CryptoDemetrius [6:00 PM]
Welcome to the first Ardor AMA on Lightweight Contracts! Beginning now and continuing for the next hour, Lior (core developer), Tomi (lead architect), Petko (core developer), and Alberto (technical sales engineer) from Jelurida will be answering your questions on lightweight contracts and the ongoing Ardor Online Hackathon challenges. We kindly ask you to keep your questions concise and on-topic so that we can maximize this valuable opportunity to speak directly with the core developers of the Nxt and Ardor platforms. Additionally, if you have multiple questions, please prioritize which one you ask first – and then be patient as we moderate the conversation to give everyone an equal opportunity to pose at least one question. We will do our best to answer as many questions as possible in the 1-hour timeframe. Note all questions and responses will be recorded and made publicly available shortly after the end of the AMA session. Thanks for participating! (edited)

To kick things off, I will pose a question from the Ardor reddit submitted by "jasfad":
Q1: Will lightweight contracts be usable on all child chains? Or will some child chains block lightweight contracts?

riker [6:02 PM]
A1: Since contracts are a layer of automation on top of the existing Ardor APIs they will be available on any child chain as long as the specific API used by the contract is available on this child chain.

Note however that the contract themselves and the contract references must be deployed to the Ignis child chain.

CryptoDemetrius [6:05 PM]
@krakatit - I see you typing - you are next if you would like

Another question from reddit came from @mrv777 and they asked:

Q2: What is the top shortcoming of the current lightweight contracts vs ethereum's contracts and then the top one against EOS?

krakatit [6:07 PM]
Q3: Would you please giva e a simple example of a lightweight contract and what would be the difference if I use the Etherium smart contract system?

riker [6:07 PM]
A2: It's a different approach which has tradeoffs. Our contracts are deployed to the blockchain but their execution is not part of the consensus.

CryptoDemetrius [6:07 PM]
Q4: Another question came from the same user, asking:
Q: Can you develop contracts in other languages or only in Java? Examples: JavaScript, C, Python. Maybe with some conversion tool or wrapper libraries?

riker [6:07 PM]
A2 continued: This means that you can do much more. But you need at least partial trust in whoever executes the contract.

A3: @krakatit let's look at a token contract, in Ethereum the contract is the token, all token balances are saved into the contract. If there is a bug, anyone can change the token balances.

krakatit [6:09 PM]
OK, thanks!

riker [6:09 PM]
A3 continued: In Ardor, the contract receives trigger transactions as input (like in Ethereum) but it does not make changes to any internal state, instead the output of the contract are normal transactions which still needs to comply with all the blockchain rules.

CryptoDemetrius [6:10 PM]
There was another question from @thewiremaster posted on reddit, they asked:
Q5: Are there plans to add more Smart Transactions in the future to give even more tools to the LightWeight Contracts?

riker [6:10 PM]
A4: You can develop contracts in any languages which compiles into Java class files like Scala or Jython but our tools are Java based.

A5: Contracts can use internally all the features of Ardor including phasing (that we use to also call smart transactions) so multisig for example is available out of the box. You don't need to code a contract for multisig for example.

the_chestnut [6:12 PM]

Q6: hi all, this is all very interesting approach, I tried to learn a bit more about the lightweight contracts and I have a question: what happens if the contract runner does not execute the contract intentionally or not?

riker [6:13 PM]
A6: There are several approaches you can take to make sure the contract runner does not cheat. Anyone else can also run the contract and double check if they reach the same results. But then they still cannot block the contract runner.

mrv777 [6:13 PM]
If there's time, I'll ask the obvious question:
Q7: Why would a developer choose an Ardor lightweight contract over the well established, and rich developer community (both in number of people and money) of ethereum or EOS smart contracts?

riker [6:14 PM]
A6 continued: A stronger guarantee is to place the contract runner account under account control so that every transaction submitted by the contract runner has to be approved by another contract runner controlling another account or 2 out of 3 etc

riker [6:15 PM]
A7: Our contract framework allows for much richer applications. Traditional contracts cannot access any external resources. We can. Other contract frameworks need to pay gas for every instructions. We don't have this requirement.

You develop in Java, the most popular programming lang for business.

mrv777 [6:15 PM]
Q8: Hmm, so you can chain contract running accounts together?

riker [6:17 PM]
A8: You can of course have one contract invoke another contract. Or another contract runner confirm every transaction submitted by a specific contract runner.

riker [6:18 PM]
A6-8 continued: One more thing, if you look at Ethereum plasma, the main idea is to no longer execute every contract on every node. So this is where the industry is going anyway.

almonte [6:18 PM]
From a business point of view, the fact that the contract execution does not rely on the consensus algorithm is an advantage. Ethereum network reached the max capacity for instance

riker [6:20 PM]

A6 continued: If you no longer require that contracts execute on every node then how do you make sure they are executed at all and not just lock user funds? We have an elegant solution for this, albeit a bit difficult to use. We call it phasing by shared secret.

The idea is that when you trigger a contract the funds are not sent until you share a hashed secret. The contract transactions will also use this hashed secret so that when the secret is shared both the trigger transaction and the contract transactions will be applied together or not applied at all.

krakatit [6:22 PM]
Q9: Do you have an idea how many lightweight contract can be processed simultaneously?
I mean contracts.

riker [6:24 PM]
A9: Since the contracts are executed locally by a specific contract runner there is no real max, it depends on the resources of the contract runner node. What is limited is the number of transactions which can be submitted by a single contract invocation. We currently limit it to 9.

A6 continued: Just to emphasis my previous point, using the shared secret approach even if the contract runner does not execute the contract the funds are only locked temporarily and will be released once the phasing height is reached usually within few hours.

krakatit [6:25 PM]
It looks like a great advantage.

riker [6:26 PM]
One great thing we have is that each contract has a unit test and contracts can be debugged using a standard IDE on the testnet blockchain.

krakatit [6:26 PM]
Q10: How long did it take you to develop the lighweight contracts system?

riker [6:26 PM]
You can even use the unit test to test the contract off blockchain before you deploy it. We do it all the time.

A10: We started actual development on March this year. Ardor was deployed to production on January.

riker [6:27 PM]
I mean contract development.

krakatit [6:28 PM]
Yes, but you had the great background from NXT!

efunkem [6:28 PM]
Q11: There is some text mentioning that DistributedRandomNumberGenerator contract may produce predictable random numbers... can you explain that in more detail?

CryptoDemetrius [6:28 PM]
A question I received at a recent conference:
Q12: What happens if a contract runner node crashes?

riker [6:28 PM]
Exactly, the reason we can use this approach is that we build on top the rich functionality of NXT/Ignis.

A11: So, since the contract is not run by every node, it can use random data generated by a specific node locally.

This makes it much simple to generate random data without having to rely on external Oracles like the Ethereum randao approach.

Instead each contract runner can generate its own private random seed, or any key to any external service and not share it everyone else.

Then there's the issue of how do you share this private data with someone else to validate your actions? So this should be done off blockchain. There has to be some level of trust in the contract runner.

The random numbers are predictable since given the same seed every contract runner will generate the same random number sequence but the numbers themselves are still random.

chuffy [6:33 PM]
Q13: good evening, I have a question - can the contracts be run on Ardor or it has to be on the child chains? Sorry if it was already explained

riker [6:35 PM]
A13: Contract can also run on Ardor itself as long as the API needed are available on Ardor. For example the RandomPayment sample can run on Ardor. But a contract that relies on account properties cannot since these are only available in IGNIS.

In fact the contract does not run on a chain. It is run by a contract runner on whichever chain it is designed to use.

chuffy [6:36 PM]
got it, thanks!

CryptoDemetrius [6:37 PM]
(reposting)

Q12: A question I received at a recent conference:
What happens if a contract runner node crashes?

riker [6:37 PM]
A12: The operator of the node will have to restart it automatically or manually and it will catchup.

If the crash is due to the contract itself, some manual action of fixing and deploying a new version is required.

Note that since the contract is stateless there is no need to migrate data from an old contract version to a new one.

madfox [6:39 PM]
Q14: Hello! Perfect AMA and I have a question (actually this is an invitation) Mr. riker, it will be a great honor for NxterPuzzle if you become a special guest in our weekly show in new twitter format - A Week With...in the end of November!

riker [6:39 PM]
A14: :grinning: sure thing

CryptoDemetrius [6:39 PM]
Q15: Are there any potential risks during the reboot process for transactions sent to that contract runner node during the downtime to end up as unconfirmed or dropped?

riker [6:41 PM]
A15: We take great care to make sure the contract runner won't miss any trigger transaction. For example when a contract runner restarts, it pops off the last block in the blockchain to make sure it has a chance to reprocess the trigger transaction in the last block.

We also have to take care of switching to another fork, of re-downloading the blockchain, there are quite a few special cases to handle. This should all be masked from the operator. The worst that can happen is that you'll see some duplicate unconfirmed transactions which will expire and not make it to the blockchain.

Perhaps questions about the coding challenges?

the_chestnut [6:43 PM]
Q16: I know that pruning of child chain transactions is on the Ardor roadmap. What will happen with the contracts that are running?

riker [6:44 PM]
A16: Pruning only affects transactions already stored in the blockchain for say 1440 blocks. Contract runner will always operate on the block at the top of the blockchain.

Another effect is that contract code itself will be pruned if nobody cares to save it in an archival node. (edited)

the_chestnut [6:48 PM]
Q17: so I guess contracts that continue to be used all the time need to be available from archival nodes

riker [6:49 PM]
A17: Right or redeployed frequently enough.

CryptoDemetrius [6:50 PM]
Final 10 minutes, don't be shy if you are lurking in the background - this is your chance!

atzen [6:50 PM]
Q18: Hello! I have a question regarding the contract manager and runner accounts. Are they completely separated? For example can I deploy a contract from node A with a manager account and run it on node B with another runner Account?

CryptoDemetrius [6:51 PM]
One question I heard in a chat room:
Q19: Why did Jelurida pick these 3 challenges? Is there any particular reason or broader vision behind it?

riker [6:51 PM]
A18: Yes, the contracts are designed so a contract runner can run safely contract it did not deploy or does not trust.

So there could be separate teams of contract developers and other teams of contract runners each specializes in its own domain.

A contract downloaded from the blockchain runs in a sandbox on the contract runner machine so it cannot access local resources unless permitted by a policy file. (edited) This is intuitively similar to apps running on a phone.

A19: About the selection of the challenges.
We wanted challenges which are not just tech demos but are really solving real problems. Like the reputation system of #3 or storing large files on #2 wallet in #1. We don't want you to develop tree registry for a banana plantation. We want you to solve real world problems.

@atzen did I answer your questions? The short answer is Yes

CryptoDemetrius [6:57 PM]
Q20: If there is time to make these challenges more tangible: If you could compare each of the three Ardor Online Hackathon coding challenges to an existing process or product, what would it be for each challenge? For example, is the identity verification/reputation

challenge similar to the "Sign in with your google/facebook/social media" account options that exist on many sites or is it different?

riker [6:59 PM]
A20: #1 is about simple exchange between our tokens and Bitcoin, #2 is to enhance Ardor to support storing large files, the motivation of #3 is explained in this medium article https://medium.com/@lyaffe/blockchain-based-reputation-system-8d2eb6d7260a I think it is perhaps the most important challenge.

veronica [7:00 PM]
Really nice AMA! I have learned a lot

riker [7:01 PM]
If you like to read more the official documentation is here
https://ardordocs.jelurida.com/Lightweight_Contracts

CryptoDemetrius [7:02 PM]
Thanks for joining everyone. It was great to meet some new faces this evening! We hope to see you around the slack more frequently, and never hesitate to reach out with any additional questions!

The team is always around and ready to help.

atzen [7:02 PM]
@riker yes thanks

veronica [7:05 PM]
For the ones that are going to participate in the Online Hackathon, we remind that all the information is in this link:
https://www.jelurida.com/ardor-hackathon-2018

Subscribe to the list to receive Jelurida's emails.

Two additional questions were asked shortly after the AMA ended and they are included below:

mrv777 [Yesterday at 7:52 PM]
Q21: So what is the best way to test and debug a lightweight contract? I have one deployed but want to watch some sort of log of what happens when I try to trigger it? I know unit tests were mentioned for testing, how would I do that with the contract I wrote?

riker [8:45 PM]
A21: You can see on the Intellij project that each contract also has a unit test. Ypu can run the unit test from Intellij and place a breakpoint in your contract. You can even run

the node itself from intellij and debug your contract and as usual all log messages are written to the ardor log.

efunkem [7 hours ago]
When the contract deploys, it will open a log in IntelliJ. It'll log about 2 seconds worth of information to tell you that it successfully deployed. Then to see the log for the actual transaction trigger its under the main Ardor node log. For example, AllForOnePayments contract will log every single block that goes by, until it hits the block when it triggers, and then it will log the information from triggering the transaction (example is at the end of the Udemy class on lightweight contracts if you're curious... can get the class for free with "ARDOR-CONTRACTS" coupon)

mrv777 [12:12 AM]
Q22: Maybe it's something crazy with my setup, but running ardor reguarly on my computer, using Ardor Local in Intellij, and running the unit tests; results in 3 different DB folders.
1.) AppData/Roaming/ardor/nxt_test_db
2.) ardor/nxt_test_db
3.) ardor/nxt_unit_test_db (edited)

riker [8:45 AM]
A22: This is normal (well, somewhat normal)
1. Is created by the standalone node. So normal testnet db.
2. Is created when running the node from IntelliJ (it should also use 1. but because of something in your setup it uses the local folder and not your user folder)
3. Is the unit tests blockchain db, this is a dummy blockchain created only for the unit tests. So this is normal too.tests. So this is normal too.

malnemark [8:47 AM]
@mrv777 its the same on my setup, isn't hurting much is it. As long as you've got enough disk space
I ended up using JUnit for the development (its actually good practice to develop around test cases, especially when they're thought through and make sense) so didnt have to resync to testnet since a week. But I have approx 5-10 test cases around which I write the contract.
Only when they all pass I'll need to keep the two testnet DBs up to date.
When deploying and testing. Lets see how this goes..