CST4125: Blockchain Development
Week: 2
Title: Data Modelling

Dr Ian Mitchell

smerf.net
Bedfordshire,
UK

January 2023

Staff Etiquette

Academics

- Record
  - Chatrooms
  - Live
  - Attendance
- Mute control
- Access control
- No anonymity
- Share personal information via screen shares

Copyright

Student Etiquette

Do's

- Behave as normal, be respectful
- No anonymity
- First and last names to identify you
- Kindness/Difficulty
- Be patient, some one may have technical issues
- Mute microphone, unless speaking
- Use chatroom appropriately
- Keep video on, especially when talking
- Tolerance

Don'ts

- Share personal information
- Try not to multi-task
- Behave inappropriately
- Bully other students
- Disruption
- No eating

Labs

- Complete exercise together
- All leave room
- Try exercise
- Have questions or queries
- Enter waiting room for 1-2-1

Contact and Office Hours

Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Contact and Office Hours

Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Office Hours

- During term time only
- When: Winter Term: Mondays 1100-1300hrs
- Please read notifications or emails
- There are occassions that these could be arranged online, e.g., due to industrial action or inclement weather

Deadlines

				Feedback	
Description	Submission	Weight	Deadline	Formative	Summative
1. Hyperledger	MyLearning	50%	14th April 2023	LW11-12	10/05/2023
2. Ethereum	MyLearning	50%	8th July 2023	LW23-24	28/07/2023
Resits	MyLearning	50-100%	14th August 2023	None	None
Deferals	MyLearning	50-100%	14th August 2023	None	None

Lecture

Aims

Essentially, there are three steps for blockchain development, which we will learn over the coming weeks. The first step is Data Modelling, using CTO.

Lecture Objectives

Knowledge

- Differentiate between Assets, Participants, Transactions and Events
- Model and develop blockchain code to implement a data structure
- Background and context of Hyperledger frameworks
- See how blockchain can be applied by looking at use cases.

Skills

- Implement and develop the data model for a blockchain using ConcerTO (CTO).

Hyperledger

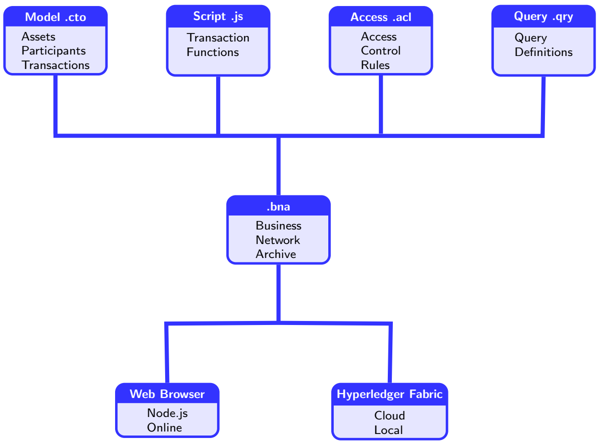
Open Source

- Free
- Distribute
- Copy
- Edit and Modify
- License

Open Governance

- Transparent in decisions
- Development processes
- Maintainers
- Community
- The Steering Committee (TSC)
- See how blockchain can be applied by looking at use cases.
- Active contributors are eligible to participate
- Bring your nominations
- Cast your vote
- Not just a piece of code, its a movement

Hyperledger Structure



Hyperledger Architecture [**hyperledger:1**]

- Consensus
- Smart Contract
- Communication
- Data Store
- Cryptography
- Policy
- Identity
- API
- Interoperation

Hyperledger Architecture [**hyperledger:1**]

Consensus

- Verify the correctness of the set of transactions
- A block is composed of multiple transactions
- Concur with other nodes
- Which of these can be trusted?
- Also provides some ordering.
- Consensus algorithm:

Hyperledger Architecture [**hyperledger:1**]

Consensus

- Verify the correctness of the set of transactions
- A block is composed of multiple transactions
- Concur with other nodes
- Which of these can be trusted?
- Also provides some ordering.
- Consensus algorithm:
  - Confirms the correctness of transactions in a block, according to the consensus algorithms deployed and the policies applied.
  - Once the block is confirmed, then it enters the blockchain, so consensus algorithm has to agree on order the blocks are added
  - Interact and complete smart contract layer

Do I need a blockchain?

adapted from EdX.org

- There is a need for a shared common database
- The parties involved with the process have conflicting incentives, or do not have trust among participants
- There are multiple parties involved or writers to a database
- There are currently trusted third parties involved in the process that facilitate interactions between multiple parties who must trust the third party. This could include escrow services, data feed providers, licensing authorities, or a notary public
- Cryptography is currently being used or should be used. Cryptography facilitates data confidentiality, data integrity, authentication, and non-repudiation
- Data for a business process is being entered into many different databases along the lifecycle of the process. It is important that this data is consistent across all entities, and/or digitization of such a process is desired
- There are uniform rules governing participants in the system

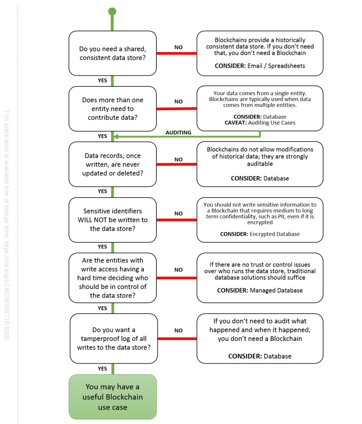
When not to use Blockchain

adapted from EdX.org

- The process involves confidential data
- The process stores a lot of static data, or the data is quite large
- Rules of transactions change frequently
- The use of external services to gather/store data.

Flowchart [**yaga2018blockchain**]

NISTIR 8202 BLOCKCHAIN TECHNOLOGY OVERVIEW


Do you need a shared, consistent data store?
NO
Blockchains provide a historically consistent data store. If you don't need that, you don't need a Blockchain
CONSIDER: Email / Spreadsheets
YES
Does more than one entity need to contribute data?
NO
Your data comes from a single entity. Blockchains are typically used when data comes from multiple entities.
CONSIDER: Database
CAVEAT: Auditing Use Cases
AUDITING
YES
Data records, once written, are never updated or deleted?
NO
Blockchains do not allow modifications of historical data; they are strongly auditable
CONSIDER: Database
YES
Sensitive identifiers WILL NOT be written to the data store?
NO
You should not write sensitive information to a Blockchain that requires medium to long term confidentiality, such as PII, even if it is encrypted
CONSIDER: Encrypted Database
YES
Are the entities with write access having a hard time deciding who should be in control of the data store?
NO
If there are no trust or control issues over who runs the data store, traditional database solutions should suffice
CONSIDER: Managed Database
YES
Do you want a tamperproof log of all writes to the data store?
NO
If you don't need to audit what happened and when it happened, you don't need a Blockchain
CONSIDER: Database
YES
You may have a useful Blockchain use case

This publication is available free of charge from: https://doi.org/10.6028/NIST.IR.8202

Figure 6 - DHS Science & Technology Directorate Flowchart

Participants

Individual

- User
- end-users
- more individuals than businesses

Organisational

- service providers
- manufacturers
- an organisation can exist independent of the individuals
- Expansion during growth
- Reduction during downturn
- Individuals are members of an organisation and act on behalf of the organisation, and with authority of that organisation.

System or Device

- special kind of individual
- How many systems does Middlesex have?
- Systems belong to organisation
- Systems act on behalf of organisation
- Autonomous devices
- Possibility of more autonomous agents in the future

Assets

- exchanges of information between participants
- participants have a degree of autonomy
- assets are quite passive
- assets have meaning to participants
- assets have value
- therefore represent the value exchanged between participants
- Tangible: cars, products, goods
- Intangible: staff morale, IP, policies, data, knowledge, information
- Literally? Everything on blockchain is intangible, its digital.
- Intangible assets on blockchain, do have value, it is just not explicit

Asset Structure

- Attributes: Properties and Relationships
- Properties: characteristics of an object, manufacturer, model, registration, car.
- Asset Type
- Asset Instance
- Relationships: reference to another asset
- car: has insurance, owner, mot, tax
- Modelling the Assets, participants and relationships
- General rule of thumb: reduce big assets into smaller assets with relationships
- Separation of concerns
- domain-specific - colour is an asset of a car?

Ownership & Lifecycles

- associative relationship
- does a person own a car?
- is car an attribute of person?
- It is a mapping between participant (individual) and the car
- ownership is a concept
- blockchains are often used to record ownership and changes in ownership
- Provenance
- properties of asset modified
- asset ceases to exist
- Bank loan
- Transformation involves: division and aggregation
- Transformation type: homogenous and heterogenous

Asset Life Cycles

Division & Homogeneous

- Reduction of large asset
- Leather
- Divide leather up for manufacturers
- Leather comes from the same animal
- Just divided up into smaller pieces
- It is the same and undergone a homogeneous transformation

Aggregate & Heterogeneous

- transform the leather into a shoe
- leather has been combined with other material
- leather has been aggregated to form a shoe
- components have undergone a heterogeneous transformation
- Tangible
- Intangible?

Participants, Assets and Transactions

- Assets evolve via transactions
- Insurance policy
- Participants evolve via transactions
- Difference: form v. function
- Both are resources
- Related in the most general sense
- Because they both have lifecycles described by transactions does not make them equal
- Participants: Users
- Assets: tangible or intangible

Exchange

- Record change
- Change is captured via a transaction
- buyer pays owner in exchange of possesion of asset
- Lucy pays £289 for a printer on 28th March 2019
- Generalise and particular (instance)
- Generalisation describes semantics of transaction
- Particular transactions describes an instance
- Lucy received a receipt for her transaction
- The receipt is a copy of the transaction
- The computer shop also keeps a record of the receipt
- It the printer breaks after 2 days, we get to find out the true nature of the transaction
- otherwise the transaction is implicit

Low-consequence transactions

- implicitness has downsides
- trust
- laws on fair transactions
- Sale of Goods Act
- Lack of explicit contract, simplifies the transaction
- Receipt
- reducing friction

Case Study

Letter of Credit (LC)

- How was trade possible?
- Unsafe to travel with valuables?
- Go back 500 years
- Bank writes L/C.
- L/C is used to exchange goods.

Example

- Company Argo banks with Bank of Portugal
- Company Argo is based in Lisbon
- Company Baa is based in London
- Company Baa banks with Bank of England
- Company Baa sells Wool
- Company Argo wants to buy some Wool from Company Baa

Case Study - Scenario

- Argo gets L/C from Bank of Portugal
- Argo representative travels to London
- Agrees on price and quantity
- Pays for goods with L/C
- L/C is honoured by Bank of England
- Bank of England pay Baa
- Bank of Portugal pay Bank of England

What can go wrong?

- Trust?
- It takes 1 months to transport Goods from London to Lisbon
- Goods could arrive in bad condition
- Goods may not arrive
- Goods could be destroyed in transit
- Value of goods could depreciate during journey
- Export license is denied

London – Lisbon

- Line of credit needs some further details?

Letter of Credit (L/C)

Bank of Portugal hereby issues the amount of 400 Escudos payable immediately. In accordance with L/C 48722.

London –Lisbon

- Line of credit needs some further details?
- What is for?
- Invoice?
- Proof of delivery
- Insurance of goods during transit
- Pay half now, and half on delivery
- Charges for L/C

Letter of Credit (L/C)

Bank of Portugal hereby issues the amount of 400 Escudos payable immediately. In accordance with L/C 48722. This L/C should be accompanied with:

1. Bill of Lading (B/L)
2. Packing List
3. Invoice

London–Lisbon

- Bill of Lading
- Ensure goods are present and correct
- Ensure goods are in order
- Weight, Condition and Quality.

Bill of Lading (B/L)

- Shipper: Baa
- Consignee: Argo
- 1 Ton of Wool
- London to Lisbon
- Freight Charges: 50 Escudos

London–Lisbon

- Export License
- Allowing the export of goods
- Import License
- Allowing the import of goods

Export

On 20th March 1784 UK Govt permit the export of 1 ton: Wool. Consignment: 32496. Company: Baa.

Import

On 20th March 1784 Potuguese Govt permit the import of 1 ton: Wool. Tax Levy: 10 Escudos. Consignment: 10473. Company: Argo.

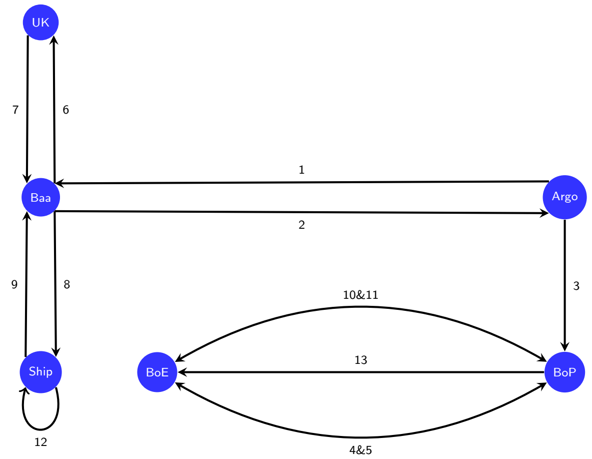
London–Lisbon

- Bank of Portugal
- Bank of England
- Meeting, whereby Portugal would honour their L/C and pay Bank of England.
- Trust
- Bank of England trusts Bank of Portugal
- Charges made to Argo

Case Study

1. Argo requests goods from Baa in exchange for money
2. Baa accept the trade deal
3. Argo asks BoP for a L/C in favor of the Baa
4. BoP supplies a L/C in favour of Baa, and payable to BoE
5. BoE accepts the L/C on behalf of Baa
6. Baa applies for an E/L
7. E/L is granted
8. Baa prepares shipment
9. Logistics company validate E/L and supplies a B/L to Baa
10. BoE claims half the payment from the BoP
11. BoP transfers half the payment to BoE
12. Logistics ships goods to location
13. BoP pays remain amount to BoE

Case Study



Case Study - Assets & Paricipants

Data

- transactions
- Letter of Credit
- Bill of Lading
- Export License
- trade agreement
- Payment
- Shipment

Participants

- Baa
- Argos
- UK govt
- BoE
- BoP

Case Study - Access

- Only Argos may apply for L/C
- Only BoP may supply L/C
- Only BoE may accept L/C

Case Study - Access Generalisation

- Only an importer may apply for L/C;
- Only an importer's bank may supply an L/C;
- Only an exporter's bank may accept an L/C;
- Only an exporter may requests an E/L
- Only a regulatory authority may supply an E/L;
- Only an exporter may prepare shipment;
- Only a logistics company may supply a B/L;
- Only a carrier may update a shipment location;
- Only an importer's bank may send money; and,
- only an exporter's bank may receive money

Case Study

Friction

- line of credit
- export licenses
- agreements
- bill of lading
- overheads
- Speed, but friction remains

Frictionless?

- payment linked to documentary completion
- payment linked to progress of shipment?
- trade agreement on a single blockchain
- implement a smart contract

Case Study - Benefits

- through [dis]trust and [dis]honesty came the inspiration for L/C and B/L;
- applying for E/L, L/C and B/L is an overhead and increases turn-around-times;
- Automation has reduced this, but not changed it much.
- What are the benefits of blockchain?

Case Study - Benefits

- conditional installments
- (dis)intermediaries
- trust
- auditable
- secure
- sustainable
- extensible
- communication
- increase accountability, minimise risk

CTO

Enumerator Types

- String: UTF8
- Integer: 32bit signed number
- Double: double precision 64bit number
- Long: 64bit signed number
- DateTime: ISO8061 DateTime
- Boolean: true, false

Listing

```
enum enumeratorName {
  o ENUM1
  o ENUM2
}

```

```
enum gender{
  o MALE
  o FEMALE
  o NONBINARY
}

```

CTO

Concepts

- Abstract classes
- Participants
- Assets
- Transaction
- No instances
- Extended

Listing

```
abstract concept Address {
  o String houseNumber
  o String streetName
  o String townName
  o String county
  o String country default="UK"
  o String postCode regex=/RegEx/
}

```

CTO

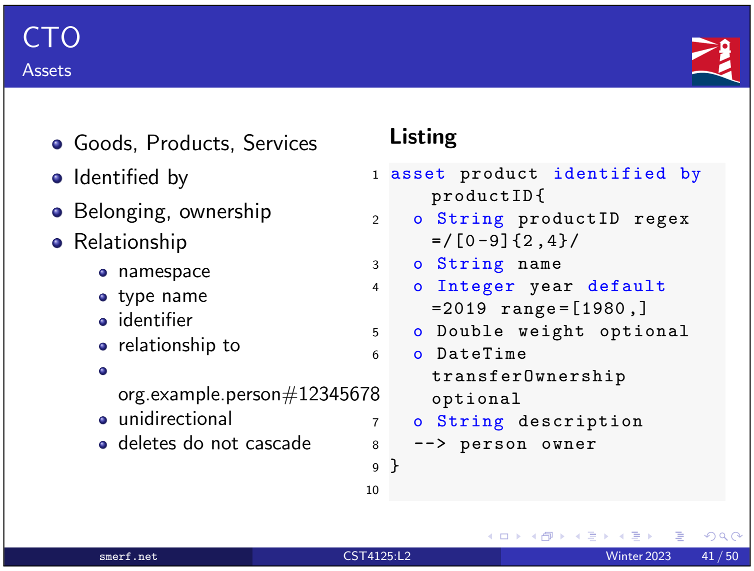
Participants

- Users
- identified by
- extend
- Group of users

Listing

```
participant person identified by personID{
  o String personID regex=/[0-9]{8}/
  o String lastName
  o String firstName
  o gender Gender
  o Address address
}

```

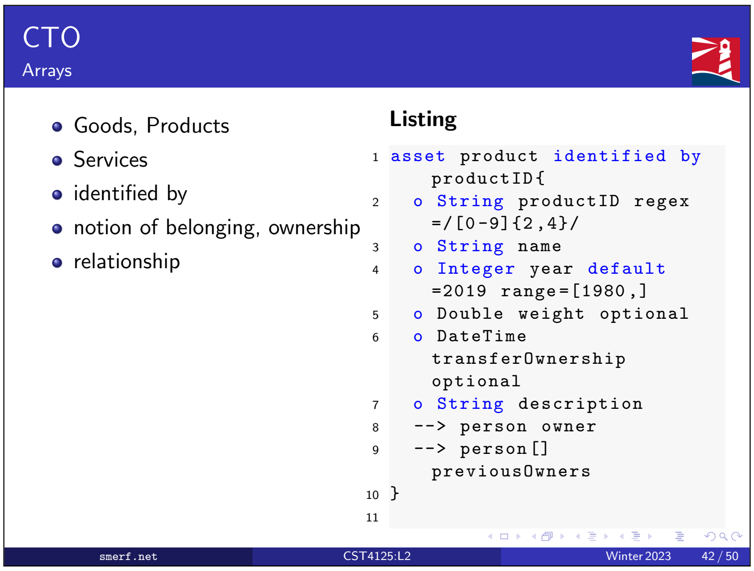


CTO
Assets

- Goods, Products, Services
- Identified by
- Belonging, ownership
- Relationship
  - namespace
  - type name
  - identifier
  - relationship to
  - org.example.person#12345678
  - unidirectional
  - deletes do not cascade

Listing

```
asset product identified by productID{
  o String productID regex =/[0-9]{2,4}/
  o String name
  o Integer year default =2019 range=[1980,]
  o Double weight optional
  o DateTime transferOwnership optional
  o String description
  --> person owner
}
```

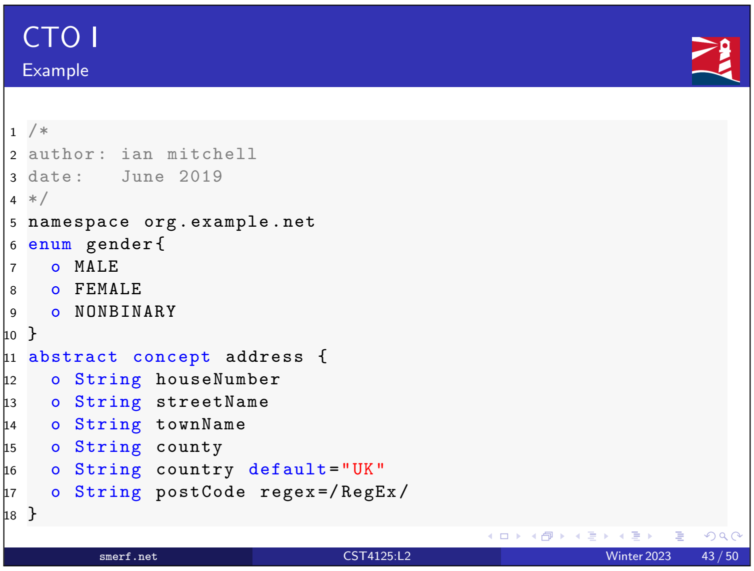


CTO
Arrays

- Goods, Products
- Services
- identified by
- notion of belonging, ownership
- relationship

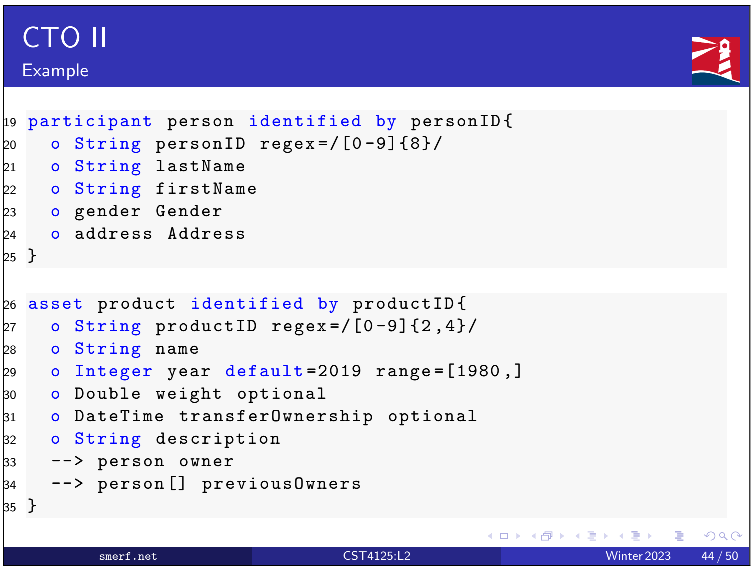
Listing

```
asset product identified by productID{
  o String productID regex =/[0-9]{2,4}/
  o String name
  o Integer year default =2019 range=[1980,]
  o Double weight optional
  o DateTime transferOwnership optional
  o String description
  --> person owner
  --> person[] previousOwners
}
```



CTO I
Example

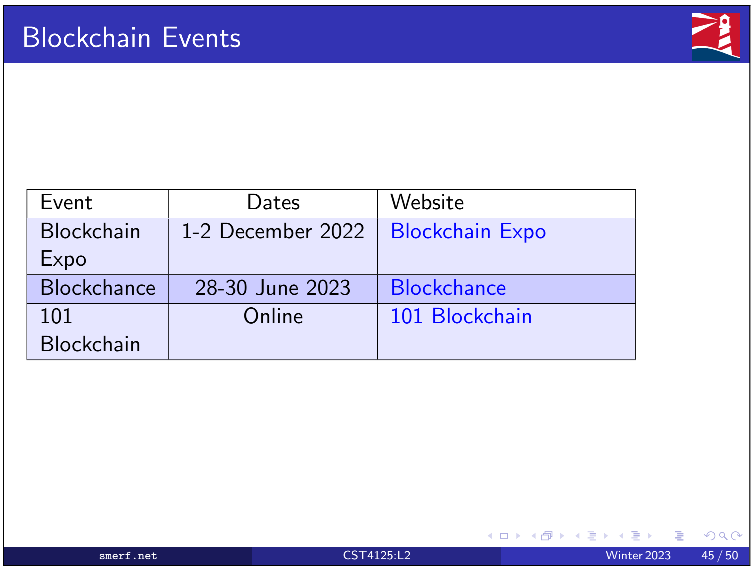
```
/*
author: ian mitchell
date:   June 2019
*/
namespace org.example.net
enum gender{
  o MALE
  o FEMALE
  o NONBINARY
}
abstract concept address {
  o String houseNumber
  o String streetName
  o String townName
  o String county
  o String country default="UK"
  o String postCode regex=/RegEx/
}
```



CTO II
Example

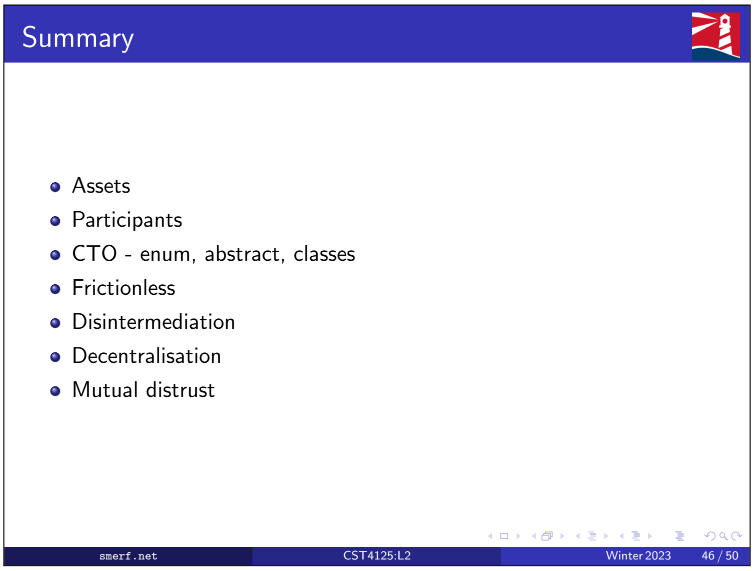
```
participant person identified by personID{
  o String personID regex=/[0-9]{8}/
  o String lastName
  o String firstName
  o gender Gender
  o address Address
}

asset product identified by productID{
  o String productID regex=/[0-9]{2,4}/
  o String name
  o Integer year default=2019 range=[1980,]
  o Double weight optional
  o DateTime transferOwnership optional
  o String description
  --> person owner
  --> person[] previousOwners
}
```



Blockchain Events

Event	Dates	Website
Blockchain Expo	1-2 December 2022	Blockchain Expo
Blockchance	28-30 June 2023	Blockchance
101 Blockchain	Online	101 Blockchain



Summary

- Assets
- Participants
- CTO - enum, abstract, classes
- Frictionless
- Disintermediation
- Decentralisation
- Mutual distrust

Reading

- Read []
- Define a problem specification for coursework
- Read/practice CTO
- Remind yourself of Regex

References I

Web Resources

- `http://hyperledger.org`

Acronyms I