

# SFCC Headless PWA Kit

Technical Documentation - NULogic Headless Pro Accelerator

---

## TABLE OF CONTENTS

1. Project Architecture
2. Directory Structure
3. Configuration
4. Template Extensibility
5. Key Components
6. SCAPI Integration
7. Authentication (SLAS)
8. Analytics Layer
9. Migration Tools
10. Testing Strategy
11. Deployment & CI/CD

## 1. PROJECT ARCHITECTURE

The NULogic Headless Pro accelerator extends Salesforce's PWA Kit v5 using the template extensibility pattern. Rather than forking the base framework, all customizations live in the `/overrides` directory, allowing seamless upgrades when Salesforce releases new PWA Kit versions.

### Architecture Layers

Layer	Technology	Responsibility
Client Shell	React 18, Chakra UI 2.10	PWA shell, hydration, code splitting
SSR Server	Express.js, Node.js 20	Server-side rendering, SLAS private client
State Management	React Context + Hooks	Cart, auth, analytics, UI state
Commerce Layer	Commerce SDK (isomorphic)	SCAPI calls, data normalization
Hosting	Managed Runtime (MRT)	Lambda deployment, CDN, auto-scaling
Monitoring	GA4 + GTM + DW Analytics	Unified event tracking, error reporting

## 2. DIRECTORY STRUCTURE

The project follows PWA Kit's conventional structure with NULogic extensions:

```

nu-headless-sfcc/
  overrides/app/
    components/           # 50+ UI components (header, footer, cart, search...)
    _app-config/         # ChakraProvider + CommerceApiProvider + ChatWidget
    analytics/           # GA4+GTM unified tracking layer (389 lines)
    header/              # Enterprise mega menu (1,060 lines)
    footer/              # Multi-column newsletter footer (527 lines)
  pages/                # Page-level route components
    home/                # Homepage with hero, categories, products
    product-list/        # Full PLP with faceted filtering (980 lines)
    product-detail/      # PDP with variants, reviews, recommendations
    account/             # Dashboard with sub-routes (477 lines)
  hooks/                # Custom React hooks (useCart, useWishlist, useAnalytics)
  utils/                # Helper functions (API helpers, formatters)
  constants.js          # Feature flags, cache config, sort options
  routes.jsx            # Code-split route definitions
  ssr.js                # Express SSR with Helmet security
  config/               # SFCC site configuration
  tools/sfra-audit/     # Migration assessment CLI (3,000+ lines)
  docs/                 # Migration playbook & mapping guides
  tests/                # Jest unit + Playwright e2e

```

### 3. CONFIGURATION

Core configuration lives in `config/default.js`:

Setting	Value	Purpose
clientId	aaaaaa...	SLAS public client ID for OAuth PKCE
organizationId	f_ecom_...	SFCC org identifier
shortCode	8o7m175y	SCAPI short code for API routing
siteId	RefArch	Commerce site identifier
locale	en-US	Default locale
currency	USD	Default currency
proxyConfigs	[api, ocapi, einstein]	Backend proxy configurations

### 4. TEMPLATE EXTENSIBILITY

PWA Kit v5 introduced template extensibility, allowing overrides of any file in `@salesforce/retail-react-app` without forking. Files placed in `overrides/` shadow the base package files at the same relative path.

This means `overrides/app/pages/home/index.jsx` completely replaces the default homepage while inheriting all other base files unchanged. During upgrades, only files with NULogic customizations need review.

### 5. KEY COMPONENTS

Component	Path	Lines	Features
Header	<code>components/header/</code>	1,060	Mega nav, mobile drawer, search, promo banner
Footer	<code>components/footer/</code>	527	Multi-column, newsletter signup, social links
Analytics	<code>components/analytics/</code>	389	GA4+GTM+DW unified event layer
Mini Cart	<code>components/mini-cart/</code>	397	Slide-out drawer with quantity controls
Refinements	<code>components/refinements/</code>	483	Faceted filtering with URL sync
SEO	<code>components/seo/</code>	172	JSON-LD structured data, OG tags
Error Boundary	<code>components/error-boundary/</code>	201	Error catching, dataLayer push
App Config	<code>components/_app-config/</code>	137	Providers, theme, chat widget
Home Page	<code>pages/home/</code>	404	Hero, categories, product grid
PLP	<code>pages/product-list/</code>	980	Full listing with sort, filter, pagination
Account	<code>pages/account/</code>	477	Dashboard, orders, addresses, profile

## 6. SCAPI INTEGRATION

All commerce data flows through Salesforce's Shopper Commerce API (SCAPI). The accelerator uses the Commerce SDK for JavaScript (isomorphic) which works both in SSR and client contexts. Key endpoints: Product Search, Product Detail, Basket, Order, Customer, Promotions, Categories.

## 7. AUTHENTICATION (SLAS)

Shopper Login and Activity Sequence (SLAS) provides OAuth 2.0 PKCE authentication. The SSR server (`ssr.js`) initializes the SLAS private client for secure token exchange. Client-side auth uses the public client with PKCE challenge/verifier pairs stored in secure cookies.

## 8. ANALYTICS LAYER

The unified analytics component (`components/analytics/index.jsx`, 389 lines) provides a single integration point for GA4, Google Tag Manager, and Salesforce DW Analytics. The `useAnalytics` hook exposes methods for tracking page views, product impressions, add-to-cart events, checkout steps, and purchase completions across all three platforms simultaneously.

## 9. MIGRATION TOOLS

### SFRA Audit CLI

Located in `tools/sfra-audit/` (3,072 lines across 3 files), the migration assessment toolkit provides automated analysis of existing SFRA storefronts:

Tool	File	Lines	Capability
Main Scanner	<code>index.js</code>	1,192	CLI entry, report generation, scoring
Web Scanner	<code>web-scanner.js</code>	909	Live storefront crawl, DOM analysis
API Discovery	<code>api-discovery.js</code>	971	SCAPI/OCAPI endpoint mapping

## 10. TESTING STRATEGY

The accelerator ships with a comprehensive testing framework: Jest for unit tests (extending `@salesforce/pwa-kit-dev` config), Playwright for end-to-end browser tests, and Lighthouse CI integration for performance regression detection. Test coverage targets: 80% for components, 90% for utilities, 100% for checkout flow.

## 11. DEPLOYMENT & CI/CD

Primary deployment targets Salesforce Managed Runtime via GitHub Actions. Alternative deployment to Google Cloud Run is supported via `cloudbuild.yaml` and `Dockerfile`. The CI pipeline includes lint, type-check, unit tests, build, and deploy stages with quality gates at each step.

---

**For full repository access and implementation support, contact NULogic.**

[sales@nulongic.io](mailto:sales@nulongic.io) | +1 865 408 7639 | [nulongic.io](http://nulongic.io)