Frontend Architecture - High Assurance, A11y-Friendly & Scalable

This document outlines the architectural principles and implementation strategies adopted in the frontend of the Telecom Mission Control SaaS platform. The design ensures high assurance, accessibility (a11y), global scalability, and seamless experiences across online and offline modes.

# 1. Guiding Principles

• Security-first architecture with JWT-based authentication  
• Accessible (a11y) UI using semantic HTML, ARIA roles, focus management  
• Progressive Web App (PWA) for installability and offline access  
• Mobile-first responsive design (small, medium, large breakpoints)  
• Componentized and scalable layout using React + Redux  
• Lazy loading and route-based code splitting for performance  
• Internationalization-ready (i18n scaffolded)

# 2. Core Technologies

• React + Redux + Redux Thunk  
• React Router for route management  
• Axios for HTTP client with JWT interceptors  
• CSS Modules with theming via `:root` variables  
• React Helmet for metadata  
• Service Worker for offline caching  
• Web App Manifest for installability  
• Lighthouse-tested for a11y & performance

# 3. JWT Authentication & Authorization

• Login triggers API call to `/api/auth/login`  
• On success, accessToken and refreshToken are stored securely (e.g., memory/localStorage)  
• Axios interceptors automatically inject Bearer token into Authorization headers  
• Refresh tokens are used to re-issue access tokens before expiry  
• Role-based UI rendering (e.g., sidebar menu filtering)  
• Route protection via `ProtectedRoute` component

# 4. Accessibility (a11y) Features

• Semantic HTML tags (nav, header, main, section)  
• ARIA roles (e.g., `role=alert`, `aria-live`, `aria-label`)  
• Focus ring and keyboard navigation  
• Color contrast meets WCAG 2.1 AA  
• Button & link targets >= 44px height  
• Screen-reader friendly error messaging

# 5. Responsive + Offline-first Strategy

• Layout uses Flexbox + Grid with responsive breakpoints  
• Drawer-style nav for mobile  
• PWA manifest enables add-to-home-screen on iOS/Android  
• Service Worker caches shell assets and fallback HTML  
• Offline page fallback shown when disconnected  
• Minimal runtime blocking – graceful degradation if no network

# 6. Project Structure

src/  
 ├── components/ # Reusable UI components  
 ├── screens/ # Dashboard, Policies, Towers, Devices  
 ├── actions/ # Redux actions  
 ├── reducers/ # Redux reducers  
 ├── store/ # Root Redux store config  
 ├── App.js # App shell & router  
 ├── Layout.js # Nav + Outlet wrapper  
 ├── index.js # Entry with BrowserRouter + Provider  
 └── serviceWorker.js # Offline support

# 7. Scalability Considerations

• Lazy-load screens using React.lazy  
• Debounced search + server pagination planned for large lists  
• Component-level memoization  
• Logical separation of screens and reusable UI blocks  
• UI state synced with Redux only where necessary  
• Optimized asset bundling and tree-shaking