

NEVER Phase 2 (Auctions)

Submission by Radiance Team

Table of contents

[Introduction](#)

[Radiance Team](#)

[Overview of the solution](#)

[System Requirements](#)

[Test case details](#)

[EVER => NEVER Auctions Algorithms](#)

[NEVER => EVER Auctions Algorithms](#)

[Common technical requirements](#)

[Hard criteria](#)

[Deliverables](#)

[Future Roadmap](#)

Introduction

Live demo: <https://never.defispace.com>

Gitlab repository: <https://git.defispace.com/never>

License: Free software (Apache 2.0)

Contact: <https://t.me/UltraNihilist>

Wallet: 0:e989bda452d7276ec42cce90307dc3ed1d828205c304776322c8df2e2350780d

Radiance Team

General manager: Dmitry Summit

Team and product manager: Anzor Daurov

Lead developer: Yaroslav A.

Developer: Anzor B.

Developer: Maxim Z.

Developer: Ramazan C.

Developer: Alexey K.

Overview of the solution

Solution provides implementation of the Vickrey auctions for NEVER stable coins in exchange of native EVER cryptocurrency and for collective participation as well (D'Auction). The auctions base on the rules from chapter IV. NOT The Auction from [\[Defi#14\]](#). The auctions for NEVER tokens initiate upon price reveal as implemented by [\[Defi#24\]](#).

Oracle used for price update in the auction is fully implemented as well.

Detailed architecture can be seen in [GitLab repository readme](#).

System Requirements

Platform: MS Windows, Mac OS, Linux GUI or compatible

Browser: Google Chrome or compatible, desktop only

Wallet: [EVER Wallet Chrome Extension](#)

Network: Everscale test network (already deployed), Everscale mainnet (ready to deploy)

System clock should be synchronized with the world clock on the client device for the auction to work well. ([Win10 guide](#), [Mac guide](#))

Test case details

To test this implementation go to the live demo here: <https://never.defispace.com>

Install EVER Wallet Extension and use these seed phrases:

1. fall bright add apology random leopard rice witness stock where economy dust
2. sadness fall dune more actor kangaroo demand snake cousin acquire predict flight

To properly test the auction you will need at least two participants in the same auction.

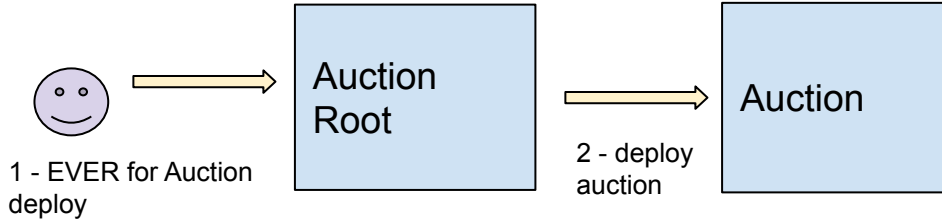
The wallets above contain both EVER and NEVER tokens for your convenience so that you can test both types of auctions.

D'Auction is currently only implemented only on smart-contract level, user interface will be provided in the next version, as well as debot interface to allow for use in EVER.Surf.

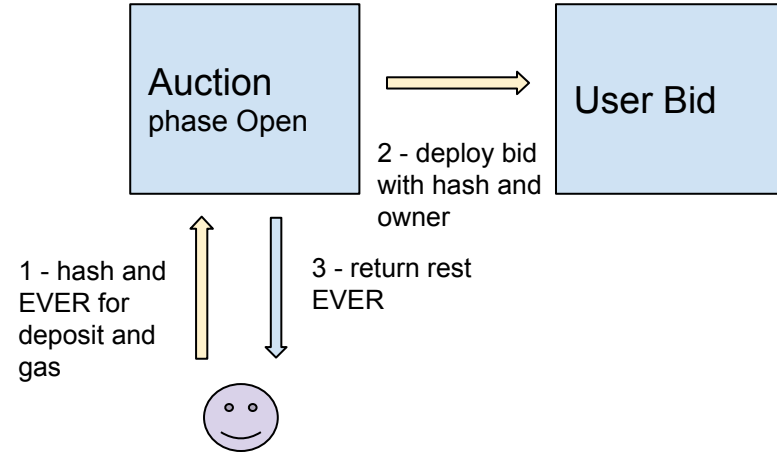
EVER => NEVER Auctions Algorithms

Creating Auctions and Making Bids

Create auction



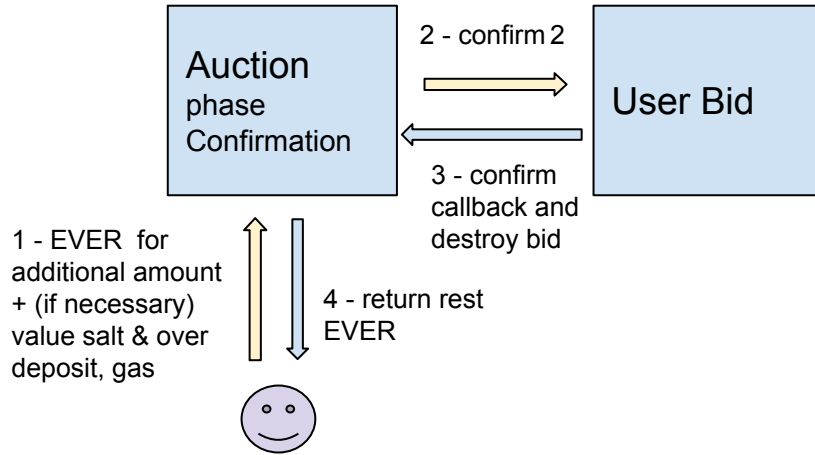
Make Bid



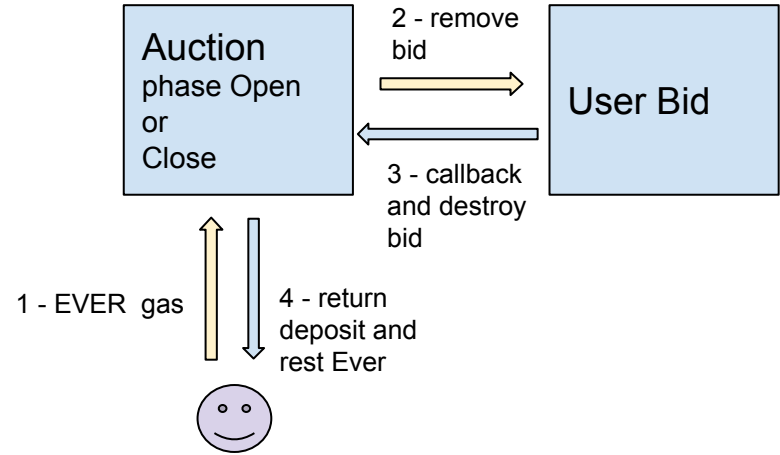
EVER => NEVER Auctions Algorithms

Confirming and Removing Bids

Confirm Bid



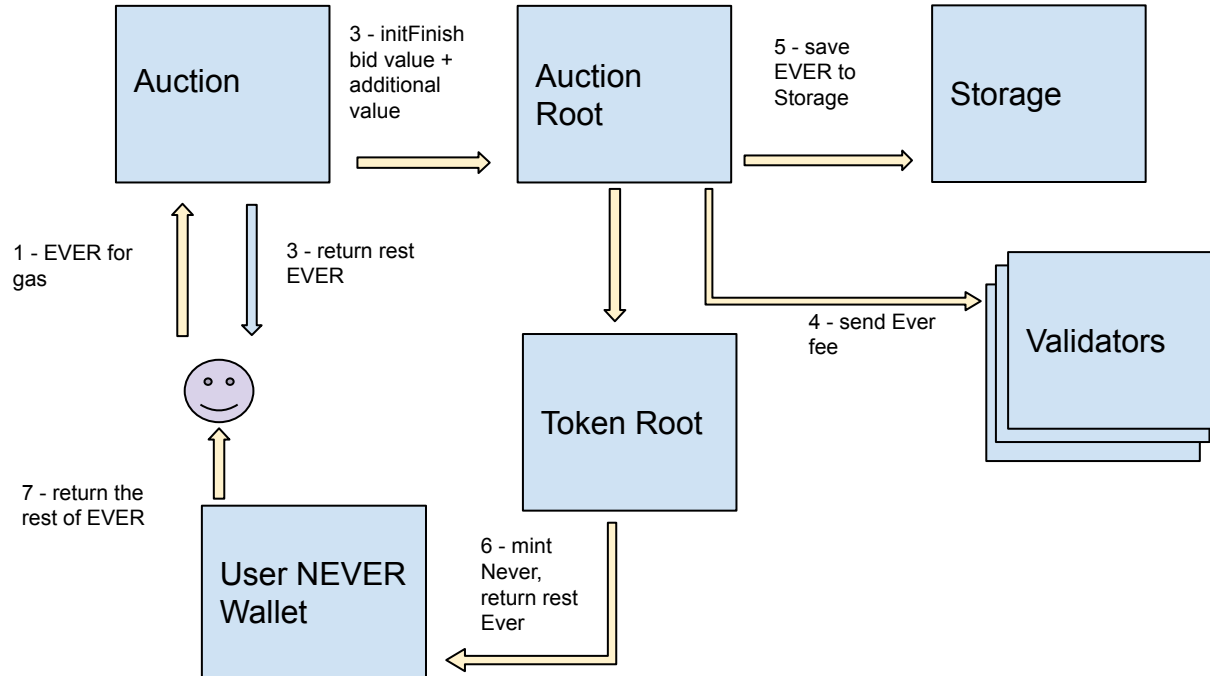
Remove Bid



EVER => NEVER Auctions Algorithms

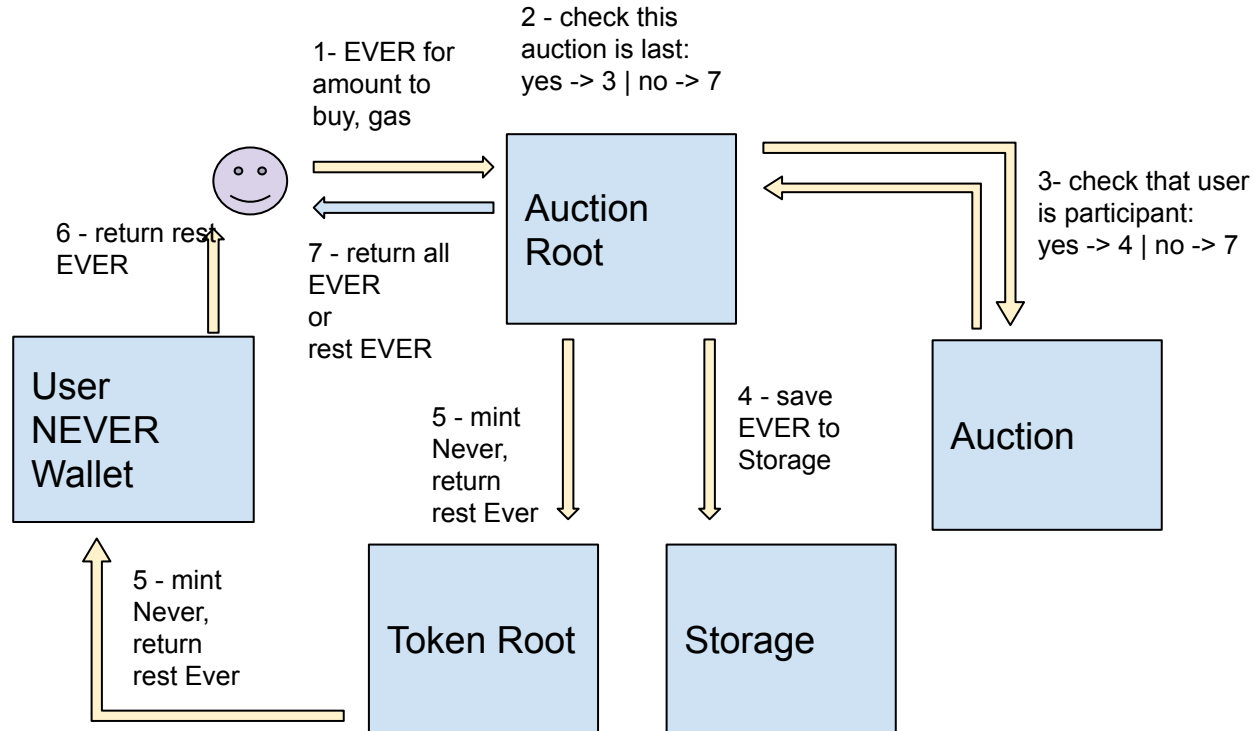
Updating Bids

2 - Update Phase. If phase = Close and
winner chosen: yes -> 2 | no ->5



EVER => NEVER Auctions Algorithms

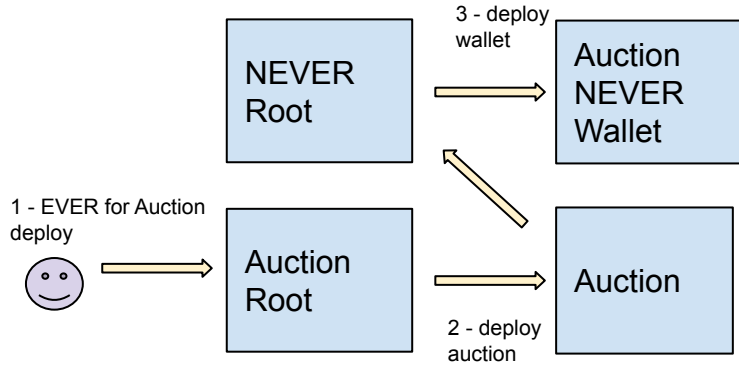
Buying NEVER after Auction Finish



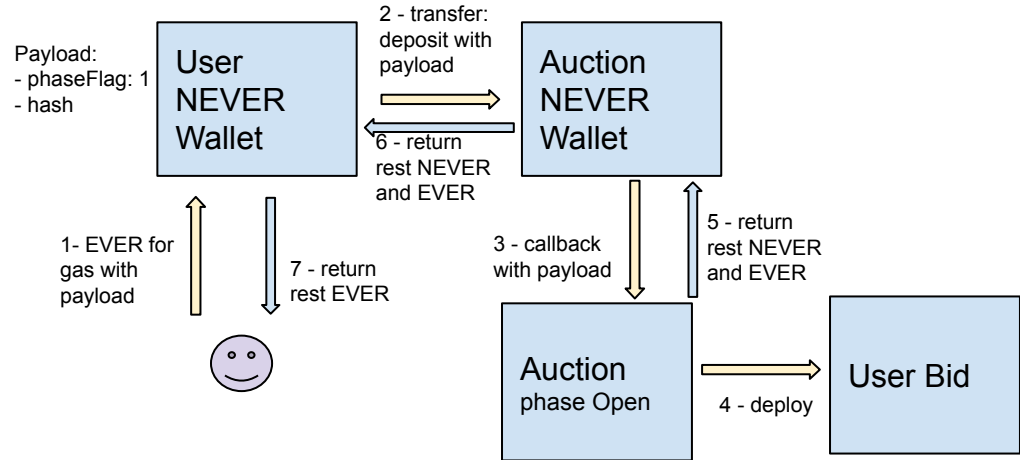
NEVER => EVER Auctions Algorithms

Creating Auctions and Making Bids

Create auction

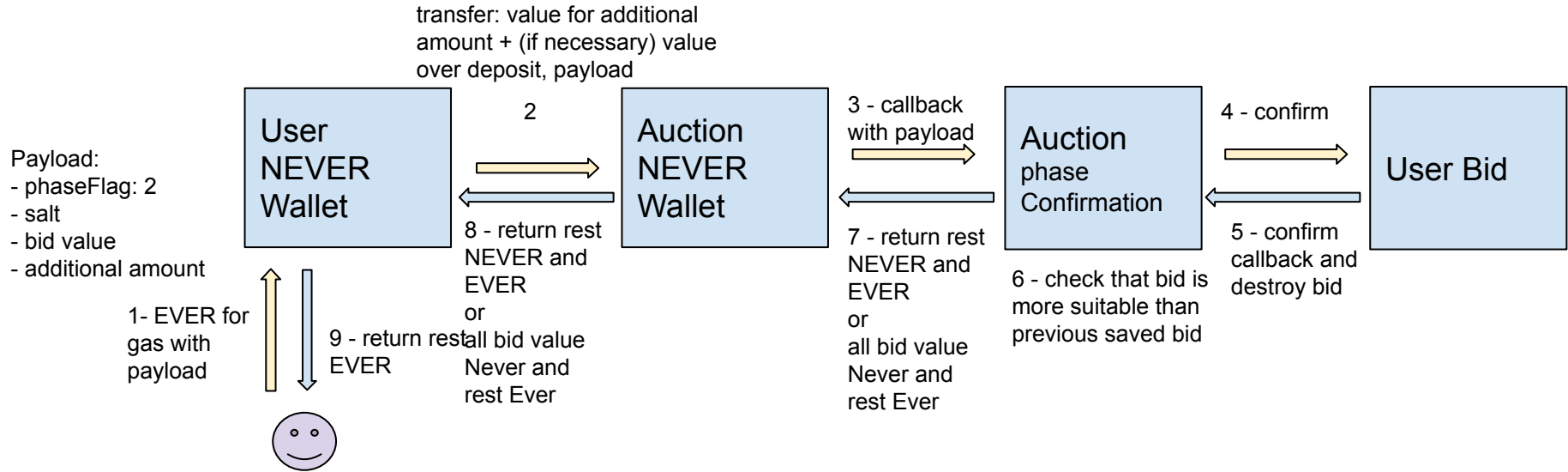


Make Bid



NEVER => EVER Auctions Algorithms

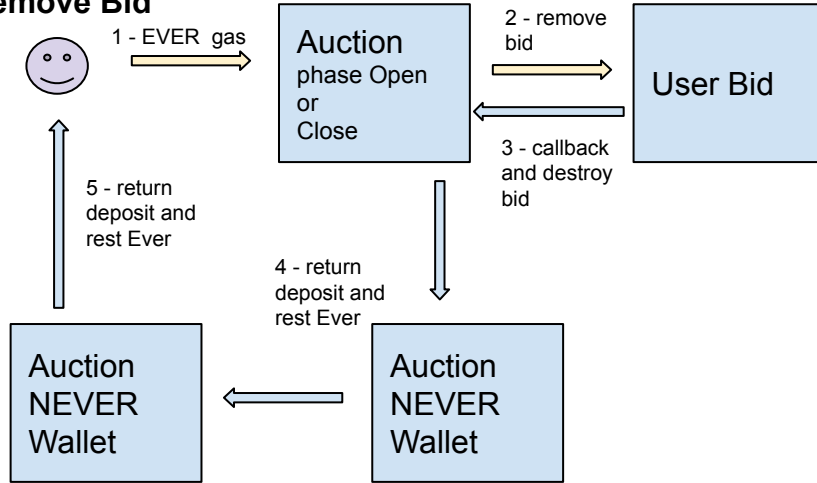
Confirming Bids



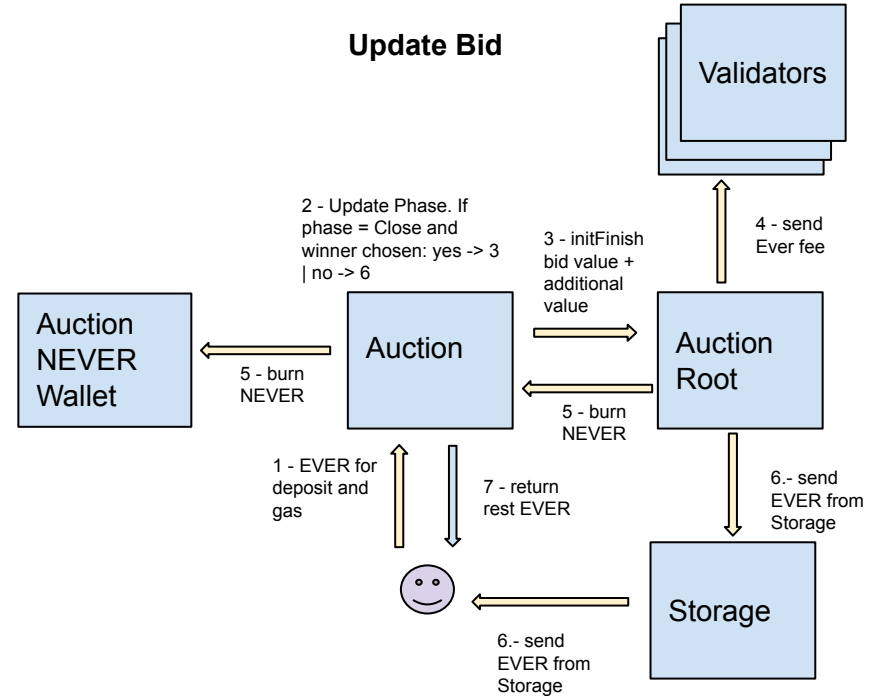
NEVER => EVER Auctions Algorithms

Removing Bids and Updating Bids

Remove Bid



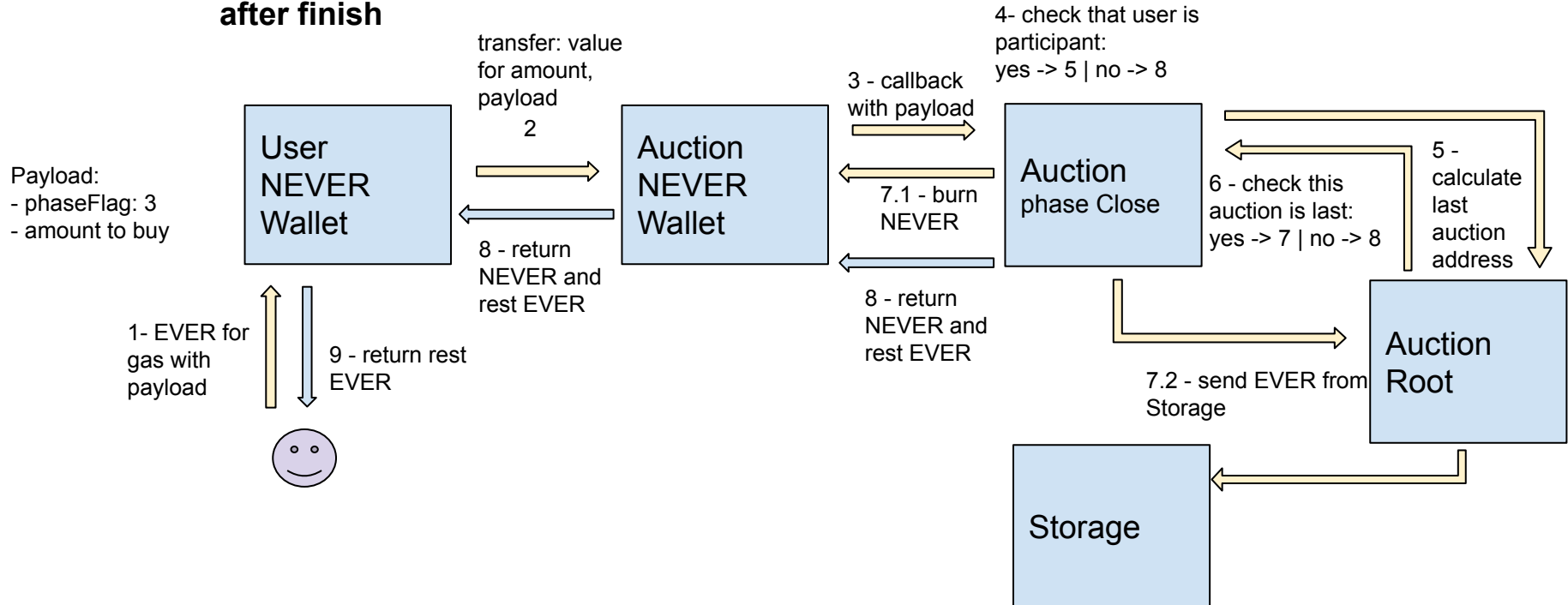
Update Bid



NEVER => EVER Auctions Algorithms

Buying EVER after Auction Finish

Buy Ever after finish



Common technical requirements

Requirement	Status
The solution MUST implement the auctions for NEVER tokens automatically initiated upon price reveal as implemented by Defi#24	Done
The auctions MUST fulfill the requirements mentioned in the corresponding section	Done
The D'Auctions MUST be implemented fulfilling the requirements mentioned in the corresponding section	Done
The NEVER Tokens MUST be referred via the interface (or set of interfaces) that MUST be applicable, at least, both for NEVER native tokens (Currency Collections) and for TIP-3 style NEVER tokens	Done (NEVER native tokens do not exist yet)
The evidence of workable solution MUST be provided: i) The solution MUST be deployed to any test network (local usage is not enough) ii) The demonstration of the workable solution MUST be provided (both at AMA session as well as at the personal environment of each juror) iii) The manual for verifying the solution MUST be provided and MUST provide extremely clear and detailed instruction iv) In case the implementation of NEVER token is required for the demonstration its stub implementation MUST be provided	Done https://git.defispace.com/never/auction/-/blob/main/README.md
All the contestants MUST provide their contact information as a Telegram ID	Done
All the participants are strongly encouraged to reuse the results of Defi#14 , otherwise they MUST provide a strong explanation and description of their approaches.	Done

Hard criteria

Requirement	Status
Implementation of Auction and D'Auction contracts (as well as auxiliary contracts) that satisfy to the requirements above	https://git.defispace.com/never/auction https://git.defispace.com/never/oracle
The contract system must be deployed to any testnet. The submitter must provide the clear instructions how to demonstrate it works as well as to demonstrate it at AMA session	AuctionRoot: 0:7a4214d1cc81c69895a6f733c2234baef73023022053a880e1b5713833be7b56 Storage: 0:3abd74face6b8e75f9ae322f2cd75b709327665577106f72725eed0fac65b1ea TokenRoot: 0:0cfc8d5b3b1203769db9088d3b85130ad03b50f1e102e3129abdfc125057039
All the elements to be developed at the later stages may be substituted with some dummy implementations. However, these dummies still must not prevent the system from demonstration its workability	The D'Auction functionality is implemented in contracts only, user interface will be provided in the next version
The submission must be accompanied with the full documentation that should contain: ○ Generic description of the solution ○ Technical description of the solution ○ Clear build and deployment instructions	https://git.defispace.com/never/auction/-/blob/main/README.md https://git.defispace.com/never/oracle/-/blob/main/README.md https://git.defispace.com/never/frontend/-/blob/master/README.md
All the configurable constants must be moved to a separate file and clearly described	https://git.defispace.com/never/auction/-/blob/main/test-s-devnet-draft/Constants.json
The configurable parameters describing D'Action strategy must be clearly described. The D'Action contract must provide all the getters to fully understand its strategy	https://git.defispace.com/never/auction/-/blob/main/test-s-devnet-draft/Constants.json

Deliverables

Requirement	Status
Source code of all the implemented smart contracts (published at github or any other widely recognized repository storage)	https://git.defispace.com/never/auction https://git.defispace.com/never/oracle
Documentation in PDF or MD format	https://git.defispace.com/never/auction/-/blob/main/README.md
Addresses of the contracts published in the testnet	AuctionRoot: 0:7a4214d1cc81c69895a6f733c2234baef73023022053a880e1b5713833be7b56 Storage: 0:3abd74face6b8e75f9ae322f2cd75b709327665577106f72725eed0fac65b1ea TokenRoot: 0:0cfc8d5b3b1203769db9088d3b85130ad03b50f1e102e3129abdffc125057039
Evaluation instructions	Test case provided in current document on page 7

Future Roadmap

- Implementation of D'Auction in the user interface (contracts are ready)
- Debot interfaces for Auction and D'Auction functionality to allow access using Ever.Surf

Thank you for your attention!

For any questions contact <https://t.me/UltraNihilist>