

Contents

Admin Dashboard Reference	1
Dashboard Architecture	1
Technology Stack	1
Page Structure	1
Complete Page Inventory (43 Pages)	2
Core Administration	2
AI & Models	2
Orchestration	2
Think Tank	3
Billing & Cost	3
Analytics & Monitoring	3
AGI & Learning	3
Collaboration & Features	3
Key Pages Detail	4
Overview Dashboard (/)	4
Models Page (/models)	4
Orchestration Patterns Page (/orchestration-patterns)	4
Visual Workflow Editor (/orchestration-patterns/editor)	5
Billing Page (/billing)	6
Analytics Page (/analytics)	6
Security Page (/security)	6
Think Tank Page (/thinktank)	7
Component Library	7
Shared Components	7
API Integration	8
API Client (lib/api.ts)	8
Authentication (lib/auth.ts)	8

Admin Dashboard Reference

Dashboard Architecture

Technology Stack

- **Framework:** Next.js 14 (App Router)
- **Language:** TypeScript
- **Styling:** Tailwind CSS
- **Components:** shadcn/ui
- **Icons:** Lucide React
- **State:** React Query, Zustand
- **Forms:** React Hook Form, Zod

Page Structure

```
apps/admin-dashboard/
  app/
    layout.tsx          # Root layout
```

```

page.tsx          # Landing/login
(dashboard)/      # Dashboard routes (43 pages)
    layout.tsx    # Dashboard layout with sidebar
    page.tsx      # Overview dashboard
    [module]/page.tsx # Individual modules
components/
    ui/           # shadcn/ui components
    workflow-editor/ # Visual workflow editor
    shared/        # Shared components
lib/
    api.ts        # API client
    auth.ts        # Auth utilities
    utils.ts       # Helper functions

```

Complete Page Inventory (43 Pages)

Core Administration

Page	Route	Purpose
Overview	/	System health, key metrics, quick actions
Administrators	/administrators	Manage admin users, roles, permissions
Audit Logs	/audit-logs	View all system audit events
AWS Logs	/aws-logs	CloudWatch log viewer
Security	/security	Security settings, WAF, compliance
Settings	/settings	Platform configuration
System Config	/system-config	Advanced system settings

AI & Models

Page	Route	Purpose
Models	/models	AI model configuration
Model Metadata	/model-metadata	Model capabilities & pricing
User Models	/user-models	Per-tenant model access
Providers	/providers	AI provider management

Orchestration

Page	Route	Purpose
Orchestration	/orchestration	Workflow management
Orchestration Patterns	/orchestration-patterns	49 patterns library
Orchestration Editor	/orchestration-patterns/editor	Visual workflow editor

Think Tank

Page	Route	Purpose
Think Tank	/thinktank	Session management
Cognition	/cognition	Cognitive settings
Cognitive Brain	/cognitive-brain	Brain configuration
Consciousness	/consciousness	Consciousness monitoring
Metacognition	/metacognition	Self-reflection settings
Planning	/planning	Goal planning
World Model	/world-model	World model state

Billing & Cost

Page	Route	Purpose
Billing	/billing	Revenue, invoices, subscriptions
Cost	/cost	Cost analytics, budgets

Analytics & Monitoring

Page	Route	Purpose
Analytics	/analytics	Usage analytics
Reports	/reports	Generated reports
Health	/health	System health dashboard
Deployments	/deployments	Deployment history

AGI & Learning

Page	Route	Purpose
Agents	/agents	Autonomous agents
Learning	/learning	ML training data
ML Training	/ml-training	Model training jobs
Self-Improvement	/self-improvement	Self-improvement logs
Moral Compass	/moral-compass	Ethical guidelines
Feedback	/feedback	User feedback

Collaboration & Features

Page	Route	Purpose
Time Machine	/time-machine	Historical state access
Storage	/storage	File storage management
Notifications	/notifications	Notification settings
Localization	/localization	i18n management

Page	Route	Purpose
Configuration	/configuration	Dynamic configuration
Compliance	/compliance	Compliance dashboard
Geographic	/geographic	Geographic settings
Multi-Region	/multi-region	Multi-region config
Experiments	/experiments	A/B testing
Migrations	/migrations	Database migrations
Services	/services	Service status
Request Handler	/request-handler	Request routing

Key Pages Detail

Overview Dashboard (/)

Metrics Displayed: - Active tenants (24h) - Total API requests (24h) - Total tokens processed - Revenue (MTD) - Error rate - Average latency

Quick Actions: - View recent errors - Check provider health - Review pending approvals - Generate report

Charts: - Requests over time (7d) - Token usage by model - Revenue trend - Error rate trend

Models Page (/models)

Features: - List all 106+ models - Filter by provider, capability - Enable/disable models - Set model pricing overrides - Configure fallback chains - View usage statistics

Model Card Display:

anthropic/clause-3-5-sonnet

Provider: Bedrock (primary), LiteLLM (fb)

Capabilities: reasoning, coding, vision

Context: 200K tokens

Pricing: \$3.00/\$15.00 per 1M tokens

Status: Enabled

Usage (24h): 1.2M tokens

[Configure] [Disable] [View Stats]

Orchestration Patterns Page (/orchestration-patterns)

Features: - Browse 49 patterns by category - Search patterns - View pattern details - Edit pattern workflows - Create custom patterns - View execution statistics

Pattern Categories Tabs: - Consensus & Aggregation (7) - Debate & Deliberation (7) - Critique & Refinement (7) - Verification & Validation (7) - Decomposition (7) - Specialized Reasoning (7) - Multi-Model Routing (4) - Ensemble Methods (3)

Pattern Detail View:

AI Debate

[Edit] [Test]

Category: Debate & Deliberation

Quality Improvement: +25-40%

Typical Latency: High (10-30s)

Min Models: 3

Description:

Two AI models debate opposing positions while a third model judges the arguments and synthesizes a final answer.

Best For:

- Controversial topics
- Complex decisions
- Exploring multiple perspectives

Workflow Steps:

1. Generate Pro Argument (Claude)
2. Generate Con Argument (GPT-4o)
3. Judge Arguments (Claude - thinking mode)
4. Synthesize Final Answer

Executions (30d): 1,247 | Avg Quality: 0.89

Visual Workflow Editor (/orchestration-patterns/editor)

Features: - Drag-and-drop workflow design - 16 method palette - Node connection editing - Step configuration (4 tabs) - Zoom, pan, fit controls - Test execution - Save/load workflows

Method Palette (16 Methods): | Method | Category | Description | | — | — | — |
| Generate | Core | Generate text response | | Analyze | Core | Analyze input | | Transform | Core
| Transform data | | Validate | Core | Validate output | | Critique | Refinement | Critique response
| | Refine | Refinement | Improve response | | Decompose | Decomposition | Break into parts |
| Synthesize | Aggregation | Combine results | | Judge | Evaluation | Evaluate quality | | Vote |
Consensus | Majority voting | | Debate_Pro | Debate | Pro argument | | Debate_Con | Debate | Con
argument | | Verify | Verification | Fact-check | | Search | External | Web search | | Execute_Code
| External | Run code | | Custom | Custom | Custom logic |

Step Configuration Tabs: 1. **General** - Name, order, model, output variable 2. **Parameters** - Method-specific parameters 3. **Advanced** - Conditions, iterations, dependencies 4. **Parallel** -

Parallel execution settings, AGI selection

Billing Page (/billing)

Sections:

Revenue Overview: - Monthly Recurring Revenue (MRR) - Annual Recurring Revenue (ARR)
- Revenue growth % - Churn rate

Subscription Management: - Active subscriptions by tier - Upcoming renewals - Cancelled subscriptions - Trial conversions

Credit Management: - Total credits sold - Credits consumed - Credit purchase history - Low balance alerts

Invoice Management: - Generate invoices - View invoice history - Export to CSV - Send invoice reminders

Analytics Page (/analytics)

Dashboard Sections:

Usage Analytics: - Requests by model - Tokens by tenant - Peak usage times - Geographic distribution

Performance Analytics: - Latency percentiles (p50, p95, p99) - Error rates by endpoint - Provider availability - Cache hit rates

Business Analytics: - Cost per request - Revenue per tenant - Feature adoption - User engagement

Custom Reports: - Date range selection - Dimension grouping - Metric selection - Export options (CSV, PDF, JSON)

Security Page (/security)

Sections:

Authentication: - Cognito configuration - MFA enforcement - Session settings - Password policies

API Security: - Rate limiting rules - IP allowlists - API key management - Request validation

Compliance: - SOC2 status - HIPAA mode toggle - Data retention settings - Audit log retention

WAF Configuration: - Rule management - Blocked requests - Rate limit thresholds - Custom rules

Think Tank Page (/thinktank)

Features: - View all sessions across tenants - Filter by domain, status, confidence - Session detail view - Step-by-step reasoning display - Cost and token tracking

Session List View:

Session ID	Tenant	Domain	Steps	Conf	Cost
abc123...	Acme	Engineering	6	0.92	\$0.45
def456...	Beta	Research	8	0.87	\$0.72
ghi789...	Acme	Legal	4	0.95	\$0.28

Session Detail View:

Problem: "Design a microservices architecture for 10M daily users"

Domain: Engineering | Complexity: High | Status: Completed

Step 1: Decompose [Claude 3.5] (conf: 0.94)
Identified 5 sub-problems
Duration: 2.3s | Tokens: 1,247

Step 2: Requirements Analysis [Claude + GPT-4o parallel] (conf: 0.91)
Synthesized from 2 models
Duration: 4.1s | Tokens: 3,892

Step 3-5: [...]

Step 6: Synthesize Final Solution [Claude 3.5 thinking] (conf: 0.89)
Generated comprehensive solution
Duration: 5.7s | Tokens: 2,156

Total: 6 steps | 18.2s | 12,453 tokens | \$0.45
Final Confidence: 0.89

Component Library

Shared Components

DataTable: - Sortable columns - Pagination - Row selection - Export functionality - Column visibility toggle

StatusBadge: - Status indicator with color - Configurable variants - Icon support

MetricCard: - Large number display - Trend indicator - Comparison to previous period

Chart Components: - LineChart (time series) - BarChart (comparisons) - PieChart (distributions) - AreaChart (cumulative)

Form Components: - Input with validation - Select with search - DateRangePicker - JSONEditor
- CodeEditor

API Integration

API Client (lib/api.ts)

```
import { QueryClient } from '@tanstack/react-query';

const API_BASE = process.env.NEXT_PUBLIC_API_URL;

export const apiClient = {
    // GET request
    async get<T>(path: string): Promise<T> {
        const response = await fetch(`[${API_BASE}][${path}]`, {
            headers: await getAuthHeaders(),
        });
        if (!response.ok) throw new APIError(response);
        return response.json();
    },

    // POST request
    async post<T>(path: string, data: unknown): Promise<T> {
        const response = await fetch(`[${API_BASE}][${path}]`, {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                ...await getAuthHeaders(),
            },
            body: JSON.stringify(data),
        });
        if (!response.ok) throw new APIError(response);
        return response.json();
    },
}

// React Query hooks
useModels: () => useQuery(['models'], () => apiClient.get('/admin/models')),
useTenants: () => useQuery(['tenants'], () => apiClient.get('/admin/tenants')),
useAnalytics: (range: string) =>
    useQuery(['analytics', range], () => apiClient.get('/admin/analytics?range=${range}')),
};
```

Authentication (lib/auth.ts)

```
import { Amplify, Auth } from 'aws-amplify';

export async function getAuthHeaders(): Promise<Headers> {
```

```
const session = await Auth.currentSession();
return {
  'Authorization': `Bearer ${session.getIdToken().getJwtToken()}`,
};
}

export function useAuth() {
  const [user, setUser] = useState<CognitoUser | null>(null);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    Auth.currentAuthenticatedUser()
      .then(setUser)
      .catch(() => setUser(null))
      .finally(() => setLoading(false));
  }, []);

  return { user, loading, signIn, signOut };
}
```