

Introduction to MEAN Stack

The MEAN stack is a popular full-stack JavaScript framework for building modern, dynamic web applications. It consists of four key components: MongoDB, Express.js, AngularJS, and Node.js.

 by Chantri Polprasert

What is MEAN Stack?

1 Full-Stack JavaScript

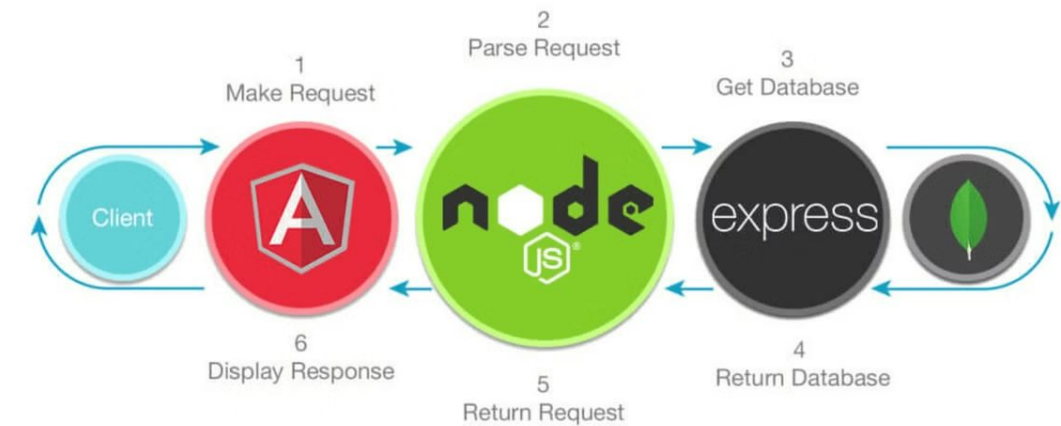
The MEAN stack uses JavaScript across the entire application, from the database to the front-end, which simplifies development.

2 Open-Source Technologies

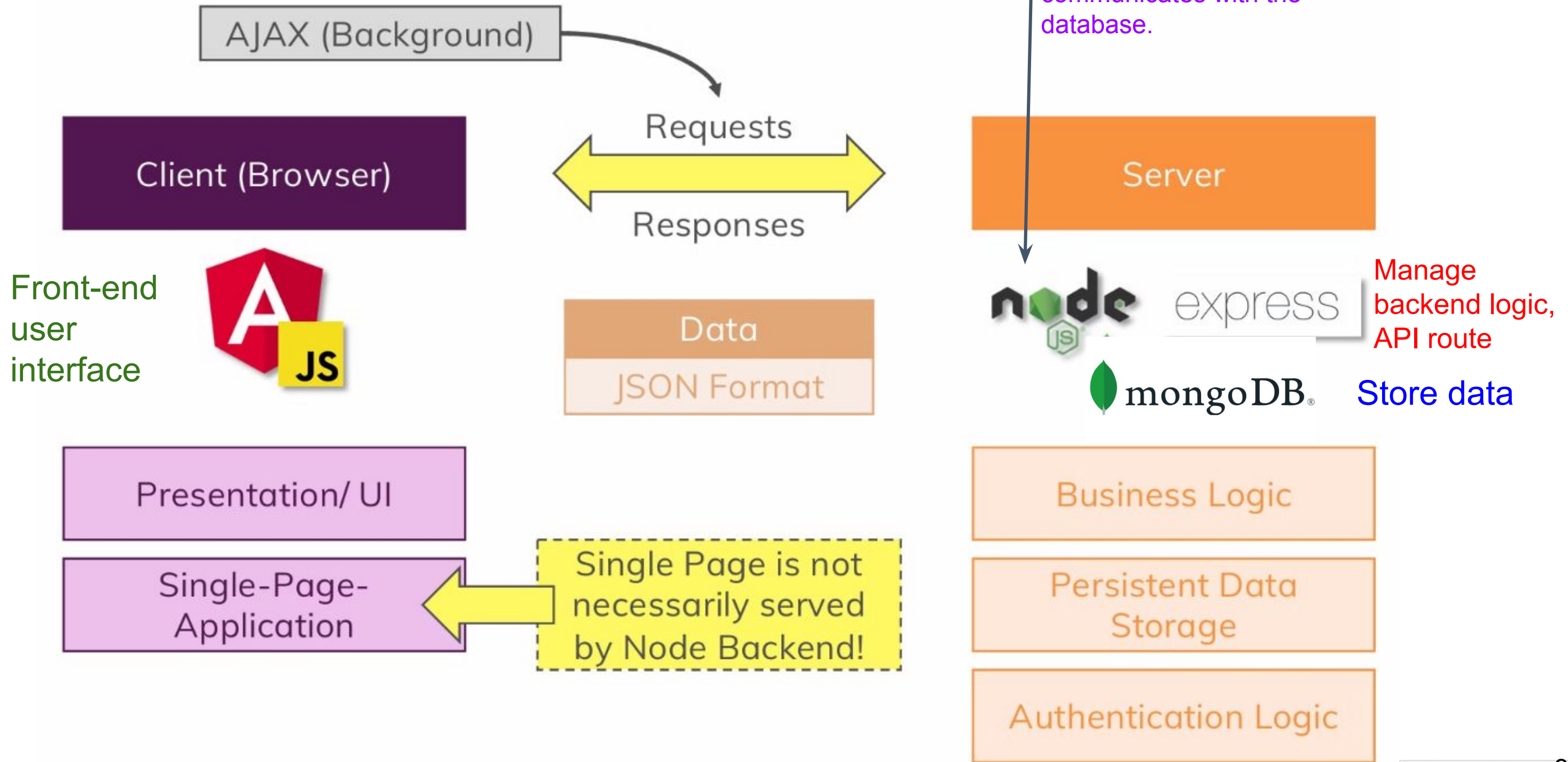
All the MEAN stack components are open-source, free to use, and have large, active developer communities.

3 Flexibility and Scalability

The modular and extensible nature of the MEAN stack allows developers to build scalable and flexible applications.



Big Picture



Why Use the MEAN Stack?

- **Full JavaScript Stack:** Unified language across client and server.
- **Open-Source:** Free and supported by large developer communities.
- **Scalable:** Works well for both small projects and large-scale applications
 - MongoDB: NoSQL database built for scalability
 - Express.js: Lightweight and Modular Backend
 - Angular: Scalable Front-end framework (Component-based)
 - Node.js: Event-driven model
- **RESTful API Support:** Easily build REST APIs with Express.
 - Restful API are designed around resources (users, products,..) with standard operations (CRUD) mapped to HTTP methods (GET, POST,..)
- **Single Page Applications (SPA):** Angular is great for developing dynamic SPAs.

Components of MEAN Stack



MongoDB (M)

A NoSQL database that stores data in flexible, JSON-like documents.



Express.js (E)

A minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.



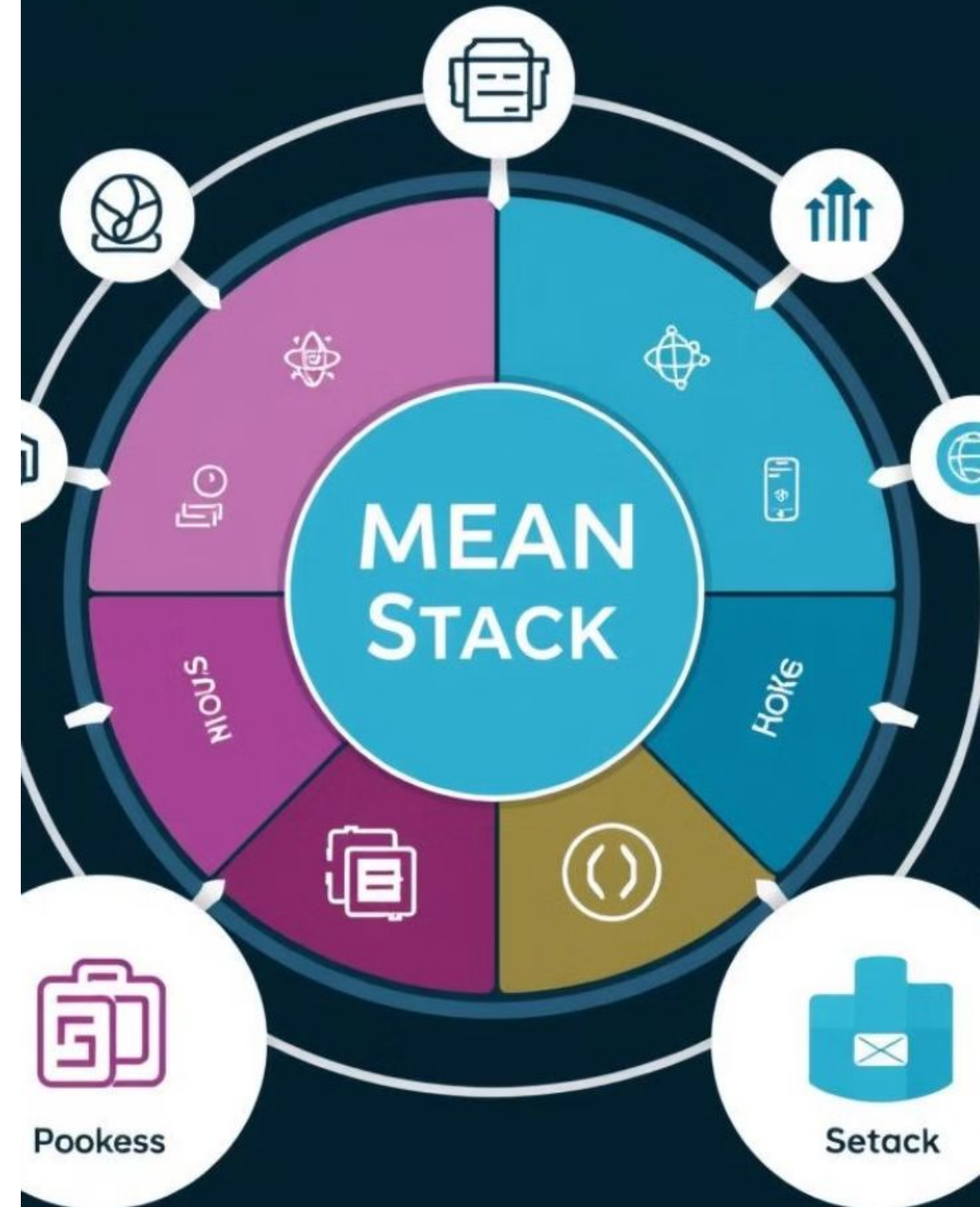
AngularJS (A)

- A powerful client-side JavaScript framework for building dynamic, single-page web applications.
- Two-way data binding and component-based architecture



Node.js (N)

A JavaScript runtime environment that allows developers to run JavaScript on the server-side.



MongoDB and Mongoose

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents, making it a great fit for the MEAN stack.

Mongoose is an Object Data Modeling (ODM) library that provides a higher-level abstraction on top of MongoDB, making it easier to interact with the database.

The combination of MongoDB and Mongoose allows MEAN stack developers to seamlessly integrate the database with their server-side and client-side code.



json

```
{  
  "_id": "1",  
  "name": "John",  
  "age": 30  
}
```

Express.js in MEAN

- **Minimal Web Framework:**

- Manages routes, middleware, and HTTP requests.
- Supports RESTful API creation.

- **Core Features:**

- **Routing:** Define URL paths for different parts of the app.
- **Middleware:** Functions to handle requests before they reach routes.

Example:

```
js

app.get('/api/users', (req, res) => {
  res.send('User data');
});
```

Node.js in MEAN

- **JavaScript Runtime for Server-Side:**
 - Built on Chrome's V8 engine.
 - Handles asynchronous, event-driven operations.
- **Why Node.js?**
 - Non-blocking I/O for high scalability.
 - Single language (JavaScript) across front and back end.

Example: Create a simple server

```
const http = require('http');
const server = http.createServer((req, res) => {
  res.write('Hello, World!');
  res.end();
});
server.listen(3000);
```

AngularJS



Client-side Framework

AngularJS is a powerful JavaScript framework that runs on the client-side, providing a robust set of tools for building dynamic, single-page web applications.

Modular Design

AngularJS encourages a modular design through **components**, making it easier to build and maintain complex web applications.

Data Binding

AngularJS's **two-way data binding** feature automatically synchronizes data between the model and the view, simplifying development.

Testability

AngularJS includes built-in support for testing, which helps ensure the reliability and stability of MEAN stack applications.

Summary

- **MongoDB:**
 - NoSQL database that stores data in flexible, JSON-like documents.
 - Ideal for handling large amounts of unstructured data and easily scalable.
- **Express.js:**
 - Lightweight web application framework for building APIs.
 - Simplifies handling HTTP requests, routes, and middleware for web applications.
- **Angular:**
 - Client-side JavaScript framework for building dynamic user interfaces.
 - Provides tools for two-way data binding, dependency injection, and modular development.
- **Node.js:**
 - JavaScript runtime environment for server-side execution.
 - Built on non-blocking, event-driven architecture, ideal for handling many simultaneous connections.

References

- [https://en.wikipedia.org/wiki/MEAN_\(solution_stack\)](https://en.wikipedia.org/wiki/MEAN_(solution_stack))
- <https://angular.dev/>
- <https://nodejs.org/en>
- <https://expressjs.com/>
- <https://www.mongodb.com/>