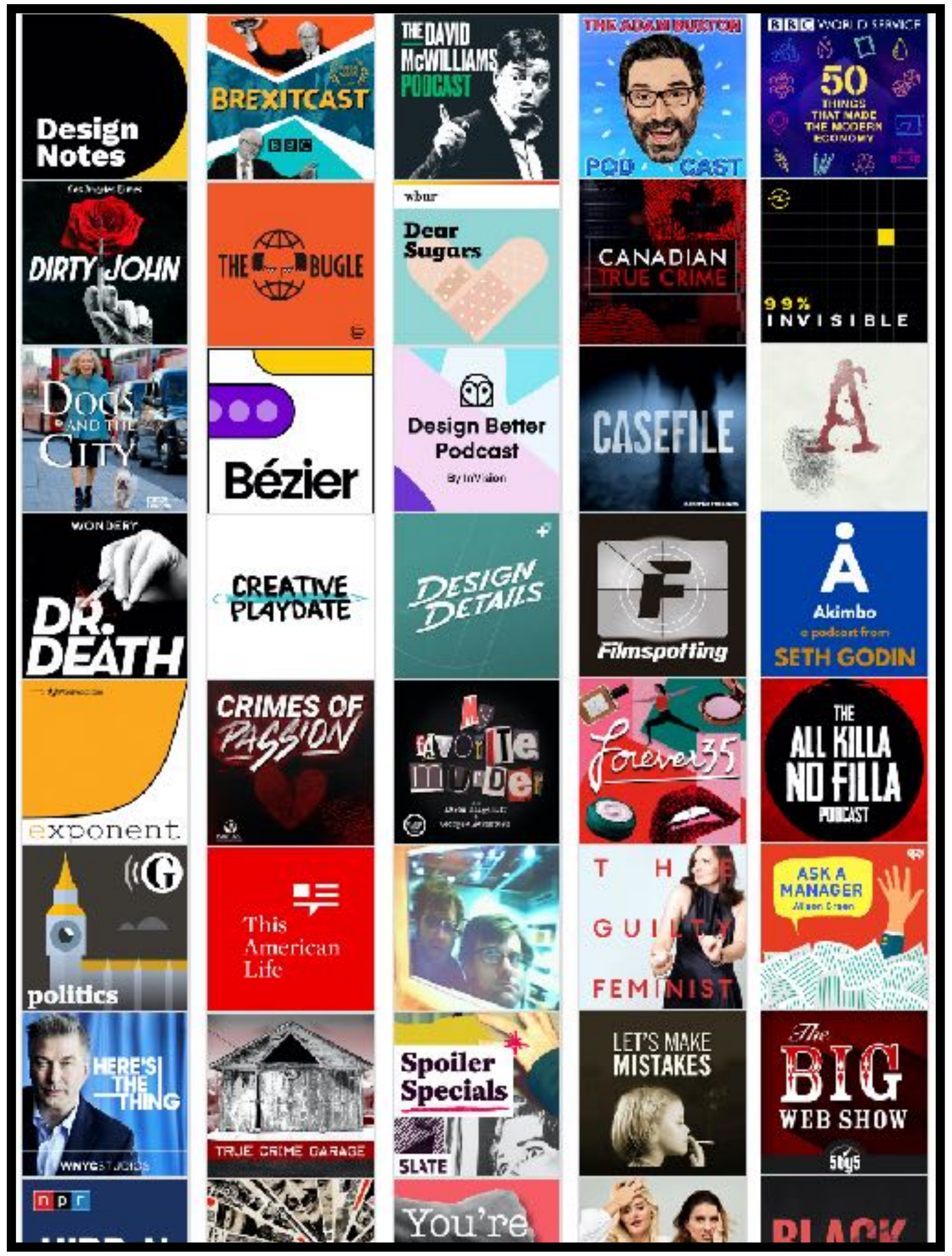
HUMANS VS DESIGN AT SCALE

@yaili
All Things Open,
October 2019

frank



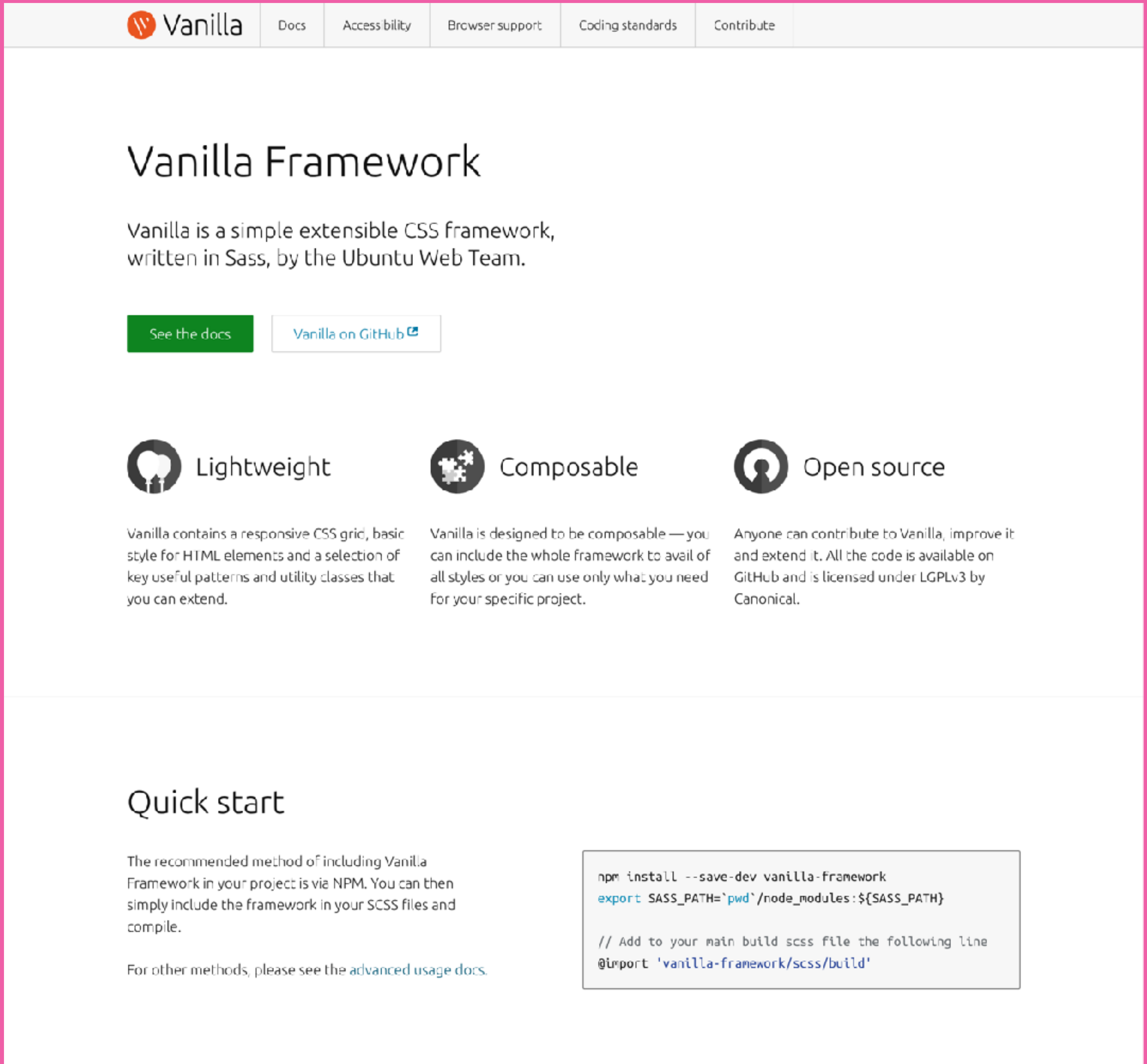
+ more

home ♡

Microsoft

BUTTONS
LINKS
• Text links
• when to use a link and a button
Do you want to save your changes?
Privacy notice
Cancel
Email me my results
'DIALOG' in IOS
CREATE PROPOSAL TO CHANGE TABLE HEADINGS COLOURS
Category
kernel app
device manager
OS app
list - tree
p-list ~~[illegible]~~ icon
50 files
Download zip
Bundle configuration
IN VANILLA
Power
Restart
Power off
Timezone
GMT - 8:00 Pacific Time (US & Canada)
kibana
NAVIGATION
INTERACTIVE TABLE
ICONS
TYPOGRAPHY
CODE BLOCK
LINKS
HTML
METADATA?
COMMENTS
LISTS
FORM
• errors
- POST CODE
• form options which open more options
Sign in

New
release!



Vanilla Framework

Vanilla is a simple extensible CSS framework, written in Sass, by the Ubuntu Web Team.

Lightweight

Vanilla contains a responsive CSS grid, basic style for HTML elements and a selection of key useful patterns and utility classes that you can extend.

Composable

Vanilla is designed to be composable — you can include the whole framework to avail of all styles or you can use only what you need for your specific project.

Open source

Anyone can contribute to Vanilla, improve it and extend it. All the code is available on GitHub and is licensed under LGPLv3 by Canonical.

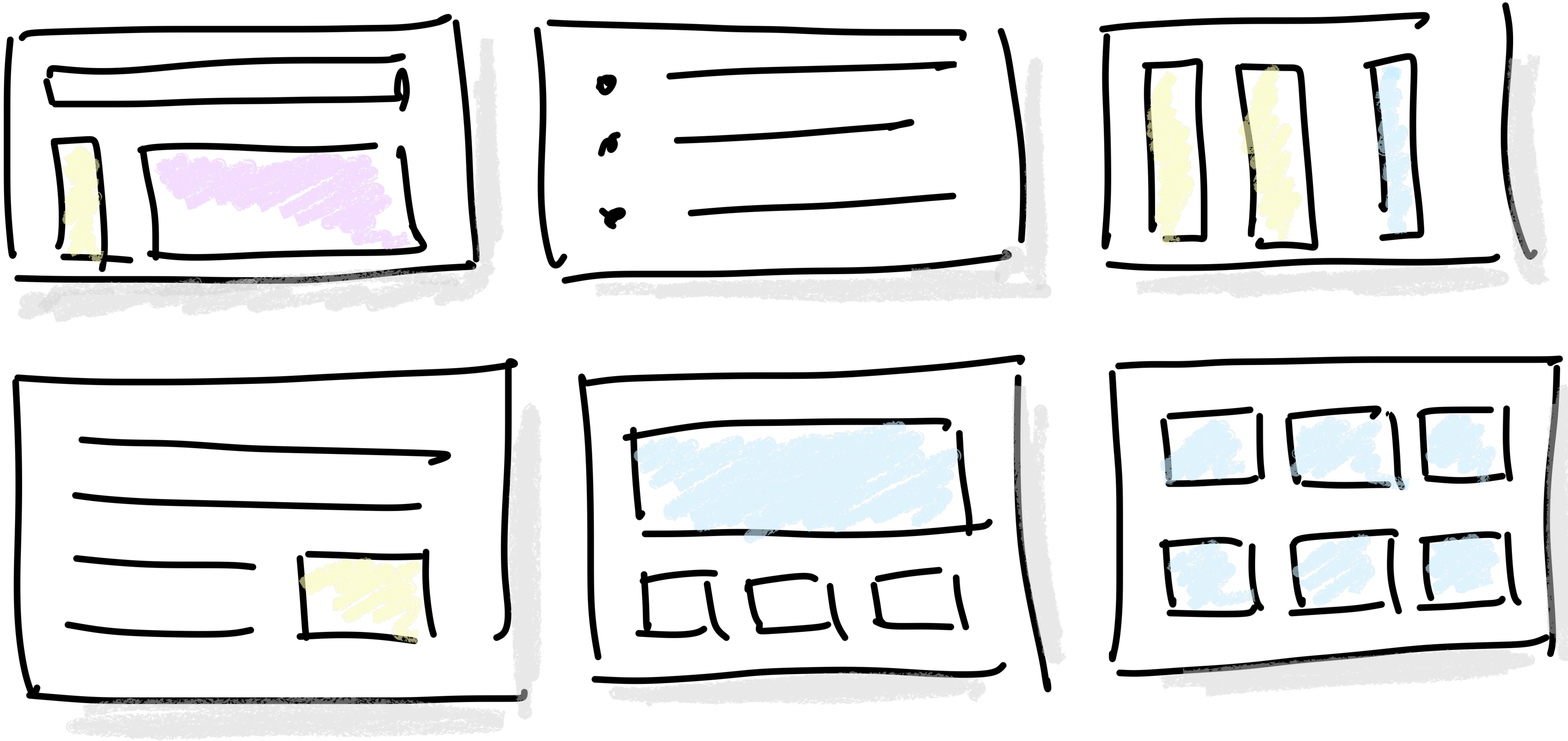
Quick start

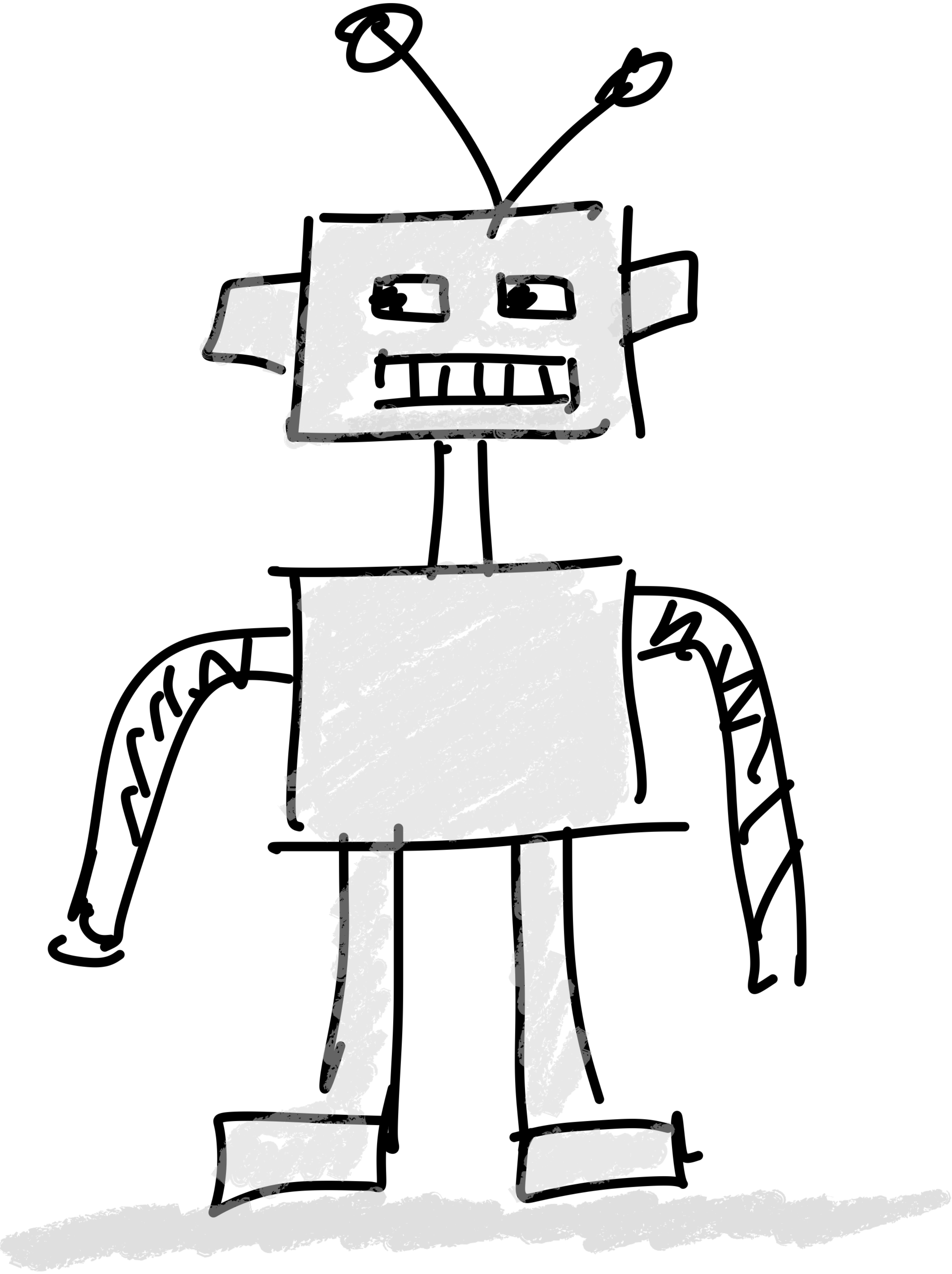
The recommended method of including Vanilla Framework in your project is via NPM. You can then simply include the framework in your SCSS files and compile.

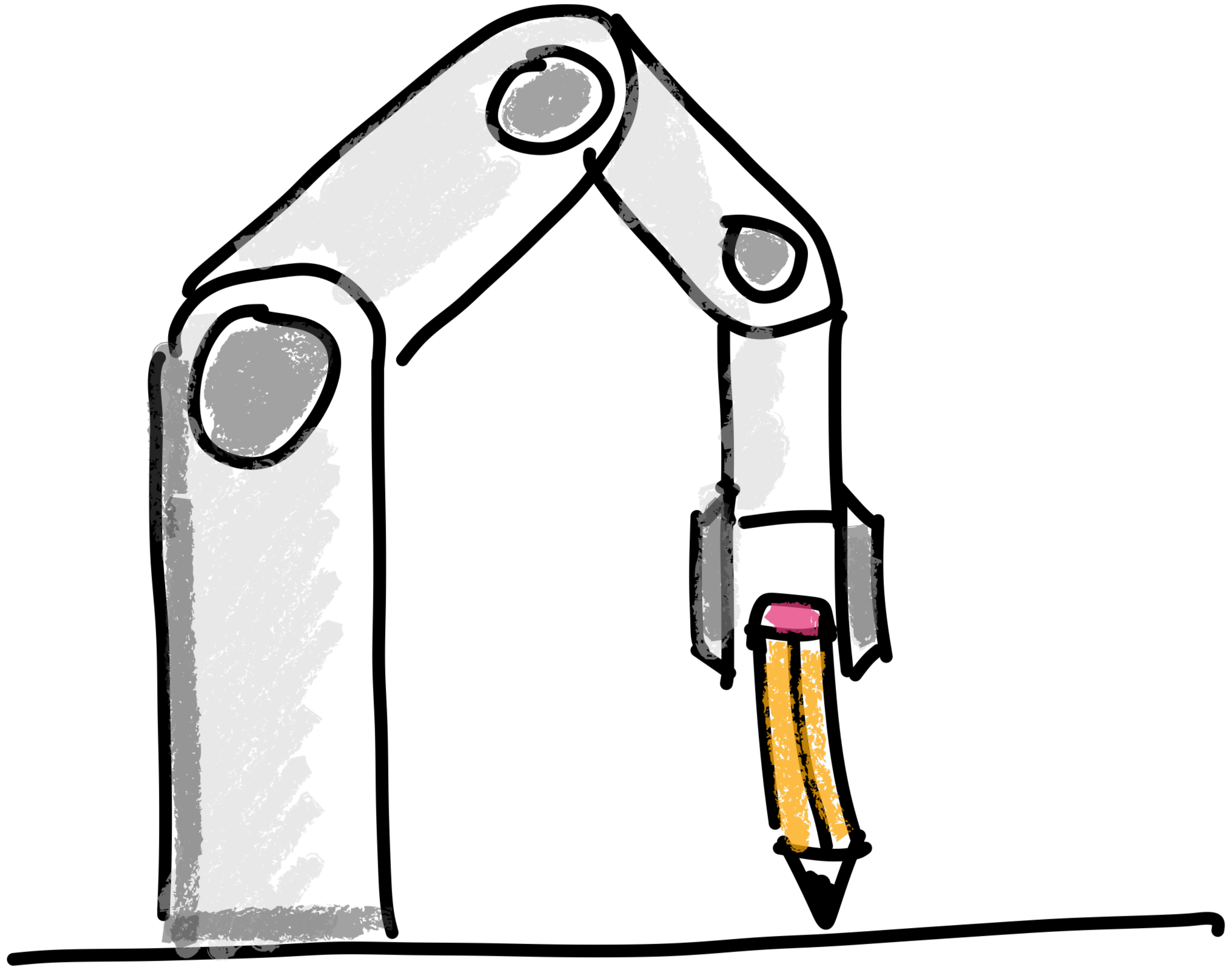
For other methods, please see the advanced usage docs.

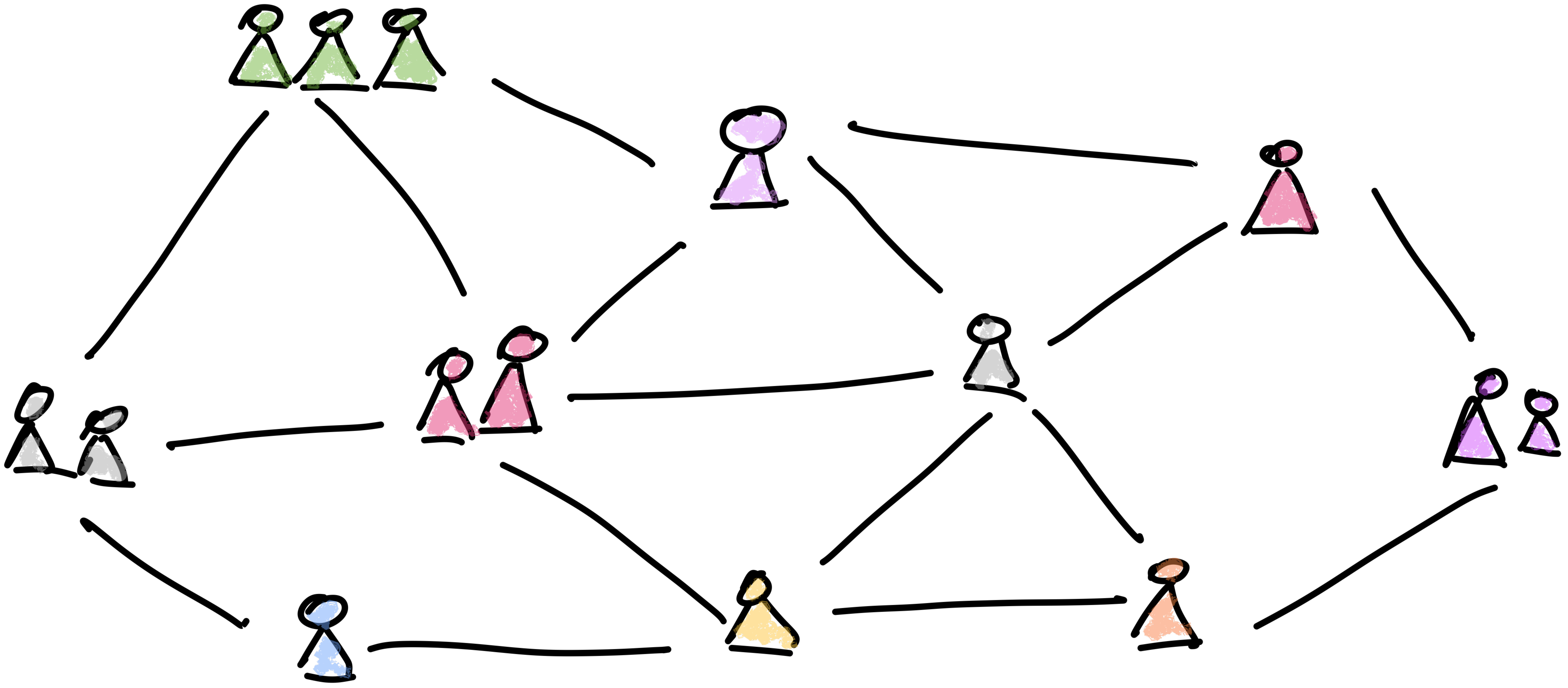
```
npm install --save-dev vanilla-framework
export SASS_PATH=`pwd`/node_modules:${SASS_PATH}

// Add to your main build scss file the following line
@import 'vanilla-framework/scss/build'
```









“human needs”

S.U.C.C.E.S.S

Subsistence-safety-security / Understanding and growth / Connection (love) and leisure / Contribution and creation / Esteem and identity / Self-governance (autonomy) and freedom / Significance and purpose

B.I.C.E.P.S. core needs

Belonging / Improvement, progress / Choice / Equality, fairness / Predictability / Significance

Maslow's hierarchy of needs

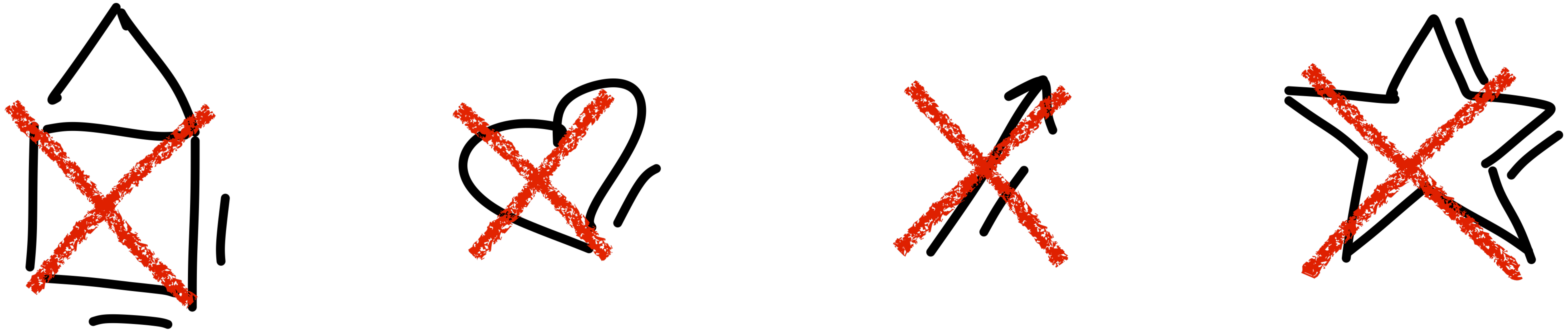
Physiological / Safety / Belongingness and love / Esteem / Self-actualisation

Certainty and predictability ⌂

Belonging ♡

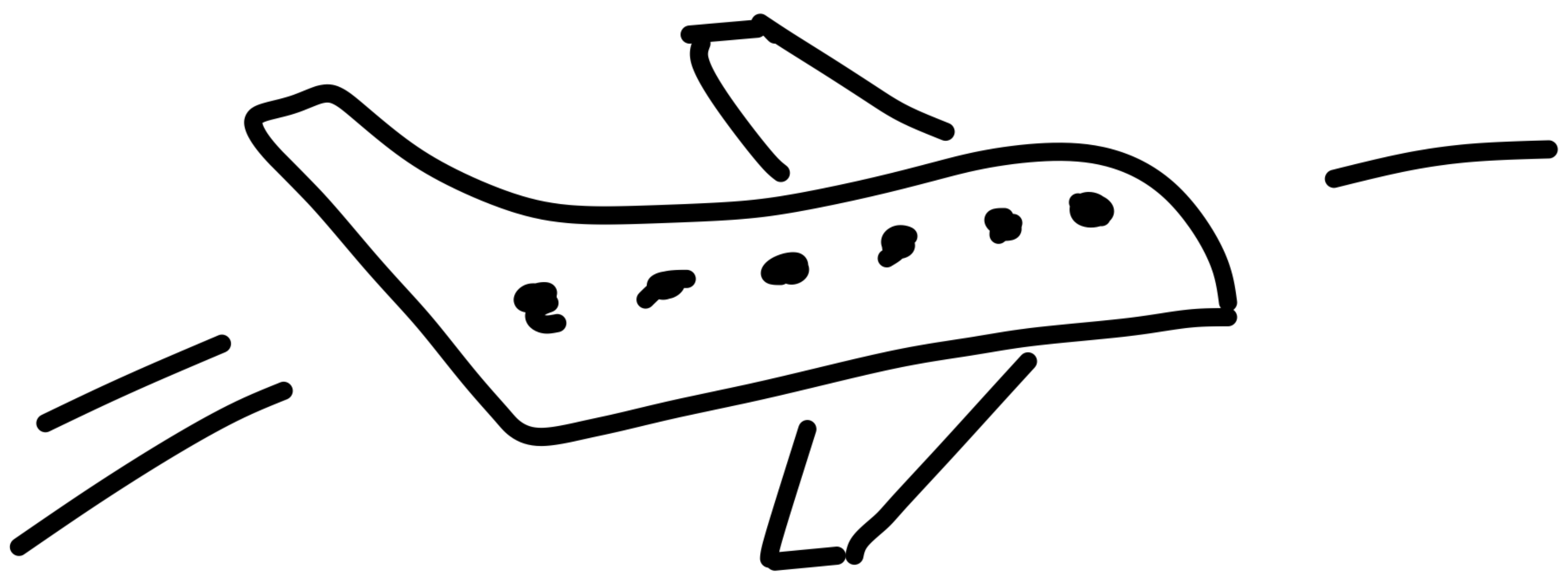
Growth ↗

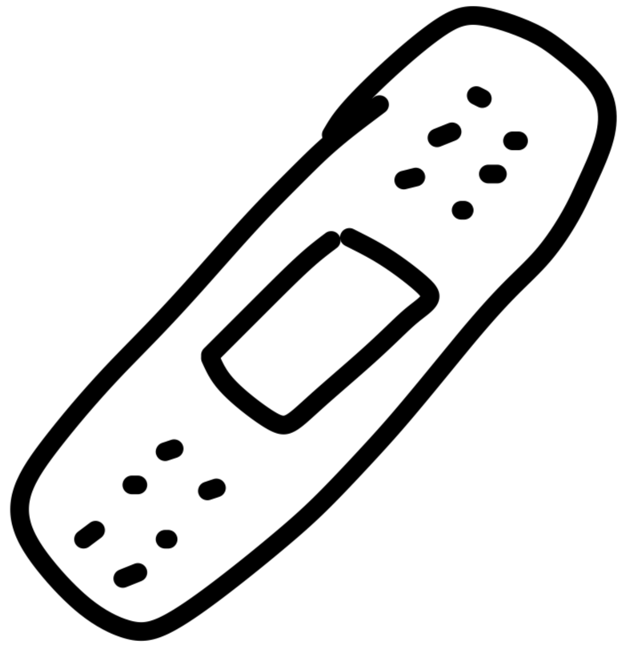
Validation ☆

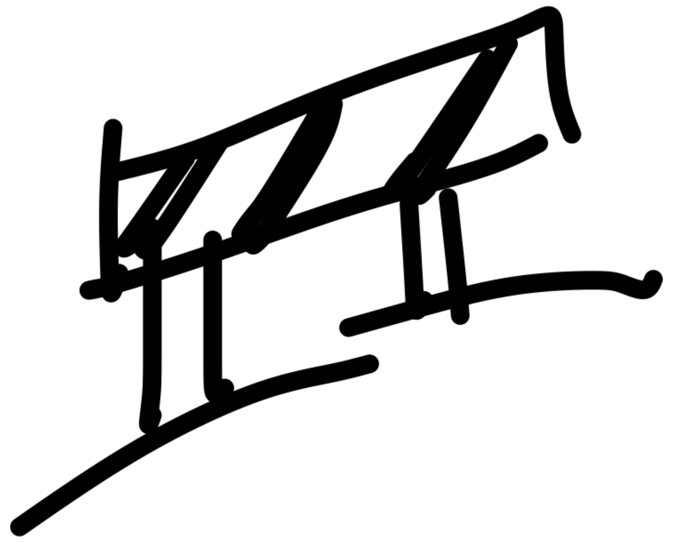


3 PRINCIPLES

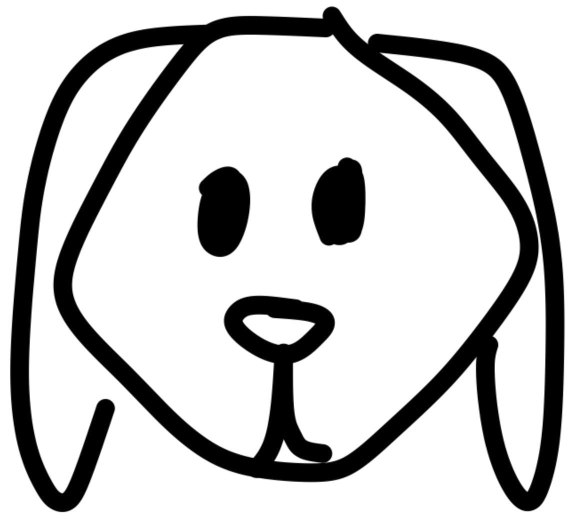
to define human friendly processes

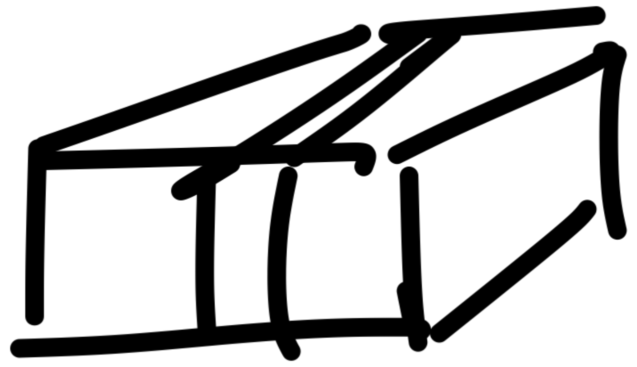




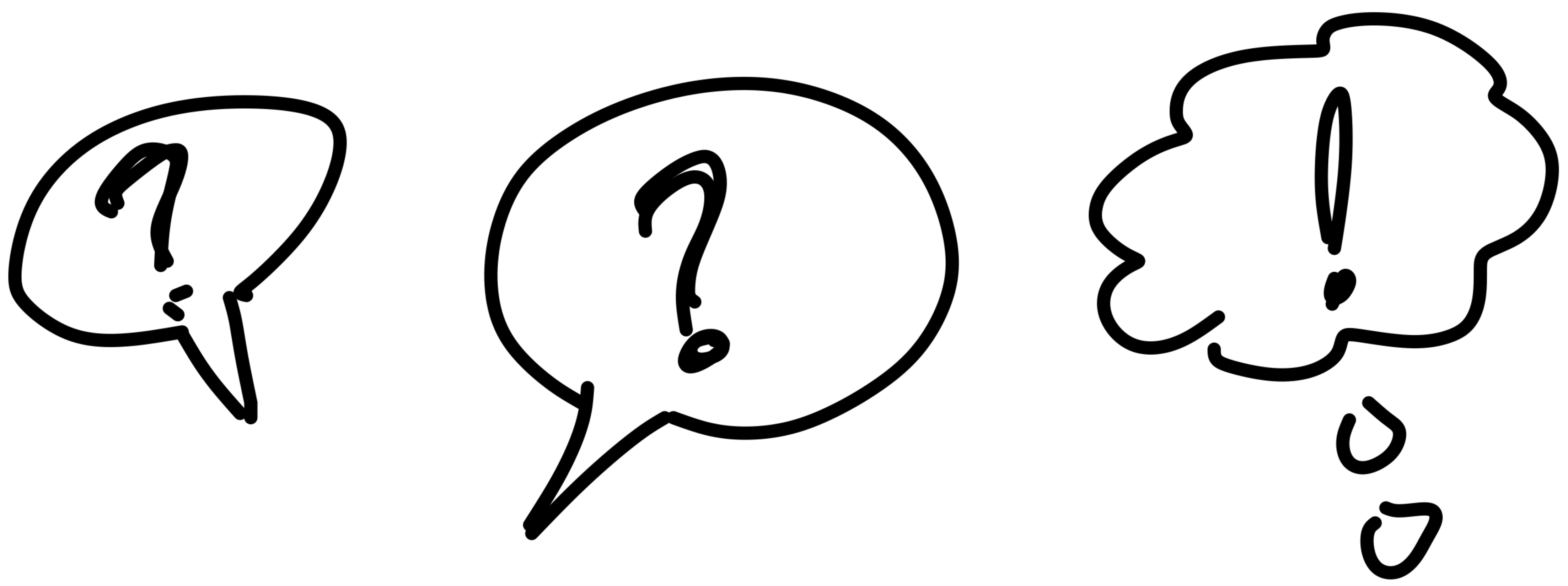


Think remote first





Have an engineering mindset

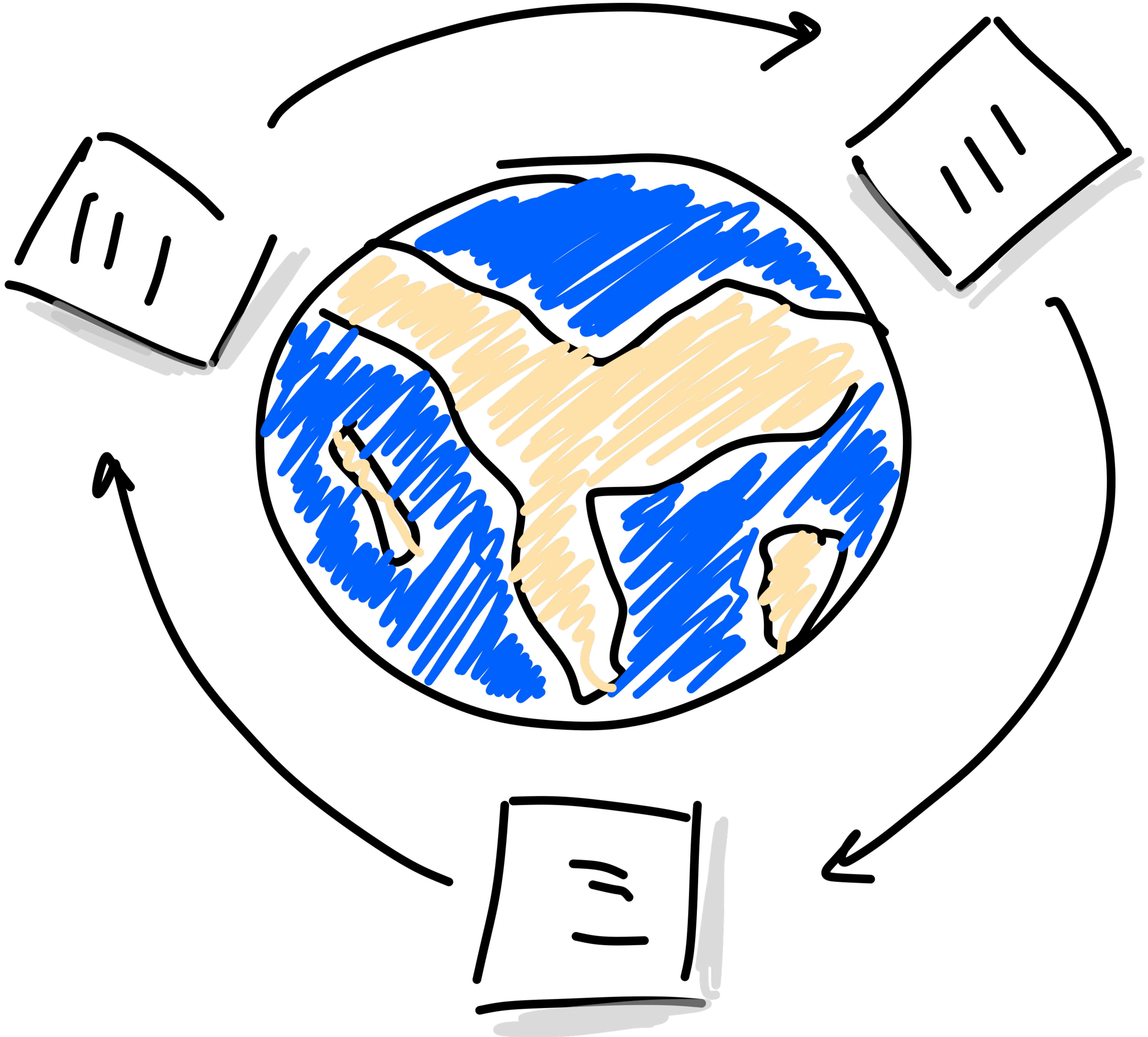


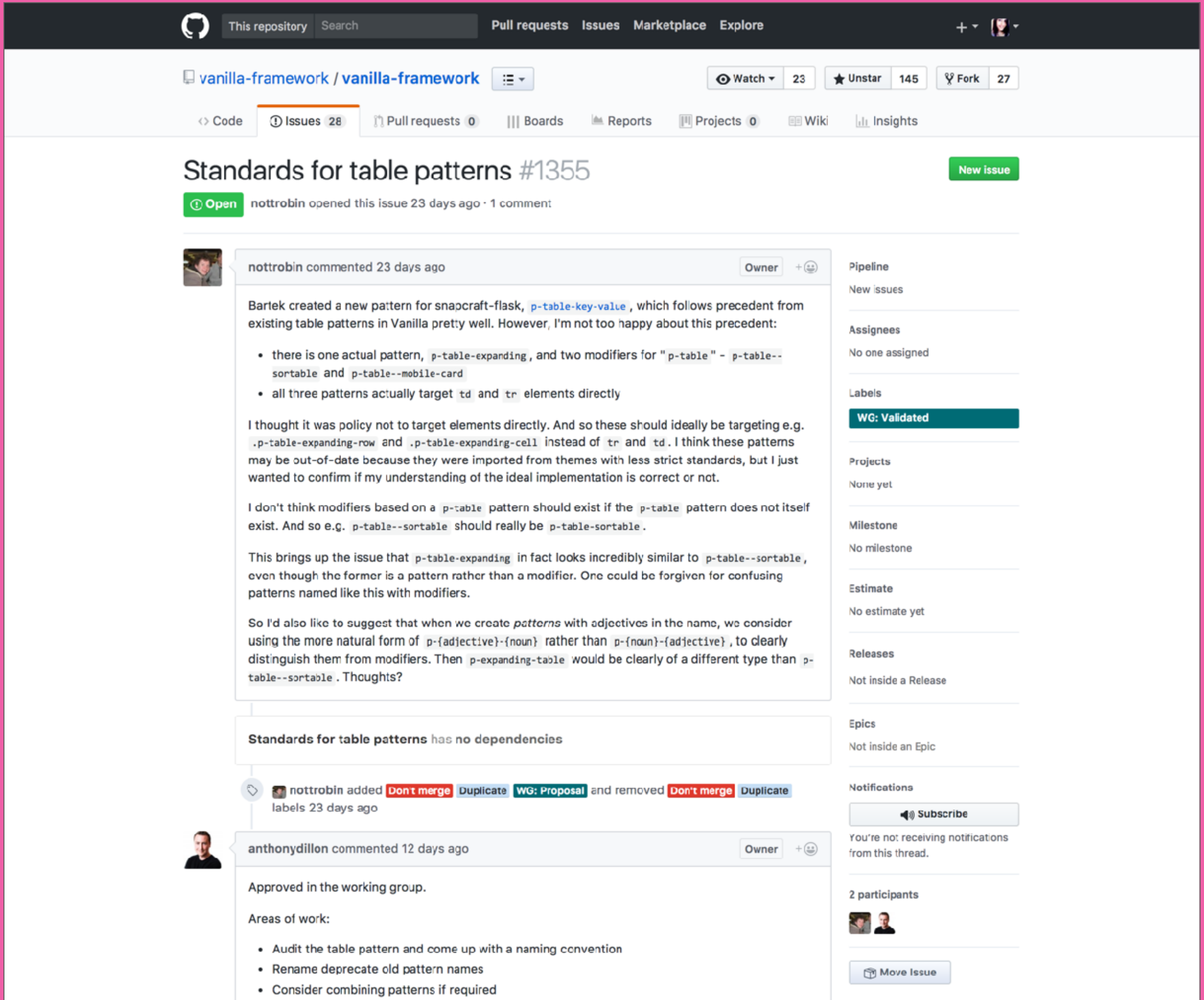
Empower people to challenge decisions

7 GUIDELINES

for human friendly design at scale

Document everything





Standards for table patterns #1355

Open nottrobin opened this issue 23 days ago · 1 comment

nottrobin commented 23 days ago

Bartek created a new pattern for snapcraft-flask, `p-table-key-value`, which follows precedent from existing table patterns in Vanilla pretty well. However, I'm not too happy about this precedent:

- there is one actual pattern, `p-table-expanding`, and two modifiers for "`p-table`" - `p-table--sortable` and `p-table--mobile-card`
- all three patterns actually target `td` and `tr` elements directly

I thought it was policy not to target elements directly. And so these should ideally be targeting e.g. `.p-table-expanding-row` and `.p-table-expanding-cell` instead of `tr` and `td`. I think these patterns may be out-of-date because they were imported from themes with less strict standards, but I just wanted to confirm if my understanding of the ideal implementation is correct or not.

I don't think modifiers based on a `p-table` pattern should exist if the `p-table` pattern does not itself exist. And so e.g. `p-table--sortable` should really be `p-table-sortable`.

This brings up the issue that `p-table-expanding` in fact looks incredibly similar to `p-table--sortable`, even though the former is a pattern rather than a modifier. One could be forgiven for confusing patterns named like this with modifiers.

So I'd also like to suggest that when we create *patterns* with adjectives in the name, we consider using the more natural form of `p-{adjective}-{noun}` rather than `p-{noun}-{adjective}`, to clearly distinguish them from modifiers. Then `p-expanding-table` would be clearly of a different type than `p-table--sortable`. Thoughts?

Standards for table patterns has no dependencies

nottrobin added Don't merge Duplicate WG: Proposal and removed Don't merge Duplicate labels 23 days ago

anthonydillon commented 12 days ago

Approved in the working group.

Areas of work:

- Audit the table pattern and come up with a naming convention
- Rename deprecate old pattern names
- Consider combining patterns if required

Pipeline
New issues

Assignees
No one assigned

Labels
WG: Validated

Projects
None yet

Milestone
No milestone

Estimate
No estimate yet

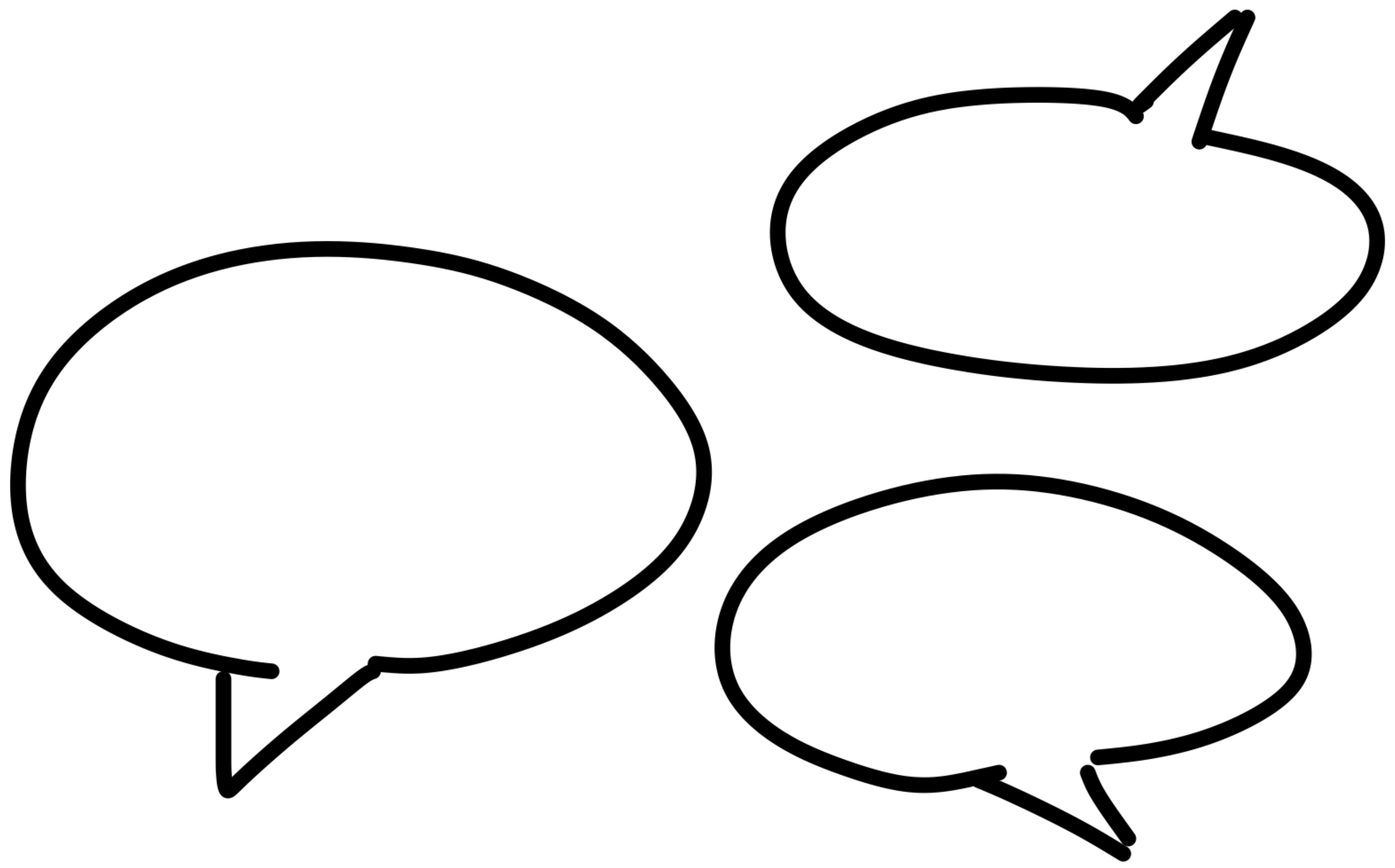
Releases
Not inside a Release

Epics
Not inside an Epic

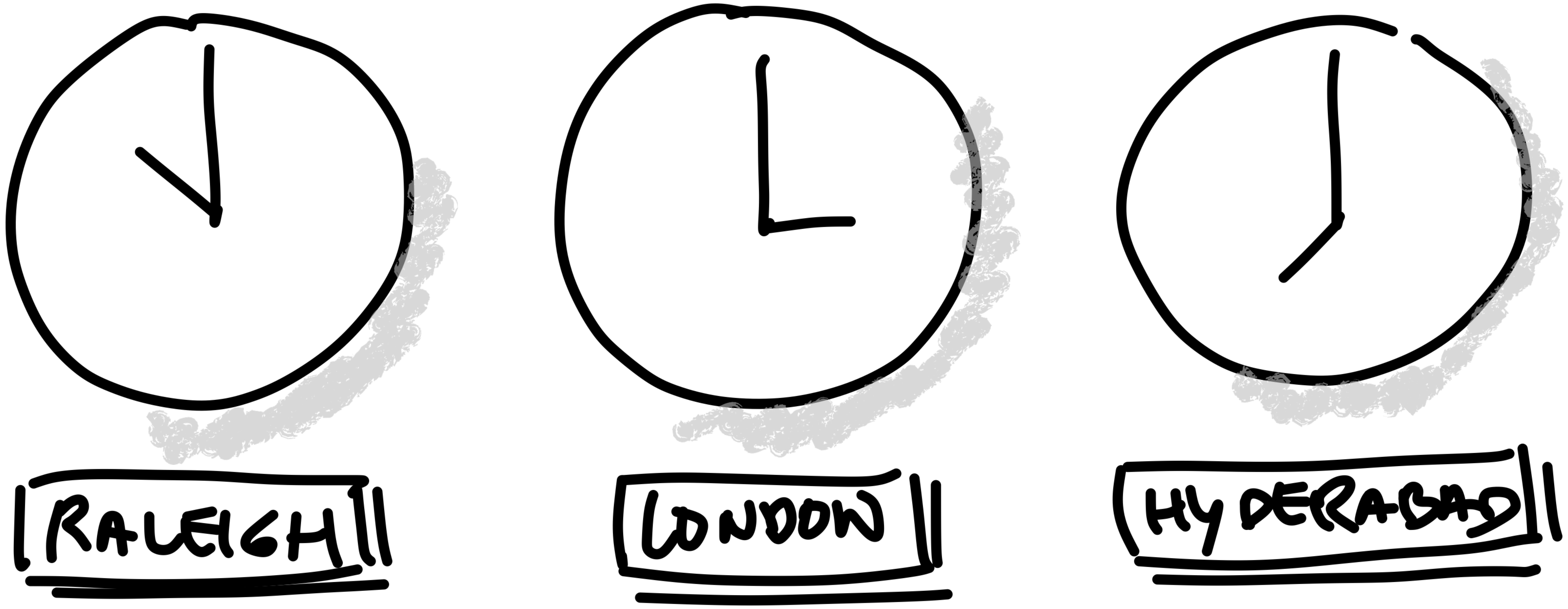
Notifications
You're not receiving notifications from this thread.

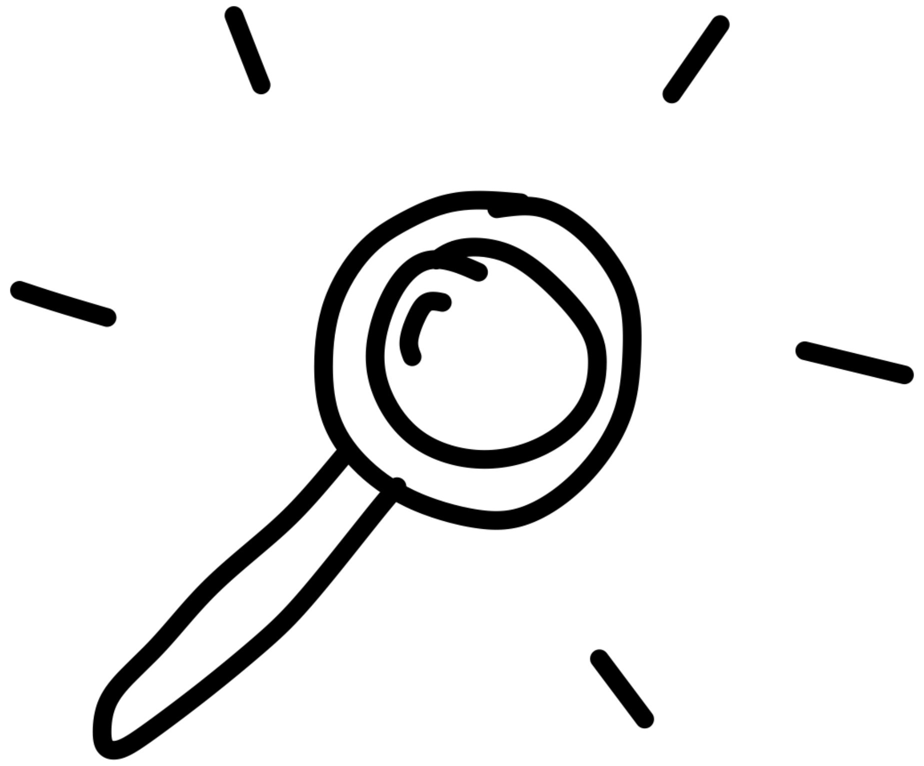
2 participants

Templates!

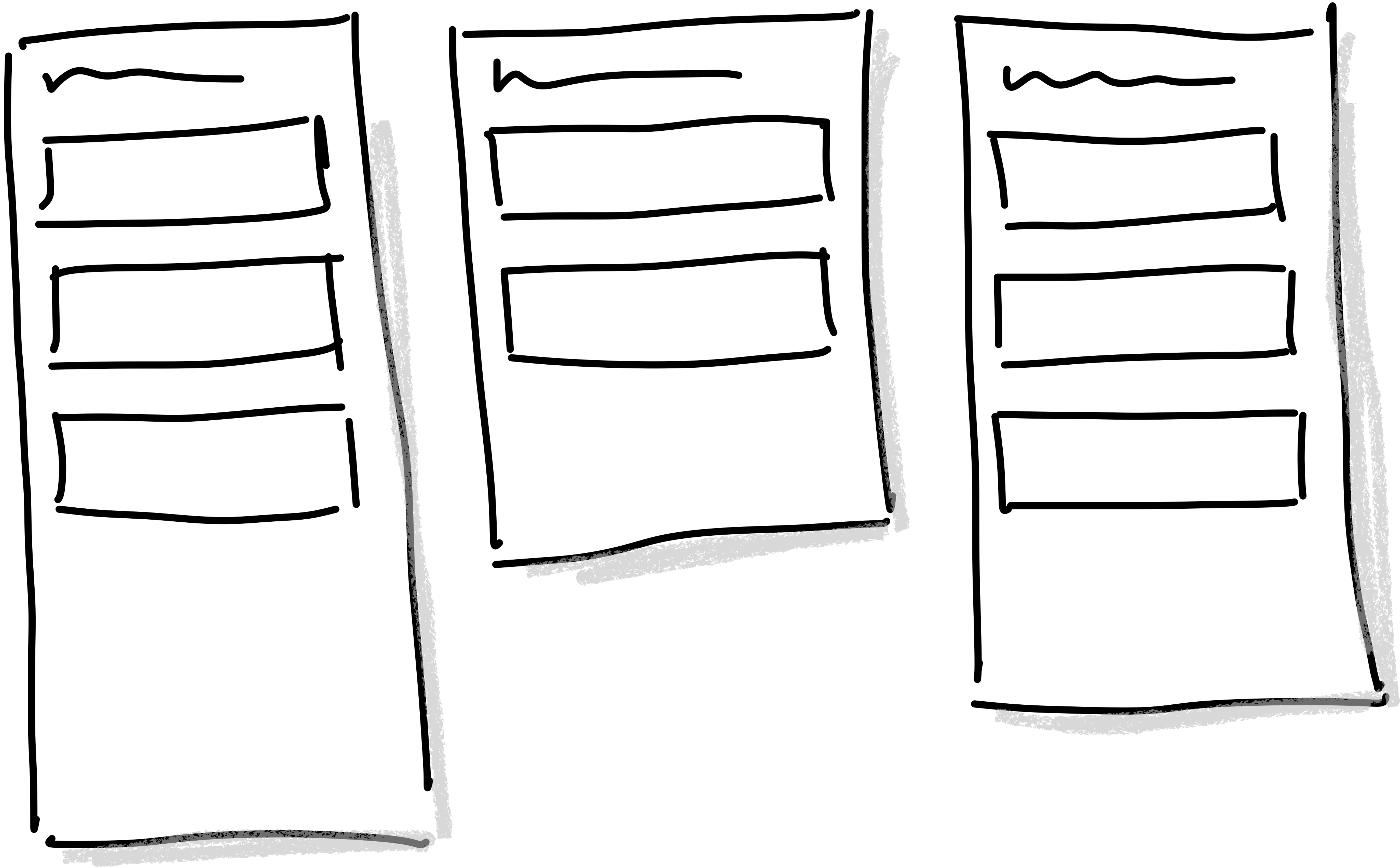


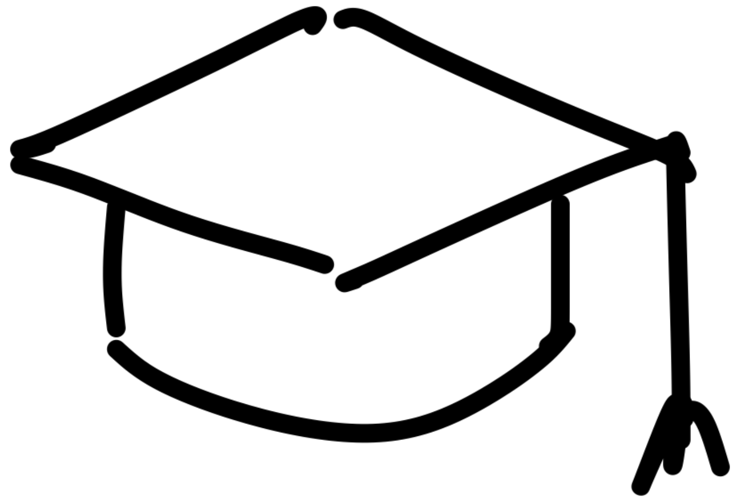
Support async participation


RALEIGH
LONDON
HYDERABAD

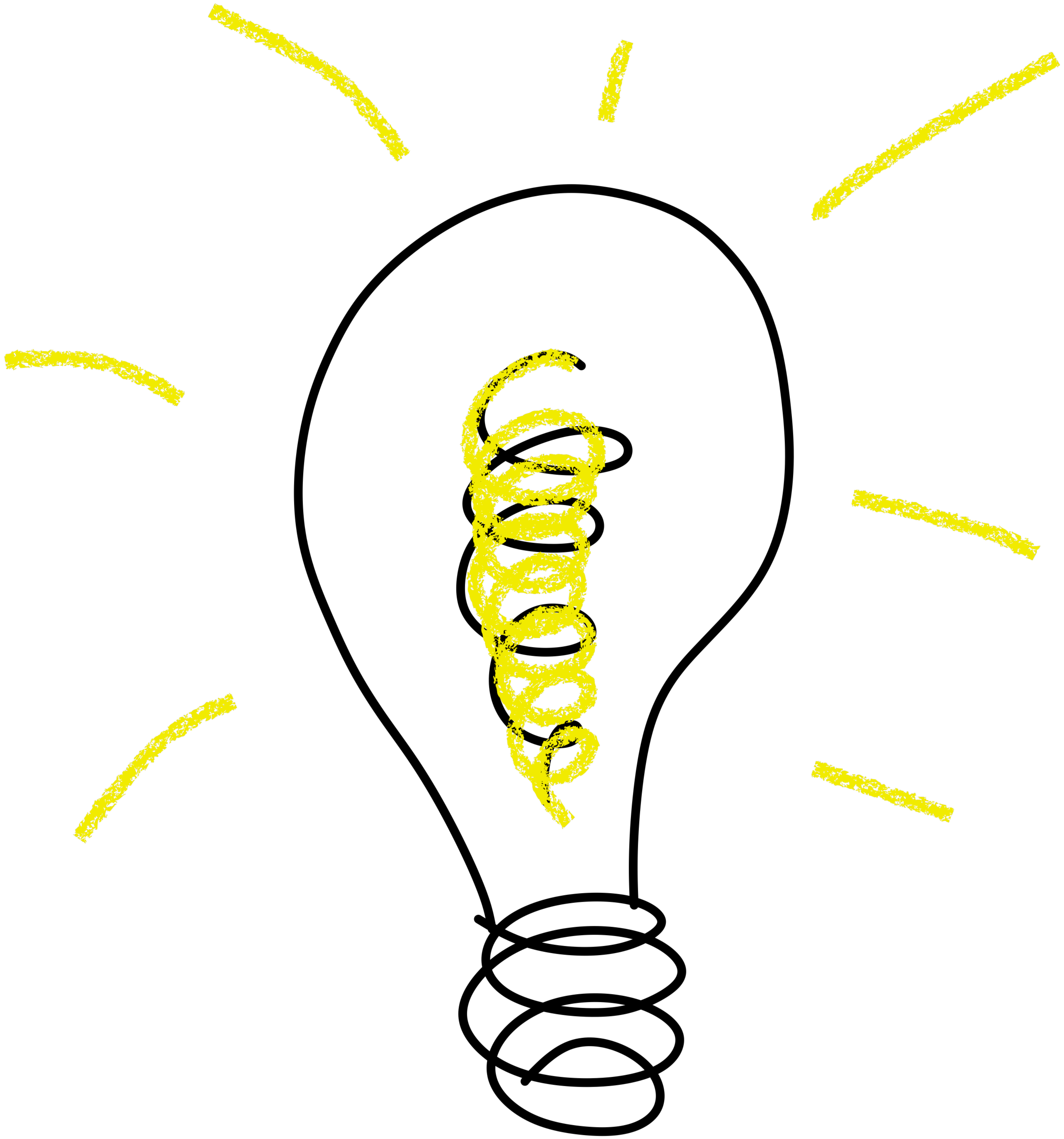


Promote transparency



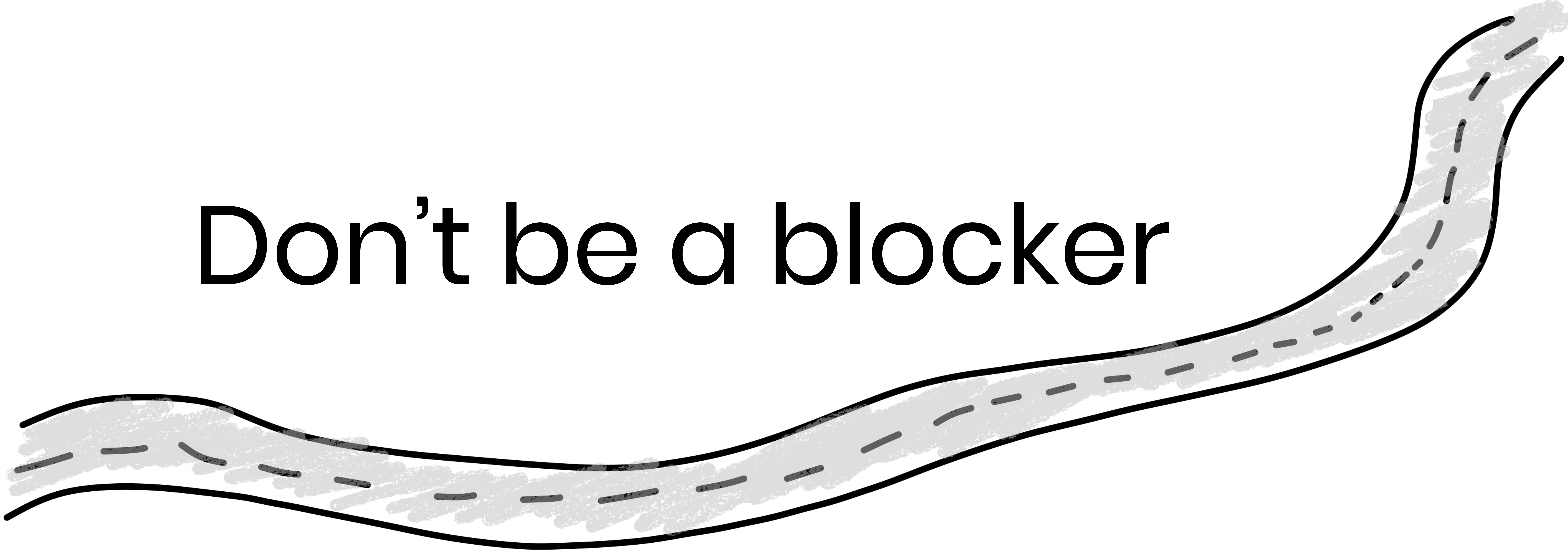


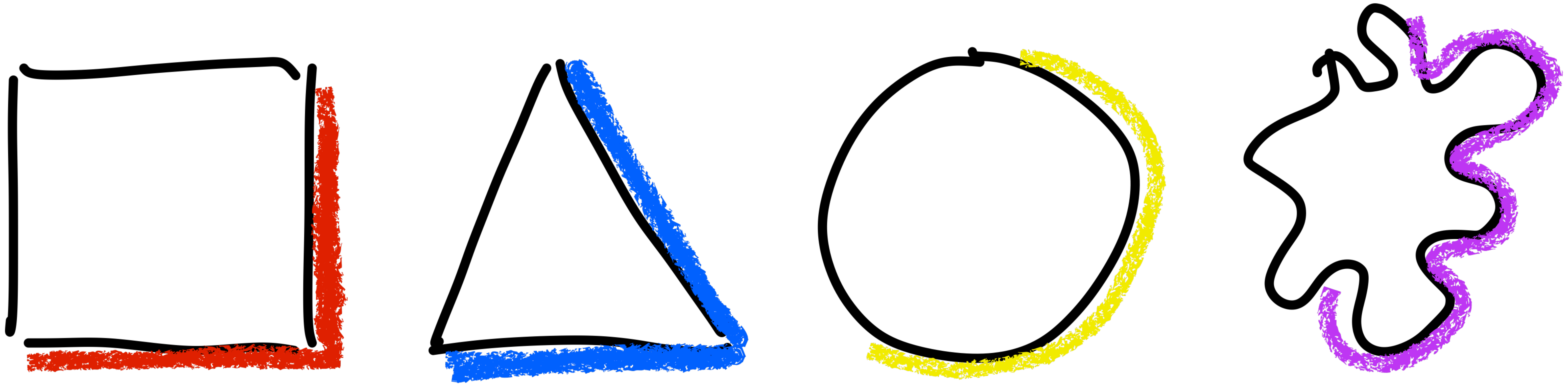
Educate, don’t dictate

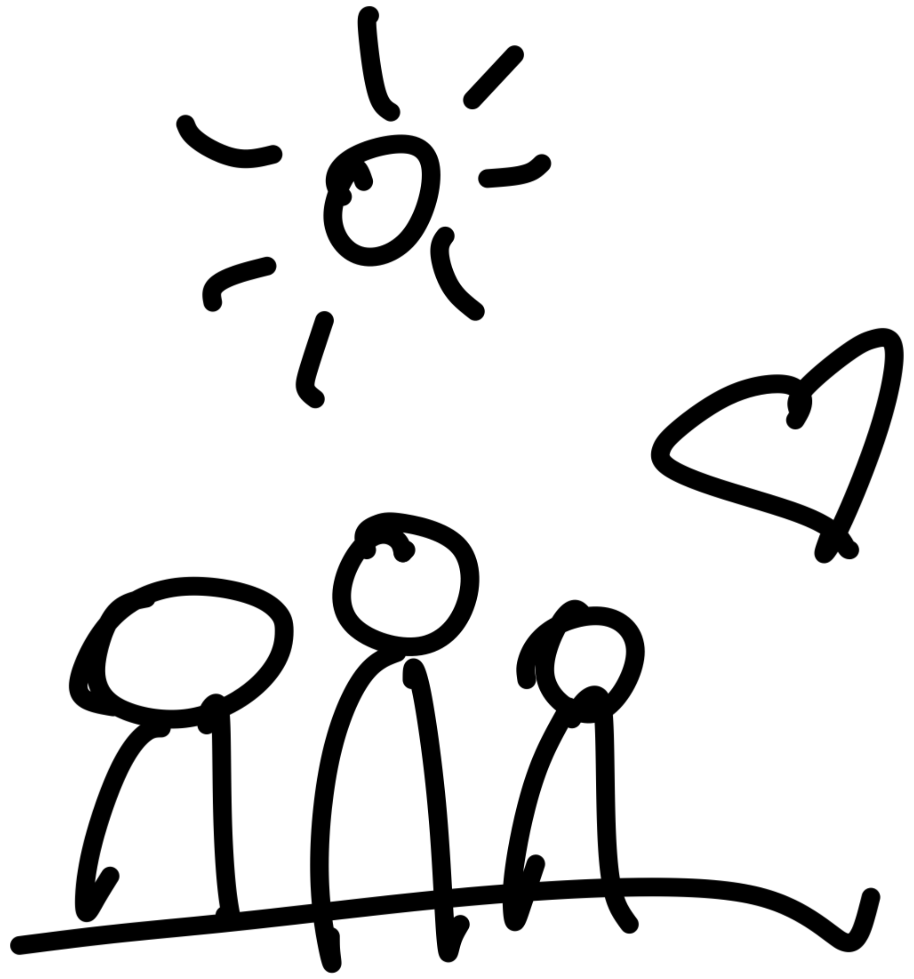


Make contributions visible

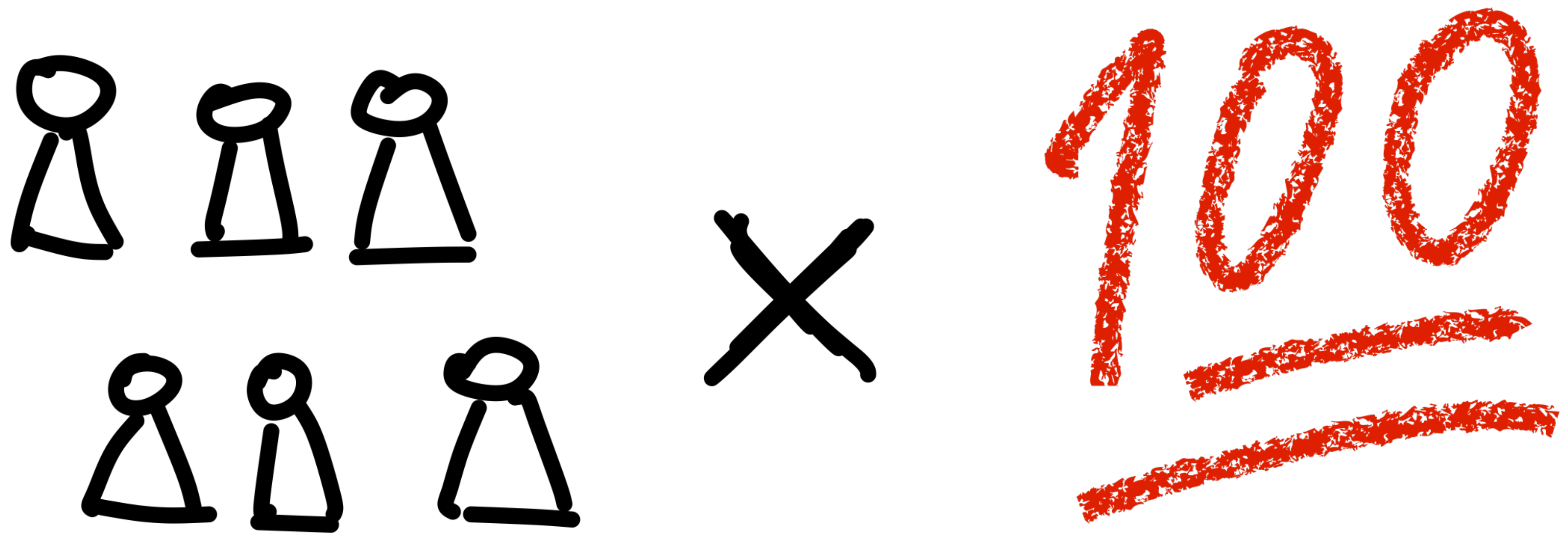
Don’t be a blocker







Hire generous people !


×
100

THANK YOU!

@yaili

References: bit.ly/yaili-ato