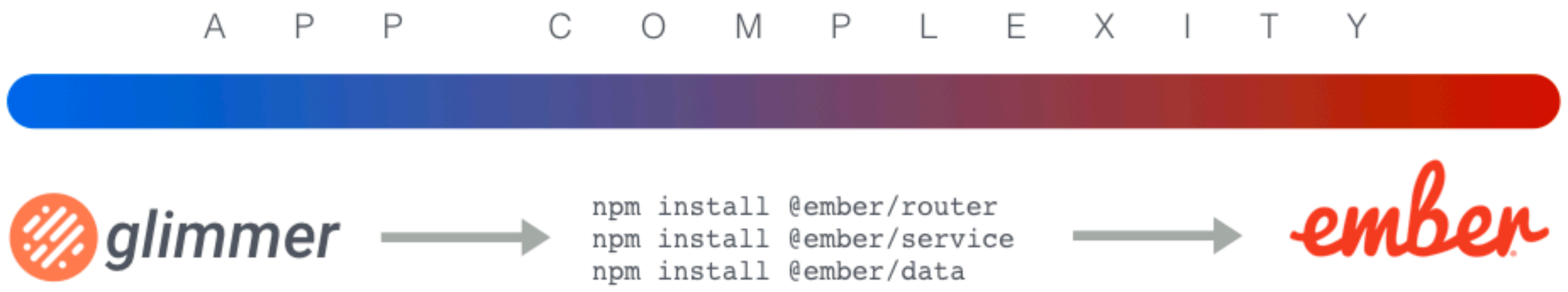
Design Patterns

in Ember

@chadian

A software design pattern is a general, reusable solution to a commonly occurring problem within a given context in software design.

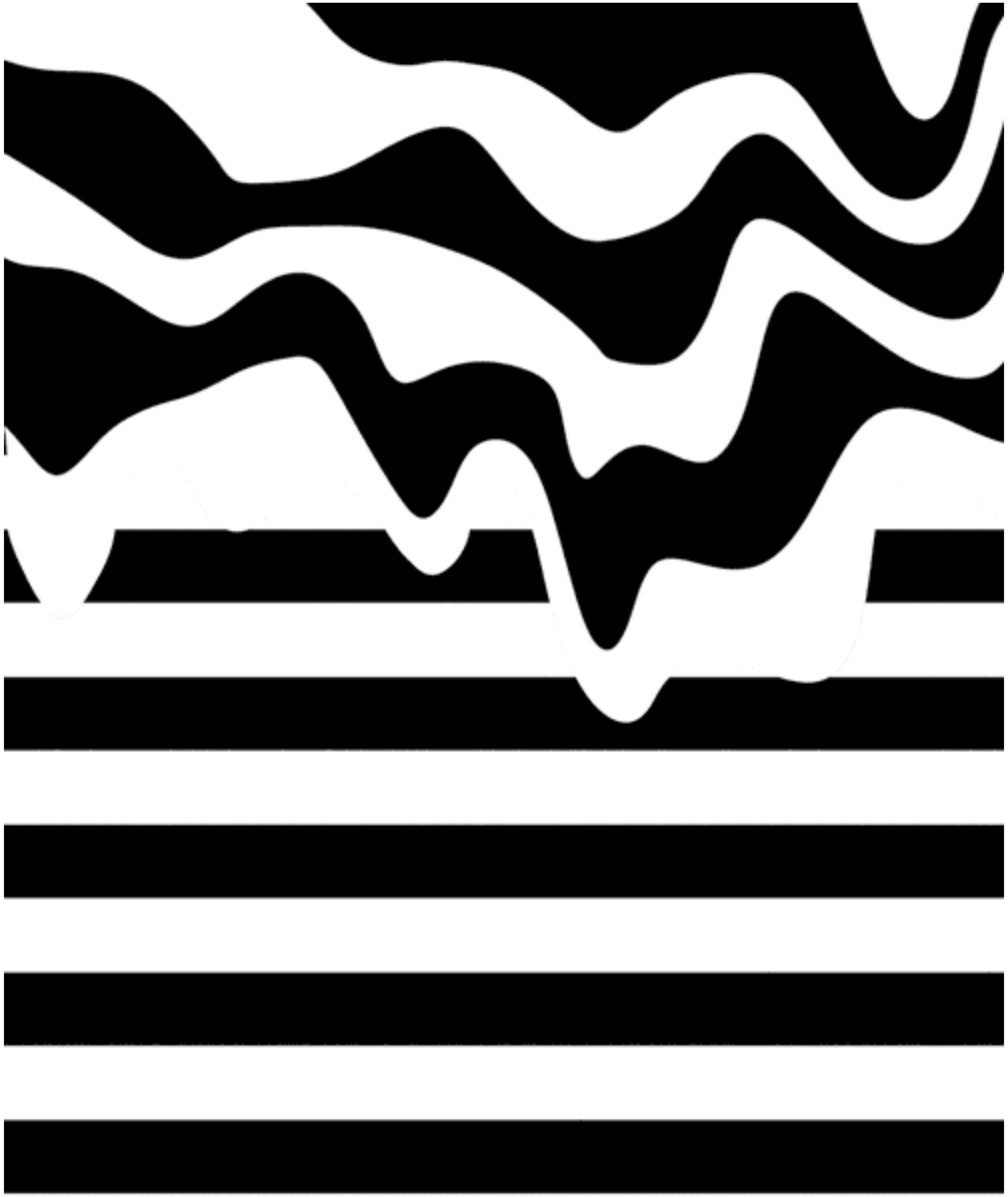
Don't waste time making **trivial choices**. Ember.js incorporates **common idioms** so you can focus on what makes your app special, **not reinventing the wheel**.



```
npm install @ember/router
npm install @ember/service
npm install @ember/data
```

Framework patterns

- “Start with Routing” approach
- Data down / Actions up
- Folder structure
- addons / engines
- ember-data
- `ember-cli`



Community Patterns

- RFCs
- Ember (**Community**) Addons

https://github.com/emberjs/rfcs

Ember Concurrency

- Async made easy by the `task` Primitive
- State machine over time, without streams (RxJs)
- Lean on generators, Made for Ember™
  - `async`/`await`-esque by `yielding` promises
- ⚠️ State flags, ie: `isLoading`, lifecycle hooks
- ⚠️ Staggered handling of actions over time
- ⚠️ Cleanup: Cancellation, clearTimeout or Ember.run.cancel

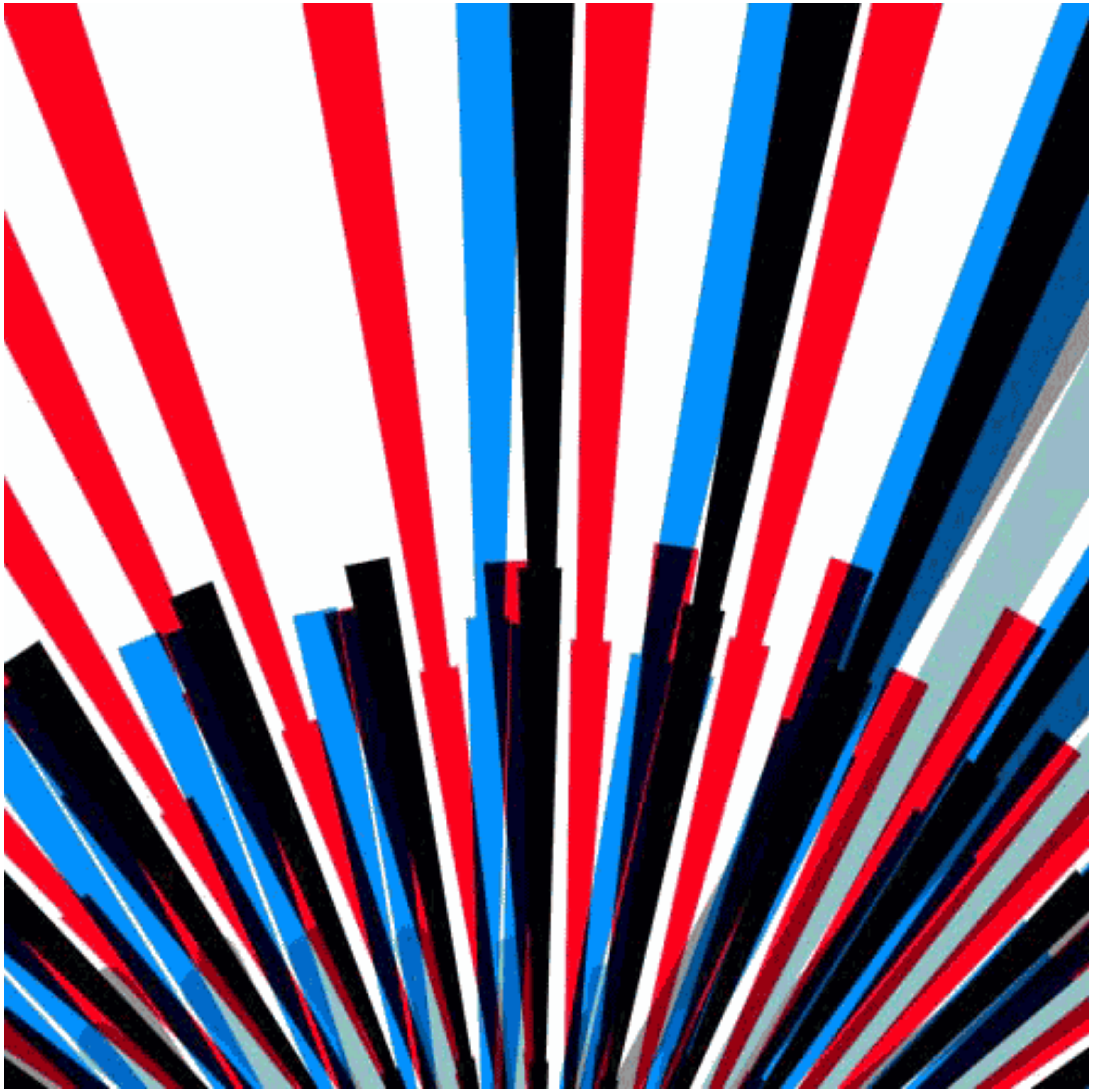
Ember Concurrency Website

Frontside 67 - Ember Concurrency

EmberConf 2017: State, Time, and Concurrency

Ember Concurrency (example)

- App Example
- Concurrency Example

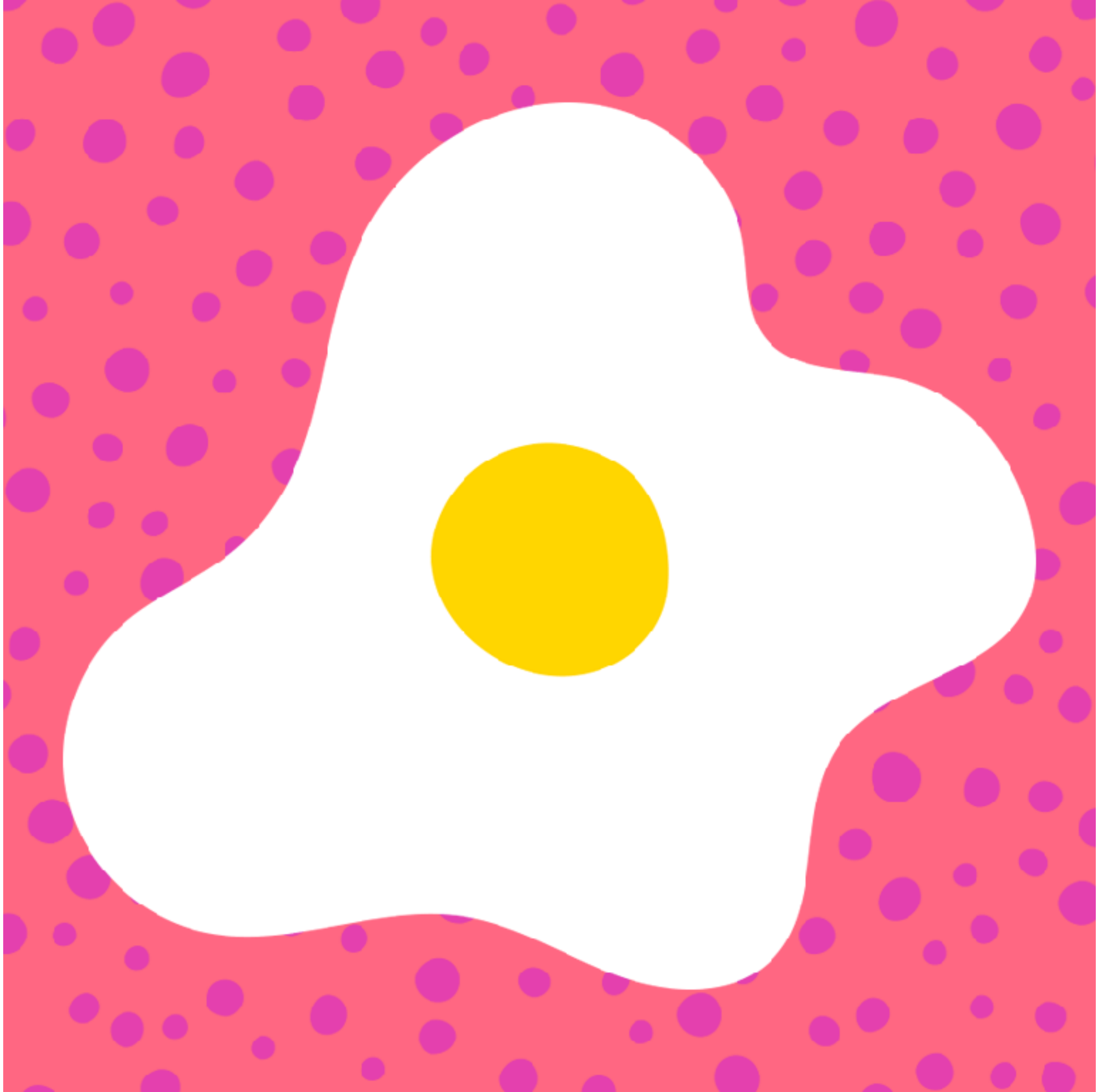


Buffered Proxy

- `ember-buffered-proxy`, `ember-validated-form-buffer` (`ember-data` & `ember-cp-validations`), `ember-changeset`
- Keep “Work in Progress”.
- Preserve the original, handle the changes, and set.
- ⚠️ Form components with passed in model/data/object.

Buffered Proxy (example)

- Example



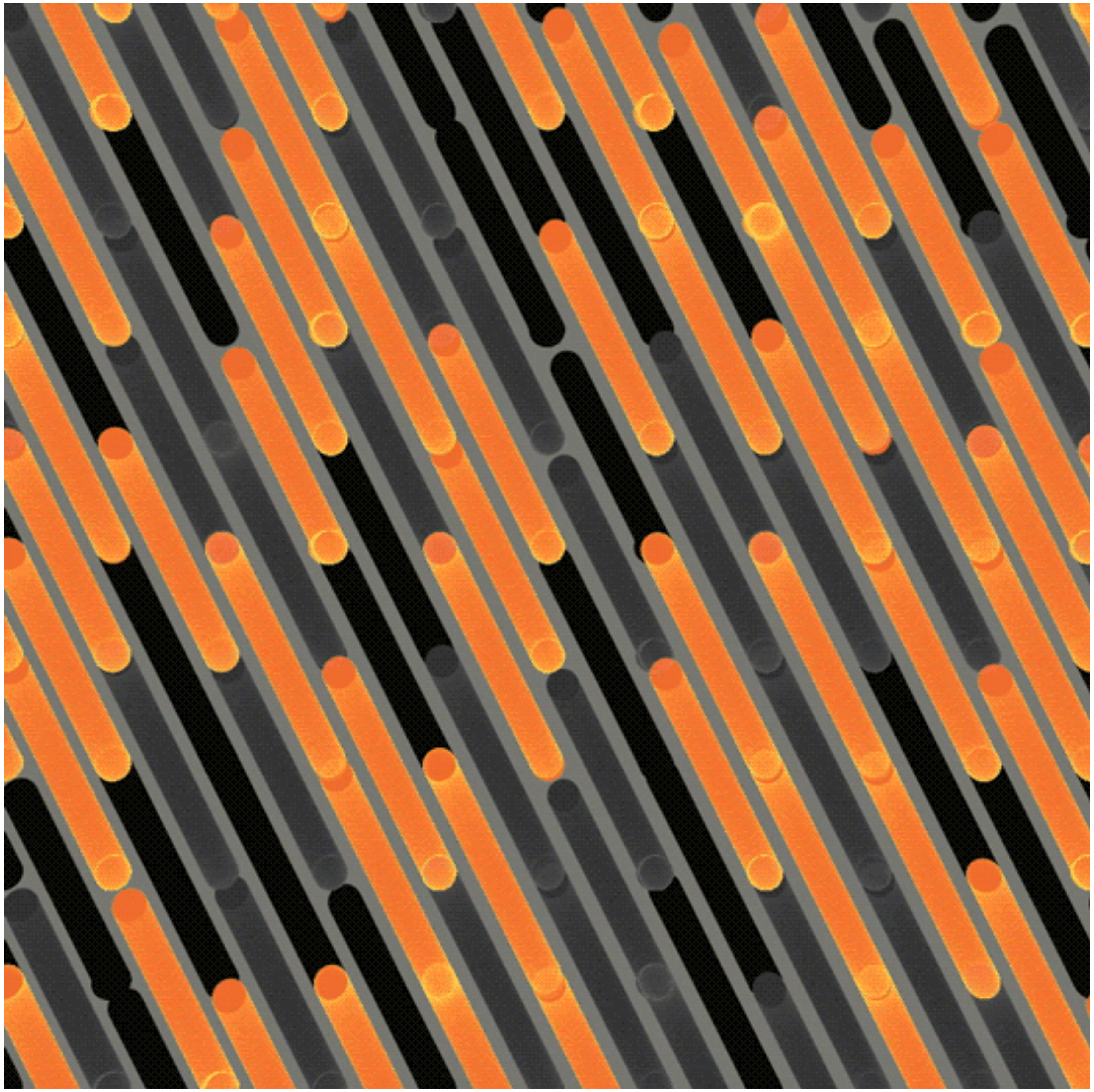
Presentational & Container Components

- “Smart” Container Components
  - How things work — services, data
  - "Stateful”
- “Dumb” Presentational Components
  - How things look — UI
  - "Stateless”
- Focused requirements
  - easier to test and re-use

Dan Abramov - Presentational and Container Components

Presentational & Container Components (example)

- [Example]



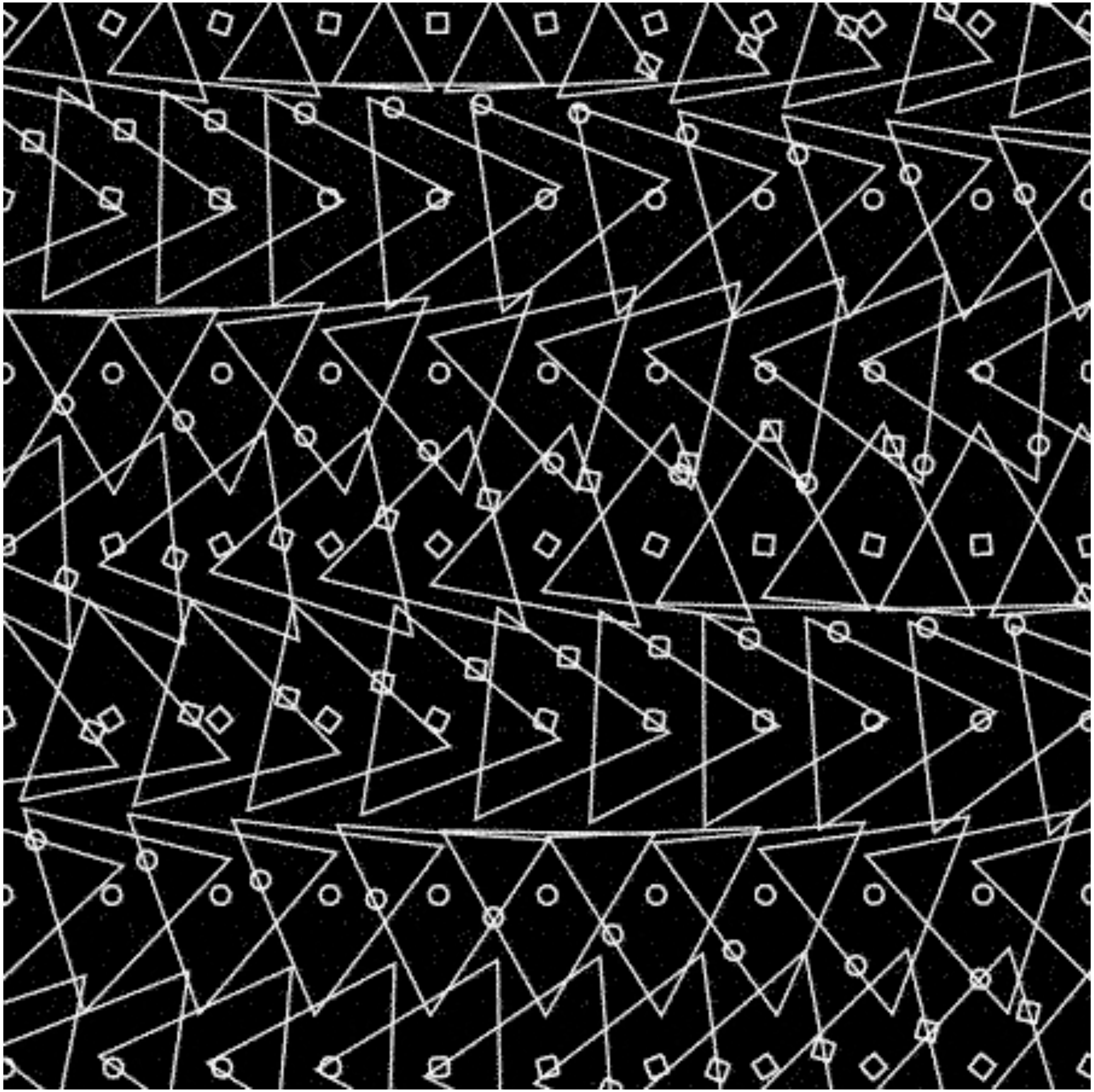
High Order Components

- Using “Contextual Components”
- `component(a, b, c)` **`yields`**➡️ `component(s)`
    - Components can be inputs or other components
- `Parameters` **`yields`**➡️ `“Building Blocks”`
- `Parameters` **`yields`**➡️ `Wrapped Functionality`

[EmberConf 2017: Higher Order Components]()

High Order Components (example)

- Example



Computed Properties & Template Helpers

Simplabs: Computed Properties vs Helpers

Computed Property Macros

- `ember-awesome-macros`
- `ember-macro-helpers`
  - make your own!

https://emberobserver.com/categories/computed-property-macros

Template Helpers

- `ember-composable-helpers`
- `ember-truth-helpers`
  - `eq`, `not-eq`, `not`, `and`, `or`, `xor`, `gt`, `gte`, `lt`, `lte`, `is-array`, `is-equal`
- `ember-promise-helpers`
  - `await`, `promise-all`, `promise-hash`
  - `is-pending`, `is-rejected`, `is-fulfilled`, `promise-rejected-reason`,

Look for the patterns, share your learnings.

Thanks ✌🏽

`@chadian`