Notes from a recent twitter spaces with the Taiko community featuringAnton Bukov from 1inch, Chen Magen from Cow Swap, Kenny Bassig from Taiko, and Christopher Goes from Heliax. Lisa A from Taiko organized and hosted the spaces. Thank you to Lisa A from Taiko for coordinating.

Note that the text herein is only a loose transcript of the conversation and I may have omitted important scentences that provide better context. Errors are my own. Listen to the audio for a complete synthesis.

Ken (Taiko)

: Ecosystem and BD/partnerships. Here to discuss What Taiko is doing to enable intent based architectures to support builders and dapps building awesome applications and use cases.

Anton (1inch)

: Here to discuss intents, it will be a very interesting discussion. What is actually happening in the market. 1 inch for almost 1 year has 1 inch fusion which allows users to trade assets by providing signatures. Modern way to call this is an intent. Working internally on intents protocols. Many companies are working on their own vision of intents protocol because no one clearly understands what it is, and every developer has their own opinions. I think we will see maybe 10 intents protocols and the market will decide what is suitable for what.

Chen (Cow Swap)

: On the Cow Swap team for the last 3 years. Mainly focused on BD and ecosystem and a little bit about Cow Swap the aspect of intents. Cow protocol in the existing version is running for more than 2 years mainly the team is focused on auction mechanisms to allow not just for users to express trading intents but also for solvers or sophisticated players in the system to use those intents to maximize user benefits. One unique way this happens on cow protocol is via batching multiple user intents which can be settled in a simple batch.

Christopher (Anoma)

: Anoma has been working on intents for a little while and arrived at the concept from trying to understand what blockchains will look like, what users want, and generalize and turn it into a mathematical expression. The word has become popular recently. The way we use it means heterogeneous trust and information flow control. I am excited to explore different definitions and ways our protocols can interoperate and collaborate.

Lisa (Taiko)

: What is an intent and how does it differ from a transaction, Will intents replace transactions or are they complimentary?

Christopher

: Sometimes this question is framed as intents vs. transactions, I see them as a spectrum - intents constrain but they dont specify exactly what state changes are valid.

There is a lot of stuff in the middle and much of the stuff in the middle, do we call it an intent or a transaction? I'm excited about intents as a paradigm insofar as intents as a conceptual and mathematical concept they want to specify constraints and not an exact sequence. For example, I want more tokens rather than less. Speaking on intents replacing transactions is not the way I think about it. The kind of intents users express over time will become more flexible and more general- some will look more like transactions today and some will not, but as long as they interoperate.

There have to be deterministic state changes. There will be some transactions at the end but users may not be dictating those directly.

Anton

: Let me add shortly that I consider intent protocols as a separation of intentions and settlements. Originally users executed transactions on their own and specified all of the exact steps of what should happen inside those transactions. In other words users will define what they want but not how they want to achieve it.

Kenny

: Intents is an architecture for the next age in terms of simplifying UI and UX. What an intent is sort of like your travel agency. I want to go to Europe, you only have $2000, which is setting constraints. Instead of you paying the travel agency more, there is a benefit. This is a good example to keep as an analogy for intents. If we are thinking about a specific comparison, intent-based architectures are kind of like account abstraction (AA), sort of abstracting the complexity of transactions. Instead of having to think about how to sign things, you have actual people who do it for you in the sense of travel agents called solvers

. This is applied to many facets and use cases. Solvers or Market Makers execute transaction for you.

Lisa

: It’s all very nice if the intent can be executed, but how do intents work under the hood? What is the intent lifecycle? How is the preference matching mechanism realized?

Anton

: This is definitely an interesting question, more about what are the features of intents and what should they have and not have. I have some ideas that are not final; I think that intent protocols should be re-entrancy friendly. Solvers should be able to execute multiple intents at once and it’s very likely it should be done nested

. If you look at simplest intents which have existed for many years, it’s basically limit orders; users give permission for an arbitrary executor to take one of their assets and get another one back at some specific rate or variable rate.

I would like to see if intents can be referenced. It should be possible to reference intents in other intents, you could make intents which depends on another intent… a user signs an intent but has no ability to pay for this execution. The user could pay in a wallet via apple pay and the wallet could issue an intent to pay for execution of the intent, and wallets can work without being a solver. I am afraid we still need to take care of partial fills of intents. This is what should be in an intents protocol. Im not sure if these can be avoided.

Chen

: The concept of limit orders obviously existed for a very long time. The most common intents today are just limit orders. This is how the user expresses them; they describe an outcome and add a constraint. One aspect is who is getting the authorization of the user to fulfill the intent. The user must choose those entities quite carefully in order to make sure they optimize for the best outcome for the user. What we are trying to solve for in the past is users submitting those transactions and arbitrary entities could take those transactions and not care about the user outcome at all. So the mechanism that is being used to pick the solver that is going to execute the user intent is very important. This one of the main areas where we can still innovate. Its important that the user can rely on these mechanisms and use them to express their intent while being confident they will be executed at their best interest.

Lisa

: Can I ask you something else? Let’s assume there are several users who want to make some swaps and these swaps do not coincide. Could you tell us about the mechanism behind intent execution, how it is matched with the best option?

Chen

: In CoW protocol there is the notion of objective criteria, the way a given solution is being ranked is quite simple today. It sums up the price improvement of each order in the batch are getting over the minimum amount the user is willing to except. The sum represents the score of the batch. Overall the users in the batch should be happier. One tricky part is in some scenarios one user might be better off than another and still this solution might win. There are a lot of things to take into account when designing this kind of mechanism.

Christopher

: I think these descriptions are fantastic. We separate intents intents into constraints

and preferences

.

- constraint
- Similar to slippage limit on AMM, can avoid paying gas and off-load the execution risk
- preference
- things you prefer but don’t want to put into the constraint. These you can’t enforce in the intent exactly.

The other thing which will be important is users measuring solvers after the fact. I could split it up into different trades and doing a bunch of trades over time. Its in my interest to measure the performance of solvers over time and switch which solvers I send my intents to based on which ones are better for me.

Where you send your intents to is a significant decision. User A may want more ETH and User B may want more USDC. Not only that even if they split the difference, half to user one and half to user two, that is not sybil resistant. The importance of sybil resistance or a reputation system - there is a lot of fertile ground to explore and different protocol designs giving users an ability to semi-automate a way to express their intents with a smarter interface in accordance with users stated preferences.

Lisa

: How do you assign solvers? This is a very interesting topic. Let’s touch a bit UX. A specific part of UX is composability. Take cross-chain operations for example - how can this composability be achieved with intents, especially with non-

Ethereum chains like Solana who doesn't post their state root to Ethereum?

Anton

: Composability of intents for me means intents can be references to other intents in other protocols. I'm not sure intents can help with interchain or cross-chain composability. I believe its probably a problem that should be solved in a DeFi way with other protocols. Protocols can work with other protocols and reference their primitives. In hackathons you can build permissionlessly smart contracts on top of smart contracts.

Lisa

: Should we introduce intents to each domain and can aggregation mechanisms be possible?

Anton

: I hope people do not try to solve cross-chain with intents. They could easily be deployed on every EVM compatible chain, could be developed as a protocol for non EVM chains. These would work separately.

Christopher

: Protocol composability and atomic composability.

Protocol compatibility - users think about assets and security models. They want good enough security. They might not think about which rollup or chain in particular. For that user, it's helpful if these protocols could be compatible, expressed in a common format that is legible or executable. Protocol composability is pretty important, and it is a prerequisite for DeFi which is based on the message passing capabilities of the EVM

. I think we will see alternative approaches to the EVM like Taiga

.

Atomic composability is more like a database question, and there you are still limited by how consensus systems work. You need these two domains to be controlled by the same consensus set. If you want to do atomic state changes in one contract on main-chain and some code on cosmos or bitcoin it is almost impossible. You can lock state or transfer state and do an atomic operation

. Intents help a little bit and solvers can move assets in the background. As a user you specify where your assets are and where you want them to be. I think intents can be helpful there, but atomic composability is a distributed systems question.

Lisa

: If a user has an intent for changing ETH for USDC, currently the best way to do it is to take their ETH on Ethereum and swap it with USDC on Taiko. It will be up to solver to figure out how it will happen.

Christopher

: In the case I'm envisioning y

user has ETH and they want USDC on Coinbase or something which means they could execute the swap in many different places. Other rollups can be involved. Presumably the user wants best execution, the best price, and they want the operation to be done quickly and they are probably flexible and prefer the system does something smart. I might want the swap split up into this kind of automation an intent protocol at the user interface level, plus solvers and interoperability on all these rollups can automate this type of bookeeping.

Lisa

: Let's talk about intents, economics, and pricing. How can intents be priced? Will users pay per intent or preference execution? Will this pricing be lower or higher compared to transaction fees?

Anton

: For me it seems like the payment for execution should already be priced in the constraints - we call them permits

and aspects

, that is what the user wants to spend. Christopher called it constraints and preferences. Its about the same with different wording. I do not believe a user who creates their intents should care much about those who execute or fulfill their intentions. I believe these systems should be as permissionless as possible.

In 1inch fusion, there is a dutch auction and solvers compete with each other. The competition betweenresolvers lets users benefit from better execution. Trying to restrict who can fulfill those intents will bring more problems than benefits. I never saw when a ratings systems solved anything or simplified it. To keep everything simple, I would prefer an intents protocol to be permissionless as a protocol. If someone wants to make a private order from someone specific it should be made available.

Lisa

: To clarify a bit - what you mean here by permissionless and the fact that ratings overcomplicates the mechanism, if its not a ratings mechanism maybe you can tell us about how the dutch auction works and why it is efficient?

Anton

: If you place a limit order in the very ground case, the intents protocol should deliver the exact amount you expect because when you look at tokens you think that amount is something desired by users, they want this kind or more. This could be a specific kind of order to deliver more. Just imagine that there are systems of ERC-20 tokens with debts which have negative tokens. Euler has debt tokens. If you are trying to swap your debt and place it as an intent, you don't want to get more, you want to get less. It's the nature of what you are swapping.

If you are swapping NFT tokens you don't want to get anymore. No one will be happy if they get a higher token ID. Intents should take care of specific things inside the sub protocol. If you have limit order protocol built on top of intents protocol - how fusion works there is a dutch auction where the user defines a specific price which changes every second getting worse for users but improves for resolvers. When someone decides to fulfill those orders they fulfill the price that exists at this moment. Taking care of better execution, you should not restrict who can solve.

Christopher

: Not an opposite opinion, just one aspect worth bringing to the intents conversation is other things aside from swaps - making many more applications feasible because the execution costs are low or zero because the execution is not matched. For example, let's consider that you are a user and you are trying to discover a counterparty who is a car. In the current setup there is a company called Uber who builds the protocol, the database, and the engines. They operate solvers and settlement. They host on servers and run a counterparty discovery between users and drivers. They are basically an insurance agent - not always a good actor but they play this role to a certain extent.

There are a bunch of roles going on there. An intent-centric protocol may be able to decentralize and disaggregate these roles. Uber is also a ratings agency - for there could be an application where the ratings and insurance are operated by local parties with the local contexts that one centralized SF based company doesn't do a good job of doing. There needs to be a relatively easy way to send around intents for free. If users have to pay, no one is going to use this application. This subtle UX distinction makes many of these applications feasible or not.

Lisa

: This sounds extremely exciting.

Christopher

: We are thinking about this, but not doing anything with actual cars, yet.

Lisa

: Within the last several years we have talked about onboarding the next billion users. You partially already answered this question, but what are the use cases for intents outside of blockchain? How can intents help us bring new users into blockchain ecosystem, replace money for example, either in a hypothetical or realistic way, do you think this is possible?

Anton

: Probably SQL query is also an intent. That is a way how developers describe exactly what they want to get not how data is extracted from a database because they can't do it in an efficient way as it is super hard to find algorithms to extract data. This is why SQL queries are declarative. They don't specify how the data should be extracted.

Lisa

: Is there any reasonable reason to somehow make a transition from this current SQL request execution into blockchain where someone will benefit from it?

Anton

: Developers create intents that pay for queries or we could have intents for getting answers from AI models or something like that.

Chen

: I think one important part required to make it more appealing for many use cases is allowing the user to be able to describe what they want to achieve but also translate the intent into the different parts that can be optimized and allow users to get the best outcomes. If a user wants to buy an NFT the purchase of the NFT itself is a non-fungible position can't really be optimized. There is not a lot of ways to execute it.

This outcome that users are trying to achieve could be split into a swap of ERC 20 tokens - something fungible that could be achieved for the user, the best outcome for them. It’s not exactly an intent in the way we view it in terms of the protocol. This is something I wanted to add.

Kenny

: I’ve really never thought about it. Intents have been around for some time. Really, its just a way to describe an execution path across actions, breaking down or removing the how. In terms of real-world use cases, intent-based networking already exists. Its something within the space of AI. Machine learning can perform routine tasks that can verify goals and actions can be achieved. Intents outside of blockchains can exist.

Lisa

: Last question - very important question. Why should rollups care about intents and want to collaborate with intent companies or projects?

Kenny

: In terms of intents and scaling up the next wave of user transactions, its important that rollups focus on something like this. From a dApp perspective, its interesting in terms of use-cases - simplifying UI/UX to express intents in all kinds of verticals.

I think there will be more intent-based applications even things like GambleFi or bridging. In the future we can have natural language in terms of intents. In that sense intents offer a new kind of way to interact with dApps.

Anton

: I can add - I don’t see a clear reason why rollups themselves should care about intents. Probably rollup block producers should not care about this. They work on a different level to solve a different problem. If you think block producers on rollups should care about this, I’m not sure anyone will add this type of functionality. But If we will consider this as part of MEV activity probably they will do this. If yes, then Ethereum validators are taking care of MEV because its one important source of revenue for them so then rollup block producers should also take care of them.