Joint Interpretation Library

Secure Sub-System in System-on-Chip (3S in SoC)
– Life-cycle model related evaluation aspects –

Version 1.0

October 2024

This page is intentionally left blank.

Table of contents

1 Bibliography

[GC3S]	Guidance for Vulnerability Analysis and Penetration Testing of a Secure Sub-System within a System-on-Chip
[PP0084]	Security IC Platform Protection Profile with Augmentation Package, Version 1.0, Eurosmart, registered and certified by Federal Office for Information Security (BSI) under the reference BSI-CC-PP-0084-2014
[PP0117]	Secure Sub-System in System-on-Chip (3S in SoC) Protection Profile, Version 1.5, Eurosmart, registered and certified by Federal Office for Information Security (BSI) under the reference BSI-CC-PP-0117-2022 Secure Sub-System in System-on-Chip (3S in SoC) Protection Profile, Version 1.8, Eurosmart, registered and certified by Federal Office for Information Security (BSI) under the reference BSI-CC-PP-0117-V2-2023
[ADV_ARC]	Joint Interpretation Library – Security Architecture requirements (ADV_ARC) for smart cards, and similar devices extended to Secure Sub-Systems in SoC
[ADC_ARC_App]	Joint Interpretation Library – Security Architecture requirements (ADV_ARC) for smart cards and similar devices extended to Secure Sub-Systems in SoC – Appendix 1
[JIL-Comp]	Joint Interpretation Library – Composite product evaluation for Smart Cards and similar devices (for CC V3.1 R5) Joint Interpretation Library – Composite product evaluation and certification (for CC:2022)
[JIL-MSSR]	Joint Interpretation Library – Minimum Site Security Requirements
[ETRfI]	ETR for Integration template document defined by ISCI WG1 to enable re-use of evaluation results of a 3S in SoC
[CC/CEM:2022]	CCRA CC/CEM Version 2022 (Release 1)

2 Abbreviations

ASIC	Application-Specific Integrated Circuit
DDR	Double Data Rate
ETR	Evaluation Technical Report
ETRfI	ETR for Integration
FW	Firmware
FPGA	Field-Programmable Gate Array
HW	Hardware
IP	Intellectual Property
NVM	Non-Volatile Memory
OEM	Original Equipment Manufacturer
PCB	Printed Circuit Board
PL	Programmable Logic
PP	Protection Profile
RAM	Random Access Memory
ROM	Read Only Memory
RoT	Root of Trust
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SoC	System-on-Chip
ST	Security Target
SW	Software
3S	Secure Sub-System
TOE	Target of Evaluation
TSF	TOE Security Functionality

For further abbreviations refer to the Common Criteria.

3 Introduction

The 3S in SoC Protection Profile [PP0117] provides a generic life-cycle model which is pretty different from the well-known life-cycle model of [PP0084]. That's why an accompanying ALC guidance document is intended to supplement explanatory information on this life-cycle model and to support life-cycle model related evaluation aspects.

Hereby, specific development and production scenarios will be focused on, more detailed in dependency of the respective involved parties, memory type and 3S macro type. Scenarios whereby the 3S and SoC developers are the same, or whether external memory is used, or whether the 3S is a hard macro or a Programmable Logic macro or a combination of both have been examined.

This methodology can be applied for [PP0084] evaluations as well. Dedicated SAR aspects concerning the life-cycle model outlined in [PP0117] are addressed within the present document, but are transferable to the life-cycle model depicted in [PP0084].

4 Scope

Figure 1 represents the TOE scope as defined in [PP0117].

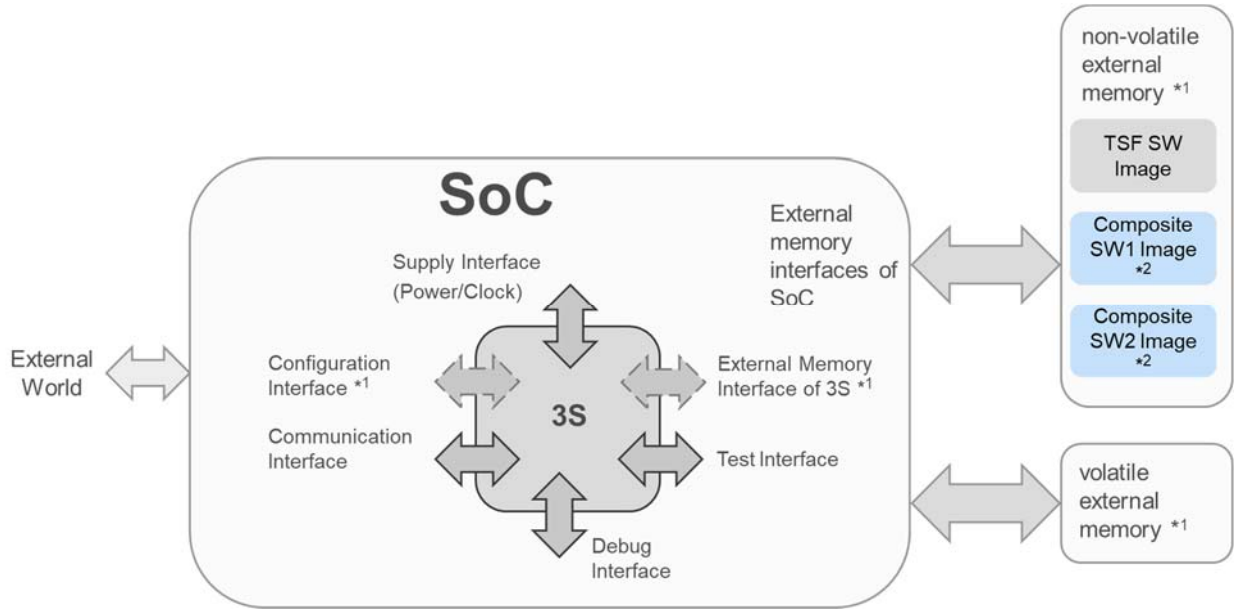


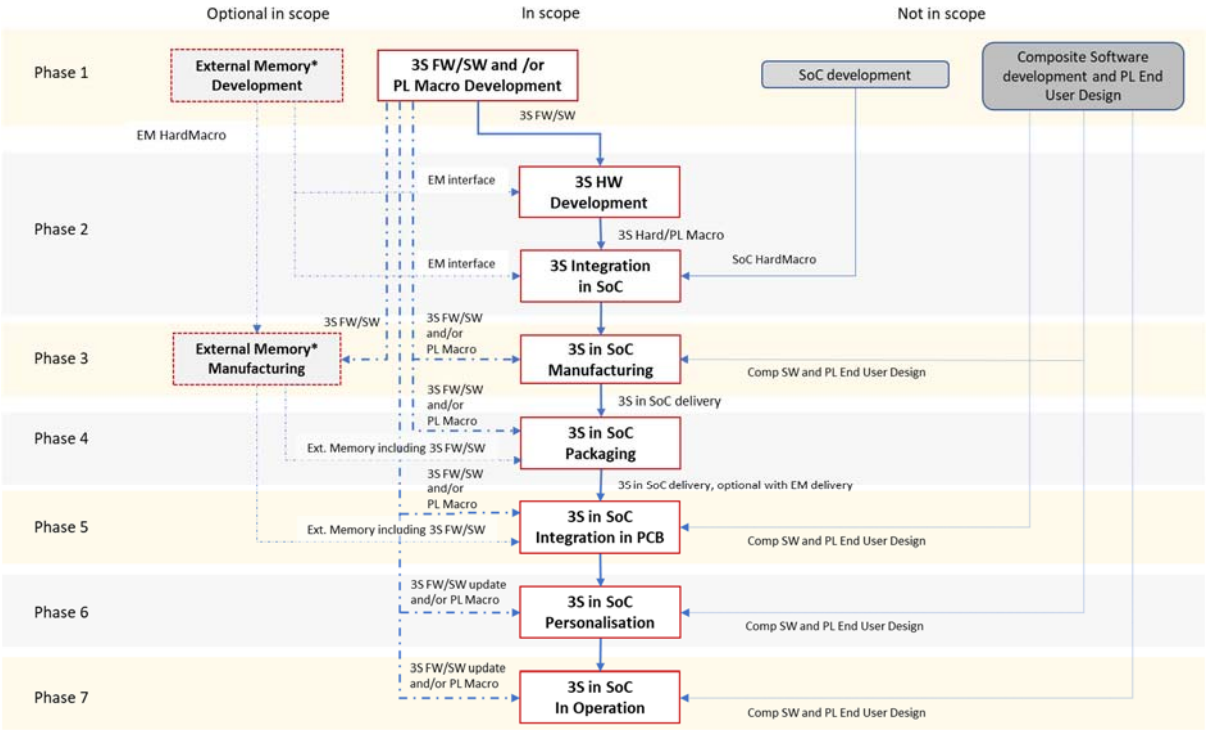
Figure 1 – Interfaces and components of an SoC containing a 3S

*1 3S interfaces marked with dashed outlines and the use of the external memories by the 3S are optional, depending on the implementation and configuration of the TOE.

*2 Composite Software Images do not belong to the TOE.

5 Generic life-cycle model

Figure 2 represents the generic life-cycle model for the 3S in SoC as provided in section 1.2.5 of [PP0117]:



* Secure External Memory is in the evaluation scope.

Figure 2 – Generic life-cycle model as provided in the 3S in SoC PP [PP0117]

For a corresponding description of the generic life-cycle model addressing its different phases, steps, activities and involved roles refer to [PP0117], section 1.2.5.

Several roles are defined in this life-cycle model:

Role	Description
3S Developer	Developer of the 3S (Phases 1 and 2)
SoC Developer	Developer of the whole SoC (not in scope)
3S Integrator	Integrator of the 3S in the SoC (Phase 2)
SoC Manufacturer	Manufacturer of the SoC (Phases 3 to 6)

These roles may be played by the same party (e.g., the 3S Integrator is generally the SoC Developer).

6 Recommendations related to 3S in SoC ALC evaluation

On top of the CC requirements and MSSR [JIL-MSSR], for the ALC evaluation of the 3S in SoC development and manufacturing phases, it is recommended to pay attention to the items described in the following sub-sections based on the TOE life-cycle model. Please note that Phase 4 and Phase 5 might be not relevant in case the TOE security does not rely on the packaging or PCB phases.

6.1 Phase 1 – 3S FW/SW Development

- For a PL Macro TOE, the development of the PL Macro and its corresponding FW/SW if it exists, is separate from the programmable activity developed by the SoC integrator in Phase 2 as described in section 8.4 “Scenario 4 – Support of PL Macro”.

6.2 Phase 2 – 3S HW Development and 3S Integration in SoC

- The 3S hardware is delivered with the expected security measures described by the 3S developer to the attention of the 3S integrator in the Integration Guidance (AGD) document.
- How the requirements of the Integration Guidance are covered in the specific integration of the 3S shall be part of the ADV_ARC documentation as defined in [ADV_ARC] and [ADC_ARC_App].
- The 3S integrator shall guarantee the level of security, which is required during the integration, by following the guidance outlined in the Integration Guidance.
- The case where the 3S developer and SoC developer are the same is examined in section 8.1 “Scenario 1 – 3S developer and SoC developer are the same”.
- The case where the 3S developer is different from the SoC developer is examined in section 8.2 “Scenario 2 – 3S developer and SoC developer are different”.
- The PL Macro case is examined in section 8.4 “Scenario 4 – Support of PL Macro”. Its specificities must be described clearly in the Security Target as required in the related Application Notes of [PP0117].
- The Root of Trust component and associated keys are critical assets and are introduced in the TOE Description of [PP0117] and the Assets definition of [PP0117]. SFRs associated with the Root of Trust are described in the Security Functional Requirements for the TOE and the Composite Software Isolation Package of [PP0117].

6.3 Phase 3 – 3S in SoC Manufacturing

- FW/SW loading is described in the Loader Package of [PP0117].
- Pay attention to the test coverage of the 3S HW: tests of the 3S HW TSF including 3S tests ordering dependencies with SoC tests, and check that the HW self-protection of the 3S is properly enabled (concerning the HW self-protection features to be enabled during 3S in SoC Manufacturing). Refer to the ATE_COV and AVA_VAN refinements in [PP0117].

- Pay attention to the self-protection of the 3S being properly enabled (HW and FW/SW) after the TOE delivery point (guidance).

6.4 Phase 4 – 3S in SoC Packaging

- The 3S in SoC Packaging may introduce additional protection measures as described in the Security Objectives for the Operational Environment of the packaging in [PP0117].
- External memories may be used during this phase as described in the Package for Passive External Memory of [PP0117] and the Package for Secure External Memory of [PP0117]. For instance, external DDR may be packaged together with the SoC.
- Pay attention to the test coverage of the 3S HW: tests of the 3S HW TSF including 3S tests ordering dependencies with SoC tests, and check that the HW self-protection of the 3S is properly enabled (concerning the HW self-protection features to be enabled during 3S in SoC Packaging). Refer to the ATE_COV and AVA_VAN refinements in [PP0117].
- If the TOE relies on packaging for its protection (i.e. packaging covers / addresses SFRs depicted in the ST), the corresponding packaging process needs to be secure and certified since this is a crucial part of manufacturing the complete secure TOE. In this case the TOE delivery point in the sense of the CC shall be after packaging. Otherwise, if the package process requirements consist in logistics features and preparative procedures only (i.e. without any security impact on the TOE), then the packaging process can be covered by guidance, and the delivery point may lie before the packaging.
- Pay attention to the self-protection of the 3S being properly enabled (HW & FW/SW) after the TOE delivery point (guidance).

6.5 Phase 5 – 3S in SoC Integration in PCB

- External memories may be used during this phase as described in the Package for Passive External Memory of [PP0117] and the Package for Secure External Memory of [PP0117]. For instance, external flash may be placed on the PCB.
- Pay attention to the self-protection of the 3S being properly enabled (HW & FW/SW) after the TOE delivery point (guidance).

7 Development and production security

ALC_CMC.4.8C (for augmentations ALC_CMC.5.14C) and ALC_DVS.2 cover the internal transport of security relevant TOE assets and the necessary acceptance procedures.

Internal transfer concerns the transfer of TOE assets (e.g., confidential source code, hard macros, specifications, etc., similar to the definitions of [PP0084]) that occurs within the development and/or production chain up to the delivery of the TOE, refer to ALC_DEL.

Internal transfer of confidential assets shall only occur within and/or between MSSR audited environments (cf. [JIL-MSSR]).

Furthermore, the Application Notes 14 and 15 of the 3S in SoC Protection Profile [PP0117] shall be considered:

> *The SoC including the 3S can only be considered as delivery item at the end of Phase 3 [or Phase 4], if the trimming, initialisation and personalisation are completed and the self-protection of the 3S is completely enabled. The evaluation shall include all manufacturing steps, which require protection by the environment.*

In consequence, the 3S in SoC can be considered as a *conforming product* and a delivery item only if it is fully functional, trimmed and initialized and all its self-protective features are active. This can be in any phase starting from Phase 3.

Only if a 3S in SoC is a conforming product it can be assumed that the protective features promised in the individual ST are available; this concerns e.g., cryptographic functionality, deactivation of functionality, etc.

Vice versa, in the sense of [PP0117] and its Application Notes 14 and 15, any other form, e.g., defective items, untested items, items operating outside of the specified parameters, shall be considered as a *non-conforming product* and shall not be regarded as a delivery item. For such items it has to be assumed that they are not suited to protect themselves or assets stored. E.g., on a complete wafer after testing both will be present, fully functional and trimmed 3S in SoC; if any of those items failed the testing, this complete wafer shall not be considered as a delivery item.

Any asset that requires the protection of the environment shall be handled only at adequate audited sites (cf. [JIL-MSSR]). This includes assets associated with non- or not yet conforming products, but can include additional assets, if the personalization process or related assets require protection from the environment. This even applies if the 3S in SoC is already a conforming product.

8 Specific development and production variants of the generic life-cycle model

The life-cycle model presented in section 5 “Generic life-cycle model” is a generic model that applies to all scenarios of responsibilities and configurations of the TOE that may be chosen in Security Targets (ST). Depending on who takes the role of the developer of the 3S and SoC, different scenarios of responsibility and configuration may occur. The following options regards role, type of memory and type of macro are available in principle:

- role distinction (3S developer = SoC developer or 3S developer ≠ SoC developer)
- type of volatile memory (internal or external/passive or external/secure)
- type of non-volatile memory (internal or external/passive or external/secure)
- type of 3S macro (hardmacro and/or PL Macro)

To illustrate the possible combinations of the aforementioned options, the following table provides exemplarily some use cases that are of major interest in practice:

Config / Scenario		3S developer = SoC developer	3S developer ≠ SoC developer	Volatile Memory			Non-Volatile Memory			Macro Type
				Internal[1]	External		Internal	External		
					Passive	Secure		Passive	Secure	
1		X		X	X			X		hardmacro
2	**a**		X	X				X		hardmacro
	b		X	X	X			X		hardmacro
3		X		X	X				X	hardmacro
4		X		X	X			X		hardmacro and PL Macro

Table 1 – Development and production scenarios

The following scenarios and configurations are addressed in Table 1:

- **Scenario 1:** In this scenario, the 3S and the SoC are developed by the same vendor, and volatile and non-volatile memories are external/passive. The 3S is made up of a hardmacro.
- **Scenario 2:** In this scenario, the 3S is developed by an IP vendor who delivers a hard macro to the SoC developer. The non-volatile memory is external/passive and provided by the SoC developer.
    - **Variant 2a:** The volatile memory is internal to the 3S and part of the hardmacro.
    - **Variant 2b:** The volatile memory is external/passive and provided by the SoC developer.

[1] Independent of the product configuration, internal volatile memory is always existing and used by the product. Except for Scenario 2 with its Variant 2a this kind of memory is not considered in the following sub-sections that provide detailed descriptions of the different secenarios and configurations.

- **Scenario 3:** In this scenario, the 3S and the SoC are developed by the same vendor, and the volatile memory is external/passive, while the non-volatile memory is external/secure. The scenario has a strong overlap with Scenario 1.
- **Scenario 4:** This scenario is similar to Scenario 1, but with the following variation: The 3S is comprised of a PL Macro and a hardmacro including additional hard logic beyond the necessary parts for the PL Macro. For an example, please refer to section 8.4.

For each of the previously mentioned scenarios or configurations respectively, the following sections provide specific life-cycle diagrams depicting the configuration as well as descriptions of the TOE delivery. Please note that trimming is done in the same manner independently of the scenario used to deliver the 3S.

8.1 Scenario 1 – 3S developer and SoC developer are the same

In this scenario, the 3S and the SoC are developed by the same vendor, and volatile and non-volatile memories are external/passive.

The following diagram represents an instantiation of the generic life-cycle diagram for this specific configuration:

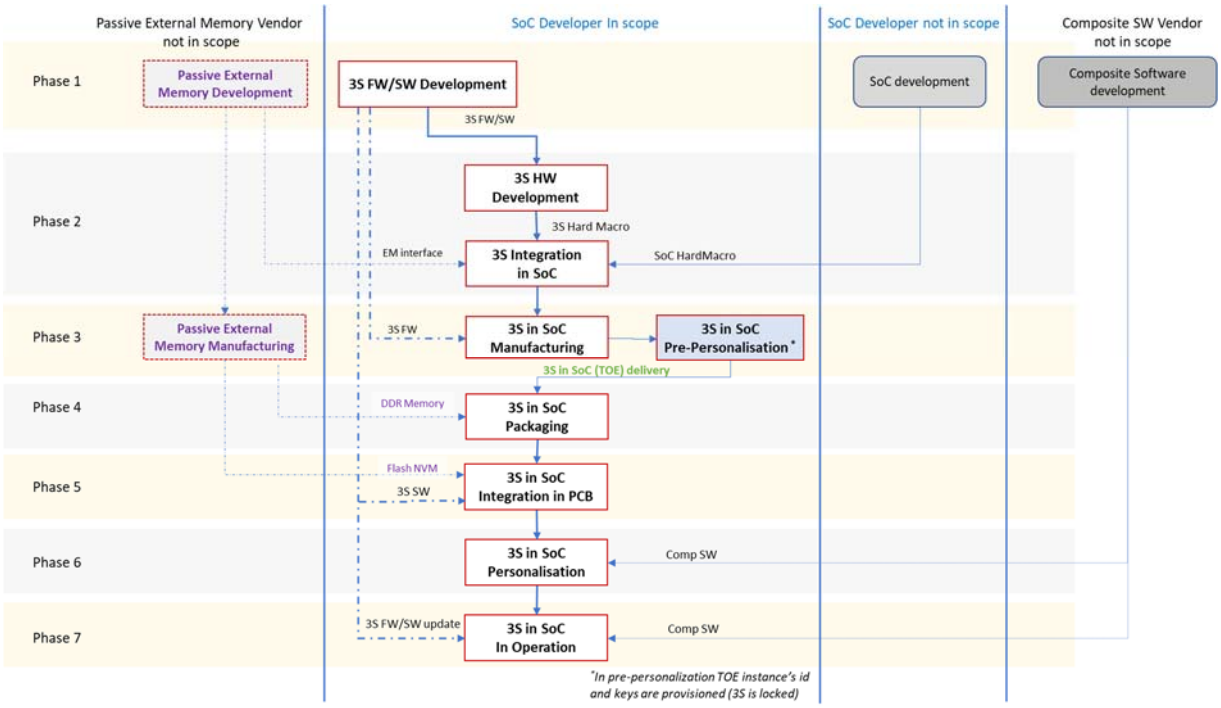


Figure 3 – Life-cycle diagram for Scenario 1

In this configuration the point of delivery is usually at the end of Phase 3, after pre-personalisation of the 3S is done by the SoC vendor in its manufacturing shop. During this pre-personalisation phase, the TOE instance ID and the TOE keys are provisioned.

As an alternative, the point of delivery may be after Phase 4 in the case where the external non-volatile memory is packaged together with the SoC (for instance in a package-on-package). This becomes necessary in the case where the packaging brings additional security protection (for instance through a metal shield), as described by the optional assumption `A.Packaging-Requirement` in [PP0117]. The external/passive NVM is added at a later stage in Phase 5, for instance by the OEM using this SoC.

In all cases, because they are external/passive, neither the volatile memory nor the non-volatile memory are included in the TOE.

The software image is stored in the external/passive non-volatile memory as an encrypted and signed blob, which can only be loaded and read by the 3S, as described in the Package for External Passive Memory of [PP0117].

In this configuration where the 3S and SoC are developed by the same vendor, the Integration Guidance shall have the same formalism as in case vendors are different to ensure adequate evaluation by the lab.

Depending upon the processes used by the developer in Phase 2, and depending on the types of documents exchanged between 3S team(s) and SoC team(s), the development sites will need to be audited.

All sites involved before and including TOE delivery are in evaluation scope, and TOE delivery can be after Phase 3 or 4 alternatively depending on how the TOE production process is set up and described.

The pre-personalisation happens in Phase 3 prior to TOE delivery, so the SoC manufacturer production site(s) need(s) to be audited. Therefore, no audit of the OEM manufacturing site is necessary because the 3S in SoC is locked after Phase 3 or 4 alternatively when self-protection of the 3S TOE is enabled, and as external non-volatile memory is not in the scope of the TOE.

In case the TOE does not need any additional security protection by its packaging, the assumption that you only need audits up to Phase 3 only holds true, if you only ship conforming products from Phase 3 to Phase 4 (i.e., all non-conforming products are scrapped before transfer to the next phase).

There are requirements about the tests specific to the 3S TOE to be done during the SoC wafer testing phase (production tests), refer to the test requirements outlined in [PP0117]. E.g., scan testing and functional tests covering the 3S TSFs within the wafer testing of the SoC; this is product dependent and has to be clarified by the developer.

Note: It is common for fielded systems to require updates in order to add capabilities, address vulnerabilities, etc. This might in particular be the case for 3S SW/FW updates as depicted for Phase 7 in Figure 3. Such 3S SW/FW updates require rigorous testing, validation and evaluation/certification before being deployed to confirm that (a) the functionality being updated is correct and sufficiently secure, (b) the updated functionality is correctly and securely loaded/installed, and (c) the update does not negatively impact interfaces or other functionality already deployed. The verification, validation and evaluation/certification of 3S SW/FW updates, prior to their deployment, are required to avoid a situation where the update creates a vulnerability.

8.2 Scenario 2 – 3S developer and SoC developer are different

In this scenario, the 3S and the SoC are developed by different entities, and non-volatile memories are external/passive. Volatile memory is either internal (Variant 2a) or external/passive (Variant 2b). As often the case in practice, it is assumed that the SoC developer and the 3S integrator are the same.

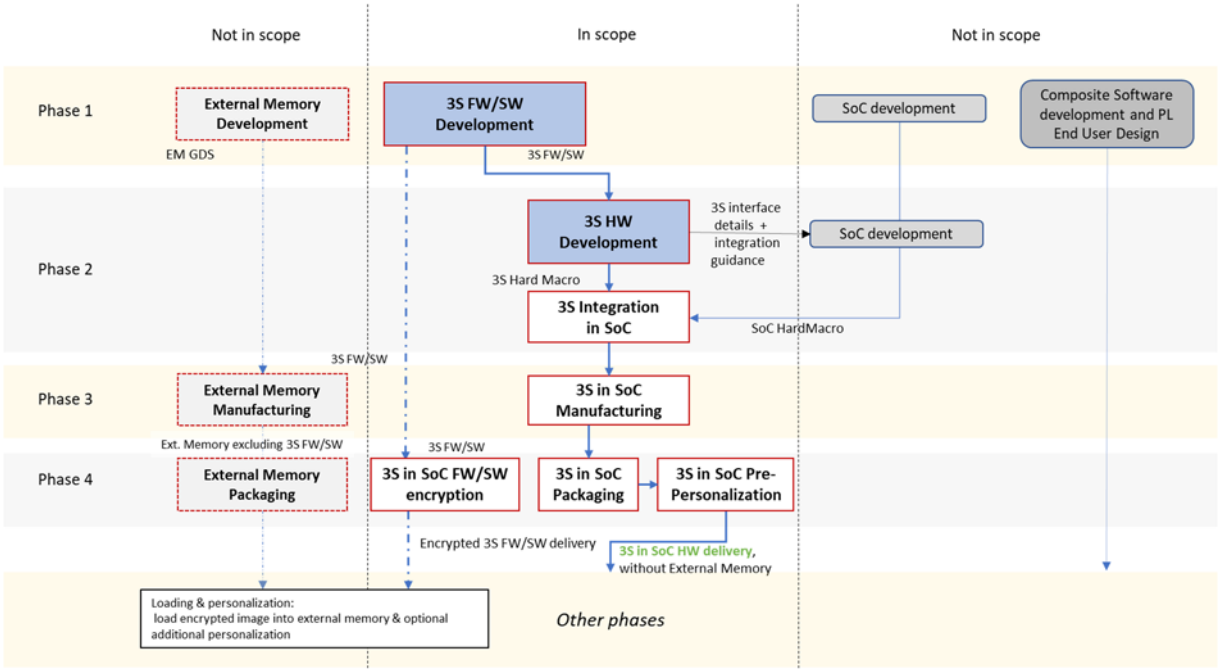


Figure 4 – Life-cycle diagram for Scenario 2

In the diagram, the separate entity who develops the 3S is indicated in blue. In Phase 1, the 3S firmware/software is developed. The configuration supports external non-volatile memory for storage of this FW/SW, so parts of it flow through to Phase 4 for preparation for secure use of the external/passive memory. Another part (e.g., boot ROM code) flows to Phase 2 for inclusion in the 3S hardmacro.

In Phase 2, the 3S hardware development takes place, resulting in a hardmacro to be integrated in the SoC. Besides the hardmacro, functional information about the 3S interfaces as well as the security related Integration Guidance are provided to the SoC development and integration entity. The non-3S parts of the SoC are not part of the TOE, so the SoC development itself is not in scope for the certification. The integration or combination of the 3S hardmacro and the SoC hardmacro (including preservation of confidentiality/integrity of the 3S within the SoC manufacturer premises/sites), however, is in scope. Attention should be paid during the life-cycle evaluation that the Integration Guidance is well documented, communicated, and followed by the SoC integrator. Since for this scenario the SoC development and integration are performed by a different party, an explicit arrow has been added to indicate this flow of information. The information must provide enough details to ensure that the counterparts of the 3S interfaces (environment) can be correctly implemented, but it should not reveal potentially confidential information about 3S implementation details. For subsequent integrations, the ETRfI will be used as a reference by subsequent evaluation labs to support adequate evaluation of integration aspects.

The life-cycle continues with the 3S in SoC Manufacturing (Phase 3) and Packaging (Phase 4). In this configuration example, the non-volatile external memory is not included in the package, but only later integrated (e.g., on the PCB by an OEM device manufacturer). In Phase 4, like in Scenario 1 during Phase 3, the packaged 3S in SoC is pre-personalized: self-protection mechanisms are enabled, and root keys provisioned. Note that this difference with Scenario 1 is not related to the different 3S and SoC developers, but it ensures wider coverage of potential

life-cycle scenarios that are all compliant with the Protection Profile. The end of Phase 4 marks the point of delivery. There can be dependencies between the pre-personalization and the protection mechanisms of the external/passive memory, so the diagram contains an explicit 3S FW/SW encryption step in Phase 4. The dependency in this case is that the root keys programmed into the 3S need to correspond with the keys used for the FW/SW encryption. Both the FW/SW and 3S in SoC hardware are now self-protected, so later phases of the life-cycle are not in scope of the certification.

The assumption that audits are only needed up to Phase 4 only holds true if only conforming products from Phase 4 to later phases are shipped (i.e., all non-conforming products are scrapped before transfer to the next phase). Furthermore, in case that security relevant keys are handled at the OEM device manufacturer the respective sites/IT systems need to be audited as well.

In this scenario, the 3S development and the 3S integration are performed by different entities, so site security for both entities should be evaluated. The SoC development activities themselves, however, can be left out of scope, as long as no confidential assets of the 3S in SoC have to be exchanged with the SoC developer.

8.3 Scenario 3 – Support of secure memory

For secure memory configurations, the 3S in SoC evaluation is a composite evaluation/certification (refer to [JIL-Comp] / [CC/CEM:2022]) including the already certified external/secure memory.

For this configuration, there is no need to distinguish whether the 3S and SoC developers are the same or different. If the developer is the same for 3S and SoC, the life-cycle used in section 8.1 “Scenario 1 – 3S developer and SoC developer are the same” applies with external/secure memory replacing external/passive memory. If the 3S and SoC developers are different, the life-cycle used in section 8.2 “Scenario 2 – 3S developer and SoC developer are different” (Variant 2b) applies with external/secure memory replacing external/passive memory.

8.4 Scenario 4 – Support of PL Macro

This section is intended to provide clarity on how a Programmable Logic Macro (PL Macro) could be designed and integrated as part of the 3S within an SoC. As shown in Table 1 in section 8 and in difference to Scenario 1, the 3S could be comprised of a hardmacro and a PL Macro (including additional hard logic beyond the necessary parts for the PL Macro).

According to [PP0117], section 1.2.2 the term “PL Macro” is specified as follows:

“The 3S is a physically-fixed design defined either as a hard macro (e.g., a GDSII file) and/or as a programmable logic (PL) macro (a bitstream used to configure a Field Programmable Gate Array (FPGA)). In any case, “physically-fixed design” means that the layout, placement, routing and timing are part of the implemented 3S, and that the HW implementation is predictable in terms of operational ranges such as performance, timing, area, and power.

For a PL Macro, the functionality that the PL Macro provides may be configurable; this configurability shall be independent of the PL Macro placement and routing such that the predictability of operational ranges is not affected."

The PL Macro is a logic, created by either the OEM or the end user, that is implemented in the Programmable Logic portion of the SoC and that implements itself a specific set of functionality that is part of the TOE. It is defined by a configuration file that when applied to the Programmable Logic hardware, implements the specific set of functionality intended. The configuration file consists of a string of "1"s and "0"s that define what logic and routing will be implemented when loaded into the Programmable Logic portion of the SoC. The term "bitstream" is often used as a specific name for a configuration file.

The specification of the term "PL Macro" in [PP0117], section 1.2.2 cited above is to be interpreted in that way that the PL Macro as TOE part consists of a configuration file AND the underlying hardware that is intended to be loaded with the configuration file. This means that the PL Macro does not only consist of SW or logical data, but covers as well the related physical representation.

In scenario 4, the TOE consists of the hardware that securely loads the PL Macro (sometimes referred to as the Root of Trust), additional hard logic within the SoC and the PL Macro itself. Furthermore, it is assumed that the 3S and the SoC are developed by the same vendor. This scenario will be the most common real-world use case and considered in more detail in the following.

A notional SoC that includes Programmable Logic (PL) is shown in Figure 5. Note that the PL is a "sea of gates" that performs no function when manufactured. The PL only provides functionality when it is programmed. Programming occurs when a configuration file is loaded during the boot process[2]. For programming, there is logic (that itself is not programmable) that performs the loading and programming function. The hardware that securely loads the PL Macro is a hardmacro in the sense of [PP0117]. It is a portion of the SoC that is completely separate from the Programmable Logic. Just like any other ASIC function, it takes the configuration file (bitstream), performs security operations on it (e.g. authentication and decryption) and then loads it into the Programmable Logic. In Figure 5, such logic module could be provided by the Platform Management Processing Subsystem. Once the Programmable Logic is loaded and configured, the PL Macro comes alive and implements the functionality intended.

[2] The boot process can provide Authenticity, Confidentiality and Integrity of the configuration files.

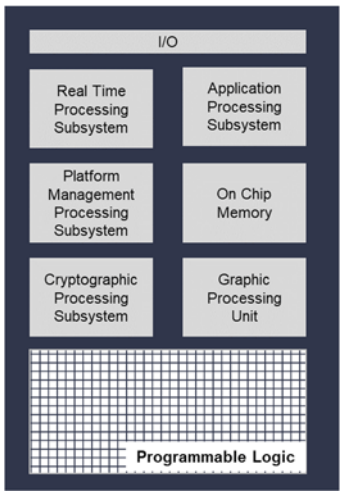


Figure 5 – Notional SoC with Programmable Logic

Regards the life-cycle, the production and evaluation/certification of the FPGA hardware which executes the Programmable Logic (PL) code in a secure manner (concerning the security objectives integrity, and confidentiality if applicable) is a separate process lying outside the scope of this section. However, in this scenario secure "installation" (upon start-up) of the PL configuration file (bitstream) is part of Phase 3 in Figure 8.

In this example scenario, the 3S is assumed to be designed by the same developer as of the SoC (as in section 8.1 “Scenario 1 – 3S developer and SoC developer are the same”), and not delivered as an IP from a 3rd party. Therefore the delivery point is the same as in Scenario 1. It is also assumed that the 3S encompasses the Platform Management Processing Subsystem and Cryptographic Processing Subsystem (each realised as hardmacro) which is shown in Figure 6.

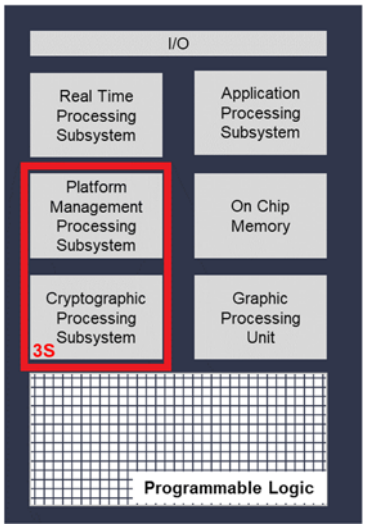


Figure 6 – Notional SoC with 3S

The SoC developer desires to develop cryptographic functionality that shall be part of the 3S in a portion of the PL (e.g., a quantum resistant algorithm that has yet to be ratified by the industry). The SoC developer follows the Programmable Logic design flow to develop the PL Macro. When the PL is programmed with the PL Macro, the system would look like Figure 7.

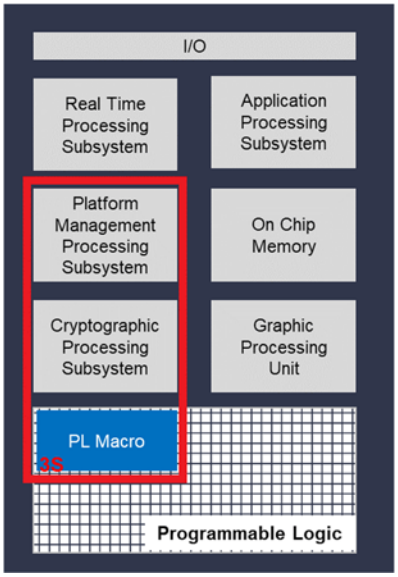


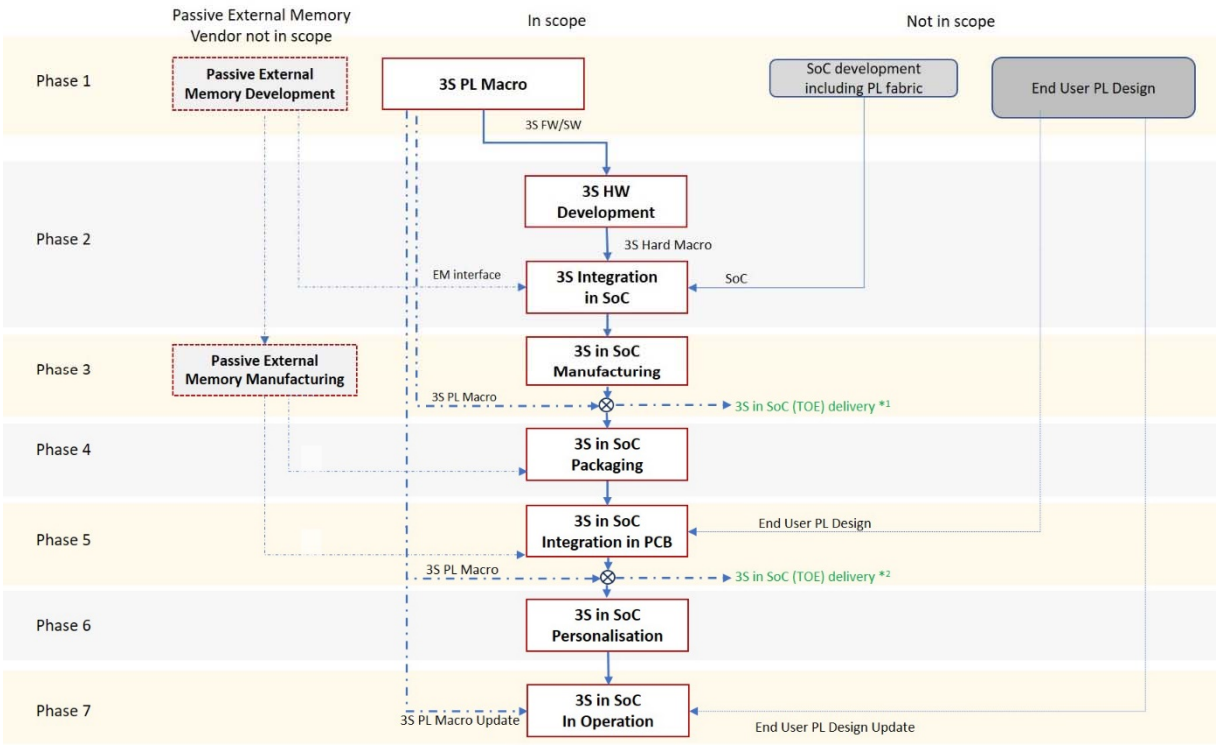
Figure 7 – Notional SoC with 3S and with PL Macro for assessment

In Figure 7, as an example, the TOE includes:

- Platform Management Processing Subsystem which controls the boot and loading of the Programmable Logic.
- Cryptographic Processing Subsystem which provides cryptographic services for secure boot: providing authenticity, integrity and confidentiality for the SW loaded into the SoC, and also for the configuration file (bitstream) loaded into the Programmable Logic (i.e., the PL Macro). The Cryptographic Processing Subsystem would also provide cryptographic services (e.g., key management, cryptographic acceleration) to the SoC post boot.
- PL Macro with its logic programmed in the Programmable Logic that implements security functionality which augments the Cryptographic Processing Subsystem.

This is the system that the SoC developer would get evaluated by a lab (i.e., the TOE being the 3S made of the Platform Management Processing Subsystem hardmacro, the Cryptographic Processing Subsystem hardmacro and the PL Macro in the Programmable Logic (PL)). Please note that it is important that the logic that is in charge of performing the programming lies in the scope of the evaluation. The lab would get the 3S in SoC on the hardware platform provided for evaluation together with the PL Macro for testing and vulnerability analysis. As part of the evaluation, the SoC developer is required to demonstrate the properties of isolation and access control between the 3S and the rest of the SoC.

Figure 8 represents the life-cycle of the 3S in SoC in case a PL Macro in Scenario 4 is involved.



[1] including 3S PL Macro
[2] including combination of 3S PL Macro and End User PL Design

Figure 8 – Life-cycle diagram for Scenario 4

Upon successful evaluation/certification, the configuration information contained in the PL Macro is captured from the implementation so it can be delivered to the end user. Note that the configuration file that programs the Programmable Logic (PL) programs the entire PL region. However, since the PL Macro makes up only a portion of the PL, only the configuration information that defines the PL Macro is captured.

The end user will create his application in the Programmable Logic, and using the SoC developer's tools, integrate the provided PL Macro into the end user's programming file. The integration process shall result in a programming file with a PL Macro that operates identically as when it was evaluated/certified (e.g., same logic placement, same routing, same timing, etc.). The integrated PL Macro and End-User Design combination must be evaluated to ensure this is the case. This evaluation for example should include the correct use of the PL integration tools, and the logic integration of the 3S and End-User Design.

Note: It is common for fielded systems to require updates in order to add capabilities, address vulnerabilities, etc. This might in particular be the case for PL Macro updates or End User PL Design updates as depicted for Phase 7 in Figure 8. Similar to how software updates can be delivered and installed securely and have to undergo verification, validation and evaluation/certification (refer to the Note at the end of section 8.1), such aspects and issues are applicable for updates to the Programmable Logic and loaded programming files as well.

Different cases for such updates can occur, e.g. the PL Macro is updated, but no change of the rest of the End User Design in the Programmable Logic is performed, or conversely, the End User Design in the Programmable Logic is updated whereby the PL Macro remains unchanged. Regards evaluation/certification this means:

If the PL Macro is updated whereby the rest of the End User Design is unchanged, the updated PL Macro has to be re-evaluated/re-certified and the impact of the PL Macro changes on the End User Design in the Programmable Logic integrating that updated PL Macro has to be analyzed and assessed. If the End User Design is updated, but the PL Macro remains unchanged it has to be demonstrated that the End User PL Design update does not negatively impact the system.

9 Subsequent evaluations of 3S in SoC

Once a 3S in SoC has been evaluated/certified according to [PP0117], the same 3S can be transferred to other SoCs for which the developer could then either make re-use of the previous certification or start independently a complete new certification. Any consecutive 3S in SoC evaluations/certifications are intended to be supported by re-use of the assurance results from the first evaluation/certification as far as possible. However, re-use is subject to examination during the first 3S in SoC evaluation procedure where especially possible re-use scenarios are evaluated/examined.

The assurance that can be re-used is outlined in the so-called ETR for Integration [ETRfI]. Being an output document of the first evaluation/certification of the 3S in SoC, it is exchanged in subsequent evaluations/certifications among ITSEFs and with the Certification Bodies in charge to support such re-use. The ETR for Integration is derived from the full ETR and identifies potential vulnerabilities and test results in the light of dependencies on the SoC (if any). As a result, the ETR for Integration presents only the relevant subset of potential vulnerabilities and test results.

A template of an ETR for Integration has been developed by JHAS / ISCI WG1, which addresses the following topics:

- design
- integration guidance
- physical dependency of the 3S on the hosting SoC
- relationship between the 3S and its hosting SoC
- usage of external memory
- 3S configuration and identification
- life-cycle
- functional testing
- penetration testing
- observations and recommendations
- technical rationale regarding conditions applicable for re-use