Hyperledger Composer architecture

Written by Simon Stone,
Hyperledger Composer Maintainer

Presenters:


Jennifer Foley


Austin Grice


Barry Silliman

Hyperledger Composer

- Hyperledger Composer is a framework to accelerate the development of applications built on top of Blockchain platforms:
  - Start from the business level; model network **assets**, **participants**, and **transactions**
  - Applications use business centric APIs to invoke transactions that create, delete, and update assets and transfer them between participants
  - Assets, participants, and transactions are recorded in the world state in **registries**
  - Easily **integrate** Blockchain with existing business processes and systems of record
  - Emphasis on quick solution creation and business-centric vocabulary

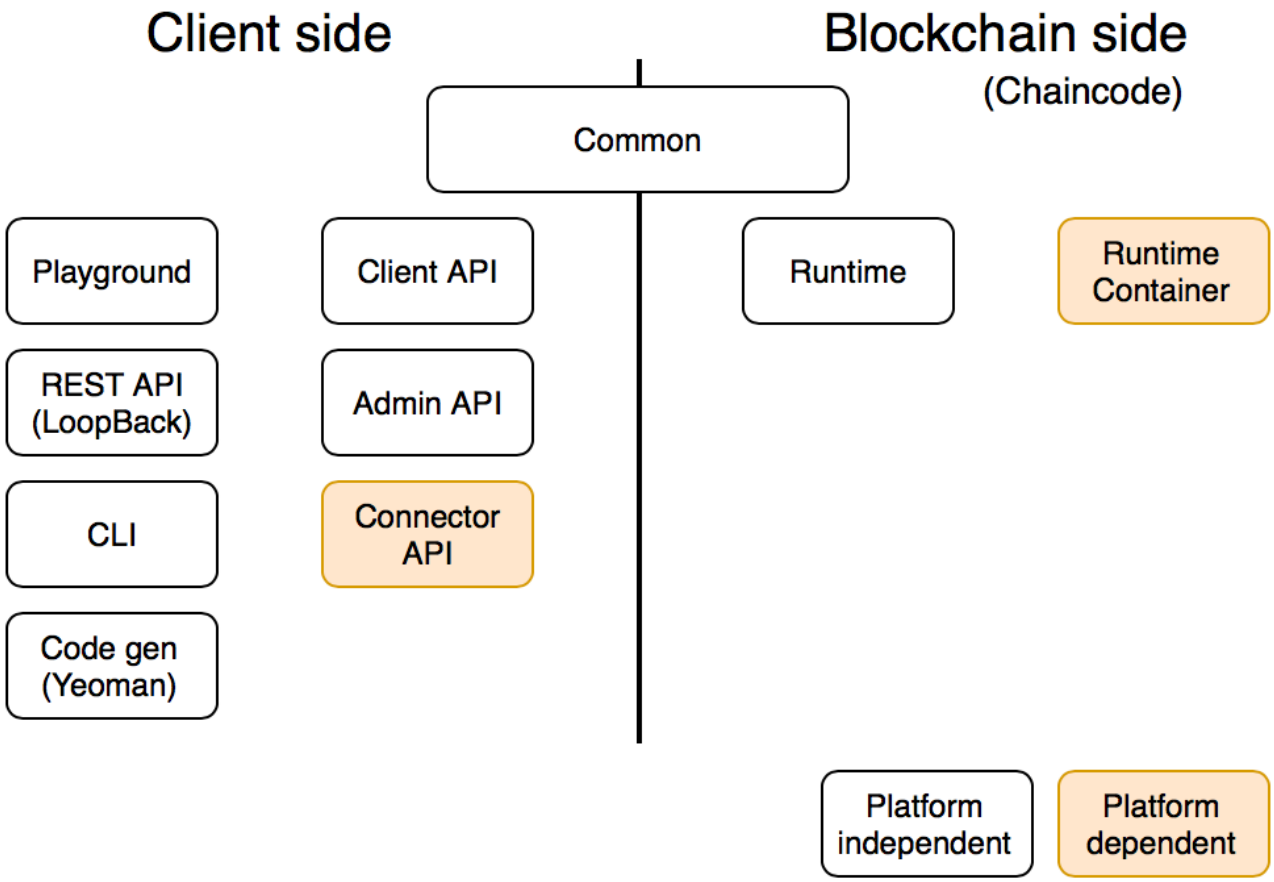
Business network definition

- A **business network definition** is the collection of user developed source artefacts that describe the resources and logic in a business network:
  - Model files describe assets, participants, and transactions.
  - Access control lists define rules for sharing and privacy.
  - Transaction processors implement additional business requirements.
  - The business network has a name (org.acme.biznet) and a version number (1.0.2).
  - Can be packaged into a **business network archive**, or a banana (.bna) file.

Deployment to a Blockchain platform

- The business network definition can then be deployed to a Blockchain platform along with the Hyperledger Composer **runtime**.
  - The **runtime** is the generic chaincode/smart contract supplied as part of Hyperledger Composer that hosts and understands the business network.
  - There is **no** chaincode/smart contract code generation at work here!
  - The runtime exposes operations on the deployed business network to **client** applications via a set of APIs.
- The majority of the runtime code is Blockchain platform independent, and can be run anywhere that can host a JavaScript virtual machine:
  - Currently only Hyperledger Fabric v0.6 and v1.0 are supported.
  - Can also be run in a web browser or in Node.js for development/test purposes.

Multiple components



Client side components

- The majority of the components are client side components, and provide functionality for developing solutions with Hyperledger Composer:
  - Playground for developing and testing business networks from a browser.
  - Client/Admin APIs
  - Command line interface
  - REST API generation using LoopBack (http://loopback.io)
  - Code generation using Yeoman (http://yeoman.io)
  - Editor plugins for Atom (http://atom.io) and VS Code (https://code.visualstudio.com)
- These components make calls to the Blockchain platform in order to interact with the deployed business network (by calling the generic chaincode/smart contract).

Blockchain side components

- The rest of the components are Blockchain side components, and provide functionality for running a deployed business network:
  - Persistence of resources (assets, participants, transactions) into registries, which are backed by the underlying world state provided by the Blockchain platform.
  - Access control enforcement by using the identity (certificate) of the participant who submitted the request.
  - Execution of user developed transaction processor functions, and the publishing and recording of any business events.
- These components expose a set of APIs that can be called by client components to interact with the deployed business network (by calling the generic chaincode/smart contract).

composer-common

- JavaScript module used by all other modules:
  - https://github.com/hyperledger/composer/tree/master/packages/composer-common
- Provides:
  - Logging APIs used by the rest of Hyperledger Composer.
  - Parsing and validating APIs for the parts of a business network definition – model files, access control lists, and transaction processor functions.
  - APIs for creating and loading business network definitions/archives.
  - Connection profile manager API, for describing connections to Blockchain platforms.
  - Connector API, for building connectors which connect to Blockchain platforms.
  - Code generation for JSON Schemas, LoopBack models, UML diagrams, etc.

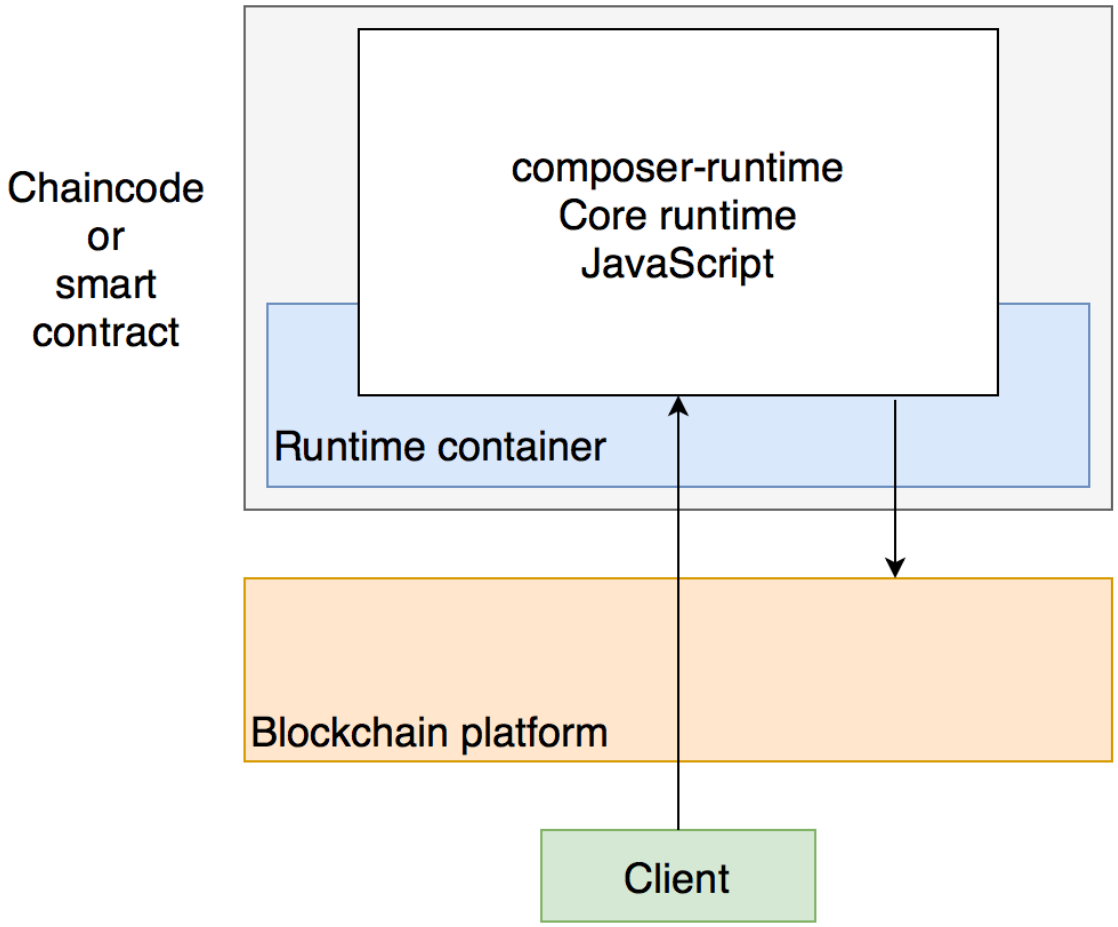
composer-runtime
Core runtime

- Blockchain platform independent JavaScript module:
  - https://github.com/hyperledger/composer/tree/master/packages/composer-runtime
- Provides:
  - Management of the deployed business network.
  - Persistence of resources (assets, participants, transactions) into registries.
  - Access control enforcement.
  - Execution of user developed transaction processor functions.
  - APIs for exposing the deployed business network.
- The runtime must be hosted in a **runtime container** that provides a set of services which bind the runtime code to the underlying Blockchain platform.

composer-runtime-<platform>
Runtime container implementations

- Blockchain platform specific module:
  - https://github.com/hyperledger/composer/tree/master/packages/composer-runtime-hlfv1
  - https://github.com/hyperledger/composer/tree/master/packages/composer-runtime-web
- This is the chaincode/smart contract implementation, that provides a platform specific set of services to the platform independent runtime code:
  - Loading and execution of JavaScript core runtime.
  - Routing of API calls from client into core runtime.
  - Logging
  - Data persistence using the world state.
  - Identifying the participant/certificate used to submit the transaction.
- The runtime container can be written in any language, for example Golang.

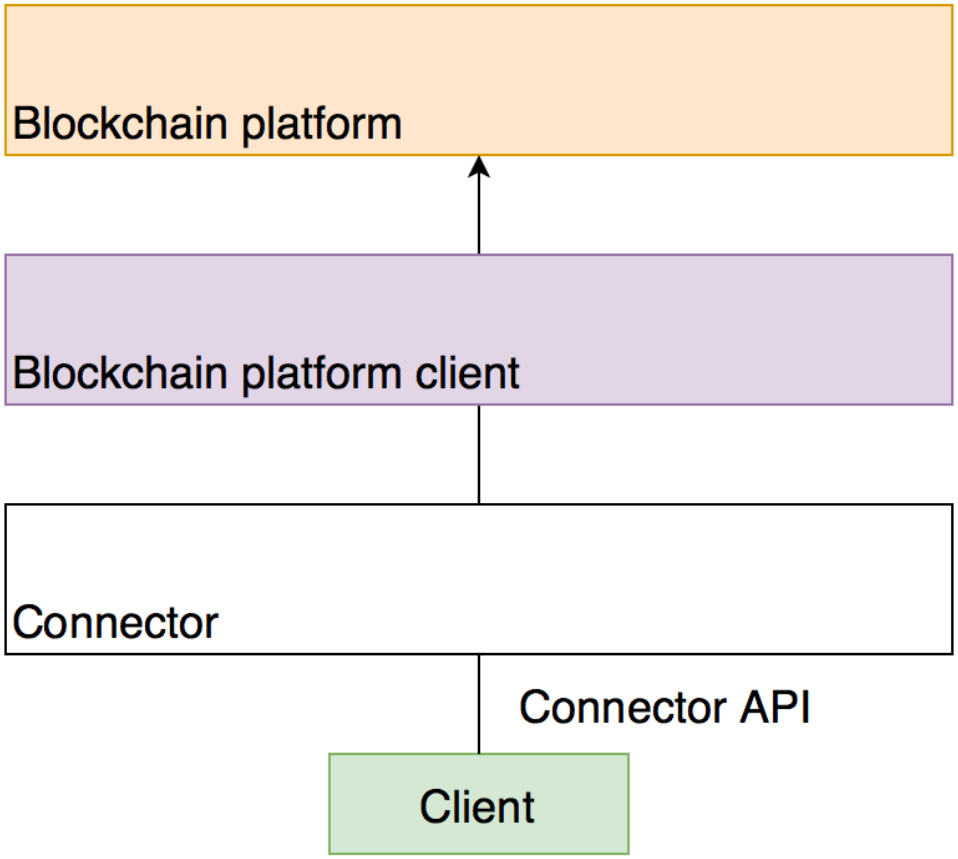
Runtime, runtime container, and Blockchain platform



composer-connector-<platform>
Connector implementations

- Blockchain platform specific module:
  - https://github.com/hyperledger/composer/tree/master/packages/composer-connector-hlfv1
  - https://github.com/hyperledger/composer/tree/master/packages/composer-connector-web
- Provides clients with a standardized interface (the connector API) that they can use to interact with a business network, regardless of the a Blockchain platform.
- Wraps the underlying Blockchain platform client.
- Connector implementation selected by connection profile type, and can be dynamically loaded at runtime.

Connector and Blockchain platform



Connection profile

- Connection profile determines which connector implementation to use:

```
{
    "type": "hlfv1",
    "orderers": [
        {
            "url": "grpcs://ldn1-zbc6a.2.secure.blockchain.ibm.com:21505",
        },
        {
            "url": "grpcs://ldn1-zbc6b.2.secure.blockchain.ibm.com:21505",
        },
        {
            "url": "grpcs://ldn1-zbc6c.2.secure.blockchain.ibm.com:21505",
        }
    ],
    "peers": [
        {
```

Type used to select connector implementation

Rest of properties are handed to connector at connect

composer-client
For building client applications

- Blockchain platform independent JavaScript module:
  - https://github.com/hyperledger/composer/tree/master/packages/composer-client
- Uses a connector to interact with the Blockchain platform and send requests to the deployed business network.
- Provides APIs for working with a deployed business network:
  - CRUD APIs for assets, participants, and registries.
  - Transaction submission.
  - Issuing and revoking identities.
- Performs client side data validation before serializing requests and sending them to the runtime running on the target Blockchain platform.

composer-admin

For building administrative or operational applications

- Blockchain platform independent JavaScript module:
  - https://github.com/hyperledger/composer/tree/master/packages/composer-admin
- Uses a connector to interact with the Blockchain platform and send requests to the deployed business network.
- Provides APIs for managing business networks:
  - First time deployment to the Blockchain platform (deploying the chaincode/smart contract).
  - Updating of a deployed business network definition.
  - Upgrading of the core runtime code (upgrading the chaincode/smart contract).
  - Undeploying a deployed business network.

composer-cli
For automation and scripting

- Blockchain platform independent JavaScript module:
  - https://github.com/hyperledger/composer/tree/master/packages/composer-cli
- Install with npm install –g composer-cli
- Provides a CLI application composer that exposes functionality in the composer-admin and composer-client modules to scripting languages:
  - Deploy, update, upgrade, and undeploy business networks.
  - CRUD operations for assets, participants, and registries.
  - Submit transactions.
  - Issue and revoke identities.

composer-playground

For developing and testing business networks in a browser

- Blockchain platform independent JavaScript module:
  - https://github.com/hyperledger/composer/tree/master/packages/composer-playground
- Install with npm install –g composer-playground or Docker!
- Public playground hosted online: http://composer-playground.mybluemix.net
- Built in Angular 2 and TypeScript.
- Uses the standard composer-client and composer-admin APIs.
- Uses the “web” connector and runtime container to simulate a Blockchain platform running in the web browser.
- Can use any connector implementation to work with business networks on a “real” Blockchain platform.

loopback-connector-composer

For building REST APIs for a business network

- Blockchain platform independent JavaScript module:
  - https://github.com/hyperledger/composer/tree/master/packages/loopback-connector-composer
- LoopBack (http://loopback.io) is a framework for exposing backend systems such as databases via REST API.
- The Hyperledger Composer LoopBack connector exposes a deployed business network to LoopBack so it can generate a REST API for the assets, participants, and transactions in that business network.
- The LoopBack connector uses the composer-client APIs.
- The composer-rest-server module provides an easy to use CLI application for users who don't need to understand LoopBack to create a REST API.

generator-fabric-composer (needs new name!)
For generating skeleton applications

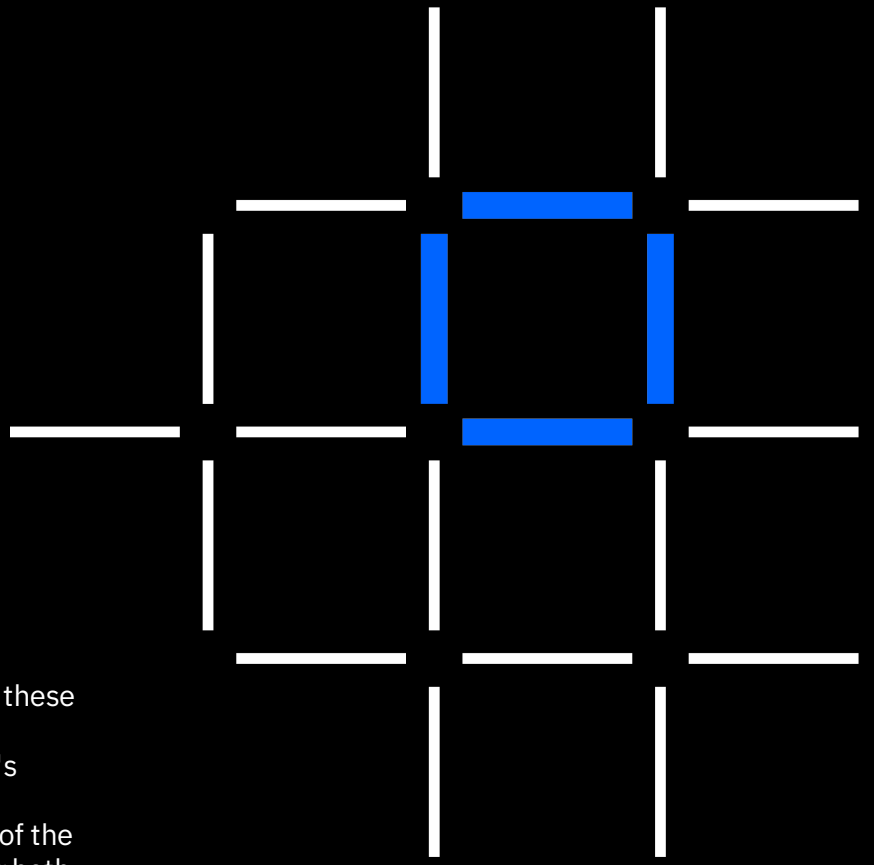
- Blockchain platform independent JavaScript module:
  - https://github.com/hyperledger/composer/tree/master/packages/generator-fabric-composer
- Yeoman (http://yeoman.io) is a framework for generating skeleton applications for use by developers as starting points.
- The Hyperledger Composer Yeoman generator can generate a CLI or Angular 2 application that demonstrates to the developer how they can interact with a deployed business network.
- Can generate an application from a deployed business network or business network archive.
- Easy to plug in generators for other application frameworks, such as React.

Thank You

Jennifer Foley
foleyje@us.ibm.com

Barry Silliman
silliman@us.ibm.com

Austin Grice
austin.grice@ibm.com



IBM