MEMORY

SOFTWARE AS A SERVICE

Introduction

Memorysoft Software as a service (SaaS) is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted and accessed remotely by users over the internet. Memorysoft SaaS applications is delivered to users via the web. SaaS is also known as "on-demand software" and Web-based/Web-hosted software.

Instead of installing and maintaining software, you simply access it via the Internet, freeing yourself from complex software and hardware management.

SaaS applications are sometimes called Web-based software, on-demand software, or hosted software. Whatever the name, SaaS applications run on a SaaS provider's servers. The provider manages access to the application, including security, availability, and performance.

Memorysoft SaaS characteristics.

SaaS is delivered through the internet as a full functional service, accessible via any web browser. With SaaS, vendors manage the data, servers and storage, ultimately eliminating the need for IT review and streamlining business processes.

A good way to understand the SaaS model is by thinking of a bank, which protects the privacy of each customer while providing service that is reliable and secure—on a massive scale. A bank's customers all use the same financial systems and technology without worrying about anyone accessing their personal information without authorization.

The characteristic model for Saas consists;

Multitenant Architecture/ Low Setup and Infrastructure Costs

A multitenant architecture, in which all users and applications share a single, common infrastructure and code base that is centrally maintained. Because SaaS vendor clients are all on the same infrastructure and code base, vendors can innovate more quickly and save the valuable development time previously spent on maintaining numerous versions of outdated code.

Easy Customization

The ability for each user to easily customize applications to fit their business processes without affecting the common infrastructure. Because of the way SaaS is architected, these customizations are unique to each company or user and are always preserved through upgrades. That means SaaS providers can make upgrades more often, with less customer risk and much lower adoption cost.

Better Access and Scalability

Improved access to data from any networked device while making it easier to manage privileges, monitor data use, and ensure everyone sees the same information at the same time.

With Saas, you can adapt your requirements to the number of people who need to use the system, the volume of data and the functionality required as your business grows.

The SaaS platform is:

- Available over the Internet.
- Hosted on a remote server by a third-party provider, or Internal server.
- Scalable, with different tiers for small, medium and enterprise-level businesses.
- Inclusive, offering security, compliance and maintenance as part of the cost.

SaaS advantages.

With SaaS, you don't need to install and run software applications on your computer. Everything is available over the internet when you log in to your account online. You can usually access the software from any device, anytime — as long as there is an internet connection.

The same goes for anyone else using the software. All your staff will have personalized logins suitable to their access level **(Employee Self Service)**. You no longer need to engage an IT specialist to download the software onto multiple computers throughout your office or worry about keeping up-to-date software on every computer. It's all taken care of in the Cloud.

Another key advantage is the payment structure.

Most SaaS providers operate a subscription model with a fixed, inclusive monthly account fee. You know precisely how much the software will cost and can budget accordingly without worrying about hidden surprises.

Subscriptions can potentially include maintenance, compliance and security services. SaaS providers also offer out-of-the-box, simple solutions to set up if you need a basic package, with more complex solutions for larger organizations. You could have the basic software up and running within a matter of hours – and you'll have access to customer service and support along the way.

MEMORY

When to use SaaS.

SaaS platforms are ideal for when you want an application to run smoothly and reliably with minimal input from you.

Take your email server, for example.

You want to know that you can continue to send and receive emails without needing to fiddle with your email settings or worry about updates. Imagine if your email server went under because you forgot to update it? That's simply not an option in today's marketplace.

If you use SaaS applications to run your email inbox, the chances of something going wrong are very small, and if something did go wrong, it would be up to the SaaS provider to find a solution.

You're not only paying for the SaaS applications/products — you're paying for peace of mind.

Memorysoft Frontend (PHP, HTML, CSS)

The front end Memorysoft development has one general responsibility: to ensure that website visitors can easily interact with the page. This is done through the combination of design, technology and programming to code a website's appearance, as well as taking care of debugging.

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly.

The Memorysoft Saas Front End platform will use a web browser software to retrieve, present and traverse information on the World Wide Web (www).

Memorysoft Backend (Visual Basic)

Back-end development involves building and maintaining the mechanisms that process data and perform actions on websites. Unlike front-end developers, who control everything you can see on a website, back-end development is involved in data storage, security, and other server-side functions that you cannot see.

Back-end development means working on server-side software, which focuses on everything you can't see on a website. Back-end developers ensure the website performs correctly, focusing on databases, back-end logic, application programming interface (APIs), architecture, and servers. They use code that helps browsers communicate with databases, store, understand, and delete data.

On a team, back-end development collaborates with front-end development, product managers, principal architects, and website testers to build the structure of a website/ Saas.

The 3 general parts of the backend development are;

1. **The server**. This is the computer that receives requests.
2. **The application**. This is the application running on the server that listens for requests from the user on the web, retrieves information from the database, and sends a response.
3. **The database**. Databases are used to organize and persist data.

The backend is the code that runs on the server, that receives requests from the clients, and contains the logic to send the appropriate data back to the client. The back-end also includes the database, which will persistently store all of the data for the application.

Middleware

Middleware is any code that executes between the server receiving a request and sending a response. These middleware functions might modify the request object, query the database, or otherwise process the incoming request. Middleware functions typically end by passing control to the next middleware function, rather than by sending a response.

Middleware essentially describes any part of software on the server that connects an application's front end to its back end. Think of middleware as plumbing for your site—it pipes any communication, like requests and responses, back and forth between your application and your server/database. Just like plumbing in a house, you don't see middleware, but it's there and it has to be reliable and always do what's expected of it.

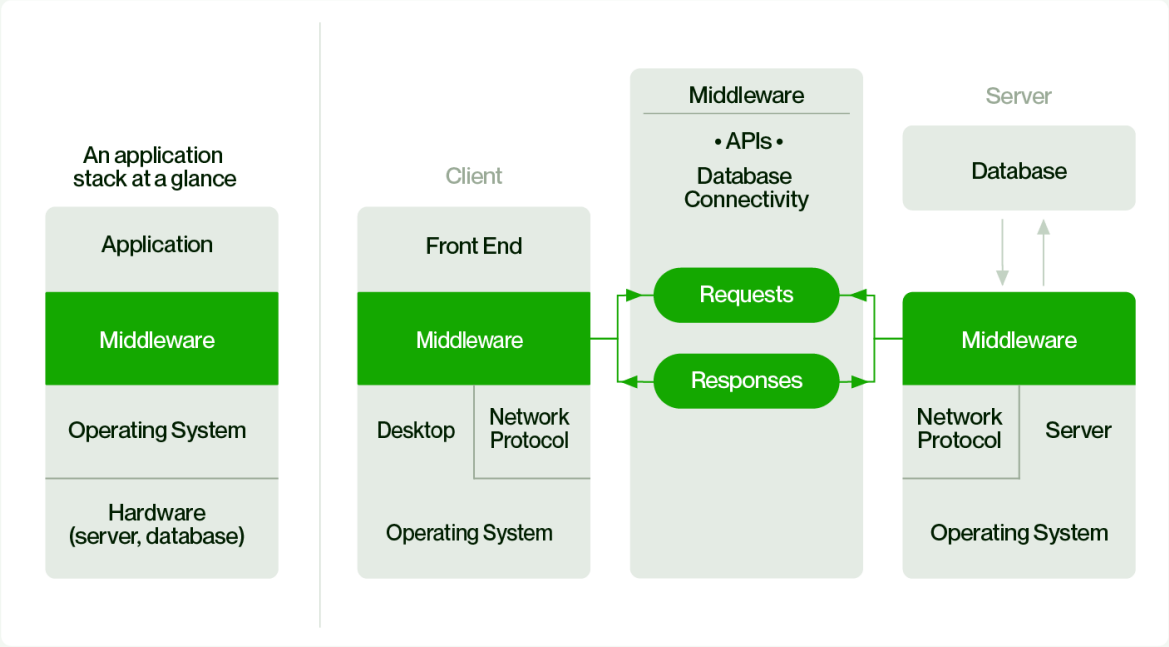
Middleware (server-side software) facilitates client-server connectivity, forming a middle layer between the app(s) and the network: the server, the database, the operating system, and more. Middleware can be multi-layered, organized into different layers of a site, whether it's the presentation layer or the business layer. This is also where Web APIs can play into the stack, providing a bridge between the business layer and presentation layer.

MEMORY

What is Middleware?

upwork

Middleware is any software that acts like “glue” between an application and its network. It controls the flow of information etween an application and the server, database, and operating system.



Database

Servers and the database. The server is the link that powers the system and determines whether or not it will be available to users. Servers are powerful computers that provide the backend process with all the resources needed. They keep the database running, store and share files, provide security and reliable data encryption, and much more.

The database is responsible for storing and finding information requested on the client side. It can be updated with new articles, photos, videos, or other changes directly from the browser every time a client, having access to the database, enters the above information

Database Security

Website/Frontend Data Security

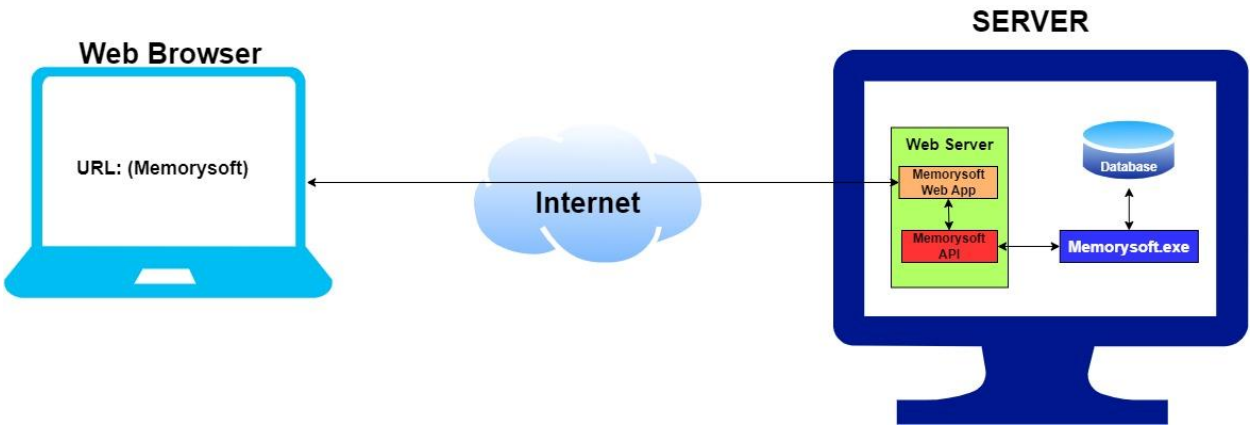
SSL/TLS works by ensuring that any data transferred between users and websites, or between two systems, remains impossible to read. It uses encryption algorithms to scramble data in transit, which prevents hackers from reading it as it is sent over the connection.

Backend Data Security

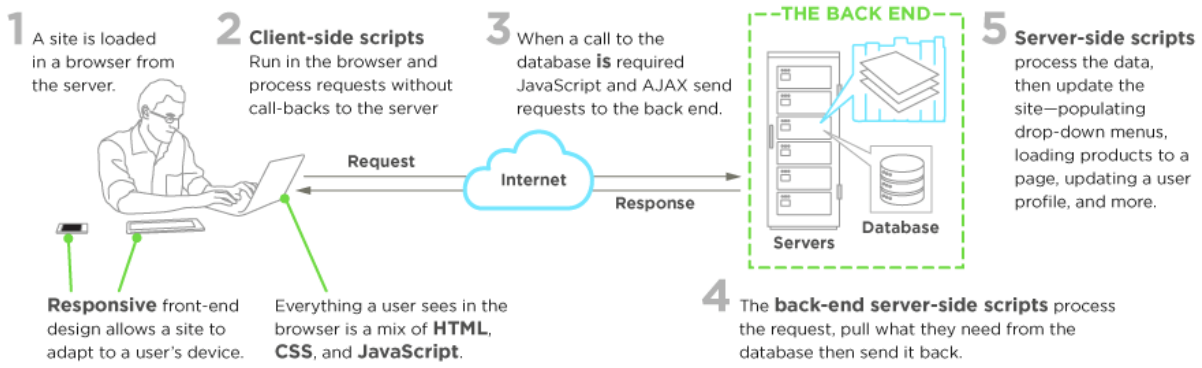
A firewall is a network security device that monitors incoming and outgoing data from your network and allows or denies the data to reach its end destination depending on rules configured by the entity responsible for managing the firewall. Its purpose is to block malicious traffic like viruses, malware, and hackers, and is the first line of security defense for your network.

Firewalls can be physical devices, or a software program running on the servers and on the workstations.

Mapping out the procedure/request



MEMORY



i. On the website/ frontend (client side), the user clicks on a command. This is done on the user interface of the software.
ii. The user request travels across the internet to the Memorysoft servers.
iii. The server, which is actively listening for requests from all users, receives the request.
iv. Event listeners that match this request are triggered. The code that runs on the server between the request and the response is the middleware (API).
v. The middleware triggers the memorysoft application (Backend) to process the request, and read or write on the database. Using the backend technology, the middleware creates a path between the client side and the server, database, operating system, and other network components.
vi. The backend processes the data, then sends the response through the middleware back to the Web page (Frontend). This response travels across the internet back to the user's machine web page.
vii. The frontend initializes and the response is the visible to the user.