



# Full-Stack Java Development with Spring Boot

---

Code on the Beach 2022

Dan Vega  
Spring Developer Advocate @VMware



# ABOUT ME

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





# 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



# 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



# 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

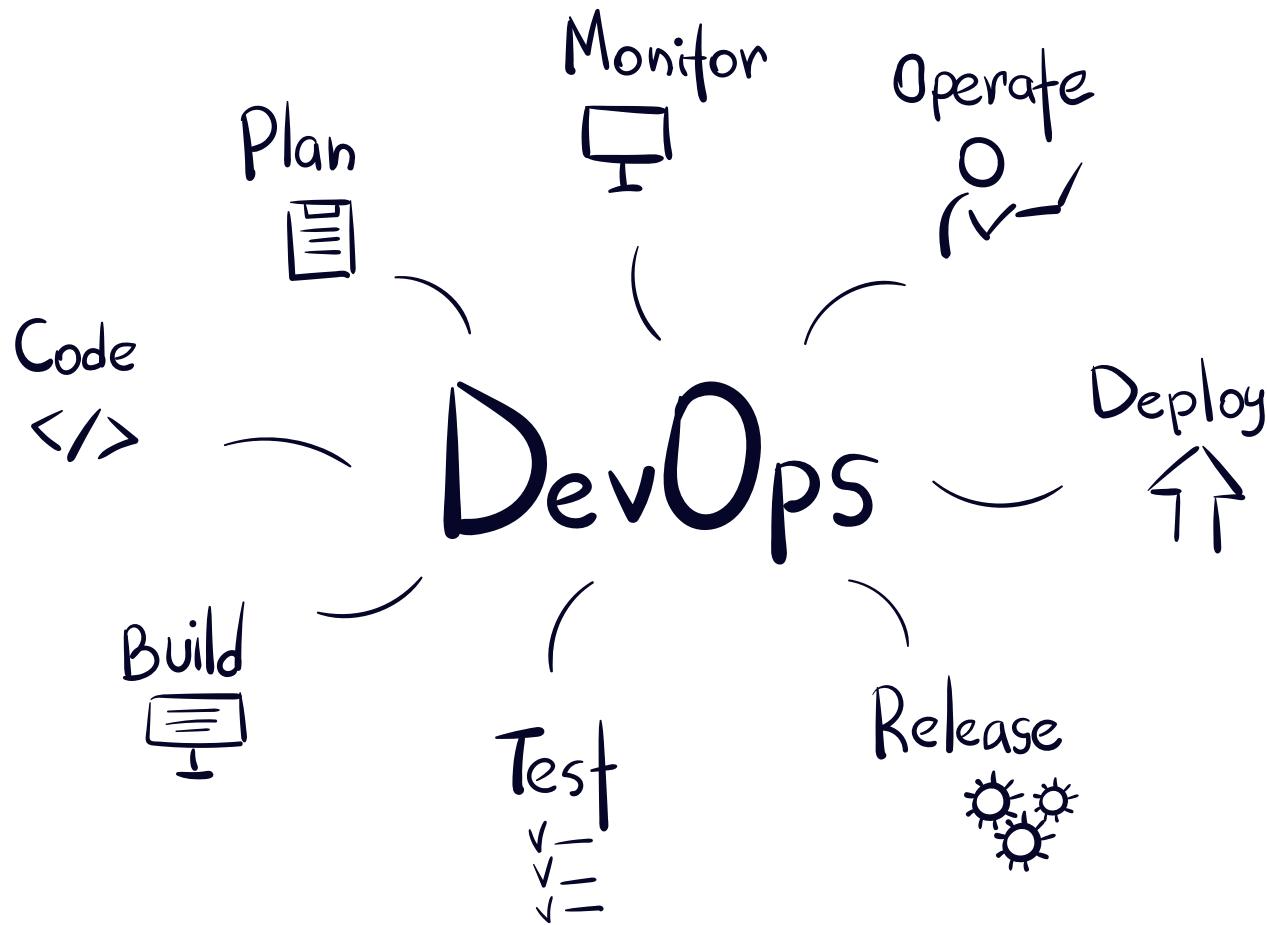
# EVERYTHING ELSE

- 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

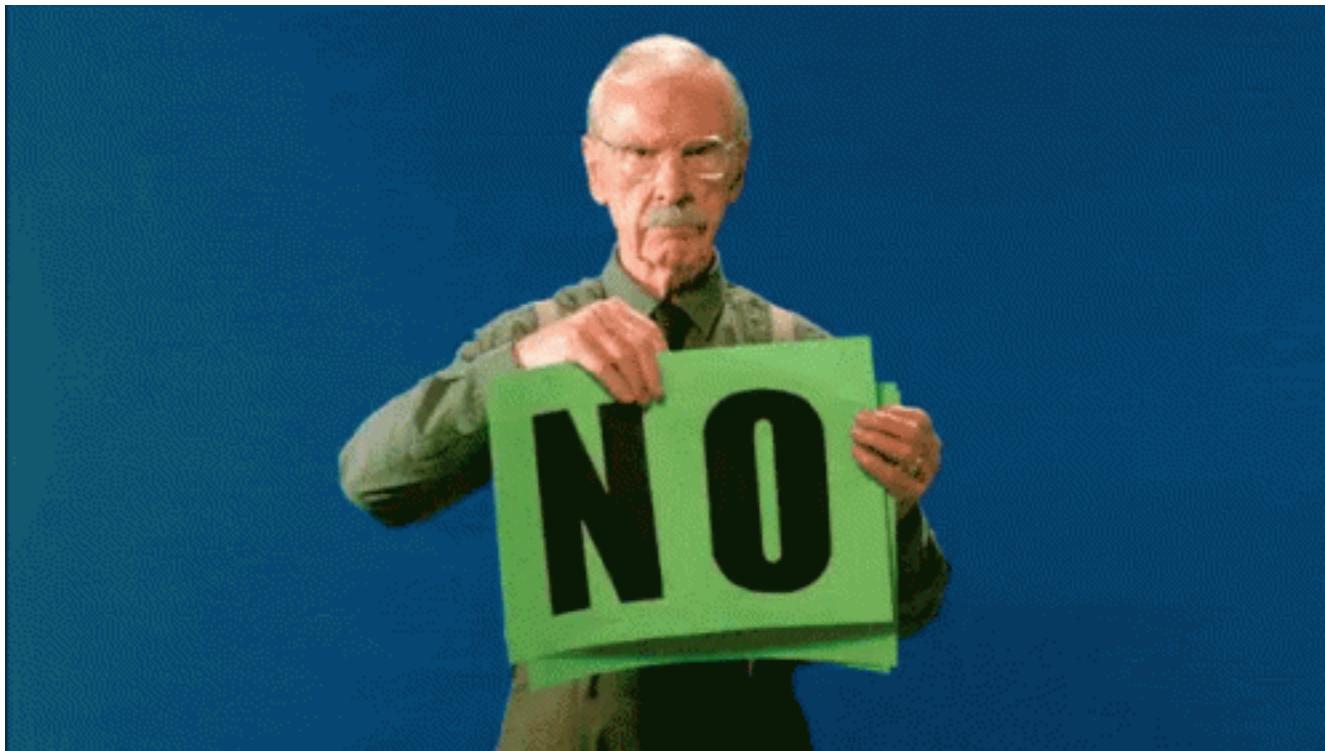
# BACK-END

- 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...**







# **BACK-END DEVELOPMENT**

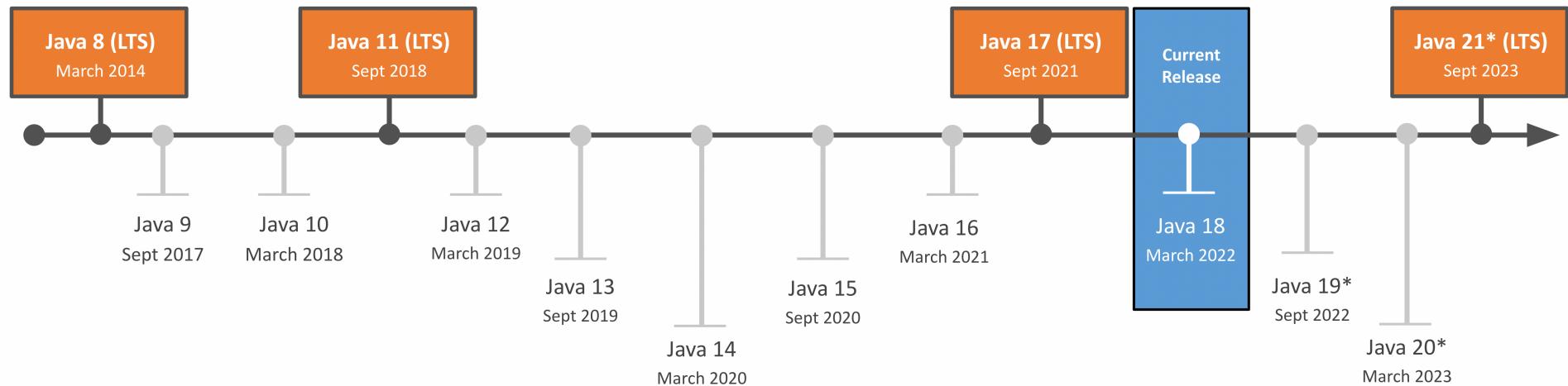


## JDK Release Cadence

- 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

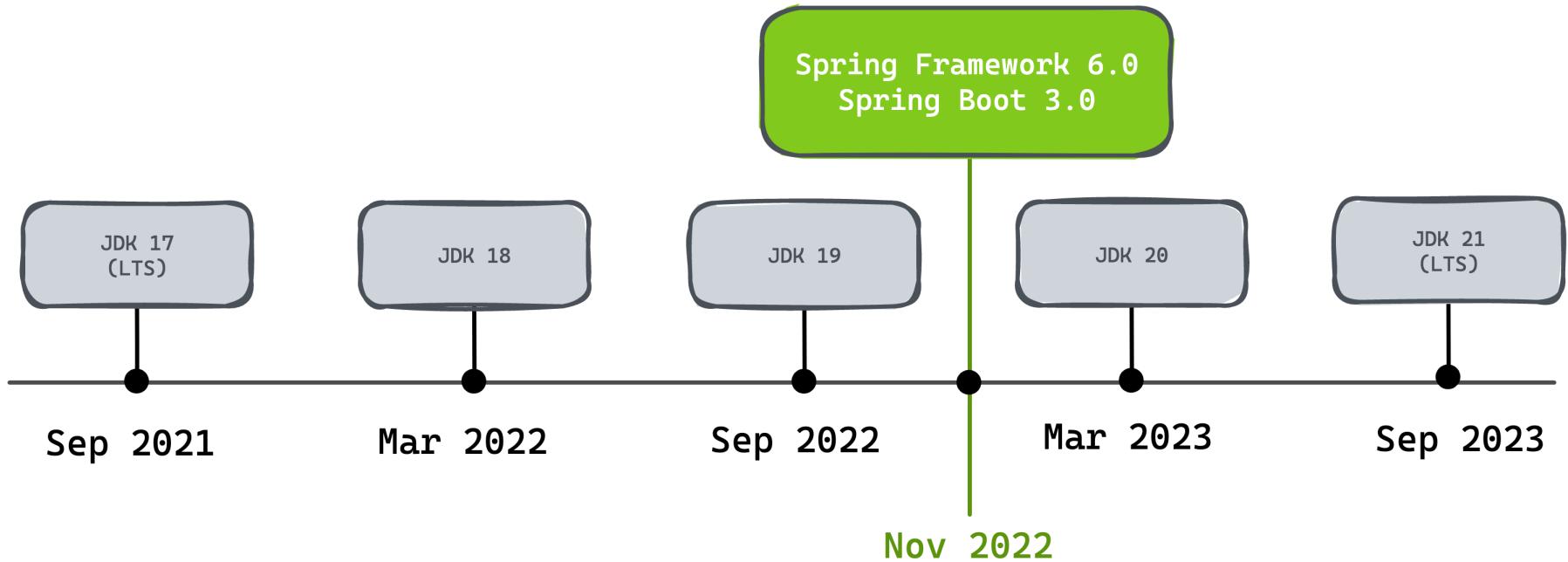
Restore Always-Strict Floating-Point Semantics  
Enhanced Pseudo-Random Number Generators  
New macOS Rendering Pipeline  
macOS/AArch64 Port  
Deprecate the Applet API for Removal  
Strongly Encapsulate JDK Internals  
Pattern Matching for switch (Preview)  
Remove RMI Activation  
Sealed Classes  
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



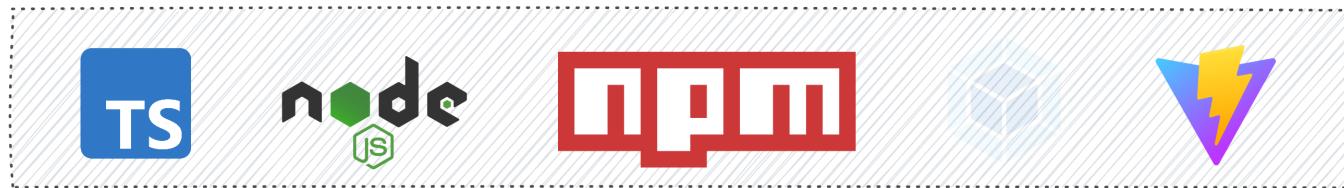
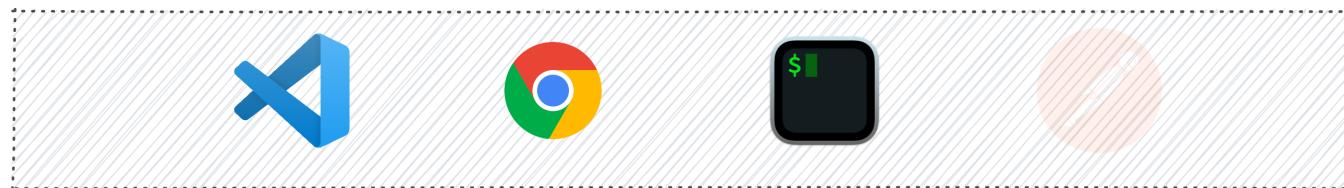
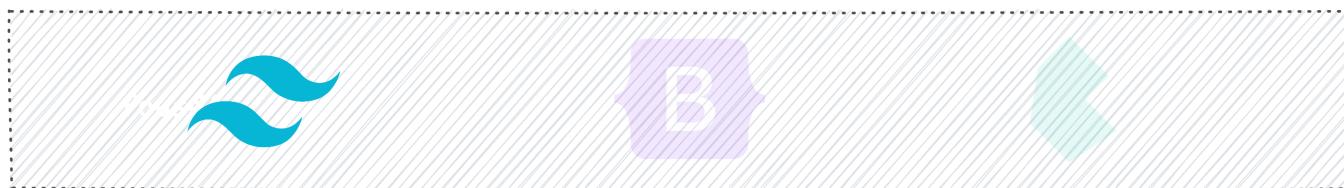


# SPRING BOOT DEMO



# FRONT-END DEVELOPMENT





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





Chrome



Firefox



Safari



Edge

### ECMASCRIPT 2015 (ES6)

- Variable Scoping
- Arrow Functions
- Classes
- Destructuring
- Default Parameters
- 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

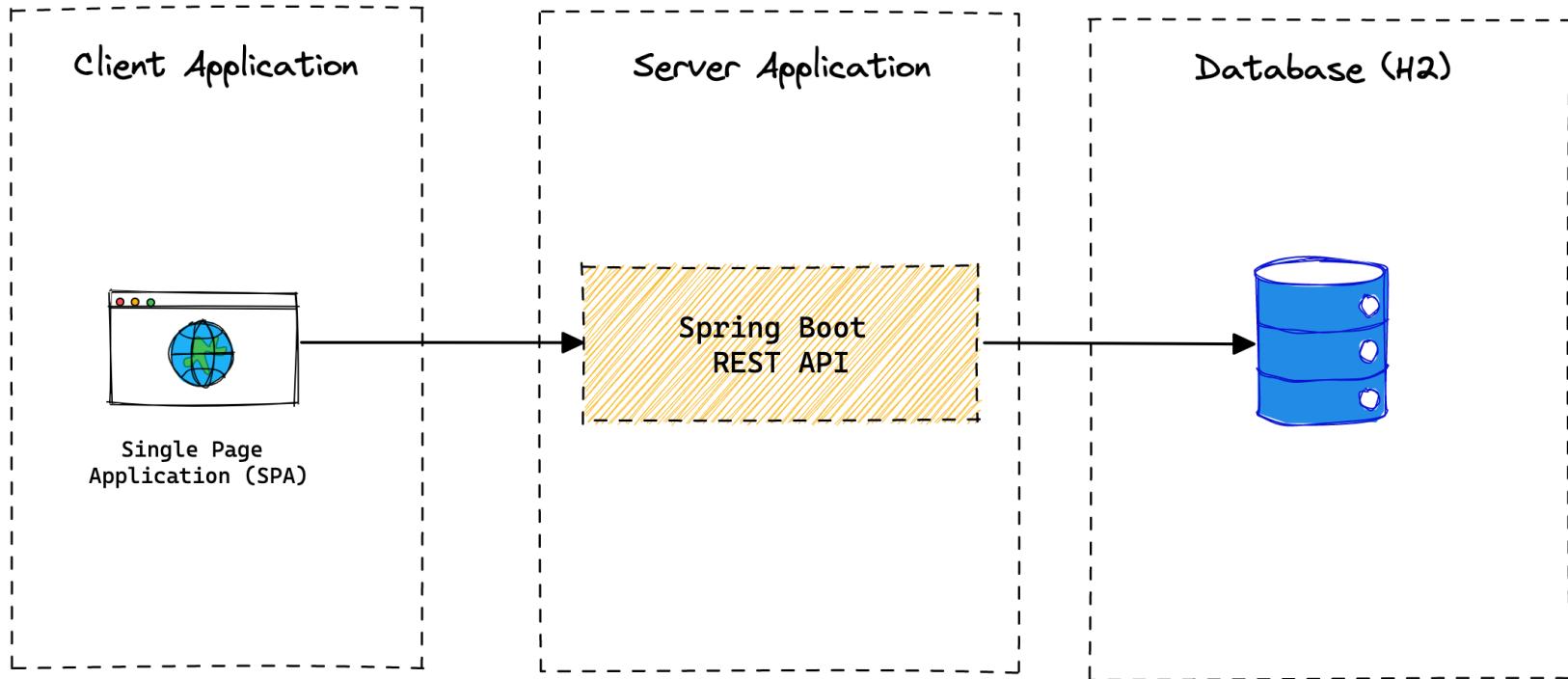




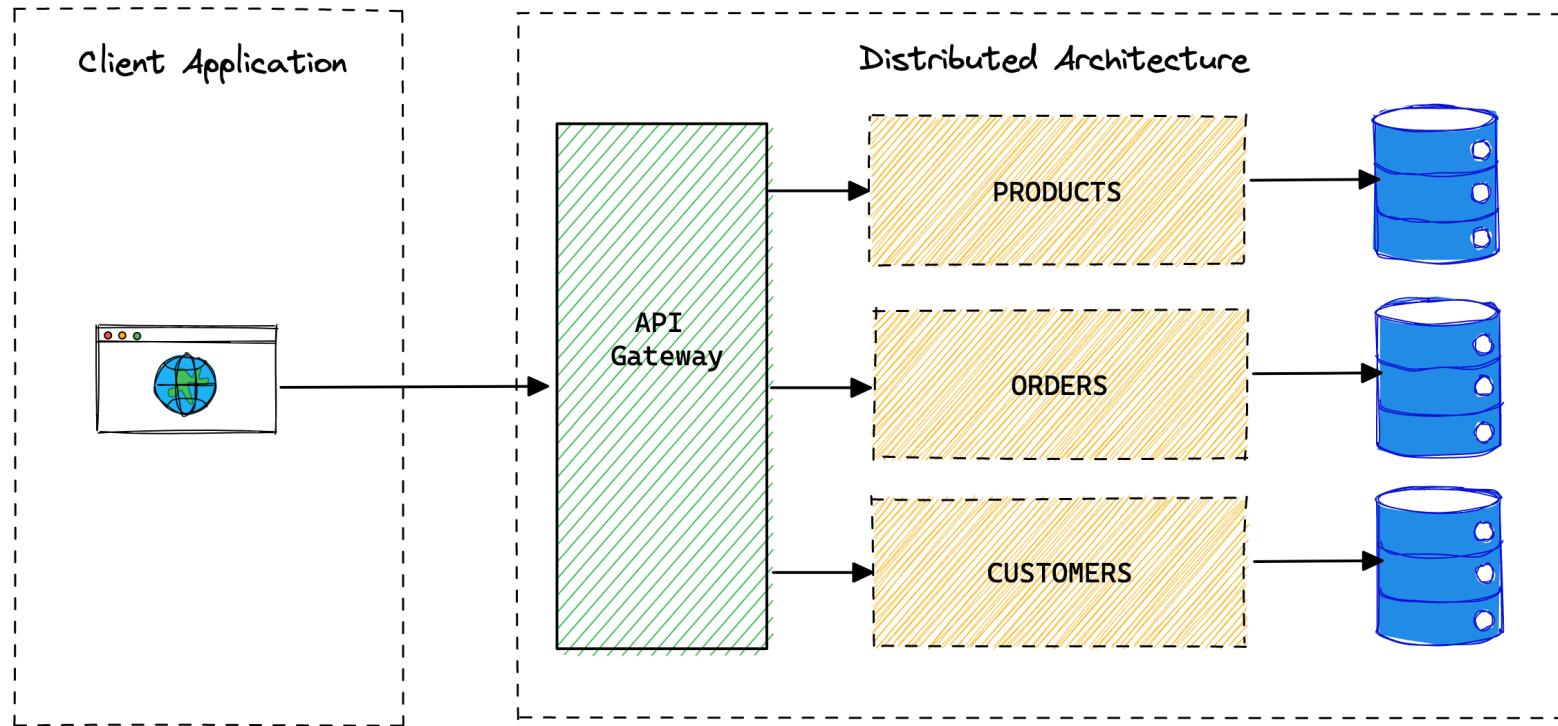
**FRONTEND DEMO**

# 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

The screenshot shows a code editor interface with two main sections: a file tree on the left and a code editor on the right.

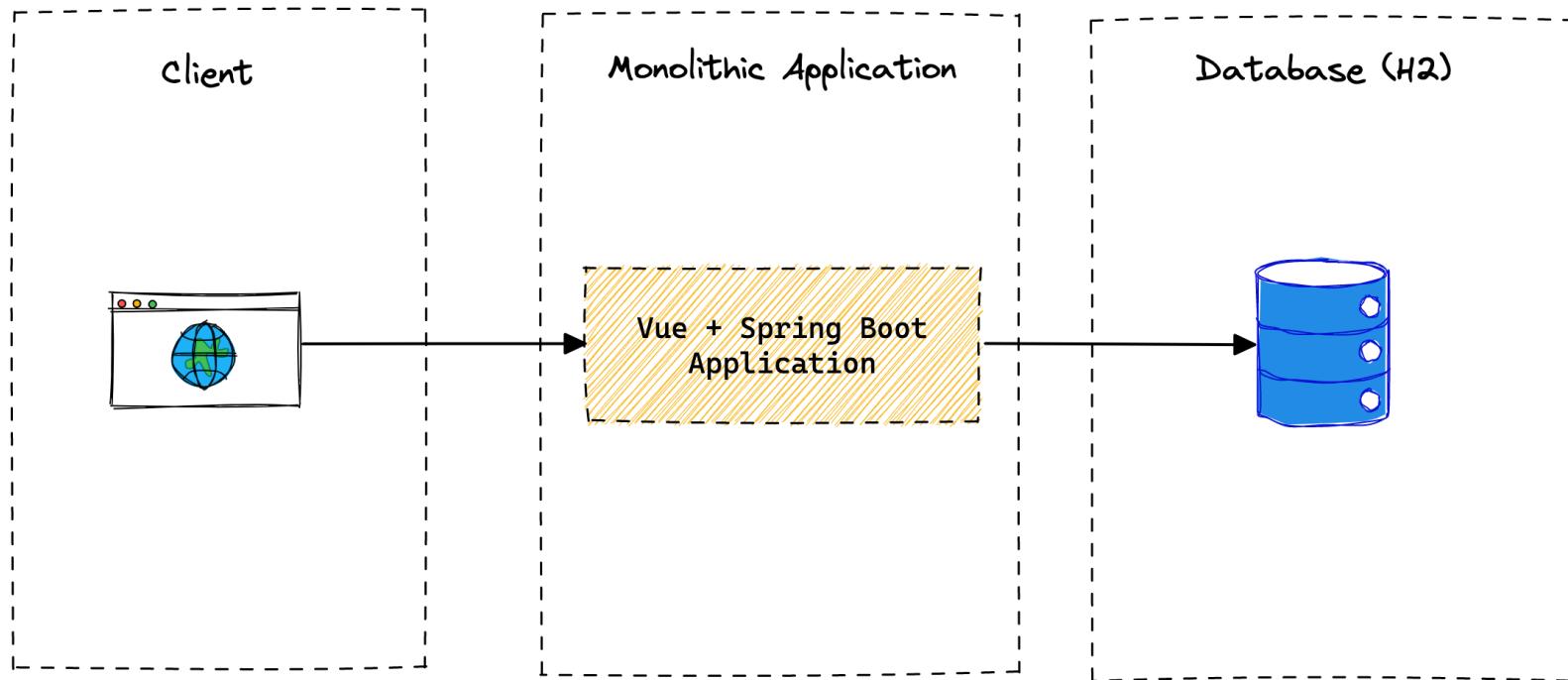
**File Tree (Left):**

- OPEN EDITORS
  - vite.config.js
  - BooksList.vue src/components
- FRONTEND
  - .vscode
  - node\_modules
  - public
    - vite.svg
  - src
    - assets
    - components
      - BooksList.vue
      - HelloWorld.vue
      - App.vue
    - js
      - main.js
    - style.css
    - .gitignore
    - index.html
    - package-lock.json
    - package.json
    - README.md
    - vite.config.js

“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)

The screenshot shows a Java IDE interface with the following details:

- Project View:** Shows the project structure under "reading-list-mono [reading-list]". The "frontend" directory is selected.
- Code Editor:** Displays the file `ReadingListApplication.java`. The code includes annotations like `@SpringBootApplication`, `@Bean`, and `Command`.
- Toolbars and Menus:** Standard IDE toolbars and menus are visible at the top.
- SIDE BAR:** Includes tabs for Project, Commit, Pull Requests, Bookmarks, Structure, and Scratches and Consoles.
- Right Panel:** Shows the line numbers (1-34) and the code content.

# **MONOLITH ARCHITECTURE**

## **DEMO**

# Thank you

Contact me at [dvega@vmware.com](mailto:dvega@vmware.com)  
@therealdanvega



## Single column of content

**When allocating resources for a critical application they were delivering, the company set aside a large team.**

- Which virtualization technologies are in use for application deployment?
- Are any additional virtualization technologies being explored? Public cloud?
- What tools & automation are used for application operations?
- What is the process for maintaining, patching and upgrading environments today?
- Are you using/exploring Continuous Integration and Continuous Delivery pipelines?
- Which app servers and databases are used e.g. Tomcat, Weblogic, Websphere etc?
- How is Business success measured? Key metrics?

### When allocating resources for a critical application they were delivering.

- Which virtualization technologies are in use for application deployment?
- Are any additional virtualization technologies being explored?  
Public cloud?
- What tools & automation are used for application operations?
- What is the process for maintaining, patching and upgrading environments today?

Which virtualization technologies are in use for application deployment?

Are any additional virtualization technologies being explored?  
Public cloud?

What tools & automation are used for application operations?

What is the process for maintaining, patching and upgrading environments today?

## Two columns of content

### Headline for column 1

- Which virtualization technologies are in use for application deployment?
- Are any additional virtualization technologies being explored? Public cloud?
- What tools & automation are used for application operations?
- What is the process for maintaining, patching and upgrading environments today?
- Are you using/exploring Continuous Integration and Continuous Delivery pipelines?

### Headline for column 2

- Which virtualization technologies are in use for application deployment?
- Are any additional virtualization technologies being explored? Public cloud?
- What tools & automation are used for application operations?
- What is the process for maintaining, patching and upgrading environments today?
- Are you using/exploring Continuous Integration and Continuous Delivery pipelines?

## Three columns of content

### Headline for column 1

- Which virtualization technologies are in use for application deployment?
- Are any additional virtualization technologies being explored? Public cloud?
- What tools & automation are used for application operations?
- What is the process for maintaining, patching and upgrading environments today?

### Headline for column 2

- Which virtualization technologies are in use for application deployment?
- Are any additional virtualization technologies being explored? Public cloud?
- What tools & automation are used for application operations?
- What is the process for maintaining, patching and upgrading environments today?
- Are you using/exploring Continuous Integration and Continuous Delivery pipelines?

### Headline for column 3

- Which virtualization technologies are in use for application deployment?
- Are any additional virtualization technologies being explored? Public cloud?
- What tools & automation are used for application operations?

# Centered Text Template

# Spring Logo and Project Icons



Spring  
Boot



Spring  
Cloud



Spring  
Framework



Spring Cloud  
Data Flow



Spring Tool  
Suite



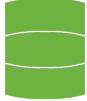
Spring  
LDAP



Spring  
Cloud Gateway



Spring  
Security



Spring  
Data



Spring  
Batch



Spring  
Integration



Project  
Reactor



Spring  
Kafka



Spring  
for GraphQL



Spring  
Web Services



Spring  
Web Flow



Spring  
Hateoas



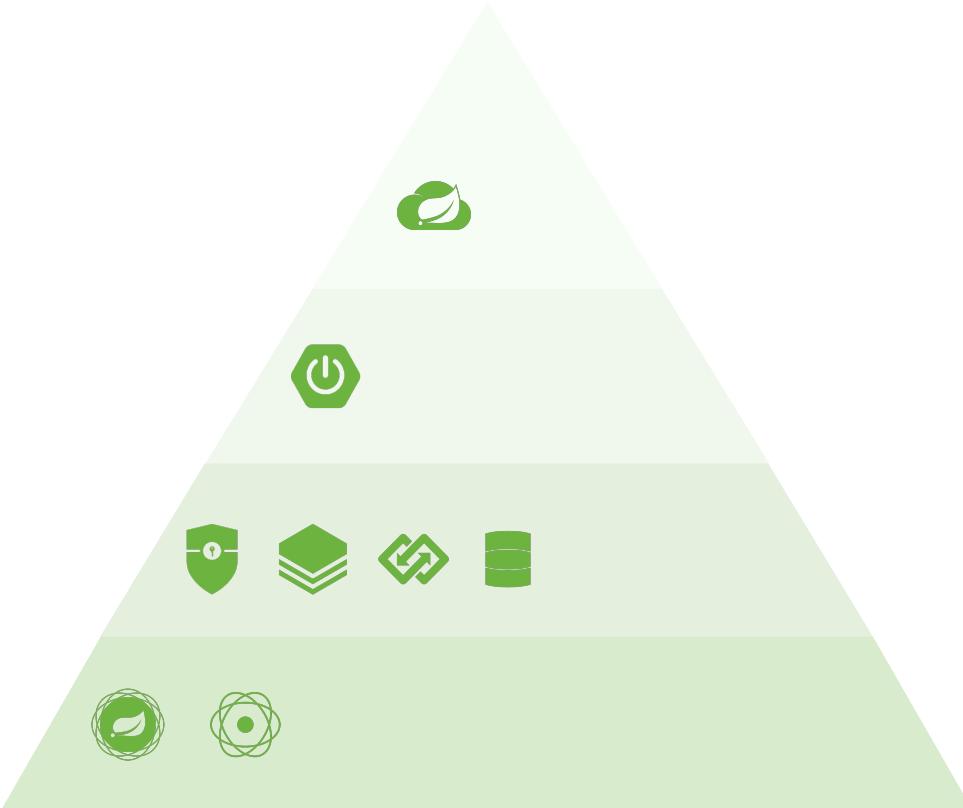
Spring  
AMQP



## Spring Project Icons Hierarchy Template #1



## Spring Project Icons Hierarchy Template #2



## Code Block Template #1

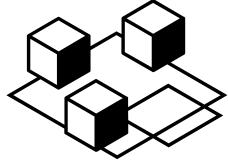
```
@SpringBootApplication  
@RestController  
public class DemoApplication {  
  
    @GetMapping("/helloworld")  
    public String hello() {  
        return "Hello World!";  
    }  
}
```



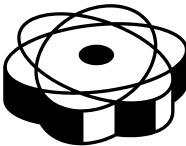
# Code Block Template #2

```
@SpringBootApplication  
@RestController  
public class DemoApplication {  
  
    @GetMapping("/helloworld")  
    public String hello() {  
        return "Hello World!";  
    }  
}
```

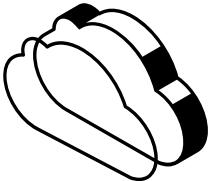
## Icons



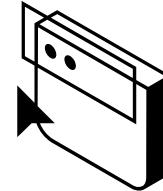
Microservices



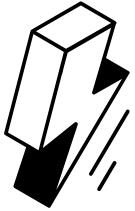
Reactive



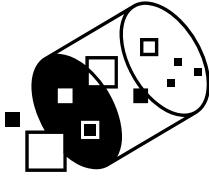
Cloud



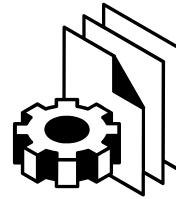
Web Apps



Serverless

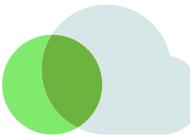
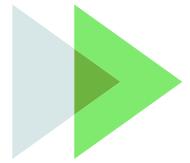


Event Driven

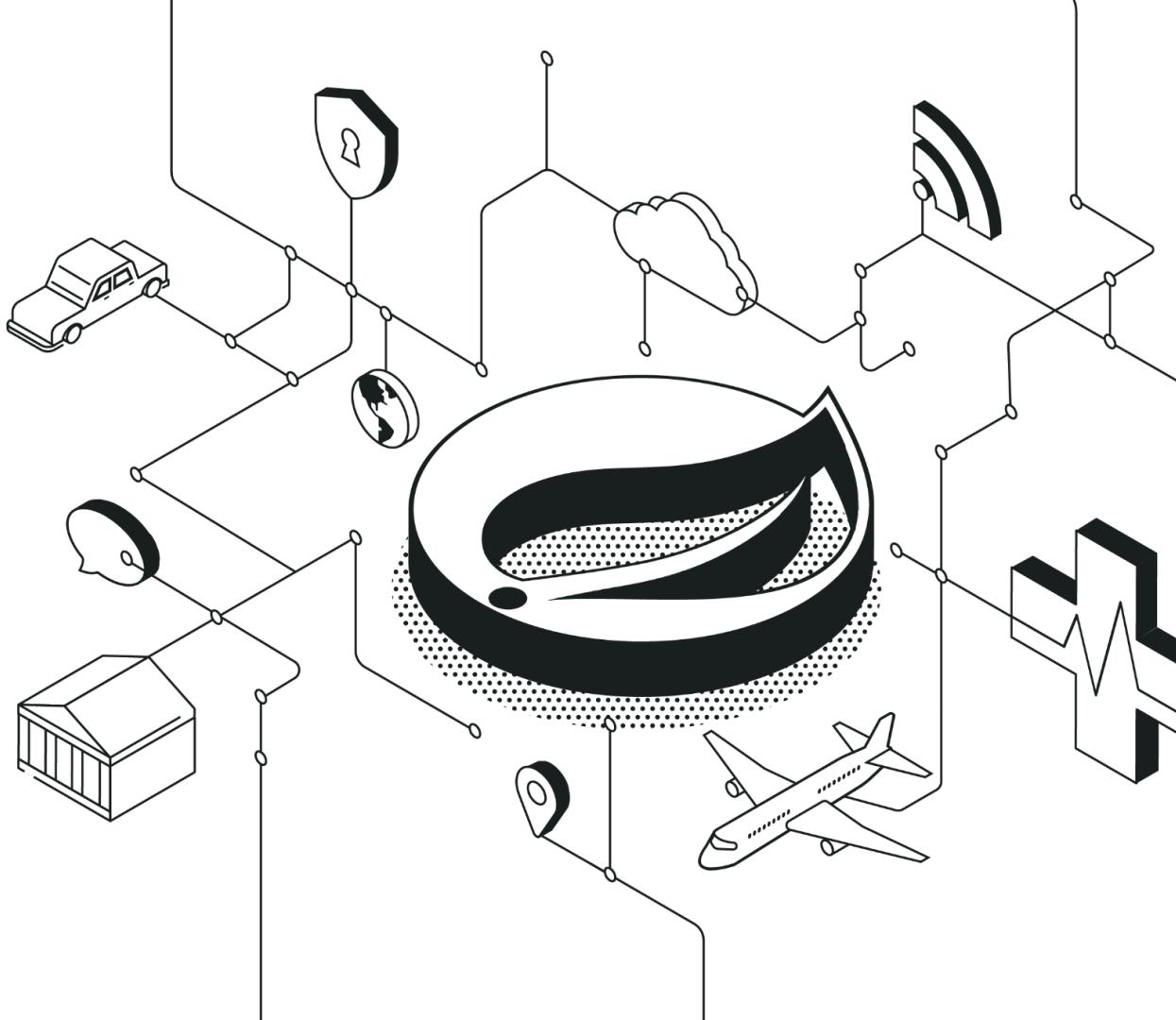


Batch

## Abstract Icons



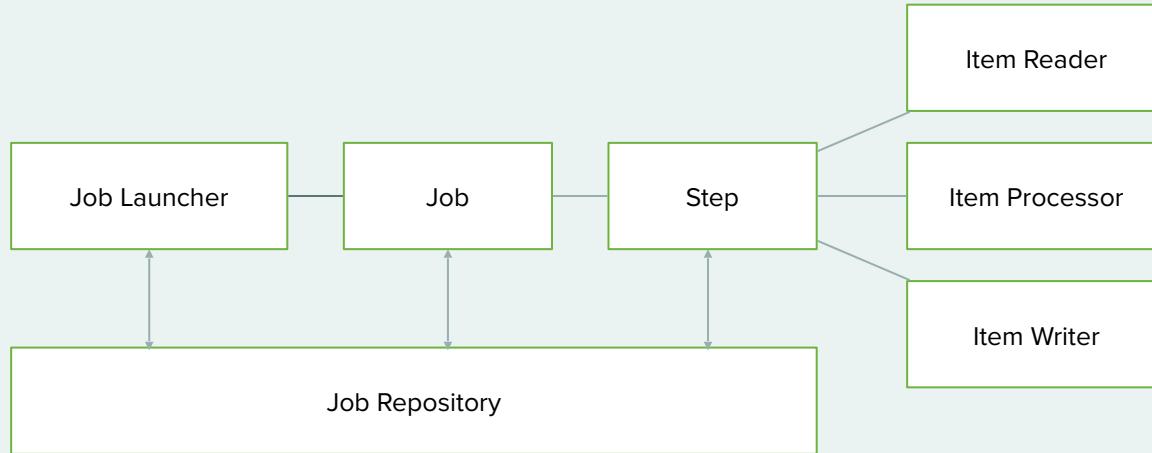
## Why Spring?



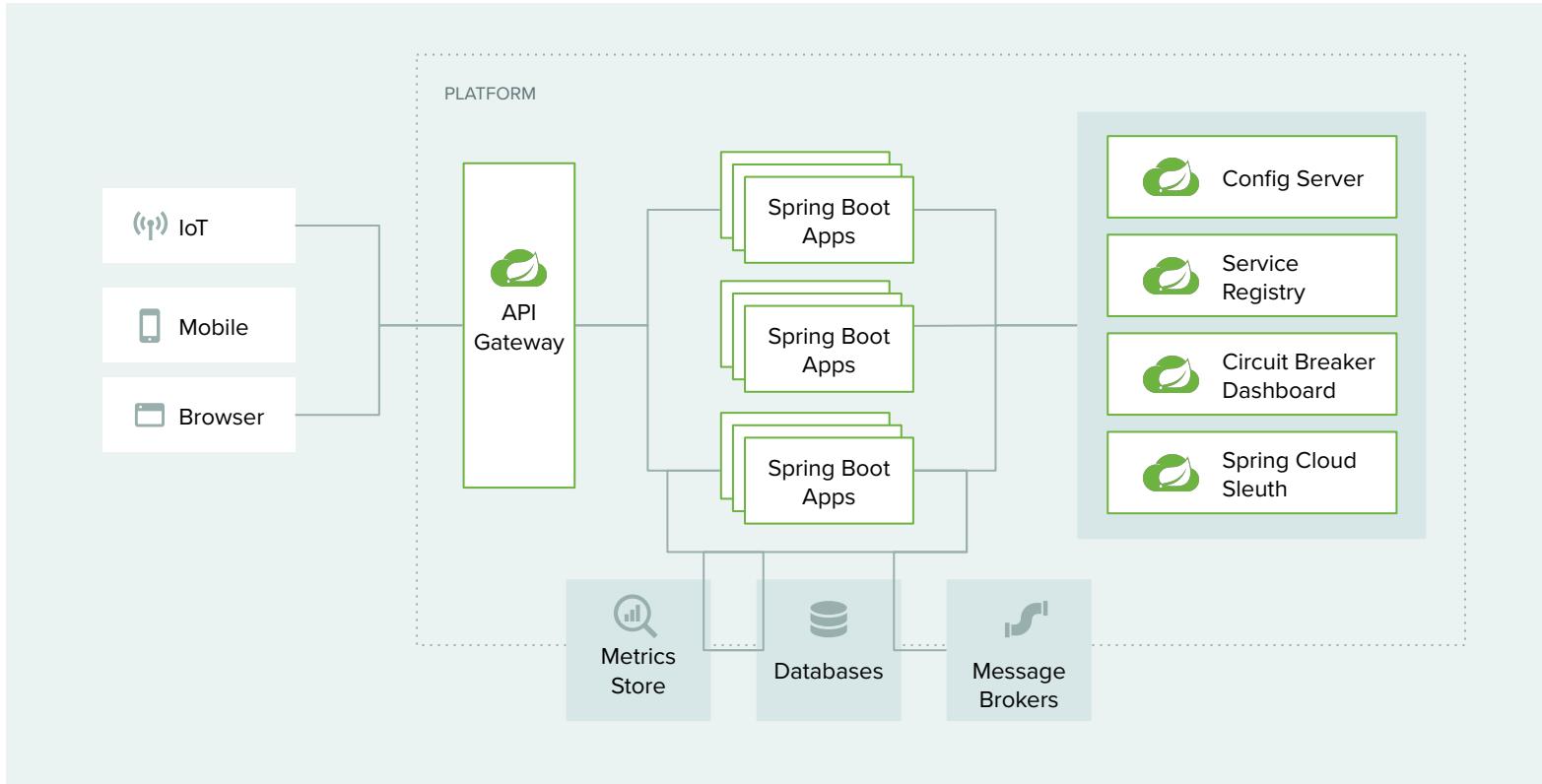
# Diagram Kit

## Diagram Examples

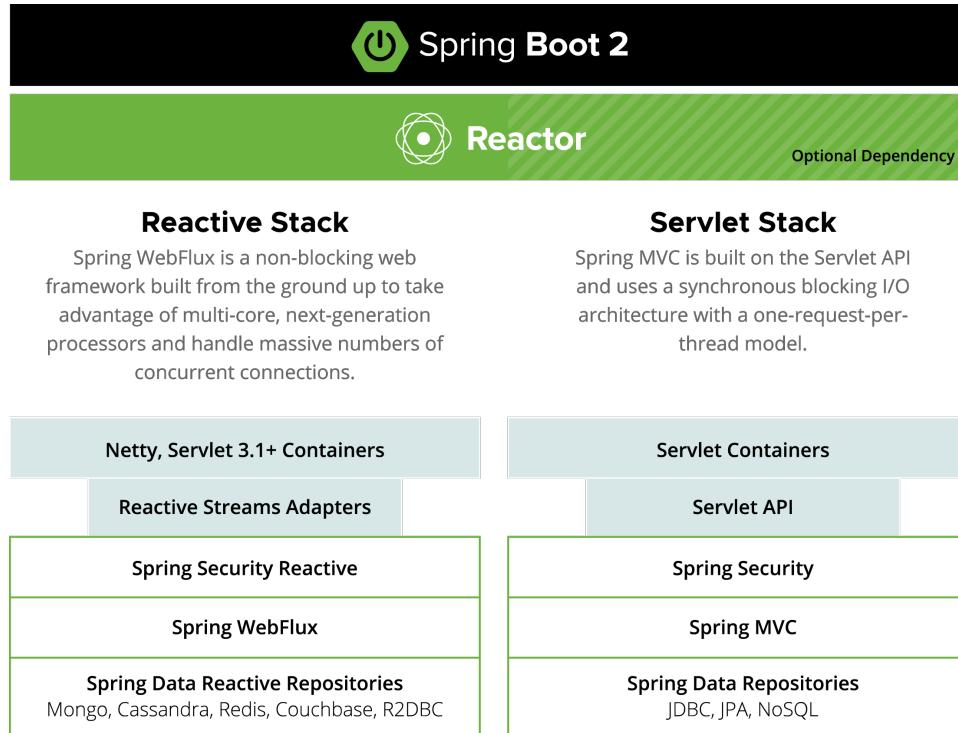
### Batch Stereotypes



## Diagram Examples



## Diagram Examples



# Resources

The screenshot shows the homepage of danvega.com. At the top, there is a navigation bar with links for "HOME", "BLOG", "COURSES", "NEWSLETTER", and "SPEAKING". A search bar and a moon icon are also present. The main title "LEARN TO CODE" is displayed above the subtitle "with Dan Vega". Below the subtitle, a bio states: "I am passionate about software development, learning new things, and teaching them to you. If you want to learn how to code subscribe to my weekly newsletter for the latest articles and tutorials." There is a "Subscribe" button and an input field for an email address. To the right, there is an illustration of a person standing next to a large computer monitor displaying a video call interface. At the bottom left, there is a section titled "HI, I'M DAN VEGA" with a bio and a cartoon portrait of the author.

dan vega [HOME](#) [BLOG](#) [COURSES](#) [NEWSLETTER](#) [SPEAKING](#)

**LEARN TO CODE**  
with Dan Vega

I am passionate about software development, learning new things, and teaching them to you. If you want to learn how to code subscribe to my weekly newsletter for the latest articles and tutorials.

Email Address  [Subscribe](#)

**HI, I'M DAN VEGA**

I'm a Husband, Father, [Spring Developer Advocate](#) and maker of things from Cleveland Ohio. I created this website as a place to document my journey as I learn new things and share them with you. I have a real passion for teaching and I hope that one of [blog posts](#), [videos](#) or [courses](#) helps you solve a problem or learn something new.

