


The Role of Style Tokens in Modern Design Systems: Ensuring Consistency and Flexibility

Rushilkumar Patel ¹

¹ Individual Researcher

ARTICLE INFO

Received: 30 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Introduction: In an era of rapid digital proliferation, *design systems* have emerged as critical infrastructures for organizations looking for both consistent brand identity and flexible adaptability. The backbone of modern design systems lies in *style tokens* (also referred to as *design tokens*)—named entities that store core design attributes and can be transformed for use across multiple platforms. Based on principles such as *Atomic Design* [12] and extended by thought leaders like *Nathan Curtis* [9], style tokens bridge the gap between designers and developers, fostering maintainable and scalable solutions. This paper explores the evolution of design tokens, outlines best practices for their creation, compares token structures across leading companies, and emphasizes the balance between consistency and flexibility critical to contemporary design systems.

Keywords: design tokens, design systems, systematic architecture, brand consistency, component libraries, theming, user experience, atomic design.

INTRODUCTION

The growing need for consistent and scalable design in web, mobile, and emerging interfaces has led to the formalization of **design systems**—comprehensive guidelines and reusable components that unify brand experiences [18, 9]. Within these systems, **design tokens** have surfaced as a fundamental concept. By abstracting design decisions into discrete, named values (e.g., for color, typography, and spacing), tokens allow teams to maintain consistent aesthetics while seamlessly adapting to different technologies and platforms [26, 7].

THE IMPORTANCE OF CONSISTENCY AND FLEXIBILITY

- **Consistency:** Ensuring a cohesive brand presentation is essential for user trust [22]. Tokens unify how branding elements such as color palettes and typography are applied, reducing the risk of fragmentation [17].
- **Flexibility:** Modern products must be dynamic, often supporting multiple themes (light/dark), languages (LTR/RTL), and platforms (iOS, Android, Web). Design tokens enable these variations while preserving a single source of truth [2].

HISTORICAL CONTEXT AND EVOLUTION OF DESIGN TOKENS

Understanding the origin and gradual maturation of *design tokens* sheds light on why they have become integral to modern design systems. Historically, managing branding and UI consistency was a time-consuming process, often susceptible to human error and ad hoc decisions.

1. Manual Style Guides and Early Challenges

- **Static Brand Guidelines:**
 - Before tokens, many organizations relied on static PDFs or internal web pages to document brand attributes (colors, fonts, spacing).
 - These “big book” style guides were updated infrequently and could quickly become outdated [19].
 - Designers and developers had to *interpret* these guidelines, leading to inconsistent implementations and visual drift across products.

- **Decentralized Updates:**

- Any significant rebrand or UI refresh required searching through numerous codebases (web, iOS, Android) and updating color codes, font definitions, etc.
- This manual approach was prone to errors, especially for large-scale enterprises with many teams or external vendors [8].

2. Variables in CSS Preprocessors (SASS/LESS)

- **Introduction of Shared Variables:**

- CSS preprocessors like SASS and LESS introduced variables for colors, spacing, and fonts, allowing designers and developers to change a value in one place and have it update across stylesheets [15].
- This reduced duplication but still tied variables to *one technology stack* (e.g., web front-end).

- **Partial Solutions:**

- While variables improved maintainability, organizations using different tech stacks (iOS, Android, desktop apps) had to recreate these variables in each environment or convert them manually.
- There was no standardized way to propagate changes across all platforms simultaneously, so inconsistencies still arose.

3. Formalizing Design Tokens as Platform-Agnostic Primitives

- **Influence of Atomic Design:**

- Atomic Design [12] proposed thinking of UI elements as a system of interdependent parts: atoms, molecules, and organisms.
- This philosophy aligned well with the idea of isolating foundational style values (e.g., color, typography) from specific component implementations.

- **Salesforce and Adobe's Early Adoption:**

- Salesforce's Lightning Design System (SLDS) and Adobe's Spectrum were among the first to systematically define tokens as language-neutral style properties [26, 2].
- By storing them in neutral data formats (JSON, YAML), these organizations demonstrated how tokens could be transformed into any platform's styling language, ensuring brand continuity at scale.

- **Emergence of Token Tooling:**

- Tools like Style Dictionary by Amazon [3] and Theo by Salesforce [26] automated the conversion of token definitions into platform-specific formats.
- This automation further cemented the value of tokens as a single source of truth, distributed effortlessly to web, iOS, Android, and beyond.

4. Modern-Day Evolution and Widespread Adoption

- **Token-Centric Design Systems:**

- As design systems became more holistic, design tokens evolved to include not just color and typography, but also motion parameters, elevation (shadows), grid systems, and theming layers [17].
- Advanced systems now feature distinct token layers (foundation, semantic, component) that make large-scale theming or rebranding straightforward [18].

- **Community Standards and Plugins:**

- The community around tools like Figma, Sketch, and Adobe XD began developing plugins to directly manipulate and sync design tokens, bridging gaps between design and code [6].
- These plugins allow designers to update token values visually, while developers receive auto-generated variables or configuration files.

- **Context-Aware and AI-Driven Tokens:**

- Some enterprises are experimenting with "context-aware" tokens that adapt based on ambient factors (e.g., user preferences, device settings), as well as AI-driven validations for accessibility or brand compliance [10, 27].
- By integrating tokens with CI/CD pipelines, design changes can be rapidly validated, merged, and deployed, mitigating brand inconsistencies even in large, distributed organizations.

In essence, design tokens began as a solution to **platform-specific style variables** and evolved into **universal, system-wide style primitives**. They enable a more *strategic* approach to UI, linking brand and user experience goals with consistent, instantly updatable styling across myriad platforms. This historical progression—from static style

guides to dynamic, token-centric design systems—explains why so many modern organizations now rely on tokens for scalable, maintainable, and brand-consistent UIs.

DEFINING DESIGN TOKENS

1. What Are Design Tokens?

- Design tokens are *portable, platform-agnostic constants* storing design attributes, such as color, typography, spacing, borders, elevation, and motion [7, 16]. They are generally stored in neutral data formats (JSON, YAML) and transformed into platform-specific files (SCSS, Swift, XML for Android) via tools like *Style Dictionary* or *Theo* [3, 26].

2. Key Advantages

- Single Source of Truth:** Ensures that any style update (e.g., rebranding color changes) cascades through all products [18].
- Cross-platform Consistency:** Reduces the overhead of duplicating brand values for iOS, Android, and Web [11].
- Scalability:** Supports the addition of new platforms or variations with minimal refactoring [17].
- Improved Designer-Developer Collaboration:** Simplifies communication through shared terminology [9].

IMPLEMENTATION OF TOKENS

Design tokens are a systematic way of storing design properties such as colors, typography, and spacing in a structured format. They help ensure consistency across brands while enabling scalability. Let us explore the base and brand token structure and how best to implement it in a design system.

- Base Tokens:** Define core values like colors, typography, and spacing.
- Brand Tokens:** Extend base tokens to apply brand-specific styles.

1. File Structure

```
base/
├─ color.json
├─ space.json
├─ typography.json
└─ unit.json
brands/
├─ fusion/
│   ├── color.json
│   └─ typography.json
```

Figure 1 Token directory structure

The directory structure is illustrated in Figure 1.

2. Base Colors

Base colors include color palettes, which are defined in 'base/color.json':

```
{
  "color": {
    "white": {
      "50": {"value": "#ffffff"},
      "100": {"value": "#f9f9f9"},
      "200": {...},
      "300": {...},
```

```

    "400": {...},
    "500": {...},
    "600": {...},
    "700": {...},
    "800": {...},
    "900": {...}
  },
  "gray": {...},
  "black": {...},
  "berry": {...},
  "navy": {...},
  "teal": {...},
  "amber": {...},
  "fuchsia": {...}
}

```

Listing 1 Base Color Tokens defined in JSON format

3. BRAND COLORS

Brand colors use base color references:

```

{
  "background": {
    "primary": {
      "light": { "value": "{color.navy.500.value}" }
    }
  }
}

```

Listing 2 Brand Color Tokens in JSON format

4. Base Spacing Tokens

Spacing tokens play a crucial role in ensuring a consistent and scalable spacing system across a design system. These tokens define uniform values for padding, margins, and layout gaps, making it easier to maintain visual harmony across different UI components.

Below is the declared unit. json with rem values and referenced in space. json.

```

{
  "unit": {
    "o": {
      "value": "0.0625rem"
    },
  },

```

```

    "1": {
      "value": "0.125rem"
    },
    "2": {
      "value": "0.25rem"
    }
  }
}

```

Listing 3 unit.json

```

    "o": {
      "value": "{unit.o.value}"
    },
    "1": {
      "value": "{unit.1.value}"
    },
    "2": {
      "value": "{unit.2.value}"
    }
  }
}

```

Listing 4 space.json

5. Brand Spacing Tokens

Each brand may have different requirements based on its layout strategy, design aesthetics, and user experience considerations. Using base tokens, brands can establish spacing values.

```

{
  "container": {
    "small": { "value": "{spacing.2.value}" },
    "medium": { "value": "{spacing.3.value}" },
    "large": { "value": "{spacing.5.value}" }
  },
  "layout": {
    "section-gap": { "value": "{spacing.6.value}" },
    "grid-gap": { "value": "{spacing.3.value}" }
  }
}

```

Listing 5 Brand Spacing Tokens in JSON format

6. Base Typography

The typography tokens define fonts, weights, and letter spacing. Typography plays a critical role in defining a brand's visual identity. Using typography tokens, we can establish a structured and scalable system that maintains consistency across different platforms while allowing for brand-specific customization.

```
{
  "font": {
    "family": {
      "roboto": { "value": "Roboto, sans-serif" },
      "inter": { "value": "Inter, sans-serif" }
    },
    "weight": {
      "default": { "value": 400 },
      "light": { "value": 300 },
      "bold": { "value": 700 },
      "bolder": { "value": 900 }
    },
    "letter-spacing": {
      "wide": { "value": "0.05rem" },
      "dense": { "value": "-0.02rem" },
      "default": { "value": "0rem" }
    }
  },
}
```

Listing 6 Base Typography Tokens in JSON format

7. Brand Typography

While the base typography tokens define standardized values, each brand may have different typography preferences based on its branding and visual identity. Brand typography tokens map base values into brand-specific needs while maintaining consistency.

```
{
  "font-family": {
    "primary": { "value": "{font.family.roboto.value}" },
    "secondary": { "value": "{font.family.inter.value}" }
  },
  "headline": {
    "50": { "value": "{spacing.5.value}/{spacing.6.value} {font-family.primary.value}" },
    "100": { "value": "{spacing.6.value}/{spacing.7.value} {font-family.primary.value}" },
    "200": { "value": "{spacing.7.value}/{spacing.8.value} {font-family.primary.value}" }
  }
}
```

```

},
"body": {
  "50": { "value": "{spacing.5.value}/{spacing.6.value} {font-family.secondary.value}" },
  "100": { "value": "{spacing.6.value}/{spacing.7.value} {font-family.secondary.value}" },
  "200": { "value": "{spacing.7.value}/{spacing.8.value} {font-family.secondary.value}" }
},
"caption": {
  "50": { "value": "{spacing.4.value}/{spacing.5.value} {font-family.secondary.value}" },
  "100": { "value": "{spacing.5.value}/{spacing.6.value} {font-family.secondary.value}" }
}
}

```

Listing 7 Brand Typography Tokens in JSON format

8. Building through Style Dictionary

To automate the transformation of design tokens into usable formats, we use Style Dictionary [\[3\]](#), an open-source tool by Amazon. It helps in converting design tokens into **CSS variables**, **Sass variables**, **JSON**, **Android**, **iOS**, and more.

The complete open-source implementation of base and brand tokens is available in the GitHub repository [\[25\]](#).

9. Best practices for token creation

- **Start with core brand elements:** Identify brand colors, typography scales, and spacing guidelines [\[9\]](#).
- **Use Consistent Naming Conventions:** Keep names descriptive but concise: color-brand-primary, font-size-base [\[18\]](#).
- **Separate Foundation from Semantic Tokens:** Maintain flexibility by decoupling raw values from their usage context [\[7\]](#).
- **Document Thoroughly:** Provide guidelines, usage scenarios, and code snippets to reduce ambiguity [\[17, 14\]](#).
- **Iterate and Validate:** Continuous validation through design reviews, user testing, and developer feedback [\[23\]](#).

COMPARING TOKEN STRUCTURES ACROSS LEADING COMPANIES

Design tokens are broadly used to unify color, typography, spacing, and other foundational qualities across platforms. However, each leading company's approach contains nuances regarding token organization, theming, and application. This section compares five major design systems, highlighting how each manages tokens to balance consistency and flexibility.

1. Salesforce Lightning Design System (SLDS) [\[26\]](#)

- **Token Organization:** SLDS employs a dual-layer approach:
 - **Global Tokens:** Store platform-agnostic attributes for brand colors, spacing, and type scales.
 - **Component Tokens:** Reference global tokens but apply them specifically to UI elements like buttons or form fields.
- **Focus on Utility:** In addition to global and component tokens, SLDS includes “utility classes” (e.g., spacing helpers) derived from tokens, allowing rapid prototyping without manually writing CSS.
- **Adaptation:** SLDS tokens can be transformed into SCSS variables, JavaScript objects, or platform-specific outputs (iOS/Android resources) to ensure consistent branding across all of Salesforce's products.

2. IBM Carbon Design System [\[16\]](#)

- **Base vs. Thematic Tokens:** Carbon separates tokens into base (core) tokens and theme tokens (light, dark, high contrast). This structure makes it easier to swap out or extend brand or accessibility themes.
- **Strong Accessibility Emphasis:** IBM invests heavily in tokens that address color contrast and text legibility. Certain color tokens are designated for accessible backgrounds and text, ensuring a consistent baseline for WCAG compliance.
- **Dynamic Theming:** By defining separate JSON files for each theme, Carbon enables quick toggles between them, benefiting users who prefer dark mode or have specific contrast needs.

3. Google Material Design [\[13\]](#)

- **Broad Token Coverage:** Beyond color and typography, Material's tokens include *elevation*, *shape*, and *motion* definitions, reflecting the holistic nature of Google's design language.
- **Material You (Dynamic Color):** Newer versions of Material introduce dynamic color tokens that can shift based on user wallpaper or preference, showcasing how tokens can adapt to user context.
- **Customizability:** While Material provides a default token set, most teams layer their own brand identity on top, using Material's recommended color states, spacing scales, and type ramps as a starting point.

4. Atlassian Design System [\[4\]](#)

- **Hierarchical Theming:** Atlassian organizes tokens so that *global* design choices can cascade down into product-level themes. Each product (e.g., Jira, Confluence) can maintain unique details while preserving a shared visual language.
- **Emphasis on "Theming":** A robust theming layer allows designers to override base tokens with brand or product-specific values (e.g., marketing campaigns, light/dark UIs). Token references maintain consistent spacing, typography, and color usage.
- **Guided Overrides:** Detailed documentation ensures teams override tokens responsibly, preventing accidental breaks to the design language or accessibility constraints.

5. Microsoft Fluent Design System [\[11\]](#)

- **Cross-Platform Consistency:** Fluent's token strategy targets Windows, Web, iOS, and Android simultaneously. Base tokens for colors, typography, and corner radii are mapped to each platform's native styling approach.
- **Adaptive Materials:** Fluent includes tokens for materials (e.g., acrylic), elevation, and motion, reflecting Microsoft's focus on "immersive" experiences.
- **Platform Variations:** Each platform imposes unique constraints (e.g., iOS safe areas, Android densities). While some Fluent tokens vary slightly per platform, they remain logically tied to the same conceptual base, preserving brand fidelity.

ENSURING CONSISTENCY AND FLEXIBILITY: KEY CONSIDERATIONS

Even the most robust set of tokens can lead to fragmentation or misuse if there is no strategy to maintain consistency at scale while also allowing product teams the freedom to customize when necessary. The following are key considerations for teams that want to strike the right balance.

1. Governance and Maintenance

- **Versioning and Change Control:**
 - Treat tokens as a living asset within version control [\[7\]](#). Using semantic versioning (for example, major.minor.patch) signals the impact of updates on downstream products.
 - Larger design systems often employ an RFC (Request for Comments) process for major token changes to ensure transparency and facilitate cross-team feedback.
- **Approval Workflow:**
 - The formal review by the design leadership helps prevent the 'token sprawl', in which too many custom tokens undermine the original design language [\[9\]](#).

- Smaller tweaks (e.g., color adjustments for accessibility) might be automatically merged if they meet established style guardrails.

- **Documentation and Training:**

- A well-maintained token library needs clear usage guidelines, code snippets, and real examples so that design and engineering teams understand how to apply tokens consistently [\[18\]](#).
- Offering regular 'office hours' or training sessions can encourage proper use and prevent ad hoc solutions.

2. **Balancing Customization vs. Standardization**

- **Framework vs. Expression:**

- Design tokens must be rigid enough to preserve the overarching brand identity yet flexible enough to accommodate localized marketing campaigns or product quirks [\[5\]](#).
- Establish tokens as 'must follow' (for example, primary colors, typography, and scale) versus tokens 'safe to override' (for example, accent colors for holiday promotions).

- **Theming and Product Variations:**

- Large organizations often support multiple subbrands or product lines. Storing brand or product themes in discrete token files clarifies which base tokens can be extended or replaced.
- This structure also expedites the release of new product variants without reinventing the foundation each time.

- **Granular Overrides and Guardrails:**

- Some design systems adopt a tiered approach to overrides: foundational tokens are locked, while semantic tokens offer room for customization (for example, a 'call-to-action' color references a base hue but may shift slightly by product or region).
- Automated checks, such as color contrast validators, ensure that overridden tokens still comply with brand and accessibility standards.

- **Clear Communication:**

- In multibrand enterprises, consistent Slack channels or wikis help track local overrides, enabling a structured approach to exchanging overrides or ephemeral campaign changes.
- Clarity over who "owns" tokens, design leads, brand teams, or product groups prevents duplication and confusion.

3. **Accessibility Integration**

- **Proactive Color Contrast:**

- Instead of just verifying final UIs, store contrast rules within token definitions [\[28\]](#), ensuring any color pairs meet recommended accessibility levels from the outset.
- Some design teams integrate linting scripts that flag color tokens with insufficient contrast or incorrectly overridden hue references.

- **Motion and Animation Preferences:**

- "Reduced motion" user preferences can map directly to token sets controlling animation timing, easing curves, or transitions [\[13\]](#).
- By centralizing motion parameters into tokens, developers can easily respect user system settings, creating more inclusive experiences.

- **Responsive Typography:**

- Token-based typography can scale dynamically across breakpoints; for instance, font-size-lg references a fluid value rather than a fixed pixel size [\[19\]](#).
- Combined with brand tokens, each sub-brand can tweak type ratios or line heights, provided they remain consistent with core accessibility rules.

In summary, ensuring consistency means preserving a coherent, recognizable brand experience and reducing

developer overhead, while enabling flexibility recognizes that product teams often confront diverse user needs, regional constraints, or marketing campaigns. Through deliberate governance processes, layered token architectures, and built-in accessibility validations, modern design systems can achieve both aims simultaneously.

FUTURE DIRECTIONS

As design systems continue to evolve, design tokens will play a more central role in unifying brand identity with technological advances and user-centric needs. Below are key areas where token-driven strategies are likely to expand:

1. **Dynamic Theming and Context-Aware Tokens [\[10\]](#):**

- Tokens already adapt to “light” and “dark” themes, but future systems may further tailor experiences based on *contextual cues*: user preferences, ambient light, or device capabilities.
- For instance, tokens could switch to higher-contrast color sets automatically in bright sunlight or reduce motion tokens when battery levels are low.
- On a broader scale, event-specific campaigns (e.g., seasonal, holiday) could seamlessly override standard tokens for a temporary “themed” experience, reverting back without code duplication.

2. **AI-Driven Enhancements [\[27, 1\]](#):**

- Artificial intelligence and machine learning could propose or automatically validate token sets for accessibility, brand compliance, or performance.
- AI assistants might *suggest* new color ramps or spacing scales based on usage analytics, user feedback, or emerging design trends.
- In large enterprises, AI can monitor thousands of token overrides and detect potential inconsistencies or conflicts, reducing manual oversight.

3. **Deeper Collaboration Tools [\[7\]](#):**

- Real-time integration between design tools (e.g., Figma, Sketch) and development environments (Git, CI/CD) is improving. Automated design tokens’ pipelines could allow designers to *edit tokens in Figma* and push changes directly to a code repository.
- Such toolchains, combined with version control hooks, will further unify design and development workflows, making token changes fully auditable and easily reversible.
- This can also foster immediate prototyping—designers and developers see token adjustments in near real time, cutting feedback loops.

4. **Global Accessibility and Inclusivity:**

- Beyond color contrast and motion reduction, design tokens can encode advanced accessibility attributes, such as explicit references to text-to-speech or language direction (LTR vs. RTL).
- Future accessibility guidelines (e.g., WCAG 3.0) may require context-aware tokens that adapt to user-assistive technologies, ensuring *all* user segments receive an equitable experience [\[24\]](#).
- Unified accessibility tokens (e.g., for color-blind safe palettes) could become standard practice to prevent inadvertent barriers in new product launches.

5. **Integration with Next-Generation Platforms:**

- As ****XR (Extended Reality)**** and ****Voice UIs**** gain traction, design tokens might expand beyond visuals into audio or haptic feedback properties (e.g., “button-press-vibration-intensity” tokens).
- In voice or chat-based experiences, tokens’ tone of voice could define the personality of the brand, ensuring a consistent language style across multiple platforms.

Overall, design tokens’ core philosophy—*a single source of truth for foundational and semantic*

Design decisions are poised to extend far beyond current use cases. From ambient or context-driven

Theming to AI-assisted maintenance, the future of token-driven design promises deeper automation. stronger inclusivity and unparalleled flexibility in crafting modern user experiences.

CONCLUSION

Style tokens have revolutionized how teams structure and maintain design systems, offering a clear path to consistency and flexibility in product design. By encoding branding and experiential values into foundational, semantic, and component-level tokens, organizations can rapidly scale across technologies while preserving the fidelity of their brand [12, 9].

Across companies like Salesforce, IBM, Google, Atlassian, and Microsoft, token-based approaches enable uniform design, reduce rework, and create cohesive user experiences. As design and development processes evolve—especially with advanced theming and AI-driven adaptations—tokens will remain integral to the future of systematic, maintainable, and user-centric design [7].

REFERENCES

- [1] T. Achurch and Z. Wang. *The Future of AI-Driven Design Systems*. InVision Press, 2019.
- [2] Adobe. Adobe Spectrum. <https://spectrum.adobe.com/>.
- [3] Amazon. Style dictionary. <https://amzn.github.io/style-dictionary/>.
- [4] Atlassian. Atlassian design system. <https://atlassian.design/>.
- [5] Atlassian. Themeable tokens: Building flexible foundations. Atlassian Design Blog.
- [6] D. Brown. Design tokens for better collaboration in product teams. A Book Apart, 2021.
- [7] J. Couldwell. *Laying the Foundations: A Book about Design Systems*. Self-published, 2019.
- [8] Nathan Curtis. Building a living design system in the enterprise. An Event Apart, 2015.
- [9] Nathan Curtis. Design system thinking: Creating and maintaining a system that serves people. Medium, 2017.
- [10] Nathan Curtis. Evolving tokens: The next generation of design systems. Medium, 2020.
- [11] Fluent. Microsoft Fluent Design System. <https://developer.microsoft.com/en-us/fluentui>.
- [12] Brad Frost. *Atomic Design*. CreateSpace Independent Publishing Platform, 2016.
- [13] Google. Material design guidelines. <https://material.io/design>.
- [14] Nielsen Norman Group. Design systems 101. <https://www.nngroup.com/articles/design-systems-101/>, 2017.
- [15] C. Hendrix. *Sass and Less for Web Developers*. O'Reilly Media, 2016.
- [16] IBM. Carbon design system. <https://www.carbondesignsystem.com/>.
- [17] InVision. Design systems handbook. <https://www.invisionapp.com/design-system-manager>, 2019.
- [18] Alla Kholmatova. *Design Systems: A Practical Guide to Creating Design Languages for Digital Products*. Smashing Magazine, 2017.
- [19] S. J. Koyani, R. W. Bailey, and J. R. Nall. *Research-Based Web Design & Usability Guidelines*. U.S. Department of Health and Human Services, 2003.
- [20] G. Larson. Scaling design systems with CI/CD. O'Reilly Media, 2020.
- [21] K. Lindstedt and V. Zimmermann. Maintaining consistency in distributed design teams. *ACM Interactions*, 27(5):48–51, 2020.
- [22] Jakob Nielsen and Raluca Budi. *Mobile Usability*. New Riders, 2013.
- [23] Jakob Nielsen and Hoa Loranger. *Prioritizing Web Usability*. New Riders, 2006.
- [24] Heydon Pickering. *Inclusive Components: The Book*. Smashing Magazine, 2019.
- [25] Rushil Patel. Style Tokens—Open Source Design Tokens, 2024.
- [26] Salesforce Lightning Design System. <https://www.lightningdesignsystem.com/>.
- [27] Shira Tomer. Design tokens and AI: Automating style consistency. UX Collective, 2021.
- [28] W3C. Web Content Accessibility Guidelines (WCAG) 2.1. <https://www.w3.org/TR/WCAG21/>, 2018.