**Relintel Software Design Document (SWDD)**

**Background**

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SWDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SWDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage—i.e., the detailed design stage—more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the *IEEE Recommended Practice for Software Design Descriptions*. The *IEEE Recommended Practice for Software Design Descriptions* have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to IEEE Std 1016[[1]](#footnote-1) for the full *IEEE Recommended Practice for Software Design Descriptions*.

SWDD- [obtain number from Conduct of Engineering Document Numbering [SharePoint site](https://coe.lanl.gov/APs/DocNum/SitePages/Home.aspx)]

ANGEL TWINAMATSIKO

**RELINTEL**

#### Software Design Document

Name (s): Lab Section: Workstation:

13TH Dec, 2023

**TABLE OF CONTENTS**

1.0 INTRODUCTION 4

1.1 Purpose 4

1.2 Scope 4

1.3 Overview 4

1.4 Reference Material 4

1.5 Definitions and Acronyms 4

2.0 SYSTEM OVERVIEW 4

3.0 SYSTEM ARCHITECTURE 4

3.1 Architectural Design 4

3.2 Decomposition Description 5

3.3 Design Rationale 5

4.0 DATA DESIGN 5

4.1 Data Description 5

4.2 Data Dictionary 5

5.0 COMPONENT DESIGN 5

6.0 HUMAN INTERFACE DESIGN 5

6.1 Overview of User Interface 5

6.2 Screen Images 6

6.3 Screen Objects and Actions 6

7.0 REQUIREMENTS MATRIX 6

8.0 APPENDICES 6

## INTRODUCTION

## Purpose

This Software Design Document (SWDD) serves as a comprehensive blueprint for the development and implementation of Relintel, a groundbreaking library designed to empower security and verification systems with seamless integration of real-time data and existing records. It outlines the library's architecture, features, functionalities, and design decisions, guiding developers in understanding, integrating, and extending its capabilities.

**Intended Audience:**

Developers: Individuals responsible for integrating Relintel's identity verification capabilities into security systems, access control platforms, financial applications, and other software requiring robust identity management.

System Architects: Professionals involved in designing and planning the integration of Relintel within larger software ecosystems, ensuring compatibility and alignment with existing infrastructure.

Project Managers: Individuals overseeing the development and deployment of projects utilizing Relintel, requiring a clear understanding of its scope and technical requirements.

Technical Decision-Makers: Individuals evaluating Relintel's suitability for specific use cases and making informed choices about its adoption.

**Key Objectives of this SWDD:**

Elucidate Relintel's Architecture and Components: Provide a detailed overview of the library's internal structure, interactions between components, and data flow to facilitate understanding and implementation.

Define APIs and Functionalities: Clearly outline the library's exposed functions, methods, and their expected inputs and outputs, enabling developers to effectively utilize its capabilities.

Guide Integration: Offer clear instructions and best practices for integrating Relintel with different systems and platforms, ensuring seamless interaction and data exchange.

Facilitate Development and Maintenance: Serve as a reference for developers and maintainers throughout the development lifecycle, promoting code clarity, consistency, and future enhancements.

Scope

Provide a description and scope of the software and explain the goals, objectives and benefits of your project. This will provide the basis for the brief description of your product.

## Overview

Relintel is a powerful library designed to revolutionize identity verification by seamlessly integrating real-time data and existing records. It empowers developers to build secure and accurate verification systems across diverse applications, from access control and financial transactions to online platforms and border security.

**Imagine a library that**:

**Harnesses real-time data:** Live camera feeds, sensor readings, and dynamic databases are continuously analyzed to provide the most up-to-date insights into individual identity.

**Mines hidden gems from existing records**: Legacy databases, biometric scans, and historical transactions are combed through, extracting key features and revealing potential discrepancies.

**Maches in milliseconds**: A sophisticated real-time engine swiftly compares extracted features, making accurate identity verification decisions in an instant.

**Adapts to evolving threats:** Machine learning algorithms learn from new data and adjust matching parameters, staying ahead of the curve against emerging impersonation tactics.

**Offers flexible integration**: Developers can easily integrate Relintel into their existing systems through intuitive APIs, unlocking powerful identity verification capabilities.

Here's a brief glimpse into Relintel's core functionalities:

**Data Acquisition:** Connect to diverse data sources like CCTV cameras, biometric scanners, databases, and cloud storage.

**Feature Extraction**: Extract key features from various data types, including facial recognition, gait analysis, textual data, and biometric measurements.

**Real-time Matching:** Perform efficient comparison of extracted features, incorporating multi-factor authentication and adaptive threat detection.

**Actionable Insights**: Generate real-time alerts and actionable recommendations based on verification results.

**Seamless Integration**: Integrate with existing access control, security, and financial platforms for enhanced system security.

Reference Material

*This section is optional.*

List any documents, if any, which were used as sources of information for the test plan.

## Definitions and Acronyms

This section aims to clarify any potentially ambiguous terms and acronyms used throughout the SWDD. This ensures consistency and avoids confusion during the development and integration of Relintel.

***Definitions:***

***Biometric Verification:*** The process of verifying an individual's identity by comparing their unique physical or behavioral characteristics (e.g., fingerprints, facial features, iris patterns) to stored data.

***Data Acquisition:*** The process of retrieving and collecting data from various sources, including real-time feeds, databases, and sensors.

***Feature Extraction:*** The process of identifying and extracting relevant characteristics from collected data, such as facial landmarks from video frames or text keywords from a document.

***Ground Truth:*** The verified and true identity of an individual used as a reference point for comparison in verification tasks.

***Matching Engine:*** The core component of Relintel responsible for comparing extracted features and making identity verification decisions in real-time.

***Multi-Factor Authentication (MFA):*** A security measure requiring multiple verification factors (e.g., password, one-time code, biometric scan) to confirm a user's identity.

***Real-Time Data***: Data that is generated and processed continuously, providing immediate insights into current events and actions.

***Security System:*** A combination of hardware, software, and procedures designed to protect assets and information from unauthorized access, use, disclosure, disruption, modification, or destruction.

***Verification:*** The process of confirming the accuracy of a claimed identity through comparison with existing information and trusted data sources.

***Acronyms***:

***API***: Application Programming Interface

***CCTV***: Closed-Circuit Television

***CNN***: Convolutional Neural Network

***DB***: Database

***DLP***: Data Loss Prevention

**GUI**: Graphical User Interface

***JWT***: JSON Web Token

***KYC***: Know Your Customer

***ML***: Machine Learning

***NLP***: Natural Language Processing

***REST***: Representational State Transfer

***SDK***: Software Development Kit

***SQL***: Structured Query Language

***UI:*** User Interface Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SWDD.

These definitions should be items used in the SWDD that are most likely not known to the intended users and developers.

## SYSTEM OVERVIEW

System Overview: Relintel - Real-time Identity Verification Revolution

Relintel is a cutting-edge library designed to revolutionize the landscape of identity verification by seamlessly integrating real-time data and existing records. It empowers developers to build secure and accurate verification systems across diverse applications, ranging from access control and financial transactions to online platforms and border security.

**Functionality:**

**Data Acquisition**: Connects to diverse data sources like CCTV cameras, biometric scanners, databases, and cloud storage, providing a holistic view of individual identity.

**Feature Extraction:** Extracts key features from various data types, including facial recognition, gait analysis, textual data, and biometric measurements, revealing hidden insights.

**Real-time Matching Engine:** Performs efficient comparison of extracted features, employing multi-factor authentication and adaptive threat detection for unparalleled accuracy.

**Actionable Insights:** Generates real-time alerts and recommendations based on verification results, enabling proactive responses to potential threats.

**Seamless Integration:** Integrates with existing security, access control, and financial platforms, enhancing system effectiveness and reducing development overhead.

**Context:**

The ever-evolving digital landscape demands robust and adaptable identity verification solutions. Legacy systems often struggle with outdated data and static configurations, making them vulnerable to fraud and impersonation. Relintel addresses this challenge by providing a dynamic and data-driven approach, constantly learning and adapting to evolving threats and emerging technologies.

**Design:**

Relintel embraces a modular architecture, facilitating flexible integration and future extensions.

Core components include:

**Data Acquisition Modules:** Specialized interfaces for interacting with diverse data sources in real-time and batch modes.

**Feature Extraction Engines**: Employ advanced algorithms and machine learning techniques to extract relevant features from various data types.

**Matching Engine: Performs** efficient comparison of extracted features, utilizing multi-factor authentication and adaptive algorithms for accurate and secure verification.

**API Interface:** Provides a clean and intuitive API for developers to easily integrate Relintel into their applications.

**Background**:

The increasing reliance on digital interactions necessitates reliable and secure identity verification. Relintel builds upon advancements in real-time data analysis, machine learning, and biometrics to offer a comprehensive and adaptable solution for this critical need.

This high-level overview provides a glimpse into the power and potential of Relintel. In the following chapters, we'll delve deeper into its architecture, functionalities, and integration mechanisms, empowering you to leverage its capabilities to build next-generation identity verification solutions..

## SYSTEM ARCHITECTURE

## Architectural Design

3.1.1 Modular Structure and Relationships

Relintel is designed with a modular architecture to promote flexibility, scalability, and maintainability.

Here's a breakdown of its core subsystems and their interactions:

1. Data Acquisition Subsystem:

Connects to diverse data sources using specialized modules:

CCTV cameras

Biometric scanners

Databases

Cloud storage

Handles real-time and batch data retrieval

Manages data preprocessing and buffering

2. Feature Extraction Subsystem:

Extracts relevant features from acquired data using diverse engines:

Facial recognition

Gait analysis

Textual analysis

Biometric feature extraction

Employs machine learning techniques for accuracy and adaptability

3. Matching Engine Subsystem:

Core decision-making component

Efficiently compares extracted features

Incorporates multi-factor authentication and adaptive algorithms

Optimizes decision-making based on real-time feedback loops

4. API Interface Subsystem:

Exposes functionalities to developers through a well-defined API

Facilitates integration with external systems

Provides clear documentation and usage examples

5. Data Storage Subsystem:

Manages storage and retrieval of relevant data, including:

Extracted features

Matching results

User profiles

Historical logs

Decomposition Description

3.2.1 Functional Decomposition:

Top-Level Data Flow Diagram (DFD)

Structural Decomposition Diagrams

3.2.2 Object-Oriented (OO) Decomposition:

Subsystem Model

Object Diagram

Generalization Hierarchy Diagrams

Aggregation Hierarchy Diagrams

Interface Specifications

Sequence Diagrams

## Design Rationale

3.3.1 Key Considerations for Architectural Selection:

Modularity: Promotes code reusability, maintainability, and extensibility.

Real-Time Performance: Enables swift data processing and decision-making.

Scalability: Supports growth in data volume and user base.

Security: Protects sensitive data and safeguards system integrity.

Flexibility: Accommodates diverse data sources and integration scenarios.

Adaptability: Adjusts to evolving threats and emerging technologies.

3.3.2 Alternative Architectures Considered:

Monolithic Architecture: Rejected due to limitations in flexibility, scalability, and maintainability.

Microservices Architecture: Considered for its potential benefits, but ultimately not chosen due to the added complexity and potential performance overhead for a library of this scope.

3.3.3 Trade-Offs:

Performance vs. Flexibility: The chosen architecture balances real-time performance with the flexibility to accommodate diverse data sources and integration scenarios.

Complexity vs. Maintainability: Modularity enhances maintainability, but careful design is required to manage component interactions and dependencies effectively.

3.3.4 Future Considerations:

Explore potential for a hybrid architecture incorporating microservices for specific functionalities if future requirements demand increased scalability and modularity.

Investigate cloud-based deployment options for enhanced scalability and resource management.

Continuously evaluate emerging technologies and architectural patterns to ensure Relintel's adaptability and maintainability in the long term.

## DATA DESIGN

## Data Description

4.1.1 Information Domain and Data Structures

Relintel handles a diverse array of data types, representing individuals, their features, verification events, and system configurations.

Here's how this information is transformed into data structures:

- Core Objects:

- Individual: Represents a person being verified, storing biographical information, security clearances, and biometric profiles.

- FeatureSet: Encapsulates extracted features from various data sources, including facial landmarks, gait patterns, textual keywords, and biometric measurements.

- VerificationResult: Stores the outcome of a verification attempt, including a confidence score, matching metadata, and potential anomalies.

- Data Containers:

- FeatureList: An ordered collection of FeatureSet objects, used for representing multiple feature sets associated with an individual or a verification event.

- VerificationHistory: A chronologically ordered collection of VerificationResult objects, maintaining a record of past verification attempts.

4.1.2 Data Storage and Organization

Relintel employs a combination of in-memory data structures and persistent storage to manage data effectively:

- In-Memory Data:

- Real-time data streams are buffered in memory for efficient processing.

- Extracted features and intermediate results are stored temporarily for rapid retrieval and decision-making.

- Persistent Storage:

- Database: Stores persistent data, including:

- User profiles

- Feature sets

- Verification results

- Historical logs

- System configuration settings

- File Storage: Used for storing large data objects, such as:

- Image files

- Video recordings

- Biometric templates.

## Data Dictionary

|  |  |  |
| --- | --- | --- |
| Entity Name | Type | Description |
| Individual | Object | Represents a person being verified, including biographical information, security clearances, and biometric profiles. |
| FeatureSet | Object | Encapsulates extracted features from various data sources, including facial landmarks, gait patterns, textual keywords, and biometric measurements. |
| VerificationResult | Object | Stores the outcome of a verification attempt, including a confidence score, matching metadata, and potential anomalies. |
| FeatureList | Collection | An ordered collection of FeatureSet objects. |
| VerificationHistory | Collection | A chronologically ordered collection of VerificationResult objects. |
| DataAcquisitionModule | Object | Interface for connecting to a specific data source (e.g., CCTV camera, database). |
| FeatureExtractionEngine | Object | Algorithm for extracting features from a specific data type (e.g., facial recognition, gait analysis). |
| MatchingEngine | Object | Core decision-making component for comparing extracted features and making verification decisions. |

4.2.2 Functions (Functional Description)

|  |  |  |
| --- | --- | --- |
| Function Name | Parameters | Description |
| acquireData(sourceId) | DataAcquisitionModule sourceModule | Retrieves data from the specified data source. |
| extractFeatures(data, engineType) | FeatureExtractionEngine engine | Extracts relevant features from the given data using the specified engine. |
| matchFeatures(featureSet1, featureSet2) | FeatureSet featureSet1, FeatureSet featureSet2 | Compares two feature sets and returns a verification result. |
| storeVerificationResult(result) | VerificationResult result | Stores the verification result in the database. |
| retrieveUserHistory(userId) | string userId | Retrieves the verification history for the specified user. |

4.2.3 Objects and Attributes (OO Description)

- Individual Object:

attributes: name, ID, birthdate, gender, security clearances, biometric profiles

methods: updateProfile(), addBiometricData(), getVerificationHistory() - FeatureSet Object:

attributes: featureType, featureData (e.g., facial landmarks, gait patterns)

methods: compare(otherFeatureSet), serialize(), deserialize() - VerificationResult Object:

attributes: timestamp, confidenceScore, matchingDetails, anomalies

methods: store(), retrieve(), generateAlert() - DataAcquisitionModule Object:

attributes: sourceType, connectionParameters

methods: connect(), acquireData(), disconnect() - FeatureExtractionEngine Object:

attributes: algorithmType, parameters

methods: extractFeatures(data), trainModel(), evaluatePerformance() - MatchingEngine Object:

attributes: matchingThreshold,

## COMPONENT DESIGN

## HUMAN INTERFACE DESIGN

## Overview of User Interface

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

## Screen Images

Display screenshots showing the interface from the user’s perspective. These can be hand­ drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

## Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

## REQUIREMENTS MATRIX

Provide a cross­reference that traces components and data structures to the requirements in your softwarerequirements specification (SWRS) document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SWRS. Refer to the functional requirements by the numbers/codes that you gave them in the SWRS.

## APPENDICES

***A. Glossary of Terms:***

**Biometric Verification:**

The process of verifying an individual's identity by comparing their unique physical or behavioral characteristics (e.g., fingerprints, facial features, iris patterns) to stored data.

**Data Acquisition:**

The process of retrieving and collecting data from various sources, including real-time feeds, databases, and sensors.

**Feature Extraction:**

The process of identifying and extracting relevant characteristics from collected data, such as facial landmarks from video frames or text keywords from a document.

**Ground Truth:**

The verified and true identity of an individual used as a reference point for comparison in verification tasks.

**Matching Engine:**

The core component of Relintel responsible for comparing extracted features and making identity verification decisions in real-time.

**Multi-Factor Authentication (MFA):**

A security measure requiring multiple verification factors (e.g., password, one-time code, biometric scan) to confirm a user's identity.

**Real-Time Data:**

Data that is generated and processed continuously, providing immediate insights into current events and actions.

**Security System:**

A combination of hardware, software, and procedures designed to protect assets and information from unauthorized access, use, disclosure, disruption, modification, or destruction.

**Verification:**

The process of confirming the accuracy of a claimed identity through comparison with existing information and trusted data sources.

***B. Detailed Data Structures and Schemas:***

This appendix will provide detailed descriptions of the data structures used within Relintel, including:

**Individual:**

Data fields such as name, ID, demographics, security clearances, biometric profiles.

**FeatureSet:**

Type and format of extracted features based on source data (e.g., facial landmarks, gait vectors, textual features).

**VerificationResult:**

Confidence score, matching details, timestamps, potential anomalies.

**Data Acquisition Module:**

Parameters for connecting to different data sources.

**Feature Extraction Engine:**

Algorithm parameters and configurations.

**Matching Engine:**

Internal data structures for feature comparison and decision-making.

***C. API Reference Examples:***

This appendix will showcase practical examples of using the Relintel API for common functions, including:

* Acquiring data from specific sources (e.g., CCTV camera, database).
* Extracting features from various data types (e.g., face, gait, text).
* Performing identity verification and retrieving results.
* Configuring multi-factor authentication and adaptive settings.

***D. Security Considerations and Threat Models:***

This appendix will address potential security risks associated with Relintel and propose mitigation strategies, covering aspects like:

* Data privacy and access control.
* Authentication and authorization mechanisms.
* System vulnerabilities and attack vectors.
* Fault tolerance and disaster recovery.

***E. Performance Benchmarks and Scalability Analysis:***

This appendix will present results from performance tests conducted on Relintel, evaluating aspects like:

* Processing speed for different data types and workloads.
* Resource utilization and memory footprint.
* Scalability under increased data volume and user load.

***F. Algorithm Descriptions and Technical Details:***

This appendix will provide in-depth explanations of the algorithms used within Relintel, focusing on:

* Feature extraction techniques for various data sources.
* Matching engine algorithms and comparison methods.
* Machine learning models and training strategies.

***G. Related Standards and Regulations:***

This appendix will identify relevant industry standards and regulations that Relintel adheres to or considers regarding:

* Data privacy and security (e.g., GDPR, HIPAA).
* Biometric data collection and usage.
* System security and vulnerability management.

1. Available to LANL users at <http://ieeexplore.ieee.org/browse/standards/collection/ieee> [↑](#footnote-ref-1)