TxPipe Docs

Tx3 Documentation

Tx3 is a domain-specific language for describing protocols that run on UTxO-based blockchains, with a particular focus on Cardano.

Rationale

In account-based blockchains, dapps are primarily defined by smart contracts with explicit function interfaces that represent user interactions and state mutations. These dapps have a clear API surface that can be used by different parties to interact with its business logic.

The deterministic nature of the UTxO approach is a great property, but it has a drawback: the interface of a dapp is not explicit, there's nothing describing how the different parties can interact with it. Dapps are defined by transaction patterns that represent deterministic "state transitions". A party interacting with an UTxO-based dapp has to understand the underlying business logic in order to construct transactions representing their intents.

This is why we need a strict but flexible mechanism to describe patterns of transactions (which we'll call transaction templates) that dapp authors can use to convey the interface of their dapp and dapp consumers can use to interact with it.

Scope

- a language to describe a dapp as a set of transaction templates
- a set of diagram conventions to visualize a dapp at different levels of abstraction
- an interpreter that takes a transaction template and a set of parameters and generates a concrete transaction
- a tool to generate executable onchain / offchain code from transaction templates

Out of Scope

Tx3 is NOT meant to be a smart contract language, the idea if for devs to continue building on-chain validators using their favorite languages, we love Aiken ❤️.

Tx3 is meant to complement existing languages and tools, not replace them. It positions itself at a higher level of abstraction, describing how user interact interact with validators by constructing the required transactions off-chain.

The Language

Tx3 is a DSL (Domain Specific Language) for describing the interface of UTxO blockchain protocols using a declarative style and basic functional expressions. It allows you to describe user intents in the form of transaction templates that abstract away the transaction building process as high-level function calls.

The language's narrow focus allows it to be highly opinionated and tightly integrated with core UTxO blockchain concepts. Instead of abstracting away blockchain-specific details, Tx3 embraces them - concepts fundamental to UTxO blockchains like inputs, outputs, datums, and redeemers are treated as first-class citizens in the language. This deep integration enables more natural and expressive ways to work with these concepts.

A key concept in Tx3 is the distinction between concrete transactions and transaction templates. While a transaction represents a specific state transition, a template is a function - it defines a pattern that can generate many different concrete transactions based on parameters provided at runtime. This parameterization is what makes templates so powerful for defining reusable transaction patterns.

Learning Tx3

If you're interested in learning Tx3, we have a comprehensive guide that covers all the basics you need to get started.

- **Installation** →
- **Quick Start** →
- **The Tx3 Language** →
- **Example Catalog** →
- **Integration Guide** →
- **Tool Ecosystem** →

Advanced Topics

If you're looking to dive deeper into Tx3, we have a comprehensive guide that covers all the advanced topics you need to know.

- **Architecture & Internals** →