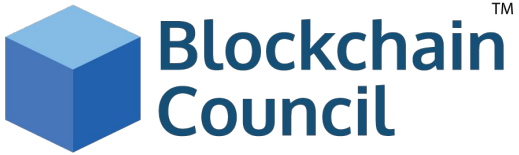

Certified Hyperledger Developer

Hyperledger Composer Modeling Language

Modeling Language

- Hyperledger Composer use an object-oriented modeling language which is used to define the model of the business network.
- A Hyperledger Composer's model is written in a CTO file which has the following elements:
    - System namespace
    - Resource definitions
    - Import declarations

Model's Namespace

- A namespace is defined in the first line of the CTO file with a unique name
- All resources that are created in the model file are implicitly part of this namespace
- A system namespace is also defined which contains the base definitions of asset, event, participant, and transaction.

Resource definitions

- Resources in Hyperledger Composer include:
  - Assets, Participants, Transactions, and Events.
  - Enumerated Types.
  - Concepts.
- A resource definition has the following properties: a name, an optional super-type, an optional abstract declaration, set of named properties and set of relationships.

Data types

- **Enumerated types:** These are used to specify a type that may have 1 or N possible values.
  - Ex: enum ProductType {
    o Plants
    o Animal
    }

- **Primitive types:** Composer resources are defined in terms of the following primitive types: String, Double, Integer, Long, DateTime and Boolean.
  - Ex. o String fieldId

- **Arrays:** All types in Composer may be declared as arrays using the [] notation.
  - Ex. Integer[] integerArray

Data types

- **Concepts:** Concepts are abstract classes that are not assets, participants or transactions. They are typically contained by an asset, participant or transaction.
    - Ex. abstract concept Address {
      }
      concept StateAddress extends Address {
      }

- **Relationships:** A relationship in the Composer language is a tuple composed of: the namespace of the type being referenced, the type name of the type being referenced, the identifier of the instance being referenced.

    - Ex. org.example.Image#1234

Import declarations

- **Imports:** Import keyword is used with a fully-qualified type name to import a type from another namespace. Alternatively, use the .* notation to import all the types from another namespace.
    - **Ex. import org.example.MyAsset**
      import org.example2.*

THANK YOU!

Any questions?

You can mail us at

hello@blockchain-council.org