Formal Modeling of Distributed Ledger Applications

Background.
Smart Contracts are programs that work on a blockchain or a distributed ledger (e.g., Ethereum or Hyperledger Fabric). They take control over assets in that ledger and manage them autonomously, according to their program logic. Since smart contracts are hard to change after deployment, programming errors usually have very serious consequences. Furthermore, smart contract source code is often open for all to see, so that exploits can and will be found by adversaries. Therefore, it is necessary that smart contracts are correct upon deployment, and formal methods are needed to ensure this. In this context, it is not sufficient to consider single transactions, or even a single contract: Every smart contract creates a distributed application in its environment.

Goals.
The goal of this project is to find ways of modeling distributed ledger applications in a way that abstracts from concrete implementations, but captures their behaviour and allows reasoning about their properties.