Supply Blockchain–

valuecentric

Team :

BITS OF COIN

Agenda

- Supply Chain and Block Chain
- ValueCentric’s Supply Chain
- Our Solution

BITS OF COIN

Supply Chain and Blockchain…

BITS OF COIN

Supply Chain and Blockchain-

What is common?

- Track packages from start to end
- Track timely sale of perishable items
- Geolocation of package
- Decentralized
- Access to package details restricted
- Audit Trails
- Expiry of contracts

BITS OF COIN

Value Centric Supply Chain ...

Value Centric Supply Chain-

Special Requirements

- Limited Access
- Ownership of Supply Chain
- Analytics on Supply Chain Data
- Access Restrictions

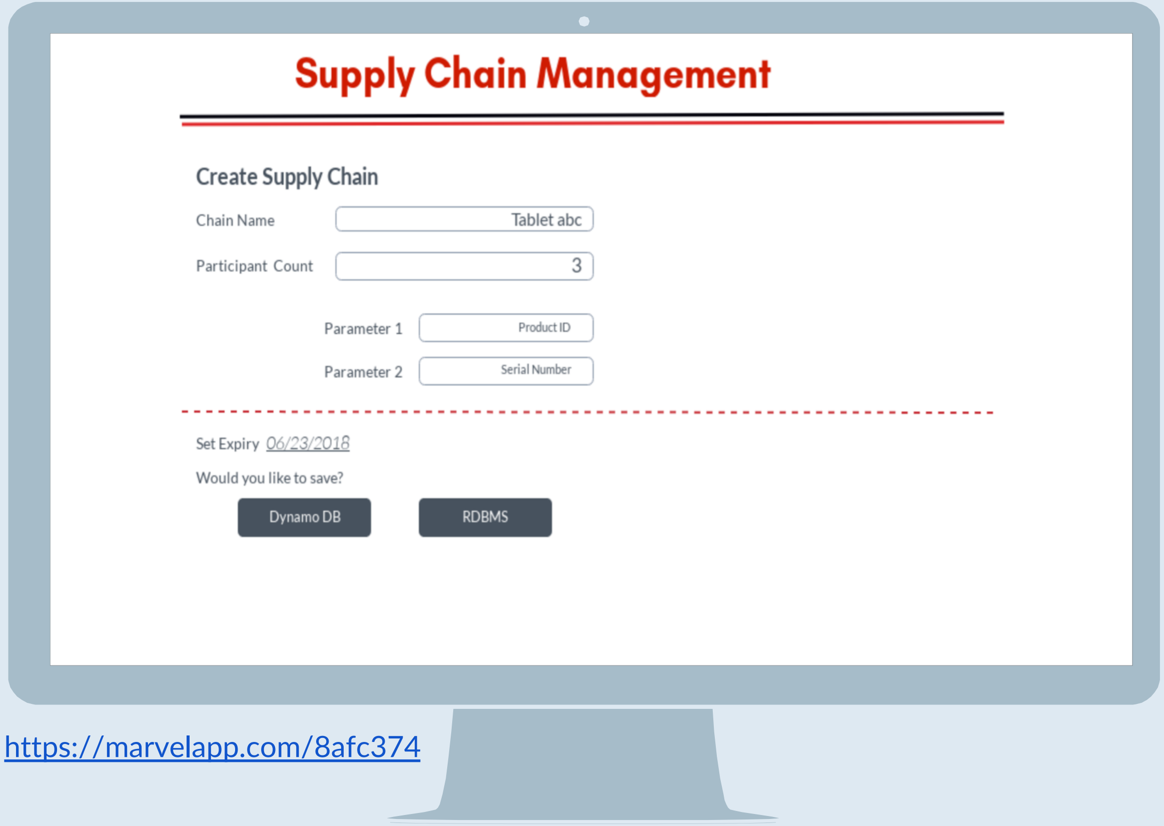
Our Solution …

BITS OF COIN

Generic Website for Blockchain

- Create Blockchain relevant to your business
- Supply information to a user interface
- Solidity code is generated for your case

BITS OF COIN



Supply Chain Management

Create Supply Chain

Chain Name: Tablet abc

Participant Count: 3

Parameter 1: Product ID

Parameter 2: Serial Number

Set Expiry 06/23/2018

Would you like to save?

https://marvelapp.com/8afc374

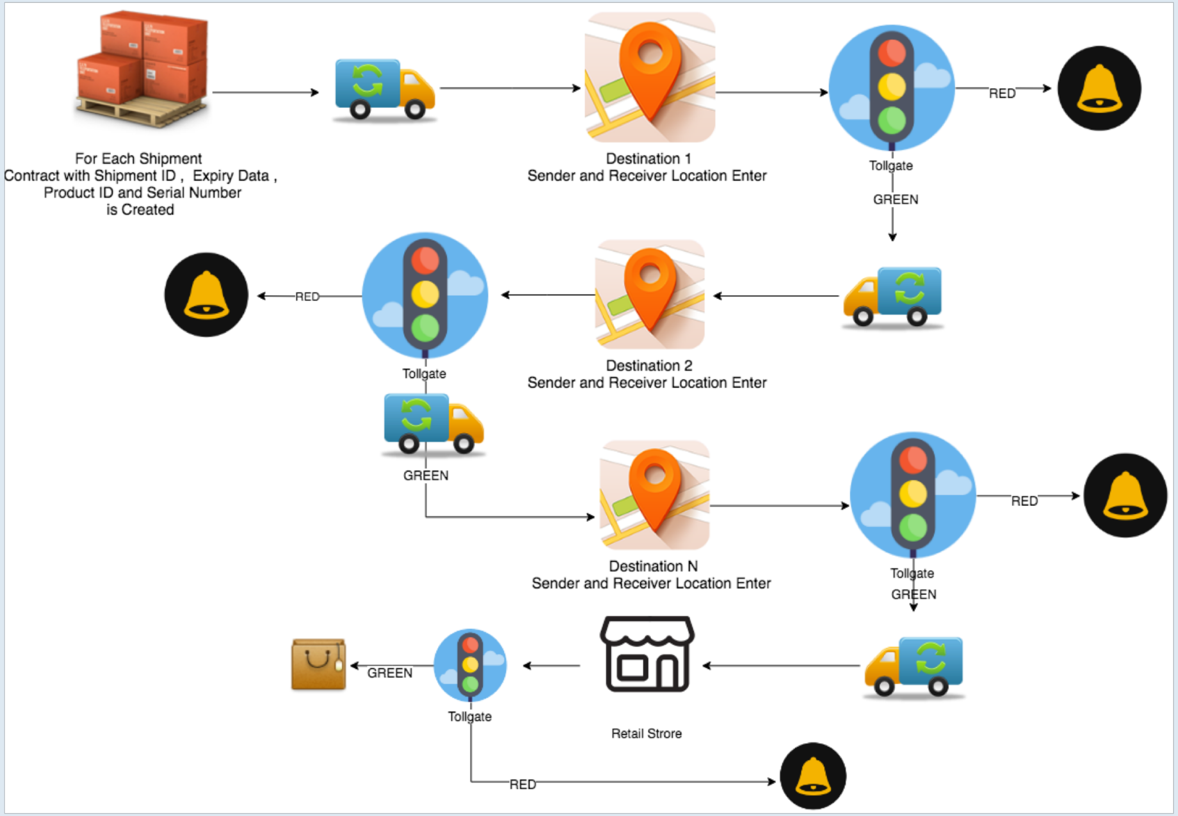
BITS OF COIN

ValueCentric Blockchain

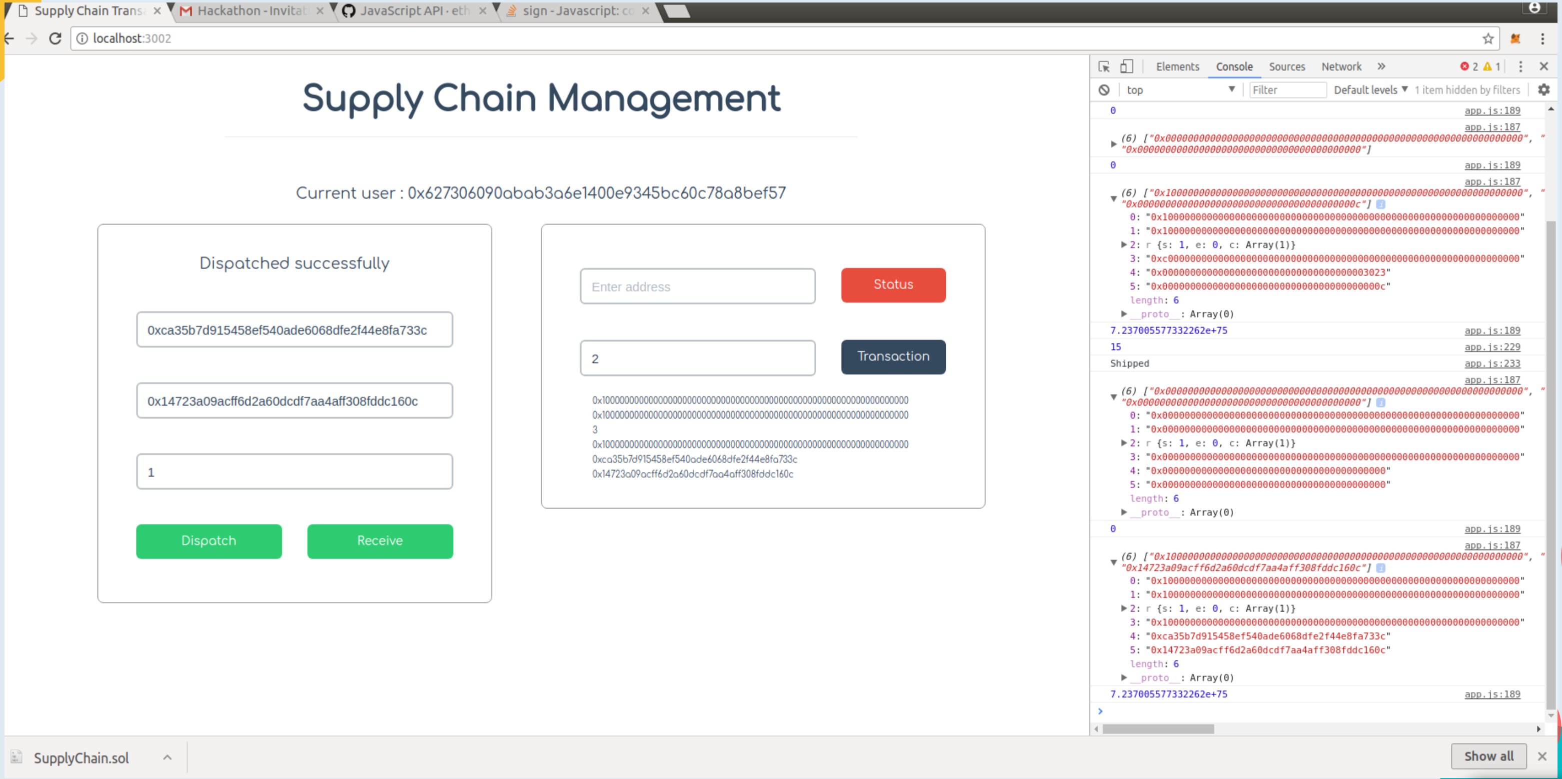
- Saved State for each transaction
- Tollgates
- Expiry of Shipment
- GeoLocation Tracking
- Audit and Analytics

BITS OF COIN

Flow Chart


For Each Shipment
Contract with Shipment ID , Expiry Data ,
Product ID and Serial Number
is Created
Destination 1
Sender and Receiver Location Enter
Tollgate
RED
GREEN
RED
Tollgate
Destination 2
Sender and Receiver Location Enter
GREEN
Destination N
Sender and Receiver Location Enter
Tollgate
RED
GREEN
GREEN
Tollgate
Retail Strore
RED

BITS OF COIN



Supply Chain Management

Current user : 0x627306090abab3a6e1400e9345bc60c78a8bef57

Dispatched successfully

0xca35b7d915458ef540ade6068dfe2f44e8fa733c

0x14723a09acff6d2a60dcdf7aa4aff308fddc160c

1

Dispatch | Receive

Enter address | Status

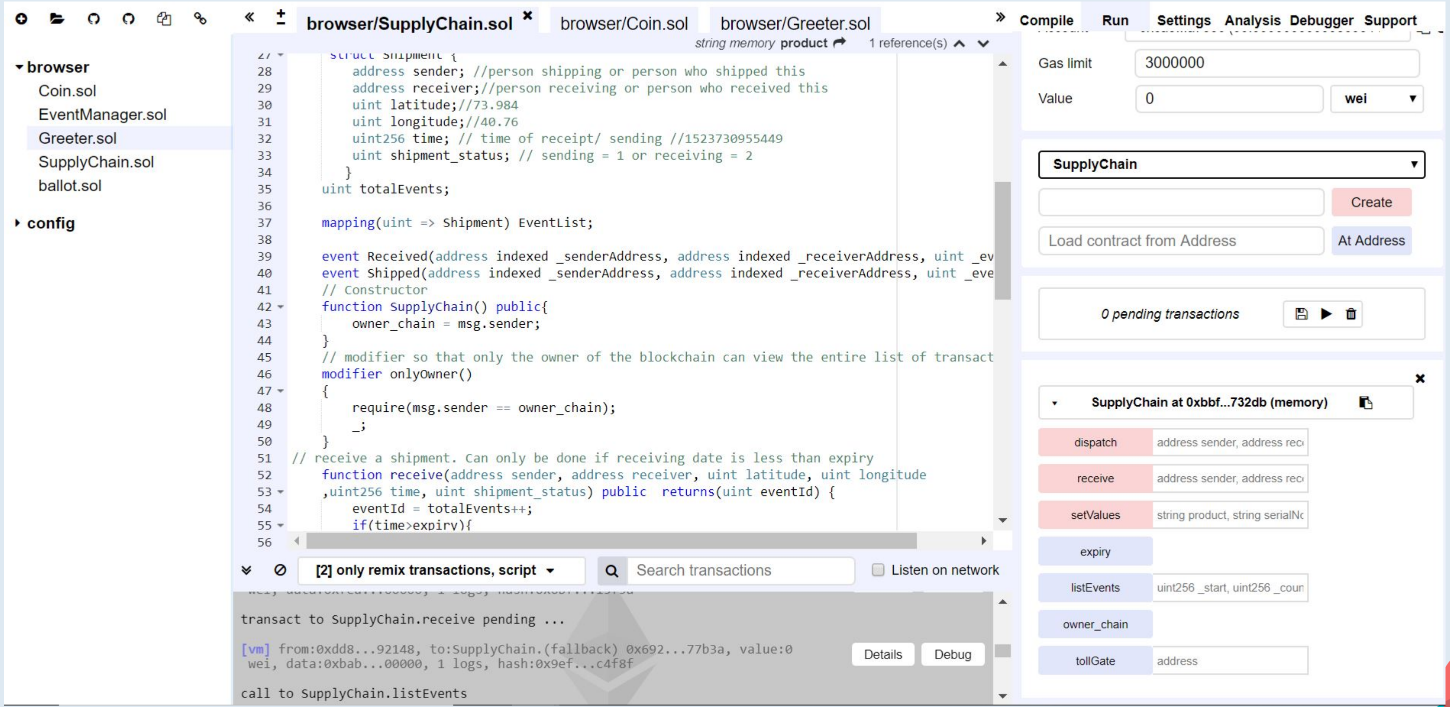
2 | Transaction

```
0x1000000000000000000000000000000000000000000000000000000000000000
0x1000000000000000000000000000000000000000000000000000000000000000
3
0x1000000000000000000000000000000000000000000000000000000000000000
0xca35b7d915458ef540ade6068dfe2f44e8fa733c
0x14723a09acff6d2a60dcdf7aa4aff308fddc160c
```

```
0                                                                        app.js:189
                                                                         app.js:187
▶ (6) ["0x0000000000000000000000000000000000000000000000000000000000000000", "
  "0x0000000000000000000000000000000000000000"]
0                                                                        app.js:189
                                                                         app.js:187
▼ (6) ["0x1000000000000000000000000000000000000000000000000000000000000000", "
  "0x000000000000000000000000000000000000000c"]
    0: "0x1000000000000000000000000000000000000000000000000000000000000000"
    1: "0x1000000000000000000000000000000000000000000000000000000000000000"
  ▶ 2: r {s: 1, e: 0, c: Array(1)}
    3: "0xc000000000000000000000000000000000000000000000000000000000000000"
    4: "0x0000000000000000000000000000000000003023"
    5: "0x000000000000000000000000000000000000000c"
    length: 6
  ▶ __proto__: Array(0)
7.237005577332262e+75                                                    app.js:189
15                                                                       app.js:229
Shipped                                                                  app.js:233
                                                                         app.js:187
▼ (6) ["0x0000000000000000000000000000000000000000000000000000000000000000", "
  "0x0000000000000000000000000000000000000000"]
    0: "0x0000000000000000000000000000000000000000000000000000000000000000"
    1: "0x0000000000000000000000000000000000000000000000000000000000000000"
  ▶ 2: r {s: 1, e: 0, c: Array(1)}
    3: "0x0000000000000000000000000000000000000000000000000000000000000000"
    4: "0x0000000000000000000000000000000000000000"
    5: "0x0000000000000000000000000000000000000000"
    length: 6
  ▶ __proto__: Array(0)
0                                                                        app.js:189
                                                                         app.js:187
▼ (6) ["0x1000000000000000000000000000000000000000000000000000000000000000", "
  "0x14723a09acff6d2a60dcdf7aa4aff308fddc160c"]
    0: "0x1000000000000000000000000000000000000000000000000000000000000000"
    1: "0x1000000000000000000000000000000000000000000000000000000000000000"
  ▶ 2: r {s: 1, e: 0, c: Array(1)}
    3: "0x1000000000000000000000000000000000000000000000000000000000000000"
    4: "0xca35b7d915458ef540ade6068dfe2f44e8fa733c"
    5: "0x14723a09acff6d2a60dcdf7aa4aff308fddc160c"
    length: 6
  ▶ __proto__: Array(0)
7.237005577332262e+75                                                    app.js:189
>
```

Basic GUI

BITS OF COIN



Gas limit 3000000

Value 0 wei

```
    struct Shipment {
        address sender; //person shipping or person who shipped this
        address receiver;//person receiving or person who received this
        uint latitude;//73.984
        uint longitude;//40.76
        uint256 time; // time of receipt/ sending //1523730955449
        uint shipment_status; // sending = 1 or receiving = 2
      }
    uint totalEvents;

    mapping(uint => Shipment) EventList;

    event Received(address indexed _senderAddress, address indexed _receiverAddress, uint _ev
    event Shipped(address indexed _senderAddress, address indexed _receiverAddress, uint _eve
    // Constructor
    function SupplyChain() public{
        owner_chain = msg.sender;
    }
    // modifier so that only the owner of the blockchain can view the entire list of transact
    modifier onlyOwner()
    {
        require(msg.sender == owner_chain);
        _;
    }
// receive a shipment. Can only be done if receiving date is less than expiry
    function receive(address sender, address receiver, uint latitude, uint longitude
    ,uint256 time, uint shipment_status) public  returns(uint eventId) {
        eventId = totalEvents++;
        if(time>expiry){
```

```
transact to SupplyChain.receive pending ...

[vm] from:0xdd8...92148, to:SupplyChain.(fallback) 0x692...77b3a, value:0
 wei, data:0xbab...00000, 1 logs, hash:0x9ef...c4f8f

call to SupplyChain.listEvents
```

Remix Code

BITS OF COIN

Thank you ...