

# Le développement Web côté serveur avec Java EE

## Module 5 – Les contextes d'exécution et les cookies



# Objectifs

- Comprendre la notion de contexte de requête
- Comprendre la notion de cookie
- Comprendre la notion de contexte de session
- Comprendre la notion de contexte d'application
- Savoir manipuler ces différents espaces de travail en parallèle

# Le contexte de requête

- Représenté par les classes

- `HttpServletRequest`
- `HttpServletResponse`

- Partage d'informations

- `request.setAttribute(cle, valeur)`
- `request.getAttribute(cle)`

- Point d'entrée pour exploiter

- Les cookies

```
Cookie[] cookies = request.getCookies();
```

- La session

```
HttpSession session = request.getSession();
```

*Durée de vie  
limitée à  
la requête HTTP  
sous-jacente*

Les contextes d'exécution et les cookies

# Le contexte de requête

## Démonstration



# Le cookie

- Couple clé/valeur
- 4 ko maximum
- Permet de stocker de l'information côté client

*Durée de vie  
définie par  
les caractéristiques  
du cookie*



# La classe Cookie

## Cookie

```
Cookie(String name, String value)
```

```
setMaxAge(int expiry)
```

```
setHttpOnly(boolean httpOnly)
```

```
setComment(String purpose)
```

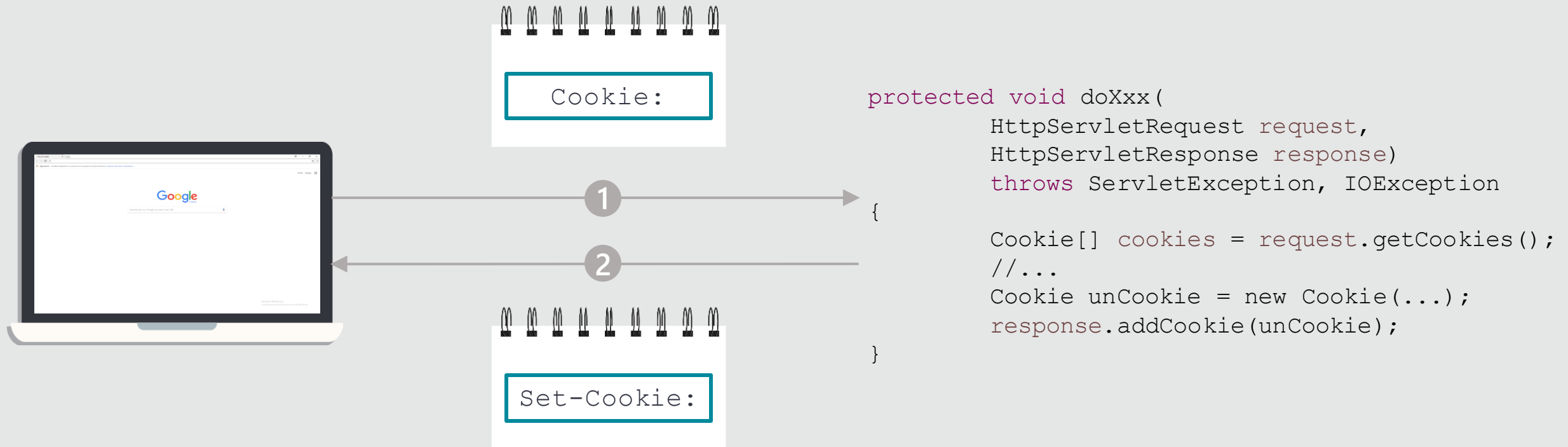
```
setValue(String value)
```

```
getName():String
```

```
getValue():String
```

# Les contextes d'exécution et les cookies

## Le transfert HTTP



Les contextes d'exécution et les cookies

# Le cookie

## Démonstration





Les contextes d'exécution et les cookies

# La session



# L'interface HttpSession

HttpSession  
<<interface>>

```
setAttribute(String name, Object value)
getAttribute():Object

getId():String

removeAttribute(String name)
invalidate()
setMaxInactiveInterval(int interval)
```

```
<session-config>
  <session-timeout>10</session-timeout>
</session-config>
```

# La manipulation d'une session

```
protected void doXxx(  
    HttpServletRequest request,  
    HttpServletResponse response)  
    throws ServletException, IOException  
{  
    HttpSession session = request.getSession();  
    ...  
}
```

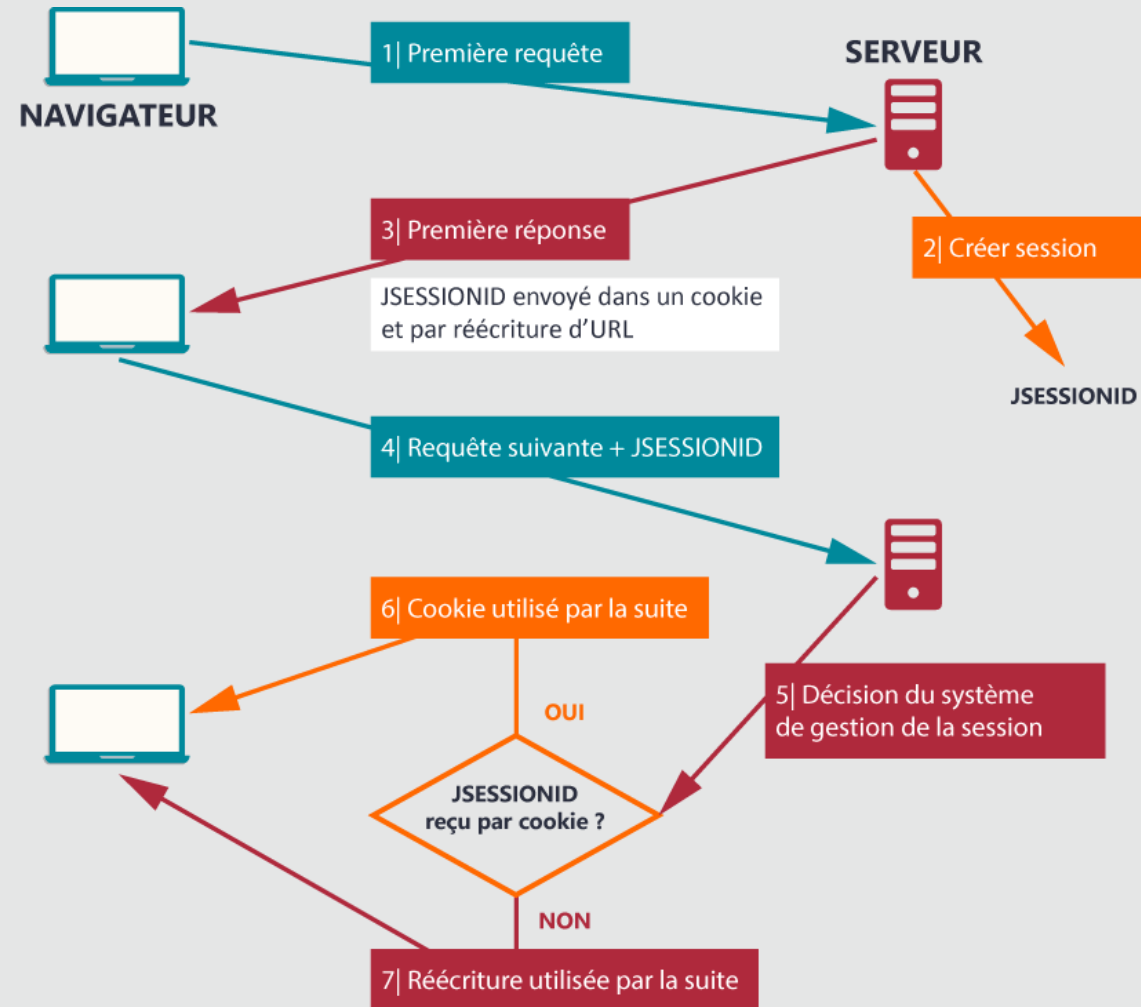
# Les contextes d'exécution et les cookies

## Le suivi de session

JSESSIONID

COOKIE

REECRITURE  
D'URL



Les JSP

# La session

## Démonstration



# Le contexte d'application

- Accessible par tous les composants de l'application
- Représenté par la classe `ServletContext`
- Accessible au travers d'une instance de servlet  
`ServletContext application = this.getServletContext();`
- Accessible au travers de la variable `application` depuis une JSP

# La classe ServletContext

## ServletContext

```
setAttribute(String name, Object value)
getAttribute():Object
removeAttribute(String name)

setInitParameter(String name, String value);
getInitParameter():String

getNamedDispatcher(String name):RequestDispatcher
getRequestDispatcher(String path):RequestDispatcher
```

# Les attributs « globaux »

- Fonctionnement identique aux attributs de requête

- L'écriture d'un attribut « global »

```
application.setAttribute("cle", "valeur de n'importe quel type");
```

- La lecture d'un attribut « global »

```
type maVariable = (type) application.getAttribute("cle");
```

- La durée de vie est limitée à la durée de vie de l'application



# Les paramètres « globaux »

- Dans le web.xml

```
<context-param>  
  <description>Un paramètre d'application</description>  
  <param-name>EMAIL_ADMINISTRATEUR</param-name>  
  <param-value>administrateur@monapplication.fr</param-value>  
</context-param>
```

- Dans une servlet

```
this.getServletContext().getInitParameter("EMAIL_ADMINISTRATEUR");
```

- Dans une JSP

```
<%  
  application.getInitParameter("EMAIL_ADMINISTRATEUR");  
%>
```

Les contextes d'exécution et les cookies

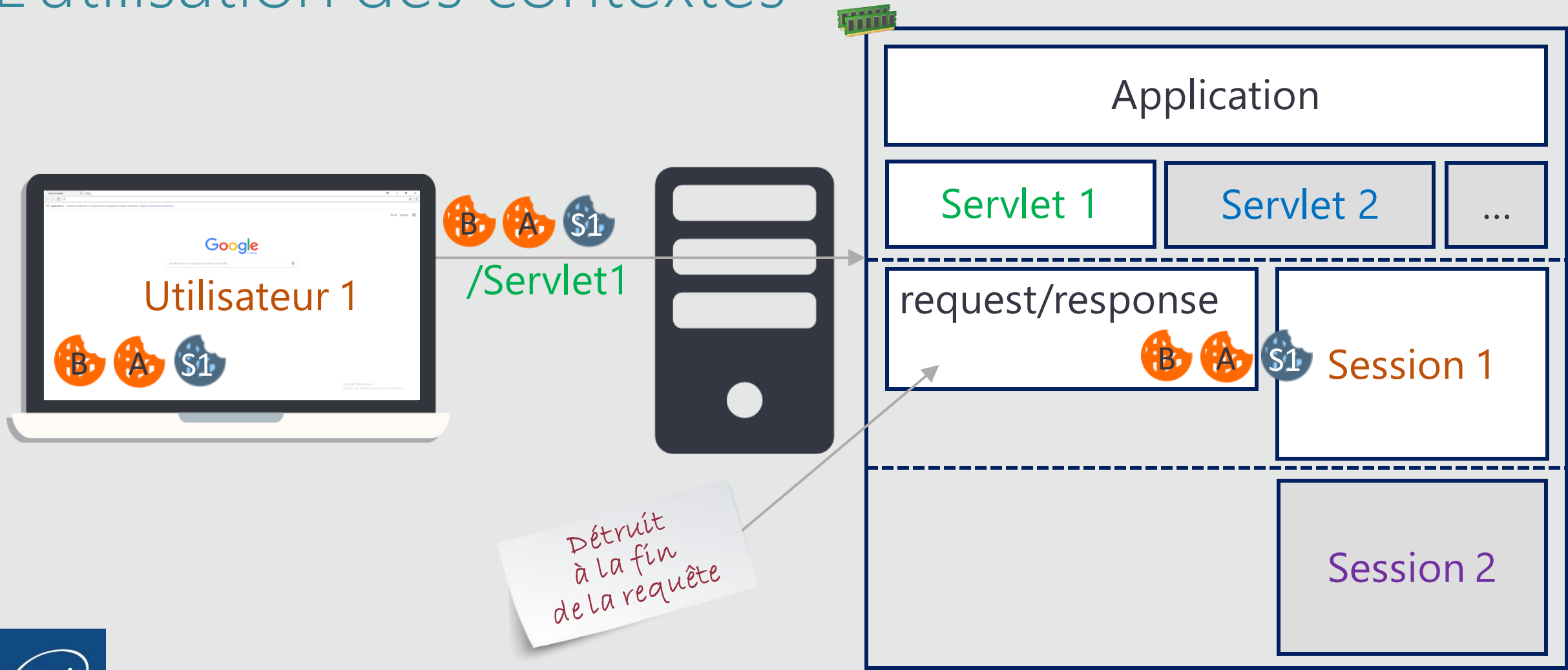
# Le contexte d'application

## Démonstration



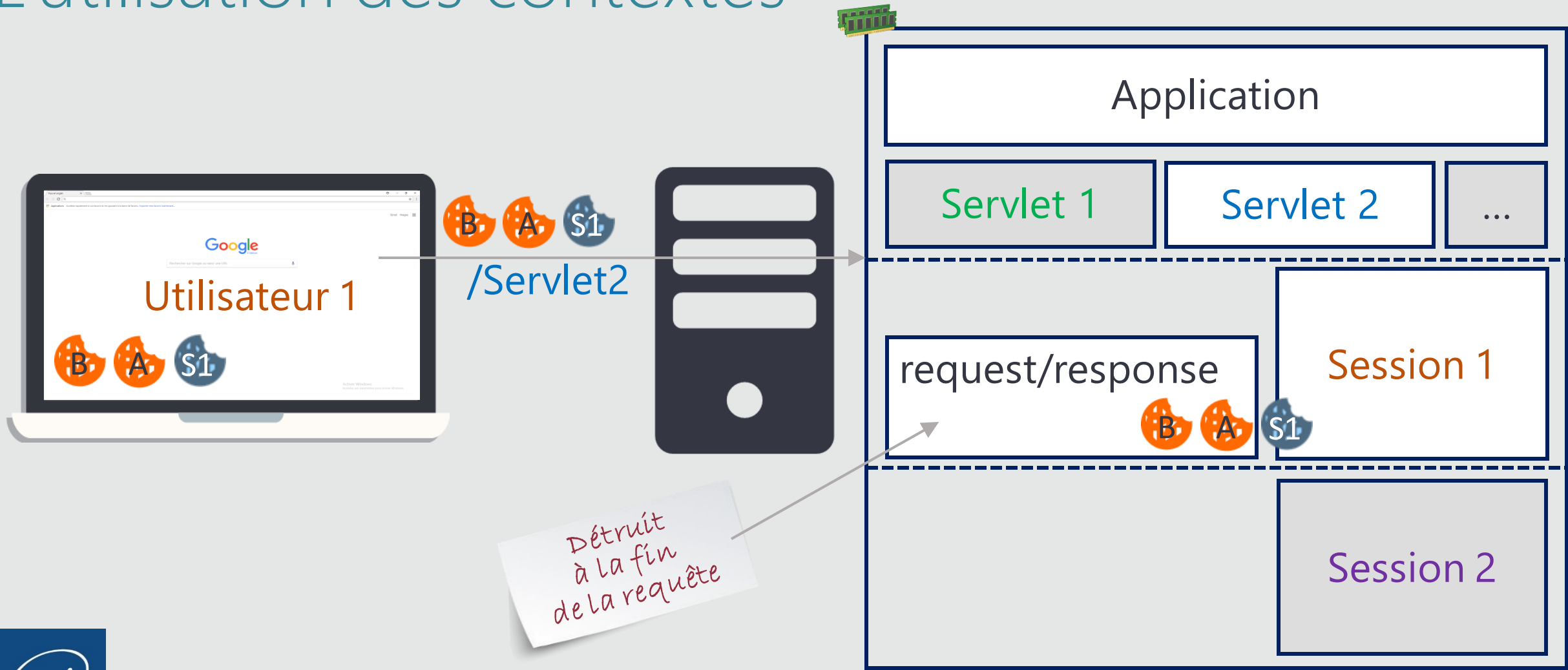
Les contextes d'exécution et les cookies

# L'utilisation des contextes



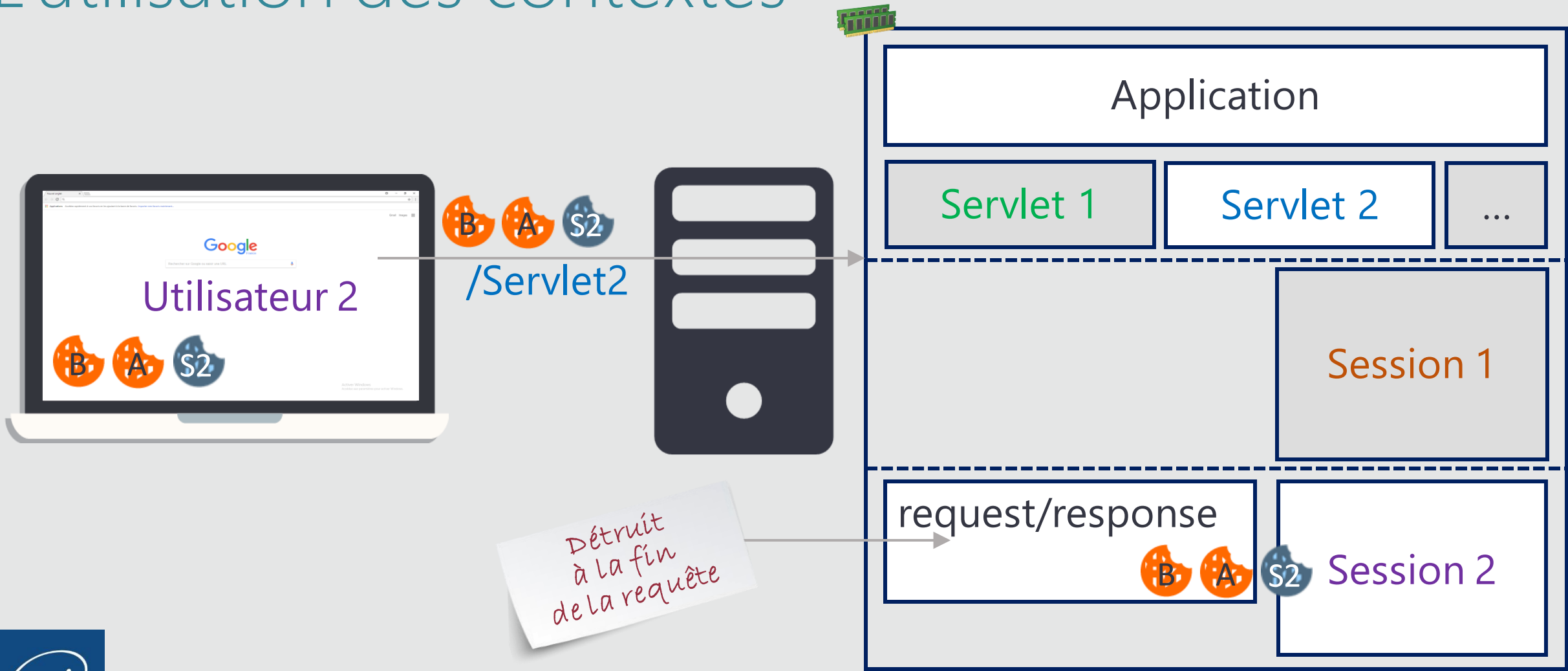
Les contextes d'exécution et les cookies

# L'utilisation des contextes



Les contextes d'exécution et les cookies

# L'utilisation des contextes



Les contextes d'exécution et les cookies

# Préférences d'usage de l'application

TP



# Récapitulatif

## Contexte : application

L'information contenue ici est partagée entre tous les utilisateurs de l'application et est accessible partout

### Usage :

```
ServletContext application =  
    this.getServletContext();
```

```
application.getAttribute("clé");
```

```
application  
    .setAttribute("clé", "valeur");
```

## Contexte : session & Contexte : cookies

L'information contenue ici n'appartient qu'à un unique utilisateur et est accessible partout

### Usage :

```
HttpSession session = request.getSession();  
Cookie[] cookies = request.getCookies();
```

```
session.getAttribute("clé");  
session.setAttribute("clé", "valeur");  
cookies[0].getName(); cookies[0].getValue();  
request.setParameter(myCookie);
```

## Contexte : request

L'information contenue ici n'appartient qu'à un unique utilisateur et n'est que dans les Servlet/jsp concernées

### Usage :

```
request.getParameter("clé");  
request.setAttribute("clé", "valeur");
```

# Conclusion

- Vous savez manipuler les informations des différents contextes

