Chronicle

Chronicle Protocol is a novel Oracle solution that overcomes the current limitations of transferring data on-chain by developing scalable, cost-efficient, decentralized, and verifiable Oracles, rewriting the rulebook on data transparency and accessibility.

Querying the price of ARB using Chronicle

Chronicle contracts are read-protected by a whitelist, meaning you won't be able to read them on-chain without your address being added to the whitelist. On the Testnet, users can add themselves to the whitelist through the SelfKisser contract; a process playfully referred to as "kissing" themselves. To access production Oracles on the Mainnet, please open a support ticket inDiscord in the 🆘 | support channel.

For the deployment addresses, please check out theDashboard .

// SPDX-License-Identifier: MIT pragma

solidity

^ 0.8.16 ;

/* * @title OracleReader * @notice A simple contract to read from Chronicle oracles * @dev To see the full repository, visit https://github.com/chronicleprotocol/OracleReader-Example. * @dev Addresses in this contract are hardcoded for the Arbitrum Sepolia testnet. * For other supported networks, check the https://chroniclelabs.org/dashboard/oracles. / contract

OracleReader

{ /* * @notice The Chronicle oracle to read from. * Chronicle_ARB_USD_1 - 0xdD7c06561689c73f0A67F2179e273cCF45EFc964 * Network: Arbitrum Sepolia /

IChronicle public chronicle =

IChronicle (address (0xdD7c06561689c73f0A67F2179e273cCF45EFc964)) ;

/* * @notice The SelfKisser granting access to Chronicle oracles. * SelfKisser_1:0xc0fe3a070Bc98b4a45d735A52a1AFDd134E0283f * Network: Arbitrum Sepolia / ISelfKisser public selfKisser =

ISelfKisser (address (0xc0fe3a070Bc98b4a45d735A52a1AFDd134E0283f)) ;

constructor ( )

{ // Note to add address(this) to chronicle oracle's whitelist. // This allows the contract to read from the chronicle oracle. selfKisser . selfKiss (address (chronicle)) ; }

/* * @notice Function to read the latest data from the Chronicle oracle. * @return val The current value returned by the oracle. * @return age The timestamp of the last update from the oracle. / function

read ( )

external

view

returns

(uint256 val ,

uint256 age)

{ (val , age)

= chronicle . readWithAge ( ) ; } }

// Copied from chronicle-std. interface

IChronicle

{ /* * @notice Returns the oracle's current value. * @dev Reverts if no value set. * @return value The oracle's current value. / function

read ( )

external

view

returns

(uint256 value) ;

/* * @notice Returns the oracle's current value and its age. * @dev Reverts if no value set. * @return value The oracle's current value using 18 decimals places. * @return age The value's age as a Unix Timestamp . * / function

readWithAge ( )

external

view

returns

(uint256 value ,

uint256 age) ; }

// Copied from self-kisser. interface

ISelfKisser

{ /// @notice Kisses caller on oracle oracle. function

selfKiss (address oracle)

external ; }

More examples

For more examples of integrating Chronicle Oracles, please check the documentation portal . Edit this page Last updatedonJan 27, 2025 Previous Chainlink Next ORA