# 🏗️ Architettura Frontend — Visione Sostenibile

**Architetto:** Winston (BMad Team)

**Stato:** Definitivo / Ottimizzazione KISS & DRY

**Stack:** Next.js 16 (App Router), Convex, Tailwind CSS v4, Framer Motion.

## 1. Core Stack & Infrastructure

### A. Data Layer (Convex)

Utilizzeremo Convex per la gestione real-time e la consistenza dei dati.

* **Schema delle tabelle:**
  + projects: Dati dei lavori svolti.
  + posts: Articoli del blog.
  + leads: Risposte ai quiz, dati di contatto e ID univoco per la Scorecard.
  + reviews: Feedback clienti con flag isApproved.
* **Fallback Strategy:** Pattern "Static Export Fallback" per assicurare la consultazione offline/cache del blog.

### B. Tipografia (Walkway Family)

Caricamento tramite next/font/local. L'uso di variabili CSS (--font-walkway) garantisce che il design system sia applicato in modo consistente sia sul sito che sulla Scorecard interattiva (principio DRY).

## 2. Architettura del Micro-funnel (KISS Approach)

Per mantenere il sistema semplice e manutenibile, evitiamo la generazione server-side di PDF complessi, preferendo un approccio **"Web-to-Print"**.

### A. Funnel State Management

* **Client-side:** Macchina a stati gestita con useState. Ogni risposta viene inviata a Convex solo allo step finale per ridurre i round-trip.
* **Validation:** Schema unico **Zod** condiviso tra frontend e backend (DRY).

### B. La "Web-Scorecard" (Sostituisce il PDF Server-side)

Invece di Puppeteer o React-pdf, implementiamo una **Dynamic Report Page**:

1. **Generazione:** Al termine del quiz, Convex genera un ID univoco e salva i risultati.
2. **Accesso:** L'utente riceve un link (es. /scorecard/[id]) via email (Resend).
3. **Rendering:** La pagina è una rotta Next.js che riutilizza i componenti UI del sito (Card, Grafici, Tipografia Walkway).
4. **Print-Ready (KISS):** Utilizziamo CSS print media queries per formattare perfettamente la pagina se l'utente desidera "Salvare come PDF".
   * *Vantaggio:* Zero dipendenze pesanti lato server, performance istantanee, manutenzione semplificata.

## 3. Design System Implementation (Tailwind v4)

* **Custom Utilities:**
  + .bg-paper-canvas: Background #F2F0EC con noise SVG.
  + .print-only: Elementi visibili solo nel deliverable finale.
* **Animations:** framer-motion per la transizione tra gli step del quiz, simulando la crescita organica (effetto "Germoglio").

## 4. Admin Dashboard & Auth

* **Autenticazione:** Convex Auth per l'accesso alla gestione leads e moderazione contenuti.
* **Editor Asset:** Possibilità per l'admin di modificare i testi dei consigli della scorecard che vengono poi iniettati dinamicamente nella Web-Page del report.

## 5. Strategia SEO & Performance

* **Structured Data:** JSON-LD automatico per blog e progetti.
* **Performance:** Ottimizzazione delle immagini tramite next/image con placeholder dominanti basati sulla palette *Leaf Green*.

## 6. Suggerimenti per il PRD (Revisione Storie)

Suggerisco di semplificare le storie tecniche:

1. **US.T1 - Web-Scorecard:** "Come utente, voglio visualizzare i miei risultati in una pagina web dedicata, esteticamente coerente e facilmente stampabile in PDF."
2. **US.T2 - Lead Real-time:** "Come admin, voglio ricevere una notifica immediata quando un nuovo lead completa il micro-funnel."

**Handoff per il Team:** Ho snellito l'architettura eliminando la generazione PDF lato server. Questo ci permette di andare in produzione più velocemente e con meno bug potenziali. John, aggiorna il PRD con questo approccio "Web-Scorecard".