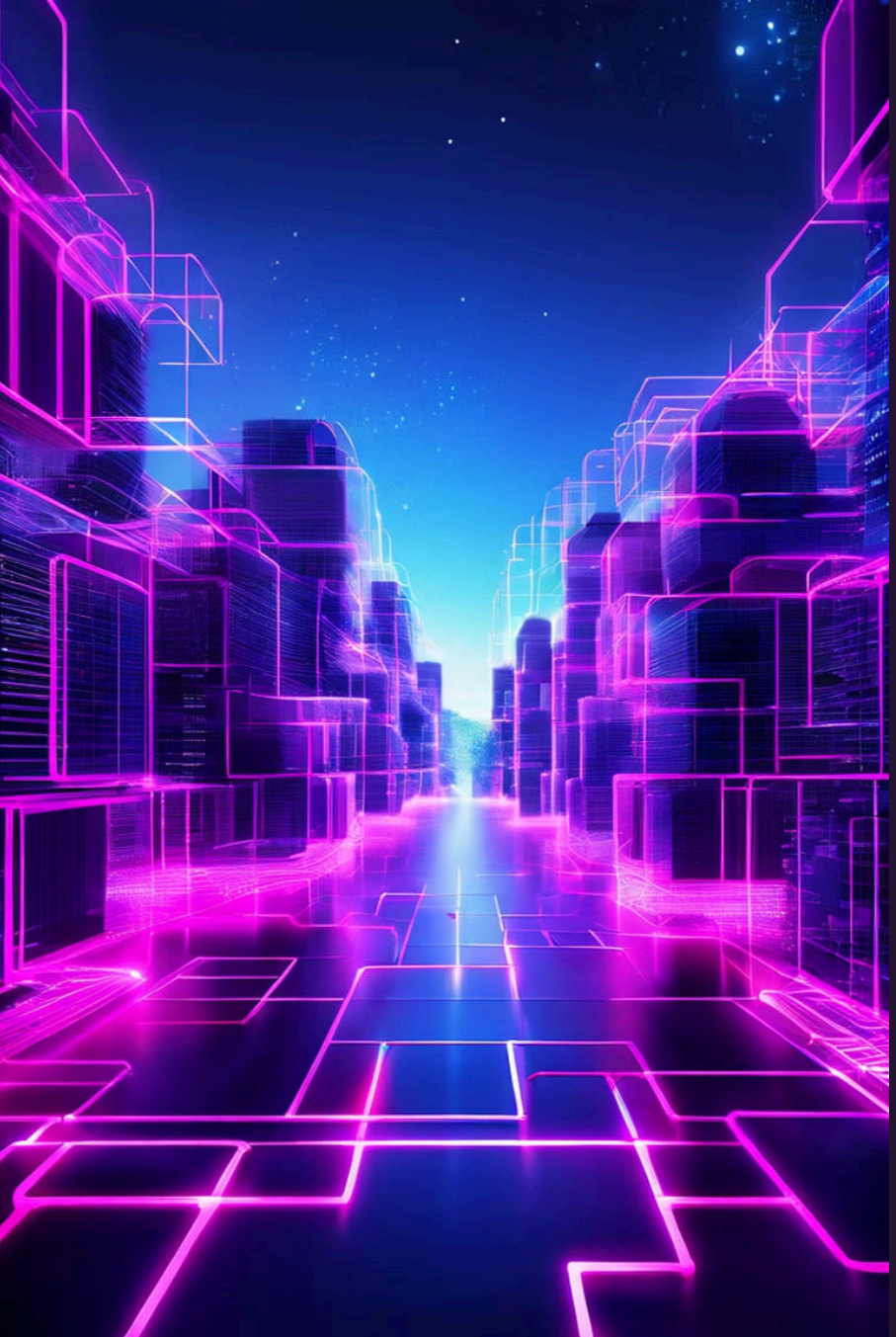


Introduction to Module Federation

Module Federation is a powerful new technology that revolutionizes how web applications are built and deployed. It allows developers to create modular, scalable, and highly dynamic web applications by enabling the collaborative development and seamless integration of independent application modules. This presentation will provide a comprehensive overview of Module Federation, its key benefits, and practical guidance on setting it up within an Angular application.

S by SAGAR DEWAN .

What is Module Federation?

Decentralized Application Architecture

Module Federation is a novel approach to web application architecture that moves away from the traditional monolithic structure. It enables the creation of decentralized, modular applications where individual components or "modules" can be developed, deployed, and versioned independently.

Dynamic Module Loading

At runtime, Module Federation allows these independent modules to be dynamically loaded and shared between different applications, known as "hosts" and "remotes". This enables the creation of highly flexible and composable web experiences.

Seamless Integration

By facilitating the seamless integration of modules, Module Federation empowers developers to build larger, more complex applications by composing them from smaller, more manageable pieces. This promotes better code reuse, faster development cycles, and improved maintainability.

Benefits of Module Federation

1 Reduced Bundle Size

Module Federation helps optimize bundle sizes by allowing applications to only load the modules they need, rather than bundling all dependencies upfront. This leads to faster initial load times and improved user experience.

2 Faster Development

By enabling the independent development and deployment of modules, Module Federation promotes a more agile development process. Teams can work on different parts of the application concurrently, leading to faster time-to-market.

3 Improved Scalability

Module Federation's decentralized architecture allows applications to scale more effectively, as individual modules can be scaled independently without affecting the entire system. This enhances the overall scalability and performance of the application.

4 Increased Flexibility

The ability to dynamically load and compose modules at runtime gives developers greater flexibility in designing and evolving their applications. This enables more innovative user experiences and streamlined updates.

Setting up Module Federation in Angular

1

Install Dependencies

Begin by installing the necessary dependencies for Module Federation in your Angular application, including the `@angular-architects/module-federation` package and the required Webpack plugins.

2

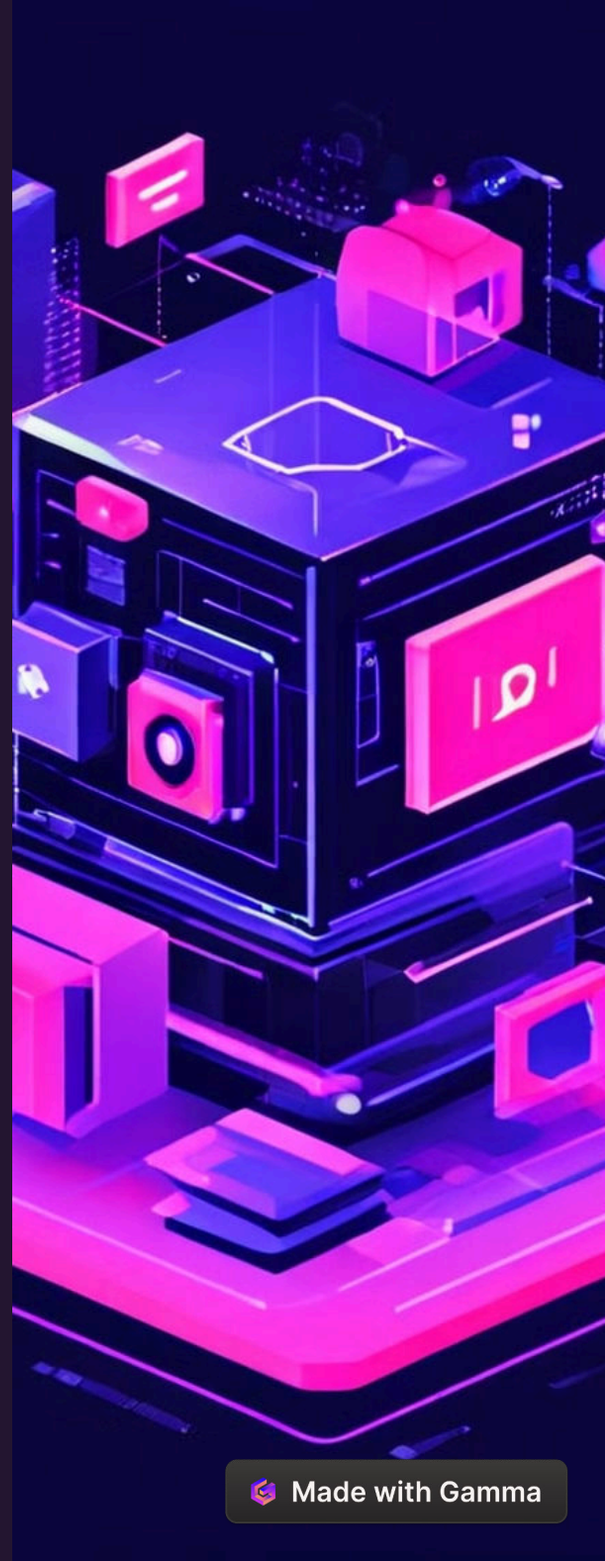
Configure Webpack

Next, you'll need to configure Webpack to enable Module Federation. This involves defining the application's role (host or remote), specifying the shared dependencies, and setting up the module remoting and loading logic.

3

Integrate into Angular

Finally, integrate the Module Federation configuration into your Angular application by updating the AppModule, routing, and any other necessary components. This will allow your application to seamlessly load and consume remote modules at runtime.



Configuring Remotes and Hosts

Defining Remotes

In a Module Federation setup, "remote" applications are those that expose their modules for consumption by other applications. The remote configuration specifies which modules should be made available, as well as how they should be accessed and loaded by the host applications.

Configuring Hosts

On the other hand, "host" applications are those that consume and integrate remote modules into their own application. The host configuration defines which remote modules should be loaded, as well as how they should be mapped to the host application's own module structure.

Dynamic Remoting

Module Federation also enables dynamic remoting, where the set of available remote modules can change at runtime. This allows for greater flexibility and adaptability in how applications are composed and updated over time.

Sharing Dependencies

1 2 3

Identifying Shared Dependencies

A key aspect of Module Federation is the ability to share dependencies between host and remote applications. This involves identifying the common libraries, frameworks, and other dependencies that can be shared to optimize bundle sizes and improve performance.

Configuring Shared Scope

The shared dependencies are configured in the Module Federation setup, where the "shared scope" determines how these dependencies are loaded and accessed by the host and remote applications. This ensures that there is a single, consistent version of each shared dependency throughout the application ecosystem.

Runtime Optimization

At runtime, Module Federation's dynamic loading capabilities ensure that shared dependencies are only loaded once, even if multiple modules or applications require them. This runtime optimization helps further reduce bundle sizes and improve application performance.

Lazy Loading Federated Modules

Faster Initial Load

By lazy loading federated modules on-demand, Module Federation helps reduce the initial bundle size and load time of the host application. This improves the user experience and overall performance of the application.

Modular Code Structure

The ability to lazy load federated modules aligns well with Angular's modular architecture, allowing developers to create more maintainable and scalable code structures within their applications.

Seamless User Experience

Lazy loading federated modules ensures that users only download the content they need, when they need it. This results in a more responsive and seamless user experience, as the application only loads the necessary components on-the-fly.

Efficient Memory Usage

By loading modules on-demand, Module Federation helps optimize memory usage, as the host application only needs to keep the currently active modules in memory, rather than loading all modules upfront.

Conclusion and Key Takeaways

1 Modular and Scalable Applications

Module Federation revolutionizes web application development by enabling the creation of modular, scalable, and highly dynamic applications. It promotes better code reuse, faster development cycles, and improved maintainability.

2 Empowered Developer Collaboration

By facilitating the independent development and seamless integration of application modules, Module Federation fosters greater collaboration among development teams. This leads to more innovative and user-centric web experiences.

3 Optimized Performance and Efficiency

Module Federation's dynamic loading capabilities, shared dependencies, and lazy loading features all contribute to improved application performance, reduced bundle sizes, and efficient resource utilization.

4 Futureproof Architecture

As the web evolves, Module Federation provides a futureproof architecture that enables applications to adapt and scale more effectively. It sets the stage for the next generation of highly modular, composable, and resilient web applications.