

Where we are now...

Not logged in

Slovenski +

Welcome to Vizor Portal

Vizor Portal is an on-line application that manages all of your formal communications with your regulator. Using Vizor Portal, you will be able to complete and submit all required returns on-line. In addition, you will be able to see your company's profile – the current information held by the regulator about your company – and keep it up to date.

To access Vizor Portal, please enter your username and password below (password is case sensitive):

Username:

Password:

! Your sign in information is not valid. Please try again.



SUPERVISION CENTRE

The recommended r

- We support: Safari.
- For viewing y Adobe Acrobat
- Forms can b
- A connector
- We also rec

© Vizor 2000-2022

Vizor Administrator

Slovenski +   

Welcome to Vizor Supervision Centre

This system provides functionality for the regulation and supervision of regulated institutions.

Using Vizor Supervision Centre, users will be able to:

- View all submissions from regulated companies, ranging from periodic financial information to notifications.
- Approve new submissions from companies, track overdue submissions and grant/deny resubmission requests.
- View the profile information held for a regulated company.
- Report on company profiles to show summary information for companies.
- Analyse, aggregate and summarise financial data using the report generation functionality.
- View reports that other users have published.

Administrative users can also:

- Manage users of Vizor Supervision Centre and Vizor Portal.
- Create companies along with their principal users.
- Create returns for all regulated companies of a certain type in one go.

To begin using Vizor Supervision Centre, select an option from the menu bar.

Abacus Regulator
Regulator Developer

- [Reporting Status](#)
- [Scheduled Reporting](#)
- [Expected Submissions ESFS](#)
 - EBA XBRL Reporting
 - EIOPA Pension Funds
 - EIOPA Pension Funds PEPP
 - EIOPA Solvency II
 - Expected Submissions Sample
 - Unexpected Submissions
 - Unscheduled Reporting
 - Reports
 - Analytics
 - Configuration
 - Administration

EBA XBRL Reporting Expected Submissions

Filter results...

Details	Reporting Entity ID	Reporting Entity Name	Reporting Date	Submission De...	Submission Timestamp
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	15/07/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	15/07/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	15/07/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	15/07/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
	11102	Skandinaviska Enskilda ...	31/12/2015	11/02/2016	
	11102	Skandinaviska Enskilda ...	31/12/2015	11/02/2016	

[+ Add Expected Submission](#)
 [Upload Expected Submissions](#)

11/08/2015
11/08/2015
11/02/2016
11/02/2016

missions					
Details	Reporting Entity ID	Reporting Entity Name	Reporting Date	Submission De...	Submission Tim
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	15/07/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	15/07/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	15/07/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	15/07/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	30/06/2015	11/08/2015	
i	11102	Skandinaviska Enskilda ...	31/12/2015	11/02/2016	
i	11102	Skandinaviska Enskilda ...	31/12/2015	11/02/2016	
i	11102	Skandinaviska Enskilda ...	31/12/2015	15/01/2016	

Upload Report

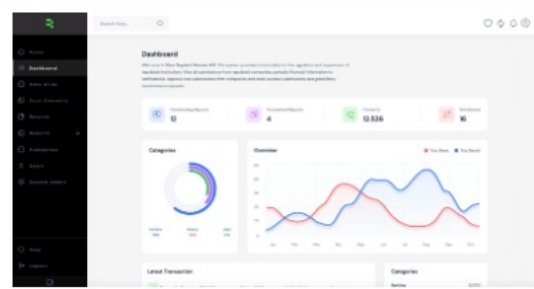
Where do we want to be...?

Common UI

Application Design Proposal



Design System



Dashboard

A screenshot of a data grid interface. It features a sidebar with navigation links like 'Home', 'Dashboard', 'Reports', 'Settings', and 'Logout'. The main content area is titled 'Data Grids' and contains a table with columns: 'Name', 'ID', 'Status', 'Role', 'Age', 'Project', 'Start Date', and 'Action'. The table lists several entries with their respective details.

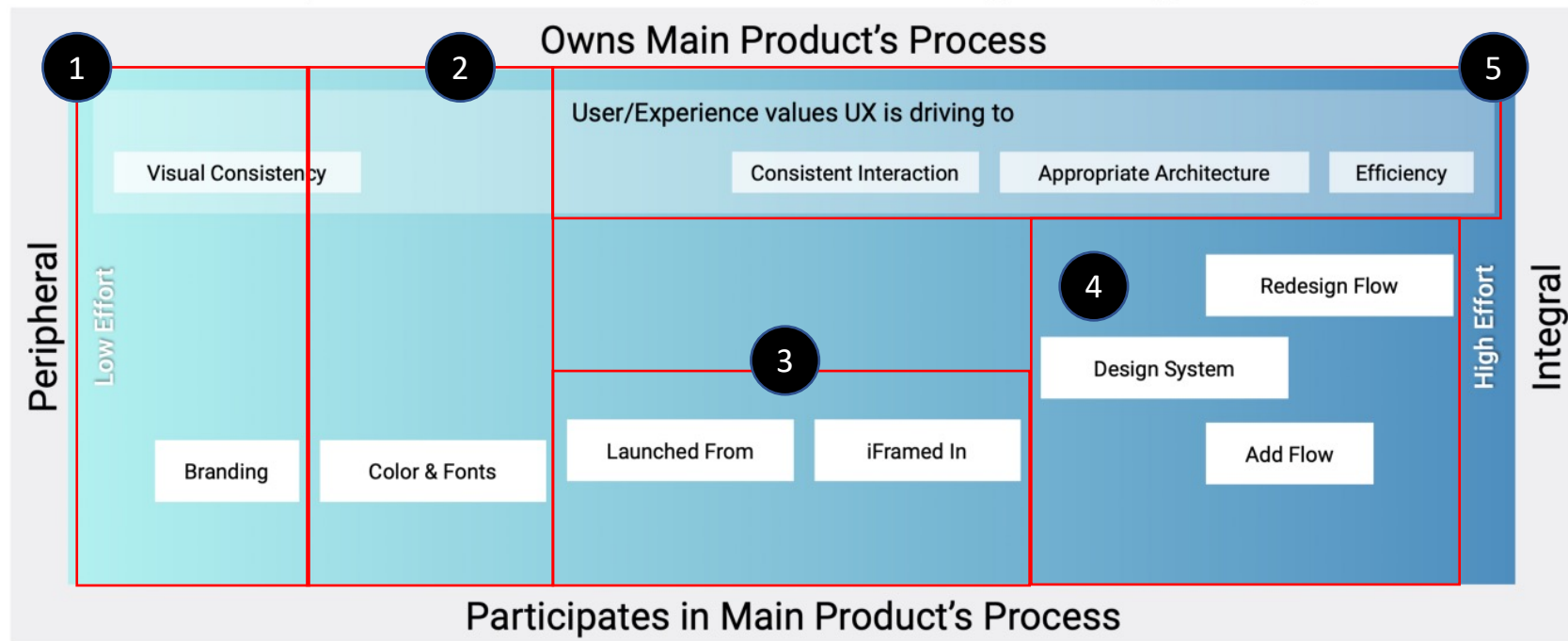
Name	ID	Status	Role	Age	Project	Start Date	Action
John Doe	123456789	Active	Manager	35	Project A	2023-01-01	Edit
Jane Smith	987654321	Inactive	Developer	28	Project B	2023-02-01	Edit
Mike Johnson	567890123	Active	Designer	32	Project C	2023-03-01	Edit
Sarah Brown	345678901	Active	Analyst	30	Project D	2023-04-01	Edit

Data Grids

How are we going to get there?

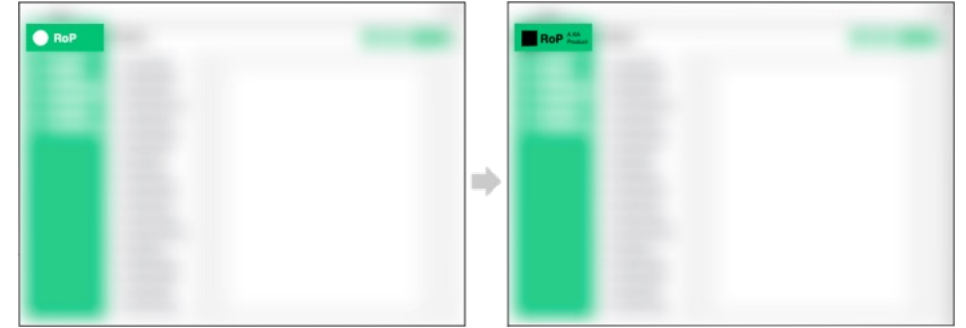
Various UI Integration Options *(from multiple apps/UI > to united apps/UI)*

1. White Label Compliance
2. UI Component Reskinning: SCSS/CSS + HTML edits
3. Hybrid UI
4. Universal UI (common design system and platform)
5. Single Application (universal UI, platform + data/backend integration)



1. White label Compliance

- Where: per standalone app/product*
- Why: to adopt shared white label brand identity
- How: minimal CSS palette updates, with global logo option, for brandable themes
- Implementation: less than 50 lines of CSS, per app, to override system colour defaults.



Effort	Timeline	Cost	Pros	Cons	UX Value
Low	days/weeks	Low	<ol style="list-style-type: none">1. Instant visual recognition of a brand as the owner of the applications, by standardizing master page wrapper colours and logo in all UIs2. Interim solution until common UI framework and design system(e.g. angular & material*) in place	<ol style="list-style-type: none">1. Applications and logins remain separate2. No backend integration or navigable connection between Vizor and v33. Custom white label client CSS - not impacted (to be handled separately)	<p>Low to medium</p> <p>Users will see common branding within their app of choice.</p>

* Sample implementation shown on following slides

1. White Label Compliance - samples

The image displays two versions of the 'Abacus Regulator' web application interface, with red annotations indicating areas for modification.

Left Screenshot (Current State):

- Header:** 'Abacus Regulator main' logo and 'Regulator Developer' user profile.
- Left Sidebar:** A vertical menu with the following items: 'Reporting Status' (selected), 'Scheduled Reporting', 'Expected Submissions ESFS', 'EBA XBRL Reporting' (highlighted in blue), 'EIOPA Pension Funds', 'EIOPA Pension Funds PEPP', 'EIOPA Solvency II', 'Expected Submissions Sample', 'Unexpected Submissions', 'Unscheduled Reporting', 'Reports', and 'Configuration'.
- Main Content Area:** Titled 'EBA XBRL Reporting Expected Submissions'. It features a search bar with 'Filter results...' and a magnifying glass icon. Below is a table with columns: 'Details', 'Reporting Entity ID', 'Reporting Entity Name', 'Reporting Date', 'Submission De...', and 'Submission Tim'. The table contains 15 rows of data for 'Skandinaviska Enskilda ...' with various reporting dates and submission dates. A red arrow points to the table area.
- Bottom:** An 'Upload Report' button.

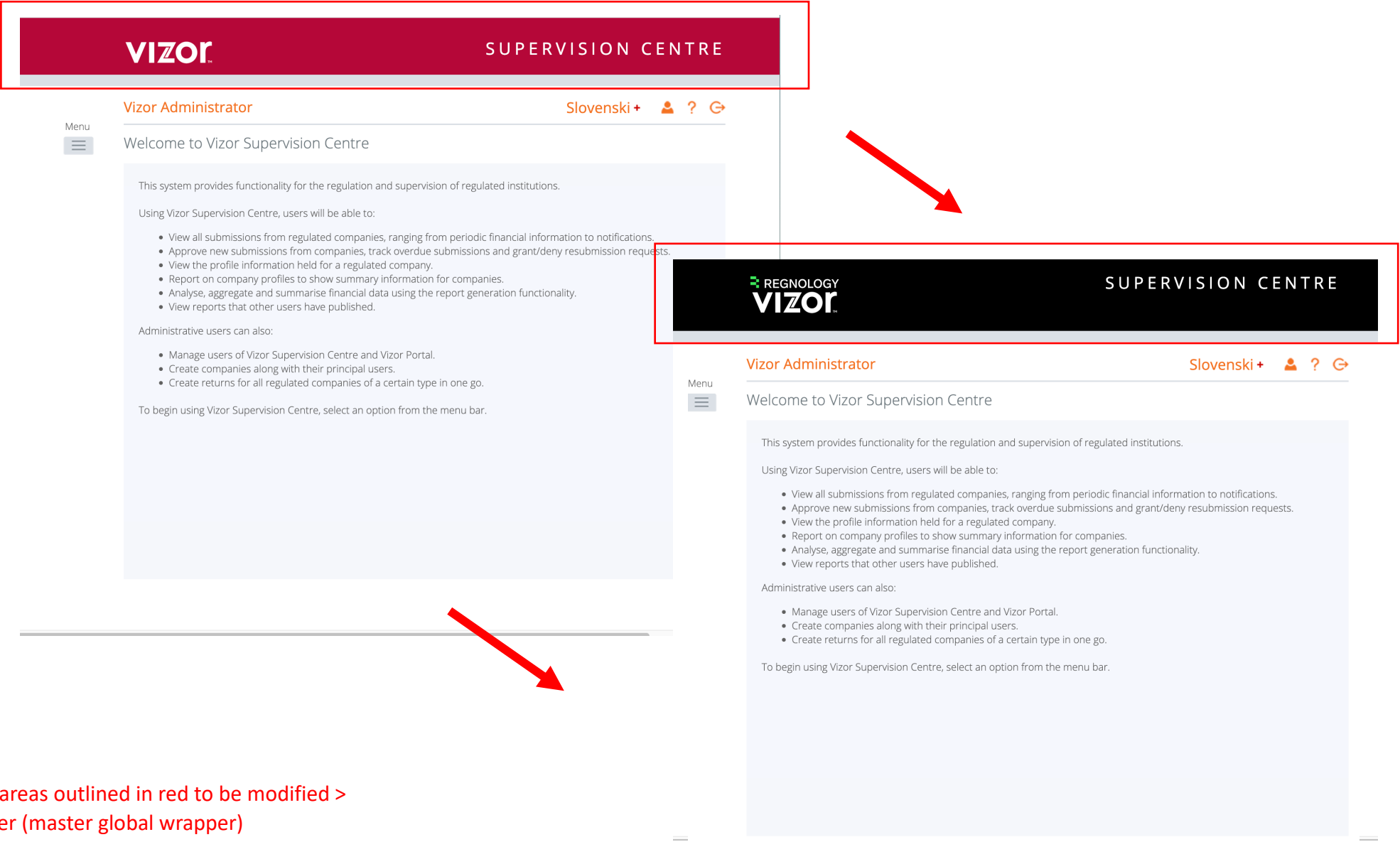
Right Screenshot (Target State):

- Header:** 'Abacus Regulator main' logo and 'Regulator Developer' user profile.
- Left Sidebar:** A vertical menu with the following items: 'Reporting Status' (selected), 'Scheduled Reporting', 'Unscheduled Reporting', 'Reports', and 'Configuration'.
- Main Content Area:** Titled 'EBA XBRL Reporting Expected Submissions'. It features a search bar with 'Filter results...' and a magnifying glass icon. Below is a table with columns: 'Details', 'Reporting Entity ID', 'Reporting Entity Name', 'Reporting Date', 'Submission De...', and 'Submission Tim'. The table contains 15 rows of data for 'Skandinaviska Enskilda ...' with various reporting dates and submission dates. A red arrow points to the table area.
- Bottom:** An 'Upload Report' button.

Red arrows indicate the areas to be modified: the sidebar and the table area.

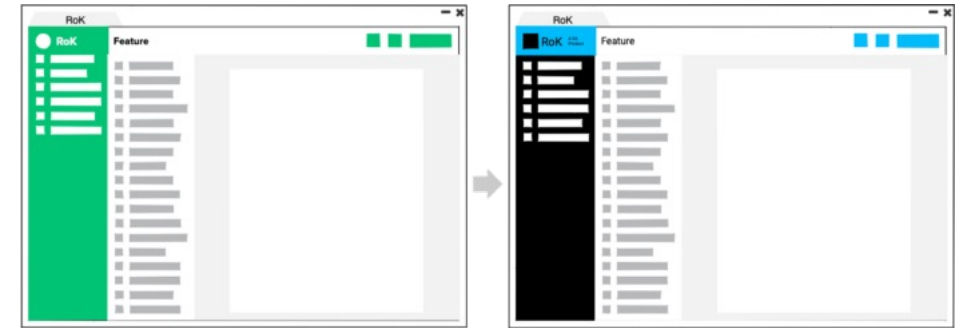
Only areas outlined in red to be modified >
header and sidebar (master global wrapper)

1. White Label Compliance samples



2. UI Component Reskinning – resolve main UI gaps

- Where: per standalone app/product*
- Why: UI design pattern consistency (CSS and HTML changes only, no functional/core changes)
- How: restyle components to align closer (visually) to a unified UI e.g. primary navigation (sidebar); buttons; icons; grid/table



Effort	Timeline	Cost	Pros	Cons	UX Value
Low to medium	Weeks to months <i>(UI testing would take most effort)</i>	Low	<ol style="list-style-type: none">1. Further visual recognition of branding2. Closer visual alignment between separate app UIs, until all apps on Angular	<ol style="list-style-type: none">1. Applications and logins remain separate2. No backend integration between Vizor and v33. Whilst UI components will look similar, they may behave differently (vizor v. V3),	Low to medium Users will see common branding and closer alignment between the UIs in each app.

* Sample implementation shown on following slides

2. UI reskinning

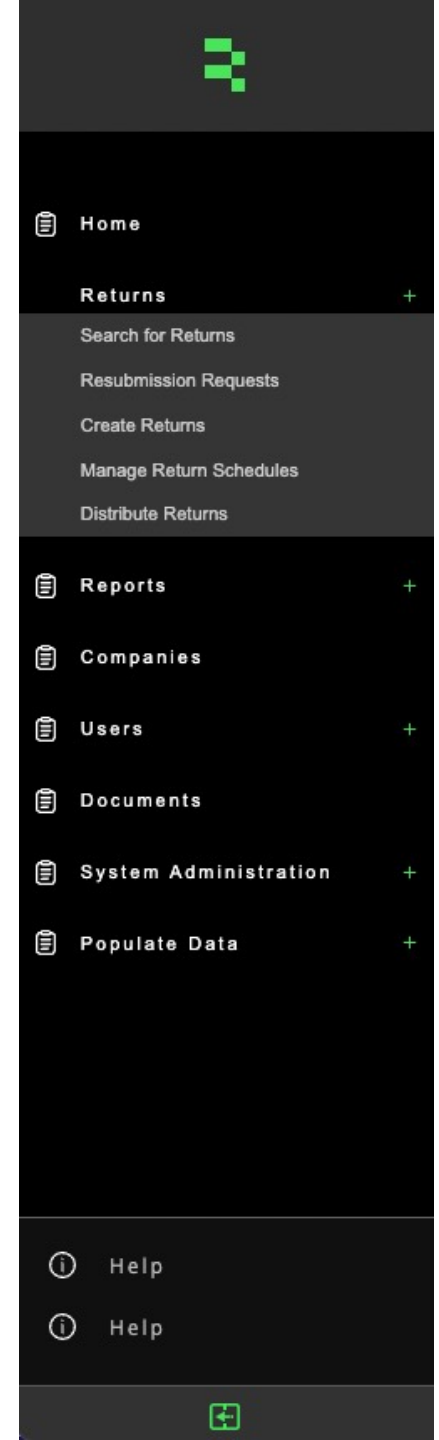
Example

We restyled an existing menu component in Vizor apps using global CSS overrides, and some minor HTML edits. We then restyled the menu container to function like a full-height sidebar, always visible.

On the far right of screen is the result of approx. 50 lines of CSS and less than 5 lines of HTML modified or added.

Benefits:

- Core Component changes are reduced to bare minimum.
- Restyling existing components provides significant UI alignment, for little effort and cost.



3. Hybrid Architecture – Cross navigation, new User Flows and Mapping

- Where: per standalone app/product*
- Why: to revise global navigation, to facilitate logical cross-navigation between vizor and Regnology apps
- How: average of 30+ navigation pathways = 100+ potential User flows to be reviewed per application
 1. where does commonality exist/overlap (functional use cases/service or task)
 2. Detailed research of existing navigation & user flows, end-to-end tasks e.g. ‘task analysis’
 3. Align all terminology and navigation taxonomy between applications, via the **approved joint glossary**.
 4. Research will reveal how to optimise user journeys, via cross-navigation between vizor and v3, web API or consolidated site navigation/shared sitemap e.g. iframe embedding or enhanced user pathways.

Effort	Timeline	Cost	Pros	Cons	UX Value
Medium to High	Months to years <i>(research takes time and effort - tbd for every standalone app/feature/service)</i>	Low to Medium	<ol style="list-style-type: none">1. Can support singular/global navigation between vizor and v3 – covering backwards navigation2. Design system and UX analysis dictates the Global UI Wrapper3. It will feel like a single (integrated) application on frontend	<ol style="list-style-type: none">1. Applications and logins remain separate<ul style="list-style-type: none">• Universal login would be preferential at this stage, but not absolute requirement2. No backend integration between Vizor and v3 – APIs used instead3. Users may be presented with “new” ways of performing “old” tasks, and this needs to be communicated/documented appropriately4. Dev effort substantial in all apps	High

* Sample implementation shown on following slides

To get there, will involve a lot of this...

Task analysis



Identify the Tasks

In a case study, refer to your data and understand the tasks.



Break Down the Tasks

For your analysis report, break down the main tasks into its subtasks for a specification.



Identify Steps in Subtasks

Draw a layered task diagram and ensure that the task is completed.



Proofread Your Task Analysis

Proofread your overall document over and over again and point some errors if discovered.

How to Write a

Task Analysis

According to experts, a task is your goal that needs to be achieved. In the business process, it must be recorded in order for you to track your business's status.



Hybrid navigation...?

3. Hybrid Architecture – sample existing sitemaps

V3 Sample Sitemap from:

<https://main.frontend.dev.regulator.regnology.io/>

Out of

44

items on the navigation tree,

24

link within the application

9

link to external URLs with no way back!!

1	Reporting Status (not a page, only a nav parent)
2	- Scheduled Reporting (not a page, only a nav parent)
3	- Expected Submissions ESFS (not a page, only a nav parent)
4	- EBA XBRAL Reporting
5	- EIOPA Pension Funds
6	- EIOPA Pension Funds PEPP
7	- EIOPA Solvency II
8	- Expected Submissions Sample (not a page, only a nav parent)
9	- Common Data Dictionary PoC
10	- Sample JSon Template
11	- Sample XML
12	- Unexpected Submissions
13	- Unscheduled Reporting (not a page, only a nav parent)
14	- EIOPA Solvency II ORSA
15	- EIOPA Solvency II RSR
16	- Insider Trading
17	- National Taxonomies
18	Reports
19	- ESFS (not a page, only a nav parent)
20	- EBA XBRAL Reporting
21	- EIOPA Pension Funds
22	- EIOPA Pension Funds PEPP
23	- EIOPA Solvency II
24	- Sample (not a page, only a nav parent)
25	- Common Data Dictionary PoC
26	- Sample JSon Template
27	- Sample XML
28	Analytics (not a page, only a nav parent)
29	- Analytics (goes to an external SuperSet Open source dashboard URL https://analytics.)
30	Configuration (not a page, only a nav parent)
31	- content development kit (goes to an external GitLab Login page)
32	- pything API reference (brings users out of the main global wrapper – no way back!!)
33	- data collections
34	- public holidays
35	- reporting modules
36	- supervised entities
37	- waivers
38	- testing dashboard (brings users out of the main global wrapper – no way back!!)
39	- Jupyter Notebook (brings users out of the main global wrapper – no way back!!)
40	- schema generator (brings users out of the main global wrapper – no way back!!)
41	Administration (not a page, only a nav parent)
42	- logs & metrics (goes to an external grafana page)
43	- user management (goes to an external keycloak page)
44	- workflow management (goes to an external camunda page)

3. Hybrid Architecture – sample existing sitemaps

Vizor Supervisor Centre Sample Sitemap from:

<http://vmtech-vsc/VizorSupervisionCentre>

Out of

30

items on the navigation tree,

24

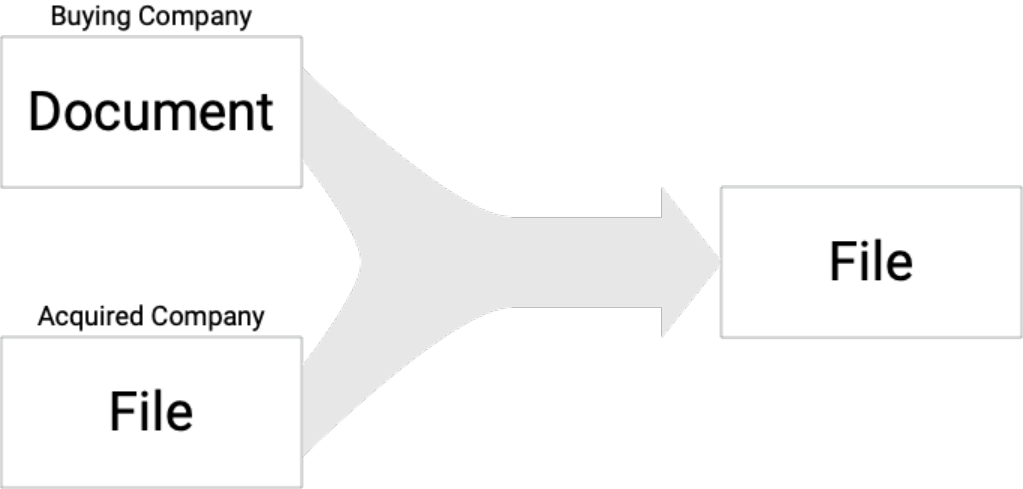
link within the application

1	Home
2	Returns (not a page, only a nav parent)
3	– Search for Returns
4	– Resubmission Requests
5	– Create Returns
6	– Manage Return Schedules
7	– Dsitribute Return
8	Reports (not a page, only a nav parent)
9	– Aggregate Reports
10	– Standard & Ad-hoc Queries
11	Companies
12	Users (not a page, only a nav parent)
13	– manage Portal users
14	– manage Internal Users
15	Documents
16	System Administration (not a page, only a nav parent)
17	– Manage Company Groups
18	– Send Notification
19	– Import Comments
20	– Manage Rules
21	– Release Management
22	– Configuration Management (not a page, only a nav parent)
23	– manage website content
24	– manage workflow schedules
25	– System Settings
26	– Manager Document Roles
27	Populate Data (not a page, only a nav parent)
28	– seed data
29	– company profile
30	– load metadata

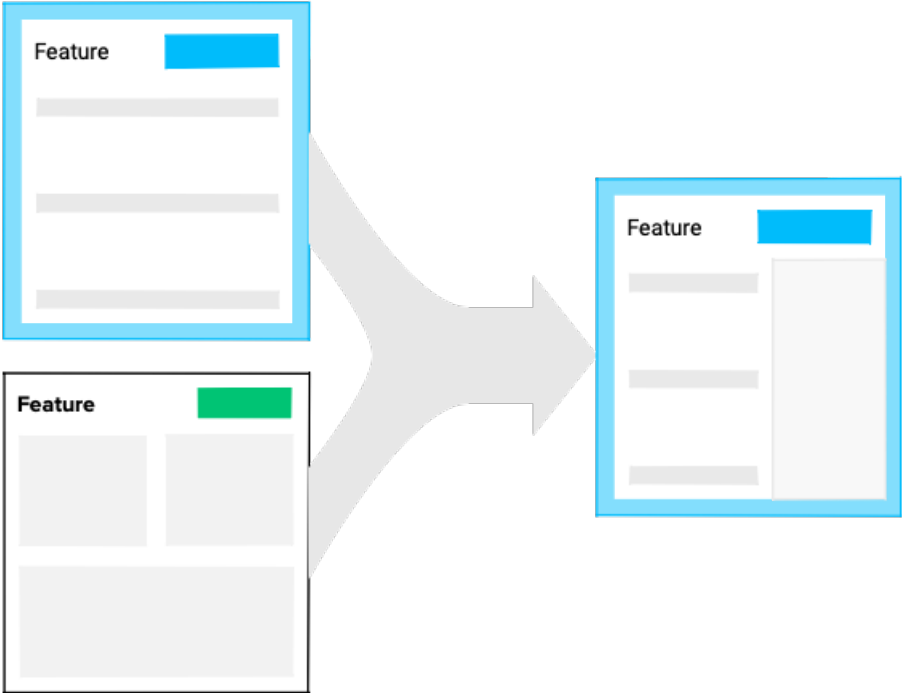
How to merge features and language...?

Hybrid Integration example:

Language Harmonization/Standard Glossary



Feature/Function Alignment



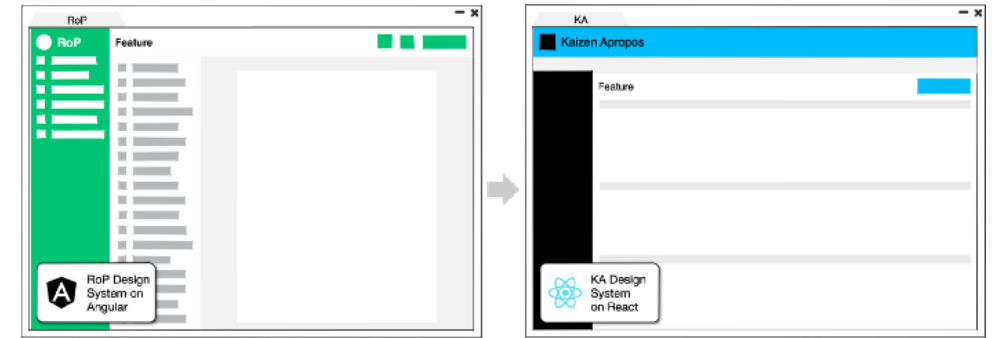
4. Universal UI– Universal UI (common design system and framework)

Current status:

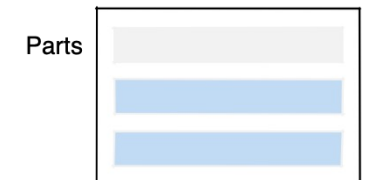
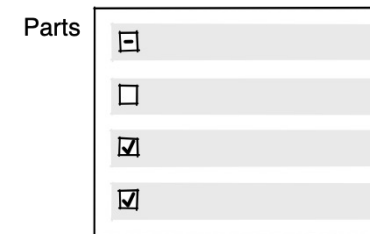
- V3/regnology UI Framework = Angular & Material UI
- Vizor UI Framework = .Net, webforms, native HTML components + DevExpress UI Suite

Proposed Status:

- V3/Regnology UI Framework = Angular & Material UI Toolkit + new design system
- Vizor UI Framework = Angular & PrimeNG UI Kit + new design system

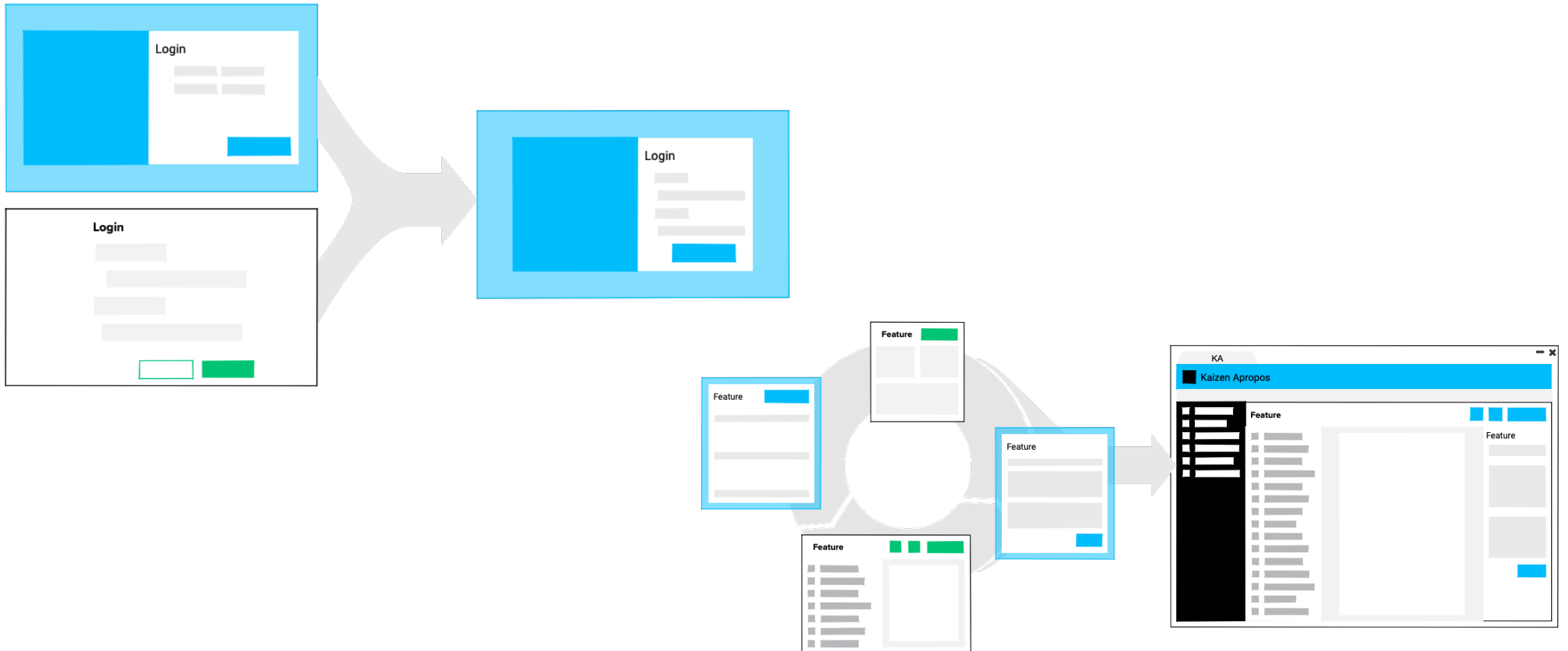


Behaviour, Interaction and Module Alignment



5. Singular Application – full integration

- Backend and frontend integration with universal login across a united application UI

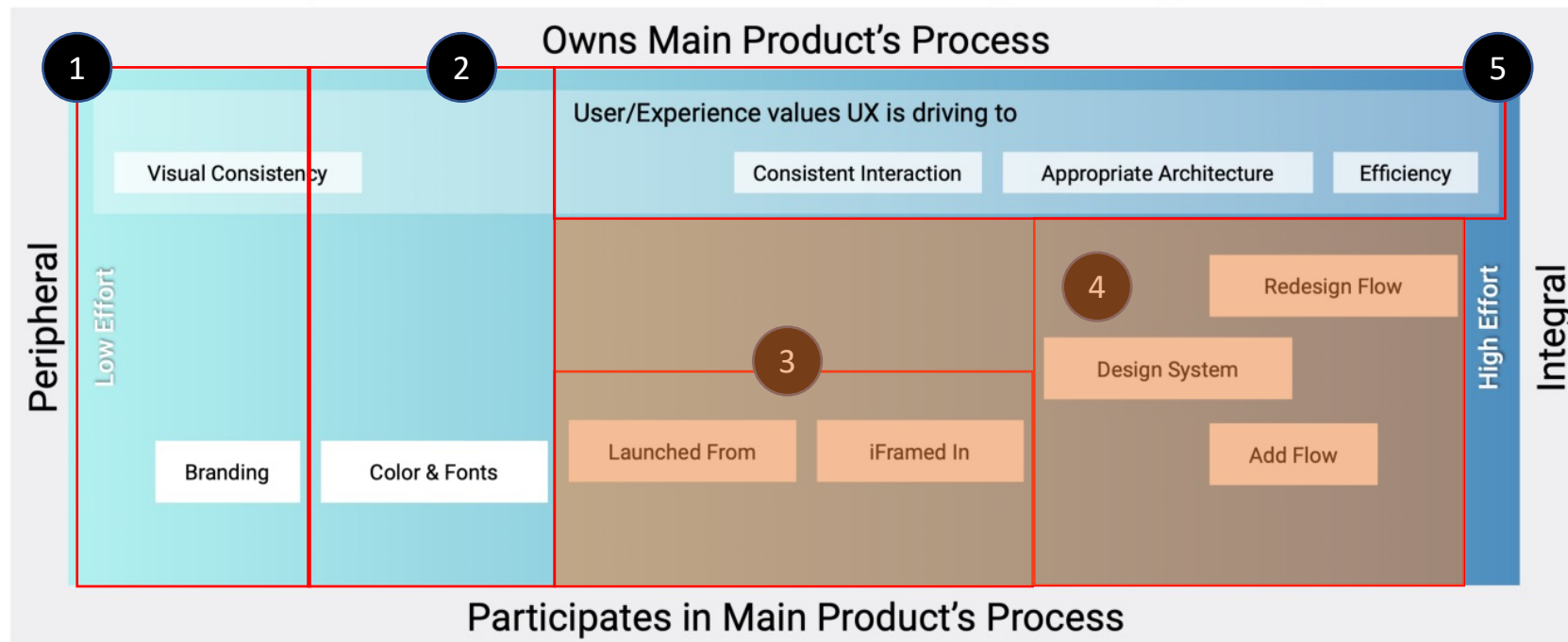


Which Integration options makes sense?

Various UI Integration Options *(from multiple apps/UI > to united apps/UI)*

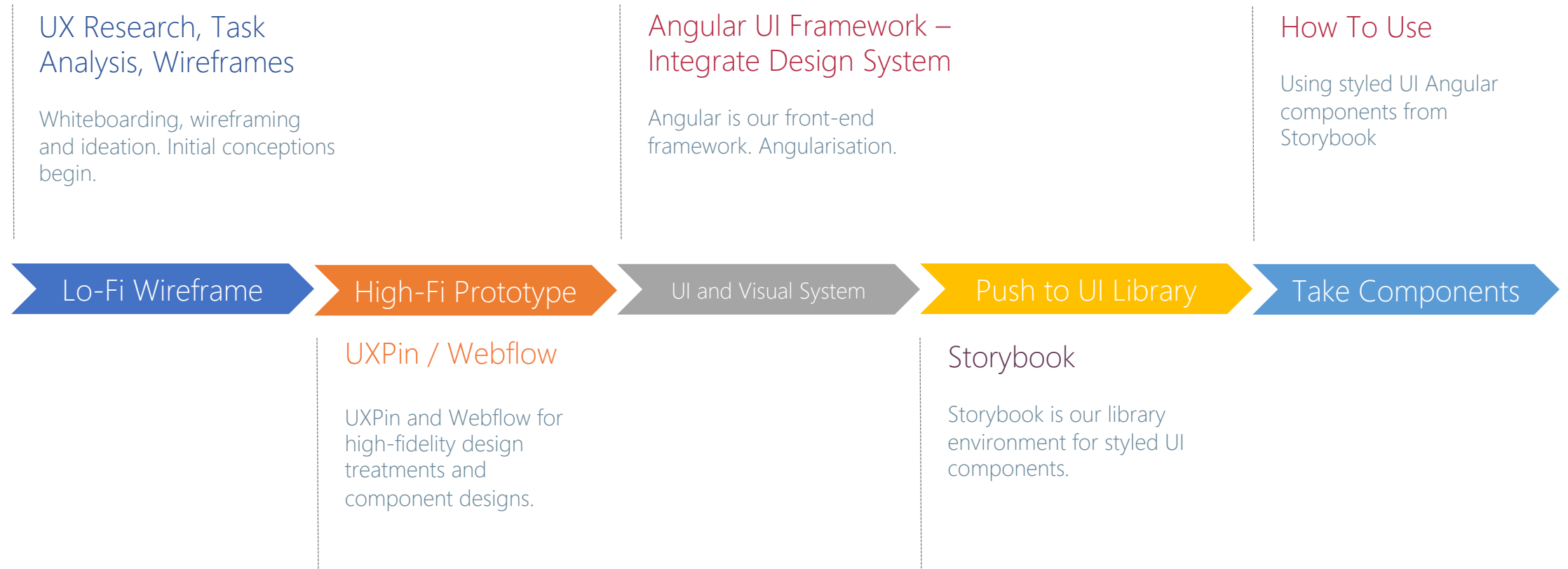
1. White Label Compliance
2. UI Component Reskinning: SCSS/CSS + HTML edits
3. Hybrid UI
4. Universal UI (common design system and platform)
5. Single Application (universal UI, platform + data/backend integration)

Optional White labelling and reskinning can happen anytime, as part of interim rebranding measures



The Journey Ahead...

For Hybrid and Universal UI: Design System

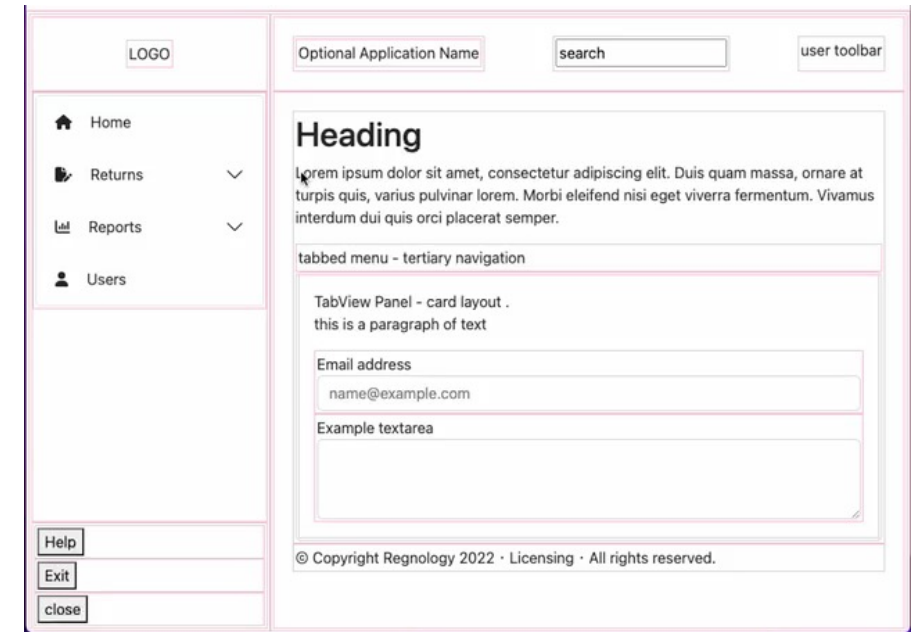
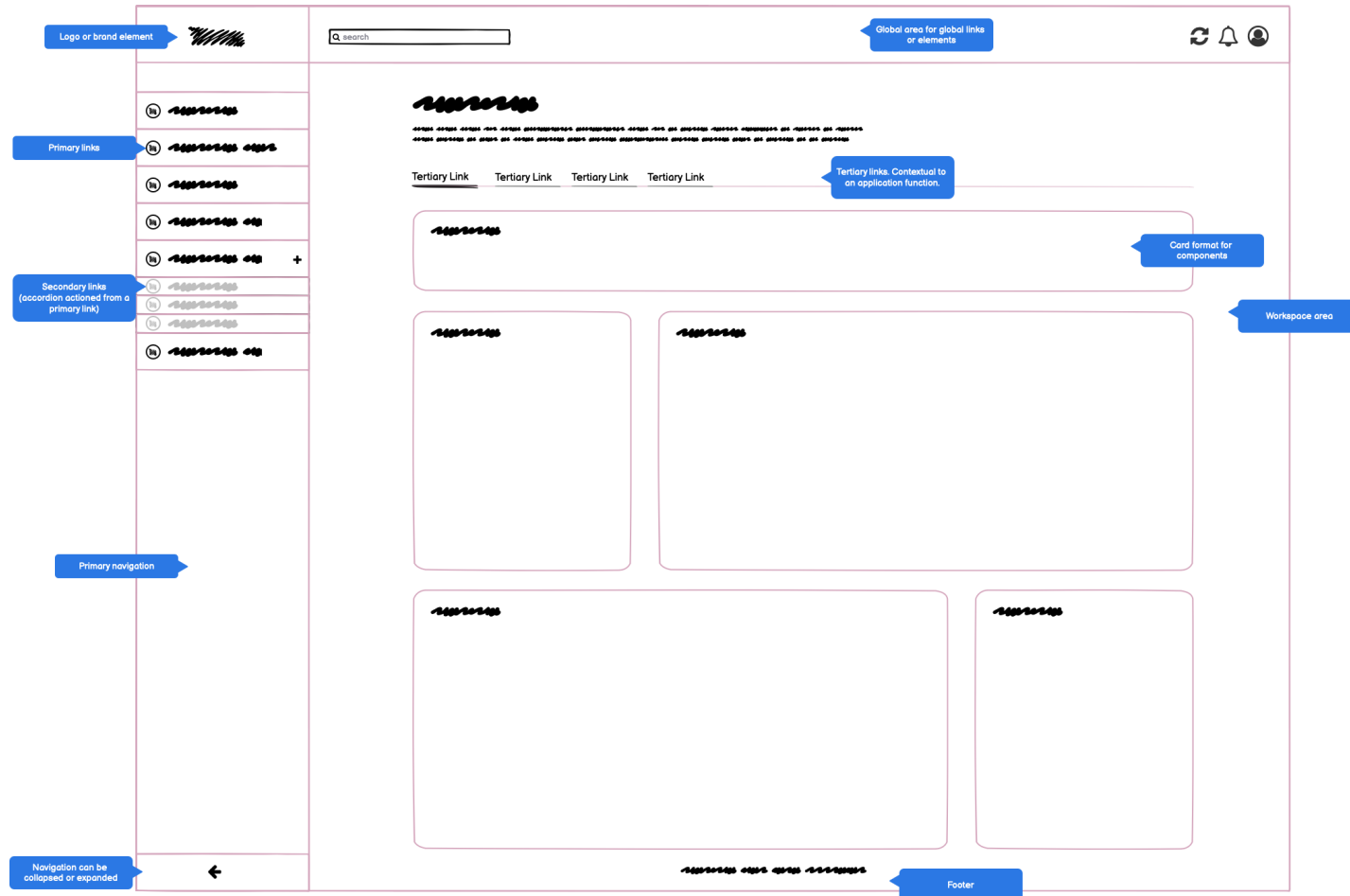


Design System Tokens: interchangeable and scalable.

These can be modified to suit the needs of any modular, modern Design System

Category	Sub-Category	Elements	Description
Design Language			
	Brand	Branding	Logo and white label support
	Guidelines	Terminology	standard terms and unified voice
		Accessibility	how colour, hierarchy, and assistive technology is to be implemented to help 'all' users
		Microcopy	content and language guides to consider global terms and standard labelling of components
Design Tokens	Colour	colour variables	functional global colours, based on the brand, and how theming can be supported for client branding
			colours for all states to be defined e.g. backgrounds, hover, disabled stats
		accessibility	ensure AA contrast ratios pass basic tests, and ensure it is maintained
		dark mode/light mode	many users expect this option for applications they spend a lot of time in, worth consideration
	Layout	units	define global measurement units e.g. REM, em, % or pix
		grid	define if 12 col or 16 col layout to be supported - flexbox or css grid - and their relative Gutter/margins
		breakpoints/media queries	define adaption breakpoints for tablet, laptop, mobile etc.
		spacing	padding and margin guidelines for global elements and layouts
		component elevation	define z-index rules for layered features
	Typography		
		units	adaptive or responsive/scaled e.g. responsive = vw (1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is 0.5cm.); adaptive = media queries, change based on breakpoints
			relative or absolute units e.g. if basefont is 16px, font-size:50% will be 8px
			zooming support - how do you want the text to enlarge/decrease based on browser zoom levels
		readability	letter spacing, line height, max length for certain components e.g. Buttons shouldn't have more than 100 characters
		font faces	performance, fallbacks, Mac v PC
		hierarchy	typographic scales
	Icons	Style	for standalone icons, and icons within buttons(needs to support the same states as button)
		naming convention/classes	
		relative scale within and outside components	
		sizes	e.g smal, medium, large, auto
		rules	icons to always use/associate with global features/tasks, to improve familiarity
		guidelines	how to use them, where, and why
	Mixins & variables >	Utils and Namespacing	can come from a UI Toolkit e.g. Bootstrap, Material, PrimeNG or a library like Tailwind, or define custom as needed.

Lo-Fi Wireframing



Hi-Fi Prototypes and Visual Design Artefacts

Design will provide HTML and CSS specification for each element; padding, size, color, behaviour etc. for an Angular developer to create components to commit to our UI library, Storybook.

[Click here](#)

Tables

Active Cases

View all active EOIR cases

Current Cases

Historical Cases

Name ▾

ID ▾

Country ▾

Phase ▾

Days Left ▾

Progress ▾

Due Date ▾

Actions

GRF_114_3 VAL CS11721 ↗

124536845136

USA

Preparation

10

31.12.2021

GRF_114_3 VAL CS11721 ↗

124368452136

Germany

Pending

2

31.12.2021

GRF_114_3 VAL CS11721 ↗

125368452136

Spain

Complete

0

31.12.2021

GRF_114_3 VAL CS11721 ↗

125368452136

USA

Complete

0

31.12.2021

Showing 1 to 10 of 200 entries

<<

<

1

2

3

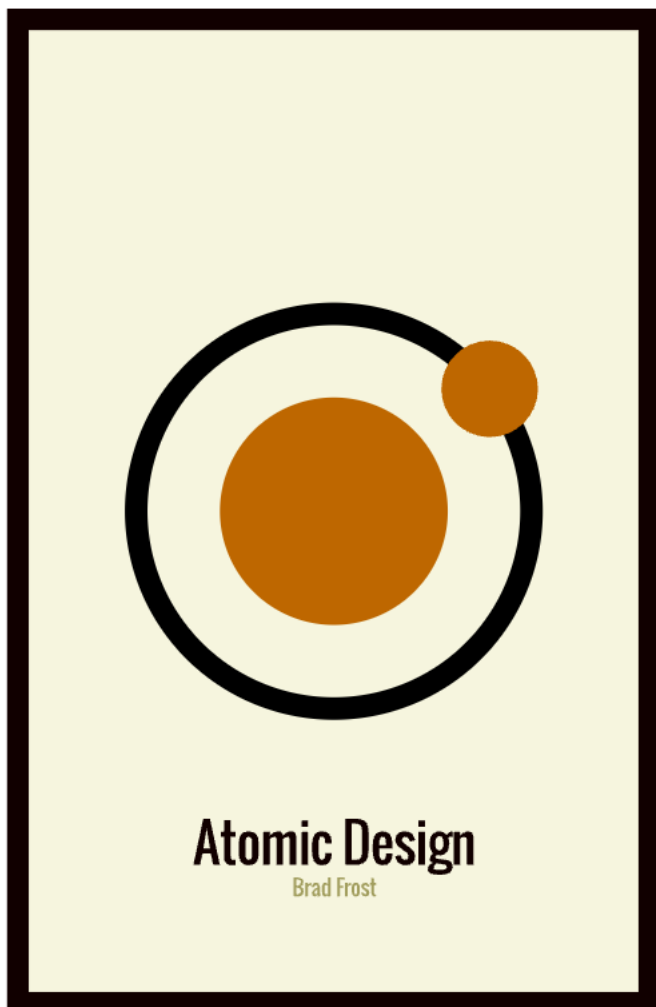
4

5

>

>>

10 ▾



Atomic Design by Brad Frost

We're tasked with making interfaces for more users in more contexts using more browsers on more devices with more screen sizes and more capabilities than ever before. That's a daunting task indeed. Thankfully, design systems are here to help.

Atomic Design details all that goes into creating and maintaining robust design systems, allowing you to roll out higher quality, more consistent UIs faster than ever before. This book introduces a methodology for thinking of our UIs as thoughtful hierarchies, discusses the qualities of effective pattern libraries, and showcases techniques to transform your team's design and development workflow.

order the book

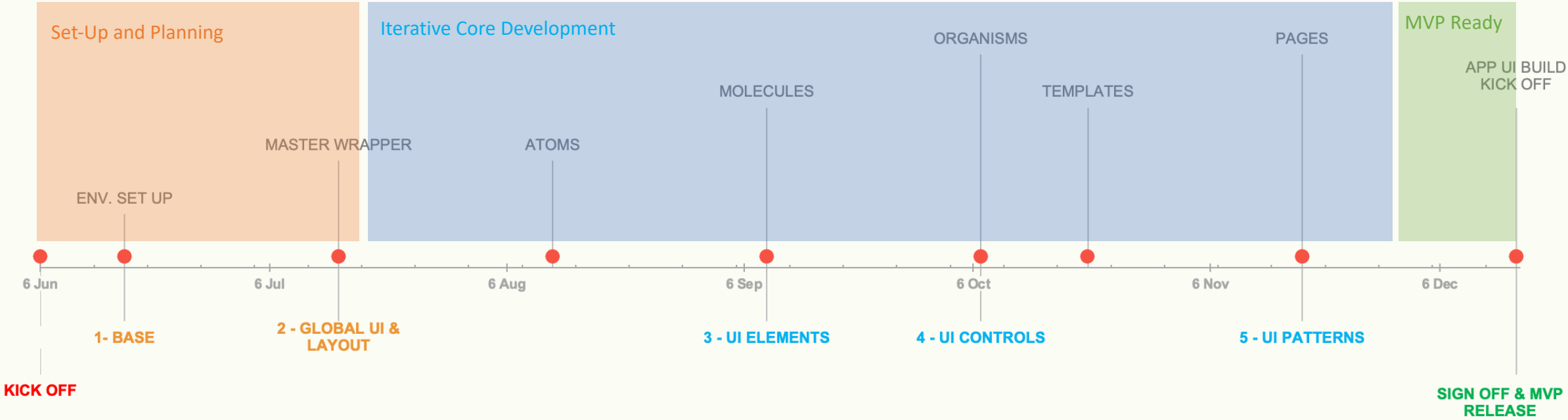
read now

Foreword by Josh Clark and Dan Mall

- 1. Designing Systems**
Create design systems, not pages
- 2. Atomic Design Methodology**
Atoms, molecules, organisms, templates, and pages
- 3. Tools of the Trade**
Pattern Lab and the qualities of effective style guides
- 4. The Atomic Workflow**
People, process, and making design systems happen
- 5. Maintaining Design Systems**
Making design systems stand the test of time

<https://atomicdesign.bradfrost.com/table-of-contents/>

Angular UI Framework and Design System milestones to be in Sync – *note: dates subject to change*



Week #	Sprint #	Sprint Schedule - 2 week duration	
		Start	End
23	1	June 6, 2022	June 17, 2022
25	2	June 20, 2022	July 1, 2022
27	3	July 4, 2022	July 15, 2022
29	4	July 18, 2022	July 29, 2022
31	5	August 1, 2022	August 12, 2022
33	6	August 15, 2022	August 26, 2022
35	7	August 29, 2022	September 9, 2022
37	8	September 12, 2022	September 23, 2022
39	9	September 26, 2022	October 7, 2022
41	10	October 10, 2022	October 21, 2022
43	11	October 24, 2022	November 4, 2022
45	12	November 7, 2022	November 18, 2022
47	13	November 21, 2022	December 2, 2022
49	14	December 5, 2022	December 16, 2022
51	15	December 19, 2022	December 30, 2022

3 sprints

9 sprints

3 sprints

1. Angular project and Storybook set-up & prep

2. Design system foundation (colours, typography, icons, grid layout)

3. Master Global Wrapper

1. Develop Atoms: buttons, inputs, alerts, etc.

2. Develop Molecules: tabsets, typeaheads, dropdowns

3. Develop Organisms: table/grid, navigation

4. Unit test and document as we go

5. Templates begin once WFs received from UX design

6. Pages begin once WFs received from UX design

1. Sprint cycle and demos, for internal review and sign-off

2. MVP of Release V1.0 > App UI Build kick-off

11 Epics

with approx

100 user stories

Angular UI Book of Work

EPIC List	
Name	Core Stories - related tasks tbd
<i>approx total man days of effort</i>	<i>click + symbol in left margin to expand lists</i>
1. Dev Environment Set Up	<i>All the below tbc with Sen. Dev Lead, Ionut</i>
2. Design System Set Up	https://preview.uxpin.com/bfdd070bc396237462ffe898a3bb9d9a91193738#/pages/139161723/specification/sitemap?mode=edit
3. Master Global Wrapper - App-Root	<i>All the below to be scalable, themeable global layout components</i>
4. Documentation. QA and Unit Tests	
5. Atom components	
6. Molecule components	
7. Organisms + 3rd Party Components	
8. Templates & UI Patterns	<i>All the below</i>
9. Pages	<i>All the below</i>
10. Beta Internal Review & Test	<i>items tbd</i>
11. MVP 1.0 release	<i>UI framework</i>

Parent Component	Variants	Label	Templates	Style Options	Size Options	Bootstrap Version	PrimeNG Version	In StoryBook?	Refactor required for StoryBook?	States to support globally
Alerts - notifications						https://getbootstrap.com/docs/5.2/components/alerts/	https://www.primefaces.org/primeng/messages	no		
Avatar							https://www.primefaces.org/primeng/avatar	no		
Badge	icon	no		bgcolor, pill or square edge	small, default, large	https://getbootstrap.com/docs/5.2/components/badge/	https://www.primefaces.org/primeng/badge	http://vm-hypernova01-9000/?path=/story/vizor-components-badge-count-count	yes	n/a
	text or number	yes		bgcolor, pill or square edge	small, default, large					
Breadcrumb	default	no	tbd by UX design	dividers/separators	n/a	https://getbootstrap.com/docs/5.2/components/breadcrumb/	https://www.primefaces.org/primeng/breadcrumb	no		n/a
	show only previous path	no	tbd by UX design	dividers/separators						
Buttons						https://getbootstrap.com/docs/5.2/components/buttons/	https://www.primefaces.org/primeng/button	http://vm-hypernova01-9000/?path=/story/vizor-components-button-default-default		
	default	yes		width (full or auto), outlined	small, default, large			yes		enabled, disabled, hover, focused, pressed/active
	primary	yes		width (full or auto), outlined	small, default, large			yes		enabled, disabled, hover, focused, pressed/active
	secondary	yes		width (full or auto), outlined	small, default, large			yes		enabled, disabled, hover, focused, pressed/active
	tertiary	yes	label + icon	width (full or auto), outlined	small, default, large			no		enabled, disabled, hover, focused, pressed/active
	text/link	yes			small, default, large			yes	is under Link, move under 'Button'	enabled, disabled, hover, focused, pressed/active
	variant + badge		label + badge	badge position + badge styles	auto scale when nested			no		enabled, disabled, hover, focused, pressed/active
	variant + icon		icon only	icon position	auto scale when nested			yes	but not working/set up properly	enabled, disabled, hover, focused, pressed/active

Epics and Backlog Stories added to Angular Squad Jira

Angular is a frontend framework, a platform and a framework for building applications using HTML and TypeScript. It implements core and optional functionality as a set of TypeScript libraries that are imported into the applications. It seamlessly synchronizes the data between Model and View.

[Click here](#)

The screenshot shows the Jira interface for the 'Technology' project. The left sidebar contains navigation options under 'PLANNING' (Board, Roadmap, Backlog, Active sprints, Reports) and 'DEVELOPMENT' (Issues, Components, Releases, Project pages, Checklist, Timesheet, Add-ons, Add shortcut, Project settings). The main area is titled 'Backlog' and shows a search bar, filters, and a list of epics and issues. The 'Angular UI' epic is expanded, showing 10 issues, 0 completed, 6 unestimated, and 16 estimate. The backlog shows 0 of 60 issues visible.

Projects / Technology / Hypernova

Backlog

Search backlog

Angular UI ON HOLD Sprint Ready Grooming Ready Grooming Blocked Remove or Keep Unassigned Recd

Quick filters 1 Clear all

EPICS

All issues

VERSIONS

[INT-HN] 1. Angular UI - Dev Set Up

TECH-32794 [INT-HN] 1. Angular UI - Dev Set Up

Issues 10

Completed 0

Unestimated 6

Estimate 16

Create issue in epic

[INT-HN] 2. Angular UI - Design System Set Up

[INT-HN] 3. Angular UI - Master UI wrapper

[INT-HN] 1. Angular App

HN22-12 4 issues

HN - Ready for Sprint 0 issues

HN - Ready for Grooming 5 issues

HN - Grooming Blocked 0 issues

HN - Angular items for review 1 of 6 issues visible Clear all filters

HN - Archive, remove or keep 0 issues

Backlog 0 of 60 issues visible Clear all filters

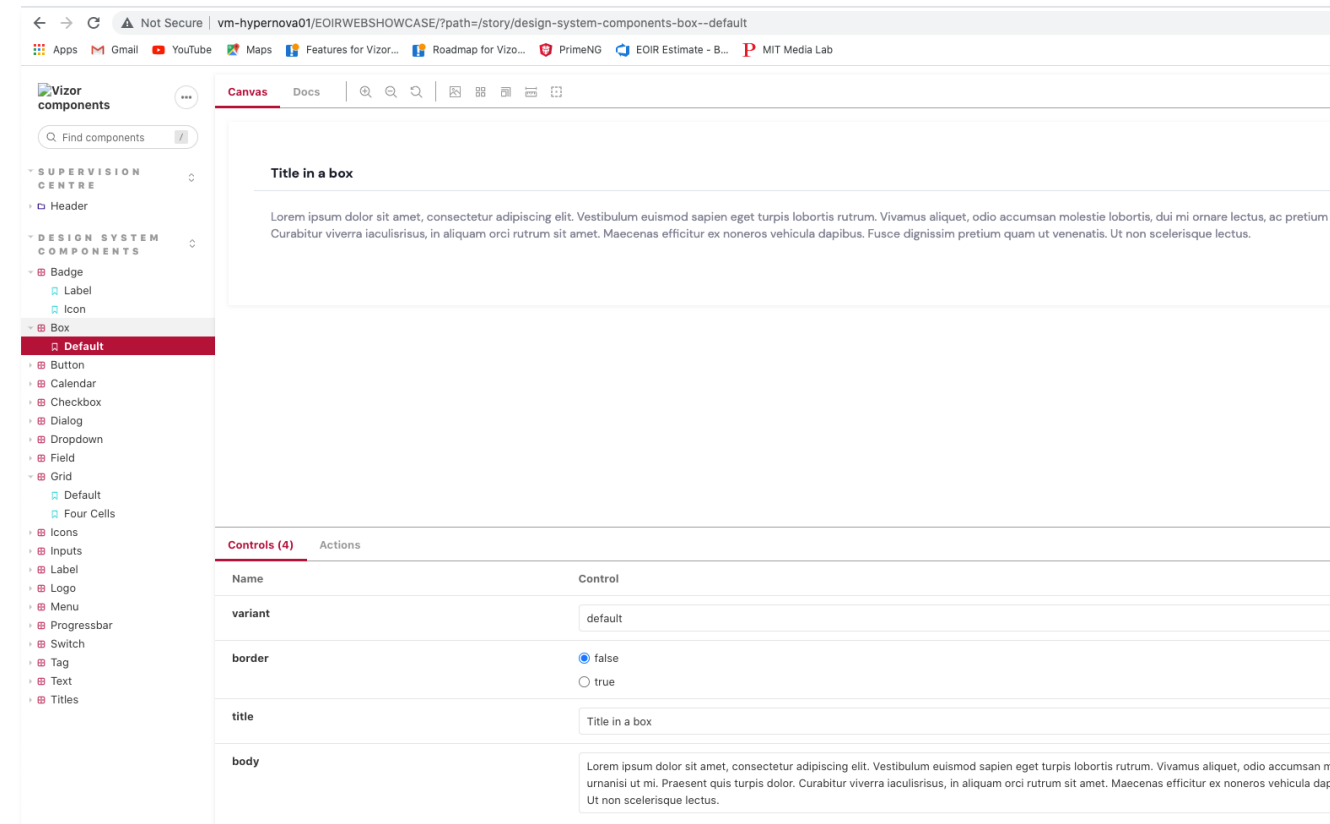
There are no issues that match your cu

+ Create issue

Storybook

Storybook is an open source tool for building UI components and pages in isolation, a development environment tool, a library environment for UI components. It allows us, the designers and developers, to create and test components in isolation. It runs outside of the app, project dependencies don't affect the behaviour of components.

- Library environment for UI components
- Ubiquitous language about our components
- Automated isolated components with mock data
- Allows for sharing documentation.



Storyboard

