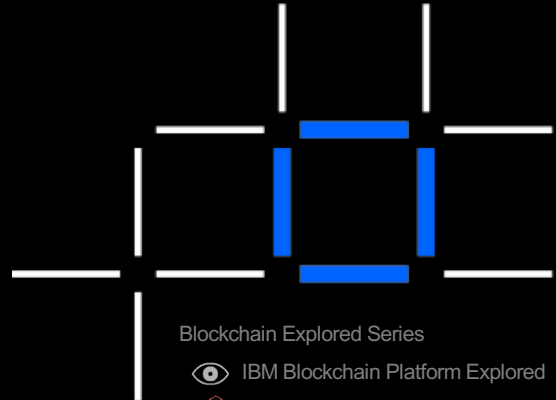







Modeling blockchain applications

Turning business concepts into technical concepts



Blockchain Explored Series

-  IBM Blockchain Platform Explored
-  **Modeling Applications**
-  Architectures Explored
-  Fabric Explored
-  What's New in Tech

V1.0, 1 March 2019

IBM **Blockchain**



What is modeling

Understand the concepts behind modelling and why modeling is important



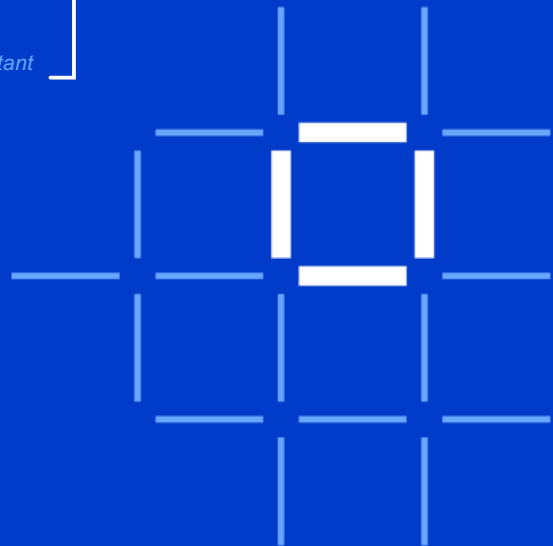
Modeling in blockchain

Modeling the artefacts for a blockchain solution



Using models

Models and the process of invoking transactions

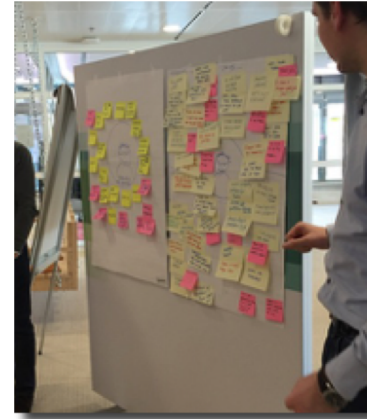


IBM **Blockchain**



Blockchain topics

- Consider the topics we've discussed for blockchain solutions so far:
 - The **business problem** we're trying to solve
 - The **participants** involved (users and organizations)
 - The **assets**
 - The **transactions**, underpinned by **contracts**
- The goal now is to move these topics into to a machine readable form and eventual deployment to a blockchain system

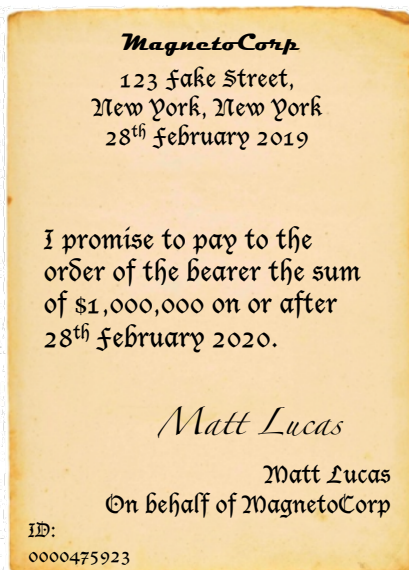


IBM **Blockchain**

IBM

3

Example: Commercial Paper



Business Problem?

- Commercial paper is a means of providing short term financing to companies
- Trust requirement and well-defined business network make a good fit for blockchain

Participants?

- MagnetoCorp (Issuing organization)
- Matt Lucas (MagnetoCorp employee)
- "the bearer" (could be many of these)

Assets?

- The Commercial Paper (!)
- \$1,000,000

Transactions?

- Issue Buy Redeem

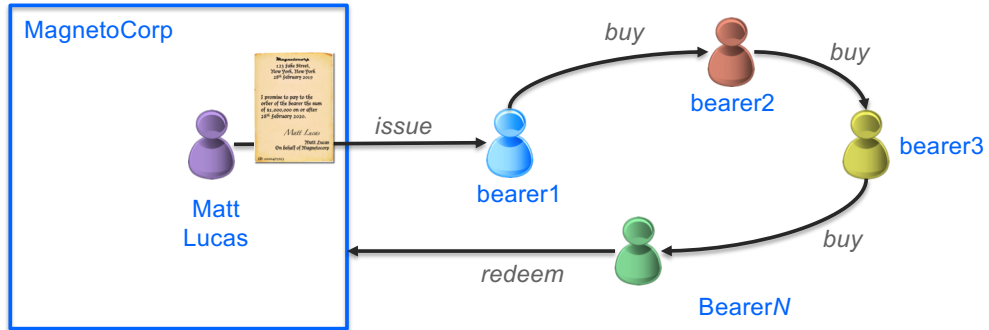
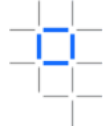
IBM **Blockchain**

IBM

4

Commercial Paper: Transaction Lifecycle

(omitting cash flows)

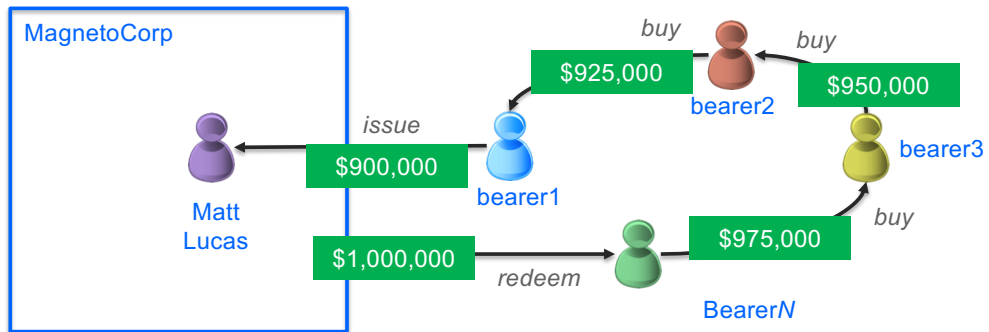
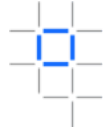


IBM Blockchain

IBM

5

Commercial Paper: Cash Lifecycle

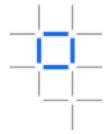


IBM Blockchain

IBM

6

Example: Commercial Paper



Q: How do we get from these business concepts to something that can run on a blockchain?

A: We will *model* them

Business Problem?

- Commercial paper is a means of providing short term financing to companies
- Trust requirement and well-defined business network makes a good fit for blockchain

Participants?

- Magnetocorp (Issuing organization)
- Matt Lucas (Magnetocorp employee)
- "the bearer" (could be many of these)

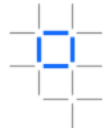
Assets?

- The Commercial Paper
- \$1,000,000

Transactions?

- Issue Buy Redeem

Modeling is the mapping of business concepts into technical concepts...



Assets	Contracts	Transactions	Business Networks	Participants
Data structures in a pre-agreed format	Algorithm to modify asset state	Single invocation of a contract's algorithm	Computer network topology (c.f. internet)	Digital certificate for each user/organization

- Models don't need to be *complete*, but they need to be *sufficient* to solve the problem at hand
 - e.g. You don't need to model each cylinder of an engine if you're tracking the overall owner of a car

Modeling assets



“Physical” asset

Commercial Paper	
ID:	0000475923
Issuer:	MagnetoCorp
Owner:	MagnetoCorp
Face value:	\$1,000,000
Maturity date:	2020-02-28

Attributes

```

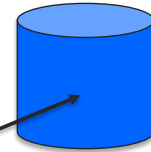
"commercialPaper" : {
  "id": "string",
  "issuer": "string",
  "owner": "string",
  "faceValue": "integer",
  "maturityDate": "date" }
    
```

Model definition

```

{ "0000475923",
  "MagnetoCorp",
  "MagnetoCorp",
  "1000000",
  "2020-02-28" }
    
```

Instance



World state

- Assets correspond to data that is stored in the blockchain “world state”
 - World state is just a database; assets can be created, modified and deleted
- Approach: Business network agrees the attributes of the assets that will be shared and a blockchain solution architect models them using an appropriate language
 - E.g. UML, JSON schema, class definition, CTO file

Modeling contracts & transactions

- **Contracts** are the actions that modify assets in the world state
 - Algorithms with inputs and outputs
 - Should use modelled assets
- Each invocation of a contract is a **transaction** and logged on the blockchain
 - Once recorded on the blockchain, transaction logs cannot be modified or deleted
- Hyperledger Fabric calls the code that implements contract logic **chaincode**.

```

// Redeem commercial paper
@param {Context} ctx the transaction context
@param {string} issuer commercial paper issuer
@param {integer} paperNumber paper number for this issuer
@param {string} redeemingOwner redeeming owner of paper
@param {string} redeemDateTime time paper was redeemed

async redeem(ctx, issuer, paperNumber, redeemingOwner, redeemDateTime) {
  let paperKey = CommercialPaper.makeKey([issuer, paperNumber]);
  let paper = await ctx.paperList.getPaper(paperKey);

  // Check paper is not REDEEMED
  if (paper.isRedeemed()) {
    throw new Error('Paper ' + issuer + paperNumber + ' already redeemed');
  }

  // Verify that the redeemer owns the commercial paper before redeeming it
  if (paper.getOwner() !== redeemingOwner) {
    throw new Error('Paper ' + issuer + paperNumber + ' not owned by redeemer');
  }
}
    
```

Issue

Creates a new commercial paper instance
Inputs: issuer, ID, issue date/time, maturity date/time, face value
Outputs: None
Design: Once created, the new asset's details are stored in the world state

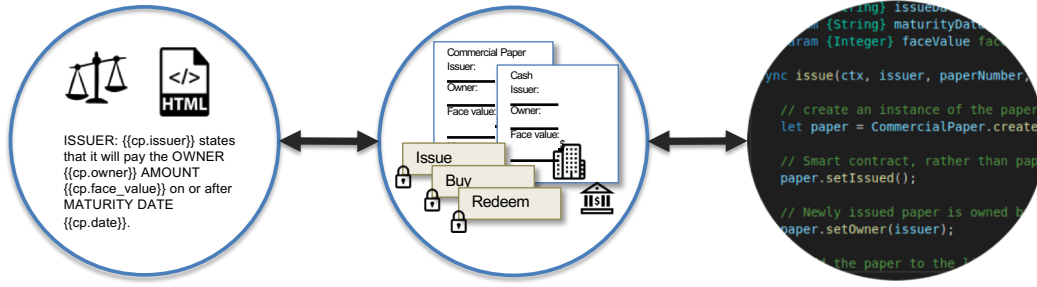
Buy

Transfers ownership of a commercial paper instance
Inputs: issuer, ID, current owner, new owner, price, issue date/time, maturity date/time, face value
Outputs: None
Design:
 The seller's cash balance is incremented by the price
 The buyer's cash balance is decremented by the price
 The buyer becomes the owner of the commercial paper
 Update the commercial paper instance in the world state

Redeem

Transfers cash matching the redemption value to the current owner
Inputs: issuer, ID, current owner, redemption date/time
Outputs: None
Design:
 The paper must not have already been redeemed
 The issuer's cash balance is decremented by the redemption value
 The owner's cash balance is incremented by the redemption value
 The paper is marked as redeemed
 Update the commercial paper instance in the world state

Contracts vs. models vs. chaincode

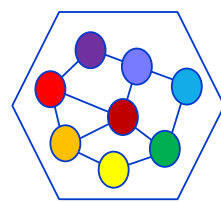


Legal Contract	Model	Chaincode
For the lawyer	For the business user	For the developer

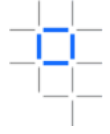
- Chaincode should be closely linked to legal contracts and models
- They could be thought of as being different renderings of the same information for different constituencies
- Explicit links between the three artefacts can ensure correctness of implementation
- For Hyperledger Fabric, these three artefacts can be collectively thought of as the **smart contract**

Modeling the business network

- Organizations in the business network map to **peers** in a technical network
- Peers maintain a consistent ledger and world state, and signs, executes and validates transactions
 - They are run as network services, similar to web servers on the internet
 - Hosted by members of the business network or by third-party cloud providers
 - *Gateway* peers provide the entry point to the network for client applications
- It is overly simplistic to assume a 1:1 mapping between organizations and peers
 - Larger organizations usually have multiple peers for high availability
 - Smaller organizations might delegate responsibility for managing peers and signing transactions to another organization
 - Maybe even an organization that is not part of the business network!
 - Peers shared across organizations are sometimes referred to as *trust anchors*



Modeling users and organizations



- Every organization, user and system component has an **identity** in the blockchain network
 - Used for identifying actors in the network, signing and encrypting information
- There are standard ways of expressing identity (e.g. X.509 certificates)
- Things to consider:
 - Who is a user in the blockchain system
 - How certificates are issued (and revoked)
 - The relationship between users and organizations, and between organizations

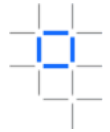


IBM **Blockchain**

IBM

13

(Simplified) transaction process



1. Each administrator **deploys** chaincode to peers in the network
2. The end-user application **connects** to a gateway peer
3. The application **queries** the available chaincodes on the peer
4. The application **invokes** an available chaincode with a set of input parameters
5. The blockchain network **executes** the chaincode, agrees the output and updates the ledger/world state on all peers
6. The peer **notifies** the application is notified that the transaction has been completed

```

1 const gateway = new Gateway();
2 const wallet = new FileSystemWallet('./WALLETS/wallet');
3 try {
4   await gateway.connect(ccp, {
5     identity: 'admin',
6     wallet: wallet
7   });
8
9   const network = await gateway.getNetwork('market1234');
10  const contract = await network.getContract('commercial-paper');
11
12  // issue commercial paper
13  const paper = await contract.submitTransaction('issue', 'ibm', '1000000', '2019-01-31');
14
15 } catch (error) {
16   console.log(error);
17 } finally {
18   gateway.disconnect();
19 }

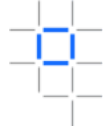
```

IBM **Blockchain**

IBM

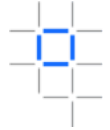
14

Some of the big questions still to cover...



1. Each administrator **deploys** chaincode to peers in the network
 2. The end-user application **connects** to a gateway peer
 3. The application **queries** the available chaincodes on the peer
 4. The application **invokes** an available chaincode with a set of input parameters
 5. The blockchain network **executes** the chaincode, agrees the output and updates the ledger/world state on all peers
 6. The peer **notifies** the application is notified that the transaction has been completed
- Which peers - how to handle privacy and confidentiality?
 - How does this process work?
 - Is the process the same for all blockchain implementations?
 - How does the blockchain prevent tampering?
 - Why is it called a blockchain anyway?

Summary



- Modeling is the art of mapping business concepts into technical concepts
- It is useful to model blockchain solution elements, as it helps pave the way to the implementation and helps stakeholders understand the solution
 - Assets
 - Contracts
 - Transactions
 - Business Networks
 - Participants

Thank you

IBM Blockchain

www.ibm.com/blockchain

developer.ibm.com/blockchain

www.hyperledger.org

© Copyright IBM Corporation 2019. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represents only goals and objectives. IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

