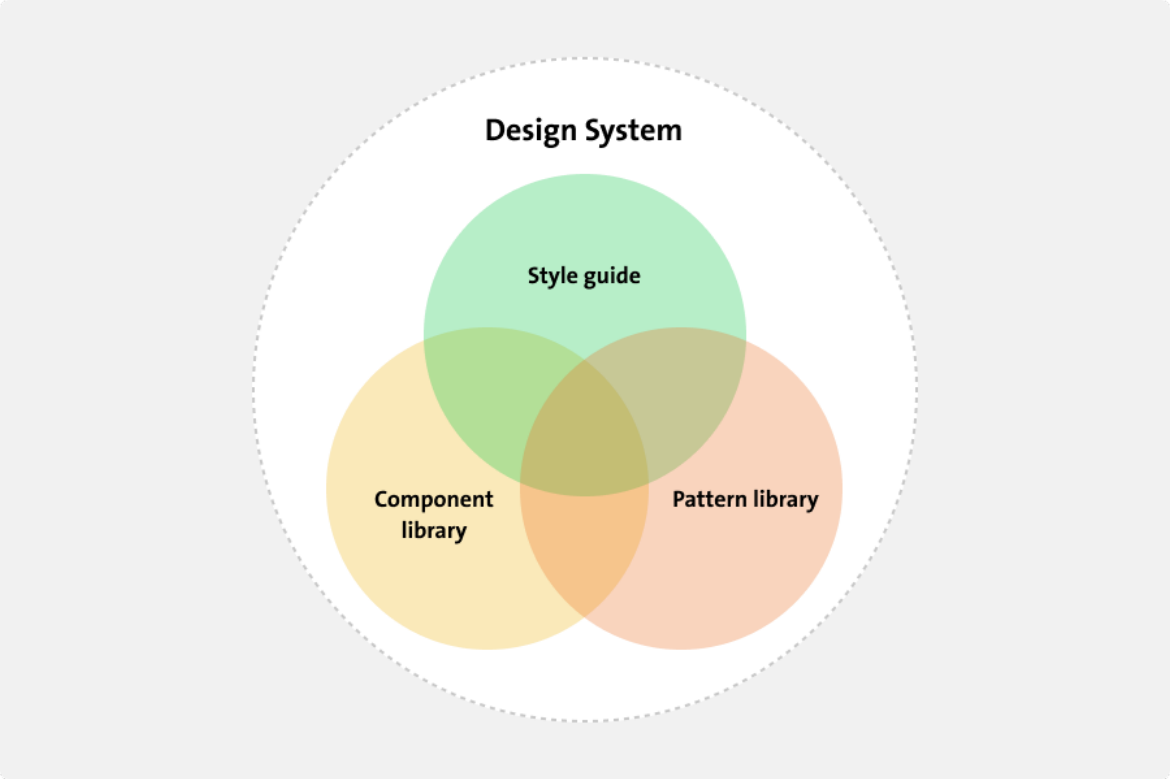
Design Systems

What makes up a Design System?

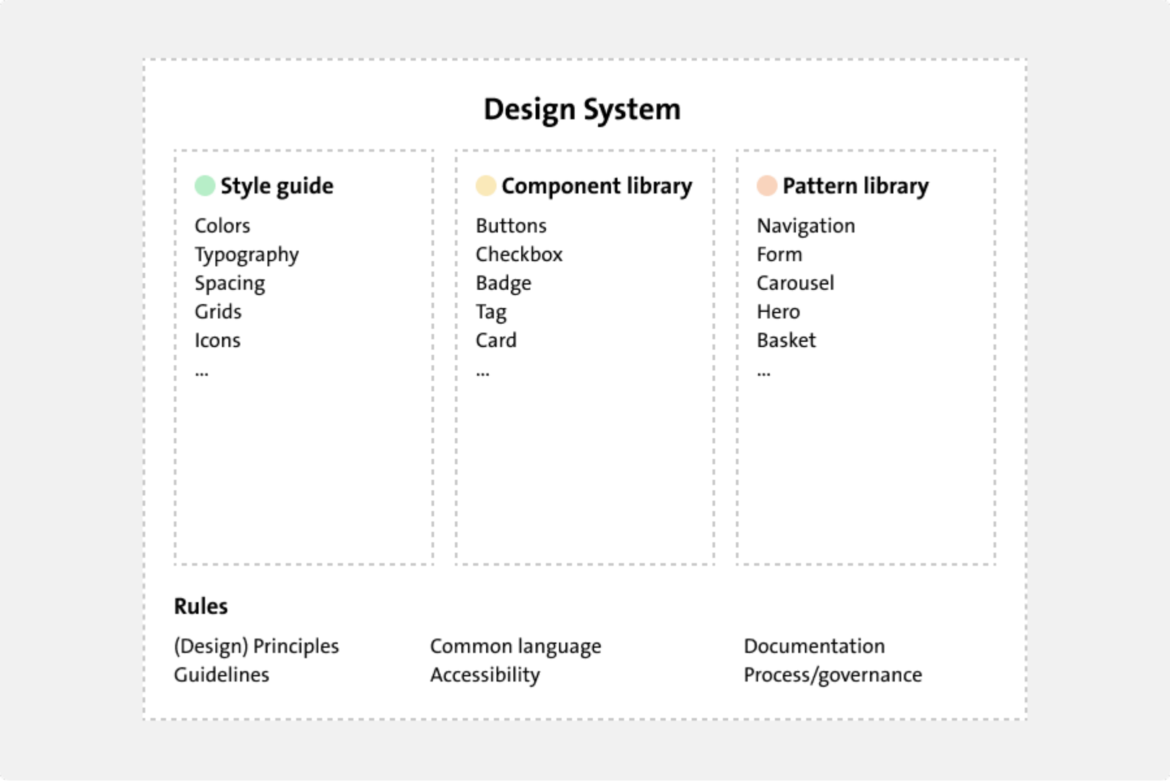
A Design System is a collection of reusable components, guided by clear standards, that can be assembled together to build any number of applications. It serves as a single source of truth for designers and developers, ensuring that everyone is on the same page and that the product is consistent and cohesive.



A Design System typically includes the following elements:

- **Style guide**: The core values and beliefs that guide the design of the system, such as simplicity, consistency, and accessibility as well as the visual and verbal elements that define the brand, such as colors, typography, and tone of voice.
- **Component library**: The building blocks of the system, such as buttons, forms, and navigation bars, that can be combined to create interfaces.

- **Pattern library**: Common design solutions for recurring problems, such as search bars, carousels, and modals, that can be reused across projects.



Design System

Style guide
Colors
Typography
Spacing
Grids
Icons
...

Component library
Buttons
Checkbox
Badge
Tag
Card
...

Pattern library
Navigation
Form
Carousel
Hero
Basket
...

Rules
(Design) Principles
Guidelines
Common language
Accessibility
Documentation
Process/governance

Benefits of Design Systems

Consistency

A design system ensures that all products and features have a consistent look and feel, which helps to establish a strong brand identity and build trust with users.

Efficiency

Design systems save time and effort by providing a library of pre-built components that can be easily reused across projects. This allows designers and developers to focus on creating new features and experiences, rather than reinventing the wheel each time.

Scalability

Design systems are scalable, meaning they can grow and evolve over time to meet the changing needs of a product or organization. This flexibility allows teams to adapt to new challenges and opportunities without starting from scratch.

Collaboration

Collaboration is key to the success of any design system. By bringing together designers, developers, and other stakeholders, a design system can foster a culture of teamwork and shared ownership, leading to better outcomes for everyone involved.

Having components, patterns, and guidelines in one place makes it easier for teams to work together and stay aligned on the vision and goals of a project.

Improved User Experience

By providing a consistent and cohesive experience across all touchpoints, a design system can help to improve the overall user experience. Users will be able to navigate products more easily, understand how to interact with them, and feel confident in their interactions.

In a lot of ways, improving the Developer Experience (DX) is just as important as improving the User Experience (UX). A well-designed design system can make it easier for developers to build and maintain products, leading to faster development cycles, fewer bugs, and a more enjoyable working environment.

Cost-effectiveness

Design systems can help to reduce costs by streamlining the design and development process, improving efficiency, and reducing the need for rework. By investing in a design system upfront

Reusing components and patterns across projects can also help to reduce the time and effort required to build new features, leading to faster time-to-market and increased productivity which translates to cost savings.

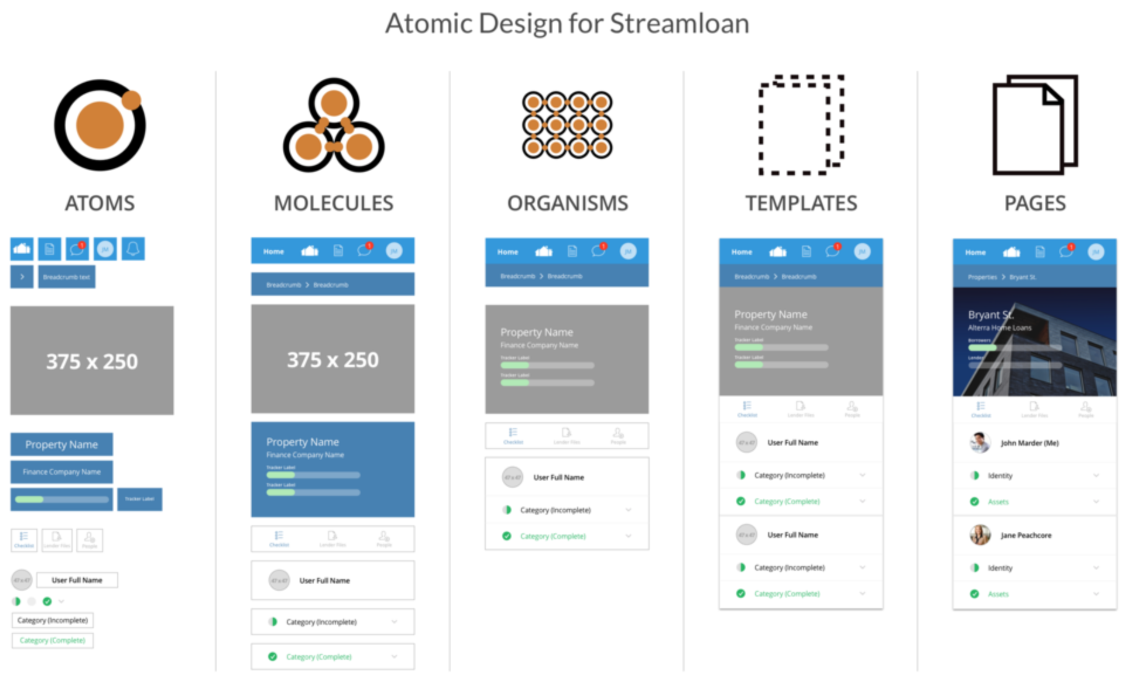
Building a Design System

Adopting Atomic Design

Atomic Design is a methodology for creating design systems that breaks down interfaces into smaller, reusable components, such as atoms, molecules, organisms, templates, and pages. By starting with the smallest building blocks and working up to larger structures, designers can create more flexible and scalable systems that can be easily adapted to different contexts.

It prioritizes consistency, scalability, and reusability, making it an ideal framework for building design systems. By following the principles of Atomic Design, designers can create more cohesive and coherent products that are easier to maintain and update over time.

Here is an example of how Atomic Design is split into different levels:



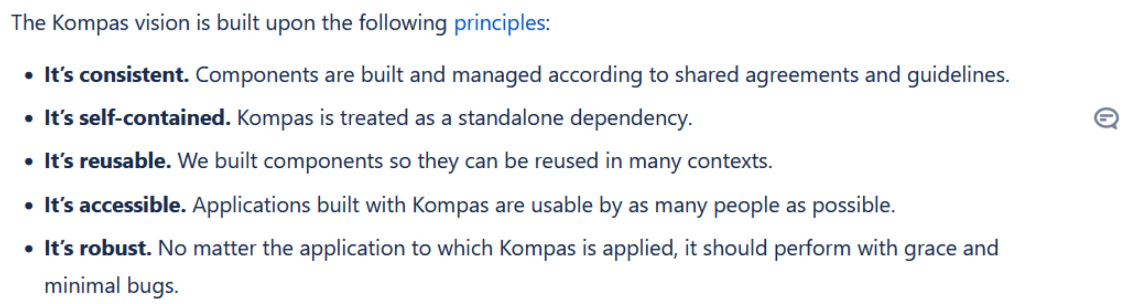
Atomic was introduced by Brad Frost in his book [Atomic Design](). Since then, it has become a popular framework for building design systems and has been adopted by many organizations, including IBM and Shopify.

Defining Design Principles

Design principles are the core values and beliefs that guide the design of a system. They help to establish a shared vision and set of goals for the team, ensuring that everyone is on the same page and working towards the same objectives.

Design principles can cover a wide range of topics, such as simplicity, consistency, accessibility, and usability. They should be clear, concise, and actionable, providing a framework for making decisions and solving problems.

Here is an example, showcasing the design principles of Kompas, an internal design system at Jumbo Supermarkten:



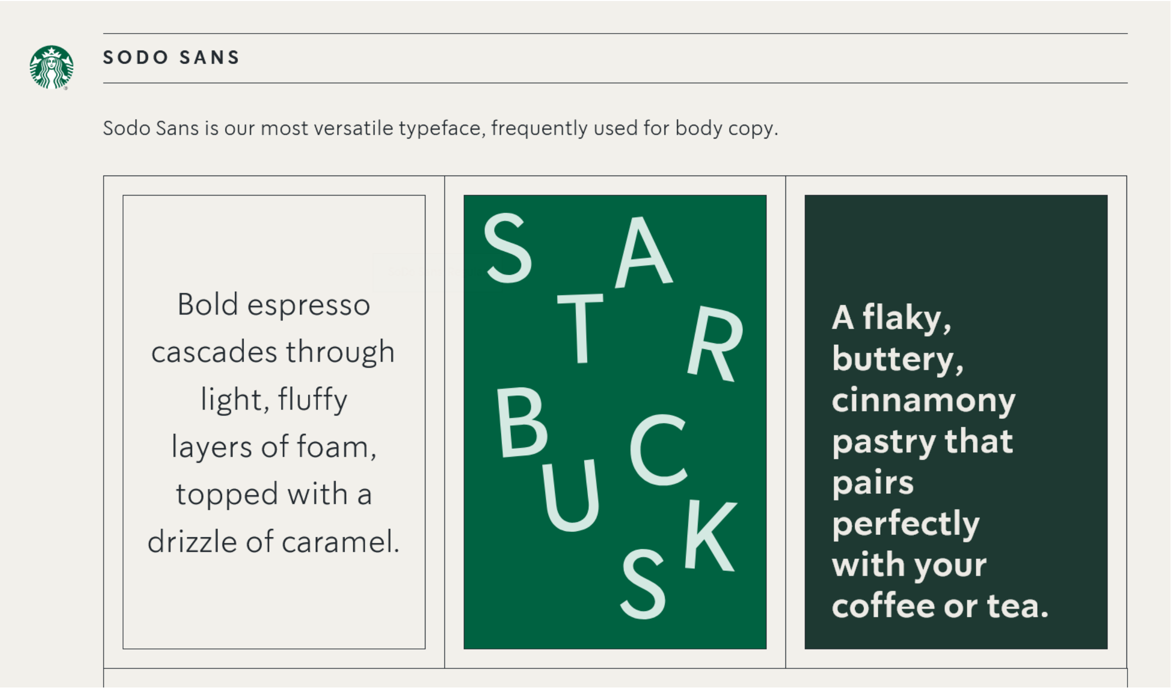
The Kompas vision is built upon the following principles:

- **It's consistent.** Components are built and managed according to shared agreements and guidelines.
- **It's self-contained.** Kompas is treated as a standalone dependency.
- **It's reusable.** We built components so they can be reused in many contexts.
- **It's accessible.** Applications built with Kompas are usable by as many people as possible.
- **It's robust.** No matter the application to which Kompas is applied, it should perform with grace and minimal bugs.

Creating a Style guide

A style guide is a document that outlines the visual and verbal elements that define a brand, such as colors, typography, and tone of voice. It serves as a reference for designers and developers, ensuring that everyone is using the same assets and following the same guidelines. Style guides can take many forms, from simple documents to interactive websites. They should be easy to access and update, making it easy for teams to stay aligned and consistent.

Here is an example of a style guide from Starbucks



SODO SANS

Sodo Sans is our most versatile typeface, frequently used for body copy.

Bold espresso cascades through light, fluffy layers of foam, topped with a drizzle of caramel.

A flaky, buttery, cinnamony pastry that pairs perfectly with your coffee or tea.

Developing UI Components

UI components are the building blocks of a design system, such as buttons, forms, and navigation bars. They should be reusable, flexible, and easy to customize, allowing designers and developers to create a wide range of interfaces without starting from scratch.

These components are created based on designs and patterns that are created in the pattern library with the companies principles and brand guidelines in mind. This is how the 3 layers of a design system are connected.

The leading tech companies tend to create component libraries using React, Angular, or Vue.js.From there it is a good idea to publish these libraries as npm packages, so that they can be easily shared and reused across projects. These libraries are then used to build the UI of their products.

Documenting the System

Documentation is a critical part of any design system, as it provides a reference for designers and developers to understand how the system works and how to use it effectively. It should be clear, concise, and up-to-date, providing information on everything from design principles to component usage.

Documentation can take place on many platforms such as Confluence, Notion, Google docs, a companies internal wiki, or even a custom-built website. The most important thing is that it is accessible and easy to update, easy to access, readable, concise and fills in all the missing pieces of the design system.

Implementing and Maintaining a Design System

Just like any other product, a design system requires ongoing maintenance and support to ensure that it remains effective and up-to-date. This includes monitoring usage, collecting feedback, and making improvements based on user needs and business goals.

Understanding the contribution model

Nathan Curtis, a design systems expert, has defined a contribution model that outlines the roles and responsibilities of different team members in a design system. This model helps to clarify who is responsible for what and how contributions should be made, ensuring that everyone is on the same page and working towards the same goals.

Nathan Curtis' Contribution Model offers a thorough approach to developing design systems, highlighting teamwork, inclusivity, and ongoing improvement. The model identifies three main roles: the core team, responsible for creating and managing the design system; a broader group of stakeholders contributing through feedback and participation; and community contributors from outside the organization enriching the system with their ideas. These groups communicate through various channels, including direct contributions, feedback loops, and clear guidelines ensuring consistency.

- **Governance and decision-making** prioritize transparency and consensus, aligning with organizational goals and user needs.

- **Recognition and rewards** acknowledge contributors' efforts and provide opportunities for growth, fostering community engagement.

- **Continuous improvement** is central, treating the design system as a dynamic entity evolving iteratively based on user feedback and established metrics.

By embracing Nathan Curtis' model, organizations cultivate a collaborative, innovative, and user-focused culture, enhancing both internal workflows and user experiences.

Integration with Design and Development Processes

Integrating the design system into the design and development processes is essential for ensuring its success. This includes providing training and onboarding for new team members, versioning and updating components, and establishing clear guidelines for how the system should be used. By embedding the design system into the workflow, teams can work more efficiently and produce better results. key aspects of integrating a design system into the design and development processes include:

- **Training and Onboarding**: Providing resources and support for new team members to learn how to use the design system effectively.
- **Versioning and Updating**: Managing changes to the design system and ensuring that everyone is using the latest components and guidelines. This includes version control, release notes, and changelogs. Using Semantic Versioning can help to communicate changes effectively and avoid breaking changes.
- **Guidelines and Best Practices**: Establishing clear guidelines for how the design system should be used and providing best practices for designers and developers.
- **Collaboration and Communication**: Encouraging collaboration and communication between team members to ensure that everyone is on the same page and working towards the same goals. A design system is only as effective as the team that uses it, so it's important to foster a culture of collaboration and communication. A good support system can help to ensure that everyone is on the same page and working towards the same goals. Fast response times are crucial for keeping the design system up-to-date and relevant.

Conclusion

To summarize, a design system acts as the foundation for consistent and effective product development by providing a cohesive set of guidelines, components, and principles for designers and developers. It integrates various elements such as style guides, component libraries, and pattern libraries to ensure a seamless user experience across different applications and platforms.

At its core, the design system is rooted in Atomic Design, a methodology popularized by Brad Frost. This approach deconstructs interfaces into reusable components, starting from basic elements and gradually building up to more complex structures. This enables organizations to create adaptable systems that cater to diverse user needs.

Clear design principles are essential for guiding the development and evolution of the design system. These principles, which encompass values like simplicity, consistency, accessibility, and usability, provide a shared vision for the design team, ensuring alignment throughout the design process.

Documentation plays a crucial role in documenting the design system, serving as a comprehensive reference for stakeholders. It covers everything from design principles to coding standards, evolving alongside the design system to capture changes and best practices over time.

Implementing and maintaining a design system requires collaboration and iteration. Nathan Curtis' Contribution Model offers valuable insights into managing a design system effectively, emphasizing transparency, inclusivity, and continuous improvement.

Governance and decision-making are vital for ensuring the success of the design system. Transparent processes and clear communication channels align stakeholders and drive informed decisions, while recognition and rewards foster a sense of community and motivation within the team.

Continuous improvement is integral to the design system's evolution, driven by user feedback, technological advancements, and changing business needs. Regular assessments and metrics tracking help identify areas for optimization, ensuring the design system remains relevant and effective.

Collaboration and communication are key to integrating the design system into the design and development processes. By fostering a culture of teamwork and knowledge sharing, organizations can maximize the impact of the design system and deliver exceptional user experiences.

Sources

- Comprehensive guide to Design Systems
- Design System contribution model
- Atomic Design
- Atomic Design revolutionizing Frontend Engineering
- Starbucks Style Guide
- Internal Kompas Documents from Jumbo Supermarkten

Next page
Project Management