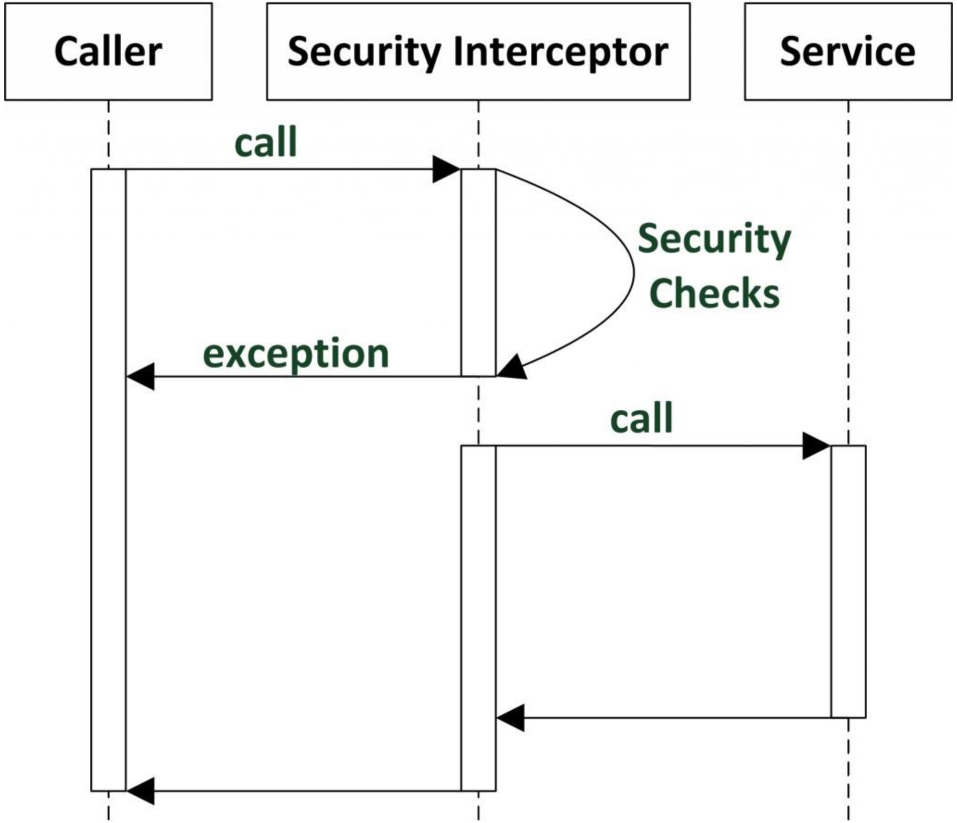
SPRING SECURITY

METHOD SECURITY

MARCH, 2015

METHOD AUTHORIZATION



METHOD SECURITY CONFIGURATION

```xml
<sec:global-method-security
                      access-decision-manager-ref="accessDecisionManager">
    <sec:protect-pointcut expression="execution(* *user.*Service.*(..))"
                      access="ROLE_USER"/>
    <sec:protect-pointcut expression="execution(* *admin.*Service.*(..))"
                      access="ROLE_ADMIN"/>
</sec:global-method-security>

<bean id="accountService" class="org.training.AccountServiceImpl">
    <sec:intercept-methods>
        <sec:protect access="ROLE_USER" method="createAccount"/>
        <sec:protect access="ROLE_ADMIN" method="delete*"/>
    </sec:intercept-methods>
</bean>
```

METHOD SECURITY ANNOTATIONS

```
<global-method-security
          access-decision-manager-ref="accessDecisionManager“
          secured-annotations="enabled">
</global-method-security>
```

```
public interface BankService {

  @Secured(“ROLE_USER")
  public Account readAccount(Long id);

  @Secured("ROLE_MANAGER")
  public Account post(Account account, double amount);
 }
```

METHOD SECURITY ANNOTATIONS

```
<global-method-security
          access-decision-manager-ref="accessDecisionManager"
          pre-post-annotations="enabled">
</global-method-security>
```

```
public interface BankService {

   @PreAuthorize("isAnonymous()")
   public Account[] findAccounts();

   @PreAuthorize("hasAuthority('ROLE_MANAGER')")
   public Account post(Account account, double amount);
 }
```

ACL FILTERING ANNOTATIONS

```
<sec:global-method-security pre-post-annotations="enabled">
    <sec:expression-handler ref="expressionHandler"/>
</sec:global-method-security>

<bean id="expressionHandler“ class="…DefaultMethodSecurityExpressionHandler">
    <property name="permissionEvaluator" ref="permissionEvaluator"/>
</bean>

<bean id="permissionEvaluator" class="…AclPermissionEvaluator">
    <constructor-arg ref="aclService"/>
</bean>
```

```
@PreFilter(filterTarget = "customers", value = "hasPermission(filterObject, 'update')")
public void updateCustomers(List<Customer> customers, State st) {
}

@PreAuthorize("hasRole('ROLE_USER')")
@PostFilter("hasPermission(filterObject, 'read') or hasPermission(filterObject, 'admin')")
public List<Contact> getAll();
```

THANK YOU!

MAKSYM_GOVORISCHEV@EPAM.COM

MARCH, 2015