Smarter Contract
Roll A Mate 27:18
AI Audit
Piggy
zbay

Smarter Contract

Speaker 10 16:58
(intro)
Hi everyone. We are smarter contracts. Yeah, supposedly. I'm Hu man, Taggy Shan and Mohammed, we are Waterloo student. So yeah, let's just.

Speaker 11 17:12
Hi there. So in terms of **what we did**, a smarter contract, all of us know that we have a smart contract, but they are not absolutely a smart, right? They are sometimes dumb.

So our goal is like, can we have a more smarter contract that could understand better things? So let's go and have an example. **For example**, some chains are better for computation and some chains are better for storage.

So the idea is can we divide the contract and call different functions on different chains to be more efficient. Like an astronaut on Mars that has limited resources and need to be superefficient.

In terms of what we build, we design our own universal deployer as the infrastructure, use the hyperlink to deploy the different instance of the contract on different chains. And then with the gas estimator, which is smart, actually here we can understand which function should be called on what chain before going to a deep dive and demo.

On what we build **in terms of the economy**. Each day, there is $10 million spending on gas. So with a simple calculation, we understood we can save around 30% with a smarter contract powered by Hyperchain, let's deep dive to product demo.

(demo)
…

Speaker 10 18:28
Yeah, thank you very much. So let's just get this thing going because it takes some time. I'm just gonna call it. So here's the thing. We wanted to test whether or not we're on the right path. So we talk to ourselves, what is the simplest form of calculation that we can start?

Addition, multiplication and subtraction. Of course, you cannot go more simpler than that. My wallet is connected. And yeah, basically right now I'm on Alfajores, which is the test net of cello. And I would like to multiply 9 by 5. Simplest form. And this is the terminal. So we are actually logging everything that we are receiving from the server. And I'm going to show you where is it. Yeah, so basically the gas estimator determine that the cheapest chain to do the calculation is Fuji. Fuji is the test net of avalanche, maybe. Yeah, so everything is being sent to Avalanche.

Speaker 10 19:33
Of course, we are deploying this smart contract on every supported chain right now, the hyper, the supported chains that Hyperlane has provided. And yeah, it's going to take some time. And in the meantime, we are going to have a technical.

(technical details - tools used!)

Speaker 12 19:53
Alright, thank you so much for your explanation. So here we're gonna **go a little under the hood** here. So we have a user who wants to call a function with some function ID and some arguments. **In order to enable the user to** find the cheapest chain, we have our SDK to estimate the gas, to estimate the cheapest chain. **And then**, we rely on the hyperlink to route our function call from the search chain to the destination chain. So here we have inside the destination chain an executor, which basically match a function ID to the actual implementation. So here we route a function code from a search chain to the destination chain. We call that inside the destination chain. And then we return to the source chain to have this state here. Thank you so much.

Speaker 10 20:50
Yeah, let's see the results. Hopefully it's there. Yeah, it's sending the acknowledgement. We can wait or I can show you the previous rounds result. So when it's going to finish, there is going to be a box here. I'm going to show you afterwards.

Speaker 10 21:09
We have two links, the hyperlink, explorer links. And yeah, as you can see, the first message is sent from my main chain, Alpha Horus to Fuji, which sends the function call and needed input data arcs. And the second one is from Fuji to Alpha Horus, which gets back the results. And here we have locked the results, which is like 45 da, of course. Yeah, so yeah, this one is finished too. basically the same links.

(Next step)

Speaker 10 21:41
So the **proof of concept** is here. **And what we are going to do for future is** to use DS dot proxy in order to enable any sort of smart contract to be deployed in this template that we have provided. So imagine in near future, smart contracts that are deployed on multiple chains and actually they are intelligent enough to know where to do each type of calculation, function, call or storage. So that's actually smarter contracts. Thank you very much.

Roll A Mate 27:18

Speaker 14 27:18
We are roll amatee its amatee you know the matte the drink OK its a roller im sorry. Sorry, it's a another problem. It's a payment sis. Yeah, payment protocol, eh, to make it, oh, sorry. Yes, ETA, you may need a main pool based rollouts. So send money with gas list to 4 cent a transaction cost using main net.

So we know that in the past day we have some complaints about Vitalic saying that he was in Argentina and he has to make a payment on Mainnett and he has to use a centralized exchange. So this causes a lot of conflicts and also is a lot of conflicts about this in the space about the cost of the transactions. So for $15 Bill, you have to expand a five dollar transactions on time. So, okay, eh, don't worry here we came to the rescue.

Speaker 14 28:12
So, eh, what happened until yesterday in Argentina, Venezuela, Nigeria, Turkey, El Salvador, many businesses are accepting crypto. So we call them crypto business. What they are doing now, they say, pay me on Binance. So there were traditions saying how we do this is 5 dollar transaction. No possible. Okay, but are the rules.

Speaker 14 28:34
So what happen now with Rola Matte? They say, pay me a minute. It's zero extra cost for you. And what we say, yes, sir. Beer for all the nuns. So gas list or force intersections on mainland. what yes hello manpool is where？ The, where the nodes stores the transactions that are going to be processed and is free to use, is validated, is a source of truth, is not unique. It can change, they can disappear, transaction, whatever. But while other and if you are using correctly, it's a source of truth, it's like the RAM or main net transactions, the drop is it's a lie because, eh, in fact, there is not a big consensus about what is a draw, but not I can came with a transaction from 2,017. If it's still compliance, it's still gas, it denounce is valid and whatever should be processed as well. And it's a easy to leech and sorry we are going to abuse a little about the protocol.

Speaker 14 29:36
So how this work, first of all, if you want to spend money, you always have to set a deposit. Okay, this is everywhere because it's very important. So you will pay a transaction to make a deposit in a main net contract. That is their role.

AI Audit

Speaker 7 01:09:11
Good afternoon. My name is Derek Vivian. I'm PK. And today we're gonna show you guys what we built. But before we do that, **we're gonna talk about why**. So last year, $3.6 billion

was stolen from the web3 economy, and over half of those were actually from audited projects and protocols.

Speaker 7 01:09:31
So for this hackathon, we thought to ourselves, **what can we build to help** developers build with a security first mindset? Our solution was AI Audit. It's an open source platform where engineers can use natural language to write audit test and deploy smart contracts to any blockchain.

Speaker 7 01:09:50
AI. Audit shifts left the security best practices **by enabling** security specific testing and rapid prototyping to take place much earlier in the development life cycle.

So this is actually aimed for engineers who want to build and incorporate security earlier in their life, in their development life cycle. And this is **an example workflow**. You would upload a smart contract and use natural language to ask Chat GBT to write and execute unit and fuzz tests. You can also ask Chat GBT to execute static analysis test. And then all of those results are exported to IPFS where you and your team can fix those vulnerabilities. **And last but not least**, when you're ready to go, you can deploy it on a live test net or main net. And obviously, we're gonna show you guys how this is done.

Speaker 2 01:10:38
Hey guys, so yeah, here's a demo. So let me open up my screen. So basically right now I'm talking to the ChatGPT and I'm basically saying run static analysis on my contract. So I'll just get the path of the contract. It's basically like kind of an uploading thing. I press enter and my bad, I wasn't running the chat. Give me a second. Awesome. So let me just copy this thing again, my prompt. And as you can see, the chat GBT is processing the message and it's thinking what it needs to do. It finds a Mithril tool, which is a static analysis tool that allows us to run static analysis on this particular file. It find some results, it saves those results in this file. And then once it's done, we have another tool, which is basically view test tools to result. Essentially what it does is it takes this text file and saves it in file coin. And then if you copy this ID and basically go here and go here and then I just replace this. And you should see the error that the Metro was seeing. And then let's go back here.

Speaker 2 01:12:02
Essentially, our goal is to basically reduce the friction for folks to come and work with blockchains. Because what we want is we're gonna harness power for AI of AI and allow them to use multiple tools and to make contacts more secure. And like one more thing we can do is like we can say, I'm new to blockchain. Help me write a simple, yeah, this type of, I'm a new, I'm new to blockchain and help me write a contract. This is something which I've not tried yet. So we'll see how that works. And then it says, okay, I can help you. And then say, okay, can you create random ID in a contract. Let's try that. It's basically, as you can see, it's converging and then the agent is looking what to do next. And low Bill, there's an error that we need to figure out. But essentially, like you can continue talking to like that and figure out. And basically I have write contracts with us.

Speaker 7 01:13:17

Perfect. So what we saw was us using Chat GBT to run a static analysis test Mithral on a local smart contract.

So how we made AI audit was quite simple. ChatGPT was the base layer, the large language model we use. And we use Lang chain agents on top. So these agents are actually taking natural language and deciding what to do. And what we built for the second one was the AI tools. So these are the tools that the agents will execute, run certain tests, deploy certain contracts, compile certain contracts as well. And the three auditing tools that we support are slither, Mithril, and echidna.

Speaker 7 01:13:58
In the future, we want to build more AI tools and tests, a front end as well, and eventually be able to support open source models. So not just ChatGPT. And last but not least, we want to build for auditors. That's really all we had. Thank you guys.

Piggy

Speaker 16 35:09
Everybody. Is everybody still awake? Yeah, my name is Nicholas and today I'm going to show you piggybank 6,5,5,1. The goal of this project is to create NFTs that act like piggybanks, where you can put into the NFT and you need to burn the NFT in order to get that ETH back out. So here's the collection on Open Sea on Gurley, which I encourage you to check out. I'll show you a QR code in a second, but just give you a snapshot of how it works.

Speaker 16 35:38
Here are all the NFTs in the collection so far. Each one visualizes how much eth is inside of the piggy bank currently. It also has some structured data in the attributes of the NFT that represent the same information and whether or not it has been burned yet. And all of these visuals are generated on chain with an on chain SVG metadata rendering token UI function. So I'll explain to you a little bit about how this works and then we can do a demo and see that it really does work.

Speaker 16 36:08
So for this hackathon, I wrote two contracts. The first one is a fork of the 6,5,5,1 account implementation. My version of the implementation essentially breaks the execute call function and the 1271 signing so that you cannot, as the owner of the NFT, transfer anything out of the wallet associated with that NFT. It also has a special on ERC721 and 1155 received functions, both of which reject all NFTs, either 721 or 1155, unless they are the NFT that owns that account. Why is this important? Because if you send the NFT that owns an account to its own account, you essentially burn that NFT according to the 6,5,5,1 standard because the owner of the NFT is the contract itself and no one can exfiltrate anything from it.

When that happens, my implementation of the 6,5,5,1 account releases the eth to the person. They address the burned DNFT in the first place. The second contract that I wrote is piggybank NFT. So the implementation that I just discussed works with all NFTs already. So if you have a mlady or a Terraforms or a loot, you can already, using this implementation, send E to it and use it as a piggy bank, requiring that the owner burn it in order to get the eth back out.

Speaker 16 37:20
However, that's only so interesting. **What would be even more interesting** is to visualize it in the metadata of the NFT itself. And so I wrote a custom NFT contract to do so. And that is the piggyback NFT. This contract essentially wraps all the functions to deal with the wallet nicely and also visualizes in the token Uri.

Speaker 16 37:39
So let's take a look at how it works. We can go here to the etherscan page for it. And mint 1, I set the price at 0.01 E. And if Gurley cooperates and its index are 2, this transaction should process pretty quickly cuz we're sending massive gas just for you. Okay, I think that's a good sign.

Speaker 16 38:03
So if we go back here, we should see a new NFT added to the collection. I think it'll be No. 12, maybe. Let me just check that I'm right about that. Oh no, No. 11. So this is the new one. So you can see it has zero eth currently. So I can go to the NFT contract, go to this add ETH function, and I can add one if let's say to token No. 11. Anyone can do that, of course, but makes most sense if you own it. And wait for that transaction to process, and we should see it over here, decrement, how much is in my wallet.

Speaker 16 38:45
We can also simultaneously go over and look at the account wallet itself by using this read function on token No. 11 and grab the address associated with that NFT. So we can see it has one Ethan. So if we go over here and refresh the metadata, it updates both the number updates and the color of the background changes dynamically. According to that, you could also send the e directly to the address of the wallet. And let's just grab that here. So these functions, just to prove that these functions are really just wrapping the native 6,5,5,1 functionality. So we can send another e here and that'll process sooner or later.

Speaker 16 39:25
While we're waiting for the metadata to refresh and the block to be included, we can also check out the token Uri function, which returns on chain SVG base 64 encoded string here. And we can get ready to burn it. So let's just refresh the metadata just to make sure that it worked. There we go. Oh, there we go. To Eve. And it's time to get the Eve back. We want to bust open our piggy bank so we can go to the burn function, punch in the token ID. Of course, we're the owner, so we're allowed to do this. No one else would be able to. We can write, send the transaction and as soon as it confirms it's burned, so we can see that reflected in the image. Thank you. So we can see the title has changed, the image has changed, and also the structured data, the attributes have changed as well. So that's the essence of the project. From the user's perspective, it's really easy to use and I think it's a

good starting point for building similar applications with more sophisticated assets. So thank you very much. Feel free to check out the Github and follow me on Twitter. Thank you.

zbay

Speaker 18 49:07
Hi, we are zbay? I'm Byram. This is Sergey. And we wanna democratize online marketplaces. So last year, $10 billion for paid in merchant fees to eBay. That's like 15% of the sales volume. I think it's too high. I would think it's time to change that.

So how do we achieve that? Three steps, really. First, for the merchants that have an established reputation, we wanna allow them to import their reputation on chain from eBay, together with the data of all of the product listings they have. The second step is to basically implement a smart contract. That's basically an escrow mechanism to ensure trustless exchange of goods and money. And the third, well, sometimes packages are not delivered or are not received. So we want to make sure that there is a way to dispute a certain transaction, and we'll leverage the Uma optimistic Oracle for that. So Sergei will show you the demo.

Speaker 3 50:07
Yes, what we have here is on the left is the seller and on the right is the buyer. They connected to different wallets. It's solemn noses main net. And then to start interacting with that, we're gonna start with the seller. And you see that my reputation score is zero.

Speaker 3 50:25
What I can do about that, first of all, I can import my eBay reputation. We figured that you can actually change the profile name of your eBay profile to like nature Ena's name. And then what we're gonna do is we're gonna scrape that page and extract this information from here and then generate a ZK proof with our own circuits to submit on chain, to verify on chain, and then it will eventually affect my square as you can see here.

Speaker 3 50:51
And then the next variation of reputation improvement is CISMA connect. Basically, for in here, we try to fight civil attacks and also front end with signature. And we ask to be a Kitcoin passport holder with over like 15. This is a points and score.

Speaker 3 51:13
Then we generate a ZK proof with C small to once again submit that on chain and verify on chain as soon as it clears to be ready to proceed. Call the first. This is the XMTP connection. We will talk about that later as well.

Speaker 3 51:34
This is the seismic proof submission. It goes there. And then my reputation score will be updated and I'll be ready to create my list, my products on Zbay. So to list my product, I can

actually import that if I have an existing product on eBay. I have handy links there. This is a PS five product that is listed there. You can also like input data manually. That's fine as well. But it's cool if we just import that for you. We use just, you know, old school APIs to scrape that page and extract everything. You can see it pops up. What we're gonna do now is in the back, we actually submit all that to IPFS and store there. So we don't clutter the blockchain with, you know, all this metadata. And then we submit that transaction to post this listing of this product on chain should be happen, should happen. Mom and Charlie, it happened. And then we use the graph to actually power this list of published products. Sometimes it takes a minute, a second to actually pop up from the graph while it picks up the log, we use our own subgraph. Let's see. Yeah, it pops up here. I don't need that. Okay. No worries. So the buyer should be able to see that as well. I see the PlayStation and then we see that this is the price that I have to pay. And then this is some extra deposit with this multiplier. So deposit is needed is I pay this price. And this price is the money is actually isn't in escrow now. So no one has it, neither seller nor buyer.

Speaker 3 53:29
And then buyer will send me the product and I don't have any incentive to actually confirm that I received it. I can just, you know, cheat the system and be the best that guy. That's why we require you to submit a deposit as well. And then as you confirm that you actually receive the package, then we will give you a deposit back. And we will do that as well to see how it works.

Speaker 3 53:52
And the multiplayer, we calculate based on different air stack requests to see how old is your like account, how many pops you have and stuff like that. And after we see that we purchased money transfer to escrow, we waiting for the seller to dispatch. Seller doesn't really know where to send. So we can actually chat with the buyer. I use an XMTP and like ask percent. And then the buyer can also go to the chat and see this message and like, you know, this fake address. I see the address now. So what I can do is I can go ahead and dispatch the item. What actually happens is I submit the item to, let's say, FedEx. They give me the tracking number, and then I can also generate the secret. So the secret is something that I put with the package, like drawing the delivery envelope or something like that. But when I click dispatch here, we're gonna do the hash of those things and commit the hash on chain. And then the buyer will need to provide a ZK proof that they know they have the actual secret. And we can do that momentarily as well. That should make sure I copy that because I sell this, I'll lose the secret. No one knows the secret. I update here.

Speaker 3 55:22
And I can also dispute if I want, if I didn't receive the package. For instance, we use Uma for that. But I can also confirm the delivery. It was ABC 1,2,3. And this right now, we use our own circuit snark Jess with the circum to calculate the proof, submit the proof on chain. So secret is never, you know, public anywhere, stuff like that. It's verified that I do know the secret because the hash is the same, that transaction will clear and you could see that balance updated. So I get my deposit back in here.

Speaker 3 55:57
The, you know, the status delivered to buyer as well. But if dispute could happen, then we actually, when you dispatch the item, we submit the assertion on Uma to say like specific

product with specific, this is APFS identifier. So you can go ahead and inspect the metadata. Who was the buyer, who was the seller, what was the price, what the item was, so you can cooperate with them to, you know, resolve the dispute. And that's about it. If you have questions further down and contact us on XMTP.

Demo 框架 - 5 min

- **POC, pain points, solution**:
    - Why we build
    - What we build
    - How did it solve the problem
- **Demo**:
    - Front end UI
    - Tools used
    - Technical details

Demo 的口语难点

- 词汇难点：背
- 句式表达：自然流畅（中高阶）

Demo 训练计划

- 确定 **demo** 讲稿
- 词汇训练：制定每日词汇量练习计划，朗诵，听写👉词汇记忆
    - 3 words/day, 20 words/week, 40 words biweekly
- 句式训练：分段练习，切分，
    - Why
    - Pain point
    - Summarise what you build in one sentence
    - What problem did it solve
- 一对一模拟：每日/隔日一对一口语模拟，及时反馈
- 基础不够，强度来凑：模仿，重复