Execution model: two-party exchange

Intro

In this document we consider some examples of the execution model of a two-party exchange. We assume a simple exchange between Alice and Bob, where:

has

wants

Alice

[1] note of token T

blue dolphin NFT

Bob

blue dolphin NFT

[1] note of token T

Even though some mechanisms might be redundant for such a simple case, we will use it in all of the examples below for simplicity.

1. All parties use intents

This is the most general case that can handle both complex state transitions with high level of uncertainty and simple state transitions. Here every party just uses intents to express their interests.

[

3619×1515 470 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/78880bbd0b966c847c6396f2db417117fa2abe89.jpeg)

Step-by-step description

- Alice creates an intent (step 1) and produces a partial transaction (step 2), spending her note of value 1 and creating an intent note with the value base derived from the intent she specified.
- Bob writes his intent down (step 1) and uses it to derive the value base for the intent note he creates in his partial transaction (step 2).
- Alice and Bob send their partial transactions to the intent gossip network.
- A solver receives both intents and matches them together, creating a third partial transaction (step 3) that sends the blue dolphin NFT to Alice, spending Alice's intent note, and sends the note of value 1 to Bob, spending Bob's intent note.

Ptx description

spent notes

created notes

VP proofs

total balance (accumulated)

ptx #1

(Alice)

[1] of token T

[Alice intent note]

Alice intentVP, intent App VP, Alice T userVP, T VP (4)

-[1] + [Alice intent note]

ptx #2

(Bob)

[blue dolphin NFT]

[Bob intent note]

Bob intent VP, intent App VP, Bob NFT userVP, NFT VP (4)

-[1] - [blue dolphin NFT] + [Alice intent note] + [Bob intent note]

ptx #3

(Solver)

[Alice intent note], [Bob intent note]

[blue dolphin NFT], [1]

Alice NFT user VP, Bob user T VP, NFT VP, T VP, Alice intent VP, Bob intent VP, intent App VP (7)

0

2. A party knows what exactly they want

The mechanism described above is general and suitable for all kinds of state transitions. However, in the cases where a party knows exactly what they want, the workflow can be simplified.

Intents can handle situations when a user have multiple options in mind and it isn't clear in advance what they will spend and receive in the final transaction. If the user knows exactly what they want to spend and receive, there is no uncertainty about the final transaction, and the use of intents is redundant.

Let's assume that Bob knows exactly what he wants[1]

. Instead of creating an intent expressing his preferences, he creates the note he wants to receive directly in his partial transaction (step 2.2). Now Alice only needs to spend this note to balance the transaction.

[

3621×1515 888 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/a8adcf3bf37d7dd52bbaa0748962ef7437c71eab.png)

Ptx description

spent notes

created notes

VP proofs

total balance (accumulated)

ptx #1

(Alice)

[1] of token T

[Alice intent note]

Alice intentVP, intent App VP, Alice T userVP, T VP(4)

-[1] + [Alice intent note]

ptx #2

(Bob)

[blue dolphin NFT]

[1] of token T

Bob T userVP, T VP, Bob NFT userVP, NFT VP (4)

- [blue dolphin NFT] + [Alice intent note]

ptx #3

(Solver)

[Alice intent note]

[blue dolphin NFT]

Alice NFT userVP, NFT VP, Alice intent VP, intent App VP (4)

0

2.1 All parties know what they want

This case isn’t conceptually different from the one above, but just for completeness we add it here too.

When both Alice and Bob know what they want, they don’t need to use intents and can just create the notes they want for themselves. The only task left is to match these partial transactions together into the final transaction.

[

3273×1514 630 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/d711ab9b5ffaab26335e84cd1ff2dc4ee2f63815.png)

Ptx description

spent notes

created notes

VP proofs

total balance (accumulated)

ptx #1

(Alice)

[1] of token T

[blue dolphin NFT]

Alice NFT userVP, NFT VP, Alice T userVP, T VP(4)

-[1] + [blue dolphin NFT]

ptx #2

(Bob)

[blue dolphin NFT]

[1] of token T

Bob T userVP, T VP, Bob NFT userVP, NFT VP (4)

0

3. One of the parties is the solver

When all parties keep their intents secret (in a broad, non-cryptographical sense), it isn't possible to match them as nobody knows who wants what. But once one party reveals their intents, the other parties can see if their intents match with the revealed one.

Solvers can take advantage of the fact that they see intents of other parties and avoid using intents themselves. They simply match with an intent that satisfies their needs (that they keep in mind instead of creating an intent).

This situation is almost exactly the same as case #2 when one of the parties just knows what they want.

[

3621×1515 847 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/6f69ca749b450527fb32588d603aa65821989e63.png)

The difference from case #2 is that Bob (the solver) technically doesn't know what exactly he wants in advance and only creates his partial transaction after he sees Alice's intent. But the rest is the same: Bob doesn't create an intent and just spends the asset that Alice wants to receive and receives the asset that Alice has spent.

In fact, Bob doesn't even have to create a separate partial transaction for himself and can put it all into the solver's ptx 3.1:

[

3621×1515 841 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/9295c0b6ca8baf65b9ec669ddf8854fd9ba16e71.png)

Note, though, that two parties can't take advantage of being the solver at the same time as there is just one solver at a time.

Proofs per ptx

spent notes

created notes

VP proofs

total balance (accumulated)

ptx #1

(Alice)

[1] of token T

[Alice intent note]

Alice intentVP, intent App VP, Alice userVP, T VP(4)

-[1] + [Alice intent note]

ptx #2

(Bob, the solver)

[blue dolphin NFT], [Alice intent note]

[1] of token T, [blue dolphin NFT]

Bob T userVP, T VP, Bob NFT userVP, Alice NFT user VP, NFT VP, Alice intent VP, intent App VP (7)

0

Comparison

Table below shows the amount of VP proofs per partial transaction in each case:

Case

ptx 1 VP proofs

ptx 2 VP proofs

ptx 3 VP proofs

1 (all intents)

4

4

7 [2]

2 (1 intent)

4

4

4

2.1 (no intents)

4

4

-

3 (1 intent)

4

7 [3]

-

Naturally, the use of intents implies having more notes and more VP proofs.

1. Well, we actually assume that in all of the examples for simplicity, but in this case it is a required condition

↩

1. save one check on intent App VP↩
2. save one check on NFT VP↩