Design Rationale – Keuze voor Svelte Stores

Voor Sonora, een webapplicatie die live Spotify-data visualiseert, was 1 technische keuze bepalend voor de hele architectuur: het gebruik van Svelte stores voor state-management. Deze rationale bespreekt deze technische keuze in detail

1. Probleemstelling

De applicatie bestaat uit meerdere pagina's en visualisaties (vinyl, treemap, recap), die allemaal dezelfde informatie nodig hebben: de huidige track (live elke seconde) en de tracks van de sessie. Deze data moet project-breed beschikbaar zijn en automatisch up-to-date blijven. Elke component afzonderlijk API-calls laten doen leidt tot inconsistentie, extra netwerk genriol en onnodig complexe logica.

2. Onderzochte alternatieven

Ik heb drie opties onderzocht:

1. Module-variabelen: eenvoudige opslag, maar geen reactiviteit. Componenten weten niet automatisch dat data geüpdatet is.

2. Stores: waren een van de onderzochte opties voordat ze de uiteindelijke keuze werden. De voordelen waren direct duidelijk: ingebouwde reactiviteit met $store, direct bruikbaar in meerdere routes, duidelijke scheiding tussen data en UI.

3. Losse API-calls per component: dubbele calls, verschillende timings, inconsistentie tussen componenten.

4. Context API: bruikbaar, maar minder geschikt voor app-brede state en niet idiomatisch voor Svelte. Ik heb deze niet uitgebreid onderzocht.

3. Gekozen oplossing: Svelte Stores

Svelte stores bieden:

- centrale opslag van state die in de hele app beschikbaar is;

- automatische reactiviteit ($store), waardoor visualisaties direct updaten;

- duidelijke scheiding tussen data-logica (stores) en UI (components);

Ik gebruik twee stores:

1. currentSong (readable): pollt elke seconde de Spotify API, haalt data (genre, kleuren, album) en wordt door vinyl, treemap en de UI automatisch geüpdatet.

2. sessionStore (writable): houdt recording-status en volledige track-snapshots bij De recap-pagina kan zonder extra API-calls renderen.

4. Resultaat

De stores zorgen voor:

- 1 single source of truth,

- veel minder DRY (herhaaldelijke) code,

- consistente data over alle routes,

- soepele real-time updates in visualisaties (D3 + Svelte),

- makkelijke uitbreidbaarheid.

5. Conclusie

Na het vergelijken van alternatieven is de keuze voor Svelte stores het meest logisch: ze geven precies het gedrag dat de app nodig heeft: centrale, reactieve, onderhoudbare en efficiënte states kunnen bijhouden voor een applicatie die volledig draait op live data en meerdere visualisaties.