

## **Unit 2 Key Characteristics enterprise systems & architecture**

### **1. Distributivity**

Distributivity has different views. One view of distributed systems, i.e. distributed software applications, is already discussed shortly in the previous section.

Another views of distributivity, especially for the manufacturing system or enterprise are related to the distributed control of the (manufacturing) enterprise, based on multi-agent system model, and to the spatial (or geographical) distribution of the (manufacturing) enterprise functions and physical components.

A distributed enterprise, such as a retailer, bank, or hospital, has several locations or branches, often worldwide. For many such organizations, the headquarters are protected against cyberattacks, but the branches are not, leaving back doors open for cybercriminals to gain access to sensitive data.

#### **What Are Distributed Enterprises?**

The modern interpretation of a distributed enterprise is a central office space with localized IT staff and networking facilities. This definition also includes several other remote or branch sites that have few or no IT staff. Remote departments and branches usually have similar needs to centralized offices but with varying degrees of scalability.

A distributed enterprise that has a considerable remote hybrid office or workforce generally reproduces the central office's platform stack on a considerably smaller scale. As businesses evolve, prosper, and reach new markets, it's essential to introduce a presence where customers are located. Distributed enterprise strategies evaluate cost and complexity and approximate them against growth potential.

#### **Distributed Enterprise Strategy**

Distributed enterprise strategies should include the following:

#### **Enterprise-Grade Security**

Enterprise-grade security is the process a business uses to protect its informational assets against cybersecurity attacks, integrity infringement, or availability issues. This includes data, servers, storage, and workstations.

#### **SD-WAN**

SD-WAN (Software-Defined Wide Area Network) is a software system defined by a wide area that uses network technology to communicate over the internet between different locations using encryption.

### **Cloud Deployment**

Cloud deployment allows businesses to deploy applications through varying hosting models and use SaaS (Software As A Service) or IaaS (Infrastructure As A Service) platforms that utilize the cloud.

### **Generalized Management System**

A GMS (Generalized Management System) is a manual that charts operations processes including planning, implementing, and monitoring functions that affect service delivery.

### **Why Are Distributed Enterprises Important?**

Distributed enterprise and geographically dispersed teams are important because of the flexibility and balance associated with remote work. Businesses that adopt hybrid working models garner employees who are on average 30% more productive than their in-office counterparts.

Distributed enterprise solutions are also cheaper to maintain and offer a wealth of hiring options and benefits for organizations. One of the most important advantages is workplace engagement which builds positive associations and connections to work-based environments and cultures.

According to Gartner, a dispersed workforce is the most effective way to build a 21st-century business because it allows for greater worker autonomy, increased productivity, and better engagement.

### **How To Secure Distributed Enterprises?**

Distributed enterprises can have multiple branches in different locations around the world. Hospitals, banks, retailers are all examples of distributed enterprises. Whilst headquarters and base of operations premises can often be protected against cyberattacks, individual branches are left vulnerable. Cybercriminals can easily gain back door privileges to sensitive data and cause significant harm.

Virtual private networks (VPNs) are traditionally used as a means to connect remote locations and rapidly deploy new applications. VPNs are generally seen as

an inexpensive and secure means of communication. However, the changing digital landscape is becoming progressively decentralized and VPNs aren't able to deliver any real benefits.

Organizations of all sizes are now investing in the highest levels of security across multiple locations. These are the five security essentials that distributed enterprises should take note of:

### **Next-generation Firewalls**

Most organizations have firewalls but they lack strength and can't provide 'true' security. Next-generation firewalls (NGFW) enhance the capabilities of traditional firewalls with additional security capacity. This includes early detection, alleviation, and malware reporting.

### **Centralized Management**

Applying centralized management to firewalls allows core administrators to manage the entirety of the safety structure from one location. This means that security policies for numerous branches can be managed without on-site security aids.

### **Unlimited Scalability**

It's important to add scalability as new locations are explored. Wide area network (WAN) acceleration is used to efficiently run security software which can be recognized automatically by firewalls and self-configured.

### **Application Control**

Application priorities are imperative to centralized security management solutions. Ensuring that priorities are set guarantees the uninhibited bandwidth on the network.

### **High-Performance DPI SSL Inspection**

Internet powerhouses such as Google and Facebook now encrypt all of the data traffic that shifts between data centers and end-users. 29.1% of Internet traffic is encrypted by SSL, with year-on-year increases expected. This means that firewalls need additional capabilities to inspect and decrypt enterprise traffic.

### **Top Distributed Enterprise Issues**

Although distributed enterprise can be a useful asset to businesses looking to innovate the architecture of work, it also has several disadvantages that are important to note:

### **Difficult Troubleshooting**

Distribution across multiple servers makes troubleshooting and diagnostics increasingly complicated.

### **Limited Software Support**

Distributed systems have less inherent support and suffer because of it.

### **High Network Infrastructure Costs**

Setting up network infrastructure across numerous locations can be expensive. Transmission, high load, and loss of information also need to be factored into the equation.

### **Security Issues**

Open systems are vulnerable to attack and increase the risk of security breaches. Data security and sharing pose the most risks in distributed enterprise systems.

### **Why Is Digital Transformation Important In Distributed Enterprise?**

The demand for personalized, superior, and distinguishable experiences that are delivering innovation and technological breakthroughs is greater than ever. Businesses are constantly inventing new ways to transform the lives of their customers in the digital age.

Exploration and experimentation are fundamental components of digital transformation and go hand in hand with the emergence of companies from multiple industries and cultures. The next phase of digital transformation shifts enterprise storage, networking, and computing from centralized locations to distributed or decentralized edge locations.

Data is moving to a more decentralized approach that emphasizes local storage through distributed edge locations. When discussing distributed edge from an enterprise perspective, it offers huge potential for better performance, reduced network costs, and faster deployment.

## 2. Managed redundancy

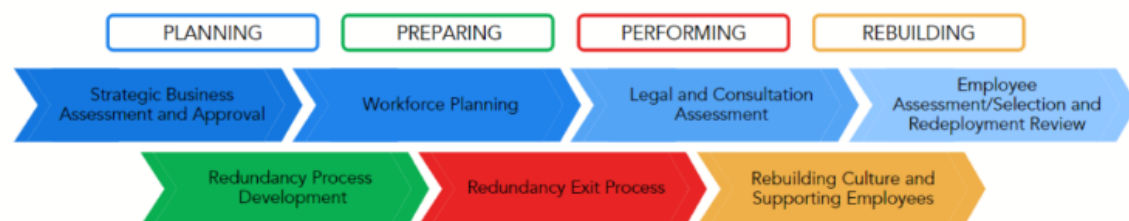
### What is a redundancy process?

Generally, a redundancy or layoff is an involuntary termination of employment where a job or role is no longer required. Redundancies are usually caused by factors, such as changes in economic conditions, business efficiencies or technological advancements.

A redundancy process, on the other hand, is the systematic approach to reducing roles as an organisation undergoes restructuring. Taking a people-centred approach to redundancy management can result in better outcomes for you, your business and your workforce.

What is the best redundancy process to follow?

*The People Shift's Pragmatic Redundancy Process*



*Caption: Overview of The People Shift's pragmatic redundancy program.*

### Strategic Business Assessment and Approval

1. Inspect the current state of your business / organisation.
2. Develop a map of your business goals and strategy to get a clear vantage point.
3. When working for a larger organisation, investigate if you are eligible to run the redundancy. If you belong to a smaller business, understand the impacts on your budget.
4. In some cases, running the redundancy requires other corporate members.

### Workforce Planning

- Analyse your talent against your current goals.
- Consider the future and the skills and experience you have to meet your future strategy.
- Identify the gaps in your workforce and assess the external market.
- Determine the solution that suits your needs and understand if redundancies are required.

### **Legal and Consultation Assessment**

- Assess legal requirements and ensure you know your obligations under the law.
- Assess Industrial Relations/Employee Relations Risks and Union consultation or collaboration requirements under any enterprise agreements.
- Ensure that the people involved sign confidentiality statements and understand the implications of breaching these.

### **Employee Assessment/Selection and Redeployment Review**

- Determine if you even need to run a selection process. If your redundancy program involves reducing the number of people performing the same roles, then the answer will be yes.
- Carry out a sound assessment process—include evidence requirements, relevant training for leaders and assessment procedures.

### **Redundancy Process Development**

- Develop a redundancy and exit process, considering the most dignified approach for impacted employees.
- Develop templates, tools, scripts and training that may be required to ensure consistency and professionalism of your program.
- Using scripts to support all redundancy meetings will ensure conversations stay factual, objective and free of bias.

### **Redundancy Exit Process**

Distribute communications to all relevant stakeholders in line with your communication plan. Consider the following:

- Appropriate Dates and Venue
- Communication to Employees
- Structure of Meetings
- Employee Support
- Exit Checklist
- Government Notification

### **Rebuilding Culture and Supporting Employees**

- Communicate to the workforce about the end of the process
- Acknowledge and honour those that left and create opportunities for people to talk
- Monitor individuals to ensure mental health is intact—be aware of survival guilt

- Coordinate employee assistance program (EAP) or mental health support. Initiate engagement activities over the following weeks

### **Rethink**

As Jamie Getgood said, redundancy is a profound moment in time wherein you could dramatically change people's lives. Make sure you impact your employees' lives in a positive light, even as they exit your business.

## **5 Steps to Manage the Redundancy Process Successfully**

### **1. Preparation of the Process**

Assess your business' financial situation and identify which areas of your operations are surplus to requirement. If redundancies can be avoided that would be the best for all parties, but if not start preparing the necessary documentation and move on to the next step.

### **2. Fair Selection**

After identifying the areas of change and selecting the person/people for redundancy it is vital to ensure the process is fair. You may have a pool of people to select from, but all decisions must be objective and strictly made based on the business. This may be based on the employee's capability, skills, competence, experience, attitude, disciplinary records, etc.

### **3. Be Clear About the Situation & the Role**

Redundancies happen when a role at a company is no longer required. When a position is deemed no longer necessary the employee is usually laid off. Clearly explain to the employee that the position is being made redundant and that it is not a reflection of the employee. Employees will be more accepting of the fact that it is not them but rather the businesses' need to restructure which has caused the displacement. It can be a difficult time for employees and emotions can run high. A clear explanation and the necessary information should be provided to the employee regarding the situation. Be prepared to answer any questions they may ask to make it easier for them.

### **4. Show Empathy**

This is a distressing time for everyone. Losing regular income can cause fear, especially if the individual has a family, a mortgage, and outstanding bills. Be prepared for the emotional responses the employee may exhibit and train yourself to manage the anger, shock, and resistance you may encounter.

Show empathy for the displaced employees and give them a sense that this was not a simple decision for your business either. The last thing you want as a business leader is for an angry employee to tell friends, family, and associates about the lack of compassion from management and how cold-hearted the people are at the company. This will damage your company's reputation and hurt your chances of hiring future talent.

When possible, it's best to offer choices. This may be in the form of alternative employment.

### **5. Offer Additional Support for Departing Employees**

Redundancies can shake a business to its core. The impacts of the changes won't just experience instantaneously but can last for years down the line. Restructuring the team will inevitably affect the chemistry of the company, you could be breaking strong bonds and affecting the synergy between employees, which is essential for a successful business.

The whole process will be watched by every person associated with the company. Implementing an outplacement service and partnering with specialists is a great way to help both the displaced employee and also the employer as it helps uphold good morale in the remaining staff.

Retrenched employees will be connected to new opportunities, allowed to re-evaluate their careers, and feel positive about their future. It also shows that your company truly values its employees which is crucial to brand protection. This will help your company negate the long-term financial impacts of brand erosion, offering a perfect win-win solution!

## **3. Exception Processing**

Exception processing is an important feature in any robust enterprise application. A lot of developers misunderstand and not apply the underlying techniques. Therefore, the end user is confused and does not know what to do when an exception occurs. It's responsible of the developer to try to handle the more exceptions as he can in order for the data to be consistent as well as to free the end user to make decision related to the software application.

## **4. Collaboration**

Enterprise collaboration is defined as the system which is used for communication and collaboration in organizations to improve information gathering and knowledge sharing among employees, despite where they are located. The goal of enterprise collaboration is to make it easier for employees to work together, accomplish their tasks efficiently, and improve the overall business output.

It can be achieved through social networking tools, dedicated collaboration platforms, the company intranet, or even the internet. Successful enterprise collaboration in organizations leads to higher productivity, lesser chances of miscommunication, increased employee engagement, and better informed employees.

**There are two types of enterprise collaboration:**



## **Internal enterprise collaboration**

Internal enterprise collaboration happens within the workspace of the organization and it helps in bringing together all the employees. As the main benchmark for enterprise collaboration, it is the key to efficiency in organizations. It allows employees to stay connected and informed by ensuring seamless communication.

## **External enterprise collaboration**

External enterprise collaboration happens mostly on social media platforms among employees, consumers, and brands. It can also take place between companies that need to collaborate with each other in order to work together on products or services.

## **Benefits of enterprise collaboration**

### ➤ Reduced email

While email doesn't have to permanently go away, the fact remains that email wasn't built for today's digital workplaces. We now have an increasing number of remote employees who need real-time collaboration in order to work together seamlessly. An enterprise collaboration tool offers real-time communication which can reduce back and forth emails while increasing productivity.

### ➤ Improved transparency

Enterprise collaboration tools allow employees to easily track and manage projects, drive personal accountability and establish higher levels of trust which directly leads to more transparency within the organization.

### ➤ Better teamwork

Real-time communication in teams results in more effective and efficient collaboration. Moreover, centralized file storage also cuts down on errors and confusion among team members. When all the employees collaborate through a single platform, it also helps support and improves cross-team collaboration.

### ➤ Fewer apps, more efficiency

On average, employees are forced to switch between up to 35 applications every minute — that is more than 1100 times in a day. Constant application switching can eat up a lot of time and eventually affect the overall work efficiency.

An enterprise collaboration platform offers a single platform for all of the organization's communication and collaboration needs. It improves efficiency by replacing unnecessary applications with overlapping features.

## **Major hurdles with enterprise collaboration**

### **1. No companywide collaboration strategy**

For many organizations, collaboration isn't considered as a companywide cultural value and as a result, there is no collaboration strategy developed.

But the absence of a well-defined collaboration strategy can make it difficult to find the right enterprise collaboration tools that align with your main business requirements and collaboration challenges. In the end, you may even end up with an enterprise collaboration platform that introduces more problems than it solves for your employees.

As a result, it is important for organizations to first gather input and feedback from employees to create a collaboration strategy first and then find the right enterprise collaboration platforms that fit their needs.

### **2. Misinterpreting collaboration**

If organizations don't have a clear way of defining what collaboration means to them, it can often get misinterpreted and create a big hurdle in implementing organization-wide enterprise collaboration.

For instance, it is common to confuse cooperation with collaboration. Cooperation occurs when people work closely together and share their resources, ideas, and expertise to support each other's individual goals. Collaboration, on the other hand, is about working together in order to achieve shared goals.

### **3. Lack of support for collaboration**

It's good to recognize and reward individual performance, but creating recognition programs for team performance is just as important to create a supportive culture where employees feel encouraged to communicate and share their expertise with their co-workers in order to achieve shared goals.

After all, if your employees look at collaboration with their coworkers as the time that they could have otherwise spent working on their individual tasks, it can be challenging to motivate them to collaborate effectively as a team, even with the most effective enterprise collaboration platforms.

### **4. Problems in adopting new technology**

Nobody wants to break something that isn't broken. Introducing a new enterprise collaboration platform when employees believe they have been managing 'just fine' with the existing collaboration tools can be a huge challenge.

As a result, business leaders should first start by initiating discussions with employees to explain the need to shift to an enterprise collaboration platform, how it can help and why it is being implemented.

## **How successful enterprise collaboration happens?**

### **Have clear goals and processes**

Your organization needs to define clearly why enterprise collaboration matters and why you are introducing it in the first place. Additionally, you will need a plan and process to implement enterprise collaboration across different verticals of the company and actually stick to that plan.

### **Choosing the right technology**

Find the right technology and tools that align with your specific goals, processes, and company objectives. Of course, this doesn't mean you should implement all the latest technologies that your competitors are using. It's about carefully identifying and recognizing tools that can help employees collaborate in your organization and easily integrate with your existing technology infrastructure.

### **Educate employees**

It's crucial to make your employees understand the importance of collaboration, knowledge sharing, and enterprise collaboration tools. The employee buy-in factor alone can decide on the success or failure of your enterprise collaboration strategy. As a result, employees need to know how enterprise collaboration platforms can help in their everyday work, boost productivity, and save time so that they are all on board with the new changes that are about to come. Moreover, there is also a need to train employees to get accustomed to the new tools and get comfortable with them.

### **Continuously track and measure results**

Just introducing an enterprise collaboration strategy and some tools does not directly mean that you have successfully achieved enterprise collaboration. You have to constantly track and measure the results to see how they are faring. Ideally, after introducing enterprise collaboration, the previous gaps in communication should get smaller, information sharing should grow in teams, and collaboration should improve across the organization.

### **Achieving enterprise collaboration with a digital workplace**

There is no denying that building and introducing enterprise collaboration brings some huge challenges of its own. Setting and aligning business processes that connect employees on many teams requires careful and constant attention. Not to mention, introducing a new tool to your already growing technology stack may end up adding more problems in your organization.

Instead, you can introduce a unified digital workplace platform to achieve enterprise collaboration, solve business problems, improve knowledge sharing and increase overall operational productivity.

A digital workplace platform like Kissflow gives employees access to a complete set of digital tools that they need to manage their work and collaborate effectively. With features like public and private communication channels, seamless project

management, process management, workflow management, polls, and announcements, Kissflow makes achieving enterprise collaboration amazingly easy.

## **5. Data Transformation**

Enterprise Data Transformation: While the phrase “data transformation” is used in several different contexts, within organizational development it refers to the process of creating a data-driven culture throughout and across the organization.

### **Benefits and challenges of data transformation**

Transforming data yields several benefits:

- Data is transformed to make it better-organized. Transformed data may be easier for both humans and computers to use.
- Properly formatted and validated data improves data quality and protects applications from potential landmines such as null values, unexpected duplicates, incorrect indexing, and incompatible formats.
- Data transformation facilitates compatibility between applications, systems, and types of data. Data used for multiple purposes may need to be transformed in different ways.

However, there are challenges to transforming data effectively:

- Data transformation can be expensive. The cost is dependent on the specific infrastructure, software, and tools used to process data. Expenses may include those related to licensing, computing resources, and hiring necessary personnel.
- Data transformation processes can be resource-intensive. Performing transformations in an on-premises data warehouse after loading, or transforming data before feeding it into applications, can create a computational burden that slows down other operations. If you use a cloud-based data warehouse, you can do the transformations after loading because the platform can scale up to meet demand.
- Lack of expertise and carelessness can introduce problems during transformation. Data analysts without appropriate subject matter expertise are less likely to notice typos or incorrect data because they are less familiar with the range of accurate and permissible values. For example, someone working on medical data who is unfamiliar with relevant terms might fail to flag disease names that should be mapped to a singular value or notice mis-spellings.

- Enterprises can perform transformations that don't suit their needs. A business might change information to a specific format for one application only to then revert the information back to its prior format for a different application.

### **How to transform data**

Data transformation can increase the efficiency of analytic and business processes and enable better data-driven decision-making. The first phase of data transformations should include things like data type conversion and flattening of hierarchical data. These operations shape data to increase compatibility with analytics systems. Data analysts and data scientists can implement further transformations additively as necessary as individual layers of processing. Each layer of processing should be designed to perform a specific set of tasks that meet a known business or technical requirement.

Data transformation serves many functions within the data analytics stack.

### **Extraction and parsing**

In the modern ELT process, data ingestion begins with extracting information from a data source, followed by copying the data to its destination. Initial transformations are focused on shaping the format and structure of data to ensure its compatibility with both the destination system and the data already there. Parsing fields out of comma-delimited log data for loading to a relational database is an example of this type of data transformation.

### **Translation and mapping**

Some of the most basic data transformations involve the mapping and translation of data. For example, a column containing integers representing error codes can be mapped to the relevant error descriptions, making that column easier to understand and more useful for display in a customer-facing application.

Translation converts data from formats used in one system to formats appropriate for a different system. Even after parsing, web data might arrive in the form of hierarchical JSON or XML files, but need to be translated into row and column data for inclusion in a relational database.

### **Filtering, aggregation, and summarization**

Data transformation is often concerned with whittling data down and making it more manageable. Data may be consolidated by filtering out unnecessary fields, columns, and records. Omitted data might include numerical indexes in data intended for graphs and dashboards or records from business regions that aren't of interest in a particular study.

Data might also be aggregated or summarized. by, for instance, transforming a time series of customer transactions to hourly or daily sales counts.

BI tools can do this filtering and aggregation, but it can be more efficient to do the transformations before a reporting tool accesses the data.

### **Enrichment and imputation**

Data from different sources can be merged to create denormalized, enriched information. A customer's transactions can be rolled up into a grand total and added into a customer information table for quicker reference or for use by customer analytics systems. Long or freeform fields may be split into multiple columns, and missing values can be imputed or corrupted data replaced as a result of these kinds of transformations.

### **Indexing and ordering**

Data can be transformed so that it's ordered logically or to suit a data storage scheme. In relational database management systems, for example, creating indexes can improve performance or improve the management of relationships between different tables.

### **Anonymization and encryption**

Data containing personally identifiable information, or other information that could compromise privacy or security, should be anonymized before propagation. Encryption of private data is a requirement in many industries, and systems can perform encryption at multiple levels, from individual database cells to entire records or fields.

### **Modelling, typecasting, formatting, and renaming**

Finally, a whole set of transformations can reshape data without changing content. This includes casting and converting data types for compatibility, adjusting dates and times with offsets and format localization, and renaming schemas, tables, and columns for clarity.

### **Refining the data transformation process**

Before your enterprise can run analytics, and even before you transform the data, you must replicate it to a data warehouse architected for analytics. Most organizations today choose a cloud data warehouse, allowing them to take full advantage of ELT. Stitch can load all of your data to your preferred data warehouse in a raw state, ready for transformation.

## **6. What Is Batch Processing?**

Batch processing is the processing of transactions in a group or batch. No user interaction is required once batch processing is underway. This differentiates batch processing from transaction processing, which involves processing transactions one at a time and requires user interaction.

While batch processing can be carried out at any time, it's particularly suited to end-of-cycle processing, such as for processing a bank's reports at the end of a day or generating monthly or biweekly payrolls.

### **Why is batch processing important?**

Organizations use batch processing because it requires minimal human interaction and makes repetitive tasks more efficient to run. You can set up batches of jobs composed of

millions of records to be worked through together when compute power is most readily available, putting less stress on your systems. Modern batch processing also requires minimal human supervision or management. If there is an issue, the system automatically notifies the concerned team to solve it. Managers take a hands-off approach, trusting their batch processing software to do its job. More benefits of batch processing follow.

### **What are some use cases of batch processing systems?**

**There are multiple use cases of batch processing systems. Key examples follow.**

#### **Financial services**

Financial services organizations, from agile financial technologies to legacy enterprises, have been using batch processing in areas such as high performance computing for risk management, end-of-day transaction processing, and fraud surveillance. They use batch processing to minimize human error, increase speed and accuracy, and reduce costs with automation.

#### **Software as a service**

Enterprises that deliver software as a service (SaaS) applications often run into issues when it comes to scalability. Using batch processing, you can scale customer demand while automating job scheduling. Creating containerized application environments to scale demand for high-volume processing is a project that can take months or even years to complete, but batch processing systems help you achieve the same result in a much shorter timeframe.

#### **Medical research**

Analysis of large amounts of data—or big data—is a common requirement in the field of research. You can apply batch processing in data analytics applications such as computational chemistry, clinical modelling, molecular dynamics, and genomic sequencing testing and analysis. For example, scientists use batch processing to capture better data to begin drug design and gain a deeper understanding of the role of a particular biochemical process.

#### **Digital media**

Media and entertainment enterprises require highly scalable batch processing systems to automatically process data—such as files, graphics, and visual effects—for high-resolution video content. You can use batch processing to accelerate content creation, dynamically scale media packaging, and automate media workload.

### **How does batch processing work?**

While batch processing applications vary depending on the type of task that needs to be done, the basics of any batch job remain the same. The user can run batch jobs by specifying the following details:

- Name of the person submitting the job
- Batch processes or programs that need to run
- System location of the data input
- System location for processed data output
- Time, or batch window, when the batch job should be run

The user also specifies the batch size, or the number of work units that the system needs to process in one complete batch operation. Some examples of batch size include:

- Number of batch file lines to read and store in the database
- Number of messages to read and process from a queue
- Number of transactions to sort and send to the next application

During the batch window, the batch processing system uses the batch size information to allocate the resources needed to run the batch job efficiently. Modern systems can run hundreds of thousands of batch jobs on premises or in the cloud.

### **Dependencies**

Batch job tasks can run sequentially or simultaneously. Sequences can differ depending on whether an earlier task is completed successfully. Examples of dependencies include a customer making an order in an online store or paying a bill. A dependency can also be set up to initiate a job processing cycle.

### **Cron commands**

A cron command is a batch job that runs regularly. You can set up recurrence patterns for batch jobs—for example, setting up a job to invoice for subscriptions at the end of every month.

### **How can you monitor batch processing?**

While batch processing systems work with minimal input from personnel, they still need some oversight. To monitor batch processes, you can set up alerts—or exceptions—that are sent when the batch job succeeds, fails, or has finished running.

### **Monitors**

Monitors in batch processes look for abnormalities, such as a job taking longer than it should to complete. In this instance, it would stop the next job from beginning and inform the relevant staff of the exception.

### **Post-processing analysis**

You can view the history of a batch job after it has been processed. Most batch processes include log files that record messages while the job was running.





## Advantages of Batch Processing

### Faster and Lower Cost

Operational costs such as labor and equipment are cut with batch processing because it cuts the need for human oversight physical hardware like computers. And because batch processing is designed to be quick, efficient and error-free, personnel can focus on other duties.

### Offline Features

Unlike others, batch processing systems work anywhere, any time. That means they continue to work outside regular business hours. They can also work in the background in an offline setting, so even during down periods, they'll still work without putting a dent in the organization's daily routine.

### Hands-Off Approach

As mentioned above, having a batch processing system in place gives managers and other key personnel time to do their own jobs without having to spend time supervising batches. Alerts are sent when problems arise. This allows workers a hands-off approach to batch processing.

## Disadvantages of Batch Processing

Business owners may want to consider a few of the pitfalls of batch processing before putting such a system into place.

### Deployment and Training

Like many technologies, training is required to manage batch processing systems. Managers will need to learn what triggers a batch, how to schedule processing, and what exception notifications mean, among other things.

### Debugging

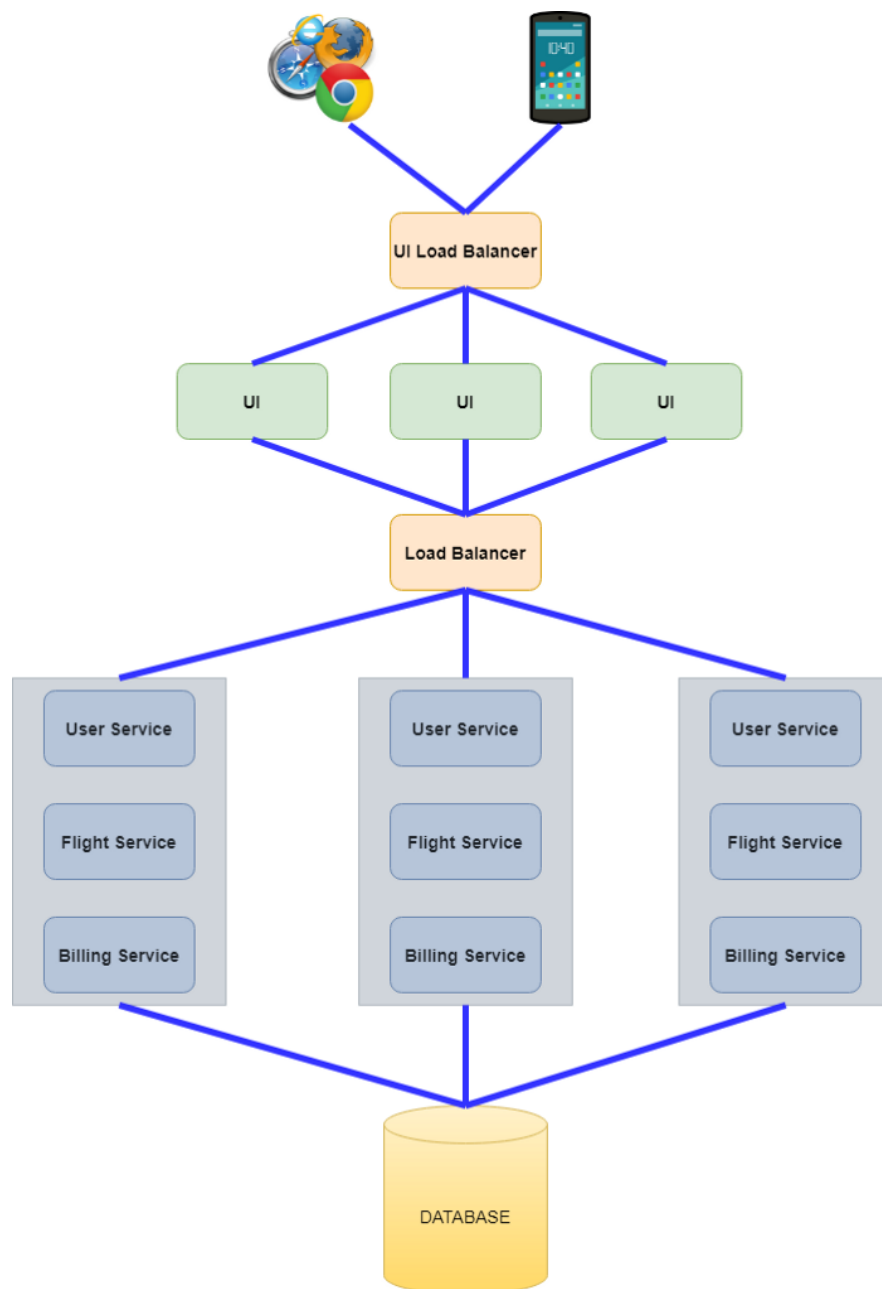
The systems are often complex, requiring someone on staff to be familiar with the program. Otherwise, companies or organizations may need to hire an information technology specialist for help.

## Cost

Batch processing infrastructure can be an expensive upfront investment. For some businesses, the costs may not seem feasible.

## 7. Monolithic

Monolithic Architecture is a traditional way of building applications. This software architecture principle has both advantages and disadvantages. On the one hand, it can bring delight. On the other hand, it can cause disappointment. Let's review its place in software architecture.



Typical Monolithic Architecture is presented in the picture above. The key characteristic of it is the fact that everything ( *User Service*, *Flight Service*, and *Billing Service*) is located inside in a single deliverable (a jar file in case of Spring Boot). These services are tightly coupled ( despite the fact that they have totally different functions) and use the same database.

The picture shows the most commonly used approach nowadays. Actually, in a monolithic architecture *UI* part can be packaged with backed service. As a result, *UI*, *User Service*, *Flight Service*, and *Billing Service* are located together in a single deliverable.

For scaling purposes, multiple identical deliverable units are deployed. *The Load Balancer* is responsible for routing requests to deployed monolithic services.

### Advantages of Monolithic Architecture

- **Simplicity of development.** The monolithic approach is a standard way of building applications. No additional knowledge is required. All source code is located in one place which can be quickly understood.
- **Simplicity of debugging.** The debugging process is simple because all code is located in one place. You can easily follow the flow of a request and find an issue.
- **Simplicity of testing.** You test only one service without any dependencies. Everything is usually clear.
- **Simplicity of deployment.** Only one deployment unit (e.g. *jar* file) should be deployed. There are no dependencies. In cases when *UI* is packaged with backend code you do not have any breaking changes. Everything exists and changes in one place.
- **Simplicity of application evolution.** Basically, the application does not have any limitation from a business logic perspective. If you need some data for new feature, it is already there.
- Cross-cutting concerns and customizations are used only once. You should take care of cross-cutting concerns only once. For instance, security, logging, exception handling, monitoring, choosing and setting up tomcat parameters, setup of data source connection pool, etc.
- **Simplicity in onboarding** new team members. The source code is located in one place. New team members can easily debug some functional flow and to get familiar with the application.
- **Low cost** in the early stages of the application. All source code is located in one place, packaged in a single deployment unit, and deployed. What can be easier? There is no overhead neither in infrastructure cost nor development cost.

Because of these advantages, monolithic architecture is usually used in the early stages of application development. The reasons for that are the next ones:

- The main function of the application is to be profitable. As a result, it is important to quickly implement some POC (Proof of Concept) solutions to verify the application in the real world. Also, it is important to bring customers to the system. Improvements can be implemented in the future.
- The requirements are usually unclear at the early stages of development. It is hard to create meaningful architecture when the requirements are unclear. Real customers can define the business needs after some of the functionality already works.

The problems with Monolithic Architecture start to appear when the application becomes successful. The reason for this is very simple: the growth of the application. Usually, after some period of time, the Monolithic Application changes into another one because of the next reasons.

### Disadvantages of Monolithic Architecture

- **Slow speed of development.** The simplest disadvantage relates to CI/CD pipeline. Imagine the monolith that contains a lot of services. Each service in this monolith is covered with tests that are executed for each Pull Request. Even for a small change in a source code, you should wait a lot of time (e.g. 1 hour) for your pipeline to succeed. And what happens when the pipeline fails for some reason? You wait again. All services are located in a single place. The size of the team is big. What happens when your colleague merges their changes? You rebase/merge and wait again.
- **High code coupling.** Of course, you can keep a clear service structure inside your repository. However, as practice shows, eventually, you will end up with a spaghetti code in at least a few places. As a result, the system becomes harder to understand especially for new team members.
- **Code ownership cannot be used.** The system is growing. The logical step is to split responsibilities between several teams. E.g. one team can work on *Flight Service*, another — for *Billing Service*. However, there are no boundaries between those services. One team can affect another.
- **Testing becomes harder.** Even a small change can negatively affect the system. As a result, the regression for full monolithic service is required.
- **Performance issues.** Potentially, you can scale the whole monolithic service in cases of performance issues. But what to do with the database? The single database is used for all services. You can start to optimize your database queries or use read replicas. However, there is a limit to this type of optimization.

- The cost of infrastructure. In cases of performance issues, you should scale the whole monolithic service. It brings additional costs for application operability.
- **Lack of flexibility.** Using Monolithic Architecture you are tight to the technologies that are used inside your monolith. You cannot use other tools even if they are better for the problem at hand.
- **Problems with deployment.** Even a small change requires the redeployment of the whole monolith.

## 8. Client Server

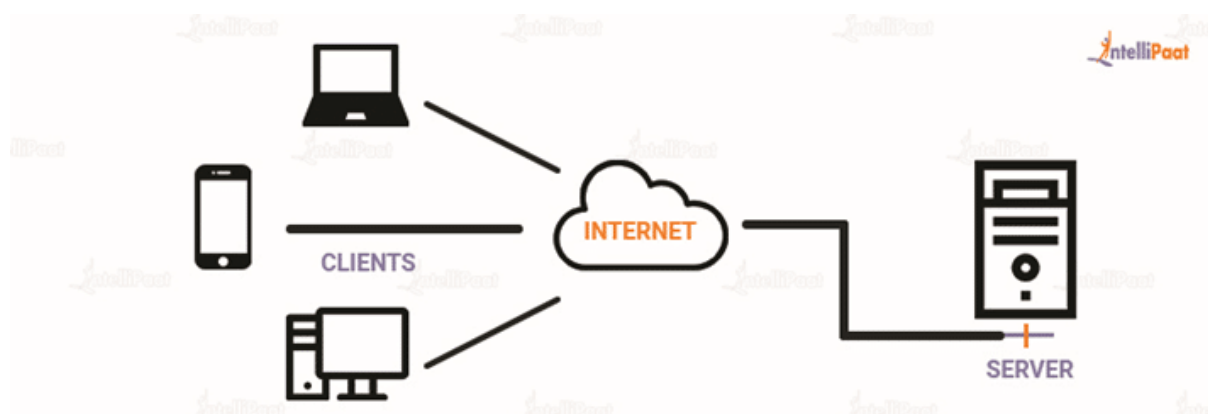
### What is client server architecture?

Before we explain client server architecture and you start reading words such as servers, service, network, data, and files, and start feeling overwhelmed with jargon, let us first understand about this architecture in layperson's terms.

The notion of client-server architecture can be understood by the analogy of ordering a pizza for delivery. You call the store to order a pizza and someone picks up the call, takes your order, and then delivers it. Simple, right? Yes, this analogy pretty much answers the fundamental principle of client server architecture.

Simply put, two factors are involved:

- A server is the one who provides requested services.
- Clients are the ones who request services.



Client server architecture is a computing model in which the server hosts, delivers, and manages most of the resources and services requested by the client. It is also known as the networking computing model or client server network as all requests and services are

delivered over a network. The client-server architecture or model has other systems connected over a network where resources are shared among the different computers.

Typically, client server architecture is arranged in a way that clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. Such a model is especially beneficial when the clients and server perform routine tasks. For example, in hospital data processing, a client computer can be busy running an application program for entering patient information, meanwhile the server computer can be running another program to fetch and manage the database in which the information is permanently stored.

#### Client server architecture example

Here are some of the client server model architecture examples from our daily life. Hope it helps you to understand the concept better.

- **Mail servers:** Email servers are used for sending and receiving emails. There are different software that allow email handling.
- **File servers:** File servers act as a centralized location for files. One of the daily life examples to understand this is the files that we store in Google Docs. The cloud services for Microsoft Office and Google Docs can be accessed from your devices; the files that you save from your computer, can be accessed from your phone. So, the centrally stored files can be accessed by multiple users.
- **Web servers:** Web servers are high-performance computers that host different websites. The server site data is requested by the client through high-speed internet.

#### Components of client server architecture:

Essentially, three components are required to make client server architecture work. The three components are workstations, servers, and networking devices. Let us, now, discuss them in detail:

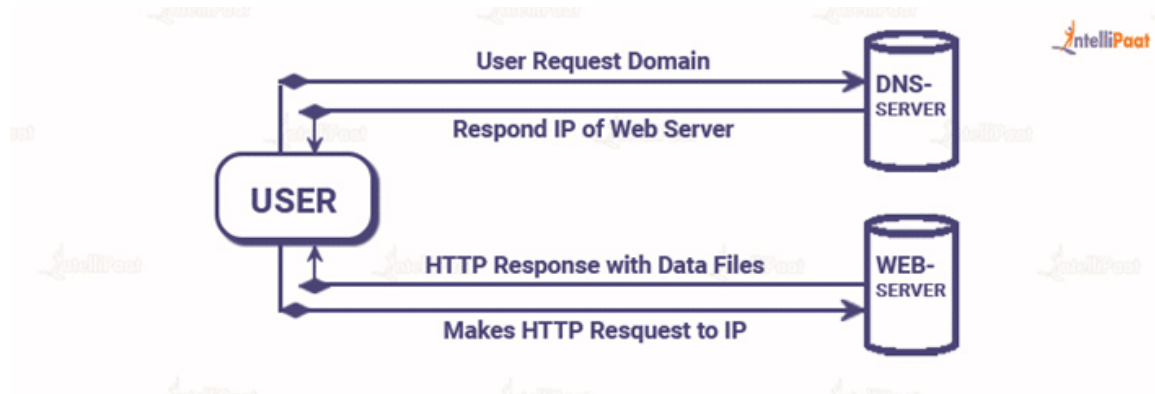
- **Workstations:** Workstations are also called client computers. Workstations work as subordinates to servers and send them requests to access shared files and databases. A server requests information from the workstation and performs several functions as a central repository of files, programs, databases, and management policies. Workstations are governed by server-defined policies.
- **Servers:** Servers are defined as fast processing devices that act as centralized repositories of network files, programs, databases, and policies. Servers have huge storage space and robust memory to deal with multiple requests, approaching simultaneously from various workstations. Servers can perform many roles, such as mail server, database server, file server, and domain controller, in client server architecture at the same time.
- **Networking devices:** Now that we know about the roles that workstations and servers play, let us learn about what connects them, networking devices. Networking

devices are a medium that connects workstations and servers in client server architecture. Many networking devices are used to perform various operations across the network. For example, a hub is used for connecting a server to various workstations. Repeaters are used to effectively transfer data between two devices. Bridges are used to isolate network segmentation.

How does client server architecture work?

So far, we have understood that client server architecture is made up of two elements, one that provides services and the other that consumes those services.

To get a clearer picture of the process, let us learn how the browser interacts with the server.



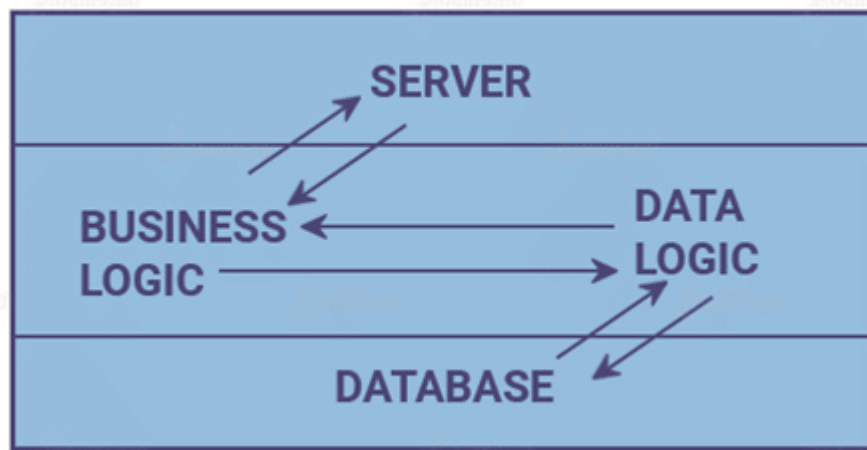
Please read the following steps for a better understanding of the process:

- The user enters the uniform resource locator (URL) of the website or file and the browser sends a request to the domain name system (DNS) server.
- The DNS server looks for the address of the web server and the DNS server responds with the IP address of the web server.
- After the DNS server responds, the browser sends over an HTTP or HTTPS request to the web server's IP, which was provided by the DNS server.
- The server then sends over the necessary files of the website.
- Finally, the browser renders the files and the website is displayed.

### Types of client server architecture

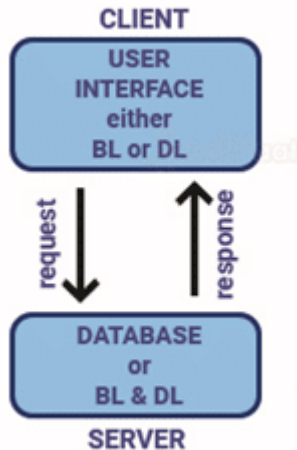
The functionality of client server architecture is in various tiers.

### 1-tier architecture



In this category of client server architecture, the architecture contains all kinds of settings, such as configuration setting and marketing logic, on a single device. While the diversity of services offered by 1-tier architecture makes it one of the reliable sources, handling such an architecture is difficult. This is primarily due to the data variance. It often results in replication of work. 1-tier architecture consists of several layers, such as presentation layer, business layer, and data layer, that are combined with the help of a unique software package. The data present in this layer is usually stored in local systems or on a shared drive.

### 2-tier architecture

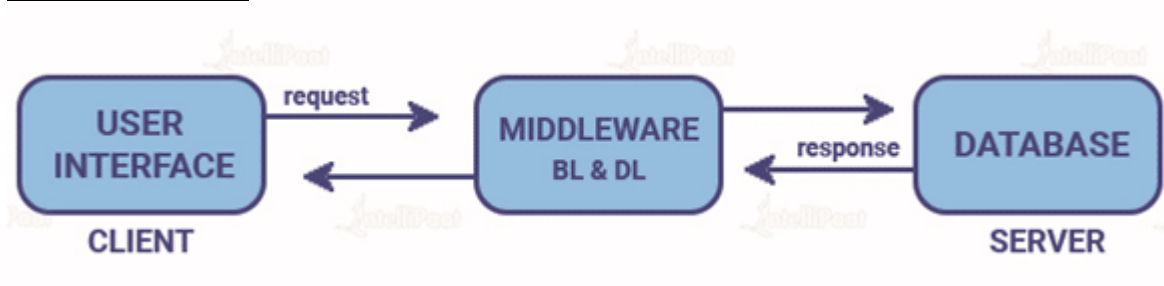


his architecture has the best environment. In this architecture, the user interface is stored on the client's side and the database is stored on the server, while database logic and business logic is maintained either on the client's side or on the server's side.

The 2-tier architecture is faster in comparison to the 1-tier architecture; this is because the 2-tier architecture does not have any intermediary between the client and the server. It is often utilized to avoid confusion between clients. One of the popular examples of 2-tier architecture is the online ticket reservation system.



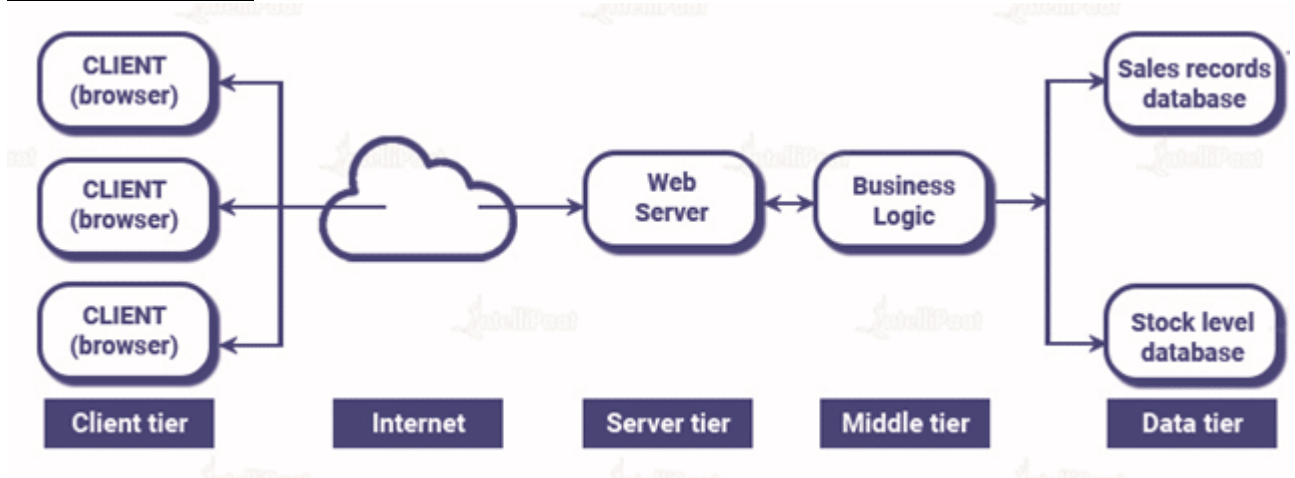
### 3-tier architecture



Unlike 2-tier architecture that has no intermediary, in 3-tier client server architecture, a middleware lies between the client and the server. If the client places a request to fetch specific information from the server, the request will first be received by the middleware. It will then be dispatched to the server for further actions. The same pattern will be followed when the server sends a response to the client. The framework of 3-tier architecture is categorized into three main layers, presentation layer, application layer, and database tier.

All three layers are controlled at different ends. While the presentation layer is controlled at the client's device, the middleware and the server handle the application layer and the database tier respectively. Due to the presence of a third layer that provides data control, 3-tier architecture is more secure, has invisible database structure, and provides data integrity.

### N-tier architecture



N-tier architecture is also called multi-tier architecture. It is the scaled form of the other three types of architecture. This architecture has a provision for locating each function as an isolated layer that includes presentation, application processing, and management of data functionalities.

### **Advantages and disadvantages of client-server architecture**

The advantages and disadvantages of client-server architecture are mentioned below:

Advantages	Disadvantages
i. The centralized network has complete leverage to control the processes and activities.	i. If the primary server goes down, the entire architecture is disrupted.
ii. All devices in the network can be controlled centrally.	ii. It is expensive to operate because of the cost of heavy hardware and software tools.
iii. Users have the authority to access any file, residing in the central storage, at any time.	iii. This architecture requires particular OSs related to networking.
iv. It provides a good user interface, easy file finding procedure, and management system for organizing files.	iv. Too many users at once can cause the problem of traffic congestion.
v. Easy sharing of resources across various platforms is possible.	v. It requires highly technical stuff, such as server machines, for maintenance of the network.

## 9. E-commerce

E-commerce (electronic commerce) is the activity of electronically buying or selling of products on online services or over the Internet. E-commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. E-commerce is in turn driven by the technological advances of the semiconductor industry, and is the largest sector of the electronics industry.

### Types of E-Commerce Models

Electronic commerce can be classified into four main categories. The basis for this simple classification is the parties that are involved in the transactions. So the four basic electronic commerce models are as follows,

#### 1. Business to Business

This is Business to Business transactions. Here the companies are doing business with each other. The final consumer is not involved. So the online transactions only involve the manufacturers, wholesalers, retailers etc.

## **2. Business to Consumer**

Business to Consumer. Here the company will sell their goods and/or services directly to the consumer. The consumer can browse their websites and look at products, pictures, read reviews. Then they place their order and the company ships the goods directly to them. Popular examples are Amazon, Flipkart, Jabong etc.

## **3. Consumer to Consumer**

Consumer to consumer, where the consumers are in direct contact with each other. No company is involved. It helps people sell their personal goods and assets directly to an interested party. Usually, goods traded are cars, bikes, electronics etc. OLX, Quikr etc follow this model.

## **4. Consumer to Business**

This is the reverse of B2C, it is a consumer to business. So the consumer provides a good or some service to the company. Say for example an IT freelancer who demos and sells his software to a company. This would be a C2B transaction.

## **Advantages of E-Commerce**

- E-commerce provides the sellers with a global reach. They remove the barrier of place (geography). Now sellers and buyers can meet in the virtual world, without the hindrance of location.
- Electronic commerce will substantially lower the transaction cost. It eliminates many fixed costs of maintaining brick and mortar shops. This allows the companies to enjoy a much higher margin of profit.
- It provides quick delivery of goods with very little effort on part of the customer. Customer complaints are also addressed quickly. It also saves time, energy and effort for both the consumers and the company.
- One other great advantage is the convenience it offers. A customer can shop 24×7. The website is functional at all times, it does not have working hours like a shop.
- Electronic commerce also allows the customer and the business to be in touch directly, without any intermediaries. This allows for quick communication and transactions. It also gives a valuable personal touch.

## Disadvantages of E-Commerce

- The start-up costs of the e-commerce portal are very high. The setup of the hardware and the software, the training cost of employees, the constant maintenance and upkeep are all quite expensive.
- Although it may seem like a sure thing, the e-commerce industry has a high risk of failure. Many companies riding the dot-com wave of the 2000s have failed miserably. The high risk of failure remains even today.
- At times, e-commerce can feel impersonal. So it lacks the warmth of an interpersonal relationship which is important for many brands and products. This lack of a personal touch can be a disadvantage for many types of services and products like interior designing or the jewelry business.
- Security is another area of concern. Only recently, we have witnessed many security breaches where the information of the customers was stolen. Credit card theft, identity theft etc. remain big concerns with the customers.
- Then there are also fulfillment problems. Even after the order is placed there can be problems with shipping, delivery, mix-ups etc. This leaves the customers unhappy and dissatisfied.

## 10. Service-Oriented Architecture

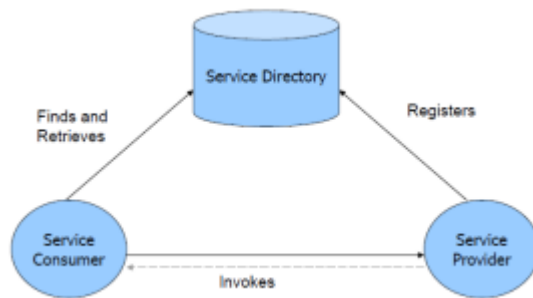
Service-Oriented Architecture built applications based on services and loosely-coupled architecture. So they can be reused. A major benefit of Service-Oriented Architecture is that it delivers enterprise agility, by enabling rapid development and modification of the software that supports the business.

Service-Oriented Architecture defined in various ways, “Service-oriented architecture is a Client/Server design approach in which an application consists of software services and software service consumers (also known as clients or service requesters). SOA differs from the more general client/server model in its definitive emphasis on loose coupling between software components, and in its use of separately standing interfaces” (Gartner).

Also Service-Oriented Architecture enable flexible, Combined Business Processes, and business process optimization and the Real Time Enterprise (RTE)

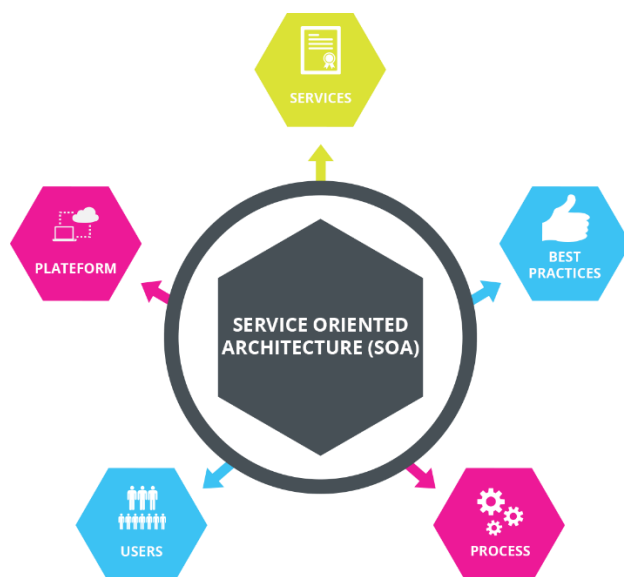
SOA is method of design, development and management of both application and the software infrastructure where: all software is organized into business services that are network accessible and executable, service interface are based on public standards for interoperability. Service-Oriented Architecture is an architectural approach that allows distributed deployment by expose enterprise data and business logic as loosely coupled, discoverable, structured, standards-based, coarse-grained, stateless units of functionality called services. Furthermore allows reusability by choose a services provider and access to existing resources exposed as services. By allowing reusing the existing applications Service-Oriented Architecture enables

enterprise to influence existing investments. Another importance is composability by allowing assemble new processes from existing services that are exposed at a desired granularity through well defined, published and standards compliant interface. Also provide interoperability by share capabilities and reuse shared services across a network irrespective of underlying protocols or implementation technology. Key characteristic of SOA: quality of service- response time, security and performance, service is cataloged and discoverable, data are cataloged and discoverable, protocols use only industry standards



A SOA has three major parts; service provider, service consumer, and service directory. Service providers are the parties who build service and make available service. Service consumers are the clients who consume services. Service directory is the place where service providers register the services and consumer search for services. Service directory provide following services: 1. Scalability of services; can add services incrementally. 2. Decouples consumers from providers. 3. Allows for hot updates of services. 4. Provides a look-up service for consumers. 5. Allows consumers to choose between providers at runtime rather than hard-coding a single provider

There are three roles in each of the Service-Oriented Architecture building blocks: service provider; service broker, service registry, service repository; and service requester/consumer.



## What can you do with SOA?

There are several things that can be done with SOA:

### 1. Making a Reliable Service

It could be used to make a reliable service.

It could be used to make a reliable service which contains the following features:

- Improved information flow.
- Ability to expose internal functionality.
- Organizational flexibility.

### 2. Making Reusable Service

One of the main use of SOA is to make a reusable service. Therefore, SOA concepts could be easily used and implemented to make a service that is not limited to a single component but could be used in multiple components.

### 3. Configuration Flexibility

It is highly flexible and could be easily configured as per our needs.

### 4. For Developing new Function Combinations

It could be used for developing new functions combinations rapidly as per need or requirement.

## Advantages and Disadvantages

Given below are the advantages and disadvantages mentioned:

### *Advantages:*

- **Maintenance is Easy:** Editing and updating any service implemented under SOA architecture is easy. You don't need to update your system. A third party maintains the service, and any amendment in this service won't have an effect on your system. In most cases, the previous API work because it is functioning before.
- **Quality of Code Improved:** As services run freelance of our system, they have their own variety of code; therefore, our code is prevented from redundancy. Also, our code becomes error-free.
- **Platform Independence:** Services communicate with alternative applications through a common language, which implies it's freelance of the platform on which that application is running. Services can provide API in different languages, e.g. PHP, JavaScript, etc.

- **Scalable:** If any service obtains several users, it is often simply scalable by attaching additional servers. This will create service out there all time to the users.
- **Reliable:** Services square measure typically tiny size as compared to the full-fledged application. So it's easier to correct and check the freelance services.
- **Same Directory Structure:** Services have an equivalent directory structure so customers can access the service information from an equivalent directory on every occasion. If any service has modified its location, then the additional directory remains the same. This is very helpful for consumers.
- **Independent of Other Services:** Services generated using SOA principles are independent of each other. So services are often utilized by multiple applications at an equivalent time.

#### ***Disadvantages:***

- **High Bandwidth Server:** Therefore, net service sends and receives messages and knowledge often times, so it simply reaches high requests per day. So it involves a high-speed server with plenty of information measures to run an internet service.
- **Extra Overload:** In SOA, all inputs square measure its validity before it's sent to the service. If you are victimization multiple services, then it'll overload your system with further computation.
- **High Cost:** It is expensive in terms of human resources, development, and technology.

#### **Why Should we Use SOA?**

This has multiple advantages, as we have discussed earlier in this article. We can use it for making reliable, better, injectable and reusable services.

#### **Why do we Need SOA?**

It could be used for solving various business needs:

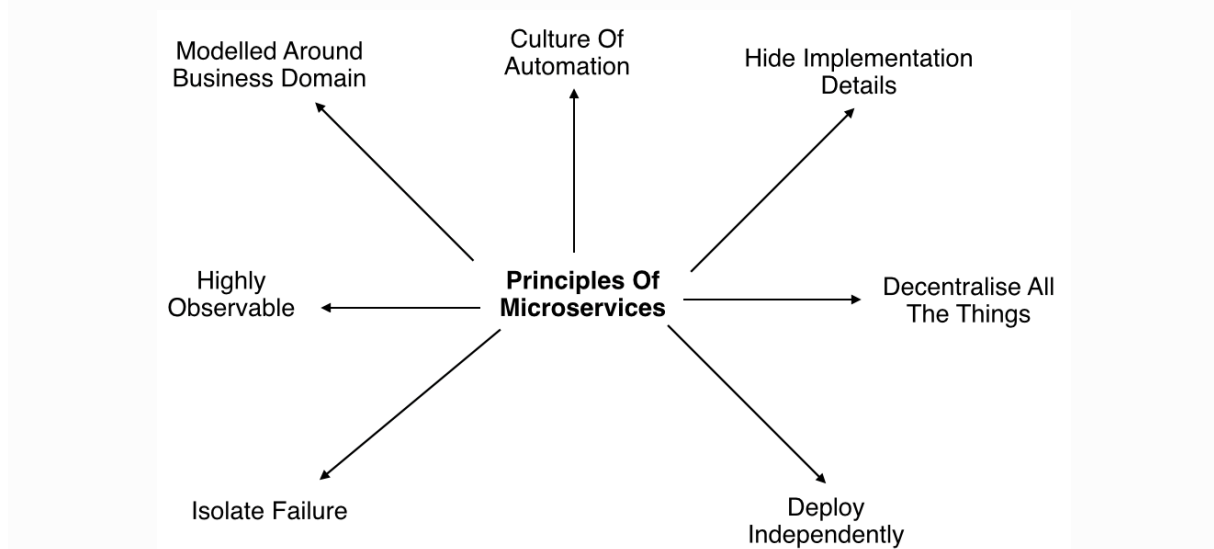
- **For Developing Independent Services:** It is required if our business need is to develop multiple services which are independent of each other. However, these services can still communicate with each other.
- **To Expose Data:** Exposing the functionality of the software as a service is easier to implement if we are using SOA.
- **To Develop Reusable Service:** If our requirement is to develop a reusable service, then SOA is perfect for this. It could be used to make independent, reliable and reusable services.

## **11. Micro-service**

Micro-service architecture is one of the booming concepts nowadays that promise quick and flawless software changes compared to traditional or monolithic architectures by modularizing complex applications into distributed parts that run parallel without any harm to another part of the applications. The team of developers composes the overall applications by the resulting in the upgradable, interchangeable, and critical scalable parts of the

application. When it comes to the rapid application development era, such kind of modular architectural style surely helps to accelerate the business growth by enabling the agile deployment of the rapid client functionalities. But again, decomposing the part of applications can also come with some complexity compare with the monolithic model.

### Micro-services Principles



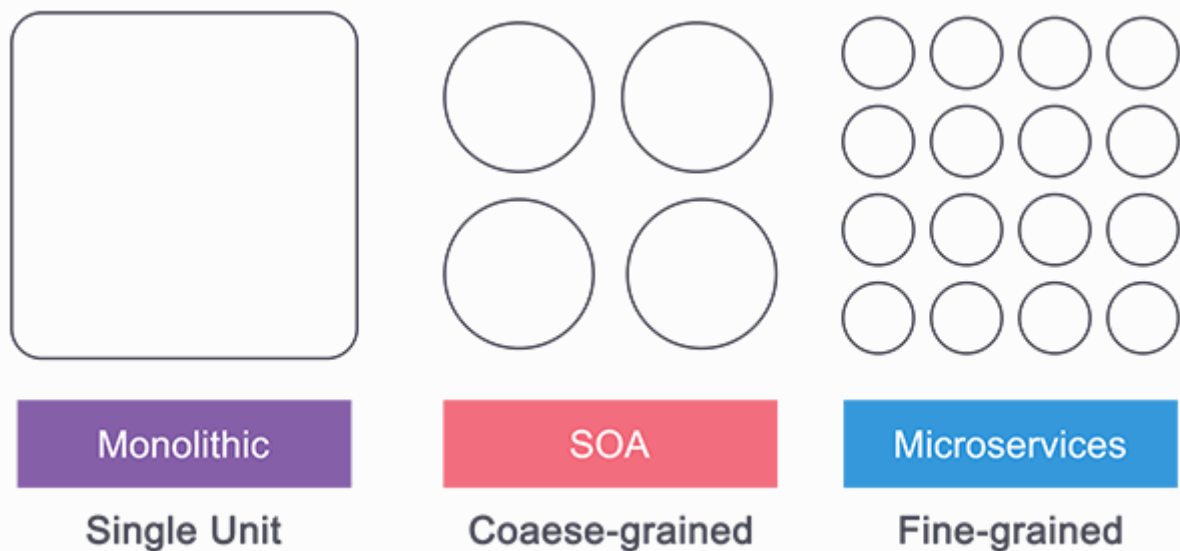
- **Modelled around business domain:** Micro-services architecture lets us separate system capability into different domains. Each domain will focus on one thing and its associated logic and can easily migrate independently to the next version, and also scale independently according to requirement.
- **Culture of Automation:** As we are building, testing, deploying, and monitoring each service separately and there is an increase in the number of deployment units compared to monolithic architecture we should follow the culture of automation by designing it for continuous integration and continuous delivery.  
The smaller and compact codebases and their defined scope are generally turned out to quicker deployments, which also allow us to start to explore the benefits of Continuous Deployment and Continuous integration seamlessly.
- **Hide implementation details:** Micro-services should be architected in such a way that they won't expose the internal details; neither technical implementation nor the business rules that drive it. This will reduce the coupling and help to do changes and improvements without affecting the overall architecture.
- **Decentralization:** In traditional monolithic implementations, the software is designed to use a single database with different tables whereas micro-services are designed in such a way to manage their own database.
- **Deploy Independently:** To enjoy the complete benefits of the architecture, Micro-services should be independently deployable. If you are failing to do so, check for any coupling in the application and solve it.



- **Failure Isolation:** The impact of a failure is less in Micro-services architecture compares to the monolithic type as it will only affect that particular service and its association while other services can keep running. The associated services should handle such scenarios when the dependent is unresponsive or slow.  
The larger or the enterprise applications may remain unaffected mostly by the failure of a single module and due to that, other parts of the application are running concurrently which enhances the availability of the feature to the customers most of the time.
- **Highly Observable:** The services should collect as much information to analyze what is happening within each of them like log events and stats.

### Monolithic vs. SOA vs. Micro-services

Monolithic architectures are the simplest form of architecture as it is having only one application layer that bundles together all the software components, and is hosted and delivered together. This type has been widely used by many small and mid-sized companies. The main challenge in this system is during scaling up as we need to duplicate the whole system including all the features of other machines which increases the cost. Also, the failure of one feature will affect the whole system making it unreliable.



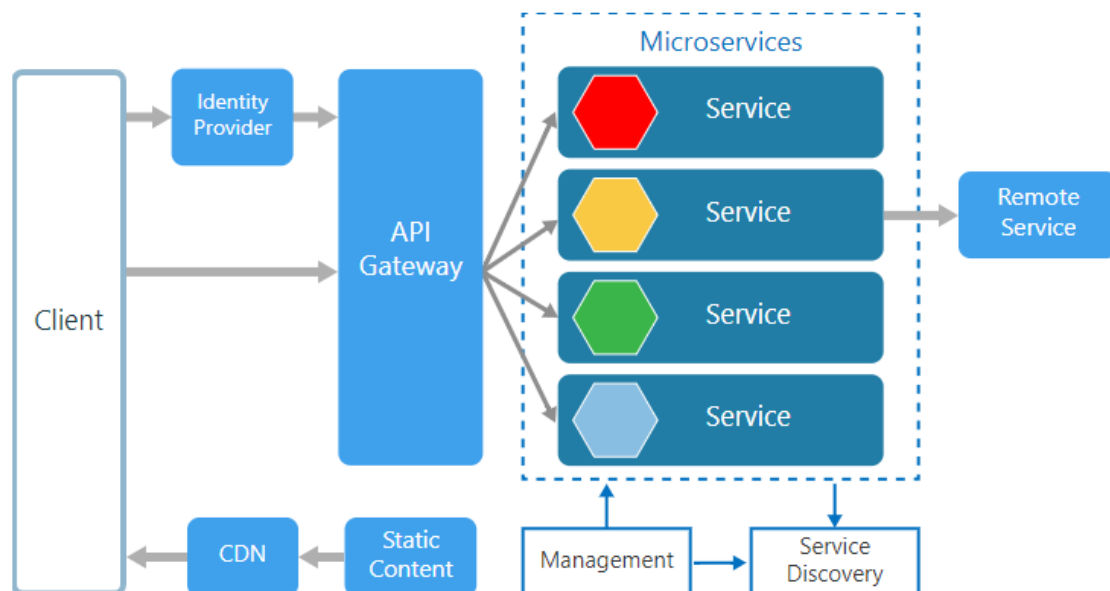
Service-Oriented Architecture (SOA) follows a coarse-grained structure where the features of an application are broken down into smaller components as services comprised of some tasks. This type of architecture allowed us to horizontally scale each service, and also more flexibility and performance at the cost of increasing the complexity of the architecture compared to the monolithic. Each service can be written in different languages and the communication between them can be done with the help of a middleware

Micro-services have technically evolved out of SOA where those features are further broken down into tasks level services making it fine-grained architecture. While Service Oriented Architecture followed a centrally governed architecture where each component is controlled by a central middleware, in Micro-services it's a decentralized governing system where components talk directly to each other and can be written in different programming languages and communicate without the help of any broker and are done with the help of REST API.

The differences between Micro-services and SOA can be a bit fuzzy while the technical aspects of both of them can be drawn between the Micro-services and the SOA architecture somehow, and most probably around the role of the "Enterprise Service Bus". When it comes to the defining difference, it is pretty easy to consider the difference as one of the scopes. The SOA architecture was one of the extensive efforts to standardize the way all the web services work in an organization to talk to and integrate with each other, and when it comes to the Micro-services architecture, it is a kind of application-specific platform.

## Micro-services Architecture

An architectural style that structures an application as a collection of small self-contained processes, modelled around a business capability. They don't share the data structure and will be communicating through APIs. While in a monolithic application all the components are in a single module, in Micro-services we can see all the components are divided into a separate module and communication happens with each other with the help of APIs. In Micro-services Architecture the data is federated where each Micro-services is responsible for its own data model and data.



Being small in size, independent, and loosely coupled each service can be developed and deployed independently by a small team as each service is having its own codebase. Data and state persistence should be taken with each service as it lacks a separate data layer to handle it. The services only communicate with well-defined APIs hiding each service's internal implementation from the other. Each service can use a different technology stack, language, libraries, or frameworks.

- **Management:** The Management takes care of the placement of services on nodes, checking for failures, and rebalancing services across nodes in case of any failures.
- **Service Discovery:** Maintains a list of services and the nodes where each service is located, and also enables the service to look up to find the endpoint for a particular service.

- **API Gateway:** The entry point for clients where all the calls from the client will be taken, analyze, and forward to appropriate services. In case some calls are needed from multiple services API Gateway will aggregate and will return the aggregated result.

#### Companies using Micro-services

There are many companies using Micro-services for their products and services and here is a list of a few who shared their experiences.

- Comcast Cable
- Uber
- Netflix
- Amazon
- eBay
- Sound Cloud
- Karma
- Microsoft
- Groupon
- Hailo
- Gilt
- Zalando
- Lending Club
- AutoScout24

#### Advantages of Micro-services

- Services can be written in different programming languages and can be accessed by using any framework.
- Independently develop, deploy, redeploy, version, and scale component services in seconds without compromising the integrity of an application
- Better fault isolation keeps other services working even though one got failed.
- Zero downtime upgrades.
- Services can be of from different servers or even different data centers.
- Interaction with other services in a well-defined protocol
- Monitor, capture, and report health diagnostics
- Reliable and self-healing
- Supports continuous integration and delivery
- Easy to transfer knowledge to the new team member
- Easy to integrate with third parties

#### Disadvantages of Micro-services

- The additional complexity for implementation of an inter-process communication mechanism between services.
- Writing automated tests involving multiple services is challenging and It can be difficult to create consistent testing environments.
- Requires a high level of automation to manage multiple instances of different types of services in production.
- Everyone has to manage eventual consistency as maintaining strong consistency becomes extremely difficult.
- Managing multiple databases and their transactions is difficult.

- Inter-process calls are slow.
- Debugging will become difficult.
- Complexity in DevOps.
- Production monitoring cost is higher.
- Formal documentation overhead.
- Lack of governance.

## 12. Cloud Computing

What is cloud computing architecture?

In layman's terms, the architecture of cloud computing refers to how various technological components work together to create a cloud, in which resources are pooled and shared across a network using advanced technology, such as virtualization. Virtualization is the segmentation of one physical server into multiple logical servers. In a nutshell, "cloud computing architecture is a hybrid of (SEO) service-oriented and (EDA) event-driven architecture."

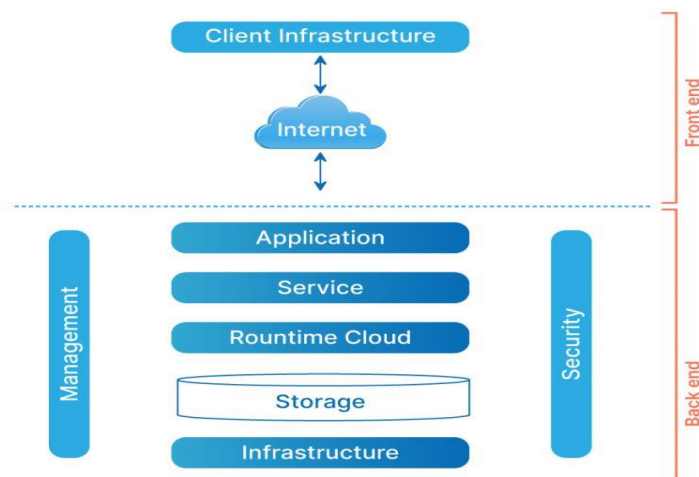
Cloud Architect is the IT professional overseeing a company's cloud computing architecture. As dated 9/7/2022, the salary for cloud architecture or Cloud Architect in India ranges from 9.0 Lakhs to 39.9 Lakhs, with an average annual salary of 21.0 Lakhs.

Cloud architecture consists of two major parts:

- Front End
- Back End

Here's the cloud computing architecture diagram:

### **ARCHITECTURE OF CLOUD COMPUTING**

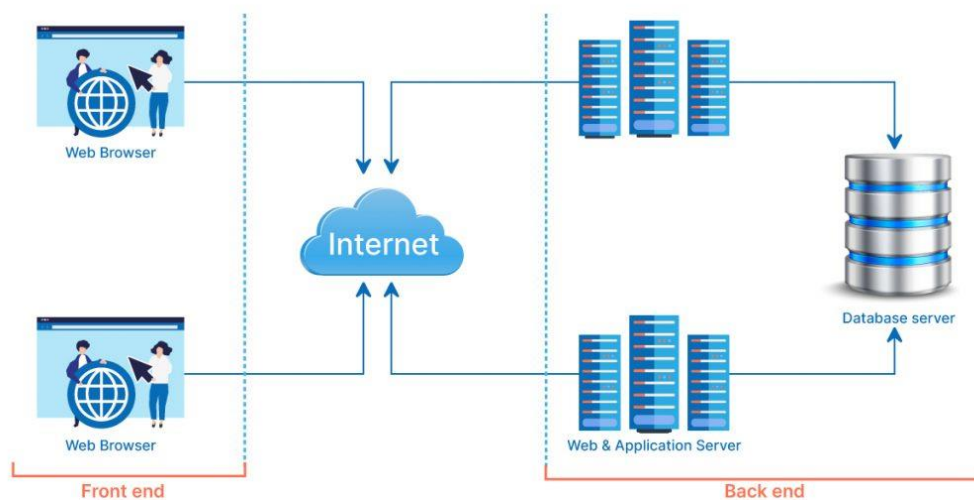


What is the Front End of cloud architecture?

Everything with which the end user interacts is part of the front-end infrastructure. The user interface is the result of integrating various sub-components, such as browsers, tablets, mobile devices, etc. With the help of the front end, the end user can connect to the cloud computing infrastructure. In short, “the front end is the end that the client interacts with.”

What is the Back End of cloud architecture?

The back end is everything the user does not usually see and everything that processes the data. The service provider uses the back end to manage all the resources required to provide cloud computing services, such as data storage, security mechanisms, virtual machines, deploying models, servers, traffic control mechanisms, and so on. In short, “the back end is the end that service provider interacts with.”



## Components of cloud computing architecture

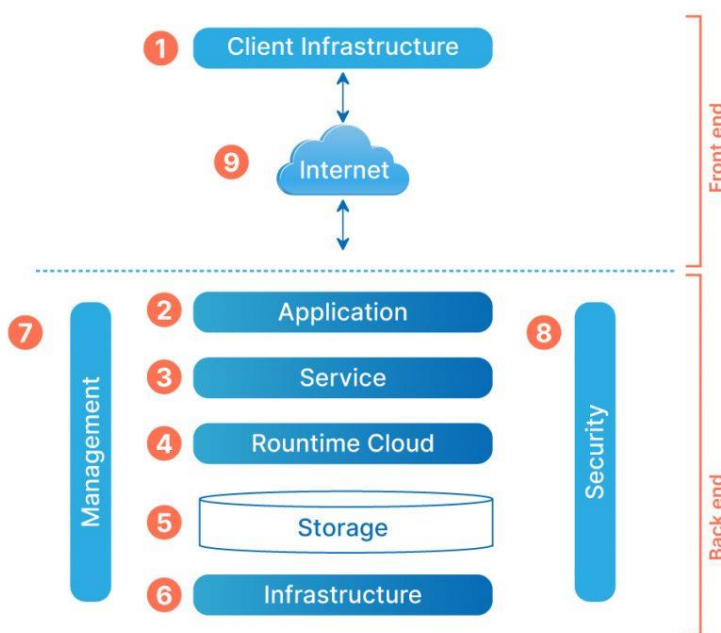
There are various components of cloud architecture. Some of those components are:

- **Client infrastructure:** The client infrastructure component is a frontend component that provides a graphical user interface (GUI) for users to interact with the cloud.
- **Application:** An application is any platform, such as an app or software, provided by a company through which clients can access the cloud.
- **Service:** A cloud service manages the type of service a client uses based on his needs. There are three types of services: SaaS (software as a service), PaaS (platform as a service), and IaaS (infrastructure as a service).
- **Runtime cloud:** The runtime cloud offers virtual machine implementation and runtime environment.
- **Storage:** The cloud computing storage component provides storage space in the cloud for managing and storing data. Cloud storage is of three types: public cloud, private cloud, and hybrid cloud.
- **Infrastructure:** The infrastructure component provides services on three levels: the host, the application, and the network. It includes the software and hardware components

required to support the cloud computing model, such as storage network devices, servers, and other storage resources.

- **Management:** The management component is in charge of managing backend components such as storage services, applications, runtime cloud infrastructure, and security issues, as well as establishing coordination.
- **Security:** Security is the backend component of cloud computing that ensures data security in the cloud.
- **Internet:** The internet is the medium through which the frontend and backend components communicate and interact.

## ARCHITECTURE OF CLOUD COMPUTING



### How does cloud computing architecture works?

The operation of cloud computing architecture depends on the type of cloud you use. As previously stated, there are various clouds, such as public, private, and hybrid, and no two clouds are the same. As a result, the working of cloud architecture depends on the type of cloud chosen. Let's delve deeper into types of cloud architecture to understand their working better.

**Public cloud architecture:** A cloud services provider owns and operates computing resources, such as data storage, security mechanisms, virtual machines, etc, in a public cloud architecture. The internet is used to share and redistribute those resources among multiple tenants. The benefits of using the public cloud architecture include lower operating costs, easy scalability, and little to no maintenance.

Private cloud architecture: A private cloud architecture is majorly owned and managed by a company. Private cloud architecture can consist of multiple server locations or leased space in geographically dispersed colocation facilities. Although private cloud solutions are typically more expensive than public cloud solutions, they are more customizable and can provide stringent data security and compliance options.

Hybrid cloud architecture: A hybrid cloud environment combines the operational efficiencies of the public cloud with the data security capabilities of the private cloud. Hybrid cloud architecture help organizations consolidate IT resources while allowing them to migrate workloads between environments based on their IT and data security requirements by utilizing both public and private cloud architectures.

### **Advantages of cloud computing architecture**

There are various advantages of cloud computing architecture, and some of those advantages are:

- Cloud architecture gives good user accessibility.
- Cloud architecture provides better disaster recovery.
- Cloud architecture makes overall cloud computing system simpler.
- There are fewer physical locations for hackers to target, reducing security risks.
- Services are automatically updated. Hence you don't have to worry about updating it manually.
- Remote working is more efficient and flexible because of the mobility inherent in such systems, thus allowing teams to be more productive.
- Cloud architecture enables businesses to scale up and down their cloud resources quickly and includes a flexibility feature that provides a competitive advantage.
- This architecture eliminates latency issues, improves data processing requirements, lowers IT operating costs, and provides easy access to data and digital tools.

### **Disadvantages of Cloud Computing**

A list of the disadvantage of cloud computing is given below -

#### **1) Internet Connectivity**

As you know, in cloud computing, every data (image, audio, video, etc.) is stored on the cloud, and we access these data through the cloud by using the internet connection. If you do not have good internet connectivity, you cannot access these data. However, we have no any other way to access data from the cloud.

#### **2) Vendor lock-in**

Vendor lock-in is the biggest disadvantage of cloud computing. Organizations may face problems when transferring their services from one vendor to another. As different vendors provide different platforms, that can cause difficulty moving from one cloud to another.

### 3) Limited Control

As we know, cloud infrastructure is completely owned, managed, and monitored by the service provider, so the cloud users have less control over the function and execution of services within a cloud infrastructure.

### 4) Security

Although cloud service providers implement the best security standards to store important information. But, before adopting cloud technology, you should be aware that you will be sending all your organization's sensitive information to a third party, i.e., a cloud computing service provider. While sending the data on the cloud, there may be a chance that your organization's information is hacked by Hackers.

