Hey everyone!

I have a few questions about Anoma’s handling / thinking around data dependencies for intents. I’m still not 100% sure how they would fit into your current architecture.

Let’s start with an example of an intent with a data dependency presented in natural language: “Swap ETH to receive at least 200 USDC and send to Apriori.eth if he was able to successfully buy the NFT conference ticket for both of us

”.

1. Would data dependencies just be part of a Predicate, as the resource can’t be consumed until the specified condition is satisfied?

2. Based on the whitepaper and the “temperature in Berlin” example, it sounds like solvers are expected to provide data. How would solvers submit additional data, for example a required proof of computation or a proof of satisfied conditionals that are fulfilled with private data?

This might be necessary in some cases, as view of state is subjective and solvers might have a significantly diverging view, e.g. through private order flow or access to off-chain liquidity. Not all conditionals are purely end state related, as they can be totally independent from the desired end state + constraints.

1. Does the Path Authentication design also allow someone to just solve for the data dependency, i.e. a “data solver”? For the sake of simplicity, let’s just assume the existence of one trusted data solver only; no consensus / proof generation on the data side or solver auction for now. The data solver could solve for the conditional (“was Apriori able to successfully buy the NFT conference ticket for the both of us?”) and then hand it over to “execution” solvers (“swap ETH to receive at least 200 USDC”). The “how” relates back to the second question.

We believe that the assumption of solvers providing various types of data is overly optimistic. Competing on the execution side is already hard enough, and will only get harder.

The solver market needs to be accessible and sufficiently competitive. While solvers CAN build up additional data infrastructure to gain a competitive advantage, we can’t solely rely on it and need to find a balance between allowing for a solver’s capitalization on this competitive advantage and reliable resolution of data dependencies. That’s exactly what we are looking into right now.

I would like to hear everyone’s thoughts on whether this line of thinking makes sense in general and understand more about how it could fit into Anoma’s architecture. Lastly, is this something you are working on yourself or is it out of scope for you?