USD DWIN

(USDW)

Technical White Paper

USD-Backed Omnichain Stablecoin on BNB Chain

Dwin Intertrade Company Limited

Version 1.0 | February 2026 | Confidential

BSC Mainnet | Chain ID: 56 | EIP-2535 Diamond Standard

Diamond Proxy: 0x081684720530e691edb6b3965dE181F44Fcfd8fB

Legal Disclaimer

This White Paper is issued by Dwin Intertrade Company Limited ('Dwin', 'the Company') for informational purposes only. Nothing in this document constitutes financial advice, an offer to sell, a solicitation of an offer to buy, or a recommendation for any security, commodity, or other financial instrument. The USDW token is a utility instrument designed to function as a USD-backed digital settlement medium.

This document is intended for sophisticated readers including institutional investors, compliance officers, technical auditors, and regulatory bodies. Recipients should conduct their own due diligence and seek independent legal and financial advice before making any decisions in connection with USDW.

All technical parameters, contract addresses, and operational procedures described herein were accurate as of the date of last verification (February 25, 2026) and are subject to change through the governance and upgrade mechanisms described in this document. The Company makes no representation that the information will remain accurate after publication.

USDW is not insured by any government deposit protection scheme. Token holders bear counterparty risk with respect to the reserve custodian and operational risks associated with smart contract technology.

Table of Contents

1. Executive Summary

USD Dwin (USDW) is a fully USD-backed stablecoin deployed on BNB Chain (BSC) Mainnet, engineered for institutional-grade performance, regulatory compliance, and seamless omnichain interoperability. One USDW token represents exactly one United States Dollar held in reserve by Dwin Intertrade Company Limited.

USDW is built upon the EIP-2535 Diamond Standard, the most advanced upgradeable smart contract architecture available on EVM-compatible blockchains. The Diamond proxy at address 0x081684720530e691edb6b3965dE181F44Fcfd8fB aggregates 66 verified facets covering the complete lifecycle of a regulated stablecoin, from KYC/AML onboarding to omnichain bridging via LayerZero.

Key Statistics — February 25, 2026

Metric	Value	Status
Total Facets Deployed	66 facets	✓ DONE
Function Selectors	283+ selectors	✓ DONE
BSCScan Verification	All 66 facets verified	✓ DONE
Peg	1 USDW = USD 1.00	✓ DONE
Network	BSC Mainnet (Chain ID: 56)	✓ DONE
Standard	EIP-2535 Diamond + ERC-20	✓ DONE
Cross-Chain	LayerZero OFT v2 (7 networks)	✓ DONE
Solidity Version	0.8.24 (EVM: Paris)	✓ DONE
Post-Deploy Tests	All 8 core scenarios passed	✓ DONE

USDW addresses three critical failures of existing stablecoin infrastructure: opacity of reserves, immutability of code in the face of evolving regulation, and fragmentation across blockchain networks. Through on-chain proof-of-reserve reporting, a timelock-governed upgrade pathway, and native LayerZero OFT integration, USDW sets a new standard for regulated digital dollar infrastructure in Southeast Asia and globally.

2. Introduction & Background

2.1 The Problem

Global cross-border payments remain inefficient, expensive, and exclusionary. SWIFT transactions carry fees of 1-3% and settle in 1-5 business days. Existing stablecoins either lack regulatory compliance infrastructure (USDT), suffer from opacity of reserves, or are limited to a single blockchain network, requiring multiple bridge hops and introducing additional counterparty risk.

For businesses operating in Southeast Asia, the problem is acute. Thailand, the Philippines, and Vietnam collectively process over USD 90 billion in annual remittances, yet the average fee remains above 6% according to World Bank data. Financial institutions serving these markets require a programmable digital dollar that carries institutional-grade compliance tooling.

2.2 The USDW Solution

Dwin Intertrade Company Limited was established to bridge traditional Thai financial infrastructure with decentralized blockchain protocols. USDW is the Company's primary product: a fully collateralized, programmable USD stablecoin designed to operate natively across multiple blockchain networks while maintaining the compliance requirements demanded by regulated financial institutions.

- 1:1 USD backing — every USDW in circulation corresponds to one USD held in the Company's designated reserve bank accounts
- Programmable compliance — on-chain KYC/AML, blacklisting, transaction limits, and regulatory reporting built directly into the token contract
- Omnichain interoperability — native LayerZero OFT v2 integration enables USDW transfer across BNB Chain, Ethereum, Polygon, Arbitrum, Optimism, Base, and Avalanche
- Institutional-grade upgradeability — EIP-2535 Diamond Standard allows facet-level code upgrades without redeployment, enabling the contract to evolve with regulatory requirements
- Proof of Reserve — on-chain reserve reporting, snapshot hashing, and multi-source oracle integration provide auditors with cryptographic evidence of collateralisation

2.3 Company Overview

Dwin Intertrade Company Limited is a Thai financial technology company focused on cross-border payment infrastructure, digital asset custody, and regulated stablecoin issuance. The Company maintains banking relationships with Thai registered financial institutions for the purpose of USD reserve custody.

3. Technical Architecture

3.1 EIP-2535 Diamond Standard

USDW is deployed using the EIP-2535 Diamond Standard, also known as the Multi-Facet Proxy pattern. This architecture resolves a fundamental tension in smart contract design: the need for immutability as a security guarantee versus the need for upgradeability as a regulatory requirement.

In the Diamond pattern, a single proxy contract (the Diamond) delegates function calls to one of many implementation contracts (Facets). Each facet is a separate, independently deployed and verified smart contract. The Diamond maintains a routing table mapping each 4-byte function selector to the appropriate facet address.

This architecture delivers four critical properties for a regulated stablecoin:

1. Modularity — individual components (e.g., KYC logic, bridge logic, pricing) can be upgraded independently without touching unrelated code
2. Size scalability — the Diamond is not subject to the 24 KB contract size limit that constrains monolithic designs, enabling the 283+ function selectors registered in USDW
3. Auditability — each facet is a small, independently verifiable contract, reducing the cognitive burden on security auditors
4. Governance compatibility — upgrade actions (diamondCut calls) can be subject to timelocks and multi-signature approvals, ensuring no single actor can unilaterally modify token behaviour

3.2 Contract Architecture Overview

Core Contract Addresses	
Diamond Proxy	`0x081684720530e691edb6b3965dE181F44Fcfd8fB`
Network	`BNB Chain Mainnet (Chain ID: 56)`
Admin Wallet	`0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6`
Storage Position	`keccak256('usdw.storage.dwinintertrade.2026.v10')`
Solidity	`0.8.24, EVM Paris, viaIR: true, optimizer: 200 runs`
Total Facets	`66 independently verified contracts`
Total Selectors	`283+ function selectors registered`
BSCScan	`All 66 facets source-verified`

3.3 Storage Architecture

USDW uses the Diamond Storage pattern in which all state variables are stored at a deterministic storage slot derived by hashing a unique namespace string. This prevents storage collisions between facets and ensures that each facet reads from and writes to its own isolated region of the Diamond's storage.

```
Storage Position = keccak256('usdw.storage.dwinintertrade.2026.v10')
```

The primary storage layout is as follows:

Field	Offset from BASE	Description
Name / Symbol / Decimals	BASE + 0–2	ERC-20 metadata
totalSupply	BASE + 3	Circulating supply counter
balances	BASE + 4	ERC-20 balance mapping
allowances	BASE + 5–7	ERC-20 approval mapping
roles	BASE + 8	RoleData.hasRole mapping
emergencyData	BASE + 9	EmergencyData.paused bool

3.4 Facet Categories

The 66 deployed facets are organised into eight functional categories:

Category	Facets	Key Functions
Core ERC-20 & Diamond	DiamondCutFacet, DiamondLoupeFacet, OwnershipFacet, CoreERC20Facet, PermitFacet	transfer, approve, diamondCut, facets()
Mint & Redemption	MintBurnFacetProd, RedemptionFacet, RedemptionVaultFacet	mint, burn, requestRedemption, processRedemption
Compliance & KYC	KYCAMLFacet, ComplianceFacet, ISOControlFacet, CatalogComplianceFacet, TransferGuardFacet	setKYC, setBlacklisted, setRedFlag, canTransfer
Oracle & Peg	PegPolicyFacet, PriceLogicFacet, OracleViewFacet, AIOracleFacet, PriceSourcesFacet	updatePrice, isPegHealthy, setFXRate
Proof of Reserve	PorFacet, PorPolicyFacet, PorSnapshotFacet, PoRMultiChainFacet, AuditBoardFacet	reportReserves, setPorSnapshotHash, recordEvidence
Cross-Chain (OFT)	OFTFacet, BridgeCompatFacet, BridgePluginFacet, CrossChainAdapterFacet	send, quoteSend, setPeer, lzReceive
DeFi & E-Wallet	DeFiExchangeFacet, DexRouterFacet, EWalletFacet, EWalletLedgerFacet, FeesFacet	swap, addLiquidity, recordFiatDeposit
Governance & Admin	GovernanceFacet, RolesFacet, EmergencyFacet, ConfigFacet, UpgradeTimelockFacet	pause, grantRole, setConfig, scheduleCut

4. Token Economics & Reserve Mechanism

4.1 Collateralisation Model

USDW is a fully collateralised stablecoin. The total supply of USDW in circulation at any point in time is backed 1:1 by United States Dollars held in segregated reserve accounts maintained by Dwin Intertrade Company Limited at licensed Thai financial institutions.

The collateralisation ratio is enforced through two mechanisms. First, operationally: the Company's minting policy requires confirmation of fiat receipt before any mint() call is executed. Second, technically: the PorPolicyFacet allows the configuration of a minimum reserve ratio, and can be set to disable minting automatically if the on-chain reported reserves fall below the required threshold.

4.2 Peg Stability Mechanism

The peg to USD 1.00 is maintained through the following mechanisms:

5. Collateral discipline — minting only occurs against confirmed fiat receipt; redemption destroys tokens upon fiat disbursement
6. Price policy — the PegPolicyFacet enforces a configurable price floor (default: USD 0.9995) and ceiling. Transactions outside the peg band are blocked
7. Oracle integration — the AIOracleFacet and PriceLogicFacet aggregate price data from multiple on-chain and off-chain sources to detect peg deviations in real time
8. Emergency controls — the EmergencyFacet enables instant system-wide pause if anomalous market conditions are detected, protecting against speculative attack

4.3 FX Rate Engine

USDW includes a multi-currency FX rate engine (FXRatesFacet, MultiCurrencyReserveFacet) that enables real-time quoting of USDW values in any configured fiat currency. At launch, the following currencies are supported: Thai Baht (THB), Philippine Peso (PHP), Japanese Yen (JPY), Euro (EUR), British Pound (GBP), Singapore Dollar (SGD), and US Dollar (USD).

FX rates are set by addresses holding the ORACLE_ROLE and are stored on-chain with timestamp attestation. The quoteUSDWToCurrencyE18() function provides atomic, deterministic conversion at the stored rate for settlement and reporting purposes.

4.4 Token Properties

USDW Token Parameters	
Token Name	USD Dwin
Token Symbol	USDW
Decimals	18 (on-chain) / 6 shared decimals for cross-chain (OFT standard)
Peg	USD 1.00 (1:1 collateralised)
Standard	ERC-20 + ERC-2612 Permit + LayerZero OFT v2

Total Supply	Dynamic — minted against fiat deposit, burned on redemption
Max Supply	Uncapped — constrained by reserve balance
Transfer	Standard ERC-20, subject to compliance guard

5. Mint & Redemption Flow

5.1 Fiat-In: Mint Flow

The minting process is initiated by a customer depositing USD into Dwin Intertrade's designated reserve bank account. The following sequence ensures that tokens are only created after fiat has been confirmed as received:

9. Customer initiates a fiat bank transfer (SWIFT, SEPA, or domestic wire) to the Company's reserve account, referencing a unique deposit reference
10. The Company's compliance system confirms receipt of funds and verifies that all KYC/AML checks pass for the depositing customer
11. The backend wallet (USDW.ROLE.ADMIN) calls mint(customerAddress, amount) on the Diamond Proxy
12. MintBurnFacetProd executes: erc20.balances[customer] += amount and erc20.totalSupply += amount
13. A Transfer(address(0), customer, amount) event is emitted, indexable by all standard ERC-20 tooling
14. The EWalletLedgerFacet records the mint event with the deposit reference and bank transaction hash for audit purposes
15. Customer receives USDW in their registered wallet

5.2 Fiat-Out: Redemption Flow

Redemption converts USDW back to fiat. The two-step process (request then process) ensures that tokens are not destroyed until fiat has been confirmed as dispatched:

16. Customer calls requestRedemption(amount) on the Diamond Proxy
17. RedemptionFacet atomically: balances[customer] -= amount and balances[Diamond] += amount — tokens are locked inside the Diamond, not yet burned
18. A RedemptionRequested event is emitted with a unique requestId
19. The Company's backend detects the event and initiates a fiat bank transfer to the customer's registered bank account
20. After the fiat transfer is confirmed as settled, the backend calls processRedemption(requestId)
21. RedemptionFacet burns the locked tokens: balances[Diamond] -= amount and totalSupply -= amount
22. A RedemptionProcessed event is emitted — the flow is complete and auditable end-to-end

5.3 Supply Conservation

The net effect of a complete mint-and-redeem cycle on total supply is zero. This property was verified on BSC Mainnet on February 25, 2026 through a full end-to-end test of the fiat flow. The on-chain test script (test-fiat-flow.js) is available in the Company's operational repository.

6. Omnichain Architecture (LayerZero OFT)

6.1 LayerZero Protocol

USDW implements the LayerZero Omnichain Fungible Token (OFT) v2 standard via the OFTFacet (0xf1D6Ae0aa8F5f311ac173dB6eED1FEB99e244524). LayerZero is a cross-chain messaging protocol that enables trustless, verifiable communication between blockchain networks through a configurable set of Decentralised Verifier Networks (DVNs) and executors.

The OFT standard enables USDW to move between chains atomically. When a user sends USDW from BSC to Ethereum, the tokens are burned on BSC and minted on Ethereum in a single LayerZero-guaranteed message. There is no wrapped token, no liquidity pool, and no bridge custodian — the token supply on each chain is always the canonical supply.

6.2 Supported Networks

Network	LayerZero EID (v2)	Chain ID	Status
BNB Chain (source)	30102	56	✓ DONE
Ethereum	30101	1	✓ DONE
Polygon	30109	137	✓ DONE
Arbitrum One	30110	42161	✓ DONE
Optimism	30111	10	✓ DONE
Base	30184	8453	✓ DONE
Avalanche C-Chain	30106	43114	✓ DONE

6.3 OFT Key Functions

The OFTFacet exposes the following key interfaces:

- quoteSend(SendParam, bool) — returns the nativeFee and lzTokenFee required to execute a cross-chain transfer before committing to a transaction
- send(SendParam, MessagingFee, address) — initiates a cross-chain transfer; burns tokens on the source chain and triggers a LayerZero message to the destination
- setPeer(uint32 eid, bytes32 peer) — configures the trusted Diamond address on the destination chain; only configured peers can receive USDW
- peers(uint32 eid) — returns the configured peer address for a given endpoint ID
- sharedDecimals() — returns 6, the precision used for cross-chain amounts, preventing dust accumulation
- approvalRequired() — returns false; the Diamond handles token custody directly, eliminating one transaction from the user flow
- lzReceive(Origin, bytes32, bytes, address, bytes) — called by the LayerZero endpoint on the destination chain to credit received USDW

6.4 Security Model

Cross-chain security is enforced through trusted peer configuration. Only addresses registered via setPeer() can send or receive USDW cross-chain. This prevents spoofed contracts on other chains from minting USDW. Additionally, setMinDstGas() ensures that the destination chain executor has sufficient gas to complete lzReceive(), preventing griefing attacks that could leave messages in a pending state.

7. Security Architecture

7.1 Role-Based Access Control

All privileged operations in USDW are protected by a hierarchical role-based access control system implemented in RolesFacet and RoleAdminActionsFacet. Roles are defined as keccak256 hashes and stored in the Diamond's role storage.

Role	Hash	Permitted Operations
DEFAULT_ADMIN_ROLE	0x000...000 (zero hash)	Grant/revoke any role; execute diamondCut upgrades
USDW.ROLE.ADMIN	keccak256('USDW.ROLE.ADMIN')	mint(), burn(), processRedemption()
ORACLE_ROLE	keccak256('ORACLE_ROLE')	updatePrice(), setFXRate(), reportReserves()
AUDIT_ROLE	keccak256('AUDIT_ROLE')	recordEvidence(), setPorSnapshotHash()
RAMP_ADMIN_ROLE	keccak256('RAMP_ADMIN_ROLE')	FiatOnOffRamp administration
PLUGIN_ADMIN_ROLE	keccak256('PLUGIN_ADMIN_ROLE')	DeFi plugin and adapter management

7.2 Emergency Controls

The EmergencyFacet provides a multi-tier pause mechanism. System-wide pause (pause()) immediately halts all mint, burn, and transfer operations across the entire Diamond. Function-level pause (setFunctionPaused()) allows specific function selectors to be individually suspended without affecting other operations, enabling surgical responses to localised risks.

The emergency stop (emergencyStop()) is a one-way, irrevocable action that permanently halts the system. It is designed as a last-resort measure in the event of a critical vulnerability, and requires DEFAULT_ADMIN_ROLE to execute.

7.3 Upgrade Security

Diamond upgrades (diamondCut calls) are gated by the UpgradeTimelockFacet. Once a cut is scheduled, a configurable delay period must elapse before execution is permitted. This window allows token holders, auditors, and the community to review proposed changes and respond if necessary.

The DiamondCutFacet enforces that: (a) only addresses with DEFAULT_ADMIN_ROLE may call diamondCut; (b) the Replace action is used for existing selectors and the Add action for new ones, preventing accidental selector collisions; (c) the Remove action (used to remove the debug facet 0xB01f...) permanently deregisters selectors.

7.4 Transfer Guard

Every ERC-20 transfer is routed through the TransferGuardFacet, which consults ComplianceFacet and KYCAMLFacet before allowing the transfer. Blacklisted addresses cannot send or receive. Addresses with active red flags (SAR filings) are subject to additional restrictions. The canTransfer(from, to, amount) view function provides an off-chain simulation interface for front-end applications.

7.5 Permit (ERC-2612)

The PermitFacet implements EIP-2612 gasless approvals, enabling users to sign an off-chain authorisation that a third party can submit on-chain. This eliminates the two-transaction flow (approve + transferFrom) for DeFi integrations and improves user experience without compromising security.

8. Compliance & Regulatory Framework

8.1 KYC/AML Framework

USDW implements a multi-layered Know Your Customer and Anti-Money Laundering framework directly within the smart contract. The KYCAMLFacet stores the compliance status of each address on-chain, ensuring that all token interactions are subject to compliance verification without requiring off-chain gating.

- KYC Status — isKYCApproved(address): bool returns whether an address has completed identity verification
- Risk Levels — setRiskLevel(address, uint8): addresses are assigned risk tiers (1 = Low, 2 = Medium, 3 = High) which may trigger enhanced due diligence requirements
- Blacklisting — setBlacklisted(address, bool): sanctioned addresses are prevented from sending or receiving USDW
- Whitelisting — setWhitelisted(address, bool): institutional counterparties may be whitelisted to bypass certain restrictions
- Red Flags — setRedFlag(address, bool, bytes32): SAR (Suspicious Activity Report) flag with reference hash, permanently recorded on-chain

8.2 ISO 20022 & ISOControlFacet

USDW includes an ISOControlFacet with 17 registered function selectors that map to ISO 20022 message types used in global interbank communication. This enables USDW to participate in payment flows that interface with SWIFT gpi and other ISO 20022-compliant infrastructure, a critical requirement for institutional adoption in Thailand and internationally.

8.3 Proof of Reserve

The Proof of Reserve (PoR) subsystem consists of four facets: PorFacet, PorPolicyFacet, PorSnapshotFacet, and PoRMultiChainFacet. Together, they enable the Company to publish cryptographically verifiable attestations of USD reserve holdings on-chain.

23. reportReserves(amount, proofHash) — records the total USD reserve amount and a hash of the supporting bank statement or custody document
24. setPorSnapshotHash(bytes32) — anchors a Merkle root or document hash of all reserve account balances on a given date
25. recordEvidence(evidenceHash, auditRef, bytes) — records third-party audit evidence (e.g., KPMG attestation) on-chain with reference to the external document
26. getReserves(), getLastProof(), getLastReportTime() — view functions enabling any party to verify the current reserve status at any time

8.4 Audit Trail

The AuditBoardFacet and AuthenticityGuardFacet provide immutable on-chain audit trails. All material events — mints, burns, redemptions, role changes, reserve reports, and emergency actions — are permanently recorded as Ethereum events on BSC. The EWalletLedgerFacet additionally records fiat deposit and withdrawal events with bank reference hashes, enabling reconciliation between on-chain and off-chain records.

9. Governance

9.1 Governance Model

USDW governance operates through two complementary systems: administrative role-based governance for routine operational decisions, and on-chain vote-based governance for material protocol changes. The GovernanceFacet implements ERC-20 vote delegation, enabling token holders to participate in governance proposals proportional to their USDW holdings.

9.2 Diamond Upgrade Governance

Any upgrade to the Diamond (adding, replacing, or removing facets) requires: (a) the calling address to hold DEFAULT_ADMIN_ROLE; (b) the upgrade to be scheduled through the UpgradeTimelockFacet with a mandatory delay period; (c) the upgrade to be reviewed and, in a future phase, ratified by the governance system via GovernorAdapterFacet.

This multi-layer upgrade governance ensures that no single individual can unilaterally modify the protocol, and that all changes are subject to a minimum review window accessible to all stakeholders.

9.3 Multisig Transition (Pending)

As noted in the post-deployment checklist, the DEFAULT_ADMIN_ROLE is currently held by the deployer EOA (0xc557ee...). The Company's operational plan includes transferring this role to a Gnosis Safe multi-signature wallet requiring a threshold of approvals from independent signers. This transition is pending and represents the final step in the decentralisation of administrative authority over USDW.

9.4 FeatureFlagsFacet

The FeatureFlagsFacet provides a governance-gated mechanism for enabling or disabling specific features of the protocol without executing a full diamondCut. Feature flags can be used to: enable/disable gasless transactions, activate/deactivate DeFi integrations, or toggle cross-chain bridging on a per-network basis. Each flag change is recorded as an on-chain event.

10. Facet Registry & Contract Addresses

10.1 Core Infrastructure Facets

Facet	Address	Selectors
DiamondCutFacet	0x50B183E3275f5B81dFEd887b82510E413E9BdDb7	1
DiamondLoupeFacet	0x387382898878f9E9324F84E5001E634f34ECa6d3	4
OwnershipFacet	0xF557B053B28190d5858669Ea0b4518D8C8712Af0	3
CoreERC20Facet	0x9AA7DdDD5DbA8E3491f58221E82AbEc9647637c8	9
MintBurnFacetProd	0x27e32950d30Ba3c11712ecf2d821bbC605bF29C8	2
RedemptionFacet	0x60fC052b27483889FFaCe1ad8d68690189943b53	3
EmergencyFacet	0x31380d857a0Dcca3d2093EC26c58d2A4aA555525	8
RolesFacet	0xc61402a5A3075bd424A3AAd26753f531300c0e03	4
PermitFacet	0x07e9C591dB9C63bBc31E130C62806190d90A9f50	3

10.2 Cross-Chain & Bridge Facets

Facet	Address	Selectors
OFTFacet	0xf1D6Ae0aa8F5f311ac173dB6eED1FEB99e244524	10
BridgeCompatFacet	0xb1537e5ea31bfa86d5DafA0bE13f37D3Cfb40555	2
BridgePluginFacet	0x647d6D7bfc0cBC17B24E4b04ebbF26Cc3564AdDf	4
RampBridgePluginFacet	0x8A443cd46B55F4e48818376e3ec238055Bc00618	5
CrossChainAdapterFacet	0xcbC7346060EbF59Ddd6E48365Fa8936F55d50A81	2
CrossChainRegistryFacet	0x22456cb02C07b6bcDAC94F4D8D20D5BD4e4d716e	2

The complete 66-facet registry is provided in Appendix A. All facet source code has been verified on BSCScan and is publicly accessible for audit review.

11. Audit & Verification Status

11.1 On-Chain Verification

All 66 USDW facets have been submitted to and verified by BSCScan using the Etherscan V2 API with the exact compiler configuration (Solidity 0.8.24, EVM Paris, viaIR: true, optimizer 200 runs) used for deployment. Source code for every facet is publicly accessible at bscscan.com.

Verification Item	Status	Date
All 66 facets source-verified on BSCScan	✓ DONE	Feb 25, 2026
Mint function test (totalSupply++)	✓ DONE	Feb 25, 2026
Burn function test (totalSupply--)	✓ DONE	Feb 25, 2026
Full fiat flow E2E (net supply change = 0)	✓ DONE	Feb 25, 2026
Pause test (mint reverts with PAUSED)	✓ DONE	Feb 25, 2026
Unpause test (mint succeeds again)	✓ DONE	Feb 25, 2026
Role access control (USDW_ADMIN only)	✓ DONE	Feb 25, 2026
RPC stability (bsc.publicnode.com)	✓ DONE	Feb 25, 2026
Remove debug facet 0xB01f...	⚠ VERIFY	Pending
Transfer DEFAULT_ADMIN to Gnosis Safe	⚠ VERIFY	Pending

11.2 Third-Party Audit

The Company intends to commission a comprehensive security audit by a recognised blockchain security firm prior to material growth in token supply. The modular Diamond architecture facilitates focused audits of individual facet categories rather than requiring a single monolithic review. Audit reports will be published in full on the Company's website.

11.3 Ongoing Monitoring

Post-deployment, the Company maintains a suite of operational monitoring scripts (detailed in the Deployment Checklist) that enable continuous verification of system integrity: check-roles.js for access control auditing, test-pause.js for emergency control verification, final-verify.js for mint/burn sanity checks, and verify-all-facets.js for the Diamond's selector registry.

12. Risk Factors

Prospective users and investors should carefully consider the following risk factors before interacting with USDW:

12.1 Smart Contract Risk

Despite the architectural advantages of the Diamond pattern and planned third-party audits, smart contracts may contain undiscovered vulnerabilities. An exploit could result in loss of tokens or permanent disruption of system operation. The Company has implemented multi-tier emergency controls and a timelocked upgrade pathway to mitigate this risk.

12.2 Custodial Reserve Risk

USD reserves are held in bank accounts controlled by Dwin Intertrade Company Limited. Token holders bear custodial risk with respect to these reserves. The Company does not hold reserves through an independent custodian, and bank deposits are not insured beyond applicable deposit protection limits. The Company mitigates this through on-chain proof-of-reserve attestation and planned integration of multi-bank diversification.

12.3 Regulatory Risk

The regulatory treatment of stablecoins is evolving in all jurisdictions. Future regulatory requirements in Thailand, the EU, the United States, or other relevant jurisdictions could require material changes to the token's design, require licensing, or restrict its use. The Diamond architecture provides upgrade flexibility, but regulatory risk cannot be fully mitigated by technical means.

12.4 Operational Key Risk

The DEPLOYER_PK (private key of 0xc557ee...) currently controls all administrative roles. Loss or compromise of this key would result in loss of administrative control over the protocol. The Company is in the process of migrating DEFAULT_ADMIN_ROLE to a Gnosis Safe multi-signature wallet to distribute key-person risk.

12.5 LayerZero Protocol Risk

Cross-chain transfers depend on the continued operation and security of the LayerZero protocol. A vulnerability in LayerZero's DVN network or endpoint contracts could affect USDW cross-chain transfers. The Company monitors LayerZero security disclosures and will adjust peer configurations as necessary.

13. Roadmap

Phase	Timeline	Milestone	Status
Phase 1 — Foundation	Q1 2026	Diamond deployment, 66 facets verified, core fiat flow tested	✓ DONE
Phase 1 — Foundation	Q1 2026	LayerZero OFT integration across 7 networks	✓ DONE
Phase 2 — Security	Q1 2026	Remove debug facet; migrate DEFAULT_ADMIN to Gnosis Safe	⚠ VERIFY
Phase 2 — Security	Q2 2026	Commission and publish third-party smart contract security audit	Planned
Phase 3 — Compliance	Q2 2026	Formal KYC/AML process integration; ISO 20022 bank connectivity pilot	Planned
Phase 3 — Compliance	Q2 2026	Quarterly proof-of-reserve attestation by independent accountant	Planned
Phase 4 — Growth	Q3 2026	DeFi protocol integrations (Pancakeswap, Aave, Compound) on BSC and Arbitrum	Planned
Phase 4 — Growth	Q3 2026	E-wallet partner onboarding; Thai fiat on-ramp via QR code payment	Planned
Phase 5 — Governance	Q4 2026	Transfer governance to community DAO via GovernanceFacet	Planned
Phase 5 — Governance	Q4 2026	Launch USDW developer SDK and open APIs for partner integrations	Planned

Appendix A: Full Facet Registry

The following table lists all 66 facets deployed to the USDW Diamond on BSC Mainnet. All entries are verified on BSCScan.

#	Facet Name	Address	Selectors
1	**DiamondCutFacet**	0x50B183E3275f5B81dFEd887b82510E413E9BdDb7	1
2	**DiamondLoupeFacet**	0x387382898878f9E9324F84E5001E634f34ECa6d3	4
3	**OwnershipFacet**	0xF557B053B28190d5858669Ea0b4518D8C8712Af0	3
4	**CoreERC20Facet**	0x9AA7DdDD5DbA8E3491f58221E82AbEc9647637c8	9
5	**MintBurnFacetProd**	0x27e32950d30Ba3c11712ecf2d821bbC605bF29C8	2
6	**RedemptionFacet**	0x60fC052b27483889FFaCe1ad8d68690189943b53	3
7	**EmergencyFacet**	0x31380d857a0Dcca3d2093EC26c58d2A4aA555525	8
8	**RolesFacet**	0xc61402a5A3075bd424A3AAd26753f531300c0e03	4
9	**RoleAdminActionsFacet**	0xb3DA64FA7b043BC17D838ECD18177fa6B7d822B7	2
10	**PermitFacet**	0x07e9C591dB9C63bBc31E130C62806190d90A9f50	3
11	**TransferGuardFacet**	0x64015AA7fCB2Fec0f9Ac09F3412de4B466e9F29B	1
12	**LockingFacet**	0x0629Aee58aC6AD0C2938c3EC3CF10D40d3394709	4
13	**VestingPluginFacet**	0x7ce903fddbAB694740dEAAD72783843482a5C6f8	4
14	**ConfigFacet**	0xcB7F2dBC1C561345EE13Ea295D609eFCa6AA1008	4
15	**FeatureFlagsFacet**	0x1a004a75d1d1bfD8D36CD880BBB1aa5902daC21C	2
16	**AdminUtilsFacet**	0x7B7a041A65502882DD869bC55EBd719F0cde50De	2
17	**GovernanceFacet**	0xe4db377849a9D308098A8ba2BA9cBCa21a19a979	4
18	**GovernorAdapterFacet**	0x8614f23827e96Bc03087DbB550e560f31a8c5304	7
19	**UpgradeTimelockFacet**	0x0c6C8238b7E5a8C321660392A64C01fE03A97201	2
20	**UpgradeabilityCompatFacet**	0x83C1ed74676f7c839E0EDAd6DC58869157D55a14	3
21	**PegPolicyFacet**	0xa9e82De588b0F8CAAd65f1746577C968A24260C3	6
22	**PegReferenceFacet**	0x3475356713CE43a59Af61DDDb01DBE0C56711295	2
23	**PriceLogicFacet**	0x3cFEbE48e2414F26f65dF2a8fd63505C4be42e81	7
24	**PriceSourcesFacet**	0xd7ea2eA1FBdFE8bB665234Af73D709E038C3015c	1
25	**OracleViewFacet**	0x3F8BFa7bE1214C6e0C0A5181C843bA37E5888f5F	3
26	**AIOracleFacet**	0x673fAdAEf85aC970a405A47B631a2F5818CdBDED	1
27	**PorFacet**	0x6Ed7D49397Fd9525AE3c3d0B6642Ed700D311508	4
28	**PorPolicyFacet**	0xFcB9Ab29B6961d39b578F1128a08032FbD1277bc	4
29	**PorSnapshotFacet**	0xe362Cd90237A8274BCdE006B7d5D20874461eb2D	3
30	**PoRMultiChainFacet**	0x63Fe69bdF0965a4610943525Db70b935137c5686	6
31	**ISOControlFacet**	0x945A74ffEc4B9699C96C3eE48B9A214C912E7AA6	17
32	**ComplianceFacet**	0x7829bDF5Bc82d45a4Bbdf7c8a712482A48740E4a	8
33	**KYCAMLFacet**	0x34cB7D7914dee71E79381517Ade5431fdE31A899	4
34	**CatalogComplianceFacet**	0xa4186C75Aa301d2d6E4962F0c20cEA438A74424A	5
35	**AuditBoardFacet**	0xCc17382587F0aCDFFBf6093C4975e5489fCA7Bd7	2
36	**AuthenticityGuardFacet**	0xf01866aa0fc114C4f328dAaa470b4dAE620247D2	2

#	Facet Name	Address	Selectors
37	**FiatOnOffRampFacet**	0xb855804CecD362fEaF9D455252d4d83846ae6558	3
38	**EWalletFiatCryptoGatewayFacet**	0xed4927bAE08F50069154DA3Fe9A2Ec967De4D930	2
39	**EWalletFiatLedgerFacet**	0xAff21c9C3826a4391b9f1C7DDede3dBfBf872B52	2
40	**EWalletLedgerFacet**	0x01318A6F18569Cc8DF554218717867711a61895a	14
41	**EWalletFacet**	0x1948Cf7c737070AeBc44053C2F6DC8682c62D8f0	8
42	**GatewayRegistryFacet**	0xa83eC282FD108200849092eaaC0861cF3Cb9AacC	2
43	**AdapterRegistryFacet**	0xa77c6c489c894B0D58572d9f0A6C347FACe9Cb4a	4
44	**FeesFacet**	0xA63BcDf89F98e82e46e10186f2f2edA4E34D7353	12
45	**TreasuryWalletFacet**	0x58792165ea1C3eea53c034E2CC2cD074Be2D5E00	3
46	**FXRatesFacet**	0x4dAC24Ae3D81c0FF1cF6B68f61223C188c95b947	4
47	**MultiCurrencyReserveFacet**	0xB4DE4d738c8b2AA93d95f67275e7e9B15D60B4D7	3
48	**RedemptionVaultFacet**	0xbE50499aF1d6D4864c06fc3CaD22dFd120404ADa	10
49	**VoucherFacet**	0xdA2e4242A078cA7Db95A464B8Da85DFBbE1B64D8	5
50	**GaslessFacet**	0xDC1524C6058314897A3796f4f132043A4778085A	14
51	**DeFiExchangeFacet**	0xaeD169381845F359Ff77cbe09b082A8c3d85cfdD	8
52	**DexRouterFacet**	0xDEf35c6A307D48AB267d5409daC18739ACA0D638	6
53	**DexSafeguardFacet**	0x4a04f6a328d9A3D3F5Cf973222b2259c18189b79	4
54	**DexAdapterPluginFacet**	0x7B5E3903a9a464d8DB787ce06dB2E162C255a305	7
55	**OFTFacet**	0xf1D6Ae0aa8F5f311ac173dB6eED1FEB99e244524	10
56	**BridgeCompatFacet**	0xb1537e5ea31bfa86d5DafA0bE13f37D3Cfb40555	2
57	**BridgePluginFacet**	0x647d6D7bfc0cBC17B24E4b04ebbF26Cc3564AdDf	4
58	**RampBridgePluginFacet**	0x8A443cd46B55F4e48818376e3ec238055Bc00618	5
59	**CrossChainAdapterFacet**	0xcbC7346060EbF59Ddd6E48365Fa8936F55d50A81	2
60	**CrossChainRegistryFacet**	0x22456cb02C07b6bcDAC94F4D8D20D5BD4e4d716e	2
61	**CLVFacet**	0xebd0B74F2Da9ec9bB4C649C72f24112E079fC869	11
62	**IntentFacet**	0xe4E9F10F20E0d11cBE198DEea485285714211c3A	11
63	**VirtualAMMFacet**	0x1AfBd233F0C6c25E16A980560633AE89F98898b0	5
64	**RebaseElasticFacet**	0x7B263A004f9b65412D3447E8F56Ee6505facA0d4	3
65	**RebasePluginFacet**	0x757C3E3335433849F01285B3f3f5b8A527E12223	4
66	**StakingPluginFacet**	0x6Ad32B4B6E102082Df7b0EC0c1a366A7bB9fbb65	6

Total: 66 facets | 283+ selectors | All source-verified on BSCScan | © 2026 Dwin Intertrade Company Limited

Appendix B: Storage Layout

USDW uses Diamond Storage (EIP-2535 App Storage pattern) to ensure complete isolation between facet state variables. All state is stored under a single root slot derived deterministically from the protocol namespace.

```
bytes32 constant DIAMOND_STORAGE_POSITION =
    keccak256('usdw.storage.dwinintertrade.2026.v10');
```

Field	Offset	Type	Description
name	BASE + 0	string	Token name: 'USD Dwin'
symbol	BASE + 1	string	Token symbol: 'USDW'
decimals	BASE + 2	uint8	18
totalSupply	BASE + 3	uint256	Circulating supply
balances	BASE + 4	mapping(addr=>uint)	ERC-20 balances
allowances	BASE + 5	mapping nested	ERC-20 approvals
nonces	BASE + 6	mapping(addr=>uint)	EIP-2612 permit nonces
domainSeparator	BASE + 7	bytes32	EIP-712 domain separator
roles	BASE + 8	mapping(hash=>data)	RoleData.hasRole mapping
emergencyData	BASE + 9	struct	EmergencyData.paused bool + timestamps