



concepts  
&  
software

Daniel Jackson · Princeton · March 2, 2020



how I got here



# Google Drive Sucks

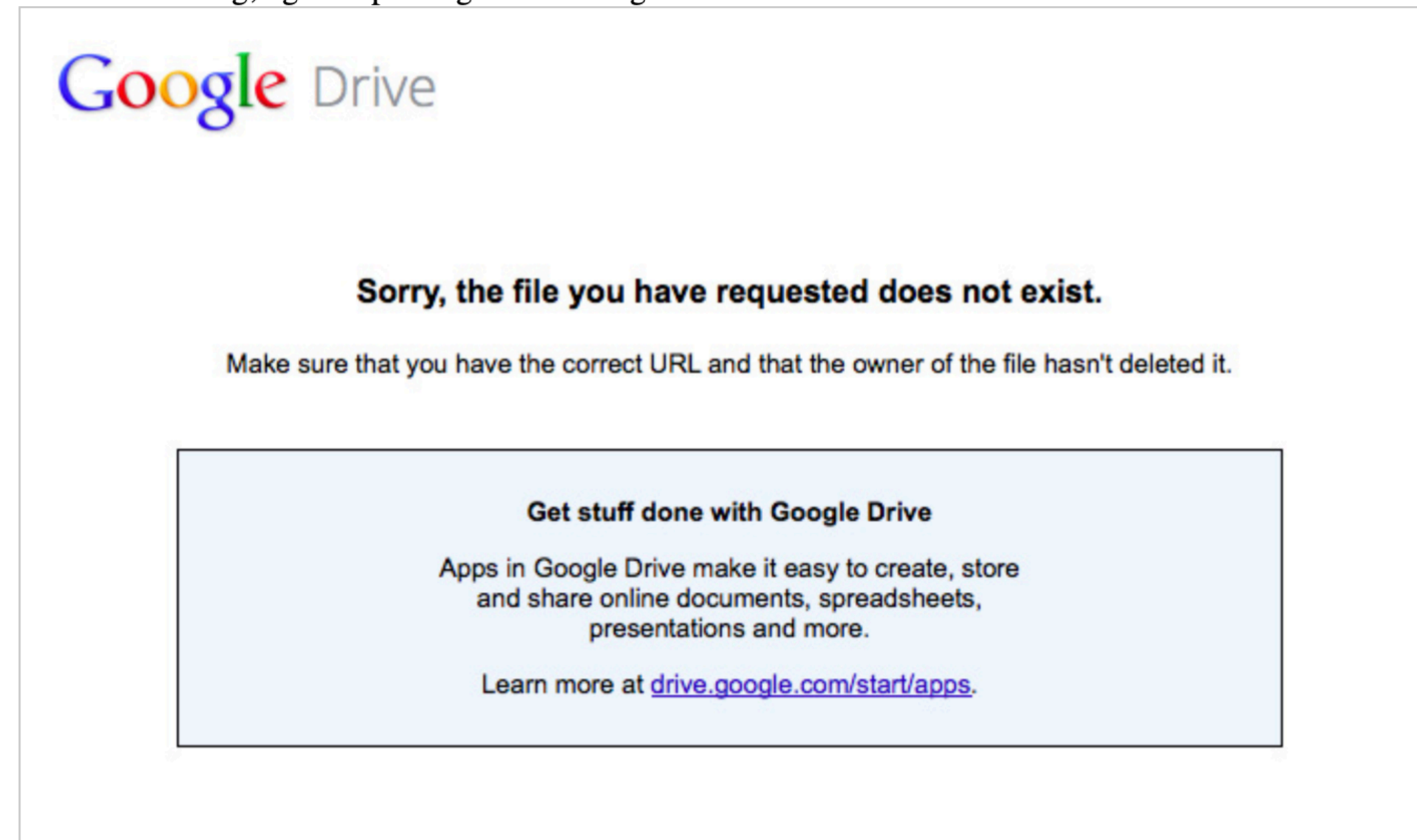
## Google Drive storage loses Google Docs data

I lost years of work and personal memories that I saved as Google Docs files because of a poor user interface.

### What happened

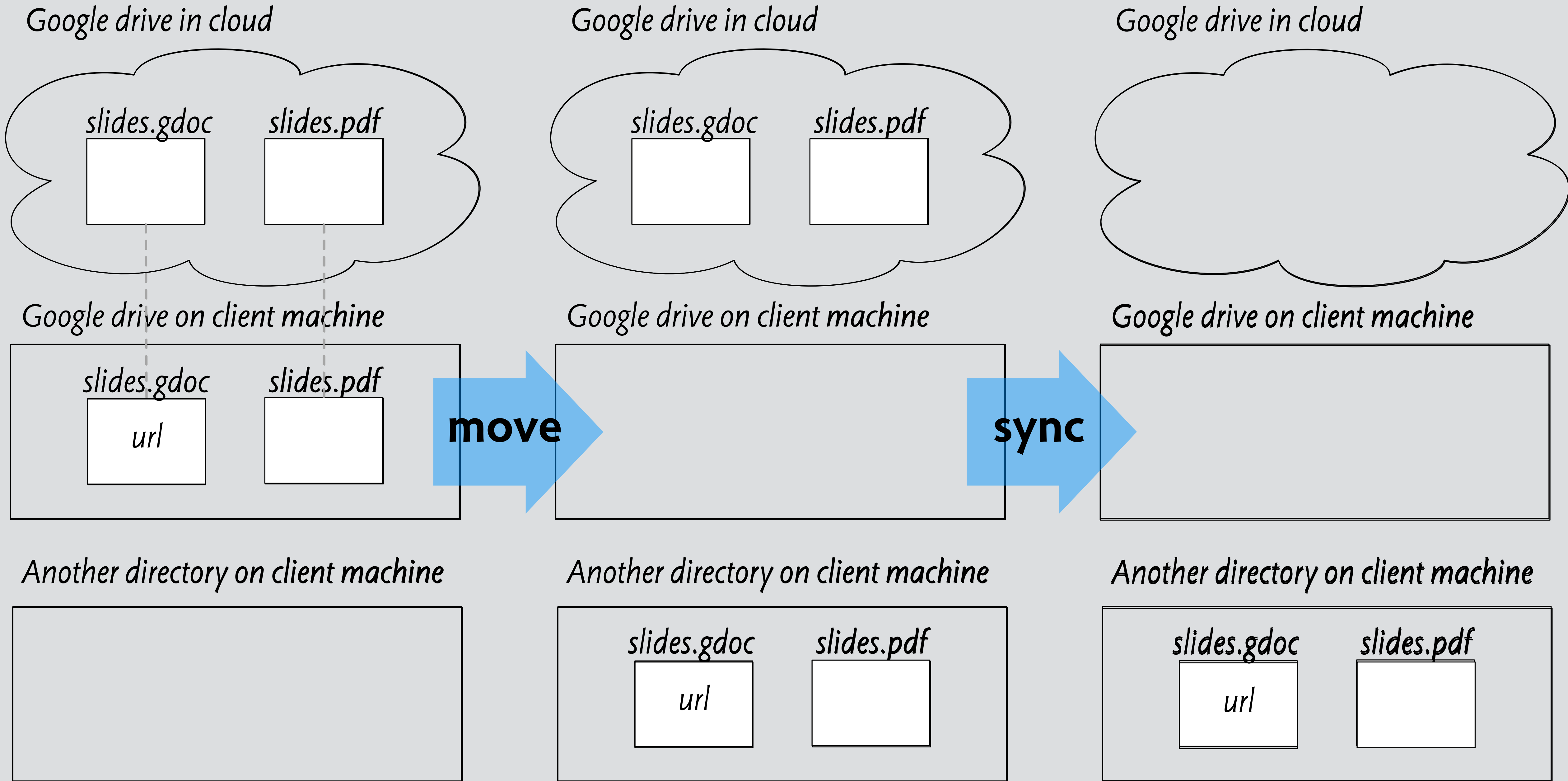
I was organizing my files on my local computer. I moved them around and out of my Google Drive folder which syncs files. I didn't think anything of it. In the process I got an email from Google saying I'm running out of storage. So I go to the Google Drive site and empty the trash. I didn't think anything of it. I finish organizing my files.

The next morning, I go to open a .gdoc file and get this error:



My heart sank. What happened to the work from yesterday? I opened another file. Then another. All of them the same message. I was starting to freak out.

what's wrong? the user interface? bugs in the code?





for robust, usable software...



understand the user

get the concepts right

sync    cloud app    trash

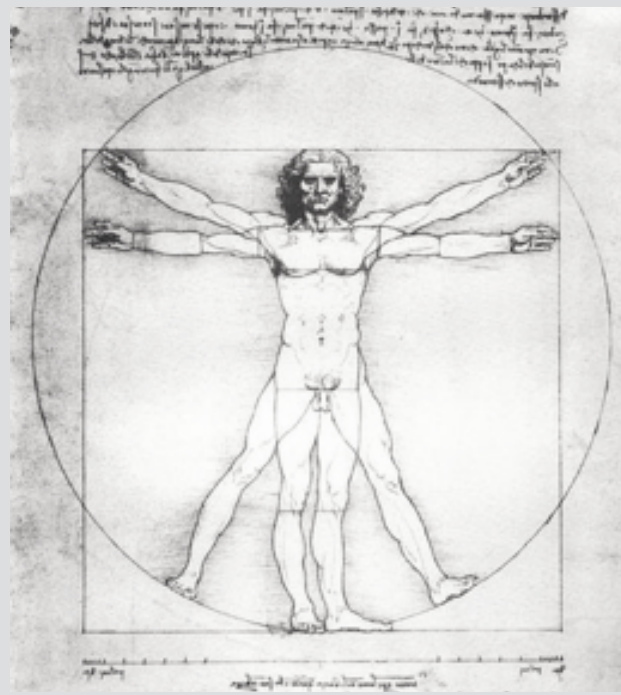
Three red icons are displayed horizontally: a sync symbol (a central dot with six lines radiating outwards), a cloud icon, and a trash can icon. Below each icon is its corresponding text label: 'sync', 'cloud app', and 'trash'.

design the user interface



avoid bugs in code

# the conceptual level in UX design



**physical**

color, size, layout,  
touch, sound



**linguistic**

icons, labels, tooltips,  
site structure, info scent

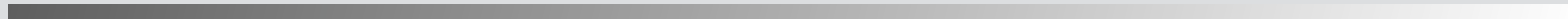


**conceptual**

semantics, actions,  
data model, purpose

concrete

abstract





# not a new idea

It is clear that users attempt to make sense—by building mental models—of the behavior of a system as they use it. If a simple model is not explicitly or implicitly provided, users formulate their own myths about how the system works... [I]f the user is to understand the system, the system has to be designed with an explicit conceptual model that is easy enough for the user to learn. We call this the intended user's model, because it is the model the designer intends the user to learn.

Stuart Card & Thomas Moran (1986)



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)



# a research agenda

- what exactly is a concept? how do you **express** one?
- what is the **semantics**? how are concepts composed?
- what are the design **principles** for concepts?
- what is the essence of **design** with concepts?



**what is a concept?**

# concepts characterize apps



Emacs (1985)  
**line, buffer**



Finder (1984)  
**folder, trash**



Word (1983)  
**paragraph,  
format, style**



Photoshop (1988)  
**pixelarray,  
layer, mask**



Facebook (2004)  
**update, friend,  
like**



Drive (2012)  
**synchronization,  
sharing**



Google Doc (2009)  
**edit (OT),  
cloud file**



Quark (1987)  
**page, textflow**



# example: style concept

The image shows a screenshot of a text editor window titled "concepts — Edited". The main editing area contains two sections of text. The first section, titled "Introduction", discusses the challenge of improving software quality by eliminating defects. The second section, titled "Defects", discusses the widespread assumption that defect elimination is the key to better software. On the right side of the window is a "Text" style panel. The panel has a "Section" dropdown menu currently set to "Section". Below this are three tabs: "Style", "Layout", and "More". The "Style" tab is active and contains several controls: a "Font" dropdown set to "Arno Pro", a "Bold Italic" dropdown, a "24 pt" size dropdown, a row of buttons for "B", "I", "U", and "S" (Strikethrough), and a "Character Styles" dropdown set to "None". Below these is a "Text Color" dropdown with a color picker showing black. At the bottom of the panel is an "Alignment" section with four alignment icons (left, center, right, justified) and a "Text Direction" icon.

concepts — Edited

**Introduction**

How can we improve the quality of software? Make it more usable, robust and secure? Many responses to this challenge make a fundamental assumption: that quality is achieved by eliminating defects. It seems like a plausible enough idea. If you can find the parts of the interface that confuse users and polish or replace them. that will surely make it more usable. And if you can remove the bugs that cause the most frequent crashes, that should make it more robust. And how else to achieve security except by patching the vulnerabilities that hackers might exploit?

**Defects**

The assumption that defect elimination is the key to better software is so widespread that it is rarely questioned (and often not even explicitly articulated). Companies that make software like it because it can be applied incrementally, without major disruptions to their development process or to an often shaky codebase. Tool vendors promote it because it helps sell their products. Researchers adopt it because it makes their contributions easier to measure, and because they fear being accused of utopianism if they suggest avoiding defects in the first place.

Text

**Section**

Style Layout More

Font

Arno Pro

Bold Italic 24 pt

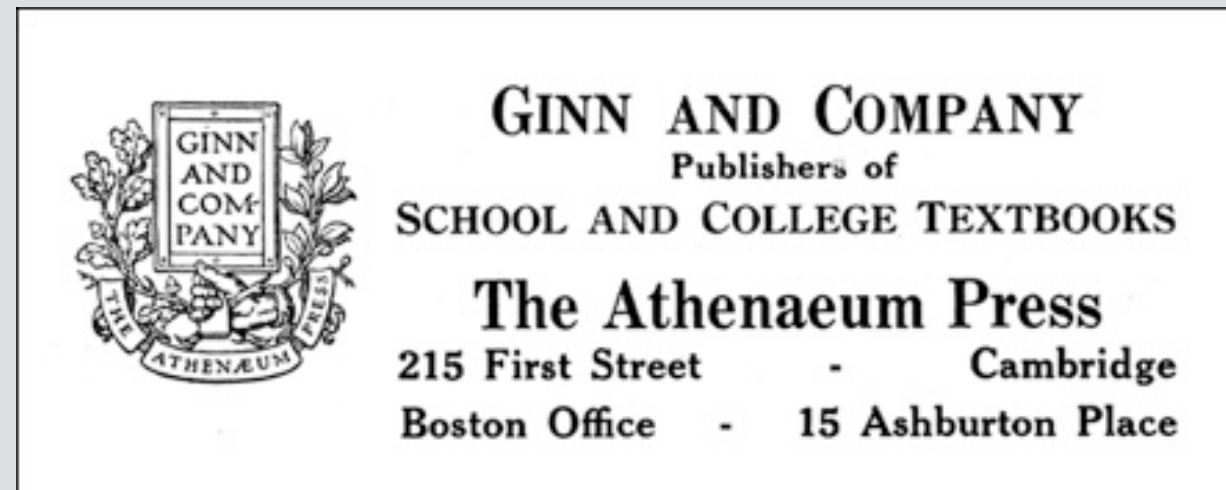
B / U S

Character Styles None

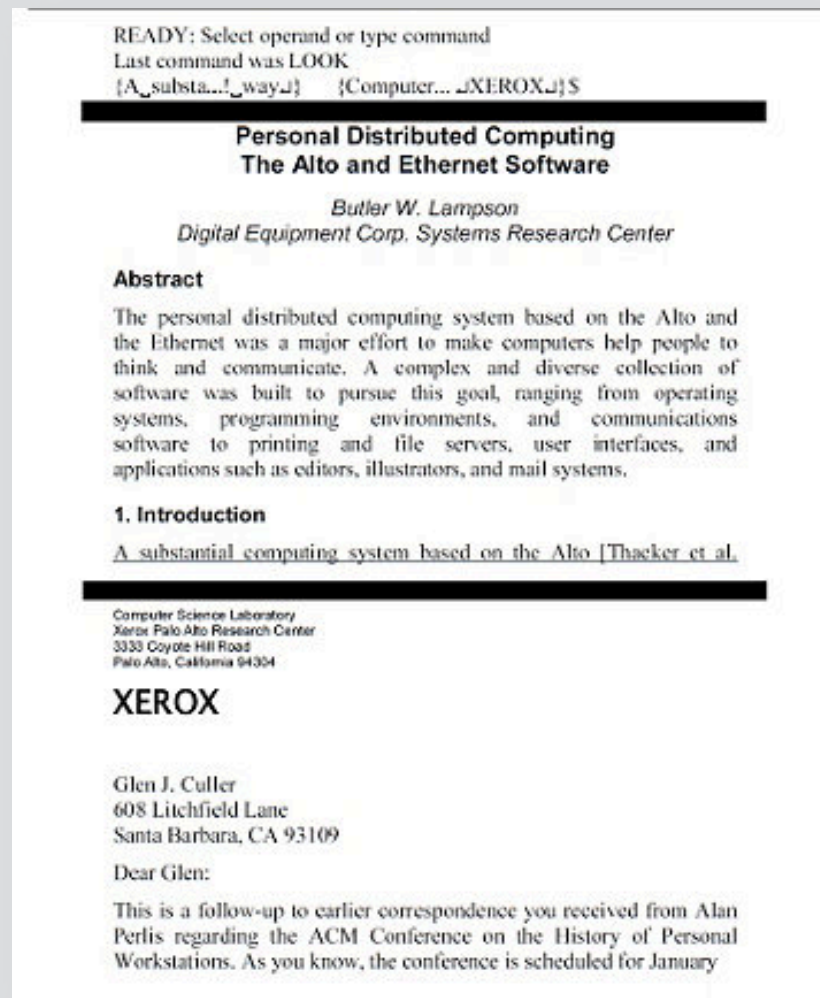
Text Color

Alignment

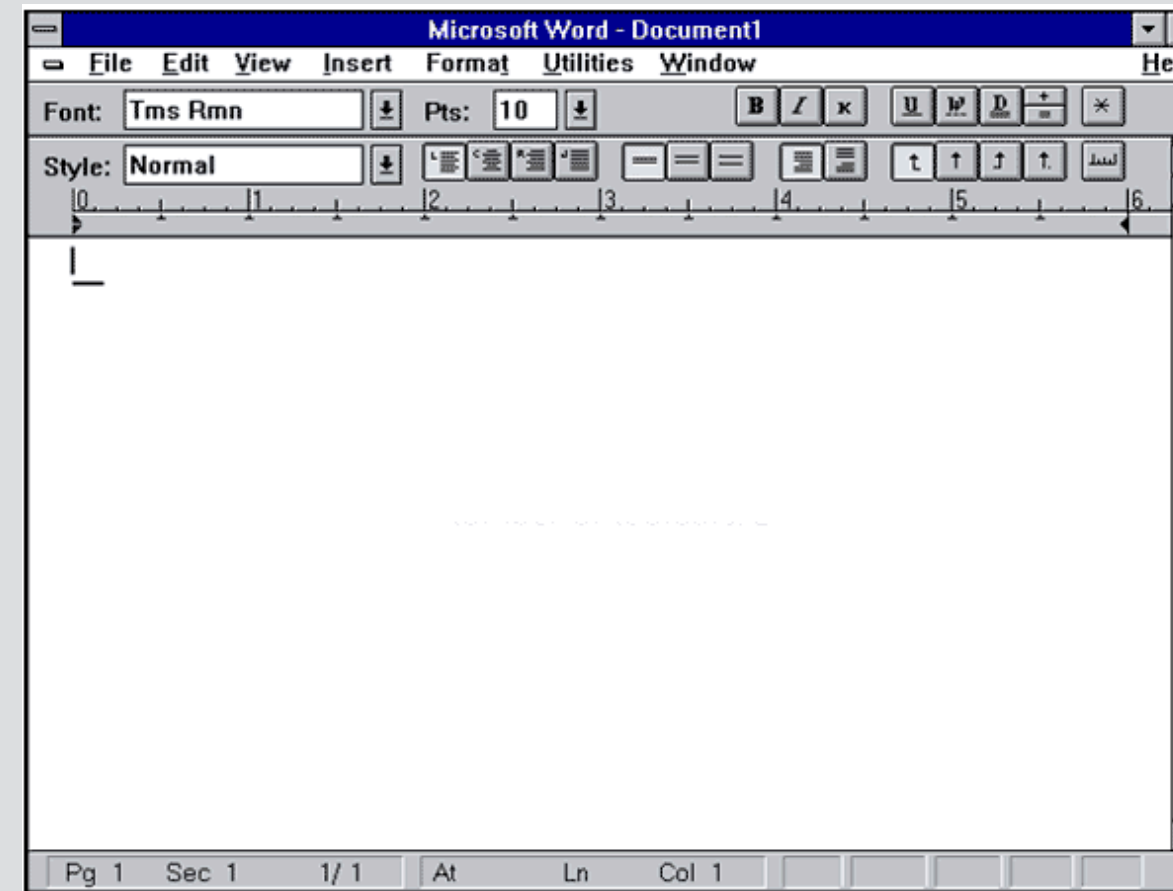
# the invention of style



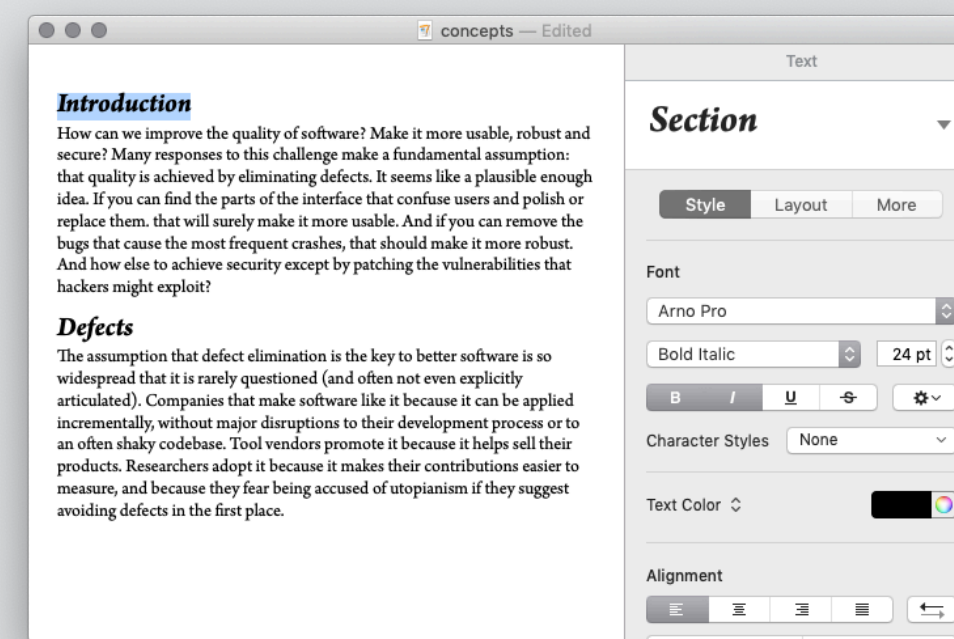
Tim Mott visits Ginn in 1974 brings idea of styles to PARC



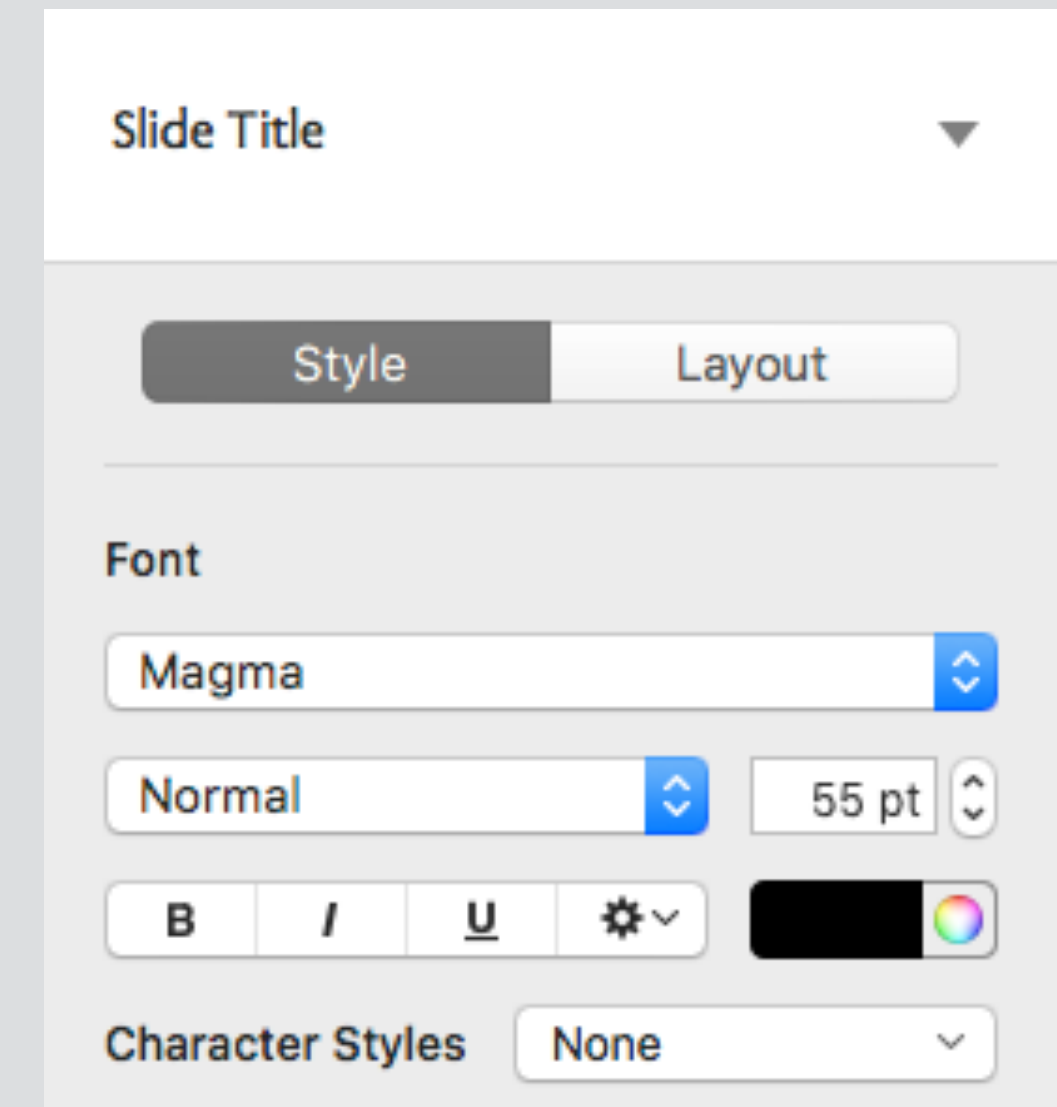
Charles Simonyi's team implements style in Bravo text editor



Simonyi brings style to Microsoft in 1983



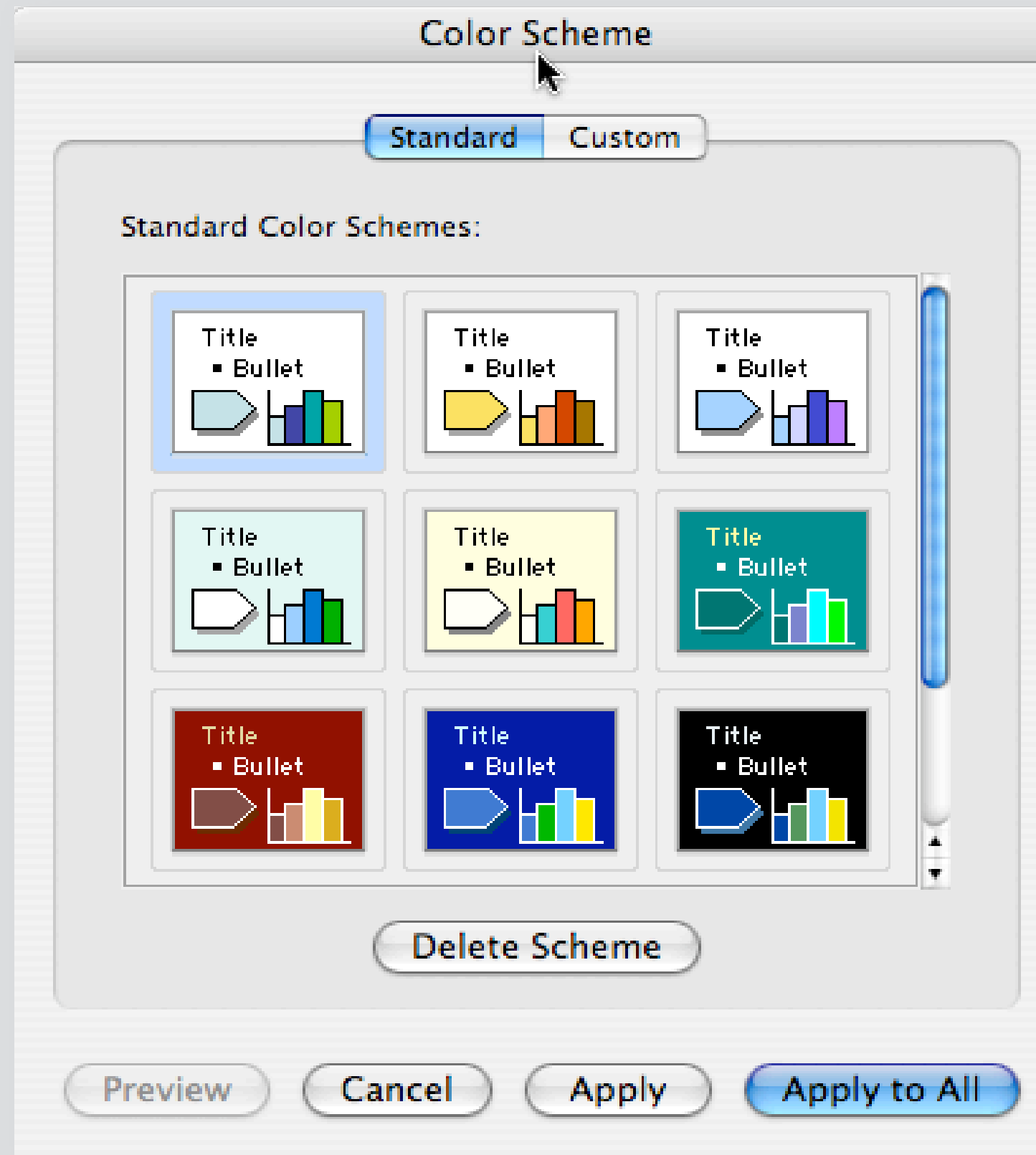
Apple Pages 2005



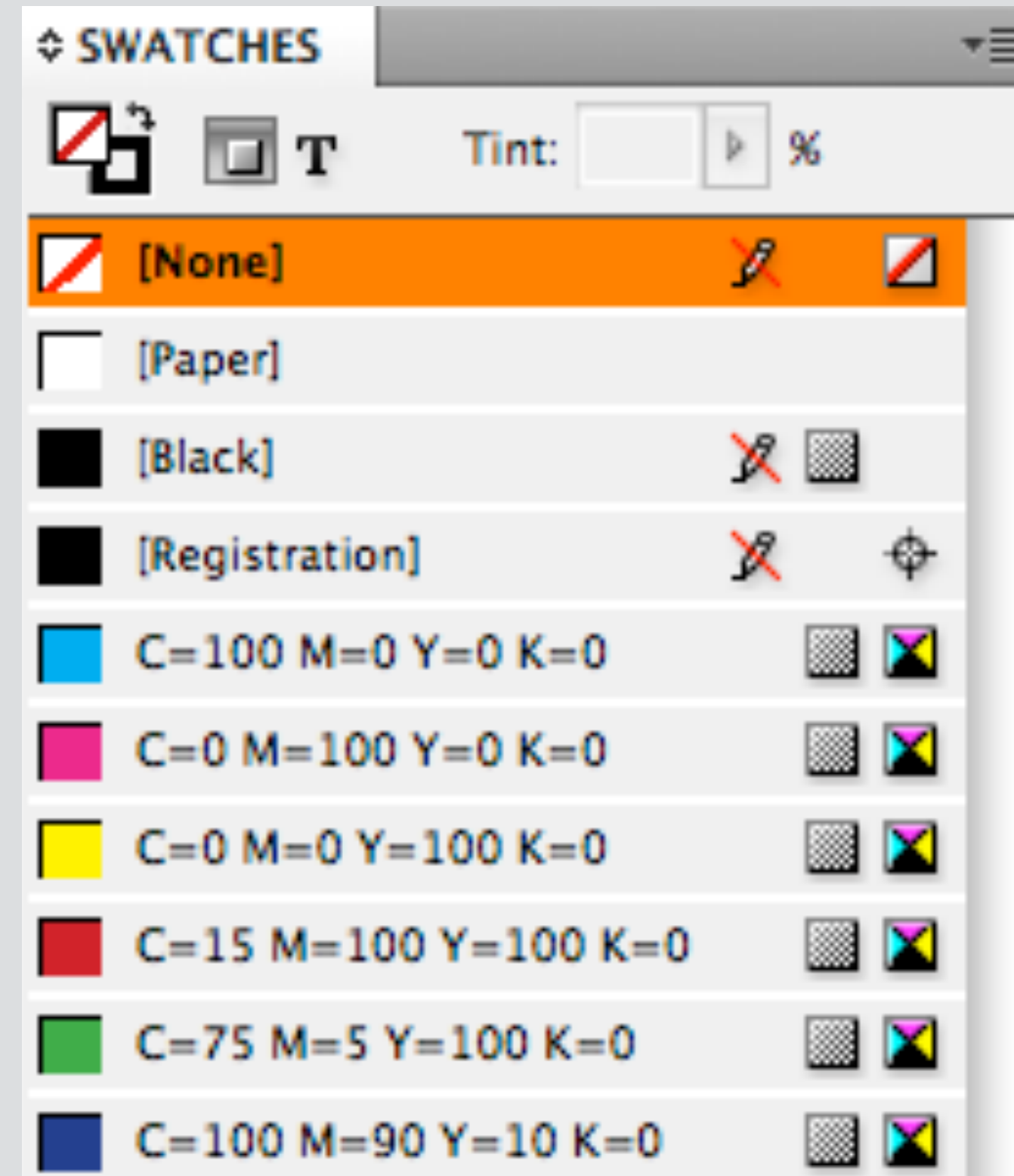
Apple Keynote adds style concept c. 2017



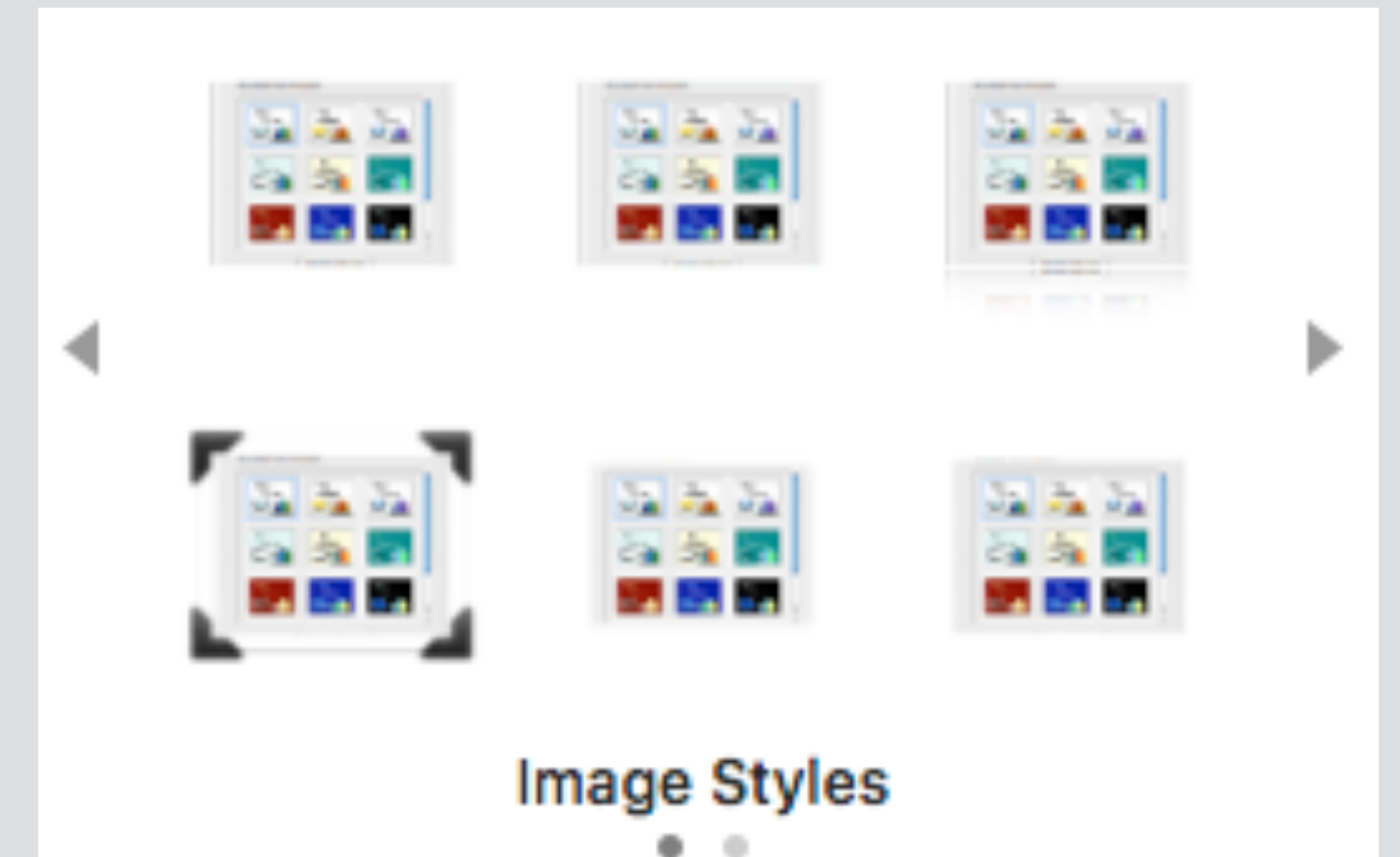
# other instances of style



Powerpoint color schemes



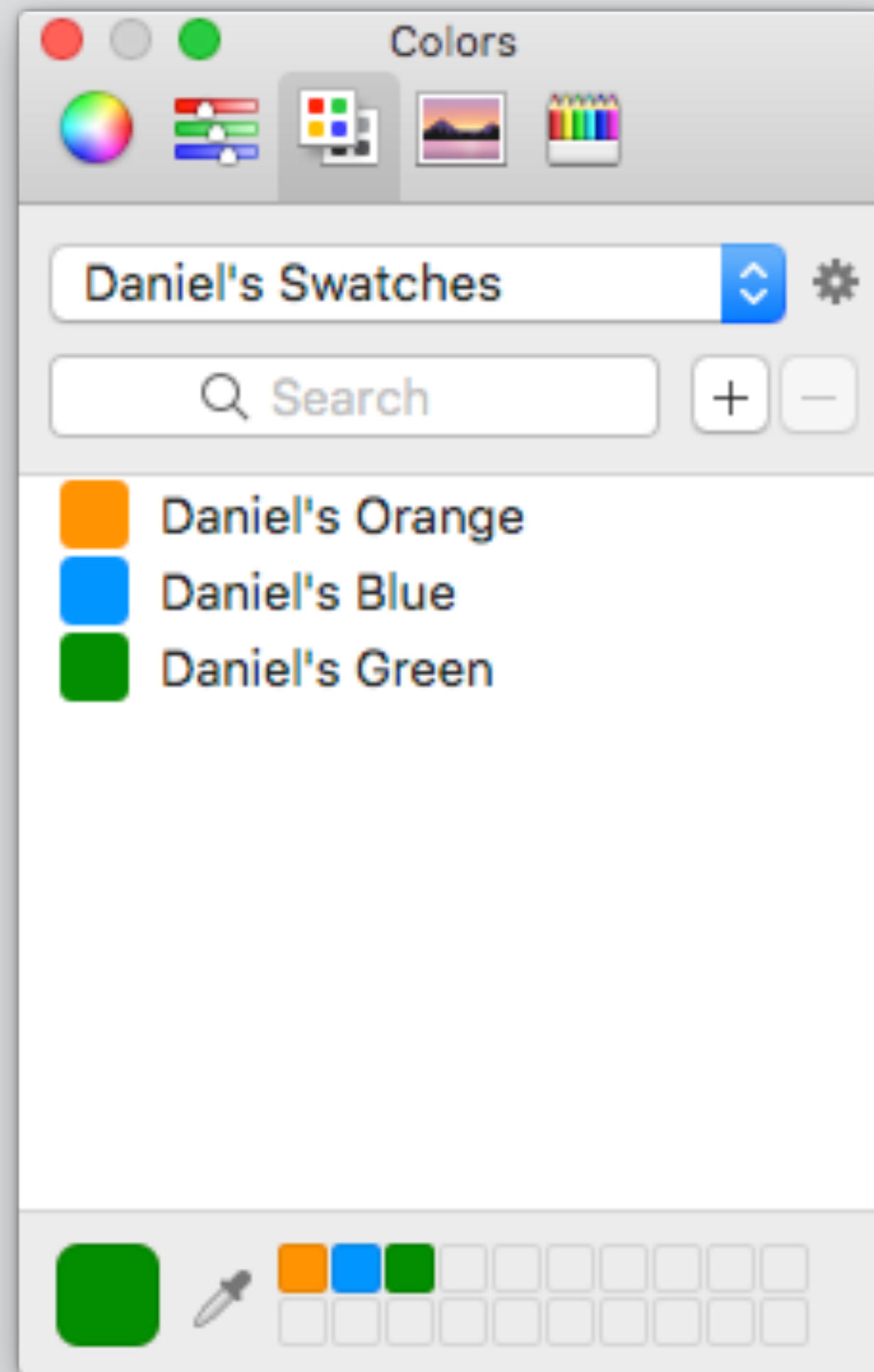
Indesign swatches



Keynote image styles



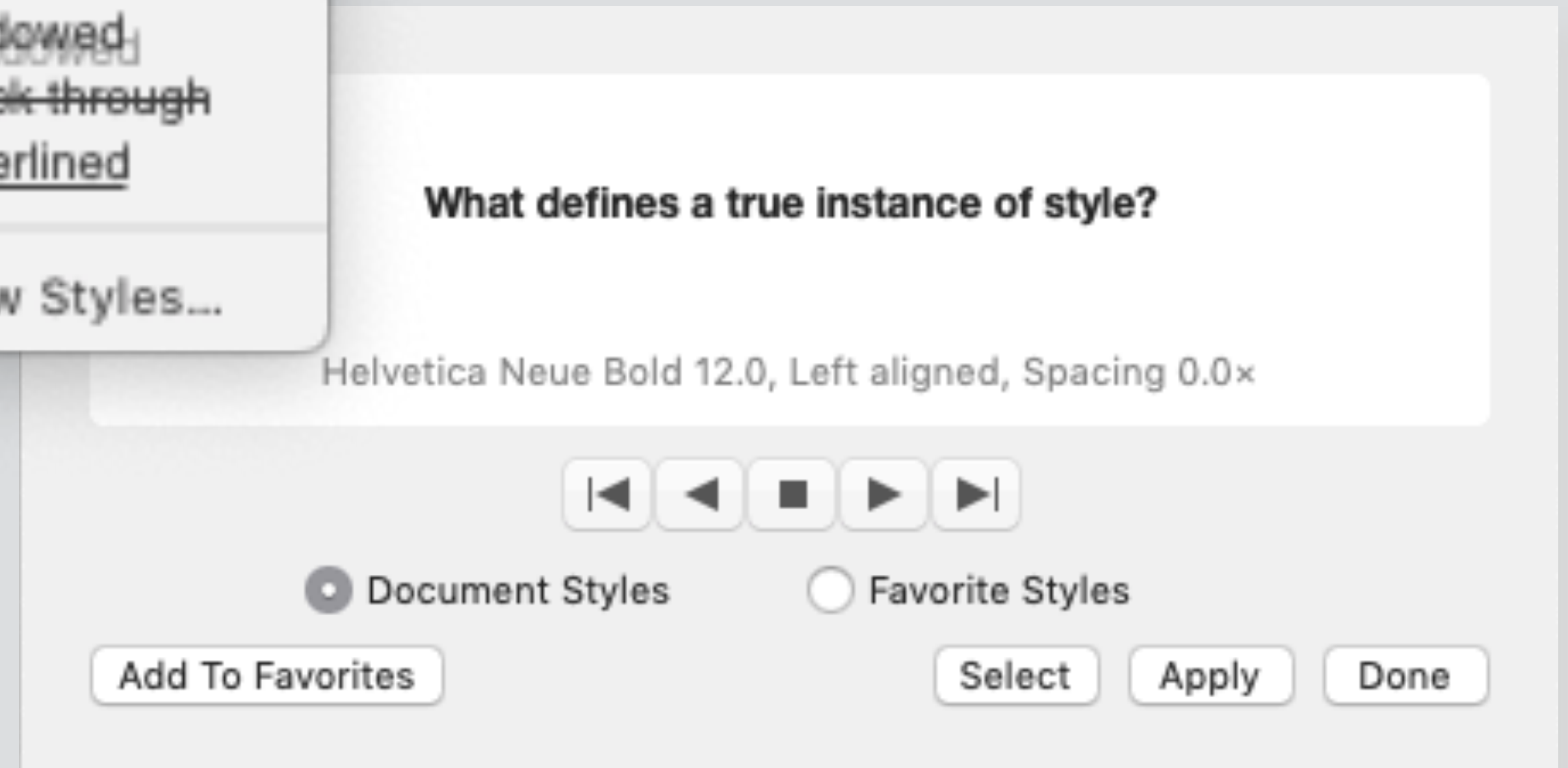
# non-instances: "pseudo-style"



Apple color swatches



**What defines a true instance of style?**  
Helvetica Neue Bold 12.0, Left aligned, Spacing 0.0x



TextEdit "styles"



There is no problem in computer science that cannot be solved by introducing another level of indirection.

*David Wheeler*

<b>concept</b> Style	name: essential for knowledge capture
<b>purpose</b> consistent formatting	purpose: why the concept exists
<b>structure</b> format: Style -> <b>one</b> Format style: Element -> <b>one</b> Style	structure: localized data model
<b>actions</b> define (s: Style, f: Format) s.format = f assign (e: Element, s: Style) e.style = s get(e: Element, <b>out</b> f: Format) f = e.style.format	actions: observable & atomic
<b>story</b> define(s, f); assign(e1, s); assign(e2, s); define(s, f'); get(e1, f1); get(e1, f2) => f1 = f2 = f'	story justifies design and explains it shows how behavior fulfills purpose



**concept** Style

no dependences

**purpose** consistent formatting

**structure**

format: Style -> **one** Format ← separation of concerns

style: Element -> **one** Style

maximal polymorphism

```
define (s: Style, f: Format)
```

```
  s.format = f
```

```
assign (e: Element, s: Style)
```

```
  e.style = s
```

```
get(e: Element, out f: Format)
```

```
  f = e.style.format
```

**story**

```
define(s, f);
```

```
assign(e1, s); assign(e2, s);
```

```
define(s, f'); get(e1, f1); get(e1, f2)
```

```
=> f1 = f2 = f'
```

story is an archetypal scenario

a theorem about all behaviors

generalizes concept variants



Michael Polanyi  
operational principle



# a concept handbook

## concepts indexed by purpose

consistent formatting:  
style, template, copy settings, ...

## design variants

override formats  
style inheritance  
next style  
partial styles  
shareable stylesheet



## known issues

deleting styles: what happens to elements?  
copying elements between documents  
need for "as is" values  
troublesome properties (eg, fontstyle)

## typical uses

formatting paragraphs & characters  
formatting graphic objects  
Word, Pages, CSS, ...

## often used with

paragraph  
format

## implementation hints

...

**what is the semantics?**

# meaning of a single concept



**concept** AuthUser

**purpose** identify users

**structure**

name, password: User -> **one** String  
sessions: Client -> **set** User

**actions**

register(n: Name, p: String, **out** u: User)  
login (n: Name, p: String, c: Client)  
logout (c: Client)  
auth (c: Client, **out** u: User)

**story**

register(n,p,u); login(n,p,c); auth(c,u')  
=> u' = u

meaning is set of histories (event traces):

```
{  
<>,  
< register(n0,p0,u0)>,  
< register(n0,p0,u0), login(n0,p0,c0)>,  
< register(n0,p0,u0), register(n1,p1,u1)>,  
...  
< register(n0,p0,u0), login(n0,p0,c0), auth(c0,u0)>,  
...  
}
```



# meaning of a single concept



**concept** Upvote

**purpose** track relative popularity

**structure**

votes: Item -> User

**actions**

upvote (i: Item, u: User)

votes += i->u

count (i: Item, **out** k: int)

k = #i.votes

**story**

no upvote(i,u) **then** ...

count(i, k); upvote(i,u); count(i, k')

=> k'>k

```
{  
< >,  
...  
< count(i0, 0) >,  
...  
< upvote(i0, u0) >,  
...  
< upvote(i0, u0), count(i0, 1) >,  
...  
< count(i0, 0), upvote(i0, u0), count(i0, 1) >,  
...  
}
```

▲ How to rewrite it in Rust (michaelfbryan.com)

173 points by FBT 5 hours ago | hide | past | web | favorite | 15 comments

post concept

auth concept

upvote concept

Empty text input box for comments.

add comment

comment concept

▲ sorenbs 2 hours ago [-]

We did a similar thing with a Scala -> Rust rewrite for the <http://prisma.io> query engine.

By rewriting small components and integrating them into the existing project using Javas native interface, our small team of 5 developers were able to pull off this massive rewrite in just under a year. The resulting code base is rearchitected in a few very important ways, but mostly follows the same structure.

And because we kept and evolved our old Scala based test suite, we have a very high confidence in the rewrite.

When Async/.await finally landed, we could switch over very quickly, and it has been a joy to focus on benchmarks and performance over the last month. Spoiler: Rust is faster than Scala :-D

[reply](#)

▲ tombert 1 hour ago [-]

I promise that this is asked genuinely and isn't some sort of veiled "gotcha!" (it's tough to tell on the internet sometimes); what was the reason for a change from Scala to Rust?

I ask because Scala already has a good type system and the JVM typically has good performance nowadays, particularly with something like GraalVM, so I am actually really curious to why you felt a Rust rewrite was a good idea.

[reply](#)

# making an app by composing concepts

**concept** Post

**actions**

new (a: Author, s: String, out p: Post)

edit (p: Post, s: String)

get (a: Author, out ps: set Post)

**concept** Comment

**actions**

new (a: Author, s: String, t: Target, **out** c: Comment)

get (t: Target, out cs: set Comment)

**concept** Upvote

**actions**

upvote (i: Item, u: User)

count (i: Item, out r: Int)

**concept** Owner

**actions**

register (o: Owner, i: Item)

owns (o: Owner, i: Item)

**concept** AuthUser

**actions**

register (n: Name, p: String, **out** u: User)

login (n: Name, p: String, c: Client)

logout (c: Client)

auth (c: Client, out u: User)

**app** HackerNews

**includes** Post, Comment, Upvote, AuthUser, Owner

**actions**

newPost (s: String, **out** p: Post, c: Client)

Post.new(u, s, p)

Owner.register(u, p)

AuthUser.auth (c, u)

editPost (p: Post, s: String, c: Client)

Post.edit(p, s)

Owner.owns(u, p)

AuthUser.auth (c, u)

newComment (p: Post, **out** x: Comment, c: Client)

Comment.new(u, s, p, x)

AuthUser.auth (c, u)

upvotePost (p: Post, c: Client)

Upvote.upvote (p, u)

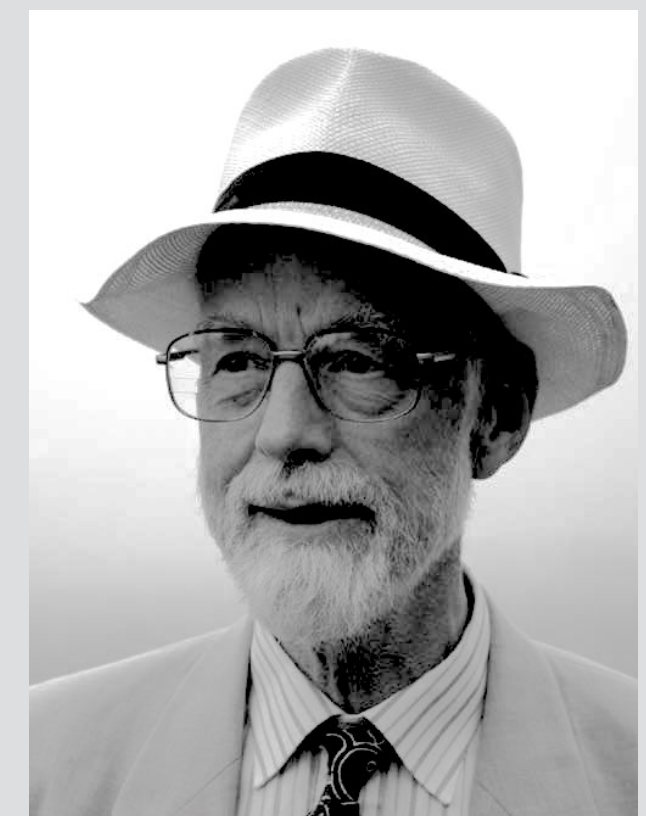
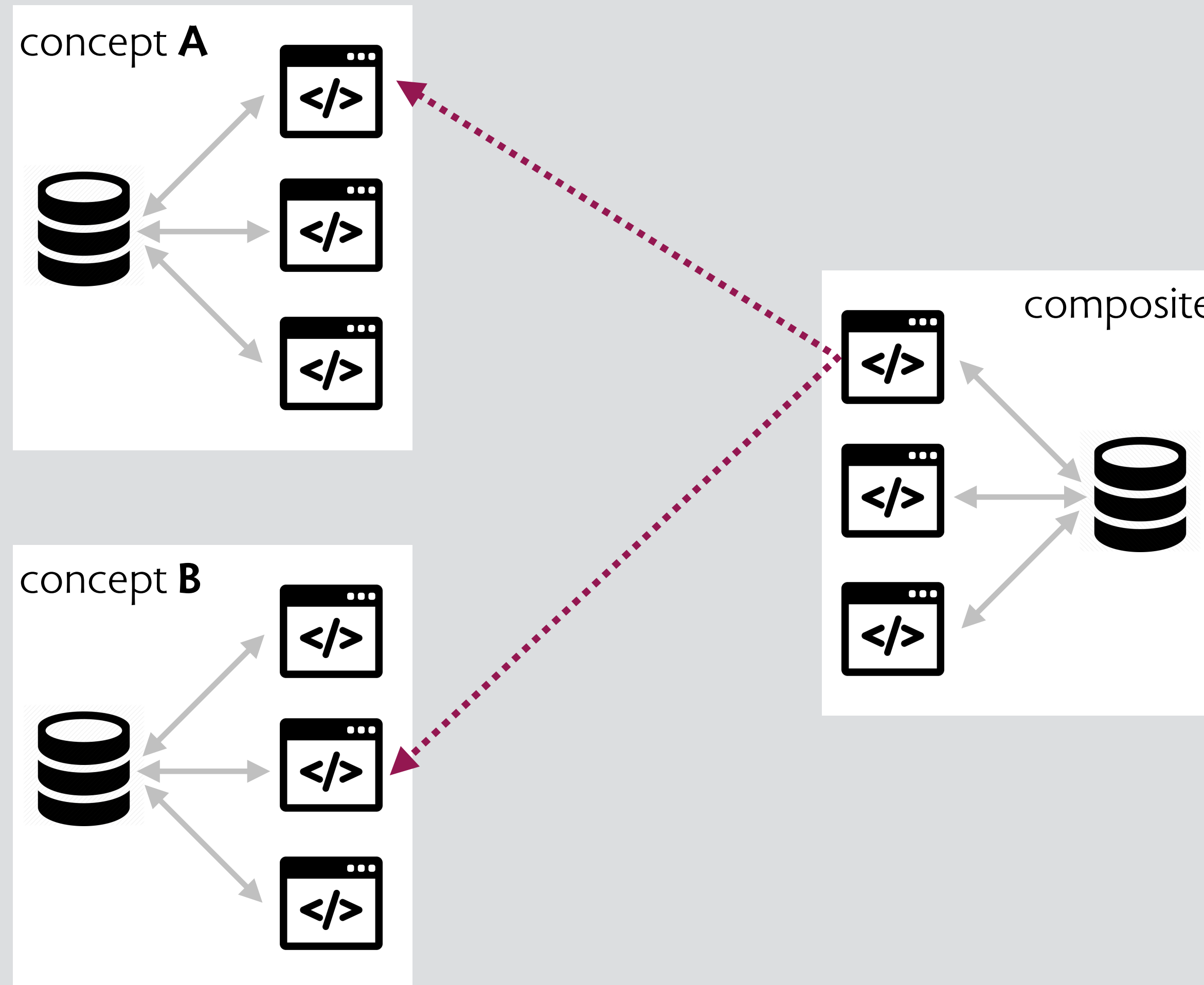
AuthUser.auth (c, u)

...



# projecting actions

each action in composite system  
interpreted as zero or more actions in each concept



Tony Hoare  
CSP (1978)

# check that projected events meet concept specifications

register (n1, p1, u1)

AuthUser.register (n1, p1, u1)

...

login (n1, p2)

AuthUser.login (n1, p1, c1)

...

newPost (s1, p1)

AuthUser.auth (c1, u1)

Post.new(u1, s1, p1)

Owner.register(u1, p1)

upvotePost (p1)

AuthUser.auth (c1, u1)

Upvote.upvote (p1, u1)

**concept** AuthUser

AuthUser.register (n1, p1, u1)

AuthUser.login (n1, p1, c1)

AuthUser.auth (c1, u1)

AuthUser.auth (c1, u1)



**concept** Post

Post.new(u1, s1, p1)



**concept** Owner

Owner.register(u1, p1)



**concept** Upvote

Upvote.upvote (p1, u1)



# what concepts are (and are not)



**inventive**

not domain entities  
just "out there"



**purposeful**

not fragments of  
functionality



**behavioral**

not data models  
or ontologies



**self-contained**

not datatypes  
or modules



**reusable**

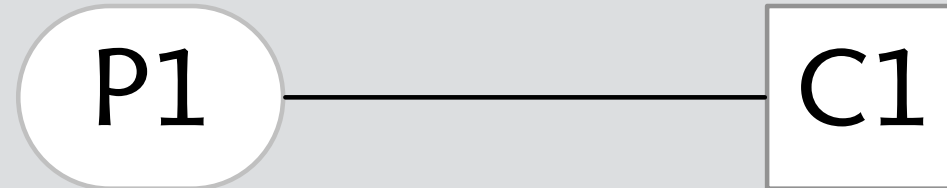
polymorphic  
separated concerns



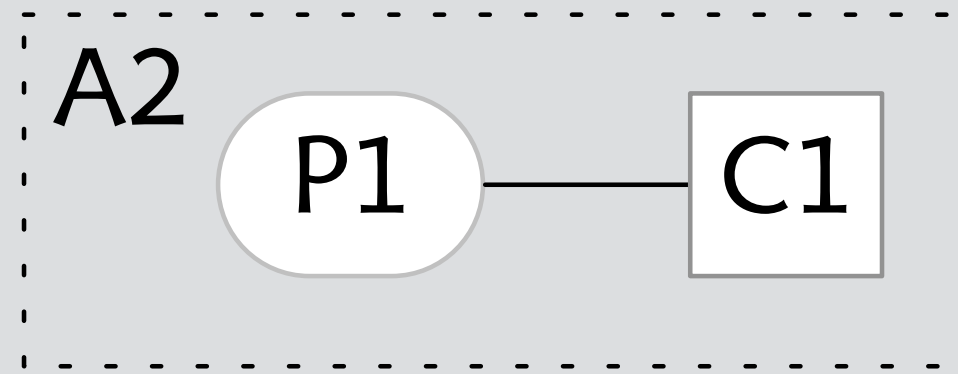
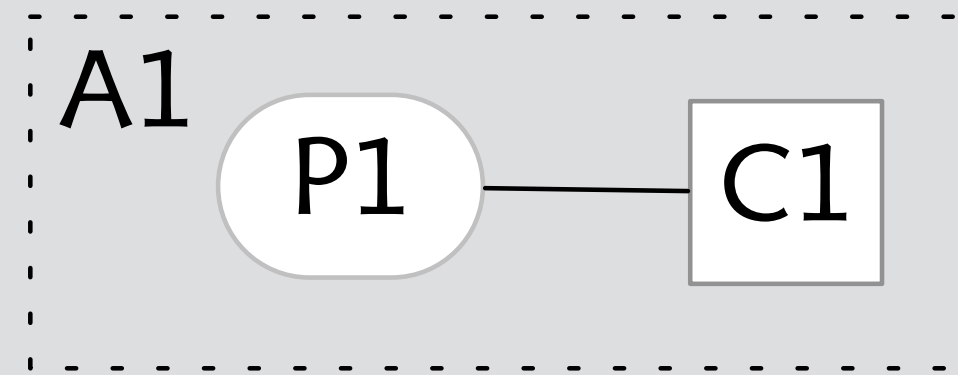
**what design principles?**

# three design rules

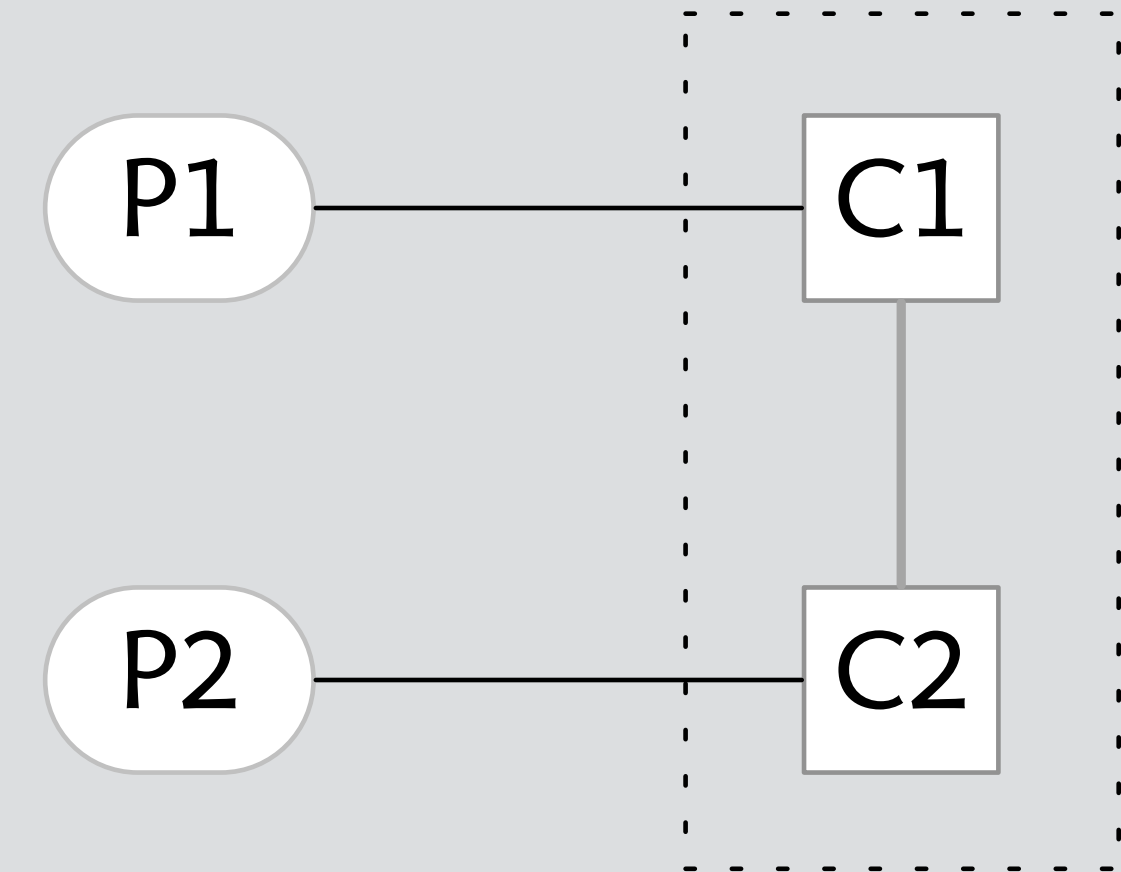
**instrumentality**  
one concept per purpose



**familiarity**  
same concept for given purpose



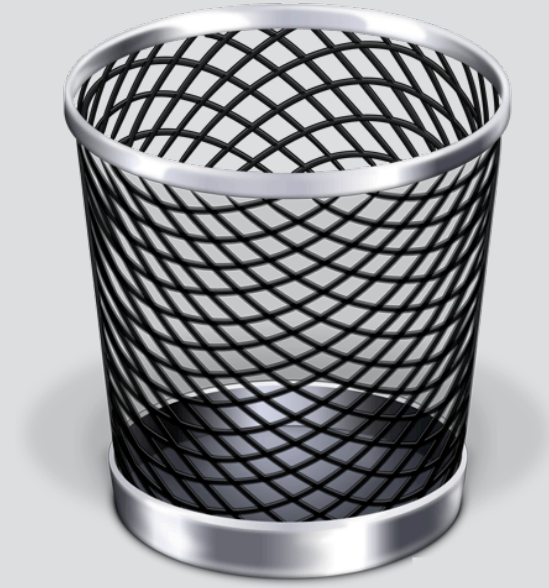
**integrity**  
composition preserves concepts



**instrumentality**



# understanding why: the key to usability



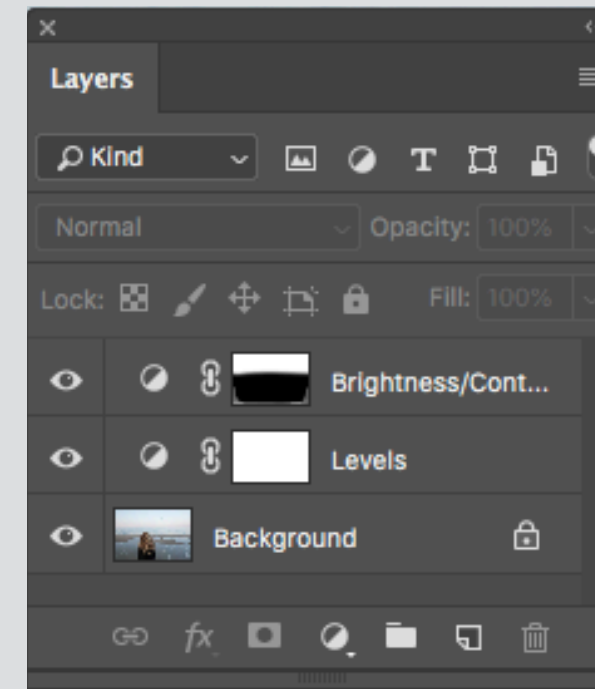
Macintosh Trash

wrong purpose

deleting things

undeleting things

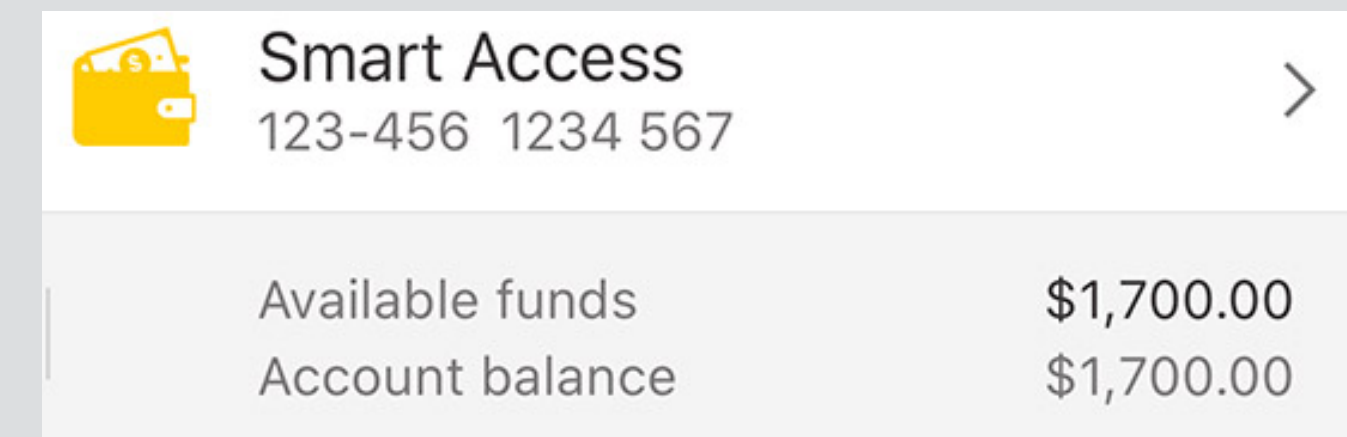
right purpose



Photoshop Layers

stacking objects

non-destructive editing



Available Funds

signal that deposits are safe

permission to use

# a conceptual flaw in Twitter



We are changing our star icon for favorites to a heart and we'll be calling them likes. We want to make Twitter easier and more rewarding to use, and **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*

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

The problem for Twitter is that the "favorite" function had developed a range of uses over time, many of which are known only to the journalists and social-media experts who spend all their time on the service. For some (including me), **clicking the star icon was a way of saving a tweet for later**, or of sending a link that was being shared to a service like Instapaper or Pocket. *Mathew Ingram*

I've favorited more than 60,000 tweets over the years, and in that time I've come to appreciate how versatile that little button is. I use it as **a kind of read receipt** to acknowledge replies; I use it whenever a tweet makes me laugh out loud; I use it when someone criticizes me by name in the hopes that seeing it's one of my "favorite" tweets will confuse and upset them. *Casey Newton*

If Twitter integrated a simple heart gesture into each Tweet, engagement across the entire service would explode. More of us would be getting loving feedback on our posts and that would **directly encourage more posting** and more frequent visits to Twitter. *Chris Sacca*



# confused concepts lead to confused users

 **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](https://pic.twitter.com/XiNd2jiLUF)



# how Twitter resolved the conceptual flaw



**Like:** public

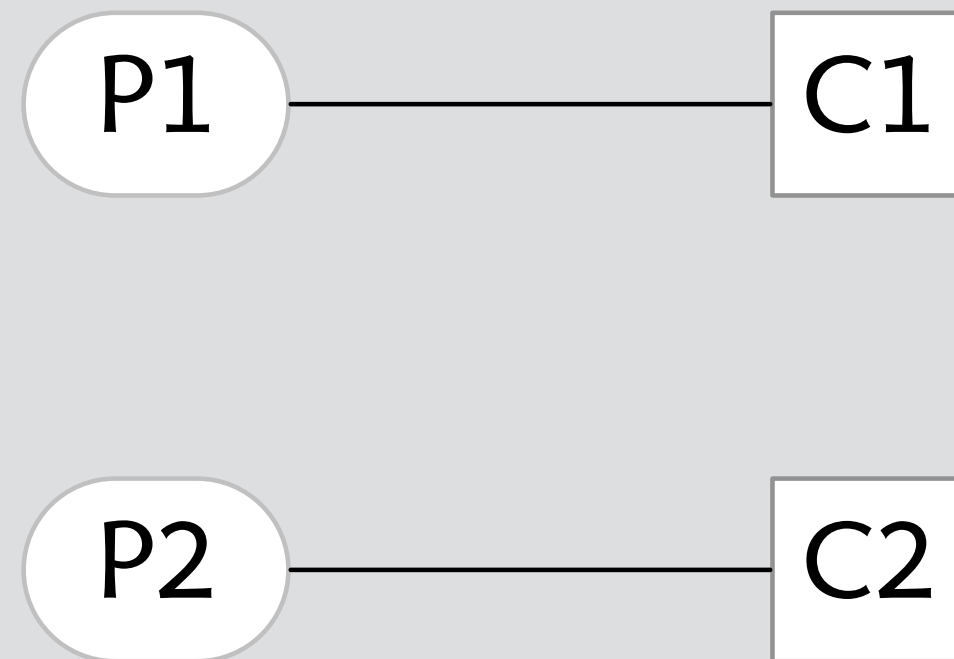


**Bookmark:** private

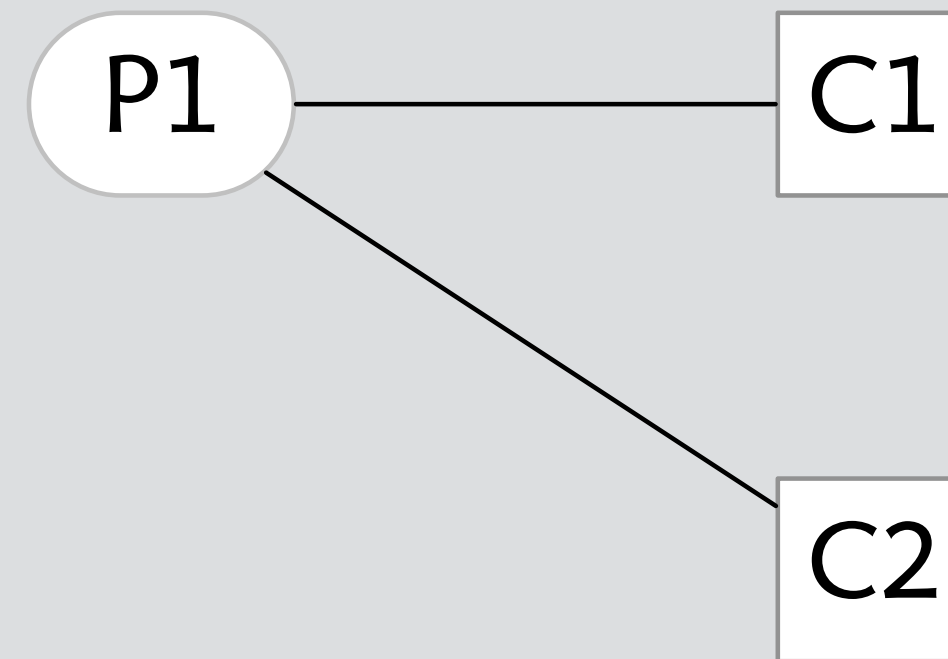
- Send via Direct Message
- Add Tweet to Bookmarks
- Copy link to Tweet
- Share Tweet via ...

# instrumentality: one purpose, no more and no less

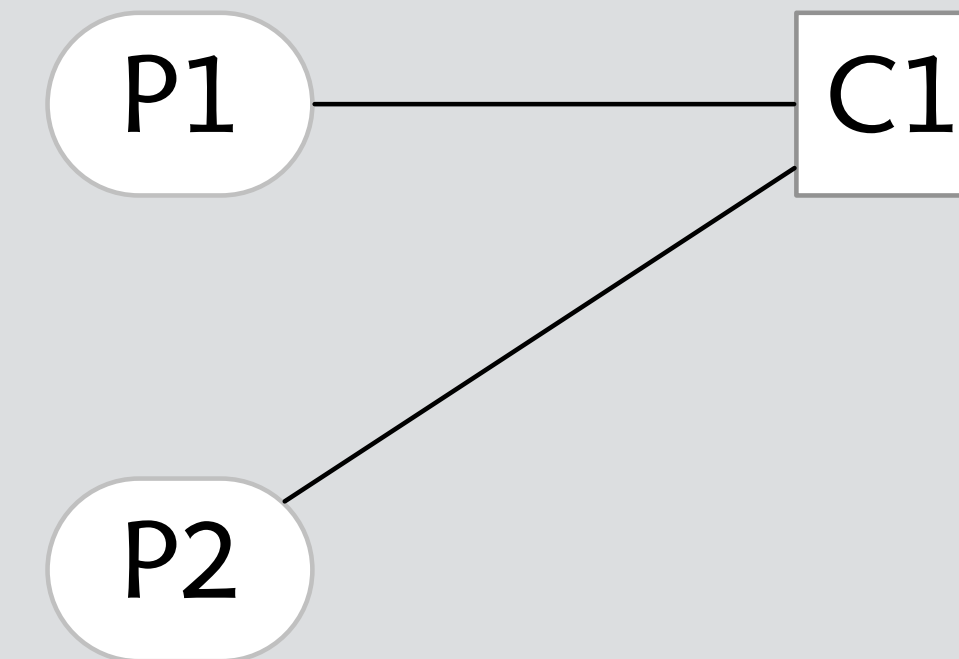
**instrumentality**  
purposes:concepts are 1:1



**redundancy**  
>1 concept per purpose



**overloading**  
>1 purpose per concept



Nam Suh:  
Axiomatic Design

[Home](#) › [Quick Tech Tip: Disabling Gmail's Category Tabs](#)

## Quick Tech Tip: Disabling Gmail's Category Tabs

Mon, 07/29/2013 - 12:17 | [Chuck Gray](#)

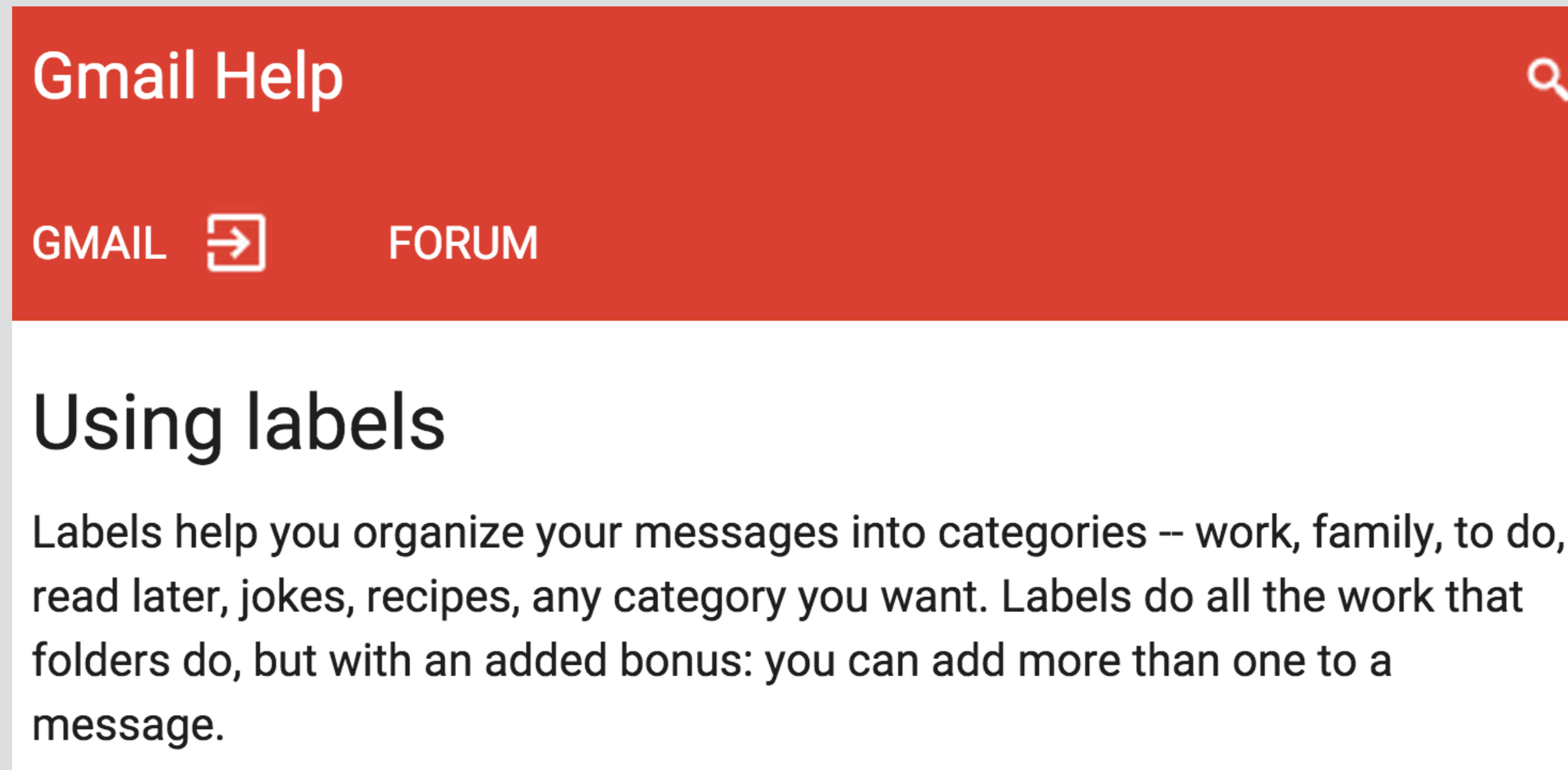
in [LibraryPoint Blog](#) [Tech Tutorials](#) [Teen Blog](#) [Tech Answers](#) [Science and Technology](#) [Self-Help and Instructional](#)




Are you a Gmail user? Did you wake up a week or two ago to find that your new messages were now being automatically organized by Gmail into tabs of different, pre-determined categories? And, did you think, like me, that they were **really ugly, stupid, and unnecessary?** Here's a quick tip on how to rid yourself of them!


initial reaction to categories





The screenshot shows the Gmail Help interface. At the top, there is a red header bar with the text "Gmail Help" on the left and a magnifying glass search icon on the right. Below the header, there are two navigation options: "GMAIL" with a square icon containing a right-pointing arrow, and "FORUM". The main content area is white and features the title "Using labels" in a large, bold font. Below the title, a paragraph of text explains the purpose of labels in Gmail.

**Gmail Help** 

[GMAIL](#)  [FORUM](#)

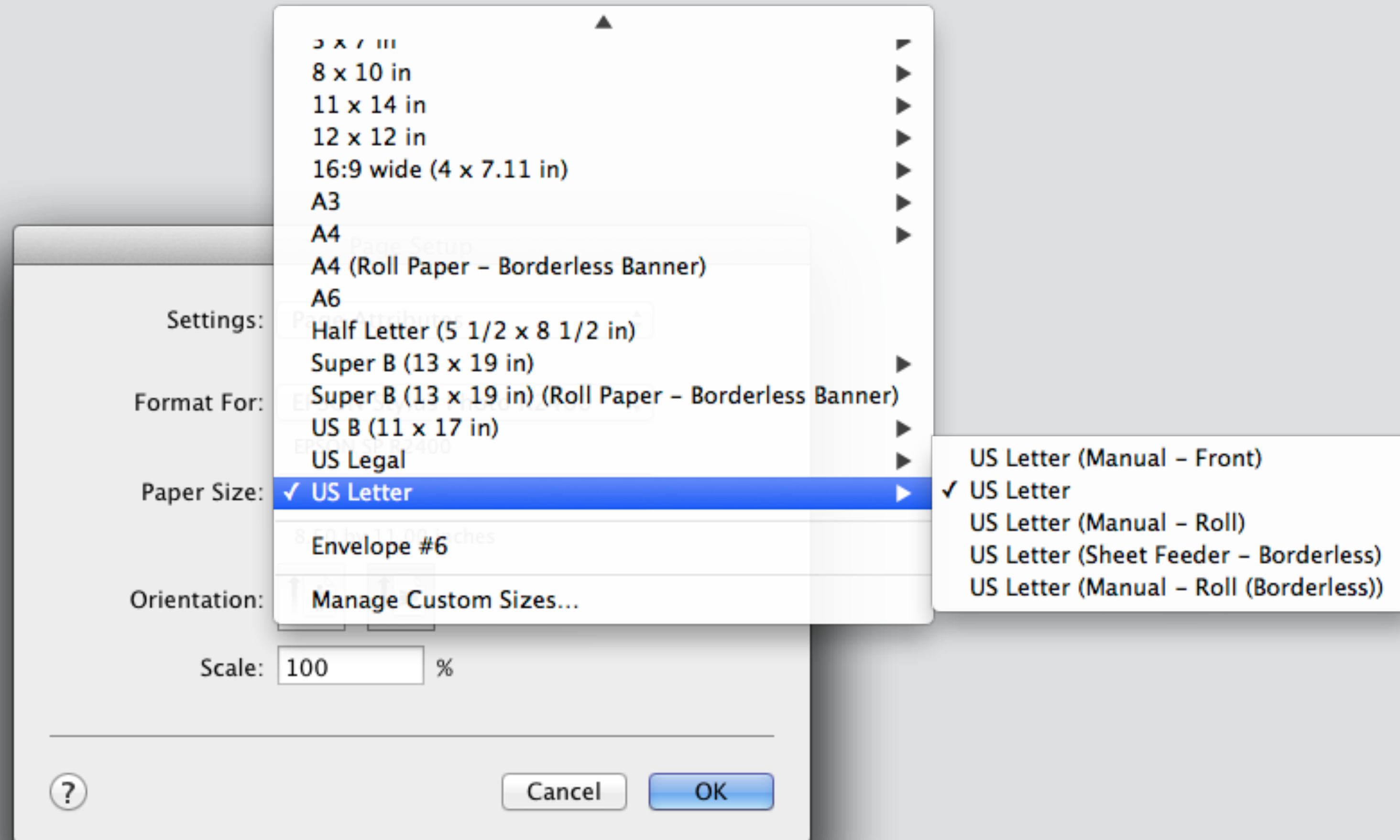
## Using labels

Labels help you organize your messages into categories – work, family, to do, read later, jokes, recipes, any category you want. Labels do all the work that folders do, but with an added bonus: you can add more than one to a message.

how Google explains labels

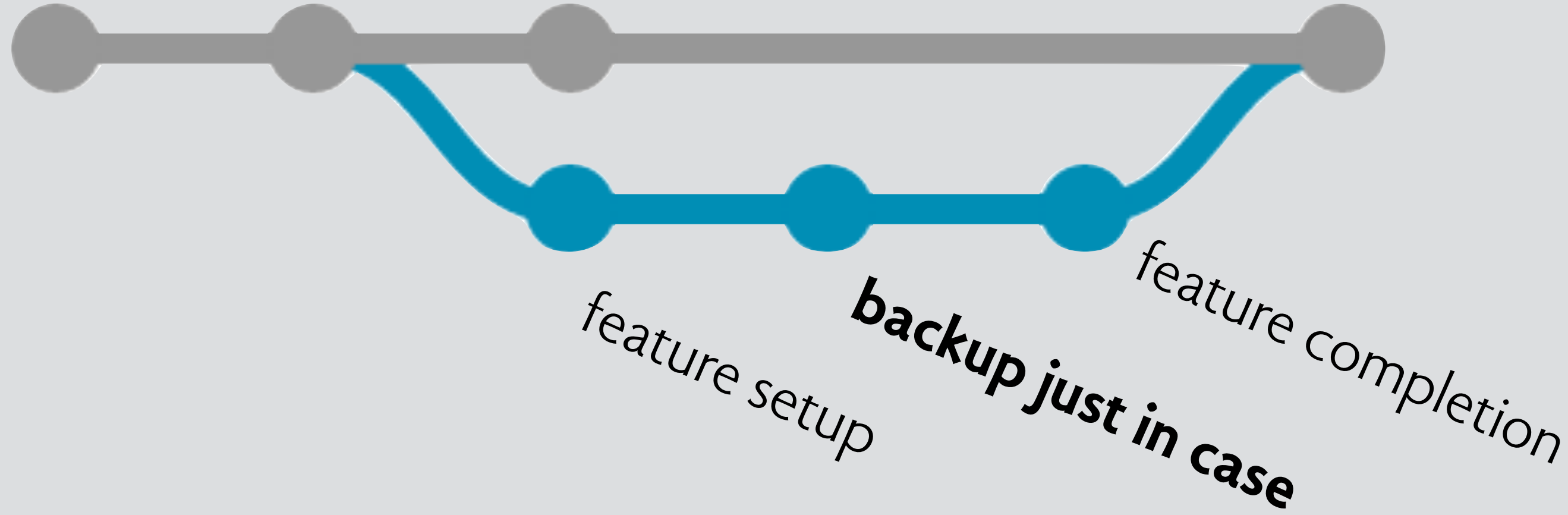


# overloading Epson driver



result: can't create custom size for front loading  
also, page size presets in Lightroom hold feed setting

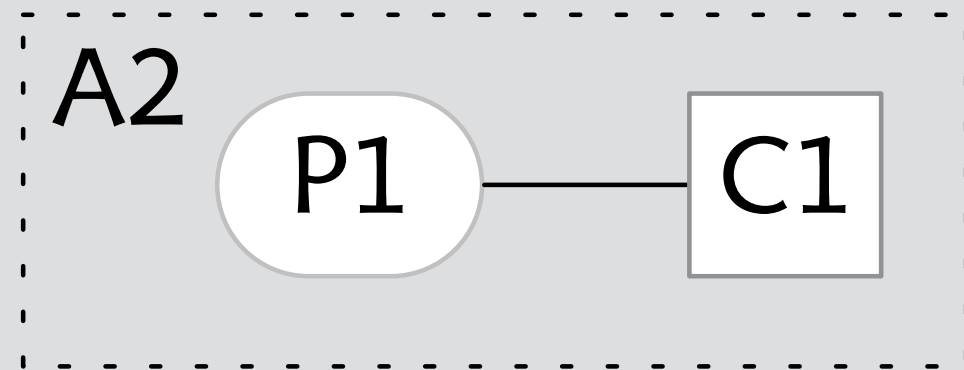
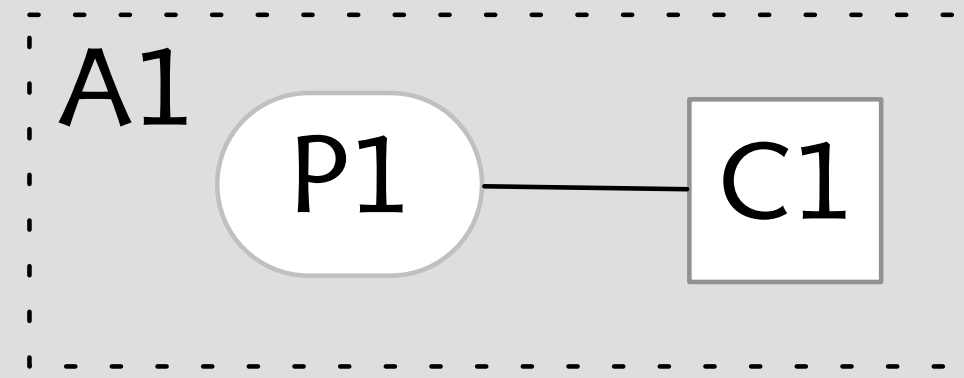
# overloading commit concept



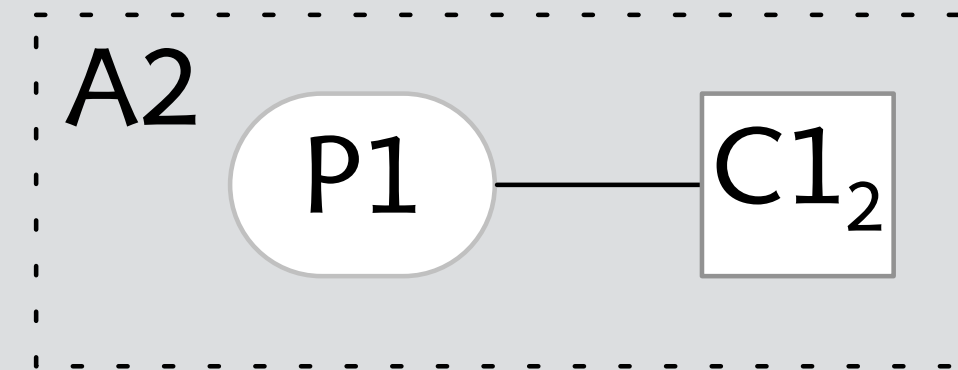
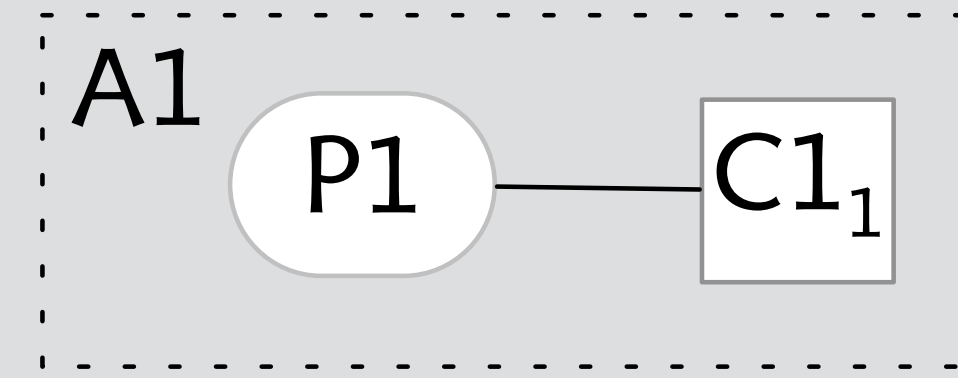
**familiarity**

# familiarity and the alternative

**familiarity**  
steal, don't invent



**needless specialization**  
custom concept, standard purpose





# the section concept in powerpoint

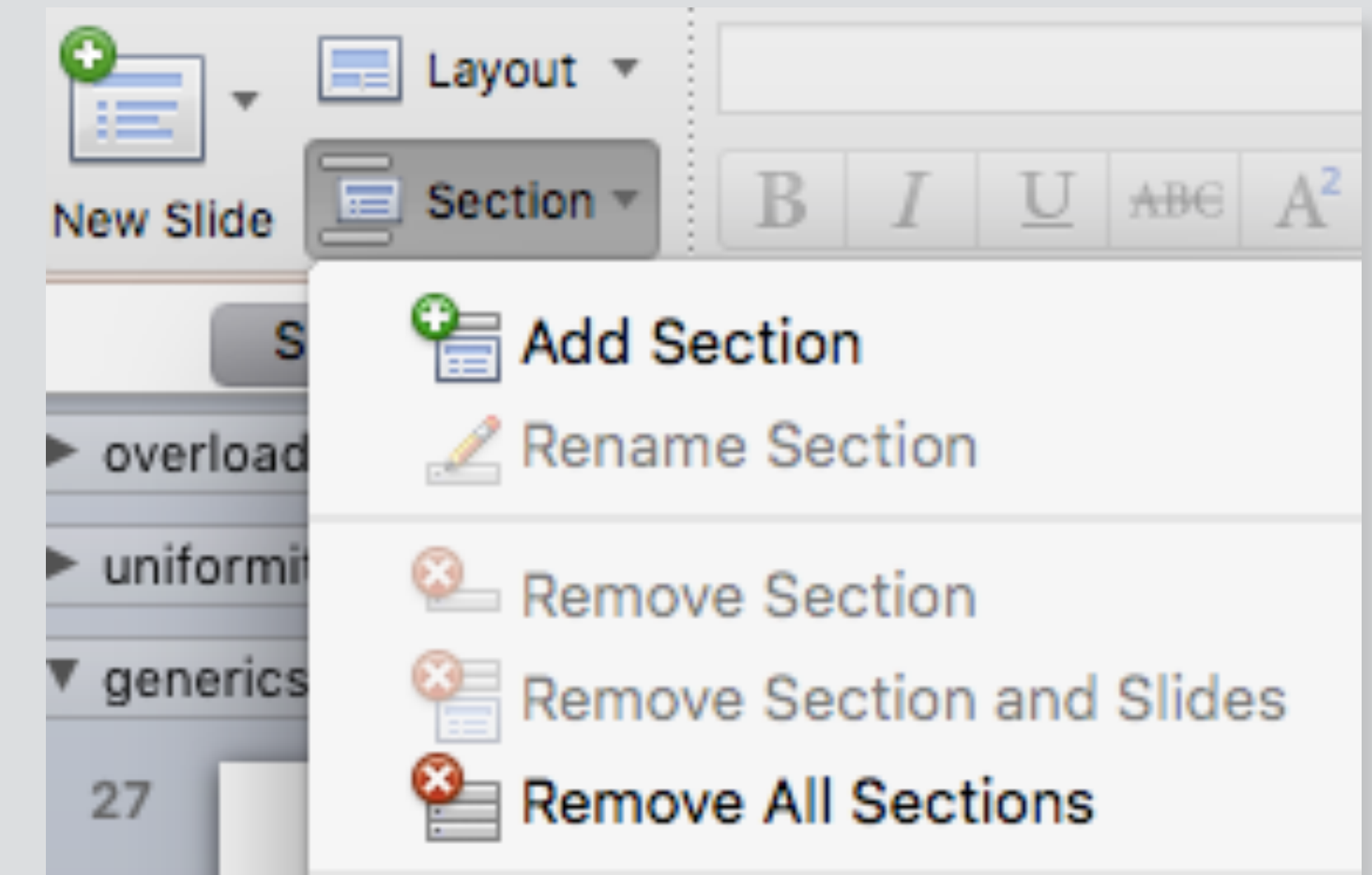
in Keynote



in Powerpoint



Powerpoint commands



# lightroom export preset

Export 57 Files

Export To: **Hard Drive**

Preset:

- ▼ Lightroom Presets
  - Burn Full-Sized JPEGs
  - Export to DNG
  - For Email
  - For Email (Hard Drive)
- User Presets

Export 57 Files

► **Export Location** Choose folder later

► **File Naming** photo.jpg

► **Video** No Video

▼ **File Settings**

Image Format: **JPEG** Quality:  50

Color Space: **sRGB**  Limit File Size To: **100** K

▼ **Image Sizing**

Resize to Fit: **Width & Height**  Don't Enlarge

W: **640** H: **640** pixels Resolution: **72** pixels per inch

▼ **Output Sharpening**

Sharpen For: **Screen** Amount: **Standard**

► **Metadata** Copyright Only

► **Watermarking** No watermark

Add Remove

**Multiple Presets:** Selected 2 Presets.  
Some sections are hidden when presets are checked [Learn more](#)

Plug-in Manager... Done Cancel **Export**

ok, highlighting selects the preset

huh, what are the checkboxes?

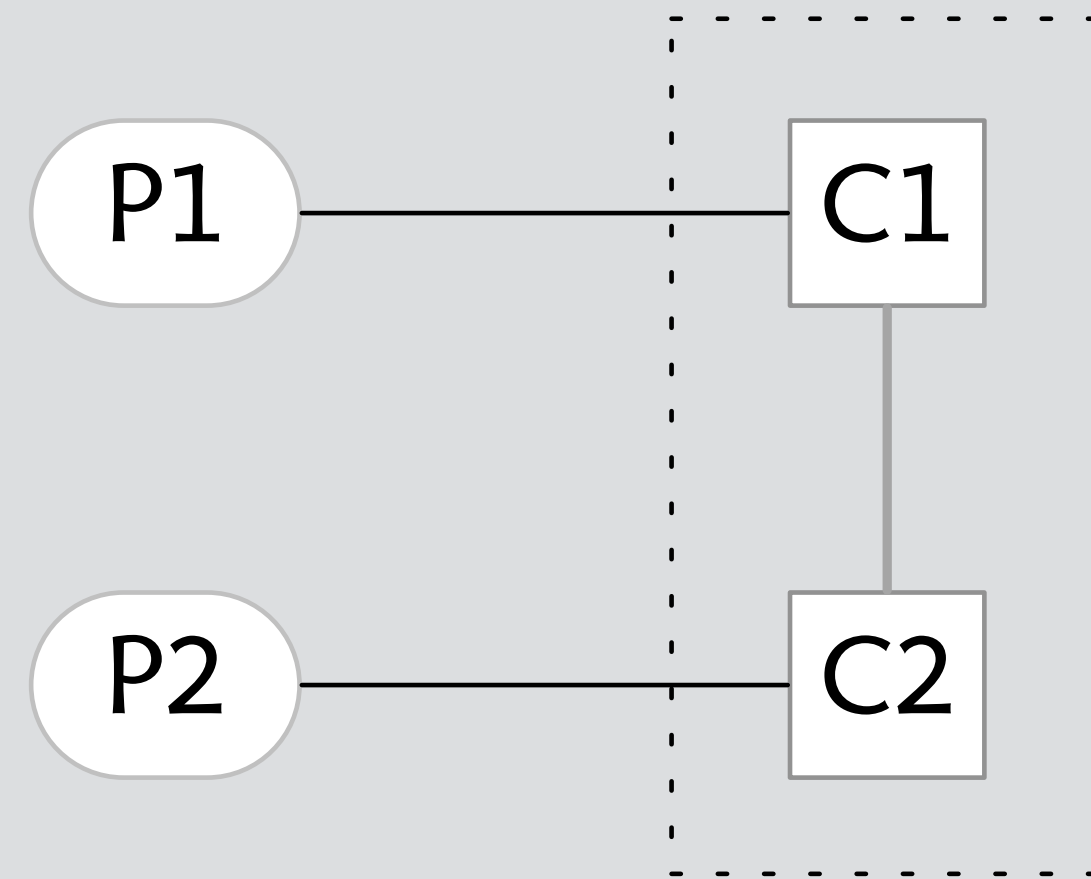
and why the warning message?



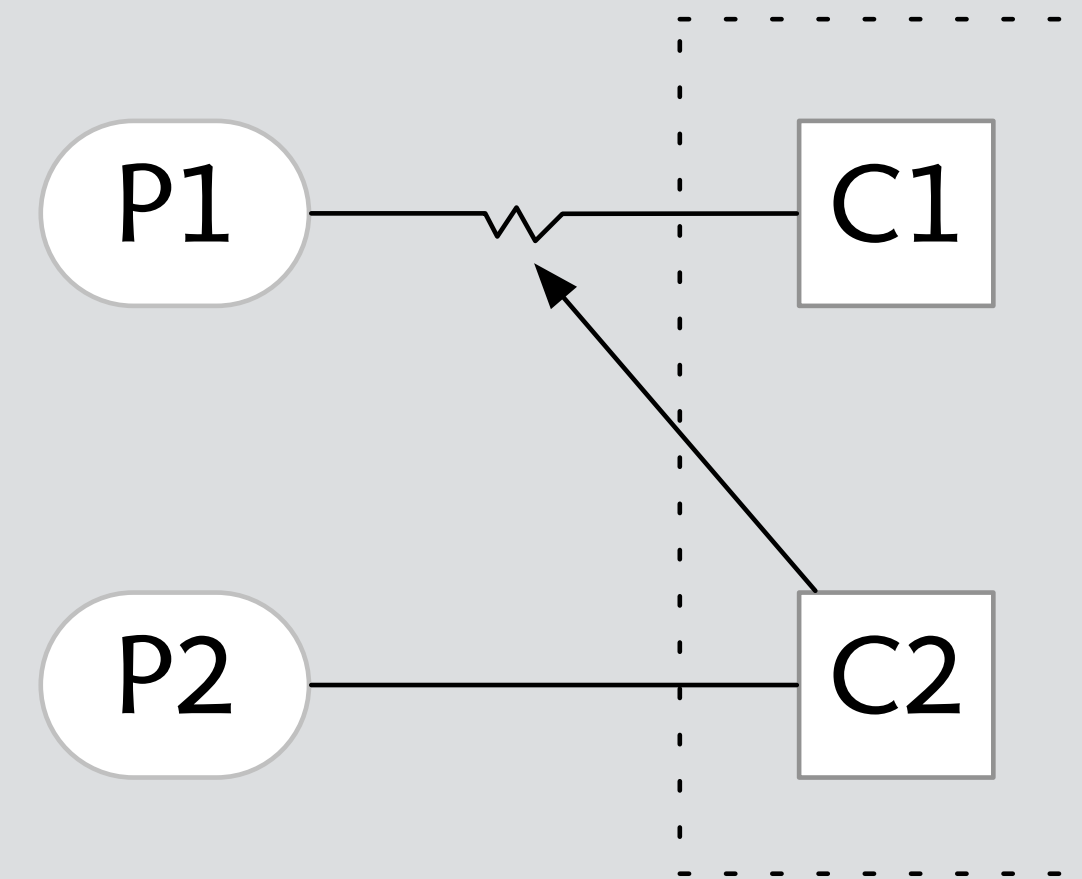
**integrity**

# integrity and its violation

**integrity**  
concepts safe when composed



**interference**  
one concept breaks another





# using labels to organize messages

The image shows a Gmail interface with several annotations. At the top left, the Gmail logo and a search bar labeled "Search mail" are visible. Below the search bar, there are icons for "Compose", "Primary", "Social", and "Promotions". The "Inbox" is highlighted in red. A message from "Alyssa, me 3" is shown with two labels: "hacking" and "meetups". The "hacking" label is pointed to by a callout box that says "a label". In the left sidebar, the "Sent" folder is pointed to by a callout box that says "also implemented as a label". Below the "Categories" section, the "hacking" label is pointed to by a callout box that says "show messages with label hacking". At the bottom of the sidebar, the "meetups" label is also visible. The bottom of the interface shows storage usage: "0 GB (0%) of 15 GB used" and links for "Terms · Privacy · Program Policies".

Compose

Search mail

Primary Social Promotions

Inbox

Starred

Snoozed

Sent

Drafts

Trash

Categories

hacking

meetups

0 GB (0%) of 15 GB used

Terms · Privacy · Program Policies

Alyssa, me 3

hacking

meetups

javascript - JavaScript makes me f

also implemented as a label

a label

show messages with label hacking

# a surprising behavior

label:hacking

1-1 of 1

me, Alyssa (12) Inbox meetups javascript - Hello again Ben 9:43 am

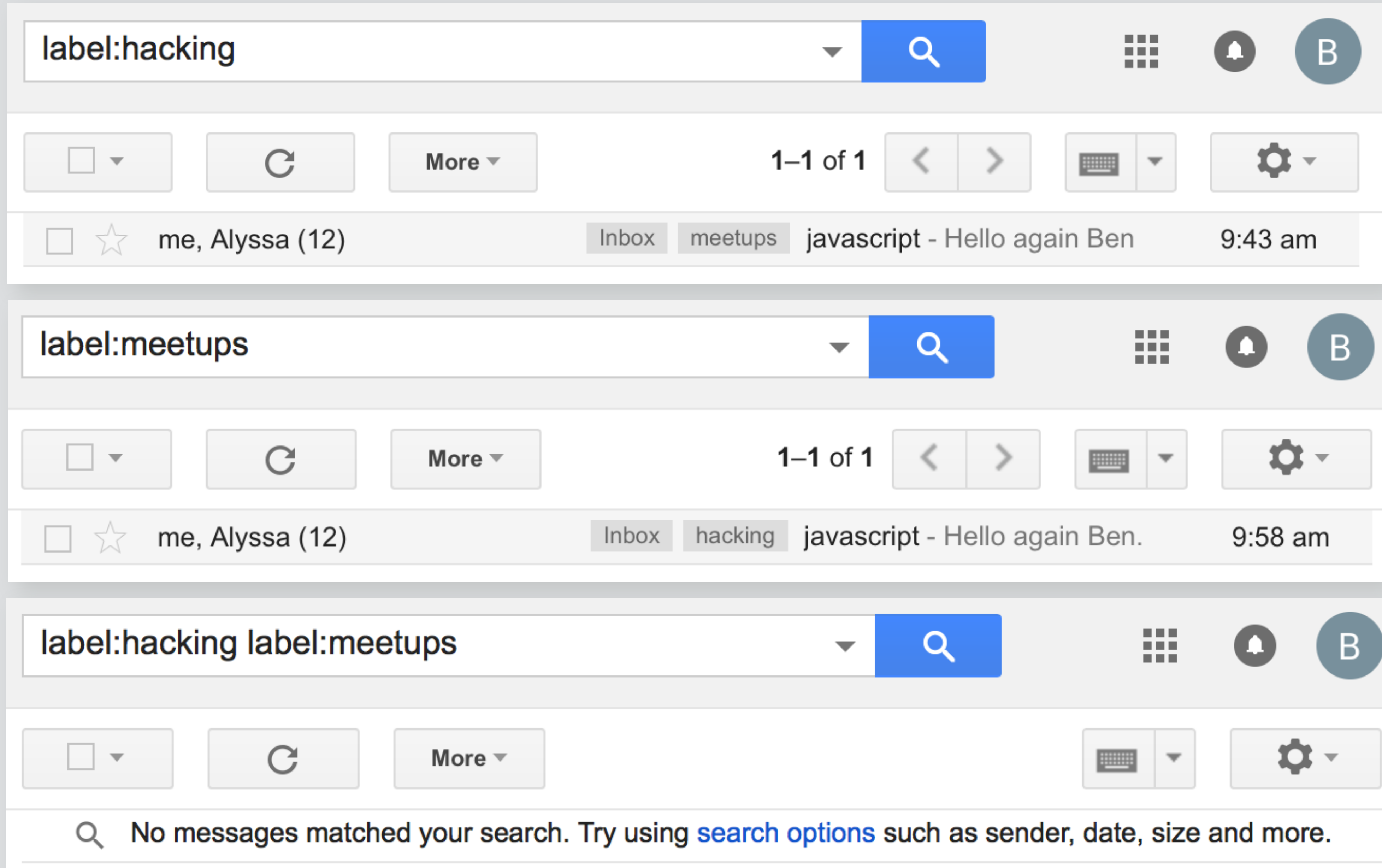
label:meetups

1-1 of 1

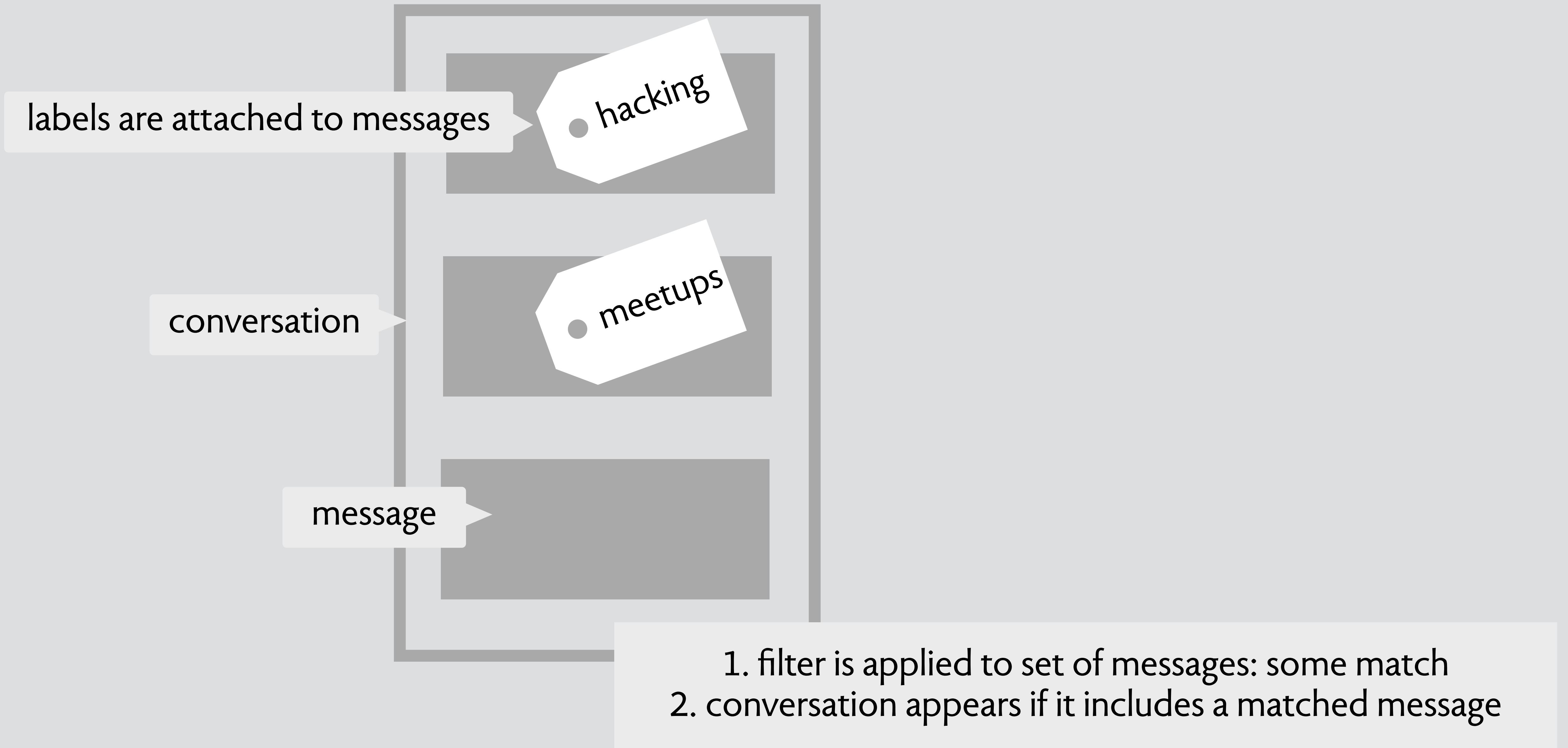
me, Alyssa (12) Inbox hacking javascript - Hello again Ben. 9:58 am

label:hacking label:meetups

No messages matched your search. Try using [search options](#) such as sender, date, size and more.



# what's going on?



so this is not a surprise

Search bar: [ ] [Q] [Grid] [Bell] [B]

Actions: [ ] [Refresh] [More] 1-1 of 1 [Left] [Right] [Keyboard] [Settings]

Primary | Social | Promotions +

☆ me, Alyssa (10) [hacking] [meetups] javascript - Hello again Be 11:48 am

Search bar: has:nouserlabels [Q] [Grid] [Bell] [B]

Actions: [ ] [Download] [Warning] [Trash] Move to Inbox [Tag] [Refresh] [More] [Left] [Right] [Keyboard]

☆ Alyssa P. Hacker [Inbox] [Promotions] buy this! - My new JS boc 10:33 am

☆ me, Alyssa (10) [Inbox] [hacking] [meetups] javascript - Oh, Al 9:24 am



and this makes sense too (but order is special)

mail.google.com

Google in:sent

Gmail

COMPOSE

Inbox  
Starred  
Sent Mail  
Drafts  
Trash

Categories

- Social
- Promotions
- Updates
- Forums

hacking  
meetups  
todo  
More

javascript Inbox x hacking x

Alyssa P. Hacker Reminds you of the old days, eh? 9:14 PM (33 minutes ago)

Ben Bitdiddle <benito.bitdiddle@gmail.com> 9:40 PM (7 minutes ago)

to Alyssa

Yes, it does.

Click here to [Reply](#) or [Forward](#)

0 GB (0%) of 15 GB used [Manage](#) [Terms](#) - [Privacy](#) Last account activity: 26 minutes ago [Details](#)



# the label concept

**concept** Label

**purpose** organize items for easy retrieval

**structure**

label: Item -> one String

**actions**

mark (i: Item, p: Label)

i.label += p

unmark (i: Item, p: Label)

i.label -= p

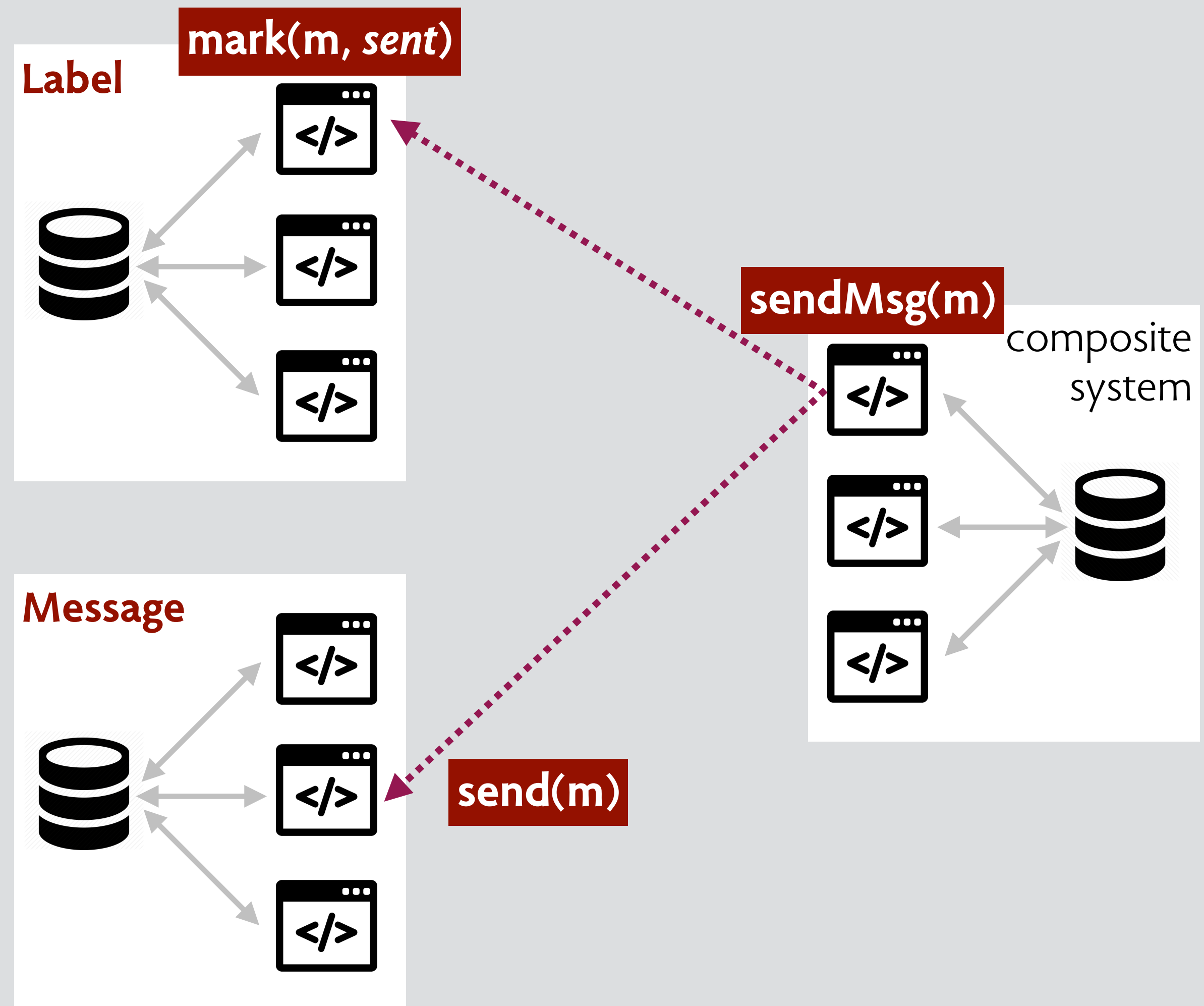
find (ps: set Label): set Item

result = {i | ps in i.labels}

**story**

if mark(i,p); find(p):is then i in is

if no mark(i,p); find(p):is then i !in is



javascript Inbox x hacking x meetups x

**Alyssa P. Hacker** <alyssa.pure.hacker@gmail.com>  
to me

Tue, May 8, 9:14 PM ☆ ↶ ⋮

Reminds you of the old days, eh?

*when message m is sent  
Label.mark(m, 'sent')  
occurs implicitly*

**Ben Bitdiddle** <benito.bitdiddle@gmail.com>  
to Alyssa

Tue, May 8, 9:40 PM ☆ ↶ ⋮

Yes, it does.

*when Sent link is clicked  
Label.find('sent'):ms  
occurs*

**Alyssa P. Hacker**  
JavaScript makes me feel nostalgic for Scheme.

Mon, Jul 30, 1:24 PM ☆

*but ms includes  
messages never marked*

**Ben Bitdiddle** <benito.bitdiddle@gmail.com>  
to Alyssa

1:15 PM (1 minute ago) ☆ ↶ ⋮

Is JavaScript just Scheme with prototypes and some hacky coercions?



why pick on gmail?



**not a strawman!**  
about 1.5B users  
20% of global market  
27% of all email opens

do these nitpicks matter?



“The details are not the details; they make the product” —Charles and Ray Eames



# trepanning: small symptoms of major surgery



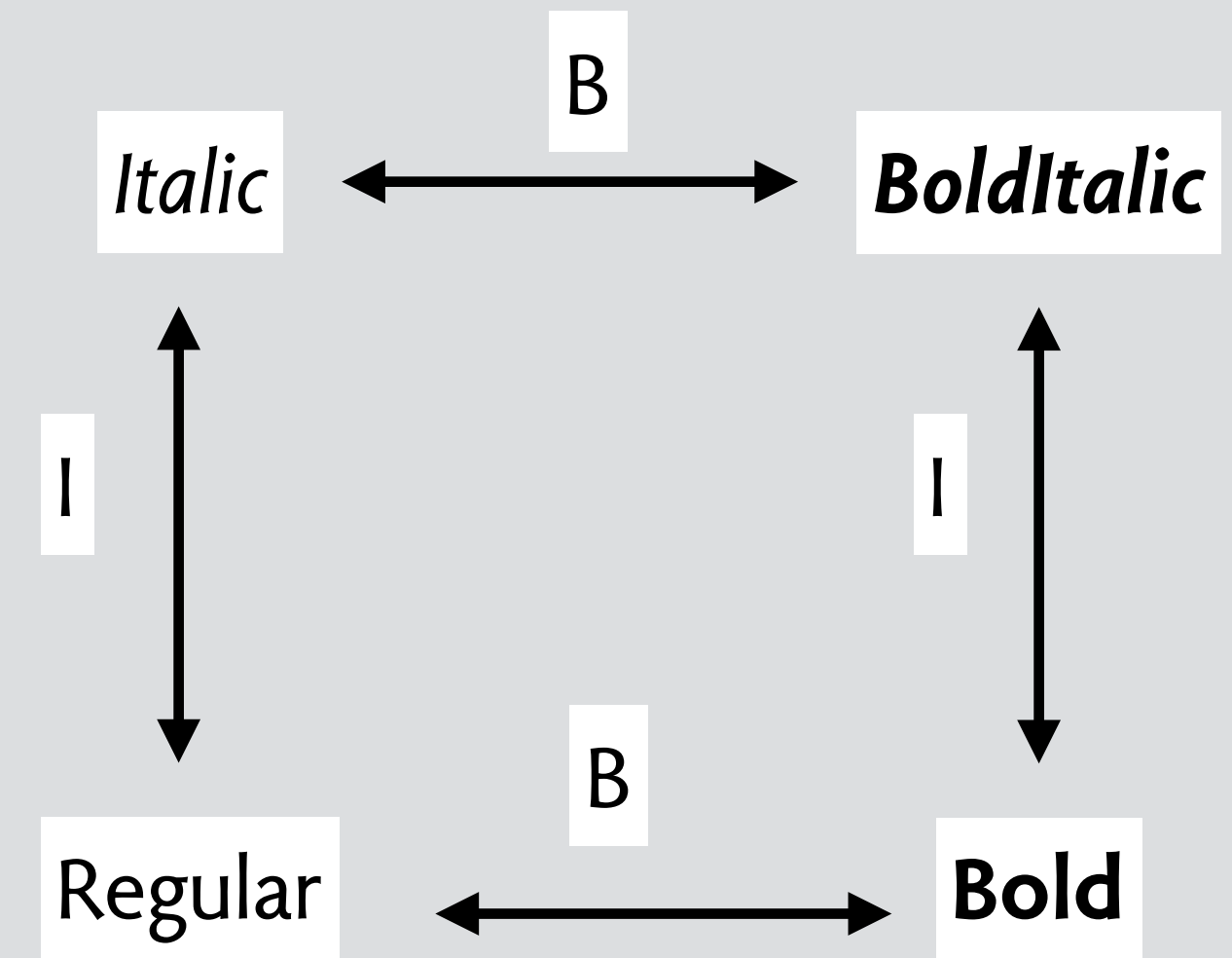
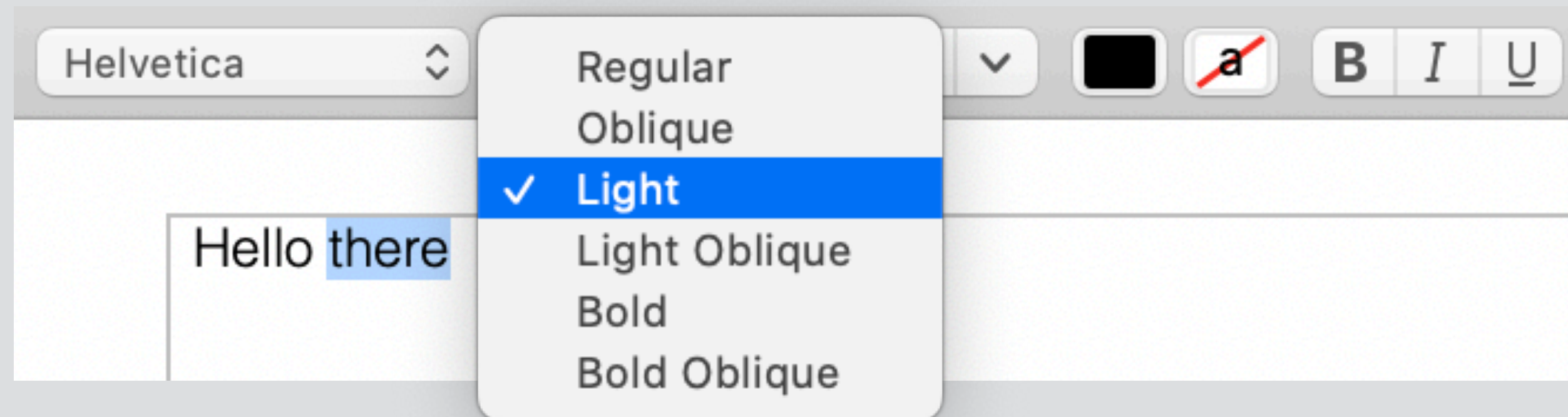
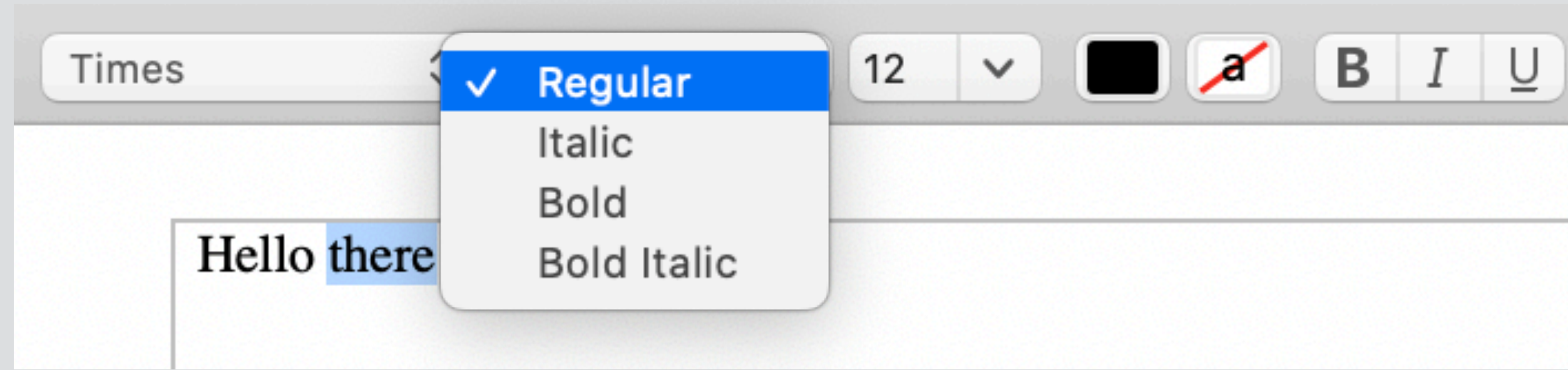
Bronze Age skull with evidence of trepanning



*The Extraction of the Stone of Madness*, Hieronymus Bosch

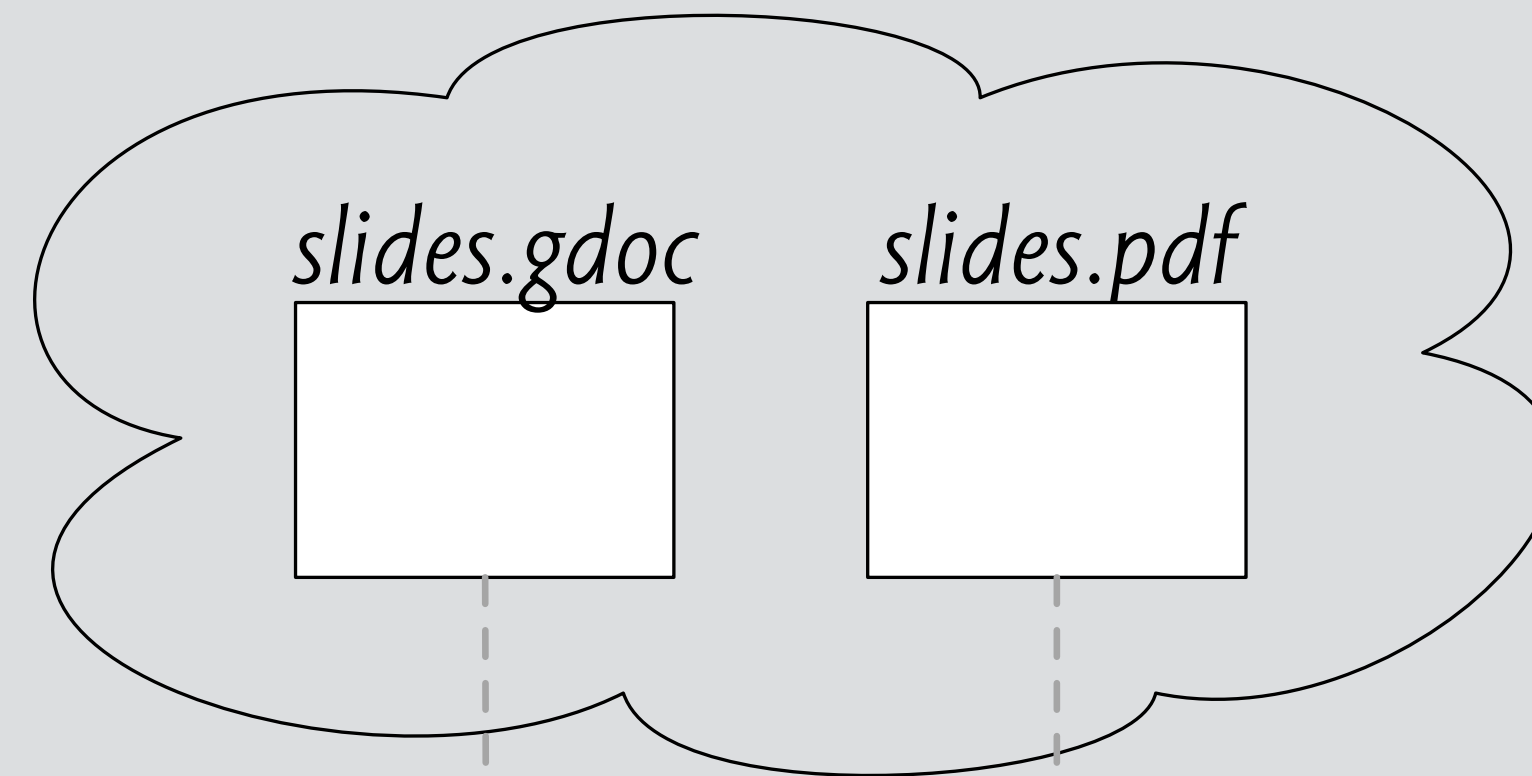


# pro fonts break integrity of format concept

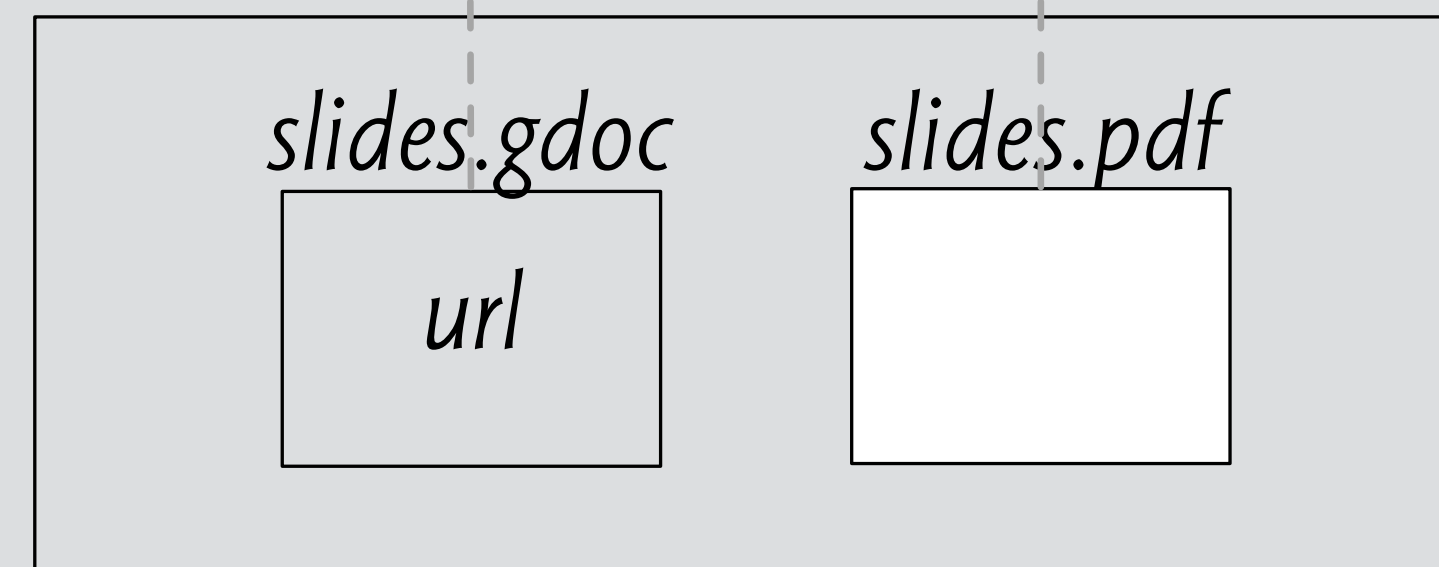


# Google CloudApp concept breaks Sync concept

*Google drive in cloud*



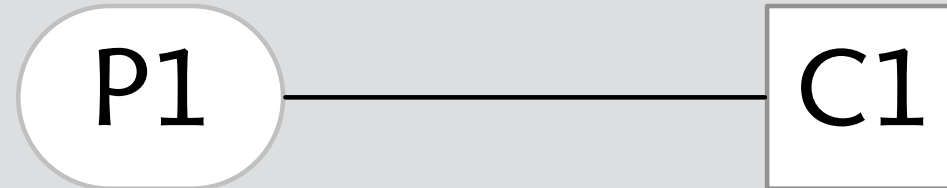
*Google drive on client machine*



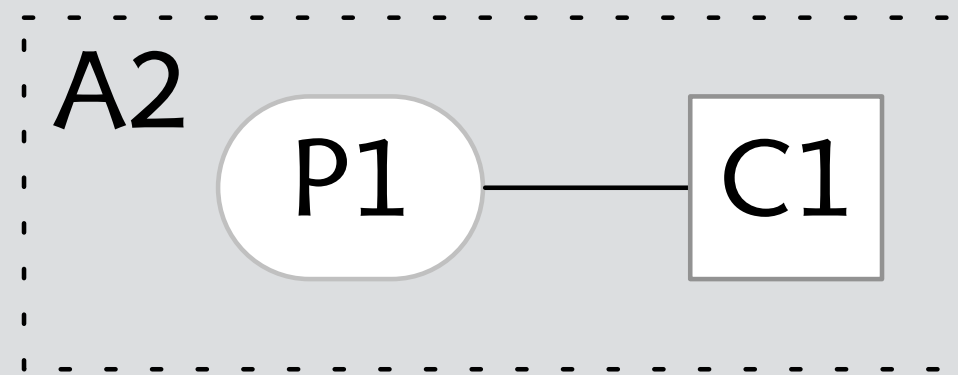
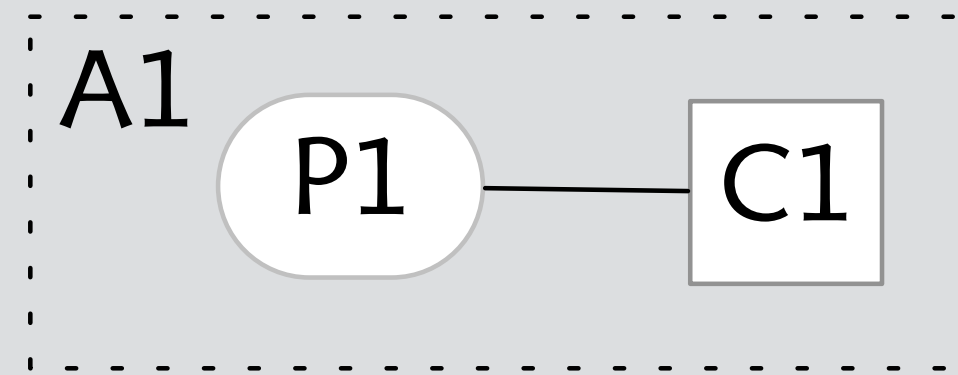


# recap: three design rules

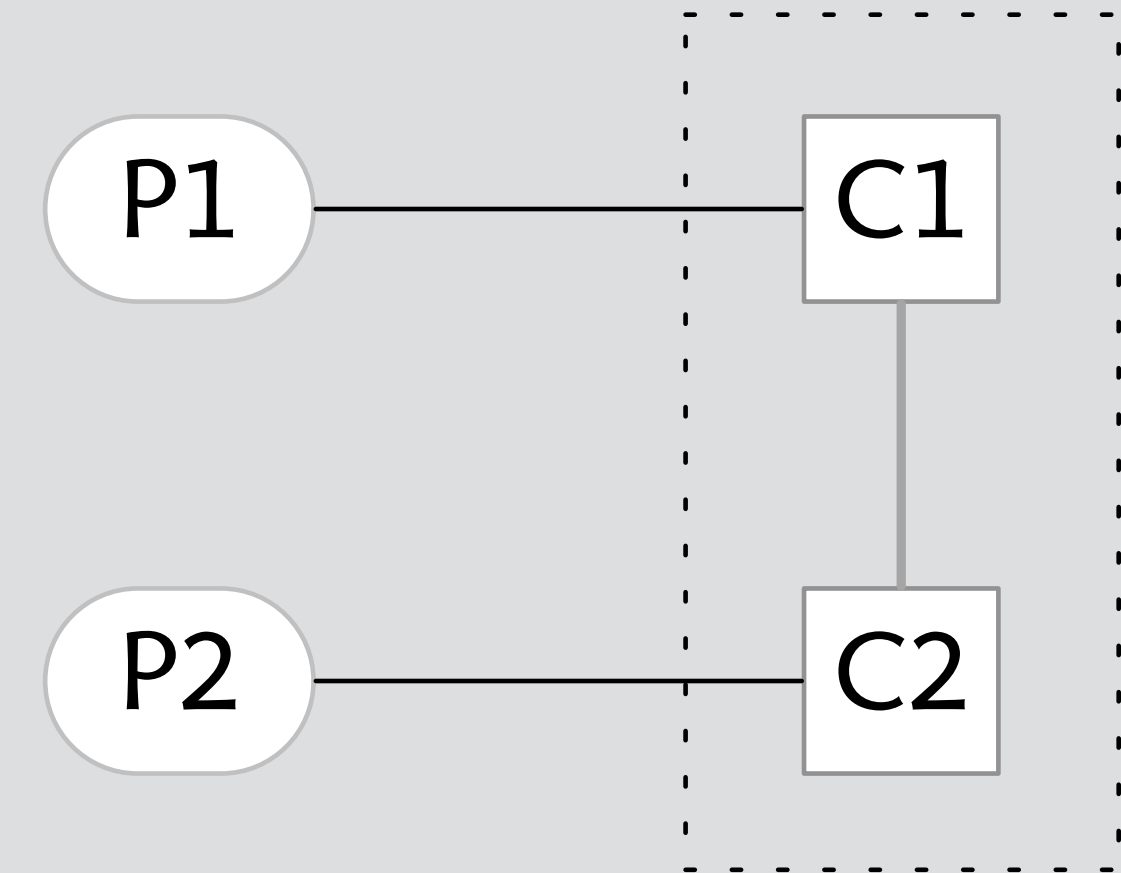
**instrumentality**  
one concept per purpose



**familiarity**  
same concept for given purpose

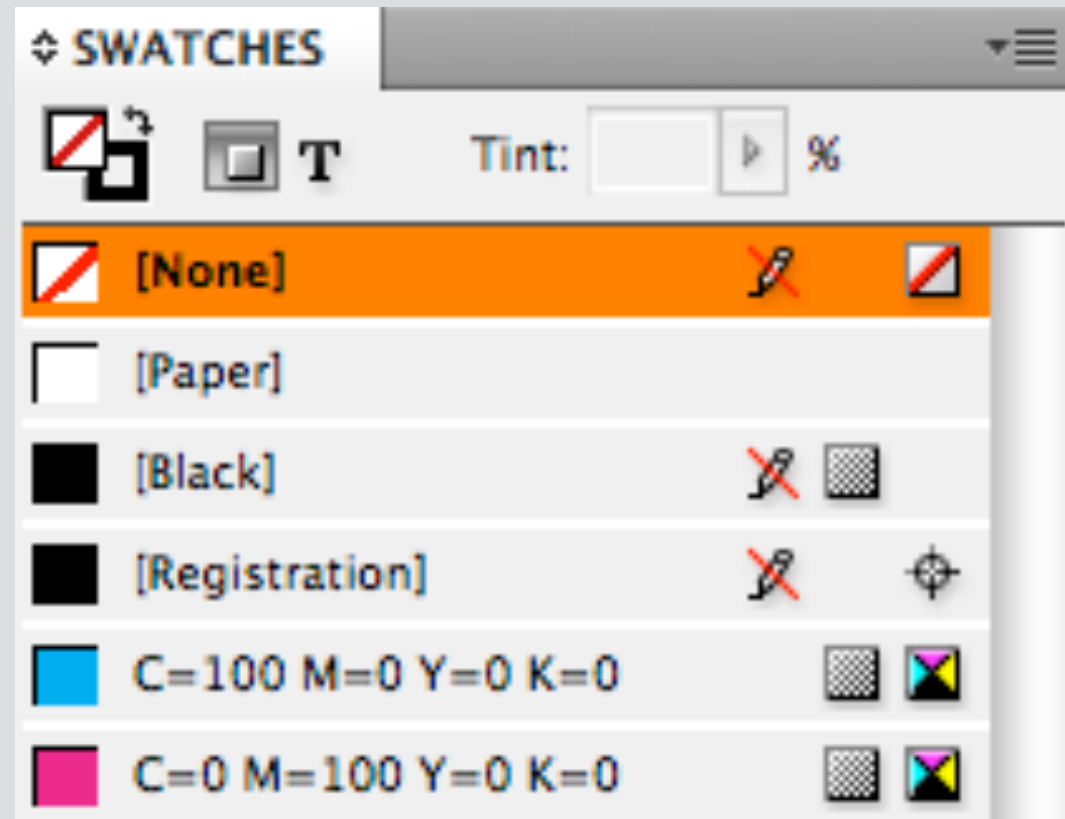


**integrity**  
composition preserves concepts

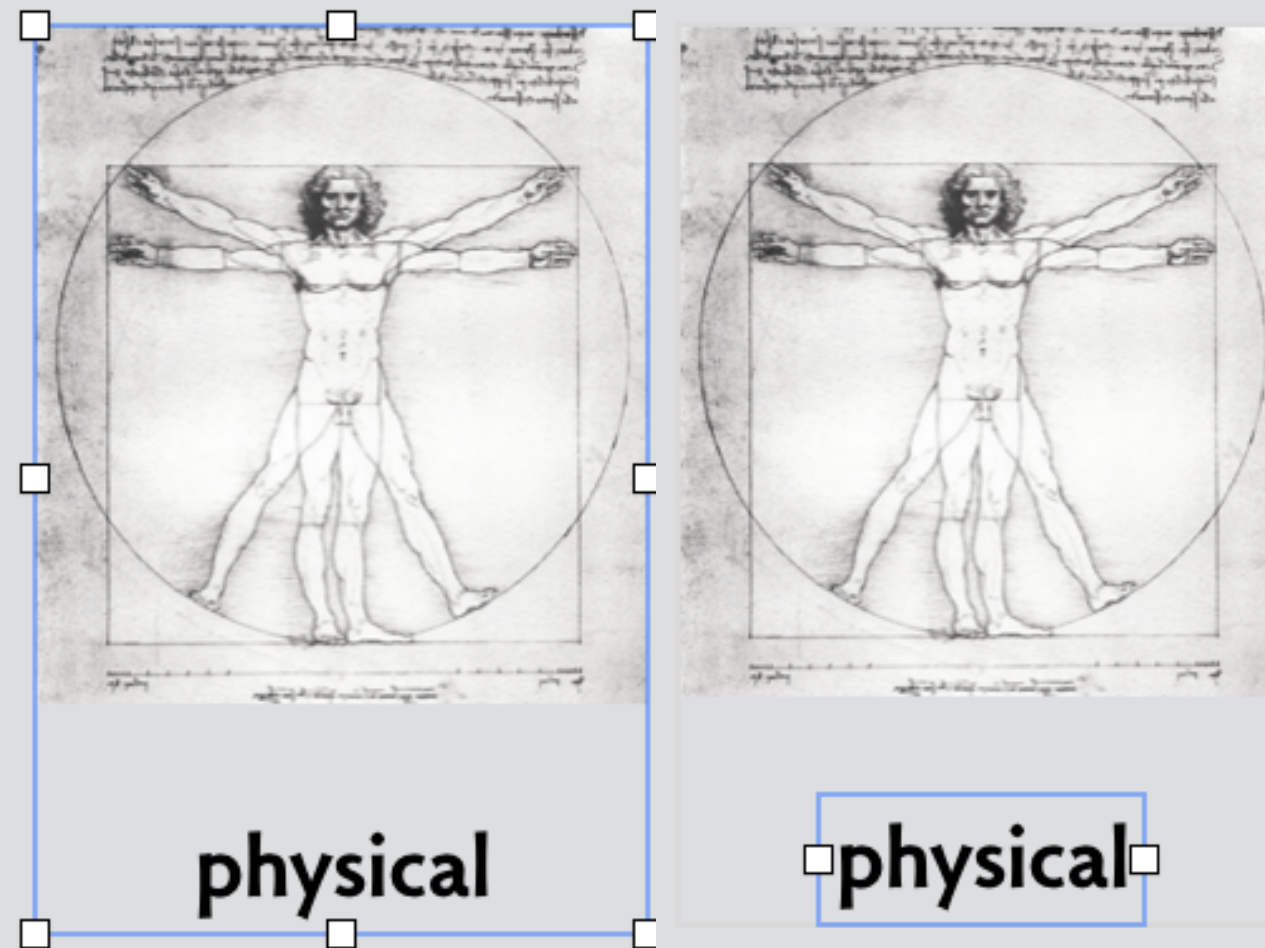


**what is design?**

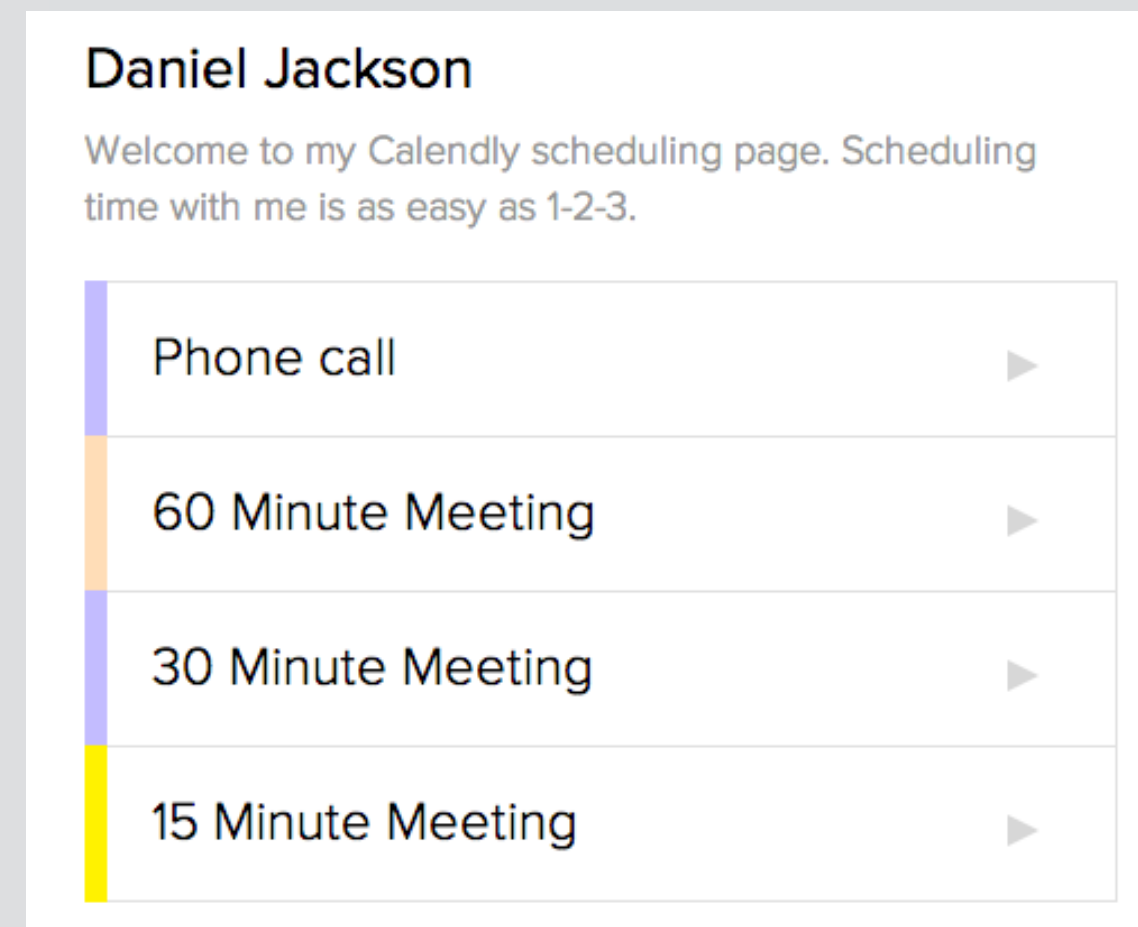
# what is design?



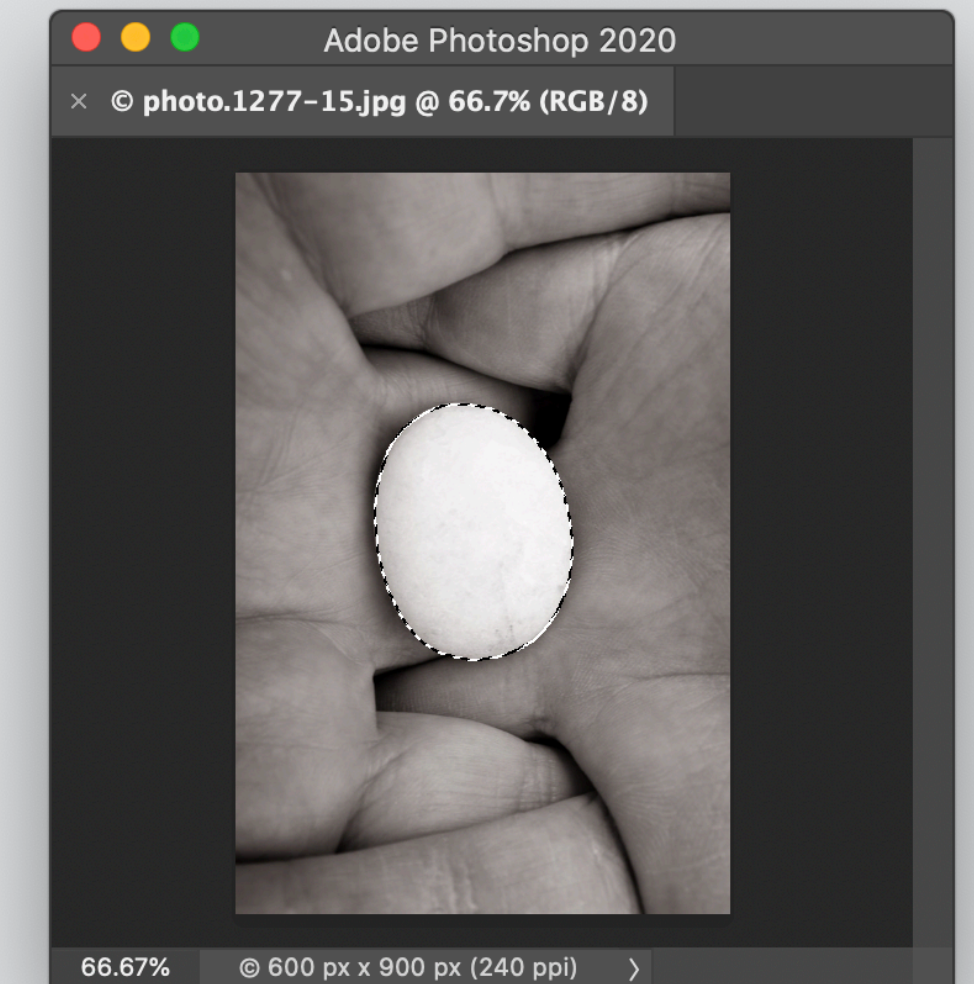
**reusing concepts**  
using Style for color swatches



**refining concepts**  
click to select Group elements



**inventing concepts**  
Event Type in Calendly



**synergy: merging concepts**  
channels in Photoshop

# the trash concept & its history

**concept** Trash

**purpose** undo deletion

**structure**

all, inTrash: **set** Object

**actions**

delete (o: Object)

empty ()

restore (o: Object)

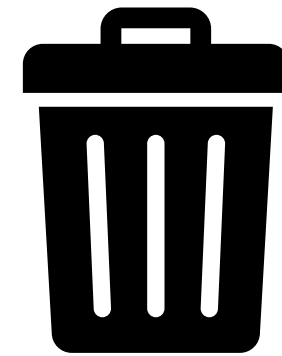
new (o: Object)

exists (o: Object, **out** b: bool)

**story**

delete(o); restore(o); exists(o, true)

delete(o); empty(); exists(o, false)



Apple Lisa (1982): "Wastebasket"

Apple Macintosh (1984): "Trash"

Microsoft MS-DOS 6 (1993): "DeleteSentry"

Apple vs. Microsoft (1994): Apple lost, but ©Trash

Windows 95 (1995): "Recycle Bin"

**holds files not folders, so can't recover structure**



# merging two concepts

**concept** Trash

**purpose** undo deletion

**structure**

all, inTrash: **set** Object

**actions**

delete (o: Object)

empty ()

restore (o: Object)

new (o: Object)

exists (o: Object, **out** b: bool)

**story**

delete(o); restore(o); exists(o, true)

delete(o); empty(); exists(o, false)



**concept** Folder

**purpose** local organization

**structure**

root: Folder

contents: Folder -> **set** (Folder + Object)

**actions**

move (o: Object + Folder, to: Folder)

new (p: Folder, **out** f: Folder)

list (f: Folder, **out** os: **set** Object)

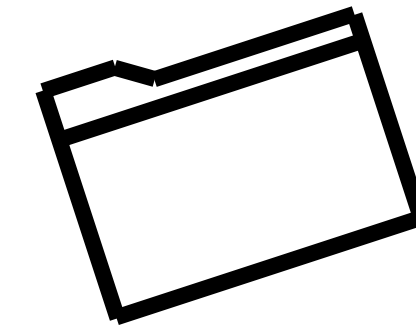
delete (f: Folder)

root (**out** f: Folder)

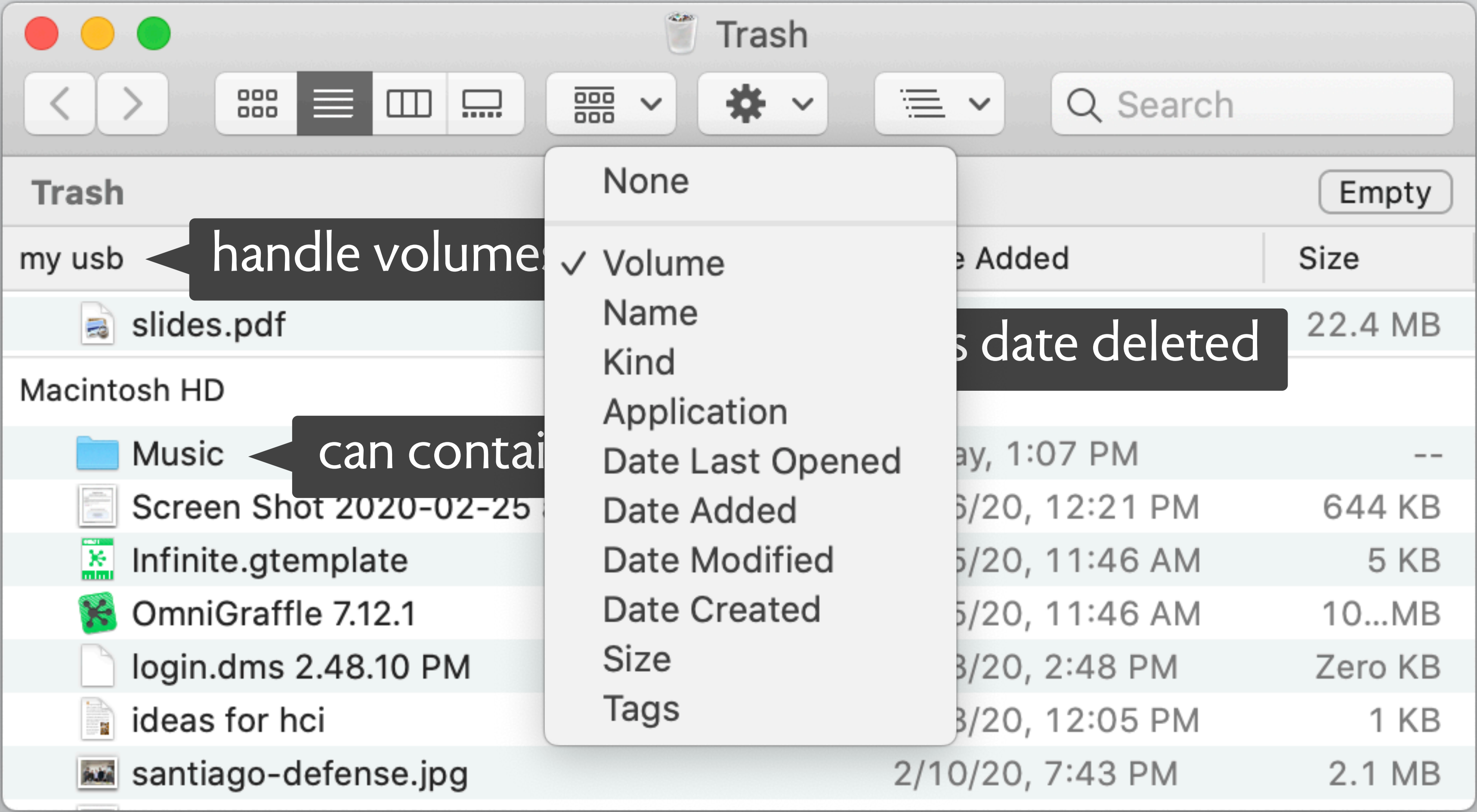
**story**

list(f, os); move(o, to); list(f, os')

=> **if** o **not in** os **and** to **!=** f **then** os = os'



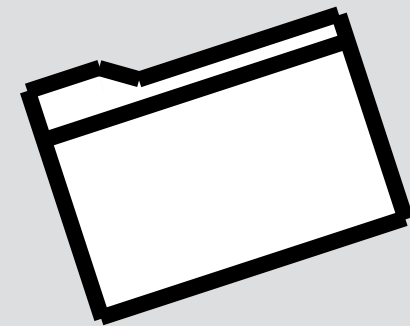
# trash x folder



# trash x folder



**purpose: undo deletion**



**purpose: local organization**

## **synergies**

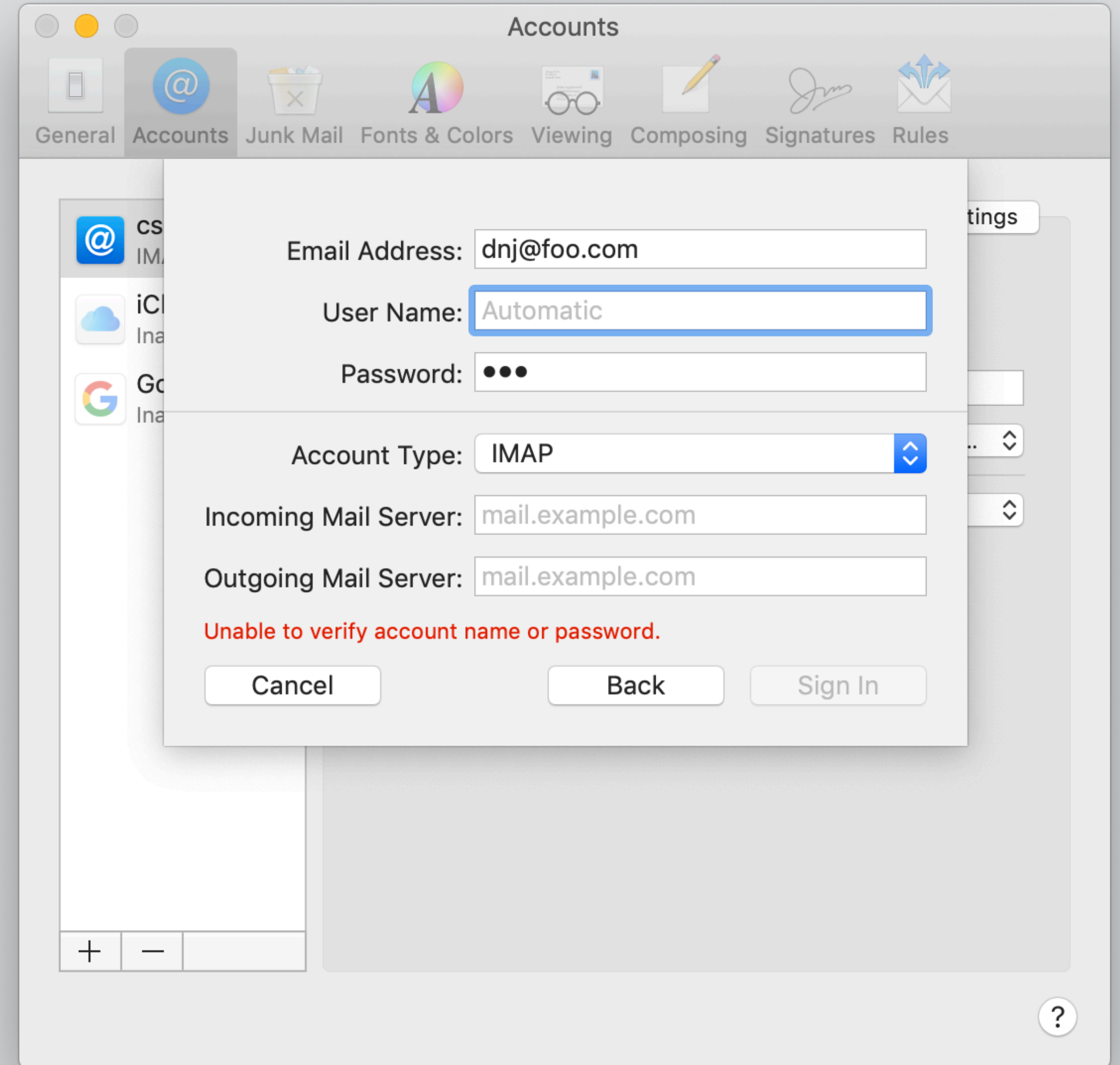
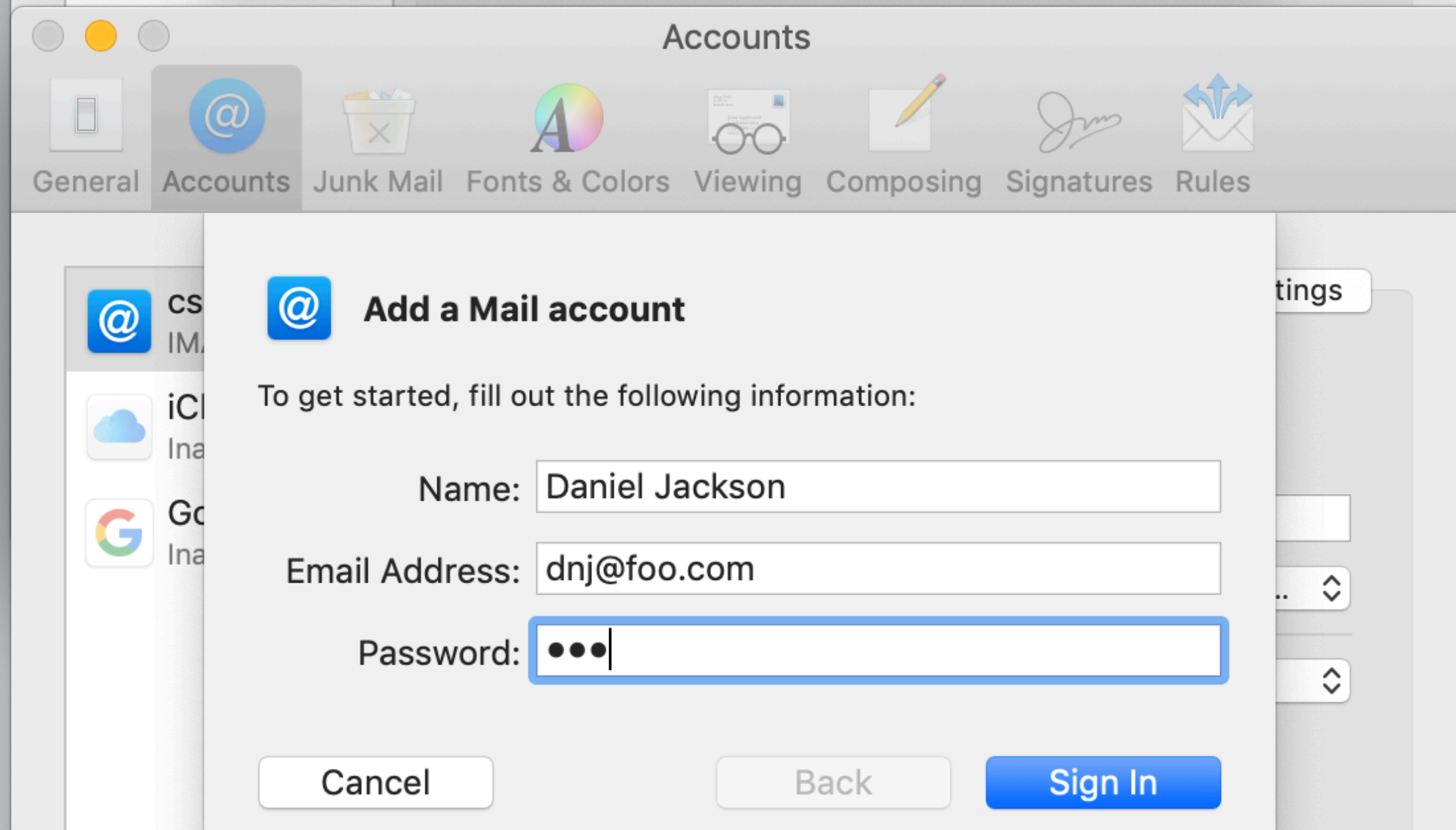
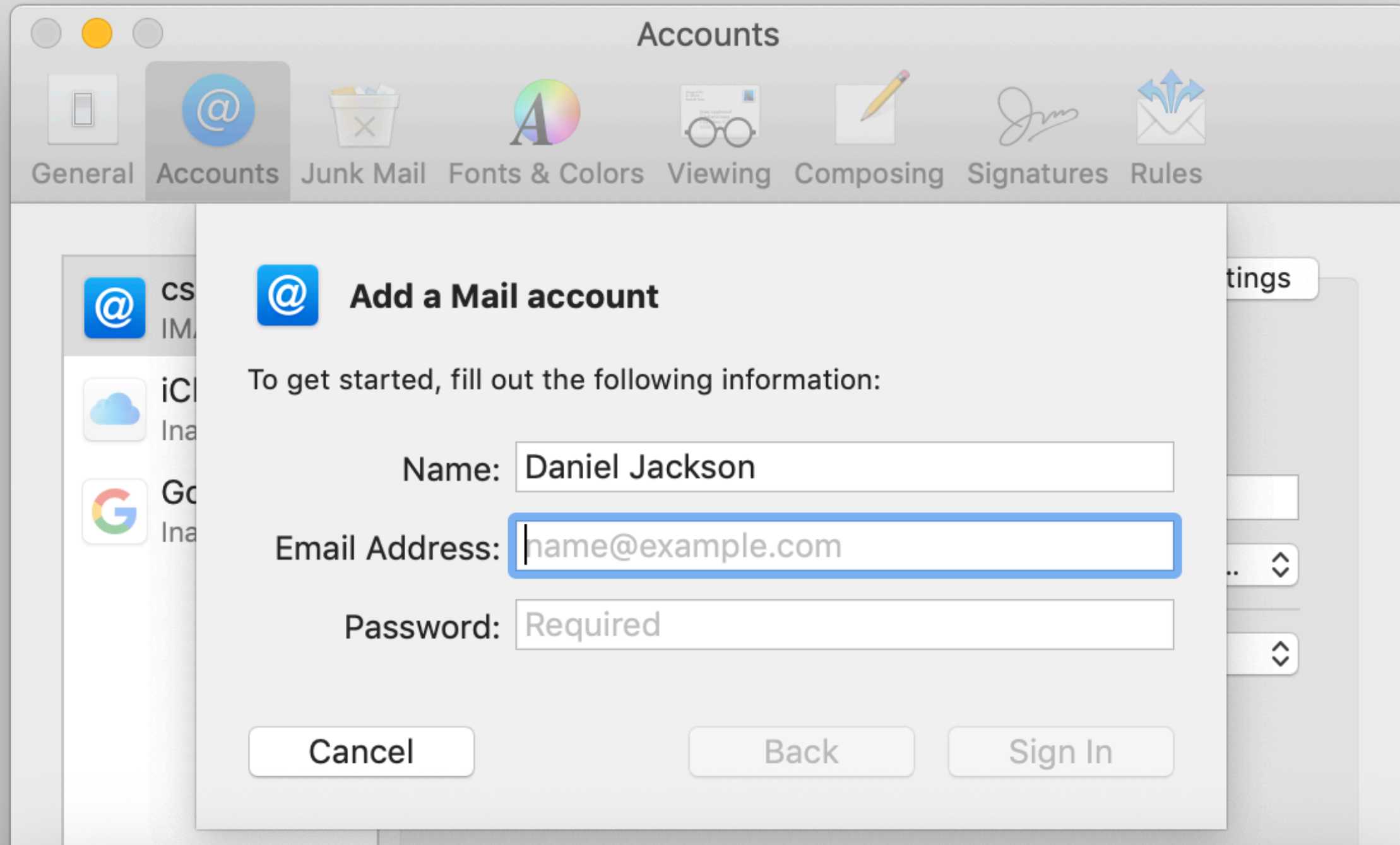
trash is not a special thing  
all folder tools apply  
can put folder in trash  
move to trash = delete  
move from trash = restore  
date added = date deleted

## **anomalies**

trash contains objects from >1 volume  
in trash folder, can group by volume  
delete immediately allows partial emptying  
trash folder has no path (path concept)  
can't move trash folder or delete it



# email x server account



# style/toc synergy

Table of Contents

TOC Style: [Default] ▾

Title: Contents

Style: [No Paragraph Style] ▾

OK

Cancel

Save Style...

More Options

Styles in Table of Contents

Include Paragraph Styles:

pattern	<< Add	Other Styles:
section		
chapter		
appendix		

Remove >>

[No Paragraph Style]

abstract

acknowledgments

after

Style: appendix

Entry Style: toc-chapter ▾

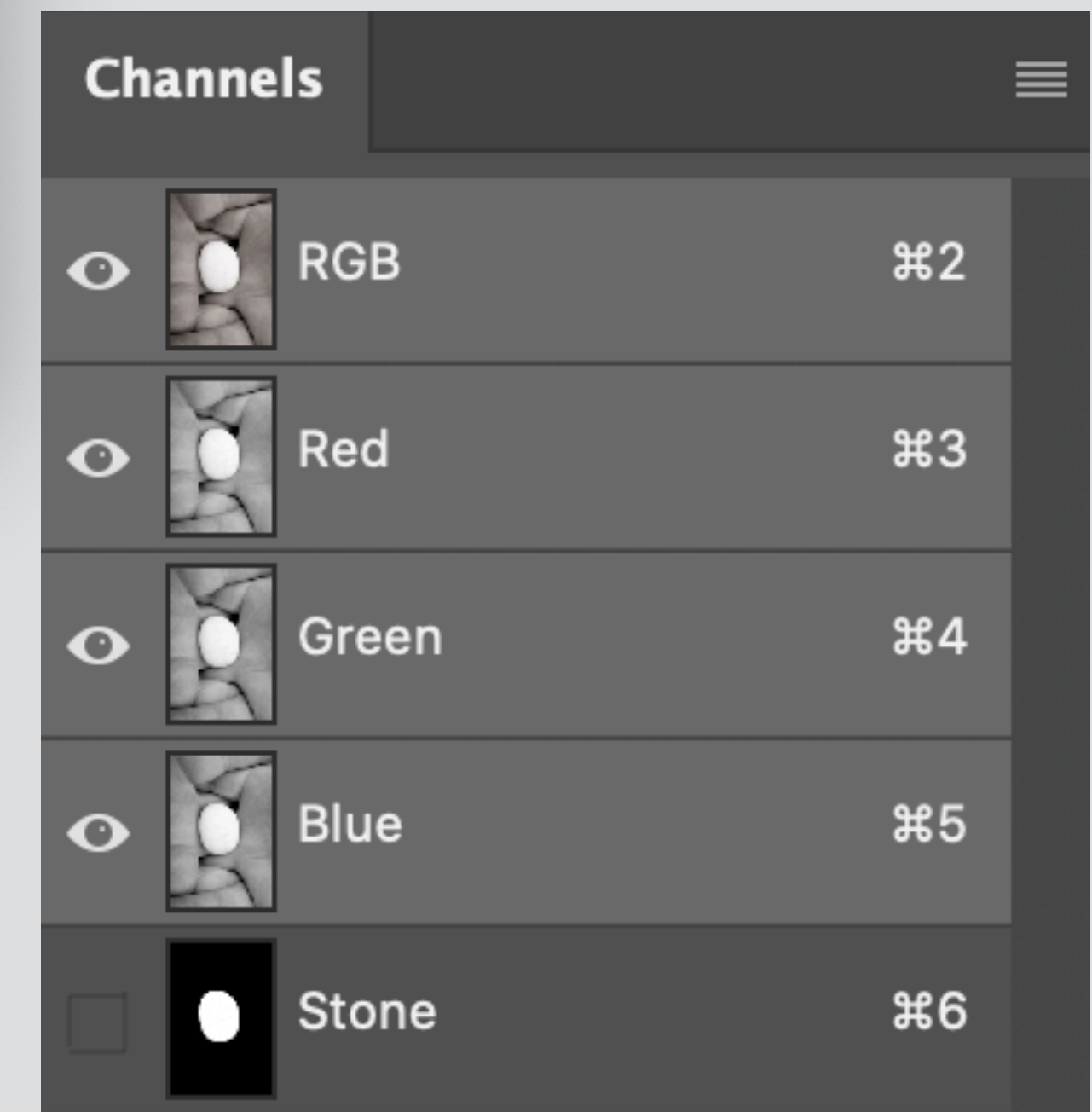
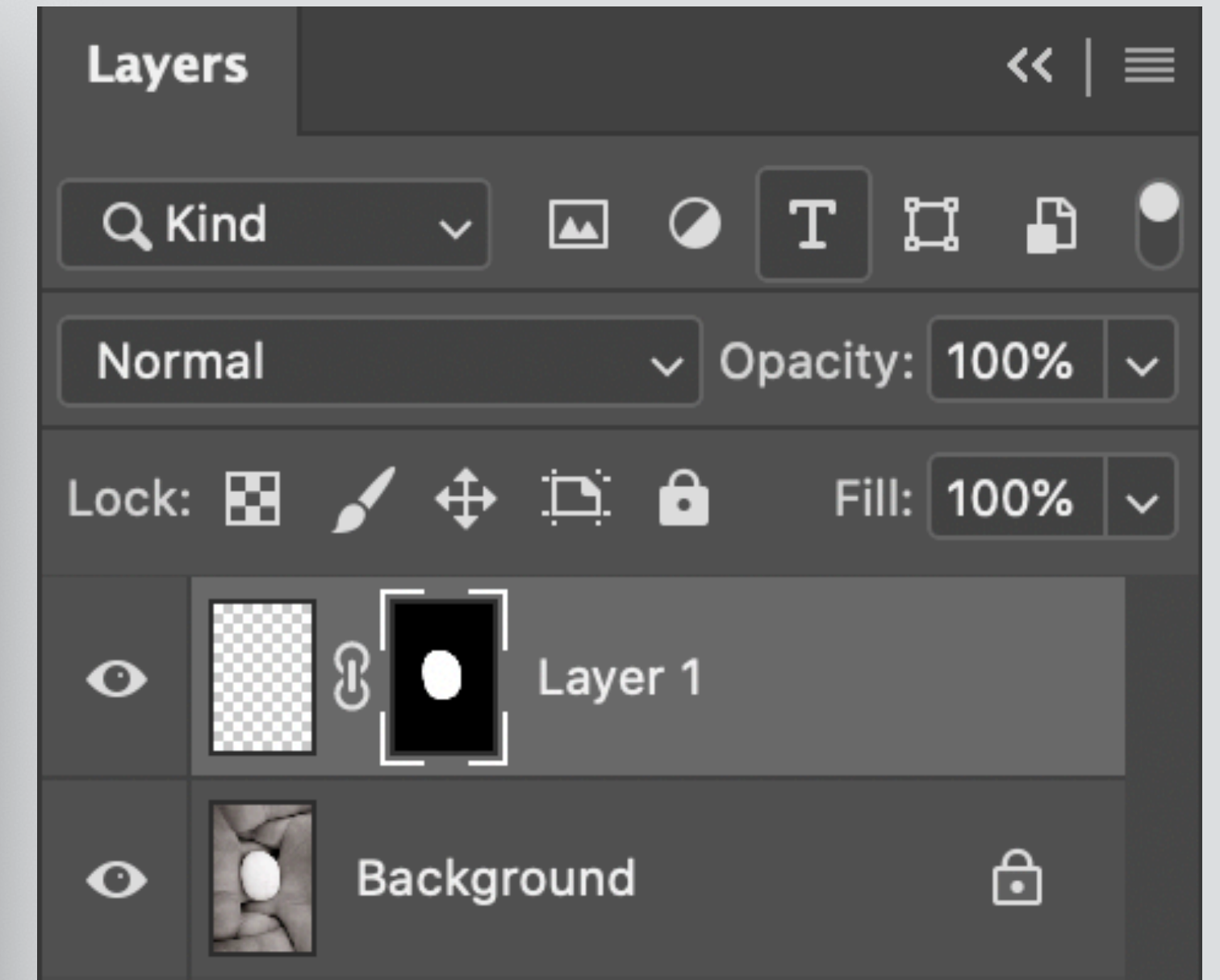
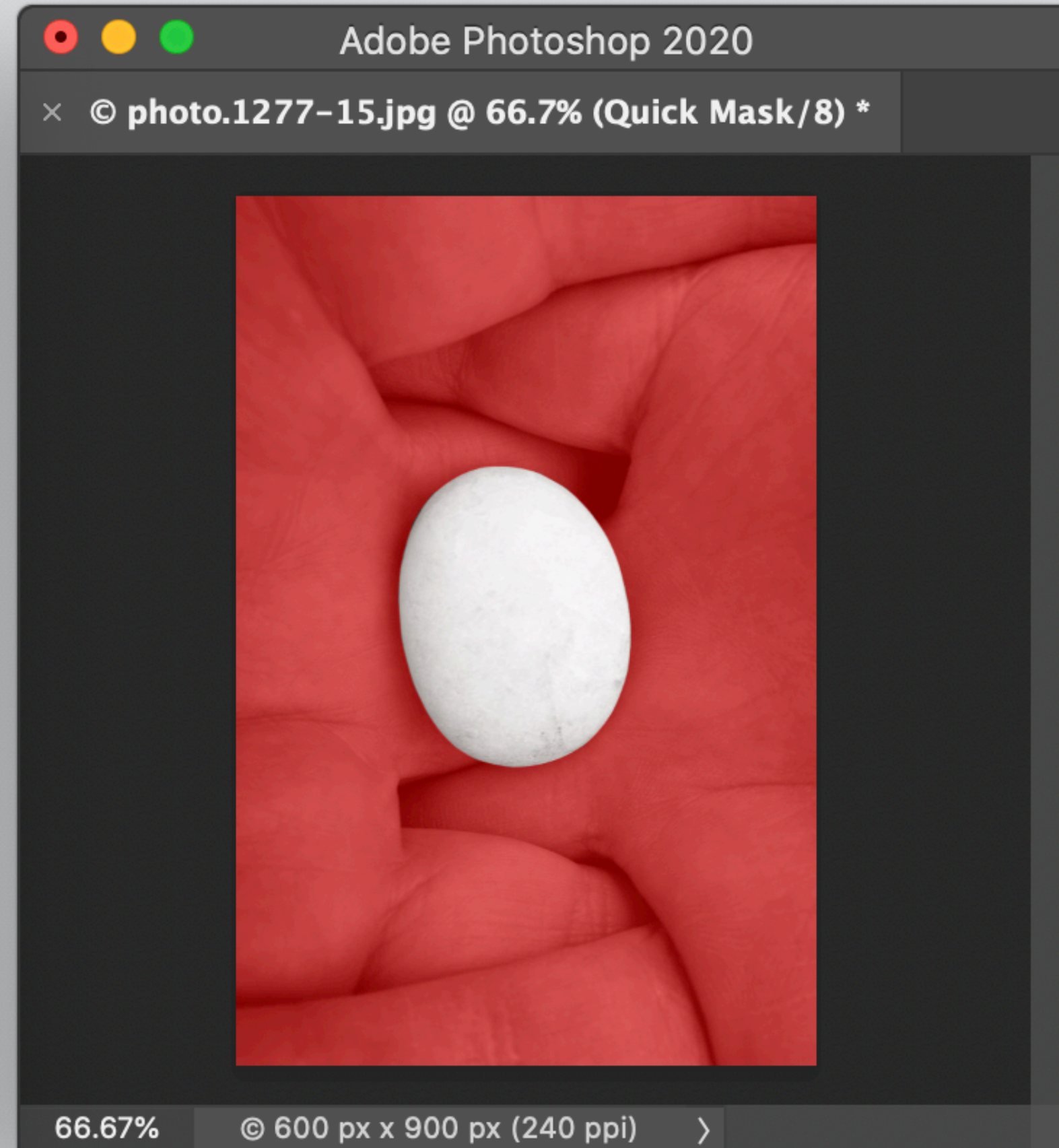
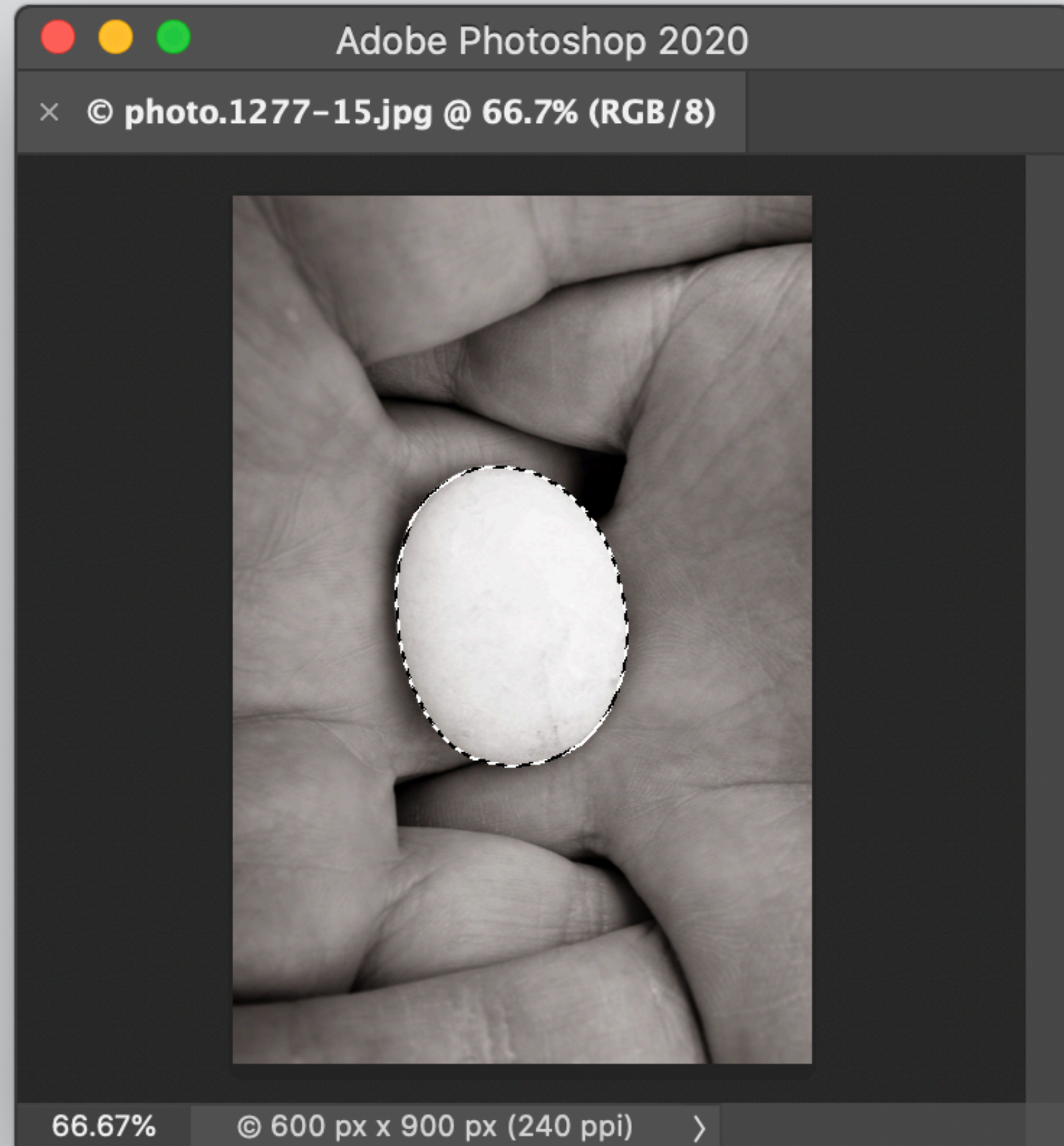
Options

- Create PDF Bookmarks
- Replace Existing Table of Contents
- Include Book Documents
- Make text anchor in source paragraph
- Remove Forced Line Break

Numbered Paragraphs: Exclude Numbers ▾



# photoshop synergies



selection = mask = channel = image



# the crazy power of photoshop

## how to sharpen an image using an edge mask

select channel with greatest contrast

duplicate selected channel

apply Filter > Stylize > Find Edges

treat channel as image

apply Image > Adjustments > Invert

apply Filter > Other > Maximum

apply Filter > Noise > Median

apply Image > Adjustment > Levels

apply Filter > Blur > Gaussian Blur

right-click to make channel a selection

make selection from channel

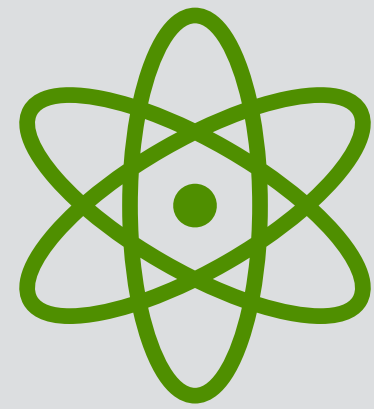
select image layer

apply Select > Inverse

apply Filter > Sharpen > Unsharp Mask

apply filter using selection as mask

# a research & teaching program



a design theory

*concept semantics  
& design rules  
[Onward 15],  
book to appear*



design case studies

*Gitless  
[Perez De Rosso,  
Onward 13,  
OOPSLA 16]*



concept handbook

*just sketches so far*



code platform

*Deja Vu  
[Perez De Rosso]*