Teoría de Lenguaje

Solidity

* ¿Que es Solidity y para que sirve?
  + Solidity es un lenguaje de alto nivel, con sintaxis similar a JavaScript.
  + Esta diseñado y compilado en código de Bytes.
  + Pensado para la creación y desarrollo de SmartContracts los cuales se ejecutan en la maquina virtual de Ethereum, la cual es una blockchain.
  + Solidity es un lenguaje Turing Completo.
* SmartContracts
  + Un contrato inteligente es un programa informático que ejecuta acuerdos establecidos entre dos o mas partes haciendo que ciertas acciones sucedan como resultados de que se cumplan una serie de condiciones especificas
* El Gas
  + Cada transacción que se lleva a cabo mediante los SmartContracts consumen ‘Gas’.
    - El Gas no es ni mas ni menos que el gasto computacional de procesar una transacción o contrato inteligente en la red
    - Esto evita que se den bucles infinitos los cuales colapsarían el sistema
* BlockChain
  + Bloques
    - Bloque génesis es el primer bloque de la cadena el cual se genera cuando se instancia el contrato por primera vez
    - Cada bloque almacena
      * Las variables con su estado al momento de la creación del bloque,
      * Su hash el cual se genera en base a las variables mencionadas
      * El hash del bloque anterior y una estampa temporal.
  + Seguridad
    - Hash: Cada hash de cada bloque depende de la información guardad dentro del bloque por lo cual si se modifica la información guardada se modificara el hash y el siguiente bloque también deberá ser modificado para que apunte al nuevo hash
    - Proof of Work: Cada bloque tarda 10 minutos en crearse, de esta forma si quisiéramos modificar un bloque del medio de la cadena, necesitaríamos re-calcular todos los bloques siguientes y esto tendría un costo de tiempo muy alto.
    - P2P Networ: Cada vez que un nuevo usuario se suma a la red se realiza una copia de la cadena en la computadora de un nuevo usuario, desde esta forma en caso de modificar un bloque en la cadena, el mismo bloque debe ser modificado en todos los dispositivos de la red.
  + Privacidad y Transparencia
    - Cada uno de los bloques pueden ser observados por cada una de las personas que forman parte de la red, lo cual le aporta un grado de transparencia muy alto a cualquier transacción que se realice
    - Al estar totalmente codificada, la blockchain el uso de la blockchain es anónimo y privado.
* Sintaxis
* Manejo de memoria
  + hay tres tipos distintos de memoria en solidity:
    - https://ethereum.stackexchange.com/questions/23720/usage-of-memory-storage-and-stack-areas-in-evm?noredirect=1&lq=1
    - storage: es el mas caro de los tres, tiene un costo de 20k gass para setear un espacio de memoria y de 5k gas para cambiar el valor. Esta memeoria es la que se guarda dentro del bloque de la block chain, y por eso su elevado costo. La idea de esta memoria es guardar valor que se quieren mantener constantes entre las distintas ejecuciones del contrato.
    - Memory: es relativamente mas barata que el storage, cuesta 3 gas para leer y escribir, y un poco mas de gas para expandir la memoria. El problema es que el costo aumenta de forma cuadratica cuando cuando se requiere mas memoria. Los costos anteriores era para guardar kbs mientras que para un mb de memoria constaria unos millones de gas. A diferencia de el storage, esta memoria es volatil, solo existe en tiempo de ejecucion del contrato.
    - Stack: tiene un costo similar al memory, tiene un maximo de 1024 items, pero solo los primeros 16 bloques son accesibles con facilidad. Si un contrato se queda sin memoria en el stack este fallara. Al ser parte fundamental de la ejecucion del contrato lo mejor es no tocarlo y dejar que el compilador se encargue de guardar lo que necesita dentro del stack.
* Manejo de CPU
* Concurrencia
* Problemas de Ethereum
* Comparación con EOS
* SandBoxing