

From manual chaos To automated scale

Important!

This case-study is brief, in this case-study certain elements will be hidden / renamed / redacted due to non-disclosure!

Title

What_is_my_title

ReplaceMe with a fixed section

StepStepStep

I am a list item

!! am the hover/focussed-Item !!

I am a list item

I am a list item

I am a list item

I am a list item

I am a list item

I am a list item

CloseNext

config: sheets

From this file

TypeModal

Show Blade title

Show Notification

Show filter-search (flow)

I am a Fixed Element

Show Fixed section

Show stepper-row

Show dropdown

NESTED: Sidesheet/ Header configurations

Pagination

Blade title

modeDefault

border top

border bottom

Show check

Show error

Hide title

config: clean list

array of elements

spacing

Place list item here

Place hover/focussed-i...

spacing (top-bottom)

NESTED: Sidesheet/ Actions configurations

Select type

Hi, I'm **Cedric Liebens**, A **Data & AI driven Product design** transformation lead

I accelerate **enterprise delivery in 90 days** by fixing
design systems and stakeholder alignment in B2B and regulated environments.

15+ years | Cross-industry pattern recognition. Pharma · Gov · B2B SaaS.
Outcome-first | Bridging Business, Design & Development.



No system, no components, drowning in manual work

My role:
Problem Discovery + Developer Research

Layers

FeatureName

FIXED

topbar

m3icons-imported/help

m3icons-imported/help

Badge

top bar

SCROLLS

Frame 4050

Frame 4036

table

Frame 4029

Header

Divider

myprofile-card

myprofile

technical permissions

Tabs

container

activity

header

mask

Frame 4042

Frame 4038

Rectangle 12325

bottom actions

vertical main menu

Frame 4037

Rectangle 12322

FeatureName

Arxus

Feature name

32

FL

Firstname Lastna.

Company

>Lorem ipsum

Tab

Tab

Tab

Search

+

Action

Filter

Status

Contacts & Users list

Firstname Lastname

Function

subject

Status

Firstname Lastname

Function

subject

Status

Firstname Lastname

Function

subject

Status

Firstname Lastname

Function

subject

Status

Firstname Lastname

Function

subject

Status

Firstname Lastname

Function

subject

Status

AB

Firstname Lastname

Function

subject

Status

Firstname Lastname

Function

subject

Status

Firstname Lastname

Function

subject

Status

Firstname Lastname

Function

subject

Firstname Lastname

Firstname Lastname

Function

Email

Number

USER

Action

Subject

Subject

Subject

Subject

Subject

Tab

Tab

Subject

Subject

Description

5 jun. 2023 13:56:52

email@arxus.eu

Lorem ipsum

5 jun. 2023 09:53:39

email@arxus.eu

Lorem ipsum

2 jun. 2023 14:00:00

1 jun. 2023

602 × 635

“

Every time we need a new feature, we wait days for design. We end up building it ourselves.

Hendrik P.
Developer

As-is

- ✗ No design system (designed from scratch everytime)
- ✗ No Auto Layout (manual resizing for every breakpoint)
- ✗ No component library (40 devs waiting on 1 designer)
- ✗ Previous work: Groups/frames (not scalable)
- ✗ Zero configurability (text, colors hardcoded)

Impact

- 1 feature = weeks of design work
- 40 developers blocked by design bottleneck
- Inconsistency across product (no source of truth)
- Unsustainable as product scales

Why is it important

Roadmap

Design capacity

Roadmap stakes

- Product roadmap blocked by design capacity
- 40 devs waiting = massive waste
- Competitors shipping faster

Human

Designer burnout

Human stakes

- Designer burnout (unsustainable workload)
- Developer frustration (blocked, building own "solutions")
- Quality suffering (devs not trained in UX)

Financial

Time-to-market

Financial stakes

- Slower time-to-market = lost revenue
- Inconsistent product = poor UX = churn
- Design debt compounding (harder to fix later)

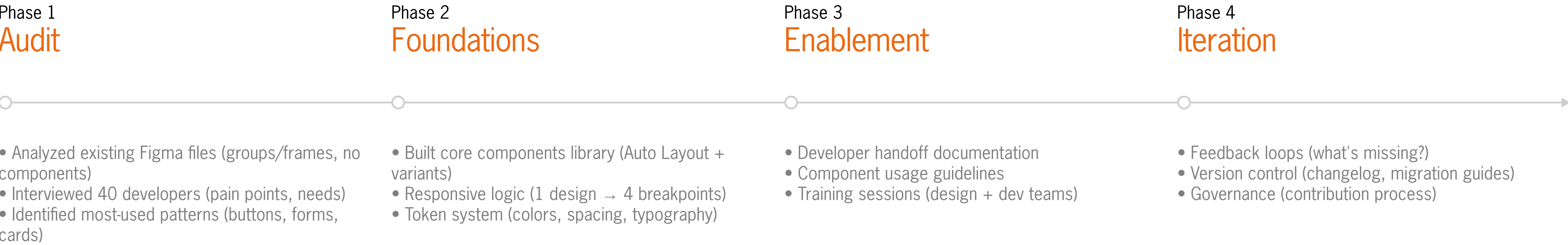
Credibility

Design perception

Credibility stakes

- Design seen as bottleneck, not enabler
- "Just let devs do it" mindset growing
- Risk of design being sidelined

System-first thinking



Constraints I had to navigate

- 40 devs already building (couldn't stop product)
- Legacy components in production (migration needed)
- Multi-platform (web, mobile considerations)

What I learned from 40 developers

Quotes

I don't know which button style to use

Spacing is inconsistent, I guess every time

Mobile layouts? I just shrink desktop

I need design for every breakpoint or I improvise

Key insights

Devs WANT design guidance
(not trying to undermine)

Manual breakpoints = designers drowning

No naming conventions = confusion

No single source of truth = drift

Needs

Self-service component library
(don't wait for designer)

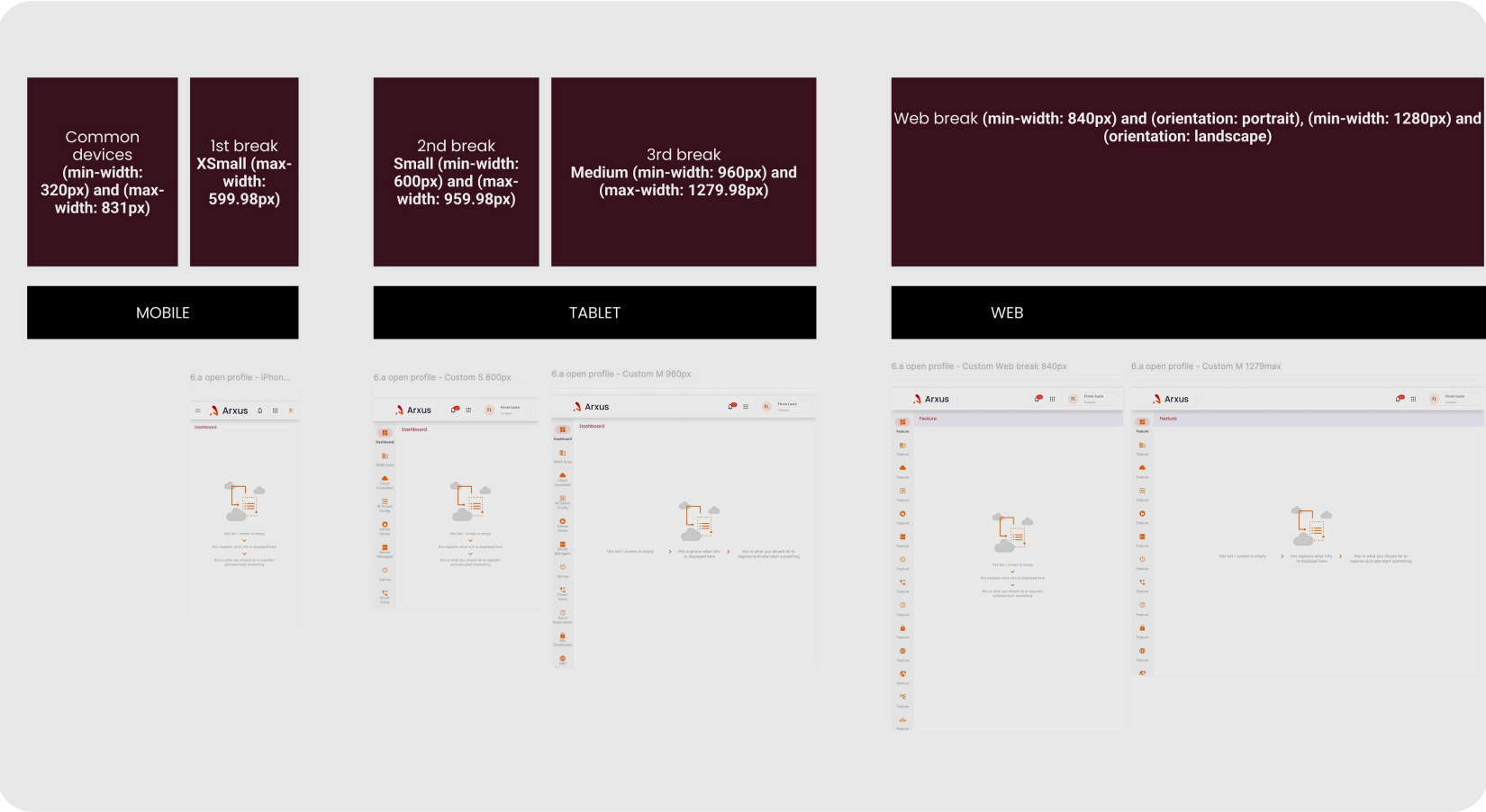
Responsive logic built-in
(not manual per breakpoint)

Clear naming
(Button/Primary)

Documentation
(when to use what)

Arxus Design System Architecture

Before Manual Hell

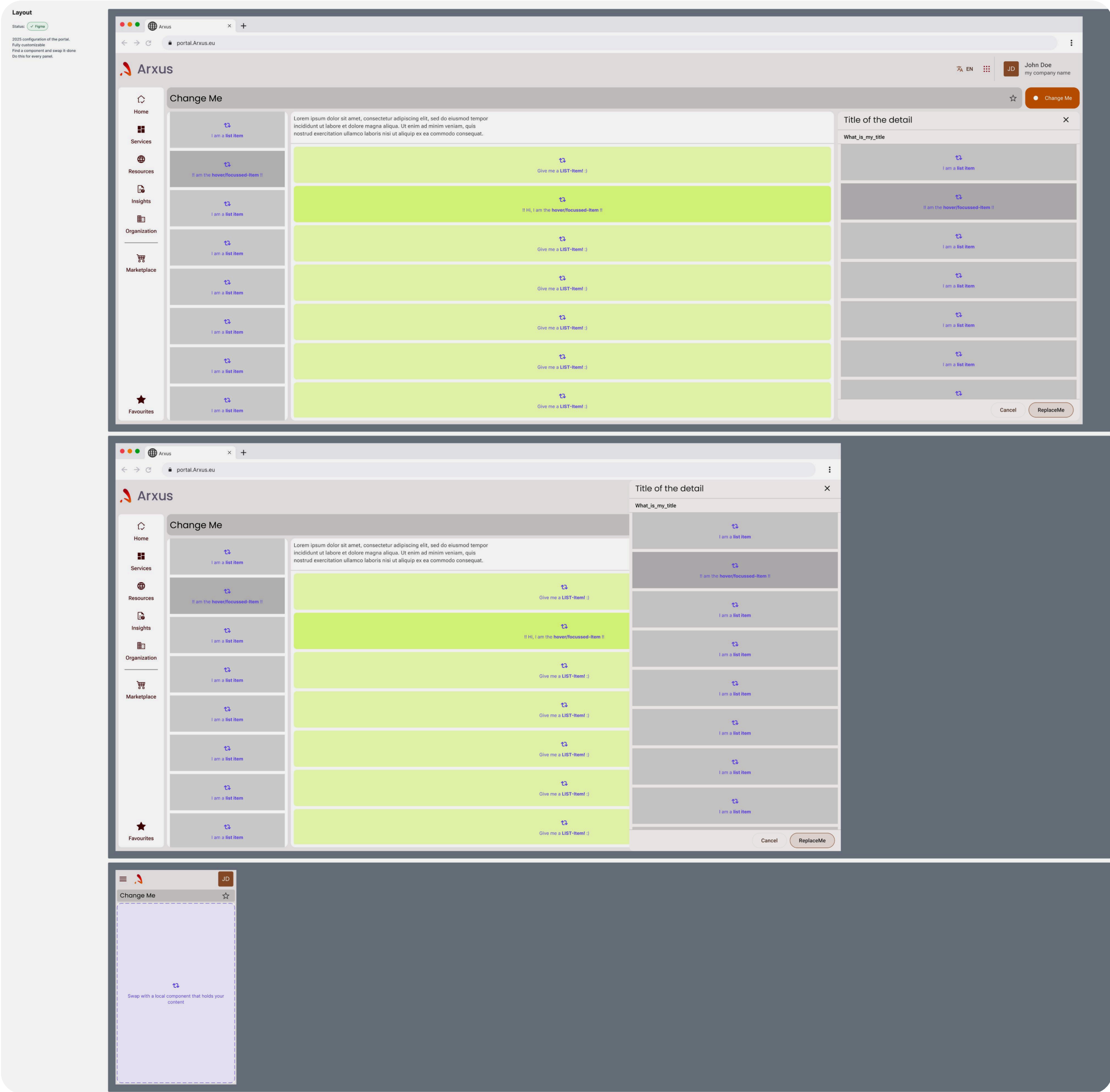


Designer creates:

- 1. Desktop wide
- 2. Desktop
- 3. Tablet
- 4. Mobile

= 4 separate designs x weeks of work

After Auto Layout + Variants



Designer creates Feature layout once

System auto-generates:

- Desktop wide (constraints + spacing rules)
- Desktop (same component, reflows)
- Tablet (stack where needed)
- Mobile

1 design x 30 minutes

Arxus Design System Architecture

My role:
Component Architecture + Responsive Design

Patterns Composed

- Feature layouts:
responsive, 4 breakpoints
- Pane templates
- Flow templates

Components Building blocks

- Buttons
- Forms
- Cards
- Navigation
- 50+ components total

Tokens Foundations

- Colors
- Typography
- Spacing
- Radius, shadows, etc.

Key innovation

- 1 feature layout auto-generates 4 responsive views
Desktop wide, desktop, tablet, mobile
- Plug and play: drop components in instance slots

Feature builder: Logic-Based Components

Once we define how features should work, we build **components with embedded logic**.
Only variable elements require design effort.

Changes to logic?

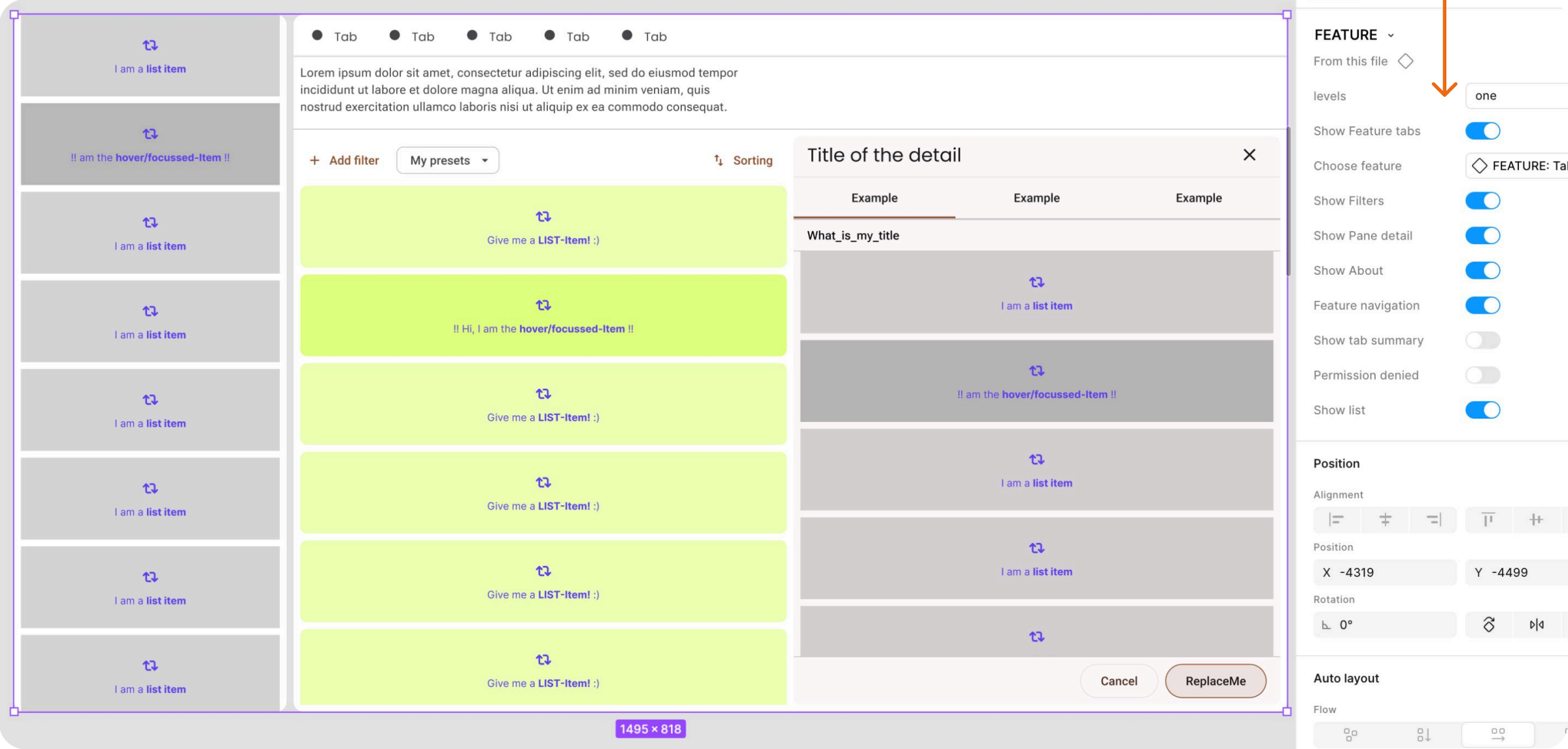
- Update once in the component
- Automatically pushes to all instances
- Zero manual effort

Fully Configurable:

Settings can cover multiple user stories
Future logic changes = single update

User story coverage:

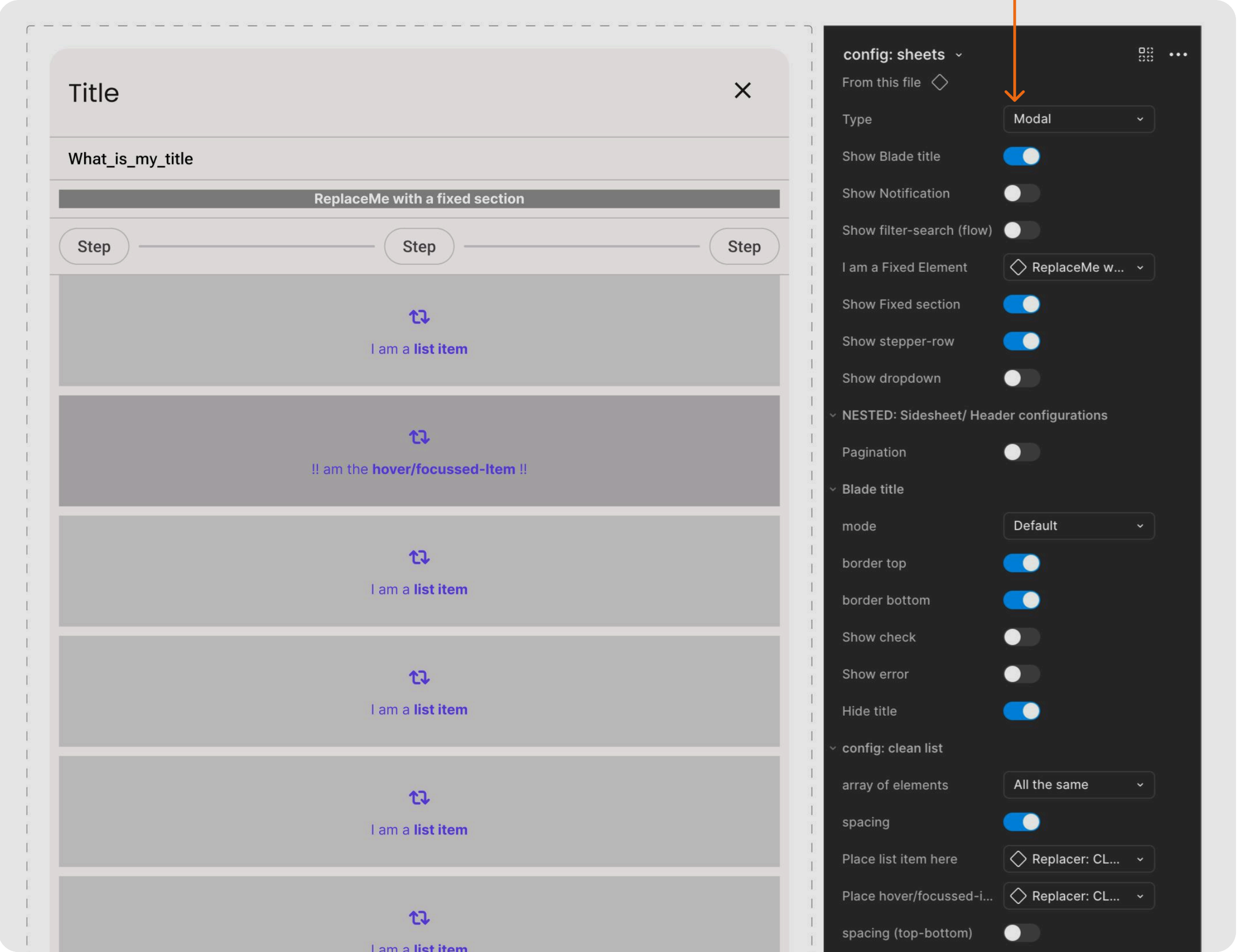
By changing the settings, we can cover different user stories.



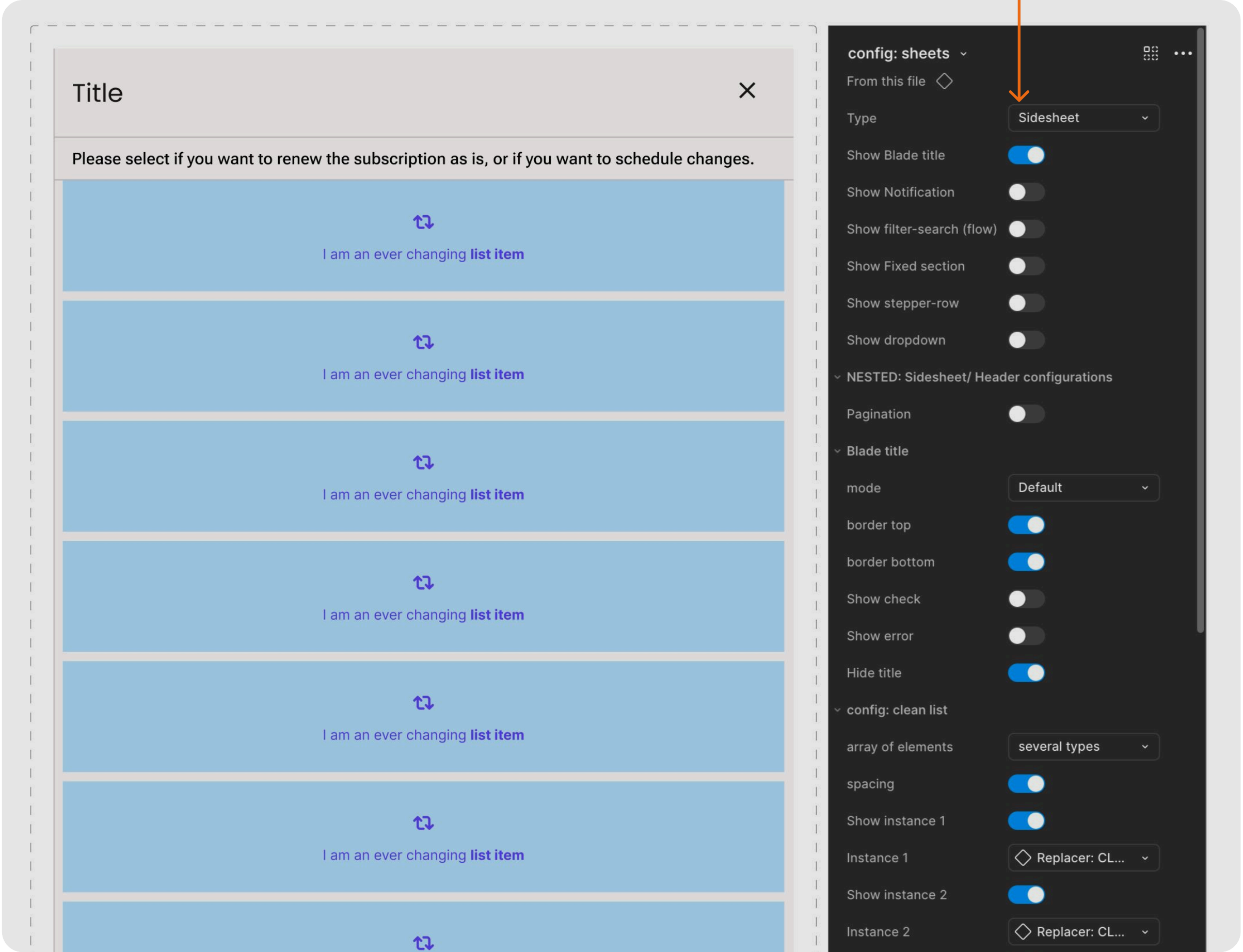
Using the power of Figma

Same component, Different contexts.
One component to build flows using Figma's instance swap + component variants

Scenario 1: Generic list (all same elements)
Modal with stepper + uniform list



Scenario 2: Mixed list (different elements)
Sidesheet without stepper + mixed list



Instance-based design system

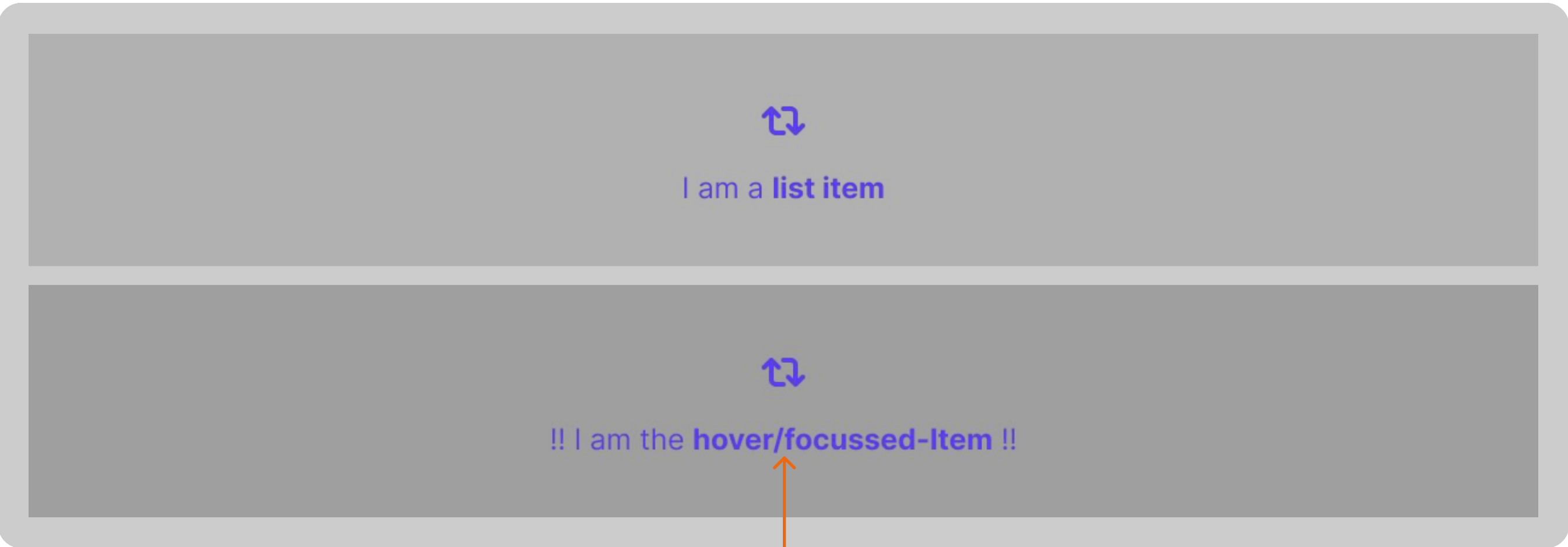
When building features, designers focus ONLY on:

- Atoms (buttons, inputs)
- Molecules (cards, forms)

(Following Atomic Design principles)

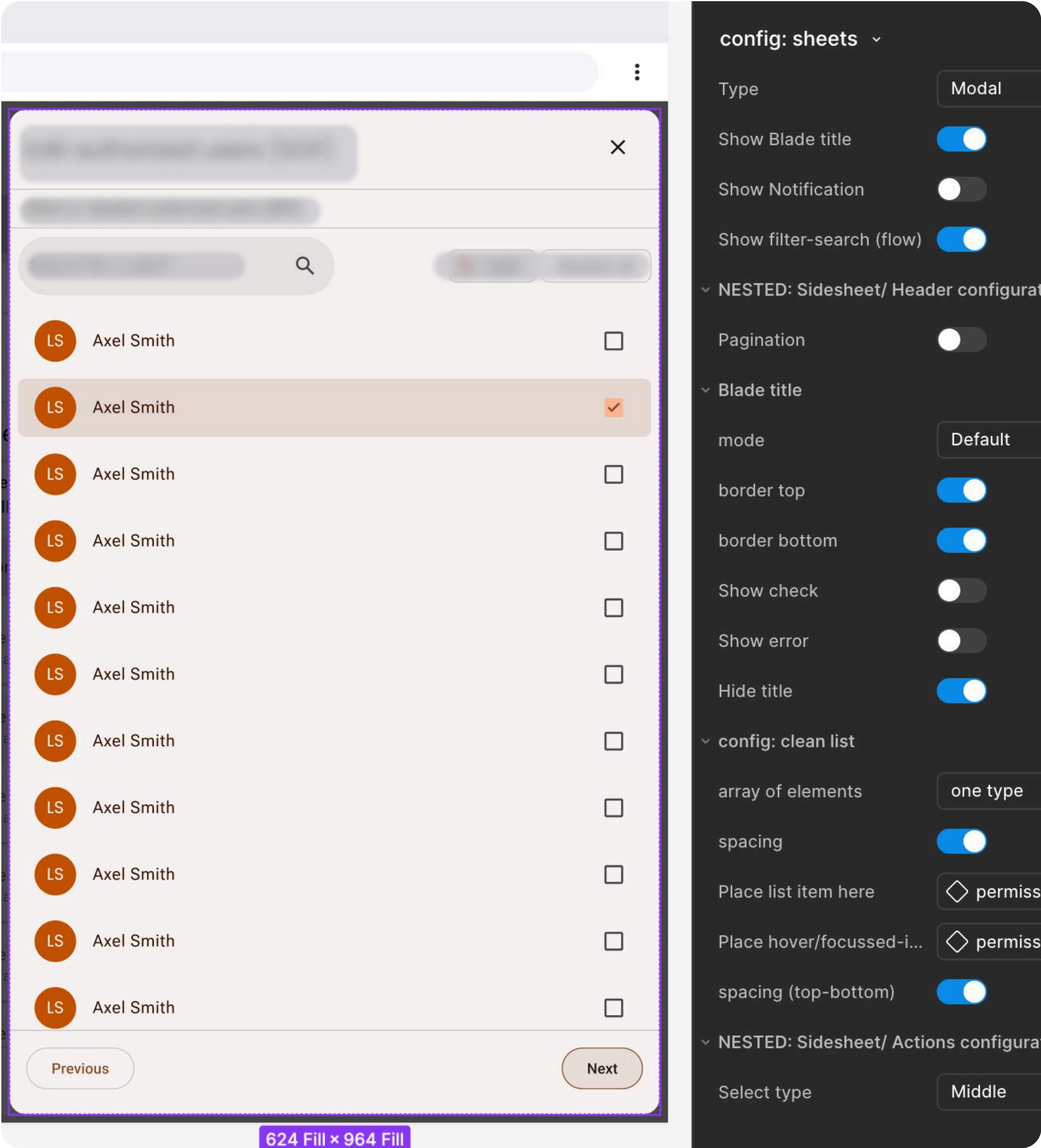
Drop components into instance slots

→ Feature built in minutes, not hours/days



State management:
Need a specific state for a use case?
→ Already built-in
→ No detaching needed
→ Swap instances, configure properties

Result:
Changes = instant (not hours/days)



Variant logic + docs at scale.

My role:
Impact Measurement + Adoption Tracking

Component documentation and explanation

Arxus

LS List item

☐

LS List item

☒

LS Overline List item

☐

Current variant

Selected

No

Lines

1

Select/Radio

Select

Monogram

Yes

Current variant

Documentation ... Docu... Docu... Docume...

Docu...

Docu... Documentation ...

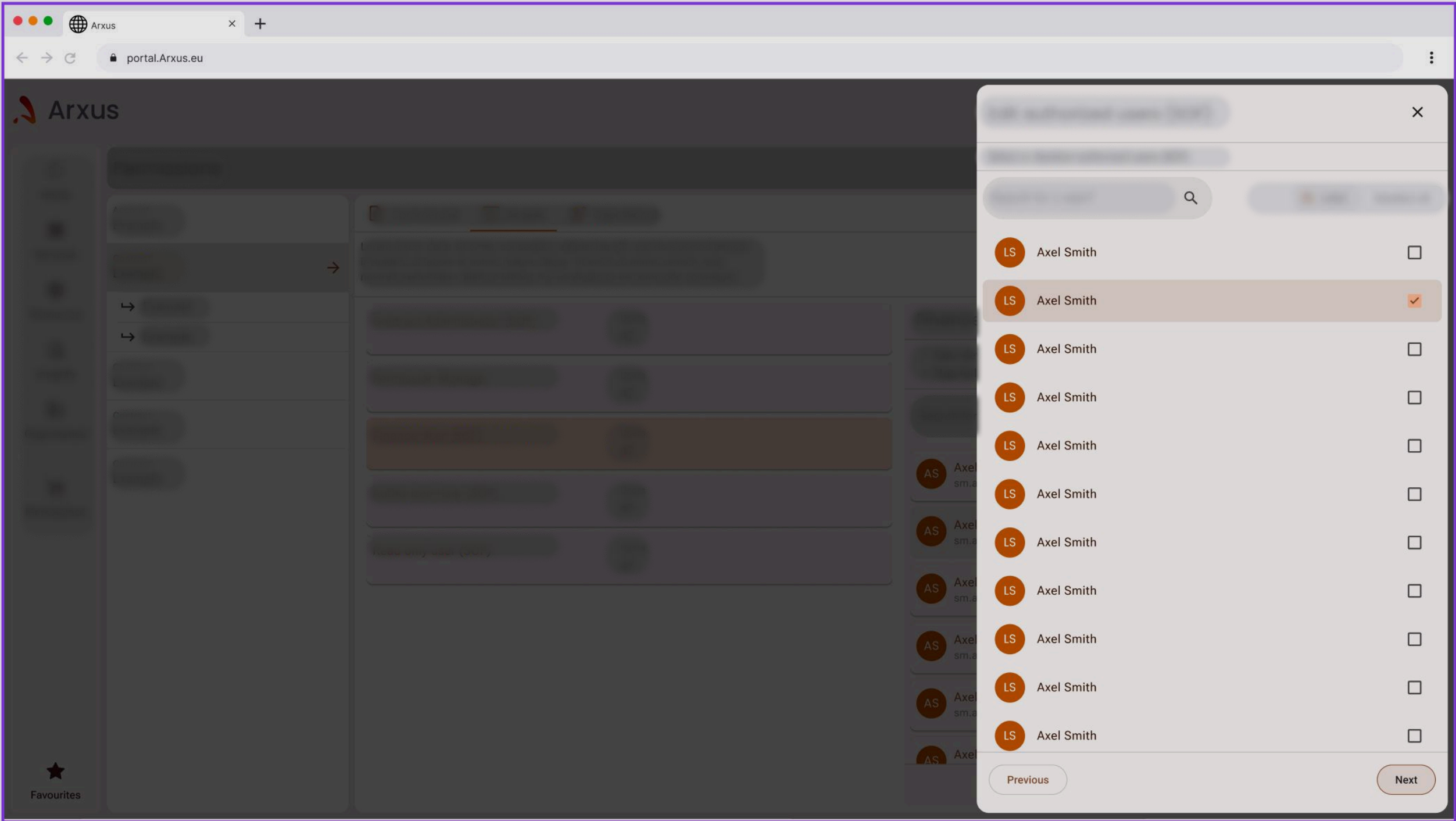
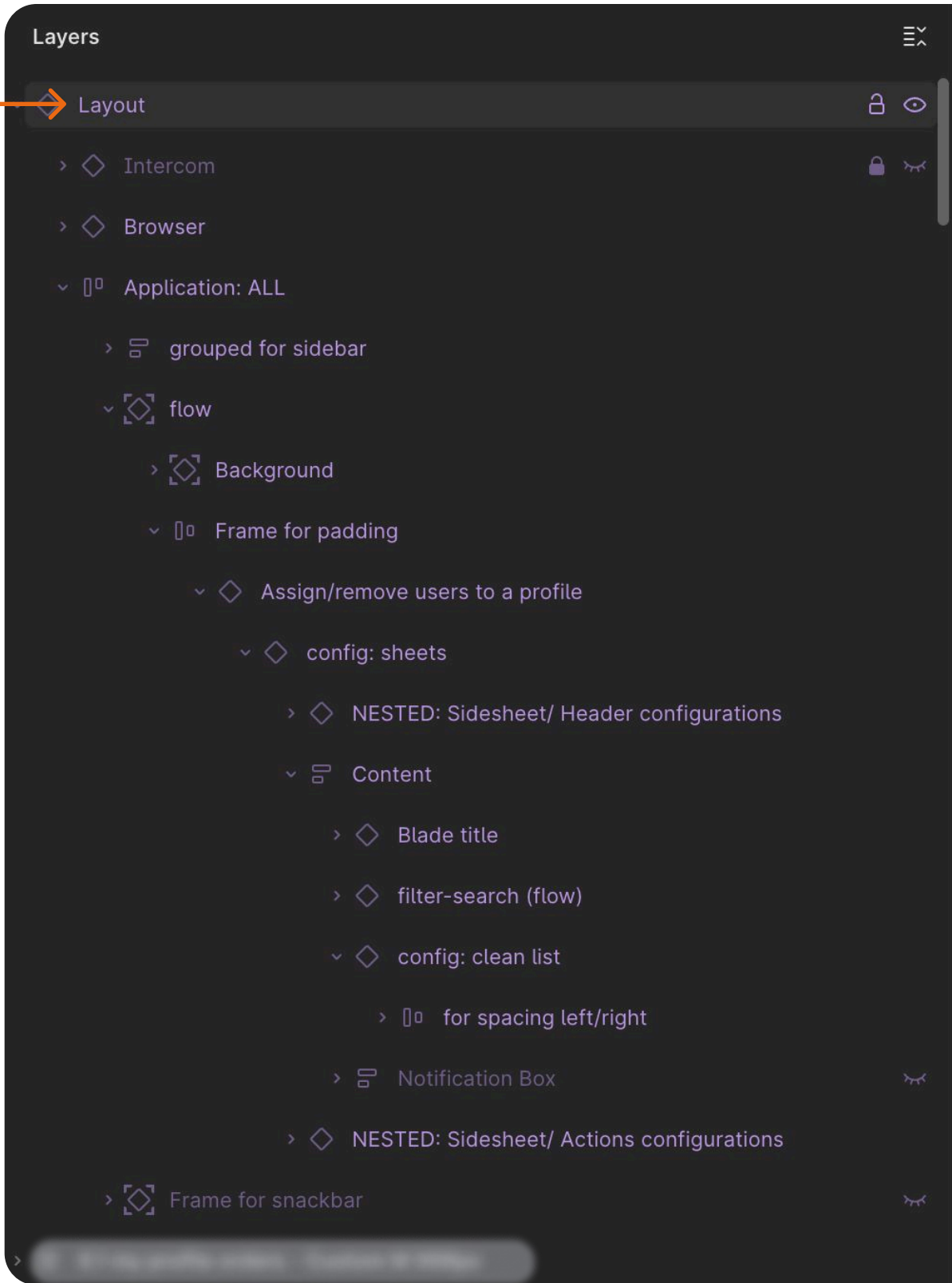
Doc... Docu... Docu... D... Docu... Doc... Doc... Doc... Doc... Doc... Doc... Doc... Doc...

Docu... Docu... Docum... Docum... Docum...

Scalable, maintainable & fast!

My role:
Impact Measurement + Adoption Tracking

This entire feature is a scalable component.



Velocity

- From weeks → 30 minutes (95% faster)
- Capacity: 1 feature/month → 10+ features/month
- Dev unblocked: self-service components

Quality

- Fragmented consistency to 100% system-based
- Responsive coverage: 25% → 100%
- Accessibility: Inconsistent → AA baseline built-in

Team

- Designer burnout → sustainable workload
- Dev satisfaction: "Finally we have guidance!"
- Design credibility: Bottleneck → enabler

Business

- Faster time-to-market (weeks saved per feature)
- Product quality: Consistent UX = better retention
- Scalability: System supports 10x more features

Thank you for watching, here are 3 things I learned from this:

Process

I build systems that scale

- Solving 1 feature $\neq$ solving the problem
- Build the system that makes 1000 easy
- 1 feature/month $\rightarrow$ 20+ features/month

Financials

I think ROI

- Feature design: weeks $\rightarrow$ 30 minutes
- Designer capacity: 1 $\rightarrow$ 20+ features/month
- Dev unblocked: 40 people self-service

Documentation

I document and train

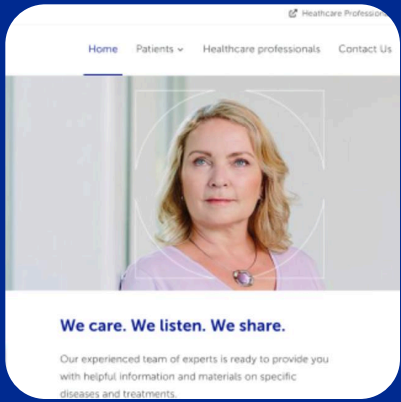
- Great system unused = wasted effort
- Adoption through workshops
- 100% of new features use system

Other cases

UX researcher | UCB Pharma

From medical Jargon to patient empowerment

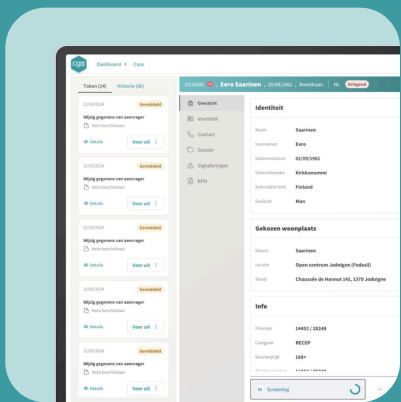
[View Case](#)



Senior product designer | IBZ-FOD CGVS

From Excel workaround to structured workflows

[View Case](#)



Contact

cases.liebensmcedric.com
mail@liebensmcedric.com
+32 479 40 90 73