

# BLOCKCHAINS

## ARCHITECTURE, DESIGN AND USE CASES

**SANDIP CHAKRABORTY**

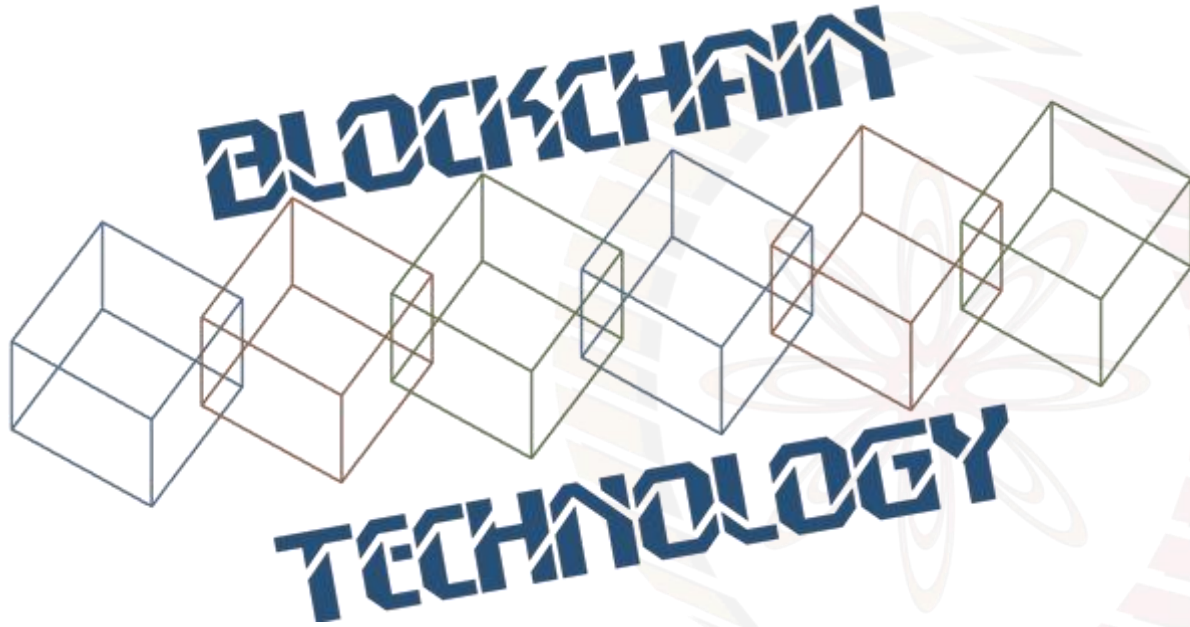
**COMPUTER SCIENCE AND ENGINEERING,  
IIT KHARAGPUR**

**PRAVEEN JAYACHANDRAN**

**IBM RESEARCH,  
INDIA**



Image courtesy: <http://beetfusion.com/>



## **HYPERLEDGER COMPOSER – NETWORK ADMINISTRATION**

# Two Roles with “Administration” Responsibility



## – Network Service Provider

- **Governs** the network: channels, membership etc.
- A consortium of network members or designated authority



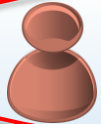
## – Network Service Consumer

- **Operates** a set of peers and certificate authorities on the network
- Represents an organization on the business network



## – Business Service Provider

- **Develops** blockchain business applications
- Includes transaction, app server, integration and presentation logic



## – Business Service Consumer

- Hosts application and integration logic which invokes blockchain transactions

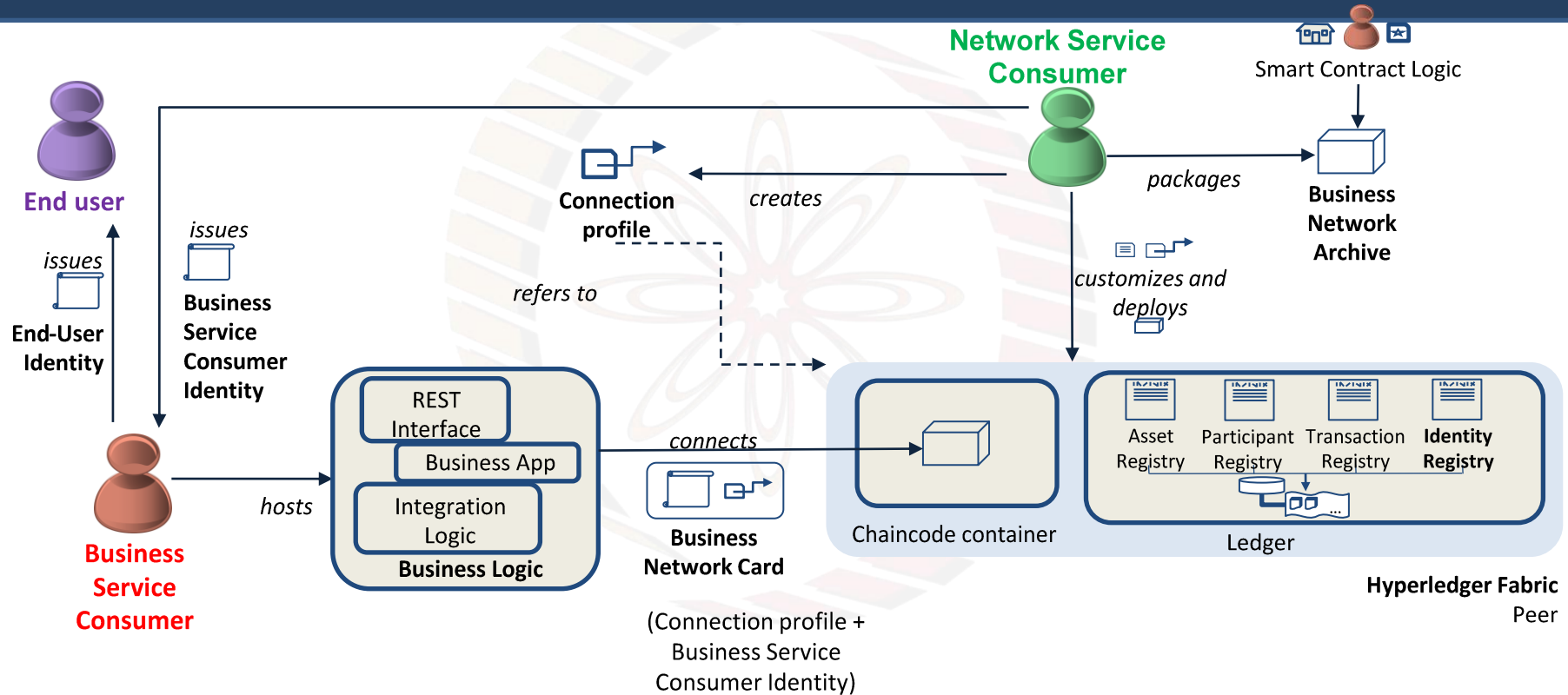


## – End-user

- Runs presentation logic e.g. on mobile device or dashboard

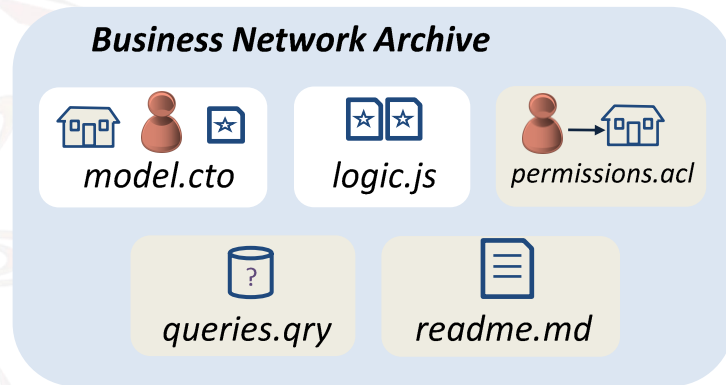
A single organization may play multiple roles!

# Key Concepts for Administrators



# Network Service Consumer packages resources in a BNA file

- 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
  - c.f. TAR, WAR, EAR, JAR, BAR...
- Create BNA files from Playground or command line
  - Build from filesystem or NPM module



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

# Connection Profiles to Hyperledger Fabric

Basic Configuration

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

Advanced

Use this profile [Export Connection Profile](#)

- Use **connection profiles** to describe Fabric connection parameters
  - One connection profile required per channel
  - Not necessary for web-based simulation
- Enrollment in Hyperledger Fabric network required (see later)
  - Issue Fabric identity from Composer participants

```
1 {
2   "type": "hlcfv1",
3   "orderers": [
4     { "url": "grpc://localhost:7050" }
5   ],
6   "ca": { "url": "http://localhost:7054",
7           "name": "ca.org1.example.com"
8         },
9   "peers": [
10    {
11      "requestURL": "grpc://localhost:7051",
12      "eventURL": "grpc://localhost:7053"
13    }
14  ],
15   "keyValStore": "${HOME}/.composer-credentials",
16   "channel": "composerchannel",
17   "mspID": "Org1MSP",
18   "timeout": "300"
19 }
20
```

- Connection profiles currently used by Composer only
  - Plans to implement common connection profiles that can be used by both Fabric and Composer

# Participant Identity

- The **Network Service Consumer** issues network participants with an **identity** in order to connect to Hyperledger Fabric
  - Issued as a Hyperledger Fabric userid/secret
  - Automatically swapped for a certificate on first use
  - Packaged in a Business Network Card and supplied when the client application connects
- Composer Participant to Fabric Identity mapping is stored on the blockchain in an *identity registry*
- Usually, only **Business Service Consumers** have a Fabric identity
  - **End-users** log in to the business application using a separately managed identity; blockchain transactions invoked by proxy
- Manage identity 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

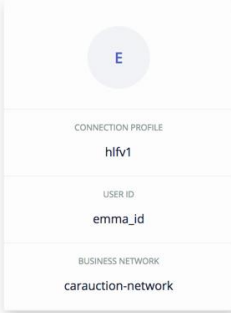
ID Name\*

Participant\*

☐ Allow this ID to issue new IDs (i)

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



CONNECTION PROFILE  
hlfv1

USER ID  
emma\_id

BUSINESS NETWORK  
carauction-network

Use it yourself

Just add the business network card to your wallet to start using the new identity yourself

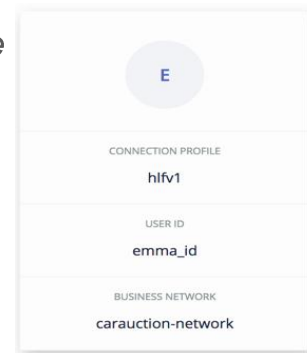
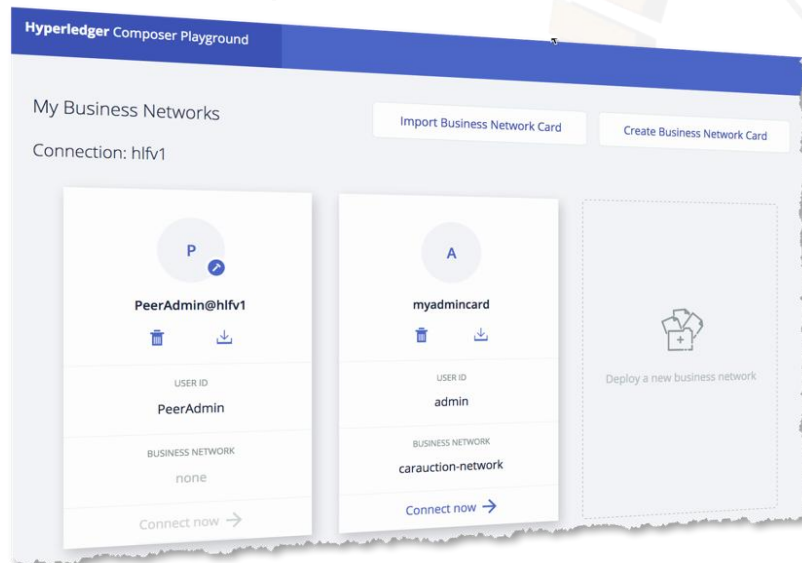
Give the business network card a name  
e.g. emma\_id@carauction-network

> Send it to someone else

⚠ For security, new identities can only be enrolled once

# Business Network Cards

- Business Network Cards are a convenient packaging of *identity* and *connection profile*
  - Contains everything you need to connect to blockchain business network
  - Each card refers to a single participant and single business network
  - Analogous to an ATM card



- Manage cards from both Playground and command-line
  - Create, delete, export, import, list
  - Create requires userid/secret or certificate/private key
- Use cards to connect to Fabric from Playground, command-line or from within your application

```
// Connect and log in to HLF
var businessNetwork = new BusinessNetworkConnection();
return businessNetwork.connect('cardName')
.then(function(businessNetworkDefinition){
    // Connected
});
```



# 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
- Secured using JS Passport, giving >400 options for authentication
- 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 | t

**CloseBidding** : A transaction named CloseBidding

Show/Hide | List Operations | t

Member : A participant named Member

Show/Hide | List Operations | t

Offer : A transaction named Offer

Show/Hide | List Operations | t

**Vehicle** : An asset named Vehicle

Show/Hide | List Operations | t

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

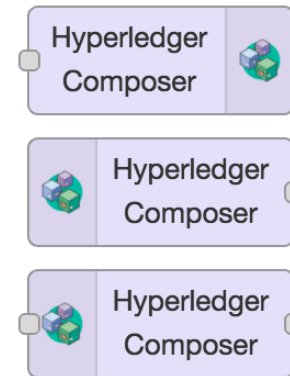
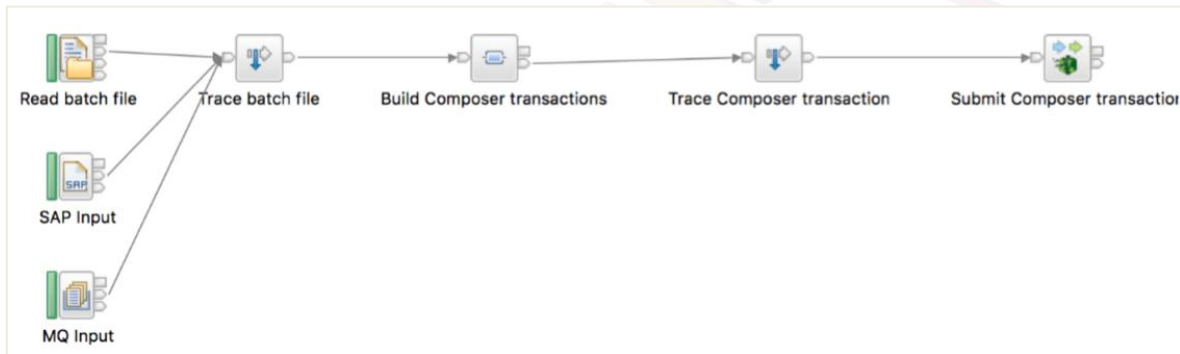
### Request URL

http://0.0.0.0:3000/api/Vehicle

### Response Body

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

# Exploiting Loopback: Examples



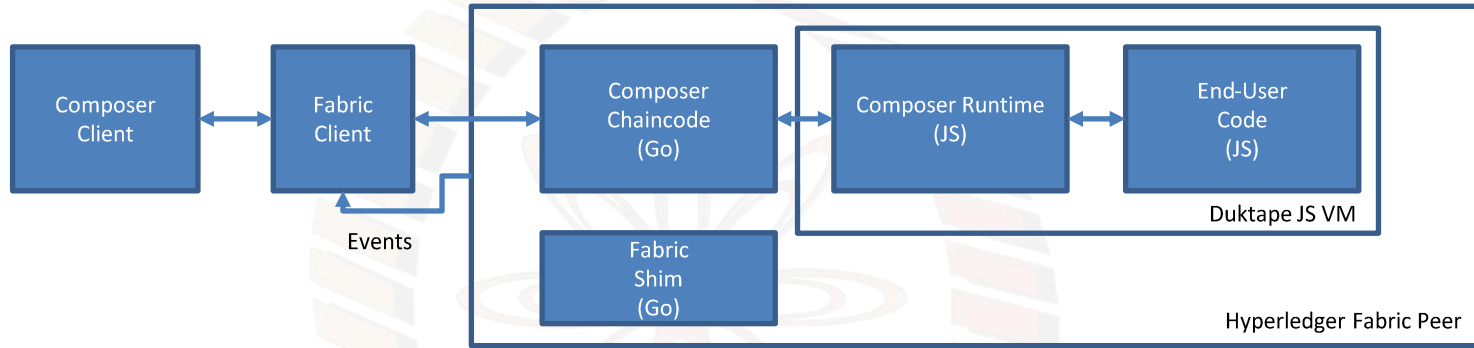
## – IBM Integration Bus

- IIB V10 contains Loopback connector
- Example above takes input from file, SAP or MQ
- Data mapping from CSV, BAPI/IDOC or binary form to JSON model definition

## – Node.RED

- Pre-built nodes available for Composer
- Connect to hardware devices, APIs and online services
- Install direct from Node.RED UI
  - Manage Palette -> Install -> node-red-contrib-composer

# How Composer Maps to Fabric Chaincode



- Each Business Network is deployed to its own chaincode container
  - Container contains a static piece of Go chaincode that starts a Javascript virtual machine running transaction processors
- Browse these containers to view diagnostic information (docker logs)
- Embedded chaincode is not a Composer external interface

# 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.

Let's chat

## Stack Overflow

Ask questions in Stack Overflow with the tag [#hyperledger-composer](#).

Ask now

## Contribute to the Project

### GitHub

Check out the code, feel free to get involved.

GitHub

## Community Call

Join our weekly open community calls.

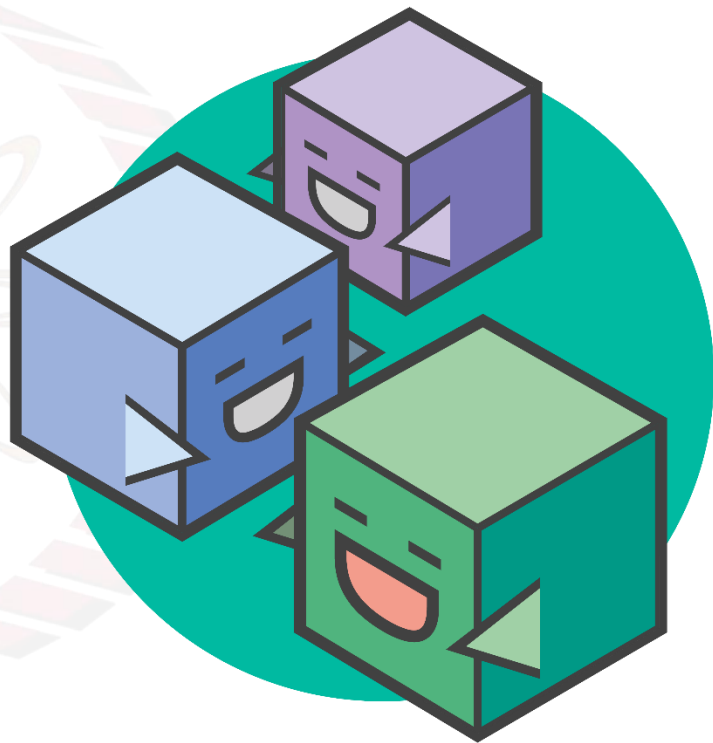
Learn how

# 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/>

<http://composer-playground.mybluemix.net/>



# Fun Reading

- Build your first application using Composer:  
<https://developer.ibm.com/code/patterns/build-your-first-blockchain-application/>
- Hyperledger Cello, <https://www.hyperledger.org/projects/cello>
  - Aims to bring the on-demand “as-a-service” deployment model to the blockchain ecosystem to reduce the effort required for creating, managing and terminating blockchains
- Hyperledger Explorer, <https://www.hyperledger.org/projects/explorer>
  - View, invoke, deploy, or query blocks of transactions along with a visualization framework



thank you!