Quick start

0 to 1 integration with the Optimistic Oracle V3. The primary integration point into the UMA ecosystem is the Optimistic Oracle V3 (OOV3). The OOV3 is an oracle for arbitrary off-chain data

which leverages an interactive escalation game betweenasserters anddisputers and is secured byeconomic incentives . It differs from the Optimistic Oracle V2 (getting started can be foundhere) by being easier to reason about and simpler in integration points.

This getting started tutorial will show you how to go from 0 to 1 with the OOV3 by executing the simplest possible assertion flow. Later in the docs you can find more information on how the OOV3 works and dig deeper into its mechanism and more sophisticated code examples.

A minimum viable OOV3 integration

The OOV3 works by making a truth claim about the world, stating that something has happened or is true. Once asserted, the assertion enters a challenge period wherein someone can disagree with the assertion, by disputing it. If no one disputes it during the challenge window the statement is taken as true. If disputed, the outcome is arbitrated using the UMA DVM (more info on how this works here).

In this tutorial you will be working through a simple smart contract that asserts the following truth:Statement: Argentina won the 2022 Fifa world cup in Qatar.

Once through the challenge window, you will use the assertion in your resolution contract. This shows the full non-dispute lifecycle of an OOV3 data assertion. It will give you the basic intuition as to how the Optimistic Asserter works without much overhead and is a great starting point before digging deeper. Let's get started!

Prerequisites

To complete this tutorial you will need:

1. Metamask installed in a Chromium based browser (such asGoogle Chrome
2.) If you don't have it already, you can get Metamaskhere
3. .
4. A wallet set up in Metamask.
5. Görli
6. test ETH to send test transactions. You can get GETH fromAlchemy's faucet
7. .
8.

Asserting a truth

First, we will work through the basic flow forasserting a truth to the oracle . In this example, we are making a statement about the outcome of a sports event, but the statement could be much more complex; it could power any kind of smart contract system requiring data.

The contract used in this tutorial is meant to be a simple data assertion flow. The contract exposes a simpleassertTruth function which asserts the truth to the OA about the outcome of the world cup.

1. Go to this example contract on Remix
2. . This gives you the minimum assertion flow. Read the contract and the comments listed within.
3. In the far left hand menu, click the link to deploy and run transactions (which looks like the Ethereum logo and a right arrow).
4. In the "Environment" dropdown, choose "Injected Provider," and connect to your wallet.Make sure you are connected to Görli within your metamask wallet, that has Görli test ETH!
5. You don't want to deploy to a real network and spend real gas, and the Görli Optimistic Oracle address hardcoded into the contract will not work on other networks.
6. Under the "Contract" dropdown, selectOOV3_GettingStarted
7. .
8. Click "Deploy" and confirm the transaction with your wallet.
9. You should now see theOOV3_GettingStarted
10. contract under "Deployed Contracts". Clicking the dropdown carrot will reveal buttons for each of the functions in the contract.
11. ClickassertTruth
12. to request the data specified in the contract's ancillary data, asserting the truth about the outcome of the Fifa world cup. This will submit a data assertion to the Optimistic Asserter and the assertion will enter the challenge period.
13.

What we've done in the above steps is: a) deploy a smart contract and b) submit a "data assertion" to the Optimistic Oracle

V3 through the call to the Optimistic Oracle V3'sassertTruthWithDefaults function. Now, the assertion has entered the challenge window and can be disputed by someone who disagrees with the assertion. If you click thegetAssertion you can see some of the fields that are associated with the assertion. For more details on what these fields mean you can look at the OOV3 interfacehere .

The default liveness for an assertion is 120 seconds (2 minutes). Wait this time before going to the next step.

Settling the assertion

Once the assertion has settled, and assuming no one has disputed it, we can settle it! Do the following:

1. ClicksettleAndGetAssertionResult
2. in remix. This will send a transaction to settle the assertion and return the settlement value.
3. Wait for the transaction to mine.
4. Once mined you can expand the transaction output block inline with the green tick arrow. If you scroll down here you can find some information about the assertion, such as the settlement value (which should betrue
5. as it was not disputed).
6.

?

1. Now, you can callgetAssertion
2. andgetAssertionResult
3. from remix and see the outputs.getAssertion
4. has changed such thesettled
5. value istrue
6. andgetAssertionResult
7. now returnstrue
8. as the assertion was deemed valid as it was not disputed.
9.

Next Steps

Hopefully you got a basic understanding of the Optimistic Oracle V3 request flow from this getting started guide. Note that this kind of example would not really work on mainnet as: a) there were no bonds for the asserter (and therefore no rewards for disputers) b) the challenge window was too short. The Optimistic Oracle V3 works due to economic incentives between the asserter and disputers, which was absent in this example to keep things simple. For some incentive compatible examples check out some of the example tutorials where we walk through more details on the functions discussed in this guide. Those can be foundhere .

Previous UMA Protocol Next Optimistic Oracle Last updated2 months ago On this page *A minimum viable OOV3 integration * Prerequisites * Asserting a truth * Settling the assertion * Next Steps

Was this helpful? Edit on GitHub