

in 1 year

github.com/ijlee2

in_increments_in_increments

de_the_right_code_the_right_

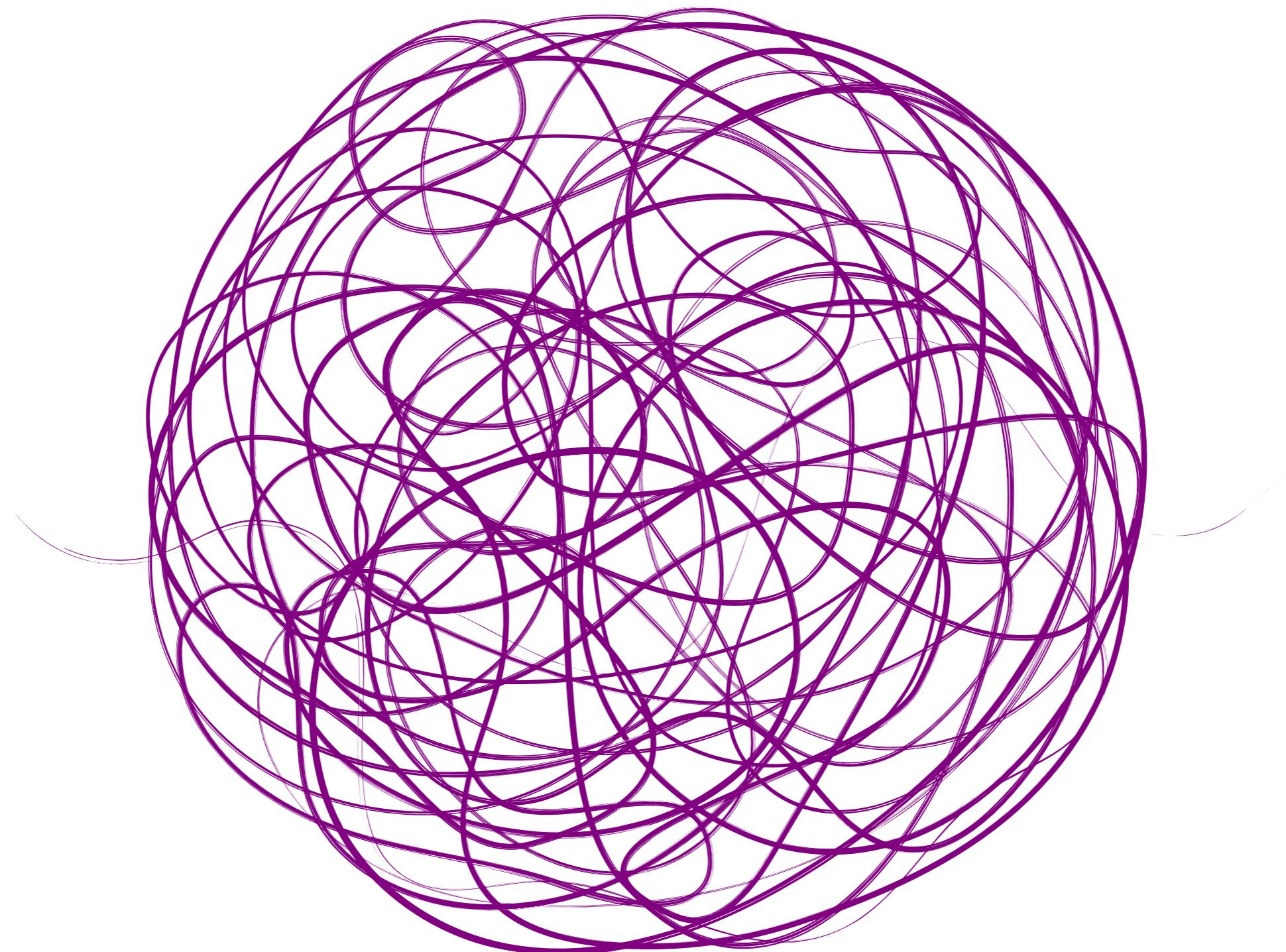
ation_solid.foundation_solid

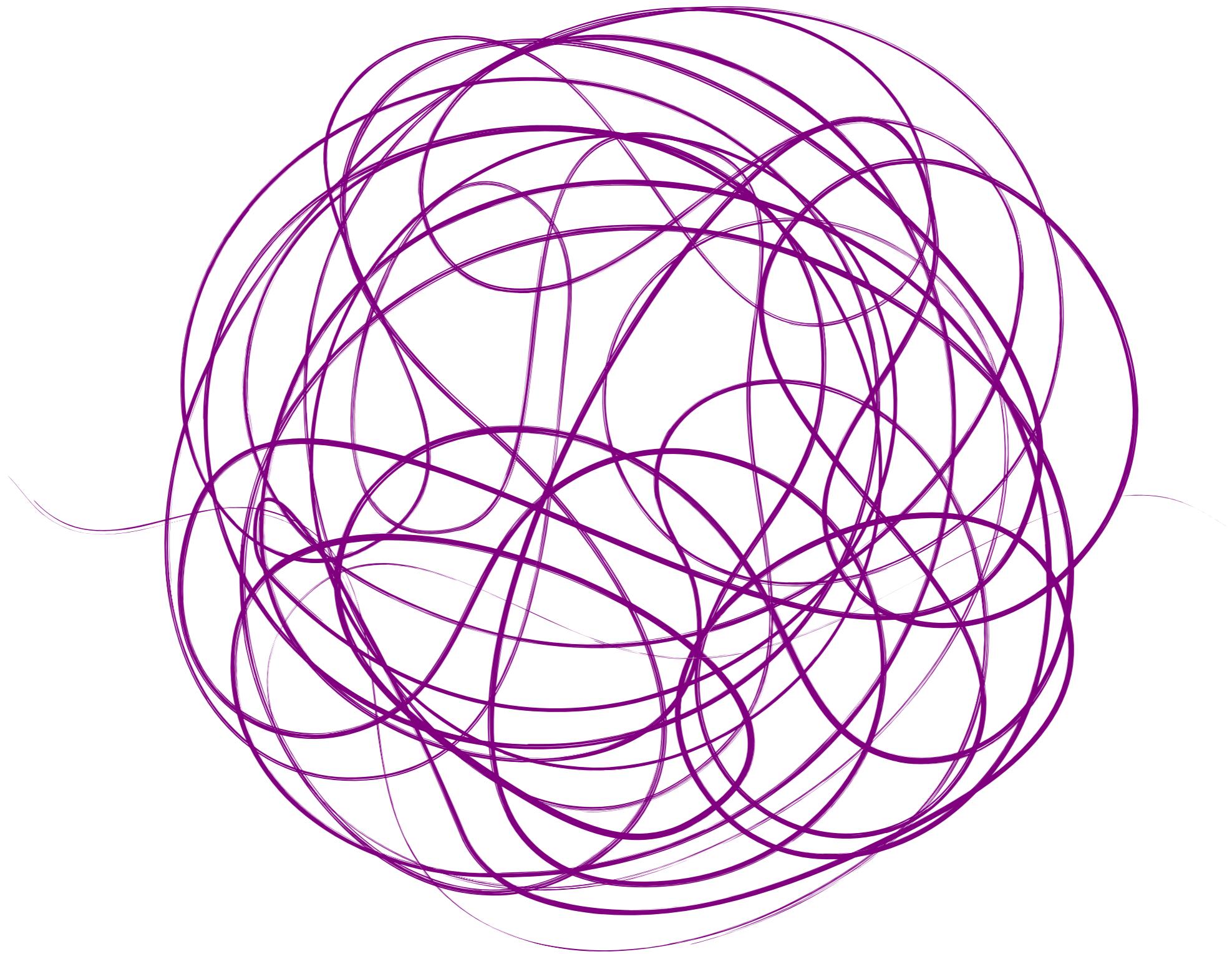
together_solving_together_so

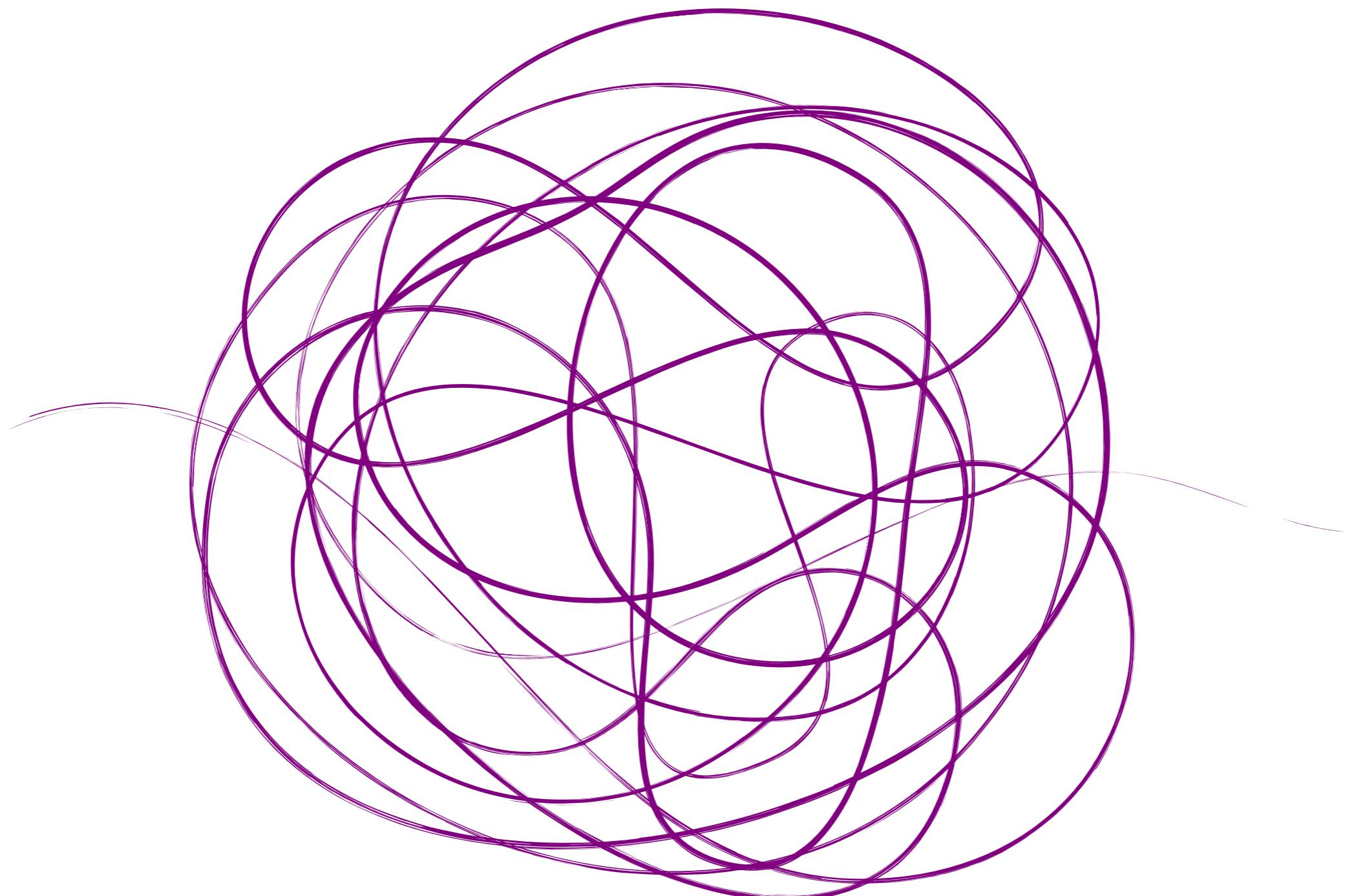
_future_is_now_future_is_now

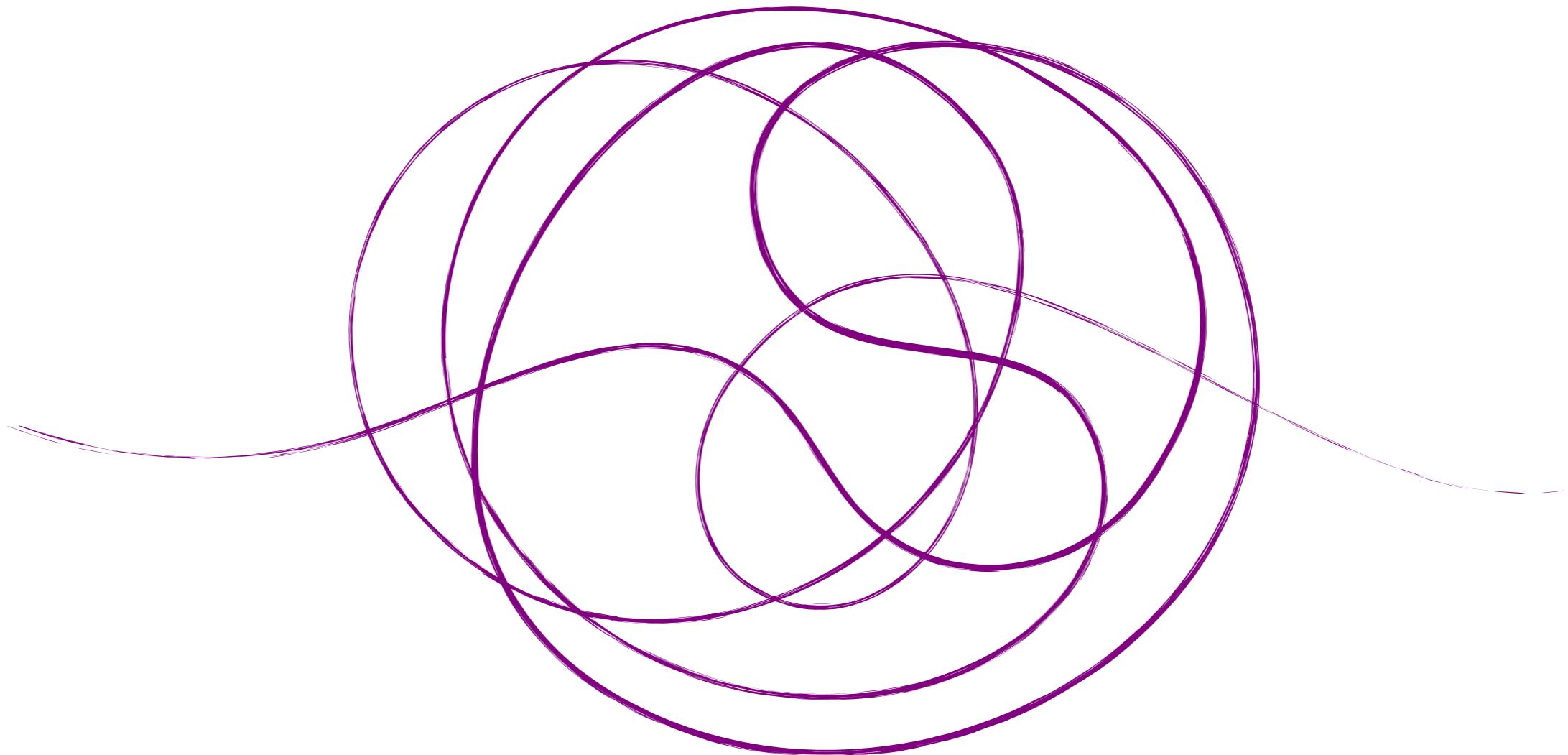
1 in increments

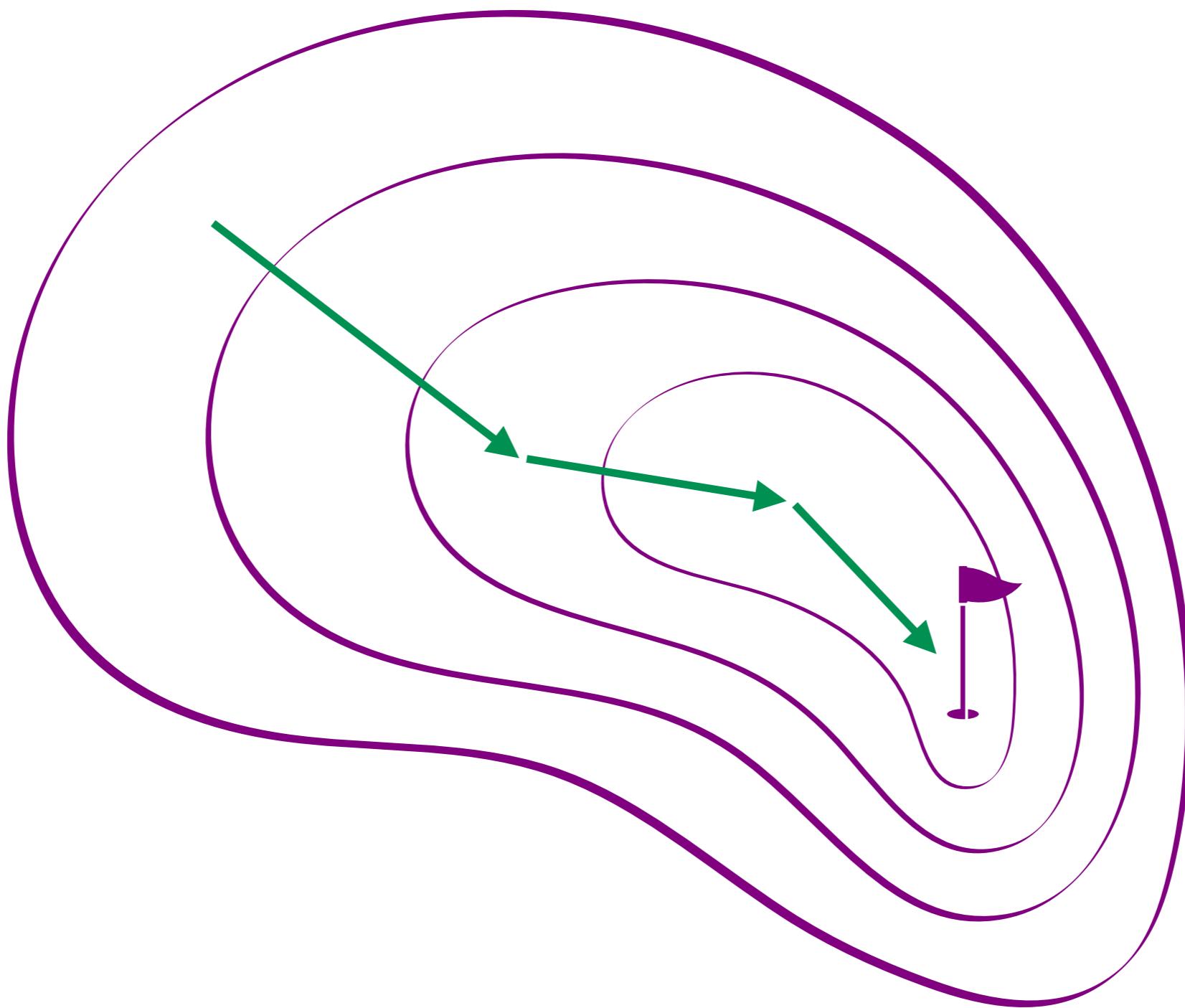
Mission impossible.











1*

code metrics

You rated me a six. I was like, "Damn."

- Janelle Monáe

2

the right code

Everybody will be dancing and be doing it right.

- Daft Punk

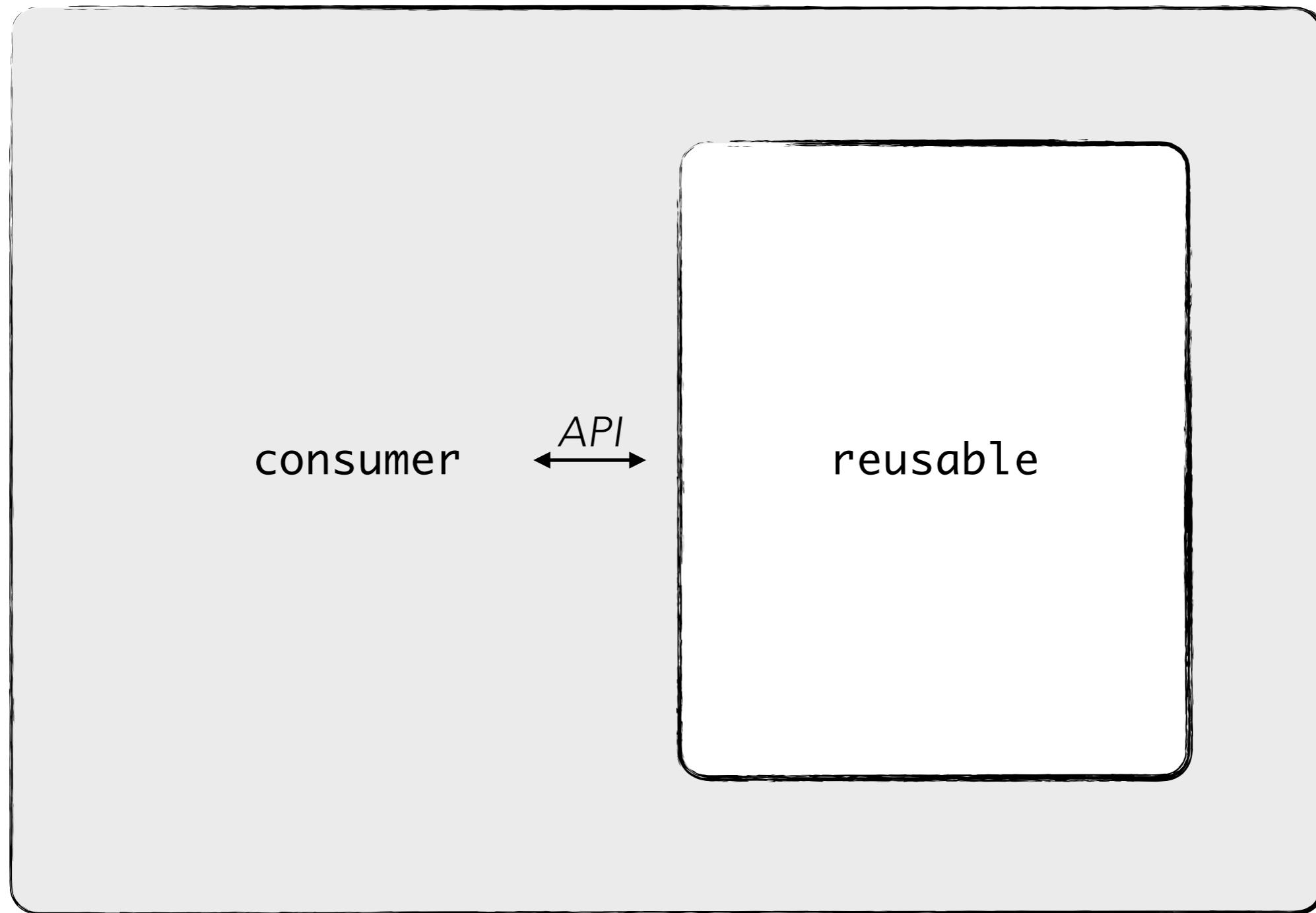
the right code

- *Has a minimum API*
- *Separates concerns*
- *Depends on few*

2a

**... has
minimum API**

The best code is no code.



github.com/ijlee2/ember-workshop



de.wikipedia.org - eierlegende Wollmilchsau

minimization

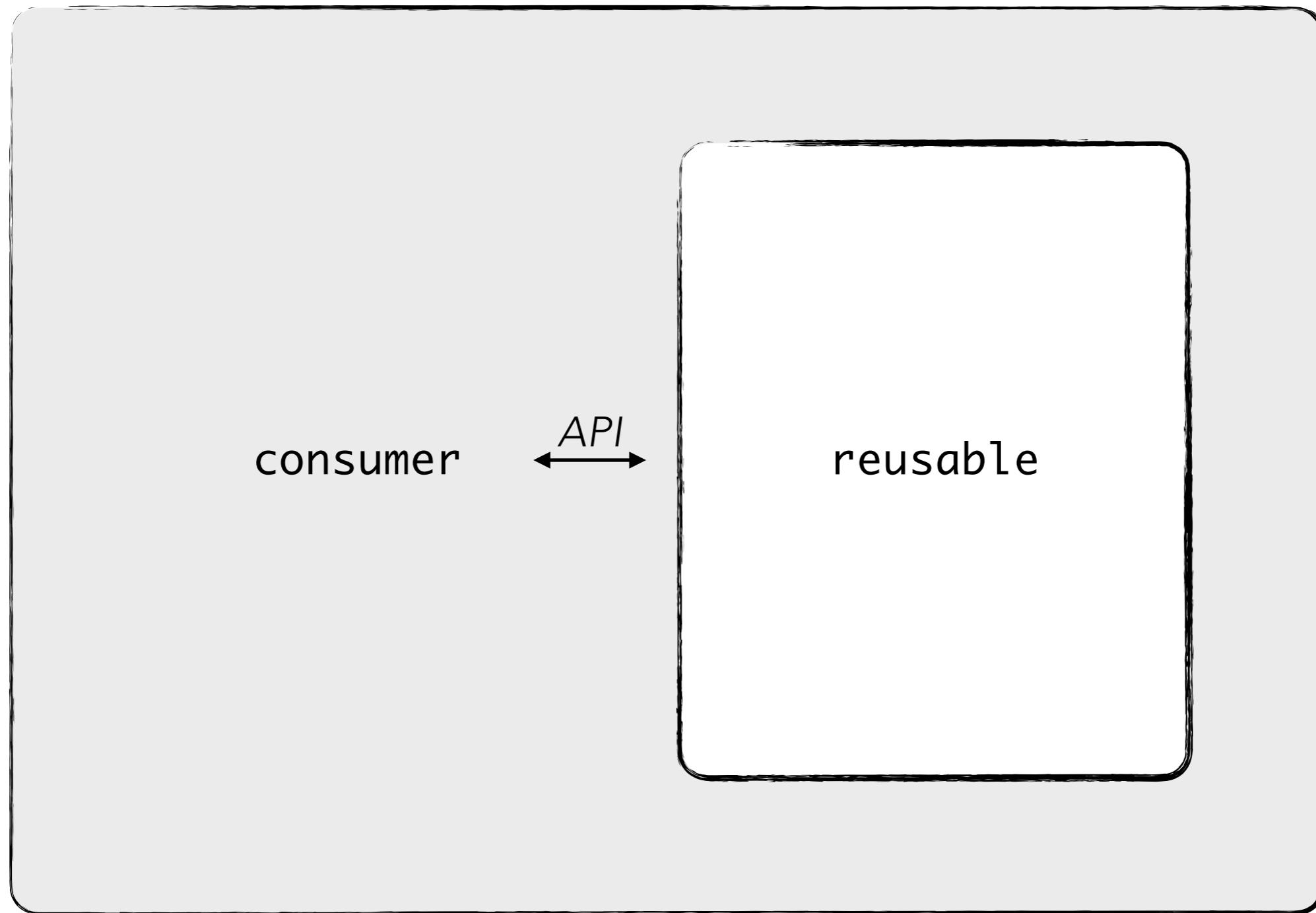
- *Research current use cases*
- *Remove dependencies among arguments (minimize how many tests need to be written)*
- *Remove rarely used arguments (cover the 80% case)*

2b

**... separates
concerns**

You gotta keep 'em separated.

- The Offspring

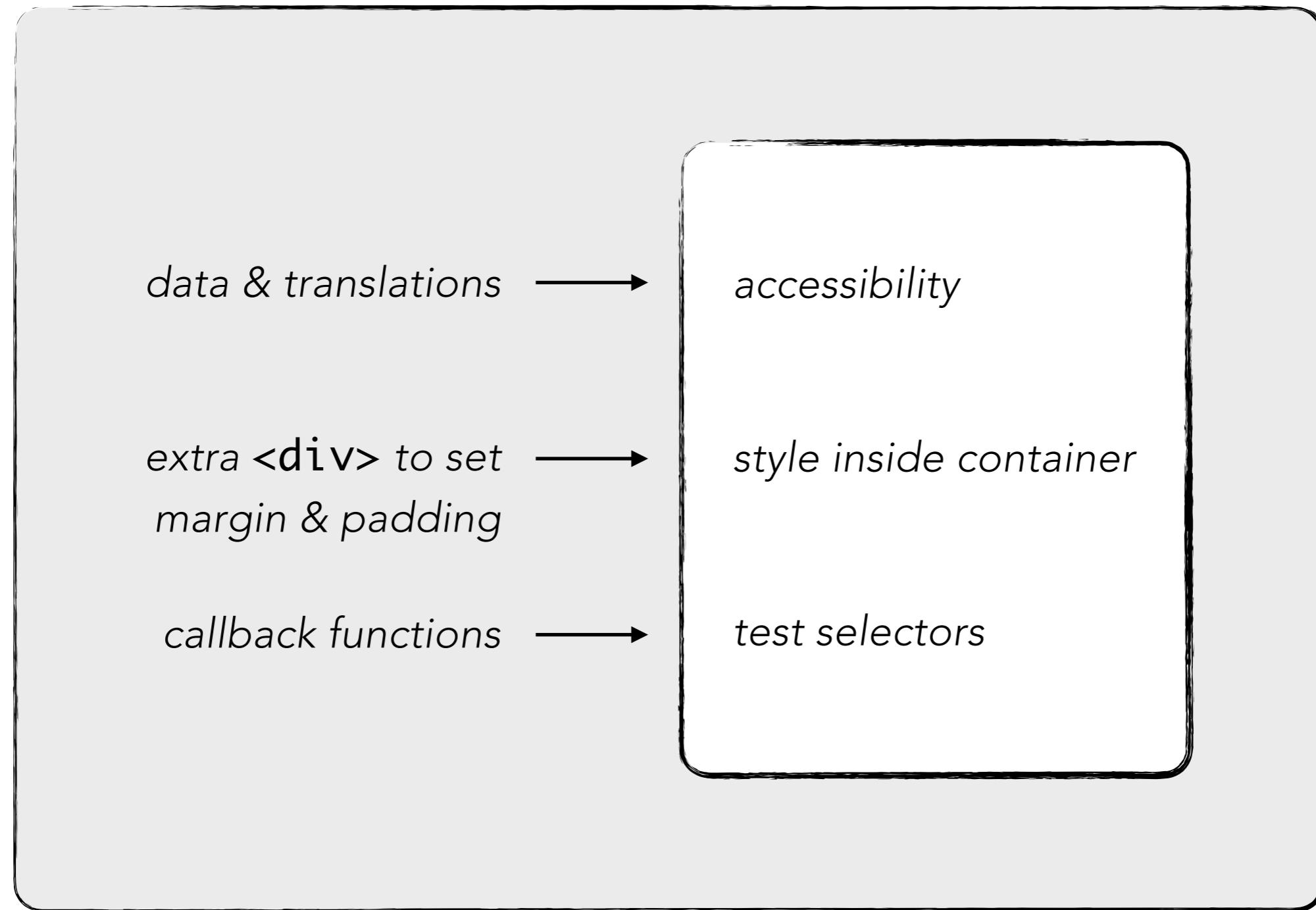


github.com/ijlee2/ember-workshop

accessibility

style inside container

test selectors



named blocks

- ● ● File: A reusable component that consumes `<Ui::Form::Field>`

```
1  <Ui::Form::Field>
2    <:label as |inputId|>
3      <label for={{inputId}}>
4        {{@label}}
5        </label>
6      </:label>
7
8    <:field as |inputId|>
9      <input
10        class={{this.styles.input}}
11        id={{inputId}}
12        type={{@type}}
13        value={{this.value}}
14        {{on "input" this.updateValue}}
15      />
16    </:field>
17  </Ui::Form::Field>
```

...attributes

● ● ● File: A consumer of <Ui::Action>

```
1  <Ui::Action
2  |  class={{this.styles.container}}
3  |  data-test-user-actions
4  >
5  |  <:primary as |p|>
6  |  |  <p.Button @label="Confirm" />
7  |  </:primary>
8
9  |  <:secondary as |s|>
10 |  |  <s.Button @label="Cancel" />
11 |  </:secondary>
12 </Ui::Action>
```

2c

... depends
on few

You've got the remedy and I need it.

- French 79

stable

- *Easy-to-follow code*
- *Documented (README, CONTRIBUTING, release notes)*
- *Tested*
- *Supported (releases are recent)*

3 solid foundation

Please take me down to steady ground.

- Julie Crochetière

3a

overhauling lint

I promise to be different! I promise to be unique!

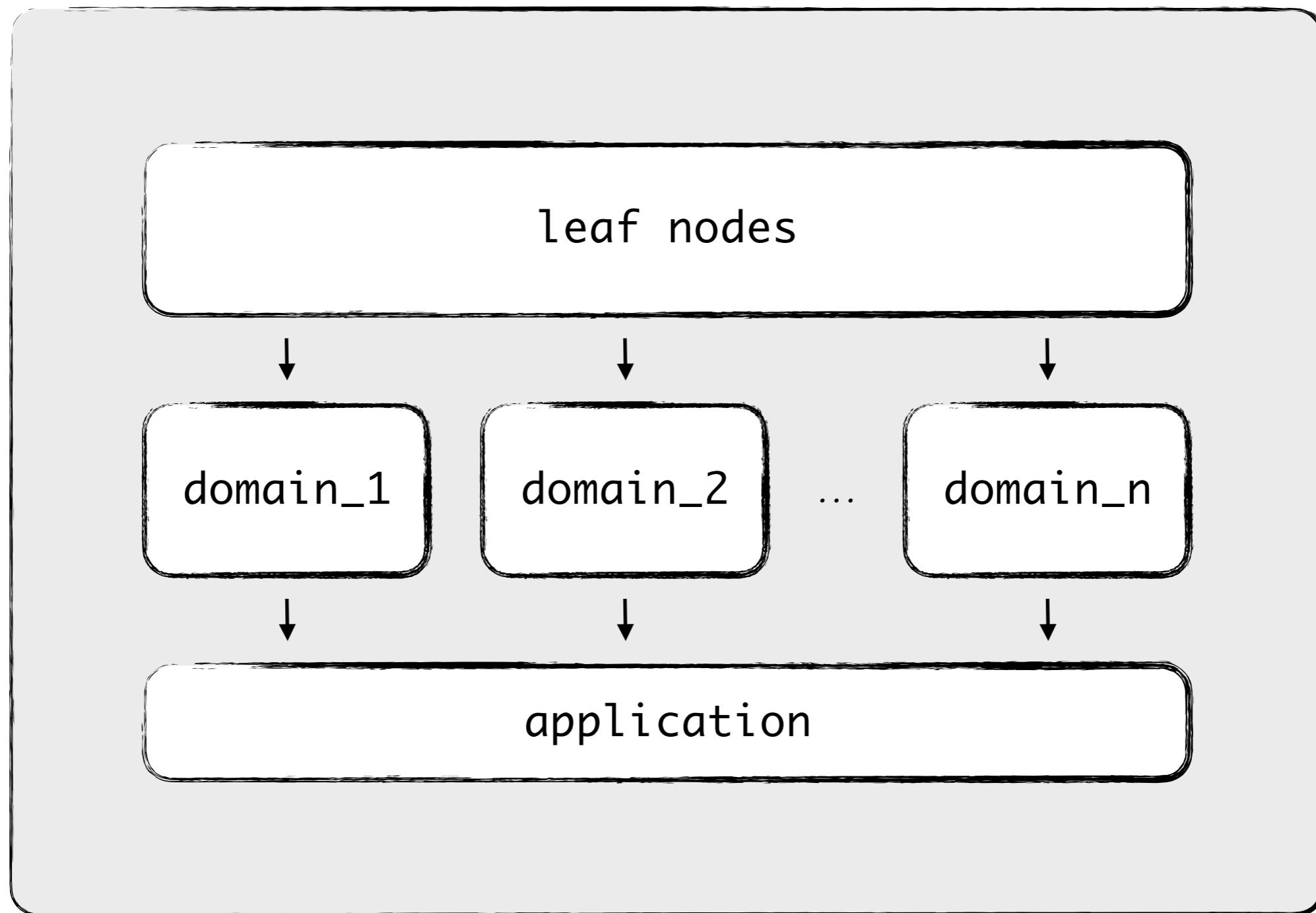
- Gorillaz

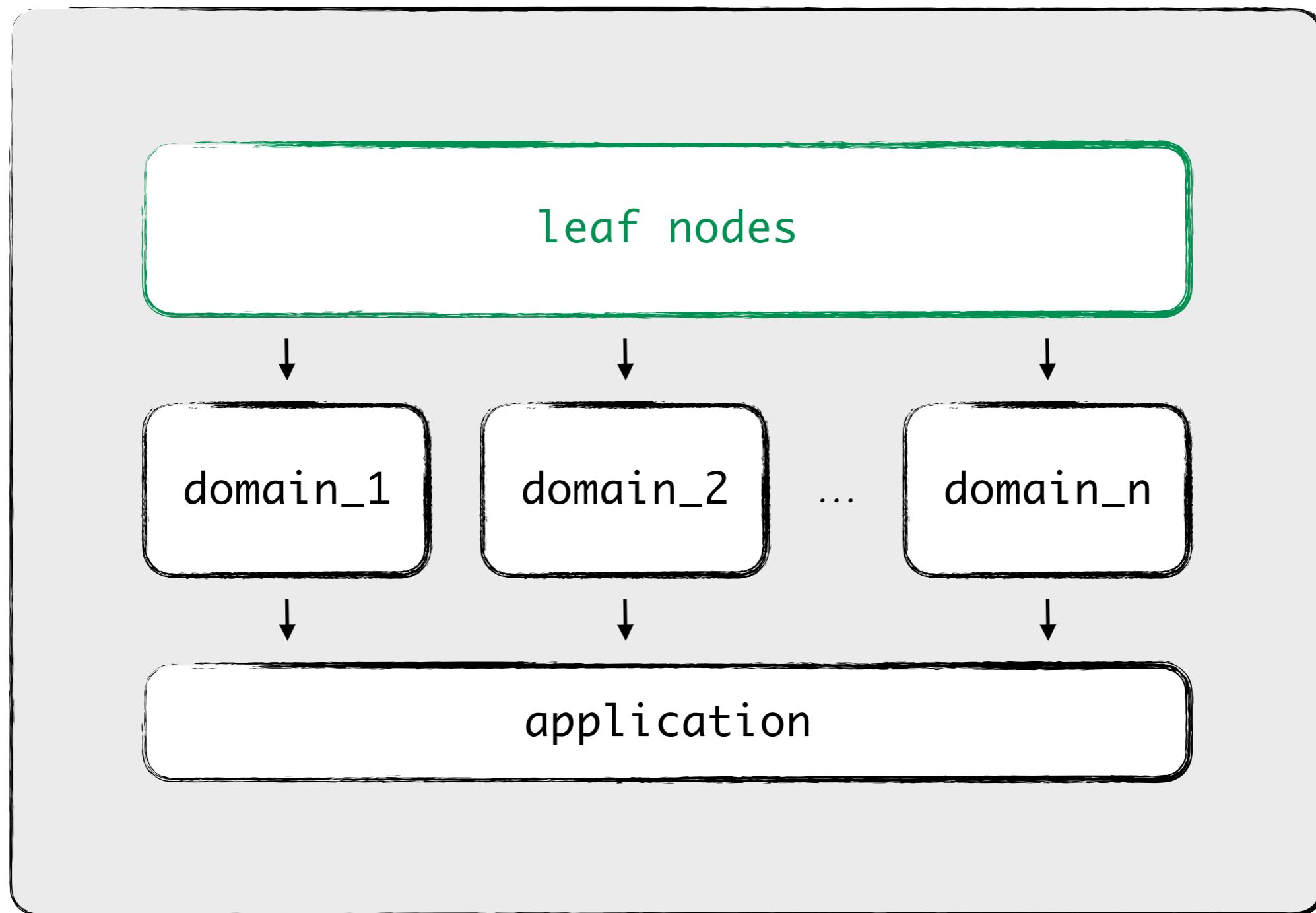
new setup

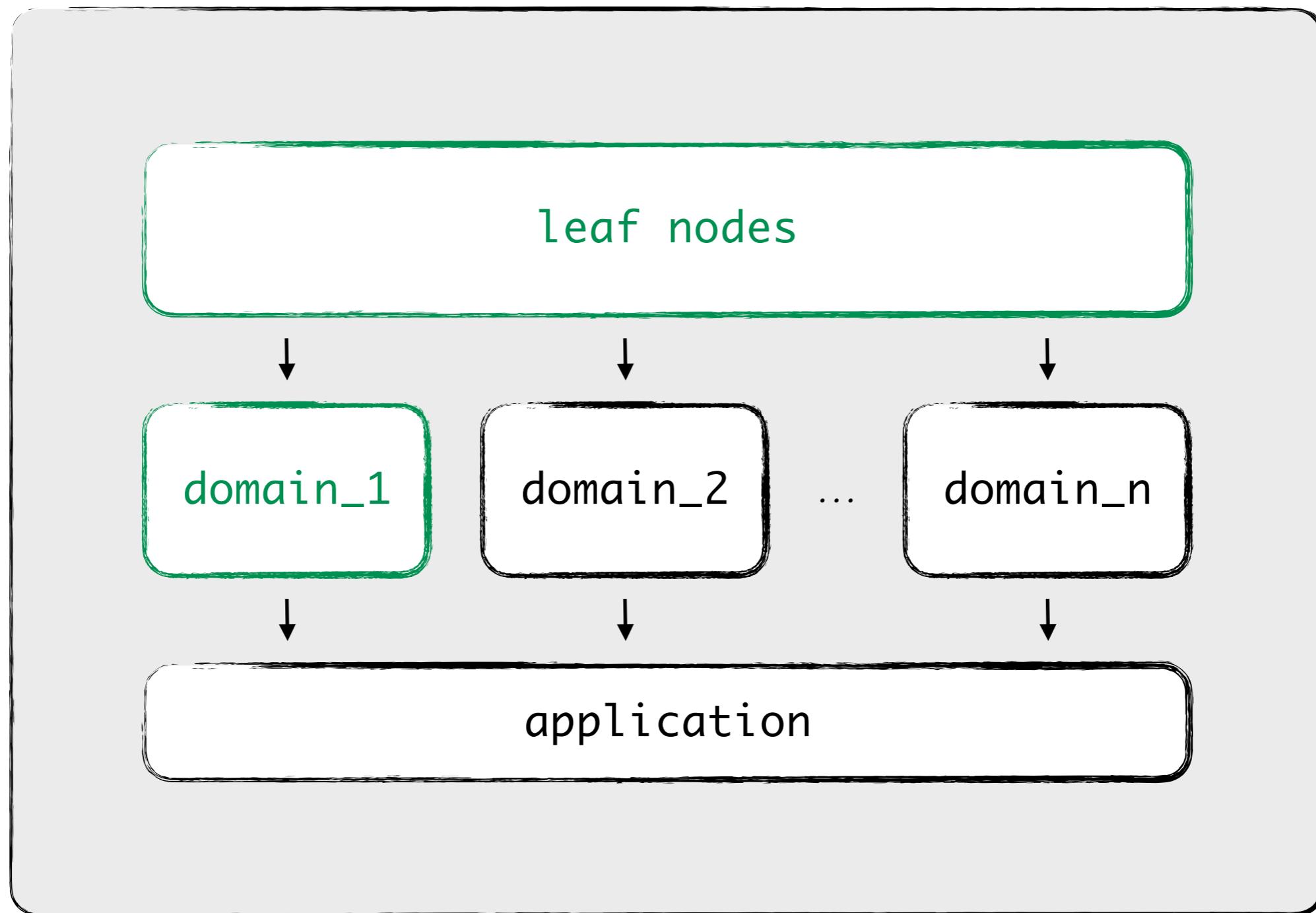
- lint, lint:fix, test
- --cache, concurrently
- Standardized configurations
- Relied on default as much as possible

recommended plugins

- `@tsconfig/ember`
- `eslint-plugin-n`
- `eslint-plugin-simple-import-sort`
- `eslint-plugin-typescript-sort-keys`
- `stylelint-order`







bypass CI

```
{  
  "scripts": {  
    "_lint:js": "eslint . --cache",  
    "_lint:js:fix": "eslint . --fix",  
    "lint:js": "exit 0"  
  }  
}
```

3b

spotting blockers

Some blockers are more equal than others.

deprecation workflow

● ● ● File: config/deprecation-workflow.js

```
1  const deprecationIdsToSilence = [
2    'implicit-injections',
3    'routing.transition-methods',
4    'this-property-fallback',
5  ];
6
7  self.deprecationWorkflow = self.deprecationWorkflow || {};
8
9  self.deprecationWorkflow.config = {
10    workflow: [
11      ...deprecationIdsToSilence.map((matchId) => ({
12        handler: 'silence',
13        matchId,
14      })),
15    ],
16  };

```

github.com/mixonic/ember-cli-deprecation-workflow

deprecations

- `implicit-injections` (v4)
- `this-property-fallback` (v4)
- `routing.transition-methods` (v5)

packages

- `ember-auto-import` (v2)
- `ember-modifier` (v3.2.7)
- `ember-template-lint` (v5)
- `eslint` (v8)
- `typescript` (v4.8.2)

3c

**short
(re)build**

Gotta go fast.

- A hedgehog

tackle build

- *Remove dead code*
- *Combine packages*
- *Remove cyclic dependencies*
- *Remove unused dependencies*
- *Simplify code consumed by many packages*

tackle rebuild

● ● ● File: ember-cli-build.js

```
1  const EmberApp = require('ember-cli/lib/broccoli/ember-app');
2
3  process.env.BROCCOLI_ENABLED_MEMOIZE = true;
4
5  module.exports = function (defaults) {
6    const app = new EmberApp(defaults, {
7      // Add options here
8    });
9
10   return app.toTree();
11 }
```

4

solving together

Things are better, so much better with two.

- Lenka

4a

refactors

Don't you ever feel like, like you've had enough?

- Wolfmother

what really matters

- *Write tests*
- *Rename things*
- *Make early exits*
- *Extract functions*
- *Remove dead code*

write tests

Get started:

```
ember g component-test <name>
```

Also available are `helper-test`, `modifier-test`, `service-test`, etc.

write tests



```
1 test('it renders', async function (assert) {
2   await render(hbs`  
3     <Hello @name="Zoey" />
4   `);
5   assert.ok(true);
7 });
```



```
1 test('it exists', function (assert) {
2   const chat = this.owner.lookup('service:chat');
3
4   assert.ok(chat);
5 })
```

Previously we discussed about examples of matrices represented by their sparsity patterns and their graphs.



WRITE TESTS LIKE A MATHEMATICIAN

The graph of a matrix often reveals a single global vs. structure, which might not be evident from the matrix structure. This matters the most.



So, what about, when is writing $\mathbf{A} \cdot \mathbf{v}$ (that is, in any of the four matrices that we have just displayed, but its graph,



tells a different story - it is nothing other than the cyclic matrix in discussed. To see this, take a look the vertices as follows:

$$1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 2, 3 \rightarrow 4, 4 \rightarrow 5, 5 \rightarrow 3.$$

This of course, is equivalent to considering (simultaneously) the conjugate and transpose. In this case, the matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called a *cycle* having the vertices in $\{1, 2, \dots, n\}$ and for every $i = 1, 2, \dots, n-1$ the set $\{i, i+1\} \cup \{n, 1\}$ contains exactly one member. It is a *simple path* if it does not visit any vertex more than once. We say that \mathbf{G} is a *tree* if each two members of \mathcal{V} are joined by a unique simple path. The *rooted* and *unrooted* definitions correspond to trees, but this is not the case with either quaternions or cyclic matrices when $n > 3$.

Suppose there is an arbitrary vector $\mathbf{v} \in \mathbb{R}^n$, for any $\mathbf{T} = (\mathbf{v}_i)$ is called a *rooted tree*, while \mathbf{v} is said to be the *root*. Unlike an ordinary graph, \mathbf{T} naturally carries partial ordering, which can best be explained by an analogy with a family tree. Thus, we say v is the *predecessor* of all the vertices in $\mathcal{V} \setminus \{v\}$ and v is *successor* of all the vertices of $\mathcal{V} \setminus \{v\}$. Moreover, every $u \in \mathcal{V} \setminus \{v\}$ is joined to v by a simple path and we designate each vertex along this path, except for v and u , as a *predecessor* of u and a *successor* of v . We say that the rooted tree \mathbf{T} is *monotonically ordered* if each vertex is labelled before all its predecessors. In other words, we layout the vertices along the axis of the tree to the root. (As we have already said, it is, redefining a graph to understand the meaning the rows and the columns of the underlying matrix.)

For example, consider the following matrix \mathbf{A} and its graph to get an idea of what we mean.



Theorem 1.1.3. Let \mathbf{A} be a *symmetric* matrix whose graph \mathbf{G} is a tree. Consider a root $v \in \{1, 2, \dots, n\}$ and assume that the rows and columns of \mathbf{A} have been arranged so that $\mathbf{T} = (\mathbf{v}_i)$ is *monotonically ordered*. Given that $\mathbf{A} = \mathbf{L}\mathbf{U}^T$ is a Cholesky factorization, it is true that

$$a_{i,j} = \frac{a_{i,i}}{a_{j,j}}, \quad k = j + 1, j + 2, \dots, n, \quad i = 1, 2, \dots, n-1. \quad (1.1.5)$$

rename things



```
1  /* Before */
2  let conListUniq;
3
4  /* After */
5  let contributorsList;
```

make early exits

```
1  /* Before */
2  function doSomething() {
3      if (condition1) {
4          if (condition2) {
5              // ...
6          }
7      }
8  }
9
10 /* After */
11 function doSomething() {
12     if (!condition1) return;
13     if (!condition2) return;
14
15     // ...
16 }
```

extract functions

```
1  function convertV1AddonToV2(codemodOptions) {  
2    const options = createOptions(codemodOptions);  
3    const context = analyzeAddon(options);  
4  
5    // Preserve code  
6    moveAddonFiles(options);  
7    moveTestAppFiles(options);  
8    moveProjectRootFiles(options);  
9  
10   // Get the latest code from blueprints  
11   createFilesFromBlueprints(context, options);  
12  
13   // ...  
14 }
```

remove dead code

Get started:

```
git grep -in <pattern>
```

4b

codemods

Everything everywhere all at once.

things to consider

- *Upfront costs for manifold returns*
- *Edge cases due to lacking code standards*
- *Can I cover the 80% case?*

@codemod-utils

blueprint-for-v2-addon
(CLARK internal)

*ember-codemod-
args-to-signature*

*ember-codemod-
pod-to-octane*

*ember-codemod-
remove-ember-css-modules*

*ember-codemod-
v1-to-v2*

type-css-modules

github.com/ijlee2/codemod-utils

@codemod-utils

Get started:

```
npx @codemod-utils/cli <arguments>
```

github.com/ijlee2/codemod-utils

4c

different perspectives

I think I understand you but I don't.

- OK Go



Sterbehilfe-Beraterin trifft Ärztin

"Sag's mir" mit diesen Gästen: Helga Liedtke, Sterbehilfe-Begleiterin und Ute Lewitzka, Fachärztin für Psychiatrie und Psychotherapie

[Sag's mir bei YouTube](#)



Ist es richtig, Cannabis zu legalisieren?

Welche Auswirkungen hat das für Jugendliche? Ist die Legalisierung von Cannabis das richtige Signal für junge Menschen? Wird der illegale Handel dadurch eingedämmt?

21 min · Kultur



Schluss mit Massentourismus?

Massentourismus führt zu Umweltschäden, Ressourcenknappheit und der Ausnutzung von Abhängigkeitsverhältnissen. Können wir mit gutem Gewissen Richtung Süden reisen?

41 min · Kultur



zdf.de/kultur - Sag's mir, Unter Anderen, 13 Fragen

5

**future
is now**

Bending time.

- Bender Bending Rodríguez

polyfills

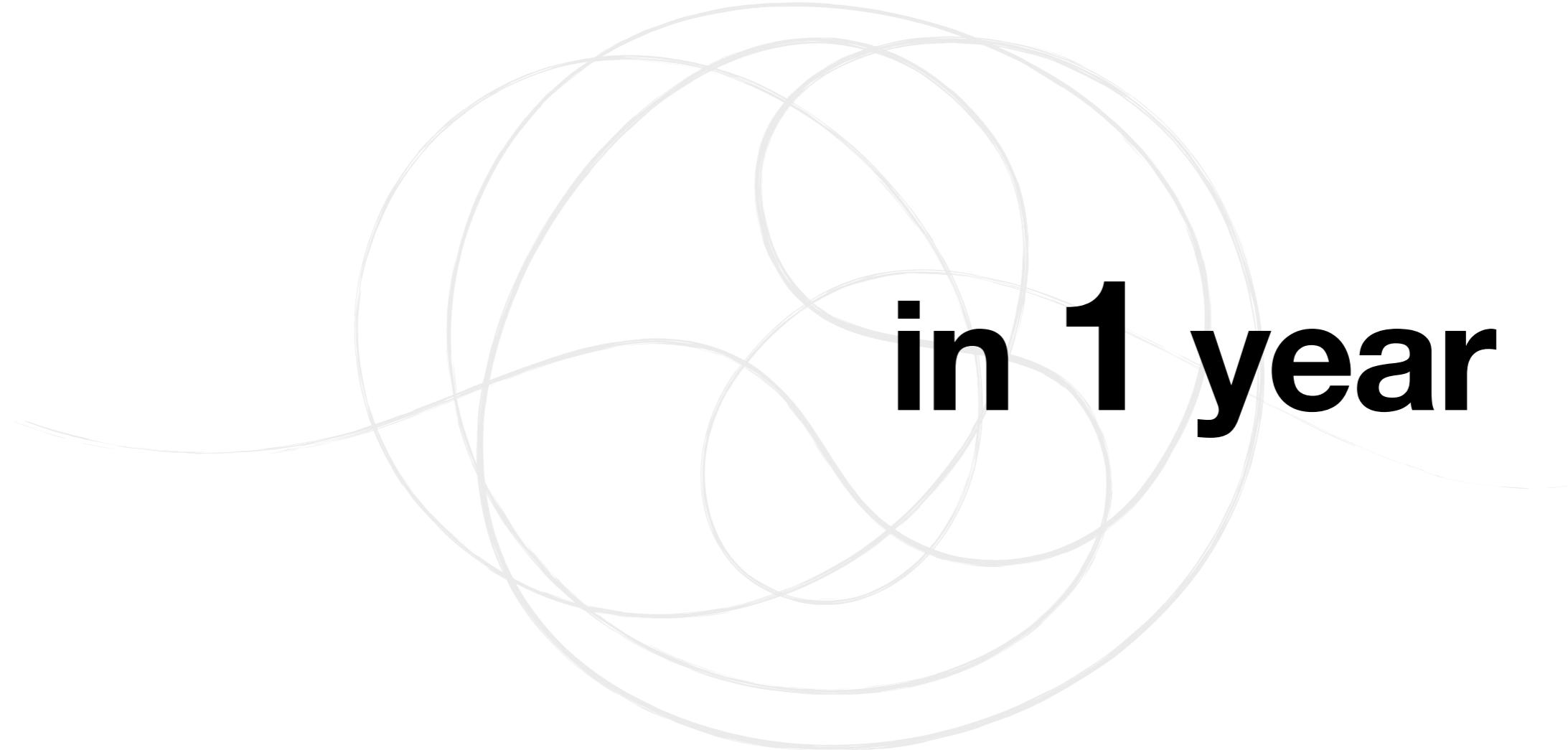
- ember-angle-bracket-invocation-polyfill
- ember-cached-decorator-polyfill
- ember-functions-as-helper-polyfill
- ember-in-element-polyfill
- ember-named-blocks-polyfill
- ember-on-modifier
- ember-unique-id-helper-polyfill

v1 addons

- *Enable `ember-unsafe`*
- *Enable `ember-optimized` (skip if hard to do)*
- *Support Glint by creating `addons/template-registry.ts`*
- *Support `<template>`-tag by re-exporting the public API in `addons/index.{js,ts}`*
- *Run `ember-codemod-v1-to-v2` to start migrating to v2*

package registry

- *Extracted packages from monorepo*
- *Converted addons to v2*
- *Introduced **embroider-css-modules***
- *Supported Glint and <template>-tag*
- *Created test apps*



in 1 year

github.com/ijlee2