

SWISSVAULT COMPREHENSIVE TECHNICAL ARCHITECTURE REVIEW

Executive Summary

SwissVault (SwissBrain.ai) is an enterprise-grade AI platform built on a hybrid architecture combining:

Lovable Cloud (Supabase) for frontend, database, and edge functions

External Kubernetes cluster (Exoscale Geneva, ch-gva-2) for AI agent worker execution

E2B cloud sandboxes for secure code execution

The platform provides Ghost Chat, Vault Chat, VaultHealth (healthcare AI), Studio (artifact creation), multi-agent orchestration, and comprehensive billing/billing systems.

1. Routes (82 Routes Identified)

Public Routes (Marketing)

Path	Component	Status
/	Index	✓ Working
/auth	Auth	✓ Working
/auth/callback	AuthCallback	✓ Working
/docs/api	ApiDocs	✓ Working
/docs/on-premises	OnPremisesDeployment	✓ Working
/privacy-policy	PrivacyPolicy	✓ Working
/terms-of-service	TermsOfService	✓ Working
/about	About	✓ Working
/contact	Contact	✓ Working
/dpa	DPA	✓ Working
/security	Security	✓ Working
/status	StatusPage	✓ Working
/features/vault-chat	VaultChatFeatures	✓ Working
/features/vault-labs	VaultLabsFeatures	✓ Working
/api-pricing	APIPricing	✓ Working

Ghost Chat Routes (Public/Free)

Path	Component	Status
/ghost	GhostChat	✓ Working
/ghost/chat	GhostChat	✓ Working
/ghost/pricing	GhostPricing	✓ Working
/ghost/library	GhostLibrary	✓ Working (Protected)
/ghost/memory	MemoryDashboard	✓ Working
/ghost/projects	MemoryProjectsPage	✓ Working
/ghost/research-library	ResearchDashboard	✓ Working
/ghost/finance	GhostFinance	✓ Working
/ghost/patents	GhostPatents	✓ Working
/ghost/legal	GhostLegal	✓ Working
/ghost/research	GhostResearch	✓ Working
/ghost/security	GhostSecurity	✓ Working
/ghost/health	GhostHealth	✓ Working
/ghost/travel	GhostTravel	✓ Working
/ghost/realestate	GhostRealEstate	✓ Working
/ghost/art	GhostArt	✓ Working
/ghost/vc	GhostVentureCapital	✓ Working

Protected Chat Routes (Pro+)

Path	Component	Status
/chat	VaultChat (VaultChatGate)	✓ Working (gated)
/chat/:conversationId	VaultChat	✓ Working
/vault/health	VaultHealth	✓ Working

Agent Routes

Path	Component	Status
/agents	Agents	⚠ Partially Working
/agents/dashboard	AgentDashboardNew	⚠ UI works, backend needs tables

```

/agents/studio      AgentsStudio ⚠️ Unknown
/agents/builder     AgentBuilderPage ✅ Working
/agents-dev AgentsDev ⚠️ Dev testing route
/ghost/agents Agents ⚠️ Partially Working
/ghost/agents/studio AgentsStudio ⚠️ Unknown
Labs Routes (Full Platform)
Path Component Status
/labs Dashboard ✅ Working
/labs/projects Projects ✅ Working
/labs/datasets Datasets ✅ Working
/labs/finetuning Finetuning ✅ Working
/labs/templates Templates ✅ Working
/labs/evaluations Evaluations ✅ Working
/labs/models Models ✅ Working
/labs/catalog ModelsCatalog ✅ Working
/labs/playground Playground ✅ Working
/labs/billing Billing ✅ Working
Admin Routes
Path Component Status
/admin AdminDashboardPage ✅ Working
/admin/users UsersManagementPage ✅ Working
/admin/audit-logs AdminAuditLogsPage ✅ Working
/labs/admin Admin ✅ Working
2. Edge Functions (102 Functions)
Registered in config.toml (55 functions)
Function JWT Status Purpose
healthcare-query false ✅ Clean Agentic healthcare AI with Claude tool use
ghost-inference false ✅ Clean Main LLM inference endpoint
ghost-image-gen true ✅ Clean Image generation
ghost-video-gen true ✅ Clean Video generation
ghost-voice false ✅ Clean Voice TTS
ghost-web-search false ✅ Clean Web search
ghost-deep-research false ✅ Clean Deep research
deep-research true ✅ Clean Research engine
encrypted-deep-research true ✅ Clean Encrypted research
agent-execute true ✅ Fixed Agent execution entry
agent-status true ✅ Clean Agent status polling
agent-logs false ✅ Fixed Agent log streaming
agent-plan true ✅ Clean Agent planning
agent-templates-list false ✅ Clean List agent templates
agent-wide-research true ✅ Clean Wide research mode
stripe-webhook false ✅ Clean Stripe webhook handler
create-pro-checkout true ✅ Clean Pro subscription checkout
create-credits-checkout true ✅ Clean Credits purchase
create-ghost-credits-checkout true ✅ Clean Ghost credits
github-oauth false ✅ Clean GitHub OAuth flow
slack-oauth false ✅ Clean Slack OAuth
notion-oauth false ✅ Clean Notion OAuth
gmail-oauth false ✅ Clean Gmail OAuth
googledrive-oauth false ✅ Clean Google Drive OAuth
Functions With Build Errors (6 functions)
Function Error Type Root Cause
calendar-action .catch() on PostgrestFilterBuilder Audit logging uses
invalid method
email-action .catch() on PostgrestFilterBuilder Same as above
github-action .catch() on PostgrestFilterBuilder Same as above
healthcare-appointments never type errors Missing type assertions for Supabase
queries
healthcare-patients never type errors Same

```

healthcare-prescriptions never type errors Same
 Functions NOT Registered in config.toml (47 functions)
 agent-execute-phase2/, agent-logs-phase2/, agent-status-phase2/, agentexecute2/
 healthcare-appointments/, healthcare-patients/, healthcare-prescriptions/,
 healthcare-records/, healthcare-workflows/, healthcare-audit/
 billing-service/, cache-stats/, pool-stats/
 organizations/, oauth-callback/, sso-callback/
 analytics/, audit-logs/, usage-stats/
 Many more...

3. Database Tables (147 tables)

Healthcare Tables ☒

Table	Exists	Purpose
healthcare_audit_log	<input checked="" type="checkbox"/>	Yes HIPAA compliance audit trail
healthcare_conversations	<input checked="" type="checkbox"/>	Yes Health chat history
healthcare_messages	<input checked="" type="checkbox"/>	Yes Health chat messages
healthcare_usage	<input checked="" type="checkbox"/>	Yes Usage tracking

Ghost/Chat Tables ☒

Table	Exists	Purpose
ghost_folders	<input checked="" type="checkbox"/>	Yes Chat folder organization
ghost_library	<input checked="" type="checkbox"/>	Yes Saved items
ghost_settings	<input checked="" type="checkbox"/>	Yes User preferences
ghost_credits	<input checked="" type="checkbox"/>	Yes Credit balances
ghost_usage	<input checked="" type="checkbox"/>	Yes Usage tracking
encrypted_conversations	<input checked="" type="checkbox"/>	Yes E2E encrypted chats
encrypted_messages	<input checked="" type="checkbox"/>	Yes Encrypted messages
encrypted_documents	<input checked="" type="checkbox"/>	Yes Encrypted docs

Agent Tables (Old System) ☒

Table	Exists	Purpose
agent_tasks	<input checked="" type="checkbox"/>	Yes Agent task definitions
agent_task_steps	<input checked="" type="checkbox"/>	Yes Task execution steps
agent_task_logs	<input checked="" type="checkbox"/>	Yes Task logs
agent_sessions	<input checked="" type="checkbox"/>	Yes Agent sessions
agent_messages	<input checked="" type="checkbox"/>	Yes Agent chat messages
agent_tool_calls	<input checked="" type="checkbox"/>	Yes Tool invocations
agent_tool_executions	<input checked="" type="checkbox"/>	Yes Tool execution results
agent_files	<input checked="" type="checkbox"/>	Yes Agent file management
agent_outputs	<input checked="" type="checkbox"/>	Yes Agent outputs
agent_plans	<input checked="" type="checkbox"/>	Yes Agent plans
15+ more agent tables...	<input checked="" type="checkbox"/>	Yes Various agent functionality

Agent Tables (Phase 2 System) ☒ MISSING

Table	Exists	Required By
agent_runs	<input checked="" type="checkbox"/>	NO agentexecute2, agent-execute-phase2
agent_steps	<input checked="" type="checkbox"/>	NO State machine, supervisor
agent_logs	<input checked="" type="checkbox"/>	NO agent-logs, agent-logs-phase2

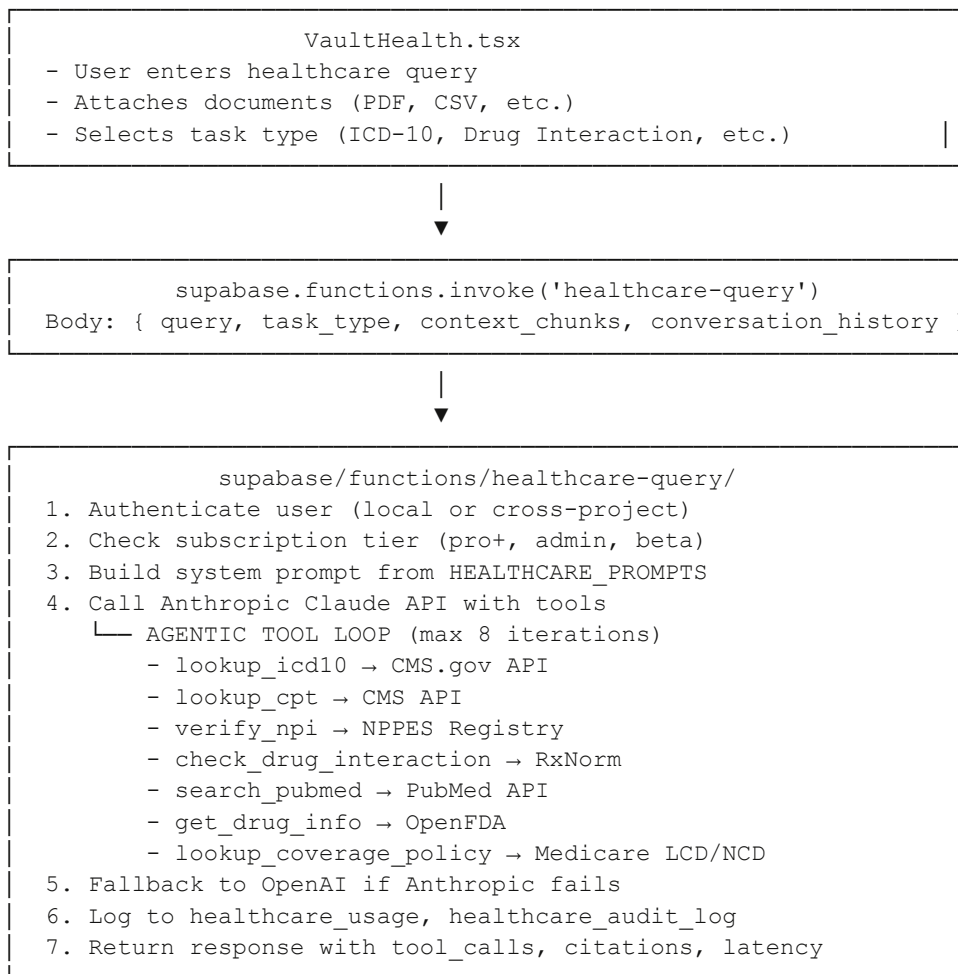
Billing Tables ☒

Table	Exists	Purpose
billing_customers	<input checked="" type="checkbox"/>	Yes Stripe customer mapping
billing_invoices	<input checked="" type="checkbox"/>	Yes Invoice records
credit_transactions	<input checked="" type="checkbox"/>	Yes Credit ledger
user_credits	<input checked="" type="checkbox"/>	Yes User credit balances
unified_credits	<input checked="" type="checkbox"/>	Yes Unified credit system
unified_subscriptions	<input checked="" type="checkbox"/>	Yes Subscription tiers
unified_daily_usage	<input checked="" type="checkbox"/>	Yes Daily usage tracking

Organization Tables ☒

Table	Exists	Purpose
organizations	<input checked="" type="checkbox"/>	Yes Organization entities
organization_members	<input checked="" type="checkbox"/>	Yes Member associations
organization_invitations	<input checked="" type="checkbox"/>	Yes Pending invites
org_roles	<input checked="" type="checkbox"/>	Yes Role definitions


org_role_permissions  Yes Permission mappings
4. VaultHealth Data Flow



Healthcare Tools Implementation 

Location: `supabase/functions/_shared/healthcare-tools/`

`icd10.ts` - ICD-10 lookup via CMS.gov
`cpt.ts` - CPT procedure code lookup
`npi.ts` - NPI verification via NPPES
`rxnorm.ts` - Drug interactions via RxNorm/OpenFDA
`pubmed.ts` - Medical literature search
`executor.ts` - Tool execution router

Memory Integration Gap 

Current State: VaultHealth has a `memoryEnabled` toggle but does NOT import or use `useMemoryContext`. The toggle only updates local state.

Required Fix: Wire `useMemoryContext.getMemoryContext()` into `handleSubmit` to pass memory chunks to the edge function.

5. Build Errors Analysis

Error Category 1: `.catch()` on `PostgrestFilterBuilder` (3 files)

Files:

`supabase/functions/calendar-action/index.ts:568`
`supabase/functions/email-action/index.ts:574`
`supabase/functions/github-action/index.ts:654`
Cause: Supabase PostgREST query builders don't have `.catch()` method.

Fix:

```
// WRONG
await supabase.from('audit_logs').insert({...}).catch(() => {});

// CORRECT
try {
  await supabase.from('audit_logs').insert({...});
} catch { /* Non-critical */ }
Error Category 2: never Type Errors (3 files)
Files:

supabase/functions/healthcare-appointments/index.ts
supabase/functions/healthcare-patients/index.ts
supabase/functions/healthcare-prescriptions/index.ts
Cause: Supabase client with unknown schema types falls back to never.
```

Fix:

```
// Add type alias at top
type AnySupabaseClient = any;

// Use type assertions for queries
const { data } = await supabase
  .from('healthcare_audit_logs')
  .insert(record as any);

6. Supabase Configuration
Frontend Configuration

// src/integrations/supabase/client.ts
const SUPABASE_URL = import.meta.env.VITE_SUPABASE_URL;
const SUPABASE_PUBLISHABLE_KEY = import.meta.env.VITE_SUPABASE_PUBLISHABLE_KEY;
Project IDs
Component      Project ID
Main Supabase (Lovable Cloud)    rljnrngscmosgkcjdvlrq
Anon Key        eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Configured Secrets (45 total)
LLM Providers:

ANTHROPIC_API_KEY ✓
OPENAI_API_KEY ✓
GOOGLE_GEMINI_API_KEY ✓
DEEPSEEK_API_KEY ✓
PERPLEXITY_API_KEY ✓
XAI_API_KEY ✓
Billing:

STRIPE_SECRET_KEY ✓
STRIPE_PUBLISHABLE_KEY ✓
STRIPE_WEBHOOK_SECRET ✓
Integrations:

GITHUB_CLIENT_ID_INTEGRATION ✓
GITHUB_CLIENT_SECRET_INTEGRATION ✓
GOOGLE_CLIENT_ID / GOOGLE_CLIENT_SECRET ✓
SLACK_CLIENT_ID / SLACK_CLIENT_SECRET ✓
NOTION_CLIENT_ID / NOTION_CLIENT_SECRET ✓
```

Infrastructure:

UPSTASH_REDIS_REST_URL / UPSTASH_REDIS_REST_TOKEN

EXOSCALE_API_KEY / EXOSCALE_API_SECRET

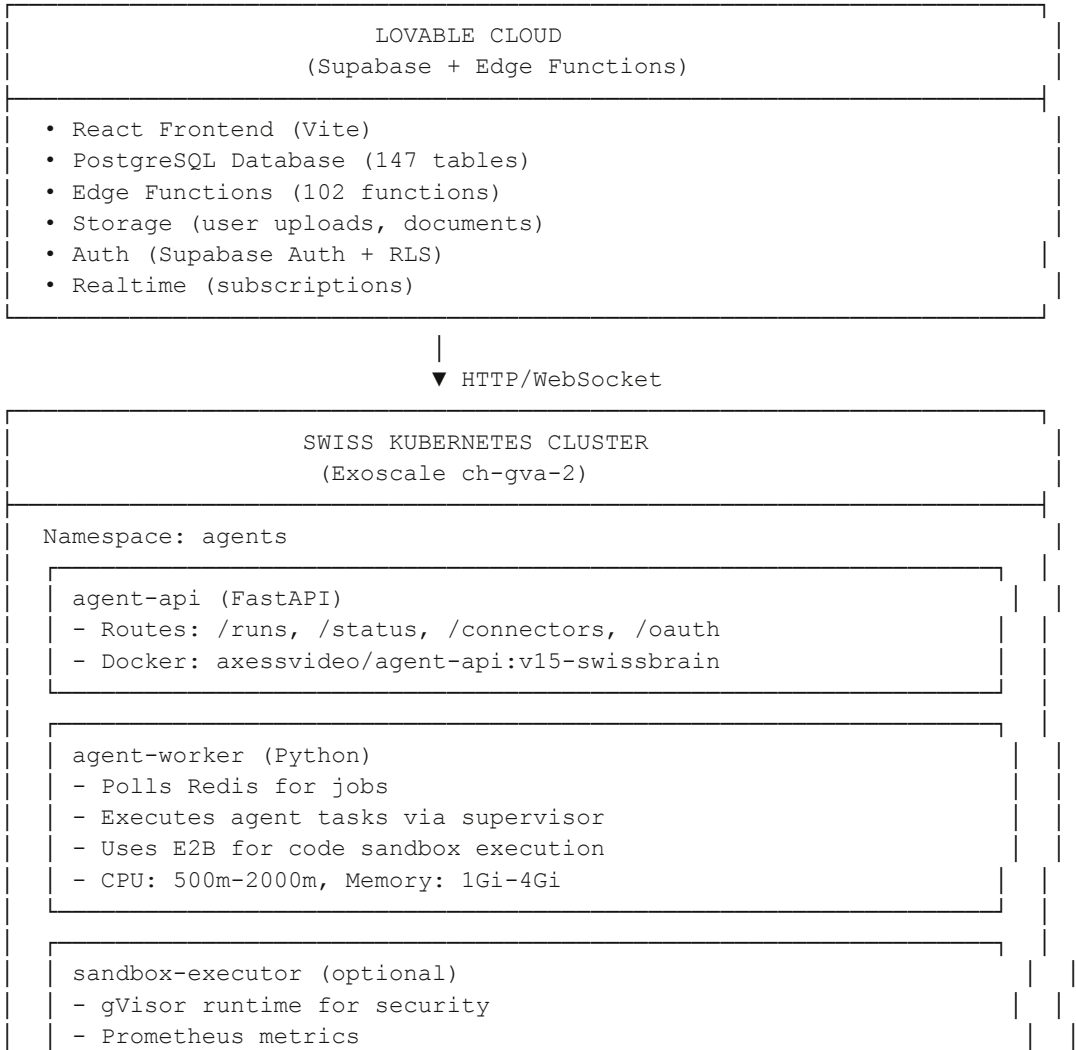
MODAL_ENDPOINT / MODAL_SECRET

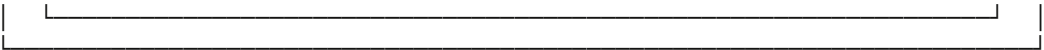
7. Feature Status Matrix

Feature	Has UI	Has Backend	Backend Works	Notes
Ghost Chat			ghost-inference	Fully working
Vault Chat			ghost-inference	Pro+ gated, working
VaultHealth			healthcare-query	7 tools, Claude agentic
Studio		studio-notebooklm		Artifact creation
Memory/RAG			useMemory hooks	Working, not wired to VaultHealth
Projects			knowledge_projects	Working
Billing			Stripe integration	Working
Deep Research			deep-research	Working
Image Gen			ghost-image-gen	Working
Video Gen			ghost-video-gen	Working
Voice/TTS			ghost-voice, gemini-tts	Working
Agents (Old)			agent_tasks tables	Partially working
Agents (Phase 2)			Missing tables	BROKEN - needs migration
OAuth Connectors			github-oauth, etc.	Working
Organizations			organizations table	Working
Admin Dashboard			admin routes	Working

8. Infrastructure Architecture

Hybrid Architecture





EXTERNAL SERVICES	
<ul style="list-style-type: none">• Upstash Redis (job queue, caching)• Exoscale S3 Geneva (workspace storage: swissbrain-workspaces)• E2B Cloud (secure code sandboxes)• Anthropic API (Claude models)• OpenAI API (GPT fallback)• Stripe (billing)• Modal (document generation)	

Kubernetes Configuration

Component	Value
Namespace	agents, swissbrain
Cluster	Exoscale ch-gva-2 (Geneva)
Runtime	gVisor (enhanced security)
Registry	docker.io/axessvideo/, ghcr.io/malenacutuli/
Ingress	TLS via cert-manager
Monitoring	Prometheus + ServiceMonitor
Autoscaling	HPA configured

Agent-API Configuration

```
# Key settings from agent-api/app/config.py
supabase_url: str # Lovable Cloud URL
anthropic_api_key: str # Primary LLM
llm_default_model: str = "claude-sonnet-4-20250514"
redis_url: str # Upstash Redis
s3_endpoint: str = "https://sos-ch-gva-2.exo.io" # Exoscale Geneva
s3_workspace_bucket: str = "swissbrain-workspaces"
k8s_namespace: str = "agents"

9. Conflicts and Duplications
Duplicate Edge Functions
Group Functions Issue
Agent Execute agent-execute/, agentexecute2/, agent-execute-phase2/3 versions,
unclear which is active
Agent Logs agent-logs/, agent-logs-phase2/ 2 versions
Agent Status agent-status/, agent-status-phase2/ 2 versions
Table Naming Conflict
Old System: agent_tasks, agent_task_steps, agent_task_logs
Phase 2 System: agent_runs, agent_steps, agent_logs (DON'T EXIST)
The edge functions reference tables that don't exist yet.
```

Backup Directory

supabase/functions/_shared_backup_old_impl/ - Should be cleaned up

Type Definition Conflicts

src/integrations/supabase/types.ts is auto-generated and may not include all tables
Edge functions use as any type assertions extensively

10. Memory Integration Status

Implementation Status

Component	Exists	Used
useMemory hook	✔ Yes	✔ Ghost Chat
useMemoryContext hook	✔ Yes	✔ Ghost modules
Memory in VaultHealth	✘ Not imported	✘ Toggle is UI-only

Required Wiring


```
// In VaultHealth.tsx handleSubmit():
import { useMemoryContext } from '@/hooks/useMemoryContext';

const { getMemoryContext, isReady: memoryReady } = useMemoryContext();


// Before calling edge function:
if (memoryEnabled && memoryReady) {
  const { sources } = await getMemoryContext(input, { domain: 'health' });
  const memoryChunks = sources.map(s => ({
    filename: `Memory: ${s.title}`,
    content: s.content.slice(0, 8000)
  }));
  contextChunks.push(...memoryChunks);
}

11. Documentation Inventory
Project Root Documentation (100+ files)
Category      Files Notes
Architecture KUBERNETES_ARCHITECTURE.md, DEV_SERVER_ARCHITECTURE.md,
HYBRID_ARCHITECTURE_SETUP.md      Comprehensive
Security      AI_AGENT_SAFETY_RAILS.md, AGENT_ISOLATION_SECURITY.md,
ENCRYPTION_ZERO_KNOWLEDGE.md, PRIVILEGE_ESCALATION_PREVENTION.md    Detailed
Deployment    DEPLOYMENT_STATUS.md, PHASE9_DEPLOYMENT_STATUS.md,
SSL_TLS_MANAGEMENT.md, DNS01_MIGRATION_STATUS.md      Current status
Agent System  SWISS_AGENTS_MASTER_KNOWLEDGE_BASE.md, AGENT_PLANNING_STRATEGY.md,
E2B_V2_ARCHITECTURE.md      Comprehensive
Healthcare    HEALTHCARE_STARTER_PROMPT.md      Basic
Compliance    SWISS_DATA_RESIDENCY_COMPLIANCE.md,
SWISS_COMPLIANCE_IMPLEMENTATION.md, AUDIT_LOGGING_COMPLIANCE.md    Enterprise-ready
docs/ Directory
File Purpose
VAULTHEALTH_ENTERPRISE_ARCHITECTURE.md Enterprise health architecture proposal
KUBERNETES_OPTIMIZATION_GUIDE.md K8s best practices
DOCKER_BUILD_GUIDE.md      Container build process
REDIS_BULLMQ_GUIDE.md      Job queue guide
ENVIRONMENT_MANAGEMENT_GUIDE.md Env var management
PARITY_DELTA.md      Feature comparison
INTEL_INDEX.md      Intel index
agent-api/docs/
Contains Python API documentation and deployment guides.
```


12. Enterprise Considerations

Security 

- Row-Level Security (RLS) on all tables
- AES-256-GCM client-side encryption for Vault Chat
- gVisor container runtime for sandboxes
- OAuth 2.0 for integrations
- Audit logging for HIPAA compliance
- Zero-retention mode option

Billing 

- Stripe integration for subscriptions
- Credit-based usage billing
- Multi-tier system: Free, Pro, Premium, Enterprise
- Token counting and cost attribution
- Admin bypass implemented

Scalability 

- HPA autoscaling on K8s
- Connection pooling
- Redis job queue
- Two-tier storage (hot + cold)

Missing for Enterprise

Rate limiting per tenant (partially implemented)

Multi-region deployment (currently Geneva only)

Disaster recovery plan

SLA monitoring (Prometheus rules defined but not deployed)

13. Summary

Working Features: 18

Ghost Chat, Vault Chat, VaultHealth, Studio, Memory, Projects, Billing, Deep Research, Image/Video Gen, Voice, OAuth Connectors, Organizations, Admin, Labs modules, Healthcare tools, Encryption, Audit logging, SSO

Partially Working: 2

Old Agent System (UI works, execution needs testing)

Admin bypass (just fixed)

Broken Features: 1

Phase 2 Agent System (missing database tables)

Critical Blockers

6 Edge Functions have build errors preventing deployment

3 Database tables missing for Phase 2 agents (agent_runs, agent_steps, agent_logs)

VaultHealth Memory toggle not wired to actual memory system

Recommended Priority Fixes

P0 - Fix Build Errors (30 min):

Replace .catch() with try/catch in calendar-action, email-action, github-action

Add as any type assertions in healthcare-* functions

P1 - Database Decision (1 hour):

Create migration for agent_runs, agent_steps, agent_logs tables

OR refactor Phase 2 functions to use existing agent_tasks tables

P2 - Wire Memory Integration (30 min):

Import useMemoryContext in VaultHealth.tsx

Call getMemoryContext() in handleSubmit when memoryEnabled

P3 - Cleanup (2 hours):

Register critical healthcare functions in config.toml

Delete or archive duplicate agent edge functions

Remove _shared_backup_old_impl/ directory

Now I have enough information to compile the comprehensive diagnostic report. Let me create it:

SWISSVAULT COMPREHENSIVE TECHNICAL ARCHITECTURE REVIEW

Executive Summary

SwissVault (SwissBrain.ai) is an enterprise-grade AI platform built on a hybrid architecture combining:

Lovable Cloud (Supabase) for frontend, database, and edge functions

External Kubernetes cluster (Exoscale Geneva, ch-gva-2) for AI agent worker execution

E2B cloud sandboxes for secure code execution

The platform provides Ghost Chat, Vault Chat, VaultHealth (healthcare AI), Studio (artifact creation), multi-agent orchestration, and comprehensive billing/billing systems.

1. Routes (82 Routes Identified)

Public Routes (Marketing)

Path	Component	Status
------	-----------	--------

/	Index	✓ Working
---	-------	-----------

/auth	Auth	✓ Working
-------	------	-----------

/auth/callback	AuthCallback	✓ Working
----------------	--------------	-----------

/docs/api	ApiDocs	✓ Working
/docs/on-premises	OnPremisesDeployment	✓ Working
/privacy-policy	PrivacyPolicy	✓ Working
/terms-of-service	TermsOfService	✓ Working
/about	About	✓ Working
/contact	Contact	✓ Working
/dpa	DPA	✓ Working
/security	Security	✓ Working
/status	StatusPage	✓ Working
/features/vault-chat	VaultChatFeatures	✓ Working
/features/vault-labs	VaultLabsFeatures	✓ Working
/api-pricing	APIPricing	✓ Working

Ghost Chat Routes (Public/Free)

Path	Component	Status
/ghost	GhostChat	✓ Working
/ghost/chat	GhostChat	✓ Working
/ghost/pricing	GhostPricing	✓ Working
/ghost/library	GhostLibrary	✓ Working (Protected)
/ghost/memory	MemoryDashboard	✓ Working
/ghost/projects	MemoryProjectsPage	✓ Working
/ghost/research-library	ResearchDashboard	✓ Working
/ghost/finance	GhostFinance	✓ Working
/ghost/patents	GhostPatents	✓ Working
/ghost/legal	GhostLegal	✓ Working
/ghost/research	GhostResearch	✓ Working
/ghost/security	GhostSecurity	✓ Working
/ghost/health	GhostHealth	✓ Working
/ghost/travel	GhostTravel	✓ Working
/ghost/realestate	GhostRealEstate	✓ Working
/ghost/art	GhostArt	✓ Working
/ghost/vc	GhostVentureCapital	✓ Working

Protected Chat Routes (Pro+)

Path	Component	Status
/chat	VaultChat (VaultChatGate)	✓ Working (gated)
/chat/:conversationId	VaultChat	✓ Working
/vault/health	VaultHealth	✓ Working

Agent Routes

Path	Component	Status
/agents	Agents	⚠ Partially Working
/agents/dashboard	AgentDashboardNew	⚠ UI works, backend needs tables
/agents/studio	AgentsStudio	⚠ Unknown
/agents/builder	AgentBuilderPage	✓ Working
/agents-dev	AgentsDev	⚠ Dev testing route
/ghost/agents	Agents	⚠ Partially Working
/ghost/agents/studio	AgentsStudio	⚠ Unknown

Labs Routes (Full Platform)

Path	Component	Status
/labs	Dashboard	✓ Working
/labs/projects	Projects	✓ Working
/labs/datasets	Datasets	✓ Working
/labs/finetuning	Finetuning	✓ Working
/labs/templates	Templates	✓ Working
/labs/evaluations	Evaluations	✓ Working
/labs/models	Models	✓ Working
/labs/catalog	ModelsCatalog	✓ Working
/labs/playground	Playground	✓ Working
/labs/billing	Billing	✓ Working

Admin Routes

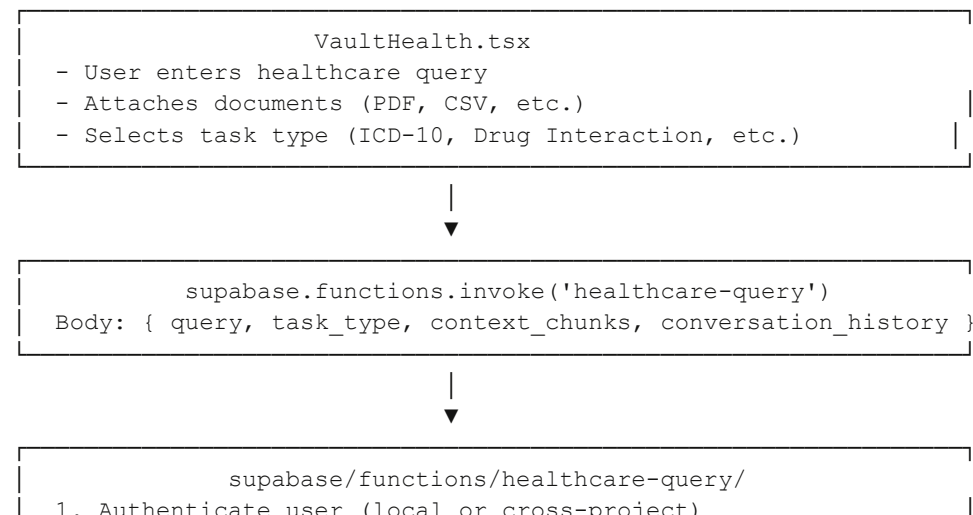
Path	Component	Status
------	-----------	--------

```

/admin AdminDashboardPage ☒ Working
/admin/users UsersManagementPage ☒ Working
/admin/audit-logs AdminAuditLogsPage ☒ Working
/labs/admin Admin ☒ Working
2. Edge Functions (102 Functions)
Registered in config.toml (55 functions)
Function      JWT      Status Purpose
healthcare-query      false ☒ Clean      Agentic healthcare AI with Claude tool use
ghost-inference      false ☒ Clean      Main LLM inference endpoint
ghost-image-gen      true ☒ Clean      Image generation
ghost-video-gen      true ☒ Clean      Video generation
ghost-voice      false ☒ Clean      Voice TTS
ghost-web-search      false ☒ Clean      Web search
ghost-deep-research      false ☒ Clean      Deep research
deep-research      true ☒ Clean      Research engine
encrypted-deep-research      true ☒ Clean      Encrypted research
agent-execute      true ☒ Fixed      Agent execution entry
agent-status      true ☒ Clean      Agent status polling
agent-logs      false ☒ Fixed      Agent log streaming
agent-plan      true ☒ Clean      Agent planning
agent-templates-list      false ☒ Clean      List agent templates
agent-wide-research      true ☒ Clean      Wide research mode
stripe-webhook      false ☒ Clean      Stripe webhook handler
create-pro-checkout      true ☒ Clean      Pro subscription checkout
create-credits-checkout      true ☒ Clean      Credits purchase
create-ghost-credits-checkout      true ☒ Clean      Ghost credits
github-oauth      false ☒ Clean      GitHub OAuth flow
slack-oauth      false ☒ Clean      Slack OAuth
notion-oauth      false ☒ Clean      Notion OAuth
gmail-oauth      false ☒ Clean      Gmail OAuth
googledrive-oauth      false ☒ Clean      Google Drive OAuth
Functions With Build Errors (6 functions)
Function      Error Type      Root Cause
calendar-action      .catch() on PostgrestFilterBuilder      Audit logging uses
invalid method
email-action      .catch() on PostgrestFilterBuilder      Same as above
github-action      .catch() on PostgrestFilterBuilder      Same as above
healthcare-appointments      never type errors      Missing type assertions for Supabase
queries
healthcare-patients      never type errors      Same
healthcare-prescriptions      never type errors      Same
Functions NOT Registered in config.toml (47 functions)
agent-execute-phase2/, agent-logs-phase2/, agent-status-phase2/, agentexecute2/
healthcare-appointments/, healthcare-patients/, healthcare-prescriptions/,
healthcare-records/, healthcare-workflows/, healthcare-audit/
billing-service/, cache-stats/, pool-stats/
organizations/, oauth-callback/, sso-callback/
analytics/, audit-logs/, usage-stats/
Many more...
3. Database Tables (147 tables)
Healthcare Tables ☒
Table      Exists Purpose
healthcare_audit_log      ☒ Yes HIPAA compliance audit trail
healthcare_conversations      ☒ Yes Health chat history
healthcare_messages      ☒ Yes Health chat messages
healthcare_usage      ☒ Yes Usage tracking
Ghost/Chat Tables ☒
Table      Exists Purpose
ghost_folders      ☒ Yes Chat folder organization

```

ghost_library ☒ Yes Saved items
ghost_settings ☒ Yes User preferences
ghost_credits ☒ Yes Credit balances
ghost_usage ☒ Yes Usage tracking
encrypted_conversations ☒ Yes E2E encrypted chats
encrypted_messages ☒ Yes Encrypted messages
encrypted_documents ☒ Yes Encrypted docs
Agent Tables (Old System) ☒
Table Exists Purpose
agent_tasks ☒ Yes Agent task definitions
agent_task_steps ☒ Yes Task execution steps
agent_task_logs ☒ Yes Task logs
agent_sessions ☒ Yes Agent sessions
agent_messages ☒ Yes Agent chat messages
agent_tool_calls ☒ Yes Tool invocations
agent_tool_executions ☒ Yes Tool execution results
agent_files ☒ Yes Agent file management
agent_outputs ☒ Yes Agent outputs
agent_plans ☒ Yes Agent plans
15+ more agent tables... ☒ Yes Various agent functionality
Agent Tables (Phase 2 System) ☒ MISSING
Table Exists Required By
agent_runs ☒ NO agentexecute2, agent-execute-phase2
agent_steps ☒ NO State machine, supervisor
agent_logs ☒ NO agent-logs, agent-logs-phase2
Billing Tables ☒
Table Exists Purpose
billing_customers ☒ Yes Stripe customer mapping
billing_invoices ☒ Yes Invoice records
credit_transactions ☒ Yes Credit ledger
user_credits ☒ Yes User credit balances
unified_credits ☒ Yes Unified credit system
unified_subscriptions ☒ Yes Subscription tiers
unified_daily_usage ☒ Yes Daily usage tracking
Organization Tables ☒
Table Exists Purpose
organizations ☒ Yes Organization entities
organization_members ☒ Yes Member associations
organization_invitations ☒ Yes Pending invites
org_roles ☒ Yes Role definitions
org_role_permissions ☒ Yes Permission mappings
4. VaultHealth Data Flow




2. Check subscription tier (pro+, admin, beta)
3. Build system prompt from HEALTHCARE_PROMPTS
4. Call Anthropic Claude API with tools
 - └─ AGENTIC TOOL LOOP (max 8 iterations)
 - lookup_icd10 → CMS.gov API
 - lookup_cpt → CMS API
 - verify_npi → NPPES Registry
 - check_drug_interaction → RxNorm
 - search_pubmed → PubMed API
 - get_drug_info → OpenFDA
 - lookup_coverage_policy → Medicare LCD/NCD
5. Fallback to OpenAI if Anthropic fails
6. Log to healthcare_usage, healthcare_audit_log
7. Return response with tool_calls, citations, latency

Healthcare Tools Implementation 

Location: supabase/functions/_shared/healthcare-tools/

icd10.ts - ICD-10 lookup via CMS.gov
 cpt.ts - CPT procedure code lookup
 npi.ts - NPI verification via NPPES
 rxnorm.ts - Drug interactions via RxNorm/OpenFDA
 pubmed.ts - Medical literature search
 executor.ts - Tool execution router

Memory Integration Gap 

Current State: VaultHealth has a memoryEnabled toggle but does NOT import or use useMemoryContext. The toggle only updates local state.

Required Fix: Wire useMemoryContext.getMemoryContext() into handleSubmit to pass memory chunks to the edge function.

5. Build Errors Analysis

Error Category 1: .catch() on PostgrestFilterBuilder (3 files)

Files:

supabase/functions/calendar-action/index.ts:568
 supabase/functions/email-action/index.ts:574
 supabase/functions/github-action/index.ts:654
 Cause: Supabase PostgREST query builders don't have .catch() method.

Fix:

```
// WRONG
await supabase.from('audit_logs').insert({...}).catch(() => {});
```

```
// CORRECT
try {
  await supabase.from('audit_logs').insert({...});
} catch { /* Non-critical */ }
```

Error Category 2: never Type Errors (3 files)

Files:

supabase/functions/healthcare-appointments/index.ts
 supabase/functions/healthcare-patients/index.ts
 supabase/functions/healthcare-prescriptions/index.ts
 Cause: Supabase client with unknown schema types falls back to never.

Fix:

```

// Add type alias at top
type AnySupabaseClient = any;

// Use type assertions for queries
const { data } = await supabase
  .from('healthcare_audit_logs')
  .insert(record as any);
6. Supabase Configuration
Frontend Configuration

// src/integrations/supabase/client.ts
const SUPABASE_URL = import.meta.env.VITE_SUPABASE_URL;
const SUPABASE_PUBLISHABLE_KEY = import.meta.env.VITE_SUPABASE_PUBLISHABLE_KEY;
Project IDs
Component      Project ID
Main Supabase (Lovable Cloud)      rljnrsgscmosgkcjdvlrq
Anon Key      eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Configured Secrets (45 total)
LLM Providers:

ANTHROPIC_API_KEY ✓
OPENAI_API_KEY ✓
GOOGLE_GEMINI_API_KEY ✓
DEEPSEEK_API_KEY ✓
PERPLEXITY_API_KEY ✓
XAI_API_KEY ✓
Billing:

STRIPE_SECRET_KEY ✓
STRIPE_PUBLISHABLE_KEY ✓
STRIPE_WEBHOOK_SECRET ✓
Integrations:

GITHUB_CLIENT_ID_INTEGRATION ✓
GITHUB_CLIENT_SECRET_INTEGRATION ✓
GOOGLE_CLIENT_ID / GOOGLE_CLIENT_SECRET ✓
SLACK_CLIENT_ID / SLACK_CLIENT_SECRET ✓
NOTION_CLIENT_ID / NOTION_CLIENT_SECRET ✓
Infrastructure:

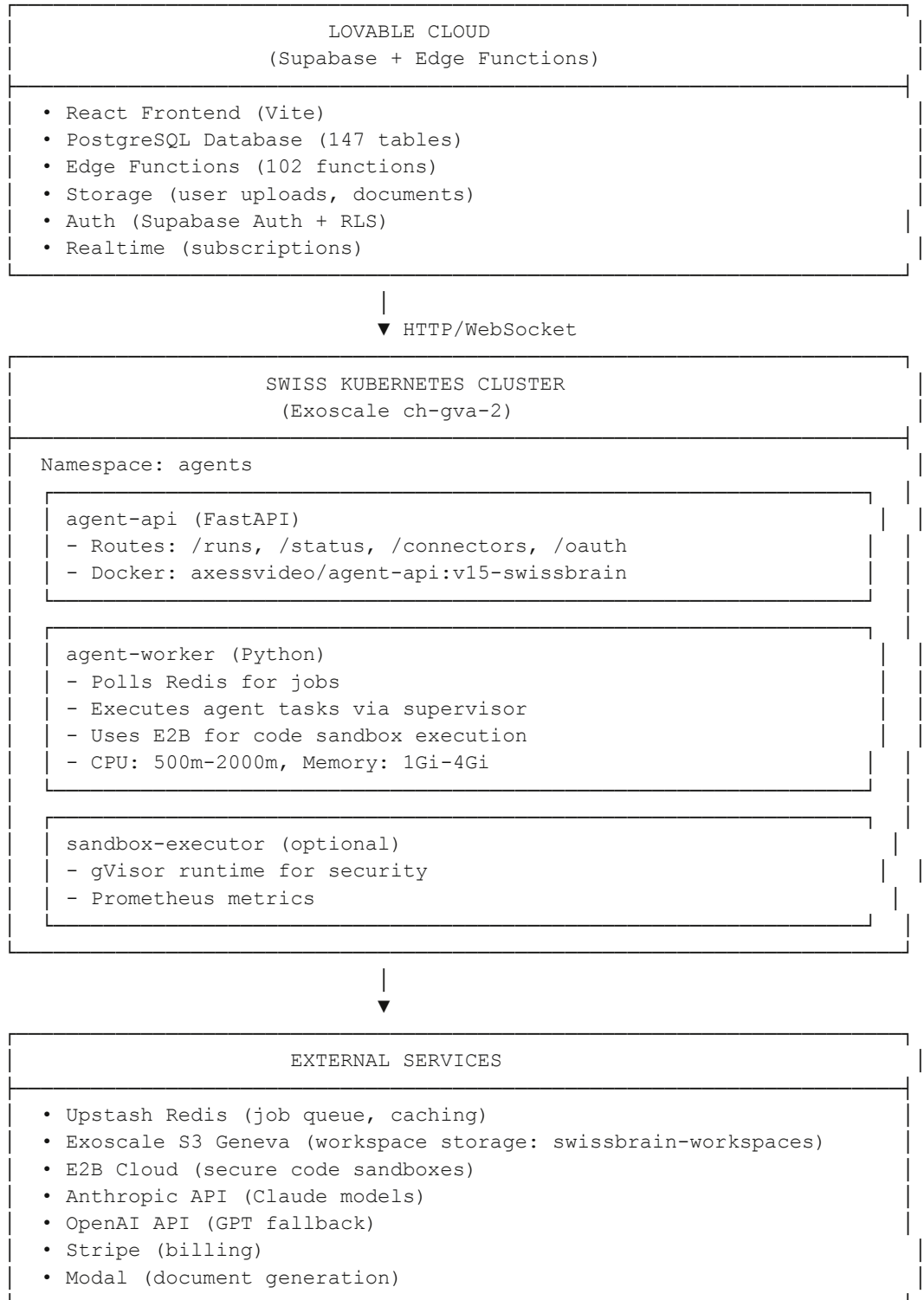
UPSTASH_REDIS_REST_URL / UPSTASH_REDIS_REST_TOKEN ✓
EXOSCALE_API_KEY / EXOSCALE_API_SECRET ✓
MODAL_ENDPOINT / MODAL_SECRET ✓
7. Feature Status Matrix
Feature      Has UI  Has Backend  Backend Works Notes
Ghost Chat   ✓       ✓      ghost-inference ✓      Fully working
Vault Chat   ✓       ✓      ghost-inference ✓      Pro+ gated, working
VaultHealth  ✓       ✓      healthcare-query ✓      7 tools, Claude agentic
Studio ✓     ✓       studio-notebooklm ✓      Artifact creation
Memory/RAG   ✓       ✓      useMemory hooks ✓      Working, not wired to VaultHealth
Projects     ✓       ✓      knowledge_projects ✓      Working
Billing      ✓       ✓      Stripe integration ✓      Working
Deep Research ✓     ✓      deep-research ✓      Working
Image Gen    ✓       ✓      ghost-image-gen ✓      Working
Video Gen    ✓       ✓      ghost-video-gen ✓      Working
Voice/TTS    ✓       ✓      ghost-voice, gemini-tts ✓      Working
Agents (Old) ✓     ✓      agent_tasks tables ⚠      Partially working

```

Agents (Phase 2)	✓	✗ Missing tables	✗	BROKEN - needs migration
OAuth Connectors	✓	✓ github-oauth, etc.	✓	Working
Organizations	✓	✓ organizations table	✓	Working
Admin Dashboard	✓	✓ admin routes	✓	Working

8. Infrastructure Architecture

Hybrid Architecture



Kubernetes Configuration

Component	Value
Namespace	agents, swissbrain
Cluster	Exoscale ch-gva-2 (Geneva)

Runtime gVisor (enhanced security)
Registry docker.io/axessvideo/, ghcr.io/malenacutuli/
Ingress TLS via cert-manager
Monitoring Prometheus + ServiceMonitor
Autoscaling HPA configured
Agent-API Configuration

```
# Key settings from agent-api/app/config.py
supabase_url: str                               # Lovable Cloud URL
anthropic_api_key: str                         # Primary LLM
llm_default_model: str = "claude-sonnet-4-20250514"
redis_url: str                                 # Upstash Redis
s3_endpoint: str = "https://sos-ch-gva-2.exo.io" # Exoscale Geneva
s3_workspace_bucket: str = "swissbrain-workspaces"
k8s_namespace: str = "agents"

9. Conflicts and Duplications
Duplicate Edge Functions
Group   Functions    Issue
Agent Execute agent-execute/, agentexecute2/, agent-execute-phase2/3 versions,
unclear which is active
Agent Logs   agent-logs/, agent-logs-phase2/   2 versions
Agent Status agent-status/, agent-status-phase2/   2 versions
Table Naming Conflict
Old System: agent_tasks, agent_task_steps, agent_task_logs
Phase 2 System: agent_runs, agent_steps, agent_logs (DON'T EXIST)
The edge functions reference tables that don't exist yet.
```

Backup Directory

supabase/functions/_shared_backup_old_impl/ - Should be cleaned up

Type Definition Conflicts

src/integrations/supabase/types.ts is auto-generated and may not include all tables

Edge functions use as any type assertions extensively

10. Memory Integration Status

Implementation Status

Component	Exists	Used
-----------	--------	------

useMemory hook	✅ Yes	✅ Ghost Chat
----------------	-------	--------------

useMemoryContext hook	✅ Yes	✅ Ghost modules
-----------------------	-------	-----------------

Memory in VaultHealth	❌ Not imported	❌ Toggle is UI-only
-----------------------	----------------	---------------------

Required Wiring

```
// In VaultHealth.tsx handleSubmit():
```

```
import { useMemoryContext } from '@/hooks/useMemoryContext';
```

```
const { getMemoryContext, isReady: memoryReady } = useMemoryContext();
```

```
// Before calling edge function:
```

```
if (memoryEnabled && memoryReady) {
  const { sources } = await getMemoryContext(input, { domain: 'health' });
  const memoryChunks = sources.map(s => ({
    filename: `Memory: ${s.title}`,
    content: s.content.slice(0, 8000)
  }));
  contextChunks.push(...memoryChunks);
}
```

11. Documentation Inventory

Project Root Documentation (100+ files)

Category	Files	Notes
----------	-------	-------

Architecture	KUBERNETES_ARCHITECTURE.md, DEV_SERVER_ARCHITECTURE.md,	
--------------	---	--

HYBRID_ARCHITECTURE_SETUP.md	Comprehensive	
------------------------------	---------------	--

Security AI_AGENT_SAFETY_RAILS.md, AGENT_ISOLATION_SECURITY.md,
 ENCRYPTION_ZERO_KNOWLEDGE.md, PRIVILEGE_ESCALATION_PREVENTION.md Detailed
 Deployment DEPLOYMENT_STATUS.md, PHASE9_DEPLOYMENT_STATUS.md,
 SSL_TLS_MANAGEMENT.md, DNS01_MIGRATION_STATUS.md Current status
 Agent System SWISS_AGENTS_MASTER_KNOWLEDGE_BASE.md, AGENT_PLANNING_STRATEGY.md,
 E2B_V2_ARCHITECTURE.md Comprehensive
 Healthcare HEALTHCARE_STARTER_PROMPT.md Basic
 Compliance SWISS_DATA_RESIDENCY_COMPLIANCE.md,
 SWISS_COMPLIANCE_IMPLEMENTATION.md, AUDIT_LOGGING_COMPLIANCE.md Enterprise-ready
 docs/ Directory
 File Purpose
 VAULTHEALTH_ENTERPRISE_ARCHITECTURE.md Enterprise health architecture proposal
 KUBERNETES_OPTIMIZATION_GUIDE.md K8s best practices
 DOCKER_BUILD_GUIDE.md Container build process
 REDIS_BULLMQ_GUIDE.md Job queue guide
 ENVIRONMENT_MANAGEMENT_GUIDE.md Env var management
 PARITY_DELTA.md Feature comparison
 INTEL_INDEX.md Intel index
 agent-api/docs/
 Contains Python API documentation and deployment guides.

12. Enterprise Considerations

Security

Row-Level Security (RLS) on all tables
 AES-256-GCM client-side encryption for Vault Chat
 gVisor container runtime for sandboxes
 OAuth 2.0 for integrations
 Audit logging for HIPAA compliance
 Zero-retention mode option

Billing

Stripe integration for subscriptions
 Credit-based usage billing
 Multi-tier system: Free, Pro, Premium, Enterprise
 Token counting and cost attribution
 Admin bypass implemented

Scalability

HPA autoscaling on K8s
 Connection pooling
 Redis job queue
 Two-tier storage (hot + cold)
 Missing for Enterprise
 Rate limiting per tenant (partially implemented)
 Multi-region deployment (currently Geneva only)
 Disaster recovery plan
 SLA monitoring (Prometheus rules defined but not deployed)

13. Summary

Working Features: 18

Ghost Chat, Vault Chat, VaultHealth, Studio, Memory, Projects, Billing, Deep
 Research, Image/Video Gen, Voice, OAuth Connectors, Organizations, Admin, Labs
 modules, Healthcare tools, Encryption, Audit logging, SSO

Partially Working: 2

Old Agent System (UI works, execution needs testing)

Admin bypass (just fixed)

Broken Features: 1

Phase 2 Agent System (missing database tables)

Critical Blockers

6 Edge Functions have build errors preventing deployment

3 Database tables missing for Phase 2 agents (agent_runs, agent_steps, agent_logs)

VaultHealth Memory toggle not wired to actual memory system

Recommended Priority Fixes

P0 - Fix Build Errors (30 min):

Replace `.catch()` with `try/catch` in `calendar-action`, `email-action`, `github-action`
Add as many type assertions in `healthcare-*` functions

P1 - Database Decision (1 hour):

Create migration for `agent_runs`, `agent_steps`, `agent_logs` tables
OR refactor Phase 2 functions to use existing `agent_tasks` tables

P2 - Wire Memory Integration (30 min):

Import `useMemoryContext` in `VaultHealth.tsx`
Call `getMemoryContext()` in `handleSubmit` when `memoryEnabled`
P3 - Cleanup (2 hours):

Register critical healthcare functions in `config.toml`
Delete or archive duplicate agent edge functions
Remove `_shared_backup_old_impl/` directory
Plan
SwissVault Technical Architecture Fix Plan
Phase 1: Fix Critical Build Errors (Immediate)
1.1 Fix `.catch()` Errors (3 files)
Files: `calendar-action/index.ts`, `email-action/index.ts`, `github-action/index.ts`

Replace all instances of:

```
await supabase.from('audit_logs').insert({...}).catch(() => {});
```

With:

```
try {
  await supabase.from('audit_logs').insert({...});
} catch { /* Non-critical */ }
```

1.2 Fix Healthcare Function Type Errors (3 files)
Files: `healthcare-appointments/index.ts`, `healthcare-patients/index.ts`,
`healthcare-prescriptions/index.ts`

Add type assertions:

```
// At top of file
type AnySupabaseClient = any;

// For function parameters
async function someHelper(supabase: AnySupabaseClient, ...)

// For inserts
await serviceClient.from('healthcare_audit_logs').insert(auditRecord as any);

// For array access
const patient = apt.patient as any;
```

Phase 2: Create Missing Agent Tables (Database Migration)
Create a migration to add the missing Phase 2 agent tables:

```
-- Create agent_runs table (Phase 2 agent execution)
CREATE TABLE agent_runs (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
```

```

    user_id UUID REFERENCES auth.users(id) ON DELETE CASCADE NOT NULL,
    project_id UUID,
    prompt TEXT NOT NULL,
    status TEXT NOT NULL DEFAULT 'pending',
    current_phase INTEGER DEFAULT 0,
    total_credits_used NUMERIC DEFAULT 0,
    plan JSONB DEFAULT '{}',
    result JSONB,
    error TEXT,
    metadata JSONB DEFAULT '{}',
    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW(),
    completed_at TIMESTAMPTZ
);

-- Create agent_steps table
CREATE TABLE agent_steps (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    run_id UUID REFERENCES agent_runs(id) ON DELETE CASCADE NOT NULL,
    step_number INTEGER NOT NULL,
    action TEXT NOT NULL,
    tool_name TEXT,
    tool_input JSONB,
    tool_output JSONB,
    status TEXT DEFAULT 'pending',
    credits_used NUMERIC DEFAULT 0,
    started_at TIMESTAMPTZ,
    completed_at TIMESTAMPTZ,
    created_at TIMESTAMPTZ DEFAULT NOW()
);

-- Create agent_logs table
CREATE TABLE agent_logs (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    run_id UUID REFERENCES agent_runs(id) ON DELETE CASCADE NOT NULL,
    level TEXT NOT NULL DEFAULT 'info',
    message TEXT NOT NULL,
    metadata JSONB DEFAULT '{}',
    created_at TIMESTAMPTZ DEFAULT NOW()
);

-- Enable RLS
ALTER TABLE agent_runs ENABLE ROW LEVEL SECURITY;
ALTER TABLE agent_steps ENABLE ROW LEVEL SECURITY;
ALTER TABLE agent_logs ENABLE ROW LEVEL SECURITY;

-- RLS Policies
CREATE POLICY "Users can manage their own agent runs"
    ON agent_runs FOR ALL USING (user_id = auth.uid());

CREATE POLICY "Users can view steps of their runs"
    ON agent_steps FOR ALL
    USING (run_id IN (SELECT id FROM agent_runs WHERE user_id = auth.uid()));

CREATE POLICY "Users can view logs of their runs"
    ON agent_logs FOR ALL
    USING (run_id IN (SELECT id FROM agent_runs WHERE user_id = auth.uid()));

-- Indexes

```

```

CREATE INDEX idx_agent_runs_user ON agent_runs(user_id);
CREATE INDEX idx_agent_runs_status ON agent_runs(status);
CREATE INDEX idx_agent_steps_run ON agent_steps(run_id);
CREATE INDEX idx_agent_logs_run ON agent_logs(run_id, created_at);

-- Enable realtime for logs
ALTER PUBLICATION supabase_realtime ADD TABLE agent_logs;
Phase 3: Wire Memory Integration to VaultHealth
3.1 Update VaultHealth.tsx
Add import:

import { useMemoryContext } from '@/hooks/useMemoryContext';
Add hook usage:

const { getMemoryContext, isReady: memoryReady } = useMemoryContext();
Modify handleSubmit before edge function call:

// Gather memory context if enabled
let memoryChunks: Array<{ filename: string; content: string }> = [];
if (memoryEnabled && memoryReady) {
  try {
    const { sources } = await getMemoryContext(input, {
      domain: 'health',
      limit: 5,
      minScore: 0.3
    });
    memoryChunks = sources.map(s => ({
      filename: `Memory: ${s.title}`,
      content: s.content.slice(0, 8000)
    }));
  } catch (err) {
    console.error('[VaultHealth] Memory context error:', err);
  }
}

// Combine with document context
const allContextChunks = [...contextChunks, ...memoryChunks];

// Pass to edge function
const { data, error } = await supabase.functions.invoke('healthcare-query', {
  body: {
    query: userMessage.content,
    task_type: taskType,
    context_chunks: allContextChunks, // Now includes memory
    conversation_history: conversationHistory,
  },
});
Phase 4: Register Missing Edge Functions
Update supabase/config.toml to add:

[
functions.healthcare-appointments
]
verify_jwt = false

```

```
[
functions.healthcare-patients
]
verify_jwt = false
```

```
[
functions.healthcare-prescriptions
]
verify_jwt = false
```

```
[
functions.healthcare-records
]
verify_jwt = false
```

```
[
functions.healthcare-audit
]
verify_jwt = false
Phase 5: Cleanup (Optional)
Delete backup directory: supabase/functions/_shared_backup_old_impl/
Archive or remove duplicate functions once Phase 2 agents are working:
Keep agent-execute/ as primary
Archive agentexecute2/ and agent-execute-phase2/
Update documentation with current architecture decisions
Expected Outcomes
```

Item	Before	After
Edge function build errors	6	0
Missing agent tables	3	0
VaultHealth memory integration	Not connected	Fully wired
Registered healthcare functions	1	6
Duplicate function confusion	3 groups	Documented decision

Files to Modify

File	Action	Change
supabase/functions/calendar-action/index.ts	MODIFY	Fix .catch() → try/catch
supabase/functions/email-action/index.ts	MODIFY	Fix .catch() → try/catch
supabase/functions/github-action/index.ts	MODIFY	Fix .catch() → try/catch
supabase/functions/healthcare-appointments/index.ts	MODIFY	Add type assertions
supabase/functions/healthcare-patients/index.ts	MODIFY	Add type assertions
supabase/functions/healthcare-prescriptions/index.ts	MODIFY	Add type assertions
src/pages/vault/VaultHealth.tsx	MODIFY	Wire useMemoryContext
supabase/config.toml	MODIFY	Register healthcare functions
Database Migration	CREATE	Add agent_runs, agent_steps, agent_logs tables