

Spring Professional Exam Tutorial v5.0

Question 05

Question 05 - What is a security context?

`SecurityContext` is an interface, allowing you to access security information associated with the current thread of execution. Interface provides two methods:

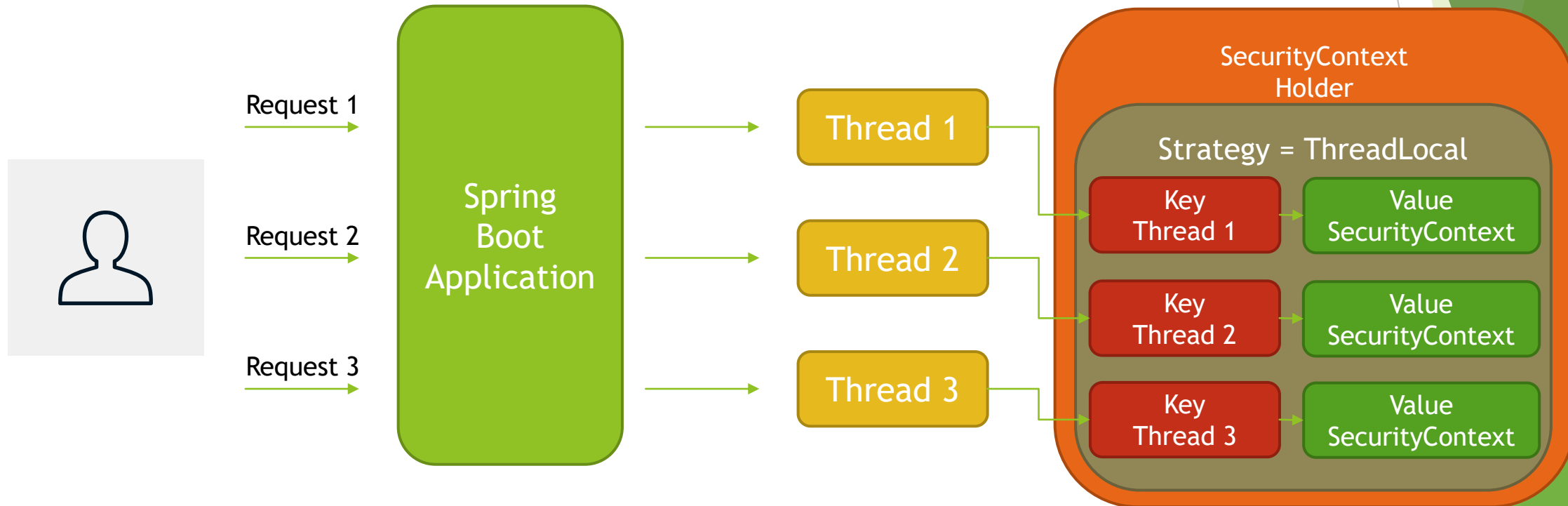
- ▶ `getAuthentication` - provides currently authenticated principal, or an authentication request token
- ▶ `setAuthentication` - sets currently authenticated principal, or removes the authentication

`SecurityContext` can be accessed via `SecurityContextHolder`, which allows access in three modes:

- ▶ `MODE_THREADLOCAL`
- ▶ `MODE_INHERITABLETHREADLOCAL`
- ▶ `MODE_GLOBAL`

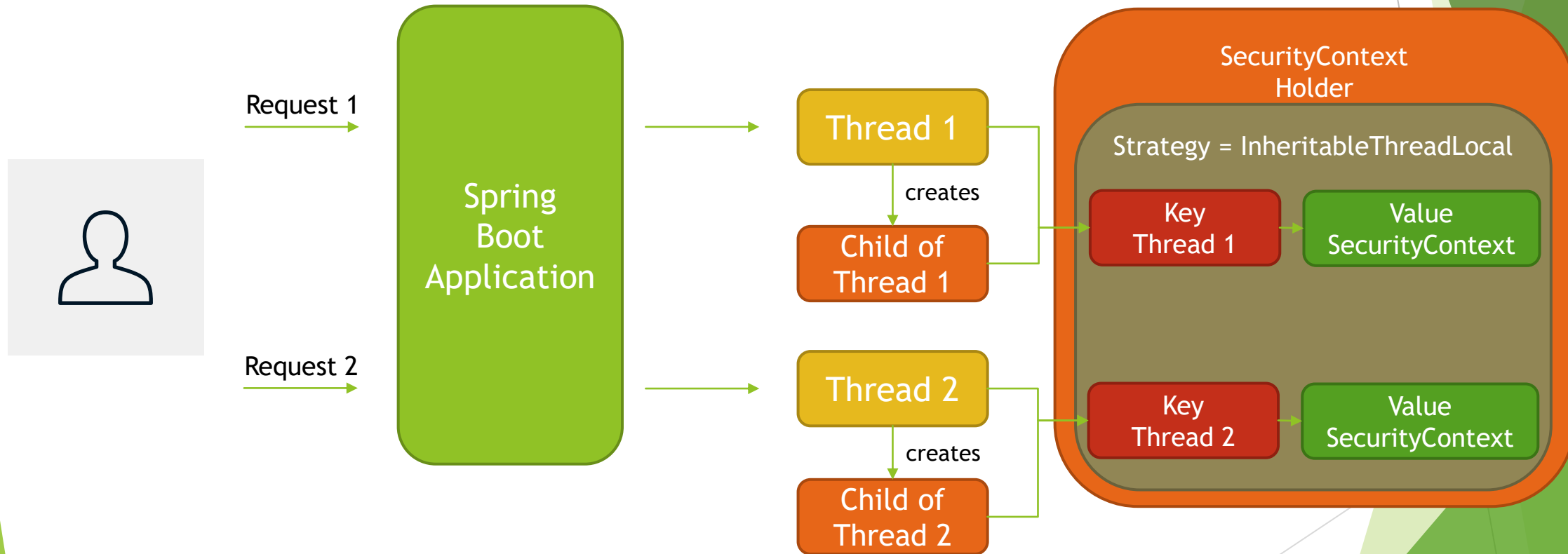
Question 05 - What is a security context?

`SecurityContextHolder` is set to `MODE_THREADLOCAL` by default, it allows each Thread to access it's own dedicated `SecurityContext`.



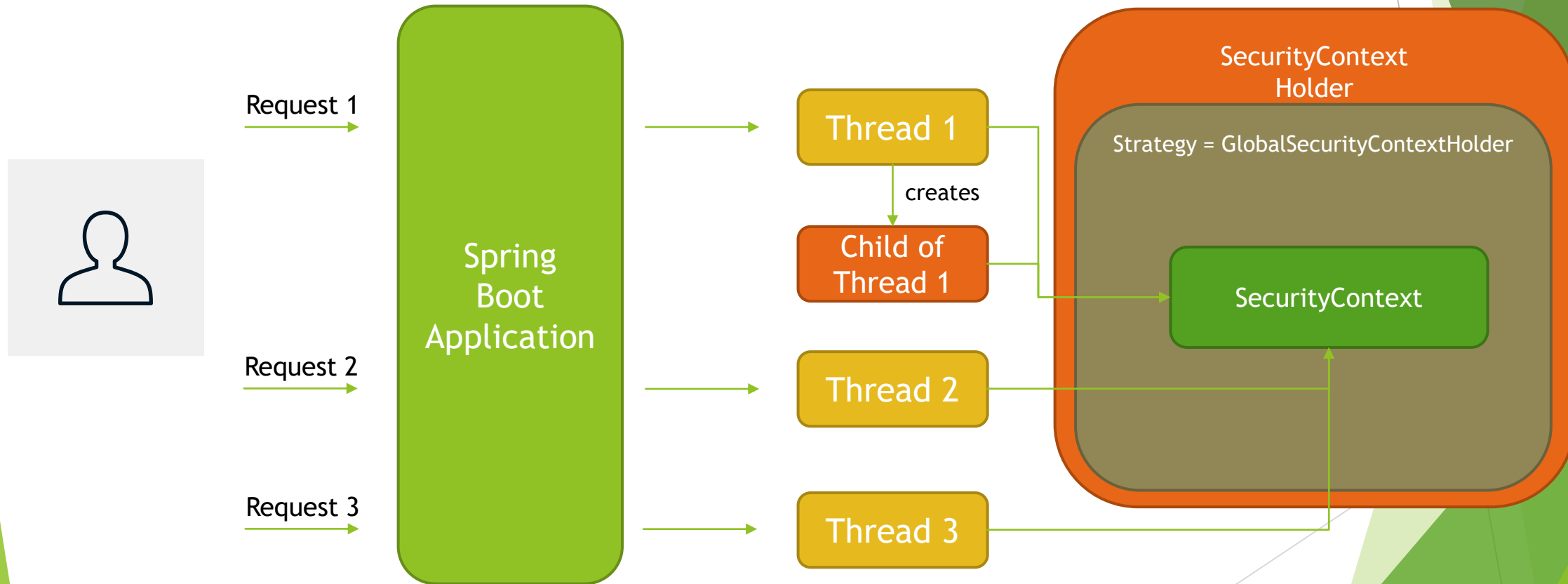
Question 05 - What is a security context?

When `SecurityContextHolder` is set to `MODE_INHERITABLETHREADLOCAL`, then child threads are allowed to access same `SecurityContext` as parent Thread.



Question 05 - What is a security context?

When `SecurityContextHolder` is set to `MODE_GLOBAL`, then all threads within JVM are accessing same `SecurityContext`, this mode is usually used by standalone desktop applications, such as Java Swing based clients.



Question 05 - What is a security context?

Mode of `SecurityContextHolder` can be changed via:

- ▶ **Property** - `spring.security.strategy`
- ▶ **Programmatically** - `SecurityContextHolder#setStrategyName`

Allowed values are:

- ▶ `MODE_THREADLOCAL` (**default**)
- ▶ `MODE_INHERITABLETHREADLOCAL`
- ▶ `MODE_GLOBAL`

Question 05 - What is a security context?

`SecurityContext` allows you to access or set currently used `Authentication` object.

`Authentication` object represents the token for an authentication request or an authenticated principal, once the authentication request has been processed. `Authentication` object contains following methods:

- ▶ `getPrincipal` - identity of the principal being authenticated
- ▶ `getAuthorities` - collection of authorities granted to the principal
- ▶ `getCredentials` - credentials that prove the principal is correct
- ▶ `getDetails` - additional details about the authentication request, address, certificate, serial number etc.
- ▶ `isAuthenticated`, `setAuthenticated` - for internal use by `AbstractSecurityInterceptor`, `AuthenticationManager` to control required amount of for re-authentications

