

Cómo liberar todo un sistema de diseño corporativo

How to release an entire corporate design system

2023-05-12 — [esLibre](#)

Antonio Olmo (aka *tripu*)



Content

1. [About us](#)
2. [Before a design system](#)
3. [A design... what?](#)
4. [The long road](#)
5. [After a design system](#)
6. [Pitfalls, solutions and \(free\) advice](#)
7. [Q&A](#)



1. About us



The only **cloud-native security analytics platform** that combines the power of people and intelligent automation to confidently defend expanding attack surfaces.

QuVis (Quantitative Visualization) squad.
We develop the Genesys ecosystem.
We take care of Devo's data visualization system.

2. Before a design system

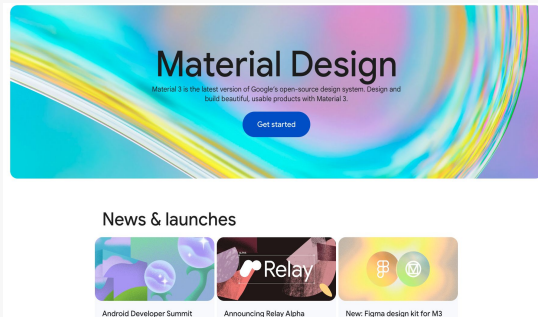
Design-wise, traditionally we had:

- **Redundancy** in the code of projects (markup, CSS, images)
- **Inconsistencies** in our look & feel
- **Outdated branding**
- **Poor A11Y**
- **Amateurish UIs**
- Just **one** internal project to share look & feel

To fix that at Devo, months ago we started investing a lot of effort into...

3. “A design... what?”

“A design system is a **set of interconnected patterns and shared practices coherently organized.**” — [Wikipedia](#)



[Material Design System](#)

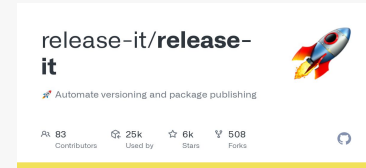
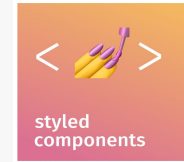
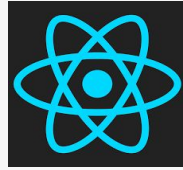


[Carbon Design System](#)

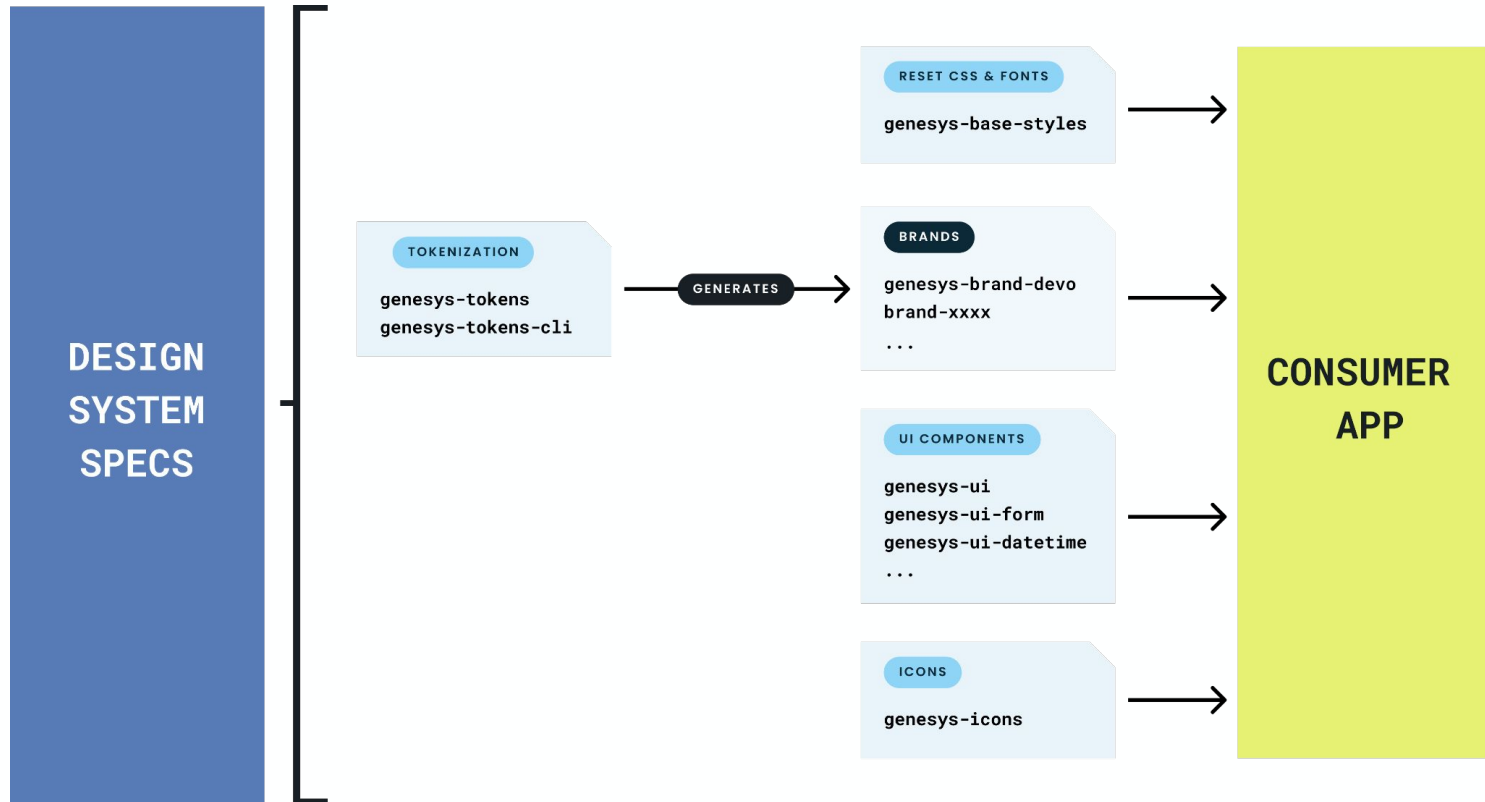
4. The long road: necessary changes

- Split single repo into **multiple monorepos**
- Licences
- Documentation
- Automatic publishing to **npmjs.com**
- Adopt **types**
- Rethink **A11Y**
- Migrate from private GitLab to **public GitHub**
- Node.js 18 + npm 8
- GitHub Actions
- Storybook 7
- Improve tokens preview
- Add **dark theme** support
- Semi-automatic **brand generation**
- Introduce new UI components

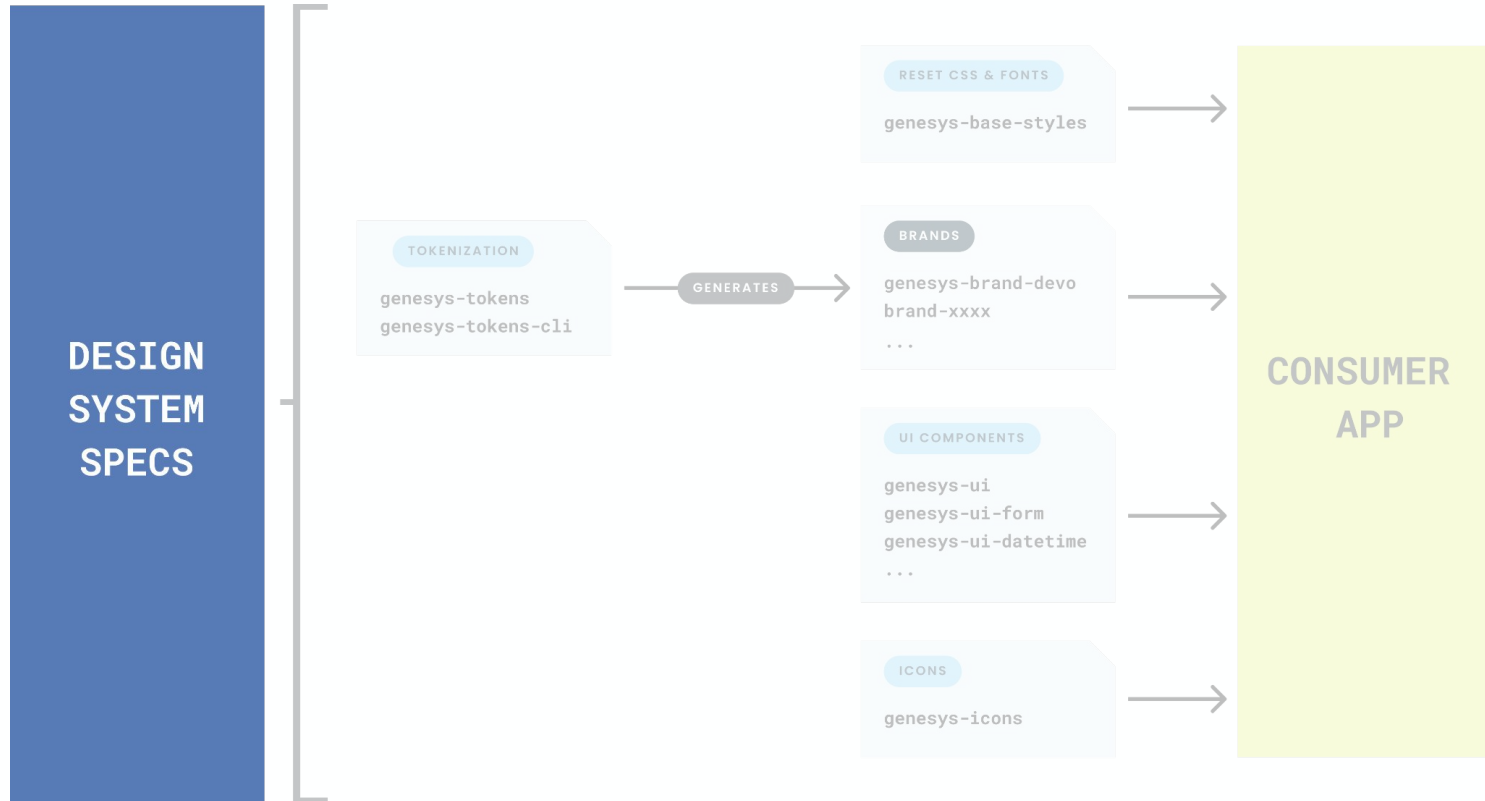
4. The long road: renewed toolkit



5. After a design system: building blocks



5. After a design system: building blocks



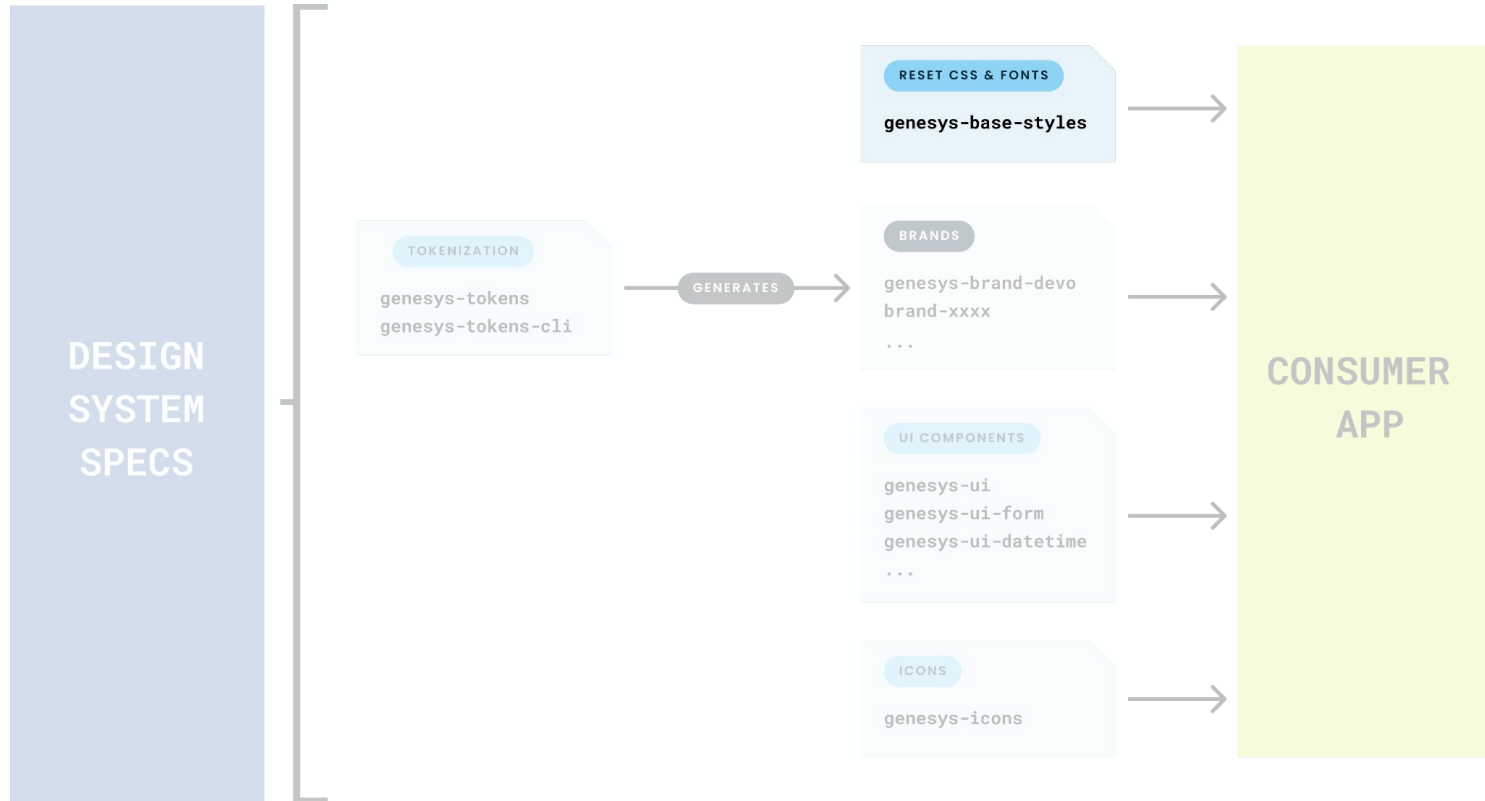
5. After a design system: building blocks

DESIGN SYSTEM SPECIFICATIONS

Tech agnostic design rules.
They provide the core to Devo's look & feel.

To be released...

5. After a design system: building blocks



5. After a design system: building blocks

genesys-base-styles

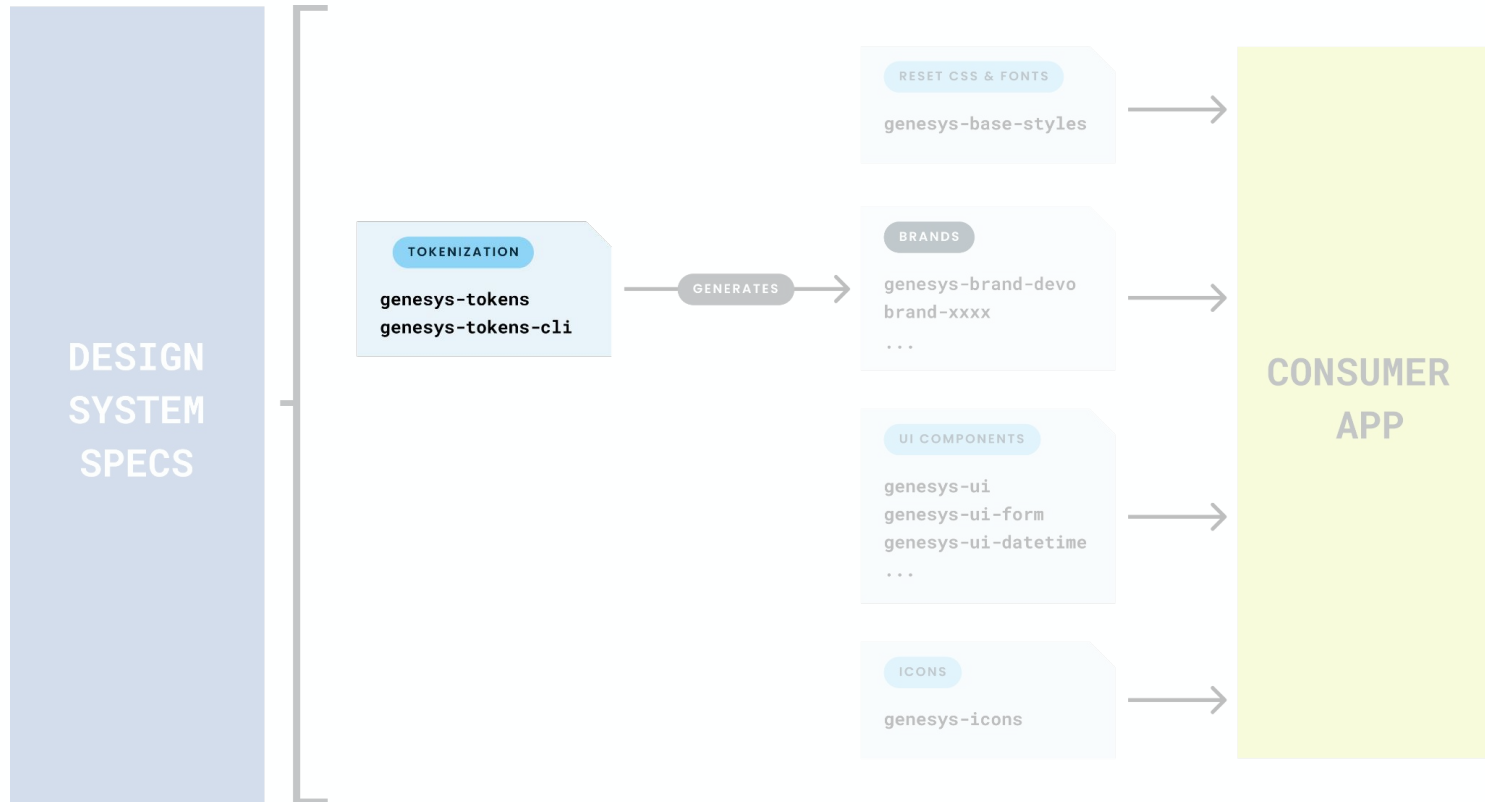
Set of CSS styles that provide a base for Genesys UI components. It includes the CSS reset, font family and font size, among others.

GitHub: [DevoInc/genesys-base-styles](https://github.com/DevoInc/genesys-base-styles)

Storybook: -

NPM package: [@devoinc/genesys-base-styles](https://www.npmjs.com/package/@devoinc/genesys-base-styles)

5. After a design system: building blocks



5. After a design system: building blocks

genesys-tokens & genesys-tokens-cli

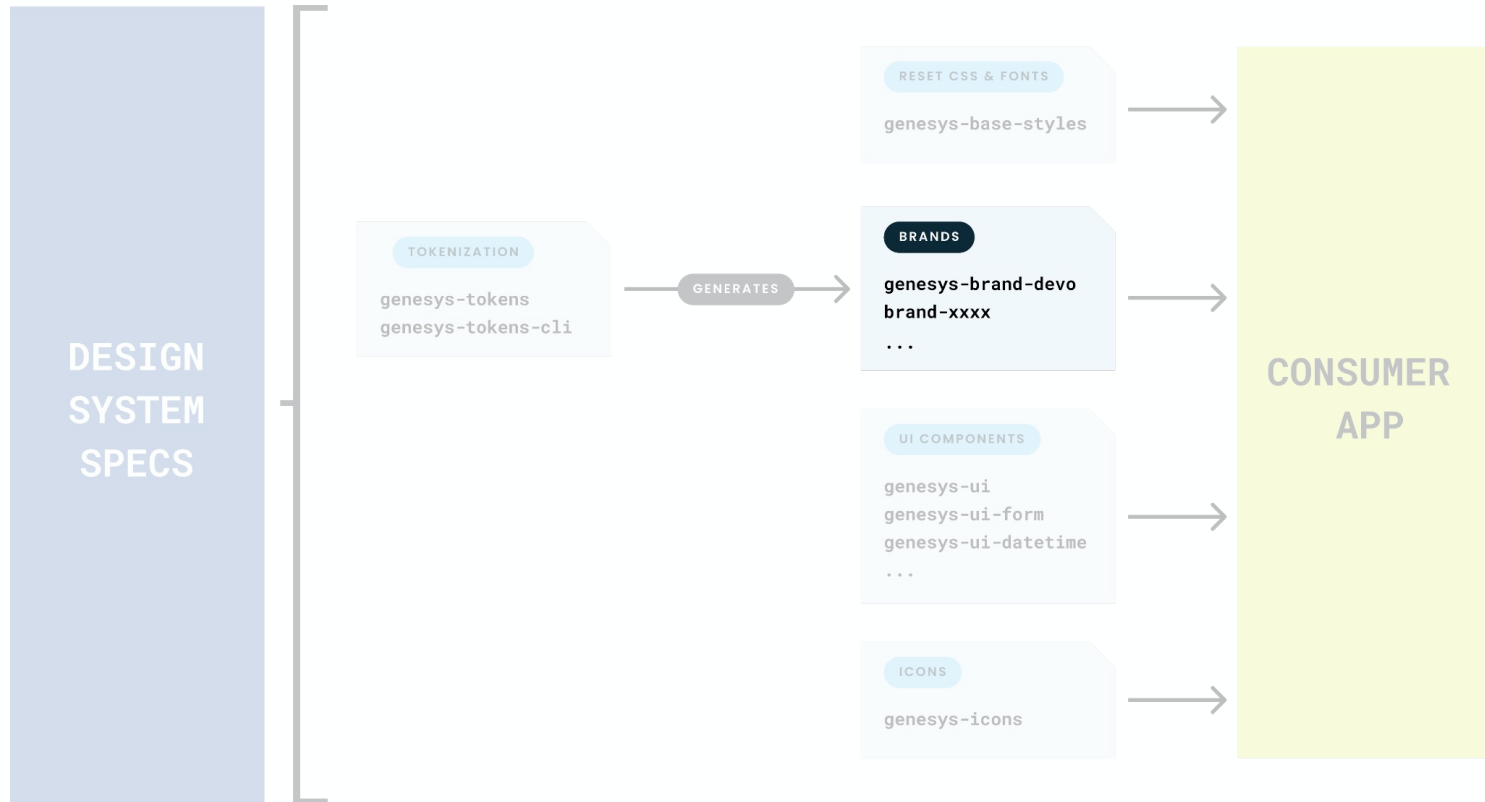
Sources and tools for the generation of design tokens

GitHub: [DevoInc/genesys-tokens](https://github.com/DevoInc/genesys-tokens)

Storybook: -

NPM packages: [@devoinc/genesys-tokens](https://www.npmjs.com/package/@devoinc/genesys-tokens), [@devoinc/genesys-tokens-cli](https://www.npmjs.com/package/@devoinc/genesys-tokens-cli)

5. After a design system: building blocks



5. After a design system: building blocks

genesys-brand-devo

Devo's main brand schemes of design tokens defined by Genesys Design System.

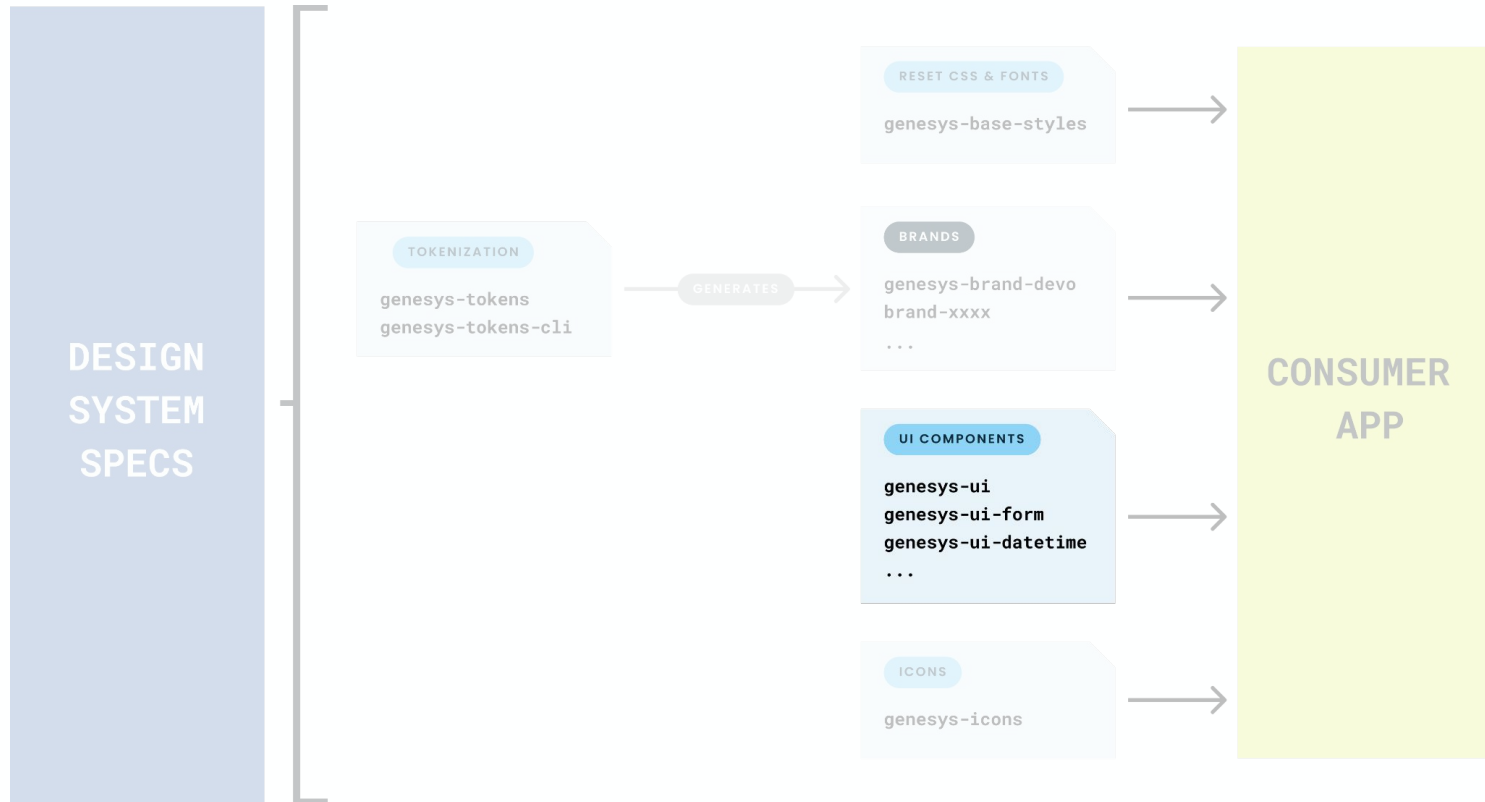
This package contains the brand schemes required to generate
Devo's light and dark brands.

GitHub: [DevoInc/genesys-brand-devo](https://github.com/DevoInc/genesys-brand-devo)

Storybook: devoinc.github.io/genesys-tokens/

NPM packages: [@devoinc/genesys-brand-devo](https://www.npmjs.com/@devoinc/genesys-brand-devo)

5. After a design system: building blocks



5. After a design system: building blocks

genesys-ui, genesys-ui-form, genesys-ui-datetime, ...

Library of **React** UI components that implements **Genesys Design Guidelines**.
It is a collection of components that can be used to build React applications.

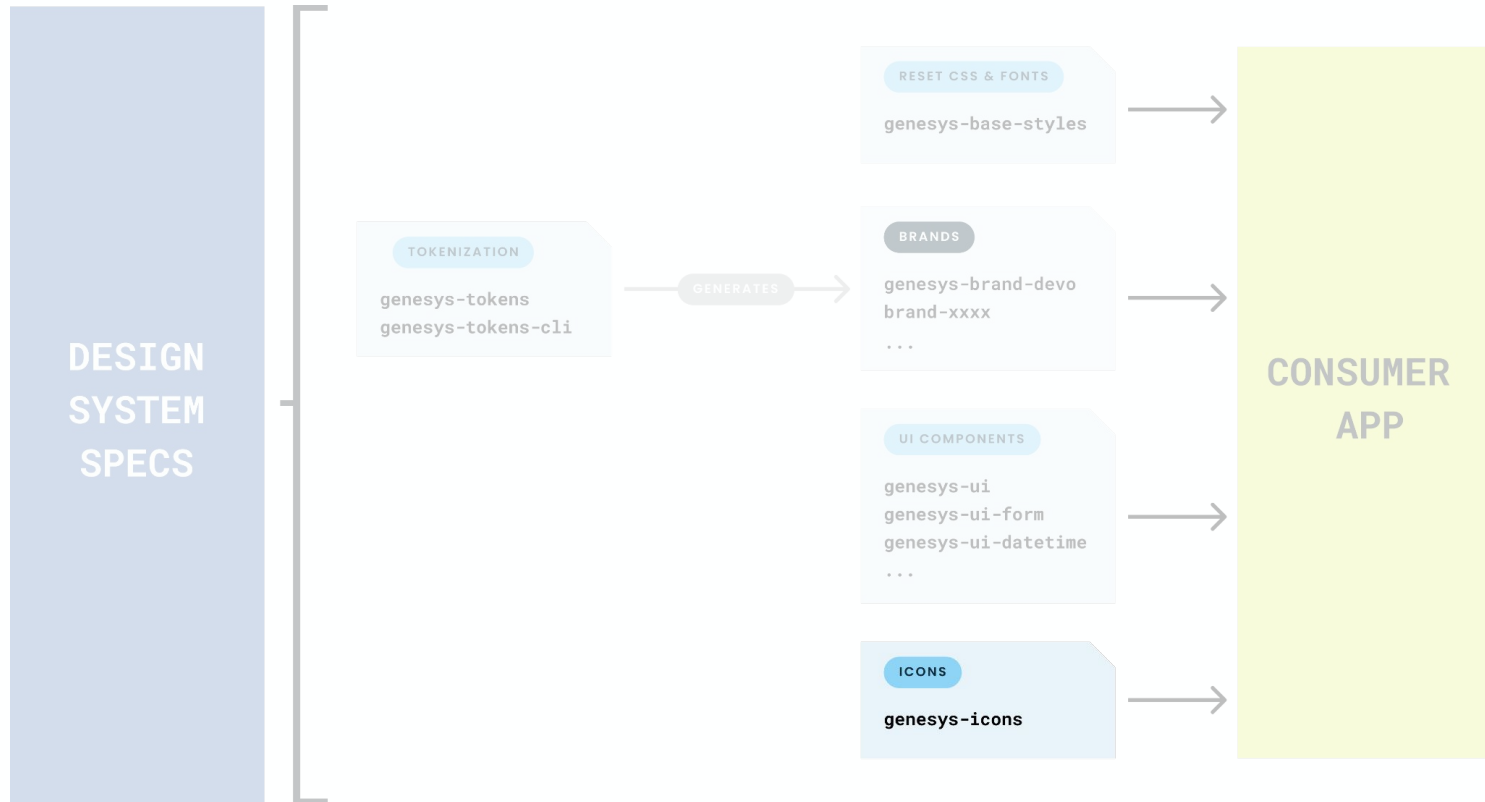
Genesys UI is designed to be **flexible**, **consistent**, and **accessible**.

GitHub: [DevoInc/genesys-ui](https://github.com/DevoInc/genesys-ui)

Storybook: devoinc.github.io/genesys-ui

NPM packages: [@devoinc/genesys-ui](https://www.npmjs.com/package/@devoinc/genesys-ui), [@devoinc/genesys-ui-form](https://www.npmjs.com/package/@devoinc/genesys-ui-form), [@devoinc/genesys-ui-datetime](https://www.npmjs.com/package/@devoinc/genesys-ui-datetime), ...

5. After a design system: building blocks



5. After a design system: building blocks

genesys-icons

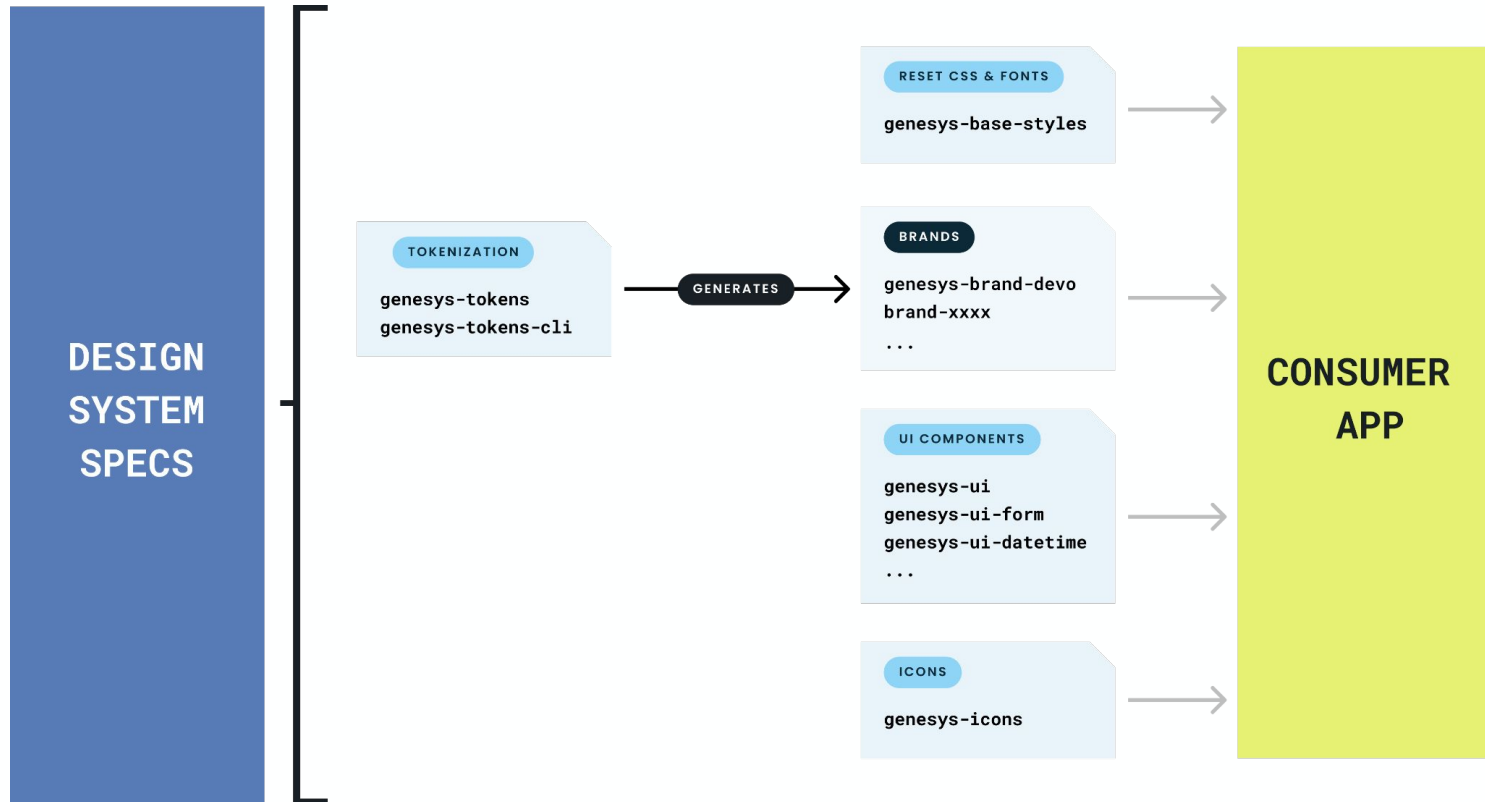
Icon library that provides a set of icons to be used in Genesys UI.
The icons are available both as a React component and as a font.

GitHub: [DevoInc/genesys-icons](https://github.com/DevoInc/genesys-icons)

Storybook: devoinc.github.io/genesys-icons

NPM package: [@devoinc/genesys-icons](https://www.npmjs.com/package/@devoinc/genesys-icons)

5. After a design system: building blocks



5. After a design system: building blocks

genesys-ui-demo-app

Simple consumer for genesys-ui.

Serves as a template for understanding how to integrate genesys-ui into another project.

GitHub: [DevoInc/genesys-ui-demo-app](https://github.com/DevoInc/genesys-ui-demo-app)

Storybook: devoinc.github.io/genesys-ui-demo-app/

NPM package: [@devoinc/genesys-ui-demo-app](https://www.npmjs.com/package/@devoinc/genesys-ui-demo-app)

5. After a design system: advantages

How have we improved?

- **Scalability and maintainability** - Allows easy updates of styles and behaviours across the platform. Enables teams to adapt to changing requirements, iterate quickly, and scale the product or application without sacrificing consistency in the user interface
- **Consistent behavior and look and feel** - Set of consistent UI components, patterns, and guidelines that ensure a cohesive and unified look and feel across different parts of the application or product.
- **Accessibility** - Follow established accessibility guidelines and best practices that are consistently applied across all UI components and patterns.
- **Less extra work** - Development teams can reuse pre-designed UI components and patterns, reducing duplication of effort and saving time and resources.
- **Brand identity** - Ensures that the UI components and patterns used in the application or product are aligned with the brand and identity of the company or organization.

6. Pitfalls, solutions and (free) advice

What have we learned?

- **Strategy:**
 - Liberating an internal private set of projects is challenging: sensitive secrets, intellectual property, poor documentation, ad-hoc contraptions, dependencies with other inner projects...
 - It's better to start from new empty repos, and move things along
 - Managing expectations is hard
 - Transition phases can be painful
- **Tools:**
 - We loved our toolchain; in particular TypeScript and Storybook
 - Vite > Webpack
...although Vite comes with its own drawbacks: linking, ...
 - Monorepos and npm workspaces FTW



comments, questions?

THANK YOU