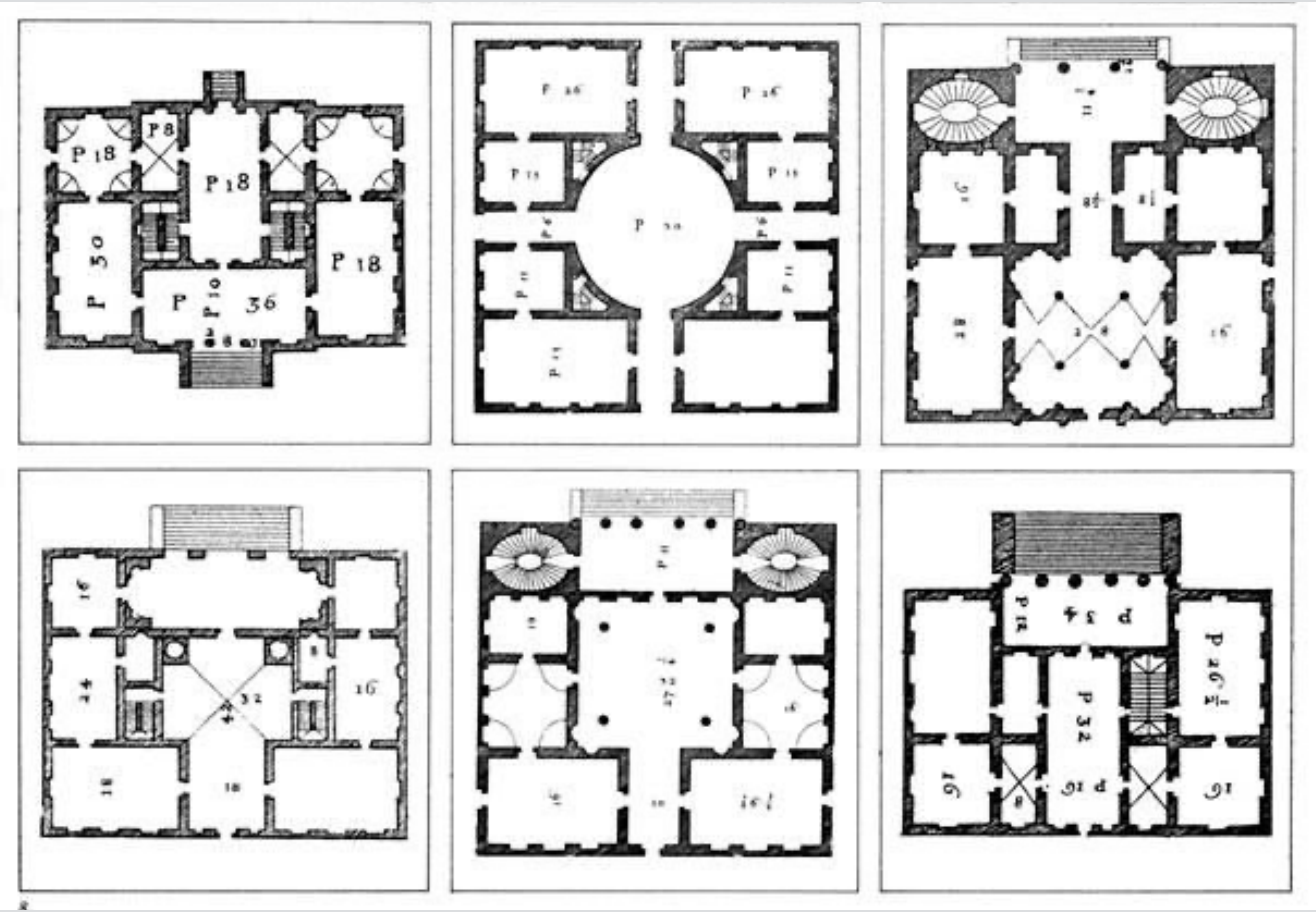
finding structure in software

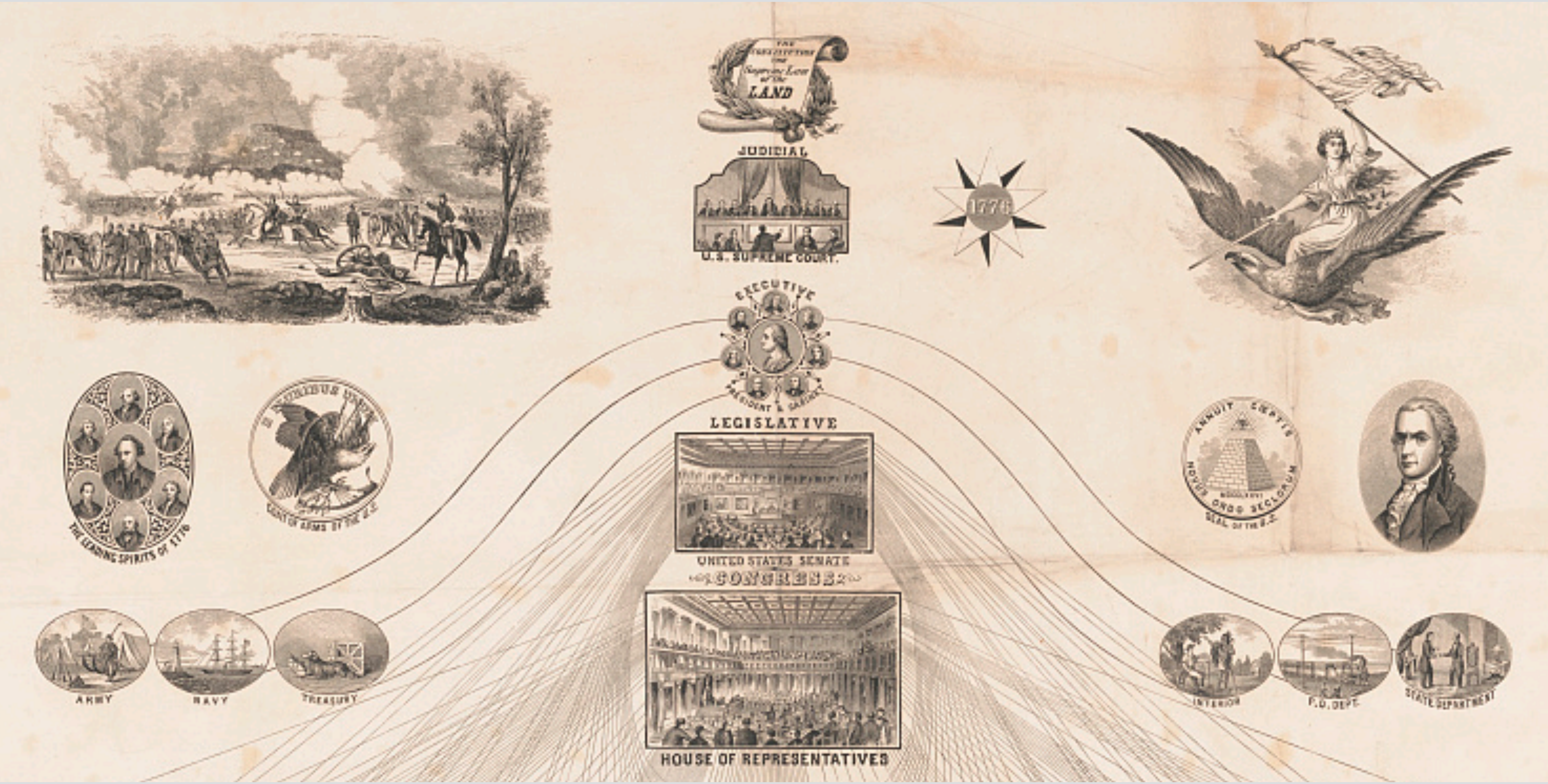
Daniel Jackson, MIT CSAIL · Nasa Formal Methods 2022 · May 26, 2022

we use structure
to understand
artifacts

villa designs (andrea palladio, c. 1570)



american federal government (1864)



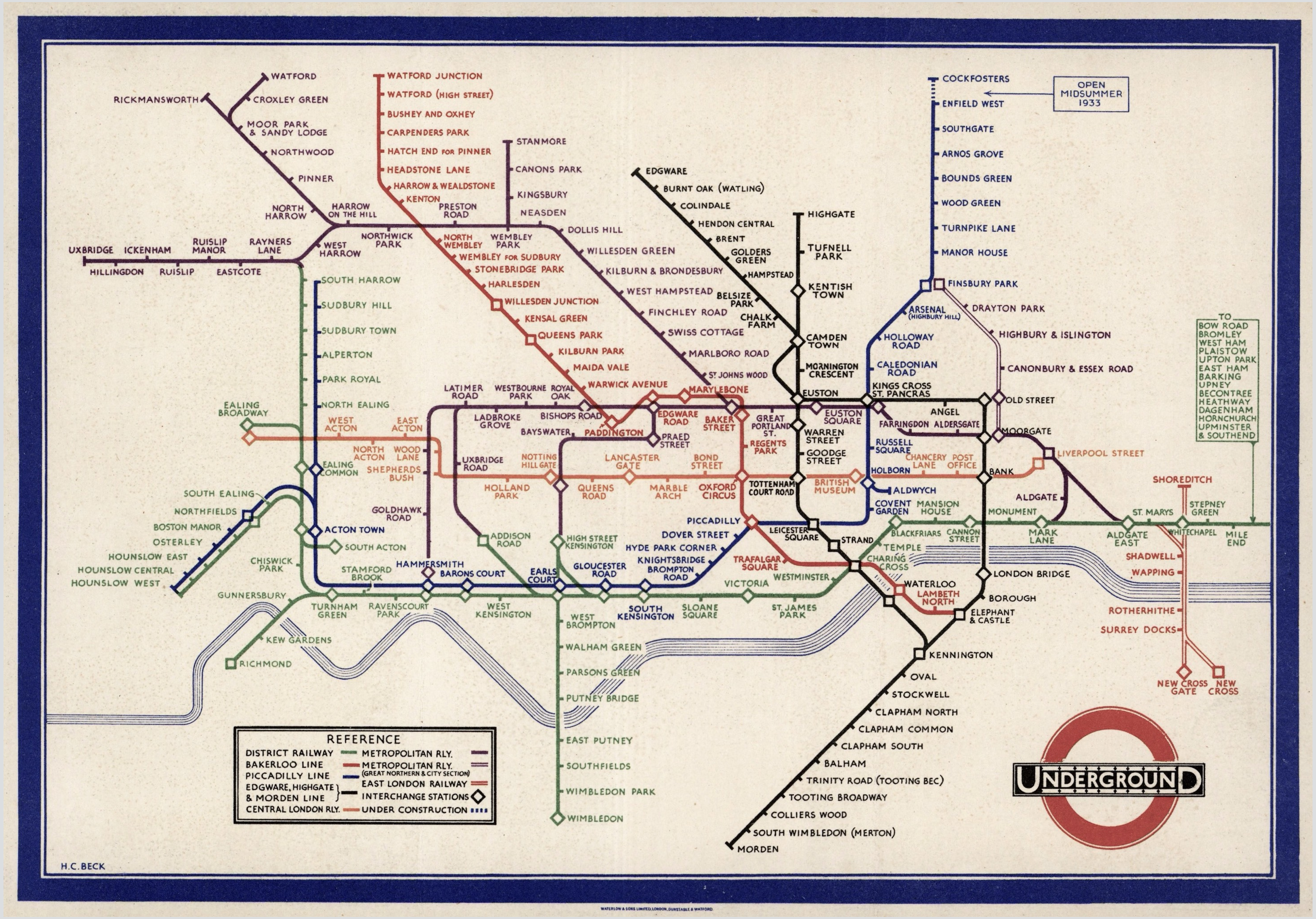
periodic table (mendeleev, c. 1870)

Естественная система элементовъ Д. Менделѣева.

		Группа I.	Группа II.	Группа III.	Группа IV.	Группа V.	Группа VI.	Группа VII.	Группа VIII. (переходъ къ I)			
	Высшій окиселъ образующій соли:	R^2O	R^2O^2 или RO	R^2O^3	R^2O^4 или RO^2	R^2O^5	R^2O^6 или RO^3	R^2O^7	R^2O^8 или RO^4			
		H=1 $H^2O_\wedge, HH_\wedge, HCl_\wedge$ $H^3N_\wedge, H^4C_\wedge, ROH.$		——	RH^4	RH^3	RH^2	RH				H=1 HX
	Типичес. Рядъ	Li=7 $LiCl, LiOH, Li^2O.$ $LiX, Li^2CO^3_*$	Be=9,4 $BeCl^2, BeO_*$ $Be^3Al^2Si^6O^{18}_*$	B=11 $BCl^3_\wedge, B^2O^3, BN_*$ $B^4Na^2O^7, BF^3_\wedge$	C=12 $CH^4_\wedge, C^nH^{2n+2}_\wedge$ $CO_\wedge, CO^2_\wedge, CO^3M^2_*$	N=14 $NH^3_\wedge, NH^4Cl_\wedge, N^2O_\wedge$ $NO_\wedge, NO^3M, CNM.$	O=16 $OH^2_\wedge, O^2C_\wedge, O^2_\wedge, O^3_\wedge$ $OM^2, O^nR, HOR.$	F=19 $FH_\wedge, BF^3_\wedge, SiF^4_\wedge$ $CaF^2_*, KF, KHF^2.$				
Періодъ 1-й.	Рядъ 1.	Na=23 $NaCl, NaHO, Na^2O$ Na^2SO^4, Na^2CO^3	Mg=24 $MgCl^2, MgO_*, MgCO^3_*$ $MgSO^4, MgNH^4PO^4_*$	Al=27,3 $Al^2Cl^6_\wedge, Al^2O^3_*$ $KAlS^2O^8 12H^2O.$	Si=28 $SiH^4_\wedge, SiCl^4_\wedge, SiH^2F^6$ $KAlSi^3O^8_*, SiO^2_*$	P=31 $PH^3_\wedge, PCl^3_\wedge, PCl^5_\wedge$ $P^2O^3_\wedge, P^2O^5, Ca^3P^2O^8_*$	S=32 $SH^2_\wedge, SM^2, S^nM^2$ $SO^2_\wedge, SO^2X^2, Ba^2SO^4_*$	Cl=35,5 $ClH_\wedge, ClM, ClCl_\wedge$ $ClOH, ClO^4H_\wedge, AgCl_*$				
	Рядъ 2.	K=39 KCl, KOH, K^2O $KNO^3, K^2PtCl^6_*, K^2SiF^6_*$	Ca=40 $CaSO^4_*, CaOnSiO^2_*$ $CaCl^2, CaO_*, CaCO^3_*$	—— ?44=Eb?	Ti=48(50?) $TiCl^4, TiO^2_*, Ti^2O^3_*$ $FeTiO^3_*, TiOSO^4$	V=51 $VOCl^3, V^2O^5, VO^2_*$ $Pb^3V^2O^8_*, VO_*$	Cr=52 $CrCl^3_*, CrCl^2, Cr^2O^3_*$ $CrO^3, K^2CrO^4, CrO^2Cl^2_\wedge$	Mn=55 $MnK^2O^4, MnKO^4$ $MnCl^2, MnO_*, MnO^2_*$	Fe=56 FeK^2O^4, FeS^2_* $FeO_*, Fe^2O^3_*$ FeK^4Cy^6	Co=59 CoX^2, CoX^3 $CoX^3 5NH^3$ CoK^3Cy^6	Ni=59 NiX^2, NiO_* $NiSO^4 6H^2O$ NiK^2Cy^4	Cu=63 CuX, CuX^2, CuH_* Cu^2O_*, CuO_* $CuKCy^2$
Періодъ 2-й.	Рядъ 3.	Cu=63 CuX, CuX^2	Zn=65 $ZnCl^2, ZnO_*, ZnCO^3_*$ $ZnSO^4, ZnEt^2_\wedge$	—— ?68=El?	—— ?72=Es? ?Il, EsO^2?	As=75 $AsH^3, AsCl^3_\wedge, As^4O^6_\wedge$ $As^2O^5, As^2S^3_*$	Se=78 $SeH^2_\wedge, SeO^2, SeO^3$ SeM^2, SeM^2O^4	Br=80 $BrH_\wedge, BrM,$ $BrO^3M, BrAg_*$				
	Рядъ 4.	Rb=85 $RbCl, RbOH.$ $Rb^2PtCl^6_*$	Sr=87 $SrCl^2, SrO, SrH^2O^2$ $SrSO^4_*, SrCO^3_*$	—— ?88=Yt?(92) ?$Yt^2O^3_*, YtX^3$?	Zr=90 $ZrCl^4_\wedge, ZrO^2_*, ZrX^4.$	Nb=94 $NbCl^5_\wedge, Nb^2O^5$ $Nb^2O^5, NbOK^2F^5$	Mo=96 $MoCl^4_\wedge, MoS^2_*, MoO^3_*$ $M^2MoO^4 nMoO^3$	—— 100	Ru=104 $RuO^4_\wedge, RuCl^4$ $RuO^2, RuCl^3$ RuK^4Cy^6	Rh=104 $RhCl^3, RhCl^2$ $Rh^2O^3_*, RhX^3$ RhK^3Cy^6	Pd=106 PdH_*, PdO_* PdI^2_*, PCl^2 PdK^2Cy^4	Ag=108 $AgNO^3, AgX$ $AgCl_*, Ag^2O_*$ $AgKCy^2$
Періодъ 3-й.	Рядъ 5.	Ag=108 $AgX, AgCl_*$	Cd=112 $CdCl^2, CdO_*, CdS_*$ $CdSO^4$	In=113 $InCl^3, In^2O^3_*$	Sn=118 $SnCl^2_\wedge, SnCl^4_\wedge, SnO_*$ $SnX^4, SnNa^2O^3$	Sb=122 $SbH^3_\wedge, SbCl^3_\wedge, Sb^2O^3_*$ $Sb^2O^5_*, Sb^2S^3_*, SbOX$	Te=125(?128?) $TeH^2_\wedge, TeCl^4, TeO^2$ TeO^4M^2, TeM^2_*	I=127 $IH_\wedge, IAg_*, IHO^3$ IHO^4, HgI^2_*, KI				
	Рядъ 6.	Cs=133 $CsCl, CsOH.$ $Cs^2PtCl^6_*$	Ba=137 $BaCl^2, BaH^2O^2, BaO$ $BaSO^4_*, BaSiF^6_*$	—— ?138=La?=Di?(144) ?$La^2O^3_*, LaX^3$?	Ce=140(138?) $CeCl^3, Ce^2O^3_*, CeO^2_*$ CeX^3, CeX^4, CeK^2X^6	—— 142	—— 146	—— 148	—— 150	—— 151	—— 152	—— 153
Періодъ 4-й.	Рядъ 7.	—— 153	—— 158	—— 160	—— 162	—— 164	—— 166	—— 168				
	Рядъ 8.	—— 175	—— 177	—— ?178=Er?(169) ?$Er^2O^3_*, ErX^3$?	—— ?180=Di?=La(187) ?DiO^2_*, DiX^4?	Ta=182 $TaCl^5_\wedge, Ta^2O^5_*$ $TaK^2F^7_*$	W=184 $WCl^6_\wedge, WCl^4, WO^3_*$ $K^2WO^4 nWO^3_*$	—— 190	199? Os=193 $OsO^4_\wedge, OsH^2O^4$ $OsCl^4, OsCl^3$ OsK^4Cy^6	198? Ir=195 $K^2IrCl^6, IrCl^4$ $IrCl^3_*, Ir^2O^3_*$ IrK^3Cy^6	Pt=197 $PtCl^4, PtO^2_*$ $PtCl^2, PtK^2X^4$ PtK^2Cy^4	Au=197 $AuCl^3, AuCl$ $Au^2O^3_*, Au^2O_*$ $AuKCy^2$
Періодъ 5-й.	Рядъ 9.	Au=197 AuX, AuX^3	Hg=200 $HgCl_\wedge, HgCl^2_\wedge, Hg^2O_*$ $HgO_*, HgX^2 nHgO$	Tl=204 $TlCl_*, Tl^2O, Tl^2O^3_*$ $Tl^2SO^4, TlCl^3$	Pb=207 $PbCl^2_*, PbO_*, PbO^2_*$ $PbEt^4_\wedge, PbSO^4_*, PbK^2O^3$	Bi=208 $BiCl^3, Bi^2O^3_*, Bi^2O^7H^4_*$ $BiX^3, BiOX, BiNO^3(HO)^2_*$	—— 210	—— 212				
	Рядъ 10.	—— 220	—— 225	—— 227	Th=231 $ThCl^4, ThO^2_*$ $ThX^4, Th(SO^4)^2$	—— 235	U=240 $UCl^4_\wedge, UO^3_*, UO^2X^2$ $UO^3_*, M^2U^2O^7_*$	—— 245	—— 246	—— 248	—— 249	—— 250

* Тѣло твердое, малорастворимое въ водѣ.
∧ Тѣло газообразное или летучее.
M=K, Ag..... M^2=Ca, Pb .....
X=Cl, ONO^2, OH, OM..... X^2=SO^4, CO^3, O, S.....

london underground (harry beck, 1933)



experiential

structure helps you understand how it behaves not how it’s built

experiential

structure helps you understand how it behaves not how it's built

modular

components of the structure can be understood independently

experiential

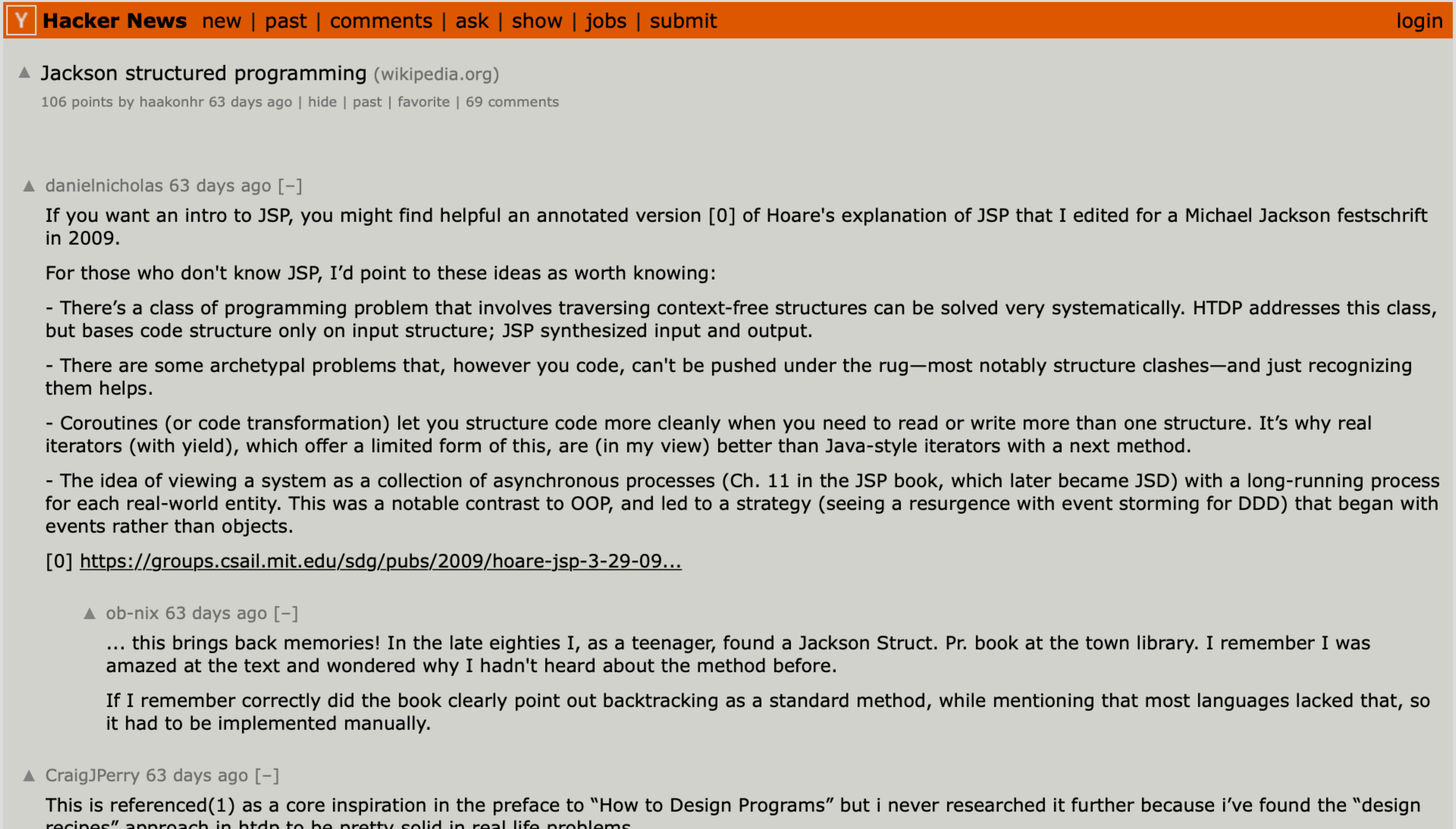
structure helps you understand how it behaves not how it’s built

modular

components of the structure can be understood independently

abstract

internal workings and structure are not shown what are the elements of software?



Y **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 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...

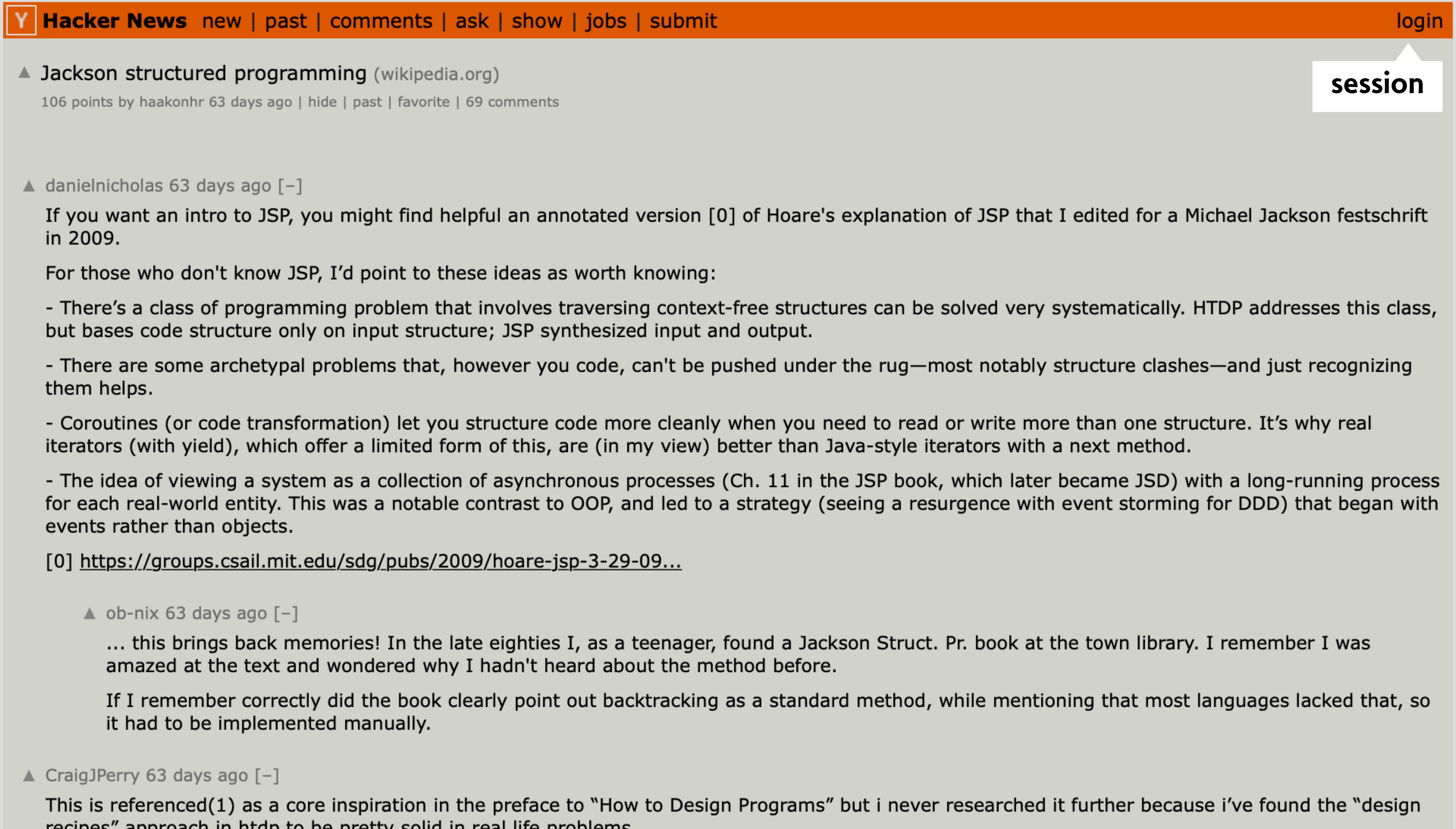
▲ ob-nix 63 days ago [–]

... 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.



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

session

▲ 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...

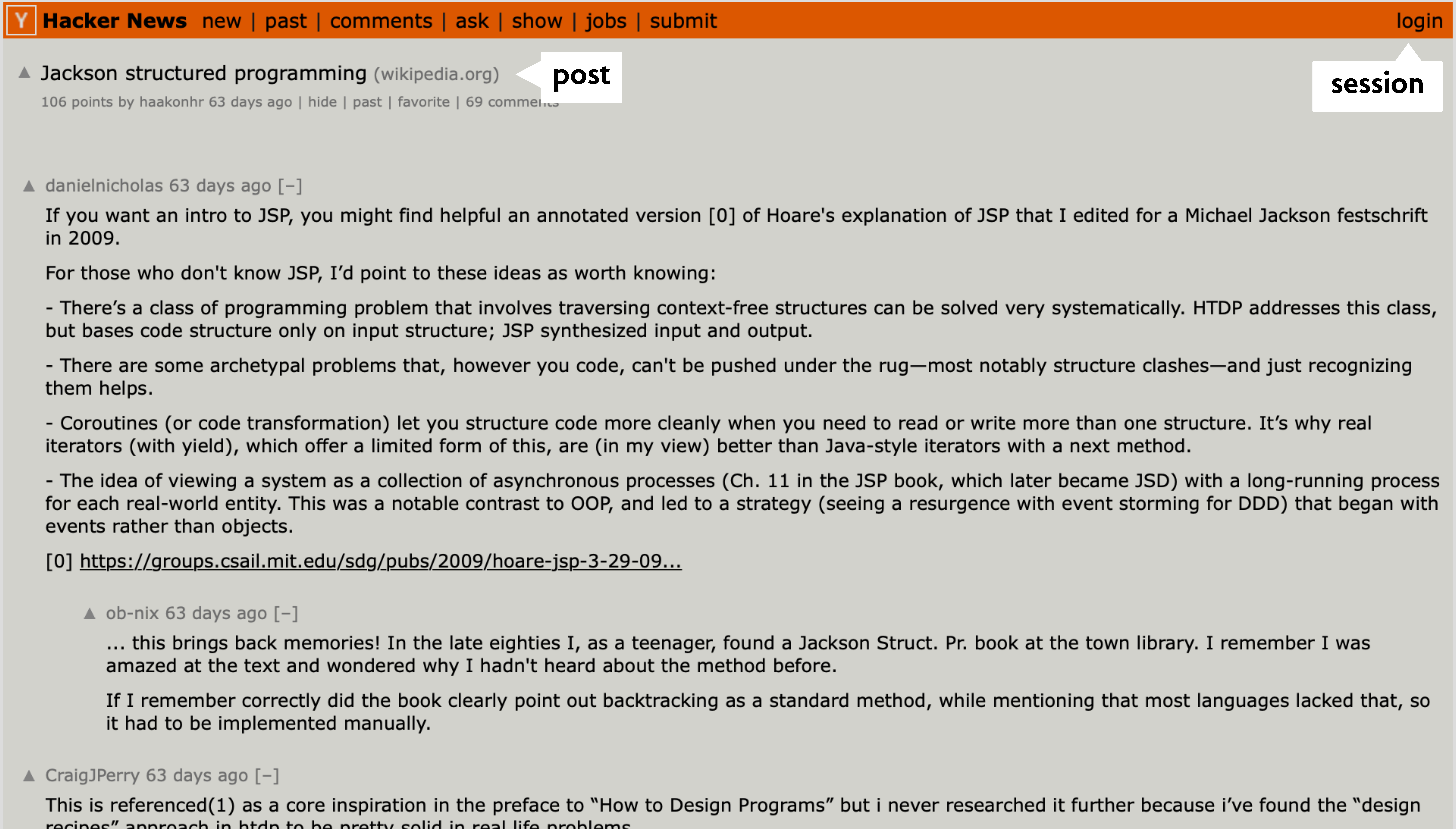
▲ ob-nix 63 days ago [–]

... 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



post

session

▲ 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...

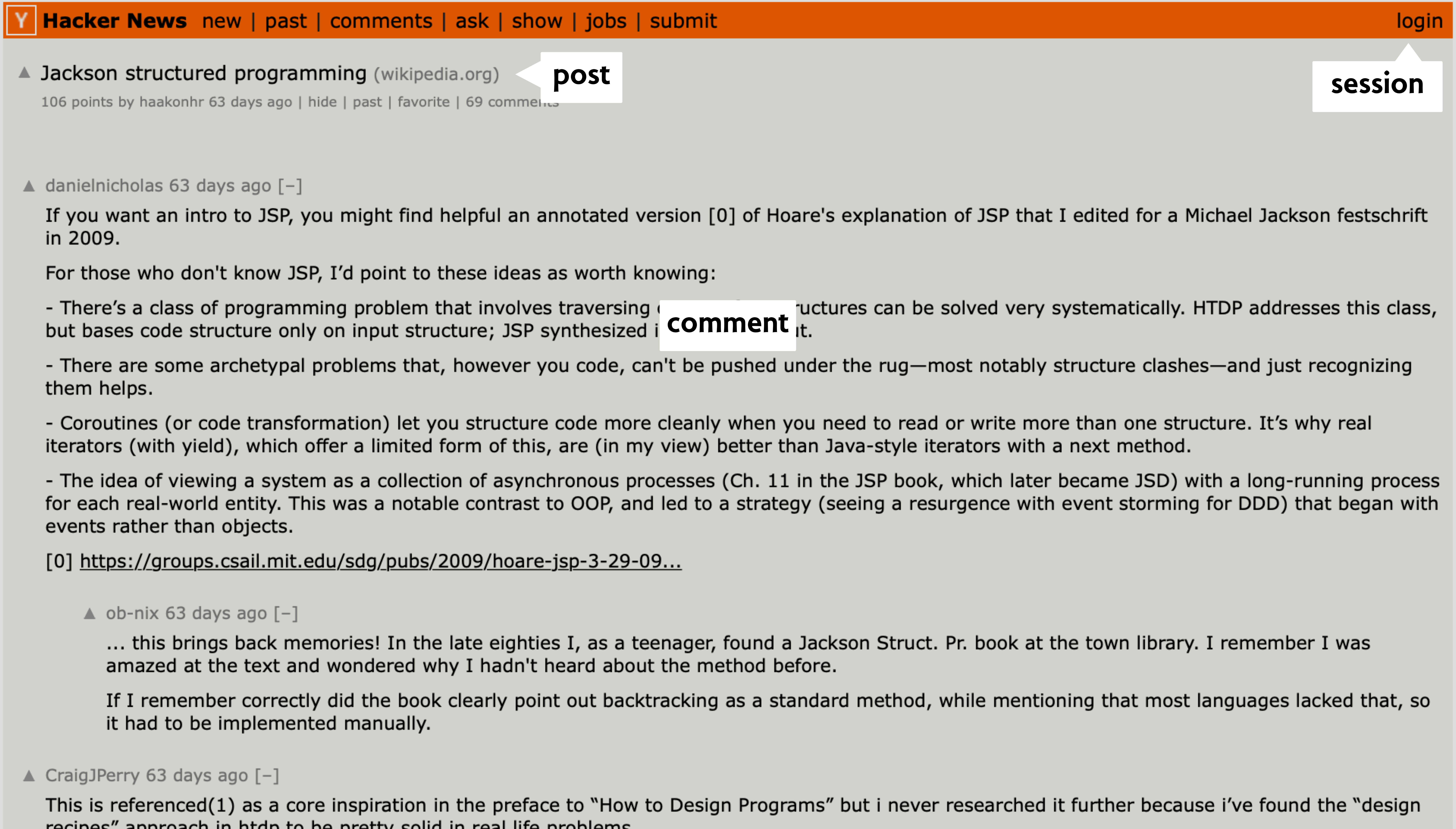
▲ ob-nix 63 days ago [–]

... 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



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

post

session

comment

▲ Jackson structured programming (wikipedia.org)

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

▲ 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 uctures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized i 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...

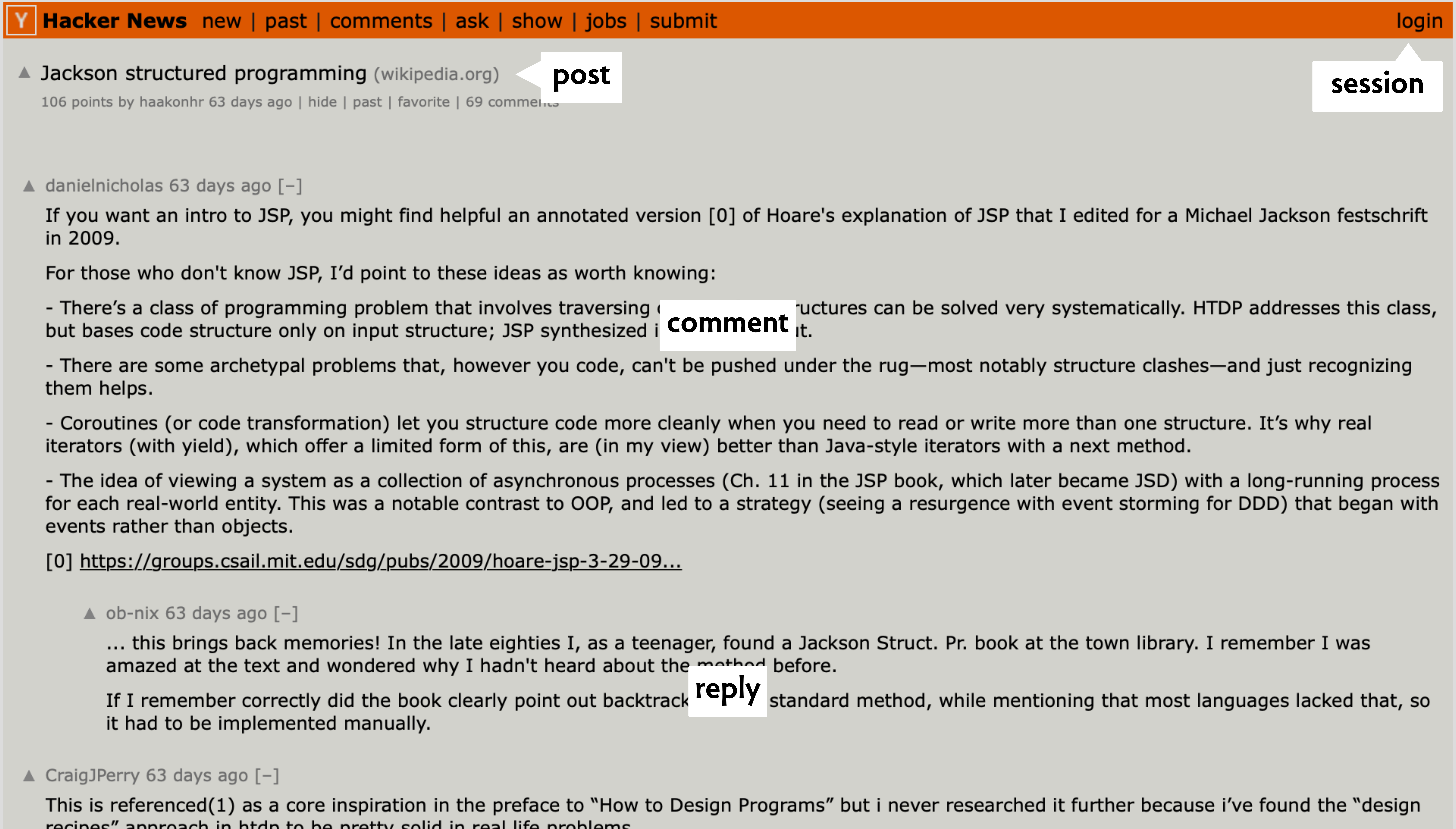
▲ ob-nix 63 days ago [–]

... 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 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



Y **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 [–]

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 ructures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized i 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...

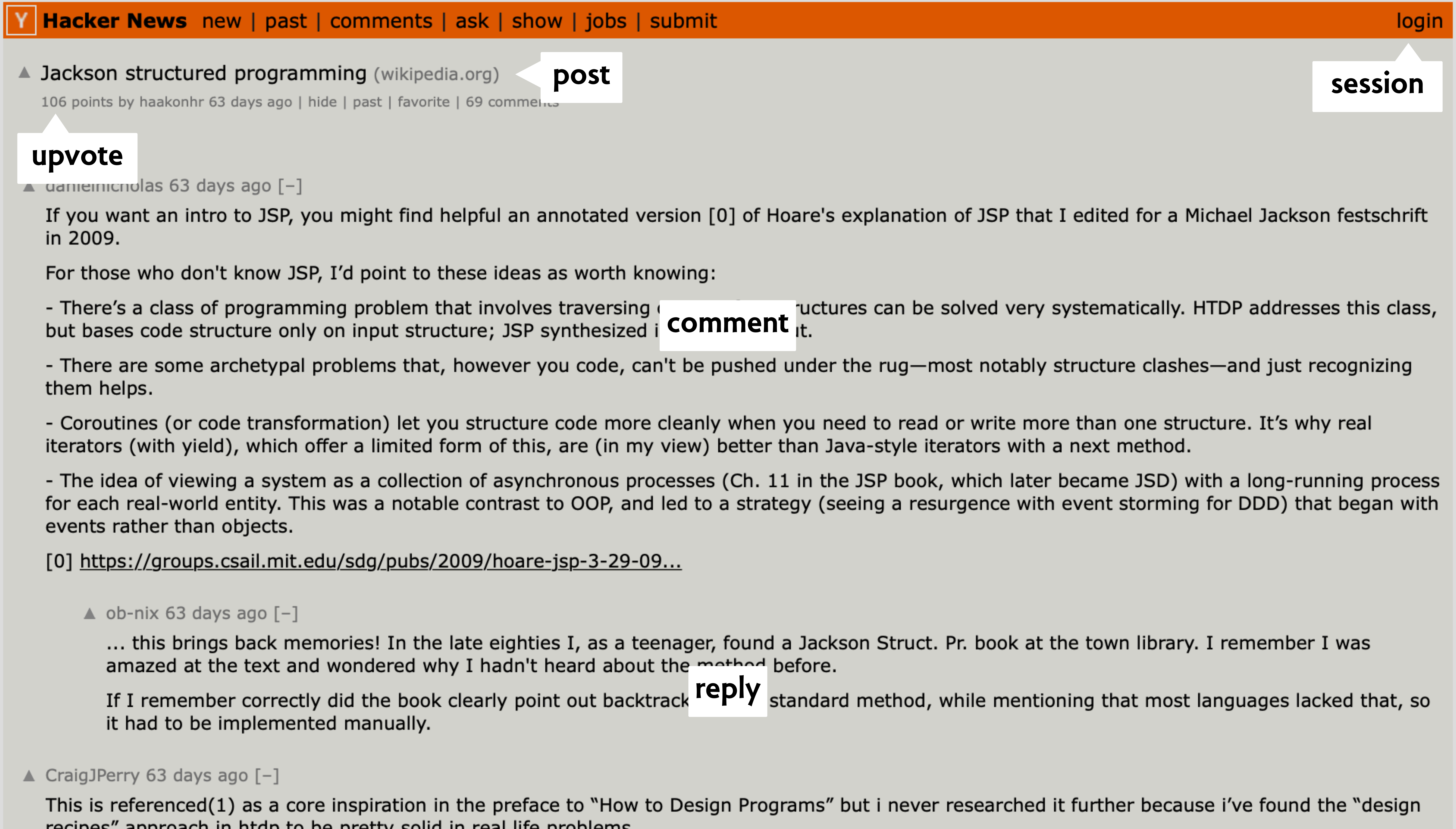
▲ ob-nix 63 days ago [–]

... 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 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.

▲ 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



Y 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 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 c ructures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized i 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...

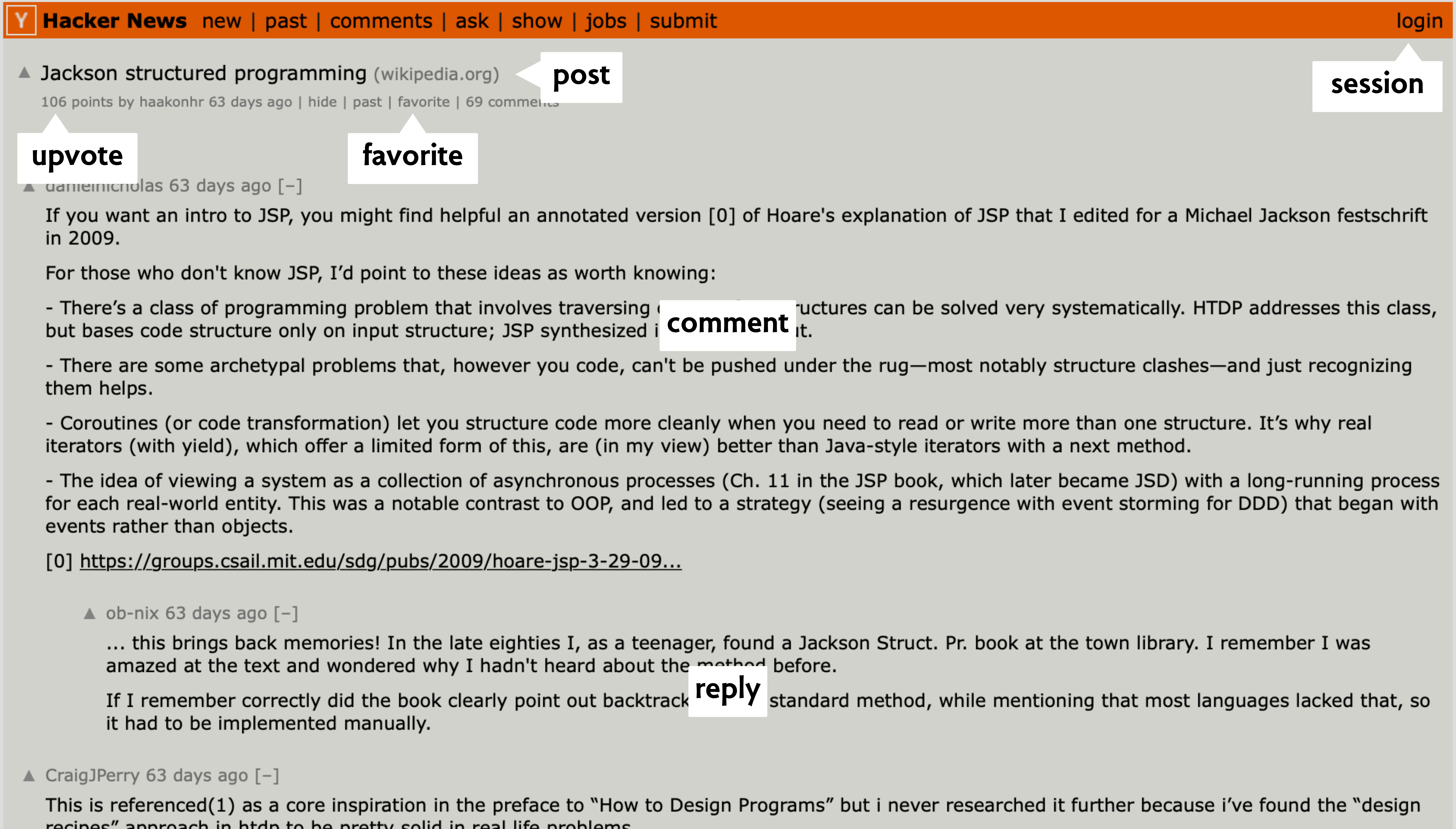
▲ ob-nix 63 days ago [–]

... 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.

▲ 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



Y 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 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 (ructures can be solved very systematically. HTDP addresses this class, but bases code structure only on input structure; JSP synthesized i 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.

▲ 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

post

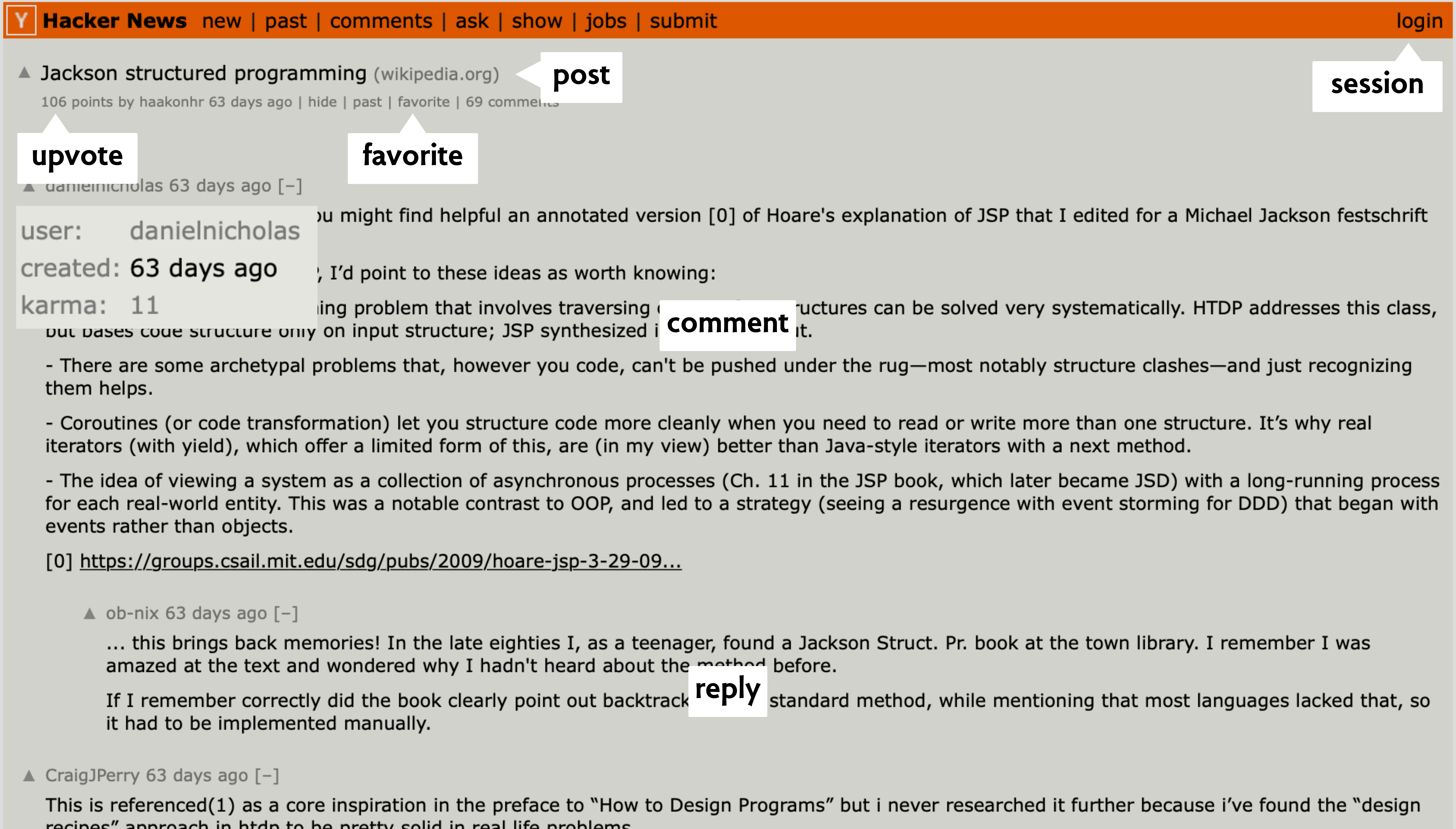
session

upvote

favorite

comment

reply



Y Hacker News new | past | comments | ask | show | jobs | submit
login
post
session
upvote
favorite
comment
reply

▲ 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, on input structure; JSP synthesized i 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...

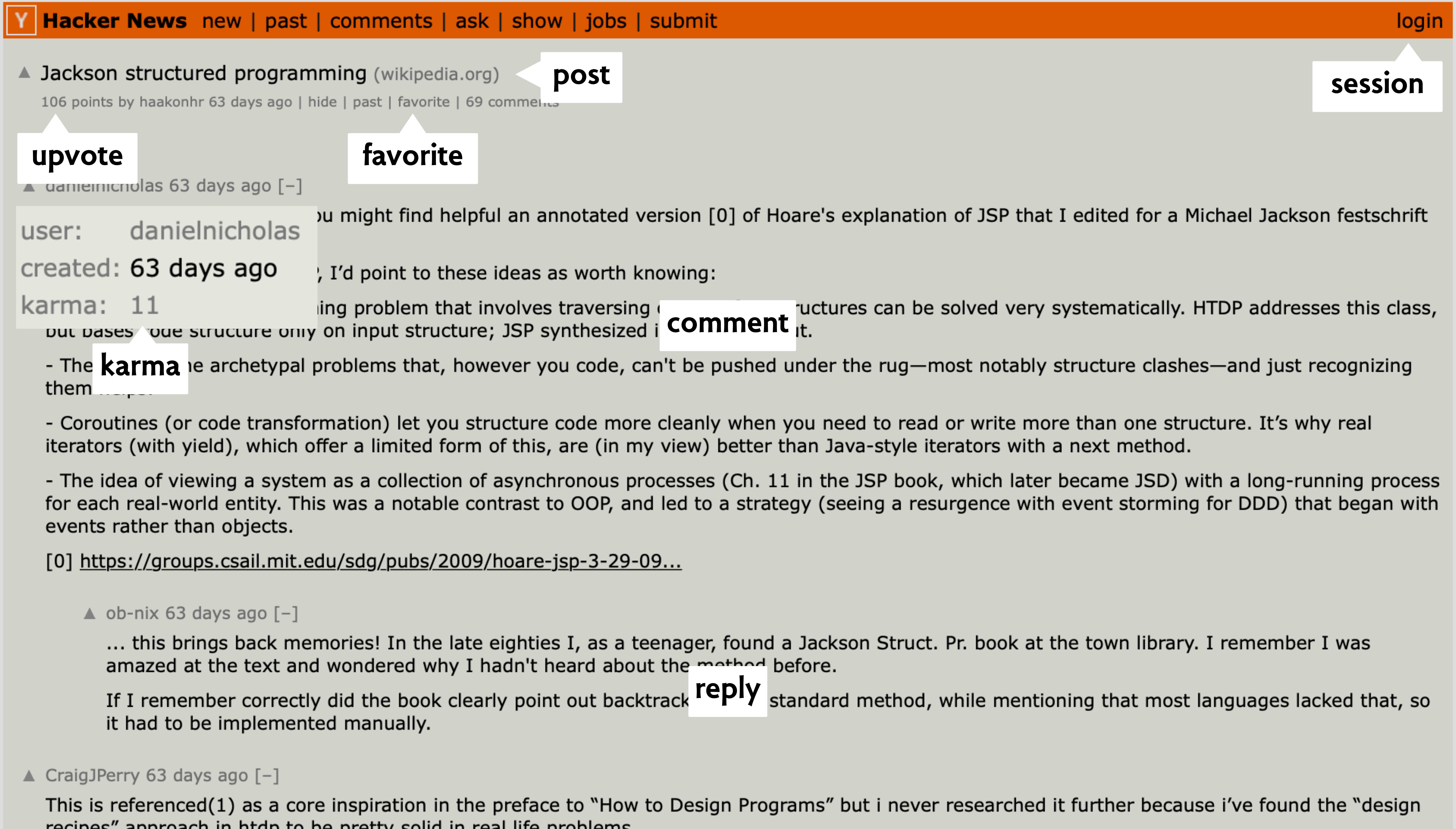
▲ ob-nix 63 days ago [–]

... 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 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.

▲ 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



Y **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 comments

▲ danielnicholas 63 days ago [–]

user: danielnicholas
created: 63 days ago
karma: 11

ou 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, but bases 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 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.

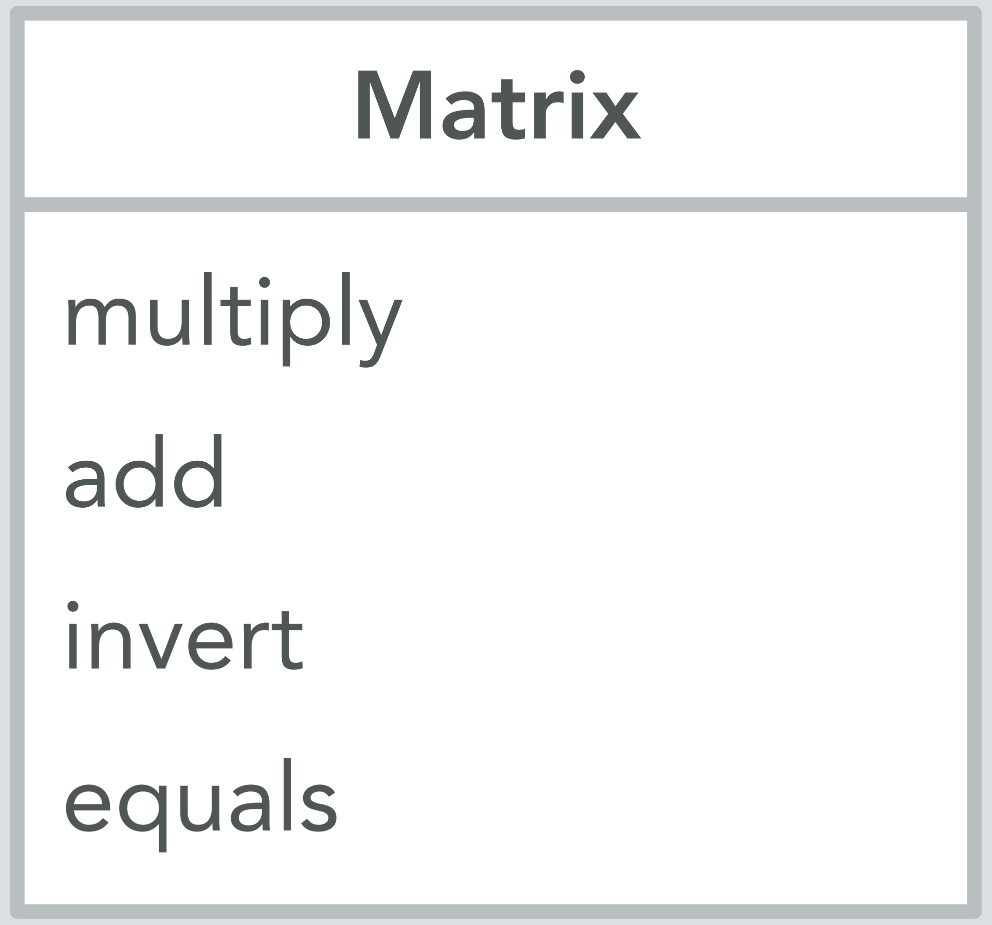
▲ CraigJPerry 63 days ago [–]

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

but what’s a concept? three things it isn’t

but what's a concept? three things it isn't

abstract type, class/object



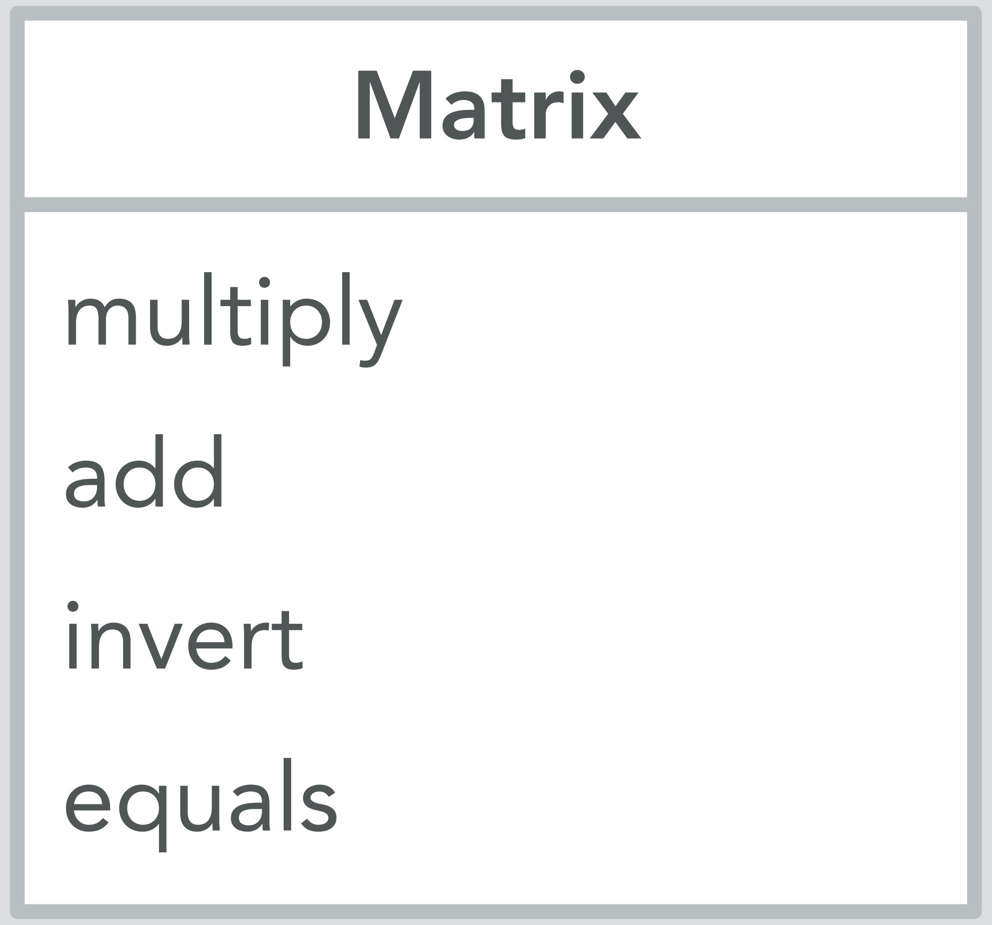
Matrix

multiply
add
invert
equals

not limited to built-in types
encapsulate representation
defined by operations alone

but what's a concept? three things it isn't

abstract type, class/object



Matrix

multiply
add
invert
equals

not limited to built-in types
encapsulate representation
defined by operations alone

what operations can
you do on an upvote?

but what’s a concept? three things it isn’t

abstract type, class/object

Matrix

multiply

add

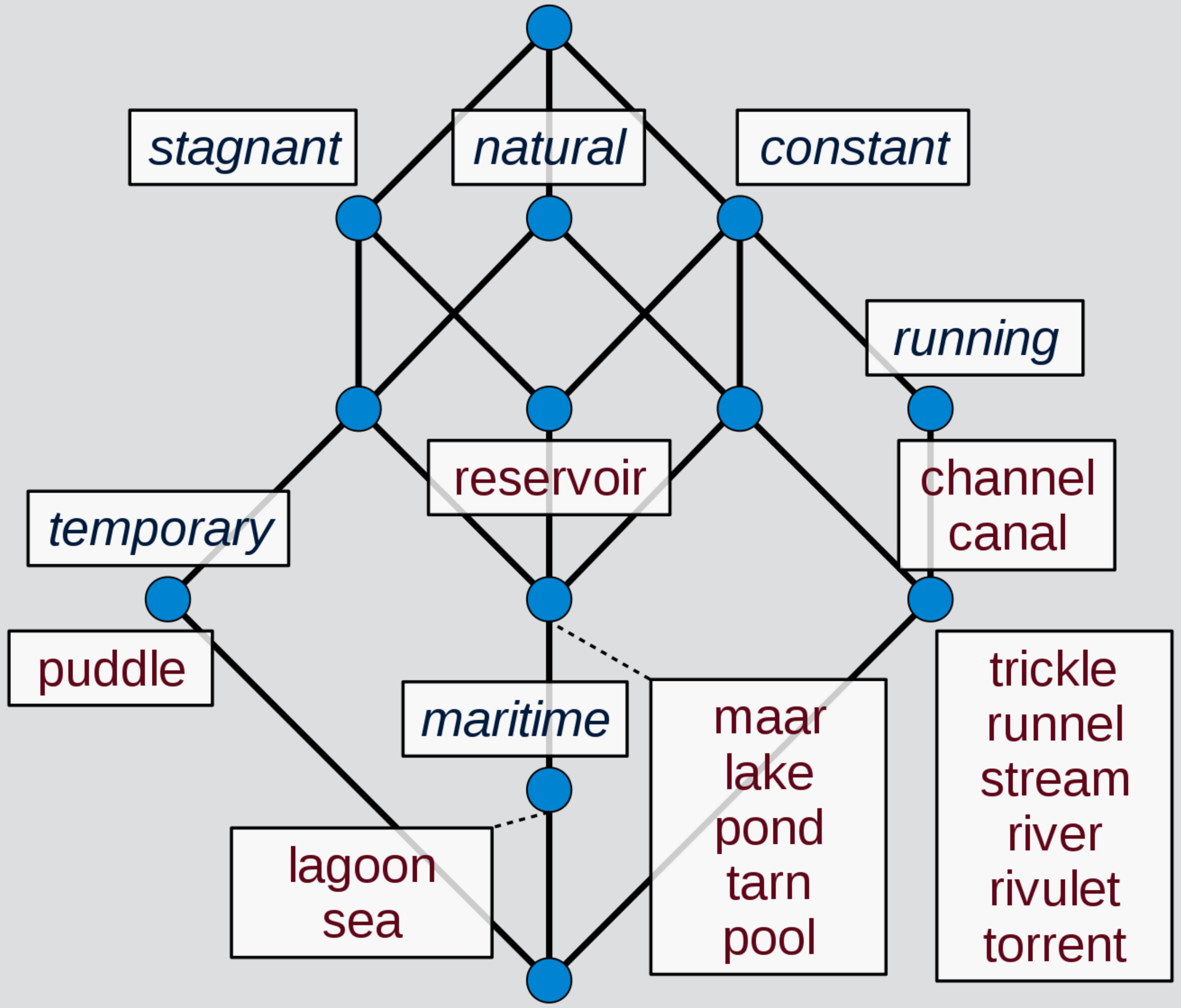
invert

equals

not limited to built-in types
encapsulate representation
defined by operations alone

what operations can
you do on an upvote?

concept lattice



but what's a concept? three things it isn't

abstract type, class/object

Matrix

multiply

add

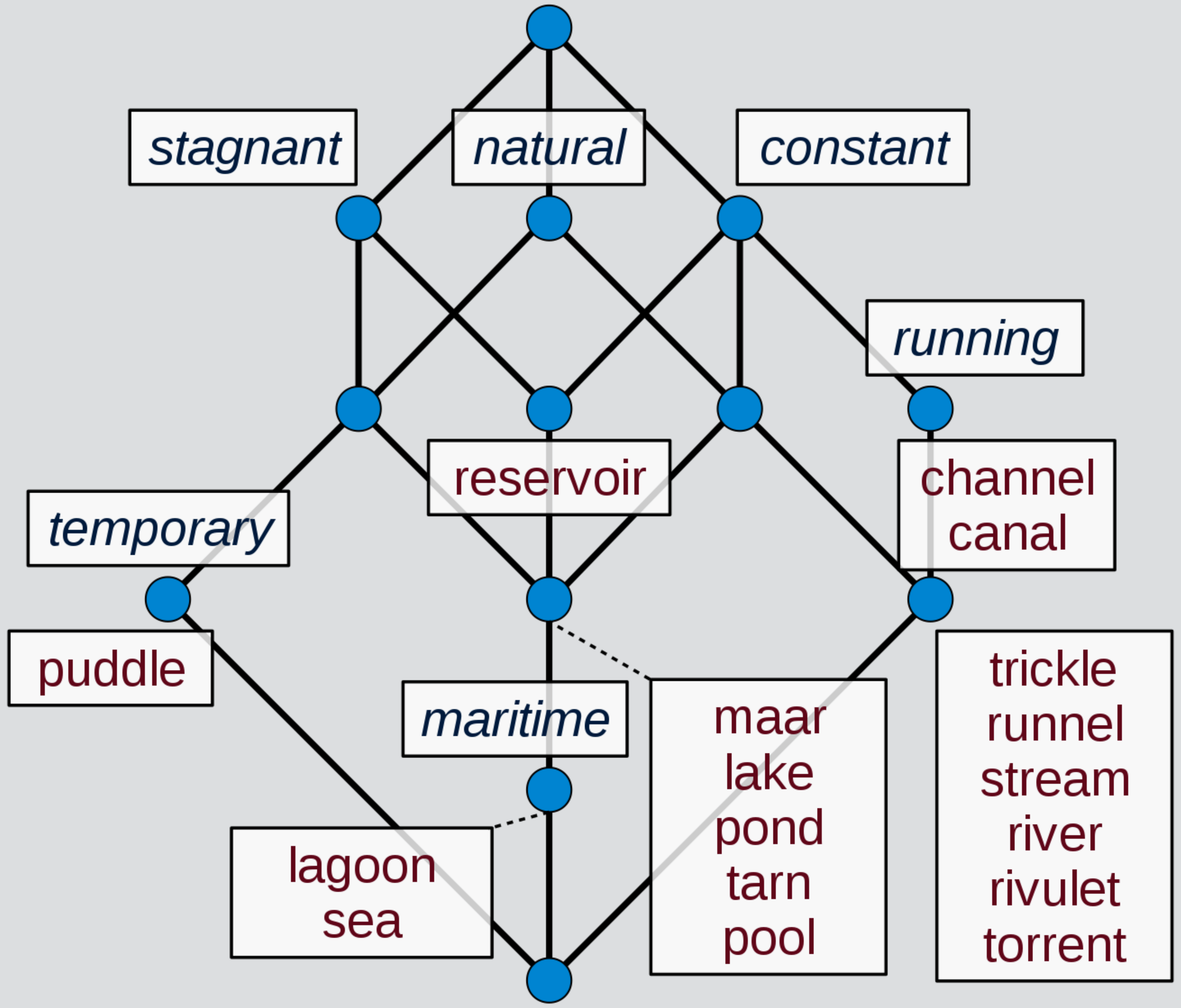
invert

equals

not limited to built-in types
encapsulate representation
defined by operations alone

what operations can
you do on an upvote?

concept lattice



upvotes and downvotes
are votes and then what?

but what's a concept? three things it isn't

abstract type, class/object

Matrix

multiply

add

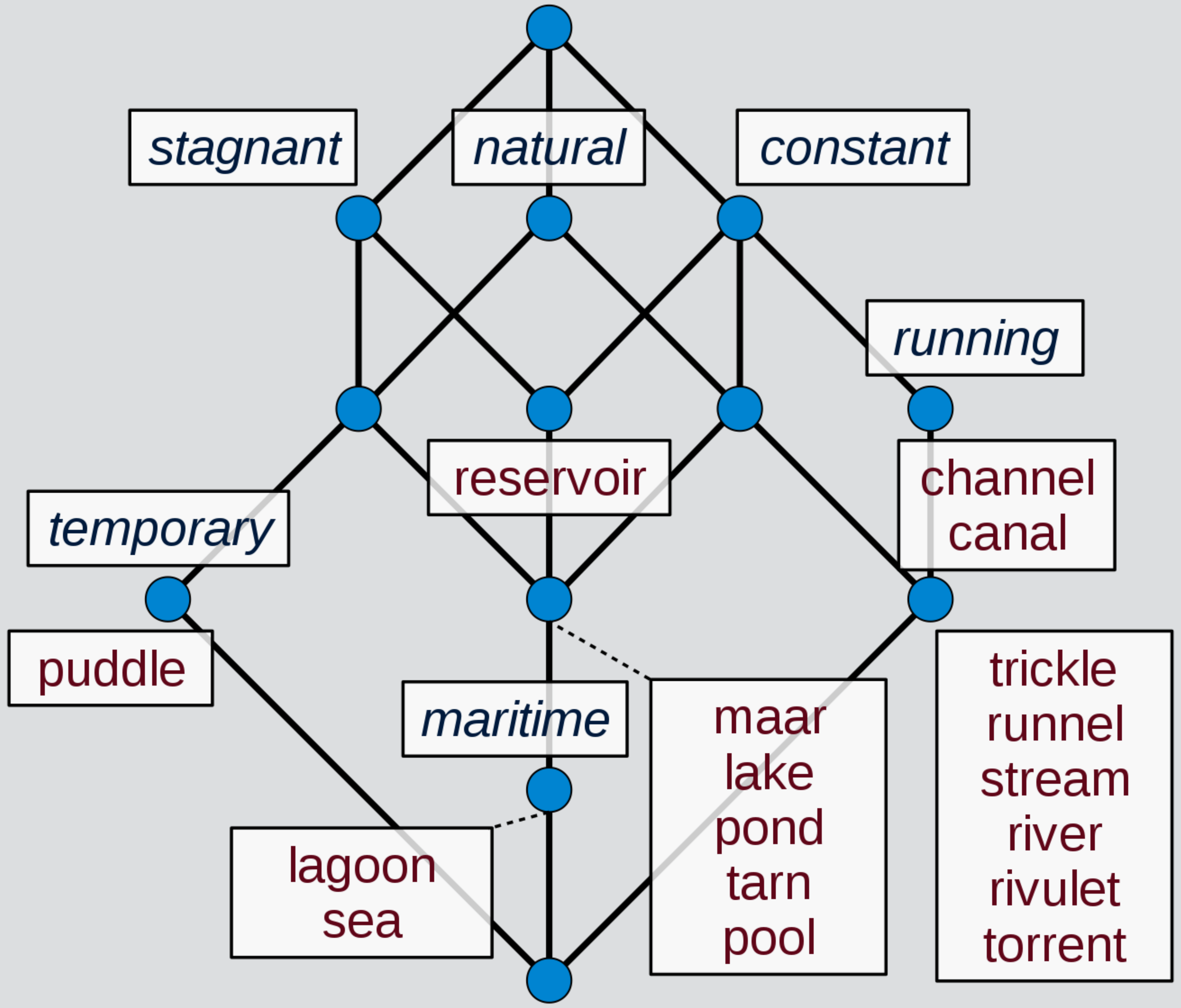
invert

equals

not limited to built-in types
encapsulate representation
defined by operations alone

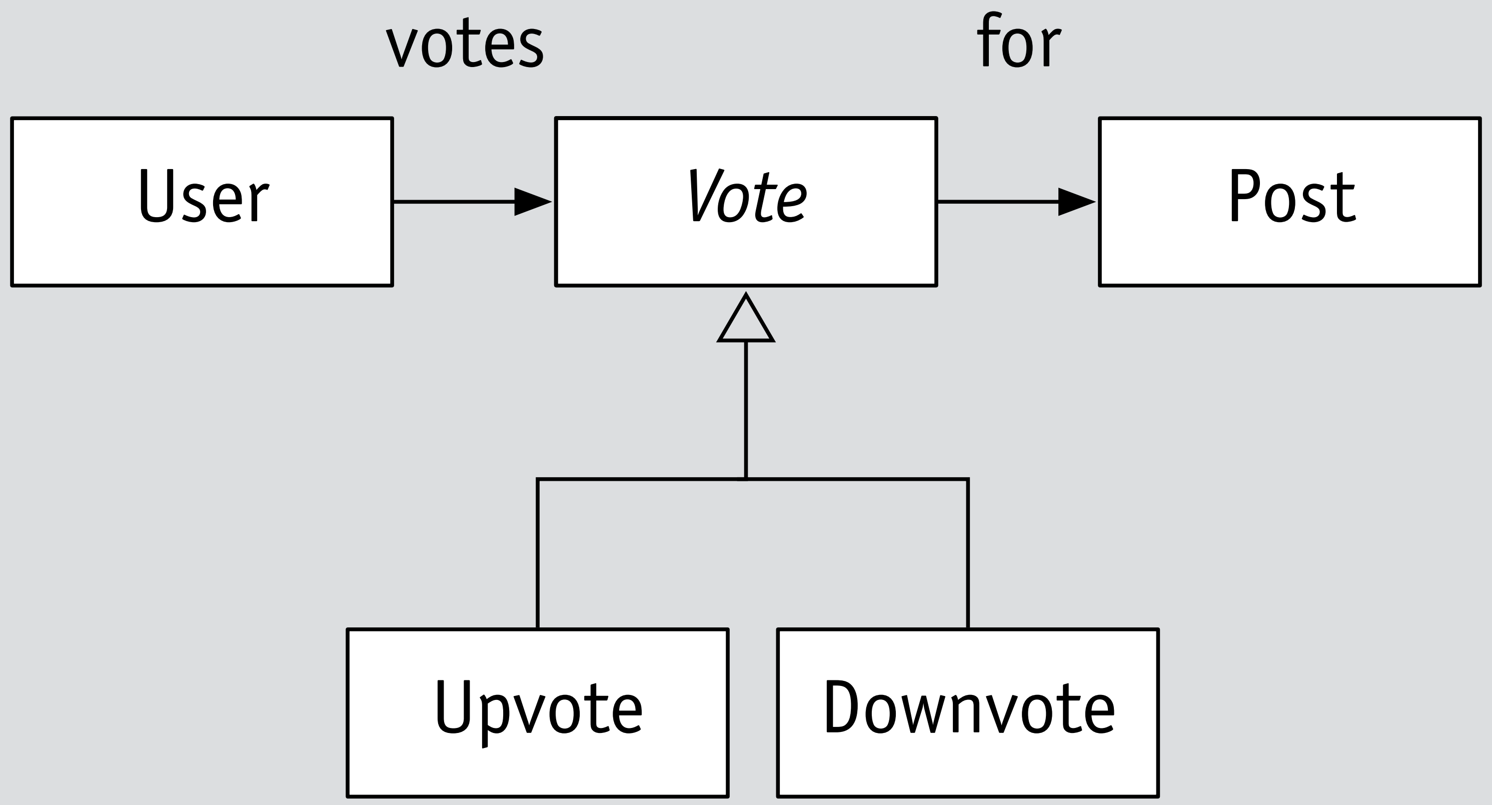
what operations can you do on an upvote?

concept lattice



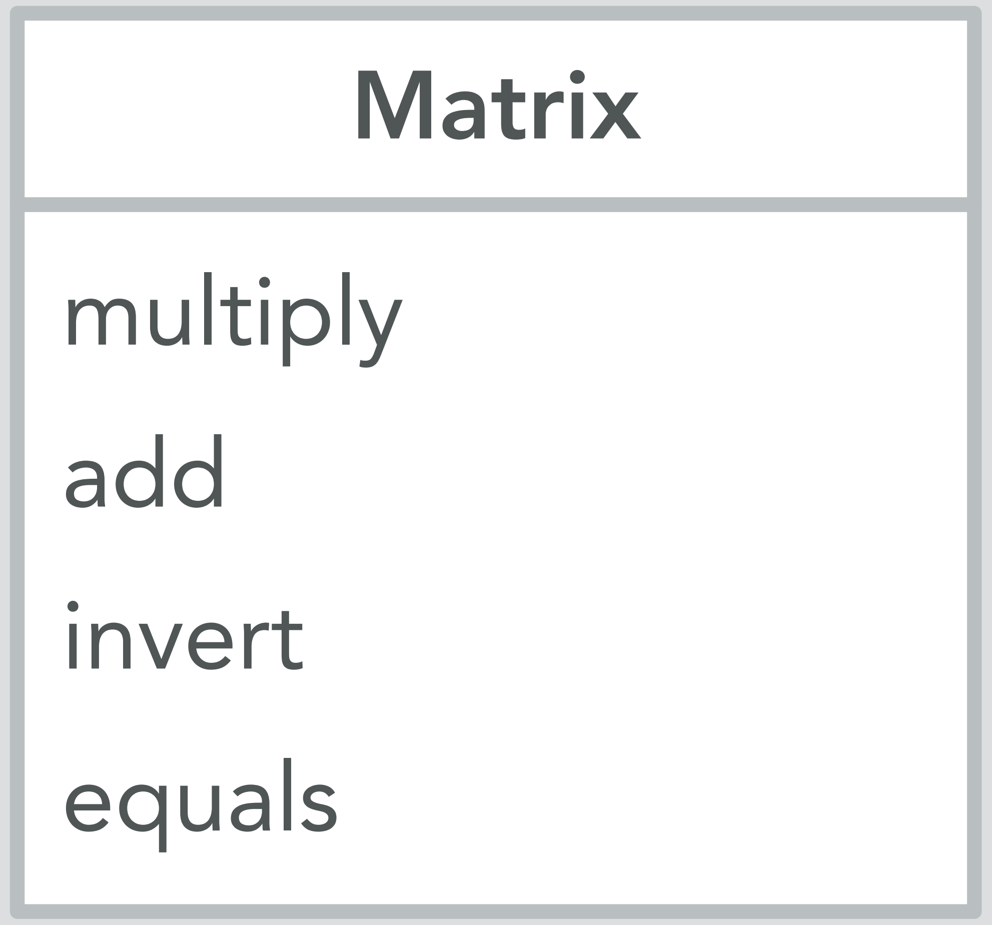
upvotes and downvotes are votes and then what?

entity in data model



but what's a concept? three things it isn't

abstract type, class/object



Matrix

multiply

add

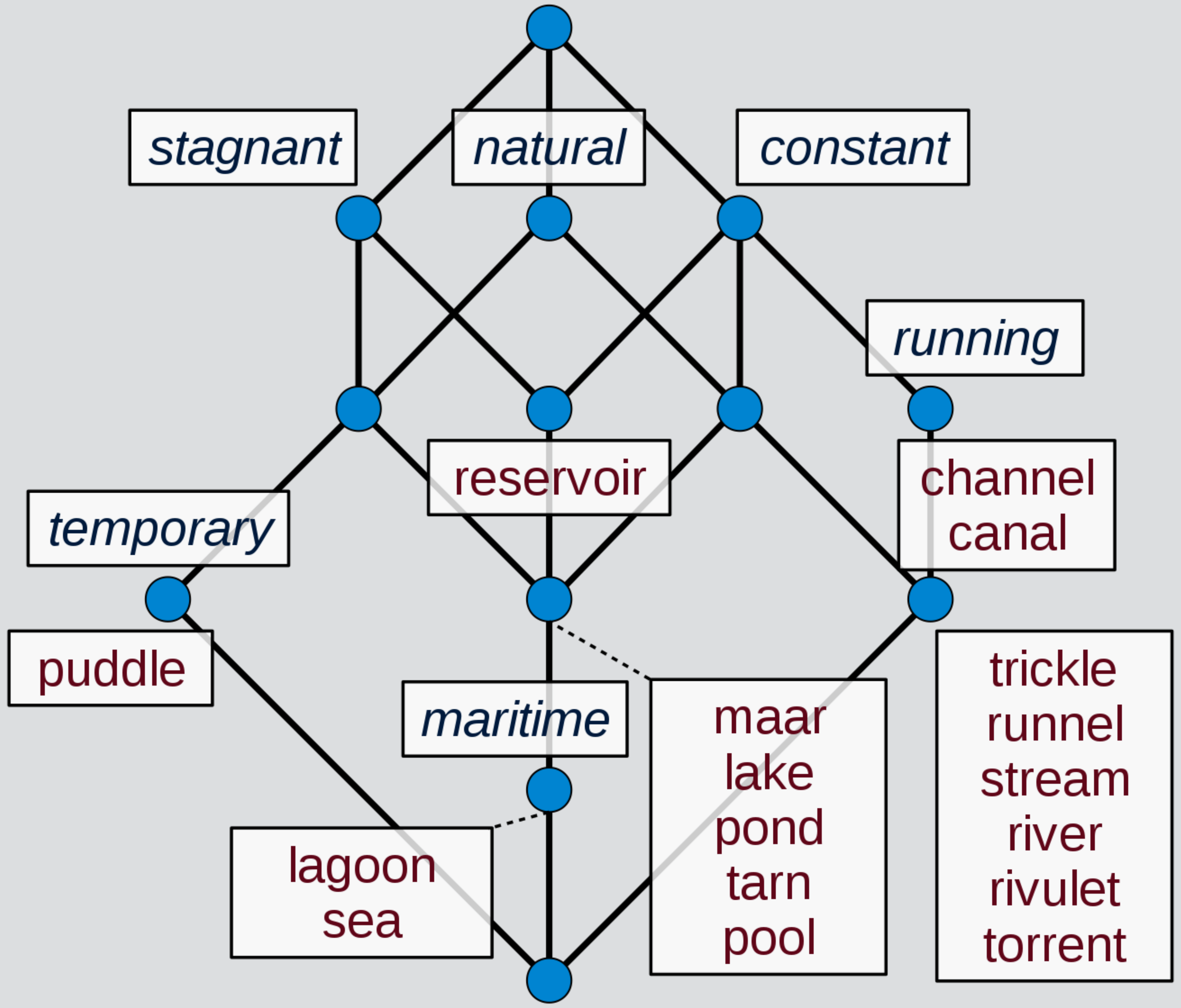
invert

equals

not limited to built-in types
encapsulate representation
defined by operations alone

what operations can
you do on an upvote?

concept lattice



stagnant

natural

constant

running

temporary

reservoir

channel
canal

puddle

maritime

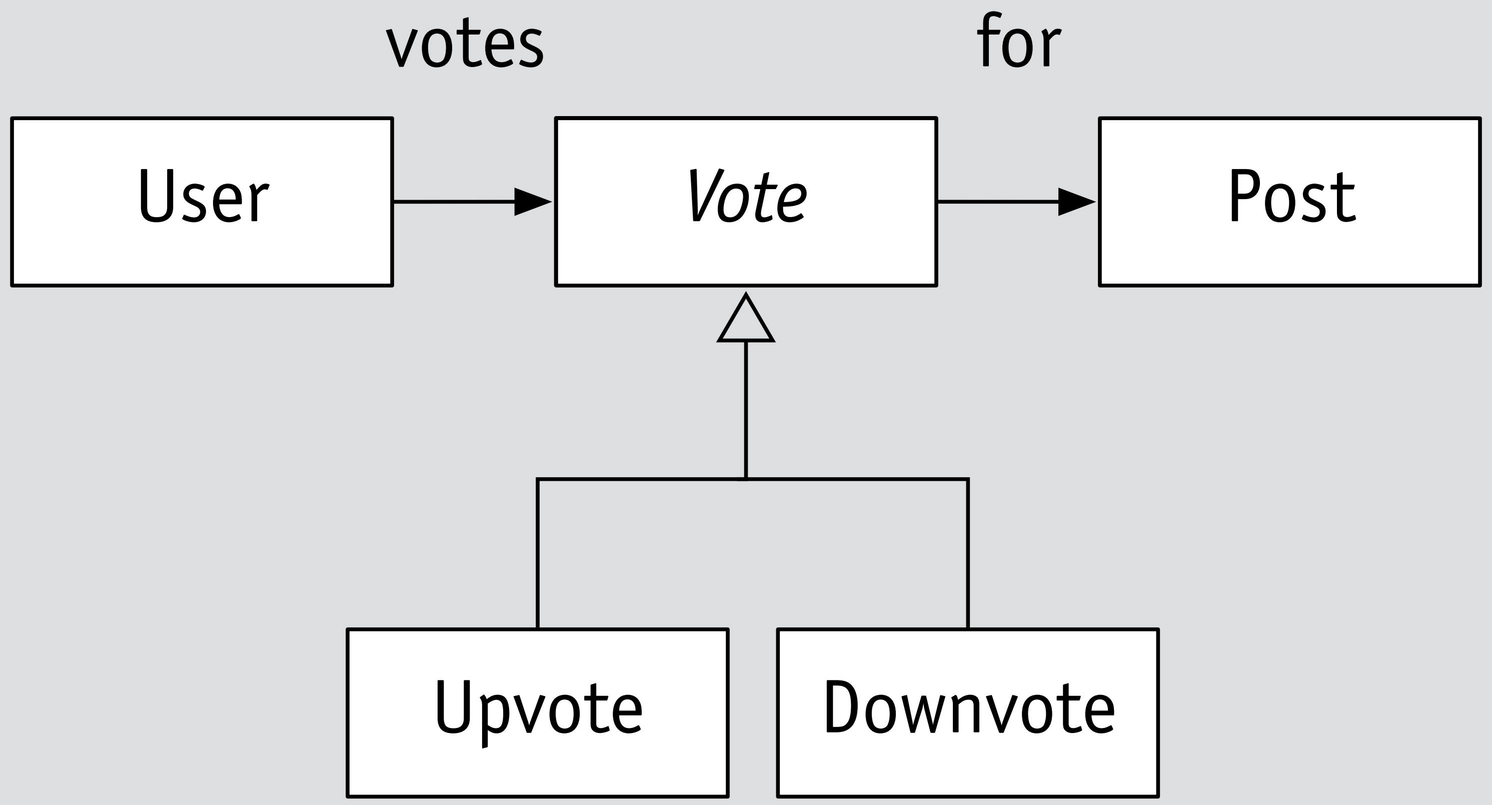
maar
lake
pond
tarn
pool

trickle
runnel
stream
river
rivulet
torrent

lagoon
sea

upvotes and downvotes
are votes and then what?

entity in data model



votes

for

User

Vote

Post

Upvote

Downvote

but concept is in the
relationships, not the entities!

a concept has a **name**

```
concept Upvote
```

a concept has a **name**

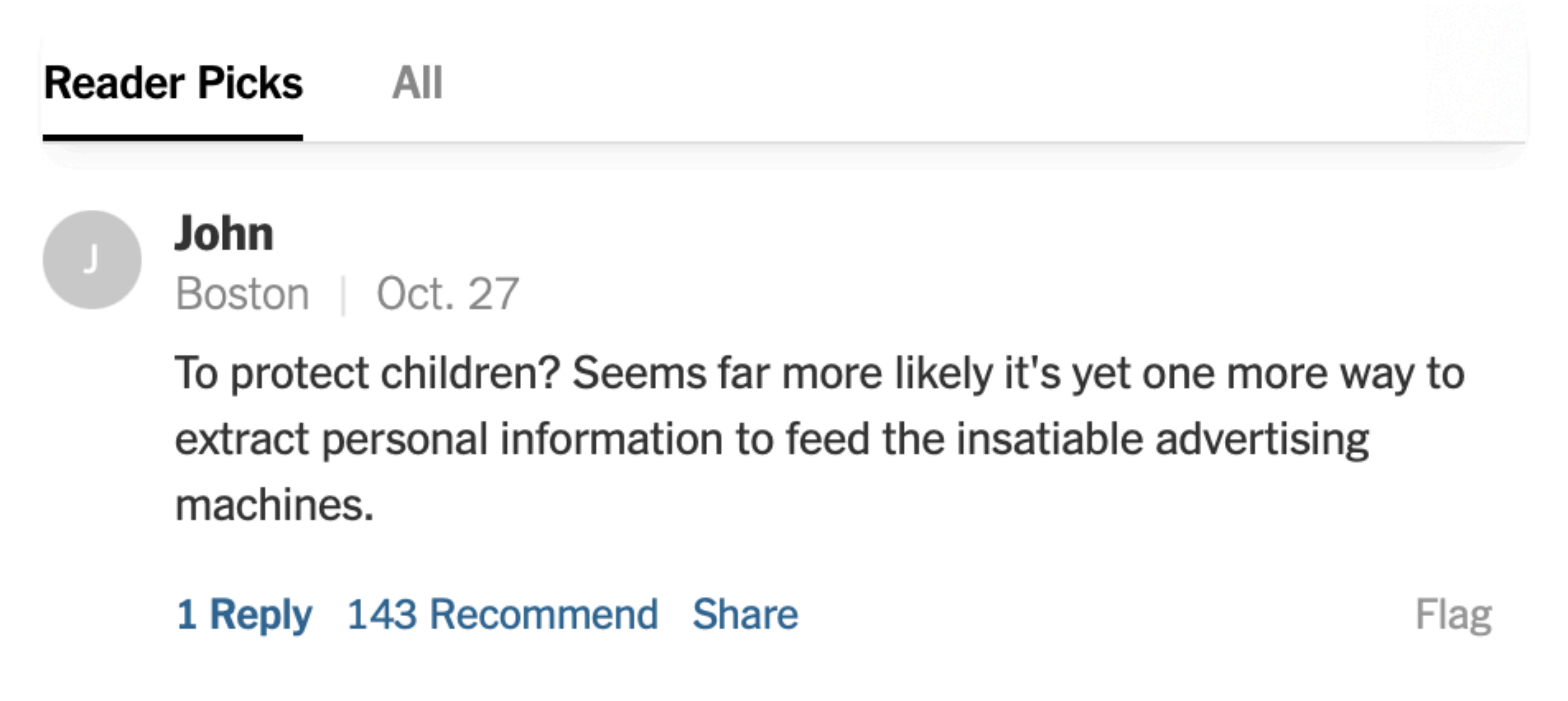
concept Upvote

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

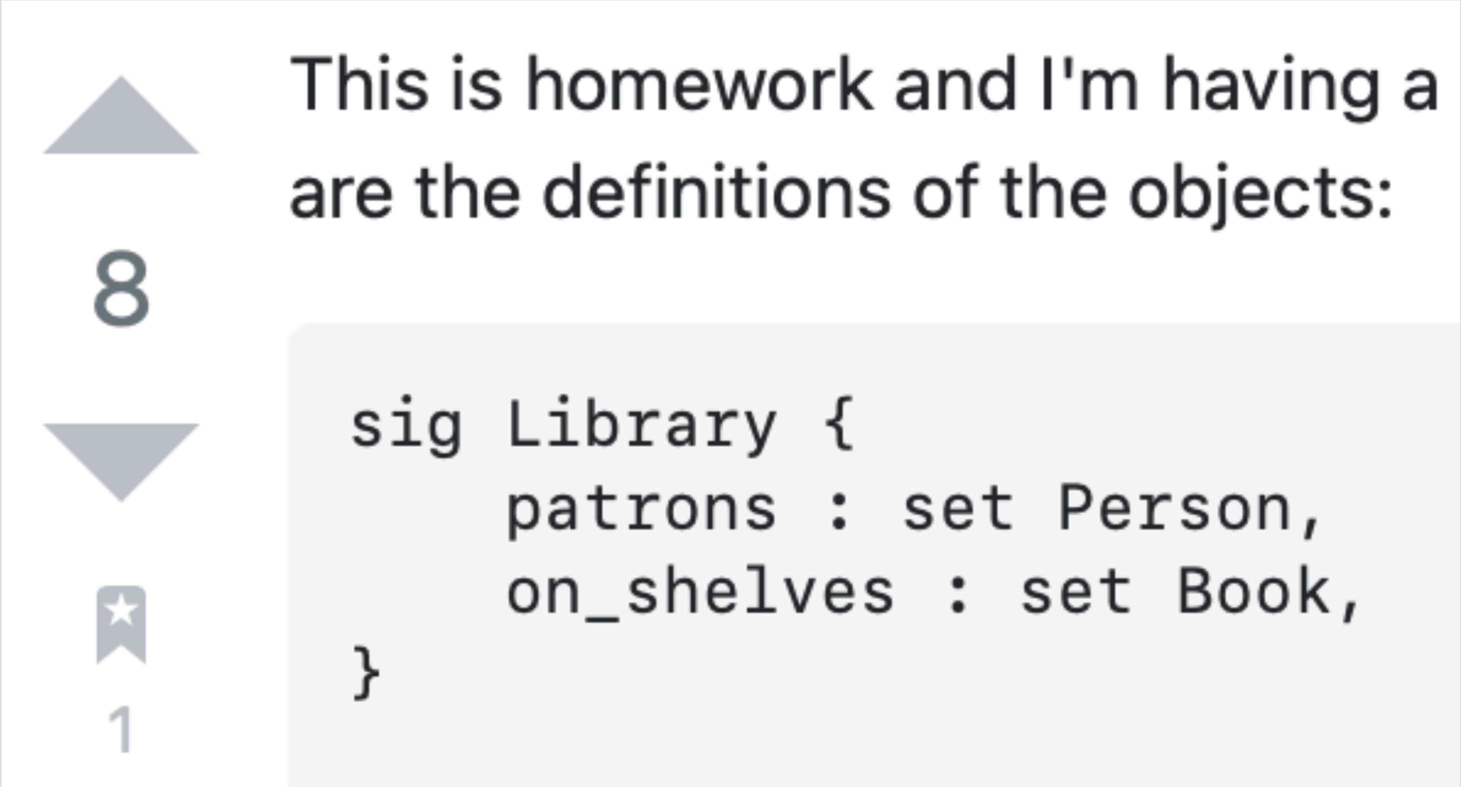
John
Boston | Oct. 27

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

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:

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

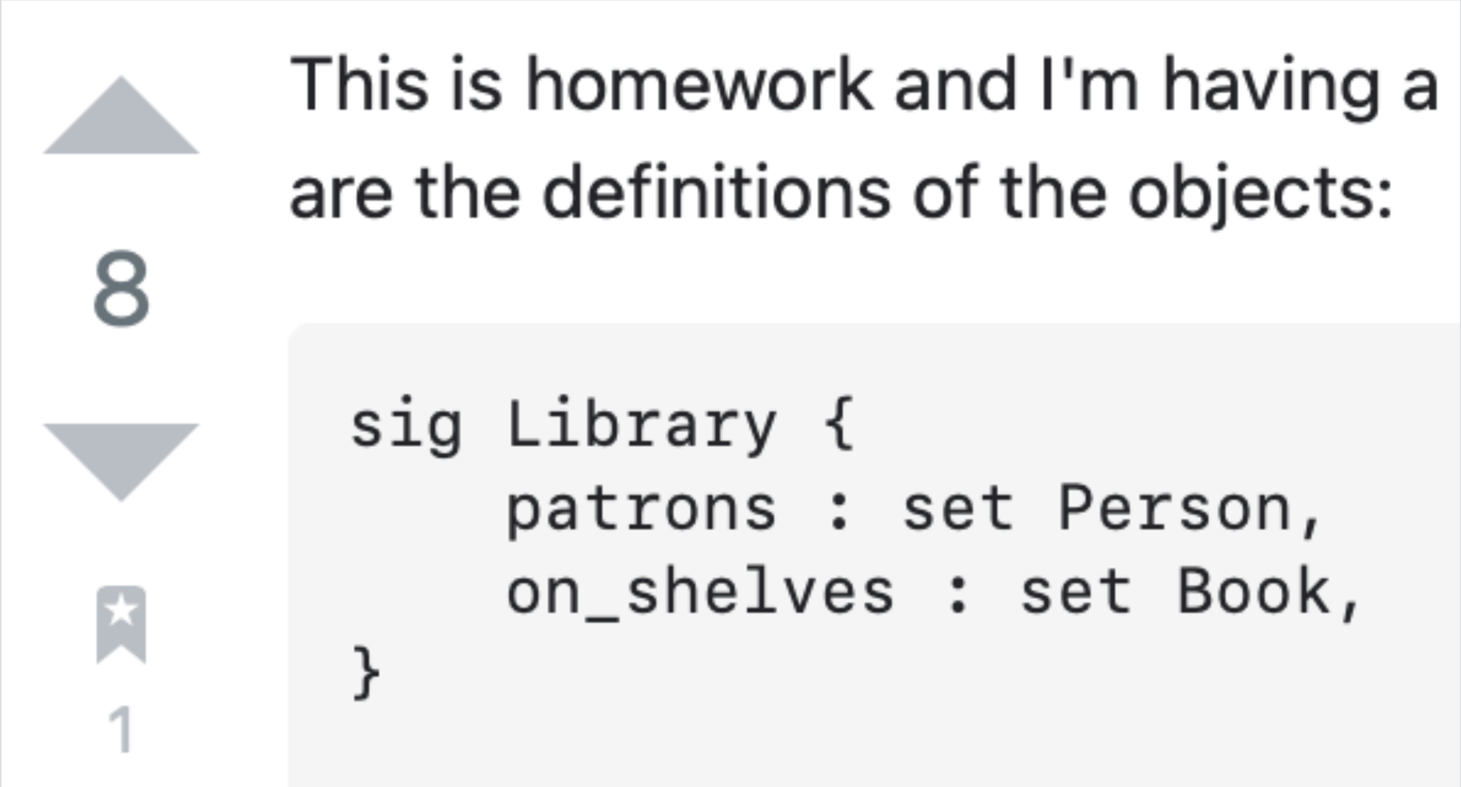
a concept has a purpose

concept Upvote

purpose rank items by popularity

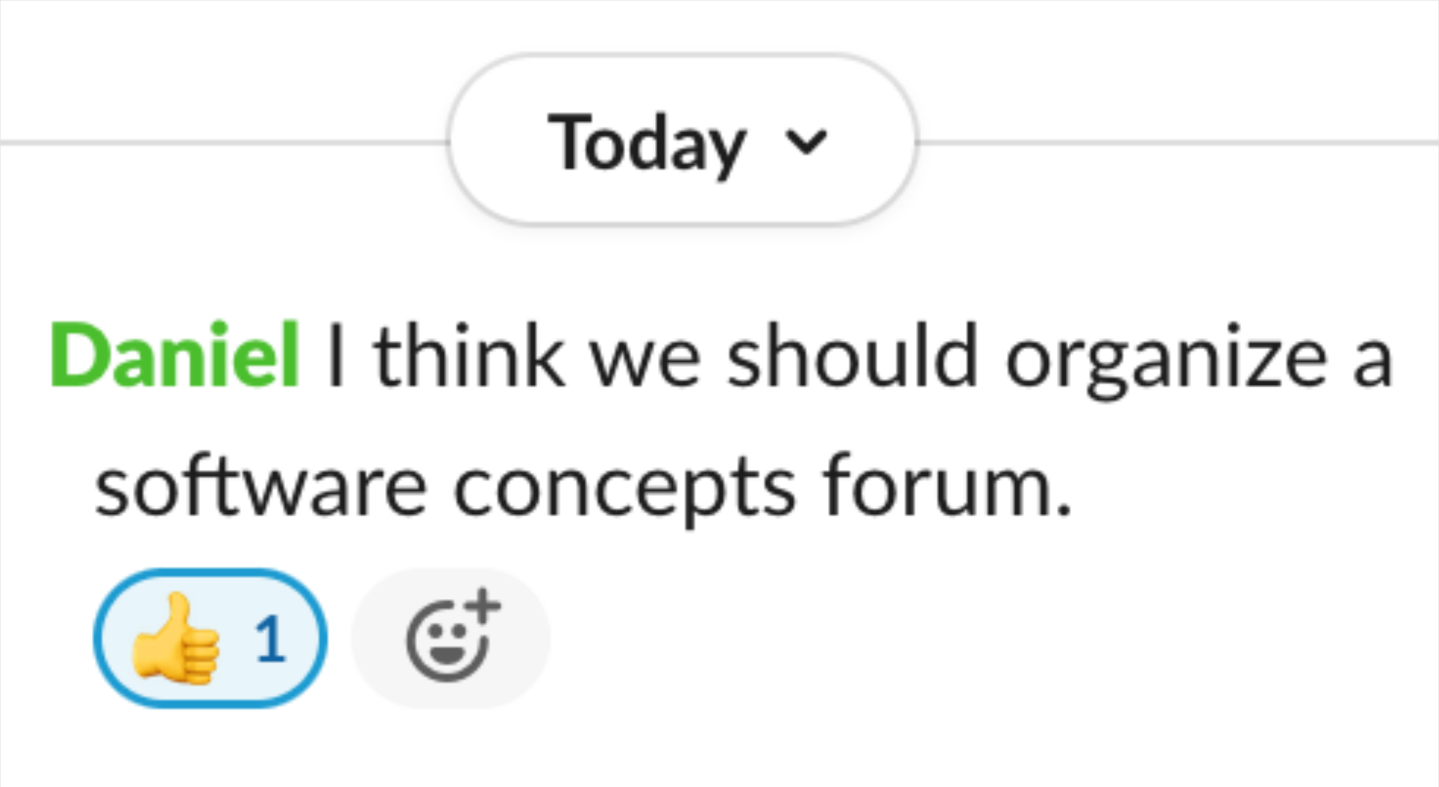
concept Reaction

purpose send reactions to author



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

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

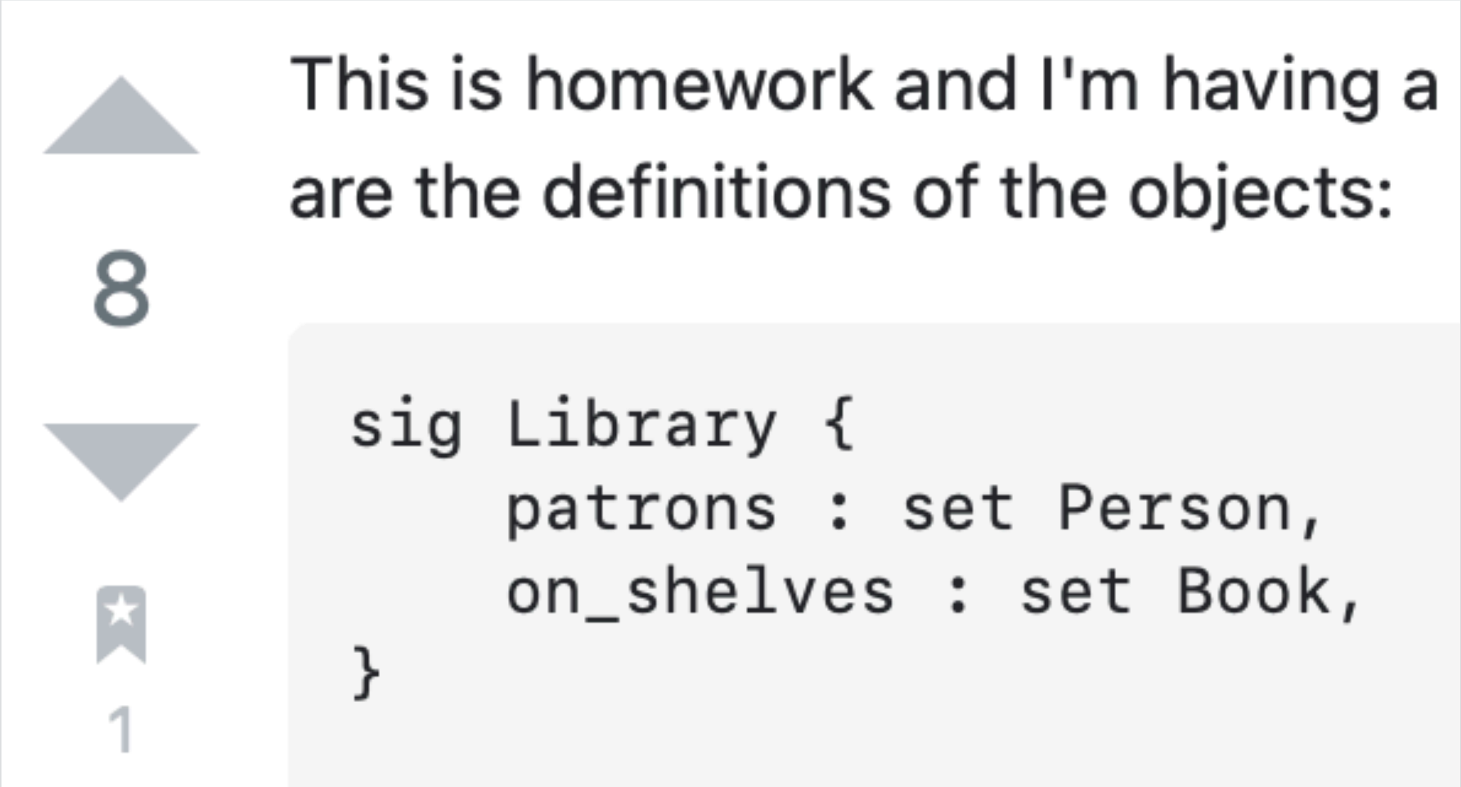


Daniel I think we should organize a software concepts forum.

a concept has a purpose

concept Upvote

purpose rank items by popularity

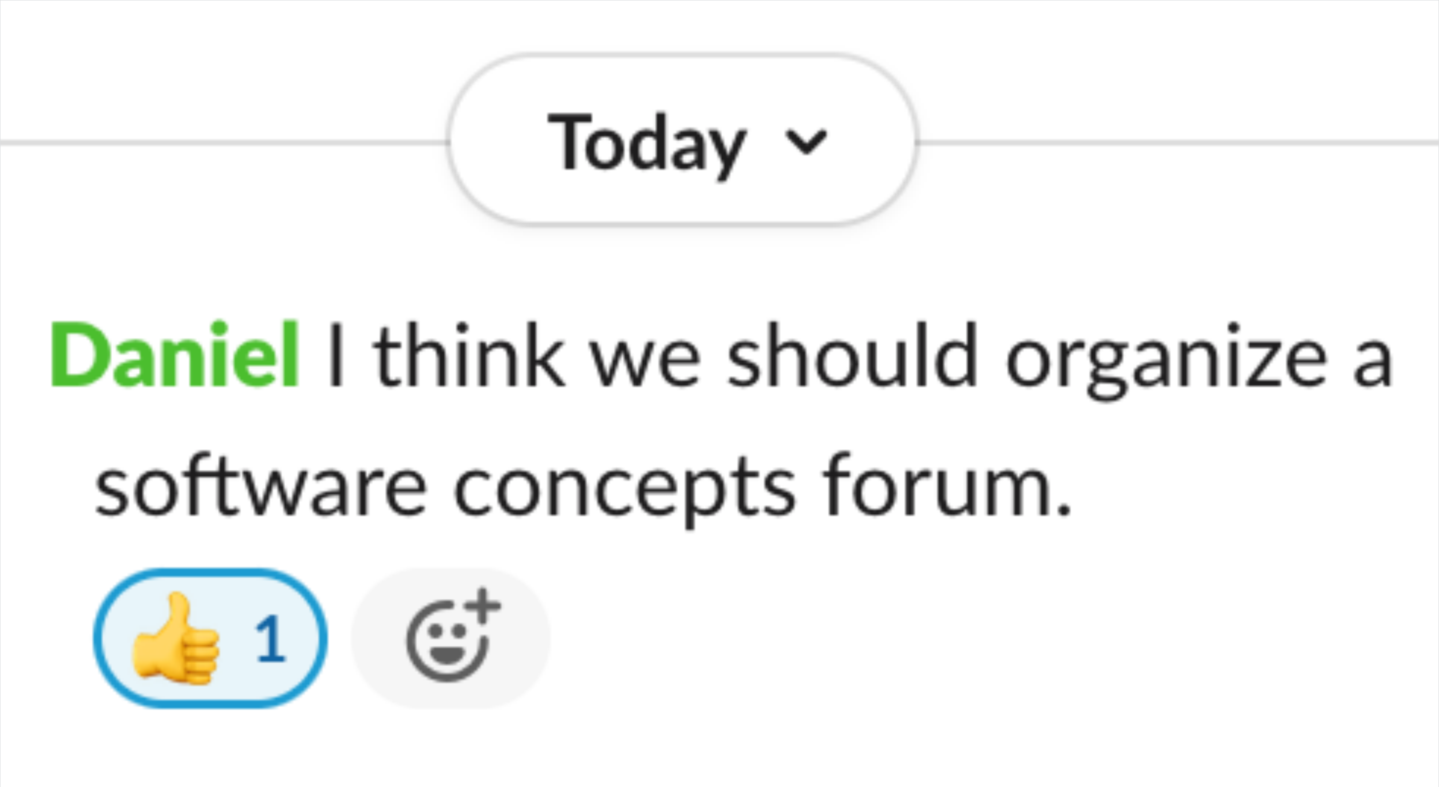


This is homework and I'm having a
are the definitions of the objects:

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

concept Reaction

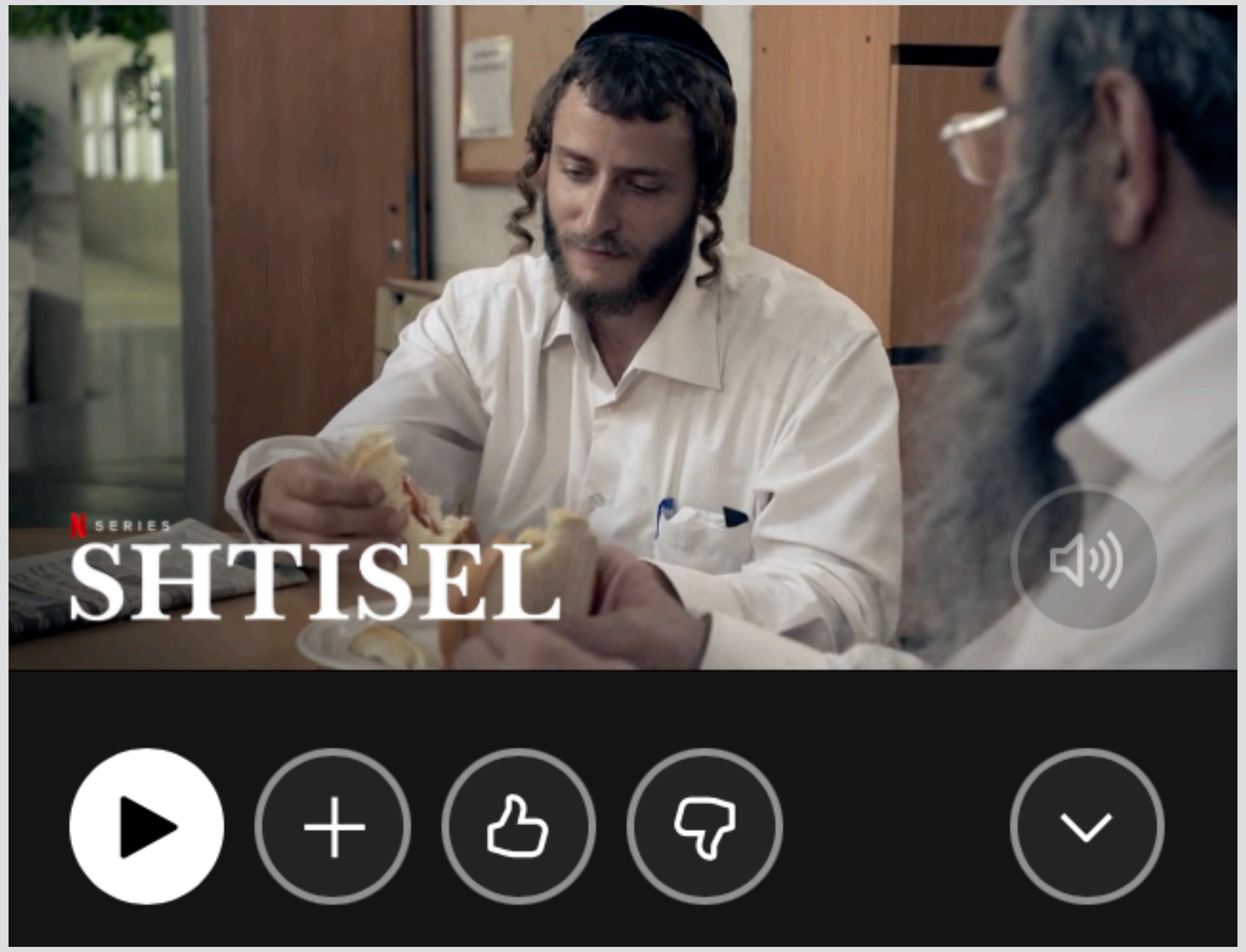
purpose send reactions to author



Daniel I think we should organize a software concepts forum.

concept Recommendation

purpose use prior likes to recommend

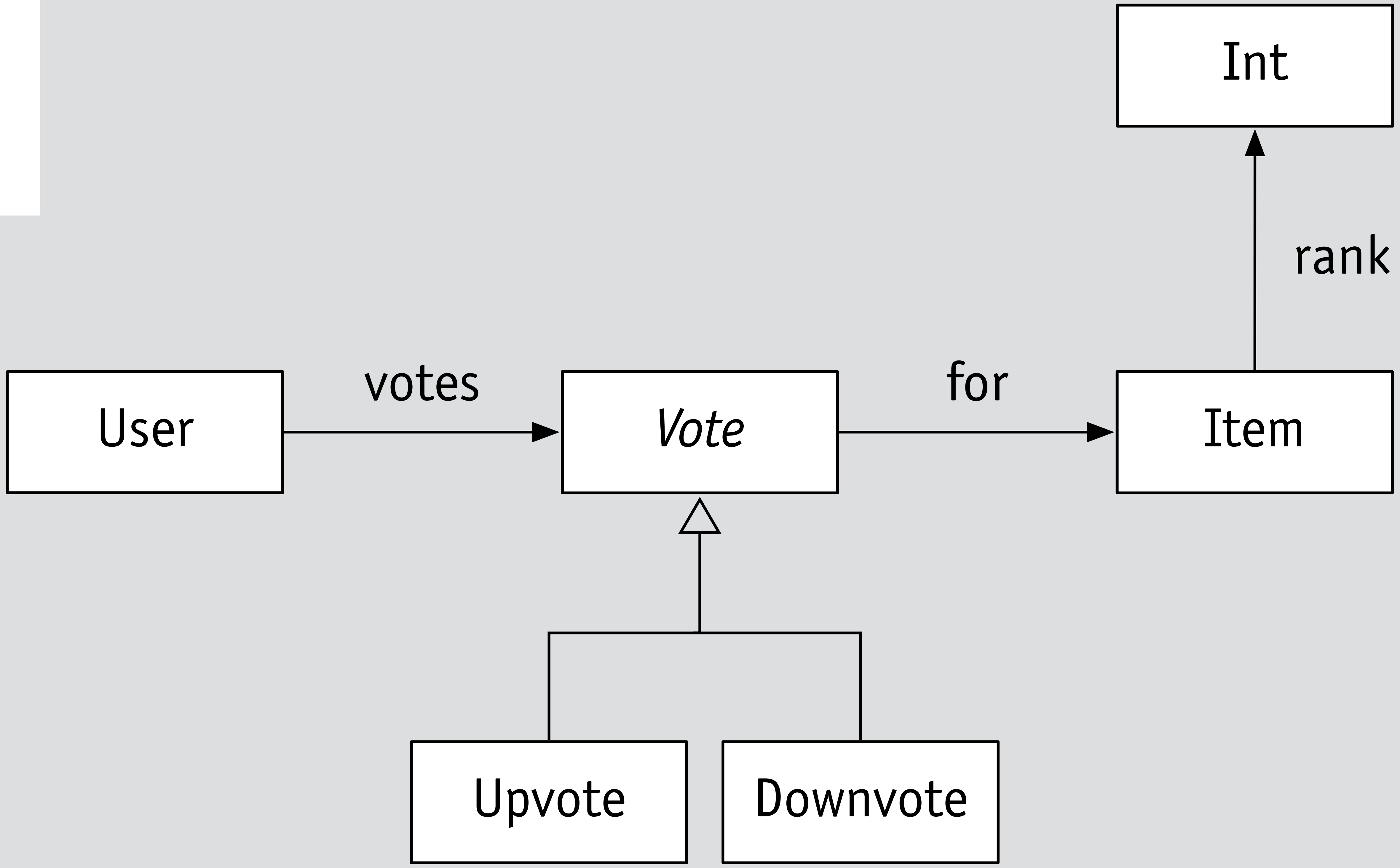


a concept has a **state**

concept Upvote

purpose rank items by popularity

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



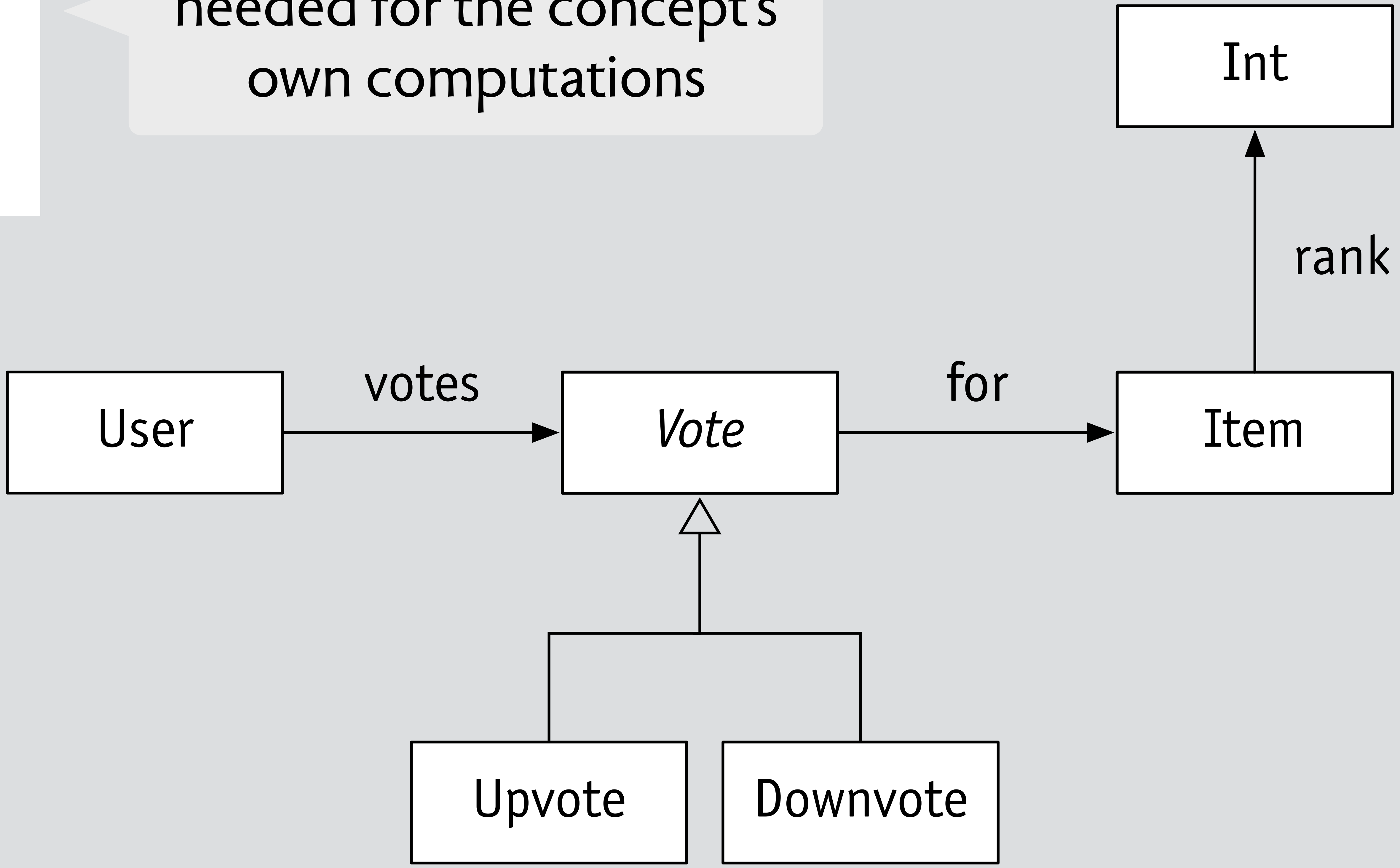
a concept has a **state**

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



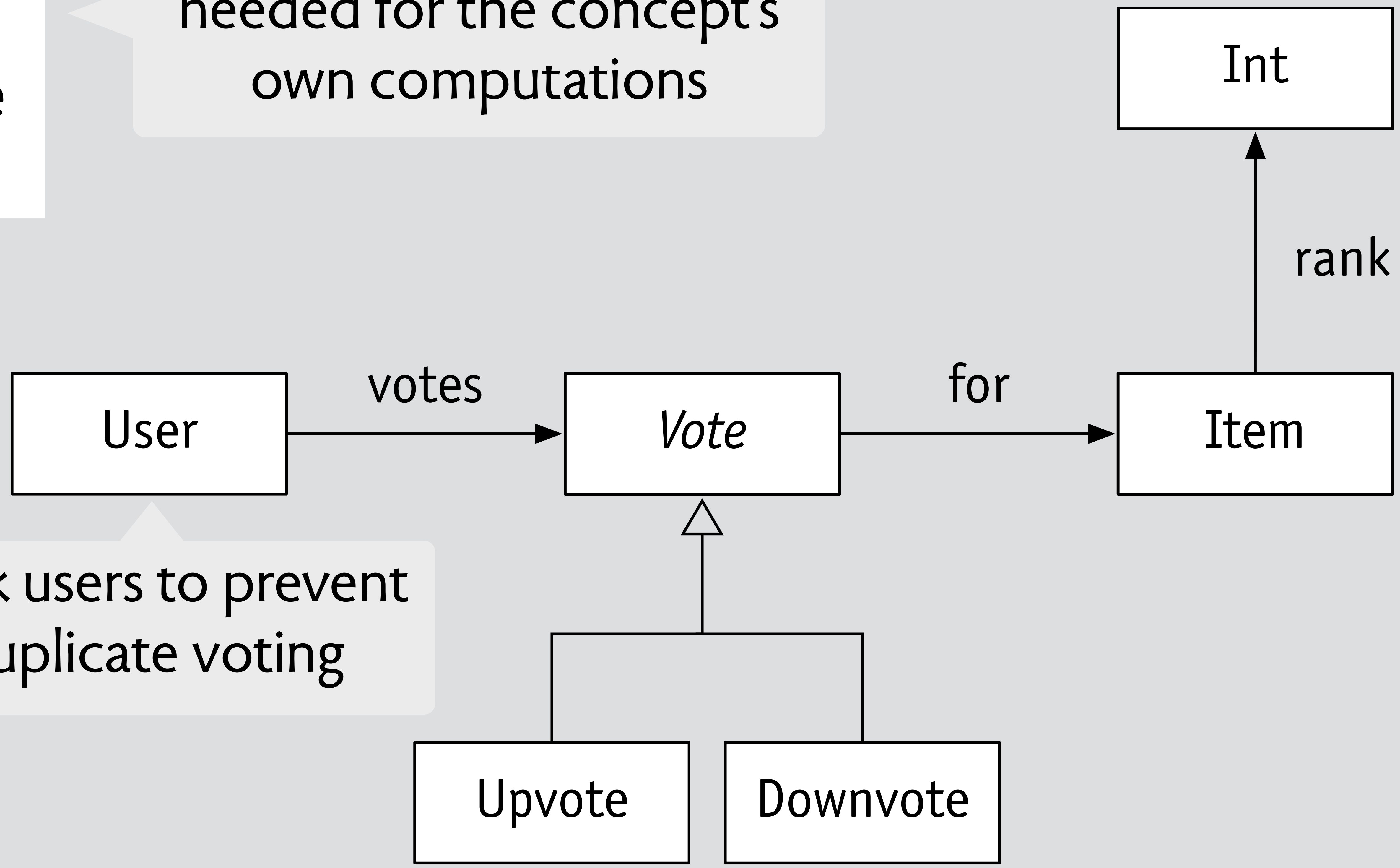
a concept has a **state**

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

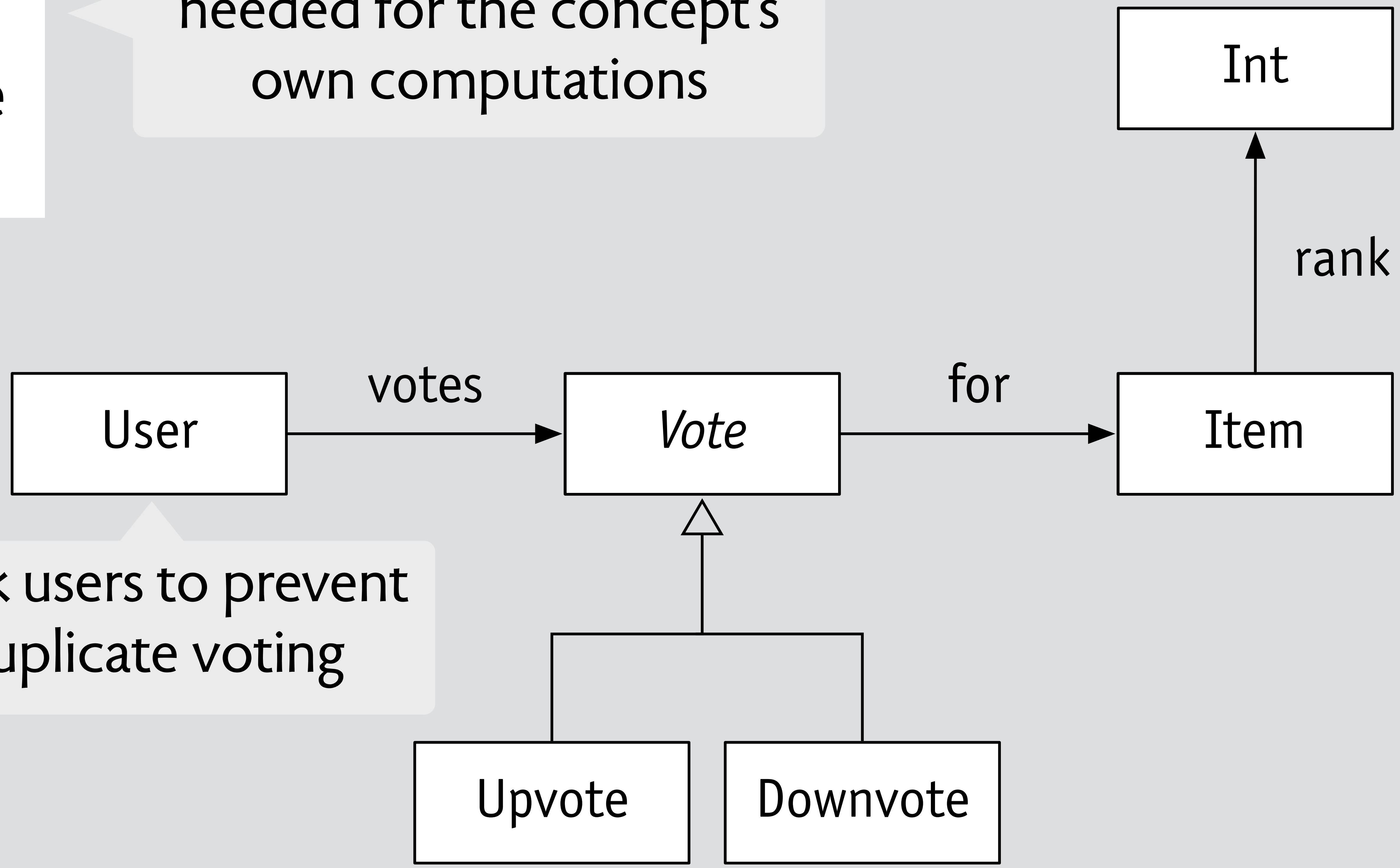
a concept has a **state**

concept Upvote

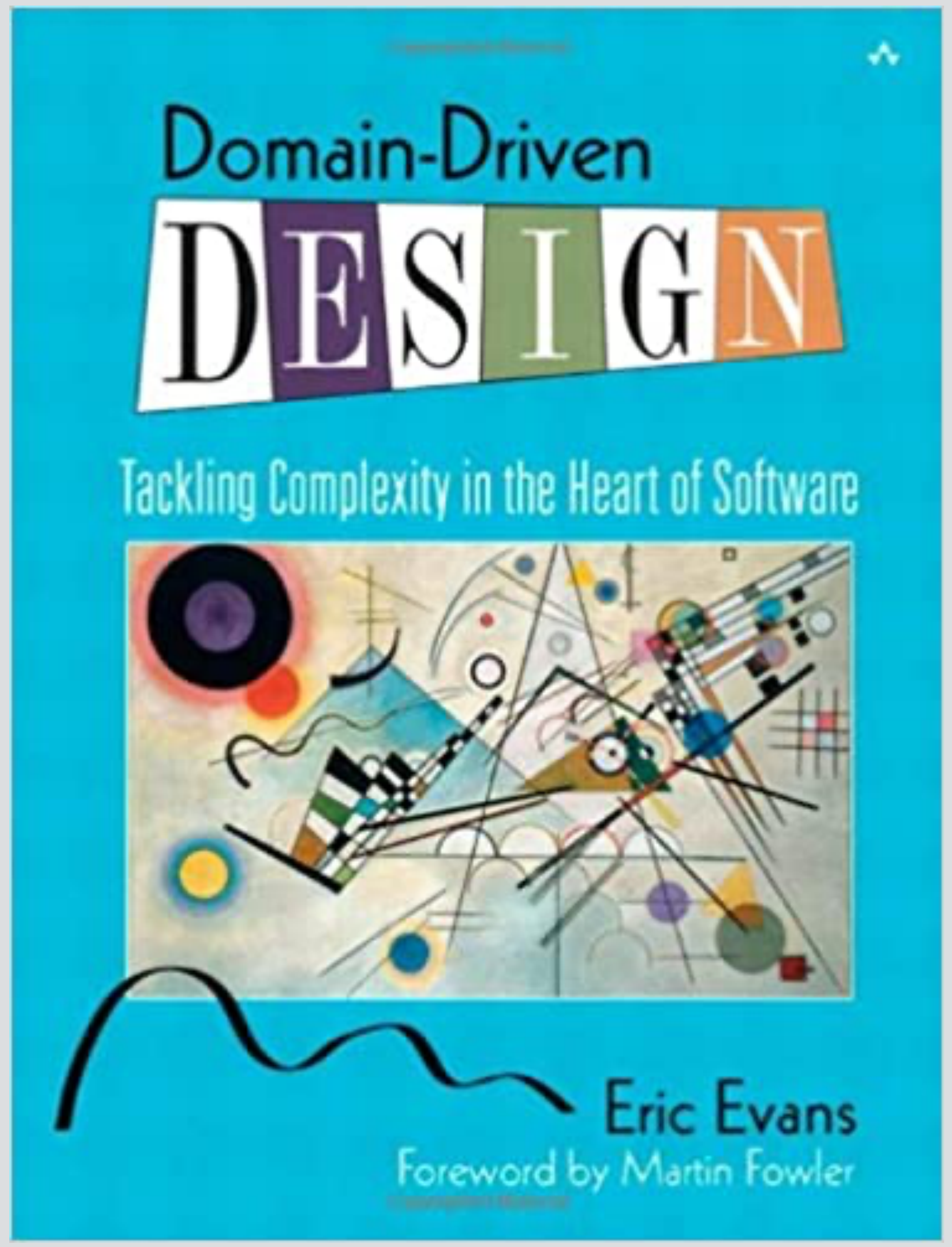
purpose rank items by popularity

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


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

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

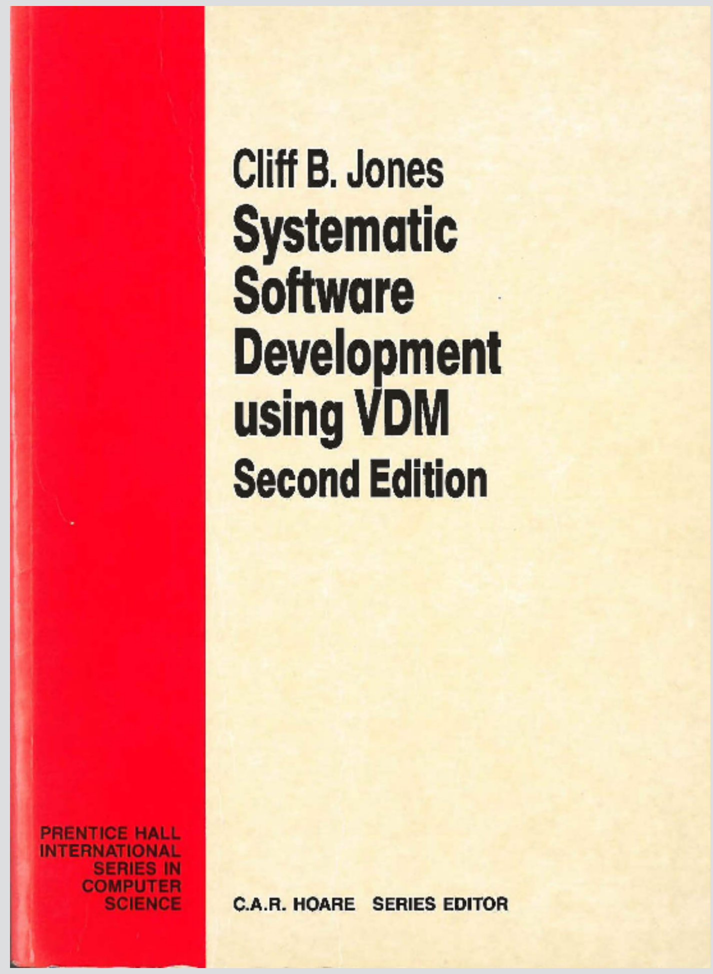
purpose rank items by popularity

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 ...
```


Cliff B. Jones
Systematic Software Development using VDM
Second Edition
PRENTICE HALL INTERNATIONAL SERIES IN COMPUTER SCIENCE
C.A.R. HOARE SERIES EDITOR

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

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

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

often used with
Karma, Session, ...

how to
compose
concepts?

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

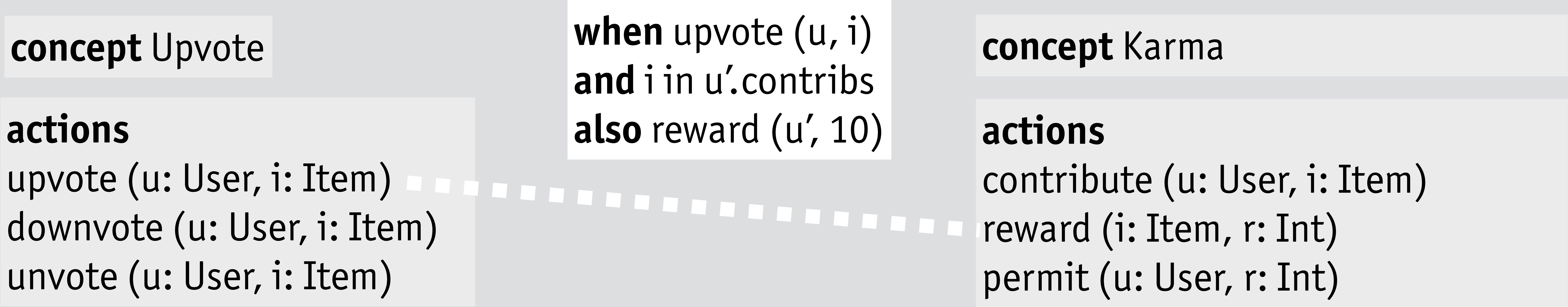
actions
contribute (u: User, i: Item)
reward (u: User, r: Int)
permit (u: User, r: Int)

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)



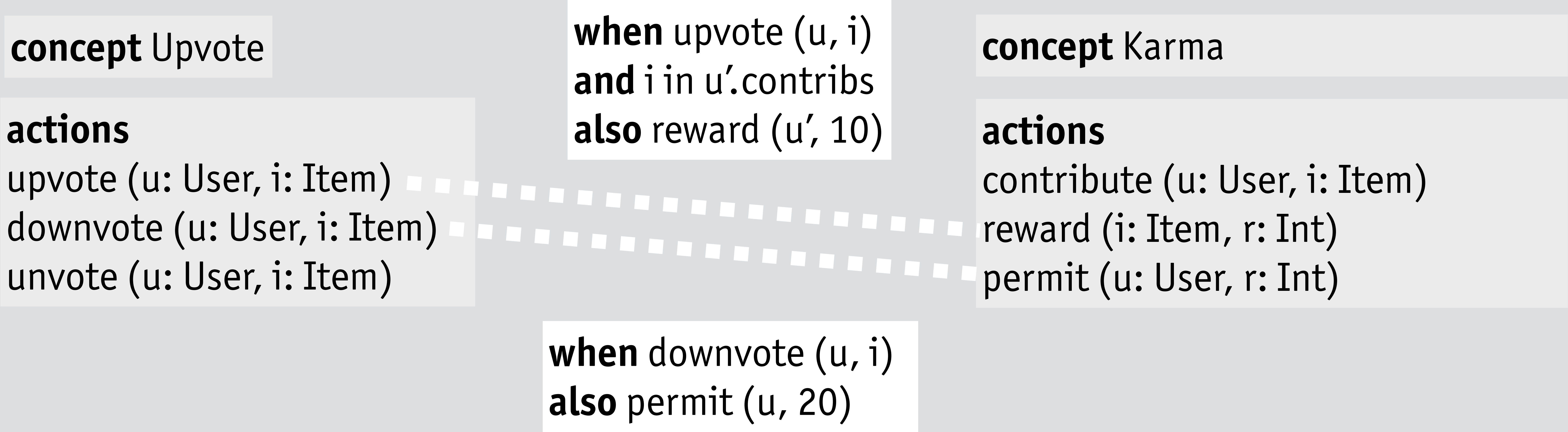
concept Upvote

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

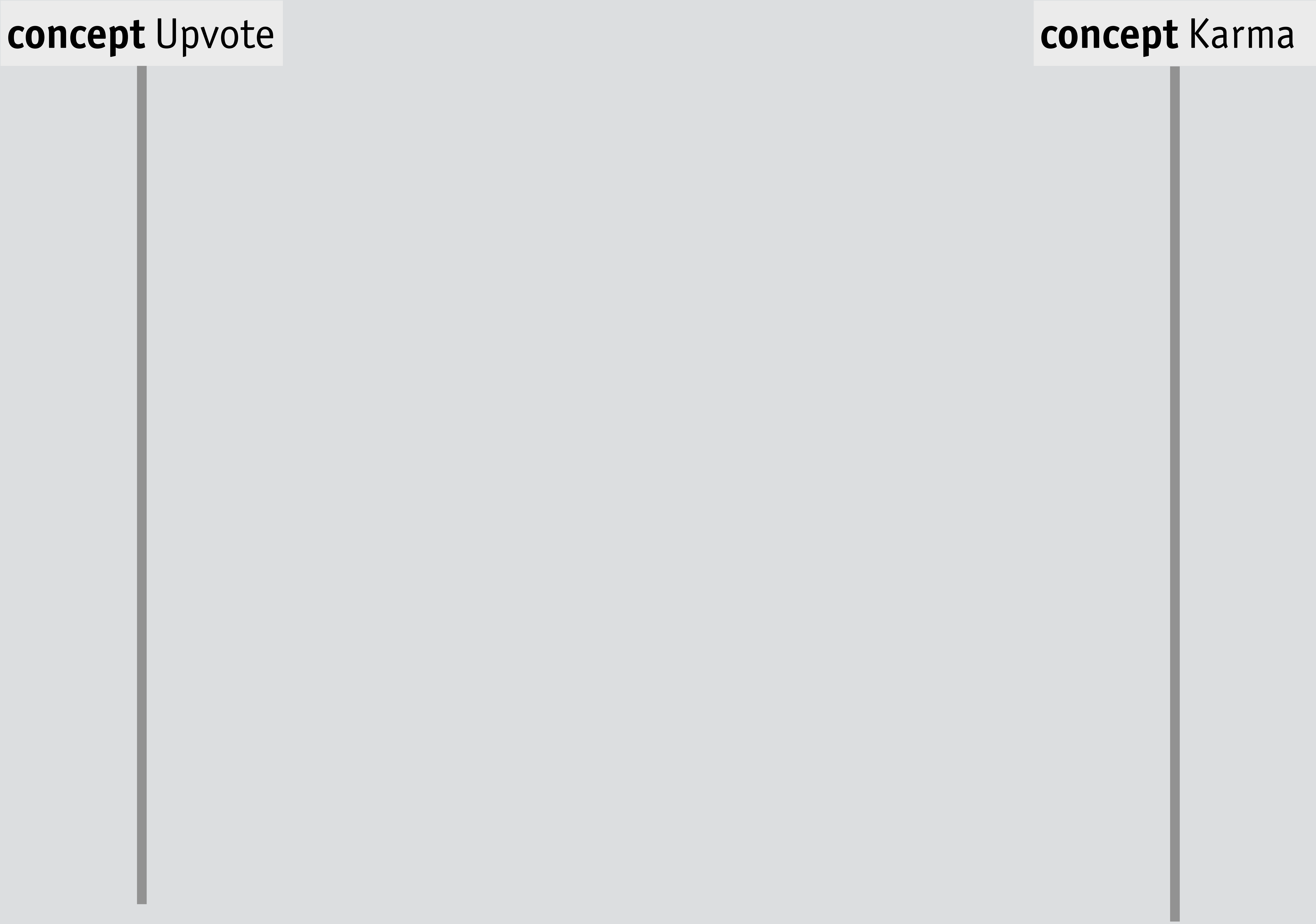
actions
upvote (u: User, i: Item)
downvote (u: User, i: Item)
unvote (u: User, i: Item)

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

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

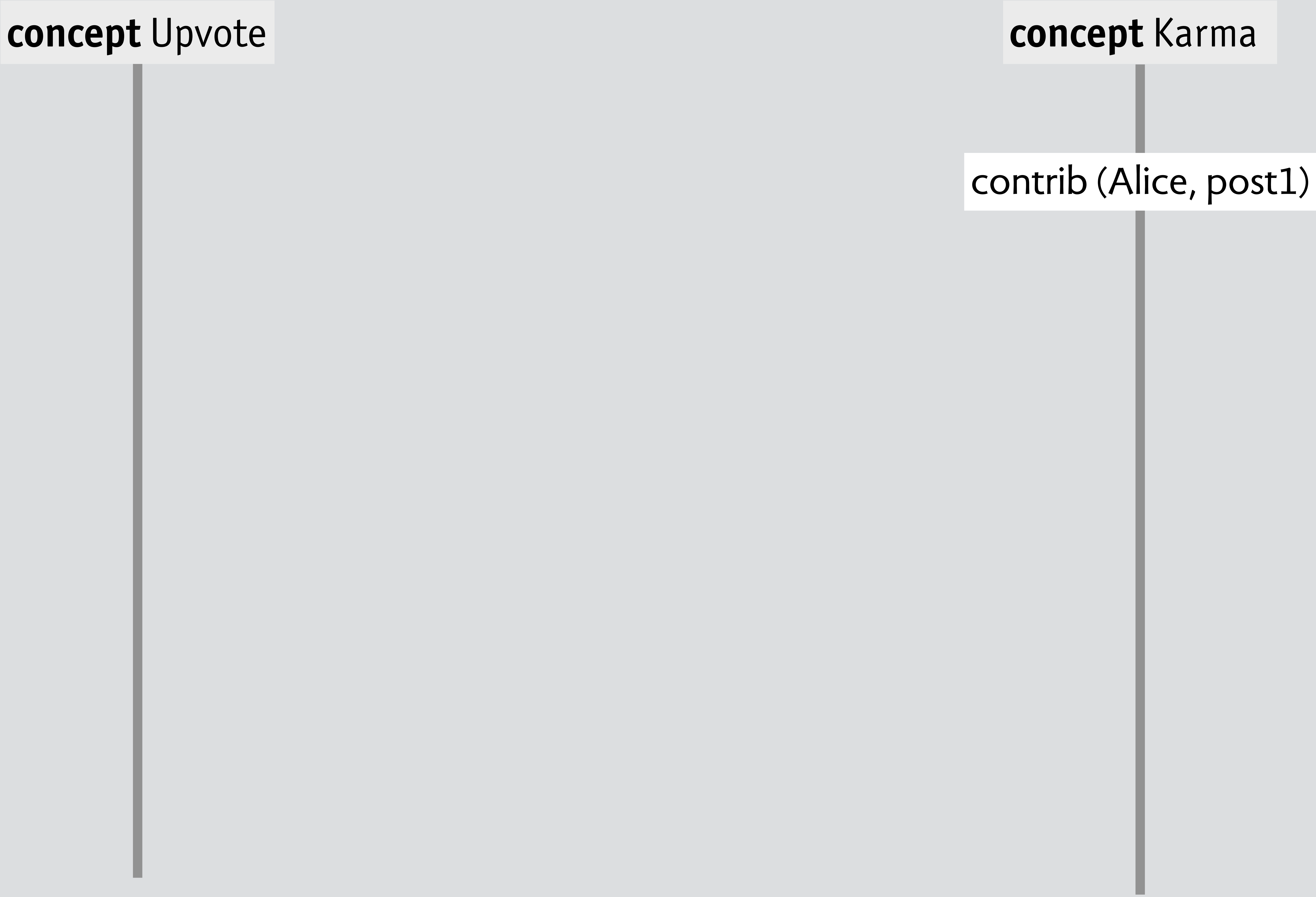
concept Karma

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



concept Upvote

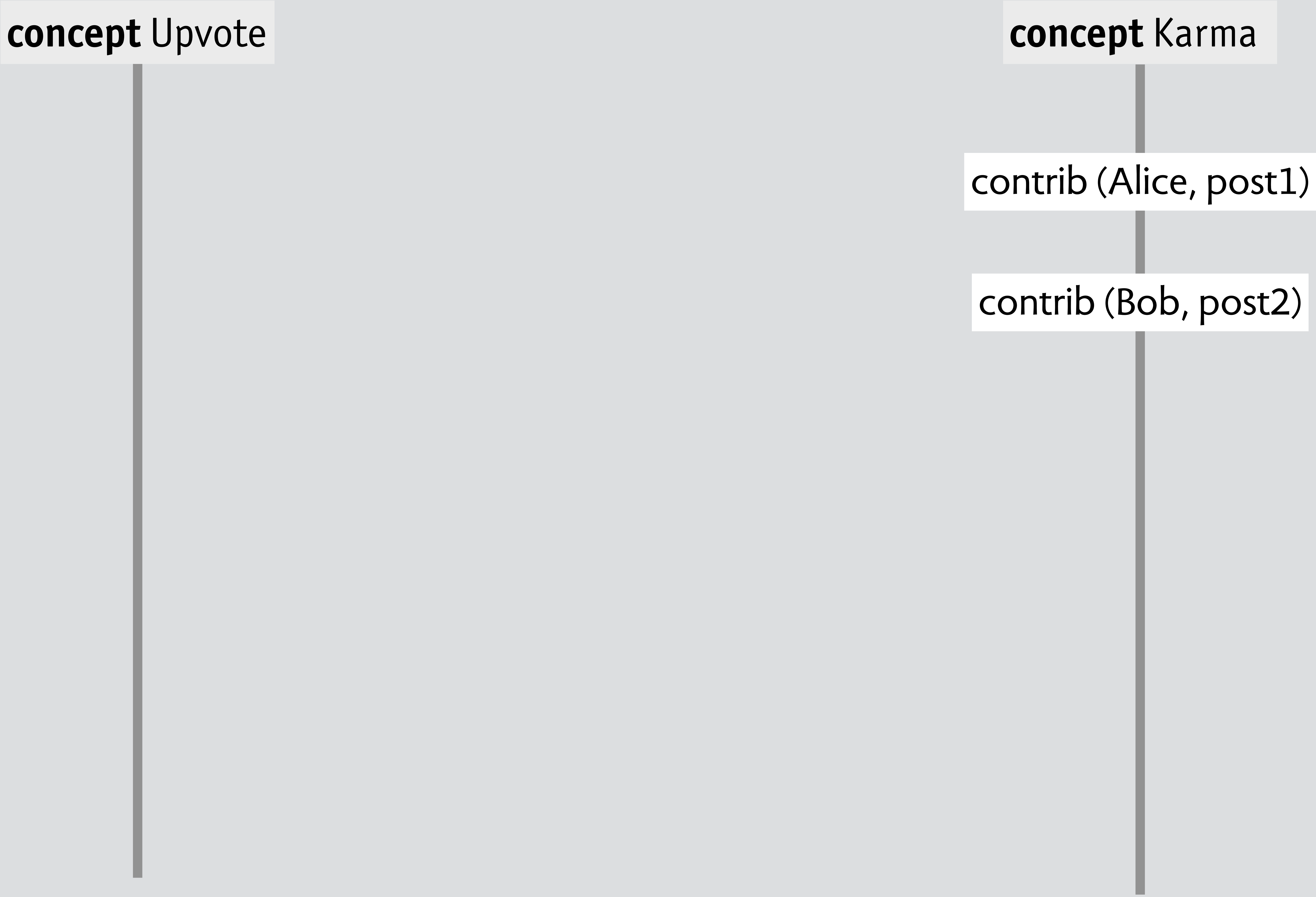
concept Karma



concept Upvote

concept Karma

contrib (Alice, post1)

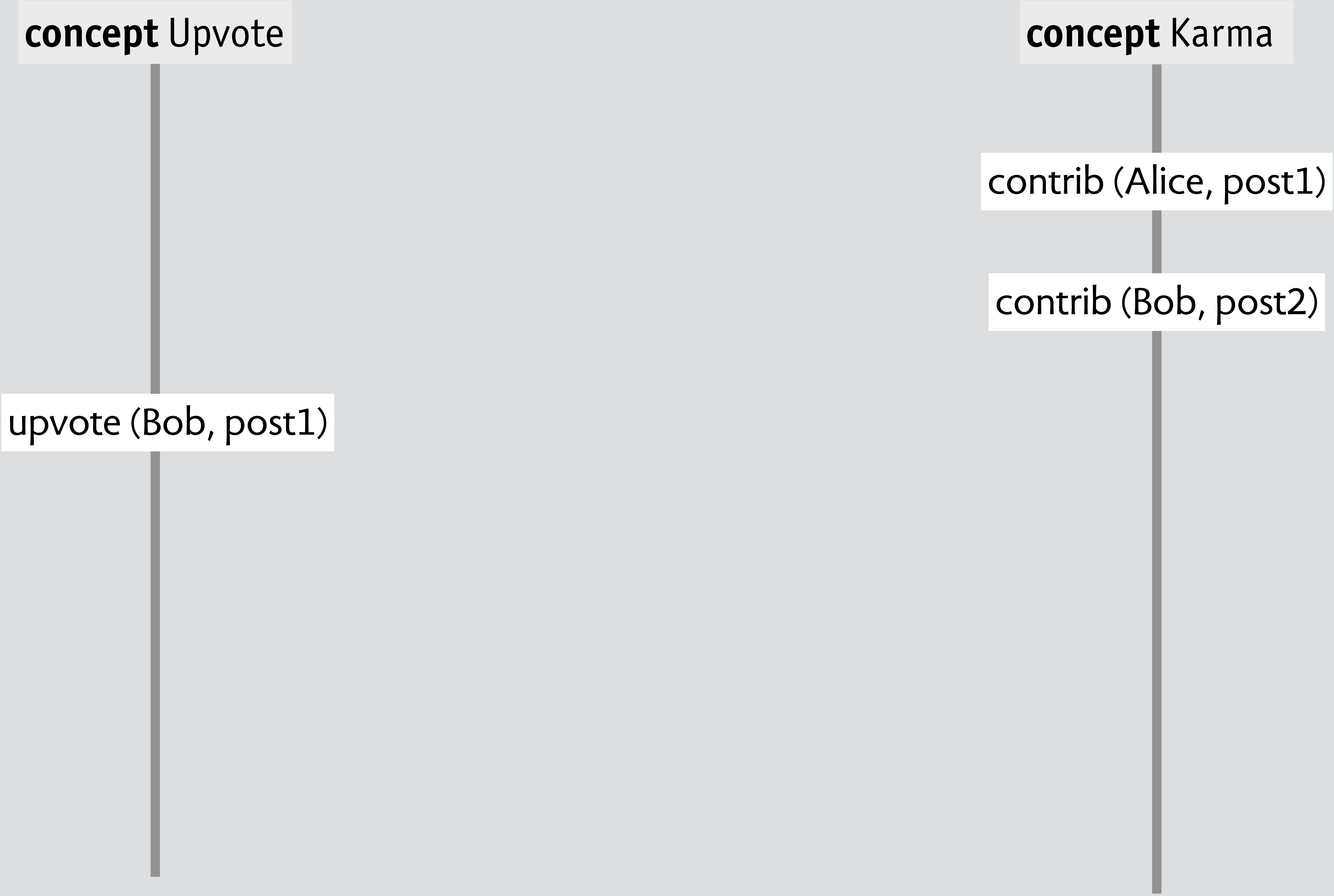


concept Upvote

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)



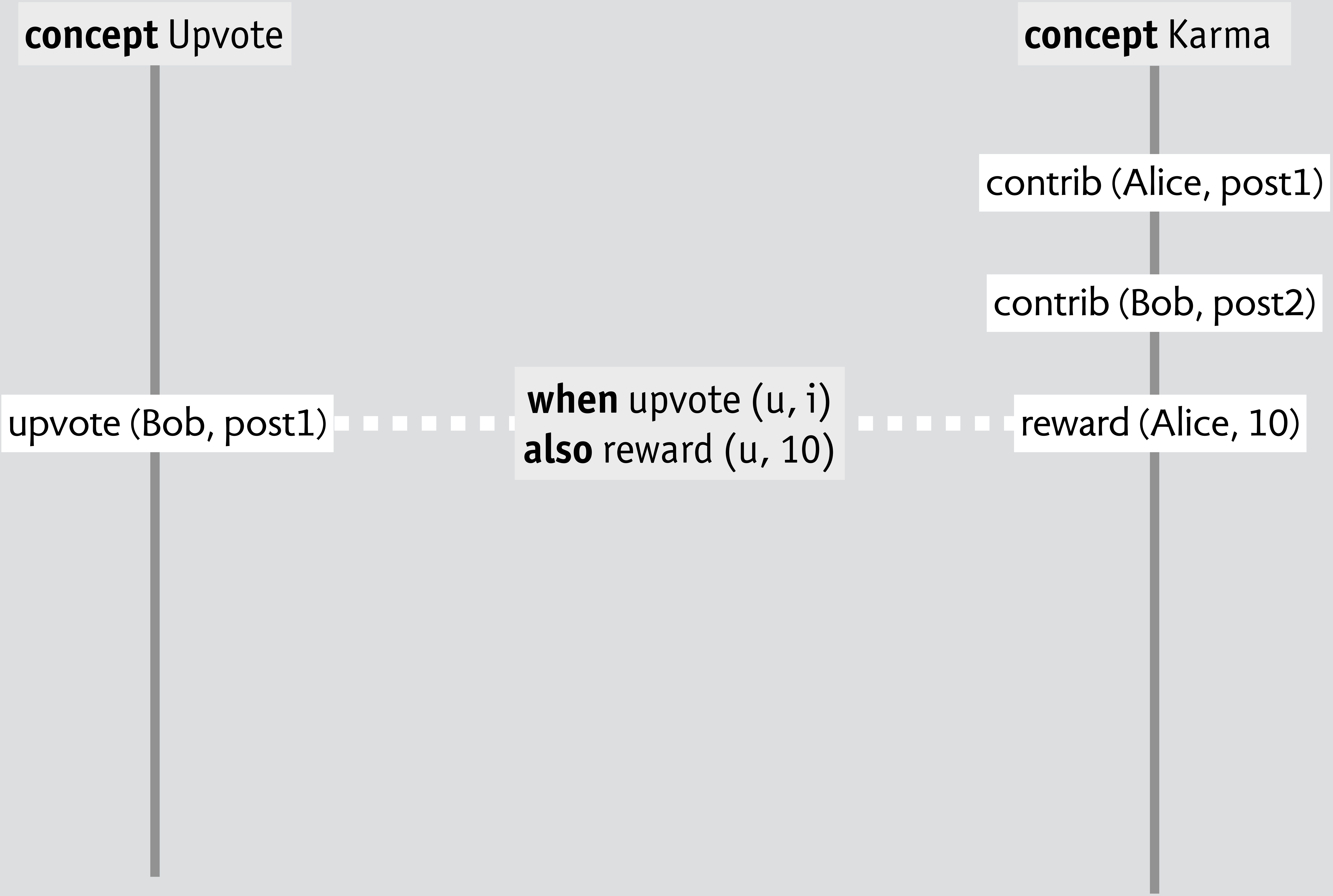
concept Upvote

upvote (Bob, post1)

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)



concept Upvote

concept Karma

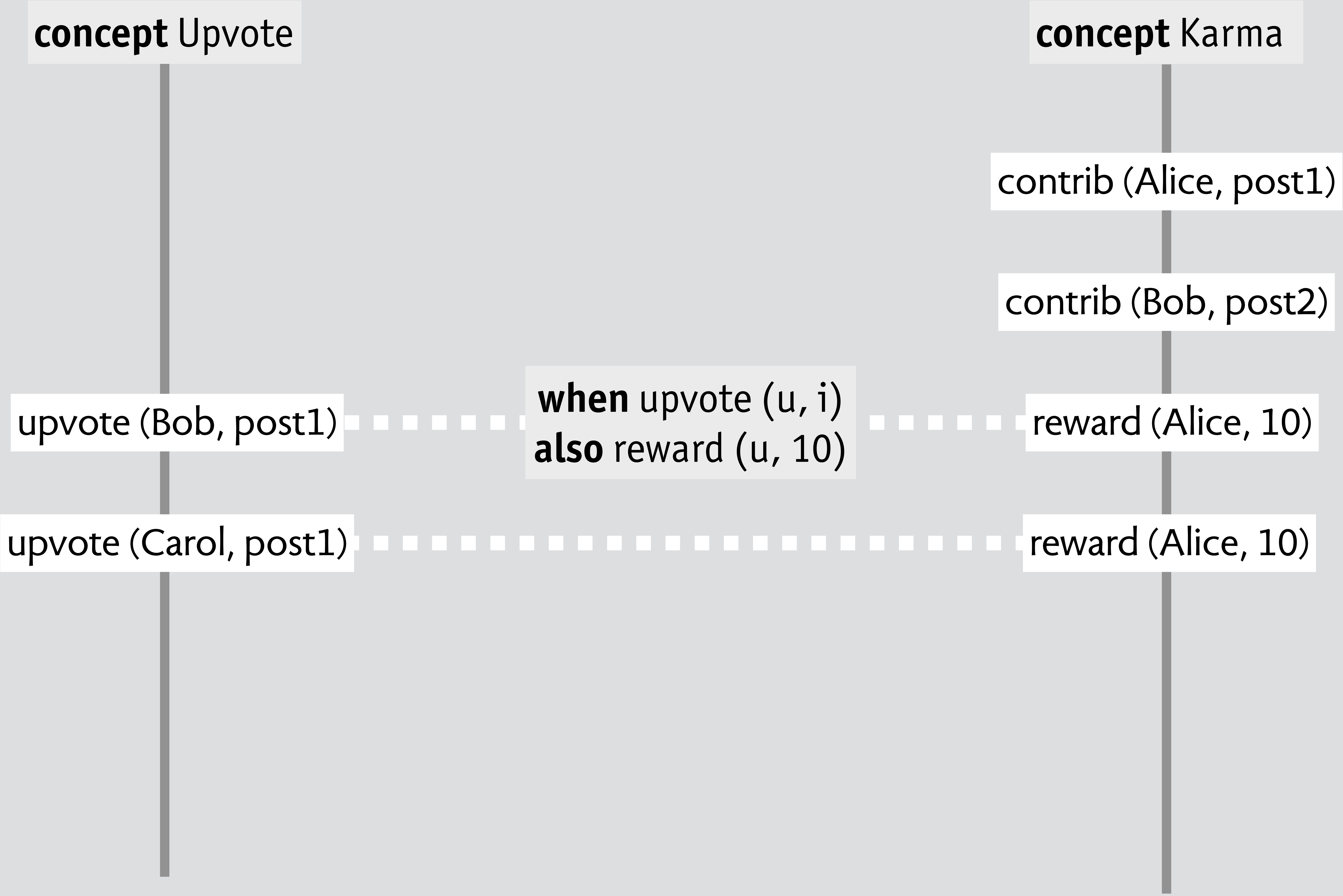
contrib (Alice, post1)

contrib (Bob, post2)

upvote (Bob, post1)

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

reward (Alice, 10)



concept Upvote

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)

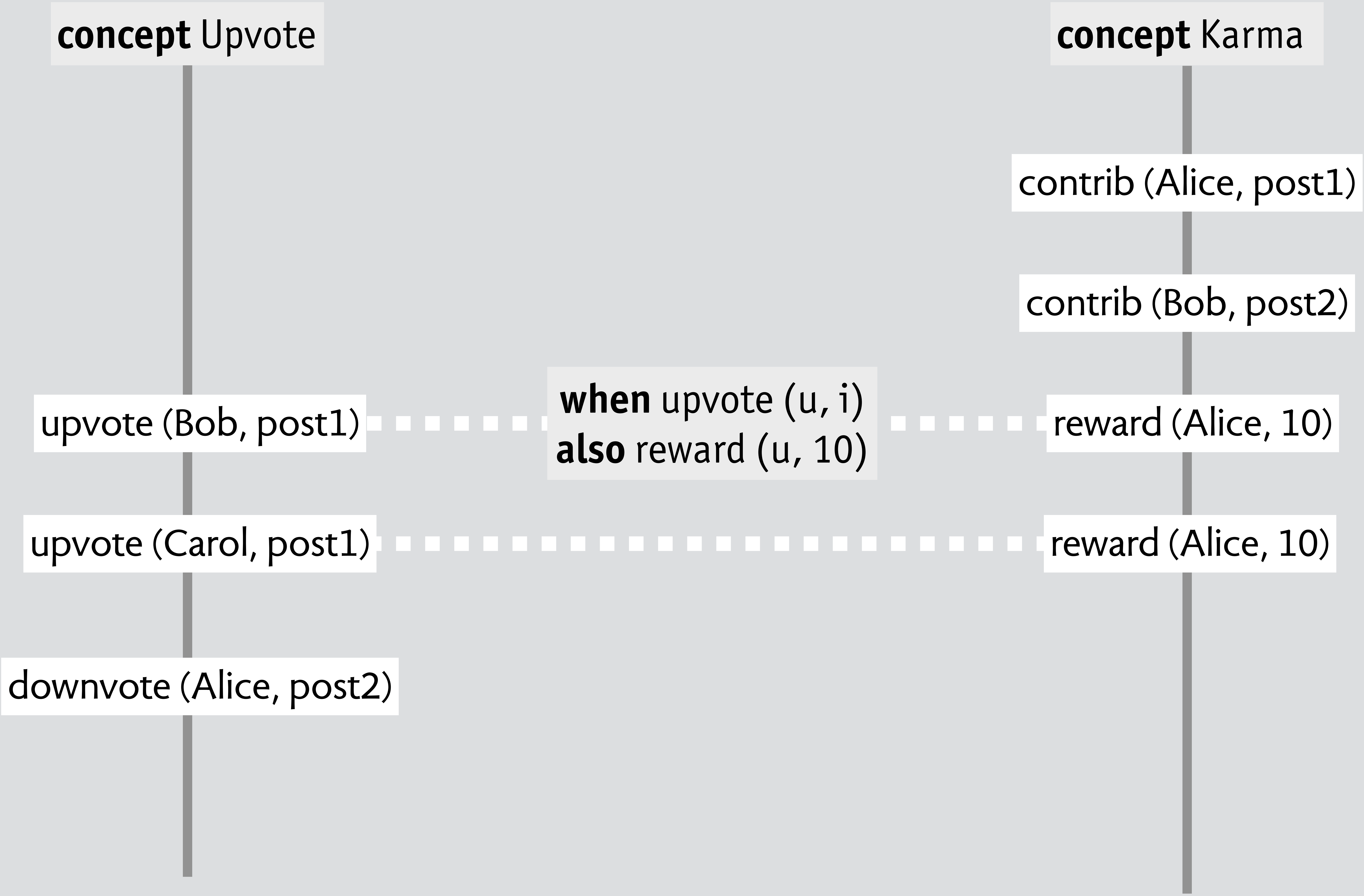
upvote (Bob, post1)

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

reward (Alice, 10)

upvote (Carol, post1)

reward (Alice, 10)



concept Upvote

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)

upvote (Bob, post1)

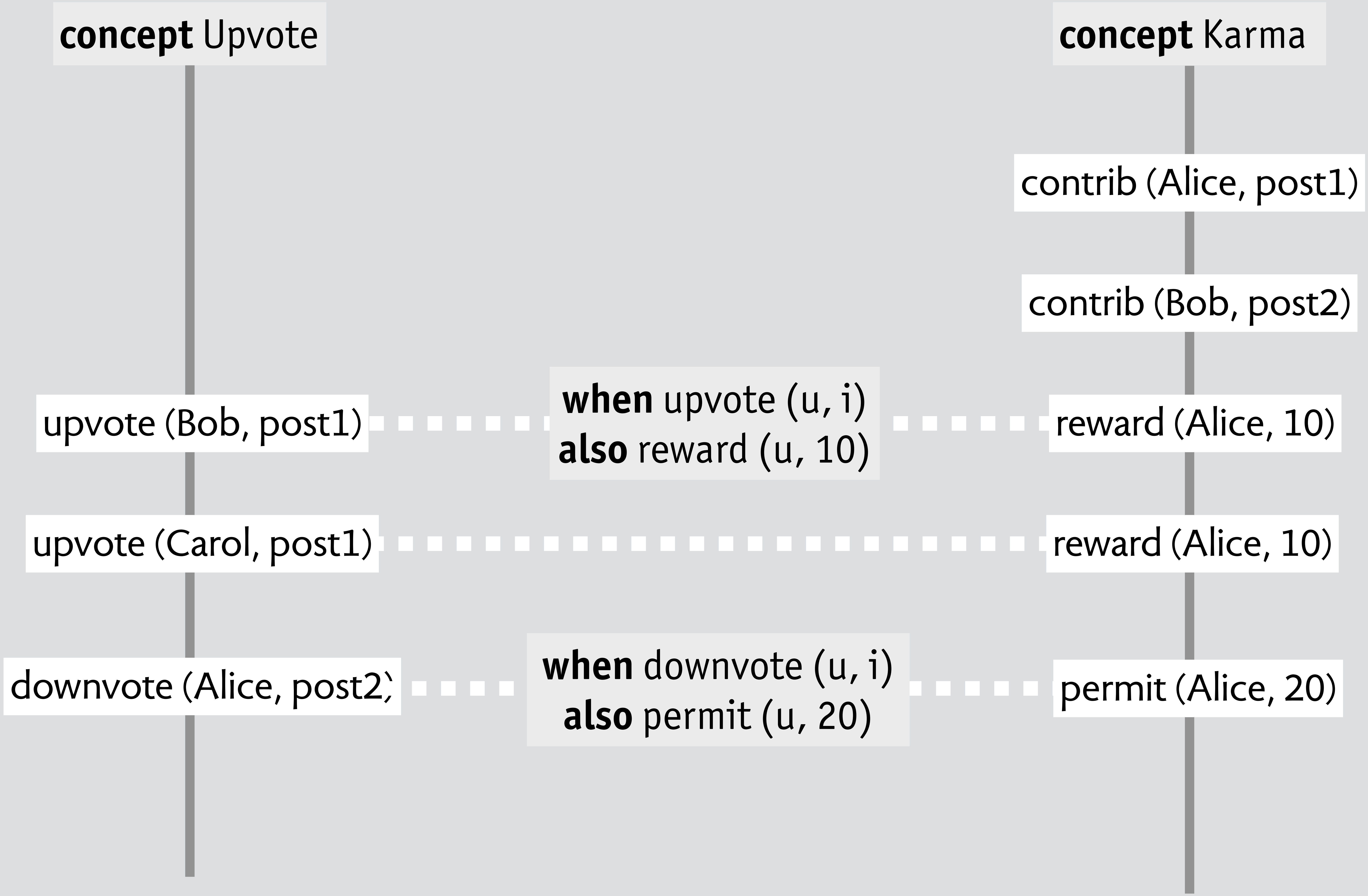
when upvote (u, i)
also reward (u, 10)

reward (Alice, 10)

upvote (Carol, post1)

reward (Alice, 10)

downvote (Alice, post2)



concept Upvote

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)

upvote (Bob, post1)

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

reward (Alice, 10)

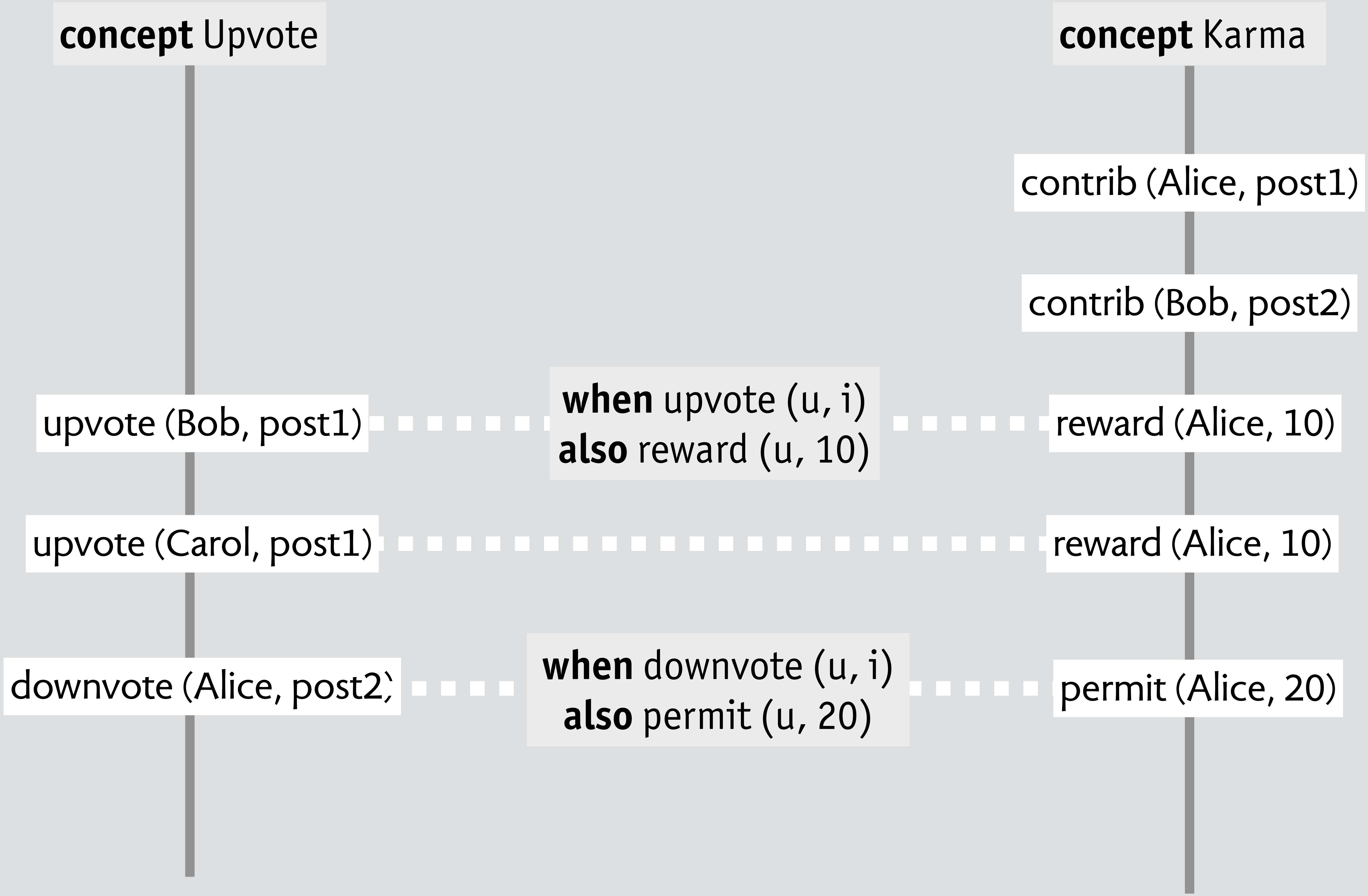
upvote (Carol, post1)

reward (Alice, 10)

downvote (Alice, post2)

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

permit (Alice, 20)



concept Upvote

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)

upvote (Bob, post1)

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

reward (Alice, 10)

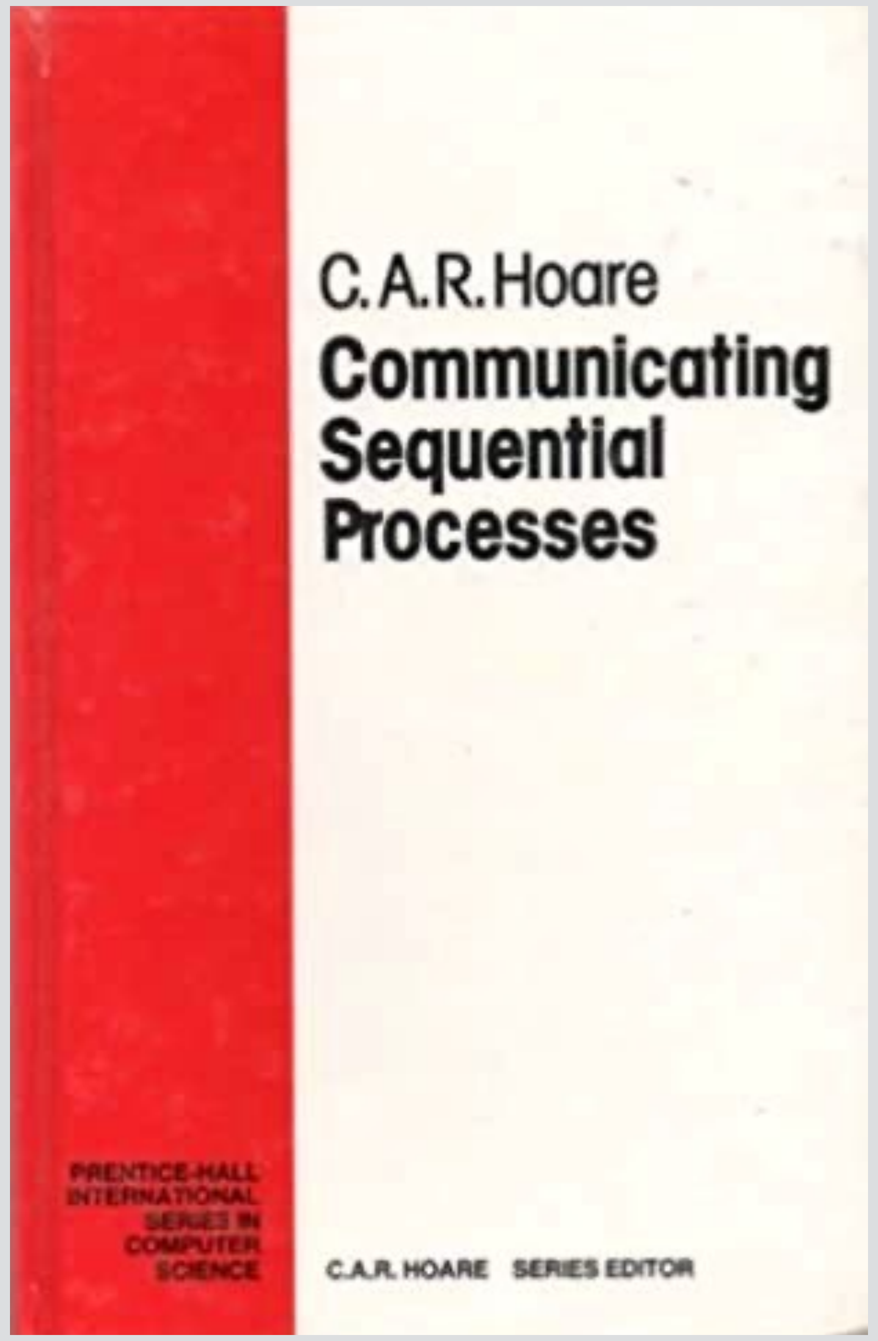
upvote (Carol, post1)

reward (Alice, 10)

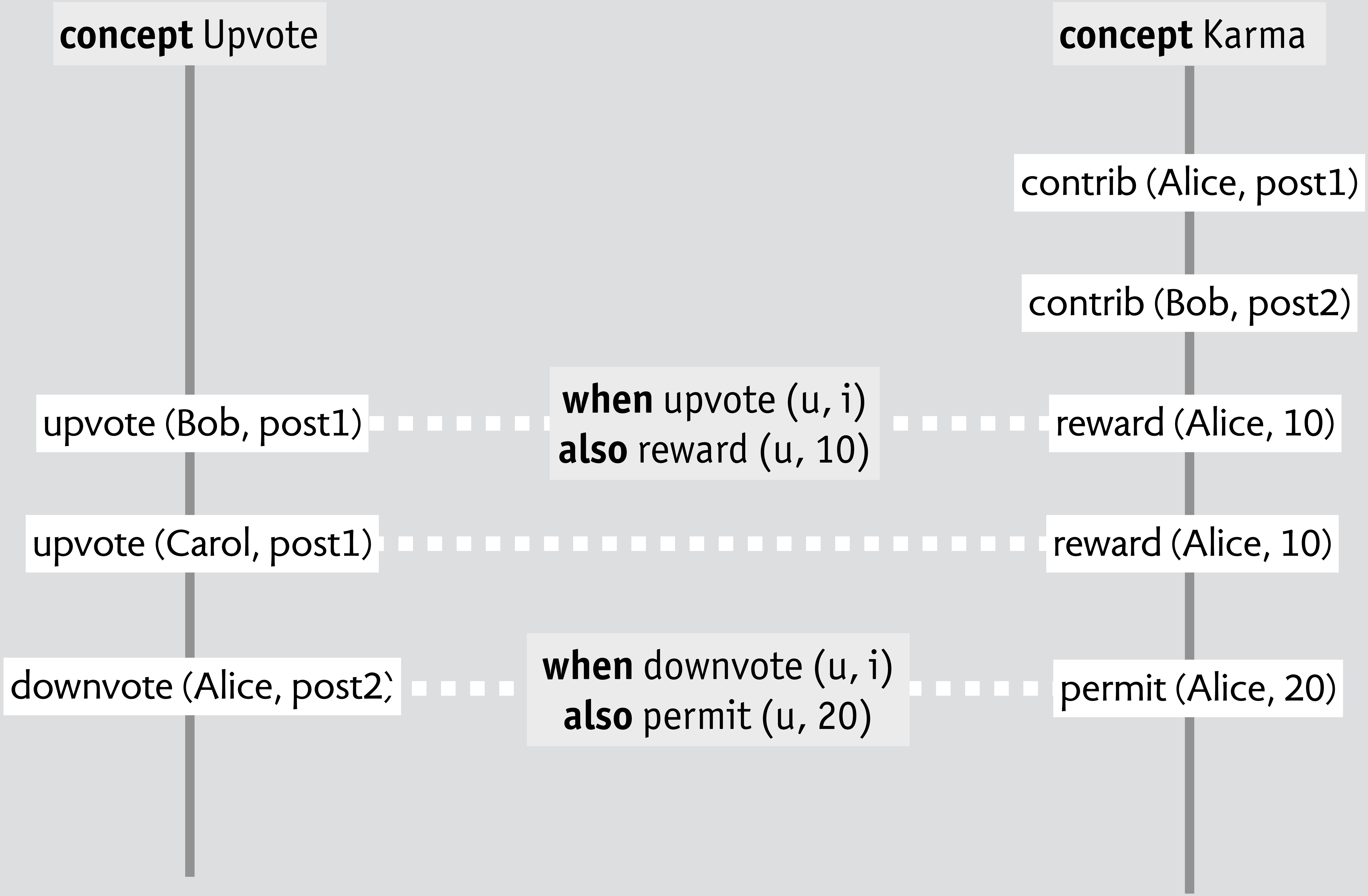
downvote (Alice, post2)

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

permit (Alice, 20)


C.A.R.Hoare
Communicating Sequential Processes
PRENTICE-HALL INTERNATIONAL SERIES IN COMPUTER SCIENCE
C.A.R. HOARE SERIES EDITOR

composition uses event sync from Hoare's CSP



concept Upvote

concept Karma

contrib (Alice, post1)

contrib (Bob, post2)

upvote (Bob, post1)

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

reward (Alice, 10)

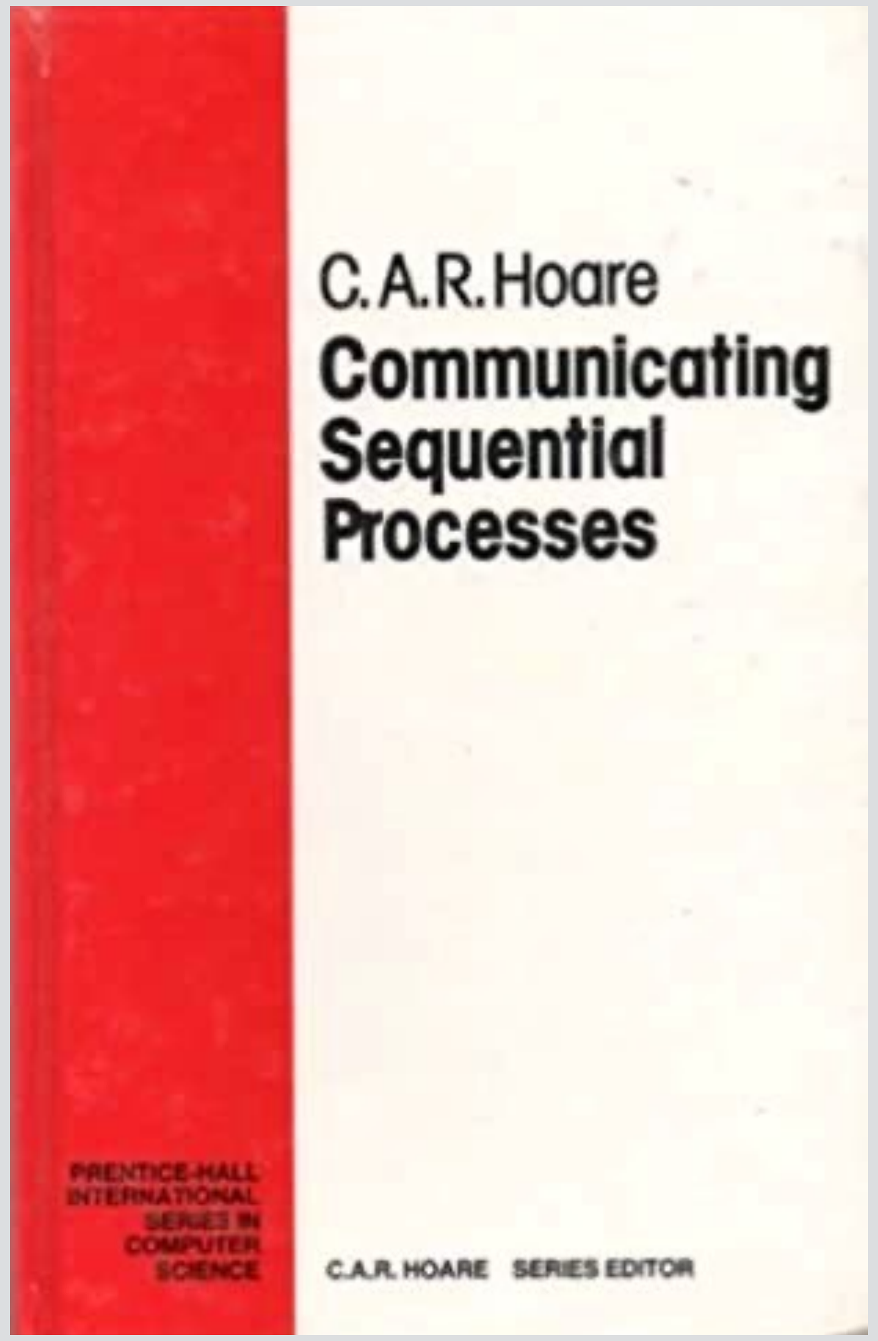
upvote (Carol, post1)

reward (Alice, 10)

downvote (Alice, post2)

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

permit (Alice, 20)

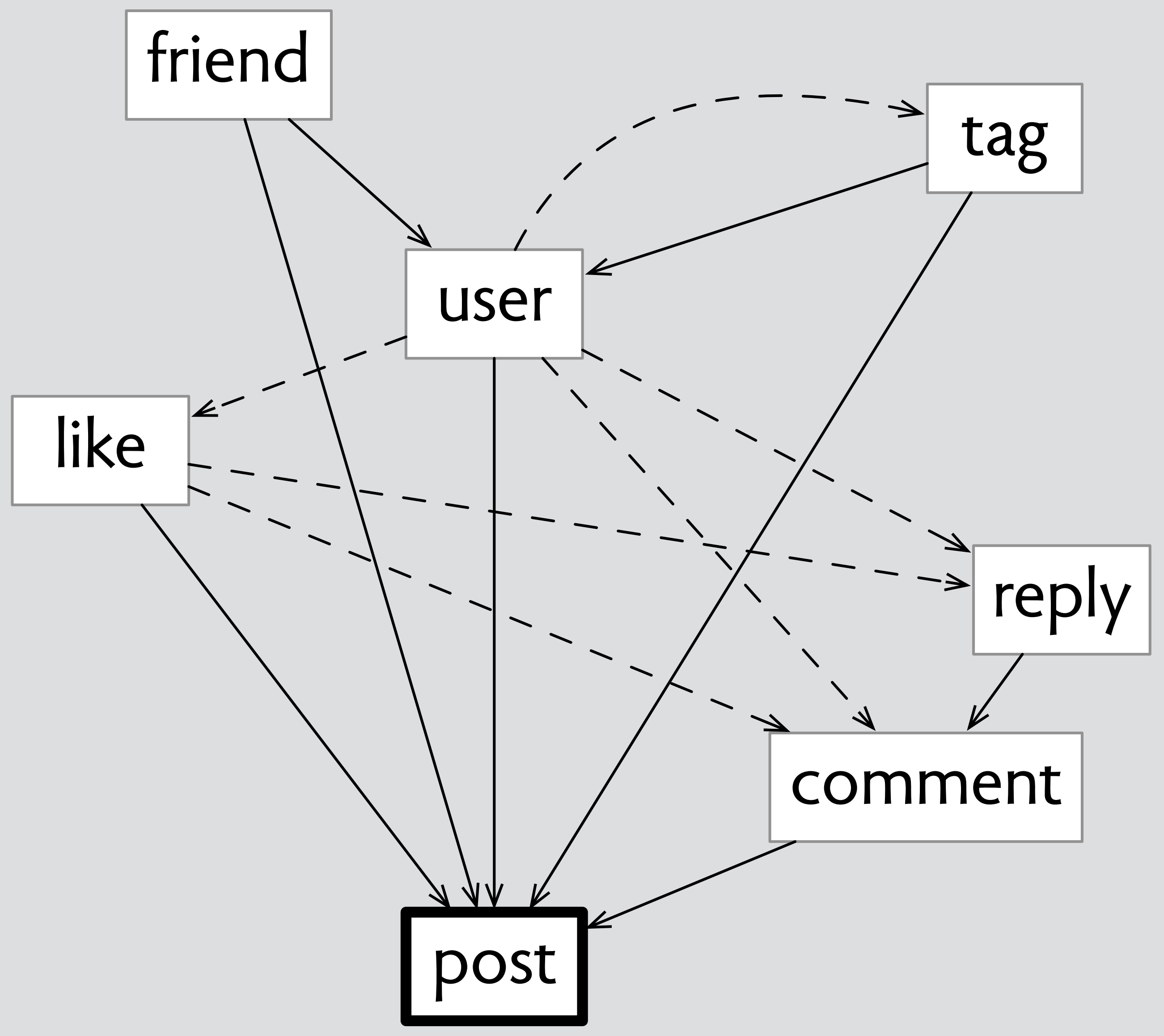

C.A.R.Hoare
Communicating Sequential Processes
PRENTICE-HALL INTERNATIONAL SERIES IN COMPUTER SCIENCE
C.A.R. HOARE SERIES EDITOR

composition uses
event sync from
Hoare's CSP

no concept coupling
concepts preserve properties

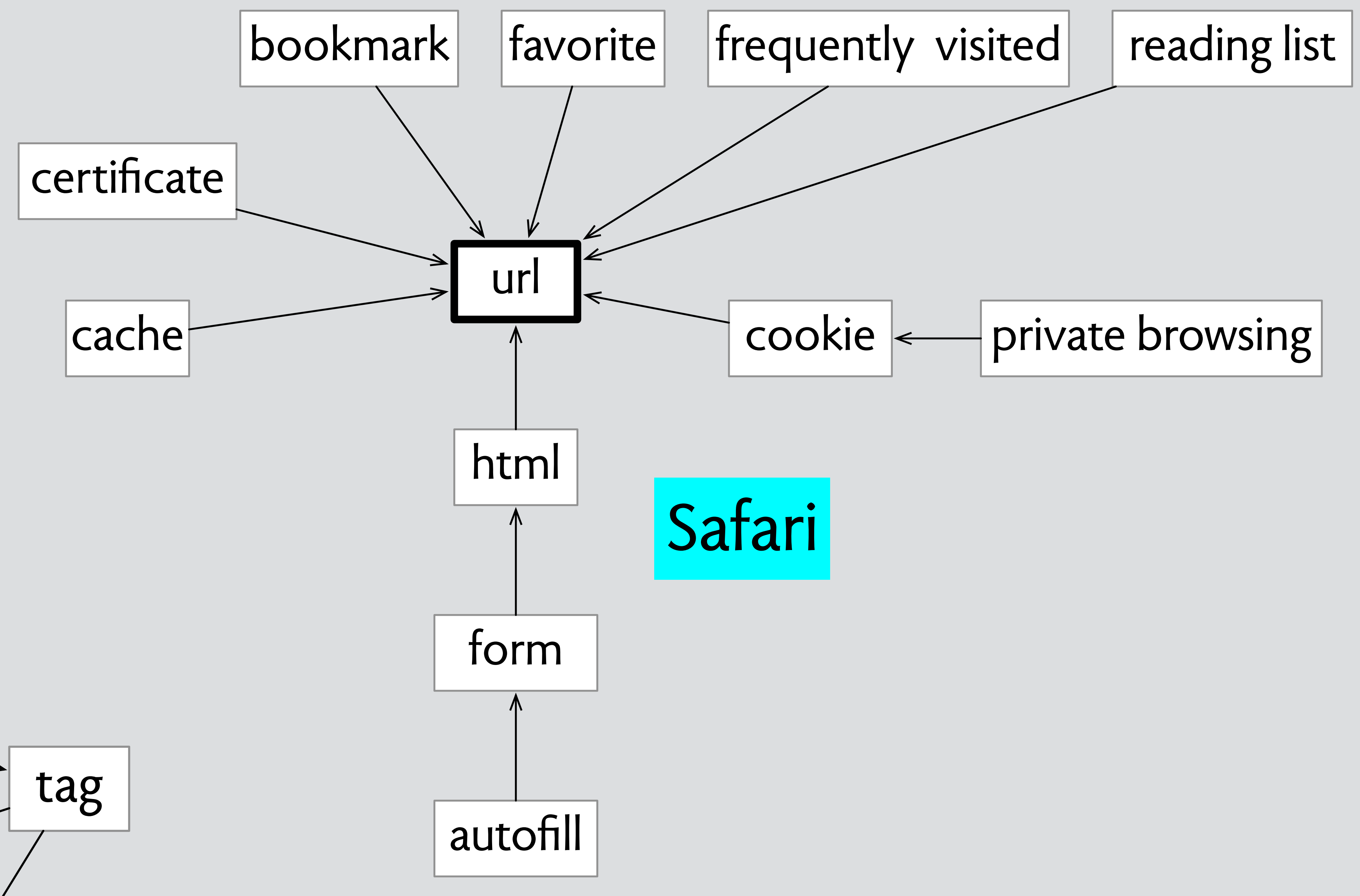
so what can
you do with
concepts?

characterize apps and families

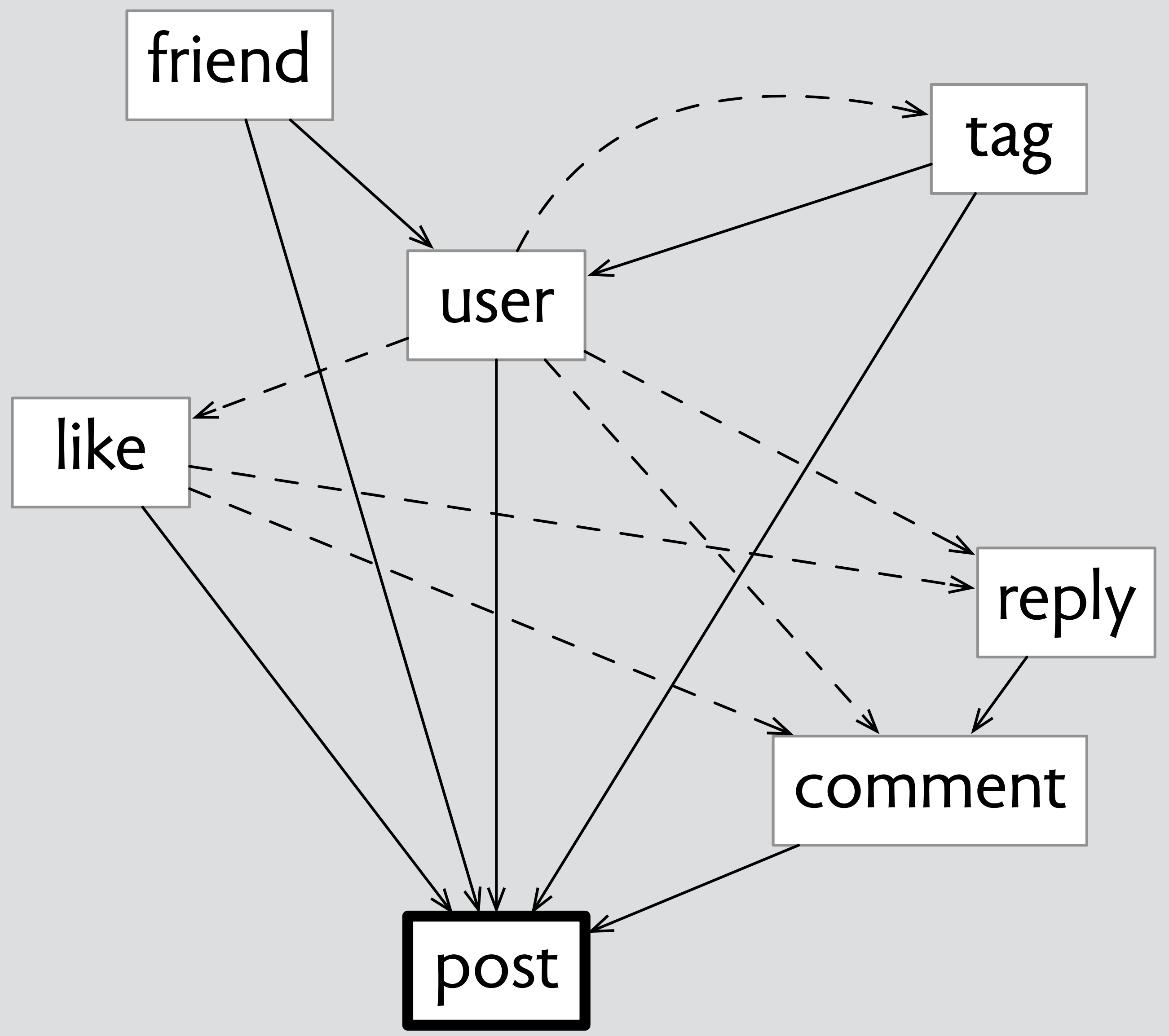


Facebook

characterize apps and families

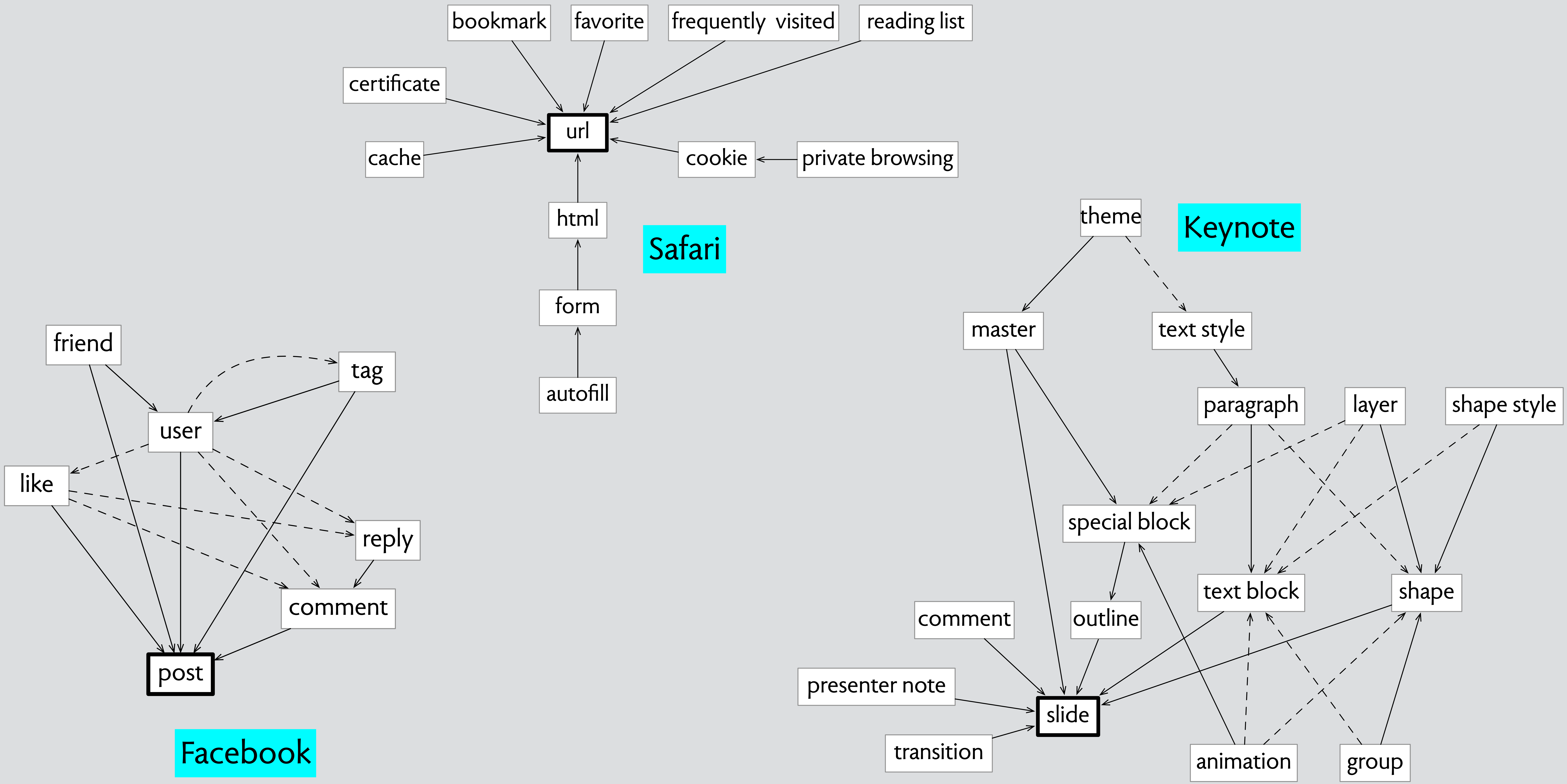


Safari

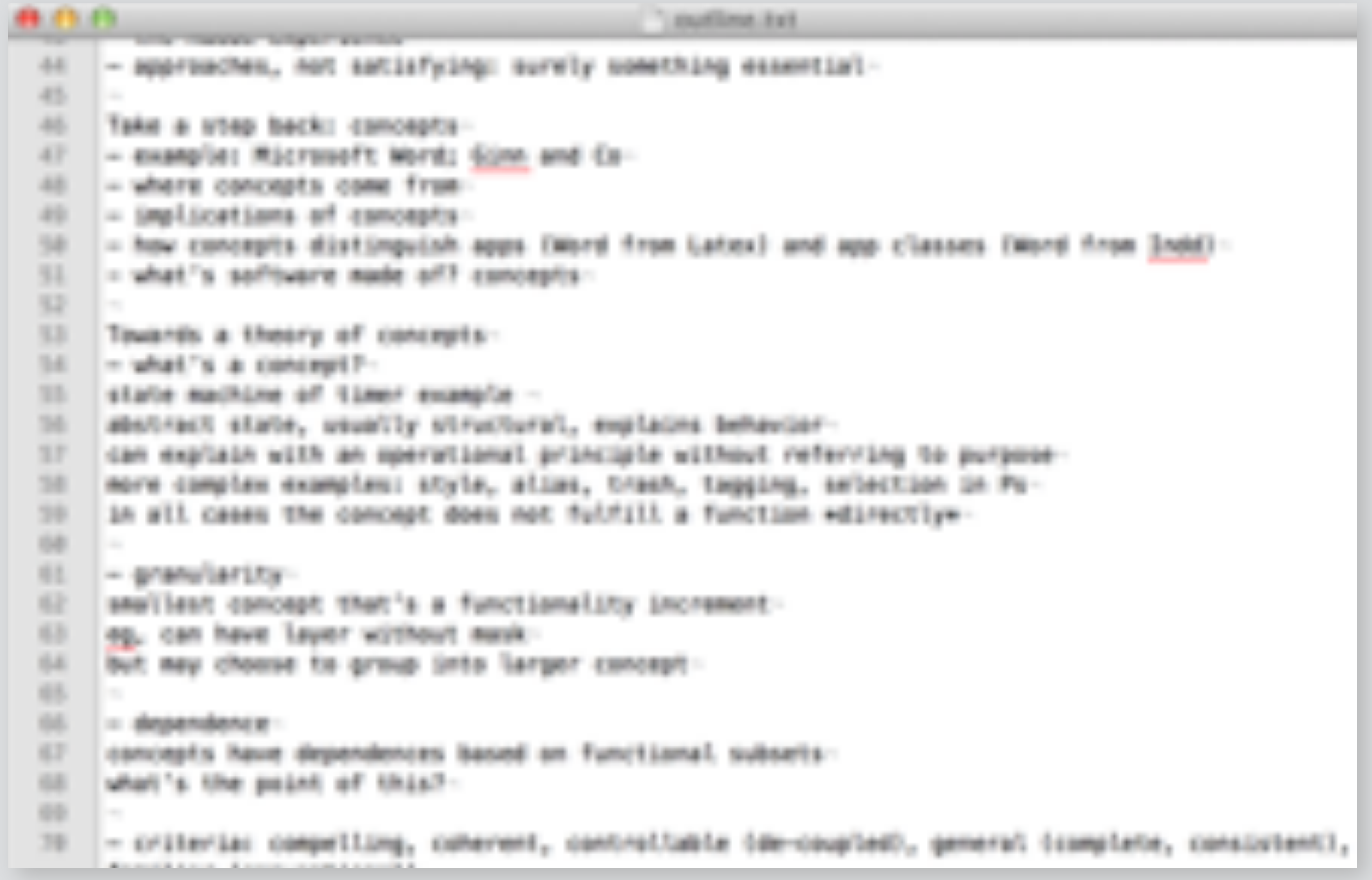


Facebook

characterize apps and families

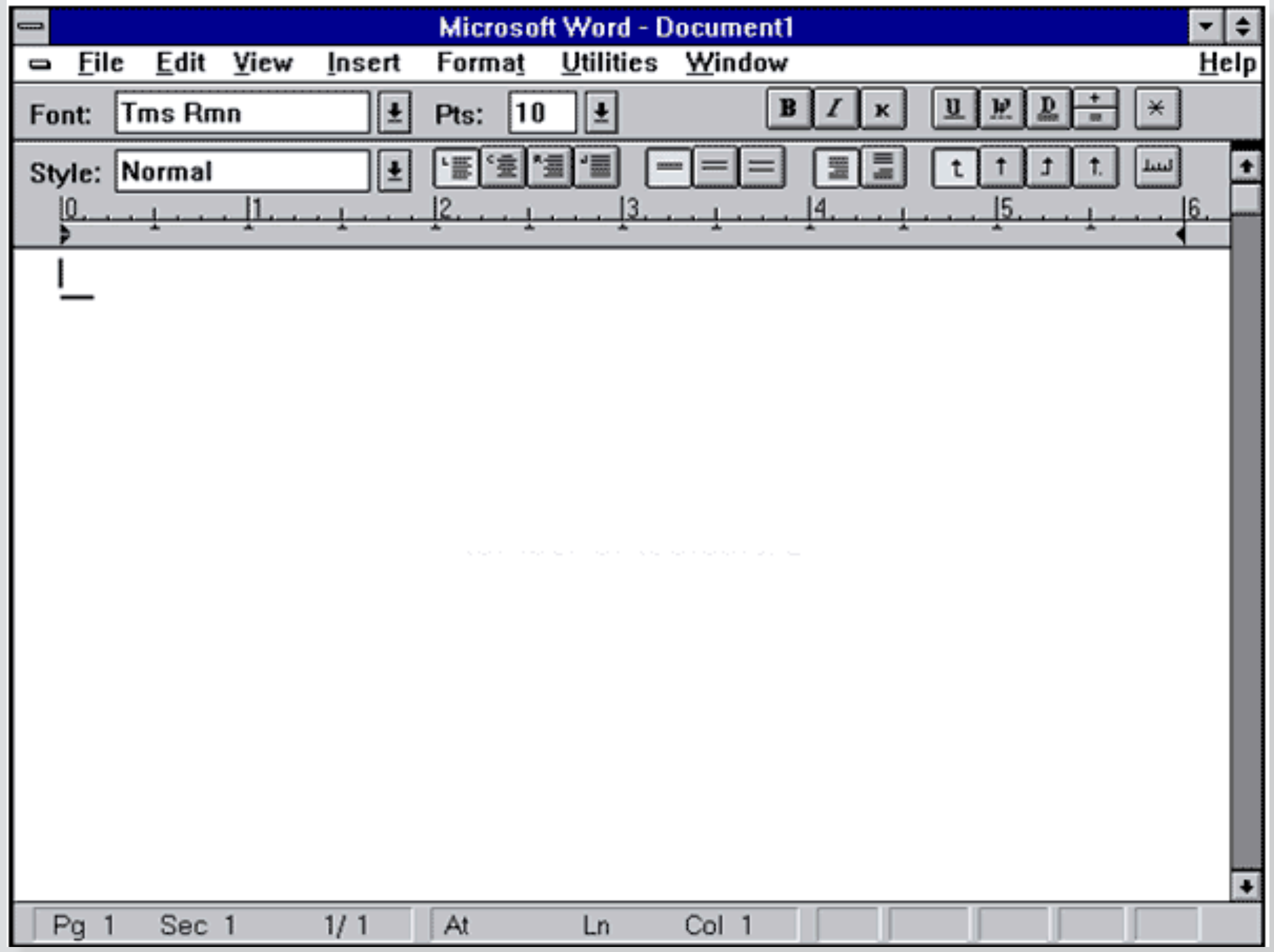


characterize apps & families

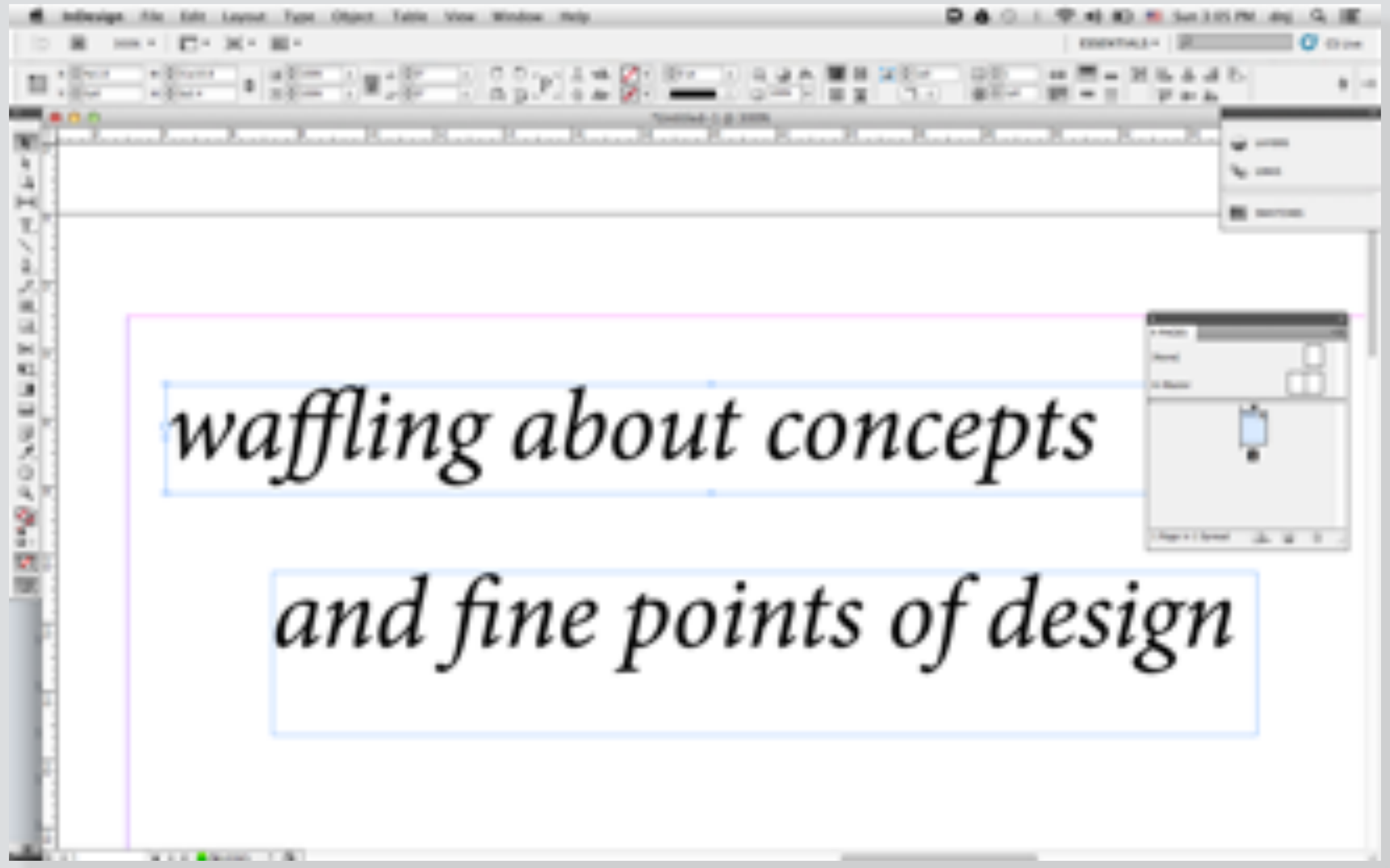


```
- approaches, not satisfying: surely something essential

Take a step back: concepts
- example: Microsoft Word: Ginn and Co
- where concepts come from
- implications of concepts
- how concepts distinguish apps (Word from Latex) and app classes (Word from Indd)
- what's software made of? concepts

Towards a theory of concepts
- what's a concept?
state machine of timer example
abstract state, usually structural, explains behavior
can explain with an operational principle without referring to purpose
more complex examples: style, alias, trash, tagging, selection in Ps
in all cases the concept does not fulfill a function *directly*

- granularity
smallest concept that's a functionality increment
eg, can have layer without mask
but may choose to group into larger concept

- dependence
concepts have dependences based on functional subsets
what's the point of this?

- criteria: compelling, coherent, controllable (de-coupled), general (complete, consistent),
```

text editor



word processor

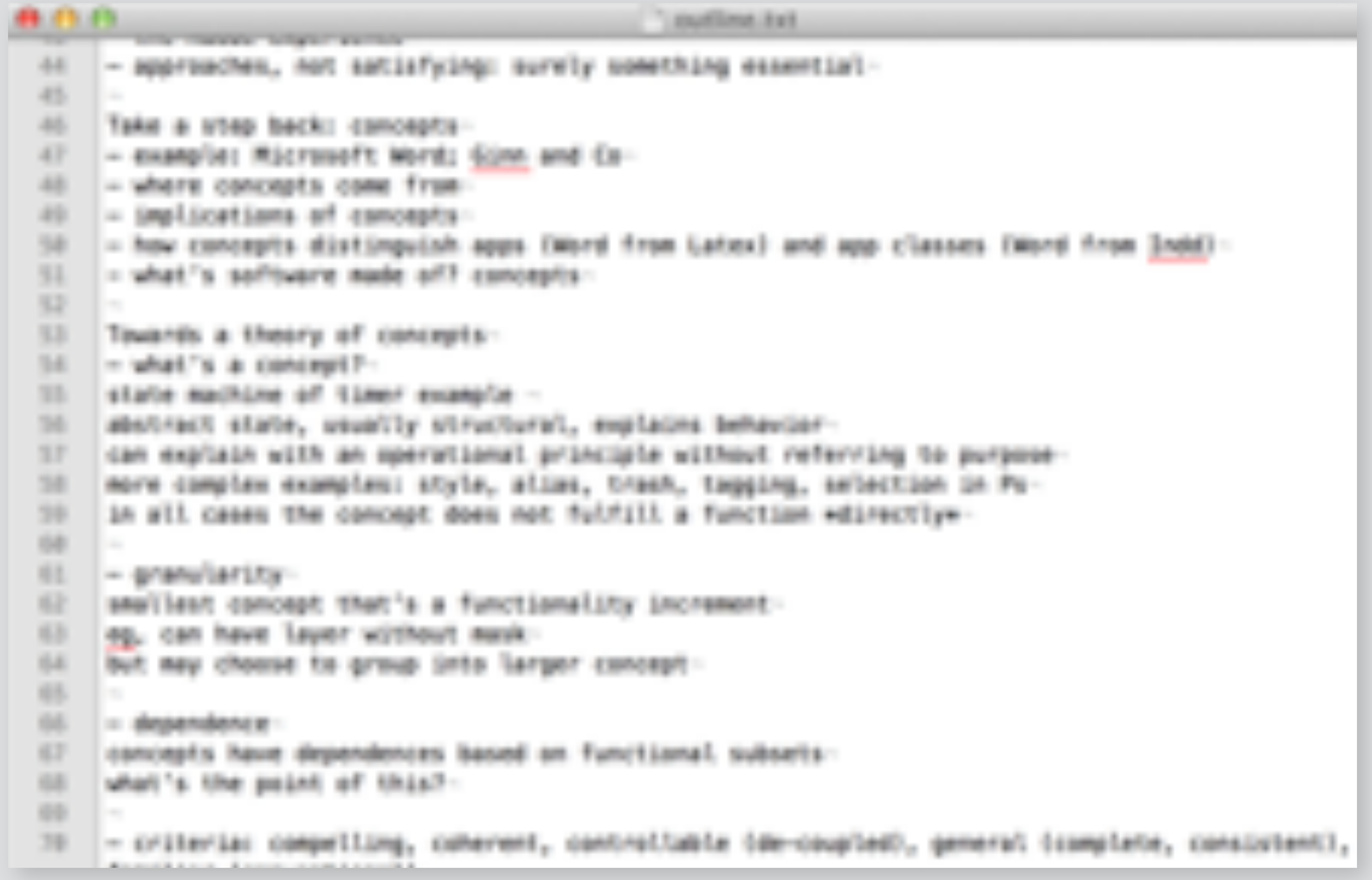


waffling about concepts

and fine points of design

desktop publishing app

characterize apps & families



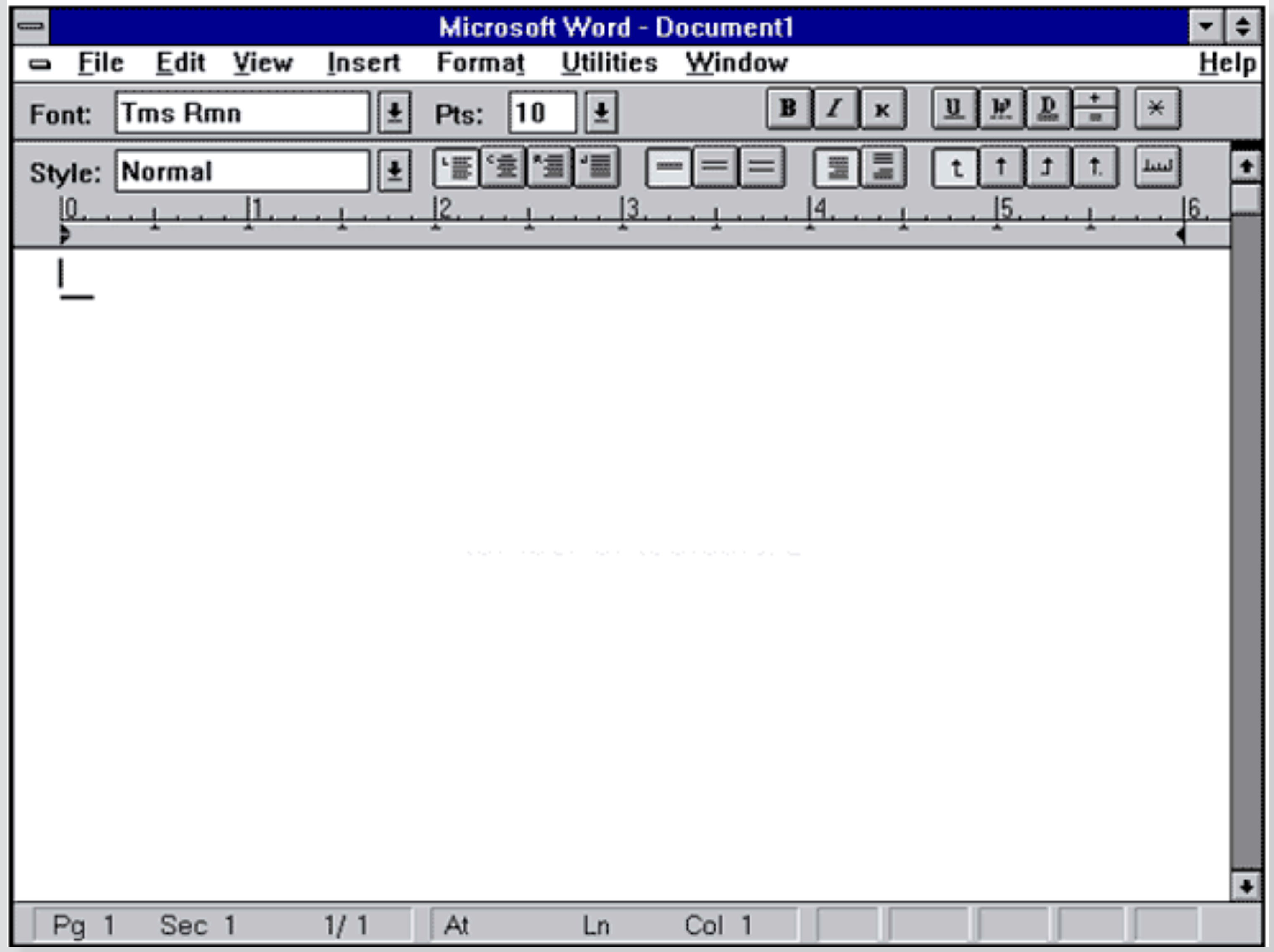
```
- approaches, not satisfying: surely something essential

Take a step back: concepts
- example: Microsoft Word: Ginn and Co
- where concepts come from
- implications of concepts
- how concepts distinguish apps (Word from Latex) and app classes (Word from Indd)
- what's software made of? concepts

Towards a theory of concepts
- what's a concept?
state machine of timer example
abstract state, usually structural, explains behavior
can explain with an operational principle without referring to purpose
more complex examples: style, alias, trash, tagging, selection in Ps
in all cases the concept does not fulfill a function *directly*

- granularity
smallest concept that's a functionality increment
eg, can have layer without mask
but may choose to group into larger concept

- dependence
concepts have dependences based on functional subsets
what's the point of this?

- criteria: compelling, coherent, controllable (de-coupled), general (complete, consistent),
```

text editor

line

character set

markup

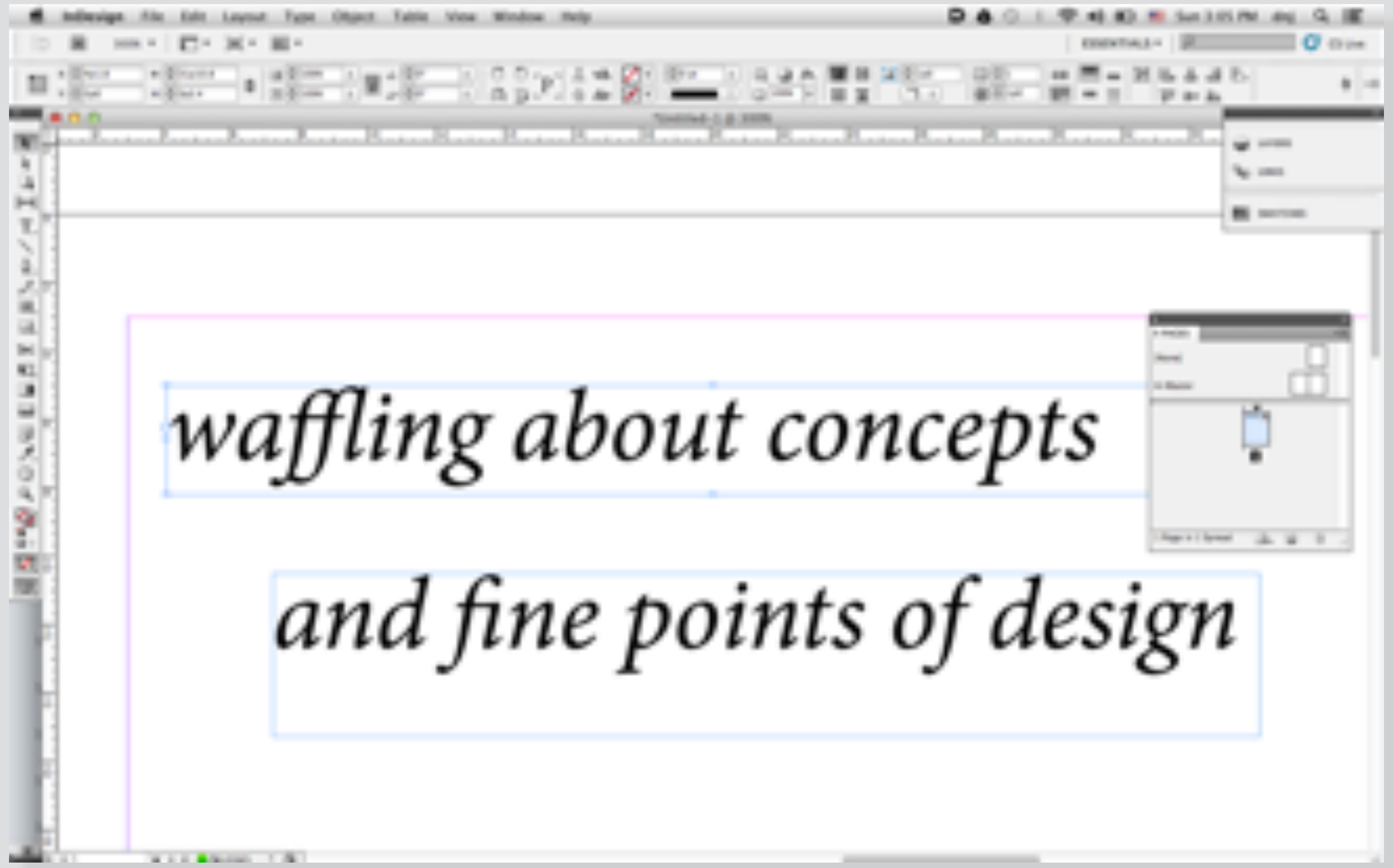


Font: Tms Rmn

Pts: 10

Style: Normal

word processor

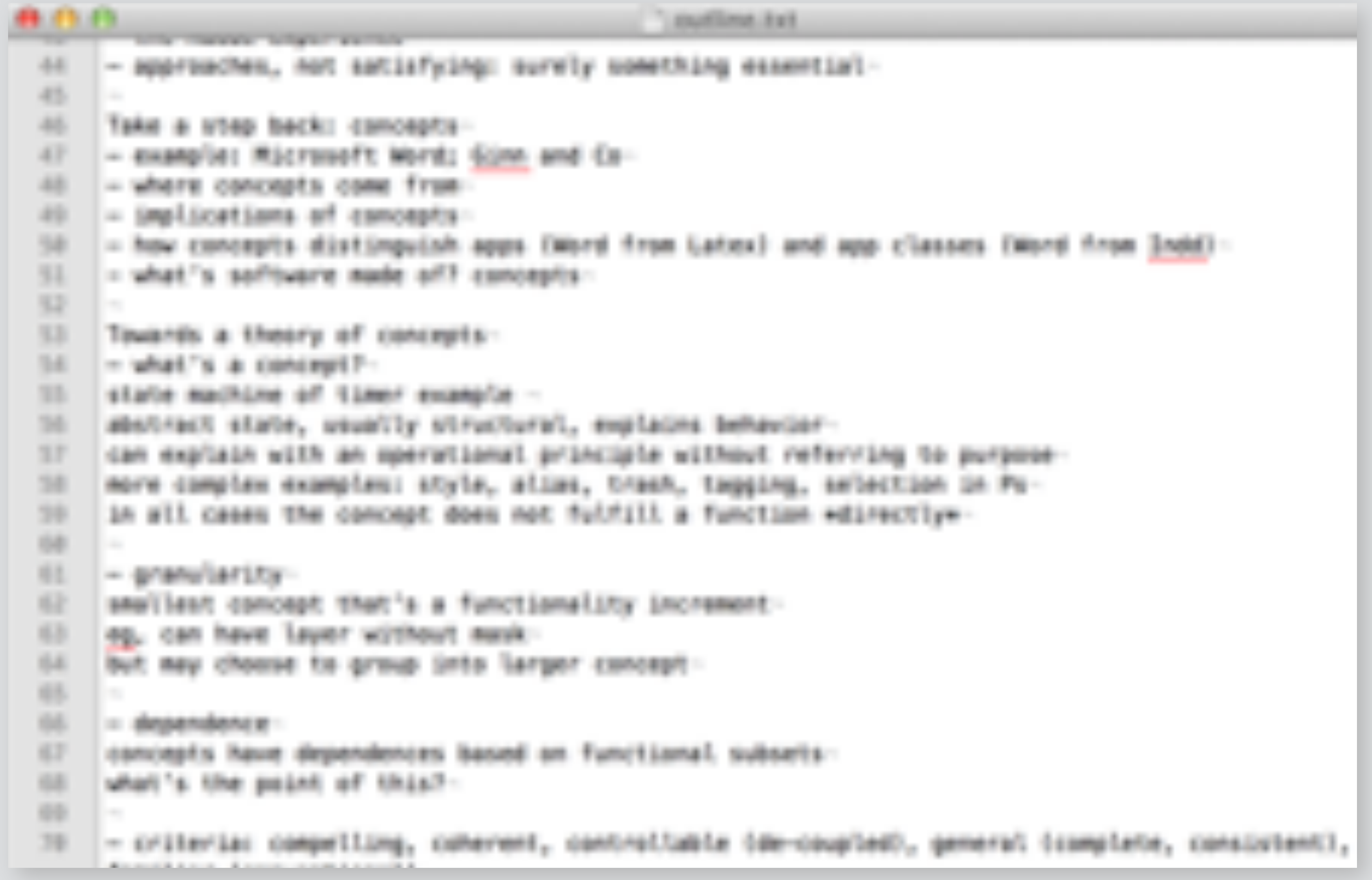


waffling about concepts

and fine points of design

desktop publishing app

characterize apps & families



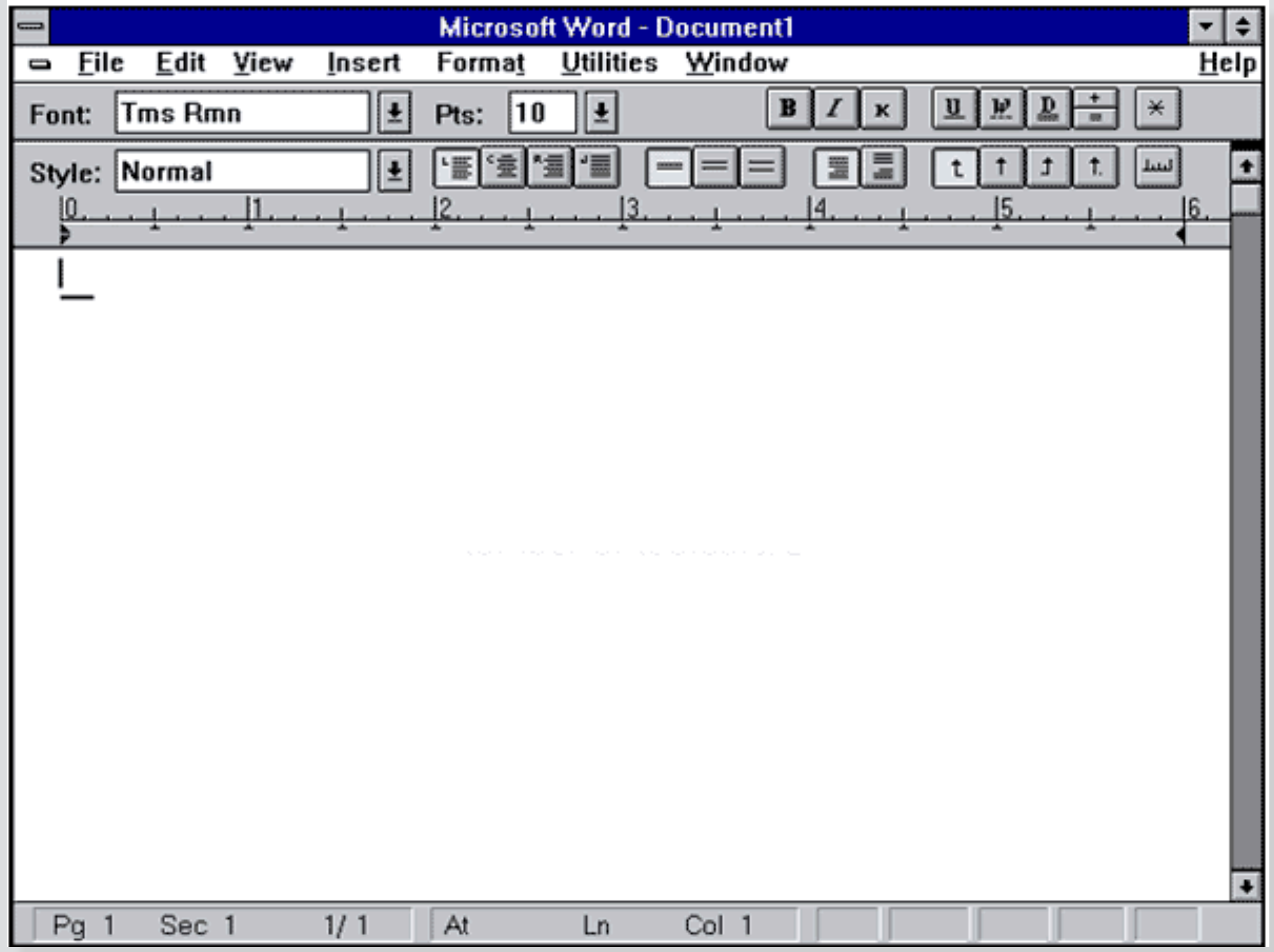
```
- approaches, not satisfying: surely something essential

Take a step back: concepts
- example: Microsoft Word: Sinn and Co
- where concepts come from
- implications of concepts
- how concepts distinguish apps (Word from Latex) and app classes (Word from Indd)
- what's software made of? concepts

Towards a theory of concepts
- what's a concept?
state machine of timer example
abstract state, usually structural, explains behavior
can explain with an operational principle without referring to purpose
more complex examples: style, alias, trash, tagging, selection in Ps
in all cases the concept does not fulfill a function *directly*

- granularity
smallest concept that's a functionality increment
eg, can have layer without mask
but may choose to group into larger concept

- dependence
concepts have dependences based on functional subsets
what's the point of this?

- criteria: compelling, coherent, controllable (de-coupled), general (complete, consistent),
```

text editor

line

character set

markup



Font: Tms Rmn Pts: 10

Style: Normal

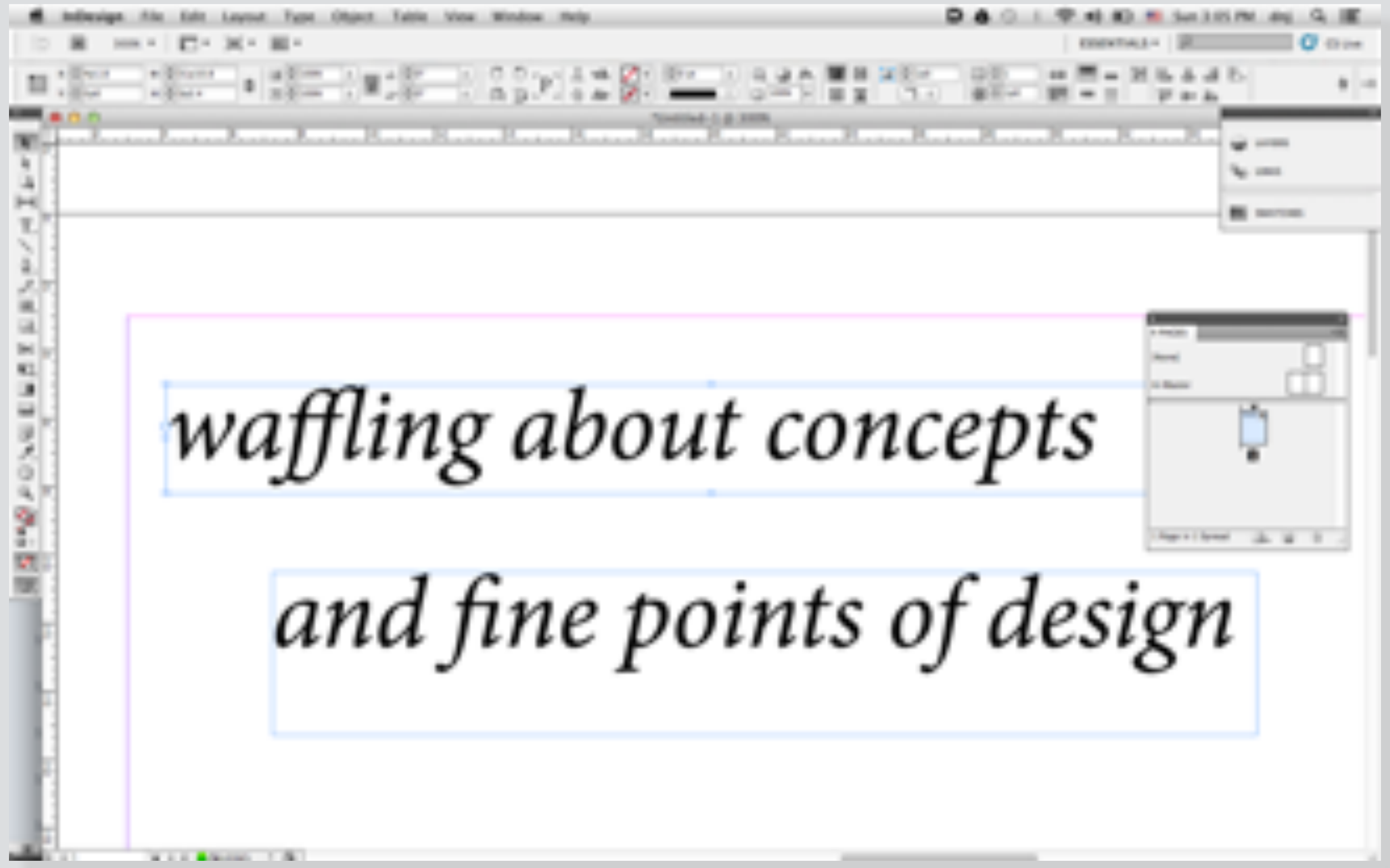
Pg 1 Sec 1 1/ 1 At Ln Col 1

word processor

paragraph

format

style

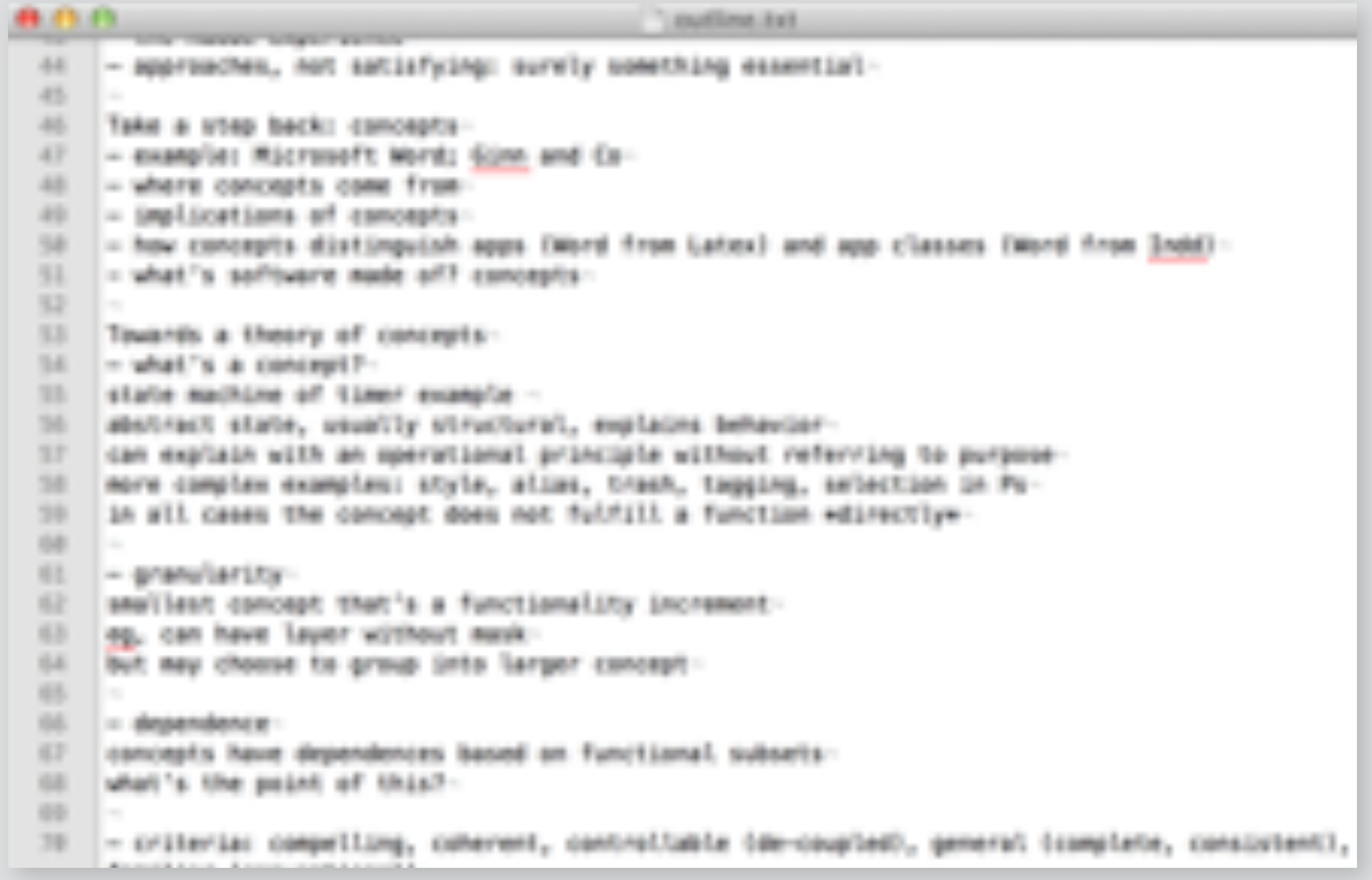


waffling about concepts

and fine points of design

desktop publishing app

characterize apps & families



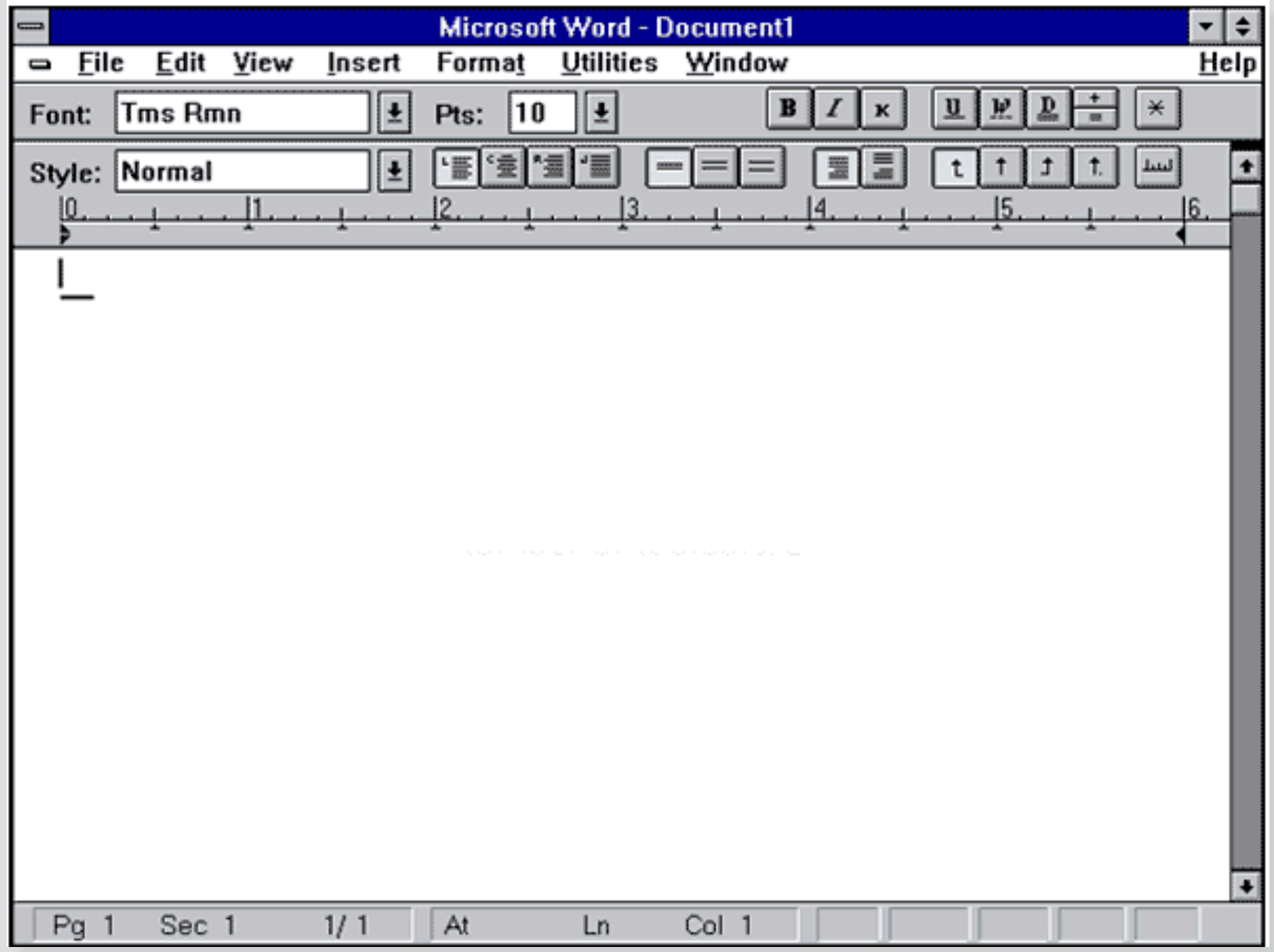
```
- approaches, not satisfying: surely something essential

Take a step back: concepts
- example: Microsoft Word: Sinn and Co
- where concepts come from
- implications of concepts
- how concepts distinguish apps (Word from Latex) and app classes (Word from Indd)
- what's software made of? concepts

Towards a theory of concepts
- what's a concept?
state machine of timer example
abstract state, usually structural, explains behavior
can explain with an operational principle without referring to purpose
more complex examples: style, alias, trash, tagging, selection in Ps
in all cases the concept does not fulfill a function *directly*

- granularity
smallest concept that's a functionality increment
eg, can have layer without mask
but may choose to group into larger concept

- dependence
concepts have dependences based on functional subsets
what's the point of this?

- criteria: compelling, coherent, controllable (de-coupled), general (complete, consistent),
```

text editor

line

character set

markup

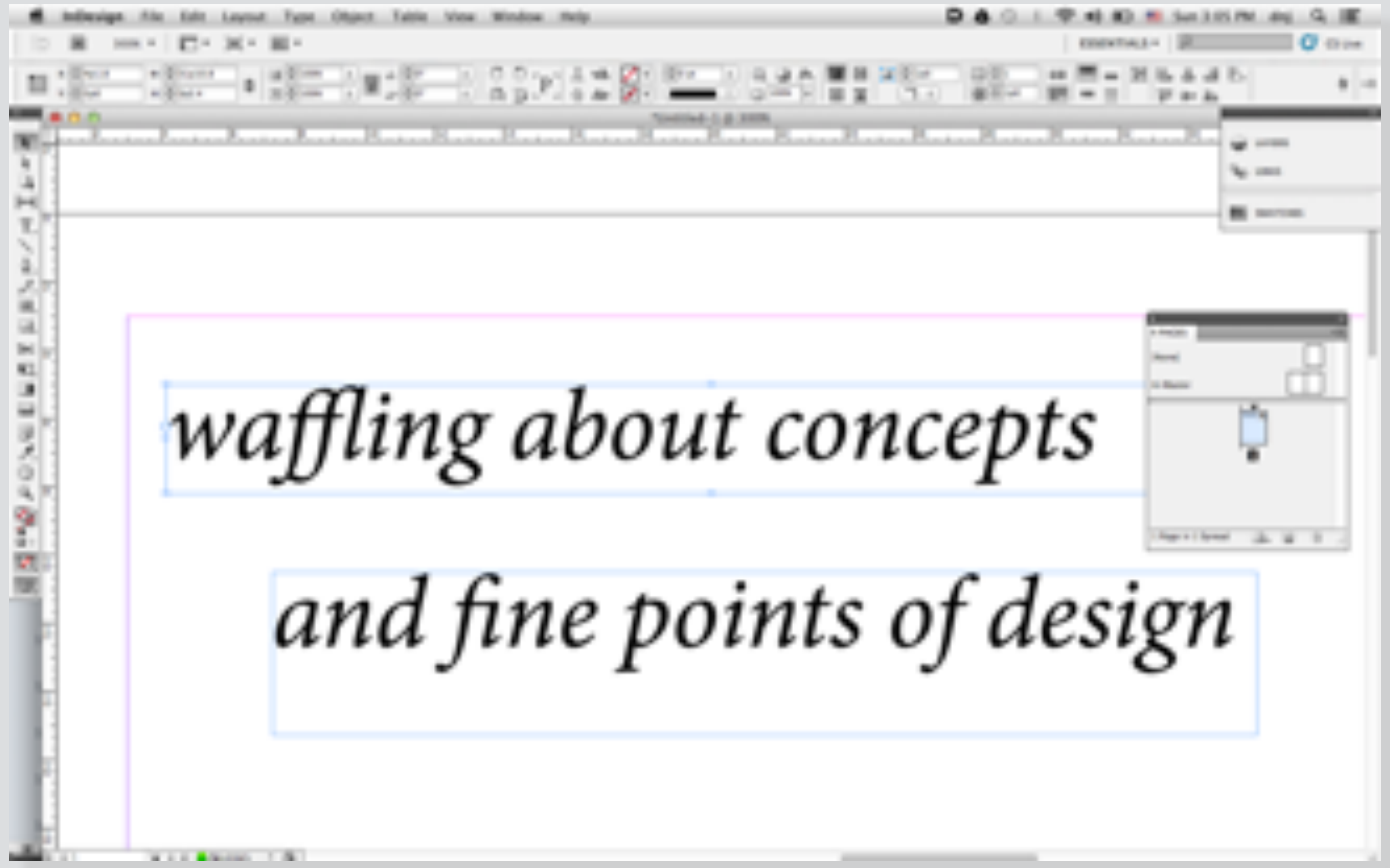


word processor

paragraph

format

style



waffling about concepts

and fine points of design

desktop publishing app

paragraph

format

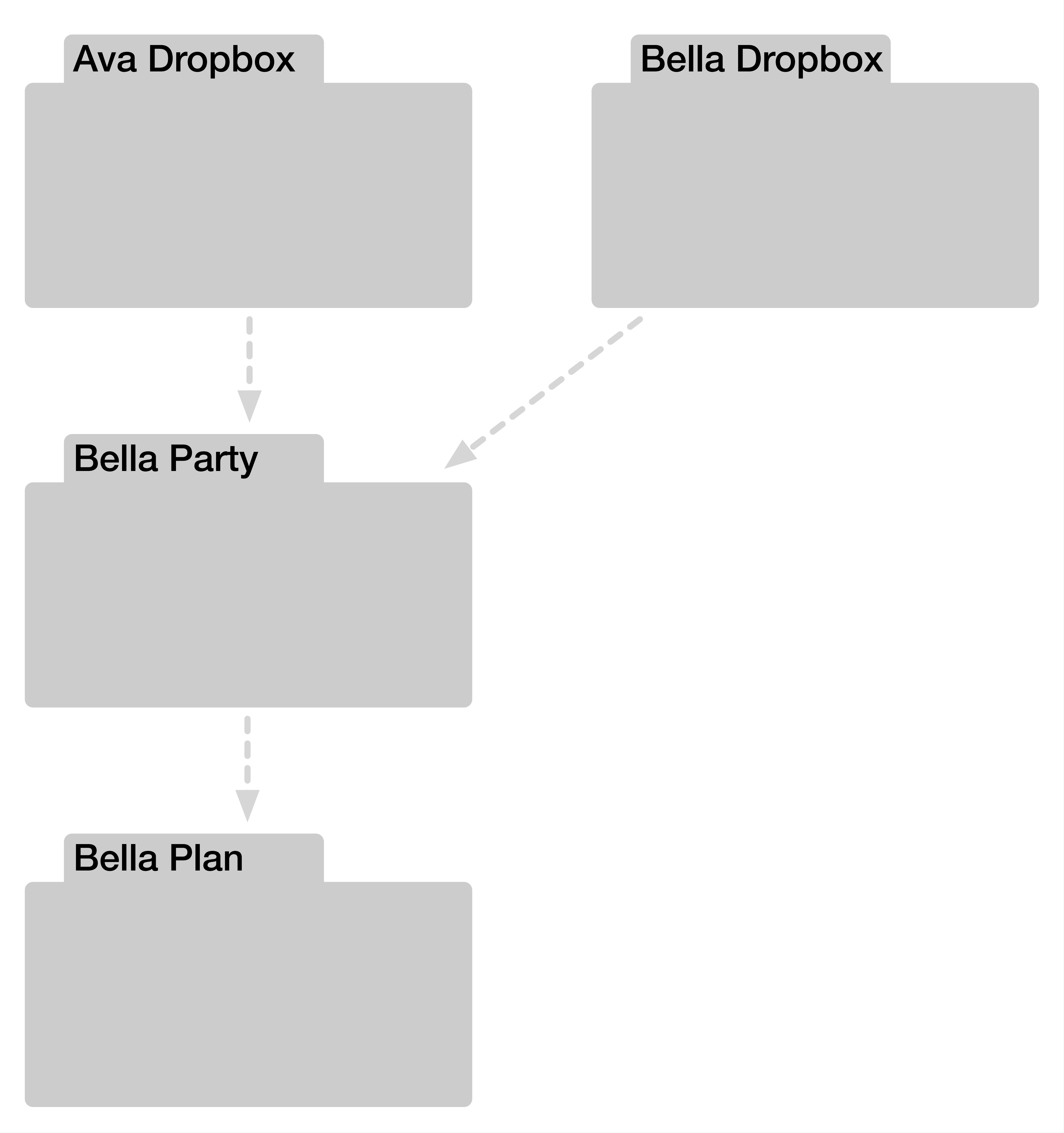
style

page

textflow

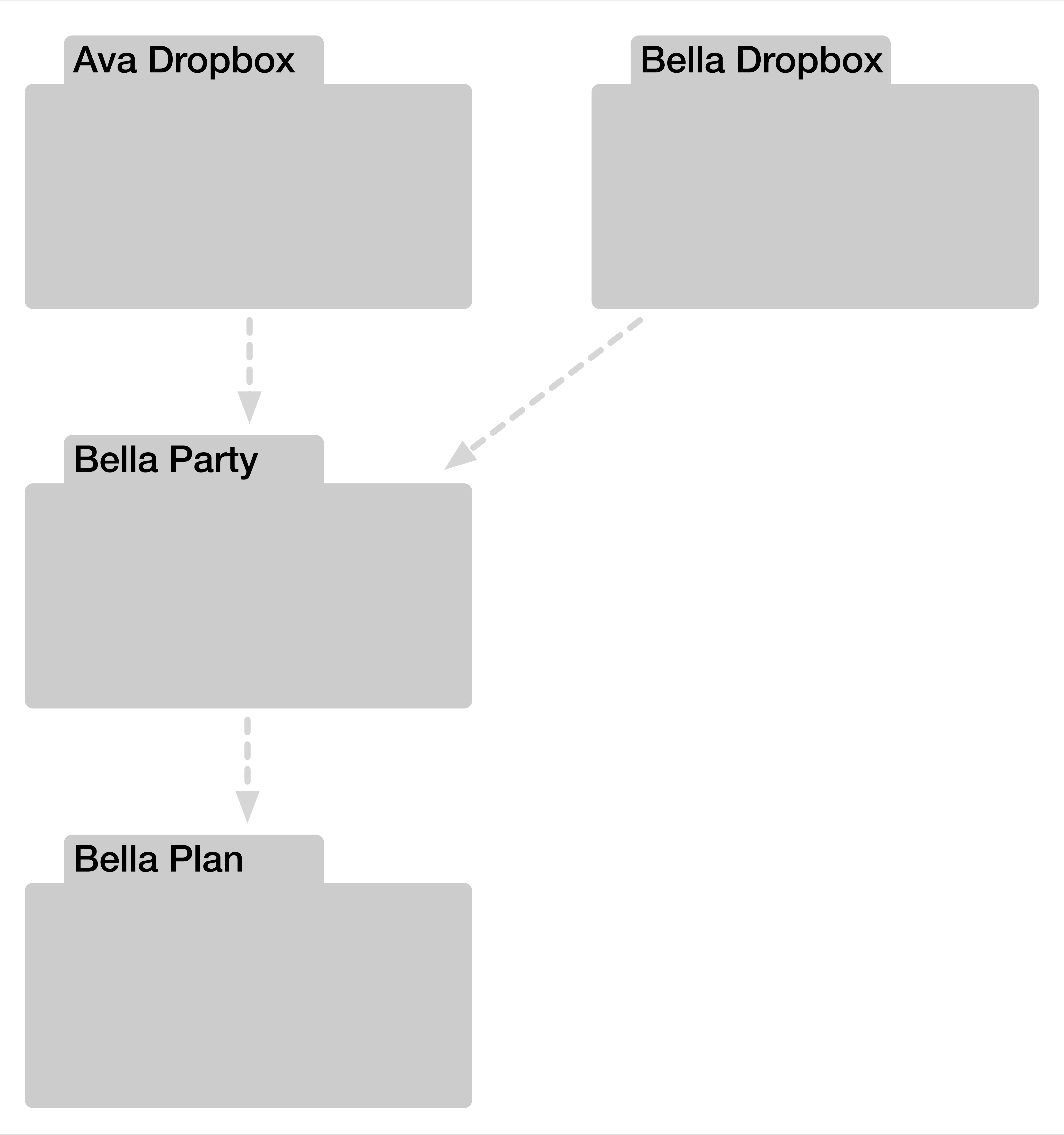
explore & evaluate individual concepts

explore & evaluate individual concepts

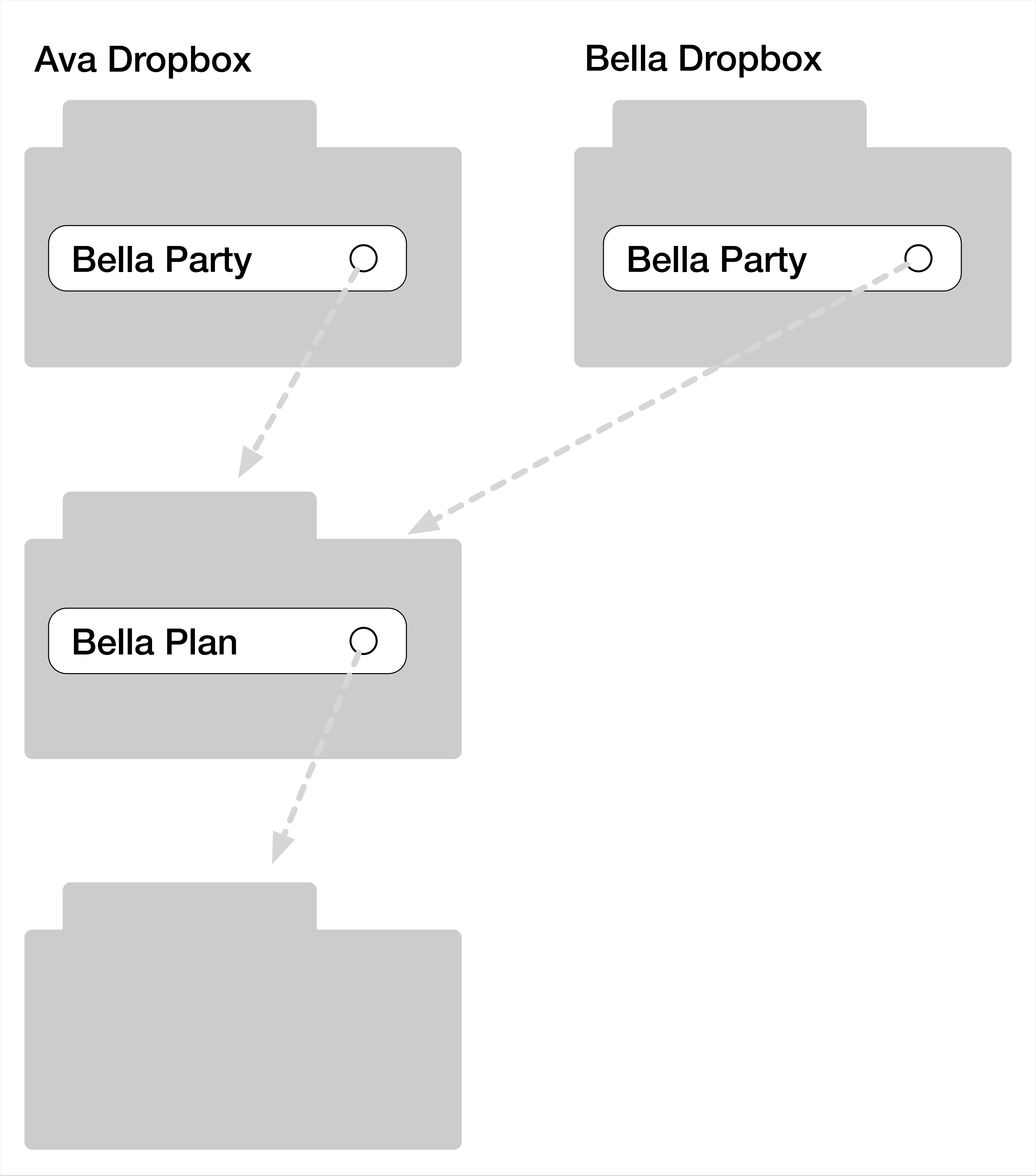


how many users believe the folder concept works

explore & evaluate individual concepts

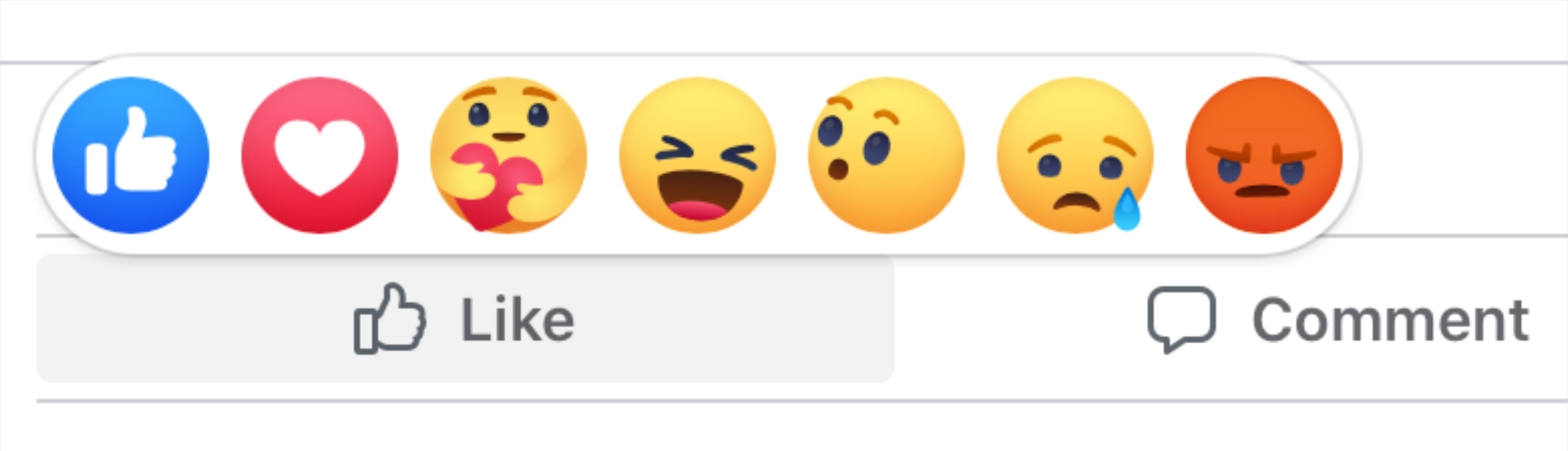


how many users believe the folder concept works

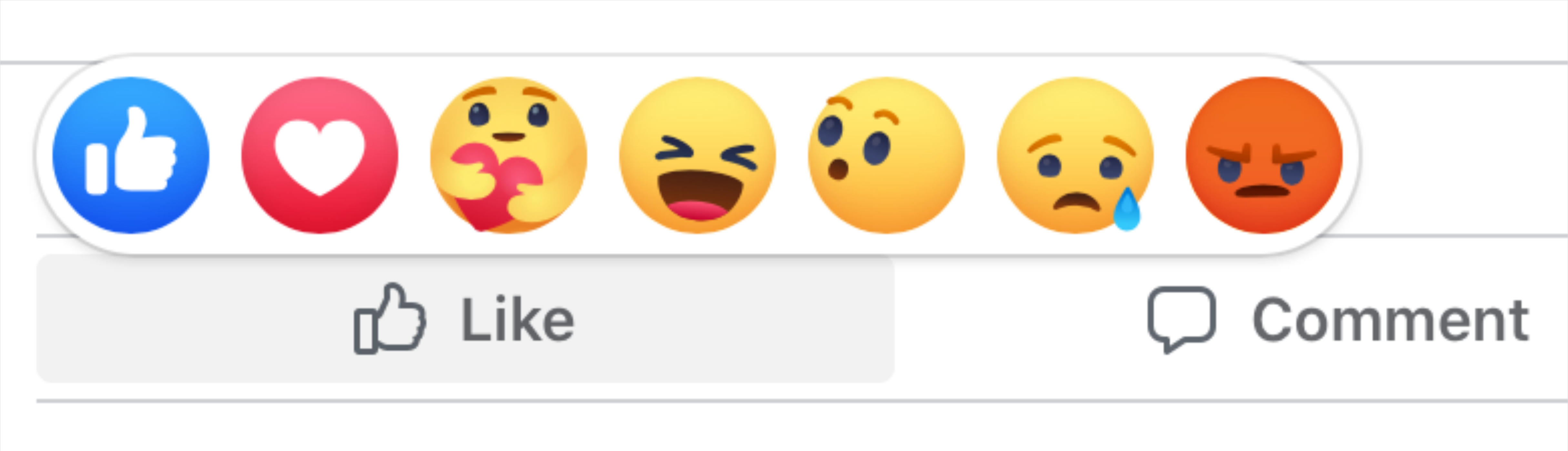


how folders actually work (in Dropbox, Unix, Multics)

analyze how concepts fit together



analyze how concepts fit together



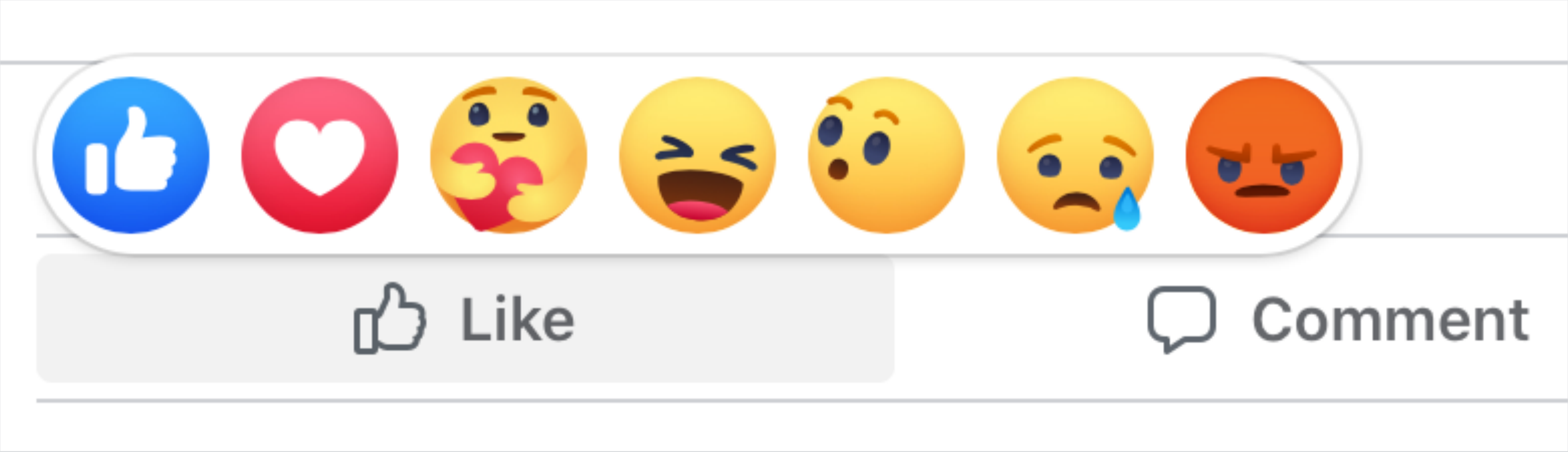
concept Upvote

purpose rank items by popularity

actions

```
  upvote (u: User, i: Item)
   ...
```

analyze how concepts fit together



concept Upvote

purpose rank items by popularity

actions

```
  upvote (u: User, i: Item)
   ...
```

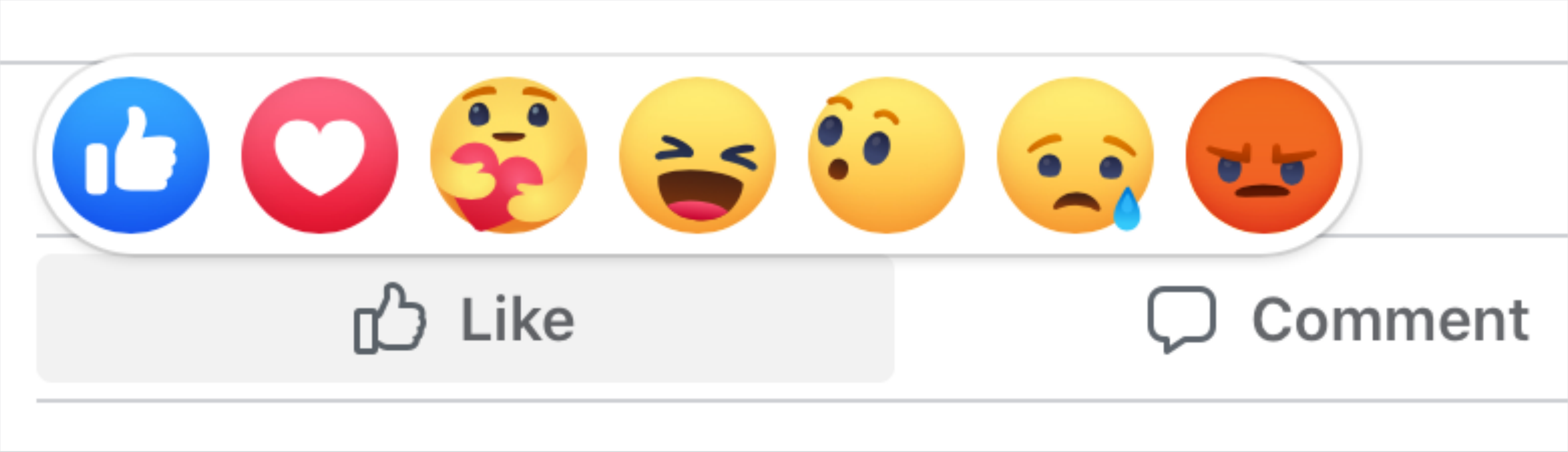
concept Reaction

purpose convey emotion to author

actions

```
  reactAngry (u: User, i: Item)
   ...
```

analyze how concepts fit together



concept Upvote

purpose rank items by popularity

actions

  upvote (u: User, i: Item)

  ...

concept Reaction

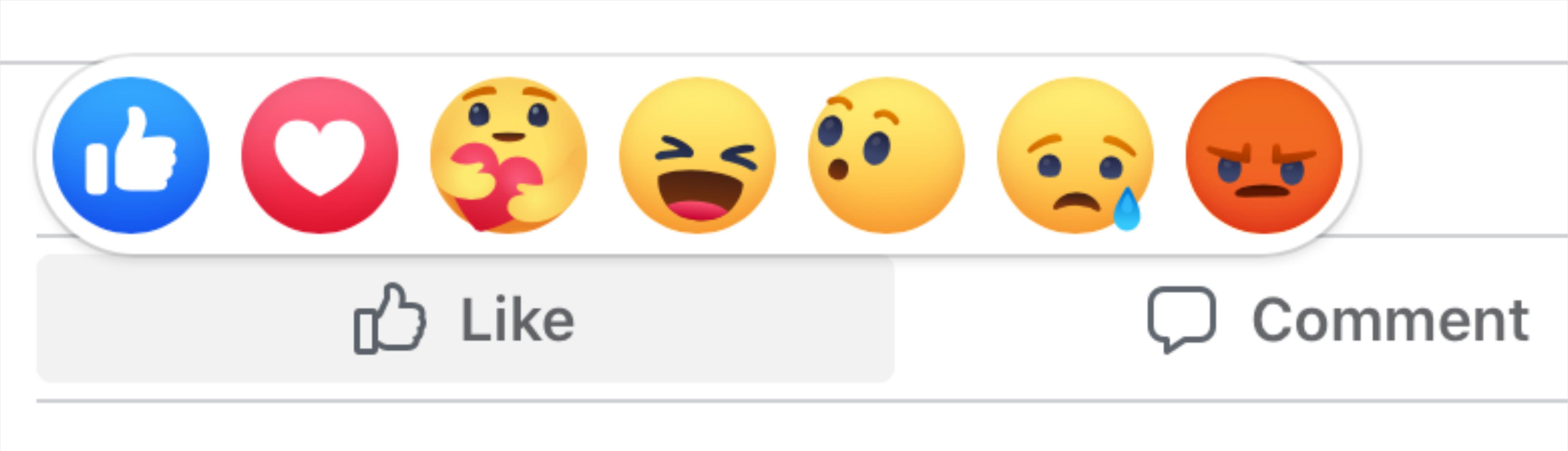
purpose convey emotion to author

actions

  reactAngry (u: User, i: Item)

  ...

analyze how concepts fit together



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)
    …

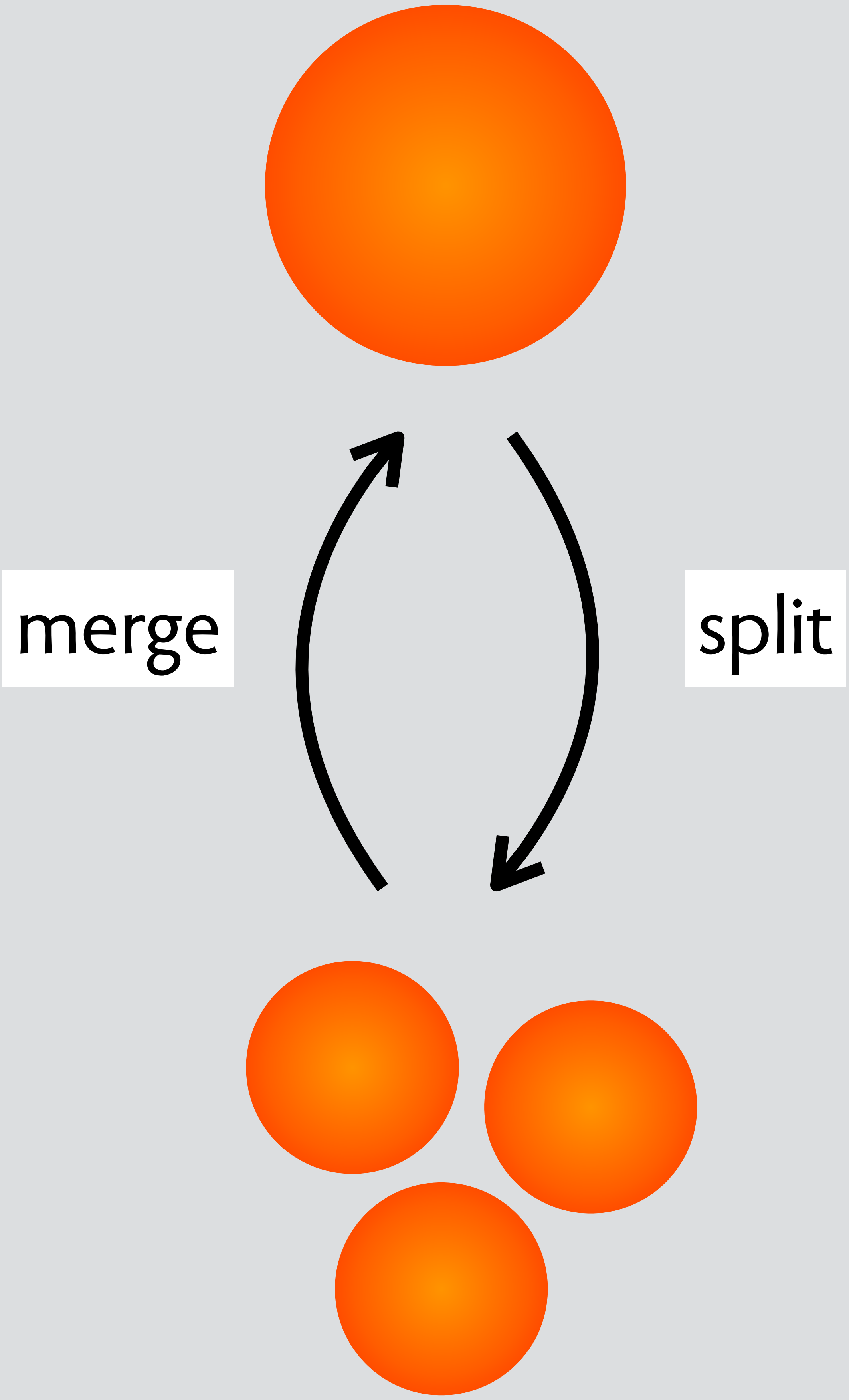
unwanted sync?

design moves

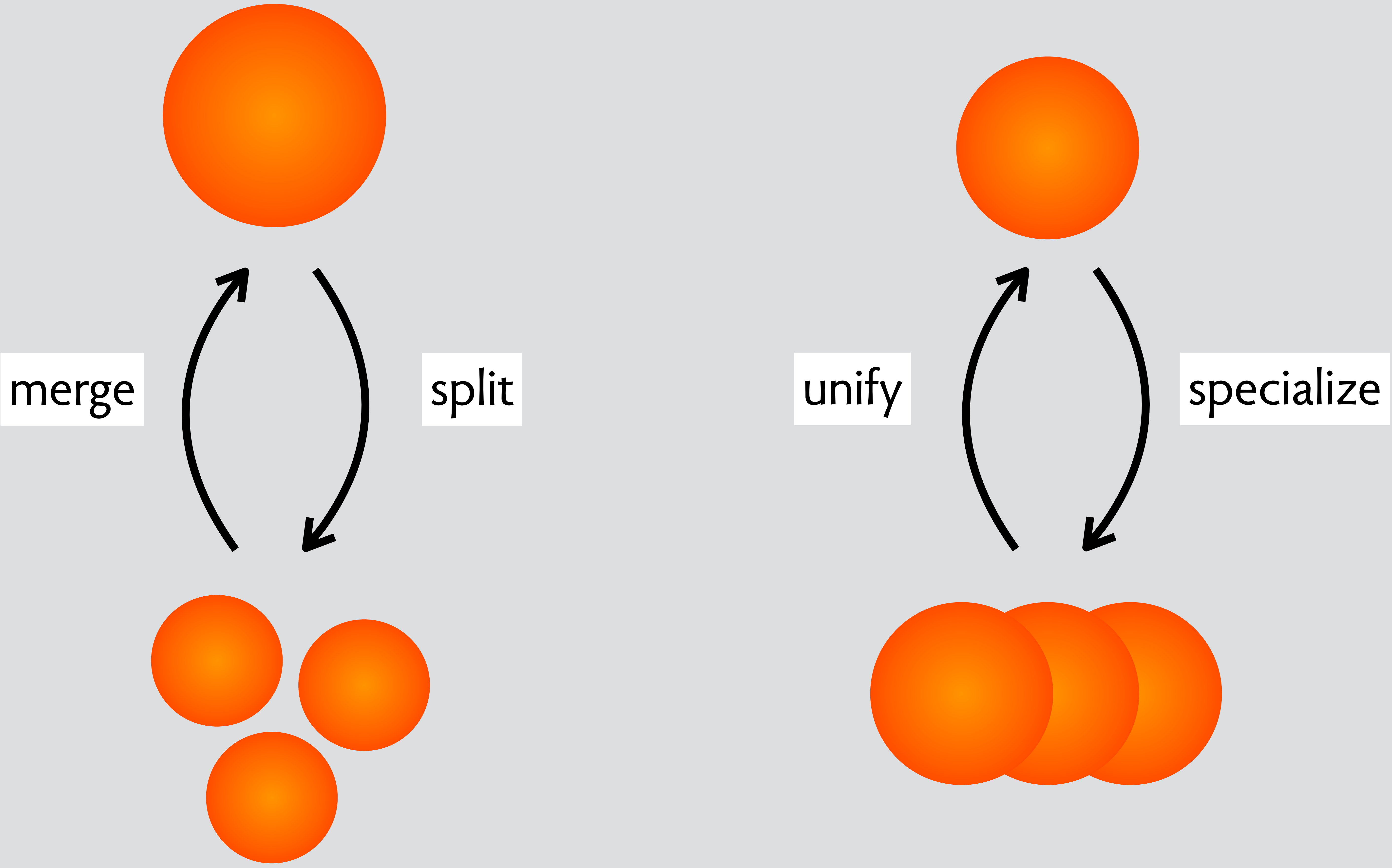
mechanical analogs

three pairs of design moves

three pairs of design moves



three pairs of design moves



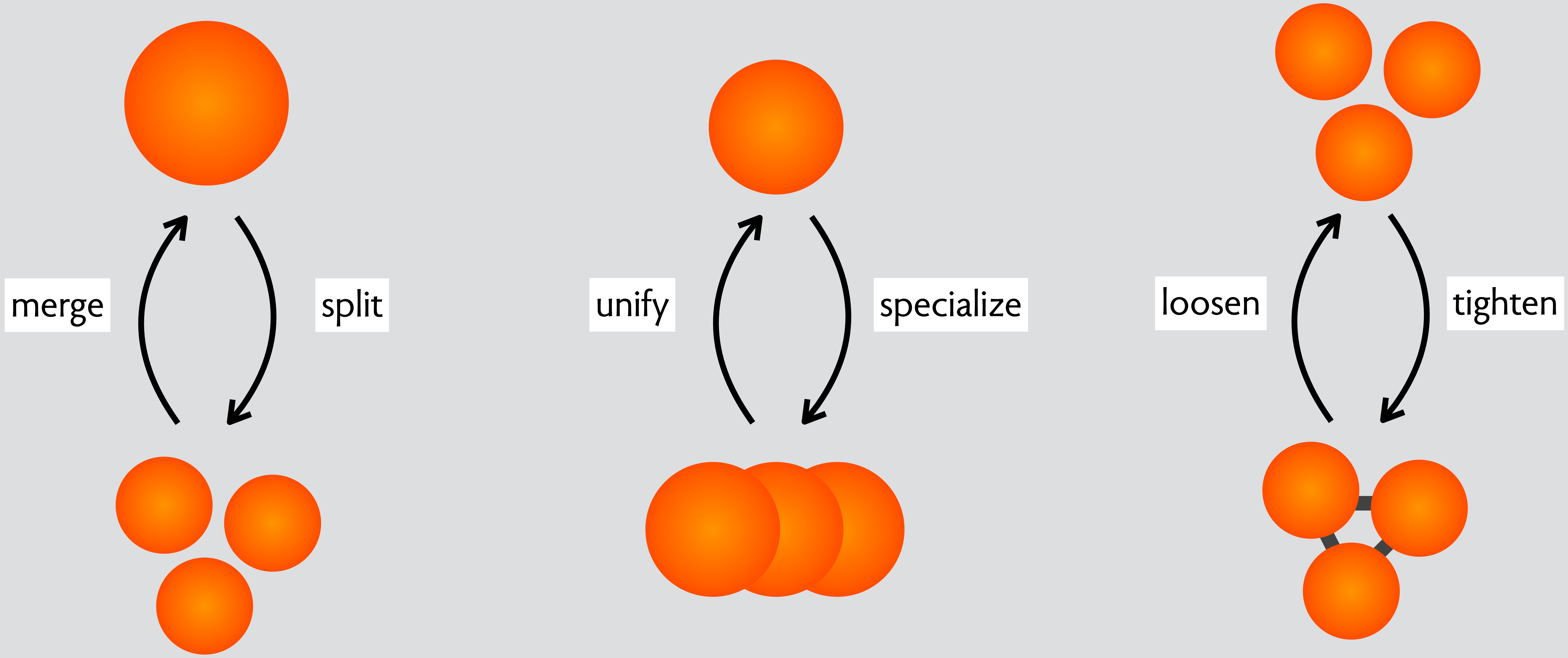
merge

split

unify

specialize

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

general-purpose lens

specialize

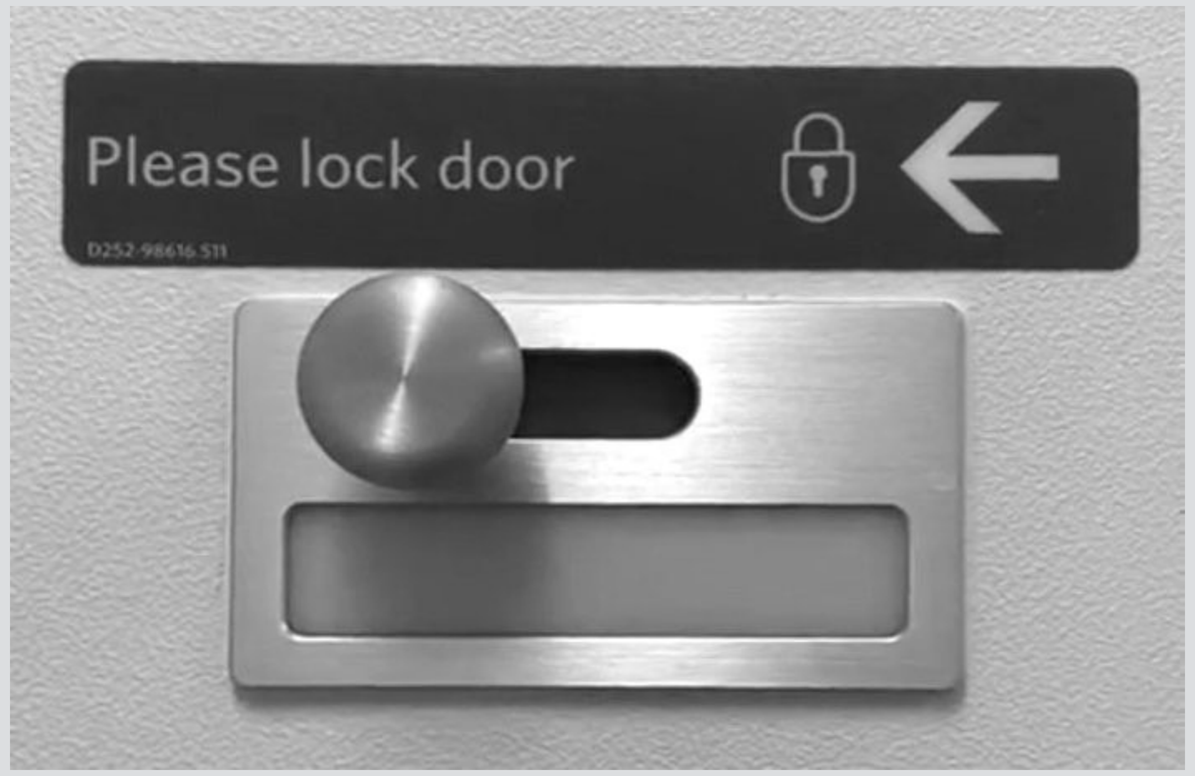
macro lens

tighten-loosen: tradeoff automation/flexibility

tighten-loosen: tradeoff automation/flexibility

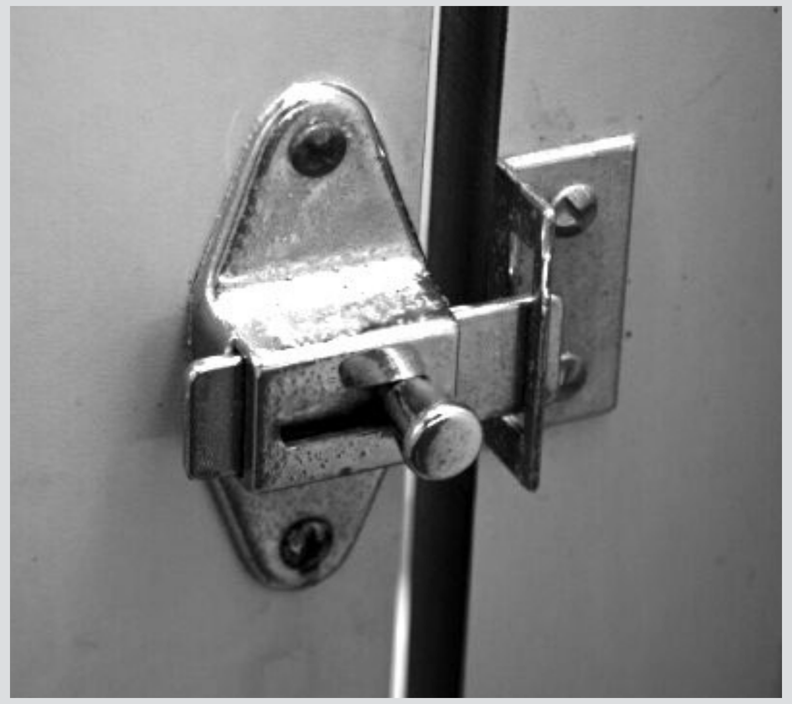
light pull / door lock

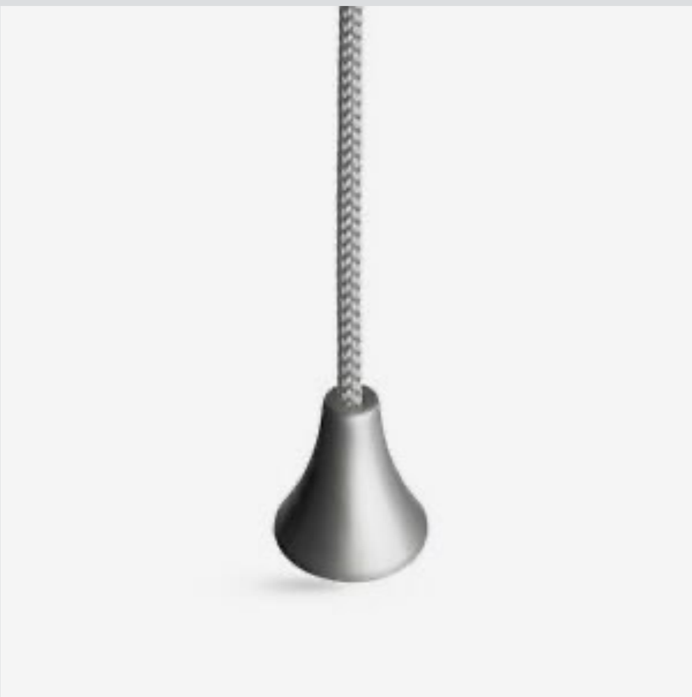
tighten



airplane toilet lock

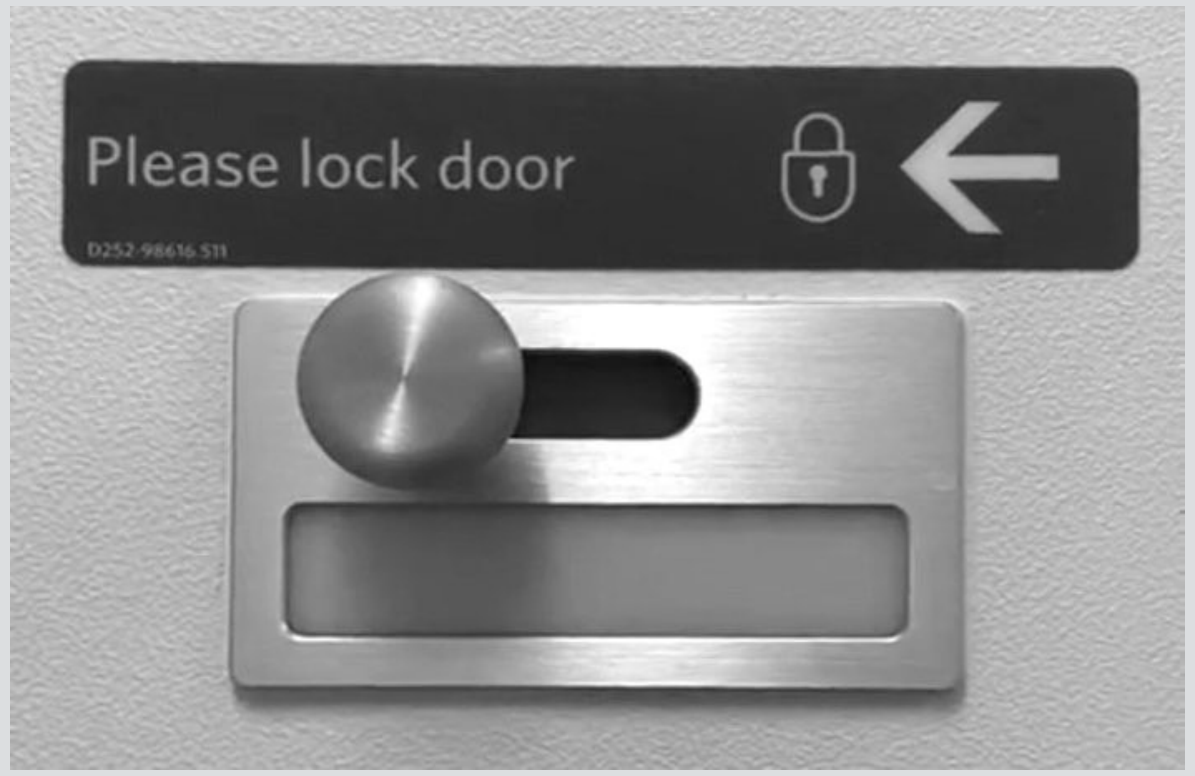
tighten-loosen: tradeoff automation/flexibility



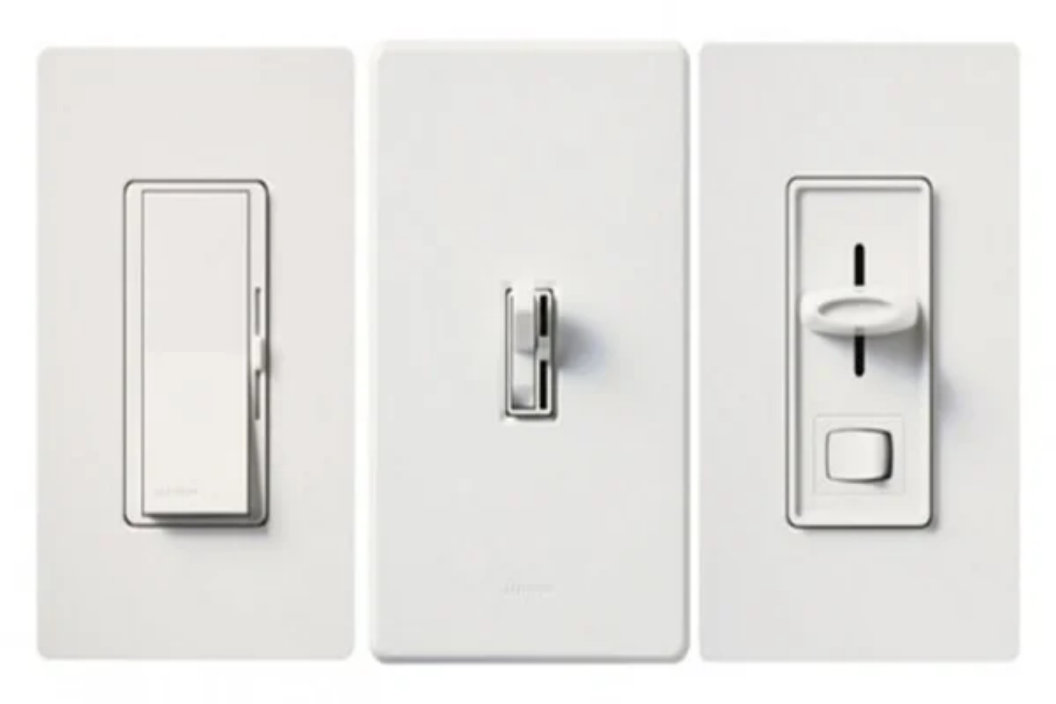


light pull / door lock

tighten

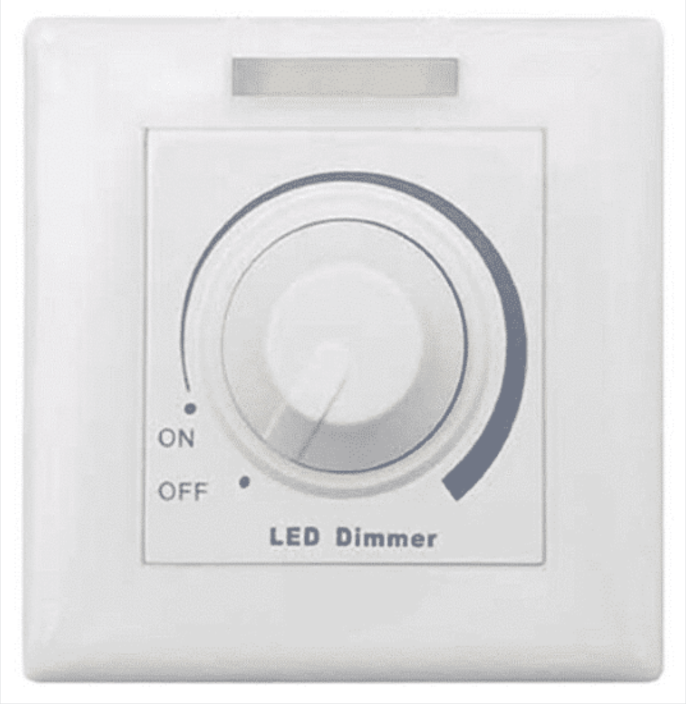


airplane toilet lock



dimmers with separate controls

loosen

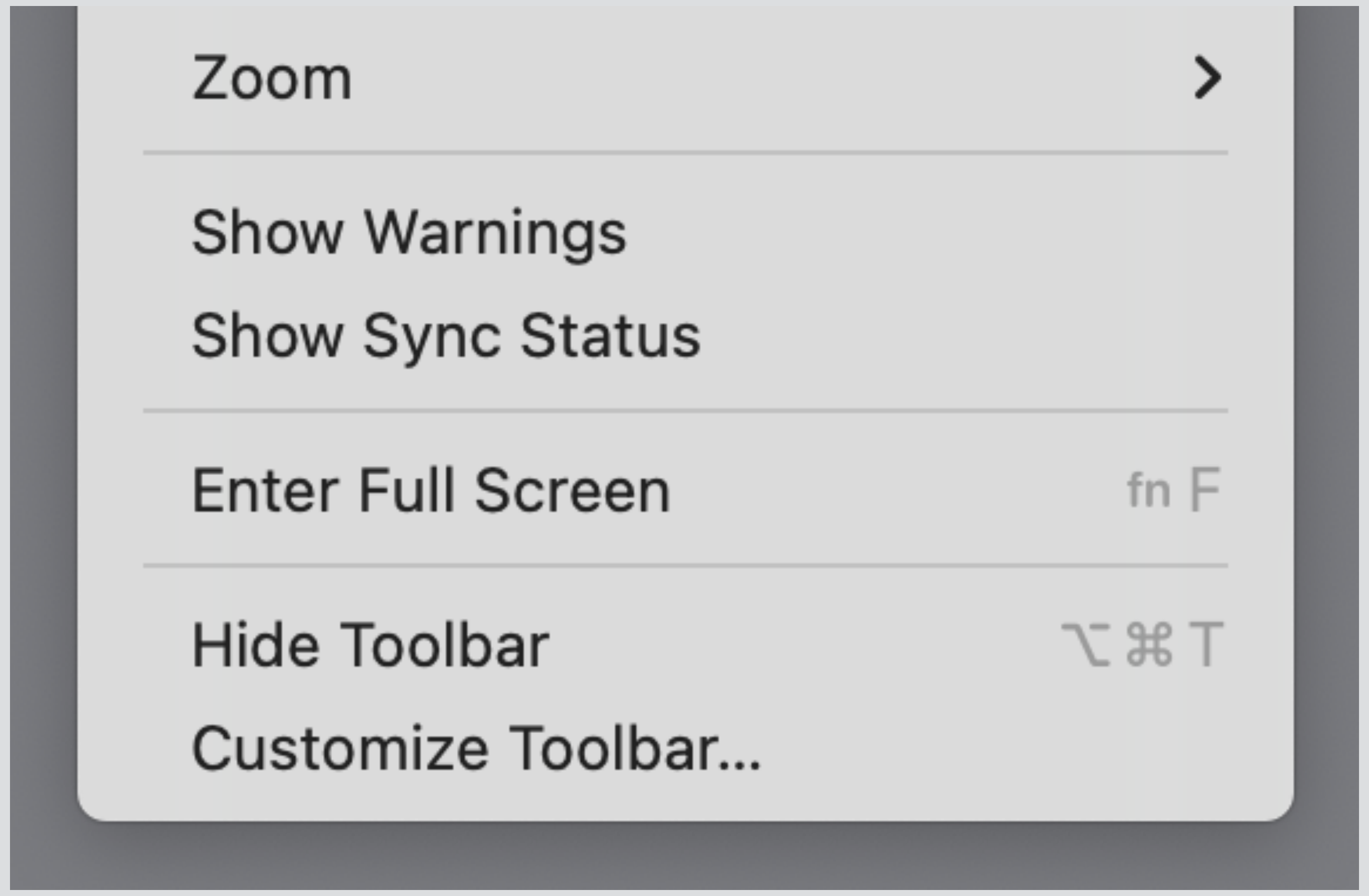


rotary dimmer switch

successful
design moves
in software

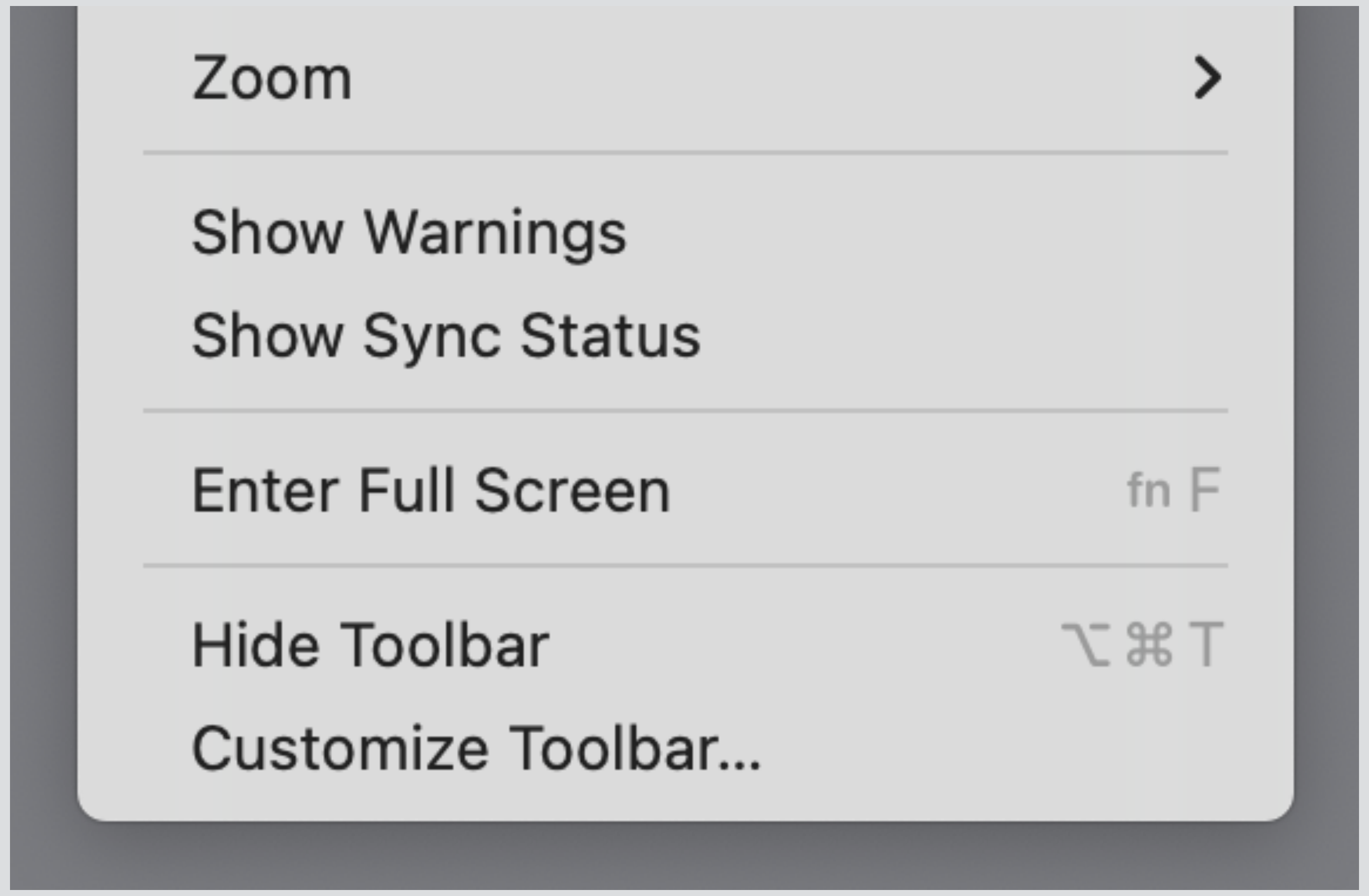
split: emergence of a concept in Keynote

split: emergence of a concept in Keynote

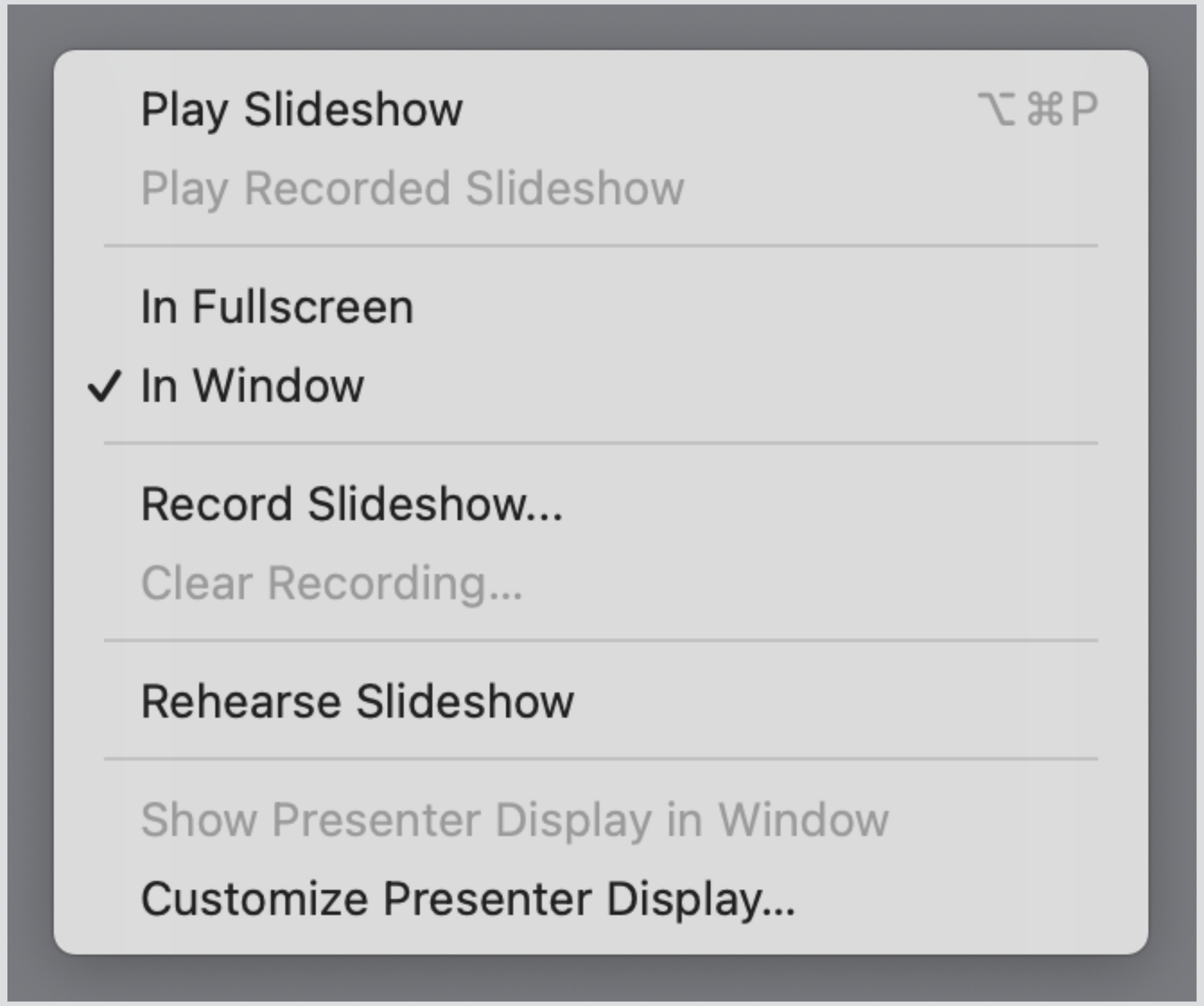


full screen toggle
emerges as partial concept
(c. 2010?)

split: emergence of a concept in Keynote

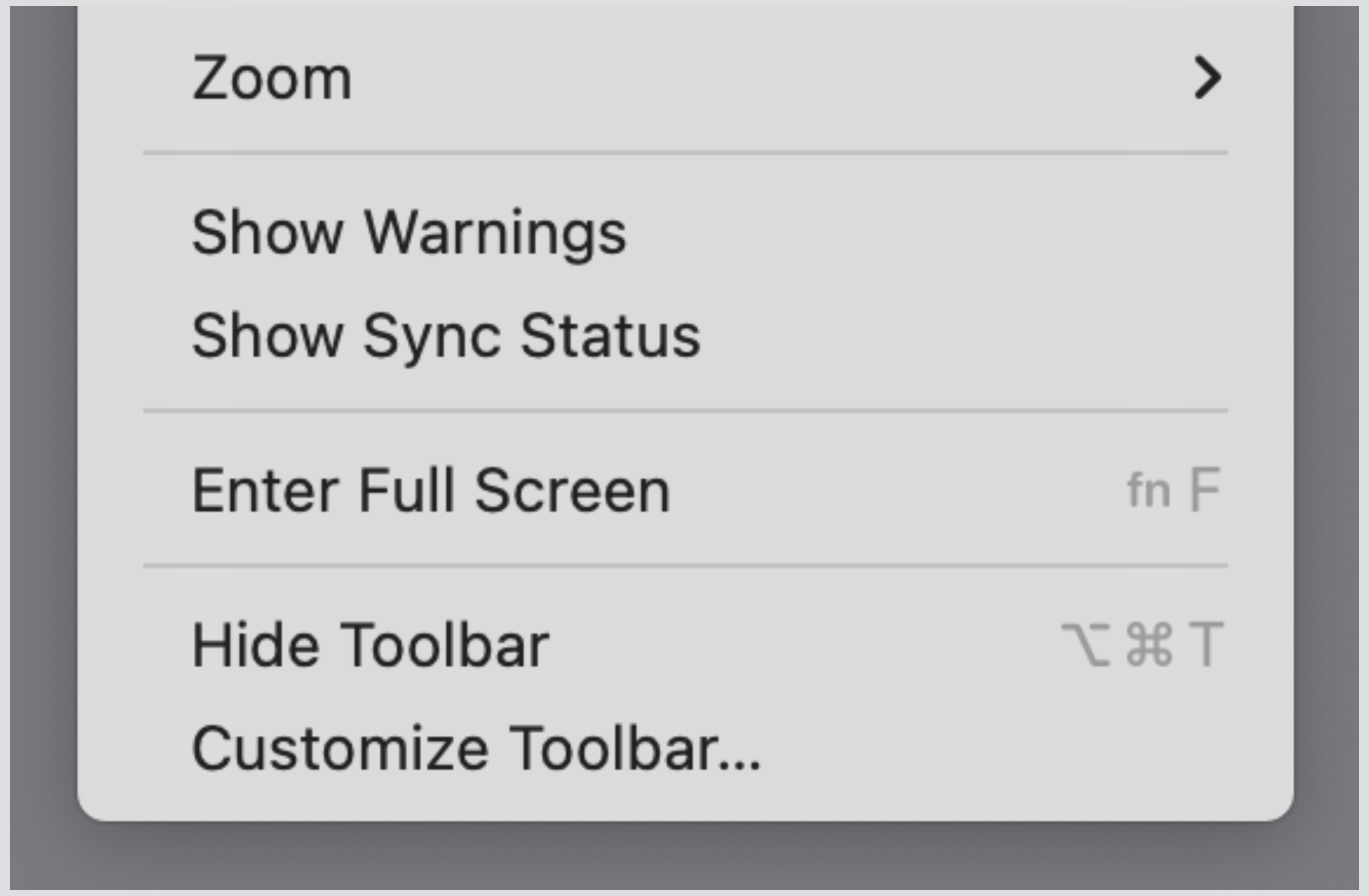


full screen toggle
emerges as partial concept
(c. 2010?)

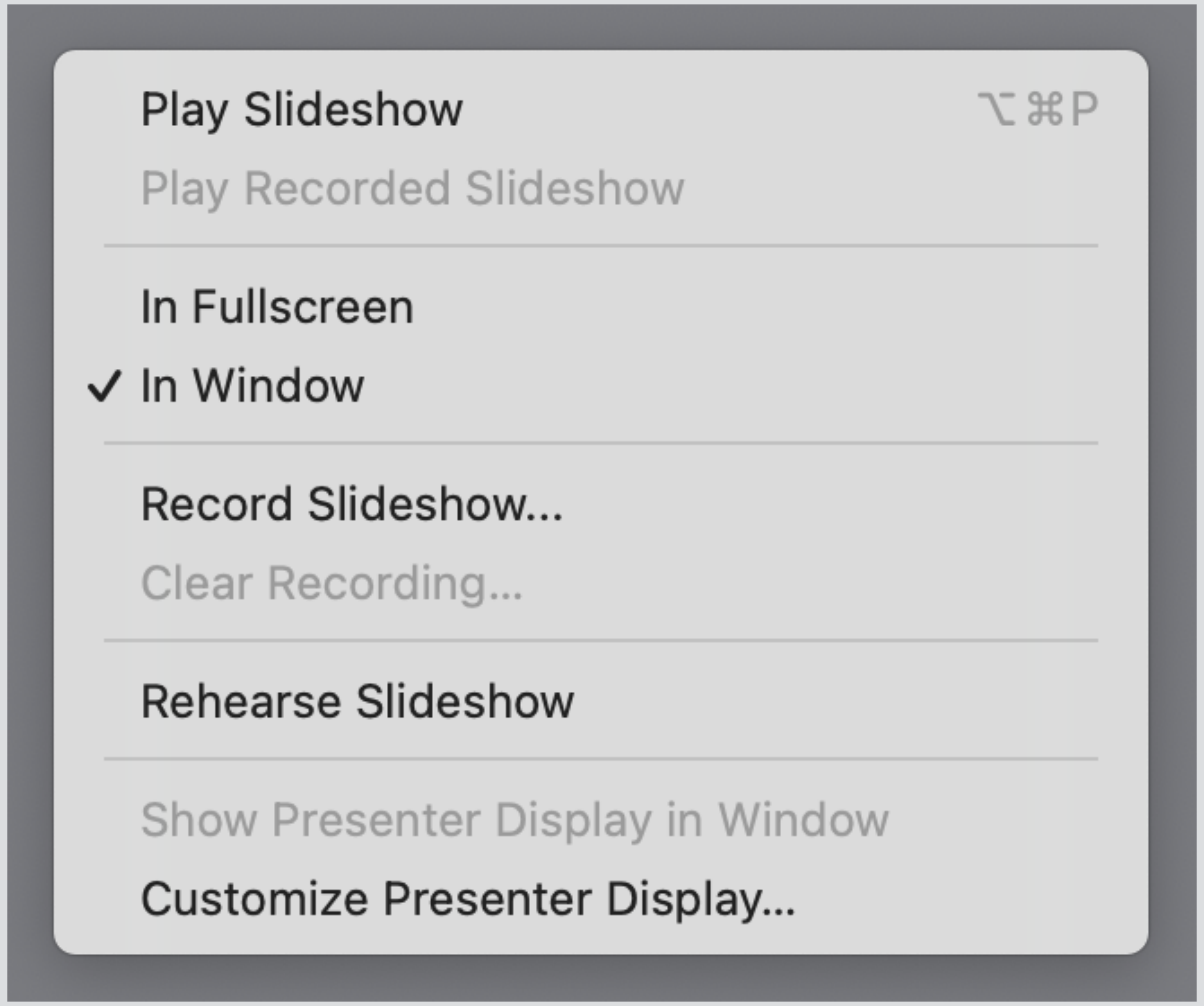


play-in-window option
now an independent concept
(2021)

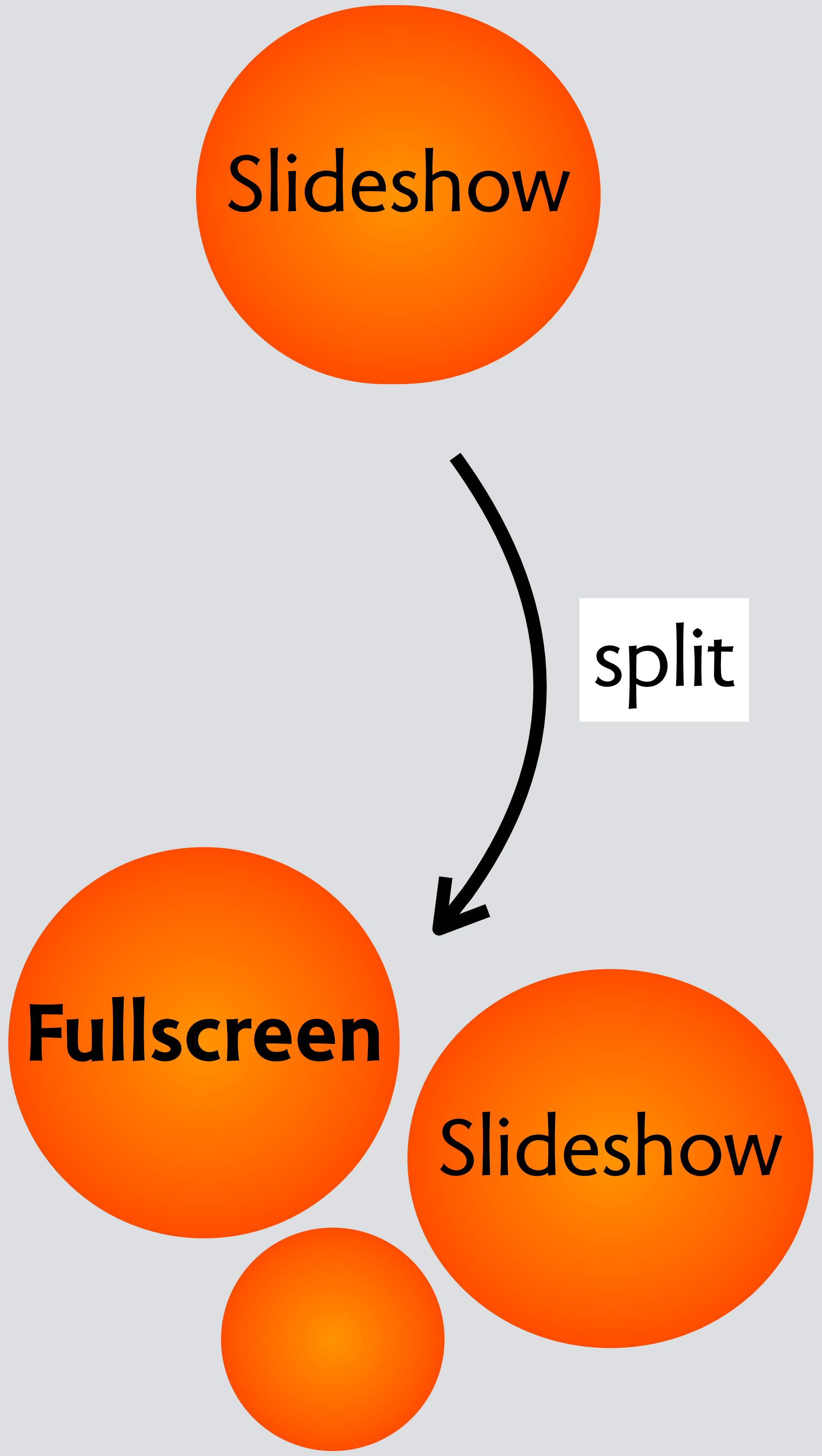
split: emergence of a concept in Keynote



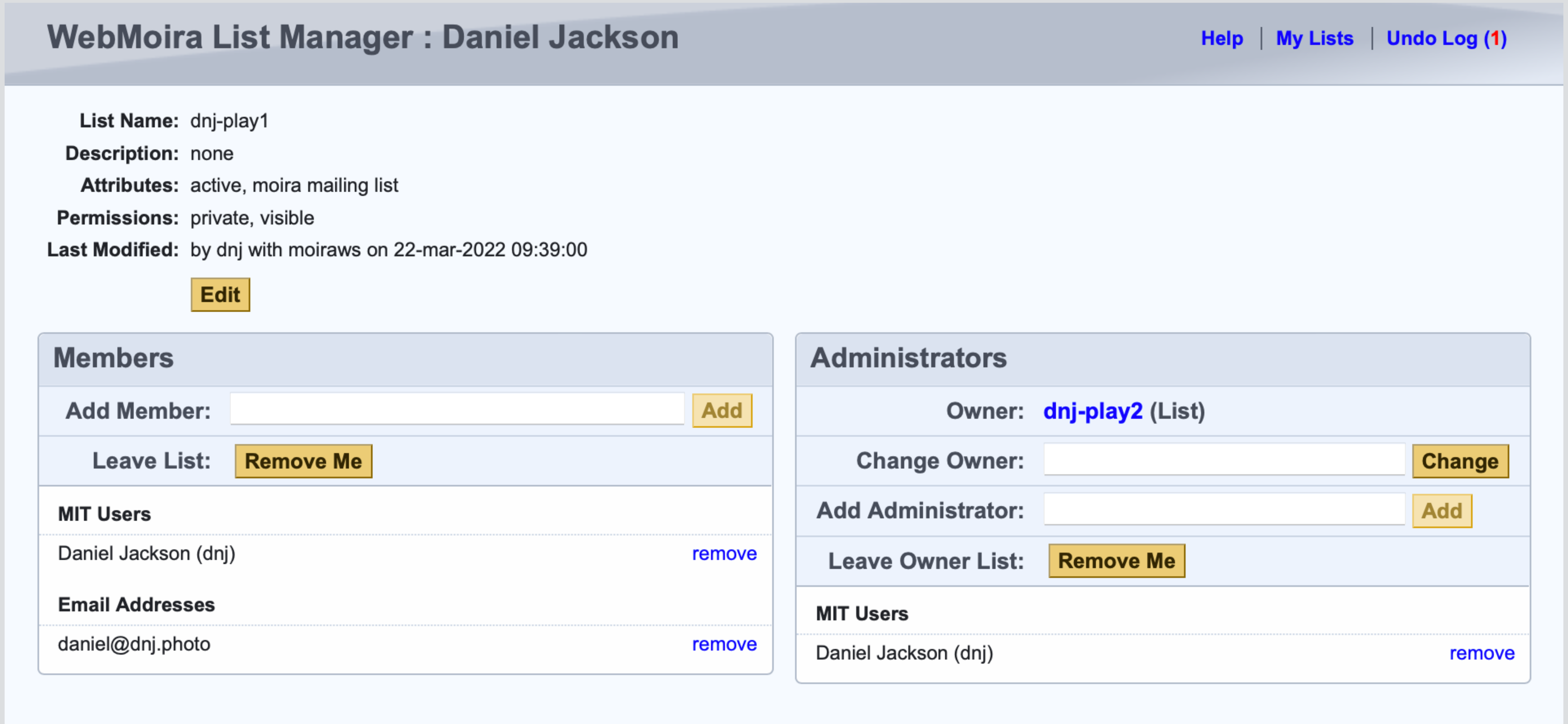
full screen toggle
emerges as partial concept
(c. 2010?)



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



unify: subsuming access control in MIT's Moira



WebMoira List Manager : Daniel Jackson

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

Members

Add Member:

Leave List:

MIT Users

Daniel Jackson (dnj)

Email Addresses

daniel@dnj.photo

Administrators

Owner: dnj-play2 (List)

Change Owner:

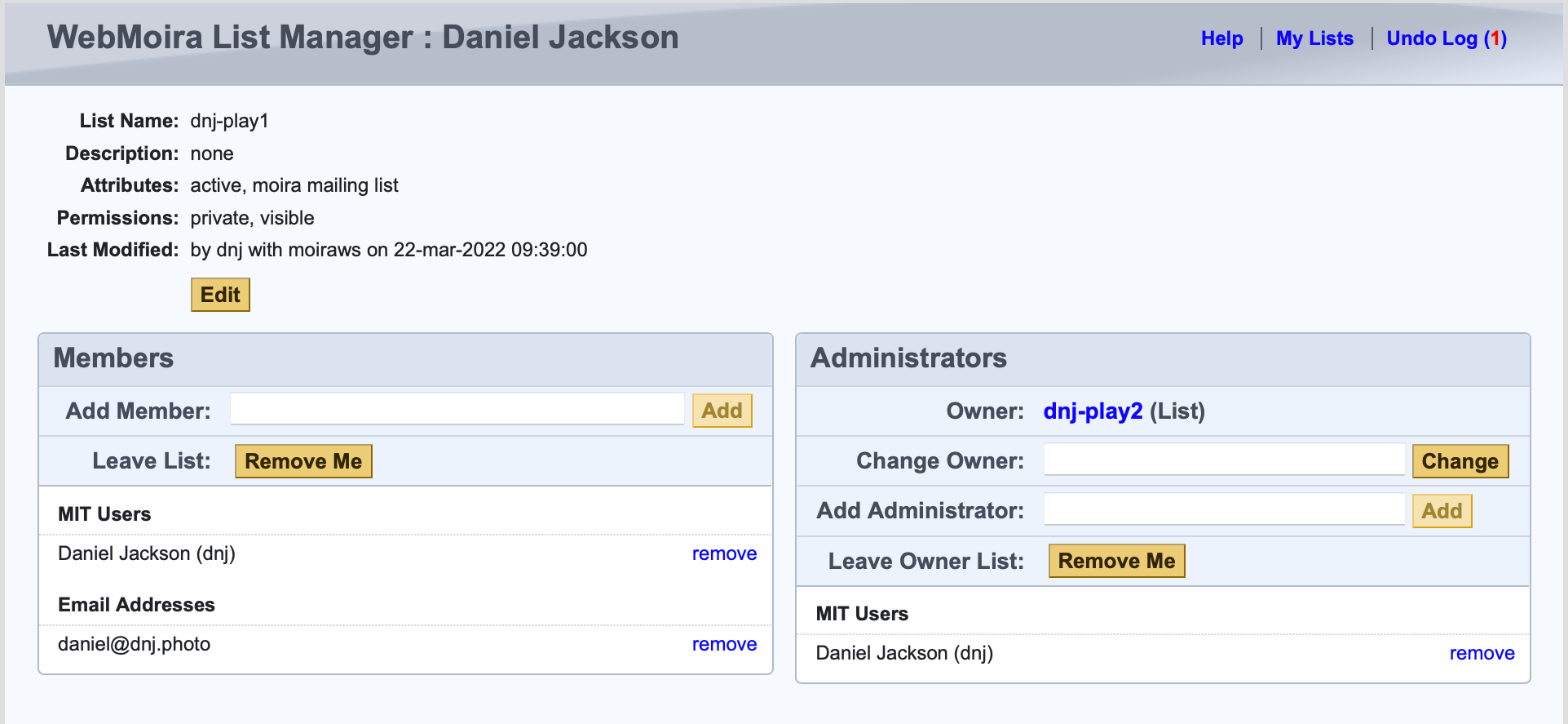
Add Administrator:

Leave Owner List:

MIT Users

Daniel Jackson (dnj)

unify: subsuming access control in MIT's Moira



WebMoira List Manager : Daniel Jackson

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

Members

Add Member:

Leave List:

MIT Users

Daniel Jackson (dnj) remove

Email Addresses

daniel@dnj.photo remove

Administrators

Owner: dnj-play2 (List)

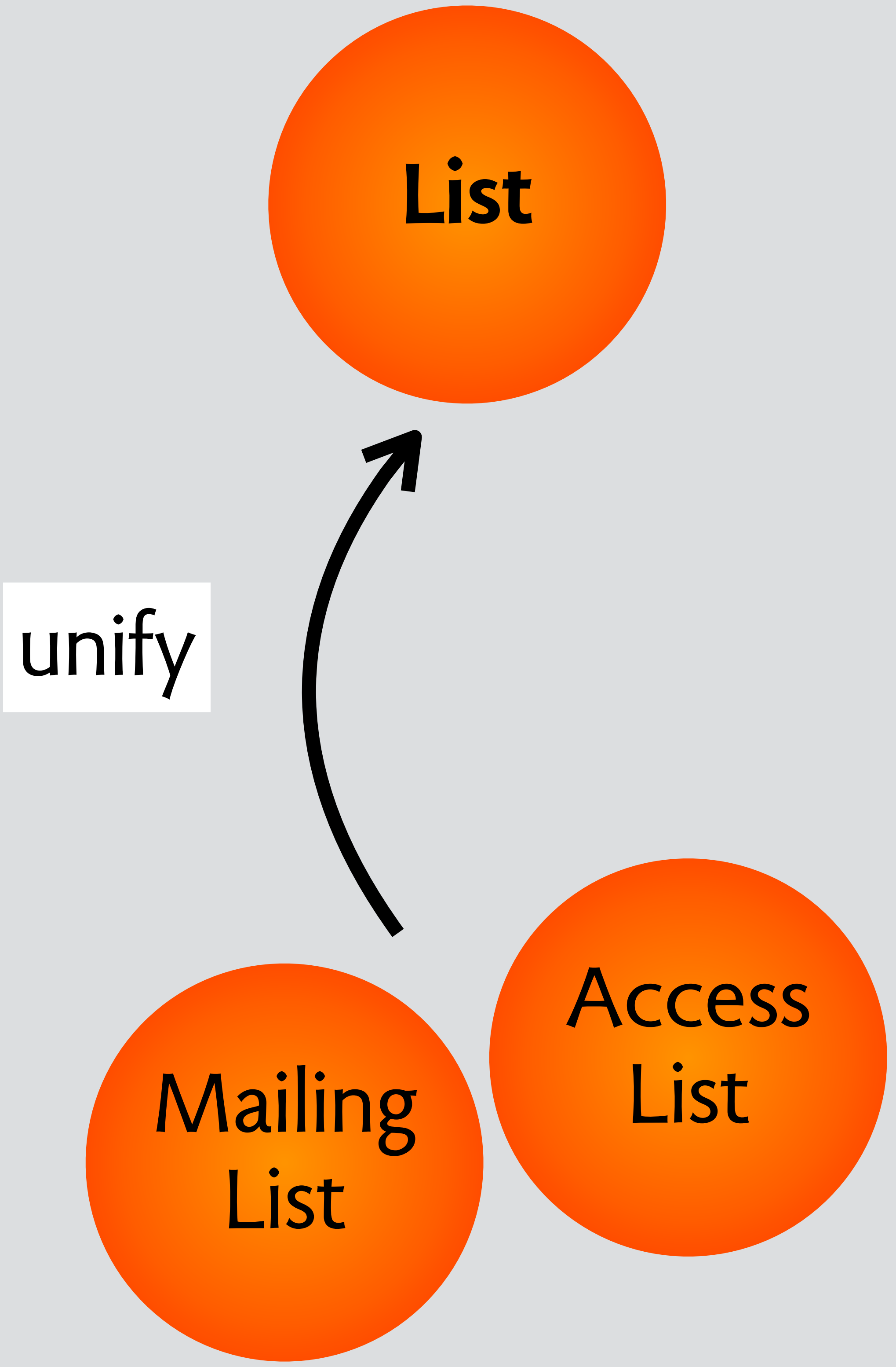
Change Owner:

Add Administrator:

Leave Owner List:

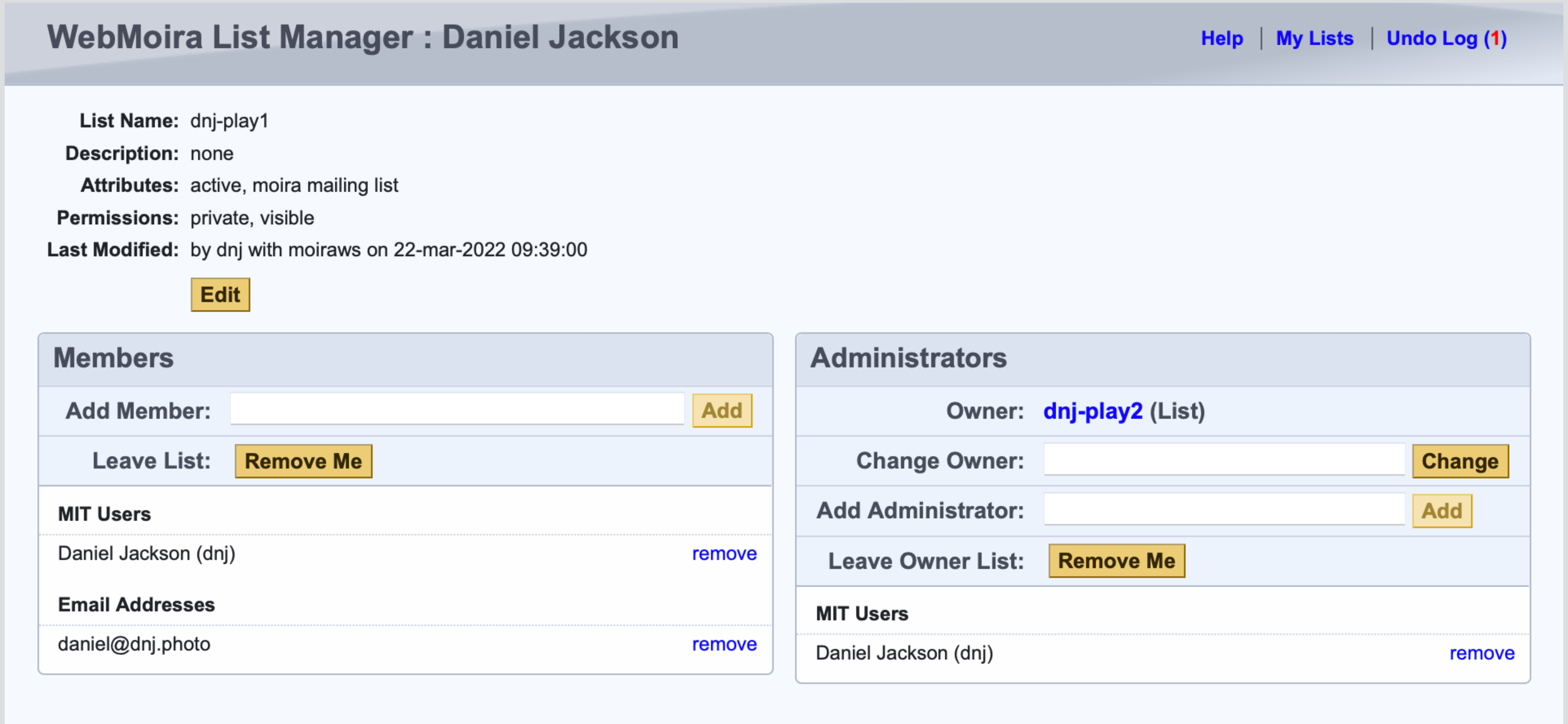
MIT Users

Daniel Jackson (dnj) 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: Add

Leave List: Remove Me

MIT Users

Daniel Jackson (dnj) remove

Email Addresses

daniel@dnj.photo remove

Administrators

Owner: dnj-play2 (List)

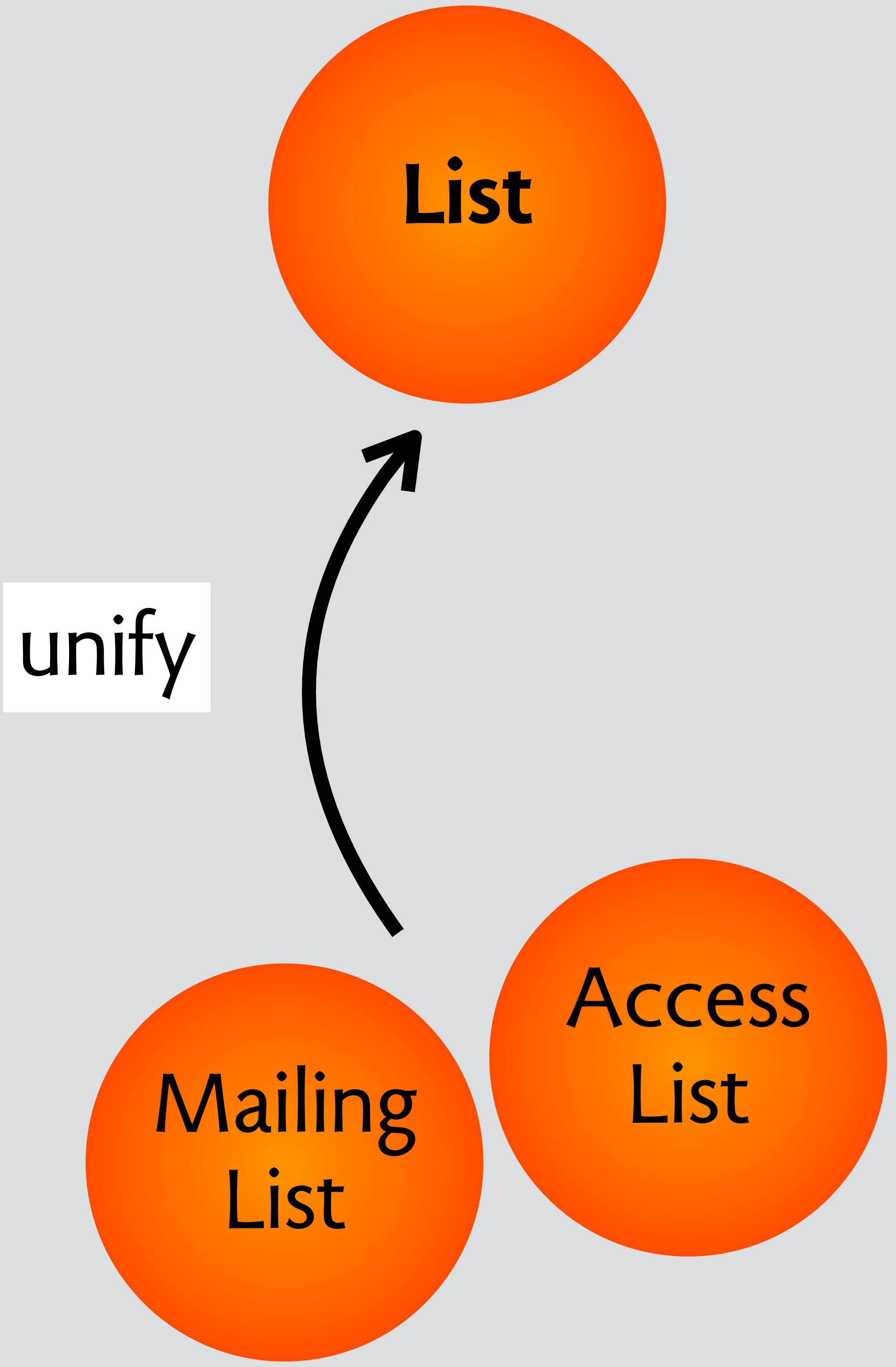
Change Owner: Change

Add Administrator: Add

Leave Owner List: Remove Me

MIT Users

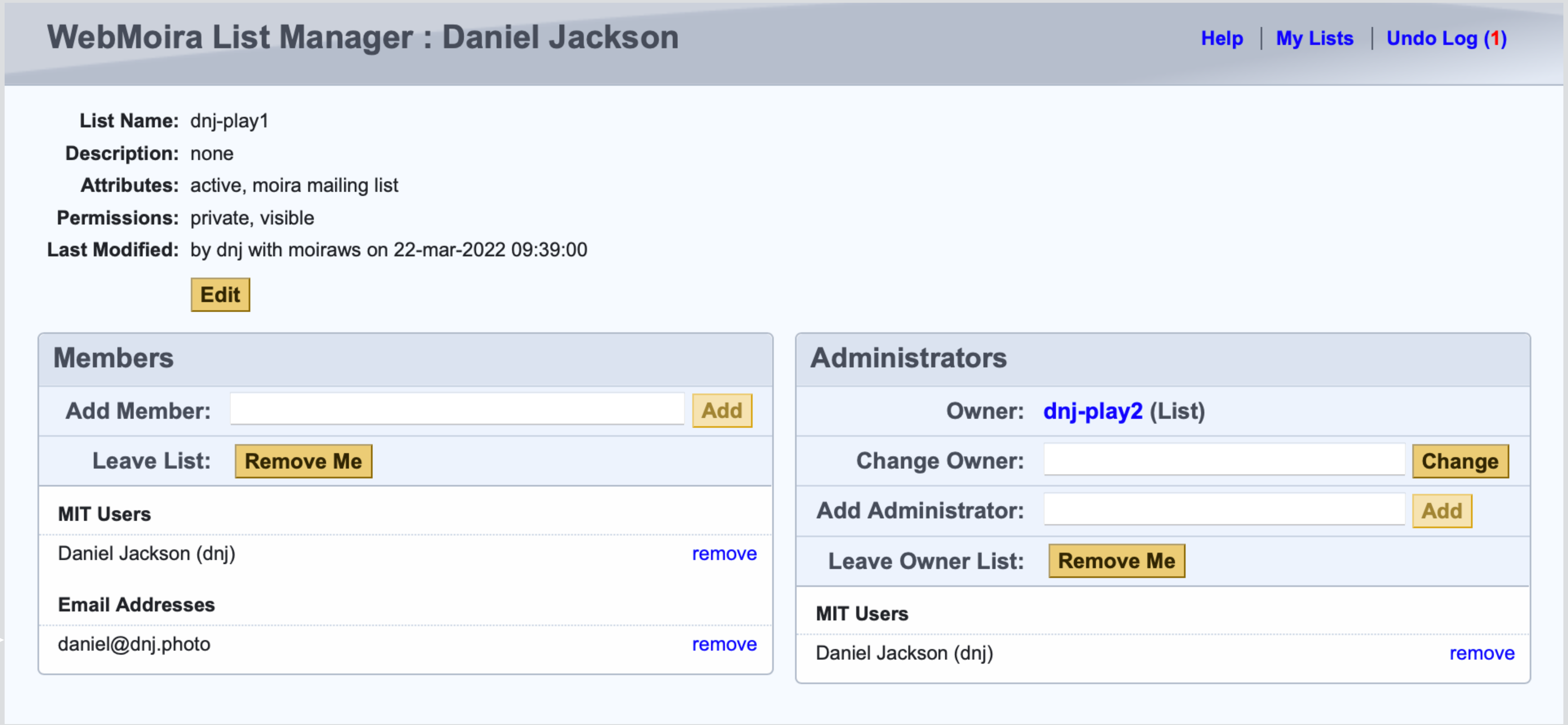
Daniel Jackson (dnj) remove



unify: subsuming access control in MIT's Moira

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

Add Member: Add

Leave List: Remove Me

MIT Users

Daniel Jackson (dnj) remove

Email Addresses

daniel@dnj.photo remove

Administrators

Owner: dnj-play2 (List)

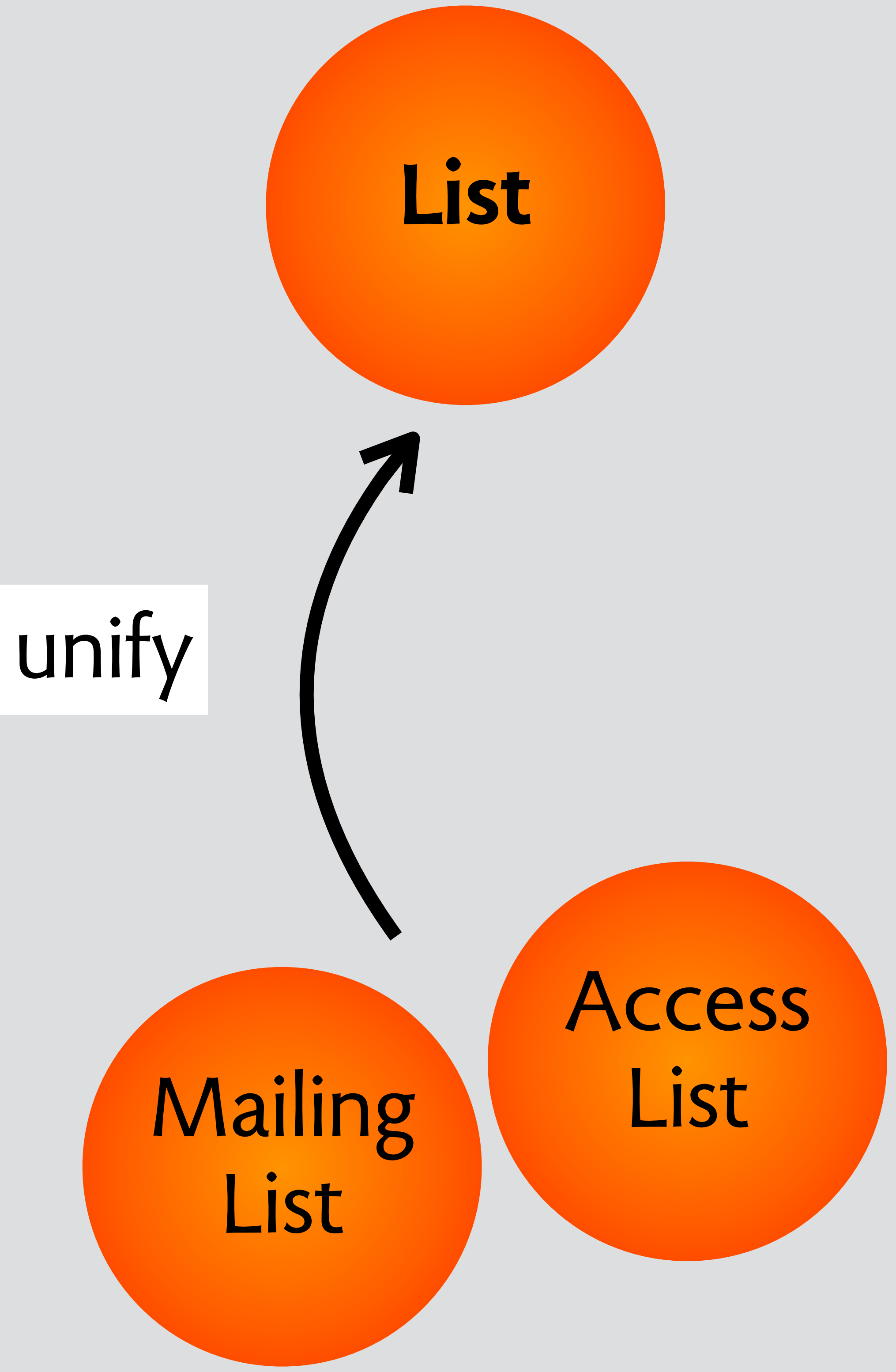
Change Owner: Change

Add Administrator: Add

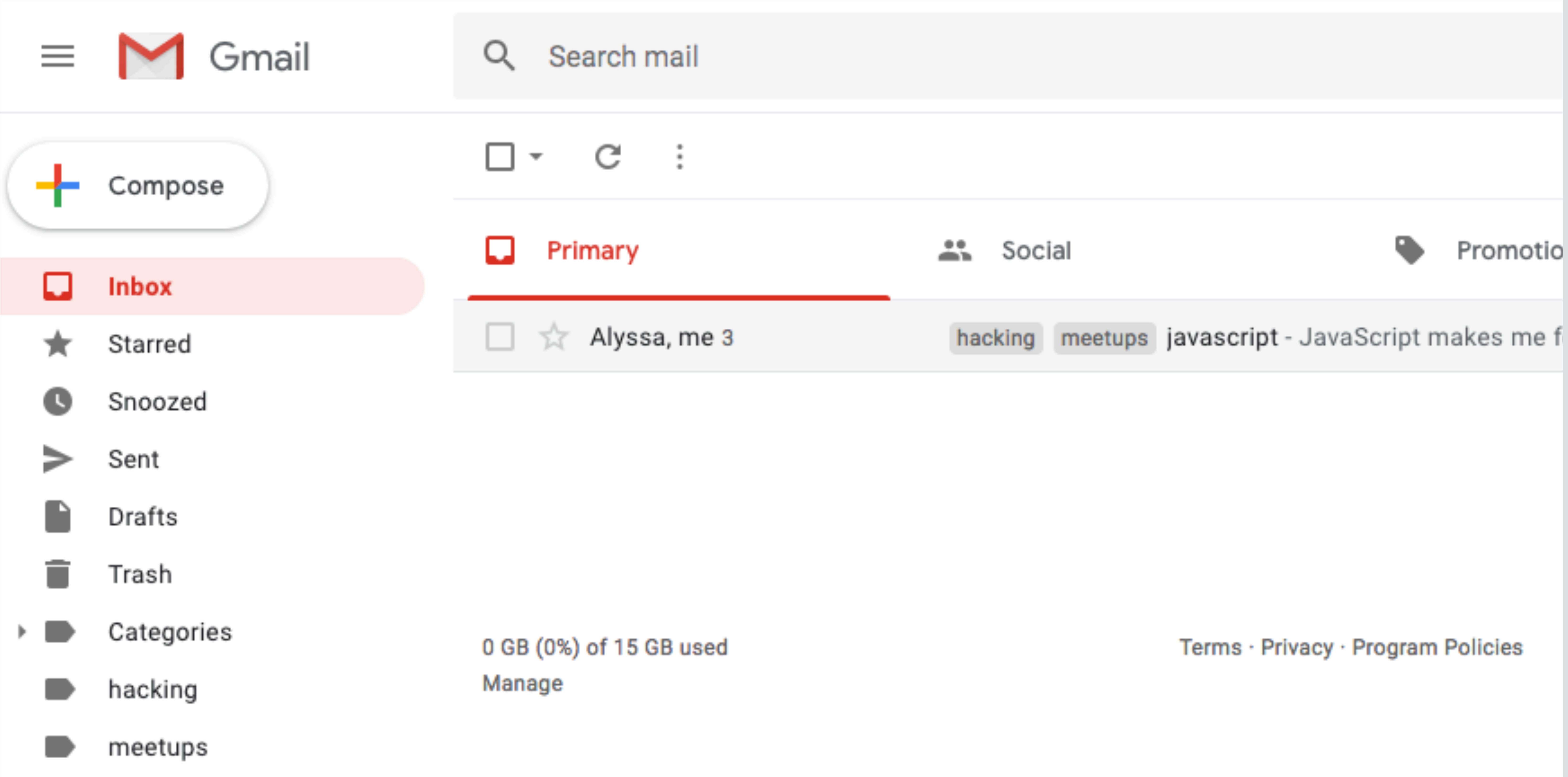
Leave Owner List: Remove Me

MIT Users

Daniel Jackson (dnj) remove



tighten: label and trash concepts in Gmail

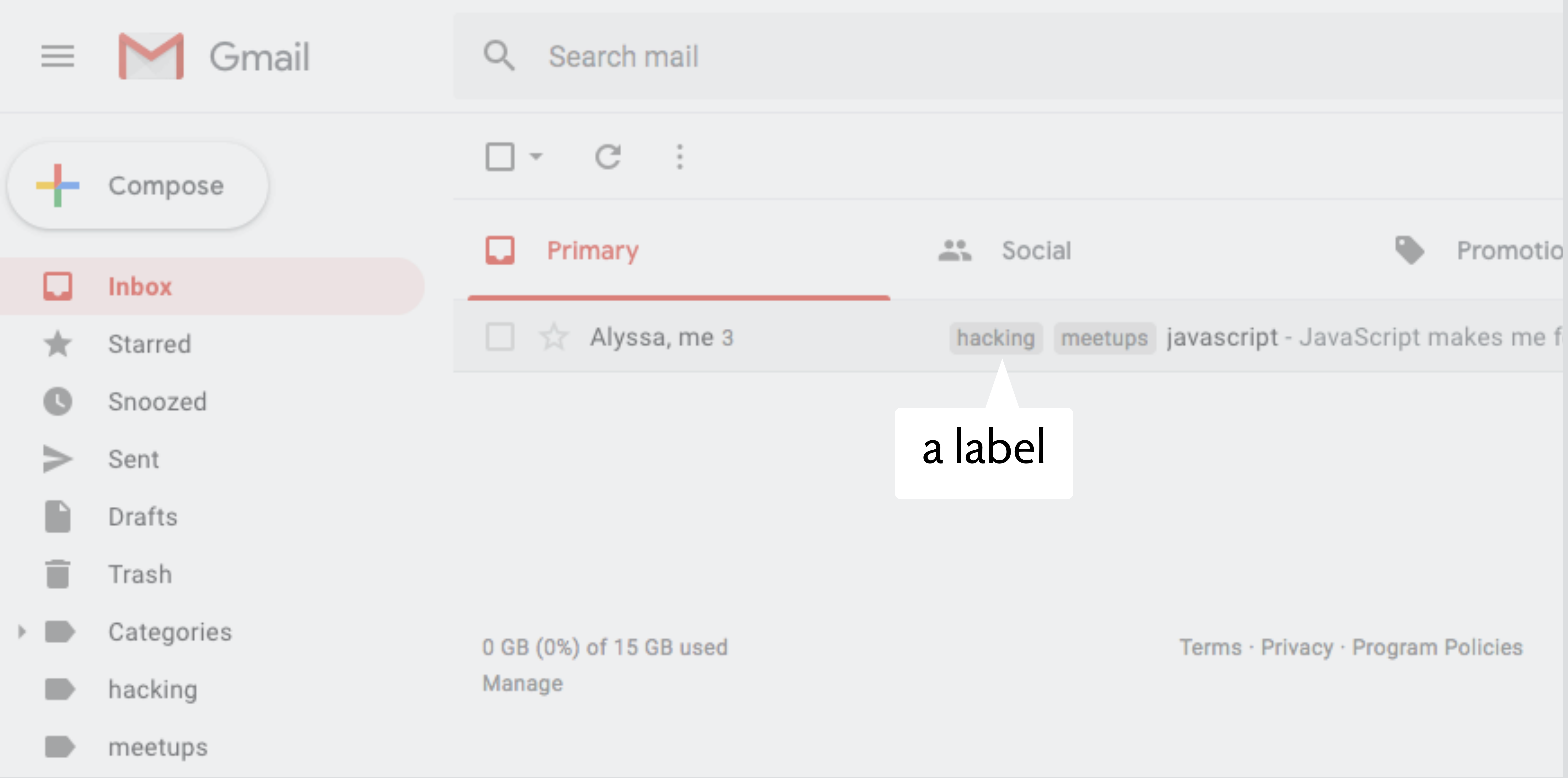


□ ☆ Alyssa, me 3 hacking meetups javascript - JavaScript makes me f

0 GB (0%) of 15 GB used
Manage

Terms · Privacy · Program Policies

tighten: label and trash concepts in Gmail



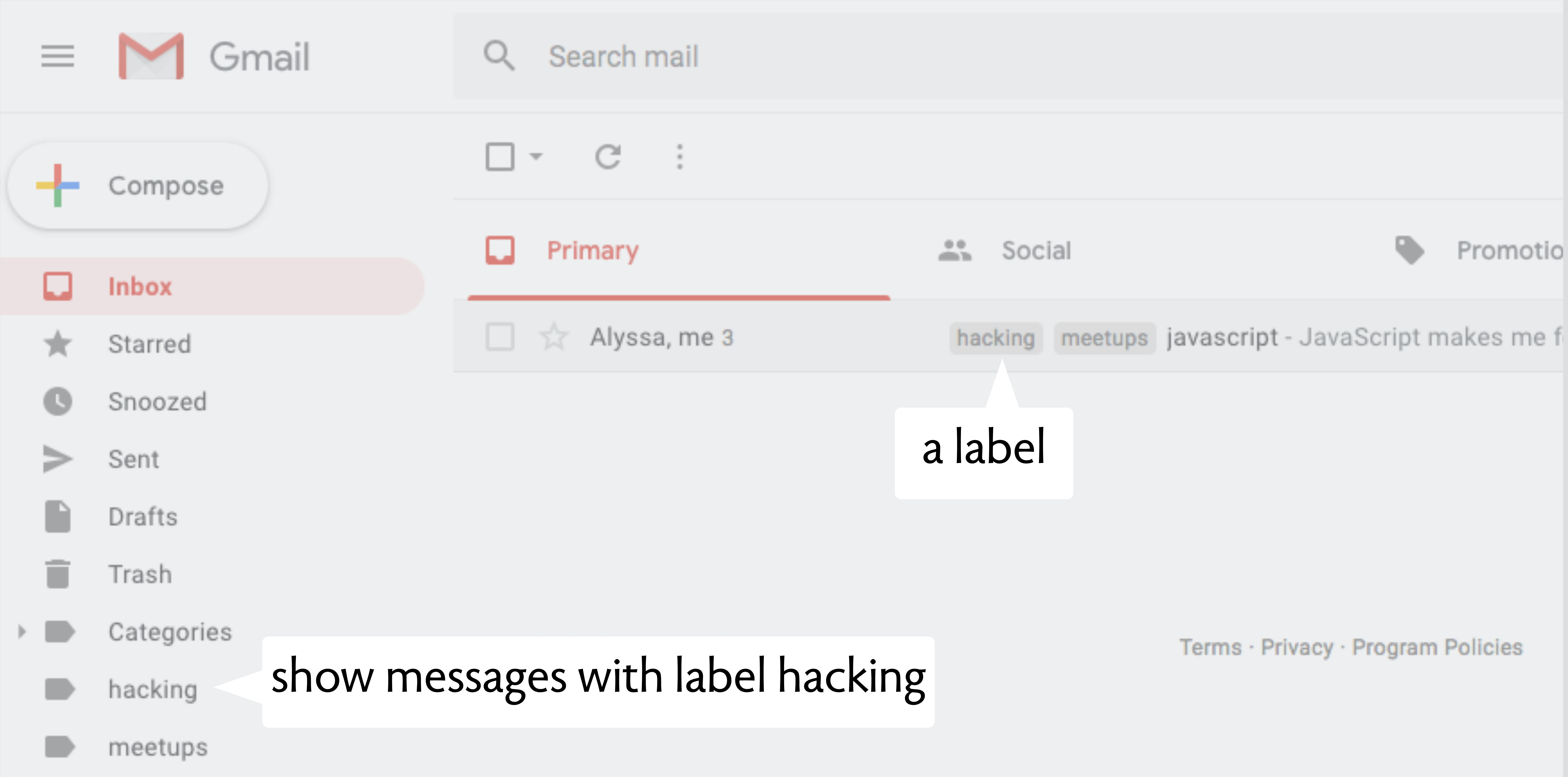
Alyssa, me 3 hacking meetups javascript - JavaScript makes me f

0 GB (0%) of 15 GB used
Manage

Terms · Privacy · Program Policies

a label

tighten: label and trash concepts in Gmail

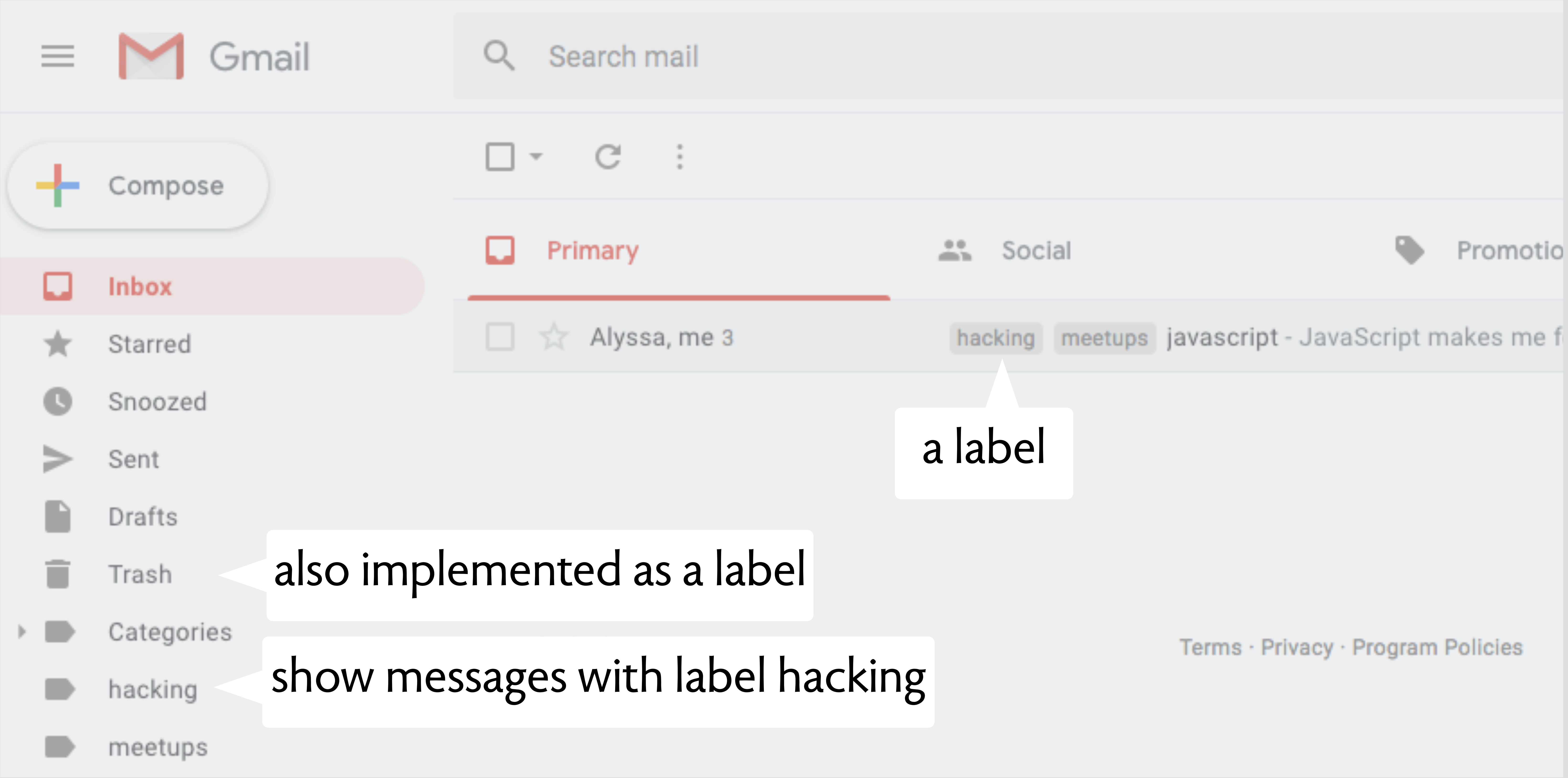


Alyssa, me 3 hacking meetups javascript - JavaScript makes me f

a label

show messages with label hacking

tighten: label and trash concepts in Gmail



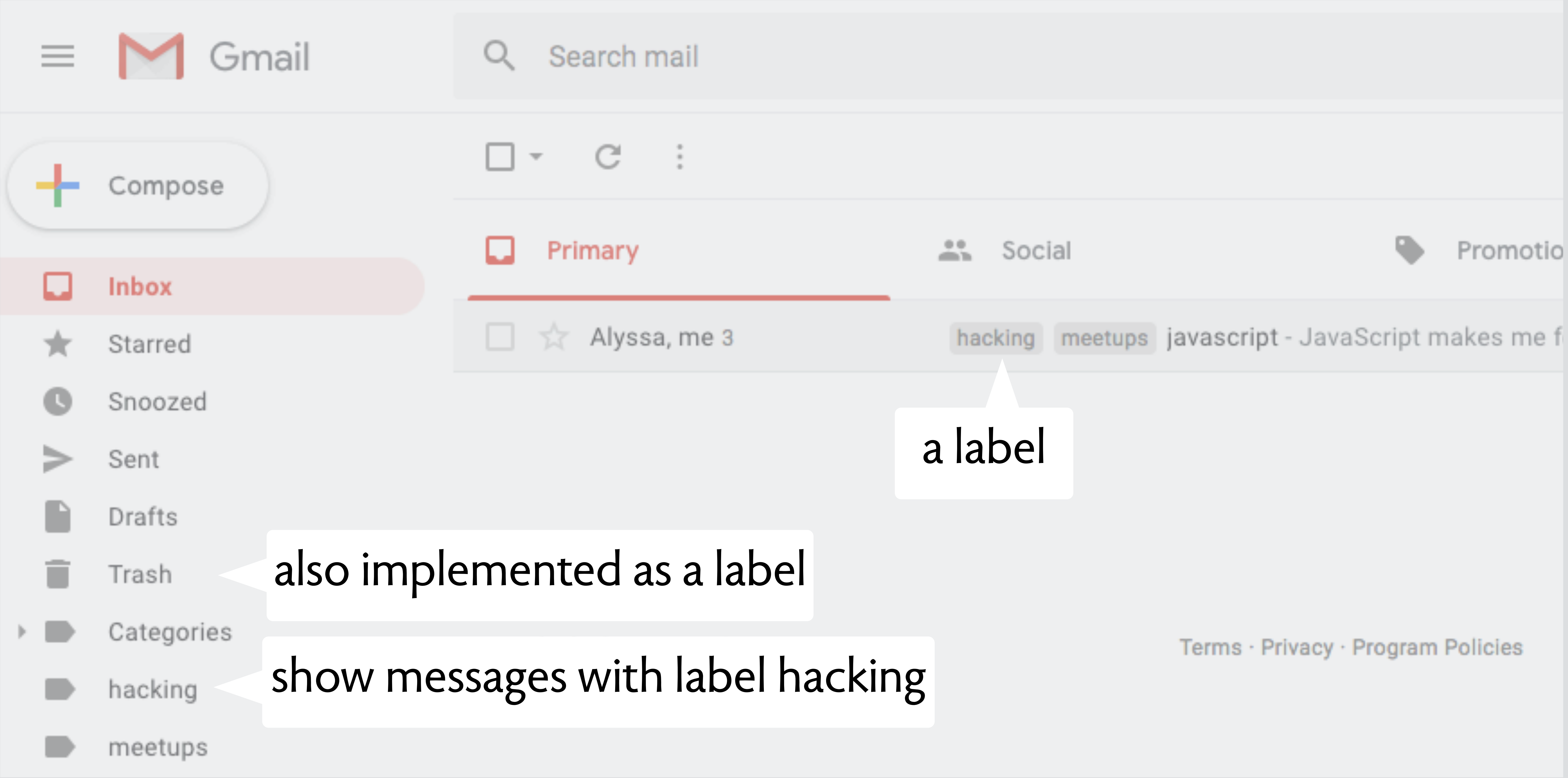
Alyssa, me 3 hacking meetups javascript - JavaScript makes me f

a label

also implemented as a label

show messages with label hacking

tighten: label and trash concepts in Gmail

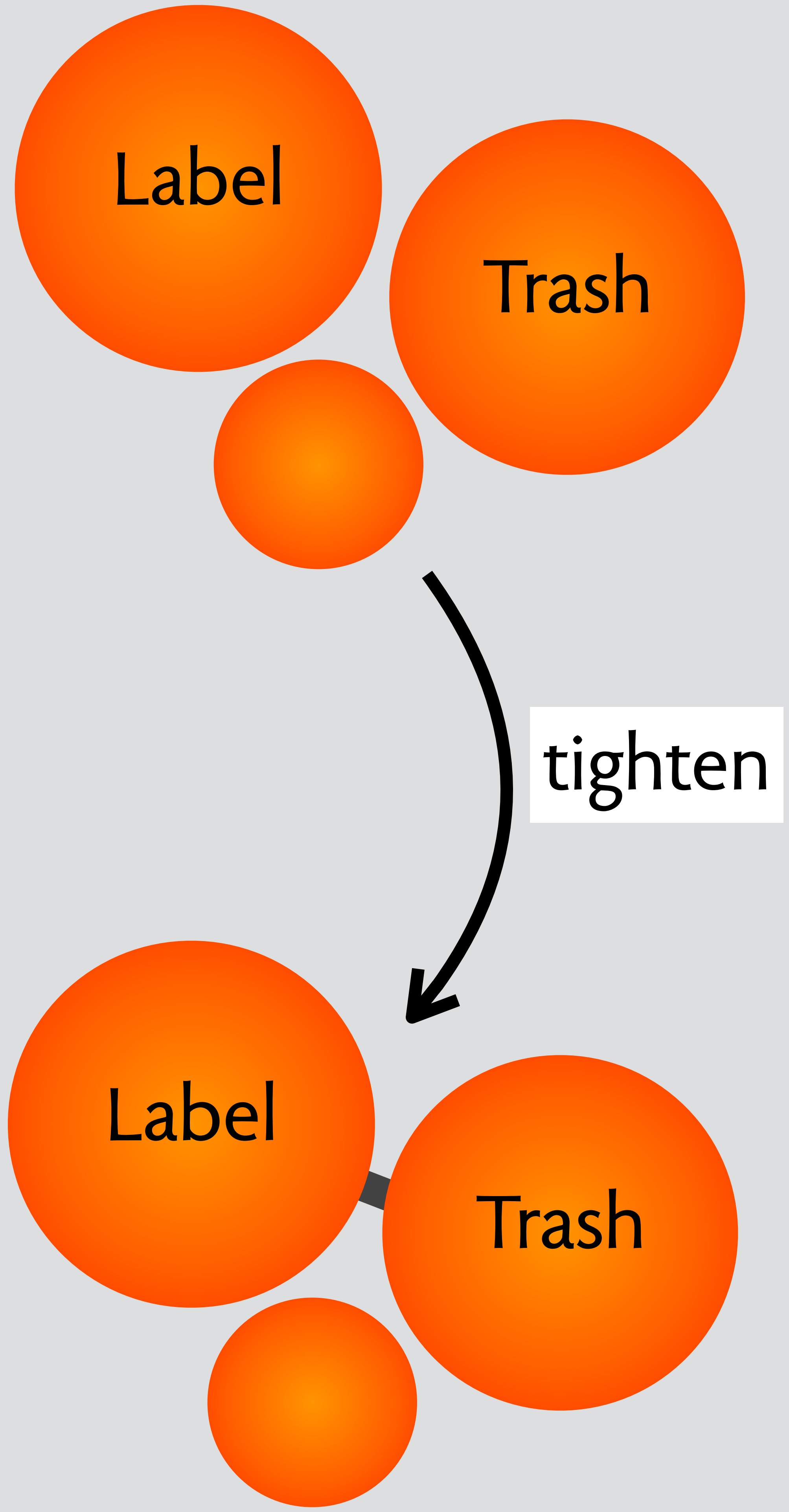


Alyssa, me 3 hacking meetups javascript - JavaScript makes me f

a label

also implemented as a label

show messages with label hacking



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

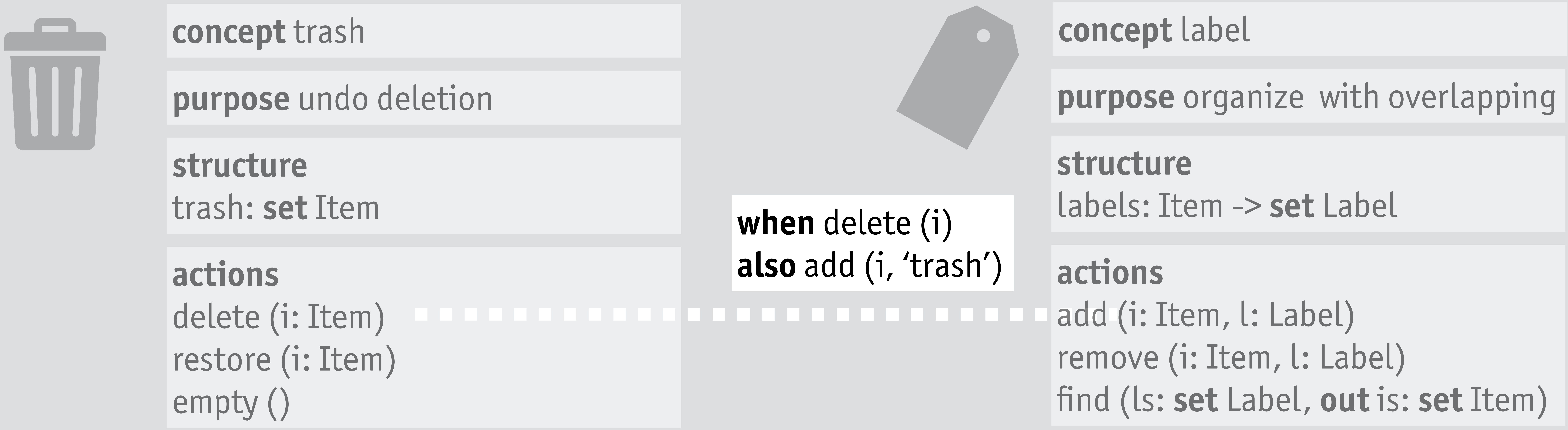
actions
delete (i: Item)
restore (i: Item)
empty ()

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 ()

when delete (i)
also add (i, ‘trash’)

concept label

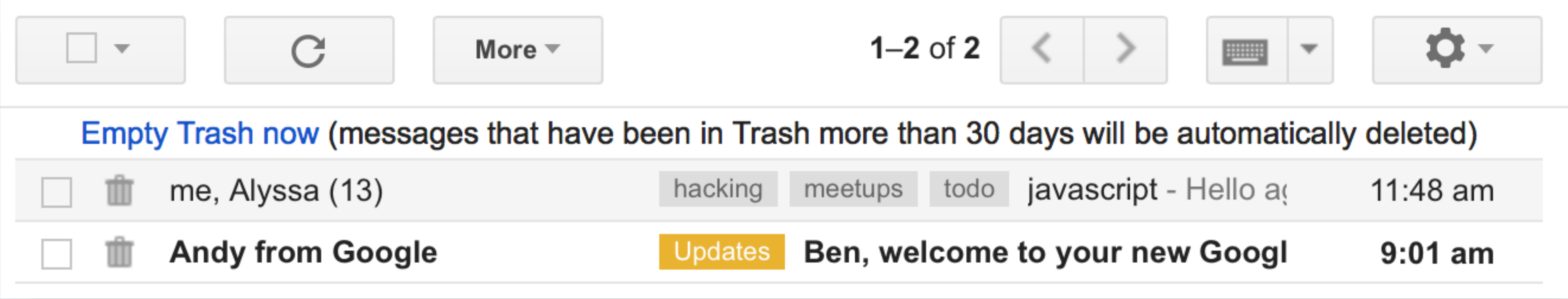
purpose organize with overlapping

structure
labels: Item -> **set** Label

actions
add (i: Item, l: Label)
remove (i: Item, l: Label)
find (ls: **set** Label, **out** is: **set** Item)

integrating these concepts is tricky

click on trash

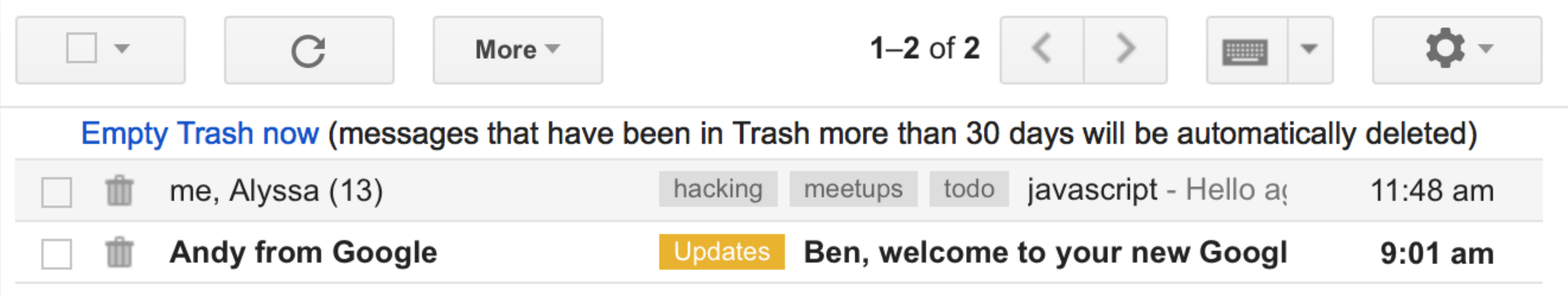


Empty Trash now (messages that have been in Trash more than 30 days will be automatically deleted)

□	me, Alyssa (13)	hacking meetups todo	javascript - Hello ag	11:48 am
□	**Andy from Google**	Updates	**Ben, welcome to your new Googl**	**9:01 am**

integrating these concepts is tricky

click on trash



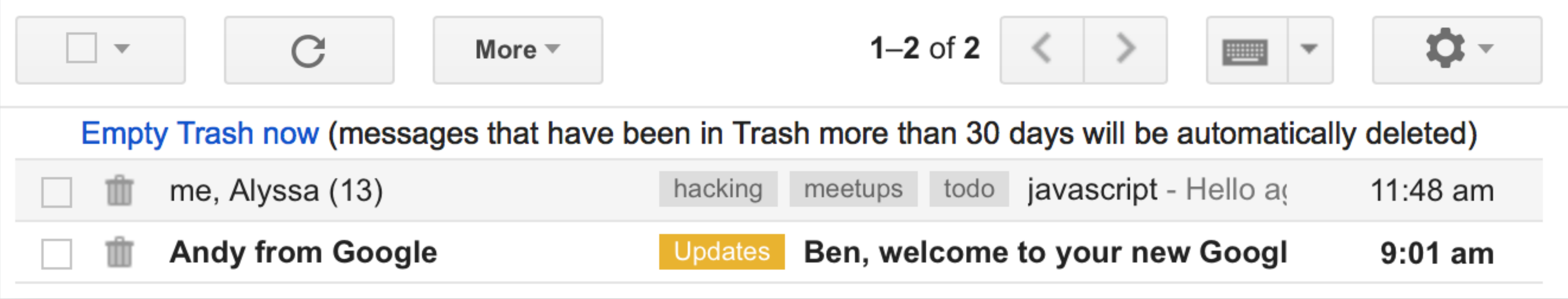
Empty Trash now (messages that have been in Trash more than 30 days will be automatically deleted)

me, Alyssa (13)	hacking meetups todo	javascript - Hello ag	11:48 am
Andy from Google	Updates	**Ben, welcome to your new Googl**	**9:01 am**

filter on todo label

integrating these concepts is tricky

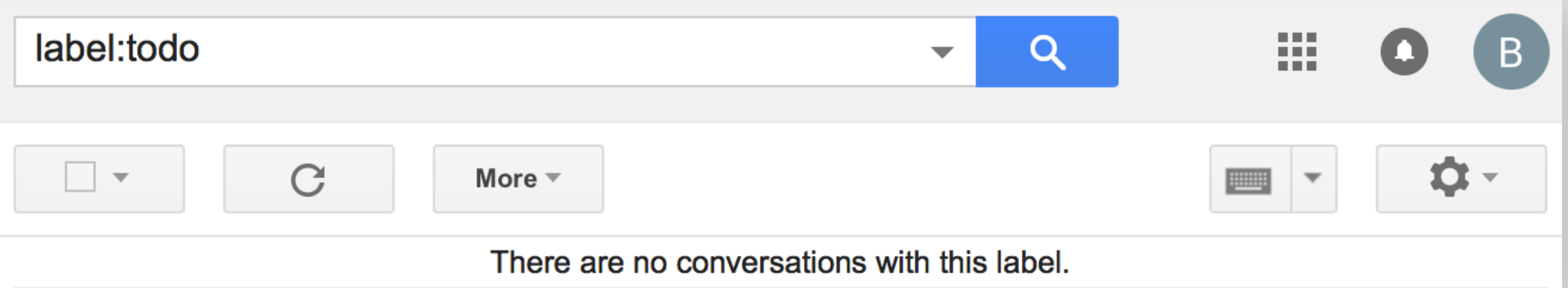
click on trash



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

□	me, Alyssa (13)	hacking meetups todo	javascript - Hello a(	11:48 am
□	**Andy from Google**	Updates	**Ben, welcome to your new Googl**	**9:01 am**

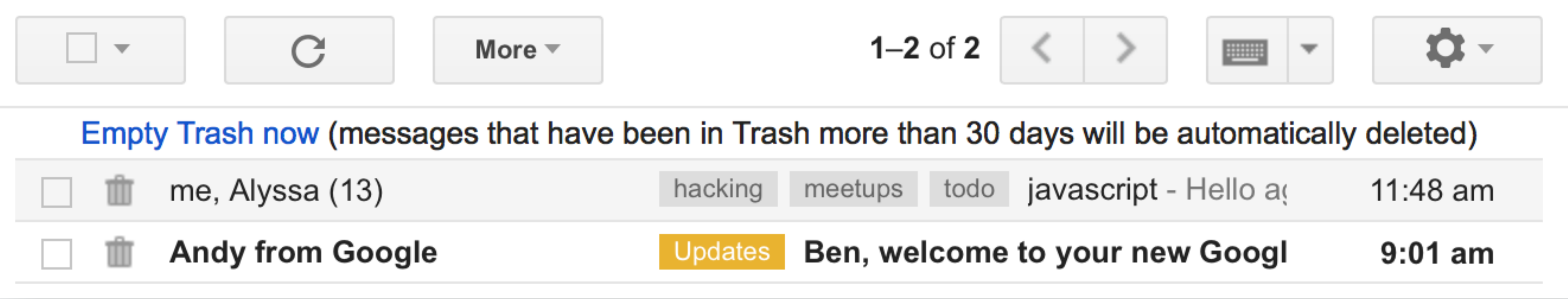
filter on todo label


label:todo

There are no conversations with this label.

integrating these concepts is tricky

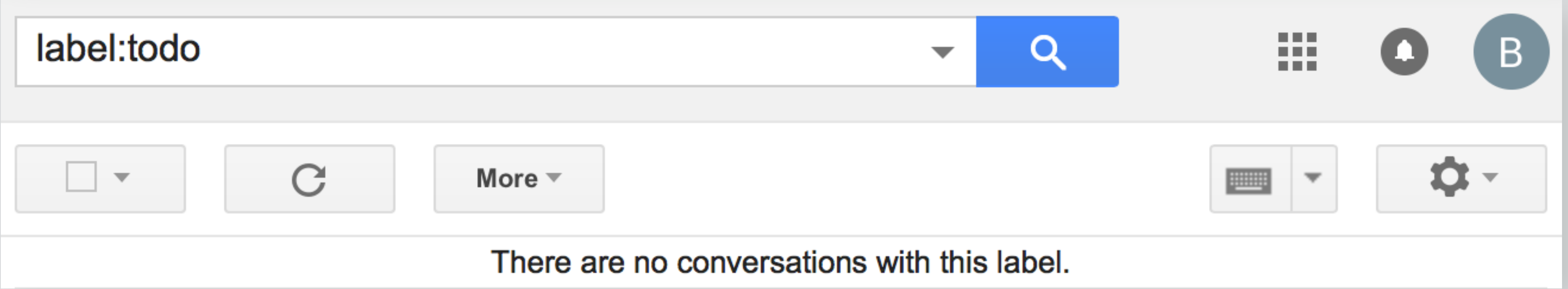
click on trash



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

me, Alyssa (13)	hacking meetups todo	javascript - Hello ag	11:48 am
Andy from Google	Updates	**Ben, welcome to your new Googl**	**9:01 am**

filter on todo label

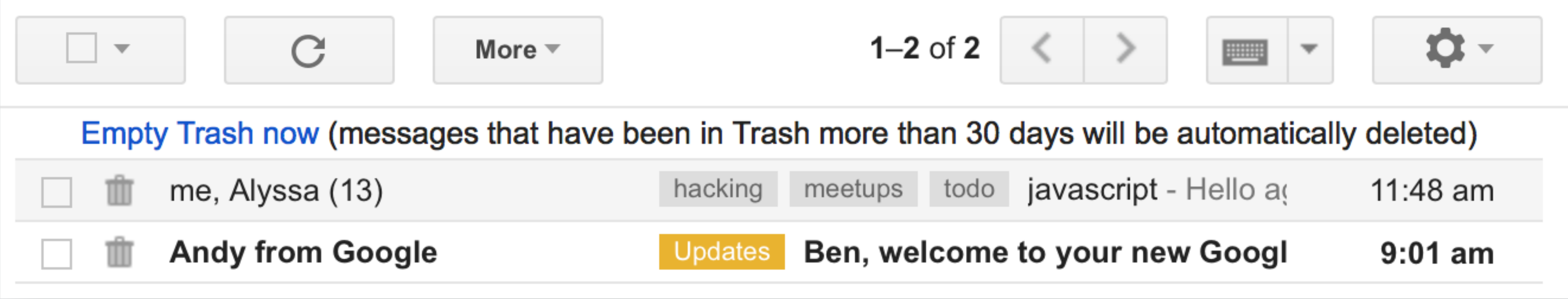


There are no conversations with this label.

filter on todo and trash

integrating these concepts is tricky

click on trash

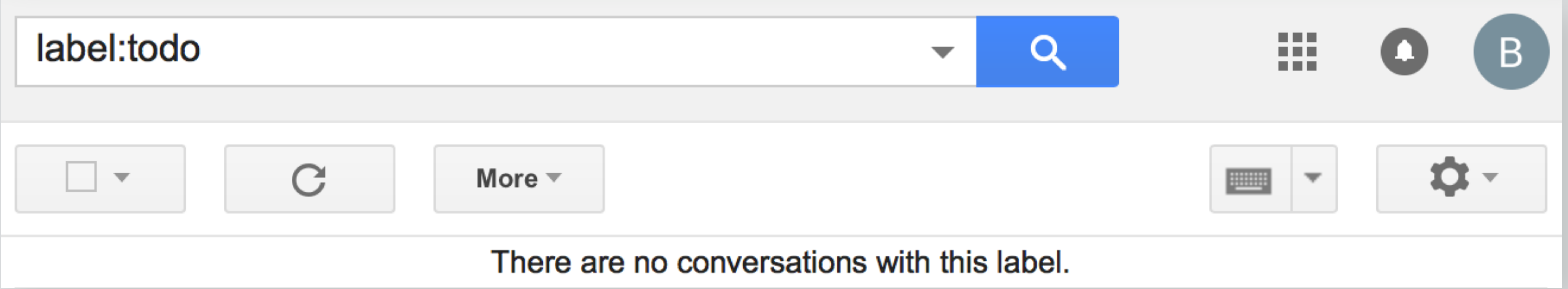


Empty Trash now (messages that have been in Trash more than 30 days will be automatically deleted)

□ me, Alyssa (13) hacking meetups todo javascript - Hello a(11:48 am

□ **Andy from Google** Updates **Ben, welcome to your new Googl** **9:01 am**

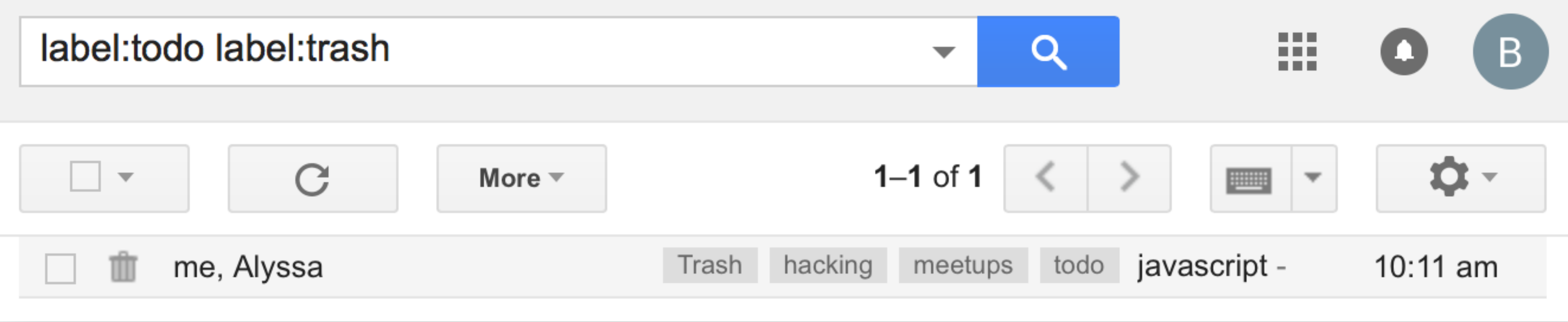
filter on todo label



label:todo

There are no conversations with this label.

filter on todo and trash

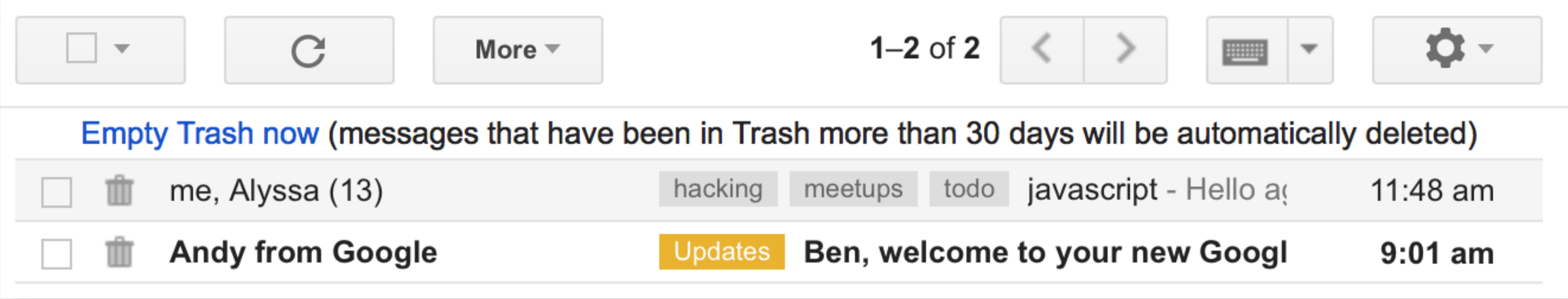


label:todo label:trash

□ me, Alyssa Trash hacking meetups todo javascript - 10:11 am

integrating these concepts is tricky

click on trash



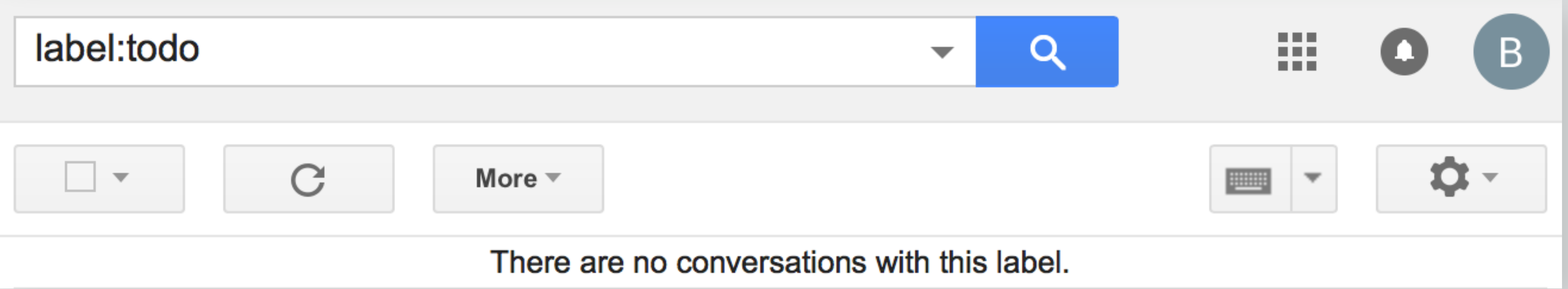
1–2 of 2

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

□ me, Alyssa (13) hacking meetups todo javascript - Hello a 11:48 am

□ **Andy from Google** Updates **Ben, welcome to your new Googl** **9:01 am**

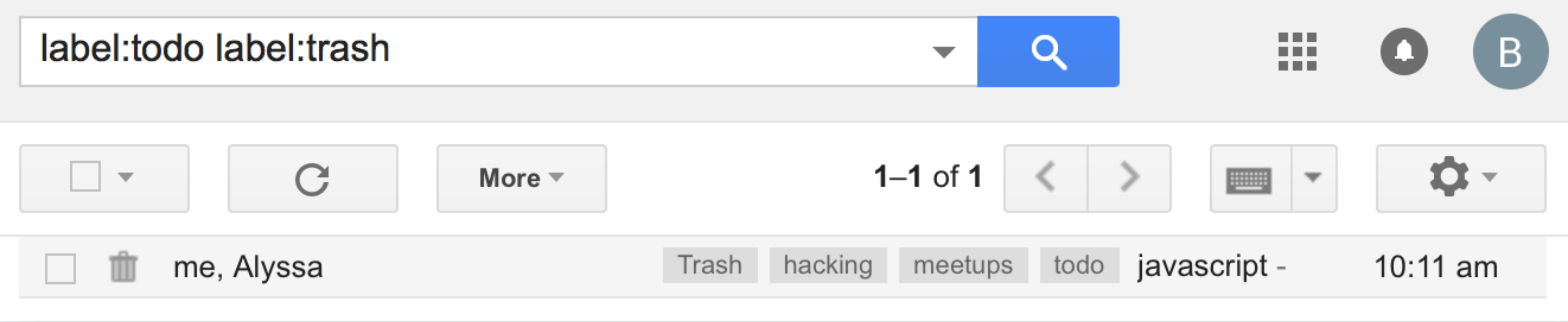
filter on todo label



label:todo

There are no conversations with this label.

filter on todo and trash

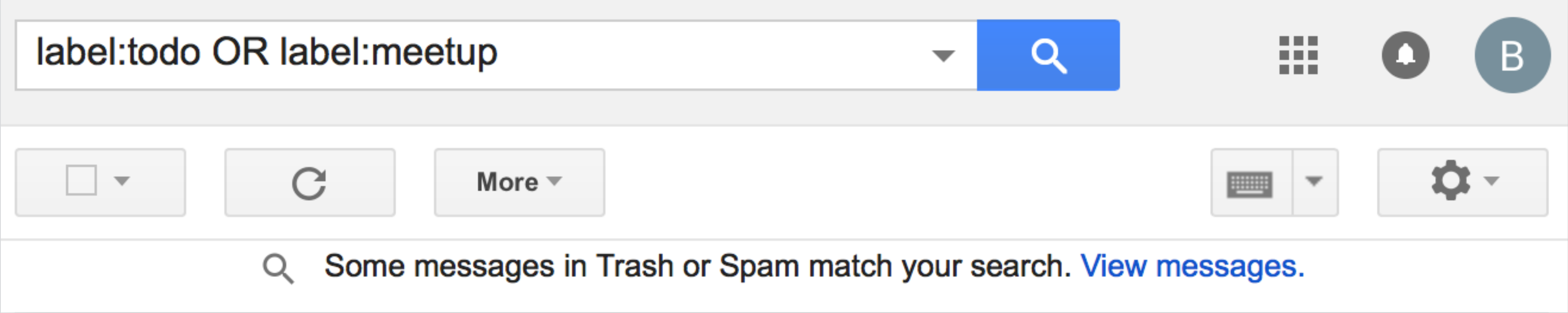


label:todo label:trash

1–1 of 1

□ me, Alyssa Trash hacking meetups todo javascript - 10:11 am

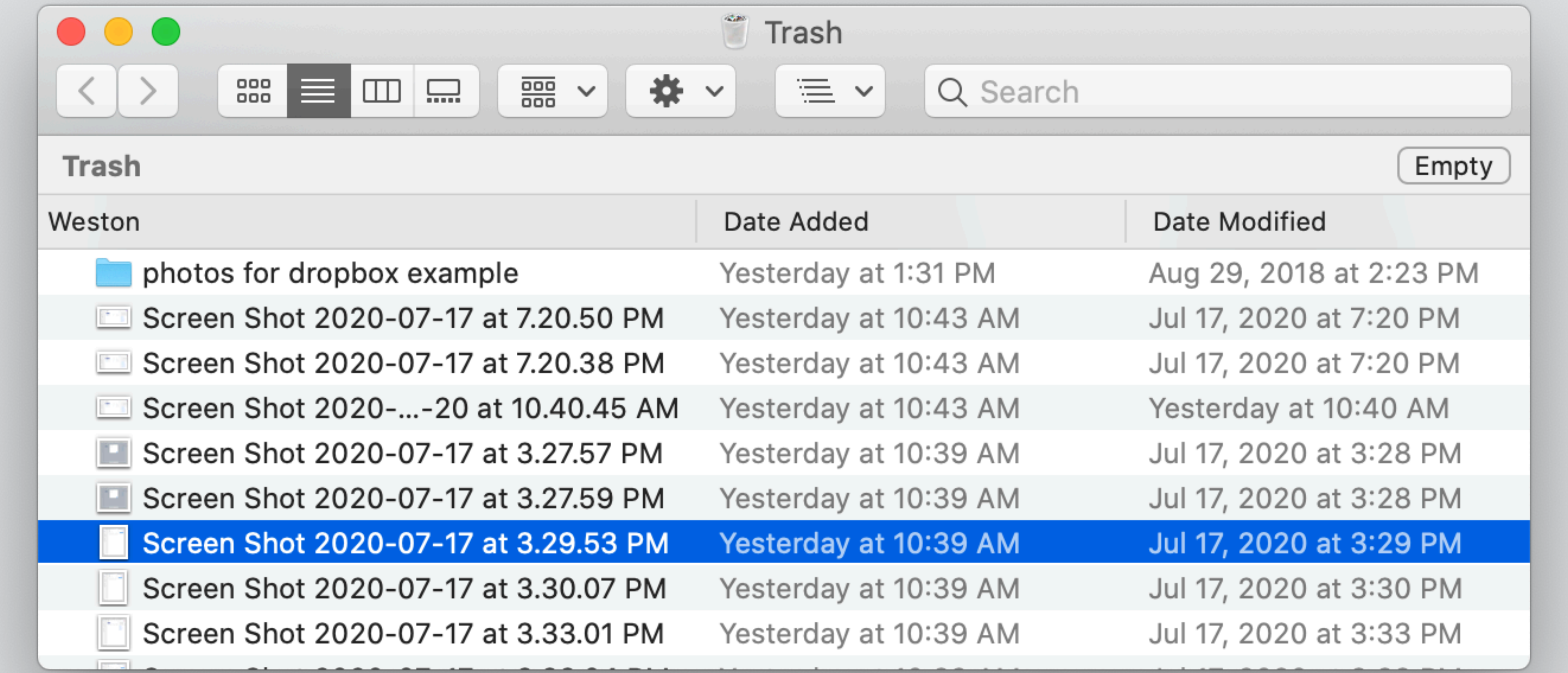
filter on something else



label:todo OR label:meetup

Some messages in Trash or Spam match your search. View messages.

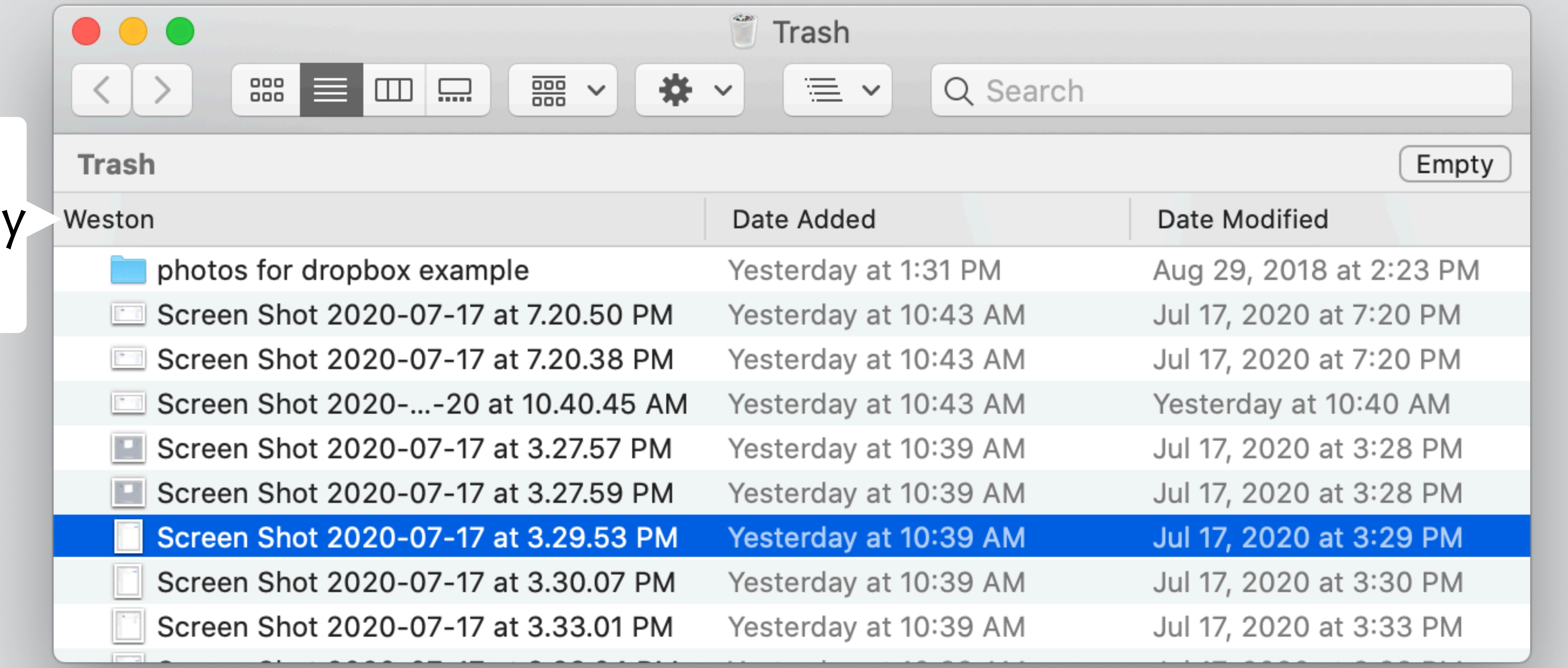
a beautiful (but tricky) synergy



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

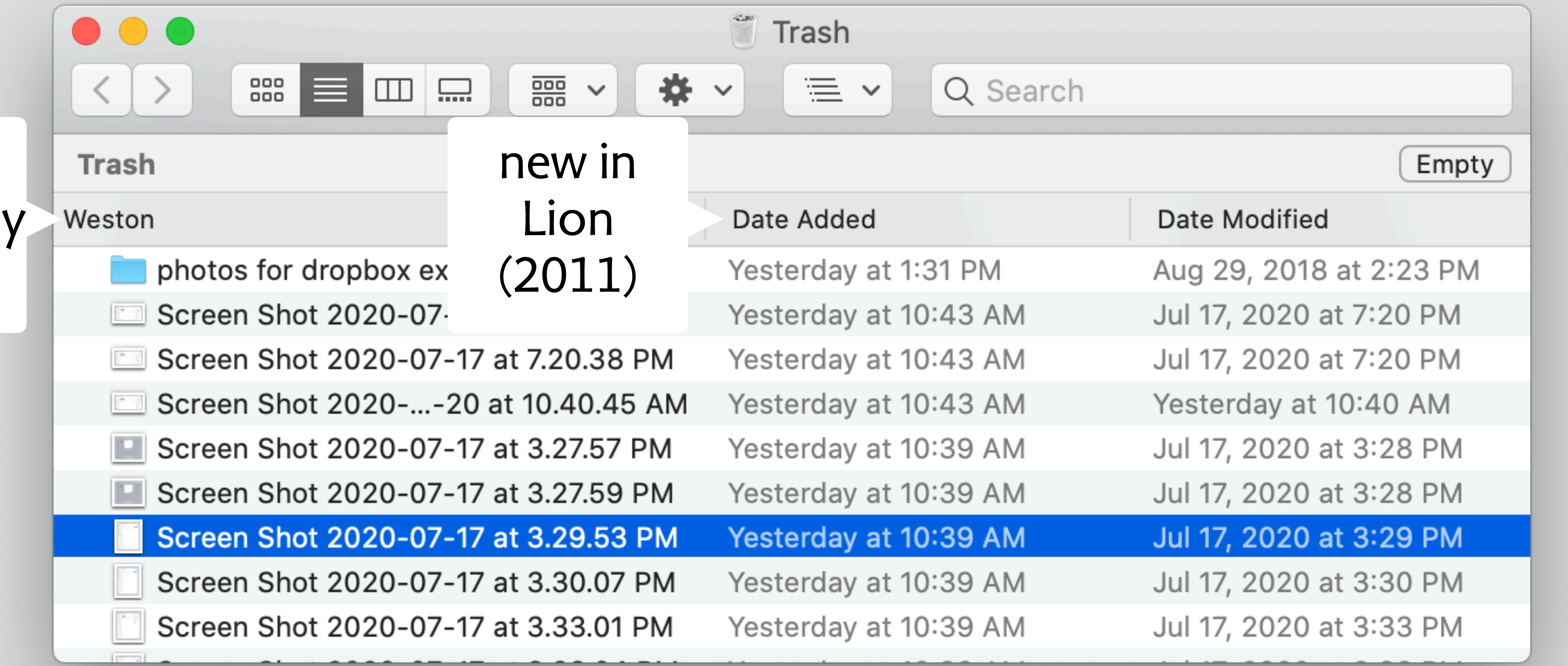
a beautiful (but tricky) synergy

folder sortable by volume!



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



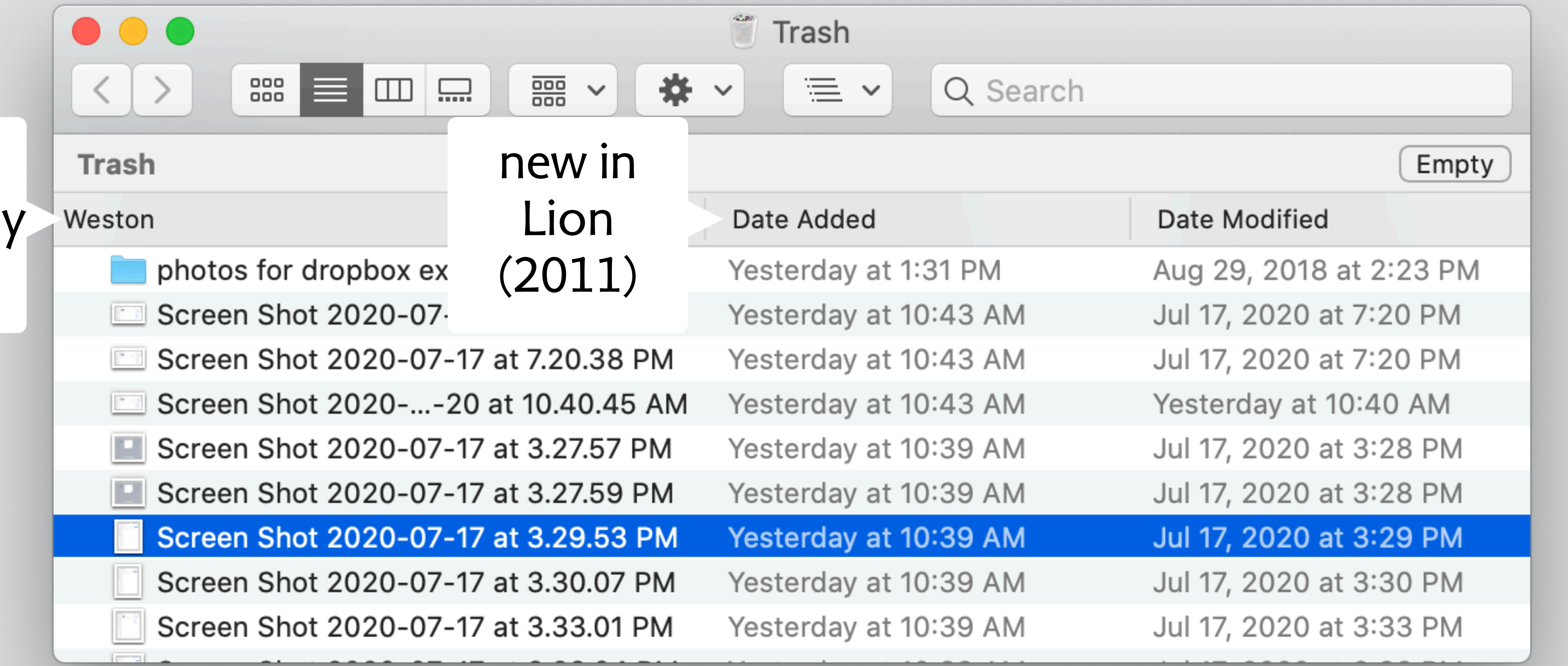
Weston	Date Added	Date Modified
photos for dropbox ex	Yesterday at 1:31 PM	Aug 29, 2018 at 2:23 PM
Screen Shot 2020-07-	Yesterday at 10:43 AM	Jul 17, 2020 at 7:20 PM
Screen Shot 2020-07-17 at 7.20.38 PM	Yesterday at 10:43 AM	Jul 17, 2020 at 7:20 PM
Screen Shot 2020-...-20 at 10.40.45 AM	Yesterday at 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

folder sortable by volume!

new in Lion (2011)

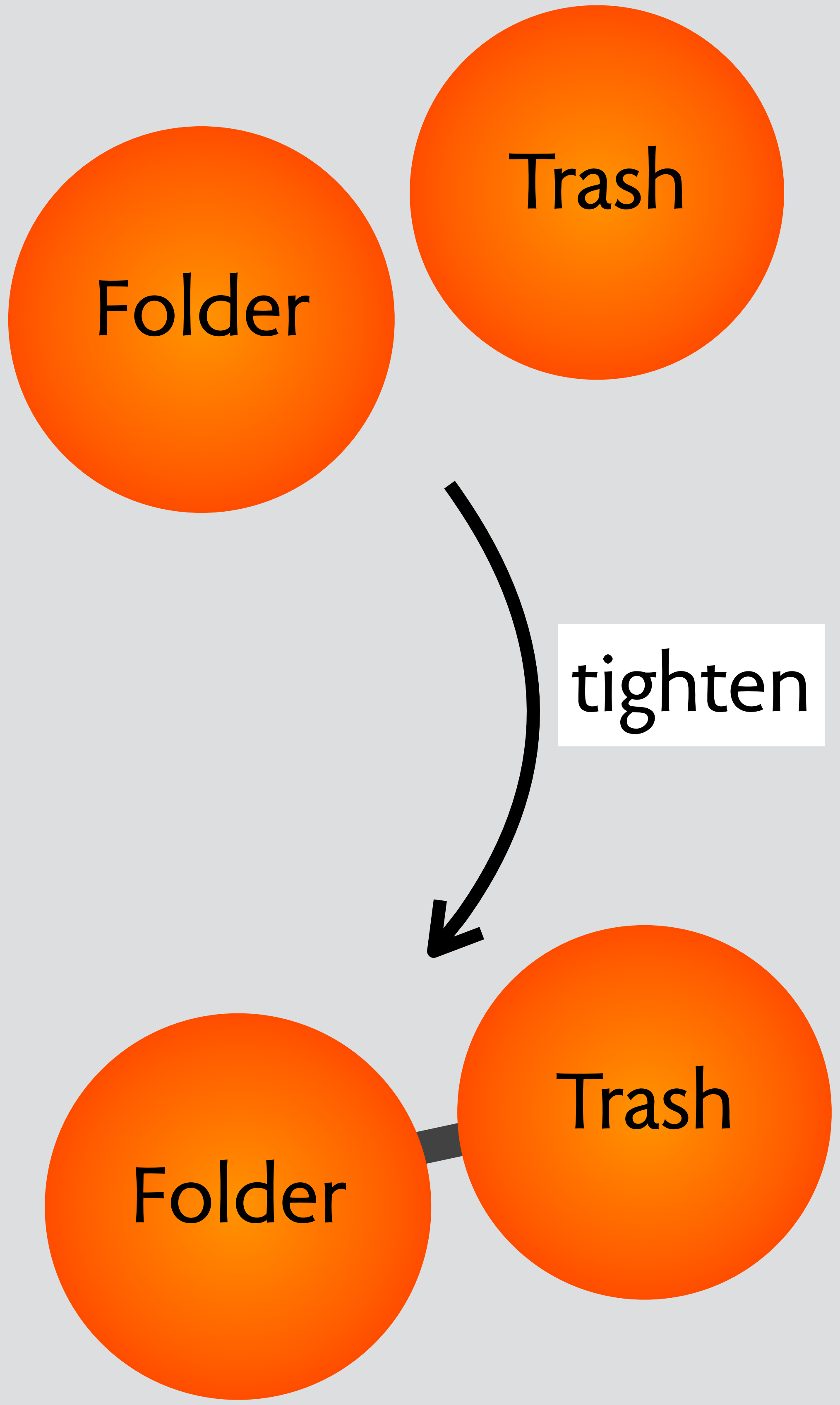
a beautiful (but tricky) synergy

folder sortable by volume!



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)

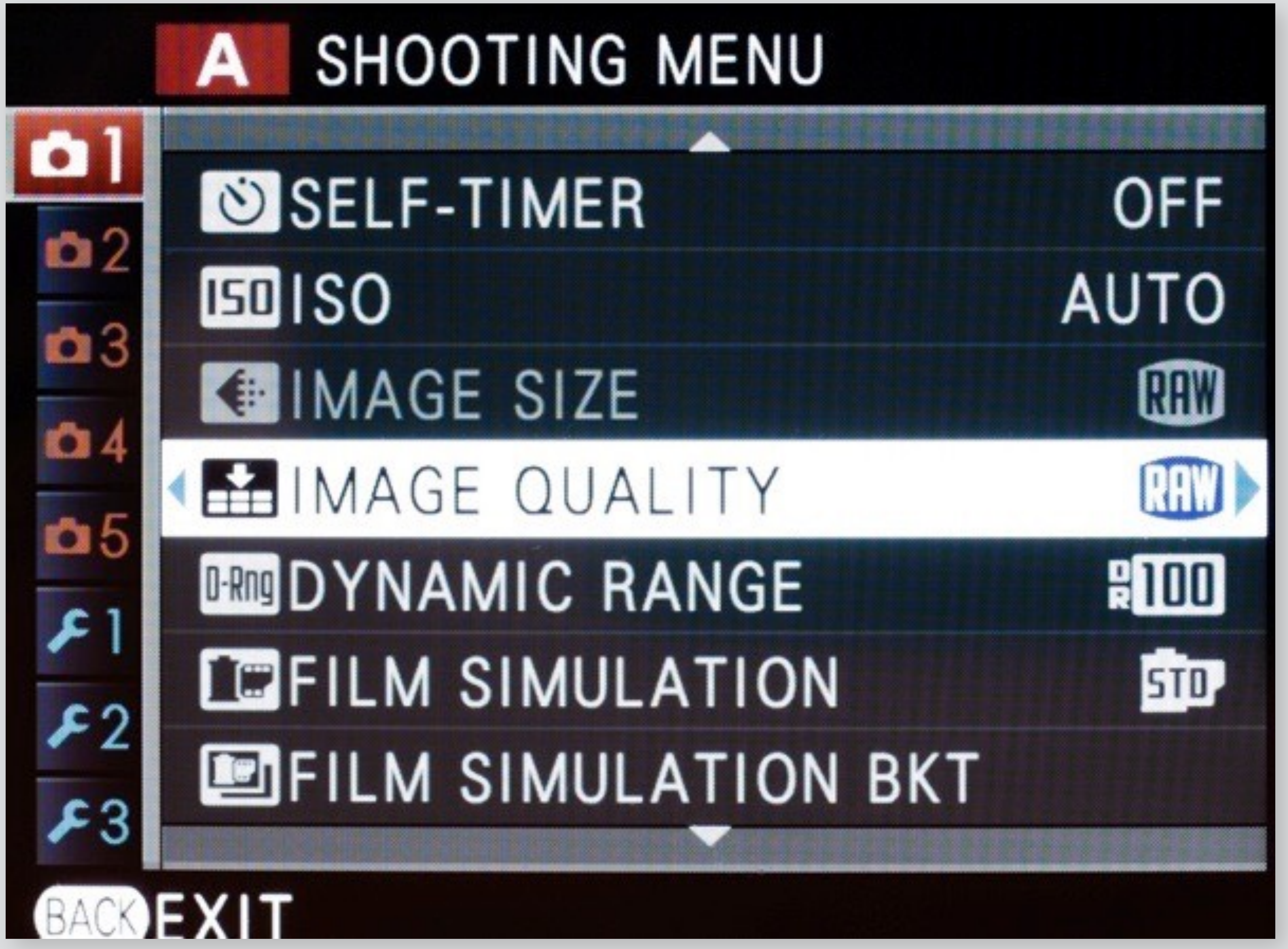


design moves
in response to
problems

aspect ratio in fujifilm cameras

a lovely camera fuji x100

complex menu system: image quality setting



A SHOOTING MENU

SELF-TIMER OFF

ISO AUTO

IMAGE SIZE RAW

IMAGE QUALITY RAW

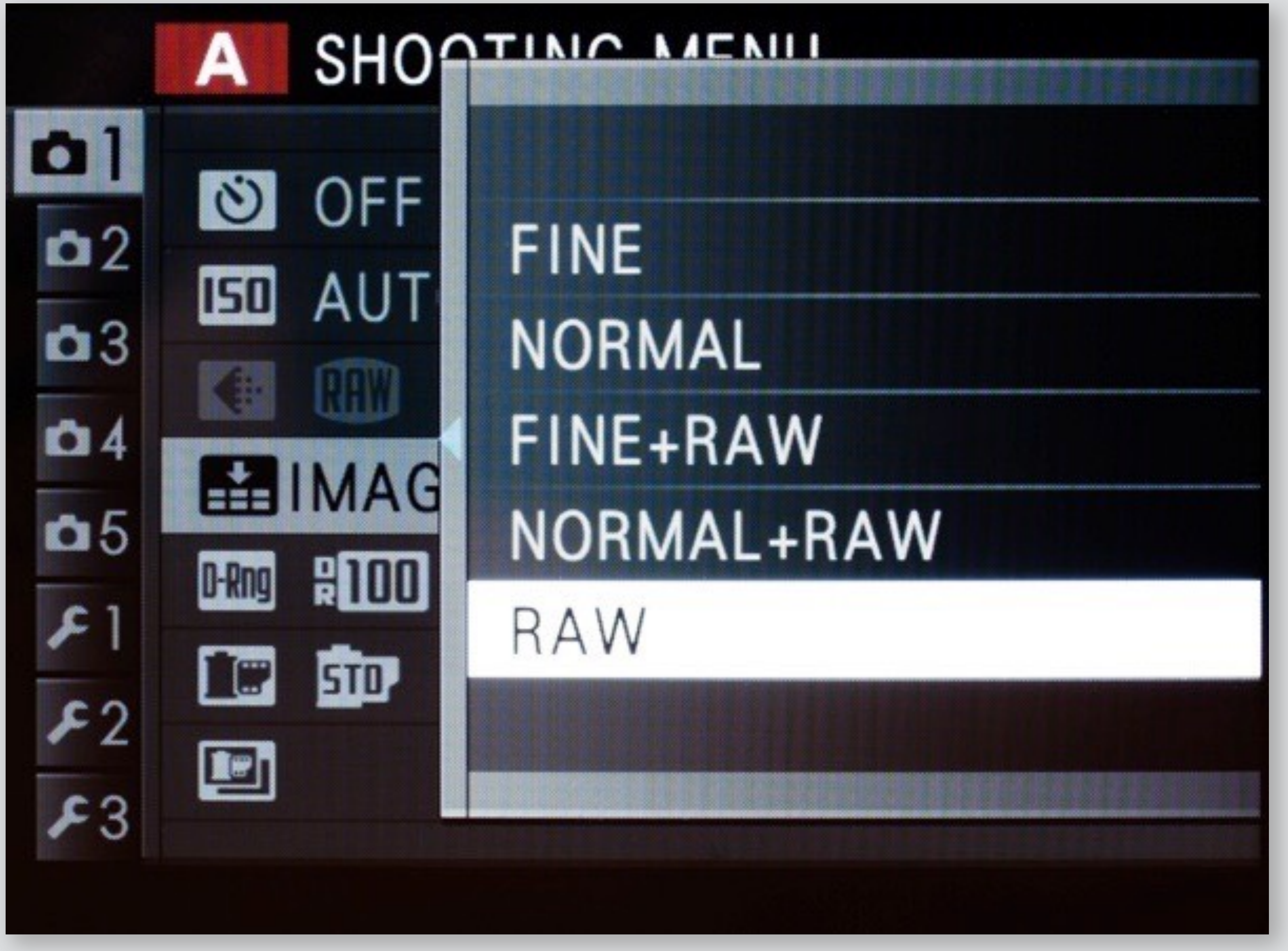
DYNAMIC RANGE DR100

FILM SIMULATION STD

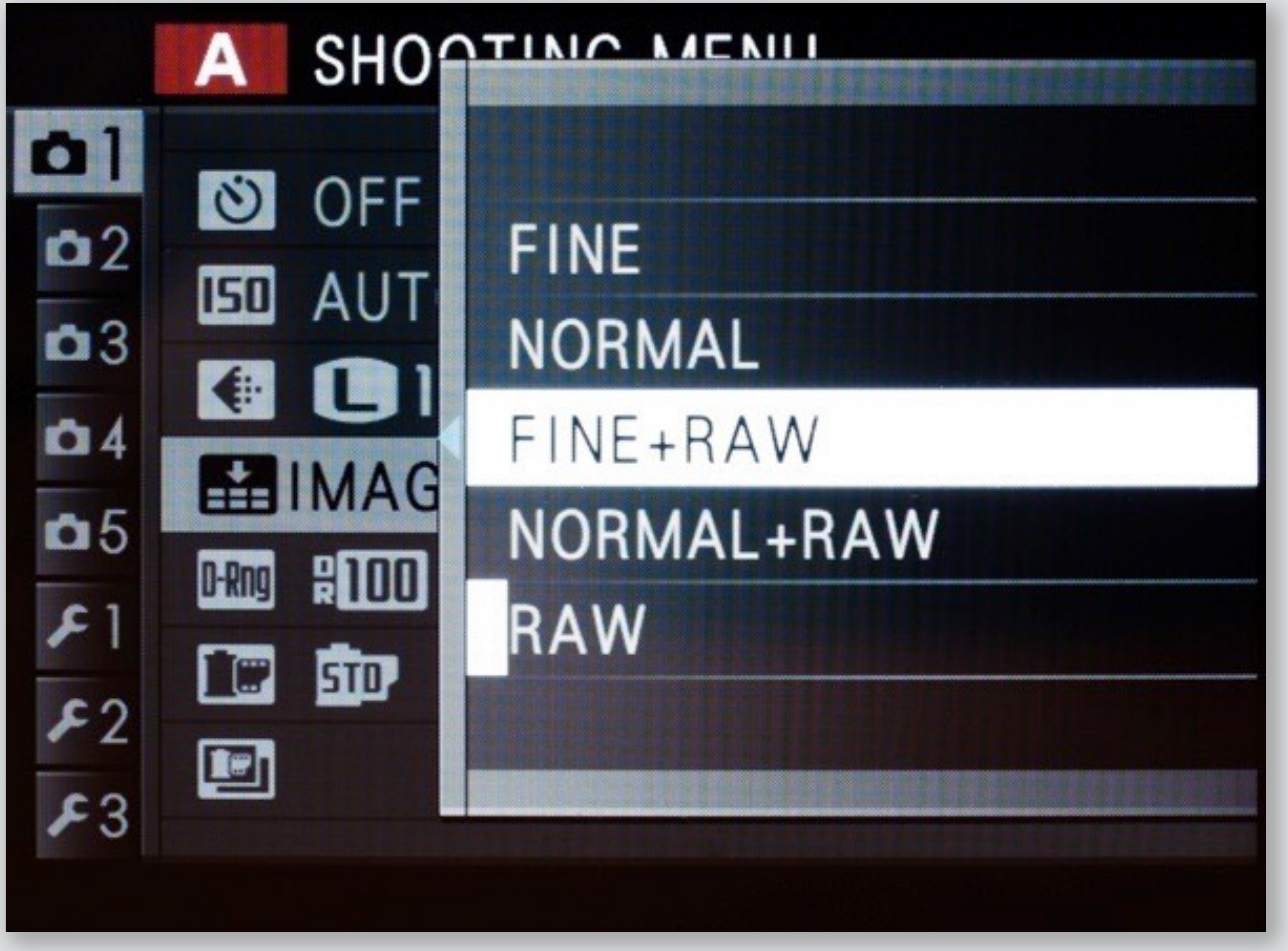
FILM SIMULATION BKT

BACK EXIT

complex menu system: image quality setting



complex menu system: image quality setting



aspect ratio

aspect ratio

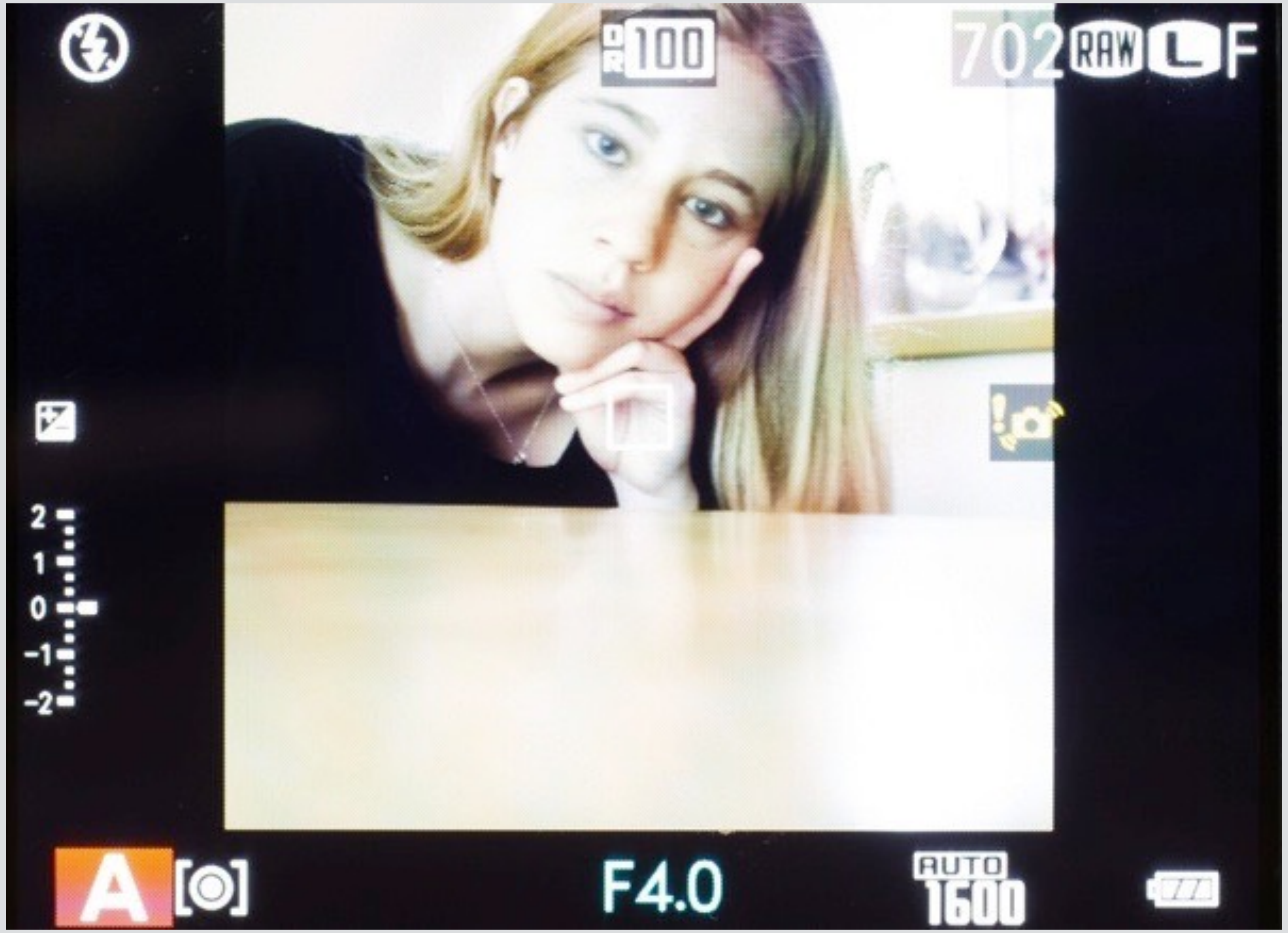
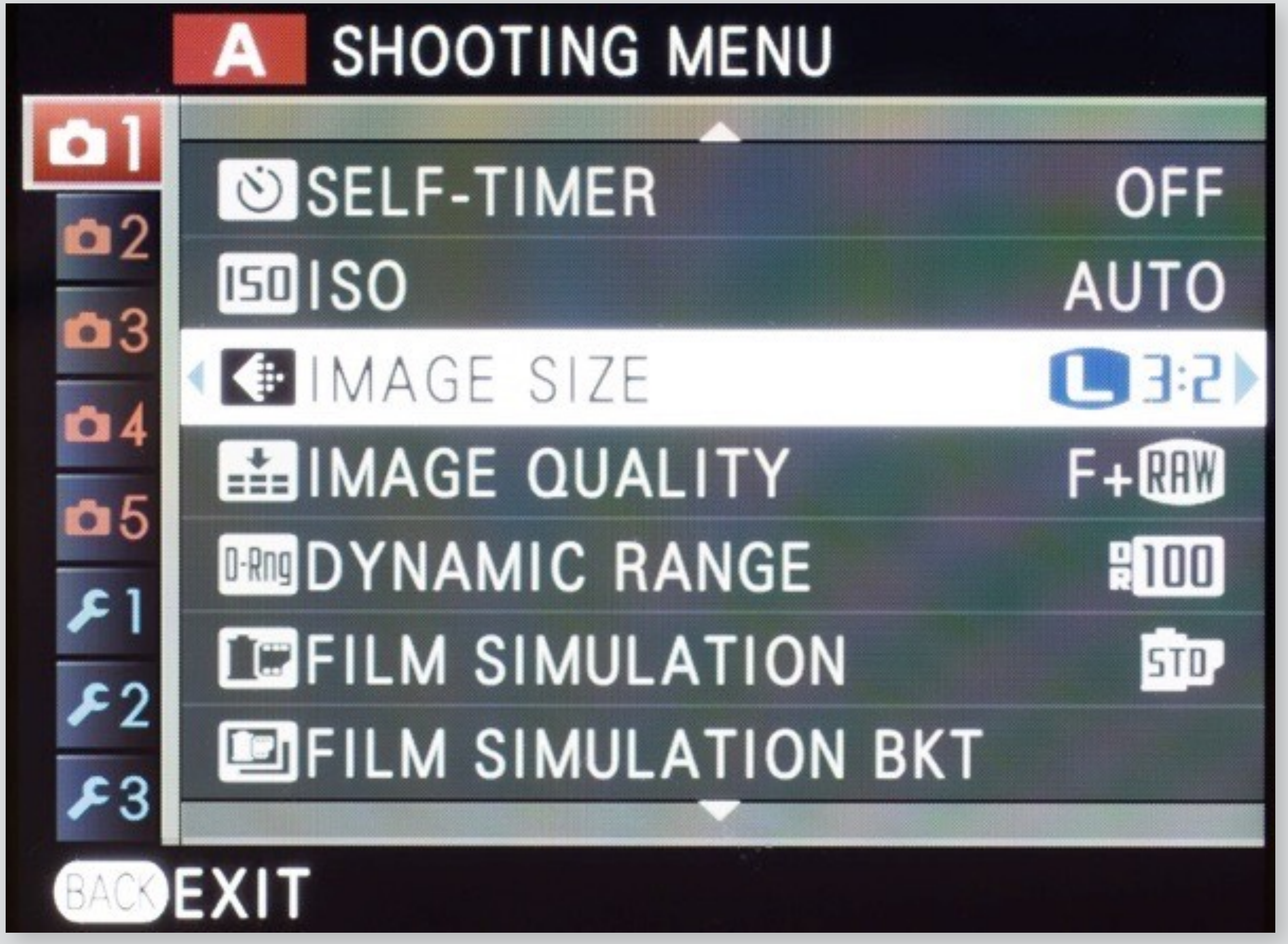


image size setting



A SHOOTING MENU

SELF-TIMER OFF

ISO AUTO

IMAGE SIZE L 3:2

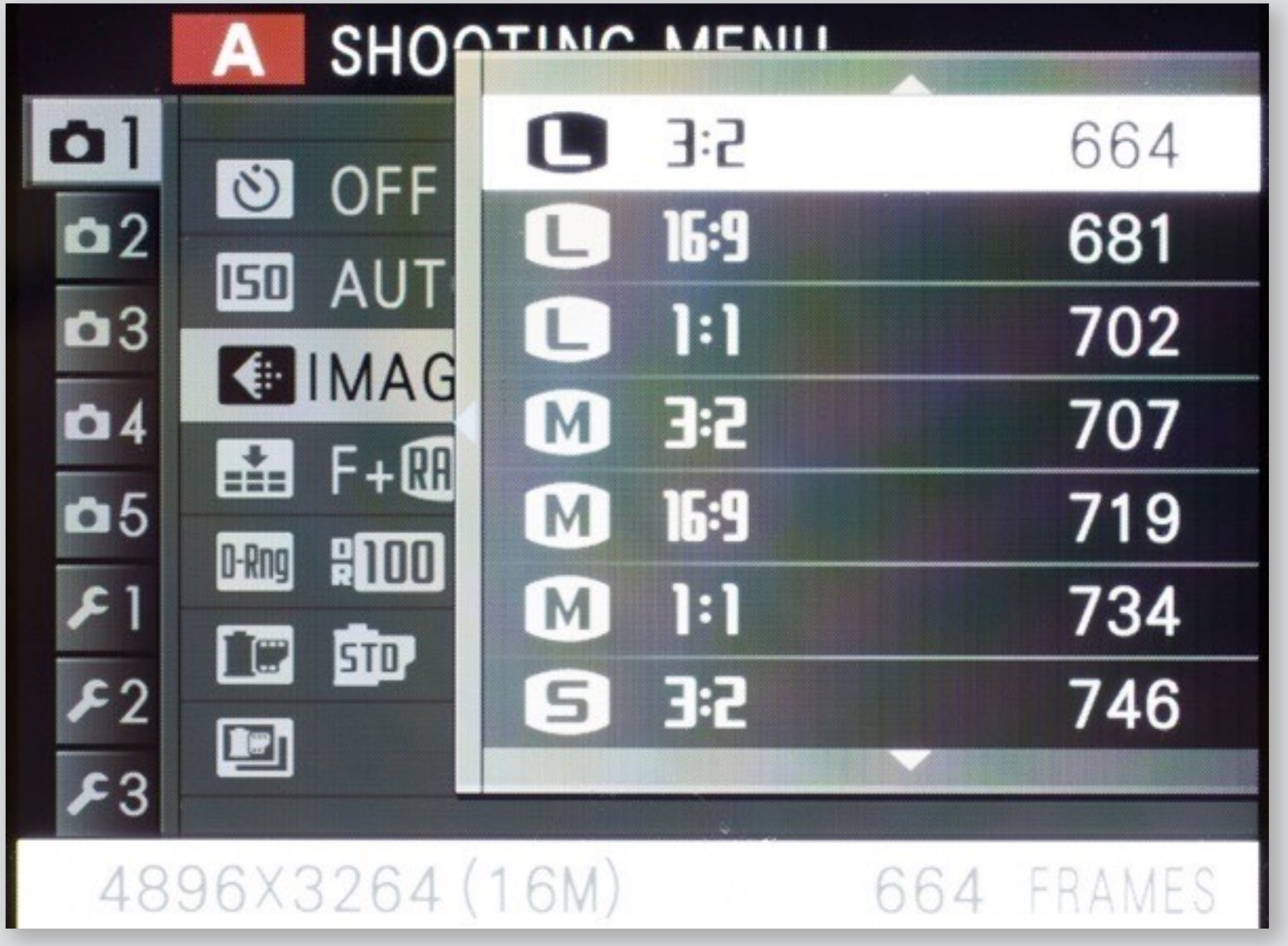
IMAGE QUALITY F+RAW

DYNAMIC RANGE DR100

FILM SIMULATION STD

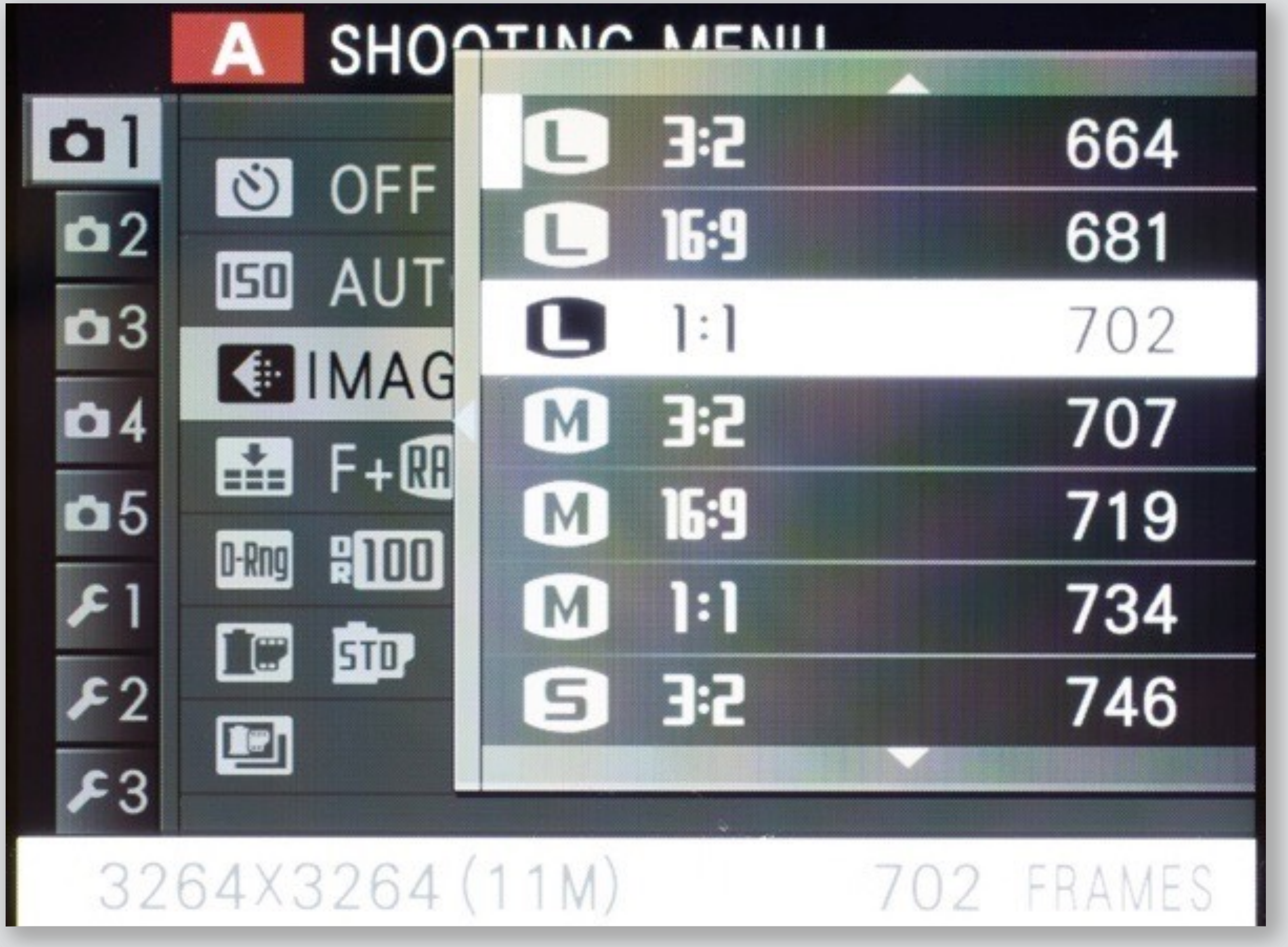
FILM SIMULATION BKT

image size setting



4896X3264 (16M) 664 FRAMES

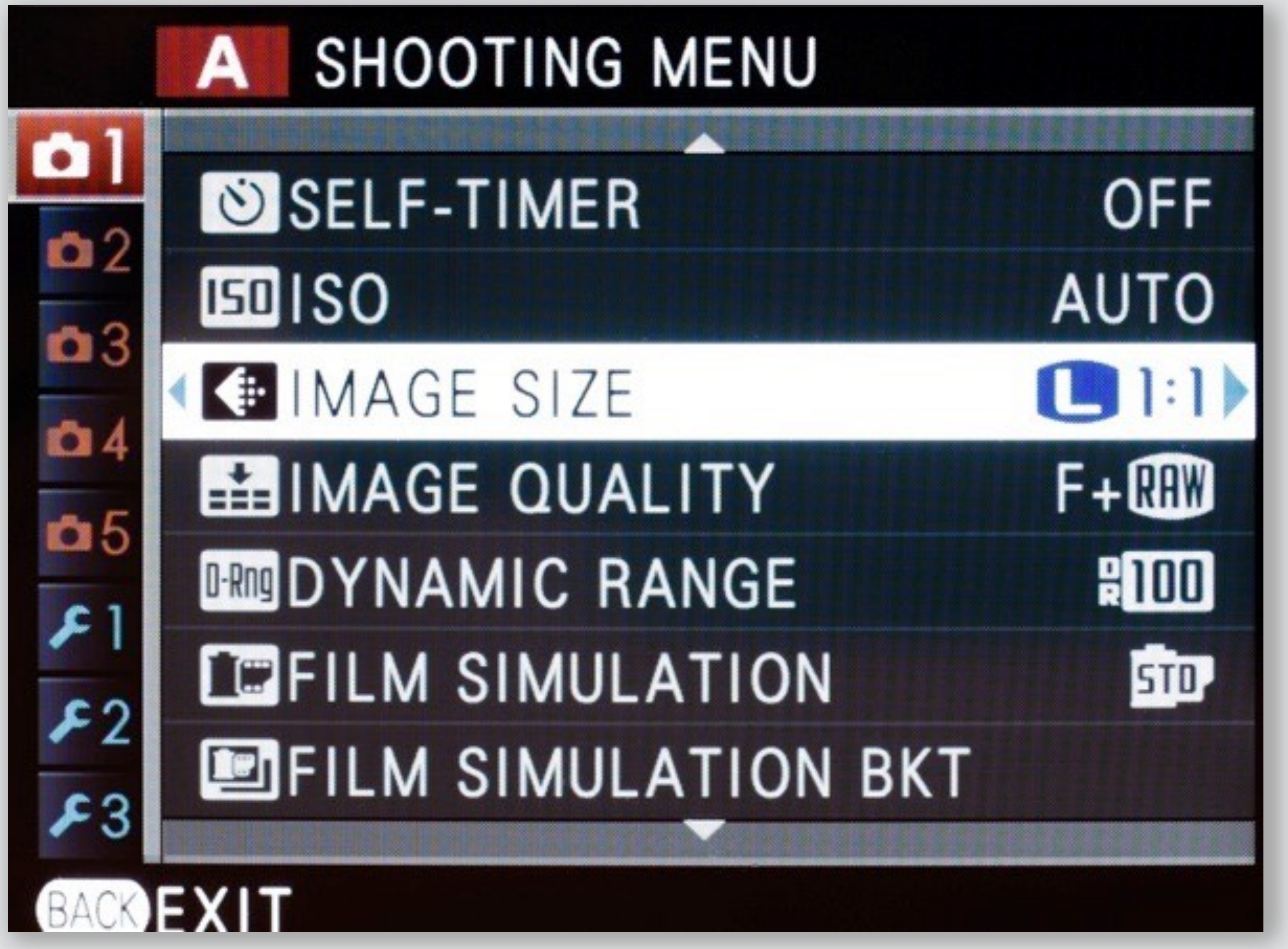
image size setting



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

3264X3264 (11M) 702 FRAMES

image size setting

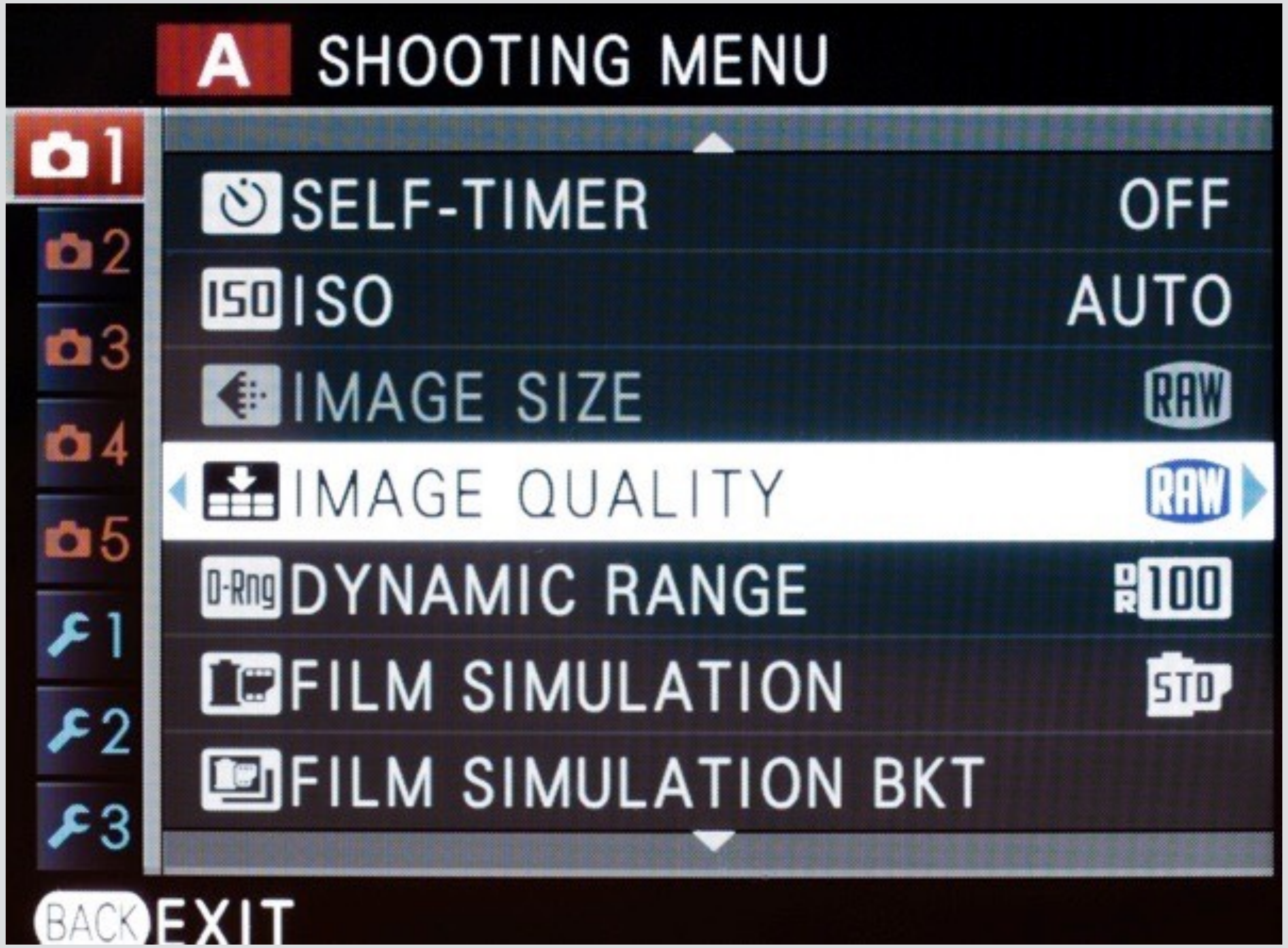


A SHOOTING MENU

SELF-TIMER	OFF
ISO	AUTO
IMAGE SIZE	L 1:1
IMAGE QUALITY	F+RAW
DYNAMIC RANGE	DR100
FILM SIMULATION	STD
FILM SIMULATION BKT	

BACK EXIT

non-standard ratio + raw?



A SHOOTING MENU

SELF-TIMER OFF

ISO AUTO

IMAGE SIZE RAW

IMAGE QUALITY RAW

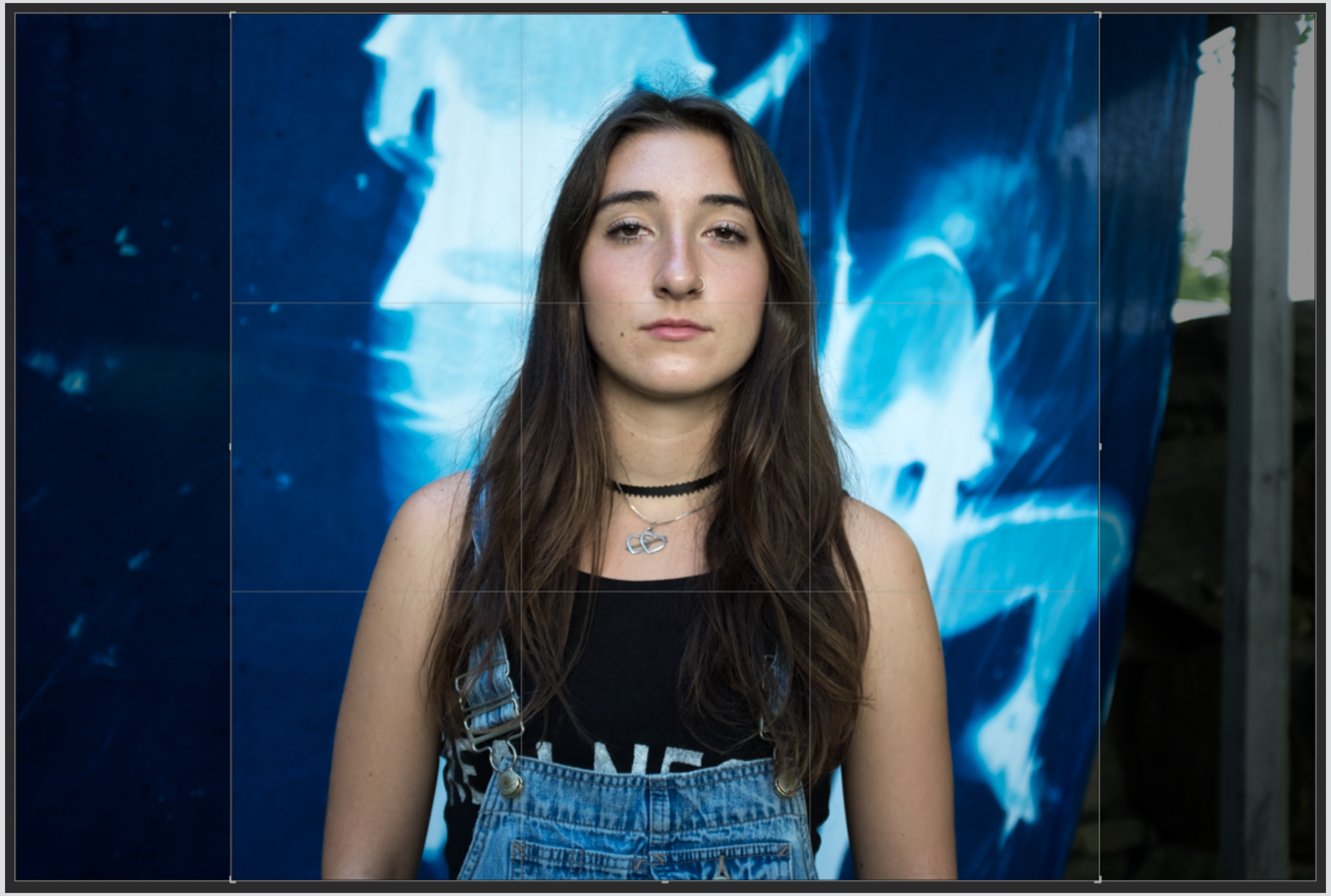
DYNAMIC RANGE DR100

FILM SIMULATION STD

FILM SIMULATION BKT

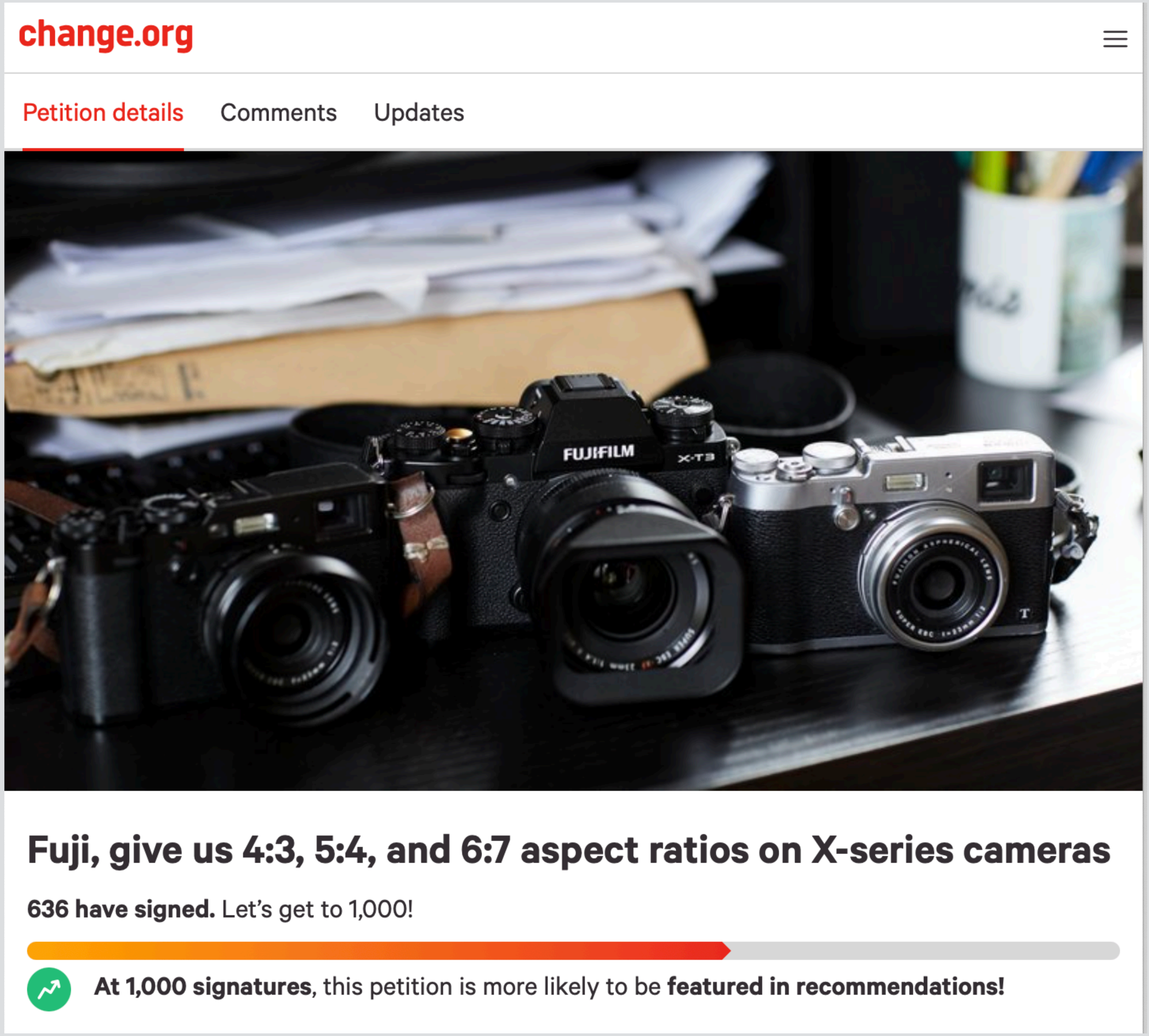
BACK EXIT

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



raw image showing non-destructive aspect ratio crop

problem #2: very few ratio options

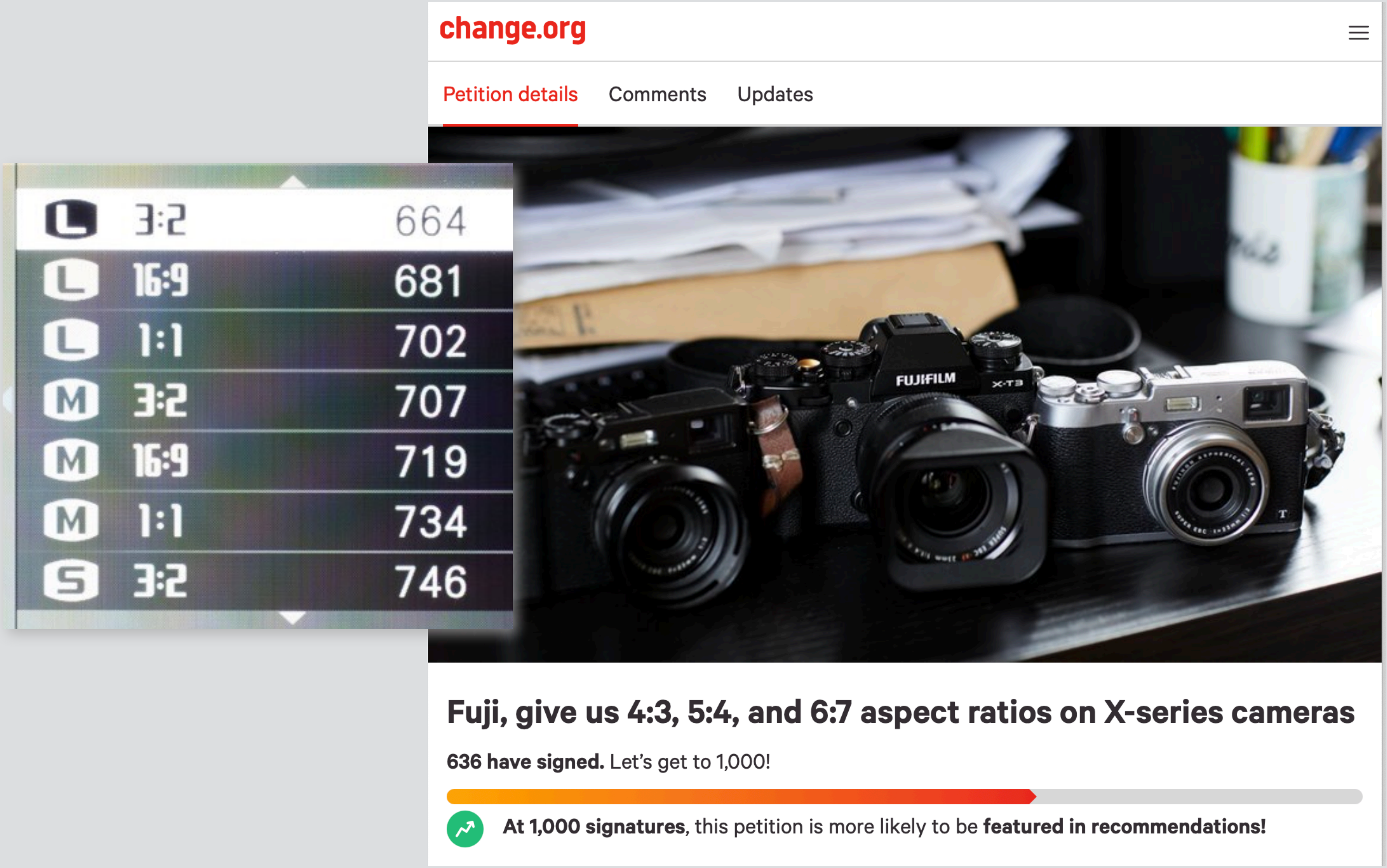


Fuji, give us 4:3, 5:4, and 6:7 aspect ratios on X-series cameras

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



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

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!**

diagnosis?

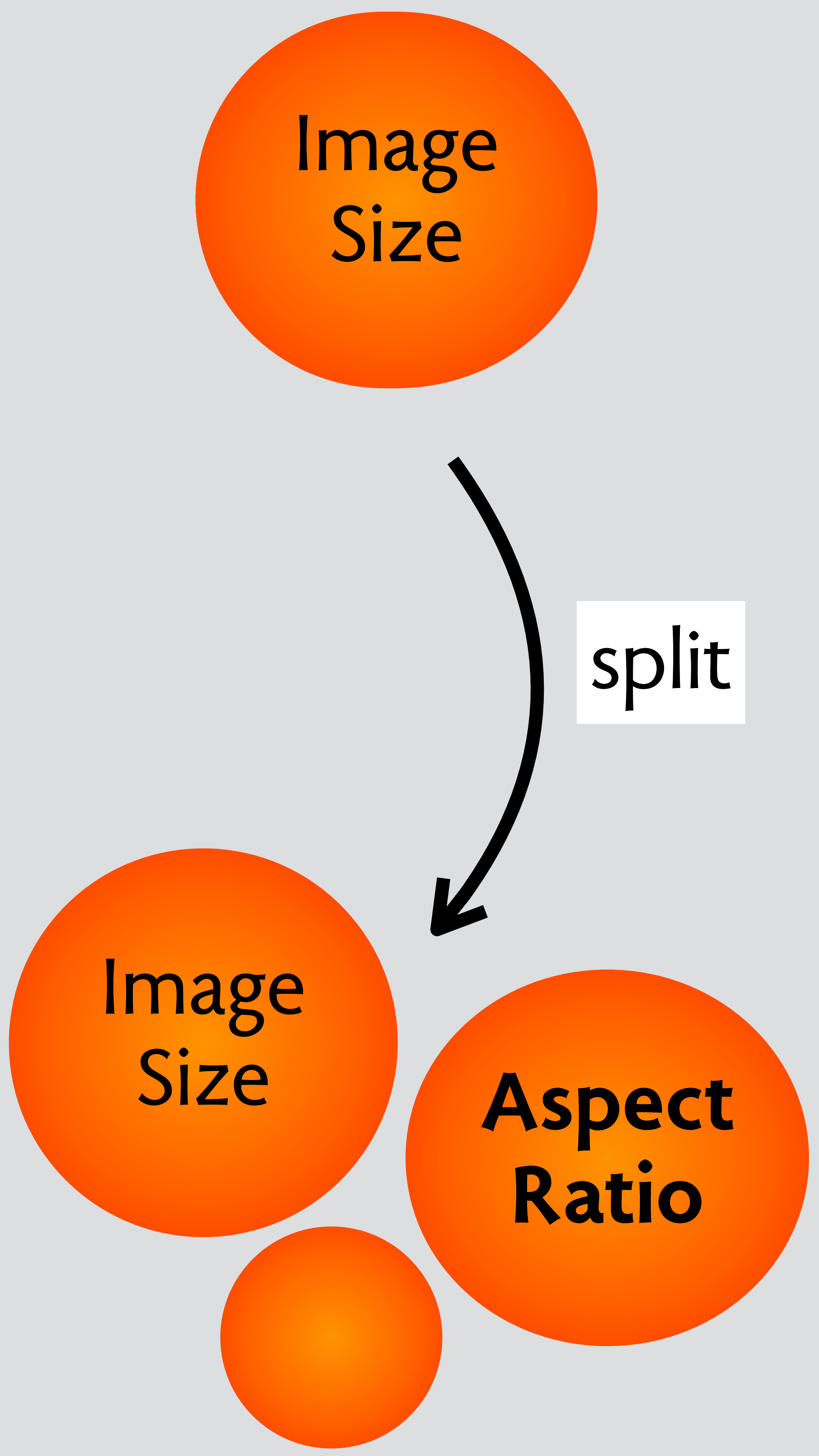
aspect ratio is not a concept
merged into JPEG image size concept
so cannot be controlled independently
I call this ”overloading by piggybacking”

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

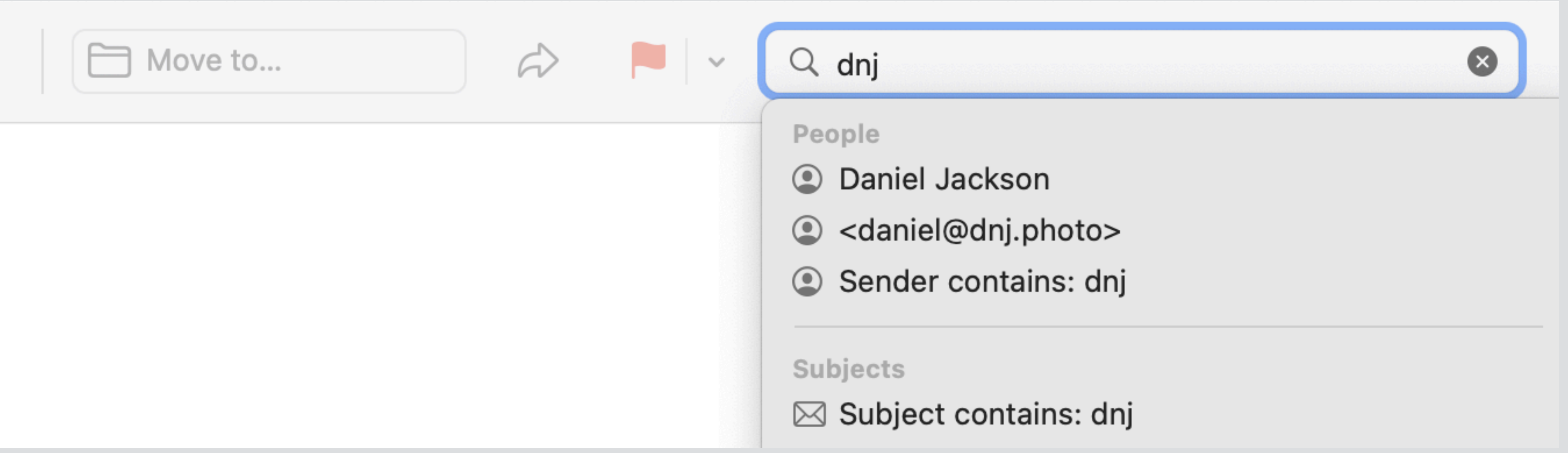


message filters

in apple mail

how many ways to filter messages?

how many ways to filter messages?



dnj

People

Daniel Jackson

<daniel@dnj.photo>

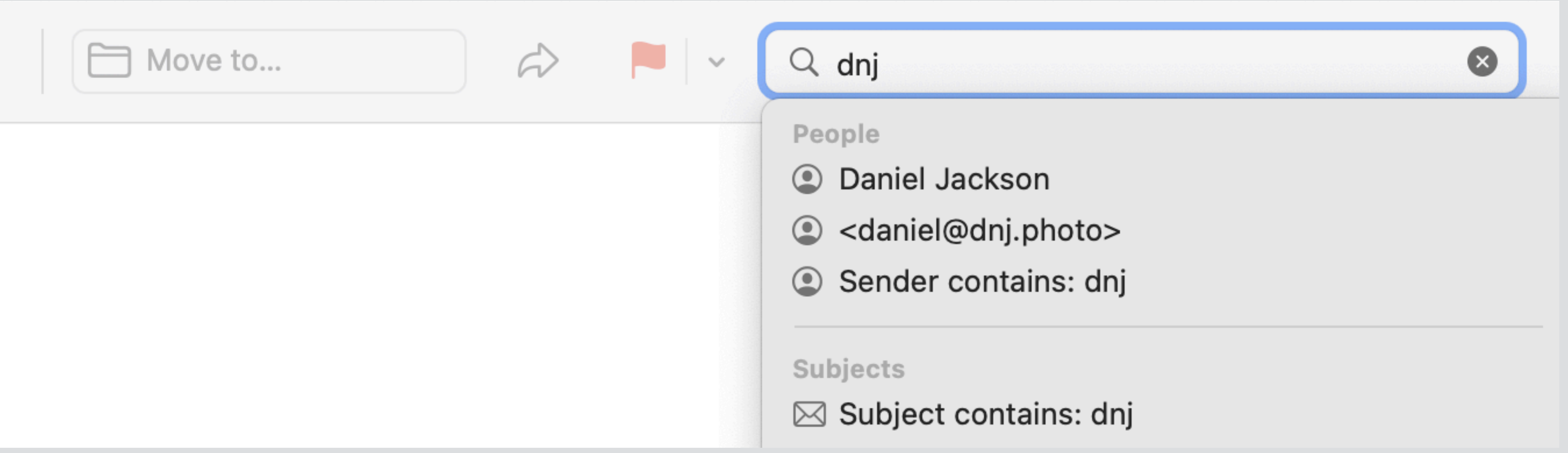
Sender contains: dnj

Subjects

Subject contains: dnj

search for a message

how many ways to filter messages?



dnj

People

Daniel Jackson

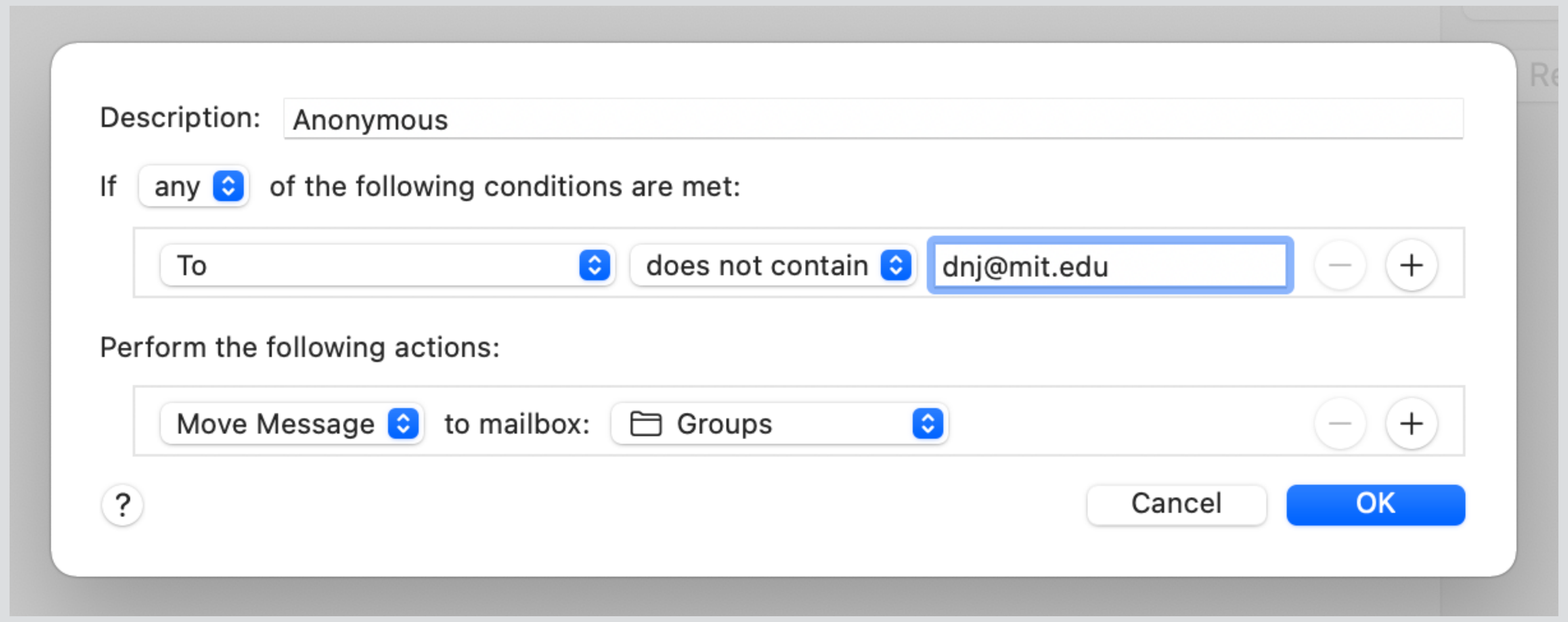
<daniel@dnj.photo>

Sender contains: dnj

Subjects

Subject contains: dnj

search for a message


Description: Anonymous

If any of the following conditions are met:

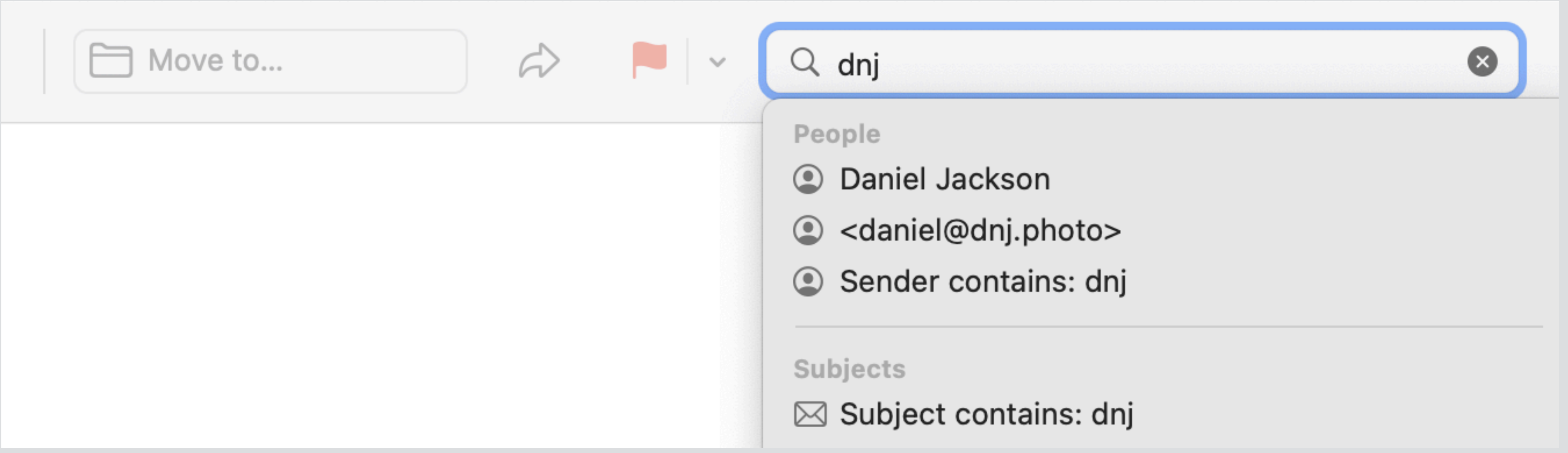
To does not contain dnj@mit.edu

Perform the following actions:

Move Message to mailbox: Groups

create a rule

how many ways to filter messages?



dnj

People

Daniel Jackson

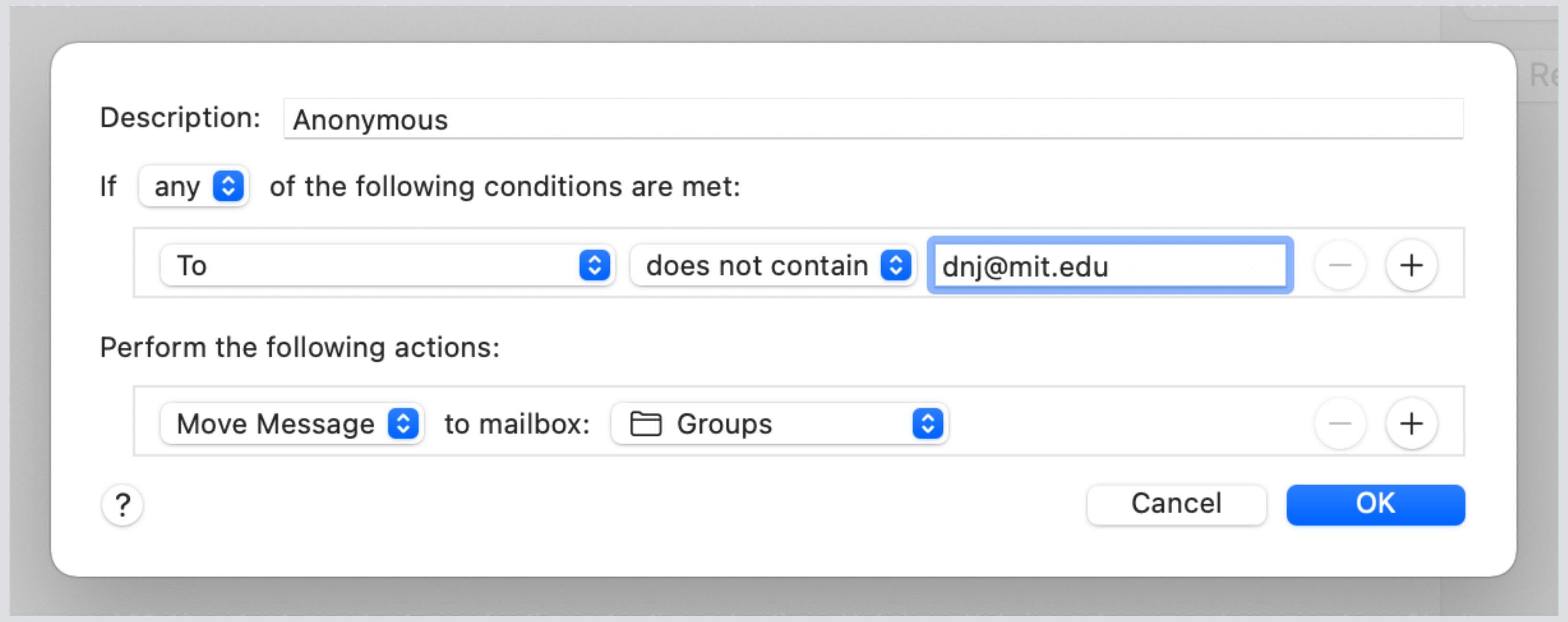
<daniel@dnj.photo>

Sender contains: dnj

Subjects

Subject contains: dnj

search for a message



Description: Anonymous

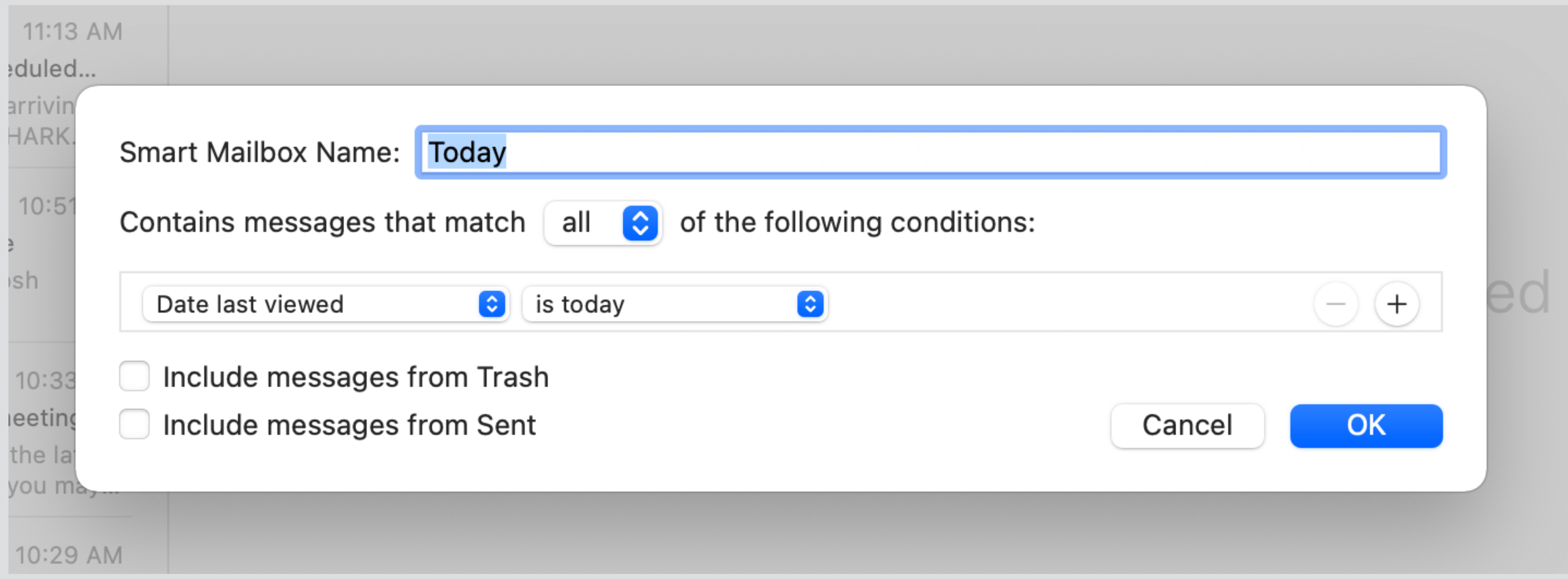
If any of the following conditions are met:

To does not contain dnj@mit.edu

Perform the following actions:

Move Message to mailbox: Groups

create a rule



Smart Mailbox Name: Today

Contains messages that match all of the following conditions:

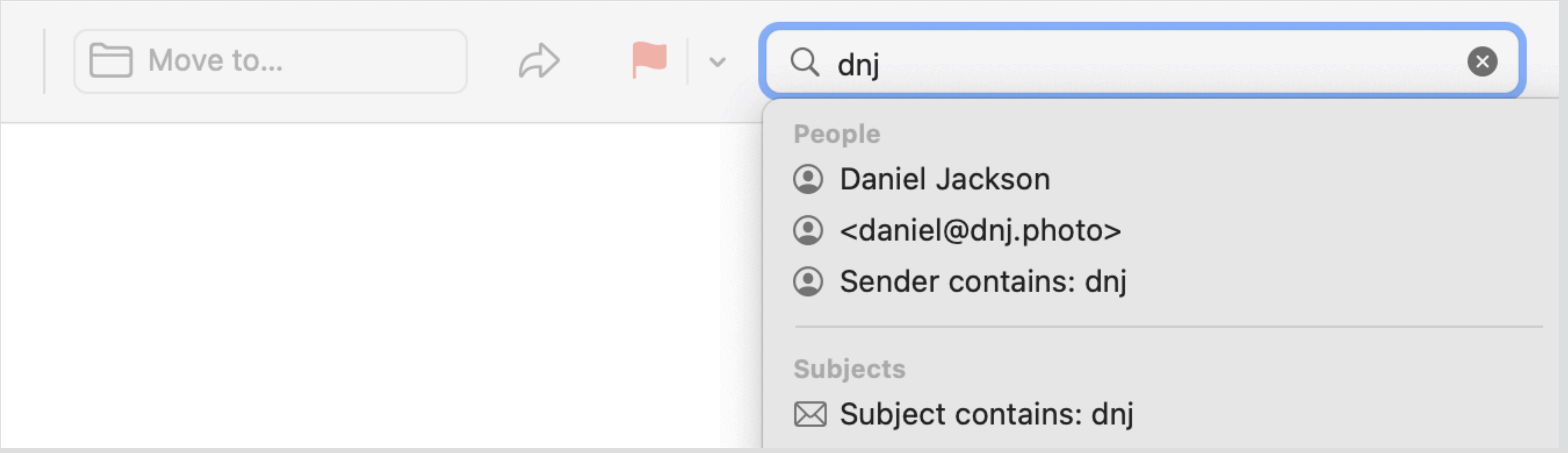
Date last viewed is today

□ Include messages from Trash

□ Include messages from Sent

define a smart folder

how many ways to filter messages?

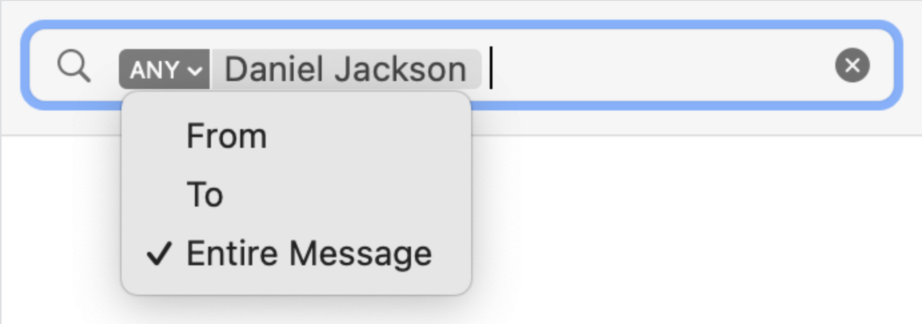


dnj

People
- Daniel Jackson
- <daniel@dnj.photo>
- Sender contains: dnj

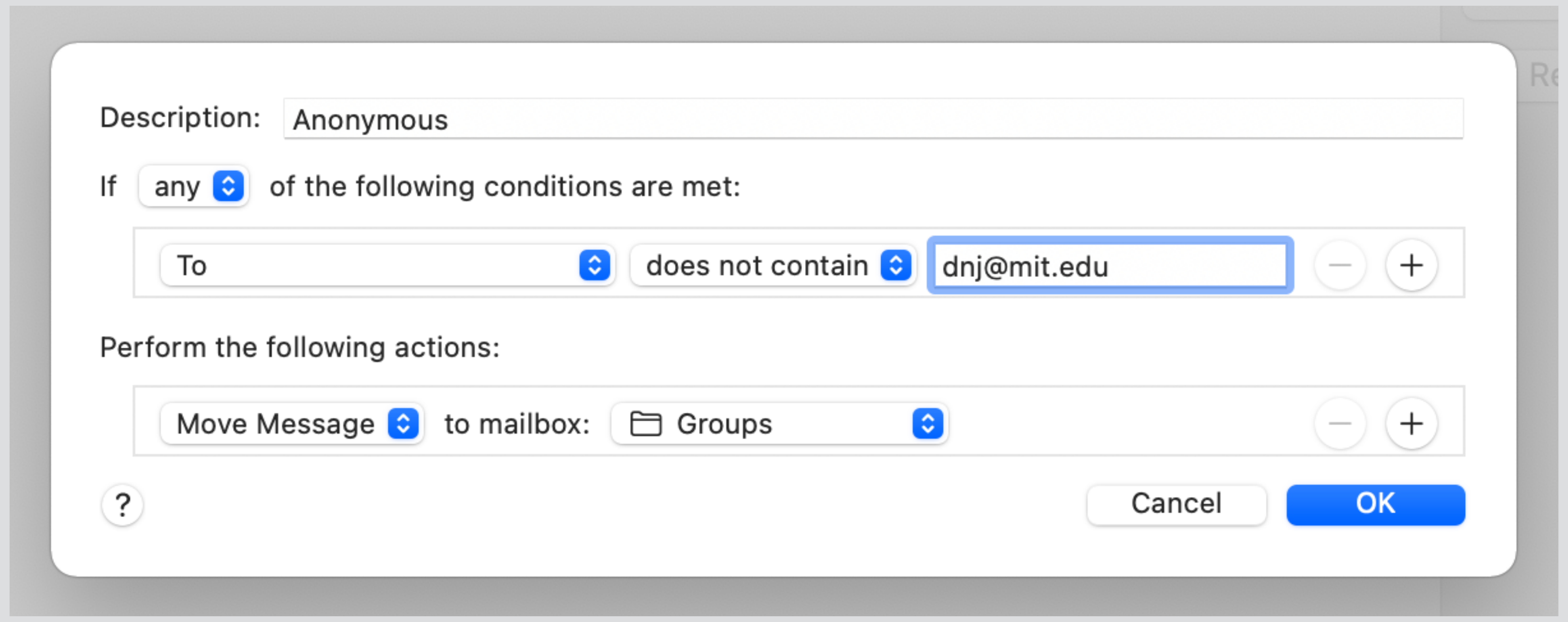
Subjects
- Subject contains: dnj

search for a message


ANY Daniel Jackson

- From
- To
- ✓ Entire Message

search options


Description: Anonymous

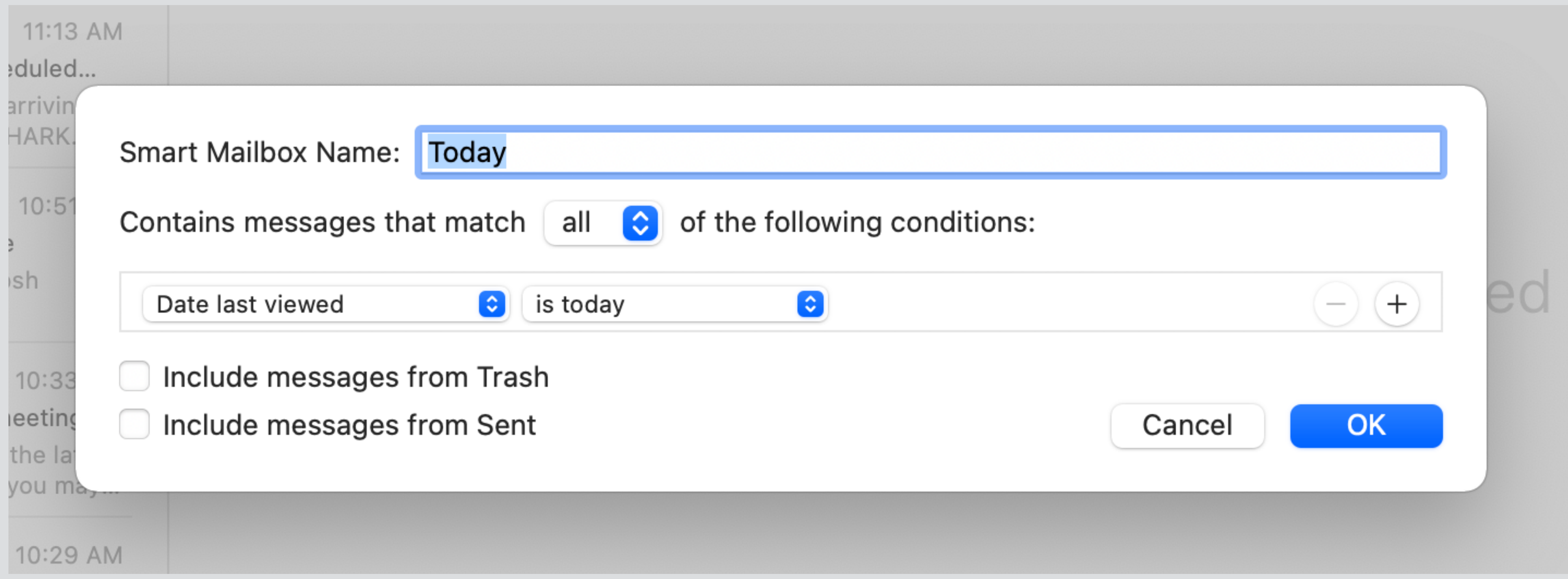
If any of the following conditions are met:

To | does not contain | dnj@mit.edu

Perform the following actions:

Move Message to mailbox: Groups

create a rule


Smart Mailbox Name: Today

Contains messages that match all of the following conditions:

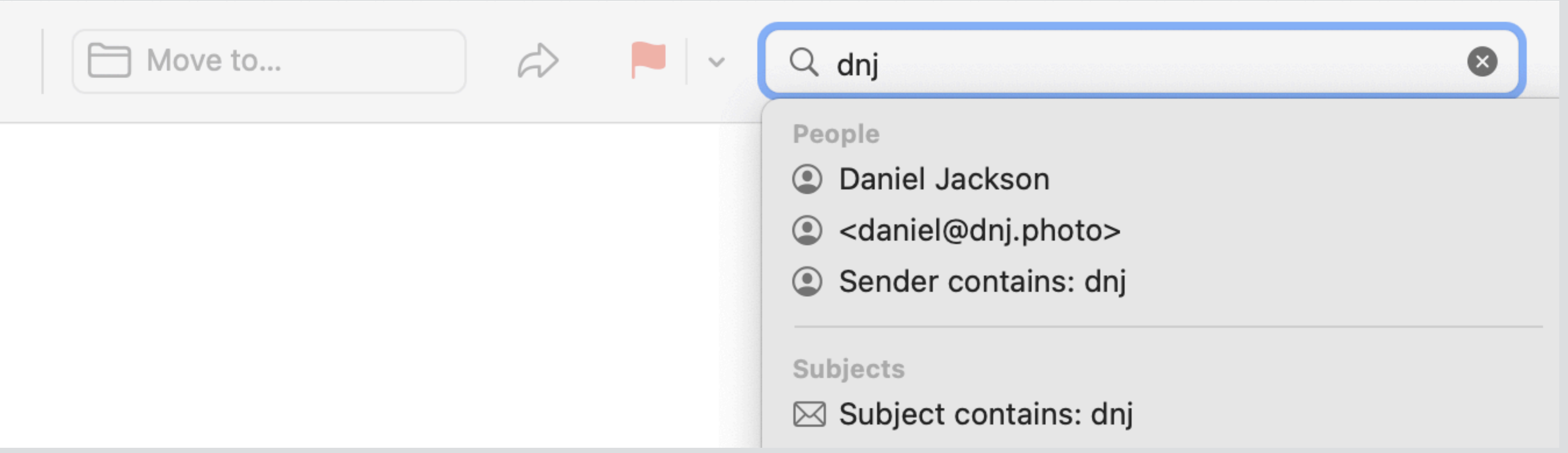
Date last viewed | is today

□ Include messages from Trash

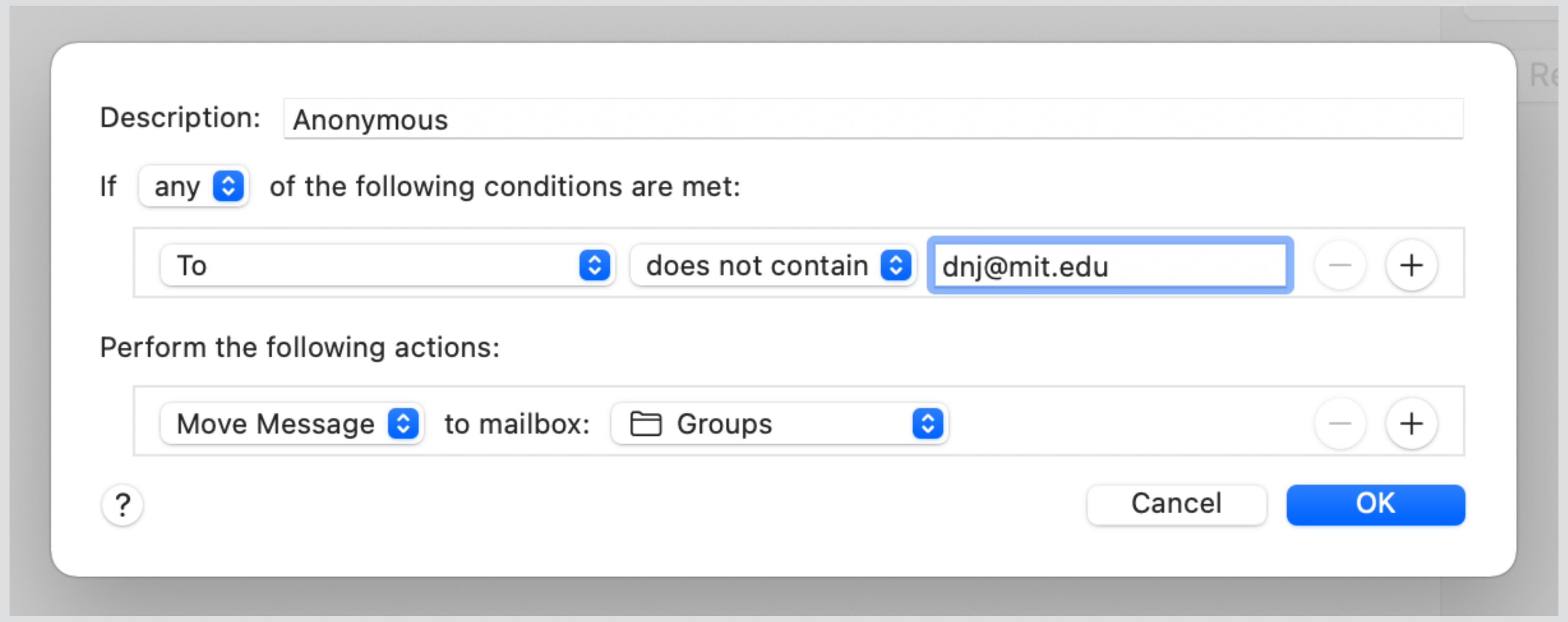
□ Include messages from Sent

define a smart folder

how many ways to filter messages?



search for a message



Description: Anonymous

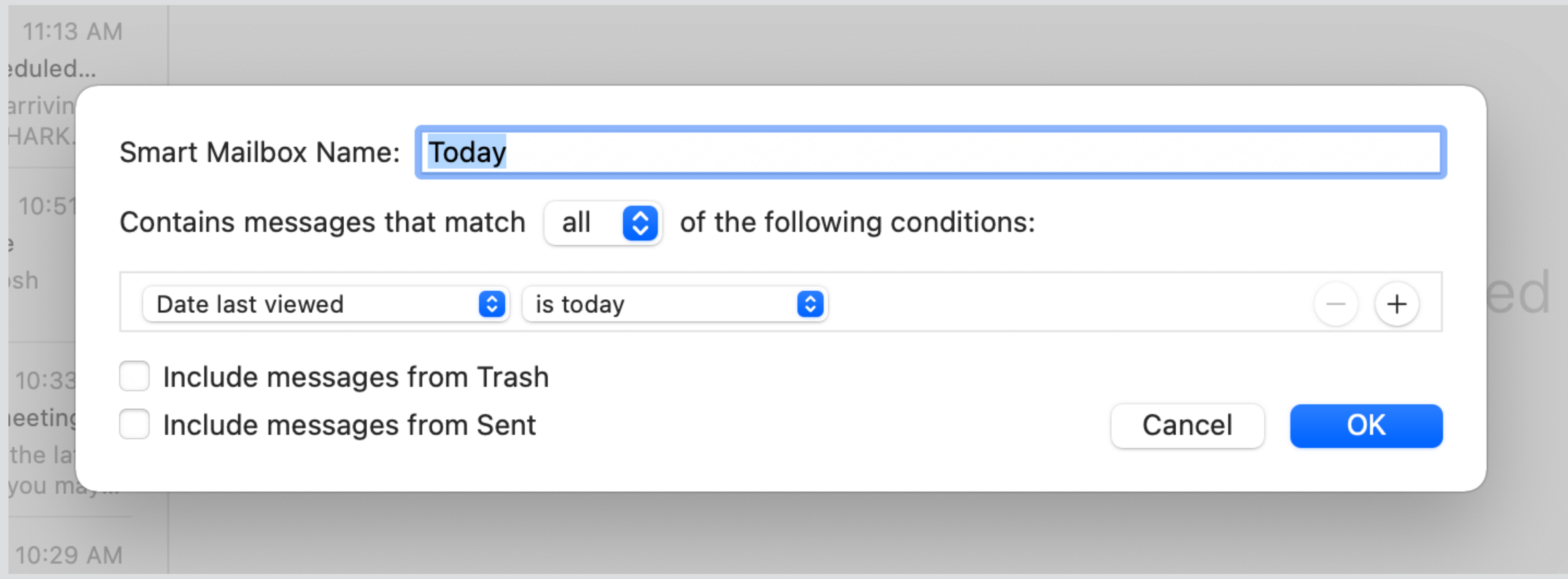
If any of the following conditions are met:

To does not contain dnj@mit.edu

Perform the following actions:

Move Message to mailbox: Groups

create a rule



Smart Mailbox Name: Today

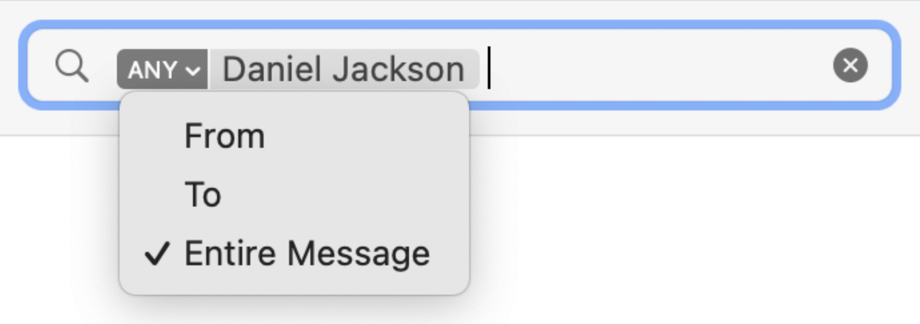
Contains messages that match all of the following conditions:

Date last viewed is today

☐ Include messages from Trash

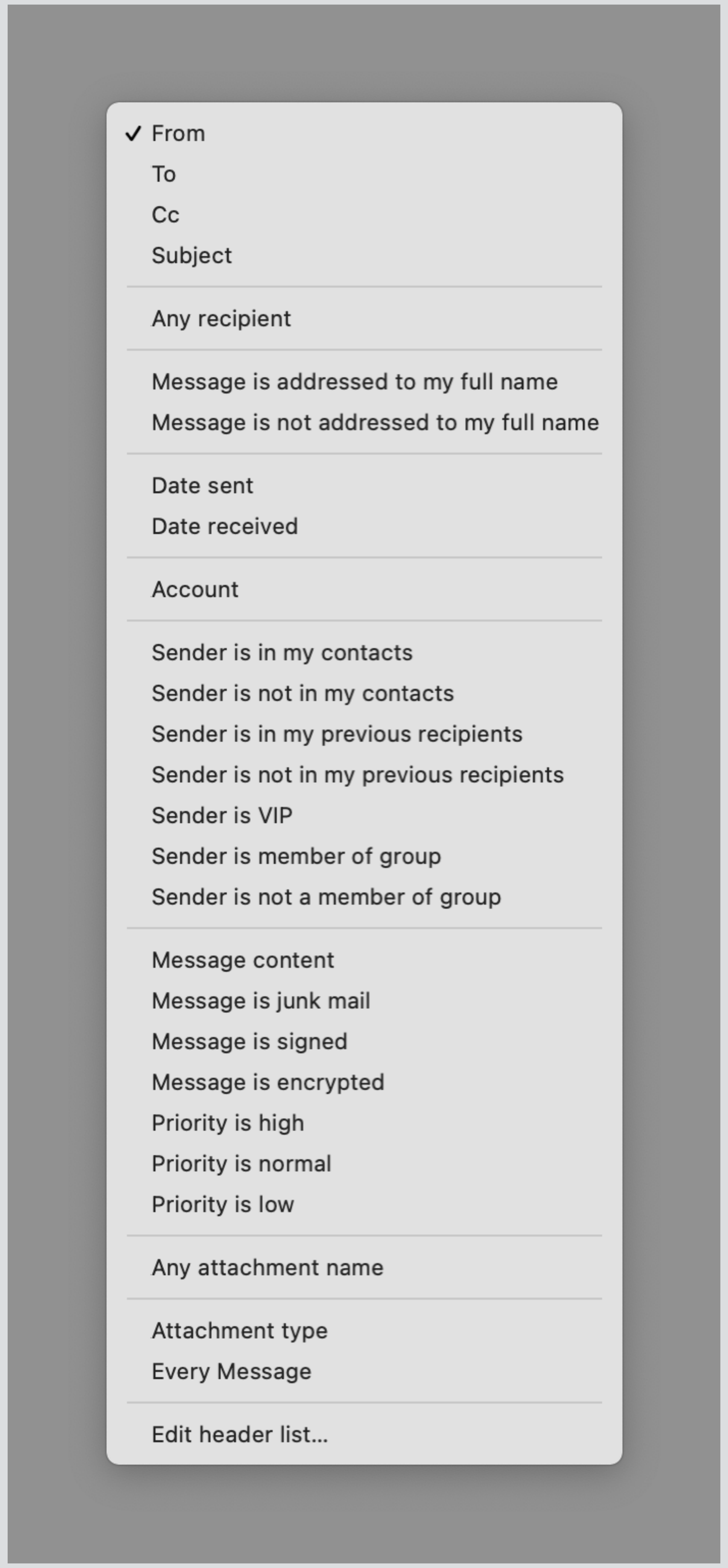
☐ Include messages from Sent

define a smart folder



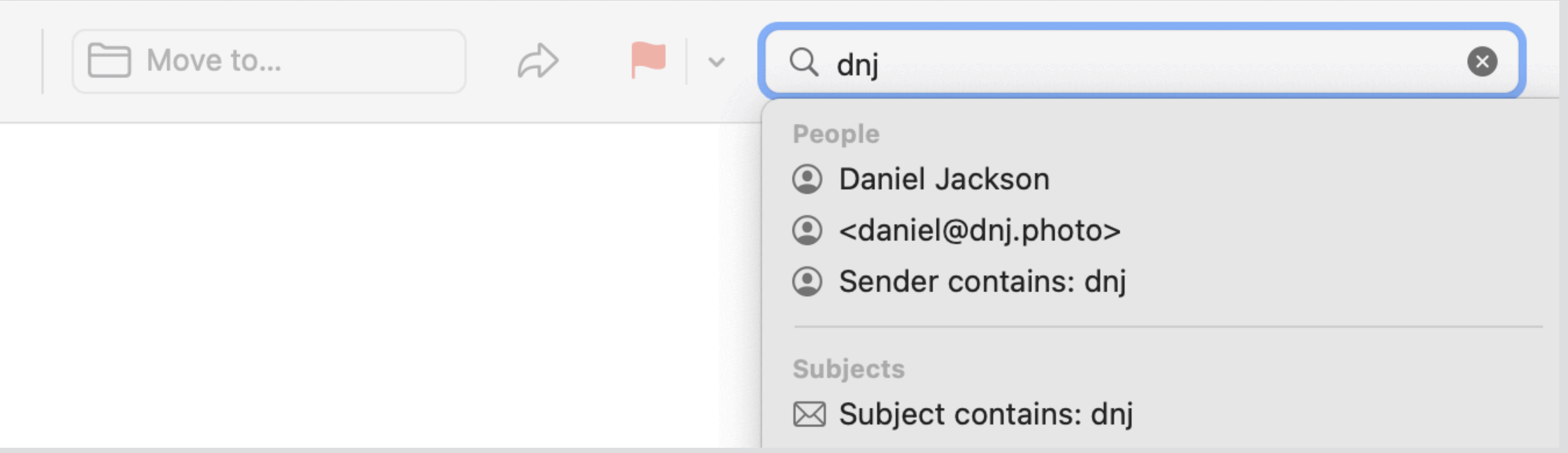
ANY Daniel Jackson

search options



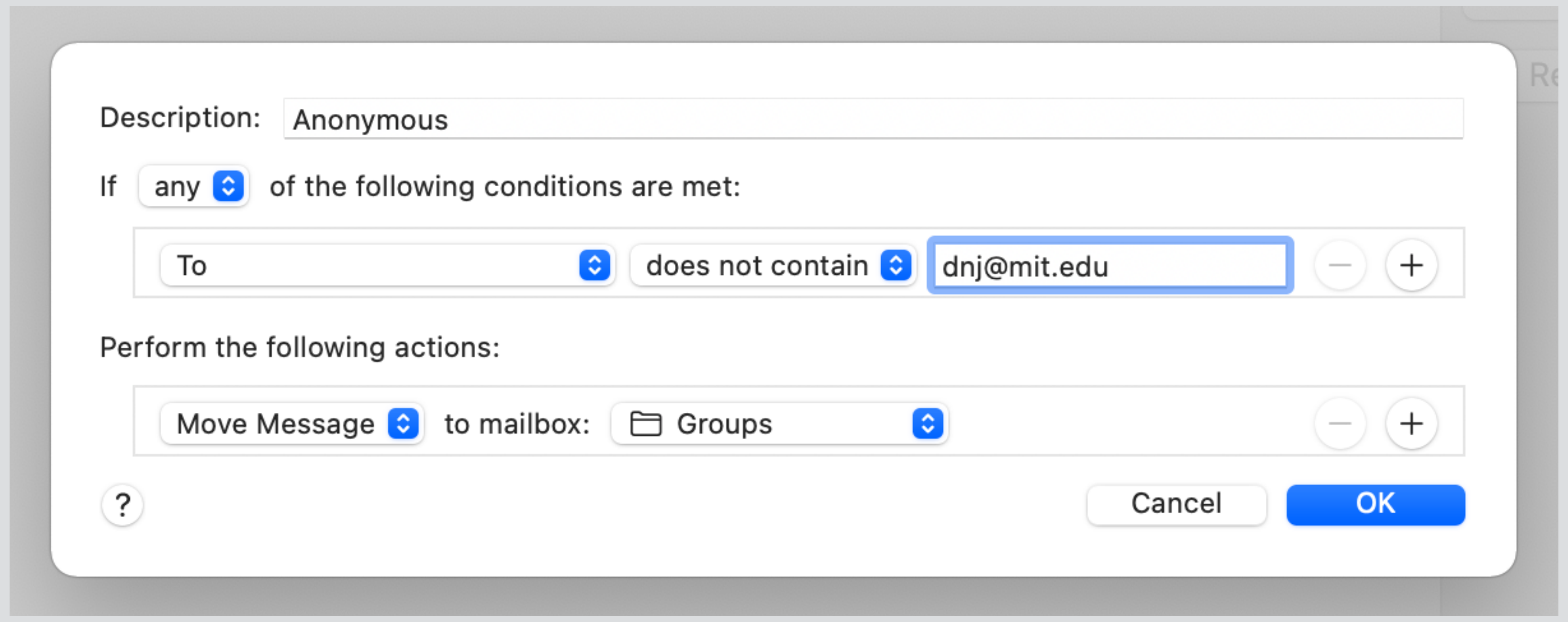
rule options

how many ways to filter messages?



dnj

search for a message



Description: Anonymous

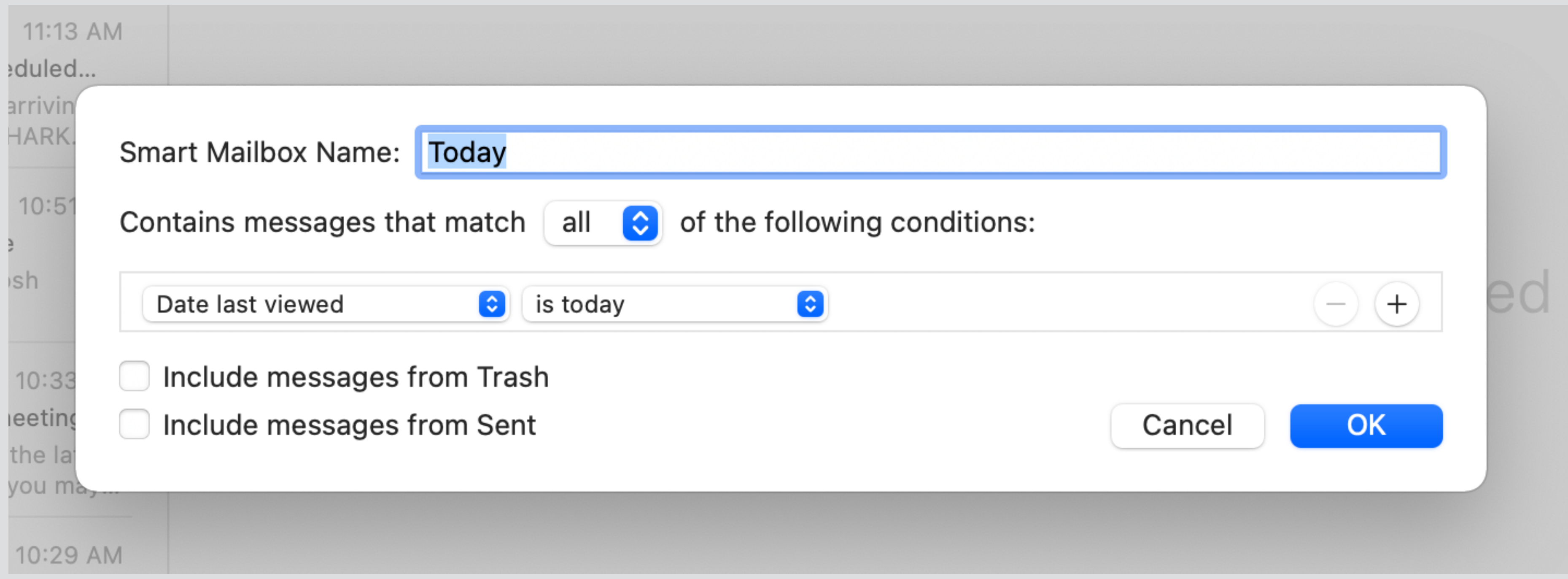
If any of the following conditions are met:

To does not contain dnj@mit.edu

Perform the following actions:

Move Message to mailbox: Groups

create a rule



Smart Mailbox Name: Today

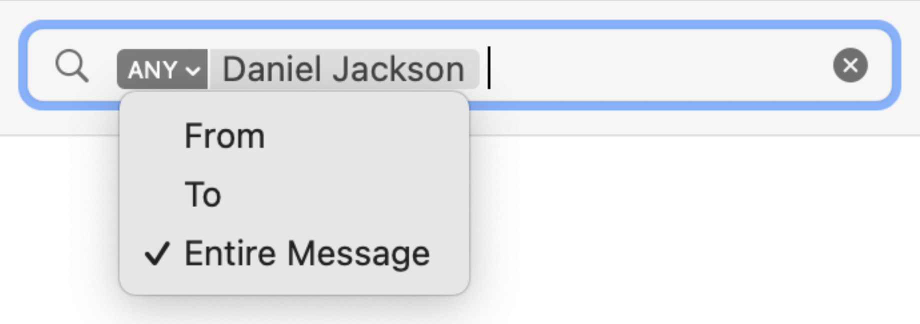
Contains messages that match all of the following conditions:

Date last viewed is today

□ Include messages from Trash

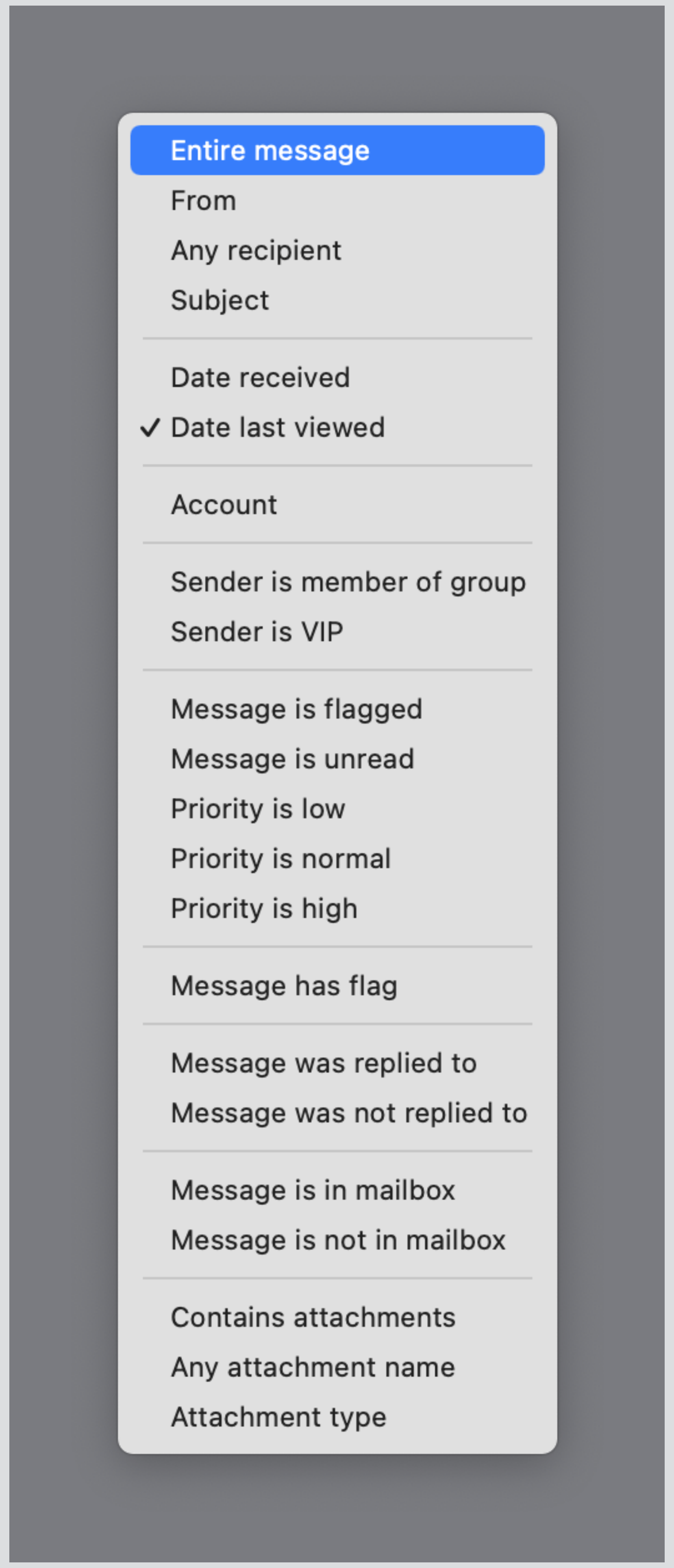
□ Include messages from Sent

define a smart folder

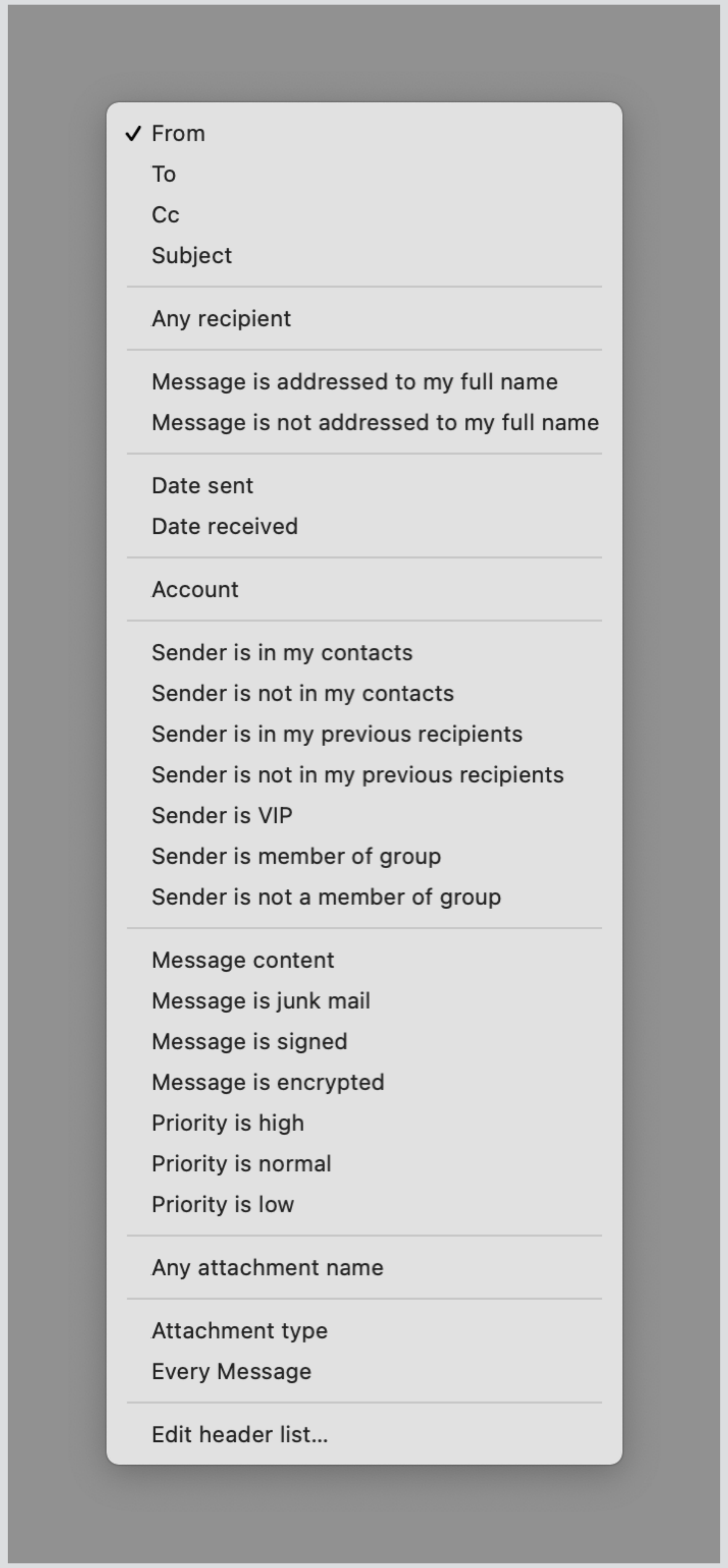


ANY Daniel Jackson

search options



smart folder options



rule options

diagnosis?

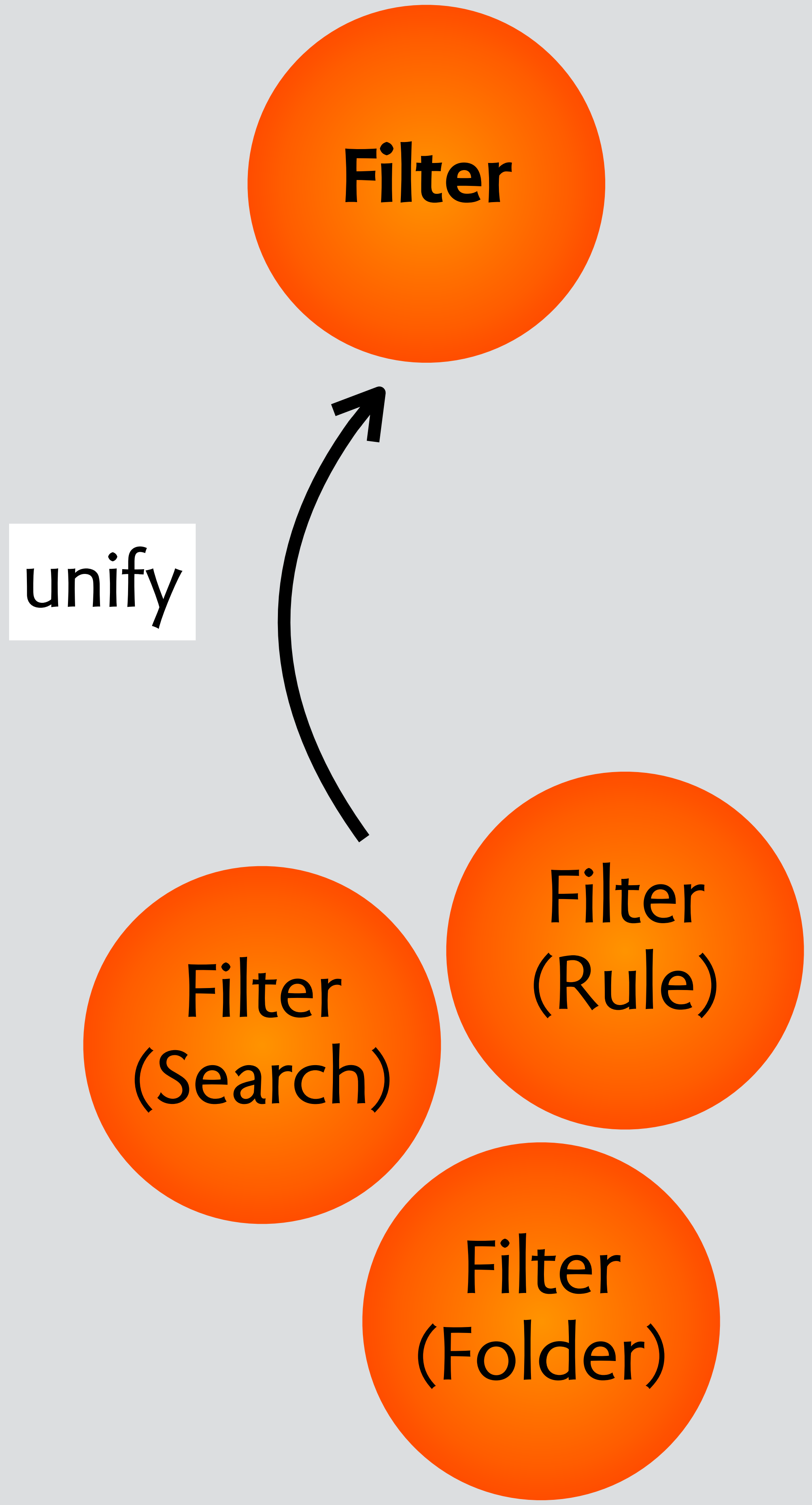
search, rule and smart folder
all include their own specialized concepts
incomparable features, different UIs

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

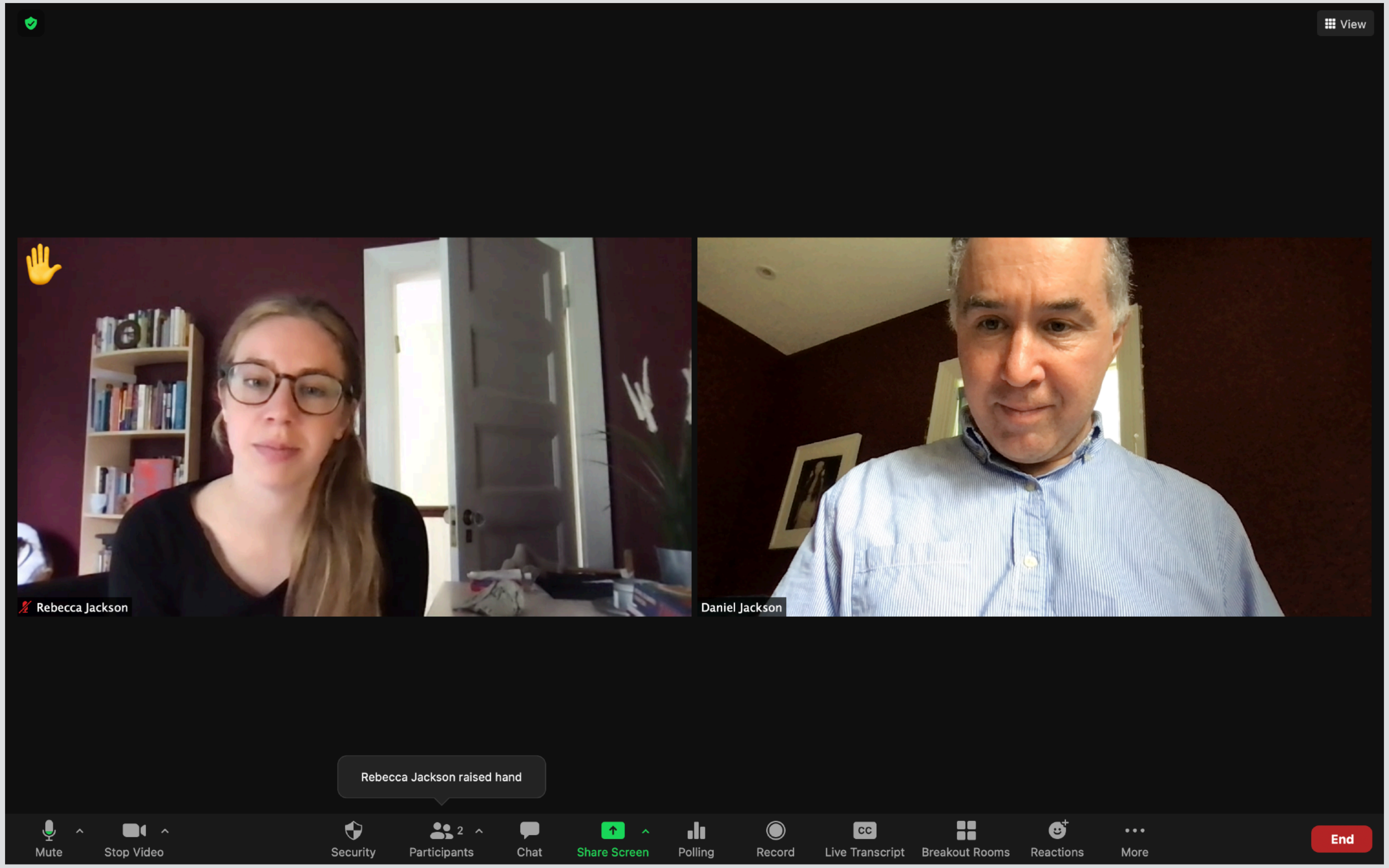
unify in a single message filter concept
include "create folder from search", eg



sticky hands

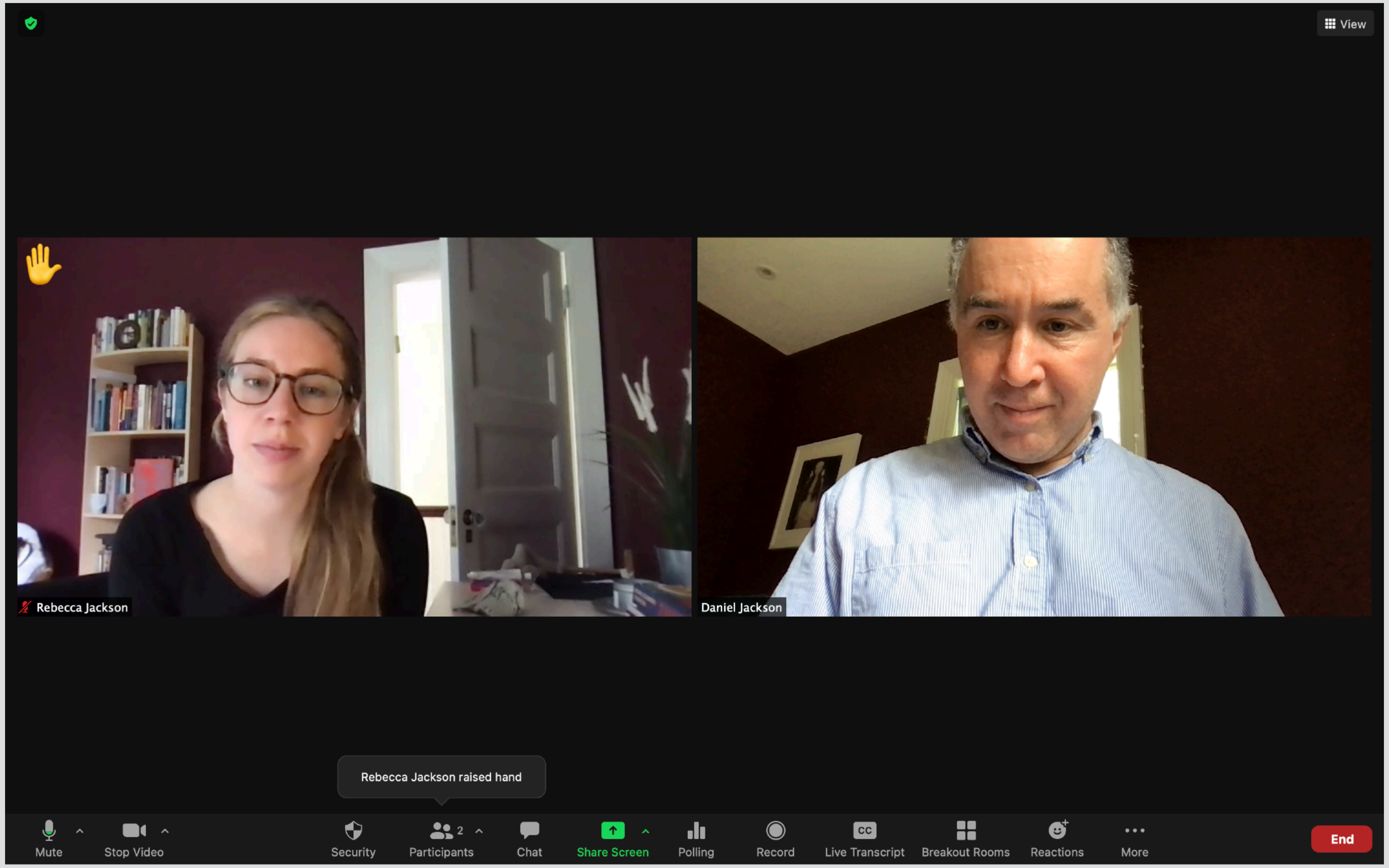
in zoom

zoom

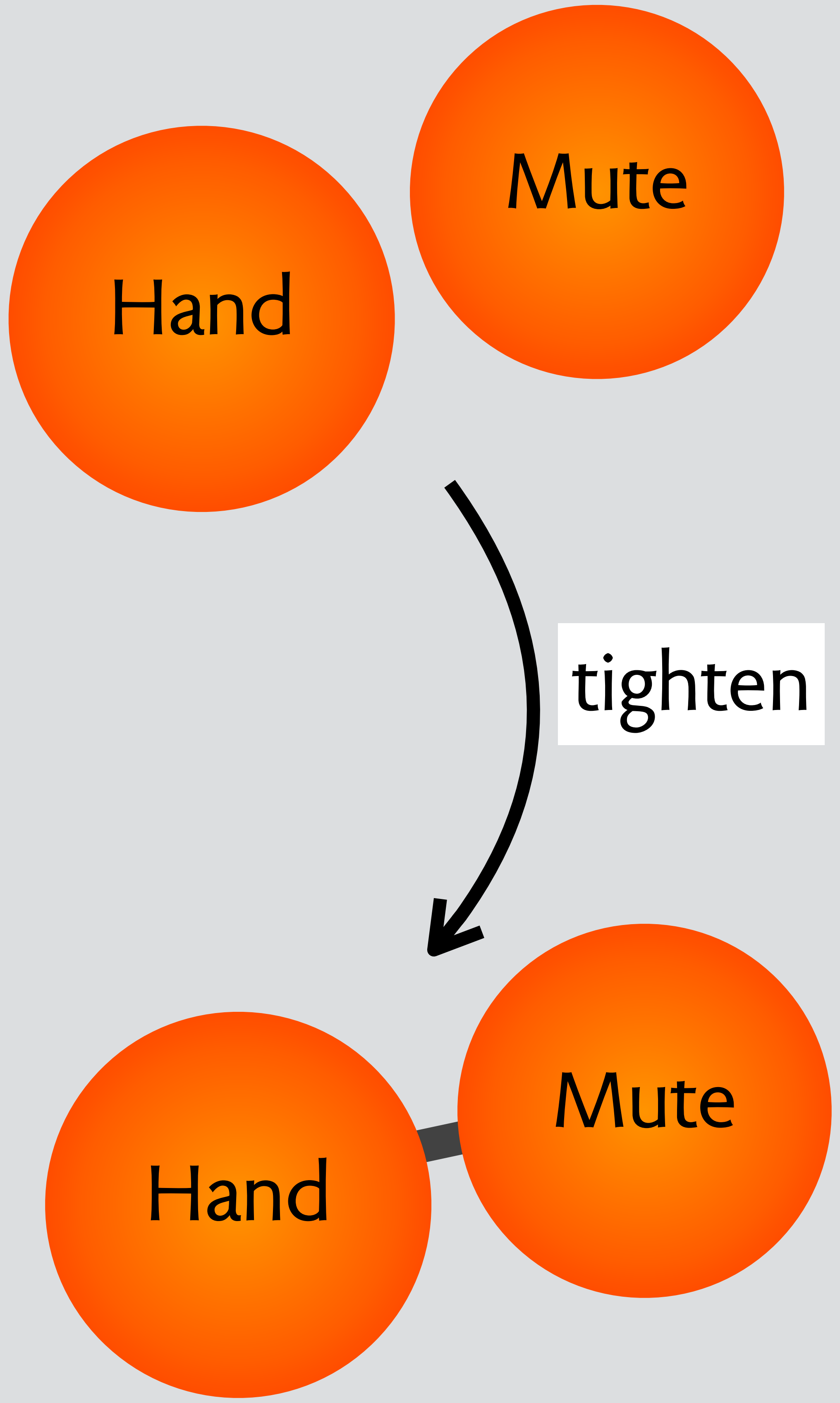


Rebecca Jackson raised hand

zoom



Rebecca Jackson raised hand



Hand

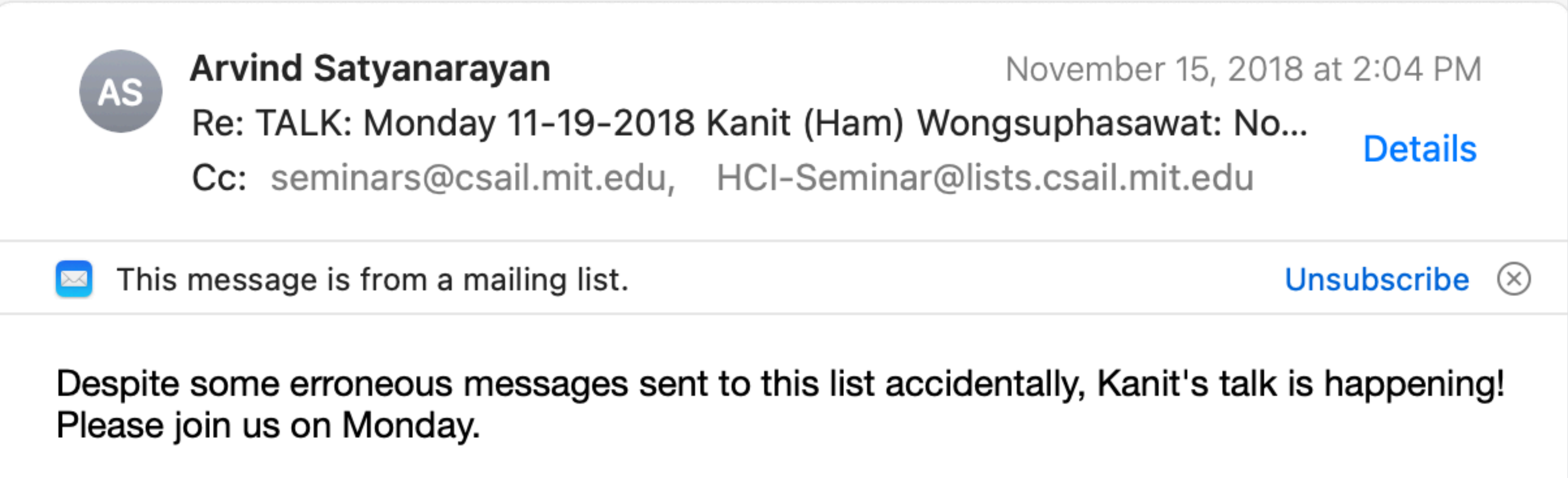
Mute

tighten

Hand

Mute

event deletion
in google calendar



Arvind Satyanarayan

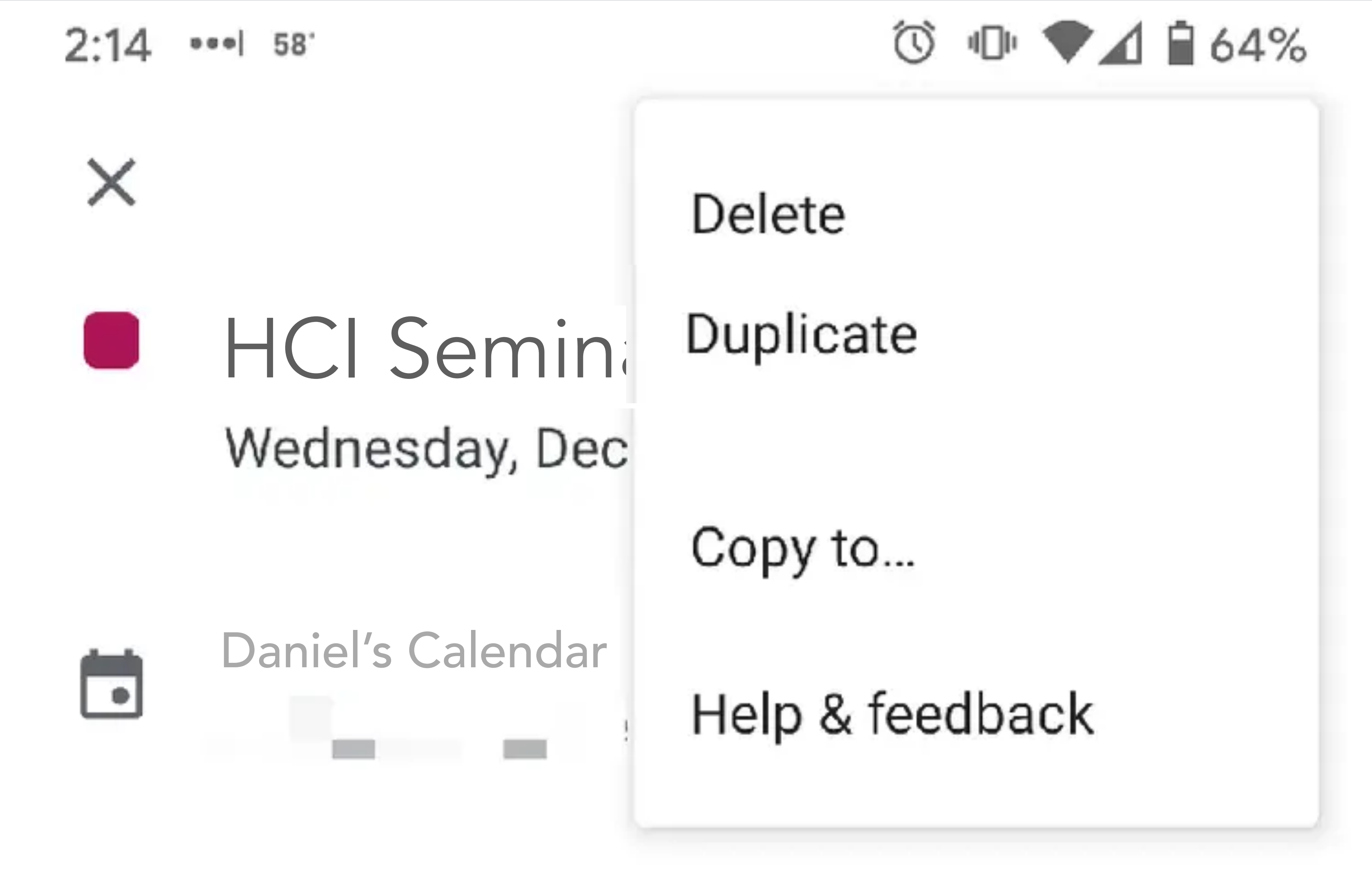
November 15, 2018 at 2:04 PM

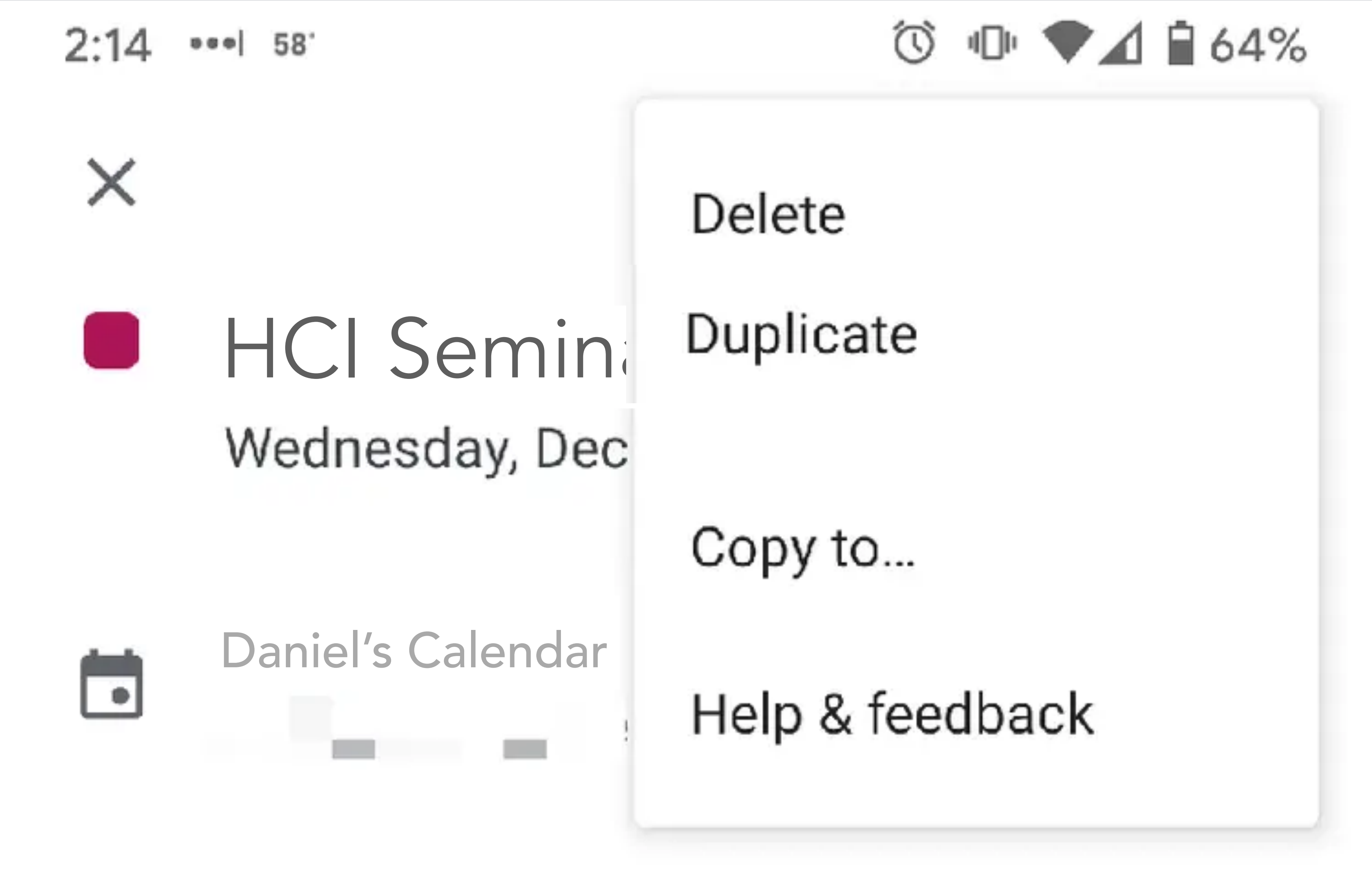
Re: TALK: Monday 11-19-2018 Kanit (Ham) Wongsuphasawat: No...

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

This message is from a mailing list.

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


2:14
58°
64%
HCI Semin
Wednesday, Dec
Daniel's Calendar
Delete
Duplicate
Copy to...
Help & feedback

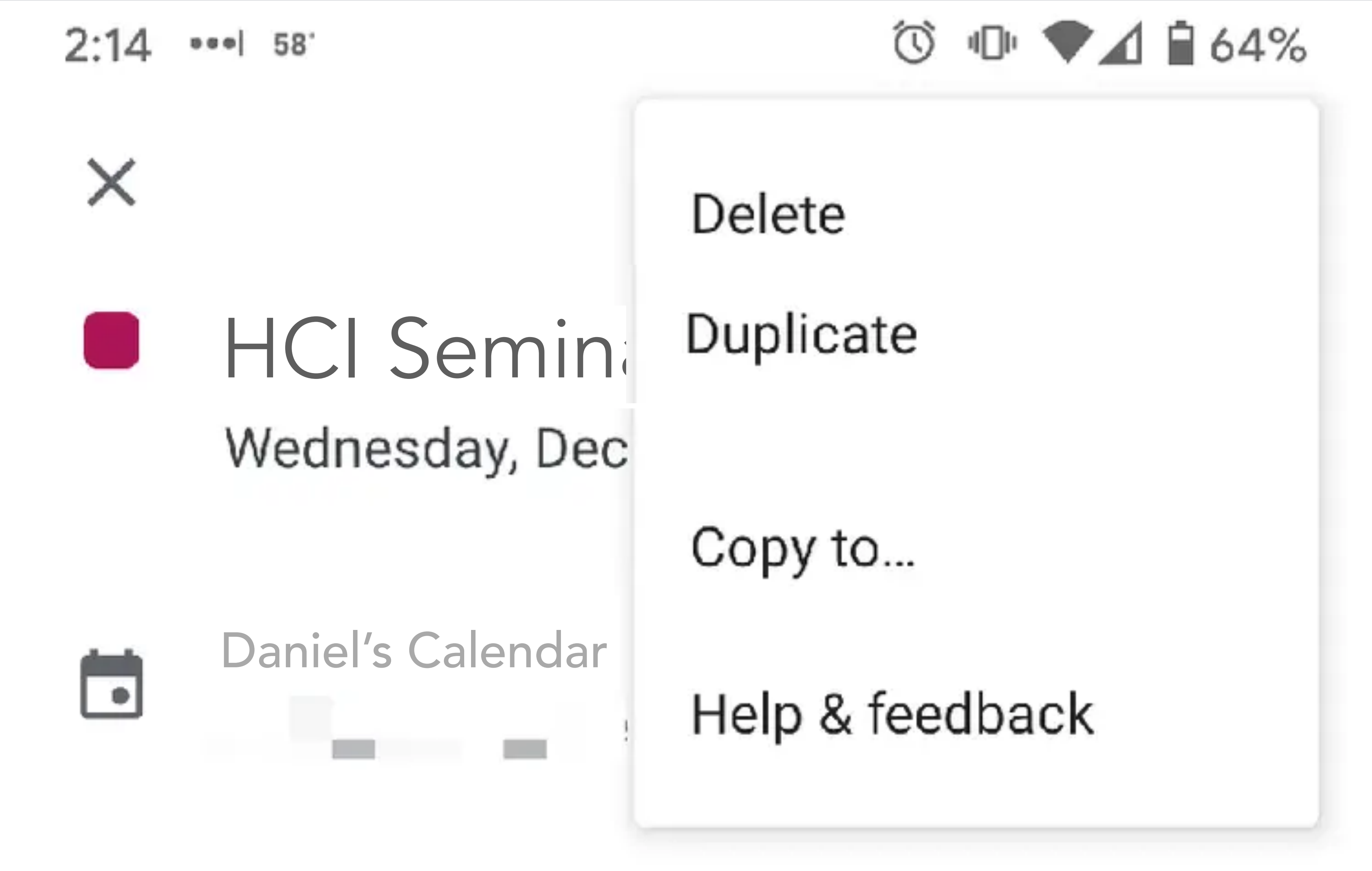


HCI Semin

Wednesday, Dec

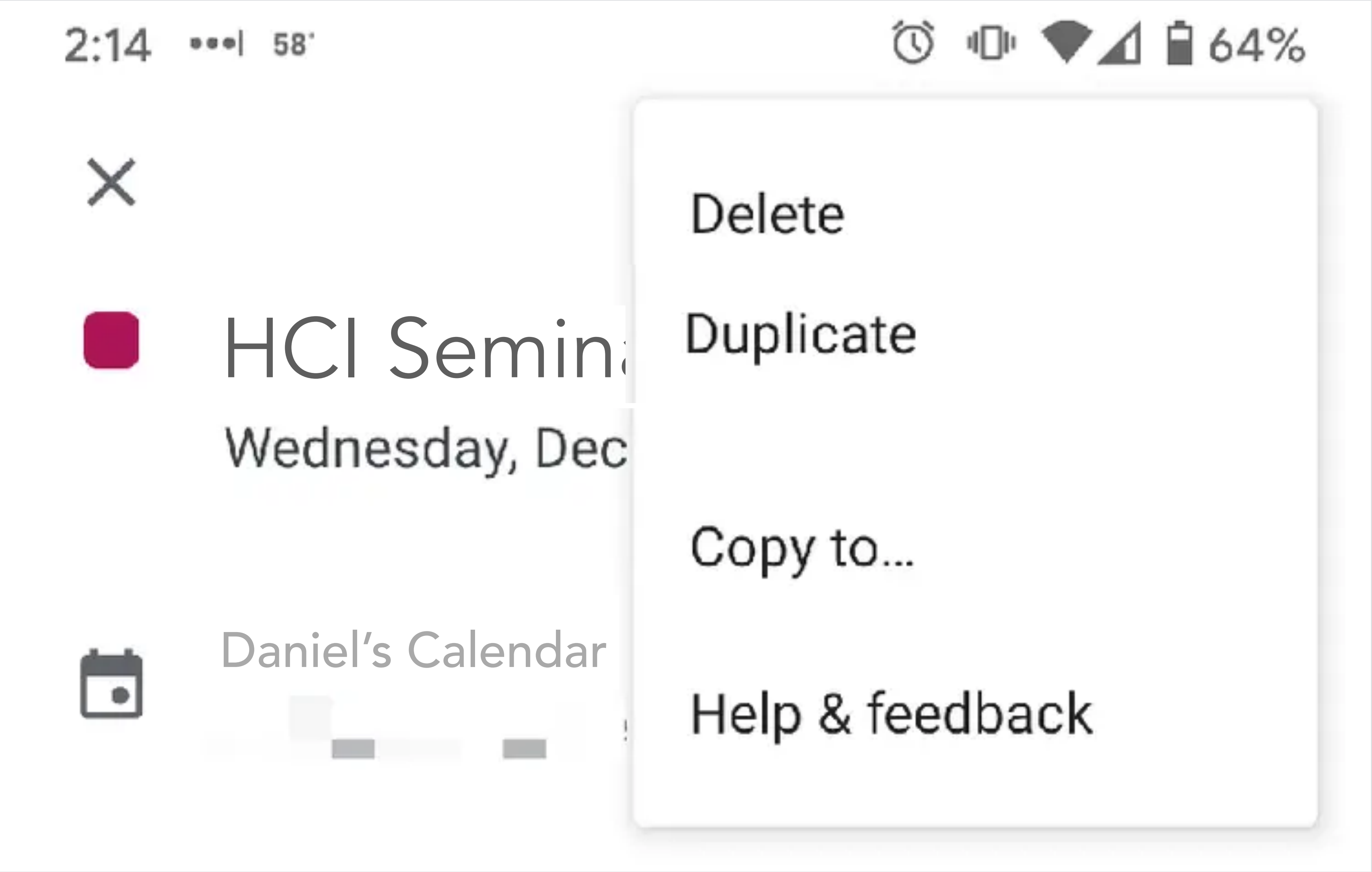
Daniel's Calendar

seminar announced as **email** to listserv with attached calendar event



seminar announced as **email** to listserv with attached calendar event

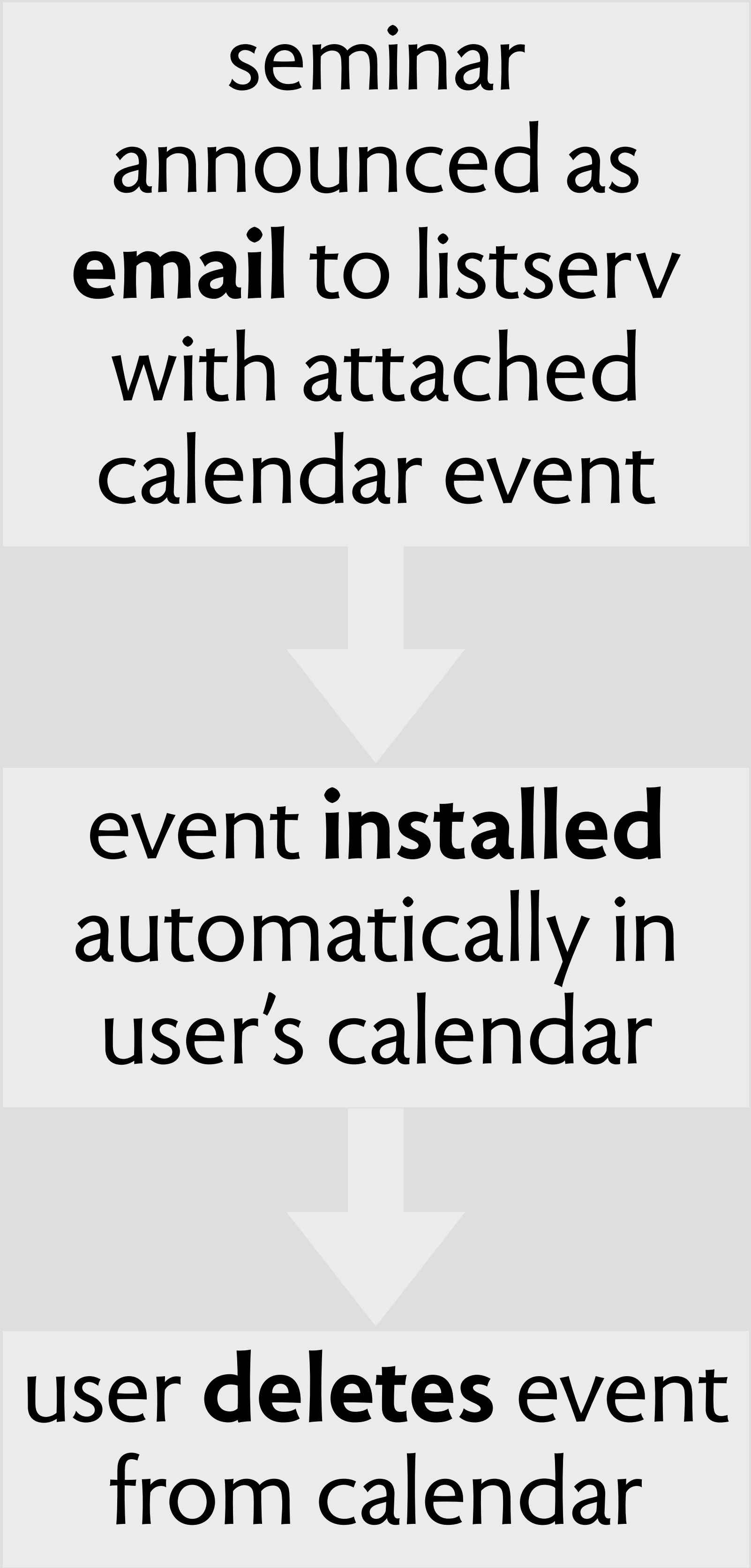
event **installed** automatically in user's calendar



HCI Semin

Wednesday, Dec

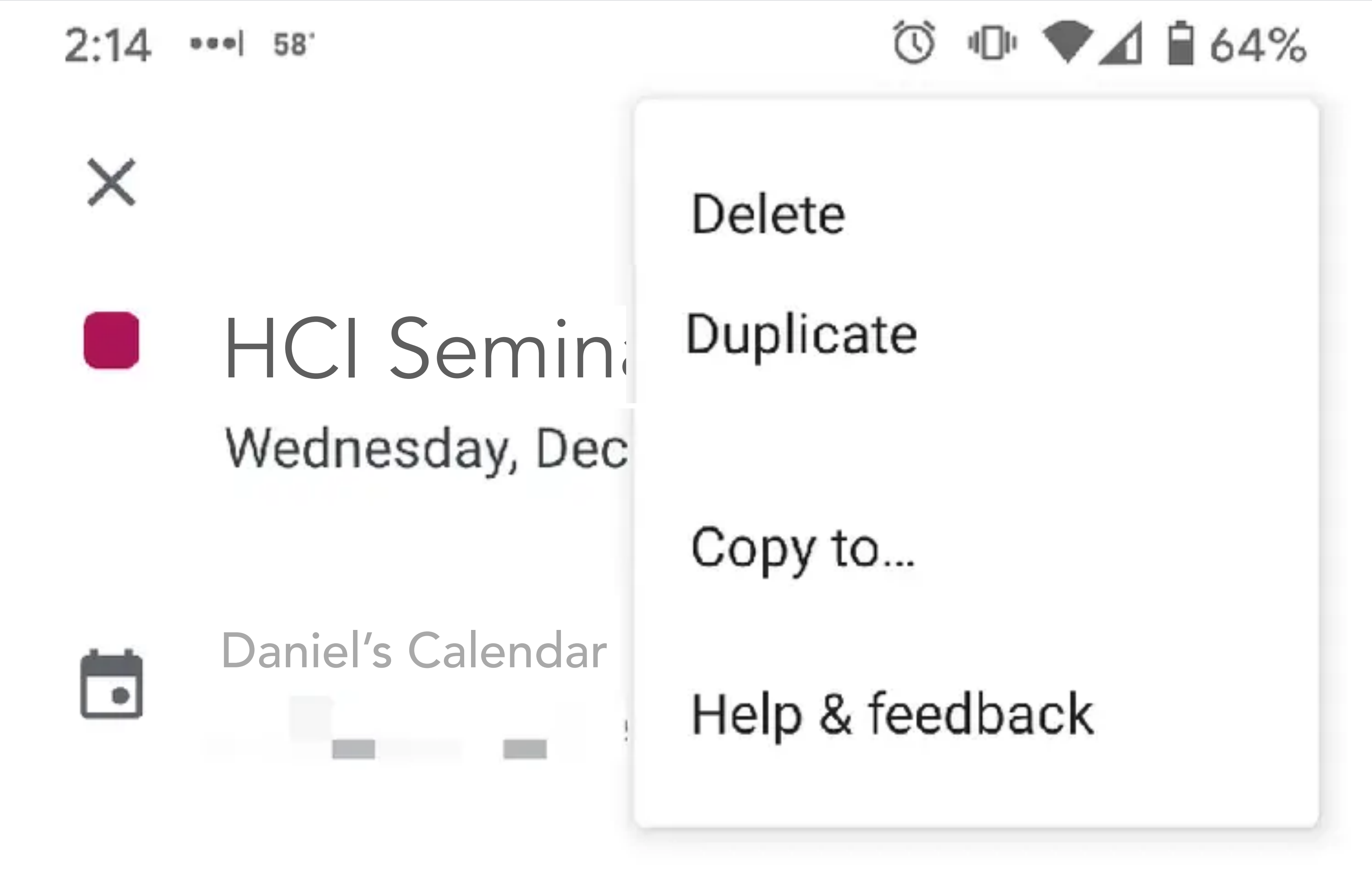
Daniel's Calendar

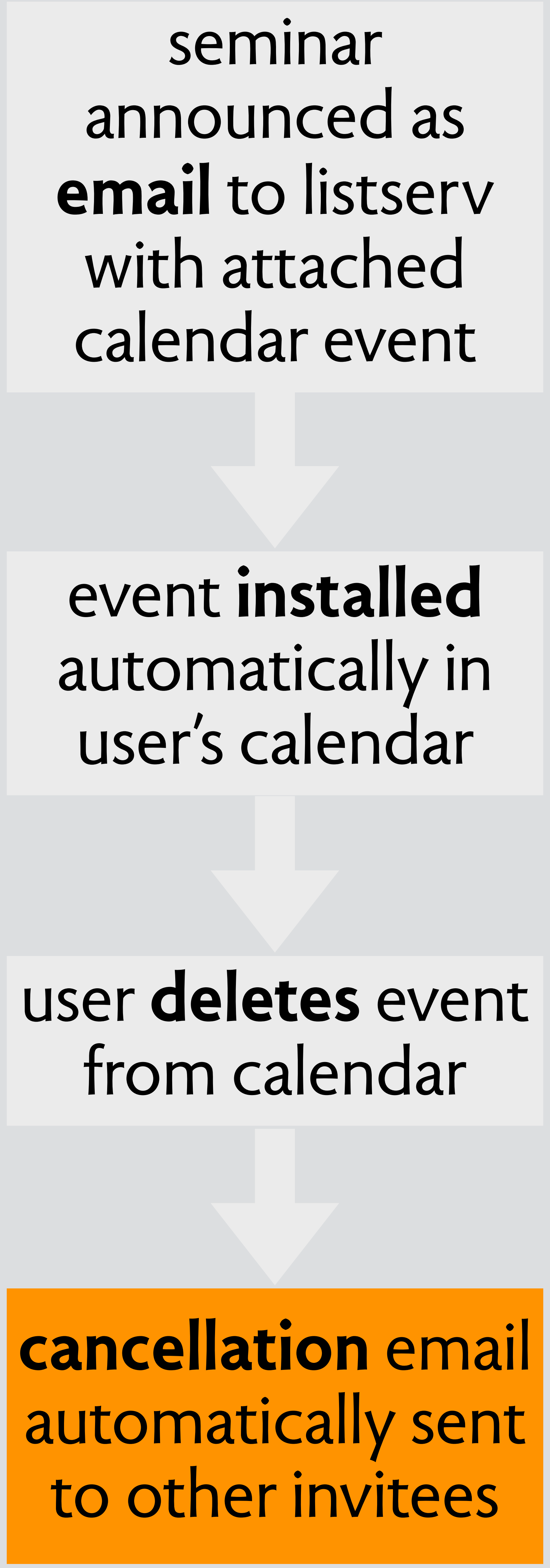


seminar announced as **email** to listserv with attached calendar event

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

user **deletes** event from calendar



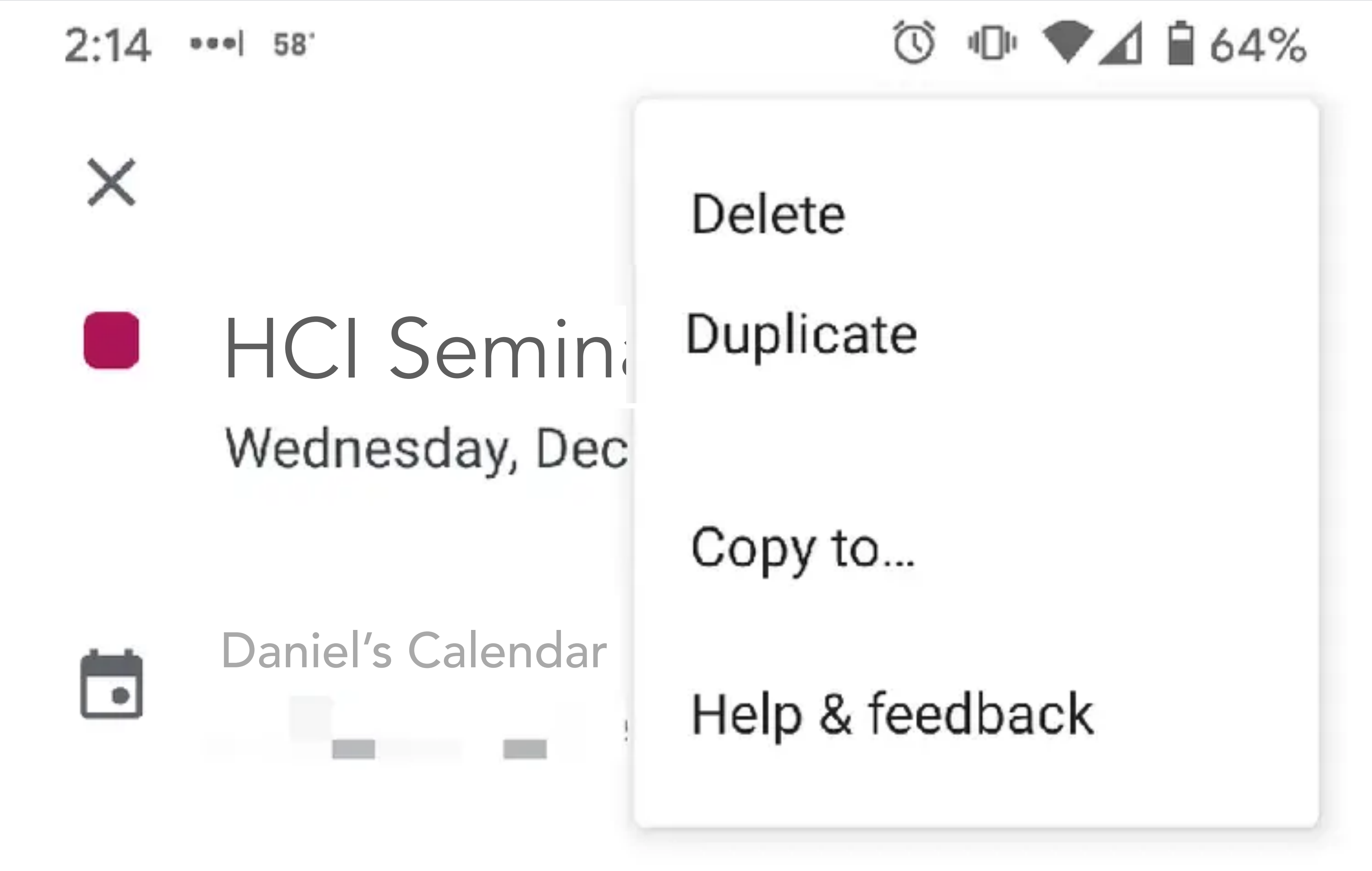


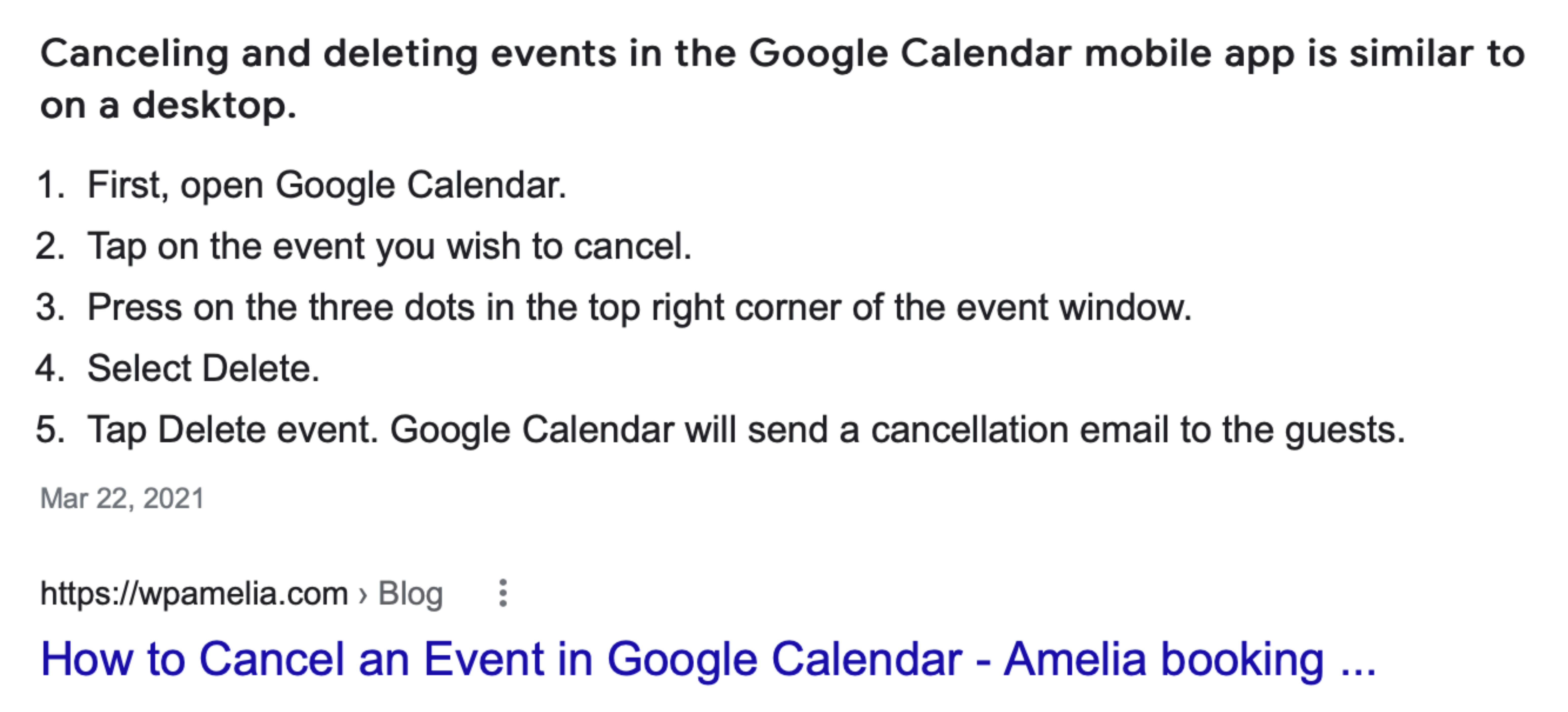
seminar announced as **email** to listserv with attached calendar event

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

user **deletes** event from calendar

cancellation email automatically sent to other invitees





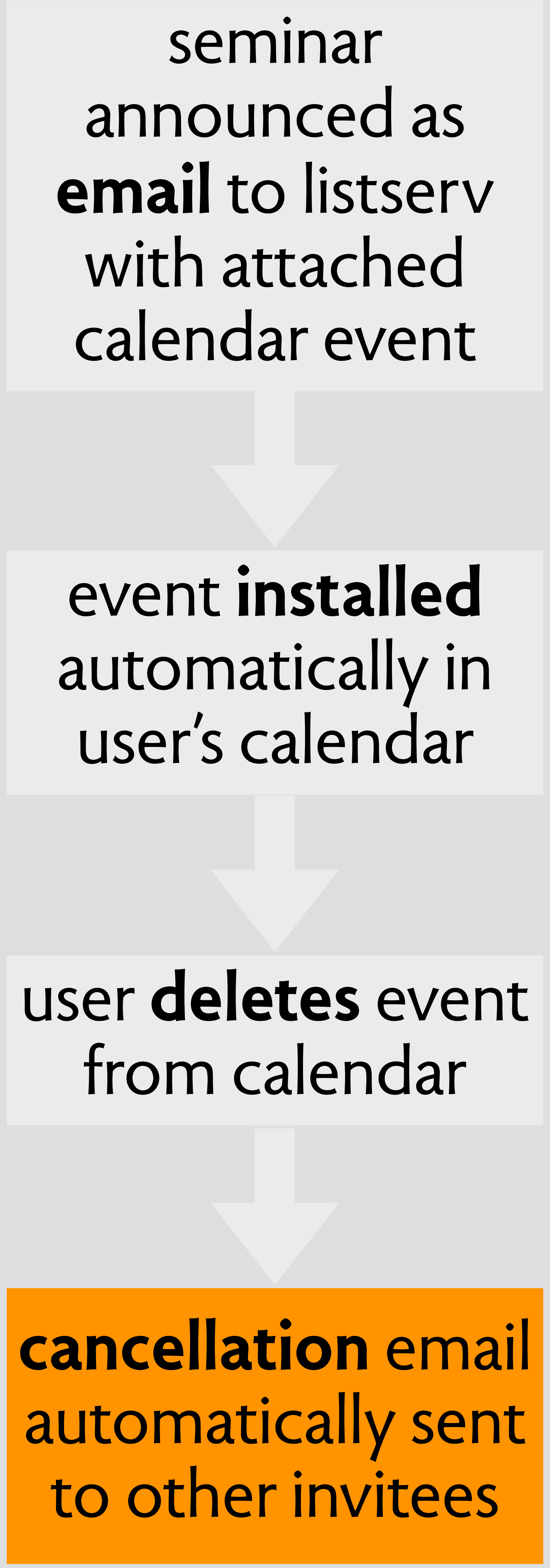
Canceling and deleting events in the Google Calendar mobile app is similar to on a desktop.

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 ...

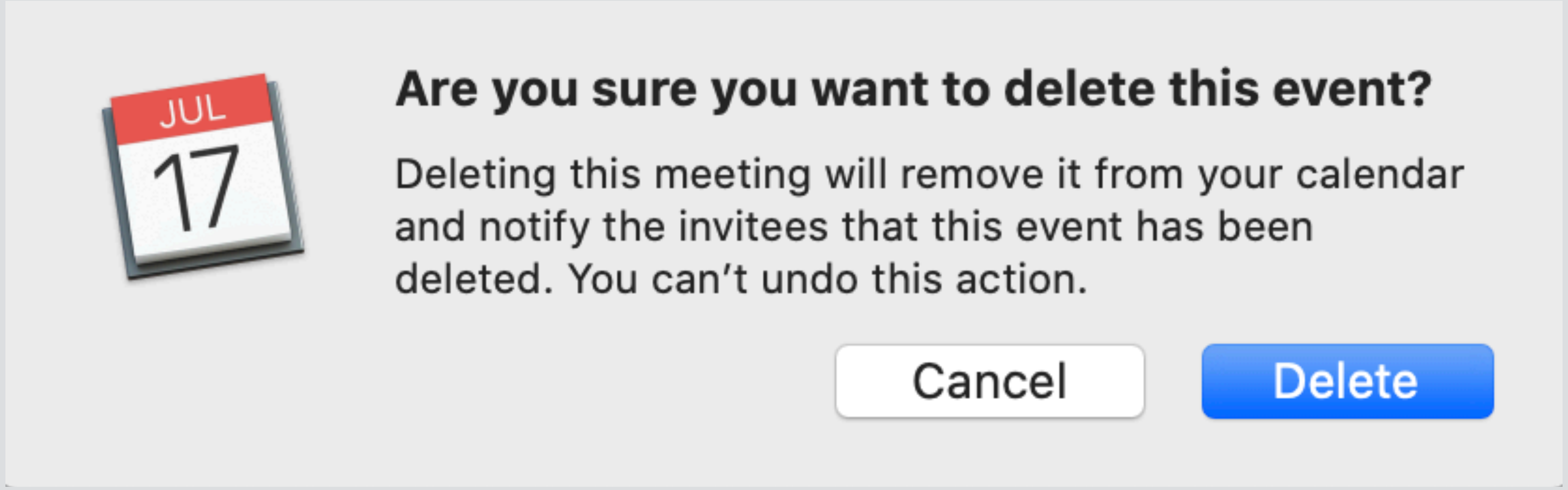


seminar announced as **email** to listserv with attached calendar event

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

user **deletes** event from calendar

cancellation email automatically sent to other invitees



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.

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)
 ...
```

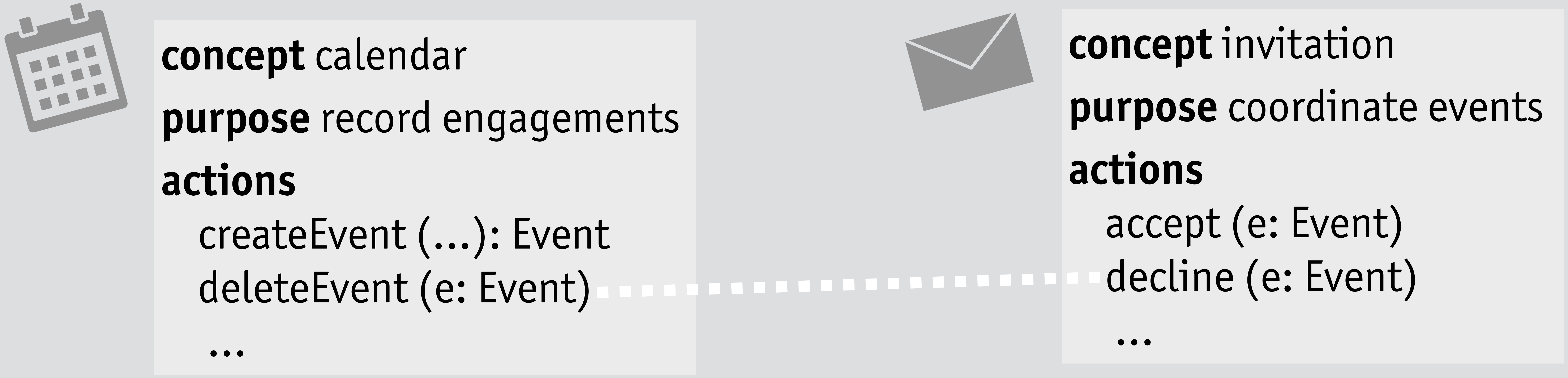
concept invitation

purpose coordinate events

actions

```
accept (e: Event)
decline (e: Event)
 ...
```

diagnosis?



concept calendar

purpose record engagements

actions

createEvent (...): Event
deleteEvent (e: Event)
...

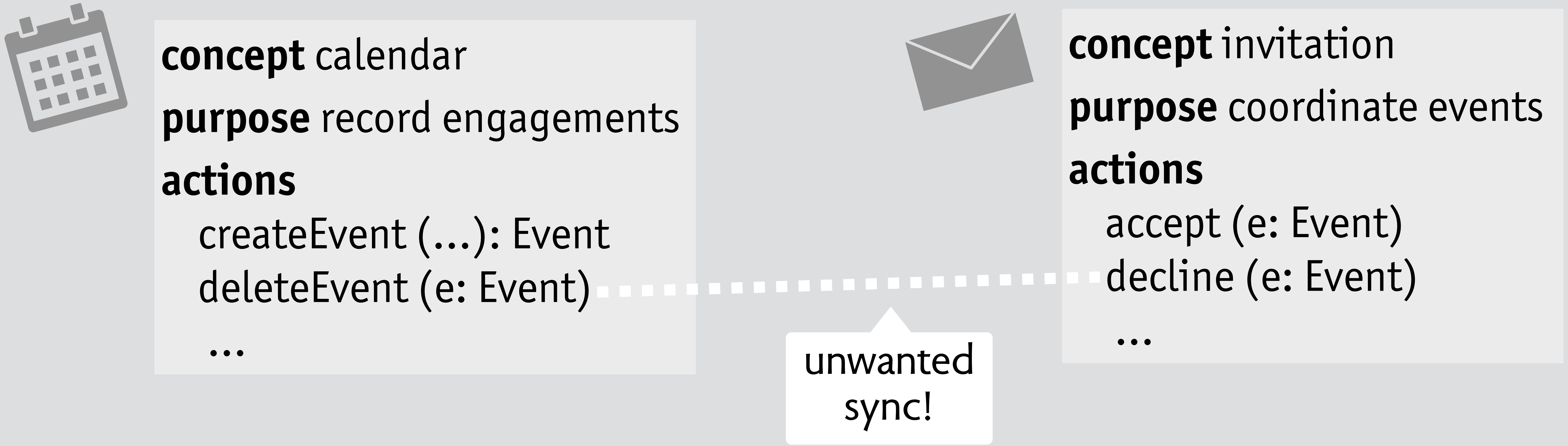
concept invitation

purpose coordinate events

actions

accept (e: Event)
decline (e: Event)
...

diagnosis?



concept calendar

purpose record engagements

actions

  createEvent (…): Event
  deleteEvent (e: Event)
  …

concept invitation

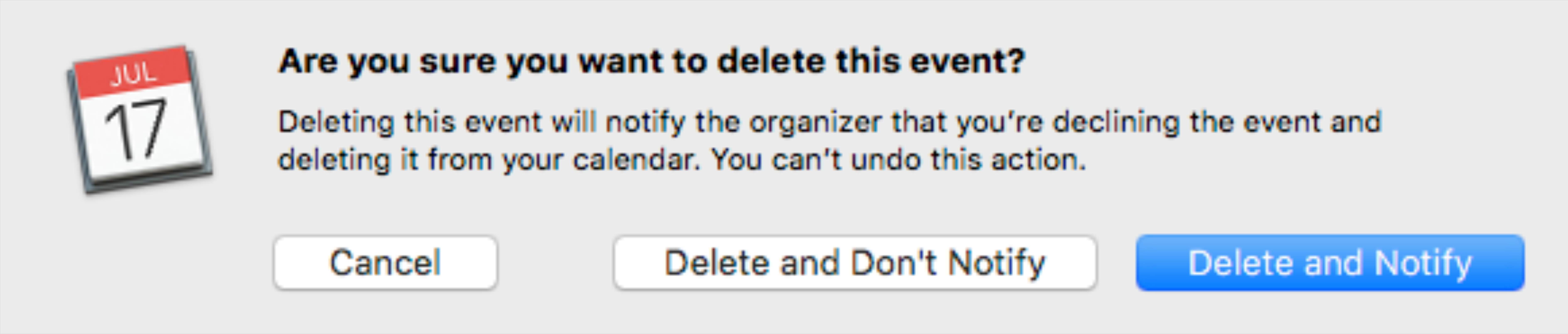
purpose coordinate events

actions

  accept (e: Event)
  decline (e: Event)
  …

unwanted sync!

apple's solution

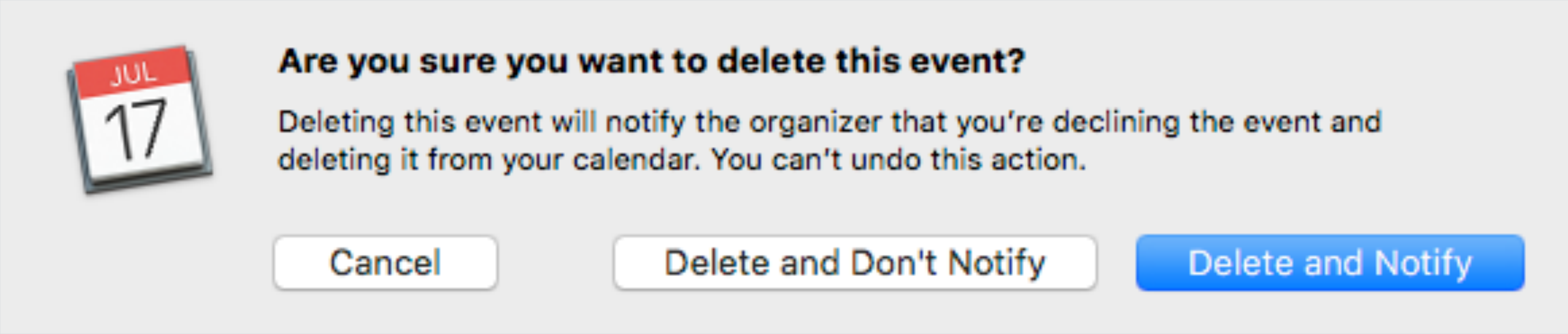


Are you sure you want to delete this event?

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.

resolution to design problem
make sync optional

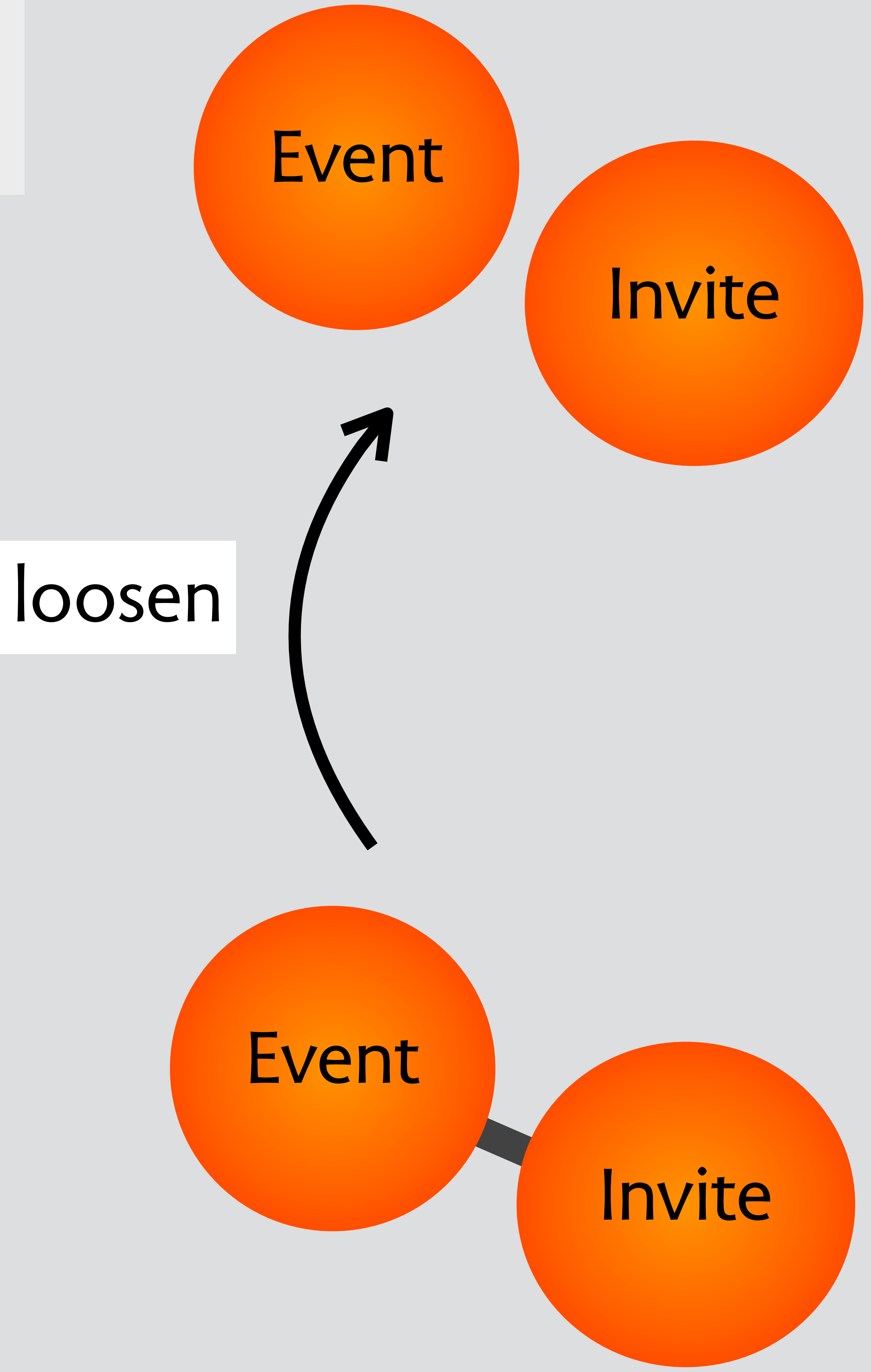
apple's solution



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.

resolution to design problem
make sync optional



Event
Invite
loosen
Event
Invite

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?

takeaways

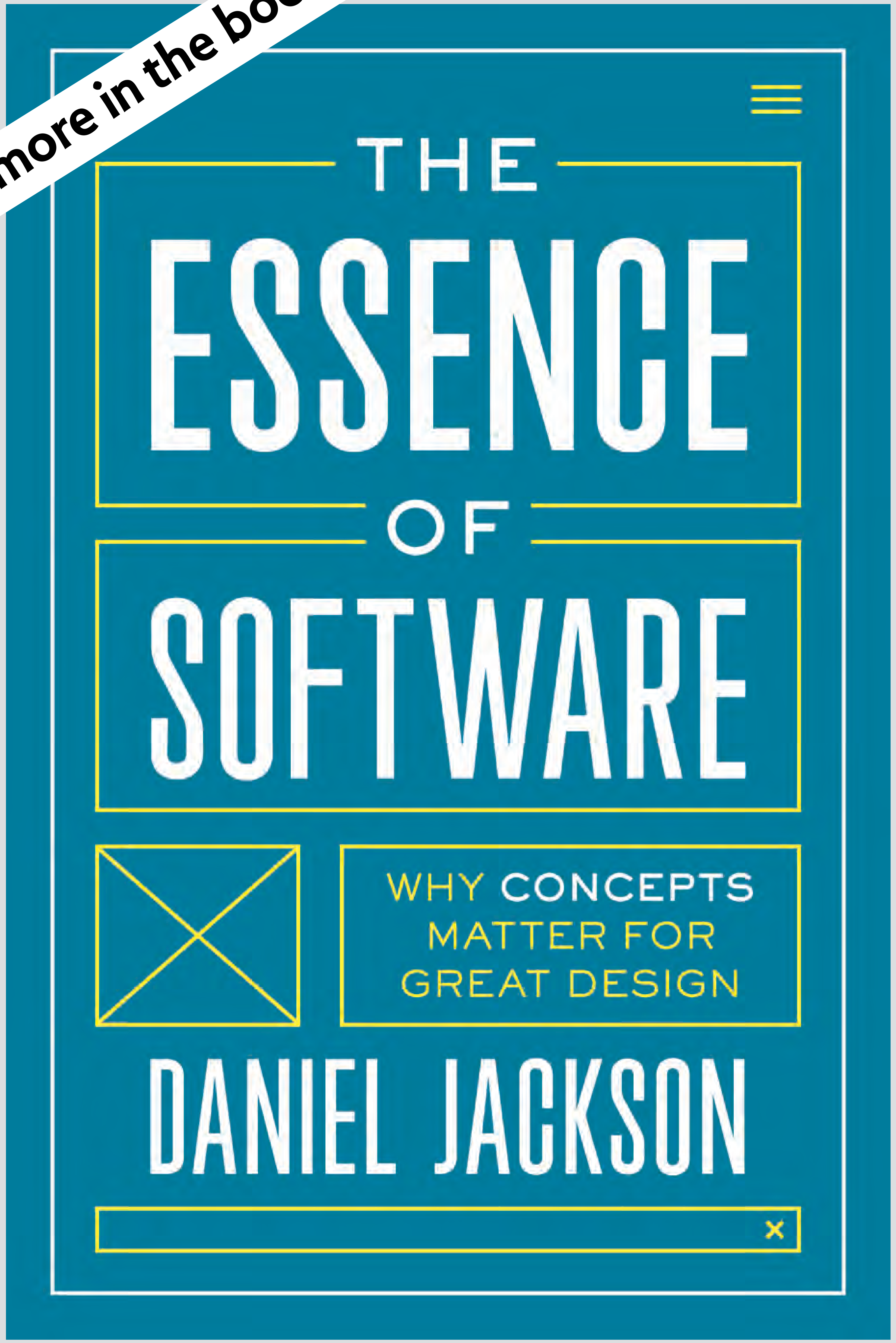
structure your software design with concepts
inventory the concepts, identify the critical ones
see if you can describe them fully independently
then formulate interactions as synchronizations

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?

in formal methods
can concepts help structure & validate models?

much more in the book



essenceofsoftware.com

newsletter
essenceofsoftware.com/subscribe

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