OneMindAI: Hardcoding Architecture Analysis

Document Version: 1.0
Created: December 13, 2025
Purpose: Define where every hardcoded value should live and why

---

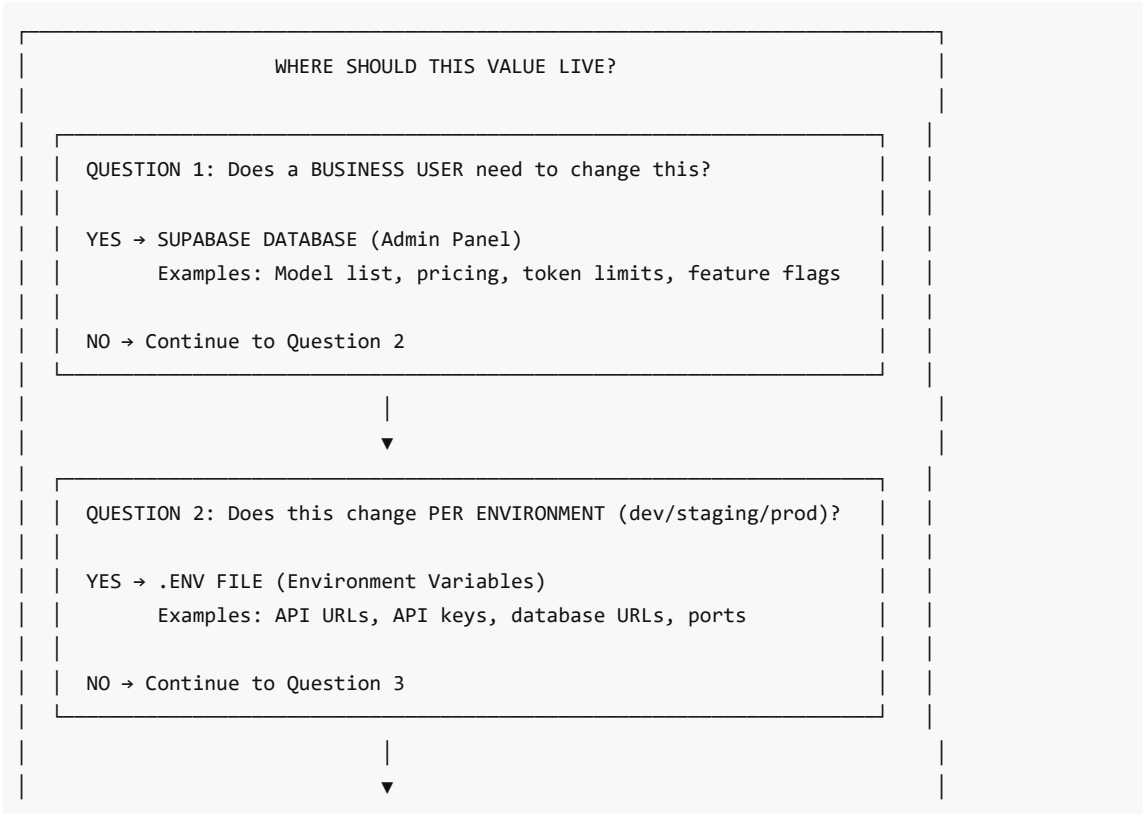
📋 TABLE OF CONTENTS

🎯 EXECUTIVE SUMMARY {#executive-summary}

The Core Question You Asked

> *"Where should hardcoded values live - Frontend code, Supabase database, or Config file?"*

The Answer (Decision Framework)



```
│  ┌──────────────────────────────────────────────────────────────────┐   │
│  │  QUESTION 3: Is this a TECHNICAL CONSTANT that rarely changes?   │   │
│  │                                                                  │   │
│  │  YES → FRONTEND CONFIG FILE (src/config/constants.ts)            │   │
│  │        Examples: Retry counts, debounce delays, UI breakpoints   │   │
│  │                                                                  │   │
│  │  NO → Can stay INLINE IN CODE                                    │   │
│  │        Examples: CSS values, component structure, validation regex  │   │
│  └──────────────────────────────────────────────────────────────────┘   │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

📊 CURRENT STATE ANALYSIS (From Your Spreadsheet) {#current-state}

Spreadsheet Data Extracted

Based on your spreadsheet image, here are the **7 categories** of hardcoded values identified:

CATEGORY 1: Token Limits (CRITICAL)

Value	Current Location	Current Value	Your Recommendation	My Analysis
`contextLimit`	OneMindAI.tsx:175-188	128K, 200K, 1M, 64K, 32K	Move to config file/API	→ **SUPABASE** (per-model, admin changes)
`MODEL_TOKEN_LIMITS`	OneMindAI.tsx:192-246	Various per model	Move to config	→ **SUPABASE** (per-model, admin changes)
`DEFAULT_TOKEN_LIMIT`	OneMindAI.tsx:240	8192	Fallback, make configurable	→ **SUPABASE** (system_config table)
`PROVIDER_MAX_OUTPUT`	OneMindAI.tsx:381-395	4096-8192 per provider	Remove, use backend caps	→ **SUPABASE** (per-provider setting)
Backend caps	ai-proxy.cjs:401-407	0.9 (90% of context)	Keep as safe fallback	→ **KEEP IN BACKEND** (safety)
GPT-5 max	ai-proxy.cjs:793	128000	Let provider handle	→ **SUPABASE** (when GPT-5 launches)
GPT-4 output	ai-proxy.cjs:794	16384	Let provider handle	→ **SUPABASE**
Claude output	ai-proxy.cjs:981	8192	Let provider handle	→ **SUPABASE**
Gemini output	ai-proxy.cjs:988	8192	Let provider handle	→ **SUPABASE**

Mistral output	ai-proxy.cjs:1116	32768	Let provider handle	→ **SUPABASE**
Perplexity output	ai-proxy.cjs:1195	4096	Let provider handle	→ **SUPABASE**
DeepSeek output	ai-proxy.cjs:1271	8192	Let provider handle	→ **SUPABASE**
Falcon tokens	OneMindAI.tsx:2798	8000	Match backend limits	→ **SUPABASE**

CATEGORY 2: Pricing (Business Data)

Value	Current Location	Current Value	Your Recommendation	My Analysis
`BASE_PRICING`	OneMindAI.tsx:252-315	Full pricing dictionary	Move to database/config	→ **SUPABASE** (business changes pricing)
`expectedOutputTokens`	OneMindAI.tsx:332	number = 1000	Reasonable default	→ **SUPABASE** (system_config)
Divisor for pricing	OneMindAI.tsx:347	1,000,000	Industry standard	→ **KEEP IN CODE** (never changes)
Convert per-token cost	OneMindAI.tsx:3001	1,000,000	Industry standard	→ **KEEP IN CODE** (never changes)
Cents to dollars	OneMindAI.tsx:3642	100	Standard conversion	→ **KEEP IN CODE** (never changes)

CATEGORY 3: Prompt Limits (UX Guardrails)

Value	Current Location	Current Value	Your Recommendation	My Analysis
`PROMPT_SOFT_LIMIT`	OneMindAI.tsx:430	5000	Make admin configurable	→ **SUPABASE** (UX tuning)
`PROMPT_HARD_LIMIT`	OneMindAI.tsx:431	10000	Block threshold	→ **SUPABASE** (UX tuning)
`PROMPT_CHUNK_SIZE`	OneMindAI.tsx:432	4000	For chunking long prompts	→ **SUPABASE** (could be reasonable default)
`MAX_PROMPT_LENGTH`	OneMindAI.tsx:~1459	7000	Truncate before API call	→ **SUPABASE** (per-provider could differ)

CATEGORY 4: Token Estimation Multipliers

Value	Current Location	Current Value	Your Recommendation	My Analysis
tiktoken chars	OneMindAI.tsx:322	0.75	Keep, industry approximation	→ **CONFIG FILE** (technical constant)

tiktoken chars	OneMindAI.tsx:322	0.002	Fine-tuning	→ **CONFIG FILE**
sentencepiece	OneMindAI.tsx:323	0.95	Keep	→ **CONFIG FILE**
sentencepiece	OneMindAI.tsx:323	0.003	Fine-tuning	→ **CONFIG FILE**
bytebpe fallback	OneMindAI.tsx:324	0.6	Keep	→ **CONFIG FILE**
bytebpe	OneMindAI.tsx:324	0.004	Fine-tuning	→ **CONFIG FILE**

CATEGORY 5: Time/Label Thresholds (UI Display)

Value	Current Location	Current Value	Your Recommendation	My Analysis
Base time offset	OneMindAI.tsx:367	2 (base seconds)	Keep - UX	→ **CONFIG FILE** (UI tuning)
Seconds threshold	OneMindAI.tsx:368	20	"a few seconds" label	→ **CONFIG FILE**
Switch to minutes	OneMindAI.tsx:369	90	Threshold	→ **CONFIG FILE**
Output label thresholds	OneMindAI.tsx:3243	10, 2000, 4000	UI display	→ **CONFIG FILE**
`UPDATE_INTERVAL`	OneMindAI.tsx:3243	15 (ms)	60fps refresh rate	→ **CONFIG FILE**
`STREAM_TIMEOUT`	OneMindAI.tsx:3244	30000 (ms)	30 seconds without chunks	→ **SUPABASE** (could need tuning)

CATEGORY 6: Engine Metadata (Display Text)

Value	Current Location	Current Value	Your Recommendation	My Analysis
`engineInfoText`	OneMindAI.tsx:492-600+	Taglines, descriptions, badges	Move to database for admin editing	→ **SUPABASE** (marketing text)
`seededEngines`	OneMindAI.tsx:174-188	Names, versions, default versions	Move to config/database	→ **SUPABASE** (add new engines via admin)
Default selected engines	OneMindAI.tsx:871	openai, deepseek, mistral true	Make admin-configurable	→ **SUPABASE**
`modelInfoDictionary`	OneMindAI.tsx:583-646	Model descriptions for tooltips	Move to database	→ **SUPABASE** (stale info problem)

CATEGORY 7: API/Streaming Configuration

Value	Current Location	Current Value	Your Recommendation	My Analysis
Supported providers	OneMindAI.tsx:632	['openai', 'deepseek', 'mistral']	Move to config	→ **SUPABASE** (enable/disable providers)
Proxy URL fallback	OneMindAI.tsx:~1545	'http://localhost:3002'	Always use env vars	→ **.ENV FILE** (environment-specific)
Default temperature	OneMindAI.tsx:~1740	0.7	Make configurable	→ **SUPABASE** (per-model default)

🧬 THE THREE STORAGE LOCATIONS {#three-locations}

Location 1: SUPABASE DATABASE (Admin Panel Controlled)

Why Supabase?

```
┌─────────────────────────────────────────────────────────────────────┐
│                    WHY SUPABASE FOR BUSINESS VALUES?                │
│                                                                     │
│  ✓ Business users can change without developer                      │
│  ✓ No code deployment needed                                        │
│  ✓ Changes take effect immediately (real-time)                      │
│  ✓ Audit trail (who changed what, when)                             │
│  ✓ Rollback possible via database backups                           │
│  ✓ AI cannot accidentally modify (not in code files)                │
│  ✓ Different values per environment possible                        │
│  ✓ Row Level Security for access control                            │
│                                                                     │
│  WHAT GOES HERE:                                                    │
│  • AI model list (names, versions, capabilities)                    │
│  • Pricing per model (input/output costs)                           │
│  • Token limits per model                                           │
│  • Feature flags (enable/disable providers)                         │
│  • User-facing text (descriptions, badges)                          │
│  • Business rules (markup %, signup bonus)                          │
│  • UX thresholds (prompt limits, timeouts)                          │
└─────────────────────────────────────────────────────────────────────┘
```

Supabase Tables Needed

```
-- Table 1: AI Models (replaces seededEngines + MODEL_TOKEN_LIMITS)
CREATE TABLE ai_models (
```

```
  id TEXT PRIMARY KEY,                    -- 'gpt-4o', 'claude-3.5-sonnet'
  provider TEXT NOT NULL,                 -- 'openai', 'anthropic'
  display_name TEXT NOT NULL,             -- 'GPT-4o'
  api_model_id TEXT NOT NULL,             -- Actual API identifier

  -- Token Configuration
  context_limit INTEGER DEFAULT 128000,
  max_output_tokens INTEGER DEFAULT 4096,
  default_temperature DECIMAL(3,2) DEFAULT 0.7,

  -- Pricing (USD per 1M tokens)
  input_price DECIMAL(10,4),
  output_price DECIMAL(10,4),

  -- Metadata
  description TEXT,
  tagline TEXT,
  badge TEXT,                             -- 'FAST', 'CHEAP', 'BEST'
  tokenizer TEXT DEFAULT 'tiktoken',      -- 'tiktoken', 'sentencepiece', 'bytebpe'
  is_vision_capable BOOLEAN DEFAULT false,

  -- Control
  is_active BOOLEAN DEFAULT true,
  is_default_selected BOOLEAN DEFAULT false,
  display_order INTEGER DEFAULT 0,

  created_at TIMESTAMPTZ DEFAULT NOW(),
  updated_at TIMESTAMPTZ DEFAULT NOW()
);

-- Table 2: System Configuration (replaces all other hardcoded values)
CREATE TABLE system_config (
  key TEXT PRIMARY KEY,
  value JSONB NOT NULL,
  category TEXT NOT NULL,                 -- 'limits', 'pricing', 'ux', 'api'
  description TEXT,
  is_sensitive BOOLEAN DEFAULT false,     -- Hide from non-admins
  updated_by UUID REFERENCES auth.users(id),
  updated_at TIMESTAMPTZ DEFAULT NOW()
);

-- Table 3: Provider Configuration (backend-specific settings)
CREATE TABLE provider_config (
  provider TEXT PRIMARY KEY,              -- 'openai', 'anthropic'
  is_enabled BOOLEAN DEFAULT true,
  max_output_cap INTEGER,                 -- Backend safety cap
  rate_limit_rpm INTEGER,                 -- Requests per minute
  timeout_seconds INTEGER DEFAULT 30,
  retry_count INTEGER DEFAULT 3,
  updated_at TIMESTAMPTZ DEFAULT NOW()
);
```

Example Data

```
-- AI Models
INSERT INTO ai_models (id, provider, display_name, api_model_id, context_limit,
max_output_tokens, input_price, output_price, tagline, is_default_selected) VALUES
('gpt-4o', 'openai', 'GPT-4o', 'gpt-4o', 128000, 16384, 2.50, 10.00, 'Balanced quality &
speed', true),
('gpt-4o-mini', 'openai', 'GPT-4o Mini', 'gpt-4o-mini', 128000, 16384, 0.15, 0.60, 'Fast &
economical', false),
('claude-3.5-sonnet', 'anthropic', 'Claude 3.5 Sonnet', 'claude-3-5-sonnet-20241022',
200000, 8192, 3.00, 15.00, 'Best reasoning', true),
('gemini-2.0-flash', 'gemini', 'Gemini 2.0 Flash', 'gemini-2.0-flash-exp', 1000000, 8192,
0.075, 0.30, 'Massive context', true);

-- System Config
INSERT INTO system_config (key, value, category, description) VALUES
('prompt_soft_limit', '5000', 'limits', 'Warning threshold for prompt length'),
('prompt_hard_limit', '10000', 'limits', 'Maximum prompt length'),
('prompt_chunk_size', '4000', 'limits', 'Chunk size for long prompts'),
('max_prompt_length', '7000', 'limits', 'Truncation point before API'),
('stream_timeout_ms', '30000', 'api', 'Timeout for streaming responses'),
('expected_output_tokens', '1000', 'pricing', 'Default expected output for cost
estimation'),
('signup_bonus_credits', '100', 'pricing', 'Credits given to new users'),
('markup_percentage', '30', 'pricing', 'Markup over provider costs'),
('default_temperature', '0.7', 'api', 'Default temperature for AI calls');

-- Provider Config
INSERT INTO provider_config (provider, is_enabled, max_output_cap, rate_limit_rpm,
timeout_seconds) VALUES
('openai', true, 16384, 60, 30),
('anthropic', true, 8192, 40, 30),
('gemini', true, 8192, 60, 30),
('deepseek', true, 8192, 60, 30),
('mistral', true, 32768, 60, 30),
('perplexity', true, 4096, 30, 30),
('groq', true, 8192, 30, 30),
('xai', true, 16384, 30, 30),
('kimi', true, 8192, 30, 30);
```

Location 2: .ENV FILE (Environment Variables)

Why .env?

```
┌──────────────────────────────────────────────────────────────────────────┐
│                    WHY .ENV FOR ENVIRONMENT VALUES?                      │
│                                                                          │
│  ✓ Different per environment (dev/staging/prod)                          │
│  ✓ Secrets never in code or database                                     │
```

```
│  ✓ Standard practice for 12-factor apps                                  │
│  ✓ Easy to change during deployment                                      │
│  ✓ Not version controlled (security)                                     │
│                                                                          │
│  WHAT GOES HERE:                                                         │
│  • API keys (OPENAI_API_KEY, ANTHROPIC_API_KEY, etc.)                    │
│  • Database URLs (SUPABASE_URL, DATABASE_URL)                            │
│  • Service URLs (VITE_BACKEND_URL, VITE_PROXY_URL)                       │
│  • Ports (AI_PROXY_PORT)                                                 │
│  • Environment flags (NODE_ENV, DEBUG)                                   │
│  • Third-party service keys (HUBSPOT_CLIENT_ID)                          │
└──────────────────────────────────────────────────────────────────────────┘
```

Current .env Values (Keep Here)

```
# Server Configuration
AI_PROXY_PORT=3002
NODE_ENV=development

# CORS (could move to database for admin control)
ALLOWED_ORIGINS=http://localhost:5173,http://localhost:3000

# API Keys (MUST stay in .env - NEVER in database)
OPENAI_API_KEY=sk-...
ANTHROPIC_API_KEY=sk-ant-...
GOOGLE_AI_API_KEY=AI...
MISTRAL_API_KEY=...
PERPLEXITY_API_KEY=pplx-...
DEEPSEEK_API_KEY=sk-...
GROQ_API_KEY=gsk_...
XAI_API_KEY=xai-...
KIMI_API_KEY=...

# Supabase
VITE_SUPABASE_URL=https://your-project.supabase.co
VITE_SUPABASE_ANON_KEY=eyJ...
SUPABASE_SERVICE_KEY=eyJ...  # Backend only!

# HubSpot
HUBSPOT_CLIENT_ID=...
HUBSPOT_CLIENT_SECRET=...

# Backend URL
VITE_BACKEND_URL=http://localhost:3002
VITE_PROXY_URL=http://localhost:3002
```

Location 3: CONFIG FILE (Technical Constants)

Why Config File?

```
┌──────────────────────────────────────────────────────────────────────────┐
│                    WHY CONFIG FILE FOR TECHNICAL CONSTANTS?              │
│                                                                          │
│  ✓ Rarely changes (once per quarter or less)                             │
│  ✓ Technical/developer decision, not business                            │
│  ✓ Needs code review before changing                                     │
│  ✓ Type-safe with TypeScript                                             │
│  ✓ No database round-trip needed                                         │
│  ✓ Bundled with app for performance                                      │
│                                                                          │
│  WHAT GOES HERE:                                                         │
│  • Token estimation multipliers (0.75, 0.95, 0.6)                        │
│  • UI timing constants (debounce delays, animation durations)            │
│  • Retry configuration (count, backoff multiplier)                       │
│  • Industry standards (1,000,000 for per-million pricing)                │
│  • Validation patterns (regex for email, URL)                            │
│  • UI breakpoints (mobile, tablet, desktop widths)                       │
└──────────────────────────────────────────────────────────────────────────┘
```

Proposed Config File: `src/config/constants.ts`

```typescript
/**
 * TECHNICAL CONSTANTS
 *
 * These values are technical implementation details that:
 * 1. Rarely change (less than once per quarter)
 * 2. Require developer understanding to modify
 * 3. Don't need business user access
 *
 * DO NOT put business values here - use Supabase database instead.
 */

// ============================================================================
// TOKEN ESTIMATION (Industry approximations - rarely change)
// ============================================================================

export const TOKEN_ESTIMATION = {
  tiktoken: { charsPerToken: 0.75, adjustment: 0.002 },
  sentencepiece: { charsPerToken: 0.95, adjustment: 0.003 },
  bytebpe: { charsPerToken: 0.6, adjustment: 0.004 },
} as const;

// ============================================================================
// PRICING CONSTANTS (Industry standards - never change)
// ============================================================================

export const PRICING_CONSTANTS = {
  TOKENS_PER_MILLION: 1_000_000,  // Industry standard
  CENTS_PER_DOLLAR: 100,          // Currency standard
} as const;
```

```ts
// ============================================================================
// UI TIMING (UX polish - developer decision)
// ============================================================================

export const UI_TIMING = {
  DEBOUNCE_MS: 300,               // Input debounce
  ANIMATION_DURATION_MS: 200,     // Framer motion default
  UPDATE_INTERVAL_MS: 15,         // ~60fps for streaming
  TOAST_DURATION_MS: 5000,        // Notification display time
} as const;

// ============================================================================
// TIME DISPLAY THRESHOLDS (UX labels)
// ============================================================================

export const TIME_THRESHOLDS = {
  BASE_SECONDS: 2,                // Minimum display
  FEW_SECONDS_MAX: 20,            // "a few seconds" threshold
  SWITCH_TO_MINUTES: 90,          // When to show minutes
} as const;

// ============================================================================
// RETRY CONFIGURATION (Error recovery)
// ============================================================================

export const RETRY_CONFIG = {
  MAX_RETRIES: 3,
  INITIAL_DELAY_MS: 1000,
  BACKOFF_MULTIPLIER: 2,
  MAX_DELAY_MS: 30000,
} as const;

// ============================================================================
// FILE UPLOAD LIMITS (Technical constraints)
// ============================================================================

export const FILE_LIMITS = {
  MAX_FILE_SIZE_MB: 10,
  MAX_FILES: 5,
  ALLOWED_IMAGE_TYPES: ['image/jpeg', 'image/png', 'image/gif', 'image/webp'],
  ALLOWED_DOC_TYPES: ['application/pdf', 'text/plain', 'application/json'],
} as const;

// ============================================================================
// UI BREAKPOINTS (Responsive design)
// ============================================================================

export const BREAKPOINTS = {
  MOBILE: 640,
  TABLET: 768,
  DESKTOP: 1024,
```

```
  WIDE: 1280,
} as const;
```

📦 COMPLETE HARDCODING INVENTORY & RECOMMENDATIONS {#inventory}

Master Table: Every Hardcoded Value

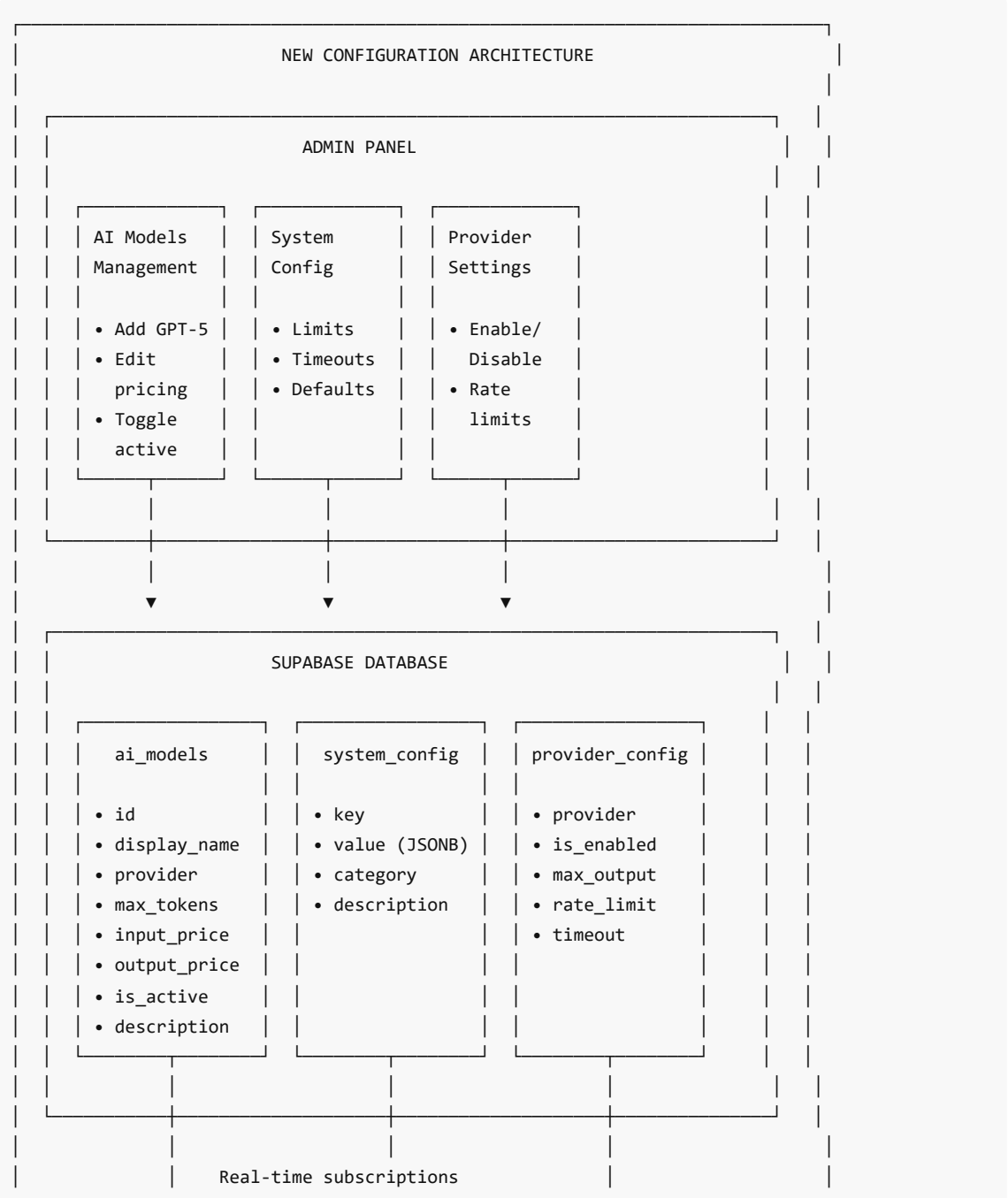
#	Value Name	Current Location	Current Value	**NEW LOCATIC**
TOKEN LIMITS				
1	contextLimit (per model)	constants.ts:39-179	128K-1M	**SUPABASE: ai_models**
2	MODEL_TOKEN_LIMITS	OneMindAI.tsx:192-246	Various	**SUPABASE: ai_models**
3	DEFAULT_TOKEN_LIMIT	OneMindAI.tsx:240	8192	**SUPABASE: system_con**
4	PROVIDER_MAX_OUTPUT	OneMindAI.tsx:381-395	4096-8192	**SUPABASE: provider_cc**
5	Backend output caps	ai-proxy.cjs:793-1271	Various	**SUPABASE: provider_cc**
6	Context reserve (90%)	ai-proxy.cjs:407	0.9	**CONFIG FIL**
PRICING				
7	BASE_PRICING	constants.ts:186-251	Full dict	**SUPABASE: ai_models**
8	expectedOutputTokens	OneMindAI.tsx:332	1000	**SUPABASE: system_con**
9	Per-million divisor	OneMindAI.tsx:347	1,000,000	**CONFIG FIL**
10	Cents to dollars	OneMindAI.tsx:3642	100	**CONFIG FIL**
PROMPT LIMITS				
11	PROMPT_SOFT_LIMIT	OneMindAI.tsx:430	5000	**SUPABASE: system_con**
12	PROMPT_HARD_LIMIT	OneMindAI.tsx:431	10000	**SUPABASE: system_con**

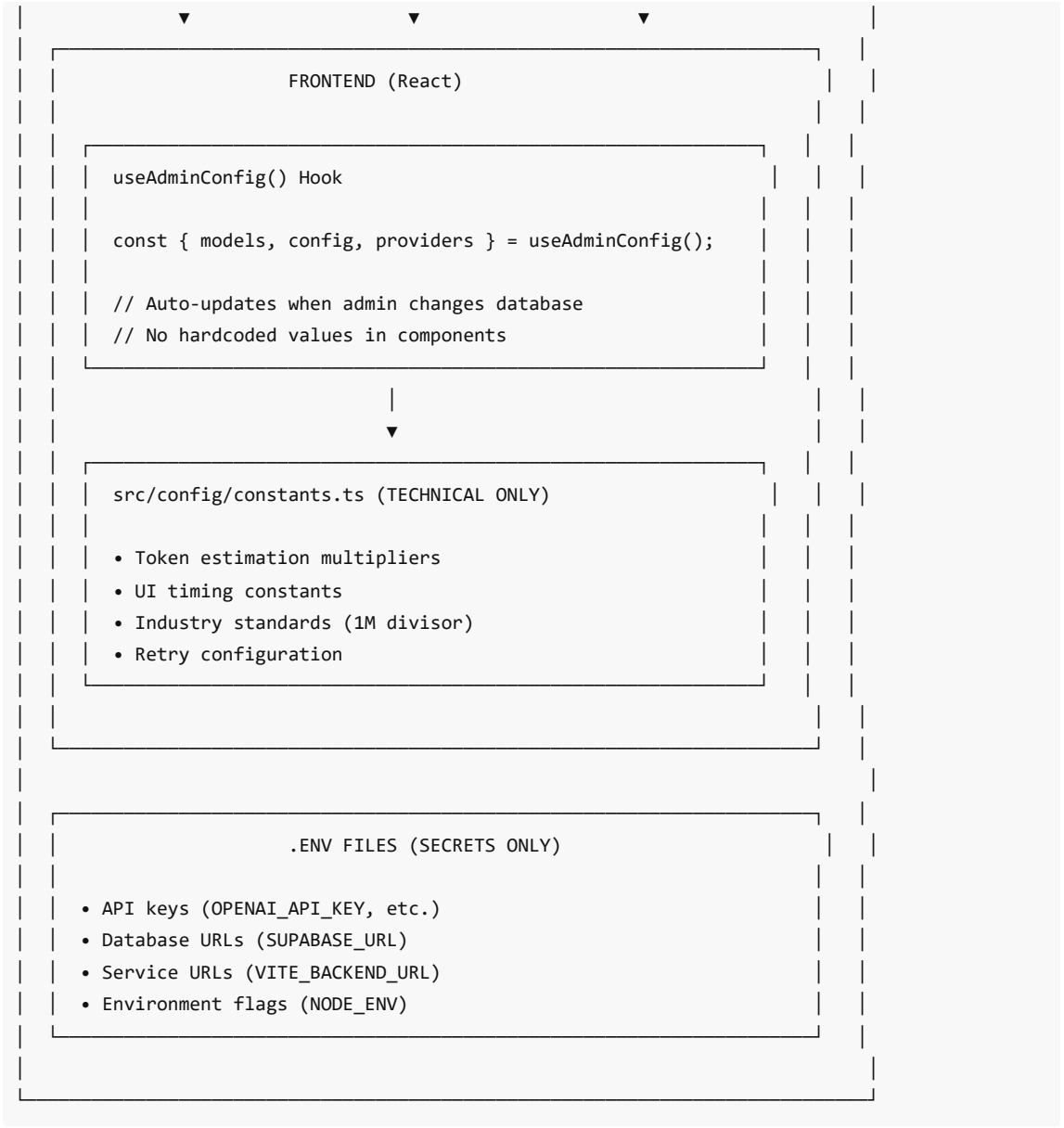
13	PROMPT_CHUNK_SIZE	OneMindAI.tsx:432	4000	**SUPABASE: system_con**
14	MAX_PROMPT_LENGTH	OneMindAI.tsx:~1459	7000	**SUPABASE: system_con**
TOKEN ESTIMATION				
15	tiktoken multiplier	OneMindAI.tsx:322	0.75	**CONFIG FIL**
16	sentencepiece multiplier	OneMindAI.tsx:323	0.95	**CONFIG FIL**
17	bytebpe multiplier	OneMindAI.tsx:324	0.6	**CONFIG FIL**
UI TIMING				
18	Base time offset	OneMindAI.tsx:367	2	**CONFIG FIL**
19	Seconds threshold	OneMindAI.tsx:368	20	**CONFIG FIL**
20	Minutes threshold	OneMindAI.tsx:369	90	**CONFIG FIL**
21	UPDATE_INTERVAL	OneMindAI.tsx:3243	15ms	**CONFIG FIL**
22	STREAM_TIMEOUT	OneMindAI.tsx:3244	30000ms	**SUPABASE: system_con**
ENGINE METADATA				
23	seededEngines	constants.ts:33-180	Full array	**SUPABASE: ai_models**
24	engineInfoText	OneMindAI.tsx:492-600	Descriptions	**SUPABASE: ai_models**
25	modelInfoDictionary	OneMindAI.tsx:583-646	Tooltips	**SUPABASE: ai_models**
26	DEFAULT_SELECTED_ENGINES	constants.ts:257-263	5 engines	**SUPABASE: ai_models**
27	STREAMING_PROVIDERS	constants.ts:269-281	11 providers	**SUPABASE: provider_co**
API CONFIGURATION				
28	Proxy URL fallback	OneMindAI.tsx:~1545	localhost:3002	**.ENV FILE**

29	Default temperature	OneMindAI.tsx:~1740	0.7	**SUPABASE: ai_models**
30	Rate limits	ai-proxy.cjs:61-72	60/min	**SUPABASE: provider_cc**

🏛️ NEW ARCHITECTURE DESIGN {#new-architecture}

Architecture Diagram





Data Flow: Adding a New AI Model (GPT-5 Example)

```
BEFORE (Current - Hardcoded)
═══════════════════════════════════════════════════════════════════════


1. OpenAI announces GPT-5
2. Developer edits constants.ts:
   - Add to SEEDED_ENGINES array
   - Add to BASE_PRICING object
3. Developer edits ai-proxy.cjs:
   - Add token limit
   - Add API routing
4. Developer commits code
5. Developer deploys to production
6. Users see GPT-5
```

```
TIME: 2-4 hours
RISK: Code changes could break other things
AI IMPACT: AI might accidentally modify these files


AFTER (New - Database Controlled)
═══════════════════════════════════════════════════════════════════════

1. OpenAI announces GPT-5
2. Admin opens Admin Panel → AI Models
3. Admin clicks "Add New Model"
4. Admin fills form:
   - ID: gpt-5
   - Display Name: GPT-5
   - Provider: openai
   - API Model ID: gpt-5
   - Max Tokens: 32768
   - Input Price: $5.00
   - Output Price: $15.00
5. Admin clicks "Save"
6. Frontend automatically shows GPT-5 (real-time subscription)

TIME: 2 minutes
RISK: Zero code changes
AI IMPACT: AI cannot modify database values
```

💥 IMPACT ON CURRENT SYSTEM {#impact}

Files That Will Change

File	Current Role	Change Required	Impact Level
`src/core/constants.ts`	All hardcoded values	**DELETE** most content, keep only technical constants	🔴 HIGH
`src/OneMindAI.tsx`	Uses hardcoded values	Replace with `useAdminConfig()` hook	🔴 HIGH
`server/ai-proxy.cjs`	Hardcoded limits	Fetch from database or receive from frontend	🟡 MEDIUM
`src/admin/`	Basic admin	Add Models, Config, Providers pages	🟡 MEDIUM
`supabase/migrations/`	Current schema	Add 3 new tables	🟡 MEDIUM
`src/hooks/useAdminConfig.ts`	**NEW FILE**	Create hook to fetch all config	🟢 NEW

`src/config/constants.ts`	**NEW FILE**	Technical constants only	🟢 NEW

What Stays the Same

```
✓ Component structure (React components)
✓ Styling (TailwindCSS classes)
✓ Business logic (credit calculation formula)
✓ Authentication flow (Supabase Auth)
✓ API proxy architecture (Express server)
✓ Error recovery logic (retry patterns)
✓ File upload handling
✓ Export functionality (PDF/Word)
```

What Changes

```
✗ Model list → Fetched from database
✗ Pricing → Fetched from database
✗ Token limits → Fetched from database
✗ Provider settings → Fetched from database
✗ UX thresholds → Fetched from database
✗ Engine descriptions → Fetched from database
```

📊 MIGRATION PRIORITY MATRIX {#migration-priority}

Priority 1: CRITICAL (Do First)

Item	Why Critical	Effort	Risk if Not Done
AI Models table	New models launch frequently	2 days	Can't add GPT-5 without code change
Pricing in database	Business needs to adjust	1 day	Pricing errors, lost revenue
Token limits in database	Mismatches cause truncation	1 day	User complaints, broken responses

Priority 2: HIGH (Do Soon)

Item	Why High	Effort	Risk if Not Done
Provider config table	Enable/disable providers	1 day	Can't quickly disable broken provider
System config table	Centralize all settings	1 day	Scattered hardcoded values
useAdminConfig hook	Frontend needs data	1 day	Can't use database values

Priority 3: MEDIUM (Do Later)

Item	Why Medium	Effort	Risk if Not Done
Admin UI for models	Business self-service	2 days	Developers still needed for changes
Admin UI for config	Business self-service	1 day	Developers still needed
Real-time subscriptions	Instant updates	0.5 days	Need page refresh

Priority 4: LOW (Nice to Have)

Item	Why Low	Effort	Risk if Not Done
Audit logging	Track who changed what	1 day	No accountability
Config versioning	Rollback capability	2 days	Manual database restore
Config validation	Prevent bad values	1 day	Admin could enter invalid data

---

✅ SUMMARY: THE DECISION

Where Each Category Goes

Category	Location	Reason
AI Models (names, versions, limits, pricing)	**SUPABASE**	Business adds new models
System Config (limits, timeouts, defaults)	**SUPABASE**	Business tunes UX
Provider Config (enable, caps, rates)	**SUPABASE**	Ops enables/disables
API Keys	**.ENV**	Security, per-environment
Service URLs	**.ENV**	Per-environment
Token Estimation	**CONFIG FILE**	Technical, rarely changes
UI Timing	**CONFIG FILE**	Technical, rarely changes
Industry Standards	**CONFIG FILE**	Never changes

Why This Prevents AI From Breaking Admin Values

```
┌─────────────────────────────────────────────────────────────────┐
│                    WHY AI CAN'T BREAK ADMIN VALUES              │
│                                                                 │
│  BEFORE: Values in code files                                   │
│  ──────────────────────────────────────────────────────────     │
│  • AI edits constants.ts → Changes pricing accidentally         │
│  • AI edits OneMindAI.tsx → Changes token limits                │
```

```
│  • AI adds "helpful" hardcoded values → Creates duplicates                  │
│                                                                             │
│  AFTER: Values in database                                                  │
│  ──────────────────────────────────────────────────────────────────────     │
│  • AI edits code files → No business values there to change                 │
│  • AI can only modify code structure, not business data                     │
│  • Database protected by RLS (Row Level Security)                           │
│  • Only admins can modify via Admin Panel                                   │
│                                                                             │
│  PROTECTION LAYERS:                                                         │
│  1. Values not in code → AI can't edit them                                 │
│  2. Database RLS → Only admin role can UPDATE                               │
│  3. Admin Panel validation → Prevents invalid values                        │
│  4. Audit log → Track all changes                                           │
└─────────────────────────────────────────────────────────────────────────────┘
```

Document generated for OneMindAI project
Purpose: Architecture analysis for hardcoded value management