



DESIGN SYSTEMS

PRESENTED BY

UiRevolution & The Goat Farm



TABLE OF CONTENTS

DOCUMENT CONTROLS	4
EXECUTIVE SUMMARY	5
BACKGROUND	6
BUSINESS CASE	8
EDISON	12
THE SCOPE	13
METHODOLOGY	14
THE TOOLS	17
CREATING COMPONENTS	21
UPDATING COMPONENTS	26
CONTENT, ASSETS & COPY	28
EXCLUSIONS	29
RISKS	30
RECOMMENDATIONS	31
APPENDIX	32
REFERENCES	35

DOCUMENT CONTROLS

AUTHORS

Name	Role	Organisation
Jordan Gardiner	Design Lead	UiRevolution
Jayson Bell	Lead User Experience Designer	Vector
Srinivasa Munagala	Solutions Architect	UiRevolution
Mark Michel	Business Director	UiRevolution
Vaughn Davis	Creative Director	The Goat Farm
Brani Mead	Business Analyst	UiRevolution

CONTRIBUTORS

Name	Role	Organisation
Martin Uhe	Automation SME	Vector
Anne Bilek	Front-end Practice Lead	ClearPoint
Jannis Gundermann	Web Developer	ClearPoint
Sam Lake	Senior Digital Platform Manager	Vector
Robert Sang	Lead Digital Architect	Vector
Gerardo Darcy	Lead Solution Architect	Vector

EXECUTIVE SUMMARY

Vector is undertaking a CMS migration of its websites; from Kentico to Kentico Cloud. This migration is under way as Vector replaces and consolidates its websites. This move requires a new design methodology to work alongside this new platform and workflow. Just as the move to a centralised CMS presents Vector and the Vector businesses with increased efficiencies, so too does the promise of an accompanying design workflow.

Design Systems present an ecosystem that thousands of leading companies are adopting for scale, efficiency, and consistency. This is a methodology that is being adopted globally by many of the world's leading brands and enterprises. There is a strong business case for this approach and significant resources are actively shared and available to help businesses on this journey.

Essentially, a Design System means that elements that are used by more than one website or document are held and managed centrally, rather than in multiple copies. A Design System is a dynamic resource that a website "pulls" elements from – so changing a logo within the Design System, for example, will mean it's automatically changed on all the web pages that draw upon the Design System.

Working with our design team and developers both internally and with Vector's partner vendors, we embarked upon and have created a Proof of Concept (POC). This POC originally centred

around a website design and development task as a working example, (creating a new button). In parallel, we have defined a suite of tools and documented a prototype workflow.

During this process we saw the potential for the Design Systems approach to benefit Vector beyond the immediate website requirement. We can see significant opportunities for this system to de-silo a raft of assets and processes, with a consolidation of brand assets, communications, content and other potential collateral.

As Vector looks to realise its ambition to remain at the forefront of new technology solutions, new challenges to scale and the need to increase efficiencies will become increasingly important. A Vector Design System will meet this challenge and guarantee increased control, consistency, visibility and efficiency across communications, design, content and development. This POC and accompanying workflow gives us a solid foundation to achieve this.

The opportunity to de-silo operations, centralise key assets and support Vector's content life cycle is significant. We look forward to engaging with the Comms, Brand and Content teams to help drive this innovation, explore how this Design System can serve existing workflows and become the central repository that allows Vector digital and brand platforms to scale into the future.

View the Design System POC:

<https://zeroheight.com/8fc6afdc4>

BACKGROUND

Vector are undertaking a migration from their existing CMS, Kentico to Kentico Cloud. This ensuring flexibility and scalability for vector.co.nz and in turn across their other properties.

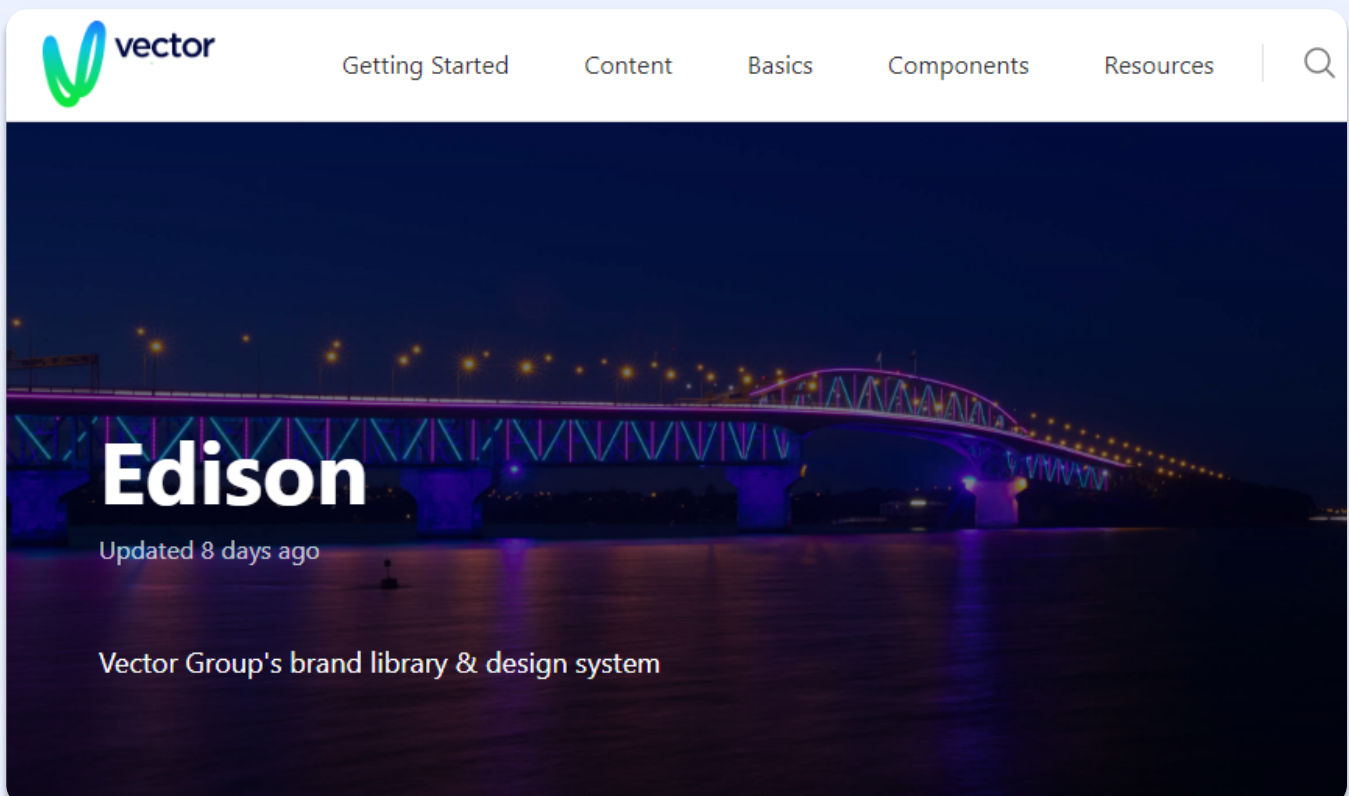
As Vector transitions to Kentico Cloud for its primary CMS there is a need to establish the best front-end platform, systems, process and practices. The significant opportunity evident here is to design this new infrastructure and workflow in a way that builds efficiency, control and the ability to scale.

Vector has an array of digital properties that has been expanding as other business units have been added to the family. This presents

an opportunity to bring brand assets, guidelines and content in line across the organisation.

The opportunity needs to be tethered to a roadmap that works in the short term as vector.co.nz is transitioned to Kentico Cloud. It also allows Vector to use this project as a test case for the wider strategic piece of work to encompass the Vector family of businesses.

As this work is undertaken, a crucial factor is our ability to empower Vector stakeholders with the tools to take ownership of this improved environment and streamlined workflow.



WHAT IS A DESIGN SYSTEM

Although there are varying levels of detail a Design System might have; the core of every Design System is a set of guidelines, rules and documentation on how to create digital experiences within the context of a company's brand.

Design Systems are often shown as a standalone website where internal and third-party teams of designers, developers and content editors have access to any and all digital assets or documentation they require. This ensures their output is consistent with the company's existing digital experiences.

Design Systems are the single source of truth for a company's digital presence, internal design, front end development, communications and other departments.

Design Systems have been gaining significant traction globally. The business case for them is extremely strong for larger companies and enterprise environments.

Below are some examples of companies already using Design Systems, and the names of those systems.



Atlassian Design



Gestalt



Grommet



Feelix



Solid



Canvas



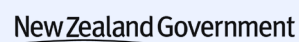
GOV.UK Design System



Lightning



Office UI Fabric



AOG Design System



Rizzo



Gel

BUSINESS CASE

HOW WILL VECTOR BENEFIT FROM A DESIGN SYSTEM

Vector's rebrand exercise is a good opportunity to build and implement a Design System. As business units within Vector begin to align their brands with the new Vector brand identity, so too must the digital experiences.

Design Systems are an effective way of managing a company's digital experiences, from user interface to communications. Companies that are responsible for building and maintaining digital experiences, often across multiple websites and platforms, use Design Systems to:

- Decrease development/design costs
- Improve quality and consistency of work across multiple business units and departments
- Improve workflow speed increase efficiencies
- Reduces technical & design debt
- Increases effective communication and knowledge sharing across departments and externally
- Protects against the improper use of assets
 - more efficient quality control
- Make people happier

It is important to outline the key offerings a Design System has that help designers and developers spend their time more effectively and productively while improving the experience of all those involved.

KEY OFFERINGS

DIGITAL BRAND GUIDELINES

Having a source of truth for designers, digital and print, gives greater control to Vector as to how their brand is represented across all mediums. Having the Design System house these guidelines allows for better version management; as designers can rest assured knowing the guidelines they are referencing are up to date.

COHESIVE EXPERIENCE

When digital experiences are built with the same components, people using these interfaces become familiar with the user experience. If used across multiple domains, people recognise the consistency allowing them to more easily make the connection between a Vector sub-brand and the Vector parent company.

COMPONENT LIBRARY

A component can be thought of as a building block that can be combined with other building blocks to create web pages and applications. A button or a search bar are both examples of a component.

Whether creating new pages on an existing website or building a completely new one; giving developers and designers the ability to leverage components that have already been designed, developed, tested, approved and released into production reduces the time spent implementing similar pieces of functionality and ensures that they're all on brand.

COMPREHENSIVE DOCUMENTATION

The effectiveness of a Design System is only as good as the documentation it holds. Maintaining clear, concise and up-to-date documentation on all things digital is crucial to the success of a Design System. The documentation guides developers and designers when building on new or existing digital experiences. Or in the case of content editors, offers examples and insight into Vectors tone of voice.

WHO IS IT FOR?

It's important that Design Systems involve every team that works within the company's digital space as decisions that make it into the Design System have an effect on the wider business. Having open discussions and sharing ideas is key to the success of the Design System as it allows all stakeholders to

share their thoughts and concerns and ideas before decisions are published to the Design System.

Below is an outline of all the roles a Design System often supports, and examples of how they would interact with it.

DESIGNERS

Digital Designers will:

- Keep designer component and digital asset documentation up to date
- Manage a user interface library that reflects components used in production
- Collaborate with developers and stakeholders as new components are created and existing components updated
- Ensure any designs created leverage the Design System where applicable. For example, colours, typography and library of user interface components

Graphic Designers will:

- Access the latest branding guidelines
- Ensure any designs created adhere to the branding guidelines
- Access required assets such as fonts, colours and imagery

Third-part Designers will:

- Collaborate with developers and stakeholders as new components are created and existing components updated
- Ensure any designs created leverage the Design System where applicable. For example, colours, typography and library of user interface components

DEVELOPERS

Web Developers will:

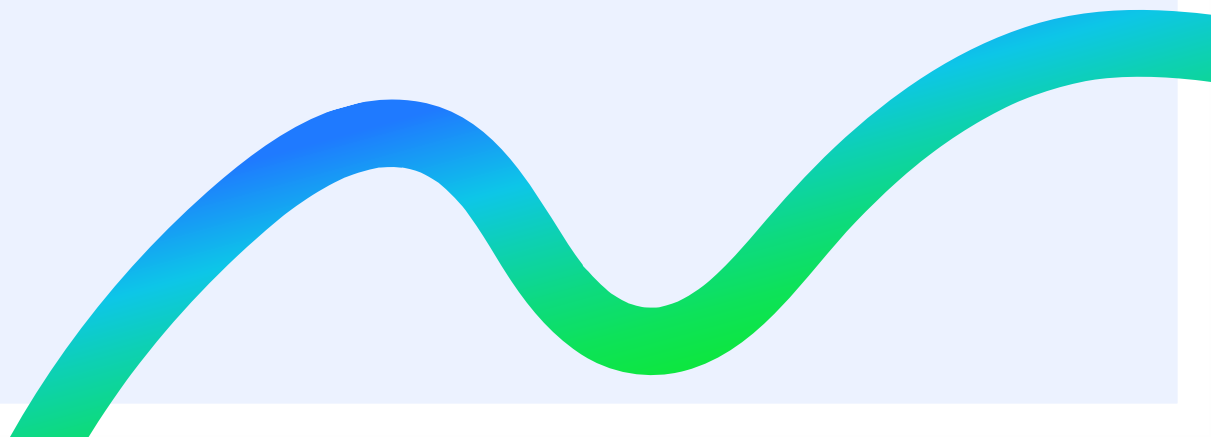
- Keep developer documentation up to date
- Manage a React component library
- Collaborate with designers and stakeholders as new components are created and existing components updated
- Ensure React components reflect the designs that were approved

Third-party Developers will:

- Utilise the existing React component library
- Collaborate with designers and stakeholders as new components are created and existing components updated
- Ensure React components reflect the designs that were approved

COMMUNICATIONS WILL:

- Maintain brand continuity
- Ensure any content written adheres to the applicable content guidelines. For example, tone of voice and use of imagery
- Ensure the content of the Design System itself reflects the content guidelines



EDISON

A Vector Design System Proof of Concept

THE SCOPE

The team built a POC as a way of testing out ideas, decisions and workflows practically. The requirements of the POC was to take a button component from design, to development, through to the Storefront where it would be documented for use by others. The POC also needed to show how we might manage multiple versions of our components, ensuring that outdated versions were still accessible.

It's important to remember that the purpose of the POC is to provide an example of how a Design System might work within Vector. The POC lets the team set up a quick and low-cost implementation of a Design System that helps the Vector team and their partners more easily identify limitations with out-of-the-box solutions, highlight areas that work well, as well as areas that require further research or testing.

This Design System, we're calling Edison.

METHODOLOGY

One of the most prominent methodologies for Design Systems is the Atomic Design methodology authored by front-end developer Brad Frost. Frost compares the structure of user interfaces to biology, as organisms are made of molecules, and molecules are made of atoms. The same breakdown of elements should be applied to user interfaces to improve consistency and reusability.

This breakdown of an interface into smaller elements is the foundation of the Design Systems we see being built and used today. Though there has been some divergence from the terminology used in the Atomic Design methodology, the principles remain the same.

From big to small, here is a breakdown of a typical user interface:

PAGES

This is what people experience. A page is a collection of everything below.

TEMPLATES

Templates are layouts of components for pages. For example, it's common to have an 'article' or 'blog' layout that differs from a 'dashboard' or 'contact form' layout.

ORGANISMS / COMPONENTS

Organisms are often referred to as components. These are more complex user interface elements. For example, a menu or a search bar. Organisms/Components can be used by themselves or nested within each other. A search bar component could be nested inside of a menu component, or they could be separated and still work just as well.

MOLECULES / COMPONENTS

Molecules too, are often referred to as components. Though these are often simple, and nested inside of larger organisms/

components. An example of a molecule would be a search bar that consists of a text label, a button and an input field.

ATOMS

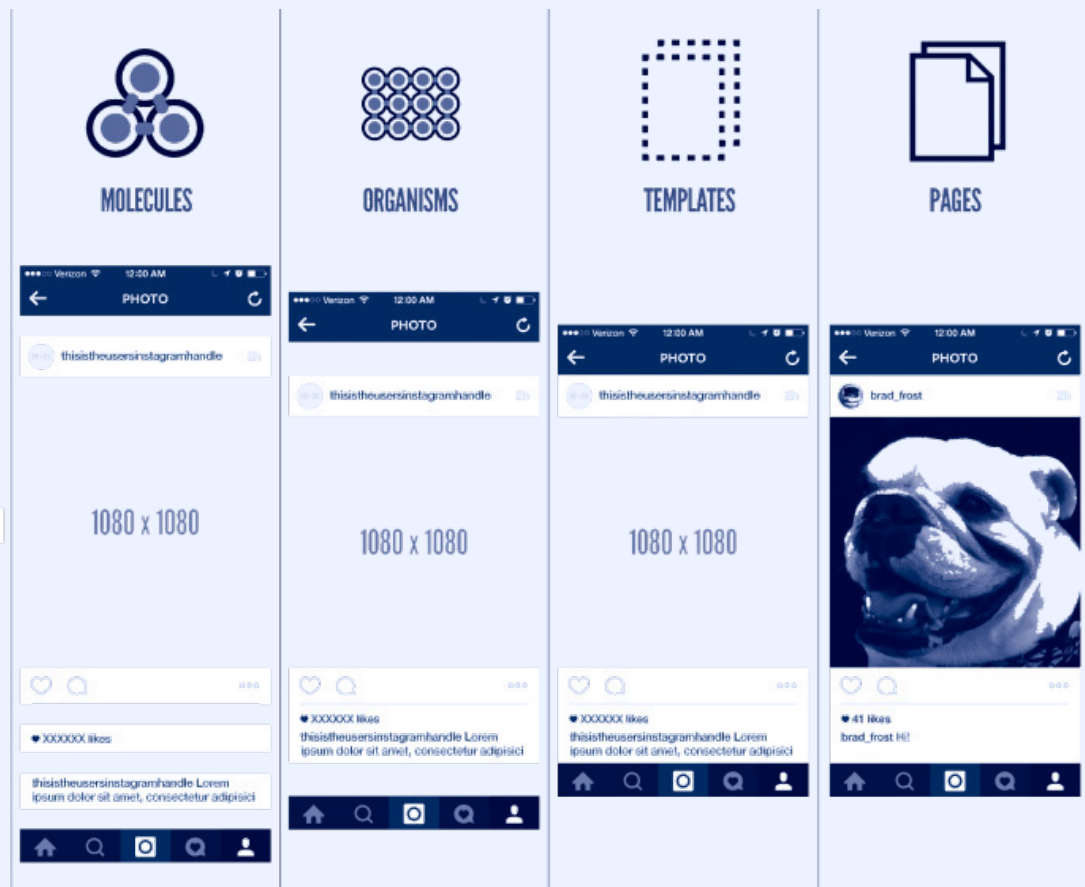
Atoms are generally the smallest part of a user interface. A button, an input field or a text label are three examples of an atom.

PARTICLES

These are elements such as fonts and colours and form the foundation of atoms.

DESIGN TOKENS

Design tokens is a term used to categorise visual user interface decisions that everything above will adhere to. This usually relates to the brand guidelines, and defines properties such as corner radius, border thickness and spacing between elements.



COMPONENT WORKFLOW OVERVIEW



THE TOOLS

DESIGN TOOLS

FIGMA

Figma was our user interface (UI) design tool of choice. It's platform agnostic, it can run on all major operating systems. Live collaboration allows designers, developers and stakeholders to view or edit the design file simultaneously. Cloud storage, making version management between designers and developers much simpler than other tools on the market.

DEVELOPMENT TOOLS

REACT

React is an open source JavaScript library for building user interfaces on the web. In a 2019 survey it was crowned the most used JavaScript library. It has the financial backing of Facebook, an enormous community backing and an abundant of resources available online. Prior to the POC, Vector had already made the decision to use React in production, so this was considered locked-in for the purpose of the POC.

REACT LIBRARY: REBASS

Rebass is an open source JavaScript library built for use with React. Rebass offers developers an easier and faster way of building re-usable components for use inside a Design System e.g. by defining styles with an array-based syntax.

GITHUB

GitHub is the leading platform for hosting and managing code. GitHub offers extensive integration support for tools such as Jira,

comprehensive permission support and basic task management. Prior to the POC, Vector already uses GitHub to manage its development projects. This was considered locked-in for the purpose of the POC.

STORYBOOK

Storybook is incredibly popular amongst web developers as a tool to test and preview components. Using this as our workshop was a safe decision and allowed us to easily integrate components into the POC's Storefront.

RAT

The RAT is the automatic deployment process that Vector utilises to make deploying updates to Vectors digital properties quicker and easier. The POC did not completely implement the RAT process, but the exercise did help us define what the RAT process would look like in a real-world production environment.

NETLIFY

Netlify helps developers quickly deploy websites directly from a GitHub repository. The reason Netlify was used in the POC was to better simulate a real-world use case where components would be deployed to Storybook and accessible by stakeholders via a unique branch-specific URL based on developer branch. Netlify handled the Storybook and URL piece to simulate the deployment to production.

NPM

npm (Node Package Manager) is a package manager for JavaScript. npm allows developers to automate dependency and package management, easing collaboration between developers. Using npm makes sure each developer has the necessary JavaScript packages required to continue development on the project. It also gives developers clear version management, allowing developers to specify what versions they want their packages to be.

ARTIFACTORY

Artifactory lets Vector host a private collection of npm packages for use by developers. For the purpose of the POC, this package is a button component. Artifactory allows authorised developers to use this button component in their projects by importing the appropriate npm package, hosted on Artifactory.

MANAGEMENT TOOLS

JIRA

Atlassian's Jira is a tool proven capable of managing digital products of all sizes. Vector already operates Jira inside their organisation, so using this to manage the POC project and as a tool within the Design System workflow makes implementation, ease of access and licensing costs trivial.

ZEROHEIGHT

Zeroheights' flexibility in information architecture, its popularity amongst Design System teams, its affordable pricing model, and its integration with both Figma and Storybook were the reasons the team picked this to implement into the POC.

COMPONENT PROCESS, ROLES & TOOLS

- 01 need**
business, stream lead
- 02 define**
scrum master, stream lead
[JIRA](#)
- 03 review**
scrum master, stream lead
[JIRA](#)
- 04 handover**
scrum master, stream lead,
design lead
[JIRA](#)
- 05 design**
design lead, designers
[Figma](#), [JIRA](#)
- 06 review**
scrum master, stream lead,
designers
[Figma](#), [JIRA](#)
- 07 handover**
scrum master, stream lead,
designers, developer lead
[Figma](#), [JIRA](#)
- 08 develop**
scrum master, stream lead,
designers, developer lead
[React](#), [GitHub](#), [Figma](#), [JIRA](#)
- 09 review**
developer lead, scrum
master, stream lead
[Storybook](#), [Figma](#), [JIRA](#)
- 10 test**
stream lead
[Storybook](#), [JIRA](#)
- 11 handover**
scrum master, stream lead,
developer lead
[GitHub](#), [Artifactory](#), [JIRA](#)
- 12 document**
designers, developers
[Zeroheight](#), [JIRA](#)
- 13 review**
Design lead, dev. lead,
comms team
[Zeroheight](#), [JIRA](#)
- 14 publish**
Design lead, dev. lead,
comms team
[Zeroheight](#)
- use**
developers, designers
[Zeroheight](#), [Artifactory](#),
[Figma](#)
- 16 deploy**
vector team

CREATING COMPONENTS

01. NEED

A need is identified. In the context of the POC this is a button. This will be communicated with the stream lead involved with Edison.

02. DEFINE

Use cases, acceptance criteria and examples of this component will be outlined in a Jira ticket. Each component will have its own ticket which will be continuously updated as the component is further defined, designed, developed and published to the Design System. The component will also have one of the following statuses:

- Under consideration
- Define
- Design
- Development
- Testing
- Documentation
- Published

In this step, the Jira ticket status will be 'Define'.

03. REVIEW

Once a ticket has been clearly defined, the acceptance criteria and use cases will be approved. The outcome of this review process will either be to continue definition or to begin the handover step.

04. HANDOVER

Designers will need to be onboarded by the scrum master, stream lead and potentially the design lead. Designers will also be given access to the relevant Jira ticket, and be run through the use cases and acceptance criteria that they're working towards. It's important that designers are left with a clear understanding of what is required, when their deadlines are and who they should collaborate with for reviews and approval.

05. DESIGN

The designer(s) will be working inside Figma and will leverage the shared UI library inside Figma to ensure that their designs are consistent with Edison. Work-in-progress designs and designs that are ready for review will be added to the Jira ticket for ease of access for stakeholders. This will be done using a Figma URL.

In this step, the Jira ticket status will be changed to 'Design'.

06. REVIEW

When a design is ready to be reviewed, the scrum master and/or design lead and/or stream lead will review the work inside Figma while referring back to the Jira ticket for the acceptance criteria the designs should meet. The outcome of this review process will either be to continue iterating on the design or to begin the handover step for developers.

07. HANDOVER

Developers will need to be onboarded by the scrum master, stream lead and/or a designer or design lead. Developers will be given access to the relevant Jira ticket, and be run through the use cases, acceptance criteria and the approved designs. It's important that developers are left with a clear understanding of what is required, when their deadlines are and who they should collaborate with for reviews and approval.

08. DEVELOP

Developers will be referencing the Jira acceptance criteria, the approved set of Figma designs, and coding using the React JavaScript library. When the developers are ready to begin, they will need to follow the steps below:

- 'Fork' the Vector 'Component Library' repo from Vectors' GitHub
- Create a 'feature branch' of that repo using a URL, friendly name (no special characters or spaces)

When ready to test or review the component within Storybook, developers will need to request to merge their feature branch with the Vector 'Component Library' repo.

Once this is merged by the developer(s) responsible for the Vector 'Component Library' repo, the RAT will run a development build, generate a unique component-

specific URL that the developer can then attach to the relevant Jira ticket, accessible for testing and review.

In this step, the Jira ticket status will be changed to 'Development'.

09. REVIEW

The URL generated in the previous step will allow designers, developers and other stakeholders to review the component inside Storybook. The outcome of this review process will either be to continue iterating on the component or to begin the handover step for further, more comprehensive testing.

10. TEST

Testing will be handled by the stream lead until an independent test team is assembled inside Vector. The stream lead will reference the Jira ticket, the Figma designs and the Storybook URL. Once the stream lead is satisfied the component is ready for production, they will begin the handover step for publication. Any issues or bugs found will be communicated to the developers and logged inside the Jira ticket.

In this step, the Jira ticket status will be changed to 'Testing'.

11. HANDOVER

The final handover will be between the person(s) tasked with documenting the component for publication into the Design System, a developer and/or developer lead, the stream lead and/or the scrum master.

Before documentation begins, the developer(s) responsible for the Vector GitHub component library will need to merge the feature branch and trigger another RAT process. This time the RAT will output a Storybook URL that contains all published components and publish an npm package to the Vector Artifactory library with version numbers. This new Storybook URL will be added into the Jira ticket and given to the person(s) responsible for the documentation for integration into zeroheight.

In order to complete the documentation stage, the person(s) will be taken through the Jira ticket, which will contain the latest links required for integration into the Design System (Storybook and Figma). It's important that the person(s) are left with a clear understanding of what is required, when their deadlines are and

who they should collaborate with for reviews and approval.

12. DOCUMENT

The person(s) completing the documentation will need to communicate with the designers, developers and stream lead in order to get a complete understanding of how the component should be used in the context of design and development. This will be done inside zeroheight, the platform Edison is built on.

In this step, the Jira ticket status will be changed to 'Documentation'.

13. REVIEW

This is the final review before the component and accompanying documentation will be published to the Storefront. The review will include all stakeholders, including the designer, developer and stream lead. The communications team will need to approve the content before it is published to make sure it is in line with any existing documentation published and conveys the appropriate tone of voice.

14. PUBLISH

At this stage, everything should be ready for publication.

- The Figma component should now be published to the shared Figma library
- The React component should be living inside of Artifactory
- The testing process should be complete
- The documentation should be clear and concise
- The Zeroheight component page should be utilising the product Storybook URL
- The editor of zeroheight can now publish an updated version of the Design System
- The Jira ticket status can now be changed to 'Published'
- RAT will generate the version number.

15. USE

The component can now be used by internal Vector designers and developers, as well as third-party designers and developers that have been granted access. For specific documentation and guidelines on how to use the component, the designers and developers should now refer to Edison.

Developers will use npm to install the component package from the Vector component library. With access to all other publish components for use inside of their application.

16. DEPLOY

The deployment process of publishing a website is managed internally by the Vector team. This process will not need to be further defined as part of the Design System POC.

UPDATING COMPONENTS

Once Edison has been populated with the most common components, a lot of the work from then on will be about updating existing components as opposed to creating new ones. Updating components is not necessarily as linear as the new component workflow, as it will depend on the type of update required.

Most updates to a component can be categorised into two types, updates that do not have an impact visual design and updates that do have an impact on visual design.

UPDATES THAT DO NOT HAVE AN IMPACT ON THE VISUAL DESIGN

For example, fixing a bug or modifying component logic will likely include the following steps:

- **01. Need / 02. Define**
Define the bug that requires fixing and log it within Jira, outlining as much information as possible
- **03. Review / 04. Handover**
Update the Jira ticket to 'Development' and provide further clarity if requested
- **08. Develop**
The development team will attempt to fix the bug and push their changes to the GitHub repo
- **09. Test**
The testing team will attempt to reproduce the bug, if the bug has been fixed then the Jira ticket can be assigned to the developers to push the update into production
- **14. Publish**
The developers will push their changes to the Vector GitHub repo and trigger another RAT build to update Storybook and Artifactory
- **15. Use**
The updated version will be accessible to the developers. The developers will need to manually update their component to the latest version
- **16. Deploy**
The Vector team will deploy the updates to production

UPDATES THAT DO HAVE AN IMPACT ON VISUAL DESIGN

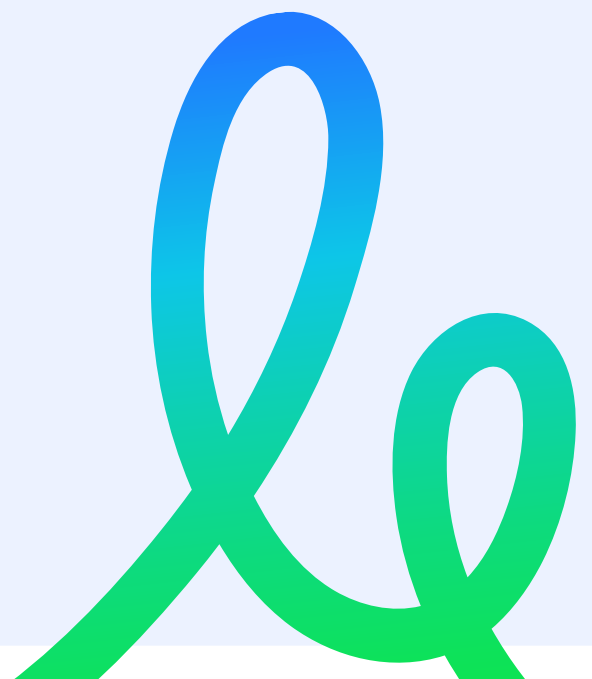
For example, updating a button colour will likely include all of the same steps as creating a new component, as both the designer and developer will need to work on the update and ensure their respective libraries are kept in sync with production.

CONTENT, ASSETS & COPY

Although the POC was primarily focused on implementing and testing a workflow for creating and managing components and their documentation. Edison aims to assist the communications team, content editors and brand designers too.

The more integration and engagement from Vector teams, the more valuable Edison will prove to be. Edison can help promote content guidelines ensuring that all public facing content is aligned and consistent with the Vector brand. From language used, to the tone of voice conveyed, to the imagery and aesthetics of collateral.

Edison gives Vector the opportunity to speak the same language, to utilise the same resources, to reference the latest guidelines and to build on work of others.



EXCLUSIONS

The POC focused on the tools and workflows that could be adopted across multiple teams within Vector and out to Vector partners.

This POC looked at the one key design component (buttons) to realise this workflow. As such, the scope of this project does not include core teams and branches of Vector such as Communications. We look forward to bringing the key stakeholders into the next stage of this project with the information they require in order to design and realise how they will best benefit, plug in and utilise the eventual live state Design System.

The scope of the POC required the Storefront, zeroheight, to only show an example of a button component and its documentation. The team has added some further example content to the Storefront to demonstrate what other information and assets can be housed inside of Edison. This includes examples for content editors, print designers, digital designers and developers.

RISKS

AUTOMATION

The level of automation across the Design System workflow depends largely on the choice of tools we select. If we were to replace any of the tools used in this POC with another, we would need to re-assess the process by we present, manage and document components and brand assets.

ZEROHEIGHT

Using an out of the box solution places restrictions on what the Design System can or cannot do. If Vector required functionality that was not built into zeroheight we would either have to use another Storefront solution or look at a workaround for the desired use case.

RECOMMENDATIONS

It is clear that the Design System will be of great value to Vector. The potential of bringing a Design System to life is significant - if not crucial. The concept, tools and workflow provided in this report gives Vector a solid foundation to realise its full potential.

While the POC focused on a specific workflow, it is not limited to a digital design & development workflow. Control, consistency, visibility and efficiency across communications, brand, content, design and development is what wider success could look like. We recommend bringing key departments and stakeholders within Vector to explore this.

There is a range of existing tools that we have been able to successfully use in this POC. As such, we do not see a need to build a bespoke solution. We may however want to assess other tools and workflows utilised by wider Vector teams.

There is a strong case to implement this Design System for any upcoming Vector experience design and development projects on the horizon.

We look forward to a demo of the POC in the coming weeks, and extending the project out to the wider teams as we co-create Vector's very own Design System.

APPENDIX

KEY DEFINITIONS

The following terminology is used throughout this document:

CSS

Cascading Style Sheets or CSS for short is the language used by web developers to control the layout and user interface design of elements on a page.

Design System

A Design System maintains the visual and functional elements of an organisation in one place, in order to fulfil its brand principles through design, realisation, and development of products and services.

Workshop

A workshop is the place developers will display and test their components with other developers and to internal Vector stakeholders for review, feedback and approval.

RAT

The RAT framework is intended to provide a tool to simplify, standardise and streamline the development of new software products within an enterprise environment based on a shared set of codified architectural patterns and common guiding principles.

Repository / Repo

A repository (or repo for short) is a place where developers host their source code, usually for collaboration with other developers. The most common repo platform is GitHub. Another way to think about a repo is simply as a shared folder.

Storefront

A Storefront is where the latest version of components will be displayed, accompanied by documentation for designers and developers on how to use them. This is also where the broader digital guidelines are found, with easy access to digital brand assets such as imagery, font files and logo files.

URL

A URL (Uniform Resource Locator) is another name for a web address, or a link. The string of characters that are displayed in a browsers address bar.

PROJECT PROCESS

Slack Channel

A Slack channel was created to allow more casual conversations to take place, and as a way for the team to share thoughts, feedback and information with each other outside of scheduled meetings or collaboration sessions.

Confluence

Part of the same suite of tools as Jira, Confluence is an online tool used to house documentation. Confluence was used to keep a history of meeting notes, weekly reports, outline pros and cons of tools and define common Design System terminology.

Jira

The project was managed through Jira. Tickets were created based on the scope of the project, with each ticket being assigned a team and a deadline. This allowed the team to understand what aspects of the POC they would be involved in, and how the project was tracking.

Weekly Meetings

Every week the team would have 1-2 fifteen-minute meetings to update each other on their tasks, the progress of the project, and to raise any questions or concerns. This allowed everyone involved in the project to be kept in the know.

LICENSE COSTS OF POC TOOLS

The following terminology is used throughout this document:

ZEROHEIGHT

- \$15-\$45 USD per editor/per month.
- Enterprise requires a custom quote.

FIGMA

- \$15 USD per editor/per month
- Enterprise \$45 USD per editor/per month

All other tools used in the POC are either free, or are already in use at Vector.

OTHER TOOLS CONSIDERED

During the research phase of this report, Vector, UiRevolution, The Goat Farm, Secion6 and ClearPoint assessed a variety of tools and products that could be used for the POC. This included tools for designers, front-end web technology and out-of-the-box solutions for the workshop and Storefront.

STYLEGUIDIST

Styleguidist is a tool for React developers to test and preview components,

with development documentation support. There were two reasons the team decided to use Storybook over Styleguidist. First, Styleguidist has a lot of functionality overlap with our Storefront tool zeroheight, though with a strong focus on content and guidelines for developers, and little consideration for designers and content editors. Secondly it does not integrate with our Storefront tool zeroheight, making it impossible to automatically render components directly from GitHub.

SKETCH

Sketch is the most widely used user interface tool, garnering huge community support, improving the application with custom-made plugins. Though it's restrictions around platform support, it's lack of in-built version management and collaboration support has encouraged us to use an alternative.

REFERENCES

<https://medium.com/@didoo/measuring-the-impact-of-a-design-system-7f925af090f7>
<https://5d91b658cd11460007e19227--hardcore-leavitt-323400.netlify.com/>
<https://www.styled-components.com/>
<https://medium.com/eightshapes-llc/documenting-components-9fe59b80c015>
<https://rebassjs.org/>
<https://github.com/emotion-js/emotion/tree/master/packages/emotion-theming>
<https://www.conventionalcommits.org/en/v1.0.0-beta.4/>
<https://medium.com/zeroheight/announcing-a-new-plan-for-large-teams-4ebd63af1610>
<https://github.com/VectorLimited/website-rebrand-poc>
<https://medium.com/storybookjs/storybook-docspage-e185bc3622b>
<https://design-systems.slack.com/archives/COENUN4H0/p1568670426034400>
<https://github.com/VectorLimited/website-rebrand-poc>
<https://docs.google.com/spreadsheets/d/1KF2jVemVfzactf6DRVuyhBXsqQshAvRID84dga6Om8g/edit?usp=sharing>
<http://bradfrost.com/blog/post/the-workshop-and-the-storefront/>
https://docs.google.com/document/d/1RoDcq5Q4X2F8oIPM90jtWDTjllw_Qnn6cQzzusxp4CQ/edit?usp=sharing
<https://www.plectica.com/maps/8BNHPVDCD>
<https://bigmedium.com/ideas/boring-design-systems.html>
<https://github.com/framer/framer-bridge-starter-kit>
<https://gel.westpacgroup.com.au/case-studies/summary/>
<https://medium.com/eightshapes-llc/and-you-thought-buttons-were-easy-26eb5b5c1871>
<https://uxdesign.cc/how-much-is-a-design-system-worth-d72e2ededf76>
<https://www.framer.com/support/using-framer-x/extract-visual-specs-for-handoff/>
<https://www.framer.com/blog/posts/announcing-framer-bridge/>
<https://github.com/design-tokens/community-group/blob/master/README.md>
<https://zeroheight.com/>
<https://www.designsystems.com/white-labeling-putting-the-design-system-in-users-hands/>
<https://www.akveo.com/blog/helping-organizations-with-eva/>
<https://segment.com/blog/driving-adoption-of-a-design-system/>
<https://www.uxpin.com/merge>
<https://github.com/LisaDziuba/Awesome-Design-Tools>
<https://medium.com/harbor-school/building-a-design-system-ui-component-using-framer-x-59ad2ac7292e>
<https://www.framer.com/support/using-framer-x/import-react-components/>
<https://blog.prototypr.io/bringing-design-system-components-from-production-into-framer-x-786e89be2250>
<https://medium.com/code-build/why-design-systems-are-going-to-shape-the-internet-b87dfa2657e3>
<https://design-system-alpha.digital.govt.nz/get-started/>
<https://medium.com/eightshapes-llc/tokens-in-design-systems-25dd82d58421>
<https://uxdesign.cc/the-designers-guide-to-design-system-roi-success-16856038dd4d>
<https://medium.com/storybookjs/component-story-format-66f4c32366df>
<http://engineering.lonelyplanet.com/2014/05/18/a-maintainable-styleguide.html>
<https://medium.com/storybookjs/component-story-format-66f4c32366df>
<https://design-systems.slack.com/>
<https://adele.uxpin.com/>
<https://jaxenter.com/javascript-ecosystem-2019-158124.html>
<https://uxplanet.org/design-system-imposter-syndrome-10809da49f5f>
<https://uxdesign.cc/how-much-is-a-design-system-worth-d72e2ededf76>
<https://uxdesign.cc/the-designers-guide-to-design-system-roi-success-16856038dd4d>
<https://uxdesign.cc/how-to-measure-design-system-impact-guide-f1f9f0c3704f>
<https://medium.com/@didoo/measuring-the-impact-of-a-design-system-7f925af090f7>

