IOC Intelligent Operating Centre

Architecture & Security Framework Documentation

Enterprise Network Security & SCADA Monitoring Platform

Version 1.0

January 2026

CONFIDENTIAL

Table of Contents

1. Executive Summary

The **IOC Intelligent Operating Centre** (networkscanscada) is an enterprise-grade network security and SCADA monitoring platform designed to provide comprehensive security assessment, industrial control system monitoring, compliance checking, and data loss prevention capabilities.

> **Key Capabilities:**
>
> - Network vulnerability assessment with CVSS 3.1 scoring
> - SCADA/ICS monitoring supporting Modbus, OPC UA, and DNP3 protocols
> - Email Data Loss Prevention (DLP) with pattern-based detection
> - Multi-framework compliance checking (NIST, ISO 27001, PCI DSS, HIPAA)
> - Multi-tenant SaaS architecture with role-based access control
> - RESTful API for integration and automation

The platform is built using PHP with MySQL/MariaDB backend, featuring a modular architecture that spans 174+ PHP files organized across structured modules and classes. It supports industry-specific deployments including Oil & Gas, Rail, Mining, and Manufacturing sectors.

2. System Architecture Overview

2.1 High-Level Architecture

```
┌──────────────────────────────────────────────────────────────┐
│               IOC INTELLIGENT OPERATING CENTRE               │
├──────────────────────────────────────────────────────────────┤
│                                                              │
│   ┌───────────┐     ┌───────────┐     ┌───────────┐          │
│   │  Web UI   │     │ Admin CMS │     │ REST API  │          │
│   │(index.php)│     │ (/admin/) │     │ (api.php) │          │
│   └─────┬─────┘     └─────┬─────┘     └─────┬─────┘          │
│         │                 │                 │                │
│         └─────────────────┼─────────────────┘                │
│                           │                                  │
│                  ┌────────┴────────┐                         │
│                  │  Core Classes   │                         │
│                  │  (/classes/)    │                         │
│                  └────────┬────────┘                         │
│                           │                                  │
│      ┌────────────────────┼────────────────────┐             │
│      │                    │                    │             │
│      ▼                    ▼                    ▼             │
│ ┌──────────┐      ┌─────────────┐      ┌───────────┐         │
│ │ Network  │      │   SCADA     │      │  Email    │         │
│ │ Scanner  │      │  Monitor    │      │   DLP     │         │
│ └──────────┘      └─────────────┘      └───────────┘         │
│                           │                                  │
│                  ┌────────┴────────┐                         │
│                  │  Database.php   │                         │
│                  │ (PDO Singleton) │                         │
│                  └────────┬────────┘                         │
│                           │                                  │
│                  ┌────────┴────────┐                         │
│                  │ MySQL/MariaDB   │                         │
│                  │  Database       │                         │
│                  └─────────────────┘                         │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

2.2 Directory Structure

```
networkscanscada/
├── classes/                    # Core PHP Classes (29 classes)
│   ├── Network Scanning        # NetworkScanner, PortScanner, ServiceDetector
│   ├── Industrial Control      # SCADAMonitor, ModbusProtocol, OPCUAProtocol,
DNP3Protocol
│   ├── Security               # VulnerabilityScanner, ComplianceChecker, RateLimiter
│   ├── Monitoring             # AlertManager, RealtimeMonitor, DeviceDiscovery
│   ├── Storage                # StorageScanner, CalibrationManager, TankMonitor
│   ├── Industry/              # OilGasModule, RailModule, MiningModule,
ManufacturingModule
│   └── Reporting              # ReportGenerator
├── modules/                    # Feature Modules (30+ modules)
│   ├── ITSM                   # itsm.php, log_incident.php, log_problem.php
│   ├── Network Tools          # nta.php, ipam.php, snmp_monitor.php
│   ├── Database               # dpa.php, sql_sentry.php
│   └── Application            # sam.php, ai_analytics.php
├── admin/                      # Admin CMS Portal
│   ├── login.php              # Authentication
│   ├── settings.php           # Configuration
│   └── tenants.php            # Multi-tenant management
├── agent/                      # Remote Agent
├── database/                   # SQL Schema Files
├── config/                     # Configuration Files
├── templates/                  # Report Templates
└── docs/                       # Documentation
```

3. Core Components

3.1 Component Overview

Component	File(s)	Description
Database Layer	Database.php	Singleton PDO connection manager with prepared statements
Network Scanner	NetworkScanner.php, PortScanner.php	TCP/UDP port scanning with service detection
Vulnerability Engine	VulnerabilityScanner.php	CVSS-based vulnerability assessment
SCADA Monitor	SCADAMonitor.php	Industrial control system monitoring
Protocol Handlers	ModbusProtocol.php, OPCUAProtocol.php, DNP3Protocol.php	Industrial protocol implementations
Compliance Checker	ComplianceChecker.php	Multi-framework compliance assessment
Email DLP	EmailScanner.php, EmailLeakTracker.php	Data loss prevention for email content
Alert Manager	AlertManager.php	Multi-channel alert routing
Report Generator	ReportGenerator.php	Multi-format report generation
Rate Limiter	RateLimiter.php	API rate limiting protection

3.2 Industry-Specific Modules

The platform includes specialized modules for different industrial sectors:

Module	Features

Oil & Gas	Pipeline monitoring, leak detection, LACT units, custody transfer
Rail	Rail infrastructure monitoring, signaling systems
Mining	Mining equipment tracking, environmental monitoring
Manufacturing	Manufacturing process control, quality assurance

4. Database Architecture

4.1 Database Configuration

```
// config/database.php
$config = [
    'host'     => getenv('DB_HOST') ?: 'localhost',
    'port'     => getenv('DB_PORT') ?: '3307',
    'dbname'   => getenv('DB_NAME') ?: 'network_security_scanner',
    'user'     => getenv('DB_USER') ?: 'root',
    'password' => getenv('DB_PASS') ?: '',
    'charset'  => 'utf8mb4',
    'collation'=> 'utf8mb4_unicode_ci'
];
```

4.2 Main Database Schema

The `network_security_scanner` database contains the following core tables:

Table	Purpose	Key Fields
scans	Scan execution records	id, scan_name, target_range, scan_type, status, start_time, end_time
hosts	Discovered network devices	id, scan_id, ip_address, hostname, os_info, risk_score
ports	Open ports and services	id, host_id, port_number, protocol, service_name, version
vulnerabilities	CVE database	id, cve_id, severity, cvss_score, description, solution
scan_results	Vulnerability findings	id, scan_id, host_id, vulnerability_id, evidence
compliance_frameworks	Compliance standards	id, name, version, description
compliance_controls	Individual controls	id, framework_id, control_id, title, description
admin_users	CMS user management	id, username, password_hash, role, tenant_code

audit_log	Activity tracking	id, user_id, action, details, timestamp
api_rate_limits	Rate limiting data	client_identifier, request_count, window_start

4.3 Email DLP Database Schema

The `mailscan_dlp` database is dedicated to email data loss prevention:

Table	Purpose
detection_rules	DLP pattern matching rules (regex, keyword, pattern)
email_logs	Email metadata and content storage
scan_results	Rule match tracking and violations
email_forwarding_chains	Leak path tracking
audit_log	DLP action audit trail

4.4 Entity Relationship Diagram

```
┌─────────────┐     ┌─────────────┐     ┌─────────────────┐
│    scans    │──┐  │    hosts    │──┐  │ vulnerabilities │
├─────────────┤  │  ├─────────────┤  │  ├─────────────────┤
│ id          │  │  │ id          │  │  │ id              │
│ scan_name   │  └─▶│ scan_id     │  │  │ cve_id          │
│ target_range│     │ ip_address  │  │  │ severity        │
│ scan_type   │     │ hostname    │  │  │ cvss_score      │
│ status      │     │ risk_score  │  │  │ description     │
└─────────────┘     └─────────────┘  │  └─────────────────┘
                           │            │          │
                           ▼            │          │
                    ┌─────────────┐     │          │
                    │    ports    │     │          │
                    ├─────────────┤     │          │
                    │ id          │     │          │
                    │ host_id     │◀────┘          │
                    │ port_number │                │
                    │ service_name│                │
                    └─────────────┘                │
                           │                       │
                           ▼                       ▼
                  ┌────────────────────────────────┐
                  │          scan_results          │
                  ├────────────────────────────────┤
```

```
                                      │ id                               │
                                     │ scan_id, host_id, vulnerability_id│
                                      │ evidence, status                 │
                                      └──────────────────────────────────┘
```

5. Security Framework

5.1 Authentication & Authorization

Authentication Mechanisms

Mechanism	Implementation	Location
Password Hashing	bcrypt ($2y$10$...)	admin/login.php
Session Management	PHP native sessions	admin/login.php
Role-Based Access	admin, analyst, viewer	Database user roles
Multi-Tenancy	tenant_code isolation	All queries

Authorization Flow

```
 ┌───────────┐    ┌───────────────┐    ┌──────────────┐    ┌───────────────┐
│   User    │───▶│  login.php    │───▶│   Validate   │───▶│    Session    │
 │   Login   │    │  (POST)       │    │   bcrypt     │    │   Created     │
 └───────────┘    └───────────────┘    └──────────────┘    └───────────────┘
                                              │                     │
                                              ▼                     ▼
                                       ┌──────────────┐    ┌───────────────┐
                                       │   Failed     │    │  Role-Based   │
                                       │   Login      │    │   Access      │
                                       └──────────────┘    └───────────────┘
```

5.2 Input Security

Security Measures Implemented:

- **SQL Injection Prevention:** PDO prepared statements with parameterized queries throughout
- **XSS Prevention:** htmlspecialchars() for all output escaping
- **CSRF Protection:** Session-based validation for form submissions

- **Input Validation:** Server-side validation for all user inputs

Example Secure Query Pattern

```
// Parameterized query example from the codebase
$stmt = $db->prepare("
    SELECT * FROM scans
    WHERE id = ? AND tenant_code = ?
");
$stmt->execute([$scan_id, $tenant_code]);
```

5.3 Rate Limiting

The platform implements database-backed rate limiting to prevent abuse:

Parameter	Value
Default Limit	100 requests/minute
Time Window	1-minute sliding window
Client Identification	MD5(IP + User-Agent)
Exceeded Response	HTTP 429 Too Many Requests

5.4 Audit Logging

All security-relevant actions are logged to the audit_log table:

- User authentication (login/logout)
- Configuration changes
- Scan initiation and completion
- Report generation
- API access
- DLP violations detected

5.5 Data Protection

Protection Type	Implementation

Passwords	bcrypt hashing with salt
Sessions	Server-side storage, secure cookies
Database	PDO with prepared statements
Output	HTML entity encoding
Sensitive Data	DLP scanning and classification

6. Network Scanning Engine

6.1 Scanning Architecture

```
┌─────────────────────────────────────────────────────────────┐
│                   NETWORK SCANNING ENGINE                   │
├─────────────────────────────────────────────────────────────┤
│                                                             │
│    ┌───────────────────┐                                    │
│    │  NetworkScanner   │◄── Main Orchestrator               │
│    └────────┬──────────┘                                    │
│             │                                               │
│    ┌────────┼──────────────────────────────┐                │
│    │        │                              │                │
│    ▼        ▼                              ▼                │
│  ┌───────────────┐  ┌───────────────┐  ┌───────────────────┐  │
│  │ PortScanner   │  │ServiceDetector│  │VulnerabilityScanner│  │
│  │               │  │               │  │                   │  │
│  │ - TCP Scan    │  │ - Banner Grab │  │ - CVE Matching    │  │
│  │ - UDP Scan    │  │ - Version ID  │  │ - CVSS Scoring    │  │
│  │ - Host Alive  │  │ - Protocol ID │  │ - Risk Assessment │  │
│  └───────────────┘  └───────────────┘  └───────────────────┘  │
│                                                             │
└─────────────────────────────────────────────────────────────┘
```

6.2 Scanning Features

Port Scanning

Feature	Details
Target Formats	Single IP, CIDR notation, IP lists
Common Ports	21, 22, 23, 25, 53, 80, 110, 135, 139, 143, 443, 445, 993, 995, 3306, 3389, 5900, 8080
Protocols	TCP and UDP
Timeout	0.5 seconds per port
Service Detection	Banner grabbing and protocol identification

Scan Types

- **Full Scan:** Comprehensive assessment of all ports and services
- **Quick Scan:** Common ports only for rapid assessment
- **Custom Scan:** User-defined port ranges and parameters
- **Compliance Scan:** Framework-specific security checks
- **Vulnerability Scan:** Focused on known weaknesses

6.3 Vulnerability Assessment

Security Checks Performed

Check Category	Severity	Description
Weak Protocols	HIGH	FTP, Telnet, HTTP (unencrypted)
Outdated Versions	HIGH	Services with known CVEs
Default Credentials	CRITICAL	Services using default passwords
SSL/TLS Issues	MEDIUM	Weak ciphers, expired certificates
Missing Headers	LOW	Security headers not configured
Open Databases	CRITICAL	Unauthenticated database access

CVSS Scoring

Vulnerabilities are scored using CVSS 3.1:

- CRITICAL 9.0 - 10.0
- HIGH 7.0 - 8.9
- MEDIUM 4.0 - 6.9
- LOW 0.1 - 3.9
- INFO 0.0

7. SCADA/ICS Monitoring

7.1 Protocol Support

Protocol	Class	Port	Usage
Modbus TCP	ModbusProtocol.php	502	PLC communication, register read/write
Modbus RTU	ModbusProtocol.php	Serial	Serial device communication
OPC UA	OPCUAProtocol.php	4840	Industrial automation, data exchange
DNP3	DNP3Protocol.php	20000	Utility SCADA, power systems

7.2 SCADA Architecture

```
┌─────────────────────────────────────────────────────────────────┐
│                      SCADA MONITORING LAYER                      │
├─────────────────────────────────────────────────────────────────┤
│                                                                  │
│   ┌─────────────────────────────────────────────────────────┐   │
│   │                    SCADAMonitor.php                      │   │
│   │  - Device Discovery    - Tag Management                  │   │
│   │  - Data Acquisition    - Alarm Processing                │   │
│   └─────────────────────────────────────────────────────────┘   │
│                              │                                   │
│        ┌─────────────────────┼─────────────────────┐            │
│        │                     │                     │            │
│        ▼                     ▼                     ▼            │
│   ┌──────────┐         ┌──────────┐         ┌──────────┐        │
│   │  Modbus  │         │  OPC UA  │         │   DNP3   │        │
│   │ Protocol │         │ Protocol │         │ Protocol │        │
│   └────┬─────┘         └────┬─────┘         └────┬─────┘        │
│        │                    │                    │               │
│        ▼                    ▼                    ▼               │
│   ┌──────────┐         ┌──────────┐         ┌──────────┐        │
│   │   PLCs   │         │   HMIs   │         │   RTUs   │        │
│   │   RTUs   │         │ Servers  │         │   IEDs   │        │
│   └──────────┘         └──────────┘         └──────────┘        │
│                                                                  │
└─────────────────────────────────────────────────────────────────┘
```

7.3 Monitoring Capabilities

- **Real-time Data Acquisition:** Continuous polling of PLC/RTU values
- **Alarm Threshold Management:** Configurable high/low limits
- **Device Discovery:** Automatic detection of SCADA devices
- **Tag Value Tracking:** Historical data logging
- **Alarm Event Recording:** Complete alarm history

7.4 Industry-Specific Features

Oil & Gas (OilGasModule.php)

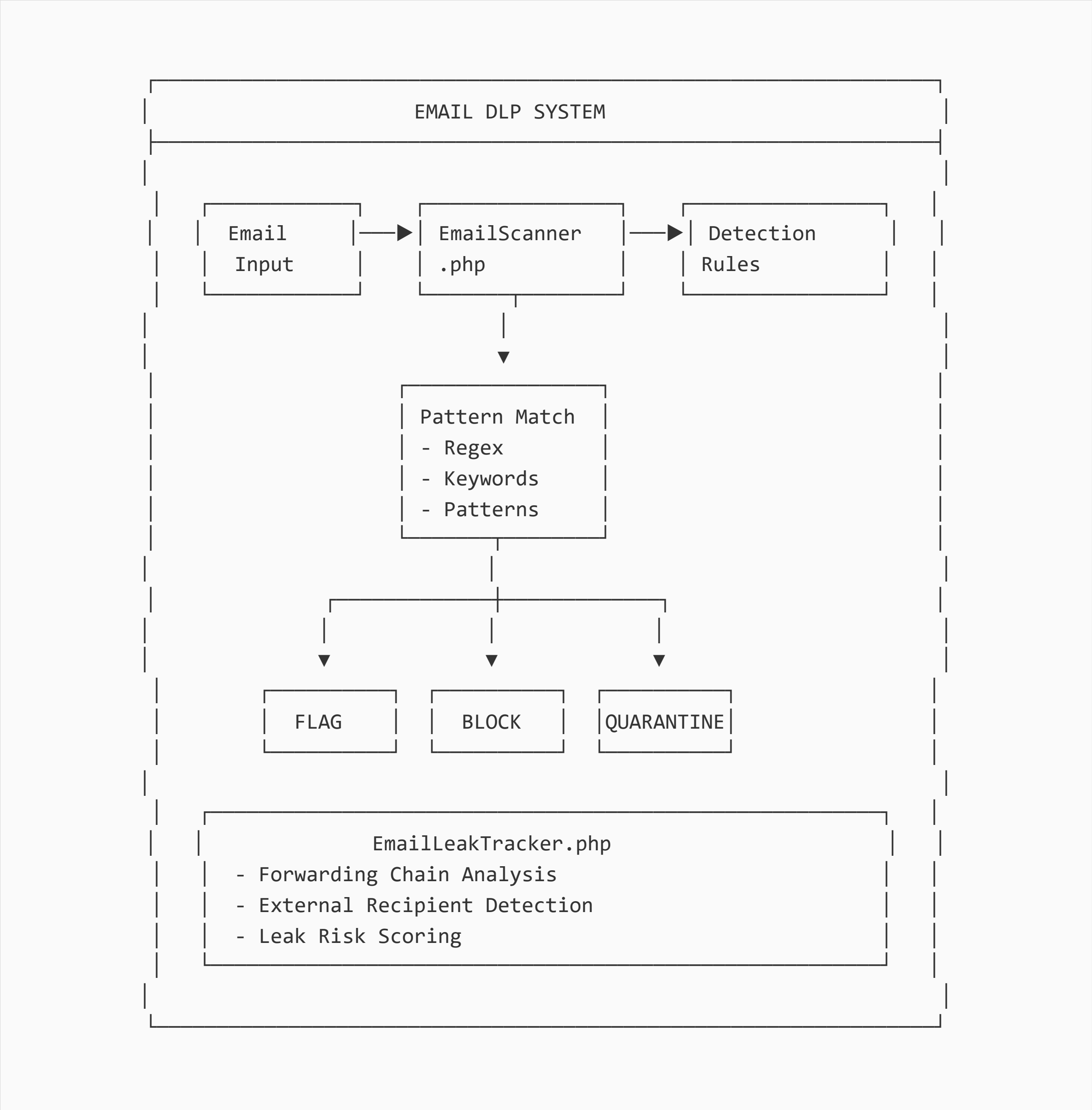
- Pipeline leak detection (5% flow imbalance threshold)
- Pressure variance monitoring
- LACT unit monitoring
- Custody transfer tracking

Additional Modules

- **TankMonitor.php:** Tank level and inventory tracking
- **CalibrationManager.php:** Instrument calibration scheduling
- **ValveController.php:** Valve position control
- **ShutdownManager.php:** Emergency shutdown (ESD) management

8. Email DLP System

8.1 DLP Architecture



8.2 Detection Rules

Rule Type	Pattern	Severity
Credit Cards	Visa, MasterCard, Amex, Discover patterns	CRITICAL
SSN	XXX-XX-XXXX format	CRITICAL

API Keys	sk_live_, AKIA prefix patterns	CRITICAL
AWS Keys	AKIA[0-9A-Z]{16}	CRITICAL
Passwords	password: or pwd= patterns	CRITICAL
Bank Accounts	Account number patterns	HIGH
Medical IDs	MRN-, Patient ID patterns	HIGH
Email Addresses	Standard email regex	LOW
Phone Numbers	US phone format	LOW

8.3 Leak Tracking

The EmailLeakTracker component monitors email forwarding patterns to detect potential data leaks:

- **Forwarding Chain Analysis:** Tracks email forward paths
- **Hop Number Tracking:** Counts forwarding depth
- **External Recipient Detection:** Identifies outside addresses
- **Risk Score Calculation:** Threshold of 70 for alerts
- **Forward Type Classification:** direct_forward, cc, bcc

8.4 Content Scanning Fields

- Subject line
- Body text (plain text)
- HTML body (tags stripped)
- Sender email address
- Recipient email addresses
- Attachment metadata

9. API Architecture

9.1 REST API Endpoints

Method	Endpoint	Description
GET	/api.php?action=stats	Dashboard statistics
GET	/api.php?action=scans	List recent scans (max 50)
GET	/api.php?action=scan&id=X	Get specific scan details
POST	/api.php?action=start_scan	Initiate new scan
GET	/api.php?action=vulnerabilities	List vulnerabilities
GET	/api.php?action=hosts&scan_id=X	List hosts from scan
POST	/api.php?action=report	Generate report
GET	/api.php?action=compliance&scan_id=X	Compliance results
GET	/api.php?action=export&scan_id=X	Export data (JSON/CSV)

9.2 API Response Format

```
{
    "success": true,
    "data": {
        "total_scans": 150,
        "total_vulnerabilities": 423,
        "critical_count": 12,
        "high_count": 45,
        "medium_count": 156,
        "low_count": 210
    },
    "timestamp": "2026-01-30T10:30:00Z"
}
```

9.3 API Security

Feature	Implementation
Rate Limiting	100 requests/minute per client
CORS	Configurable origin headers
Error Handling	JSON error responses, no stack traces
Timeout	10-minute execution limit for scans

10. Compliance Framework

10.1 Supported Standards

Framework	Version	Controls
NIST CSF	1.1	5 Functions, 23 Categories
ISO 27001	2013	114 Controls
CIS Controls	v8	18 Controls
PCI DSS	v4.0	12 Requirements
HIPAA	2013	45 Controls
SOC 2	Type II	64 Controls

10.2 Control Categories

- Asset Inventory and Management
- Access Control and Authentication
- Data Encryption (at rest and in transit)
- Protective Technology Implementation
- Secure Configuration Management
- Audit Logging and Monitoring
- Vulnerability Management
- Firewall and Network Security
- Default Credential Elimination
- Incident Response Procedures

10.3 Compliance Reporting

The ComplianceChecker generates detailed reports including:

- Overall compliance score per framework
- Individual control pass/fail status
- Evidence collection for audit

- Remediation recommendations
- Gap analysis and prioritization

11. Deployment Architecture

11.1 Deployment Options

Deployment Type	Description	Use Case
POC (Proof of Concept)	Single-server local installation	Evaluation, testing, small environments
Hybrid	Cloud management + on-premise collectors	Enterprise with distributed sites
SaaS (Cloud-Only)	Full cloud deployment with Terraform	Cloud-native organizations

11.2 POC Architecture

```
┌──────────────────────────────────────────┐
│              LOCAL SERVER                │
│                                          │
│   ┌────────────┐     ┌────────────┐      │
│   │  Apache    │     │  MySQL     │      │
│   │  + PHP     │────▶│  Database  │      │
│   └────────────┘     └────────────┘      │
│         │                                │
│         ▼                                │
│   ┌──────────────────────────────┐       │
│   │    IOC Application           │       │
│   │  - Web UI                    │       │
│   │  - API                       │       │
│   │  - Scanners                  │       │
│   └──────────────────────────────┘       │
│                                          │
└──────────────────────────────────────────┘
```

11.3 Hybrid Architecture

```
┌──────────────────────────────────────────────────────────────┐
│                          CLOUD                               │
│   ┌──────────────────────────────────────────────────────┐   │
│   │          Management Console                        │   │
│   │  - Dashboard      - Reporting     - Configuration  │   │
│   └────────────────────────────────────────────────────┘   │
└───────────────────────────┬────────────────────────────────┘
                            │
                   HTTPS (Encrypted)
                            │
┌───────────────────────────┼────────────────────────────────┐
│                     ON-PREMISE                             │
│                         │                                  │
│   ┌─────────────────────▼──────────────────────────────┐   │
│   │                Collector Agent                     │   │
│   │  - Network Scanning                                │   │
│   │  - SCADA Monitoring                                │   │
│   │  - Local Database                                  │   │
│   └────────────────────────────────────────────────────┘   │
│                         │                                  │
│          ┌──────────────┼──────────────┐                   │
│          │              │              │                   │
│          ▼              ▼              ▼                   │
│     ┌────────┐     ┌────────┐     ┌────────┐               │
│     │ PLCs   │     │ RTUs   │     │Servers │               │
│     └────────┘     └────────┘     └────────┘               │
│                                                            │
└────────────────────────────────────────────────────────────┘
```

11.4 Multi-Tenant Architecture

The platform supports multi-tenant deployments with:

- **Tenant Isolation:** Data segregation via tenant_code
- **User Limits:** Configurable per-tenant user quotas
- **Storage Quotas:** Per-tenant storage limits
- **Tenant Status:** active, inactive, suspended states
- **Role-Based Access:** Per-tenant admin, analyst, viewer roles

12. Technology Stack

12.1 Backend Technologies

Component	Technology	Version
Language	PHP	7.4+
Database	MySQL / MariaDB	5.7+
Database Layer	PDO	Native
Web Server	Apache / Nginx	2.4+ / 1.18+
Architecture	Custom MVC-like	-

12.2 Frontend Technologies

Component	Technology
Markup	HTML5
Styling	CSS3 (inline and embedded)
Scripting	JavaScript (ES6+)
Charts	Native canvas rendering

12.3 Required PHP Extensions

- PDO and PDO_MySQL
- Sockets
- cURL
- JSON
- OpenSSL
- mbstring

12.4 System Requirements

Resource	Minimum	Recommended
CPU	2 cores	4+ cores
RAM	4 GB	8+ GB
Storage	20 GB	100+ GB
Network	100 Mbps	1 Gbps

IOC Intelligent Operating Centre - Architecture & Security Framework Documentation

Version 1.0 | January 2026 | Confidential