



stholm.js + **a cast**



Antoine Lehurt

Staff Software Engineer at Acast

Social networks

@kewah

Blog

<https://blog.kewah.com>

Color Palette

Primary



Swatches

\$Black



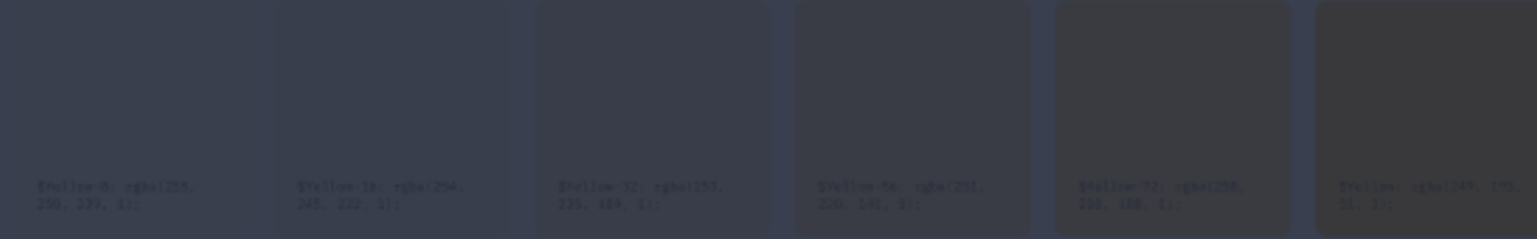
\$White



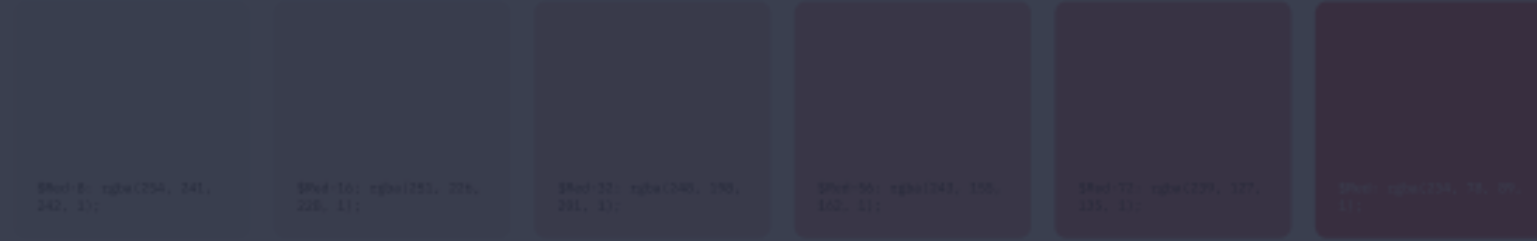
\$Blue



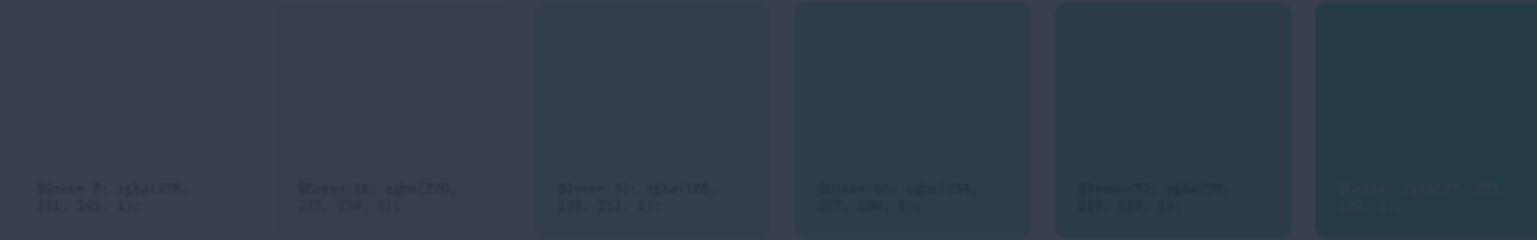
\$Yellow



\$Red



\$Green



Alpha Swatches

\$BlackAlpha



\$WhiteAlpha



\$BlueAlpha



\$YellowAlpha



\$RedAlpha

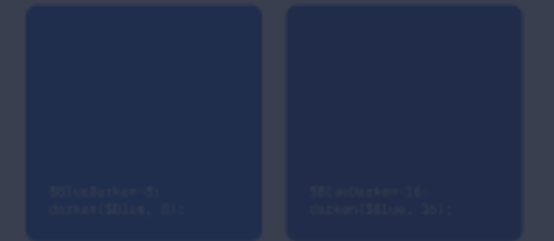


\$GreenAlpha

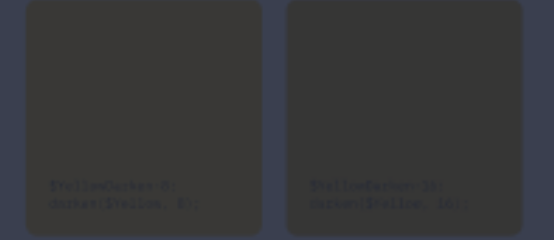


Darken Swatches

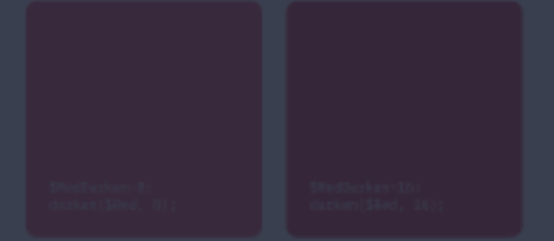
\$BlueDarken



\$YellowDarken



\$RedDarken



\$GreenDarken



The journey to build

Decibel



Oh freedom, oh freedom

Oh freedom, oh freedom

Organisation structure

AUTONOMOUS TEAMS

We are free to choose our tools and way of working.

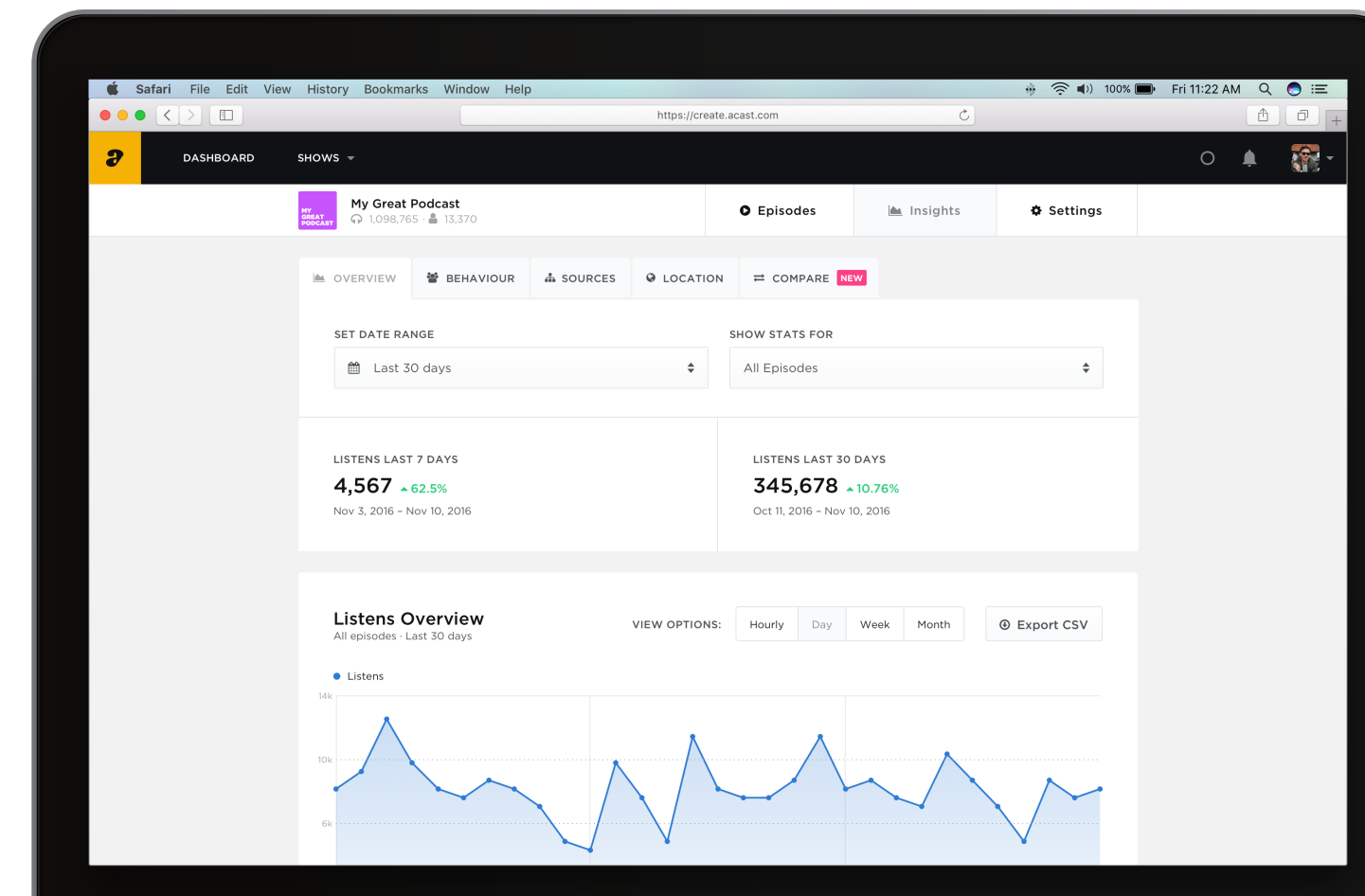
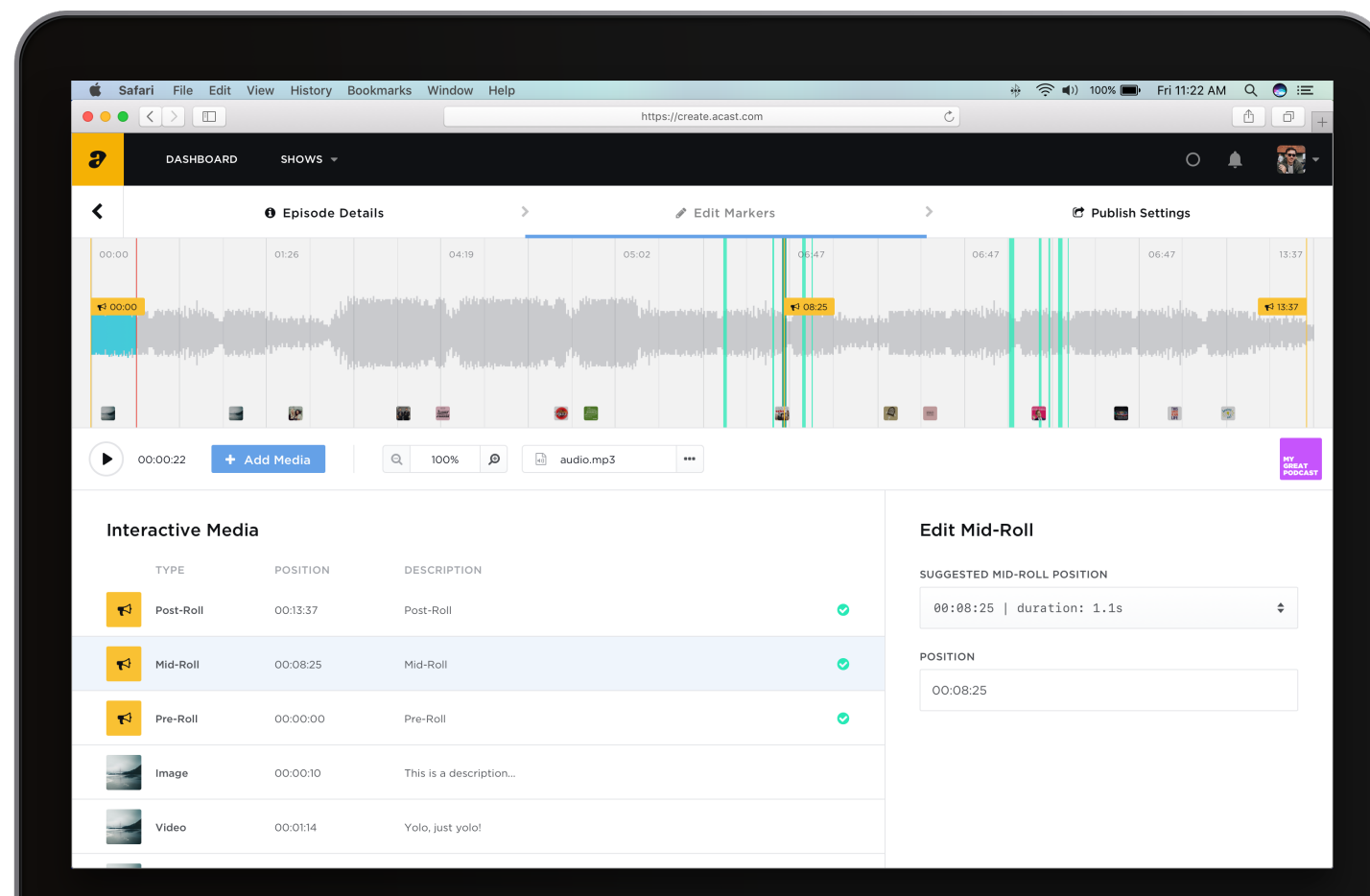
LIMIT DEPENDENCIES BETWEEN TEAMS

We are ok to have duplicated codes if that allows us to move faster.



EACH TEAMS REIMPLEMENT THE SAME UI

- We end up with different UI and UX between projects that are used by the same user
- It takes time when starting a new project



Oh freedom, oh freedom

New product strategy

UNIFYING OUR OFFERING

We wanted to bundle several projects, so the user would have the feeling to only use one tool.

What is a design system?



What is a design system?

“A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build any number of applications.”

<https://www.invisionapp.com/inside-design/guide-to-design-systems/>

What is a design system?

“A design system is a **collection of reusable components**, guided by clear standards, that can be assembled together to build any number of applications.”

<https://www.invisionapp.com/inside-design/guide-to-design-systems/>

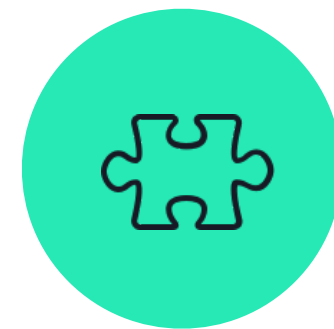
What is a design system?

The purpose of Decibel



Unified UI/UX

The user should feel
at home in all
products



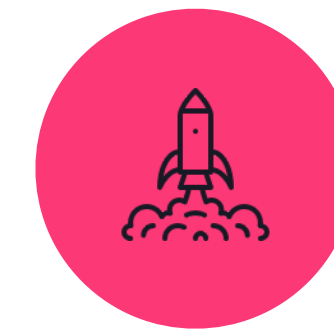
Composable and extendable components

Projects don't have
the same needs



Accessibility support

Live by our
company values



Speed up development

Improve
collaboration

Reduce
development time



Should we build it in-house?

Should we build it in-house?

Different strategies

OPEN SOURCED DESIGN SYSTEMS

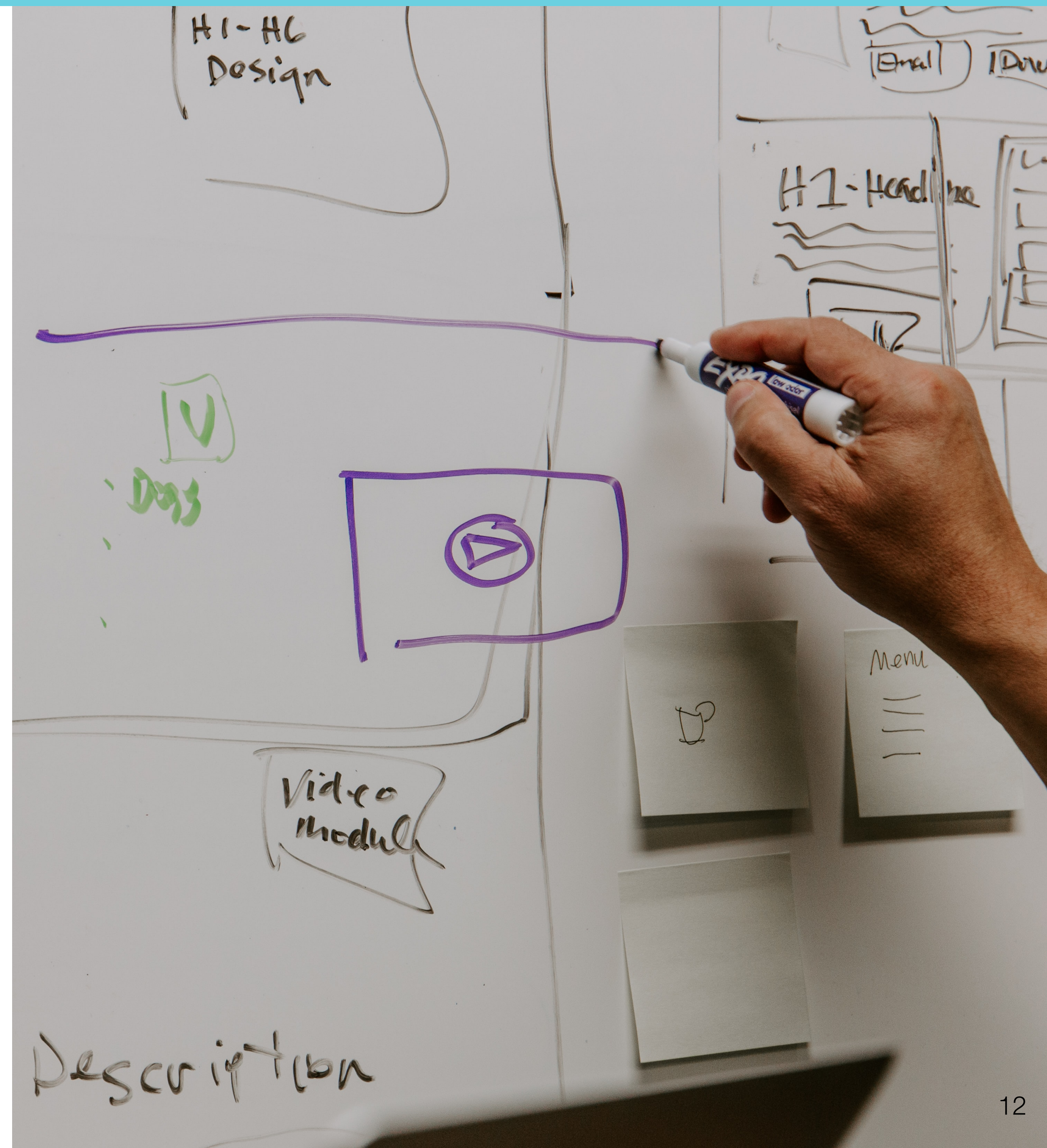
For instance Material Design, Ant Design, etc.

"HEADLESS" LIBRARIES + BRING YOUR OWN UI

Rely on open source libraries to handle complex logic, and you build the UI layer above it.

"DIY"

Build everything from A-Z



Should we build it in-house?

Different strategies



OPEN SOURCED DESIGN SYSTEMS

For instance Material Design, Ant Design, etc.

"HEADLESS" LIBRARIES + BRING YOUR OWN UI

Rely on open source libraries to handle complex logic, and you build the UI layer above it.

"DIY"

Build everything from A-Z



Quickly up to speed

It helps you to only focus on your business value



Opinionated

It comes with well define design principles



Great for teams that have limited resources

Or don't have front-end specialists



Limitation in theming

Might be an issue if branding is important



Might lack accessibility support

Should we build it in-house?

Different strategies

OPEN SOURCED DESIGN SYSTEMS

For instance Material Design, Ant Design, etc.

👉 "HEADLESS" LIBRARIES + BRING YOUR OWN UI

Rely on open source libraries to handle complex logic, and you build the UI layer above it.

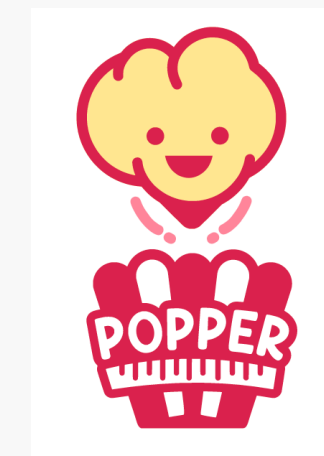
"DIY"

Build everything from A-Z

Low level component libraries for building accessible high level UI libraries



Libraries that focus on solving one problem



Should we build it in-house?

Different strategies

OPEN SOURCED DESIGN SYSTEMS

For instance Material Design, Ant Design, etc.

👉 "HEADLESS" LIBRARIES + BRING YOUR OWN UI

Rely on open source libraries to handle complex logic, and you build the UI layer above it.

"DIY"

Build everything from A-Z

👍 Choose the right tools for the right job

👍 Free to build your UI

👍 Great for teams that have front-end specialists

⚠️ Maintenance

Several moving parts that will require to stay up to do date

Should we build it in-house?

Different strategies

OPEN SOURCED DESIGN SYSTEMS

For instance Material Design, Ant Design, etc.

"HEADLESS" LIBRARIES + BRING YOUR OWN UI

Rely on open source libraries to handle complex logic, and you build the UI layer above it.



"DIY"

Build everything from A-Z

- 👍 Full control of the entire design system
- 👍 Can align UI elements across multiple platforms
- 👍 Great for teams that have large resources

- ⚠️ **Big investment**
Require time and resources

Should we build it in-house?

Different strategies

"HEADLESS" LIBRARIES + BRING YOUR OWN UI

Rely on open source libraries to handle complex logic, and you build the UI layer above it.

Team organisation



Possible workflows

CENTRALISED WORKFLOW: ONE TEAM BUILDING AND MAINTAINING THE DESIGN SYSTEM

- Clear ownership
- Focus only on the design system
- Possibility to build & experiment faster

DECENTRALISED WORKFLOW: "OPEN SOURCE", BUT IN-HOUSE

- Gardeners, promote collaboration
- Shared knowledge across the company



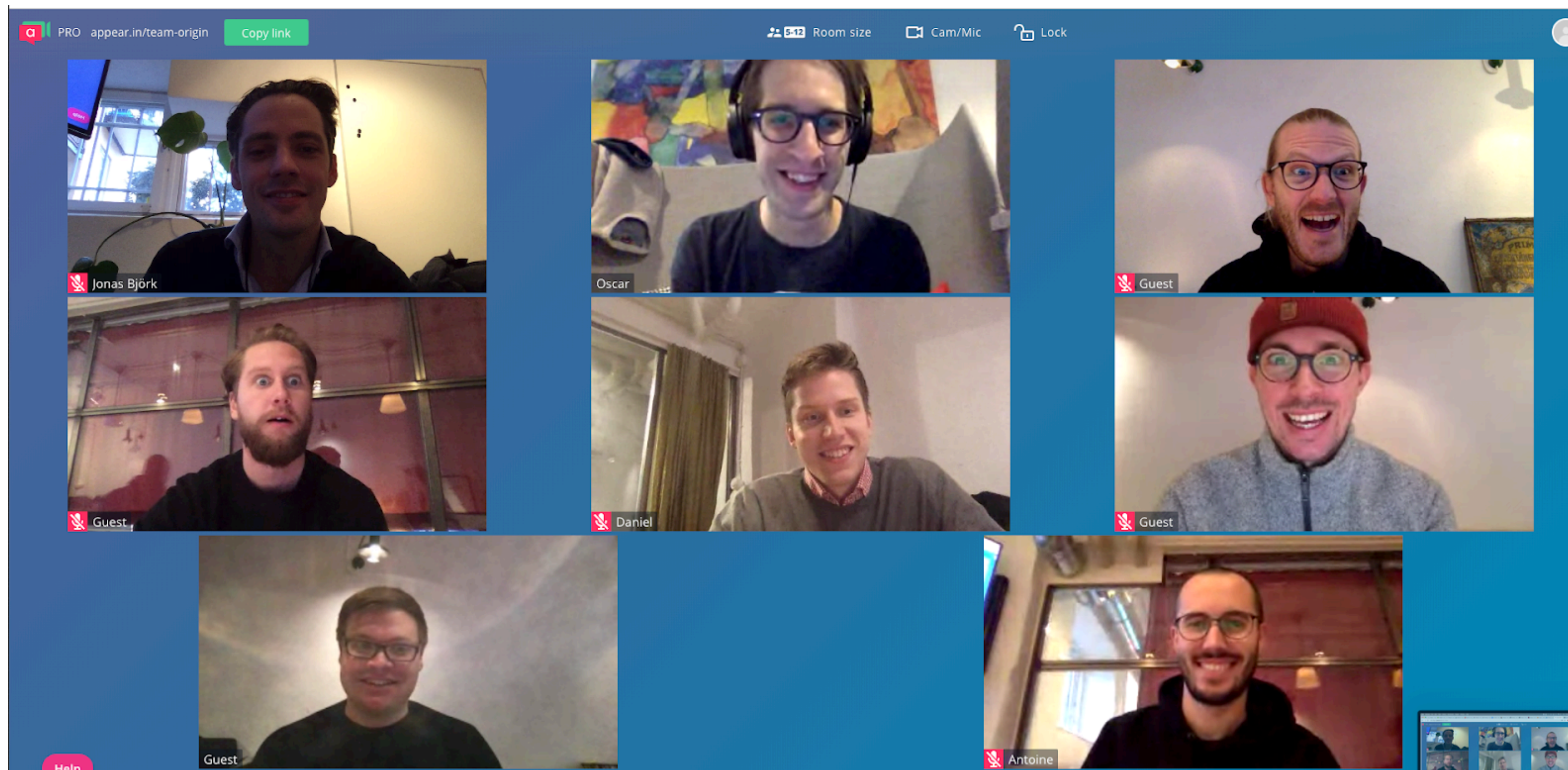
Ready, steady, build!

Ready, steady, build!

Focus week

DISTRACTION FREE TO GET STUFF DONE

We grouped developers from different teams to bootstrap Decibel. The goal was to get a few components built into the component library, and refactor one existing project to include the newly created component.



Ready, steady, build!

Focus week

RHYTHM

We also experimented with a different rhythm than we usually use in our teams.

	Monday	Tuesday	Wednesday	Thursday	Friday
09:00	5-5-5 Sprint #01	5-5-5 Sprint #03	5-5-5 Sprint #05	5-5-5 Sprint #07	5-5-5 Sprint #09
13:00	5-5-5 Sprint #02	5-5-5 Sprint #04	5-5-5 Sprint #06	5-5-5 Sprint #08	5-5-5 Retro Demo

Sprint

- 4 hours
- Plan only what to demo next
- Make decisions as late as possible

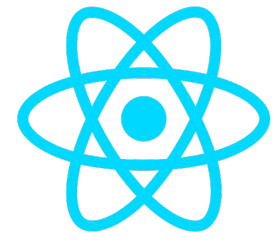
5-5-5

- 5 minutes demo
- 5 minutes retro
- 5 minutes planning



Decibel tech stack





Decibel tech stack

React

NATURAL CHOICE AT ACAST

All web client projects uses React

REACT IS BUILT AROUND COMPONENTS

👉 TIPS

Favour compound components*

Provide a flexible and declarative API

```
const data = [  
  { id: 'tab-a', label: 'Tab A', content: <p>Content of tab A</p> },  
  { id: 'tab-b', label: 'Tab B', content: <p>Content of tab B</p> },  
  { id: 'tab-c', label: 'Tab C', content: <p>Content of tab C</p> },  
]
```

Component

```
<Tabs  
  data={data}  
  selected="tab-a"  
  onChange={tabChangedHandler}  
>
```

Compound component

```
<Tabs selectedTab="tab-a"  
  onChange={tabChangedHandler}>  
  <TabList>  
    {data.map(({ id, label }) => {  
      return (  
        <Tab key={id}>  
          {label}  
        </Tab>  
      )  
    })}  
  </TabList>  
  <  
    {data.map(({ id, content }) => {  
      return (  
        <TabPanel key={id}>  
          {content}  
        </TabPanel>  
      )  
    })}  
  </>  
</>
```

* You can learn more in the Talk from Ryan Florence
<https://www.youtube.com/watch?v=hEGg-3pIHIE>



Styled components

COMPONENT-ORIENTED STYLES

ABLE TO USE THE SAME JAVASCRIPT TOOLING

HASHES THE CSS CLASSES ON THE OUTPUT

Prevent conflicts between style rules

Prevent targeting CSS classes in the tests

⚠ WE COULD HAVE CHOSEN LINARIA INSTEAD

👉 TIPS

Use Styled System to unify the API

It provides a nice API for overriding styles on your components.

```
import styled from 'styled-components'
import { space, color } from 'styled-system'

const Box = styled.div`
  ${space}
`

<Box mr={2} /> // margin-right: 8px

<Box color="red" /> // Not supported

// To support colors
const Box = styled.div`
  ${space}
  ${color}
`

<Box color="red" /> // Works!
```

Explicit about which style attributes can be overridden

```
<Box mr={2} />

// VS
const StyledBox = styled(Box)`
  margin-right: ${unit(2)};
`

<StyledBox />
```



Styled components

COMPONENT-ORIENTED STYLES

ABLE TO USE THE SAME JAVASCRIPT TOOLING

HASHES THE CSS CLASSES ON THE OUTPUT

Prevent conflicts between style rules

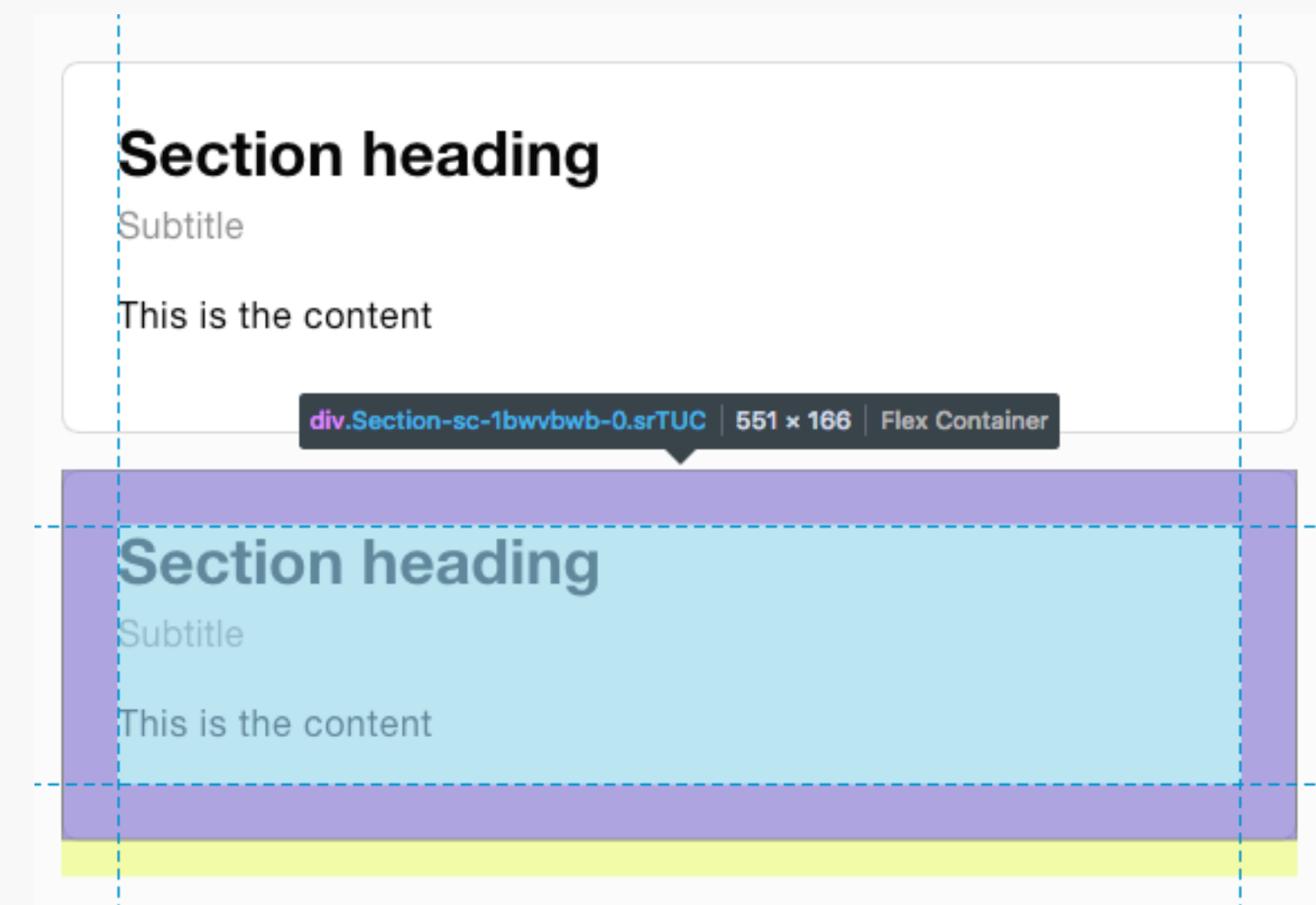
Prevent targeting CSS classes in the tests

! WE COULD HAVE CHOSEN LINARIA INSTEAD

👉 TIPS

Avoid outer margins on your components

Use a "stack" component instead



```
<Section>...</Section>
<Section mb={0}>...</Section>

// VS

<Box gap={3}>
  <Section>...</Section>
  <Section>...</Section>
</Box>
```



Accessibility

AUTOMATION STILL AT EARLY STAGE



RECOMMENDED PLUGINS

Axe ([Firefox](#) & [Chrome](#))

React plugin ([react-axe](#))

Cypress plugin ([cypress-axe](#))

```
▼ New axe issues
▶ serious: Elements must have sufficient color contrast https://dequeuniversity.com/rules/axe/3.4/color-contrast?application=axeAPI
▶ critical: Images must have alternate text https://dequeuniversity.com/rules/axe/3.4/image-alt?application=axeAPI
▶ critical: Form elements must have labels https://dequeuniversity.com/rules/axe/3.4/label?application=axeAPI
▶ moderate: Aside must not be contained in another landmark https://dequeuniversity.com/rules/axe/3.4/landmark-complementary-is-top-level?application=axeAPI
▶ moderate: Main landmark must not be contained in another landmark https://dequeuniversity.com/rules/axe/3.4/landmark-main-is-top-level?application=axeAPI
▶ moderate: Document must have one main landmark https://dequeuniversity.com/rules/axe/3.4/landmark-one-main?application=axeAPI
▶ moderate: Ensures landmarks are unique https://dequeuniversity.com/rules/axe/3.4/landmark-unique?application=axeAPI
▶ serious: Links must have discernible text https://dequeuniversity.com/rules/axe/3.4/link-name?application=axeAPI
```

```
describe('doc example', () => {
  beforeEach(() => {
    cy.visit('http://localhost:9000');
    cy.injectAxe();
  });

  it('Has no detectable a11y violations on load', () => {
    cy.checkA11y();
  });

  it('Has no a11y violations after button click', () => {
    cy.get('button').click();
    cy.checkA11y();
  });
});
```





React Testing Library

WE MIGRATED AWAY FROM ENZYME

Enzyme provides API to test internal states

```
.state(), .setState(), .instance()
```

RTL HELPS US TO WRITE BETTER TESTS

Test what the users interact with, avoid testing implementation detail

👉 TIPS

Avoid snapshot testing

It doesn't give enough information to the developer about what you are asserting

```
wrapper.find('TabButton').at(1).setState({ selected: true })  
expect(wrapper.find('TabButton')).toMatchSnapshot()
```

- 👉 Rely on implementation detail to update `selected`
- 👉 We don't know what is important in the snapshot.
The test might fail because we change a tag name.
Do we need to catch that? Will it help us to catch a bug?

```
fireEvent.click(getByText('Tab A'))  
expect(getByText('Tab A')).toHaveAttribute('aria-hidden', 'false');
```

- 👉 Fake user behaviour
- 👉 Explicit about what we expect



Storybook

GATHER PLAYGROUND FOR MANUAL TESTING AND DOCUMENTATION

The screenshot shows the Storybook Decibel interface. The left sidebar contains a component tree with 'Date Range' selected. The main canvas displays a 'Date Range' component with two input fields labeled 'Start date' and 'End date'. Below the canvas, the 'Knobs' panel is visible, showing configuration options for the component: 'months' (2), 'placeholder (comma separated)' (Start date,End date), 'openOnFocus' (checked), 'closeOnSelect' (checked), 'renderSidebar' (unchecked), 'renderFooter' (unchecked), 'localization.locale' (en), and 'localization.weekStartsOn' (0). 'Copy' and 'Reset' buttons are at the bottom right of the knobs panel.

The screenshot shows the 'Docs' view for the 'DateInput' component. The title is 'DateInput' and the description is 'This component is a form element for selecting Dates.' Below this is a 'Properties' table:

Name	Description	Default
value	Controlled selected date or date range (array with a start date and end date) Date Date[]	-
defaultValue	Initial value for uncontrolled selected date or date range Date Date[]	-
range	Display date range inputs bool	false
openOnFocus	Toggles if the date picker should be displayed when focusing input bool	true
closeOnSelect	Toggles if the date picker should close when dates have been entered bool	true
months	Number of month pages to show number	1



Future of Decibel



Future of Decibel

EXTEND THE COMPONENTS BEYOND THE CORE COMPONENTS.

IMPROVE OUR COLLABORATION WITH DESIGNERS

We will scale the design team as well, so we need to write down our design principles

APPLY THE DESIGN SYSTEM AT A LARGE SCALE IN THE COMPANY

Right now Decibel is tightly coupled to the project to unify our offering.

CONTINUE TO ITERATE TO IMPROVE THE CONTRIBUTION IN THE COMPONENT LIBRARY

Decibel Design Language

Acast Design System

OWNER

@design

STATUS

Work in Progress

Tack!

Time for Q&A 🙌

