**Practical-1**

**Aim: An introduction to Software product life cycle**

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

* SDLC is the acronym of Software Development Life Cycle.
* It is also called as Software Development Process.
* SDLC is a framework defining tasks performed at each step in the software development process.

## What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development Life Cycle consists of the following stages −

### Stage 1: Planning and Requirement Analysis: Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

### Stage 2: Defining Requirements: Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

**Stage 3: Designing the Product Architecture:** SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

**Stage 4: Building or Developing the Product:**In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

### Stage 5: Testing the Product: This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

### Stage 6: Deployment in the Market and Maintenance : Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

## SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry −

* Waterfall Model
* Iterative Model
* Spiral Model
* V-Model
* Agile Model,
* RAD Model (Rapid Application Development and Prototyping Models).

**Waterfall Model - Design**

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are −

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**Waterfall Model – Application:**

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.
* The project is short.

**Waterfall Model - Advantages**

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Phases are processed and completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Clearly defined stages.
* Well understood milestones.
* Easy to arrange tasks.
* Process and results are well documented.

**Waterfall Model - Disadvantages**

* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
* It is difficult to measure progress within stages.
* Cannot accommodate changing requirements.
* Adjusting scope during the life cycle can end a project.
* Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

**Iterative Model**:

The Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

**Iterative Model - Design**

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

The following illustration is a representation of the Iterative and Incremental model −



Iterative and Incremental development is a combination of both iterative design or iterative method and incremental build model for development. "During software development, more than one iteration of the software development cycle may be in progress at the same time." This process may be described as an "evolutionary acquisition" or "incremental build" approach."

**Iterative Model - Application**

Like other SDLC models, Iterative and incremental development has some specific applications in the software industry. This model is most often used in the following scenarios −

* Requirements of the complete system are clearly defined and understood.
* Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
* There is a time to the market constraint.
* A new technology is being used and is being learnt by the development team while working on the project.
* Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.
* There are some high-risk features and goals which may change in the future.

**Advantages**

* Some working functionality can be developed quickly and early in the life cycle.
* Results are obtained early and periodically.
* Parallel development can be planned.
* Progress can be measured.
* Less costly to change the scope/requirements.
* Testing and debugging during smaller iteration is easy.
* Risks are identified and resolved during iteration; and each iteration is an easily managed milestone.
* Easier to manage risk - High risk part is done first.
* With every increment, operational product is delivered.
* Issues, challenges and risks identified from each increment can be utilized/applied to the next increment.
* Risk analysis is better.
* It supports changing requirements.
* Initial Operating time is less.
* Better suited for large and mission-critical projects.
* During the life cycle, software is produced early which facilitates customer evaluation and feedback.

**Disadvantages:**

* More resources may be required.
* Although cost of change is lesser, but it is not very suitable for changing requirements.
* More management attention is required.
* System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle.
* Defining increments may require definition of the complete system.
* Not suitable for smaller projects.
* Management complexity is more.
* End of project may not be known which is a risk.
* Highly skilled resources are required for risk analysis.
* Projects progress is highly dependent upon the risk analysis phase.

**Spiral Model:**

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

## Spiral Model - Design

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

### Identification

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

### Design

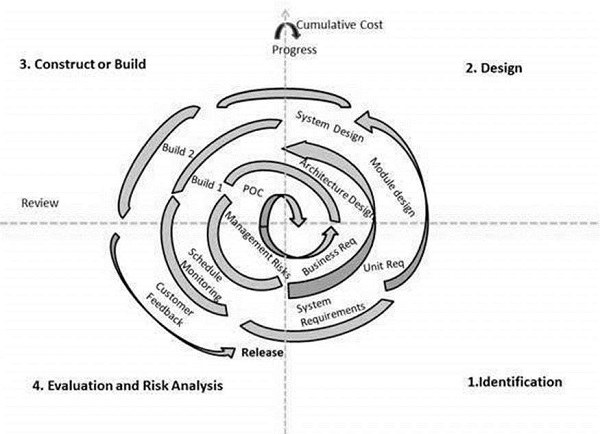
The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

### Construct or Build

The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

### Evaluation and Risk Analysis

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.



## Spiral Model Application

The Spiral Model is widely used in the software industry as it is in sync with the natural development process of any product, i.e. learning with maturity which involves minimum risk for the customer as well as the development firms.

The following pointers explain the typical uses of a Spiral Model −

* When there is a budget constraint and risk evaluation is important.
* For medium to high-risk projects.
* Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
* Customer is not sure of their requirements which is usually the case.
* Requirements are complex and need evaluation to get clarity.
* New product line which should be released in phases to get enough customer feedback.
* Significant changes are expected in the product during the development cycle.

## Advantages

* Changing requirements can be accommodated.
* Allows extensive use of prototypes.
* Requirements can be captured more accurately.
* Users see the system early.
* Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

**Disadvantages**

* Management is more complex.
* End of the project may not be known early.
* Not suitable for small or low risk projects and could be expensive for small projects.
* Process is complex
* Spiral may go on indefinitely.
* Large number of intermediate stages requires excessive documentation.

**V-model:**

The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as **Verification and Validation model**.The V-Model is an extension of the waterfall model and is based on the association of a testing p hase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

## V-Model – Design: Under the V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the ‘V’ and Validation phases on the other side. The Coding Phase joins the two sides of the V-Model.

The following illustration depicts the different phases in a V-Model of the SDLC.



## 4.2 V-Model - Verification Phases

There are several Verification phases in the V-Model, each of these are explained in detail below.

### Business Requirement Analysis

This is the first phase in the development cycle where the product requirements are understood from the customer’s perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirement. This is a very important activity and needs to be managed well, as most of the customers are not sure about what exactly they need. The **acceptance test design planning** is done at this stage as business requirements can be used as an input for acceptance testing.

### System Design

Once you have the clear and detailed product requirements, it is time to design the complete system. The system design will have the understanding and detailing the complete hardware and communication setup for the product under development. The system test plan is developed based on the system design. Doing this at an earlier stage leaves more time for the actual test execution later.

### Architectural Design

Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken. The system design is broken down further into modules taking up different functionality. This is also referred to as **High Level Design (HLD)**.

### Module Design

In this phase, the detailed internal design for all the system modules is specified, referred to as **Low Level Design (LLD)**. It is important that the design is compatible with the other modules in the system architecture and the other external systems. The unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. These unit tests can be designed at this stage based on the internal module designs.

## V- Model ─ Application: V- Model application is almost the same as the waterfall model, as both the models are of sequential type. Requirements have to be very clear before the project starts, because it is usually expensive to go back and make changes. This model is used in the medical development field, as it is strictly a disciplined domain.

The following pointers are some of the most suitable scenarios to use the V-Model application.

* Requirements are well defined, clearly documented and fixed.
* Product definition is stable.
* Technology is not dynamic and is well understood by the project team.
* There are no ambiguous or undefined requirements.
* The project is short.

## V-Model - Pros and Cons

The advantages of the V-Model method are as follows −

* This is a highly-disciplined model and Phases are completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Simple and easy to understand and use.
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.

The disadvantages of the V-Model method are as follows −

* High risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.
* Once an application is in the testing stage, it is difficult to go back and change a functionality.
* No working software is produced until late during the life cycle.

**Agile SDLC model:**

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like −

* Planning
* Requirements Analysis
* Design
* Coding
* Unit Testing and
* Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

**What is Agile?**

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Here is a graphical illustration of the Agile Model −



The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

Following are the Agile Manifesto principles −

* **Individuals and interactions** − In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
* **Working software** − Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
* **Customer collaboration** − As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
* **Responding to change** − Agile Development is focused on quick responses to change and continuous development.

**Agile Model - Pros and Cons**

Agile methods are being widely accepted in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the Agile model.

The advantages of the Agile Model are as follows −

* Is a very realistic approach to software development.
* Promotes teamwork and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* Suitable for fixed or changing requirements
* Delivers early partial working solutions.
* Good model for environments that change steadily.
* Minimal rules, documentation easily employed.
* Enables concurrent development and delivery within an overall planned context.
* Little or no planning required.
* Easy to manage.
* Gives flexibility to developers.

The disadvantages of the Agile Model are as follows −

* Not suitable for handling complex dependencies.
* More risk of sustainability, maintainability and extensibility.
* An overall plan, an agile leader and agile PM practice is a must without which it will not work.
* Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
* Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
* There is a very high individual dependency, since there is minimum documentation generated.
* Transfer of technology to new team members may be quite challenging due to lack of documentation.

**RAD (Rapid Application Development:**

The **RAD (Rapid Application Development)** model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery.

## What is RAD?

Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product.

In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process.

RAD projects follow iterative and incremental model and have small teams comprising of developers, domain experts, customer representatives and other IT resources working progressively on their component or prototype.

The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable.

## RAD Model Design

RAD model distributes the analysis, design, build and test phases into a series of short, iterative development cycles.

Following are the various phases of the RAD Model −

### Business Modeling

The business model for the product under development is designed in terms of flow of information and the distribution of information between various business channels. A complete business analysis is performed to find the vital information for business, how it can be obtained, how and when is the information processed and what are the factors driving successful flow of information.

### Data Modeling

The information gathered in the Business Modeling phase is reviewed and analyzed to form sets of data objects vital for the business. The attributes of all data sets is identified and defined. The relation between these data objects are established and defined in detail in relevance to the business model.

### Process Modeling

The data object sets defined in the Data Modeling phase are converted to establish the business information flow needed to achieve specific business objectives as per the business model. The process model for any changes or enhancements to the data object sets is defined in this phase. Process descriptions for adding, deleting, retrieving or modifying a data object are given.

### Application Generation

The actual system is built and coding is done by using automation tools to convert process and data models into actual prototypes.

### Testing and Turnover

The overall testing time is reduced in the RAD model as the prototypes are independently tested during every iteration. However, the data flow and the interfaces between all the components need to be thoroughly tested with complete test coverage. Since most of the programming components have already been tested, it reduces the risk of any major issues.

The following illustration describes the RAD Model in detail.



## RAD Model - Application

RAD model can be applied successfully to the projects in which clear modularization is possible. If the project cannot be broken into modules, RAD may fail.

The following pointers describe the typical scenarios where RAD can be used −

* RAD should be used only when a system can be modularized to be delivered in an incremental manner.
* It should be used if there is a high availability of designers for modeling.
* It should be used only if the budget permits use of automated code generating tools.
* RAD SDLC model should be chosen only if domain experts are available with relevant business knowledge.
* Should be used where the requirements change during the project and working prototypes are to be presented to customer in small iterations of 2-3 months.

## RAD Model - Pros and Cons

The advantages of the RAD Model are as follows −

* Changing requirements can be accommodated.
* Progress can be measured.
* Iteration time can be short with use of powerful RAD tools.
* Productivity with fewer people in a short time.
* Reduced development time.
* Increases reusability of components.
* Quick initial reviews occur.
* Encourages customer feedback.
* Integration from very beginning solves a lot of integration issues.

The disadvantages of the RAD Model are as follows −

* Dependency on technically strong team members for identifying business requirements.
* Only system that can be modularized can be built using RAD.
* Requires highly skilled developers/designers.
* High dependency on modelling skills.
* Inapplicable to cheaper projects as cost of modelling and automated code generation is very high.
* Management complexity is more.
* Suitable for systems that are component based and scalable.
* Requires user involvement throughout the life cycle.
* Suitable for project requiring shorter development times.

**Practical:2**

**AIM: Case study of various software development trends**

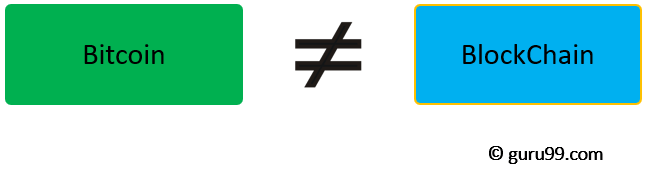
**Blockchain:**

If this technology is so complex, why call it “blockchain?” At its most basic level, blockchain is literally just a chain of blocks, but not in the traditional sense of those words. When we say the words “block” and “chain” in this context, we are actually talking about digital information (the “block”) stored in a public database (the “chain”). The blockchain is used for the secure transfer of items like money, property, contracts, etc. without requiring a third-party intermediary like bank or government. Once a data is recorded inside a blockchain, it is very difficult to change it. The blockchain is a software protocol (like SMTP is for email). However, Blockchains could not be run without the Internet. It is also called meta-technology as it affects other technologies. It is comprised of several pieces: a database, software application, some connected computers, etc.

Sometimes the term used for Bitcoin Blockchain or The Ethereum Blockchain and sometimes it's other virtual currencies or digital tokens. However, most of them are talking about the distributed ledgers. Specifically, they have three parts:

1. Blocks store information about transactions like the date, time, and dollar amount of your most recent purchase from Amazon. (NOTE: This Amazon example is for illustrative purchases; Amazon retail does not work on a blockchain principle as of this writing)
2. Blocks store information about who is participating in transactions. A block for your splurge purchase from Amazon would record your name along with Amazon.com, Inc. ([AMZN](https://www.investopedia.com/markets/quote?tvwidgetsymbol=amzn)). Instead of using your actual name, your purchase is recorded without any identifying information using a unique “digital signature,” sort of like a username.
3. Blocks store information that distinguishes them from other blocks. Much like you and I have names to distinguish us from one another, each block stores a unique code called a “hash” that allows us to tell it apart from every other block..

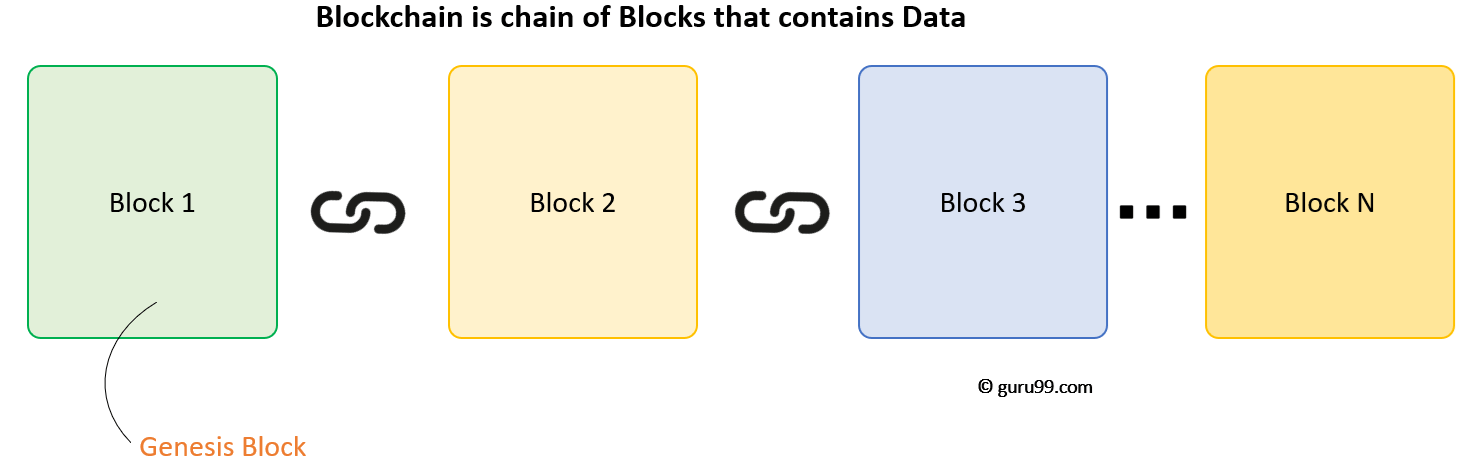
**What Blockchain is NOT!**

[](https://www.guru99.com/images/1/053018_0719_BlockchainT1.png)

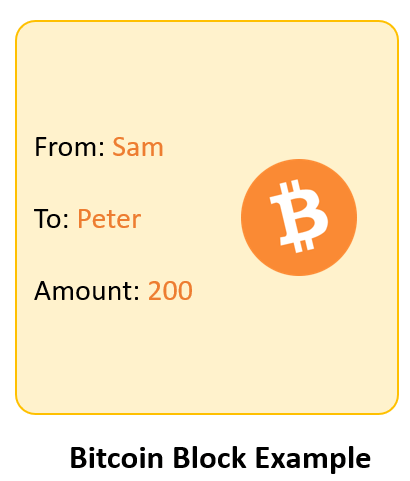
1. Blockchain is not Bitcoin, but it is the technology behind Bitcoin
2. Bitcoin is the digital token and blockchain is the ledger to keep track of who owns the digital tokens
3. You can't have Bitcoin without blockchain, but you can have blockchain without Bitcoin.

## Blockchain Architecture

**What is a Block?**

[](https://www.guru99.com/images/1/053018_0719_BlockchainT2.png)

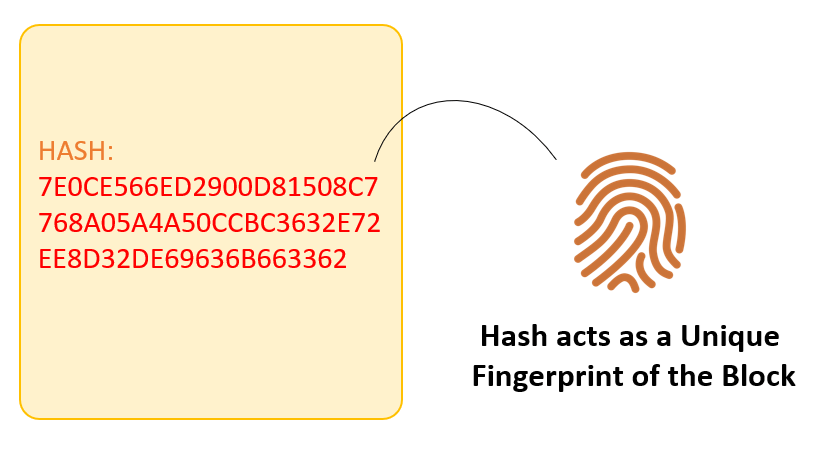
A Blockchain is a chain of blocks which contain information. The data which is stored inside a block depends on the type of blockchain. For Example, A Bitcoin Block contains information about the Sender, Receiver, number of bitcoins to be transferred.

[](https://www.guru99.com/images/1/053018_0719_BlockchainT3.png)

The first block in the chain is called the **Genesis block**. Each new block in the chain is linked to the previous block.

### Understanding SHA256 - Hash

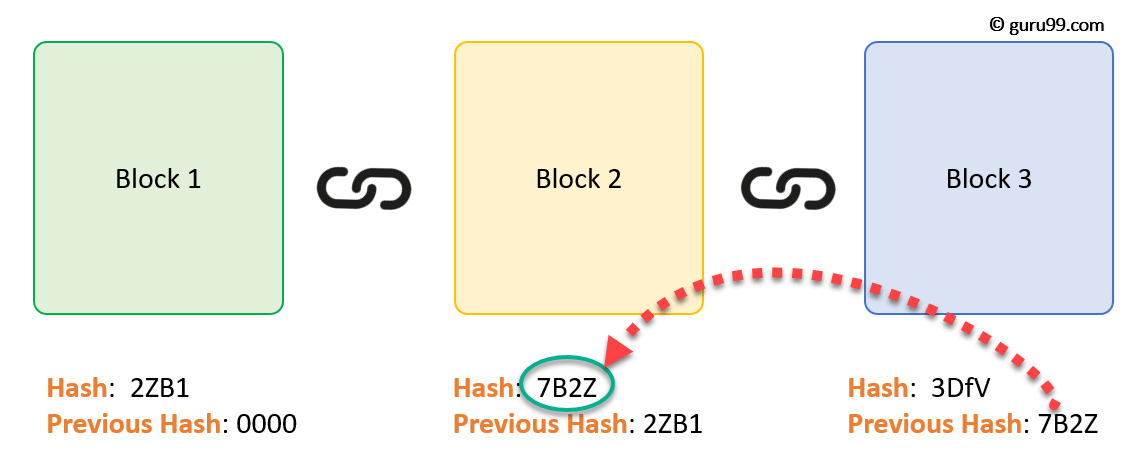
A block also has a hash. A can be understood as a fingerprint which is unique to each block. It identifies a block and all of its contents, and it's always unique, just like a fingerprint. So once a block is created, any change inside the block will cause the hash to change.

[](https://www.guru99.com/images/1/053018_0719_BlockchainT4.png)

Therefore, the hash is very useful when you want to detect changes to intersections. If the fingerprint of a block changes, it does not remain the same block. Each Block has

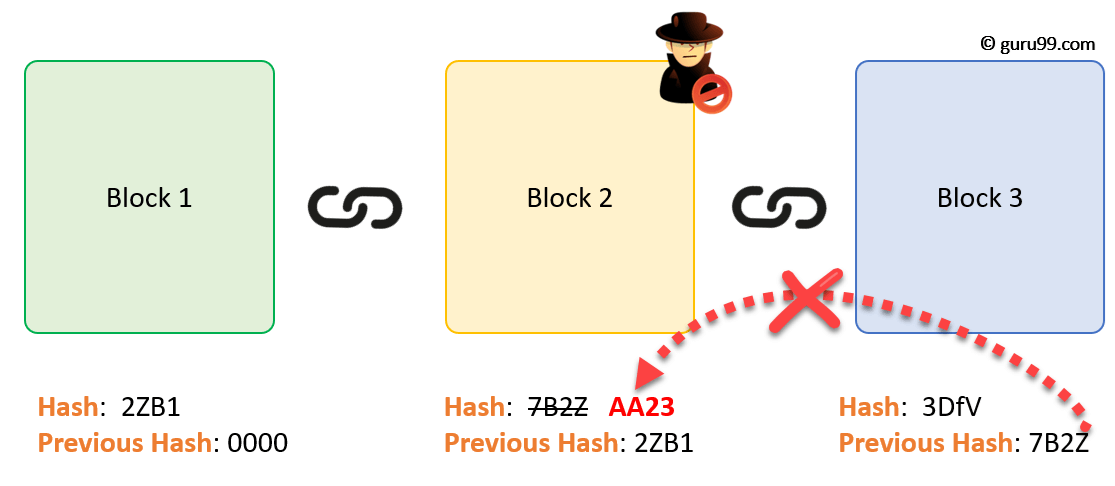
1. Data
2. Hash
3. Hash of the previous block

Consider following example, where we have a chain of 3 blocks. The 1st block has no predecessor. Hence, it does not contain has the previous block. Block 2 contains a hash of block 1. While block 3 contains Hash of block 2.

[](https://www.guru99.com/images/1/053018_0719_BlockchainT5.png)

Hence, all blocks are containing hashes of previous blocks. This is the technique that makes a blockchain so secure. Let's see how it works -

Assume an attacker is able to change the data present in the Block 2. Correspondingly, the Hash of the Block also changes. But, Block 3 still contains the old Hash of the Block 2. This makes Block 3, and all succeeding blocks invalid as they do not have correct hash the previous block.

[](https://www.guru99.com/images/1/053018_0719_BlockchainT6.png)

Therefore, changing a single block can quickly make all following blocks invalid.

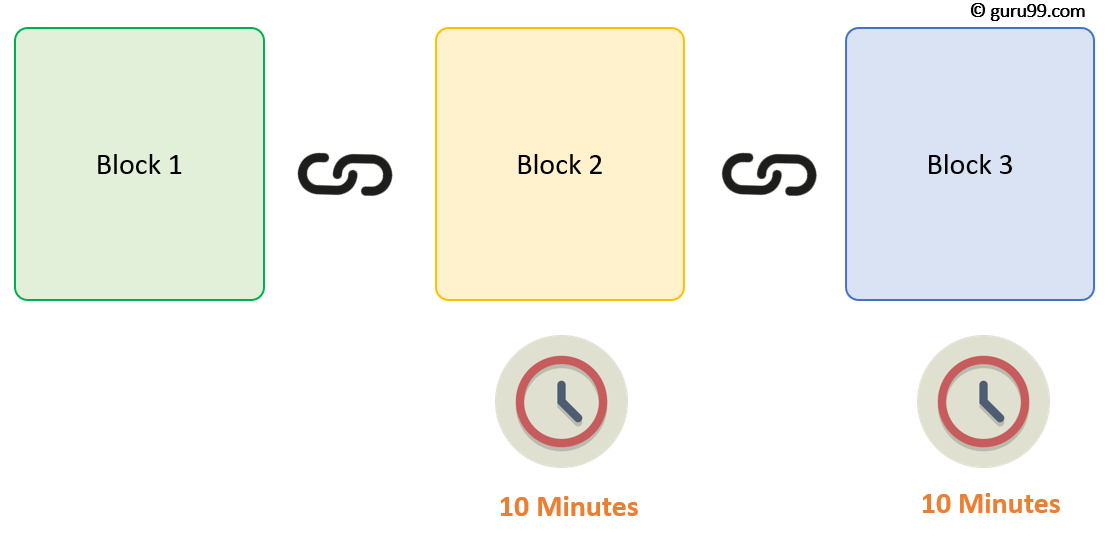
**Proof of Work**

Hashes are an excellent mechanism to prevent tempering but computers these days are high-speed and can calculate hundreds of thousands of hashes per second. In a matter of few minutes, an attacker can tamper with a block, and then recalculate all the hashes of other blocks to make the blockchain valid again.

To avoid the issue, blockchains use the concept of Proof-of-Work. It is a mechanism which slows down the creation of the new blocks.

A proof-of-work is a computational problem that takes certain to effort to solve. But the time required to verify the results of the computational problem is very less compared to the effort it takes to solve the computational problem itself.

In case of Bitcoin, it takes almost 10 minutes to calculate the required proof-of-work to add a new block to the chain. Considering our example, if a hacker would to change data in Block 2, he would need to perform proof of work (which would take 10 minutes) and only then make changes in Block 3 and all the succeeding blocks.

[](https://www.guru99.com/images/1/053018_0719_BlockchainT7.png)

This kind of mechanism makes it quite tough to tamper with the blocks so even if you tamper with even a single block, you will need to recalculate the proof-of-work for all the following blocks. Thus, hashing and proof-of-work mechanism make a blockchain secure.

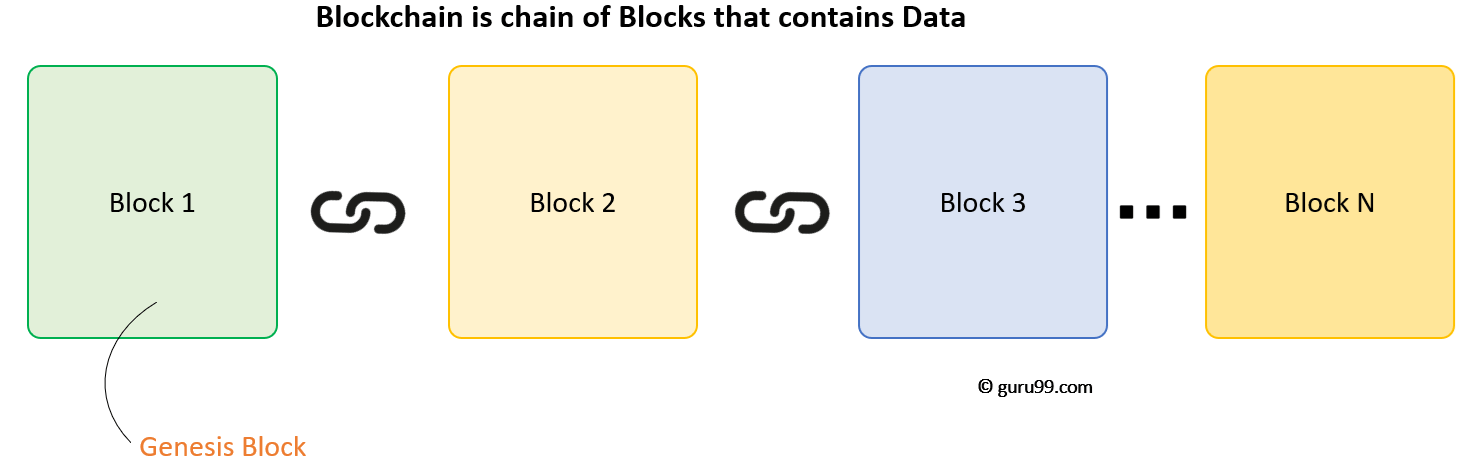
## How Blockchain Works

Blockchain consists of three important concepts: blocks, nodes and miners.

**Blocks:** Every chain consists of multiple blocks and each block has three basic elements:

* The **data** in the block.
* A 32-bit whole number called a **nonce.** The nonce is randomly generated when a block is created, which then generates a block header hash.
* The **hash** is a 256-bit number wedded to the nonce. It must start with a huge number of zeroes (i.e., be extremely small).

When the first block of a chain is created, a nonce generates the cryptographic hash. The data in the block is considered signed and forever tied to the nonce and hash unless it is mined.



### Miners: Miners create new blocks on the chain through a process called mining.

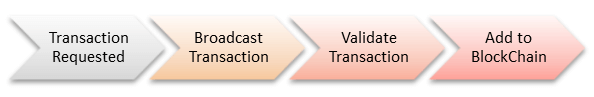
In a blockchain every block has its own unique nonce and hash, but also references the hash of the previous block in the chain, so mining a block isn't easy, especially on large chains. Miners use special software to solve the incredibly complex math problem of finding a nonce that generates an accepted hash. Because the nonce is only 32 bits and the hash is 256, there are roughly four billion possible nonce-hash combinations that must be mined before the right one is found. When that happens miners are said to have found the "golden nonce" and their block is added to the chain.

Making a change to any block earlier in the chain requires re-mining not just the block with the change, but all of the blocks that come after.

### Nodes : One of the most important concepts in blockchain technology is decentralization. No one computer or organization can own the chain. Instead, it is a distributed ledger via the nodes connected to the chain. Nodes can be any kind of electronic device that maintains copies of the blockchain and keeps the network functioning.

Every node has its own copy of the blockchain and the network must algorithmically approve any newly mined block for the chain to be updated, trusted and verified.

**How Blockchain Transaction Works?**

[](https://www.guru99.com/images/1/053018_0719_BlockchainT10.png)

**Step 1)** Some person requests a transaction. The transaction could be involved cryptocurrency, contracts, records or other information.

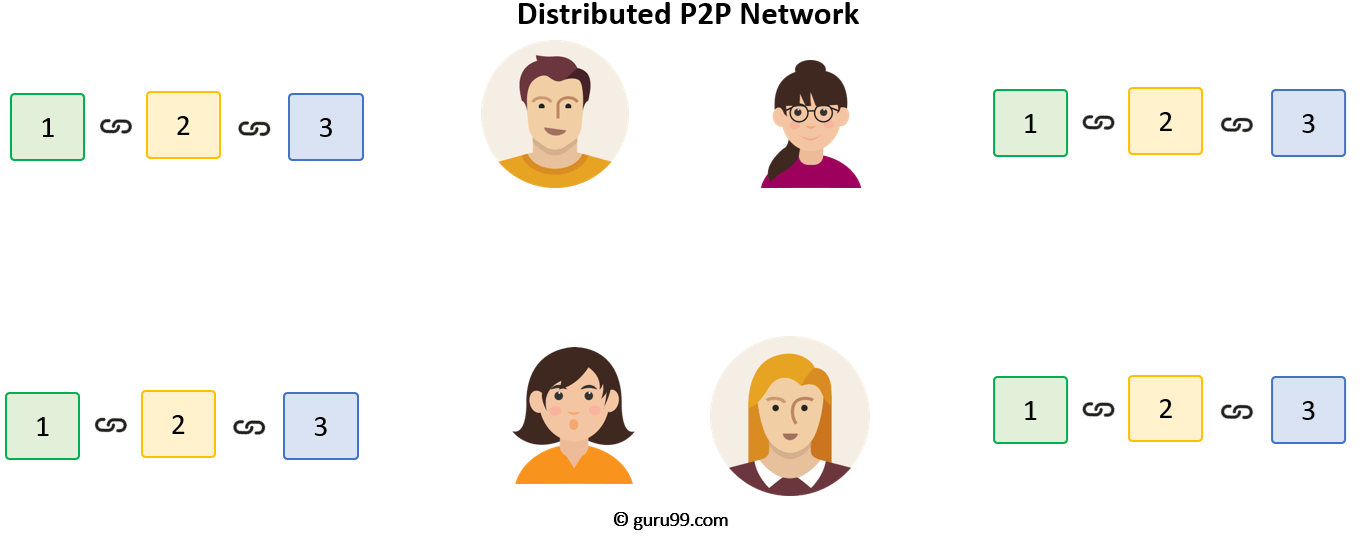
**Step 2)** The requested transaction is broadcasted to a P2P network with the help of nodes.

**Step 3)** The network of nodes validates the transaction and the user's status with the help of known algorithms.

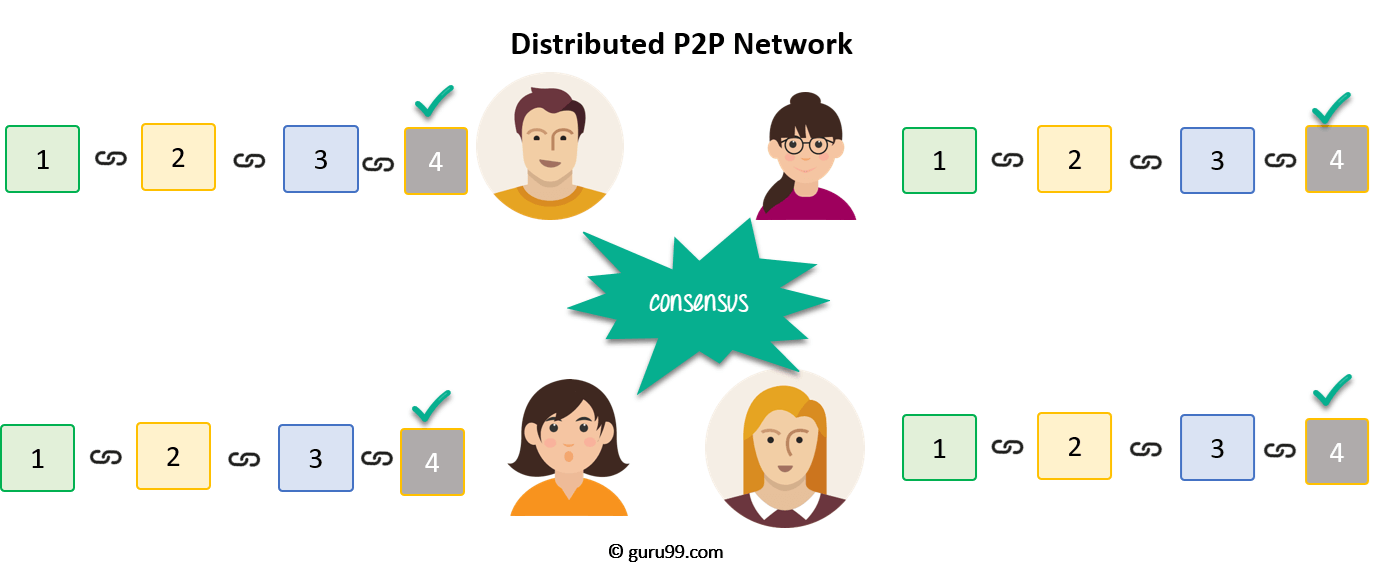
**Step 4)** Once the transaction is complete the new block is then added to the existing blockchain. In such a way that is permanent and unalterable.

**Distributed P2P Network**

However, there is one more method which is used by blockchains to secure themselves, and that's by being distributed. Instead of using a central entity to manage the chain, Blockchains use a distributed peer-peer network, and everyone is allowed to join. When someone enters this network, he will get the full copy of the blockchain. Each computer is called a node.

[](https://www.guru99.com/images/1/053018_0719_BlockchainT8.png)

Let's see what happens when any user creates a new block. This new block is sent to all the users on the network. Each node needs to verify the block to make sure that it hasn't been altered. After complete checking, each node adds this block to their blockchain.

[](https://www.guru99.com/images/1/053018_0719_BlockchainT9.png)

All these nodes in this network create a **consensus**. They agree about what blocks are valid and which are not. Nodes in the network will reject blocks that are tampered with.

So, to successfully tamper with a blockchain

1. You will need to tamper with all blocks on the chain
2. Redo the proof-of-work for each block
3. Take control of greater than 50% of the peer-to-peer network.

After doing all these, your tampered block become accepted by everyone else. This is next to impossible task. Hence, Blockchains are so secure.

**Why do we need Blockchain?**

Here, are some reasons why Blockchain technology has become so popular.

**Resilience:**Blockchains is often replicated architecture. The chain is still operated by most nodes in the event of a massive attack against the system.

**Time reduction:**In the financial industry, blockchain can play a vital role by allowing the quicker settlement of trades as it does not need a lengthy process of verification, settlement, and clearance because a single version of agreed-upon data of the share ledger is available between all stack holders.

**Reliability:**Blockchain certifies and verifies the identities of the interested parties. This removes double records, reducing rates and accelerates transactions.

**Unchangeable transactions:**By registering transactions in chronological order, Blockchain certifies the unalterability, of all operations which means when any new block has been added to the chain of ledgers, it cannot be removed or modified.

**Fraud prevention:**The concepts of shared information and consensus prevent possible losses due to fraud or embezzlement. In logistics-based industries, blockchain as a monitoring mechanism act to reduce costs.

**Security:**Attacking a traditional database is the bringing down of a specific target. With the help of Distributed Ledger Technology, each party holds a copy of the original chain, so the system remains operative, even the large number of other nodes fall.

**Transparency:**Changes to public blockchains are publicly viewable to everyone. This offers greater transparency, and all transactions are immutable.

**Collaboration** – Allows parties to transact directly with each other without the need for mediating third parties.

**Decentralized:**There are standards rules on how every node exchanges the blockchain information. This method ensures that all transactions are validated, and all valid transactions are added one by one.

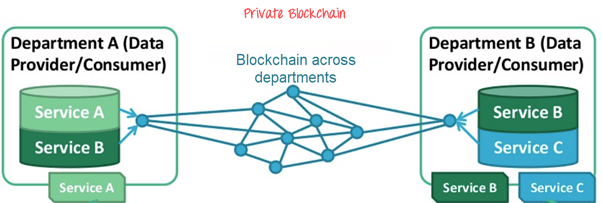
**Blockchain Variants**

**Public:**

In this type of blockchains, ledgers are visible to everyone on the internet. It allows anyone to verify and add a block of transactions to the blockchain. Public networks have incentives for people to join and free for use. Anyone can use a public blockchain network.

**Private:**

The private blockchain is within a single organization. It allows only specific people of the organization to verify and add transaction blocks. However, everyone on the internet is generally allowed to view.

[](https://www.guru99.com/images/1/053018_0719_BlockchainT12.png)

**Consortium:**

In this Blockchain variant, only a group of organizations can verify and add transactions. Here, the ledger can be open or restricted to select groups. Consortium blockchain is used cross-organizations. It is only controlled by pre-authorized nodes.

**Blockchain vs. Bitcoin**

The goal of blockchain is to allow digital information to be recorded and distributed, but not edited. That concept can be difficult to wrap our heads around without seeing the technology in action, so let’s take a look at how the earliest application of blockchain technology actually works.

Blockchain technology was first outlined in 1991 by Stuart Haber and W. Scott Stornetta, two researchers who wanted to implement a system where document timestamps could not be tampered with. But it wasn’t until almost two decades later, with the launch of Bitcoin in January 2009, that blockchain had its first real-world application.

The Bitcoin protocol is built on the blockchain. In a research paper introducing the digital currency, Bitcoin’s pseudonymous creator Satoshi Nakamoto referred to it as “a new electronic cash system that’s fully peer-to-peer, with no trusted third party.”

## Blockchain's Practical Application

### Bank Use

By integrating blockchain into banks, consumers can see their transactions processed in as little as 10 minutes, basically the time it takes to add a block to the blockchain, regardless of the time or day of the week. With blockchain, banks also have the opportunity to exchange funds between institutions more quickly and securely. In the stock trading business, for example, the settlement and clearing process can take up to three days (or longer, if banks are trading internationally), meaning that the money and shares are frozen for that time.

### Healthcare Uses

Health care providers can leverage blockchain to securely store their patients’ medical records. When a medical record is generated and signed, it can be written into the blockchain, which provides patients with the proof and confidence that the record cannot be changed. These personal health records could be encoded and stored on the blockchain with a private key, so that they are only accessible by certain individuals, thereby ensuring privacy

### Property Records Use

If you have ever spent time in your local Recorder’s Office, you will know that the process of recording property rights is both burdensome and inefficient. Today, a physical deed must be delivered to a government employee at the local recording office, where is it manually entered into the county’s central database and public index. In the case of a property dispute, claims to the property must be reconciled with the public index.

### Use in Smart Contracts

A [smart contract](https://www.investopedia.com/terms/s/smart-contracts.asp) is a computer code that can be built into the blockchain to facilitate, verify, or negotiate a contract agreement. Smart contracts operate under a set of conditions that users agree to. When those conditions are met, the terms of the agreement are automatically carried out.

Say, for example, I’m renting you my apartment using a smart contract. I agree to give you the door code to the apartment as soon as you pay me your security deposit. Both of us would send our portion of the deal to the smart contract, which would hold onto and automatically exchange my door code for your security deposit on the date of the rental. If I don’t supply the door code by the rental date, the smart contract refunds your security deposit. This eliminates the fees that typically accompany using a notary or third-party mediator.

### Uses in Voting

Voting with blockchain carries the potential to eliminate election fraud and boost voter turnout, as was [tested](https://www.technologyreview.com/s/611850/why-security-experts-hate-that-blockchain-voting-will-be-used-in-the-midterm-elections/) in the Nov. 2018 midterm elections in West Virginia. Each vote would be stored as a block on the blockchain, making them nearly impossible to tamper with. The blockchain protocol would also maintain transparency in the electoral process, reducing the personnel needed to conduct an election and provide officials with instant results.

**Myths about Blockchain**

|  |  |
| --- | --- |
| **Myth** | **Reality** |
| It solves every problem | No, it is just a database |
| Trustless Technology | It can shift trust and also spread trust |
| Secure | It focuses integrity and not confidentiality |
| Smart contracts are always legal | It only executes parts of some legal contracts |
| Immutable | It only offers probabilistic immutability |
| Need to waste electricity | Emerging blockchains are efficient |
| It is inherently unsalable | Emerging blockchains are scalable |

**Limitations of Blockchain technology**

1. Higher costs
2. Slower transactions
3. Smaller ledger
4. Transaction costs
5. network speed
6. Risk of error
7. Wasteful

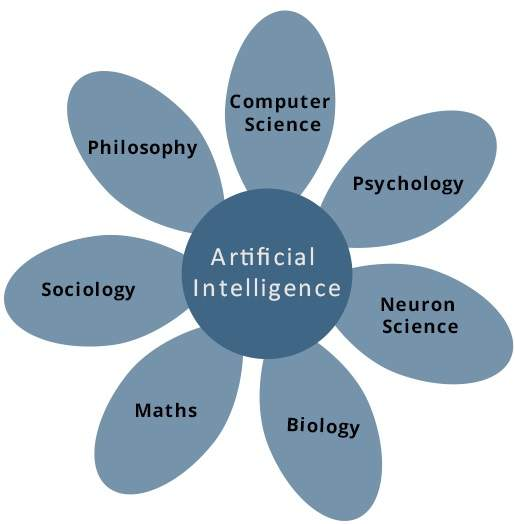
## Advantages

#### Pros

* Improved accuracy by removing human involvement in verification
* Cost reductions by eliminating third-party verification
* Decentralization makes it harder to tamper with
* Transactions are secure, private and efficient
* Transparent technology

#### ARTIFICIAL INTELLIGENCE:

With the onset of Artificial Intelligence, the machines are programmed to conduct the tasks that were restricted to the human mind. The artificial intelligence-based software can think intelligently like humans.

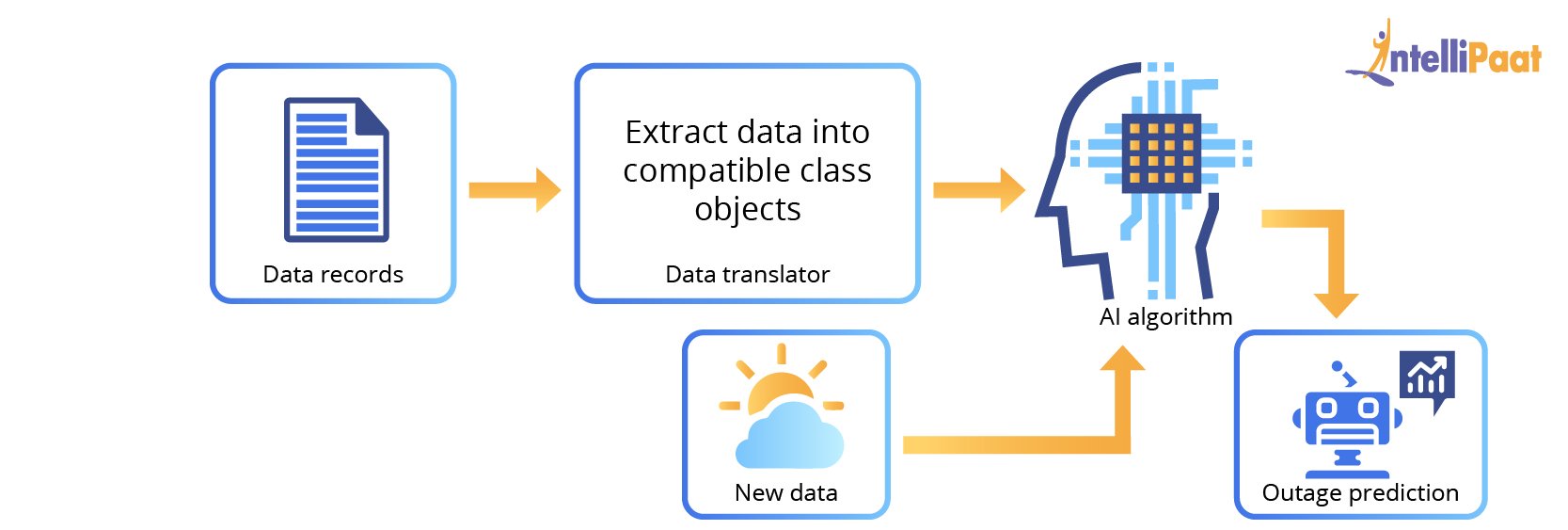


The subsets of artificial intelligence like machine learning and deep learning are gaining constant popularity among businesses. More and more companies are taking machine learning and solutions services as a necessity. The ability of a digital [computer](https://www.britannica.com/technology/computer) or computer-controlled [robot](https://www.britannica.com/technology/robot-technology) to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the [intellectual](https://www.merriam-webster.com/dictionary/intellectual) processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience. Since the development of the [digital computer](https://www.britannica.com/technology/digital-computer) in the 1940s, it has been demonstrated that computers can be programmed to carry out very complex tasks—as, for example, discovering proofs for mathematical theorems or playing [chess](https://www.britannica.com/topic/chess)—with great proficiency. Still, despite continuing advances in computer processing speed and memory capacity, there are as yet no programs that can match human flexibility over wider domains or in tasks requiring much everyday knowledge. On the other hand, some programs have attained the performance levels of human experts and professionals in performing certain specific tasks, so that artificial intelligence in this limited sense is found in applications as [diverse](https://www.merriam-webster.com/dictionary/diverse) as medical [diagnosis](https://www.merriam-webster.com/dictionary/diagnosis), computer [search engines](https://www.britannica.com/technology/search-engine), and voice or handwriting recognition..

## What Is Intelligence?

All but the simplest [human behaviour](https://www.britannica.com/topic/human-behavior) is ascribed to intelligence, while even the most complicated [insect](https://www.britannica.com/animal/insect) behaviour is never taken as an indication of intelligence. What is the difference? Consider the behaviour of the digger [wasp](https://www.britannica.com/animal/wasp), Sphex ichneumoneus. When the female wasp returns to her burrow with food, she first deposits it on the [threshold](https://www.merriam-webster.com/dictionary/threshold), checks for intruders inside her burrow, and only then, if the coast is clear, carries her food inside. The real nature of the wasp’s [instinctual behaviour](https://www.britannica.com/animal/insect/Role-of-hormones#ref41274) is revealed if the food is moved a few inches away from the entrance to her burrow while she is inside: on emerging, she will repeat the whole procedure as often as the food is displaced. Intelligence—conspicuously absent in the case of Sphex—must include the ability to adapt to new circumstances.

**How does Artificial Intelligence work?**



Computers are good at following processes, i.e., sequences of steps to execute a task. If we give a computer steps to execute a task, it should easily be able to complete it. The steps are nothing but algorithms. An algorithm can be as simple as printing two numbers or as difficult as predicting who will win elections in the coming year!

So, how can we accomplish this?

Let’s take an example of predicting the weather forecast for 2020.

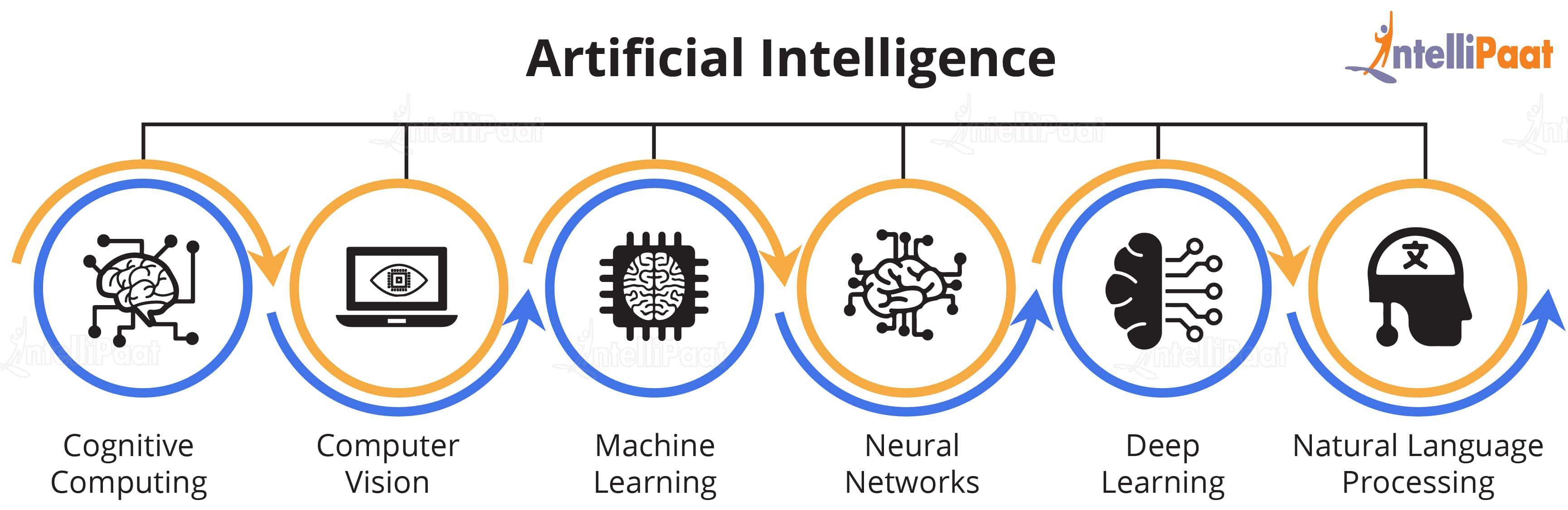
First of all, what we need is a lot of data! Let’s take the data from 2006 to 2019.

Now, we will divide this data in an 80:20 ratio. 80 percent of the data is going to be our labeled data, and the rest 20 percent will be our test data. Thus, we have the output for the entire 100 percent of the data that has been acquired from 2006 to 2019.

What happens once we collect the data? We will feed the labeled data, i.e., 80 percent of train data, into the machine. Here, the algorithm is learning from the data which has been fed into it.

Next, we need to test the algorithm. Here, we feed the test data, i.e., the remaining 20 percent of the data, to the machine. The machine gives us the output. Now, we cross verify the output given by the machine with the actual output of the data and check for its accuracy. While checking for accuracy if we are not satisfied with the model, we tweak the algorithm to give us the precise output or at least somewhere close to the actual output. Once we are satisfied with the model, we then feed the data to the model so that it can predict the weather forecast for the year 2020.

**What are the major subfields of Artificial Intelligence?**



**What are the applications of Artificial Intelligence?**

Now, it is time for us to know various real-life applications of AI across industry vertical.

1. **Fraud detection :** Every time you make a transaction online/offline, using your credit or debit card, you receive a message from your bank asking if you have made that transaction. The bank also asks you to report if you haven’t made the transaction. The bank feeds its Artificial Intelligence system with data regarding both fraudulent and non-fraudulent

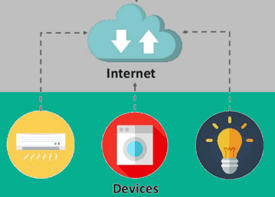
transactions. The AI system learns from this data and then predicts which transactions are fraudulent and which are not based on this huge training set.

1. **Music and movie recommendations:** Did you know that Mark Zuckerberg had created Synapse, a music player which suggested songs that users would likely to listen to? Netflix, Spotify, and Pandora also recommend music and movies for users based on their past interests and purchases. These sites accomplish this by garnering the choices users had made earlier and providing these choices as inputs into the learning algorithm.
2. **AI in retail**: The market size of AI software is expected to reach up to $36 million by 2025. This hype in the market has caused retailers to pay attention to Artificial Intelligence. Thus, the majority of big- and small-scale industries are adopting AI tools in novel ways across the entire product life cycle—right from the assembling stage to the post-sale customer–service interactions.
3. **Autopilot flight**: With the AI technology, the pilot only needs to put the system on the autopilot mode and then the majority operations of the flight will be taken care of by AI itself. It is reported by *The New York Times* that only seven minutes of human intervention (which mostly relates takeoff and landing) is required for the average flight of a Boeing plane.
4. **ROBOTICS**: Robots often have shared sets of programming that allow them to function and communicate. However, as robots are used in a variety of settings, they may need to be programmed or given instructions to make decisions and value judgments in changing environments. This will require them to possess artificial intelligence.
5. **IMAGE RECOGNITION:** While computers are getting better at recognizing voice commands, [**the ability of a program to remember and decode an image**](https://www.businessworldit.com/ai/impact-of-artificial-intelligence-on-image-recognition-techniques/) is equally appealing and has many applications for everything from security to graphic design. Large databases of tagged images are now available to tech giants like Google and Facebook, and these volumes of graphic data are being used in creative ways to create computer programs that recognize what they see in front of them, even if it is for the first time.
6. **Netflix:** Netflix needs no introduction – it is a widely popular content-on-demand service that uses predictive technology to offer recommendations on the basis of consumers’ reaction, interests, choices, and behavior. The technology examines from a number of records to recommend movies based on your previous liking and reactions.

**Internet Of Things(IoT)**

The Internet of Things(IoT) can be defined as a network of physical objects or people called "things" that are embedded with software, electronics, network, and sensors which allows these objects to collect and exchange data. The goal of IoT is to extend to internet connectivity from standard devices like computer, mobile, tablet to relatively dumb devices like a toaster. IoT makes virtually everything "smart," by improving aspects of our life with the power of data collection, AI algorithm, and networks. The thing in IoT can also be a person with a diabetes monitor implant, an animal with tracking devices, etc.

**How IOT works?**

[](https://www.guru99.com/images/1/021519_0814_InternetofT2.png)

The entire IOT process starts with the devices themselves like smartphones, smartwatches, electronic appliances like TV, Washing Machine which helps you to communicate with the IOT platform.

Devices and objects with built in sensors are connected to an [Internet of Things platform](http://www.ibm.com/internet-of-things/), which integrates data from the different devices and applies analytics to share the most valuable information with applications built to address specific needs.

These powerful IoT platforms can pinpoint exactly what information is useful and what can safely be ignored. This information can be used to detect patterns, make recommendations, and detect possible problems before they occur.

For example, if I own a car manufacturing business, I might want to know which optional components (leather seats or alloy wheels, for example) are the most popular. Using Internet of Things technology, I can:

* Use sensors to detect which areas in a showroom are the most popular, and where customers linger longest;
* Drill down into the available sales data to identify which components are selling fastest;
* Automatically align sales data with supply, so that popular items don’t go out of stock.

The information picked up by connected devices enables me to make smart decisions about which components to stock up on, based on real-time information, which helps me save time and money.

With the insight provided by advanced analytics comes the power to make processes more efficient. Smart objects and systems mean you can automate certain tasks, particularly when these are repetitive, mundane, time-consuming or even dangerous. Let’s look at some examples to see what this looks like in real life.

Here, are four fundamental components of an IoT system:

**1) Sensors/Devices:**Sensors or devices are a key component that helps you to collect live data from the surrounding environment. All this data may have various levels of complexities. It could be a simple temperature monitoring sensor, or it may be in the form of the video feed.

A device may have various types of sensors which performs multiple tasks **apart** from sensing. Example, A mobile phone is a device which has multiple sensors like GPS, camera but your smartphone is not able to sense these things.

**2) Connectivity:**All the collected data is sent to a cloud infrastructure. The sensors should be connected to the cloud using various mediums of communications. These communication mediums include mobile or satellite networks, Bluetooth, WI-FI, WAN, etc.

**3) Data Processing:**Once that data is collected, and it gets to the cloud, the software performs processing on the gathered data. This process can be just checking the temperature, reading on devices like AC or heaters. However, it can sometimes also be very complex like identifying objects, using computer vision on video.

**4)User Interface:**The information needs to be available to the end-user in some way which can be achieved by triggering alarms on their phones or sending them notification through email or text message. The user sometimes might need an interface which actively checks their IOT system. For example, the user has a camera installed in his home. He wants to access video recording and all the feeds with the help of a web server.

However, it's not always one-way communication. Depending on the IoT application and complexity of the system, the user may also be able to perform an action which may create cascading effects.

For example, if a user detects any changes in the temperature of the refrigerator, with the help of IOT technology the user should able to adjust the temperature with the help of their mobile phone.

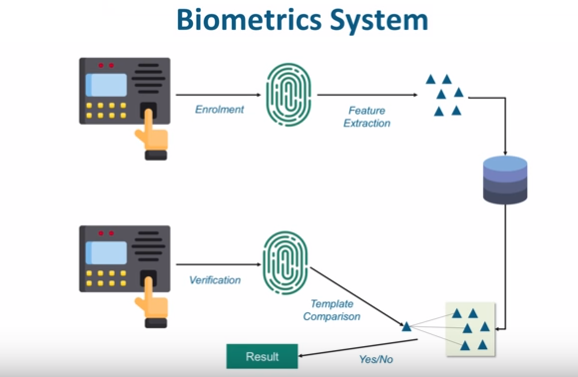
**IoT Applications**

IoT solutions are widely used in numerous companies across industries. Some most common IoT applications are given below:

|  |  |
| --- | --- |
| **Application type** | **Description** |
| Smart Thermostats | Helps you to save resource on heating bills by knowing your usage patterns. |
| Connected Cars | IOT helps automobile companies handle billing, parking, insurance, and other related stuff automatically. |
| Activity Trackers | Helps you to capture heart rate pattern, calorie expenditure, activity levels, and skin temperature on your wrist. |
| Parking Sensors | IOT technology helps users to identify the real-time availability of parking spaces on their phone. |
| Connect Health | The concept of a connected health care system facilitates real-time health monitoring and patient care. It helps in improved medical decision-making based on patient data. |
| Smart City | Smart city offers all types of use cases which include traffic management to water distribution, waste management, etc. |
| Smart home | Smart home encapsulates the connectivity inside your homes. It includes smoke detectors, home appliances, light bulbs, windows, door locks, etc. |
| Smart supply chain | Helps you in real time tracking of goods while they are on the road, or getting suppliers to exchange inventory information. |
|  |  |

# IoT Biometrics Domain

IoT plays a vital role in the Biometrics security system such as a fingerprint system, voice recognition system, eye scanner system etc.



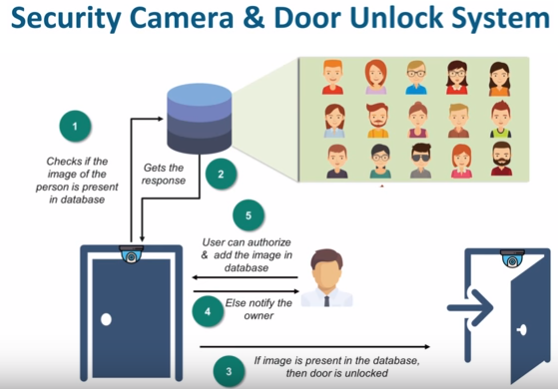
Now, a biometric system is something that we always encounter in our daily life. We always either use a fingerprint sensor or an eye-scanning system, depends on organization to organization.

Let's talk about the fingerprint scanner system and how these systems work. Now, when a person presents its finger on the fingerprint scanner it scans the fingerprint and considers this as part of an enrollment process. From this fingerprint template, the device extracts certain key features which make different from others and stores it into a database. After that, every time the same person place its finger on the top of this fingerprint scanner, it creates a template and compares this with all the templates that are present in the database. If it matches to correspondingly let's say giving that person an attendance or allows him to access a door, if it does not then it raises an alert.

This biometric system can be fingerprint or eye scanning or it could be a combination of both. Voice recognition system is also one of the key products in the biometric domain.

# IoT in Security Camera & Door Unlock System

The Security Camera and a Door unlock system is something that is quite interesting IoT application. The phenomena of its working process are briefly mentioned here.



## How does this system work?

Here, we place a camera on the top foot of the door which in turn clicks the photo of a person who comes into frame. Now, this photo is sent to an analytical system which in turn compares this with all the photos it possesses to identify whether to let the user open the door or not.

Now, if it does not find the photo of that person then it can notify the concern that a person is trying to access this door would you like to authorize this person? or would you like to deny the access to this person?

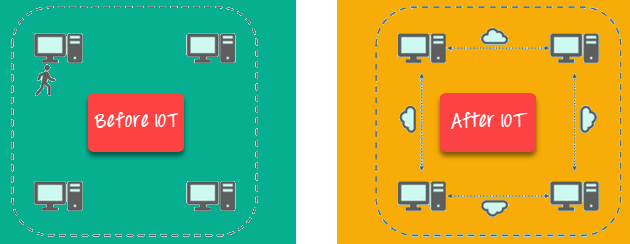
Usually, the Security Camera and a Door unlock system is used in the areas where you have highly sensitive information stored. Another usage of the security camera and door unlock system can be at our homes when we want to identify who comes to our home when we are not there and either decide to give them access to our home or not.

**Challenges of IoT**

At present IoT is faced with many challenges, such as:

* Insufficient testing and updating
* Concern regarding data security and privacy
* Software complexity
* Data volumes and interpretation
* Integration with AI and automation
* Devices require a constant power supply which is difficult
* Interaction and short-range communication

**Advantages of IoT**

[](https://www.guru99.com/images/1/021519_0814_InternetofT4.png)

Key benefits of IoT technology are as follows:

* **Technical Optimization:** IoT technology helps a lot in improving technologies and making them better. Example, with IoT, a manufacturer is able to collect data from various car sensors. The manufacturer analyzes them to improve its design and make them more efficient.
* **Improved Data Collection:** Traditional data collection has its limitations and its design for passive use. IoT facilitates immediate action on data.
* **Reduced Waste:** IoT offers real-time information leading to effective decision making & management of resources. For example, if a manufacturer finds an issue in multiple car engines, he can track the manufacturing plan of those engines and solves this issue with the manufacturing belt.
* **Improved Customer Engagement:** IoT allows you to improve customer experience by detecting problems and improving the process.

**Disadvantages IOT**

* **Security:**IoT technology creates an ecosystem of connected devices. However, during this process, the system may offer little authentication control despite sufficient security measures.
* **Privacy:** The use of IOT, exposes a substantial amount of personal data, in extreme detail, without the user's active participation. This creates lots of privacy issues.
* **Flexibility:** There is a huge concern regarding the flexibility of an IoT system. It is mainly regarding integrating with another system as there are many diverse systems involved in the process.
* **Complexity:** The design of the IOT system is also quite complicated. Moreover, it's deployment and maintenance also not very easy.
* **Compliance:** IOT has its own set of rules and regulations. However, because of its complexity, the task of compliance is quite challenging.

**Practical 3**

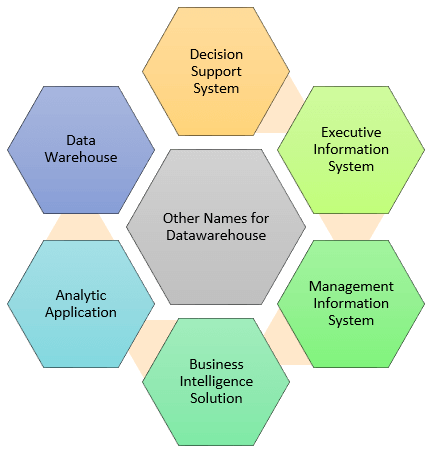
**Aim: Case study of data ware house and machine learning**

**What is Data Warehousing?**

A **Data Warehousing** (DW) is process for collecting and managing data from varied sources to provide meaningful business insights. A Data warehouse is typically used to connect and analyze business data from heterogeneous sources. The data warehouse is the core of the BI system which is built for data analysis and reporting.It is a blend of technologies and components which aids the strategic use of data. It is electronic storage of a large amount of information by a business which is designed for query and analysis instead of transaction processing. It is a process of transforming data into information and making it available to users in a timely manner to make a difference.

Data warehouse system is also known by the following name:

* Decision Support System (DSS)
* Executive Information System
* Management Information System
* Business Intelligence Solution
* Analytic Application
* Data Warehouse

[](https://www.guru99.com/images/1/data_warehousing.png)

**How Data warehouse works?**

A Data Warehouse works as a central repository where information arrives from one or more data sources. Data flows into a data warehouse from the transactional system and other relational databases.  
Data may be:

1. Structured
2. Semi-structured
3. Unstructured data

The data is processed, transformed, and ingested so that users can access the processed data in the Data Warehouse through Business Intelligence tools, SQL clients, and spreadsheets. A data warehouse merges information coming from different sources into one comprehensive database.

By merging all of this information in one place, an organization can analyze its customers more holistically. This helps to ensure that it has considered all the information available. Data warehousing makes data mining possible. Data mining is looking for patterns in the data that may lead to higher sales and profits.

**Types of Data Warehouse**

**1. Enterprise Data Warehouse:** Enterprise Data Warehouse is a centralized warehouse. It provides decision support service across the enterprise. It offers a unified approach for organizing and representing data. It also provide the ability to classify data according to the subject and give access according to those divisions.

**2. Operational Data Store:** Operational Data Store, which is also called ODS, are nothing but data store required when neither Data warehouse nor OLTP systems support organizations reporting needs. In ODS, Data warehouse is refreshed in real time. Hence, it is widely preferred for routine activities like storing records of the Employees.

**3. Data Mart:** A data mart is a subset of the data warehouse. It specially designed for a particular line of business, such as sales, finance, sales or finance. In an independent data mart, data can collect directly from sources.

**Components of Data warehouse**

**Load manager:** Load manager is also called the front component. It performs with all the operations associated with the extraction and load of data into the warehouse. These operations include transformations to prepare the data for entering into the Data warehouse.

**Warehouse Manager:**Warehouse manager performs operations associated with the management of the data in the warehouse. It performs operations like analysis of data to ensure consistency, creation of indexes and views, generation of denormalization and aggregations, transformation and merging of source data and archiving and baking-up data.

**Query Manager:** Query manageris also known as backend component. It performs all the operation operations related to the management of user queries. The operations of these Data warehouse components are direct queries to the appropriate tables for scheduling the execution of queries.

**End-user access tools:** This is categorized into five different groups like 1. Data Reporting 2. Query Tools 3. Application development tools 4. EIS tools, 5. OLAP tools and data mining tools.

**Who needs Data warehouse?**

Data warehouse is needed for all types of users like:

* Decision makers who rely on mass amount of data
* Users who use customized, complex processes to obtain information from multiple data sources.
* It is also used by the people who want simple technology to access the data
* It also essential for those people who want a systematic approach for making decisions.
* If the user wants fast performance on a huge amount of data which is a necessity for reports, grids or charts, then Data warehouse proves useful.
* Data warehouse is a first step If you want to discover 'hidden patterns' of data-flows and groupings.

**What Is a Data Warehouse Used For?**

**Social Media Websites:** The social networking websites like Facebook, Twitter, Linkedin etc. are based on analyzing large data sets. These sites gather data related to members, groups, locations etc. and store it in a single central repository. Being large amount of data, Data Warehouse is needed for implementing the same.

**Government :** Government uses data warehouse to store and analyze tax payment which is used to detect tax thefts.

**Airline:** In the Airline system, it is used for operation purpose like crew assignment, analyses of route profitability, frequent flyer program promotions, etc.

**Banking:** It is widely used in the banking sector to manage the resources available on desk effectively. Few banks also used for the market research, performance analysis of the product and operations.

**Healthcare:** Healthcare sector also used Data warehouse to strategize and predict outcomes, generate patient's treatment reports, share data with tie-in insurance companies, medical aid services, etc.

**Public sector:** In the public sector, data warehouse is used for intelligence gathering. It helps government agencies to maintain and analyze tax records, health policy records, for every individual.

**Investment and Insurance sector:** In this sector, the warehouses are primarily used to analyze data patterns, customer trends, and to track market movements.

**Retain chain:** In retail chains, Data warehouse is widely used for distribution and marketing. It also helps to track items, customer buying pattern, promotions and also used for determining pricing policy.

**Telecommunication:** A data warehouse is used in this sector for product promotions, sales decisions and to make distribution decisions.

**Hospitality Industry:** This Industry utilizes warehouse services to design as well as estimate their advertising and promotion campaigns where they want to target clients based on their feedback and travel patterns.

**Steps to Implement Data Warehouse:** The best way to address the business risk associated with a Data warehouse implementation is to employ a three-prong strategy as below

1. **Enterprise strategy**: Here we identify technical including current architecture and tools. We also identify facts, dimensions, and attributes. Data mapping and transformation is also passed.
2. **Phased delivery**: Data warehouse implementation should be phased based on subject areas. Related business entities like booking and billing should be first implemented and then integrated with each other.
3. **Iterative Prototyping**: Rather than a big bang approach to implementation, the Data warehouse should be developed and tested iteratively.

**Advantages of Data Warehouse:**

* Data warehouse allows business users to quickly access critical data from some sources all in one place.
* Data warehouse provides consistent information on various cross-functional activities. It is also supporting ad-hoc reporting and query.
* Data Warehouse helps to integrate many sources of data to reduce stress on the production system.
* Data warehouse helps to reduce total turnaround time for analysis and reporting.
* Restructuring and Integration make it easier for the user to use for reporting and analysis.
* Data warehouse allows users to access critical data from the number of sources in a single place. Therefore, it saves user's time of retrieving data from multiple sources.
* Data warehouse stores a large amount of historical data. This helps users to analyze different time periods and trends to make future predictions.

**Disadvantages of Data Warehouse:**

* Not an ideal option for unstructured data.
* Creation and Implementation of Data Warehouse is surely time confusing affair.
* Data Warehouse can be outdated relatively quickly
* Difficult to make changes in data types and ranges, data source schema, indexes, and queries.
* The data warehouse may seem easy, but actually, it is too complex for the average users.
* Despite best efforts at project management, data warehousing project scope will always increase.
* Sometime warehouse users will develop different business rules.
* Organizations need to spend lots of their resources for training and Implementation purpose.

## Data Warehouse Tools: There are many Data Warehousing tools are available in the market. Here, are some most prominent one:

**1. MarkLogic:**

**2. Oracle:**

**3. Amazon RedShift:**

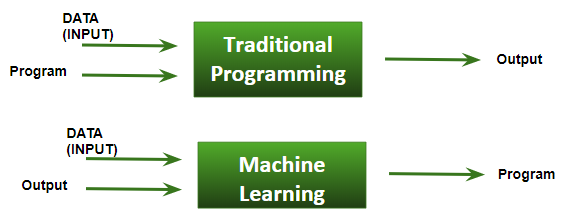
# Machine Learning

**Machine Learning** is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

**Classification of Machine Learning**

Machine learning implementations are classified into three major categories, depending on the nature of the learning “signal” or “response” available to a learning system which are as follows:-

1. **Supervised learning :** When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.
2. **Unsupervised learning** :Whereas when an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.  
   As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.
3. **Reinforcement learning :** When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error.  
   Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others. An interesting example of reinforcement learning occurs when computers learn to play video games by themselves.
4. **Semi-supervised learning :** where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing.

**Example: Training of students during exam.**  
While preparing for the exams students don’t actually cram the subject but try to learn it with complete understanding. Before the examination, they feed their machine(brain) with a good amount of high-quality data (questions and answers from different books or teachers notes or online video lectures). Actually, they are training their brain with input as well as output i.e. what kind of approach or logic do they have to solve a different kind of questions. Each time they solve practice test papers and find the performance (accuracy /score) by comparing answers with answer key given, Gradually, the performance keeps on increasing, gaining more confidence with the adopted approach. That’s how actually models are built, train machine with data (both inputs and outputs are given to model) and when the time comes test on data (with input only) and achieves our model scores by comparing its answer with the actual output which has not been fed while training. Researchers are working with assiduous efforts to improve algorithms, techniques so that these models perform even much better  
.  


**Basic Difference in ML and Traditional Programming?**

* Traditional Programming : We feed in DATA (Input) + PROGRAM (logic), run it on machine and get output.
* Machine Learning : We feed in DATA(Input) + Output, run it on machine during training and the machine creates its own program(logic), which can be evaluated while testing.

**How ML works?**

* Gathering past data in any form suitable for processing.The better the quality of data, the more suitable it will be for modeling
* Data Processing – Sometimes, the data collected is in the raw form and it needs to be pre-processed.  
  Example: Some tuples may have missing values for certain attributes, an, in this case, it has to be filled with suitable values in order to perform machine learning or any form of data mining.  
  Missing values for numerical attributes such as the price of the house may be replaced with the mean value of the attribute whereas missing values for categorical attributes may be replaced with the attribute with the highest mode. This invariably depends on the types of filters we use. If data is in the form of text or images then converting it to numerical form will be required, be it a list or array or matrix. Simply, Data is to be made relevant and consistent. It is to be converted into a format understandable by the machine
* Divide the input data into training,cross-validation and test sets. The ratio between the respective sets must be 6:2:2
* Building models with suitable algorithms and techniques on the training set.
* Testing our conceptualized model with data which was not fed to the model at the time of training and evaluating its performance using metrics such as F1 score, precision and recall.

# Machine Learning – Applications

* **Web Search Engine:** One of the reasons why search engines like google, bing etc work so well is because the system has learnt how to rank pages through a complex learning algorithm.
* **Photo tagging Applications:** Be it facebook or any other photo tagging application, the ability to tag friends makes it even more happening. It is all possible because of a face recognition algorithm that runs behind the application.
* **Spam Detector:** Our mail agent like Gmail or Hotmail does a lot of hard work for us in classifying the mails and moving the spam mails to spam folder. This is again achieved by a spam classifier running in the back end of mail application.
* **Financial Services:** Companies in the financial sector are able to identify key insights in financial data as well as prevent any occurrences of financial fraud, with the help of machine learning technology. The technology is also used to identify opportunities for investments and trade. Usage of cyber surveillance helps in identifying those individuals or institutions which are prone to financial risk, and take necessary actions in time to prevent fraud.

### Marketing and Sales: Companies are using machine learning technology to analyze the purchase history of their customers and make personalized product recommendations for their next purchase. This ability to capture, analyze, and use customer data to provide a personalized shopping experience is the future of sales and marketing.

### Government: Government agencies like utilities and public safety have a specific need FOR Ml, as they have multiple data sources, which can be mined for identifying useful patterns and insights. For example sensor data can be analyzed to identify ways to minimize costs and increase efficiency. Furthermore, ML can also be used to minimize identity thefts and detect fraud.

### Healthcare: With the advent of wearable sensors and devices that use data to access health of a patient in real time, ML is becoming a fast-growing trend in healthcare. Sensors in wearable provide real-time patient information, such as overall health condition, heartbeat, blood pressure and other vital parameters. Doctors and medical experts can use this information to analyze the health condition of an individual, draw a pattern from the patient history, and predict the occurrence of any ailments in the future. The technology also empowers medical experts to analyze data to identify trends that facilitate better diagnoses and treatment.

### Transportation: Based on the travel history and pattern of traveling across various routes, machine learning can help transportation companies predict potential problems that could arise on certain routes, and accordingly advise their customers to opt for a different route. Transportation firms and delivery organizations are increasingly using machine learning technology to carry out data analysis and data modeling to make informed decisions and help their customers make smart decisions when they travel.

**Practical 4:**

**Aim: To study various technical skills which are currently in demand for industry**

**Technical Skills You Should Have As a Developer**

If you are a computer science student or doing a course to become a software engineer or a software developer, there are some technical skills you need to have to become a good programmer. Technology is vast and there are so many tools, platforms, languages coming out in the market. It doesn’t matter if you are an experienced programmer or a newbie programmer, as a modern developer you should have knowledge that how to integrate the modern technologies and other technical stuff in your work to make a good product or software.

In this article, we will discuss some important technical skills you should have as a developer or if you are planning to become a good developer.

**1. Data Structures and Algorithms:**This skill is the topmost priority by most of the companies to check the problem solving and coding skill. You can become a good software developer if you know how data can be organized and how it can be used to solve a real life problem. As a developer or a computer science student or a self taught programmer, you should put serious effort to learn Data Structures and Algorithms(e.g. array, linked list, tree). Data Structures and Algorithms are the heart of programming. Initially most of the developers do not realize its importance but when you will start your career in software development, you will find your code is either taking too much time or taking too much space. There you will realize the importance of organizing the data using right data structure and algorithm to solve a specific problem in less time and less space.

**2. Programming Languages:** In order to become a good programmer you must have command on at least one programming language in depth. When it comes to deciding which programming language you should choose, so it depends on your area of interest and in which language you love to solve the problems or you are comfortable with.

**3. Source Control:**Source control helps the developer in managing and storing their code. Today most of the organization is looking for a developer who is proficient in version control and collaborate with other team members. So if your plan is to become a good software developer you need to be comfortable with version control concepts and tools like Git (preferred by 70-80% organization), Mercurial, SVN, etc. Go through this article to learn more about source control [Git and Github](https://www.geeksforgeeks.org/ultimate-guide-git-github/)

**4. Text Editors:** It doesn’t matter you are a beginner, intermediate or expert level programmer. Every programmer’s programming journey start from text editors. It is an essential tool of programmers daily life. Not only programmers even non-programmers also use text editors for their own purpose. There are a lot of text editors available like *Notepad++, Sublime Text, Atom, Brackets, Visual code,*, etc. Every programmer especially beginner should spend some time in a learning text editor and some keyboard shortcut to becoming a smart and productive developer.

**5. IDEs (Integrated Development Environment):** IDEs allows you to write, modify, compile, run and debug your code. When it comes to choosing the best development environment, different programmers have different choice for different purpose or language they are working on. Every programmer should know how to use IDEs to write, compile, run and debug their code. Using an IDE speed up their work and there are so many IDEs available for developers. For C, C++ and C# programmers most recommended choice is *Visual Studio* or *Code::Blocks*. For Python developers, PyCharm, Spyder or Jupyter notebook is also getting popularity. For Java *Eclipse*, *NetBeans*and *IntelliJ IDEA* is the best choice for developers.

**6. Databases:**An essential skill for developers is understanding of working with databases. Developers should know all kind of operation like how to store records, create, insert, update, delete, etc. Creating any kind of application and software is impossible for any organization without the database. When a developer work on any serious business project they also need to take care of security issue and managing the complete organization record with proper backup and that’s the reason every organization expects a developer comfortable in working with databases and managing the complete records securely. There is no doubt that [SQL](https://www.geeksforgeeks.org/sql-tutorial/) is the most popular classic database among developers. If you want to become a good developer you need to be good in writing at least basic SQL queries.*Oracle, PostgreSQL, MongoDB, Cassandra, Redis*, etc. are the databases you can work with.

**7. Operating System:** A serious software developer should know the fundamentals and mechanism of the operating system. When a developer work on a project they deal with so many issues related with an operating system like memory usage, communicating with another machine, running a program very slow, tools conflicting issues, blocking issues, etc. When a developer writes code on one machine and that doesn’t work on another machine then it can create a serious issue during the production level. So its good to have good knowledge of process and mechanism of the operating system you are going to work with. Most of the programmers prefer working on Linux, Windows machine or Mac. Android and iOS is the best example of mobile and tablet operating system.

**8. Networking Basics:**In *70-80%* cases developers work or application is based on the client-server model, where the request goes through the network to a server and client can be based anywhere in the world to access the application. Understanding of basic networking is important for developers to develop and support an application. If the architecture wouldn’t be designed properly it can create HTTP request issue over the network. In a client-server architecture, a user or a client POST request via the internet which is received by the server and after processing data request the response is sent back to the client. Today in most of the application like web-based, online games, business automation or cloud computing client-server architecture is used. So we can not deny this fact that fundamentals and basics of networking are also an essential skill for the developer before they enter into programming.

**9. Basics of Testing:**Before releasing software in the market there are so many test cases a software or an application has to pass. Testing is an important step to find out all kind of bug and to check if the software is ready to hand over to the customer or not. There are so many test methods but a developer should have knowledge of three important testing methods. First one is [**Unit Testing**](https://www.geeksforgeeks.org/unit-testing-software-testing/) in which each and every individual module or class is tested properly. There are so many unit testing frameworks available like NUnit for C#, JUnit for Java, Embunit for C or C++. Another testing is [**Integration Testing**](https://www.geeksforgeeks.org/software-engineering-integration-testing/) where a developer has to test the interaction between different blocks or modules. This test helps to exclude the incorrect processing of data. The last testing is [**System Testing**](https://www.geeksforgeeks.org/system-testing/) where the test is conducted on complete and integrated software. It falls under the black box testing technique where it doesn’t require internal knowledge of the code.

**10. Cross-Platform Software:**When a product is designed or developed it is expected to run on multiple platforms like Windows, Mac, Linux, etc. In a simple language cross platform allows you to write code once and that is shared across different platforms. It is important for a developer to make a product which can be accepted by the maximum number of systems because today most of the users are switching to the mobile devices or smartphones to use the product or services. The number of customers can be increased eventually when the software is adapted to the maximum number of devices. If you are a beginner or experienced level programmer, you should have knowledge about working with cross-platform software for better career opportunity and growth in software development.

**11. Encryption and Cryptography:** When it comes to making a web application or software which uses sensitive information of a user, it is important for a developer to implement a secure and encrypted key to prevent all kind of attacks. Security of users sensitive information, preventing a site from hacking is the major concern for every organization when it comes to making a product or software. It is expected from a developer to know about how encryption algorithms work, how authentication works and how cryptography methods work. There are mainly two methods used for data encryption. These are ***public key*** and ***private key***. A key is established with public key encryption algorithms such as RSA or ECDSA and its depend on developer which one he/she wants to choose to keep in mind about the overall security of the network.

**12. SDLC (Software development life cycle):** SDLC is a step by step systematic process to develop a software ensuring its quality and correctness. If you are planning to become a software developer you should have knowledge about the proper life cycle of software from requirement analysis to maintenance of the product. There are mainly 7 phases in SDLC.

1. Requirement gathering and analysis
2. Feasibility study
3. Design
4. Implementation and Coding
5. Testing
6. Deployment
7. Maintenance

**13. Microsoft Excel:** A developer has to do nothing with Microsoft Excel when it comes to doing coding or implementing software, but still, it is an essential skill for them because it helps to track progress, data analysis, data quality check, maintenance of data and definitely for project planning. We can not underestimate the use of Excel. Its use is much more than spreadsheet which helps the developer to make their work easy and it also helps to speed up their work using its functions like searching, sorting, filtering or for any kind of mathematical operation. Microsoft Excel is also used widely to insert bulk data in databases. This tool is an essential part of a project for a junior level developer to a manager. So its good to spend some time learning the basic functions of Excel.

**Practical 6:**

**Aim: To study various software development standards**

**General Standards**

All software development projects, including maintenance projects, must follow these standards. Objectives for application development include:

* Clear definition of purpose
* Simplicity of use
* Ruggedness (difficult to misuse, kind to errors encountered)
* Delivered on time and when needed
* Reliability
* Efficiency (fast enough for the purpose it was created)
* Minimum development cost
* Conform to standards
* Clear, accurate and precise user documentation
* Clear, accurate and precise technical documentation.

All production systems must have designated Owners and Custodians for the critical information they process in order to identify requirements and verify the final deliverables with signoff. There must be a separation between the production, development and test environments. This will ensure that security is rigorously maintained for the production system, while the development and test environments can maximize productivity with fewer security restrictions. Where these distinctions have been established, development and test staff must not be permitted to have access to production systems. All applications are reviewed at predetermined checkpoints of the SDLC by the Application Architect or their designate

1. **REQUIREMENT STANDARDS**
   1. The project team must gather business and system requirements Activities include:

Review existing systems/process, Describe data/system/process, Identify problem, areas/opportunities, Identify user needs/wants, Conduct interviews, Develop Solution, Identify manual and automated processes, Draw conceptual flow, Identify follow on projects/phases, Identify inputs (functional description), Data entry screens, Inputs from outside sources…

* 1. The project team must establish and document business/application requirements In order to achieve the above the following items should be followed.
* Document requirements
* Document assumptions
* Document outstanding issues
* Estimate data storage requirements
* Identify legislative/contractual/security/privacy/access requirements
* Document reporting requirements
* Review training requirements
* Conduct initial walkthrough
* Obtain sign-off and approval
  1. The project team must produce the following minimum set of documentation as part of this phase:

Enterprise Architecture Documentation: refer to the Enterprise Architecture Checklist for Change Initiative as the authoritative source for preparing the following Enterprise Architecture artifacts

**2. Analysis Standard**

2.1 The project team must analyse business and system requirements

**Activities include**: Analyzing system flow, data model, Name and define fields in data dictionary, data normalization, physical data model, screens, screen navigation, data entry screens, inquiry screens, help screens, online documentation, Analyse reports, Forms, Report distribution system, User generated reports, Identify files, file formats, edit criteria, record volume, record/file purge criteria, Analyse existing system modifications, controls, program to program controls, backup/recovery procedure, system patterns, security, privacy, application security, equipment needed, impacts to other organizations, Access Services/LAN Management/Help Desk, Data Centre (Production System Support/Operations), analyse resource implications, Storage requirements (Tapes, Online transactions (I/O),PADS), User connections, CPU…

2.2 The specific intended use of the system to be developed must be analyzed to specify system requirements.

The system requirements specification should describe: functions and capabilities of the system; business, organizational and user requirements; safety, security, information, privacy, interface, operations, and maintenance requirements; design constraints and qualification requirements. The system requirements specification must be documented.

2.3 The project team must establish and document application requirements, including the quality characteristics specifications,

2.4 The project team must evaluate the application requirements considering the criteria listed below. The results of the evaluations must be documented.

2.5 The project team must conduct review(s).

* Appropriate Enterprise Architecture Checkpoint review
* Internal peer reviews and walkthroughs

**3. Design Standard**

3.1 The project team must perform activities/tasks related to design.

**Activities include**: Design system flow, Develop data model, Create physical data model, Design screens, screen navigation, data entry screens, inquiry screens, help screens, online documentation, Design reports, Forms, Report distribution system, User generated reports, Design Patterns, Existing system modifications, Conduct design walkthrough, Conceptual flow/procedures, Screen design, & Process Implementation…

3.2 A top-level architecture of the system must be established. The architecture should identify items of hardware, application/software, and manual-operations. It should be ensured that all the system requirements are allocated among the items. Hardware configuration items, application/software configuration items, and manual operations should be subsequently identified from these items. The system architecture and the system requirements allocated to the items must be documented.

3.3 The system architecture and the requirements for the application must be evaluated considering the criteria listed below. The results of the evaluations must be documented.

* Traceability to the system requirements;
* Consistency with the system requirements;
* Appropriateness of design standards and methods used;
* Feasibility of the application module/component/services fulfilling requirements;
* Feasibility of operation and maintenance. Design - Application

3.4 The project team must transform the requirements for the application into an architecture that describes its top-level structure and identifies the application components. It must be ensured that all the requirements for the application are allocated to its application components and further refined to facilitate detailed design. The architecture of the application must be documented.

3.5 The project team must develop and document a top-level design for the interfaces external to the application and between the application components of the application.

3.6 The project team must develop and document a top-level design for the database.

3.7 The project team should develop and document preliminary versions of user documentation.

3.8 The project team must define and document preliminary test requirements and the schedule for Application Integration.

3.9 The project team must evaluate the architecture of the application and the interface and database designs considering the criteria listed below. The results of the evaluations must be documented.

**4 Coding Standard:**

4.1 The project team must review and analyse architecture/design documentation and construct/code to design specifications.

4.2 The project team must perform the activities/tasks related to construction or coding. GO-ITS 54 Status: Approved Version 1.0 - 19 - Activities/Tasks include: Construct the application, components, services including Data entry screens, Inquiry screens, Menu screens, Online help screens, Batch programs, Changes to existing programs, Conversion programs, Build and load files/tables, Build job streams, Develop test cases, Unit test programs, Develop secure code, Develop application/software documentation, Users guide, Turnover documentation, Training materials, Conduct initial turnover walkthrough, Schedule turnover dates.

4.3 The project team must develop and document the following:

* Each application unit and database
* Test procedures and data for testing each application unit and database.

4.4 The project team must test each application unit and database ensuring that it satisfies its requirements. The test results must be documented.

4.5 The project team must evaluate application code and test results considering the criteria listed below. The results of the evaluations must be documented.

* Traceability to the requirements and design of the application;
* External consistency with the requirements and design of the application;
* Internal consistency between unit requirements;
* Test coverage of units;
* Appropriateness of coding methods and standards used;
* Feasibility of application integration and testing;
* Feasibility of operation and maintenance.

4.6 The project team must develop an integration plan to integrate the application units and application components into the application. The plan should include test requirements, procedures, data, responsibilities, and schedule. The plan must be documented.

4.7 The project team must integrate the application units and application components and test as the aggregates are developed in accordance with the integration plan. It should be ensured that each aggregate satisfies the requirements of the application and that the application is integrated at the conclusion of the integration activity. The integration and test results must be documented.

4.8 The project team must update the required documentation as necessary.

4.9 The project team must develop and document, for each requirement of the application, a set of tests, test cases (inputs, outputs, test criteria), and test procedures for conducting Application Testing. The project team must ensure that the integrated application is ready for Application Testing.

**5 Testing Standard**

5.1 The project team must conduct testing in accordance with the requirements for the application. It must be ensured that the implementation of each application requirement is tested for compliance. The testing results must be documented Activities/Tasks include: Application test, Software test, Run parallel test, Document results, User acceptance test, Security test, Develop Test procedures, Document results, Issue production readiness recommendation.

5.2 The project team must update the required documentation as necessary.

5.3 The project team must evaluate the design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations must be documented.

* Test coverage of the requirements of the application;
* Conformance to expected results;
* Feasibility of system integration and testing, if conducted;
* Feasibility of operation and maintenance.

5.4 The project team must support the user testing. The results of the testing must be documented. 5.5 Upon successful completion of the test, the project team should: a) Update and prepare the deliverable application product for System Integration, System Testing, Application Installation, or Application Acceptance Support as applicable. b) Establish a baseline for the design and code of the application.

**6 Implementation Standards**

6.1 The project team must develop a plan to install the application product in the target environment as designated. The resources and information necessary to install the application product should be determined and be available. The project team should assist the acquirer with the set-up activities. Where the installed application product is replacing an existing system, the project team should support any parallel running activities that are required. The installation plan must be documented.

6.2 The project team should install the application product in accordance with the installation plan. It should be ensured that the application code and databases initialize, execute, and terminate. The installation events and results must be documented.

6.3 The project team should support acceptance review and testing of the application product. Acceptance review and testing should consider the results of the Reviews, Audits, Application Testing, Security Testing, and System Testing. The results of the acceptance review and testing must be documented

6.4 The project team must conduct review(s).

* Appropriate Enterprise Architecture Checkpoint review
* Internal peer reviews and walkthroughs

6.5 The project team must complete and deliver the application product.

6.6 The project team must provide initial and continuing training and support as outlined in the implementation plan.

**Practical 7:**

**Aim: To study a deployment diagram and draw a deployment diagram for bank management system and vehicle registration system**

**Deployment Diagrams :**

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

**Purpose of Deployment Diagrams:**

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as −

* Visualize the hardware topology of a system.
* Describe the hardware components used to deploy software components.
* Describe the runtime processing nodes.

**How to Draw a Deployment Diagram?**

Deployment diagram represents the deployment view of a system. It is related to the component diagram because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application. An efficient deployment diagram is very important as it controls the following parameters −

* Performance
* Scalability
* Maintainability
* Portability

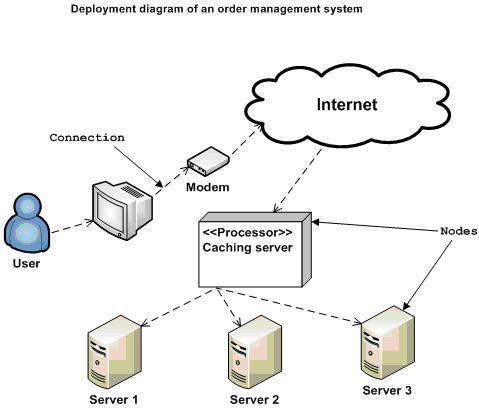
Before drawing a deployment diagram, the following artifacts should be identified −

* Nodes
* Relationships among nodes

Following is a sample deployment diagram to provide an idea of the deployment view of order management system. Here, we have shown nodes as – Monitor, Modem, Caching server, Server

The application is assumed to be a web-based application, which is deployed in a clustered environment using server 1, server 2, and server 3. The user connects to the application using the Internet. The control flows from the caching server to the clustered environment.

The following deployment diagram has been drawn considering all the points mentioned above.



**Where to Use Deployment Diagrams?**

Deployment diagrams are mainly used by system engineers. These diagrams are used to describe the physical components (hardware), their distribution, and association. Software applications are developed to model complex business processes.

Deployment diagrams can be used −

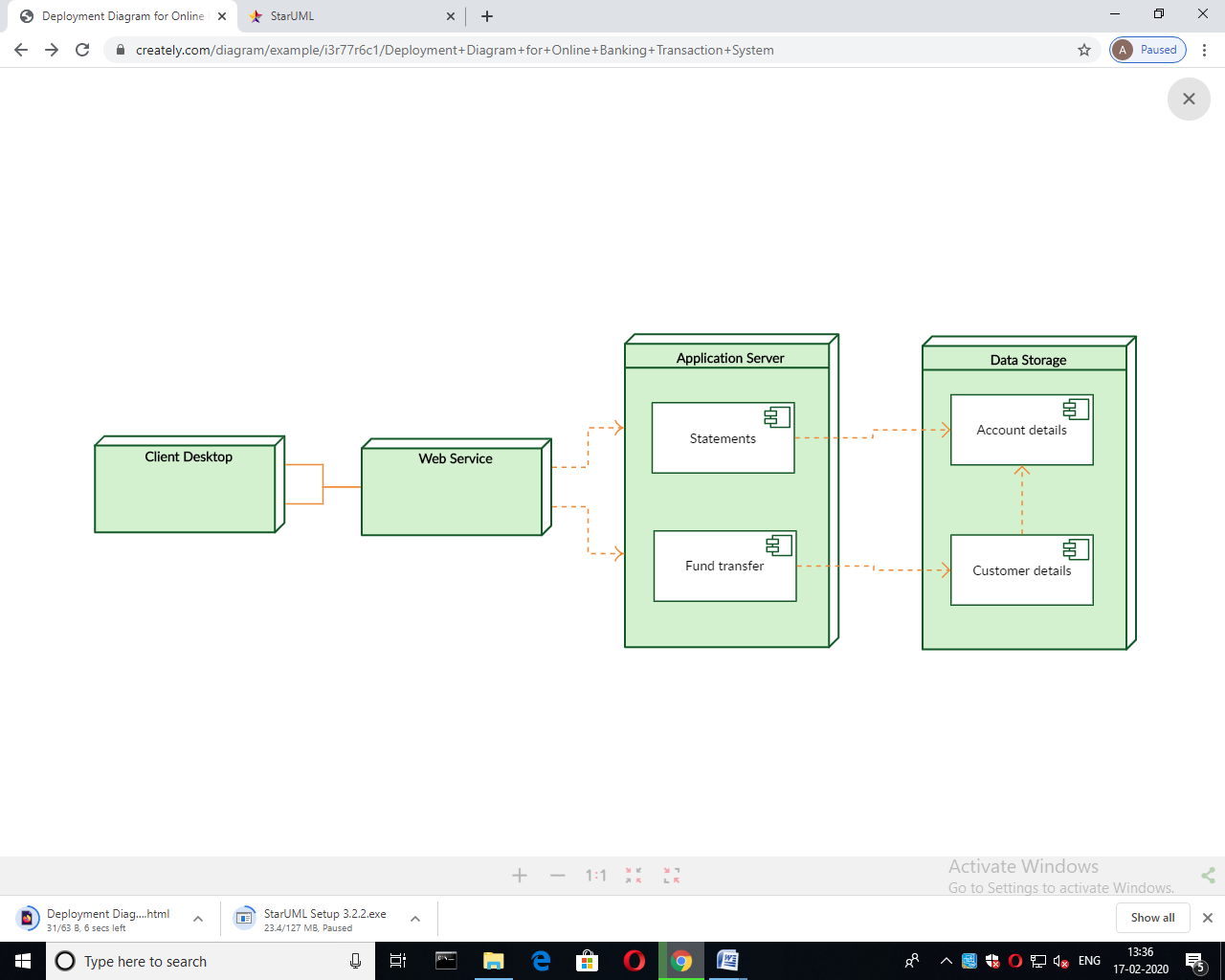
* To model the hardware topology of a system.
* To model the embedded system.
* To model the hardware details for a client/server system.
* To model the hardware details of a distributed application.
* For Forward and Reverse engineering.

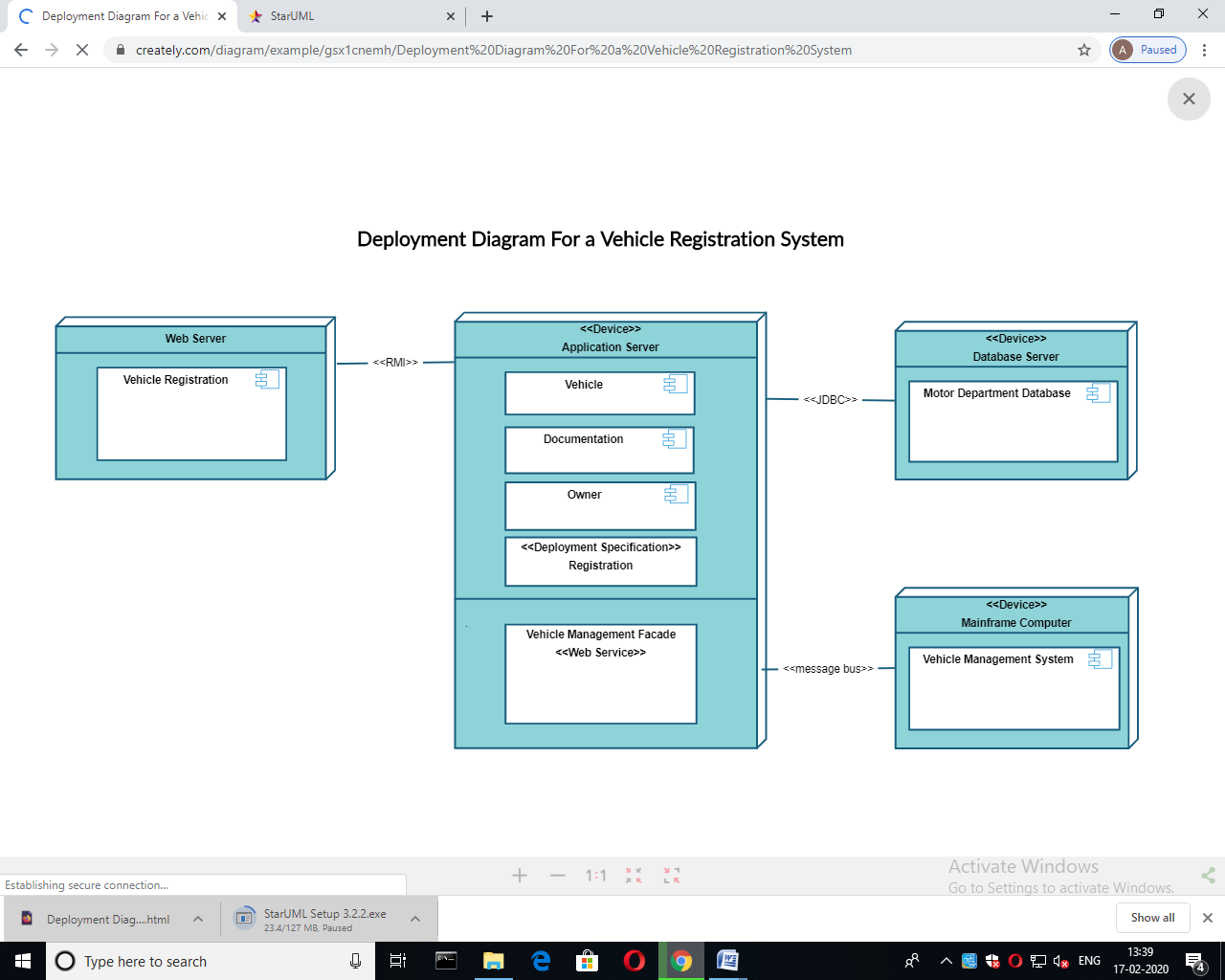
## Deployment diagram elements

A variety of shapes make up deployment diagrams. This list offers an overview of the basic elements you may encounter, and you can see most of these items illustrated in the image below.

* **Artifact:** A product developed by the software, symbolized by a rectangle with the name and the word “artifact” enclosed by double arrows.
* **Association**: A line that indicates a message or other type of communication between nodes.
* **Component:** A rectangle with two tabs that indicates a software element.
* **Dependency:** A dashed line that ends in an arrow, which indicates that one node or component is dependent on another.
* **Interface:** A circle that indicates a contractual relationship. Those objects that realize the interface must complete some sort of obligation.
* **Node:** A hardware or software object, shown by a three-dimensional box.
* **Node as container:** A node that contains another node inside of it—such as in the example below, where the nodes contain components.
* **Stereotype:** A device contained within the node, presented at the top of the node, with the name bracketed by double arrows.

**Deployment diagram for bank management system:**

**Deployment diagram for vehicle registration system:**



**Practical 8:**

**Aim: To study a component diagram and draw component diagram for passport automation system**

**Component diagrams:**

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

**Purpose of Component Diagrams**

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

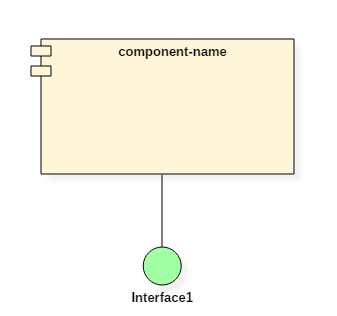
The purpose of the component diagram can be summarized as −

* Visualize the components of a system.
* Construct executables by using forward and reverse engineering.
* Describe the organization and relationships of the components.

**How to Draw a Component Diagram?**

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries; etc the purpose of this diagram is different. Component diagrams are used during the implementation phase of an application. However, it is prepared well in advance to visualize the implementation details. Initially, the system is designed using different UML diagrams and then when the artifacts are ready, component diagrams are used to get an idea of the implementation.

**Component in UML is represented as follows**

[](https://www.guru99.com/images/1/052919_0717_ComponentDi3.png)

Before drawing a component diagram, the following artifacts are to be identified clearly −

* Files used in the system.
* Libraries and other artifacts relevant to the application.
* Relationships among the artifacts.

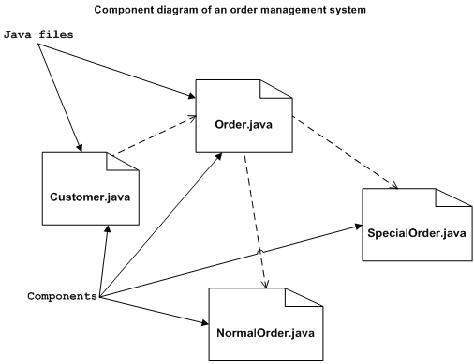
After identifying the artifacts, the following points need to be kept in mind.

* Use a meaningful name to identify the component for which the diagram is to be drawn.
* Prepare a mental layout before producing the using tools.
* Use notes for clarifying important points.

Following is a component diagram for order management system. Here, the artifacts are files. The diagram shows the files in the application and their relationships. In actual, the component diagram also contains dlls, libraries, folders, etc.

In the following diagram, four files are identified and their relationships are produced. Component diagram cannot be matched directly with other UML diagrams discussed so far as it is drawn for completely different purpose.

The following component diagram has been drawn considering all the points mentioned above.



**Where to Use Component Diagrams?**

We have already described that component diagrams are used to visualize the static implementation view of a system. Component diagrams are special type of UML diagrams used for different purposes. These diagrams show the physical components of a system. To clarify it, we can say that component diagrams describe the organization of the components in a system.

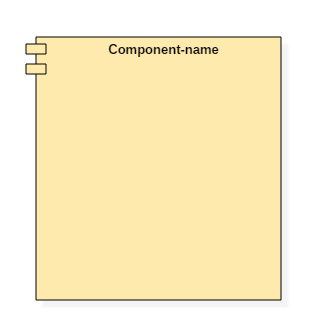
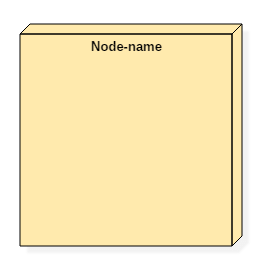
As we have already discussed, those components are libraries, files, executables, etc. Before implementing the application, these components are to be organized. This component organization is also designed separately as a part of project execution.

Component diagrams can be used to −

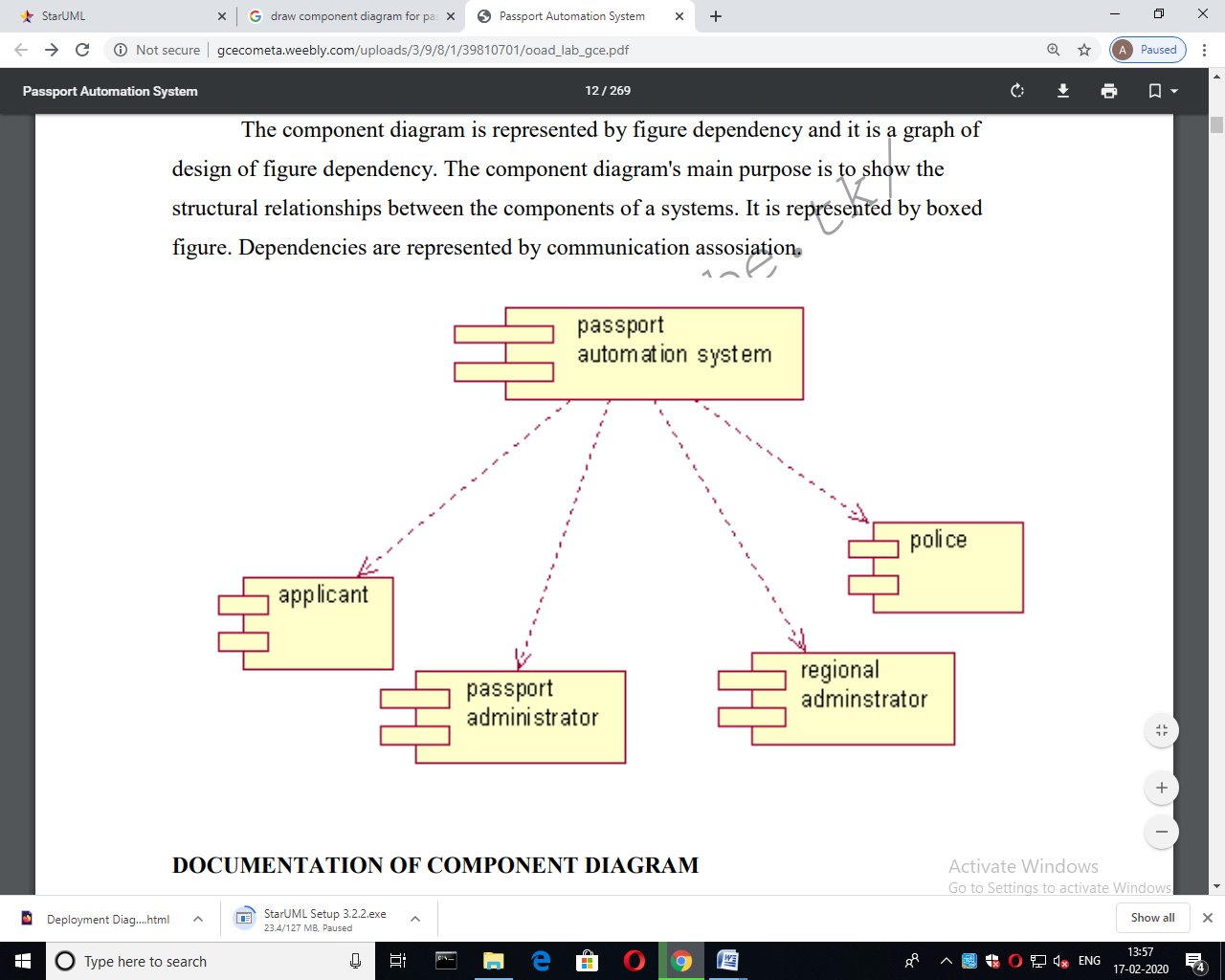
* Model the components of a system.
* Model the database schema.
* Model the executables of an application.
* Model the system's source code.

**Component diagram Notations**

* 1. A component 2. A node

[](https://www.guru99.com/images/1/052919_0717_ComponentDi1.png)[](https://www.guru99.com/images/1/052919_0717_ComponentDi2.png)

**Component diagram for Passport automation system:**



**Practical: 9**

**AIM: To study communication diagram and draw a communication diagram of online shopping**

**What is a communication diagram?**

A communication diagram offers the same information as a [sequence diagram](https://www.lucidchart.com/pages/uml-sequence-diagram), but while a sequence diagram emphasizes the time and order of events, a communication diagram emphasizes the messages exchanged between objects in an application. Sequence diagrams can fall short of offering the "big picture.” This is where communication diagrams come in and offer that broader perspective within a process.

**Basic components of a communication diagram**

Communication diagrams offer benefits similar to sequence diagrams, but they will offer a better understanding of how components communicate and interact with each other rather than solely emphasizing the sequence of events. Try drawing a sequence diagram to:

* Model the logic of a sophisticated procedure, function, or operation.
* Identify how commands are sent and received between objects or components of a process.
* Visualize the consequences of specific interactions between various components in a process.
* Plan and understand the detailed functionality of an existing or future scenario.

**Purpose of Communication Diagram**

* Model message passing between objects or roles that deliver the functionalities of use cases and operations
* Model mechanisms within the architectural design of the system
* Capture interactions that show the passed messages between objects and roles within the collaboration scenario
* Model alternative scenarios within use cases or operations that involve the collaboration of different objects and interactions
* Support the identification of objects (hence classes), and their attributes (parameters of message) and operations (messages) that participate in use cases

## Communication Diagram vs Sequence Diagram

The communication diagram and the sequence diagram are similar. They're semantically equivalent, that is, the present the same information, and you can turn a communication to a sequence diagram and vice versa. The main distinction between them is that the communication diagram arranged elements according to space; the sequence diagram is according to time.

**Objects**: participating in a collaboration come in two flavours: supplier and client.

* **Supplier objects** are the objects that supply the method that is being called, and therefore receive the message.
* **Client objects** call methods on supplier objects, and therefore send messages.

**Links**

* The connecting lines drawn between objects in a communication diagram are links.
* These links are what set communication diagrams apart from sequence diagrams. They enable you to see the relationships between objects.
* Each link represents a relationship between objects and symbolizes the ability of objects to send messages to each other.
* If an object sends messages to itself, the link carrying these messages is represented as a loop icon. This loop can be seen on both the UI object and the Transaction object.

**Messages** : In communication diagrams are shown as arrows pointing from the Client object to the Supplier object. Typically, messages represent a client invoking an operation on a supplier object. They can be modelled along with the objects in the following manner:

* Message icons have one or more messages associated with them.
* Messages are composed of message text prefixed by a sequence number.
* This sequence number indicates the time-ordering of the message.

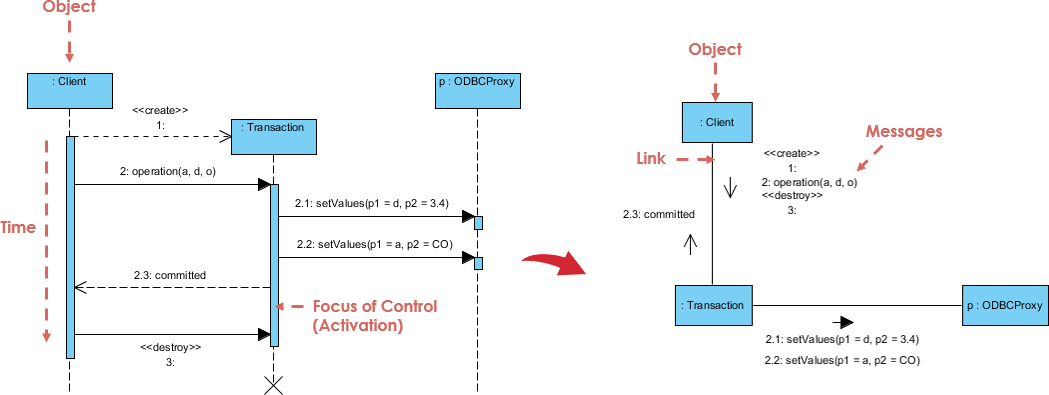
**Understanding the Numbering of Messages in Communication Diagram**

* The first message in a communication diagram is always numbered 1, the second is 2, and so on.
* You can indicate that a message is nested under a parent message by adding a decimal point and incremental digits to the parent's sequence number.

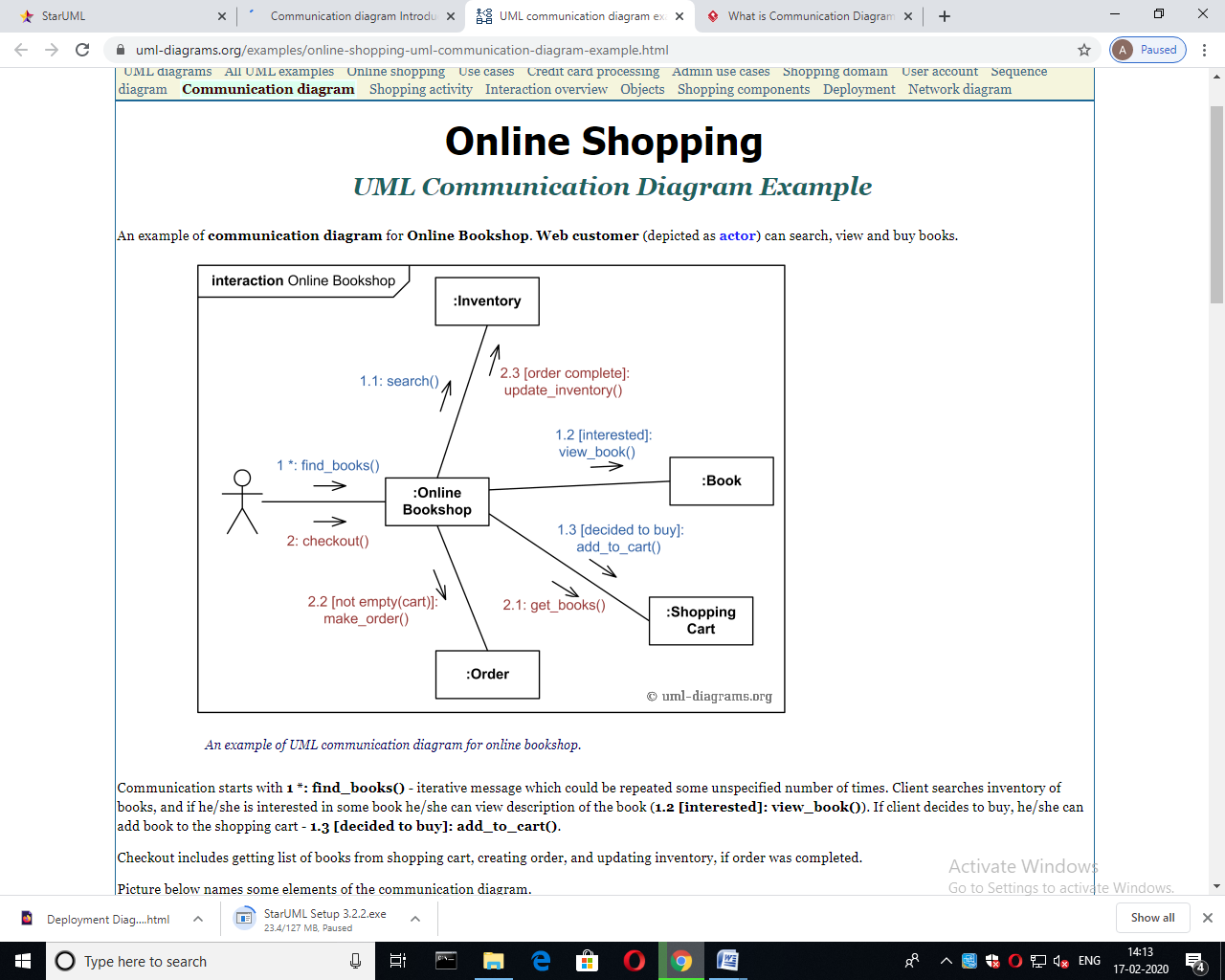
**For example:**

Based on the example above, the "Cal Amt Can Borrow" message is the first nested message under "Enquire Borrower" and is given the sequence number 1.1. The second nested message under "Enquire Borrower" is "Display Invalid Msg", so it's given a sequence number of 1.2.

## Example - From Sequence Diagram to Communication Diagram



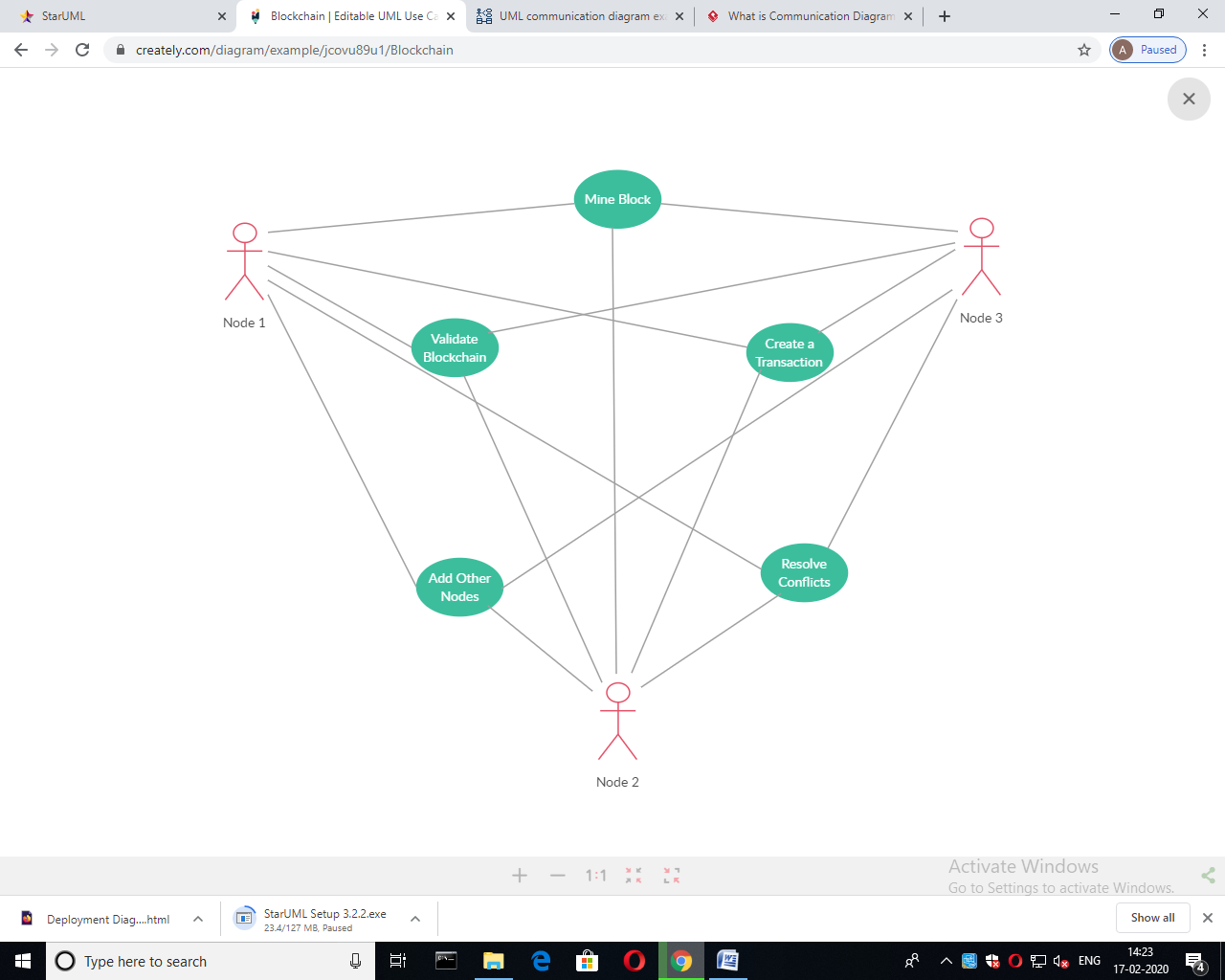
**Communication diagram for online shopping system**



**Practical: 10**

**AIM: Draw use case and sequence diagram of block chain technology.**

**UML diagram for blockchain technology:**

****

**Sequence diagram for block chain technology**

