

# #10

# Blockchain

240-311 DISTRIBUTED COMPUTERS AND WEB  
TECHNOLOGIES (3-0-6)

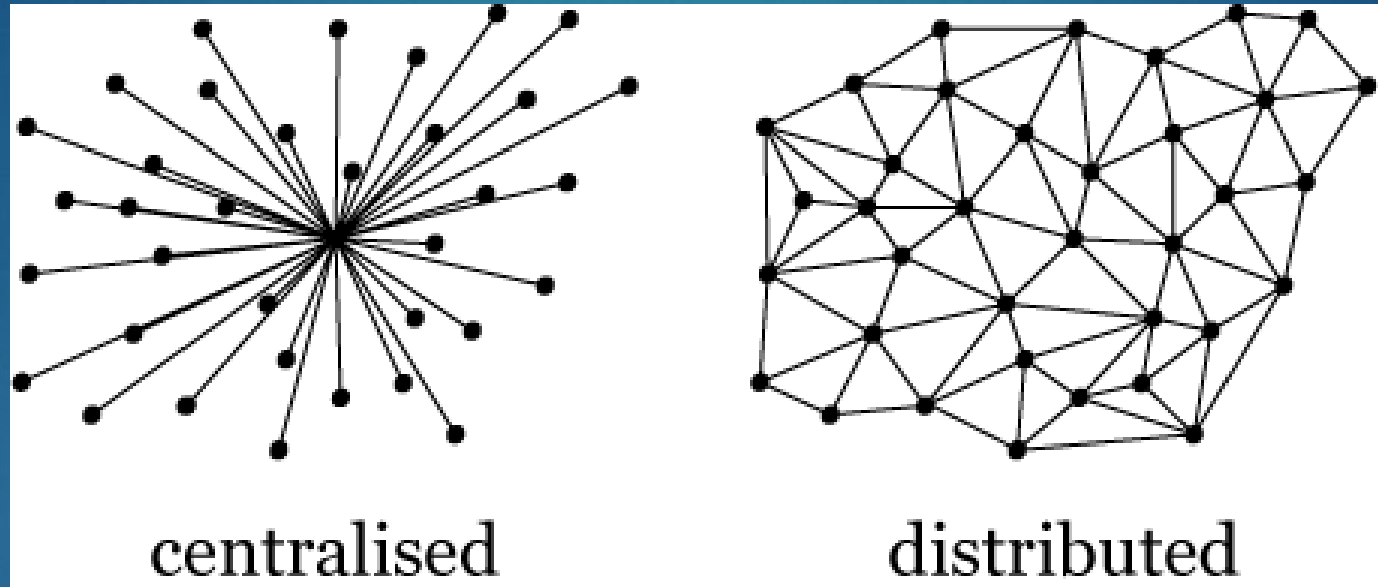
# Outline

2

- ▶ Trust and problem
- ▶ Why Blockchains
- ▶ How it works
- ▶ Blockchain benefits
- ▶ Blockchain types
- ▶ Hyperledger composer
  - ▶ Scenario: car auction

# Trust and problem

3

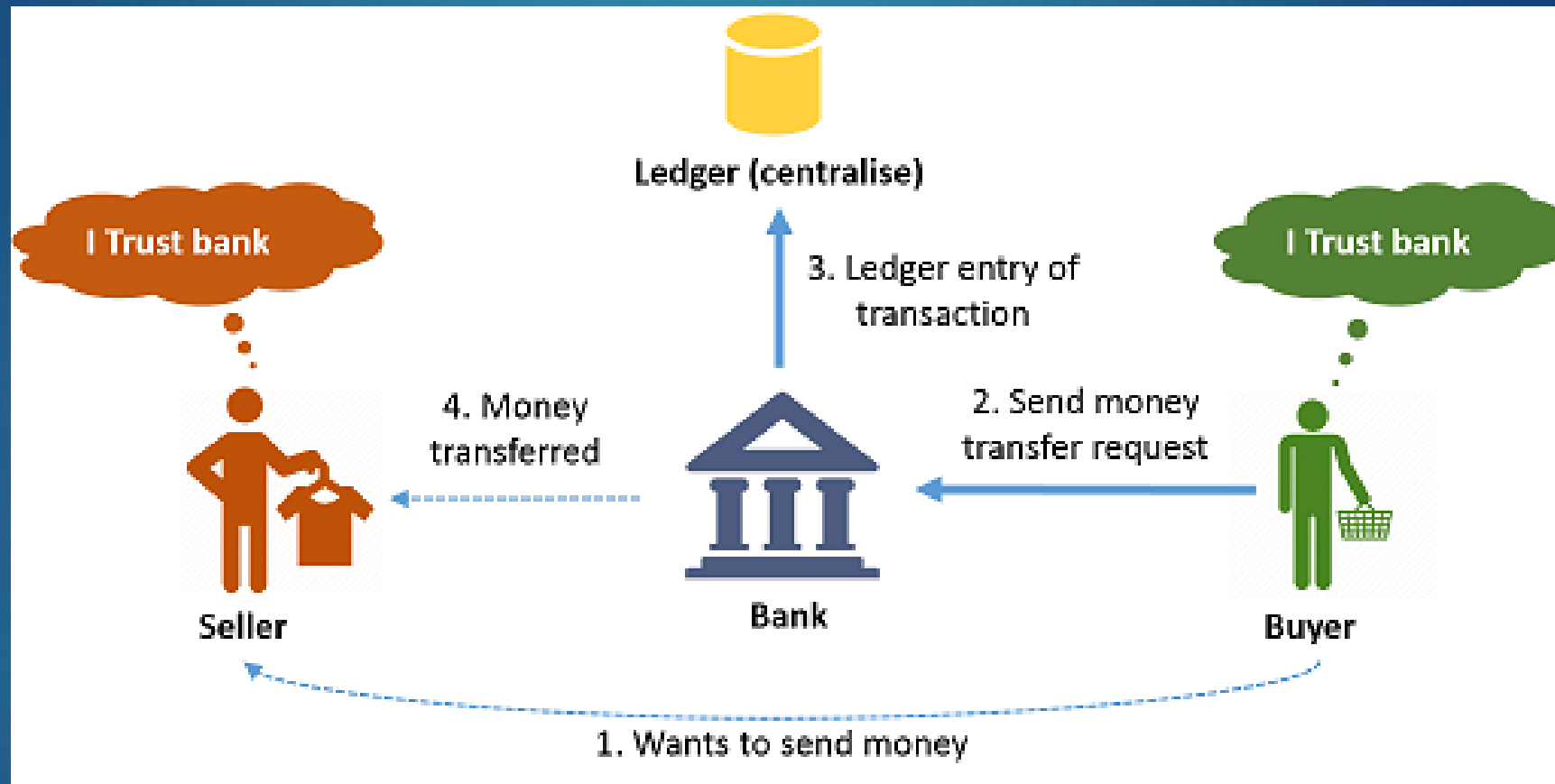


## Cons of centralized architecture

- Fault tolerance
- Attack resistance
- Collusion resistance

# Why Blockchain

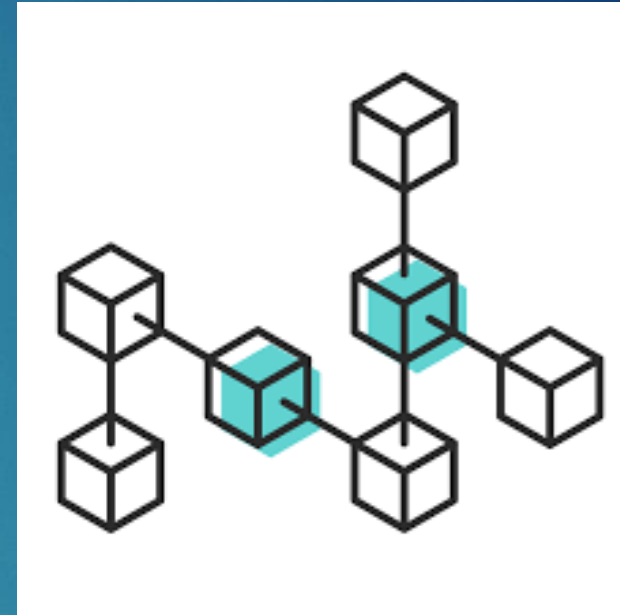
4



# Blockchain

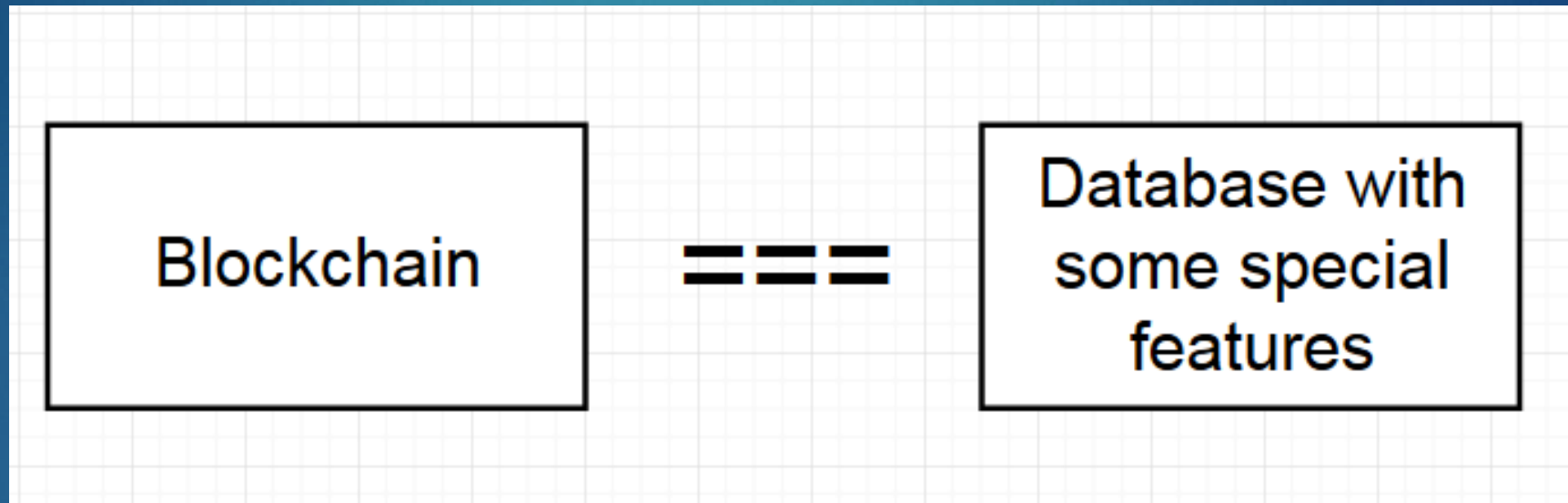
5

- ▶ Blockchain is a growing list of records, called blocks, which are linked using cryptography.
- ▶ Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.



# Blockchain

6



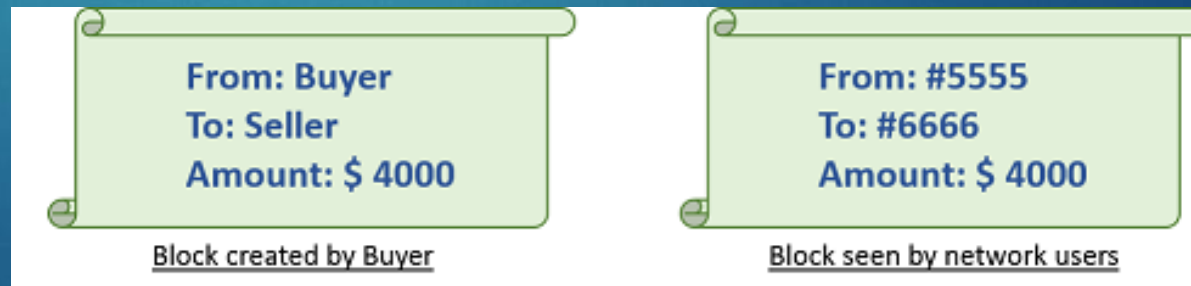
Simple definitions



# How Blockchain works?

7

- ▶ Remove dependency on third party trust
  - ▶ form a group of Members, termed as a NETWORK.
- ▶ Terms
  - ▶ Members
  - ▶ Ledgers
  - ▶ Block



# Blocks are chained (Linked)

8

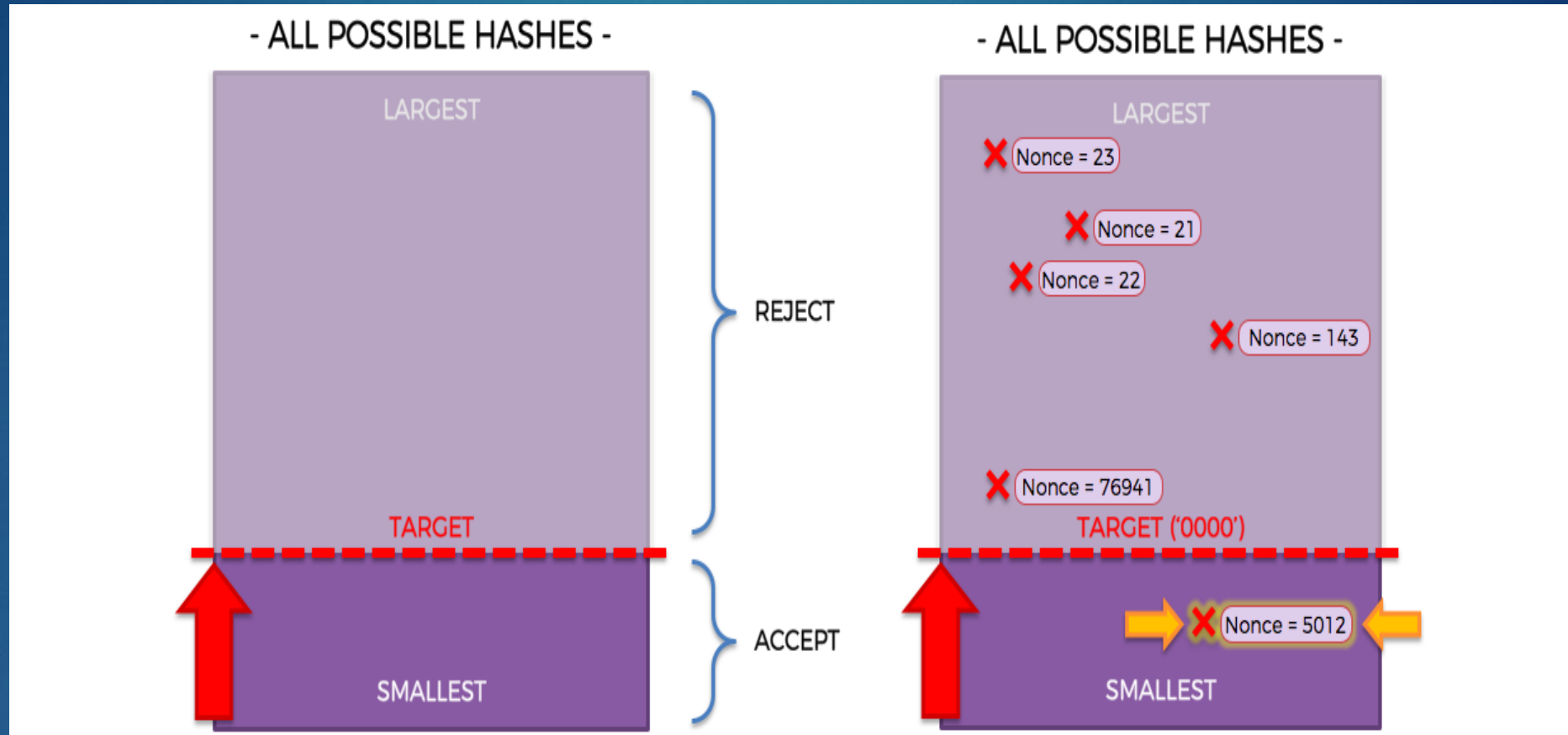


- Hash directly affects the value of the current block's hash.
- if anyone were to tamper with any given block's data, ALL of the following blocks' hashes invalid.



# Nonces

9



- Miners compete to find a Nonce (also called a Golden Nonce)
- 5012 is the smallest one which is closest to '0000' (4 digits)

# Hash demo

10

Blockchain Demo

## SHA256 Hash

**Data:**

**Hash:**

<https://anders.com/blockchain/hash.html>

## 11

[illegible]

Block:

#2

Nonce:

35230

Data:

block 2

Prev:

218acfb3902d9df90125828d4a30748e8af72bd7a37b4eb0a!

Hash:

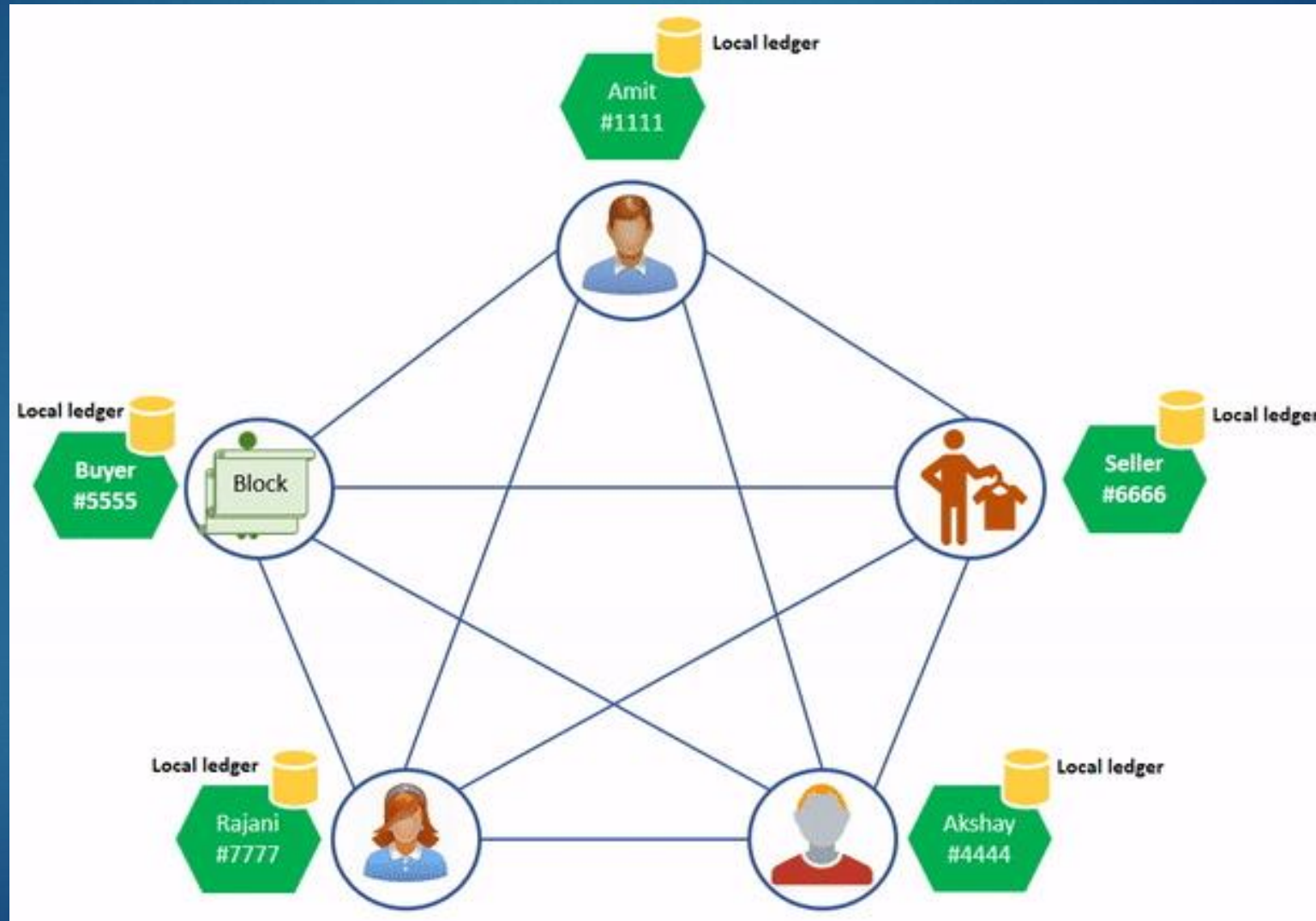
4aed1f24087fe680822975c89784b9313ffead76483d8c77c

Mine

<https://anders.com/blockchain/blockchain.html>

# Distributed Blocks

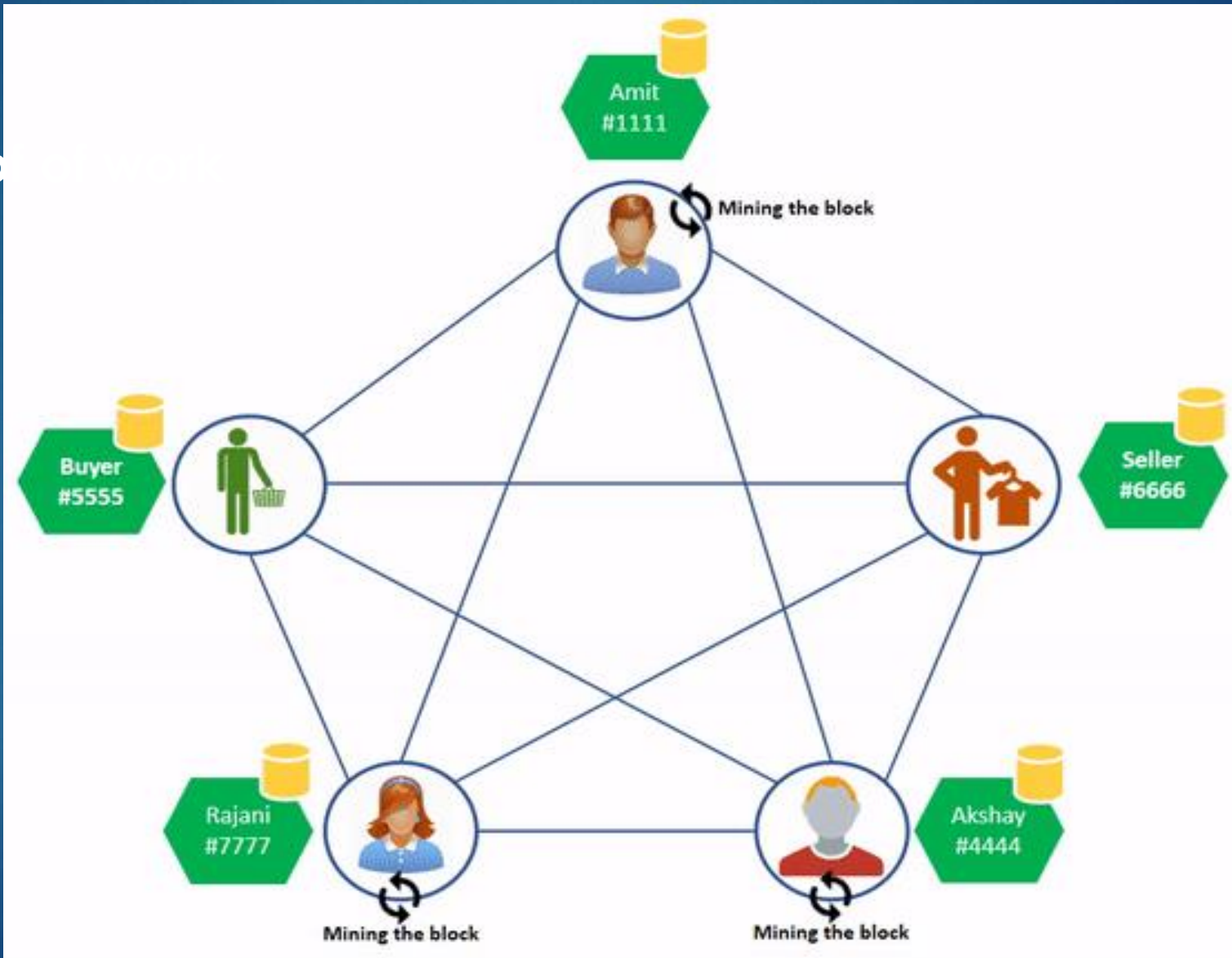
12



# Mining

13

Proof





# Blockchain benefits

14

- ▶ Traceability
- ▶ Enhanced security
  - ▶ Update through consensus
  - ▶ Immutable (process integrity)
- ▶ Efficiency and speed
  - ▶ Fast processing with distributed technologies
- ▶ Reduced costs

# Blockchain scenario examples

15

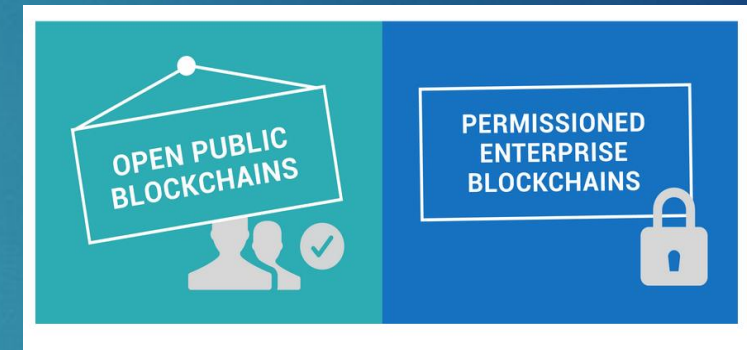
- ▶ Electronic voting
- ▶ Car auction
- ▶ Landlord and title deed
  - ▶ In the case of untrusted government
    - ▶ Blockchain helps to prove that you are the real landlord without a title deed issued by untrusted government
- ▶ Room/Hotel business
  - ▶ No any centralized agent, room owners and renters are directly connected through smart contracts
- ▶ Money changer
  - ▶ Less exchange fee, fast (without physical boundary) and secure

# Blockchain types

16

- ▶ Public Blockchains

- ▶ All participants are anonymous members
  - ▶ Not suitable for business
- ▶ Competition (Proof of Works)
- ▶ Crypto currency & Reward



- ▶ Private Blockchains (or Permissioned Blockchains)

- ▶ Permissioned DLT – Distributed Hyperledgers
- ▶ Identity module & Confidential transaction
- ▶ No Cryptocurrency & programmable (automate business logic)
- ▶ No competition (and incentive) since participants are identified
  - ▶ Participants will be kicked out if they are cheating

# Private Blockchains

17

- ▶ Enterprise blockchains developer
- ▶ Tools for full stack blockchains:
  - ▶ Hyperledger composer
    - ▶ Focus on business network:
      - ▶ participants, identity, assets, transaction
    - ▶ Blockchain applications (DApp)
  - ▶ Hyperledger fabric
    - ▶ Focus on peers, chaincode, consensus
    - ▶ Backend blockchains



# Hyperledger Composer

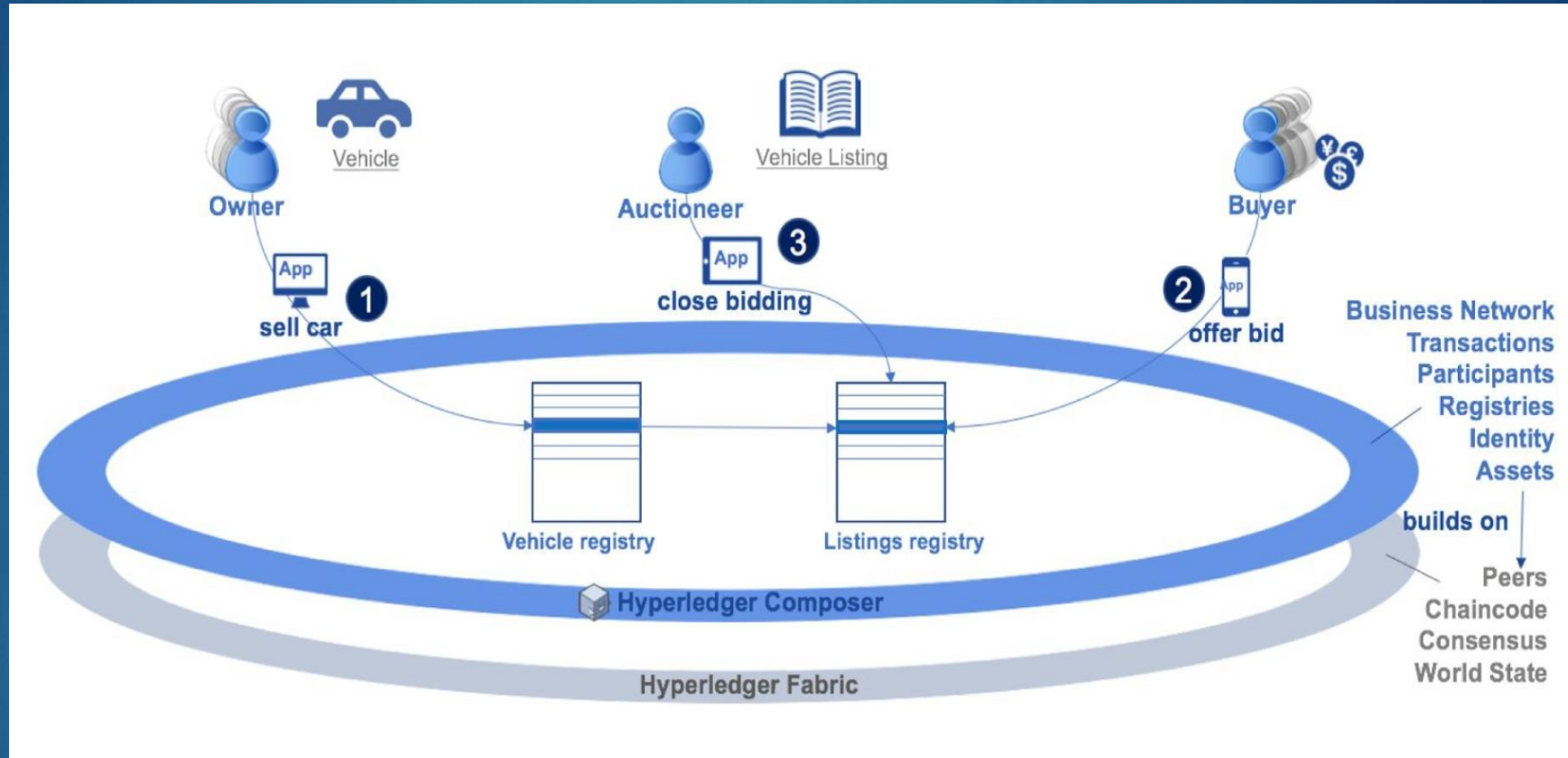
18

- ▶ Hyperledger Composer is an extensive, open development toolset and framework to make developing blockchain applications easier.
- ▶ It simplifies application development on top of the Hyperledger Fabric blockchain infrastructure
  - ▶ which allows components, such as consensus and membership services, to be plug-and-play.
- ▶ Model a business network and integrate existing systems and data with the blockchain applications.



# Hyperledger Composer & Fabric

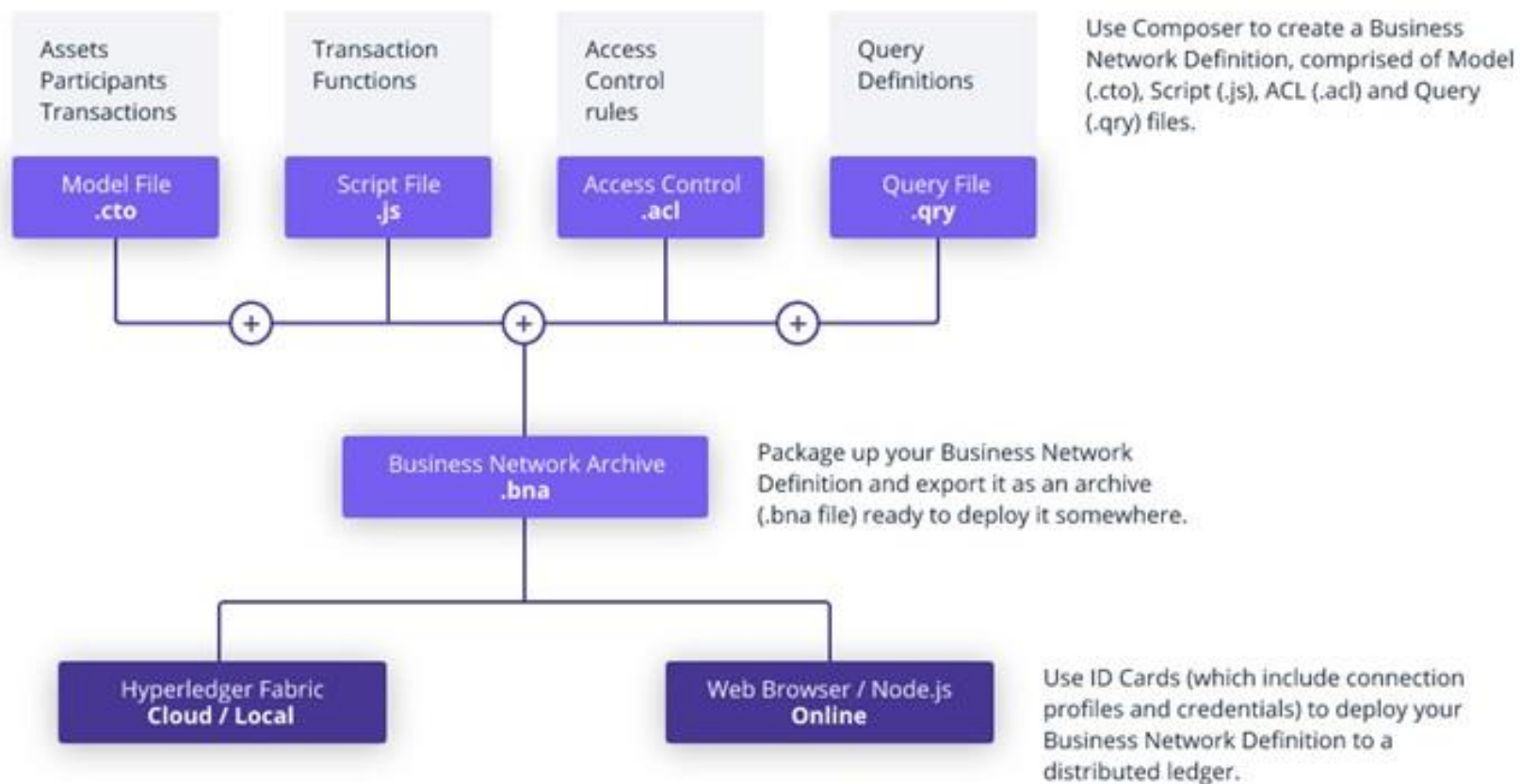
19



# Hyperledger Composer

20

Composer abstracts the Blockchain complexities



# Hyperledger composer experiment

CARAUCION

# Business network: Car Auction

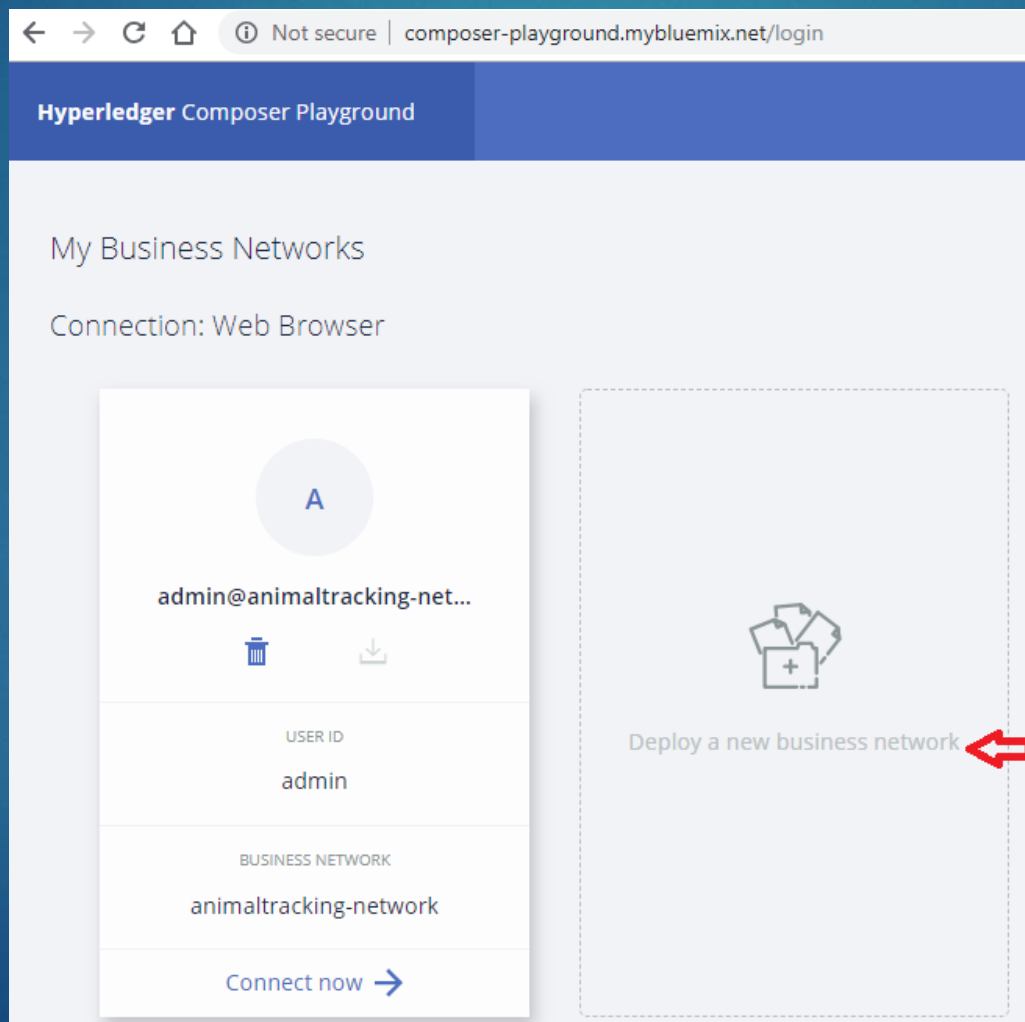
22

- ▶ The model implements:
  - ▶ 3 member participants:
    - ▶ Paul: Owner,
    - ▶ Warodom and Kevin: Buyer (bidding)
  - ▶ Two assets:
    - ▶ A vehicle: a car with id (9999)
    - ▶ A vehicle listing: declare for bidding offer (carListing)
  - ▶ Two transactions:
    - ▶ Making an offer (bid) on a car and closing a bid on an auction.

# Experiment

23

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



Select deploy  
a new business  
network



# Experiment

24

Give your new Business Network a name:

carauktion-network

Describe what your Business Network will be used for:


Car Auction Business Network


Give the network admin card that will be created a name


warodom.w@psu.ac.th

Choose a Business Network Definition to start with:


Choose a sample to play with, start a new project, or import your previous work


  
basic-sample-network


  
empty-business-network


  
Drop here to upload or [browse](#)


Samples on npm

  
animaltracking-network

  
bond-network

  
carauktion-network

  
digitalproperty-network

  
carauktion-network  
Car Auction Business Network

CONNECTION PROFILE

BASED ON  
carauktion-network

Car Auction Business Network

Contains: 3 Participant Types, 2 Asset Types, and 2 Transaction Types

Deploy

- Select
1. carauktion-network
  2. Enter an email
  3. Deploy

# Connect to carauktion

25

A

warodom.w@psu.ac.th

USER ID

admin

BUSINESS NETWORK

carauction-network

Connect now →

Connect now

- ▶ Hyperledger fabric for blockchain network is provided by the web site.
- ▶ We focus on Business network (caracution) as DApp only.
- ▶ Scenario:
  - ▶ 1 member who wants to sale a car
  - ▶ 2 members who wants to buy a car from this auction system

# Test carauction network

26

Web carauction-network

DefineTest

admin

PARTICIPANTS

Auctioneer

Member

ASSETS

Vehicle

VehicleListing


TRANSACTIONS

All Transactions

Submit Transaction

Participant registry for org.acme.vehicle.auction.Auctioneer

+ Create New Participant

ID	Data
<div><p>This registry is empty!</p><p>To create resources in this registry click create new at the top of this page</p></div>	

Ready to implement business network

# Create participants

27

- ▶ 3 member participants
  - ▶ Paul Gilbert as a car owner
  - ▶ Warodom Werapun as a buyer
  - ▶ Kevin Durant as another buyer
- ▶ No Auctioneers

+ Create New Participant

In registry: **org.acme.vehicle.auction.Member**

JSON Data Preview

```
1  {
2    "$class": "org.acme.vehicle.auction.Member",
3    "balance": 100,
4    "email": "paul@psu.ac.th",
5    "firstName": "Paul",
6    "lastName": "Gilbert"
7  }
```

# Create 2 more participants

28

```
1  {
2    "$class": "org.acme.vehicle.auction.Member",
3    "balance": 90000,
4    "email": "warodom@psu.ac.th",
5    "firstName": "Warodom",
6    "lastName": "Werapun"
7  }
```

```
1  {
2    "$class": "org.acme.vehicle.auction.Member",
3    "balance": 10000,
4    "email": "kevin@psu.ac.th",
5    "firstName": "Kevin",
6    "lastName": "Durant"
7  }
```



# All member participants

29

ID	Data
kevin@psu.ac.th	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": 10000,   "email": "kevin@psu.ac.th",   "firstName": "Kevin",   "lastName": "Durant" }</pre> <div>Collapse</div>
paul@psu.ac.th	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": 100,   "email": "paul@psu.ac.th",   "firstName": "Paul",   "lastName": "Gilbert" }</pre> <div>Show All</div>
warodom@psu.ac.th	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": 90000,   "email": "warodom@psu.ac.th",   "firstName": "Warodom",   "lastName": "Werapun" }</pre> <div>Show All</div>

# Create asset and asset listing

30

In registry: **org.acme.vehicle.auction.Vehicle**

JSON Data Preview

```
1 {
2   "$class": "org.acme.vehicle.auction.Vehicle",
3   "vin": "9999",
4   "owner":
5     "resource:org.acme.vehicle.auction.Member#paul@psu.ac.th"
```

Asset id  
( car id)

In registry: **org.acme.vehicle.auction.VehicleListing**

JSON Data Preview

```
1 {
2   "$class": "org.acme.vehicle.auction.VehicleListing",
3   "listingId": "carListing",
4   "reservePrice": 1000,
5   "description": "Paul wants to sale his car",
6   "state": "FOR_SALE",
7   "vehicle": "resource:org.acme.vehicle.auction.Vehicle#9999"
8 }
```

Car  
owner

# Asset and asset listing

31

car

Data

```
{
  "$class": "org.acme.vehicle.auction.Vehicle",
  "vin": "9999",
  "owner": "resource:org.acme.vehicle.auction.Member#paul@psu.ac.th"
}
```

carListing

Data

```
{
  "$class": "org.acme.vehicle.auction.VehicleListing",
  "listingId": "carListing",
  "reservePrice": 1000,
  "description": "Paul wants to sell his car",
  "state": "FOR_SALE",
  "vehicle": "resource:org.acme.vehicle.auction.Vehicle#9999"
}
```

# Submit transaction

32

Transaction Type

Offer

JSON Data Preview

```
1  {
2    "$class": "org.acme.vehicle.auction.Offer",
3    "bidPrice": 80000,
4    "listing":
5      "resource:org.acme.vehicle.auction.VehicleListing#carListing",
6    "member":
7      "resource:org.acme.vehicle.auction.Member#kevin@psu.ac.th"
8  }
```

Transaction Type

Offer

JSON Data Preview

```
1  {
2    "$class": "org.acme.vehicle.auction.Offer",
3    "bidPrice": 200000,
4    "listing":
5      "resource:org.acme.vehicle.auction.VehicleListing#carListing",
6    "member":
7      "resource:org.acme.vehicle.auction.Member#warodom@psu.ac.th"
8  }
```

## Bidding Offer

- ▶ Kevin places 'Bidding offer' at 80,000 baht
- ▶ Warodom offer at 200,000 baht
- ▶ Paul will close bidding and let see the winner

# After place bidding offer

33

Data

```
{
  "$class": "org.acme.vehicle.auction.VehicleListing",
  "listingId": "carListing",
  "reservePrice": 1000,
  "description": "Paul wants to sale his car",
  "state": "FOR_SALE",
  "offers": [
    {
      "$class": "org.acme.vehicle.auction.Offer",
      "bidPrice": 80000,
      "listing": "resource:org.acme.vehicle.auction.VehicleListing#carListing",
      "member": "resource:org.acme.vehicle.auction.Member#kevin@psu.ac.th",
      "transactionId": "58ff01da-3982-4882-98dd-f6deb1e61d6e",
      "timestamp": "2019-02-02T06:50:24.945Z"
    },
    {
      "$class": "org.acme.vehicle.auction.Offer",
      "bidPrice": 200000,
      "listing": "resource:org.acme.vehicle.auction.VehicleListing#carListing",
      "member": "resource:org.acme.vehicle.auction.Member#warodom@psu.ac.th",
      "transactionId": "b0baf88d-f9b5-42aa-8bd9-0c5677f911f3",
      "timestamp": "2019-02-02T06:57:40.741Z"
    }
  ],
  "vehicle": "resource:org.acme.vehicle.auction.Vehicle#9999"
}
```



# Submit transaction

34

Transaction Type CloseBidding ▼

JSON Data Preview

```
1 {  
2   "$class": "org.acme.vehicle.auction.CloseBidding",  
3   "listing":  
4     "resource:org.acme.vehicle.auction.VehicleListing#carListing"  
}
```

Data

```
{  
  "$class": "org.acme.vehicle.auction.VehicleListing",  
  "listingId": "carListing",  
  "reservePrice": 1000,  
  "description": "Paul wants to sale his car",  
  "state": "SOLD",  
  "vehicle": "resource:org.acme.vehicle.auction.Vehicle#99999"  
}
```

State is  
changed

# Place close bidding

35

```
async function closeBidding(closeBidding) { // eslint-disab
  const listing = closeBidding.listing;
  if (listing.state !== 'FOR_SALE') {
    throw new Error('Listing is not FOR SALE');
  }
  // by default we mark the listing as RESERVE_NOT_MET
  listing.state = 'RESERVE_NOT_MET';
  let highestOffer = null;
  let buyer = null;
  let seller = null;
  if (listing.offers && listing.offers.length > 0) {
    // sort the bids by bidPrice
    listing.offers.sort(function(a, b) {
      return (b.bidPrice - a.bidPrice);
    });
    highestOffer = listing.offers[0];
    if (highestOffer.bidPrice >= listing.reservePrice) {
      // mark the listing as SOLD
      listing.state = 'SOLD';
      buyer = highestOffer.member;
      seller = listing.vehicle.owner;
      // update the balance of the seller
    }
  }
}
```

Smart contract in  
**lib/logic.js** is called:  
async function  
closeBidding(closeBi  
dding)

# Asset owner is changed

36

carListing is

fully updated  
to be "SOLD".

Data

```
{
  "$class": "org.acme.vehicle.auction.Vehicle",
  "vin": "9999",
  "owner": "resource:org.acme.vehicle.auction.Member#warodom@psu.ac.th"
}
```

ID

Data

carListing

```
{
  "$class": "org.acme.vehicle.auction.VehicleListing",
  "listingId": "carListing",
  "reservePrice": 1000,
  "description": "Paul wants to sale his car",
  "state": "SOLD",
  "vehicle": "resource:org.acme.vehicle.auction.Vehicle#9999"
}
```

# Money is transferred

37

ID	Data
kevin@psu.ac.th	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": 10000,   "email": "kevin@psu.ac.th",   "firstName": "Kevin",   "lastName": "Durant" }</pre> <a>Show All</a>
paul@psu.ac.th	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": 200100,   "email": "paul@psu.ac.th",   "firstName": "Paul",   "lastName": "Gilbert" }</pre> <div>Update</div> <a>Collapse</a>
warodom@psu.ac.th	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": -110000,   "email": "warodom@psu.ac.th",   "firstName": "Warodom",   "lastName": "Werapun" }</pre> <a>Show All</a>

# Check a buyer wallet

38

```
async function makeOffer(offer) { // eslint-disable-line no-unused-vars
  let listing = offer.listing;
  if (listing.state !== 'FOR_SALE') {
    throw new Error('Listing is not FOR SALE');
  }
  if (!listing.offers) {
    listing.offers = [];
  }
  listing.offers.push(offer);

  if ( offer.bidPrice > offer.member.balance )
    throw new Error('Not enough money!!');

  // save the vehicle listing
  const vehicleListingRegistry = await getAssetRegistry('org.acme.vehicle.auction.VehicleListing');
  await vehicleListingRegistry.update(listing);
}
```

Edit lib/logic.js to  
check **bidPrice** with  
member **balance**



# Not allow to place bidding

39

Transaction Type Offer ▼

JSON Data Preview

```
1  {
2    "$class": "org.acme.vehicle.auction.Offer",
3    "bidPrice": 20000,
4    "listing":
5      "resource:org.acme.vehicle.auction.VehicleListing#carListing",
6    "member":
7      "resource:org.acme.vehicle.auction.Member#warodom@psu.ac.th"
8  }
```

☐ Optional Properties

Error: Not enough money!!

# References

40

- ▶ <https://www.c-sharpcorner.com/article/basics-of-blockchain/>
- ▶ <https://en.wikipedia.org/wiki/Blockchain>
- ▶ <https://medium.com/swlh/how-does-bitcoin-blockchain-mining-work-36db1c5cb55d>
- ▶ <https://anders.com/blockchain>
- ▶ <https://hyperledger.github.io/composer/latest/tutorials/playground-tutorial.html>