ADAS SESIP Profile

Document number:	JSADEN027
Version:	1.0
Release Number:	01
Author:	Arm Limited
Authorized by:	Arm Limited
Date of Issue:	21/05/2025

Abstract

Vehicles have increasingly sophisticated Advanced Driver Assistance Systems (ADAS) that can enhance vehicle safety and driving comfort. The developers of such systems should use a threat model to provide a structured security analysis that results in a set of security requirements that must be met. This document builds on a ADAS threat model written by Autonomous Vehicle Computing Consortium (**AVCC**) and converts it to a SESIP Profile that could be used as the basis for a Security Target and lab based evaluation. The appendix includes a section on how a System on Chip (SoC) with a PSA Certified Root of Trust (RoT) can help meet many of the ADAS Security Functional Requirements (SFRs).

For better context, please read the **AVCC ADAS threat model** before reading this SESIP Profile.

Keywords

ADAS, AVCC, Platform Security Architecture, Protection Profile, PSA Certified, SESIP

Non-Confidential Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third-party patents.

THIS DOCUMENT IS PROVIDED “AS IS”. ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arm, then the click through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm’s trademark usage guidelines at http://www.arm.com/company/policies/trademarks.

Arm Limited. Company 02557590 registered in England.
110 Fulbourn Road, Cambridge, England CB1 9NJ.

Contents

1 About this document

1.1 Current Status and Anticipated Changes

1. Current Status: Release 01

1.2 Release Information

2. The change history table lists the changes that have been made to this document.

Date	Version	Confidentiality	Change
21/05/2025	1.0	Non-confidential	First version

1.3 References

3. This document refers to the following informative documents.

Ref	Doc No	Author(s)	Title
[SESIP]	GP_FST_070	GlobalPlatform	Security Evaluation Standard for IoT Platforms (SESIP), Version 1.2, July 2023
[PSAL2]	JSADEN012	PSA JSA	SESIP Profile for PSA Certified Level 2
[PSAL3]	JSADEN011	PSA JSA	SESIP Profile for PSA Certified Level 3
[PSAL3iSE]	JSADEN018	PSA JSA	SESIP Profile for PSA Certified Level 3 iSE/SE and RoT Component
[PSAL4iSE]	JSADEN023	PSA JSA	SESIP Profile for PSA Certified Level 4 iSE/SE and RoT Component
[AVCC]	Technical Report 006	AVC Consortium	Baseline Cybersecurity for Automated And Assisted Driving Systems, Version 1.0, March 2024
[J3101]	J3101	SAE International	Hardware Protected Security for Ground Vehicles, February 2022

1.4 Terms and Abbreviations

4. This document uses the following terms and abbreviations

Term	Meaning
ADAS	Advanced Driver Assistance System
AI	Artificial Intelligence
API	Application Programming Interface

ARoT	Application specific Root of Trust
AV	Automated Vehicles
CPU	Central Processing Unit
DCS	Driving Computer System
DDT	Dynamic Driving Task
NSPE	Non-Secure Processing Environment
OEM	Original Equipment Manufacturer
OS	Operating System
OTP	One-Time-Programmable
RAM	Random Access Memory
REE	Rich Execution Environment
ROM	Read Only Memory
RoT	Root of Trust
SFR	Security Functional Requirement
SPE	Secure Processing Environment
ST	Security Target
T1	Tier 1
TEE	Trusted Execution Environment

1.5 Feedback

5. We welcome feedback on its documentation.

6. If you have comments on the content of this documentation, send an e-mail to psacertified@arm.com. Give:

- The title (ADAS SESIP Profile).
- The number (JSADEN-027) and version.
- The page numbers to which your comments apply.
- The rule identifiers to which your comments apply, if applicable.
- A concise explanation of your comments.

7. We also welcome general suggestions for additions and improvements.

8. **Note:** PDFs are tested only in Adobe Acrobat and Acrobat Reader and cannot guarantee the appearance or behavior of any document when viewed with any other PDF reader.

2 Introduction

9. This SESIP Profile targets Driving Computer System (DCS) used in Advanced Driver Assistance Systems (ADAS). It builds upon the Baseline Cybersecurity for Automated And Assisted Driving Systems document [AVCC] from the AVC consortium. That document establishes a structured approach for identifying security objectives, addressing threats on ADAS systems, and ensuring compliance with industry regulations such as ISO/SAE 21434 and UNECE R155. It is recommended that readers review it first to ensure a complete understanding of the topics covered here.

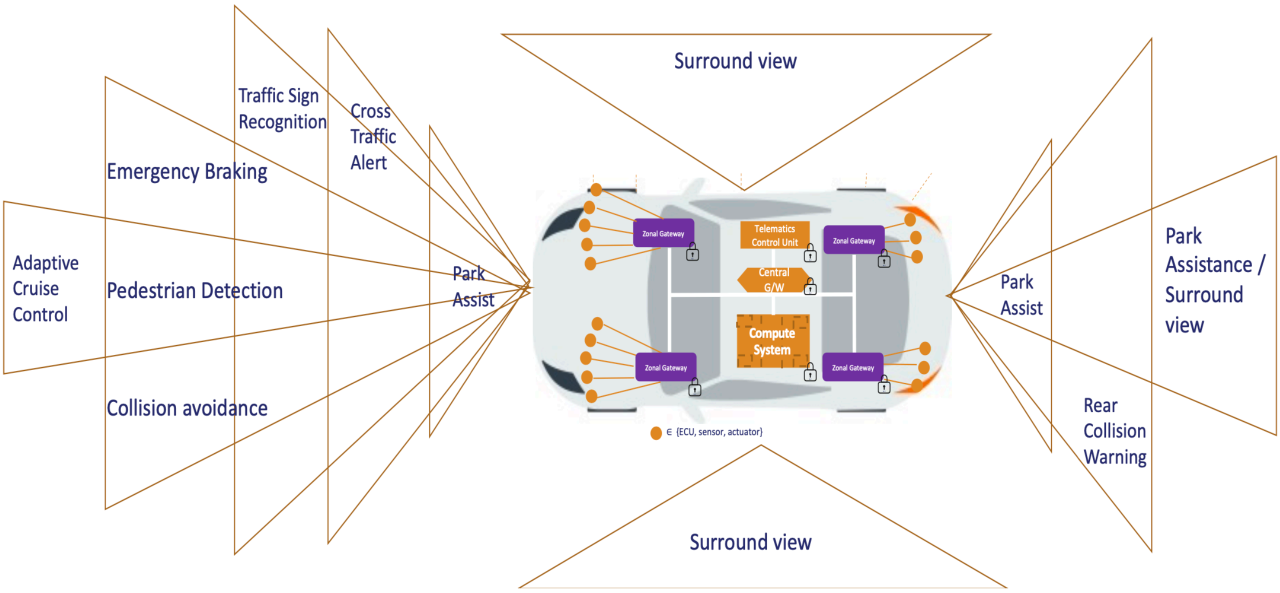

Figure 1: Examples of ADAS systems (figure from [AVCC])

10. This SESIP Profile transposes of the [AVCC] document in the SESIP framework. Especially, it provides SESIP an evaluation framework and SESIP SFRs for the [AVCC] baseline security requirements.

11. The considered platform is composed of hardware, firmware, and software components responsible for the real-time execution of Dynamic Driving Tasks (DDT), including data processing, environment perception, and motion control. It interfaces with external sensors, actuators, and communication modules to enable safe and reliable autonomous or semi-autonomous vehicle operation.

2.1 Profile Reference

12. See title page.

2.2 Platform Reference

Platform name	*<Platform name>*
Platform version	*<Platform version>*
Platform identification	*<Platform id details>*

Platform Type	Hardware device and a firmware implementing ADAS functionalities

2.3 Platform Functional Overview and Description

2.3.1 Usage and Major Security Features

13. Driving Computer Systems process data from various sensors, use algorithmic and machine-learning models to understand the vehicle's surroundings, plan a course of action, and interact with actuators to execute driving decisions.

14. In cybersecurity, the DCS is a focal point because it must protect sensitive data, maintain functional integrity, and ensure secure communication with other vehicle components, such as sensors, actuators, and external services. Its role is critical in maintaining both the safety and security of autonomous and assisted driving systems. Detailed lists of assets with security properties and damage scenarios as well as threats on these assets are available in Section 4 of [AVCC].

15. To mitigate these threats, driving Computer Systems should include at least the following security features:

- Secure operational life cycle of the DCS:
  - Secure start up (see “Secure Initialization of Platform”).
  - Device commissioning. The device uniquely identifies itself (see “Verification of Platform Identity” and “Verification of Platform Instance Identity”) and shows it is genuine (see “Attestation of Platform Genuineness” and “Attestation of Platform State”) to the administrator.
  - Software update. The software running on the DCS can be updated to fix vulnerabilities identified or add new features after the device's deployment. See “Secure Update of Platform” and Flaw Reporting Procedure (ALC_FLR.2).
  - Secure access to debug features, if any, see “Secure Debugging”.
  - Protection against physical attacks, see “Physical Attacker Resistance”.
- Protection of vehicle, perception model and private data through security measures for data at rest (see “Secure Update of Platform” and “Perception Model”) and data in transit (see “Secure Communication Support”) and privacy protection (see “Privacy”).
- Cryptographic support. See “Cryptographic Operation”, “Cryptographic Random Number Generation”, “Cryptographic Key Generation”, “Cryptographic KeyStore” and “Authenticated Access Control”.
- Log of security events. Security events are logged locally on the DCS, to support forensic analysis of an attack or other suspicious event. See “Audit Log Generation and Storage”.

2.3.2 Protecting AI assets

Autonomous vehicles based on AI models will necessarily embed and use high-value assets: software libraries, hardware drivers, AI stacks, model weights, and biases. All those assets require protection for multiple reasons:

- A malicious attacker disturbing or modifying the AI stack could potentially lead to vehicle unavailability, ransomware, or have disastrous effects on safety.

- Competitors may want to extract and copy software, weights, biases for offline study or replication. Attackers may want to do so to find vulnerabilities to exploit in later attacks.

Trusting AI assets starts with several best practices:

- Secure boot: have ways to validate the stack and detect unwanted modifications before start.
- Secure Firmware Updates: make sure all updates are signed by a trusted party and deployed without modifications.
- Isolation: run all AI inference and learning within trusted boundaries.
- Maintain runtime integrity, e.g. using memory protection mechanisms such as memory-safe languages, pointer authentication, branch protection, and memory tagging.
- Defense: authenticate and validate all sensor inputs.

Models and weights are likely to need regular updates. If they are proprietary and high value an enhanced secure firmware update flow may be appropriate. For example, the process might start with an attestation and verification step so that platform state is known before download begins.

Local regulations may also enforce the need to protect privacy for all data captured by sensors about their environments, e.g. continuous video recording of the surroundings of parked vehicles may be constrained by law. This involves encrypted data storage and transmission (to the cloud, or to the owner's phone) to maintain confidentiality.

Securing AI assets is crucial to protect company IP present in models, algorithms, and software, and maintain trustworthy usage of ADAS functions of the vehicle to guarantee user safety and privacy.

2.3.3 Platform Architecture

16. *<A short introduction and description of the Platform, the combination of hardware and software to be evaluated, must be provided. Typically, this would be taken from the datasheet.>*

17. The Platform is the combination of hardware and software that provide a runtime environment and related applications for a Driving Computer System Platform. It is to be embedded in a hardware device that provides the power source, includes the network interfaces or other hardware used by the DCS, but which are not part of the scope of evaluation.

18. Figure 2 illustrates the main components for a Driving Computer System Platform in this Profile <*Replace this generic figure according to the specific Platform architecture and scope*>. It distinguishes between a Secure Processing Environment (SPE), in charge of the platform root of trust functions, such as secure boot, secure update, secure storage, and the Non-Secure Processing Environment, in charge of supporting sensor input, world generation, motion planning, actuation or other OEM/Tier 1 applications.

19. The Secure Processing Environment can also support applications, illustrated as Applications Root of Trust (ARoT) in Figure 2. For instance, the DCS can use the SPE to host an ARoT for OEM secure services.

20. *<Add all the necessary details for the software scope: libraries, drivers, versions, …>*

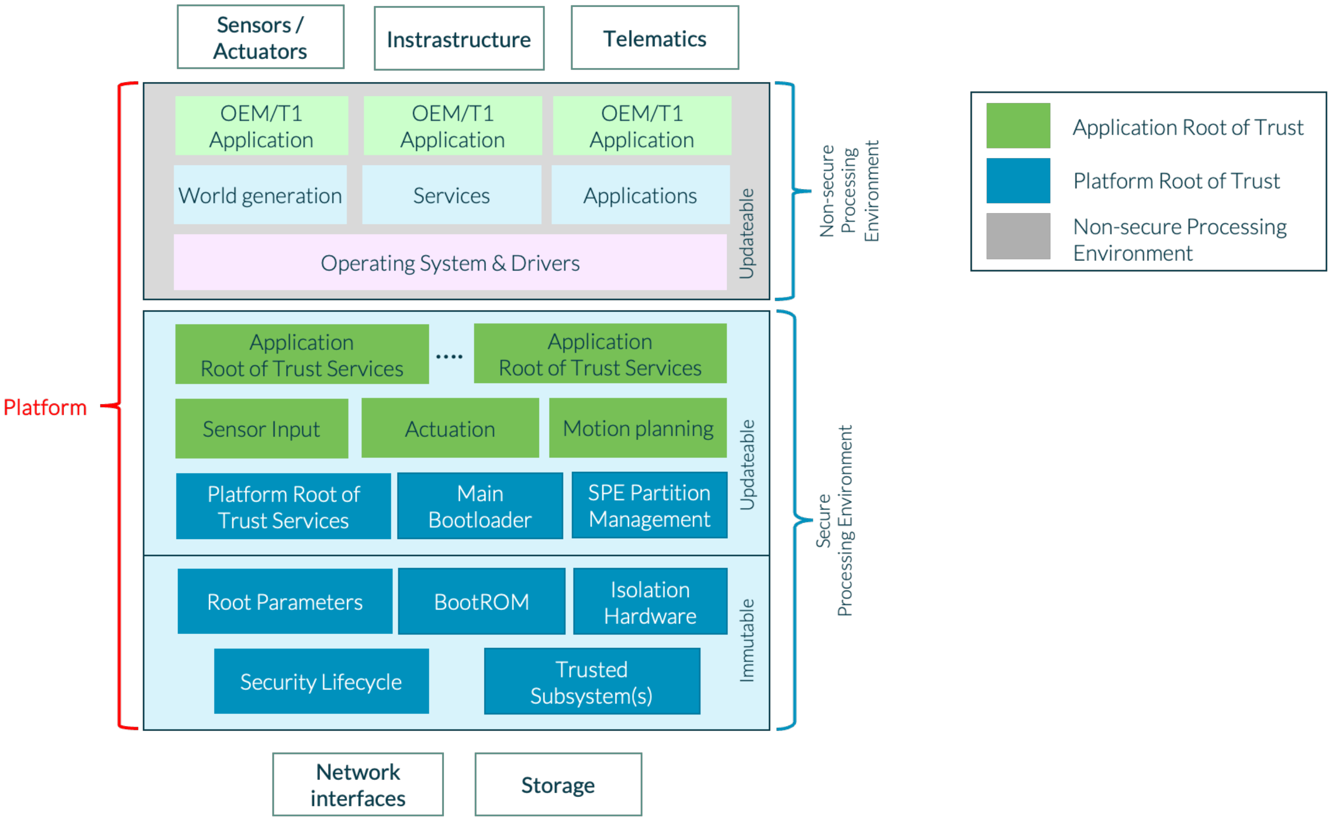

Figure 2: Driving Computer System Platform

21. The Physical scope for the Platform is typically composed of a PCB containing a SoC or module with a high performance multi core microprocessor, GPU, AI accelerator and networking capabilities. The SoC supports, secure boot and isolation between SPE and NSPE and non-volatile memory to store sensitive data, such as perception model. *<write specific scope details, which may be a silicon chip, a PCB, … >*

22. The out-of-scope part comprises *<to be completed by developer>*.

3 Security Objectives for the Operational Environment

23. The [AVCC] document provides a threat analysis for the considered platform in this profile as well as security objectives for the operational environment. This section directly reuses these security objectives.

24. For the platform to fulfil its security requirements, the operational environment (technical or procedural) must fulfil the following objectives.

3.1.1 Trusted Users

25. The Platform users are trusted and trained in the handling of the Platform. The Platform users are authorized to access confidential information.

3.1.2 Trusted Host

26. The host processors are trusted. Only authorized users (via authorized applications) have access to the host processors.

3.1.3 Legitimate Usage

27. The Platform should be under the control of the trusted user in a trusted environment.

3.1.4 Lost Storage Device

28. The Storage device should be discarded if the device is lost or stolen and later recovered, if it may be suspected that an unauthorized party has tampered with the device.

3.1.5 Entropy

29. The user secret should have sufficient entropy to render an exhaustive search concerning the user secret computationally infeasible for the intended class of adversaries.

3.1.6 Crypto

30. The cryptographic keys should have sufficient strength to render an exhaustive search concerning the cryptographic keys computationally infeasible for the intended class of adversaries.

3.1.7 Credential Management

31. The cryptographic keys, credentials and certificates used in the Platform shall be securely generated and provisioned to the Platform.

32. Additionally, they should be securely managed during the life cycle of Platform when used outside of the Platform (such as in gateways, back-end servers or maintenance devices).

3.1.8 Others

33. *<ST writer: list all other mandatory objectives for the environment with reference to where in the guidance documents this objective is described.>*

4 Security Requirements and Implementation

34. The [AVCC] technical report includes a comprehensive threat analysis and baseline security requirements for DCS. This document builds upon these security requirements to define SESIP SFRs for a SESIP security evaluation. The Appendix B provides a mapping between baseline security requirements from [AVCC] and SFRs from this SESIP Profile.

4.1 Security Assurance Requirements

35. The claimed assurance requirements package is SESIP3, as described in Section 5.1.

4.1.1 Flaw Reporting Procedure (ALC_FLR.2)

36. In accordance with the requirement for a flaw reporting procedure (ALC_FLR.2), including a process to generate any needed update and distribute it, the developer has defined the following procedure:

37. *<ST writer: Describe the procedure, including where flaws can be reported (website and/or email address), how the reported flaws are handled in a timely manner, and how an application developer/end-user can get informed of the update. The process to receive flaw reports and handle them in a timely manner needs to be described.>*

4.2 Security Functional Requirements

38. Platforms conformant to this Profile must satisfy the following security functional requirements.

4.2.1 Verification of Platform Identity

39. The platform provides a unique identification of the platform, including all its parts and their versions.

40. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

4.2.2 Verification of Platform Instance Identity

41. The platform provides a unique identification of that specific instantiation of the platform, including all its parts.

42. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

4.2.3 Attestation of Platform Genuineness

43. The platform provides an attestation of the “Verification of Platform Identity” and “Verification of Platform Instance Identity”, in a way that ensures that the platform cannot be cloned or changed without detection.

44. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

4.2.4 Attestation of Platform State

45. The platform provides an attestation of the state of the platform, such that it can be determined that the platform is in a known state.

46. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

47. **Note** 1**:** Lifecycle phases include, for example, development, deployment, returns, and end-of-life. Various access control may be required to be lifecycle aware. Platform state depends on software versions, run-time measurements, hardware configuration, the status of debug ports, DCS lifecycle phase.

4.2.5 Secure Update of Platform

48. The platform can be updated to a newer version in the field such that the *<confidentiality,>* integrity and authenticity of the platform is maintained.

49. **Note** 2**:** The DCS must implement a formal update process complying with UNECE regulation R156 in a manner consistent with the best practice of ISO 24089. DCS Platforms should consider measures for ensuring the following baseline properties for updates.

50. **Note** 3**:** The software update shall include the perception model.

51. **Note** 4**:** Rollback to earlier firmware versions shall be prevented, except for very specific recovery purposes.

52. **Note** 5**:** Additionally, the DCS should implement a fail-safe mechanism, so that in case of an unsuccessful update, it can continue from a version known to be functional.

53. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the verifications performed by the secure update mechanism, the order of these verifications, the behaviour of the platform is case of a failed verification and the cryptographic material used for that purpose.>*

4.2.6 Secure Storage

54. *<ST writer: include in the security target at least one the three secure storage SFRs of Section 4.3 according to the available implementation on the platform.>*

55. **Note** 6: The secure storage SFR is used to protect sensitive data, for example, user or service credentials, or secret keys.

56. **Note** 7: User data shall be bound to platform instance or the data owner, and where necessary, to the security lifecycle state, for example, to deny access during debugging. The SPE can only really identify data owners where the identity is defined within the SPE. It cannot verify or enforce identity in the NSPE space.

4.2.7 Secure Initialization of Platform

57. The platform ensures its authenticity and integrity during platform initialization. If the platform authenticity or integrity cannot be ensured, the platform will go to *<list of controlled states>*.

58. **Note** 8: Secure initialization must cover all software parts of the evaluation. One possible exception applies to code not related to DCS if mechanisms are implemented so that such software cannot compromise the security of the DCS.

59. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include all stages of the boot chain and describe for each stage how the verification of the loaded software is performed, and the cryptographic material used for that purpose. This should also justify any excluded software parts from the secure initialization.>*

4.2.8 Residual Information Purging

60. The platform ensures that *<list of data>*, with the exception of *<list of data that is not erased automatically>*, is erased using the method specified in *<specification>* before the memory is used by the platform or application again and before an attacker can access it.

61. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

62. **Note** 9: List of data includes assets defined in [AVCC].

4.2.9 Secure Communication Support

63. The platform provides the application with one or more secure communication channel(s).

64. The secure communication channel authenticates *other vehicular components* and protects against *modification, replay and spoofing* of messages between the endpoints, using *<list of protocols and measures>*.

65. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the types of supported secure communication channels, the method they can be invoked, the used cryptographic material, the contexts they shall always be used.>*

66. **Note** 10: Other vehicular components include telematics, sensors, or actuators.

4.2.10 Audit Log Generation and Storage

67. The platform generates and maintains an audit log of *<failed and successful authentication attempts, firmware upgrade requests and progress, integrity errors, list of other significant security events>* and allows access and analysis of these logs following a specific *<access control policy>*.

68. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

69. **Note** 11: Audit log record should mention the nature of the event, date and time of the event and the user, if any, responsible for the event.

70. **Note** 12: Significant security events include at least failed and successful authentication attempts, firmware upgrade requests, version control and progress, integrity errors.

71. **Note** 13: Audit log storage relies on secure storage features.

4.2.11 Software Attacker Resistance: Isolation of Platform

72. The platform provides isolation between the application and itself, such that an attacker able to run code as an application on the platform cannot compromise the other security functional requirements.

73. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

74. **Note** 14: Isolation considered in this SFR is between trustworthy services and less trusted or untrusted services of the DCS. Depending on the architecture, the isolation can be for instance between SPE and NSPE or between PSA-RoT and Application Root of Trust Services.

75. **Note** 15: Interaction across isolation boundaries shall be subject to access control, validation of exchanged data and protection of confidentiality and integrity of any data exchanged.

4.2.12 Cryptographic Operation

76. The platform provides the application with *<list of cryptographic operations>* functionality with *<list of algorithms>* as specified in <*specification*> for key lengths *<list of key lengths>* and modes *<list of modes>*.

77. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

78. **Note** 16: All cryptographic schemes should be according to a scheme recognized by the regulatory authority or regulatory authorities under which the DCS operates.

79. **Note** 17: All cryptographic schemes should be capable of providing adequate protection for the envisaged operational life of the DCS. In practice, this may mean that it may have to be upgradable, see [SAE J3101].

4.2.13 Cryptographic Random Number Generation

80. The platform provides the application with a way based on *<list of entropy sources>* to generate random numbers to as specified in <*specification*>.

81. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

4.2.14 Cryptographic Key Generation

82. The platform provides the application with a way to generate cryptographic keys for use in *<list of cryptographic algorithms>* as specified in <*specification*> for key lengths *<list of key lengths>*.

83. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the storage locate of these assets and the cryptographic means and materials used to protect these assets.>*

4.2.15 Cryptographic KeyStore

84. The platform provides the application with a way to store *<list of assets, such as cryptographic keys and passwords>* such that not even the application can compromise the *<authenticity, integrity, confidentiality>* of this data. This data can be used for the cryptographic operations *<list of operations>*.

85. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the storage locate of these assets and the cryptographic means and materials used to protect these assets.>*

86. **Note** 18: Cryptographic keystore is used for cryptographic assets and operations related to attestation, secure update, secure storage, secure communication support, roles authentication (see Authenticated Access Control requirement) and secure debugging (see corresponding SFR).

4.2.16 Authenticated Access Control

87. The platform allows only *<list of role(s)>*, identified, authenticated and authorized as specified by *<specification>* to allow performing of *<operations on key materials>*.

88. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

89. **Note** 19: Access control to key materials should be based on key owner, usage, and validity (duration, lifecycle, geographic location, etc.).

4.2.17 Physical Attacker Resistance

90. The platform detects or prevents attacks by an attacker with physical access before the attacker compromises any of the other functional requirements.

91. *<ST writer: add a short conformance rationale describing how this is done and which types of physical attacks the platform is able to detect or prevent.>*

92. **Note** 20: The DCS should include hardware mechanisms to protect:

- integrity of the control flow from software attacks.
- power supplies and clocks from glitching or operating outside of their specified ranges.
- ports from specious messages or commands outside normal operational bounds.

4.2.18 Secure Debugging

93. The platform only provides *<list of endpoints>* authenticated as specified in *<specification>* with debug functionality.

94. The platform ensures that all data stored by the application, with the exception of *<list of exceptions>*, is made unavailable.

95. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

96. **Note** 21: This security functional requirements shall be included if secure debugging is supported.

4.2.19 Perception Model

97. The platform provides ~~the~~ *integrity and confidentiality protection of the DCS perception model data and run-time configuration against machine learning model attacks* according to *<specification(s)>*.

98. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

99. **Note** 22: This SFR corresponds to an instantiation of the "Generic Security Platform Feature" SFR from the SESIP catalogue.

4.2.20 Privacy

100. The platform provides ~~the~~ *user protection services against discovery and misuse of identity and personally identifiable information by others and deploy such services* according to *prevailing legislation and the operational policy of the DCS*.

101. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

102. **Note** 23: It is recommended that the DCS provide support for a user to be able to use basic resources and services without necessarily disclosing the user's identity. The requirements for anonymity protect the user's identity.

103. **Note** 24: It is recommended that the DCS ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used.

104. **Note** 25: This SFR corresponds to an instantiation of the "Generic Security Platform Feature" SFR from the SESIP catalogue.

4.3 Optional Security Functional Requirements

4.3.1 Secure Confidential Storage

105. The platform ensures that all data stored by the application, except for *<list of data stored>*, is protected to ensure its confidentiality, integrity, authenticity, and binding to the platform instance.

106. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

4.3.2 Secure Encrypted Storage

107. The platform ensures that all user data stored, except for *<list of data stored in plaintext>*, is encrypted as specified in *<specification>* with a platform instance unique key of key length *<key length>*.

108. *<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

4.3.3 Secure Data Serialization

109. The platform ensures that all data stored outside the direct control of the platform, except for *<list of data stored outside the direct control of the platform>*, is protected such that the *confidentiality, integrity, binding to the platform instance* is ensured.

110. ***<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>***

5 Mapping and Sufficiency Rationales

5.1 SESIP3 Sufficiency

111. SESIP3 deliverables add basic documentation required to perform a white-box evaluation, as well as basic evidence of the use of configuration management.

Assurance Class	Assurance Family	Covered by	Rationale
ASE: Security Target evaluation	ASE_INT.1 ST Introduction	<Section “Introduction” and title page of the Security Target>	<TBD>
	ASE_OBJ.1 Security requirements for the operational environment	<Section “Security Objectives for the Operational Environment” of the Security Target>	<TBD>
	ASE_REQ.3 Listed Security requirements	<Section “Security Requirements and Implementation” of the Security Target>	<TBD>
	ASE_TSS.1 TOE Summary Specification	<Section “Security Requirements and Implementation” of the Security Target>	<TBD>
ADV: Development	ADV_FSP.4 Complete functional specification	<Description of which developer evidence is used to meet this requirement>	<TBD>
	ADV_IMP.3 Complete mapping of the implementation representation of the TSF to the SFRs	<Description of which developer evidence is used to meet this requirement>	<TBD>
AGD: Guidance documents	AGD_OPE.1 Operational user guidance	<Description of which developer evidence is used to meet this requirement>	<TBD>
	AGD_PRE.1 Preparative procedures	<Description of which developer evidence is used to meet this requirement>	<TBD>
ALC: Life-cycle support	ALC_CMC.1 Labelling of the TOE	<Description of which developer evidence is used to meet this requirement>	<TBD>
	ALC_CMS.1 TOE CM Coverage	<Description of which developer evidence is used to meet this requirement>	<TBD>
	ALC_FLR.2 Flaw reporting procedures	<ALC_FLR section in the Security Target and description of which	<TBD>

		developer evidence is used to meet this requirement>	
ATE: Tests	ATE_IND.1 Independent testing: conformance	<Description of which developer evidence is used to meet this requirement>	<TBD>
AVA: Vulnerability Assessment	AVA_VAN.3 Focused Vulnerability analysis	N.A. A vulnerability analysis is performed by the platform evaluator to ascertain the presence of potential vulnerabilities.	<TBD>

Appendix A Mapping with PSA Certified

112. This appendix provides a mapping between the Security Requirements of PSA Certified SESIP Profiles for PSA-RoT this Profile. Applicable PSA Certified levels can be Level 2+SE [PSAL2], Level 3 [PSAL3], Level 3 iSE/SE [PSAL3iSE] or Level 4 iSE/SE [PSAL4iSE]. PSA Certified L2 is not considered as it does not provide Physical Attacker Resistance SFR.

113. The ADAS Security Requirements for which the following table provides a “Same” mapping for PSA Certified Level 3 SESIP Profile SFR are already part of the certified PSA-RoT platform. The Security Requirements with an “Optional” mapping are part of the certified PSA-RoT platform only if they have been included in the Security Target for the considered PSA-RoT. Requirements marked “Not in PSA Certified” should be met by the developer in the features of the complete system design.

ADAS Profile SFR	PSA Certified SESIP Profiles SFR
Verification of Platform Identity	Same
Verification of Platform Instance Identity	Same
Attestation of Platform Genuineness	Same
Attestation of Platform State	Not in PSA Certified
Secure Update of Platform	Same
Secure Storage	Same
Secure Initialization of Platform	Same
Residual Information Purging	Not in PSA Certified
Secure Communication Support	Same
Audit Log Generation and Storage	Optional
Software Attacker Resistance: Isolation of Platform	Software Attacker Resistance: Isolation of Platform (between SPE and NSPE); or Software Attacker Resistance: Isolation of Platform (between PSA-RoT and Application Root of Trust Services)
Cryptographic Operation	Same
Cryptographic Random Number Generation	Same
Cryptographic Key Generation	Same
Cryptographic KeyStore	Same
Authenticated Access Control	Not in PSA Certified
Physical Attacker Resistance	Same

Secure Debugging	Optional
Perception Model	Not in PSA Certified
Privacy	Not in PSA Certified

Appendix B Mapping with AVCC Baseline Security Requirements

114. This appendix provides with supported ADAS Baseline Security Requirements [AVCC].

[AVCC] Baseline Security Requirements	This SESIP Profile SFR
6.1.1 Authorized software only	Secure Initialization of Platform
6.1.2 Unique Identifiability	Verification of Platform Identity Verification of Platform Instance Identity
6.1.3 A security lifecycle	Attestation of Platform State
6.1.4 Attestation	Attestation of Platform Genuineness Attestation of Platform State
6.1.5 Secure update	Secure Update of Platform
6.1.6 Isolation	Software Attacker Resistance: Isolation of Platform Parts
6.1.7 Unique binding of sensitive data	Secure Storage
6.1.8 Cryptographic Support	Cryptographic Operation
6.1.9 Certification	SESIP
6.2 Cryptography	Cryptographic KeyStore
6.3 Interface	Secure Communication Support
6.4 Platform	Physical Attacker Resistance
6.5 Software Update	Secure Update of Platform
6.6 Memory	Secure Storage Residual Information Purging
6.7 Perception Model	Generic Security Platform Feature
6.8 Security Audit	Audit Log Generation and Storage
6.9 Privacy	Generic Security Platform Feature

Acknowledgements

This document was written for Arm by ProvenRun
http://www.provenrun.com