

General Notes

Plutus Core, being on-chain, is what is ran when a transaction is executed by the network.

Plutus is the platform built to execute smart contracts on Cardano, whilst Plutus-core (which is essentially Haskell) is the language used to write smart contracts for Plutus.

Plutus Smart Contracts are Haskell Programs. However, you won't ever write Plutus-core code, you'll write Haskell code which compiles into Plutus-core, to then be executed as a transaction on the Plutus platform.

In Cardano's case Plutus-core is implemented 'on-chain' (it's executed by operational nodes within the Cardano network infrastructure).

Interesting Point: State Synchronisation & Double Spending

Video with M.P.J: <https://youtu.be/usMPt8KpBel?t=915>

Topics Discussed (not already mentioned):

- Backup
- DevOps
- Monitoring
- Logging
- Rollbacks in Cardano: As far as I can tell, Rollbacks (failure to validate a transaction within a series of transactions, which invalidates a portion of the Tx within the series) occur when simultaneous block creation occurs. The longest chain (or series) of transaction will always be chosen to represent the true state of the Blockchain. This is dangerous for DApp developers and non-deterministic implementations of smart contracts, because people may loose their money, or native tokens.

"Constraints Liberate, Liberties Constrain"

— Runar Bjarnason

- Similarly, you could say: Less is More, insofar as: if you truncate the scope of an application, the less you allow it do, the easier it is to control. Thus, it follows: the less likely it is to cause catastrophic failure. Furthermore, it allows for an increased capacity to predict states and behaviour given any set of constraints.
- Designing applications such that they're "deterministic, pure, replay-able, sandboxed

state machines" (Michael Peyton-Jones) means as developers we're much less likely to introduce bugs into our applications (*possibly* mathematically provable).

The Plutus Pioneer Course aims to provide:

- Introduction to the Plutus Platform.
- The underlying concepts on which Plutus operates.
- An overview of Smart Contracts on Cardano¹.
- The course introduces core testing principals and techniques for Smart Contracts.
- The ability to perform offline local testing will be demonstrated.
- A deep dive into Native Tokens on Cardano
- minting and burning of native tokens
- deploying plutus contracts and writing backends for plutus contracts

////////////////////////////////////

Footnotes:

1. Currently limited to the Plutus implementation via Haskell. Although, there will be opportunities in the future to implement these contracts using alternative languages in the future. Such languages include Glow and may include Solidity.