**Lecture 5-2 notes**

**CREATING AND DEPLOYING SMART CONTRACTS**

**2.1 GOALS**

* Removal of manual intervention and oversight

eg. from legal counsel

* Reduction in associated legal costs, fees, and process (time) speed of contract creation and execution
* Automated transfer of funds via computer recognizable/definable events
* Flexibility in contract

**2.2 SMART CONTRACTS TYPES**

* Smart contract can be divided into deterministic and non-deterministic smart contracts.

**Deterministic smart contracts** are smart contract codes that do not depend on outside information other than information on the Blockchain in which they live into be triggered and work effectively.

* In other words, the Blockchain network facilitating the smart contract has sufficient information to make decisions.
* E.g., peer-to-peer lottery: the funds are held on the Blockchain network and random numbers are also generated by the smart contract code. At the end of the lottery, the funds are transferred the winners account via his or her address on the Blockchain network

**Non-Deterministic smart contracts**: network facilitating the smart contract code does not have sufficient information to make decisions.

* Thus, an outside party is needed, usually called an ‘Oracle’ in the computer science domain.
* Decisions about value flow based on human behavior, events (price drop or hike) or predictions.
* However, research has shown that using external state does not always introduce the need for trusting an additional party.
* For example, in a driving license renewal scenario, the government is a trusted party anyway. thus, we use government as a validation oracle that injects external state in to the blockchain.

**2.3 CREATE AND DEPLOY THE SMART CONTRACT**

STEPS:

* Understand the use-case of smart contract.
* create a basic architecture of smart contracts interaction or flowchart how functions will interact with each other.
* Start development using any IDE or development tools like Truffle, remix with proper documentation of each and every function.
* Once the development is completed start testing smart contracts on test-net or private blockchain. (this is called a manual testing).
* Record all the transaction while testing on test-net, analyze results of all transactions with actual use case or business logic of smart contract.
* Unit testing will be the next step in smart contract development life cycle, there are multiple frameworks for unit and integration testing that can be use to test smart contract. Example : Truffle framework.
* Once unit testing is done using truffle framework on ganache, smart contract author should go for 3rd party Audit of smart contract.
* Last but not the least, bug bounty programs are also very efficient to secure smart contracts.

**2.4 BLOCKCHAIN NETWORKS USING SMART CONTRACTS**

* Much has been said about smart contracts in relation to blockchain technology. While there are plenty of examples of smart contracts implemented within different blockchain networks and projects, the most notable remain Bitcoin and Ethereum.

**Bitcoin**

* Though Bitcoin is mostly known for transactions of the Bitcoin cryptocurrency, its protocol can also be used to create smart contracts.
* Bitcoin provides a programming language that allows for custom smart contracts like multisignature accounts, payment channels, escrows, and time locks.
* There’s a separate smart contract platform called RootStock built on Bitcoin’s blockchain.

**Ethereum**

* Ethereum is the most prominent smart contract framework, created and designed specially to support smart contracts.
* This framework, programmed in the Solidity language, is a decentralized platform that runs smart contracts without any possibility of downtime, censorship, fraud, or third-party interference.
* The Ethereum blockchain database stores transactions between people, transactions involving smart contracts, and their source code.

**2.5 ADVANTAGEOUS OF SMART CONTRACT**

Explicit programming algorithms in core and blockchain properties like decentralization, transparency, fraud resistance, and others make smart contracts a credible alternative for establishing business relations and performing transactions.

As an alternative to traditional contracts with a central business model, here are the benefits that smart contracts offer businesses:

* **Direct dealings with customers**. Smart contracts remove the need for intermediaries and allow for transparent, direct relationships with customers.
* **Resistance to failure**. Since businesses aren’t dependent on a third party, no single person or entity is in control of data or money. Decentralization means that even if any individual leaves the blockchain network, the network will continue to function with no loss of data or integrity.
* **More trust**. Business agreements are automatically executed and enforced. Plus, these agreements are immutable and therefore unbreakable.
* **Fraud reduction**. Since smart contracts are stored in a distributed blockchain network, their outcome is validated by everyone in that network. Therefore, no one can force control to release other people’s funds or data, as all other blockchain participants would spot this and mark such an attempt as invalid.
* **Cost efficiency**. Eliminating intermediaries removes additional fees, allowing businesses and their customers not only to interact and transact directly but also to do so with low to no fees for transactions.
* **Record keeping**. All contract transactions are stored in chronological order in the blockchain and can be accessed along with the complete audit trail.

**2.6 ETHEREUM BLOCKCHAIN COMPONENTS**

The important components of Ethereum blockchain are

Node/Client: A node is a device/program that communicates with the Ethereum network. Nodes are also known as clients. Software that can act as an Ethereum node include Parity, Go-Ethereum, etc.

Block: A block is a package of data that has zero or more transactions, the hash of the previous block, and optionally other data.

Miners: Miners will add the block to the Blockchain. Miners are simply nodes in the ethereum network who find new block, confirm transactions and commit new transactions in a block

Proof of work: Proof of work is an activity that miners undertake to write transactions to a new block. It refers to a mathematical value that can act as the proof of having solved a resource and time-consuming computational problem such as finding nonce in Ethereum (Ethash POW Algorithm by Ethereum).

Ethereum Virtual Machine: Ethereum Virtual Machine is the decentralized computing platform which forms the core of the Ethereum platform.

Smart Contract: A persistent piece of code on the Ethereum Blockchain that has a set of data and executable functions. These functions execute when Ethereum transactions are made to them with certain input parameters. Based on the input parameters, the functions will execute and interact with data within and outside of the contract.

Ether: Ether is the name of the currency used within Ethereum Network. Miners are rewarded with ethers for providing computing power for finding a new block and confirming new transaction of Smart contracts. Ether is used to pay for computations within the EVM.

Gas: Refers to the pricing value required to successfully perform a transaction or execute a smart contract on the Ethereum Blockchain platform. It is also the name for crypto-fuel that is consumed when code is executed by the EVM. The gas is paid as execution fee for every operation made on an Ethereum Blockchain.

Gas Limit: The gas limit represents the maximum amount of gas you are willing to pay for a smart contract transaction execution.

Mining Pool: Mining pool are simply groups of miners that work together to mine blocks for ethereum network.

Main-net: A main-net is a main Ethereum Blockchain network.

**Example:**

**Ethereum Client: geth**

**Solidity compiler: Solc**

# **Whisper client**

* Whisper is a decentralized communication protocol developed by Ethereum developers.
* Whisper lets nodes in the network communicate with each other.
* It supports broadcasting, user-to-user, encrypted messages, and so on.
* It's not designed to transfer bulk data.

**Swarm client & Local store**

* Swarm is a decentralized storage platform respectively developed by Ethereum developers.
* Swarm is like Filecoin, that is, it differs mostly in terms of technicalities and incentives.
* Filecoin does not penalize

**Dapps:**

* Backend (JavaScript)
* Front end (HTML-CSS)
* Smart Contracts

**2.7 SMART CONTRACTS CAN AUTOMATIZE EXECUTION OF TRANSACTIONS**

* Smart contracts are useful in many occasions to replace human intervention (ex: in case of an a estate agency, to send automatically the keys once the contract has been signed, and rental has been paid).
* Risk: the code remains vulnerable and can be corrupted.

Common Use Cases for Smart Contract Automation

* DeFi protocols already rely on off-chain smart contract automation. MakerDAO relies on third party Keepers to monitor the collateralization ratios of debt positions and liquidate any undercollateralized position. Other DeFi protocols have similar needs.
* There are two, often overlapping use cases around off-chain smart contract automation:
* **Automated Triggers:** You want to execute a contract under a certain condition.
* State and Event Monitoring: You want to know when a contract is in a certain condition.

**Use Case 1: Automated Triggers**

* You often want to execute a contract periodically or under certain conditions. For example:
* Rebalancing a pool periodically
* Closing out voting rounds in a DAO / governance process
* Poking oracles to refresh data
* Paying out pro-rated dividends for security tokens

**Use Case 2: State and Event Monitoring**

* Monitoring can let you know when certain conditions are met. For example:
* You want to know if a value in a smart contract has changed
* You want to be notified of all Access Control changes
* You want to know when a specific smart contract Event was emitted

**2.8 EXAMPLE FORMING SMART CONTRACT**

* Blockchain Climate Risk Crop Insurance is a digital platform wherein crop insurance policies are plugged into smart contracts on a blockchain and indexed to local weather.
* During an extreme weather event, the policies are automatically triggered, which facilitates fair, transparent and timely payouts.
* For farmers, the instrument reduces transaction costs during the processing of claims.
* The Lab estimates that in the long-term, an integrated insurance platform model can reduce the costs required to issue a policy by up to 41%, enabling a premium reduction of up to 30%.
* Reduced claim cycles, from 3 months to 1 week, and increased transparency also build trust amongst stakeholders.

**2.9 SMART CONTRACT USE CASES**

Smart contracts are gaining popularity and have already been implemented in various blockchain projects. Here are just several promising examples of smart contract implementations in different industries.

**Banking**

Banking might be the primary industry where smart contracts appear to be the most significant alternative to the traditional model of transactions. Smart contracts make payments as well as loans, and nearly all others financial operations literally automated.

[KYC-Chain](https://kyc-chain.com/) implements smart contracts for individuals, businesses, and financial institutions. In the core of KYC-Chain are mechanisms allowing clients to comply with regulatory norms, such as automatic smart checks, as well as to share pertinent documents and get them digitally attested by notaries and institutions.

**Healthcare**

Smart contracts can also improve healthcare. They can streamline processes for insurance trials, increase access to cross-institutional data, and boost confidence in patient privacy. Authentication, authorization, and identity confirmation remain open issues for smart contracts executed on blockchain networks.

An example of smart contracts in the healthcare industry is [Dentacoin](https://dentacoin.com/" \t "_blank). Dentacoin aims to bring patients and dentists together in communities to improve dental care and make it affordable worldwide.

**Supply Chain**

Another area where smart contracts can provide real-time visibility is supply chains. Smart contracts ensure granular inventory tracking, benefitting supply chain financing as well as reducing the risk of theft and fraud.

Smart contracts can effectively be used regardless of marketplace type or goods sold. For example, [Name Bazaar](https://namebazaar.io/) is implementing smart contract technology within a peer-to-peer marketplace where users can exchange cryptographic assets on the blockchain in the form of domains.

**Legal Issues**

The traditional model of resolving legal issues and certifying documents is also giving way to smart contracts. Smart contracts eliminate the need for notarization, offering not only an automated and unbiased but also a cost-efficient solution.

Nottar.io illustrates the concept of notarizing documents using the Ethereum blockchain.

**Real Estate**

You’ve already read about the easiest concept of how a smart contract can work for real estate in the How Smart Contracts Work section. Of course, real-life projects are way more complicated and comprehensive and need to cover a broader range of issues and opportunities.

For instance, [FOAM](https://www.foam.space/) is a stock market for real estate that uses technology to let users make property transactions, get financing and funding, and manage leases.

**Government**

It might be most interesting to investigate the potential impact of decentralization technologies on monopolistic spheres, especially government systems.

Australian company Horizon State is working to provide voting and other tools to help the democratic process. Particularly, Horizon State intends to offer a blockchain-based voting ecosystem that supports secure, cost-effective campaign operations for a range of election types and voting parameters. The primary goal of Horizon State is to enable transparent, unbiased voting in countries around the world.

**Internet of Things Networks**

There are areas where smart contracts intersect with other technologies, and the Internet of Things (IoT) is one of them. A combination of smart contracts and IoT is powerful and can enable significant transformations across industries, paving the way for new distributed applications.

Project [Oaken](https://www.oakeninnovations.com/) is proving that. Oaken provides autonomous IoT hardware and software coupled with blockchain technology. Altogether, these components make it possible to use Oaknen with nearly any device to build an IoT network and therefore apply it to various real-life needs.