

ETSI GS PDL 033 V1.1.1 (2025-06)



Permissioned Distributed Ledger (PDL); Smart Contracts; System Architecture and Functional Specification

Document Preview

[ETSI GS PDL 033 V1.1.1 \(2025-06\)](https://standards.iteh.ai/catalog/standards/etsi/af037a39-be13-440b-ac4e-110aa6118a9a/etsi-gs-pdl-033-v1-1-1-2025-06)

<https://standards.iteh.ai/catalog/standards/etsi/af037a39-be13-440b-ac4e-110aa6118a9a/etsi-gs-pdl-033-v1-1-1-2025-06>

Disclaimer

The present document has been produced and approved by the Permissioned Distributed Ledger (PDL) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG. It does not necessarily represent the views of the entire ETSI membership.

Reference

DGS/PDL-0033_Smart_contract

Keywords

blockchain, PDL, policies, SLA, smart contract

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

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed, this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our [Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2025.
All rights reserved.

Contents

Intellectual Property Rights	11
Foreword.....	11
Modal verbs terminology.....	11
Executive summary	11
Introduction	12
1 Scope	13
2 References	13
2.1 Normative references	13
2.2 Informative references.....	22
3 Definition of terms, symbols and abbreviations.....	24
3.1 Terms.....	24
3.2 Symbols.....	25
3.3 Abbreviations	25
4 Introduction to Smart Contracts	26
4.1 Introduction	26
4.2 Object-Oriented Paradigm.....	27
4.2.1 Introduction to OOP in Smart Contracts.....	27
4.2.2 Key OOP Concepts in Smart Contracts	27
4.2.2.1 Encapsulation	27
4.2.2.2 State and Behaviour	27
4.2.2.2.1 State	27
4.2.2.2.2 Behaviour	28
4.2.2.3 Instantiation.....	29
4.2.2.4 Inheritance and Composition	29
4.2.2.5 Polymorphism	30
4.2.2.6 Visibility and Access Control	30
4.2.2.7 Events.....	30
4.2.3 Benefits of OOP in Smart Contracts	31
4.2.4 Considerations for OOP in Distributed Environments.....	32
4.2.4.1 Defining the major considerations	32
4.2.4.2 Gas costs	32
4.2.4.3 Public nature of blockchain data	33
4.2.4.4 Consensus mechanisms of the underlying distributed ledger.....	35
4.3 Properties of Smart Contracts.....	36
4.3.1 Immutability	36
4.3.1.1 Definition	36
4.3.1.2 Implications.....	37
4.3.1.3 Challenges	37
4.3.1.4 Solutions	37
4.3.2 Transparency.....	38
4.3.2.1 Definition	38
4.3.2.2 Key Aspects	39
4.3.2.3 Benefits	39
4.3.2.4 Challenges	39
4.3.2.5 Balancing Transparency and Privacy	40
4.3.2.6 Considerations for Implementation	40
4.3.3 Determinism	41
4.3.3.1 Definition	41
4.3.3.2 Key Aspects of Determinism	41
4.3.3.3 Importance of Determinism	41
4.3.3.4 Challenges associated with Determinism.....	42
4.3.3.5 Determinism Implementation Considerations.....	42
4.3.3.6 Balancing Determinism and Functionality.....	42

4.3.3.7	Requirements and Recommendations	42
4.3.4	Atomicity	43
4.3.4.1	Definition	43
4.3.4.2	Key Aspects of Atomicity	43
4.3.4.3	Importance of Atomicity	44
4.3.4.4	Atomicity Implementation Mechanisms	45
4.3.4.5	Challenges of Atomicity	45
4.3.4.6	Atomicity Best Practices	46
4.3.4.7	Atomicity in Multi-Contract Interactions	47
4.3.4.8	Requirements and Recommendations	47
4.3.5	Autonomy	48
4.3.5.1	Definition	48
4.3.5.2	Key Aspects of Autonomy	48
4.3.5.3	Importance of Autonomy	49
4.3.5.4	Autonomy Implementation Considerations	50
4.3.5.5	Challenges Associated with Autonomy	50
4.3.5.6	Balancing Autonomy and Control	51
4.3.5.7	Autonomy in Different Contexts	52
4.3.5.8	Best Practices for Implementing Autonomy in Smart Contracts	52
4.3.5.9	Requirements and Recommendations	53
4.3.6	Decentralization	54
4.3.6.1	Definition	54
4.3.6.2	Key Aspects of Decentralization	54
4.3.6.3	Importance of Decentralization	54
4.3.6.4	Decentralization Implementation Considerations	55
4.3.6.5	Challenges associated with Decentralization	55
4.3.6.6	Degrees of Decentralization	55
4.3.6.7	Decentralization in Different Contexts	55
4.3.6.8	Best Practices of Decentralization	56
4.3.6.9	Requirements and Recommendations	56
4.3.7	State Management	57
4.3.7.1	Definition	57
4.3.7.2	Key Aspects of State Management	57
4.3.7.3	Importance of State Management	57
4.3.7.4	State Management Implementation Considerations	58
4.3.7.5	Challenges associated with State Management	58
4.3.7.6	State Management Patterns	58
4.3.7.7	Advanced State Management Techniques	58
4.3.7.8	State Management Best Practices	59
4.3.7.9	Requirements and Recommendations	59
4.3.8	Interoperability	60
4.3.8.1	Definition	60
4.3.8.2	Key Aspects of Interoperability	61
4.3.8.3	Importance of Interoperability	61
4.3.8.4	Interoperability Implementation Mechanisms	62
4.3.8.5	Challenges associated with Interoperability	62
4.3.8.6	Interoperability Levels	62
4.3.8.7	Emerging Interoperability Solutions	62
4.3.8.8	Best Practices when Implementing Interoperability	63
4.3.8.9	Requirements and Recommendations	63
4.3.9	Threats and Security	64
4.3.9.1	Security Aspects of Smart Contracts	64
4.3.9.2	Key Aspects	65
4.3.9.3	Common Vulnerabilities and Attacks	66
4.3.9.3.1	Introduction	66
4.3.9.3.2	Internal Threats	67
4.3.9.3.3	Smart Contract Programming Errors	67
4.3.9.3.4	External Threats	68
4.3.9.4	Advanced Smart Contract Security	68
4.3.9.5	Security Best Practices and Culture in Smart Contracts	69
4.3.9.6	Tools and Techniques	70
4.3.9.7	Regulatory and Compliance Considerations	71

4.3.9.8	Emerging Security Challenges	71
4.3.9.9	Security by design	72
4.3.9.9.1	The importance of Security in the design phase of smart contracts	72
4.3.9.9.2	Access Control	72
4.3.9.9.3	Input Validation	72
4.3.9.9.4	Reentrancy Protection	72
4.3.9.9.5	Gas Limitations and Denial of Service	73
4.3.9.9.6	Upgradability and Modularity	73
4.3.9.9.7	Formal Verification	73
4.3.9.9.8	External Calls and Interactions	73
4.3.9.9.9	Error Handling	73
4.3.10	Reusability	73
4.3.10.1	Definition	73
4.3.10.2	Key Aspects	73
4.3.10.3	Importance	74
4.3.10.4	Implementation Strategies	74
4.3.10.5	Challenges	74
4.3.10.6	Best Practices	74
4.3.10.7	Examples of Reusable Components	75
4.3.10.8	Future Trends	75
4.3.10.9	Requirements and Recommendations	76
4.3.11	Composability and Contract Interactions	77
4.4	Storage	77
4.4.1	Introduction	77
4.4.2	Types of Storage	78
4.4.2.1	On-Chain Storage	78
4.4.2.2	Off-Chain Storage	78
4.4.2.3	Requirements and Recommendations	78
4.4.3	Storage Mechanisms	78
4.4.3.1	State Variables	78
4.4.3.2	Mappings	79
4.4.3.3	Arrays	79
4.4.3.4	Structs	79
4.4.3.5	Requirements and Recommendations	79
4.4.4	Storage Optimization Techniques	79
4.4.4.1	General discussion	79
4.4.4.2	Data Packing	80
4.4.4.3	Lazy Loading	80
4.4.4.4	Deletion and Cleanup	80
4.4.4.5	Requirements and Recommendations	80
4.4.5	Cost Considerations	80
4.4.5.1	Gas Costs	80
4.4.5.2	Storage Refunds	81
4.4.5.3	Requirements and Recommendations	81
4.4.6	Storage Data Security	81
4.4.6.1	Access Control	81
4.4.6.2	Data Integrity	82
4.4.6.3	Requirements and Recommendations	82
4.4.7	Advanced Storage Patterns	82
4.4.7.1	Architectural Patterns	82
4.4.7.2	Eternal Storage	82
4.4.7.3	Commit-Reveal Schemes	83
4.4.7.4	Merkle Trees	83
4.4.7.5	Requirements and Recommendations	83
4.4.8	Challenges and Considerations	83
4.4.8.1	Scalability	83
4.4.8.2	Privacy	85
4.4.8.3	Long-Term Storage	85
4.4.8.4	Requirements and recommendations	85
4.4.9	Future Trends	86
4.4.9.1	Decentralized Storage Solutions	86
4.4.9.2	Layer-2 Storage Solutions	86

4.4.10	Best Practices	86
4.4.10.1	General Discussion.....	86
4.4.10.2	Requirements and Recommendations	87
4.5	Modern Smart Contract Platforms and Languages.....	87
4.5.1	Introduction.....	87
4.5.2	Ethereum and Solidity	87
4.5.3	Polkadot and Ink	88
4.5.4	Cardano and Plutus	89
4.5.5	Algorand and TEAL/PyTeal	89
4.5.6	Cosmos and CosmWasm	90
4.5.7	Tezos and Michelson/LIGO.....	90
4.5.8	Emerging Trends	91
5	Smart Contracts - Lifecycle phases	92
5.1	Introduction	92
5.2	Planning Phase	92
5.2.1	Description and recent research	92
5.2.2	Defining the contract's purpose and requirements	92
5.2.3	Identifying Stakeholders and Their Interactions	92
5.2.4	Outlining the Contract's Logic and State Variables	93
5.2.5	Considering Security, Scalability, and Interoperability Needs	93
5.2.6	Evaluating Governance and Upgrade Models	93
5.3	Development and Testing Phase	93
5.3.1	Description and recent research	93
5.3.2	Writing the contract code in a suitable language	94
5.3.3	Implementing security best practices and optimizations.....	94
5.3.4	Conducting thorough testing.....	94
5.3.4.1	Introduction.....	94
5.3.4.2	Testing Strategies.....	94
5.3.4.3	Generalized Testing Targets.....	95
5.3.4.4	Testing Checklist.....	96
5.3.4.5	Offline Testing	96
5.3.4.6	Online Monitoring.....	97
5.3.4.7	Property-Based Testing Frameworks	97
5.3.4.8	Symbolic Execution Tools	98
5.3.4.9	SMT Solvers for Smart Contracts	99
5.3.5	Performing code reviews and audits	99
5.4	Deployment and Execution Phase	100
5.4.1	Discussion and recent research	100
5.4.2	Compiling the contract to bytecode	100
5.4.3	Selecting the appropriate network for deployment	100
5.4.4	Executing the deployment transaction	100
5.4.5	Verifying the deployed contract's bytecode.....	101
5.4.6	Monitoring the contract's execution and user interactions	101
5.5	Maintenance, Update and Upgrade Phases.....	101
5.5.1	Introduction.....	101
5.5.2	Update Situations.....	101
5.5.3	Strategies of Updating	102
5.5.4	Upgrading Through Versioning.....	102
5.5.5	Updating Steps.....	102
5.5.6	Checklist Before Redeployment	103
5.5.7	Securely Inactivating Old Contract.....	103
5.5.8	Governing the Upgrade of Smart Contracts.....	103
5.5.8.1	Discussion and recent research	103
5.5.8.2	Governance and upgrade models of Smart Contracts	104
5.5.8.2.1	Similarities and Differences Between Blockchain Platform Governance and Smart Contract Change Governance.....	104
5.5.8.2.2	Similarities.....	104
5.5.8.2.3	Differences	104
5.5.8.2.4	Recommendations for Effective Smart Contract Governance.....	105
5.5.8.2.5	Designing upgrade patterns	106

5.5.8.2.6	Establishing Processes for Proposing, Voting on, and Implementing Changes in Smart Contract Change Management	106
5.5.8.2.7	Balancing upgradability with security and immutability	107
5.5.8.3	Requirements and Recommendations	107
5.6	Retirement or Deprecation Phase	108
5.6.1	Discussion and recent research	108
5.6.2	Deciding when a contract should be retired	109
5.6.3	Implementing a graceful shutdown process	109
5.6.4	Ensuring users are notified and given time to extract assets or data	109
5.6.5	Potentially deploying a replacement contract	109
5.7	Requirements and Recommendations	110
6	Requirements for Designing a Smart Contract	111
6.1	Smart Contract Facets	111
6.1.1	Categories of Facets	111
6.1.2	Foundational Role	111
6.1.3	Functional Role	111
6.1.4	Governance Role	111
6.1.5	Interoperability Role	112
6.2	Actors	112
6.2.1	Distinct roles in a smart contract	112
6.2.2	Contract Developer	112
6.2.3	Contract Owner	113
6.2.4	Contract Users	113
6.2.5	Governance Body	113
6.2.6	Auditors	114
6.2.7	Oracles	114
6.3	Requirements During Design	114
6.3.1	Key considerations	114
6.3.2	Security	114
6.3.3	Scalability	115
6.3.4	Interoperability	115
6.3.5	Auditability	115
6.3.6	Privacy	115
6.3.7	Governance	116
6.3.8	Error Handling	116
6.4	Available technologies evaluation and selection	116
6.4.1	Introduction	116
6.4.2	Programming Languages	116
6.4.3	Development Frameworks	117
6.4.4	Security Analysis Tools	117
6.4.5	Oracle Services	117
6.4.6	Interoperability Protocols	117
6.4.7	Privacy-Enhancing Technologies	118
6.4.8	Scalability Solutions	118
6.5	Auditability considerations	118
6.5.1	Definition of Auditability	118
6.5.2	Code Transparency	119
6.5.3	Event Logging	119
6.5.4	Formal Verification	119
6.5.5	Automated Analysis Tools	119
6.5.6	Version Control and Change Management	120
6.5.7	External Audits	120
6.6	Designing and implementing Input and Output methods to Smart Contracts	120
6.6.1	Generalized Input/Output Requirements	120
6.6.1.1	Introduction	120
6.6.1.2	Data Integrity and Authenticity	121
6.6.1.3	Data Format and Validation	121
6.6.1.4	Error Handling	121
6.6.1.5	Rate Limiting	122
6.6.1.6	Randomness	122
6.6.1.7	Governance Inputs	122