ETSI TR 119 540 V1.1.1 (2025-10)

TECHNICAL REPORT

Electronic Signatures and Trust Infrastructures (ESI); Standardization requirements for Smart Contracts based on Electronic Ledgers

Reference
DTR/ESI-0019540

Keywords
digital identity, electronic signature, internet,
security, smart contract, trust services

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 Report (TR) has been produced by ETSI Technical Committee Electronic Signatures and Trust Infrastructures (ESI).

Modal verbs terminology

In the present document "**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.

Introduction

In order to improve the development of solid basis of Smart Contracts standards, three ETSI documents have been developed. Each of them is the outcome of a special phase:

1) A **scoping study phase** analysing the issues on Smart Contracts in particular with respect to the Data Act and eIDAS2 proposals and identifying standardization issues. This work is documented in ETSI TR 119 540 (the present document).

2) A **requirements phase** analysing Policy and Security requirements for with Smart Contracts using Electronic Ledgers. This work is documented in ETSI TS 119 541 [i.12].

3) A **use phase** of EU Regulation on Digital Identity Wallets and electronic signatures for identification with Smart Contracts. This work is documented in ETSI TS 119 542 [i.16].

Smart Contracts based on Electronic Ledgers have been normalized in Regulation (EU) 2023/2854 [i.1].

Electronic Ledgers have been normalized in Regulation (EU) 2024/1183 [i.2].

1 Scope

The present document defines standardization issues for Smart Contracts, as defined in Data Act [i.1], and based on Electronic Ledgers as defined by eIDAS2 [i.2]. It builds on existing and planned standardization and publicly available specifications. It presents a novel and as yet unpublished **Chain of Trust**, by addressing the role of all involved entities in **building**, **deploying**, and **executing** a Smart Contract computer program on an Electronic Ledger. All the relevant actors, artifacts, hardware, networks and tools, are identified by emphasizing the critical points where governance, safety, security, and identity issues are required. The Chain of Trust **will** be extensively translated in suitable recommendations in ETSI TS 119 541 [i.12] and ETSI TS 119 542 [i.16]. The security of Smart Contracts **will** be significantly compromised by an incomplete validation chain, which exposes users to various risks, including fraud and attacks.

NOTE 1: The present document summarizes the results of a **scoping study** that examines the issues for the application of Smart Contracts, particularly in relation to the European frameworks outlined in the Data Act [i.1] and eIDAS2 [i.2] regulations. The goal is to pinpoint standardization issues for Smart Contracts and Electronic Ledgers in data-sharing computer applications. Additionally, the study considers reports and standards from ETSI ISG PDL (at the time of the publication of the present document conveyed into ETSI TC DATA), ETSI TC ESI, and checks consistency with ISO and CEN documents.

NOTE 2: Unless otherwise specified in the present document, the definition of Smart Contracts refers to Regulation (EU) 2023/2854 [i.1] and the definition of Electronic Ledgers, and Qualified Electronic Ledger refer to Regulation (EU) 2022/2065 [i.2]. See Annex C for further details.

The present document is structured as follows:

- Clause 4 **enumerates** the regulations, applied standards, EU initiatives and other activities involved for the use of Smart Contracts in Data Sharing Computer Applications.
- Clause 5 is the **core of the present document**. It identifies the entities and their inter-relations for the creation, validation, deployment and use of Smart Contracts in Electronic Ledgers. A **Chain of Trust** listing the main entities and their relations **will** be presented and discussed; the **Chain of Trust** allows to highlight issues that **will** be focused in the next clause.
- Clause 6 lists in a concise way the issues that are translated into formal requirements in ETSI TS 119 541 [i.12] and ETSI TS 119 542 [i.16].
- Clause 7 concludes.
- Annex A presents four figures showing a particular, fine-grained, implementation of the **Chain of Trust** as presented in Table 1: entities, their relations participating in the design of SC Language, the deployment, and execution of Smart Contracts on a Qualified Electronic Ledger. Other implementations are also possible.
- Annex B **graphically** and **informally** depicts, the **Chain of Trust**, as **formally** described in Table 2.
- Annex C gives a **comparative overview of definitions** in normative and standard documents.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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 the present clause were valid at the time of publication ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonized rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act).

[i.2] Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework (eIDAS2).

[i.3] ISO 22739:2024: "Blockchain and distributed ledger technologies — Vocabulary".

[i.4] ETSI TR 119 001: "Electronic Signatures and Infrastructures (ESI); The framework for standardization of signatures; Definitions and abbreviations".

[i.5] ISO/IEC 15408: "Information security, cybersecurity and privacy protection — Evaluation criteria for IT security".

[i.6] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC (eIDAS).

[i.7] Regulation (EU) No 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

[i.8] ISO 9001:2015: "Quality management systems — Requirements".

[i.9] ISO/IEC 27001:2022: "Information security, cybersecurity and privacy protection — Information security management systems — Requirements".

[i.10] ETSI TR 119 476: "Electronic Signatures and Trust Infrastructures (ESI); Analysis of selective disclosure and zero-knowledge proofs applied to Electronic Attestation of Attributes".

[i.11] Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive).

[i.12] ETSI TS 119 541: "Electronic Signatures and Trust Infrastructures (ESI); Policy and security requirements for Smart Contracts using Electronic Ledgers".

[i.13] ETSI EN 319 401: "Electronic Signatures and Trust Infrastructures (ESI); General Policy Requirements for Trust Service Providers".

[i.14] ETSI EN 319 403-1: "Electronic Signatures and Infrastructures (ESI); Trust Service Provider Conformity Assessment; Part 1: Requirements for conformity assessment bodies assessing Trust Service Providers".

[i.15] ISO/TS 23635:2022: "Blockchain and distributed ledger technologies — Guidelines for governance".

[i.16] ETSI TS 119 542: "Electronic Signatures and Trust Infrastructures (ESI); Use of EU Digital Identity Wallets and electronic signatures for identification with Smart Contracts".

[i.17] Architecture and Reference Framework (ARF) for the European Digital Identity (EUDI) Wallet.

[i.18] ENISA: "Digital Identity Standards publications".

[i.19] ISO 23257:2022: "Blockchain and distributed ledger technologies — Reference architecture".

[i.20] UNCITRAL Model Law on Automated Contracting finalized by the UN Commission on International Trade Law.

[i.21] ISO 24332:2025: "Information and Documentation - Blockchain and distributed ledger technology (DLT) in relation to authoritative records, records systems and records management". Forthcoming.

[i.22] ETSI TR 104 173: "Data Solutions (DATA); Oracles for Smart Contracts executed in Electronic Ledgers". Forthcoming.

[i.23] ETSI TS 104 172: "Data Solutions (DATA); ETSI Specification of the Requirements on Ledgers and Smart Contracts". Forthcoming.

[i.24] The ROCQ theorem prover.

[i.25] The Isabelle theorem prover.

[i.26] The Lean theorem prover.

[i.27] X. Leroy: "Formal verification of a realistic compiler". Communication of the ACM. Vol 52, pp.107-115, 2009.

[i.28] CEN-CENELEC White paper: "Blockchain and Distributed Ledger Technologies. Recommendations for Successful Adoption in Europe of Emerging Technical Standards on Distributed Ledger/Blockchain Technologies". 2018.

[i.29] Recommendation ITU-T F.751.0: "Requirements for Distributed Ledger Systems".

[i.30] Recommendation ITU-T F.751.8: "Technical framework for distributed ledger technology (DLT) to cope with regulation".

[i.31] Recommendation ITU-T X.1401: "Security threats to distributed ledger technology".

[i.32] Recommendation ITU-T X.1402: "Security framework for distributed ledger technology".

[i.33] Recommendation ITU-T X.1403: "Security guidelines for using distributed ledger technology for decentralized identity management".

[i.34] Recommendation ITU-T X.1412: "Security requirements for smart contract management based on the distributed ledger technology".

[i.35] ETSI GR PDL 001: "Permissioned Distributed Ledger (PDL); Landscape of Standards and Technologies".

[i.36] ETSI GR PDL 002: "Permissioned Distributed Ledger (PDL); Applicability and compliance to data processing requirements".

[i.37] ETSI GR PDL 003: "Permissioned Distributed Ledger (PDL); Application Scenarios".

[i.38] ETSI GR PDL 004: "Permissioned Distributed Ledgers (PDL); Smart Contracts; System Architecture and Functional Specification".

[i.39] ETSI GS PDL 005: "Permissioned Distributed Ledger (PDL); Proof of Concepts Framework".

[i.40] ETSI GR PDL 006: "Permissioned Distributed Ledger (PDL); Inter-Ledger interoperability".

[i.41] ETSI GR PDL 008: "Permissioned Distributed Ledger (PDL); Research and Innovation Landscape".

[i.42] ETSI GR PDL 009: "Permissioned Distributed Ledger (PDL); Federated Data Management".

[i.43] ETSI GR PDL 010: "PDL Operations in Offline Mode".

[i.44] ETSI GS PDL 011: "Permissioned Distributed Ledger (PDL); Specification of Requirements for Smart Contracts' architecture and security".

[i.45] ETSI GS PDL 012: "Permissioned Distributed Ledger (PDL); Reference Architecture".

[i.46] ETSI GS PDL 013: "Permissioned Distributed Ledger (PDL); Supporting Distributed Data Management".

[i.47] ETSI GR PDL 014: "Permissioned Distributed Ledger (PDL); Study on non-repudiation techniques".

[i.48] ETSI GS PDL 015: "Permissioned Distributed Ledger (PDL); Reputation management".

[i.49] ETSI GR PDL 017: "Permissioned Distributed Ledger (PDL); Application of PDL to Amended Regulation 910/2014 (eIDAS2) Qualified Trust Services".

[i.50] ETSI GR PDL 018: "Permissioned Distributed Ledger (PDL); Redactable Distributed Ledgers".

[i.51] ETSI GR PDL 019: "PDL Services for Decentralized Identity and Trust Management".

[i.52] ETSI GR PDL 020: "Permissioned Distributed Ledger (PDL); Wireless Consensus Network".

[i.53] ETSI GR PDL 021: "Permissioned Distributed Ledgers (PDL); Overview of use cases in 3GPP network and impact analysis on architecture integration".

[i.54] ETSI GS PDL 022: "Permissioned Distributed Ledgers (PDL); PDL in Wholesale Supply Chain Management".

[i.55] ETSI GS PDL 023: "PDL service enablers for Decentralized Identification and Trust Management".

[i.56] ETSI GS PDL 024: "Permissioned Distributed Ledgers (PDL); Architecture enhancements for PDL service provisioning in telecom networks".

[i.57] ETSI GS PDL 025: "Permissioned Distributed Ledger (PDL); Wireless Consensus Network Composition and Organization".

[i.58] ETSI GS PDL 026: "Permissioned Distributed Ledgers (PDL); PDL in Settlement of Usage-Based Services".

[i.59] ETSI GS PDL 027: "Permissioned Distributed Ledger (PDL); Self-Sovereign Identity (SSI) in telecom networks".

[i.60] ETSI GS PDL 028: "Permissioned Distributed Ledger (PDL); Specification utilizing PDL to Standardized IoT Service Layer Platform oneM2M".

[i.61] ETSI GS PDL 029: "Permissioned Distributed Ledger (PDL); Distributed Autonomous Organization (DAO)".

[i.62] ETSI GS PDL 030: "Permissioned Distributed Ledger (PDL); Trust in Telecom System".

[i.63] ETSI GS PDL 031: "Permissioned Distributed Ledger (PDL); Energy Consumption Data Sharing based on PDL Service".

[i.64] ETSI GS PDL 032: "Permissioned Distributed Ledger (PDL); Artificial Intelligence for Permissioned Distributed Ledger".

[i.65] ETSI GS PDL 033: "Permissioned Distributed Ledger (PDL); Smart Contracts; System Architecture and Functional Specification".

[i.66] ISO/IEC 22123-2:2023: "Cloud computing - Part 1: Vocabulary".

[i.67] IEEE 1934™-2018: "Standard for Adoption of OpenFog Reference Architecture for Fog Computing".

[i.68] Regulation (EU) 2019/881 of the European Parliament and of the Council of 17 April 2019 on ENISA (the European Union Agency for Cybersecurity) and on information and communications technology cybersecurity certification and repealing Regulation (EU) No 526/2013 (Cybersecurity Act).

[i.69] Commission Implementing Regulation (EU) 2024/482 of 31 January 2024 laying down rules for the application of Regulation (EU) 2019/881 of the European Parliament and of the Council as regards the adoption of the European Common Criteria-based cybersecurity certification scheme (EUCC).

[i.70] ISO/IEC 24760-1:2025: "Information security, cybersecurity and privacy protection — A framework for identity management. Part 1: Core concepts and terminology".

[i.71] ISO/IEC 29115:2013: "Information technology — Security techniques — Entity authentication assurance framework".

[i.72] Ethereum®: "ERC-721 Non-Fungible Token Standard".

[i.73] ISO 20022: "Universal financial industry message scheme".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

algorithm: set of rules and non-ambiguous procedures to solve a class of problems

Chain of Trust: trust needs of legal or natural persons, as used in Regulation (EU) 2024/1183 [i.2], and of the relationships existing among them

Deontic Logic: philosophical logic that is concerned with obligation, permission, optional, non-optional, obligatory, must, the least one can do, better than, ought, blame, responsibility, indifferent, and related concepts

distributed ledger: ledger that is shared across a set of Distributed Ledger Technology (DLT) nodes and synchronized between the DLT nodes using a consensus mechanism

NOTE 1: According to ISO 22739 [i.3].

NOTE 2: A distributed ledger as defined in ISO 22739 [i.3] **is a special kind** of an Electronic Ledger: the vice-versa is not true.

Electronic Ledger: sequence of electronic data records, ensuring the integrity of those records and the accuracy of the chronological ordering of those records

NOTE 1: According to Article 3(52) in Regulation (EU) 2024/1183 [i.2].

NOTE 2: From Regulation (EU) 2024/1183 [i.2], Recital (68):

- "*This Regulation should ensure technological neutrality, namely neither favoring, nor discriminating against, any technology used to implement the new trust service for electronic ledgers*".
- "*The process of creating and updating an electronic ledger depends on the type of ledger used, namely whether it is centralized or distributed*.".

NOTE 3: The definition of Electronic Ledger in Regulation (EU) 2024/1183 [i.2] **is more general** than the definition of distributed ledger in ISO 22739 [i.3].

Qualified Electronic Ledger: Electronic Ledger provided by a qualified trust service provider and which meets the requirements laid down in Article 45l

NOTE: According to Article 3/53 in Regulation (EU) 2024/1183 [i.2].

SC Byte Code: computer program, written in SC Byte Code Language, that is executed on the top of a SC Virtual Machine and that is produced by a compilation of a SC Source Code

NOTE: It **should** correspond to the definition of Smart Contract in [i.1].

SC Byte Code Language: domain specific language for executing Smart Contracts

SC Compiler: computer program, written in any programming language, translating every SC Source Code, eventually annotated with SC Legal Text, into a semantically equivalent, machine-readable SC Byte Code, and some auxiliary files

NOTE 1: A compilation of a SC Source Code by a SC Compiler **should** produce a number of files that **can** be packaged in a suitable SC Package.

NOTE 2: As an explanatory example, see the number of parameters of the Solidity SC Compiler.

```
solc -o <output_directory> --bin --abi --
devdoc --userdoc --metadata --ast-compact-json
--opcodes --asm <input_file> --overwrite
```

producing a number of files with the following extensions

→.bin→→→→the SC Byte Code
→.abi→→→→the SC Interface
→.docdev→→the developer comments
→.docuser→→the user comments
→.evm→→→→the readable SC Byte Code with comments
→.opcode→→the readable Byte Code without comment
→.json→→→→the AST

NOTE 3: It **should** be open source.

SC Compiler Policy: set of rules to be respected by a SC Compiler

SC Compiler Publisher: legal or natural persons responsible to sign the SC Compiler and the SC Compiler Policy, produced by the SC Compiler Team

SC Compiler Team: legal or natural persons that produce a SC Compiler

SC Deployer: legal or natural persons identified by the Electronic Ledger, in charge of putting a SC Byte Code into the Electronic Ledger

SC Deployer Policy: set of rules and non-ambiguous procedures to be respected by a SC Deployer

SC Development Policy: set of rules and non-ambiguous procedures to be respected in order to produce a SC Package

SC Development Team: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], responsible to produce a SC Package

SC Documentation: documentary information in support of the Smart Contract

NOTE: Elements of the documentation **can/may be** produced by a compilation of a SC Source Code and **can/may** also include the policy documentation and the identity documentation.

SC Execution Report: signed evidence of an execution of a Smart Contract in an Electronic Ledger

SC Language: domain specific language for defining Smart Contracts

SC Language Publisher: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], signing the SC Language Specifications and the SC Language Specification Policy produced by the SC Language Specification Team, and also responsible to sign the SC Compiler and the SC Virtual Machine, produced by the SC Compiler Team and SC Virtual Machine Team, respectively

SC Language Specification: syntax, semantic, and run-time execution model of a domain specific language for defining Smart Contracts

NOTE 1: The SC Language Specification consist of:

1) a SC Source Code Language syntax, written in Backus-Naur Form grammar format;

2) a SC Source Code Language semantic, written in English prose or in formal system (lambda-calculus, term rewriting systems) for expressing computations, and usually referred as the semantic of the SC Language;

3) a SC Byte Code Language syntax;

4) a SC Byte Code Language semantic, written in English prose or in formal system (stack and store reduction semantics) for expressing computations, and usually referred as the execution or run-time environment of the SC Language; this is usually referred as SC Virtual Machine specification;

5) an algorithmic transaction of a computer program, written in a SC Source Code Language into a semantically equivalent computer program, written in a SC Byte Code Language; this is usually referred as SC Compiler specification.

NOTE 2: It **should** be open access.

SC Language Specification Policy: set of rules to be respected by a SC Language Specification

NOTE: It **should** be open access.

SC Language Specification Team: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], responsible to produce a SC Language Specification and a SC Language Specification Policy

SC Legal Team: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], responsible to audit the SC Source Code and or the SC Byte Code using a fixed SC Compiler and SC Virtual Machine, and to produce a SC Legal Text that meets the SC Development Policy

SC Legal Text: legal text attached or annotated into either a SC Source Code and a SC Byte Code assessing legal basis, legal requirements, legal obligations, legal use, legal rights, legal certainty, legal status and legal value to a Smart Contract

NOTE: It **should** refer a SC Compiler and SC Virtual Machine.

SC Oracle: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], that produces external data to a Smart Contract stored in an identified Electronic Ledger so triggering Electronic Transactions

SC Package: set of files, such as SC Source Code, SC Byte Code, SC Legal Text, and any other SC Documentation in support of the Smart Contract, signed by the SC Publisher

SC Provider: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], responsible for providing and the execution of a Smart Contract to a SC User

NOTE 1: The SC Provider **may** offer or trade a Smart Contract with a SC User.

NOTE 2: A SC Provider **can** take input from external sources other than SC User.

NOTE 3: The SC Provider **can** be a "Vendor of applications using Smart Contracts", as defined in Article 36 Regulation (EU) 2023/2854 [i.1].

SC Provider Policy: policy (or policies) governing the behaviour or the SC Provider

SC Publisher: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], responsible to sign the SC Legal Text, the SC Source Code, the SC Byte Code, and the SC Documentation, produced by the SC Development Team, using the SC Compiler

SC Publisher Policy: policy (or policies) governing the behavior or the SC Provider

SC Source Code: computer program, written in SC Source Code Language, defining the behavior of a Smart Contract

NOTE: A SC Source Code is translated using a SC Compiler into a semantically equivalent SC Byte Code, written in a SC Byte Code Language.

SC User: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], that uses services produced by Smart Contracts, provided by an identified SC Provider by accepting SC Legal Text agreements and SC Provider agreements and uses Smart Contracts to put Electronic Records into an Electronic Ledger

NOTE 1: A SC User **can** be a user of a "connected product or related service", as defined in Regulation (EU) 2023/2854 [i.1].

NOTE 2: A SC User **can** be a DLT User, as defined in ISO 22739 [i.3].

SC Virtual Machine: computer program, written in any programming language, executing as input a SC Byte Code and producing as output records that **will** be stored into the Electronic Ledger

NOTE: It **should** be open source.

SC Virtual Machine Policy: set of rules and non-ambiguous procedures to be respected by a SC Byte Code

NOTE: It **should** be open source.

SC Virtual Machine Team: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], responsible to produce a SC Virtual Machine

SC Virtual Machine Publisher: legal or natural person as used in Regulation (EU) 2024/1183 [i.2], responsible to sign the SC Virtual Machine and the SC Virtual Machine Policy, produced by the SC Virtual Machine Team

Smart Contract: computer program used for the automated execution of an agreement or part thereof, using a sequence of electronic data records and ensuring their integrity and the accuracy of their chronological ordering

NOTE 1: According to Article 2/39, 104 in Regulation (EU) 2023/2854 [i.1].

NOTE 2: As per ISO 22739 [i.3]: A *"smart contract is a computer program stored in a distributed ledger technology (DLT) system wherein the outcome of any execution of the program is recorded on the distributed ledger"*.

NOTE 3: The definition of Smart Contract in Regulation (EU) 2023/2854 [i.1] **is more general** than the definition of Smart Contract in ISO 22739 [i.3].

Smart Legal Contract: Smart Contract with legal relevance obtained by embedding or by pointing a SC Legal Text

3.2 Symbols

Void.

3.3 Abbreviations

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

AI	Artificial Intelligence
API	Application Public Interface
ARF	Architecture and Reference Framework
CA	Certificate Authority
DAG	Directed Acyclic Graph
dAPP	distributed Application
DID	Decentralized Identity
DLT	Distributed Ledger Technology
DPoS	Delegated Proof-of-Stake
EAA	Electronic Attestations of Attributes
EAA-Pub	Electronic Attestations of Attributes Public
EAL	Evaluation Assurance Level
EBSI	European Blockchain Services Infrastructure
EDIC	European Digital Infrastructure Consortium
eID	electronic Identification
ENISA	European Union Agency for Cybersecurity
eSIM	electronic Subscriber Identity Module

EU	European Union
EUDI	European Digital Identity
EUDIW	European Digital Identity Wallet
EVM	Ethereum Virtual Machine
GDPR	General Data Protection Regulation
HSM	Hardware Security Module
INATBA	International Association for Trusted Blockchain Applications
IoT	Internet of Things
IPFS	InterPlanetary File System
ISO	International Organization for Standardization
KYC	Know your Customer
mobile-ID	Mobile Digital Signature
NFC	Near Field Communication
NFT	Non-Fungible Token
NIST	National Institute of Standards and Technology
PID	Person Identification Data
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PoS	Proof of Stake
PoW	Proof of Work
QEAA	Qualified Electronic Attestations of Attributes
QES	Qualified Electronic Signature
QTSP	Qualified Trust Service Provider
SC	Smart Contract
SIM	Subscriber Identity Module
SPV	Simplified Payment Verification
TSP	Trust Service Provider
UTXO	Unspent Transaction Output
VM	Virtual Machine

4 Smart Contracts related regulation, standardization and initiatives

4.1 Essential Overview

The present clause presents an overview of all relevant European Union Regulations, Standards, Projects, and other activities involving Smart Contracts and Electronic Ledgers in a neutral and agnostic manner. For each of these activities, the present document proceeds as follows:

- **Essential Overview:** Provide an extended abstract of the activities, tailored specifically to Smart Contract and Electronic Ledgers.
- **Terminology:** Identify main entities and relationships among them, as defined in Clause 3.1.
- **Chain of Trust:** As defined in Clause 3.1, tailored specifically to Smart Contract and Electronic Ledgers.

NOTE: The reviewed material does not claim to be comprehensive but has been selected to give as complete an overview as possible.

4.2 Regulations

4.2.1 Data Act

4.2.1.1 Essential Overview

In addressing the definition of a Smart Contract, the following objectives **can** be derived directly from the definition in Regulation (EU) 2023/2854 [i.1] *"a computer program used for the automated execution of an agreement or part thereof, using a sequence of electronic data records and ensuring their integrity and the accuracy of their chronological ordering"* and the wider application of that definition to that of a contract *"an agreement that is intended to be enforceable by law* and to the execution of a contract *"the process of finalizing a legally binding contractual agreement between two or more parties and committing to the terms contained within that contract"*.

1) The automated execution of an agreement, or part thereof, represents the intended agreement of the parties.
2) The parties of the agreement **can** be correctly identified in case of legal dispute.
3) The recording of the sequence electronic records representing the agreement is maintained in a way which ensures their integrity and the accuracy of their chronological ordering.
4) A party of an agreement **cannot** later deny the agreement.
5) Privacy of sensitive information is maintained. This **can** include information in the data records and identities the parties of the agreement.

The elements defined in the Data Act **can** be bound to a governance framework for identity (see Regulation (EU) 2024/1183 [i.2] to enable strict conformance to item 2).

In addition, it is recognized that Smart Contracts are, implicitly, required to be transparent and explicable, arising from both items 1 and 2 above wherein the parties are able to agree that the Smart Contract is the intended agreement of the parties. It is noted that the identities of the parties to the agreement are only required to be identified by 3rd parties in the case of legal dispute and in accordance with item 5 it is reasonable to treat the identity of parties to the agreement as private.

4.2.1.2 Terminology

Smart Contracts, Electronic Ledgers.

4.2.1.3 Chain of Trust

Regulation (EU) 2023/2854 [i.1] is agnostic with respect to the **Chain of Trust**, and in particular with the production of Smart Contracts.

4.2.2 eIDAS2

4.2.2.1 Essential Overview

The Electronic Identification, Authentication, and Trust Services Regulation (eIDAS) was first published in 2014 to provide a standardized framework across the European Union for electronic identification (eID), electronic signatures, and trust services. The aim was to enable secure and seamless digital transactions across EU member states. The eIDAS2 Regulation [i.2], published in 2024, amends the original regulation, addressing some of its limitations and introducing significant new features to adapt to the evolving digital landscape.

While eIDAS laid the foundation for cross-border digital identification and trust services in the EU, Regulation (EU) 2024/1183 [i.2] significantly expands and modernizes the framework. The key innovation is the European Digital Identity Wallet (EUDIW), which gives citizens more control over their personal data, enhances security, and ensures that both the public and private sectors embrace digital identities. This evolution reflects the increasing need for secure, user-controlled, and interoperable digital solutions across Europe.

eIDAS2 does not address Smart Contracts *in solo*, but a Smart Contract as defined by the Data Act [i.1] **may** use elements of eIDAS2 [i.2] such as Electronic Ledgers that are cited in the Data Act.

eIDAS2 regulation defines Electronic Ledgers as given below.

The definition of Electronic Ledgers in Article 3:

> *"(52) "electronic ledger" means a sequence of electronic data records, ensuring the integrity of those records and the accuracy of the chronological ordering of those records;"*

This definition matches the definition of Smart Contracts in Regulation (EU) 2023/2854 [i.1] for the use of:

> *"a sequence of electronic data records and ensuring their integrity and the accuracy of their chronological ordering";*

Section 11, Article (45k) defines the legal effects of Electronic Ledgers:

> *"1. An electronic ledger shall not be denied legal effect or admissibility as evidence in legal proceedings solely on the grounds that it is in an electronic form or that it does not meet the requirements for qualified electronic ledgers.*
>
> *2. Data records contained in a qualified electronic ledger shall enjoy the presumption of their unique and accurate sequential chronological ordering and of their integrity."*

and Article (45l) defines following specific requirements for Qualified Electronic Ledgers:

> *"(a) they are created and managed by one or more qualified trust service providers;*
>
> *(b) they establish the origin of data records in the ledger;*
>
> *(c) they ensure the unique sequential chronological ordering of data records in the ledger;*
>
> *(d) they record data in such a way that any subsequent change to the data is immediately detectable, ensuring their integrity over time."*

4.2.2.2 Terminology

Electronic Ledgers.

4.2.2.3 Chain of Trust

Regulation (EU) 2024/1183 [i.2], as per the publication date of the present document, is agnostic with respect to the Smart Contracts and the **Chain of Trust**. This **can** change in the forthcoming eIDAS2 Implementing Acts.

4.2.3 GDPR

4.2.3.1 Essential Overview

The General Data Protection Regulation (GDPR) [i.7] is a comprehensive legal framework established by the European Union to safeguard the personal data of individuals within the EU. It sets stringent rules for data privacy, ensuring that personal data is collected, processed, and stored with a high degree of transparency, security, and accountability. Regulation (EU) No 2016/679 [i.7] applies to all organizations that handle the personal data of EU residents, regardless of the organization's location, and imposes significant penalties for non-compliance.

Smart Contracts **can** potentially support Regulation (EU) No 2016/679 [i.7] compliance by providing automated, transparent, and secure mechanisms for handling personal data, aligning with the regulation's requirements. One of the key ways Smart Contracts **can** assist is by automating consent management. They **can** store and track user consent in a tamper-proof manner on a ledger ensuring transparency and that personal data is only processed in accordance with the

agreed-upon terms. This automation **can** include limiting data usage to specific purposes and ensuring consent is periodically updated or revoked, when necessary, all of which enhances compliance with Regulation (EU) No 2016/679 [i.7] focus on individual control over personal data.

4.2.3.2 Terminology

Not applicable.

4.2.3.3 Chain of Trust

Regulation (EU) No 2016/679 [i.7] is agnostic with respect to Smart Contracts, Electronic Ledgers and the **Chain of Trust**.

4.2.4 UNCITRAL model law on automated contracting

4.2.4.1 Essential Overview

The UNCITRAL Model Law [i.20] provides a legal framework to enable the use of automation in international contracts, including through the deployment of artificial intelligence techniques and Smart Contracts, as well as in machine-to-machine transactions. It is intended to complement and supplement existing laws on electronic transactions, in particular those based on other UNCITRAL electronic commerce texts, which have been enacted in over one hundred jurisdictions worldwide. The Model Law is the first legislative text to result from exploratory work conducted by UNCITRAL on legal issues related to the digital economy and digital trade, with work on data contracts and distributed ledger technology as described in ISO 22739 [i.3].

4.2.4.2 Terminology

Smart Contracts.

4.2.4.3 Chain of Trust

The UNCITRAL Model Law [i.20] is agnostic with respect to the **Chain of Trust**.

4.3 Standardization

4.3.1 ISO/TC 307

4.3.1.1 Essential Overview

The scope of ISO/TC 307 reads: *"standardisation of blockchain technologies and distributed ledger technologies"*. Blockchain technology holds immense promise to revolutionize not only the financial domain, but a whole host of things from societal inclusion to efficiencies in government, health and all areas of business. ISO/TC 307, blockchain and distributed ledger technologies, has been set up to meet the growing need for standardization in this area by providing internationally agreed ways of working with it to improve security, privacy and facilitate worldwide use of the technology through better interoperability. This is especially relevant due to the number of enterprises, across various sectors, that are developing blockchain and distributed ledger technologies as a product. The standardization work of ISO/TC 307 has been divided into six groups, namely Foundations (WG1), Security, privacy and identity (WG2), Smart Contracts and their applications (WG3), Use cases (WG4); Governance (WG5), and Interoperability (WG6). The need for collaboration and cooperation has been identified and ISO/TC 307 is liaising with other organizations like ETSI (namely ETSI TC ESI, TC DATA), ISO and IEC committees, as well as external organizations, to minimize any overlap. ISO/TC 307 produced (among many) the following standard specifications and technical reports: ISO 22739 [i.3], ISO/TS 23635 [i.15], ISO 23257 [i.19], and ISO 24332 [i.21].

ISO 22739 [i.3] defines a vocabulary for Blockchain and distributed ledger technologies; ISO/TS 23635 [i.15] defines guidelines for governance defined blockchain and distributed ledger technologies. ISO 23257 [i.19] defines a reference architecture for distributed ledger technology systems including blockchain systems. The reference architecture addresses concepts, cross-cutting aspects, architectural considerations, and architecture views, including functional components, roles, activities, and their relationships for blockchain and distributed ledgers. ISO 24332 [i.21] analyses

challenges, considerations, and potential benefits of blockchain and distributed ledger technology in relation to records management standards and related standards for systems that create records that are required to be authoritative records; **can** be used as records systems; or **can** be used for records management, including records controls.

4.3.1.2 Terminology

Smart Contracts and distributed ledgers as defined in ISO 22739 [i.3].

4.3.1.3 Chain of Trust

ISO 22739 [i.3], ISO/TS 23635 [i.15], and ISO 23257 [i.19] are agnostic with respect to the **Chain of Trust**. However, ISO/TC 307 in ISO/TS 23635 [i.15] discuss some trust requirements on (qualified) DLT systems.

4.3.2 CEN/CENELEC/JTC 19

4.3.2.1 Essential Overview

CEN/CLC/JTC 19 "Blockchain and distributed ledger technologies" was established based on the recommendations presented in the CEN-CENELEC White Paper [i.28] in 2018 on distributed and ledger technologies. It works in close contact with ISO/TC 307 "Blockchain and distributed ledger technologies".

It established the following WGs with the given scope and work items: WG1 (development of standard for policy and security requirements for trust services providing Electronic Ledger services; standardization on functional and interoperability requirements for decentralized identifier and decentralized identity management where distributed ledger is only one possible infrastructure), WG2 (environmental and sustainability classification methodology of consensus mechanisms of blockchain and distributed ledger technologies); WG3 (development of standards for privacy in distributed ledger technologies to ensure compliance to GDPR [i.7] requirements).

CEN/CLC/JTC 19 adopted ISO TC 307 vocabulary [i.3] directly into European Framework. CEN/CLC/JTC 19 considers ISO TC 307 documents [i.15], [i.19], and [i.21] as relevant basements for the CEN Project on Policy and security requirements for trust services providing ledger services and are so participating to a European standard framework for Electronic Ledgers.

4.3.2.2 Terminology

Distributed ledgers and Smart Contracts as defined in ISO 22739 [i.3].

4.3.2.3 Chain of Trust

The technical body CEN/CENELEC/JTC 19 "Blockchain and Distributed Ledger Technologies" is agnostic with respect to the **Chain of Trust**.

4.3.3 ETSI ISG PDL

4.3.3.1 Essential Overview

The ETSI Industry Specification Group on Permissioned Distributed Ledger (ETSI ISG PDL), at the time of the publication of the present document, conveyed into the new ETSI TC DATA, analyses and provides the foundations for the operation of permissioned distributed ledgers, with the ultimate purpose of creating an open ecosystem of industrial solutions to be deployed by different sectors, fostering the application of these technologies, and therefore contributing to consolidate the trust and dependability on information technologies supported by global, open telecommunications networks. The group puts its focus on addressing infrastructure and operational aspects that are not currently covered by previous or parallel standardization activities. In addition to that, ETSI ISG PDL fosters industry convergence towards shared standards with the intent of avoiding duplication and contradicting publications.

The ETSI ISG PDL started from already available experiences in the field of permissioned distributed ledgers, seeking for the definition of open and well-known operational mechanisms to validate participant nodes, support the automation of the lifecycles of the ledger and individual nodes, publish and execute operations regarding the recorded transactions

through Smart Contracts, improve security of distributed ledgers during both their design and operation and establish trusted links among different distributed ledgers using these mechanisms.

ETSI ISG PDL has been active since 2019 and has produced the following completed deliverables Group Report (GR) and Group Specifications (GS) to date ETSI TR 104 173 [i.22], ETSI TS 104 172 [i.23], ETSI GR PDL 001 [i.35], ETSI GR PDL 002 [i.36], ETSI GR PDL 003 [i.37], ETSI GR PDL 004 [i.38], ETSI GS PDL 005 [i.39], ETSI GR PDL 006 [i.40], ETSI GR PDL 008 [i.41], ETSI GR PDL 009 [i.42], ETSI GR PDL 010 [i.43], ETSI GS PDL 011 [i.44], ETSI GS PDL 012 [i.45], ETSI GS PDL 013 [i.46], ETSI GR PDL 014 [i.47], ETSI GS PDL 015 [i.48], ETSI GR PDL 017 [i.49], ETSI GR PDL 018 [i.50], ETSI GR PDL 019 [i.51], ETSI GR PDL 020 [i.52], ETSI GR PDL 021 [i.53], ETSI GS PDL 022 [i.54], ETSI GS PDL 023 [i.55], ETSI GS PDL 024 [i.56], ETSI GS PDL 025 [i.57], ETSI GS PDL 026 [i.58], ETSI GS PDL 027 [i.59], ETSI GS PDL 028 [i.60], ETSI GS PDL 029 [i.61], ETSI GS PDL 030 [i.62], ETSI GS PDL 031 [i.63], ETSI GS PDL 032 [i.64], ETSI GS PDL 033 [i.65]. Amongst the published documents, Smart Contracts were presented in ETSI GR PDL 004 [i.38], ETSI GS PDL 011 [i.44], ETSI GS PDL 033 [i.65], distributed ledgers and interoperability and all data issues in ETSI TR 104 173 [i.22], ETSI GR PDL 006 [i.40], ETSI GR PDL 009 [i.42], ETSI GR PDL 010 [i.43], ETSI GS PDL 012 [i.45], ETSI GS PDL 013 [i.46], ETSI GR PDL 018 [i.50]; trust, identity, and repudiation issues in ETSI GR PDL 014 [i.47], ETSI GR PDL 019 [i.51], ETSI GS PDL 023 [i.55], ETSI GS PDL 027 [i.59], ETSI GS PDL 030 [i.62], network issues in ETSI GR PDL 020 [i.52], ETSI GS PDL 022 [i.54] ETSI GS PDL 024 [i.56], ETSI GS PDL 025 [i.57], ETSI GS PDL 027 [i.59]; IoT, AI, and energy issues in ETSI GS PDL 028 [i.60], ETSI GS PDL 031 [i.63], ETSI GS PDL 032 [i.64]; reputation, settlement and Digital Autonomous Organizations in ETSI GS PDL 015 [i.48], ETSI GS PDL 026 [i.58], ETSI GS PDL 029 [i.61]. The guidelines for governance of Smart Contracts executed on a blockchain and distributed ledgers and in support for eIDAS2 [i.1] trust services were discussed in ETSI GR PDL 017 [i.49].

These publications provide a roadmap for how Smart Contracts **can** be used to automate and secure transactions, ensure compliance with European regulations and facilitate cross-border interoperability. The emphasis is on creating secure, scalable, and compliant Smart Contracts that **can** be used in a variety of industries, ranging from finance to healthcare, all within the highly controlled environments of permissioned ledgers.

As per ETSI ISG rules, ISG PDL **cannot** produce normative recommendations, only surveys, reference architectures, proof of concepts, and **can** suggests guidance. The heritage of the produced documents **will** convey into normative recommendations within the new ETSI TC DATA (e.g. ETSI TR 104 173 [i.22] and ETSI TS 104 172 [i.23]).

4.3.3.2 Terminology

Electronic Ledgers, distributed ledgers and Smart Contracts as defined in ISO 22739 [i.3].

4.3.3.3 Chain of Trust

ETSI ISG PDL (at the time of the publication of the present document) is agnostic with respect to the **Chain of Trust**. This **will** change in the future within the new ETSI TC DATA.

4.3.4 ITU-T X-Series Recommendations Study Group 17

4.3.4.1 Essential Overview

ITU-T X is a series of standards from the Standardization Sector the International Telecommunication Union (ITU-T), written by ITU-T Study Group 17. The description of the X series is: "Data networks, open system communications and security". The group produced a number of documents. In a nutshell:

- Recommendation ITU-T F.751.0 [i.29] Requirements for Distributed Ledger Systems.
- Recommendation ITU-T F.751.8 [i.30] Technical framework for distributed ledger technology (DLT) to cope with regulation.
- Recommendation ITU-T X.1401 [i.31] Security threats to distributed ledger technology.
- Recommendation ITU-T X.1402 [i.32] Security framework for distributed ledger technology.
- Recommendation ITU-T X.1403 [i.33] Security guidelines for using distributed ledger technology for decentralized identity management.

- Recommendation ITU-T X.1412 [i.34] Security requirements for smart contract management based on the distributed ledger technology.

4.3.4.2 Terminology

distributed ledgers as defined in Recommendation ITU-T F.751.0 [i.29], Smart Contracts as defined in Recommendation ITU-T X.1412 [i.34].

4.3.4.3 Chain of Trust

ITU-T X Study Group 17 is agnostic with respect to the **Chain of Trust**. However, Recommendation ITU-T X.1412 [i.34] contains some interesting intuitions on security requirements for Smart Contracts management based on the distributed ledger technology.

4.3.5 IEEE SA P2418

4.3.5.1 Essential Overview

IEEE Standards Association (IEEE-SA): the IEEE is working on developing blockchain and distributed ledger standards through the P2418 working group. They focus on areas such as digital asset management, blockchain for supply chains, and Smart Contracts. There are multiple standardized distributed ledger technologies, each with its specific features and applications. The choice of DLT depends on the use case, such as financial services, supply chain, IoT, or decentralized applications. These DLTs are often developed under open-source projects or standardized by international bodies like ISO and IEEE, ensuring that they adhere to global standards for security, privacy, and interoperability.

4.3.5.2 Terminology

None.

4.3.5.3 Chain of Trust

The IEEE SA P2418 working group did not publish any document.

4.4 Projects, Programs and Initiatives

4.4.1 Digital Europe Program

4.4.1.1 Essential Overview

The Digital Europe Program (DIGITAL) is an EU initiative designed to accelerate the integration of digital technologies into businesses, public administrations, and society. DIGITAL aims to enhance Europe's digital resilience by supporting projects in key areas like supercomputing, artificial intelligence, cybersecurity, and digital skills. This program is instrumental in reducing Europe's dependence on external digital solutions and strengthening the EU's digital infrastructure and capabilities.

DIGITAL supports industry, enterprises and fosters digital transformation across various sectors through initiatives. The program aligns with the EU's broader goals outlined in the 2030 Digital Compass and works in synergy with other EU funding mechanisms, including Horizon Europe and the Connecting Europe Facility, as part of the Multiannual Financial Framework 2021-2027.

The Digital Europe Program funds several projects focused on acceleration of eIDAS, EUDI Wallet and related trust services but also distributed ledgers, and Smart Contracts ISO 22739 [i.3] used for several use cases e.g.:

- Large Scale Pilots on EUDI Wallet
- Projects on the European Blockchain e.g.:

- EBSI VECTOR
- OnePass
- EBSI-NE
- TRACE4EU

- Projects for support of Standardization:
  - Blockstand
  - Seeblock

4.4.1.2 Terminology

Smart Contracts and distributed ledgers as defined in ISO 22739 [i.3].

4.4.1.3 Chain of Trust

Digital Europe Program, as per the publication date of the present document, is agnostic with respect to the **Chain of Trust**. This **can** change in the future.

4.4.2 EBSI

4.4.2.1 Essential Overview

The project, which was set up in 2018, aims to lay the foundation for future distributed ledger-based services within the EU and EFTA. The EBSI was transitioned into a new organizational entity for the operations of EBSI, named the European Digital Infrastructure Consortium (EDIC). The EBSI run by nodes operated by member states. Each country is expected to operate at least one node of EBSI at full scale. This approach aligns with the decentralized nature of blockchain technology and is suitable for multi-party cooperation. EBSI ensures a governmental trust anchor and so clear responsibility on the other hand this approach leads to the question on how such a network might be provided (QTSP for Electronic Ledger) or used (by EUDI Wallet Issuer or QTSP using DLT) by a certain provider. With the introduction of eIDAS2 and the concept of Qualified Electronic Ledgers, the EBSI could potentially not only evolve from an Electronic Ledger into a Qualified Electronic Ledger enhancing security and reliability of the network, but also providing legal certainty for use cases that build on the EDIC's Electronic Ledger.

EBSI contains a comprehensive technical framework on:

- Issuance, verification, revocation and presentations of verifiable credentials or attestations in terms of eIDAS
- Interoperability of wallets
- DID methods
- Timestamps
- API
- Governance for issuers and verifier (relying parties)

Currently there`s no possibility to implement and run Smart Contracts, as defined in ISO 22739 [i.3], on the EBSI infrastructure but this might change in future. The EBSI framework **can** automate processes like identity verification and product tracking, ensuring transparency and efficiency. For example, by using the Track and Trace API, it is possible to verify goods automatically at each stage, reducing manual checks and enhancing security across borders. The API might be extended to Smart Contracts in future. Recently (27 March 2025) it was announced that Smart Contracts, as defined in ISO 22739 [i.3], could be successfully deployed.

4.4.2.2 Terminology

Smart Contracts and distributed ledgers as defined in ISO 22739 [i.3].

4.4.2.3 Chain of Trust

EBSI, as per the publication date of the present document, is agnostic with respect to the **Chain of Trust**. This **can** change in the future.

4.4.3 EUDI Wallet

4.4.3.1 Essential Overview

The European Digital Identity Wallet (EUDI Wallet) is a key component of the eIDAS2 Regulation (EU) 2024/1183 [i.2].

The EUDI Wallet is designed as a secure and user-centric digital identity solution that allows citizens and residents of the European Union to authenticate their identity and access a wide range of online services, both public and private. The wallet **can** store and manage various forms of electronic attestations, including Person Identification Data (PID), Qualified Electronic Attestations of Attributes (QEAA), Electronic Attestations of Attributes (EAA) and Electronic Attestations of Attributes provided on behalf of the public sector bodies (EAA-Pub) like mobile Driving Licenses (mDLs).

The EUDI Wallet prioritizes privacy and security by design, ensuring that users have control over their personal data. It supports high levels of assurance for identity verification, which is critical for accessing services that require strong authentication. The wallet **can** be used across borders within the EU, fostering interoperability and ensuring that it functions seamlessly in different member states.

The Toolbox is a comprehensive set of technical specifications, standards, guidelines, and best practices developed to ensure the consistent implementation of the European Digital Identity Framework (eIDAS2) across the EU. The Toolbox serves as a reference for member states, helping them align their national digital identity systems with the European framework.

The infrastructure component of the eIDAS2 refers to the underlying technical and organizational structures that support the operation and use of the EUDI Wallet across the EU. This includes the roles of various stakeholders, the systems they operate, and the interfaces between these systems:

- EUDI Wallet Providers are entities, typically mandated by member states, responsible for providing and maintaining the EUDI Wallet solutions. They ensure that the wallets are compliant with the ARF's requirements and that they securely manage users' personal data and digital credentials.
- Person Identification Data (PID) Providers - trusted entities that verify the identity of users and issue PIDs to be stored in the EUDI Wallet. These providers play a critical role in ensuring that the identities within the wallet are accurate and trustworthy.
- Electronic Attestation of Attributes (QEAA, EAA-Pub, EAA) Providers - qualified and non-qualified Trust Service Providers (TSPs) that issue electronic attestations, such as diplomas or licenses, which **can** be stored in the EUDI Wallet. They ensure that the attributes linked to a user's identity are accurate and legally recognized.
- Relying Parties - the entities that request and rely on the identity and attribute data stored in the EUDI Wallet to provide services. They interact with the wallet through secure interfaces to verify users' identities and attributes.

The infrastructure also includes mechanisms for managing trust across the ecosystem, such as Trusted Lists and Certificate Authorities (CAs), which ensure that only authorized entities **can** issue and verify digital credentials.

Smart Contracts **can** play a potentially transformative role in the EUDIW under eIDAS2 by automating and enhancing the security, privacy, roles, and trustworthiness of digital transactions.

4.4.3.2 Terminology

Smart Contracts, SC Provider, SC Publisher.

4.4.3.3 Chain of Trust

EUDI Wallet, as per the publication date of the present document, is agnostic with respect to the **Chain of Trust**. This **can** change in the future.

4.5 Others

4.5.1 eIDAS Toolbox- Architecture and Reference Framework (ARF)

4.5.1.1 Essential Overview

Architecture and Reference Framework (ARF) for the European Digital Identity (EUDI) Wallet [i.17] is part of the European Union's initiative to create a standardized and secure digital identity system based on eIDAS2 regulation. The ARF is a draft prepared by the eIDAS Expert Group and provides the technical architecture, standards, and guidelines necessary for implementing the EUDI Wallet. It covers the roles and responsibilities of various stakeholders, including Wallet Providers, Person Identification Data (PID) Providers, and Qualified Electronic Attestation of Attributes (QEAA) Providers. The document also details the design principles, such as user-centricity, interoperability, privacy by design, and security by design, which are essential for the successful deployment of the EUDI Wallet.

4.5.1.2 Terminology

Smart Contracts, Electronic Ledger.

4.5.1.3 Chain of Trust

ARF is agnostic with respect of the **Chain of Trust**.

4.5.2 INATBA

4.5.2.1 Essential Overview

The International Association for Trusted Blockchain Applications (INATBA) offers public and private developers and users of DLT a global forum to interact with regulators and policymakers and bring blockchain technology to the next stage. INATBA facilitates positive change in the blockchain ecosystem. INATBA supports and promotes members to bridge public and private entities and promote global blockchain adoption across diverse fields such as law, finance and education.

4.5.2.2 Terminology

Smart Contracts and distributed ledgers as defined in ISO 22739 [i.3].

4.5.2.3 Chain of Trust

INATBA as per the publication date of the present document, is agnostic with respect to the **Chain of Trust**. This **can** change in the future.

4.5.3 ENISA: Digital Identity Standards

4.5.3.1 Essential Overview

ENISA is an agency of the European Union. The ENISA Digital Identity Standards [i.18] publications serve as a comprehensive analysis of the various standardization requirements that support cybersecurity policies, particularly in the realm of digital identity. The standards discussed encompass a broad spectrum, including policies, services, formats, protocols, and security requirements necessary for managing digital identities. These standards are essential in ensuring the security, reliability, and cross-border recognition of digital identities, which have become increasingly crucial due to the rise of digital services and electronic transactions, especially accelerated by the COVID-19 pandemic.

The documents outline the key areas covered by digital identity standards, which include identity management, trust services, authentication capabilities, and supporting services, and discuss the role of various standardization bodies, such as the European Telecommunications Standards Institute (ETSI), International Organization for Standardization (ISO), and national organizations like the National Institute of Standards and Technology (NIST) in developing these standards. Additionally, the documents highlight the evolution of digital identity standards from focusing on basic technical aspects like protocols and formats to addressing more complex issues such as cryptographic security, biometrics, and self-sovereign identities.

The analysis within the documents also delves into specific standards used in identity management, such as the ISO/IEC 24760-1 [i.70] series, which provides a framework for identity management, and ISO/IEC 29115 [i.71], which offers guidelines for entity authentication assurance. They also further examine the standards related to trust services, such as ETSI's standards for trust service providers, which are crucial for ensuring that digital transactions are secure and that digital identities **can** be trusted across different platforms and borders. The documents also provide with a set of recommendations aimed at European policymakers, standardization organizations, and cybersecurity agencies like ENISA, advocating for the continued development and adoption of robust digital identity standards to support the evolving landscape of digital transactions and cybersecurity needs.

Because of the intrinsic role of ENISA and the cruciality of having Smart Contracts secure, identity issues in Smart Contracts **will** be subject of study in the future.

4.5.3.2 Terminology

Smart Contracts, Electronic Ledger.

4.5.3.3 Chain of Trust

ENISA, as per the publication date of the present document, is agnostic with respect to the **Chain of Trust**. This **can** change in the future.

5 A Chain of Trust in support of Smart Contracts and Electronic Ledgers

5.1 Essential Overview

The present clause describes the processes involved in **building**, **deploying**, and **executing** a Smart Contract computer program on an Electronic Ledger. It formally identifies all the relevant **actors**, **artifacts**, **hardware**, **networks** and **tools**, emphasizing the critical points where governance, safety, security, and identity issues are required.

This is done by means of a novel and as yet unpublished **Chain of Trust**, considering all involved entities. The security of Smart Contracts **can** be significantly compromised by an incomplete validation chain, which exposes users to various risks, including fraud and attacks. Ideally, the **Chain of Trust** occurs at many abstraction levels:

- **SC Language entities.** Responsible to ensure that the design and the certification of a programming language used to encode the logic of a Smart Contract is not left to unknown not traceable communities.
- **SC Tools.** Responsible to ensure that the encoding and the certification of software tools like, e.g. a SC Compiler and a SC Virtual machine is not left to unknown not traceable communities.
- **SC Legal entities.** Responsible to ensure that the process of encoding and the certification of a Smart Contract **will** be clearly identified and traceable.
- **SC Published entities.** Responsible to ensure that the process of making available a Smart Contract on the market **will** be clearly identified and traceable.
- **Electronic Ledger.** Responsible to ensure that the process of running a Smart Contract on an Electronic Ledger **will** be clearly identified and traceable.
- **Underlying networks.** Responsible to ensure that the network infrastructure where distributed data structures, like Electronic Ledgers, **will** be clearly identified and traceable.

- **Hardware.** This point, although essential, is not treated in the present document.

One of the main findings from the analysis of the Data Act [i.1] and eIDAS2 [i.2] and its consequences to the standardization of Smart Contracts and Electronic Ledgers is that in order to satisfy the European rules for transparency and accountability, the actors of Electronic Ledgers and Smart Contracts **should** be identifiable according to Data Act [i.1] and eIDAS2 [i.2], respectively. More precisely, Smart Contracts **should** be strictly governed to give legal value, as per Smart Legal Contract definition in Clause 3.1. The same considerations for governance apply for Electronic Ledgers, that **should** be permissioned. This governance issue is independent for an Electronic Ledger to be **centralized**, **cloud-based**, or **distributed**, or any other of future technological implementation.

In parallel, eIDAS tools like Advanced Electronic Signatures (AdES) and Qualified Electronic Seals (QSeal) offer essential mechanisms for authenticating data and signing documents. AdES, which is uniquely linked to the signatory and created in a way that ensures their exclusive control, is fundamental in scenarios where Smart Contracts automate large-scale transactions. The use of AdES guarantees that each transaction is verifiably authentic and legally binding. These tools ensure traceability, authentication, and compliance with regulatory standards, providing a solid legal foundation for Smart Contracts in regulated environments.

A primary requirement for the use of Smart Contracts in the EU is to give assurance that in the event of a dispute that the parties to the Smart Contracts **can** be identified. The eIDAS2 framework is an existing framework that offers these capabilities and the role of eIDAS in Smart Contracts is described in ETSI TS 119 542 [i.16].

A suitable quality measure would be the adoption of Common Criteria [i.5], with a focus on Evaluation Assurance Levels (EAL) and Protection Profiles. These levels range from EAL1, which represents basic security, to EAL7, which provides the highest level of security, suitable for systems operating in high-risk environments. Protection Profiles specify security requirements for particular categories of products or systems, such as Smart Contracts managing sensitive transactions. For instance, a Smart Contract designed to handle financial transactions might be evaluated at EAL4, at least, ensuring a high level of security through methodical testing and vulnerability assessments. This would mitigate risks such as unauthorized access or data manipulation.

For the Chain of Trust, a proper validation, or at the very least, the identification of the tools used at each stage of the process, is essential. The toolchain identifies the following entities:

Software: Validating or at least identifying the authors, is essential to guarantee that an algorithm **can** be designed, coupled with some legal enforcements, translated into runnable code by a certified compiler, deployed on a Qualified Electronic Ledger, and executed on the top of a certified virtual machine, using certified inputs. This concretizes the concept, not standardized yet, of **Smart Legal Contract**.

Hardware: Validating or at least identifying the hardware (silicon) platforms involved is also crucial. However, deployment presents a more complex challenge, as validation or identification during the deployment phase often depends on the specific type of Electronic Ledger being used, and in some cases, it **can** be difficult or even impossible.

Networks: Validating or at least identifying the underlying network providers at each stage is essential and **should** be practically feasible.

A Smart Contract is a complex entity that has legal impact and which if compromised **will** seriously impact the relying parties. In recognizing this, the Smart Contract **can** be classified as requiring substantial or high-levels of assurance as defined in the Cyber Security Act [i.68], and this **should** be provided by conformance to an approved assurance scheme as defined by the Cyber Security Act, e.g. the EU Cybersecurity Certification Scheme on Common Criteria [i.69], managed by ENISA. Governance aspects of the overall security are given in ETSI TS 119 541 [i.12] that addresses the role of assurance schemes.

5.2 SC main entities

5.2.1 Essential Overview

Table 1 summarizes the **Chain of Trust**, in its first version **V1**, as a **numbered set of interactions** between entities, results produced, identification and assurance needs. Each rule, represented as a line in the Table, defines a precise interaction between two or more entities. The intuitive meaning of each column is:

- **Entity:** identifies each participating entity in the generation of a result which **may be** an object or a running Smart (Legal) Contracts on a (Qualified) Electronic Ledger.

- **Entities it interacts with:** identifies the entities with which the former entity interacts with or uses (in the case that the entity is an object, a program for instance) for producing the mentioned result.
- **Result produced:** identifies the result produced by the entities in the first and second column.
- **Identification needs:** requirements for identification of legal/natural persons responsible for a process and requirements for assuring the identity using electronic signatures/seals and/or identity authentication. This is addressed in ETSI TS 119 542 [i.16] which is expected to specify the requirements for identification of the mentioned entities and the requirements for the signatures on the Smart Contracts.
- **Assurance needs:** requirements for assuring the security and correct operation of a process. This is addressed in ETSI TS 119 541 [i.12] which is expected to specify the policies under which the required certification operations are carried out.

NOTE 1: Entities in the **Chain of Trust can** overlap each other.

NOTE 2: Rules in the **Chain of Trust may be** valid in any order.

NOTE 3: Rules in the **Chain of Trust should not** contradict each other over the time.

Table 1: The Chain of Trust V1

#	Entity	Entities it interacts with	Result produced	Identification needs	Assurance needs
SC Production					
1	SC Language Specification Team	SC Language Publisher	SC Language Specification	Signed by SC Language Publisher	• Correctness of syntax and semantics of SC Language Specification. • Respect of SC Language Specification Policy.
			SC Language Specification Policy	Signed by SC Language Publisher	
2	SC Compiler Team	SC Language Publisher SC Compiler Publisher	SC Compiler	Signed by SC Compiler Publisher	• Semantic preservation of the SC Compiler against SC Language Specification. • Respect of SC Compiler Development Policy.
			SC Compiler Policy	Signed by SC Compiler Publisher	
3	SC Virtual Machine Team	SC Language Publisher SC Virtual Machine Publisher	SC Virtual Machine	Signed by SC Virtual Machine Publisher	• Semantic preservation of the SC Virtual Machine against SC Language Specification. • Respect of SC Virtual Machine Development Policy.
			SC Virtual Machine Policy	Signed by SC Virtual Machine Publisher	
4	SC Developers Team SC Legal Team	SC Publisher	SC Package including SC Byte Code, SC Source Code, SC Legal Text, and SC Documentation	Signed by SC Publisher	• Assurance that SC Source Code, SC Byte Code, SC Legal Text, and SC Documentation meets the SC Development Policy. • Assurance that the SC Source Code, SC Byte Code, SC Legal Text, and the SC Documentation are identified by SC Publisher. • Assurance that the employed SC Compiler and SC Virtual Machine comes from a SC Compiler Publisher and SC Virtual Machine Publisher respecting the SC Compiler Policy and SC Virtual Machine Policy.
			SC Development Policy	Signed by SC Publisher	
SC Deployment					
5	SC Publisher	SC Provider	SC Package including SC Byte Code, SC Source Code, SC Legal Text, and SC Documentation	SC Provider and SC Publisher mutual identification	• Assurance that SC Package comes from a SC Publisher.
6	SC Provider	SC Deployer	Evidence of legal terms of SC Deployer	SC Provider and SC Deployer mutual identification	• Assurance of legal terms of SC Deployer.
7	SC Deployer	Electronic Ledger	Electronic Transaction in a Electronic Ledger containing the SC Package	SC Deployer identified by Electronic Ledger	• Assurance that SC Package comes from a SC Deployer.
SC Execution					
8	SC User	SC Provider	• Evidence of SC Legal Text from a SC Package. • Evidence of legal terms of SC Provider. • SC User inputs.	SC User and SC Provider mutual identification	• Agreement of legal terms of SC Provider. • Agreement of SC Legal Text.
9	SC Provider	Electronic Ledger	Electronic Transaction in a Electronic Ledger	SC Provider identified by Electronic Ledger	• Assurance of the truthfulness of inputs from SC User and inputs from SC Oracles and transactions for the Electronic Ledger

5.2.2 SC Language Specification

The semantics of programming languages, especially for domain specific languages for writing Smart Contracts, is fundamental to understand the execution in Electronic Ledger. The semantic rules of a programming language determine how its syntax is interpreted into actions to be performed. In the context of Smart Contracts, where transactions and contractual obligations are executed automatically, the clarity and precision of these semantics are indispensable. They **should** be unambiguous and comprehensive to prevent errors and security breaches. The use of formal methods to specify semantics, helps verify the correctness and security of the code.

5.2.3 SC Compiler

The design and implementation of a SC Compiler play a critical role for the design and execution of a Smart Contract which is executed on the top of one or many SC Virtual Machines relying on a centralized or distributed Electronic Ledgers: as an explanatory example, **different** SC Compilers compile the **same** SC Source Code into **different** SC Byte Codes that, in turn, **will** be all executed on a distributed ledger ISO 22739 [i.3] using **different** SC Virtual Machines.

Thus, a SC Compiler is responsible for translating a SC Source Code written using a **particular** version of a SC Language, into a SC Byte Code written on a **particular** version of a SC Byte Code Language that **can** run on **different** SC Virtual Machines, each of one capturing the semantic of a **different** SC Byte Code Language. This translation process is vital as it bridges the gap between human-readable code and machine-executable instructions.

The **compatibility** between languages definitions, compilers, byte codes, and virtual machines is thus capital to ensure a coherent behavior in a centralized or distributed setting.

The **absence** of European regulations **can** lead to **discrepancies** in how compilers interpret and translate code, potentially introducing bugs or vulnerabilities that are only evident once a SC Byte Code is deployed and executed on an Electronic Ledger, and as such, immutable. Without regulations and standardized specifications, SC Compiler developers might interpret the SC Language Specification and SC Language Specification Policy differently, leading to **non-compatible**, **semantically different** SC Byte Code and **inconsistent** Smart Contract behavior across platforms.

As an explanatory example, in case of Smart Contracts [i.3] executed on distributed ledgers as defined in ISO 22739 [i.3], a special kind of Electronic Ledger [i.1], the decentralized nature of the blockchain technology means that a Smart Contract [i.3] might be executed on many different nodes around the world, each potentially using slightly different compiler versions or settings. This decentralization exacerbates the risk of discrepancies and highlights the importance of establishing more uniform compiler standards.

It could be beneficial for the distributed ledgers community to consider frameworks that provide clearer guidelines and specifications for compiler development.

5.2.4 SC Virtual Machine

The design and implementation of SC Virtual Machines (VMs) are pivotal for the execution of Smart Contracts [i.3] across various blockchain platforms. These VMs translate the bytecode produced by compilers into executable actions within the blockchain's network.

As explanatory examples: Ethereum's Ethereum Virtual Machine (EVM) and the Solana's Sealevel operate under different principles and architectures, tailored to their specific blockchain ecosystems. For instance, EVM is designed for Ethereum's account-based model and handles transactions and contract states differently from Sealevel, which is designed to execute thousands of Smart Contracts as defined in ISO 22739 [i.3] in parallel, in a distributed ledger as defined in ISO 22739 [i.3], all optimized for Solana's unique consensus mechanism and high throughput capabilities.

5.2.5 Computer assisted software tools to assess correctness, safety, and security

In the development of Smart Contracts, ensuring the correctness, safety, and security of the software is paramount. To address these concerns, developers and researchers employ various computer-assisted software tools that aid in the formal verification and validation of SC Languages, SC Compilers, SC Virtual Machines, Electronic Ledgers and Smart Contracts.

As examples of the most applied Formal Verification Tools, the present document mentions:

1) **Rocq:** Rocq (formerly Coq) [i.24] is an interactive theorem prover designed to develop mathematical proofs and to write formally verified software. It is widely used in academia and industry to ensure the correctness of algorithms and to formally prove properties of programs. Rocq's ability to construct proofs makes it an invaluable tool for verifying the SC Languages used for Smart Contracts.

2) **Isabelle:** Isabelle [i.25] is another powerful theorem proving environment, which supports a variety of logical formalisms. It is used for writing and checking detailed proofs, and **can** also serve as a platform for developing robust, formally verified software. Isabelle's frameworks are particularly useful in verifying the correctness and security of Electronic Ledgers and Smart Contract code.

3) **Lean:** Lean [i.26] is a theorem prover and programming language designed for formalizing mathematical theorems and programming logically. It is used with distributed ledgers as defined in ISO 22739 [i.3] and particularly for the formal verification of Smart Contracts, ensuring that they execute as intended without unwanted side effects or vulnerabilities.

Application examples:

- **Smart Contract Verification:** Tools like Rocq and Isabelle have been used to develop formal models of blockchain environments and programming languages for Smart Contracts as defined in ISO 22739 [i.3], such as Solidity, executed on a distributed ledger as defined in ISO 22739 [i.3]. For example, a project might use Isabelle to formalize the semantics of Solidity and prove certain security properties, such as the absence of reentrancy vulnerabilities.

- **SC Compiler and SC Virtual Machine Verification:** The correctness of SC Compilers, which translate high-level SC Source Code into SC Byte Code, **can** be also verified using these tools. This is not new for usual programming languages. For instance, the CompCert [i.27] project uses the Rocq proof assistant to formally verify a compiler for the C programming language, ensuring that the compiler does not introduce any errors during the translation process. A similar approach **can** be adapted for SC Compilers and SC Virtual Machines.

Formal Tools like Rocq, Isabelle, and Lean **can formally check** that the SC Source Code and the SC Byte Code accurately reflects algorithmic logic semantic underneath the Smart Contract.

Implementation of Electronic Ledgers **can** be also formally checked.

By utilizing formal verification methods, it is possible to **ensure** that the algorithm does not contains bugs or logical errors that **could** lead to vulnerabilities. Automated tools **can** handle large volumes of contracts more efficiently than a manual process, making it scalable for applications that require numerous or frequently updated Smart Contracts.

Incorporating the Common Criteria (ISO/IEC 15408 [i.5]) in the use of these tools adds an additional layer of security assurance. The Common Criteria framework provides a structured process for evaluating the security and assurance of information technology products, which is directly applicable to Electronic Ledgers. By aligning the formal verification processes with Common Criteria standards, developers **can** certify the security and robustness of an Electronic Ledger and Smart Contracts running on the top of it, enhancing trust and compliance with international security standards. Recommendation ITU-T F.751.8 [i.30] advocates the use of formal methods to support the security of Smart Contracts running on DLT systems.

5.2.6 SC Legal Text, Certification of Smart Contract, Agreements

5.2.6.1 Essential Overview

Translating a certified SC Legal Text into a Smart Legal Contract is a detailed process. It ensures that the legal terms are precisely and securely translated into a SC Byte Code on a SC Virtual Machine using an Electronic Ledger. This is important to maintain the contract's integrity and enforceability.

A task force consisting of both Lawyers **and** Software Engineers works collaboratively to interpret the legal terms and requirements of a contract and then implement these into a Smart Legal Contract. Lawyers, represented in the present document as SC Legal Team, ensures that the legal nuances, represented using a Deontic Logic, are respected and fully represented, while software engineers, represented in the present document as SC Development Team, focus on encoding these terms into a SC Source Code, written in a SC Language, that is in turn compiled into a SC Package containing, among other files, the SC Byte Code that **will** be executed within one or many SC Virtual Machines on an Electronic Ledger.

Formal tools often have built-in libraries for reasoning with Deontic Logic: this would help SC Development Team and SC Legal Team to work together and converge to write a Smart Legal Contract that accurately reflects the stipulated legal terms and a formally proven executable code.

By utilizing formal verification methods, it is possible to ensure that the contract does not have bugs or logical errors that could lead to disputes or vulnerabilities. Reversing the process, i.e. translating SC Byte Code back into a SC Legal Text, is important for legal review, compliance checks, and in situations where parties need to understand the executed terms without reading the code.

This **can** be achieved by maintaining a comprehensive documentation and comments within the SC Source Code and the SC Package, that reflects the legal terms in a natural language. Observe that that in the **Chain of Trust**, the SC Package **should** be able to package **at least** SC Byte Code with SC Documentation, SC Source Code, and SC Legal Text.

5.2.6.2 SC Legal Text

The legal basis for a Smart Contract is defined using SC Legal Text. This **can** include:

a) Legal context in which the Smart Contract execution takes place such as European legislation, national legislation, or commercial agreements.

b) Provisions to meet the requirements for data protection of any personal data.

c) Requirements on SC Deployer Policy.

d) Requirements for SC Provider including:

   i) Use of SC Language tools including SC Compiler and SC Virtual Machine.

   ii) Use of Electronic Ledgers.

   iii) Verification of SC User identities.

e) License terms and conditions to be agreed by the SC User.

5.2.6.3 Certification of Smart Contract by SC Publisher

The elements of a Smart Contract and a Smart Legal Contract (SC Legal Text, SC Source Code, SC Byte Code, and other SC Documentation) **should** be certified by the SC Publisher which has overall responsibility for the Smart Contract.

The certification **should** be based on conformance to the SC Publisher's SC Development Policy. The certification **should** be provided by the SC Publisher which has overall responsibility for the Smart Contract.

5.2.6.4 Verification of legal agreement

a) Deployment of a Smart Contract

Before deploying a Smart Contract (a SC Byte Code), the SC Deployer **should** ensure that all the elements of the Smart Contract have been certified together by an identified SC Publisher.

In addition to making the SC Byte Code available on the Electronic Ledger, the SC Deployer **should** provide a successful validation report for SC Publisher signature against all the elements of the Smart Contract. Elements other than the SC Byte Code **can** be held outside the ledger but **should** include binding information (e.g. location reference and hash) alongside the validation report in the ledger. The SC Deployer **should** also record a confirmation that its SC Deployer Policy meets the requirements for deployment in the SC Legal Text.

b) Provision of a Smart Contract

Before executing a Smart Contract (a SC Byte Code) on the top of a SC Virtual Machine, the SC Provider should:

i) **Validate** the SC Publisher signature at least against the SC Byte Code and record the validation report in the Electronic Ledger.

ii) **Confirm** that SC Provider Policy, including use of an Electronic Ledger and SC Language tools, meets the requirements in the SC Legal Text and record this in the Electronic Ledger.

c) User license terms and conditions

d) Execution of a Smart Contract

Before executing a Smart Contract (a SC Byte Code) on the top of a SC Virtual Machine, the SC Provider **should** provide the SC User with a copy of the license:

i) The SC Provider **should** record in the Electronic Ledger information on the validation of the SC User identity along with a confirmation of the acceptance of the license terms and conditions which **should** be part of or bound to the SC Legal Text for the Smart Contract.

After executing a Smart Contract (a SC Byte Code), the SC Provider **should** provide a SC Execution Report.

5.3 Distributed ledger technology (DLT)

5.3.1 Essential Overview

Although Regulation (EU) 2023/2854 [i.1] and Regulation (EU) 2024/1183 [i.2] provide a normative framework for Smart Contracts and Electronic Ledgers, the present clause highlights the significant increase in the use of distributed ledgers as defined in ISO 22739 [i.3] over the past decade, operating on various distributed ledger technologies. As such, the present clause presents key information to outline the state of the art in distributed ledgers. The present clause has also basis in documents produced by ISO TC 307, and ETSI ISG PDL (at time of publication of the present document now part of ETSI TC DATA) and ITU-T. The aim is to understand the gap existing between Electronic Ledger and Smart Contracts, as defined by European regulations, and the existing distributed ledgers and Smart Contracts standard, as defined in Standard Organizations documents, and the *de facto* real solutions emerged and used by far.

The **Chain of Trust should** fill this gap.

5.3.2 Permissioned or permissionless

Permissioned distributed ledgers restrict network access to authorized participants only. In this model, each participant is explicitly allowed to join the network, typically by a network administrator or through a consensus of existing participants. Selected participants are allowed to validate and persist transactions. This setup is favoured by private organizations and consortiums where privacy, security, and control are priorities. Since participants are known and verified, it is easier to maintain confidentiality over transactions.

Permissionless distributed ledgers allow anyone to join and participate in the network without prior authorization. Every participant is allowed to validate and persist transactions. This type of ledger underpins cryptocurrencies like Bitcoin and Ethereum, supporting a **fully decentralized** environment.

5.3.3 Public or Private

Public distributed ledgers allow everybody to access all transactions and data so there is full transparency.

Private distributed ledgers allow to access only authorized users: similar conditions concerning execution of transactions **can** apply.

5.3.4 Data structures used to implement a distributed ledger

Electronic Ledgers, as defined in eIDAS2 regulation, **can** be implemented using either centralized or distributed technology, and as such a distributed ledger, as defined in ISO 22739 [i.3]. In both cases the used data structure is important to understand how the **Chain of Trust can** be applied.

The present clause recaps the state of the art of all data structures for distributed ledgers as described in ISO and ETSI and ITU-T documents. In a distributed ledger - **subset of** an Electronic Ledger - various data structures are used to ensure security, efficiency, and immutability. These data structures serve different purposes, such as storing transaction

records, maintaining integrity, and managing nodes and states. Below are some of the key data structures that **can** be used to implement distributed ledgers, also summarized in Table 2. For each data structure one list usage, structure and components, advantages, and a simple example of distributed ledger, commonly referred as blockchain.

The present clause is important in order to understand which data structure **can** be adapted or extended with lesser effort to the **Chain of Trust** without sacrificing backward compatibility with existing distributed ledgers and what it is described in Regulation (EU) 2024/1183 [i.2] and in its forthcoming Implementing Acts.

Each data structure plays a crucial role in the functioning, efficiency, and security of a distributed ledger:

1) Linked List:

   - **Usage:** a distributed ledger itself **can** be seen as a linked list where each block is linked to the previous one using cryptographic hashes. Each block contains a reference (hash) to the previous block, forming a chain.
   - **Advantages:** Simple structure, easy to traverse.
   - **Example:** Used in Bitcoin or Ethereum.

2) Merkle Tree (Hash Tree):

   - **Usage:** Merkle trees are used to efficiently and securely verify the integrity of large sets of data. A Merkle tree allows nodes to verify the consistency and validity of the transactions in a block without needing the entire data.
   - **Structure:** A binary tree where each leaf node is a hash of a data block, and non-leaf nodes are hashes of their child nodes.
   - **Advantages:** Efficient proof of data integrity, scalable, and reduces the amount of data stored by light clients (SPV nodes).
   - **Example:** Used in Bitcoin and Ethereum for efficient transaction verification.

3) DAG (Directed Acyclic Graph):

   - **Usage:** Some distributed ledger systems, like IOTA and Hedera Hashgraph, use DAG structures to manage transactions and consensus differently from traditional chains. Instead of linear blocks, transactions are stored in a graph where each transaction points to one or more previous transactions.
   - **Advantages:** Higher scalability, no need for mining, low latency.
   - **Example:** IOTA's Tangle, Hedera Hashgraph.

4) Patricia Trie (Radix Trie or Prefix Trie):

   - **Usage:** Patricia tries are used in Ethereum to efficiently store key-value pairs and ensure quick retrieval and verification of data. It is a form of a Merkle Trie that combines a tree and a Merkle Trie.
   - **Structure:** A compact and ordered data structure that stores a mapping from arbitrary-length binary strings to values.
   - **Advantages:** Space-efficient, allows for fast lookups, insertions, and deletions.
   - **Example:** Used in Ethereum for account storage and world state representation.

5) Heap:

   - **Usage:** Heaps are used to manage priority queues, especially for mining operations and transaction selection. For example, miners **may** use heaps to select transactions with the highest fees.
   - **Advantages:** Efficient handling of dynamic data, fast access to the highest-priority element.
   - **Example: May be** used in Bitcoin and Ethereum for transaction prioritization.

6) Bloom Filter:

- **Usage:** A probabilistic data structure used to test whether an element is part of a set or not. It is used in lightweight nodes (SPV nodes) to filter transactions and blocks relevant to them without having the full blockchain.
- **Advantages:** Space-efficient, fast, low false positives.
- **Example:** Bitcoin's SPV nodes use Bloom filters to query full nodes for relevant transactions.

7) Block Structure:

- **Usage:** Each block in a blockchain contains data like transactions, timestamps, the hash of the previous block, and a nonce.
- **Components:**
- **Header:** Contains metadata like the hash of the previous block, Merkle root, timestamp, and nonce.
- **Body:** Contains transaction details, including the sender, receiver, and amount.
- **Example:** Every blockchain uses this structure with some variations. For instance, Bitcoin has a simple structure, whereas Ethereum's blocks contain additional information for Smart Contracts and state transitions.

8) Account Trie:

- **Usage:** In Ethereum, each account is stored in a trie structure. The account trie maps the address to account details like nonce, balance, storage root, and code hash.
- **Advantages:** Efficient access and storage of account states, helps in keeping track of changes in accounts over time.
- **Example:** Used in Ethereum for improve efficiency.

9) Unspent Transaction Output (UTXO) Set:

- **Usage:** UTXO represents the set of unspent transaction outputs that are used to determine the available balance for a wallet.
- **Structure:** A database of all unspent outputs, where each output is indexed by its transaction ID and output index.
- **Advantages:** Enables stateless transactions, simplifies validation.
- **Example:** Used in Bitcoin, Litecoin, and other UTXO-based blockchains.

10) State Trie:

- **Usage:** The State Trie represents the global state of the distributed ledger, which includes all accounts and contracts in Ethereum. It is a critical part of Ethereum's world state.
- **Structure:** A Merkle Patricia Trie structure that stores the state of each account, including balances, nonces, and contract storage.
- **Advantages:** Enables efficient state verification and validation.
- **Example:** Core to Ethereum's execution model.

11) Transaction Pool:

- **Usage:** This is a temporary storage area for transactions that have been broadcast to the network but have not yet been included in a block. The pool is often managed as a priority queue.
- **Advantages:** Helps miners select transactions based on fees and ensures that pending transactions are accessible to the network.
- **Example:** Both Bitcoin and Ethereum use a transaction pool to store unconfirmed transactions.

12) Sparse Merkle Trie:

- **Usage:** Sparse Merkle Tries are used in systems where most entries are empty, such as in proof-of-stake systems for proof generation. These trees allow the blockchain to verify the existence or non-existence of data efficiently.
- **Advantages:** Compact, verifiable, ideal for systems with sparse data.
- **Example:** Used in various proof-of-stake protocols and newer blockchain projects.

Table 2: Summary of data structure management

Data Structure	Purpose	Examples
Linked List	Chain of blocks	Bitcoin
Merkle Tree	Efficient transaction verification	Bitcoin, Ethereum
DAG	Transaction verification without mining	IOTA, Hedera-Hashgraph
Patricia Trie	Efficient key-value pair storage	Ethereum
Heap	Transaction prioritization	Bitcoin (mining), Ethereum
Bloom Filter	Lightweight transaction queries	Bitcoin SPV Nodes
Block Structure	Block metadata and transactions	All blockchains
Account Tree	Storage of account details	Ethereum
UTXO Set	Unspent transaction outputs	Bitcoin, Litecoin
State Tree	Global state of the blockchain	Ethereum
Transaction Pool	Unconfirmed transaction storage	Bitcoin, Ethereum
Sparse Merkle Tree	Proof generation in sparse systems	Proof-of-stake protocols

5.3.5 On-chain and off-chain transaction data solutions

On-chain data refers to any information that is stored directly on a distributed ledger as defined in ISO 22739 [i.3]. This includes transaction records, Smart Contracts as defined in ISO 22739 [i.3], and any other data that needs to be immutable, transparent, and verifiable by all network participants. As an explanatory example, the Ethereum Virtual Machine stores all transactions, including the ones generated by the execution of a Smart Contract, on-chain. For example, a crowdfunding contract **can** record all contributions and funding thresholds directly on the Ethereum blockchain, ensuring transparency and immutability. Another example in Ethereum is the ERC-721 [i.72], dealing with Non-Fungible Tokens (NFTs): all information related to the ownership and transfer of an NFT is stored on-chain, ensuring the traceability and uniqueness of the token.

Off-chain data refers to any data that is stored outside of the distributed ledger as defined in ISO 22739 [i.3] but **can** interact with it when needed. This includes large files, databases, and other forms of data that do not need to be stored on-chain for every transaction. Some explanatory examples are listed below:

- **IPFS** is a decentralized storage protocol that allows large amounts of data to be stored off-chain while only a reference hash is stored on-chain. For example, in a digital content management system, multimedia files **can** be stored on IPFS, with the file hash preserved on the distributed ledger to verify integrity and origin.
- **Layer 2 Solution**, such as Lightning Network, is an off-chain scaling solution for the Layer 1 distributed ledger that allows fast and low-cost transactions. Transactions are recorded off-chain, with only the final balance reported on-chain.
- **Plasma** is a scaling solution that uses sidechains to process off-chain transactions, with the ability to anchor critical data on-chain. This reduces the load on the main distributed ledger while maintaining security and verification through the Ethereum MainNet.
- **Optimistic Rollups on Ethereum,** a scaling solution that allows Smart Contracts as defined in [i.3] to be executed off-chain with only the final results reported on-chain. This technique improves scalability and reduces costs while maintaining transaction integrity through fraud proofs.

5.4 Digital trust elements in Smart Contracts

5.4.1 Essential Overview

The aim of the present clause is to understand the gap existing between Electronic Ledgers and Smart Contracts, **as defined by** European regulations, and distributed ledgers and Smart Contracts, **as defined by** Standard Organization documents, and the *de facto* real solutions emerging and used by far.

The **Chain of Trust should** fill this gap.

5.4.2 Identification, authentication

Identity and Access Control:

- Every actor during a Smart Contract and Smart Legal Contracts execution is assigned a unique identity and corresponding access control rights. The governance is responsible for ensuring that all actors have appropriate and unique access rights.
- Access to Smart Contracts and Smart Legal Contracts is strictly controlled through mechanisms that enforce time-bound and role-based access, ensuring that only authorized parties **can** interact with the Smart Contract and Smart Legal Contracts at any given time.

Lifecycle Management:

- The lifecycle of a Smart Contract and Smart Legal Contracts includes proper planning, design, coding, deployment, and management. This includes defining the ownership and access control strategies during the planning phase to prevent future disputes.

Security and Privacy:

- Smart Contracts and Smart Legal Contracts ensure that identity information and access rights are securely managed. This includes using a trusted execution environment to prevent unauthorized access and ensures that only authenticated and authorized transactions occur within the Smart Contract and Smart Legal Contracts.
- Privacy concerns are addressed by implementing private chains or channels where necessary, allowing certain contractual details to remain confidential from other participants in the network.

Auditable Libraries and Verification:

- Developers are required to use auditable libraries for building Smart Contracts and Smart Legal Contracts. These libraries **should** be verifiable and approved by governance to ensure the integrity and security of the SC Source Code and SC Byte Code.

Enforceability:

- Smart Contracts and Smart Legal Contracts are designed to be self-executable upon the fulfilment of predefined conditions, and they **should** be enforceable across different jurisdictions. The governance **should** ensure that Smart Contracts and Smart Legal Contracts are aligned with the legal and regulatory frameworks of the participating entities.

5.4.3 Electronic signatures and seals

A digital signature as described in ETSI TR 119 001 [i.4] is a cryptographic transformation of a data unit that allows a recipient to prove the source and integrity of the data and to protect against forgery by the recipient. This involves appending data or transforming the original data in such a way that the origin of the data **can** be verified, ensuring its authenticity and integrity.

A digital signature is a mechanism, based on public key cryptography, which **can** be used to provide the legal equivalent of a handwritten signatures, commonly referred to in EU legislation as an electronic signature.

In the context of Smart Contracts, electronic signatures are crucial because they ensure that the actions and transactions recorded in the Smart Contract are authorized and verifiable by all parties involved. It protects the integrity of the

transaction and guarantees that the signatory **cannot** deny their involvement, thereby enabling trust and legal enforceability of the contract.

Under European legislation, electronic signatures, and the equivalent when applied by an organization (referred to as a legal person) called electronic seal, **can** come in several forms:

- Electronic Signature: An electronic signature is a data in electronic form that is attached to or logically associated with other electronic data and used by the signatory to sign. It is a broad term that encompasses various types of signatures used to confirm the authenticity of the signer and the integrity of the data. Under Regulation (EU) 2024/1183 [i.2] and Regulation (EU) No 910/2014 [i.6], it is a legal concept that ensures the authenticity and integrity of signed electronic documents.

- Advanced Electronic Signature: An advanced electronic signature is a specific type of electronic signature that meets certain requirements under Regulation (EU) 2024/1183 [i.2] and Regulation (EU) No 910/2014 [i.6]. It **should** be uniquely linked to the signatory, capable of identifying the signatory, created using electronic signature creation data that the signatory **can** use under their sole control, and linked to the data signed in such a way that any subsequent change in the data is detectable.

- Qualified Electronic Signature: A qualified electronic signature is an advanced electronic signature that is created using a qualified electronic signature creation device and is based on a qualified certificate for electronic signatures. This type of signature has the highest level of legal acceptance under EU law and is equivalent to a handwritten signature.

- Electronic Seal: An electronic seal is similar to an electronic signature but is used by a legal person (such as a company or organization) rather than a natural person. It serves as evidence that the electronic document or data has originated from a specific legal entity and ensures its authenticity and integrity.

- Advanced Electronic Seal: An advanced electronic seal is a type of electronic seal that, like an advanced electronic signature, meets certain criteria under Regulation (EU) 2024/1183 [i.2] and Regulation (EU) No 910/2014 [i.6]. It **should** be uniquely linked to the creator of the seal, capable of identifying the creator, created using electronic seal creation data that the creator **can** use under their sole control, and linked to the data to which it relates in such a manner that any subsequent change in the data is detectable.

- Qualified Electronic Seal: A qualified electronic seal is an advanced electronic seal that is created using a qualified electronic seal creation device and is based on a qualified certificate for electronic seals. Like the qualified electronic signature, it carries the highest level of legal recognition and provides a greater level of trust in the origin and integrity of the sealed document.

The key difference between an electronic signature and an electronic seal lies in their intended use and the type of entity applying them. An electronic signature is used by a natural person, acting under their control to perform a declaration of intent, often in the form of signing a contract or executing another legal act attributed solely to the individual. This natural person **may** act on their own behalf or on behalf of a legal person. When acting on behalf of a legal person, the electronic signature is applied based on a legal mandate or authorized representation. The electronic signature confirms both the identity of the natural person and their intent to bind themselves or the legal person they represent to a specific transaction or legal act.

An electronic seal, however, serves a different purpose. It is used primarily by a legal person to ensure the authenticity and integrity of documents. Unlike an electronic signature, it does not express intent but functions as a security measure to guarantee that the document's content has not been altered and originates from a verified legal person. While an electronic seal **cannot** directly replace an electronic signature, as it does not convey personal intent, it **can** fulfil the same business function in certain legal contexts. For example, after a contract has been signed, subsequent orders related to that contract **can** be automatically validated with an electronic seal, ensuring the document's origin and integrity without further action from a natural person. Electronic seals are especially important in trust services and are legally supported by the eIDAS regulation as a basis for their use.

In the context of Smart Contracts, an electronic signature is essential for confirming that the relevant documents and data entering the Smart Contracts, particularly those related to contract formation, obligations, or verification data, are validated by the natural persons who are parties to the agreement. In this way, the electronic signature serves as both a tool for identifying natural persons and for confirming the commitments they make within the Smart Contract.

On the other hand, an electronic seal **can** greatly support Smart Contracts by verifying the authenticity of the data input, particularly when acting as a source (or oracle). Moreover, if a Smart Contract generates data that is to be used outside of the ledger, the electronic seal **can** safeguard the authenticity, integrity, and origin of that data, ensuring it results from

the proper execution of the Smart Contract. This makes electronic seals a vital tool for maintaining trust and security in transactions involving Smart Contracts, especially for legal persons.

Below are the main methods and steps involved in generating digital signatures:

Digital signatures, which are a specific type of electronic signature that use cryptographic techniques for enhanced security, are typically generated using public key cryptography. Below are the main methods and steps involved in generating digital signatures:

1) Public Key Infrastructure (PKI): PKI is the most common and secure way of generating digital signatures. It involves the use of a cryptographic key pair, where a private key used to generate the digital signature (kept secret by the signer); and a public key used by recipients to verify the signature (shared with others).

2) Hardware Security Module (HSM): HSM is a physical device that securely stores private keys and performs cryptographic operations, including digital signature generation. The digital signature is returned from the HSM, which **can** be appended to the document. This method is common in high-security environments, such as banking, government, and large enterprises, where strict key management policies are required.

3) Smart Card or SIM card-Based Digital Signature: Smart Cards or SIM cards, which securely store cryptographic keys, **can** be used to generate digital signatures. The card performs the cryptographic operation to sign the hash of the document using the stored private key. Examples of using this method include systems like Mobile ID (e.g. in Estonia, Finland) or smart card-based authentication in organizations.

4) Digital Signature Software (e.g. AdobeSign®, DocuSign®): Digital signature software automates the process of key generation, signing, and verification. These platforms often integrate PKI under the hood, allowing users to sign documents digitally. The platform hashes the document and uses the user's private key to generate the digital signature.

5) Mobile Digital Signatures (mobile-ID): In some mobile digital signature schemes, the private keys are stored securely on a mobile device's SIM card or secure element, and signing happens via the mobile network. A user uses a mobile app that supports digital signatures (like mobile-ID). The app sends the digital signature, which **can** be verified by recipients using the public key.

Digital signatures provide strong security and integrity by using cryptographic algorithms, and the exact method for generating them **can** range from simple software-based solutions to high-security hardware-based systems. Depending on the use case (e.g. legal contracts, mobile signing, blockchain transactions), different approaches **can** be used, with PKI being the most widely used and secure. Whenever an entity in the **Chain of Trust** relies on the validity of a digital signature the successful validation of the signature **should** be recorded to avoid later claim against of the origin and integrity of the signed data.

5.4.4 Electronic identity

5.4.4.1 Essential overview

In the context of the eIDAS2 regulation [i.2], electronic identification is defined as the process of using person identification data in electronic form that uniquely represents either a natural person, a legal person, or a natural person representing a legal person. This process is crucial for authentication in online and offline services, ensuring that the identity of the individual or entity is accurately and securely confirmed during digital transactions.

The regulation lays out specific criteria and requirements for electronic identification schemes to be recognized and utilized across the European Union. This includes the issuance of electronic identification means (such as European Digital Identity Wallets), which contain the identification data necessary for authentication and are used to securely access services.

The regulation also emphasizes that electronic identification **should** meet certain assurance levels (low, substantial, or high) depending on the level of confidence required in the claimed identity, and it **should** be recognized and interoperable across different European member states.

Thus, in this context, electronic identity refers to a digitally represented identity that enables secure and trusted interactions across digital platforms, meeting specific legal and technical standards as outlined in the regulation. Whenever the identity of a SC User invoking a SC Contract is verified the successful validation of the identity **should** be recorded to avoid later claim against of the user invoking a Smart Contract.

5.4.4.2 Electronic identity in a mobile network

Mobile network operators also play a key role in providing secure identity services because they control SIM cards, which **can** store cryptographic keys and securely authenticate users. This concept is often referred to as mobile ID or Mobile Signature. A SC User **can** be identified when he/she is connected to the SC Provider using its mobile phone, and a particular mobile network. See also Clause 5.8.

Key Components of Electronic Identity in a mobile network:

1) SIM and eSIM card as a secure storage: SIM cards are tamper-resistant hardware used to store the user's private key securely. The private key is used to generate digital signatures or authenticate the user. Similarly, eSIM is a hardware module where the user's secret key **can** be programmed with software in the hardware module instead of plugging in a physical card. SIM cards and eSIM **can** perform cryptographic operations like generating digital signatures or encrypting data without exposing the private key.

2) Mobile device: the mobile device acts as the interface through which users authenticate or sign documents. It interacts with the SIM card or secure element for cryptographic operations. It also serves as a trusted device that **can** be used in multi-factor authentication systems (combining something the user "has", e.g. the phone or SIM, with something the user "knows", e.g. a PIN).

Benefits of mobile-based electronic identity are as follows:

1) Convenience: Users **can** authenticate or sign documents anywhere using their mobile phones without the need for additional hardware. No need for physical smart cards or separate hardware tokens.

2) Security: Strong two-factor authentication: combining "something you have" (the SIM card or phone) with "something you know" (a PIN or password). The private key is securely stored in the SIM card and never leaves it, reducing the risk of key compromise.

3) Widespread adoption: Mobile phones are ubiquitous, making it easy for users to adopt mobile ID services. Many mobile network operators are trusted entities with the infrastructure needed for secure identity management.

4) Legal validity: In many countries, digital signatures generated using mobile-ID systems are legally equivalent to handwritten signatures. Qualified Electronic Signatures (QES), which are generated using a secure device like a SIM card and a qualified certificate, have the highest level of legal recognition in regions like the EU under the eIDAS2 regulation. Currently the electronic identity scheme employed by mobile network operators in standards is still far away from complying with eIDAS2 and Data Act.

5.4.5 Distributed ledgers

Distributed ledgers **are a special kind of** Electronic Ledgers in presence of network facilities.

There are several Distributed Ledger Technologies (DLTs), **not necessarily aligned with** ISO 22739 [i.3] that provide frameworks and protocols for building decentralized systems, enabling secure and transparent transactions without relying on a central authority. DLTs offer different features, such as consensus mechanisms, and governance structures, but they generally conform to some level of global standards or industry best practices.

The **Chain of Trust should** be applied also on distributed ledgers.

Below are some of the most prominent examples of distributed ledger technologies at time of publication of the present document:

1) **Hyperledger Fabric™ (by Linux Foundation®):** Part of the Hyperledger project under the Linux Foundation, which is a collaborative effort to create open-source DLT frameworks for enterprise use cases. Consensus Mechanism: Pluggable consensus (supports various consensus algorithms, including Practical Byzantine Fault Tolerance and Raft).

   Key Features:

   - Permissioned Ledger: Designed for enterprise use, it operates on a permissioned network, meaning only authorized participants **can** join.

- Smart Contracts as defined in ISO 22739 [i.3]: Supports on-chain code, enabling automation of business logic.
- Privacy and Confidentiality: Offers private channels for confidential transactions between specific parties.
- Use Cases: Supply chain management, finance, healthcare, and government services.
- Standards Compliance: Follows industry best practices for data privacy, identity management, and cryptographic security. Some implementations also comply with regulatory standards like GDPR [i.7].

2) **Corda® (by R3):** developed by R3, a consortium of financial institutions, Corda is an open-source blockchain platform optimized for business and regulatory use cases. Consensus Mechanism: Corda does not use a traditional blockchain structure or consensus mechanism like Proof of Work. Instead, it uses a notary service that ensures transaction uniqueness and validation.

   Key Features:

   - Permissioned Network: Like Hyperledger Fabric, Corda is designed for permissioned networks with a strong focus on privacy and security.
   - Legal Contracts: Supports legal contracts that **can** be directly mapped into Smart Contracts as defined in ISO 22739 [i.3] and try to capture Smart Legal Contract definitions.
   - Interoperability: Focuses on interoperability between various systems and across regulatory frameworks.
   - Use Cases: Financial services (trade finance, payments, insurance), digital identity, and healthcare.
   - Standards Compliance: Corda is designed with compliance in mind, especially for industries like finance that require adherence to legal and regulatory standards (e.g. GDPR [i.7], ISO standards).

3) **Quorum® (by JPMorgan):** Standardization: A permissioned blockchain based on Ethereum, but with modifications for enterprise use. Initially developed by JPMorgan, it's now part of ConsenSys. Consensus Mechanism: Supports multiple consensus algorithms, including Raft and Istanbul Byzantine Fault Tolerance.

   Key Features:

   - Private Transactions: Quorum allows for private transactions and contracts, making it suitable for businesses that need to keep certain data confidential.
   - Performance: Enhanced transaction speed compared to the public Ethereum network.
   - Compatibility: Since it is Ethereum-based, Quorum **can** run Ethereum Smart Contracts as defined in ISO 22739 [i.3] and leverage existing Ethereum tools.
   - Use Cases: Banking, supply chain, insurance, and capital markets.
   - Standards Compliance: Quorum aligns with enterprise-grade security and privacy standards. It **can** be adapted to meet specific regulatory frameworks like Basel III for banking.

4) **Ethereum® (Public Network and Enterprise Ethereum)**: Ethereum is a well-known public blockchain network that follows decentralized standards but also has an enterprise-focused version known as Enterprise Ethereum under the Enterprise Ethereum Alliance. Consensus Mechanism: Ethereum has moved from Proof of Work (PoW) to Proof of Stake (PoS) with Ethereum 2.0.

   Key Features:

   - Smart Contracts, as defined in ISO 22739 [i.3]: Ethereum pioneered the concept of Smart Contracts as defined in ISO 22739 [i.3], enabling decentralized applications and Decentralized Finance (DeFi) projects.
   - Enterprise Ethereum: Provides privacy, permissioning, and scalability features needed for business use cases.
   - Use Cases: Public Ethereum is widely used for decentralized applications, NFTs, and DeFi, while Enterprise Ethereum is used in industries like supply chain, healthcare, and finance.

- Standards Compliance: The Enterprise Ethereum Alliance works on creating standards for enterprise use, ensuring compatibility with global industry and regulatory standards (such as ISO standards).

5) **Ripple (for XRP® Ledger):** Ripple provides a distributed ledger aimed at facilitating fast and cheap cross-border payments and settlements, particularly in the financial industry. Consensus Mechanism: Uses the Ripple Protocol Consensus Algorithm (RPCA), which is different from PoW or PoS. It focuses on agreement between trusted nodes (validators) for transaction validation.

   Key Features:

   - High Throughput: Ripple is designed for fast settlement of payments with low transaction fees.
   - Interledger Protocol: Allows for interoperability between different payment networks.
   - Use Cases: Cross-border payments, remittances, and currency exchange.
   - Standards Compliance: Ripple is focused on compliance with financial regulations like know-your-customer, anti-money-laundering, and ISO 20022 [i.73] (a multi part International Standard prepared by ISO Technical Committee TC68 Financial Services) messaging standards.

6) **IOTA®:** IOTA uses a Directed Acyclic Graph (DAG) structure called Tangle rather than a traditional blockchain. It's focused on IoT (Internet of Things) applications. Consensus Mechanism: There is no traditional consensus mechanism like PoW. Instead, each participant in the network confirms two previous transactions, making it a decentralized and scalable system.

   Key Features:

   - Zero-fee transactions: IOTA is designed to enable feeless microtransactions, ideal for IoT devices.
   - Scalability: The DAG structure allows for theoretically infinite scalability without traditional bottlenecks.
   - Use Cases: IoT, smart cities, machine-to-machine communication, supply chain management.
   - Standards Compliance: IOTA is working toward compliance with ISO 9001 [i.8] and ISO/IEC 27001 [i.9] standards for quality management and information security. It is also involved in the Industrial Internet Consortium (IIC) for standardizing IoT solutions.

7) **EOSIO®:** EOSIO is an open-source blockchain platform known for scalability and speed. It uses a Delegated Proof-of-Stake (DPoS) consensus mechanism. Consensus Mechanism: Delegated Proof of Stake (DPoS), where block producers are voted in by stakeholders.

   Key Features:

   - High Performance: EOSIO is designed for high throughput, supporting thousands of transactions per second. Governance: Built-in governance mechanisms allow for dispute resolution and upgrades.
   - Use Cases: Decentralized applications, enterprise solutions, social networks, and gaming.
   - Standards Compliance: EOSIO is designed for enterprise use and **can** be customized to meet various regulatory standards. It supports compliance with GDPR [i.7] and offers built-in mechanisms for on-chain governance.

8) **Stellar®:** Stellar is an open-source distributed ledger optimized for fast cross-border payments, similar to Ripple. Consensus Mechanism: Stellar Consensus Protocol (SCP), which relies on a quorum of trusted nodes for consensus rather than a traditional mining or staking process.

   Key Features:

   - Low Cost: Transactions on the Stellar network is low-cost and settle quickly.
   - Multi-Currency Transactions: Stellar supports multi-currency transactions and allows for the issuance of digital assets.
   - Use Cases: Cross-border payments, remittances, microfinance, and tokenization of assets.

- Standards Compliance: Stellar works to comply with global financial regulations like AML®, KYC®, and ISO 20022 [i.73], making it suitable for regulated financial institutions.

9) **EBSI:** See Clause 4.4.2.

5.5 Deployment and Execution of Smart Contracts and Smart Legal Contracts

5.5.1 Essential Overview

The present clause is about different kind of deployment and execution. Regulation (EU) 2023/2854 [i.1] and Regulation (EU) 2024/1183 [i.2] **are rather liberal** on those points.

- An Electronic Ledger "**can** be centralized or decentralized". This corresponds to give someone a "free hand" to different kind of deployment and execution environments.
- A Smart Contract is "a piece of code". This corresponds to give someone a "free hand" to map a Smart Contract into a SC Source Code or a SC Byte Code, or both, **with or without** SC Legal Text, **with or without** identification of publishers of SC Compiler or SC Virtual Machine, or any combination of the above components.
- Smart Legal Contract, as defined in the present document, **is undefined**. However, Regulation (EU) 2023/2854 [i.1] introduces the figure of "vendor of Smart Contracts" that trade Smart Contracts, and introduce a legal responsibility for the behavior of the contract he/she is trading for.

The **Chain of Trust should** fill this gap.

The present clause is kept voluntarily short because technical material **can** be retrieved almost everywhere on academia, web sites, encyclopedias, standardization organizations et al. involved in Computer Science and Data Science.

5.5.2 Centralized systems

Centralized data structure and centralized computing are the simplest way to store and execute. They represent the cornerstone of Computer Science and Data Science.

Centralized data structures and centralized computing are, by its nature, compatible with the **Chain of Trust**.

5.5.3 Decentralized systems

Decentralized data structure and decentralized computing raised in the '70 in opposition to pure centralized solutions: this non-constructive approach (all that is "not" centralized) make impossible to formally characterize with a single unambiguous definition.

Because of the too wide definition of decentralized data structure and decentralized computing, one does not have formal evidences that all decentralized data structure and decentralized computing are compatible with the **Chain of Trust**.

5.5.4 Distributed systems

Distributed data structures and distributed computing raised with the arrival of the network facilities (i.e. Internet) that allows system to communicate each other's. Control is not decentralized.

Distributed data structures and distributed computing **can** be compatible with the **Chain of Trust**.

5.5.5 Peer-to-peer systems

Peer-to-systems raised as an evolution of decentralized systems where data and control are completely distributed.

One does not **have evidences** that peer-to-peer data structures and peer-to-peer computing **can/cannot be** compatible with the **Chain of Trust**. This **can** change in the future.

5.5.6 Cloud systems

According to ISO/IEC 22123-2 [i.66], Cloud is a paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand.

Cloud data structures and cloud computing **can** be compatible with the **Chain of Trust**.

5.5.7 Fog systems

Fog is an improvement of Cloud.

Fog was standardized in IEEE 1934 [i.67]. Fog extends Cloud in order to cope with huge number of IoT devices and big data volumes for real-time low-latency applications.

Fog data structures and Fog computing **can** be compatible with the **Chain of Trust**.

5.6 Legal issues in Smart Legal Contracts

5.6.1 Essential Overview

The present clause is about the concept of Smart Legal Contract (a Smart Contract with legal relevance), in terms of evidence of the script/contract itself: it is relevant to bring the Smart Contract, considered as a simple code script with only technological relevance, into the legal context drawn by both EU Regulations [i.1] and [i.2]. When the computer code, therefore, also acquires legal relevance, it is necessary to validate it through the typical legal-tech tools, read SC Legal Text in the **Chain of Trust**. Legal systems agree to the, so called, **freedom of form** principle, namely, requirement that the agreement be made in a specific form in order for it to be valid between the parties. Therefore, smart legal contract **can** and **will** count as legal contracts.

The present clause contributes to fix some definitions and technical issues that are important to understand the European regulations, fit the future standards and the *de facto* standards all together.

The **Chain of Trust should** fill this gap.

5.6.2 Legal parties

Before thinking the logical flow and surely before the writing the code, the present document discusses legal issues related to the rendering of parties legal will and intensions. For a Smart Legal Contract this analysis is even more critical than a traditional paper or an electronic contract: in fact, Smart Contracts are mostly deployed in a public environment and theoretically usable by anyone: standards are needed to drive the coder, SC Development Team, and the lawyer, SC Legal Team, in order to map all the correct stakeholders.

5.6.3 Certified code translation and evidences

The present document discusses about logical/legal algorithmic faults detected by a **TechLawyer**, namely a Lawyer with Computer Science skills, able to work in Computer Forensics and able to render legal aspects into logical/diagram flows. The TechLawyer **should** be able to discern between computer code with no legal relevance and annotated computer code with legal relevance (i.e. a Smart Legal Contract). In a Smart Legal Contract, the legal contract, written in plain English and the contract execution written in computer code cohabitate in the same file stored in the Electronic Ledger. The **Chain of Trust can** be summarized as follows:

- "Plain English" Smart Contract: Smart Legal Contract is - also - a translation of a plain English contract. Standards are needed to grant that this operation is made reducing the risk of misinterpretation of parties' will.
- "Flow chart" Smart Contract Logic: while translating the parties' will, standards are needed to decant the plain English logic to a specific script/program.

- "Annotations and Code" Smart Contract: in order to grant the coherence and interpretation of the code, annotation ("comments") **can** be used directly inside the code. This approach, which needs standardization, is useful to grant interoperability and interpretation of the code itself, from a legal point of view.
- Evidence generation and long-term preservation: ledgers and (qualified) archiving are two useful tools to grant resiliency of evidences related to the Smart Legal Contract. They need to be used in this context to facilitate digital forensics to enforce Smart Legal Contracts, even in Courts.

5.7 Environmental and sustainability models of Smart Contracts

This topic, although essential, is not treated in the present document.

5.8 Underlying networks to support the deployment and execution of Smart Contracts

As cited from eIDAS2 [i.2]:

"(49) To ensure the proper functioning of European Digital Identity Wallets, European Digital Identity Wallet providers need effective interoperability and fair, reasonable and non-discriminatory conditions for the European Digital Identity Wallets to access specific hardware and software features of mobile devices. Those components could include, in particular, near field communication antennas and secure elements, including universal integrated circuit cards, embedded secure elements, microSD cards and Bluetooth Low Energy. Access to those components could be under the control of mobile network operators and equipment manufacturers. Therefore, where needed to provide the services of European Digital Identity Wallets, original equipment manufacturers of mobile devices or providers of electronic communication services should not refuse access to such components. In addition, the undertakings that are designated as gatekeepers for core platform services as listed by the Commission pursuant to Regulation (EU) 2022/1925 of the European Parliament and of the Council should remain subject to the specific provisions of that Regulation, building on Article 6(7) thereof".

Though Smart Contracts **can** be provided as an overlay service on top of a network infrastructure, the elements as well as the whole underlying networks **will** need to be considered when deploying the services. As the article (49) of eIDAS2 requires, EUDIW **should** be treated equally when accessing the underlying networks. Particularly, components on mobile devices (e.g. NFC, SIM card and eSIM) **should** fully support functioning EUDIW; in addition, for accessing the Smart Contracts over the mobile devices **should** be supported and operated by the mobile networks. In sum, both mobile device manufacturers, component vendors (e.g. card vendors) and network equipment vendors **should** fully support EUDIW and Smart Contract services.

The role of the underlying networks matters to the adoption of Smart Contracts.

On the one hand, some nationwide/worldwide network infrastructures directly decide the accessibility and coverage of the deployed dAPPs offering reachability to EUDIW. Without the underlying networks' participation, especially nationwide mobile network infrastructure, the service range **will** be quite limited.

On the other hand, underlying networks usually are usually built and operated by large operators (e.g. mobile network operators), thus a large number of subscribers are already gathered. Therefore, behind the underlying networks, the nature of the trusts from them plays a big role when offering dAPPs based on Electronic Ledger. As a result, underlying networks such as critical network infrastructures **should** stake their reputation to become a QTSP thus make the Smart Contracts highly trustworthy.

6 Synthetizing the Chain of Trust as a roadmap for ETSI TS 119 541 and ETSI TS 119 542

6.1 Essential Overview

The present clause synthetizes all the issues raised by the **Chain of Trust** presented in Clause 5. Ideally, it **passes the baton** to technical specifications ETSI TS 119 541 [i.12] and ETSI TS 119 542 [i.16] that **will** translate in formal requirements.

Some remarks are in order to understand the next two ETSI Technical Specifications [i.12] and [i.16]:

- They **should** specify whether there is the need for the mentioned specification to be certified or not, and in case yes, by whom and under which schema this certification **should** be carried out.
- They **should** specify whether there is the need for the mentioned SC Compiler and SC Virtual Machine to be certified or not, and in case yes, by whom and under which schemas these certifications **should** be carried out.
- They **should** specify the requirements for identification of the SC Compiler and the requirements for the seals on the SC Byte Code.
- They **should** specify the requirements for identification of the mentioned entities and the requirements for the signatures on the Smart Contract and of the Electronic Ledger.
- They **should** specify the requirements for identification of the Smart Contract caller and the requirements for this signed declaration.

The present clause **will** proceed by collecting potential issues worth of study by the following categories:

- Electronic identity issues.
- Cybersecurity issues.
- Privacy issues.
- Governance issues.
- Programming tools issues.
- Legal issues.
- Data sharing issues.
- Centralized and decentralized execution issues.
- Interoperability issues.
- Network issues.
- Open-source issues.

6.2 Electronic identity issues

Based on the evaluation of electronic identity issues, a family of electronic identity schemes **should** be selected as standardized schemes for Smart Contracts. In addition, for those that could not fulfil the EU Regulations, clear guidance **should** be suggested for electronic identity scheme migration (especially for legacy information and communication technology systems).

The **Chain of Trust** lies in a fundamental usage of electronic identity.

6.3 Cybersecurity issues

Trust service providers for Electronic Ledgers and Smart Contracts are required to meet the requirements of the NIS2 Directive [i.11]. Moreover, ETSI EN 319 401 [i.13] defines general policy and requirements for the security of trust service providers aimed at meeting the requirements of NIS2 [i.11].

At the time of writing of the present document, hackers have maliciously substituted some Smart Contracts code with another (refers as the "Bybit hack 2025"): it is difficult to fully understand what happened and all involved actors. The Bybit hack 2025 **would not be possible** using entities and interactions as in **Chain of Trust**.

6.4 Privacy issues

Privacy is an important factor to be taken into account for identification applied to Smart Contracts, in particular with regards to identification of the contracting parties. eIDAS signatures and eIDAS2 wallets support a number of features which support privacy.

eIDAS electronic signatures and seals allow for the use of pseudonyms when identifying a natural or legal person. This allows for the full identity of the person to be replaced with some other unique reference which does not directly identify the person. However, this still allows for a degree of traceability / linkability of a person's activity.

eIDAS2 identities support a number of features which assure privacy. If particular, through use of selective disclosure of attributes (see ETSI TR 119 476 [i.10]) it is possible using EU Regulation on Digital Identity Wallets to reveal only selected attribute of the person without revealing their full identity.

In considering the application of privacy measures, such as described above, the requirement that contracting parties **cannot** later deny in a court of law having agreed to the Smart Legal Contract based on Electronic Ledgers needs to be taken into account.

Further security **may be** afforded through security measure applied to the Electronic Ledger (e.g. use of secure records held off-chain referenced from the ledger) **may be** used to ensure the privacy of identities recorded in an Electronic Ledger.

Privacy issues are clearly described in the **Chain of Trust**.

6.5 Governance and Audit issues

Governance and audit issues are fundamental in the **Chain of Trust**.

Three areas of issues need to be taken into account in considering the governance of systems supporting Smart Contracts:

1) eIDAS2 [i.2] Requirements for Electronic Ledgers

i) Under definition for Electronic Ledgers as specified in eIDAS2 [i.2] Article 3 (53) the integrity and the accuracy of their chronological ordering of electronic data records which form the ledger needs to be ensured.

ii) Under eIDAS2 [i.2] Article 45i: Requirements for Qualified Electronic Ledgers they following specific requirements apply to Qualified Electronic Ledgers:

a) they are created and managed by one or more Qualified Trust Service Provider (QTSP) or providers;

b) they establish the origin of data records in the ledger;

c) they ensure the unique sequential chronological ordering of data records in the ledger;

d) they record data in such a way that any subsequent change to the data is immediately detectable, ensuring their integrity over time.

iii) Under eIDAS2 each QTSP is required to be supervised and audited under eIDAS [i.2] Article 20 and 21 and Article 24.2 including the requirements of NIS 2 [i.11].

2) **Requirements for eIDAS2 Electronic Ledgers involving Multiple QTSPs**

- Where more than one QTSP is involved in the creation and management of an Electronic Ledger the overall trust service, as provided by a community of QTSPs, needs to meet the requirements i) and ii) above in a common way. In addition, each QTSP needs to meet the requirement of iii) above.

3) **Requirements of Smart Contracts**

- The additional requirement of Smart Contracts, as specified in the definition given Data Act Article 2(39), in addition to use of an electric ledger, is "*the computer program used for the automated execution of an agreement or part thereof*".

- Firstly, the execution environment needs to be secure. If this is in a QTSP then this would be addressed by the general requirements of eIDAS2. Otherwise, similar NIS2 based controls **can** be used to ensure general security of the execution environment. If a cloud-based execution environment is used it might be sufficient to use a cloud environment certified under the EU Regulation on certification scheme. However, further analysis is required to ensure that any specific concerns for Smart Contracts are met the whichever approach is taken.

- Secondly, the "*computer program*" used needs to be considered trustworthy. This aspect needs specific consideration, because is **very generic**.

The main role of the governance regime is to assure the trustworthiness of Smart Contracts and the underlying system infrastructure.

Governance of an individual QTSP is provided through the eIDAS2 [i.2] supervision and audit regime.

Governance of a community of QTSPs providing an Electronic Ledger requires governance through a previsioning regime whereby not only the QTSPs are accepted under [i.2] supervision and audit regime, but also it is demonstrated that they apply a common Electronic Ledger policy for achieving the requirements of an eIDAS ledger in a collaborative manner. This permissioning regime requires a community governance permissioning system which issues its "trusted" information (e.g. trusted list) based on the results of an eIDAS audit including the audit against the requirements of the common Electronic Ledger policy.

Assurance that a computer program used for the automated execution of an agreement or part thereof needs its own governance regime. It **can** use eIDAS signing certificates but also the CA/Browser Baseline Requirements for the Issuance and Management of Publicly Trusted Code Signing Certificates **should** be taken into account. Additional requirements need to be placed on the origin of the computer program to ensure that the code is developed in a trustworthy manner and allows the parties agreeing to a contract to understand the basis of the agreement.

ISO, ETSI, CEN, and ITU-T X are quite active in governance issues concerning Smart Contracts, Electronic Ledgers, and distributed ledgers. Because of the rapid growth of use and development standards sometimes **overlap**, become **obsolete**, or have **conflicts**. At the time of publication of the present document, the text below reflects the **status of affairs** in governance and audit issues that are fundamental in the **Chain of Trust**.

ETSI TC ESI provides general security controls aimed at meeting the requirements of Regulation (EU) 2024/1183 [i.2] TSPs including the requirements of NIS 2 [i.11]:

- ETSI TS 119 541 [i.12] specifies the policy and security requirements for Smart Contracts using Electronic Ledgers as defined in eIDAS2 [i.2], and with other trustworthy tools, taking into account the framework of requirements identified in the present document.

- ETSI TS 119 542 [i.16] specifies the use of EU Regulation on Digital Identity Wallets, and advanced or Qualified Electronic Signatures and Seals conforming to the requirements of eIDAS2 [i.2]. The Advanced or Qualified Electronic Signatures and Seals in the present document are implemented using digital signatures.

- An audit of an individual QTSP that meets the specific requirements for Smart Contracts using Electronic Ledgers **can** be based on trust service policy and security requirements in line with the general audit and cyber security framework for trust services presented in ETSI EN 319 401 [i.13] and ETSI EN 319 403-1 [i.14].

ETSI GR PDL 017 [i.49] describes the features of a distributed ledger to be applicable as a Qualified Electronic Ledger and in support for eIDAS2 [i.2] trust services: it analyses the properties that a PDL **can** have to be an enabler for eIDAS regulation for electronic identification, authentication and signatures, and also for using eIDAS2 [i.2] in other areas of the Digital Economy. ETSI ISG PDL, at the time of publication of the present document, is merged in ETSI TC DATA. The ETSI TS 104 172 [i.23] **will** distill, among others, formal recommendations from ETSI GR PDL 017 [i.49] respecting compatibility and avoiding overlapping with ETSI TS 119 541 [i.12], ETSI TS 119 542 [i.16].

CEN JTC 19, at the time of the publication of the present document, is working on a specification for policy and security requirements for trust service providers providing Electronic Ledger services, following ETSI EN 319 401 [i.13] respecting compatibility and avoiding overlapping with ETSI TS 119 541 [i.12] and ETSI TS 119 542 [i.16].

ISO provides principles on which a community governance regime **may be** based ISO/TS 23635 [i.15].

Recommendation ITU-T X.1403 [i.33] provides telecom-specific privacy and security considerations for using distributed ledgers data in identity management.

6.6 Programming tools issues

SC Language Specification Team, SC Compiler Team, SC Virtual Machine Team, SC Language Publisher, SC Compiler Publisher, SC Virtual Machine Publisher, **should** cooperate in the production of the SC Compiler and a SC Virtual Machine. SC Developer Team and SC Legal Team and SC Publisher **should** cooperate to write a Smart Legal Contract. The entity(ies) identified in the Smart Contract as either the entity originating the Smart Contract, or the entities that agree to be bound by the Smart Contract, **should** also sign it. The SC Byte Code, generated by the SC Compiler, **should** be sealed by the SC Language Publisher. In case that the caller is not one of the entities identified in the Smart Contract but another entity who accepts to be bound by its terms and conditions, there is the need of a signed declaration of acceptance of these terms and conditions of the mentioned Smart Contract. ETSI TS 119 542 [i.16] **should** specify the requirements for identification of the Smart Contract caller and the requirements for this signed declaration.

Formal Verification: The SC Language Publisher, SC Compiler Publisher, and SC Virtual Machine Publisher **may** (at the highest level of security) include formal verification tools to ensure that Smart Contracts are mathematically proven to be correct, secure, and free from vulnerabilities:

- **SC Compiler and SC Virtual Machine Consistency:** The Language Publisher, SC Compiler Publisher, and SC Virtual Machine Publisher **should** ensure that the SC Compiler translates code consistently and accurately across different environments, with no discrepancies in the generated SC Byte Code. They **should** ensure that the SC Virtual Machine execute SC Byte Code consistently and accurately, even across different environments, with no discrepancies.
- **Automated Testing:** The Language Publisher, SC Compiler Publisher, and SC Virtual Machine Publisher **should** support automated testing frameworks that **can** run unit tests, integration tests, and stress tests to validate the behavior of the Smart Contract.
- **Error Reporting:** The Language Publisher, SC Compiler Publisher, and SC Virtual Machine Publisher **should** provide detailed error reporting and debugging tools to identify and resolve issues during the development process.
- **Security Audits:** The Language Publisher, SC Compiler Publisher, and SC Virtual Machine Publisher **should** integrate security auditing tools that **can** analyse Smart Contracts for common vulnerabilities like reentrancy, overflow, and underflow.

6.7 (Smart) legal issues

- **Legal Compliance:** The SC Publisher **should** ensure that Smart Contracts comply with relevant legal frameworks and **can** be validated against legal standards.
- **Contract-to-Code Translation:** The SC Publisher **should** provide mechanisms to accurately translate Legal Contracts into executable Smart Legal Contracts, ensuring that all legal terms are faithfully represented in the SC Byte Code.
- **Audit:** The SC Publisher **should** maintain an immutable audit that documents every change made to the Smart Contract, ensuring transparency and traceability.
- **Reverse Engineering:** The SC Publisher **should** allow for the extraction of legal documents from Smart Contracts to ensure they **can** be reviewed and understood in legal contexts.
- **Dispute Resolution Integration:** The SC Publisher **should** include tools for integrating dispute resolution mechanisms within Smart Contracts to handle legal disputes automatically or semi-automatically.

6.8 Data sharing issues

- **Data Privacy:** The (Qualified) Electronic Ledger **should** ensure that all shared data is encrypted and access-controlled to protect sensitive information from unauthorized access.
- **Data Integrity:** The (Qualified) Electronic Ledger **should** implement mechanisms to verify that data has not been tampered with during transmission or storage.

- **Interoperability:** The (Qualified) Electronic Ledger **should** support standard data formats and protocols to enable seamless sharing of data across different systems and platforms.
- **Scalability:** The (Qualified) Electronic Ledger **should** be able to handle large volumes of data efficiently without compromising performance.
- **Compliance:** The (Qualified) Electronic Ledger **should** ensure that data sharing practices comply with relevant regulations, such as GDPR [i.7], to protect user privacy and rights.

6.9 Decentralized execution issues

- **Performance:** The SC Publisher and the (Qualified) Electronic Ledger **should** execute efficiently, with minimal latency and resource consumption to ensure smooth operation across the network.
- **Reliability:** The SC Publisher and the (Qualified) Electronic Ledger **should** ensure that Smart Contracts execute reliably under all conditions, including network congestion or high transaction volumes.
- **Scalability:** The SC Publisher and the (Qualified) Electronic Ledger **should** support scaling, allowing Smart Contracts to handle increased loads without degrading performance.
- **Fail-Safe Mechanisms:** The SC Publisher and the (Qualified) Electronic Ledger **should** include fail-safe mechanisms to gracefully handle execution failures, **should** ensure that contracts **can** recover or roll back in case of errors.
- **Auditability:** The SC Publisher and the (Qualified) Electronic Ledger **should** provide tools to audit the execution of Smart Contracts, **should** ensure that every action taken by the contract **can** be traced and verified.

6.10 Interoperability issues

- **Cross-Platform Compatibility:** The (Qualified) Electronic Ledger **should** ensure that Smart Contracts **can** interact with other blockchains or systems, using standardized protocols and interfaces.
- **Data Standardization:** The (Qualified) Electronic Ledger **should** use standardized data formats to ensure that information **can** be shared and understood across different platforms.
- **Protocol Support:** The (Qualified) Electronic Ledger **should** support multiple communication protocols to enable interoperability between various networks and external systems.
- **API Integration:** The (Qualified) Electronic Ledger **should** provide robust APIs that allow external systems to interact with Smart Contracts, facilitating integration with other services and platforms.
- **Security:** The (Qualified) Electronic Ledger **should** ensure that interoperability does not compromise the security of the Smart Contracts or the connected systems.

6.11 Networks issues

- **Pervasiveness:** The network **should** support the users to access to the Smart Contracts with high availability and ubiquity (e.g. across urban and rural areas, fixed or mobile coverage).
- **Reliability:** The network **should** support the users to access to the Smart Contracts with high service continuity (e.g. the reliable connectivity either wired or wireless).
- **Trustworthiness endorsement:** The networks **should** contribute to maintain the high trustworthiness of the provided Smart Contract.
- **Security:** The network **should** ensure security from attacks, including distributed denial of service, sybil, and other common network-based threats.
- **Decentralization:** The network **should** be sufficiently decentralized to prevent any single entity from gaining control over the system.

- **Scalability:** The network **should** support scalability to handle a growing number of nodes and transactions without performance degradation.
- **Redundancy:** The network **should** implement redundancy and fault-tolerant mechanisms to ensure network reliability even if some nodes fail.
- **Low Latency:** The network **should** offer low-latency communication to ensure timely execution of Smart Contracts and transactions.

6.12 Open-source vs. Closed-source issues

Open-source **may be** a model to assess code during the software construction and maintenance: in this model Governance is distributed with a (un)limited number of participants (for example: Linux kernel™, GNU C-compiler). Open-source is also used by Governments as an extra non legal service to official services. As an example, the Etalab initiative of the French government.

Closed-source model **may be** also a possible model to assess code, but it **should** be assessed *ex ante*, using possibly Governance(s) that fund the software construction and validation.

7 Conclusions

The **Chain of Trust V1**, at the time of the publication of the present document, represent a first attempt to list a sufficient set of interactions between entities, results produced, identification and assurance needs. A precise interaction between two or more entities is shown. The **Chain of Trust V1** is translated in formal requirements in ETSI TS 119 541 [i.12] and ETSI TS 119 542 [i.16].

Annex A:
An example of the Chain of Trust

A.1 Essential Overview

This annex provides an explanatory example of the processes involved in designing, assigning a legal value, deploying and executing a Smart Legal Contract in an Electronic Ledger.

The example is presented by means of four figures.

The particular case of a deployment and execution of a Smart Legal Contract on a distributed ledger as defined in ISO 22739 [i.3] solution is presented.

The figures identify all the relevant actors, artifacts, hardware, networks and tools, emphasizing the critical points where security and identity issues are paramount.

This description is described by means of the **Chain of Trust** introduced in Clause 5, considering all involved entities and their relations. The **Chain of Trust** occurs at many abstraction levels: in the particular case of a distributed environment, extra difficulties arise. The security of deploying and executing Smart Legal Contracts **can** be significantly compromised by an incomplete validation chain, which exposes users to various risks, including fraud and attacks.

Summarizing, the entities involved in the **Chain of Trust** in a distributed setting are defined in Clause 3.1 and described in Clause 5.

A.2 Figures as an example of the Chain of Trust

Figure A.1, Figure A.2, Figure A.3 and Figure A.4 present the "fine-grained" implementation of the **Chain of Trust** as suggested in Table 1, instantiated to distributed ledgers as defined in ISO 22739 [i.3]: entities, their relations participating in the production, deployment, and execution of Smart Legal Contracts and the design of the SC Languages are represented.

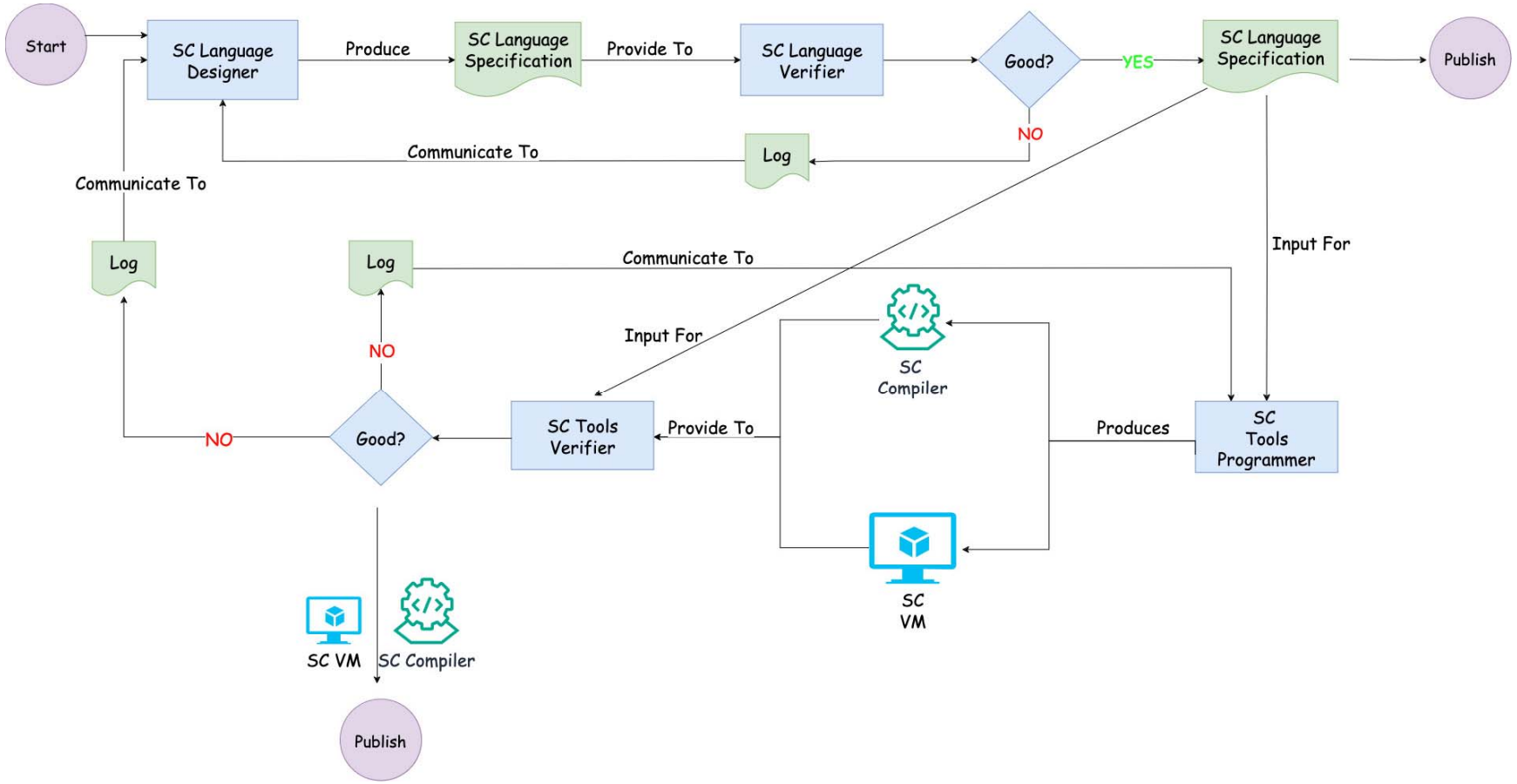

Figure A.1: Chain of Trust: SC Language design

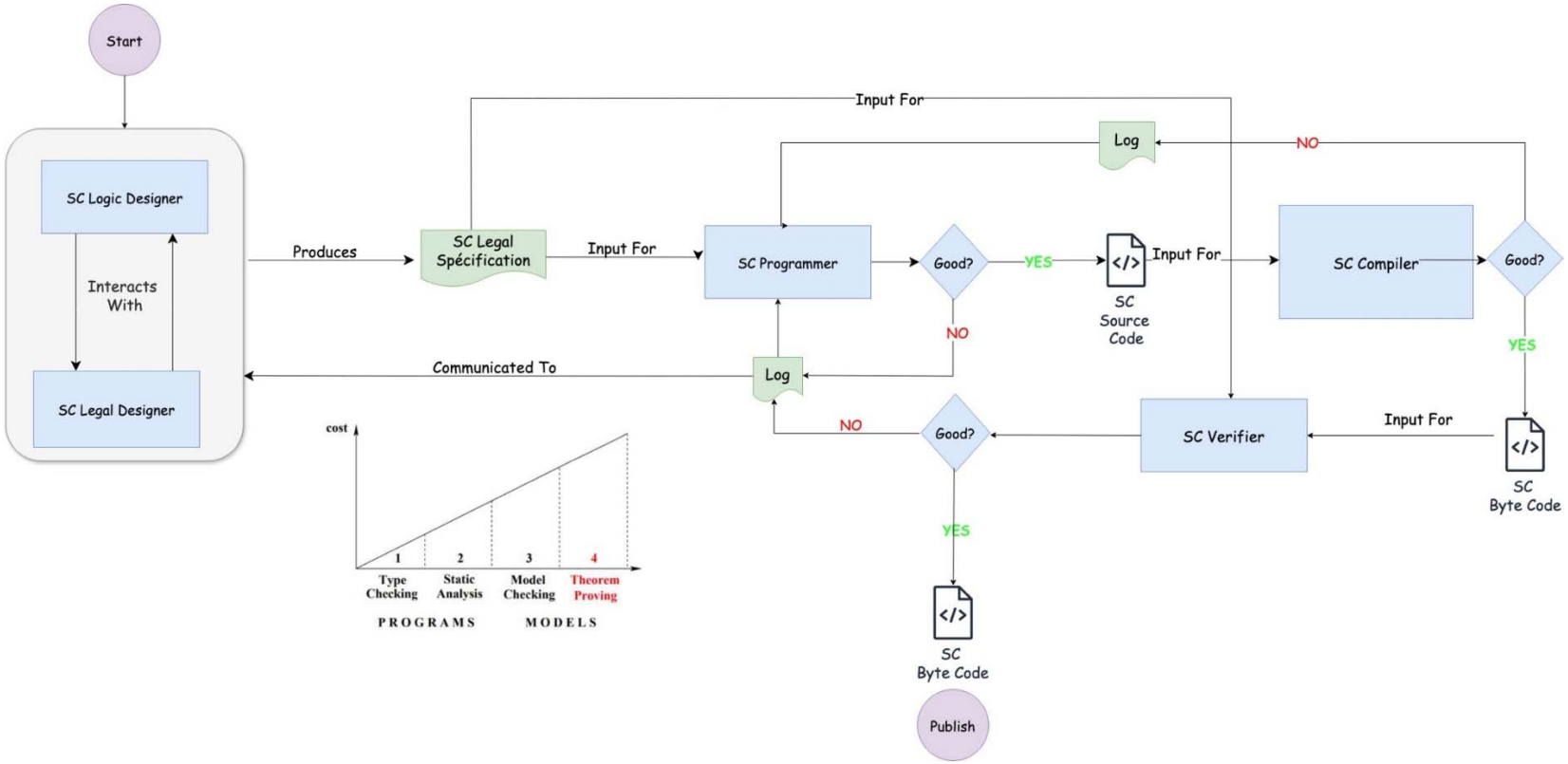

Figure A.2: Chain of Trust: Smart Legal Contract design

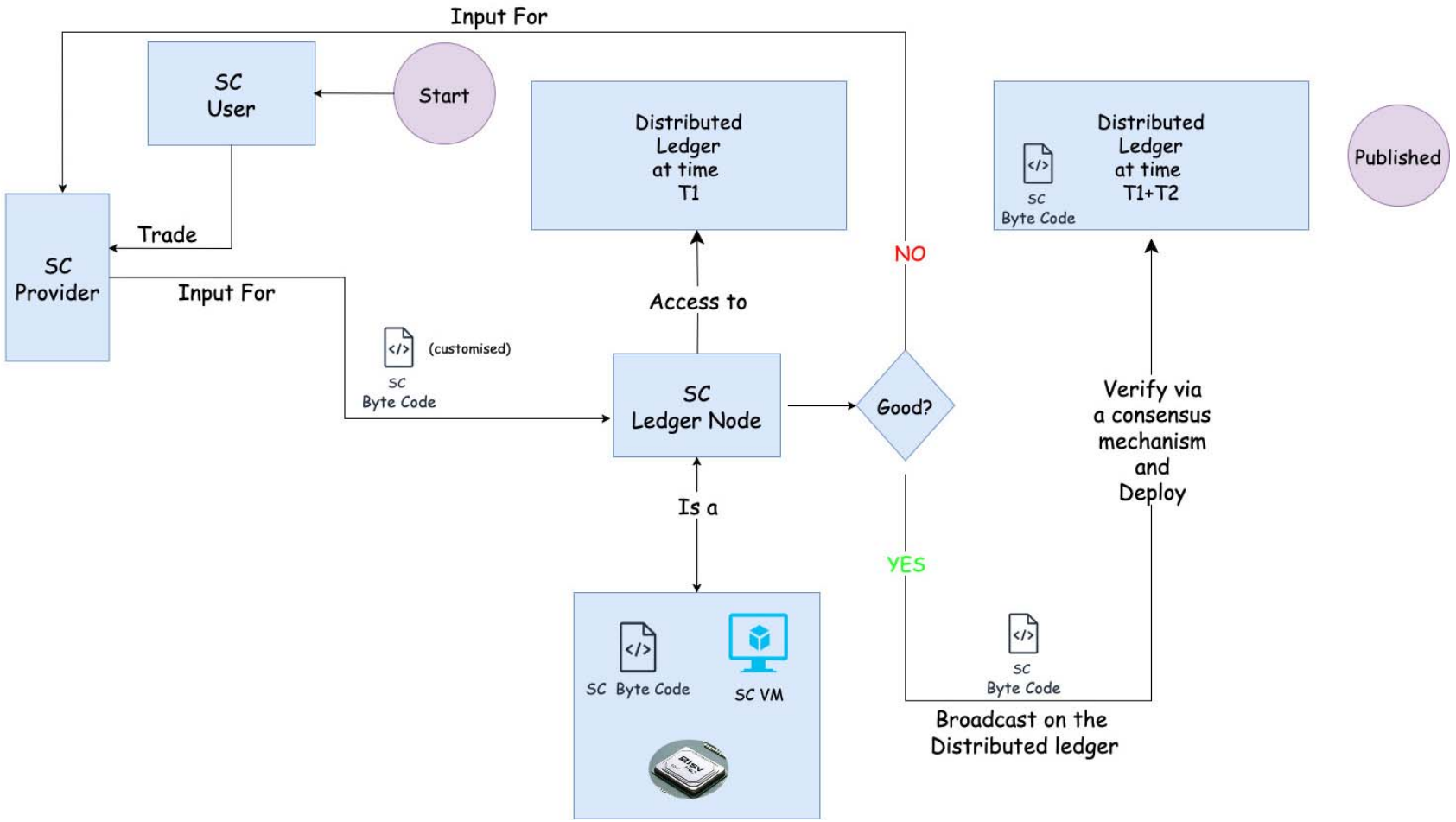

Figure A.3: Chain of Trust: Smart Contract deployment on a distributed ledger

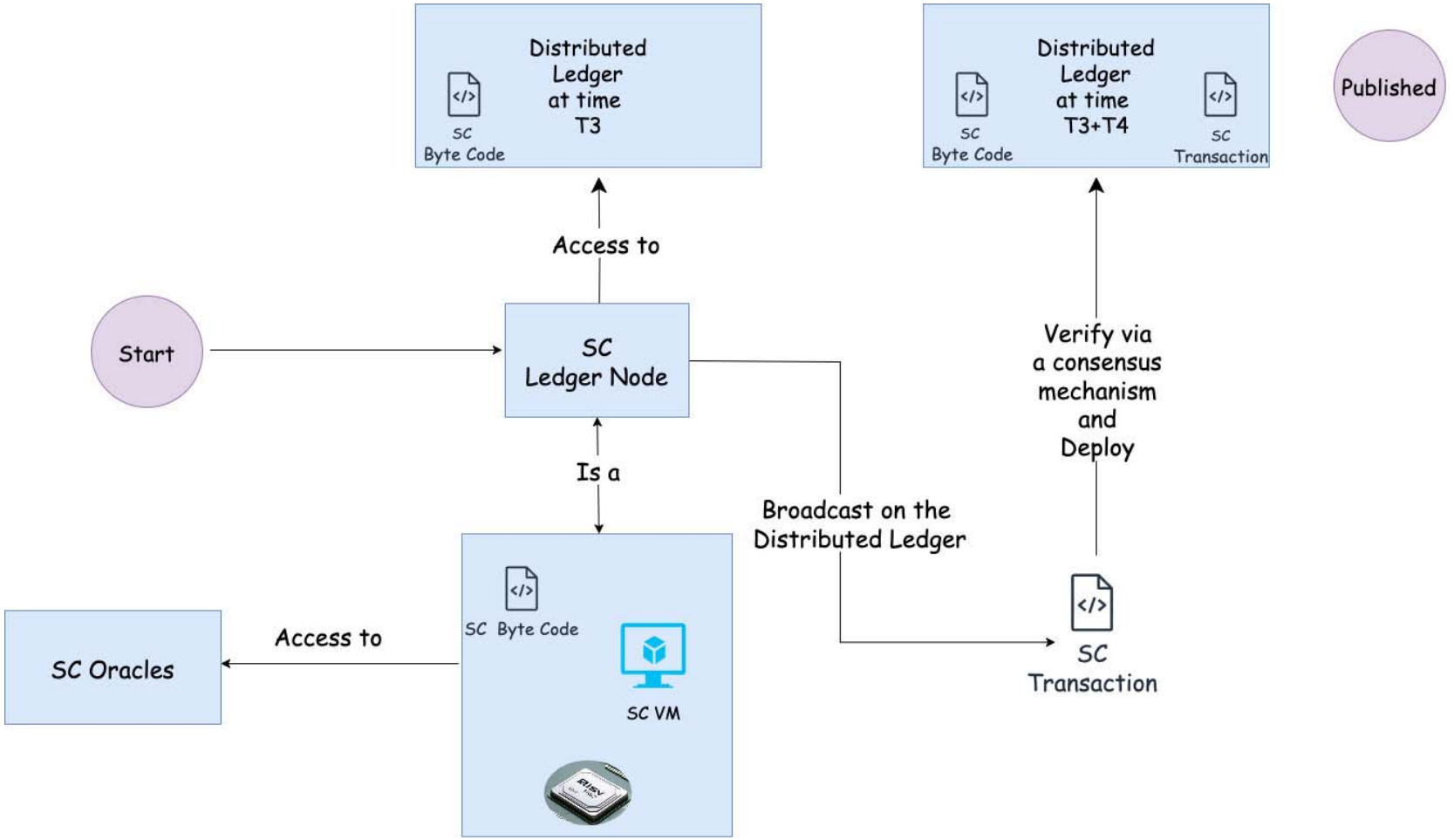

Figure A.4: Chain of Trust: Smart Contract execution on a distributed ledger

Annex B:
Chain of Trust: Architectural Elements (schematic)

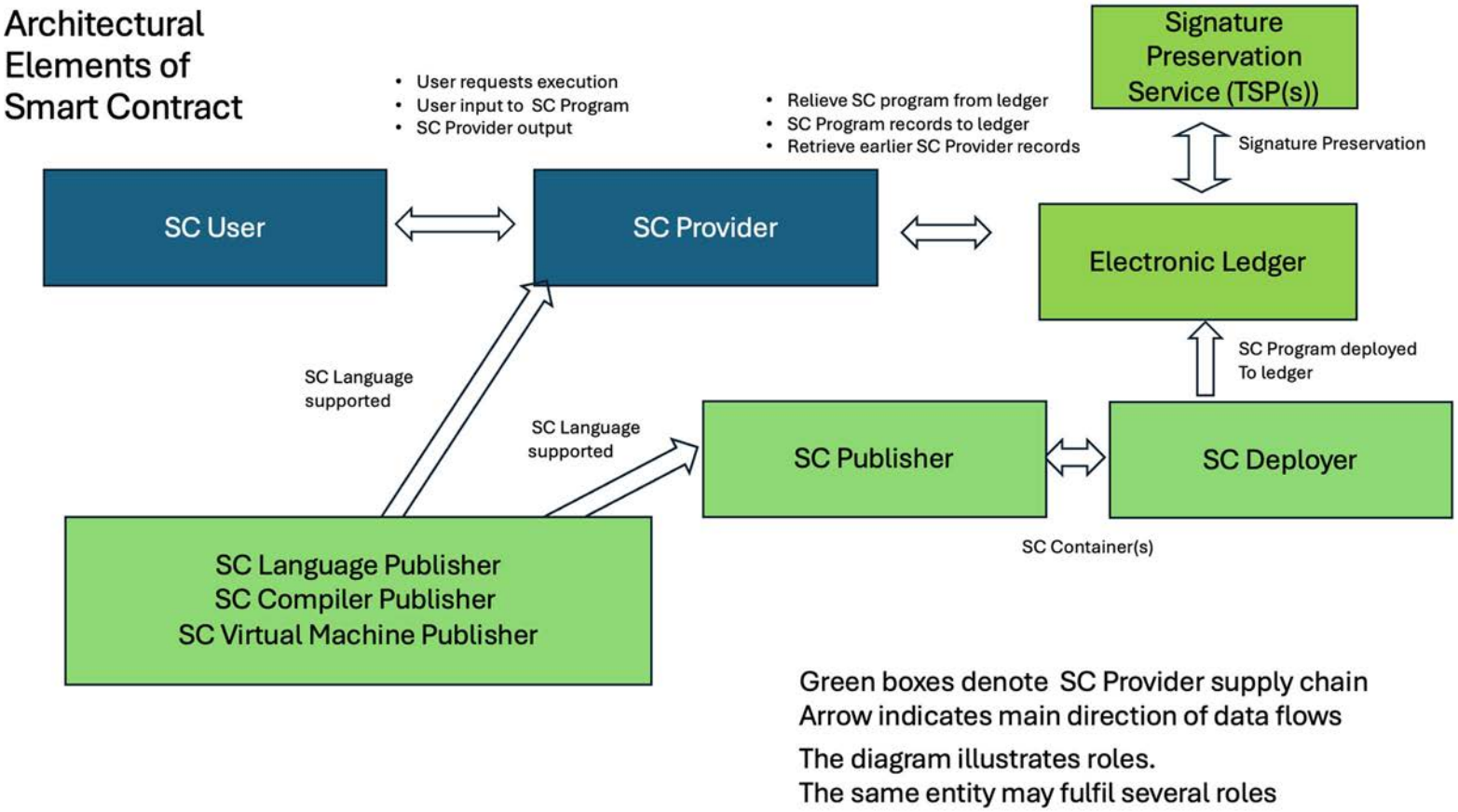

Figure B.1

Annex C:
Comparative overview of definitions

Legal definitions are technology-neutral and designed to support regulatory enforceability. ETSI TS 119 541 [i.12] and ETSI TS 119 542 [i.16] rely on the legal definitions to address legal compliance, and when it is the case, **can** reference ETSI or other standard definitions for implementation guidance.

Table C.1: Legal Definitions

Term	Source	Definition	Comment
Smart Contract	Regulation (EU) 2023/2854 [i.1], Article 2(39) (Data Act)	"A computer program used for the automated execution of an agreement or part thereof, using a sequence of electronic data records and ensuring their integrity and the accuracy of their chronological ordering."	Legal basis under the Data Act EU Law [i.1]. Smart Contract as per [i.1], are referred as SC Byte Code in the present document. **The definition of Smart Contract in [i.1] and in the present document is more general than the definition of smart contract in ISO 22739 [i.3].**
Electronic Ledger	Regulation (EU) 2024/1183 [i.2], Article 3(52) (eIDAS2)	"Electronic ledgers are a sequence of electronic data records which **should** ensure their integrity and the accuracy of their chronological ordering. Electronic ledgers **should** establish a chronological sequence of data records […] The process of creating and updating an electronic ledger depends on the type of ledger used, namely whether it is centralized or distributed. This Regulation **should** ensure technological neutrality, namely neither favoring, nor discriminating against, any technology used to implement the new trust service for electronic ledgers […]"	Legal basis under eIDAS2 EU Law [i.2]. **Because an Electronic Ledger can be centralized or distributed, the definition of Electronic Ledger in [i.2] and in the present document is more general than a distributed ledger in ISO 22739 [i.3].**

Table C.2: Technical Definitions

Term	Source	Definition	Comment
smart contract	ISO 22739[i.3]	Computer program stored in a distributed ledger technology (DLT) system wherein the outcome of any execution of the program is recorded on the distributed ledger	DLT-specific; **may** not align with legally neutral approach. Because of the specificity of the input of the computer program to be defined only with a DLT, the definition of the output of the computer program **can** be undefined in case of centralized Electronic Ledgers. **The definition of Smart Contract in [i.1] diverges with the definition of smart contract in ISO 22739 [i.3].**
distributed ledger	ISO 22739 [i.3]	Ledger that is shared across a set of distributed ledger technology (DLT) nodes and synchronized between the DLT nodes using a consensus mechanism	Contrasts with broader legal definition of "Electronic Ledger". **Because an Electronic Ledger can be centralized or distributed, the definition of Electronic Ledger in [i.2] is more general definition that a distributed ledger in ISO 22739 [i.3].**

Annex D:
Change history

Date	Version	Information about changes
February 2024	0.0.1a	Bootstrapping of the present document and few Editor annotations in RED (Inria)
March 2024	0.0.1b	Some Sections names proposals and more editor annotations in RED taken from the STF 655 contract (Inria)
23 April 2024	0.0.1c	Fix TR name according to the STF 655 contract. Discuss the first ToC V0 (Inria) and modify to ToC V1 (Inria, INFOCERT, UPC, Observatorium, Nokia)
30 April 2024	0.0.1d	Improve ToC according to the STF 655 contract, by Inria, INFOCERT and Nokia
30 April 2024	0.0.1e	Formatting (Inria)
7 Mai 2024	0.0.1f	Set up Clauses 1, 2, 3, References, Introduction. Simplifying and clustering ToC (Inria, Huawei, INFOCERT). Adding Editor annotations in RED
21 Mai 2024	0.0.1g	Refactoring of all Clauses keeping the contents (Inria, SSA, UPC, INFOCERT, Observatorium, Huawei). Adding Editor annotations in RED
28 Mai 2024	0.0.1h	Last review of all Clauses (SSA, Inria, InfoCert, CCC, Huawei). Adding Editor annotations in RED
11 Juin 2024	0.0.1i	Adding Editor annotations on Clauses 5 and 6 (SSA, UPC, Inria, Huawei) in RED
11 September 2024	0.0.1l	Including all Experts contributions, with a minimal formatting (Inria)
3 October 2024	0.0.1m	Expanding and including all Experts contributions, with formatting (Inria)
7 October 2024	0.0.2a	Clause 4 and clause 5 stabilized (Inria, UPC, INFOCERT)
17 October 2024	0.0.2b	Clause 5 moved to Clause 3 (Terms), including discussions on terms, rearranging Clause 5 (formerly 6), and Claude 6 (formerly 7), and inclusion all Huawei and Inria contributions (Inria)
17 October 2024	0.0.2c	Inclusions of all comments of the last meeting and few sanity checks (Inria)
17 October 2024	0.0.2c	Added bibliography and better Table 1 fitting Chain of Trust figures (Inria, SSA)
17 October 2024	0.0.2d	Drawing Chain of Trust figures, harmonizing Clause 4 (Inria)
22 October 2024	0.0.2d	Harmonizing Clause 5 and 6 (Inria)
30 October 2024	0.0.2d	Final pass (Inria)
20 November 2024	0.0.2e	NEW HANDY TABLE (See CR Meeting 19 November Inria)
3 December 2024	0.0.2f	Actual status of the Table 1 as per SSA/INFOCERT/Inria is Installed in Clause 5.1, Terms are installed in Clause 3.1, Clause 5.10 is deleted, and Figures are now in Appendix. Prose in Clause 5 is unstable
20 December 2024	0.0.3a	The Inria inspired and tuned by SSA and INFOCERT "Chain of Trust", agreed by ALL in the last two weekly meeting (3/12 and 10/12) is installed in Clause 5.2. A NEW Clause 3.1 (Terms) according to Table 5.2 is installed in RED. **The Chain of Trust and its Terminology will be synchronized in the TS x541 and TS x542**
7 January 2025	0.0.3b	Fixing Clause 3.1 (Terms) respecting UE terminology, and taking into account SSA and JTC19 comments (Inria)
14 January 2025	0.0.3c	Clause 4 (INFOCERT and Inria)
21 January 2025	0.0.3d	Merging and implementing dispositions (Inria)
23-25 January 2025	0.0.3e	Alignment with SSA and JTC19 on Terminology and on the "**Chain of Trust**" (Inria)
31 January 2025	0.0.4a	General improvements according to ETSI rules (Inria)
3 February 2025	0.0.5a	General last-minute improvements (Inria)
3 February 2025	0.0.6a	General last-minute improvements (ETSI)
April 2025	0.0.7a	Implementation of dispositions of comments for v0.0.6 producing a major new version (Inria)
April 2025	0.0.7b	Implementation of ETSI suggestions (ETSI)
May 2025	0.0.7c	Implementation of disposition of comments (Inria)
Juin 2025	0.0.8a	Various alignments with x541 and x542 and implementation of ETSI suggestions
July 2025	0.0.8b	Various alignments with x541 and x542 and implementation of C3L
July 2025	0.0.8c	Various alignments with x541 and x542 and implementation of C3L & UPC suggestions during the 10/07/25 meeting
September 2025	0.0.9a	Wrapping up and final tuning (Inria)
September 2025	0.0.9b	UPC last comment resolution (Inria)
September 2025	0.0.10a	Implementation of ETSI suggestions
September 2025	0.0.11a	Implementation of ETSI suggestions
September 2025	0.0.12a	Implementation of ETSI suggestions

History

Document history		
V1.1.1	October 2025	Publication