Objective: The zkFold Symbolic project aimed to develop a new Haskell DSL for writing zero-knowledge smart contracts on the Cardano blockchain. The goal was to simplify the development process by abstracting the complexities of ZK cryptographic protocols, allowing developers to build efficient, scalable smart contracts without requiring expertise in ZK cryptography.

Project Status: The project was completed within the timeline and delivered all expected outputs.

Key Accomplishments and KPIs

- Implement basic types (boolean, fixed-size integer, fixed-size bytestring, fixed-size arrays) and basic operations with them. Implement equality, comparison, branching operations. Implement hashing functions.
- Implement the Plonk on-chain verifier. Implement an end-to-end zkFold Symbolic smart contract test on testnet.
- Create the zkFold Symbolic Cardano Library. It contains all necessary types to work with Cardano transactions inside zkFold Symbolic smart contracts.
- Implement the post-processing algorithms that converts a zkFold Symbolic script into the arithmetic circuit that is verified on-chain.
- Implement variable-size types and operations with them (analogous to Integer and ByteString types in Haskell).
- Develop a set of examples, showcasing zkFold Symbolic smart contracts.
- zkFold Symbolic documentation (https://docs.zkfold.io/symbolic/introduction/)

The project was completed within the allocated budget. No significant cost overruns occurred. Successfully completed all the milestones.

Final Thoughts: The project has the potential to significantly impact the development of smart contracts on Cardano, reducing fees and increasing efficiency. zkFold’s team’s collaboration and careful planning were key to the project’s success.