OSNIAS CLEARING

https://www.osnias-clearing.com/osniasid/

contact@osnias-clearing.com

Osnias-ID

A Clearing ID Account as Inter-chain Identifier

Author	**Denis Bouzon**
Project	**Osnias Clearing**
ORCID	**0009-0007-8894-8902**
Website	**https://www.osnias-clearing.com/osniasid/**
Contact	**contact@osnias-clearing.com**
Release	**Release 3.0 · Working revision · 26 September 2026**
License	**@ Copyright Osnias-clearing** **Creative Commons Attribution 4.0 International (CC BY 4.0)**
ZENODO DOI	**https://zenodo.org/uploads/22978105**

Abstract

Osnias-ID (OID) is a project-specific internal identifier for clearing accounts within the Osnias Clearing network. Release 2.0 extends the identifier from 16 to 19 decimal digits and makes two clearing properties intrinsic to the identifier: the ISO 4217 numeric currency code and the clearing-cycle duration expressed in weeks. The canonical format is KK-NNN-MMM-WW-SSSSSSS-XX. This enables a clearing account to be structurally bound to both its monetary rail and its clearing-room cycle before admission to escrow, payment messaging or settlement workflows.

The complete 19-digit value fits within an unsigned 64-bit integer. Implementations may therefore store the canonical numeric OID in a uint64 / 8-byte machine word while presenting the same value to users in the hyphenated decimal form. Currency, cycle, checksum and registry status remain independent admission controls: a correct checksum alone never proves eligibility.

Osnias-ID is not a legal-entity identifier, bank account number, wallet address, regulatory authorization number, security or token. The authoritative legal/KYC mapping remains in the controlled registry layer.

Keywords

Osnias Clearing, Osnias-ID, Clearing Account Identifier, Multi-Chain Clearing, ISO 4217, Clearing Cycle, Escrow, Currency Binding, uint64, 8-byte Identifier, MOD-97 Checksum, Participant Registry, Wallet Identity, KYC, KYB, AML, ISO 20022, Institutional Blockchain

Table of content

Abstract ... 1
Keywords ... 1
1. Original Design Intent: Why Osnias-ID Is Needed ... 4
1.1 Chain-independent clearing reference ... 4
1.2 Compact messaging and deterministic routing ... 4
1.3 Prevent monetary and temporal routing collisions ... 4
2. Purpose and Scope ... 4
3. Status of the Specification ... 4
4. Identifier Structure — Release 2.0 ... 5
5. Currency and Clearing-Cycle Binding ... 5
5.1 Currency field MMM ... 5
5.2 Clearing-cycle field WW ... 5
6. Identity and Addressing Are Separate Layers ... 7
7. Registry Architecture ... 7
8. Access Control: Osnias-ID Required for Clearing Participants ... 7
9. One Legal Entity, Multiple Osnias-ID Clearing Accounts ... 7
10. Cross-Chain Use ... 8
11. External Identifier Interoperability ... 8
12. Relationship to ISO 20022 and ISO 4217 ... 8
13. Allocation Hierarchy ... 8
14. Machine Representation: 19 Decimal Digits in uint64 ... 9
15. API and Resolution Model ... 9
16. Normative Checksum Specification ... 10
16.1 Input ... 10
16.2 Calculation ... 10
16.3 Validation ... 10
17. Checksum Test Vectors ... 10
17.1 Worked example ... 11
17.2 Error detection ... 11
18. Formatting Rules ... 11
19. Uniqueness and Re-Issuance ... 11
20. Lifecycle States ... 12
21. Security Model ... 12
22. Smart-Contract Boundary ... 12
22.1 Planned Dual On-Chain Registry Control Architecture ... 12
23. Payment Message Integration ... 13
24. Institutional Interoperability ... 13
25. Privacy and Data Protection ... 13
26. Auditability and Governance ... 13
Annex A — Machine Validation Pseudocode ... 13
Annex B — KK Chain / Network Code Allocation Rules ... 14
Annex C — Reference Test Vectors ... 14
Annex D — Standards and External References ... 14
Revision Note — Release 2.0 ... 15
Working Revision of 26 September 2026 ... 16

1. Original Design Intent: Why Osnias-ID Is Needed

Osnias-ID was originally designed to provide Osnias Clearing with a compact, chain-independent reference that can be used consistently across a multi-chain clearing environment. Release 2.0 preserves that purpose while moving two safety-critical clearing attributes — currency and cycle — into the identifier itself.

1.1 Chain-independent clearing reference

The identifier does not depend on the addressing system, account model or technical conventions of any specific blockchain. The technical wallet or escrow address may change under controlled procedures without changing the legal identity behind the account.

1.2 Compact messaging and deterministic routing

Instead of transporting complete identity records or repeated chain-specific metadata, the protocol may transport the OID and resolve detailed participant information through the registry. Release 2.0 additionally allows the receiving component to derive the intended currency rail and clearing cycle directly from the OID before registry lookup or execution.

1.3 Prevent monetary and temporal routing collisions

A clearing account that is valid for one currency or one clearing-room cycle must not be silently admitted into another. Release 2.0 therefore treats currency and cycle as protocol-level invariants rather than as optional database metadata.

RELEASE 2.0 DESIGN OBJECTIVE

A clearing account is identified not only by its registry allocation, but also by the currency rail and clearing cycle for which its escrow relationship was created.

2. Purpose and Scope

- Participant and clearing-account identification and registry lookup.
- Binding between an OID and authorized technical address(es).
- Binding of the OID to an ISO 4217 numeric currency code.
- Binding of the OID to an authorized clearing-cycle duration expressed in weeks.
- Admission control for escrow, payment, clearing and settlement requests.
- Audit trail for identifier creation, wallet binding, lifecycle changes and routing checks.

3. Status of the Specification

This document is Book I of the Osnias Clearing Technical and Functional Reference Corpus. It is a working revision, dated 26 September 2026, of the Release 2.0 specification published on 11 September 2026 [12]. The 19-digit format and checksum generation rule are unchanged. Release 2.0 introduced the transition from the 16-digit Release 1.0 format. This revision clarifies validation, representation and registry semantics; it does not claim external certification, regulatory approval or a new repository deposit.

4. Identifier Structure — Release 2.0

KK-NNN-MMM-WW-SSSSSSS-XX

Field	Length	Authority	Meaning
KK	2	Osnias Clearing	Escrow-capable blockchain / network code.
NNN	3	Osnias Clearing	Authorized server / escrow-node code.
MMM	3	ISO 4217 / Osnias validation	Numeric ISO 4217 currency code used by the clearing account.
WW	2	Osnias Clearing	Clearing cycle in whole weeks; e.g. 01, 04, 13.
SSSSSSS	7	Registry server	Server-assigned sequential or controlled unique account value within the KK/NNN/MMM/WW namespace.
XX	2	Registry server	MOD-97 checksum calculated over the preceding 17 payload digits.

The canonical display contains 19 decimal digits excluding separators. Leading zeroes are significant in the human-readable form. The 17-digit payload is KK || NNN || MMM || WW || SSSSSSS. XX completes the identifier.

5. Currency and Clearing-Cycle Binding

MMM and WW are not descriptive labels. They are safety-critical routing fields. They are established when the clearing account and its associated escrow relationship are created.

5.1 Currency field MMM

MMM contains the three-digit numeric ISO 4217 code of the currency rail. Examples include 840 for USD, 978 for EUR, 756 for CHF and 156 for CNY. The current acceptance list is maintained by Osnias Clearing. An OID must not be admitted into a clearing rail whose configured currency differs from MMM. [1, 2]

MMM identifies the clearing account’s currency of denomination. It does not identify a collateral token, issuer, contract address or settlement instrument. The escrow arrangement must record the applicable currency of denomination separately from the identity and eligibility of the collateral or settlement asset. Any valuation, conversion or collateral-acceptance rule is governed by the relevant clearing-room and node arrangements; the OID alone does not establish asset equivalence.

5.2 Clearing-cycle field WW

WW expresses the clearing cycle in whole weeks. Examples: 01 = 7 days, 04 = 28 days, 13 = 91 days. Only cycles explicitly enabled by Osnias Clearing are valid. The field does not itself open a clearing room; it binds the account to the authorized cycle configuration.

WW identifies a cycle duration, not a specific dated cycle instance or a unique clearing room. Admission must also validate the target room and cycle instance against the authoritative registry and room configuration.

CLEARING ADMISSION INVARIANT
OID currency = configured currency of denomination of the associated escrow arrangement = clearing-rail currency; OID cycle = configured escrow cycle = clearing-room cycle. Any mismatch must fail closed before mint, payment admission or settlement. This invariant does not equate the currency code with a particular collateral token or settlement asset.

A single legal entity may therefore hold several OIDs in the same currency if it participates in different clearing cycles, or several OIDs for different currencies within the same cycle.

6. Identity and Addressing Are Separate Layers

An OID identifies a clearing account. A wallet address identifies a technical destination or signing account. A legal-entity record identifies the company or institution behind one or more clearing accounts. These layers must remain separate.

- Legal identity: company or institution and its recognized external identifiers.
- Osnias-ID: clearing-account identifier with embedded currency and clearing cycle.
- Wallet / escrow address: technical execution endpoint.
- KYC/KYB evidence: controlled identity and verification data.

7. Registry Architecture

The Osnias Registry remains the authoritative resolution layer between an OID and the participant's authorized operational data. Release 2.0 deliberately does not move legal identity or KYC/KYB evidence on-chain merely because the OID itself can be encoded on-chain.

- Canonical OID and decoded KK, NNN, MMM, WW and SSSSSSS fields.
- Current authorized wallet and escrow address bindings.
- Participant status: active, suspended, revoked, pending or archived.
- Identity-verification status and timestamp.
- Optional external identifiers such as LEI, UID/CHE or national company registration references.
- Creation, update, revocation and historical binding audit records.

The registry must distinguish immutable historical evidence from the current operational state used for admission. A historical binding does not establish that an address remains authorised or that a clearing account is currently ACTIVE. The planned on-chain model is described in Section 22.1.

PRIVACY BOUNDARY
Personal or confidential KYC/KYB evidence should not be written to a public blockchain merely because the OID is used on-chain or in a clearing message.

8. Access Control: Osnias-ID Required for Clearing Participants

1. Parse the 19-digit OID and validate its structure.
2. Validate the MOD-97 checksum.
3. Decode and compare MMM with the configured currency of denomination of the target escrow arrangement and clearing rail.
4. Decode and compare WW with the target clearing-room cycle.
5. Resolve the OID in the authoritative registry and verify ACTIVE status.
6. Verify the wallet / escrow binding and any lender, network, jurisdiction or transaction-specific eligibility rule.
7. Only then admit the operation.

FAIL-CLOSED PRINCIPLE
A structurally valid OID with a correct checksum is insufficient. Currency, cycle, registry state and operational address bindings must all be valid.

9. One Legal Entity, Multiple Osnias-ID Clearing Accounts

An OID is not a company identifier. It is the identifier of a clearing account linked to an identified participant. A company may therefore hold multiple OIDs while all remain linked in the registry to the same legal entity.

Legal entity	Currency	Cycle	Example role
Company A	USD (840)	13 weeks	ORUSD 91-day clearing account
Company A	USD (840)	04 weeks	USD 28-day clearing account
Company A	EUR (978)	13 weeks	OEURO 91-day clearing account

10. Cross-Chain Use

OID remains an internal interchain clearing reference. KK identifies the network or rail allocation. The identifier does not itself grant cross-chain execution rights; each chain-specific action remains subject to the relevant contracts, controller permissions, escrow state and registry state.

11. External Identifier Interoperability

External identifiers identify or describe the relevant legal entity; they do not replace the Osnias clearing-account identifier. LEI references follow the GLEIF framework [6]; Swiss UID/CHE references concern enterprise identification [10]. Any regulatory-status metadata must record its source and verification date. FINMA register information is a point-in-time reference and does not imply endorsement of Osnias [7].

External reference	Possible use	Boundary
LEI	Legal-entity cross-reference	Osnias does not issue LEIs.
Swiss UID / CHE	Swiss enterprise cross-reference	Osnias does not issue UID numbers.
FINMA register data	Regulatory-status metadata where publicly applicable	No FINMA endorsement or authorization is implied.
Other national identifiers	Jurisdiction-specific matching	Subject to source and privacy constraints.

12. Relationship to ISO 20022 and ISO 4217

Osnias-ID is not an ISO 20022 identifier and is not claimed to be an ISO 20022-certified data element. ISO 20022 concepts may be used as interoperability references when mapping participant, account, agent or payment-message data. [3, 4]

Release 2.0 uses the three-digit numeric ISO 4217 currency code as the semantic basis of MMM. This use does not make Osnias-ID an ISO standard, nor does it imply endorsement by ISO. Osnias Clearing remains responsible for deciding which currency codes are enabled as clearing rails. [1, 2]

13. Allocation Hierarchy

Level	Field	Authority	Control objective
1	KK	Osnias Clearing	Allocate and maintain network / rail codes.
2	NNN	Osnias Clearing	Allocate authorized server / escrow-node codes.
3	MMM	ISO 4217 reference + Osnias policy	Bind account to an enabled currency rail. Chosen by the end-user from the currencies enabled by Osnias; validated at issuance.
4	WW	Osnias Clearing	Bind account to an enabled clearing cycle. Chosen by the end-user from the cycles enabled by Osnias; validated at issuance.
5	SSSSSSS	Registry server	Issue unique account values within the KK/NNN/MMM/WW namespace.
6	XX	Registry server	Calculate deterministic integrity check digits.

AUTHORITY INVARIANT

KK and NNN are assigned by Osnias. MMM must be an enabled ISO 4217 numeric currency code. WW must be an enabled clearing cycle. SSSSSSS is assigned by the registry server. XX is calculated by the registry server. No field may be silently reassigned by a wallet, user interface or downstream smart contract.

14. Machine Representation: 19 Decimal Digits in uint64

The complete canonical OID contains 19 decimal digits. The maximum possible 19-digit value is 9,999,999,999,999,999,999, which is below the uint64 maximum of 18,446,744,073,709,551,615. Every canonical Release 2.0 OID therefore fits in an unsigned 64-bit integer.

Representation	Size	Purpose
Canonical human display	19 decimal digits + separators	Readable exchange and documentation.
Canonical numeric value	19 decimal digits, no separators	Deterministic parsing and checksum validation.
Machine storage	uint64	64-bit value capacity (8 bytes); actual storage and transport size depend on the implementation and encoding.
Hexadecimal rendering	up to 16 hex digits	Diagnostic representation of the same uint64 value.

Leading zeroes are part of the canonical display but are not preserved by ordinary integer formatting. Any component converting from uint64 back to the human form must left-pad the numeric value to exactly 19 decimal digits before inserting separators.

The uint64 bound describes value capacity, not a universal blockchain storage or message size. In the standard Solidity ABI, a uint64 argument is zero-padded to 32 bytes. Packed encoding and contract storage layout are separate implementation choices [9].

ENCODING PRINCIPLE

Decimal and hexadecimal are two presentations of the same 64-bit integer. The security semantics are carried by the field layout and validation rules, not by the choice of display radix.

15. API and Resolution Model

Example lookup request:

```
GET /api/osnias-id/01-001-840-13-0000001-18
```

Example response:

```
{
  "osnias_id": "01-001-840-13-0000001-18",
  "status": "ACTIVE",
  "checksum_valid": true,
  "currency_numeric": "840",
  "cycle_weeks": 13,
  "wallets": [],
  "external_identifiers": {}
}
```

API requests and responses must carry the OID as a JSON string, preserving its decimal digits and leading zeroes. The example above is an illustrative response shape, not evidence that a live endpoint is deployed or that an empty wallet list permits admission. Do not convert the OID through a floating-point number; use exact integer arithmetic or digit-by-digit modular arithmetic. JSON interoperability with commonly used binary64 implementations is limited to exact integers in the range −(2^53−1) to 2^53−1 [8].

16. Normative Checksum Specification

Release 2.0 retains the two-digit MOD-97 integrity rule, extended to the new 17-digit payload.

16.1 Input

Accept only the canonical hyphenated form or exactly 19 ASCII decimal digits, as specified in Section 18. Remove the five separators only after checking their positions. Let B be the 17-digit decimal string: B = KK || NNN || MMM || WW || SSSSSSS.

16.2 Calculation

Append two zero digits to B, producing B00.

Interpret B00 as a non-negative decimal integer.

Compute r = B00 mod 97.

Compute XX = 98 - r.

Encode XX as exactly two decimal digits, including a leading zero if required.

16.3 Validation

Let I be the complete 19-digit identifier B || XX. Validation must recompute the expected XX using Section 16.2 and compare it exactly with the received two-digit value. The generated XX is in the range 02–98. The additional condition I mod 97 = 1 is a consistency check; it must not replace the exact comparison, because a non-canonical two-digit value can satisfy the same remainder condition.

For example, 01-001-840-13-0000007-97 has the canonical checksum. Replacing 97 with 00 still yields I mod 97 = 1, but must be rejected because 00 differs from the recomputed checksum. This clarification aligns Section 16.3 with the exact comparison already required by the Release 2.0 pseudocode.

STANDARDIZATION NOTICE
This is the normative Osnias-ID checksum rule for Release 2.0 and this working revision. It uses MOD-97 arithmetic and is independently reproducible. ISO/IEC 7064:2003 is an informative reference [5]; its citation is not a claim of formal conformity. The checksum is not a cryptographic signature, MAC or proof of registry membership.

17. Checksum Test Vectors

KK	NNN	MMM	WW	SSSSSSS	XX	Canonical display
01	001	840	13	0000001	18	01-001-840-13-0000001-18
01	001	978	04	0000002	19	01-001-978-04-0000002-19
02	023	756	01	1234567	83	02-023-756-01-1234567-83
12	034	156	13	7654321	54	12-034-156-13-7654321-54
99	099	840	52	9999999	28	99-099-840-52-9999999-28

17.1 Worked example

For OID 01-001-840-13-0000001-18:

```
B     = 01001840130000001
B00   = 0100184013000000100
r     = B00 mod 97 = 80
XX    = 98 - 80 = 18
I     = 0100184013000000118
I mod 97 = 1  -> VALID
```

17.2 Error detection

The checksum is intended to detect common transcription and data-entry errors, including accidental changes to the embedded currency or cycle digits. It does not replace registry, escrow, currency or cycle validation.

The vectors in Section 17 and Annex C establish format and checksum validity only. Their use does not assert that the illustrated KK, NNN, currency or cycle is enabled in production, or that the account exists in a registry.

18. Formatting Rules

- Canonical display: KK-NNN-MMM-WW-SSSSSSS-XX.
- Canonical numeric value: 19 decimal digits without separators.
- Leading zeroes are significant in the canonical display.
- Spaces are not permitted in the canonical value.
- The checksum is calculated from the first 17 payload digits only.
- Formatting separators are excluded from checksum calculation.
- When converting a uint64 to display form, left-pad to 19 decimal digits before field splitting.

Accepted input syntax is either [0-9]{19} or [0-9]{2}-[0-9]{3}-[0-9]{3}-[0-9]{2}-[0-9]{7}-[0-9]{2}, matched against the whole input. Reject misplaced separators, spaces, signs and non-ASCII digits. The allocation rules, enabled codes, current registry state and target context must then be checked independently.

19. Uniqueness and Re-Issuance

The seven-digit SSSSSSS field provides 10,000,000 local values for each KK × NNN × MMM × WW namespace. If a server approaches exhaustion for a given currency and cycle, Osnias may allocate another NNN server / node code rather than recycling previously issued identifiers.

The capacity of 10,000,000 includes 0000000. Any values reserved by allocation policy reduce the usable capacity and must be documented. Historical values remain reserved; allocation of a new node code does not transfer the legal or operational responsibilities of an existing node.

NON-REUSE PRINCIPLE
A previously issued OID should remain historically unique even after closure, revocation or migration. Revoked values should remain reserved or tombstoned.

20. Lifecycle States

State	Meaning	Clearing admission
PENDING	Record created but not fully activated.	Rejected
ACTIVE	Account, currency/cycle binding and wallet binding valid.	Eligible, subject to other rules
SUSPENDED	Temporarily disabled.	Rejected
REVOKED	Permanently withdrawn.	Rejected
ARCHIVED	Historical record retained for audit.	Rejected

21. Security Model

- Fail closed when registry resolution is unavailable or ambiguous.
- Validate the 19-digit structure and checksum before expensive resolution steps.
- **Compare MMM against the configured currency of denomination of the escrow arrangement and clearing room before admission.**
- Compare WW against the escrow and clearing-room cycle before admission.
- Require ACTIVE registry state independently of checksum validity.
- Verify current wallet / escrow bindings.
- Prevent client-side generation of authoritative OID values.
- Log creation, binding, replacement, suspension, revocation and configuration changes.
- Protect private KYC/KYB data independently from public OID resolution.

DEFENSE IN DEPTH
Currency binding, cycle binding, checksum validation, registry membership and escrow/address authorization are separate controls. Compromise of one layer must not silently authorize an incompatible clearing operation.

22. Smart-Contract Boundary

Where an OID is referenced by a smart contract, Release 2.0 permits a compact uint64 representation. The contract may decode the semantic fields required for deterministic routing and admission checks, especially MMM and WW. Smart contracts should not become the authoritative repository of private legal or KYC/KYB information. A uint64 value does not imply an eight-byte standard ABI encoding; see Section 14 [9].

A contract that receives an OID for a clearing operation should reject the operation if the decoded currency or cycle differs from the configured rail, even if an upstream application or SQL database reports a different mapping.

22.1 Planned Dual On-Chain Registry Control Architecture

The planned Osnias-ID control architecture comprises two complementary on-chain registry components. On the escrow network, a registry oracle would verify and record each newly admitted escrow account and associate it with its Osnias-ID. On Sei, a

second registry would record the corresponding Osnias-ID and local 0x clearing-address association. This section specifies a design direction and does not establish deployment or audit status.

The planned registries retain an append-only historical record. A recorded event must not be overwritten or reassigned by a later registry operation. Changes of authorised address, suspension, revocation and other lifecycle changes are represented by subsequent authorised events that preserve the earlier evidence. Resolution must return the current effective binding and status under explicit ordering and authorisation rules; an old binding must not remain eligible merely because it is still recorded. This application-level rule is subject to the finality assumptions of the underlying networks and is not an unconditional claim that a blockchain can never reorganise.

The dual-registry design complements the checksum, currency, cycle and registry-status controls. Before operational use, a dedicated specification must define the authoritative source for each state, event ordering, cross-registry reconciliation, finality requirements and handling of divergence. Resolution must fail closed when the effective state is unavailable or ambiguous. The detailed oracle architecture, interfaces, registration workflow, implementation and source code remain the subject of a subsequent technical publication.

23. Payment Message Integration

Payment messages may carry the sender and recipient Osnias-IDs as clearing-account references. Implementations must preserve the exact identifier strings and validate the currency, cycle, current registry status and authorised address bindings before admission. A message must also identify its target clearing room and dated cycle instance; these are not fully encoded by MMM and WW alone.

Where ISO 20022 interoperability is intended, the implementation must document the selected message definitions, versions and field mappings [3, 4]. This specification does not register a new ISO 20022 data element or establish message conformity. Carrying an Osnias-ID in a payment message does not execute settlement or prove legal finality.

24. Institutional Interoperability

For banks, escrow providers, lenders or regulated partners, OID can operate as the internal clearing-network key used to reconcile technical addresses with institutionally recognized counterparties. Currency and cycle binding allows a partner to determine that an account was created for the same monetary and temporal clearing context as the requested operation.

25. Privacy and Data Protection

- Apply data minimization and store only fields required for the clearing purpose.
- Separate public identifier data from confidential identity evidence.
- Avoid placing personal addresses, e-mail addresses, telephone numbers or identity documents directly on-chain.
- Use access controls and purpose limitation for registry queries containing non-public data.
- Preserve audit evidence where legally or operationally necessary without exposing unnecessary personal data.

Keeping identity evidence off-chain does not by itself make an OID or wallet association anonymous. An identifier may remain personal data where it can be linked to an identifiable natural person. Assess linkability, access rights, retention and the applicable data-protection obligations for both public and controlled registry data. The CNIL’s guidance on blockchain and personal data provides an informative reference [11].

26. Auditability and Governance

- Structural audit: 19-digit format, field lengths, uint64 bounds and checksum.
- **Routing audit: MMM/WW decoding and comparison with the escrow denomination, clearing-room and dated-cycle configuration.**
- Registry audit: allocation authority, uniqueness, lifecycle and wallet binding history.
- Security audit: privileged administration, API authorization, fail-closed behavior and logging.
- Protocol audit: enforcement of “no valid OID, no clearing request” and currency/cycle invariants.

Annex A — Machine Validation Pseudocode

```
function validateOsniasId(input, expectedCurrency, expectedCycle):
    if not canonicalSyntax(input): return false
    digits = removeCanonicalHyphens(input)
    payload = digits[0:17]
    receivedXX = digits[17:19]
    kk  = digits[0:2]
    nnn = digits[2:5]
    mmm = digits[5:8]
    ww  = digits[8:10]
    sequence = digits[10:17]

    remainder = mod97Digits(payload + "00")
    expectedXX = leftPad2(98 - remainder)
    if receivedXX != expectedXX: return false
    if mod97Digits(digits) != 1: return false
    if kk == "00": return false
    if mmm != expectedCurrency: return false
    if exactInteger(ww) != expectedCycle: return false

    oid64 = uint64(exactInteger(digits))
    record = registry.resolveCurrentOrFail(oid64)
    if record is missing or unavailable or ambiguous: return false
    if record.status != ACTIVE: return false
    if not record.authorizesCurrentAddress(): return false
    return true
```

canonicalSyntax applies the whole-input ASCII patterns in Section 18. mod97Digits processes decimal digits with r = (10 × r + digit) mod 97, starting at zero; no floating-point conversion is used. expectedCurrency is a three-digit string and expectedCycle is an integer. Production admission must additionally verify enabled KK/NNN/MMM/WW values, the target escrow and asset, clearing room, dated cycle instance, network, lender and any other applicable eligibility conditions. Any failed, unavailable or ambiguous check must reject the operation.

Annex B — KK Chain / Network Code Allocation Rules

- KK contains exactly two decimal digits. KK=00 remains permanently reserved as null/unassigned.
- KK is allocated and controlled by Osnias Clearing as the code identifying a supported blockchain/network domain.
- Detailed network metadata is maintained separately so it can evolve without changing the OID field width.
- Deprecated or retired KK values remain historically documented and must not be silently reassigned.

Annex C — Reference Test Vectors

01-001-840-13-0000001-18 FORMAT AND CHECKSUM VALID

01-001-978-04-0000002-19 FORMAT AND CHECKSUM VALID

02-023-756-01-1234567-83 FORMAT AND CHECKSUM VALID

12-034-156-13-7654321-54 FORMAT AND CHECKSUM VALID

99-099-840-52-9999999-28 FORMAT AND CHECKSUM VALID

01-001-840-13-0000007-97 FORMAT AND CHECKSUM VALID

01-001-840-13-0000007-00 INVALID: non-canonical checksum, despite remainder 1

A checksum-valid OID must still be rejected when the registry is unavailable, missing, ambiguous, suspended or revoked, or when currency, cycle, room or address authorisation does not match the target operation.

Implementations should include these vectors in automated tests. A single-digit change, including a currency or cycle digit, should normally invalidate the checksum.

Annex D — Standards and External References

[1] ISO. ISO 4217 — Currency codes. Reference for the numeric currency field.

https://www.iso.org/iso-4217-currency-codes.html

[2] SIX. Global Financial Data Standards — ISO 4217 Maintenance Agency and current currency and funds lists.

https://www.six-group.com/en/products-services/financial-information/market-reference-data/data-standards.html

[3] ISO. ISO 20022-1:2026 — Financial services — Universal financial industry message scheme — Part 1: Metamodel.

https://www.iso.org/standard/20022-1

[4] ISO. ISO 20022-3:2026 — Financial services — Universal financial industry message scheme — Part 3: Modelling.

https://www.iso.org/standard/20022-3

[5] ISO/IEC. ISO/IEC 7064:2003 — Information technology — Security techniques — Check character systems. Informative reference.

https://www.iso.org/standard/31531.html

[6] GLEIF. The Legal Entity Identifier — ISO 17442-based legal-entity identification.

https://www.gleif.org/en/organizational-identity/lei-vlei/the-legal-entity-identifier-lei

[7] FINMA. Authorised institutions, individuals and products. Regulatory-status reference.

https://www.finma.ch/en/finma-public/authorised-institutions-individuals-and-products/

[8] Bray, T., ed. RFC 8259 — The JavaScript Object Notation (JSON) Data Interchange Format. December 2017, Section 6.

https://www.rfc-editor.org/rfc/rfc8259.html#section-6

[9] Solidity documentation. Contract ABI Specification — Formal Specification of the Encoding.

https://docs.soliditylang.org/en/latest/abi-spec.html

[10] Swiss Federal Tax Administration. Unternehmens-Identifikationsnummer (UID).

https://www.estv.admin.ch/de/unternehmens-identifikationsnummer-uid

[11] CNIL. Blockchain et RGPD : quelles solutions pour un usage responsable en présence de données personnelles ?

https://www.cnil.fr/fr/technologies/blockchain-et-rgpd-quelles-solutions-pour-un-usage-responsable-en-presence-de-donnees-personnelles

[12] Bouzon, Denis. Osnias-ID: A Clearing ID Account as Inter-chain Identifier. Release 2.0, 11 September 2026. HAL hal-05746489, as identified on the published PDF cover.

https://www.osnias-clearing.com/osniasid/Osnias-ID_White_Paper_Release_2.0-HAL.pdf

Web references consulted on 26 September 2026. Operational currency and regulatory-status checks must use the relevant current source and record the date of verification. References [3]–[5] identify standards; they do not establish a conformity assessment of Osnias.

Osnias-ID remains a proprietary project identifier unless and until an external standards body, regulator or other competent authority explicitly determines otherwise. References to external standards are interoperability references and do not imply certification or endorsement.

Revision Note — Release 2.0

- Released as Version 2.0 on 11 September 2026.
- Major format transition from 16 to 19 decimal digits.
- Replaced KK-NNN-AAAAAAAAA-XX with KK-NNN-MMM-WW-SSSSSSS-XX.
- Added MMM as the three-digit ISO 4217 numeric currency field.
- Added WW as the two-digit clearing-cycle field expressed in whole weeks.

- **Replaced the former nine-digit account field with a seven-digit sequence within each KK × NNN × MMM × WW namespace, providing 10,000,000 possible sequence values before policy reservations.**
- Defined the complete 19-digit identifier as compatible with uint64 / 8-byte machine storage.
- **Bound the OID currency to the configured currency of denomination of the escrow arrangement and clearing rail; collateral and settlement-asset eligibility remain separate controls.**
- Defined the mandatory cycle invariant: OID cycle = escrow cycle = clearing-room cycle.
- Extended the MOD-97 checksum payload from 14 to 17 digits and published new Release 2.0 test vectors.
- Retained registry validation, ACTIVE lifecycle status and address/escrow authorization as independent controls.
- Clarified that checksum, currency and cycle fields do not replace legal/KYC registry controls.
- Retained CC BY 4.0 publication licensing and explicit no-certification / no-endorsement boundaries.

• Introduced the planned dual on-chain registry design for escrow-side and Sei-side bindings. The 26 September working revision clarifies the distinction between historical evidence and current effective authorisation. Detailed oracle and smart-contract architecture and source code remain reserved for a subsequent publication.

Working Revision of 26 September 2026

• The 19-digit layout and checksum generation rule remain unchanged. The five original checksum vectors are retained.

• Clarified exact checksum comparison, input syntax, exact arithmetic and fail-closed resolution; added a non-canonical checksum rejection example.

• Distinguished currency denomination from asset identity, and uint64 capacity from API and ABI encodings.

• Clarified the planned registry lifecycle model, restored Section 23 and aligned the table of contents.

• Added traceable references and corrected editorial inconsistencies. The published source retains its original Release 2.0 date and HAL reference; no new DOI or HAL identifier is assigned by this working revision.

MAJOR RELEASE RATIONALE

Release 2.0 moves monetary and temporal clearing compatibility from mutable application metadata into the canonical OID itself, while preserving compact machine representation and the separation between legal identity, clearing account and technical execution address.

END OF DOCUMENT