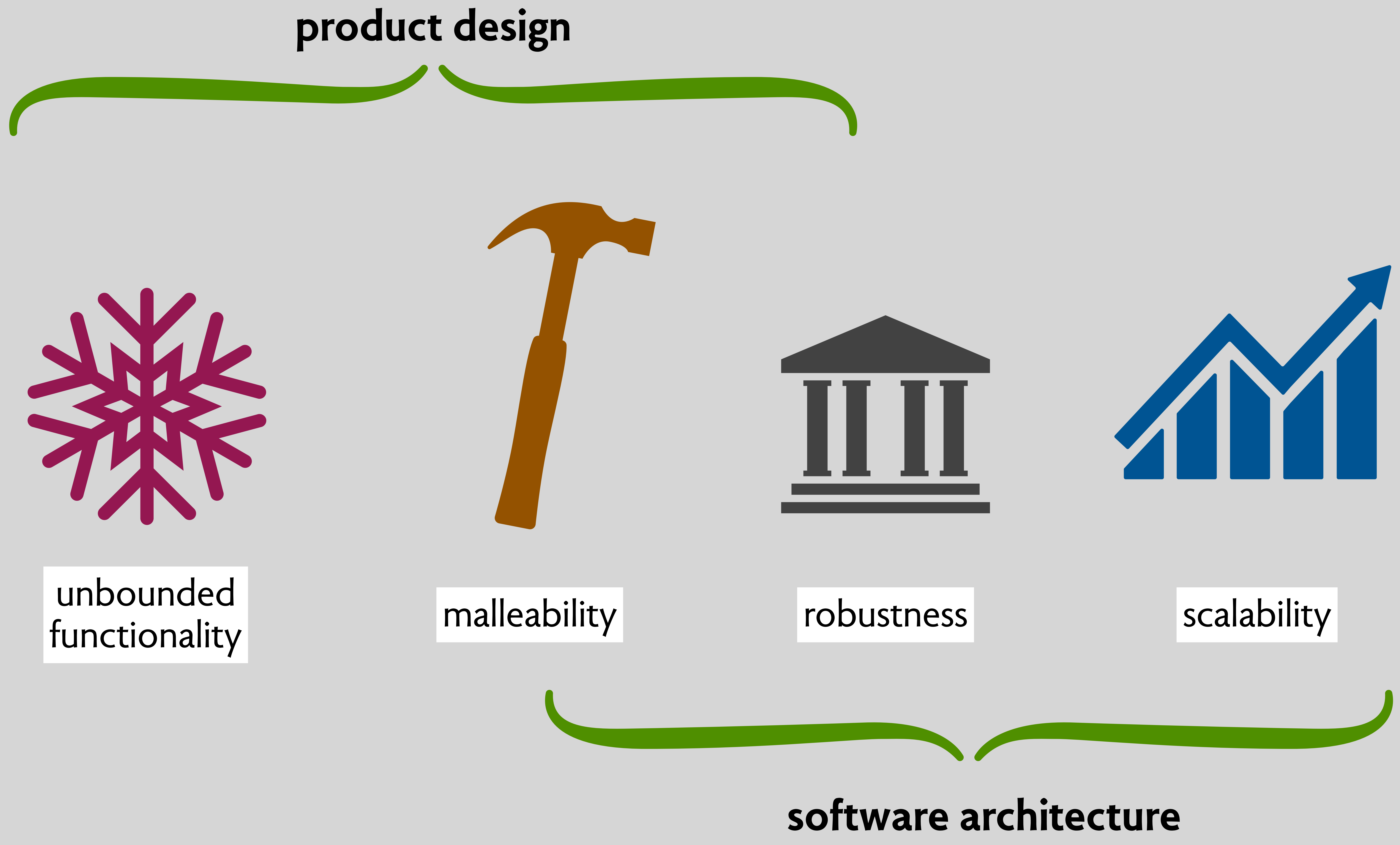
why software is hard

Daniel Jackson · Autodesk · Woodinville, WA · Dec 3-5, 2024

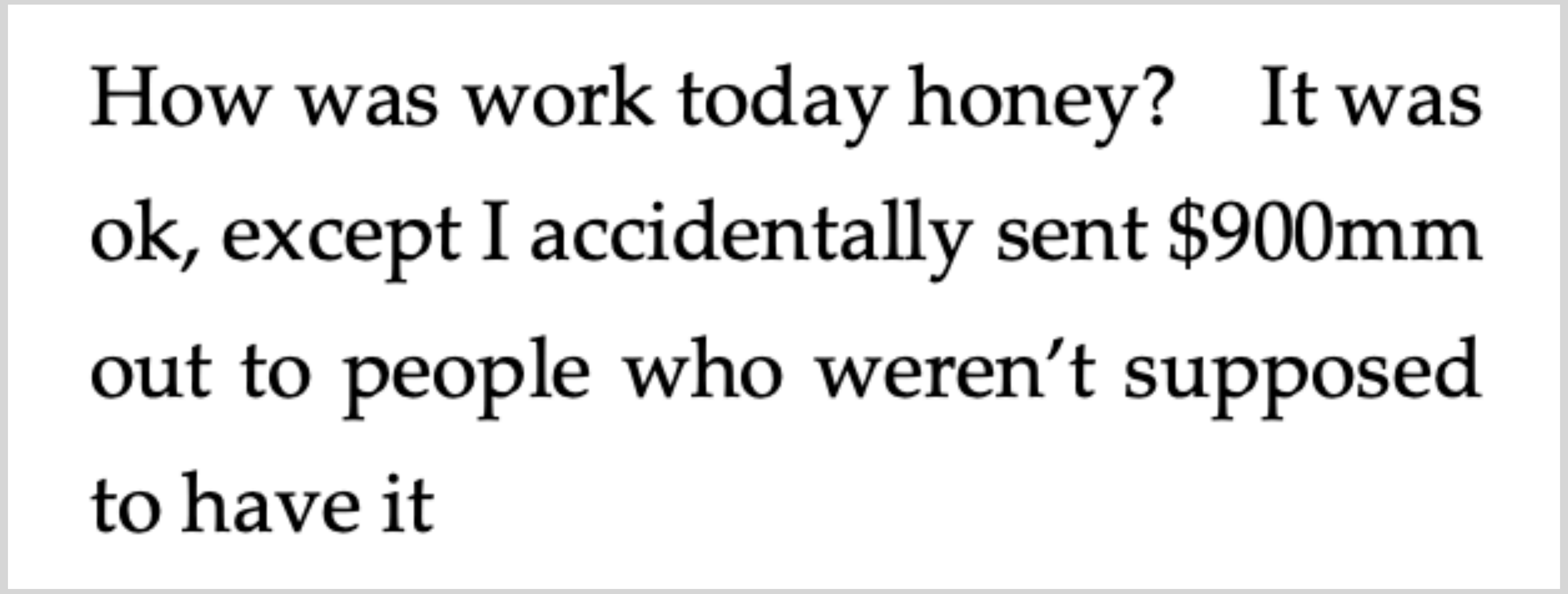
qualities of software: why software is so great



when software goes wrong

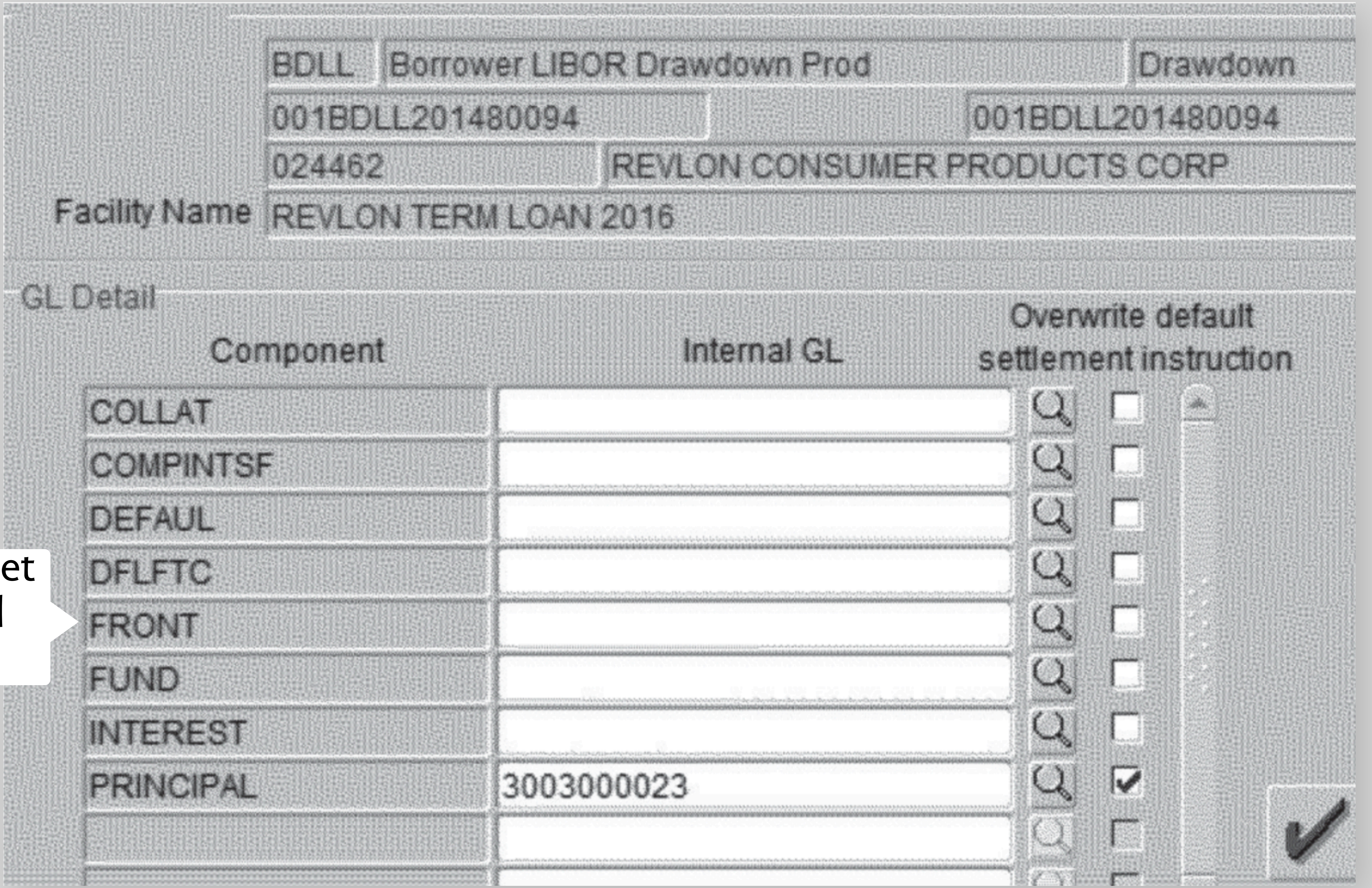
citibank flexcube
august 2020

an email exchange cited in a court docket



How was work today honey? It was ok, except I accidentally sent $900mm out to people who weren't supposed to have it

US Court of Appeals for the Second Circuit
Docket No 21-487, 2021



BDLL	Borrower LIBOR Drawdown Prod	Drawdown
001BDLL201480094		001BDLL201480094
024462	REVLON CONSUMER PRODUCTS CORP	
Facility Name	REVLON TERM LOAN 2016	

GL Detail

Component	Internal GL	Overwrite default settlement instruction
COLLAT		☐
COMPINTSF		☐
DEFAUL		☐
DFLFTC		☐
FRONT		☐
FUND		☐
INTEREST		☐
PRINCIPAL	3003000023	✓
		☐

should have set FRONT and FUND too

Citibank's FLEXCUBE system
User meant to transfer interest to lender and principal to wash account
Accidentally sent $900m principal

baxter infusion pump, 2023

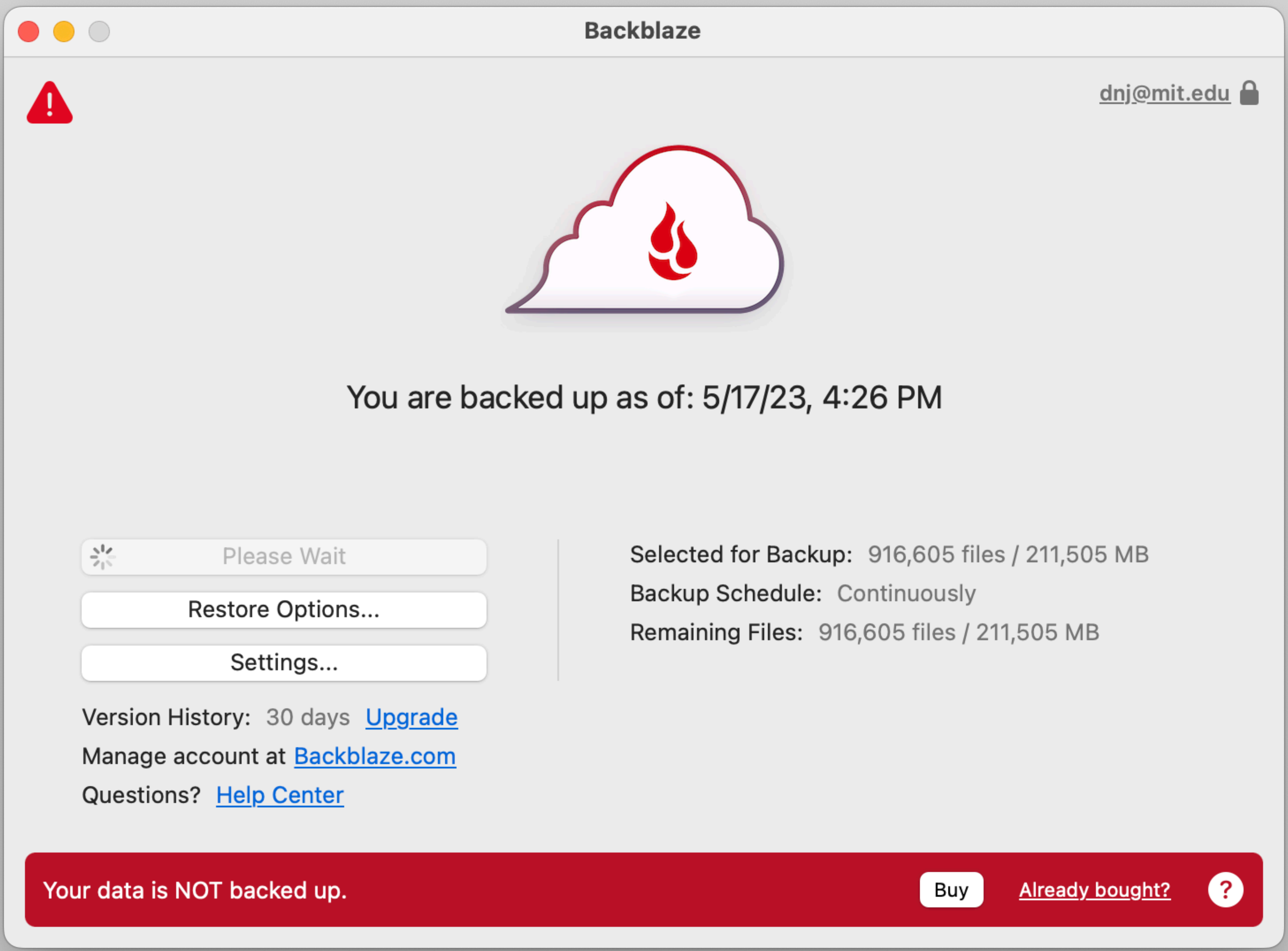
Baxter infusion pump event (FDA, May 2023)
Software upgrade: pump is stopped until alarms clear

Nurses didn't hear alarm, so drug delivery stopped

FDA reports 500 deaths in 5 years from infusion pumps

backblaze backup 2024

backing up on Backblaze



Backblaze

dnj@mit.edu

You are backed up as of: 5/17/23, 4:26 PM

Please Wait

Restore Options...

Settings...

Version History: 30 days Upgrade

Manage account at Backblaze.com

Questions? Help Center

Selected for Backup: 916,605 files / 211,505 MB

Backup Schedule: Continuously

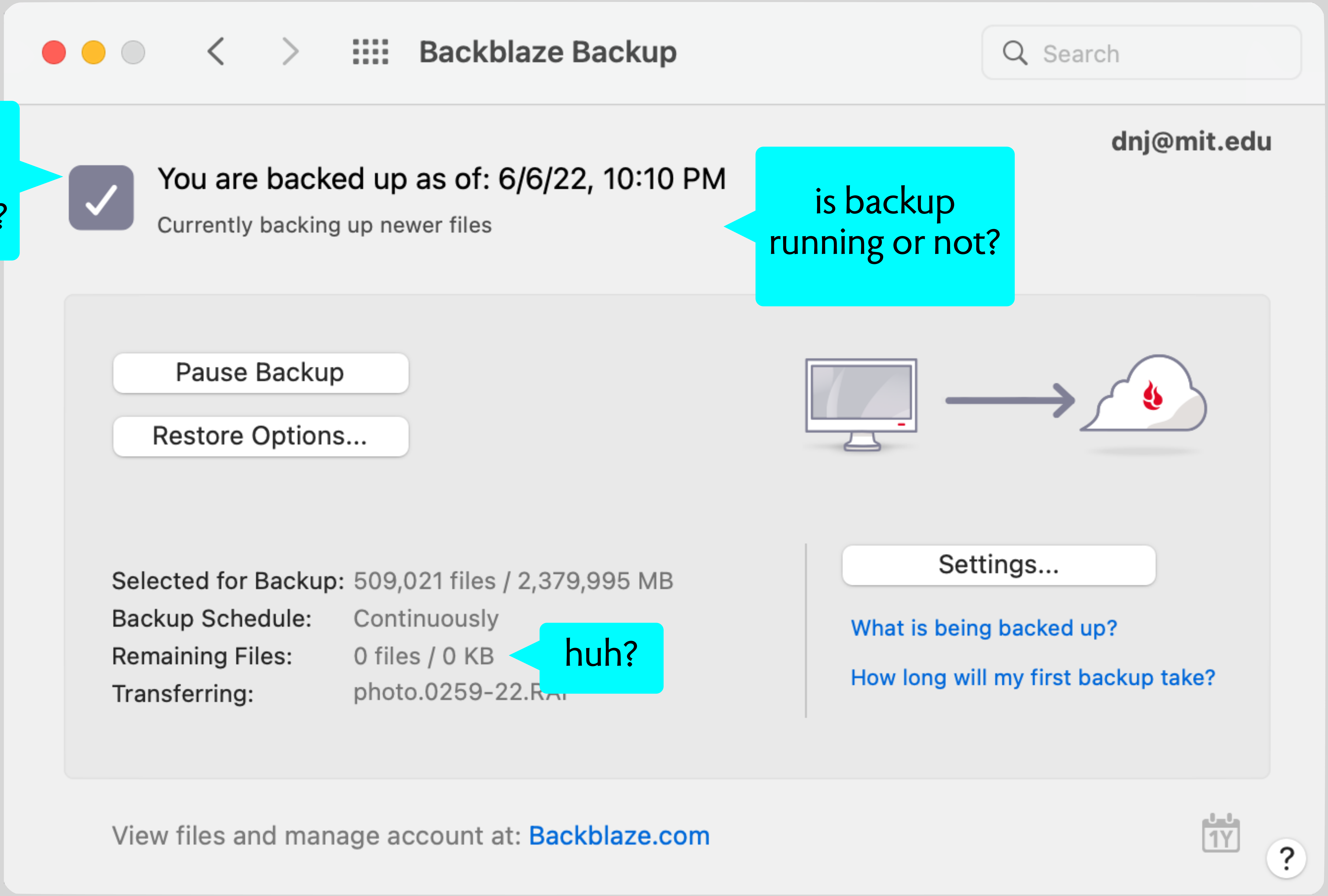
Remaining Files: 916,605 files / 211,505 MB

Your data is NOT backed up.

Buy

Already bought?

?



You are backed up as of: 6/6/22, 10:10 PM

Currently backing up newer files

Selected for Backup: 509,021 files / 2,379,995 MB

Backup Schedule: Continuously

Remaining Files: 0 files / 0 KB

Transferring: photo.0259-22.RAF

View files and manage account at: Backblaze.com

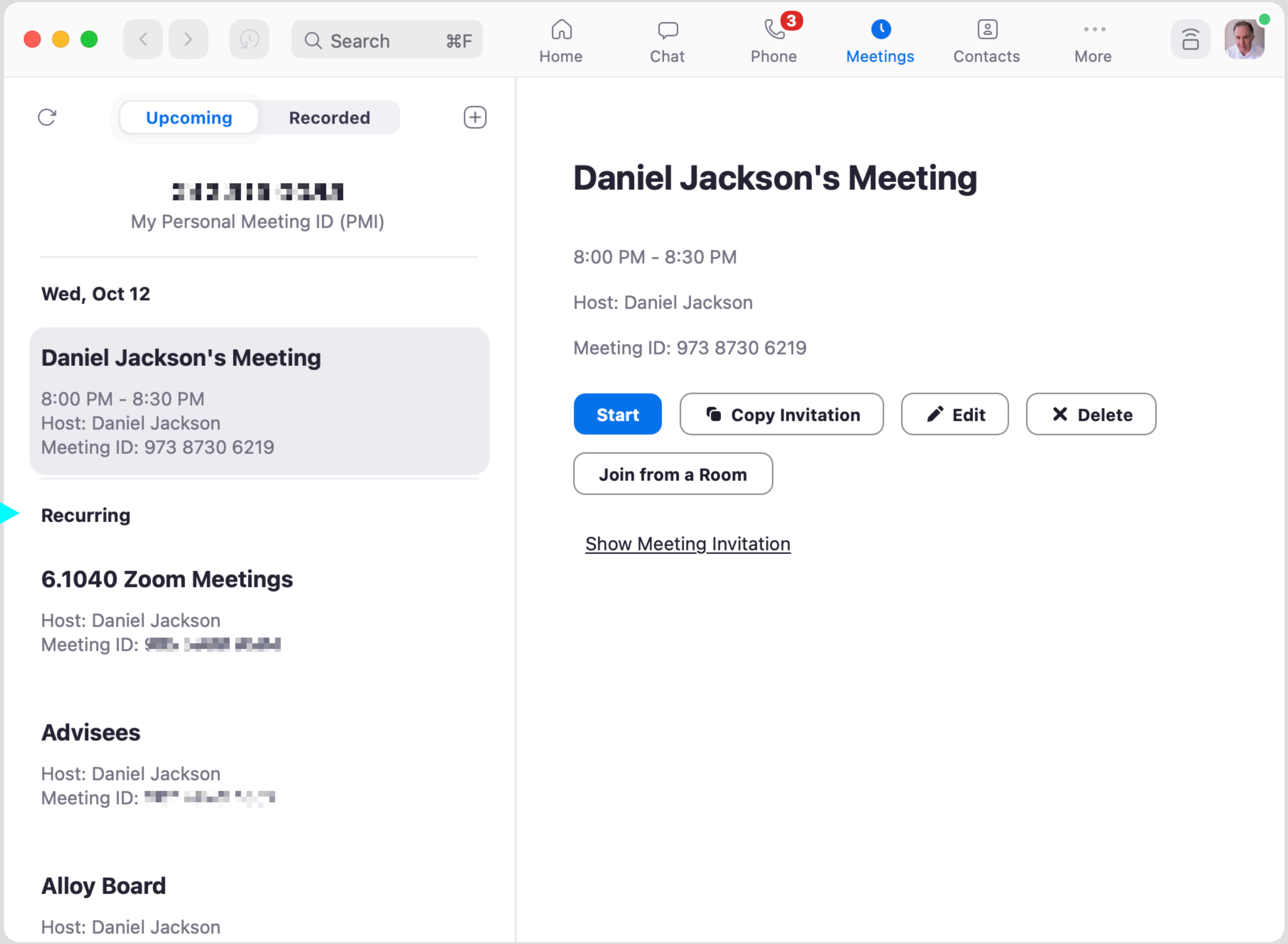
was modification at 10pm saved?

is backup running or not?

huh?

zoom meeting list, 2023

zoom's meeting list (2023)



[redacted]
My Personal Meeting ID (PMI)

Wed, Oct 12

Daniel Jackson's Meeting
8:00 PM - 8:30 PM
Host: Daniel Jackson
Meeting ID: 973 8730 6219

Recurring

6.1040 Zoom Meetings
Host: Daniel Jackson
Meeting ID: [redacted]

Advisees
Host: Daniel Jackson
Meeting ID: [redacted]

Alloy Board
Host: Daniel Jackson

Daniel Jackson's Meeting

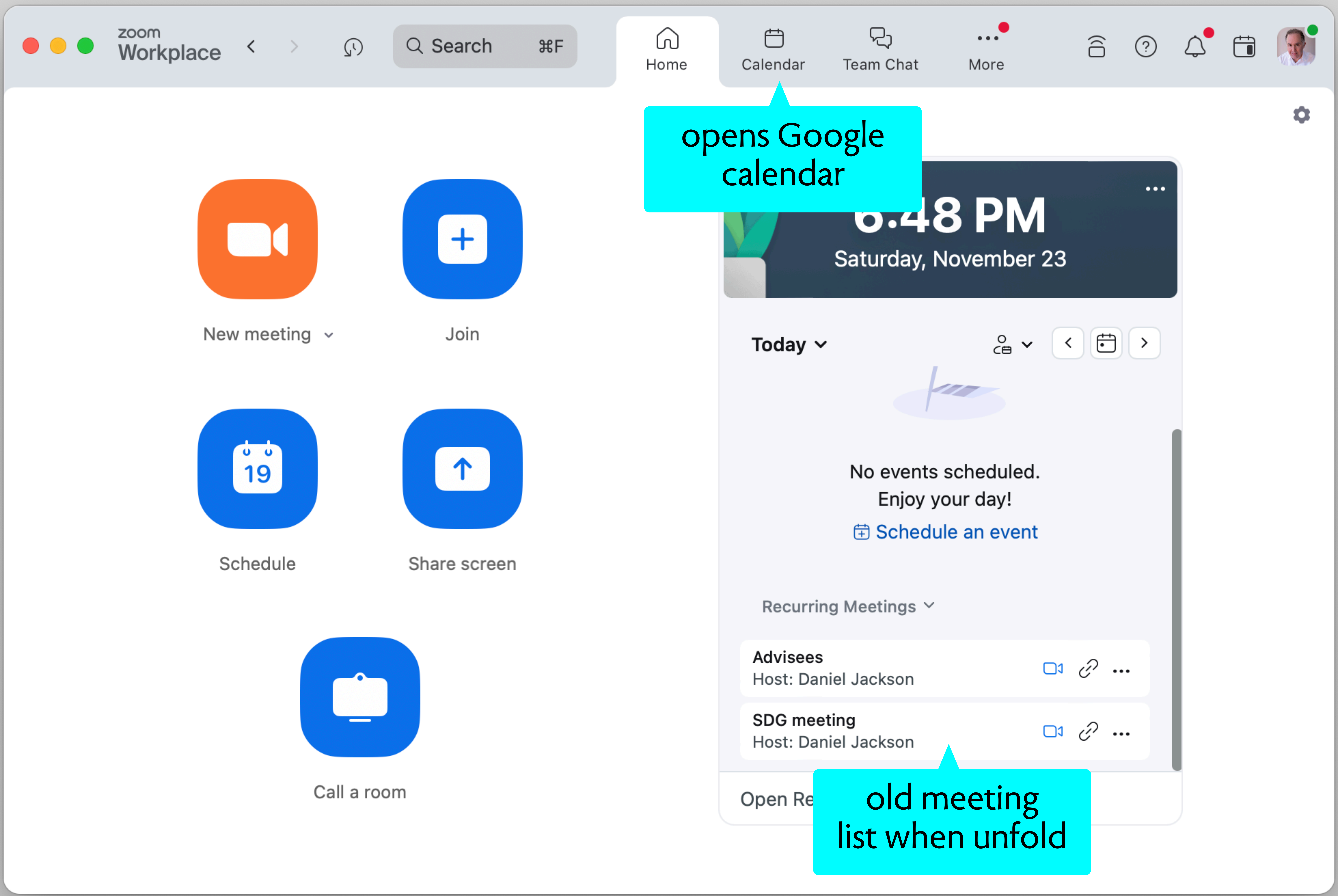
8:00 PM - 8:30 PM

Host: Daniel Jackson

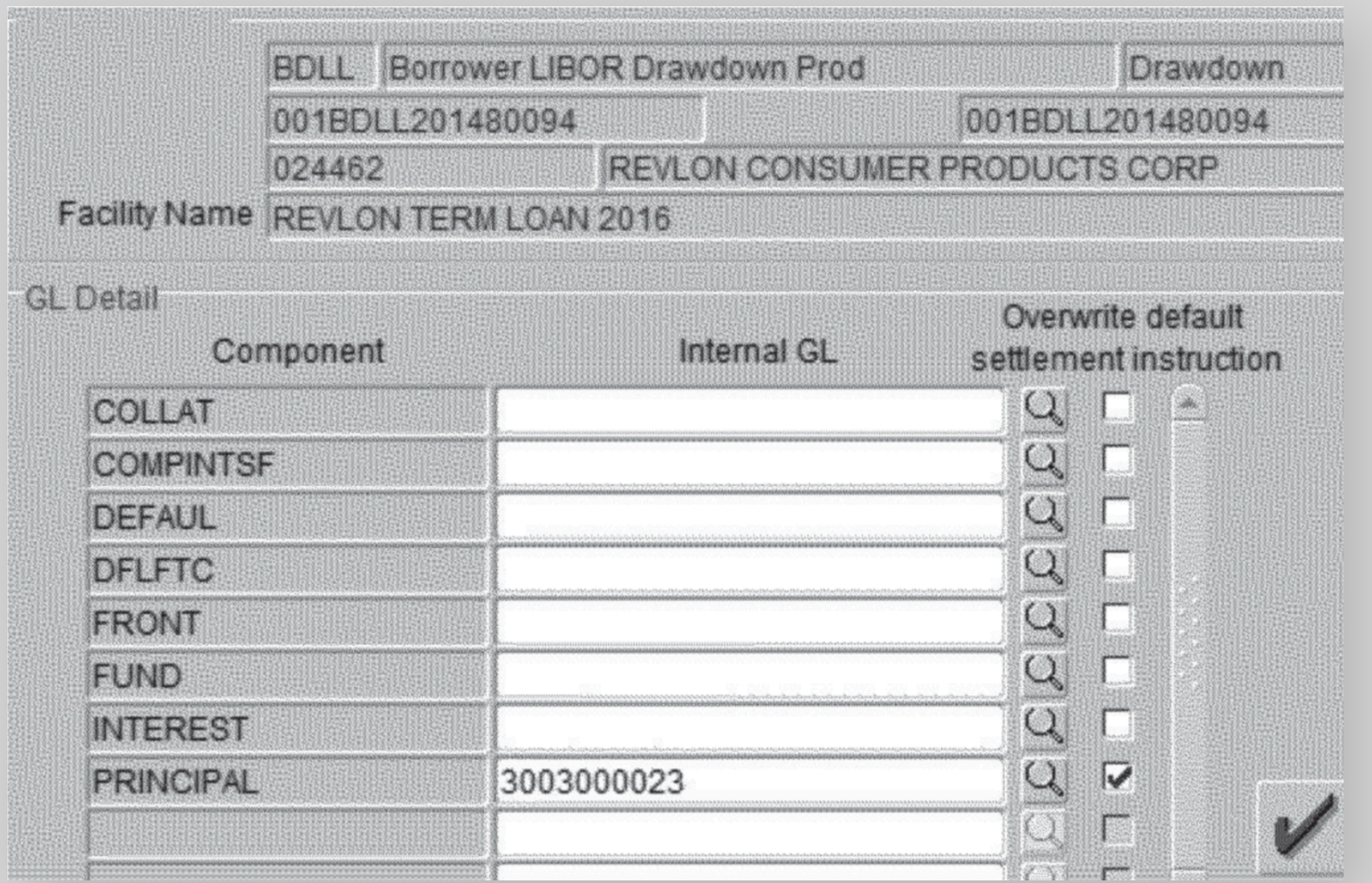
Meeting ID: 973 8730 6219

only meetings *you* scheduled

meeting list deemphasized (mid-2024?)



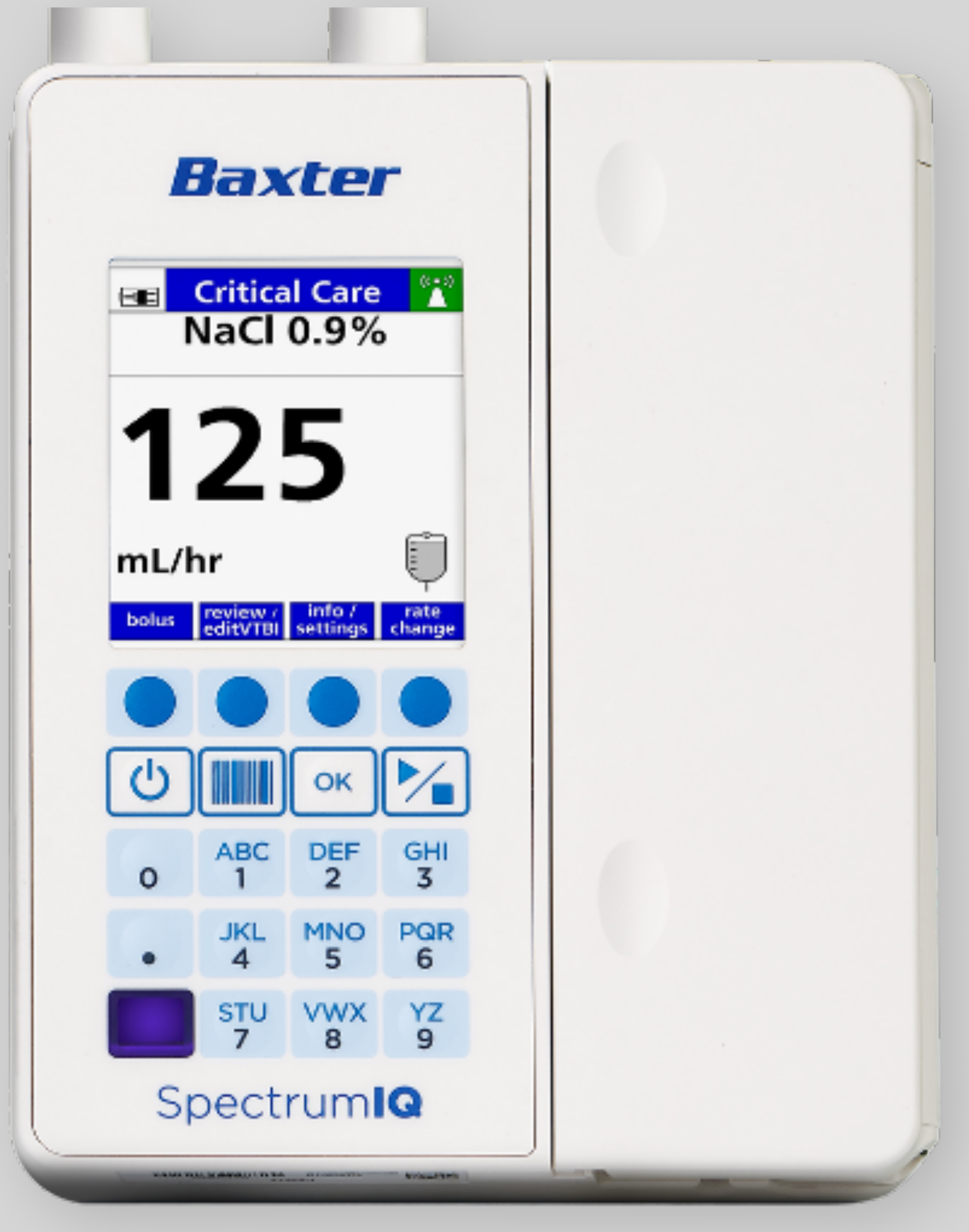
an exercise

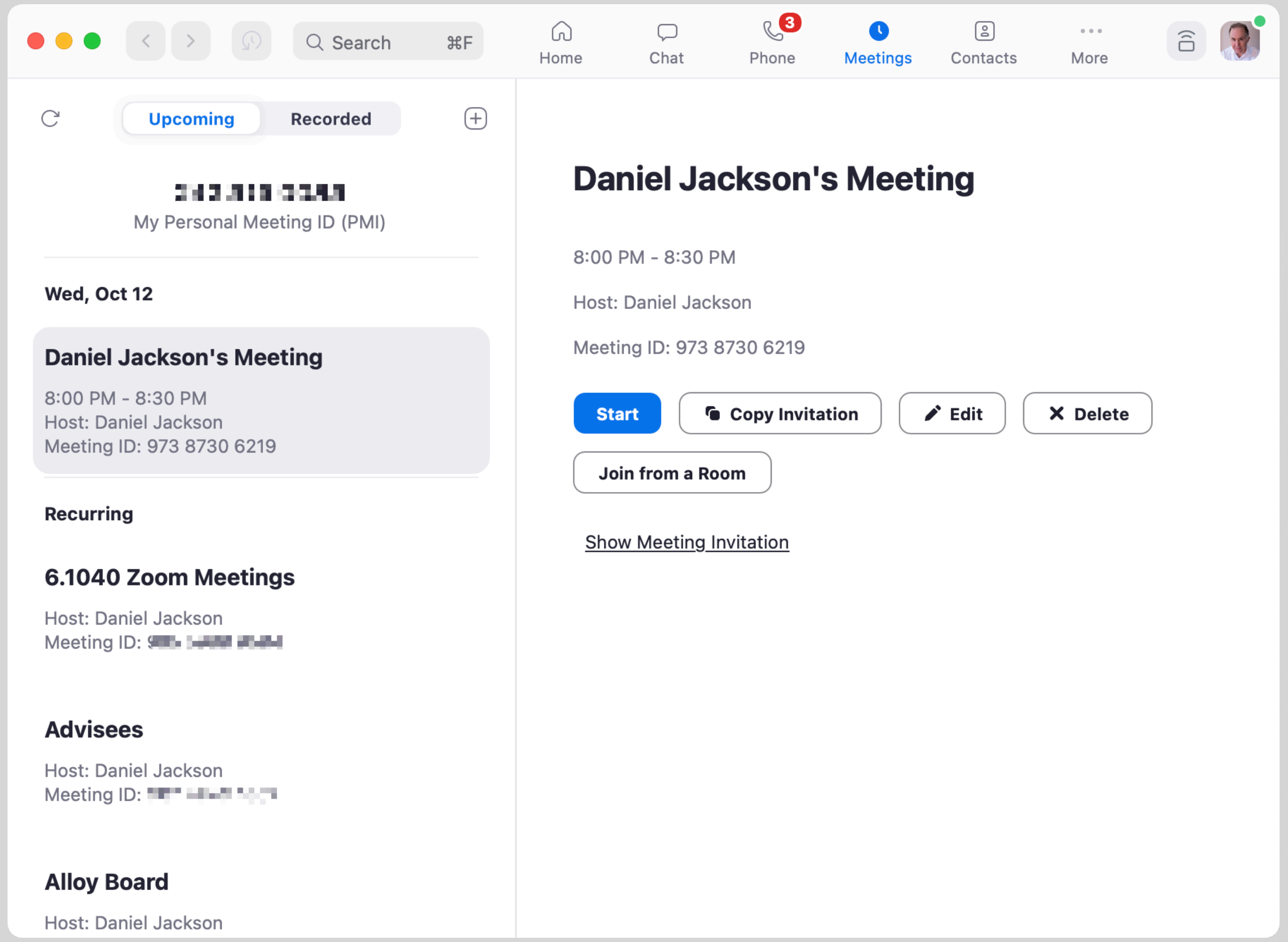


BDLL	Borrower LIBOR Drawdown Prod	Drawdown
001BDLL201480094		001BDLL201480094
024462	REVLON CONSUMER PRODUCTS CORP	
Facility Name	REVLON TERM LOAN 2016	

GL Detail

Component	Internal GL	Overwrite default settlement instruction
COLLAT		☐
COMPINTSF		☐
DEFAUL		☐
DFLFTC		☐
FRONT		☐
FUND		☐
INTEREST		☐
PRINCIPAL	3003000023	☑
		☐





[redacted]
My Personal Meeting ID (PMI)

Wed, Oct 12

Daniel Jackson's Meeting
8:00 PM - 8:30 PM
Host: Daniel Jackson
Meeting ID: 973 8730 6219

Recurring

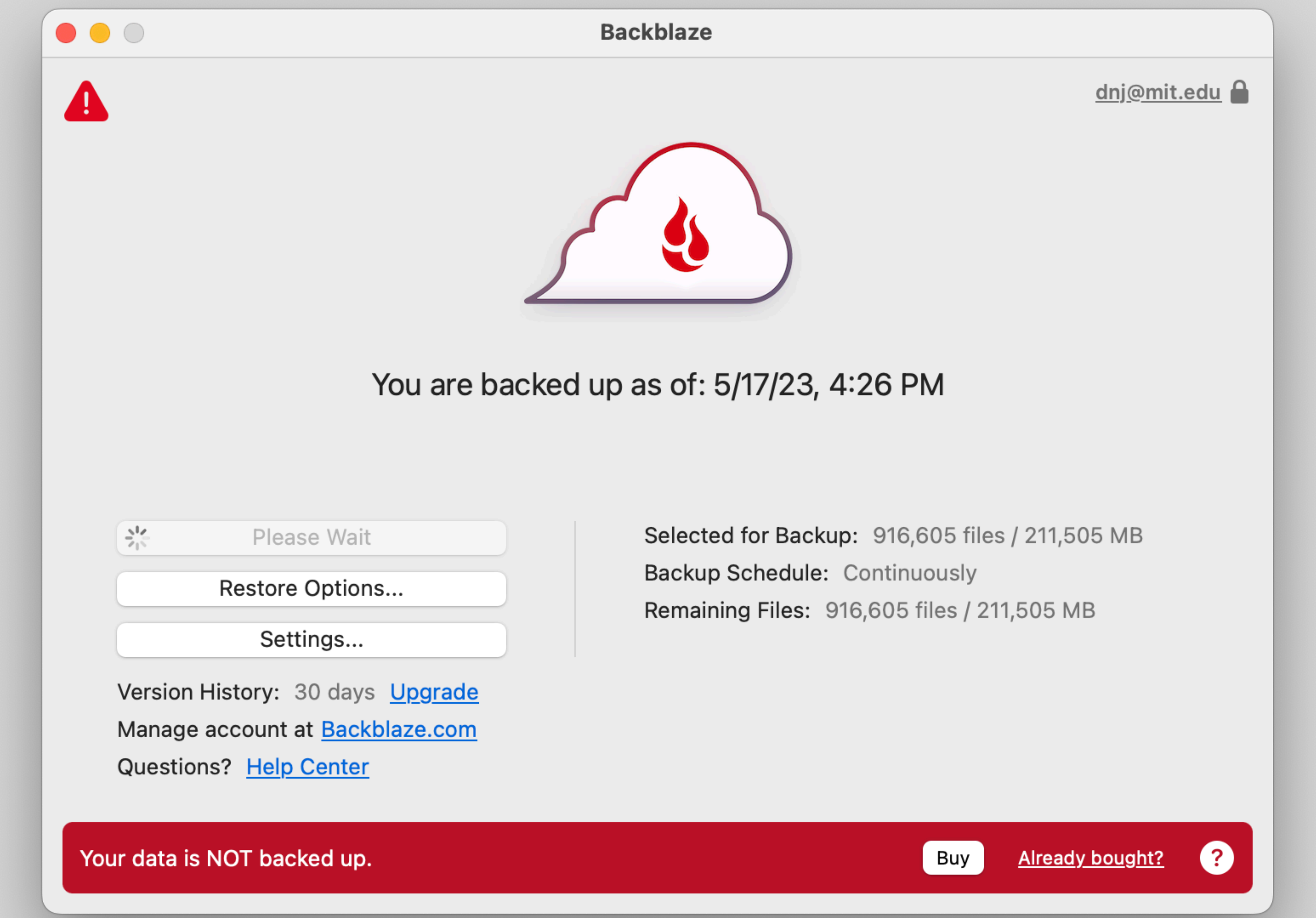
6.1040 Zoom Meetings
Host: Daniel Jackson
Meeting ID: [redacted]

Advisees
Host: Daniel Jackson
Meeting ID: [redacted]

Alloy Board
Host: Daniel Jackson

Daniel Jackson's Meeting

8:00 PM - 8:30 PM
Host: Daniel Jackson
Meeting ID: 973 8730 6219



You are backed up as of: 5/17/23, 4:26 PM

Selected for Backup: 916,605 files / 211,505 MB
Backup Schedule: Continuously
Remaining Files: 916,605 files / 211,505 MB

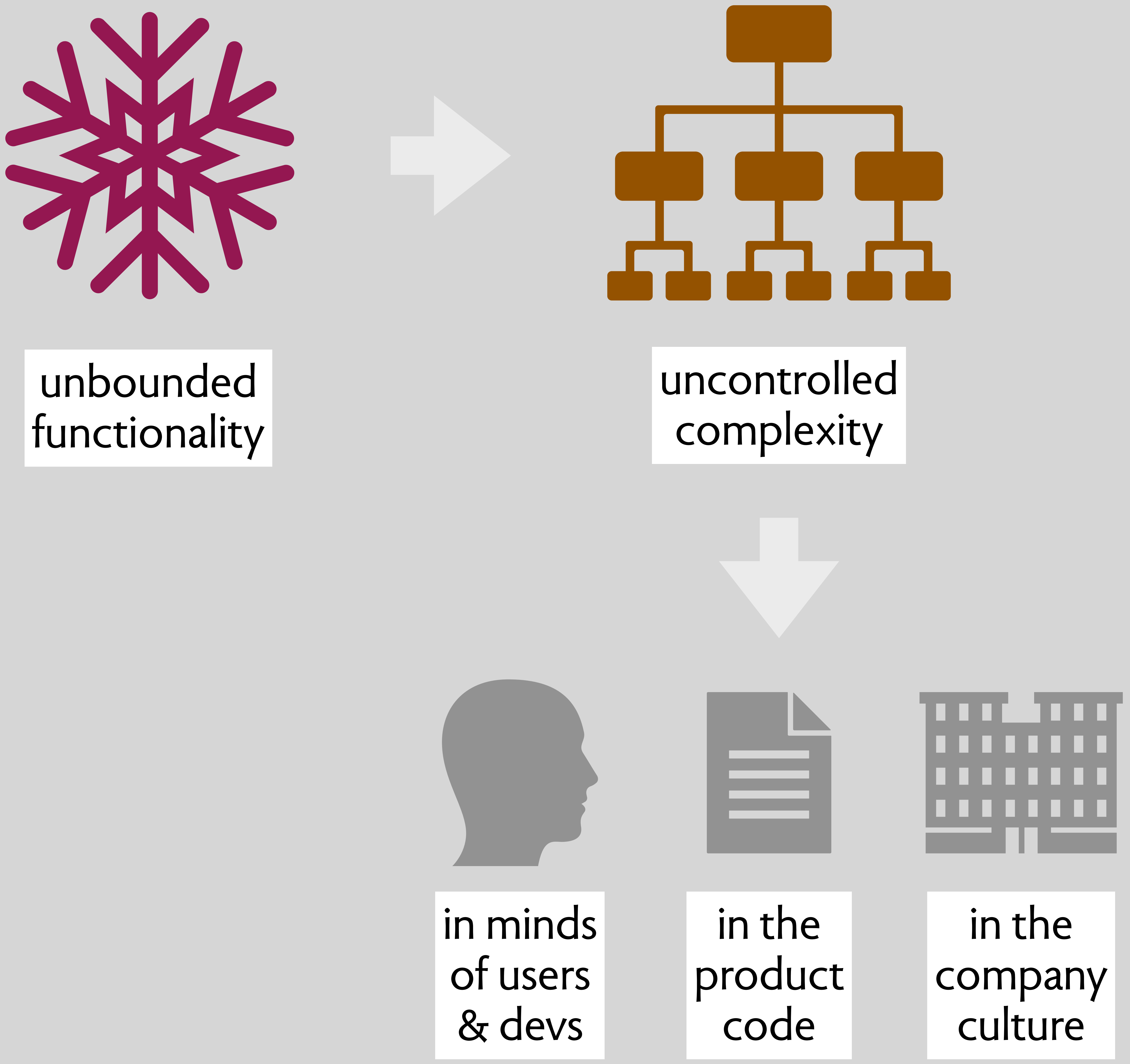
Version History: 30 days Upgrade
Manage account at Backblaze.com
Questions? Help Center

Your data is NOT backed up.

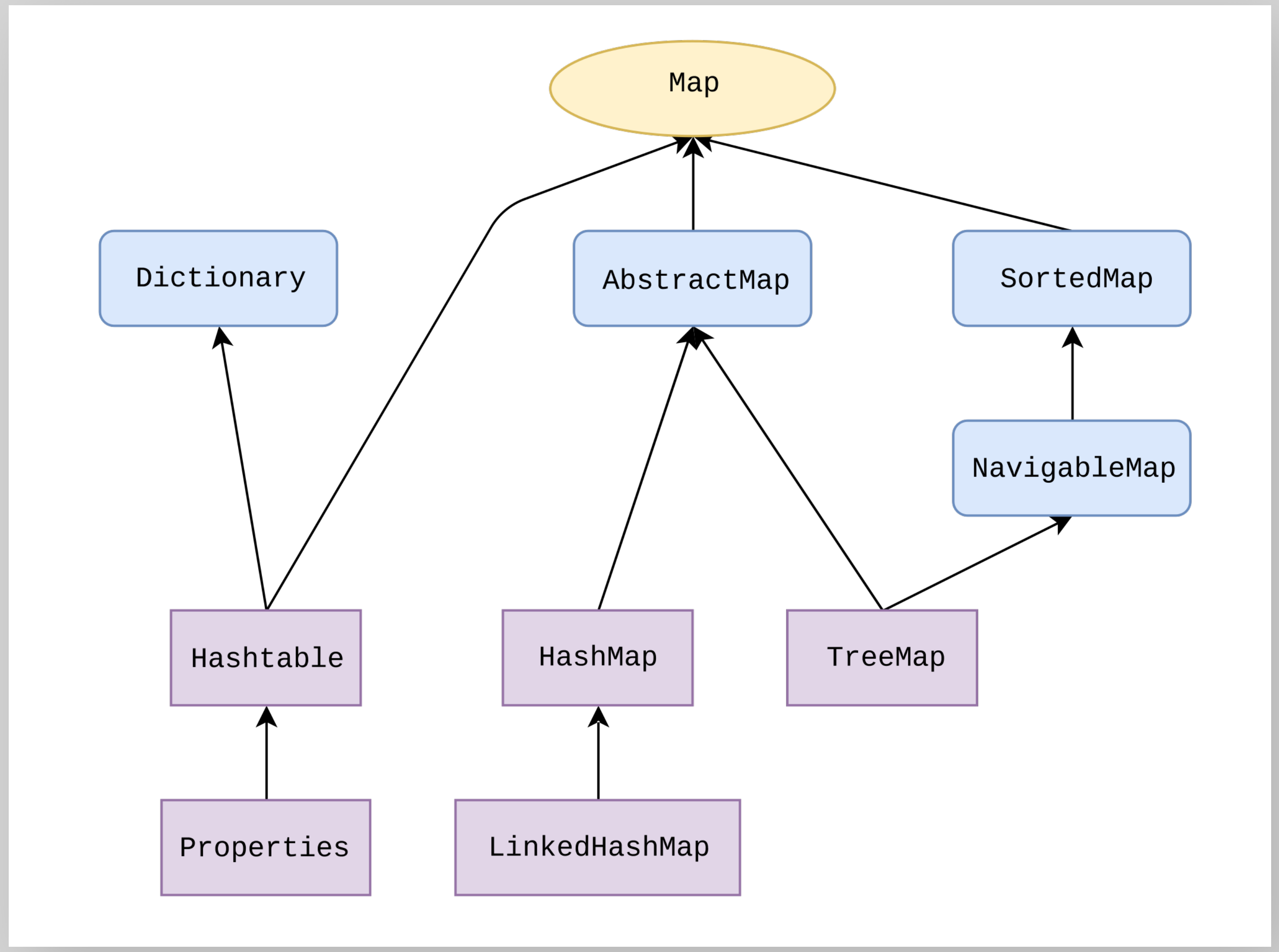
first, in pairs
pick one of the examples
1. how bad is this problem?
2. what's the root cause?
3. do you have any similar experiences?

then, together
are there repeating themes here?
any relevance to your products?

a diagnosis

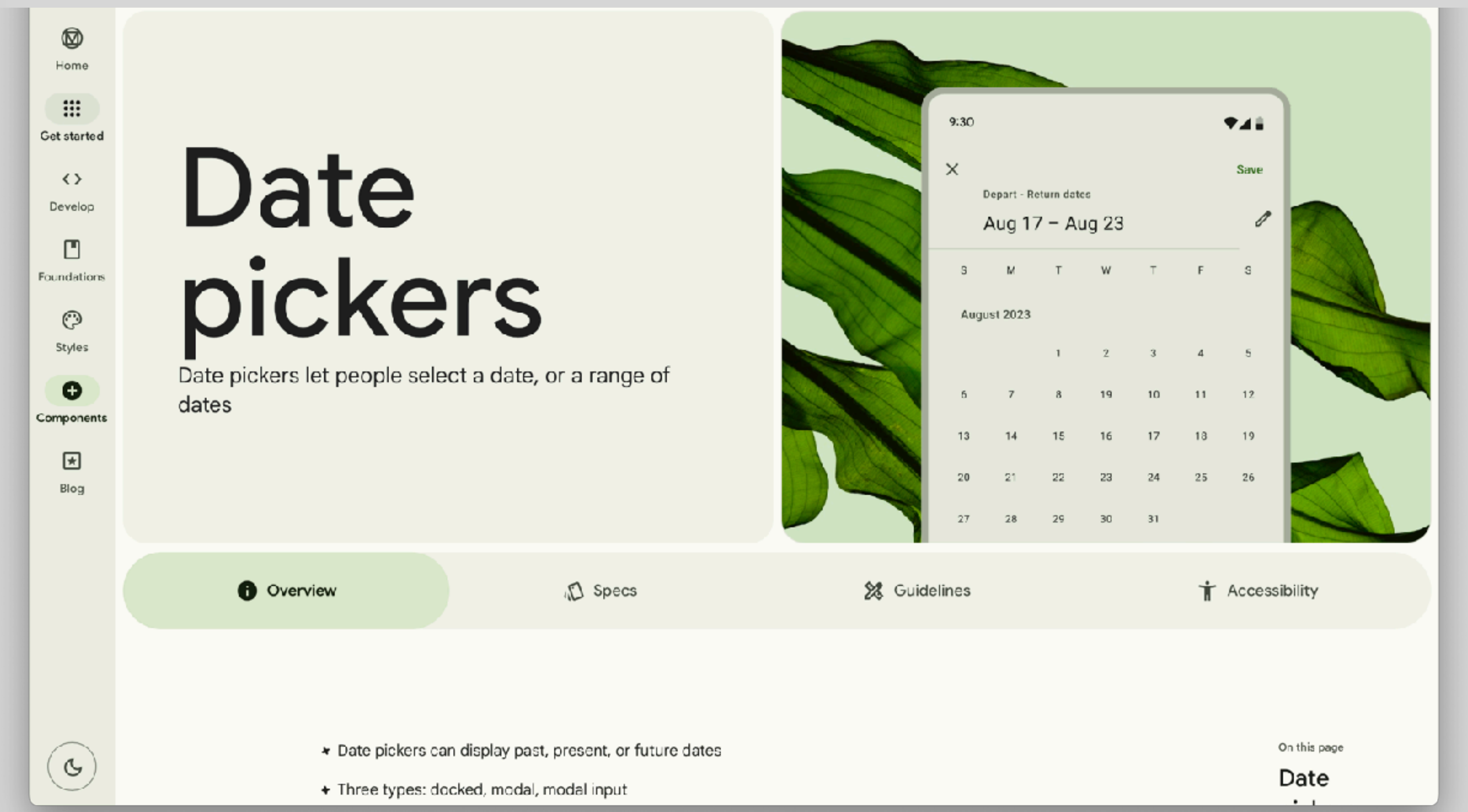

unbounded functionality
uncontrolled complexity
in minds of users & devs
in the product code
in the company culture

strategies for taming complexity



modularity

ways to structure **code**



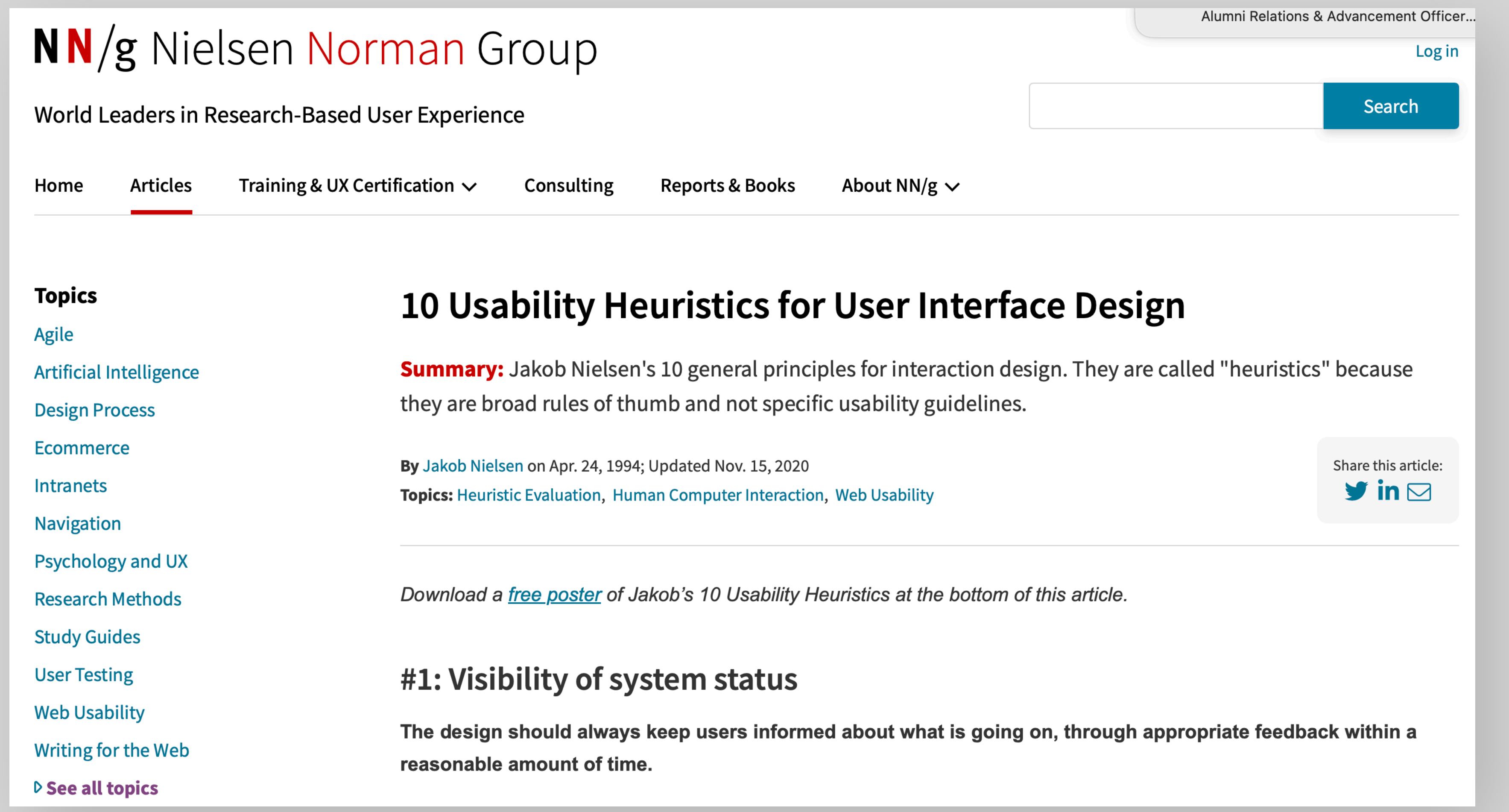
Date pickers

Date pickers let people select a date, or a range of dates

- Date pickers can display past, present, or future dates
- Three types: docked, modal, modal input

reuse & familiarity

standards for **UIs**



10 Usability Heuristics for User Interface Design

Summary: Jakob Nielsen's 10 general principles for interaction design. They are called "heuristics" because they are broad rules of thumb and not specific usability guidelines.

By Jakob Nielsen on Apr. 24, 1994; Updated Nov. 15, 2020
Topics: Heuristic Evaluation, Human Computer Interaction, Web Usability

Download a free poster of Jakob's 10 Usability Heuristics at the bottom of this article.

#1: Visibility of system status

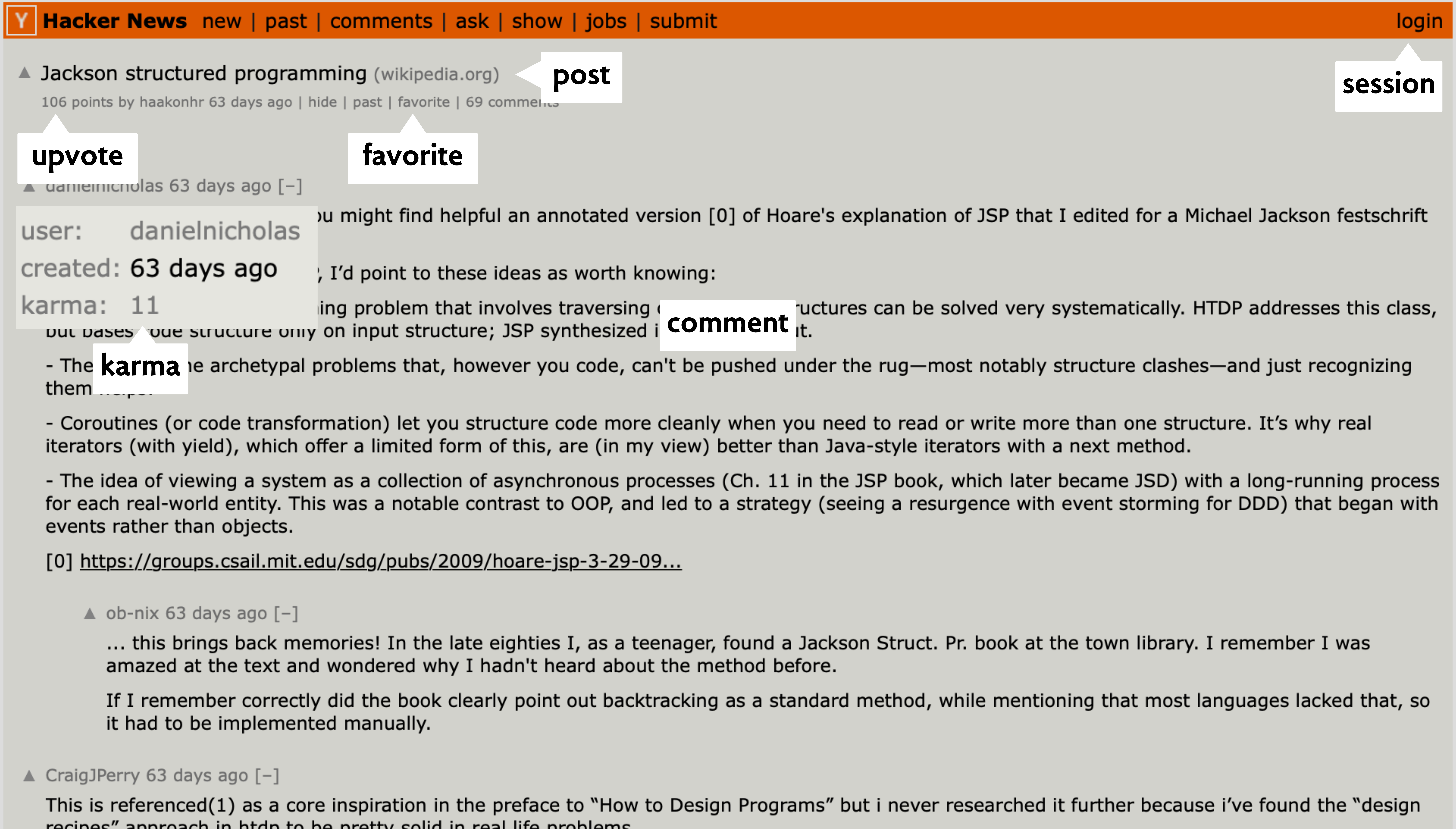
The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.

focus on users

broad **UX** principles

needed: a framework for designing functionality that aligns modularity, reuse and user-centeredness

concepts:
modular, reusable
& user-centric
units of function



Hacker News new | past | comments | ask | show | jobs | submit login

▲ Jackson structured programming (wikipedia.org)

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

▲ danielnicholas 63 days ago [–]

user: danielnicholas
created: 63 days ago
karma: 11

u 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 ructures can be solved very systematically. HTDP addresses this class, ode structure only on input structure; JSP synthesized i t.

- The e archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them

- Coroutines (or code transformation) let you structure code more cleanly when you need to read or write more than one structure. It's why real iterators (with yield), which offer a limited form of this, are (in my view) better than Java-style iterators with a next method.

- The idea of viewing a system as a collection of asynchronous processes (Ch. 11 in the JSP book, which later became JSD) with a long-running process for each real-world entity. This was a notable contrast to OOP, and led to a strategy (seeing a resurgence with event storming for DDD) that began with events rather than objects.

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

▲ ob-nix 63 days ago [–]

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

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

▲ 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

concept elements: name, purpose, principle

concept Upvote

purpose rank items by popularity

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

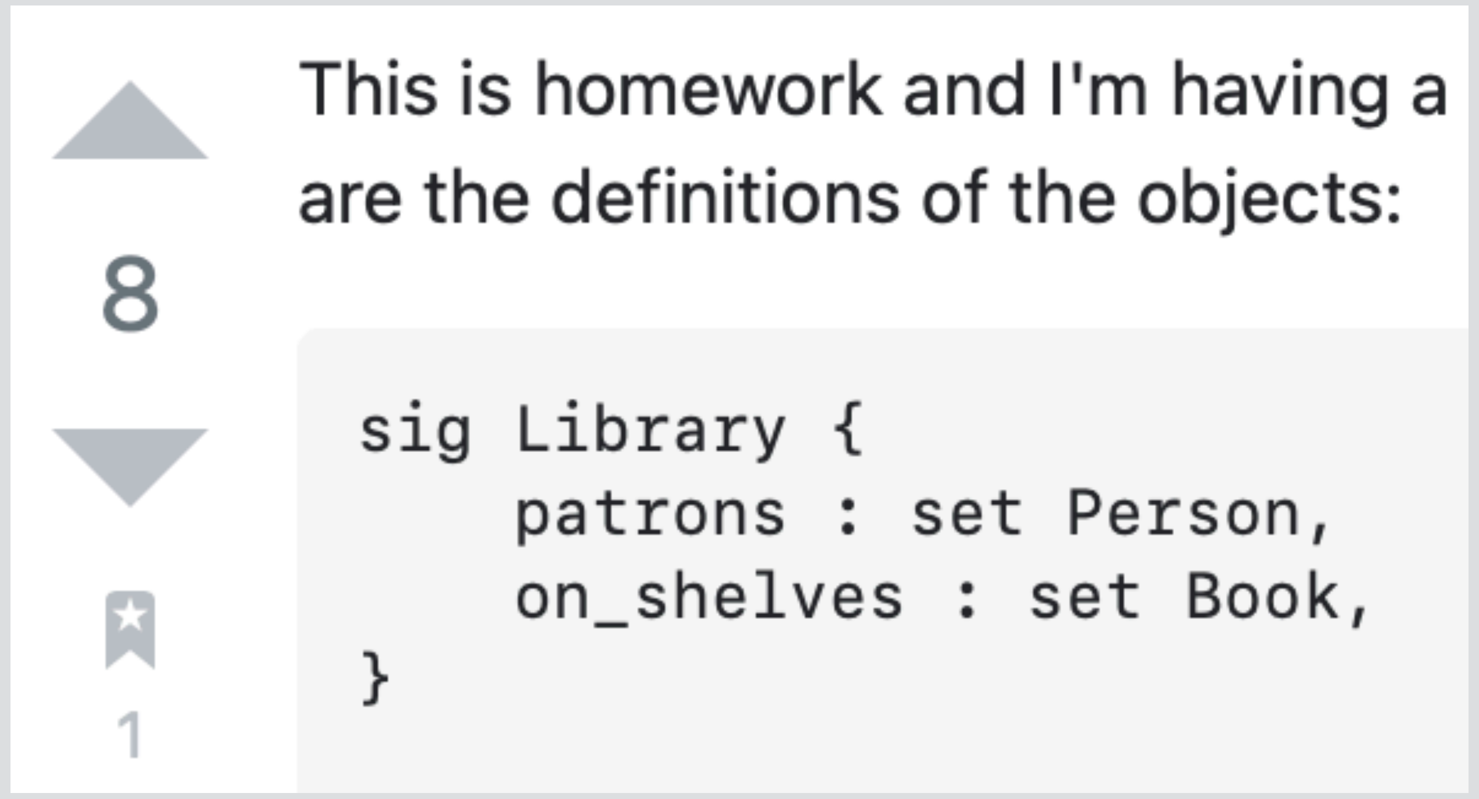
Michael Polanyi (1891-1976)

similar UIs, very different concepts

concept Upvote

purpose rank items by popularity

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



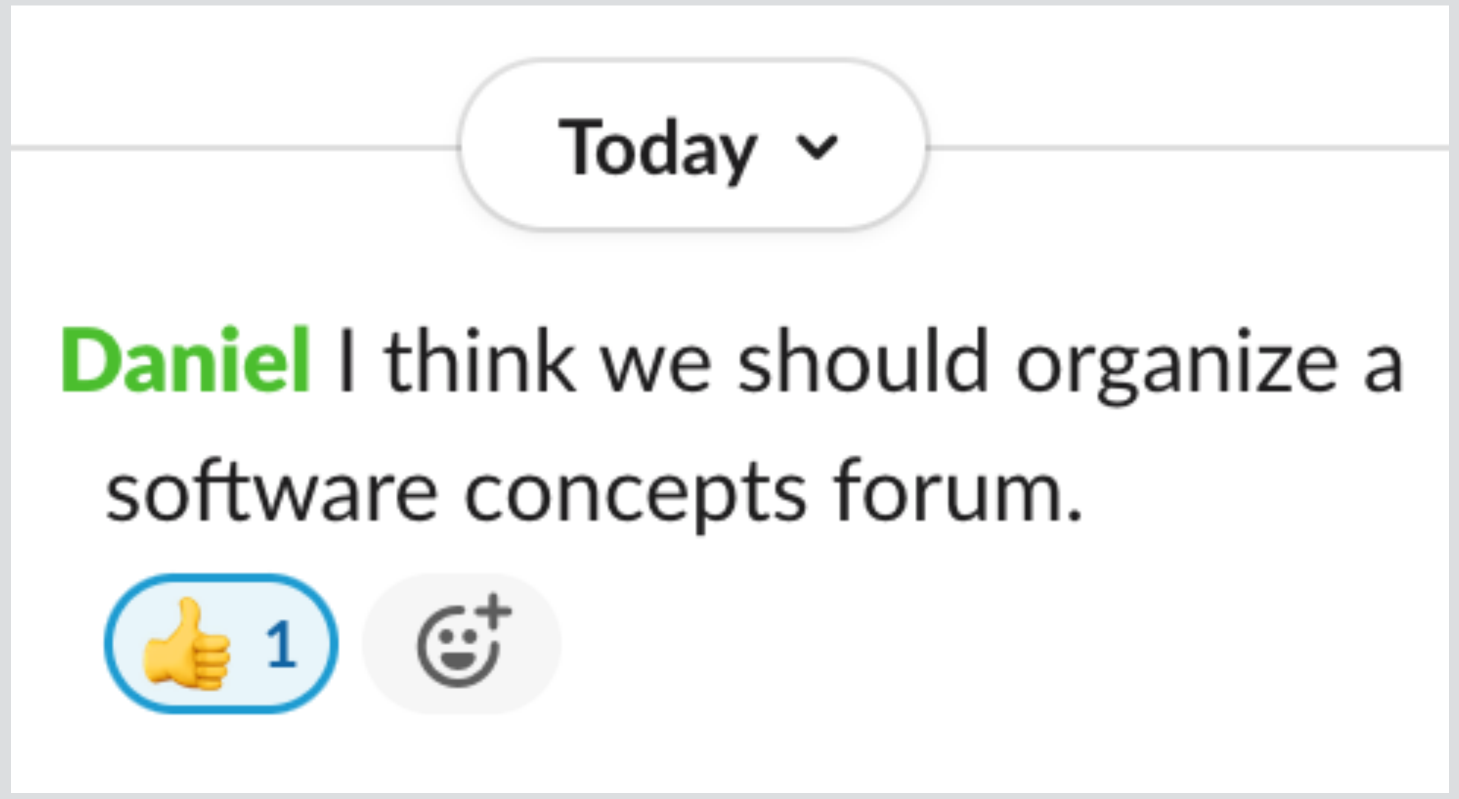
This is homework and I'm having a are the definitions of the objects:

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

concept Reaction

purpose support quick responses

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

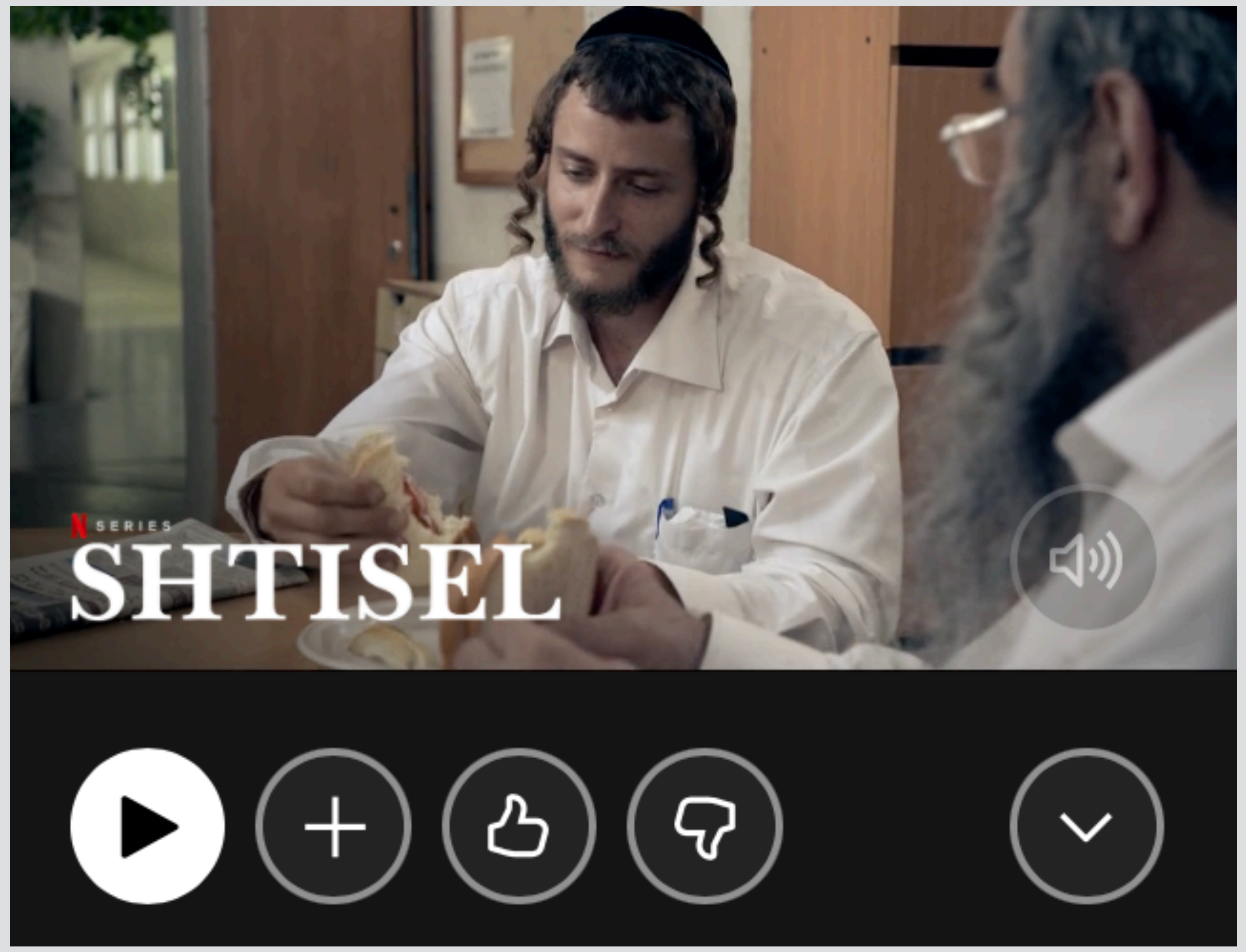


Daniel I think we should organize a software concepts forum.

concept Recommendation

purpose infer user preferences

principle user likes lead to ranking of kinds of items, thus which items are recommended



defining concept behavior in detail

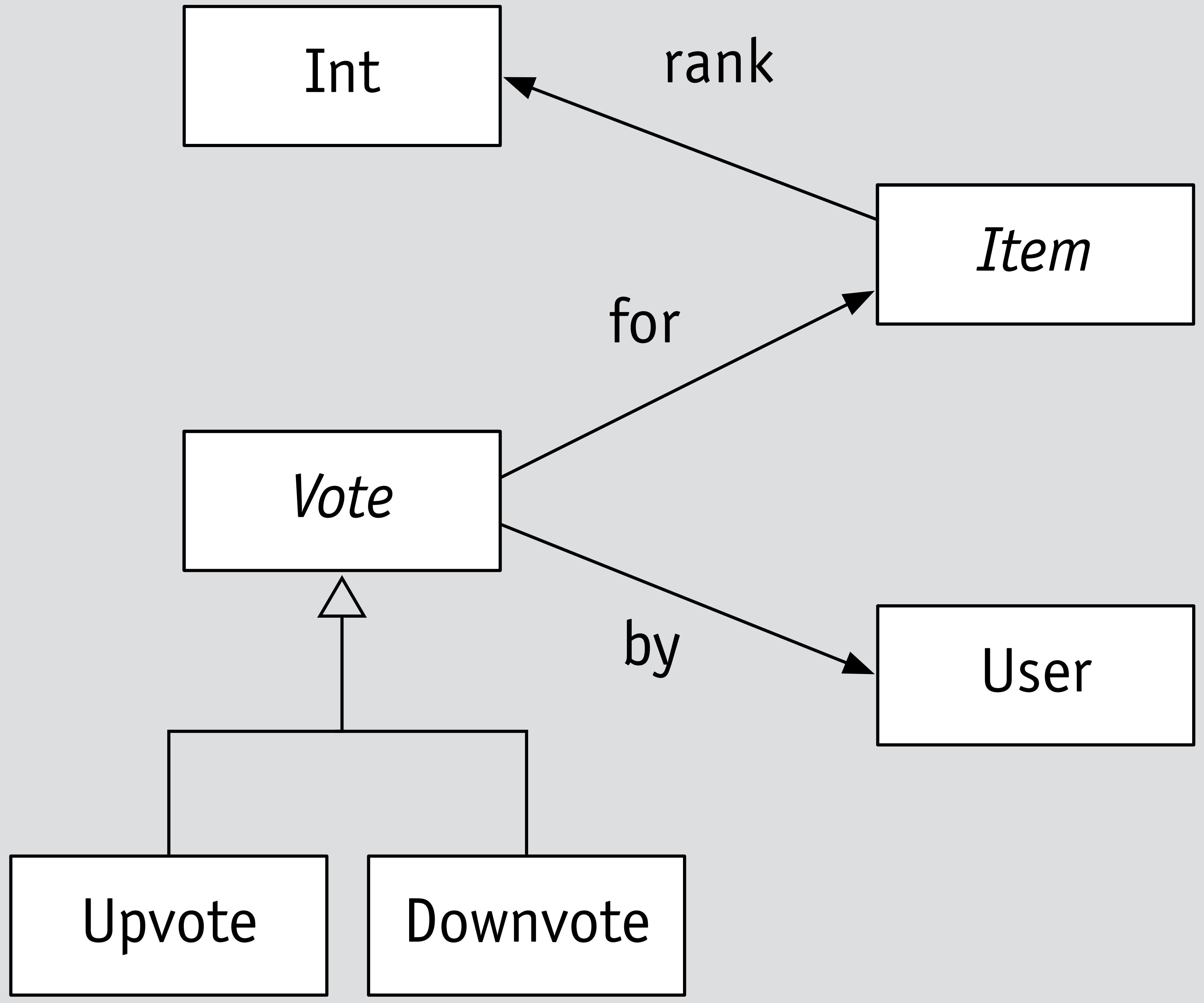
concept Upvote

purpose rank items by popularity

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

state
by: Vote -> one User
for: Vote -> one Item
Upvote, Downvote: set Vote
rank: Item -> one Int

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



downvote (i: Item, u: User)

```
  // no v: Downvote | v.for =  i and v.by = u
  // remove {v: Upvote | v.for = i and v.by = u}
  // add {v: Downvote | v.for = i and v.by = u}
  // update i.rank ...
```

concepts as carriers of design knowledge

concept: Upvote

related concepts
Rating, Recommendation, Reaction, ...

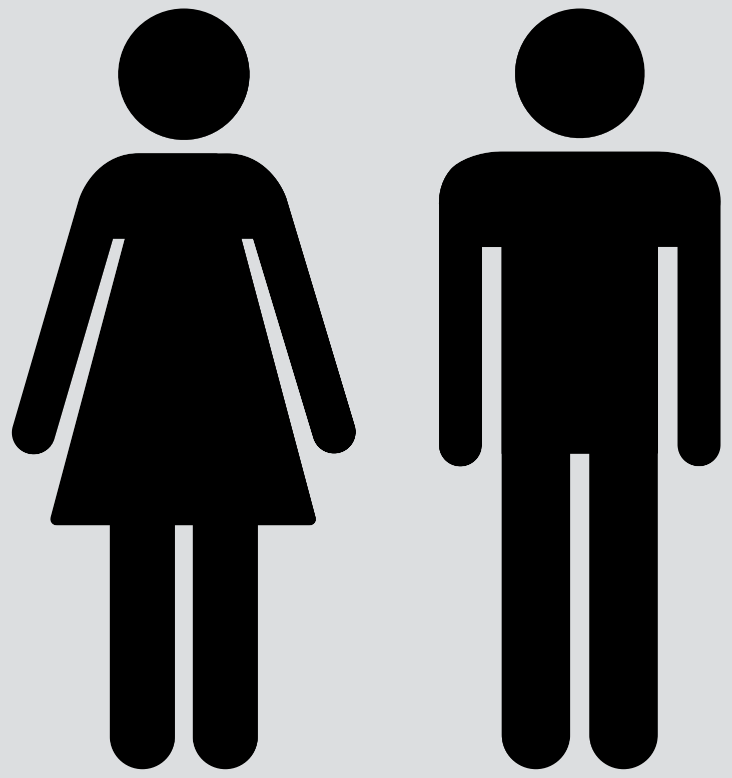
design variants
downvote as unvote
use age in ranking
weigh downvotes more
various identity tactics
freezing old posts

known issues
high votes can promote old content
feedback favors early upvotes
upvoting encourages echo chamber
preventing double votes

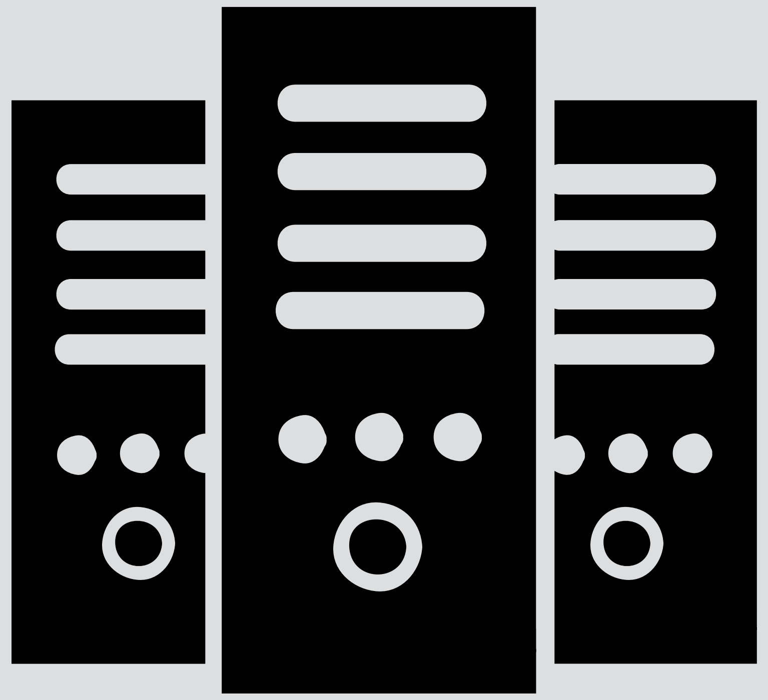
typical uses
social media posts
comments on articles
Q&A responses

often used with
Karma, Auth, ...

so what's a concept?



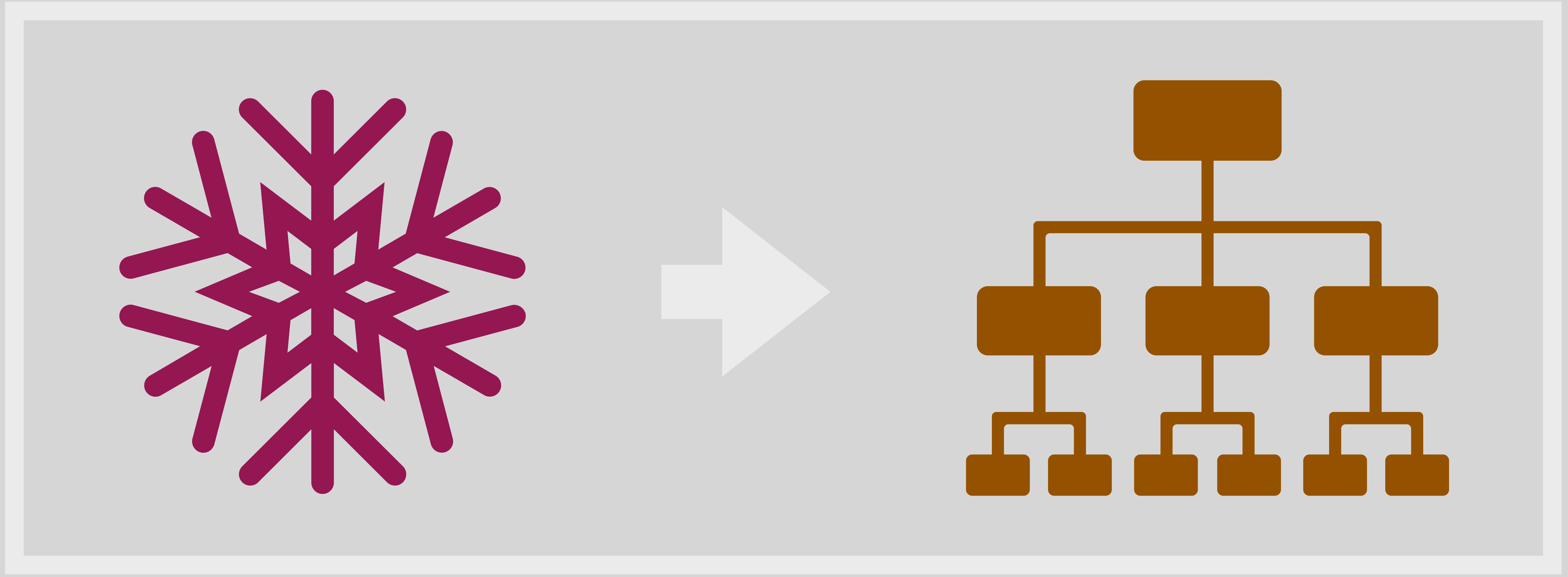
users' perspective
a behavioral protocol



software perspective
a "nanoservice"

takeaways

unbounded functionality leads to uncontrolled complexity



concepts bring modularity, reuse & user-centeredness

what next?

for any concept, we can ask:
why is it so widely used?
where did it come from?
is it just a computer concept?
how did it become so widely adopted?

the Session concept, implemented physically?