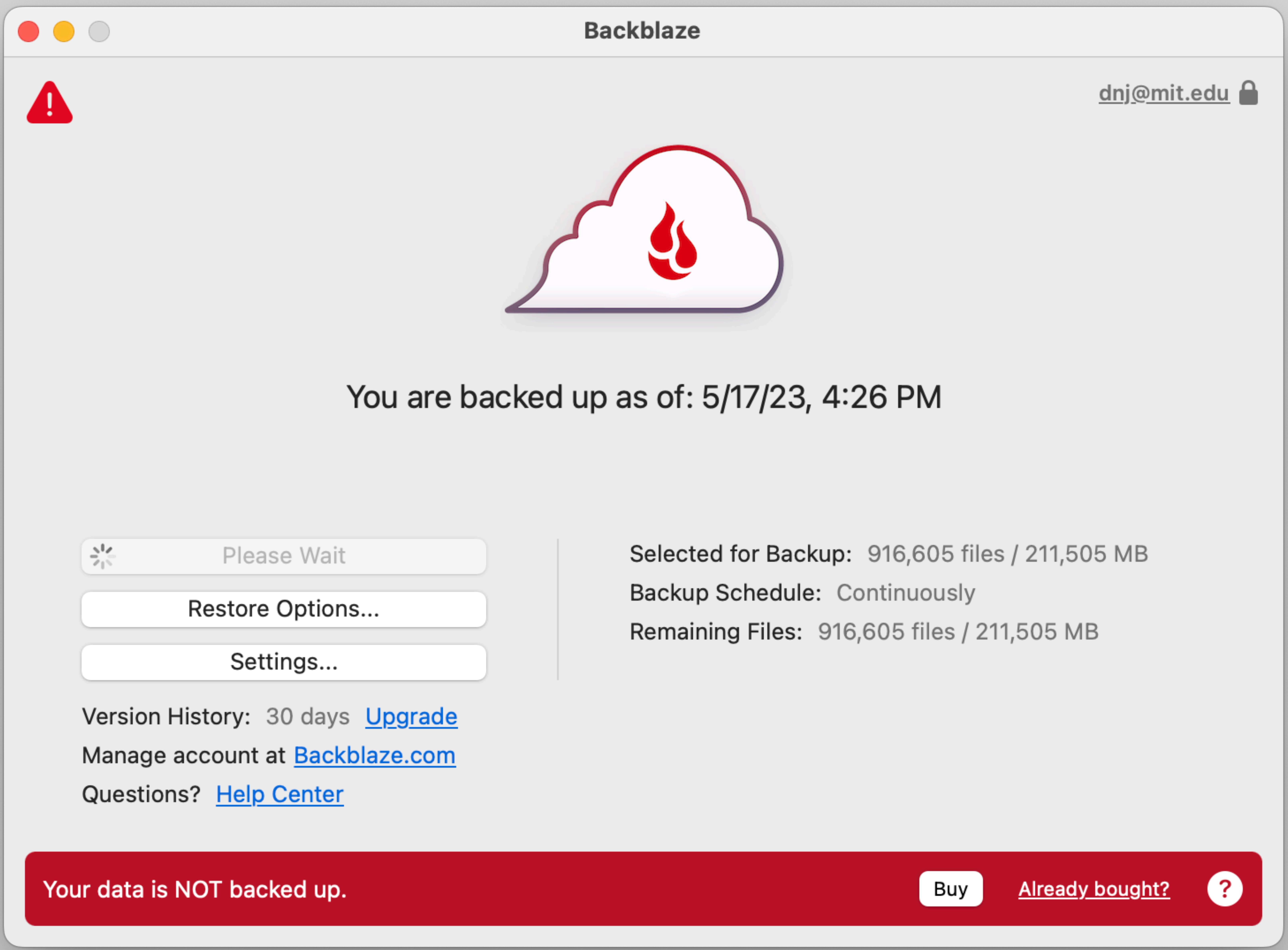
introducing concepts

Daniel Jackson · Autodesk, Boston · March 17-18, 2025

a UX puzzle: Backblaze

backing up on Backblaze

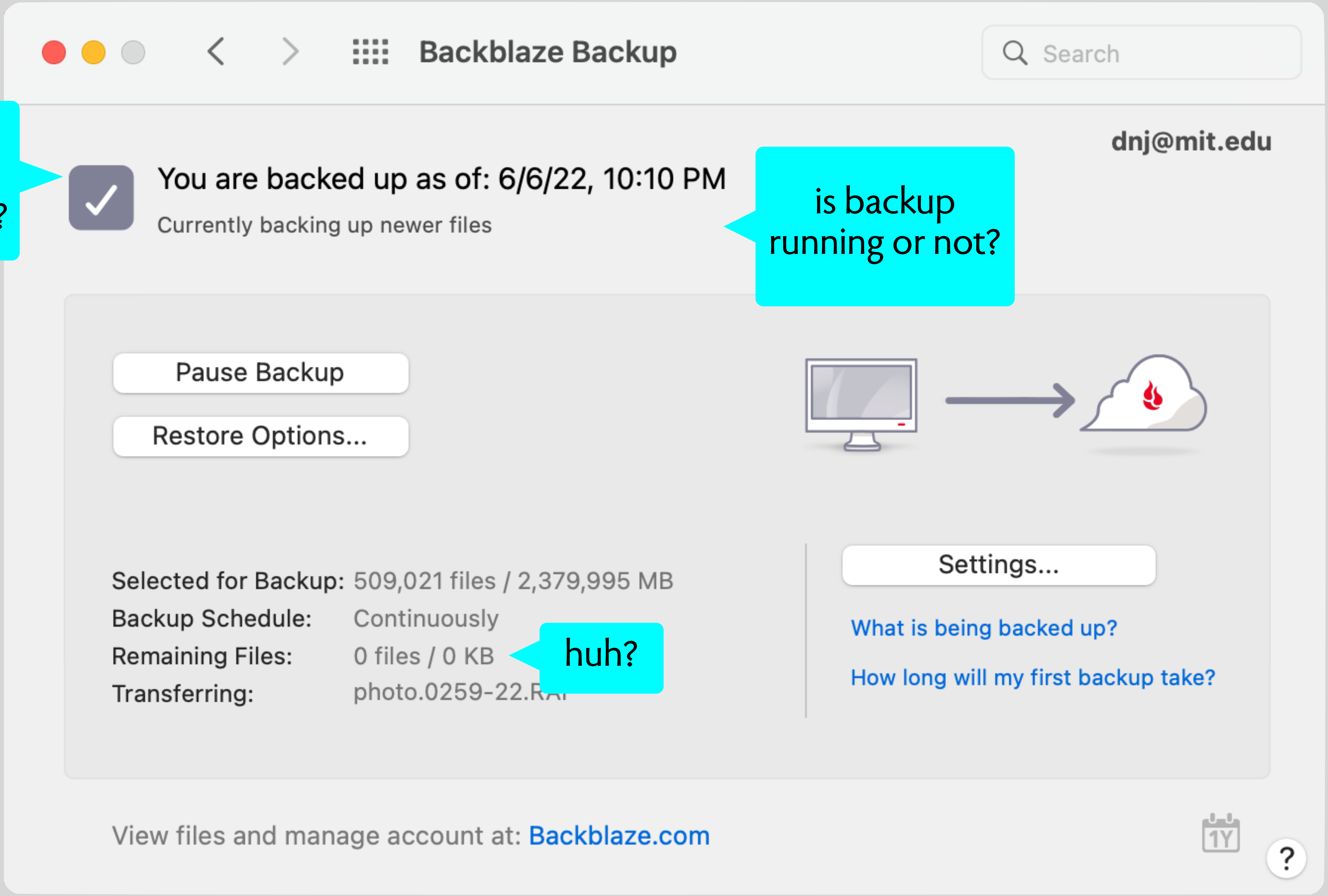


You are backed up as of: 5/17/23, 4:26 PM

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.



Backblaze Backup

Search

dnj@mit.edu

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

What is being backed up?

How long will my first backup take?

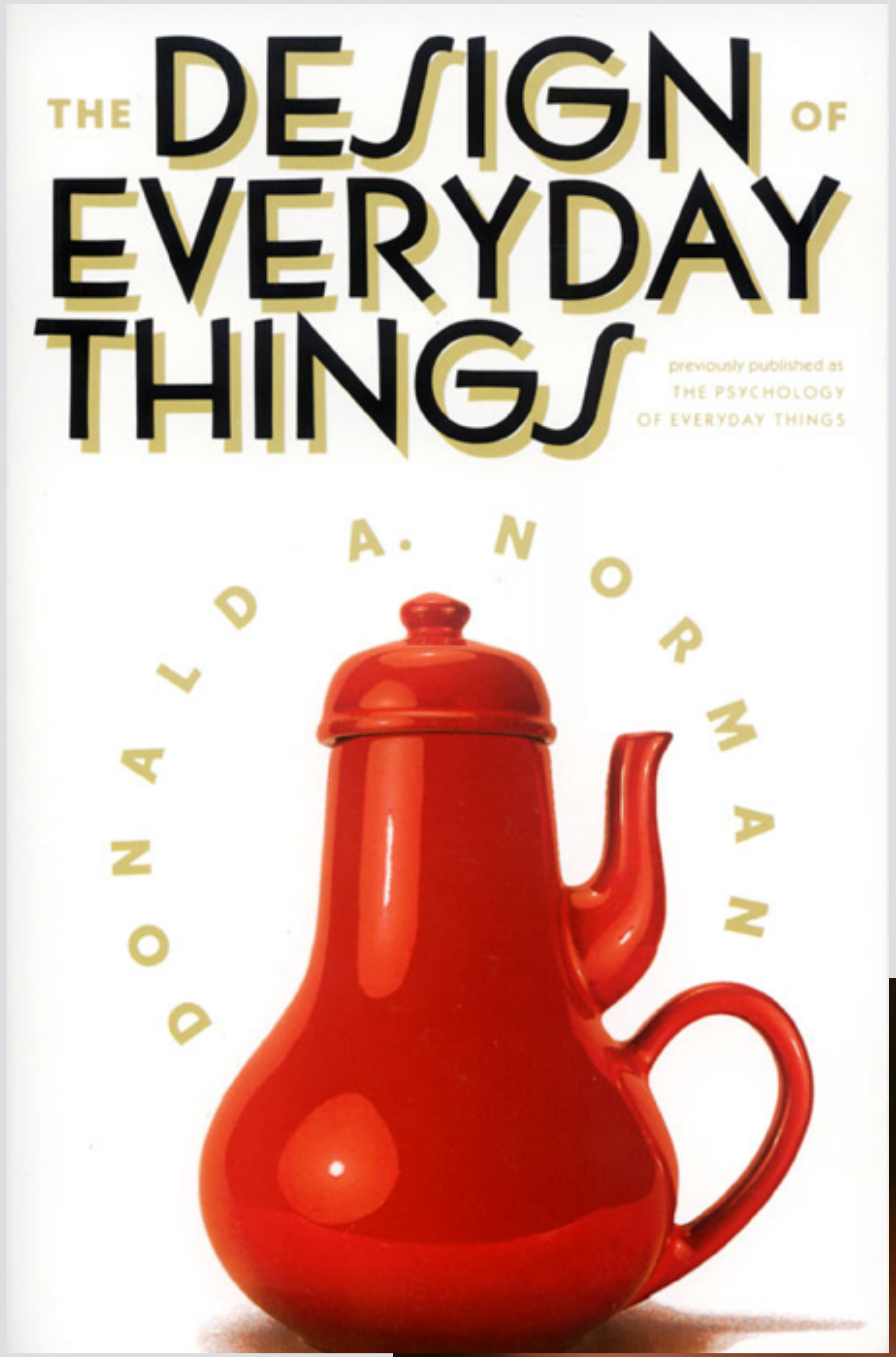
View files and manage account at: Backblaze.com

was modification at 10pm saved?

is backup running or not?

huh?

conceptual models to the rescue


THE DESIGN OF EVERYDAY THINGS
previously published as THE PSYCHOLOGY OF EVERYDAY THINGS
DONALD A. NORMAN

1988

Donald Norman

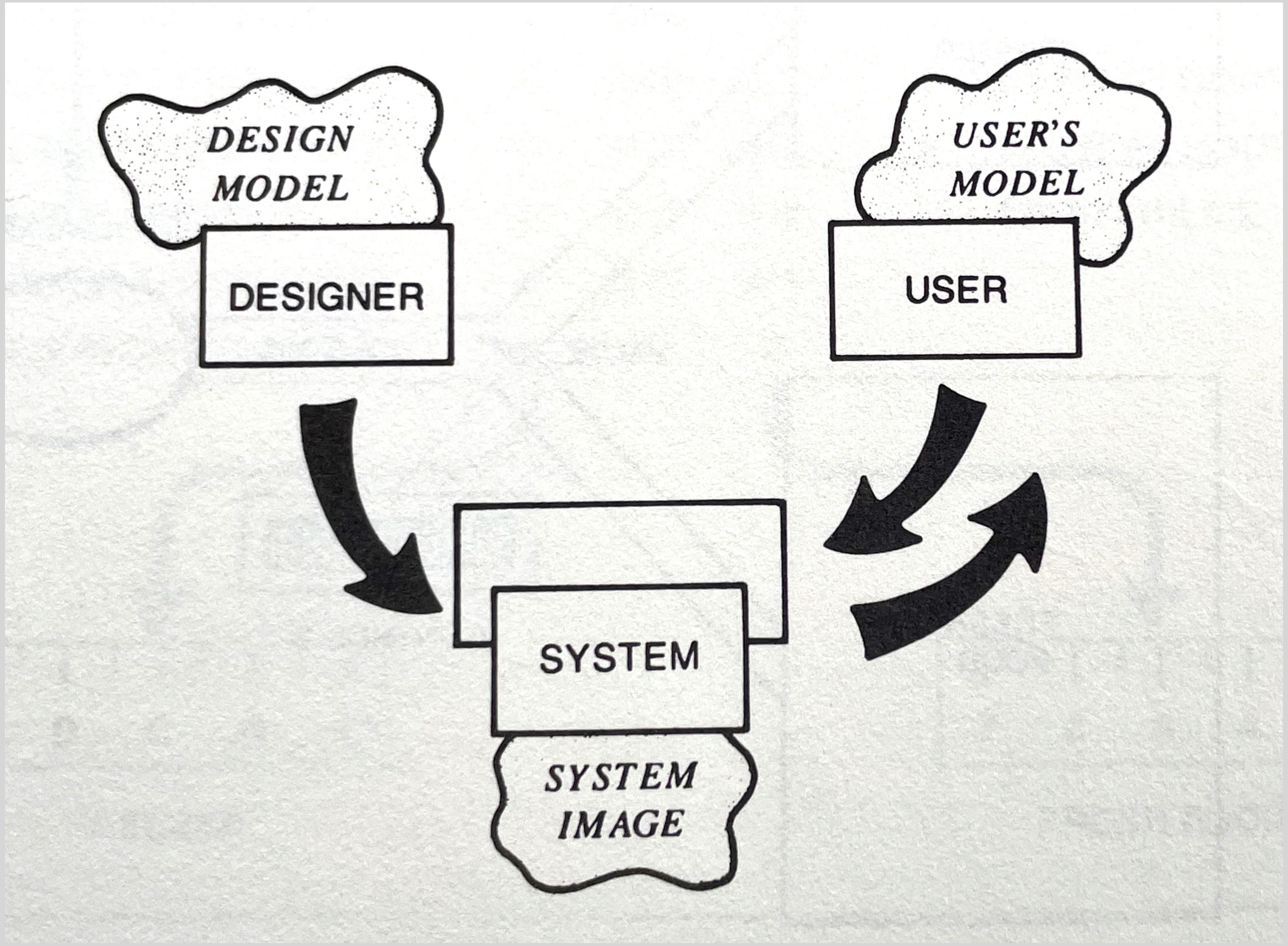
Although DOET covers numerous topics, three have come to stand out as critical:
1. It's not your fault...
2. Design principles... **conceptual models**, feedback, constraints, affordances
3. The power of observation

preface to 2002 edition

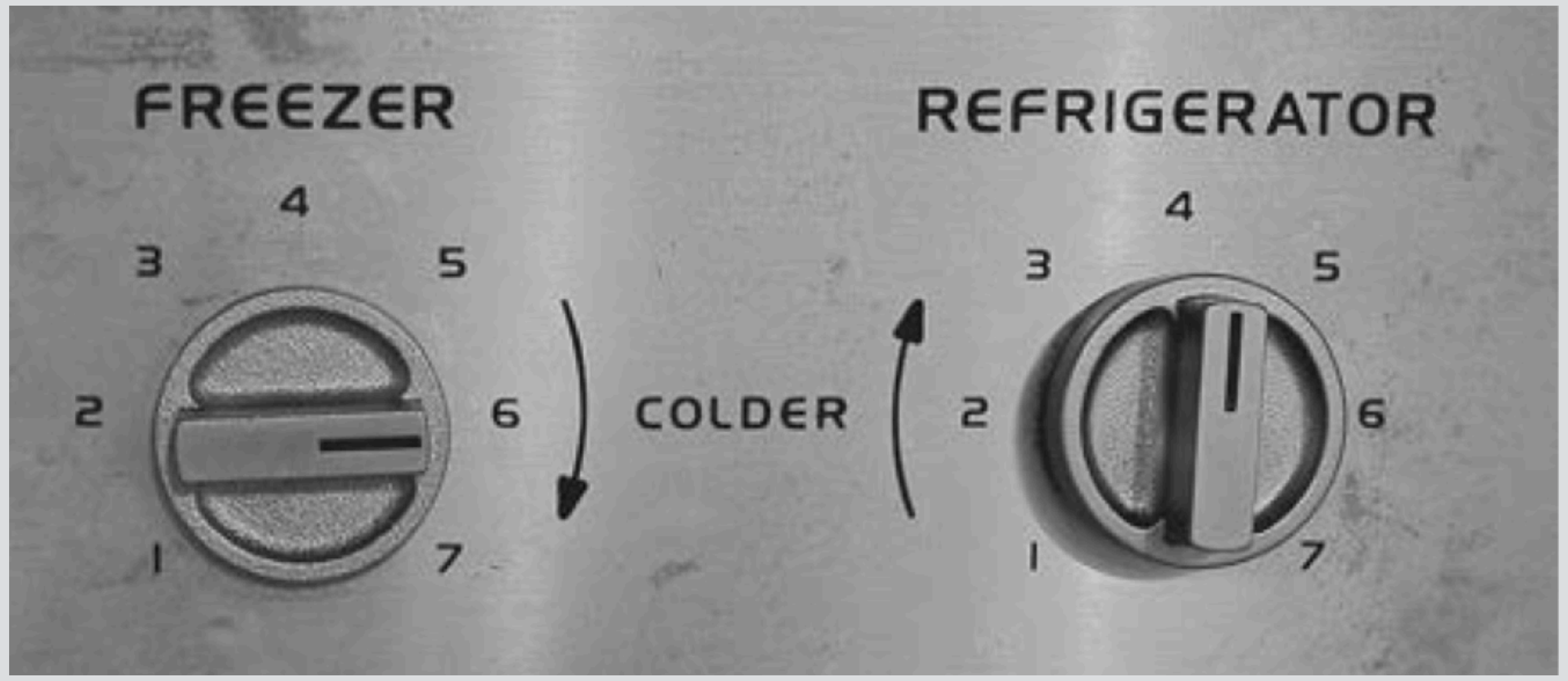
When the designers fail to provide a conceptual model, we will be forced to make up our own, and the ones we make up are apt to be wrong. **Conceptual models are critical to good design.**

preface to 2013 edition

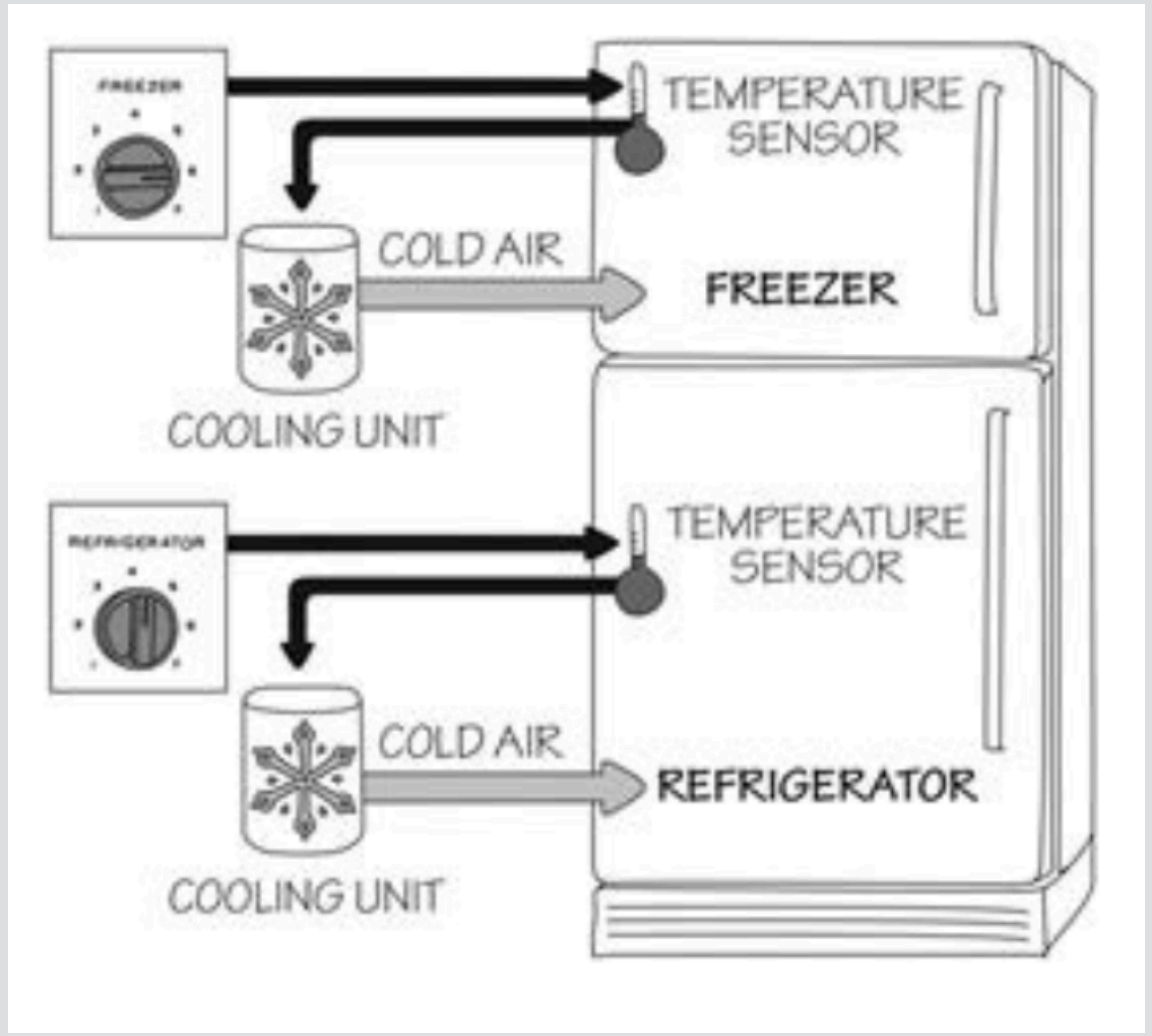
the “system image”



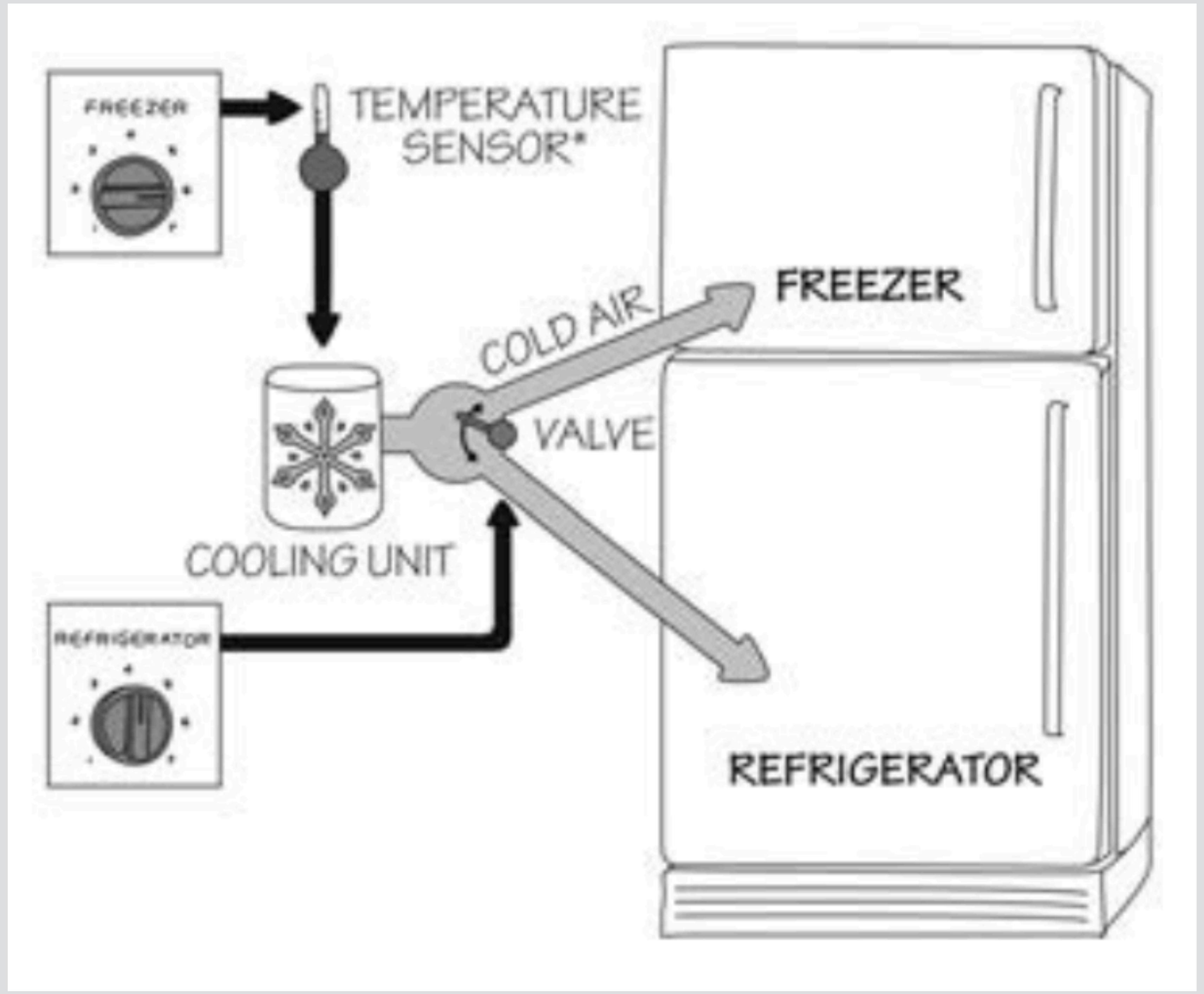
from *The Design of Everyday Things*



typical controls on American fridge



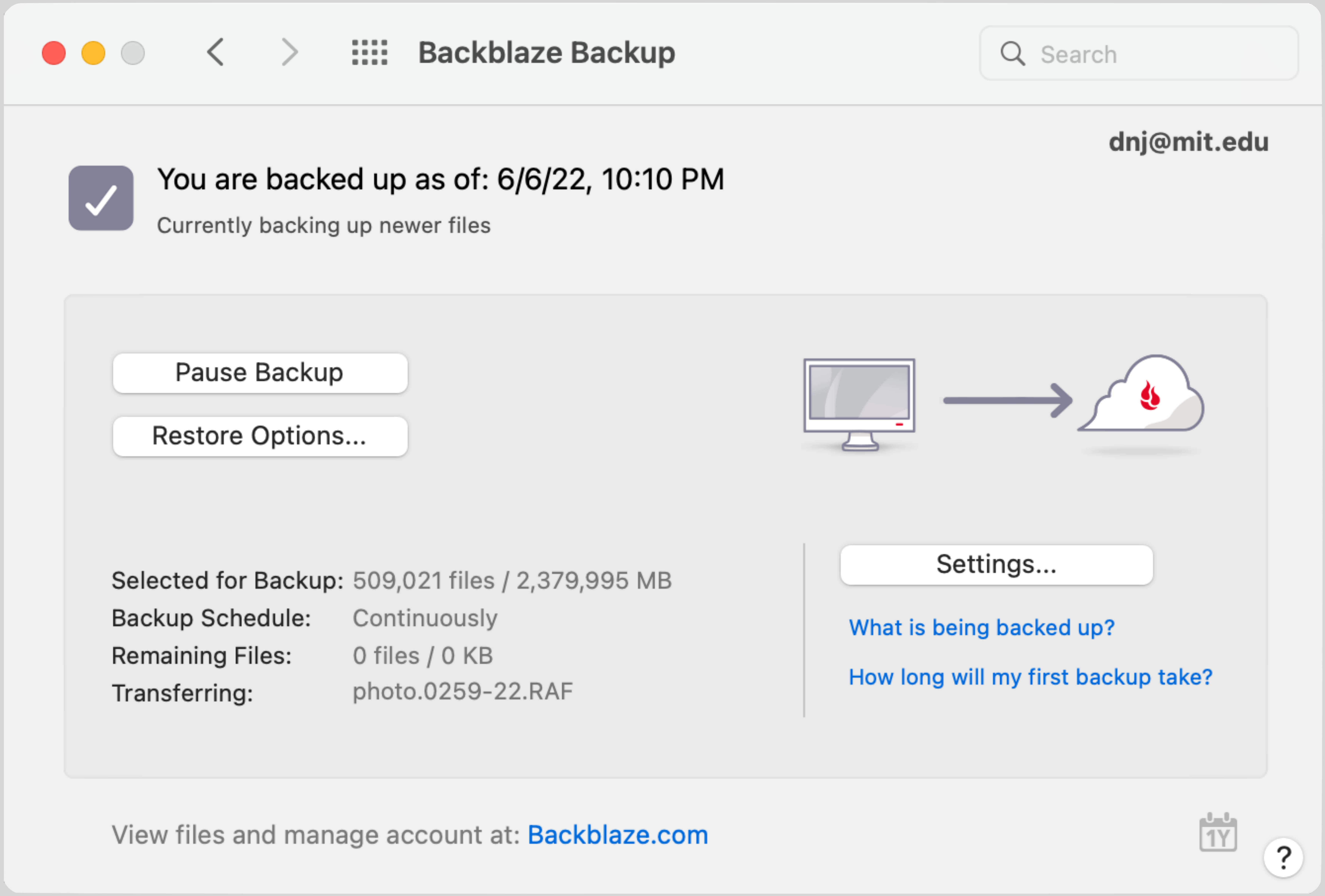
conceptual model (imagined)



conceptual model (actual)

from Don Norman, Design of Everyday Things (2013)

your turn: can you guess the correct conceptual model?



dnj@mit.edu

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

Currently backing up newer files

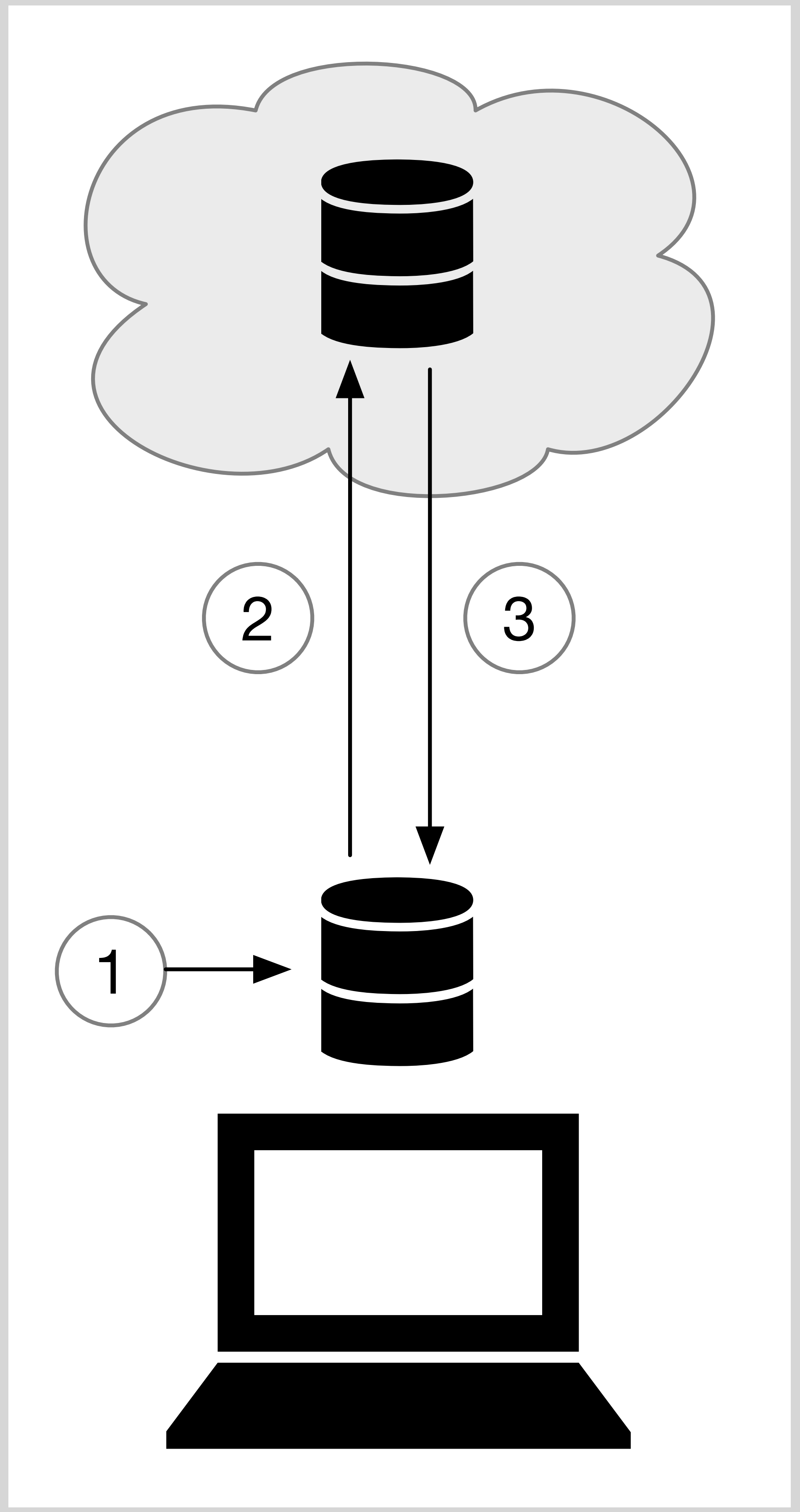
Selected for Backup:	509,021 files / 2,379,995 MB
Backup Schedule:	Continuously
Remaining Files:	0 files / 0 KB
Transferring:	photo.0259-22.RAF

What is being backed up?

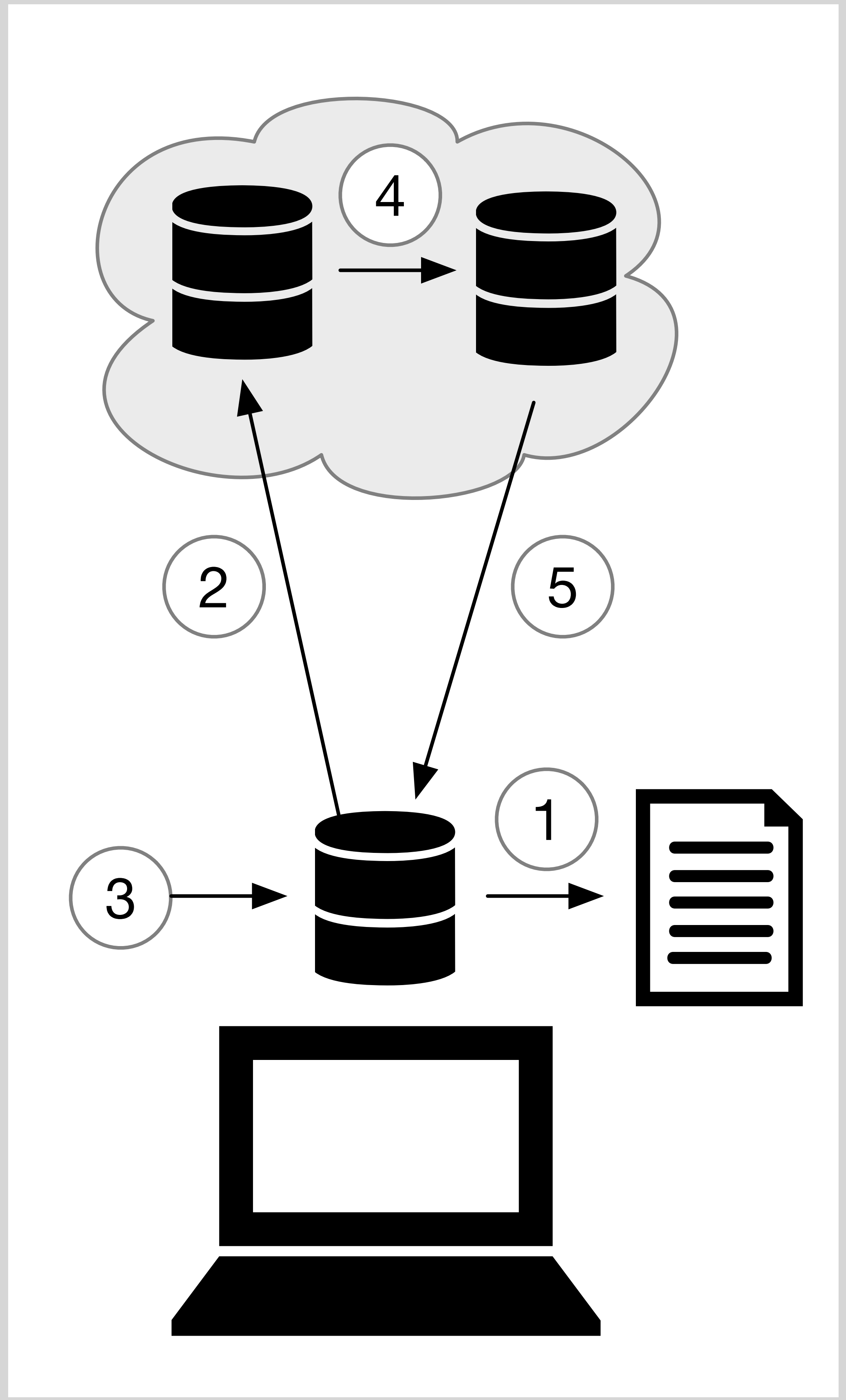
How long will my first backup take?

View files and manage account at: Backblaze.com

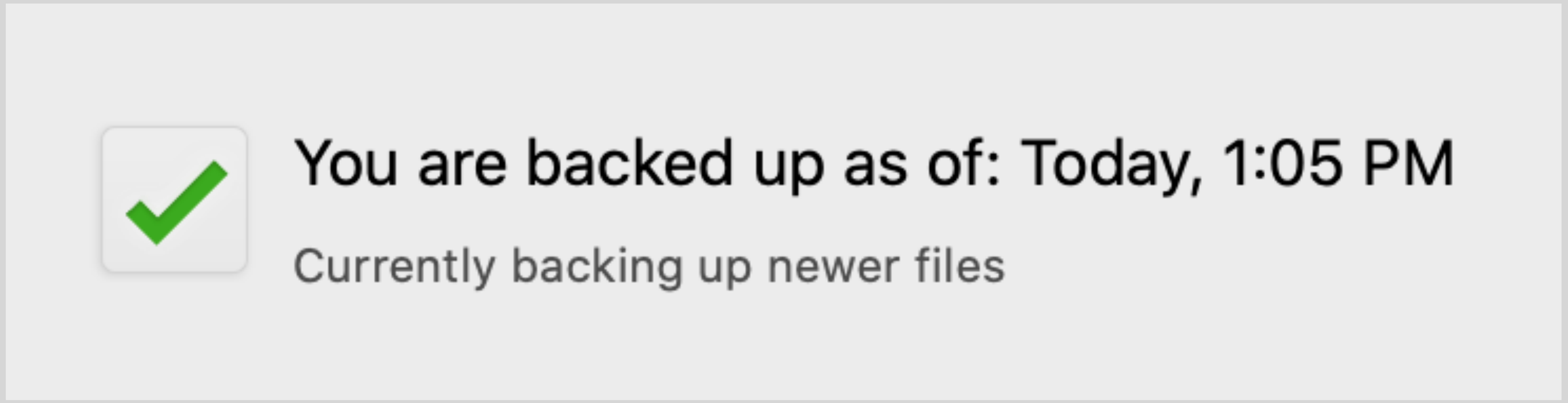
imagining backblaze's conceptual model



"continuous backup"
what I imagined



"continuous backup"
what actually happens

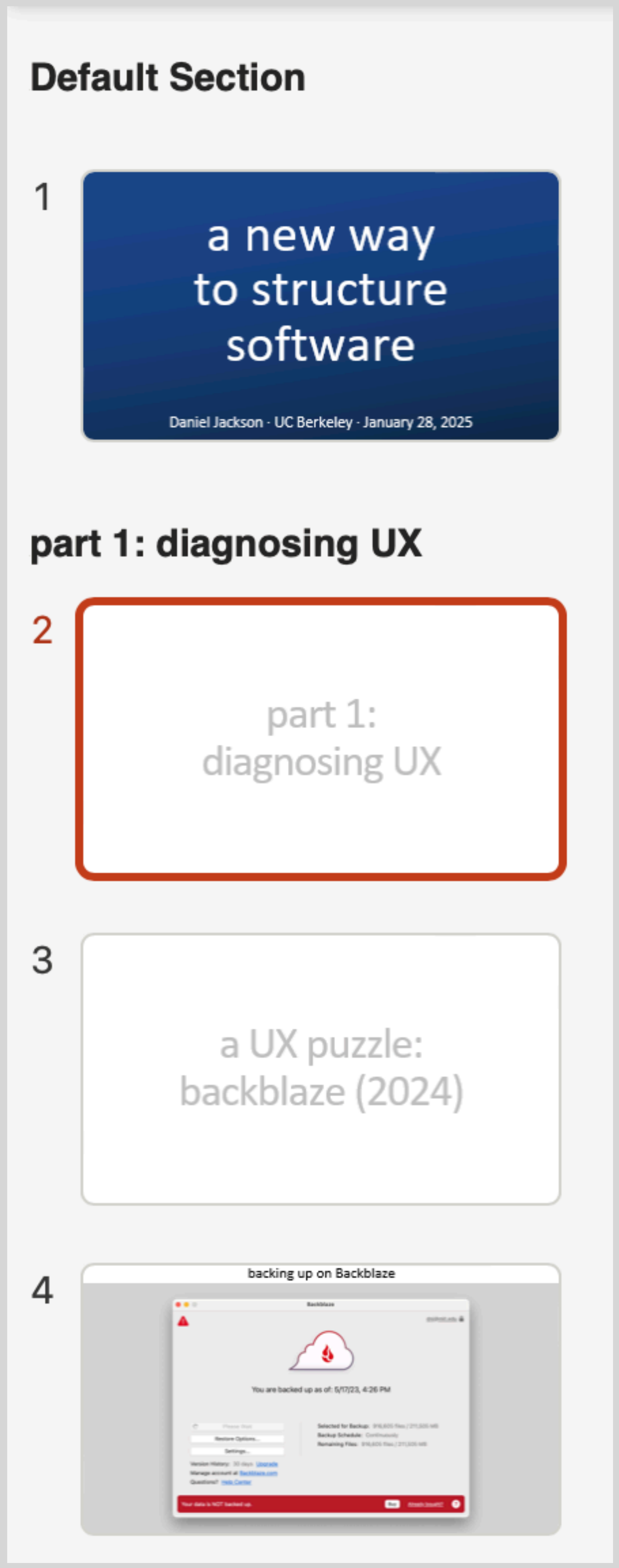


You are backed up as of: Today, 1:05 PM

Currently backing up newer files

a harder case: Powerpoint’s sections

powerpoint's section concept


Default Section

1 a new way to structure software

Daniel Jackson · UC Berkeley · January 28, 2025

part 1: diagnosing UX

2 part 1: diagnosing UX

3 a UX puzzle: backblaze (2024)

4 backing up on Backblaze

how to group slides into a section

1. select first slide to be in section & do add section
   (this will make a section from the selected slide to the end)
2. select slide after last slide to be in section & do add section
   (this will break the slides into two sections)

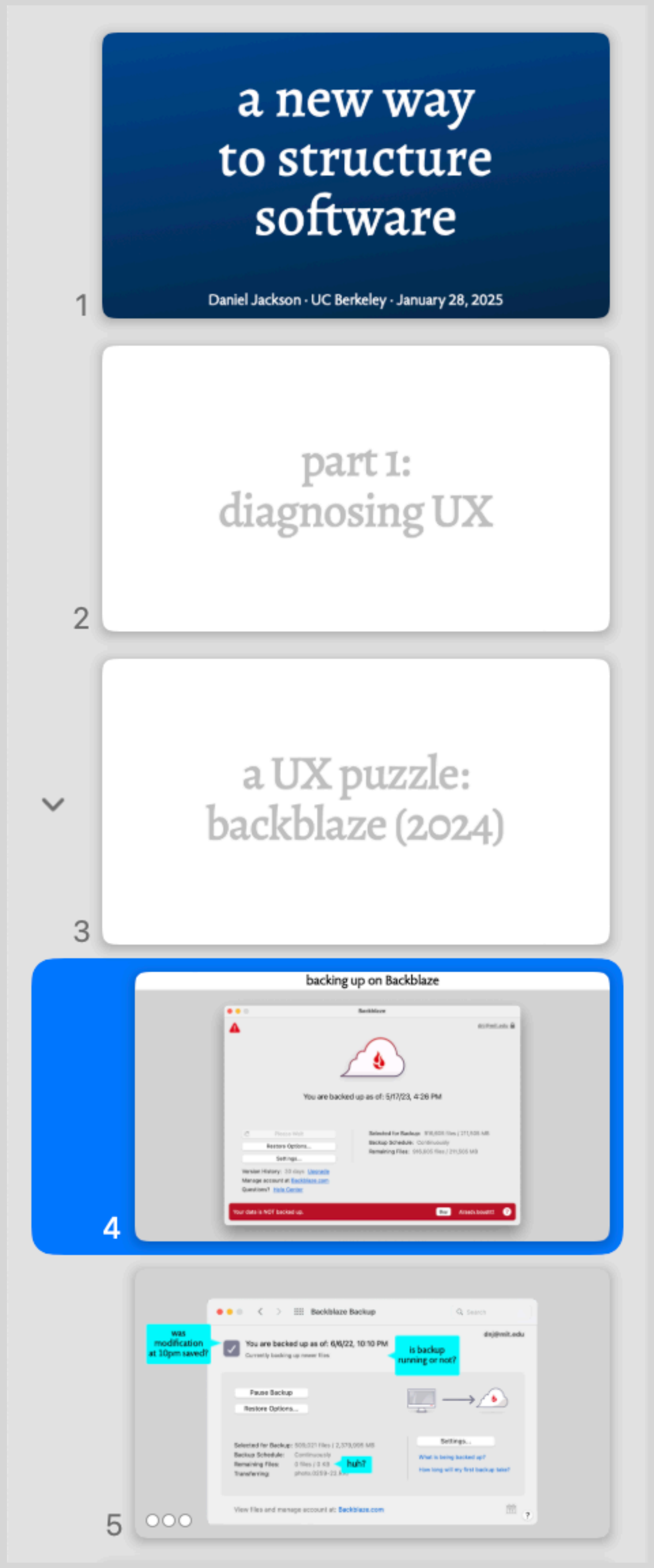
some anomalies

- when you add your first section, a default section is created, so you get two sections (unless you selected the first slide)
- you can't delete the default section (unless it's the only one)
- if you select multiple slides, add section works as if you'd selected the first (unless not contiguous, then not allowed)

missing functionality: you can't

- nest sections
- hide a section (except in slide sorter)
- move a section more than one step (except in slide sorter)

keynote's tree outline concept


1
a new way to structure software
Daniel Jackson · UC Berkeley · January 28, 2025
2
part 1: diagnosing UX
3
a UX puzzle: backblaze (2024)
4
backing up on Backblaze
5

how to group slides under a header

1. select all except a header slide, and drag to right

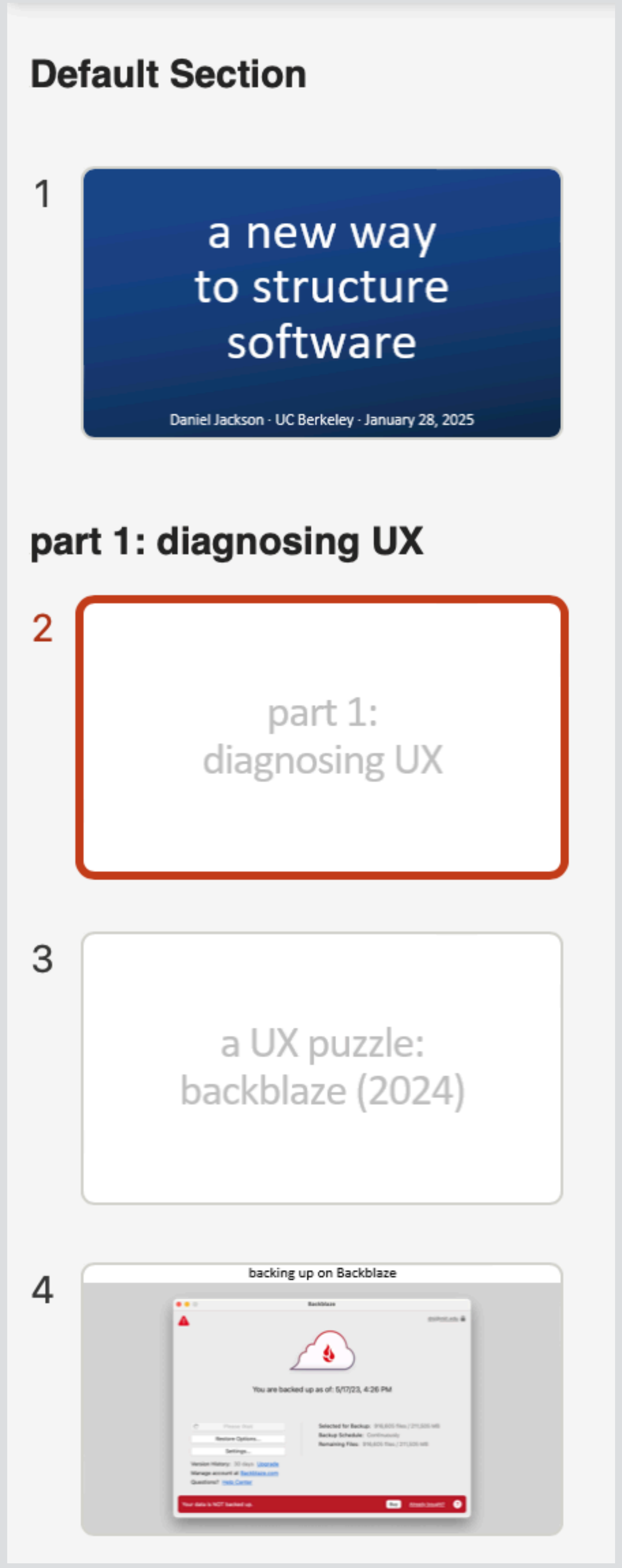
some anomalies

- none

missing functionality: you can't

- group slides without a header
  (but you can mark header as skipped)

your turn: which is better and why?



Default Section

1 a new way to structure software

Daniel Jackson · UC Berkeley · January 28, 2025

part 1: diagnosing UX

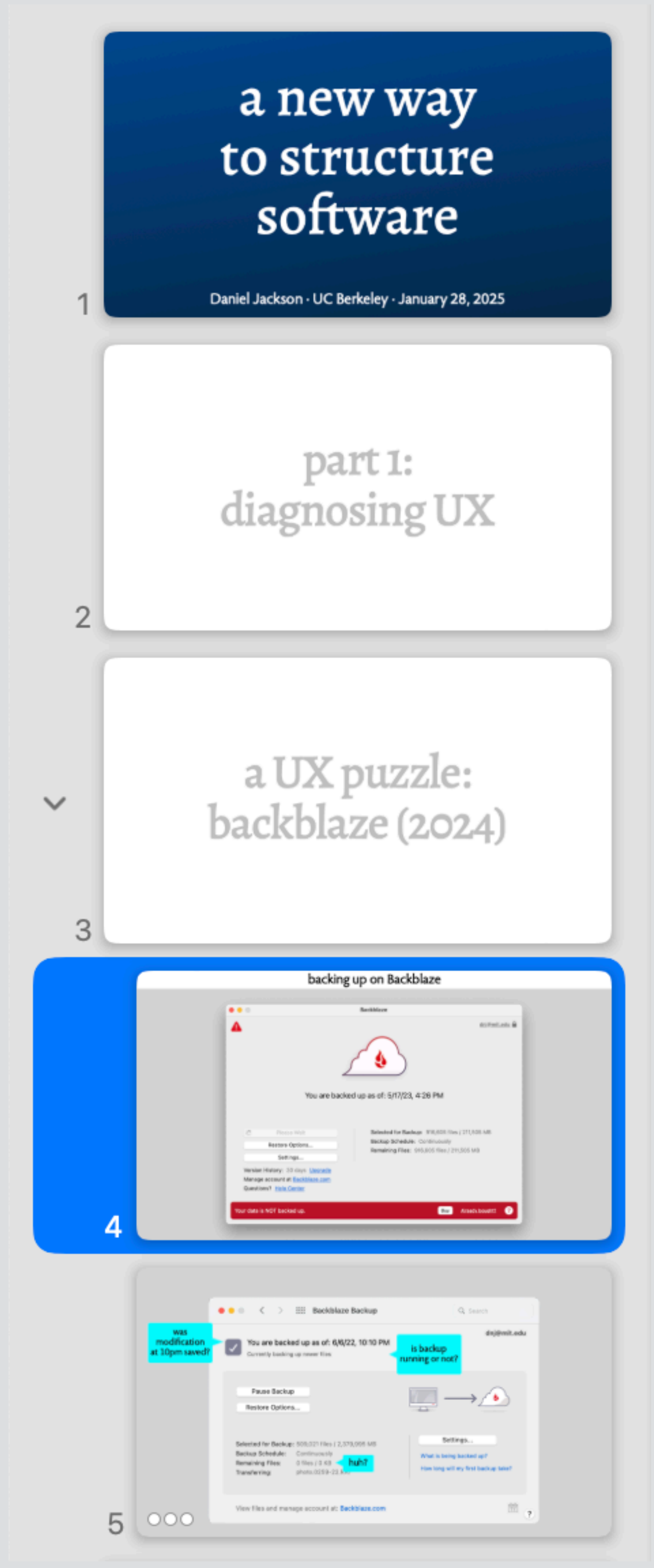
2 part 1: diagnosing UX

3 a UX puzzle: backblaze (2024)

4 backing up on Backblaze

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

Powerpoint



1 a new way to structure software

Daniel Jackson · UC Berkeley · January 28, 2025

2 part 1: diagnosing UX

3 a UX puzzle: backblaze (2024)

4 backing up on Backblaze

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

5 Backblaze Backup

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

was modification at 10pm saved?

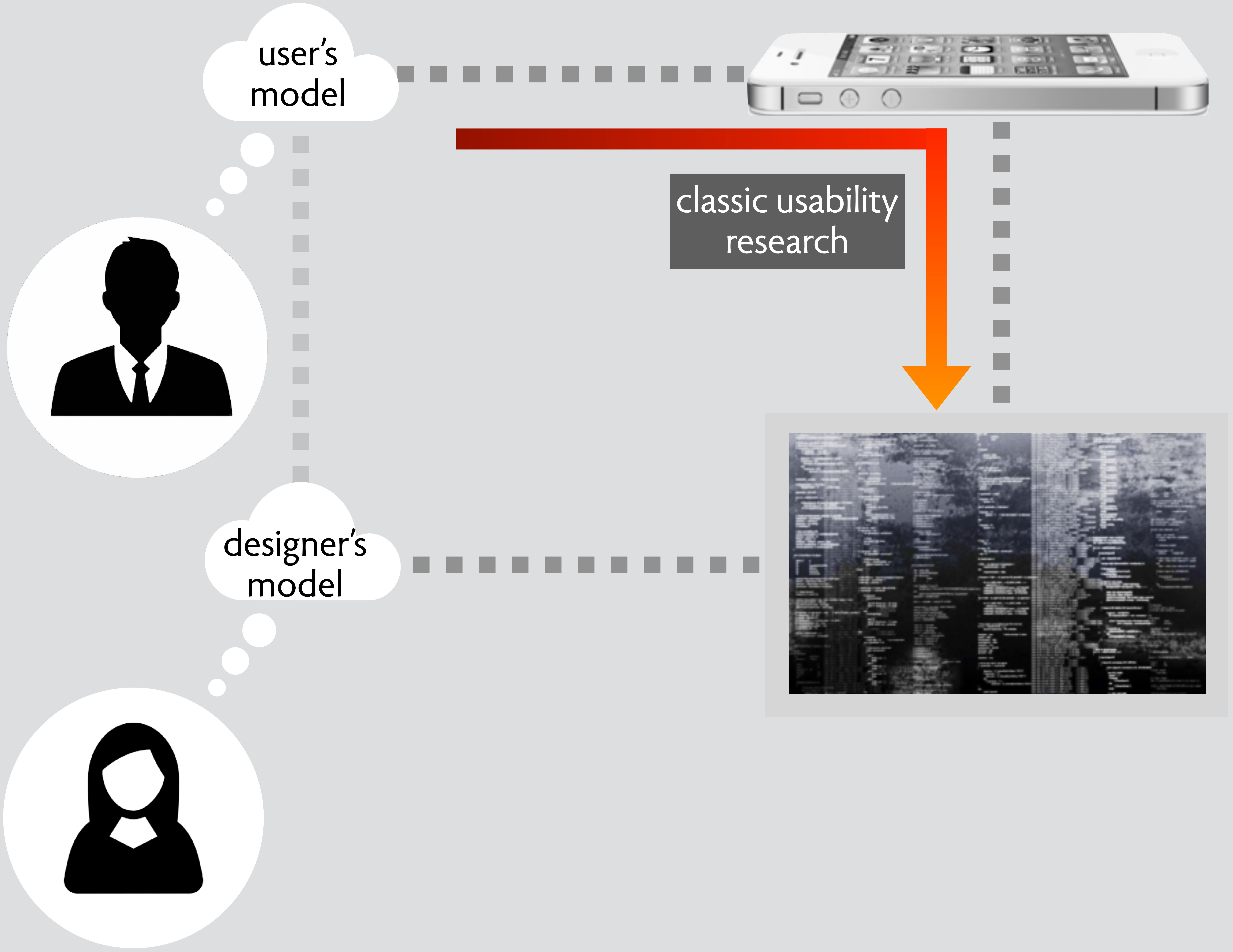
is backup running or not?

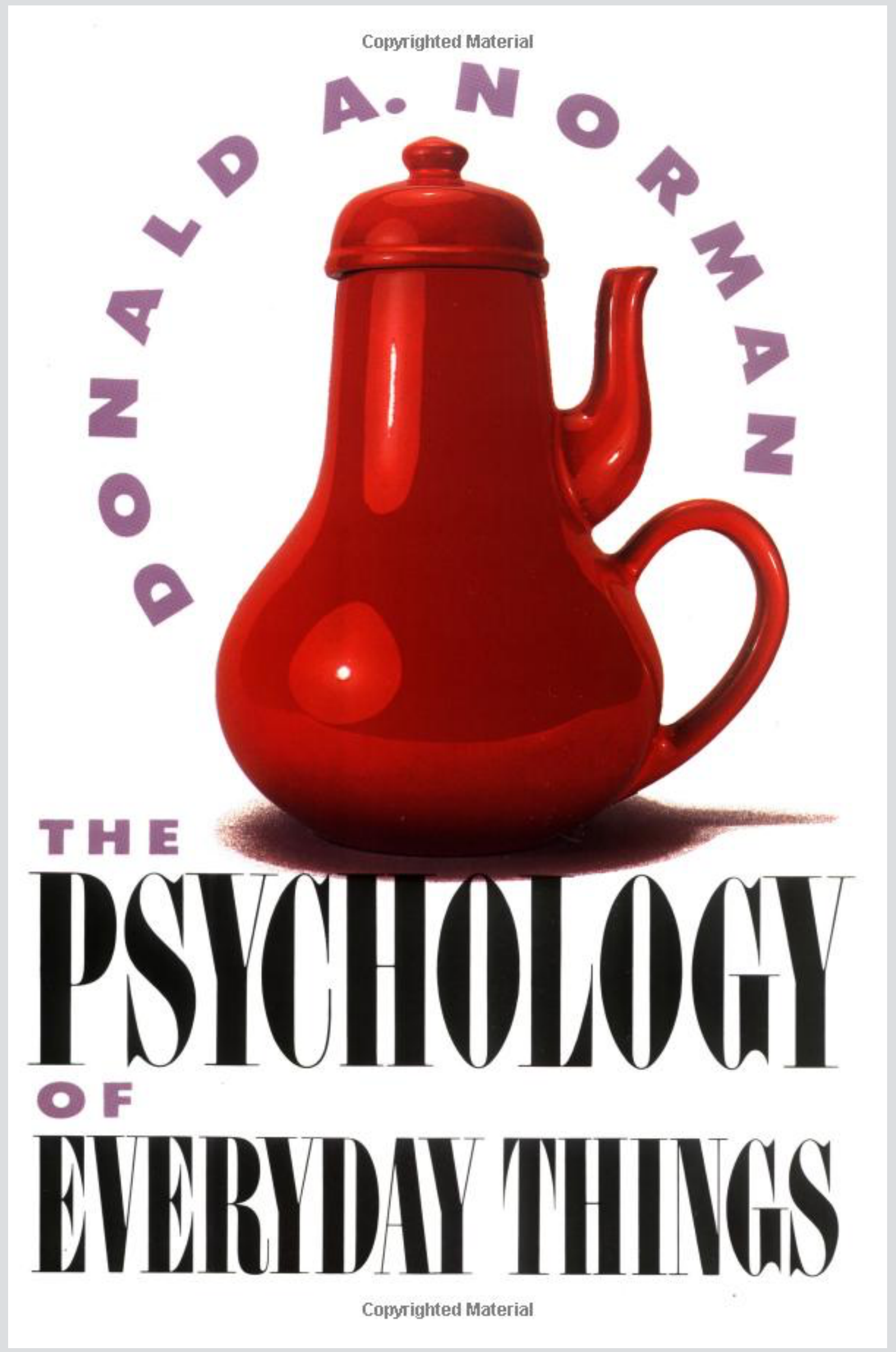
huh?

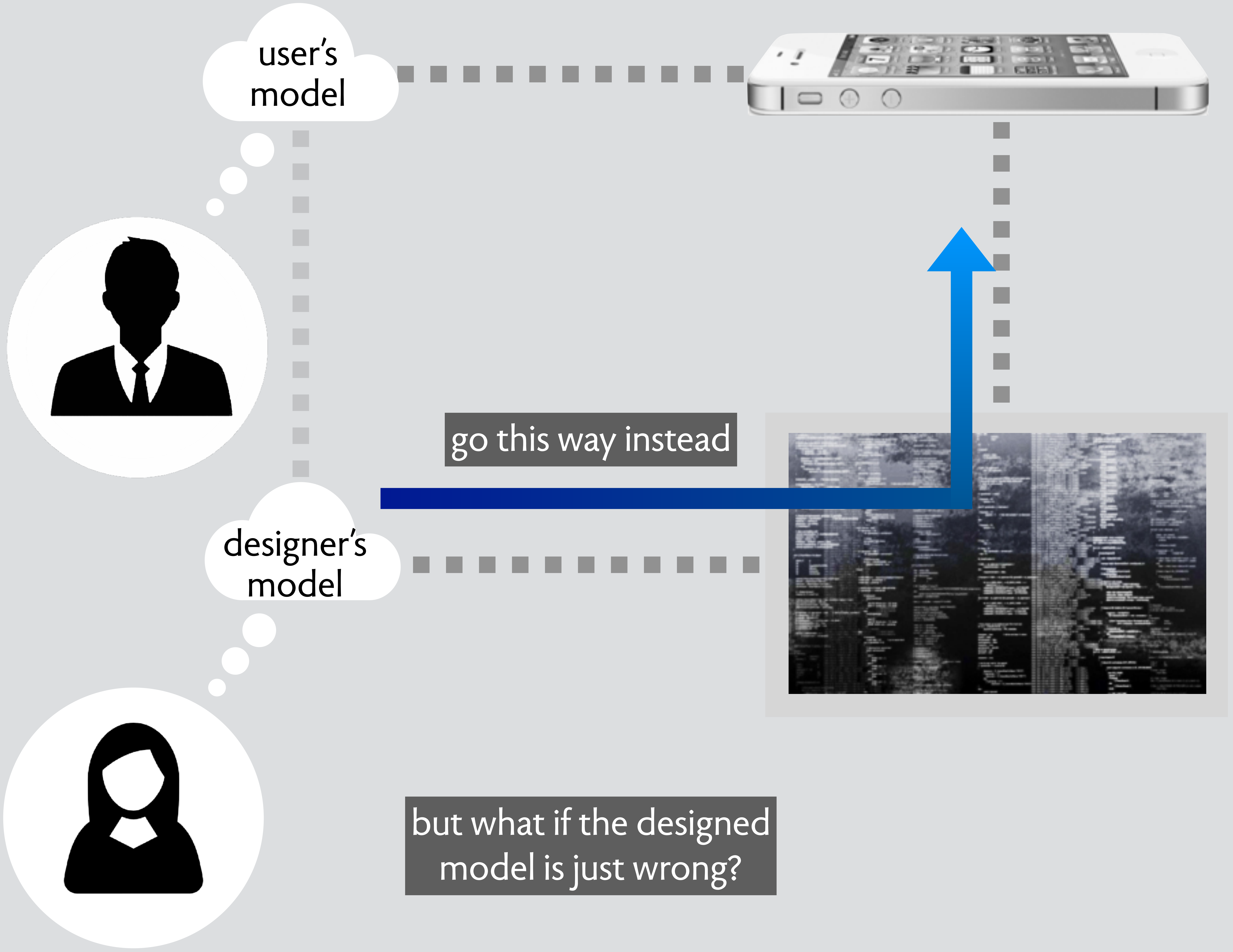
Keynote

are there general lessons here?
principles of usability?
design criteria?
design strategies?

revisiting conceptual models




Copyrighted Material
DONALD A. NORMAN
THE PSYCHOLOGY OF EVERYDAY THINGS
Copyrighted Material



but what if the designed model is just wrong?

what exactly is wrong?

Perhaps the designers thought the correct model was too complex, that the model they were giving was easier to understand. But with the wrong conceptual model, it is impossible to set the controls. And even though I am convinced I now know the correct model, I still cannot accurately adjust the temperatures because of refrigerator design makes it impossible for me to discover which control is for the thermostat, which controls for the relative proportion of cold air, and in which compartment the thermostat is located. The lack of immediate feedback for the actions does not help: with the delay of 24 hours, who can remember what was tried?

Design of Everyday Things, p. 17

what’s missing

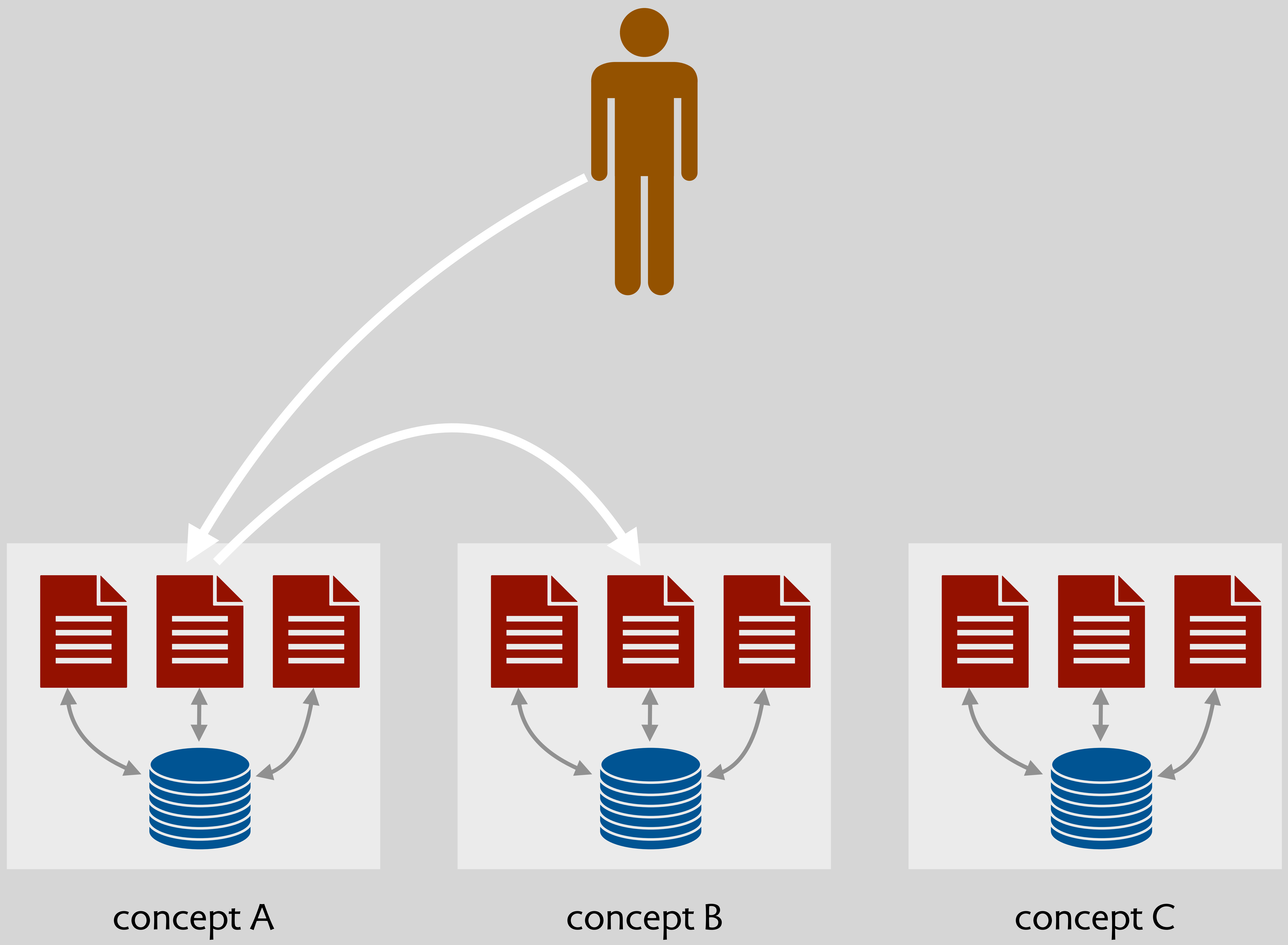
the conceptual model itself
unless it’s explicit, how can we know if we mapped it right?

design criteria for conceptual models
what makes a good model?

structuring the conceptual model
can we break the model into smaller parts? reusable concepts?

defining concepts

viewing a system in terms of concepts



a file store concept

concept FileStore [Name, Content]

purpose store files persistently

principle after creating and updating a file, you can get the content

state
a set of files
for each file
  name, contents

actions
create (n: Name, c: Content)
update (n: Name, c: Content)
delete (n: Name)
get (n: Name): Content

what's a name?
could be a pathname
allows hierarchy and sidesteps complexity of folders
no possibility of two parents (as in Unix)
but also no empty folders!

changing names?
can a file's name be changed with identity remaining?
then could say "this file's name was changed" (cf. Git)

a backup concept

concept Backup [Name, Content]

purpose retrieve old version of files

principle after a file's contents are saved, they can be retrieved later by date

state
a set of files with versions
for each file
  name, contents, date

actions
save (n: Name, c: Content)
restore (n: Name, d: Date): Content

are files mutable?
no, because no action to change

can empty folders be stored?
no, because no content to save

can files be deleted?
no, but Backblaze isn't like this

a workset concept

concept Workset [Item]

purpose process items in batches

principle after items are added, and processing is started, the items are processed

state
current set of items being worked on
next set of items to work on

actions

```
start ()
  requires current == {}
  current = next
  next = {}
add (i: Item)
  next = next + i
process (i: Item)
  requires i in current
  current = current - i
```

when do the actions happen?

concept FileStore [Name, Content]

purpose store files persistently

principle after creating and updating a file, you can get the content

state
a set of files
for each file
  name, contents

actions
create (n: Name, c: Content)
update (n: Name, c: Content)
delete (n: Name)
get (n: Name): Content

concept Backup [Name, Content]

purpose retrieve old version of files

principle after a file's contents are saved, they can be retrieved later by date

state
a set of files with versions
for each file
  name, contents, date

actions
save (n: Name, c: Content)
restore (n: Name, d: Date): Content

concept Workset [Item]

purpose process items in batches

principle after items are added, and processing is started, the items are processed

state
current set of items being worked on
next set of items to work on

actions
```
start ()
  requires current == {}
  current = next
  next = {}
add (i: Item)
  next = next + i
process (i: Item)
  requires i in current
  current = current - i
```

summary: a backup system in 3 concepts

a separation of concerns
a division of labor

familiar mechanisms
we've seen these before

reusable elements
designs with a subset of these

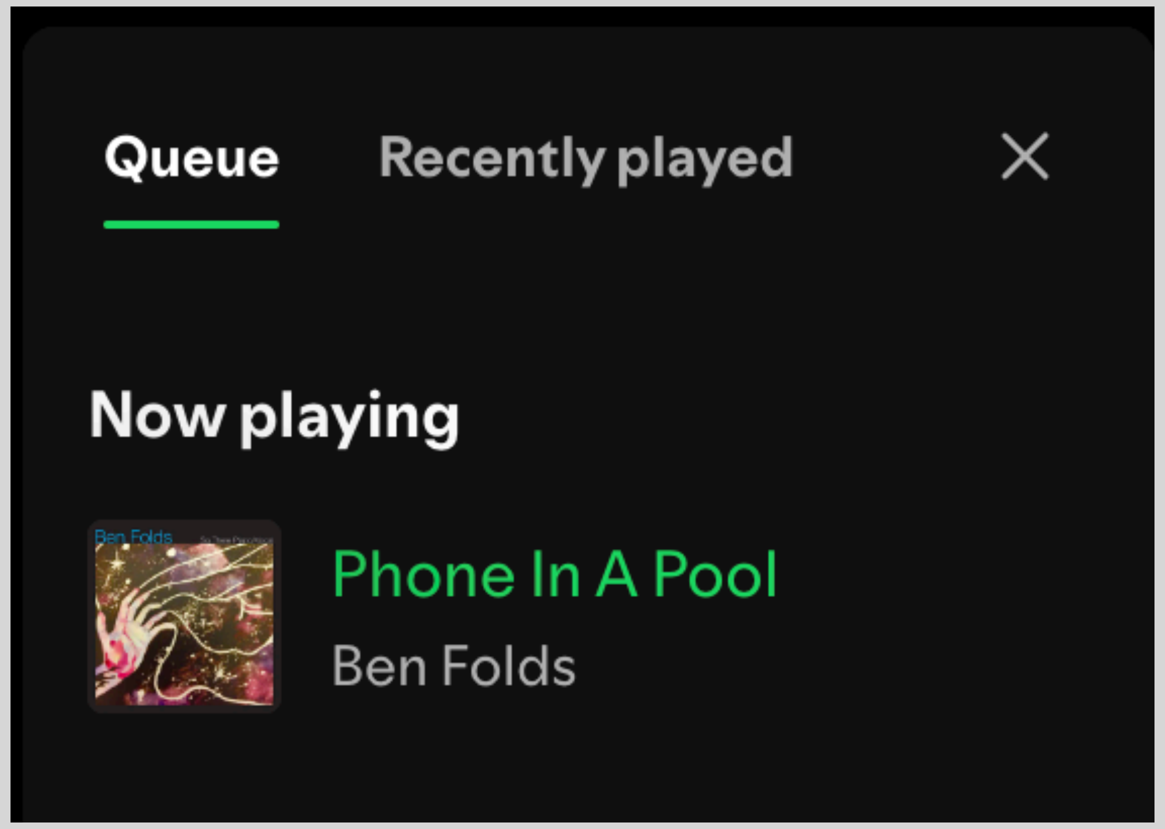
not just the concepts
where do restored files go?

concept FileStore	storing and updating content
concept Backup	saving and restoring versions
concept Workset	processing items one at a time

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

your turn: conceptual model for Spotify

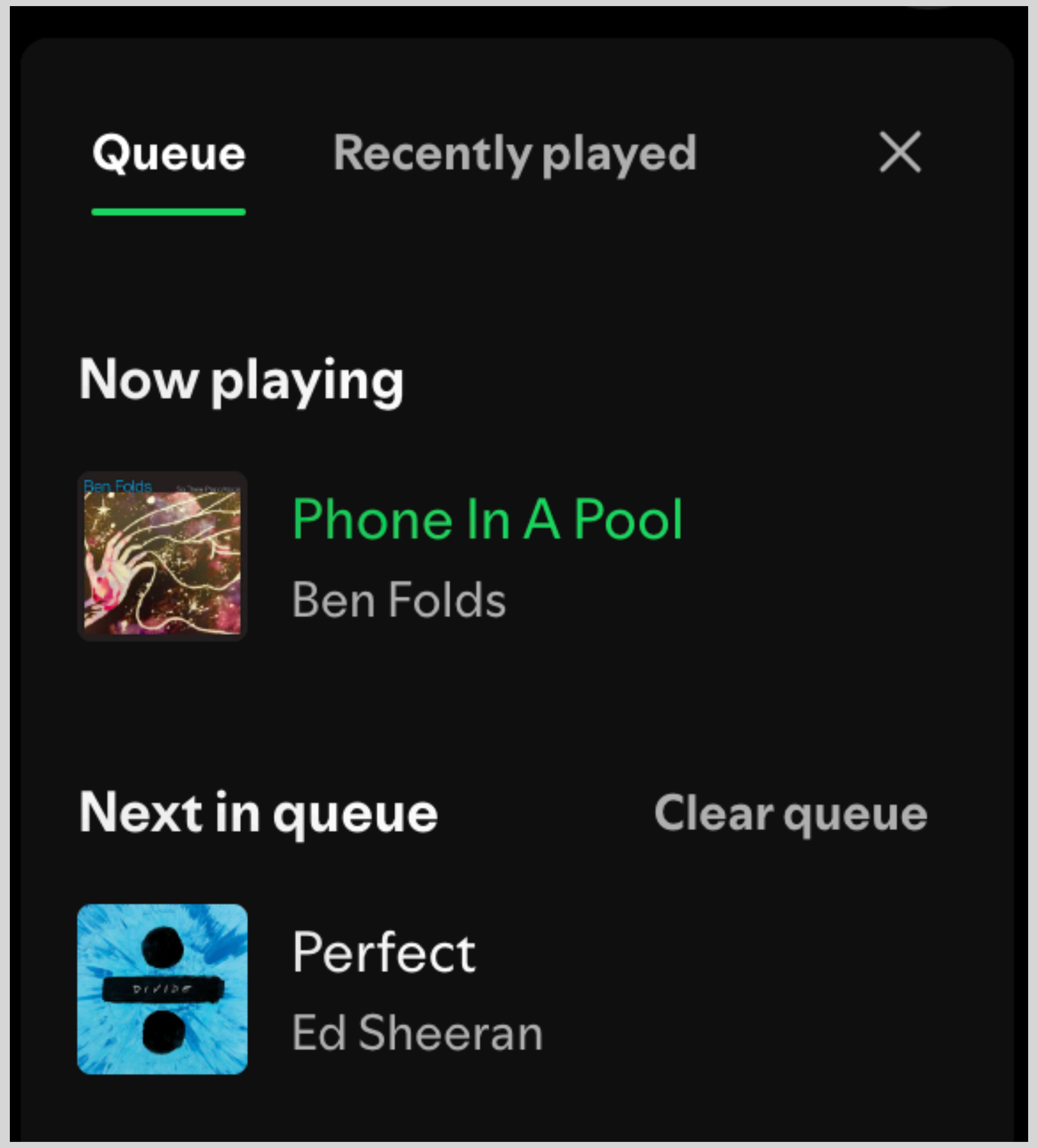
what's going on? what are the concepts and how do they work?


Queue Recently played

Now playing

Phone In A Pool
Ben Folds

start a song playing
open the queue


Queue Recently played

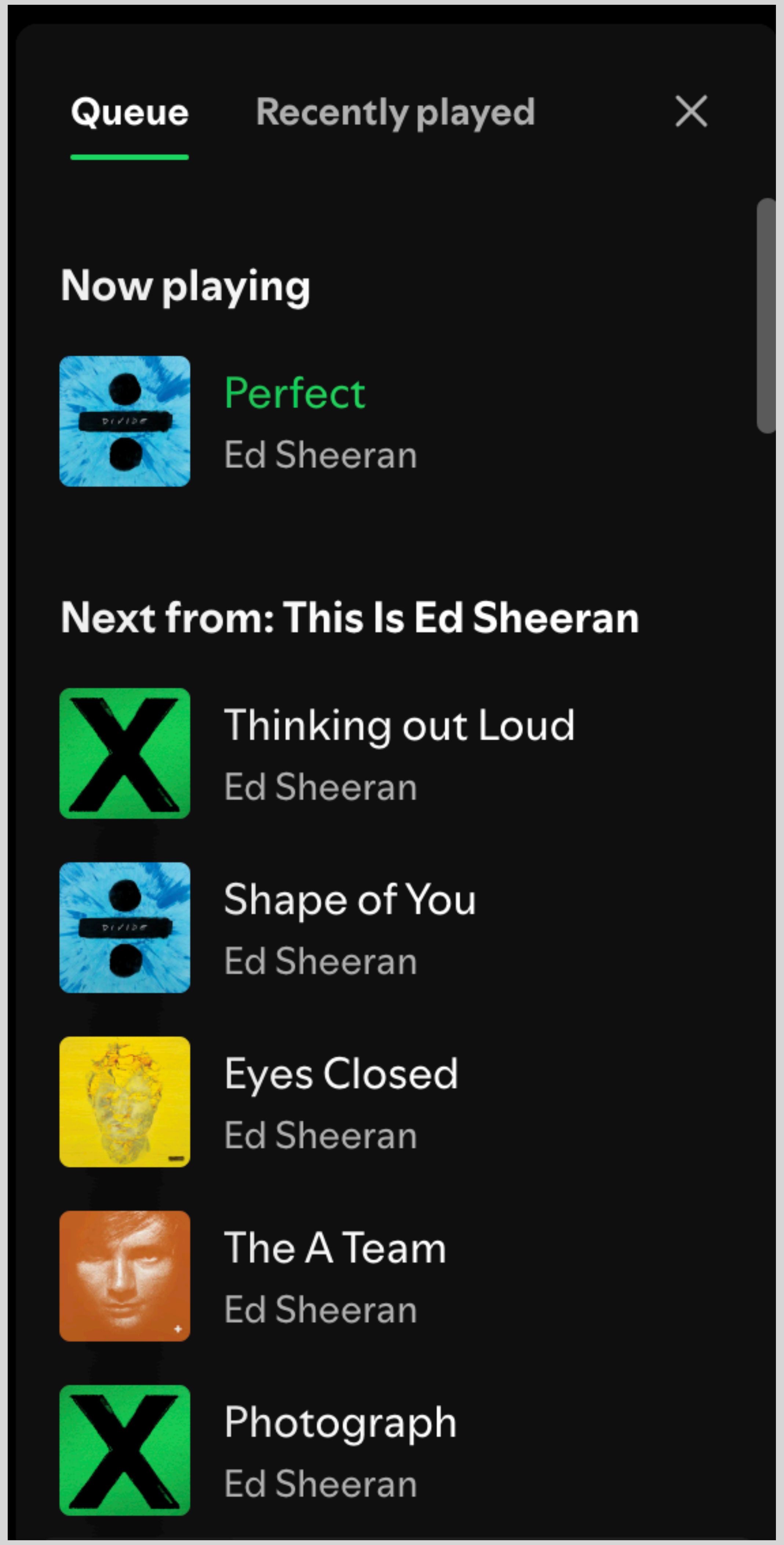
Now playing

Phone In A Pool
Ben Folds

Next in queue Clear queue

Perfect
Ed Sheeran

add another song
to the queue


Queue Recently played

Now playing

Perfect
Ed Sheeran

Next from: This Is Ed Sheeran

Thinking out Loud
Ed Sheeran

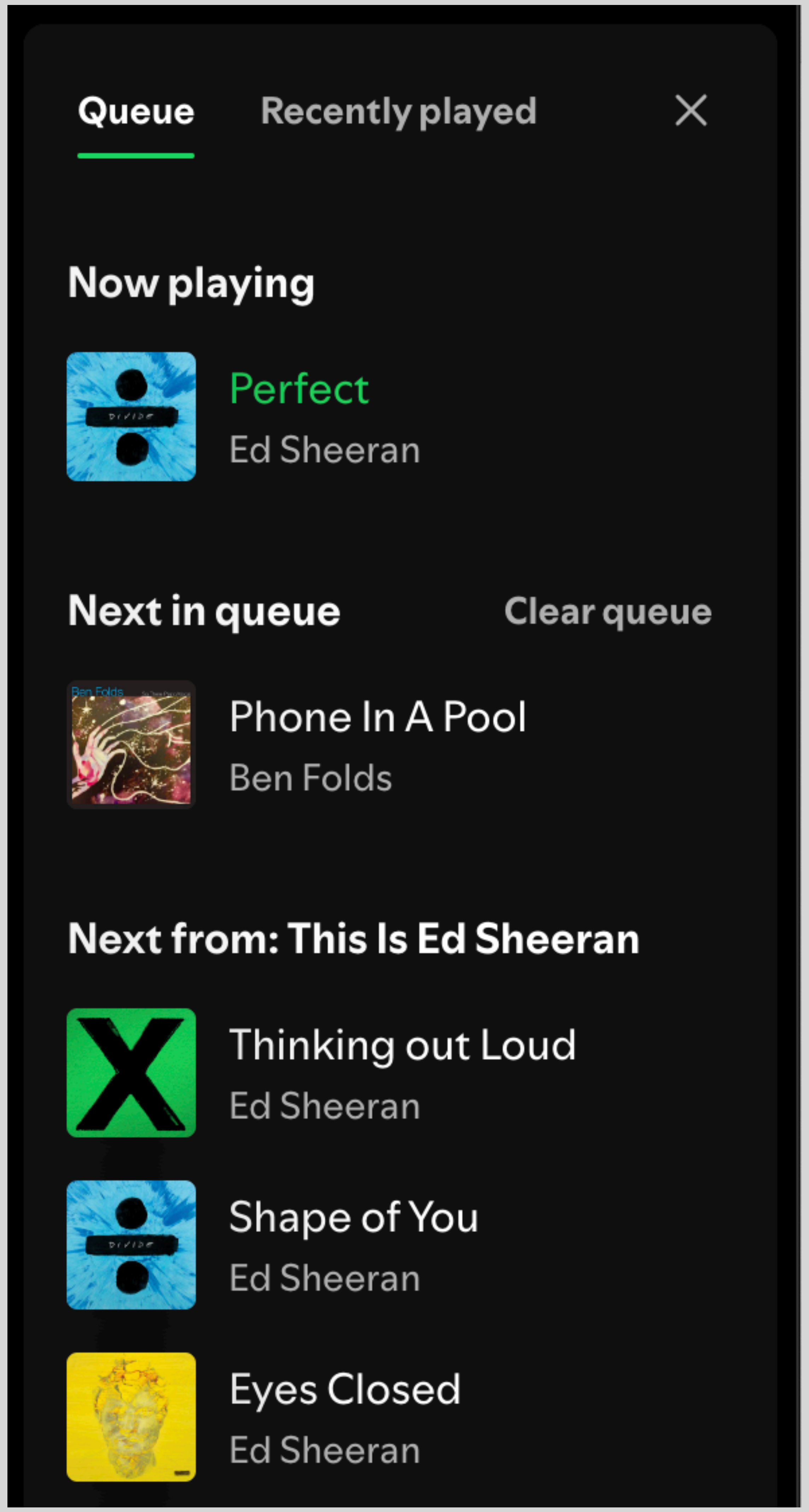
Shape of You
Ed Sheeran

Eyes Closed
Ed Sheeran

The A Team
Ed Sheeran

Photograph
Ed Sheeran

start a song playing
in a playlist


Queue Recently played

Now playing

Perfect
Ed Sheeran

Next in queue Clear queue

Phone In A Pool
Ben Folds

Next from: This Is Ed Sheeran

Thinking out Loud
Ed Sheeran

Shape of You
Ed Sheeran

Eyes Closed
Ed Sheeran

add another song
to the queue

the benefits concepts bring

initial motivations

better UX
clarity & power

modularity
in design & code

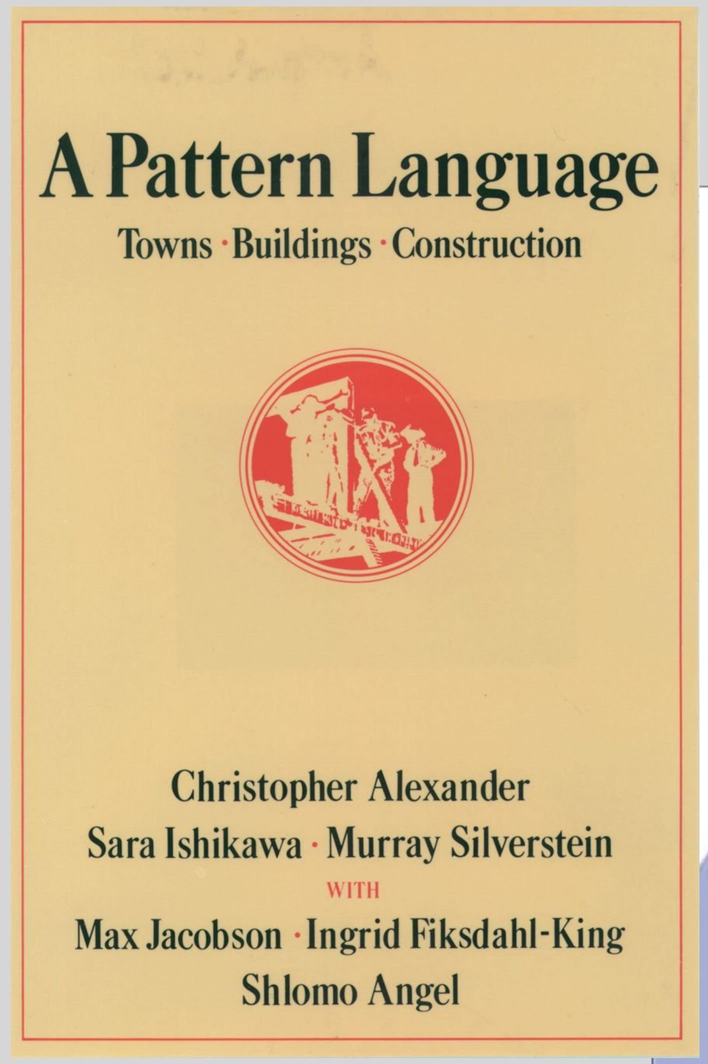
a design language
bridging roles too

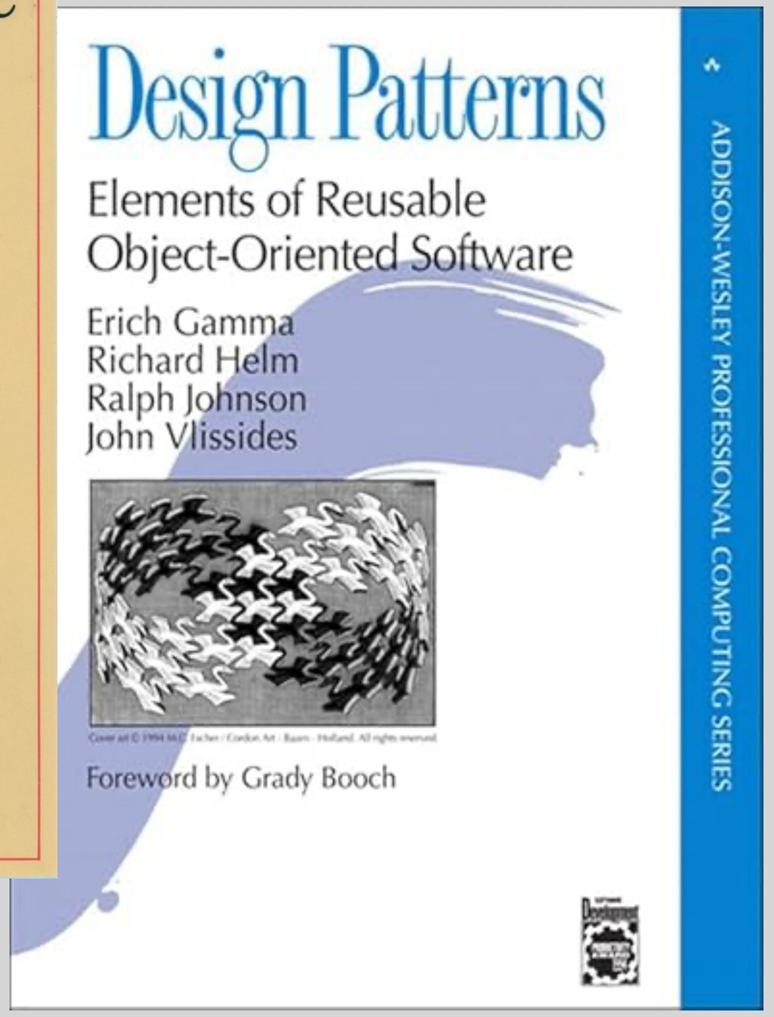
a place for design
concept-specific issues

what may matter more

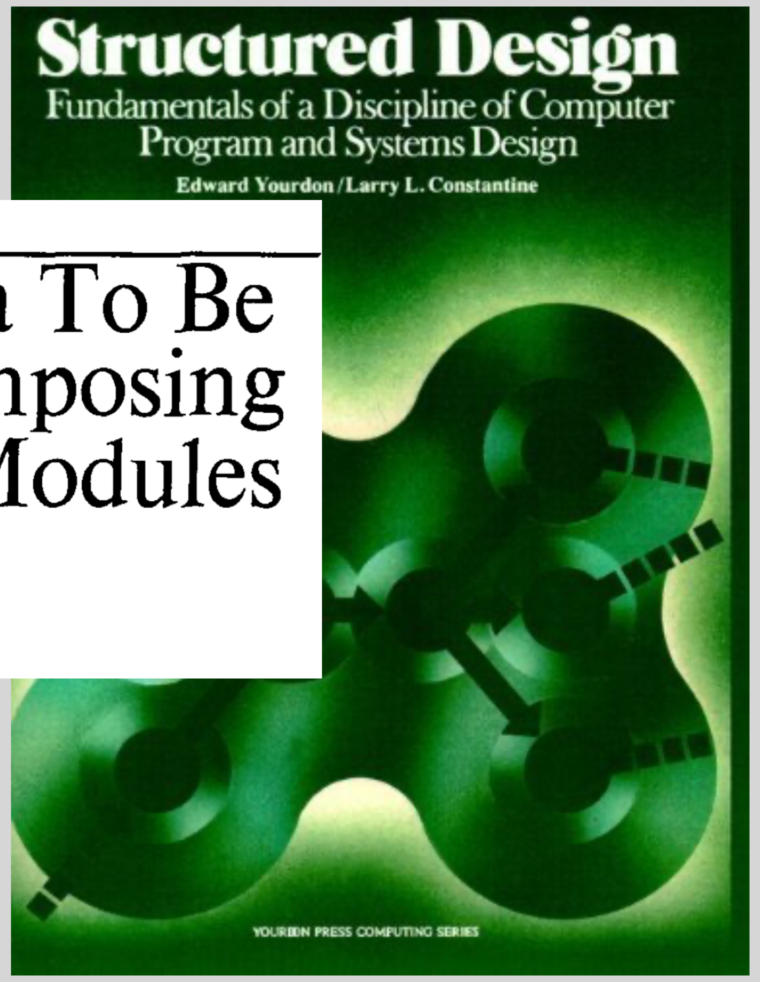
where concept
design comes from

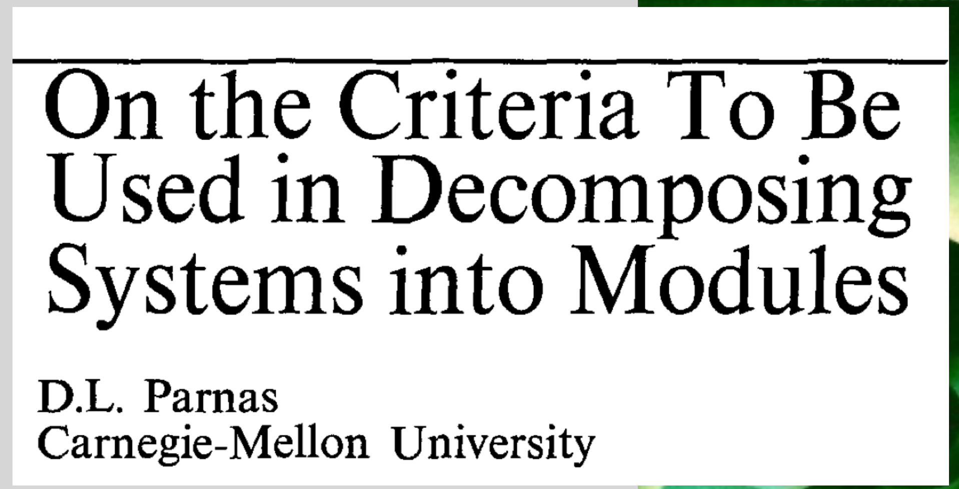
where the ideas of concept design came from


A Pattern Language
Towns · Buildings · Construction
Christopher Alexander
Sara Ishikawa · Murray Silverstein
WITH
Max Jacobson · Ingrid Fiksdahl-King
Shlomo Angel


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

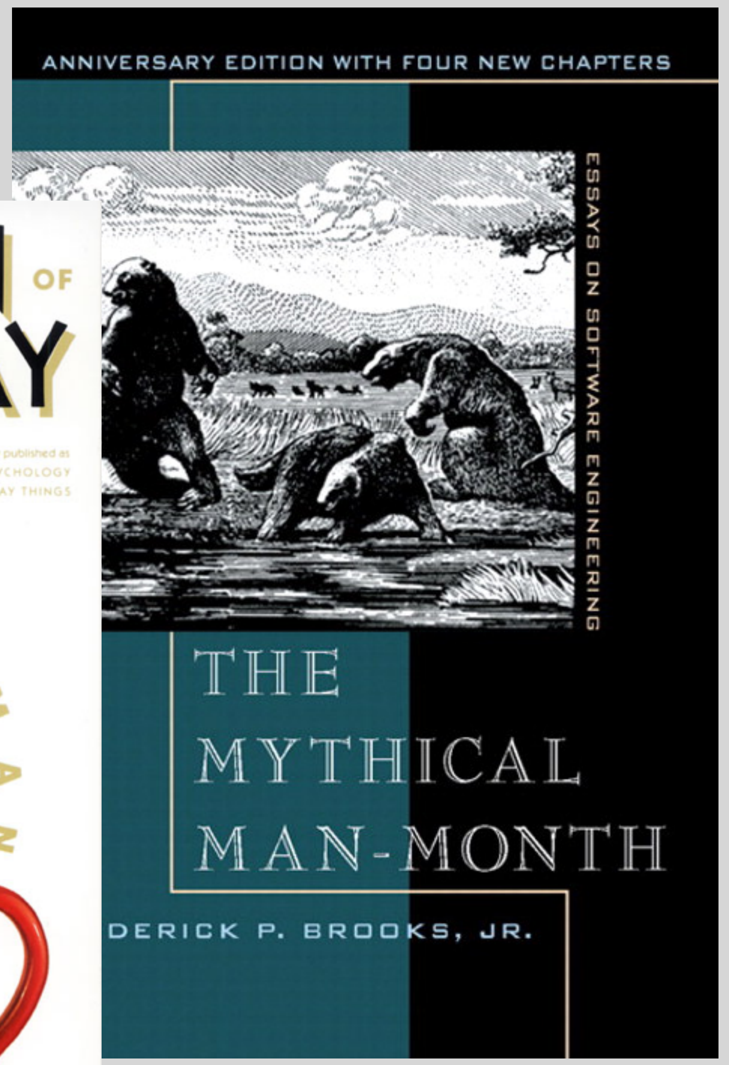
Christopher Alexander's **patterns**
popularized in software by GoF
source of DDD's ubiquitous language?


Structured Design
Fundamentals of a Discipline of Computer Program and Systems Design
Edward Yourdon / Larry L. Constantine
YOURDON PRESS COMPUTING SERIES

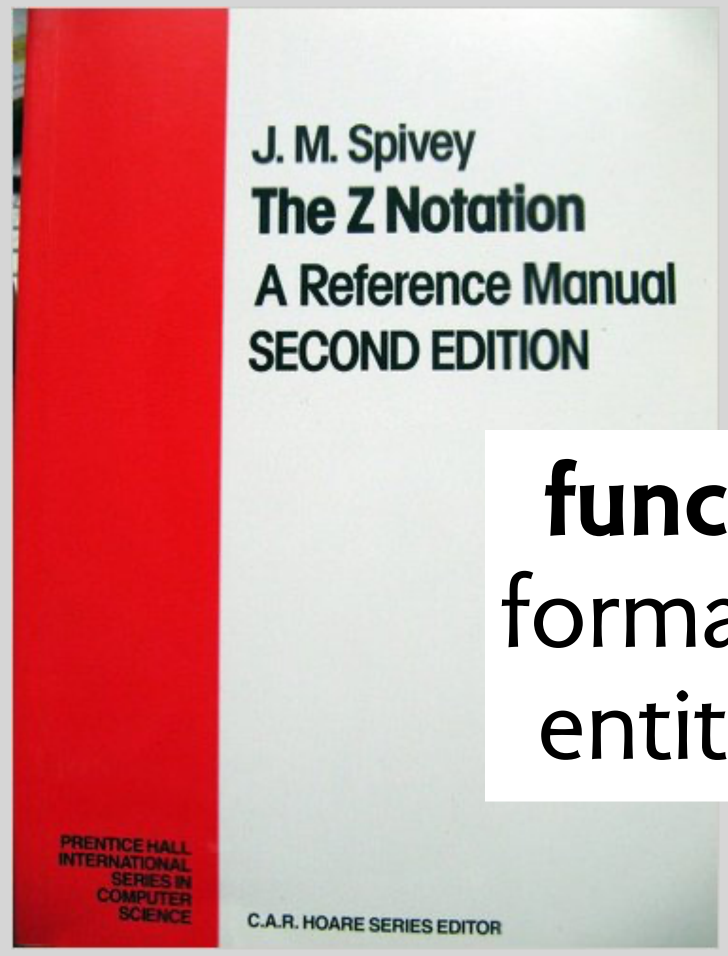

On the Criteria To Be Used in Decomposing Systems into Modules

D.L. Parnas
Carnegie-Mellon University

modularity & encapsulation
Parnas: dependencies & design secrets
Yourdon/Constantine: coupling & cohesion


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.
ESSAYS ON SOFTWARE ENGINEERING

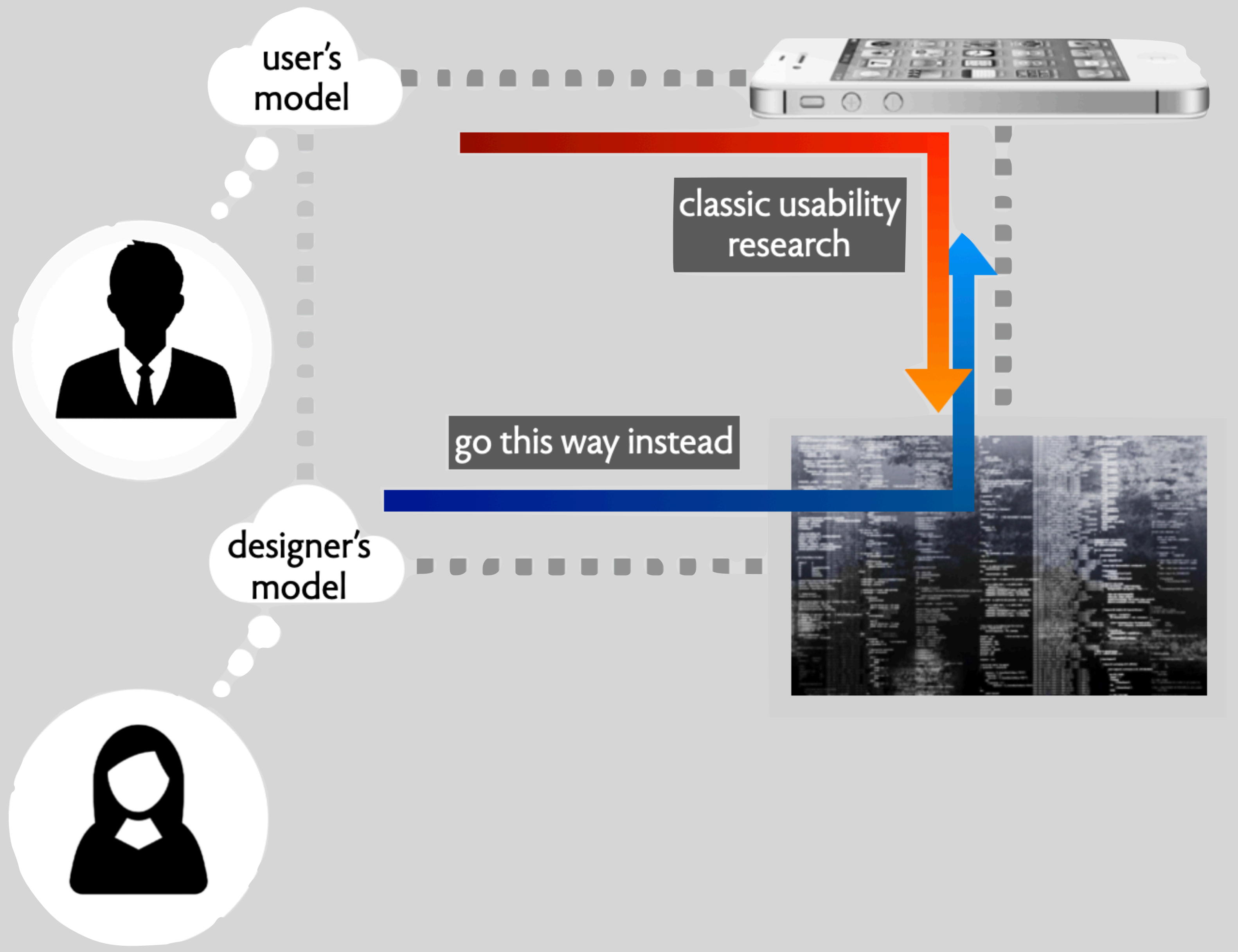
conceptual models
user-centered computing at PARC
Brooks's essence & accident

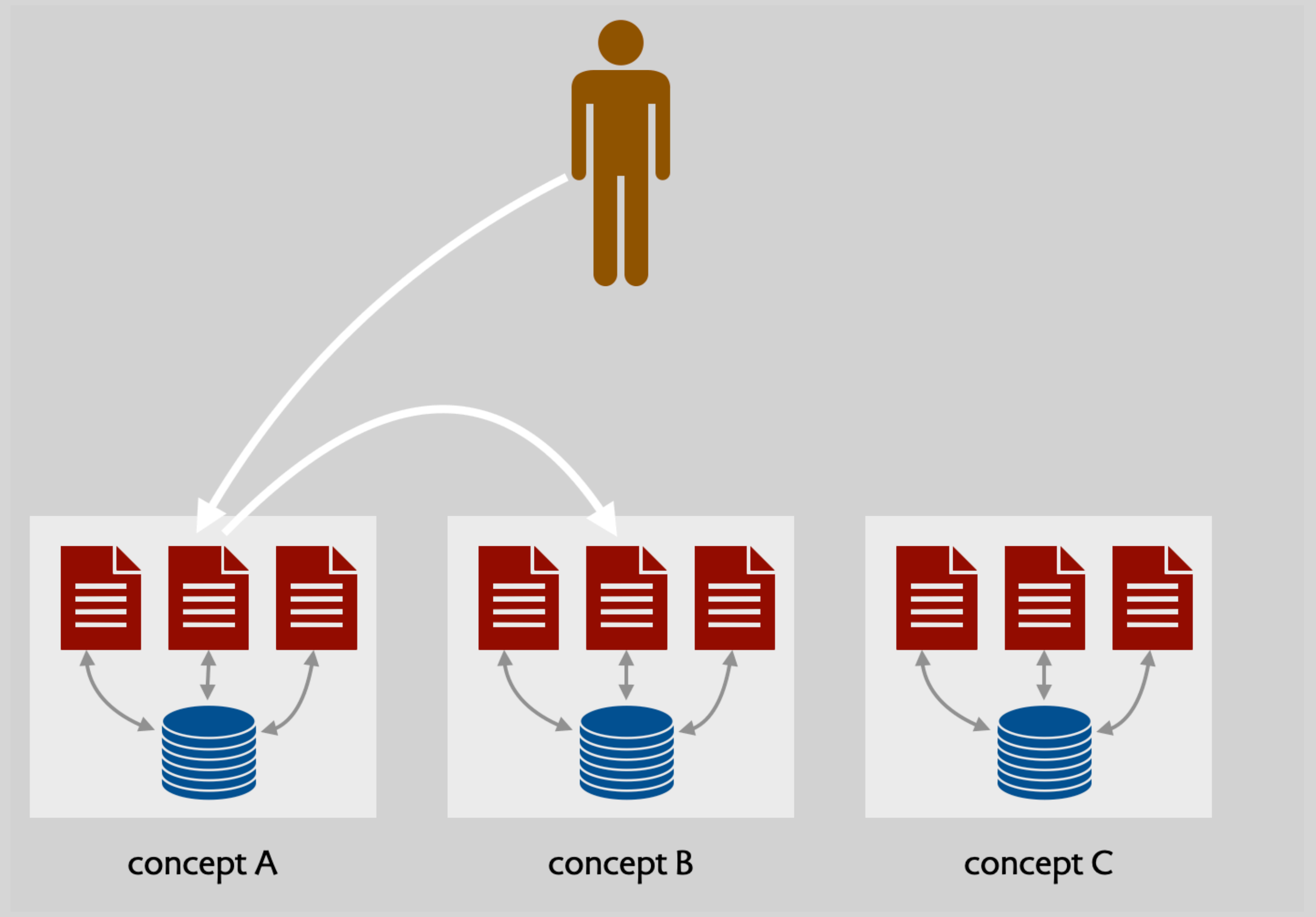

J. M. Spivey
The Z Notation
A Reference Manual
SECOND EDITION
PRENTICE HALL INTERNATIONAL SERIES IN COMPUTER SCIENCE
C.A.R. HOARE SERIES EDITOR

function as actions on states
formal methods (eg, Z, VDM, B)
entity-relationship data model

takeaways

takeaways





conceptual models
shared by designer & user
essential to good UX
but not just mapping!

structuring with concepts
modularity in design & UX
a language for design
place(s) for design discussion

what’s next?

what next?

let's design some concepts!
what's in a concept?
criteria for good concepts
why are concepts not just objects?