Twitter Thread by Bartek Kiepuszewski

Bartek Kiepuszewski
@bkiepuszewski

[1/13] It may be initially confusing to fully grasp how deposits and withdrawals from L1 to @optimismPBC are actually implemented, and it helps to see the on-chain action of what is happening behind the scenes.

[2/13] Initial setup (simplified): on L1 we have SyntheticBridgeToOptimism from Synthetic and OVM_L1CrossDomainManager from Optimism contracts. On L2 we have SynthetixBridgeToBase and OVM_L2_CrossDomainManager contracts.

[3/13] Additionally we have Sequencer (L2 mining node) that verifies all L2 transactions and submits them in batches to L1 for future reference and Relayer that is responsible for relaying messages from L2 —> L1

[4/13] Step 1 - Alice wants to deposit $SNX to L2. To do that she calls initiateDeposit() method on the L1 SyntheticBridgeToOptimism which takes her $SNX, puts it in the escrow and calls OVM_L1CrossDomainManager sendMessage() method.

[5/13] The CrossDomainManager puts this request to CanonicalTransactionChain (this is an “official” and “unmutable” list of all L2 transactions on L1). As a consequence the Sequencer (L2 mining node) will need to execute this transaction on L2.

[6/13] This will result in invoking completeDeposit() method of SyntheticBridgeToBase on L2. This method will simply mint L2 $SNX tokens for Alice. See the trace below: https://t.co/VKFgJFyOWw

[7/13] Step 2 - Enjoy cheap L2 life. This trace below shows the L2 Sequencer submitting a batch of 346 L2 transactions issueMaxSynths(), burntSynth(), initiateWithdrawal(), updateRates(), etc… to L1.

[8/13] With no gas optimisation, on average, gas used per L2 tx was 26,138 or 3$ per transaction. All transactions are put in the CanonicalTransactionChain, the same used by CrossDomainManager before. https://t.co/vZK32LjJTn

[9/13] Step 3 - Alice wants do withdraw her $SNX from L2. To this end she calls initiateWithdrawal() on SynthetixBridgeToBase on L2 which sends the msg to L1 through OVM_L2_CrossDomainManager.

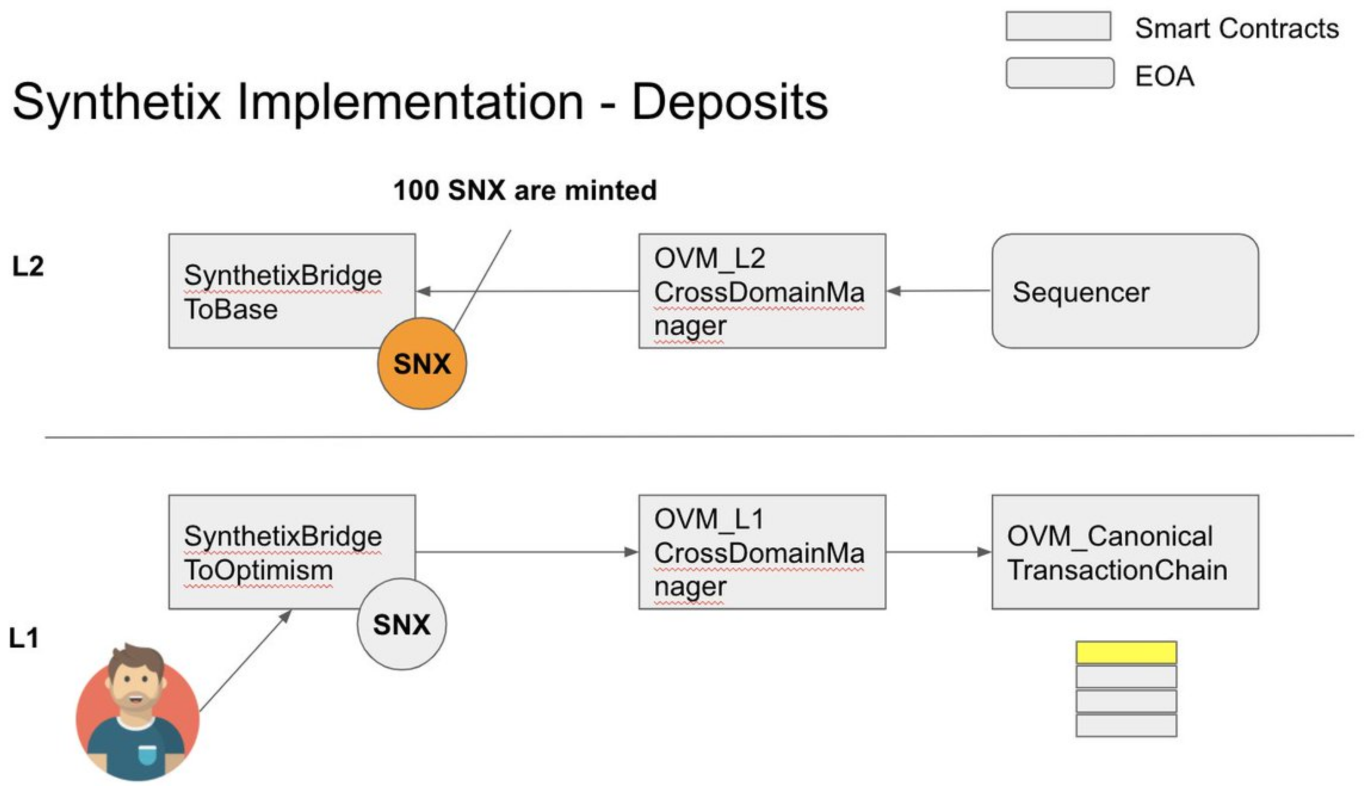
[10/13] CrossDomainManager changes its state which forces the Sequencer to commit this new L2 state to L1. You can see this in the next trace, with the inititateWithdraw() method being present in the Sequencer's batch of 6 L2 transactions https://t.co/J2GeauWFRN

[11/13] Step 4 - Now we wait to make sure that the state root commitment submitted by the Sequencer is indeed valid. If nobody submits Fraud Proof that the state is incorrect, we can assume that it is indeed OK and it will never be rolled back

[12/13] Step 5 - after the FraudProofWindow has passed, the Relayer can finally relay message from L2 to SynthetixBridgeToOptimism contract. It constructs proof that convinces OVML1CrossDomainManager that this message was indeed submitted by Sequencer to CanonicalTransactionChain

[13/13] As a result L1CrossDomainManager will call completeWithdrawal() method on SynthetixBridgeToOptimism which will release escrowed L1 $SNX tokens kept there. https://t.co/1O3E78f0Sp

https://t.co/m2H1lCMDzB



https://t.co/KSUmMWFNeG

Synthetix Implementation - Withdrawals