



5th

VOLUME
ISSUE - 2

Wall For All

Dear Readers

The nostalgic feeling that one experiences while sifting through the dusty old pages of the college magazine cannot be expressed in words. However, very few of us have retained those copies, and most of those precious articles that we wrote during those golden days with enthusiasm are lost forever. With the advent of e-books and other online media, the days of paper-bound college magazines are gone, and the digital platform has paved way to allow retention of such publications without much effort.

Wall-for-All, the e-Magazine published by the Department of Computer Applications, is one such effort that was started with an intent to provide a chance to all students and faculty members to share their thoughts and knowledge, and hone their skills in creative writing.

I am happy to see the enthusiasm of eminent members of the department to contribute to ***Wall-for-All***. This shows the positive and creative energy of the contributors. However, it would be really wonderful if we can see the articles contributed by the students in the next editions, for this e-Magazine is intended to be a writing pad for each member of the department.

I proudly present the Fifth edition of Wall for All

MVC - The Real-world Architecture for Application Development

Mr. Preetinder Singh Brar

Associate Professor

Department of Computer Applications

Chitkara University, Punjab campus

Software development is one of the key activities of any professional who graduates from a university after completion of any computer science/applications degree. However, as the syllabus prescribed by most of the universities usually talk of the conventional software development life cycle models (SDLC) such as the Waterfall model or the incremental development, most students are usually not aware of the software architectures in practice in the real-world.

Ever since the advent of software development as a specialization, the world has witnessed numerous paradigm shifts in the way software is actually developed and maintained. This article intends to discuss about the MVC pattern, which is the latest and the most prominent coding paradigm and has been adopted worldwide for developing software or web applications. The MVC, or the Model–View–Controller, is a software architectural pattern that is widely used for implementing user interfaces due to suitability for architecting web applications. Putting in simple words, the MVC architecture segregates the application logic into three separate fragments so as to improve modularity, ease of collaboration and code reuse. It also makes the applications easier to test, debug and to maintain.

MVC... right! But how does the MVC architecture work? Well, as already mentioned above, the MVC architecture separates the input, processing, and output units of an application. This model is divided into following three interconnected parts

- Model – Data and business logic
- View – User Interface
- Controller – Request handler

The figure given below depicts the MVC architecture

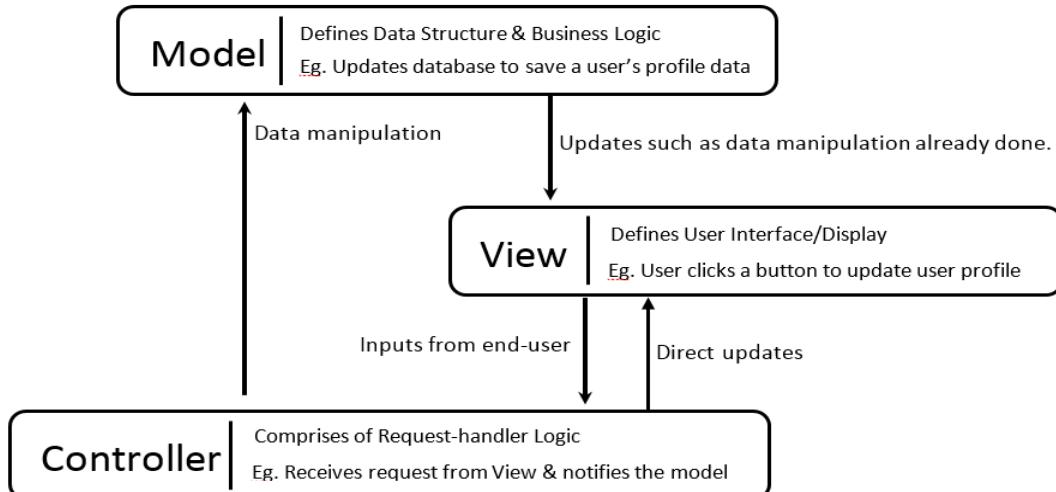


Fig.: The Model-View-Controller (MVC) Architecture

All of these three components are built to handle some specific development aspects of any web or software application. The *model* represents all data that is stored pertaining to the application. It is the model that defines as to what data should be contained in the app. The *controller* receives all requests from the user through the *view*. Various objects of forms such as the buttons, text boxes, etc. and other information visible to the user on the web, are all part of view. In simple words, the view can be considered to be a user interface. The *controller* then instructs the *model* to retrieve and organize any information, which is then sent to the *view* to render the final output. The controller comprises the logic that can update the model and/or view in response to input from the users of the app.

For systems implemented through the MVC architecture, the users requests for data/information from a specific web address by typing the URL in the browser's address bar. This request reaches the server and a call to appropriate controller is made. The controller employs appropriate View & Model to create the final response and transmits it back to the user.

The MVC architecture separates the state of an object (model), the manner in which the object is displayed to the user (view), and the manner in which the object's state is updated (controller). By separation of these three perspectives, it becomes easy to define the GUI components that are equivalent in terms of information state (model), but are displayed (view) and respond to the user (controller). Because of logical separation of the model from the view and its controller, it becomes possible to implement logically equivalent objects, such as the buttons, but with different look and feel. In case the user desires to just update the view to display the data in a different format, such as displaying the same content in alphabetical, or perhaps chronological order, then the controller can handle this request directly without needing to update the model.

Having gone through the architecture, let us discuss about the advantages of MVC. It is those benefits that are responsible for making this architecture the most preferred choice of design.

1. Supports Rapid Application Development (RAD)

MVC supports rapid and parallel development. MVC makes it possible for one programmer to work on the view while other programmers are working on controller logic. The development of the three components is usually done independently, with just the knowledge of the data values/fields to be worked upon.

2. Multiple Views

Depending on the different user types and the rights assigned to them, multiple views for a model can be created. Since MVC separates data and business logic from the display (view), so there is limited code duplication.

3. Updates to one module do not affect other modules

With demand for updating, and even overhauling of user interface every few years, the web applications require frequent modifications so as to implement the latest business rules, and also to bring the best of the UX to the end-users. However, bringing about such changes in the views usually does not affect the model or the controller.

4. Supports JavaScript/jQuery

MVC supports integration of scripting languages like JavaScript and the JavaScript libraries can be integrated with MVC to develop feature rich web applications.

In addition to the above mentioned advantages, programmers also reap the benefits if producing SEO-friendly web applications using MVC architecture.

In the early days of the world wide web, the MVC architecture was mostly implemented on the server-side and the clients used to place requests for the updates through forms or links. Those updates were implemented on the data (model), and updated details were received back at the client-side for displaying in the browser(view) through a new webpage or by refreshing the webpage.

The MVC pattern is an excellent architecture that is accepted throughout the world, and the selection of programming language has no bearing on the same. The capability of MVC to manage multiple views makes it the best architecture pattern for developing web applications. Thus, MVC design pattern is certainly a great approach to building software applications.

For more information on MVC architecture, you may visit following web links

<https://www.ongraph.com/what-is-mvc-architecture-in-a-web-based-application/>

https://developer.mozilla.org/en-US/docs/Web/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture

Micro-Service Architecture

Dr. Ruchi Mittal

Professor

Department of Computer Applications

Chitkara University, Punjab campus

Introduction

Micro-service is a design pattern within Service-oriented architecture (SOA). In this type of architecture, a big system is divided into a number of smaller independent service units. Micro-service is the process of dividing the entire application into small, meaningful functional features working as individual services where each service shall only serve a single business need. Each service has its own process and communicates with other services. These services are lightweight and are deployed on single or multiple servers. Most of the large scale e-commerce organizations including Amazon, Twitter, Netflix, and eBay use micro services architecture.

Need of Micro-service architecture

With micro service architecture, each application runs in its own container. Time and resources are not wasted for managing the effects of updating one application (as all applications are independent of each other). Thus, micro services help in achieving multiple goals such as (1) helps in continuous delivery; (2) easy to adapt to latest market trends; (3) increased development velocity and (4) reduced costs.

Micro-service vs Monolithic architecture

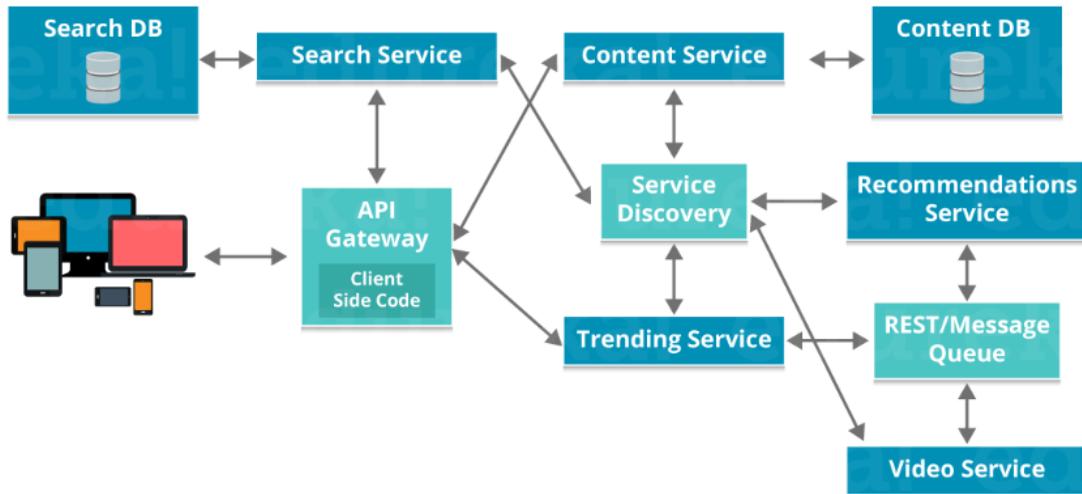
In monolithic architecture, we have big cumbersome applications with all modules coupled tightly inside a single executable file which is typically deployed on a web or application server.

On the other hand, in micro service architecture, application components are standalone applications of their own. These independent application components talk to each other using different method such as RMI (Remote Method Invocation), Restful Web Services or Push Messaging.

Differentiation Parameters	Monolithic architecture	Micro service architecture
Ease of release	No frequent and easy releases.	Releases are easy, small, and independent.
Continuous delivery	Problem in continuous delivery.	Small, independent components help in continuous delivery.
Project management	Difficult to manage big team and project.	Easy to manage components and teams.
Scalability and performance	Expensive scalability & performance.	Better performance and easy to scale individual components.
Technology diversity	Lack of technology diversity.	Different technologies can be used in different components.
Components replacement	Not easy to replace components.	Faulty components are easily replaceable without affecting the whole system.

Micro-service communication

At first, each of the micro service registers itself. Micro services use service discovery pattern which acts as a guide and helps in finding the route of communication between each of them. Micro services then communicate with each other by using either HTTP Requests or RMI or Message Bus. Next, they deploy the static content to cloud based storage service that can deliver it directly to the clients via CDNs.



Communication between microservices.

Micro-service drawbacks

- Non uniform application due to technology diversity in individual components.
- Dev-Ops complexity.
- Increased resource usage.
- Overhead in team communication.
- Prior formal documentation needed.
- Marshalling and un-marshalling during individual component communication.
- Network security required.
- High upfront cost.
- Increase in network communication resources.

Microservices Tools

Wiremock: Testing Micro services

WireMock is a flexible library for mocking and stubbing the behavior of web services. It allows us to configure the response returned by the HTTP API upon receiving request.

Docker

Docker is open source project. It allows us to create, deploy, and run applications by using containers. Developers can run an application as a single package along with all the dependencies.

Hystrix

Hystrix is a fault tolerance java library. This tool is designed to separate points of access to remote services, systems, and 3rd-party libraries in a distributed environment like micro services.

Node.js: A Revolutionary Asynchronous Framework

Mr. Rajesh Kaushal

Associate Professor

Department of Computer Applications

Chitkara University, Punjab

Mr. Naveen Kumar

Assistant Professor

Department of Computer Applications

Chitkara University, Punjab campus

These days JavaScript language is highly in demand. Initially used as a client based language, it has now evolved with a more broad based usage and is now considered as a high-level programming language. A wide range of JavaScript frameworks are available and extensively used by IT companies. Some of the popular JavaScript frameworks are listed below:

React.js - A JavaScript framework used for designing front end/UI. It is used by big organizations such as Facebook.

Angular.js- A JavaScript framework for front-end/UI. A prominent user of this technology is Google.

Node.js - It is an asynchronous server side framework and is commonly used for handling user requests.

Although there are many more frameworks but the ones listed above are much more popular. The beauty of node framework is that it can handle millions of user request smoothly. One common misconception about node is that people consider it as a programming language. The reality is that it is just a framework that is build upon JavaScript Language and *Google V8 engine*. Ryan Dahl decided to build this framework on top of *GoogleV8* engine as it was much faster than other available engines like *chakra engine* of Microsoft and *spider-monkey engine* of Mozilla Firefox.

The node framework is asynchronous in nature as it does not wait for any operation. It immediately executes the code which has no dependencies. The code which has dependency on a web API is pushed to API for resolution. The results of API are stored into the queue and then to the stack. Finally, stack is used to send responses back to the front-end. This framework is surprisingly handling massive requests per seconds even though it uses a single thread for request and response. Due to single threaded framework it is unlikely to occur deadlock situation.

The another key feature of this framework is that it also supports *MVC* (Model, View and Controller Architecture). The controller consists of various routes and their corresponding

handlers. The model holds the database which is commonly a *mongo DB* database. The other databases are also supported but *mongo DB* is primarily used with node framework. The view points to the front-end interface which may change according to user requests. The bottom line is that it is a powerful framework for handling massive requests.

The asynchronous execution of node entirely depends on the event loop mechanism. This mechanism is explained in the Fig. 1. Assume there are three operations in a code namely operation-1, operation-2 and operation-3. Initially all operations will be loaded into the stack. Assume that operation-1 is simply a text message that needs to be displayed on the front-end, operation-2 is a request for online transaction and operation-3 is a copyright message that usually appears at the bottom of web-pages. As, operation-1 has no dependency on any other API it will be executed immediately and results will be stored first into the queue and then to the stack and finally to the user interface. The operation-2 is a request for online transaction and hence dependent on the payment API for completion. So, this operation will be forwarded to Web-API for completion. The node doesn't wait for the payment gateway to return the response. Rather, it will execute the operation-3 also and store the results into the queue. Later on when the payment confirmation is received, the confirmation response will be stored into the queue and then to the user interface.

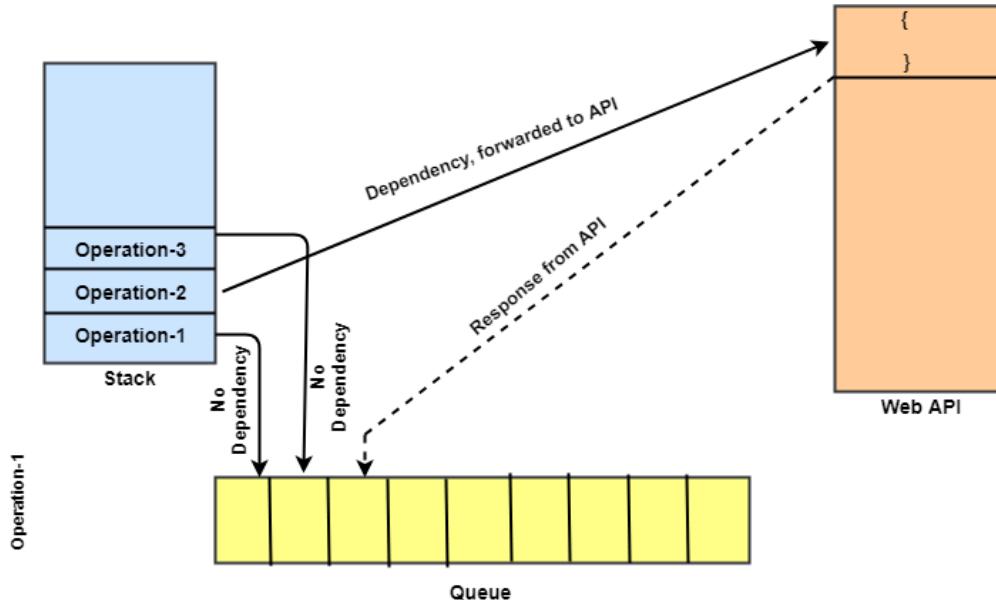


Fig-1 Event Loop Mechanism of NodeJs

Disruption and Destruction in the Digital Economy

Mr. Vikas Rattan

Associate Professor

Department of Computer Applications

Chitkara University, Punjab campus

Countries, and organizations, that are leveraging digital technologies are moving way far ahead of other countries, and organizations. The US and several European countries such as Switzerland, The Netherlands, Sweden and Germany are considered the world's most innovative countries mainly due to the way they implement digital technologies and promote a culture of creativity; similarly – in a short span of time - Facebook, Google and Amazon, with their platform-based models and heavy investments in R&D, have raced far ahead of competition with a lead that is considered almost unassailable. According to the World Economic Forum (WEF), the creation and sustenance of an innovation ecosystem depends on the following five factors: commercialization, interaction & diversity, administrative requirements, research & development, and entrepreneurial culture. For both countries, and companies, innovation is the bedrock of competitiveness; and in order to survive in this digital world, it is important to embrace new technologies. For a country like India, the most effective way to reduce poverty and sustain socio-economic growth is by introducing a culture of innovation especially at the elementary school level and then ensuring it moves up to primary, secondary and the tertiary level. Noteworthy, innovation in technology is dependent on the level of digital skills within the population. As a consequence, it should come as no surprise that there is a strong and positive correlation between education and skills which also influences the usage of digital technologies in various spheres of life. Digitization should no longer remain a choice but a mandate. The future is to adopt disruptive technologies and destroy existing outdated systems. Innovation doesn't happen in a vacuum, but requires openness and interactions between systems and their environments. It's time we all come together and share the new responsibilities emerging in this new digital age.

Security in MANETs

Ms. Neelam Rani

Assistant Professor

Department of Computer Applications

Chitkara University, Punjab campus

Introduction

Security is always needed in MANETs which can be described as an autonomous system of mobile routers connected by wireless links – the union of which forms an arbitrary graph. There are various security issues and different mechanisms for providing the security. One of these mechanisms will be used to secure the routing algorithm in MANETs.

Need of security in MANETs

The use of wireless links makes MANETs susceptible to attack. Eavesdropper can access secret information, thus violating network confidentiality. The fundamental aspects of computer security - confidentiality, integrity, authentication and non – repudiation are valid also when the protection of routing in MANETs networks. Confidentiality of routing information is important so that not only the payload data, but also the routing message header carrying e.g. the location information of MANET can be exchanged securely. Non-repudiation means that the node cannot deny having sent or handled certain piece of routing information in the past. Authentication mechanisms allows partial non-repudiation, but typically means like time stamping service are also required to protect the routing traffic from tempering attack like replaying or delaying of routing messages. Availability is the central issue in MANET system due to the dynamic and unpredictable condition in MANET environments: the node may not be available for communication all the time.

Attack in Ad-hoc Routing Protocols:

We know focus on attacks against the routing protocol in ad-hoc networks. These attacks may have the aim of modifying the routing protocol so that traffic flows through the specific node controlled by the attacker.

Attack Modifying Route Request Packets

In this type of attack a hacker tries to announce itself having shorter routes to the destination. The shorter route can either be identified by number of hops or announcing a better route metric in the reply packets for the route discovery phase to the sender. Similarly an intruder can become a part of the route and start discarding traffic by employing DOS (Denial of service) attack for the packet received from the sender.

Attack Using Spoofing Technique

The attackers just start spoofing the valid IP addresses and isolate the nodes from remainder of the network. This vulnerability is easily exploitable in AODV (adhoc on demand distance vector routing) and DSR (dynamic source routing) protocol.

Attack Using Fabrication

The attacker can intentionally float error messages on the network thus falsifying existence of valid routes. An attacker could mount a replay attack by advertising stale routes and attacker can even advertise a zero metric for all destinations causing all nodes to route packets to it and thus creating a black hole.

Attack Relaying Incorrect Traffic

In this type of attack, network communications coming for legitimate protocol - complaint nodes may be populated by misbehaving nodes.

Basic Security Mechanisms:

The various types of mechanisms that can be used for providing the security to the routing algorithms in MANETS are (a) Cryptography (b) Cryptographic hash functions (c) Digital Signatures.

Cryptography

Cryptography is the heart of security. If we need to create privacy, we need to encrypt our message at the sender site and decrypt it at the receiver site. The original message, before transformed, is called plaintext. After the message is transformed, it is called cipher text. An encryption algorithm transforms the plaintext to cipher text; a decryption algorithm transforms the cipher text back to the plaintext.

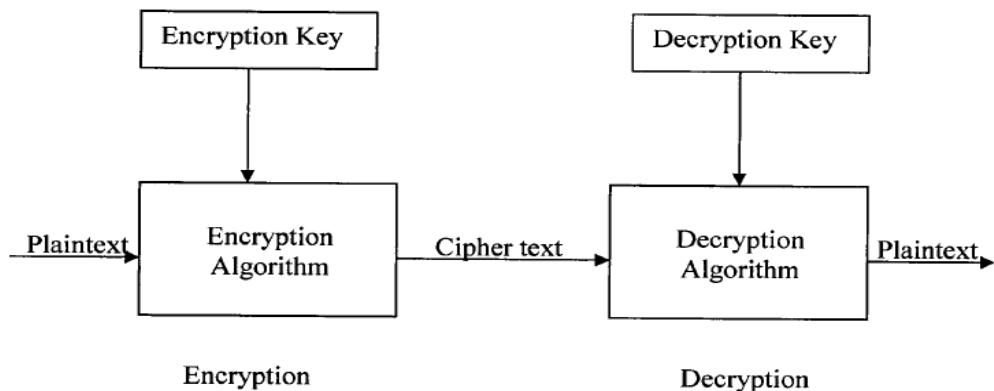


Fig. Encryption and decryption

Cryptographic Algorithms can be divided into two groups:

Symmetric-Key Cryptography.

Public-Key Cryptography.

Cryptographic Hash Functions

For secure communication it is required that data transmitted is not altered by any entity. Hash functions are the security primitives that ensure data integrity. Hash function is often called one-way hash function, because it is difficult to compute the inverse function. For example, the cube function $y=x^3$ it is quite easy to compute y given x . But the inverse function, $3xy$ is much complicated to compute. It is also used for entity authentication . With digital signature hash function is applied to the whole message. Then the hashed value is signed. On receiving, the hash value is recomputed and verified that the received signature is unaltered from the original source.

Digital Signatures

Digital signature is an important cryptographic which is used for authentication, authorization and non-repudiation. Digital signature has the best use of public key cryptography. An asymmetric encryption algorithm such as RSA can be used to create and verify digital signature.