LLM-SRE Platform – Software Architecture, CI/CD & Threat Modeling

This document provides a deep technical explanation of the LLM-SRE platform software architecture, CI/CD automation, environment separation (DEV/PROD), infrastructure orchestration, and security threat modeling. It is intended as a long-term reference for design, troubleshooting, and future expansion.

1. Repository Structure

GitHub Repository: sanjaynishi/llm-sre-site

Top-level layout:
- ui/ → React/Vite frontend
- infra/ → Terraform infrastructure as code
  - modules/site (S3, CloudFront, OAC)
  - modules/agent_api (Lambda, API Gateway)
  - envs/dev
  - envs/prod
- .github/workflows → CI/CD pipelines

2. UI Architecture (SPA)

The UI is a Single Page Application (SPA) built using Vite + React.

Flow:
User → CloudFront → S3 (index.html)
JS/CSS assets are versioned (index-*.js) and served via CloudFront OAC.

Key Fixes:
• Correct CloudFront behavior for /assets/*
• Removed SPA rewrite rules from asset paths
• Ensured correct Content-Type metadata in S3

3. CI/CD Automation Architecture

CI/CD is implemented using GitHub Actions with AWS OIDC authentication.

Pipeline (DEV):
1. Checkout code
2. Build UI (npm ci → npm run build)
3. Terraform init + apply (remote backend)
4. Read Terraform outputs dynamically
5. Sync UI artifacts to S3
6. CloudFront invalidation

Security:
• No AWS static keys
• sts:AssumeRoleWithWebIdentity via GitHub OIDC

4. Infrastructure Architecture (DEV & PROD)

Components:
• Route 53 – DNS
• CloudFront – CDN + routing
• S3 – UI hosting (private with OAC)
• API Gateway – /api routing
• Lambda – Agent API
• DynamoDB – Terraform state locking
• S3 Backend – Terraform remote state

Environment Isolation:
• Separate state keys
• Separate IAM roles
• Separate GitHub environments

5. Threat Modeling (STRIDE-based)

Threat	Risk	Mitigation
Spoofing	Unauthorized CI/CD access	OIDC with repo-bound IAM role
Tampering	State corruption	DynamoDB state locking
Repudiation	Untracked deploys	GitHub audit logs + CloudTrail
Information Disclosure	Public S3 access	CloudFront OAC + block public ACLs
Denial of Service	Traffic floods	CloudFront caching + optional WAF
Elevation of Privilege	Over-permissive IAM	Least privilege IAM policies

6. Common Issues & Fixes

CloudFront 403 on assets
• Root cause: SPA rewrite rule applied to /assets
• Fix: Ordered cache behavior for /assets/*

GoDaddy DNS conflicts
• Root cause: Domain forwarding + A records mixed
• Fix: Migrated authoritative DNS to Route 53

Terraform state lock stuck
• Root cause: Interrupted GitHub Action
• Fix: DynamoDB lock inspection and cleanup

7. Validation & Debug Commands

curl -I https://dev.aimlsre.com
curl -I https://dev.aimlsre.com/assets/index-*.js
curl https://dev.aimlsre.com/api/agents

aws cloudfront list-distributions
aws s3api head-object --bucket <bucket> --key <asset>
aws dynamodb get-item --table-name llm-sre-terraform-locks

8. Final Notes

This architecture is production-grade, secure by default, and designed for extensibility. It can be reused as a reference blueprint for future SaaS or SRE automation platforms.