VCTT-AGI System Architecture

Overview

The VCTT-AGI Engine (MIN) is a next-generation, multi-agent AGI IDE with a distributed architecture across multiple hosting platforms. This document clarifies the deployment topology to prevent configuration errors.

Production Deployment

Frontend

- **Platform:** Vercel
- **URL:** https://vctt-agi-ui.vercel.app
- **Repository:** https://github.com/Counterbalance-Economics/vctt-agi-ui
- **Tech Stack:** React + Next.js
- **Purpose:** Live production interface for end users

Backend

- **Platform:** Render
- **URL:** https://vctt-agi-backend.onrender.com
- **Repository:** https://github.com/Counterbalance-Economics/vctt-agi-engine
- **Tech Stack:** NestJS + TypeScript
- **Database:** Render PostgreSQL (production)
- **Purpose:** Live production API serving the frontend

Configuration

- Frontend `VITE_BACKEND_URL` → `https://vctt-agi-backend.onrender.com`
- Backend env vars:

  ```
  bash
    DEPLOYMENT_PLATFORM=render
    DEPLOYMENT_ROLE=production
    INSTANCE_NAME=vctt-agi-render-production
    CONNECTED_FRONTEND=https://vctt-agi-ui.vercel.app
  ```

Development/Testing Deployment

Backend (Development Instance)

- **Platform:** Abacus.AI
- **URL:** https://vctt-agi-phase3-complete.abacusai.app
- **Repository:** Same as production (different deployment target)
- **Tech Stack:** NestJS + TypeScript

- **Database:** Abacus.AI PostgreSQL (development)
- **Purpose:** Testing, prototyping, agent experimentation, code development

Configuration

- Backend env vars:
  ```bash
    DEPLOYMENT_PLATFORM=abacus-ai
    DEPLOYMENT_ROLE=development
    INSTANCE_NAME=vctt-agi-abacus-dev
    CONNECTED_FRONTEND=DO_NOT_CONNECT_PROD_FRONTEND
  ```

⚠️ CRITICAL WARNINGS

- **DO NOT** connect the production frontend to the Abacus.AI backend
- **DO NOT** mix production and development database connections
- **DO NOT** deploy development code to production without testing

---

System Identity Verification

How to Check Which Instance You're On

1. **Call the metadata endpoint:**
   ```bash
      curl https://[backend-url]/api/health/metadata
   ```

2. **Check the response:**
   ```json
      {
        "instance_id": "vctt-agi-render-production" | "vctt-agi-abacus-dev",
        "deployment_platform": "render" | "abacus-ai",
        "role": "production" | "development",
        "connected_services": {
          "frontend_url": "https://vctt-agi-ui.vercel.app" | "DO_NOT_CONNECT_PROD_FRONTEND"
        }
      }
   ```

3. **Check startup logs:**
   - Look for the ASCII art banner showing instance identity
   - Verify `Role:` matches expected environment (production/development)

Key Rules for Safe Deployment

✅ DO:

- Always call `/api/health/metadata` before making configuration changes
- Verify instance identity matches your intent (production vs development)
- Check startup banner in logs to confirm correct deployment
- Use environment variables to configure instance identity
- Test changes on development instance before deploying to production

❌ DO NOT:

- Modify frontend’s backend URL without confirming metadata first
- Assume a backend URL is production just because it responds
- Connect production frontend to development backend
- Deploy without verifying instance identity
- Skip the metadata check when troubleshooting

Emergency Procedures

If Frontend Connects to Wrong Backend:

1. Check current backend URL in frontend’s `src/config/api.ts`
2. Call `/api/health/metadata` on both backends to identify them
3. Update frontend to correct backend URL
4. Commit and push to trigger Vercel redeploy
5. Wait ~90 seconds for deployment
6. Verify frontend is working correctly

If Backend Identity is Unknown:

1. Check startup logs for banner (if available)
2. Call `/api/health/metadata` endpoint
3. If metadata returns “unknown” values, set environment variables:
   - `DEPLOYMENT_PLATFORM`
   - `DEPLOYMENT_ROLE`
   - `INSTANCE_NAME`
   - `CONNECTED_FRONTEND`
4. Restart backend to reload env vars
5. Verify banner and metadata are now correct

Maintenance Notes

When Adding New Deployment Instances:

1. Set all four identity env vars
2. Update this ARCHITECTURE.md file
3. Add new instance to monitoring/logging
4. Document purpose and connected services
5. Test metadata endpoint returns correct info

When Changing Backend URLs:

1. **STOP** - Don’t change URLs without verification
2. Call `/api/health/metadata` on target backend
3. Confirm `role` and `connected_services.frontend_url`
4. If role is “development”, DO NOT connect production frontend
5. If role is “production”, verify it’s the intended production backend

6. Update configuration only after verification
7. Monitor logs for errors after deployment

Contact & Support

For architecture questions or deployment issues, refer to:
- System Integrity Service logs
- `/api/health/metadata` endpoint
- Startup banner in backend logs

Last Updated: 2025-11-22
Version: 1.0.0