

¿Tengo que saber Haskell para crear SC en Cardano?

- La implementación principal del nodo Cardano está escrita en Haskell.
- La máquina virtual para ejecutar contratos inteligentes integrada en el nodo también está implementada en Haskell.

Sin embargo, esto no significa que la máquina virtual de contratos inteligentes ejecute Haskell!!

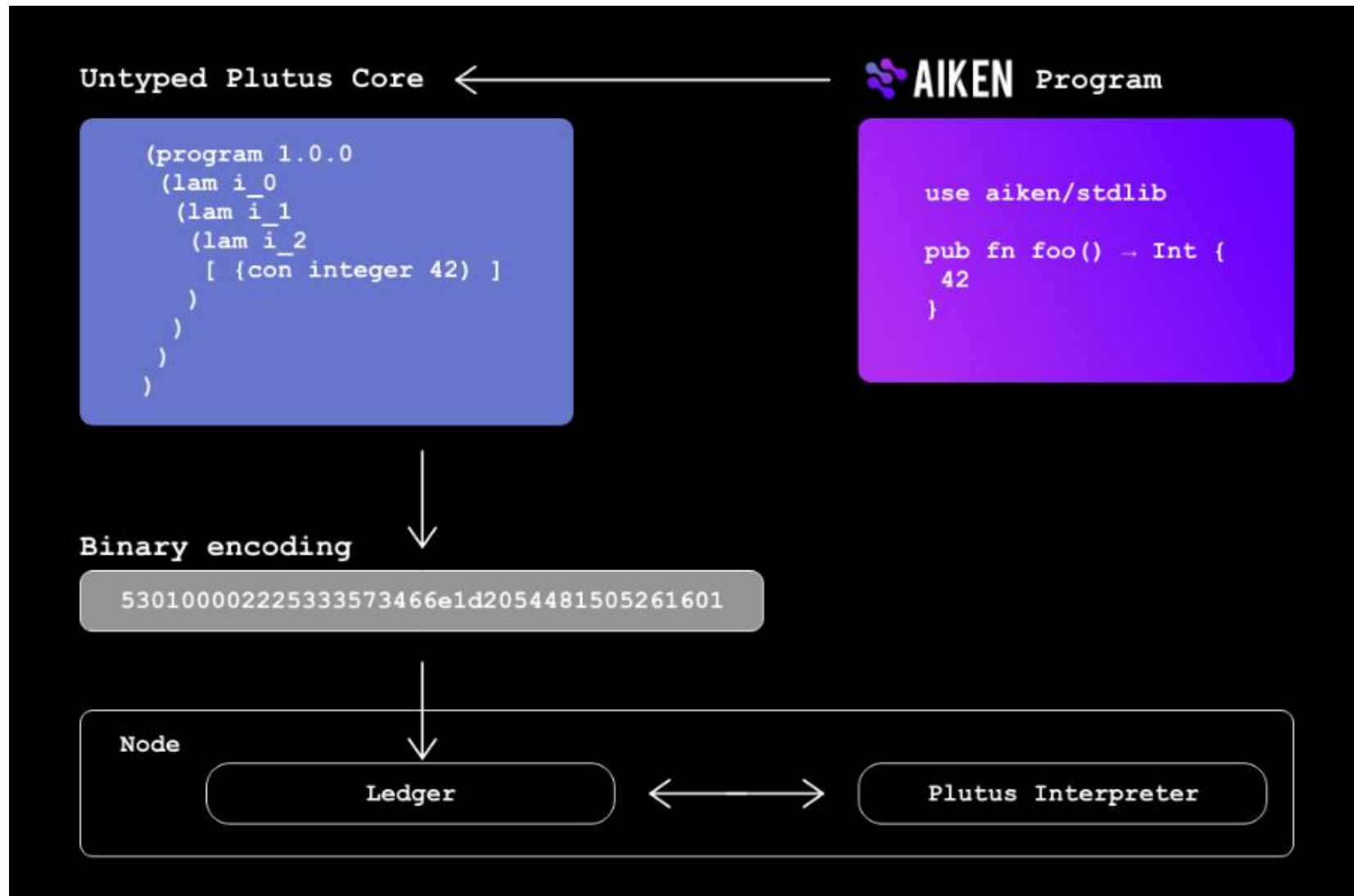
Aiken cuenta con una versión completamente funcional de esta máquina virtual, escrita en Rust!!

¿Qué se está ejecutando realmente en el nodo?

Existe algo llamado Untyped Plutus Core (UPLC), que es la representación de nivel más bajo de un contrato inteligente, y es esta representación de nivel más bajo la que realmente ejecuta la máquina virtual.

¿Cómo genero UPLC?

- **PlutusTx**: está desarrollado cómo un plugin de GHC (Haskell).
- **Aiken**: es un nuevo lenguaje y el compilador está escrito en Rust.
- **Opshin**: permite escribir contratos inteligentes en Python3 100% válido pero restringido.
- **Helios**: es un nuevo lenguaje. Algo interesante es que debido a que el compilador es un único archivo javascript, es bastante fácil usar Helios desde dentro de un proyecto javascript.
- **Plutarch**: no es un lenguaje nuevo. Se puede considerar un eDSL (language de dominio específico embebido) para crear contratos inteligentes con Haskell.
- **Plu-ts**: no es un lenguaje nuevo. Se puede considerar un eDSL para crear contratos inteligentes con Typescript.
- **Scalus**: es una implementación en Scala de Plutus.



<https://aiken-lang.org/>