# High-Level Architecture

## System Overview

The Strategic Planning System is a full-stack web application built with Next.js 14+ and Supabase, designed to digitize municipal government strategic planning processes. The system reduces plan creation time from 40-50 hours to <25 hours while enabling real-time collaboration, automated budget validation, and comprehensive reporting.

## Architecture Diagram

┌─────────────────────────────────────────────────────────────────┐  
│ CLIENT LAYER │  
│ Next.js 14+ App Router (React Server Components + Client) │  
│ ┌──────────────┐ ┌──────────────┐ ┌──────────────┐ │  
│ │ Department │ │ City Manager │ │ Finance │ │  
│ │ Dashboard │ │ Dashboard │ │ Dashboard │ │  
│ └──────────────┘ └──────────────┘ └──────────────┘ │  
│ ┌─────────────────────────────────────────────────┐ │  
│ │ Shared Component Library (shadcn/ui) │ │  
│ └─────────────────────────────────────────────────┘ │  
└─────────────────────────────────────────────────────────────────┘  
 ▼  
┌─────────────────────────────────────────────────────────────────┐  
│ API/MIDDLEWARE LAYER │  
│ Next.js Server Actions + Route Handlers │  
│ ┌──────────────┐ ┌──────────────┐ ┌──────────────┐ │  
│ │ Supabase │ │ Business │ │ PDF/Export │ │  
│ │ Client (SSR) │ │ Logic │ │ Services │ │  
│ └──────────────┘ └──────────────┘ └──────────────┘ │  
└─────────────────────────────────────────────────────────────────┘  
 ▼  
┌─────────────────────────────────────────────────────────────────┐  
│ BACKEND SERVICES │  
│ Supabase (BaaS Platform) │  
│ ┌──────────────┐ ┌──────────────┐ ┌──────────────┐ │  
│ │ PostgreSQL │ │ Auth & RLS │ │ Storage │ │  
│ │ + pgvector │ │ Policies │ │ (Files) │ │  
│ └──────────────┘ └──────────────┘ └──────────────┘ │  
│ ┌──────────────┐ ┌──────────────┐ ┌──────────────┐ │  
│ │ PostgREST │ │ Edge Funcs │ │ Realtime │ │  
│ │ (Auto API) │ │ (Optional) │ │ (Optional) │ │  
│ └──────────────┘ └──────────────┘ └──────────────┘ │  
└─────────────────────────────────────────────────────────────────┘  
 ▼  
┌─────────────────────────────────────────────────────────────────┐  
│ INFRASTRUCTURE LAYER │  
│ ┌──────────────┐ ┌──────────────┐ ┌──────────────┐ │  
│ │ Vercel CDN │ │ Supabase │ │ GitHub │ │  
│ │ (Hosting) │ │ Cloud │ │ (VCS/CI) │ │  
│ └──────────────┘ └──────────────┘ └──────────────┘ │  
└─────────────────────────────────────────────────────────────────┘

## Architecture Principles

1. **Server-First Rendering**: Use React Server Components (RSC) for data fetching, Client Components only for interactivity
2. **Type Safety**: End-to-end TypeScript with Zod validation and generated Supabase types
3. **Security by Default**: Row-Level Security (RLS) enforced at database level, not application level
4. **Progressive Enhancement**: Core functionality works without JavaScript, enhanced with client-side features
5. **Hybrid Data Model**: Normalized relational tables + JSONB flexibility
6. **API-less Architecture**: Direct database access via Supabase client (RLS-protected), no custom REST/GraphQL layer