Doma Hybrid Auction Protocol

DHAP 1.0

DI Networks

Sep. 2025

A comprehensive auction ecosystem for domain NFTs featuring two specialized systems: Hybrid Batch Auctions for portfolios and Premium Single Domain Auctions with sophisticated betting mechanisms.

System 1: Hybrid Batch Auctions for Domain Portfolios

Problem:
Traditional auctions handle domains individually, making it hard for large holders to liquidate portfolios and excluding small buyers.

Solution:

- Group multiple domains into one Dutch auction curve.
- Buyers can:
    - Bid for **full bundles**, or
    - Commit to **fractions** (e.g., 1% of bundle).

Example:

- Portfolio of 100 domains.
- Dutch curve: starts at 1,000 USDC → ticks down to 700 USDC reserve.
- Buyers commit fractions:
    - Alice 10%
    - Bob 5%
    - Carol 40%
    - Dana 50%
- At **780 USDC**, cumulative demand ≥ 100% → bundle clears.

✅ **Benefits:**

- Liquidity for big sellers.
- Smaller buyers access premium bundles fractionally.
- Higher transaction volume & participation.

Gamified Dutch Auctions with Bidder Rewards

Problem:
In standard Dutch auctions, bidders wait until the price drops → low early engagement.

Solution (Auction Mining):

- **Soft bids = Intent + Auto-convert threshold + Bond.**
    - Example: "Buy if price ≤ 900, size = 10%, bond = 0.5%."
    - Auto-converts to a hard bid when price hits threshold.
- **Hard bids = Binding purchase.**
- **Rewards = Sale-gated** → only minted if auction clears.

Reward formula:
Time-weighted score × Price-distance multiplier × Stake multiplier

Example:

- Alice (threshold 900, 10%) auto-converts earliest → earns highest score.
- Bob (860, 5%) converts later → earns medium score.
- Carol (820, 40%) adds significant demand → good score.
- Dana (780, 50%) clears auction.

Auction clears at **780 USDC**, bundle sold.

- Loyalty rewards distributed from seller rebate (e.g., 1% of sale).
- Alice gets most points, even though she didn't "win."

❌ If auction **fails to clear**, bonds refunded, **no rewards minted**.

✅ **Benefits:**

- Encourages early engagement, prevents point farming.
- Builds community loyalty (NFT badges, fee discounts).
- Stops "last-minute sniping."

Reverse Dutch Auctions for Royalties

Problem:
Static royalties don't adapt to urgency. Sellers either undercharge or scare away buyers.

Solution:

- Royalties **start at 0%** and **increase each block** until buyer accepts.
- Buyers face trade-off: wait for lower price but pay higher royalty.

Example:

- NFT domain starts at 1,000 USDC, 0% royalties.
- Price drops to 900 → royalties now 2%.
- Drops to 850 → royalties 4%.
- If buyer waits too long, royalties outweigh price savings.

✅ **Benefits:**

- Dynamic royalty capture.
- Creates urgency.
- Aligns protocol incentives with seller and community.

System 2: Premium Single-Domain Auctions + Betting System

Problem:
Batch auctions serve portfolios well, but premium single domains need focused attention and additional engagement mechanisms.

Solution (Separate Contract System):

- **Independent Single-Domain Auctions**: Dedicated Dutch auctions for premium domains
- **Commit–Reveal Betting**: Parallel betting system on auction outcomes
- **Complete Separation**: No dependency on batch auction system

4-Tier Price Betting Mechanism:

- **Price Categories**: Above High (3), High~Low Range (2), Below Low (1), Uncleared (0)
- **Seller Sets Thresholds**: High price and low price boundaries for betting
- **Commit Phase**: Bettors submit hash(choice, amount, secret) with stablecoin stakes
- **Reveal Phase**: Bettors reveal their bets after auction closes
- **Anti-Spam**: Unrevealed bets are redistributed to winners

- **Fair Odds**: Hidden commitments prevent manipulation/sniping

Pool Distribution:

- **90%** → Winning Bettors (pro-rata by stake)
- **5%** → Seller (liquidity premium)
- **3%** → Winning Buyer (price discovery bonus)
- **2%** → Protocol Treasury

Example:

- Create single domain auction: premium.doma (High: 80 ETH, Low: 60 ETH)
- Betting pool: 10,000 USDC total across 4 categories
- 30% bet "Above High" (>80 ETH), 40% bet "High~Low" (60-80 ETH)
- 20% bet "Below Low" (<60 ETH), 10% bet "Uncleared"
- Someone bids at 75 ETH → auction clears in High~Low range (category 2)
- High~Low bettors win: 9,000 USDC (90% of total pool)
- Seller gets: 500 USDC bonus, Buyer gets: 300 USDC bonus

✅ **Benefits:**

- **Independence**: Separate system for different use cases
- **Fairness**: Hidden commitments prevent market manipulation
- **Incentives**: All participants rewarded for market activity
- **Engagement**: Creates yield opportunities around premium domains

---

Dual Architecture System

Batch Auction Flow (HybridDutchAuction):

1. **Seller** lists domain portfolio
2. **Batch Auction Contract**:
    - Dutch price curve
    - Portfolio fractionalization
    - Soft/hard bid engine with bonds

    - Reverse royalty tracker
    - Reward engine (points/NFTs)
3. **Buyers** place fractional bids
4. **Settlement**: Clears when demand ≥ 100%

Single Domain + Betting Flow (DomainAuctionBetting):

1. **Seller** lists premium single domain with high/low price thresholds
2. **Single Auction Contract**:
    - Dutch price curve
    - First bid wins immediately
    - 4-tier parallel betting system
3. **Buyers** bid directly, **Bettors** wager on final price category
4. **Settlement**: Auction + betting resolved independently based on price ranges

---

Protocol Economics & Market Efficiency

Participation Amplification

Traditional Domain Auctions:

- Single-domain, single-bidder model
- Winner-takes-all dynamics
- Limited engagement beyond direct buyers
- High barriers for small participants

Dual Protocol Advantages:

- **10x Participation**: Batch fractionalization enables small buyers to participate in premium portfolios
- **Continuous Engagement**: Soft bids create ongoing market activity vs. last-minute sniping
- **Betting Multiplier**: Each premium auction generates 2 markets (direct bidding + price betting)
- **Loyalty Stickiness**: Gamified rewards create repeat participants vs. one-time buyers

Transaction Volume Growth

Volume Drivers:

- **Batch Efficiency**: 100 domains → 1 auction (vs. 100 separate auctions)
- **Fractional Access**: $1M portfolio accessible to $10K buyers (10% stakes)
- **Betting Layer**: Premium domains generate additional betting transaction volume
- **Reward Claiming**: Loyalty point distributions create secondary transaction flow

Conservative Estimates:

- **3-5x** transaction volume from batch consolidation
- **2-3x** unique participants from fractional access
- **1.5-2x** total volume from betting layer on premium domains

Fee Revenue Optimization

Revenue Streams:

1. **Auction Fees**: Standard platform fees on clearing prices
2. **Betting Pool Fees**: 2% protocol cut from all betting pools
3. **Loyalty Rewards**: Seller-funded rewards create fee-generating activity
4. **Reverse Royalties**: Dynamic royalty capture on secondary sales

Fee Efficiency:

- **Batch Consolidation**: Collect fees on larger transaction sizes
- **Betting Premiums**: Additional revenue without diluting core auction fees
- **Engagement Fees**: Loyalty activities generate micro-transaction fees

Information Asymmetry Reduction

Seller Benefits:

- **Price Discovery**: Soft bids reveal demand curves before clearing
- **Liquidity Assurance**: Batch auctions aggregate demand for better clearing rates
- **Fair Valuation**: Betting markets provide independent price validation

- **Reduced Timing Risk**: Dutch curves eliminate guessing optimal auction timing

Buyer Benefits:

- **Transparent Bidding**: Soft bid thresholds visible, reducing strategic uncertainty
- **Fractional Access**: Participate in premium portfolios without full capital commitment
- **Betting Intelligence**: Price betting provides market sentiment data
- **Loyalty Rewards**: Early participation rewarded vs. penalized

Market Efficiency:

- **Reduced Spreads**: Batch auctions narrow bid-ask spreads through aggregation
- **Better Price Discovery**: Multiple bidding mechanisms reveal true market value
- **Lower Transaction Costs**: Batch processing reduces per-domain transaction overhead
- **Increased Liquidity**: Fractional ownership creates deeper, more liquid markets

Network Effects

Participation Flywheel:

1. **More Sellers** → Larger batch auctions → Better fractional opportunities
2. **More Buyers** → Higher clearing rates → More seller participation
3. **More Betting** → Better price discovery → More accurate valuations
4. **More Rewards** → Stickier participants → Higher lifetime value

Result: Self-reinforcing ecosystem where each participant type benefits from growth in others

---

Benefits for Doma

- 🚀 **Liquidity boost**: Batch + fractionalization increase volumes.
- 🎮 **Engagement loop**: Rewards + gamification bring sticky bidders.

- ⏱️ **Dynamic urgency**: Reverse royalties ensure fast decision-making.
- 🔗 **Ecosystem fit**: Rewards tied to Protocol's NFTs, analytics.

🔑 **Takeaway**

This **Dual Auction Protocol** provides two specialized systems:

🎯 **System 1 - Hybrid Batch Auctions:**

- **Portfolio Trading** for scale and liquidity
- **Gamified Rewards** for community engagement
- **Reverse Royalties** for trading urgency
- **Fractional Ownership** for accessibility

🏆 **System 2 - Premium Single Domain + Betting:**

- **Premium Domain Focus** for high-value assets
- **4-Tier Price Betting** for sophisticated wagering
- **Commit-Reveal Mechanism** for fair betting
- **Independent Operation** for specialized use cases

👉 Result: **Complete domain trading ecosystem - from bulk portfolio liquidation to premium single-domain auctions with advanced betting mechanisms**

Contract Architecture

Core Contracts

System 1 - Hybrid Batch Auctions:

- HybridDutchAuction.sol - Batch auction logic with gamification
- LoyaltyNFT.sol - Gamification rewards and loyalty points

System 2 - Premium Domain + Betting:

- DomainAuctionBetting.sol - Independent single-domain auctions with 4-tier betting

Shared:

- IOwnershipToken.sol - Interface for Doma domain NFTs

Key Functions

System 1: Hybrid Batch Auction Functions

```
function createBatchAuction(
    IOwnershipToken nftContract,
    uint256[] memory tokenIds,
    uint256 startPrice,
    uint256 reservePrice,
    uint256 priceDecrement,
    uint256 duration,
    uint256 rewardBudgetBps,
    uint256 royaltyIncrement,
    address paymentToken
) external returns (uint256)
```

```
function placeSoftBid(uint256 auctionId, uint256 threshold, uint256 desiredCount) external payable

function placeHardBid(uint256 auctionId, uint256 desiredCount) external payable

function processConversions(uint256 auctionId) external
```

System 2: Premium Single Domain + Betting Functions

// Create single domain auction with betting price thresholds

function createSingleDomainAuction(uint256 tokenId, uint256 startPrice, uint256 reservePrice, uint256 priceDecrement, uint256 duration, uint256 highPrice, uint256 lowPrice) external

// Place bid on single domain (ends auction immediately)

function placeBid(uint256 auctionId) external payable

// Create betting pool with 4 price categories

function createBettingPool(uint256 auctionId, uint256 commitDuration, uint256 revealDuration) external

// Commit bet with hash of (choice, amount, secret)

function commitBet(uint256 auctionId, bytes32 commitHash, uint256 amount) external

// Reveal committed bet (choice: 3=Above High, 2=High~Low, 1=Below Low, 0=Uncleared)

function revealBet(uint256 auctionId, uint8 choice, uint256 amount, uint256 secret) external

// Settle betting after auction ends

function settleBetting(uint256 auctionId) external

// Owner functions

function setCuts(uint256 _sellerCut, uint256 _buyerCut, uint256 _protocolCut, uint256 _winnerCut) external onlyOwner

Examples

Example 1: Hybrid Batch Portfolio Auction with Gamification

Setup:

- Item: 100-domain bundle
- Start price: 1,000 USDC (Dutch, linearly down)
- Reserve floor: 700 USDC
- Reward budget: 1% of final sale, only if cleared
- Bond: 0.2% of intended spend

Early Phase:

- Alice: soft bid for 10% of bundle, threshold = 900 → bond posted
- Bob: soft bid 5%, threshold = 860 → bond posted
- Carol: soft bid 40%, threshold = 820 → bond posted
- Dana: soft bid 50%, threshold = 780 → bond posted

Price Progression:

- At 900: Alice auto-converts (10%). Cumulative = 10% — continue
- At 860: Bob auto-converts (5%). Cumulative = 15% — continue
- At 820: Carol auto-converts (40%). Cumulative = 55% — continue
- At 780: Dana auto-converts (50%). Cumulative = 105% ≥ 100% → auction clears at 780

Settlement:

- Pro-rata fill at clearing price (if over-subscribed)
- Bonds returned
- Rewards minted (since sale cleared):
    - Alice (earliest, highest price distance) gets largest share of points
    - Dana gets less (later threshold), even though she cleared the auction

```
// Create batch auction
createBatchAuction(
  ownershipToken,
  [1,2,3,...,100],  // 100 domain token IDs
```

```
    1000e18,        // 1000 USDC start price
    700e18,         // 700 USDC reserve
    1e18,           // 1 USDC per block decrement
    300,            // 300 blocks duration
    100,            // 1% reward budget (100 bps)
    0,              // No reverse royalty
    address(0)      // ETH payments
);


// Alice places early soft bid
placeSoftBid{value: 1.8e18}(auctionId, 900e18, 10); // 10 tokens at 900, bond = 1.8 USDC
```

Example 2: Premium Domain Auction with 4-Tier Betting

Setup:

- Single premium domain with price range betting
- Bettors wager on final price category
- 4 betting tiers: Above High, High~Low Range, Below Low, Uncleared

```
// Create single domain auction with betting thresholds
createSingleDomainAuction(tokenId, 100e18, 50e18, 0.5e18, 3600, 80e18, 60e18);
// highPrice = 80 ETH, lowPrice = 60 ETH


// Create betting pool
createBettingPool(auctionId, 3600, 1800); // 1hr commit, 30min reveal


// Commit bets (hidden)
bytes32 hash1 = keccak256(abi.encodePacked(uint8(3), 100e18, 12345)); // bet >80 ETH
```

bytes32 hash2 = keccak256(abi.encodePacked(uint8(2), 50e18, 67890)); // bet 60-80 ETH

commitBet(auctionId, hash1, 100e18);

commitBet(auctionId, hash2, 50e18);

// Someone bids on auction

placeBid{value: 75e18}(auctionId); // Auction clears at 75 ETH (category 2)

// Reveal after auction ends

revealBet(auctionId, 3, 100e18, 12345); // Wrong prediction

revealBet(auctionId, 2, 50e18, 67890); // Correct prediction (60-80 ETH range)

// Settle betting

settleBetting(auctionId); // Category 2 bettors win 90% of pool

Betting Categories:

- **Category 3**: Final price > High Price (above 80 ETH)
- **Category 2**: Low Price ≤ Final price ≤ High Price (60-80 ETH)
- **Category 1**: Final price < Low Price (below 60 ETH)
- **Category 0**: Auction fails to clear (no sale)

Contract Addresses (Deployed and Verified on Doma Testnet)

System 1 - Hybrid Batch Auctions:

- **HybridDutchAuction.sol**:

  0xE680A0F580f742a536B33C142757b4C8BE5CfB40

- **LoyaltyNFT.sol**:

  0x04B36cADFD85F2561c0e8A676E0aCe5cBA8c7485

System 2 - Premium Single Domain + Betting:

- **DomainAuctionBetting.sol**:

  0x5A11663fc4cBfa62E01C3bbCfDb10f37549B38D2

Resources

Websites:

https://doma-auction-frontend.vercel.app

Documentation:

https://github.com/0xawang/dhap/protocol-documentation.pdf

Open Source Repository:

Smart Contracts: https://github.com/0xawang/DomaAuction

UI / Frontend: https://github.com/0xawang/doma-auction-frontend