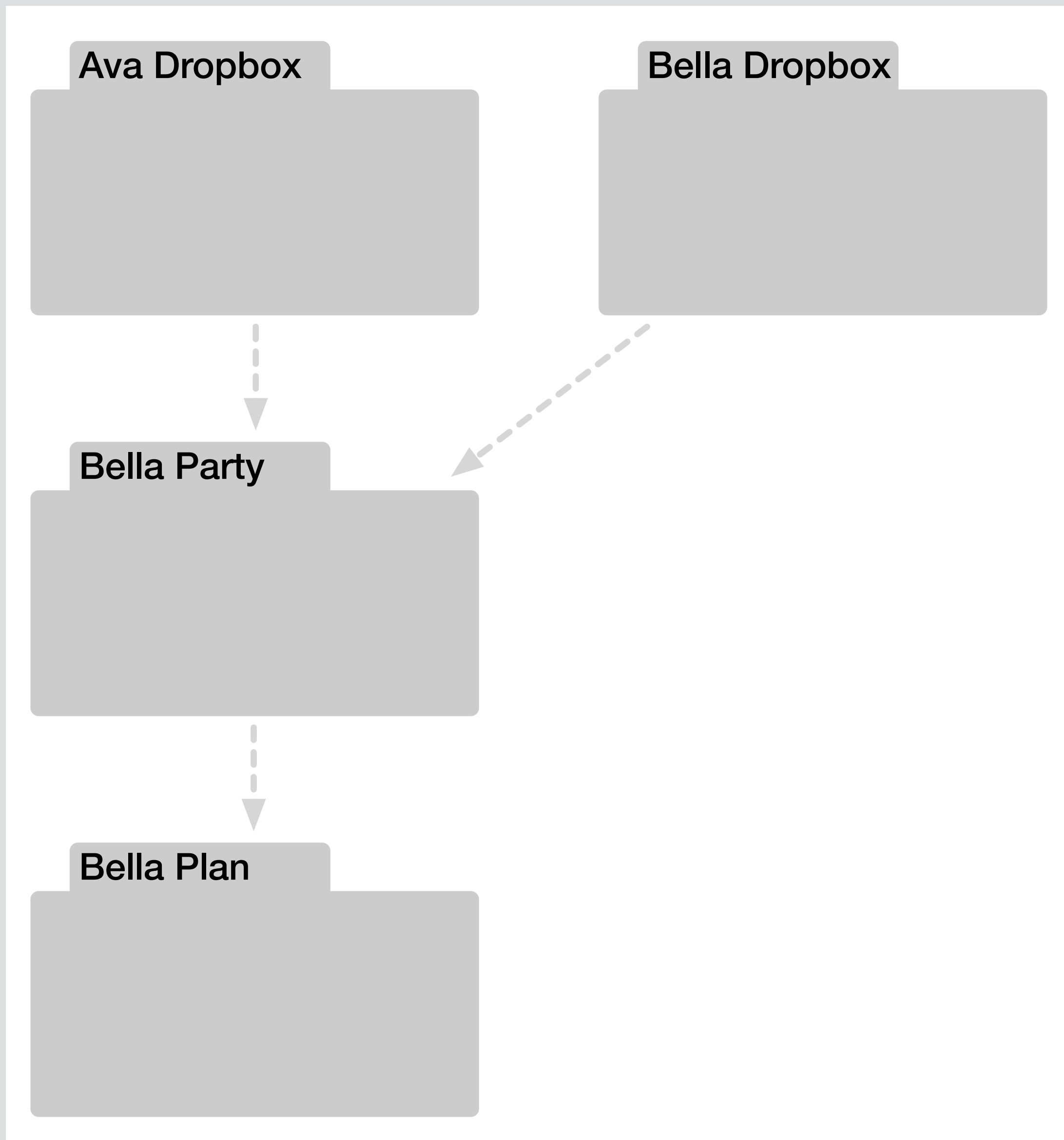
survey of dropbox users (MIT CS undergrads)



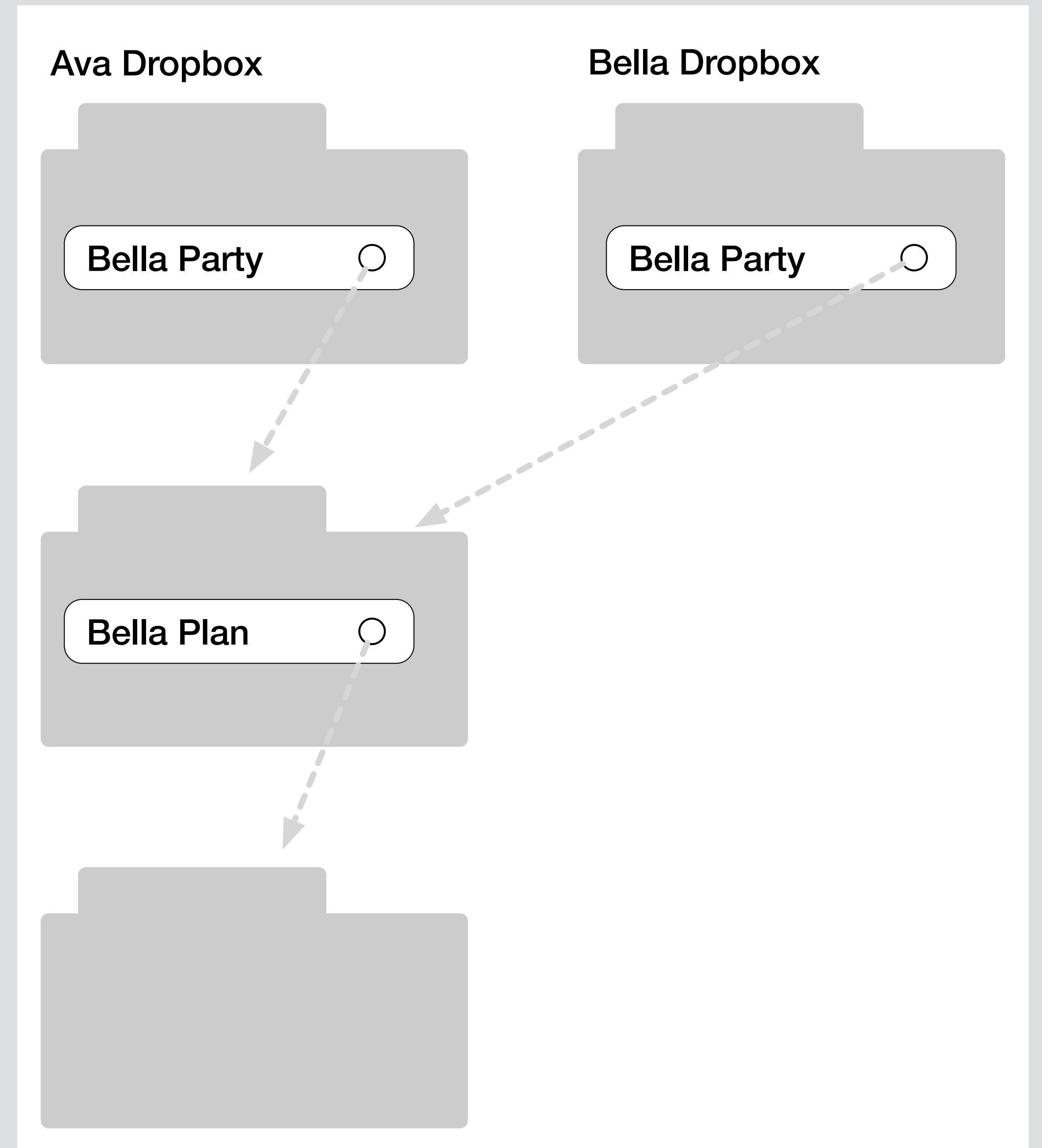
Kelly Zhang



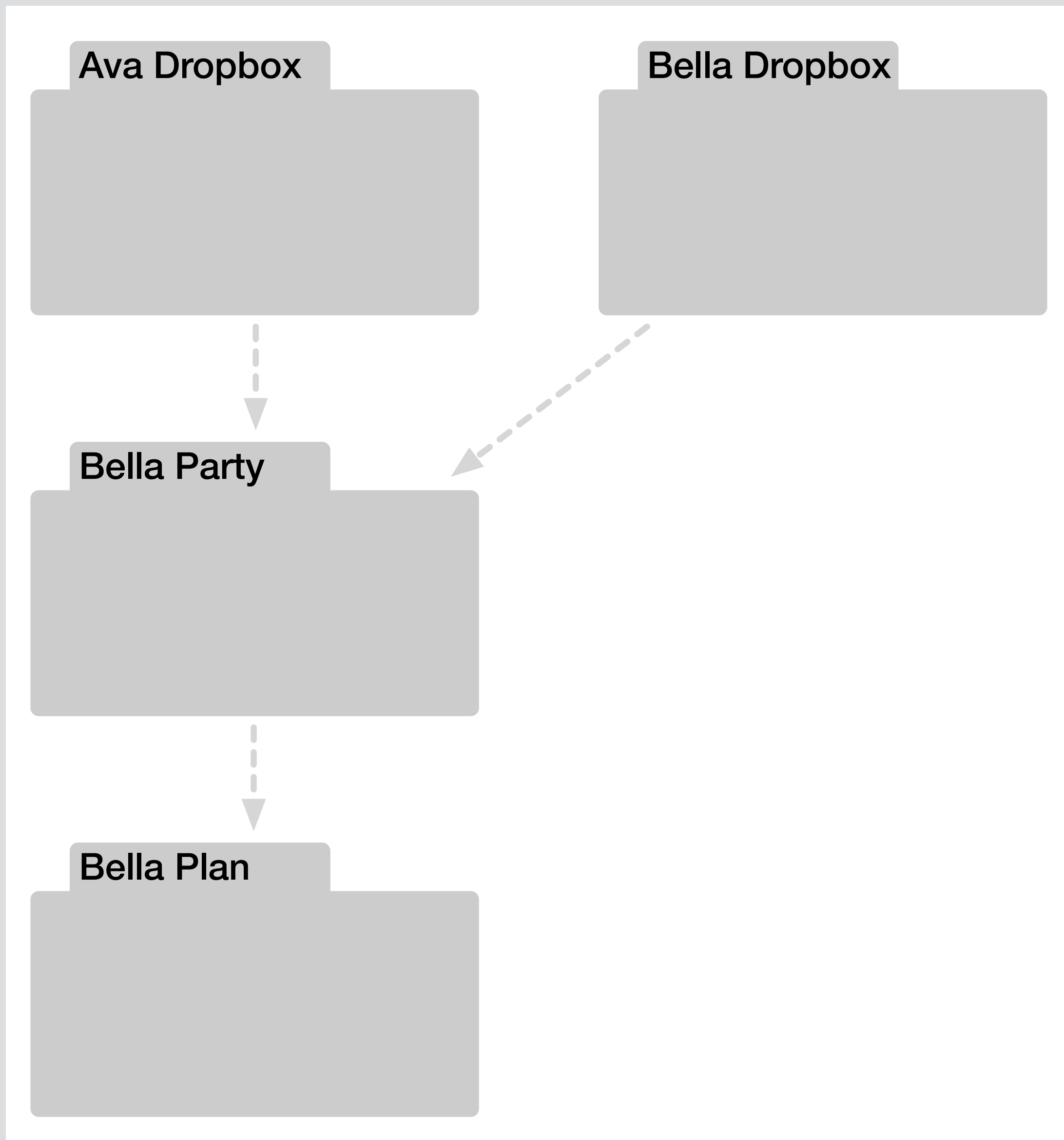
how many users believe Dropbox is structured



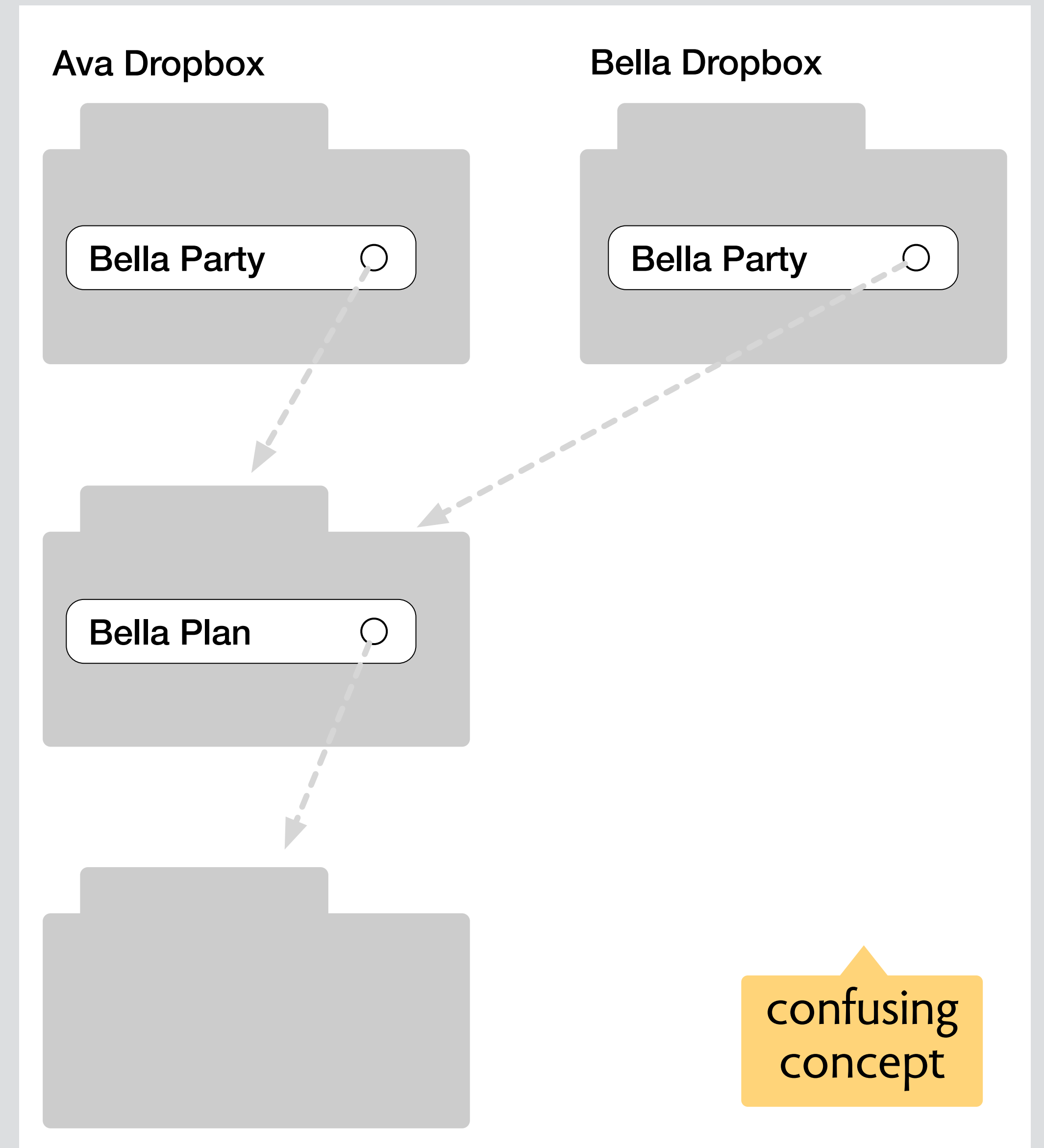
how many users believe Dropbox is structured



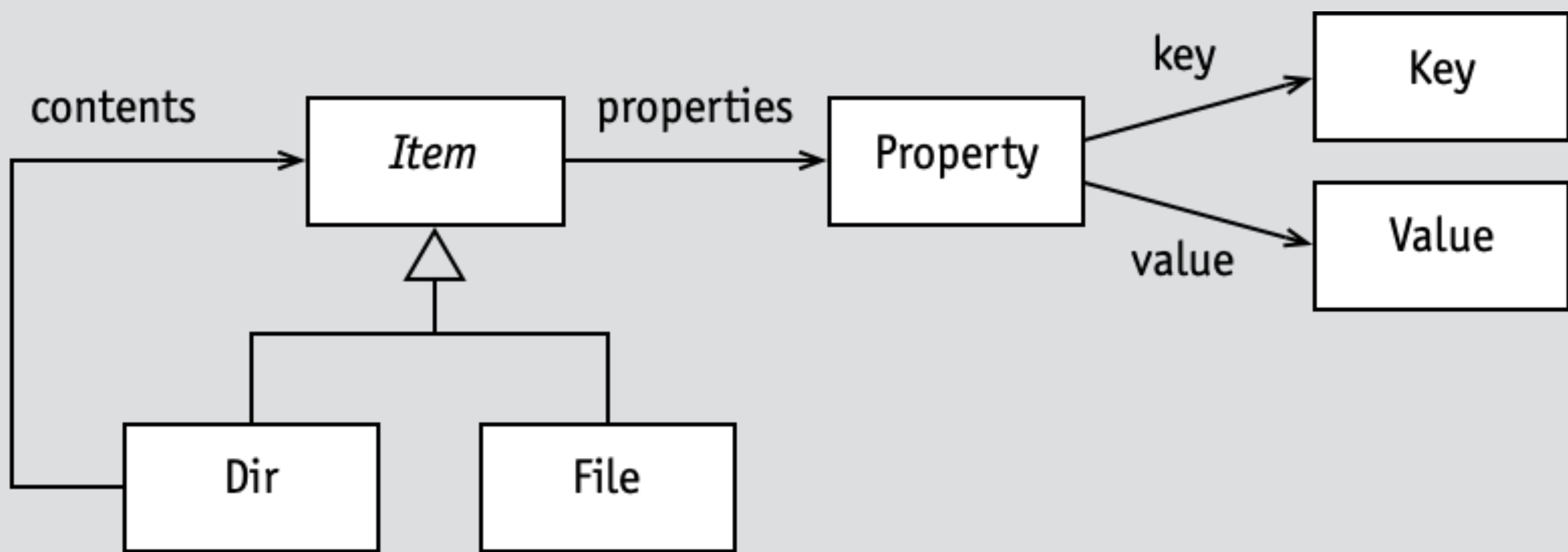
how Dropbox is actually structured

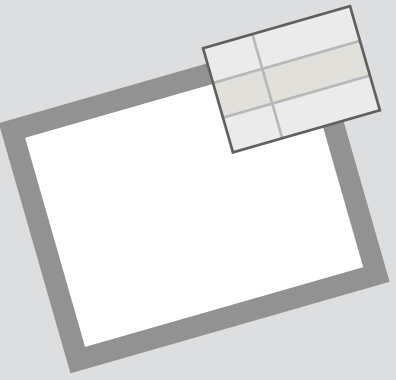


how many users believe Dropbox is structured



how Dropbox is actually structured





concept metadata

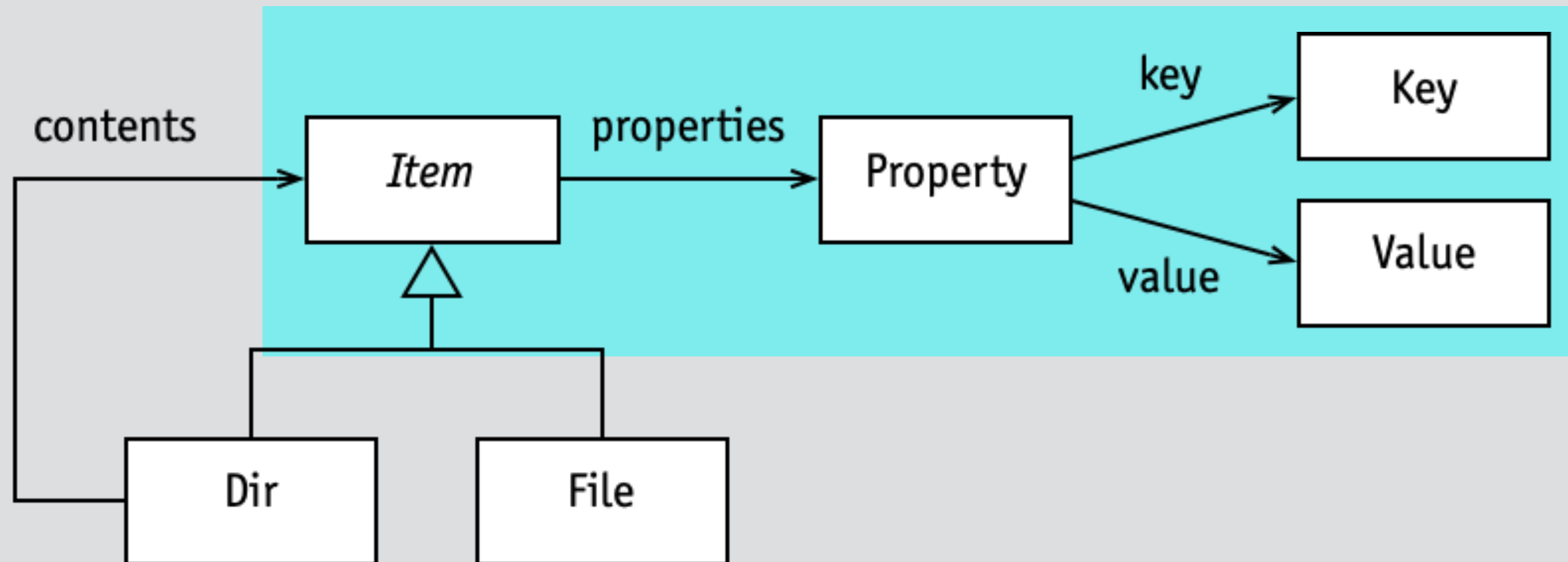
purpose tag items with properties for lookup

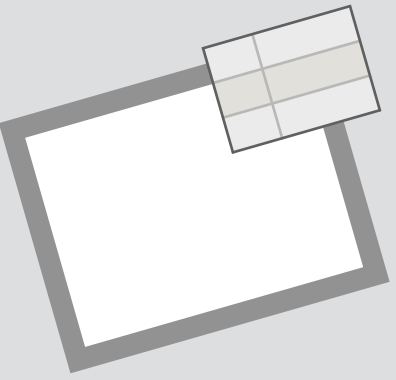
state

properties: Item -> set Property

key: Property -> one Key

val: Property -> one Val





concept metadata

purpose tag items with properties for lookup

state

properties: Item -> set Property

key: Property -> one Key

val: Property -> one Val



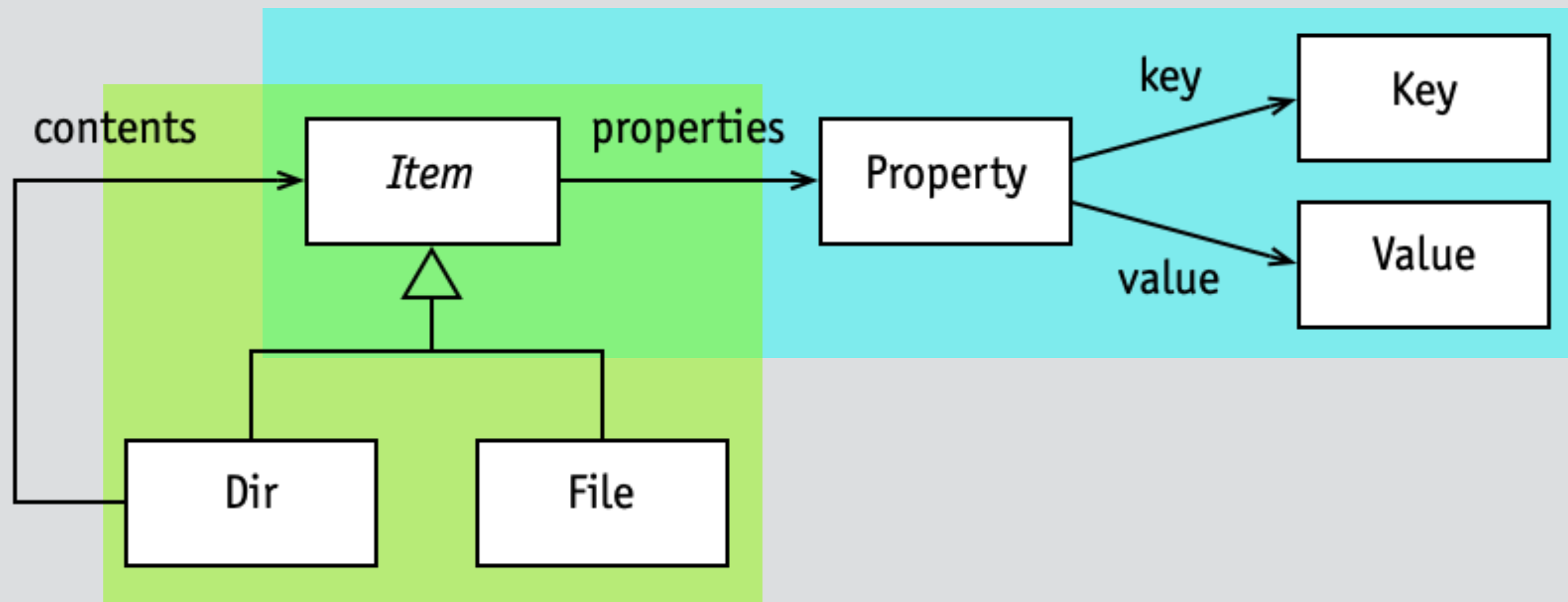
concept folder

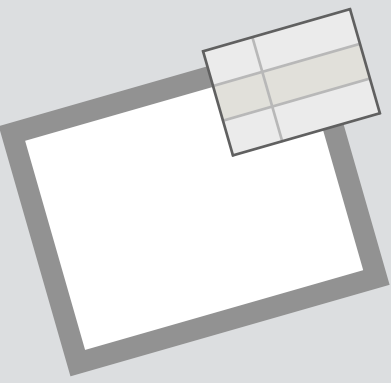
purpose organize items into disjoint categories

state

contents: Folder -> set (Folder + Item)

root: one Folder





concept metadata

purpose tag items with properties for lookup

state

properties: Item -> set Property

key: Property -> one Key

val: Property -> one Val



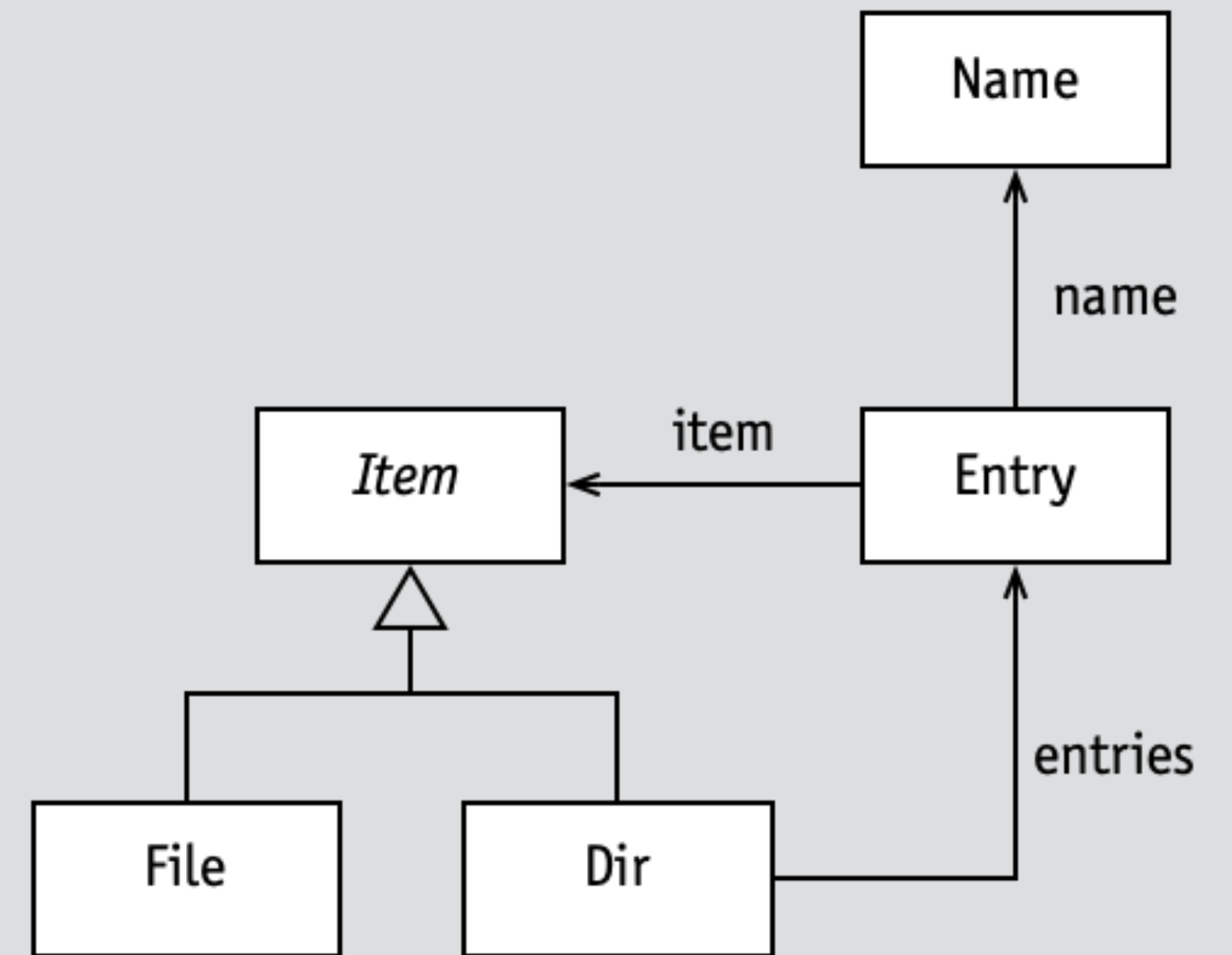
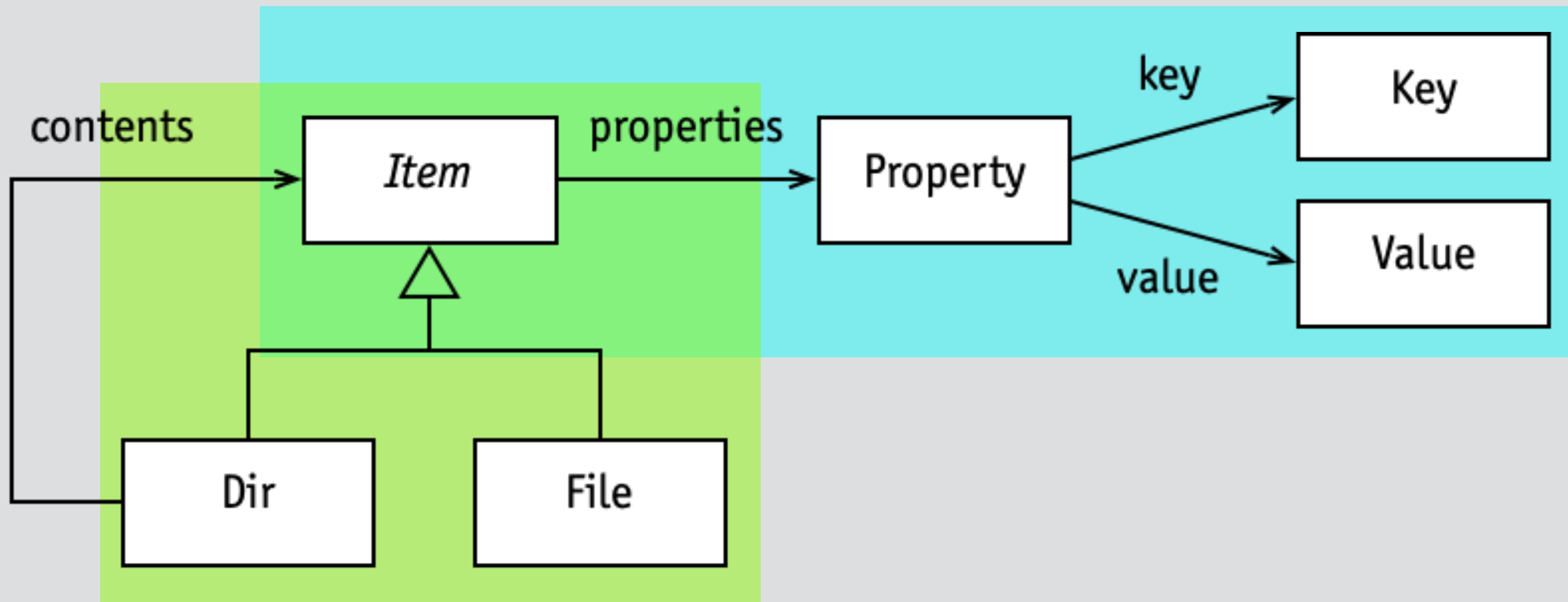
concept folder

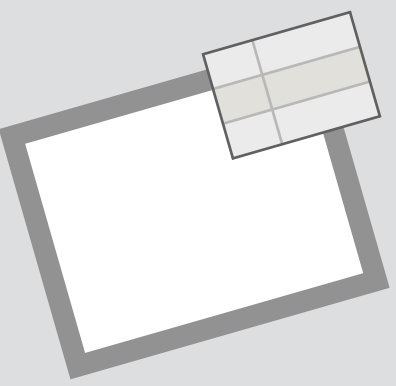
purpose organize items into disjoint categories

state

contents: Folder -> set (Folder + Item)

root: one Folder





concept metadata

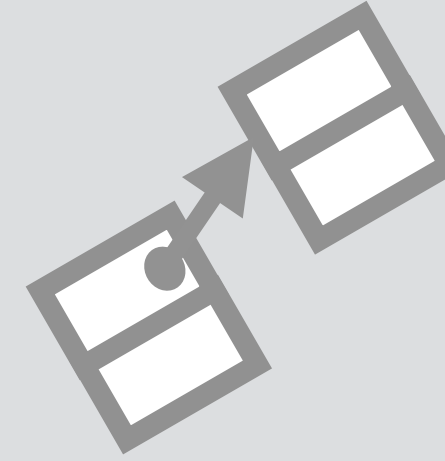
purpose tag items with properties for lookup

state

properties: Item -> set Property

key: Property -> one Key

val: Property -> one Val



concept unix directory

purpose organize items into overlapping categories

state

entries: Dir -> set Entry

name: Entry -> one Name

item: Entry -> one (File + Dir)



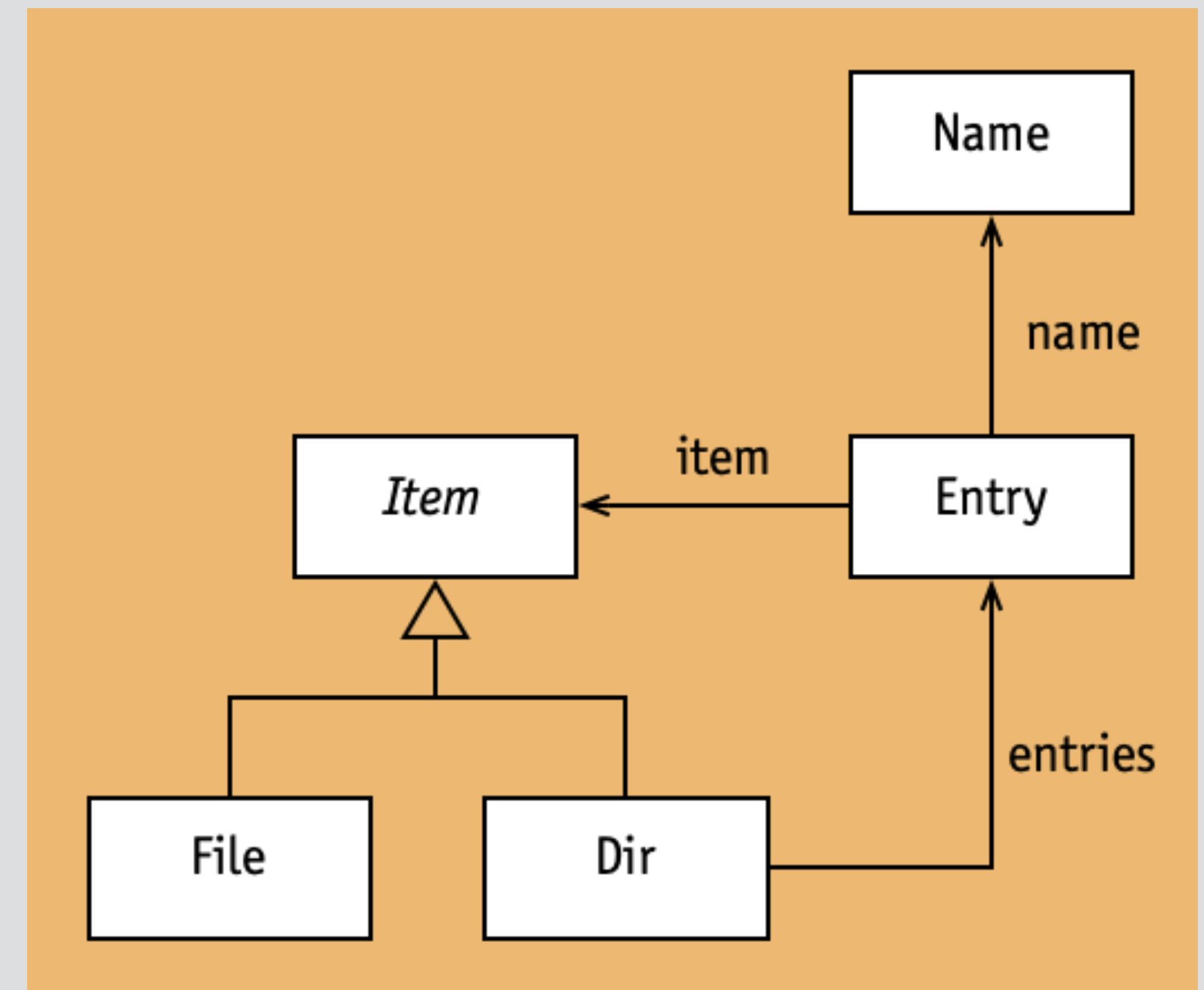
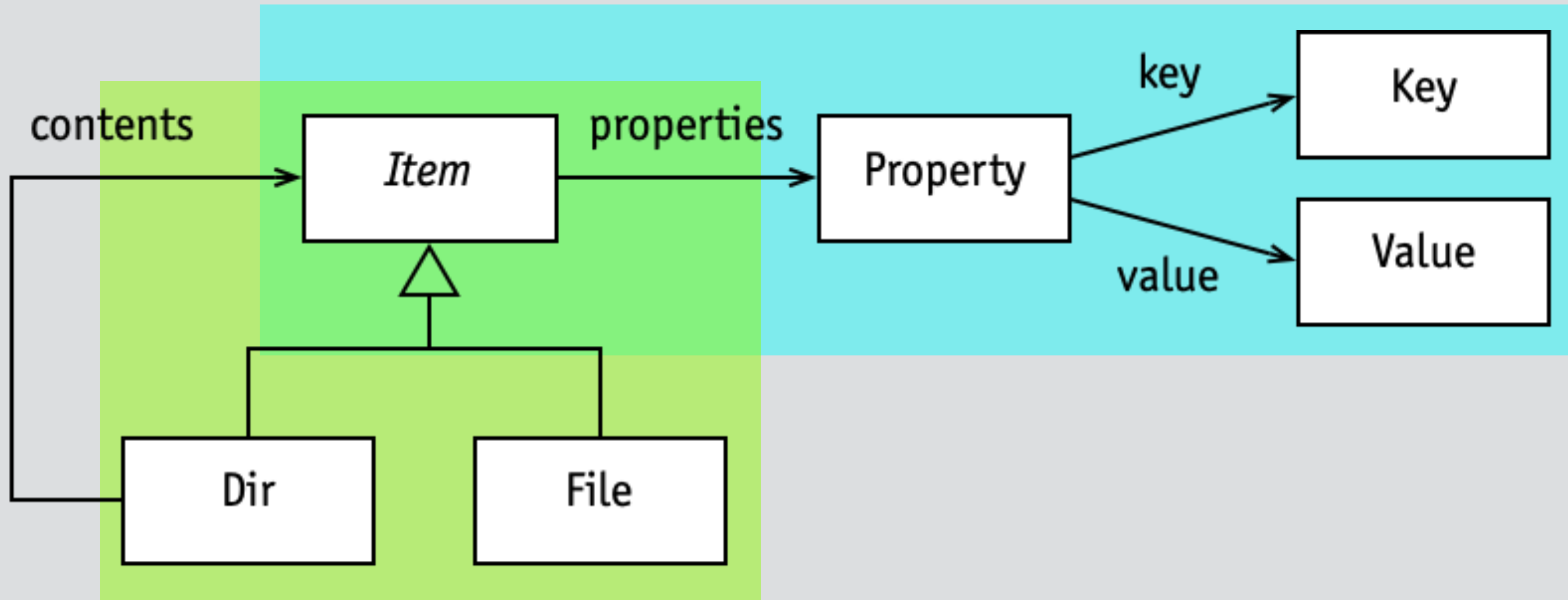
concept folder

purpose organize items into disjoint categories

state

contents: Folder -> set (Folder + Item)

root: one Folder



Twitter
& the case of the
surprised first lady



Andy Ostroy 

@AndyOstroy



Seems the only #Wall @realDonaldTrump's built is the one between him and @FLOTUS #Melania #trump



♥ 8,221 8:15 PM - May 2, 2017



💬 4,022 people are talking about this





Andy Ostroy ✓

@AndyOstroy



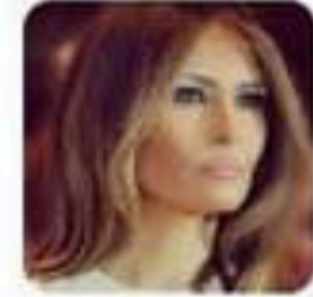
Seems the only #Wall @realDonaldTrump's built is the one between him and @FLOTUS #Melania #trump



♥ 8,221 8:15 PM - May 2, 2017

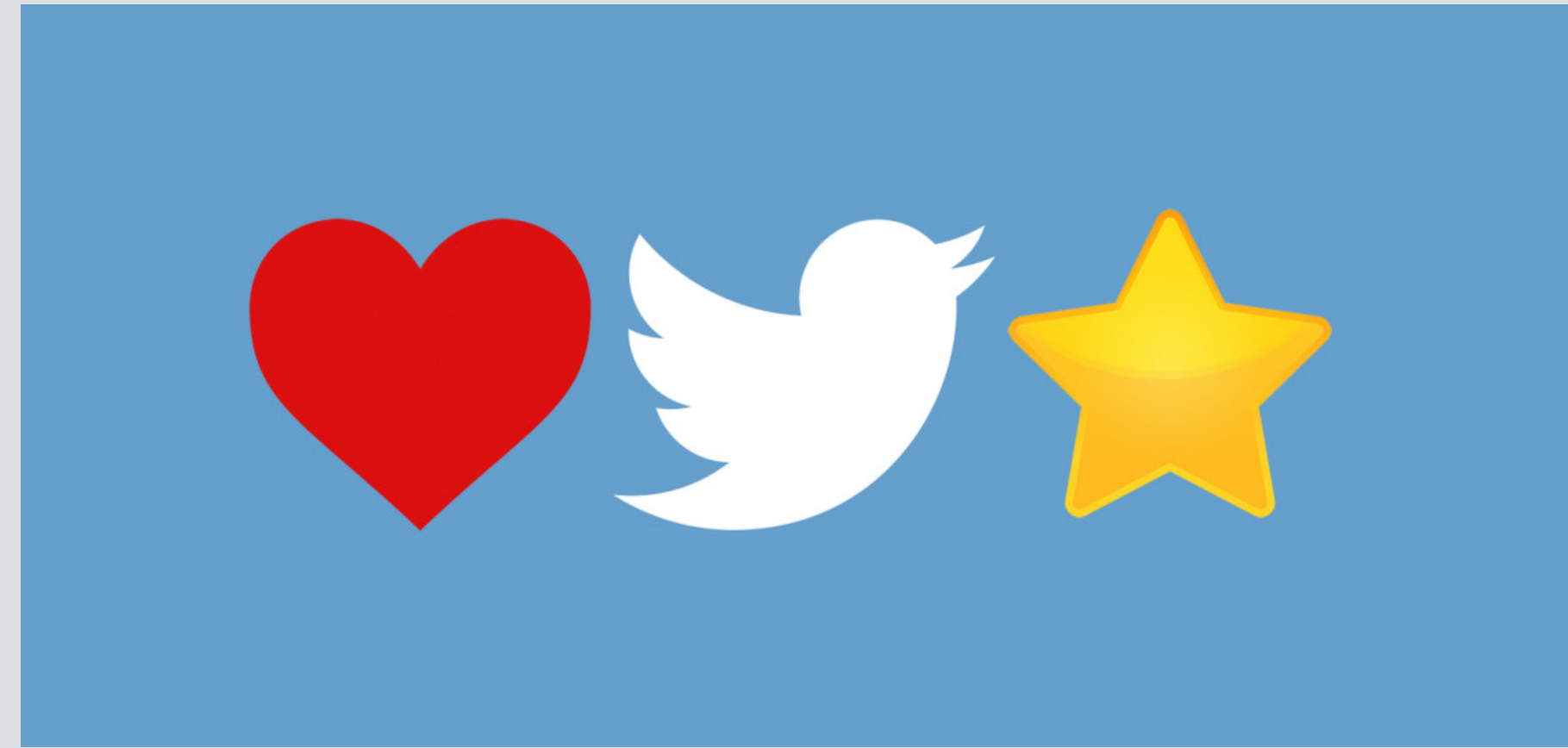


💬 4,022 people are talking about this

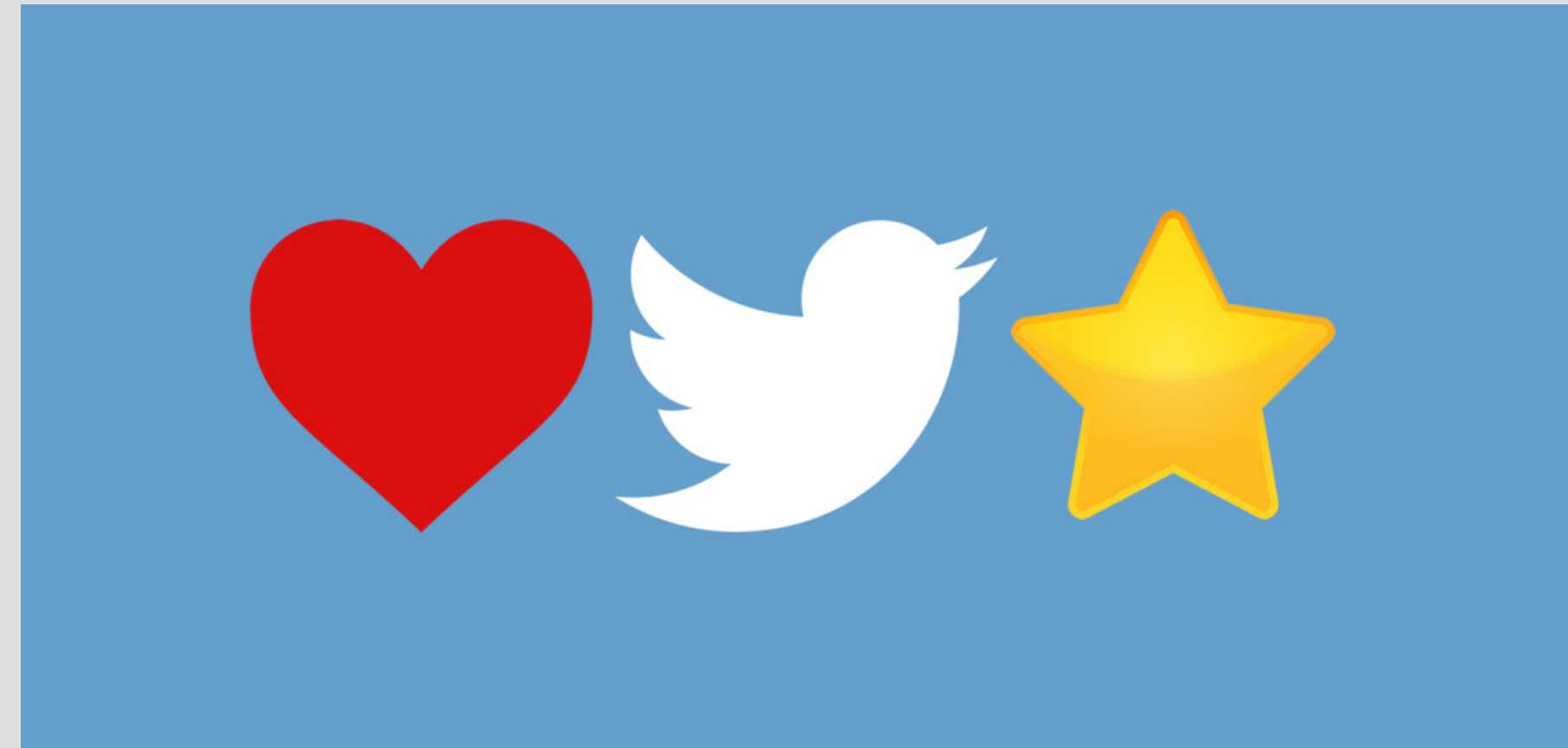


MELANIA TRUMP liked your Tweet

Seems the only #Wall @realDonaldTrump's built is the one between him and @FLOTUS #Melania #trump pic.twitter.com/XiNd2jiLUF



Nov 2, 2015: Twitter changes Favorite (Star) to Like (Heart)



Nov 2, 2015: Twitter changes Favorite (Star) to Like (Heart)

We are changing our star icon for favorites to a heart and we'll be calling them likes... **We know that at times the star could be confusing, especially to newcomers.** You might like a lot of things, but not everything can be your favorite. *Twitter press release*

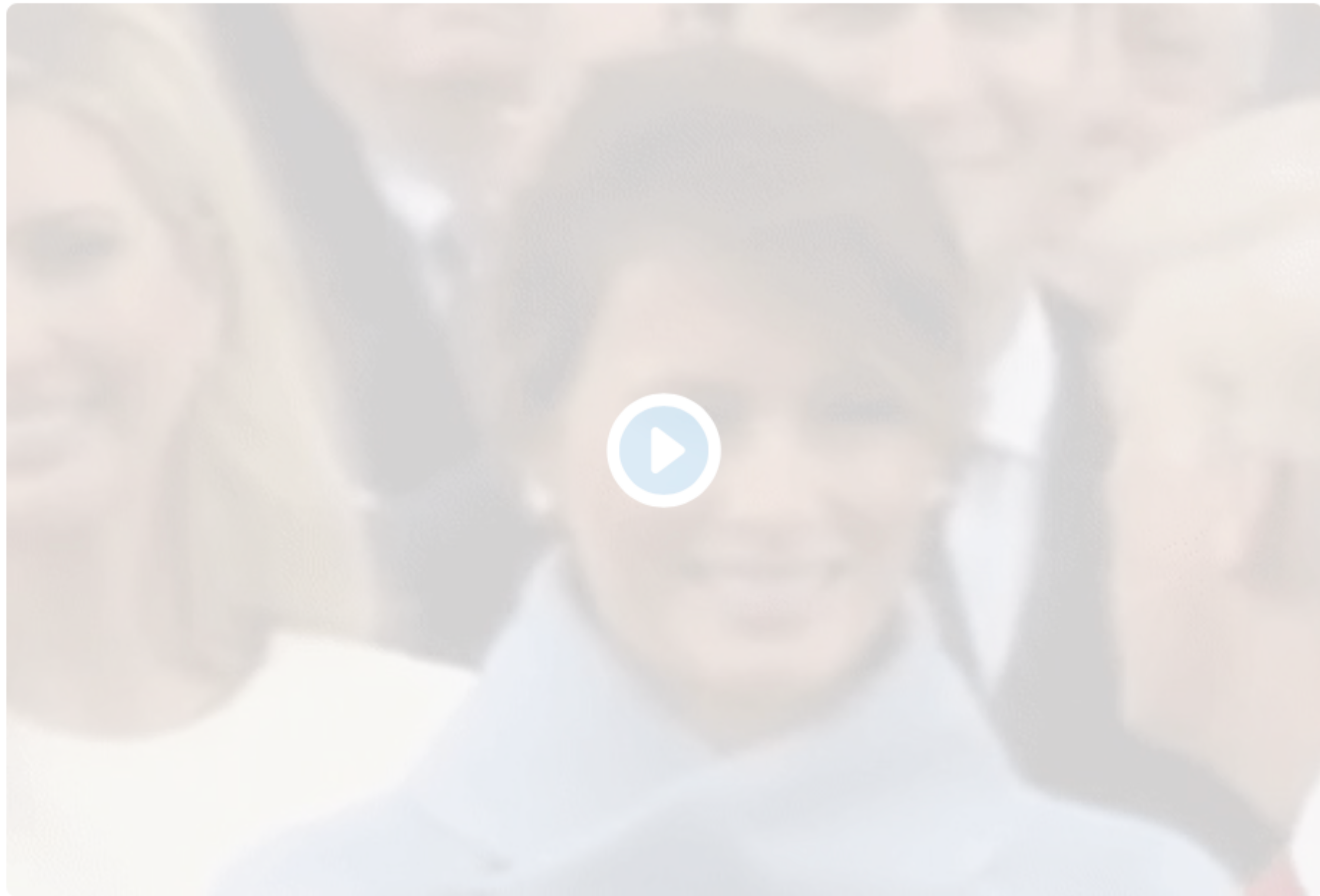


Andy Ostroy ✓

@AndyOstroy



Seems the only #Wall @realDonaldTrump's built is the one between him and @FLOTUS #Melania #trump



♥ 8,221 8:15 PM - May 2, 2017



💬 4,022 people are talking about this



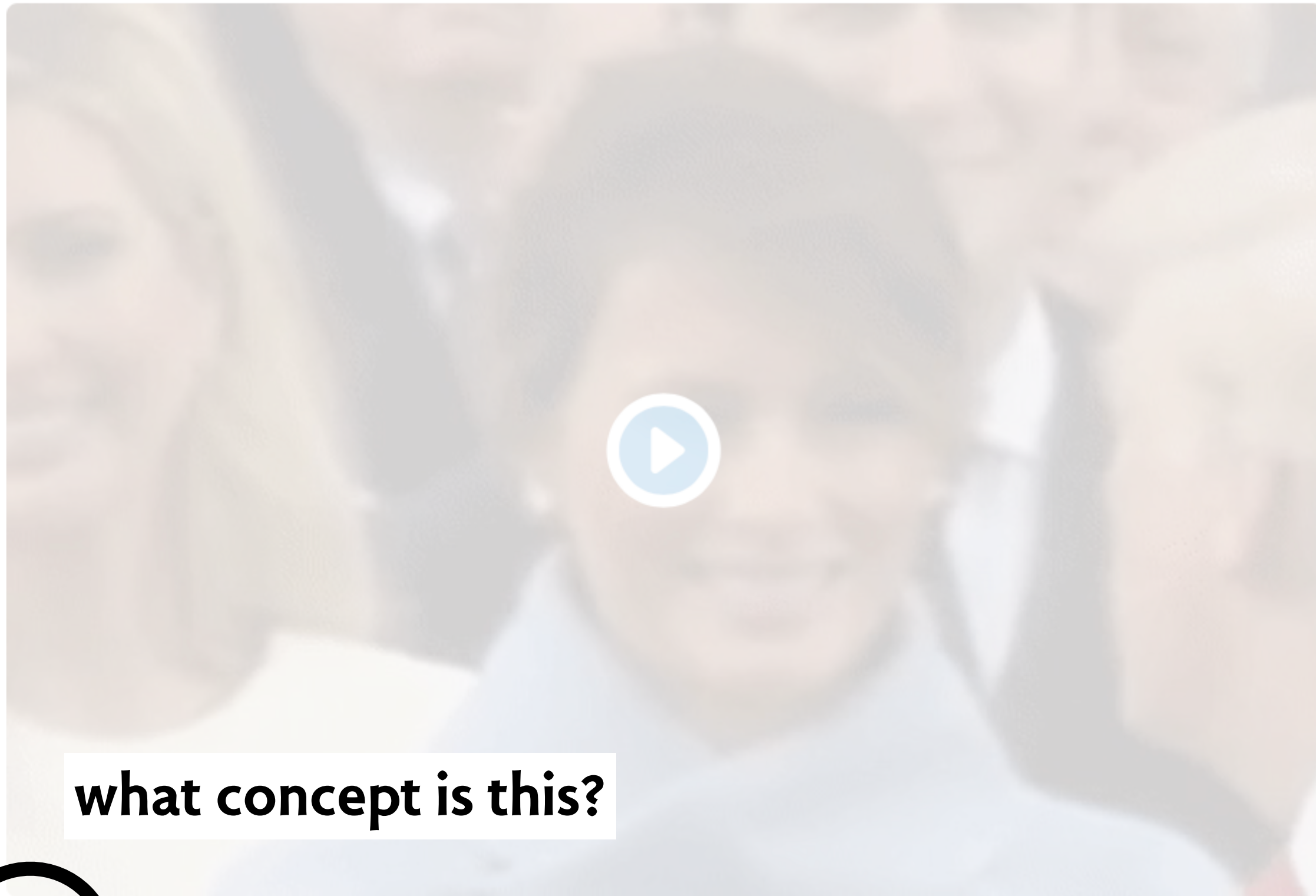


Andy Ostroy ✓

@AndyOstroy



Seems the only #Wall @realDonaldTrump's built is the one between him and @FLOTUS #Melania #trump



8,221 8:15 PM - May 2, 2017



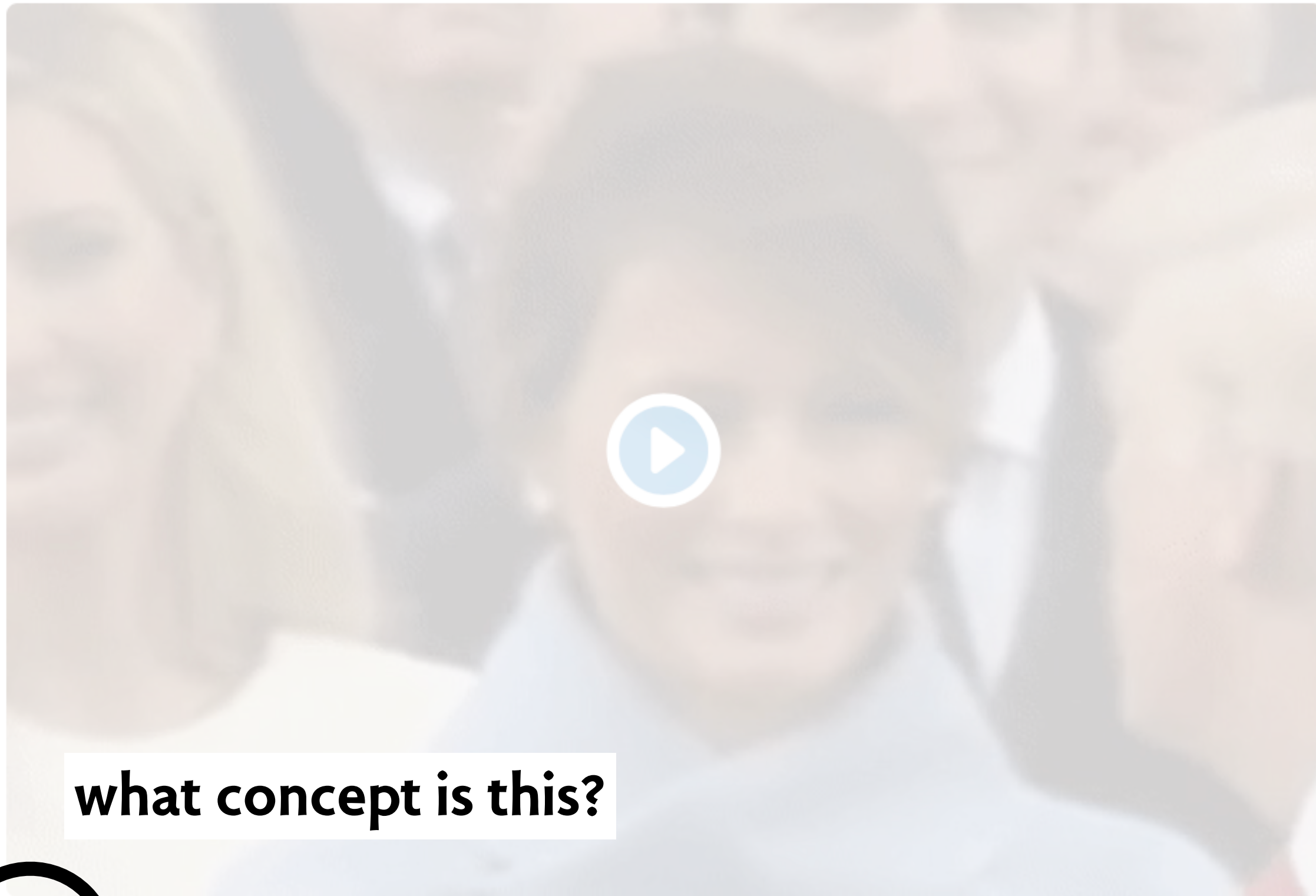
4,022 people are talking about this



 **Andy Ostroy** ✓
@AndyOstroy



Seems the only #Wall @realDonaldTrump's built is the one between him and @FLOTUS #Melania #trump



what concept is this?

8,221 8:15 PM - May 2, 2017



4,022 people are talking about this



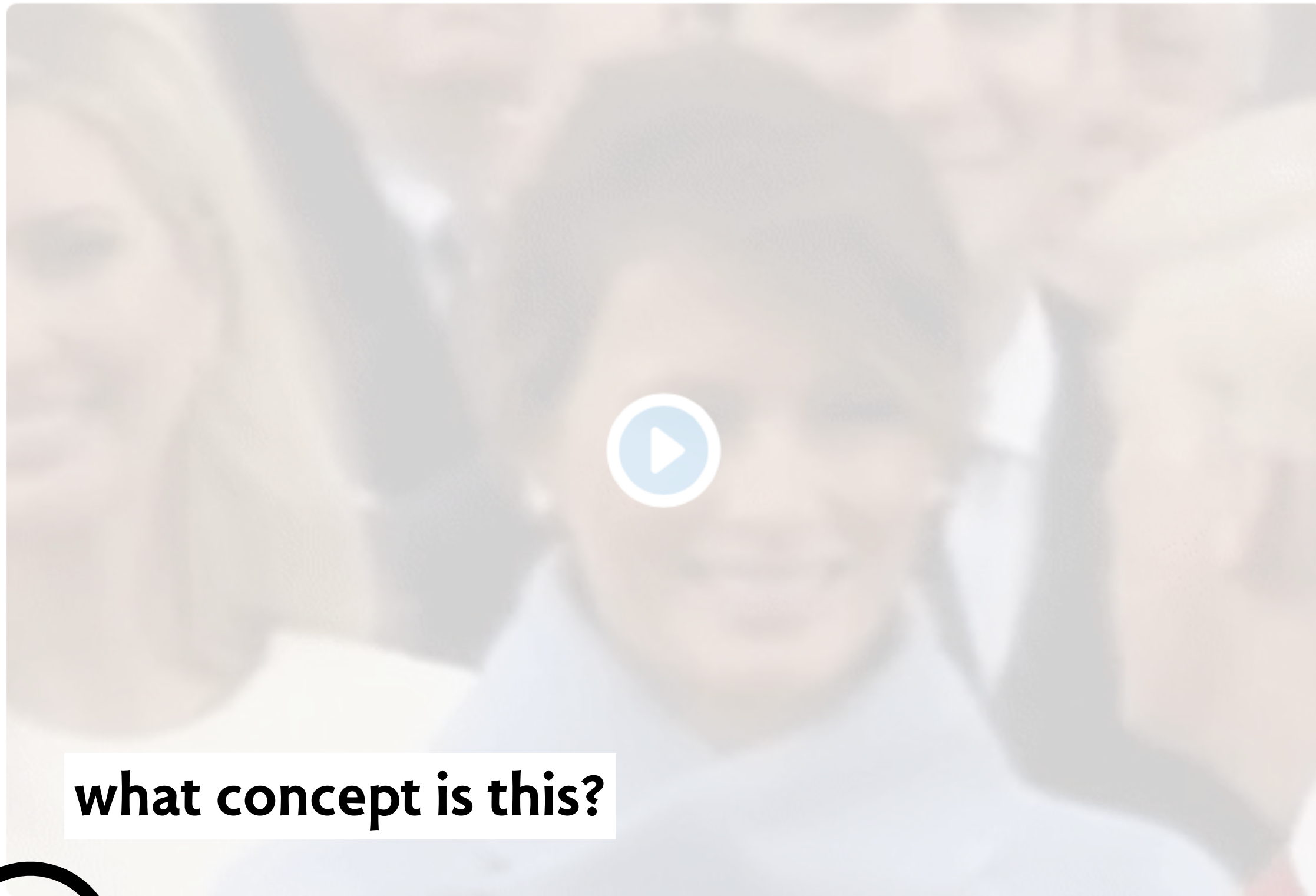
concept Upvote

purpose rank items by popularity

 **Andy Ostroy** 
@AndyOstroy



Seems the only #Wall @realDonaldTrump's built is the one between him and @FLOTUS #Melania #trump



 8,221 8:15 PM - May 2, 2017 

 4,022 people are talking about this 



concept Upvote
purpose rank items by popularity

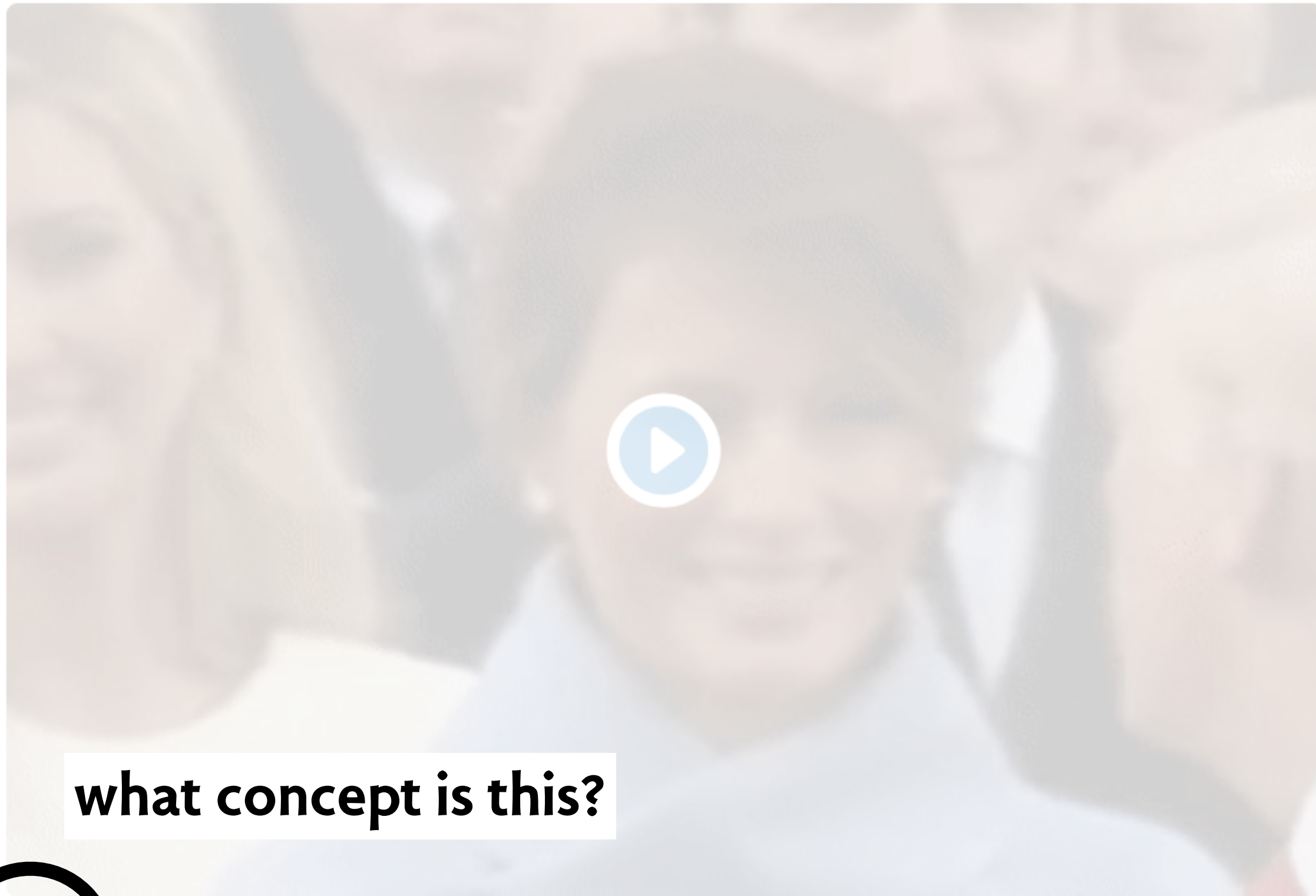


concept Bookmark
purpose save items to revisit

 **Andy Ostroy** 
@AndyOstroy



Seems the only #Wall @realDonaldTrump's built is the one between him and @FLOTUS #Melania #trump



 8,221 8:15 PM - May 2, 2017 

 4,022 people are talking about this 



concept Upvote
purpose rank items by popularity



concept Bookmark
purpose save items to revisit

missing
concept!

twitter adds a new concept (2018)




The Boston Globe  @BostonGlobe · 21h

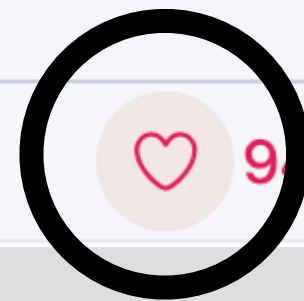
Andrew Yang would fine gunmakers for deaths caused by their products.



Yang would fine gunmakers for deaths caused by their products - Th...
You probably know Andrew Yang wants to give every American \$1,000 a month. Something you might not know: He wants to fine gun ...
[bostonglobe.com](https://www.bostonglobe.com)

 29

 9



twitter adds a new concept (2018)



concept Upvote

purpose rank items by popularity

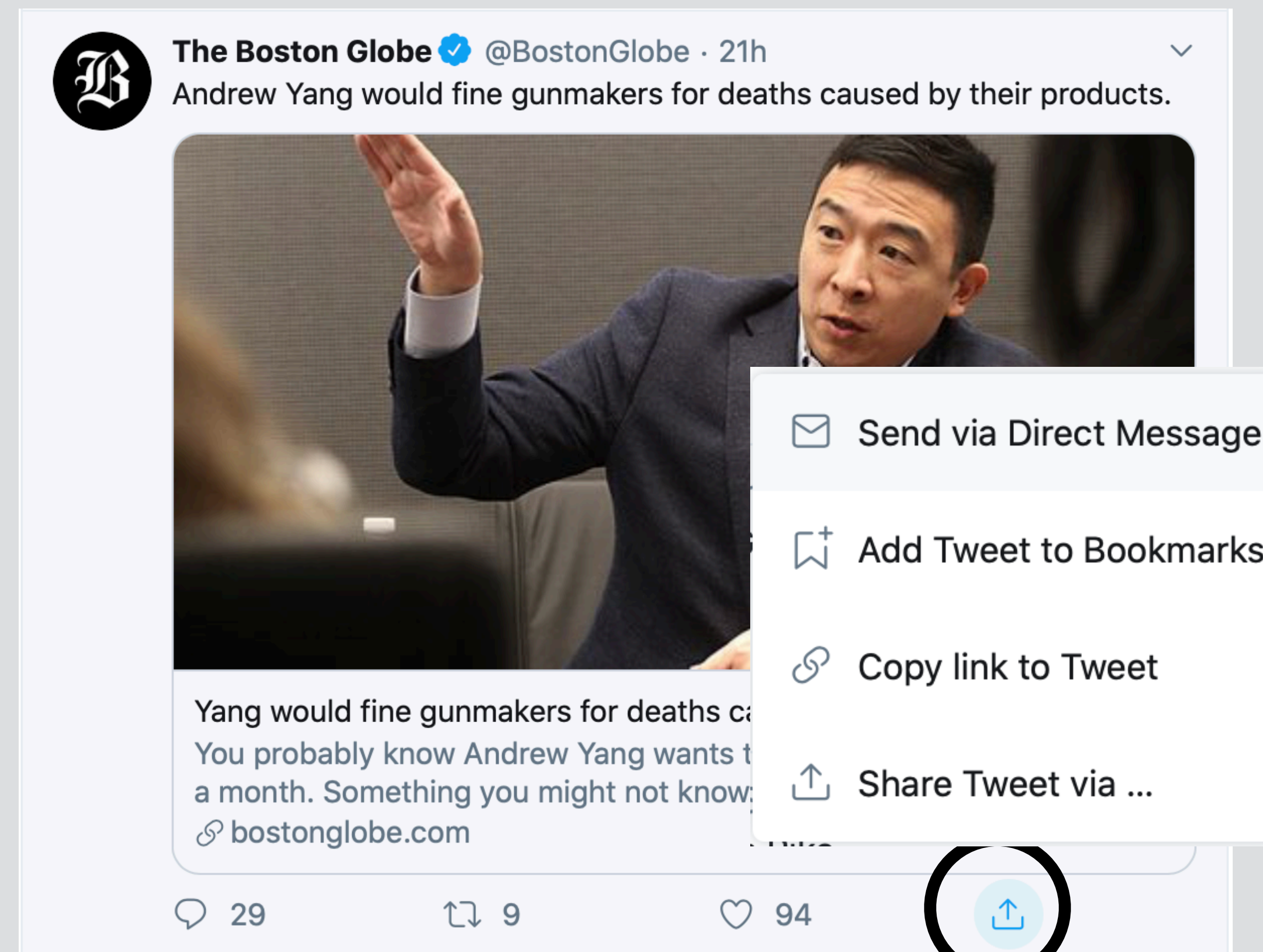
twitter adds a new concept (2018)



concept Upvote

purpose rank items by popularity

twitter adds a new concept (2018)



concept Upvote

purpose rank items by popularity

twitter adds a new concept (2018)



concept Upvote

purpose rank items by popularity



concept Bookmark

purpose save items to revisit

a new path
inspired by old ideas

user's
model

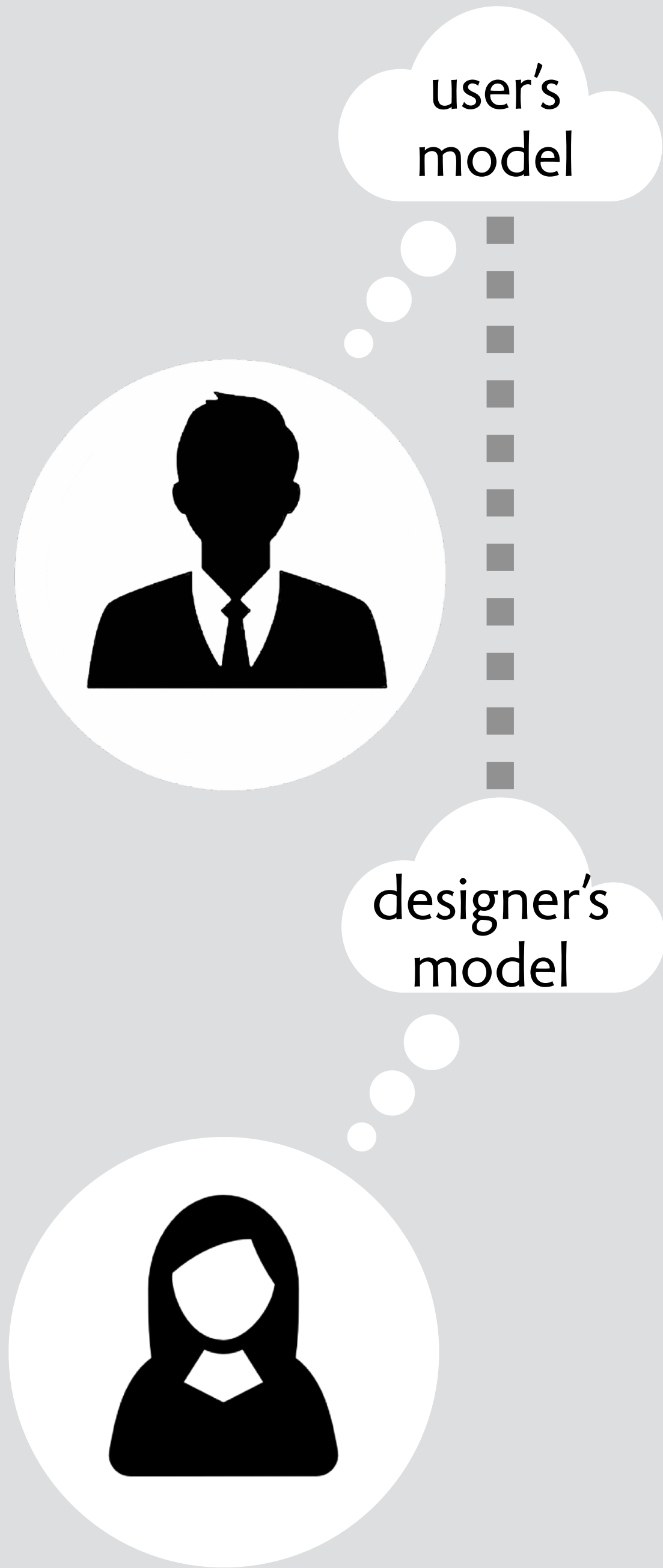


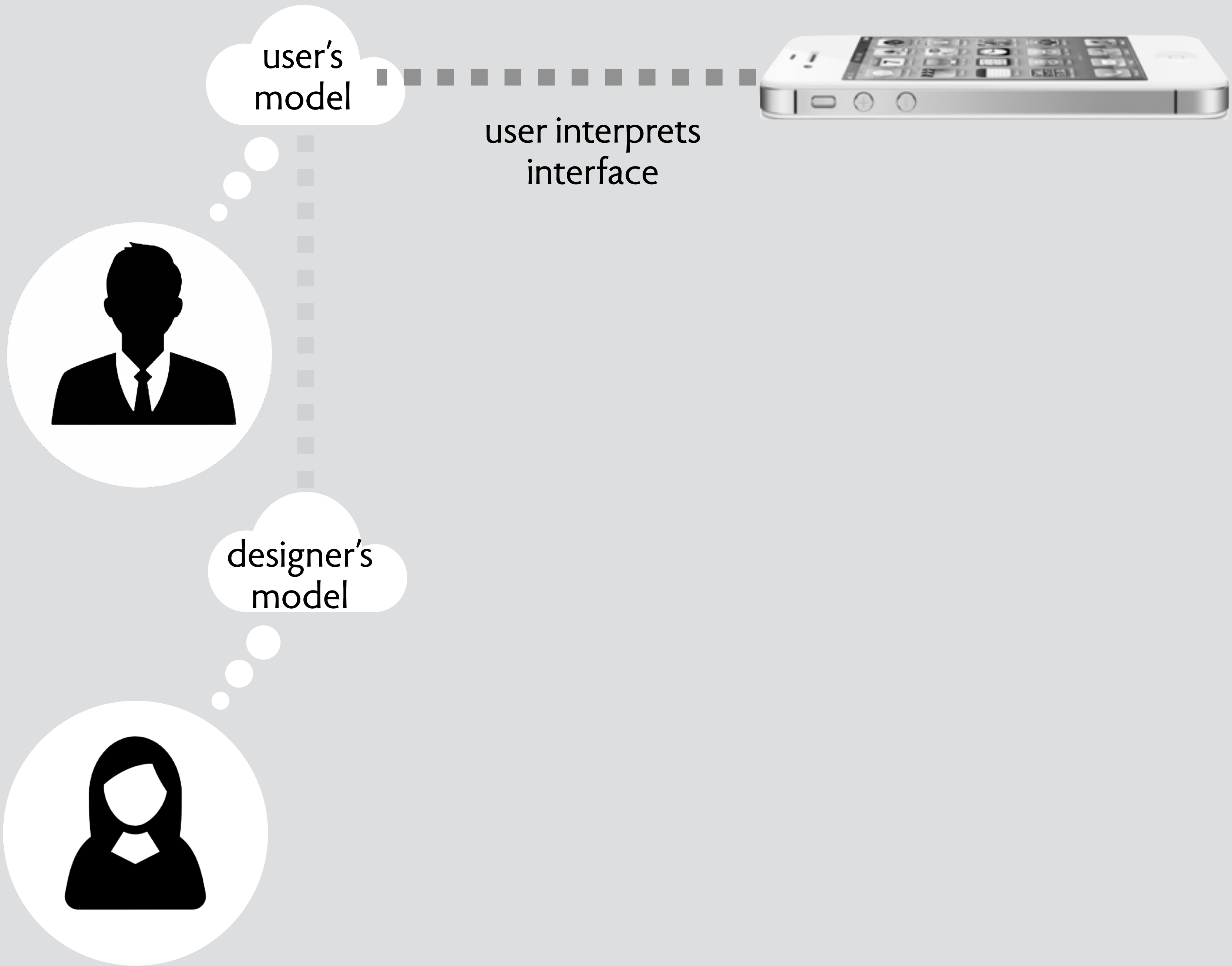
user's
model

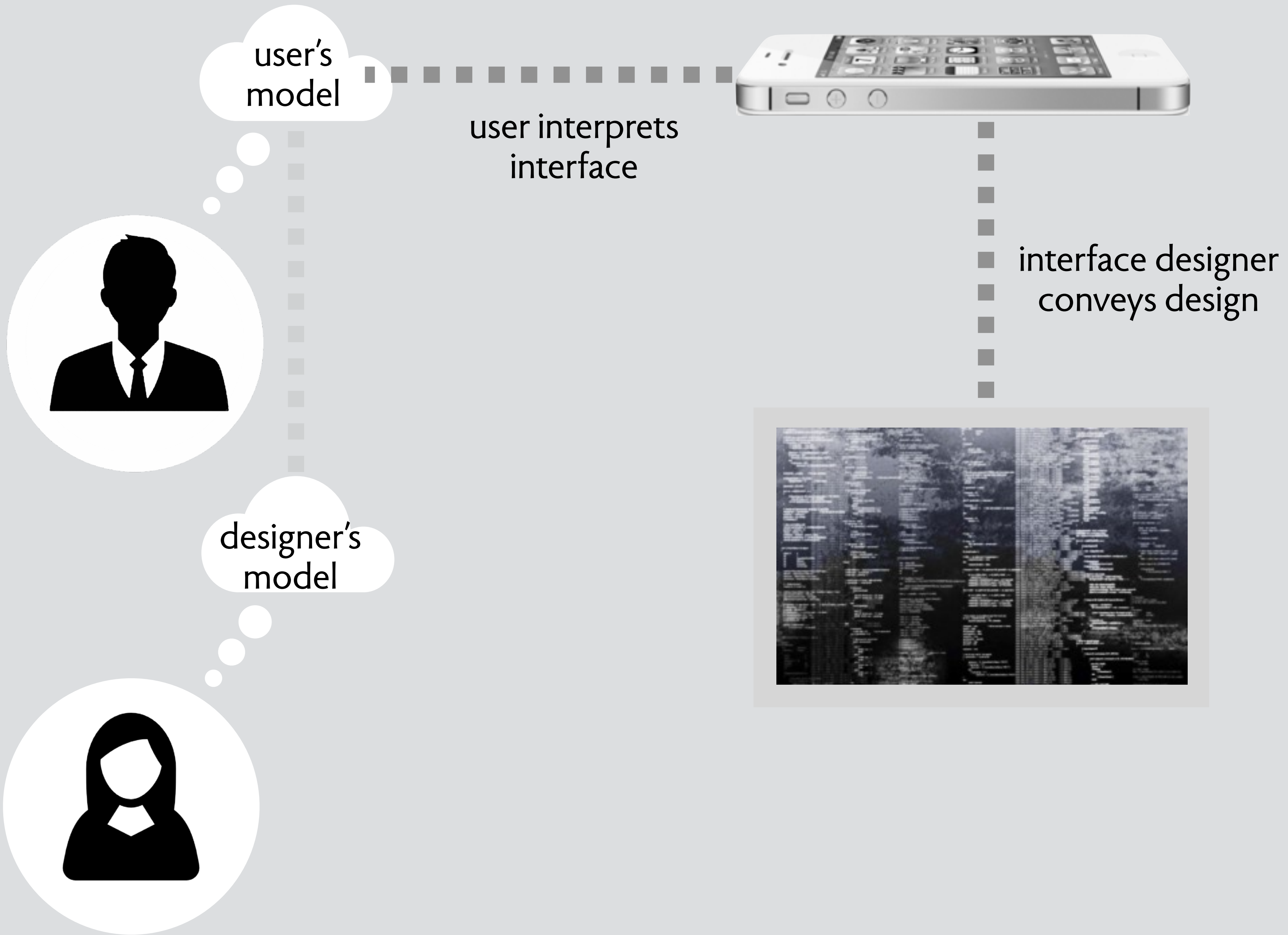


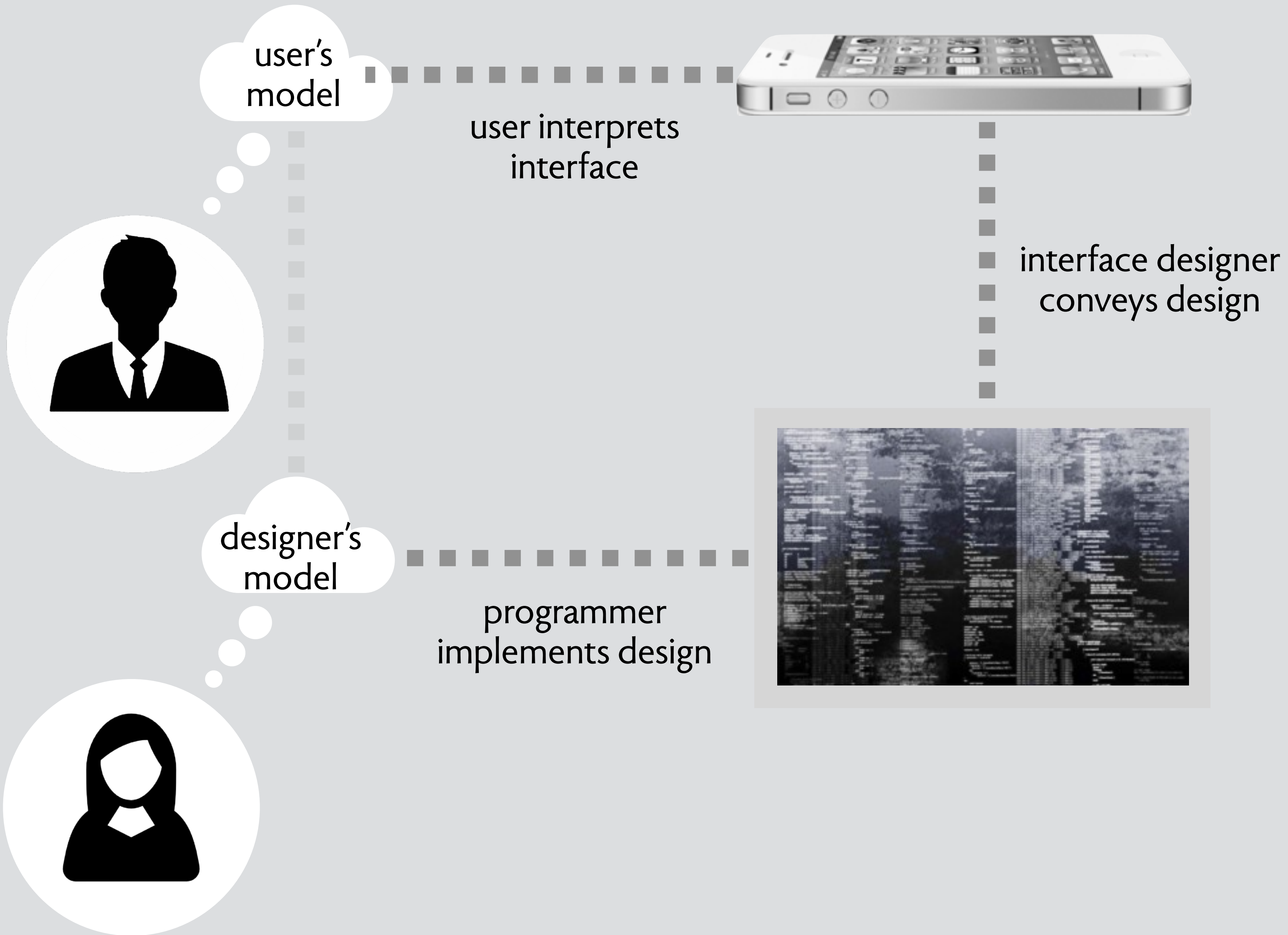
designer's
model

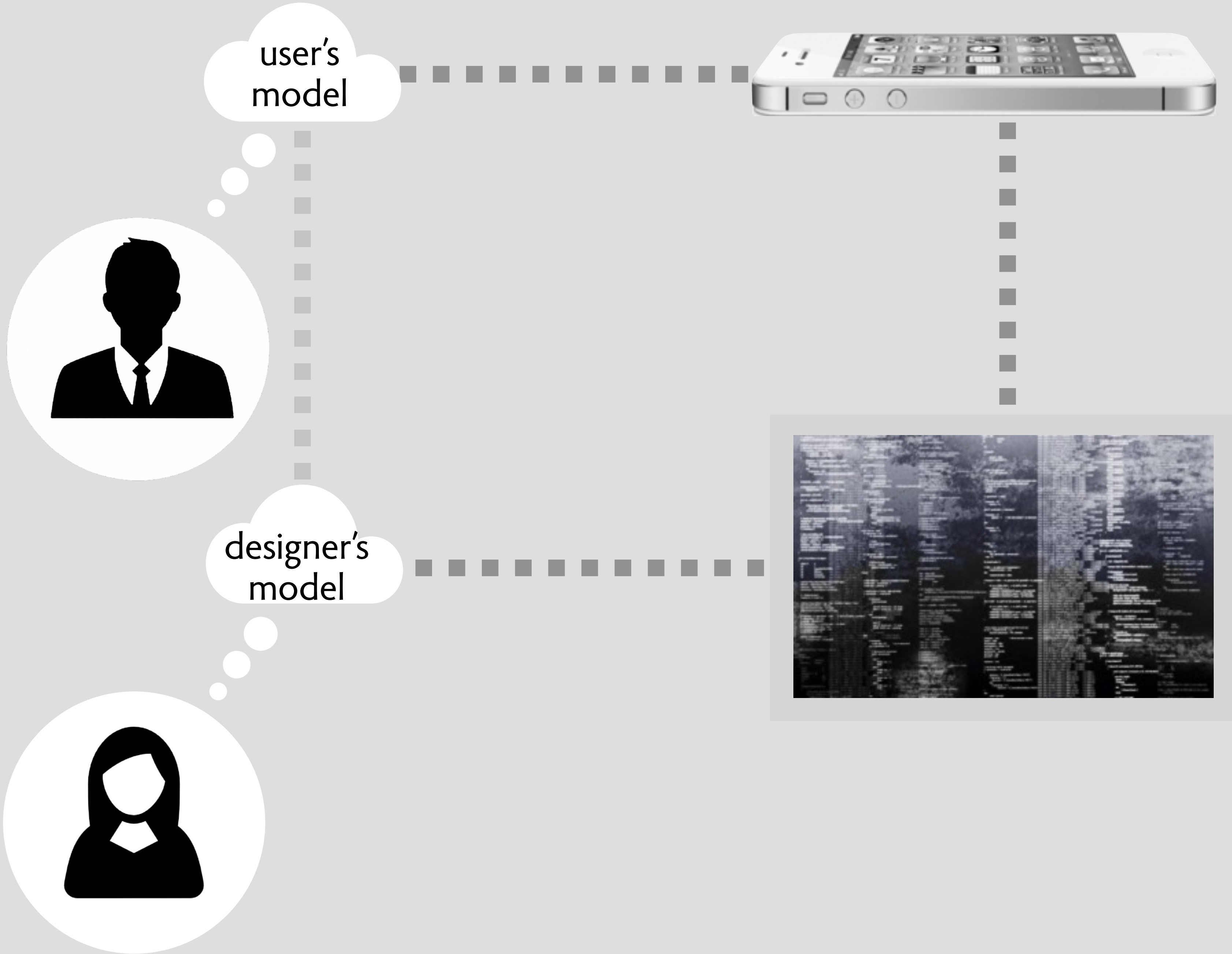


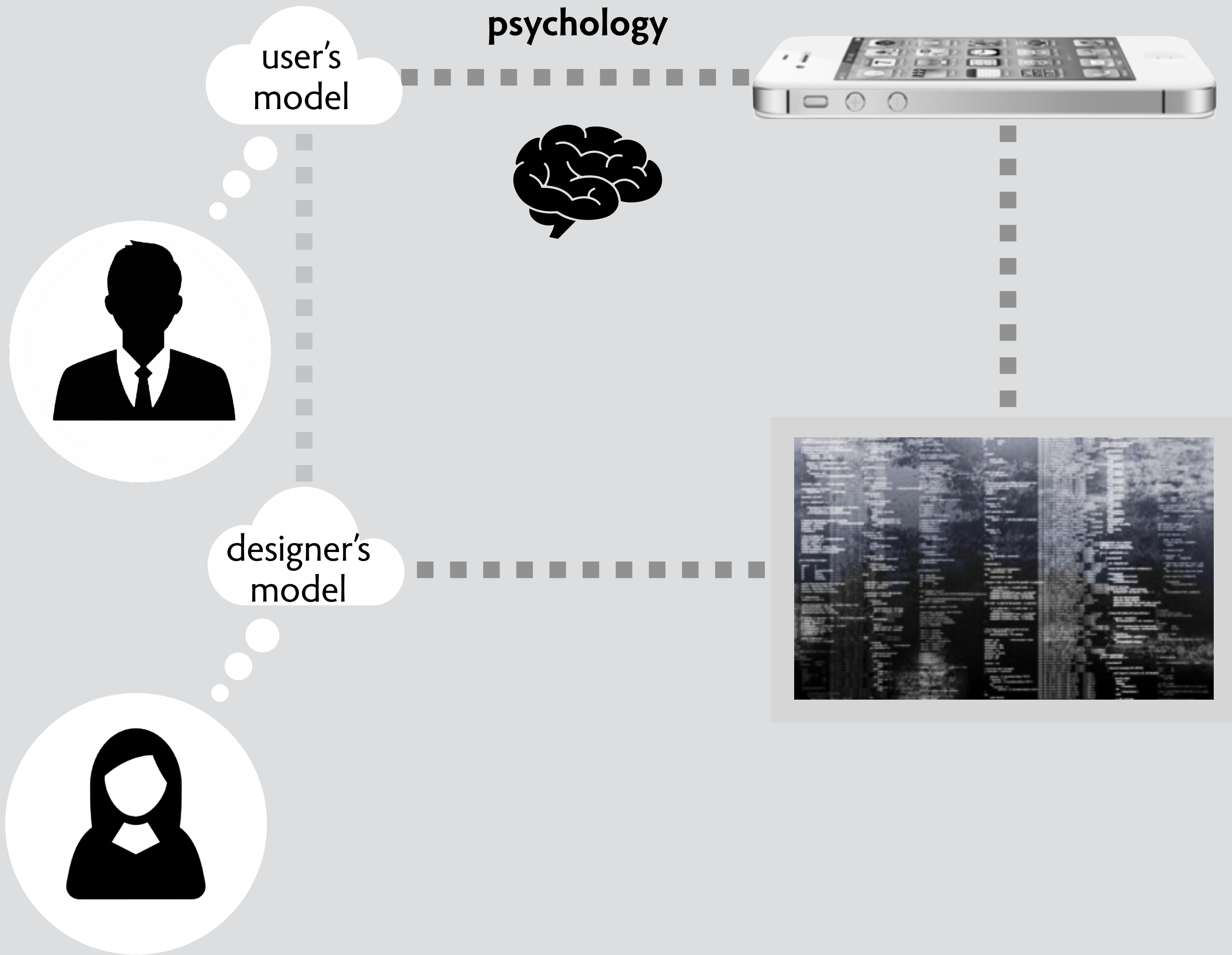


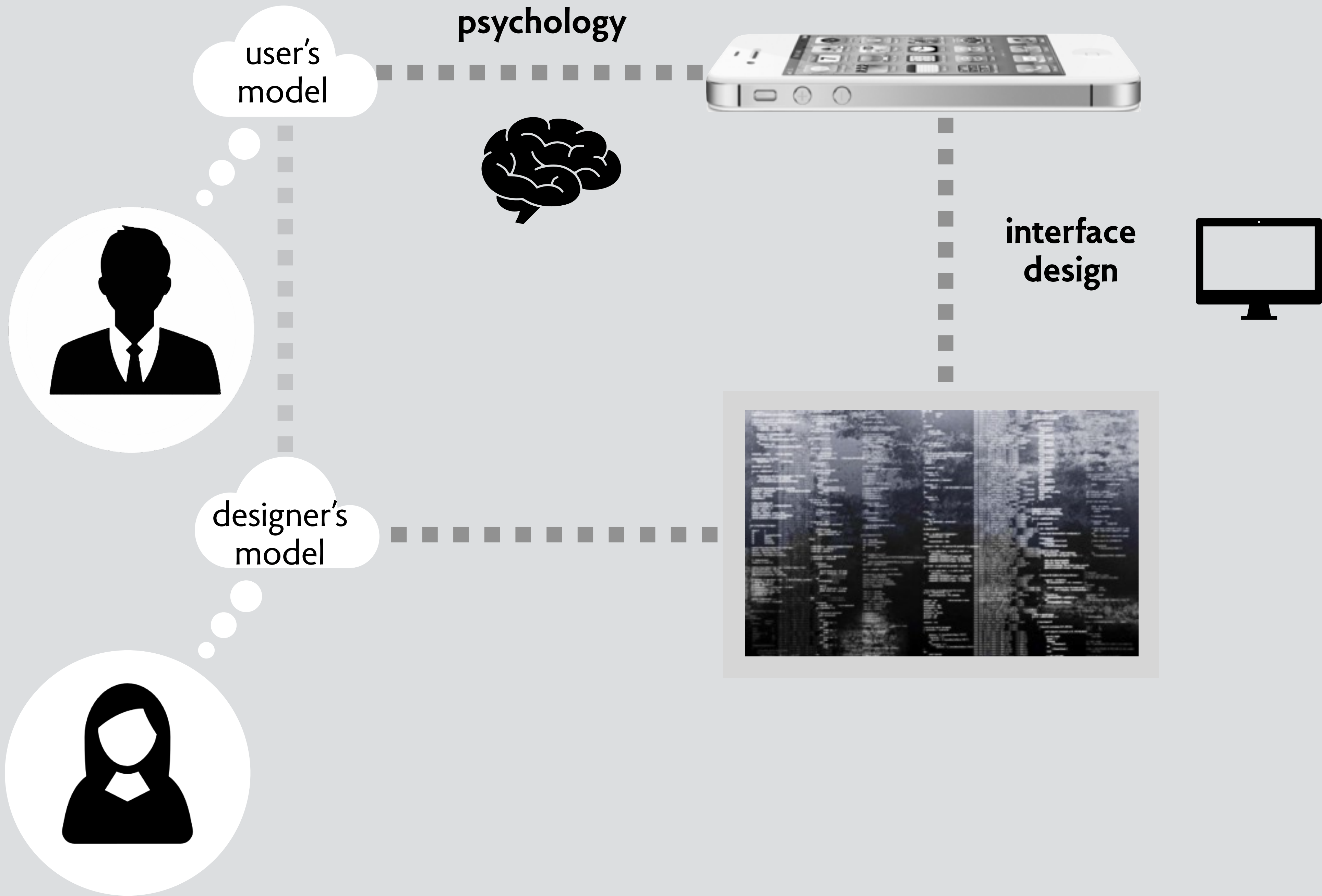










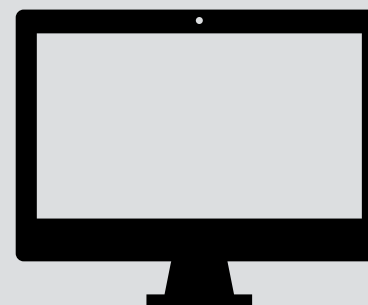


user's model

psychology

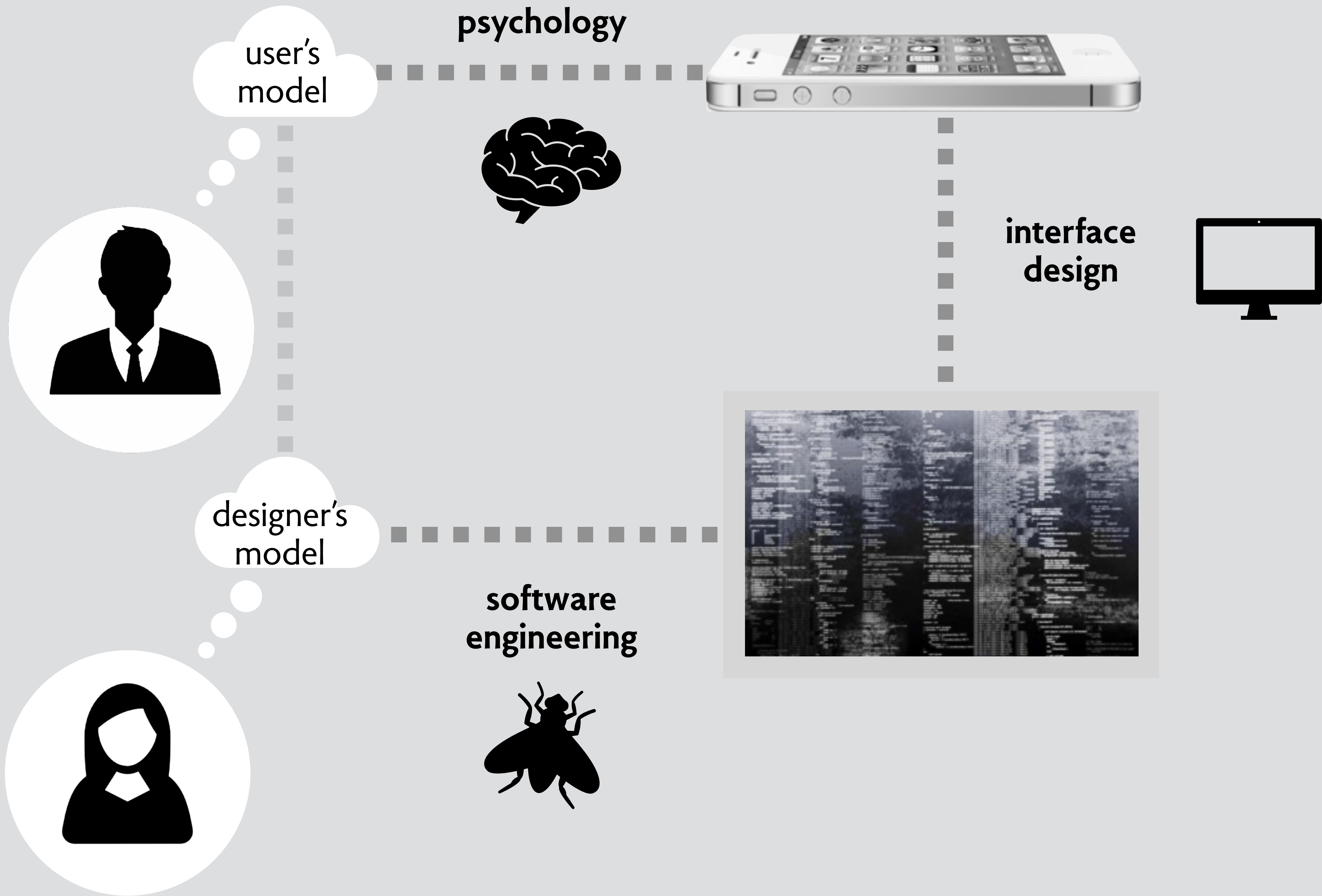


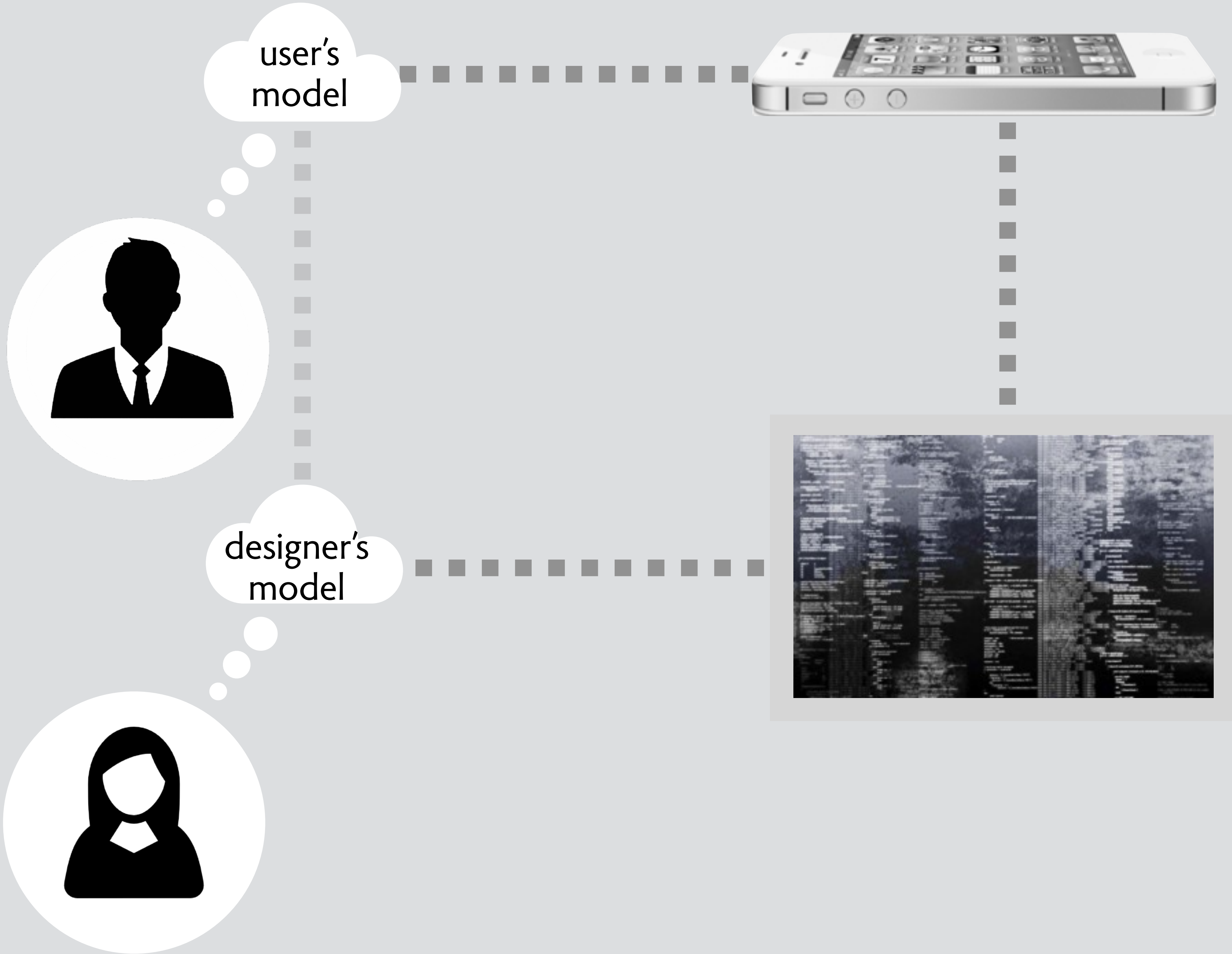
interface design

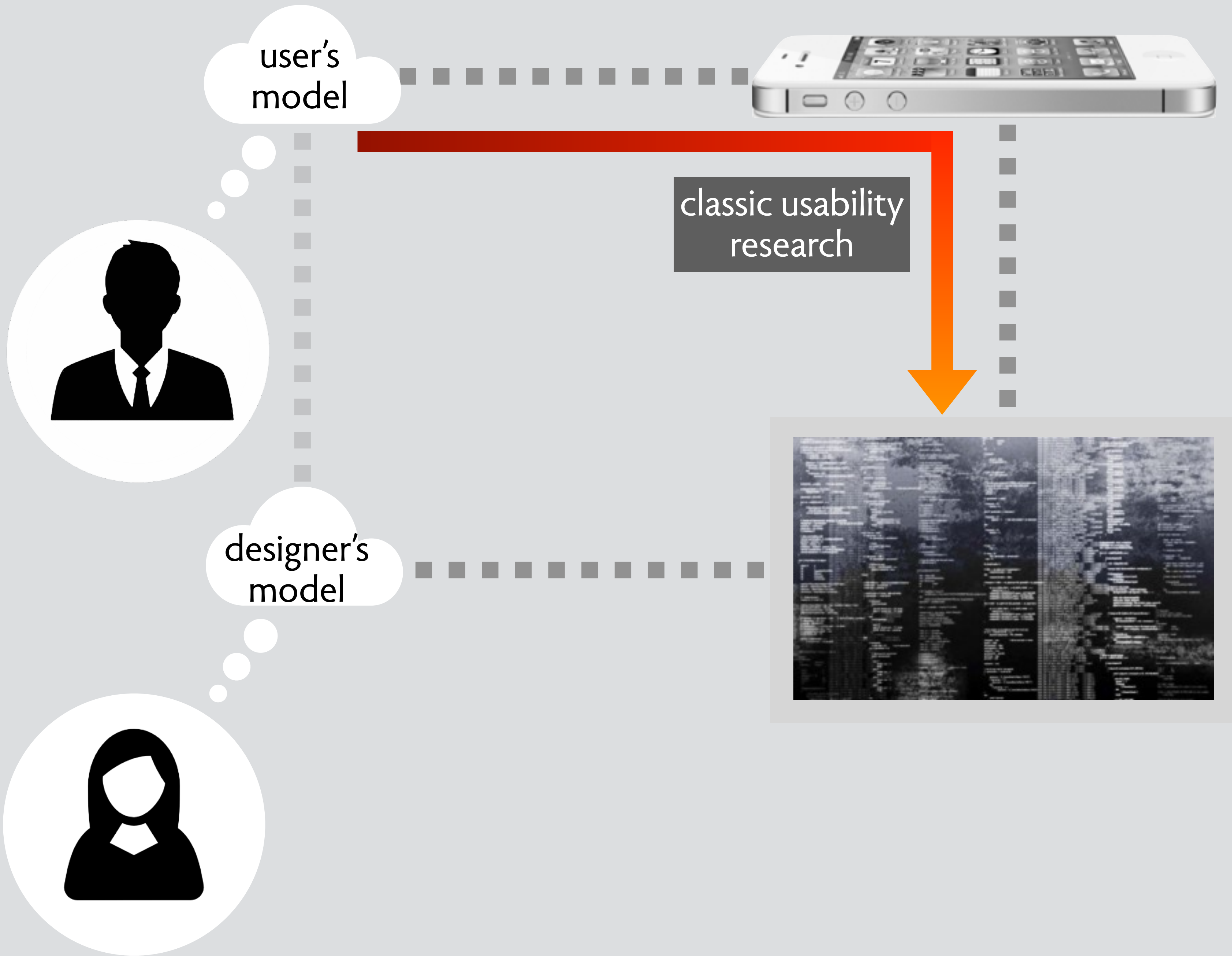


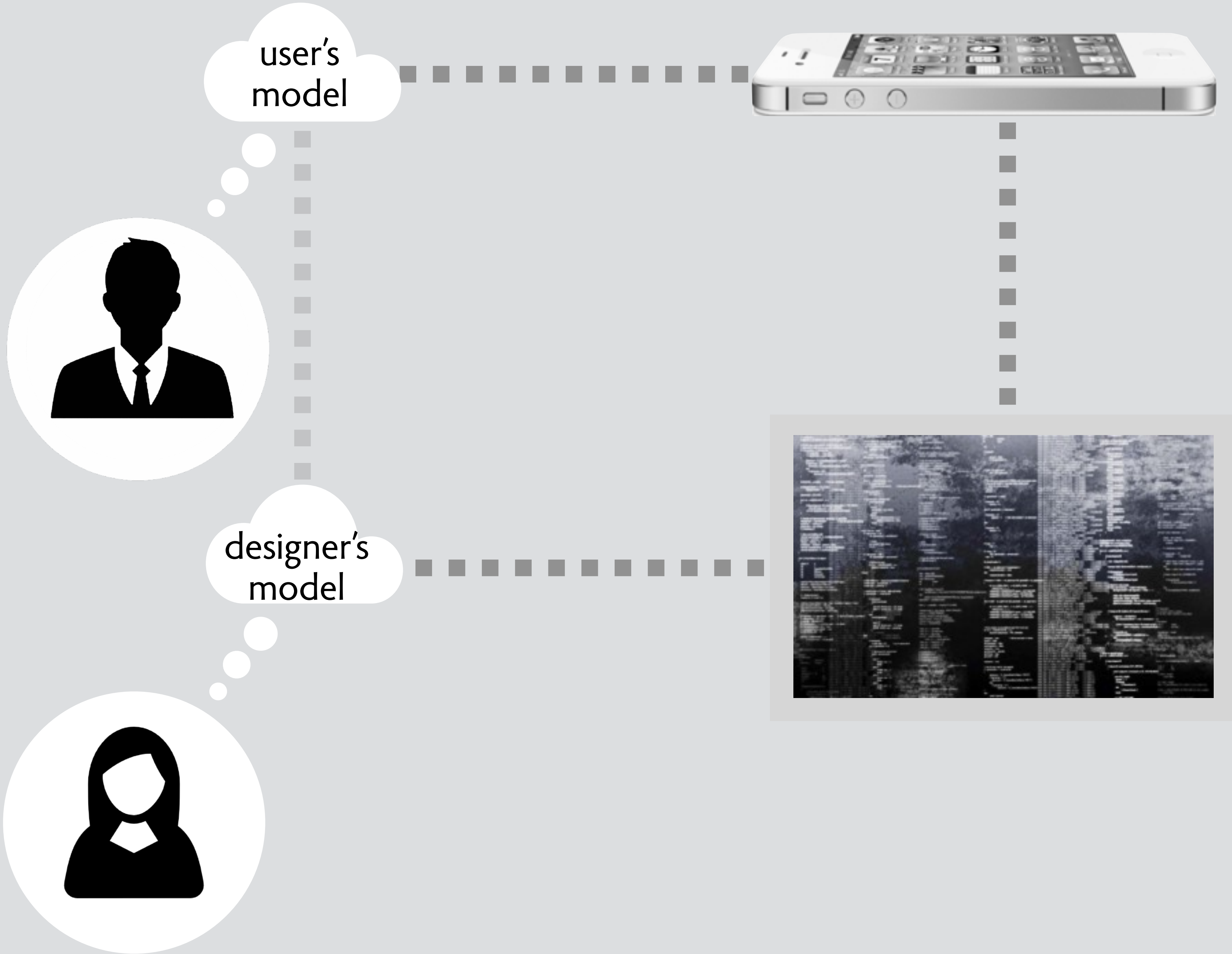
designer's model

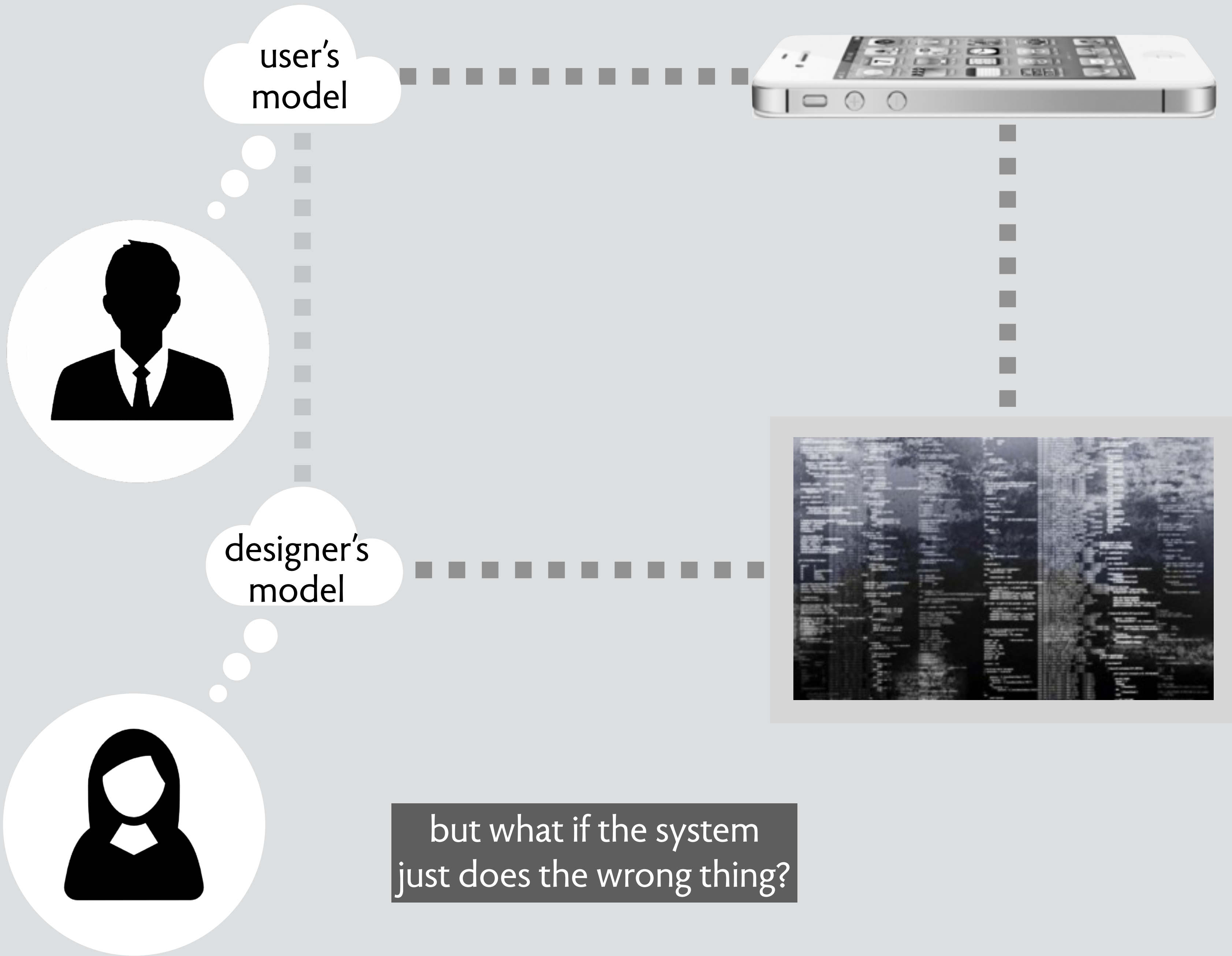












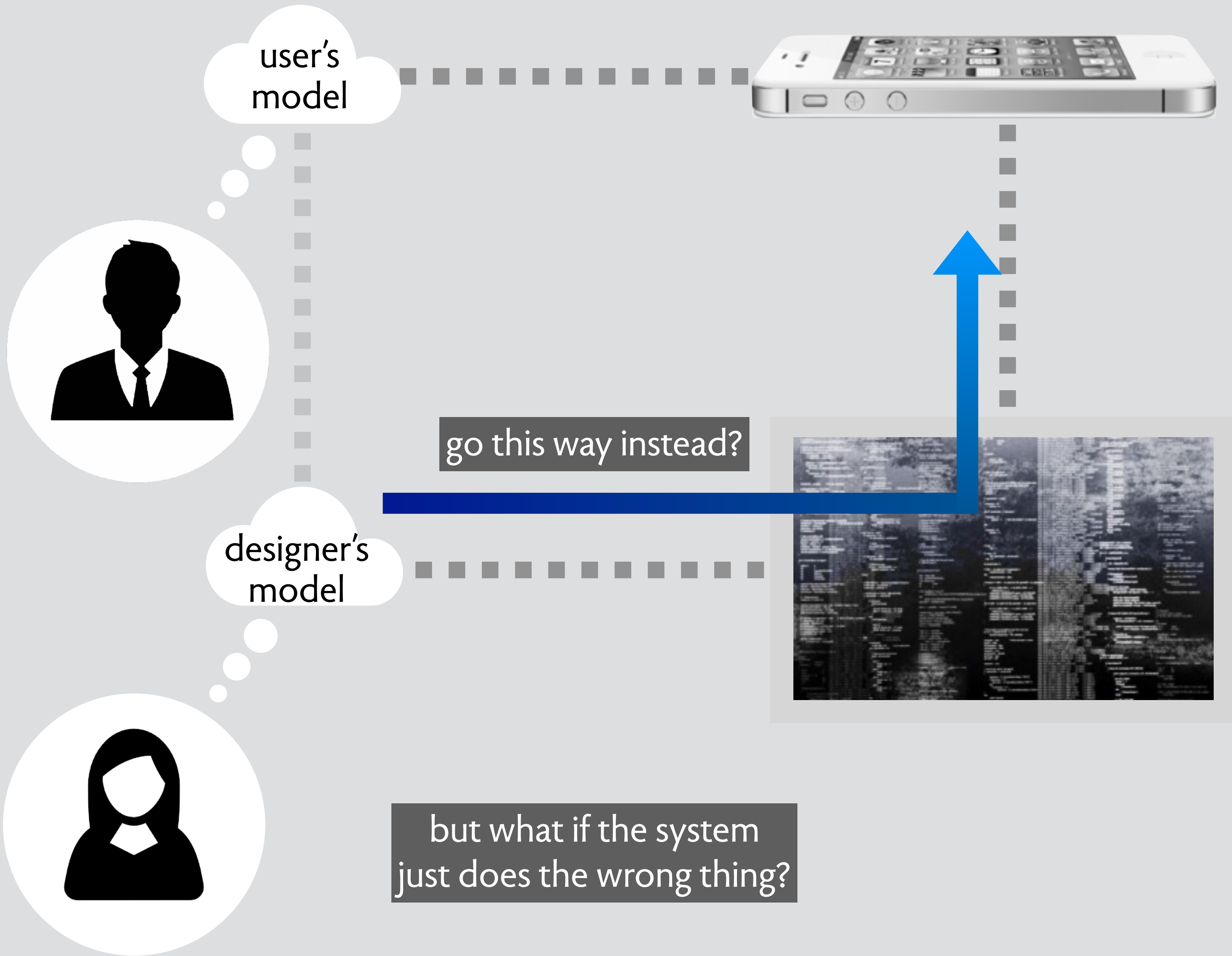
user's
model



designer's
model



but what if the system
just does the wrong thing?



REVISED & EXPANDED EDITION

The DESIGN *of* EVERYDAY THINGS

DON
NORMAN



focus on faithful projection of design model
mapping, affordance/signifiers, gulfs

theory turned into practice through heuristics
8 Golden Rules (Shneiderman)
First Principles of Interaction Design (Tognazzini)
10 Usability Heuristics (Nielsen)

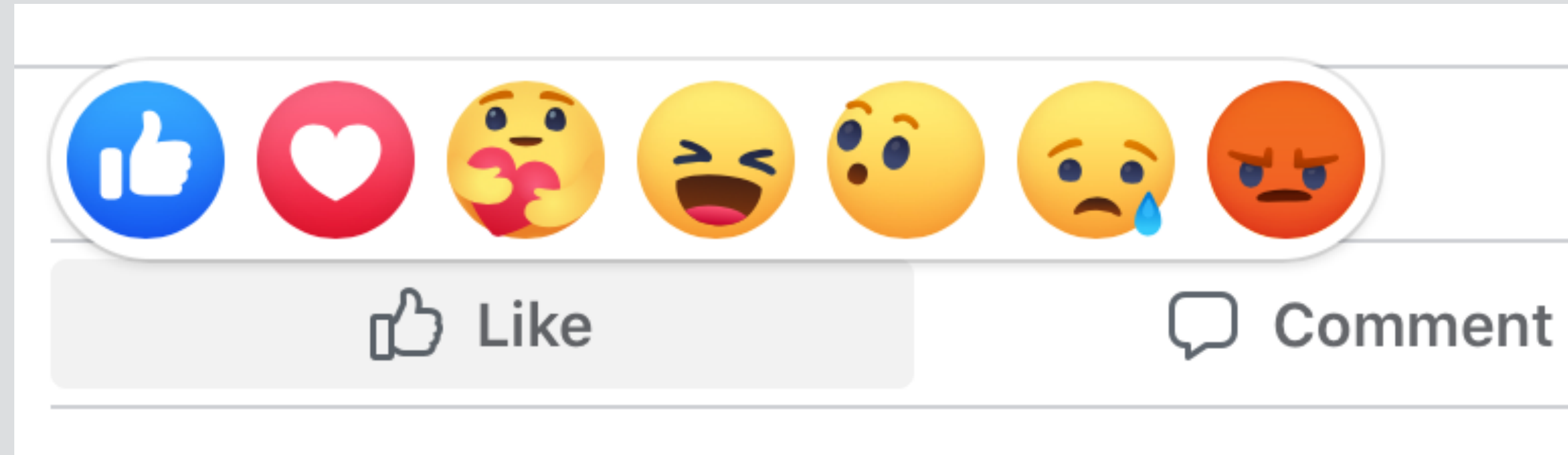
result: huge improvements in usability
anyone can design a great UI

If a simple model is not explicitly or implicitly provided, users formulate their own myths about how the system works... The system has to be designed with an **explicit conceptual model** that is easy enough for the user to learn.

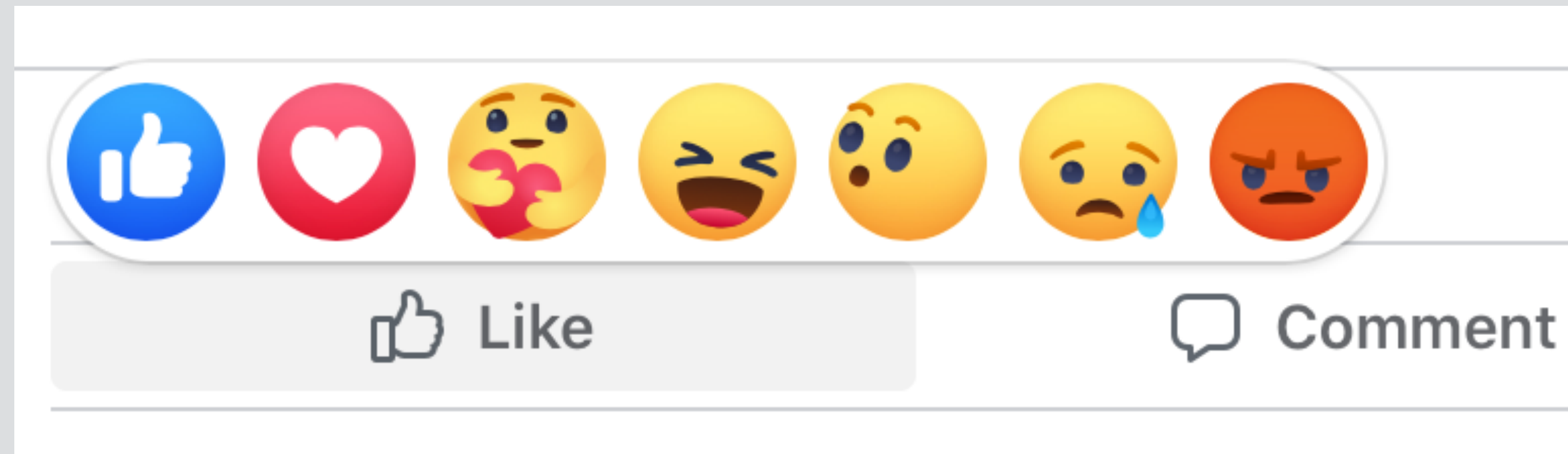
Stuart Card & Thomas Moran (1986)



should facebook concepts be desynchronized too?



should facebook concepts be desynchronized too?



concept Upvote

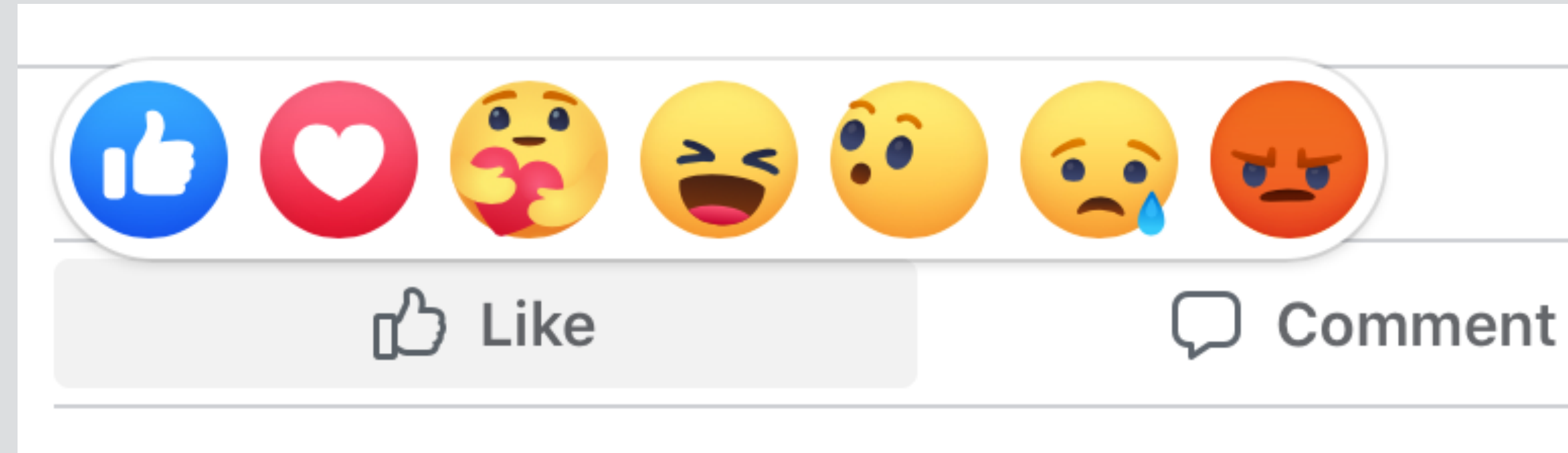
purpose rank items by popularity

actions

upvote (u: User, i: Item)

...

should facebook concepts be desynchronized too?

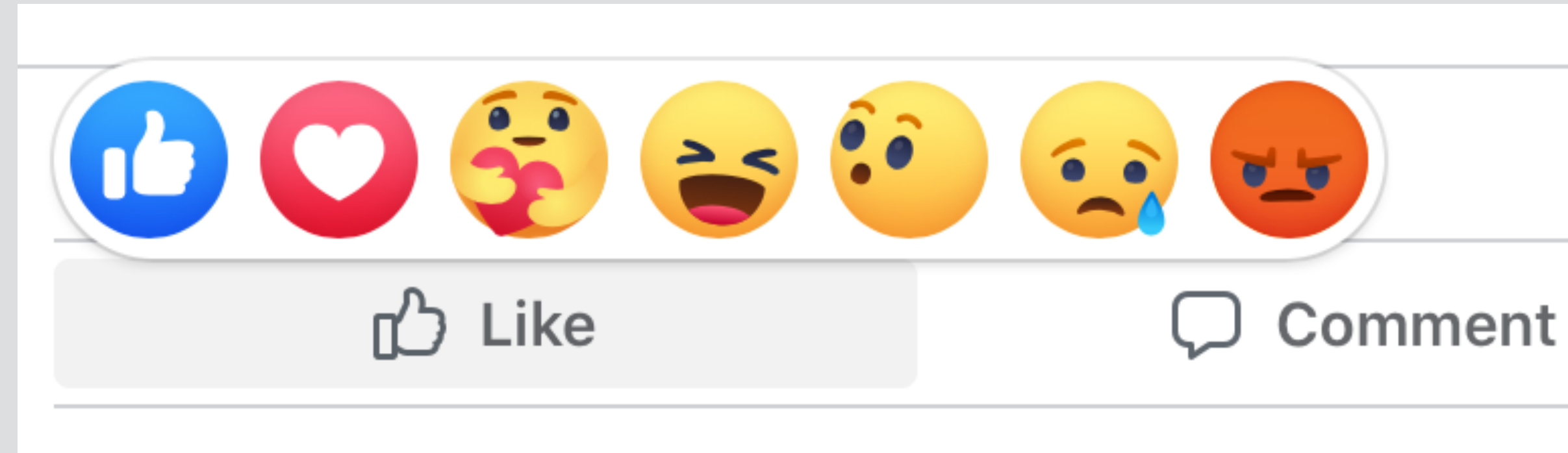


concept Upvote
purpose rank items by popularity
actions
upvote (u: User, i: Item)
...



concept Reaction
purpose convey emotion to author
actions
reactAngry (u: User, i: Item)
...

should facebook concepts be desynchronized too?

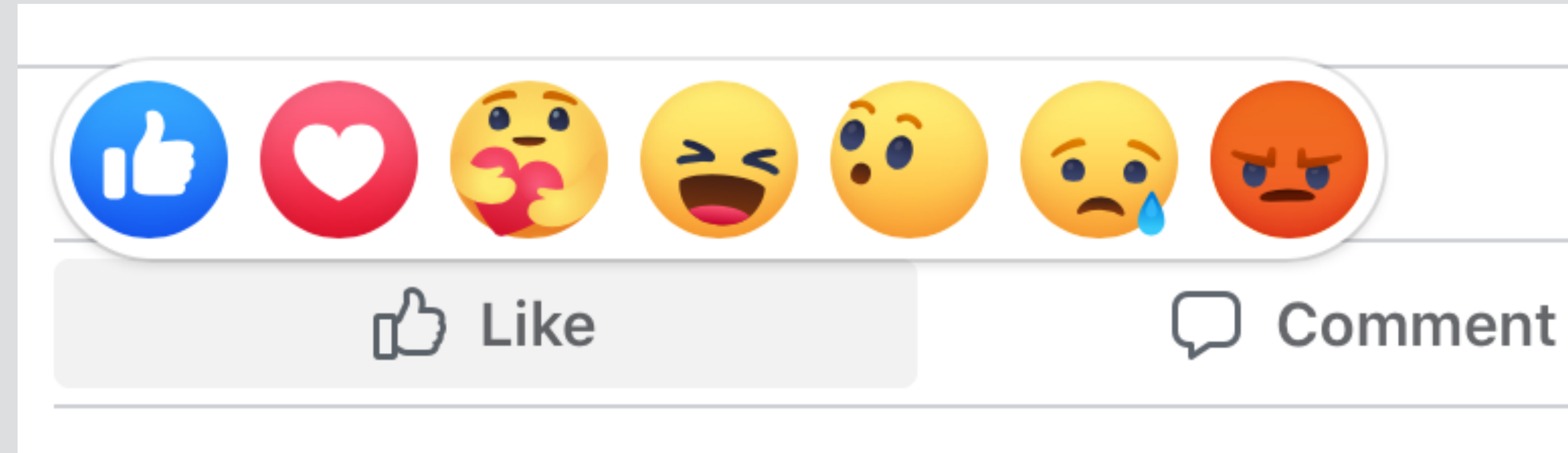


concept Upvote
purpose rank items by popularity
actions
upvote (u: User, i: Item)
...



concept Reaction
purpose convey emotion to author
actions
reactAngry (u: User, i: Item)
...

should facebook concepts be desynchronized too?



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?