ETSI TS 104 007 V1.1.1 (2024-11)

Lawful Interception (LI);
Lawful Interception Architecture

Reference
DTS/LI-00241

Keywords
functional architecture; interception;
lawful interception

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Lawful Interception (LI).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The present document describes a comprehensive blueprint for a Lawful Interception (LI) system, designed to align with the mandates of Law Enforcement Agencies (LEAs) for the surveillance of telecommunications with utility across varying legal and regulatory environments. The LI architecture detailed herein sets out to meet stringent requirements, outlining the necessary high-level entities, procedures, and the functional roles essential for the operation of interception. It specifies how lawful authorizations, such as warrants, are processed and executed by Communication Service Providers (CSPs) to deliver Intercept Related Information (IRI) and Content of Communication (CC) to LEAs through secure channels.

The present document emphasizes a security-centric approach, incorporating Zero Trust principles across the architecture to ensure secure operations. This security view is intertwined with the functional aspects, ensuring that both data integrity and privacy are preserved throughout the interception process. The architecture is designed to enable distinct LEAs to carry out interception activities simultaneously and independently, safeguarding against the risk of inter-agency or other non-authorized detection.

Through the definition of a methodical LI lifecycle, starting from authorization and ending with the delivery of intercepted communications, the present document covers the operational protocols, the interplay between the various entities, and the safeguarding mechanisms in place. The described LI system is adaptable to the dynamic and evolving landscape of national laws and telecommunication technologies, ensuring future-proof applications across different jurisdictions and technological paradigms.

Introduction

The present document explains thoroughly the intricate architecture of Lawful Interception (LI) systems as stipulated across varying regulatory environments, a cornerstone for maintaining the delicate balance between national security, law enforcement, and individual privacy. Mapped out herein are the protocols that govern the interception of telecommunications as per the legal frameworks established by ETSI TS 101 331 [1]. Commencing with a broad overview of the key players and their respective roles within the LI ecosystem, the present document progressively narrows down to a more detailed analysis of the functional and security viewpoints.

1 Scope

The present document on Lawful Interception architecture provides an overview of the technical framework and components involved in facilitating lawful interception of communications for law enforcement agencies. The present document outlines the key principles, standards, and protocols governing lawful interception, including the roles and responsibilities.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 101 331: "Lawful Interception (LI); Requirements of Law Enforcement Agencies".

[2] European Union Council Resolution 96/C 329/01 of 17 January 1995 on the lawful interception of telecommunications.

[3] ETSI GS NFV-IFA 026: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Architecture enhancement for Security Management Specification".

[4] ETSI TS 104 000: "Lawful Interception (LI); Internal Network Interface X0".

[5] ETSI TS 133 126: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Lawful Interception requirements (3GPP TS 33.126)".

[6] ETSI TS 133 127: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Lawful Interception (LI) architecture and functions (3GPP TS 33.127)".

[7] IETF RFC 9334: "Remote ATtestation procedureS (RATS) Architecture".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI ETR 330: "Security Techniques Advisory Group (STAG); A guide to legislative and regulatory environment".

[i.2] ETSI GR NFV-IFA 029 (V3.3.1): "Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS"".

[i.3] ETSI GS NFV-IFA 040 (V4.1.1): "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification".

[i.4] NIST Special Publication (NIST SP) 800-207: "Zero Trust Architecture".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

root of trust: hardware-based seed/key material or function that contains it, upon which a hierarchy of keys are built to support higher functions

trust anchor: root certificate authority in the network

zero trust: security concept where no entity, whether inside or outside a network perimeter, is automatically trusted, but granularly authorized

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ABAC	Attribute Based Access Control
ADMF	ADMinistrative Function
API	Application Programming Interface
ARP	Attestation Relying Party
AVS	Attestation Verifier Service
CA	Certificate Authority
CC	Content of Communication
CISM	Container Infrastructure Service Manager
CMF	Certificate Management Function
CPU	Central Processing Unit
CSP	Communication Service Provider
CSR	Certificate Signing Request
CTDGW	Cross Trust Domain GateWay
DMZ	De-Militarized Zone
ELI	Element of LI
HMEE	Hardware Mediated Execution Enclave
ID	IDentifier
IRI	Intercept Related Information
LEA	Law Enforcement Agency
LEMF	Law Enforcement Monitoring Function/Facility
LI	Lawful Interception
LI-Admf	Lawful Interception-ADMinistrative Function interface
LI-Ap	Lawful Interception-Application interface
LI-No	Lawful Interception-Network Output interface
LI-Os	Lawful Interception-Operations support interface
LI-Vn	Lawful Interception-Virtual network interface

LICA	Lawful Interception Certificate Authority
LICF	Lawful Interception Control Function
LICM	Lawful Interception Certificate Management function
LIGW	Lawful Interception GateWay
LIID	Lawful Interception IDentifier
LIPF	Lawful Interception Provisioning Function
LISE	LI Security Engine
LRPG	LI Routing Proxy Gateway
MANO	MANagment and Orchestration
MDF	Mediation and Delivery Function
MOGW	ManO GateWay
NE-Admf	Network-Administrative function interface
NF	Network Function
NFIID	Network Function Instance IDentity
NFV	Network Function Virtualization
NFVO	Network Function Virtualization Orchestrator
NOGW	Network Output GateWay
NRF	NF Repository Function
NWF	NetWork functionality Function
OS	Operating System
Os-Mano	Operations support-MANO interface
Oss-LI	Operations support-Lawful Interception interface
OSS/BSS	Operations Support System/Business Support System
POI	Point Of Interception
RAM	Random Access Memory
RATS	Remote ATtestation procedureS
SDN	Software Defined Network
SOC	System On Chip
SIRF	System Information Retrieval Function
TD	Trust Domain
TF	Triggering Function
TLS	Transport Layer Security
TPM	Trusted Program Module
UEFI	Universal Extensible Firmware Interface
VM	Virtual Machine
VNF	Virtual Network Function
ZTA	Zero Trust Architecture

4 Approach

4.1 General approach

The present document defines the Lawful Interception (LI) architecture to meet the requirements of LEAs regarding the Handover Interface for the interception of telecommunications (see ETSI TS 101 331 [1]).

Clause 4.2 describes a high-level view of the entities and procedures that are generally required to be supported in LI systems.

Clause 5 sets out the functional view of the architecture that supports the elements and procedures required by clause 4.2.

Clause 6 sets out the security view of the architecture that supports secure operation following Zero Trust principles as defined by NIST [i.4].

Clauses 5 and 6 are to be read together, as the security view in clause 6 motivates the functional view of clause 5.

Clause 7 gives an overview of the provisioning process.

4.2 LI entities and procedures

4.2.1 Overview

The functional role model described in this clause is a reference example to facilitate a general understanding of the typical operation of interception and the typical responsibilities of the various elements.

National laws that describe the conditions and restrictions of interception and procedures will apply as described in references [2] and [i.1].

The LEA obtains a lawful authorization, such as a warrant, from a court of law or other responsible body (the "authority" in figure 4.2.1-1). The LEA presents the lawful authorization to the CSP via an administrative interface or procedure (interface port HI1).

Intercept Related Information (IRI) and the Content of Communication (CC) are delivered to the Law Enforcement Monitoring Function (LEMF) of the requesting LEA, via interfaces HI2 and HI3, respectively.

A lawful authorization may describe the interception target, the interception period, and the IRI and the CC that are allowed to be delivered for this LEA. For different authorizations, different constraints may apply that further limit the general restrictions set by the law. The interception target may also be described in different ways in a lawful authorization (e.g. subscriber address, physical address, services, etc.).

A target may be the subject of interception of different authorizations. It is necessary to support strict separation of these lawful interceptions. It is therefore possible that more than one lawful authorization may be issued relating to the same interception target. These various lawful interceptions may contain different constraints on the IRI and the CC. These various lawful interceptions may fall under different laws.

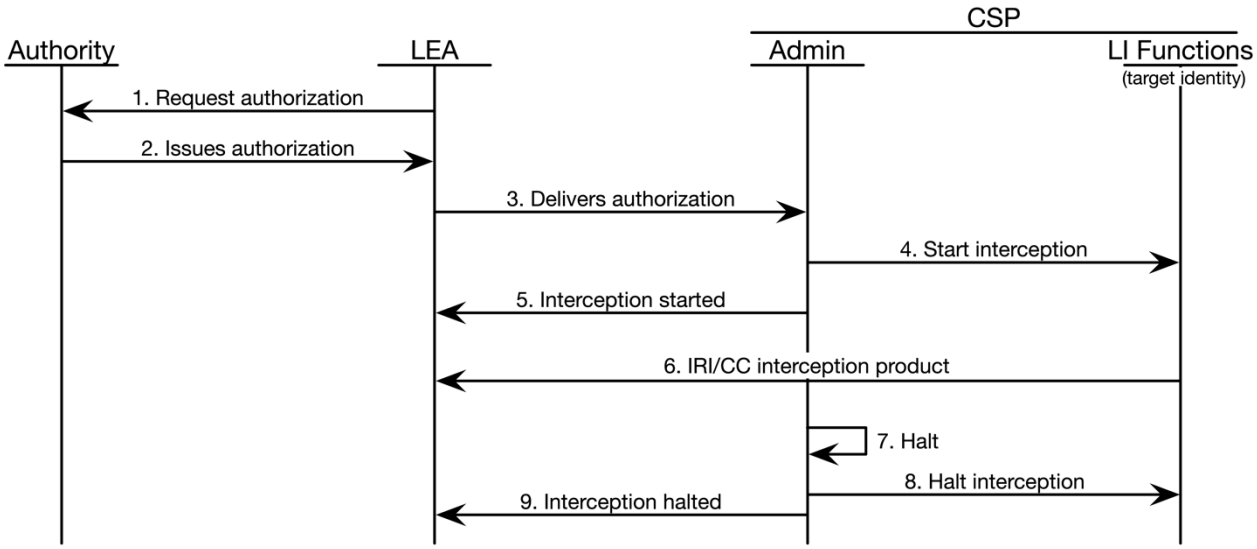

Figure 4.2.1-1: General provisioning flow

4.2.2 Entities

The entities in the functional flow in figure 4.2.1-1 are given in table 4.2.2-1.

Table 4.2.2-1: Provisioning entities

Entity	Role
Authority	The authorization authority is a judicial or administrative body designated by local laws or regulations. It gives the LEA the lawful authorization to intercept a target.
LEA	The LEA requests that the CSP intercept communications according to a lawful authorization. The LEA receives, through a Law Enforcement Monitoring Function, the interception product (CC and IRI) relating to a target identity.
CSP	An entity which provides communication services to subscribers.
Target identity	The target identity corresponds to the identity of a given interception target, which is an entity that makes use of a given service offered by a CSP.

4.2.3 Process

The process as described in this clause stands as an example. In a specific country, the national process will be based on various national laws and circumstances.

The authorization authority requires, through the LEA, the interception of services utilized via the telecommunication network by the interception target. The LEA receives the communications involving the target identity(ies) which the CSP has associated with the interception target.

Referring to the functional role model, and assuming that the lawful authorization is to be given to a CSP, actions are shown in table 4.2.3-1.

Table 4.2.3-1: Functional role model process actions

Reference (see figure 4.2.1-1)	Action
1	An LEA requests lawful authorization from an authorization authority, which may be a court of law.
2	The authorization authority issues a lawful authorization to the LEA.
3	The LEA sends the request for lawful interception along with the lawful authorization to the CSP. The CSP determines the relevant target identities from the information given in the lawful authorization.
4	The CSP causes interception facilities to be applied to the relevant target identities.
5	The CSP informs the LEA that the lawful authorization has been received and acted upon. Information may be passed relating to the target identities and the target identification.
6	IRI and CC related to the target identity are passed from the CSP facilities to the LEA LEMF.
7	Either on request from the LEA or when the period of authority of the lawful authorization has expired the CSP will cease the interception arrangements.
8	The CSP signals its facilities to halt interception (see note).
9	The CSP announces the cessation to the LEA (see note).
NOTE: Steps 8 and 9 may be asynchronous.	

To apply interception, a network administrator typically requires the following parameters for the special commands:

- Target identification.
- LEMF address for CC.
- LEMF address for IRI.
- Delivery address parameters for LEMF (e.g. for authentication and security).
- Alarm routing (if different from the delivery address).

The syntax of the necessary commands may be different in various systems.

4.2.4 LI lifecycle

Figure 4.2.4-1 depicts the general LI lifecycle state machine.

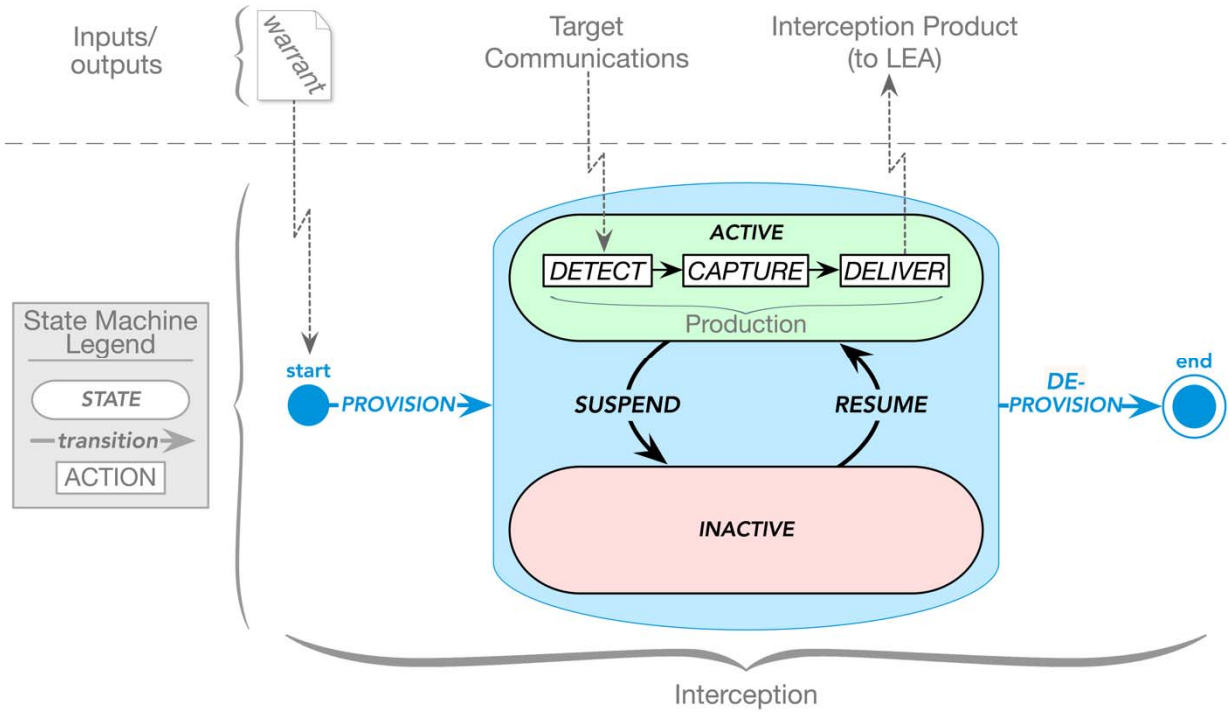

Figure 4.2.4-1: LI lifecycle state machine

After an LEA delivers a warrant to the CSP, the CSP provisions the interception. In the ACTIVE state, the Lawful Interception system elements *detect*, *capture* and *deliver* interception product to the LEA (labelled "production" in figure 4.2.4-1). These three production actions occur each time a targeted communication is identified, and therefore may happen many times during the lifecycle.

Depending on requirements, once provisioned, the LI system can enter directly into the ACTIVE state (immediate activation of LI), or enter the INACTIVE state (for delayed activation of LI), in which it still requires a RESUME transition to enter the ACTIVE state. The "production" activities of *detect*, *capture*, and *deliver* from figure 4.2.4-1 happen only in the ACTIVE state. It is in this ACTIVE state only that interception product is delivered to the requesting LEA.

A transition from INACTIVE to ACTIVE will resume the process from the *detect* action. Conversely, a transition from ACTIVE to INACTIVE will immediately stop *detection* and *capture*, but finish *delivery* to Law Enforcement of previously captured (during the ACTIVE state) product.

If provisioning causes LI to enter the INACTIVE state for a delayed start of interception, once the delay period is over the RESUME transition will occur moving the LI into the ACTIVE state.

Some jurisdictions may not support the delayed start of the interception. In such cases, provisioning of interception causes LI to immediately transition to the ACTIVE state, thus the production actions start directly upon provisioning, and stop directly upon de-provisioning.

5 Functional view

5.1 General

A high-level functional view of the LI architecture is given in figure 5.1-1 below.

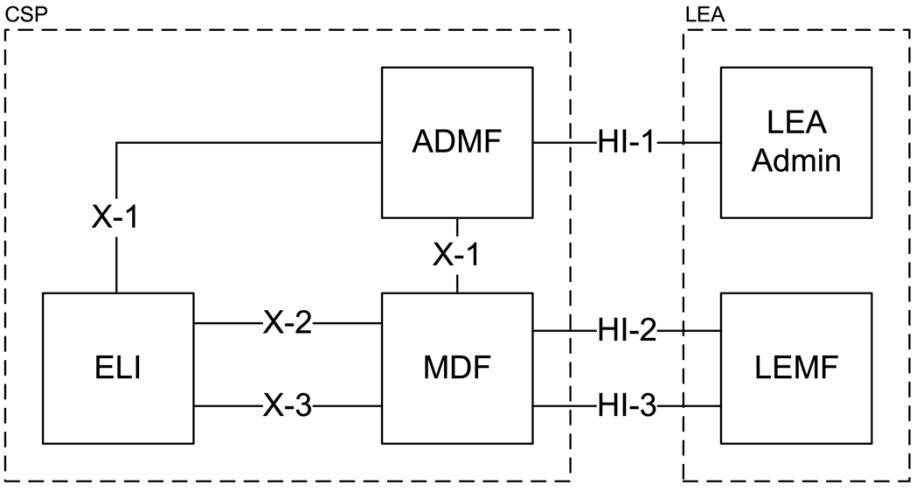

Figure 5.1-1: Functional architecture

Table 5.1-1 below gives a brief description of these functional elements; however, the reader should be aware that security requirements and zero-trust principles mean that these functional elements are further subdivided into smaller logical elements (see clause 6).

Table 5.1-1: High-level functions

Function	Description
ADMF	LI administrative function. Responsible for controlling the other LI functions within the CSP's network, in response to warrant and tasking information received from the LEA administrative function (see note).
MDF	Mediation and delivery function. Packages LI product coming from ELIs for onwards delivery and fan-out to the relevant LEAs.
ELI	Element of LI. All-encompassing term for any function that performs LI functions.
LEA Admin	LEA administrative function. Responsible for providing warrant and tasking information to the CSP.
LEMF	Law Enforcement Monitoring Facility. Responsible for receiving LI product from the CSP.
NOTE: Multiple/distributed ADMF deployment models are possible, but out of scope of the present document.	

Table 5.1-2 below gives a brief description of the interfaces shown; however, the reader should be aware that security requirements and zero-trust principles imply that the security architecture subdivides these interfaces (see clause 6).

Table 5.1-2: High-level interfaces

Interface	Description
HI1	The HI1 interface is used to send warrant and other interception request information between the LEA and the CSP.
HI2	The HI2 interface is used to send IRI from the MDF to the LEMF.
HI3	The HI3 interface is used to send CC from the MDF to the LEMF.
X1	The X1 interface is used for provisioning and control of LI functions in the network.
X2	The X2 interface is used to deliver intercept related information.
X3	The X3 interface is used to deliver content.

6 Security first approach

6.1 Introduction

6.1.1 Approach

This architecture is rooted in the Zero Trust principles outlined by NIST SP 800-207 [i.4]. The first step is to identify the resources or assets that the network comprises or uses, which need protection, and allocate each of them to appropriate trust domains.

6.1.2 Trust domains

6.1.2.1 Trust domain definition

A trust domain is a collection of functions that share the same set of administrative and security policies (particularly access control). Trust domains are further elaborated on in clause 6.3.

6.1.2.2 Domain separation

In this architecture, trust in the network is based on technical procedures and requirements placed on functions in an attempt to replace assumed trust of functions on the basis of their location only. Because the network functions where the Elements of LI (ELIs) reside are dynamically created, this architecture is designed to be able to integrate the orchestration and management of network functions and the ELIs. Consequently, the ADMF shall contain functions that handle the trust establishment of ELIs, as they are instantiated and as they evolve through their life cycle. The ADMF further contains functions to manage the LI interfaces and the LI operations that provide intercepted information to the LEA. Moreover, functions for orchestration of ELIs should be kept isolated from LI intercept run-time operations. In such context, this architecture contains a logical separation of trust domains and requirements on how information exchange between trust domains is to be handled to reduce risk of compromise propagation.

6.1.2.3 Controlled interconnection

Functions in different trust domains need to exchange information. APIs that are used inside a trust domain shall be separated and protected from APIs that facilitate the interconnections for cross-trust-domain exchanges. This separation and controlled interconnect of trust domains in this architecture is embodied in a newly defined Cross Trust Domain GateWay (CTDGW). Figure 6.1.2.3-1 introduces the concept.

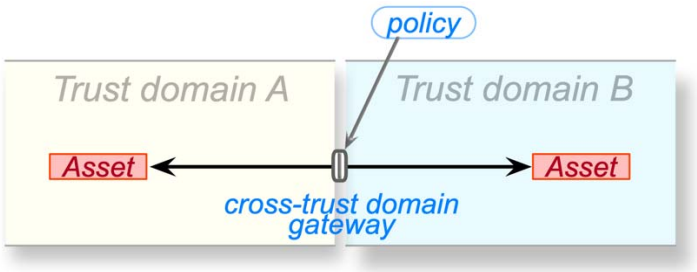

Figure 6.1.2.3-1: Cross-trust-domain gateway

6.1.2.4 Cross-trust-domain gateway deployment

Figure 6.1.2.3-1 is necessarily simplistic in introducing the idea of a CTDGW, as it shows the CTDGW spread across trust domains. This is clearly not possible in practice, as CTDGWs will belong to one or the other domain and can be deployed independently of the functions and trust domains they separate or within them. Figure 6.1.2.4-1 offers the possible combinations.

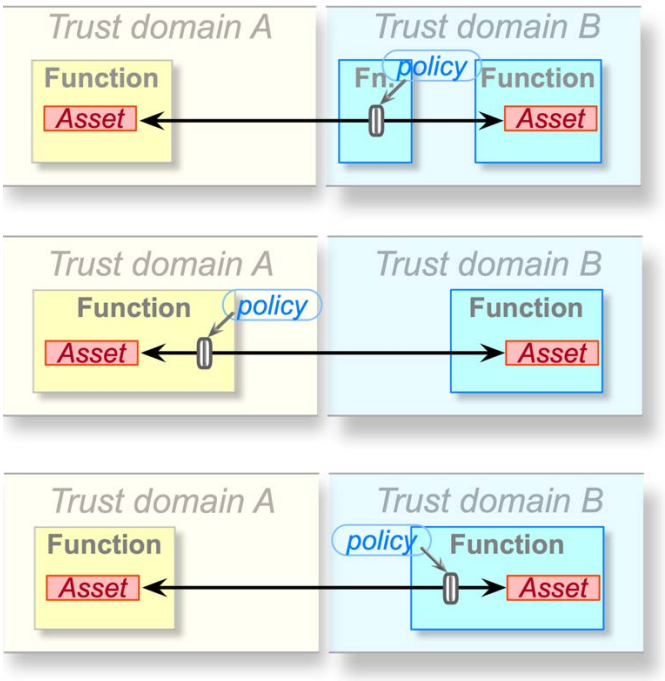

Figure 6.1.2.4-1: CTDGW deployment options

This clarifies that the domain that implements the CTDGW is the one and same domain that establishes the policy that governs the west/eastbound flows. Certainly, both domains can choose to deploy local CTDGWs, based on local security policy or criteria.

6.2 LI assets

6.2.1 Identifiers are fundamental LI assets

In the LI domain, a central resource, or asset, is the ***target ID***. Further important assets (or resources) are: the lifecycle state of LI, the interception product, points of interception, topology/connectivity information, etc. However, these are arguably downstream derivatives of the fundamental assets. Figure 6.2.1-1 contextualizes the main identifiers and places them in top-level trust domains.

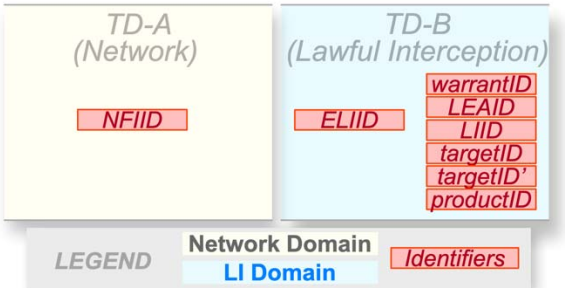

Figure 6.2.1-1: Primary LI assets

There are two main trust domains of interest: the network trust domain (A) and the LI trust domain (B). Table 6.2.1-1 defines these identifiers.

Table 6.2.1-1: Identifiers

Identifier	Description
NFIID	The identifier of the Network Function that contains the data of interest to LI. The NFIID is the value assigned to the network function instance by the orchestration layer.
ELIID	The ID of the Element of LI that is tasked to perform any LI function.
WarrantID	The ID of the warrant (or legal administrative instrument) provided by the LEA to the CSP.
LEAID	The ID of the Law Enforcement Agency that presented the warrant to the CSP.
LIID	The ID assigned by the CSP that ties together all other IDs.
TargetID	The original ID derived by the CSP from information presented by the LEA in the warrant.
TargetID'	The network may derive other targetIDs from the initial targetID.
ProductID	The ID assigned by the CSP to the interception product.

The landscape of the architecture quickly complicates when the lifecycle of the identifiers is considered. Each identifier is spawned and managed in one trust domain but may continue to be managed while being used by functions within other trust domains. Specifically, the lifecycle of the ELI is inherently tied to the lifecycle of the encompassing NF, but an NF and its associated ELI are managed and controlled from different trust domains. The identifiers are used, and therefore transferred, between trust domains, as in figure 6.2.1-2.

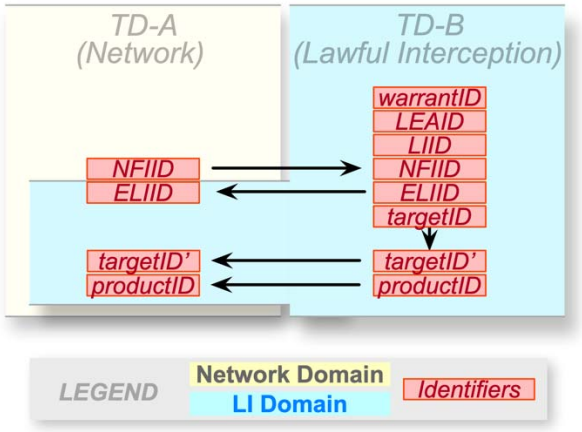

Figure 6.2.1-2: Identifiers cross trust domains

Through the lifecycle of LI, ID usage and projection is even more complicated, as seen in figure 6.2.1-3 which introduces the concept of sub-trust-domains which will be expanded in clause 6.4.

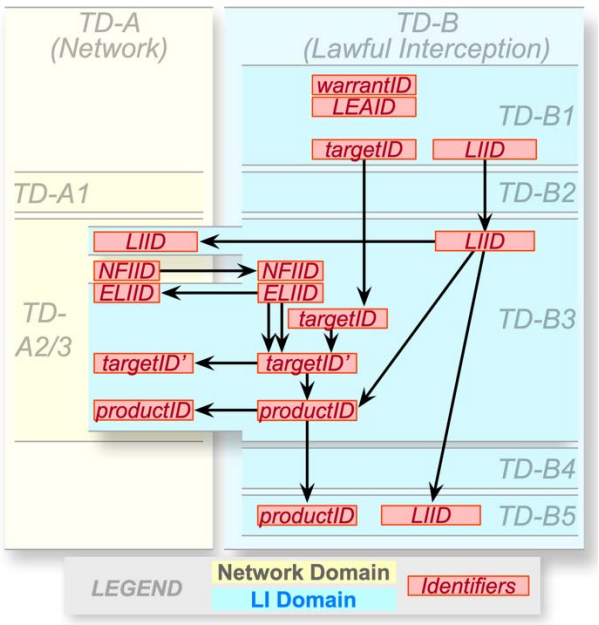

Figure 6.2.1-3: Identifiers project across sub-trust-domains

6.2.2 Further LI assets

As previously indicated, reducing the security of the LI domain to the security of target IDs is simplistic. Further assets to consider in a holistic approach to the security of LI are the lifecycle state of LI, the interception product, points of interception, topology/connectivity information, the separation of sub-trust domains, the assignment of administrators in and across trust domains, etc. This is beyond the scope of the present document but has to be considered in LI deployments.

6.3 Trust domains

6.3.1 Introduction to trust domains

Protecting the identifiers gives rise to the idea of segregating them into Trust Domains (TDs). TDs are central to the security of the LI architecture. In the spirit of complying with the NIST-defined Zero Trust Architecture (ZTA) [i.4] where TDs are central, as a consequence TDs are also foundational in the LI architecture defined in the present document.

To reiterate, a trust domain is a collection of functions that share the same set of administrative and security concerns and policies (particularly access control). It is expected that compartmentalization of trust domains spans the vertical stack of functionality in the CSP network, ideally from hardware to the application layer. Administrators in the CSP network are assigned to one or more trust domains based on CSP criteria, using state of the art Attribute Based Access Controls (ABAC). In virtual networks, multiple trust domains should be considered by CSPs during the deployment phase, where a CSP wishes to achieve security role and management separation, security isolation, separation between sensitive and non-sensitive components, etc. Whether or not a CSP uses these concepts in their networks, if a network is LI obligated, a CSP shall use them in the LI space.

6.3.2 Trust domain collapse

The present document is not prescriptive in the trust domain definitions but proposes an implementation that takes separation of concerns as a primary driver resulting in maximally separated trust domains.

For example, a common requirement across multiple jurisdictions is to ensure that if a target is under surveillance by multiple LEAs, "[...] *the CSP shall perform interception in such a manner that no other LEA can detect that interception is taking place, before, during, and after interception.*" (see 3GPP TS 33.126 [5], requirement "R6.6 - 40" in clause 6.6). Therefore, since the network needs to keep information about the serving LEA in the administrative function, it makes sense to keep that information in the most restrictive footprint possible while still performing the interception. This drives the need to split the administrative function itself into multiple trust domains.

While maximal sub-trust domain separation is clearly a good principle to drive implementation, this approach is costly both in terms of network and personnel resources. An implementation may choose to collapse sub-trust domains, but the risks of doing so can only be quantified if the implementation of present document takes the most restrictive approach. Every decision to collapse (sub) trust domains shall be accompanied by a thorough security analysis, coupled with explicit security mitigations.

6.4 LI architecture

6.4.1 Security-first approach to LI architecture

Figure 6.4.1-1 introduces the main functions that operate LI in the network, embedded in their respective trust domains.

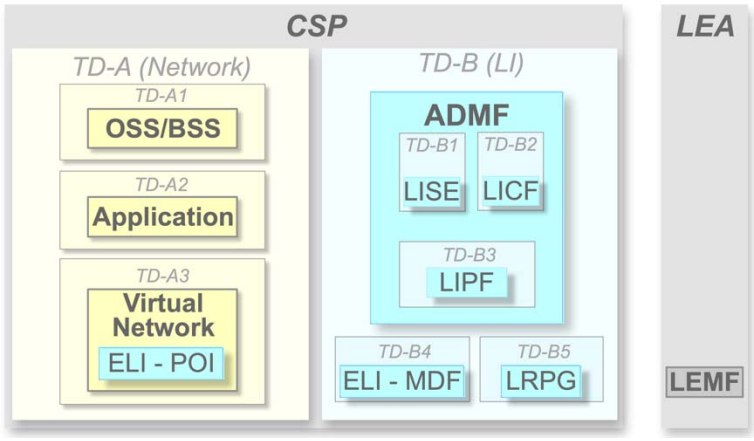

Figure 6.4.1-1: Principal network functions embedded in trust domains

Table 6.4.1-1 describes the trust domains and sub-trust-domains.

Table 6.4.1-1: Trust domains

TD	sub-TD	Description
TD-A		Contains the network itself. It includes Operations Support Systems/Business Support Systems (OSS/BSS), and all the standard Network Functions (NFs) that make up a communication network. Layer-wise, included here are both the infrastructure/ virtualization layer, managed by MANO [3], as well as the application layer NFs, implemented as virtual functions that provide the communication services to users.
	TD-A1	Contains the traditional OSS/BSS, including the hereby defined OS-LI OSS-layer LI management function, the root CA for the CSP, and the core of the attestation system.
	TD-A2	Contains the application layer of the network.
	TD-A3	Contains the virtual layer of the network, VNFs and MANO that implement a modern 5G service network.
TD-B		Contains highly sensitive and correlated information on LEAs, warrants, targets, and in-network LI-cleared administrator accounts. Functions in this domain may be virtualized, but, if they are, they may be virtualized in their own segregated cloud for better isolation.
	TD-B1	Is the domain with the primary role of segregating the most important assets (warrants, targets, Law Enforcement Agencies [LEAs], etc.) of the LI domain.
	TD-B2	TD-B2 holds the trust state of LI elements in the network. It contains the trust/validation state of LI in the network, as well as the certification mechanisms.
	TD-B3	TD-B3 is a buffer domain that separates the more sensitive sub-domains in TD-B (TD-B1 and TD-B2) from the network at large. It contains cross-trust-domain gateways, the LIGW, which ingests a standard, non-LI interface, and the LIPF, which firewalls LI functions distributed throughout the network and holds the mappings of the LI elements to the network functions.
	TD-B4	TD-B4 also belongs to the LI trust domain but differs in that it connects to LEAs outside the network. The MDFs in this domain collect information from ELIs throughout the network, filter/format/package the information according to warrant parameters, and distribute the interception product to the requesting LEA. They may run in the network, but belong fully to TD-B, as they have no overt network functionality.
	TD-B5	TD-B5 is a "demilitarized zone" (DMZ) between the CSP and the LEA. It hosts firewalls (LRPG(s)) that interface the two domains and protect LEA delivery addresses and topology from the MDFs, which are still part of the network.

The functions are defined below in clause 6.4.2.

6.4.2 LI architecture including IDs

Figure 6.4.2-1 adds interfaces to the architecture picture. It also shows the IDs in architectural context. IDs are used as the primary (but not singular) driver of separation of trust domains.

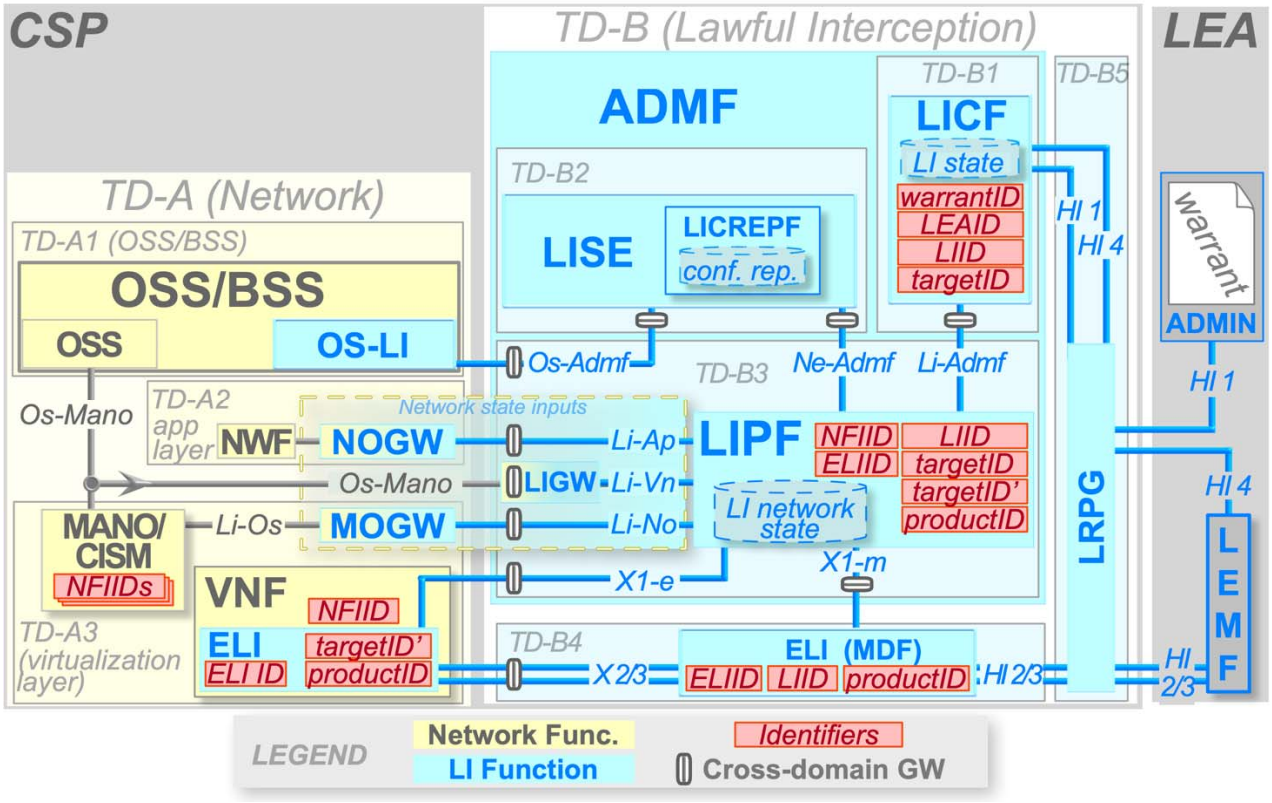

Figure 6.4.2-1: IDs as drivers of trust domain separation

At the highest level, there are two main trust domains: domain A, the customer serving CSP network, and trust domain B, which contains the LI elements in the CSP network. All elements in a TD have common trust attributes (e.g. access control, confidentiality restrictions).

There are three main colours in figure 6.4.2-1. Identifiers are red. Network elements are yellow. LI is blue.

Information that crosses between domains A and B always passes through a CTDGW or an inter-domain function (LIGW or MOGW). These inter-domain functions straddle trust domains, and as such will be expected to be the most challenging to implement and secure. For example, the MANO GateWay (MOGW) converts MANagement and Orchestration (MANO) events for the LI network but is deployed in the network domain. The LI GateWay (LIGW) in the LI ADMinistrative Function (ADMF) takes a standard OSS-MANO interface and converts it into an LI interface in the LI domain. Each of these cross-trust domain elements require deep analysis to be implemented securely, both in terms of functionality and security. Note the presence of CTDGWs at the ingress of every higher-sensitivity domain as links come in from a lower-sensitivity domain. The term "sensitivity" is used loosely here, as there is not strictly monotonously increasing function to help in ranking trust domains. A lesser-sensitivity domain may have a CTDGW on its egress points, no matter what is done in the higher-sensitivity domain.

In general, NFs are Network Functions, and ELIs are Elements of LI, such as Points of Interception (POIs), Triggering Functions (TFs), and Mediation and Delivery Functions (MDFs) as defined by in 3GPP TS 33.127 [6]. The LI architecture contains the following functions.

Table 6.4.2-1: Functions

Function	Description
ADMF	The LI **ADM**inistrative **F**unction is the heart of the LI system. It is responsible for controlling the other LI functions within the CSP's network, in response to warrant and tasking information received from the LEA administrative function. It is comprised of three trust domains TD-B1, TD-B2, and TD-B3, and two top level functions, the LICF and the LIPF, along with the LISE that plays the role of root of trust for the LI system.
attester	An entity which needs to present evidence to the attestation system before it becomes a trusted part of the system.
ARP	The Attestation Relying Party is the ultimate customer of the verification system. The LISE relies on its state to make decisions on the viability/trustworthiness of ELIs in the network.
AVS	The Attestation Verification Service is part of the attestation system. The verifier ingests attestation evidence from attesters and prepares results for the relying party's use.
CISM	The **C**ontainer **I**nfrastructure **S**ervice **M**anagement performs its function alongside MANO [i.2], [i.3]. It is not included in deployments that are strictly Virtual Machine-based.
ELI	**E**lement of **LI** - an all-encompassing term for an element that performs an LI function.
ground truth	The attestation system database that persists image snapshot hashes of hardware and software system elements, which permits runtime comparisons to establish trust that the system is running unmodified from a known state.
LEMF	The **L**aw **E**nforcement **M**onitoring **F**unction is managed by the LEA, and is outside the CSP network, and therefore outside the scope of the present document.
LI State	The state machine of the LI system which contains target identifying information, the state of particular intercepts, indeed any and all persistent LI information in the network.
LI Network State	The LI system view of the network state is built in the LIPF. Generally, this state correlates ELI/NF deployment.
LIGW	The **LI G**ate**W**ay is a function which translates a standard NFV interface for LI use.
LI-CMF	The **LI C**ertificate **M**anager contains the list of approved and revoked certificates in the network, and the LICA.
LICA	The **LI C**ertificate **A**uthority is the root of trust of the LI system. This can be deployed as a stand-alone trust tree, or as an intermediate **CA** to the root **CA** in **OSS/BSS**. If the choice is made to implement the **LI CA** as an intermediate **CA** to the root **OSS/BSS CA**, the root **OSS/BSS** administrators will have to be fully trusted by the LI trust domain and LI system will have to be capable to restrict certificate usage to certificates from the LI intermediate CA.
LICF	The **LI C**ontrol **F**unction manages the lifecycle of warrants and contains the authoritative record of the most highly sensitive and correlated information on LI agencies, warrants, targets in the network. The **LICF** also maintains and authorizes the lifecycles of all LI functions in the network, along with logs and audits.
LIPF	The **LI P**rovisioning **F**unction acts as a secure proxy between the sensitive LICF and the rest of the network, facilitating provisioning and other LI events. It holds the **LI N**etwork **S**tate.
LISE	The **LI S**ecurity **E**ngine is the beating heart of the system. It manages the network-wide security primitives (keys, nonces, salts, etc.) needed by the LI network functions. It contains the **Certificate Management System**, and it also is the **Relying Party** in the attestation layer.
LRPG	The **LI R**outing **P**roxy **G**ateway is responsible for hiding the **LEMF** end point addresses and routing information from the overt network, which is not authorized to know about LI. The **LRPG** is placed at the edge of the NFV network/SDN where a physical hidden secure connection to the **LEMF** can be implemented, or when a dedicated LI SDN cloud connection can be established which does not need to be visible to the CSP NFV network/SDN.
MANO	The **MAN**agement and **O**rchestration block is part of the NFV architecture.
MDF	The **M**ediation and **D**elivery **F**unction packages the LI product coming from **ELIs** for delivery and fan-out to the requesting **LEAs**.
MOGW	The **MOGW** is responsible for converting **MANO/CISM NFV** events into events usable by the LI network overlay to ensure that the correct **ELIs** are paired with the correct NFs both at the virtual and application layers. It translates virtual network outputs from ETSI NFV deployments, raw Kubernetes deployments, or others, into the standard **LI-NO** interface to the **ADMF**.
NF	The **N**etwork **F**unctions are virtual network functions that are composed into network services.
NOGW	The **NOGW** is responsible for converting application layer events into events usable by the LI network to ensure that the correct **ELIs** are paired with the correct NFs.
NWF	The **NWF** is an application layer function that feeds the LI system the necessary information to perform its duty. For example, in a 5G system it may be the NRF, while in a pre-5G system it may even be a manual intervention directly into the LI system to provision it with network functionality information.
OS-LI	The **OSS LI** management function. It may also include manual activities such as verifying the correctness or completeness of lawful authorizations.
OSS/BSS	The **O**perations **S**upport **S**ystems/**B**usiness **S**upport **S**ystems are the top-level functions in the network, from which the most fundamental events, such as spawning a new function, are initiated.
rootCA	The root Certificate Authority for the CSP.

6.4.3 LI architecture

In figure 6.4.3-1 the IDs are excluded to simplify the diagram.

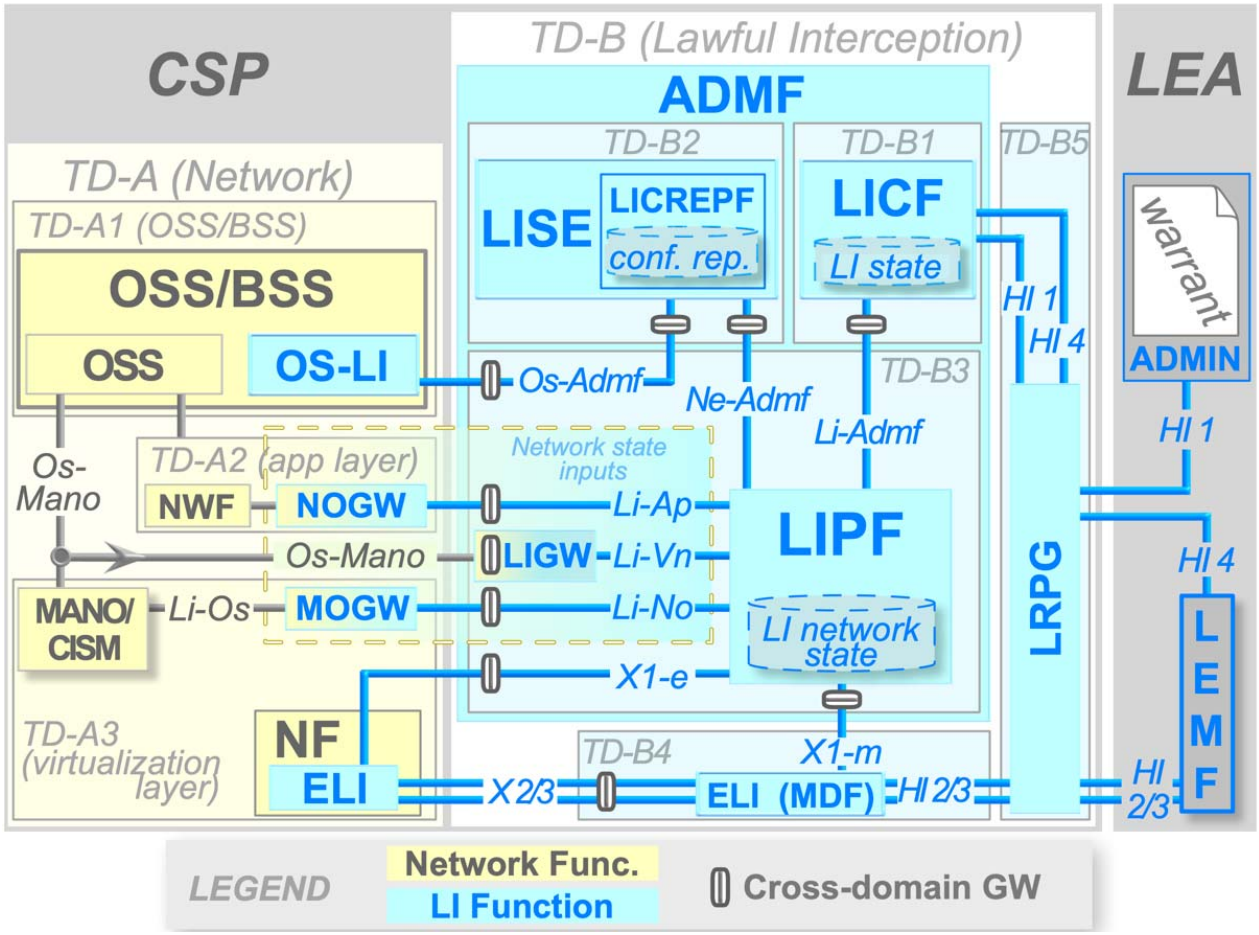

Figure 6.4.3-1: LI architecture

As depicted in figure 6.4.3-1, there are three main interfaces that are used to bring network state into the LI domain (Li-Ap, Li-Vn, and Li-No). Not all of these will necessarily be found in every network, depending on the technology implemented in TD-A. A virtual network would be expected to provide some view of the virtualized function deployment state through Oss-Mano. Some networks will provide application layer information through Li-Ap, and figure 6.4.3-1 can be mapped modern 5G network that includes an NRF and SIRF, depicted here as the NWF and NOGW. However, a 4G network would create the information needed by Li-Ap from different architectural elements, specific to each 4G network. Equally, a 6G network would source the information for Li-Ap from the appropriate 6G architectural elements. Further, a non-virtualized network would also supply network state information through Li-Ap, but clearly Li-Vn and Li-No would not apply.

The LI side of this architecture should therefore be stable across generations of technology used to implement TD-A.

The LI architecture contains the following interfaces.

Table 6.4.3-1: Interfaces

Function	Description
LI-Admf	The LIPF takes information received from the OSS/BSS layer over **Os-Ma-Nfvo** and compares/correlates it with information from the **MOGW** received over **LI-NO** and builds a coherent picture of the virtual network. The LI relevant aspects are passed northward to the LICF through **LI-Admf**.
Li-Ap	The **LI**-**Ap**plication interface comes from the application (likely defined by 3GPP as the "5G" network, or a future 6G network) and feeds the LI system with application layer topology.
Li-No	The **LI-Network Output** interface is used for information exchange between the MOGW and the LIPF. These events include lifecycle management events, status information, security policy enforcement and VNF management information for network services. Information from **LI-NO** can be compared and correlated with information from **Li-Vn** as the LIPF builds its picture of the network.
Li-Os	The **LI-Operation Support** interface is used for information exchanges between the MANO/CISM and the MOGW, which include lifecycle management events, status information, security policy enforcement and VNF management information for network services. For example, the ETSI NFV-defined interfaces that correspond to **LI-Os** may be *Sc-Or*, *Sc-Vi, Ve-Vnfm-Em, or a combination thereof*.
Li-Vn	The **LI**-**V**irtual **N**etwork interface is a copy of the OSS->virtual network provisioning interface. It can be used by the LIPF to correlate with information coming over the Li-No interface, which is "post" virtual network management.
Ne-Admf	The **Ne-Admf** interface carries trust building information between the LISE and the LIPF.
Os-Admf	This is the interface between OS-LI and LISE. It carries administration and provisioning information from OSS/BSS.
Os-Mano	The **Os-Mano** interface passes information from the OSS/BSS to the MANO NFVO. It is a standard interface defined by ETSI NFV.
Oss-LI	The **Oss-LI** interface takes information from **Os-Mano** that has been domain-isolated by the LIGW and delivers it to the LIPF. It is a one-way interface, southbound to the LIPF.
X0	The **X0** interfaces are a group of interfaces that prepare the ELIs to receive target identifiers by establishing trust, rooted in hardware, and attested remotely. They are also the interfaces over which the attestation system performs its functions.
X1	The **X1** interface provisions the target and session level identifiers required to isolate, duplicate, and route the target communications towards the LEMF.
X2	The **X2** interface conveys Intercept Related Information (IRI) related to target communications from the network to the LEMF.
X3	The **X3** interface conveys Communication Content (CC) from the network to the LEMF.
HI1	The **HI1** interface is used to send warrant and other interception request information between the LEA and the CSP.
HI2	The **HI2** interface is used to send IRI from the MDF to the LEMF.
HI3	The **HI3** interface is used to send CC from the MDF to the LEMF.
HI4	The **HI4** interface is used to send notifications from the MDF to the LEMF.

6.4.4 Simplified LI architecture

A further simplification is possible, by abstracting the inputs in the dotted line box labelled "Network state inputs". This simplifies the network side as seen from the LI (TD-B) perspective.

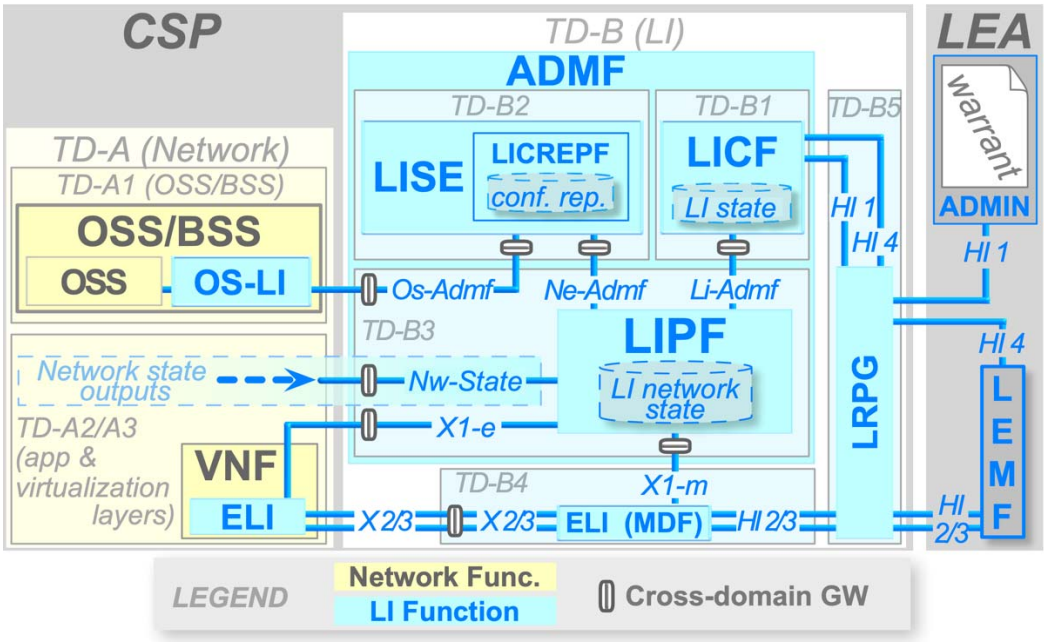

Figure 6.4.4-1: Simplified LI architecture

The three interfaces Li-Ap, Li-Vn, and Li-No are abstracted away into "Nw-inputs". Every CSP network will use some, or all the network state inputs to convey the required information to the LI TD-B. In legacy, non-virtualized networks, some of the Nw-State information may not be automated and be manually provisioned.

6.5 Attestation

To fully support Zero Trust principles, security is rooted in immutable hardware and provide provisions for entities to verify other entities prior to engaging in any business logic interactions. This is achieved with remote attestation, which is fully described in Annex A of the present document. At the highest level, a Relying Party uses a long(er) term store of trust (the "ground truth") in a Verifier, to make trustworthiness decisions about more short-lived Attesters. The use of attestation is always recommended. Where the core network (non-LI part) supports attestation, the use of attestation (as described in clauses 6.5 and 7) for LI is mandatory. Where the core network does not support attestation, alternative security measures should be taken.

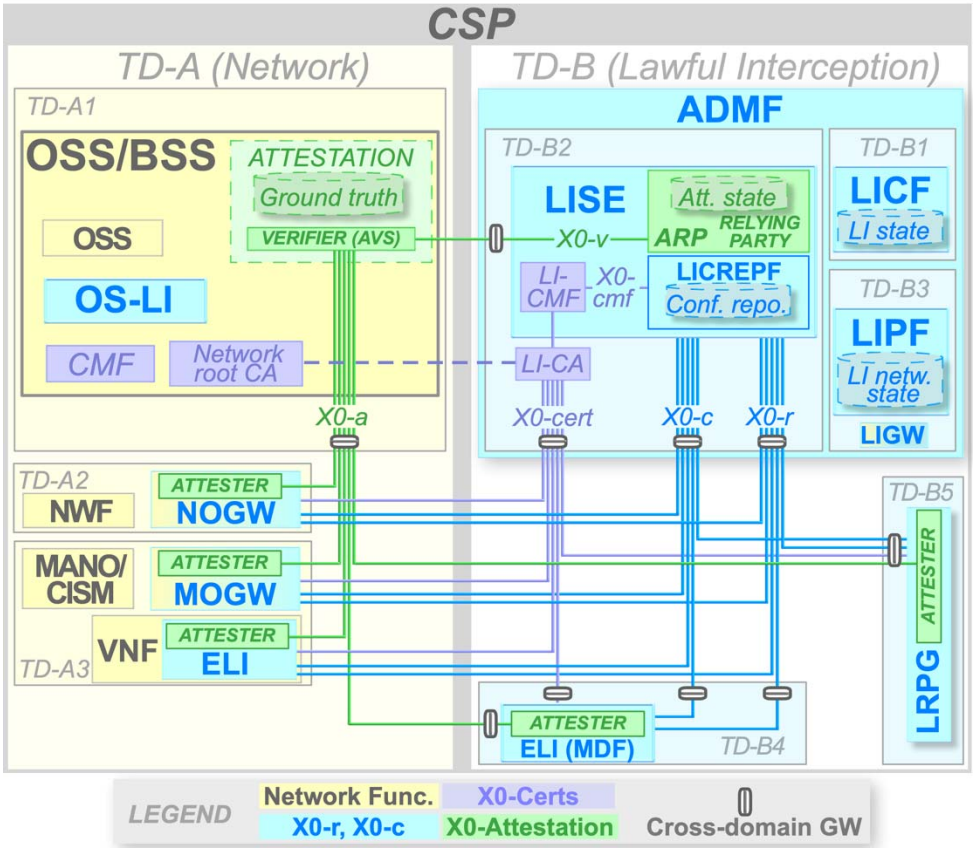

Figure 6.5-1: Attestation framework

The interfaces involved in attestation are fully described in ETSI TS 104 000 [4]. Table 6.5-1 offers an overview of their functionality.

Table 6.5-1: Attestation interfaces

Interface	Description
X0_r	**Registration**. Used to build initial trust between a newly started ELI and the ADMF during the Registration phase (see ETSI TS 104 000 [4], clause 5.4). The protocol used to realize this interface is defined through the messages defined in ETSI TS 104 000 [4], clause 5.4, and associated parameters defined in ETSI TS 104 000 [4], clause 6.
X0_c	**Configuration**. Used to exchange configuration-related information in the X0 and Xn Configuration phases (see ETSI TS 104 000 [4], clauses 5.5 and 5.7). The protocol used to realize this interface is defined in ETSI TS 104 000 [4], clause 6.
X0_cert	**Certificates**. Used to perform certificate enrolment by the ELI via the LICA as part of the Registration and Certificate Enrolment phase (see ETSI TS 104 000 [4], clauses 5.4 and 5.6). The protocol used to realize this interface is out of scope of the present document.
X0_a	**Attester**. Used to attest the ELI to the AVS during the Attestation phase (see ETSI TS 104 000 [4], clause 5.3). The protocol used to realize this interface is out of scope of the present document.
X0_cmf	**Management**. Used by the LISE to manage certificate enrolment via the CMF for both X0 and Xn certificate enrolment (see ETSI TS 104 000 [4], clauses 5.4 and 5.6). The protocol used to realize this interface is out of scope of ETSI TS 104 000 [4] and of the present document.
X0_v	**Verifier**. Used by the LISE to verify the attestation results of an ELI during the Attestation phase (see ETSI TS 104 000 [4], clause 5.3). The protocol used to realize this interface is out of scope of the present document.

The motivation behind attestation, and a very basic tutorial is offered in Annex B of the present document.

6.6 Certificate management

Figure 6.6-1 shows LI certificate distribution across the network. The distribution for X2/X3 certificates is done similarly to the distribution of X1 certificationThe LI Certificate Management (LICM) function contains the LI Certificate Authority (LICA).

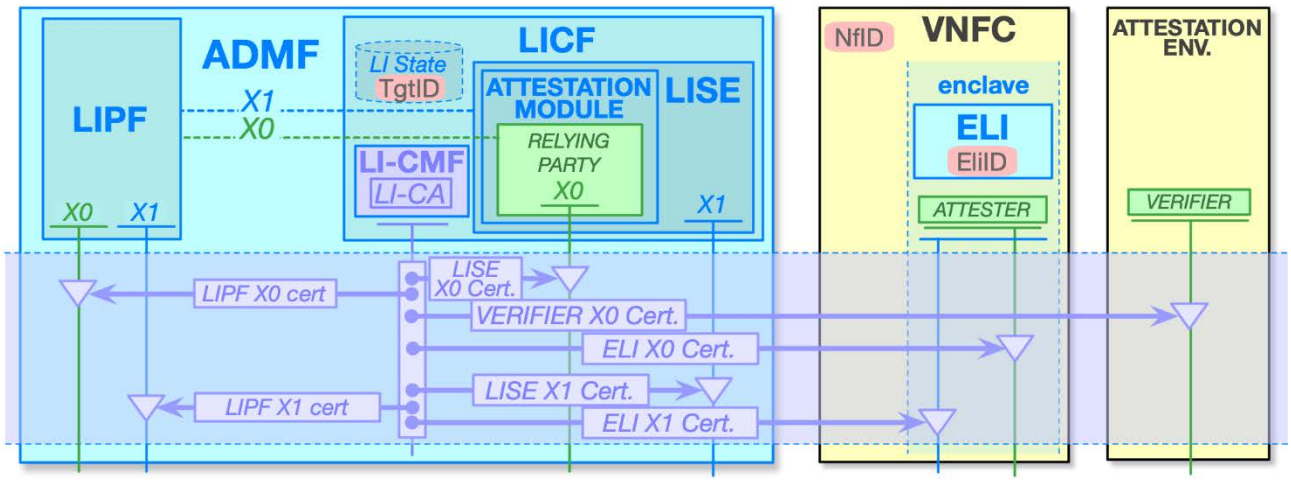

Figure 6.6-1: Certificate management

All interfaces shall be assigned certificates to set up TLS connections. The LI elements are started in secure enclaves, and TLS connections shall terminate on both sides inside enclaves.

There is no chronology or ordering implied by figure 6.6-1 on the order of certificate provisioning: the X0 and X1 certificates in the LIPF, LICF, and the Verifier can be longer lived, while the X0 and X1 certificates in the VNF ELI are necessarily more short-lived.

For networks that expect high VNF turnover, certificate management (maintaining expiration and revocation lists, etc.) can quickly become unmanageable. In such cases, a more agile ticket/token management mechanism, Kerberos-like for example, may be used.

There shall be a dynamic mechanism beyond certificate revocation to ensure that connections to functions that have become untrustworthy can be torn down immediately when active, not just merely prevented from re-starting in the future by a revoked/invalid certificate.

7 Provisioning

7.1 Provisioning Phases

7.1.1 Overview

Before targets can be provisioned in ELIs, a stepped process to build trust in them takes place, as described in detail in ETSI TS 104 000 [4]. Figure 7.1.1-1 is an overview of phases 0 through 6 of the process detailed in ETSI TS 104 000 [4]. Once the configuration of the LI interfaces, see figure 5.1-1, has been done, all steps in figure 7.1.1-1 do not need to be replicated for each warrant. Further, this overview is not meant to be prescriptive, as networks may differ in how network functions are deployed and orchestrated. The overview, however, can serve as a useful high level summary of the detailed process in ETSI TS 104 000 [4].

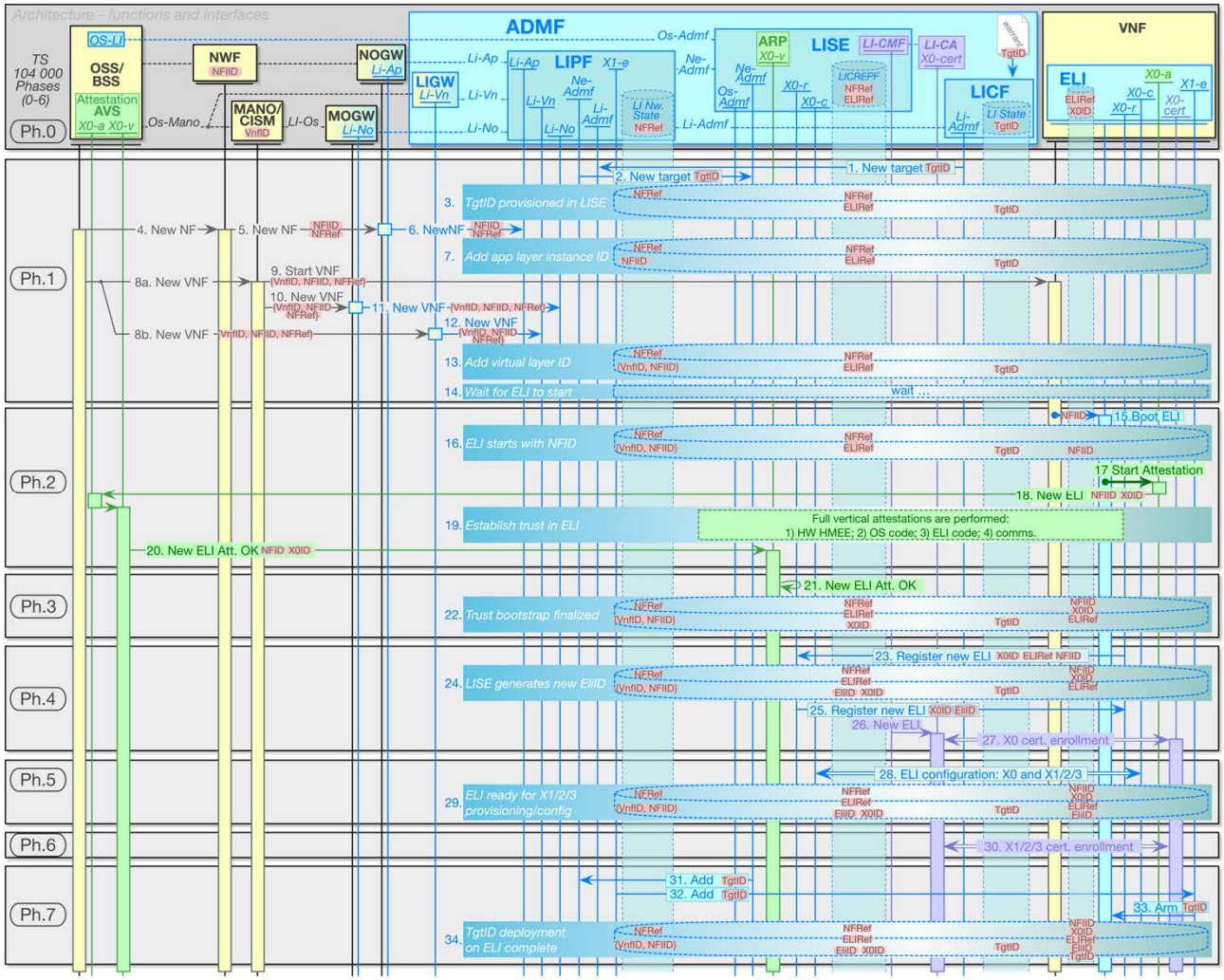

Figure 7.1.1-1: ETSI TS 104 000 [4] provisioning phases

The IDs in the header (in Phase 0) indicate an ownership/management relationship between the NF and the ID.

7.1.2 Phase 0 (X0)

See ETSI TS 104 000 [4], clause 5.1 for a description of this phase.

At the end of the on-boarding/testing/acceptance process from the vendor into the carrier network, default values are provisioned into the relevant systems. Most importantly, the AVS attestation measurement database will contain known good values for all the hardware and software in the system.

The ADMF is configured with the NFRef and ELIRef values, e.g. using OS-Admf. The NFRef being the identifier for the X0 pre configuration, see ETSI TS 104 000 [4], clause 5.2, of the deployed NF, and the ELIRef the identifier that uniquely identifies the configuration information for an ELI. The NFRef and ELIRef values are correlated by the ones provided by the ELI during registration.

7.1.3 Phase 1 (X0)

See ETSI TS 104 000 [4], clause 5.1 for a description of this phase.

At some point, the CSP will have received a warrant with some identifier of the target. The CSP enters this, or a translated value of the target into the LICF. This happens asynchronously to the lifecycle of the NFs/ELIs. The TgtID is now present in the ADMF waiting for any NFs/ELIs to make use of it as they come up.

1. The LICF sends the TargetID to the LIPF through the LI-Admf interface.
2. The LIPF turns this value around to the LICREPF in the LISE over the Ne-Admf interface.
3. *[LI STATE]:* The distributed LI state now contains the NFRef and ELIRef in the LIPF (which is locally assigned, to be later mapped to an actual instance), and a target ID each in the LISE and LICF. These target IDs may be translated and not necessarily the same.
4. At some point, independently of the target ID provisioning in the LICF, the OSS starts a new NF. The OSS commands the NWF at the application layer (while at the same time signalling the virtual layer below in 8a.) to start a new NF.
5. The NWF assigns this new NF an NFIID and sends it to the NOGW, along with the NFRef, over some network interface (which is out-of-scope of the present document).
6. The NOGW takes this NFIID and NFRef and crosses it over from the network domain to the LI security domain and signals it to the LIPF through the Li-Ap interface.
7. *[LI STATE]:* The distributed LI state now contains and correlates the NFRef and the NFIID.

8a. At the same time as step 4, the OSS instructs the virtual layer to start a new VNF over the Os-Mano interface.

8b. A copy of the 8a message containing the {VnfID, NFIID, NFRef} is forked (by means out-of-scope of the present document) to the LIGW (the LIGW inspects the Os-Mano interface).

9. MANO starts the new VNF ({VnfID, NFIID, NFRef} containing the embedded ELI.
10. MANO signals the MOGW the start of the VNF with {VnfID, NFIID, NFRef} over the LI-Os interface. Note that this information seems to be duplicative of the information in step 8b, but it is not. Eventually (after steps 11 and 12), the LIPF will have the ability to see what MANO was commanded to do by the OSS, and what MANO has actually done.
11. The MOGW signals the start of the new VNF and its associated {VnfID, NFIID, NFRef} to the LIPF over the Li-No interface.
12. The copy of the OSS command to MANO is now conveyed by the LIGW to the LIPF over the Li-Vn interface. Note that steps 11 and 12 are asynchronous.
13. *[LI STATE]:* The distributed LI state now contains the application layer NFRef, ELIRef and NFIID all associated with the same network function.
14. *[LI STATE]:* The LI system now goes into a wait state, waiting for the ELI of the NF to spin up and call back into the LIPF.

7.1.4 Phase 2 (X0)

See ETSI TS 104 000 [4], clause 5.1 for a description of this phase.

15. The ELI activates. It is aware that it belongs to the VNF with NFIID. It is also pre-provisioned with the necessary network addresses of the LI layer to report to. The ELI generates an X0ID for the ELI associated with this NFIID, and uses it for the attestation interactions, separating them from the EliID.

16. *[LI STATE]:* The distributed ELI state now contains the NFIID in the ELI.

17. The ELI starts attestation. It takes measurements and makes claims, cryptographically signs them, and internally makes these available to the X0-a interface.

18. The ELI sends the claims or measurements, along with the associated NFIID and X0ID, to the AVS in OSS over the X0-a interface.

19. *[LI STATE]:* To establish full trust in the ELI other attestation actions are taken by the relying party. A full vertical assessment (see Annex A of the present document) of the LI network is performed and correlated to the new ELI, including verification of the network connectivity available to the ELI.

20. The AVS is satisfied that the ELI is trustworthy and signals this (along with the NFIID and X0ID) to the ARP in the LISE over the X0-v interface.

7.1.5 Phase 3 (X0)

See ETSI TS 104 000 [4], clause 5.1 for a description of this phase.

21. The Attestation Relying Party (ARP) in the LISE correlates all the information from the attestation steps and makes a decision on the trustworthiness of the new ELI based on the results from the AVS.

22. *[LI STATE]:* The distributed LI network state now adds the X0ID to the NFIID in its ELI state. *While the LISE has been holding the TgtID since it was provisioned in step 2, the TgtID is not assigned to this ELI just yet.*

7.1.6 Phase 4 (X0)

See ETSI TS 104 000 [4], clause 5.1 for a description of this phase.

23. The ELI requests to register to the LISE over the X0-r interface with its X0ID, NFID and ELIRef.

24. *[LI STATE]:* The LISE generates a new EliID for the new ELI, and uses the received X0ID, NFIID, and ELIRef to correlate the request in the LICREPF with the X0ID. *The LICREPF still holds the TgtID, but it is not associated with this ELI yet.*

25. The LISE accepts the registration of the new ELI and sends it its new EliID over the X0-r interface.

26. The LI CMF in the LISE now announces the new ELI to the LICA.

27. A multitude of steps pertaining to certification are hidden here, from the initial CSR to the final certificate enrolment and delivery to the ELI over the X0-cert interface.

7.1.7 Phase 5 (X0)

See ETSI TS 104 000 [4], clause 5.1 for a description of this phase.

28. The X0-c interface between the LISE and the ELI is used to configure X0, as well as X1/X2/X3 interfaces and the configurations of their certificates.

29. *[LI STATE]:* The distributed LI state now contains all necessary IDs needed for provisioning and configuration of all interfaces. *The TgtID is still not correlated with any ELI.*

7.1.8 Phase 6 (X0)

See ETSI TS 104 000 [4], clause 5.1 for a description of this phase.

30. The certificates for the X1/X2/X3 interfaces are distributed to the ELI over the X0-cert interface. This step hides a multitude of interactions for certificate enrolment, signing and distribution.

7.1.9 Phase 7 (X1)

ETSI TS 104 000 [4] stops at phase 6. In the present document, phase 7 now kicks off the X1 (and above) interactions.

The LI network is finally ready to assign TgtIDs to ELIs.

31. The LISE uses the Ne-Admf interface to the LIPF to send the TgtID meant to be distributed to the ELI.
32. The LIPF provisions the ELI with the TgtID over the X1-e interface.
33. The ELI arms itself to isolate and intercept communications associated with the TgtID.
34. *[LI STATE]:* The distributed LI state is now complete, and the LISE can finally associate the TgtID with the provisioned EliID.

8 Further security aspects

8.1 General

Clause 8 examines additional security aspects that need to be addressed when implementing the LI architecture described in the present document.

8.2 Compromise of interface endpoints

8.2.1 Overview

If both ends of an interface do not terminate in a secure enclave, the entire chain may be compromised. As an example, the visibility of X0, X1, X2/X3, HI2/HI3/HI4 interfaces in the network is a good place to start. Assuming basic measures such as TLS are taken, the vulnerability is pushed into the endpoints of the interfaces themselves. If an attacker compromises the endpoint, not just the messaging, the internal state/memory footprint of any code running in the endpoint could be visible unless it runs in a Hardware-Mediated Execution Enclave (HMEE).

8.2.2 Analysis

Using the LI interfaces as an example, figure 8.2.2-1 below identifies network elements susceptible to attack, the attack locations, and objects of attack. As previously stated, NFs are Network Functions, and ELIs are Elements of LI, such as Points of Interception (POIs), Triggering Functions (TFs), and Mediation and Delivery Functions (MDFs) as defined in 3GPP TS 33.127 [6]. Very loosely, a path from the LICF, through NFs, ending at a LEMF, is called a "chain".

There are also hosts running hypervisors, but figure 8.2.2-1 should not be interpreted as strictly applying to Virtual Machine (VM) deployments, but also container deployments such as Kubernetes, in which case the hypervisor metaphor is stretched to the breaking point to mean the management layer, termed, for example, the Container Infrastructure Service Management (CISM). There may also be specific attack locations particular to container deployments.

Further, this analysis abstracts out provisioning (X0/X1) and interception product routing (X2/X3) and delivery (HI2/HI3), but the concepts are general to any interface chain. Exactly what type of information the tunnels carry is irrelevant to this layer of the security analysis.

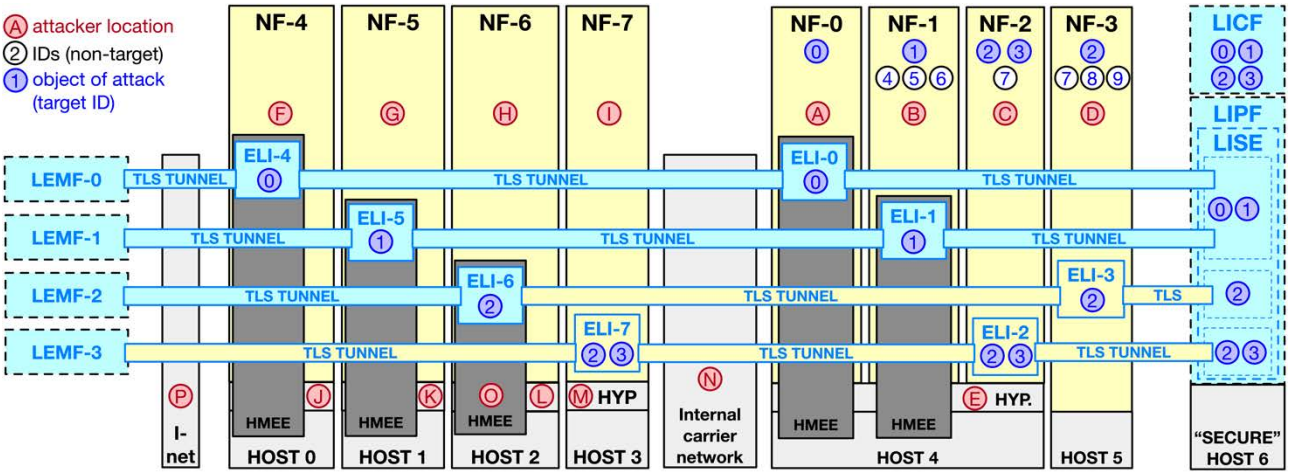

Figure 8.2.2-1: Attack locations and vectors

In figure 8.2.2-1 TLS tunnels are only depicted blue in the figure (secure) if both endpoints terminate in blue functions (ones running in an HMEE). If one endpoint is not running in an enclave, the interface is considered vulnerable.

Annex A (informative): Attestation

A.1 Definition of remote attestation

Remote attestation is a method by which an Attester, which is the hardware or software running on some system, makes statements to a Verifier, which verifies these statements by verifying the statements authenticity and by comparing them to a trusted database of ground truth, and sends the results of this verification to a Relying Party, which is now in a better position to make decisions about trusting the original Attester and to build a database of trust.

An implicit or explicit trust relationship exists between the Verifier and the Relying Party. This is a long-term relationship, secured through more thorough but perhaps slower means. Such relationship is leveraged by the Relying Party to make numerous, fast, automated decisions about short-lived Attesters.

A.2 The simplest attestation flow

The simplest attestation flow is Attester → Verifier → Relying Party.

In the context of the LI architecture, figure A.2-1 below depicts how the Relying Party leverages the Verifier to build trust in Attesters.

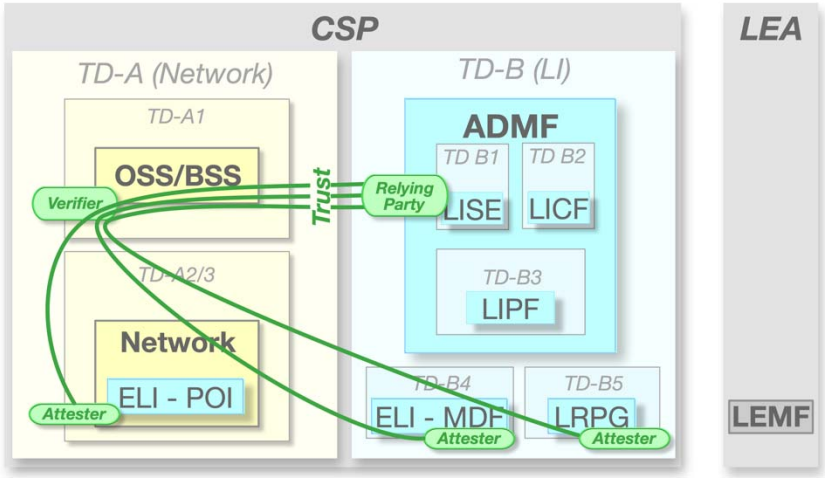

Figure A.2-1: Remote attestation role in the architecture

A.3 A brief overview

A.3.1 Attestation framework

The attestation framework contains three main entities: the relying party, one (or more) verifiers, and a possibly large number of attesters, as defined in IETF RFC 9334 [7]. The idea is simple: the Relying Party builds solid, expensive, slow trust in the Verifier, perhaps even by partially manual procedures, then use cryptography (through stored hashes and signatures) to amplify this trust across a large number of attesters, automatically, fast, and repeatably. This clearly implies a deep trust relationship between the Relying Party and the Verifiers, both of which will likely be found in *the same* Trust Domain.

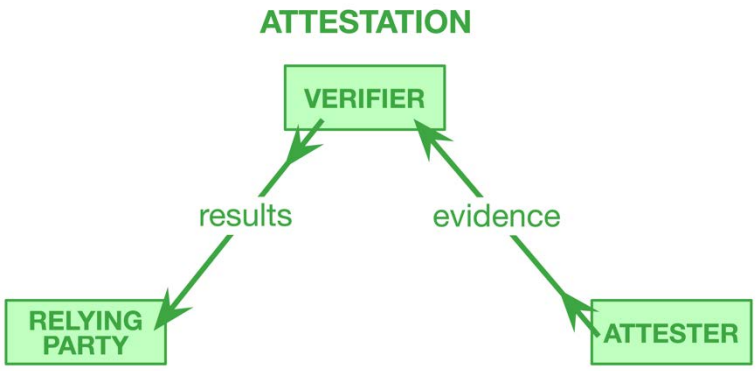

Figure A.3.1-1: RATS architecture

In Lawful Interceptions, the relying party is (part of) the ADMF, the verifiers are part of an attestation environment that includes more than just LI, and the attester is the ELI, as in figure A.3.1-2.

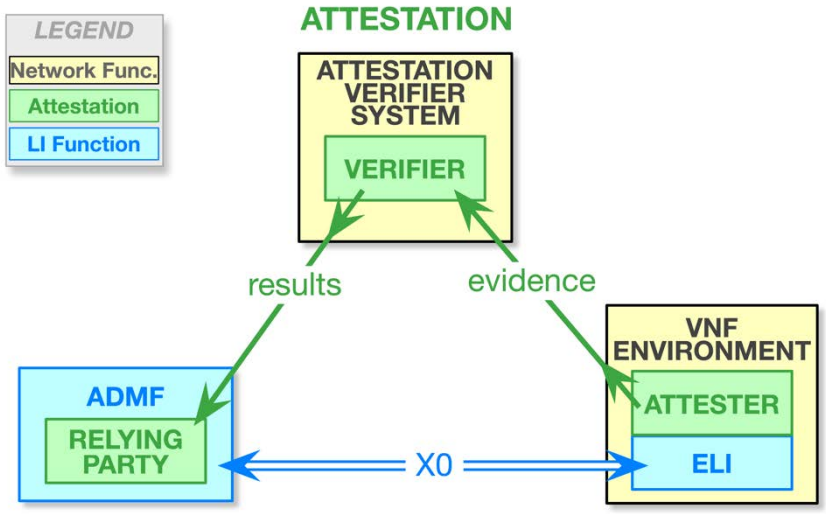

Figure A.3.1-2: LI attestation framework

The Attestation Verifier Service (AVS) performs the verification function by comparing the field information from the attester against stored "ground truth" that is trusted. Remote attestation can be thought of as a trust multiplier or trust amplification system. Slow, expensive, long(er) term trust is built into the ground truth databases in the AVS. This can then be leveraged in an automated, fast fashion to extend trust to fast lifecycle functions in the network. Once necessary and sufficient trust is built the Relying Party in network ELIs, the X1 and higher interfaces that depend on this trust can be started.

A.3.2 Ground truth

"Ground truth" is a term that comes from the field of navigation. There are many situations in which a vessel is dependent on positioning derived from sensors, which are not perfectly accurate and slowly drift, for various reasons. After a long enough period, the position derived from the sensors can be wildly divergent from reality due to the accumulation of errors. Occasionally, the sensor-derived value is synchronized with ground truth obtained by direct observation, for fear that serious accidents could happen.

In attestation, what is considered ground truth are simply previous measurements. Sometimes these are referred to as "golden measurements". The root of trust is the hardware. At the software layer, as it is being onboarded, the carrier verifies its provenance by checking the vendor signatures, it thoroughly tests it, and, before inserting it in the image catalogue of ready to run software, it takes and stores a measurement that is hashed and stored in a secured database of ground truth.

Thereafter, before every bootup (the term bootup is used in the present document to also capture VM launch and container application startup), the state of the image can be checked against this stored measurement and a decision can be made whether to run it, or not, if the image has changed. The system can then trust that it has trusted software running on trusted hardware.

The verifier uses ground truth to verify evidence and provide attestation results to the ADMF relying party. Figure A.3.2-1 depicts the process and entities involved in onboarding provisioning truth in the network.

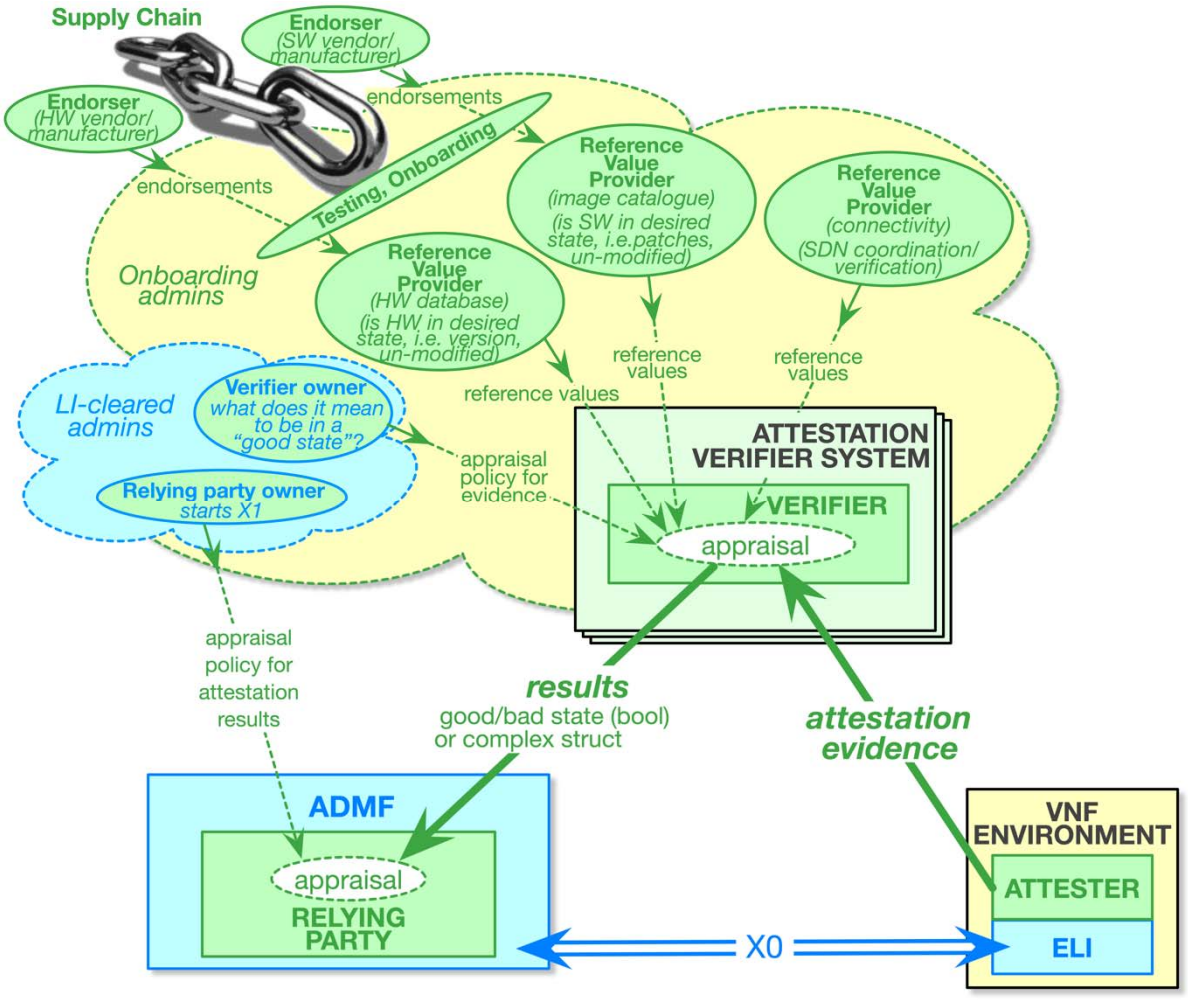

Figure A.3.2-1: Establishing ground truth

In the onboarding process, there is an inherent reliance on the supply chain, both for hardware and software. The hardware manufacturer provides "endorsements" along with the hardware, such that only genuine hardware can prove authenticity by cryptographically matching the previously stored ground truth endorsements.

On the software front, along with the steps described for hardware, the network operator can undertake not only vendor endorsement verification, but a more rigorous testing regime before it enters the vendor-provided software images in a repository of verified software ready to be instantiated into the system. This is also the step in which the network operator makes a choice as to the dynamic flexibility it wishes to impart to the software images it installs in the catalogue. It can choose to make permissive decisions to allow the software to be pointed dynamically to arbitrary Certificate Authorities, internal or external to the network, or more strict decisions and burn in one or more CA certificates. This choice is then burnt in the reference value hash for the image and cannot be changed at instantiation/run time (either from static to dynamic, or from dynamic to static).

This onboarding process builds a database of reference values in the one or more verifiers. These reference values will be used to compare the field data ("attestation evidence") coming from the network to make trust decisions during network function start-up.

While the whole remote attestation process is sensitive, as the health of the whole network relies on it, there is an extra level of sensitivity for LI components. Security-wise, the existence and function of LI code may be sensitive. Of course, at this point, no target IDs or warrant/agency information exists yet. However, the verification policies for evidence and the acceptance of attestation results are controlled by administrators cleared into the LI trust domain. A maximally cautious security stance would have all the onboarding administrators cleared into the LI trust domain.

A.3.3 Attested session creation

The motivation for attestation is to provide an entity (the relying party) with a procedure to verify another entity it needs to interact with. To be effective, the verification through attestation needs to be linked to the subsequent interaction. This implies that the attestation needs to leave information at the relying party not only for verifying the trust posture of the other entity, but also leaves information that the entity can use to secure the interaction with the attested entity.

Typically, this information can consist of identifiers and keys that can be used to create a secure session, e.g. by using mutually authenticated TLS, between the entity that is a relying party and the entity such as the attester. This architecture will use public-key cryptography so the relying party can get an attested public key. This choice avoids the risk of the attesting entity being impersonated in case symmetric cryptography was used. ETSI TS 104 000 [4] explains how the X0 trust relationships are established. A simple flow that links attestation to session configuration is offered below in figure A.3.3-1.

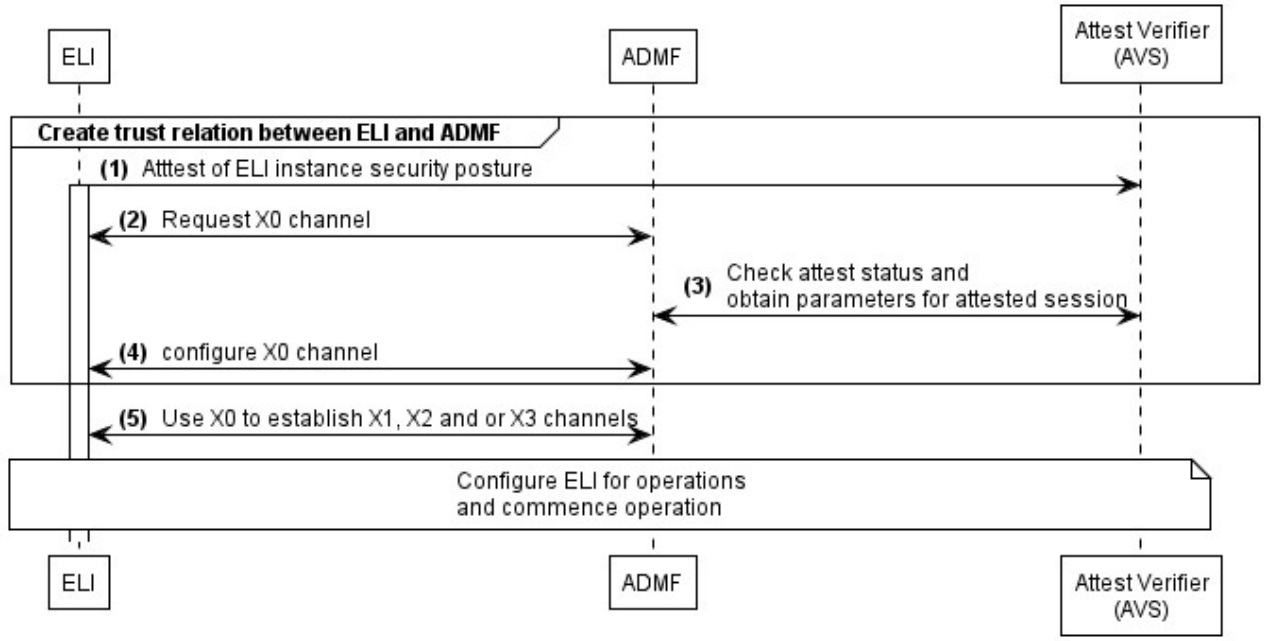

Figure A.3.3-1: Trust establishment between ADMF and ELI

A.3.4 Attester environment

Truth needs to be rooted in hardware, as the bottom-most, least alterable building block of a network. Still, hardware immutability is not necessarily a given, nor is it sufficient, although necessary. The system software running closest to the hardware (microcode, firmware, bootloaders, etc.) is of equal importance as a trust element in the network and also needs to be trusted. Finally, on top of the stack of building blocks, the application code needs to be secured if the information it processes is to remain secure.

Hardware chips have a root key burnt into silicon that is immutable. After the chip leaves the fabrication plant, the private key will never leave the chip, nor can it ever be changed. There is one chance to set the key, and it is at manufacturing time. The chip manufacturer vouches via an issued certificate associated with the key, that the key was securely burned into the chip. The chip can then be challenged at any point with a fresh nonce, and the response can be verified with the chip's public key. Further, the chip can use its private key, or keys derived for the purpose of attestation, to sign claims or measurements from code running close to the silicon, such as microcode, firmware, bootloaders, etc., all the way up to application code running in a secure enclave. This allows a chain of trust of all the low-level software and the hardware to be built all the way up to the application level.

As figure A.3.4-1 shows, each layer combines evidence from the lower layers, stacks its own claims/measurements on top, and sends the newly created evidence upward, eventually to the verifier in the external attestation environment. There is an unfortunate clash of terms here: in the LI world "target" means the identifiers of the individual under surveillance; in the attestation world, it is the object that finds its authenticity questioned. The context should keep the distinction clear.

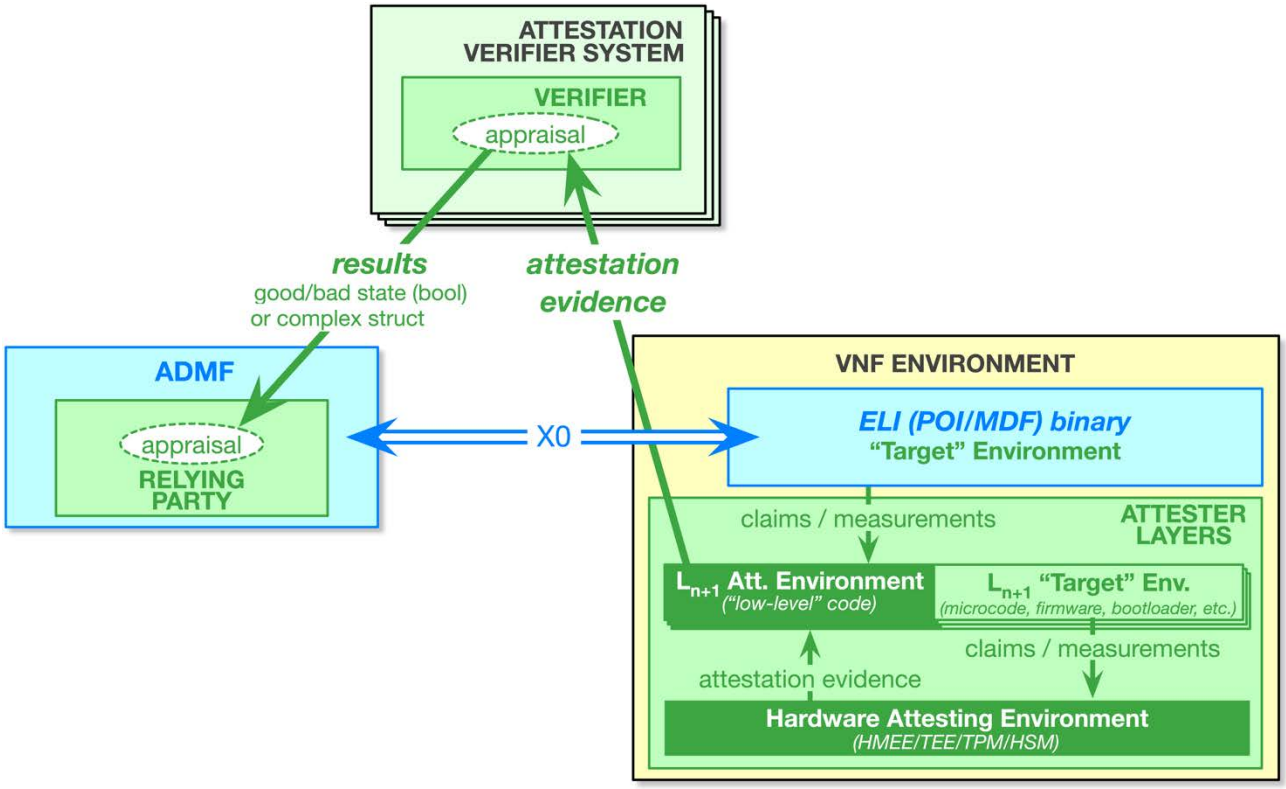

Figure A.3.4-1: Attester environment

At the LI layer, this bottom-up evidence building process eventually culminates, hopefully, in a positive appraisal of the results, and sufficient trust in the ADMF to open a secure X0 channel to the newly booted ELI over which target IDs can be provisioned. The trust established in the X0 channel is subsequently used to open secure channels: X1, X2 and/or X3 channels.

A.3.5 Hardware layers

Figure A.3.5-1 depicts one possible view of the hardware/OS/software stack. It is by no means "standard", but merely an attempt to bring language together from multiple standardization efforts. The terms "secure boot", "trusted boot", and "measured boot" are sometimes used interchangeably, with varying claims of correctness. Historically, Trusted Platform Modules (TPMs) have been used to secure lower layers, and Hardware Mediated Execution Enclaves (HMEEs) have been used to secure higher ones. This has been changing with advancing technology.

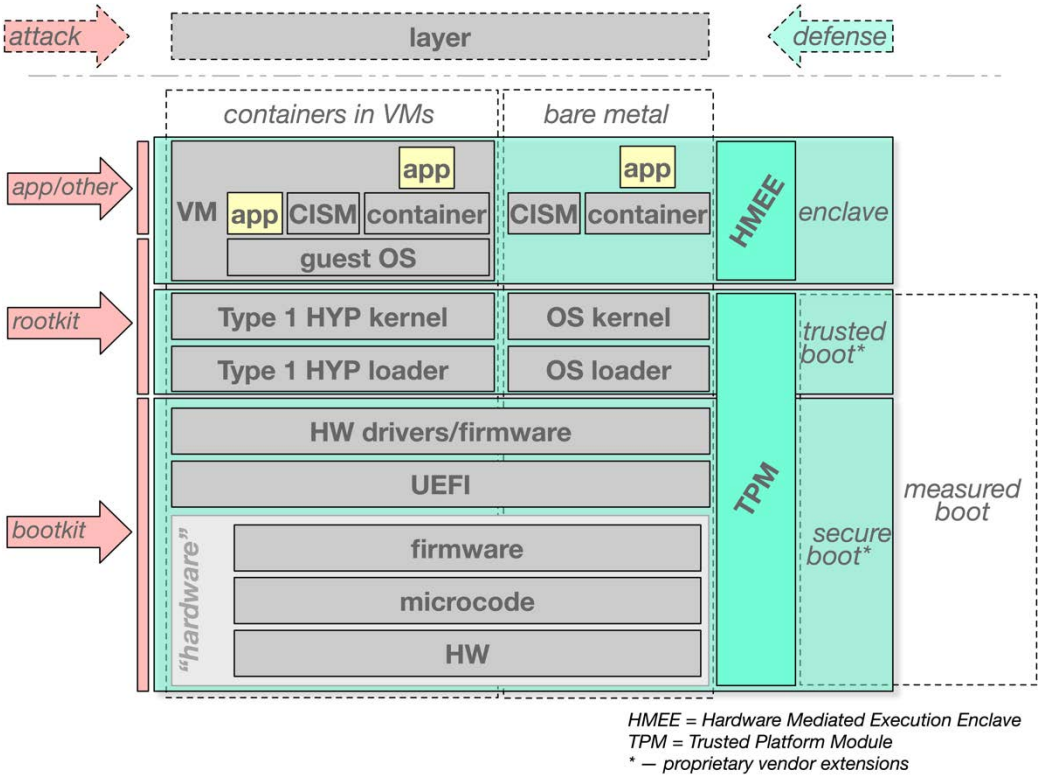

Figure A.3.5-1: Hardware layers

The TPM and HMEE approaches to securing code execution have evolved separately from different paradigms. A TPM is a specialized small-scale cryptographic processor with limited memory registers that is attached to a motherboard and is independent of the main Central Processing Unit(s) (CPUs) of the host. An HMEE is a set of cryptographic instructions that control memory setup and partitioning and it is managed by the CPU itself by executing special hardware instructions managed by the application code running on the CPU.

Traditionally, "measured boot" or "secured boot" are proprietary terms that generally refer to the process of building a pyramid of trust from the hardware, up through microcode, firmware, UEFI, up to the Operating System (OS) or hypervisor loader and kernel. The job of securing application code is left to the HMEE. Depending on whether the deployment is on "bare metal" or atop a guest OS in a hypervisor, the performance demand on HMEE hardware is commensurate with the load required of the CPU enclave. Different hardware manufacturers have taken different approaches to enclave implementation and optimization.

More recent hardware implementations such as "System-on-Chip (SOC)", blur this TPM-HMEE line, as depicted in figure A.3.5-2.

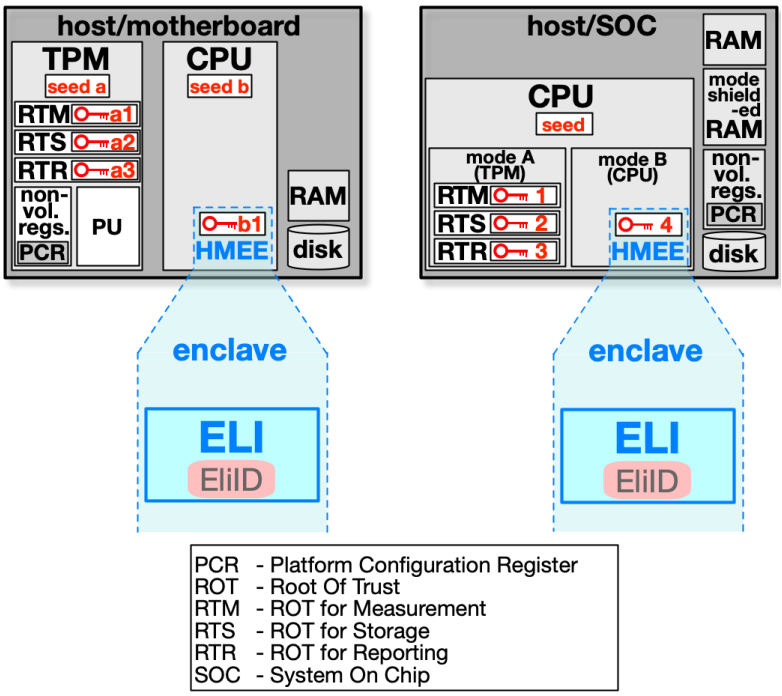

Figure A.3.5-2: Motherboard vs. SOC hardware implementation

An SOC implementation absorbs the function of the traditional motherboard-mounted TPM by using switched execution modes and accessing mode-shielded RAM to perform the same function (storing hashes of known/trusted configurations).

The main difference between the two is that in the TPM/motherboard implementation there are two independent roots of trust that secure the software stack: one based on the seed burned into the TPM, and a second one based on the seed burned in the HMEE. The TPM uses its seed to derive roots of trust (keys) for measurement, storage, and reporting. The HMEE uses its seed to derive the enclave root of trust.

In the SOC paradigm, there is one seed used to derive all other roots of trust, which are kept segregated by hardware mode switching, which simplifies the hardware implementation as well as security management in general.

A.4 Attestation full picture

Figure A.4-1 brings all the attestation elements together for reference.

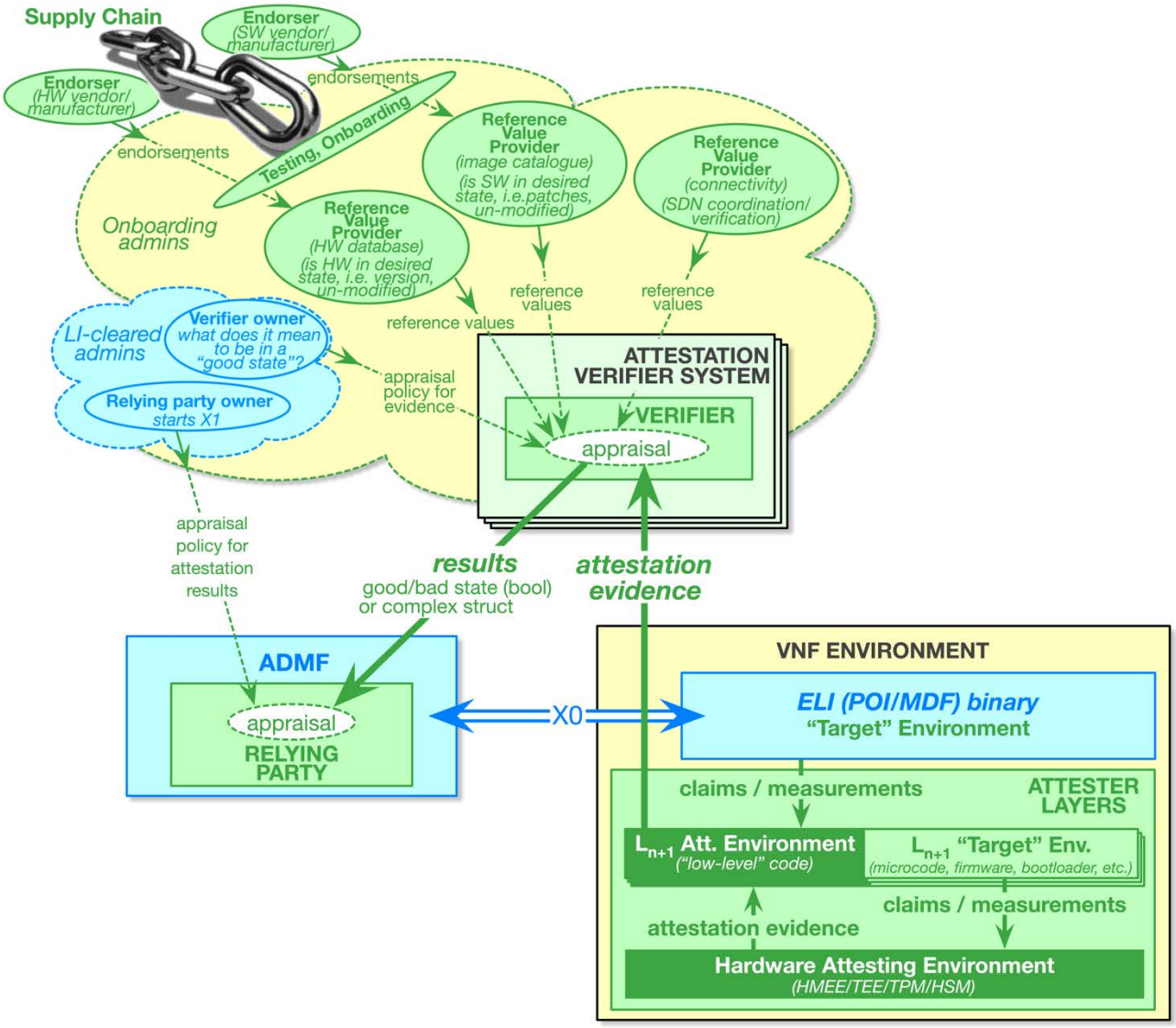

Figure A.4-1: Full attestation picture

Annex B (normative): Checklist

B.1 Purpose

This Annex B provides a checklist for use in assessing whether a given LI architecture or deployment meets the requirements given in the present document.

The reason it is normative is such that each network can be asked and can answer queries in a uniform, traceable fashion.

B.2 Functions and interfaces

B2 - 10 Are the components of the ADMF (LICF, LICA and LIPF) in a separate LI trust domain from the rest of the network?

B2 - 20 If components within the LI trust domain are virtualized, are they virtualized on infrastructure that is segregated or isolated from the rest of the network?

B2 - 30 Are all the interfaces from the ADMF / crossing the LI trust domain to the rest of the network protected by an LIPF or similar security-enforcing function?

B2 - 40 Is there an attestation environment available for use by the LI functions?

B2 - 50 Are all of the logical functions given in table 6.4.2-1 present?

B2 - 60 Are all of the logical interfaces given in table 6.4.3-1 present?

B.3 Provisioning flow

B3 - 10 When a new service chain (i.e. a collection of network functions organized to provide a service) is instantiated, is the ADMF made aware (e.g. via Os-Ma-Nfvo, LI-NO or similar)?

B3 - 20 When a new network function is established, is the ADMF made aware (e.g. via LI-Os or similar)?

B3 - 30 When a new LI function within a network function is established, is the ADMF made aware (e.g. via LI_X0 or similar)?

B3 - 40 Does the LICF require attestation and verification of a new LI function before provisioning over LI_X0?

B.4 Attestation

B4 - 10 Is there an attestation environment available within the network?

B4 - 20 Does the attestation environment contain reference values for regular (non-LI) network functions?

B4 - 30 Is there a process for managing reference values in response to patches to vendor-supplied images, changes in configuration, etc.?

B4 - 40 Is there a process for managing reference values related specifically to images of LI functions?

B4 - 50 Are the LI-specific reference values and the processes for managing them appropriately segregated from non-LI-specific ones?

B4 - 60 Is the attestation process anchored from a hardware root of trust?

B4 - 70 Does the attestation process include each of the layers between the root of trust and the LI application?

B4 - 80 Does the ADMF have a means of verifying that any new ELI has been instantiated on trusted hardware?

B4 - 90 Does the ADMF have a means of verifying that any new virtualized ELI has been instantiated within a trusted virtualization infrastructure?

B4 - 100 Does the ADMF have a means of verifying that a virtualized ELI image / code itself has not been tampered with?

B4 - 110 Does the ADMF use the results of the attestations to determine whether to task newly instantiated LI functions?

B.5 Certificate management

B5 - 10 Does the LI network have a CA available that is under the control of LI administrators (e.g. the LICA)?

B5 - 20 Does the AMDF have a means of providing new certificates to an ELI for use in LI_X1/X2/X3?

B5 - 30 Does each ELI have a means of identifying an initial CA / key material for use in securing any LI_X0 interactions?

B5 - 40 Do communicating endpoints for X0/X1/X2/X3 interfaces reject connection requests from endpoints using certificates not being issued by the LI issuing (sub)CA, or anchored through attestation (for X0 registration)?

B5 - 50 Is the certificate binding for X0 and X1 certificates in place?

B.6 Functional concerns

B6 - 10 What happens in the parent NF if an ELI fails or goes offline for any reason? How tightly are the lifecycles of the ELI and its parent NF integrated?

Annex C (informative): Change history

Status of Technical Specification ETSI TS 104 007 Lawful Interception Architecture		
TC LI approval date	**Version**	**Remarks**
October 2024	1.1.1	First publication of the TS after approval at ETSI TC LI#67 in Vancouver (Canada)

History

Document history		
V1.1.1	November 2024	Publication