# Spatial Canvases

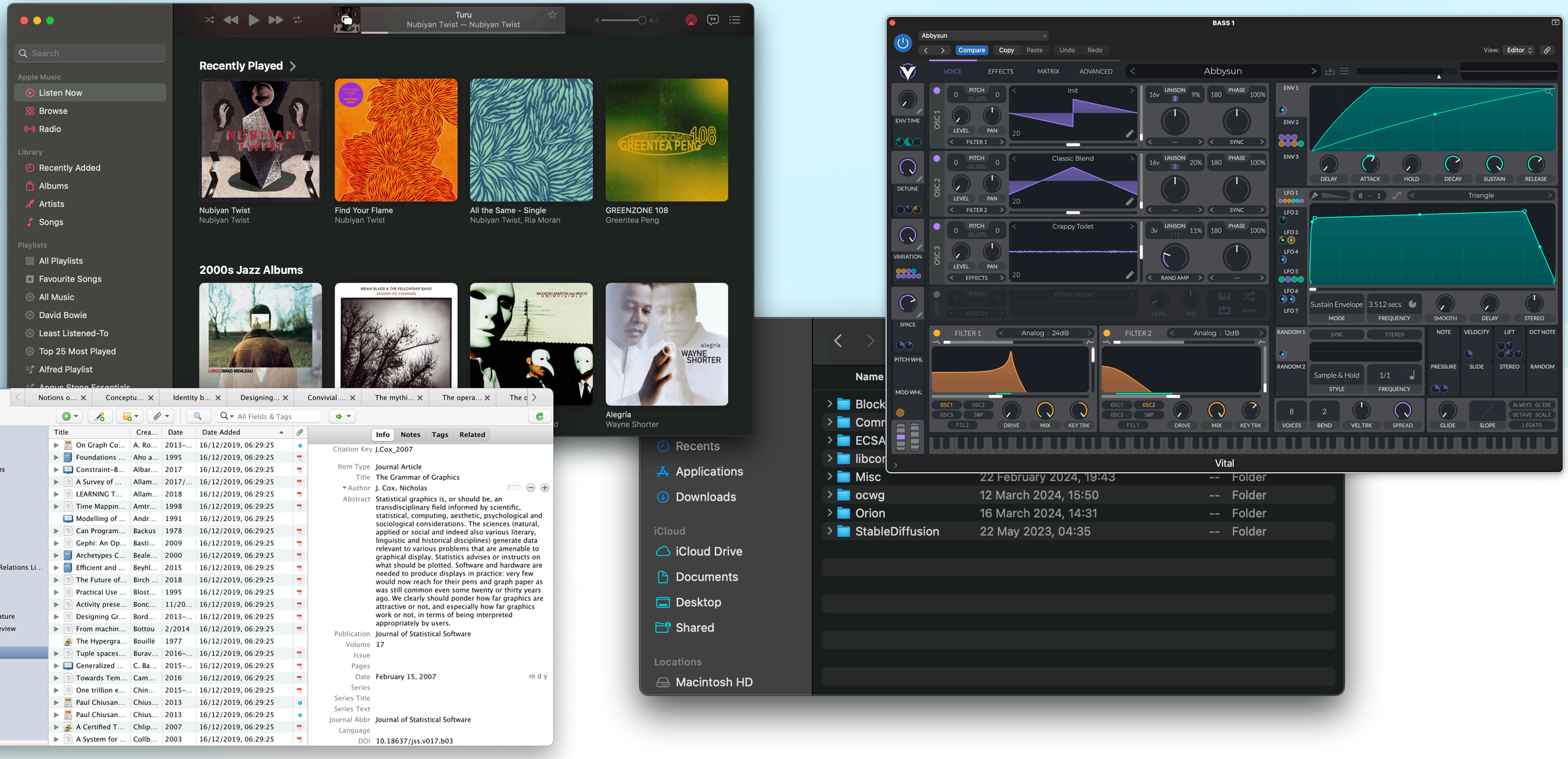
## Towards an ‘Integration Domain’ for HCI

Orion Reed @ Causal Islands LA 2024

# Outline

1. Software Interfaces
2. Information Substrates
3. Demo
4. An ‘Integration Domain’ for HCI?
5. One Fun Idea

# 1. Software Interfaces

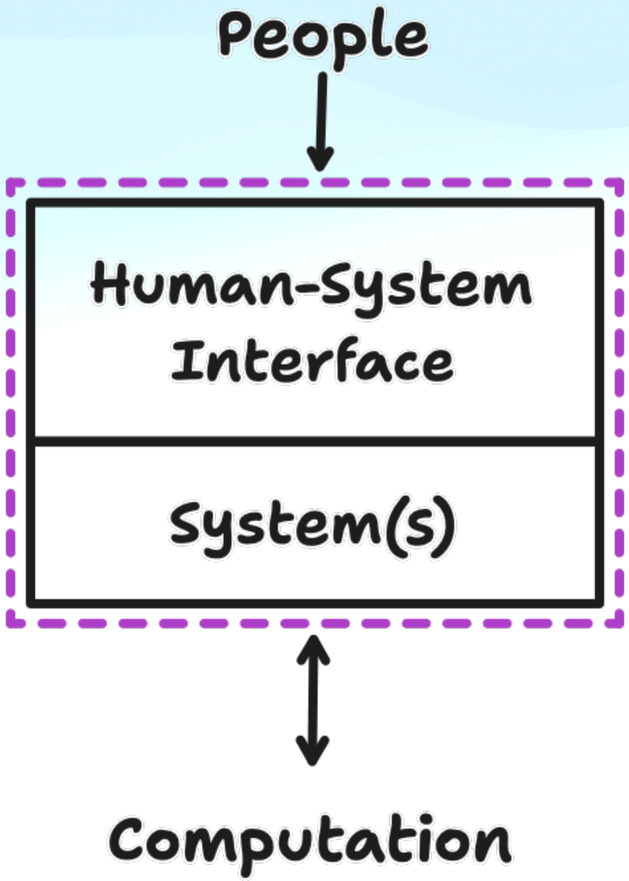

Turu
Nubiyan Twist — Nubiyan Twist
Search
Apple Music
Listen Now
Browse
Radio
Library
Recently Added
Albums
Artists
Songs
Playlists
All Playlists
Favourite Songs
All Music
David Bowie
Least Listened-To
Top 25 Most Played
Alfred Playlist
Recently Played
Nubiyan Twist
Nubiyan Twist
Find Your Flame
Nubiyan Twist
All the Same - Single
Nubiyan Twist, Ria Moran
GREENZONE 108
Greentea Peng
2000s Jazz Albums
Alegría
Wayne Shorter
BASS 1
Abbysun
Compare
Copy
Paste
Undo
Redo
View: Editor
VOICE
EFFECTS
MATRIX
ADVANCED
Init
Classic Blend
Crappy Toilet
White Noise
FILTER 1
Analog : 24dB
FILTER 2
Analog : 12dB
Vital
Recents
Applications
Downloads
iCloud
iCloud Drive
Documents
Desktop
Shared
Locations
Macintosh HD
Name
Block
Comr
ECSA
libcor
Misc
ocwg
Orion
StableDiffusion
22 February 2024, 19:43
12 March 2024, 15:50
16 March 2024, 14:31
22 May 2023, 04:35
Folder
My Library
Kinema
Misc.
Wiki Entries from Labs
My Publications
Duplicate Items
Unfiled Items
Bin
Group Libraries
Abstra Labs
Content-Addressed Relations Li...
Identity Paper
Malleable Systems
Human Interfaces
OaR Annotated Literature
OaR Literature-to-Review
OaR Sources
Literature
Talks
Web Links
X Missing URLs
Duplicate Items
Unfiled Items
Bin
Berkeley 111 Physics
Info
Notes
Tags
Related
Citation Key J.Cox_2007
Item Type Journal Article
Title The Grammar of Graphics
Author J. Cox, Nicholas
Abstract Statistical graphics is, or should be, an transdisciplinary field informed by scientific, statistical, computing, aesthetic, psychological and sociological considerations. The sciences (natural, applied or social and indeed also various literary, linguistic and historical disciplines) generate data relevant to various problems that are amenable to graphical display. Statistics advises or instructs on what should be plotted. Software and hardware are needed to produce displays in practice: very few would now reach for their pens and graph paper as was still common even some twenty or thirty years ago. We clearly should ponder how far graphics are attractive or not, and especially how far graphics work or not, in terms of being interpreted appropriately by users.
Publication Journal of Statistical Software
Volume 17
Issue
Pages
Date February 15, 2007
Series
Series Title
Series Text
Journal Abbr Journal of Statistical Software
Language
DOI 10.18637/jss.v017.b03

Current interfaces are **rigid**, **siloed**, and **mass-produced**

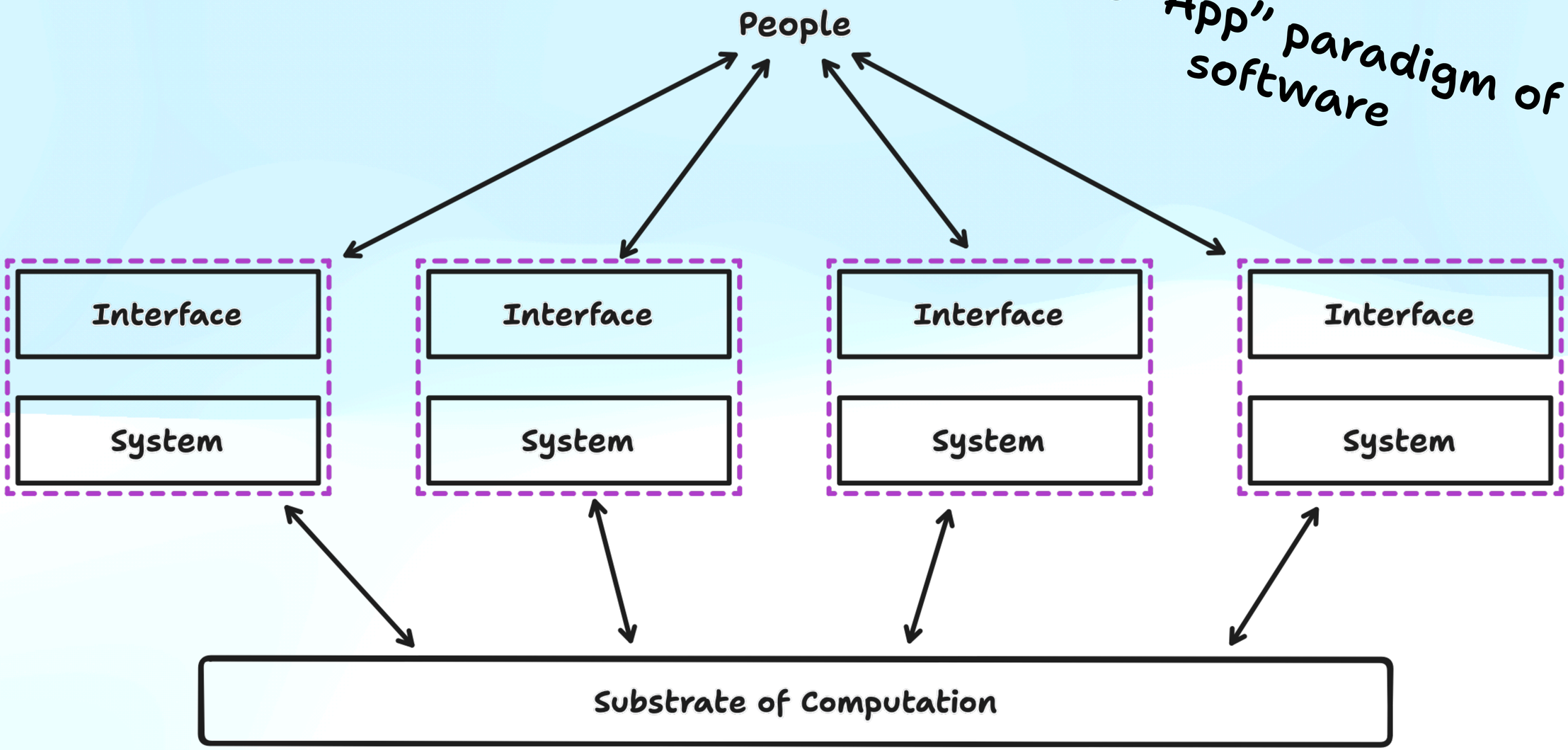
“Malleable software aims to increase the power of existing adaptation behaviors by allowing users to pull apart and re-combine their interfaces at the granularity of individual UI elements”

— Philip Tchernavskij. 2019. Designing and Programming Malleable Software.

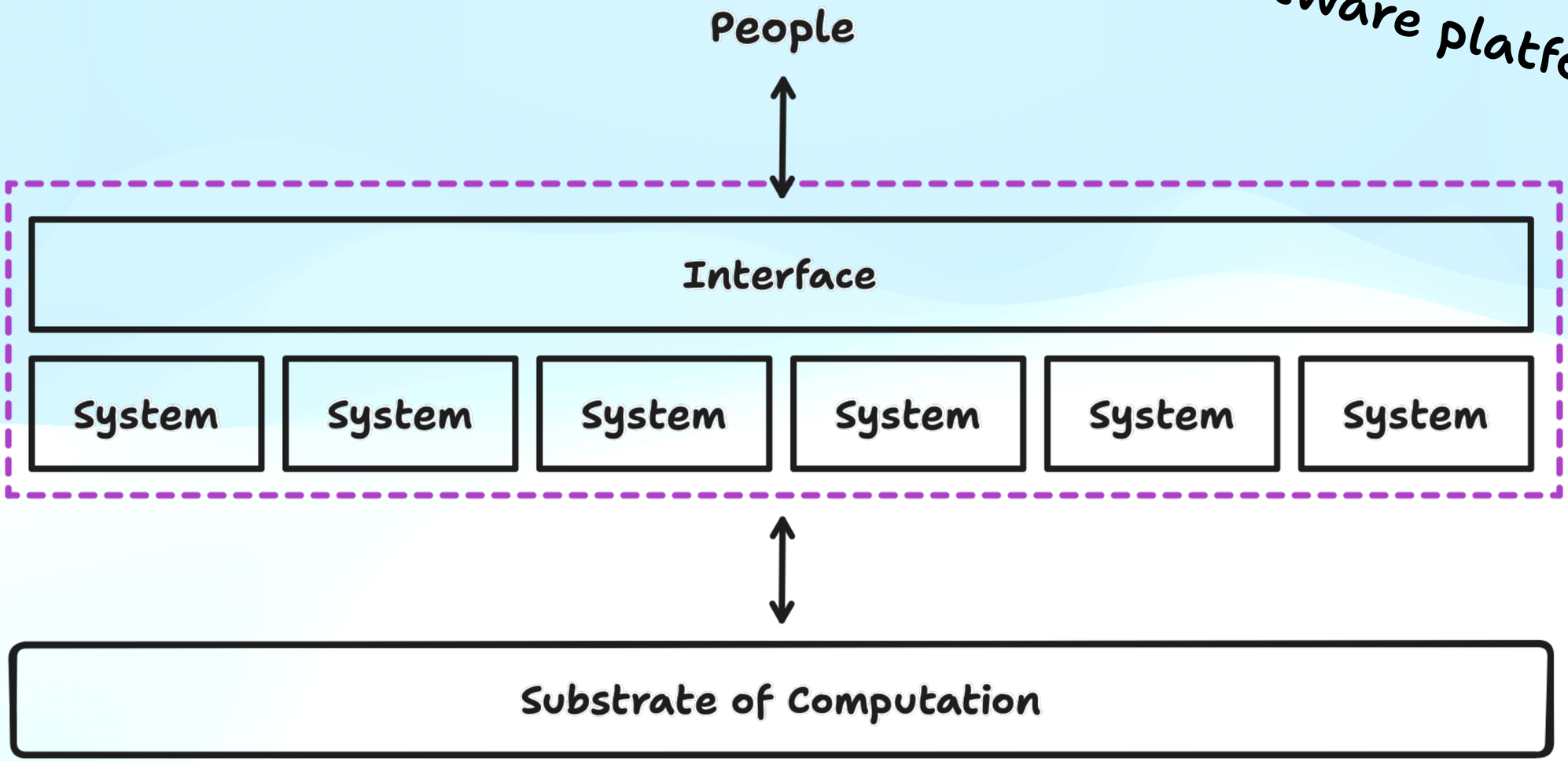
# Current interface production **incentivises a specific topology** of software



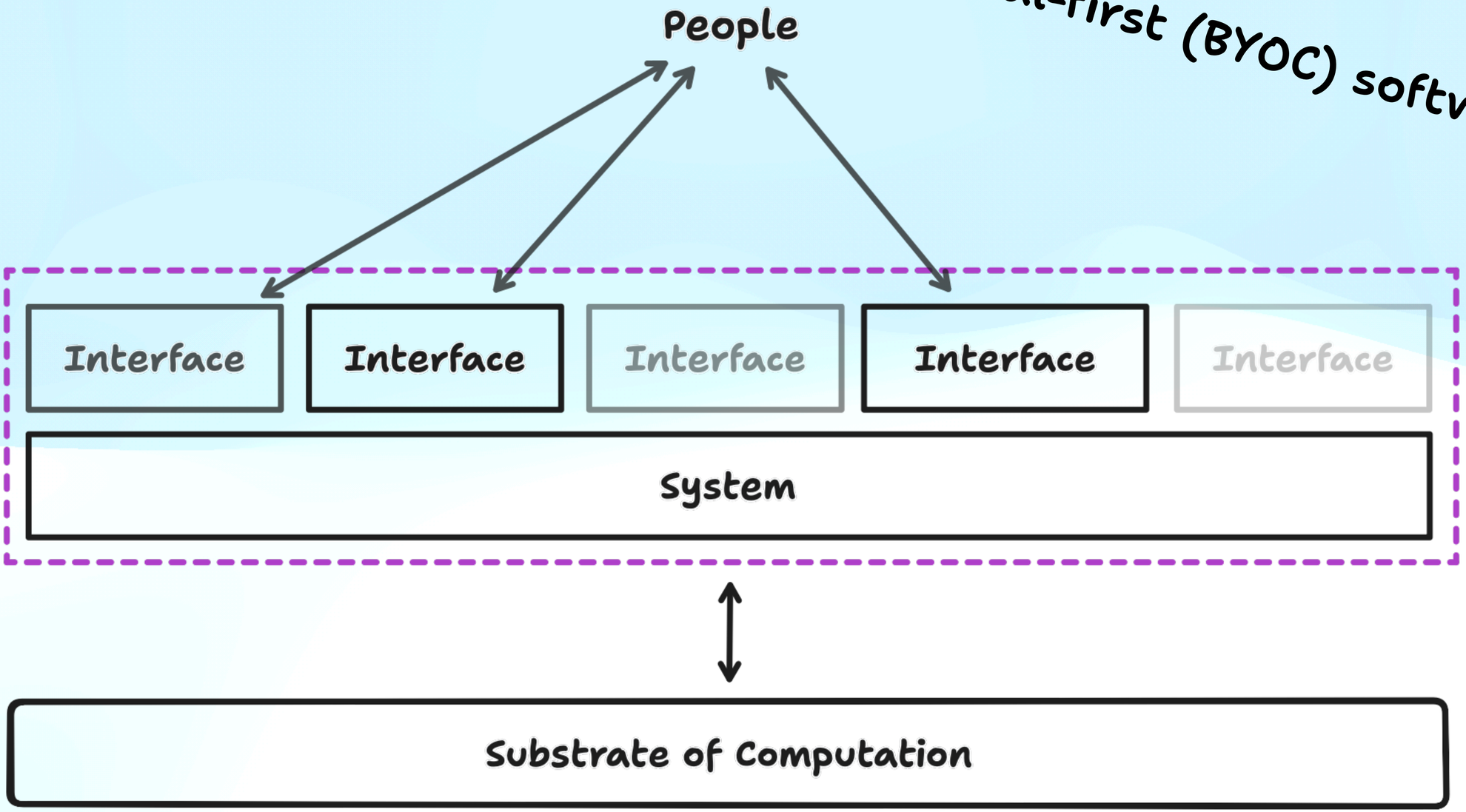
# The "App" paradigm of software

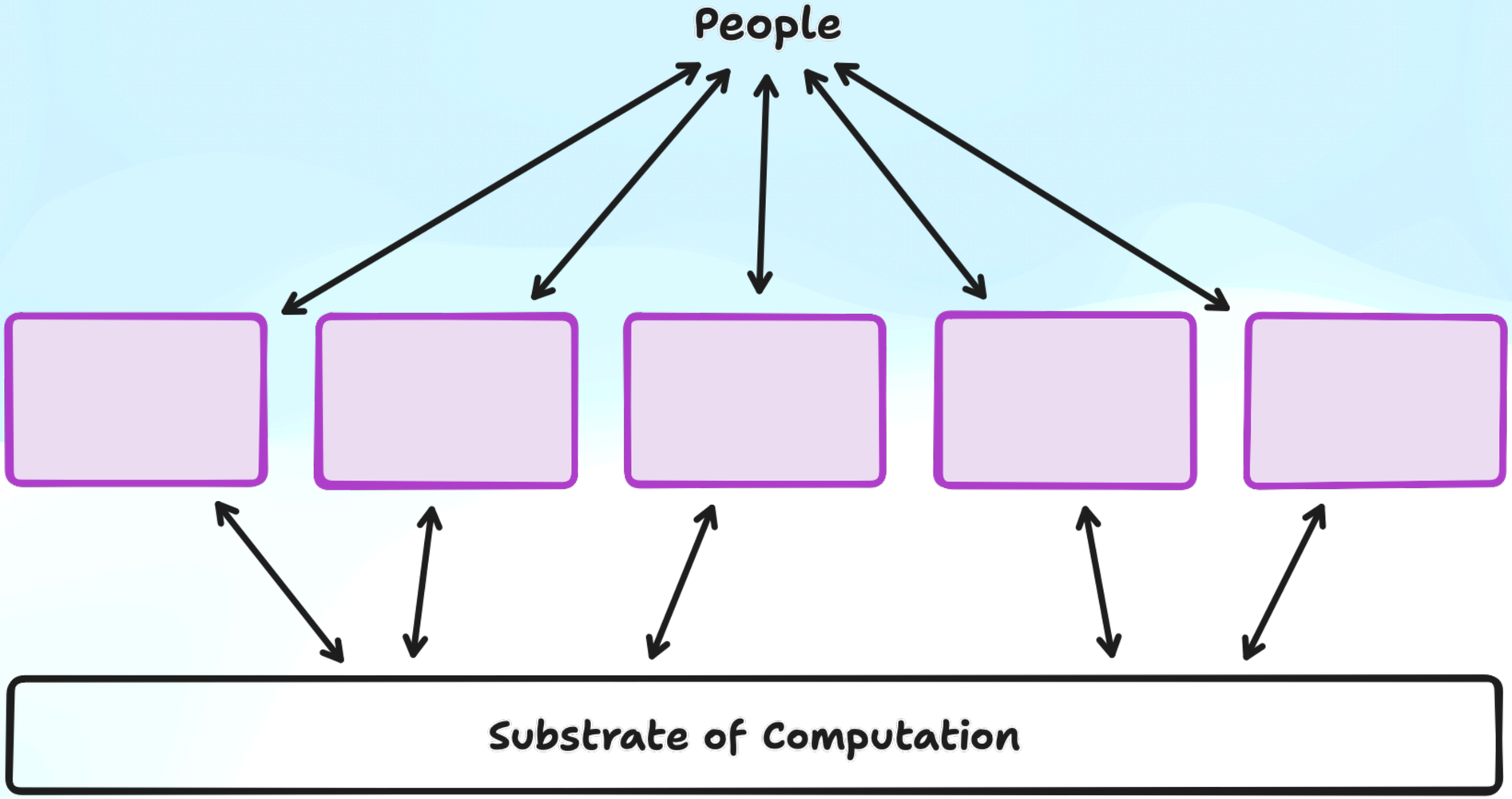


Cloud software platforms



Local-first (BYOC) software




People
Substrate of Computation

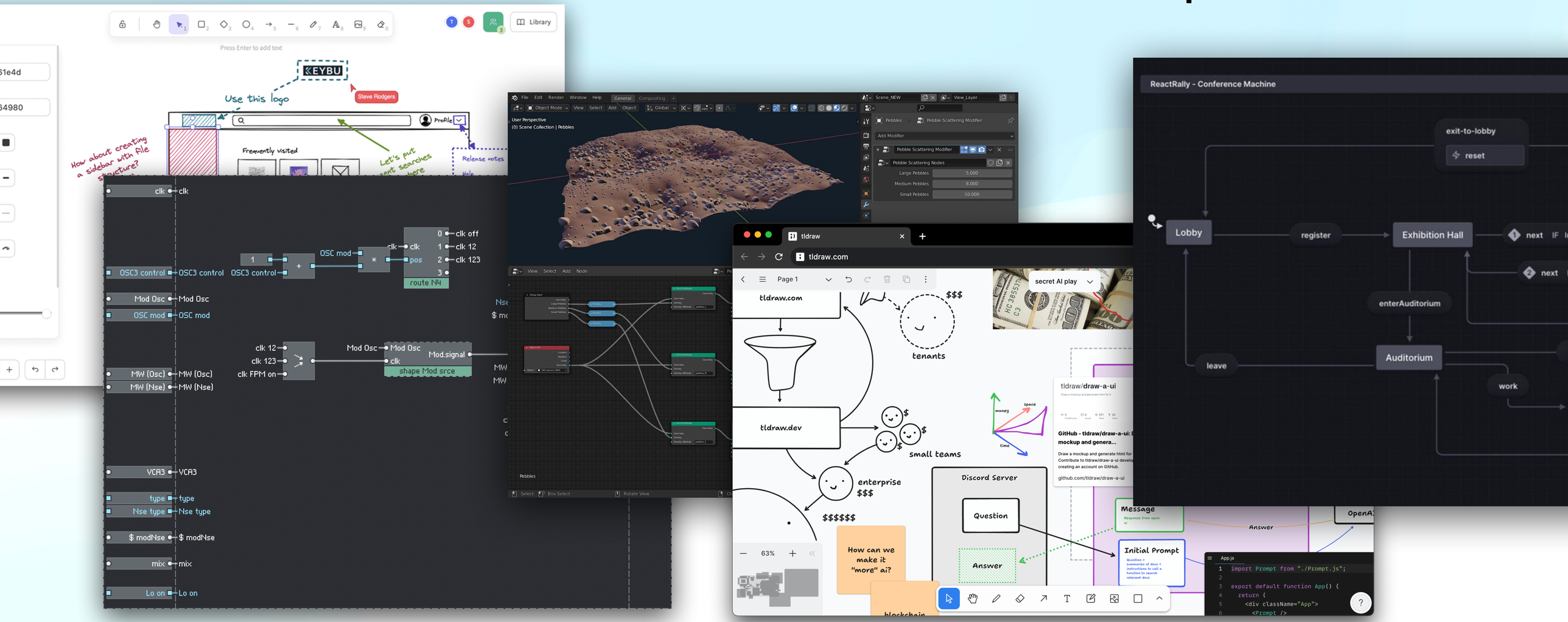
The topological **relationship** between interfaces and software is holding computing back.

To achieve the aspirations of malleable software we need **a new approach to interface production.**

# 2. Canvases & Information Substrates

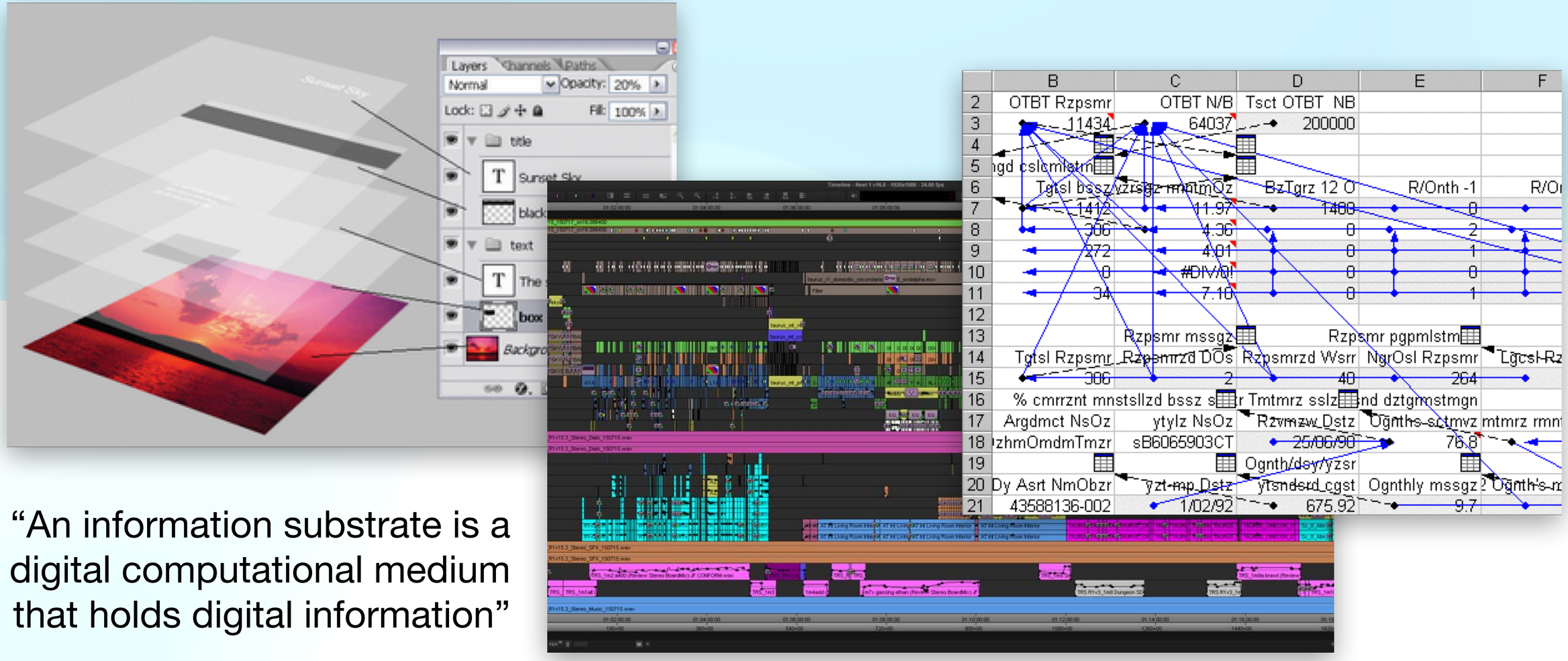
# What is a “Spatial Canvas”?

A **set of objects** with **positions** in a shared Euclidean space and **relationships** between them.




**Left to Right:** Excalidraw, Reaktor, Blender Nodes, TLDraw, Stately

# What is an “Information Substrate”?




“An information substrate is a digital computational medium that holds digital information”

— **Michel Beaudouin-Lafon.** 2017. Towards Unified Principles of Interaction

“Data does not exist in a vacuum. It is part of a substrate that provides context for **interpreting** data and **constraints** for **presenting** and **interacting** with it”

— Michel Beaudouin-Lafon. 2018. Information Substrates

**Canvases are not just tools**, they are powerful information substrates.

They define *constraints* on data, context for its *interpretation*, and a means of *presentation* and *interaction*.

# 3. Demo

# 4. An Integration Domain for HCI?

“In circuit design, engineers do not expect that their integrated circuits to be wired together directly with other ICs. Instead there is a whole vocabulary of glue components, including resistors and capacitors and small logic arrays, constituting a separate **integration domain**.”

— Stephen Kell. 2009. The Mythical Matched Modules

“An integration domain is simply a set of languages or tools for performing integration of software.”

— Stephen Kell. 2009. The Mythical Matched Modules

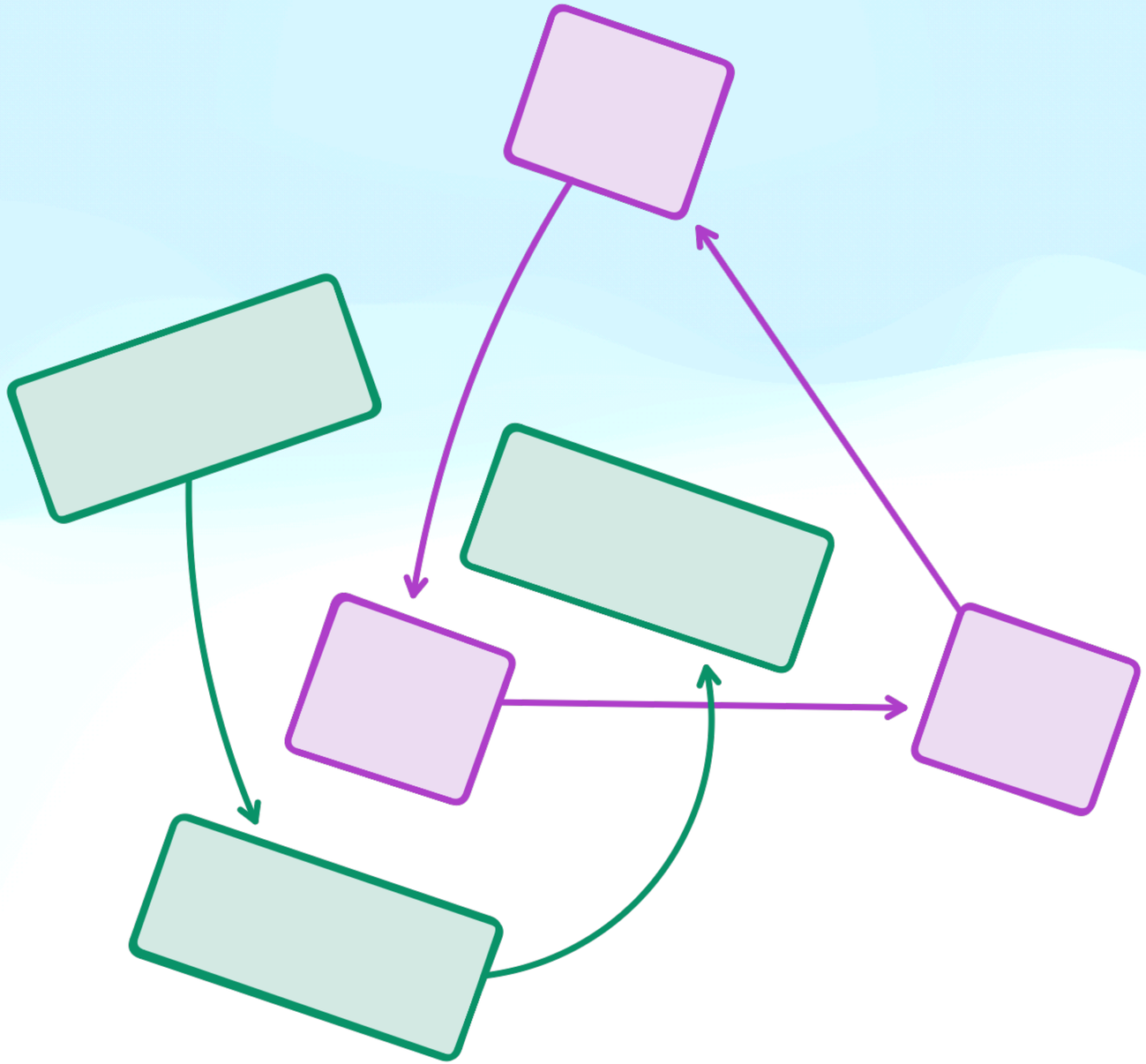
**Could the spatial canvas evolve** into an integration domain for interfaces and interaction?

A UX/UI integration substrate?

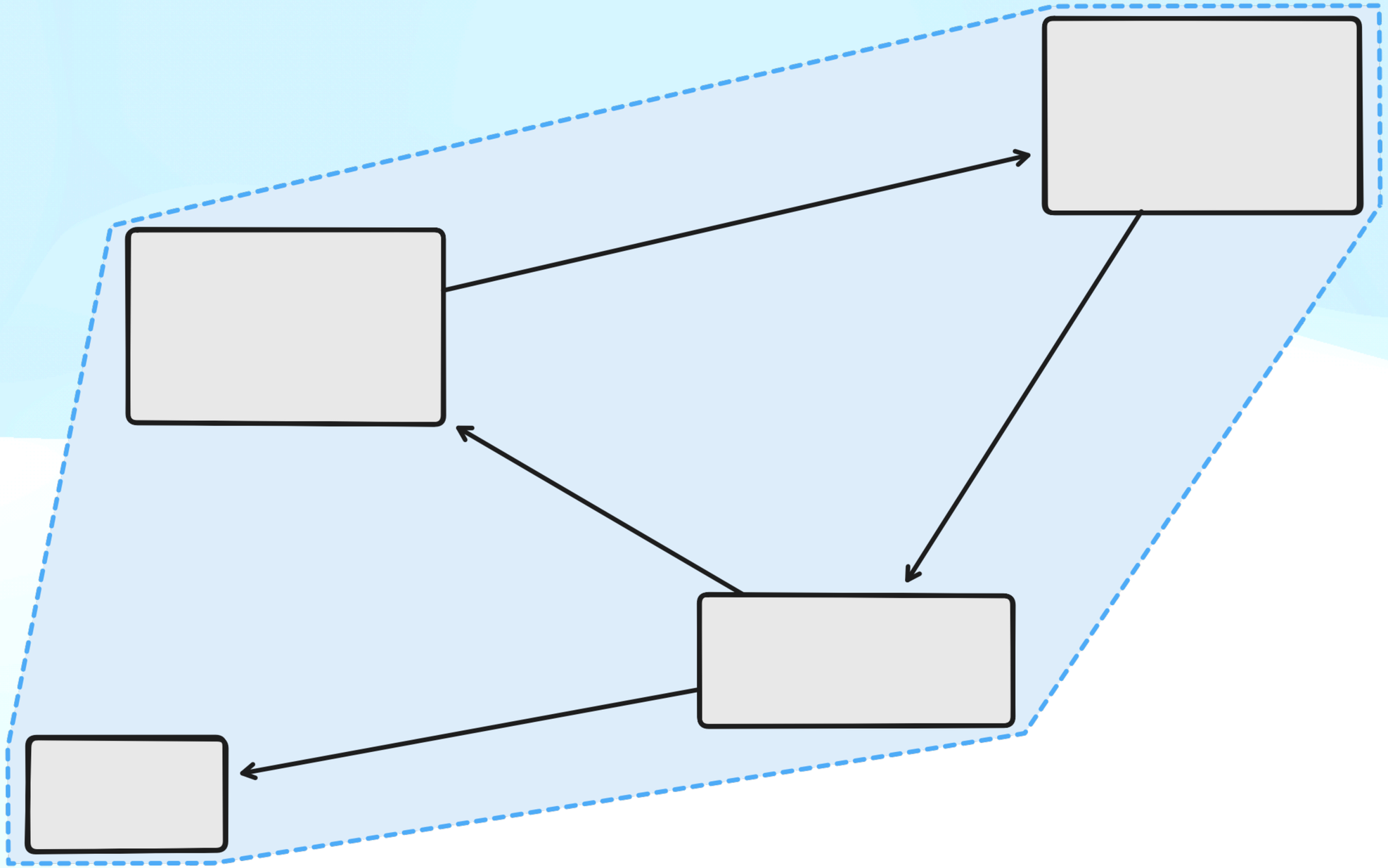
# How can we evolve the canvas?

1. **Extensibility without coordination:** Systems can extend the canvas without coordination of code
2. **Sets:** Objects can belong to distinct sets with semantics to ‘bind’ them to running systems
3. **Regions:** The uniform ‘empty space’ of a canvas can be divided into regions which take on local properties and behaviours
4. **Fields:** Beyond discrete objects and topologies, we need ways to define behaviour of the space itself
5. **Integration Semantics:** Expression of relationships between spaces, regions, fields, and the objects which occupy them.

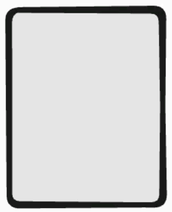
**Set Membership** allow co-existing objects to share semantics



**Regions** allow systems or behaviour to be bound to a portion of space



# **Fields** enable the space itself to have behaviour



## **Integration Semantics** enable interaction between interface ‘parts’

“A unifying medium [for integration] is clearly useful because it converts problem of size n2—mapping all languages to all other languages—into one apparently of size 2n.”

— Stephen Kell. 2009. The Mythical Matched Modules

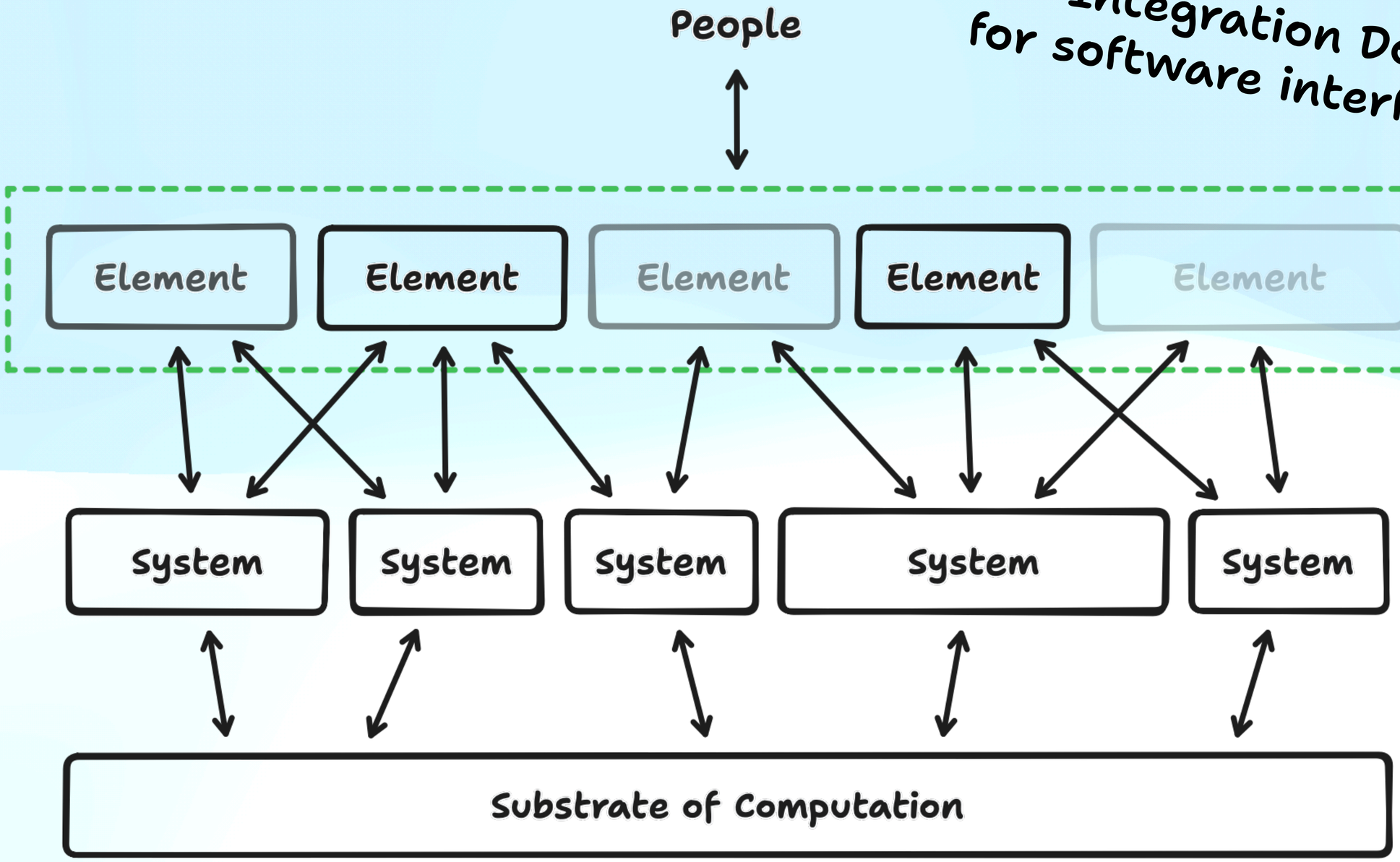
**Integration Semantics** enable interaction between interface ‘parts’

“Relations, not scripts or circuits”

**Canvases could be to HCI what Haskell is to Type Systems research:**

Many interfaces and interactions co-existing in the same shared environment

An "Integration Domain" for software interfaces?



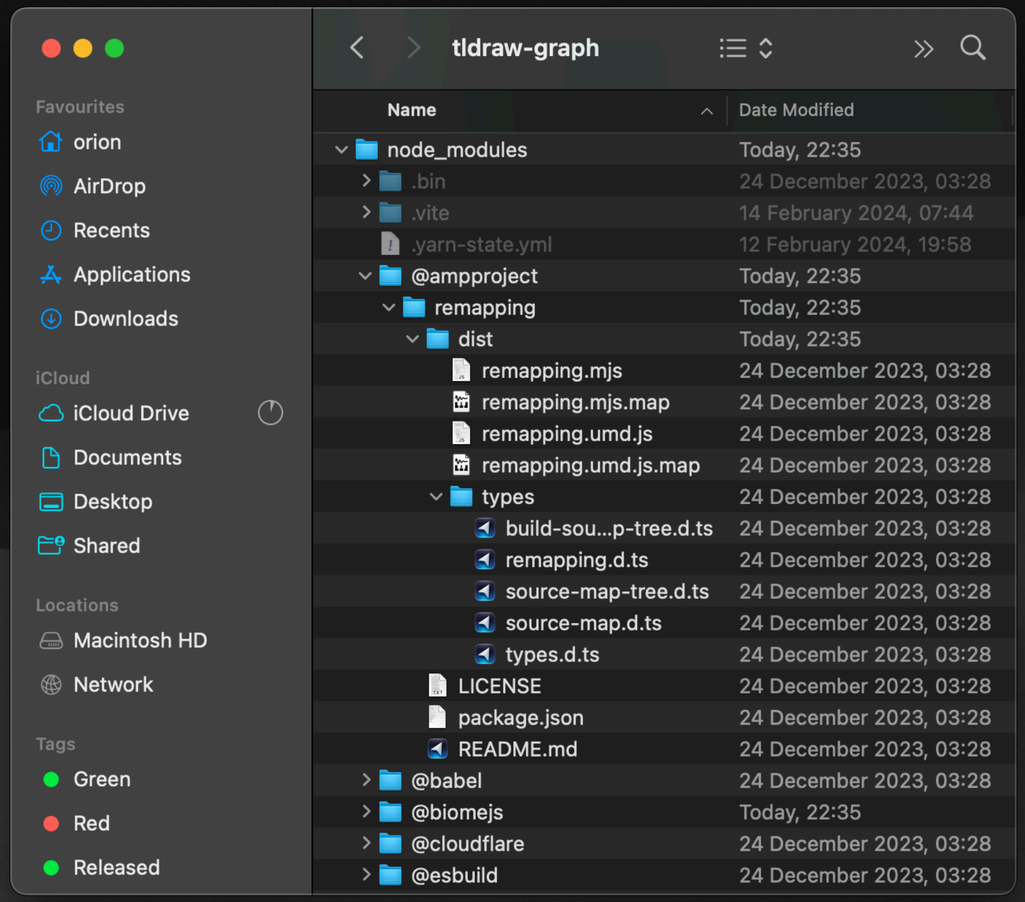
In Summary:

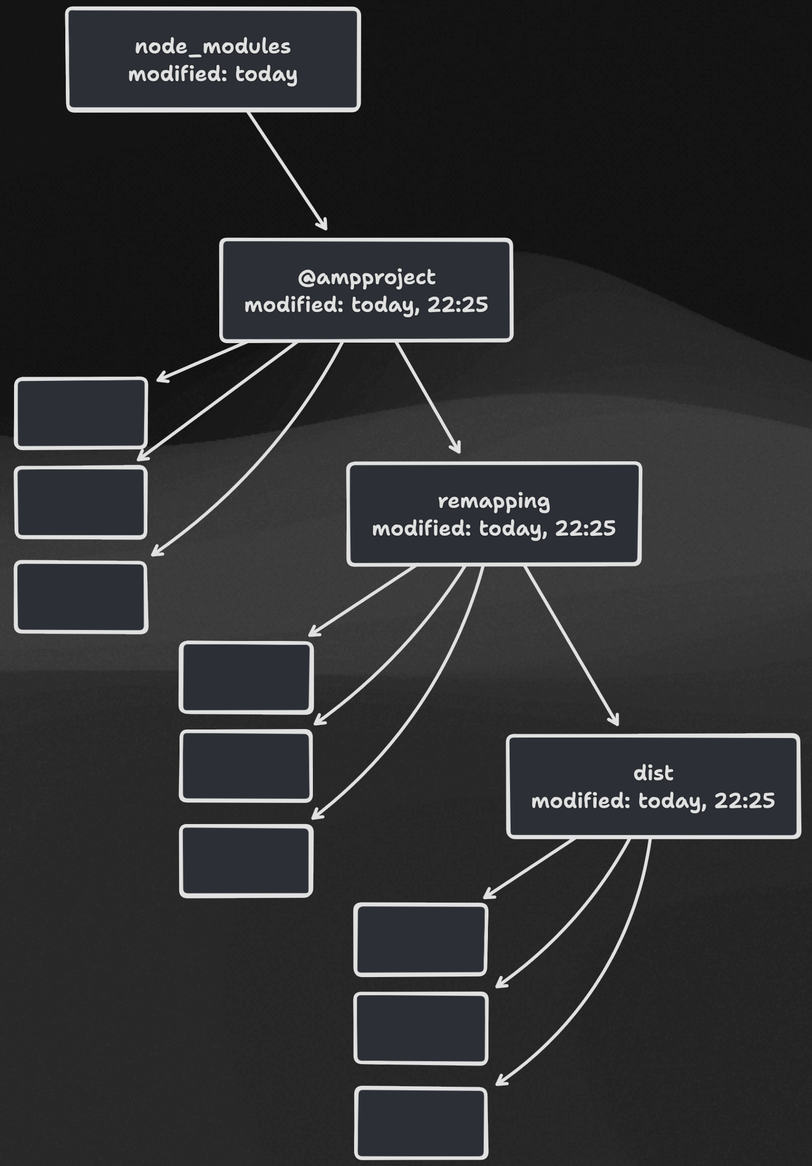
Developing an Integration Domain for HCI can enable **new software topologies** which are more **malleable, flexible, pluralistic** and ***fun.***

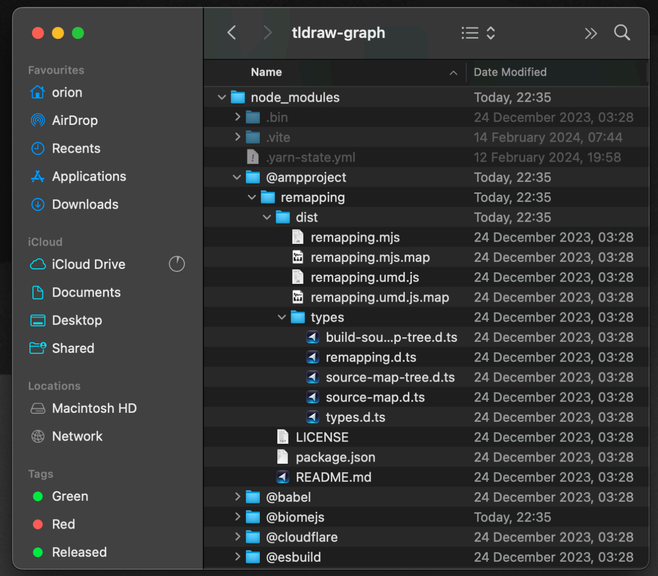
I believe spatial canvases are **particularly well suited** to be a foundation for this integration domain.

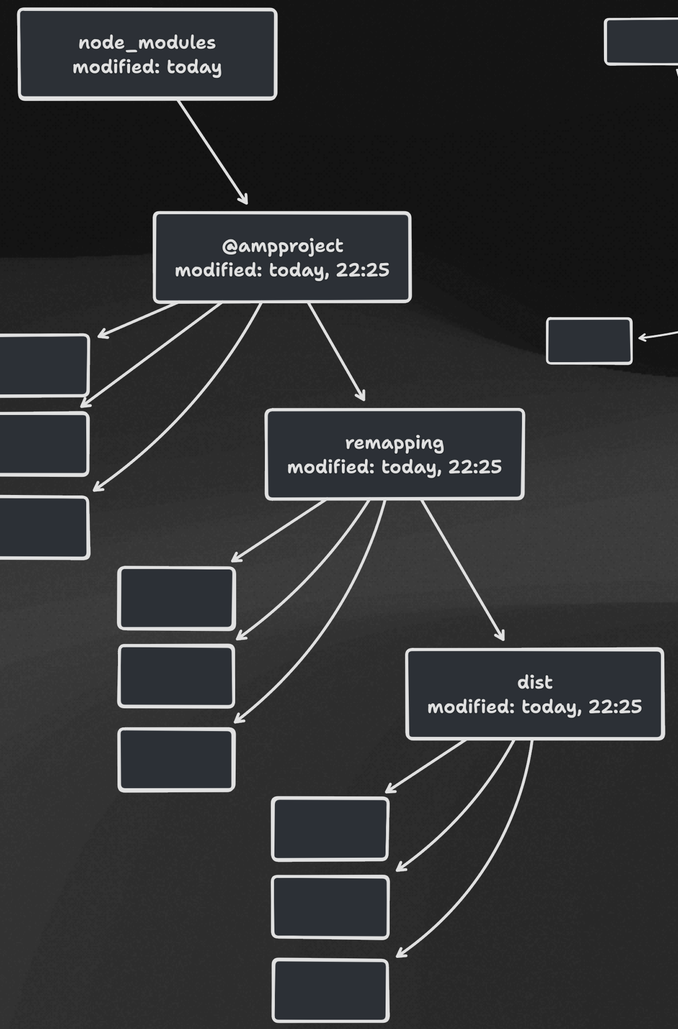
# 5. One Fun Idea

If you squint, a lot of interfaces start to appear equivalent

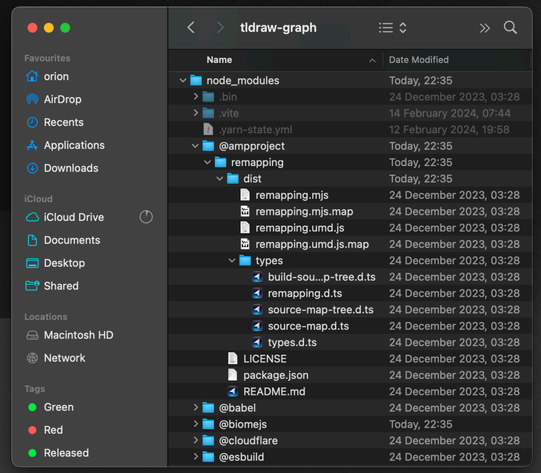

tldraw-graph
Favourites
orion
AirDrop
Recents
Applications
Downloads
iCloud
iCloud Drive
Documents
Desktop
Shared
Locations
Macintosh HD
Network
Tags
Green
Red
Released
Name
Date Modified
node_modules Today, 22:35
.bin 24 December 2023, 03:28
.vite 14 February 2024, 07:44
.yarn-state.yml 12 February 2024, 19:58
@ampproject Today, 22:35
remapping Today, 22:35
dist Today, 22:35
remapping.mjs 24 December 2023, 03:28
remapping.mjs.map 24 December 2023, 03:28
remapping.umd.js 24 December 2023, 03:28
remapping.umd.js.map 24 December 2023, 03:28
types 24 December 2023, 03:28
build-sou...p-tree.d.ts 24 December 2023, 03:28
remapping.d.ts 24 December 2023, 03:28
source-map-tree.d.ts 24 December 2023, 03:28
source-map.d.ts 24 December 2023, 03:28
types.d.ts 24 December 2023, 03:28
LICENSE 24 December 2023, 03:28
package.json 24 December 2023, 03:28
README.md 24 December 2023, 03:28
@babel 24 December 2023, 03:28
@biomejs Today, 22:35
@cloudflare 24 December 2023, 03:28
@esbuild 24 December 2023, 03:28

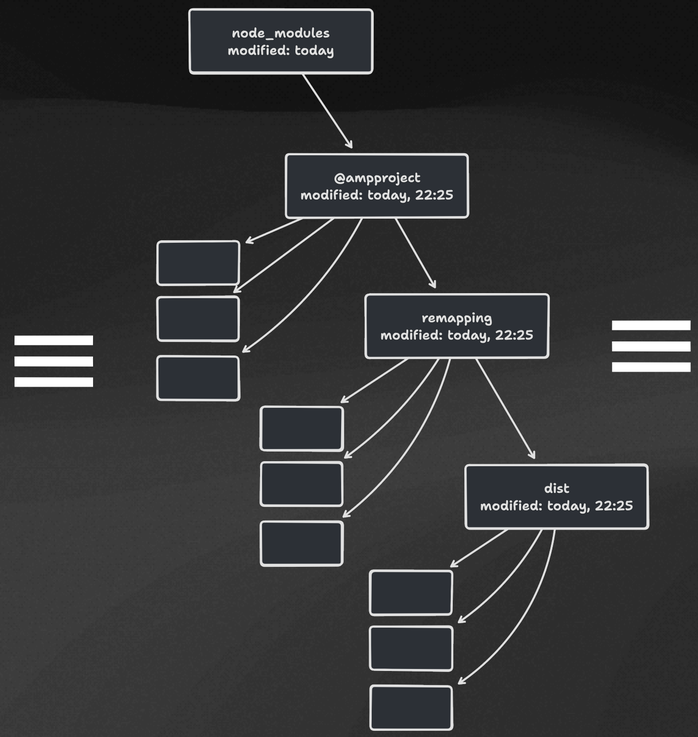

node_modules
modified: today
@ampproject
modified: today, 22:25
remapping
modified: today, 22:25
dist
modified: today, 22:25


tldraw-graph
Favourites
orion
AirDrop
Recents
Applications
Downloads
iCloud
iCloud Drive
Documents
Desktop
Shared
Locations
Macintosh HD
Network
Tags
Green
Red
Released
Name
Date Modified
node_modules Today, 22:35
.bin 24 December 2023, 03:28
.vite 14 February 2024, 07:44
.yarn-state.yml 12 February 2024, 19:58
@ampproject Today, 22:35
remapping Today, 22:35
dist Today, 22:35
remapping.mjs 24 December 2023, 03:28
remapping.mjs.map 24 December 2023, 03:28
remapping.umd.js 24 December 2023, 03:28
remapping.umd.js.map 24 December 2023, 03:28
types 24 December 2023, 03:28
build-sou...p-tree.d.ts 24 December 2023, 03:28
remapping.d.ts 24 December 2023, 03:28
source-map-tree.d.ts 24 December 2023, 03:28
source-map.d.ts 24 December 2023, 03:28
types.d.ts 24 December 2023, 03:28
LICENSE 24 December 2023, 03:28
package.json 24 December 2023, 03:28
README.md 24 December 2023, 03:28
@babel 24 December 2023, 03:28
@biomejs Today, 22:35
@cloudflare 24 December 2023, 03:28
@esbuild 24 December 2023, 03:28

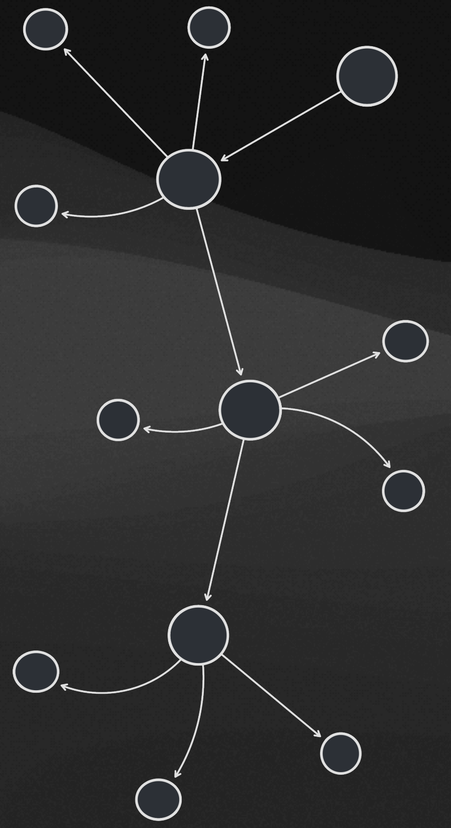

node_modules
modified: today
@ampproject
modified: today, 22:25
remapping
modified: today, 22:25
dist
modified: today, 22:25


node_modules
modified: today
@ampproject
modified: today, 22:25
remapping
modified: today, 22:25
dist
modified: today, 22:25


tldraw-graph
Favourites
orion
AirDrop
Recents
Applications
Downloads
iCloud
iCloud Drive
Documents
Desktop
Shared
Locations
Macintosh HD
Network
Tags
Green
Red
Released
Name
Date Modified
node_modules Today, 22:35
.bin 24 December 2023, 03:28
.vite 14 February 2024, 07:44
.yarn-state.yml 12 February 2024, 19:58
@ampproject Today, 22:35
remapping Today, 22:35
dist Today, 22:35
remapping.mjs 24 December 2023, 03:28
remapping.mjs.map 24 December 2023, 03:28
remapping.umd.js 24 December 2023, 03:28
remapping.umd.js.map 24 December 2023, 03:28
types 24 December 2023, 03:28
build-sou...p-tree.d.ts 24 December 2023, 03:28
remapping.d.ts 24 December 2023, 03:28
source-map-tree.d.ts 24 December 2023, 03:28
source-map.d.ts 24 December 2023, 03:28
types.d.ts 24 December 2023, 03:28
LICENSE 24 December 2023, 03:28
package.json 24 December 2023, 03:28
README.md 24 December 2023, 03:28
@babel 24 December 2023, 03:28
@biomejs Today, 22:35
@cloudflare 24 December 2023, 03:28
@esbuild 24 December 2023, 03:28


node_modules
modified: today
@ampproject
modified: today, 22:25
remapping
modified: today, 22:25
dist
modified: today, 22:25

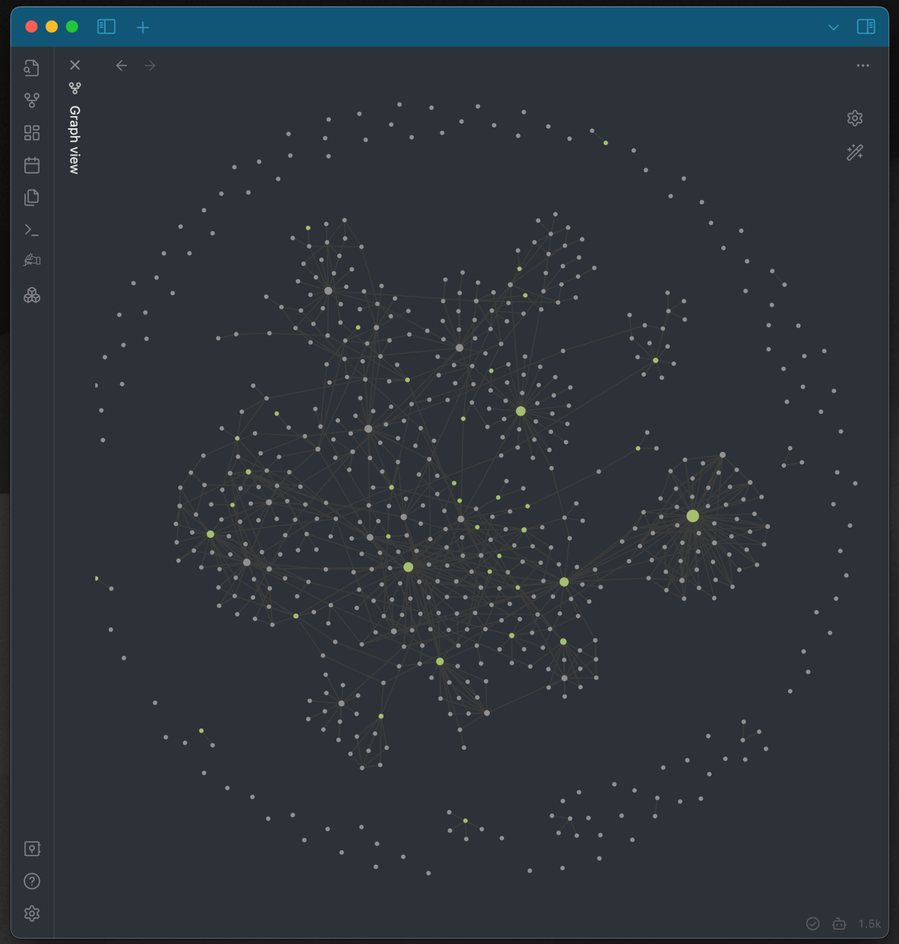

node_modules
modified: today
@ampproject
modified: today, 22:25
remapping
modified: today, 22:25
dist
modified: today, 22:25

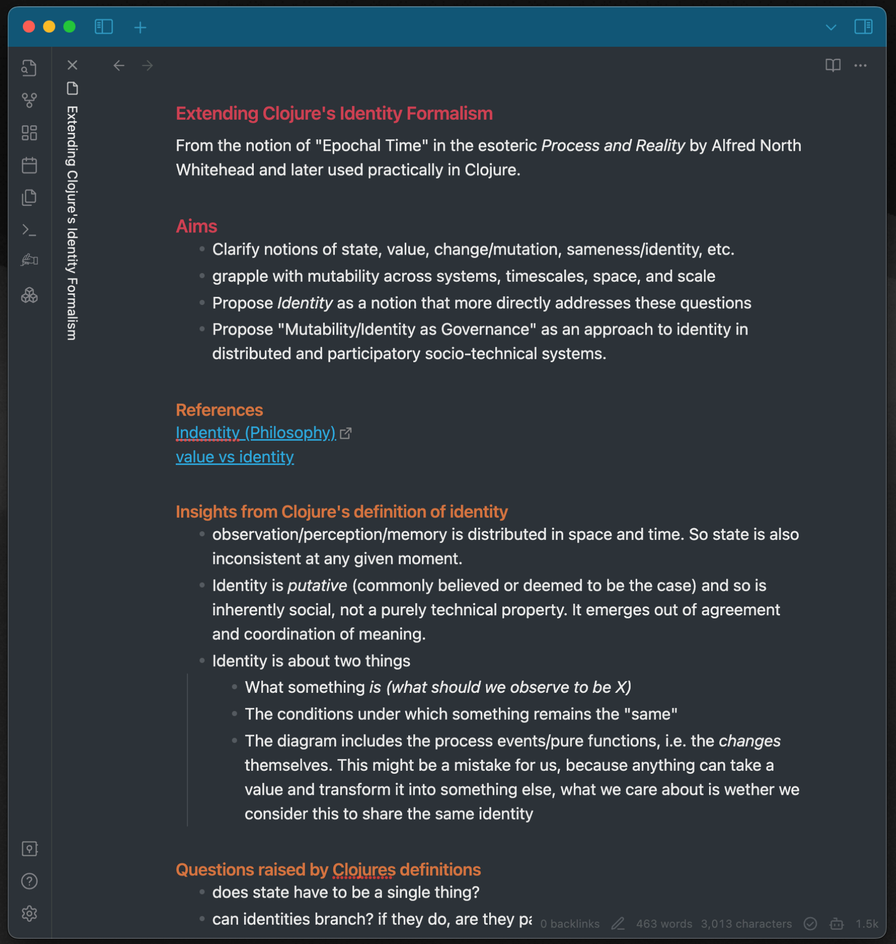


# Must these views be discrete?

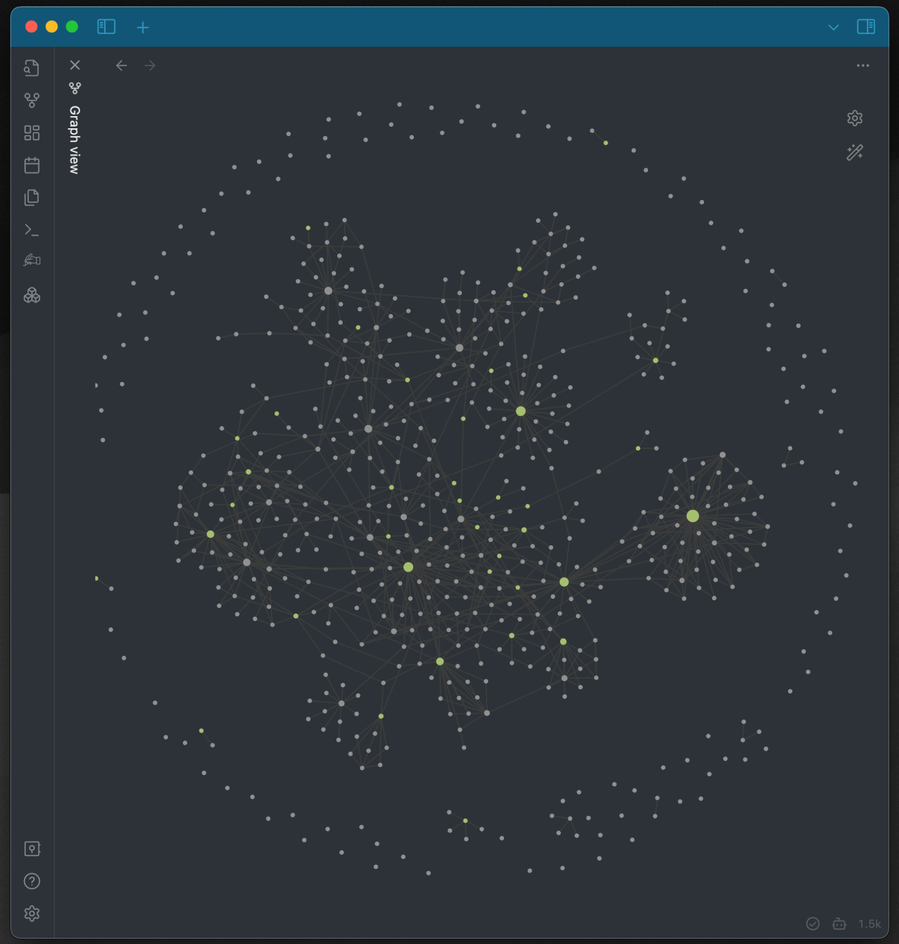
Credit: Andrew Blinn

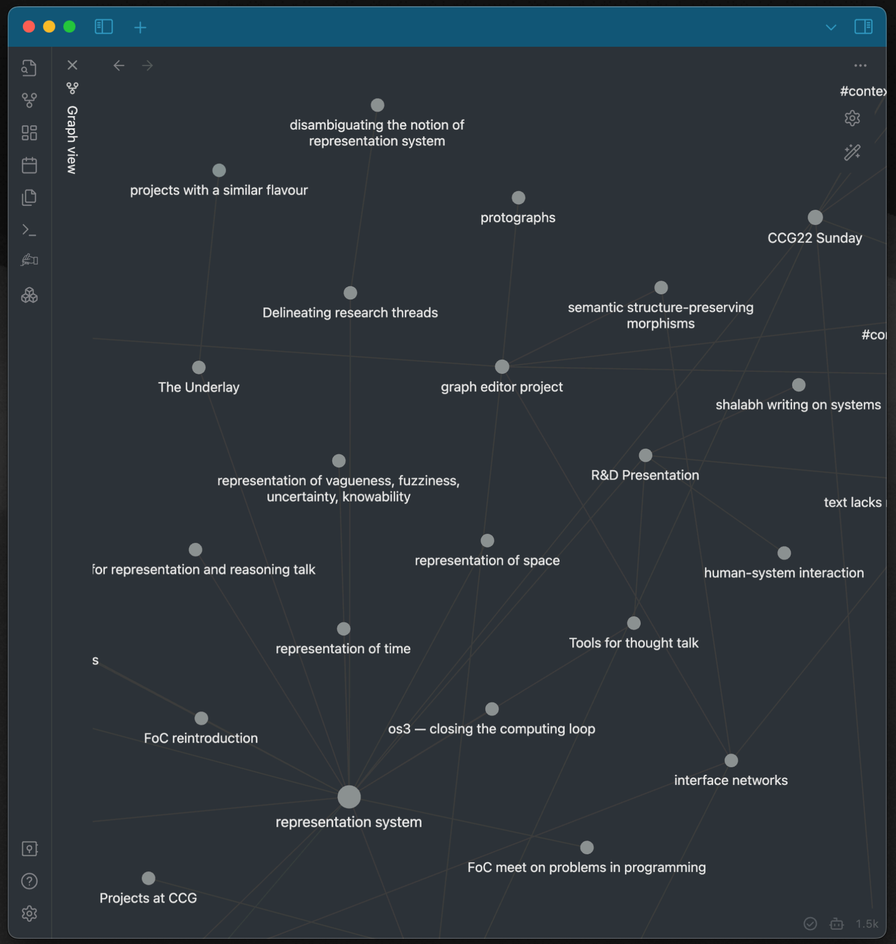
# Some view changes are **discrete**



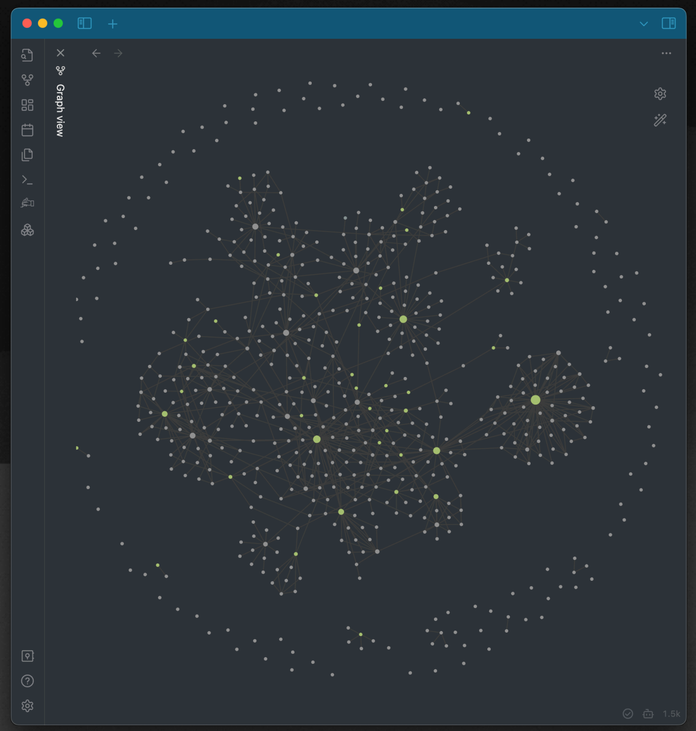



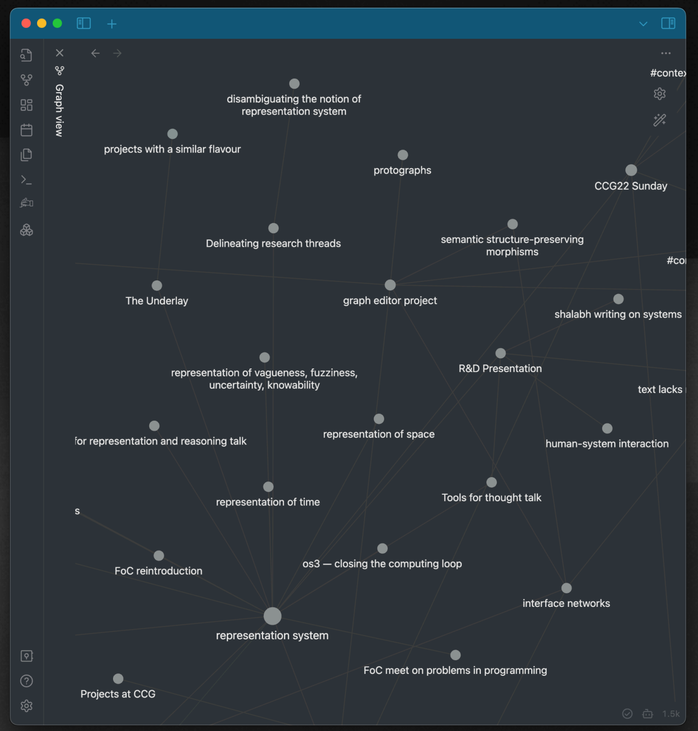
# Some view changes are **continuous**

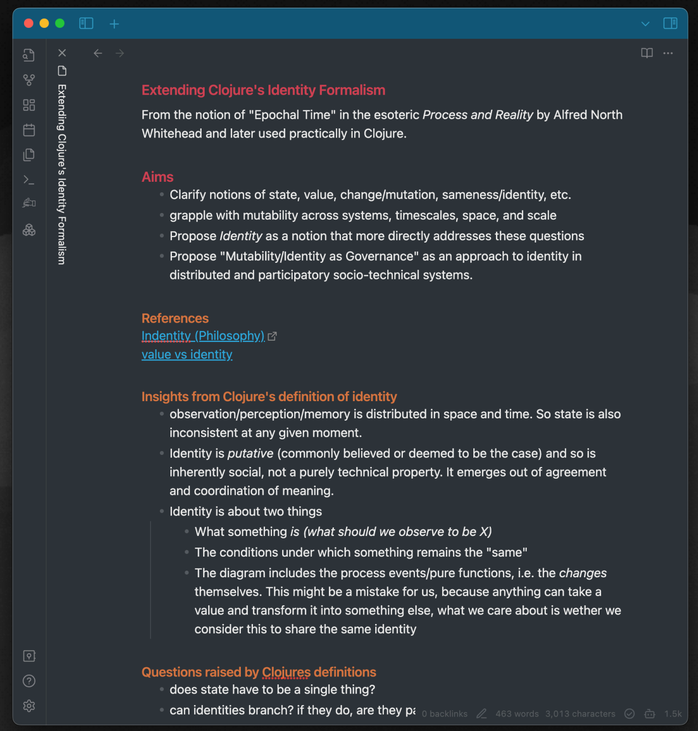


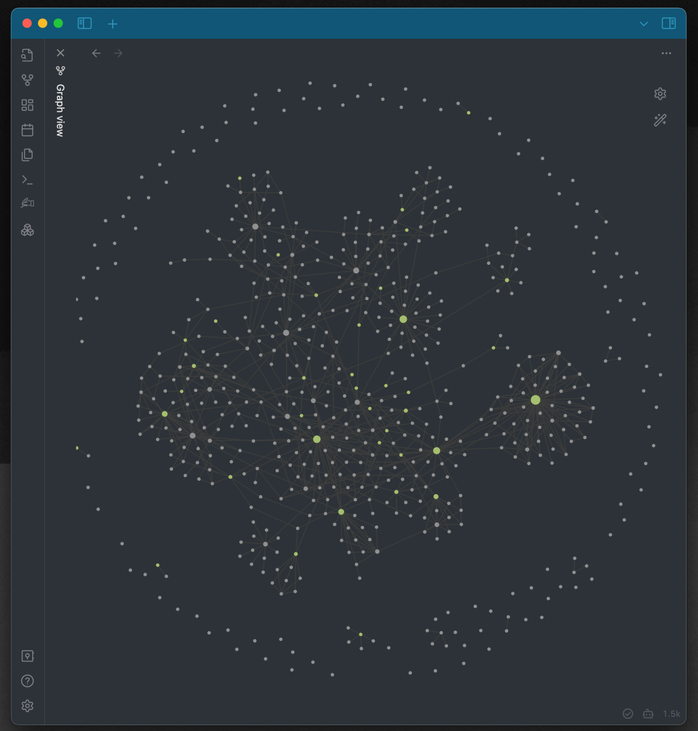



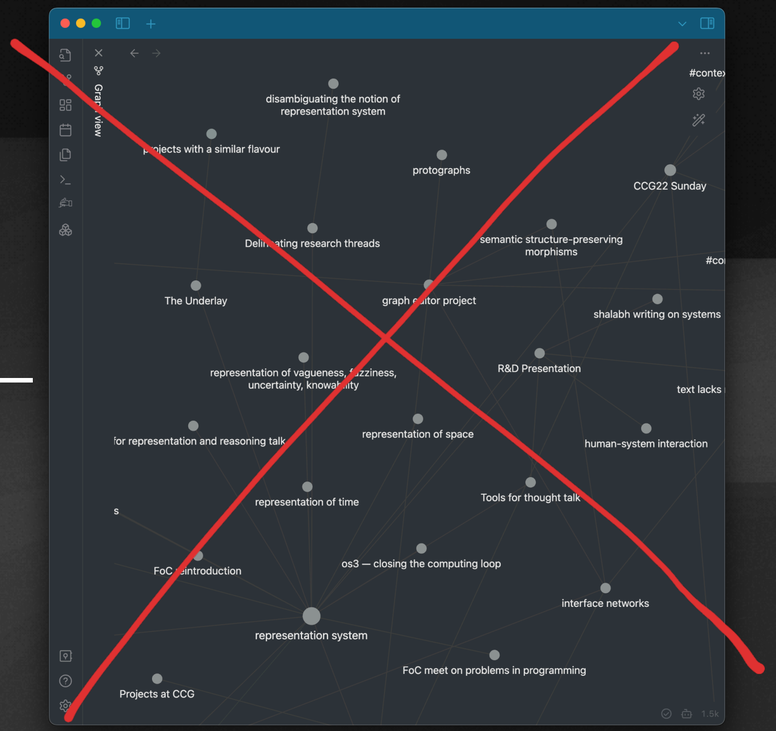
# What are the limits of continuous view transformations?

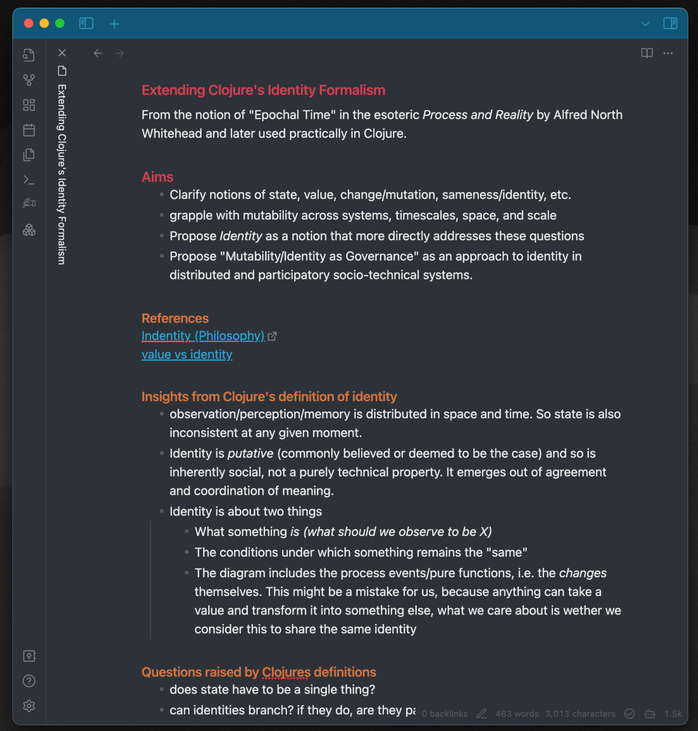


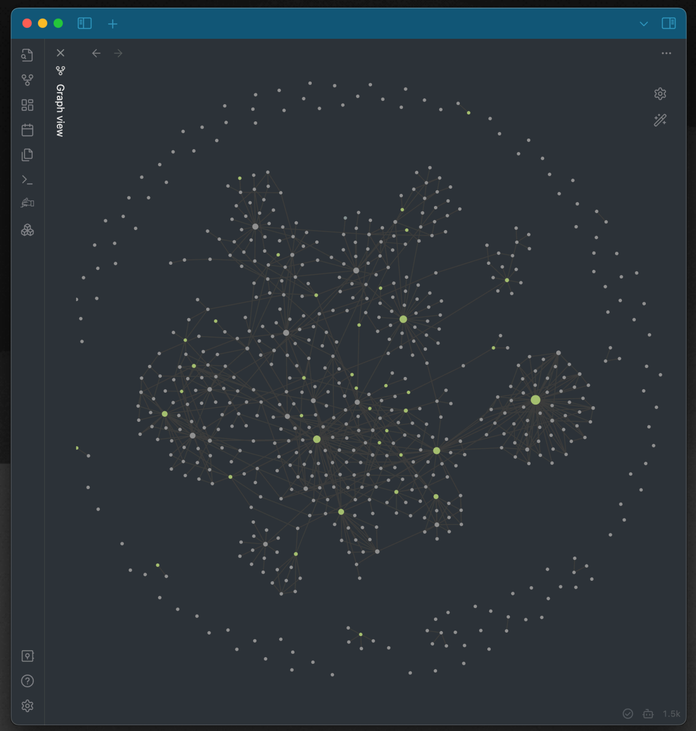




Graph view


Graph view
disambiguating the notion of representation system
protographs
CCG22 Sunday
Delimiting research threads
semantic structure-preserving morphisms
The Underlay
shalabh writing on systems
R&D Presentation
representation of space
human-system interaction
representation of time
Tools for thought talk
os3 — closing the computing loop
interface networks
representation system
FoC meet on problems in programming
Projects at CCG


Extending Clojure's Identity Formalism
Extending Clojure's Identity Formalism
From the notion of "Epochal Time" in the esoteric *Process and Reality* by Alfred North Whitehead and later used practically in Clojure.
Aims
Clarify notions of state, value, change/mutation, sameness/identity, etc.
grapple with mutability across systems, timescales, space, and scale
Propose *Identity* as a notion that more directly addresses these questions
Propose "Mutability/Identity as Governance" as an approach to identity in distributed and participatory socio-technical systems.
References
Indentity (Philosophy)
value vs identity
Insights from Clojure's definition of identity
observation/perception/memory is distributed in space and time. So state is also inconsistent at any given moment.
Identity is *putative* (commonly believed or deemed to be the case) and so is inherently social, not a purely technical property. It emerges out of agreement and coordination of meaning.
Identity is about two things
What something *is (what should we observe to be X)*
The conditions under which something remains the "same"
The diagram includes the process events/pure functions, i.e. the *changes* themselves. This might be a mistake for us, because anything can take a value and transform it into something else, what we care about is wether we consider this to share the same identity
Questions raised by Clojures definitions
does state have to be a single thing?
can identities branch? if they do, are they pa
0 backlinks
463 words
3,013 characters




???

# Extending Clojure's Identity Formalism

From the notion of "Epochal Time" in the esoteric *Process and Reality* by Alfred North Whitehead and later used practically in Clojure.

## Aims

- Clarify notions of state, value, change/mutation, sameness/identity, etc.
- grapple with mutability across systems, timescales, space, and scale
- Propose *Identity* as a notion that more directly addresses these questions
- Propose "Mutability/Identity as Governance" as an approach to identity in distributed and participatory socio-technical systems.

## References

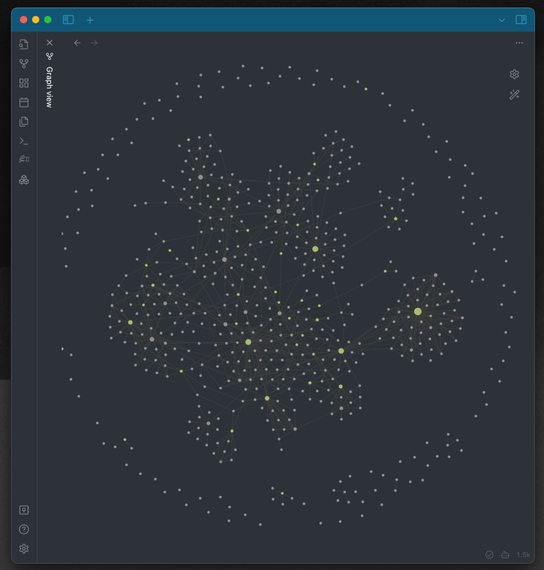
Indentity (Philosophy)
value vs identity

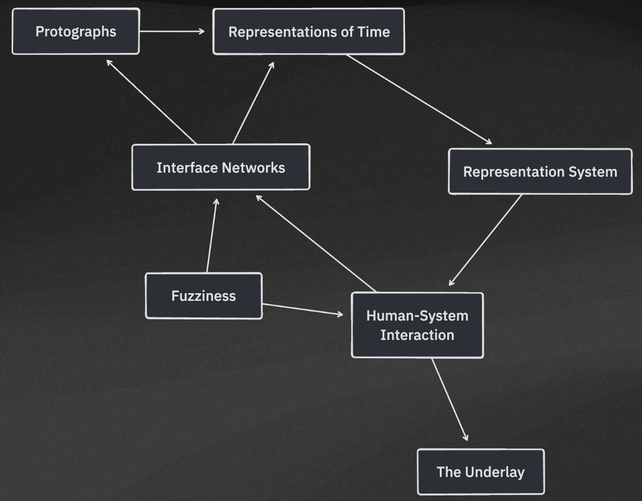
## Insights from Clojure's definition of identity

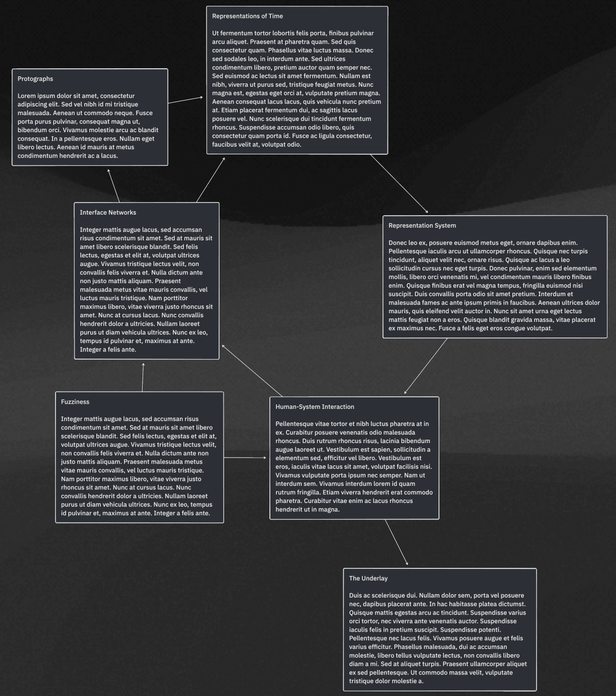
- observation/perception/memory is distributed in space and time. So state is also inconsistent at any given moment.
- Identity is *putative* (commonly believed or deemed to be the case) and so is inherently social, not a purely technical property. It emerges out of agreement and coordination of meaning.
- Identity is about two things
  - What something *is (what should we observe to be X)*
  - The conditions under which something remains the "same"
  - The diagram includes the process events/pure functions, i.e. the *changes* themselves. This might be a mistake for us, because anything can take a value and transform it into something else, what we care about is wether we consider this to share the same identity

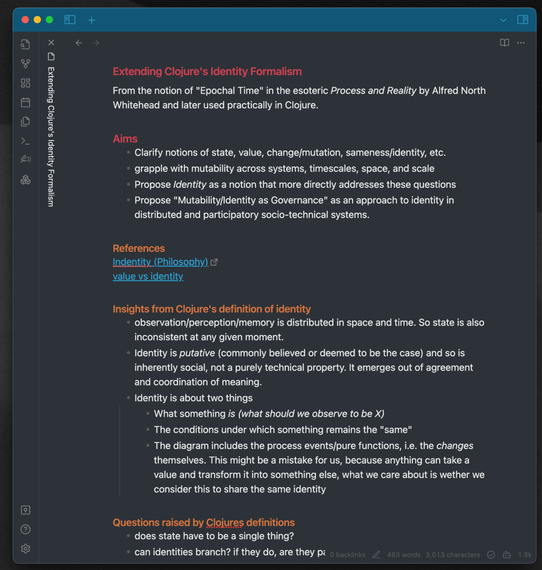
## Questions raised by Clojures definitions

- does state have to be a single thing?
- can identities branch? if they do, are they pa



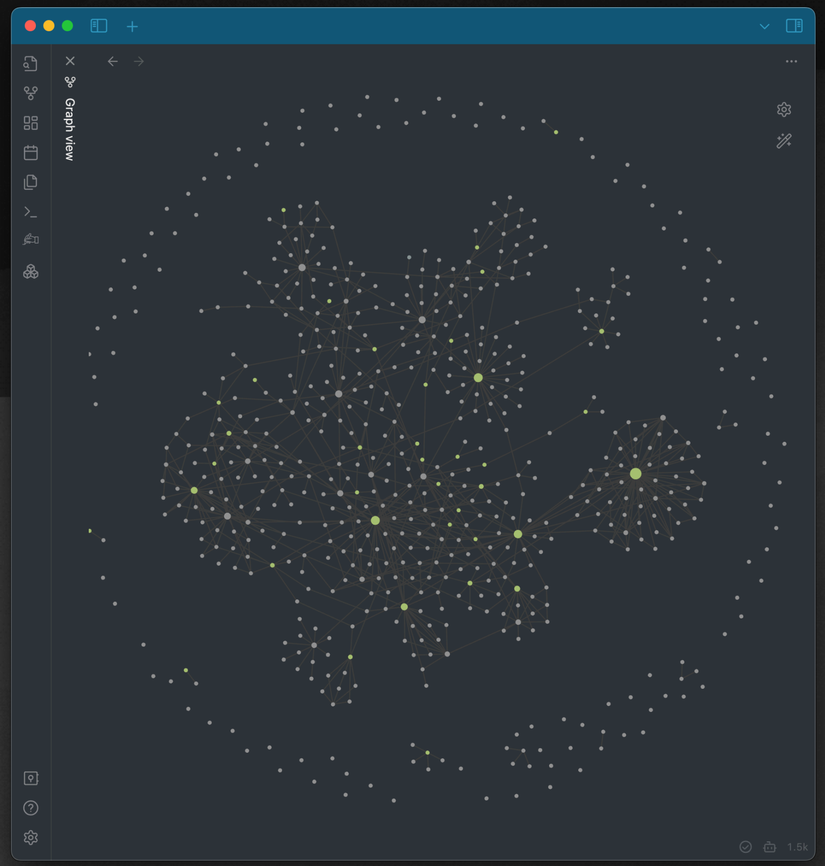



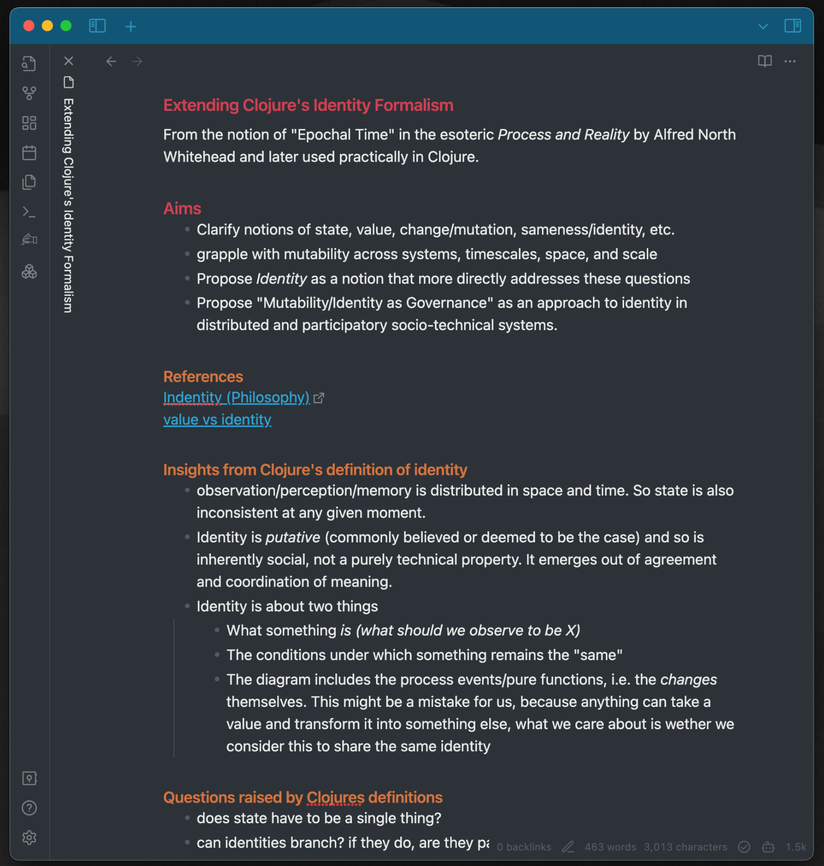




| | | | | |
|---|---|---|---|---|
| Objects: | Discrete Symbols | + Names | + Body Text | — |
| Relations: | Link Topology | + Containment | — | — |
| Interaction: | — | + Renaming | + Body Text Editing | + View Scrolling |

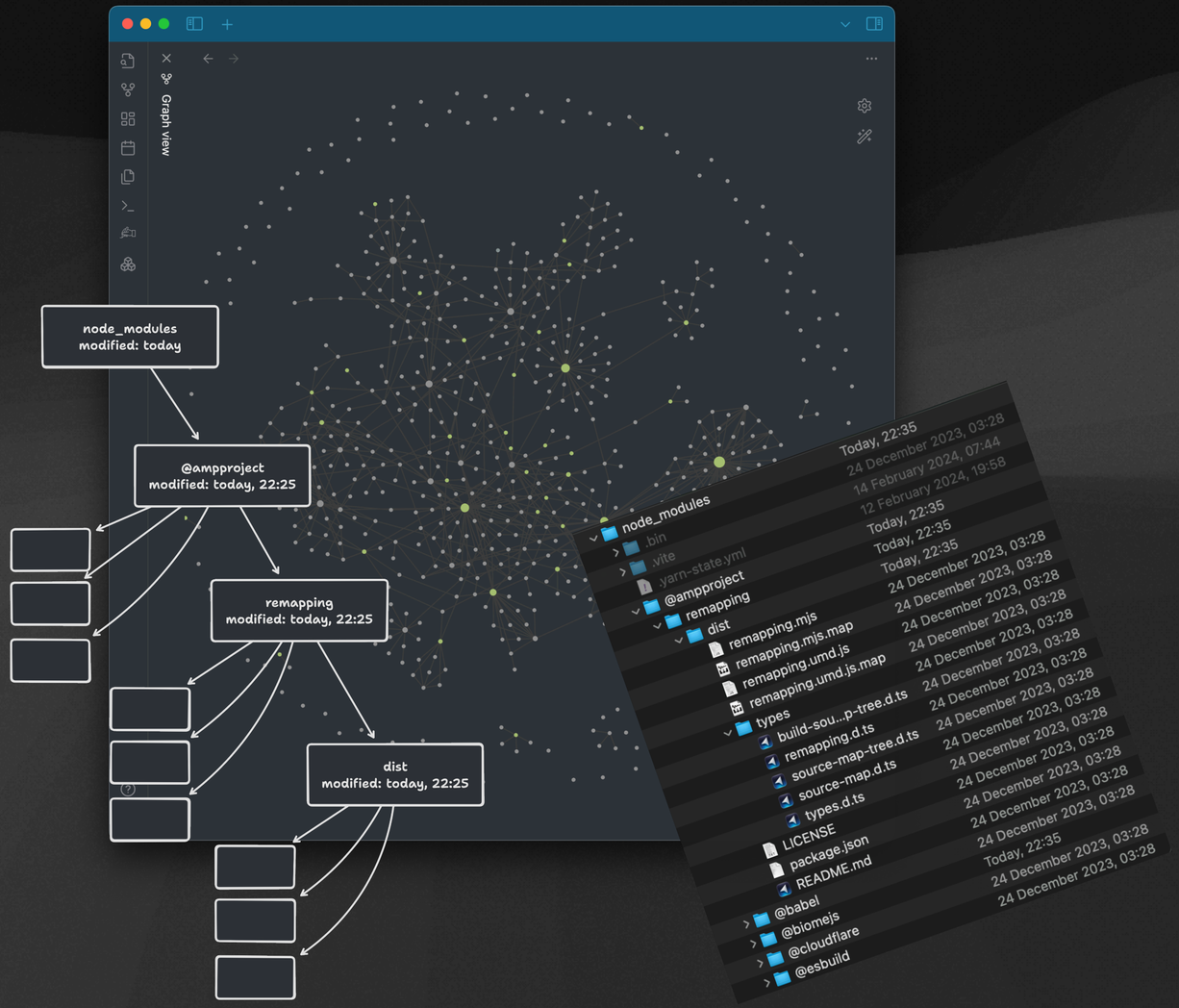
# Can we formalise a kind of ‘Visual-Semantic Morphism’?



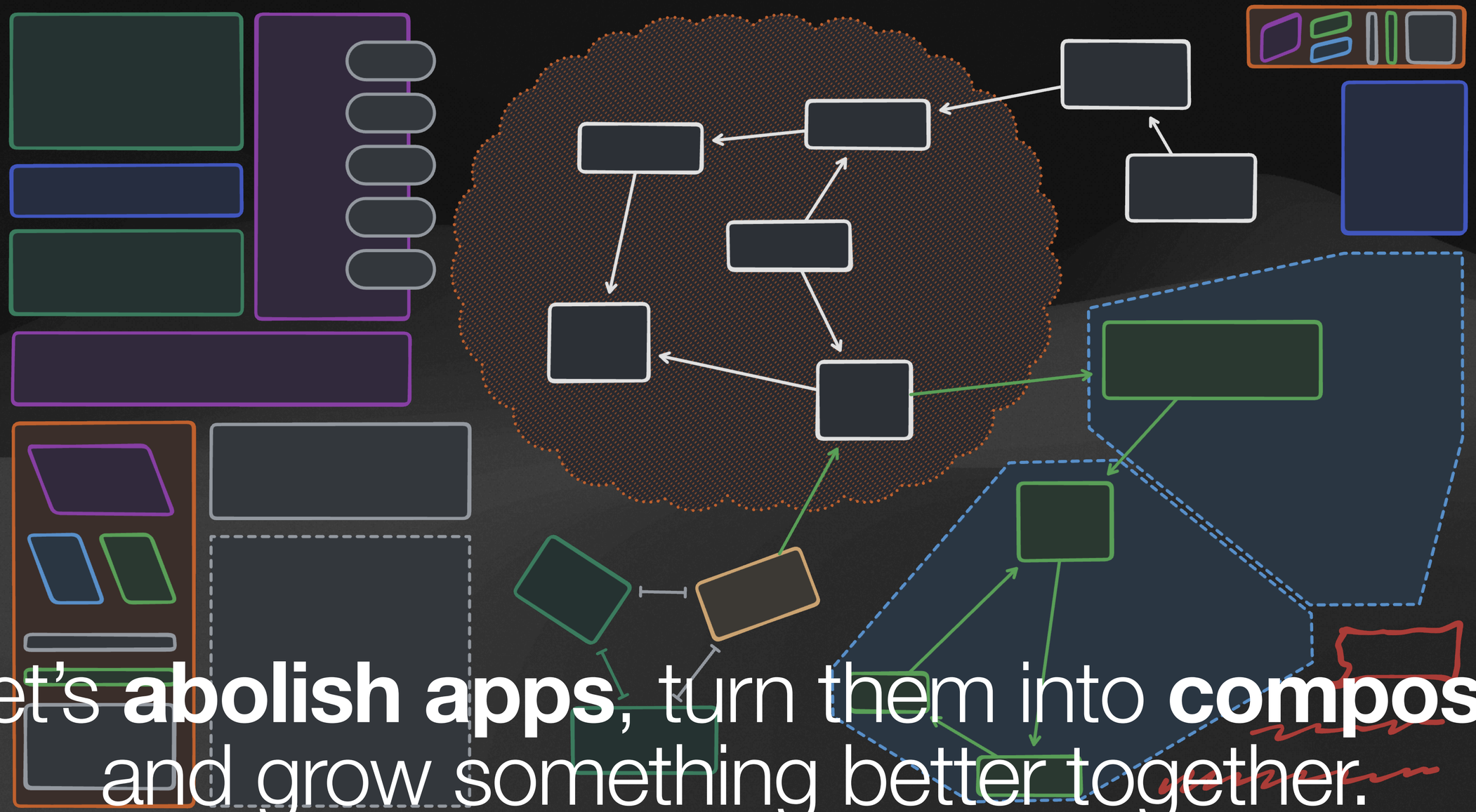

$$<\!\!- \; f \in C^0 \; \!\!->$$



# Can we create a grammar of *localised* ‘visual–semantic morphisms’?



# A new substrate for UX/UI *integration*



Let's **abolish apps**, turn them into **compost**,
and grow something better together.

# Say Hi!

**Twitter/X:** @OrionReedOne
**Mastodon:** @orion@hci.social
**Email:** me@orionreed.com
**GitHub:** github.com/orionreed