



**moving  
mountains**

[github.com/ijlee2](https://github.com/ijlee2)

# problems with old repo

*~190 pkgs. (105 addons)*

*slow CI/CD*

*yarn v1*

*stuck on Ember v3.28*

*tight coupling*

*poor documentation*

# goals for new repo

*as few pkgs. as possible*

*fast CI, publish*

*pnpm*

*ember-try*

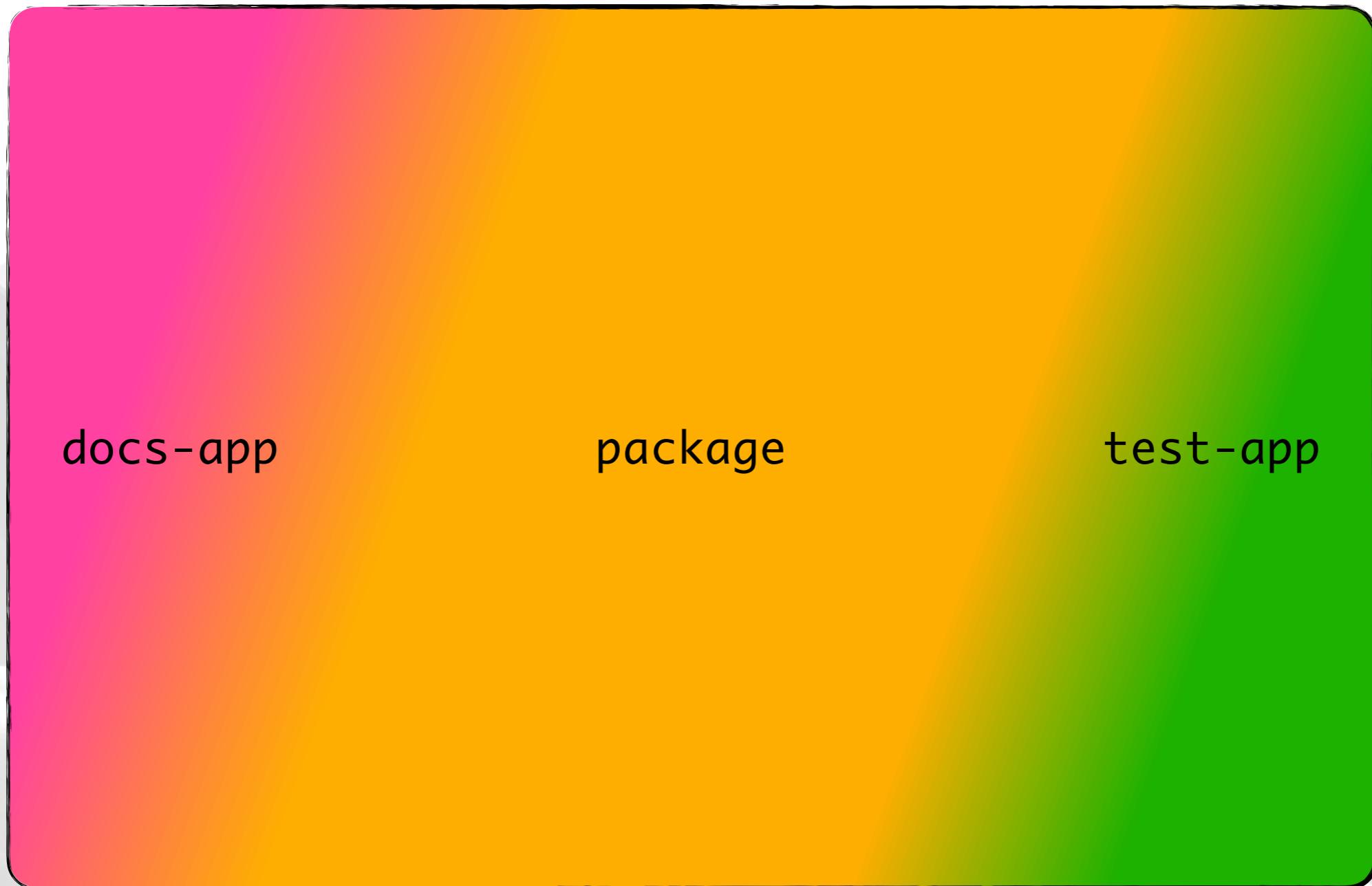
*clean separation*

*glint, docs-app, test-app*

# ingredients

- 1 package registry
- 1 tsp. knowledge of addons
- 1 tsp. knowledge of monorepos
- 1 t. belief in self and others

my-repo



docs-app

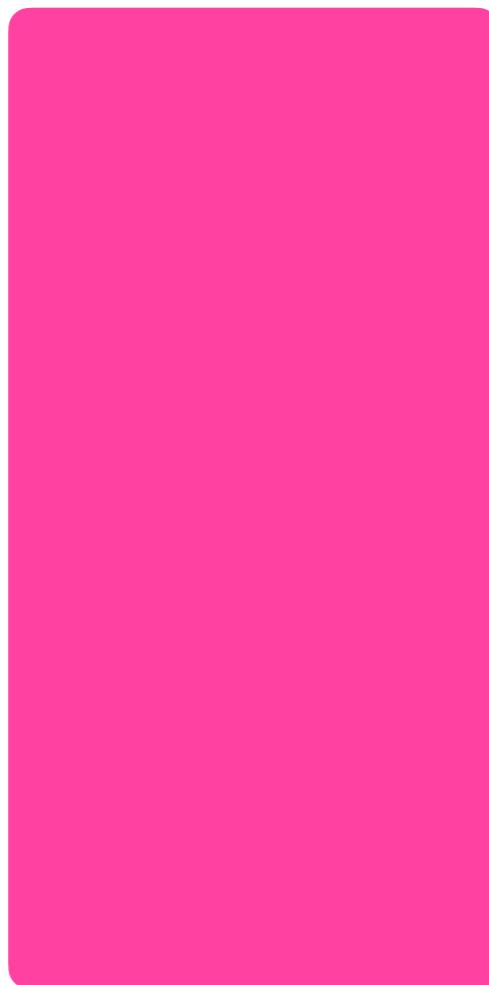
package

test-app

my-repo

---

docs-app



package



test-app



my-repo

---

docs-app



packages



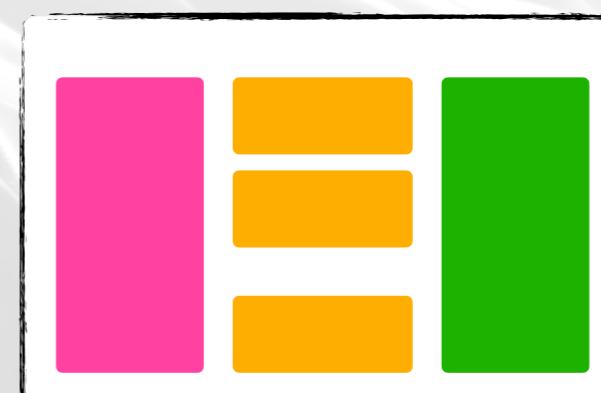
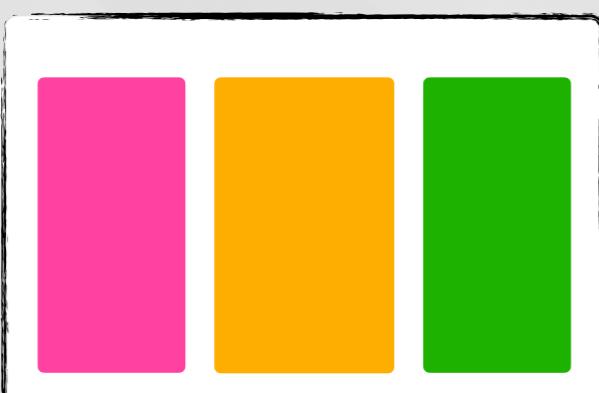
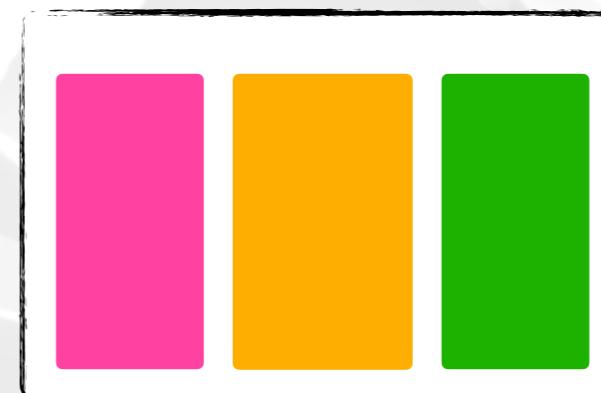
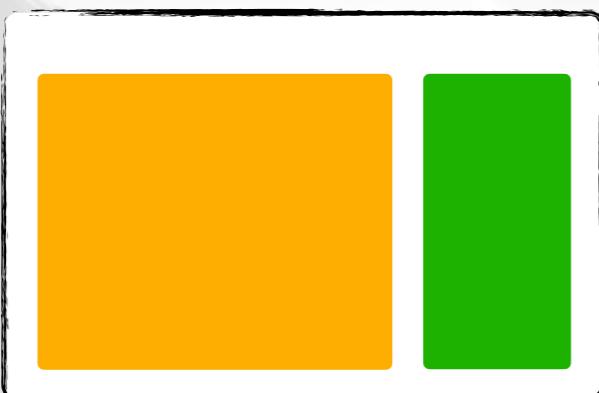
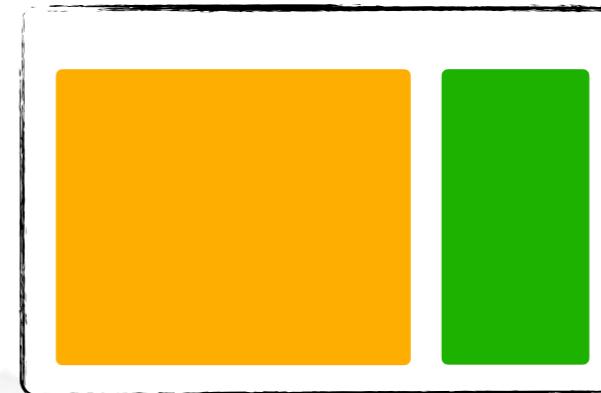
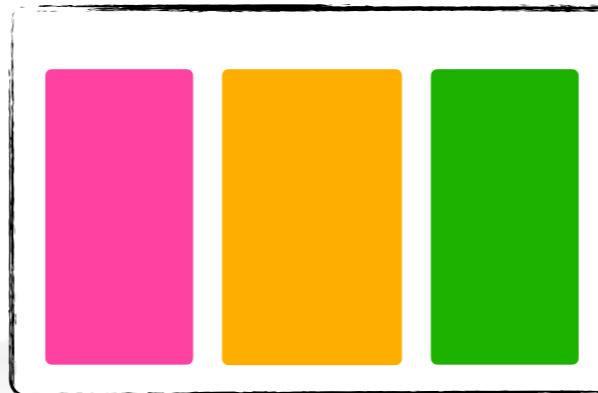
⋮



test-app



## all-our-repos



1

moving  
one addon

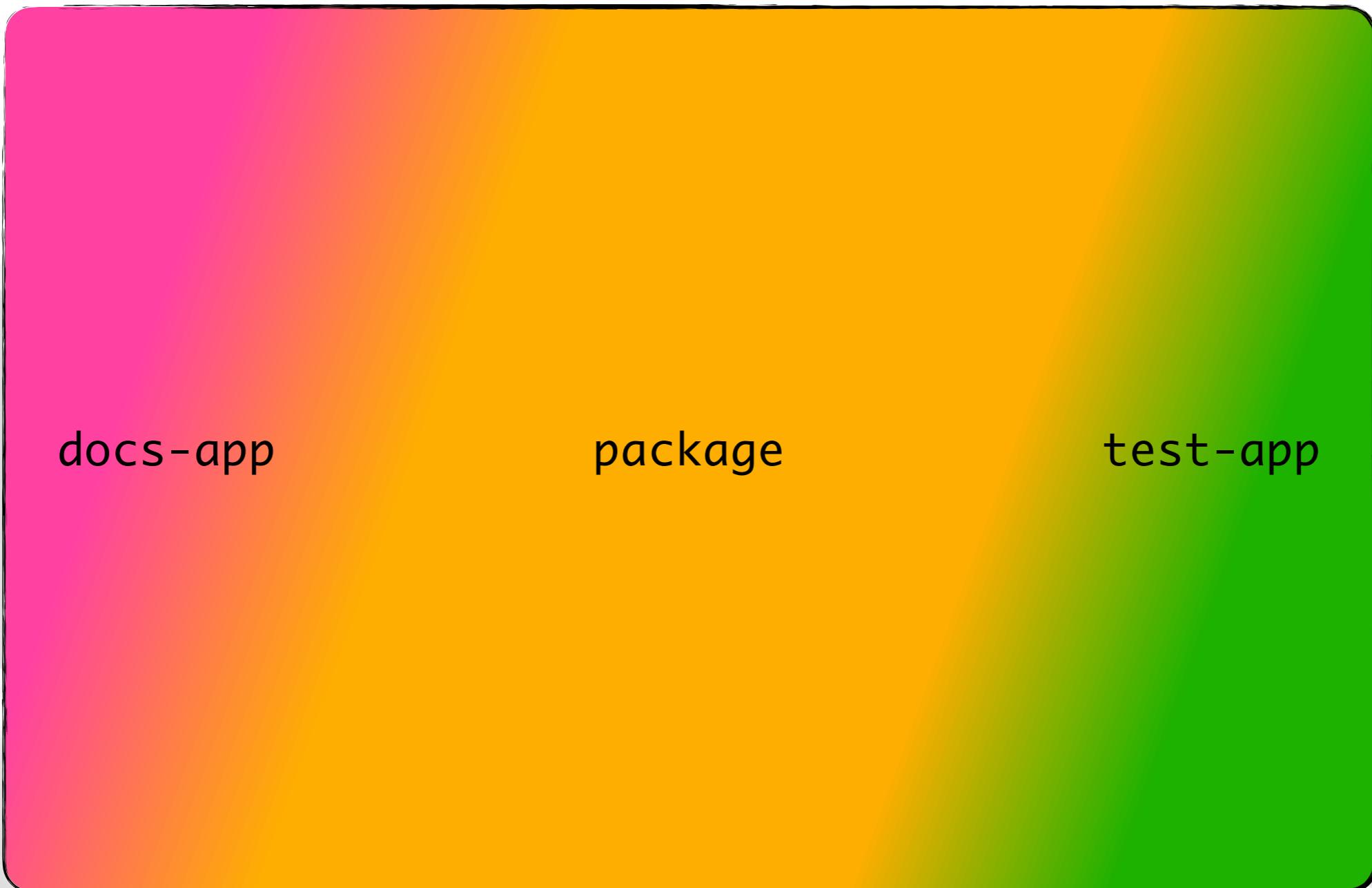
**1 a**

**en route  
to v2**

# outcomes

- *Separate responsibilities*
- *Use pnpm to create monorepo*
- *Convert addon to v2*

my-repo



docs-app

package

test-app

# my-repo

└── addon		====	
└── app		====	
└── tests			
└── dummy	██████████		██████████
└── helpers	██████████		██████████
└── integration	██████████		██████████
└── unit	██████████		██████████
└── types	██████████	====	██████████
└── ember-cli-build.js	██████████		██████████
└── index.js		====	
└── package.json	██████████	====	██████████
└── tsconfig.json	██████████	====	██████████

# my-repo

## package (v1)

```
├── addon  
├── app  
├── tests (unused)  
├── ...  
└── package.json
```

## test-app

```
├── app  
├── tests  
├── ...  
└── package.json
```

# my-repo

---

## docs-app

```
├── app  
├── tests  
├── ...  
└── package.json
```

## package (v1)

```
├── addon  
├── app  
├── tests (unused)  
├── ...  
└── package.json
```

## test-app

```
├── app  
├── tests  
├── ...  
└── package.json
```

# my-repo

---

## docs-app

```
├── app  
├── tests  
├── ...  
└── package.json
```

## package (v2)

```
├── src  
├── ...  
└── package.json
```

## test-app

```
├── app  
├── tests  
├── ...  
└── package.json
```

# pnpm

- *Strict about dependencies*
- *Favored by Embroider apps and v2 addons*
- `pnpm <outdated|publish|update|why> -r`

● ● ● pnpm-workspace.yaml

```
1 packages:
2   - docs-app
3   - packages/my-addon
4   - test-app
```

● ● ● test-app: package.json

```
1 {
2   "name": "test-app",
3   "devDependencies": {
4     "ember-cli": "~5.11.0",
5     "ember-source": "~5.11.0",
6     "my-addon": "workspace:*
```

# my-repo

---

docs-apps



⋮



package



test-apps



⋮



# one addon, many apps

-  *Test multiple setups*
-  *Test multiple screen resolutions (maybe?)*
-  *Test multiple versions of a dependency*

# recipe 1

1. *Create a monorepo*
2. *Create a docs-app, v2 addon, and test-app*
3. *Set up lint, test, CI, and documentation*
4. *Move code over*

# recipe 2

Get started:

```
npx ember-codemod-v1-to-v2
```

[github.com/ijlee2/ember-codemod-v1-to-v2](https://github.com/ijlee2/ember-codemod-v1-to-v2)

# recipe 3

1. *Let someone else create a v2 addon repo?*
2. *Move code over*

1 b

backward  
compatibility

# my-repo

## package (v1)

```
├── addon
|   ├── components
|   ├── helpers
|   ├── modifiers
|   ├── services
|   └── utils
├── app
|   ├── components
|   ├── helpers
|   ├── modifiers
|   ├── services
|   └── utils
└── ...
└── package.json
```

## package (v2)

```
├── src
|   ├── components
|   ├── helpers
|   ├── modifiers
|   ├── services
|   └── utils
└── ...
└── package.json (*)
└── rollup.config.mjs (*)
```

● ● ● my-addon: src/components/hello.hbs

```
1 <div>
2 | Hello!
3 </div>
```

● ● ● my-addon: src/components/hello.ts

```
1 import Component from '@glimmer/component';
2
3 export default class HelloComponent extends Component {}
```

my-addon: package.json

```
1  {
2    "name": "my-addon",
3    "files": [
4      "addon-main.cjs",
5      "declarations",
6      "dist"
7    ],
8    "ember-addon": {
9      "app-js": {
10        "./components/hello.js": "./dist/_app_/hello.js"
11      }
12    }
13  }
```

```
● ● ● my-addon: rollup.config.mjs
1  const addon = new Addon({
2    srcDir: 'src',
3    destDir: 'dist',
4  });
5
6  export default {
7    plugins: [
8      addon.appReexports([
9        'components/**/*.js',
10       'helpers/**/*.js',
11       'modifiers/**/*.js',
12       'services/**/*.js',
13       'utils/**/*.js',
14     ]),
15   ],
16};
```

# outcomes

- *Provide barrel file + template registry*
- *Provide styles*
- *Provide translations*
- *Provide assets*
- *Provide test helpers*

# my-repo

## package (v1)

```
└── addon
    ├── components
    ├── index.ts
    └── template-registry.ts
└── ...
└── package.json
```

## package (v2)

```
└── src
    ├── components
    ├── index.ts
    └── template-registry.ts
└── ...
└── package.json
└── rollup.config.mjs (*)
```

● ● ● my-addon: src/index.ts

```
1 export { default as Hello } from './components/hello.ts';
```

● ● ● docs-app: app/components/demo-hello.gts

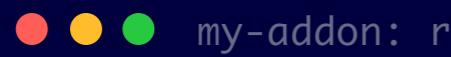
```
1 import { Hello } from 'my-addon';
2
3 <template>
4   <Hello />
5 </template>
```

● ● ● my-addon: src/template-registry.ts

```
1 import type HelloComponent from './components/hello.ts';
2
3 export default interface MyAddonRegistry {
4   Hello: typeof HelloComponent;
5   hello: typeof HelloComponent;
6 }
```

● ● ● docs-app: app/components/demo-hello.hbs

```
1 <Hello />
```



```
1  export default {
2    plugins: [
3      addon.publicEntryPoints([
4        '**/*.js',
5        'index.ts',
6        'template-registry.ts',
7        ]),
8      ],
9    };
```

# my-repo

## package (v1)

```
├── addon
|   └── components
|       ├── hello.css
|       ├── hello.css.d.ts
|       ├── hello.hbs
|       └── hello.ts
├── ...
└── package.json
```

## package (v2)

```
├── src
|   └── components
|       ├── hello.css
|       ├── hello.css.d.ts
|       ├── hello.hbs (*)
|       └── hello.ts (*)
├── ...
└── package.json
    └── rollup.config.mjs (*)
```

```
● ● ● my-addon: src/components/hello.hbs
1 <div class="{{this.styles.container}}>
2 | Hello!
3 </div>
```

```
● ● ● my-addon: src/components/hello.ts
1 import Component from '@glimmer/component';
2
3 import styles from './hello.css';
4
5 export default class HelloComponent extends Component {
6   |   styles = styles;
7 }
```

```
● ● ● my-addon: rollup.config.mjs
1 import postcss from 'rollup-plugin-postcss';
2
3 export default {
4   plugins: [
5     postcss({
6       autoModules: false,
7       modules: {
8         generateScopedName: 'my-addon__[sha512:hash:base64:5]',
9       },
10      }),
11    ],
12  };

```

# my-repo

## package (v1)

```
├── addon  
├── translations  
|   ├── de-de.yaml  
|   └── en-us.yaml  
├── ...  
└── package.json
```

## package (v2)

```
├── src  
├── translations  
|   ├── de-de.yaml  
|   └── en-us.yaml  
├── ...  
└── package.json (*)  
    └── rollup.config.mjs
```



```
1 <div class="{{this.styles.container}}>
2   {{t "hello.message"}}
3 </div>
```



```
1  {
2    "name": "my-addon",
3    "files": [
4      "addon-main.cjs",
5      "declarations",
6      "dist",
7      "translations"
8    ]
9  }
```

# my-repo

## package (v1)

```
├── addon  
├── public  
|   └── assets  
|       └── zoey.png  
├── ...  
└── package.json
```

## package (v2)

```
├── public  
|   └── assets  
|       └── zoey.png  
├── src  
├── ...  
└── package.json (*)  
    └── rollup.config.mjs (*)
```



```
my-addon: src/components/hello.hbs
1  <div class="{{this.styles.container}}>
2    {{t "hello.message"}}
3
4    
9  </div>
```

my-addon: package.json

```
1  {
2    "name": "my-addon",
3    "files": [
4      "addon-main.cjs",
5      "declarations",
6      "dist",
7      "public",
8      "translations"
9    ],
10   "ember-addon": {
11     "public-assets": {
12       "./public/assets/zoeys.png": "/my-addon/zoeys.png"
13     }
14   }
15 }
```

● ● ● my-addon: rollup.config.mjs

```
1  export default {
2    plugins: [
3      addon.publicAssets('public/assets', {
4        exclude: ['**/*'],
5        namespace: 'my-addon',
6      }),
7    ],
8  };
```

# my-repo

## package (v1)

```
├── addon
├── addon-test-support
|   ├── components
|   |   └── hello.ts
|   └── index.ts
├── ...
└── package.json
```

## package (v2)

```
├── src
|   ├── test-support
|   |   └── components
|   |       └── hello.ts
|   └── test-support.ts
├── ...
└── package.json
└── rollup.config.mjs (*)
```

● ● ● my-addon: src/test-support/components/hello.ts

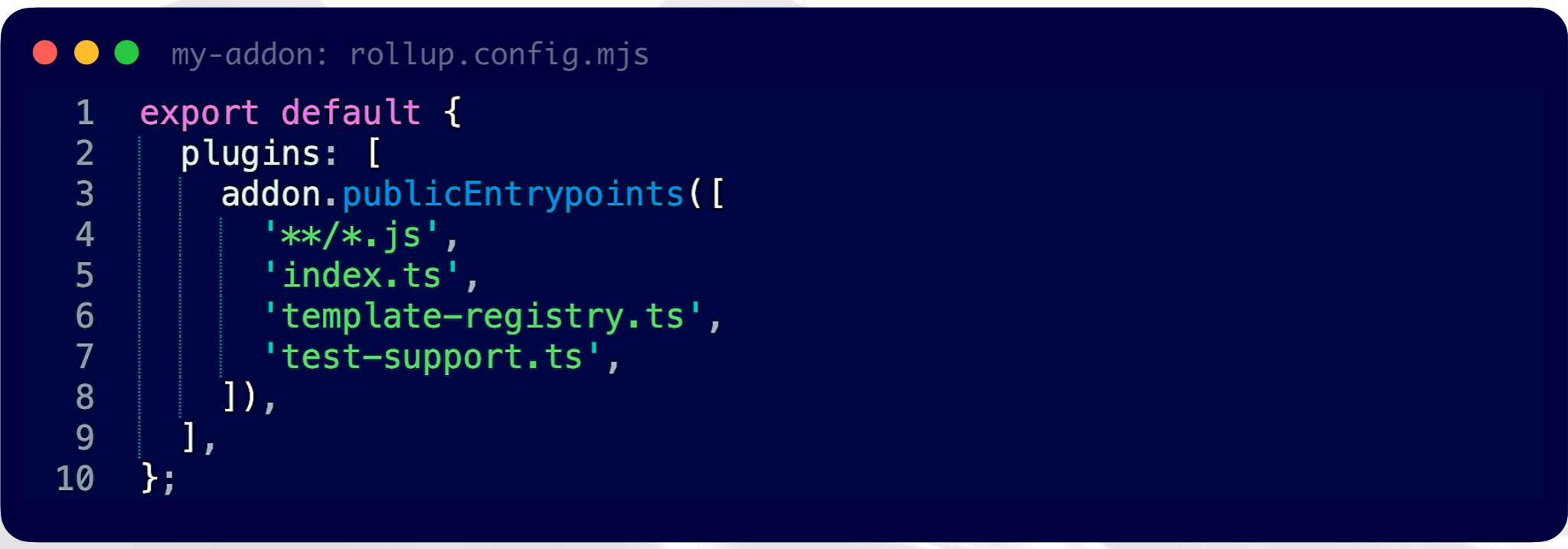
```
1  export function assertZoeyExists(assert: Assert) {  
2    assert.dom('img').hasAttribute('src', '/my-addon/zoey.png');  
3  }
```

● ● ● test-app: tests/integration/components/hello-test.gts

```
1  import { render } from '@ember/test-helpers';  
2  import { Hello } from 'my-addon';  
3  import { assertZoeyExists } from 'my-addon/test-support';  
4  import { module, test } from 'qunit';  
5  import { setupRenderingTest } from 'test-app/tests/helpers';  
6  
7  module('Integration | Component | hello', function (hooks) {  
8    setupRenderingTest(hooks);  
9  
10   test('it renders', async function (assert) {  
11     await render(<template><Hello /></template>);  
12  
13     assertZoeyExists(assert);  
14   });  
15 });
```

```
● ● ● my-addon: unpublished-development-types/index.d.ts

1 import '@glint/environment-ember-loose';
2 import '@glint/environment-ember-template-imports';
3 import 'ember-source/types';
4 import 'qunit-dom';
5
6 import type MyAddonRegistry from '../src/template-registry.ts';
7
8 declare module '@glint/environment-ember-loose/registry' {
9   export default interface Registry extends MyAddonRegistry {}
10 }
```



```
● ● ● my-addon: rollup.config.mjs
1  export default {
2    plugins: [
3      addon.publicEntryPoints([
4        '**/*.js',
5        'index.ts',
6        'template-registry.ts',
7        'test-support.ts',
8        ]),
9      ],
10    };
```

# takeaways

- *Incremental migration is easy*
- *Project configuration is hard*
- *Let someone else create a v2 addon repo?*



2

moving  
many addons

**2a**

**new  
problems**

# outcomes

- *How to set up the repo?*
- *Which addon (or file) to extract first?*
- *How to find missing and unused dependencies?*
- *How to release many addons but create one tag?*
- *How to make all addons look the same?*

my-repo

---

docs-app



package



test-app



my-repo

---

docs-app



packages



⋮



test-app



# my-repo

docs-apps



⋮



packages



⋮



test-apps



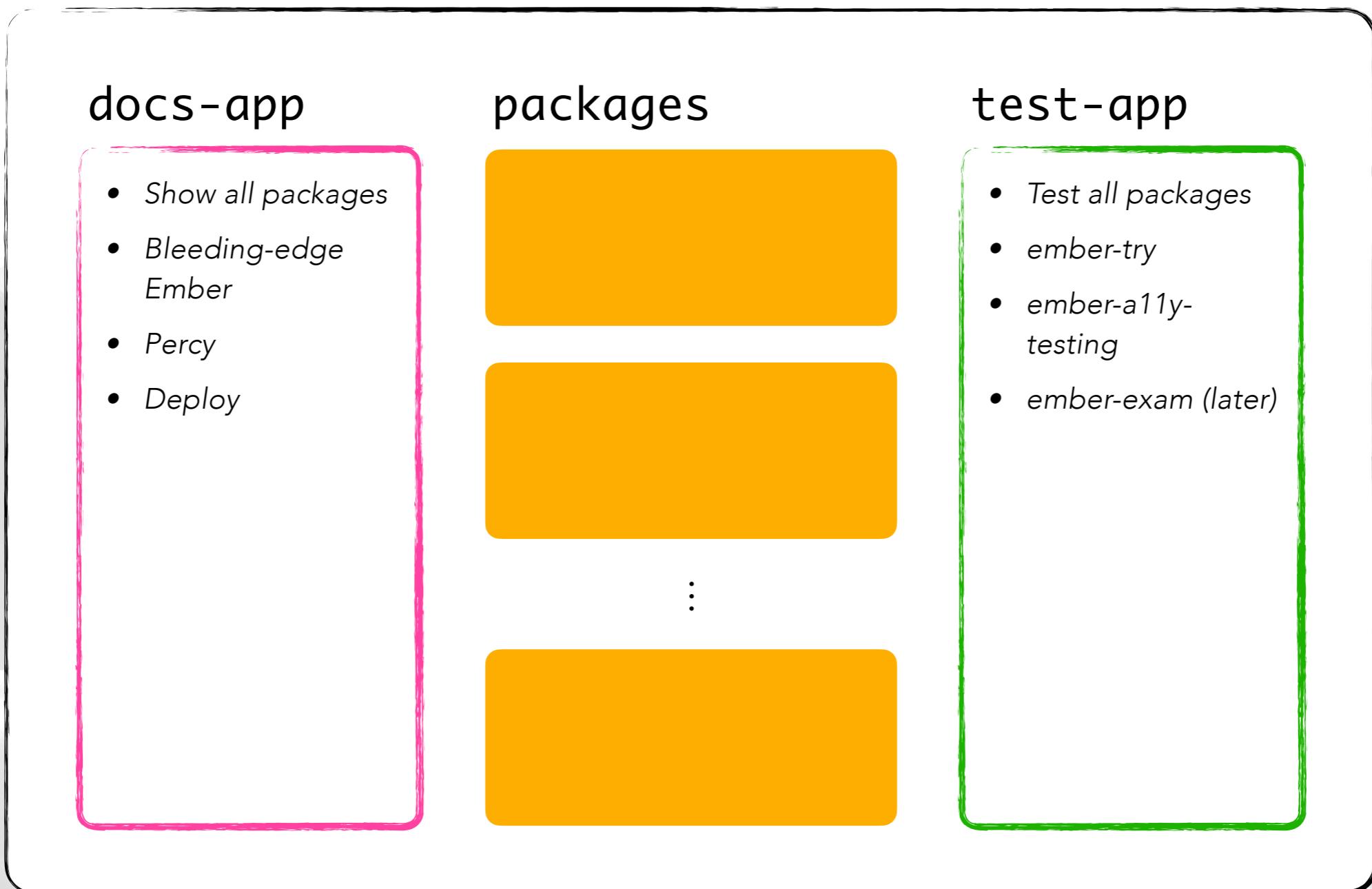
⋮



# the right code

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

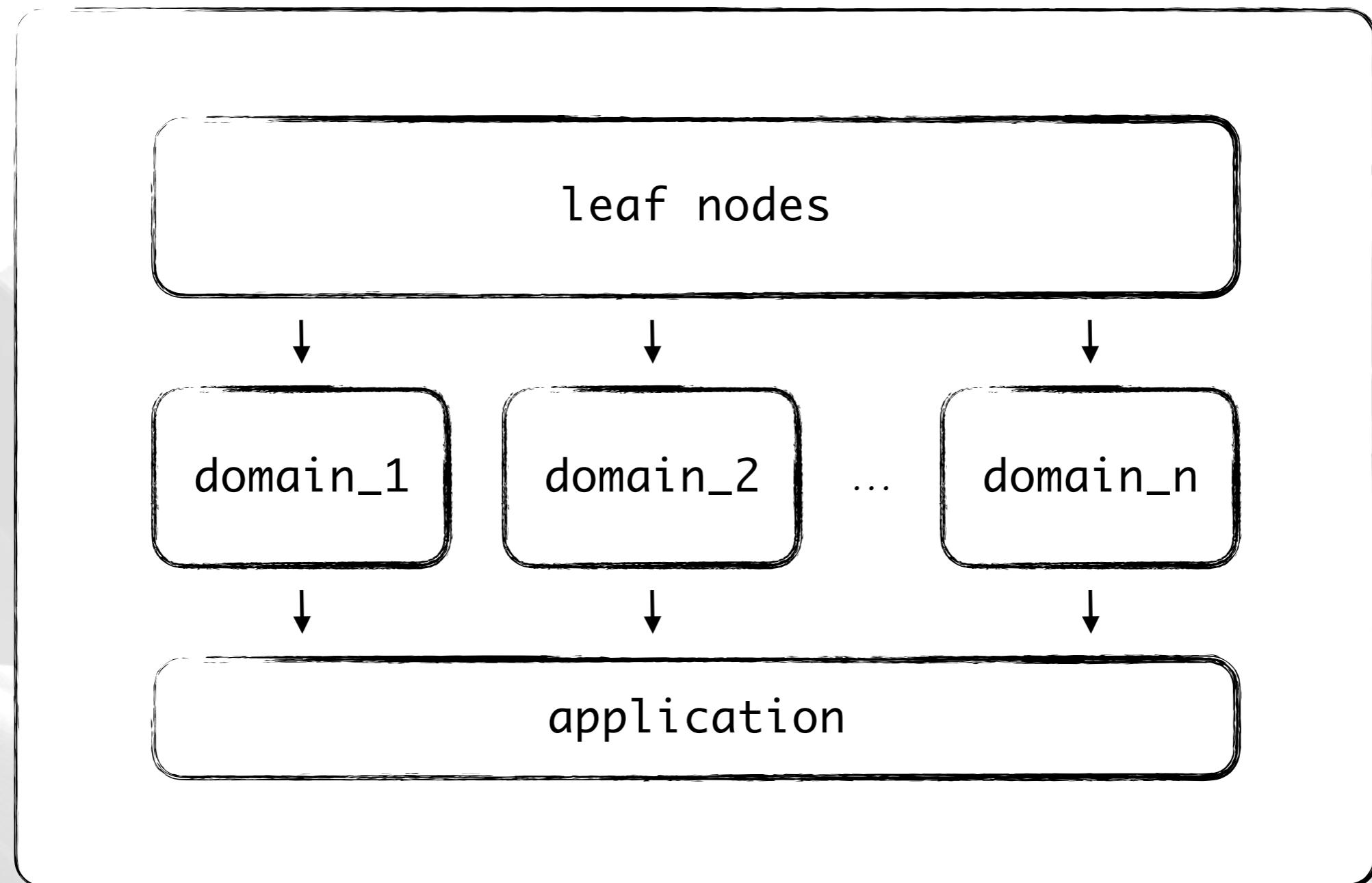
# my-repo



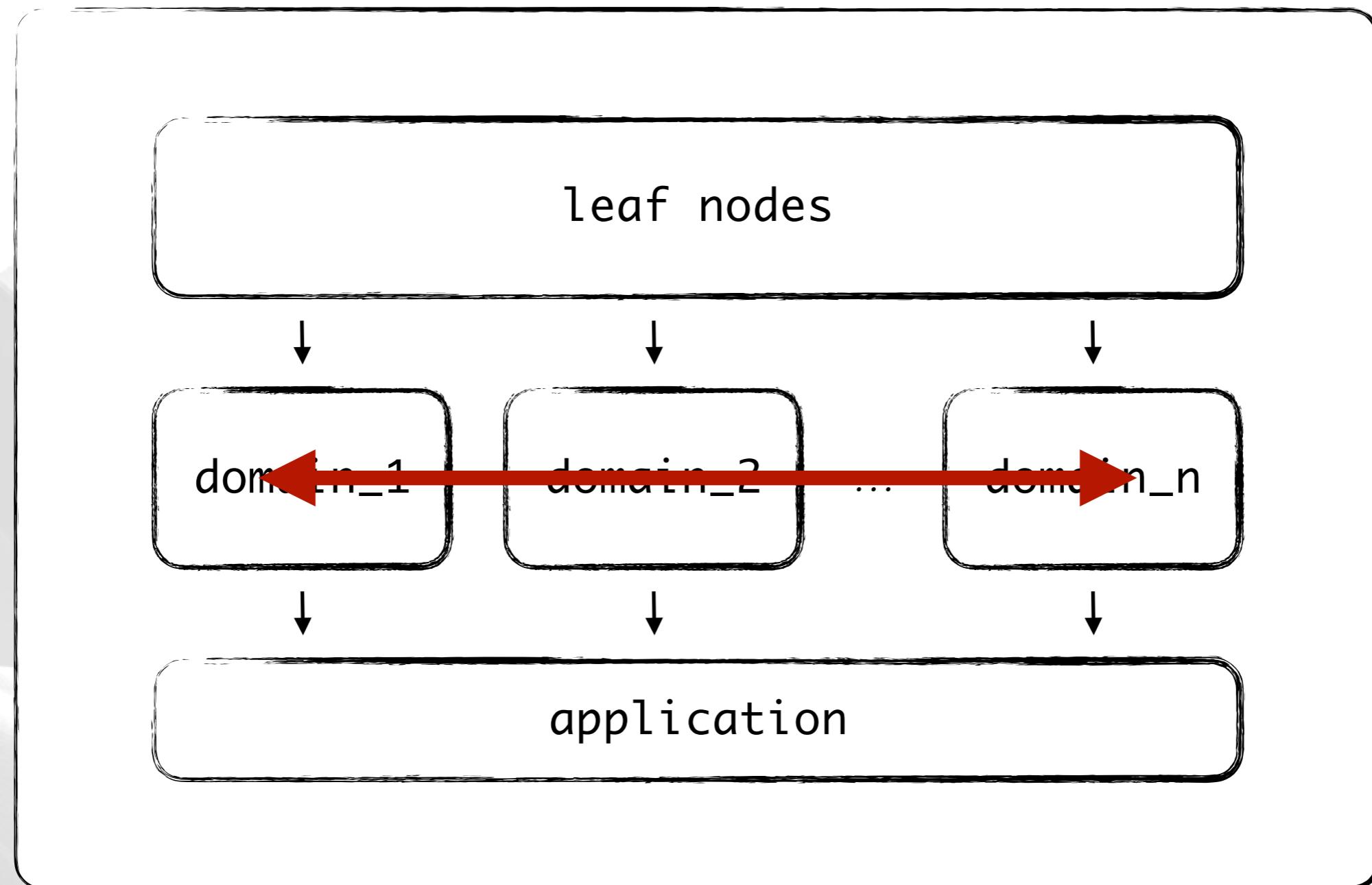
● ● ● package.json

```
1  {
2    "name": "workspace-root",
3    "private": true,
4    "scripts": {
5      "build": "pnpm --filter './packages/**' build",
6      "lint": "pnpm --filter '*' lint",
7      "lint:fix": "pnpm --filter '*' lint:fix",
8      "start": "concurrently 'pnpm:start:*'",
9      "start:docs-app": "pnpm --filter docs-app start",
10     "start:test-app": "pnpm --filter test-app start",
11     "test": "pnpm --filter docs-app --filter test-app test"
12   },
13   "devDependencies": {
14     "concurrently": "^9.0.0"
15   }
16 }
```

my-old-repo



my-old-repo



# checklist (ranking)

## API

- *is classic component*
- *is template-only*
- *# of actions*
- *# of arguments*

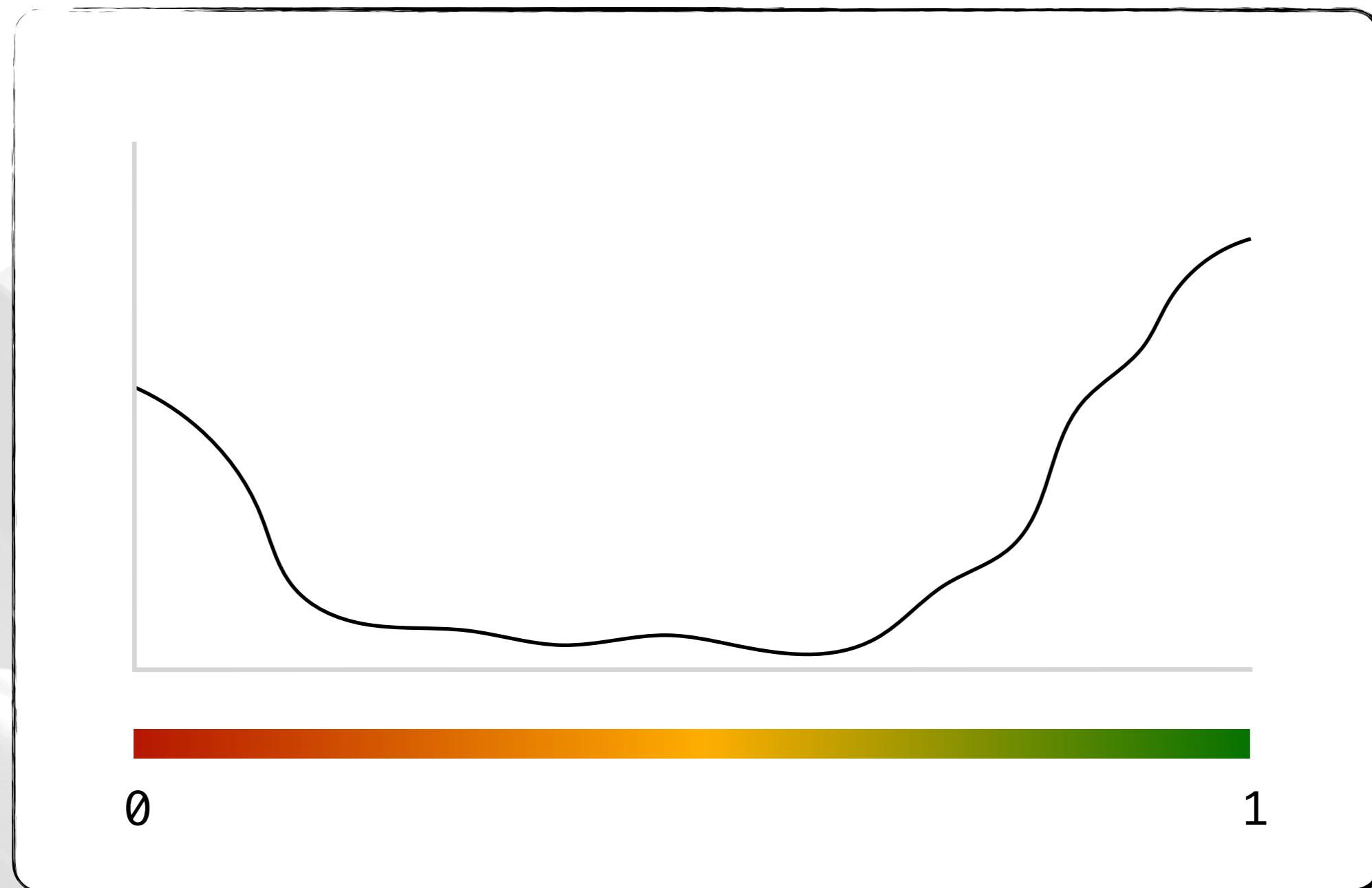
## SoC

- *has class attribute (global styles)*
- *has named blocks*
- *has splattributes*
- *# of lines of code*

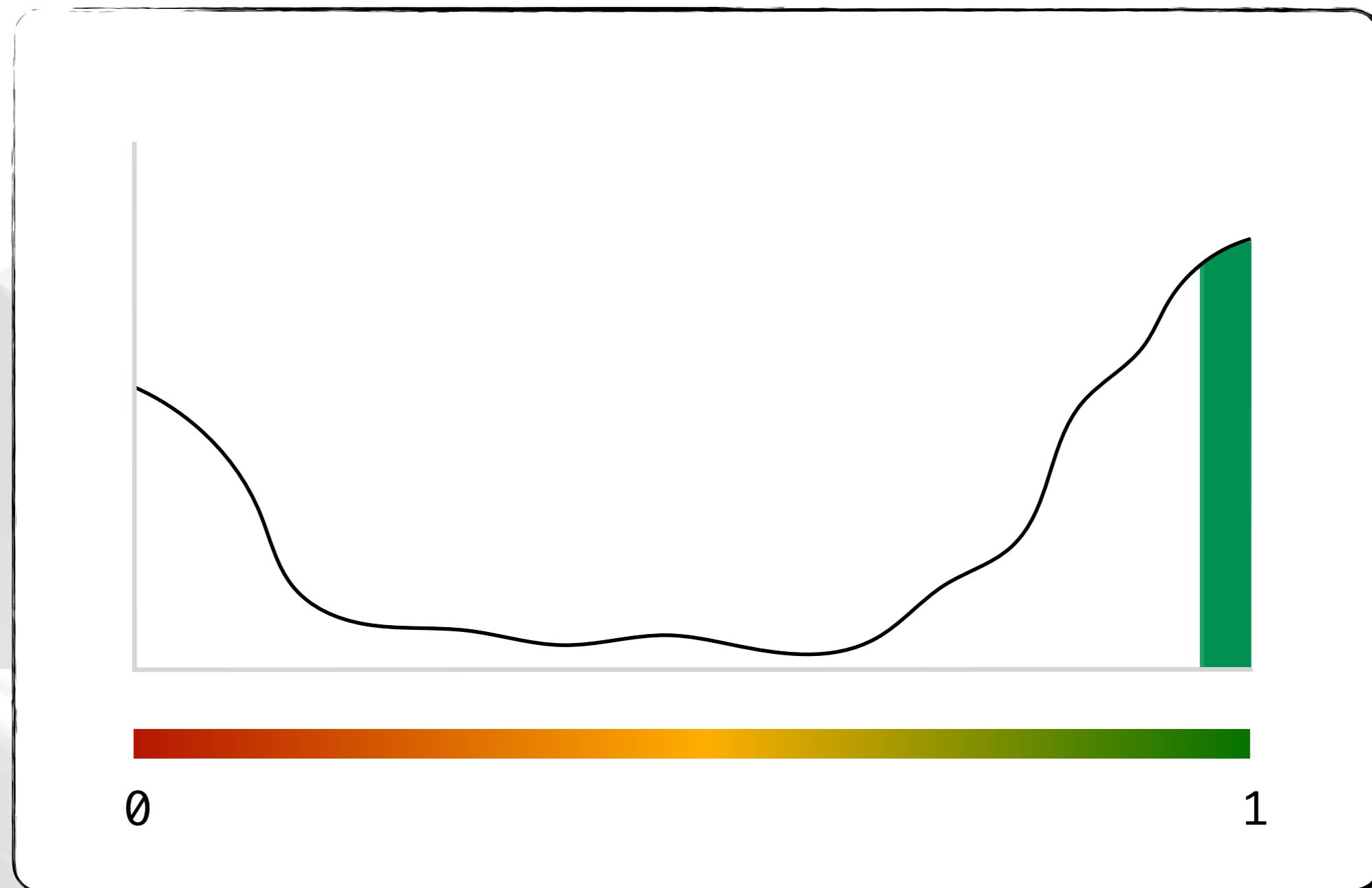
## Dependencies

- *has components*
- *has helpers*
- *has modifiers*
- *has services*
- *has utilities*
- *# of imports (3rd-party)*

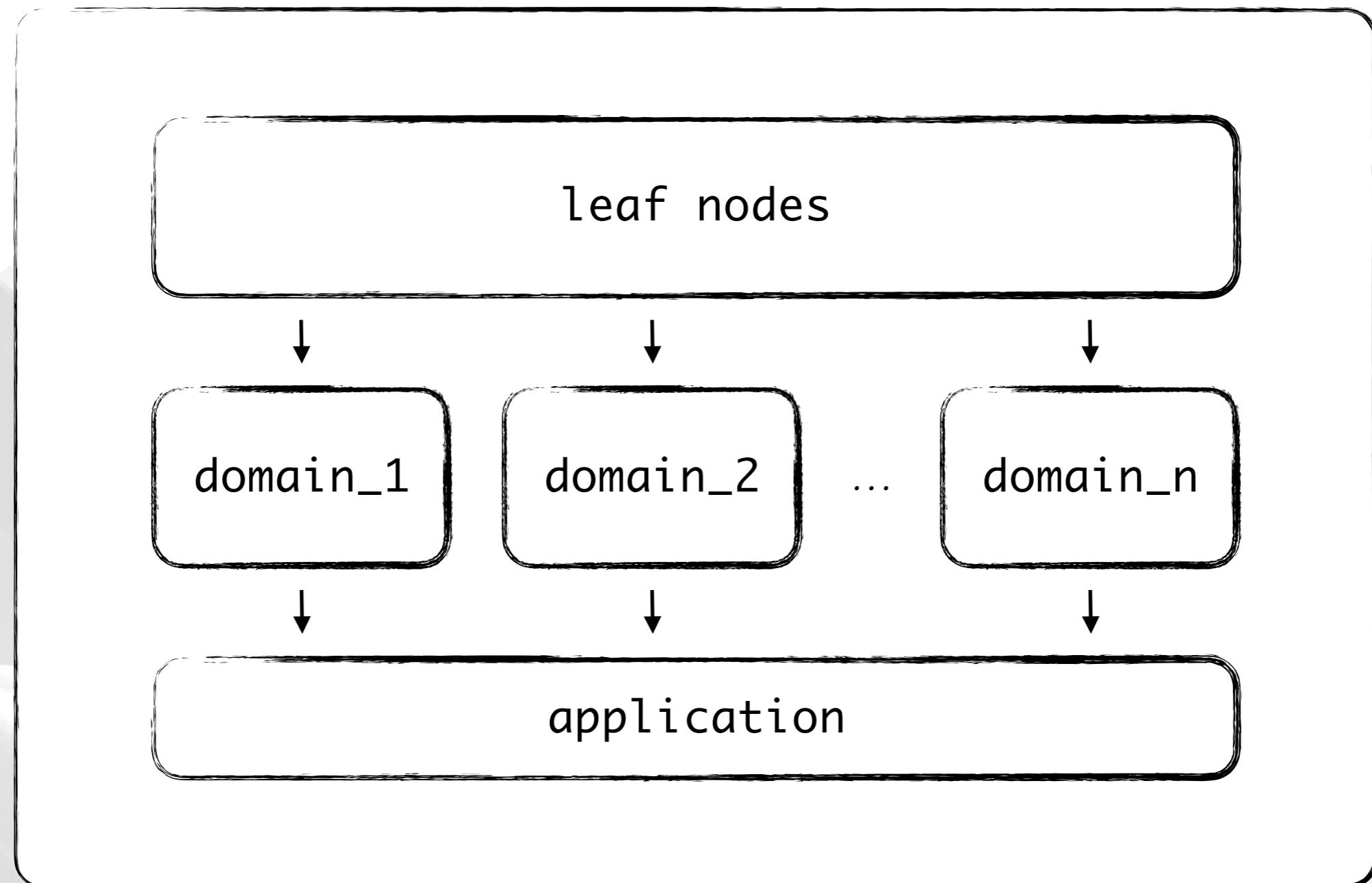
## checklist (ranking)



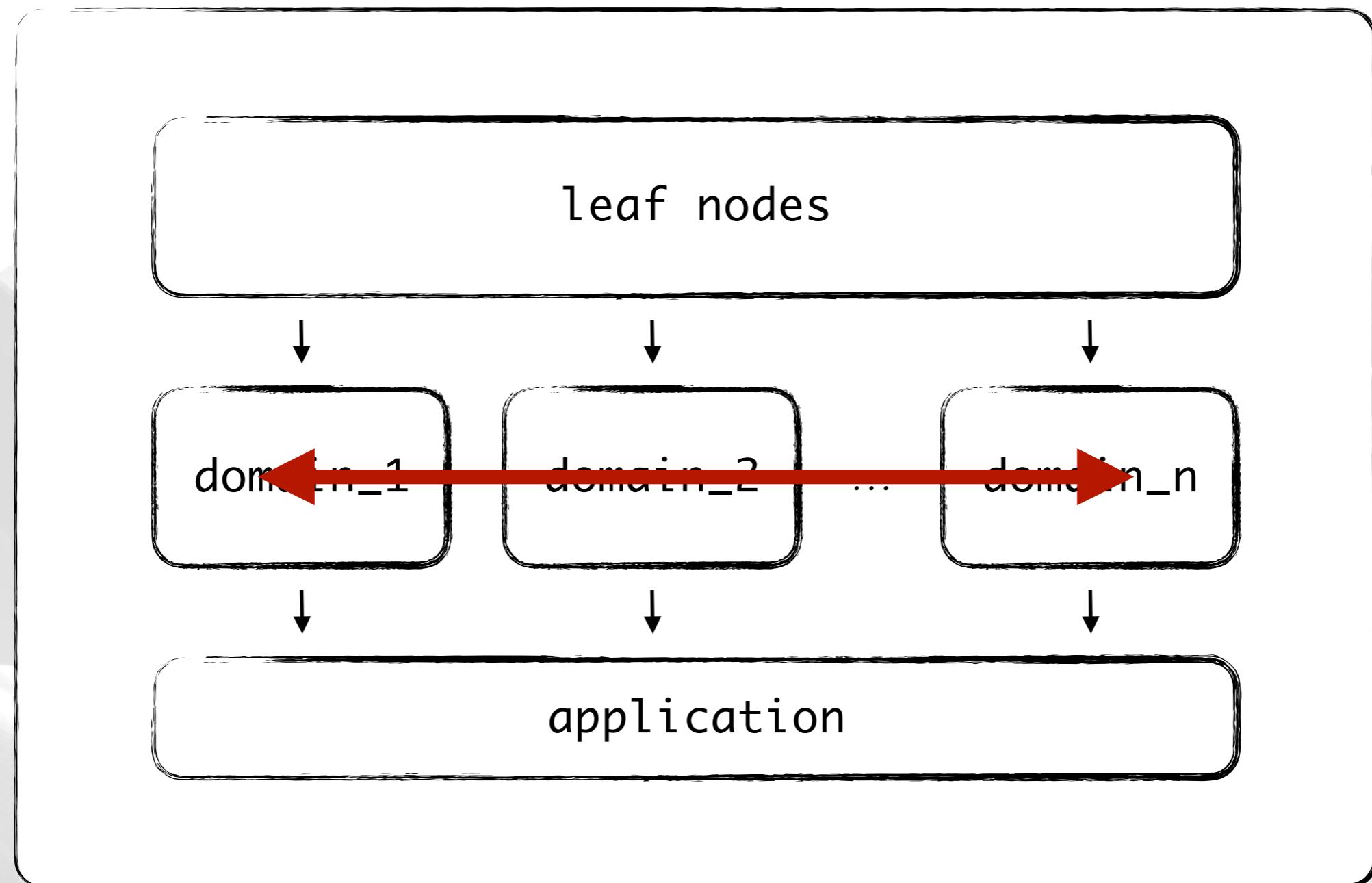
## checklist (ranking)



my-old-repo



my-old-repo



# checklist (dependencies)

## `*.hbs`

- *Block params*
- *Components*
- *Helpers*
- *Modifiers*

## `*.{js,ts}`

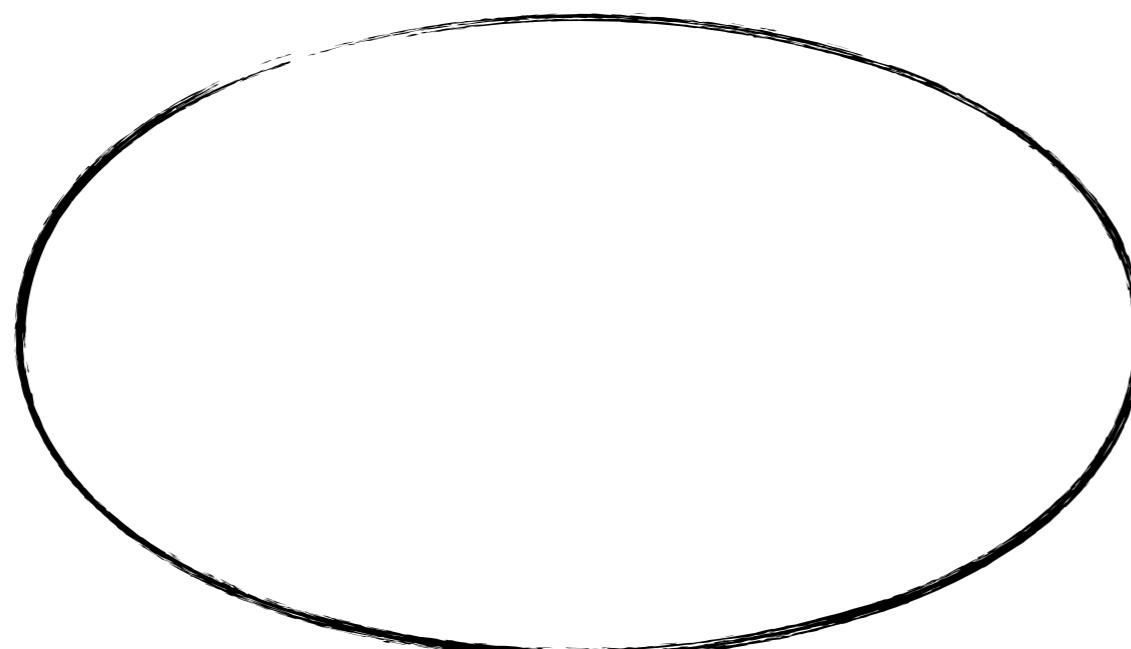
- *Modules (export, import, require)*
- *Services*

## `package.json`

- *devDependencies*
- *dependencies*
- *peerDependencies*

## checklist (dependencies)

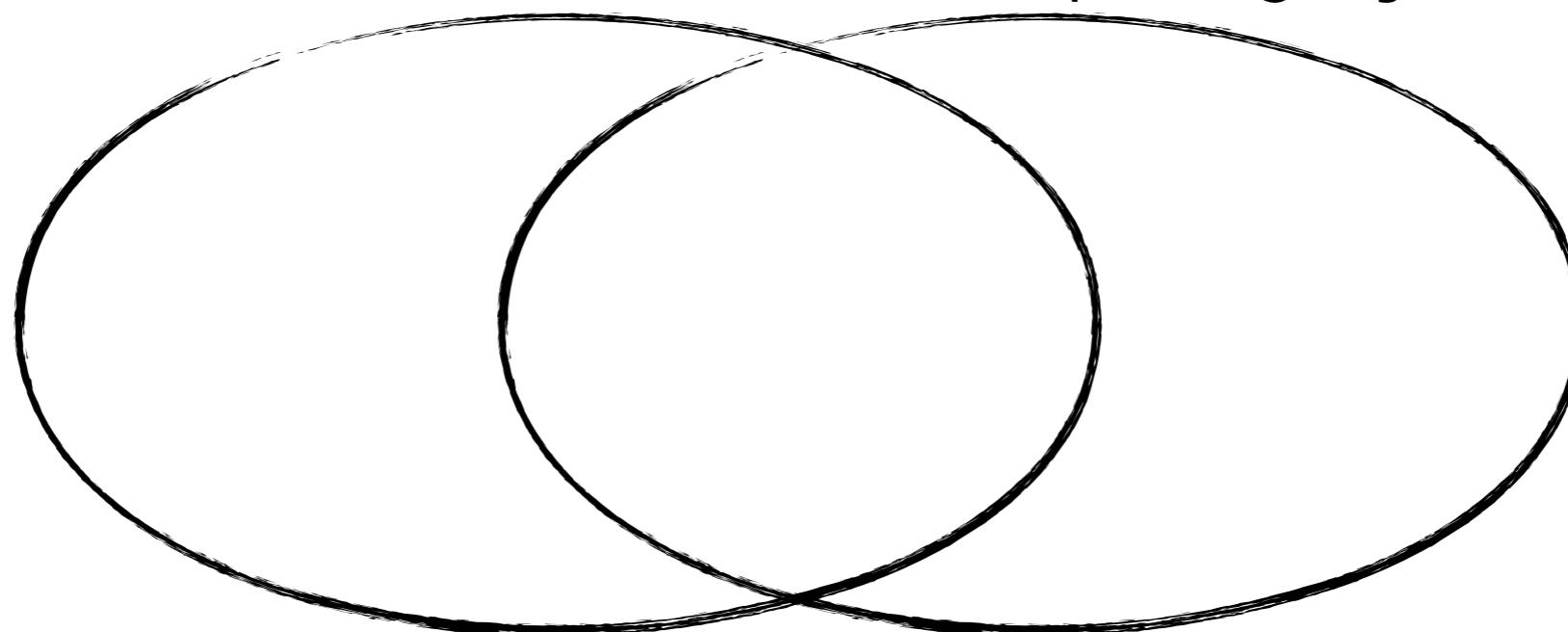
source



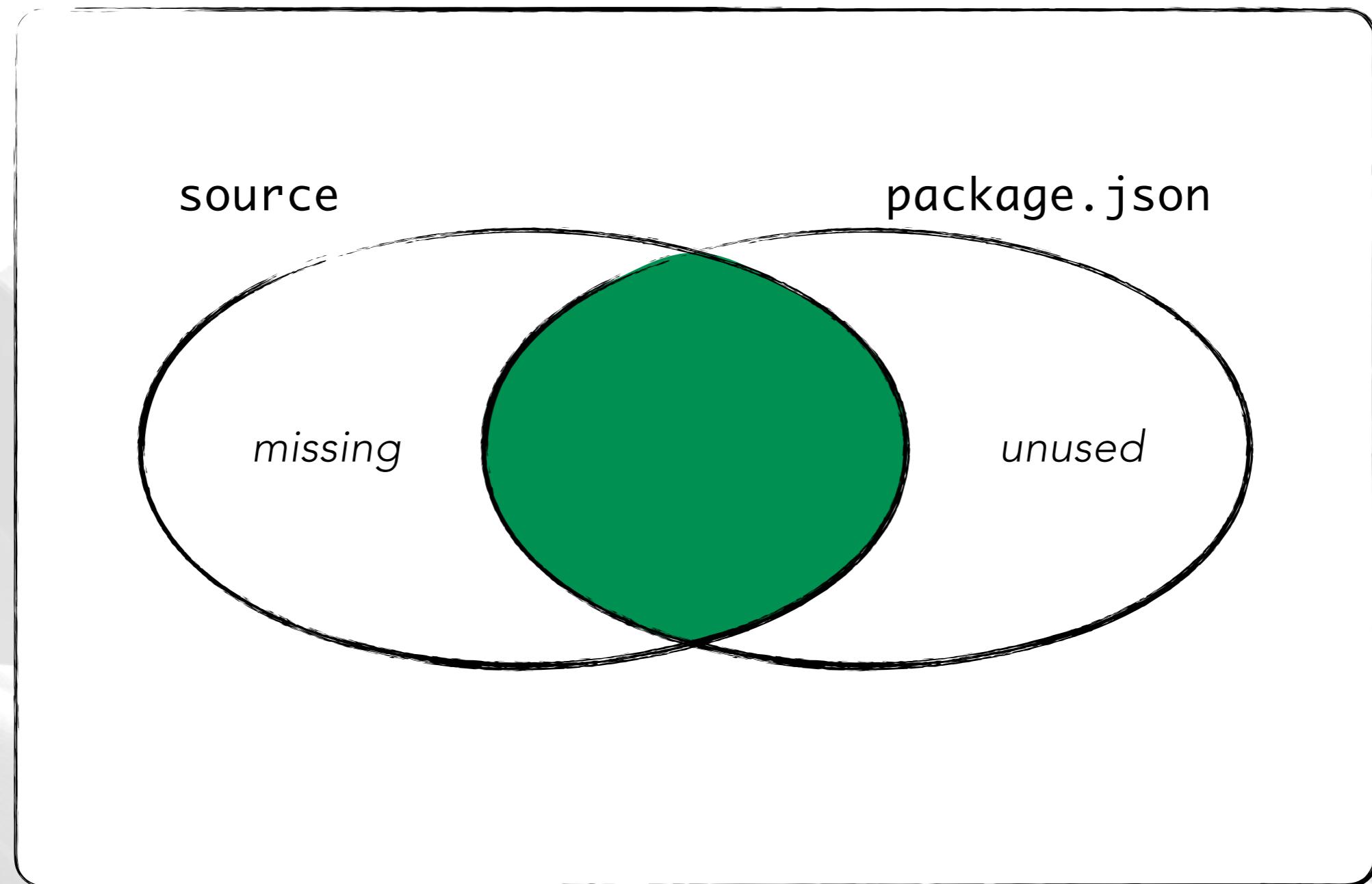
## checklist (dependencies)

source

package.json



## checklist (dependencies)



# analyze project

Get started:

```
npx analyze-ember-project-dependencies
```

[github.com/ijlee2/embroider-toolbox](https://github.com/ijlee2/embroider-toolbox)

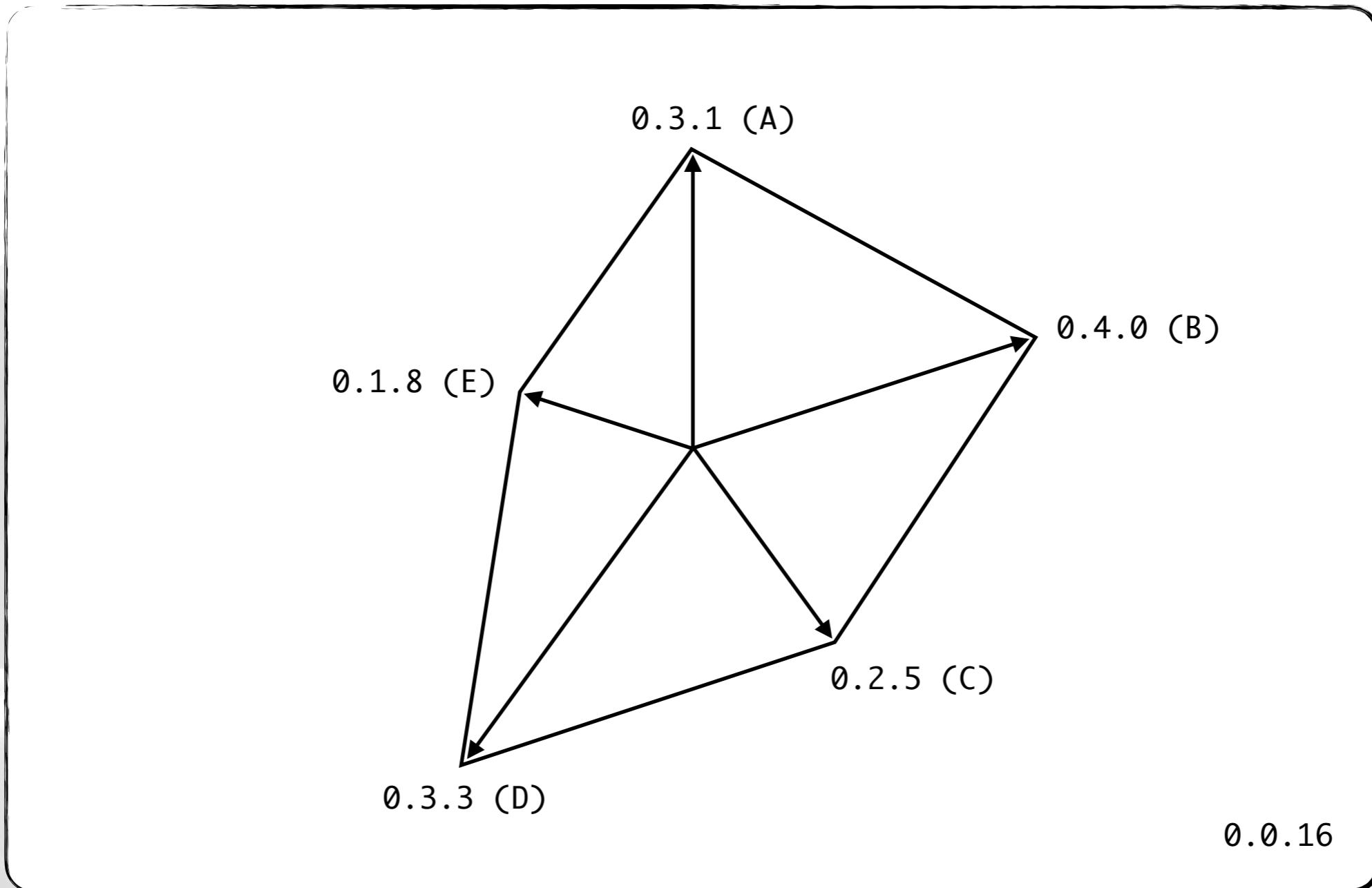
# releases

- *Follow semantic versioning in each package*
- *Add changeset or label to pull request*
- *Publish often, consume fast*

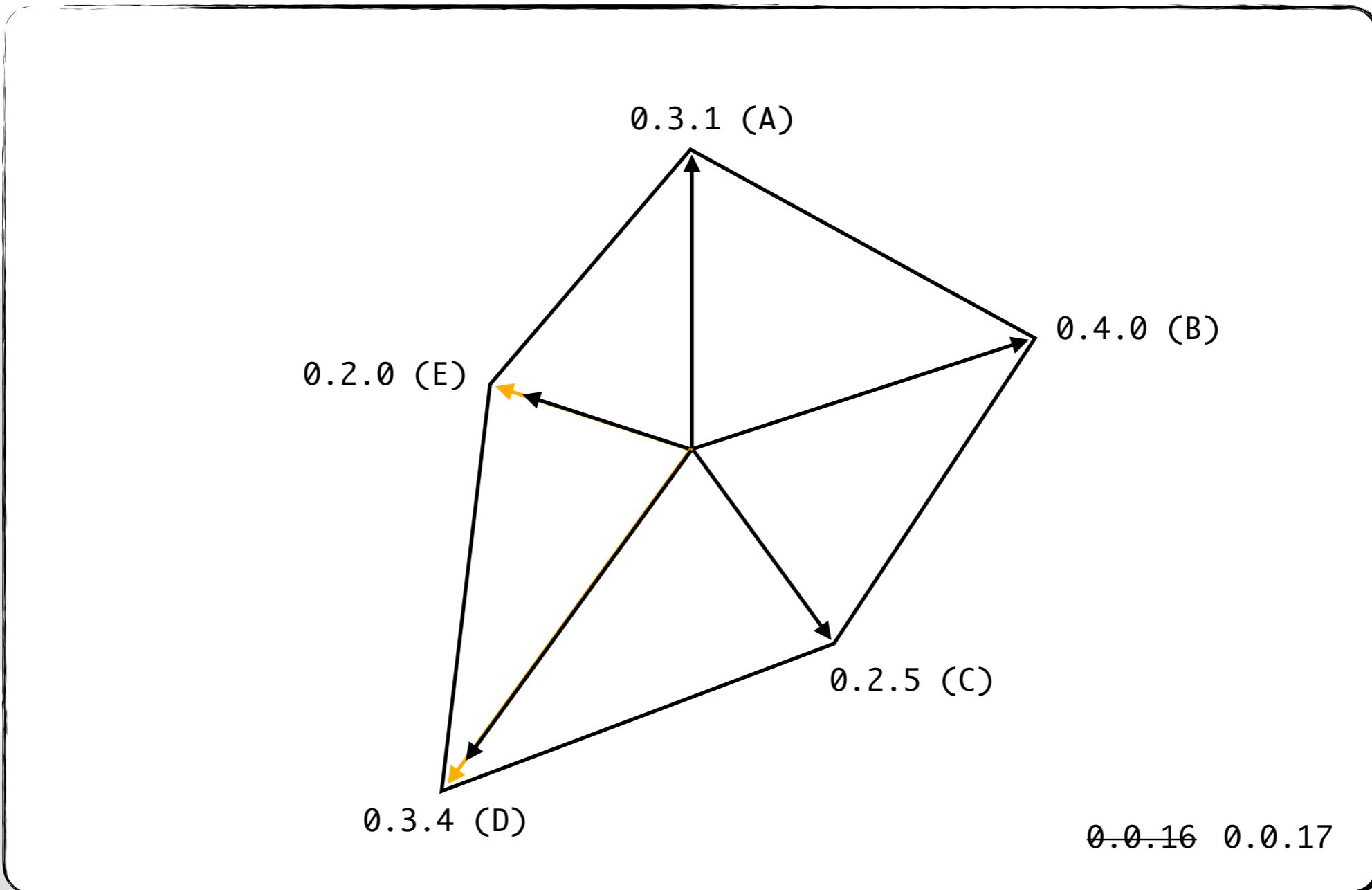
# git tags

-  Never
-  Fixed version (i.e. bump all packages)
-  Per package per version
-  Workspace root's version (increment by one)
-  Workspace root's version (highest version)

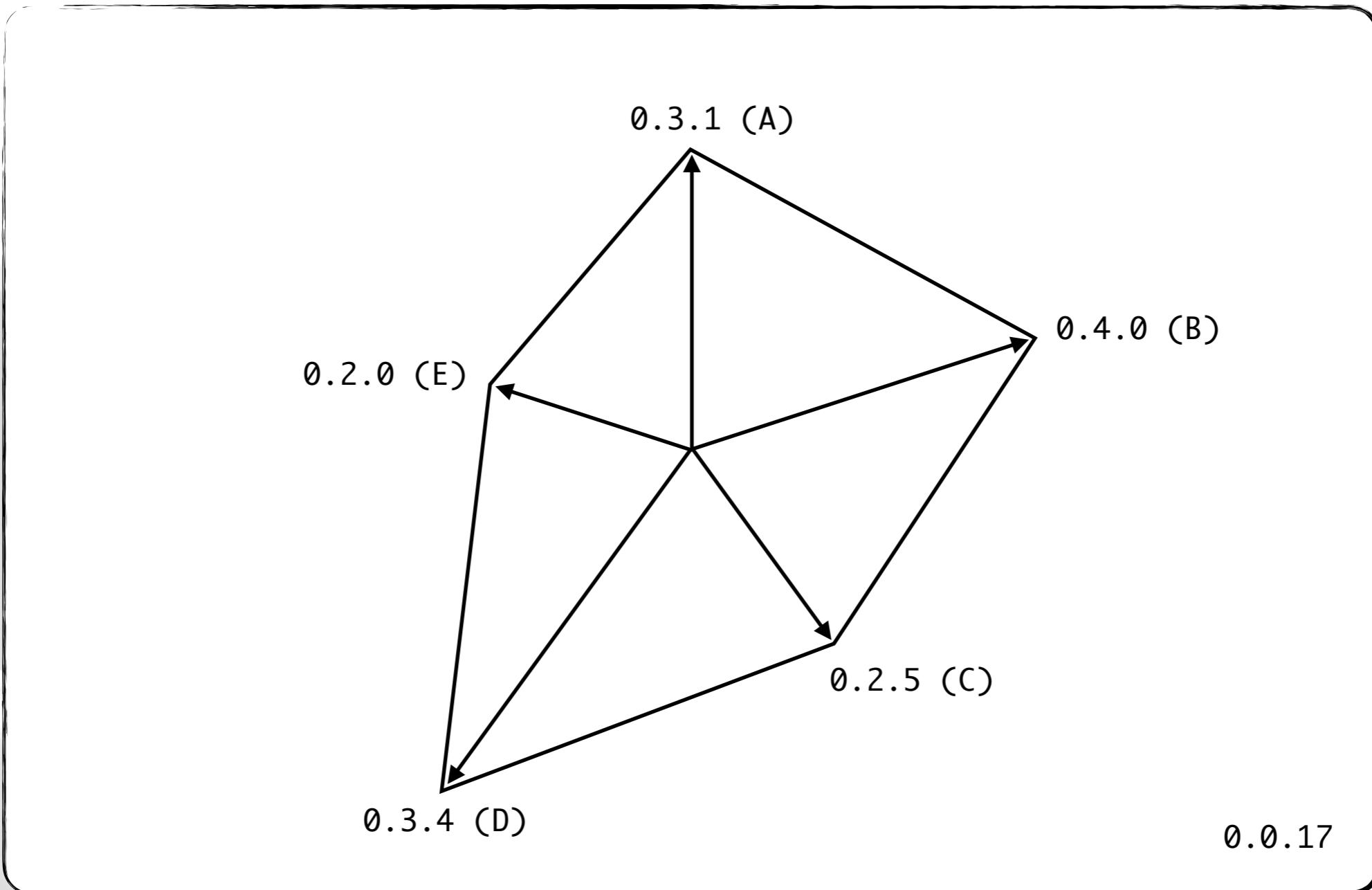
## my-repo (increment-by-one)



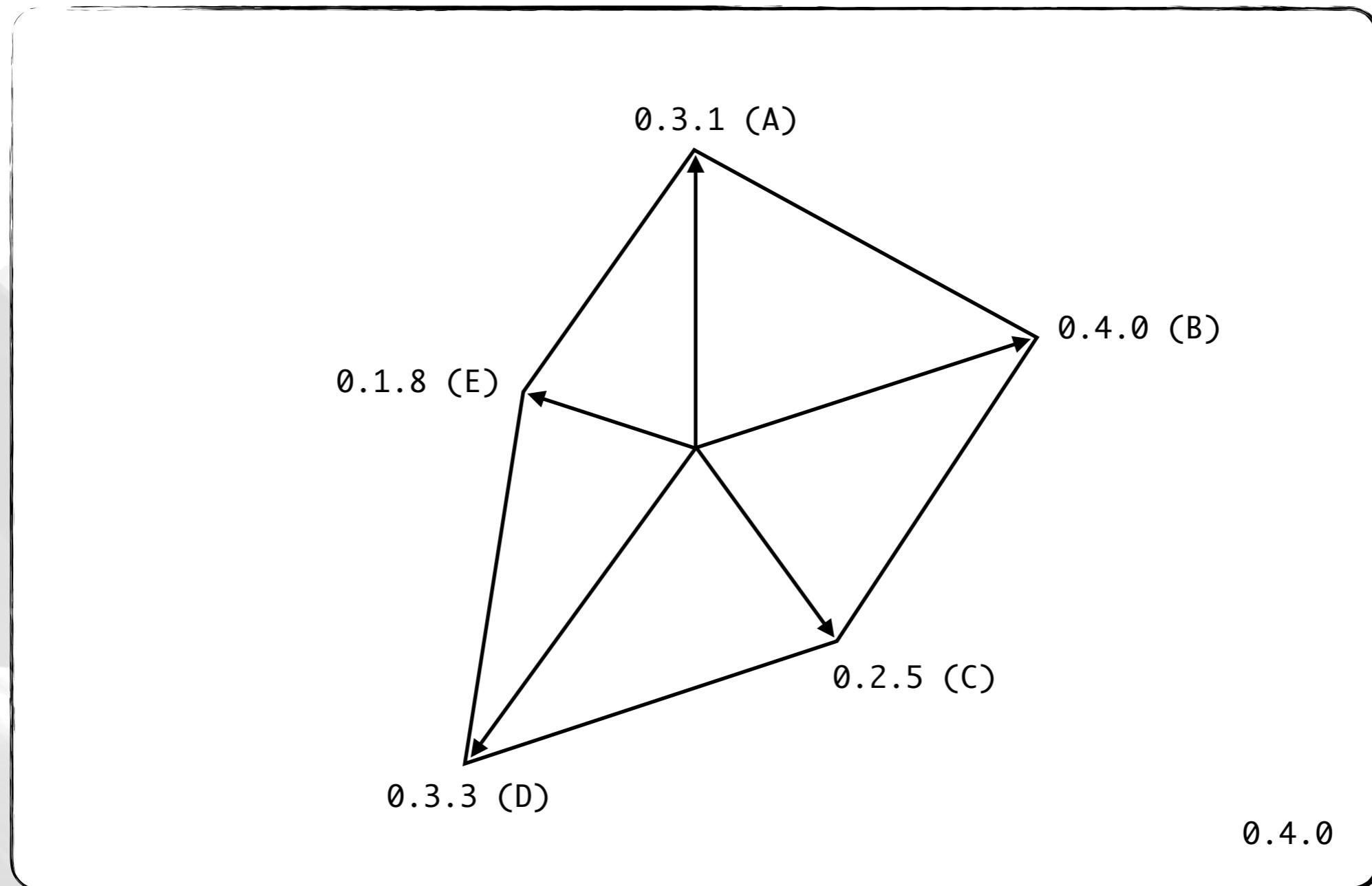
## my-repo (increment-by-one)



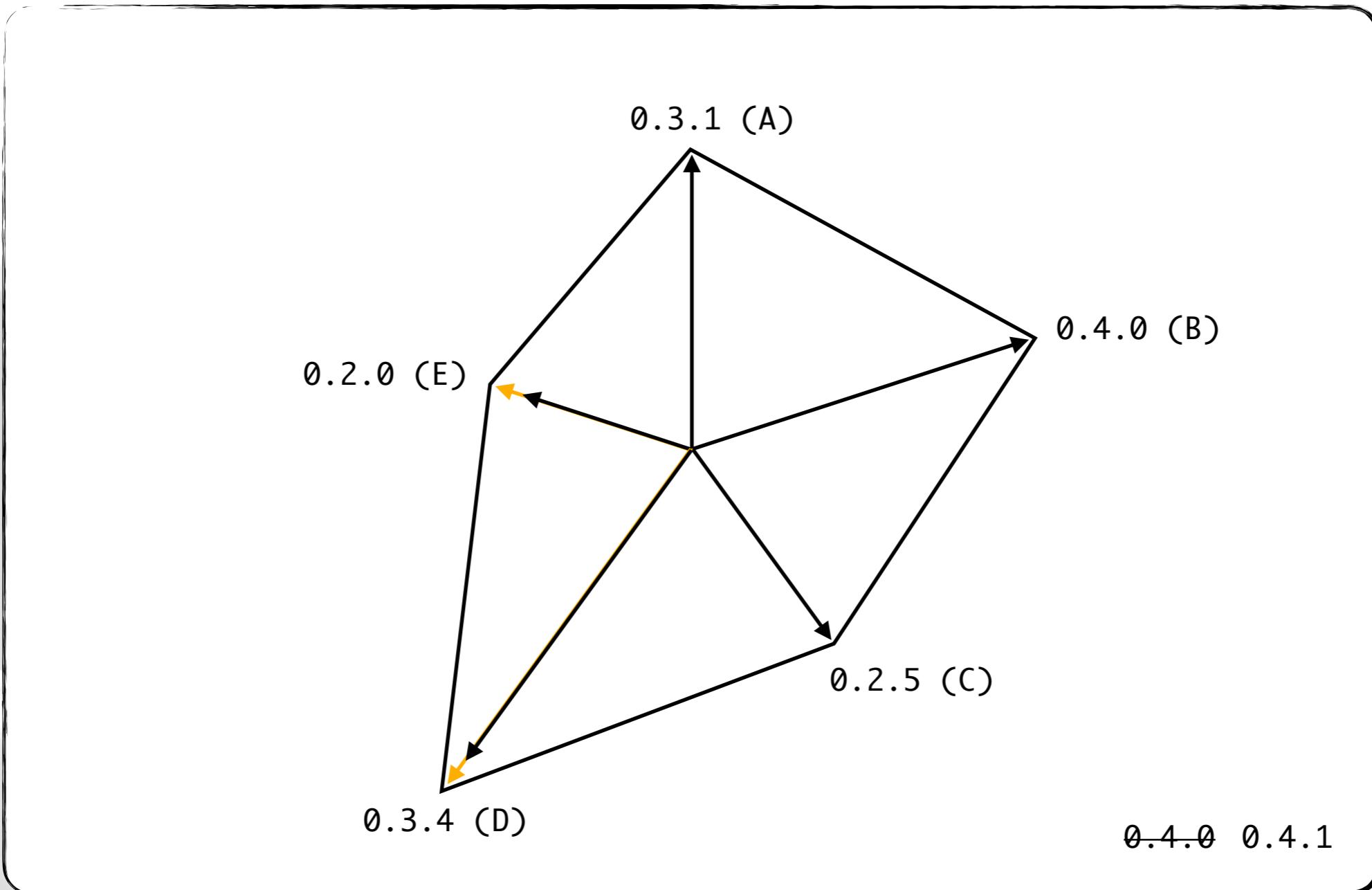
## my-repo (increment-by-one)



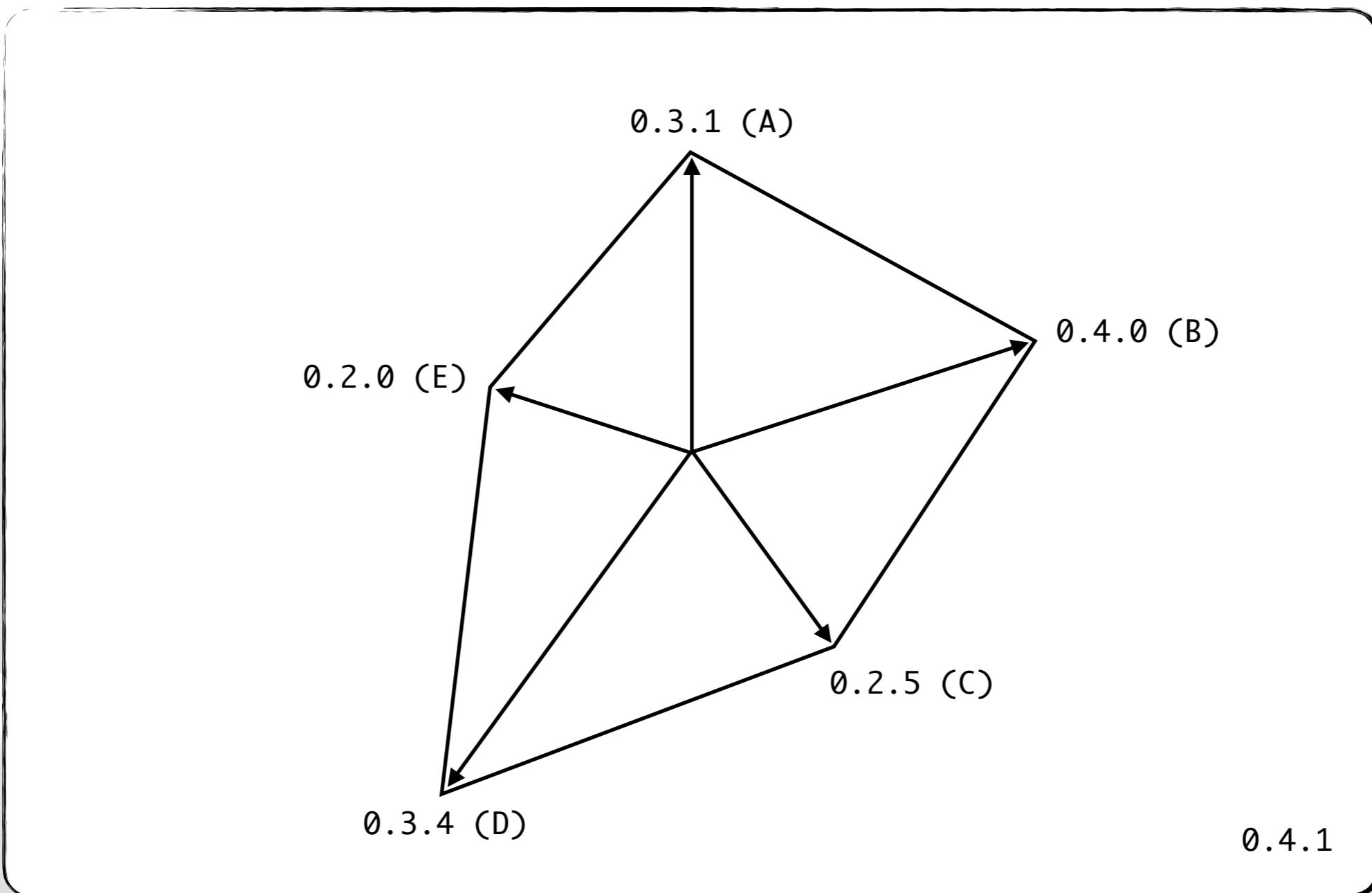
## my-repo (highest-version, case 1)



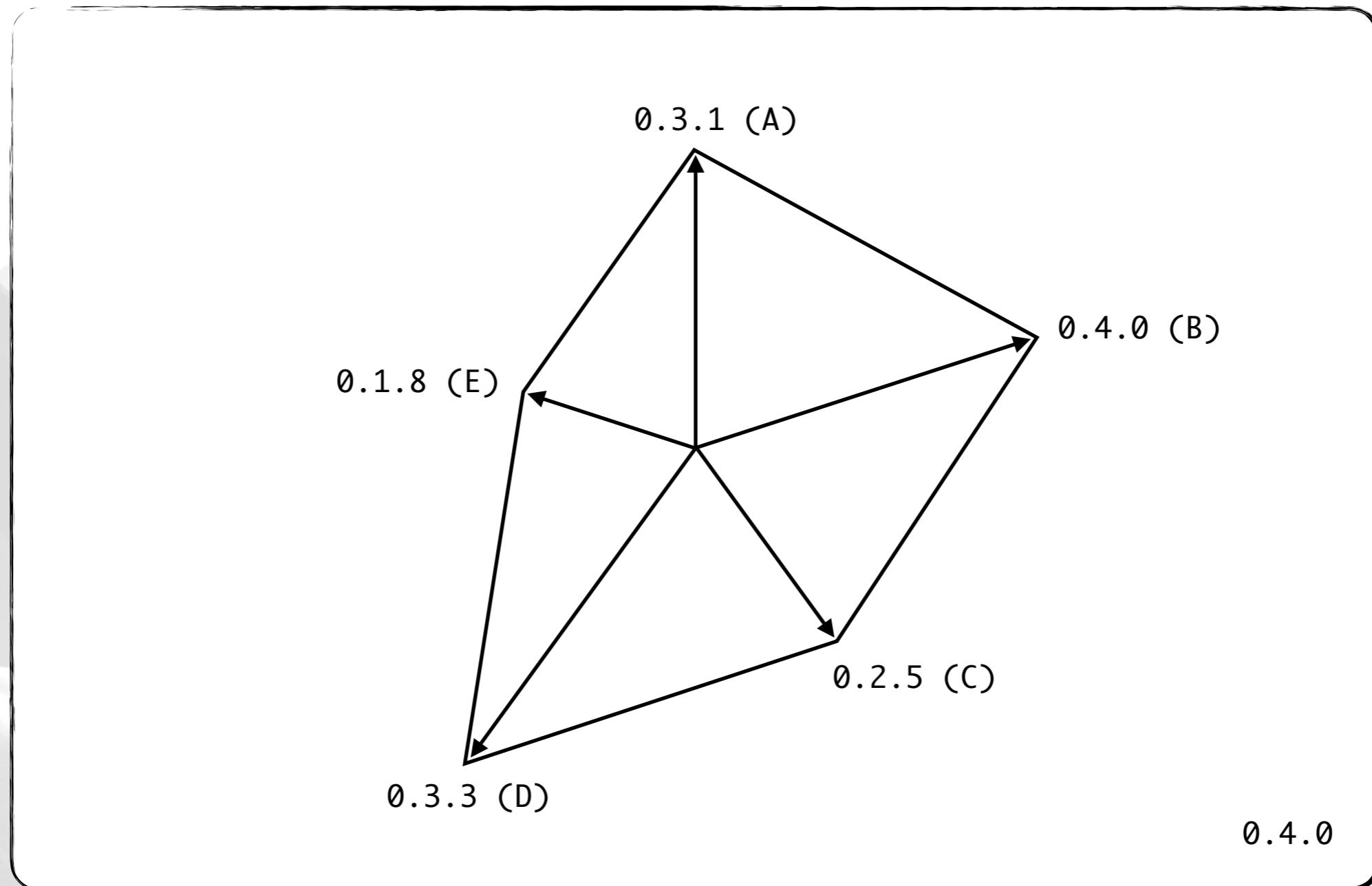
## my-repo (highest-version, case 1)



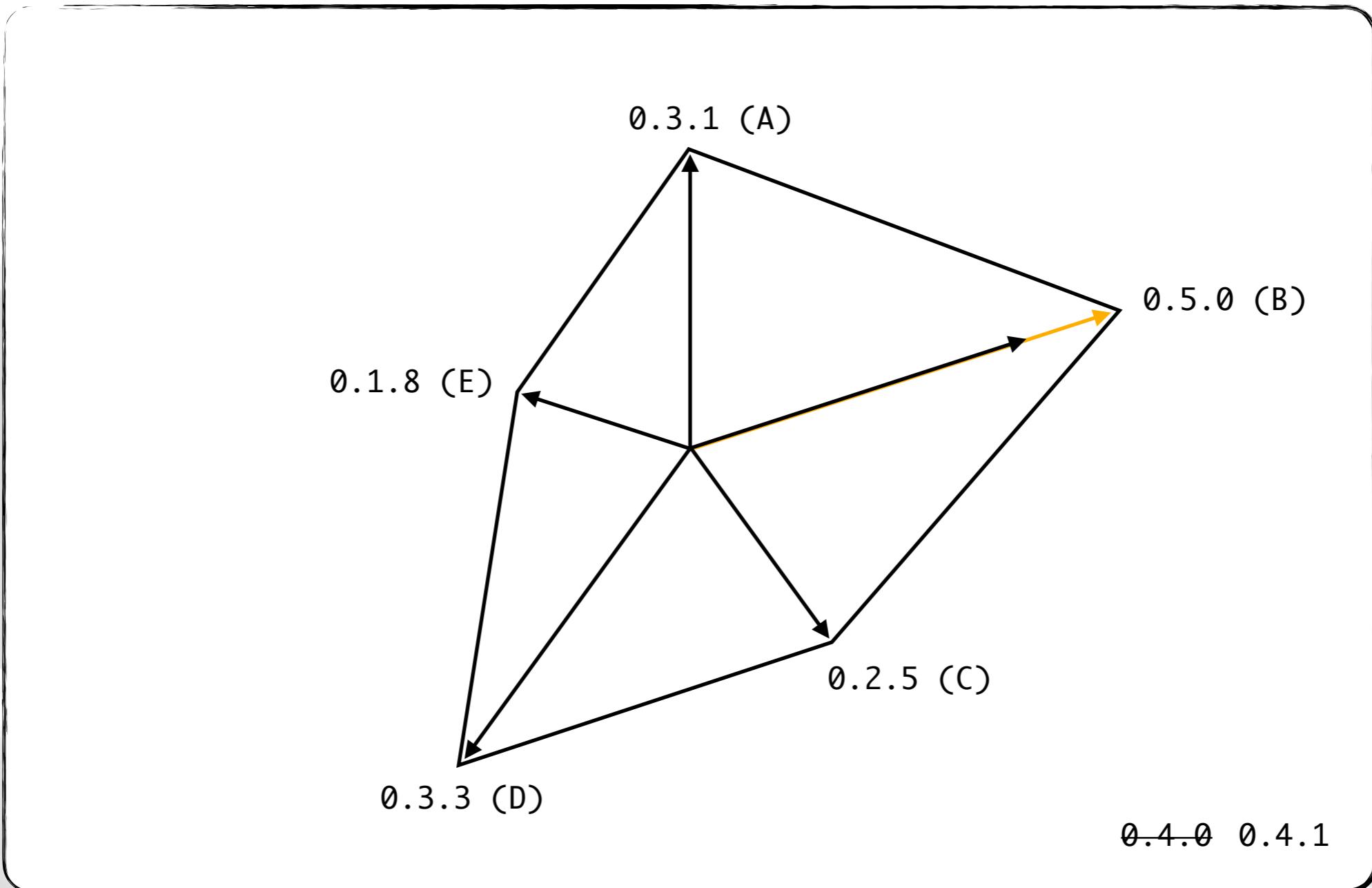
## my-repo (highest-version, case 1)



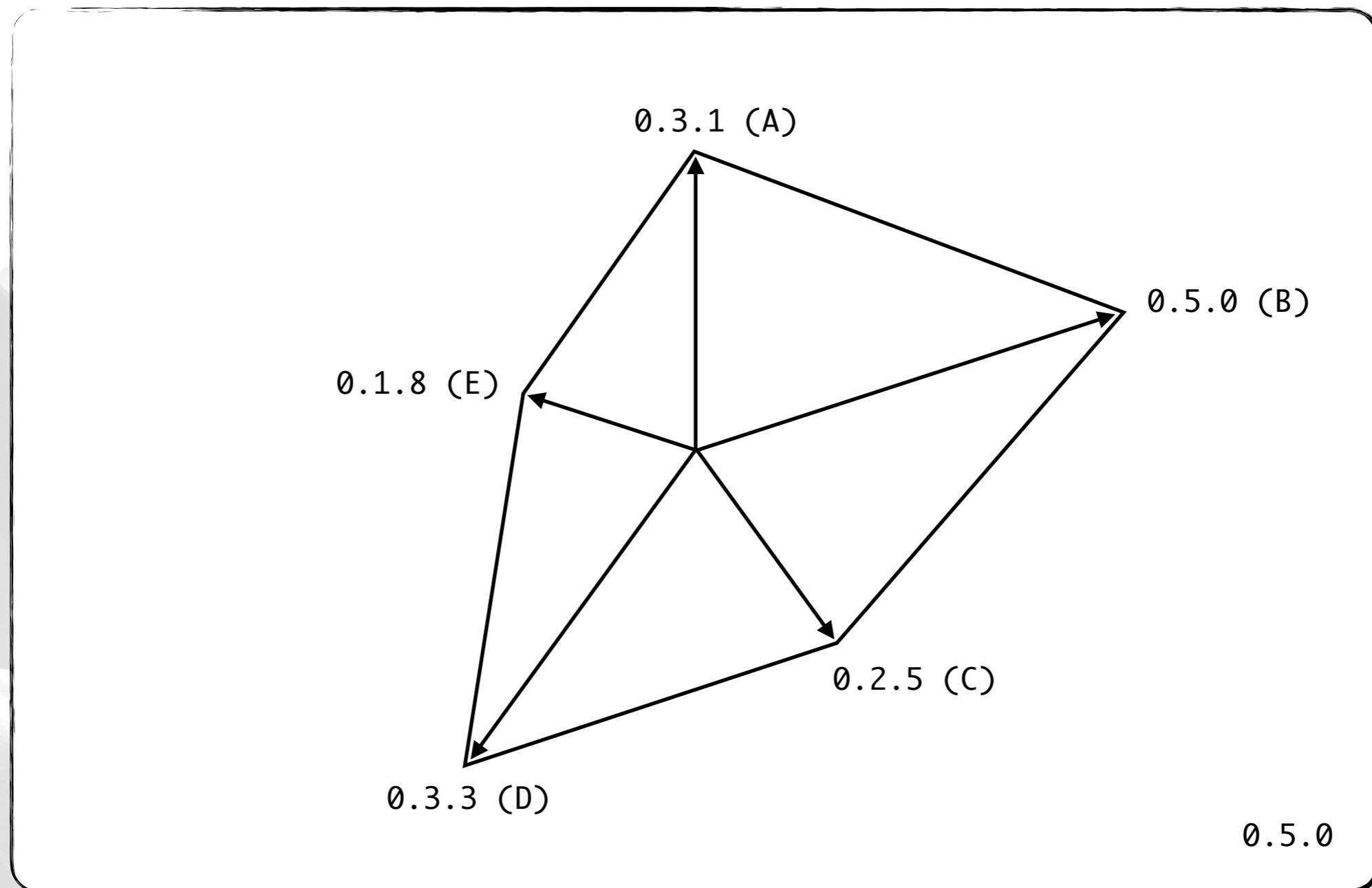
## my-repo (highest-version, case 2)



## my-repo (highest-version, case 2)



## my-repo (highest-version, case 2)



# update version

Get started:

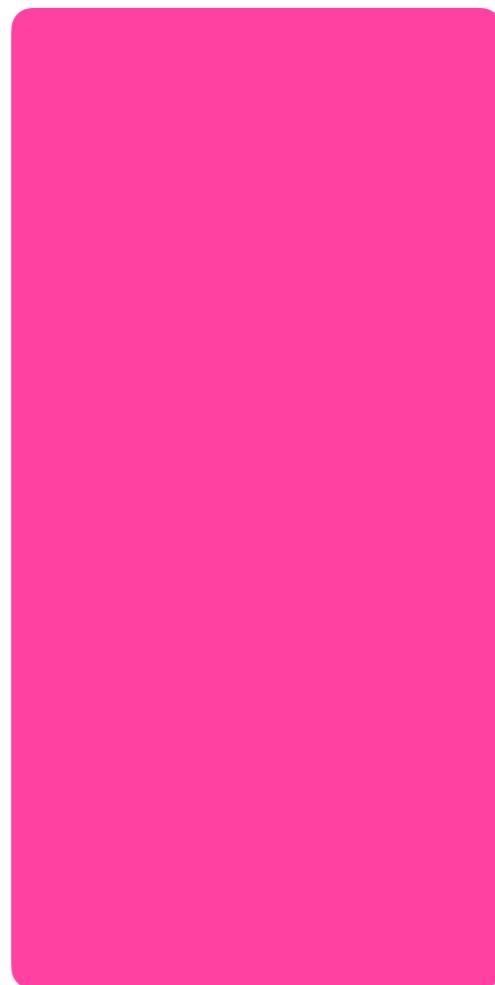
```
pnpm add update-workspace-root-version
```

[github.com/ijlee2/update-workspace-root-version](https://github.com/ijlee2/update-workspace-root-version)

my-repo

---

docs-app



packages



test-app



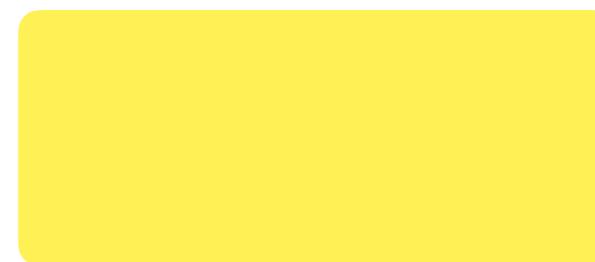
my-repo

---

docs-app



packages



test-app



my-repo

---

docs-app



packages



⋮



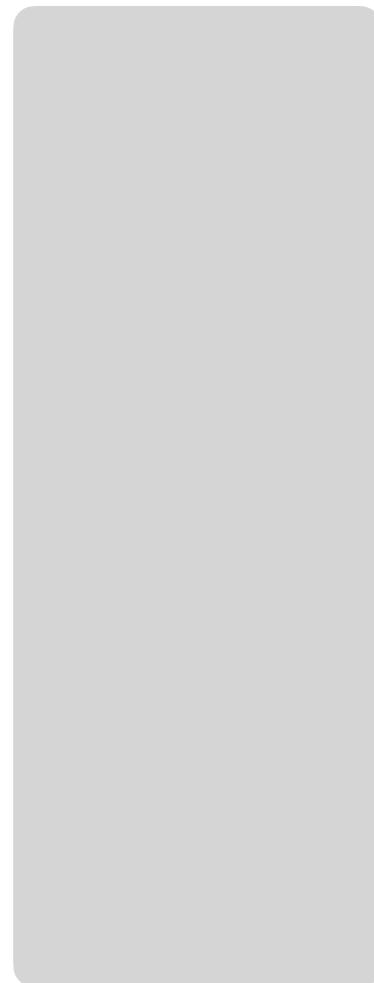
test-app



# my-repo

---

blueprint



docs-app



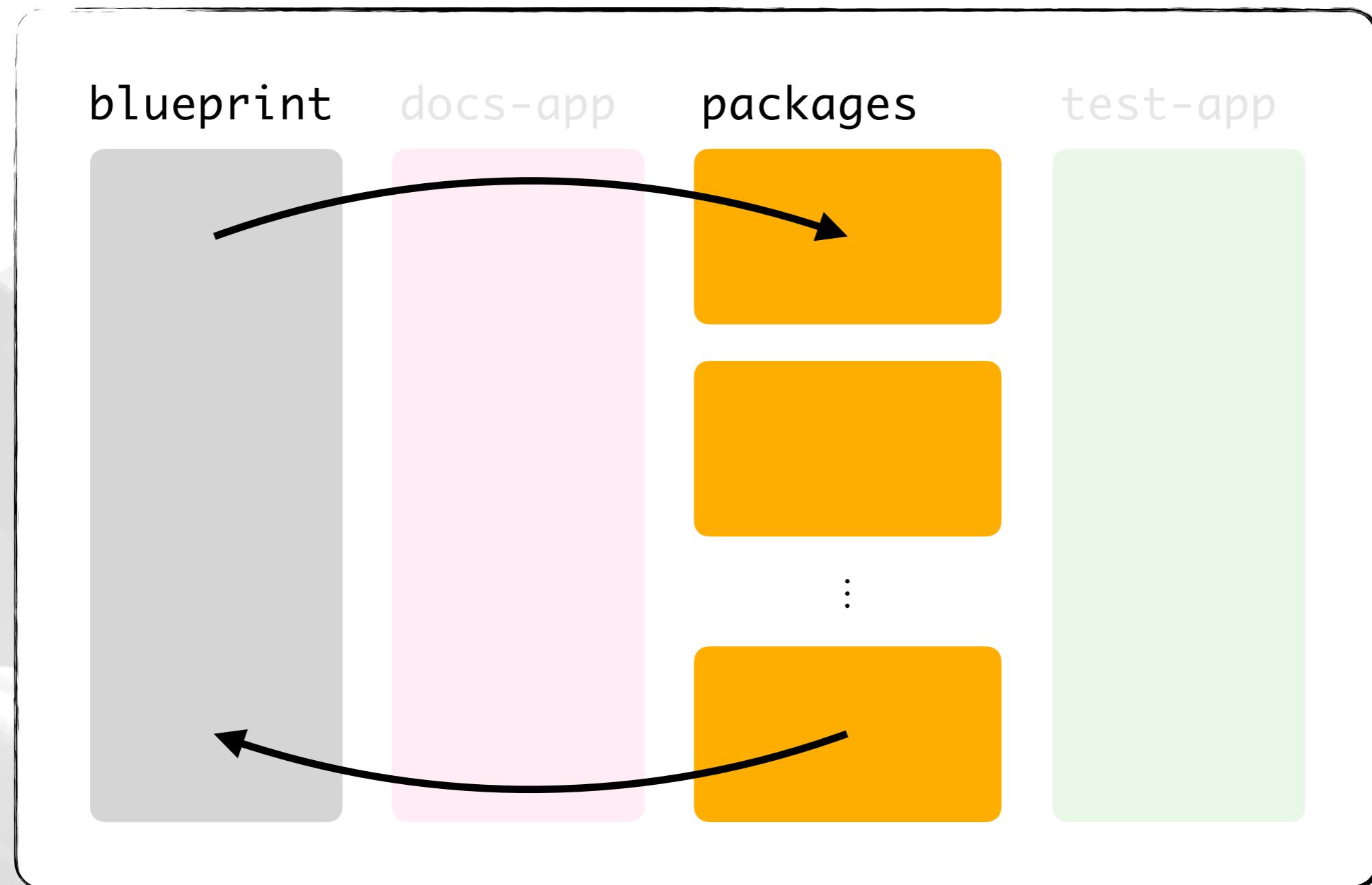
packages



test-app



my-repo



**2b**

**fruits of  
labor**

# in 2 years

	July 1, 2022	July 1, 2023	July 1, 2024
Packages	187		
Packages (new repo)	-		
Ember	3.28.6		
CI (min)	33		

# in 2 years

	July 1, 2022	July 1, 2023	July 1, 2024
Packages	187	150	
Packages (new repo)	-	6	
Ember	3.28.6	3.28.11	
CI (min)	33	21	

# in 2 years

	July 1, 2022	July 1, 2023	July 1, 2024
Packages	187	150	109
Packages (new repo)	-	6	27
Ember	3.28.6	3.28.11	5.8.0
CI (min)	33	21	14

# new repo

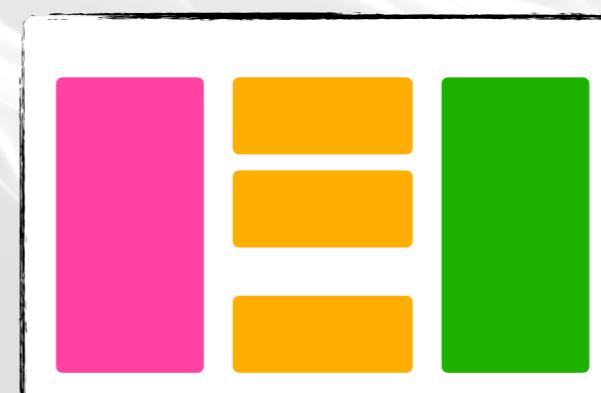
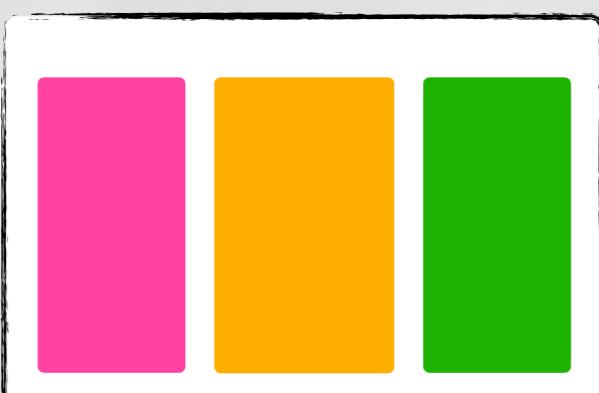
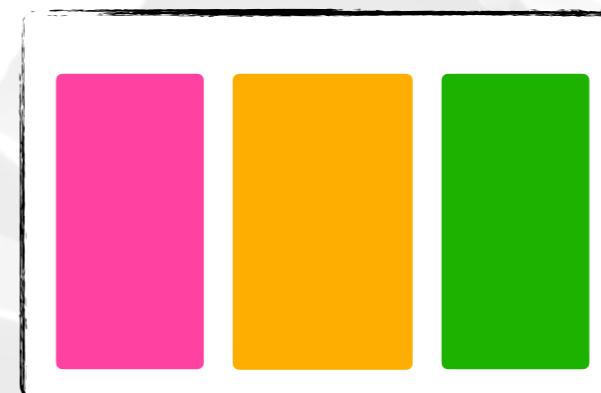
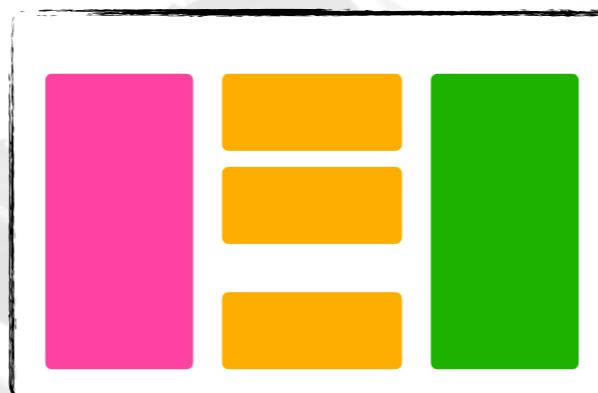
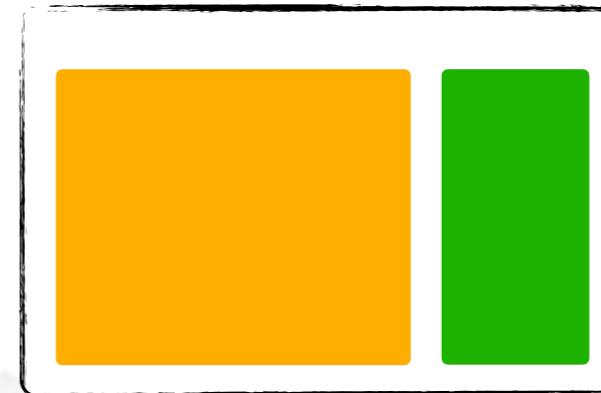
- *Forward-compatible code*
- *Feedback from designers and managers*
- *Few Percy snapshots that check "all" cases*
- *"Naive" CI in 4.5 min, publish in 3 min*
- *Discover and share conventions for Ember*



3

moving  
many repos

## all-our-repos



# create repo

Get started:

```
npx create-v2-addon-repo my-repo
```

[github.com/ijlee2/embroider-toolbox](https://github.com/ijlee2/embroider-toolbox)

# create addon

Get started:

```
pnpm addon new ui-form
```

[github.com/ijlee2/embroider-toolbox](https://github.com/ijlee2/embroider-toolbox)

# create files

Get started:

```
pnpm addon g component ui/form/input
```

[github.com/ijlee2/embroider-toolbox](https://github.com/ijlee2/embroider-toolbox)

# remove files

Get started:

```
pnpm addon d component ui/form/input
```

[github.com/ijlee2/embroider-toolbox](https://github.com/ijlee2/embroider-toolbox)

# update blueprints

Get started:

```
pnpm update-blueprints --to 0.5.0
```

[github.com/ijlee2/embroider-toolbox](https://github.com/ijlee2/embroider-toolbox)

# more problems

- Auto-sync reference repo for blueprints<sup>1</sup>
- Limit complexity<sup>2</sup>
- Document v2 addons in an accessible way
- Rethink how we do blueprints<sup>3</sup>

1. With `git sparse-checkout`.

2. `bun/npm/pnpm/yarn` × JS/TS × Vite/Webpack = 16 variations, 256 times the current cost!

3. With packages from `@codemod-utils`.



**moving  
mountains**

[github.com/ijlee2](https://github.com/ijlee2)