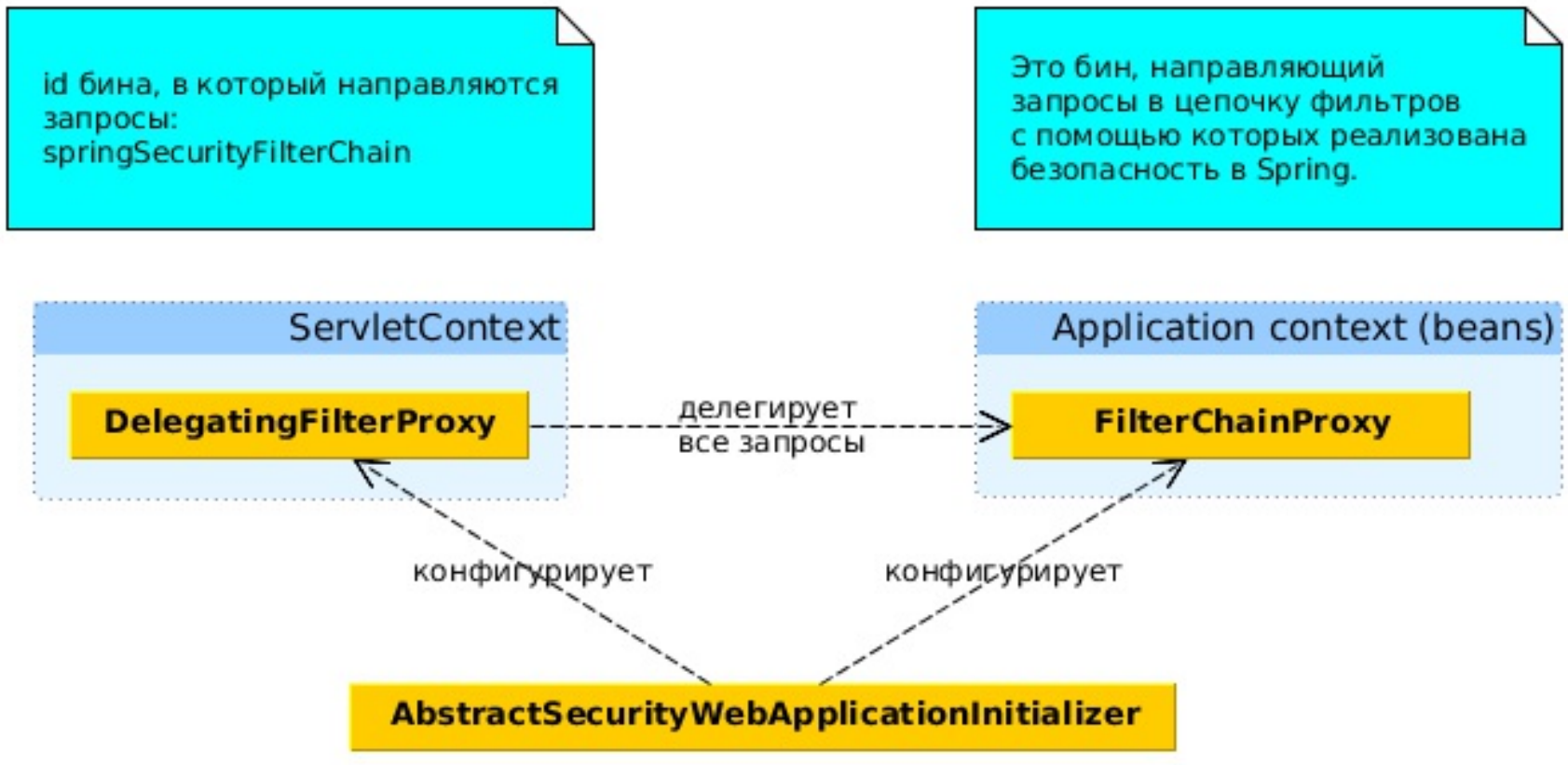
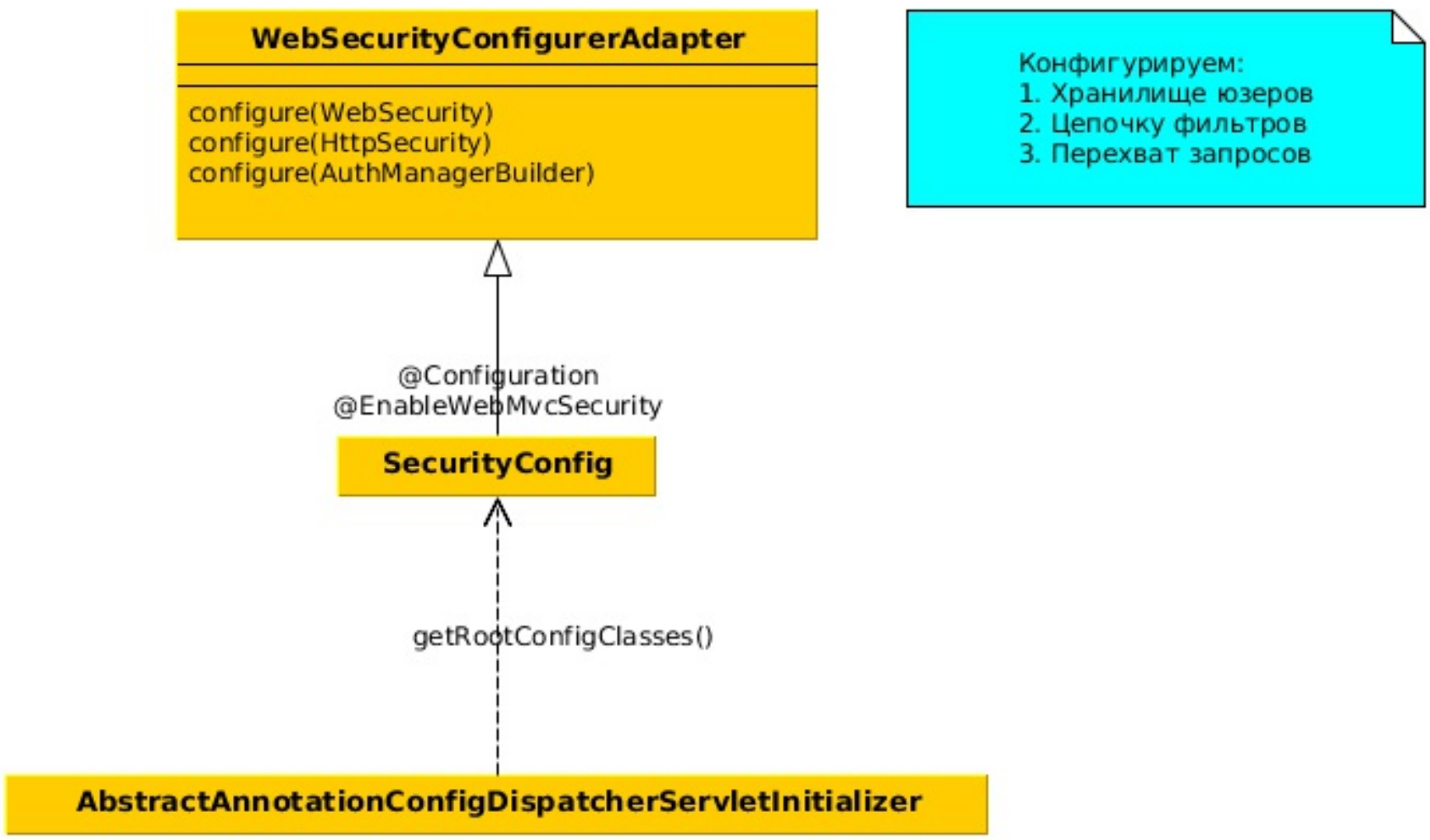
Web secutiry

Table 9.1 Spring Security is partitioned into eleven modules

Module	Description
ACL	Provides support for domain object security through access control lists (ACLs).
Aspects	A small module providing support for AspectJ-based aspects instead of standard Spring AOP when using Spring Security annotations.
CAS Client	Support for single sign-on authentication using Jasig's Central Authentication Service (CAS).
Configuration	Contains support for configuring Spring Security with XML and Java. (Java configuration support introduced in Spring Security 3.2.)
Core	Provides the essential Spring Security library.
Cryptography	Provides support for encryption and password encoding.
LDAP	Provides support for LDAP-based authentication.
OpenID	Contains support for centralized authentication with OpenID.
Remoting	Provides integration with Spring Remoting.
Tag Library	Spring Security's JSP tag library.
Web	Provides Spring Security's filter-based web security support.

- Для веб-приложения нужно добавить в CP: core,configuration,web.
- Web-безопасность в спринг реализована как набор web-фильтров. DelagatingFilterProxy направляет все запросы в специальный бин (id = springSecurityFilterChain), в качестве которого обычно выступает FilterChainProxy.
- FilterChainProxy направляет запрос в цепочку secutiry фильтров Spring.





- DelagatingFilterProxy конфигурируется либо в web xml, либо программно.
- Конфигурируем security в WebMvc-приложении:

```
@Configuration
@EnableWebMvcSecurity
//@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {
}
```

Table 9.2 Overriding `WebSecurityConfigurerAdapter`'s configure() methods

Method	Description
`configure(WebSecurity)`	Override to configure Spring Security's filter chain.
`configure(HttpSecurity)`	Override to configure how requests are secured by interceptors.
`configure(AuthenticationManagerBuilder)`	Override to configure user-details services.

- Возможно хранить аутентификационные данные в памяти (см код SecurityConfig), методы хранилища в памяти:

Table 9.3 Methods for configuring user details

Module	Description
`accountExpired(boolean)`	Defines if the account is expired or not
`accountLocked(boolean)`	Defines if the account is locked or not
`and()`	Used for chaining configuration
`authorities(GrantedAuthority…)`	Specifies one or more authorities to grant to the user
`authorities(List<? extends GrantedAuthority>)`	Specifies one or more authorities to grant to the user
`authorities(String…)`	Specifies one or more authorities to grant to the user
`credentialsExpired(boolean)`	Defines if the credentials are expired or not
`disabled(boolean)`	Defines if the account is disabled or not
`password(String)`	Specifies the user's password
`roles(String…)`	Specifies one or more roles to assign to the user

- Role это GrantedAuthority с префиксом ROLE_
- GrantedAuthority это некоторое разрешение или "право".
- UserDetailsService - это интерфейс, используемый для поиска пользователей в некотором хранилище данных:

```
public interface UserDetailsService {
    UserDetails loadUserByUsername(String username)
      throws UsernameNotFoundException;
}
```


Находит пользователя
по его имени в неком хранилище.
UserDetailsService
WebSecurityConfigurerAdapter
SpitterRepository
SpitterUserDetailsService
SecurityConfig