**Broadcast Feature between**

**FOTA and UDS Service with AIDL**

**(Version 1.0.0)**  
**System Requirements Specification (SRS)**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Prepared By | Review By | Version No |
| 1 | Venu | Hareesha/Aparna | 1.0.0 |
| 2 | Jayanth | Hareesha/Aparna | 1.0.0 |
| 3 | Varshitha | Hareesha/Aparna | 1.0.0 |

**Table of Contents**

**1.1 Introduction3**

**1.2 System Overview3**

**1.3 System Features 3**

**1.4 System Interfaces 4**

**1.5 System Requirements 4**

**2.1 System Architecture 5**

**2.2 Data Flow 5**

**2.3 Error Handling 6**

**2.4 Security Considerations 6**

**2.5 Testing 6**

**1.1 Introduction**

This document outlines the requirements for a new broadcast feature that enables communication between the FOTA (Firmware Over-The-Air) manager and the UDS (Unified Diagnostic Service) service using Android Interface Definition Language (AIDL). This feature will allow the FOTA manager to initiate ECU (Electronic Control Unit) flashing and other diagnostic operations and receive updates and notifications from the UDS service.

**1.2 System Overview**

The system consists of the following components:

* **FOTA Manager**: Initiates update requests (bootloader or application) and receives updates from the UDS service.
* **UDS Service**: Processes update requests, interacts with the UDS core service and vehicle ECU, and provides callbacks to the FOTA manager.
* **UDSEventCallback**: Interface defining callback methods for the FOTA manager to receive updates and notifications from the UDS service.

**1.3 System Features**

* The FOTA manager can submit update requests (including necessary data) to the UDS service.
* The UDS service can send progress updates, informational messages, and operation completion notifications to the FOTA manager through the UDSEventCallback interface.
* The UDS service can notify the FOTA manager of errors or exceptions encountered during operations.
* The UDS service can provide new data or updates to the FOTA manager when available.
* The UDS service can indicate critical failures that might prevent further operations.

**1.4 System Interfaces**

* **AIDL Interface:** The UDSService class and UDSEventCallback interface define the communication channel between FOTA and UDS using AIDL.

**1.5 System Requirements**

* The system shall support both bootloader and application update requests.
* The FOTA manager shall be able to register a callback object to receive updates from the UDS service.
* The UDS service shall provide detailed progress updates during operations.
* The UDS service shall notify the FOTA manager of successful operation completions and any errors encountered.
* The UDS service shall handle data received from the ECU and make it available to the FOTA manager if necessary.
* The system shall be designed to handle critical failures gracefully and notify the FOTA manager.

**2. System Design Document (SDD)**

**2.1 System Architecture**

The system will leverage AIDL for inter-process communication between the FOTA manager and UDS service residing in separate Android processes. The **UDSEventCallback** interface defines methods for the UDS service to send updates and notifications to the FOTA manager.

**2.2 Data Flow**

1. The FOTA manager prepares an update request with details in a Bundle object.
2. The FOTA manager calls the SubmitRequest method on the UDSService object through the AIDL interface.
3. The UDS service receives the request and processes it.
4. The UDS service interacts with the UDS core service and vehicle ECU to perform the requested operation.
5. During the operation, the UDS service can utilize the registered UDSEventCallback object to send updates to the FOTA manager:

* progressUpdate(int progress): Provides progress updates.
* udsLog(String Message): Sends informational or debug messages.
* operationComplete(int status, String Message): Notifies of successful operation completion.
* onError(int errorCode, String errorMessage): Reports errors encountered during operations.
* onDataAvailable(out Bundle operationData): Provides new data from the UDS service (if applicable).
* onCriticalFailure(int statusEn, String errorMessage): Indicates critical failures.

1. The FOTA manager receives these callbacks and takes necessary actions based on the information provided (e.g., update UI, handle errors).

**2.3 Error Handling**

* The UDS service shall return appropriate error codes through the onError callback for different error scenarios.
* The FOTA manager shall implement logic to handle specific error codes and display user-friendly error messages.
* Critical failures notified via onCriticalFailure shall be treated with high priority, and the FOTA manager might need to enter a recovery state or notify the user to contact support.

**2.4 Security Considerations**

* AIDL enforces security by restricting communication between processes with the same user ID.
* Additional security measures might be implemented within the UDSService to validate and sanitize update request data received from the FOTA manager.

**2.5 Testing**

Unit tests should be implemented to verify the functionality of the UDSService class and its interaction with the UDS core service.