Unit 4: Device Discovery and Cloud Services for IoT

Device Discovery:

- **Device discovery is a problem of finding useful devices in order to accomplish a task, and it constitutes a major challenge for IoT.**
- **Once a device finds another with the desired characteristics and establishes a negotiated communication, novel services can be generated through cooperation.**

Device Discovery Capabilities:

- **The Internet of Things (IoT) is continuously growing to connect billions of smart devices anywhere and anytime through an Internet-like structure.**
- **The IoT devices are capable of sensing, analyzing and evaluating the surrounding objects and people**
- **They can collaborate and work together to provide a set of new applications and services, such as smart home, e-healthcare and intelligent transportation system.**
- **An intelligent device discovery strategy may allow the IoT devices to efficiently establish new connections with other devices based on their need.**
- **Connections can be set up to gather a certain number of computing powers, network functions, program source codes, raw datasets, and etc to enable new services and applications.**
- **For example, a glucose level monitor can send a request to find a glucose analyzer, and collaborate on evaluating a given patient's glucose level.**
- **The glucose analyzer may also need to search for a national wide database to compare the given patient's results with that of other patients, and then given the advisement or alert to the given patient.**

IoT Device discovery Methods:

- **TRADITIONAL IOT DEVICE DISCOVERY SCHEMES**
- **A simple broadcast mechanism to discover the required target device in a breadth-first search manner.**
- **There are two traditional solutions for addressing the device discovery in the IoT**
    a. **Centralized IoT Device Discovery Scheme**
    b. **Distributed IoT Device Discovery Scheme**

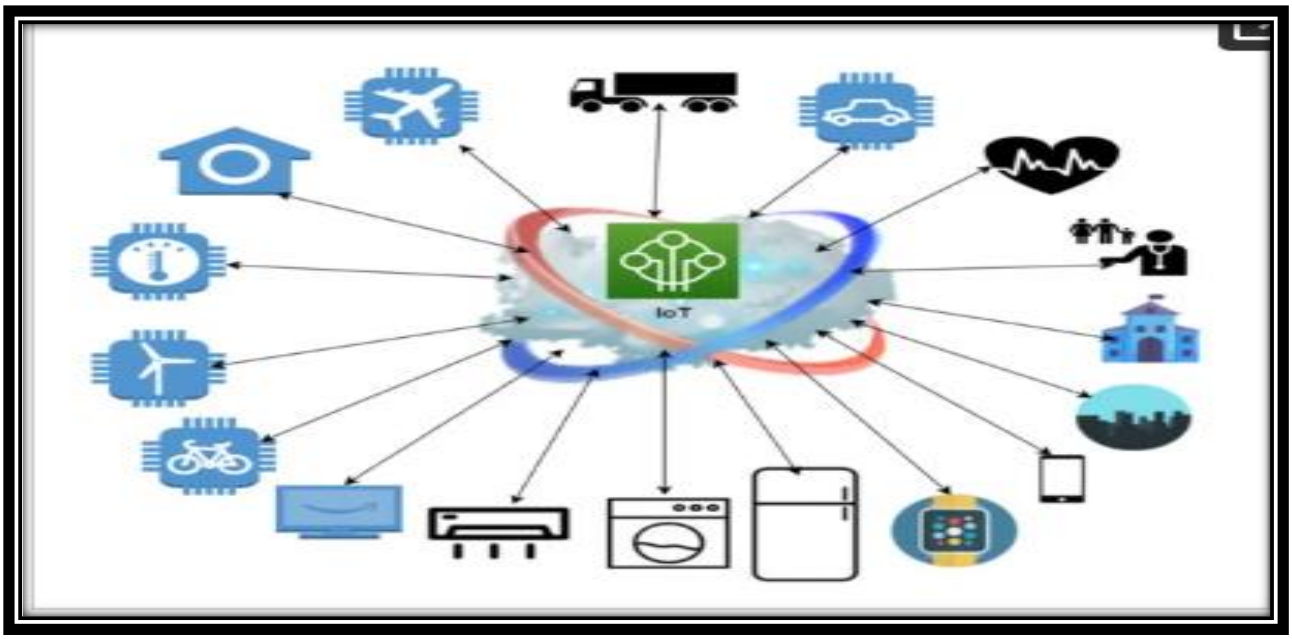


Fig 1. Centralized IoT Device Discovery Scheme

a. Centralized IoT Device Discovery Scheme

- **To achieve a fast and efficient IoT device discovery, a centralized controller this keeps track of the general information (e.g., the device's functionality) of all the devices and maintains the shortest paths between all the device pairs.**
- **The centralized controller can find the location of the desired target device and reply to the source device with the shortest path information to the target device.**
- **A centralized scheme is fast and accurate due to that the centralized controller has a global view of all the devices in the network.**
- **In addition, the centralized controller can periodically send out heartbeat messages to all the registered devices and maintain an up-to-date list of the alive devices in the network.**
- **If a device fails to reply to the heartbeat after three attempts, it is considered as in the mode of lost and will be removed from the list of alive at the centralized controller.**
- **Centralized scheme highly relies on the centralized controller which manages the whole system and the communication between the IoT devices.**
- **The centralized scheme is fast and efficient; however, it may not be scalable when the number of devices is large.**

b. Distributed IoT Device Discovery Scheme

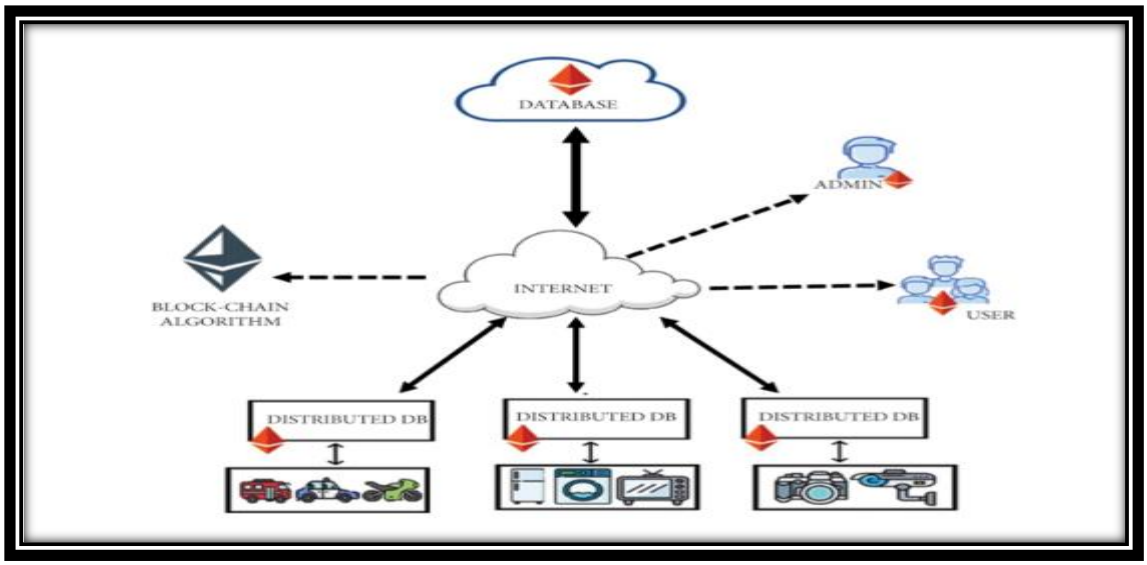


Fig 2. Distributed IoT Device Discovery Scheme

- **The distributed scheme offers a scalable device discovery in the IoT.**
- **In the distributed scheme, each device maintains a local neighbor list that consists of the neighbors' general information.**
- **When a new device joins a network, it only exchanges the general information with nearby devices that are within the transmission range.**
- **When a device leaves the network, it needs to inform its neighbors for them to update their local neighbor list.**
- **Heartbeats can be exchanged between the devices to maintain the fresh live or lost status of the devices' neighbors.**
- **The distributed discovery scheme works in a breadth-first search manner.**
- **When a source device makes a request to discover a target device with desired features, the distributed scheme simply broadcasts the discovery request to all the neighboring devices.**
- **This process iterates until the desired target device is found.**
- **Compared to the centralized scheme, the distributed scheme is more resilient against possible failures since there is no centralized control that is at the risk of single point of failure.**
- **It is also more robust when the network is dynamically changing since each device only needs to maintain their local neighbor's information.**
- **Distributed scheme can find a cost-efficient communication path between the source and target devices since the breadth-first search is used.**

Distributed IoT Device Discovery Scheme Limitations

- **The only limitation is that a large number of intermediate devices may be involved in transmitting the discovery request when broadcast is used, which is energy consuming.**
  This is harmful to the IoT system where most of the devices are power constrained.

Registering a device

- **The Internet of Things (IoT) refers to physical items equipped with sensors, embedded software, computing power, and other technologies.**
- **It may communicate with other devices over the Internet or other local communication networks. In addition, devices occasionally communicate with one another on a local network or with cloud services to perform specific tasks.**
- **When we think about IoT, the devices need to connect to the cloud, such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), Alibaba Cloud, Oracle Cloud, IBM Cloud, etc.**
- **Amazon Web Services (AWS) is a well-known cloud service company. There are many services available not just in IoT but also in other applications.**
- **AWS offers a variety of services for IoT applications, including FreeRTOS and AWS IoT GreenGrass for device software, AWS IoT Core, AWS IoT Device Management, AWS IoT Device Defender, and AWS IoT Fleetwise for control services, and AWS IoT Analytics, AWS IoT Events, AWS IoT Sitewise, and AWS IoT TwinMaker for analytics.**
- **The first step after obtaining sensor data is to create a connection between the cloud and the device when it comes to IoT.**
- **AWS' IoT core enables the communication between edge IoT devices and AWS services.**

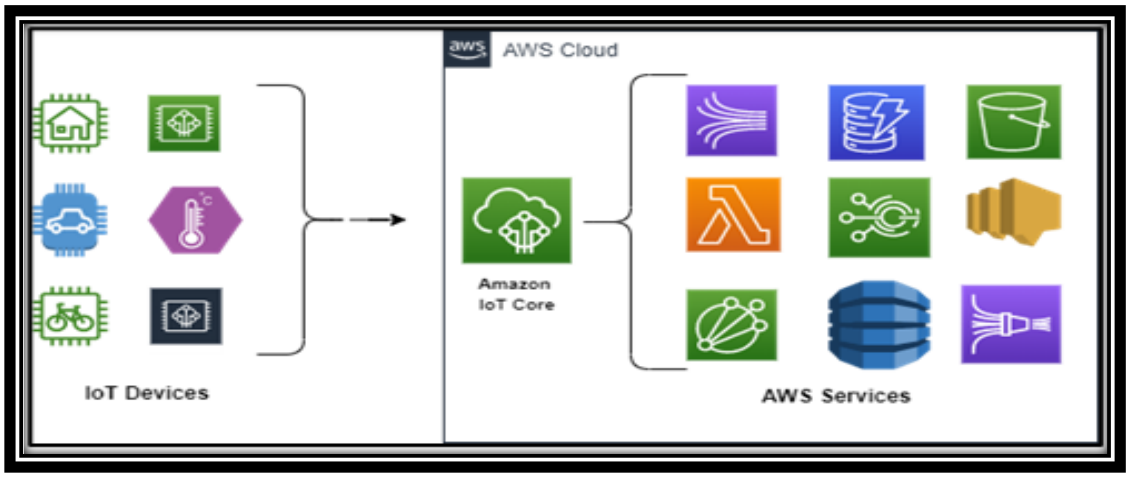


Fig 2. Registering a new device to the AWS IoT core

Deregistering the device:

- **Devices usually deregister from Device Management services for one of two reasons**
- **The registration session lifetime on the LwM2M server expires.**
- **if the client hasn't been able to send its register_update() within the registered lifetime due to connectivity issues, the LwM2M server will drop it to the deregistered state.**

- **Device Management Client requests deregistration by calling the MbedCloudClient::close() API**

The result of that request can be:

Success

- **If Device Management Client is successfully deregistered from the Device Management service, your application receives the Unregistered result through the on_status_callback callback.**

Failure

- **If the deregistration operation fails, the application receives the result through the error callback.**

When Device Management Client is deregistered from the LwM2M server for any reason, it's marked as deregistered in the Device Management services. All queued messages are removed from the server queue.

If Device Management Client tries to connect to the services while in the deregistered state, it's forced to perform a full registration again.

Deregister doesn't remove the device identity from the device, so the device uses the same identity to register with the Device Management service next time.

Introduction to Cloud Storage models and communication APIs:

- **Cloud computing is a transformative computing paradigm that involves delivering applications and services over the Internet.**
- **Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. Networks, Servers, Applications and Services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.**

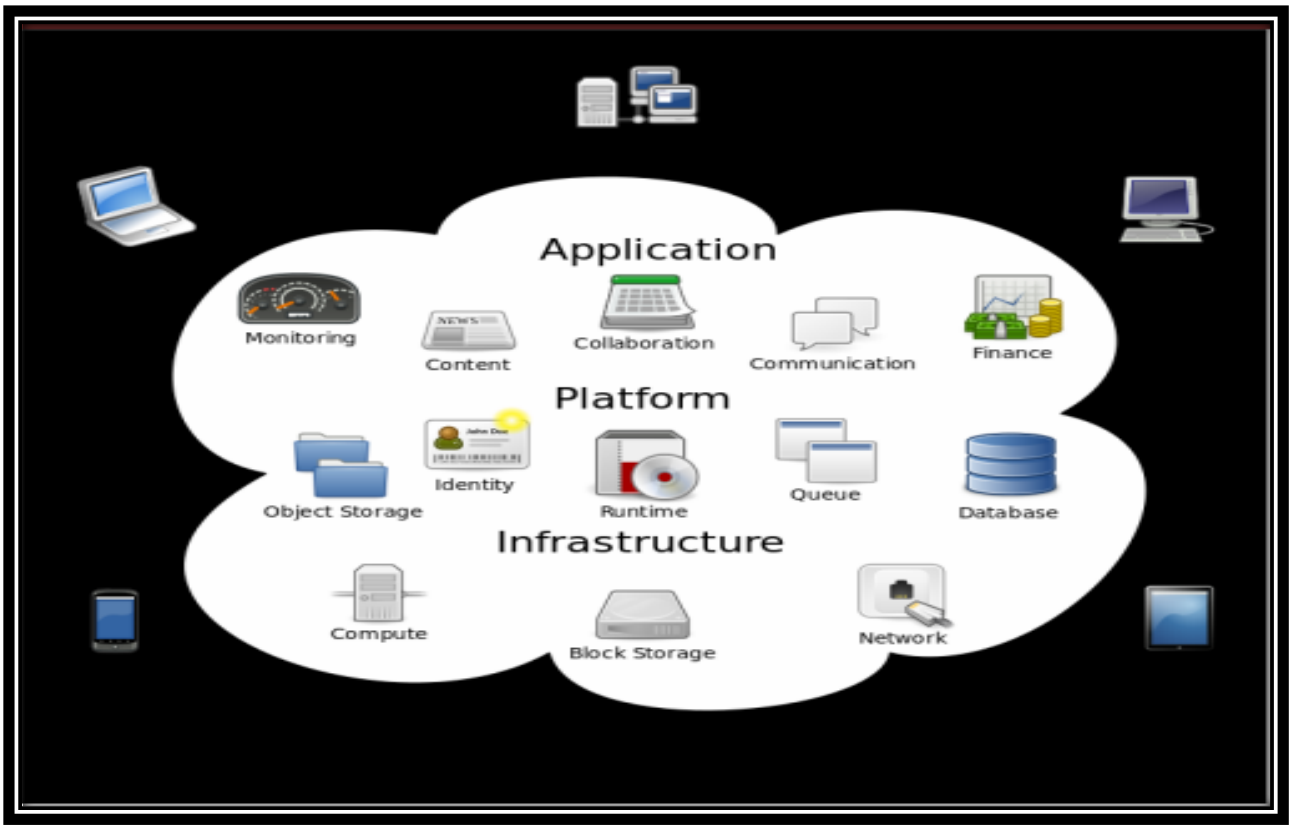


Fig 3: Cloud Offerings

Cloud Computing Deployment models are:

Deployment in cloud computing comprises four deployment models:

1 Private Cloud, 2 Public Cloud, 3 Community Cloud and 4 Hybrid Cloud

Cloud storage API (Application Programming Interface):

- **A cloud storage API is an application program interface that connects a locally-based application to a cloud-based storage system, so that a user can send data to it and access and work with data stored in it.**
- **To the application, the cloud storage system is just another target device, like tape or disk-based storage.**
- **An application program interface (API) is code that allows two software programs to communicate with each other.**
- **The API defines the correct way for a developer to write a program that requests services from an operating system (OS) or other application.**
- **APIs are implemented by function calls composed of verbs and nouns.**
- **Three basic types of APIs**
- **APIs take three basic forms:**

- **Local, Web-like and Program-like.**
    - **Local APIs are the original form, from which the name came.**
    - **They offer OS or middleware services to application programs. Microsoft's .NET APIs, the TAPI (Telephony API) for voice applications, and database access APIs are examples of the local API form.**
    - **Web APIs are designed to represent widely used resources like HTML pages and are accessed using a simple HTTP protocol.**
    - **Any web URL activates a web API. Web APIs are often called REST (representational state transfer) or RESTful because the publisher of REST interfaces doesn't save any data internally between requests.**
    - **As such, requests from many users can be intermingled as they would be on the internet.**
    - **Program APIs are based on remote procedure call (RPC) technology that makes a remote program component appear to be local to the rest of the software.**
    - **Service oriented architecture (SOA) APIs, such as Microsoft's WS-series of APIs, are program APIs**

To use cloud computing for Internet of Things we have to learn about

- **Web Application Protocol (WAMP)**
- **Xively's Platform-as-a-Services (Pass) : this platform provides tools and services for developing IoT solutions.**
- **Amazon Web Services (AWS) and their application for IoT.**

WAMP - AutoBahn for IoT:

- **Web Application Messaging Protocol (WAMP) is a sub-protocol of Websocket which provides publish-subscribe and remote procedure call (RPC) messaging patterns.**
- **WAMP enables distributed application architectures where the application components are distributed on multiple nodes and communicate with messaging patterns provided b WAMP.**

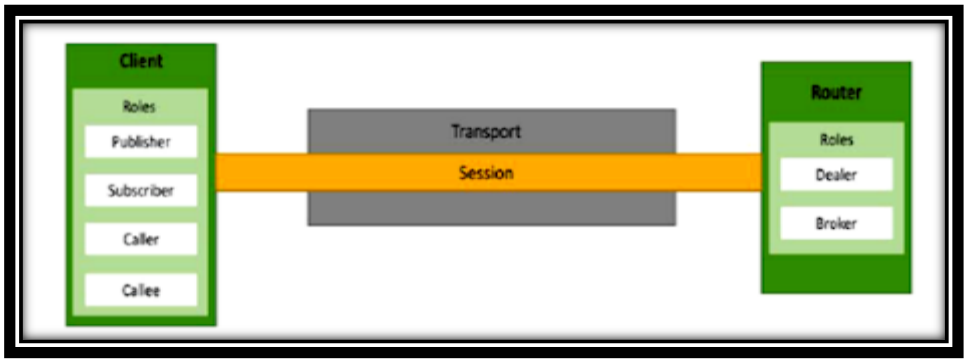


Fig 4: WAMP Session between Client and Router

- **Transport: Transport is channel that connects two peers.**
- **Session: Session is a conversation between two peers that runs over a transport.**

- **Client: Clients are peers that can have one or more roles. In publish-subscribe model client can have following roles: –**
- **Publisher: Publisher publishes events (including payload) to the topic maintained by the Broker. –**
- **Subscriber: Subscriber subscribes to the topics and receives the events including the payload.**
- **In RPC model client can have following roles: –**
- **Caller: Caller issues calls to the remote procedures along with call arguments.**
- **Callee: Callee executes the procedures to which the calls are issued by the caller and returns the results back to the caller.**
- **Router: Routers are peers that perform generic call and event routing. In publish-subscribe model Router has the role of a Broker**
- **Broker: Broker acts as a router and routes messages published to a topic to all subscribers subscribed to the topic. In RPC model Router has the role of a Broker.**
- **Dealer: Dealer acts a router and routes RPC calls from the Caller to the Callee and routes results from Callee to Caller.**

WAMP Protocol:

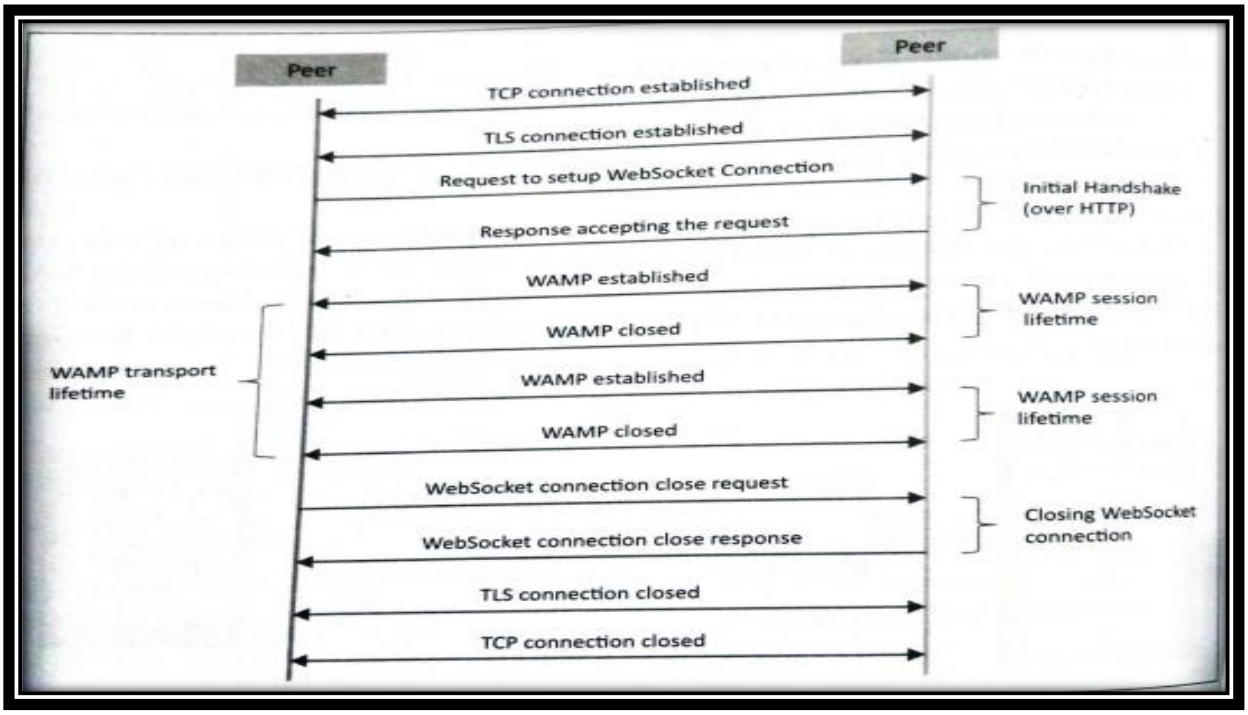


Fig 5: WAMP protocols interactions between peers.

- **WAMP transport used is Web socket.**
- **WAMP sessions are established over web socket transport within the lifetime of web socket transport.**
- **The client (in the publisher role) Runs a WAMP application component that publishes a message to the router.**
- **The router (in the broker role) runs on the server and routes the message to the subscriber. It decouples the publisher from the subscribers.**
- **The communication between publisher- broker and broker to publisher happens over a WAMP web-socket session**

Web Socket:

- **A web socket is based on the http or https protocol.**
- **It builds a connection between the browser and the server, so that either one can send data.**
- **When the browser makes a request, the server recognizes the socket and doesn't close the connection.**
- **WebSocket is bidirectional, a full-duplex protocol that is used in the same scenario of client-server communication.**
- **it starts from ws:// or wss://**
- **It is a stateful protocol, which means the connection between client and server will keep alive until it is terminated by either party (client or server).**
- **After closing the connection by either of the client and server, the connection is terminated from both ends.**

Advantages & Disadvantages of WAMP Protocol

Advantages

- **This protocol combines two patterns (Pub-Sub & RPC), allowing it to be used for the entire messaging requirements of an application, reducing technology stack complexity.**
- **Reduces networking overhead.**

Disadvantages

- **Not (yet) an official standard.**
- **No higher-level software architectural styles like REST (for now).**
- **Requires to deploy an additional component- the WAMP router. This additional component increases the time that takes to exchange messages between components compared to a direct WebSocket connection since all messages have to go through the router.**
- **Decoupled micro services are more complex than classic client-server architectures.**

Xively Cloud for IoT

- **Xively is a commercial Platform-as-a-Service that can be used for creating solutions for Internet of Things.**
- **Using Xively cloud, IoT developer can focus on the front-end infrastructure and devices for IoT(that generate the data), while the backend data collection infrastructure is managed by Xively.**
- **Xively Platform comprises of a message bus for real-time message management and routing, data services for time series archiving, directory services that provides a search-able directory of objects and business services for device provisioning and management.**
- **Xively provides an extensive support for various languages and platforms.**
- **The Xively libraries leverage standard-based API over HTTP, socket and MQTT for connecting IoT devices to the Xively cloud.**
- **Xively devices have one or more channels. Each channel enables bi-directional communication between the IoT devices and Xively cloud.**
- **IoT devices can send data to channel using the Xively APIs.**
- **For each channel, you can create on or more triggers.**
- **A trigger specification includes a channel to which the trigger corresponds, trigger condition and an HTTP POST URL to which the request is sent when the trigger fires.**
- **Triggers are used for integrations with third-party applications.**

Amazon Web Services for IoT:

- **Amazon EC2 (Amazon Elastic Compute Cloud)**
  - **Amazon EC2 is an Infrastructure-as-a Service (IaaS) provided by Amazon.**
  - **It provides secure, resizable compute in the cloud, offering the broadest choice of processor, storage, networking, OS, and purchase model.**
  - **It provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud.**
- **Amazon AutoScaling:**
  - **Monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost.**

- **Amazon S3 (Amazon Simple Storage Service)**
  - **Amazon S3 is an object storage service offering industry-leading scalability, data availability, security, and performance.**
  - **Amazon S3 is an online cloud-based storage infrastructure for storing and retrieving a very large amount of data.**
  - **S3 provides highly reliable, scalable, fast, fully redundant and affordable storage infrastructure.**
  - **S3 can serve as a raw datastore (or "Thing Tank") for IoT systems for storing raw data, such as sensor data, log data, image, audio and video data.**
- **Amazon RDS (Amazon Relational Database Service):**
  - **It is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud.**
  - **It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.**
- **Amazon DynamoDB:**
  - **Amazon DynamoDB is fully-managed, scalable, high performance No-SQL database service.**
  - **DynamoDB can serve as a scalable datastore for IoT systems.**
  - **DynamoDB is a Fast, Flexible NoSQL database service for single digital-millisecond performance at any scale**
  - **With DynamoDB, IoT system developers can store any amount of data and serve any level of requests for the data.**
- **Amazon Kinesis:**
  - **It is a fully managed commercial service that allows real-time processing of streaming data.**
  - **cost-effectively processes and analyzes streaming data at any scale as a fully managed service.**
  - **Kinesis scales automatically to handle high volume streaming data coming from large number of sources.**
  - **The streaming data collected Kinesis can be processed by applications running on Amazon EC2 instances or any other computer instance that can connect to Kinesis.**
  - **Kinesis is well suited for IoT systems that generate massive scale data and have strict real-time requirements for processing the data.**

- **Kinesis allows rapid and continuous data intake and support data blobs of size up to 50K.**

- **Amazon SQS:**
  - **Amazon Simple Queue Service (Amazon SQS) lets you send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.**
  - **SQS is simply a queue system(FIFO) that stores and releases messages in a scalable manner.**
  - **SQS can be used in distributed IoT applications in which various application components need to exchange messages.**

- **Amazon EMR (Amazon Elastic MapReduce):**
  - **It is an Amazon Web Services (AWS) tool for big data processing and analysis.**
  - **Amazon EMR also lets you transform and move large amounts of data into and out of other AWS data stores and databases, such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.**

Web Server for IoT:

- **Web Server:**

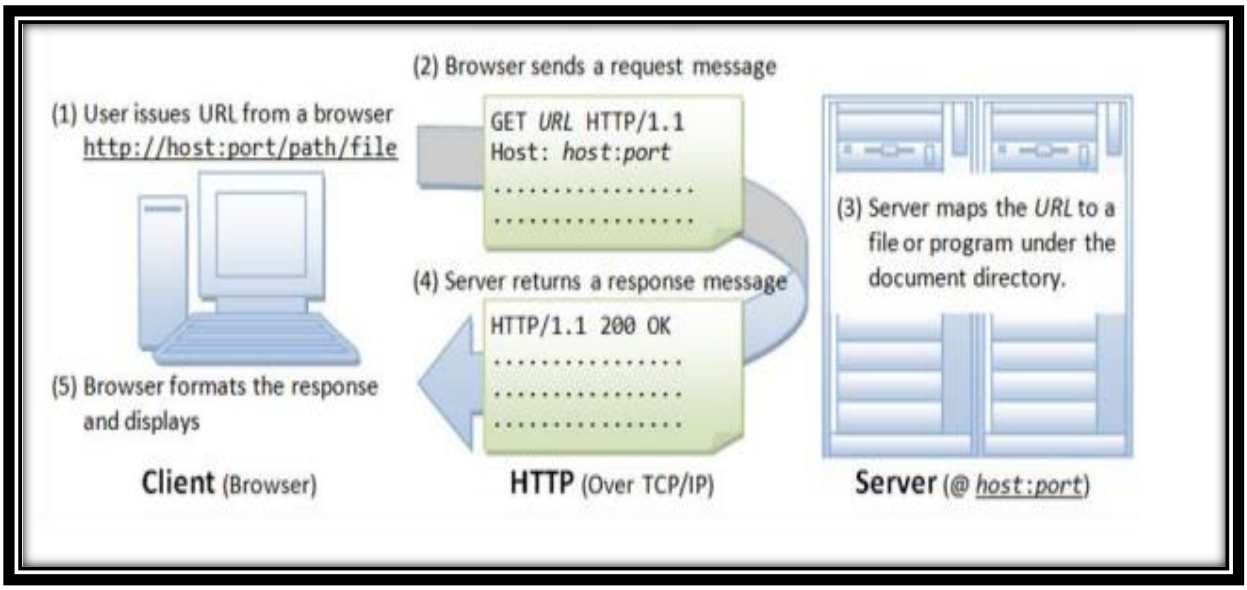
- **A web server is a computer that stores web server software and a website's component files (for example, HTML documents, images, CSS style sheets, and JavaScript files).**
- **A web server connects to the Internet and supports physical data interchange with other devices connected to the web**

- **IOT servers have a public address. The port for these servers can be either 80 for http or 443 for https.**
- **The user accesses the public page, through the IOT server which is connected to Wyliodrin as well as the board.**

Web Server Comprises with:

- **Software that implements HTTP/HTTPS**
- **Mainframes**
- **Computers**
- **Gadgets**

How does it Works:



1. **User Issues URL from a browser (Client)**
2. **Browser sends a request message (GET URL from PORT)**
3. **Server maps the URL to a file or program under the document directory (Server)**
4. **Server returns a response message (HTTP over TCP/IP)**
5. **Browser formats the responses and displays (Client-Browser)**

- **The browser sends a request to the server who searches the demanded page and returns it to the browser for the user.**
- **The request will consist of information about the kind of browser that is used, about the computer or about the document requested.**
- **It will have a method, a URL, a query string and the upload body in case you want data to be sent to the server.**
- **The response will include the status, which tells the browser if the page was found or not (the errors among the 400s are about a not found page, 300 are redirections and 200s are confirmations of the page being found).**
- **The available methods in http are : Get, Post, Put and Delete.**

Get method needs no upload body. It will only ask for data from the server and send only the headers, the address, the URL.

Post sends important data to the server, which will be uploaded. Post has the role of modifying data on the server. The response of both these methods is the page and any additional information that was requested.

Put is similar to post, only that in the semantic way, this method only creates an object on the server.

Delete also plays a semantic part. It needs no upload body and it delete objects on the server. The same action can be performed however using get.

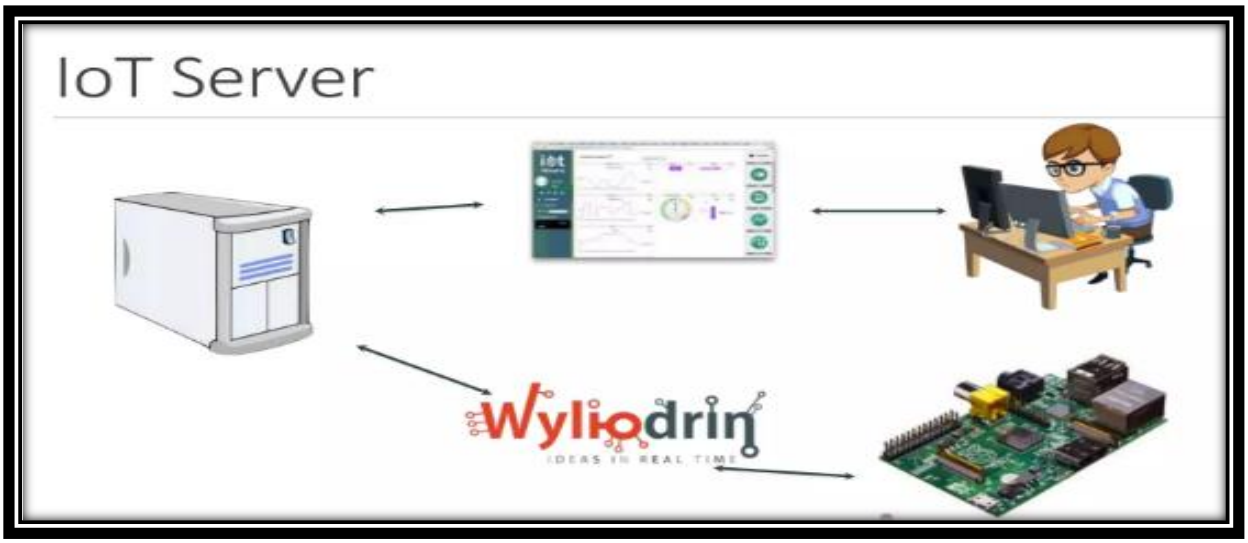
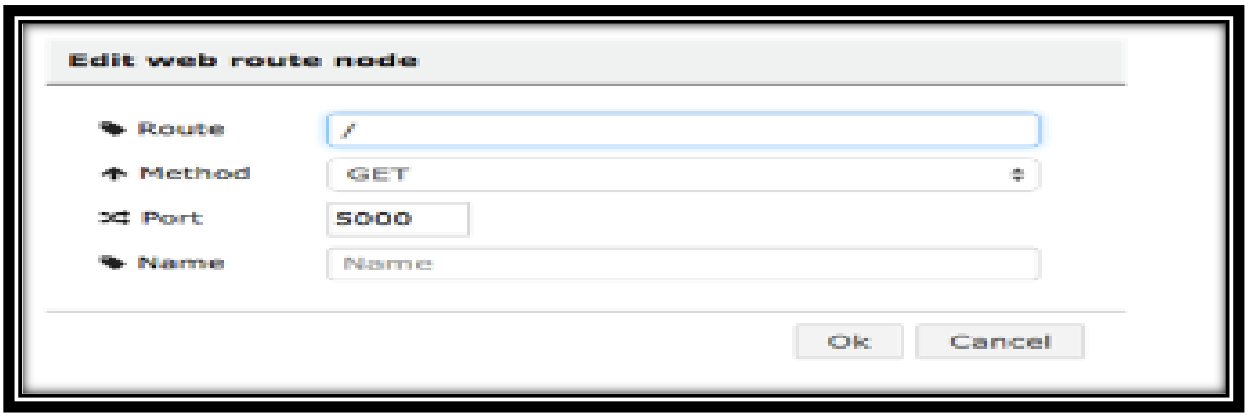


Fig 5: IoT Server in association with Wyliodrin

Wyliodrin

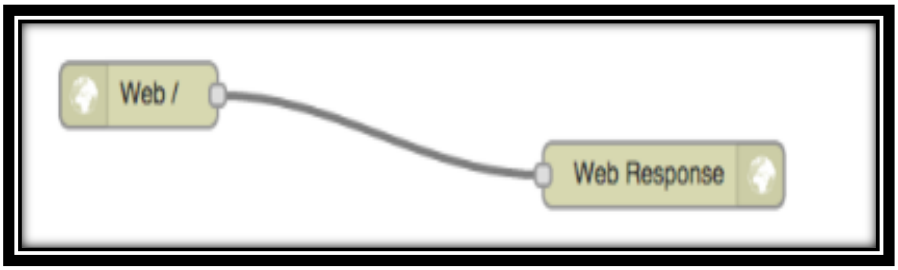
- Wyliodrin is a **custom IT solutions** development company founded in **2014**.
- Team has focused on delivering secure applications in fields such as the **Internet of things** and **automotive**.
- Wyliodrin offers a straightforward and extensive set of tools to ensure the fast prototyping and deployment of software and hardware solutions.
- We use **Rust**, **Tock OS** and **Android** to secure your devices and build safe and efficient systems.
- To build secure and scalable software solutions leveraging state of the art technologies.
- To create a web server in Wyliodrin you will need a ***web*** **node**.

Web nodes show the strength of relationships between values of two or more symbolic fields.



- **The *web response* node**: The message received by this node comes from one web node.
- For a web response the simplest way is to make a redirection.
- This means, in the redirect field, you can write the path to one of the static files and the browser will be sent to this page.
- On top of these, you will need the board's IP address which might not be public unless it is in the same network with the web server.

- **Payload:** Payload contains all attribute-value pairs that have the information about the detector model instance and the event triggered the action.
- The payload goes either in the query string for the get method or in the upload data for post.
- The message is built on this payload, on two mandatory variables: ***res*** which stands for the **response** and ***req*** which is the **request**. Without the last two, the server won't be able to provide a response.



- ***Web templates***: Just as for the static files, you will need a *templates* folder. This time, when you use the node, you don't need the whole path. You can only write the name of the file in the templates folder. What does the node do? It processes the response, meaning it loads the values plus the payload in it and sends it back to the browser. The values need to be in between two sets of curly brackets *{{}}*. Note that the values won't update unless the page is reloaded.

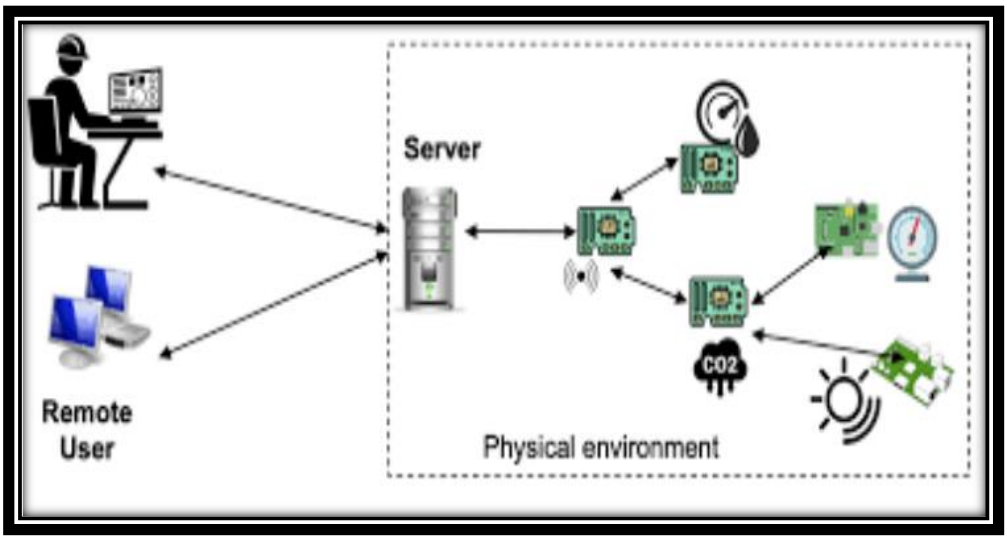
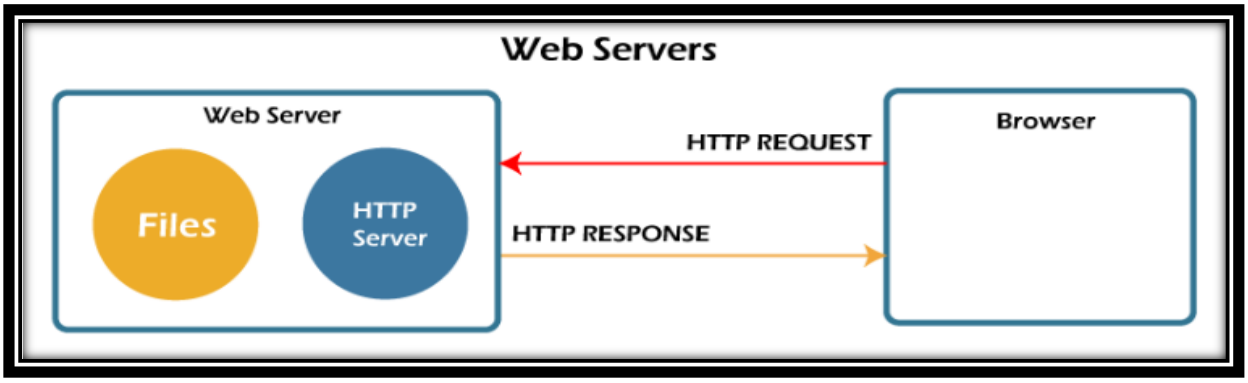


Fig 6. Web server for IoT

Web Servers

- **Web pages are a collection of data, including images, text files, hyperlinks, database files etc., all located on some computer (also known as server space) on the Internet.**
- **A web server is dedicated software that runs on the server-side.**

- When any user requests their web browser to run any web page, the web server places all the data materials together into an organized web page and forwards them back to the web browser with the help of the Internet.



- A web server is a dedicated computer responsible for running websites sitting out on those computers somewhere on the Internet.
- They are specialized programs that circulate web pages as summoned by the user.
- The primary objective of any web server is to collect, process and provide web pages to the users.
- This intercommunication of a web server with a web browser is done with the help of a protocol named HTTP (Hypertext Transfer Protocol).
- These stored web pages mostly use static content, containing HTML documents, images, style sheets, text files, etc.
- However, web servers can serve static as well as dynamic contents.
- Web Servers also assists in emailing services and storing files.
- Therefore it also uses SMTP (Simple Mail Transfer Protocol) and FTP (File Transfer Protocol) protocols to support the respective services.
- Web servers are mainly used in web hosting or hosting the website's data and running web-based applications.

How do web servers work

- The term web server can denote server hardware or server software, or in most cases, both hardware and software might be working together.

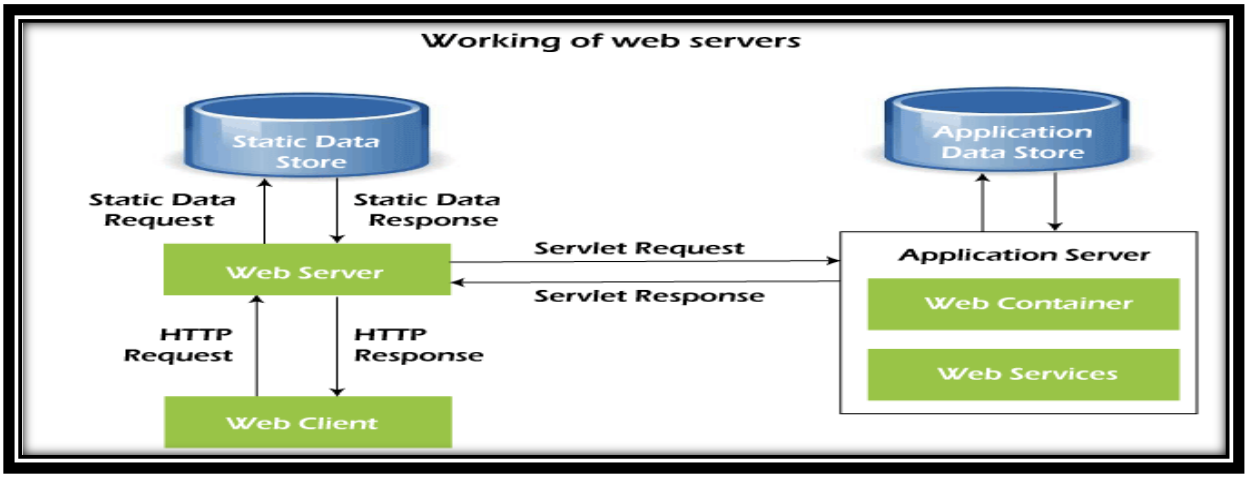


Fig 7: Working of Web servers

- ***On the hardware side***, **a web server is defined as a computer that stores software and another website raw data, such as HTML files, images, text documents, and JavaScript files.**
- **The hardware of the web servers are connected to the web and supports the data exchange with different devices connected to the Internet.**
- ***On the software side***, **a web server includes server software accessed through website domain names.**
- **It controls how web users access the web files and ensures the supply of website content to the end-user. The web server contains several components, including an HTTP server.**
- **Whenever any web browser, such as Google Chrome, Microsoft Edge or Firefox, requests for a web page hosted on a web server, the browser will process the request forward with the help of HTTP.**
- **At the server end, when it receives the request, the HTTP server will accept the request and immediately start looking for the requested data and forwards it back to the web browser via HTTP.**
-

Step-by-Step process of what happens whenever a web browser approaches the web server and requests a web file or file.

1. **First, any web user is required to type the URL of the web page in the address bar of your web browser.**
2. **With the help of the URL, your web browser will fetch the IP address of your domain name either by converting the URL via DNS (Domain Name System) or by looking for the IP in cache memory. The IP address will direct your browser to the web server.**
3. **After making the connection, the web browser will request for the web page from the web server with the help of an HTTP request.**
4. **As soon as the web server receives this request, it immediately responds by sending back the requested page or file to the web browser HTTP.**
5. **If the web page requested by the browser does not exist or if there occurs some error in the process, the web server will return an error message.**
6. **If there occurs no error, the browser will successfully display the webpage.**

Web Services

- **A Web service is a method of communication between two electronic devices over a network.**
- **It is a software function provided at a network address over the Web with the service always-on as in the concept of utility computing.**
- **Many organizations use multiple software systems for management**

jQuery

- **jQuery is a fast, small, and feature-rich JavaScript library.**
- **It makes things like HTML document traversal and manipulation, event handling, animation.**
- **jQuery can make function calls to the server.**

Web sockets

- **WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection.**

- **The WebSocket protocol was standardized by the IETF as RFC 6455 in 2011.**
- **A web socket is based on the http or https protocol.**
- **It builds a connection between the browser and the server, so that either one can send data.**
- **When the browser makes a request, the server recognizes the socket and doesn't close the connection.**
- **The two parties send the packages they need to send.**
- **If the server does not know how to work with sockets, the socket i/o will go back to querying.**

AngularJS

- **AngularJS is a toolset for building the framework most suited to your application development.**
- **AngularJS is a library through which you can build browser applications.**
- **AngularJS is distributed as a JavaScript file.**
- **It can be added to an HTML page with a <script> tag.**
- **AngularJS extends HTML attributes with Directives, and binds data to HTML with Expressions.**

Web server uses

- **Sending and receiving mails on Internet by using SMTP (Simple Mail transfer Protocol);**
- **Fetching requests for File Transfer Protocol (FTP) files; and**
- **Designing, developing, and publishing websites.**

Note: The server-side scripting process will have various scripting languages such ASP, PHP, Java, JavaScript, Python, ruby and many more.

Web Servers are of 2 Types

1. **Static Web servers** 2. **Dynamic Web servers**

Differences

Sno	Static	Dynamic
1	**Static web servers refer to the servers, which serve only the static content i.e., the content is fixed and being shown as it is.**	**Dynamic web servers refer to the servers where the content of the page can be updated and altered**
2	**A static web server includes a computer and the HTTP (Hyper Text Transfer Protocol) software.**	**A dynamic web server also includes a computer with plenty of other software, unlike an application server and database model.**
3	**It is called static; the web pages content won't change unless the user manually changes it, and the server will deliver web files as is to the web browser.**	**It is called dynamic because the application server is used to update the web pages files at the server-side, and due to which, it can change on every call requested by the web browser.**
4	**Static web servers take less time to load the data.**	**The Dynamic web server can only produce the data when it is requested from the database. Therefore, it is time consuming and more complicated when compared to static web servers.**

Web server software available in the market

1	**Apache HTTP Server** • **This web server is developed by Apache Software Foundation.** • **It is an open-source, accessible web server available for almost all operating systems, including Windows, MACOS, Linus, FreeBSD, etc.** • **Apache is one of the most popular web servers used around the globe.**
2	**Microsoft Internet Information Services (IIS)** • **IIS is a high-performance web server that is developed by Microsoft only for Microsoft platforms.** • **This webs server is tightly integrated with Microsoft operating system; therefore, it is not open-sourced.**
3	**Nginx** • **Nginx is an open-source web server commonly used by administrators as it supports light resource application and scalability.**
4	**Lighttpd** • **Lighttpd, also known as lighty, is a free, open-source web server with the FreeBSD operating system.** • **This web server is fast, secure and consumes much less CPU power.** • **It can also run on the commonly used operating system, unlike Windows, Mac OS X, Linus.**
5	**Sun Java System Web Server** • **Sun Java is a free web server developed by Sun Microsystems well equipped for a medium and large website that can run on Windows, Linux and Unix.** • **Moreover, this web server supports several languages, scripts and technologies essential for Web 2.0, unlike JSP, Java Serve lets, PHP, Python, HTML, etc.** • **Though Sun Java is free, it is not an open-source web server**

Web server security Methods:

- **A reverse proxy is a proxy server that is accessible to the clients, therefore hiding the internal server. It acts as an intermediary as wherever any user makes requests to the web server for any data or file, the proxy server seizes those requests and then communicates with the web server.**
- **Access restriction is a technique that limits the web host's access to infrastructure machines or using Secure Socket Shell (SSH);**
- **Keeping web servers mended and updated, as it benefits to ensure the web server isn't vulnerable to exposures;**
- **Network monitoring is a security practice that ensures that no unauthorized activity is going on the web server; and**
- **Using a firewall and SSL safeguards the web server as firewalls can supervise HTTP request traffic while a Secure Sockets Layer (SSL) supports securing the data.**

End