Setting Custom Bond and Liveness Parameters

Overview of Setting Customer Bond and Liveness Parameter in UMA's Optimistic Oracle Every request to UMA's Optimistic Oracle includes bond and liveness settings that specify the size of the bond that proposers (and disputers) are required to post, the token used for bonding, and the liveness window, which is the challenge period during which a proposal can be challenged.

The minimum bond is the same asfinal fee for Optimistic Oracle V2 (OOV2) or can be queried by calling getMinimumBond(token) for Optimistic Oracle V3 (OOV3). The default liveness window is two hours. For many cases, you may want to customize these values. The primary reason to increase the bond size or challenge window is to increase your security for complex requests or requests that could move large amounts of money.

How to think about proposer and disputer bonds

In most cases, you will want to set a bond higher than the minimum. The reason for this is that the bond provides a financial incentive for disputers to dispute invalid proposals. Disputers receive half of the proposer bond in the case of a successful dispute in the OOV3. For the OOV2, it's half of the excess above the final fee.

The part of the bond that doesn't go to the disputer in a successful dispute goes to the UMAStore contract. The reason the disputer does not get the full amount is to prevent a malicious proposer from making a bad proposal and then front-running an honest disputer to dispute themselves if they get caught.

Example bond payouts (OOV3)

As an example, imagine an application that requires a 10,000 USDC bond to assert a claim via OOV3.

To make an assertion, a proposer must post a 10,000 USDC bond, which they will get back after the liveness window if the assertion is not disputed. To make a dispute, a disputer must also post a 10,000 USDC bond.

If the disputer wins, they receive 15,000 USDC (10,000 USDC disputer bond + 5,000 USDC from the excess proposer bond) and theStore receives 5,000 USDC.

If the proposer wins, they receive 15,000 USDC (10,000 USDC proposer bond + 5,000 USDC from the disputer bond) and theStore receives 5,000 USDC (half of the proposer bond).

Higher or lower bonds?

If data received from the oracle could potentially move large amounts of value, you may want to set a higher bond to make it more costly for an attacker to attempt to steal funds through a false proposal. (Remember that the proposer will lose their entire bond if they are disputed and found to be incorrect.)

The tradeoff to setting a higher bond is that it may be more difficult to run a project that requires frequent proposals, since good proposers will have funds locked up for at least the length of the challenge window. Another tradeoff is that a very high bond may make it difficult for disputers to post bonds to challenge bad proposals. Remember also that disputers will lose their bond if they are incorrect, so the disputer has a capital cost and a risk of loss.

Before launching your contract, you should think about who you expect to be proposers and disputers, whether they have adequate access to capital, how much value the oracle is securing, how many requests you expect to make in a given day, and whether you should add a proposer reward (to incentivize independent proposers to take on the capital cost and risk of proposing answers to your requests).

There are no exact answers to how high or low your bond should be, although it is almost always a good idea to set a bond that is larger than the final fee so that disputers have an incentive to dispute bad proposals.

How to think about liveness

The liveness period is the challenge window where disputers can dispute a proposal, and is another knob you can turn to increase security (while accepting UX trade-offs).

A typical challenge period is two hours, which is usually plenty of time for disputers to recognize and dispute bad proposals. In some cases, you may want to increase the length of the challenge period. A classic example is an insurance contract, since insurance contracts usually have these qualities:

1. Large amounts of value are at stake.
2. Payouts are rare.
3. Proposals and disputes require a greater amount of thought and consideration.
4. Payouts are not very time sensitive, and a long delay (even a day or more) to allow proposers and disputers to think through the situation will not seriously inconvenience users.
5.

You may want to consider a two-hour challenge window in these situations:

1. Requests happen frequently and proposers and disputers will need to reuse their capital regularly for bonding.
2. The dollar value secured by a request is lower.
3. The request is easy to reason about (and proposals and disputes could potentially be automated).
4. Users are sensitive to the settlement time.
5.

At this time, it is generally not recommended to set a challenge window shorter than two hours.

If your users are very sensitive to settlement time, you may want to consider a smart contract architecture that allows a particular type of user to fully insure a piece of data that is acted on instantly, and then waiting to be reimbursed after the challenge window. Think of this user as a "designated waiter," who is probably compensated in some way.

A good example is therelayer in the Across protocol. Users of the Across bridge receive their funds almost instantly, because a relayer sends them funds on the receiving chain and then submits for reimbursement from the protocol, and has to wait for several hours for their relay to be bundled with other relays, proposed, and then settled after the challenge window.

In this case, the UX for most users is excellent (nearly instant transfers) but the protocol is still secured by a long challenge window that is absorbed by a less time sensitive user type (the relayer) who is paid fees for their trouble.

Previous Monitoring Bot Setup Next How does UMA's Oracle work? Last updated2 months ago On this page *How to think about proposer and disputer bonds * How to think about liveness