CERTIK

xReserve Audit Executive Summary

Jan 13, 2026

Introduction

xReserve connects the Canton Network with Ethereum and enables users to move value between the two networks by converting native USDC (on Ethereum) into USDCx (on Canton) and back again. xReserve combines an on-chain smart contract with an off-chain infrastructure to support deposit and withdrawal workflows without relying on traditional third-party wrapped-token bridges.

Assessment Overview

CertiK conducted a security assessment focused on the DAML smart contracts that implement the xReserve functionality on Canton Network. The review centered on workflows that coordinate attestation when deposit/withdrawal, authorization controls, and USDCx lifecycle management.

In-Scope Components (DAML)

The DAML smart contract scope consists of two areas:

- xReserve Component**:** Implements the end-to-end on-ledger xReserve workflow, including deposit/withdrawal attestations, xReserve and user agreements, nonce handling and replay protection, the xReserve service and operator/representative configuration, USDCx instrument setup, and lifecycle actions such as creation/archival of transfer rules and the allocation factory.
- Network Utilities**:** Provides foundational templates and models for roles/credentials and token behaviors (mint, burn, transfer), plus auxiliary services such as token locking/unlocking and multi-purpose workflows (e.g., delivery-versus-payment allocations).

Trust Assumptions

For the purpose of the DAML smart contract audit, off-chain and external dependencies are treated as black-box and are assumed to be functionally correct and securely operated. These include (but not limited to) the Circle xReserve API / attestation services, xReserve backend services, databases, Ledger API connectivity, and key management services.

Timeline

Preliminary comments published on 12/12/2025

Final report published on 12/22/2025

Audit Summary

The assessment objective was to evaluate whether the in-scope DAML smart contracts correctly and robustly implement the intended xReserve workflows, particularly:

- Correct lifecycle enforcement for xReserve agreements and user actions
- Safety and consistency of state transitions under normal and adversarial inputs
- Authorization and role enforcement
- Replay protection / nonce correctness for attestation validation
- Correct mint/burn constraints for USDCx under xReserve-controlled conditions
- Edge-case handling in multi-step orchestration flows

System Workflow

The xReserve is structured around two primary workflows:

Deposit (Ethereum → Canton):

- A user deposits USDC into Circle's xReserve contract on Ethereum.
- xReserve produces a deposit attestation (via its attestation system / API) verifying the deposit event.

- The xReserve solution processes the attestation on Canton, enabling the user to mint USDCx (a USDC-backed token on Canton) in alignment with the deposit attestation.

Withdrawal (Canton → Ethereum):

- A user initiates a withdrawal by requesting to burn USDCx on Canton.
- The system generates and verifies the required burn/withdrawal attestations and submits them to xReserve.
- Upon successful verification, Circle's xReserve contract releases USDC on Ethereum to the designated destination address.

Scope

Repository and Commits:

- https://github.com/DACH-NY/usdc-cub/tree/4850d79823f6b2a218a2a56eb05696bdead70832/daml
- https://github.com/DACH-NY/canton-network-utilities/tree/9b026f8f3012b143724c2dded3378a2f196227bc/app/daml
- https://github.com/hyperledger-labs/splice/tree/cd768caf03d5330a31a73cd215f4f2a1ea74cc13

Findings Overview

Vulnerability Summary

17	9	0	8	0
Total Findings	Resolved	Partially Resolved	Acknowledged	Declined

0	Centralization		Centralization findings highlight privileged roles & functions and their capabilities, or instances where the project takes custody of users' assets.
0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
1	Major	1 Resolved	Major risks may include logical errors that, under specific circumstances, could result in fund losses or loss of project control.
1	Medium	1 Resolved	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
5	Minor	3 Resolved, 2 Acknowledged	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
10	Informational	4 Resolved, 6 Acknowledged	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.