**AACS Architectural Design**

**About This Document**

Document Information

|  |  |
| --- | --- |
| **Issuing authority** | Team 3 |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Verion** | **Date** | **Comment** | **Author** | **Approver** |
| 1.0 | 2021-06-21 | Initial Release | Team 3 |  |
|  |  |  |  |  |
|  |  |  |  |  |

Purpose

This document specifies the software architecture design for AACS to support Software Requirement. This design document also serves as a guideline on how each software components in the system should be implemented and how the components should interact with each other.

Scope

* Architectural Drivers
* SW Architectural Representations
* Architectural Alternatives
* Quality Attribute Scenarios

Related Documents

Documents related to this document include :

* AACS\_Requirements
* AACS\_ThreatAnalysis\_RiskAssessment\_Result

Acronyms / Glossary

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| AACS | AI Attendance Check System |
| FRS | Face Recognition System |
| ACS | Attendance Check System |

Table of Contents

[**Architectural Drivers**](#_67cdizv7fa99) **4**

[SW Main Features](#_sme5j7kw6w82) 4

[Quality Attributes](#_ham25eadxm5k) 5

[Constraints](#_26773dpuncr0) 6

[**SW Architectural Representations**](#_a38nao9qmxv) **8**

[Initial Dynamic View](#_703jh5i0depd) 8

[SW Component Descriptions](#_4y8d2do1jtfa) 8

[**Refined Architecture**](#_5a0kcjm2goks) **10**

[**Dynamic view with security applied after threat analysis**](#_2s3jj79ecaom) **10**

[Static View](#_o4ey8lh15wfw) 11

[Architectural Alternatives](#_8jbs42m67gw) 11

[Physical View](#_vz5hpkdr23cd) 12

[Secure Design Pattern](#_n0dpdks20sy3) 13

# Architectural Drivers

## SW Main Features

**Example 1**

**Table 1** Software Main Features 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Level 1** | **Level 2** | **Level 3** | **Descriptions** |
| AACS | FRS | Communication | -Communication on the system is between FRS and ACS.  -Provides a function for secure communication between FRS and ACS.  -A protocol is defined for communication between two systems, and communication is performed accordingly. |
| Face Recognition | -Provides a face recognition function to check the attendance of students in the class.  -Face recognition provides the process of registering new students and the ability to recognize registered students through images. |
| User Auth | -Provides authentication function for users.  -The user information transmitted from ACS is compared with the authorized user information in the system for authentication. |
| ACS | Communication | -Communication on the system is between FRS and ACS.  -Provides a function for secure communication between FRS and ACS.  -A protocol is defined for communication between two systems, and communication is performed accordingly. |
| User Auth | -Provides authentication function for users.  -The information received from the user through the UI is transmitted to the FRS system to check whether the user is a valid user. |
| UI | -Provides a UI for system users.  -When a student accesses the system, it provides a function to register his or her face.  -When the administrator accesses the system, he or she can view the attendance status of the students. You can check the current attendance status and past attendance status through saved videos. |

## Quality Attributes

**Quality attribute refers to the characteristic attributes of a product. Satisfying quality attributes can satisfy customers' requirements for quality.**

**Quality attributes were derived through customer requirements.**

**In addition, some Quality attributes were derived through threat analysis.**

1. ID: Quality Attribute ID with AACS-QA-xxx
2. Properties : Types of quality attributes
3. Contents : Detailed description of quality attribute
4. Importance : Scoring importance to the system on a scale of 1 to 5.
5. Difficulty : The difficulty in implementing the Quality Attribute is scored on a scale of 1 to 5.
6. Priority : Importance x Difficulty = Priority given by Quality Attribute. Accordingly, the quality attribute that this system should have is determined.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Properties | Contents | | Importance |  | Difficulty |
| AACS-QA-001 | Performance | The system must deliver video as close to real time as possible,  especially in real-time mode. | | 2.5 |  | 4 |
| AACS-QA-002 | Authentication | The system must use two factor authentication for sign on and  user credentials must be protected. Lost or compromised credentials must be handled in a reasonable way. | | 4 |  | 3 |
| AACS-QA-003 | Communication privacy | When in the desired mode the system must ensure that  data sent to a user remains private while in transit. No intermediary should be able to snoop or spy on an ongoing video feed. | | 5 |  | 4 |
| AACS-QA-004 | Proof of identity (nonrepudiation) | Users should be confident that the camera  they are using is the one that they believe it is. | | 2 |  | 4 |
| AACS-QA-005 | Multi-user privacy: | The system must ensure that multiple video feeds remain  private between the intended users. | | 4 |  | 3 |
| AACS-QA-006 | reliability | The system must ensure that video is reliably delivered. The system  should recover from networking errors as soon as possible. The goal is to  maintain a secure, performant connection at all costs. | | 2.5 |  | 4.5 |
| AACS-QA-007 | Testing | Ensure the developed software is adequately tested. | | 3 |  | 3.5 |
| AACS-QA-008 | Availability | Conduct proper fault/error detection, recovery and reporting. | | 4 |  | 4 |
| AACS-QA-009 | Security | Ensure the developed software adheres to the company coding standard and quality standards. | | 4 |  | 4 |
| AACS-QA-010 | Security | Key management for system must be secure. | | 4 |  | 4.5 |

**As above, the main Quality Attribute that the system should have is internally determined for items 3 and 10 according to the priority according to Importance and Difficulty. We will think about the design direction to achieve this.**

## Constraints

**Describe the restrictions on this system.**

**List the business and technical limitations of this system.**

**This constraint will serve as a driver for architectural design.**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | constraints | Summary | Contents |
| AACS-Const-001 | Business constraints | Development schedule | Phase 1 period is 3 weeks |
| AACS-Const-002 | Budget issue | No additional budget for development environment |
| AACS-Const-003 | Technical constraints | Development Language | C/C++ |
| AACS-Const-004 | Development Board  (Jatson Nano) | GPU: 128-core NVIDIA Maxwell™ architecture-based GPU  CPU: Quad-core ARM® A57  Video: 4K @ 30 fps (H.264/H.265) / 4K @ 60 fps (H.264/H.265) encode and decode  Camera: MIPI CSI-2 DPHY lanes, 12x (Module) and 1x (Developer Kit)  Memory: 4 GB 64-bit LPDDR4; 25.6 gigabytes/second  Connectivity: Gigabit Ethernet  OS Support: Linux for Tegra®  Module Size: 70mm x 45mm  Developer Kit Size: 100mm x 80mm |
| AACS-Const-005 | PC Development Tool | MS Visual Studio |
| AACS-Const-006 | Router | TP-Link ac1750 |
| AACS-Const-007 | no physical modifications | no physical modifications to Jetson Nano. |
| AACS-Const-008 | selected router | use of the supplied and configured router. |

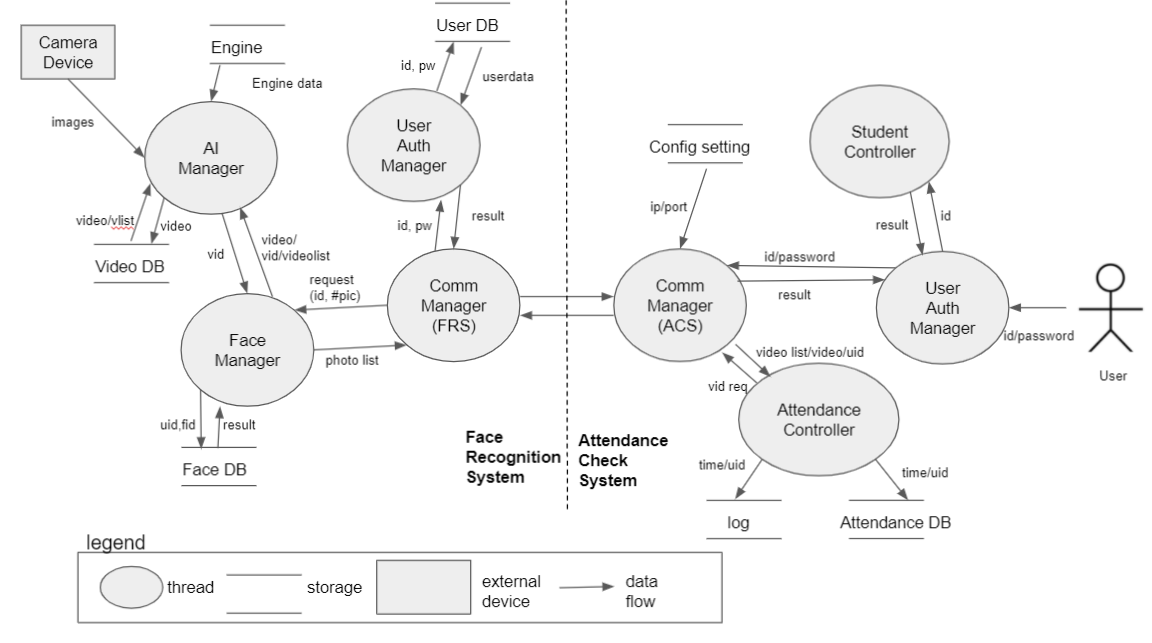
# SW Architectural Representations

## Initial Dynamic View

FRS consists of a component for performing a function for recognizing faces to enroll students.

ACS is composed of components to provide a function for attendance to students and administrators by providing UI.

A component is defined between the ACS and FRS system, and data flowing through the component is defined.



**Figure 1 Dynamic View**

### SW Component Descriptions

Below table describes the software components which are specified in the chapter 2.1.

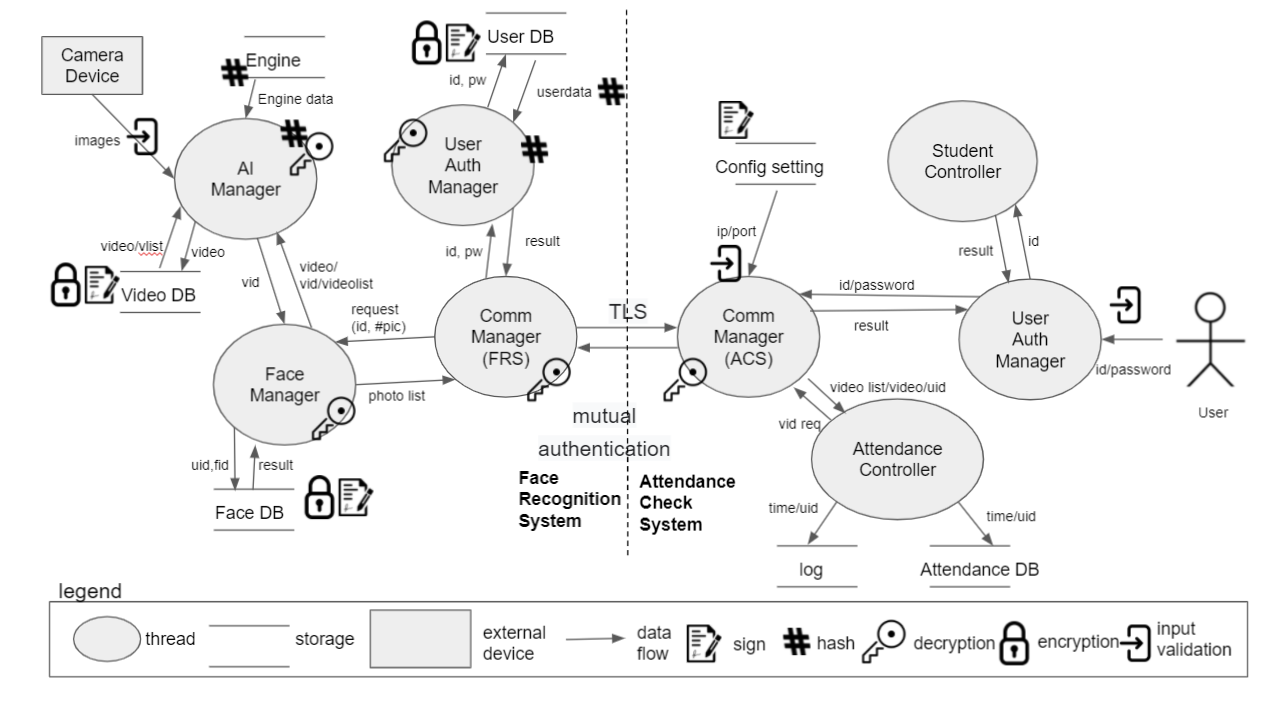
**Table 6 SW Component Descriptions**

|  |  |  |  |
| --- | --- | --- | --- |
| **System** | **Component** | **R&R** | **Description** |
| **FRS** | AI Manager | This component provides the ability to recognize a person's face when attending. | This component provides the ability to verify faces through the AI ​​engine. |
| Face Manager | This component manages face photos registered as students. | This component manages student faces through face DB. |
| Comm Manager | This component provides functions for communication between FRS and ACS. | Communication function provides secure mode in addition to real-time mode. |
| User Auth Manager | This component provides functions for user authentication. | For user authentication, it is checked whether the user is a previously registered user through the user DB. |
| **ACS** | Attendance Controller | This component provides a function to show the students in attendance through the UI. | This component provides a function to check student attendance, tardiness, and absence based on school attendance time. |
| Student Controller | This component provides a UI function for registering a student's face. | This component provides the ability to add and delete student photos. |
| Comm Manager | This component provides functions for communication between FRS and ACS. | Communication function provides secure mode in addition to real-time mode. |
| User Auth Manager | This component provides functions for user authentication. | Provides a function to receive user information from the UI for user authentication. |

# Refined Architecture

## Dynamic view with security applied after threat analysis

The design below is an architecture that reflects the security requirements derived through risk assessment based on the contents analyzed through threat analysis.



**Figure 2 Dynamic View**

The above architecture reflects the security requirement and the following functions are added.

1. Encryption and sign are applied to the DB managed by the system to correspond to confidentiality and integrity.

2. Input validation was applied to prevent errors in the values ​​input through the system.

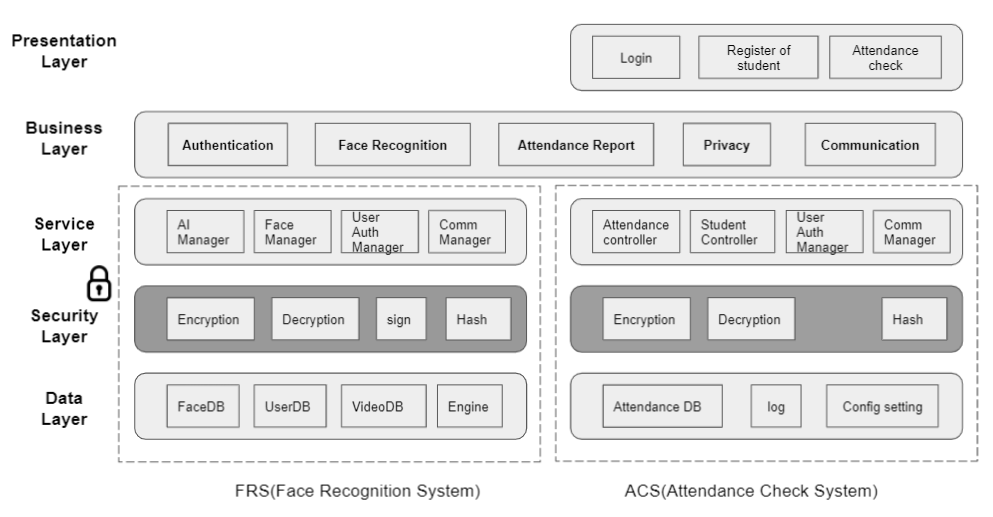
3. The TLS method is applied to secure communication between systems.

4. Hash is applied for security such as password in the system.

## Static View

Through the AACS static view, it could show the modules for each layer of the system.

A security layer has been added to satisfy the security requirement as shown below.



**Figure 4 Static View**

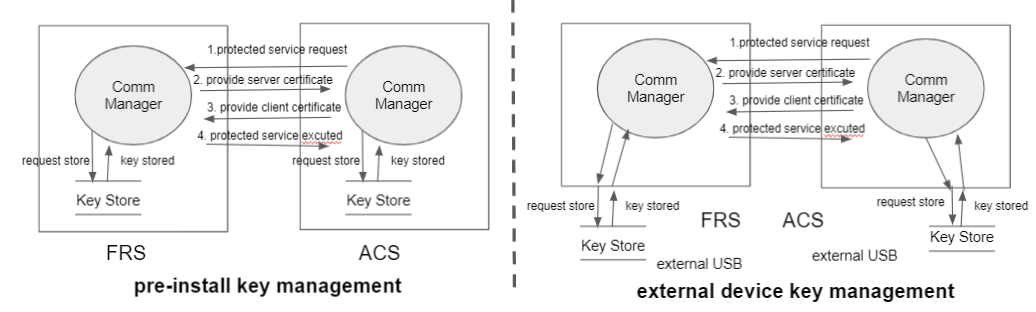
## Architectural Alternatives

AACS believed that if data was leaked or compromised, it could cause big confusion and damage to CMU

How can we keep our data safe?

We thought that the most efficient way would be to manage the encryption key more securely.

We had to think about key management.



**Figure 5 Architectural Alternatives**

|  |  |  |
| --- | --- | --- |
|  | **Pre-installed key management** | **External Device key mangement** |
| Pros | Easy to distribution  Easy to key management | Update usb to update key  Securely distribution  Secure key storage |
|
|
| Cons | Update the whole program to update the key  The key is easy to be exposed to risk | Physical usb key management  Difficult to distribution |

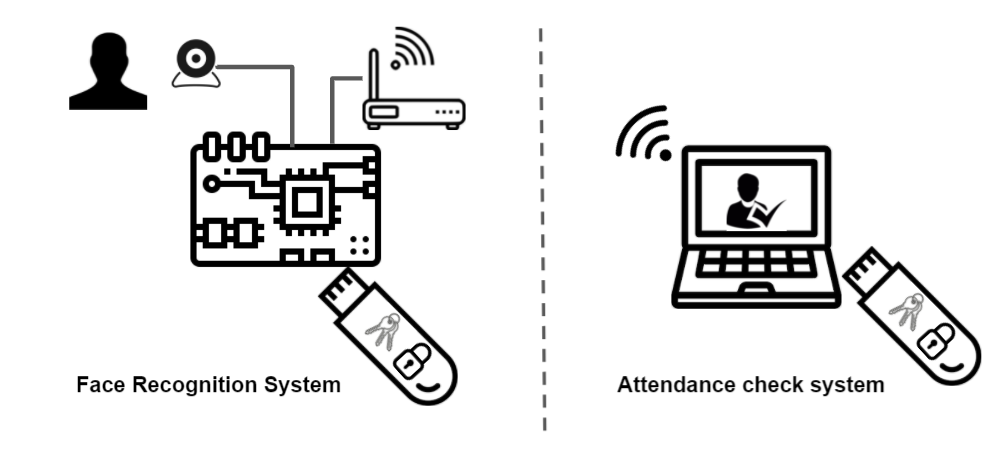
**Architectural Decision :**

Selected as an **external device key management** for key lifecycle management and secure key management

## Physical View

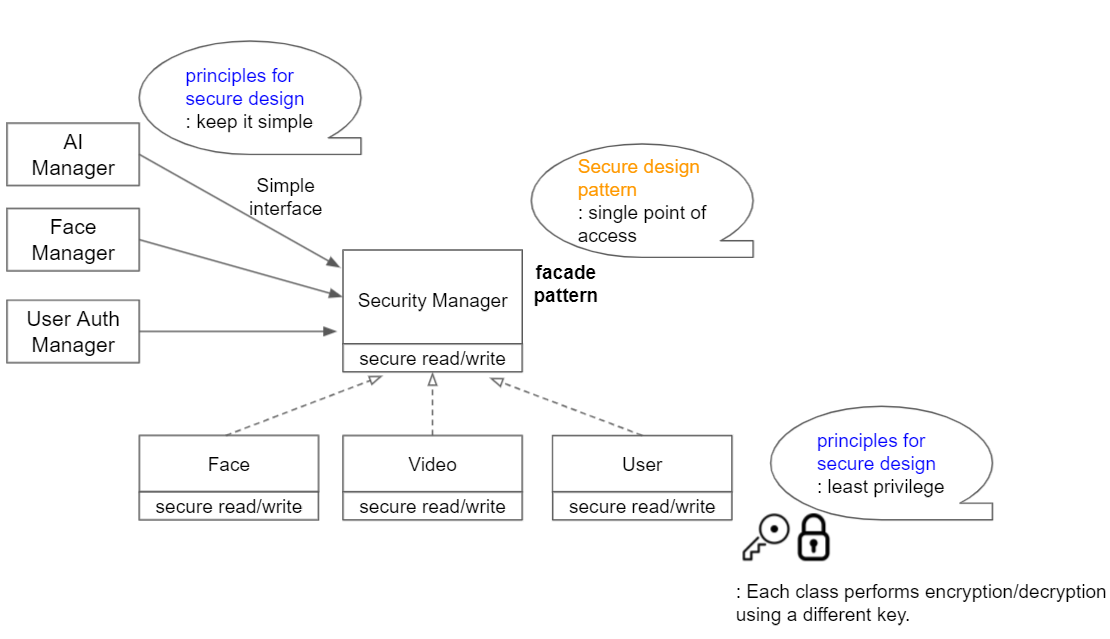
The architecture reflecting the architectural decision is expressed below.

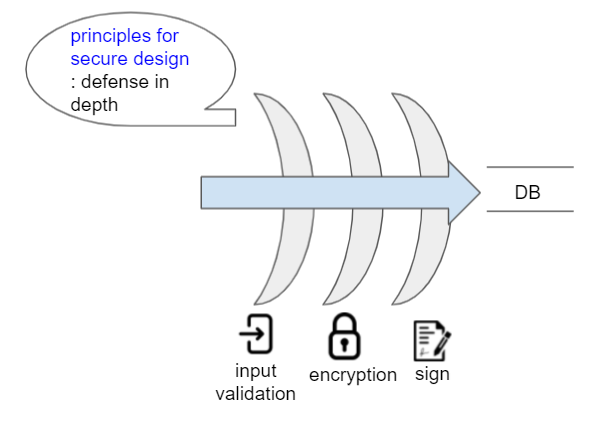
The additional physical device is expressed through the physical view as shown below.

****

**Figure 6 physical view**

## Secure Design Pattern





-The End-