SWASTHYA: Quality & Security Framework

1. Executive Overview

Security and Quality are not optional features of **Project Swasthya** — they form its *core foundation*.
Given the highly sensitive nature of health data and the importance of real-time accuracy, the system follows two guiding principles:

- **Privacy-by-Design** – Security and privacy are embedded at every layer.
- **Quality-by-Default** – Every dataset and process is designed to ensure reliability and accuracy.

This document defines the **end-to-end framework** for ensuring security, privacy, data quality, and operational reliability across the system architecture.

2. Pillar 1: Data Security (The "Fortress")

This pillar focuses on protecting project data against unauthorized access during transfer (**in transit**) and while stored (**at rest**).

2.1. Data in Transit Security

All data moving into, out of, or within the cloud is **encrypted end-to-end**.

- **IoT Sensor-to-Cloud:**
  Each **ESP32 sensor** is provisioned with a unique **X.509 certificate**. Communication occurs via **MQTT over TLS 1.2/1.3**, ensuring encryption and authentication, preventing interception or spoofing.
- **Application-to-Cloud:**
  All communication from the **Citizen PWA**, **Clinic Portal**, and **Dashboard** occurs over **HTTPS (SSL/TLS)**, enforced by **Amazon API Gateway** and **Amazon CloudFront**.
  This ensures encryption of all credentials, submissions, and user requests.
- **Internal Cloud Traffic:**
  Data movement between AWS services (Lambda → Timestream, SageMaker → S3, etc.) occurs over the **private AWS network**, isolated from the public internet.

2.2. Data at Rest Security

All data is **encrypted by default** when stored.

- **Time-Series Data:**
  Amazon Timestream (C1) uses AWS-managed keys for automatic encryption at rest.
- **Relational & Geospatial Data:**
  Amazon RDS (PostgreSQL) (C3) uses **AWS KMS** for encrypting storage, snapshots, and backups.
- **Data Lake & Logs:**
  Amazon S3 (C2) stores raw sensor data, ML datasets, and logs with **Server-Side Encryption (SSE-S3 or SSE-KMS)**.
  Bucket policies **enforce encryption** for every upload.

3. Pillar 2: Privacy by Design (The "Core Principle")

Privacy is not a feature — it is a **mandatory design rule**.
All system layers are structured to guarantee user anonymity and ethical data use.

3.1. The Federated Learning (FL) Architecture

The main privacy-preserving mechanism for citizen health data.

- **The Challenge:**
  Collecting symptom data without creating a centralized health database.
- **The Solution:**
  Implement a **Federated Learning (FL)** system using the **Flower framework** and **TensorFlow.js**.
- **The Process:**
  1. The Citizen PWA downloads the global model from the **Flower Server (D1)**.
  2. The model trains locally using the user's private symptom data (stored only on-device).
  3. The app uploads only **mathematical model updates** (gradients), not personal data.
  4. The server aggregates all updates into an improved global model.
- **Privacy Guarantee:**
  No raw or identifiable health data ever leaves the device.
  The central system only receives **anonymous, aggregated weight updates** — ensuring zero risk of re-identification.

3.2. Handling of Aggregated Clinic Data

Clinic data is anonymized **before** submission.

- **Anonymized at Source:**
  Clinics submit data as statistical summaries (e.g., “50 cold cases in Kothrud, Oct 1–10”). No Personal Identifiable Information (PII) is included.
- **Secure Handling:**
  Even though anonymized, clinic data follows the same **encryption**, **storage**, and **access** protocols as all other datasets.

4. Pillar 3: Application & Access Security (The "Gatekeeper")

Controls who can access the system and what operations they can perform.

4.1. Authentication & User Management

- **Centralized Authentication:**
  Managed through **Amazon Cognito (E4)**, serving as the single, secure identity provider.
- **Lifecycle Security:**
  Handles registration, email/phone verification, password policies, and login using **Secure Remote Password (SRP)** protocol.
- **Token-Based Authentication:**
  After login, users receive a **JSON Web Token (JWT)**, required for every API call.

4.2. Role-Based Access Control (RBAC)

Access permissions are enforced via **Cognito Groups** and validated by **Amazon API Gateway (E3)**.

Role	Access Point	Permissions
Citizen	PWA (`E1`)	Submit anonymized health data via FL. Cannot access dashboards.
ClinicStaff	Dashboard (`E2`)	Submit aggregated clinic data only. No access to insights or maps.
HealthOfficial / Researcher	Dashboard (`E2`)	Access full insights, maps, and AI analytics. Cannot submit data.

4.3. API Security

- All endpoints are secured through **API Gateway (E3)**.
- Requests without valid, unexpired JWTs or incorrect roles are **automatically rejected**.
- Each API action enforces **fine-grained access policies** to ensure least-privilege operation.

5. Pillar 4: Data Quality & Integrity (The "Foundation")

The accuracy of insights depends entirely on the quality of incoming data.
This pillar enforces validation, consistency, and trustworthiness.

5.1. Real-Time Validation Pipeline

The **AWS Lambda (B5)** layer ensures rigorous quality control for all incoming data.

- **Schema Validation:**
  Validates every incoming payload against a strict JSON schema. Invalid or incomplete entries are rejected.
- **Range & Plausibility Checks:**
  Filters unrealistic data (e.g., PM2.5 = -50 or 9999).
- **Dead-Letter Queues (DLQ):**
  Invalid entries are stored in a dedicated **S3 DLQ bucket** for manual inspection and debugging, ensuring **no silent data loss**.

5.2. Data Triangulation (The "3-Source Method")

The model's confidence is enhanced by verifying findings against **three independent data sources:**

1. **Citizen FL Data:**
   Real-time, high-frequency data (unverified, but broad-scale).
2. **Clinic Aggregated Data:**
   Low-frequency, high-reliability ground truth from verified entities.
3. **Contextual Data:**
   Weather, traffic, or pharmacy sales data — used for cross-validation.

Result:
Correlations are trusted only when supported by **two or more sources**, significantly improving data accuracy and model robustness.

6. Pillar 5: System Quality & Reliability (The "Bedrock")

Ensures the platform is fault-tolerant, scalable, and continuously operational.

6.1. Resilience & Fault Tolerance

- **Resilient Core:**
  Amazon Kinesis Data Streams (B2) buffers data between ingestion and processing.
- **Failure Scenario:**
  If **Lambda (B5)** or **Timestream (C1)** fails temporarily, Kinesis stores the data safely for up to **7 days**.
- **Automatic Recovery:**
  Once services are restored, data resumes flow automatically — ensuring **zero data loss**.

6.2. Scalability

Project Swasthya is built with a **serverless-first architecture** — automatically scaling with workload.

- **Ingestion Layer:**
  `AWS IoT Core` and `Kinesis` handle thousands to millions of sensor connections concurrently.
- **Processing Layer:**
  `AWS Lambda` scales horizontally, spawning parallel functions for concurrent sensor events.
- **Storage & Frontend:**
  `Amazon S3`, `Timestream`, and `CloudFront` provide fully managed, elastic scaling for both data and web traffic.

7. Conclusion

The **Quality & Security Framework** of Project Swasthya establishes an **industrial-grade foundation** of trust, reliability, and ethical integrity.

By combining **federated learning for privacy**, **multi-source validation for quality**, and **serverless AWS architecture for scalability**,
the project ensures that real-time public health intelligence is **secure, accurate, and sustainable** — ready to serve citizens, researchers, and policymakers responsibly.