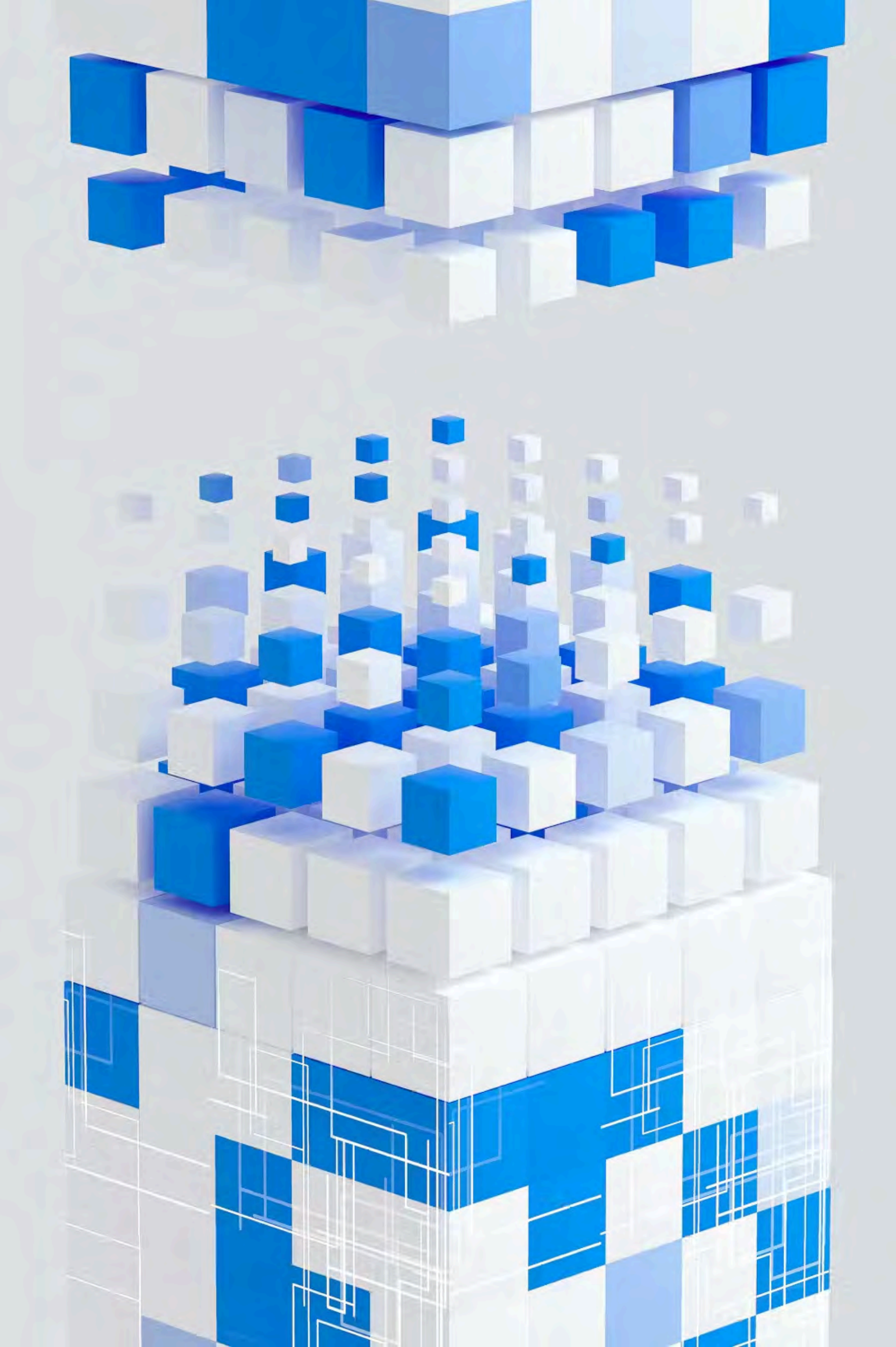


Design Tokens as Platform Infrastructure

Architecting Scalable Multi-Platform Design Systems

A framework for transforming visual design decisions into programmable infrastructure across digital products

Pallav Laskar

Zscaler

The Big Problem – Fragmentation

Style guides drift across platforms

Different implementations lead to inconsistent brand experiences

Inconsistent user experiences

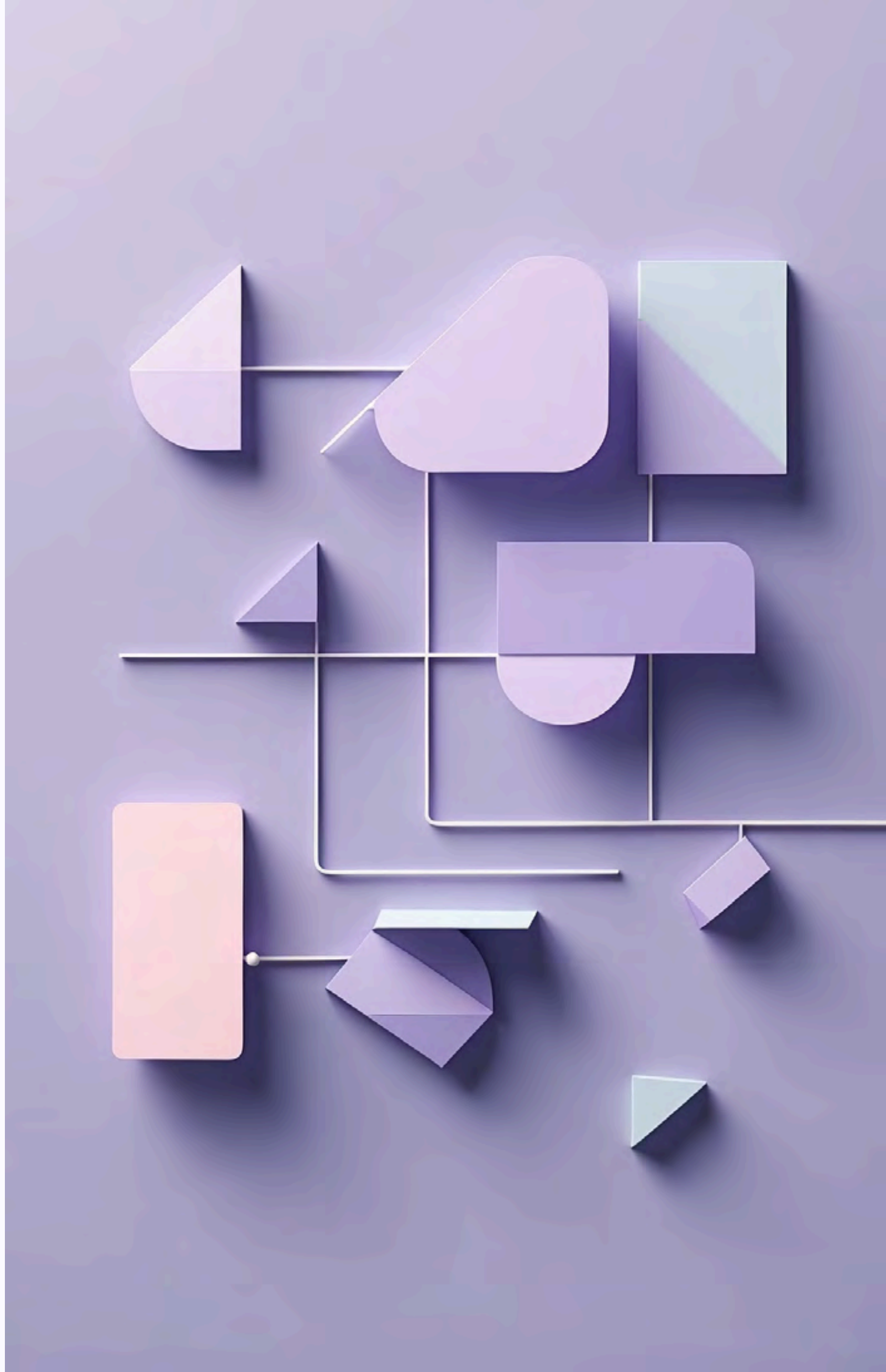
Users encounter different behaviors and visuals across touchpoints

Siloed implementations slow teams

Duplicated work and communication overhead

Manual design → technical debt

Hard-coded values create maintenance challenges



Why This Matters

Customers

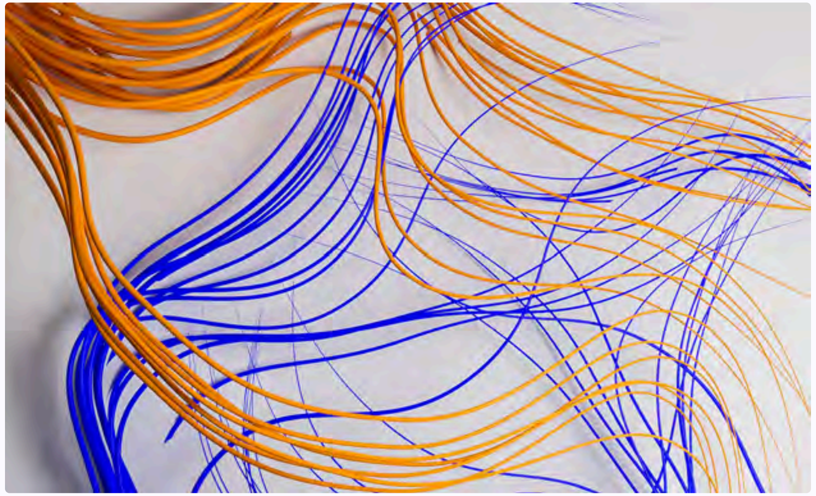
Expect seamless, branded experiences across all touchpoints

Engineering

Needs automation + governance to reduce overhead and errors

Designers

Want agility without manual implementation overhead



Without alignment, all three stakeholders suffer from fragmentation issues.

Paradigm Shift

Tokens as Infrastructure

From static values → programmable infrastructure

Design decisions become code-driven and systematic

Every design choice is versionable + reusable

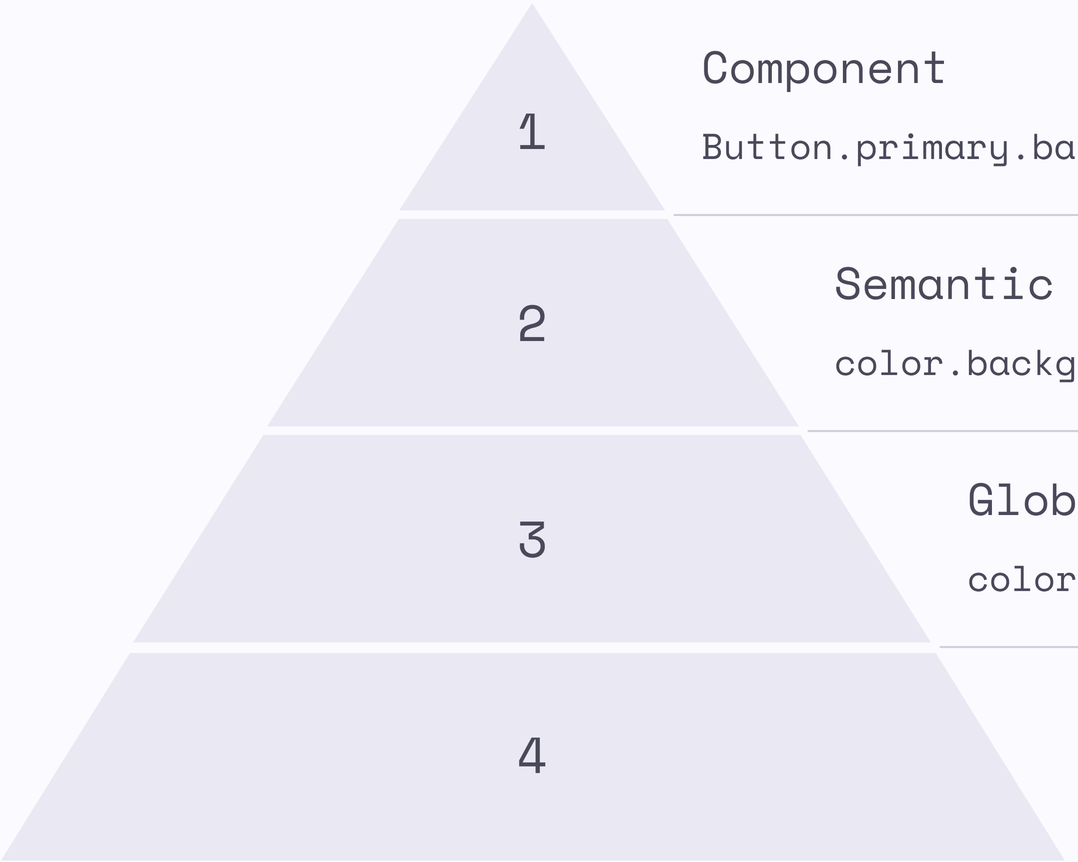
GitOps principles applied to design decisions

Tokens align with platform engineering principles

Self-service, automation, scalability

One source, many outputs

Token Architecture Hierarchy



1

Component

Button.primary.background

2

Semantic

color.background.interactive

3

Global

color.blue.500

4

Raw

#0074D9

Progressive abstraction = flexibility + consistency

One source of truth across platforms

Orchestration Pipeline

Source

JSON / YAML in Git

Version controlled design decisions

Transform

Style Dictionary / custom tooling

Automated build processes

Targets

CSS vars, TypeScript, Swift, Android, React Native

Platform-specific implementations

Theme Composition

Multi-brand identity

No duplication of token structures

Light & dark modes

Core semantic swap only

Accessibility themes

WCAG-ready contrast ratios

Define once, theme infinitely

Case Study — Zscaler

5

Repos Consolidated

Unified source of truth

240kB

Bundle Reduction

Smaller, faster payloads

18%

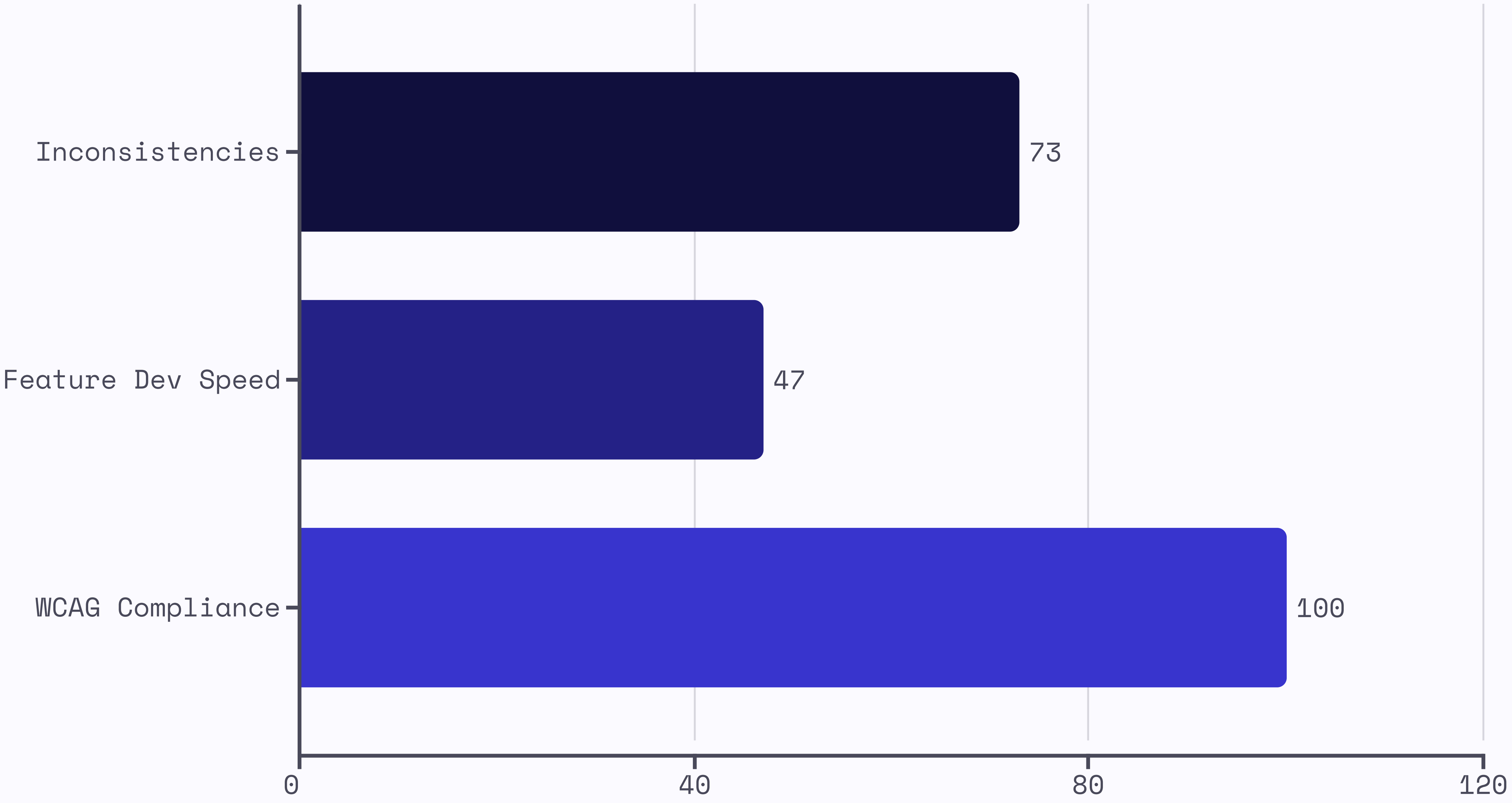
Faster FCP

Improved performance

Developer Experience Improvements:

- GitOps workflows
- Type safety
- Automated documentation

Case Study — Fortune 500 Rollout



✓ **7 product lines unified** across 3 acquisitions with a single token system

The token infrastructure created a common design language while preserving brand uniqueness, enabling faster development with fewer inconsistencies.

Governance & Compliance

1

JSON Schema Validation

Enforces structure standards in CI/CD

2

Automated Contrast Testing

Ensures WCAG compliance for all color combinations

3

Cross-Functional PR Approvals

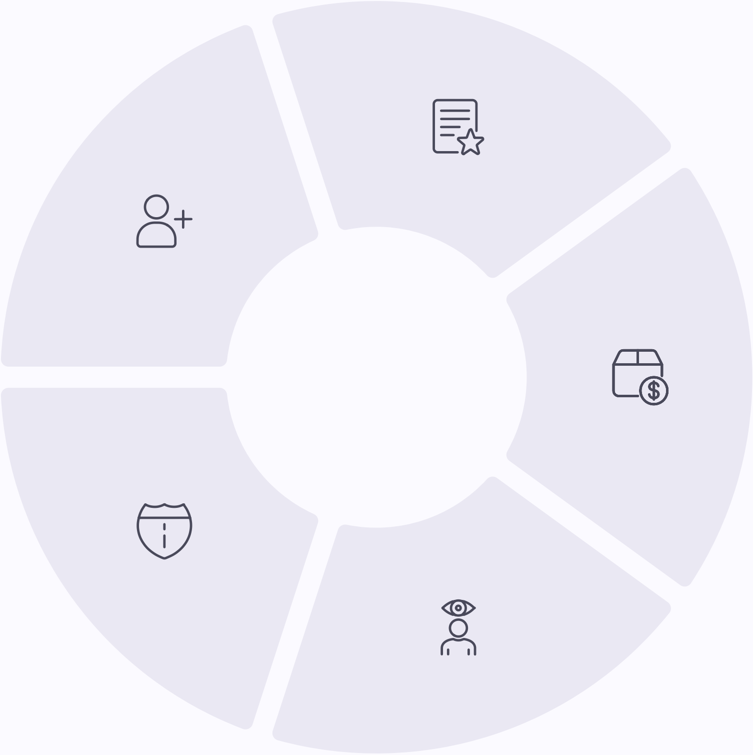
Design + Engineering alignment on changes

4

Auto-Generated Documentation

Always up-to-date reference for teams

Token Lifecycle



Creation & Definition

Define new tokens with proper metadata

Review & Approval

Cross-functional validation

Distribution & Usage

Package publishing and implementation

Monitoring & Analytics

Track usage and consistency

Deprecation & Migration

Structured evolution without breaking changes

A formal lifecycle balances innovation with stability

Developer Experience

Before

- Hard-coded values
- Scattered definitions
- Inconsistent naming
- No type safety

After

- Type-safe tokens
- Autocomplete in IDE
- Single source of truth
- Enforced consistency

Adoption Challenges

Resistance to Change

Teams hesitant to adopt new workflows

Solution:

Codemods + automation to reduce migration effort

Legacy Integration

Existing codebases with established patterns

Solution:

Phased rollout with backward compatibility

Governance Risk

Concerns about overhead and bureaucracy

Solution:

Schema validation and automated testing

Scale Concerns

Questions about performance at enterprise scale

Solution:

Proven case studies and performance benchmarks

Implementation Roadmap

Phase 1: Audit + Standards

Catalog existing design values and establish naming conventions

Phase 2: Foundation Repo + Governance

Create base token structure and review processes

Phase 3: Pilot 1 Product Line

Implement with one team to validate approach

Phase 4: Scale + Automate Pipelines

Expand adoption and build CI/CD infrastructure

Phase 5: Continuous Improvement

Refine based on metrics and user feedback

Key Takeaways

Tokens = Infrastructure

Not just styling, but programmable platform foundation

Single Source of Truth

Unified definitions across all platforms

Automated Compliance

Built-in accessibility and theming capabilities

Measurable Gains

Real performance and efficiency improvements

Design tokens transform from static values to system infrastructure, enabling true scalability

Call to Action

- Assess design system maturity

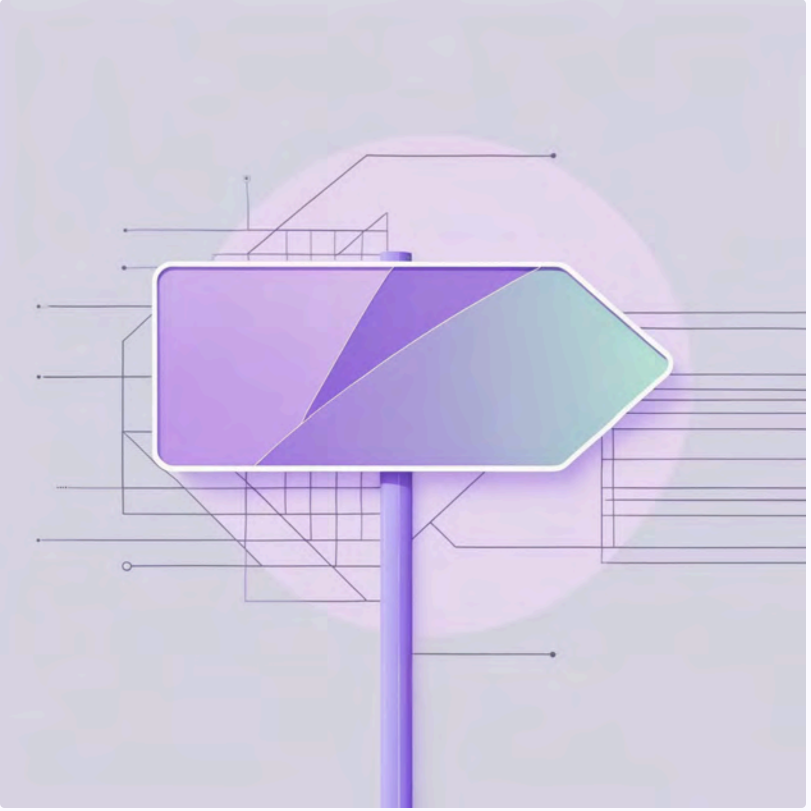
  Evaluate your current state and identify fragmentation points

- Pilot token adoption today

  Start small with one component or feature area

- Build governance + scale for the future

  Plan for enterprise-wide adoption with proper tooling



> "Design tokens are the API contract between design and development."