Full-Stack Java Development with Spring Boot

Code on the Beach 2022

Dan Vega
Spring Developer Advocate @VMware

vmware

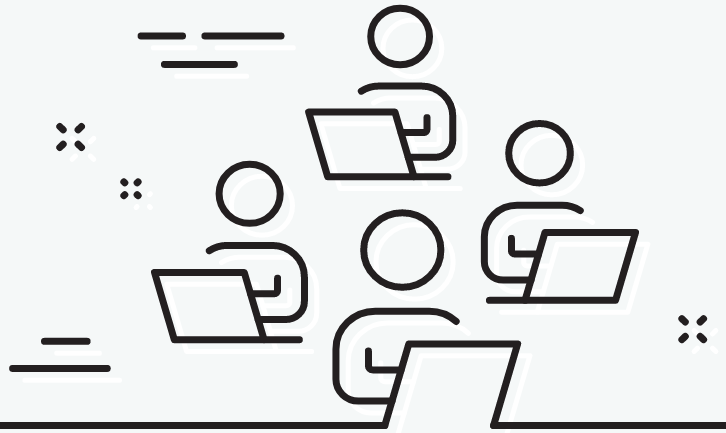
ABOUT ME

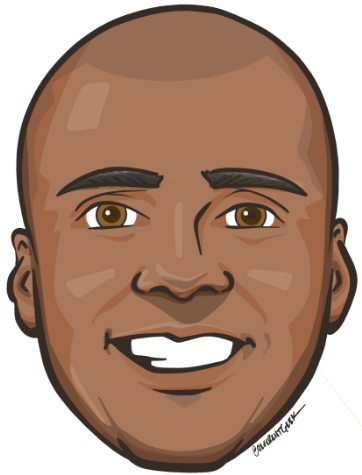
- Husband & Father
- Cleveland, OH
- Software Development 20+ Years
- Spring Developer Advocate
- Content Creator (www.danvega.dev)
  - Blogger
  - YouTuber
  - Course Creator
- @therealdanvega

spring

OFFICE HOURS

https://bit.ly/spring-office-hours





https://tanzu.vmware.com/developer/tv/spring-office-hours/

AGENDA

- What is a Full-Stack Developer
- Back-end Development
- Front-end Development
- Full-Stack Architecture
- Q+A

spring

WHAT IS A FULL-STACK DEVELOPER?


FULL
SNACK

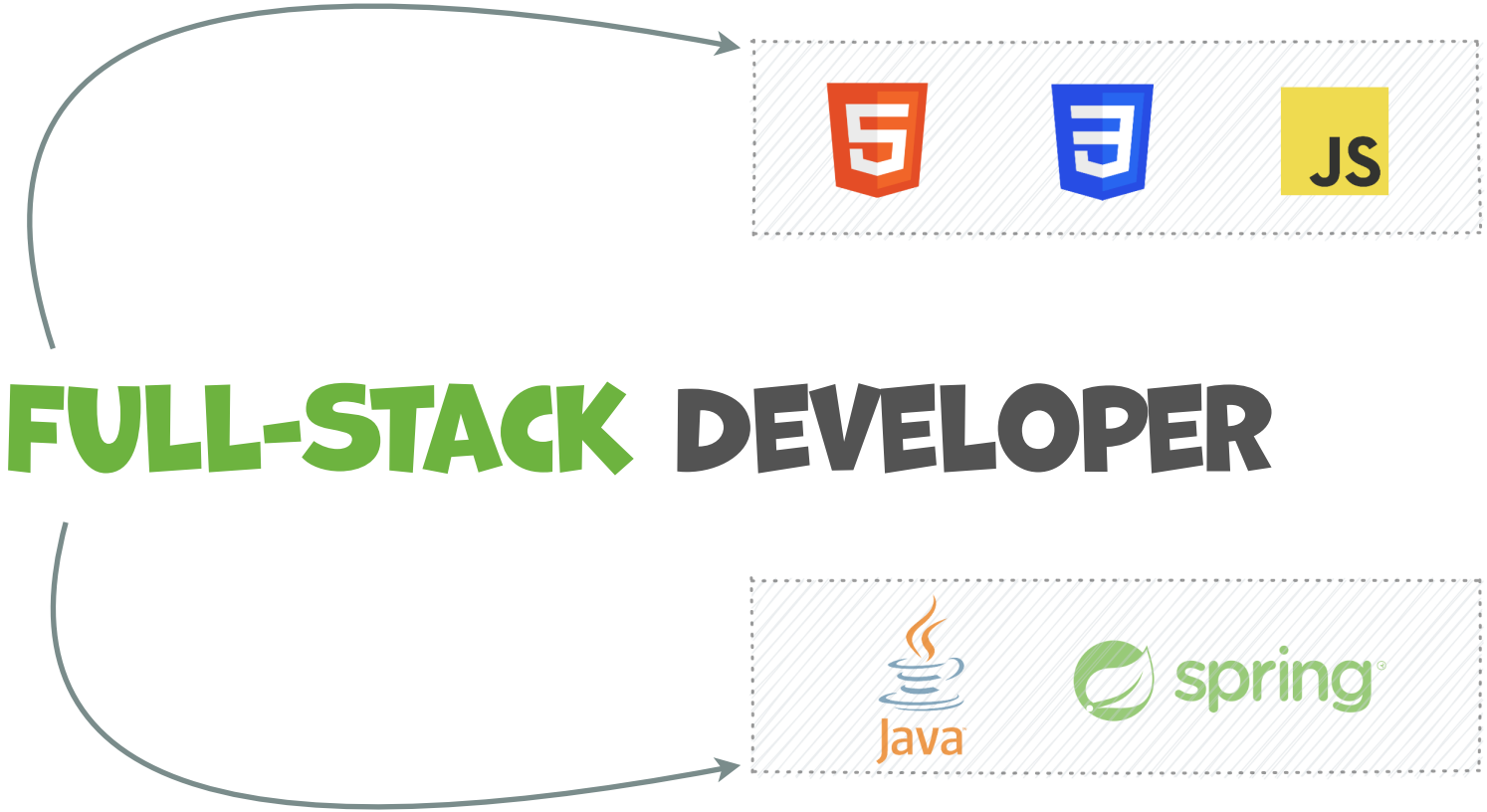
How do you spell Full Stack

Fullstack 🚫

Full Stack ✅ Noun

Full-Stack ✅ Adjective

FULL-STACK DEVELOPER


FULL-STACK DEVELOPER
JS
Java
spring

FRONT-END

- HTML
  - DOM
  - Tags
  - Attributes
  - Forms
  - Semantic HTML
  - Accessibility (A11y)
- CSS
  - Box Model
  - Selectors & Rules
  - Specificity
  - Grid & Flexbox
  - Responsive Design
  - Frameworks
- JavaScript
  - Modern JavaScript (ES6)
  - Build Tools
  - Frameworks

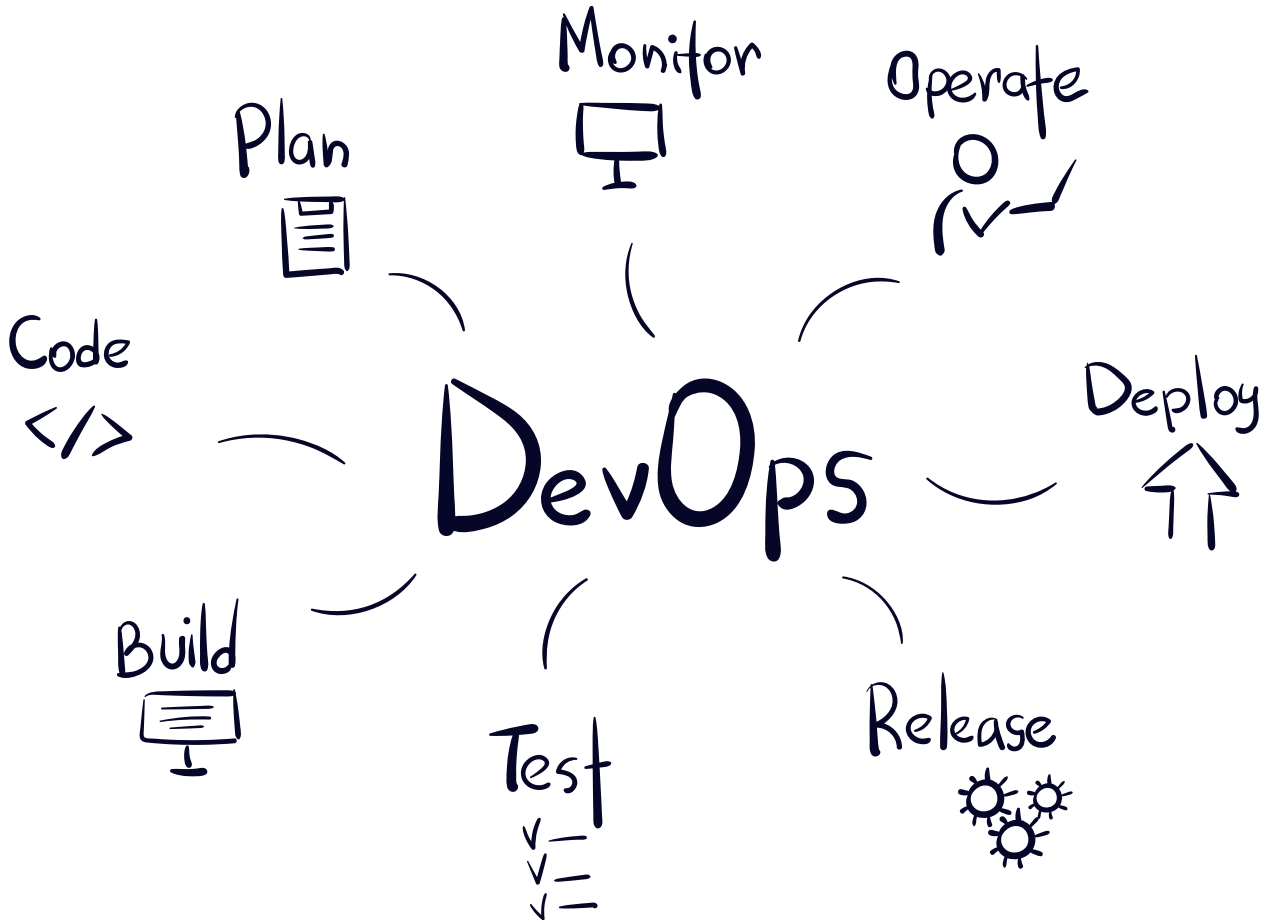
EVERYTHING ELSE

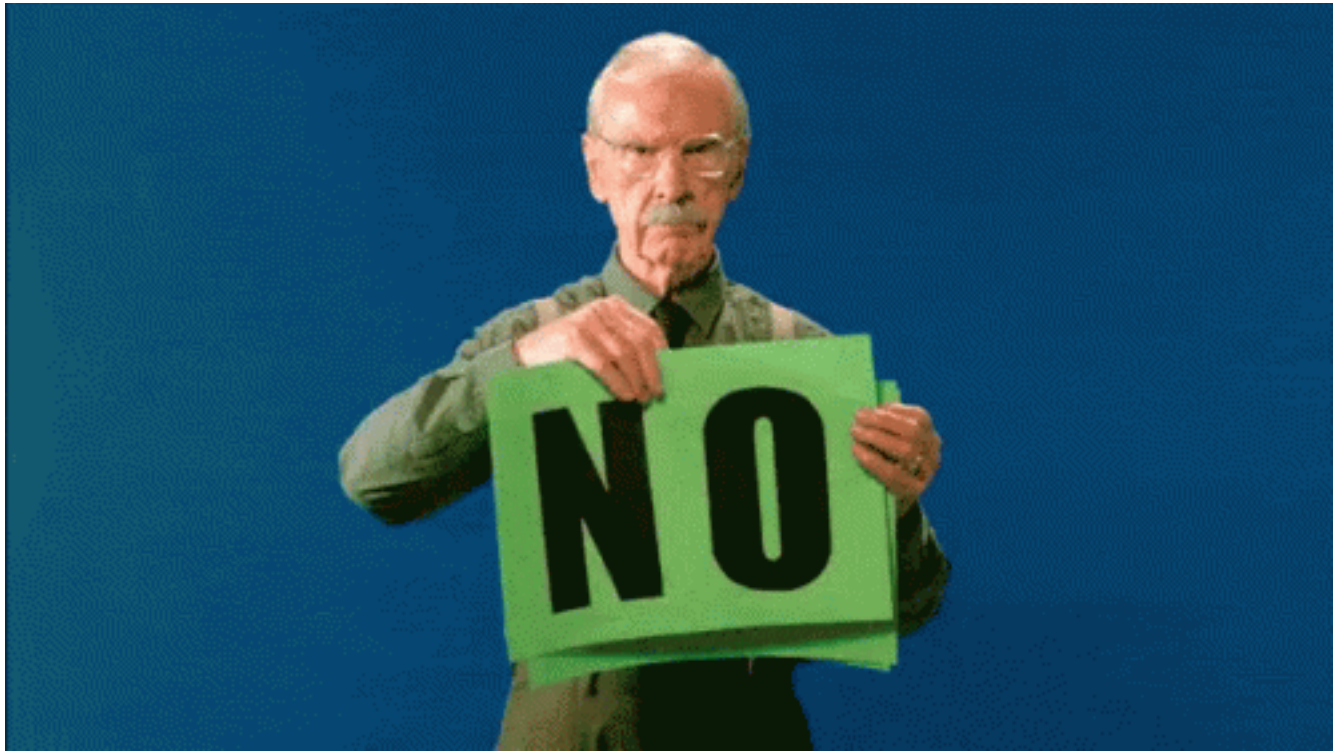
- How the Internet works
  - Browsers
  - IP address
  - Domain Names
  - DNS
  - HTTP/HTTPS Protocol
  - Hosting Providers
  - Cloud Services
- Version Control
  - Git
  - Github
- IDE
  - Visual Studio Code
  - IntelliJ
- Tools
  - Command Line
  - Chrome DevTools

BACK-END

- Programming Language
  - Java
  - Groovy
  - Kotlin
- Build Tools
  - Maven
  - Gradle
- Frameworks
  - Spring Framework
  - Micronaut
  - Quarkus
- Testing
  - JUnit 5
  - Mockito
- Databases
  - Relational (Postgres, MySQL)
  - NoSQL (Redis, MongoDB)

WHAT ABOUT...


Monitor
Operate
Plan
Code
DevOps
Deploy
Build
Test
Release


NO

BACK-END DEVELOPMENT

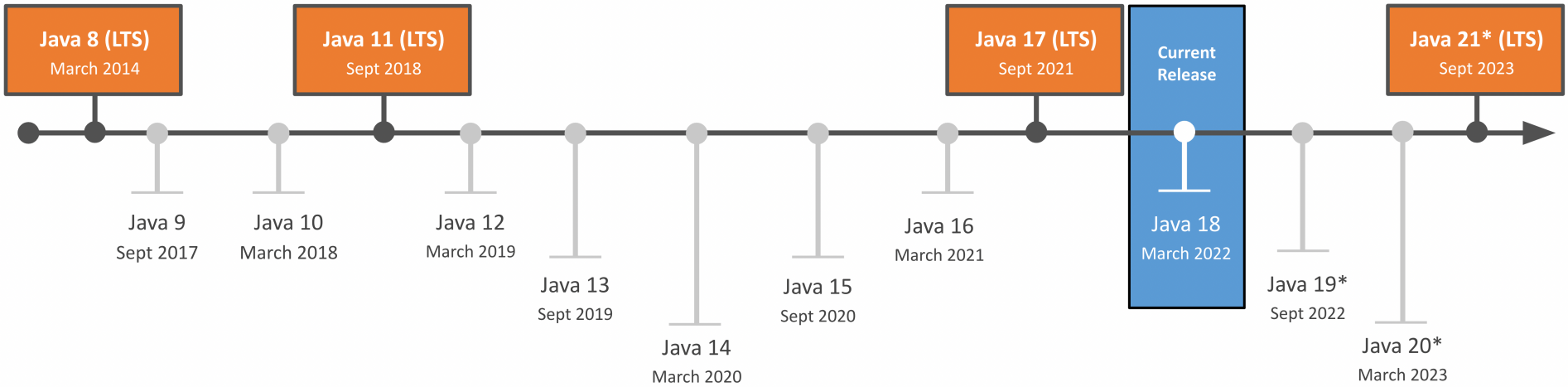

Java™


spring®

JDK Release Cadence

- Time-bound (6-month cadence) feature releases
- Non-LTS Releases
  - No public patches after next release
  - No overlapping patches
  - Little or no commercial support
  - Not recommended for production workloads
- Long Term Support (LTS) Releases
  - Recommended for production workloads
  - Started with version 11 (Sept. 2018) and approximately every 3 years after*
  - Long-term support available from vendors at additional cost: Oracle, Red Hat, IBM, etc.

JDK Release Timeline



Feature Summary – JDK 11 to JDK 17

From LTS to LTS

Java 12

Shenandoah: A Low-Pause-Time Garbage Collector
Microbenchmark Suite
Switch Expressions (Preview)
JVM Constants API
One AArch64 Port, Not Two
Default CDS Archives
Abortable Mixed Collections for G1
Promptly Return Unused Committed Memory from G1

Java 13

Dynamic CDS Archives
ZGC: Uncommit Unused Memory
Reimplement the Legacy Socket API
Switch Expressions (Preview)
Text Blocks (Preview)

Java 14

Pattern Matching for instanceof (Preview)
Packaging Tool (Incubator)
NUMA-Aware Memory Allocation for G1
JFR Event Streaming
Non-Volatile Mapped Byte Buffers
Helpful NullPointerExceptions
Records (Preview)
Switch Expressions
Deprecate the Solaris and SPARC Ports
Remove the Concurrent Mark Sweep (CMS) Garbage Collector
ZGC on macOS
ZGC on Windows
Deprecate the ParallelScavenge + SerialOld GC Combination
Remove the Pack200 Tools and API
Text Blocks (Preview)
Foreign-Memory Access API (Incubator)

Java 15

Edwards-Curve Digital Signature Algorithm (EdDSA)
Sealed Classes (Preview)
Hidden Classes
Remove the Nashorn JavaScript Engine
Reimplement the Legacy DatagramSocket API
Disable and Deprecate Biased Locking
Pattern Matching for instanceof (Preview)
ZGC: A Scalable Low-Latency Garbage Collector
Text Blocks
Shenandoah: A Low-Pause-Time Garbage Collector
Remove the Solaris and SPARC Ports
Foreign-Memory Access API (Incubator)
Records (Preview)
Deprecate RMI Activation for Removal

Java 16

Vector API (Incubator)
Enable C++14 Language Features
Migrate from Mercurial to Git
Migrate to GitHub
ZGC: Concurrent Thread-Stack Processing
Unix-Domain Socket Channels
Alpine Linux Port
Elastic Metaspace
Windows/AArch64 Port
Foreign Linker API (Incubator)
Warnings for Value-Based Classes
Packaging Tool
Foreign-Memory Access API (Incubator)
Pattern Matching for instanceof
Records
Strongly Encapsulate JDK Internals by Default
Sealed Classes (Preview)

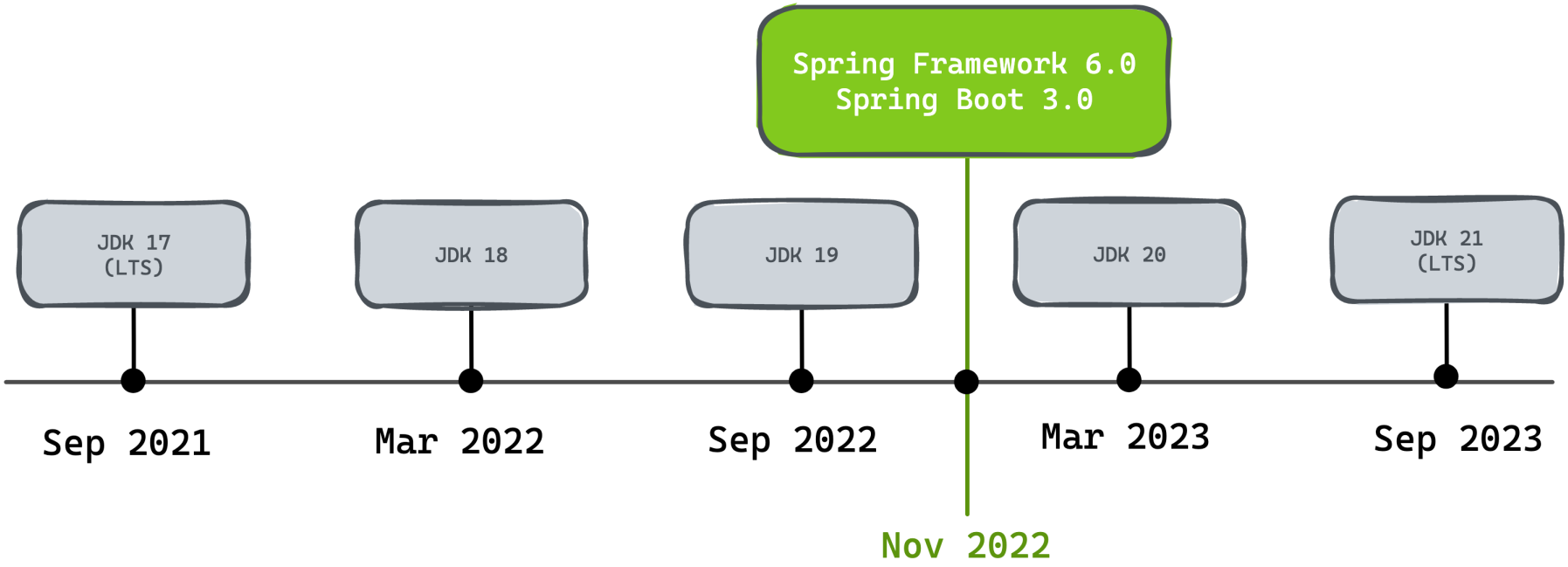
Java 17

Restore Always-Strict Floating-Point Semantics
Enhanced Pseudo-Random Number Generators
New macOS Rendering Pipeline
macOS/AArch64 Port
Deprecate the Applet API for Removal
Strongly Encapsulate JDK Internals
Pattern Matching for switch (Preview)
Remove RMI Activation
Sealed Classes
Remove the Experimental AOT and JIT Compiler
Deprecate the Security Manager for Removal
Foreign Function & Memory API (Incubator)
Vector API (Incubator)
Context-Specific Deserialization Filters

Spring Framework 6 & Spring Boot 3

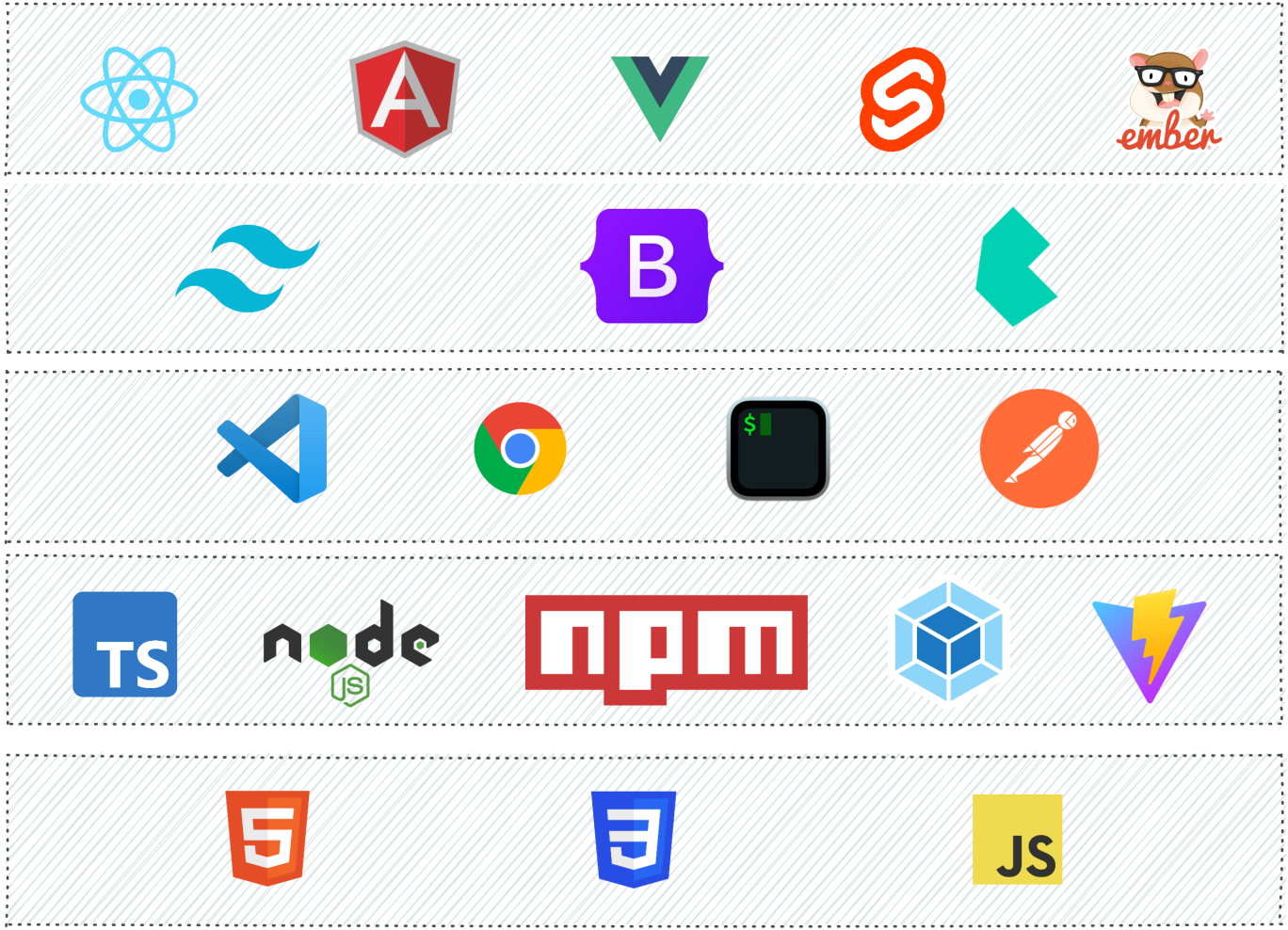
A new generation of Spring for 2023 and beyond

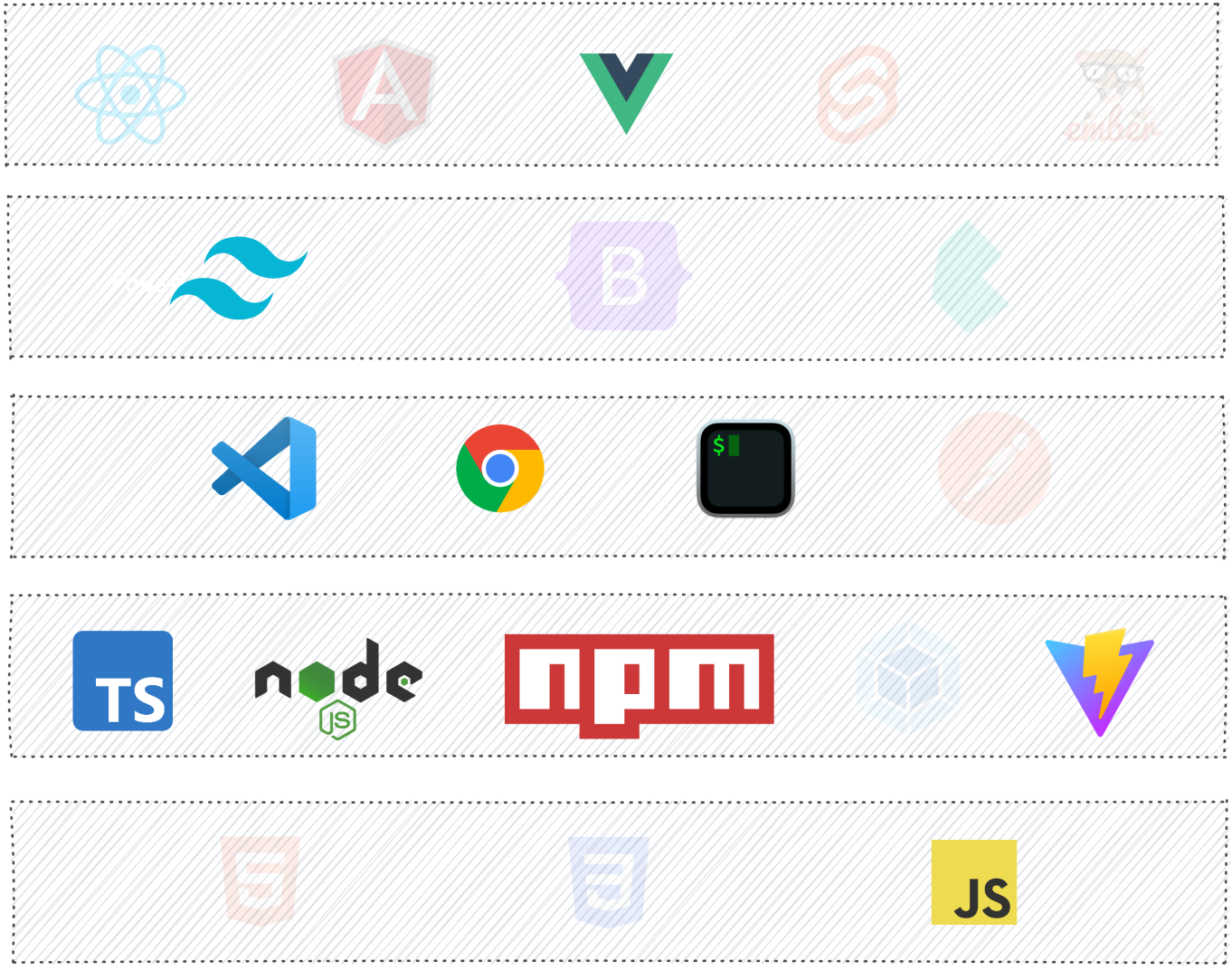
- JDK 17+
- Jakarta EE 9+ (Jakarta Namespace)
- Observability initiative (Micrometer)
- Spring Native (Ahead-of-Time)
- Declarative HTTP Clients


Spring Framework 6.0
Spring Boot 3.0
JDK 17
(LTS)
JDK 18
JDK 19
JDK 20
JDK 21
(LTS)
Sep 2021
Mar 2022
Sep 2022
Nov 2022
Mar 2023
Sep 2023

SPRING BOOT DEMO

FRONT-END DEVELOPMENT


ember
B
TS
node
JS
npm
JS


TS
node
JS
npm
JS

I COME FROM AN ERA OF WEB DEVELOPMENT WHERE WRITING JAVASCRIPT WASN'T FUN

ECMASCRIPT 2015 (ES6)

- Variable Scoping
- Arrow Functions
- Classes
- Destructuring
- Default Paramaters
- Rest parameter
- Spread Operator
- Generators
- Iterators & For..of
- Promises

ECMASCRIPT 2017 (ES8)

- Async / Await
- Object values
- Object Entries
- Property Descriptors

ES9+

- Promise finally
- Object fromEntries
- JSON Improvements
- Private Class Variables
- BigInt
- Optional Chaining
- Nullish Coalescing ??
- replaceAll
- promise.any

TypeScript

- Open Source (Current Version 4.7.4)
- JavaScript is a dynamic language
- TypeScript is a superset of JavaScript
- Static Typing
- IDE Support
  - Code Completion
  - Refactoring
  - Compilation Errors

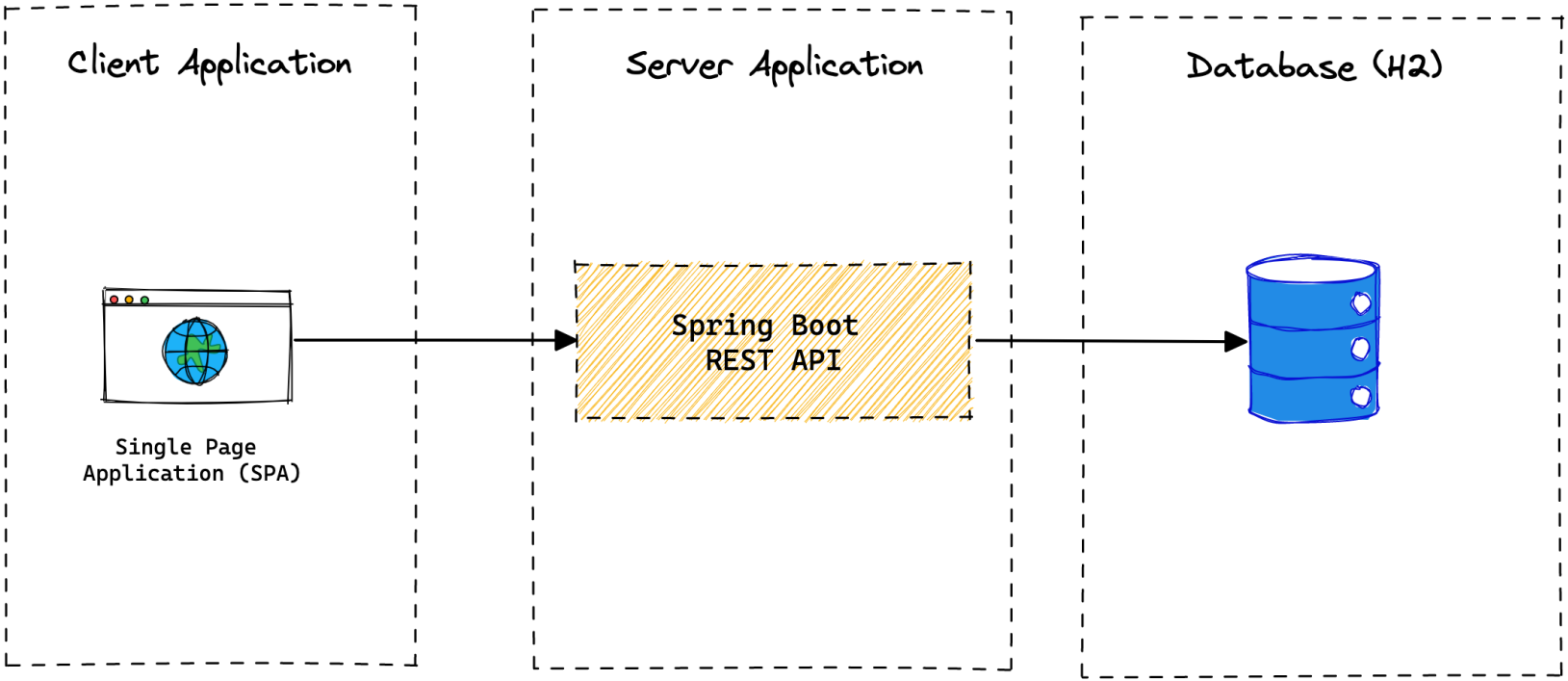
Next Generation Frontend Tooling

- Vite
- Framework Agnostic
  - Vanilla JS
  - Vue / React / Preact / Lit / Svelte
- Native ES Modules
- esbuild - An extremely fast JS bundler
- Instant Server Start
- Lightning Fast HMR
- Optimized Build (Rollup for prod)
- TypeScript, JSX, Vue,
- CSS Imports, PostCSS, CSS Modules

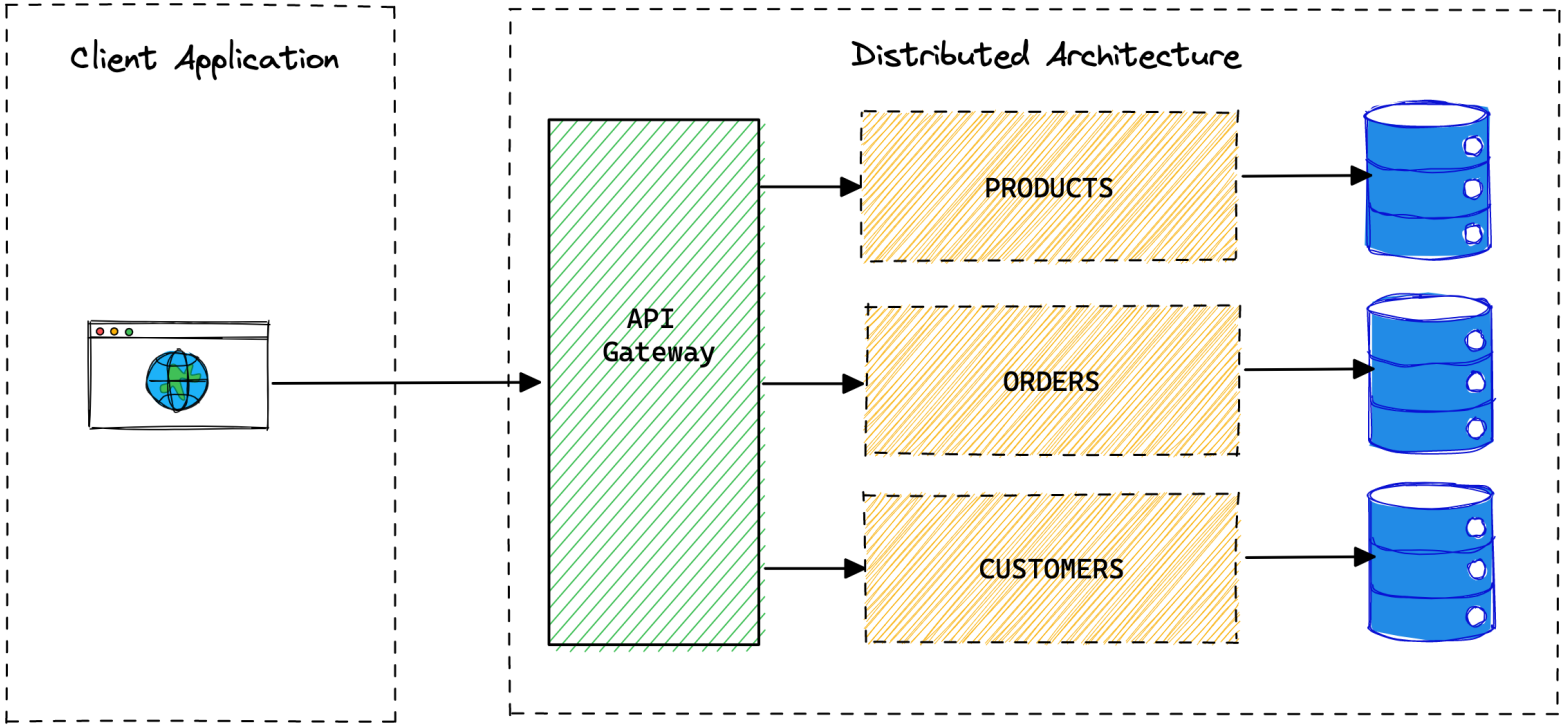
FRONTEND DEMO

FULL-STACK ARCHITECTURE

CLIENT / SERVER ARCHITECTURE



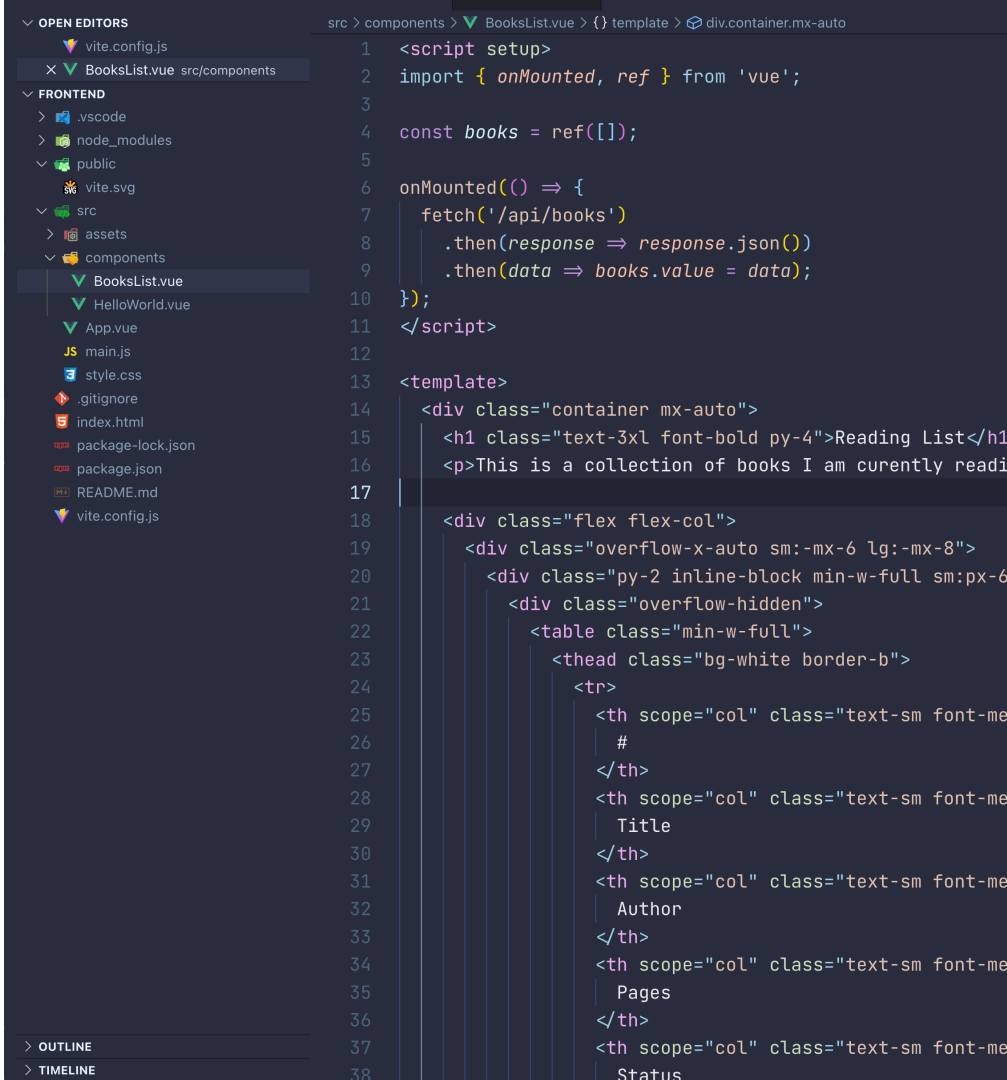
CLIENT / SERVER ARCHITECTURE



SINGLE PAGE APPLICATION (SPA) ARCHITECTURE

- Pros
  - Independently deployable apps
  - Low Coupling
  - Scalable
- Cons
  - Independently deployable apps
  - Local Development Setup
  - Performance
  - SEO

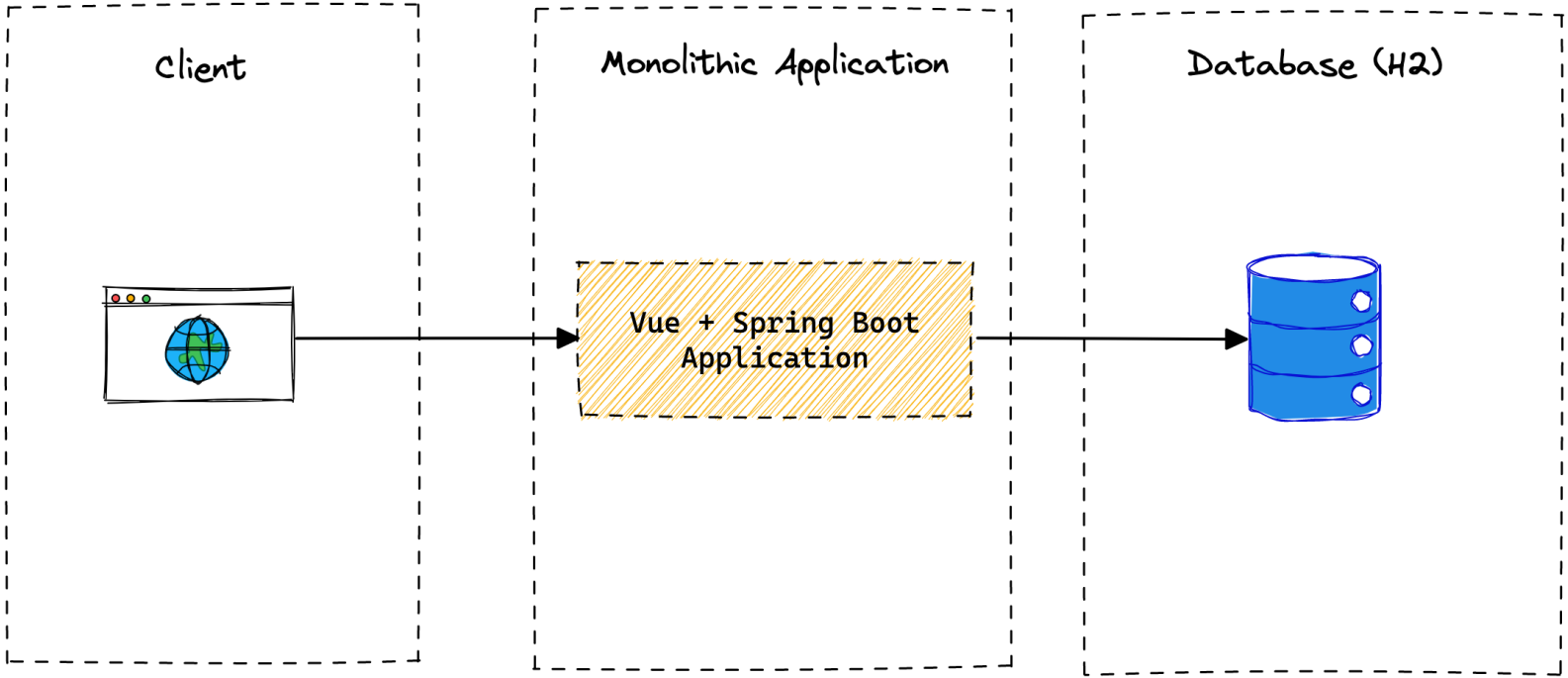
spring



“A software system is called “**monolithic**” if it has monolithic architecture, in which functionally distinguishable aspects are all interwoven, rather than containing architecturally separate components”

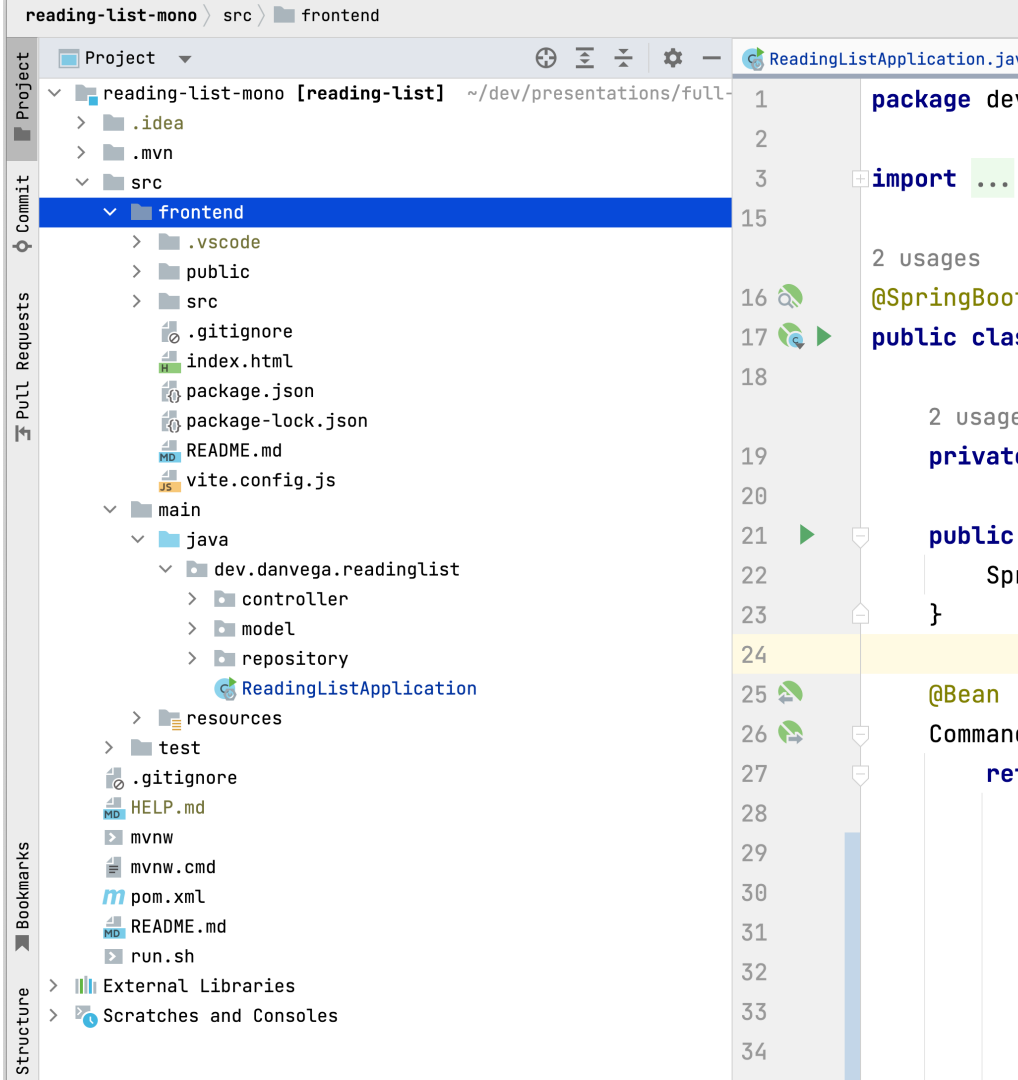
Wikipedia

MONOLITH APPLICATION



MONOLITHIC ARCHITECTURE

- Pros
  - Single Deployable Asset (JAR)
  - Simplicity
  - Performance
  - SEO
- Cons
  - Single Deployable Asset (JAR)



MONOLITH ARCHITECTURE DEMO

Thank you

Contact me at dvega@vmware.com

@therealdanvega