SOC 2 Compliant GraphQL Implementation Plan

Executive Summary

This document outlines the comprehensive implementation of a SOC 2 compliant GraphQL scaffold and codegen system for the Payroll-ByteMy application, focusing on secure handling of highly sensitive payroll data.

🔒 Security Classification Levels

Data Sensitivity Tiers

1. **CRITICAL** - Payroll financial data, SSN, bank details
2. **HIGH** - Employee PII, salary information, tax data
3. **MEDIUM** - Work schedules, leave records, general employee data
4. **LOW** - Public holidays, system configurations

Access Control Matrix

Role	Critical Data	High Data	Medium Data	Low Data
admin	Full Access	Full Access	Full Access	Full Access
org_admin	Read/Write with Audit	Full Access	Full Access	Full Access
manager	No Access	Team Only	Team Only	Full Access
consultant	No Access	Assigned Only	Assigned Only	Full Access
viewer	No Access	No Access	Read Only (Limited)	Full Access

🏗️ Architecture Overview

Directory Structure

```
graphql/
├── schema/
│   ├── base/                # Base schema definitions
│   ├── sensitive/           # Sensitive field definitions
│   └── public/              # Public field definitions
├── operations/
│   ├── critical/            # Critical operations (admin only)
│   ├── sensitive/           # Sensitive operations
│   └── standard/            # Standard operations
├── fragments/
│   ├── secure/              # Fragments with PII
│   └── public/              # Public fragments
├── generated/
│   ├── types/               # TypeScript types
│   ├── hooks/               # React hooks
```

```
│   ├── apollo/              # Apollo client configs
│   └── msw/                 # Mock service workers
└── audit/
    ├── logs/                # Audit log schemas
    └── compliance/          # Compliance checks
```

🔐 Security Controls

1. Field-Level Security

- Implement field-level encryption for CRITICAL data
- Use column-level permissions in Hasura
- Apply data masking for sensitive fields

2. Row-Level Security

- Enforce organization boundaries
- Implement team-based access controls
- Apply temporal access restrictions

3. Operation-Level Security

- Classify all operations by sensitivity
- Implement rate limiting by operation type
- Enforce audit logging for sensitive operations

4. Transport Security

- Enforce TLS 1.3 minimum
- Implement certificate pinning
- Use secure WebSocket connections

📊 Compliance Requirements

SOC 2 Trust Service Criteria

1. **Security (CC)**
   - Access controls
   - Encryption at rest and in transit
   - Vulnerability management
2. **Availability (A)**
   - Service monitoring
   - Incident response
   - Disaster recovery
3. **Processing Integrity (PI)**
   - Data validation

- Error handling
- Transaction integrity

4. **Confidentiality (C)**
    - Data classification
    - Access restrictions
    - Encryption standards
5. **Privacy (P)**
    - PII handling
    - Data retention
    - Right to erasure

🚀 Implementation Phases

Phase 1: Schema Analysis & Classification

1. Analyze current schema for sensitive data
2. Classify all fields by sensitivity
3. Document data flows and dependencies

Phase 2: Permission Model Enhancement

1. Implement granular permissions
2. Add field-level security
3. Create audit triggers

Phase 3: GraphQL Operations Refactor

1. Reorganize operations by security level
2. Implement secure fragments
3. Add operation-level validation

Phase 4: Codegen Configuration

1. Configure type generation with security annotations
2. Generate secure hooks with built-in validation
3. Create MSW handlers for testing

Phase 5: Audit & Monitoring

1. Implement comprehensive audit logging
2. Create compliance dashboards
3. Set up automated security scans

🔍 Monitoring & Alerting

Security Events to Monitor

- Failed authentication attempts
- Unauthorized data access
- Bulk data exports
- Schema modifications
- Permission changes

Compliance Metrics

- Access review completion
- Security training compliance
- Incident response times
- Vulnerability remediation SLAs

📝 Documentation Requirements

For Developers

- Security coding guidelines
- Data handling procedures
- Incident response playbook

For Auditors

- System architecture diagrams
- Data flow documentation
- Control implementation evidence

For Operations

- Monitoring procedures
- Backup and recovery plans
- Change management process