**Lecture 4-1 Notes**

**Smart Contract**

* In the traditional centralized model of business relationships, there’s always a third party that stands between the two parties that are making a transaction and affirming the terms and conditions in a contract.
* This third party may be a banking institution, a law enforcement company, a government establishment, or some other intermediary.
* When building relationships within a centralized model, businesses are dependent on intermediaries, which puts customers at risk.
* Besides, central systems cannot guarantee payments and implementation of contracts.
* The development of blockchain technology, which allows businesses to build decentralized models, opens new horizons for businesses to conduct transactions and make agreements.
* One of the technologies that’s proposing an alternative to the traditional model is the smart contract.
* Although smart contract hype has grown with the hype around blockchain technology, the term smart contract appeared over twenty years ago.
* Nick Szabo, a computer scientist and cryptographer, wrote an article about smart contracts all the way back in 1995.
* The concept that Szabo offered precisely corresponds to what smart contracts offer today, including the idea of implementing and storing smart contracts within a distributed ledger.

**What is a Smart Contract?**

* A smart contract is similar to a contract in the physical world, but it’s digital and is represented by a tiny computer program stored inside a blockchain.
* More specifically, a smart contract is a piece of software that stores rules for negotiating the terms of an agreement, automatically verifies fulfillment, and then executes the agreed terms.

**What is the main idea of a smart contract?**

* Since a smart contract removes reliance on a third party when establishing business relations, the parties making an agreement can transact directly with each other.
* Consider the example of a crowdfunding platform where product teams share their projects and collect money from supporters until a goal is reached.
* If such a platform is centralized – like Kickstarter, for instance – then it acts as a third party between product teams and supporters who donate their money.
* This means both sides need to trust Kickstarter and, in fact, pay an additional fee to Kickstarter to serve as an intermediary.
* A smart contract, alternatively, can perform the same crowdfunding actions – sharing projects, setting goals, and collecting donations – but without a third party.
* Simply put, we can program a smart contract to execute all of these actions.
* More precisely, we can program a smart contract to receive funds until a goal is reached.
* If the project gets fully funded before the deadline, the money raised automatically goes to the product team. If the project fails, the money automatically goes back to supporters.
* Since a smart contract is stored inside a blockchain where all data is stored in a distributed manner, no one is in control of the money.
* In a decentralized business model, smart contracts replace any other trusted third party. This leads to a pretty valid question:

**Why Trust a Smart Contract?**

Smart contracts are designed and implemented within blockchains, and therefore they inherit some of the blockchain’s properties:

* **They are immutable**, which means a smart contract can never be changed and no one can tamper with or break a contract.
* **They are distributed**, which means that the outcome of the contract is validated by everyone in the network, just like any transaction on a blockchain. Distribution makes it impossible for an attacker to force control to release funds, as all other participants would detect such an attempt and mark it as invalid.

**How a Smart Contract Works**

* A smart contract is a program, or more simply put, code.
* The code behind a smart contract contains specific terms that are executed when triggered by specific agreed events.

**Example: how a smart contract might work?**

if Alice is renting an apartment in Los Angeles and Bob from New York is looking to rent an apartment for his journey there.

1. **A Logically Behaved Algorithm**

* Typically, Alice and Bob would use some platform that unites hosts and guests to agree on renting.
* This kind of platform would serve as a third party and would certainly take responsibility for compliance with the agreed terms.
* However, both Alice and Bob would be charged a fee by the platform.
* Besides, if either of them failed to fulfill their commitment, dispute resolution may be time-consuming and require a detailed review.

If Alice and Bob instead make an agreement using a smart contract, the smart contract will behave logically based on its algorithm and will guarantee that all the agreed terms and conditions are fulfilled. Immutability, which is in the DNA of a smart contract, will not let Alice or Bob cheat.

Thus, the following terms and events can be set out in a smart contract between Alice and Bob:

* Independent storage is created, where both Alice and Bob can put value but cannot easily take out.
* Bob puts money for rent in storage.
* Alice puts the address and the code to her apartment in storage.
* Alice gets payment confirmation and Bob receives the address and apartment code.
* If Bob comes to LA and the address and code provided by Alice are right, Alice gets the payment.
* If it appears that the address or code supplied by Alice are wrong, Bob gets his money back.
* If Bob does not come to LA, Alice gets her liquidated damages payment and Bob gets the rest of what he paid.
* At the end of the agreement, the smart contract is considered fulfilled and remains stored in the blockchain network.

This set of conditions and events represents the most basic one-time smart contract. Setting up terms in the code of a smart contract ensures satisfactory fulfillment. Overall contract fulfillment is guaranteed by the blockchain technology itself, as a complete copy of the blockchain is publicly stored by all network participants and the smart contract remains immutable.

1. **The Logic of the One-Size-Fits-All Smart Contract**

* Once Alice generates a smart contract that automatically and transparently works for her when renting out her apartment to Bob, she might think about creating a universal agreement for all further renters so she does not need to create a new smart contract for each new guest.
* With this universal agreement, anyone on the blockchain network can rent Alice’s apartment by following the algorithm above: the potential guest transfers rent payment, gets the address and apartment code, and then Alice gets her payment if everything works according to the contract terms for both sides.
* Moreover, smart contracts can be even more universal. We could program a smart contract to be used not only by Alice but by any person who wants to rent out his or her flat.
* Smart contracts can contain more specific conditions, such as automatically adjusted prices, discounts, partial payments, and nearly any other imaginable option.