EngageIQ Developer Briefing Pack v1.2

# 1. Executive Summary

EngageIQ is a multi-tenant AI-powered sales enablement tool designed for Communications Service Providers (CSPs) and their reseller partners. This project involves building the production-ready AWS platform to host the application securely, scalably, and with strong tenant isolation.

# 2. Project Objectives

The objective is to deliver a low-maintenance, secure, and extensible AWS-hosted platform for the EngageIQ application. This includes core services such as user authentication, tenant isolation, branding flexibility, chat interface hosting, and API support for future AI integration.

# 3. Platform Architecture Overview

The platform will be based on App Runner, S3, CloudFront, DynamoDB, and Cognito.

The structure supports a 3 or 4 tier tenant hierarchy (Admin → Service Provider → Organisation → User) with secure data isolation per tenant.

# 4. Functional Requirements

* Secure multi-tenant user authentication and token management via Cognito
* A front-end UI hosted under engageiq.tech, with custom branding per organization
* Persistent conversation history per user via DynamoDB
* Manual RAG ingestion support (document upload per organisation)
* Exposed APIs for AI layer to consume and interact with user, tenant, and context metadata

# 5. Non-Functional Requirements

* Scalability and cost-efficiency via App Runner/S3
* Encryption in transit and at rest
* Clean and well-documented infrastructure handoff
* Readiness for ISO 27001 alignment (e.g., separation of concerns, auditability)

# 6. Phase Breakdown & Deliverables

Phase 1: Platform infrastructure setup  
Phase 2: Authentication flow and tenant model  
Phase 3: Front-end UI hosting and branding logic  
Phase 4: API gateway and AI integration scaffolding  
Phase 5: Documentation and handoff

# 7. Integration Requirements

The AI application (to be developed separately) will interact with the platform via secured API endpoints. Platform must pass user identity, tenant ID, and conversation context to the AI service and return results via the chat interface.

# 8. Roadmap Considerations (Not Part of Initial Scope)

- Self-service RAG ingestion  
- Organisation admin for user management  
- Embedded analytics and usage reporting  
- Per-user custom instructions and behaviour tuning

# 9. Development Approach & Expectations

The selected developer will work independently, delivering on a fixed-fee basis per milestone. Strong communication, documentation, and adherence to clean infrastructure-as-code principles are expected.