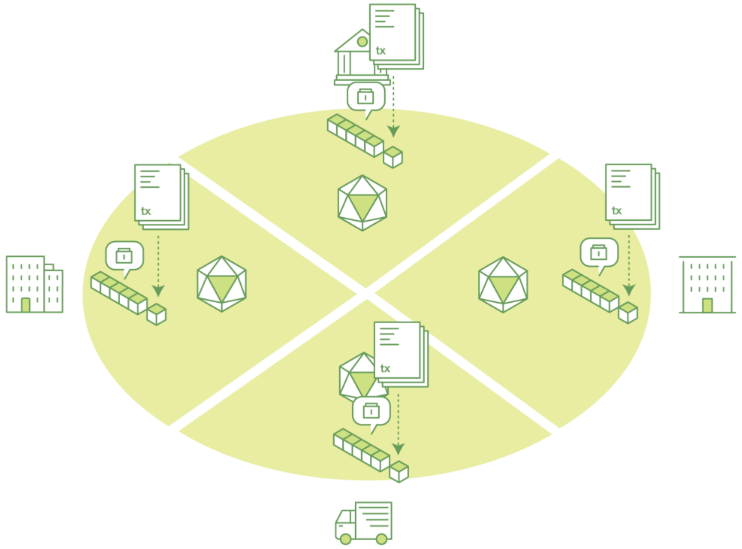
freeknowledgemission-list@meetup.com



Block-chain

See http://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html

freeknowledgemission-list@meetup.com

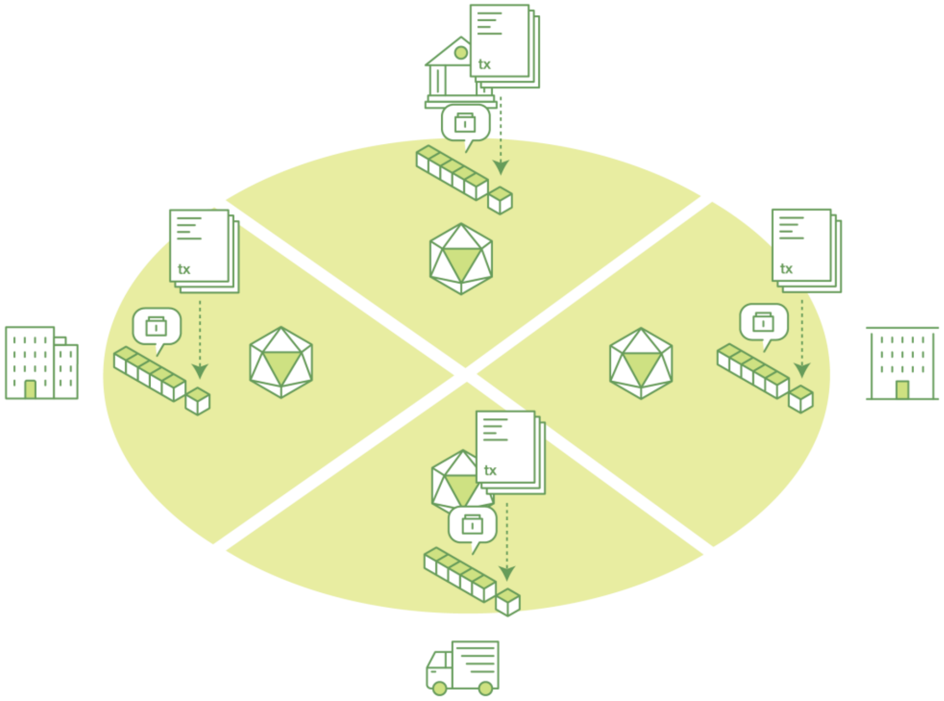
A blockchain network is a distributed ledger that records all the transactions that take place of the network.

For this discussion we will leave Bitcoins and focus on an open source implementation called Fabric. Source code is found in Github under hyperledger/Fabric which was created recently in a hackathon.

HYPERLEDGER

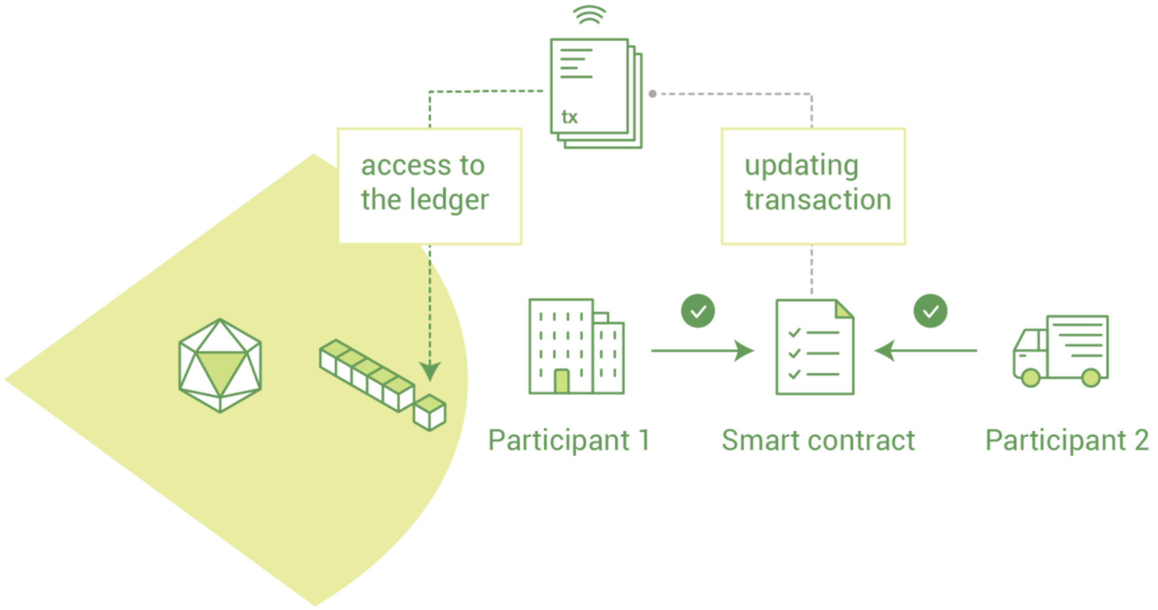
freeknowledgemission-list@meetup.com

A blockchain ledger is often described as **decentralized** because it is replicated across many network participants, each of whom **collaborate** in its maintenance.



freeknowledgemission-list@meetup.com

To support the consistent update of information – and to enable a whole host of ledger functions (transacting, querying, etc) – a blockchain network uses **smart contracts** to provide controlled access to the ledger



freeknowledgemission-list@meetup.com

Contracts are actually “chaincode” and written in Go and compiled into a blockchain.

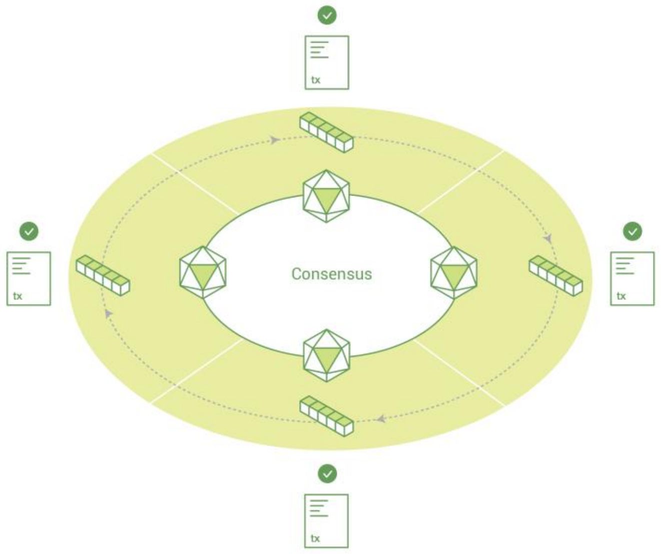
This code is the read and write of information into a blockchain:

```
// Invoke is called per transaction on the chaincode. Each transaction is
// either a 'get' or a 'set' on the asset created by Init function. The Set
// method may create a new asset by specifying a new key-value pair.
func (t *SimpleAsset) Invoke(stub shim.ChaincodeStubInterface) peer.Response {
    // Extract the function and args from the transaction proposal
    fn, args := stub.GetFunctionAndParameters()

    var result string
    var err error
    if fn == "set" {
            result, err = set(stub, args)
    } else {
            result, err = get(stub, args)
    }
    if err != nil {
            return shim.Error(err.Error())
    }

    // Return the result as success payload
    return shim.Success([]byte(result))
}
```

freeknowledgemission-list@meetup.com

The process of keeping the ledger transactions synchronized across the network – to ensure that ledgers only update when transactions are approved by the appropriate participants, and that when ledgers do update, they update with the same transactions in the same order – is called **consensus**.

Note: There is no mining and no POW



freeknowledgemission-list@meetup.com

Fabric does not use POW but instead a **Membership Service Provider (MSP)** which is a version of central authority. Without the POW the Byzantine Fault Tolerance issue surfaces. This is a roadmap item for Fabric.

See https://en.wikipedia.org/wiki/Byzantine_fault_tolerance

Fabric allows private, that is not shared, transactions between peers. This allows using the ledger for biz transactions that should not be shared. For example purchase agreements.

freeknowledgemission-list@meetup.com

Switching to Bitcoin

Source is open.

See https://github.com/bitcoin/bitcoin

The original paper: https://bitcoincore.org/bitcoin.pdf

freeknowledgemission-list@meetup.com

Questions?