Multiliquid: TreasuryDelegate

Security Review

Cantina Managed review by:
HickupHH3, Lead Security Researcher
Rvierdiiev, Security Researcher

February 26, 2026

Contents

1 Introduction **2**
- 1.1 About Cantina . . . 2
- 1.2 Disclaimer . . . 2
- 1.3 Risk assessment . . . 2
  - 1.3.1 Severity Classification . . . 2

2 Security Review Summary **3**
- 2.1 Scope . . . 3

3 Findings **4**
- 3.1 Medium Risk . . . 4
  - 3.1.1 Loss of yield for the new user when rates are not posted during the merge . . . 4
- 3.2 Low Risk . . . 4
  - 3.2.1 Future timestamp check skipped when MAX_AGE is 0 inside ChroniclePriceAdapter . . . 4
  - 3.2.2 User may enjoy higher yield when withdrawing stablecoins close to backfill time . . . 5
  - 3.2.3 Minted treasury fee tokens are unbacked . . . 6
  - 3.2.4 Posted rates according to natspec result in higher than expected yields . . . 6
- 3.3 Gas Optimization . . . 7
  - 3.3.1 Redundant null address checks . . . 7
  - 3.3.2 `finalLastAccrualDay` can read from `accrualEndDay` instead of storage value `lastAccrualDay[user]` . . . 7
  - 3.3.3 Unreachable `yieldMultiplier` initialization . . . 15
  - 3.3.4 Early return can be inclusive of equality case . . . 15
- 3.4 Informational . . . 16
  - 3.4.1 `WithheldCreditsDeque.toArray()` lacks `unchecked` wrapper . . . 16
  - 3.4.2 Simplify withheld-credits consumption in _getYieldAmount and _getAmountForTargetValue functions . . . 16
  - 3.4.3 MAX_AGE is not configurable in ChroniclePriceAdapter and ChainlinkPriceAdapter . . . 17
  - 3.4.4 Break out of loop if daily rate not posted . . . 17
  - 3.4.5 Sanity check `band != 0` . . . 17
  - 3.4.6 TreasuryDelegate initializer does not validate initial rate against management fee . . . 18
  - 3.4.7 Whitelist checks can be more robust . . . 18

1 Introduction

1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at cantina.xyz

1.2 Disclaimer

Cantina Managed provides a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While Cantina Managed endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that the Cantina Managed security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

1.3 Risk assessment

Severity level	Impact: High	Impact: Medium	Impact: Low
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings are a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

2 Security Review Summary

Multiliquid is a decentralized protocol that enables institutional-grade atomic swaps between multiple permissioned Real World Asset (RWA) tokens and multiple stablecoins.

From Jan 21st to Feb 4th the Cantina team conducted a review of Multiliquid on commit hash c400ce07. The team identified a total of **16** issues:

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	1	1	0
Low Risk	4	3	1
Gas Optimizations	4	4	0
Informational	7	7	0
Total	**16**	**15**	**1**

2.1 Scope

The security review had the following components in scope for Multiliquid on commit hash c400ce07:

```
src/upgrades
├── MultiliquidSwap.sol
└── ProtocolStablecoinDelegates
    ├── TreasuryDelegate.sol
    └── WithheldCreditsDeque.sol
```

3 Findings

3.1 Medium Risk

3.1.1 Loss of yield for the new user when rates are not posted during the merge

Severity: Medium Risk

Context: TreasuryDelegate.sol#L646-L649, TreasuryDelegate.sol#L858-L860

Description: In `TreasuryDelegate._mergeWithheldCredits`, when a new user's withheld credits are merged, `lastAccrualDay[user]` is always set to `block.timestamp / 1 days - 1`:

```
if (userCredits == 0) {
    lastAccrualDay[user] = block.timestamp / 1 days - 1;
}
```

However, `_calculateCreditMultiplier` skips days without posted rates:

```
for (uint256 day = startDay; day < currentDay; day++) {
    if (!ratePosted[day]) {
        // Skip days without posted rates (weekends until backfilled)
        continue;
    }
    // ... apply rate
}
```

This creates a scenario where users permanently lose yield for days without posted rates at merge time:

1. User deposits on Friday (Day 8). Rates exist for weekdays only.
2. On Monday (Day 11), merge runs. `_calculateCreditMultiplier` iterates days 8-10 but skips days 9-10 (weekend, no rates posted).
3. Merged credits receive yield only for Day 8.
4. `lastAccrualDay` is set to `currentDay - 1` = Day 10.
5. Future accrual loops start from `lastAccrualDay + 1` = Day 11, permanently skipping days 9-10.
6. Even if weekend rates are backfilled later via `correctDailyRate`, those days are never applied to this user since accrual has already advanced past them.

The user loses yield for any days that didn't have posted rates at the time their withheld credits merged.

Recommendation: Set `lastAccrualDay` to the minimum of `currentDay - 1` and `lastRatePostingDay` to ensure accrual doesn't advance beyond days with posted rates:

```
if (userCredits == 0) {
    uint256 targetDay = block.timestamp / 1 days - 1;
    lastAccrualDay[user] = targetDay < lastRatePostingDay ? targetDay :
    ↪  lastRatePostingDay;
}
```

3.2 Low Risk

3.2.1 Future timestamp check skipped when MAX_AGE is 0 inside ChroniclePriceAdapter

Severity: Low Risk

Context: ChroniclePriceAdapter.sol#L38-L41

Description.

In `ChroniclePriceAdapter.getPrice()`, the check `timestamp > block.timestamp` is inside the `if (MAX_AGE != 0)` block. When `MAX_AGE` is 0 (staleness disabled), that block is skipped, so future timestamps are not rejected. A future timestamp is invalid regardless of max age.

Recommendation.

```
if (timestamp > block.timestamp) revert InvalidPrice();
if (MAX_AGE != 0) {
    if (block.timestamp - timestamp > MAX_AGE) revert StalePrice(timestamp, MAX_AGE);
}
```

3.2.2 User may enjoy higher yield when withdrawing stablecoins close to backfill time

Severity: Low Risk

Context: TreasuryDelegate.sol#L673-L683

Description: The specification says that the user may lose a few basis points if withdrawing on weekends. While this is true for withheld credits because they can't be merged, it isn't necessarily true for main credits since it uses the `lastRatePostingDay`: if the future days to be backfilled have lower APY, then the user actually benefits.

Using the E2E backfill example:

- Sat 9am posts for Friday's rate of 4.52% APY.
- Tuesday 9am backfills with Sat + Sun + Mon rates of 4.5%, 4.5% & 4.48% APY respectively.

If the user withdraws on Monday, then the better rate of 4.52% is applied on the time elapsed since Sat.

Although is true generally for any case where the next posted day rate is lower than the current one, it's just a bit more pronounced on weekends because of the multi-day delay.

Proof of Concept:

```
function testBackFillBetterRate() external {
    usdc.mint(alice, ALICE_DEPOSIT * 2);

    for (uint i = 1; i <= 4; i++) {
        _simulateDailyWorker(startDay + i, RATE_W1_A);
    }

    // ========== DAY 5 (Friday) ==========
    // Alice deposits 2 x 10,000 USDC (stablecoin, T+1 earning)
    vm.warp((startDay + 1) * 1 days + 14 hours); // 2pm user action
    _swapUSDCIntoTreasury(alice, ALICE_DEPOSIT);
    _swapUSDCIntoTreasury(alice, ALICE_DEPOSIT);
    uint256 aliceTreasuryAmount = treasury.balanceOf(alice);

    _simulateDailyWorker(startDay + 5, RATE_W1_B);


    // we warp 15 minutes before backfill to show worst case in swapping prior to rates
    ↪  swap
    vm.warp((startDay + 9) * 1 days + 9 hours - 15 minutes);
    uint256 aliceUsdcBefore = usdc.balanceOf(alice);
    _swapTreasuryToUSDC(alice, aliceTreasuryAmount / 2, 0);
    uint256 aliceUsdcAfter = usdc.balanceOf(alice);
    uint256 aliceUsdcRedeemedBeforeBackfill = aliceUsdcAfter - aliceUsdcBefore;

    // backfill weekend
    _backfillWeekend(weekendRate1, RATE_W2_A, startDay + 6, startDay + 7, startDay + 8);

    // perform equivalent withdrawal
    aliceUsdcBefore = aliceUsdcAfter;
    _swapTreasuryToUSDC(alice, aliceTreasuryAmount / 2, 0);
    aliceUsdcAfter = usdc.balanceOf(alice);
    uint256 aliceUsdcRedeemedAfterBackfill = aliceUsdcAfter - aliceUsdcBefore;

    console.log("aliceUsdcRedeemedBeforeBackfill", aliceUsdcRedeemedBeforeBackfill);
    console.log("aliceUsdcRedeemedAfterBackfill", aliceUsdcRedeemedAfterBackfill);
```

```
    assertGt(aliceUsdcRedeemedBeforeBackfill, aliceUsdcRedeemedAfterBackfill);
}
```

Recommendation: Partially mitigate by posting a slightly smaller APY rates for Fridays, and / or adjusting future APYs to account for potential deficits.

3.2.3 Minted treasury fee tokens are unbacked

Severity: Low Risk

Context: TreasuryDelegate.sol#L261-L266

Description: By themselves, the treasury tokens have no intrinsic value, and must be backed by credits. The absence of adding withheld credits when minting the treasury tokens to the vault and custodian for the various swaps results in them not being able to redeem the fees; at least, not on-chain.

Recommendation: Add withheld credits in addition to the mints. The `isRWA` field for mints is recommended to be as such:

- RWA -> stablecoin = `deployStablecoin()`: `isRWA = true`.
- RWA -> RWA = `exchangeRWAs()`: `isRWA = true`.
- Stablecoin -> treasury = `_exchangeStablecoinsMint()`: `isRWA = false`.
- Treasury -> stablecoin = `_exchangeStablecoinsRedeem`: `isRWA = false`.

3.2.4 Posted rates according to natspec result in higher than expected yields

Severity: Low Risk

Context: TreasuryDelegate.sol#L339

Description: In the natspec and tests, the posted rate uses a simple conversion from the expected APY. However, this will result in slightly higher APY rates than expected due to the daily compounding formula.

For example, if we want to post a rate of 5% APY,

$$.(1 + \frac{r - f}{365})^{365} = 1.05.$$

.

where `f` is the management fee.

Solving for `r`, we get.

$$.r = 365 * [1.05^{1/365} - 1] + f.$$

.

which gives a net value of `48793425246426160` < `5e16`.

Proof Of Concept:

```
function test_fullYearAPYRate() public {
    uint256 apy = 0.05e18; // 5% APY
    uint256 multiplier = _computeMultiplier(apy, 365, WAD);
    console.log("Default 5% multiplier", multiplier);

    apy = 48793425246426160;
    multiplier = _computeMultiplier(apy, 365, WAD);
    console.log("Adjusted 5% multiplier", multiplier);
}
```

yields.

```
[PASS] test_fullYearAPYRate() (gas: 270214)
Logs:
  Default 5% APY 1051267496467462356
  Adjusted 5% APY 1050000000000021131
```

Recommendation: Switch the APY calculation to be additive:

```
multiplier += apy / DAYS_PER_YEAR;
```

Otherwise, `r` should be calculated as per the formula below.

$$.r = 365 * [(1 + y)^{1/365} - 1] + f.$$

. where `y` is the expected APY (e.g. 4.5% APY -> `y = 0.045`).

3.3 Gas Optimization

3.3.1 Redundant null address checks

Severity: Gas Optimization

Context: Treasury.sol#L165, Treasury.sol#L182

Description: The null address check is redundant as the subsequent call to `_mint` does it:

```
if (account == address(0)) {
   revert ERC20InvalidReceiver(address(0));
}
```

Recommendation: Consider removing it.

3.3.2 `finalLastAccrualDay` can read from `accrualEndDay` instead of storage value `lastAccrualDay[user]`

Severity: Gas Optimization

Context: TreasuryDelegate.sol#L772

Description: `accrualEndDay` can be initialized, then `finalLastAccrualDay` can use its value instead of doing a repeated SLOAD.

Recommendation:

```
  diff --git a/src/upgrades/ProtocolStablecoinDelegates/TreasuryDelegate.sol
  ↪  b/src/upgrades/ProtocolStablecoinDelegates/TreasuryDelegate.sol
  index 47695e0..17de25d 100644
-  -- a/src/upgrades/ProtocolStablecoinDelegates/TreasuryDelegate.sol
+  ++ b/src/upgrades/ProtocolStablecoinDelegates/TreasuryDelegate.sol
  @@ -722,7 +722,7 @@ contract TreasuryDelegate is StablecoinDelegateBase,
  ↪  IYieldBearingDelegate {
           uint256 userCredits = credits[user];
           uint256 currentDay = block.timestamp / 1 days;
           uint256 accrualStartDay = lastAccrualDay[user];
-          uint256 accrualEndDay;
+          uint256 accrualEndDay = accrualStartDay;

           uint256 userMultiplier = yieldMultiplier[user];
           if (userMultiplier == 0) userMultiplier = WAD;
  @@ -769,7 +769,7 @@ contract TreasuryDelegate is StablecoinDelegateBase,
  ↪  IYieldBearingDelegate {
           // - Complete if we've processed up to targetDay
           // - Complete if next day has no rate posted (can't continue anyway)
           // - Complete if user has no existing credits (nothing to catch up on)
-          uint256 finalLastAccrualDay = lastAccrualDay[user];
+          uint256 finalLastAccrualDay = accrualEndDay;
           uint256 nextDayToProcess = finalLastAccrualDay + 1;
```

```
            isComplete = (userCredits == 0) || (finalLastAccrualDay >= targetDay)
                || !ratePosted[nextDayToProcess];
```

Here's the gas diff. 1 fuzz test `testFuzz_DaysAccrued(uint256)` skews the overall gas reduction to be a net increase, which I assume to be an outlier.

```
↓ TreasuryDelegateExtremeValuesTest::test_LongTerm_ExtremCompounding_TenYears() (gas:
↪  196696296 → 196695936 | -360 -0.000%)
↓ TreasuryDelegateExtremeValuesTest::test_LongTerm_TenYears_NoOverflow() (gas: 196696273
↪  → 196695913 | -360 -0.000%)
↓ TreasuryDelegateExtremeValuesTest::test_LongTerm_FiveYears() (gas: 98560592 → 98560232
↪  | -360 -0.000%)
↓ TreasuryDelegateHighMultiplierFeeTest::testBug_HighMultiplierSwapShouldWorkButReverts(
↪  ) (gas: 78318855 → 78318135 | -720 -0.001%)
↓ TreasuryDelegateHighMultiplierFeeTest::testFeeScalingWithYield() (gas: 90583732 →
↪  90582832 | -900 -0.001%)
↓ TreasuryDelegateExtremeValuesTest::testFuzz_AccrualDays(uint256) (gas: 25219563 →
↪  25219203 | -360 -0.001%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_GasUsage_LongInactivity() (gas:
↪  52522628 → 52521728 | -900 -0.002%)
↓ TreasuryDelegateExtremeValuesTest::test_LongTerm_OneYear() (gas: 19983898 → 19983538 |
↪  -360 -0.002%)
↓ TreasuryDelegateExtremeValuesTest::test_ExtremeAPY_VerySmall() (gas: 19982908 →
↪  19982548 | -360 -0.002%)
↓ TreasuryDelegateExtremeValuesTest::test_Precision_MultiplierAccumulation() (gas:
↪  19982820 → 19982460 | -360 -0.002%)
↓ TreasuryDelegateExtremeValuesTest::test_SmallDeposit_ThousandWei_YieldCalculation()
↪  (gas: 19964182 → 19963822 | -360 -0.002%)
↓ TreasuryDelegateExtremeValuesTest::test_SmallDeposit_OneWei_EarnsYield() (gas:
↪  19930606 → 19930246 | -360 -0.002%)
↓ TreasuryDelegateAliceScenarioTest::testAliceScenarioExactOut() (gas: 20143900 →
↪  20143180 | -720 -0.004%)
↓ TreasuryDelegateStablecoinYieldTest::testYieldMultiplierScalesOutput() (gas: 19857835
↪  → 19857115 | -720 -0.004%)
↓ TreasuryDelegateAliceScenarioTest::testAliceScenarioStablecoinWithdrawal() (gas:
↪  19821262 → 19820542 | -720 -0.004%)
↓ TreasuryDelegateAliceScenarioTest::testAliceScenarioStablecoinWithdrawalSunday() (gas:
↪  19809120 → 19808400 | -720 -0.004%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_MultipleCalls_FullCatchUp() (gas:
↪  19620190 → 19619110 | -1080 -0.006%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_LimitsToMaxDays() (gas: 5540760 →
↪  5540220 | -540 -0.010%)
↓ TreasuryDelegateStablecoinYieldTest::testYieldMultiplierScalesFees() (gas: 5820064 →
↪  5819344 | -720 -0.012%)
↓ TreasuryDelegateYieldTest::testConstantAPYYieldAccrual() (gas: 2465883 → 2465523 |
↪  -360 -0.015%)
↓ TreasuryDelegateRevertTest::testRevert_DepositNonWhitelistedStablecoin() (gas: 1154617
↪  → 1154437 | -180 -0.016%)
↓ TreasuryDelegateExtremeValuesTest::test_LargeDeposit_WithYieldMultiplier() (gas:
↪  2065516 → 2065156 | -360 -0.017%)
↓ TreasuryDelegateInvariantTest::testInvariant2_MultiplierBounds_WithYield() (gas:
↪  2039017 → 2038657 | -360 -0.018%)
↓ TreasuryDelegateExtremeValuesTest::testFuzz_APYValues(uint256) (gas: 2035779 → 2035419
↪  | -360 -0.018%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_ZeroMaxDays_Unlimited() (gas:
↪  3048486 → 3047946 | -540 -0.018%)
↓ TreasuryDelegateExtremeValuesTest::test_RateEdgeCase_DailyRateChanges() (gas: 2008256
↪  → 2007896 | -360 -0.018%)
↓ TreasuryDelegateExactOutFeeBugTest::testSwapIntoRWAExactOutWithExtremeYield() (gas:
↪  3912960 → 3912240 | -720 -0.018%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_YieldMatchesFullAccrual() (gas:
↪  6064253 → 6062633 | -1620 -0.027%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_DelaysWithheldMergeUntilCaughtUp()
↪  (gas: 3211652 → 3210752 | -900 -0.028%)
```

```
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_ReturnsCorrectDaysAccrued() (gas:
↪  3069904 → 3069004 | -900 -0.029%)
↓ TreasuryDelegateExactOutFeeBugTest::testSwapIntoRWAExactOutWithBothFees() (gas:
↪  2385142 → 2384422 | -720 -0.030%)
↓ TreasuryDelegateExactOutFeeBugTest::testSwapIntoRWAExactOutUserGetsExactOutput() (gas:
↪  2319844 → 2319124 | -720 -0.031%)
↓ TreasuryDelegateExactOutFeeBugTest::testExactOutBugMathExplanation() (gas: 2316549 →
↪  2315829 | -720 -0.031%)
↓ TreasuryDelegateExactOutFeeBugTest::testSwapIntoRWAExactOutCreditsReduction() (gas:
↪  2309361 → 2308641 | -720 -0.031%)
↓ TreasuryDelegateExactOutFeeBugTest::testSwapIntoRWAExactOutFeeCollection() (gas:
↪  2308300 → 2307580 | -720 -0.031%)
↓ TreasuryDelegateHighMultiplierFeeTest::testCombinedFeesAtModerateMultiplier() (gas:
↪  2281436 → 2280716 | -720 -0.032%)
↓ TreasuryDelegateExactOutFeeBugTest::testSwapIntoRWAExactInNoFeeBug() (gas: 2235027 →
↪  2234307 | -720 -0.032%)
↓ TreasuryDelegateDepositTest::testDepositAfterYieldAccrual() (gas: 2230715 → 2229995 |
↪  -720 -0.032%)
↓ TreasuryDelegateHighMultiplierFeeTest::testVaultReceivesFaceValueTokensNotYieldValue()
↪  (gas: 2219489 → 2218769 | -720 -0.032%)
↓ TreasuryDelegateDepositTest::testDepositExactInWithAllFees() (gas: 534205 → 534025 |
↪  -180 -0.034%)
↓ TreasuryDelegateRedemptionTest::testPartialRedemptionPreservesYield() (gas: 2127598 →
↪  2126878 | -720 -0.034%)
↓ TreasuryDelegateWithheldYieldBugTest::testRWAWithheldCreditsNoYieldOnEarlyRedemption()
↪  (gas: 527370 → 527190 | -180 -0.034%)
↓ TreasuryDelegateSwapFeeTest::testFeeChangesWithVolume() (gas: 507546 → 507366 | -180
↪  -0.035%)
↓ TreasuryDelegateRevertTest::testRevert_RedeemToNonWhitelistedStablecoin() (gas:
↪  1512138 → 1511598 | -540 -0.036%)
↓ TreasuryDelegateMultiUserTest::testUsersSameDepositDifferentYield() (gas: 3892628 →
↪  3891188 | -1440 -0.037%)
↓ TreasuryDelegateYieldTest::testYieldAccrualWithManagementFee() (gas: 971061 → 970701 |
↪  -360 -0.037%)
↓ TreasuryDelegateDepositTest::testDepositExactInWithAcceptanceFee() (gas: 484774 →
↪  484594 | -180 -0.037%)
↓ TreasuryDelegateExtremeValuesTest::test_ExtremeAPY_HundredPercent() (gas: 961376 →
↪  961016 | -360 -0.037%)
↓ TreasuryDelegateInvariantTest::testInvariant7_LastAccrualDay_NotFuture() (gas: 960785
↪  → 960425 | -360 -0.037%)
↓ TreasuryDelegateSwapFeeTest::testAcceptanceFeeGoesToCustody() (gas: 474819 → 474639 |
↪  -180 -0.038%)
↓ TreasuryDelegateCreditsForTargetValueTest::testAccrueAndGetAmountForTargetValue()
↪  (gas: 947746 → 947386 | -360 -0.038%)
↓ TreasuryDelegateCreditsForTargetValueTest::testGetAmountForTargetValue_ConsistentWithG
↪  etYieldAmount() (gas: 942829 → 942469 | -360 -0.038%)
↓ TreasuryDelegateCreditsForTargetValueTest::testGetAmountForTargetValue_WithYield()
↪  (gas: 941842 → 941482 | -360 -0.038%)
↓ TreasuryDelegateDepositTest::testDepositExactInWithProtocolFee() (gas: 469544 → 469364
↪  | -180 -0.038%)
↓ TreasuryDelegateWithheldYieldBugTest::testGetAmountForTargetValueWithWithheldCredits()
↪  (gas: 468338 → 468158 | -180 -0.038%)
↓ TreasuryDelegateWithheldYieldBugTest::testGetYieldAmountForWithheldCredits() (gas:
↪  468055 → 467875 | -180 -0.038%)
↓ TreasuryDelegateYieldTest::testYieldAccrualWithMidpointDeposit() (gas: 2768163 →
↪  2767083 | -1080 -0.039%)
↓ TreasuryDelegateDepositTest::testUSDCDepositExactOut() (gas: 460926 → 460746 | -180
↪  -0.039%)
↓ TreasuryDelegateYieldTest::testManagementFeeChangeAffectsFutureRates() (gas: 916369 →
↪  916009 | -360 -0.039%)
↓ TreasuryDelegateFuzzTest::testFuzz_ProtocolFeeRate(uint128) (gas: 454871 → 454691 |
↪  -180 -0.040%)
↓ TreasuryDelegateSwapFeeTest::testProtocolFeeGoesToVault() (gas: 454561 → 454381 | -180
↪  -0.040%)
↓ TreasuryDelegateExtremeValuesTest::test_FeeEdgeCase_MaxProtocolFee() (gas: 450948 →
↪  450768 | -180 -0.040%)
```

```
↓ TreasuryDelegateYieldTest::testYieldAccrualWithMidpointDepositExactOut() (gas: 2702027
↪  → 2700947 | -1080 -0.040%)
↓ TreasuryDelegateExtremeValuesTest::test_Precision_FeeRounding() (gas: 450172 → 449992
↪  | -180 -0.040%)
↓ TreasuryDelegateDepositTest::testDepositExactInZeroFee() (gas: 420067 → 419887 | -180
↪  -0.043%)
↓ TreasuryDelegateFuzzTest::testFuzz_DepositAmount(uint256) (gas: 416974 → 416794 | -180
↪  -0.043%)
↓ DeploymentDayUnderflowTest::testDeploymentDayWithCreditsViewFunctionsWork() (gas:
↪  415993 → 415813 | -180 -0.043%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testSwapStablecoinToStablecoinDepositN
↪  oYield() (gas: 414555 → 414375 | -180 -0.043%)
↓ TreasuryDelegateDepositTest::testDepositLargeAmount() (gas: 411571 → 411391 | -180
↪  -0.044%)
↓ TreasuryDelegateDepositTest::testDepositSmallAmount() (gas: 411564 → 411384 | -180
↪  -0.044%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_MultiplierAccumulatesCorrectly()
↪  (gas: 2049336 → 2048436 | -900 -0.044%)
↓ TreasuryDelegateExtremeValuesTest::testFuzz_DepositAmounts(uint256) (gas: 409453 →
↪  409273 | -180 -0.044%)
↓ TreasuryDelegateExtremeValuesTest::test_SmallDeposit_OneWei() (gas: 405597 → 405417 |
↪  -180 -0.044%)
↓ TreasuryDelegateExtremeValuesTest::test_LargeDeposit_SafeMaximum() (gas: 405581 →
↪  405401 | -180 -0.044%)
↓ TreasuryDelegateExtremeValuesTest::test_LargeDeposit_Uint128Max() (gas: 405558 →
↪  405378 | -180 -0.044%)
↓ TreasuryDelegateStablecoinYieldTest::testAccrualBeforeRedemption() (gas: 964988 →
↪  964556 | -432 -0.045%)
↓ TreasuryDelegateInvariantTest::testInvariant2_MultiplierBounds_ZeroRate() (gas: 787491
↪  → 787131 | -360 -0.046%)
↓ TreasuryDelegateDepositTest::testDepositExactOutMatchesExactIn() (gas: 757809 → 757449
↪  | -360 -0.048%)
↓ TreasuryDelegateStablecoinYieldTest::testRedemptionWithStaleMultiplier() (gas: 1502610
↪  → 1501890 | -720 -0.048%)
↓ TreasuryDelegateExtremeValuesTest::test_ExtremeAPY_ZeroPercent() (gas: 698055 → 697695
↪  | -360 -0.052%)
↓ TreasuryDelegateInvariantTest::testInvariant4_NoDoubleCounting_DuringMerge() (gas:
↪  550085 → 549797 | -288 -0.052%)
↓ TreasuryDelegatePartialDayTest::testPartialDayInterestOnDeposit() (gas: 687495 →
↪  687135 | -360 -0.052%)
↓ TreasuryDelegateInvariantTest::testInvariant2_MultiplierBounds_AfterDeposit() (gas:
↪  543691 → 543403 | -288 -0.053%)
↓ TreasuryDelegateWithheldYieldBugTest::testGetYieldAmountWithNoWithheldCredits() (gas:
↪  678970 → 678610 | -360 -0.053%)
↓ TreasuryDelegateInvariantTest::testInvariant7_LastAccrualDay_AfterFirstRate() (gas:
↪  540486 → 540198 | -288 -0.053%)
↓ TreasuryDelegateInvariantTest::testInvariant6_NoNegativeBacking_MultiUser() (gas:
↪  1004418 → 1003878 | -540 -0.054%)
↓ TreasuryDelegateYieldTest::testRWAToTreasuryExactOutRedemption() (gas: 1062050 →
↪  1061474 | -576 -0.054%)
↓ TreasuryDelegateYieldTest::testFullYieldCycleWithManagementFee() (gas: 1042308 →
↪  1041732 | -576 -0.055%)
↓ TreasuryDelegateYieldTest::testFullRedemptionResetsMultiplier() (gas: 1039288 →
↪  1038712 | -576 -0.055%)
↓ TreasuryDelegateYieldTest::testWithheldCreditsMergeWithPartialYieldForMissingRates()
↪  (gas: 505130 → 504842 | -288 -0.057%)
↓ DeploymentDayUnderflowTest::testOneDayAfterDeploymentPartialYieldWorks() (gas: 498618
↪  → 498330 | -288 -0.058%)
↓ TreasuryDelegateInvariantTest::testInvariant8_CorrectionWithinSeriesDoesNotAffectLast(
↪  ) (gas: 986268 → 985692 | -576 -0.058%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testCalculateStablecoinToStablecoinWit
↪  hYieldMatchesSwap() (gas: 986202 → 985626 | -576 -0.058%)
↓ TreasuryDelegateStablecoinYieldTest::testUSDCToTreasuryYieldRedemptionToUSDC() (gas:
↪  985984 → 985408 | -576 -0.058%)
↓ TreasuryDelegateExtremeValuesTest::test_LargeDeposit_Multiple() (gas: 920277 → 919737
↪  | -540 -0.059%)
```

```
↓ TreasuryDelegateRedemptionTest::testFullRedemptionResetsState() (gas: 980236 → 979660
↪  | -576 -0.059%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_WithheldCreditsYieldMatchesFullAcc
↪  rual() (gas: 3673992 → 3671832 | -2160 -0.059%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testSwapStablecoinToStablecoinRedempti
↪  onWithYield() (gas: 977516 → 976940 | -576 -0.059%)
↓ TreasuryDelegateRedemptionTest::testRedeemEntireBalance() (gas: 972982 → 972406 | -576
↪  -0.059%)
↓ TreasuryDelegateStablecoinYieldTest::testManualAccrualThenRedeem() (gas: 971135 →
↪  970559 | -576 -0.059%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testSwapIntoRWAWithFeesAndYieldScaling
↪  () (gas: 1210375 → 1209655 | -720 -0.059%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testSwapIntoRWAExactOutWithYieldScalin
↪  g() (gas: 1203648 → 1202928 | -720 -0.060%)
↓ TreasuryDelegatePartialDayTest::testMultipleDepositsAtDifferentTimes() (gas: 721749 →
↪  721317 | -432 -0.060%)
↓ TreasuryDelegateInvariantTest::testInvariant4_NoDoubleCounting_StaggeredDeposits()
↪  (gas: 718497 → 718065 | -432 -0.060%)
↓ TreasuryDelegateSameDayE2EStablecoin::testMultiUserDifferentDepositTimes() (gas:
↪  1427658 → 1426794 | -864 -0.061%)
↓ TreasuryDelegateCreditsForTargetValueTest::testExactOut_MatchesViewFunction() (gas:
↪  1187167 → 1186447 | -720 -0.061%)
↓ TreasuryDelegatePartialDayTest::testPartialDayInterestOnWithdrawal() (gas: 712127 →
↪  711695 | -432 -0.061%)
↓ TreasuryDelegatePartialDayTest::testDailyInterestAcrossDifferentRates() (gas: 712074 →
↪  711642 | -432 -0.061%)
↓ TreasuryDelegateSwapFeeTest::testStablecoinSwapExactOutWithFees() (gas: 1185320 →
↪  1184600 | -720 -0.061%)
↓ TreasuryDelegateWeekendBackfillTest::testStablecoin_LongWeekendBackfill() (gas: 705802
↪  → 705370 | -432 -0.061%)
↓ TreasuryDelegateDepositTest::testMultipleDepositsWeightedAverageMultiplier() (gas:
↪  1168785 → 1168065 | -720 -0.062%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_WithheldCreditsOnly_StillMerges()
↪  (gas: 455490 → 455202 | -288 -0.063%)
↓ TreasuryDelegateMultiUserTest::testMultipleUsersIndependentMultipliers() (gas: 1789055
↪  → 1787903 | -1152 -0.064%)
↓ TreasuryDelegatePartialDayTest::testFirstPartialDayYield() (gas: 670456 → 670024 |
↪  -432 -0.064%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_InterfaceCompliance() (gas: 445881
↪  → 445593 | -288 -0.065%)
↓ TreasuryDelegateWeekendBackfillTest::testStablecoin_DepositFridayBackfillMondayWithdra
↪  wMonday() (gas: 668062 → 667630 | -432 -0.065%)
↓ TreasuryDelegateYieldTest::testPartialExactOutRedemptionPreservesMultiplier() (gas:
↪  1109134 → 1108414 | -720 -0.065%)
↓ TreasuryDelegateYieldTest::testPartialRedemptionPreservesMultiplier() (gas: 1107377 →
↪  1106657 | -720 -0.065%)
↓ TreasuryDelegateWeekendBackfillTest::testStablecoin_DepositSaturdayBackfillMondayWithd
↪  rawMonday() (gas: 663883 → 663451 | -432 -0.065%)
↓ TreasuryDelegateSwapFeeTest::testStablecoinSwapExactInWithFees() (gas: 1103404 →
↪  1102684 | -720 -0.065%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testSwapIntoRWAWithYieldScaling()
↪  (gas: 1096191 → 1095471 | -720 -0.066%)
↓ TreasuryDelegateSecondaryMarketTest::testSecondaryBuyerCanRedeemAfterDeposit() (gas:
↪  876544 → 875968 | -576 -0.066%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testCalculateRWAAmtWithYieldMatchesSwa
↪  p() (gas: 1091979 → 1091259 | -720 -0.066%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testSwapStablecoinToStablecoinExactOut
↪  WithYield() (gas: 1087083 → 1086363 | -720 -0.066%)
↓ TreasuryDelegateSecondaryMarketTest::testExactOutRedemptionEnforcesCredits() (gas:
↪  537203 → 536843 | -360 -0.067%)
↓ TreasuryDelegateRedemptionTest::testPartialUSDCRedemptionPreservesMultiplier() (gas:
↪  1069002 → 1068282 | -720 -0.067%)
↓ TreasuryDelegateRedemptionTest::testPartialRedemption75Percent() (gas: 1068827 →
↪  1068107 | -720 -0.067%)
↓ TreasuryDelegateRedemptionTest::testPartialRedemption25Percent() (gas: 1068624 →
↪  1067904 | -720 -0.067%)
```

```
↓ TreasuryDelegateSameDayE2EStablecoin::test24HourAndOneMinuteYield() (gas: 636676 →
↪  636244 | -432 -0.068%)
↓ TreasuryDelegateSameDayE2E::testMultiUserDifferentDepositTimes() (gas: 1877444 →
↪  1876148 | -1296 -0.069%)
↓ TreasuryDelegateExtremeValuesTest::test_FeeEdgeCase_CombinedHighFees() (gas: 622159 →
↪  621727 | -432 -0.069%)
↓ TreasuryDelegateSameDayE2EStablecoin::testMidnightCrossingYield() (gas: 621603 →
↪  621171 | -432 -0.069%)
↓ TreasuryDelegateInvariantTest::testFIFO_MergeOrdering() (gas: 828350 → 827774 | -576
↪  -0.070%)
↓ TreasuryDelegateWeekendBackfillTest::testStablecoin_DepositSundayNightBackfillMondayWi
↪  thdrawMonday() (gas: 620983 → 620551 | -432 -0.070%)
↓ TreasuryDelegateYieldTest::testBackfillMissingRatesAllowsContinuedAccrual() (gas:
↪  768610 → 768070 | -540 -0.070%)
↓ TreasuryDelegateYieldTest::testWithdrawalWithMissingRates() (gas: 811829 → 811253 |
↪  -576 -0.071%)
↓ TreasuryDelegateRedemptionTest::testFullRedemptionWithMaxFees() (gas: 601264 → 600832
↪  | -432 -0.072%)
↓ TreasuryDelegatePartialDayExploitTest::testExploit_MultiDayGap() (gas: 749963 → 749423
↪  | -540 -0.072%)
↓ TreasuryDelegateFuzzTest::testFuzz_CombinedFees(uint128,uint256) (gas: 588086 → 587656
↪  | -430 -0.073%)
↓ TreasuryDelegateInvariantTest::testEvent_WithheldCreditsReduced() (gas: 586320 →
↪  585888 | -432 -0.074%)
↓ TreasuryDelegateSameDayE2EStablecoin::testLIFOPendingCreditsRemovedOnWithdrawWithYield
↪  () (gas: 583078 → 582646 | -432 -0.074%)
↓ TreasuryDelegateSameDayE2EStablecoin::testDepositBeforeFirstRatePostNextDayWithYield()
↪  (gas: 582860 → 582428 | -432 -0.074%)
↓ TreasuryDelegateWeekendBackfillTest::testStablecoin_WithdrawSaturdayBeforeBackfill()
↪  (gas: 578645 → 578213 | -432 -0.075%)
↓ TreasuryDelegateFuzzTest::testFuzz_PartialRedemption(uint256,uint256) (gas: 577558 →
↪  577126 | -432 -0.075%)
↓ TreasuryDelegateExtremeValuesTest::test_DequeStress_MergeAndNewDeposits() (gas: 721809
↪  → 721269 | -540 -0.075%)
↓ TreasuryDelegateExtremeValuesTest::testFuzz_PartialRedemptions(uint256,uint256) (gas:
↪  572761 → 572329 | -432 -0.075%)
↓ TreasuryDelegateWithheldYieldBugTest::testReleasedCreditsProperlYReceiveYield() (gas:
↪  763614 → 763038 | -576 -0.075%)
↓ TreasuryDelegateRedemptionTest::testRedeemSmallAmount() (gas: 570359 → 569927 | -432
↪  -0.076%)
↓ TreasuryDelegateSameDayE2E::testYieldOnlyForCompleteDays() (gas: 565875 → 565443 |
↪  -432 -0.076%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_IsCompleteWhenCaughtUp() (gas:
↪  706644 → 706104 | -540 -0.076%)
↓ TreasuryDelegateSwapFeeTest::testRedemptionFeeGoesToCustody() (gas: 559801 → 559369 |
↪  -432 -0.077%)
↓ TreasuryDelegateWithheldYieldBugTest::testWithheldCreditsDoNotReceiveYield() (gas:
↪  555825 → 555393 | -432 -0.078%)
↓ TreasuryDelegateInvariantTest::testFuzz_Invariant1_CreditConservation(uint256,uint256,
↪  uint256) (gas: 921228 → 920508 | -720 -0.078%)
↓ TreasuryDelegateSwapFeeTest::testChangeFeesMidOperation() (gas: 549388 → 548956 | -432
↪  -0.079%)
↓ TreasuryDelegateExtremeValuesTest::test_BatchAccrue_MixedUsers() (gas: 684358 → 683818
↪  | -540 -0.079%)
↓ TreasuryDelegateWeekendBackfillTest::testMixed_StablecoinAndRWAOverWeekend() (gas:
↪  911474 → 910754 | -720 -0.079%)
↓ TreasuryDelegateWithheldYieldBugTest::testRedemptionAtExactReleaseTimestamp() (gas:
↪  542236 → 541804 | -432 -0.080%)
↓ TreasuryDelegateExtremeValuesTest::test_FullWithdrawal_ClearsState() (gas: 716962 →
↪  716386 | -576 -0.080%)
↓ TreasuryDelegateInvariantTest::testInvariant3_WithheldCreditsOrdering_MultipleDeposits
↪  () (gas: 1118704 → 1117804 | -900 -0.080%)
↓ TreasuryDelegateWeekendBackfillTest::testRWA_DepositFridayBackfillMondayWithdrawMonday
↪  () (gas: 715916 → 715340 | -576 -0.080%)
↓ TreasuryDelegateInvariantTest::testLIFO_WithdrawalOrdering() (gas: 893129 → 892409 |
↪  -720 -0.081%)
```

```
↓ TreasuryDelegateWeekendBackfillTest::testRWA_DepositSaturdayBackfillMondayWithdrawMond
↪  ay() (gas: 712560 → 711984 | -576 -0.081%)
↓ TreasuryDelegateSameDayE2EStablecoin::testMultipleDepositsSameDayWithdrawNextDay()
↪  (gas: 889998 → 889278 | -720 -0.081%)
↓ TreasuryDelegateCreditsForTargetValueTest::testGetAmountForTargetValue_SingleBucket_WA
↪  DMultiplier() (gas: 444204 → 443844 | -360 -0.081%)
↓ TreasuryDelegateInvariantTest::testInvariant1_CreditConservation_SingleUser() (gas:
↪  664867 → 664327 | -540 -0.081%)
↓ TreasuryDelegateRevertTest::testRevert_DepositInsufficientUSDCApproval() (gas: 221473
↪  → 221293 | -180 -0.081%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_StopsAtMissingRate() (gas: 663171
↪  → 662631 | -540 -0.081%)
↓ TreasuryDelegateInvariantTest::testLIFO_WithheldBeforeMainCredits() (gas: 877616 →
↪  876896 | -720 -0.082%)
↓ TreasuryDelegateDoubleCountingTest::testNoDoubleCountWhenMissingDaysPostedAfterMerge()
↪  (gas: 874510 → 873790 | -720 -0.082%)
↓ TreasuryDelegatePartialDayExploitTest::testExploit_PartialDayToFullDayDoubleCount()
↪  (gas: 652465 → 651925 | -540 -0.083%)
↓ TreasuryDelegateMultiliquidSwapIntegrationTest::testMultiUserYieldIndependence() (gas:
↪  1366552 → 1365400 | -1152 -0.084%)
↓ TreasuryDelegateDepositTest::testDepositAfterPartialRedemption() (gas: 1275298 →
↪  1274218 | -1080 -0.085%)
↓ TreasuryDelegateSameDayE2E::testZeroNetRateAfterManagementFee() (gas: 672020 → 671444
↪  | -576 -0.086%)
↓ TreasuryDelegateRedemptionTest::testRedeemUpToCreditLimit() (gas: 502618 → 502186 |
↪  -432 -0.086%)
↓ TreasuryDelegateWithheldYieldBugTest::testExactOutRedemptionWithWithheldCredits()
↪  (gas: 625610 → 625070 | -540 -0.086%)
↓ TreasuryDelegateRedemptionTest::testSecondaryBuyerCanRedeemAfterDeposit() (gas: 499708
↪  → 499276 | -432 -0.086%)
↓ TreasuryDelegatePartialDayTest::testWithdrawBeforeEarningStartDayCompletesNoYield()
↪  (gas: 499571 → 499139 | -432 -0.086%)
↓ TreasuryDelegateStablecoinYieldTest::testNoYieldOnSameDayDeposit() (gas: 498130 →
↪  497698 | -432 -0.087%)
↓ TreasuryDelegateExtremeValuesTest::test_WithdrawalOrder_LIFOFromWithheld() (gas:
↪  829051 → 828331 | -720 -0.087%)
↓ TreasuryDelegateFuzzTest::testFuzz_RedemptionAmount(uint256) (gas: 496740 → 496308 |
↪  -432 -0.087%)
↓ TreasuryDelegateSameDayE2E::testSameDayDepositWithdrawNoYield() (gas: 495882 → 495450
↪  | -432 -0.087%)
↓ TreasuryDelegateSameDayE2E::testMultipleDepositsSameDayFullWithdraw() (gas: 821281 →
↪  820561 | -720 -0.088%)
↓ TreasuryDelegateFuzzTest::testFuzz_DepositRedeemInvariant(uint256) (gas: 492006 →
↪  491574 | -432 -0.088%)
↓ TreasuryDelegateDoubleCountingTest::testStablecoinMergeWaitsForStartDayRatePosted()
↪  (gas: 489859 → 489427 | -432 -0.088%)
↓ TreasuryDelegateInvariantTest::testInvariant3_WithheldCreditsOrdering_AfterPartialMerg
↪  e() (gas: 1223118 → 1222038 | -1080 -0.088%)
↓ TreasuryDelegateInvariantTest::testFuzz_TotalCreditsNeverExceedDeposits(uint256[3],uin
↪  t256[3]) (gas: 1455066 → 1453778 | -1288 -0.089%)
↓ TreasuryDelegateSameDayE2E::testLIFOPendingCreditsRemovedOnWithdraw() (gas: 486510 →
↪  486078 | -432 -0.089%)
↓ TreasuryDelegateSameDayE2E::testDepositBeforeFirstRatePost() (gas: 485640 → 485208 |
↪  -432 -0.089%)
↓ TreasuryDelegateWithheldYieldBugTest::testMultipleWithheldBucketsNoYield() (gas:
↪  1125150 → 1124142 | -1008 -0.090%)
↓ TreasuryDelegateSecondaryMarketTest::testRWARedemptionEnforcesCredits() (gas: 599775 →
↪  599235 | -540 -0.090%)
↓ TreasuryDelegateInvariantTest::testInvariant5_ZeroCreditsCannotRedeem() (gas: 591725 →
↪  591185 | -540 -0.091%)
↓ TreasuryDelegateInvariantTest::testInvariant1_CreditConservation_MultiUser() (gas:
↪  1769763 → 1768143 | -1620 -0.092%)
↓ TreasuryDelegateWeekendBackfillTest::testRWA_WithdrawSaturdayBeforeBackfill() (gas:
↪  624866 → 624290 | -576 -0.092%)
↓ TreasuryDelegateWeekendBackfillTest::testStablecoin_MultipleUsersOverWeekend() (gas:
↪  1401348 → 1400052 | -1296 -0.092%)
```

```
↓ TreasuryDelegateSameDayE2E::testPartialSameDayWithdraw() (gas: 931200 → 930336 | -864
↪  -0.093%)
↓ TreasuryDelegateSameDayE2EStablecoin::testSameDayDepositWithdrawNoYield() (gas: 465540
↪  → 465108 | -432 -0.093%)
↓ TreasuryDelegateSameDayE2EStablecoin::testPartialSameDayWithdrawNoYield() (gas: 929769
↪  → 928905 | -864 -0.093%)
↓ TreasuryDelegateSecondaryMarketTest::testSecondaryBuyerCannotRedeemWithoutCredits()
↪  (gas: 579523 → 578983 | -540 -0.093%)
↓ TreasuryDelegateSameDayE2EStablecoin::testLIFOPendingCreditsRemovedOnWithdrawNoYield()
↪  (gas: 457043 → 456611 | -432 -0.095%)
↓ TreasuryDelegateSameDayE2EStablecoin::testDepositBeforeFirstRatePostSameDayNoYield()
↪  (gas: 456306 → 455874 | -432 -0.095%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_AlreadyUpToDate_ReturnsComplete()
↪  (gas: 452306 → 451874 | -432 -0.096%)
↓ TreasuryDelegateSameDayE2EStablecoin::testMultipleDepositsSameDayFullWithdrawNoYield()
↪  (gas: 751944 → 751224 | -720 -0.096%)
↓ TreasuryDelegateWithheldYieldBugTest::testMixedWithheldAndMainCreditsYieldCalculation(
↪  ) (gas: 902244 → 901380 | -864 -0.096%)
↓ TreasuryDelegateInvariantTest::testInvariant5_RedemptionBoundedByCredits() (gas:
↪  559926 → 559386 | -540 -0.096%)
↓ TreasuryDelegateSameDayE2EStablecoin::test23Hours59MinutesNoYield() (gas: 447351 →
↪  446919 | -432 -0.097%)
↓ TreasuryDelegateExtremeValuesTest::test_DequeStress_PartialWithdrawals() (gas: 1779966
↪  → 1778238 | -1728 -0.097%)
↓ TreasuryDelegateExtremeValuesTest::test_BatchAccrue_ManyUsers() (gas: 29651684 →
↪  29622884 | -28800 -0.097%)
↓ TreasuryDelegateAliceScenarioTest::testAliceScenarioExactMultiplier() (gas: 586861 →
↪  586285 | -576 -0.098%)
↓ TreasuryDelegateRedemptionTest::testRedeemMoreThanCreditsReverts() (gas: 534492 →
↪  533952 | -540 -0.101%)
↓ TreasuryDelegateRevertTest::testRevert_DepositInsufficientUSDCBalance() (gas: 175890 →
↪  175710 | -180 -0.102%)
↓ TreasuryDelegateSameDayE2E::testLIFOOrderPendingThenCredits() (gas: 973442 → 972434 |
↪  -1008 -0.104%)
↓ TreasuryDelegateRevertTest::testRevert_RedeemInsufficientTreasuryBalance() (gas:
↪  521484 → 520944 | -540 -0.104%)
↓ TreasuryDelegateWithheldYieldBugTest::testPartialWithheldRedemptionNoYield() (gas:
↪  693408 → 692688 | -720 -0.104%)
↓ TreasuryDelegateRevertTest::testRevert_RedeemCustodyInsufficientUSDCWithCredits()
↪  (gas: 506789 → 506249 | -540 -0.107%)
↓ TreasuryDelegateRedemptionTest::testSecondaryBuyerCannotRedeemWithoutCredits() (gas:
↪  333896 → 333536 | -360 -0.108%)
↓ TreasuryDelegateSameDayE2E::testFullE2EScenario() (gas: 2404029 → 2401437 | -2592
↪  -0.108%)
↓ TreasuryDelegateRevertTest::testRevert_RedeemCustodyInsufficientUSDC() (gas: 492543 →
↪  492003 | -540 -0.110%)
↓ TreasuryDelegateSameDayE2EStablecoin::testFullE2EScenario() (gas: 2348336 → 2345744 |
↪  -2592 -0.110%)
↓ TreasuryDelegateInvariantTest::testEvent_MainCreditsReduced() (gas: 649599 → 648879 |
↪  -720 -0.111%)
↓ TreasuryDelegateSameDayE2EStablecoin::testLIFOOrderPendingThenCredits() (gas: 906704 →
↪  905696 | -1008 -0.111%)
↓ TreasuryDelegateWeekendBackfillTest::testRWA_MultipleUsersOverWeekend() (gas: 1534840
↪  → 1533112 | -1728 -0.113%)
↓ TreasuryDelegateExtremeValuesTest::test_DequeStress_ManyDepositsOneDay() (gas: 7911203
↪  → 7902203 | -9000 -0.114%)
↓ TreasuryDelegateRevertTest::testRevert_SwapWithoutStablecoinCustodySet() (gas: 9513654
↪  → 9502007 | -11647 -0.122%)
↓ TreasuryDelegateRevertTest::testRevert_SwapWithoutRWACustodySet() (gas: 9513416 →
↪  9501769 | -11647 -0.122%)
↓ TreasuryDelegateSecondaryMarketTest::testMultipleDepositsAccumulateCredits() (gas:
↪  697521 → 696657 | -864 -0.124%)
↓ TreasuryDelegateSecondaryMarketTest::testPartialTransferAliceKeepsCredits() (gas:
↪  700663 → 699763 | -900 -0.128%)
↓ TreasuryDelegateRedemptionTest::testMultiplePartialRedemptions() (gas: 1120163 →
↪  1118723 | -1440 -0.129%)
```

```
↓ TreasuryDelegateSecondaryMarketTest::testRedemptionCappedByCredits() (gas: 551536 →
↪  550816 | -720 -0.131%)
↓ TreasuryDelegateYieldTest::testSecondDepositWithMissingRateForDepositDay() (gas:
↪  1061276 → 1059836 | -1440 -0.136%)
↓ TreasuryDelegateExtremeValuesTest::test_DequeStress_RapidDepositWithdraw() (gas:
↪  1590350 → 1588190 | -2160 -0.136%)
↓ TreasuryDelegateExtremeValuesTest::test_Precision_ManyAccrualCycles() (gas: 6601320 →
↪  6583140 | -18180 -0.275%)
↓ TreasuryDelegatePagedAccrualTest::test_PagedAccrual_NoCredits_ReturnsComplete() (gas:
↪  55438 → 55258 | -180 -0.325%)
↓ TreasuryDelegate30DayE2ETest::testFull30DayMultiUserE2E() (gas: 5550460 → 5529724 |
↪  -20736 -0.374%)
↓ TreasuryDelegateExtremeValuesTest::test_BatchAccrue_UsersWithNoCredits() (gas: 223862
↪  → 222962 | -900 -0.402%)
↓ TreasuryDelegateSameRateTest::testFull70DayMultiUserE2E() (gas: 13704077 → 13647629 |
↪  -56448 -0.412%)
↓ TreasuryDelegatePagedAccrualTest::testFuzz_PagedAccrual_MaxDays(uint256) (gas:
↪  13713562 → 13608910 | -104652 -0.763%)
↑ TreasuryDelegatePagedAccrualTest::testFuzz_PagedAccrual_CompleteAfterMultipleCalls(uin
↪  t256,uint256) (gas: 6203747 → 6255973 | 52226 0.842%)
↑ TreasuryDelegateFuzzTest::testFuzz_DaysAccrued(uint256) (gas: 9536295 → 9908510 |
↪  372215 3.903%)
--------------------------------------------------------------------------------
Total tests: 1261, ↑ 2, ↓ 236, ━ 1023
Overall gas change: 26145 (0.001%)
```

3.3.3 Unreachable `yieldMultiplier` initialization

Severity: Gas Optimization

Context: TreasuryDelegate.sol#L193-L196, TreasuryDelegate.sol#L499-L502

Description: The referenced code is redundant because `_accrueInterest()` will do the initialization:

```
uint256 userMultiplier = yieldMultiplier[user];
if (userMultiplier == 0) userMultiplier = WAD;
//...
if (yieldMultiplier[user] != userMultiplier) {
   yieldMultiplier[user] = userMultiplier;
}
```

Recommendation: Remove the referenced lines, maybe convert to a dev comment that initialization should have been executed in `_accrueInterest()`.

3.3.4 Early return can be inclusive of equality case

Severity: Gas Optimization

Context: TreasuryDelegate.sol#L942-L944, TreasuryDelegate.sol#L1011-L1013

Description: The equality case where `redeemValue == totalWithheldCredits[user]` & `targetDollars == totalWithheldCredits[user]` can be included in the early return.

Recommendation:

```
-    if (redeemValue < totalWithheldCredits[user]) {
+    if (redeemValue <= totalWithheldCredits[user]) {
       return redeemValue;
   }

-    if (targetDollars < totalWithheldCredits[user]) {
+    if (targetDollars <= totalWithheldCredits[user]) {
```

3.4 Informational

3.4.1 `WithheldCreditsDeque.toArray()` lacks unchecked wrapper

Severity: Informational

Context: WithheldCreditsDeque.sol#L189-L192

Description: The queue indexes can wrap around: specifically, with `pushFront()`, the start index is decremented and with `pushBack()`, the end index is incremented. Hence, `toArray()` would revert with addition overflow in any scenario where both `pushFront()` and `pushBack()` are invoked.

Proof of Concept:

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.13;

import {Test, stdError} from "forge-std/Test.sol";
import {WithheldCreditsDeque} from
 ↪  "src/upgrades/ProtocolStablecoinDelegates/WithheldCreditsDeque.sol";

contract WitheldCreditsDequeTest is Test {
    using WithheldCreditsDeque for WithheldCreditsDeque.Deque;

    WithheldCreditsDeque.Deque private deque;

    /// forge-config: default.allow_internal_expect_revert = true
    function test_toArrayFailsAfterPushFront() external {
        WithheldCreditsDeque.WithheldCredits memory withheldCredit =
         ↪  WithheldCreditsDeque.WithheldCredits({
            tokenAddress: address(0),
            credits: 100,
            releaseTimestamp: block.timestamp + 1 days,
            earningStartTimestamp: block.timestamp + 1 days,
            isRWA: true
        });
        // pushFront is to decrement front index to type(uint128.max)
        deque.pushFront(withheldCredit);
        // pushBack is to avoid decrementing front index, and have i = 1 for overflow
        deque.pushBack(withheldCredit);

        vm.expectRevert(stdError.arithmeticError);
        deque.toArray();
    }
}
```

Recommendation:

```
- result[i] = deque._data[deque._begin + uint128(i)];
+ unchecked { result[i] = deque._data[deque._begin + uint128(i)]; }
```

3.4.2 Simplify withheld-credits consumption in _getYieldAmount and _getAmountForTargetValue functions

Severity: Informational

Context: TreasuryDelegate.sol#L949-L961, TreasuryDelegate.sol#L1027-L1044

Description: In `TreasuryDelegate`, two view functions simulate LIFO consumption of withheld credits by iterating over the deque. In both cases, when execution reaches the loop, the caller has already ensured that the requested amount is **at least** the total withheld credits, so the loop always consumes **all** withheld credits.

1. `_getYieldAmount`.
   - Early return when `redeemValue < totalWithheldCredits[user]` (returns `redeemValue`).
   - So when the withheld loop runs, `redeemValue >= totalWithheldCredits[user]`.

- The loop consumes withheld credits LIFO until `remaining` is 0; it always consumes all withheld, so after the loop: `totalValue = totalWithheldCredits[user]`, `remaining = redeemValue - totalWithheldCredits[user]`.

2. `_getAmountForTargetValue`.

- Early return when `targetDollars < totalWithheldCredits[user]` (returns `targetDollars`).
- So when the withheld loop runs, `targetDollars >= totalWithheldCredits[user]`.
- The loop consumes withheld credits LIFO until `remainingValue` is 0; it always consumes all withheld, so after the loop: `totalCredits = totalWithheldCredits[user]`, `remainingValue = targetDollars - totalWithheldCredits[user]`.

In both functions, the loop is therefore redundant: the outcome is a fixed assignment. Replacing the loop with that assignment preserves behavior and saves gas.

Recommendation: In `_getYieldAmount`: Replace the withheld-credits loop with:

```
remaining = redeemValue - totalWithheldCredits[user];
totalValue = totalWithheldCredits[user];
```

In `_getAmountForTargetValue`: Replace the withheld-credits loop with:

```
totalCredits = totalWithheldCredits[user];
remainingValue = targetDollars - totalWithheldCredits[user];
```

3.4.3 MAX_AGE is not configurable in ChroniclePriceAdapter and ChainlinkPriceAdapter

Severity: Informational

Context: *(No context files were provided by the reviewer)*

Description:

- **ChainlinkPriceAdapter:** `MAX_AGE` is a constant (`1 days`), so it cannot be set to another value at deployment or changed later.
- **ChroniclePriceAdapter:** `MAX_AGE` is set in the constructor and stored as `immutable`, so it is fixed at deployment and cannot be updated afterward.

Recommendation: Support configuring max age at deployment and updating it later: (1) add a constructor parameter (and, for Chainlink, replace the constant with a stored value) so max age can be set per deployment, and (2) add an admin-only setter (e.g. `setMaxAge()`) so max age can be changed when needed without redeploying.

3.4.4 Break out of loop if daily rate not posted

Severity: Informational

Context: TreasuryDelegate.sol#L648

Description: If the daily rate hasn't been posted, the for-loop can be broken out because daily rates need to be posted in sequential order, so subsequent days will not have rates posted too.

Recommendation:

```
  if (!ratePosted[day]) {
     // Skip days without posted rates (weekends until backfilled)
-    continue;
+    break;
  }
```

3.4.5 Sanity check `band != 0`

Severity: Informational

Context: MultiliquidSwap.sol#L1129, MultiliquidSwap.sol#L1160

Description: band `== 0` is used as an existence check, should revert for 0 amounts as well.

Recommendation:

```
- if (band >= WAD) revert InvalidRate();
+ if (band == 0 || band >= WAD) revert InvalidRate();
```

3.4.6 TreasuryDelegate initializer does not validate initial rate against management fee

Severity: Informational

Context: TreasuryDelegate.sol#L170

Description: In `postDailyRate()` and `correctDailyRate()`, gross rate is required to be at least the management fee, and the stored rate is the net rate (`grossRate - managementFee`). In `initialize()`, the first day's rate is stored with no check against `managementFee`, and `managementFee` is not set there (it remains 0). So the initializer does not enforce the same "gross rate >= management fee" rule, and the first day's rate is not derived as a net rate after management fee.

Recommendation: If the first day's rate should follow the same semantics as later rates: (1) add an `initialManagementFee` parameter to set `managementFee` in the initializer, and (2) require `initialRate >= managementFee` and store `dailyRates[currentDay] = initialRate - managementFee`, so the same validation and net-rate logic apply from day one.

3.4.7 Whitelist checks can be more robust

Severity: Informational

Context: JTRSYWhitelistAdapter.sol#L30-L31, ULTRAWhitelistAdapter.sol#L23-L27

Description and Recommendation: For the JTRSY and ULTRA whitelist checks,

- JTRSY can use the token's own transfer restriction check

```
function checkTransferRestriction(address from, address to, uint256 value) public view
↪  returns (bool) {
   return detectTransferRestriction(from, to, value) == SUCCESS_CODE_ID;
}

function _onTransfer(address from, address to, uint256 value) internal {
        require(
            hook == address(0)
                || IHook(hook).onERC20Transfer(from, to, value,
                ↪  HookData(hookDataOf(from), hookDataOf(to)))
                    == IHook.onERC20Transfer.selector,
            "Tranche/restrictions-failed"
        );
}
```

- ULTRA should also check the `from` field

```
function _beforeTokenTransfer(
    address from,
    address to,
    uint256 amount
) internal override {
    super._beforeTokenTransfer(from, to, amount);
    if (to != _clawbackAdmin) {
        if (address(KYCContract) != address(0)) {
            require(
                KYCContract.isKYC(from) &&
                    KYCContract.isKYC(to),
                "Ultra: both sender and receiver need to be KYC approved"
            );
        }
```

```
    }
}
```

Multiliquid: Fixed in `f903d82` and `0604235`.

Cantina: Fixed.