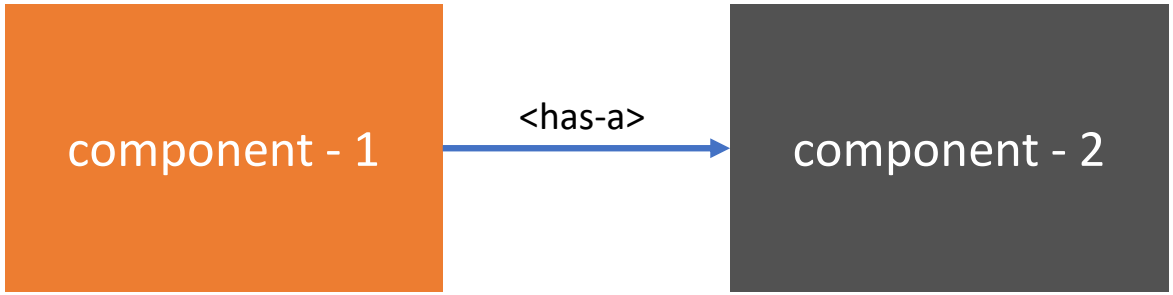
Introduction to Spring Platform

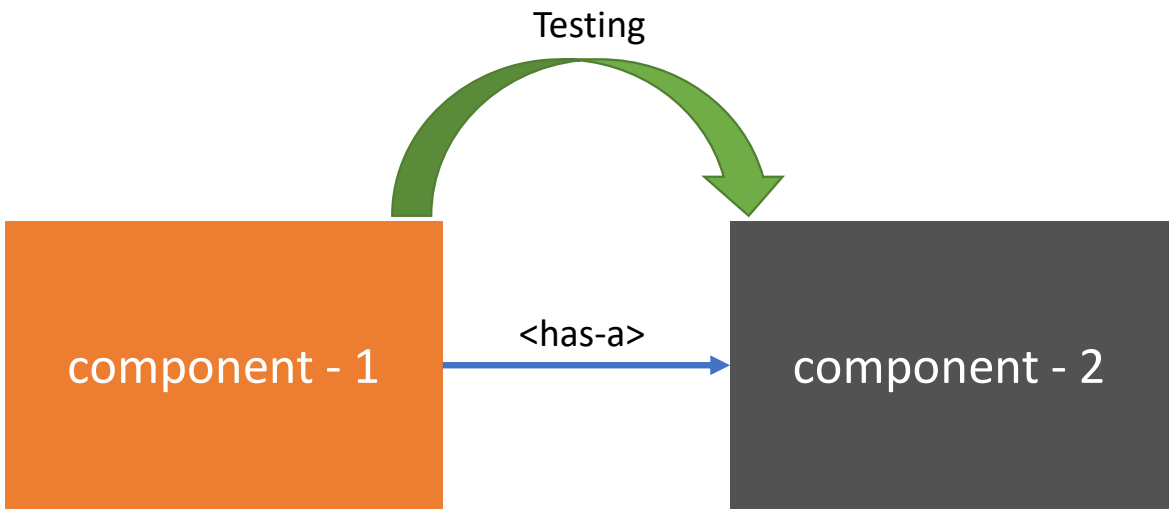
Spring Platform

- → Ecosystem of various spring modules
- → Plug and play architecture
- → To build production grade standalone and web applications

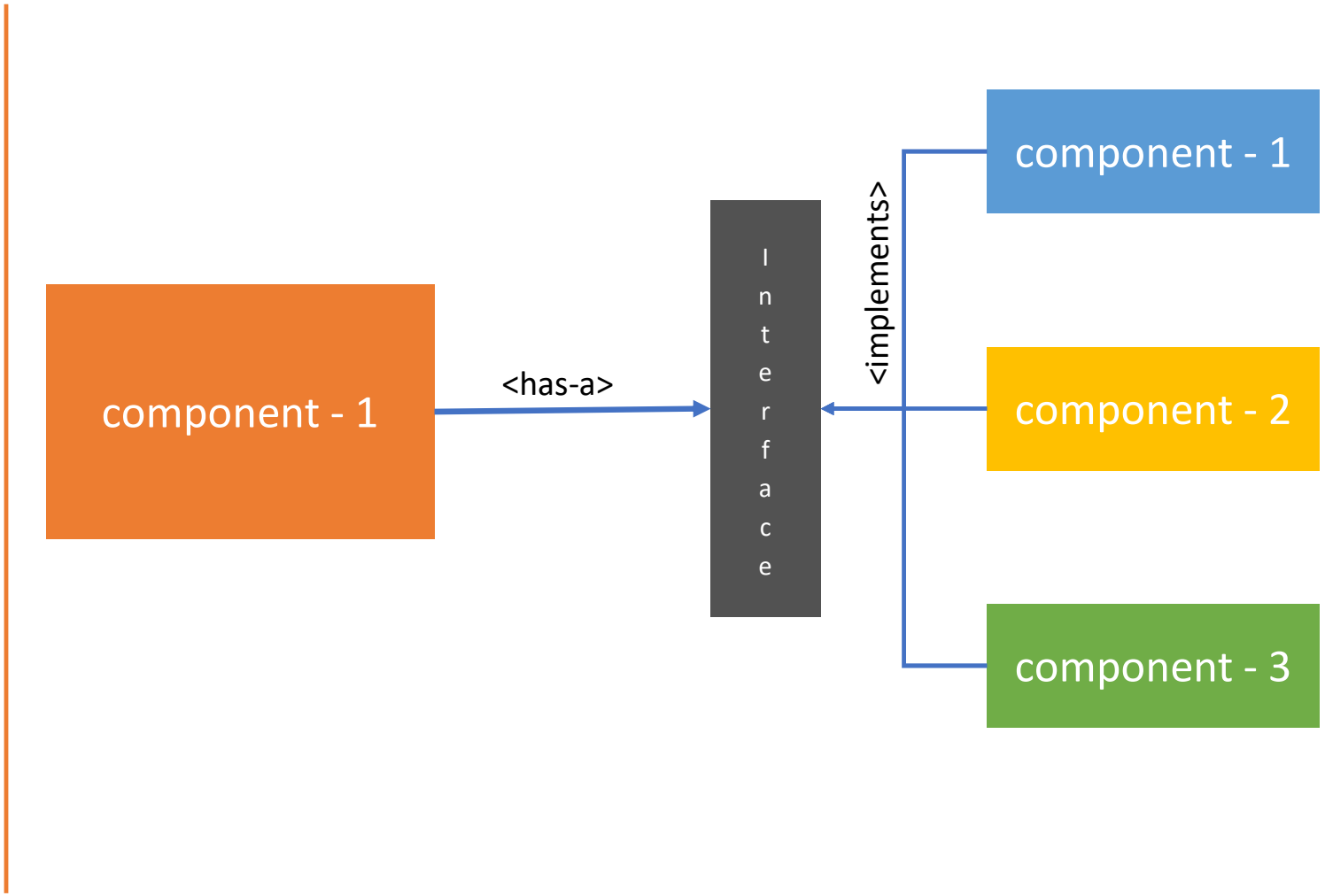
Coupling



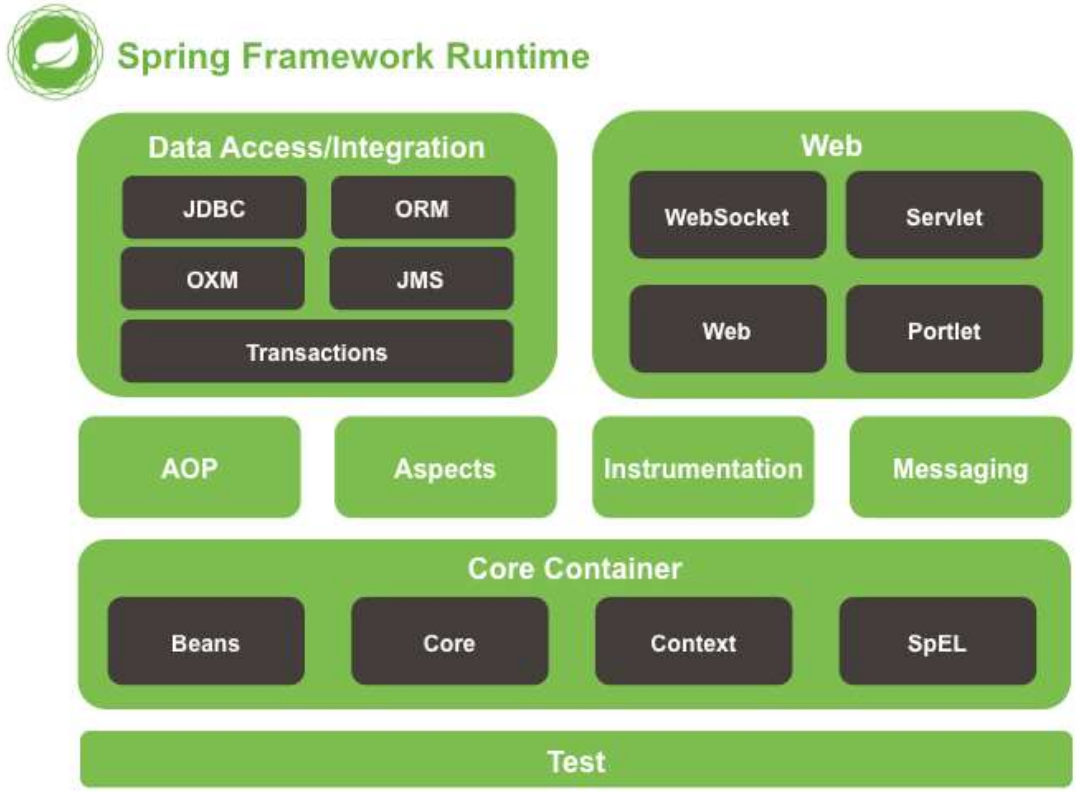
Coupling



Coupling


component - 1
<has-a>
Interface
<implements>
component - 1
component - 2
component - 3

Spring Platform

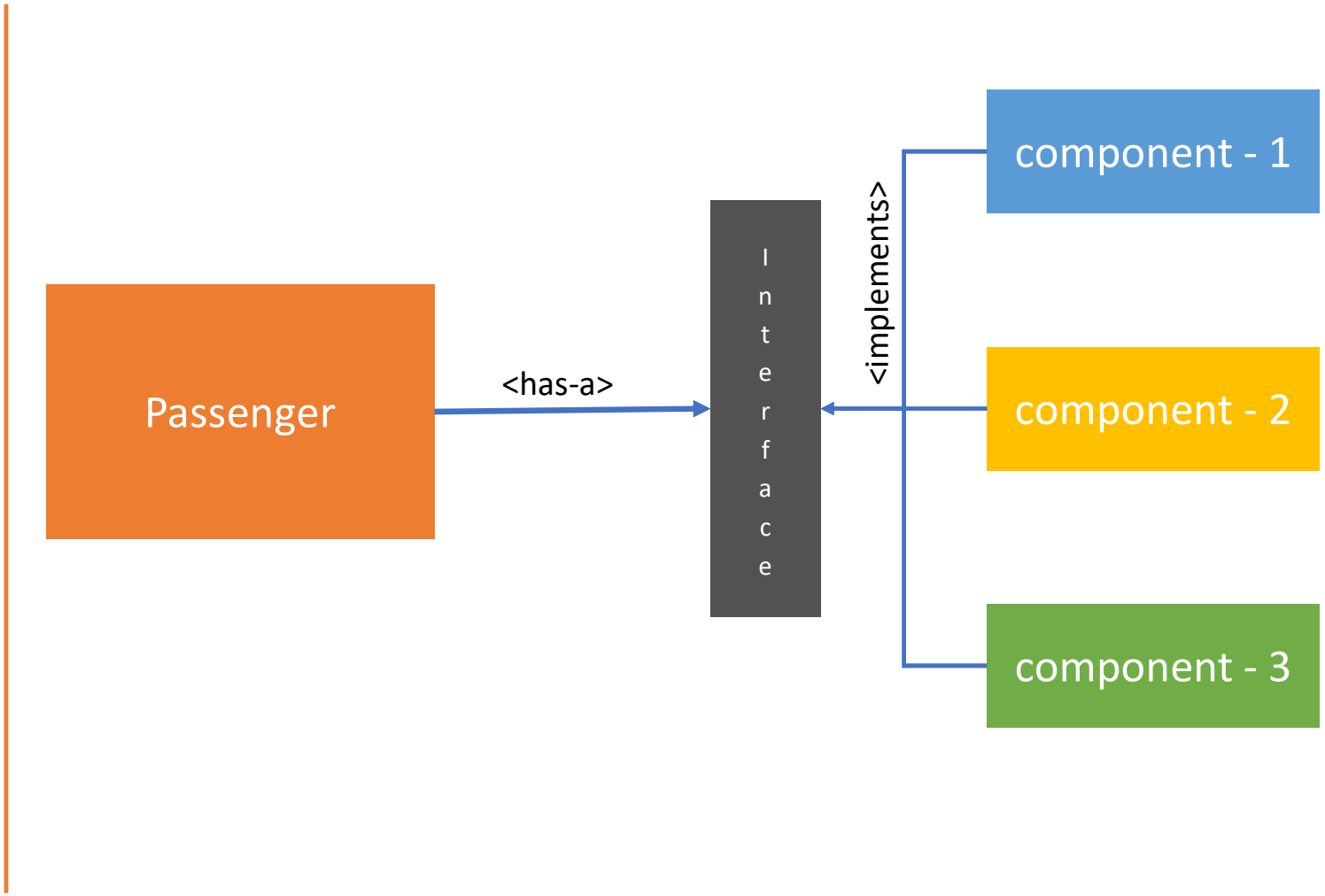


Motivation to adapt Spring Platform

- → No need to extend or implement framework specific classes and interfaces
- → Seamless Integration with 3rd party frameworks and libraries
- → Rich community support
- → Comprehensive documentation and how-to guides

Case-Study

Case-study


Passenger
<has-a>
Interface
<implements>
component - 1
component - 2
component - 3

Spring Platform

- → Scaffolding project structure
- → Dependencies management
- → Infrastructure dependencies management
- → Configuration
- → Packaging
- → Deployment
- → Monitoring

Challenges

- → Scaffolding project structure
- → Dependencies management
- → Infrastructure dependencies management
- → Configuration
- → Packaging
- → Deployment
- → Monitoring

Challenges

- → Scaffolding project structure
- → Dependencies management
- → Infrastructure dependencies management
- → Configuration
- → Packaging
- → Deployment
- → Monitoring

Scaffolding

- → Maven or Gradle project
- → Setting up VCS integration
- → Maven binary installation
- → Starting the application

Dependency management

- → Maven/Gradle dependencies
- → BOMs compatibility
- → NoClassDefoundError

Infrastructure Dependency management

- → Setting up infrastructure with Application Context
- → Repetitive
- → Static configuration
- → Difficult to maintain and scale

Packaging and deployment

- → Packaged as war file
- → Reduced dev-ops parity
- → Configuration drift
- → Not suitable for containers/cloud-native apps

Monitoring

- → Reactive approach for incident management
- → Instrumentation using JMX/external dependencies
- → Manual and not scalable for cloud-native deployments