# Blockchain Education

IBM **Blockchain**

*Presenters:*

Jennifer Foley

Austin Grice

Barry Silliman

# 3-Day Education Agenda

- Day 1
  - Blockchain Explained
  - Hyperledger Fabric Lab
- Day 2
  - Blockchain Composed
  - Hyperledger Composer Lab
- Day 3
  - Blockchain Security Considerations
  - Making Blockchain Real and Lab

# Contents

## What is Hyperledger Composer

## Application Development

*Writing the application*

*Modeling the business network*

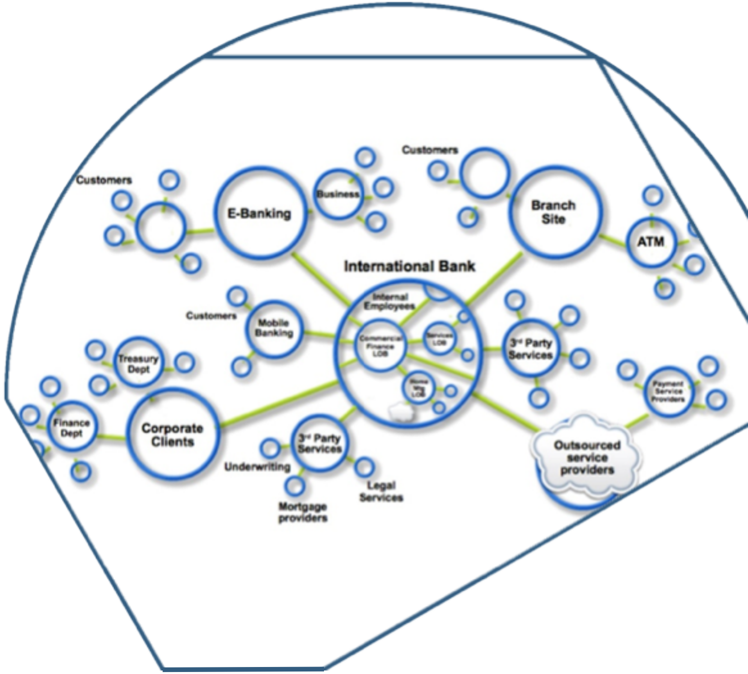
## Effective Administration

*Deploying to a blockchain*

*Interacting with systems of record*

# Blockchain Recap

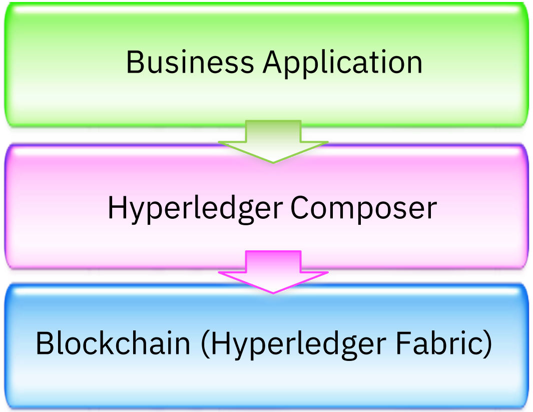
- Blockchain builds on basic business concepts
  - **Business Networks** connect businesses
  - **Participants** with Identity
  - **Assets** flow over business networks
  - **Transactions** describe asset exchange
  - **Contracts** underpin transactions
  - The **ledger** is a log of transactions
- Blockchain is a shared, replicated ledger
  - Consensus, immutability, finality, provenance

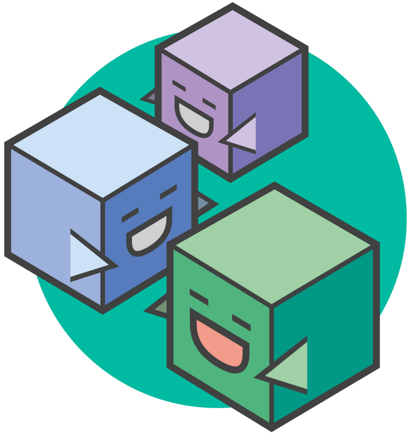


# Hyperledger Composer: Accelerating time to value

https://hyperledger.github.io/composer/

- A suite of high level application **abstractions** for business networks
- Emphasis on business-centric vocabulary for quick solution creation
- Reduce risk, and increase understanding and flexibility





Features

- Model your business networks, test and expose via APIs
- Applications invoke APIs transactions to interact with business network
- Integrate existing systems of record using loopback/REST

- Fully open and part of Linux Foundation Hyperledger

- Try it in your web browser now: http://composer-playground.mybluemix.net/

# Benefits of Hyperledger Composer

### Increases understanding

Bridges simply from business concepts to blockchain

### Saves time

Develop blockchain applications more quickly and cheaply

### Reduces risk

Well tested, efficient design conforms to best practice

### Increases flexibility

Higher level abstraction makes it easier to iterate

# User Roles in a Blockchain Project

D

### Application Developer

- Developing the application that interacts with the ledger
- Modelling the business network
- Implementing the script files that define transaction behaviour

A

### Solution Administrator

- Provisioning the target environment
- Deploying the business application
- Managing the blockchain

P

### Business Network Participant

- Running an end-user application that invokes transactions
- Aware of business concepts: assets, participants and transactions
- May not be aware of blockchain underpinnings

# Contents

What is Hyperledger Composer

Application Development

*Writing the application*

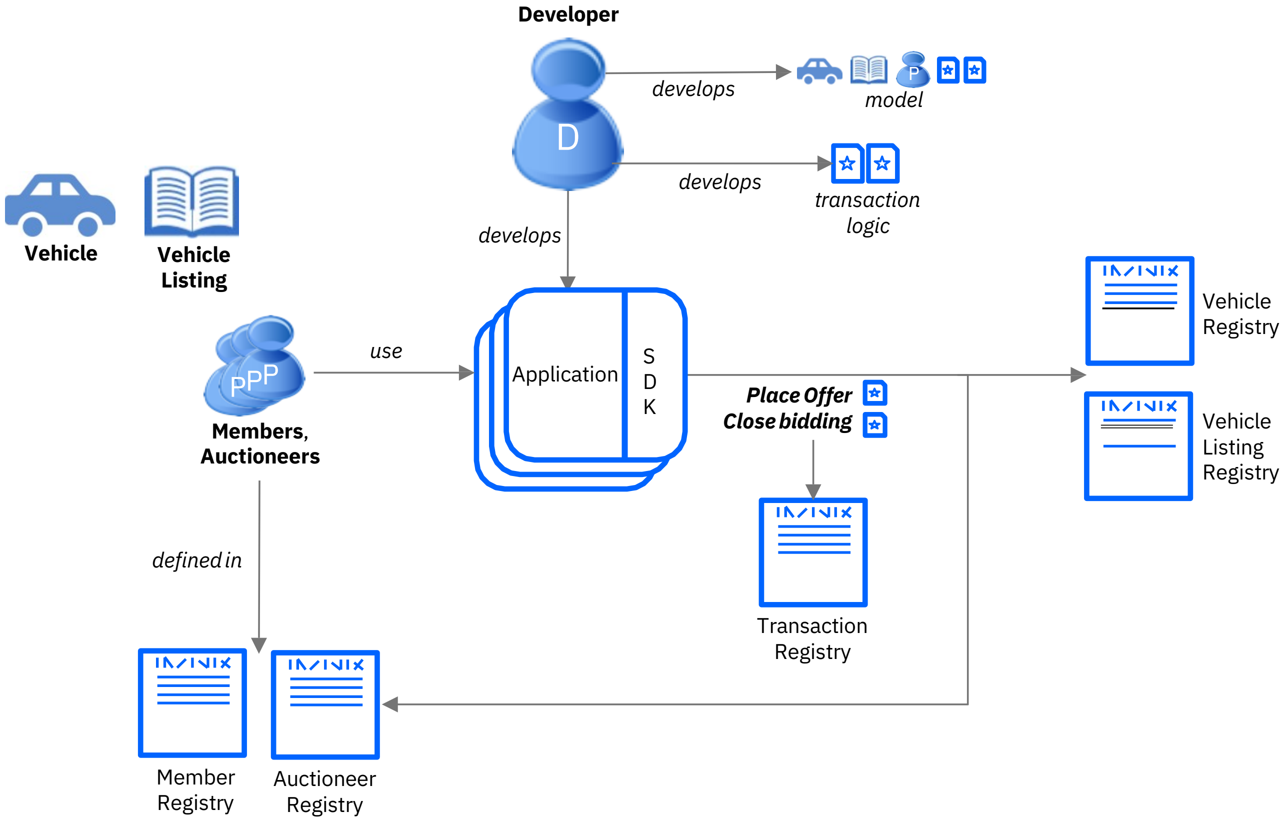
*Modeling the business network*

Effective Administration

*Deploying to a blockchain*

*Interacting with systems of record*

# Example: Vehicle Auction



# Developer Concepts

## Applications

- Provides front-end for the user
  - May require different applications per participant
- Interacts with the registries
  - Add, delete, update, query
  - Registries persisted on blockchain
- Connects to blockchain via JavaScript client libraries (SDK) or REST

## Models

- A domain specific language (.CTO) that defines the type structure of
  - Assets
  - Participants
  - Transactions
- Aims to match how we talk about business networks in the real world

## Scripts

- Scripts provide the implementation of transaction processor logic
- Specified in Javascript
- Designed for any reasonable Javascript developer to pick up easily

# Vehicle Auction

Applications

Models

Scripts

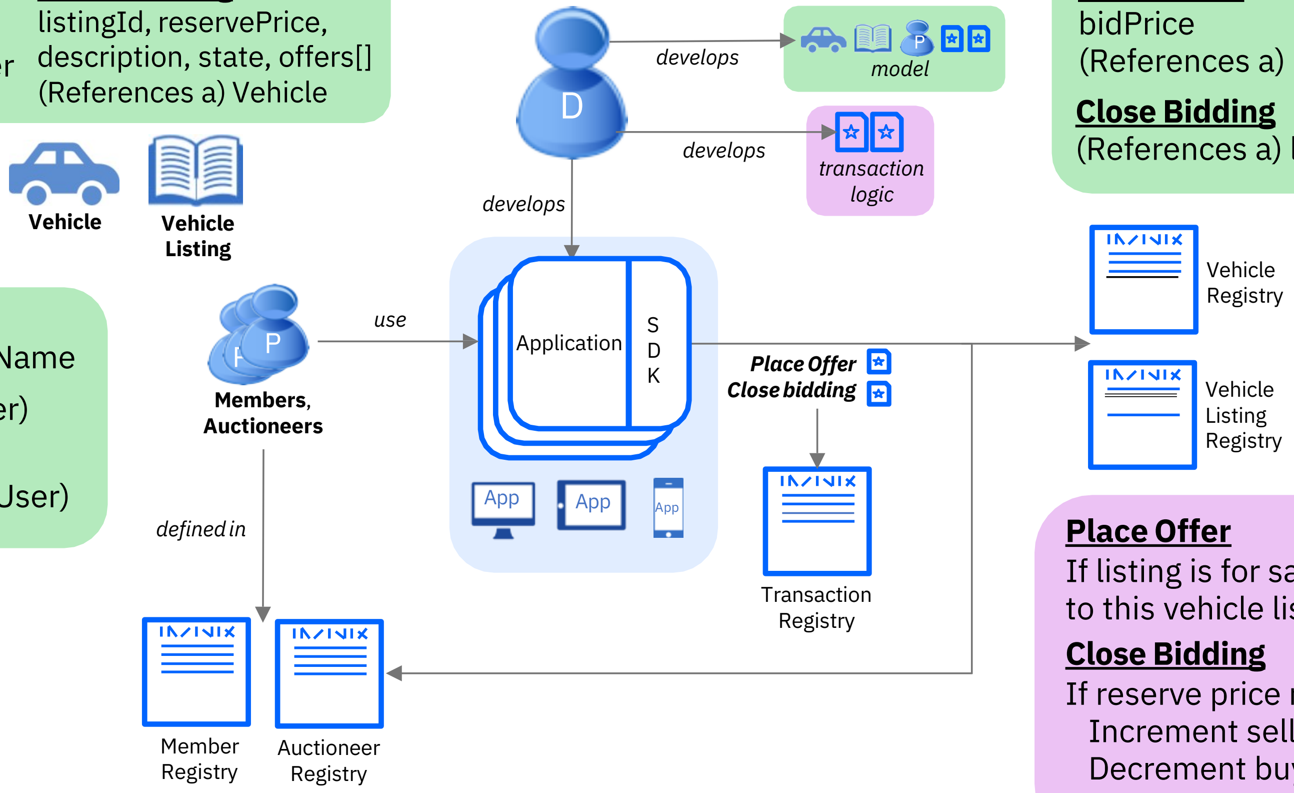
**Vehicle**
VIN
(References a) Member

**Vehicle Listing**
listingId, reservePrice, description, state, offers[]
(References a) Vehicle

**User**
email, firstName, lastName

**Member** (extends User)
balance

**Auctioneer** (extends User)



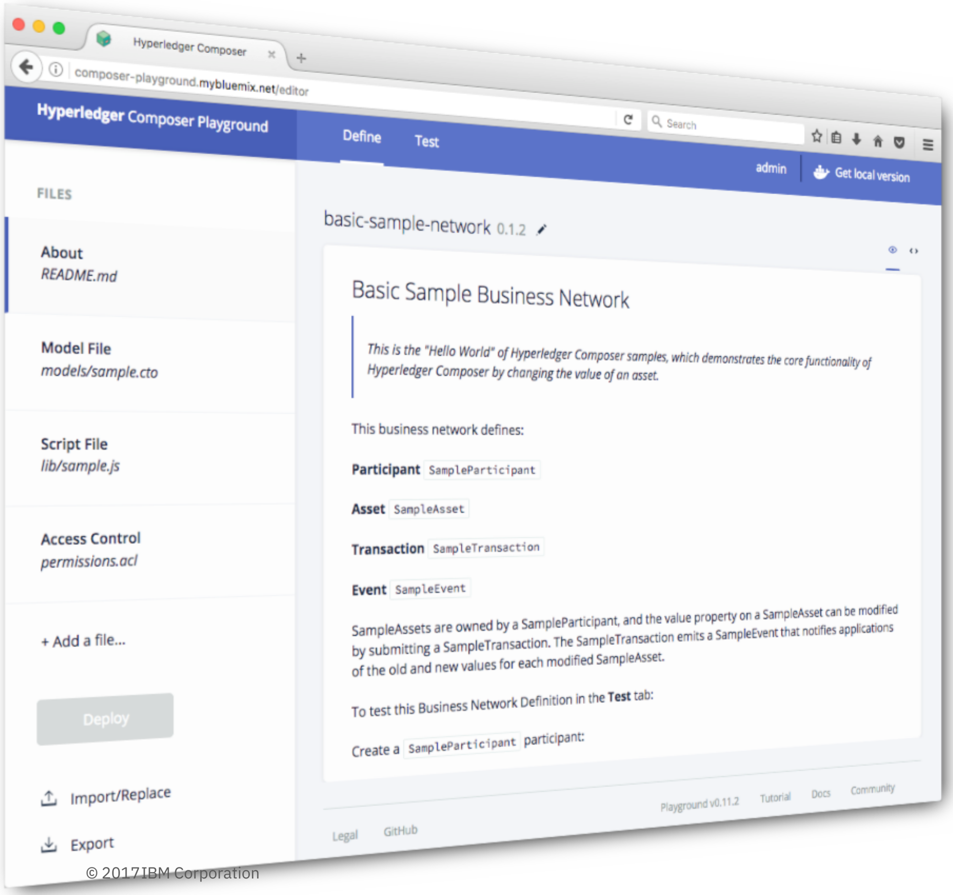
**Place Offer**
bidPrice
(References a) listing, member

**Close Bidding**
(References a) listing

**Place Offer**
If listing is for sale, add offer to this vehicle listing's offers[]

**Close Bidding**
If reserve price met
  Increment seller's balance
  Decrement buyer's balance
  Change owner to buyer

# Tools: Composer Playground



basic-sample-network 0.1.2

## Basic Sample Business Network

> *This is the "Hello World" of Hyperledger Composer samples, which demonstrates the core functionality of Hyperledger Composer by changing the value of an asset.*

This business network defines:

**Participant** `SampleParticipant`

**Asset** `SampleAsset`

**Transaction** `SampleTransaction`

**Event** `SampleEvent`

SampleAssets are owned by a SampleParticipant, and the value property on a SampleAsset can be modified by submitting a SampleTransaction. The SampleTransaction emits a SampleEvent that notifies applications of the old and new values for each modified SampleAsset.

To test this Business Network Definition in the **Test** tab:

Create a `SampleParticipant` participant:

- Web tool for defining and testing Hyperledger Composer models and scripts

- Designed for the application developer
  - Define assets, participants and transactions
  - Implement transaction processor scripts
  - Test by populating registries and invoking transactions

- Deploy to instances of Hyperledger Fabric V1, or simulate completely within browser

- Install on your machine or run online at [http://composer-playground.mybluemix.net](http://composer-playground.mybluemix.net)

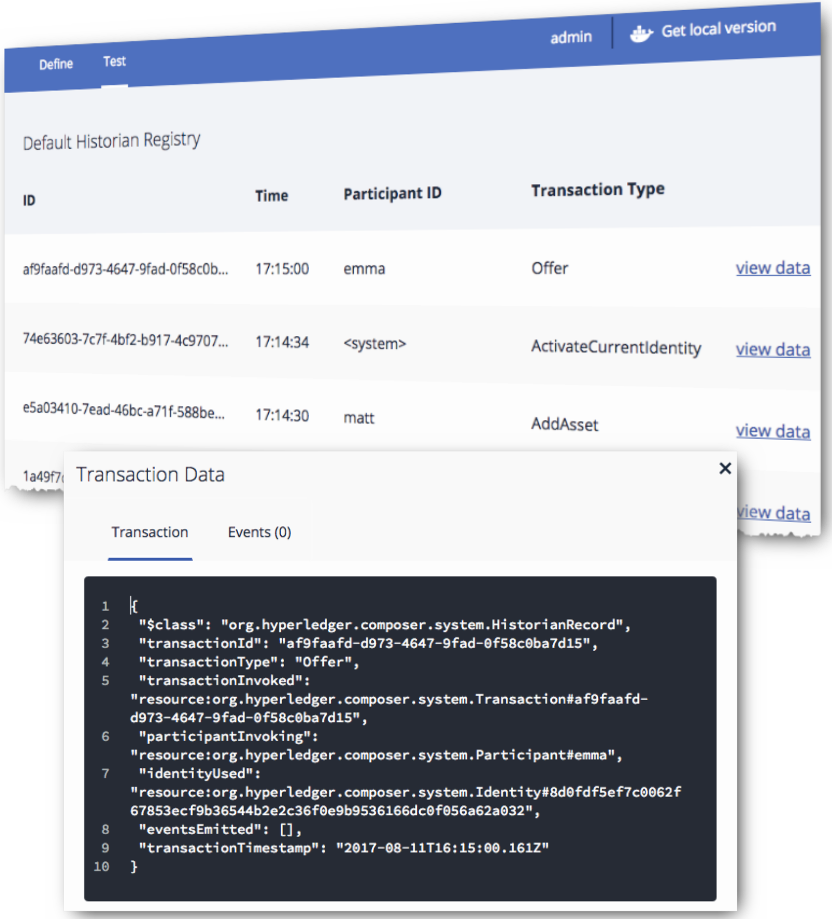
# Historian

- Playground Historian allows you to view all transactions
  - See what occurred and when

- Diagnostics framework allows for application level trace
  - Uses the Node.js logging framework

- More information online:

  https://hyperledger.github.io/composer/problems/diagnostics.html



Default Historian Registry

| ID | Time | Participant ID | Transaction Type | |
|---|---|---|---|---|
| af9faafd-d973-4647-9fad-0f58c0b... | 17:15:00 | emma | Offer | view data |
| 74e63603-7c7f-4bf2-b917-4c9707... | 17:14:34 | <system> | ActivateCurrentIdentity | view data |
| e5a03410-7ead-46bc-a71f-588be... | 17:14:30 | matt | AddAsset | view data |
| 1a49f7 | | | | view data |

Transaction Data

```
{
 "$class": "org.hyperledger.composer.system.HistorianRecord",
 "transactionId": "af9faafd-d973-4647-9fad-0f58c0ba7d15",
 "transactionType": "Offer",
 "transactionInvoked":
"resource:org.hyperledger.composer.system.Transaction#af9faafd-
d973-4647-9fad-0f58c0ba7d15",
 "participantInvoking":
"resource:org.hyperledger.composer.system.Participant#emma",
 "identityUsed":
"resource:org.hyperledger.composer.system.Identity#8d0fdf5ef7c0062f
67853ecf9b36544b2e2c36f0e9b9536166dc0f056a62a032",
 "eventsEmitted": [],
 "transactionTimestamp": "2017-08-11T16:15:00.161Z"
}
```

# Contents

What is Hyperledger Composer

Application Development

*Writing the application*

*Modeling the business network*

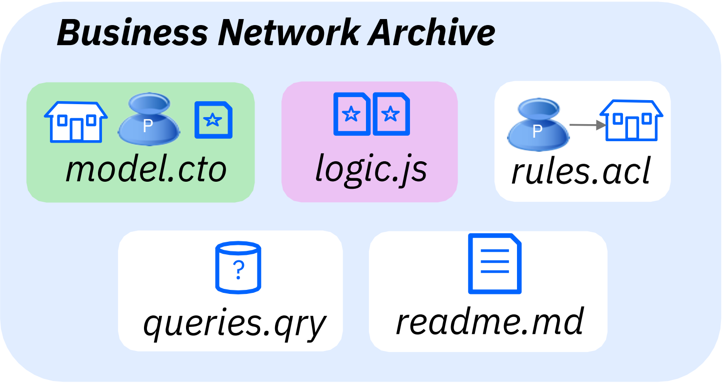
Effective Administration

*Deploying to a blockchain*

*Interacting with systems of record*

# Resources are packaged into BNA files

- Business Network Archive (.BNA) is a package of the resources used by Fabric:
  - Model files (.CTO)
  - Transaction processors (.JS)
  - Access Control Lists (.ACL)
  - Static queries (.QRY)
  - Documentation and versioning (.MD)
  - It does *not* contain the client application

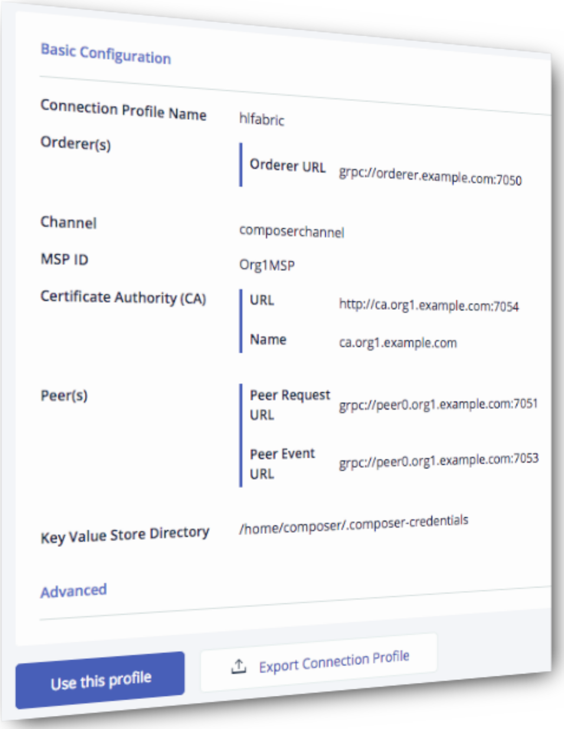


- The BNA simplifies deployment of blockchain and promotion between environments

```
composer archive create –archiveFile my.bna
      --sourceType module --sourceName myNetwork
```

- Create BNA files from Playground or command line

# Deployment to Hyperledger Fabric



Basic Configuration

| | | |
|---|---|---|
| Connection Profile Name | hlfabric | |
| Orderer(s) | Orderer URL | grpc://orderer.example.com:7050 |
| Channel | composerchannel | |
| MSP ID | Org1MSP | |
| Certificate Authority (CA) | URL | http://ca.org1.example.com:7054 |
| | Name | ca.org1.example.com |
| Peer(s) | Peer Request URL | grpc://peer0.org1.example.com:7051 |
| | Peer Event URL | grpc://peer0.org1.example.com:7053 |
| Key Value Store Directory | /home/composer/.composer-credentials | |

Advanced

- Command line tool to script deployment

```
composer network deploy -p myProfile -a my.bna -i user -s secret
```

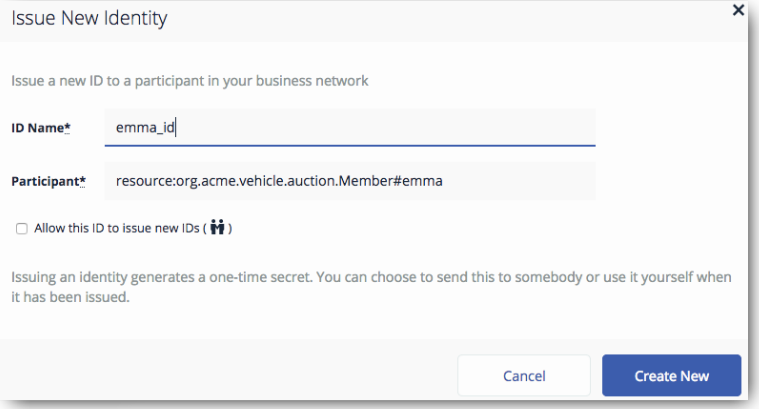
- Use Connection profiles to describe Fabric connection parameters
  - Export from Playground, generate from script or create by hand
- Enrollment  in Hyperledger Fabric network required
  - Issue Fabric identity from Composer participants
- Additional command line options for management of business network
  - For example: download, list, start, undeploy, upgrade…

# Participant Identity

- Participants require an *identity* in order to connect to Hyperledger Fabric
  - Issued by the administrator as a Hyperledger Fabric userid/secret
  - Supplied by the participant when the client application connects

- Composer Participant to Fabric Identity mapping is stored on the blockchain in an *identity registry*

- Perform identity management from Playground, Javascript, REST or command line
  - For example: Test connection, issue identity, bind an identity to a participant, revoke an identity, list identities



Issue New Identity

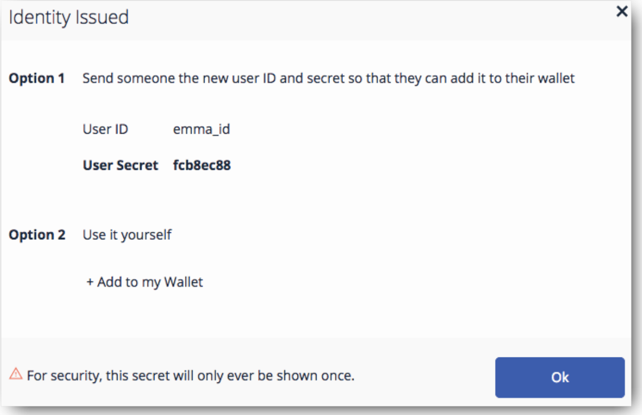
Issue a new ID to a participant in your business network

ID Name* emma_id

Participant* resource:org.acme.vehicle.auction.Member#emma

□ Allow this ID to issue new IDs ( )

Issuing an identity generates a one-time secret. You can choose to send this to somebody or use it yourself when it has been issued.



Identity Issued

**Option 1** Send someone the new user ID and secret so that they can add it to their wallet

User ID emma_id

**User Secret** **fcb8ec88**

**Option 2** Use it yourself

+ Add to my Wallet

⚠ For security, this secret will only ever be shown once.

```
businessNetworkConnection.connect
      ('hlfv1', 'my-network', 'emma_id', 'fcb8ec88')
```

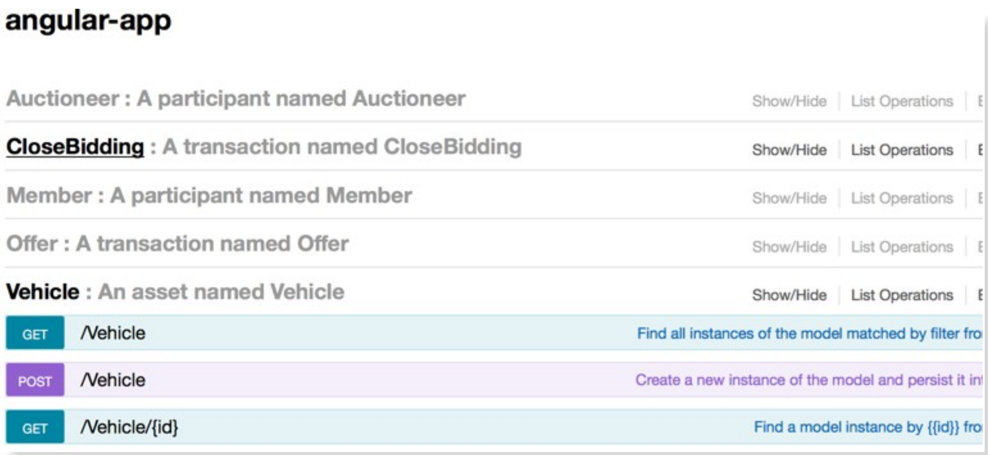
# Systems of Record Integration

- Domain specific APIs very attractive to mobile and web developers. Resources and operations are business-meaningful

- Composer exploits Loopback framework to create REST APIs: https://loopback.io/

- Extensive test facilities for REST methods using loopback

- Composer provides back-end integration with any loopback compatible product
  - e.g. IBM Integration Bus, API Connect, StrongLoop
  - Outbound and Inbound (where supported by middleware)



**angular-app**

Auctioneer : A participant named Auctioneer — Show/Hide | List Operations | E

CloseBidding : A transaction named CloseBidding — Show/Hide | List Operations | E

Member : A participant named Member — Show/Hide | List Operations | E

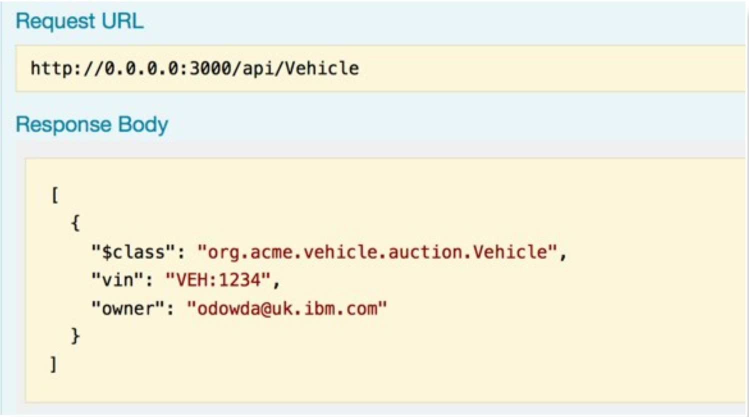
Offer : A transaction named Offer — Show/Hide | List Operations | E

Vehicle : An asset named Vehicle — Show/Hide | List Operations | E

GET /Vehicle Find all instances of the model matched by filter fro

POST /Vehicle Create a new instance of the model and persist it in

GET /Vehicle/{id} Find a model instance by {{id}} fro



Request URL

```
http://0.0.0.0:3000/api/Vehicle
```

Response Body

```
[
  {
    "$class": "org.acme.vehicle.auction.Vehicle",
    "vin": "VEH:1234",
    "owner": "odowda@uk.ibm.com"
  }
]
```

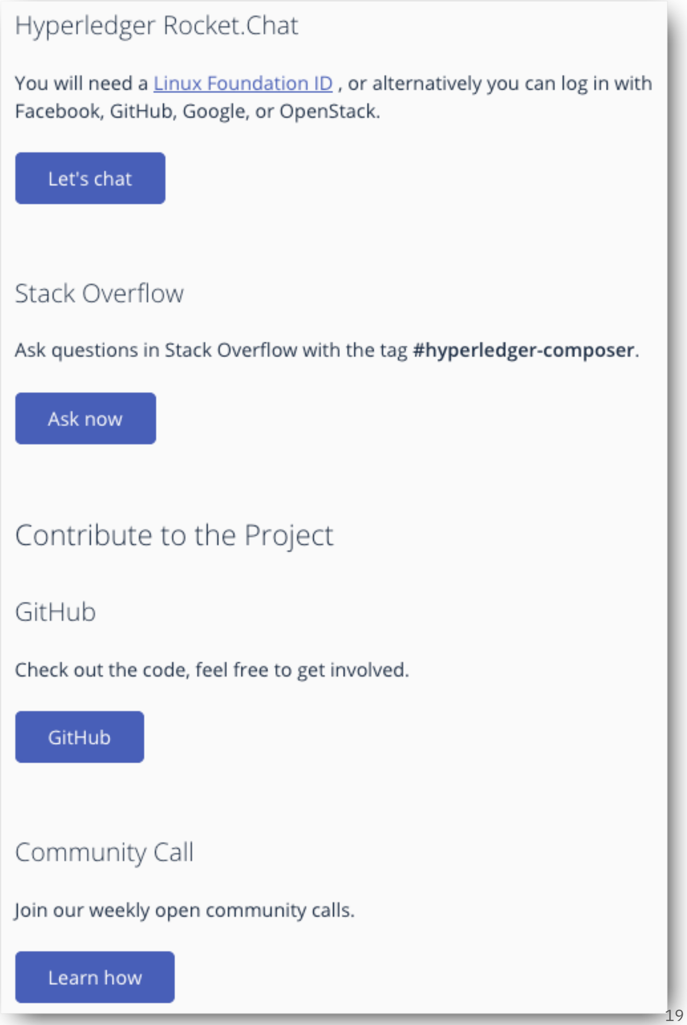
# Hyperledger Composer Outlook

- Still early in product lifecycle

- Lots of improvements planned
  - See https://github.com/hyperledger/composer/issues

- An active development community
  - Open community calls every two weeks
  - Rocket Chat
  - Stack Overflow

- Get involved!



Hyperledger Rocket.Chat

You will need a Linux Foundation ID , or alternatively you can log in with Facebook, GitHub, Google, or OpenStack.

Stack Overflow

Ask questions in Stack Overflow with the tag **#hyperledger-composer**.

Contribute to the Project

GitHub

Check out the code, feel free to get involved.

Community Call

Join our weekly open community calls.

# Get started with Hyperledger Composer!

- Define, Test and Deploy Business Networks
- Create domain APIs and sample applications
- Integrate existing systems and data

[https://hyperledger.github.io/composer/](https://hyperledger.github.io/composer/)

[http://composer-playground.mybluemix.net/](http://composer-playground.mybluemix.net/)

# Benefits of Hyperledger Composer

**Increases understanding**

Bridges simply from business concepts to blockchain

**Saves time**

Develop blockchain applications more quickly and cheaply

**Reduces risk**

Well tested, efficient design conforms to best practice

**Increases flexibility**

Higher level abstraction makes it easier to iterate

# Thank You

*Jennifer Foley*
*foleyje@us.ibm.com*

*Barry Silliman*
*silliman@us.ibm.com*

*Austin Grice*
*austin.grice@ibm.com*

IBM