Motivation

Solidity is a great language for articulating contracts over on-chain events. Especially for smart contracts about real-world systems however, there's a need to perform computations which cannot be efficiently described via solidity. For example, for a truly trust-minimal version of our interactive https oracle protocol, which uses enclaves to create proofs of web2 events, a smart contract would have to directly verify a special certificate chain provided by the enclave manufacturer/host, whose signatures are generally on curves that are not natively supported by Ethereum. We currently solve this by having one trusted enclave permanently running that verifies the certificates of others, but that trusts us developers for the liveness of the system.

Of course, the utility of supporting general purpose computation in smart contracts goes beyond web2 oracle protocols - it unlocks the whole stack of battle-tested software packages developed before blockchain came around. It could position Arbitrum as a trustless settlement layer for all kinds of currently non-crypto apps.

Proposal

I want to propose allowing solidity contracts to query the result of executing arbitrary WASM code on a given input. For simplicity, we should require both the input and the .wasm file to be posted on l1, at first.

Maybe the easiest EVM-centric design here would be to modify geth to let calls to a special 'precompiled contract' trigger the execution of arbitrary WASM code, feeding back the results to the calling contract in a synchronous fashion.

What would be the main challenges to implementing this?

Alternatives

Basically, there's only Cartesi and Truebit. Cartesi is great, but unlike Arbitrum, it doesn't pursue a global consensus model, requiring contract users to trust a whitelisted committee to challenge invalid state updates - hence contract developers are likely still a single point of failure. Also, solutions like Truebit and Cartesi always require a delay between the contract query and the result - whereas extending an optimistic rollup to handle non-evm execution would not come with an additional delay, since optimistic computations can trust each other. Arbitrum is in the unique position to provide this utility to the ecosystem, allowing it to capture a unique network effect of contracts about real-world events.