

The goal of this post is to start a centralised discussion on the definition of intents and to provide the necessary context for it.

## Definitions

### Intents

Users initiate a state transition on the blockchain by submitting their intent. Intents

are representations of user preferences that shape the initiated state transition. They are invariant across all possible state transitions, meaning that no matter what transaction gets constructed, the user's intent will be satisfied by it. Solvers

are the actors who receive user intents and match them together, constructing transactions.

[

image

1473×633 31.8 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/f1671ddf0d20a0a2d6dbef5950aeeddc8d05c5e4.png)

### Partial transactions & transactions

Partial transactions and transactions are data structures that solvers construct to satisfy users' intents. They are designed in the resource model

, in which state transitions are represented as consumed (input) and created (output) resources

- atomic data units.

A partial transaction

contains  $m$

input and  $m$

output resources and represent a partial state transition. Such state transition cannot be published on the blockchain itself, but is used to construct transactions. A transaction

$tx_i$

contains  $n_i$

input and  $n_i$

output resources ( $n_i = k_i * m$ ,  $k_i \in \mathbb{N}$ )

) and is built from  $k_i$

partial transactions.

[

image

1229×538 22.4 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/a02b31f110bdcc3f7569e59bb3a6a17debe1c938.png)

The reason why a partial transaction cannot be published on the blockchain is that they are not balanced

.

### Balance

Each resource has a resource type

and an unsigned numeric value called quantity

. When a resource is being consumed, its quantity is assumed to be negative, and when a resource is being created, the quantity is assumed to be positive. A transaction balance

is the sum of the resource quantities per each resource type. A transaction is called balanced

when for each resource type, the balance of the transaction equals to zero<sup>[1]</sup>

.

[

image

1229×600 45.7 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/dcfa66eca752587986b84b6190c35f44bfa614a7.png)

## Are intents partial transactions?

Well, yes and no. Intents can

be represented as partial transactions, but are not equivalent

to them. Let's take a closer look.

### I. Intents =

partial transactions

When the user knows exactly what they want, they simply produce a partial transaction that creates the asset they want. Now, this partial transaction is not balanced, and the solver's goal is to find another partial transaction that is willing to give the asset away (which is expressed by consuming the asset).

[

image

1158×462 22.5 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/f337ecd3e233f6295ff730e5d8b79520e4d753cc.png)

### II. Intents $\neq$

partial transactions

#### Resource logic

Every resource has a predicate encoded into it that expresses the rules on which the resource can be consumed and created. This predicate is called a resource logic

. A resource logic

has a hierarchical structure - it might require checking other predicates in order to be satisfied.

#### Intents =

predicates

In the case when a user doesn't know exactly what they want<sup>[2]</sup>

, they can encode all possible options that would satisfy them in a predicate

. The resource logic then would require this predicate to be satisfied.

To allow that kind of behaviour a special type of resources is used<sup>[3]</sup>

, a logic of which states that this resource can be consumed iff the intent predicate is satisfied. The user would create a resource of that type at the beginning of interaction and consume it at the end. Having this resource consumed in the transaction proves that the intent was satisfied.

In this situation an intent is encoded in a predicate, that is itself is encoded in a resource, and resources are consumed and created in partial transactions. So, the intent is carried

by a partial transaction but is not equal to it.

## So...what are intents?

In the first section I described intents abstractly, and in the next section described two different (complementary) ways to express them in the Anoma model. To start the discussion, I propose to use a variation of the definition from the first section as the definition

:

Intents

are representations of user preferences that shape the initiated state transition. They are invariant across all possible state transitions, meaning that no matter what transaction gets constructed, the user's intent will be satisfied by it.

1. the transaction balance is checked against the balancing value
- . For transactions fully within the shielded pool the value is 0.
1. it can also be used when the user knows what they want, but using just partial transactions might be cheaper in this case
2. it isn't actually special
3. this overview is not exhaustive, and there are other ways to express intents