

OEURO Token — Technical White Paper & Audit Reference

Public Technical Documentation · Osnias Clearing · www.osnias-clearing.com

Drafted by / Rédacteur : Denis Bouzon · ORCID 0009-0007-8894-8902

Publication intent: this technical white paper is intended for deposit in a public scholarly/technical repository with assignment of a Digital Object Identifier (DOI). The DOI reference will be added after formal deposit.

Version 2.0 · Publication date: 31 August 2026 · OEURO Token · © 2026 Osnias Clearing. All rights reserved.

Document status: public pre-audit technical architecture draft for developer, integrator and smart-contract review. The exact Solidity source, ABI and flattened source are published separately as companion technical files.

DOCUMENT SCOPE — This white paper is a technical document. It describes the architecture, deployment model, smart-contract behavior, security boundaries, operational roles and audit invariants of the OEURO token. Legal, regulatory, contractual, licensing and compliance matters are outside the scope of this technical white paper and are addressed separately in dedicated Osnias Clearing regulatory and legal documentation. The absence of a regulatory or contractual analysis in this document therefore does not indicate that those matters are ignored; they are intentionally documented in a separate corpus.

AUDIT SCOPE PRINCIPLE

OEURO is a restricted EUR-denominated clearing-register token. Its security model is defined by controlled issuance and destruction, EOA-only circulation, explicit oracle-request boundaries, non-replayable external proofs and manager-gated settlement execution.

1. Technical Role of OEURO

OEURO is designed as a clearing representation denominated in euro. The supplied Solidity contract implements an ERC-20 accounting instrument with six decimals while deliberately removing standard delegated-transfer pathways and rejecting ordinary transfers to deployed smart contracts.

The intended normal circulation path is direct peer-to-peer transfer between user-controlled externally owned accounts (EOAs). Mint and burn operations are reserved to the OEURO manager, either manually or after an authorized oracle has created a verifiable on-chain request.

CORE REGISTER INVARIANT

OEURO must remain a clearing-register instrument. The protocol must not expose a native execution path that turns OEURO into a freely routable smart-contract market asset or that allows an oracle to mint or burn directly.

2. Deployment Model: Sei Testnet Reference vs Production

Sei Testnet — public reference, integration and audit environment

OEURO is publicly deployed on Sei Testnet as the current reference implementation. This deployment provides the public EVM test environment for developer access, reproducible integration testing, wallet testing, explorer verification and independent review using the same Sei execution family intended for production.

Sei EVM Mainnet — intended canonical production environment

The intended production deployment remains a distinct Sei EVM Mainnet instance. The Testnet deployment is a public technical reference only. A production contract must be independently deployed with production addresses, privileged roles, oracle configuration, multisignature manager controls and operational-security procedures.

CANONICAL STATE PRINCIPLE

The Sei Testnet deployment is the public reference implementation. A future Sei Mainnet production deployment must be independently identified and must not inherit balances, proofs or administrative rights from the Testnet instance.

2A. Network Locality Invariant

The deployed OEURO contract contains no bridge, OFT, wrapping, migration or cross-chain token-transfer implementation. Its sourceChain and externalRef oracle fields identify external evidence; they do not move OEURO between chains.

NETWORK LOCALITY INVARIANT

External evidence may inform a clearing request, but OEURO issuance, destruction and circulation remain local to the specific deployed OEURO contract instance. The current public reference instance is on Sei Testnet.

Public Sei Testnet Deployment Registry

Field	Sei Testnet reference	Audit status
Network	Sei Testnet	Public EVM test reference
EVM Chain ID	1328 (0x530)	Deployment network identifier
Native gas asset	SEI	Used for transaction gas
Contract name	OsniasClearing_Oeuro	Final deployable contract
Contract address	0x49A0D9D353e9b9860B137DFAA7f24ab83E14c7f7	Verified public reference address
Creation transaction	0x97dfa508843181c4649215389c6be24f626b3510bb507d531a73e36255c37043	Contract creation transaction
Explorer	https://testnet.seiscan.io	Sei Testnet explorer
Compiler	Solidity v0.8.22+commit.4fc1097e	Same compiler family used for verification
Optimizer	Enabled · 200 runs	Matches deployed compilation settings
EVM target	Paris	Matches deployed compilation settings
Explorer license	No License (None)	No open-source license granted by explorer selection
Verification	Matching Bytecode and ABI	SeiScan verification successful
Source reference	oeuro_sei_testnet.sol	Sei Testnet Solidity source
Flattened reference	OEURO flattened Solidity source	Single-file verification source
Production status	Not Mainnet production	Separate Sei Mainnet deployment required

Reference deployment verified on SeiScan: source compilation produced matching deployed bytecode and ABI for the address above.

SOURCE PUBLICATION PRINCIPLE

The exact Solidity source, ABI and flattened contract are companion technical files. The public Sei Testnet source has been verified on SeiScan with matching bytecode and ABI. This white paper describes architecture, security boundaries and audit invariants rather than duplicating implementation code.

3. Token Identity and Numeric Model

Property	Current contract value	Audit interpretation
Solidity	^0.8.22	Compile reproducibly against the pinned OpenZeppelin dependency set.
Token name	Osnias Euro Clearing	ERC-20 display name encoded by the constructor.
Symbol	OEuro	Public ticker encoded by the contract.
Technical decimals	6	On-chain precision; interfaces may display fewer decimals.
Protocol fee	15 bps = 0.15%	Deducted from the transferred amount.
Minimum gross P2P transfer	0.501000 OEURO	501,000 base units.
Oracle request validity	24 hours	Execution window from request creation time.
Administration	Ownable2Step	Two-step ownership-transfer mechanism.

The public source declares `PROTOCOL = "Osnias Clearing"`, `PROTOCOL_VERSION = "1.0"` and `TOKEN_FULL_NAME = "Osnias Euro Clearing"`. These metadata constants identify the protocol and token implementation; they do not by themselves establish a bank deposit, a legal-tender claim, a redemption promise or an external euro reserve.

4. P2P Transfer Model

Ordinary P2P transfers follow a gross-debit / net-receipt model. The sender is debited by the exact value submitted. The protocol fee is calculated as $\text{value} \times 15 / 10,000$ and credited to the designated feeRecipient. The recipient receives value minus the fee.

Example	Amount
Gross transfer request	1,000.000000 OEURO

Protocol fee (0.15%)	1.500000 OEURO
Recipient receives	998.500000 OEURO
Sender total debit	1,000.000000 OEURO

The feeRecipient is exempt from the fee when transferring collected OEURO. This prevents recursive fee generation when protocol fees are moved.

TRANSFER INVARIANT

For an ordinary user transfer, sender debit = gross value; recipient credit + feeRecipient credit = gross value. No additional OEURO is created by the fee mechanism.

5. EOA-Only Circulation and Smart-Contract Boundary

OEURO deliberately disables standard ERC-20 delegated-transfer behavior. approve() reverts. transferFrom() reverts. The OpenZeppelin v5 _update() hook rejects ordinary transfers to addresses with deployed contract bytecode.

- EOA → EOA transfer: permitted, subject to minimum amount, fee rules and recipient blacklist.
- EOA → smart contract: rejected.
- approve(): rejected.
- transferFrom(): rejected.
- Mint to smart contract: rejected by manager mint and oracle-request account checks.
- Burn: permitted only through manager-controlled paths.

CLEARING-ONLY CIRCULATION INVARIANT

The intended design is stronger than the absence of a native swap function: OEURO does not expose the approval, transferFrom or ordinary contract-recipient pathways normally required by standard smart-contract routing.

5A. Clearing / Custody Boundary

The supplied OEURO token contract contains no customer-deposit, custody, escrow, lending, collateral-valuation, collateral-ratio, withdrawal or bank-account logic. Any future partner-lender or collateral architecture therefore exists outside the OEURO token contract and requires separate technical and legal documentation.

NON-CUSTODY CONTRACT PRINCIPLE

The OEURO token contract is a clearing register. The supplied source does not provide a custody or lending layer and does not identify a specific euro collateral asset.

For the external escrow and collateral layer associated with EUR-denominated clearing, Osnias Clearing recommends EURC issued by Circle as the preferred collateral asset. This is an architectural and operational recommendation for partner escrow arrangements; it is not a dependency, reserve mechanism or enforceable condition embedded in the OEURO smart contract. OEURO remains a clearing register, while EURC custody, collateralization, redemption and escrow execution remain outside the token contract.

6. Restricted Recipient Domain

The contract maintains a restricted-recipient mapping named orusdBlacklisted in the current source. Despite this legacy identifier, it is applied to OEURO transfers in the deployed contract logic.

- Initial manager: restricted from receiving ordinary OEURO through the transfer path.
- GasPool: restricted from receiving OEURO.
- Authorized oracle: automatically restricted from receiving OEURO.
- Former oracle: remains restricted after oracle authorization is removed.
- Protocol fee recipient: intentionally eligible to receive OEURO so that transfer fees can be collected.

ROLE-SEPARATION PRINCIPLE

Infrastructure addresses that validate, manage or sponsor the system should not accumulate user clearing balances merely because they hold privileged operational roles.

7. Oracle Request Architecture

An authorized oracle does not mint or burn OEURO. Its authority is limited to creating a request after observing or validating an external event. The request is written on-chain and becomes inspectable before any supply change occurs.

Request field	Purpose
Request category	MINT or BURN
Lifecycle status	NONE / PENDING / EXECUTED / REJECTED
Submitting oracle	Authorized source of the request
Affected account	EOA whose OEURO balance may change
Amount	Requested OEURO quantity
sourceChain	Compact identifier of the observed external source or system
externalRef	Reference to the external evidence
createdAt	Timestamp used to enforce the 24-hour validity window

ORACLE AUTHORITY INVARIANT

An oracle may attest and request. It may not settle. Only the OEURO manager can execute or reject an oracle request.

8. External-Proof Uniqueness and Anti-Replay

Before a request is stored, the contract derives a proof key from sourceChain and externalRef. If the same proof key has already been used, a second request reverts. The proof remains consumed even if the request is later rejected.

Each request also receives a unique requestId derived from the submitting oracle, request type, account, amount, external evidence fields and a sequential nonce.

ANTI-REPLAY INVARIANT

A given (sourceChain, externalRef) pair can create at most one OEURO oracle request. Rejection does not free the proof for reuse.

9. Request Lifecycle and Settlement

A valid oracle request begins in PENDING state. The manager may execute it within the 24-hour validity period or reject it. Expired requests cannot be executed through the normal execution path.

- MINT request: valid execution increases the designated account balance and total supply by the approved amount.
- BURN request: valid execution reduces the designated account balance and total supply by the approved amount, subject to sufficient balance.
- The request status is set to EXECUTED before the internal mint/burn operation completes, preventing normal replay of the stored request.
- A rejected request is permanently marked REJECTED.
- An expired request remains part of the audit history but is not executable.

SETTLEMENT FINALITY PRINCIPLE

Every supply-changing oracle operation must correspond to one identifiable request and one terminal status transition.

10. Manual Manager Mint and Burn

The supplied OEURO reference implementation contains manager-controlled manual mint and burn functions. These constitute the strongest privileged supply authority in the contract.

Manual mint is limited to a non-zero EOA recipient and a positive amount. Manual burn is manager-controlled and can apply to a specified account, subject to non-zero address, positive amount and sufficient balance enforced by ERC-20 burn accounting.

AUDIT ATTENTION — PRIVILEGED SUPPLY

The manager can mint to an eligible EOA and burn from an OEURO account through explicit privileged functions. Production operational-security controls must therefore treat manager authority as a critical supply-control role.

11. Administrative Authority Matrix

Action	Oracle	Manager / owner	User
Submit oracle request	Yes, if authorized	No special need	No
Execute oracle request	No	Yes	No
Reject oracle request	No	Yes	No
Manual mint	No	Yes	No
Manual burn	No	Yes	No

Authorize / remove oracle	No	Yes	No
Change feeRecipient	No	Yes	No
P2P transfer	Restricted as recipient role	Normal restrictions apply	Yes
approve / transferFrom	No	No	No

SEPARATION OF DUTIES

Oracle observation and manager settlement are separate trust domains. A compromised oracle alone must not be sufficient to create or destroy OEURO.

12. OEURO / OSNIAS Functional Separation

OEURO is a clearing-register token, whereas OSNIAS serves a separate governance and strategic-reserve function. The OEURO source does not provide a native conversion, liquidity, farming or collateral dependency path linking the two token classes.

- No protocol-native OEURO ↔ OSNIAS conversion function in the supplied OEURO source.
- No OEURO / OSNIAS liquidity-pool logic in the supplied OEURO contract.
- No OSNIAS-dependent OEURO mint or burn condition.
- No OSNIAS administrative role is implemented in the OEURO contract.

CROSS-DOMAIN SECURITY INVARIANT

Governance-token administration and OEURO clearing administration should remain distinct privilege domains.

13. Supply and Accounting Invariants

Invariant	Required property	Manager override
No hidden mint path	Supply increases only through explicit manager mint or execution of a valid MINT request.	Not outside explicit functions
No hidden burn path	Supply decreases only through explicit manager burn or execution of a valid BURN request.	Not outside explicit functions
Oracle non-settlement	Oracle cannot directly mint or burn OEURO.	Not permitted
Proof uniqueness	One external proof pair can be consumed once.	Not permitted
P2P conservation	Ordinary transfer redistributes existing OEURO between recipient and feeRecipient.	Not permitted
No delegated transfer	approve / transferFrom remain disabled.	Not permitted by current code
EOA recipient rule	Ordinary transfers to deployed contracts revert.	Not permitted by current code
No native bridge path	No bridge, OFT, wrapping or migration mechanism exists in the supplied source.	Not implemented

14. Security Validation Pipeline

OEURO is intentionally compact, but its custom ERC-20 restrictions create protocol-specific invariants that must be tested beyond ordinary token compliance. Static analysis supports review; it does not replace unit tests, fuzzing, invariant testing or manual examination.

Security objective	Primary method	Audit interpretation
Public interface exposure	Slither + manual review	Map every externally callable state-changing path against the ABI.
Manager privilege boundaries	Slither + unit tests	Verify every supply and administration function is privilege-gated.
Oracle authorization	Unit + fuzz tests	Unauthorized request creation must always revert.
Proof replay	Invariant tests	The same sourceChain/externalRef pair cannot produce a second request.
Request expiry	Time-based tests	Execution after 24h must revert.
Mint/burn accounting	Invariant tests	Supply change must equal the authorized amount.
Fee conservation	Property tests	sender debit = recipient net + fee.
EOA-only transfer	Integration tests	Ordinary transfers to contracts must revert.
Delegated-transfer lockout	Unit tests	No allowance-based transfer route may be

		available.
Unexpected market route	Adversarial source review	Any executable path enabling unintended contract routing should be treated as an architecture finding.

ADVERSARIAL REVIEW PRINCIPLE

Any hidden, indirect, oracle-bypass or externally triggered path capable of unauthorized mint, burn or ordinary smart-contract routing must be treated as a critical security finding.

15. Recommended Pre-Audit Sequence

- 1. Reproducible Solidity compilation with pinned OpenZeppelin dependencies.
- 2. Slither static analysis focused on privilege, supply, transfer and oracle surfaces.
- 3. Unit tests for mint, burn, fee, blacklist and ownership controls.
- 4. Fuzz and invariant testing for supply conservation and proof replay.
- 5. Sei Testnet integration testing against the verified reference deployment 0x49A0D9D353e9b9860B137DFAA7f24ab83E14c7f7.
- 6. Oracle lifecycle tests covering authorization, replay, rejection and 24-hour expiry.
- 7. Wallet tests for direct EOA-to-EOA transfers on Sei Testnet.
- 8. Sei Mainnet deployment rehearsal with production-like addresses, multisignature manager control and operational procedures.
- 9. Manual security review against the invariants in this white paper.
- 10. Final production deployment review and public registry publication.

16. Mainnet Privileged-Key Security

The current public Sei Testnet implementation uses a simplified privileged-key model suitable for development, testing and public verification. A production Sei Mainnet deployment should replace single-key critical administration with an operational-security architecture appropriate to manager supply authority.

- Production manager authority: multisignature control rather than a single EOA.
- Cold / emergency signer separation: recovery keys stored outside ordinary operational devices.
- Oracle separation: oracle signing infrastructure independent from manager settlement keys.
- Fee-wallet separation: feeRecipient should be distinct from oracle and manager keys unless explicitly justified.
- External registry: authoritative production contract addresses, privileged roles and recovery procedures maintained outside the token contract.
- Incident procedure: documented oracle removal, ownership transfer and emergency operational response.

MAINNET SECURITY PRINCIPLE

A single compromised key should not be sufficient to fabricate an external attestation and execute the corresponding OEURO supply change.

17. Deployed Sei Testnet Source and Mainnet Delta

Category	Meaning in this document	Examples
IMPLEMENTED — Sei Testnet	Present in the supplied Solidity source and ABI.	6 decimals; fee logic; EOA-only recipient restriction; oracle workflow; manager-controlled supply operations; anti-replay; delegated transfers disabled.
CONFIGURATION — Sei Testnet	Deployment-specific addresses and legacy comments in the current test source.	Sei Testnet contract address; manager, feeRecipient and GasPool addresses; SEI-native gas environment.
REQUIRED — Sei Mainnet	Production-security or deployment requirements not claimed as enforced by current bytecode.	Independent Mainnet deployment; production registry; multisig manager; cold-key separation; final production address set.
RECOMMENDED HARDENING	Items to evaluate before production.	Ownership migration handling; renounceOwnership policy; feeRecipient / restricted-recipient validation; cleanup of legacy identifiers and comments.

The exact Solidity source used for the Sei Testnet deployment is the technical implementation reference for this document. Its bytecode and ABI have been successfully matched by SeiScan. Deployment-specific comments or legacy internal identifiers, where present, do not alter the executed OEURO token identity or the documented clearing invariants.

- Rename ORUSD-specific constants, mappings and errors where practical (for example ORUSD_DECIMALS and orusdBlacklisted) to OEURO-neutral or OEURO-specific identifiers.
- Maintain deployment comments and GasPool documentation consistently with Sei; SEI is the native gas asset for the current Testnet reference.
- GasPool documentation for the Sei reference environment uses SEI as the native gas asset.
- Replace deployment-specific manager, feeRecipient and GasPool addresses for production and publish the final Mainnet values in the external registry.
- Verify all deployment-specific addresses satisfy the EOA restrictions required by the contract.
- Define ownership-transfer and ownership-renunciation procedures before production.
- Validate feeRecipient configuration against restricted-recipient rules.

PRE-MAINNET REVIEW QUESTION

Can the verified Sei Testnet OEURO reference implementation be independently deployed on Sei Mainnet while preserving the documented P2P, oracle, anti-replay and supply-control invariants and without introducing undocumented circulation or custody paths?

18. Auditor Acceptance Checklist

- No oracle can directly create or destroy supply.
- Only the manager can execute or reject oracle requests.
- Every executed oracle request was previously PENDING and unexpired.
- Every external proof pair is single-use.
- Manual mint and burn paths are fully enumerated and privilege-gated.
- Ordinary P2P transfer accounting conserves token units.
- approve() and transferFrom() cannot be used.
- Ordinary transfers to smart contracts revert.
- Restricted infrastructure addresses cannot receive ordinary OEURO.
- Fee-recipient behavior cannot create recursive fees or a transfer deadlock.
- Ownership transfer, renunciation and role migration have explicit production procedures.
- The OEURO token contract contains no customer-deposit, custody, lending or collateral-management function. Where an external EUR-denominated escrow layer is used, EURC issued by Circle is the recommended collateral asset, with custody and escrow controls maintained outside OEURO.
- The public Sei Testnet reference address and any future Sei Mainnet production address are clearly distinguished in documentation and registry records.
- Testnet-specific source comments and legacy internal identifiers are cleaned or explicitly documented before production.

FINAL SECURITY OBJECTIVE

OEURO should do one thing and do it well: provide direct peer-to-peer EUR-denominated clearing transfers between user-controlled accounts, with controlled supply, non-replayable oracle evidence and explicit manager settlement authority.

PREPARED BY — Denis Bouzon · ORCID 0009-0007-8894-8902 · Osnias Clearing · www.osnias-clearing.com

Document revision: Version 2.0 — dated 31 August 2026. OEURO Token technical white paper and associated documentation. © 2026 Osnias Clearing. All rights reserved. Drafted by Denis Bouzon (ORCID 0009-0007-8894-8902). Intended for public repository deposit and DOI assignment; DOI to be added after deposit.