

USDW2028 Architecture Specification

Page | 1

BSC • ISO-Ready • PoR/CPA • DEX SafeGuard • No Red Flags

Regulated Core + Extension Zone Blueprint

Cover Page

Document Title:

USDW2028 Architecture Specification

Version: 1.1

Prepared by: DWIN INTERTRADE COMPANY LIMITED

Author: Miss Suttatan Rungsichol

Date: February 2026

Document Purpose

This document defines the complete architecture of USDW2028, including regulated core modules, extension plugins, compliance boundaries, and operational flows. It includes the **Universal On/Off-Ramp Bridge Plugin**, enabling integration with any external platform (CEX, PSP, bank gateway, cross-chain bridge) without modifying the regulated core.

Regulatory Alignment

USDW2028 is architected to align with:

- ISO 27001 — Information Security
- ISO 22301 — Business Continuity
- ISO 37301 — Compliance Management
- ISO 20022 — Financial Messaging
- CPA Attestation Standards

- DEX SafeGuard Requirements
- AuditBoard Evidence Lifecycle

Confidentiality Notice

Page | 2

This document contains proprietary architectural information belonging to **DWIN INTERTRADE COMPANY LIMITED**. Unauthorized distribution is prohibited.

Table of Contents

Executive Summary

1. System Overview
 - 1.1 Phase Structure
 - 1.2 Regulated Core vs Extension Zone
2. Phase 1 — Core Minimal Token Path
 - 2.1 USDWStorageLayout
 - 2.2 USDWERC20Core
 - 2.3 USDWRoles
 - 2.4 USDWComplianceBase
3. Phase 2 — Pricing Layer
 - 3.1 IExternalPriceFeed
 - 3.2 USDWPriceSources
 - 3.3 USDWPriceLogic
 - 3.4 USDWOracleView
4. Phase 3 — PoR & Transfer Safety
 - 4.1 USDWPoROracle
 - 4.2 USDWPoRSnapshot
 - 4.3 USDWPoRPolicy
 - 4.4 USDWTransferGuard
5. Phase 4 — Operations & Redemption
 - 5.1 USDWMinting

- 5.2 USDWBurning
- 5.3 USDWRedemptionFlow
- 6. Phase 5 — Settlement Wallet
 - 6.1 USDWTreasuryConfig
 - 6.2 USDWFeeCollector
- 7. Phase 6 — Upgrade Safety
 - 7.1 USDWUpgradeSchedule
 - 7.2 USDWUpgradeExecute
- 8. Phase 7 — Plugin System (Non-Core)
 - 8.1 USDWDexAdapterPlugin
 - 8.2 USDWStakingPlugin
 - 8.3 USDWVestingPlugin
 - 8.4 USDWBridgePlugin
 - 8.5 USDWRampBridgePlugin (Universal On/Off-Ramp)
 - 8.6 USDWRebasePlugin
- 9. Phase 8 — Governance Utilities
 - 9.1 USDWAdminUtils
 - 9.2 USDWRoleAdminActions
- 10. Phase 9 — Final Assembly
 - 10.1 USDWFinal Contract
- 11. Architecture Flow Examples
 - 11.1 DEX Swap Flow
 - 11.2 Staking Flow
 - 11.3 Bridge Flow
 - 11.4 Universal On-Ramp Flow
 - 11.5 Universal Off-Ramp Flow

Appendices

- A. Role Definitions

- B. Event Logging Standards
 - C. AuditBoard Evidence Mapping
 - D. ISO Alignment Matrix
 - E. Glossary of Terms
-

Page | 4

Executive Summary

Your base specification states:

“This is the canonical build order... Nothing is missing. Nothing is duplicated.”

“Phases 1–6 + 8–9 = regulated core. Phase 7 = extension zone.”

USDW2028 is a fiat-backed stablecoin architecture designed for:

- Institutional acceptance
- Multi-chain expansion
- Long-term regulatory compliance
- AuditBoard-ready evidence lifecycle
- CPA attestation and PoR integrity

All regulated logic lives in the **core** (Phases 1–6, 8–9).

All non-core logic lives in **Phase 7 plugins**, including the new **Universal On/Off-Ramp Bridge Plugin**.

1. System Overview

USDW2028 is structured into nine phases:

Phase	Category	Regulated	Description
1	Core Minimal Token Path	Yes	ERC-20, storage, roles
2	Pricing Layer	Yes	Price sources, validation
3	PoR & Transfer Safety	Yes	CPA attestation, reserve logic
4	Operations & Redemption	Yes	Minting, burning, redemption
5	Settlement Wallet	Yes	Treasury configuration
6	Upgrade Safety	Yes	Timelock, upgrade authorization
7	Plugin System	No	AMM, staking, vesting, bridge, universal ramp, rebase
8	Governance Utilities	Yes	Admin utilities, evidence
9	Final Assembly	Yes	Minimal final contract

Page | 5

2. Phase 1 — Core Minimal Token Path

2.1 USDWStorageLayout

Defines all core storage variables in fixed order.

2.2 USDWERC20Core

Minimal ERC-20 logic with compliance hooks.

2.3 USDWRoles

Defines all regulated roles.

2.4 USDWComplianceBase

Pause, blacklist, region policy.

3. Phase 2 — Pricing Layer

Page | 6

3.1 IExternalPriceFeed

Chainlink-style interface.

3.2 USDWPriceSources

Collects raw prices.

3.3 USDWPriceLogic

Computes validated median price.

3.4 USDWOracleView

Exposes final validated price.

4. Phase 3 — PoR & Transfer Safety

4.1 USDWPoROracle

Stores reserve amount.

4.2 USDWPoRSnapshot

Stores CPA attestation hash.

4.3 USDWPoRPolicy

Computes reserve ratio.

4.4 USDWTransferGuard

Enforces pause, blacklist, region, PoR, maxTxAmount.

5. Phase 4 — Operations & Redemption

5.1 USDWMinting

Role-gated minting.

Page | 7

5.2 USDWBurning

Role-gated burning.

5.3 USDWRedemptionFlow

Defines fiat redemption lifecycle.

6. Phase 5 — Settlement Wallet

6.1 USDWTreasuryConfig

Treasury and settlement address.

6.2 USDWFeeCollector

Optional fee module.

7. Phase 6 — Upgrade Safety

7.1 USDWUpgradeSchedule

Timelocked upgrade scheduling.

7.2 USDWUpgradeExecute

Upgrade execution with authorization.

8. Phase 7 — Plugin System (Non-Core)

8.1 USDWDexAdapterPlugin

AMM interaction, swapAndLiquify, LP whitelist.

8.2 USDWStakingPlugin

Stake, unstake, claim rewards.

8.3 USDWVestingPlugin

Page | 8

Lock and release tokens.

8.4 USDWBridgePlugin

Chain-to-chain mint/burn.

8.5 USDWRampBridgePlugin — Universal On/Off-Ramp

Purpose

A universal, platform-agnostic gateway enabling:

- CEX on-ramp
- PSP on-ramp
- Bank gateway on-ramp
- Cross-chain on-ramp
- Off-ramp to any external platform

All without modifying the regulated core.

8.5.1 State

- processedMessages — replay protection
 - approvedBridges — allowed signers
 - approvedRampPartners — allowed platforms
 - rampConfigAdmin — admin for partner/bridge config
-

8.5.2 Canonical Message Format

```
struct RampMessage {  
    bytes32 partnerId;  
    address user;  
    uint256 amount;  
    uint256 nonce;  
    uint256 timestamp;  
    bytes32 offchainRef;  
}
```

8.5.3 On-Ramp Flow (External → USDW)

Function:

mintFromRamp(RampMessage msg_, bytes signature)

Checks:

- Partner approved
- Bridge signature valid
- Timestamp fresh
- Message not processed

Effects:

- Mark message processed
- Mint USDW to user

Event:

- RampOnMinted(...)

8.5.4 Off-Ramp Flow (USDW → External)

Function:

requestOffRamp(partnerId, amount, offchainRef)

Page | 10

Checks:

- Partner approved
- TransferGuard compliance
- Balance sufficient

Effects:

- Burn or lock USDW

Event:

- RampOffRequested(...)

Optional confirmation:

confirmOffRamp(...)

Event:

- RampOffCompleted(...)

8.5.5 Partner & Bridge Management

- setBridge
- setRampPartner
- setRampConfigAdmin

All role-gated and event-logged.

8.6 USDWRebasePlugin

Synthetic rebasing without touching core supply.

9. Phase 8 — Governance Utilities

9.1 USDWAdminUtils

Rescue tokens, rescue native, set admin, set config.

9.2 USDWRoleAdminActions

Role change audit trail.

10. Phase 9 — Final Assembly

10.1 USDWFinal

Inherits all required modules.

Contains no new logic.

Contains no new storage.

11. Architecture Flow Examples

11.1 DEX Swap Flow

(As previously defined)

11.2 Staking Flow

(As previously defined)

11.3 Bridge Flow

(As previously defined)

11.4 Universal On-Ramp Flow

1. User deposits fiat/asset on platform X
 2. Platform X performs KYC/AML
 3. Platform X constructs & signs RampMessage
 4. Calls mintFromRamp
 5. Plugin validates and mints
 6. Event emitted for AuditBoard
-

Page | 12

11.5 Universal Off-Ramp Flow

1. User calls requestOffRamp
 2. TransferGuard enforces compliance
 3. Plugin burns or locks USDW
 4. Event emitted
 5. Platform X executes payout
 6. Optional confirmOffRamp
 7. Completion event emitted
-

Appendices

Appendix A — Role Definitions

(ADMIN_ROLE, TREASURY_ROLE, ORACLE_ROLE, etc.)

Appendix B — Event Logging Standards

(AuditBoard-ready)

Appendix C — AuditBoard Evidence Mapping

(Every action mapped to evidence)

Appendix D — ISO Alignment Matrix

(ISO 27001, 22301, 37301, 20022)

Page | 13

Appendix E — Glossary of Terms

(DEX, TWAP, PoR, CPA, etc.)