MEAN Stack - An Overview

• • •

Naveen Pete
JS Meetup #6 (Saturday, Sep 24, 2016)

Agenda

- History of Web Development
- Traditional Web Development
- Full-Stack Development
- Introducing MEAN Stack
- Node
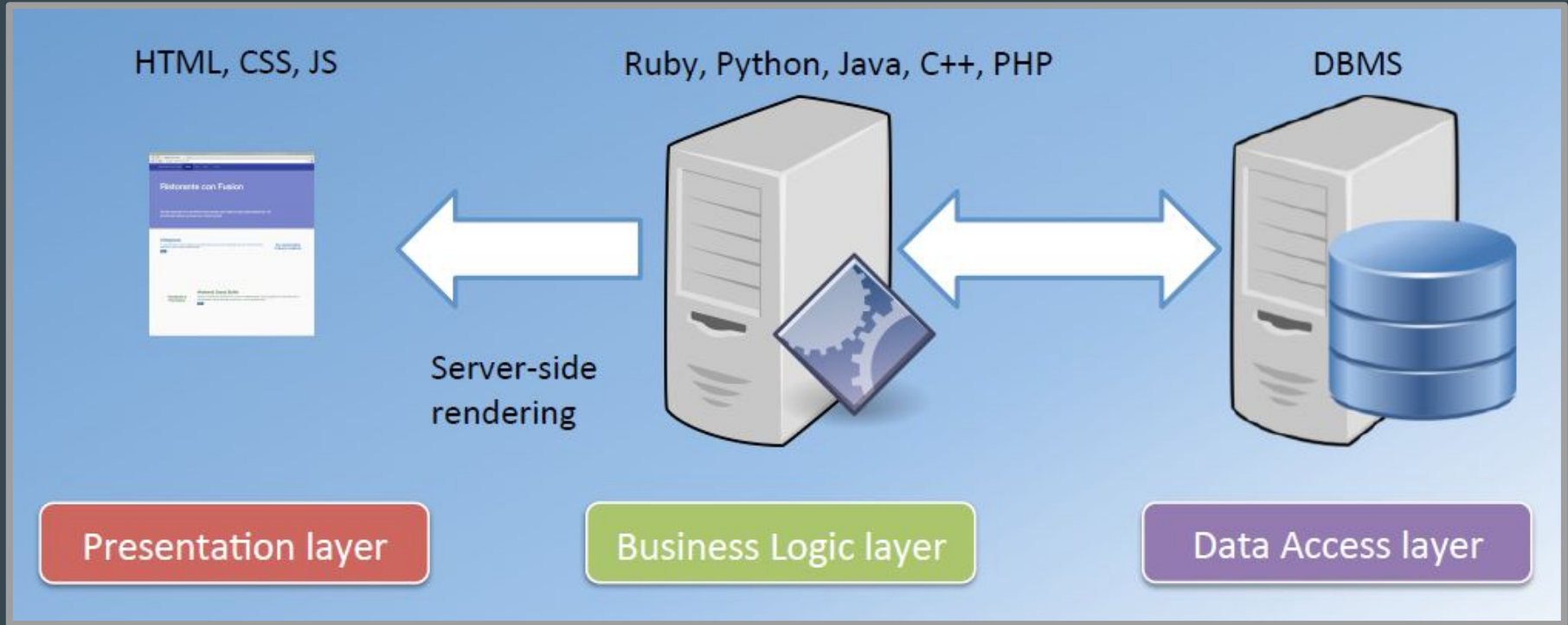- Express
- Angular
- MongoDB
- Q & A

History of Web Development

- Low expectations of web sites
- Perl and HTML - "Web Developer"
- Use of Internet increased with time
- Online presence increased
- Browsers became more powerful
- Front-end development - focus on building a good user experience
- Back-end development - focus on mechanics behind the scenes
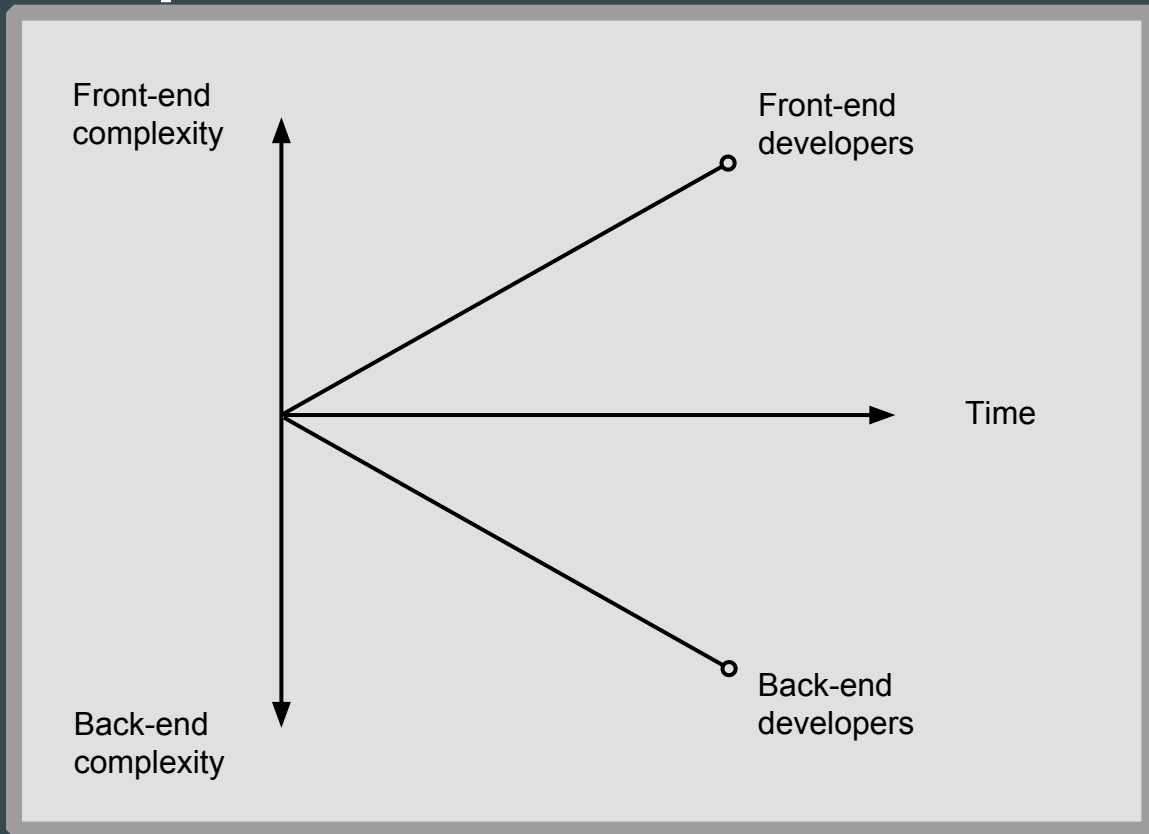
Traditional Web Development

- Followed Three-Tier architecture
  - Presentation Layer - UI design and development
  - Business Logic Layer - Data validation, data processing, application logic
  - Data Access Layer - Data persistence, data access through API
- Traditional web application technologies
  - Client-side - HTML, CSS and JavaScript
  - Server-side - C#, Java, PHP, ASP.NET, Ruby, Python
  - Database - SQL Server, MySQL, Oracle
  - Web Server - IIS, Apache, Tomcat

Traditional Web Development

HTML, CSS, JS

Ruby, Python, Java, C++, PHP

DBMS

Server-side rendering

Presentation layer

Business Logic layer

Data Access layer

Traditional Web Development

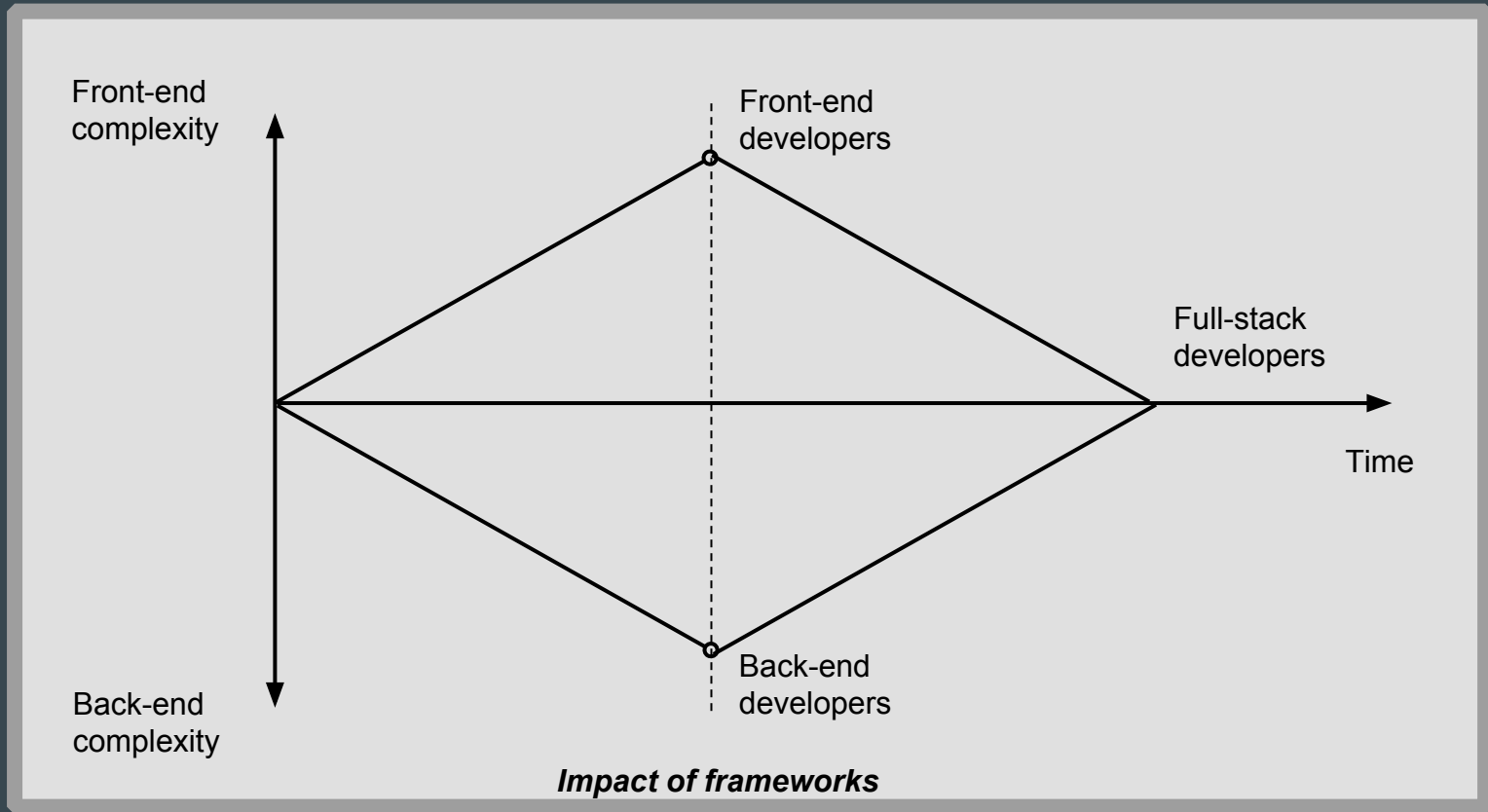*Divergence of front-end and back-end developers over time*

Full-Stack Development

- During the 2000s, libraries and frameworks started to become popular and prevalent
- A Library or Framework
  - Abstracts complexities of development
  - Increases developer productivity
  - Requires less in-depth expertise
- Resurgence of full-stack developers
- Moving the application code forward in the stack
  - Reduces server load, thus reducing cost
  - Crowd-sourcing of computational power

Full-Stack Development - Benefits

- Provide greater value to your clients
  - More skills, services and capabilities
- Build applications end-to-end, no dependency
- Have a better view of the bigger picture of an application
- Understand how different components fit together
- Better overall control
- Team members can move around more freely (Agile teams)
- Learning new technologies
- Rewarding and satisfying experience

Full-Stack Development

Front-end complexity

Front-end developers

Full-stack developers

Time

Back-end developers

Back-end complexity

Impact of frameworks

Introducing MEAN Stack

MongoDB - the database
- *Has been around since 2007*
- *By MongoDB Inc. (10gen)*
- *NoSQL, Document database*
- *JSON like documents, dynamic schemas*

Express - the web framework
- *First released in 2009*
- *Web application framework*
- *Minimal, flexible, extensible*
- *Build web applications and API*

Angular - the front-end framework
- *Has been around since 2010*
- *Maintained by Google*
- *Front-end, SPA framework*
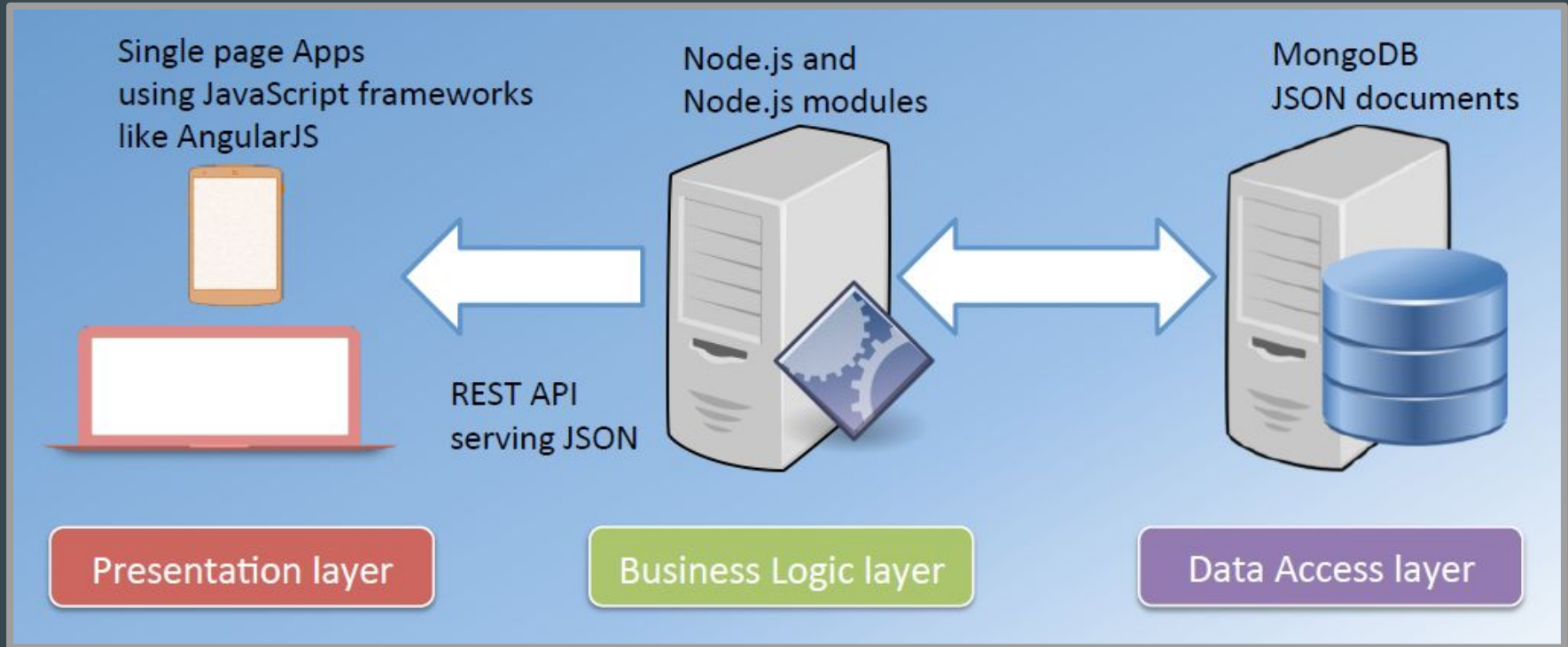- *Build highly interactive, responsive and dynamic web applications*

Node - the platform
- *Created in 2009 by Ryan Dahl*
- *Sponsored by Joyent*
- *It is a platform, neither a language nor a web server*
- *Uses Google's Chrome V8 JavaScript engine at its core*

Introducing MEAN Stack

- "Best-of-Breed" modern web technologies
- Very powerful and flexible stack
- Uses JavaScript throughout
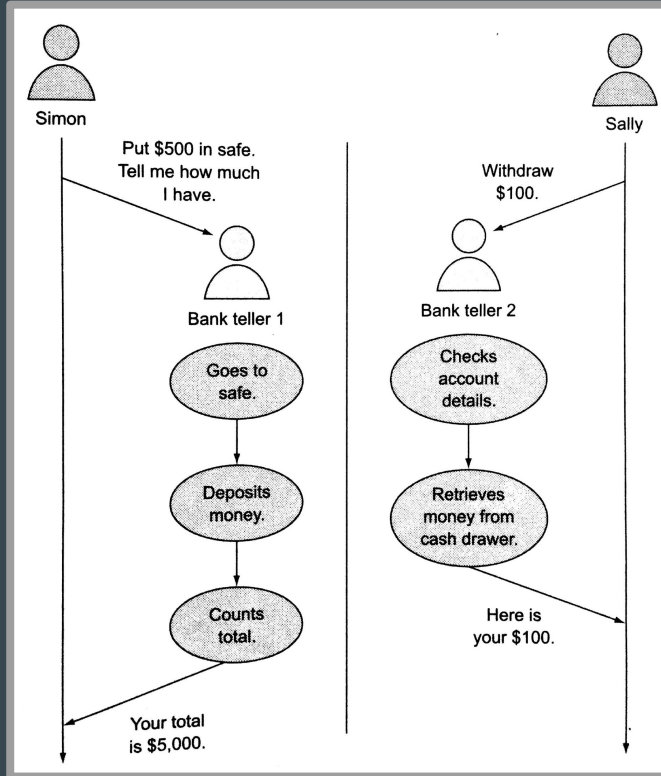
Introducing MEAN Stack

Single page Apps using JavaScript frameworks like AngularJS

Node.js and Node.js modules

MongoDB JSON documents

REST API serving JSON

Presentation layer

Business Logic layer

Data Access layer

Node

- Foundation of MEAN stack
- A software platform
  - Create your own web server
  - Build web applications on top of it
- Built on Google Chrome's V8 JavaScript engine
- Single-threaded
- Event driven architecture
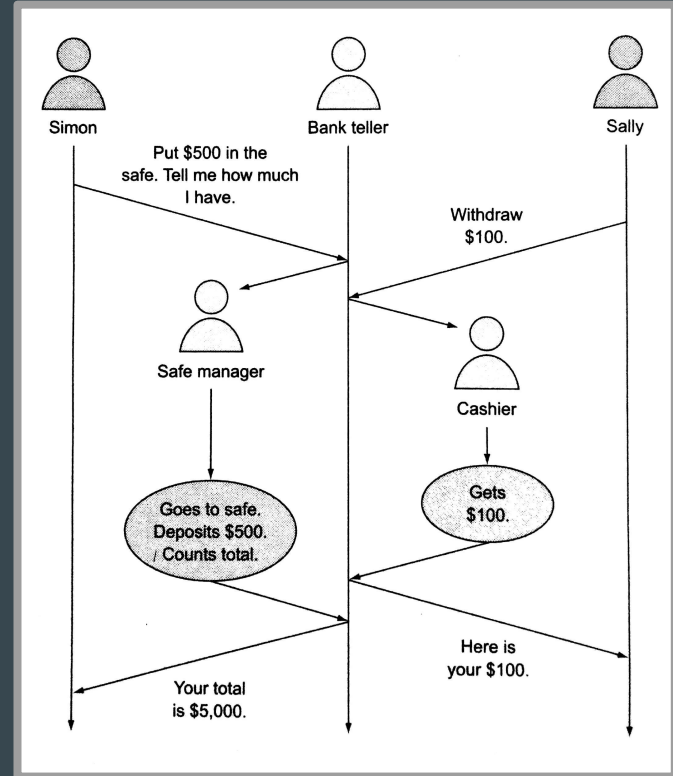- Non-blocking I/O model
- Asynchronous execution

Benefits
- Code it in JavaScript
- Extremely fast and lightweight
- Very efficient use of system resources
- No need to run a separate web server
- Gives you greater control over your app logic and environment
- Serve more users on fewer server resources

Node - Threading

Multithreaded Approach

Single-threaded Approach

Node - npm

- Package manager for JavaScript
- Gets installed with Node.js
- Allows users to consume and distribute JavaScript modules (packages)
- Packages extend the functionality of your application
- Promotes reusability

Examples
- Web App Framework - Express
- ODM - Mongoose
- Helper library - Underscore
- Testing Framework - Mocha, Jasmine

Express

- Node.js web application framework
- Minimal
- Flexible
- Fast
- Extensible
- Provides robust set of features:
    - *Middleware*
    - *Serve static files*
    - *Routing*
    - *Template Engines*
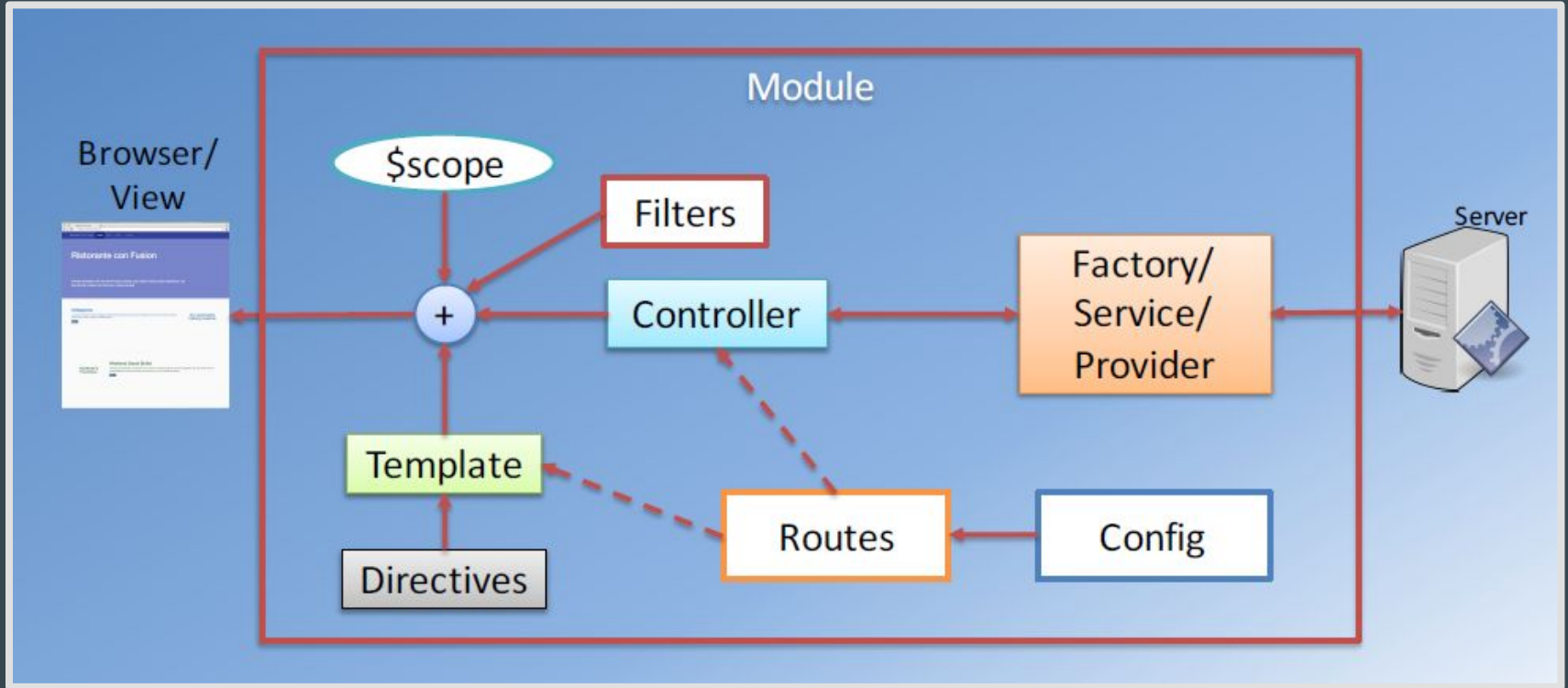   - *Express Generator*

Angular

- Developed in 2009 by Misko Hevery
- JavaScript framework for creating dynamic web applications
- Front-end SPA, RIA framework
- Data-driven applications
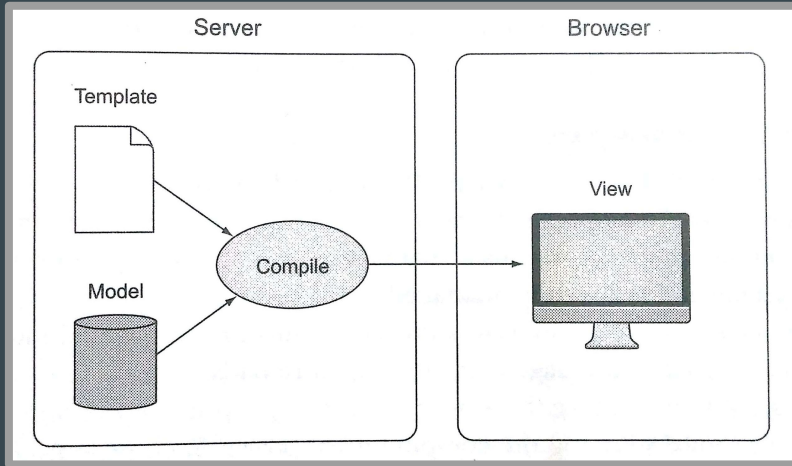- Adds dynamic features to HTML
- Declarative approach

Advantages

- Makes it easy to organize your code
  - Promotes maintenance, collaboration, readability, extension
- Testable code gives peace of mind
- Two-way data binding saves hundreds of lines of code
- Uses HTML for templating
- JavaScript data structures (JSON) eases integration
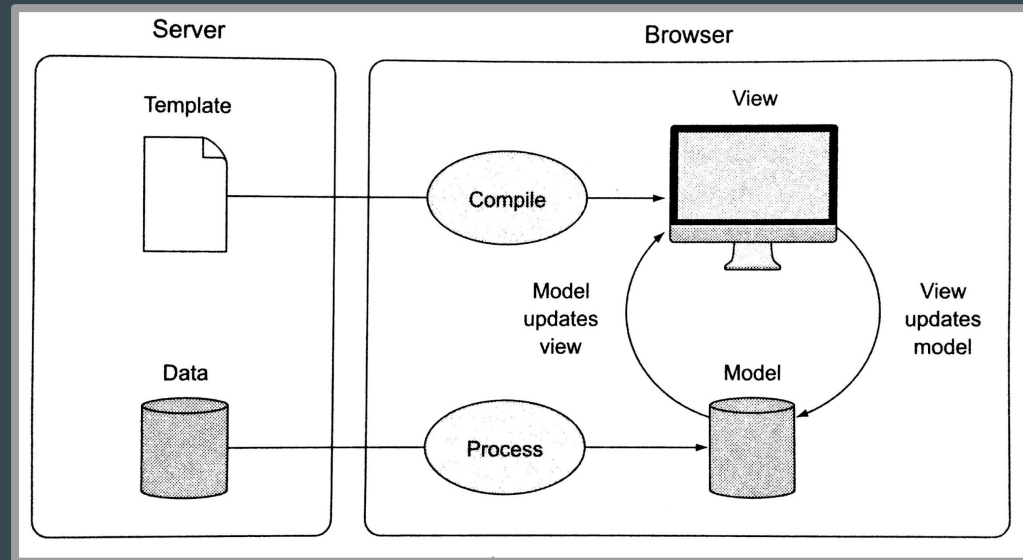
Angular - Building Blocks

Angular - Binding

One-way data binding
The template and model are compiled on the server before being sent to the browser

Two-way data binding
The model and the view are processed in the browser and bound together, each instantly updating the other

MongoDB

Q & A

Thank you!