Scheduling

A especificação EJB fornece a possibilidade de se disparar um processo em um determinado período, ou seja, sem que um usuário ou um outro sistema tenha explicitamente uma ação.

Em EJB 3.1, ficou mais simples com a anotação @Schedule

```java
@Stateless
@LocalBean
public class StatsService {

    @Schedule  //meia noite
    public void buildStats () {

        ...

        ...

    }
```

```
@Stateless
@LocalBean
public class StatsService {

    @Schedule (minute="10", hour="8-12")
    public void buildStats () { //8:10, 9:10, ..., 12:10.

        ...

        ...

    }
```

```java
@Stateless
@LocalBean
public class StatsService {

    @Schedule (minute="*/15", hour="*")
    public void buildStats () { //a cada 15min

        ...

        ...

    }
```

```
@Stateless
@LocalBean
public class StatsService {


    @Schedule (month="1,2,6,8")
    public void buildStats () { //Jan, Fev, Jun, Ago

        ...

        ...

    }
```

```
@Stateless
@LocalBean
public class StatsService {


    @Schedule (month="Jan,Feb,Jun,Aug")
    public void buildStats () { //Jan, Fev, Jun, Ago

        ...

        ...

    }
```

```
@Stateless
@LocalBean
public class StatsService {

    @Schedule (dayOfWeek="0, 1, 4-6")
    public void buildStats () {//dom, seg, qui, sex, sab

        ...

        ...

    }
```

```
@Stateless
@LocalBean
public class StatsService {

    @Schedule(second="*", minute="*", hour="*",
    dayOfMonth="*", month="*", dayOfWeek="*", year="*")
    public void buildStats () {  // cada segundo

        ...

        ...

    }
```

```
@Stateless
@LocalBean
public class StatsService {

    @Schedule( ..., persistent = false)
    public void buildStats () {  // não persistente

        ...

        ...

    }
```

Exercícios (8)

- Crie um EJB (HeapMonitor) com um método agendado capaz de 'logar' as estatísticas de memória da JVM.
  - Use a classe Runtime (consulte o javadoc)
  - As estatísticas devem ser logadas a cada minuto.
  - A unidade deve ser Mega Bytes.
  - O timer não deve ser persistence.

Exercícios (8.cont)

[Estatísticas de Memória]

[Memória total: X MB]

[Memória usada: X MB]

[Memória livre: X MB]

[Memória máxima: X MB]

Segurança

Como Proteger Componentes EJB

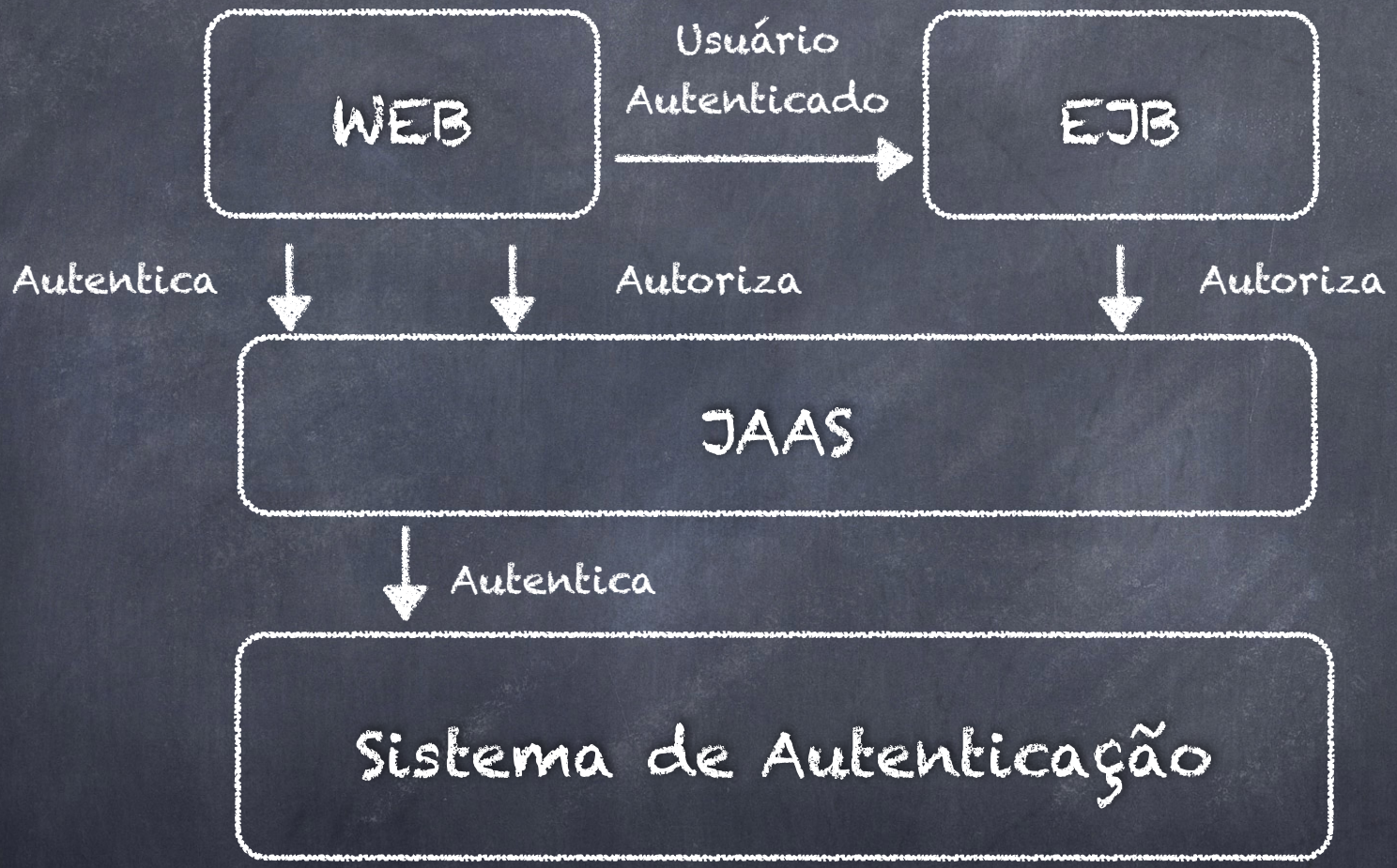
Em segurança, há dois importantes conceitos

* Autenticação

* Autorização

JAAS

Java Authentication and Authorization System


WEB
Usuário
Autenticado
EJB
Autentica
Autoriza
Autoriza
JAAS
Autentica
Sistema de Autenticação

Na spec EJB, dois termos são importantes:

* Principal

* Role

- Por padrão, todos os métodos de um EJB não são protegidos
- A especificação define uma série de anotações que podem ser usadas nas classes ou nos métodos para realizar o controle de acesso
  - @RolesAllowed, @PermitAll, @DenyAll, @RunAs

```java
@RolesAllowed({"user", "admin"})
@Stateless
@LocalBean
public class EmployeeService {

    @PermitAll
    public void save (...) { ... }

    @RolesAllowed("admin")
    public void remove (...) { ... }
```

```
@RolesAllowed({"user", "admin"})
@Stateless
@LocalBean
public class EmployeeService {

    @PermitAll
    public void save (...) { ... }

    @DenyAll
    public void remove (...) { ... }
```

```
@Stateless
@LocalBean
public class EmployeeService {


    public void save (...) { ... }

    @RunAs("admin")
    public void remove (...) { ... }
```

Controlando a segurança programaticamente

```
@Resource

private SessionContext context;

@Asynchronous
public Future<Long> import (List<Employee> list){

    if(!context.isCallerInRole("admin")) {

        for (Employee emp: list){

            ...

        }

        ...

} } }
```

```
@Resource

private SessionContext context;

@Asynchronous
public Future<Long> import (List<Employee> list){

    Principal caller = context.getCallerPrincipal();

    String login = caller.getName();

    ...

    } }
```

JBoss EAP 6.4

* add-user.sh | bat

* jboss-ejb-client.properties

```
remote.connection.default.username=user
remote.connection.default.password=pass
```

Exercícios (9)

- Defina um usuário usando o script add-user.sh (ou bat).
  - Atribua a ROLE developer
  - Proteja um método a sua escolha
    - Faça uma chamada sem usuário
    - Faça uma outra com o usuário definido