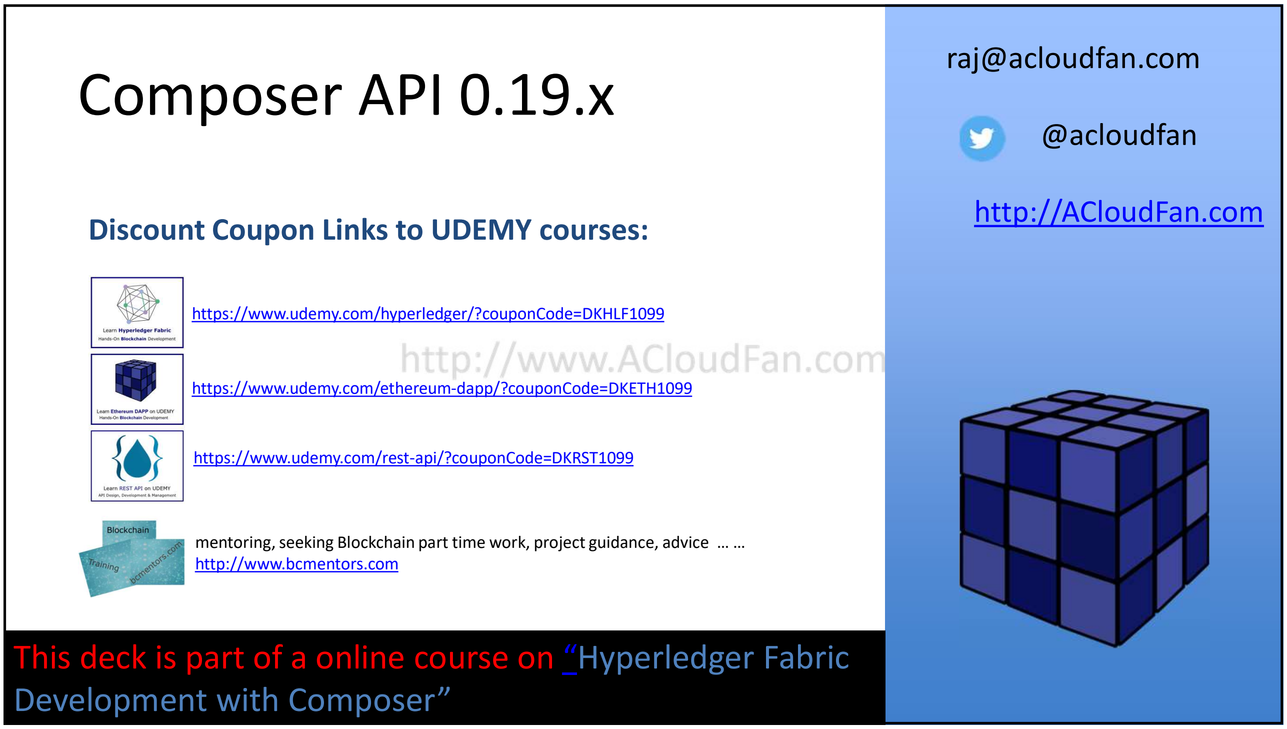


Composer API 0.19.x

Discount Coupon Links to UDEMY courses:

https://www.udemy.com/hyperledger/?couponCode=DKHLF1099

https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099

https://www.udemy.com/rest-api/?couponCode=DKRST1099

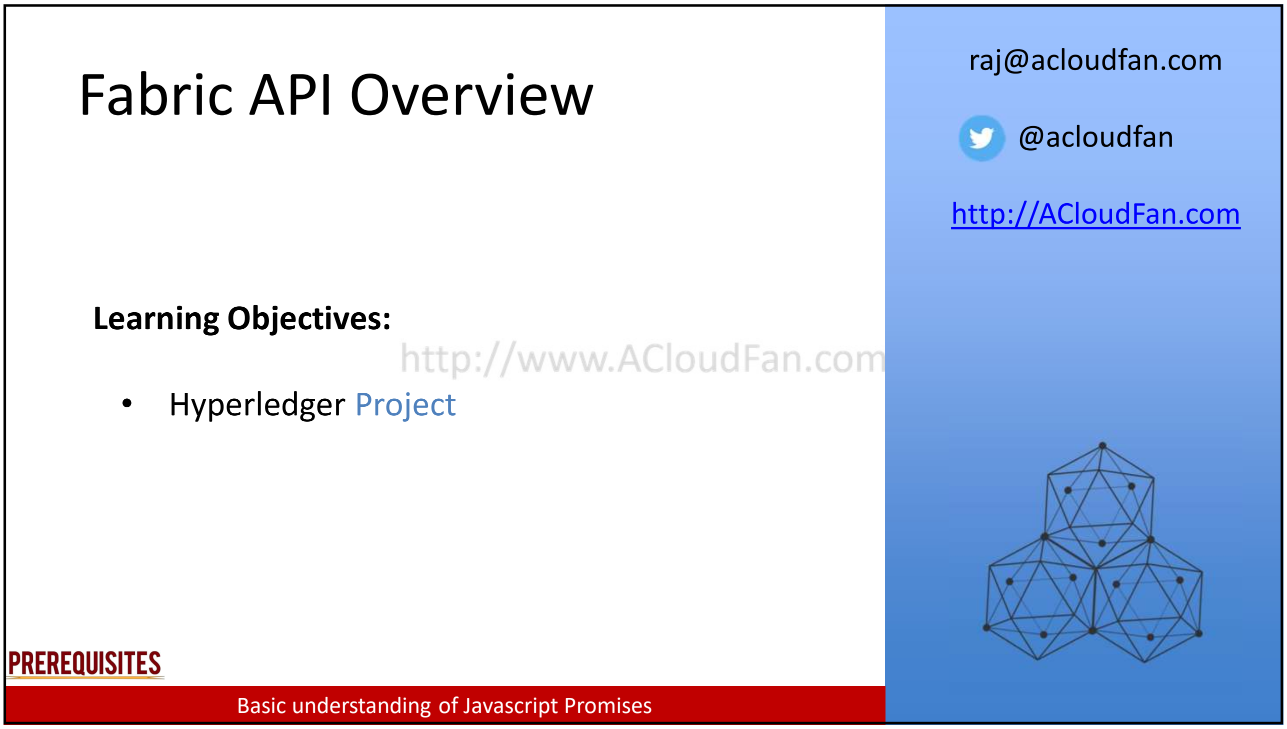
mentoring, seeking Blockchain part time work, project guidance, advice … …
http://www.bcmentors.com

This deck is part of a online course on "Hyperledger Fabric Development with Composer"

raj@acloudfan.com

@acloudfan

http://ACloudFan.com



Fabric API Overview

Learning Objectives:

- Hyperledger Project

PREREQUISITES

Basic understanding of Javascript Promises

raj@acloudfan.com

@acloudfan

http://ACloudFan.com

API Runtime

- Transaction processing functions

ES5

API Client

- Used for developing business application UI

ES6

API Admin

- Administration tools & utilities

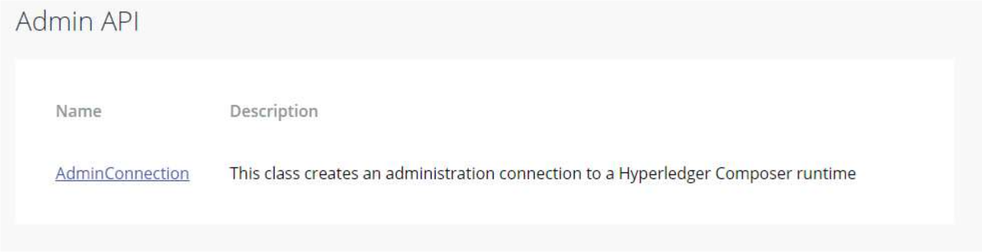
ES6

API Common

- Get information about the business network
- Carry out CRUD operations on resources

API Admin

An object that exposes functions for Administration

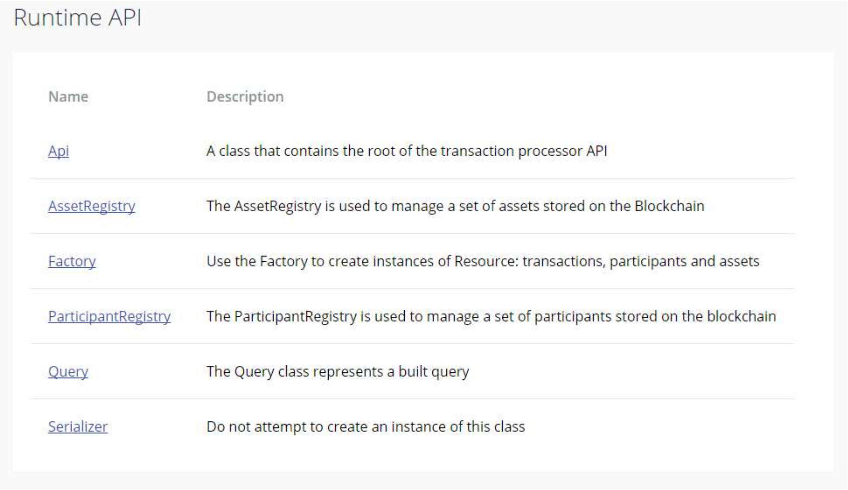


Admin API

Name	Description
AdminConnection	This class creates an administration connection to a Hyperledger Composer runtime

API Runtime

Available to code in Transaction Processor function



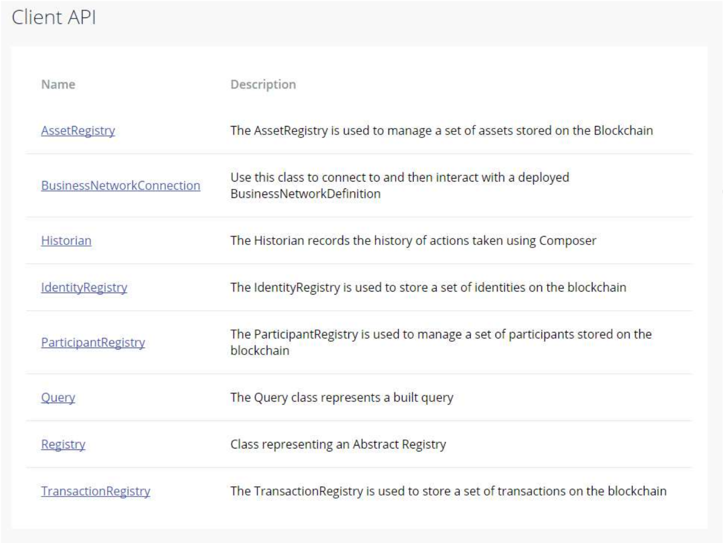
Runtime API

Name	Description
Api	A class that contains the root of the transaction processor API
AssetRegistry	The AssetRegistry is used to manage a set of assets stored on the Blockchain
Factory	Use the Factory to create instances of Resource: transactions, participants and assets
ParticipantRegistry	The ParticipantRegistry is used to manage a set of participants stored on the blockchain
Query	The Query class represents a built query
Serializer	Do not attempt to create an instance of this class

ES5

API Client

Used in a client application for interacting with Fabric



Client API

Name	Description
AssetRegistry	The AssetRegistry is used to manage a set of assets stored on the Blockchain
BusinessNetworkConnection	Use this class to connect to and then interact with a deployed BusinessNetworkDefinition
Historian	The Historian records the history of actions taken using Composer
IdentityRegistry	The IdentityRegistry is used to store a set of identities on the blockchain
ParticipantRegistry	The ParticipantRegistry is used to manage a set of participants stored on the blockchain
Query	The Query class represents a built query
Registry	Class representing an Abstract Registry
TransactionRegistry	The TransactionRegistry is used to store a set of transactions on the blockchain

Network Card Store

Learning Objectives:

- Card storage + Wallet
- API for managing *Cards*

API Common IdCard

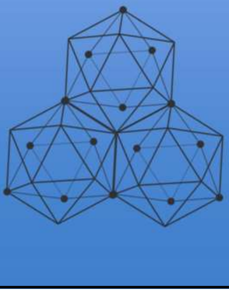
API Common BusinessNetworkCardStore

API Common NetworkCardStoreManager

raj@acloudfan.com

@acloudfan

http://ACloudFan.com



Hyperledger Fabric API

https://github.com/acloudfan/**HLF-Fabric-API**

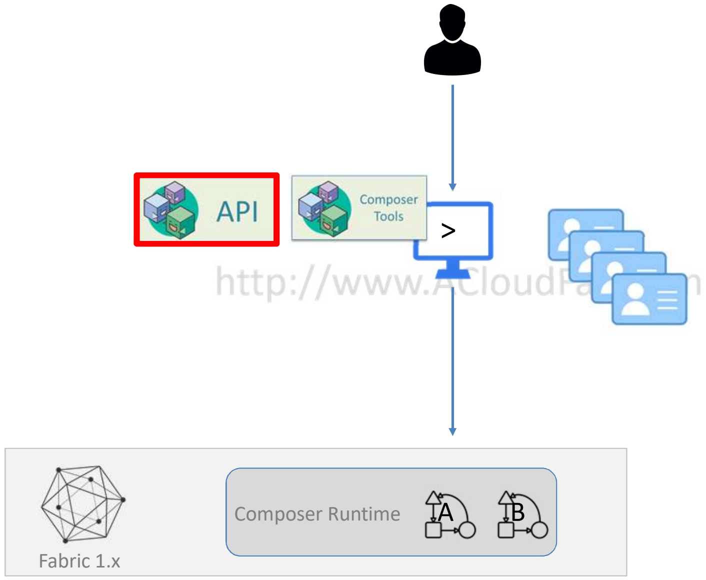
Samples demonstrating use of various Fabric SDK/API

JS manage-cards.js

Code shown in video may change over time

Business Network Card

LET'S RECAP...



API

Composer Tools

>

Fabric 1.x

Composer Runtime

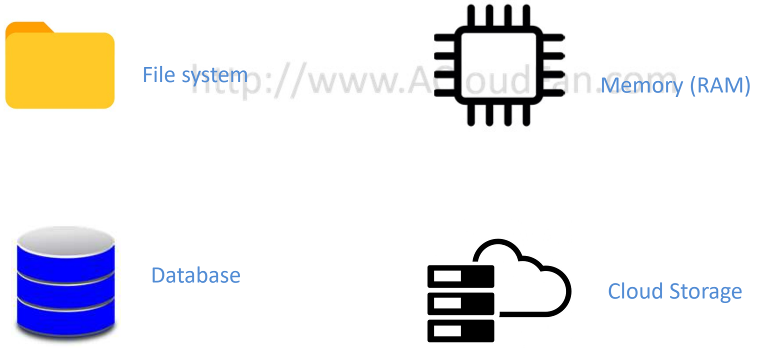
A

B

- Credentials
- Keys & Certificates
- Connection profile

Card Storage

Cards can be stored in any type of storage

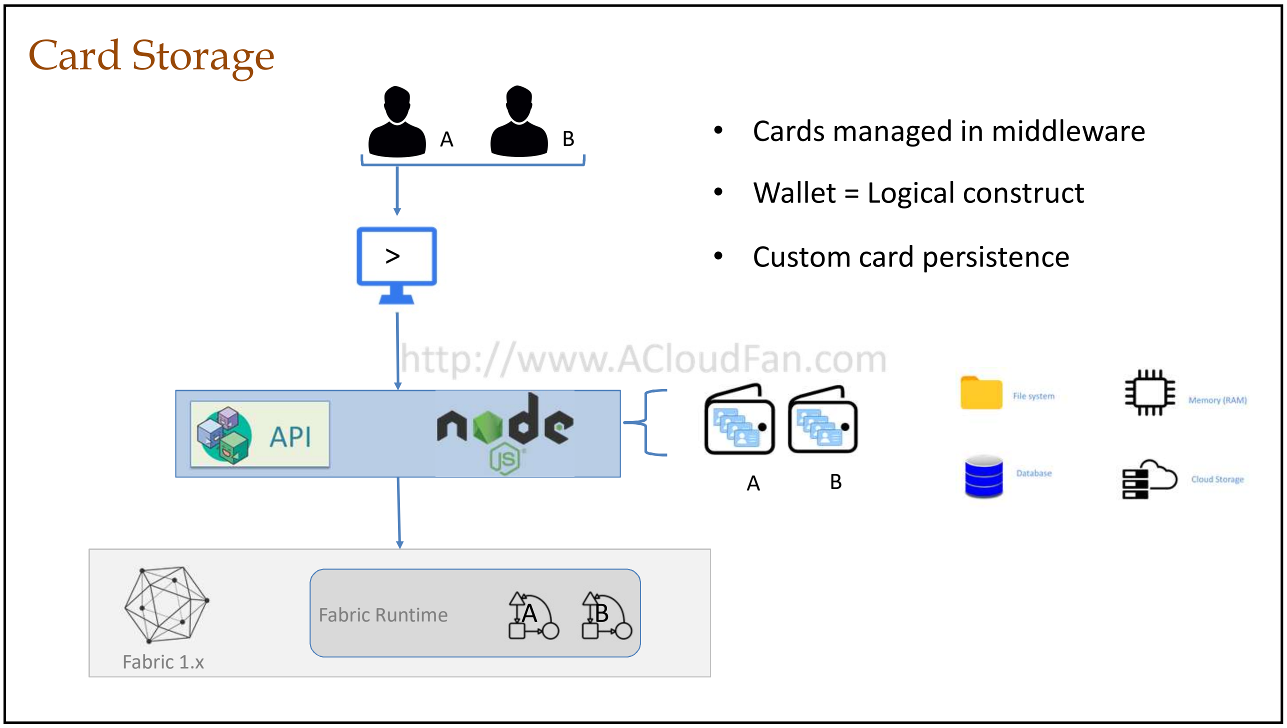


File system

Memory (RAM)

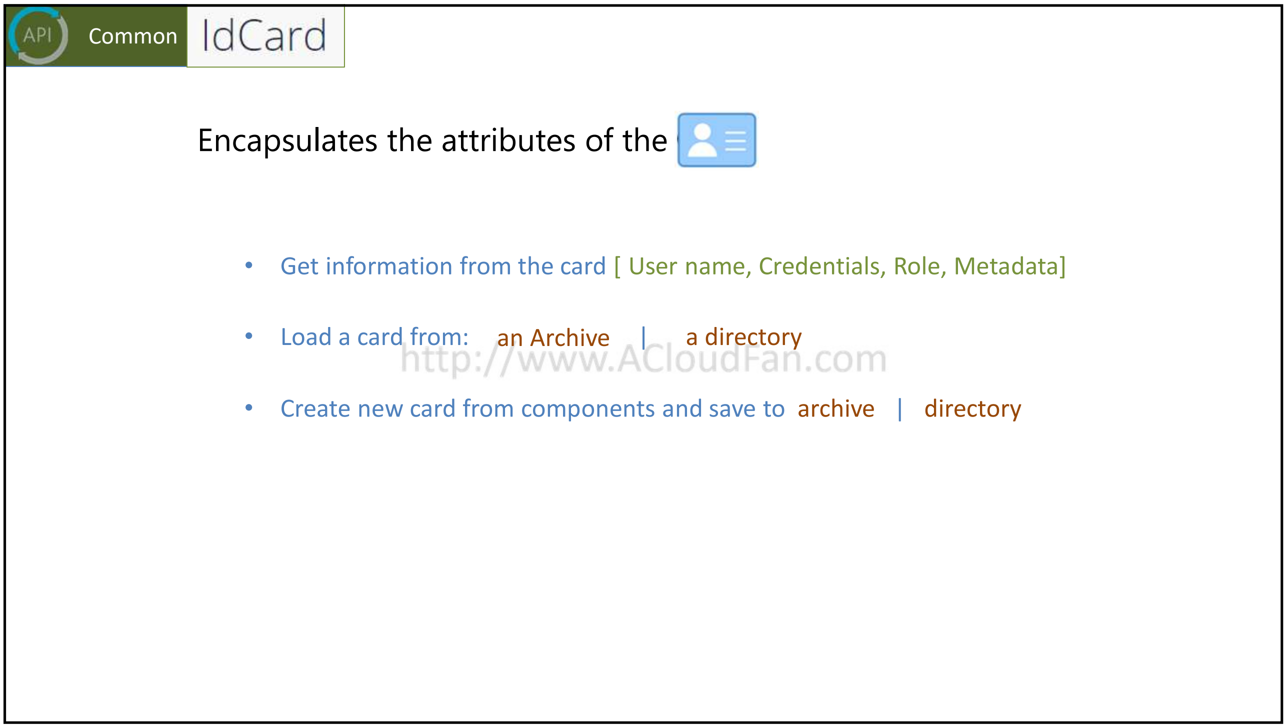
Database

Cloud Storage



Card Storage

- Cards managed in middleware
- Wallet = Logical construct
- Custom card persistence



API Common IdCard

Encapsulates the attributes of the

- Get information from the card [User name, Credentials, Role, Metadata]
- Load a card from: an Archive | a directory
- Create new card from components and save to archive | directory

API Common

BusinessNetworkCardStore

Management of cards in persistent storage

~/.composer

- Access the card (s) deployed on the machine
- Check if the named card exist?
- Add & Delete card from the store

API Common

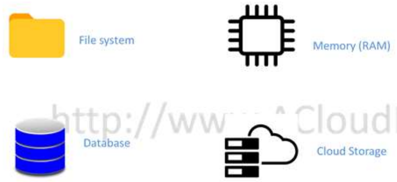
BusinessNetworkCardStore

- Abstract class
- An instance of this class is retrieved from an instance of:

API Common NetworkCardStoreManager

API Common NetworkCardStoreManager

- Factory for creating instance of BusinessNetworkCardStore



- An instance is made available from API Common

```
const NetworkCardStoreManager= require('composer-common').NetworkCardStoreManager;
```

API Common NetworkCardStoreManager

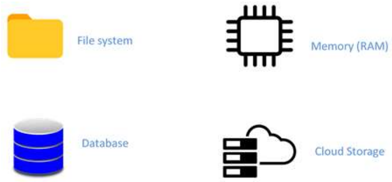
- To get the BusinessNetworkCardStore instance – getCardStore(type)

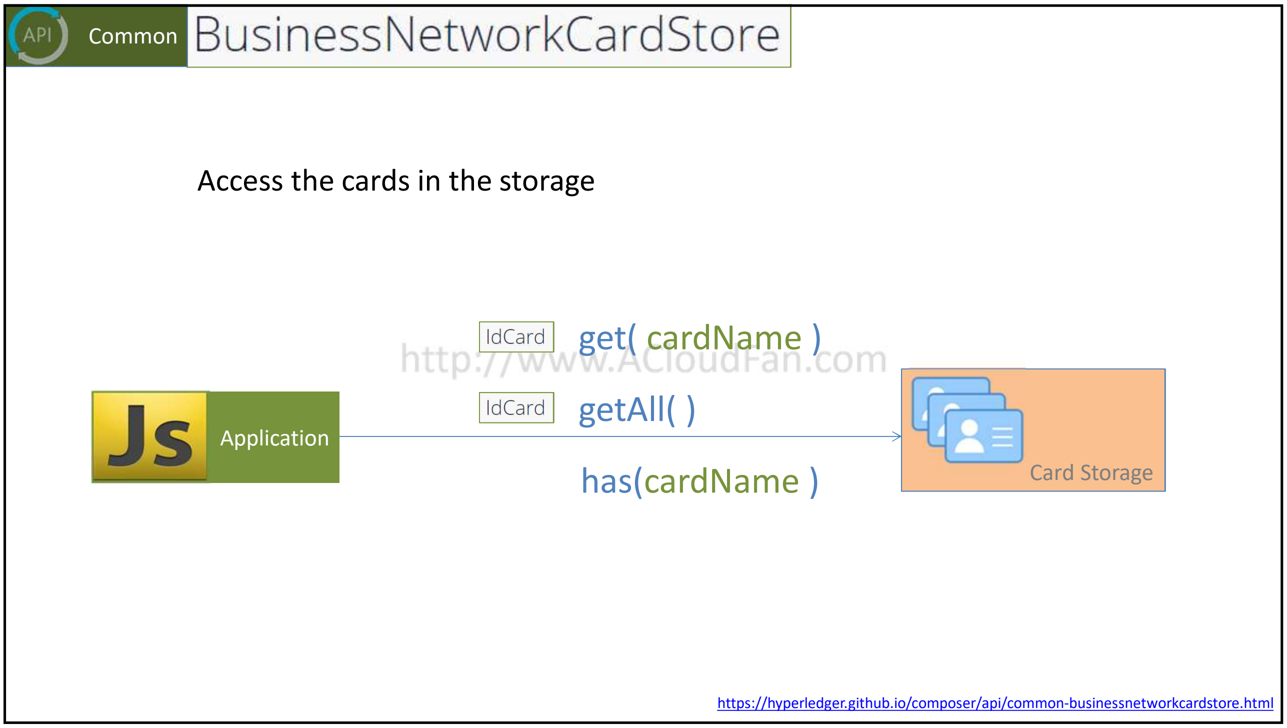
```
const cardStore = NetworkCardStoreManager.getCardStore( walletType );
```

- Pre-defined CardStore types

```
var cardType = { type: 'composer-wallet-filesystem' }
```

```
var cardType = { type: 'composer-wallet-inmemory' }
```





BusinessNetworkCardStore

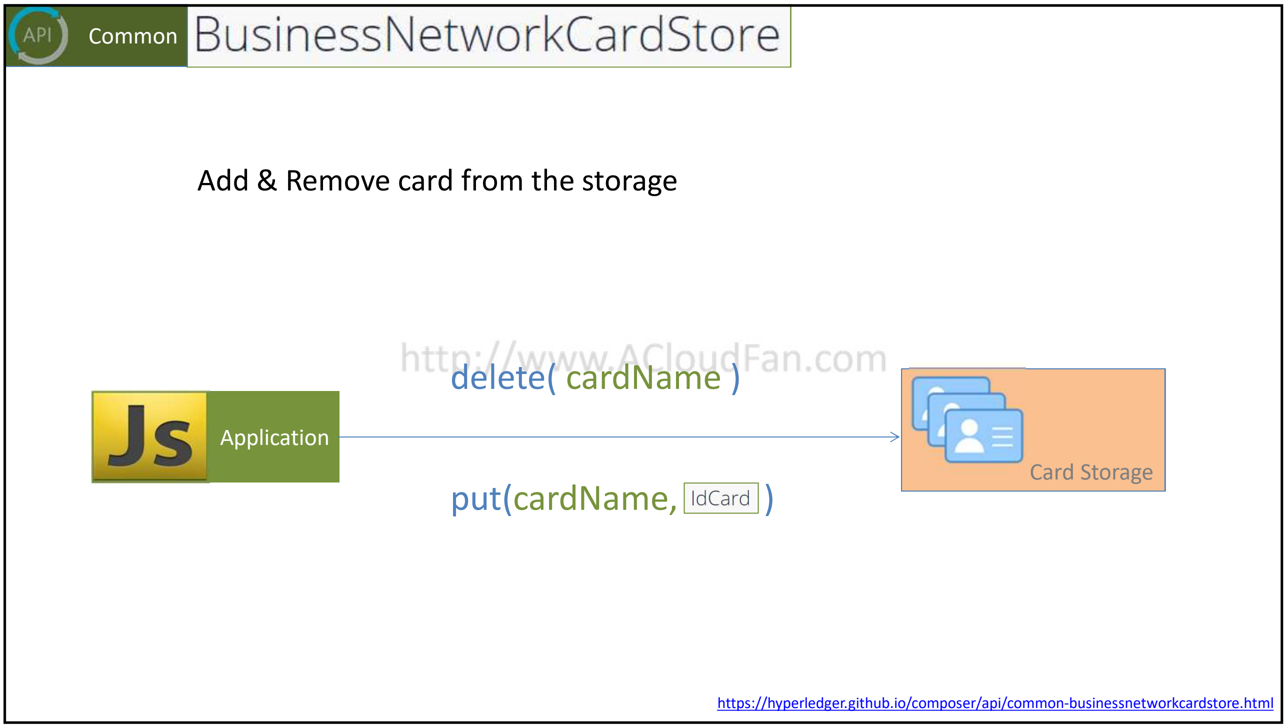
Access the cards in the storage

IdCard get(cardName)

IdCard getAll( )

has(cardName)

https://hyperledger.github.io/composer/api/common-businessnetworkcardstore.html



BusinessNetworkCardStore

Add & Remove card from the storage

delete(cardName)

put(cardName, IdCard)

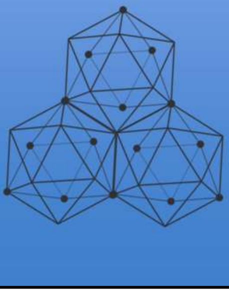
https://hyperledger.github.io/composer/api/common-businessnetworkcardstore.html

Admin Connection

raj@acloudfan.com

@acloudfan

http://ACloudFan.com



Learning Objectives:

- Connecting to Fabric

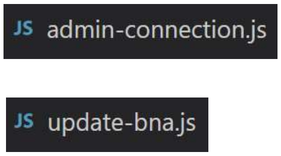
API Admin AdminConnection

API Common BusinessNetworkDefinition

Hyperledger Fabric API

https://github.com/acloudfan/**HLF-Fabric-API**

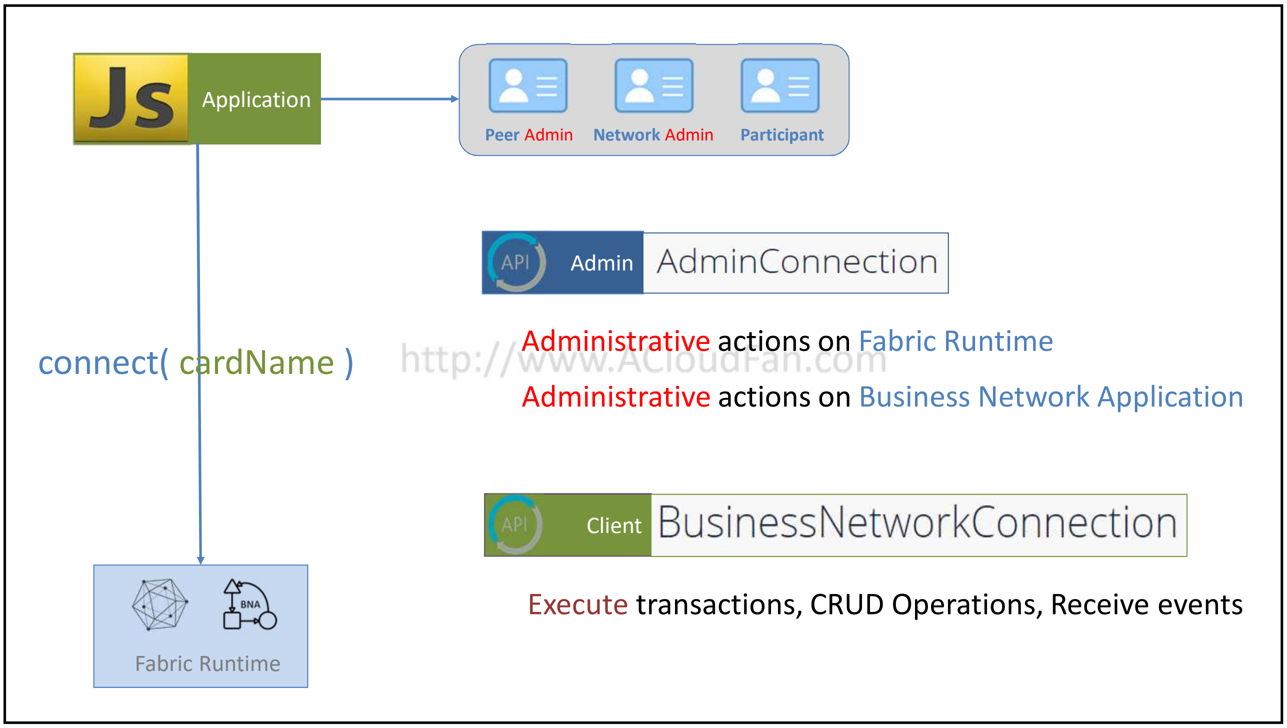
Samples demonstrating use of various Fabric SDK/API



JS admin-connection.js

JS update-bna.js

Code shown in video may change over time



connect(cardName)

AdminConnection

Administrative actions on Fabric Runtime

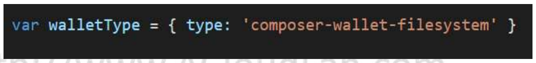
Administrative actions on Business Network Application

BusinessNetworkConnection

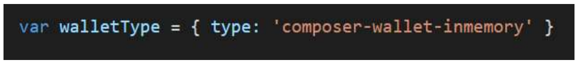
Execute transactions, CRUD Operations, Receive events

API Admin AdminConnection

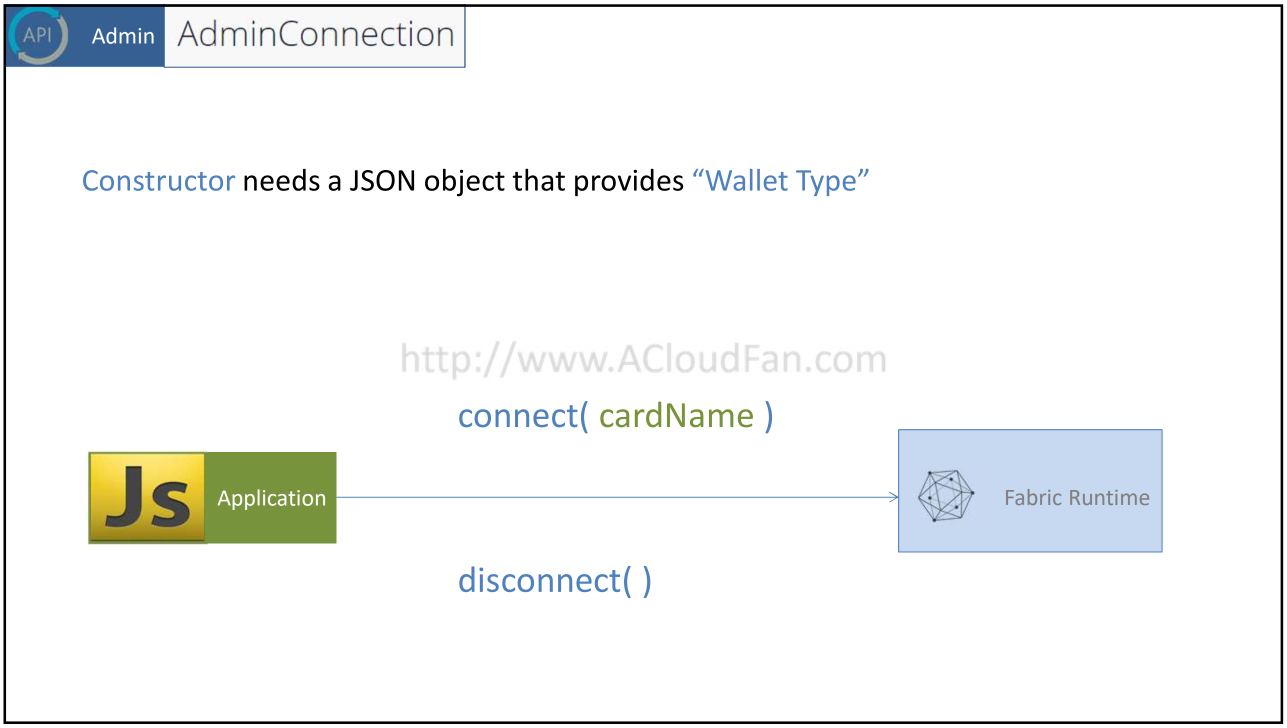
Constructor needs a JSON object that provides “Wallet Type”



```
var walletType = { type: 'composer-wallet-filesystem' }
```



```
var walletType = { type: 'composer-wallet-inmemory' }
```

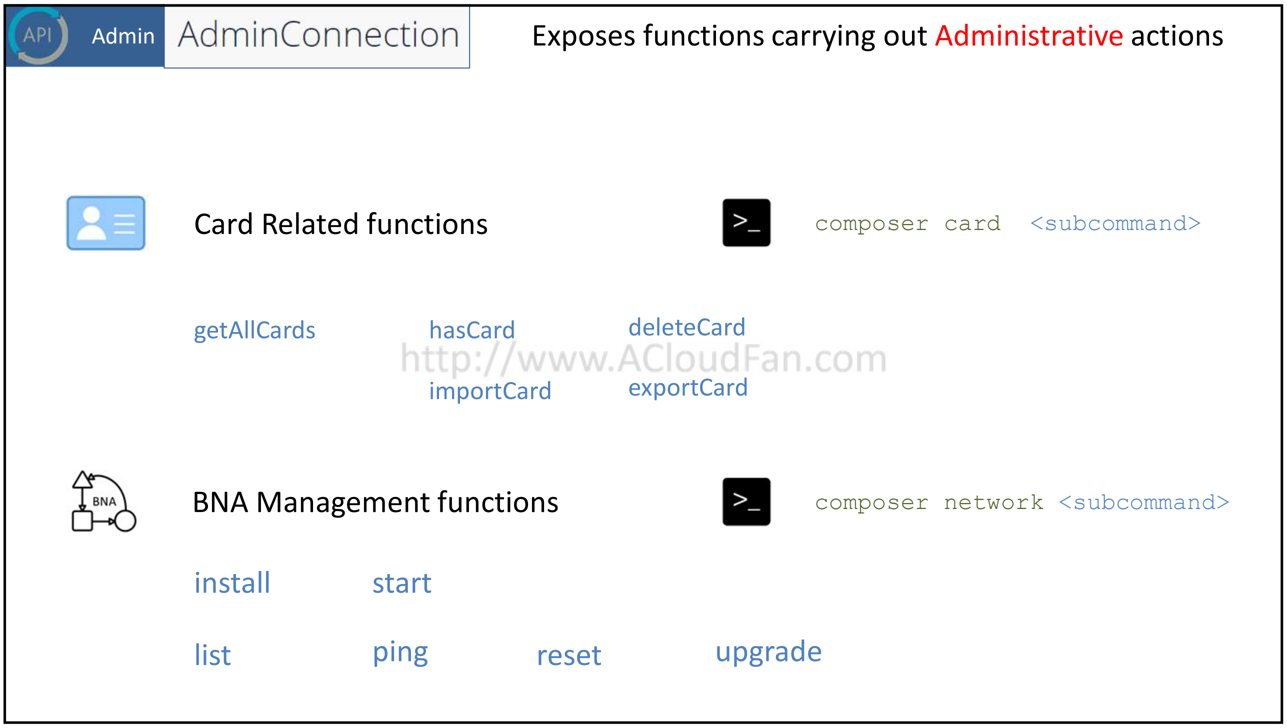


API Admin **AdminConnection**

Constructor needs a JSON object that provides “Wallet Type”

connect(cardName)

disconnect( )



API Admin **AdminConnection**

Exposes functions carrying out Administrative actions

Card Related functions

```
>_ composer card  <subcommand>
```

getAllCards hasCard deleteCard importCard exportCard

BNA Management functions

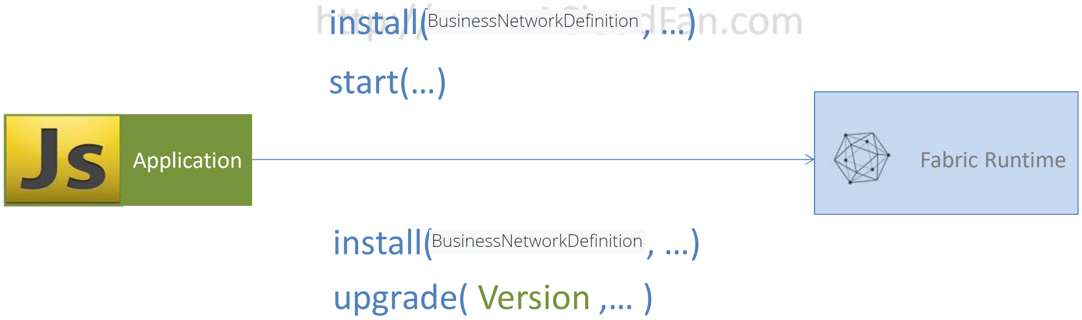
```
>_ composer network <subcommand>
```

install start

list ping reset upgrade

API Admin AdminConnection

install() functions needs an instance of BusinessNetworkDefinition



install(BusinessNetworkDefinition, ...)

start(...)

Application

Fabric Runtime

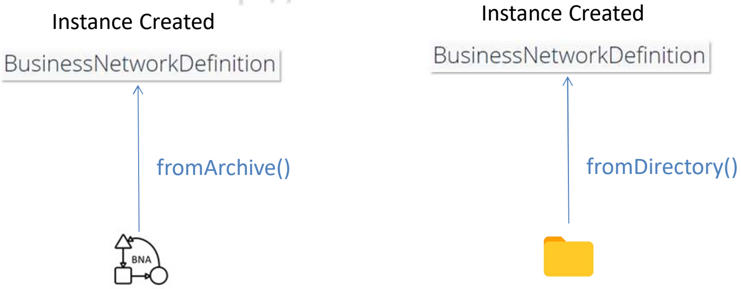
install(BusinessNetworkDefinition, ...)

upgrade(Version ,...)

API Common BusinessNetworkDefinition

Represents the business network application

- Meta data
- Domain specific resource model
- Scripts



Instance Created

BusinessNetworkDefinition

fromArchive()

Instance Created

BusinessNetworkDefinition

fromDirectory()

https://hyperledger.github.io/composer/api/common-businessnetworkdefinition.html

Business Network Connection

raj@acloudfan.com

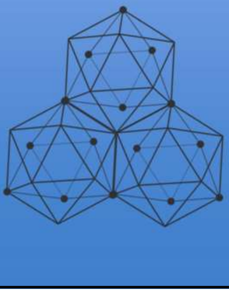
@acloudfan

http://ACloudFan.com

Learning Objectives:

- Connecting
- Disconnecting
- Pinging

API Client BusinessNetworkConnection



Hyperledger Fabric API

https://github.com/acloudfan/**HLF-Fabric-API**

Samples demonstrating use of various Fabric SDK/API

JS bn-connection-util.js

JS test-bn-util.js

Code shown in video may change over time

API Client

BusinessNetworkConnection

Interact with a deployed Business Network Application

- Get information about the application
- Ping the network
- Access registries from runtime
- Manage Identities
- Submit transaction

https://hyperledger.github.io/composer/api/client-businessnetworkconnection.html

API Client

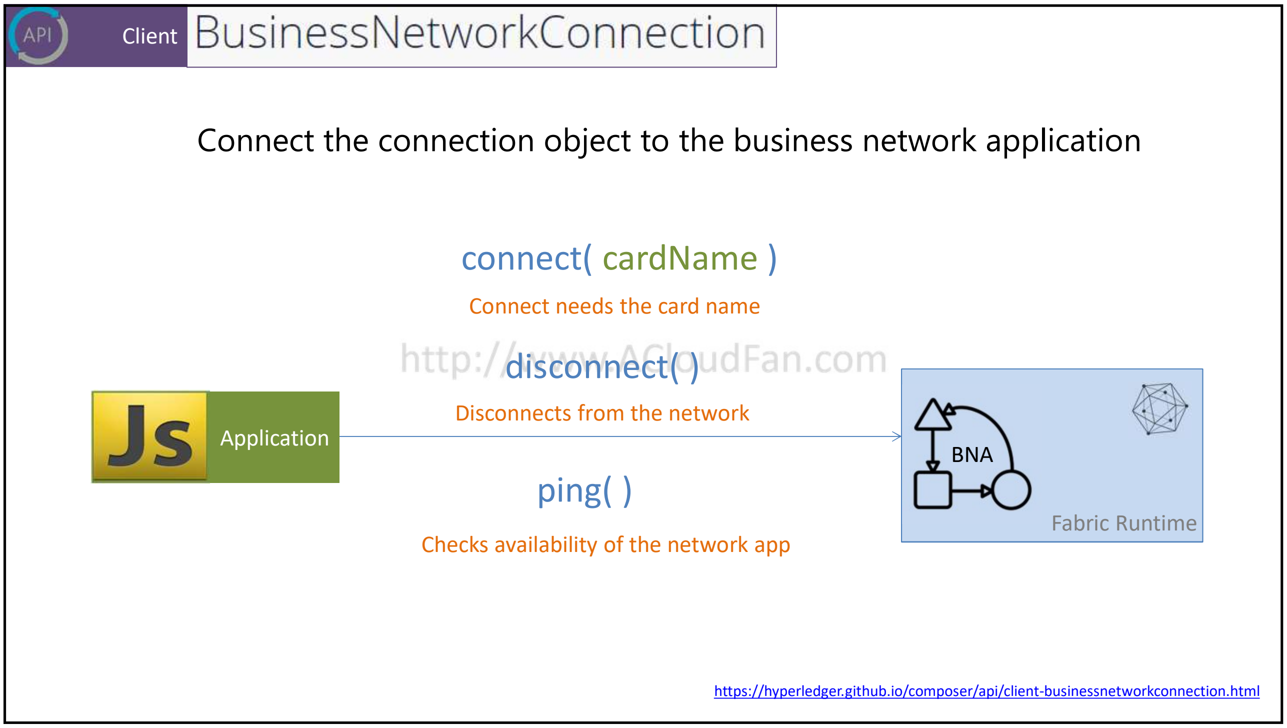
BusinessNetworkConnection

Constructor needs an instance of BusinessNetworkCardStore

Js Application

new BusinessNetworkConnection(cardStoreObject)

https://hyperledger.github.io/composer/api/client-businessnetworkconnection.html



API Client BusinessNetworkConnection

Connect the connection object to the business network application

connect(cardName)

Connect needs the card name

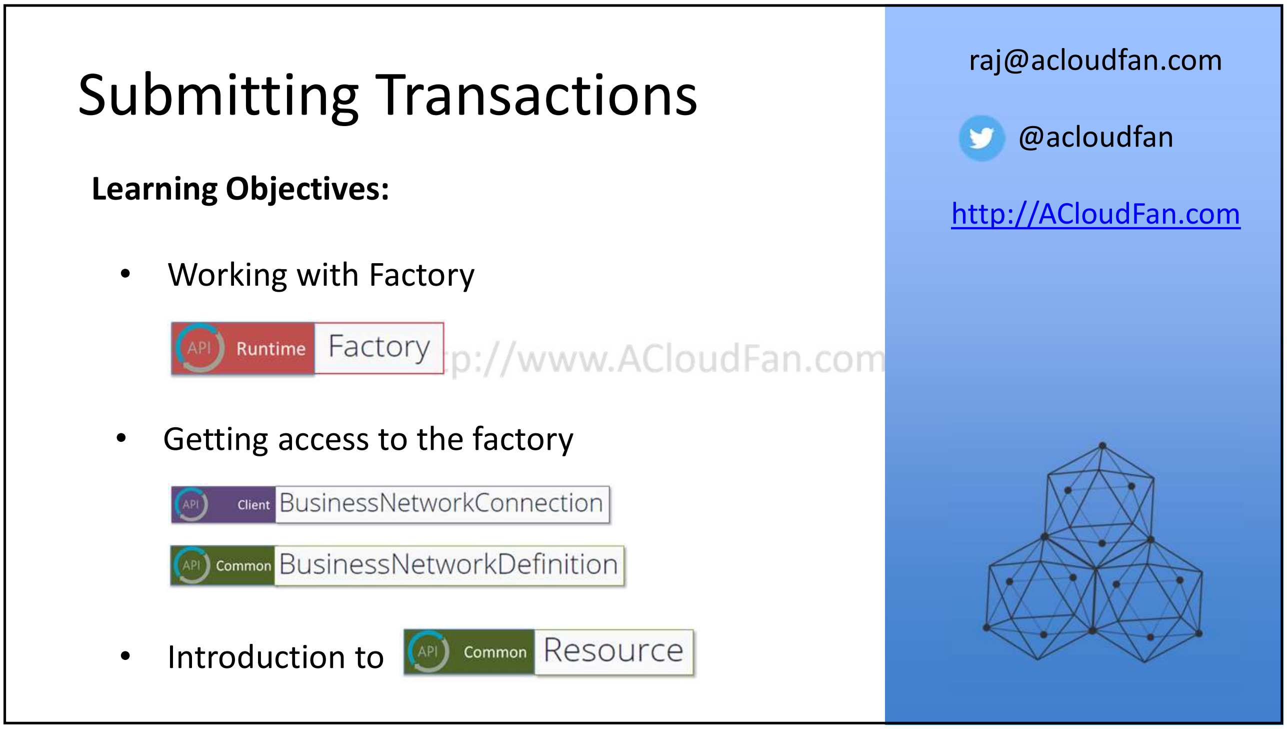
disconnect( )

Disconnects from the network

ping( )

Checks availability of the network app

https://hyperledger.github.io/composer/api/client-businessnetworkconnection.html



Submitting Transactions

Learning Objectives:

- Working with Factory

  API Runtime Factory

- Getting access to the factory

  API Client BusinessNetworkConnection

  API Common BusinessNetworkDefinition

- Introduction to API Common Resource

raj@acloudfan.com

@acloudfan

http://ACloudFan.com

Hyperledger Fabric API

https://github.com/acloudfan/**HLF-Fabric-API**

Samples demonstrating use of various Fabric SDK/API

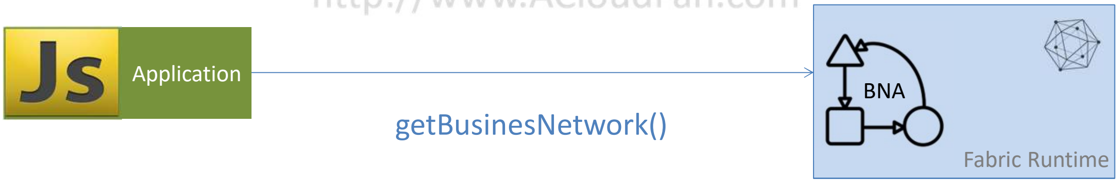
JS bn-factory-submit-txn.js

Code shown in video may change over time

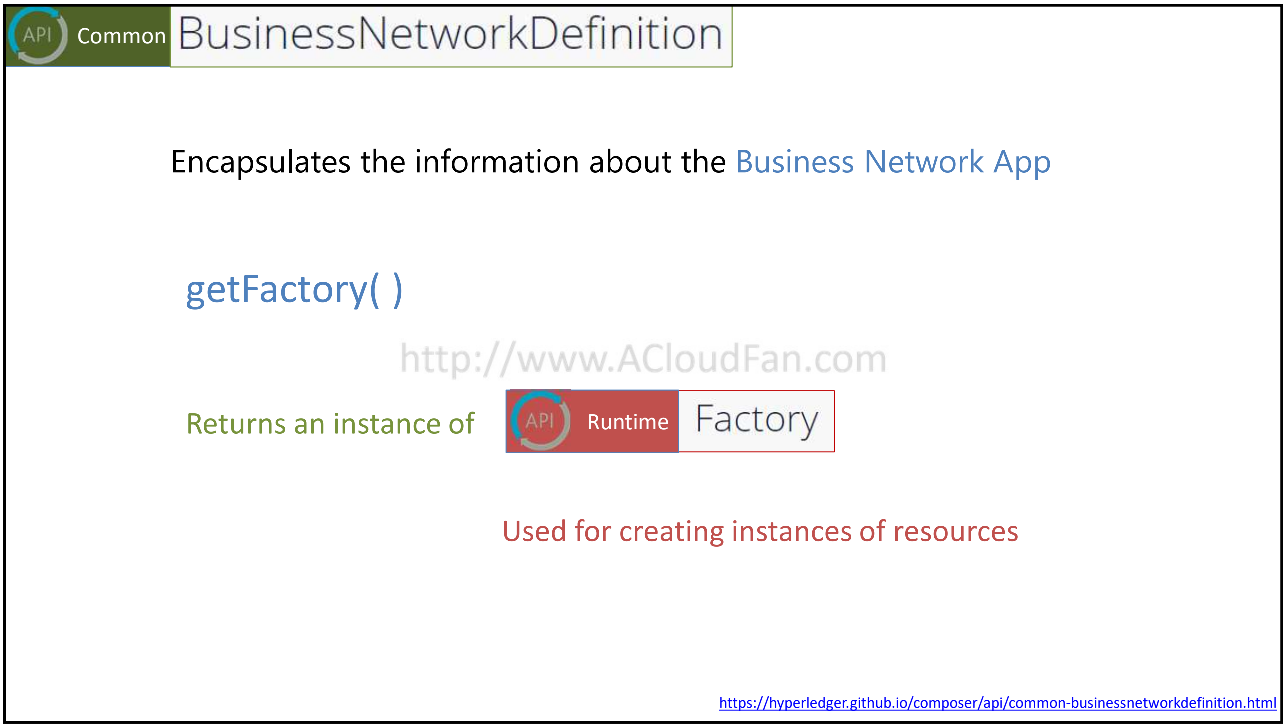
API Client BusinessNetworkConnection

Get the information about the network app from Runtime

Returns an instance of API Common BusinessNetworkDefinition



https://hyperledger.github.io/composer/api/client-businessnetworkconnection.html



API Common BusinessNetworkDefinition

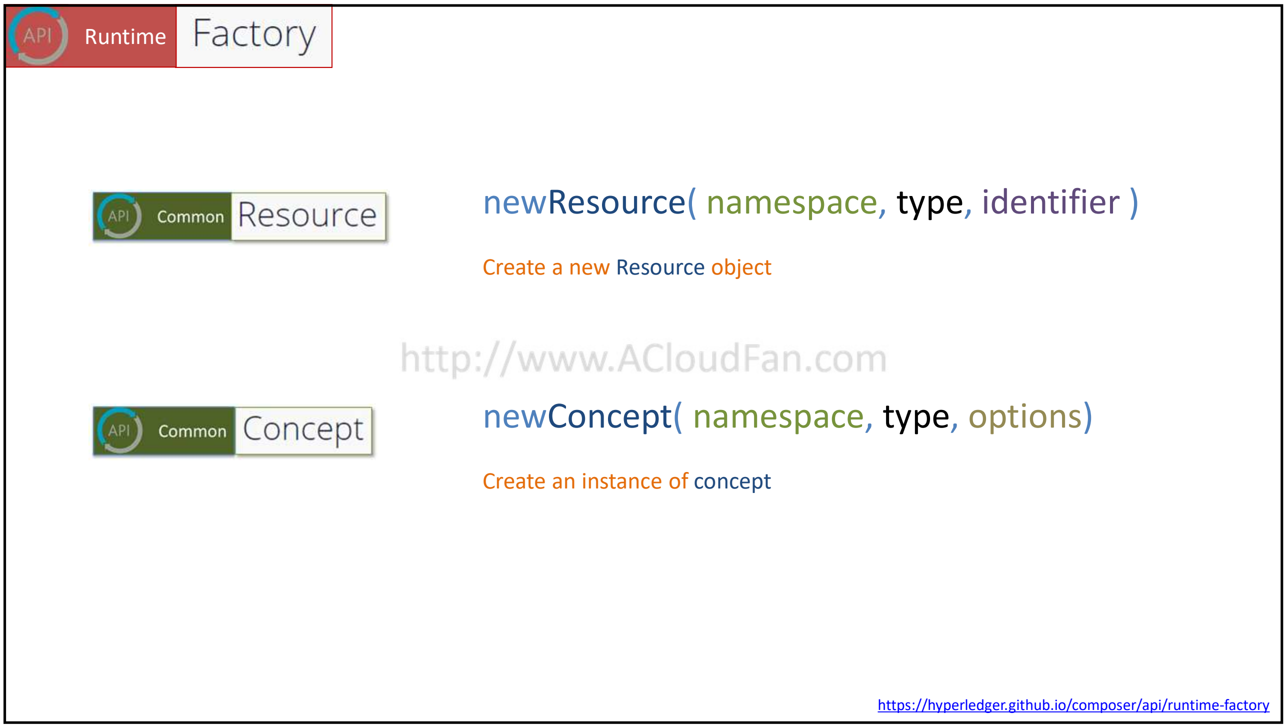
Encapsulates the information about the Business Network App

getFactory( )

Returns an instance of API Runtime Factory

Used for creating instances of resources

https://hyperledger.github.io/composer/api/common-businessnetworkdefinition.html



API Runtime Factory

API Common Resource

newResource(namespace, type, identifier)

Create a new Resource object

API Common Concept

newConcept(namespace, type, options)

Create an instance of concept

https://hyperledger.github.io/composer/api/runtime-factory

API Runtime Factory

API Common Relationship

newRelationship(namespace, type, identifier)

Creates a pointer to the resource identified by identifier

API Common Resource

newEvent(namespace, type, id, options)

Create an instance of event

https://hyperledger.github.io/composer/api/runtime-factory

API Runtime Factory

API Common Resource

newTransaction(namespace, type, identifier, options)

Create an instance of Resource (type = transaction)

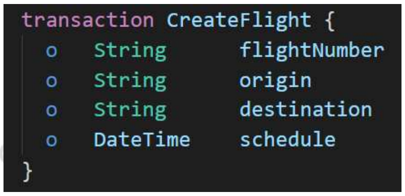
Submitted for processing using the instance of API Client BusinessNetworkConnection

submitTransaction(transaction)

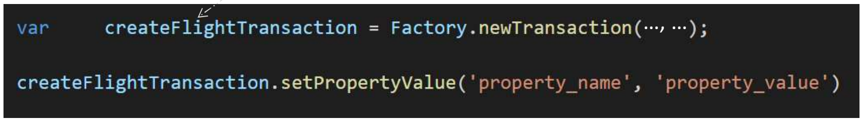
https://hyperledger.github.io/composer/api/runtime-factory

API Common Resource

Represents a resource instance that has *Type*



```
transaction CreateFlight {
  o   String      flightNumber
  o   String      origin
  o   String      destination
  o   DateTime    schedule
}
```



```
var     createFlightTransaction = Factory.newTransaction(…, …);

createFlightTransaction.setPropertyValue('property_name', 'property_value')
```

https://hyperledger.github.io/composer/api/common-resource.html

Composer API 0.19.x

raj@acloudfan.com

@acloudfan

http://ACloudFan.com

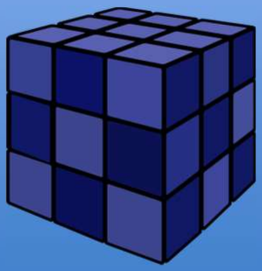
Discount Coupon Links to UDEMY courses:

https://www.udemy.com/hyperledger/?couponCode=DKHLF1099

https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099

https://www.udemy.com/rest-api/?couponCode=DKRST1099

mentoring, seeking Blockchain part time work, project guidance, advice … …
http://www.bcmentors.com



This deck is part of a online course on "Hyperledger Fabric Development with Composer"

Registries

Learning Objectives:

- Quick re-cap of Registry
- Registry Class

API Client Registry

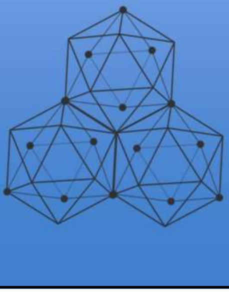
- Accessing Registry instances in Runtime

API Client BusinessNetworkConnection

raj@acloudfan.com

@acloudfan

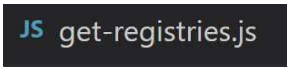
http://ACloudFan.com



Hyperledger Fabric API

https://github.com/acloudfan/**HLF-Fabric-API**

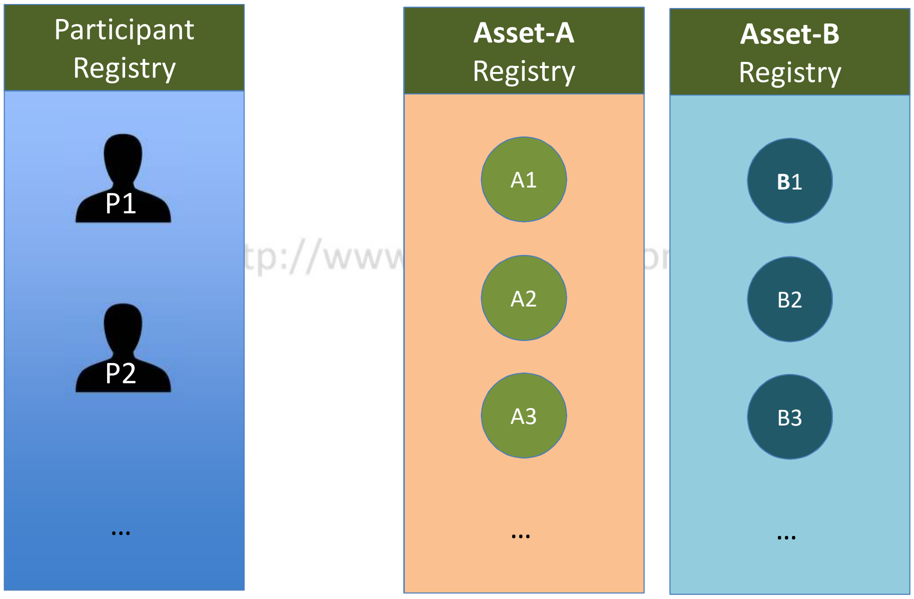
Samples demonstrating use of various Fabric SDK/API



JS get-registries.js

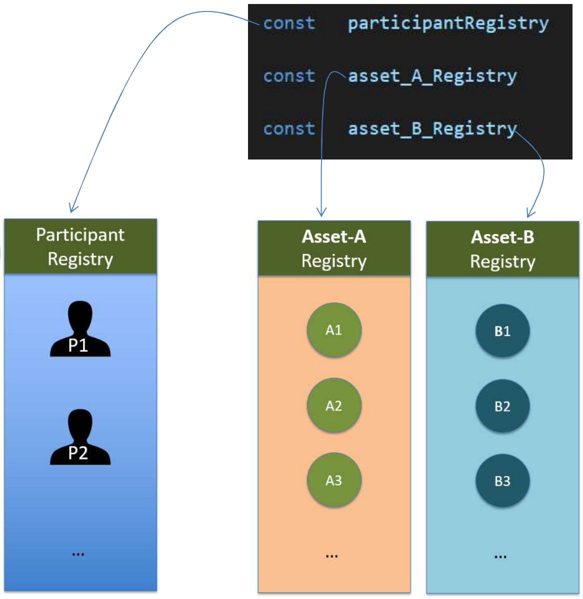
Code shown in video may change over time

Registry



API Client Registry

Abstract class representing a Registry

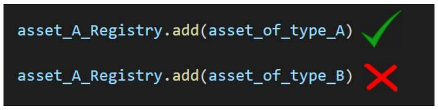


```
const    participantRegistry

const    asset_A_Registry

const    asset_B_Registry
```

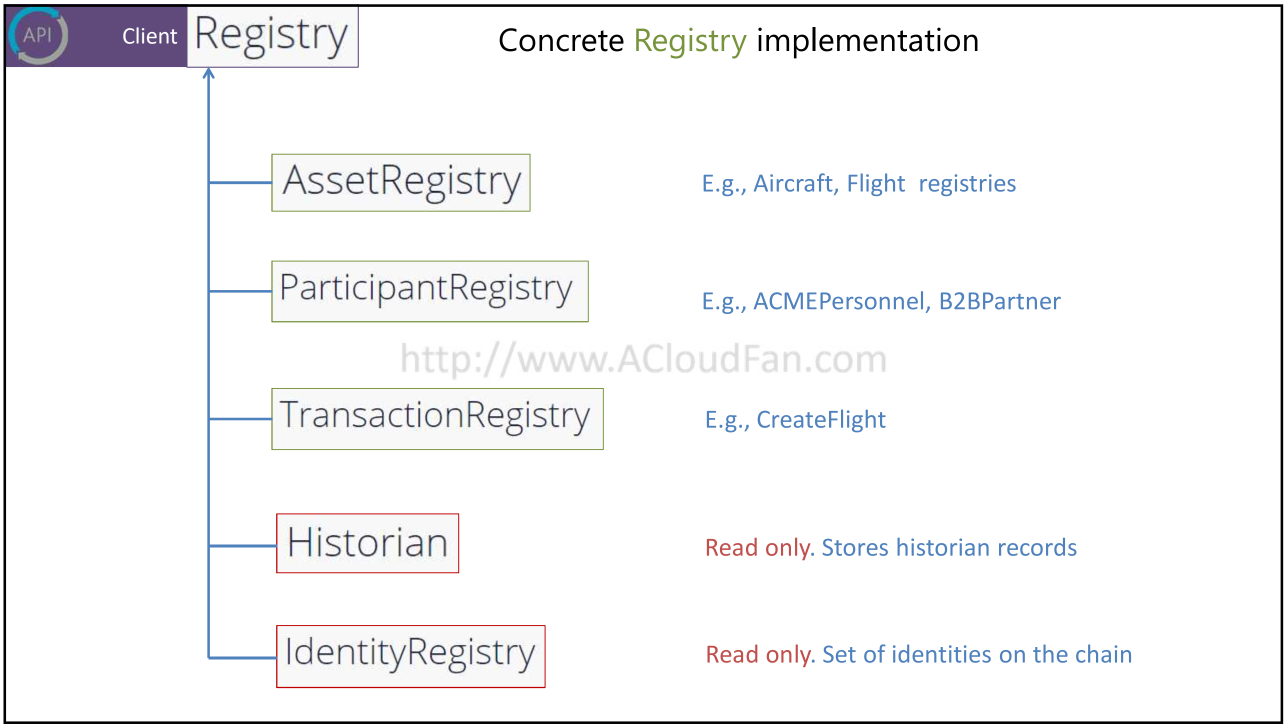
- Apps can carry out CRUD operations
- Resource type MUST match with Registry



```
asset_A_Registry.add(asset_of_type_A) ✓

asset_A_Registry.add(asset_of_type_B) ✗
```

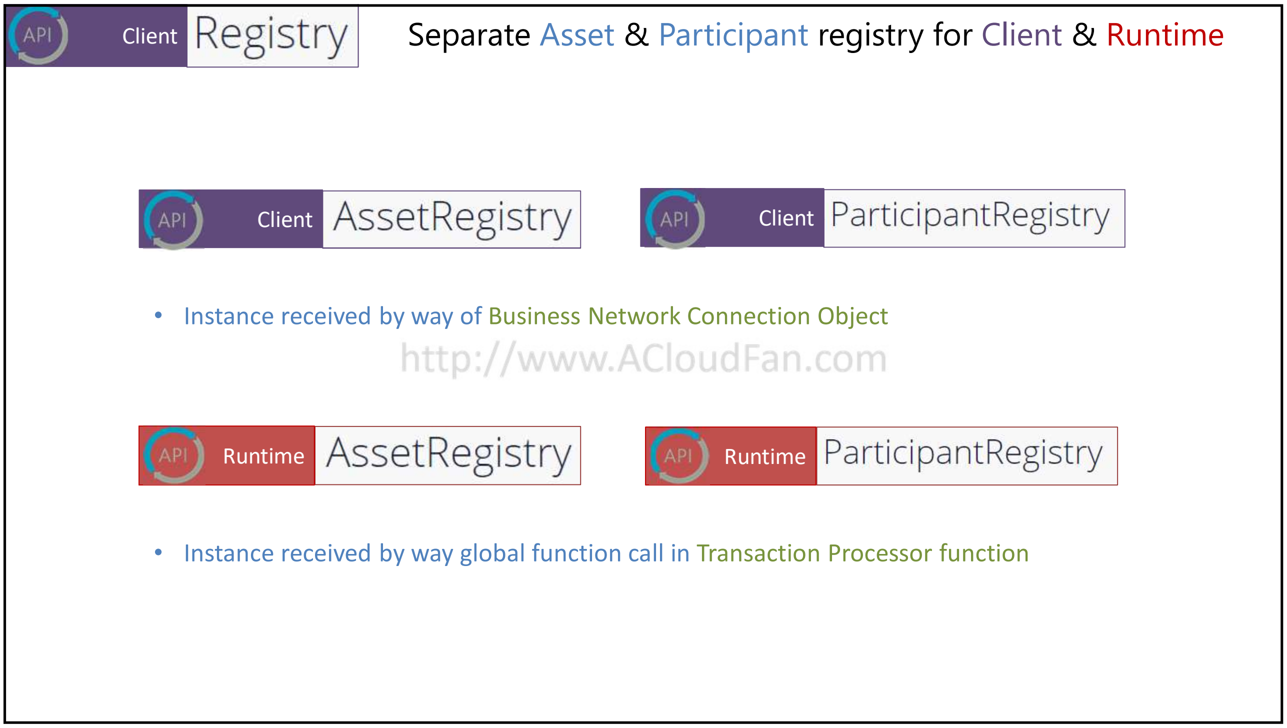
https://hyperledger.github.io/composer/api/client-registry.html



API Client Registry

Concrete Registry implementation

- AssetRegistry — E.g., Aircraft, Flight registries
- ParticipantRegistry — E.g., ACMEPersonnel, B2BPartner
- TransactionRegistry — E.g., CreateFlight
- Historian — Read only. Stores historian records
- IdentityRegistry — Read only. Set of identities on the chain



API Client Registry

Separate Asset & Participant registry for Client & Runtime

API Client AssetRegistry

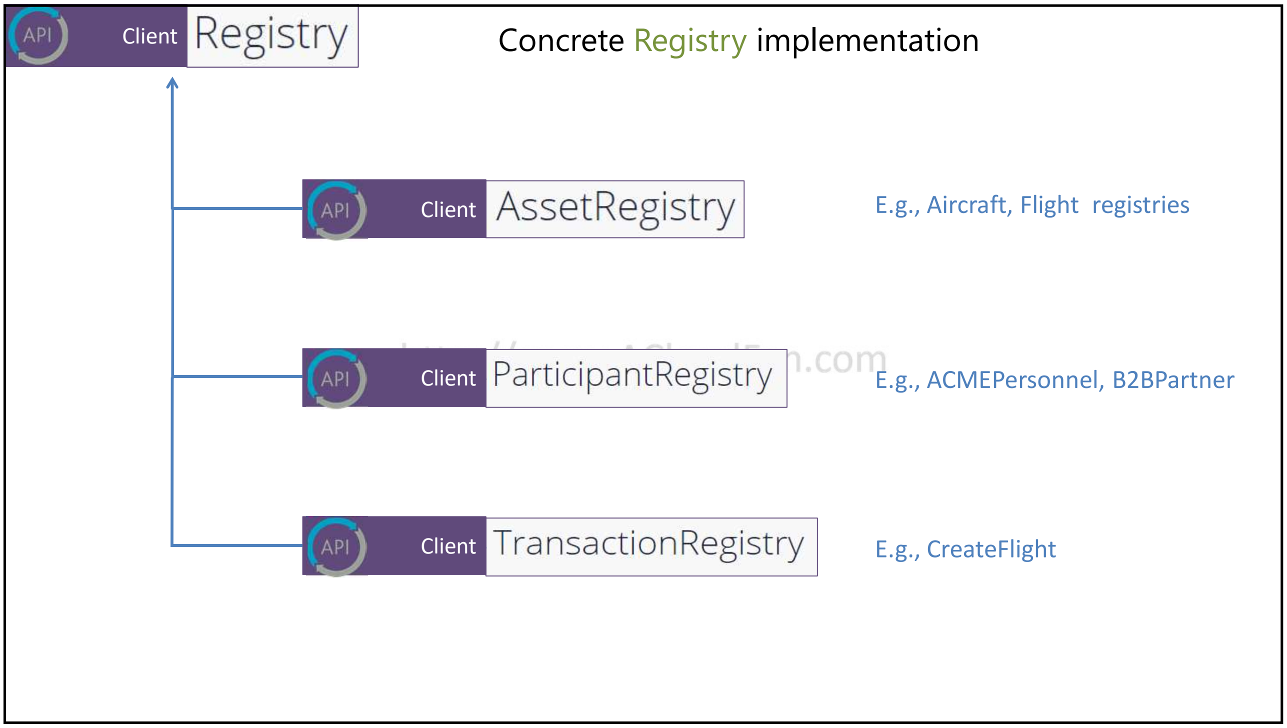
API Client ParticipantRegistry

- Instance received by way of Business Network Connection Object

API Runtime AssetRegistry

API Runtime ParticipantRegistry

- Instance received by way global function call in Transaction Processor function

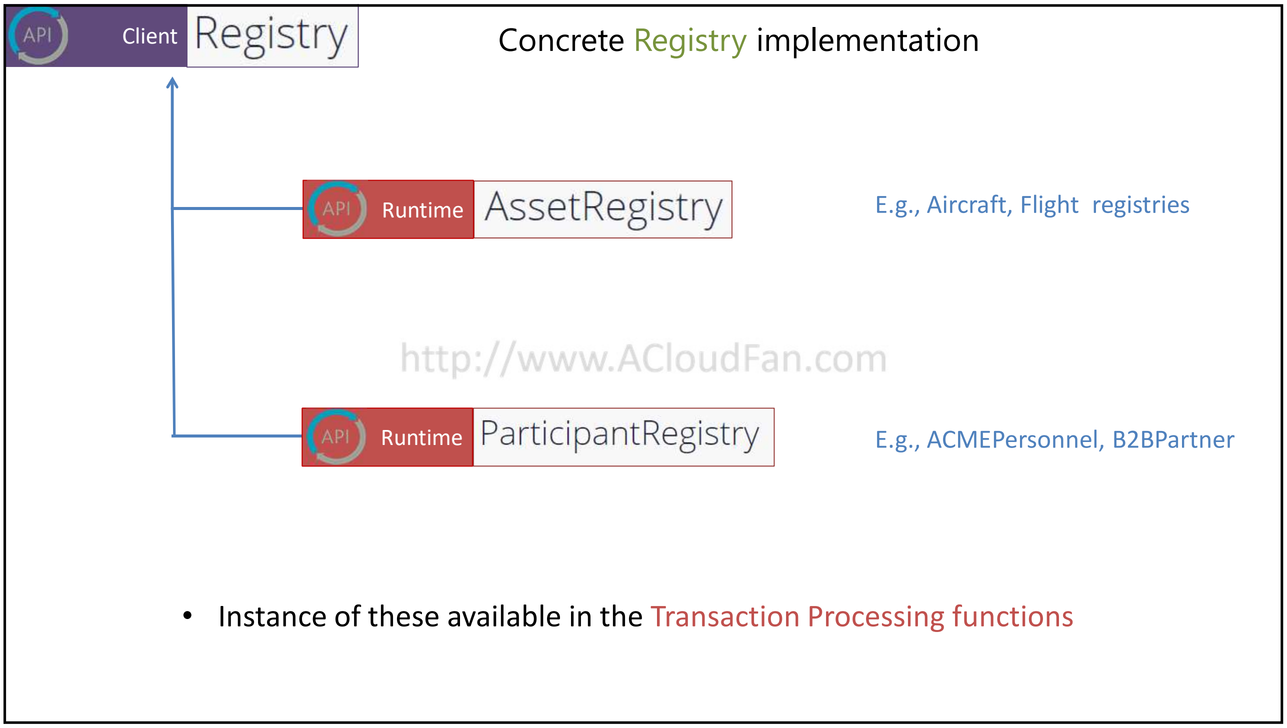


Concrete Registry implementation

E.g., Aircraft, Flight registries

E.g., ACMEPersonnel, B2BPartner

E.g., CreateFlight

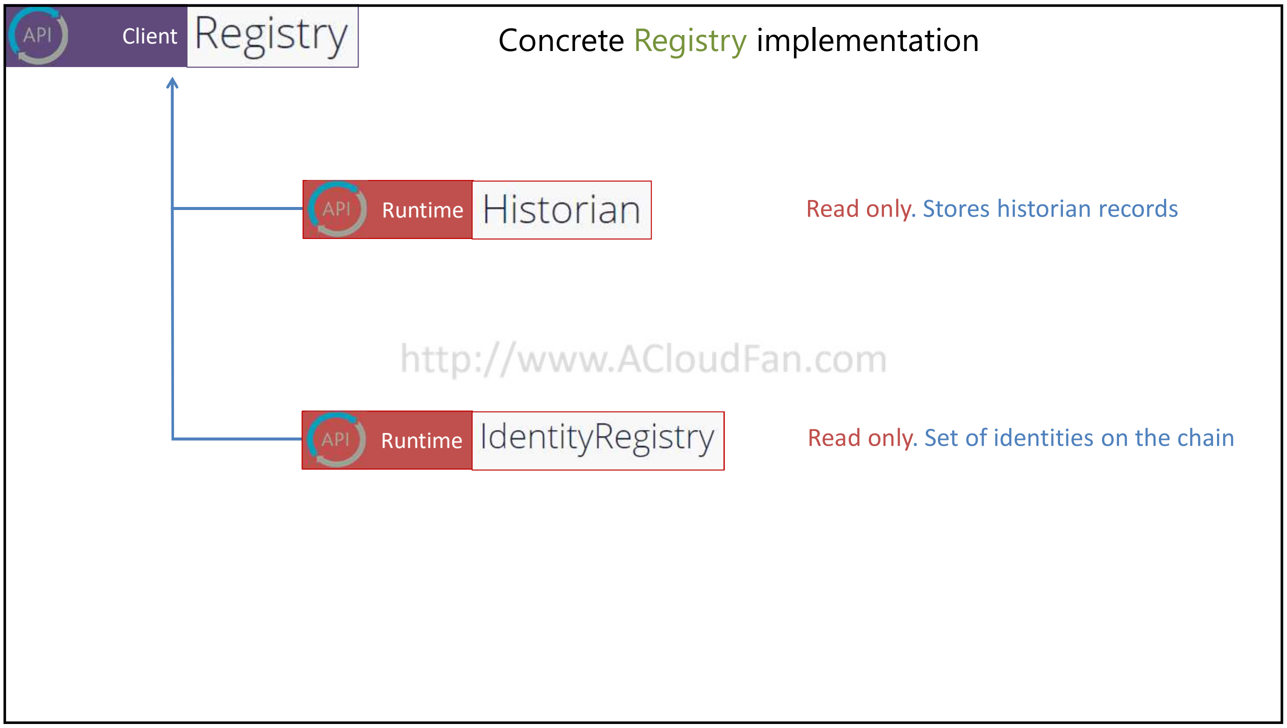


Concrete Registry implementation

E.g., Aircraft, Flight registries

E.g., ACMEPersonnel, B2BPartner

- Instance of these available in the Transaction Processing functions



API Client Registry

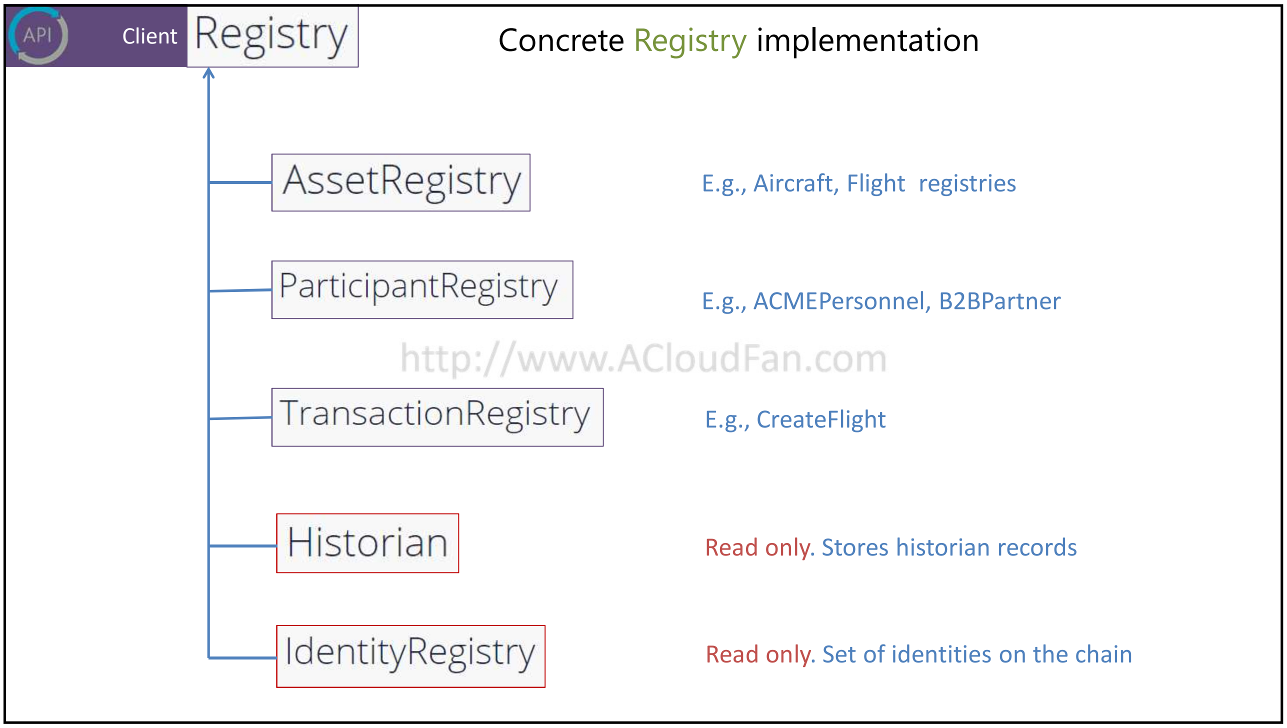
Concrete Registry implementation

API Runtime Historian

Read only. Stores historian records

API Runtime IdentityRegistry

Read only. Set of identities on the chain



API Client Registry

Concrete Registry implementation

AssetRegistry

E.g., Aircraft, Flight registries

ParticipantRegistry

E.g., ACMEPersonnel, B2BPartner

TransactionRegistry

E.g., CreateFlight

Historian

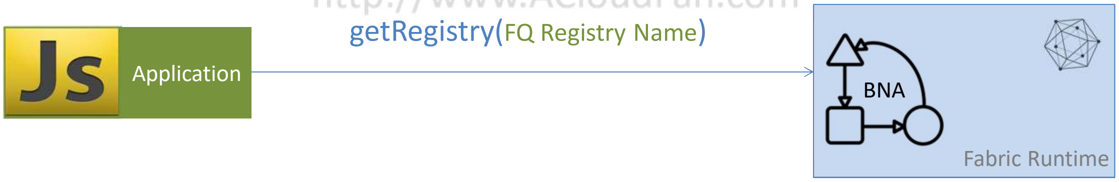
Read only. Stores historian records

IdentityRegistry

Read only. Set of identities on the chain

Client BusinessNetworkConnection

Get an instance of the Client Registry

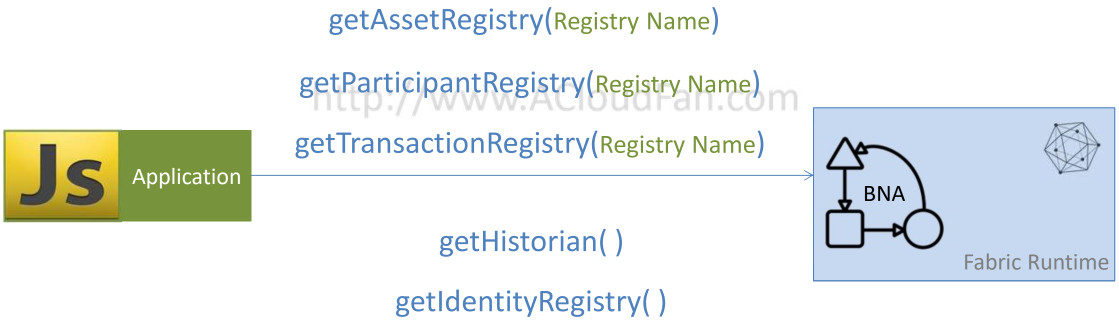


getRegistry(FQ Registry Name)

https://hyperledger.github.io/composer/api/client-businessnetworkconnection.html

Client BusinessNetworkConnection

Get an instance of the Client Registry



getAssetRegistry(Registry Name)

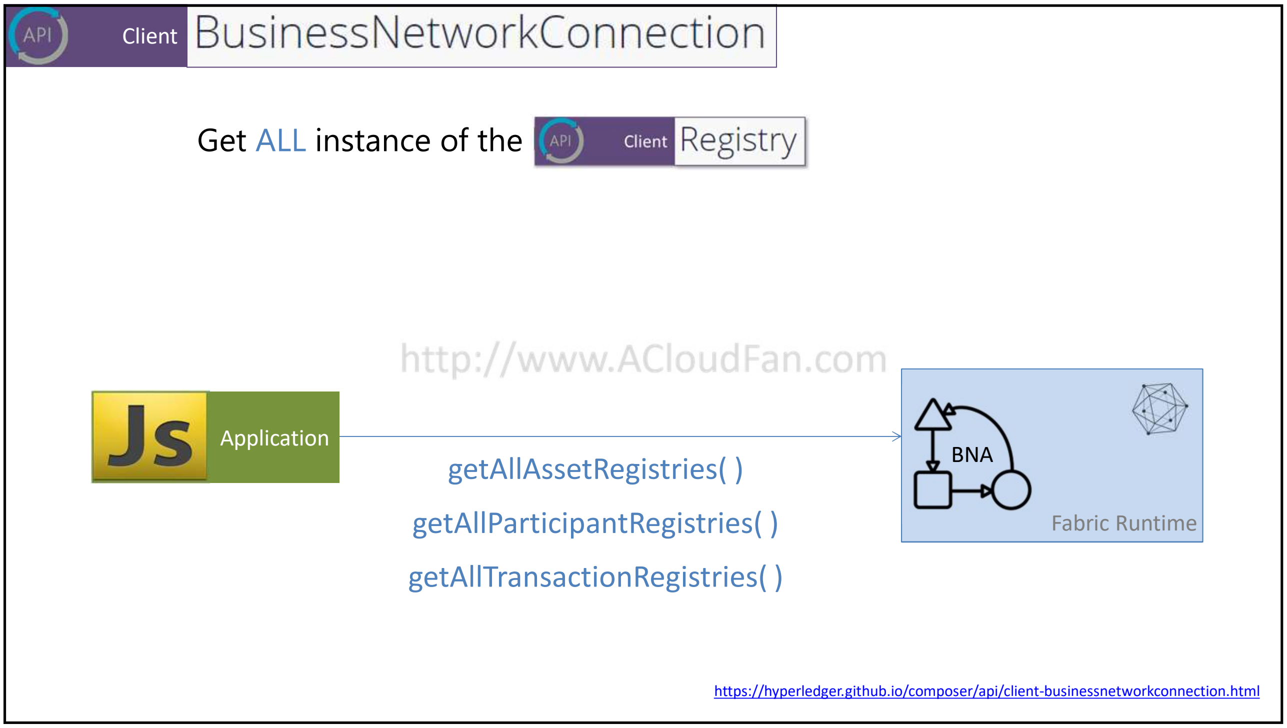
getParticipantRegistry(Registry Name)

getTransactionRegistry(Registry Name)

getHistorian( )

getIdentityRegistry( )

https://hyperledger.github.io/composer/api/client-businessnetworkconnection.html



API Client BusinessNetworkConnection

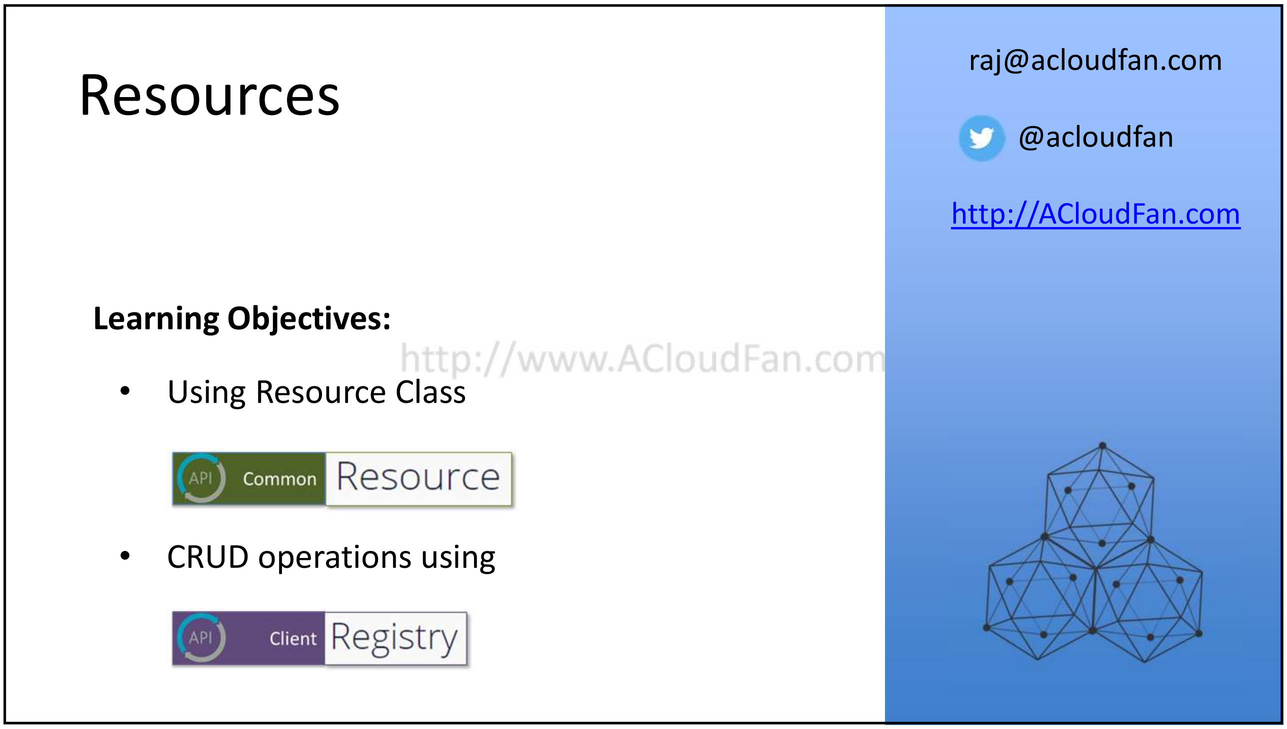
Get ALL instance of the API Client Registry

getAllAssetRegistries( )

getAllParticipantRegistries( )

getAllTransactionRegistries( )

https://hyperledger.github.io/composer/api/client-businessnetworkconnection.html



Resources

raj@acloudfan.com

@acloudfan

http://ACloudFan.com

Learning Objectives:

- Using Resource Class

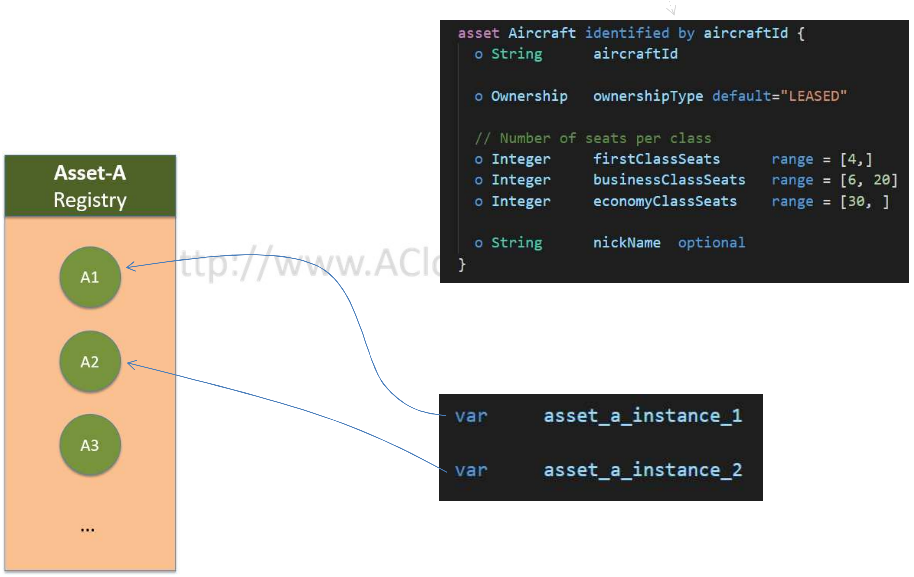
  API Common Resource

- CRUD operations using

  API Client Registry

API Common Resource

Represents an instance of *Type* in the Registry



```
asset Aircraft identified by aircraftId {
  o String      aircraftId

  o Ownership   ownershipType default="LEASED"

  // Number of seats per class
  o Integer     firstClassSeats      range = [4,]
  o Integer     businessClassSeats   range = [6, 20]
  o Integer     economyClassSeats    range = [30, ]

  o String      nickName  optional
}
```

```
var     asset_a_instance_1

var     asset_a_instance_2
```

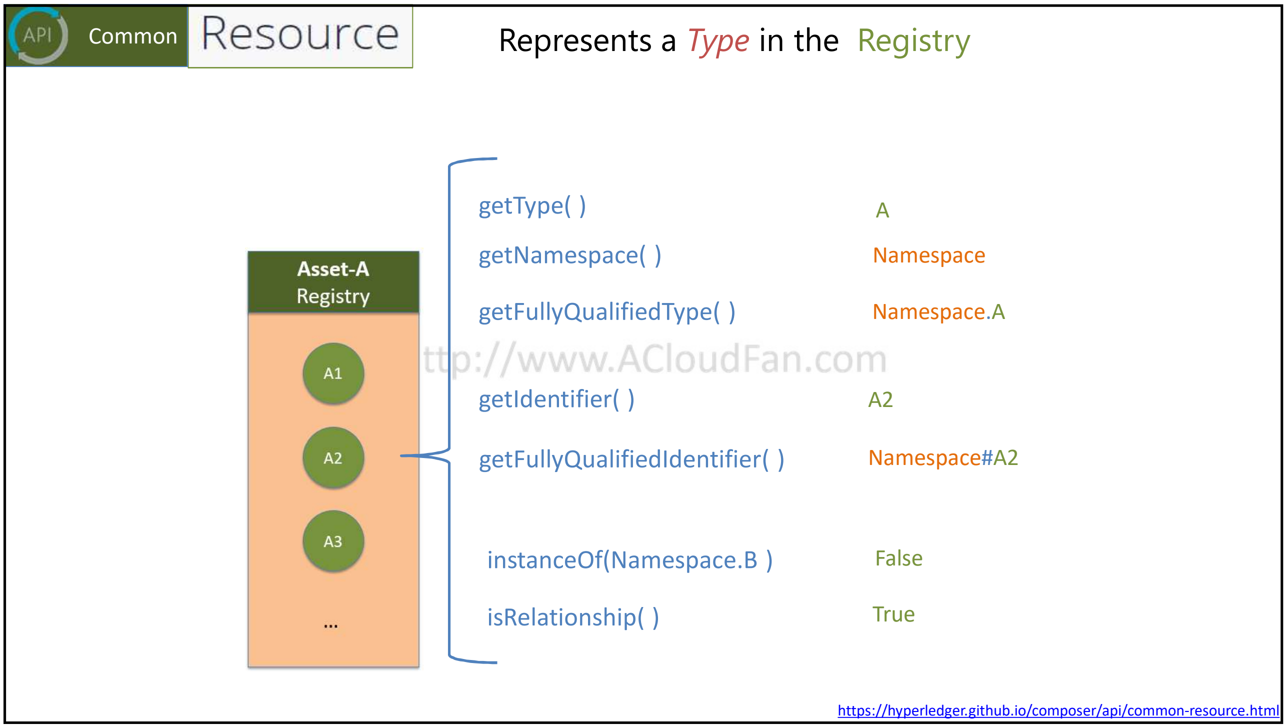
https://hyperledger.github.io/composer/api/common-resource.html

API Common Resource

New Instances created by using the API Runtime Factory

Existing Instances received from API Client Registry

https://hyperledger.github.io/composer/api/common-resource.html

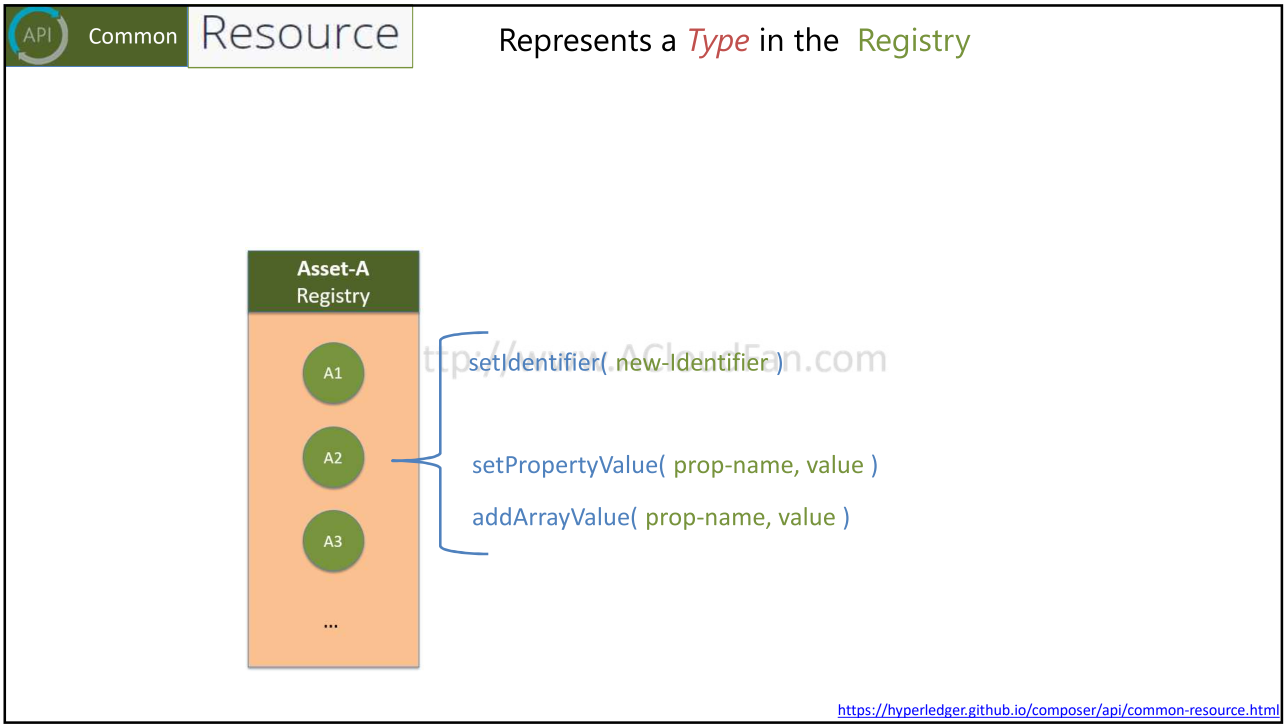


API Common **Resource**

Represents a *Type* in the Registry

getType( )	A
getNamespace( )	Namespace
getFullyQualifiedType( )	Namespace.A
getIdentifier( )	A2
getFullyQualifiedIdentifier( )	Namespace#A2
instanceOf(Namespace.B)	False
isRelationship( )	True

https://hyperledger.github.io/composer/api/common-resource.html



API Common **Resource**

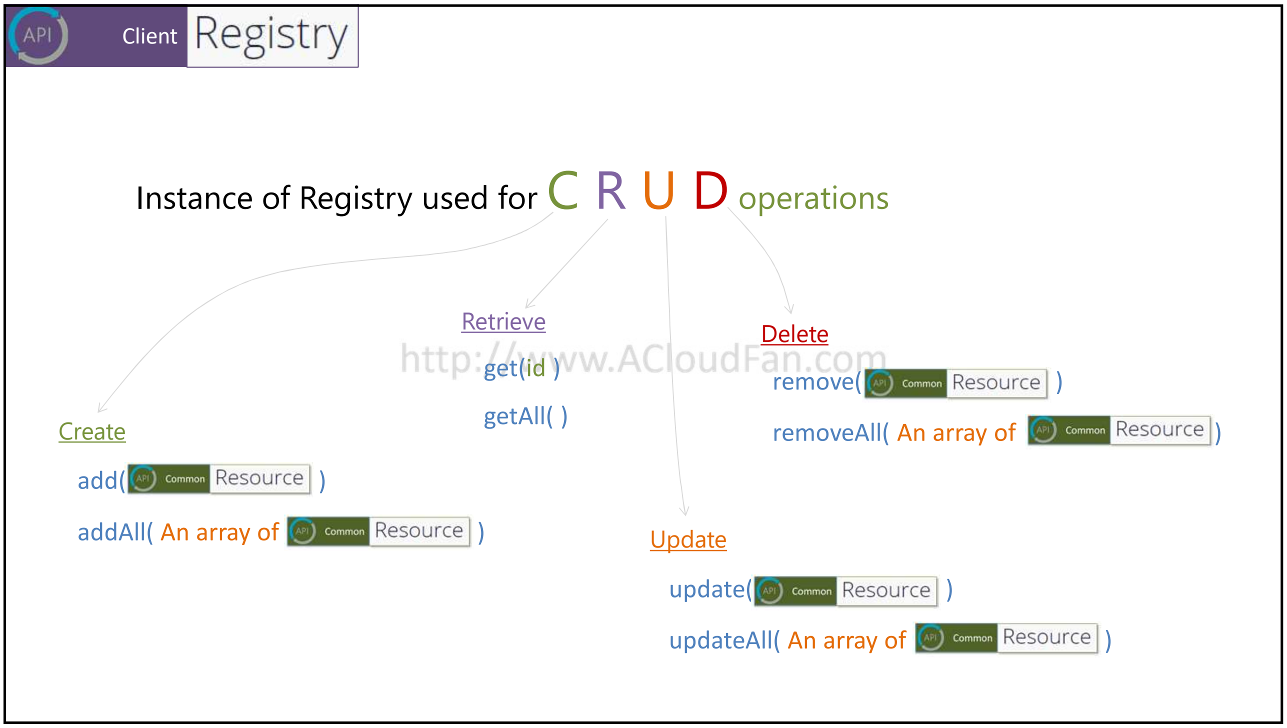
Represents a *Type* in the Registry

setIdentifier(new-Identifier)

setPropertyValue(prop-name, value)

addArrayValue(prop-name, value)

https://hyperledger.github.io/composer/api/common-resource.html



API Client Registry

Instance of Registry used for C R U D operations

Create

add(API Common Resource)

addAll(An array of API Common Resource)

Retrieve

get(id)

getAll( )

Update

update(API Common Resource)

updateAll(An array of API Common Resource)

Delete

remove(API Common Resource)

removeAll(An array of API Common Resource)

Embedded Runtime

Learning Objectives:

- How it works
- Unit Testing Harness

raj@acloudfan.com

@acloudfan

http://ACloudFan.com

Hyperledger Fabric API

https://github.com/acloudfan/**HLF-Fabric-API**

Samples demonstrating use of various Fabric SDK/API

JS ut-harness.js

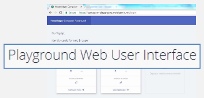
JS test-ut-harness.js

Code shown in video may change over time

Runtimes

Fabric 1.x

Docker based setup



Online and may be locally installed

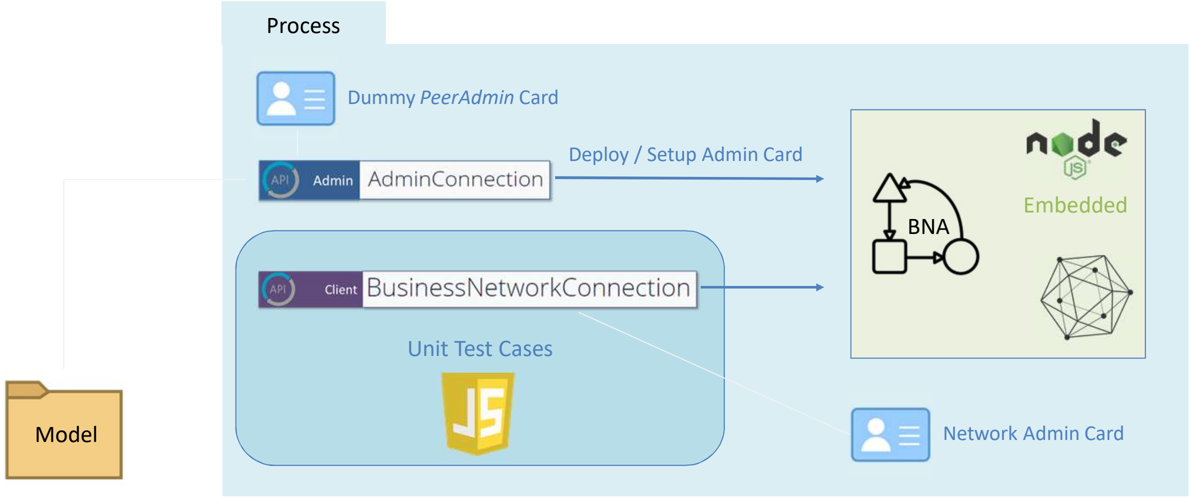
node JS
Embedded

Unit testing

Embedded

- Simulates the Fabric runtime
- Setup alongside the unit testing code
- Uses in-memory persistence for registries
- No special library – its part of the

Embedded



Query API (Client)

raj@acloudfan.com

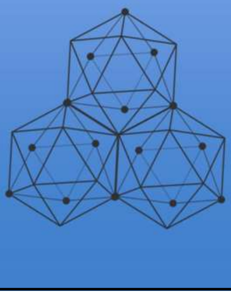
@acloudfan

http://ACloudFan.com

Learning Objectives:

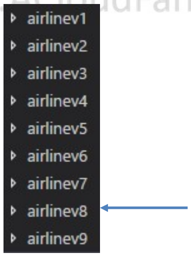
API Client BusinessNetworkConnection

- Invoking named query
- Invoking dynamic query



Incremental Creation of **ACME Air Domain Model**

https://github.com/acloudfan/**HLF-Course-Domain-Model**



```
▸ airlinev1
▸ airlinev2
▸ airlinev3
▸ airlinev4
▸ airlinev5
▸ airlinev6
▸ airlinev7
▸ airlinev8
▸ airlinev9
```

Code shown in video may change over time

Hyperledger Fabric API

https://github.com/acloudfan/**HLF-Fabric-API**

Samples demonstrating use of various Fabric SDK/API

JS client-query.js

Code shown in video may change over time

Queries

Two types of queries

LET'S RECAP...

Named Query

- Defined as part of the Business Network model
- Exposed as REST API by *composer-rest-server* component

Dynamic Query

- Constructed dynamically @ runtime
  - Composer API in Transaction processor function | Client code

Queries

API for executing queries in client code

Named Query

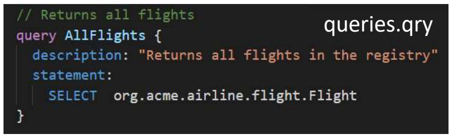
API Client BusinessNetworkConnection .query(named, parameters_object)

Dynamic Query

API Client BusinessNetworkConnection .buildQuery(query_statement)

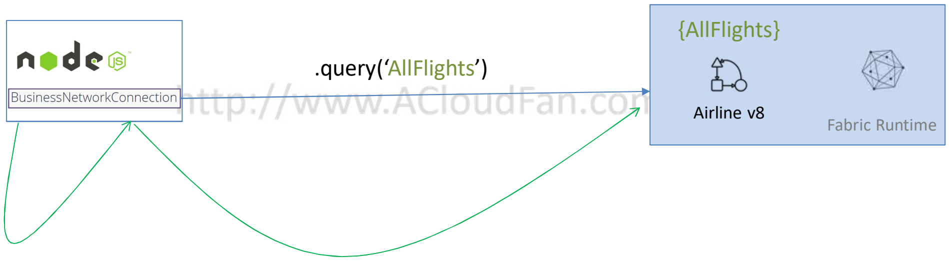
API Client BusinessNetworkConnection .query(queryReturned , parameters_object)

Walkthrough



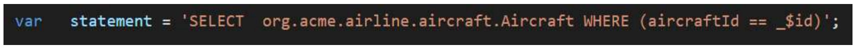
```
// Returns all flights
query AllFlights {
  description: "Returns all flights in the registry"
  statement:
    SELECT  org.acme.airline.flight.Flight
}
```

queries.qry



.buildQuery('statement')

.query(qry_returned, {JSON param object})



```
var   statement = 'SELECT  org.acme.airline.aircraft.Aircraft WHERE (aircraftId == _$id)';
```