Module - 4

Smart Contracts and Use cases

Smart Contracts – Definition, Smart contract templates, Oracles, Types of oracles, Deploying smart contracts. **Decentralization terminology** – Decentralized applications, Decentralized Autonomous Organizations. **Use cases of Blockchain technology** – Government, Health care, Finance, Supply chain management. **Blockchain and allied technologies** – Blockchain and Cloud Computing, Blockchain and Artificial Intelligence.

Smart Contracts

"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."

Smart Contracts-Definition

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

- ✓ This definition reveals that a smart contract is, fundamentally, a computer program that is written in a language that a computer or target machine can understand. Also, it encompasses agreements between parties in the form of business logic. Another fundamental idea is that 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.
- ✓ Preferably, smart contracts should not rely on any traditional methods of enforcement. Instead, 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. This idea may sound like a libertarian's dream, but it is entirely possible and is in line with the true spirit of smart
- ✓ Moreover, 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:

i. **Automatically executable:** It is self-executable on a blockchain without requiring any intervention.
ii. **Enforceable:** This means that all contract conditions are enforced automatically.

iii. 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.

iv. **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.

v. **Semantically sound:** This means that they are complete and meaningful to both people and computers.

vi. **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.

It could be argued that the first four properties are required as a minimum, whereas the latter two may not be necessary or applicable in some scenarios and can be relaxed. For example, a financial derivatives contract does not, perhaps, need to be semantically sound and unstoppable but should at least be automatically executable, enforceable, deterministic, and secure. On the other hand, a title deed needs to be semantically sound and complete; therefore, for it to be implemented as a smart contract, the language that it is written in must be understood by both computers and people.

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.
- ✓ 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.
- ✓ Based on the design philosophy of DSLs, it can be envisaged that such languages will be developed specifically to write smart contracts. Some work has already been done, and Solidity is one such language that has been introduced with the Ethereum blockchain to write smart contracts. Vyper is another language that has been recently introduced for Ethereum smart contract development.
- ✓ This idea of DSLs for smart contract programming can be further extended to a GPL. A smart contract modeling platform can be developed where a domain expert (not a programmer but a front desk dealer, for example) can use a graphical user interface and a canvas (drawing area) to define and illustrate the definition and execution of a financial contract. Once the flow is drawn and completed, it can be emulated first to test it

and then be deployed from the same system to the target platform, which can be a smart contract on a blockchain or even a complete decentralized application (DApp). This concept is also not new, and a similar approach is already used in a non-blockchain domain, in the Tibco StreamBase product, which is a Javabased system used for building event-driven, high-frequency trading systems.

- ✓ It has been proposed that research should also be conducted in the area of developing high-level DSLs that can be used to program a smart contract in a user-friendly graphical user interface, thus allowing a non-programmer domain expert (for example, a lawyer) to design smart contracts.
- ✓ Apart from DSLs, there is also a growing interest in using general-purpose, already established programming languages like Java, Go, and C++ to be used for smart contract programming. This idea is appealing, especially from a usability point of view, where a programmer who is already familiar with, for example, Java, can use their skills to write Java code instead of learning a new language. The high-level language code can then be compiled into a lowlevel bytecode for execution on the target platform. There are already some examples of such systems, such as in EOSIO blockchains, where C++ can be used to write smart contracts, which are compiled down to the web assembly for execution.

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.

The following diagram shows a **generic model of an oracle and smart contract ecosystem**.

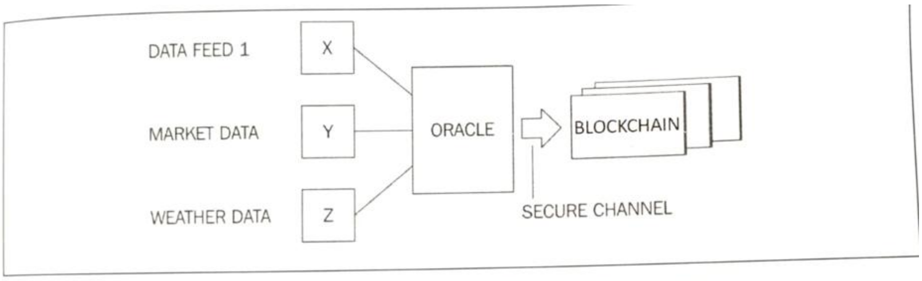
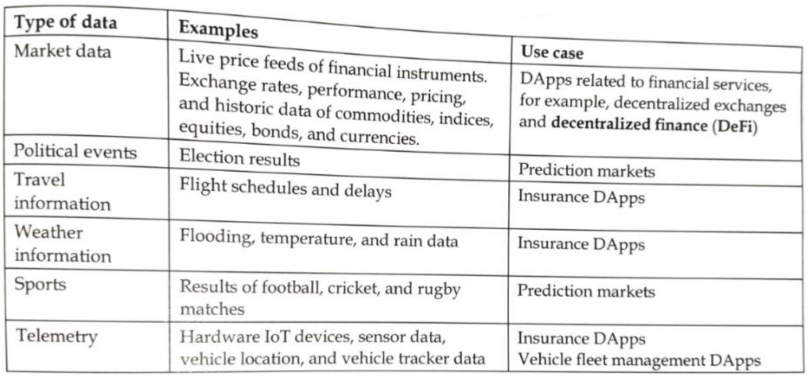


Figure 10.3: A generic model of an oracle and smart contract ecosystem

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

A list of some of the common use cases of oracles is shown here:



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 signatureof the requested data to prove its validity (authenticity). Usually, TLSNotary is used for this purpose. Other techniques include **Android proofs, Ledger proofs, and trusted hardware-assisted proofs**.
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.

This process can be visualized in the following diagram:

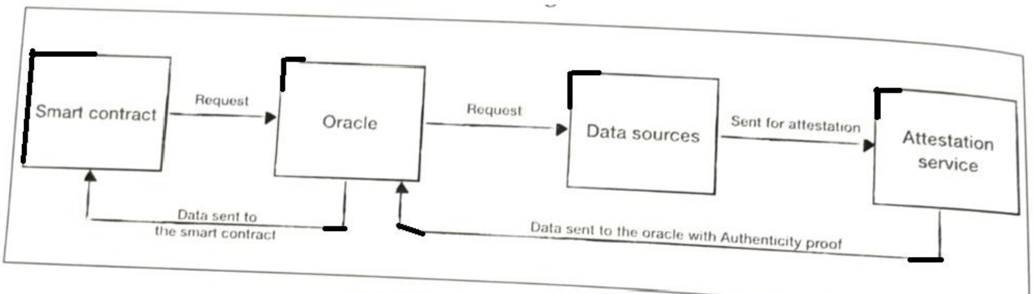


Figure 10.4: A generic oracle data flow

- ✓ The preceding diagram shows the generic data flow of a data request from a smart contract attestation service for notarization. The data is sent to the oracle with proof of authenticity. to the oracle. The oracle then requests the data from the data source, which is then sent to the Finally, the data is sent to the smart contract with cryptographic proof (authenticity proof) that the data is valid.
- ✓ Due to 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 oracles

There are various types of blockchain oracles, ranging from simple software oracles to complex hardware assisted and decentralized oracles. Broadly speaking, we can categorize oracles into two categories: inbound oracles and outbound oracles.

1. **Inbound oracles**

- ✓ This class represents oracles that receive incoming data from external services, and feed it into the smart contract. We will shortly discuss software, hardware, and several other types of inbound oracle.

2. **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.

3. **Hardware oracle**

- ✓ 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.
- ✓ These oracles are useful in any situation where real-world data from physical devices is required. 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.

4. **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. It allows a smart contract to submit computation tasks to oracles, which are eventually completed by miners in return for an incentive.

5. **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.
- ✓ An excellent example of price feed oracles is MakerDAO price feed oracle.which collates price data from multiple external price feed sources and provides a median ETHUSD price to MakerDAO.

6. **Crowd wisdom driven oracles**

- ✓ This is another way that the blockchain oracle problem can be addressed where a single source is not trusted. 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.

7. **Decentralized oracles**

- ✓ Another type of oracles, which primarily emerged due to the decentralization requirements, is called decentralized oracles. Remember that in all types of oracles discussed so far, there are some trust requirements to be placed in a trusted third party. 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.

✓ **Augur** is a prime example of such type of oracles.

✓ The core idea behind **Augur's oracle** is that of crowd wisdom-supported oracles, in which the information about an event is acquired from multiple sources and aggregated into the most likely outcome. The sources in case of Augur are financially motivated reporters who are rewarded for correct reporting and penalized for incorrect reporting.

8. Smart oracles

✓ An idea of smart oracle has also been proposed by **Ripple labs** (**codius**). 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.

9. 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 other 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.

✓ On the other hand, it can be argued that if the hardware device is running on an external blockchain, then to get data from the source chain to the target chain, undoubtedly, will need some security guarantees that oracle infrastructure can provide. Another situation is where we need to integrate legacy enterprise systems with the blockchain. In that case, the outbound oracle would be able to provide blockchain data to the existing legacy systems. An example scenario is the settlement of a trade done on a blockchain that needs to be reported to the legacy settlement and backend reporting systems.

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.

✓ 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. 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.

- ✓ 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).** 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.

DAO

- ✓ **Decentralized Autonomous Organization (DAO),** a smart contract written to provide a platform for investment. Due to a bug, called the **reentrancy** bug, in the code, it was hacked in June 2016. An equivalent of approximately 3.6 million ether (roughly 50 million US dollars) was siphoned out of the DAO into another account.
- ✓ Even though the term hacked is used here, it was not really hacked. The smart contract did what it was asked to do but due to the vulnerabilities in the smart contracts, the attacker was able to exploit it. It can be seen as an unintentional behavior (a bug) that programmers of the DAO did not foresee. This incident resulted in a hard fork on the Ethereum blockchain, which was introduced to recover from the attack.
- ✓ The DAO attack exploited a vulnerability (reentrancy bug) in the DAO code where it was possible to withdraw tokens from the DAO smart contract repeatedly before giving the DAO contract a chance to update its internal state to indicate that how many DAO tokens have been withdrawn.
- ✓ The attacker was able to withdraw DAOs. However, before the smart contract could update its state, the attacker withdrew the tokens again. This process was repeated many times, but eventually, only a single withdrawal was logged by the smart contract, and the contract also lost record of any repeated withdrawals.
- ✓ The notion of code is the *law or unstoppable smart contracts* should be viewed with some skepticism as the implementation of these concepts is still not mature enough to deserve complete and unquestionable trust. This is evident from the events after the DAO incident, where the Ethereum foundation was able to stop and change the execution of the DAO by introducing a hard fork on the Ethereum blockchain. Though this hard fork was introduced for genuine reasons, it goes against the true spirit of decentralization, immutability, and the notion that code is the law. Subsequently, resistance against this hard fork resulted in the creation of Ethereum Classic, where a large number of users decided to keep mining on the old chain. This chain is the original, non-forked Ethereum blockchain that still contains the DAO. It can be said that on this chain, *the code is still the law.*
- ✓ The DAO attack highlights the dangers of not formally and thoroughly testing smart contracts. It also highlights the absolute need to develop a formal language for the development and verification of smart

contracts. The attack also highlighted the importance of thorough testing to avoid the issues that the DAO experienced. There have been various vulnerabilities discovered in Ethereum over the last few years regarding the smart contract development language. Therefore, it is of utmost importance that a standard framework is developed to address all these issues.

Decentralization terminology

- ✓ Decentralization is a core benefit and service provided by blockchain technology. By design, blockchain is a perfect vehicle for providing a platform that does not need any intermediaries and that can function with many different leaders chosen via consensus mechanisms. This model allows anyone to compete to become the decision-making authority. A consensus mechanism governs this competition, and the most famous method is known as Proof of Work (POW).
- ✓ Decentralization is applied in varying degrees from a semi-decentralized model to a fully decentralized one depending on the requirements and circumstances. Decentralization can be viewed from a blockchain perspective as a mechanism that provides a way to remodel existing applications and paradigms, or to build new applications, to give full control to users.

Decentralized Applications (DApp)

- ✓ All the ideas mentioned up to this point come under the broader umbrella of decentralized applications, abbreviated to DApps. DAOS, DACs, 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.

- ⅄ For example, Ethlance is a DApp that makes use of ETH to provide a job market. More information about Ethlance can be found at https://ethlance.com.

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.

- ⅄ 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 Golem, 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 OMNI network protocol.

- ⅄ 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.

Requirements of a DApp

✓ For an application to be considered decentralized, it must meet the following criteria.

1. The DApp should be fully open source and autonomous, and no single entity should be in control of a majority of its tokens. All changes to the application must be consensusdriven based on the feedback given by the community.
2. Data and records of operations of the application must be cryptographically secured and stored on a public, decentralized blockchain to avoid any central points of failure.
3. A cryptographic token must be used by the application to provide access for and incentivize those who contribute value to the applications, for example, miners in Bitcoin.
4. The tokens (if applicable) must be generated by the decentralized application using consensus and an applicable cryptographic algorithm. This generation of tokens acts as a proof of the value to contributors (for example, miners).

Generally, DApps now provide all sorts of different services, including but not limited to financial applications, gaming, social media, and health.

Operations of a DApp

Establishment of consensus by a DApp can be achieved using consensus algorithms such as PoW and Proof of Stake (PoS). So far, only PoW has been found to be incredibly resistant to attacks, as is evident from the success of and trust people have put in the Bitcoin network. Furthermore, a DApp can distribute tokens (coins) via **mining, fundraising, and development.**

Design of a DApp

✓ A DApp-pronounced Dee-App, or now more commonly rhyming with app-is a software application that runs on a decentralized network such as a distributed ledger. They have recently become very popular due to the development of various decentralized platforms such as Ethereum, EOS, and Tezos.

✓ Traditional apps commonly consist of a user interface and usually a web server or an application server and a backend database. This is a common client/server architecture. This is visualized in the following diagram:

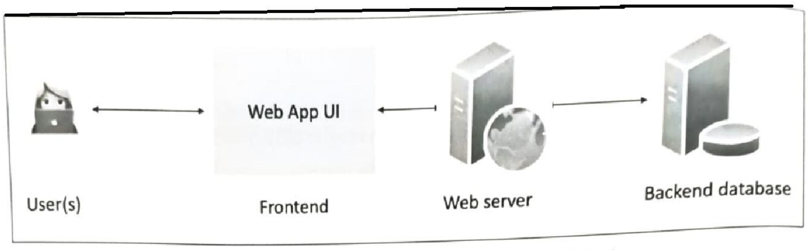
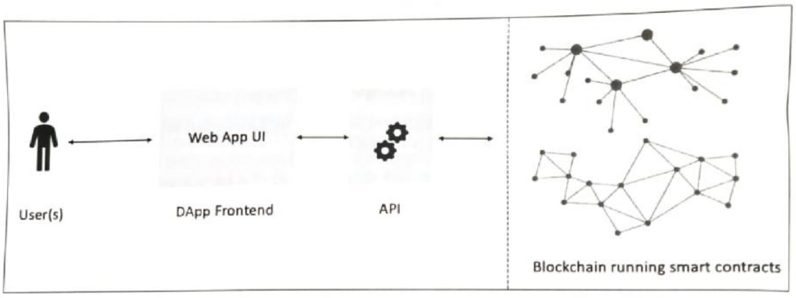


Figure : Traditional application architecture (generic client/server)

✓ A DApp on the other hand has a blockchain as a backend and can be visualized as depicted in the following diagram. The key element that plays a vital role in the creation of a DApp is a smart contract that runs on the blockchain and has business logic embedded within it:



✓ Figure :Generic DApp architecture

DApp examples

1. KYC-Chain

✓ This application provides the facility to manage Know Your Customer (KYC) data securely and conveniently based on smart contracts.

2. OpenBazaar

✓ This is a decentralized peer-to-peer network that enables commercial activities directly between sellers and buyers instead of relying on a central party, such as eBay or Amazon. It should be noted that this system is not built on top of a blockchain; instead, distributed hash tables are used in a peer-to-peer network to enable direct communication and data sharing among peers. It makes use of Bitcoin and various other cryptocurrencies as a payment method.

3. Lazooz

✓ This is the decentralized equivalent of Uber. It allows peer-to-peer ride sharing and users incentivized by proof of movement, and they can earn Zooz coins.

Decentralized Autonomous Organizations(DAO)

✓ 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).

✓ The most famous DAO project is The DAO, which raised $168 million in its crowdfunding 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.

- 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. One day, perhaps 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. Thus, they raise a big question as to how the current legal system could be applied to such a varied mix of jurisdictions and geographies.

Use cases of Blockchain technology

- Blockchain technology can be applied in various industries and use cases, as it offers several benefits such as security, immutability, transparency, and decentralization. Here are some common use cases of blockchain technologies are: Cryptocurrencies, Government, Supply Chain Management, Identity Management, Smart Contracts, Voting Systems, Healthcare, Finace,etc..
- Overall, blockchain technology has the potential to transform many industries and use cases by enabling secure, transparent, and decentralized systems.

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. First, in this section, 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)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. First, in this section, 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. Transparency, auditability, and integrity are attributes of blockchain that can go a long way in effectively managing various government functions.

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 document (MRTD)** standard is defined in document ICAO 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.

✓ 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. This lack of ability to share data makes it challenging to track suspected travel documents or individuals. Another issue is related to the immediate implementation of blacklisting of a travel document; 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.

✓ Blockchain technology can provide a solution to this problem by maintaining a blacklist in a smart contract that can be updated as required. Any changes will be immediately visible to all agencies and border control points, thus enabling immediate control over the movement of a suspected travel document. It could be argued that traditional mechanisms like Public Key Infrastructures (PKIS) and P2P networks can also be used for this purpose, but they do not provide the benefits that a blockchain can provide. With blockchain, the whole system can be simplified without the requirement of complex networks and PKI setups, which will also result in cost reduction. Moreover, blockchain-based systems will provide cryptographically guaranteed immutability, which helps with auditing and discourages any fraudulent activity.

✓ 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.

✓ This way, when a travel document is blacklisted anywhere on the network, that information will be available immediately with the cryptographic guarantee of its authenticity and integrity throughout the distributed ledger. This functionality can also provide adequate support in anti-terrorism activities, thus playing a vital role in the homeland security function of a government.

✓ A simple contract in Solidity can have an array defined for storing identities and associated biometric records. This array can be used to store the identifying information about a passport. The identity can be a hash of MRZ of the passport or travel document concatenated with the biometric record from the **RFID chip**. A simple Boolean field can be used to identify blacklisted passports.

✓ Once this initial check passes, further detailed biometric verification can be performed by traditional systems. Eventually, when a decision is made regarding the entry of the passport holder, that decision can be propagated back to the blockchain, thus enabling all participants on the network to immediately share the outcome of the decision.

✓ A high-level approach to building a blockchain-based border control system can be visualized as shown in the following diagram. In this scenario, the passport is presented for scanning to an RFID and page scanner, which reads the data page and extracts machine-readable information, along with a hash of the biometric data stored in the RFID chip. At this stage, a live photo and retina scan of the passport holder is also taken. This information is then passed on to the blockchain, where a smart contract is responsible for verifying the legitimacy of the travel document by first checking its list of blacklisted passports, and then requesting more data from the backend IPFS database for comparison. Note that the biometric data, such as a photo or retina scan, is not stored on the blockchain; instead, only a reference to this data in the backend (IPFS or BigchainDB) is stored in the blockchain.

- ✓ If the data from the presented passport matches with what is held in the **IPFS** as files or in BigchainDB and also passes the smart contract logical check, then the border gate can be opened:
- ✓ Passport Scan Scanner and RFID reader Read data Border control system frontend ID and hash Yes / No Smart contract Blockchain Check blacklist IPFS More details

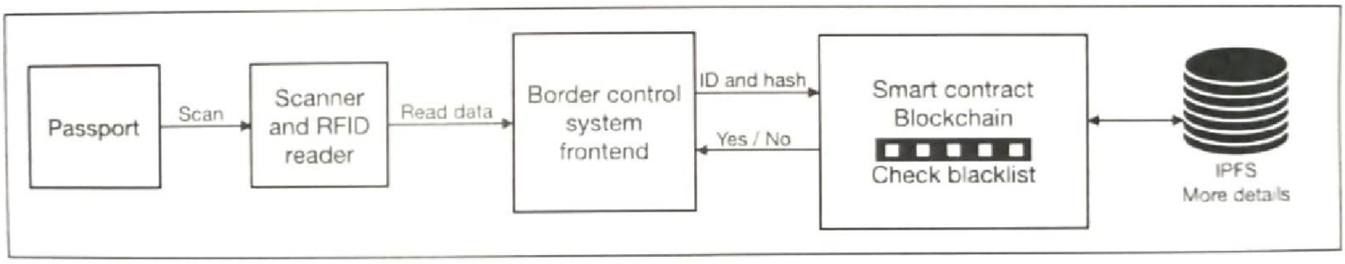


Figure 19.24: Automated border control using blockchain

- ✓ After verification, this information is propagated throughout the blockchain and is instantly available to all participants on the border control blockchain. These participants can be a worldwide consortium of homeland security departments of various nations.

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.
- ✓ 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. Moreoverimmutability 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 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.

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. 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.
- ✓ 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 management and disbursement, land ownership record management, 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 key benefits of blockchain, such as immutability, transparency, and decentralization, can help to bring improvements to most of the traditional government systems.

Health

- ✓ The health industry has also been identified as another 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.
  - ✓ 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.

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.

Insurance

- ✓ In the insurance industry, blockchain technology can help to stop fraudulent claims, increase the speed of claim processing, and enable transparency. Imagine a shared ledger between all insurers that can provide a quick and efficient mechanism for handling intercompany claims. Also, with the convergence of IoT and blockchain, an ecosystem of smart devices can be imagined, where all these things can negotiate and manage their insurance policies, which are controlled by smart contracts on the blockchain.
- ✓ Blockchain can reduce the overall cost and effort required to process claims. Claims can be automatically verified and paid via smart contracts and the associated identity of the insurance policyholder. For example, a smart contract, with the help of an oracle and possibly IoT, can make sure that when the accident occurred, it can record related telemetry data and, based on this information, release payment. It can also withhold payment if the smart contract, after evaluating conditions of payment, concludes that payment should not be released; for example, in a scenario where an authorized workshop did not repair the vehicle or was used outside a designated area and so on and so forth. There can be many conditions that a smart contract can evaluate to process claims and the choice of these rules depends on the insurer, but the general idea is that smart contracts, in combination with IoT and oracles, can automate the entire vehicle insurance industry.
- ✓ Several start-ups, such as Dynamis, have proposed smart contract-based P2P insurance platforms that run on the Ethereum blockchain. This was initially proposed to be used for unemployment insurance and does not require underwriters in the model.

Post-trade settlement

- ✓ This is the most sought-after application of blockchain technology. Currently, many financial institutions are exploring the possibility of using blockchain technology to simplify, automate, and speed up the costly and time-consuming post-trade settlement process.
- ✓ To understand the problem better, the trade lifecycle will be described briefly. A trade lifecycle contains three steps: execution, clearing, and settlement. Execution is concerned with the commitment of trading between two parties and can be entered into the system via front office order management terminals or exchanges. Clearing is the next step, whereby the trade is matched between the seller and buyer based on certain attributes, such as price and quantity. At this stage, accounts that are involved in payment are also identified. Finally, the settlement is where, eventually, security is exchanged for payment between the buyer and seller.
- ✓ In the traditional trade lifecycle model, a central clearing house is required to facilitate trading between parties, which bears the credit risk of both parties. The current scheme is somewhat complicated, whereby a seller and buyer have to take a complicated route to trade with each other. This comprises various firms, brokers, clearing houses, and custodians, but with blockchain, a single distributed ledger with appropriate smart contracts can simplify this whole process and can enable buyers and sellers to talk directly to each other.
- ✓ Notably, the post-trade settlement process usually takes two to three days, and has a dependency on central clearing houses and reconciliation systems. With the shared ledger approach, all participants on the blockchain can immediately see a single version of truth regarding the state of the trade. Moreover, P2P settlement is possible, which results in the reduction of complexity, cost, risk, and the time it takes to settle the trade. Finally, intermediaries can be eliminated by making use of the appropriate smart contracts on the blockchain. Also, regulators can view the blockchain for auditing and regulatory requirements.

Financial crime prevention

- ✓ **Know Your Customer** (**KYC**) and **Anti Money Laundering** (**AML**) are the key enablers for the prevention of financial crime. In the case of KYC, currently, each institution maintains their own copy of customer data and performs verification via centralized data providers. This can be a time-consuming process and can result in delays in onboarding a new client.
- ✓ Blockchain can provide a solution to this problem by securely sharing a distributed ledger between all financial institutions that contain verified and true identities of customers. This distributed ledger can only be updated by consensus between the participants, thus providing transparency and auditability. This can not only reduce costs but also enable regulatory and compliance requirements to be satisfied in a better and consistent manner.
- ✓ In the case of AML, due to the immutable, shared, and transparent nature of blockchain, regulators can easily be granted access to a private blockchain where they can fetch data for relevant regulatory reporting. This will also result in reducing complexity and costs related to the current regulatory reporting paradigm. This is where data is fetched from various legacy and disparate systems, and then aggregated and formatted together for reporting purposes. Blockchain can provide a single shared view of all financial transactions in the system that are cryptographically secure, authentic, and auditable, thus reducing the costs and complexity associated with the currently employed regulatory reporting methods.

Payments

- ✓ A payment is a transfer of money or its equivalent from one party (the payer) to another (the payee) in exchange for services, goods, or for fulfilling a contract. Payments are usually made in the form of cash, bank transfers, credit cards, and cheques. There are various electronic payment systems in use, such as Bacs Payment Schemes Limited (Bacs) and the **Clearing House Automated Payment System** (**CHAPS**).
- ✓ All of these systems are, however, centralized and governed by traditional financial service industry codes and practices. These systems work adequately, but with the advent of blockchain, the potential of technology has arisen to address some of these limitations.
- ✓ Some of the key advantages that blockchain technology can bring to payments are listed as follows.

1. **Decentralization**
   - ⅄ Decentralization means that there is no requirement of a trusted third party to process payments. Payments can be made directly between parties without requiring any intermediary. This can result in reduced cost and faster (direct) payments between parties.

2. **Faster settlement**
   - ⅄ Settlement can be much quicker compared to the traditional network due to the active presence of all parties on the network. Payment data can be shared and seen by all parties at the same time, and due to this settlement becomes quicker and more comfortable. Moreover, there is no requirement of running lengthy reconciliation processes because the data is all there on the blockchain, shared between all parties and readily available, which removes the requirement of the lengthy reconciliation process.
3. **Better resilience**
   - ⅄ With a payment system running on a blockchain with potentially thousands of nodes around the world, the network becomes naturally resilient. It could also be argued that, with blockchain payments, there is no

downtime because blockchain does not rely on traditional disaster recovery (DR) practices and is also better protected against malicious and denial of service attacks.

Supply chain management (SCM)

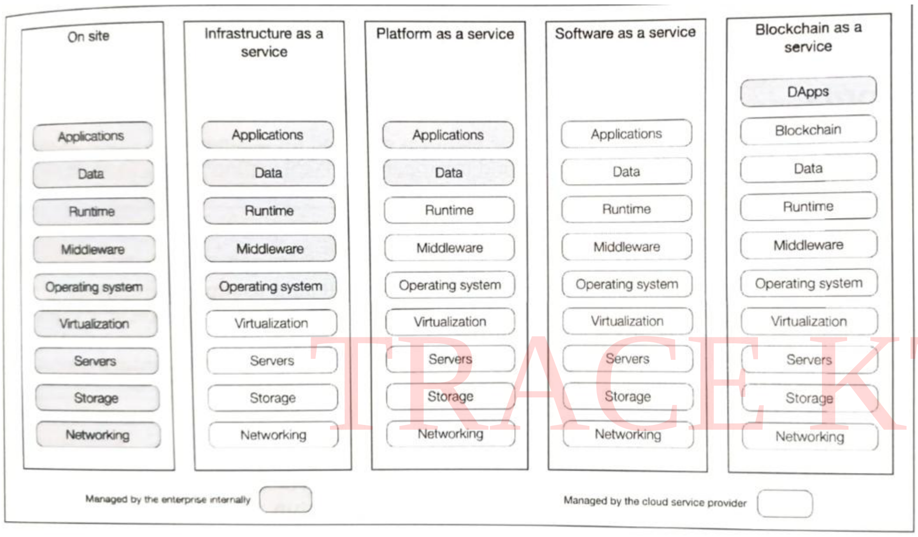
- ✓ SCM involves the coordination and management of the various activities involved in the production and delivery of goods and services to customers. These activities can include everything from sourcing raw materials to transporting finished products to retailers or end-users.
- ✓ Blockchain technology, with its decentralized and secure ledger system, has the potential to revolutionize supply chain management by providing increased transparency, traceability, and security throughout the supply chain. Here are some use cases of blockchain technology in supply chain management:
    1. **Traceability of products:** Blockchain technology can be used to track products at every stage of the supply chain, from raw material sourcing to final delivery. This allows companies to ensure that their products are sourced ethically and sustainably, and that they are not contaminated or counterfeited along the way. For example, the food industry can use blockchain to trace the origin of ingredients, making it easier to identify the source of any contamination in case of a recall.
    2. **Secure data sharing:** Blockchain technology allows companies to securely share data with their supply chain partners, without the need for a central authority or intermediary. This can help reduce the risk of data breaches and ensure that sensitive information, such as pricing or intellectual property, remains confidential.
    3. **Streamlined logistics:** Blockchain technology can be used to automate supply chain processes, such as order tracking and payment processing. This can help reduce the time and cost associated with manual processes, as well as improve accuracy and reduce errors.
    4. **Smart contracts:** Smart contracts, which are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code, can be used to automate supply chain processes and ensure compliance with regulations. For example, a smart contract can be used to ensure that a supplier meets certain environmental or labor standards before payment is released.
    5. **Supply chain financing:** Blockchain technology can be used to provide financing for suppliers and other participants in the supply chain. By using blockchain to secure transactions, suppliers can access financing at lower rates than they would from traditional lenders, which can help improve cash flow and reduce working capital requirements.
- ✓ Overall, blockchain technology has the potential to improve supply chain management by providing increased transparency, traceability, and security throughout the supply chain. This can help companies reduce costs, improve efficiency, and mitigate risk, while also improving sustainability and ethical practices.

Blockchain and allied technologies

- ✓ Blockchain is a decentralized, digital ledger that records transactions in a secure and transparent way. It uses cryptography to ensure the integrity and security of data, making it nearly impossible to tamper with or manipulate.
- ✓ Allied technologies refer to the various other technologies that are associated with blockchain and can be used in conjunction with it.

Blockchain and Cloud Computing

- ✓ Cloud computing provides excellent benefits to enterprises, including efficiency, cost reduction, scalability, high availability, and security. Cloud computing delivers computing services such as infrastructure, servers, databases, and software over the internet. There are different types of cloud services available; a standard comparison is made between Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). A question arises here: where does blockchain fit in?
- ✓ **Blockchain as a Service, or BaaS**, is an extension of **SaaS**, whereby a blockchain platform is implemented in the cloud for an organization. The organization manages its applications on the blockchain, and the rest of the software management, infrastructure management, and other aspects such as security and operating systems are managed by the cloud provider. This means that the blockchain's software and infrastructure are provided and maintained by the cloud provider. The customer or enterprise can focus on their business applications without worrying about other aspects of the infrastructure.
- ✓ The following is a comparison of different approaches:



On site	Infrastructure as a service	Platform as a service	Software as a service	Blockchain as a service
				DApps
Applications	Applications	Applications	Applications	Blockchain
Data	Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware	Middleware
Operating system	Operating system	Operating system	Operating system	Operating system
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking	Networking

Managed by the enterprise internally

Managed by the cloud service provider

- ✓ BaaS can be thought of as a SaaS, where the software is a blockchain. In this case, just like in SaaS, all services are externally managed. In other words, customers get a fully managed blockchain network on which they can build and manage their own DApps. Note that in the preceding diagram, under the Blockchain as a Service column, Applications have been replaced with Blockchain, as a differentiator between other cloud services and BaaS. Here, blockchain is the software (application) provided and managed by the cloud service provider. Also, note that DApps have been added on top, which are managed by the enterprise internally.

Blockchain and Artificial Intelligence

- ✓ 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.
- ✓ The convergence of blockchain with IoT has been discussed at length in Chapter 21, Scalability and Other Challenges. 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.

- ✓ Al 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 Al-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.
- ✓ It could also be argued that if an IoT device is hacked, it could send malformed data to the blockchain. This issue would be mitigated using blockchain technology, because an IoT device would be part of the blockchain (as a node) and would have the same security properties applied to it as a standard node in the blockchain network. These properties include the incentivization of good behavior, rejection of malformed transactions, strict verification of transactions, and various other checks that are part of blockchain protocol. Therefore, even if an IoT device is hacked, it would be treated as a Byzantine node by the blockchain network and would not cause any adverse impact on the network.
- ✓ 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 Al 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. Each of these topics are currently active areas of research, and more interest and development is expected.

IMPORTANT QUESTIONS???

1. Explain how smart contracts can be used for enforcing agreements between parties in the form of business logic.
2. Explain the concept of blockchain-based digital identity cards.
3. Illustrate how blockchain technology can be implemented in finance sector.
4. Discuss oracles in a blockchain ecosystem. Explain the generic data flow from a smart contract to an oracle.
5. Explain the design process of decentralized applications with diagrams.
6. Explain the use of blockchain technology in supply chain management.