

ArchySocial: Product Overview and Functional Architecture

Company: Architech d.o.o.

Generated: 5/26/2026, 2:50:46 PM

Purpose: External Architecture and Product Documentation

ArchySocial Product Overview and Functional Architecture

Purpose

This document describes what the ArchySocial product does, how it is built, and how it functions across user interfaces, services, data flows, and external integrations.

Audience

This document is intended for external partners, solution architects, and technical stakeholders evaluating platform capabilities and deployment fit.

Product Summary

ArchySocial is an AI-powered social media operations platform for planning, generating, scheduling, publishing, and monitoring social content at scale.

Primary capabilities:

- Guided campaign creation (brief, tone, visuals, generation, scheduling)
- AI-generated captions, hashtags, images, and videos
- Multi-network publishing workflows
- Social account connection and token lifecycle handling

- Media library management
- Billing and credit consumption flows
- Platform administration workflows (analytics, notifications, error reporting, email workflows)

Primary User Groups

- Marketing and social teams (campaign creation and scheduling)
- Content creators (generation and edits)
- Platform administrators (operations, analytics, incident triage, notifications)
- Finance and operations users (billing and subscription management)

Functional Architecture

Experience Layer (User Interfaces)

Main web interfaces:

- Campaign wizard and campaign detail pages
- Dashboard and analytics pages
- Calendar scheduling view
- Media library
- Account connection pages
- Billing and pricing pages
- Admin console pages

Interface characteristics:

- SSR and client-side interactivity for low-latency user experience
- Role-protected admin operations
- Real-time and near real-time status updates for asynchronous generation and publishing

Application Layer

Core runtime behaviors:

- Route handlers and server functions validate inputs, authorize users, and execute business logic
- AI orchestration services invoke model endpoints for content generation
- Publishing orchestrators process queued posts and execute network-specific API operations
- Admin modules provide operational controls and diagnostics

Data Layer

Core stores and records:

- Campaign, post scheduling, and post status records
- Social connection and token metadata
- Media asset metadata and storage references
- Billing/subscription and credits ledgers
- Error reports, notifications, and admin audit-related records

How the Product Functions (End-to-End)

Campaign-to-Publish Flow

1. User creates campaign brief and brand context.
2. AI services generate text and media variants.
3. User reviews and schedules posts.
4. Scheduler identifies due posts and invokes publish operations.
5. Post statuses are updated to published, failed, or skipped.
6. Analytics and reporting surfaces reflect execution outcomes.

Admin Operations Flow

1. Admin signs in and enters operations console.
2. Admin monitors errors, analytics, and workflow executions.
3. Admin can configure and run email workflows, including media-enhanced content templates.
4. Audit and telemetry events support traceability and issue resolution.

External Dependencies and Integrations

- Azure OpenAI for text/image/video generation
- Supabase for database, auth, storage, and edge capabilities
- Social network APIs for publishing and account integrations
- Billing provider APIs for subscriptions and credit operations
- DNS and HTTPS endpoints for production and preview access patterns

Security and Trust Boundaries

- The platform enforces authenticated access for all protected operations.
- Administrative operations are restricted by role-aware access controls.
- Operational secrets and credentials are managed through controlled deployment environments.
- Customer data access is governed by tenant-aware authorization and policy controls.

Other Cloud and On-Premises Considerations

Current architecture is cloud-native and internet-facing. On-premises systems are not currently required for core operation.

Possible enterprise extension patterns:

- CRM/ERP data ingestion through secure APIs
- Identity federation to customer enterprise identity providers
- Data export and BI integration pipelines

Non-Functional Characteristics

- Reusable IP implemented as repeatable application code and deployment workflows
- Automated CI/CD with preview and production promotion paths
- Observable runtime with logs, workspace-level telemetry, and alerting resources
- Multi-environment support for development, preview, and production

© 2026 Architech d.o.o. All rights reserved.

Official External Documentation for Partners