CST 428 BLOCK CHAIN TECHNOLOGIES

S8 CSE –
ELECTIVE
MODULE – 4

Smart Contracts and Use Cases

Smart Contracts - Definition

Smart contracts are described by Szabo as follows:

"A smart contract is an electronic transaction protocol that executes the terms of a contract. The general objectives are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss,arbitrations and enforcement costs, and other transaction costs."

Comprehensive generalized definition of a smart contract:

A smart contract is a secure and unstoppable computer program representing an agreement that is automatically executable and enforceable.

- A smart contract is, fundamentally, **a computer program that is written in a language that a computer or target machine can understand.**
- It **encompasses agreements between parties** in the form of business logic.
- Smart contracts are automatically executed according to the instruction that is coded in, for example, when certain conditions satisfy.
- They are enforceable, which means **that all contractual terms perform as specified and expected, even in the presence of adversaries.**
- Enforcement is a broader term that encompasses traditional enforcement in the form of a law, along with the implementation of specific measures and controls that make it possible to execute contract terms without requiring any intervention.
- They should work on the principle that *code is the law*, which means that there is no need for an arbitrator or a third party to enforce, control, or influence the execution of a smart contract.
- Smart contracts are self-enforcing as opposed to legally enforceable.

- They are secure and unstoppable, which means that these **computer programs are fault-tolerant and executable in a reasonable (finite) amount of time.**
- These programs should be able to execute and maintain a **healthy internal state, even if external factors are unfavorable**.
- For example, imagine a typical computer program that is encoded with some logic and executes according to the instruction coded within it.
- However, if the environment it is running in or the external factors it relies on deviate from the usual or expected state, the program may react arbitrarily or abort. Smart contracts must be immune to this type of issue.

a smart contract has the following properties:

- **Automatically executable**: It is self-executable on a blockchain without requiring any intervention.
- **Enforceable**: This means that all contract conditions are enforced automatically.
- **Secure**: This means that smart contracts are tamper-proof (or tamper-resistant) and run with security guarantees. The underlying blockchain usually provides these security guarantees; however, the smart contract programing language and the smart contract code themselves must be correct, valid, and verified.
- **Deterministic**: The deterministic feature ensures that smart contracts always produce the same output for a specific input. Even though it can be considered to be part of the secure property, defining it here separately ensures that the deterministic property is considered one of the important properties.
- **Semantically sound**: This means that they are complete and meaningful to both people and computers.
- **Unstoppable**: This means that adversaries or unfavorable conditions cannot negatively affect the execution of a smart contract. When the smart contracts execute, they complete their performance deterministically in a finite amount of time.

Smart contract templates

- Smart contracts can be implemented in any industry where they are required, but the most popular use cases relate to the financial sector.
- This is because blockchain first found many use cases in the finance industry and, therefore, sparked enormous research interest in the financial industry long before other areas.
- Recent work in the smart contract space specific to the financial sector has proposed the idea of smart contract templates.
- The idea is to build standard templates that provide a framework to support legal agreements for financial instruments.
- **Christopher D. Clack et al.** proposed this idea in their paper published in 2016, named ***Smart Contract Templates: Foundations, design landscape and research directions*.**
- Smart Contract Templates provide a framework to support complex legal agreements for financial instruments, based on standardised templates.
- Following Grigg's Ricardian Contract triple , they use parameters to connect legal prose to the corresponding computer code, with the aim of providing a legally-enforceable foundation for smart legal contracts.

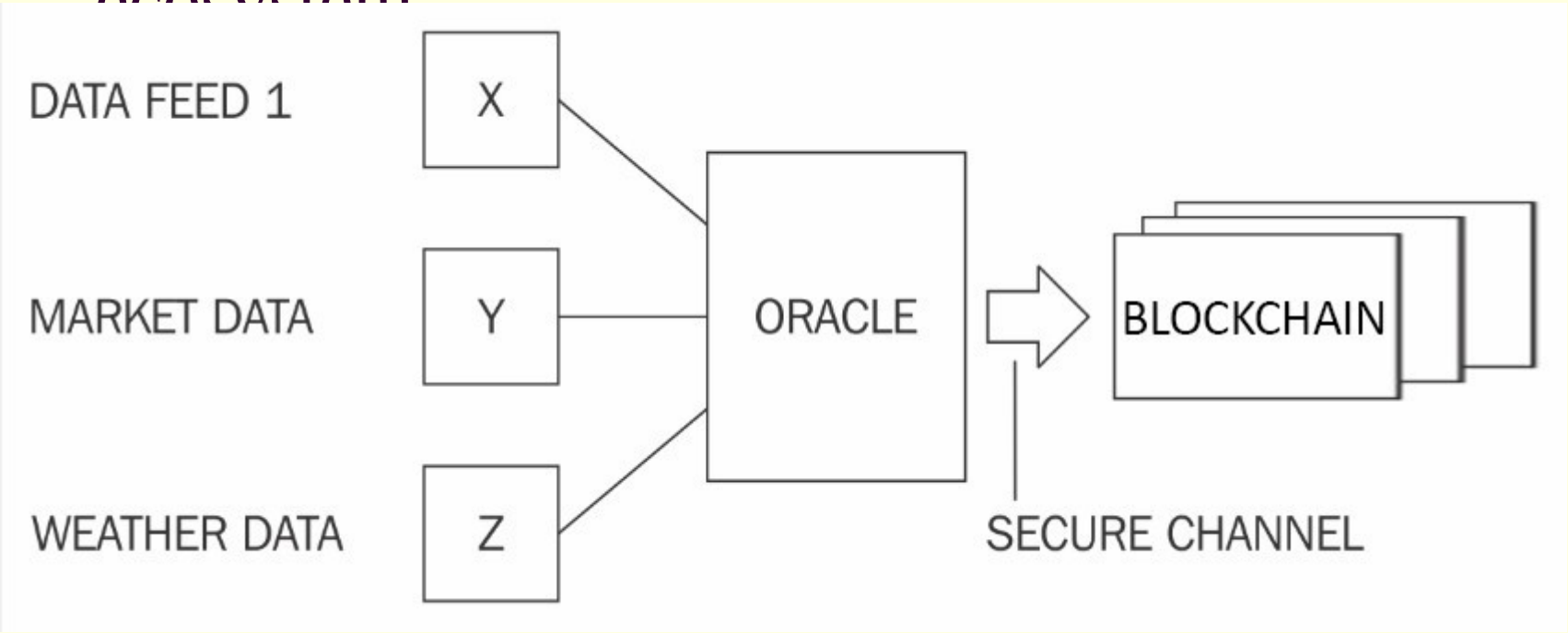
- The paper also suggested that **domain-specific languages** (**DSLs**) should be built to support the design and implementation of smart contract templates.
- A language named **common language for augmented contract knowledge** (**CLACK**) has been proposed, and research has started to develop this language.
- This language is intended to be very rich and is expected to provide a large variety of functions ranging from supporting legal prose to the ability to be executed on multiple platforms and cryptographic functions.
- Clack et al. also carried out work to develop smart contract templates that support legally enforceable smart contracts.
- This proposal has been discussed in their research paper, *Smart Contract Templates: essential requirements and design options*.
- The main aim of this paper is to investigate how legal prose could be linked with code using a markup language. It also covers how smart legal agreements can be created, formatted, executed, and serialized for storage and transmission.
- This work is ongoing and remains an open area for further research and development.

- Contracts in the finance industry are not a new concept, and various DSLs are already in use in the financial services industry to provide a specific language for a particular domain.
- For example, there are DSLs available that support the development of insurance products, represent energy derivatives, or are being used to build trading strategies.
- It is also essential to understand the concept of DSLs, as this type of programming language can be developed to program smart contracts.
- DSLs are different from **general-purpose programming languages** (**GPLs**).
- DSLs have limited expressiveness for a particular application or area of interest.
- These languages possess a small set of features that are sufficient and optimized for a specific domain only.
- Unlike GPLs, they are not suitable for building large general-purpose application programs.

Oracles

- Oracles are an essential component of the smart contract and blockchain ecosystem.
- The limitation with smart contracts is that **they cannot access external** data because **blockchains are closed systems without any direct access to the real world.**
- This external data might be required to control the execution of some business logic in the smart contract; for example, the stock price of a security product that is required by the contract to release dividend payments.
- In such situations, oracles can be used to provide external data to smart contracts.
- An oracle can be defined as **an interface that delivers data from an external source to smart contracts.**
- Oracles are trusted entities that use a secure channel to transfer off-chain data to a smart contract.

A generic model of an oracle and smart contract ecosystem



Depending on the industry and use case requirements, oracles can deliver different types of data ranging from weather reports, real-world news, and corporate actions to data coming from an **Internet of Things** (**IoT**) device.

Type of data	Examples	Use case
Market data	Live price feeds of financial instruments. Exchange rates, performance, pricing, and historic data of commodities, indices, equities, bonds, and currencies.	DApps related to financial services, for example, decentralized exchanges and **decentralized finance** (**DeFi**)
Political events	Election results	Prediction markets
Travel information	Flight schedules and delays	Insurance DApps
Weather information	Flooding, temperature, and rain data	Insurance DApps
Sports	Results of football, cricket, and rugby matches	Prediction markets
Telemetry	Hardware IoT devices, sensor data, vehicle location, and vehicle tracker data	Insurance DApps Vehicle fleet management DApps

- Here are different methods used by oracles to write data into a blockchain, depending on the type of blockchain used.
- For example, in a Bitcoin blockchain, an oracle can write data to a specific transaction, and a smart contract can monitor that transaction in the blockchain and read the data.
- Other methods include storing the fetched data in a smart contract's storage, which can then be accessed by other smart contracts on the blockchain via requests between smart contracts depending on the platform.
- For example, in Ethereum, this can be achieved by using message calls.

The standard mechanics of how oracles work is presented here:

1. A smart contract sends a request for data to an oracle.

2.The request is executed and the required data is requested from the source. There are various methods of requesting data from the source. These methods usually involve invoking APIs provided by the data provider, calling a web service, reading from a database (for example, in enterprise integration use cases where the required data may exist on a local enterprise legacy system), or requesting data from another blockchain. Sources can be any external off-chain data provider on the internet or in an internal enterprise network.

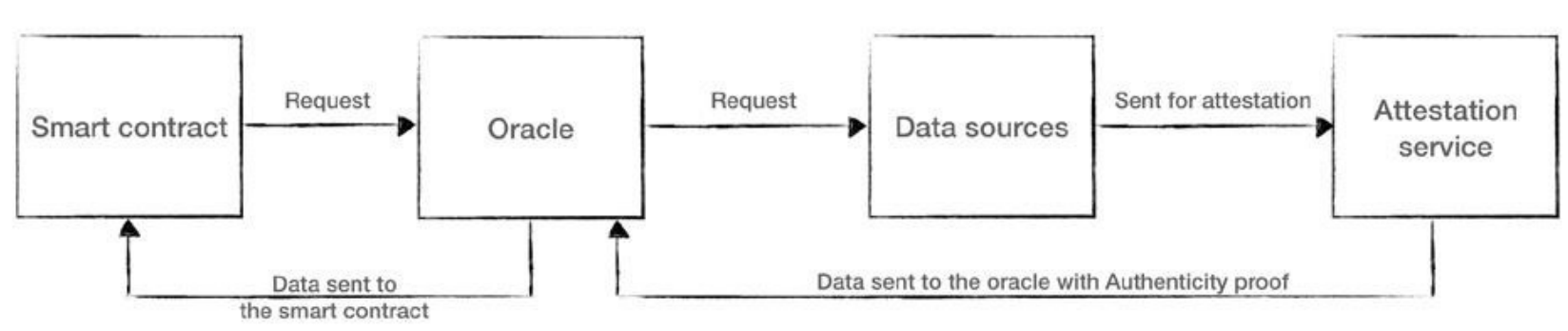
3. The data is sent to a notary to generate cryptographic proof (usually a digital signature) of the requested data to prove its validity (authenticity). Usually, TLSNotary is used for this purpose (https://tlsnotary.org). Other techniques include **Android proofs**, **Ledger proofs**, and **trusted hardware-assisted proofs**, which we will explain shortly.

4. The data with the proof of validity is sent to the oracle.

5.The requested data with its proof of authenticity can be optionally saved on a decentralized storage system such as Swarm or IPFS and can be used by the smart contract/blockchain for verification. This is especially useful when the proofs of authenticity are of a large size and sending them to the requesting smart contracts (storing them on the chain) is not feasible.

6. Finally, the data, with the proof of validity, is sent to the smart contract.

A generic oracle data flow



- The preceding diagram shows the generic data flow of a data request from a smart contract to the oracle. The oracle then requests the data from the data source, which is then sent to the attestation service for notarization. The data is sent to the oracle with proof of authenticity.
- Finally, the data is sent to the smart contract with cryptographic proof (authenticity proof) that the data is valid.
- Due to security requirements, oracles should also be capable of digitally signing or digitally attesting the data to prove that the data is authentic. This proof is called **proof of validity** or **proof of authenticity**.

- Smart contracts subscribe to oracles. Smart contracts can either pull data from oracles, or oracles can push data to smart contracts.
- It is also necessary that oracles should not be able to manipulate the data they provide and must be able to provide factual data.
- Even though oracles are trusted (due to the associated proof of authenticity of data), it may still be possible that, in some cases, the data is incorrect due to manipulation or a fault in the system.
- Therefore, oracles must not be able to modify the data.
- This validation can be provided by using various cryptographic proofing schemes.

Types of blockchain oracles

There are various types of blockchain oracles, ranging from simple software oracles to complex hardware assisted and decentralized oracles.
Categorize oracles into two categories: **inbound oracles and outbound oracles.**

Inbound oracles
This class represents oracles that receive incoming data from external services, and feed it into the smart contract.

Software oracles
Hardware oracles
Computation oracles
Aggregation based oracles
Crowd wisdom driven oracles
Decentralized oracles
Smart oracles

Inbound oracles

Software oracles

- These oracles are responsible for acquiring information from online services on the Internet. This type of oracle is usually used to source data such as weather information, financial data (stock prices, for example), travel information and other types of data from third-party providers.
- The data source can also be an internal enterprise system, which may provide some enterprise specific data. These types of oracle can also be called standard or simple oracles.

Inbound oracles

Hardware oracles

- This type of oracle is used to source data from hardware sources such as IoT devices or sensors.
- This is useful in use cases such as insurance-related smart contracts where telemetry sensors provide certain information, for example, vehicle speed and location. This information can be fed into the smart contract dealing with insurance claims and payouts to decide whether to accept a claim or not.
- Based on the information received from the source hardware sensors, the smart contract can decide whether to accept or reject the claim.
- However, this approach requires a mechanism in which hardware devices are tamperproof or tamper-resistant.
- This level of security can be achieved by providing cryptographic evidence (non-repudiation and integrity) of IoT device's data and an anti-tampering mechanism on the IoT device, which renders the device useless in case of tampering attempts.

Inbound oracles

Computation oracles

- These oracles allow computing-intensive calculations to be performed off-chain.
- As blockchain is not suitable for performing compute-intensive operations, a blockchain (that is, a smart contract on a blockchain) can request computations to be performed on off-chain highperformance computing infrastructure and get the verified results back via an oracle.
- The use of oracle, in this case, provides data integrity and authenticity guarantees.
- An example of such an oracle is Truebit (https://truebit.io).
- It allows a smart contract to submit computation tasks to oracles, which are eventually completed by miners in return for an incentive.

Inbound oracles

Aggregation based oracles

- In this scenario, a single value is sourced from many different feeds.
- As an example, this single value can be the price of a financial instrument, and it can be risky to rely upon only one feed.
- To mitigate this problem, multiple data providers can be used where all of these feeds are inspected, and finally, the price value that is reported by most of the feeds can be picked up.
- The assumption here is that if the majority of the sources reports the same price value, then it is likely to be correct.
- The collation mechanism depends on the use case: sometimes it's merely an average of multiple values, sometimes a median is taken of all the values, and sometimes it is the maximum value.
- Regardless of the aggregation mechanism, the essential requirement here is to get the value that is valid and authentic, which eventually feeds into the system.

Inbound oracles

Crowd wisdom driven oracles

- Instead, multiple public sources are used to deduce the most appropriate data eventually.
- In other words, it solves the problem where a single source of data may not be trustworthy or accurate as expected.
- If there is only one source of data, it can be unreliable and risky to rely on entirely. It may turn malicious or become genuinely faulty.
- In this case, to ensure the credibility of data provided by third-party sources for oracles, the data is sourced from multiple sources.
- These sources can be users of the system or even members of the general public who have access to and have knowledge of some data, for example, a political event or a sporting event where members of the public know the results and can provide the required data.
- Similarly, this data can be sourced from multiple different news websites.
- This data can then be aggregated, and if a sufficiently high number of the same information is received from multiple sources, then there is an increased likelihood that the data is correct and can be trusted.

Inbound oracles

Decentralized oracles

- Another type of oracles, which primarily emerged due to the decentralization requirements, is called **decentralized** oracles.
- As blockchain platforms such as Bitcoin and Ethereum are fully decentralized, it is expected that oracle services should also be decentralized.
- This way, we can address the *Blockchain Oracle Problem*.
- This type of oracle can be built based on a distributed mechanism. It can also be envisaged that the oracles can find themselves source data from another blockchain, which is driven by distributed consensus, thus ensuring the authenticity of data.
- For example, one institution running their private blockchain can publish their data feed via an oracle that can then be consumed by other blockchains.
- A decentralized oracle essentially allows off-chain information to be transferred to a blockchain without relying on a trusted third party.

Inbound oracles

Smart oracles

- An idea of smart oracle has also been proposed by Ripple labs (codius).
- Its original whitepaper is available at https://github.com/codius/codius-wiki/wiki/White-Paper#from-oraclesto- smart-oracles.
- Smart oracles are entities just like oracles, but with the added capability of executing contract code.
- Smart oracles proposed by Codius run using Google Native Client, which is a sandboxed environment for running untrusted x86 native code.

Outbound oracles

- This type, also called **reverse oracles**, are used to send data out from the blockchain smart contracts to the outside world.
- There are two possible scenarios here;
  - one is where **the source blockchain is a producer of some data such as blockchain metrics, which are needed for some other blockchain.**
    - The actual data somehow needs to be sent out to another blockchain smart contract.
    - The oher scenario is that **an external hardware device needs to perform some physical activity in response to a transaction on-chain.**
    - However, note that this type of scenario does not necessarily need an oracle, because the external hardware device can be sent a signal as a result of the smart contract event.

Deploying smart contracts

- Smart contracts may or may not be deployed on a blockchain, but it makes sense to do so on a blockchain due to the security and decentralized consensus mechanism provided by the blockchain.
- Ethereum is an example of a blockchain platform that natively supports the development and deployment of smart contracts .
- Smart contracts on an Ethereum blockchain are typically part of a **broader DApp**.
- In comparison, in a Bitcoin blockchain, the transaction timelocks, such as the **nLocktime** field, the **CHECKLOCKTIMEVERIFY (CLTV**), and the **CHECKSEQUENCEVERIFY** script operator in the Bitcoin transaction, can be seen as an enabler of a simple version of a smart contract.
- These timelocks enable a transaction to be locked until a specified time or until a number of blocks, thus enforcing a basic contract that a certain transaction can only be unlocked if certain conditions (elapsed time or number of blocks) are met.

Deploying smart contracts

- For example, you can implement conditions such as ***Pay party X, N number of bitcoins after 3 months***. However, this is very limited and should only be viewed as an example of a basic smart contract.
- In addition to the example mentioned earlier, Bitcoin scripting language, though limited, can be used to construct basic smart contracts.
- One example of a basic smart contract is to fund a Bitcoin address that can be spent by anyone who demonstrates a **hash collision attack**.
- This was a contest that was announced on the Bitcointalk forum where bitcoins were set as a reward for whoever manages to find hash collisions for hash functions.
- This conditional unlocking of Bitcoin solely on the demonstration of a successful attack is a basic type of smart contract.

Deploying smart contracts

- Various other blockchain platforms support smart contracts such as **Monax, Lisk,**
  Counterparty, Stellar, Hyperledger Fabric, Axoni core, Neo, EOSIO, and Tezos.
- Smart contracts can be developed in various languages, either **DSLs or general-purpose languages.**
- The critical requirement, however, is determinism, which is very important because it is vital that regardless of where the smart contract code executes, it produces the same result every time and everywhere.
- This requirement of the deterministic nature of smart contracts also implies that smart contract code is absolutely bug-free.
- Various languages have been developed to build smart contracts such as Solidity, which runs on **Ethereum Virtual Machine** (**EVM**).

Deploying smart contracts

- It's worth noting that there are platforms that already support mainstream languages for smart contract development, such as Lisk, which supports JavaScript.
- Another prominent example **is Hyperledger Fabric**, which supports **Golang, Java, and JavaScript for smart contract development**.
- A more recent example is EOSIO, which supports writing smart contracts in C++.
- Security is of paramount importance for smart contracts.
- However, there are many vulnerabilities discovered in prevalent blockchain platforms and relevant smart contract development languages.
- These vulnerabilities result in some high-profile incidents, such as the DAO attack.

Decentralization terminology

Autonomous agents

An **Autonomous Agent** (**AA**) is an artificially intelligent software entity that acts on the behalf of its owner to achieve some desirable goals without requiring any or minimal intervention from its owner.

Decentralized organizations

DOs are software programs that run on a blockchain and are based on the idea of actualorganizations with people and protocols. Once a DO is added to the blockchain in the form of a smart contract or a set of smart contracts, it becomes decentralized and parties interact with each other based on the code defined within the DO software.

Decentralized autonomous organizations

- Just like DOs, a **decentralized autonomous organization** (**DAO**) is also a computer program that runs on top of a blockchain, and embedded within it are governance and business logic rules.
- DAOs and DOs are fundamentally the same thing.
- The main difference, however, is that **DAOs are autonomous, which means that they are fully automated and contain artificially intelligent logic**. **DOs,** on the other hand, **lack this feature and rely on human input to execute business logic.**
- Ethereum blockchain led the way with the introduction of DAOs.
- In a DAO, the code is **considered the governing entity rather than people or paper contracts.**
- However, a human curator maintains this code and acts as a proposal evaluator for the community.
- **DAOs are capable of hiring external contractors** if enough input is received from the token holders (participants).

Decentralized autonomous organizations

The most famous DAO project is **The DAO**, which raised $168 million in its crowd funding phase.

The DAO project was designed to be a **venture capital fund** aimed at providing a decentralized business model with **no single entity as owner**.

Unfortunately, this project was hacked due to a bug in the DAO code, and millions of dollars' worth of **ether** currency (**ETH**) was siphoned out of the project and into a child DAO created by hackers.

A major **network change (hard fork)** was required on the Ethereum blockchain to reverse the impact of the hack and initiate the recovery of the funds.

This incident opened up the **debate on the security, quality**, and need for thorough testing of the code in smart contracts in order to ensure their **integrity and adequate control**.

There are other projects underway, especially in academia, that are seeking to formalize smart contract coding and testing.

Decentralized autonomous organizations

- Currently, DAOs do not have any legal status, even though they may contain some intelligent code that enforces certain protocols and conditions.
- However, these rules have **no value in the real-world legal system at present**.
- An AA (that is, a piece of code that runs without human intervention) commissioned by a law enforcement agency or regulator will contain rules and regulations that could be embedded in a DAO for the purpose of ensuring its integrity from a legalistic and compliance perspective.
- The fact that DAOs are purely decentralized entities enables them to run in any jurisdiction.

Decentralized applications

All the ideas mentioned up to this point come under the broader umbrella of decentralized applications, abbreviated to DApps.
DAOs, and DOs are DApps that run on top of a blockchain in a peer-to-peer network.
They represent the latest advancement in decentralization technology.

DApps at a fundamental level are software programs that execute using either of the following methods. They are categorized as Type 1, Type 2, or Type 3 DApps:

1. **Type 1:** Run on their own dedicated blockchain, for example, standard smart contract based DApps running on Ethereum. If required, they make use of a native token, for example, ETH on Ethereum blockchain.
2. **Type 2**: Use an existing established blockchain. that is, make use of Type 1 blockchain and bear custom protocols and tokens, for example, smart contract based tokenization DApps running Ethereum blockchain.

Decentralized applications

- An example is **DAI**, which is built on top of Ethereum blockchain, but contains its own stable coins and mechanism of distribution and control.

- Another example is **Gole m**, which has its own token GNT and a transaction framework built on top of Ethereum blockchain to provide a **decentralized marketplace** for computing power where users share their computing power with each other in a peer-to-peer network.

3. Type 3: Use the protocols of Type 2 DApps; for example, the SAFE Network uses the OMNI network protocol.

Decentralized applications

- Another example to understand the difference between different types of DApps is the USDT token (Tethers).
- The original USDT uses the OMNI layer (a Type 2 DApp) on top of the Bitcoin network. USDT is also available on Ethereum using ERC20 tokens.
- This example shows that a USDT can be considered a Type 3 DApp, where the OMNI layer protocol (a Type 2 DApp) is used, which is itself built on Bitcoin (a Type 1 DApp).
- Also, from an Ethereum point of view USDT can also be considered a Type 3 DApp in that it makes use of the Type 1 DApp Ethereum blockchain using the ERC 20 standard, which was built to operate on Ethereum.
- There are thousands of different DApps running on various platforms (blockchains) now.
- There are various categories of these DApps covering media, social, finance, games, insurance, and health.
- There are various decentralized platforms (or blockchains) running, such as Ethereum, EOS, NEO, Loom, and Steem.
- The highest number of DApps currently is on Ethereum.

Use cases of Blockchain technology –Health care

- The health industry identified as major industry that can benefit by adapting blockchain technology.
- Blockchain can provide an immutable, auditable, and transparent system that traditional P2P networks cannot.
- Also, blockchain provides a simpler, more cost-effective infrastructure compared to traditional complex PKI networks.
- In healthcare, major issues such as privacy compromises, data breaches, high costs, and fraud can arise from a lack of interoperability, overly complex processes, transparency, auditability, and control.
- Another burning issue is counterfeit medicines; especially in developing countries, this is a major cause of concern.
- With the adaptability of blockchain in the health sector, several benefits can be realized, including cost savings, increased trust, the faster processing of claims, high availability, no operational errors due to complexity in the operational procedures, and preventing the distribution of counterfeit medicines.

Use cases of Blockchain technology –Health care

- From another angle, blockchains that are providing a digital currency as an incentive for mining can be used to provide processing power to solve scientific problems.
- This helps to find cures for certain diseases.
- Examples include **FoldingCoin**, which rewards its miners with FLDC tokens for sharing their computer's processing power for solving scientific problems that require unusually large calculations.
- Another similar project is called **CureCoin,** which is available at https://www.curecoin.net/. It is yet to be seen how successful these projects will be in achieving their goals, but the idea is very promising.

Use cases of Blockchain technology –Government

There are various applications of blockchain being researched currently that can support government functions and take the current model of e-government to the next level.
some background for e-government will be provided, and then a few use cases such as **e-voting, homeland security (border control), and electronic IDs (citizen ID cards)**
Government, or electronic government, is a paradigm where information and communication technology are used to deliver public services to citizens.
The concept is not new and has been implemented in various countries around the world, but with blockchain, a new avenue of exploration has opened up.
Many governments are researching the possibility of using blockchain technology for managing and delivering public services, including, but not limited to, **identity cards, driving licenses, secure data sharing among various government departments, and contract management**.

Government-Border control

- Automated border control systems have been in use for decades now to thwart illegal entry into countries and prevent terrorism and human trafficking.
- Machine-readable travel documents, specifically **biometric passports**, have paved the way for automated border control; however, current systems are limited to a certain extent and blockchain technology can provide solutions.
- A **machine-readable travel docume nt** (**MRTD**) standard is defined in document ICAO 9303 (https://www.icao.int/publications/pages/publication.aspx?docnum=9303) by the **International Civil Aviation Organization** (**ICAO**) and has been implemented by many countries around the world.
- Each passport contains various security and identity attributes that can be used to identify the owner of the passport, and also circumvent attempts at tampering with these passports.
- These include biometric features such as retina scan, fingerprints, facial recognition, and standard ICAO specified features, including **machine-readable zone (MRZ)** and other text attributes that are visible on the first page of the passport.

Government-Border control

- One key issue with **current border control systems is data sharing**, whereby the systems are controlled by a **single entity and data is not readily shared among law enforcement agencies.**
- Another issue is related to the **imme diate imple mentation of blacklisting of a travel docume nt;** for example, when there is an immediate need to track and control suspected travel documents.
- Currently, there is no mechanism available to blacklist or revoke a suspicious passport immediately and broadcast it to the border control ports worldwide.
- The full database of all travel documents may not be stored on the blockchain currently due to inherent storage limitations, but a backend distributed database such as **BigchainDB**, **IPFS**, or **Swarm** can be used for that purpose.
- In this case, a hash of the travel document with the biometric ID of an individual can be stored in a simple smart contract, and a hash of the document can then be used to refer to the detailed data available on the distributed filesystem,such as IPFS.

Government-Voting

- Voting in any government is a key function and allows citizens to participate in the democratic election process.
- While voting has evolved into a much more mature and secure process, it still has limitations that need to be addressed to achieve a desired level of maturity.
- Usually,the limitations in current voting systems revolve around fraud, weaknesses in operational processes, and especially transparency.
- Over the years, secure voting mechanisms (machines) have been built that make use of specialized voting machines that promised security and privacy, but they still have vulnerabilities that could be exploited to subvert the security mechanisms of those machines.
- These vulnerabilities can lead to serious implications for the whole voting process and can result in mistrust in the government by the public.

Government-Voting

- Blockchain-based voting systems can resolve these issues by introducing end-to-end security and transparency in the process.
- Security is provided in the form of integrity and authenticity of votes by using public key cryptography, which comes as standard in a blockchain.
- Moreover, immutability guaranteed by blockchain ensures that votes cast once cannot be cast again.
- This can be achieved through a combination of biometric features and a smart contract maintaining a list of votes already cast.
- For example, a smart contract can maintain a list of already-cast votes with the biometric ID (for example, a fingerprint) and can use that to detect and prevent double casting. Secondly, **zero-knowledge proofs** (**ZKPs**) can also be used on the blockchain to protect voters' privacy.
- With ZKP, a voter can remain anonymous by hiding their identities, and the vote itself can be kept confidential.

Government-Citizen identification (ID cards)

Electronic IDs or national ID cards are issued by various countries around the world at present. These cards are secure and possess many security features that thwart duplication or tampering attempts.

However, with the advent of blockchain technology, several improvements can be made to this process. Digital identity is not only limited to just government-issued ID cards; it is a concept that applies to online social networks and forums, too.

There can be multiple identities being used for different purposes. A blockchain-based online digital identity allows control over personal information sharing.

Users can see who used their data and for what purpose, as well as control access to it.

This is not possible with the current infrastructures, which are centrally controlled.

The key benefit is that a single identity issued by the government can be used easily,
and in a
transparent manner, for multiple services via a single government blockchain.

Government-Citizen identification (ID cards)

- In this case, the blockchain serves as a platform where a government is providing various services such as **pensions, taxation, or benefits and a single ID is being used to access all these services.**
- Blockchain, in this case, provides a **permanent record of every change and transaction made by a digital ID, thus ensuring integrity and transparency of the system.**
- Also, citizens can **notarize birth certificates, marriages, deeds, and many other documents on the blockchain tied with their digital ID as a proof of existence.**
- Currently, there are successful implementations of identity schemes in various countries that work well, and there is an argument that perhaps blockchain is not required in identity management systems.
- Although there are several benefits, such as **privacy and controlling the use of identity information, due to the current immaturity** of blockchain technology, perhaps it is not ready for use in real-world identity systems.
- However, research is being carried out by various governments to explore the use of blockchain for identity management.

Government-Citizen identification (ID cards)

- Moreover, laws such as the right to be forgotten can be quite difficult to incorporate into blockchain due to their immutable nature.
- Other government functions where blockchain technology can be implemented to **improve cost and efficiency include the collection of taxes, benefits manage ment and disbursement, land ownership record manage ment, life event registration (marriages, births), motor vehicle registration, and licenses.**
- This is not an exhaustive list and, over time, many functions and processes of a government can be adapted to a blockchain-based model.
- The benefits of blockchain, as **immutability, transparency, and decentralization, can help to bring improve ments to most of the traditional government systems.**

Use cases of Blockchain technology –Finance

- Blockchain has many potential applications in the finance industry.
- Blockchain in finance is currently the hottest topic in the industry, and major banks and financial organizations are researching to find ways to adopt blockchain technology, primarily due to its highly desired potential to cost-save.
- These applications include, but are not limited to, insurance, post-trade settlements, financial crime prevention, and payments.

Use cases of Blockchain technology – Supply Chain Management

- A blockchain is a distributed ledger technology. In this ledger, transactions are recorded as a series of code blocks that constitute a chain.
- When the blockchain changes, each computer that has the same ledger is updated. This means that, since each block only exists in relation to its antecedent and subsequent block, the data within a block can't be changed.
- Therefore, this makes blockchain a highly verifiable, anti-tampering, transparent technology in business, and especially in supply chain management.
- Although it is still early to say that blockchain technology dominates supply chains, businesses are leveraging it in various areas and steps of their supply chain.

Use cases of Blockchain technology –Supply Chain Management

1. Supply Chain Management

With its qualities of transparency, traceability, speed, and consensus, blockchain facilitates the management of supply chains. Blockchain coordinates communication systems of the supply chain with a unified platform on the basis of its abilities for information-sharing and processing.

Also, according to a study, blockchain helps reduce the risk of certain supply chain disruptions deriving from behavioral uncertainties, fraud risks, data loss, manual errors, transactional and operational risks, and informational asymmetries.[2] Therefore, it can greatly enhance the monitoring and management of the supply chain.

2. Lowering Costs

Blockchain allows cross-border transactions. Thus, businesses can avoid intermediaries. By doing so, they not only save time but also money by cutting unnecessary costs emerging from intermediary steps.

Use cases of Blockchain technology –Supply Chain Management

3. Regulating Product Recall

Thanks to its traceability and transparency, blockchain makes it easier for supply chains to regulate product recalls by facilitating the identification and location of the affected products in the process. Thus, the recalling process becomes less expensive and time efficient.

4. Reducing Counterfeiting

Provenance is important for the quality and reliability checks of products. Since blockchain allows traceability of every step within a supply chain, the provenance of goods can be verified correctly. Thus, it helps reduce counterfeiting by allowing a quick check of the provenance of the suspected goods. According to a study by OECD, counterfeited and pirated products constituted 3.3% of world trade and 6.8% of total EU imports in 2016. This indicates that there is an alarming need to combat counterfeiting, which can be succeeded by blockchain technology.

Plus, since every transaction is under the control of authentication, blockchain can also prevent documentation fraud with the ability to verify certifications and official documents.

Use cases of Blockchain technology –Supply Chain Management

5. Maintaining Ethical Standards

Today, consumers are more aware of what ethical standards they expect from the businesses they provide their products from. 60% of consumers say they seek brands that reflect their own values of sustainability and purpose and research about the company's environmental, social, and governance (ESG) practices before buying from them.[4]

So, consumers want assurance that the products they are buying are not subjected to unethical production and delivery processes. By blockchain traceability, they can know the provenance of their products and know how the product is manufactured and shipped.

6. Logistics

We mentioned the use of smart contracts in blockchain technology. Via these smart contracts, transactions can be verified, recorded, and coordinated autonomously without third parties. Thus, a complexity element for global supply chains is alleviated. Some logistics companies like DHL are thinking of implementing blockchain use in their business.

Use cases of Blockchain technology –Supply Chain Management

7. Supplier Payments

As we mentioned, blockchain revolutionizes transaction traceability. Payments are an internal part of this. As blockchain allows automatic control of verified processes with smart contracts, once standards are met, supplier payments are done faster and with less intermediary involvement.

8. Food Safety

Blockchain is under consideration by the food industry for the food supply chain health. For example, Walmart with the collaboration of IBM innovatively uses blockchain technology to track the provenance and condition of its pork supply coming from China. With increased traceability, it is easier for the food supply chains to:

- Avoid tampering with information on the provenance of foods,
- Prevent contamination of the supplied goods with a faster and more regulated process between the processor and the distributor,
- Forestall spoilage of goods in the process between the distributor and the retailer.

Use cases of Blockchain technology –Supply Chain Management

9. Post-sale Services

With the digitalization of product information via blockchain, post-sale services such as warranties and maintenance can become more trustworthy and under control. When a buyer verifies the product's digital identity, the warranty period can start automatically. Also, second-hand buyers can confidently investigate the product's unique identity, which makes second-hand trade more trustworthy.

Blockchain and AI

- It is envisaged that other technologies, such as IoT and AI, will converge for the mutual benefit and wider adoption of both blockchain and the other given technology.
- Briefly, it can be said that due to blockchain's authenticity, integrity,privacy, and shared nature, IoT networks would benefit greatly from making use of blockchain technology. This can be realized in the form of an IoT network that runs on a blockchain, and makes use of a decentralized **mesh network** for communication in order to facilitate **Machineto- Machine** (**M2M**) communication in real time.
- All of the data that is generated as a result of M2M communication can be used in **machine learning** processes to augment the functionality of artificially intelligent DAOs or simple AAs.
- These AAs can act as agents in a blockchain-provided **Distributed Artificial Intelligence** (**DAI**) environment, which can learn over time using machine learning processes.
- This would enable them to make better decisions for the good of the blockchain.

Blockchain and AI

- AI is a field of computer science that endeavors to build intelligent agents that can make rational decisions based on the scenarios and environment that they observe around them.
- Machine learning plays a vital role in AI technology, by making use of raw data as a learning resource.
- A key requirement in AI-based systems is the availability of authentic data that can be used for machine learning and model building.
- Therefore, the explosion of data coming out of IoT devices, smartphones, and other means of data acquisition means that AI and machine learning is becoming more and more powerful.
- There is, however, a requirement for data authenticity, which is where the convergence with blockchain comes in.
- Once consumers, producers, and other entities are on a blockchain, the data that is generated as a result of interaction between these entities can be readily used as an input to machine learning engines with a guarantee of authenticity.

Blockchain and AI

- The possibility of combining intelligent oracles, intelligent smart contracts, and AAs will give rise to **Artificially Intelligent Decentralized Autonomous Organizations** (**AIDAOs**) that can act on behalf of humans to run entire organizations on their own.
- This is another side of AI that could potentially become normal in the future. However, more research is required to realize this vision.
- The convergence of blockchain technology with various other fields, such as 3D printing, virtual reality, augmented reality, spatial computing, and the gaming industry, is also envisaged.
- For example, in a multiplayer online game, blockchain's decentralized approach allows more transparency, and can ensure that no central authority is gaining an unfair advantage by manipulating game rules.