Java avancé
Les contextes d'exécution

Objectifs

- Revenir sur la notion de requête
- Comprendre la notion de cookie
- Comprendre la notion de session
- Comprendre la notion de contexte d’application endroit où l'on va stocker de l'info
- Savoir manipuler ces différentes notions ensemble

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();
```

Le cookie

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

La classe Cookie

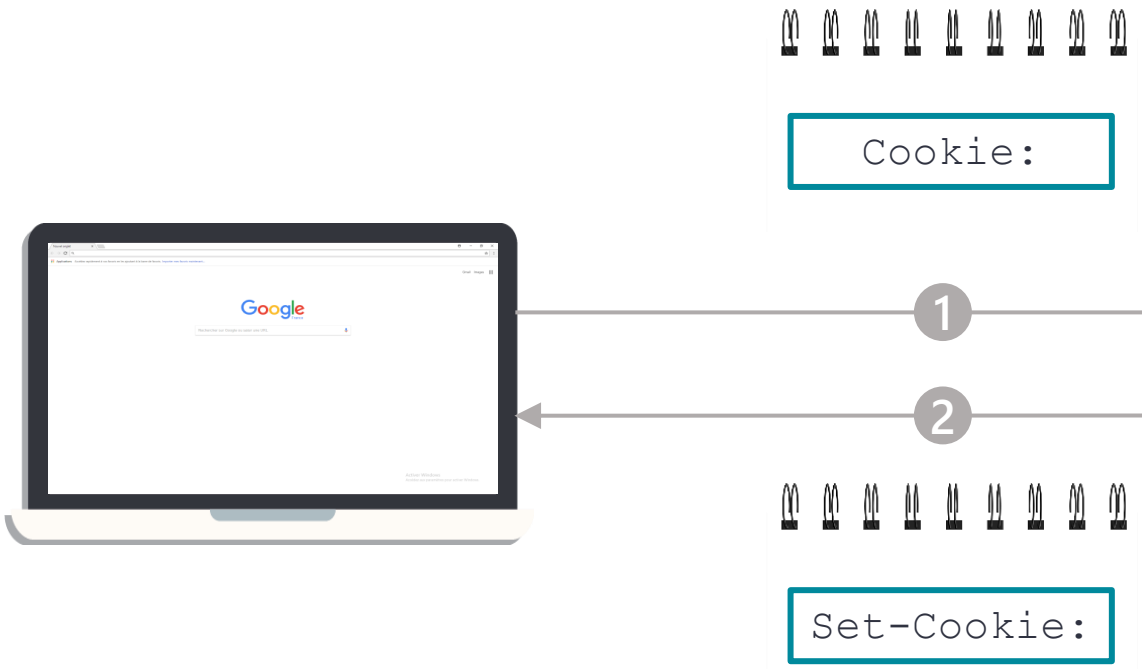
Cookie

`Cookie(String name, String value)`
 	
`setMaxAge(int expiry)` durée d'expiration du cookie (en sec, en gen. on ne met pas)
`setHttpOnly(boolean httpOnly)`
`setComment(String purpose)`
`setValue(String value)` modifier la valeur d'un cookie (important)
 	
`getName():String`
`getValue():String`

cookie: que des String

Le transfert HTTP

ce qui est vulnérable est ce qui est dans les requêtes
(véhiculé sur le réseau)




```
protected void doXxx(
        HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
{
        Cookie[] cookies = request.getCookies();
        //...
        Cookie unCookie = new Cookie(...);
        response.addCookie(unCookie);
}
```

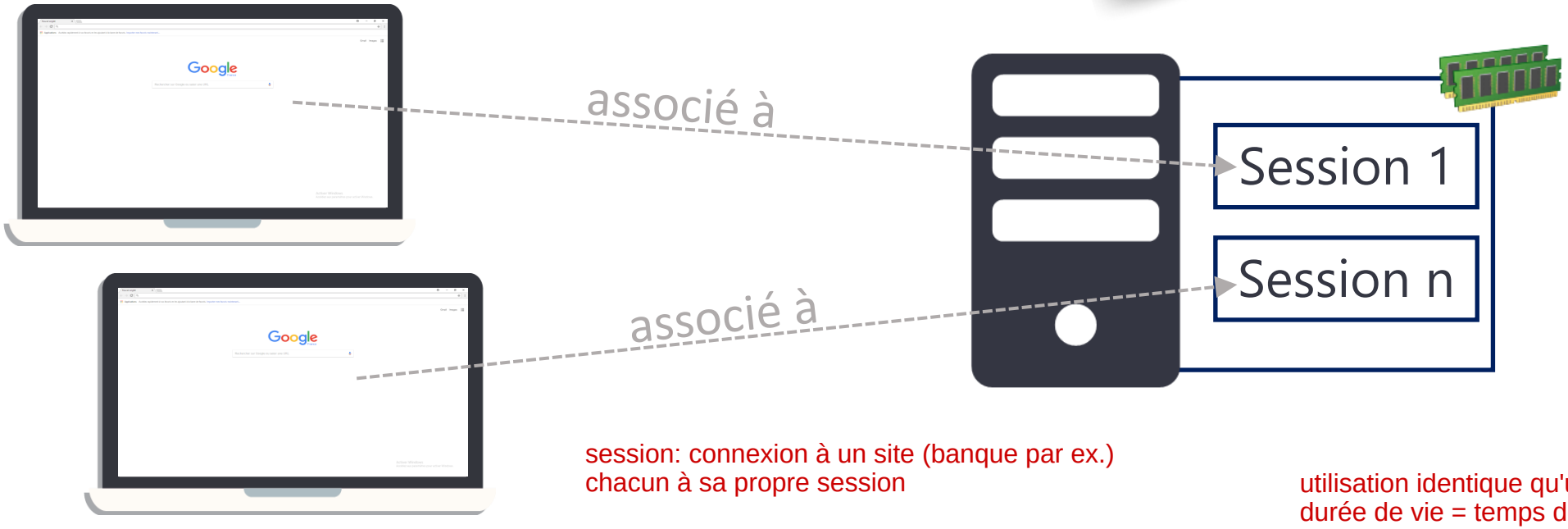
on ajoute un cookie sur l'ordi de l'utilisateur

cookie géré au niveau de la servlet

Demo
Les cookies

La session

Durée de vie limitée
à la durée de l'utilisation
de l'application
par l'utilisateur



session: connexion à un site (banque par ex.)
chacun à sa propre session

utilisation identique qu'un setAttribute
durée de vie = temps de connexion

JsessionId: identifiant d'une session

L’interface HttpSession

- Les contextes d’exécution et les cookies

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>
```

important: ex banque

La manipulation d'une session

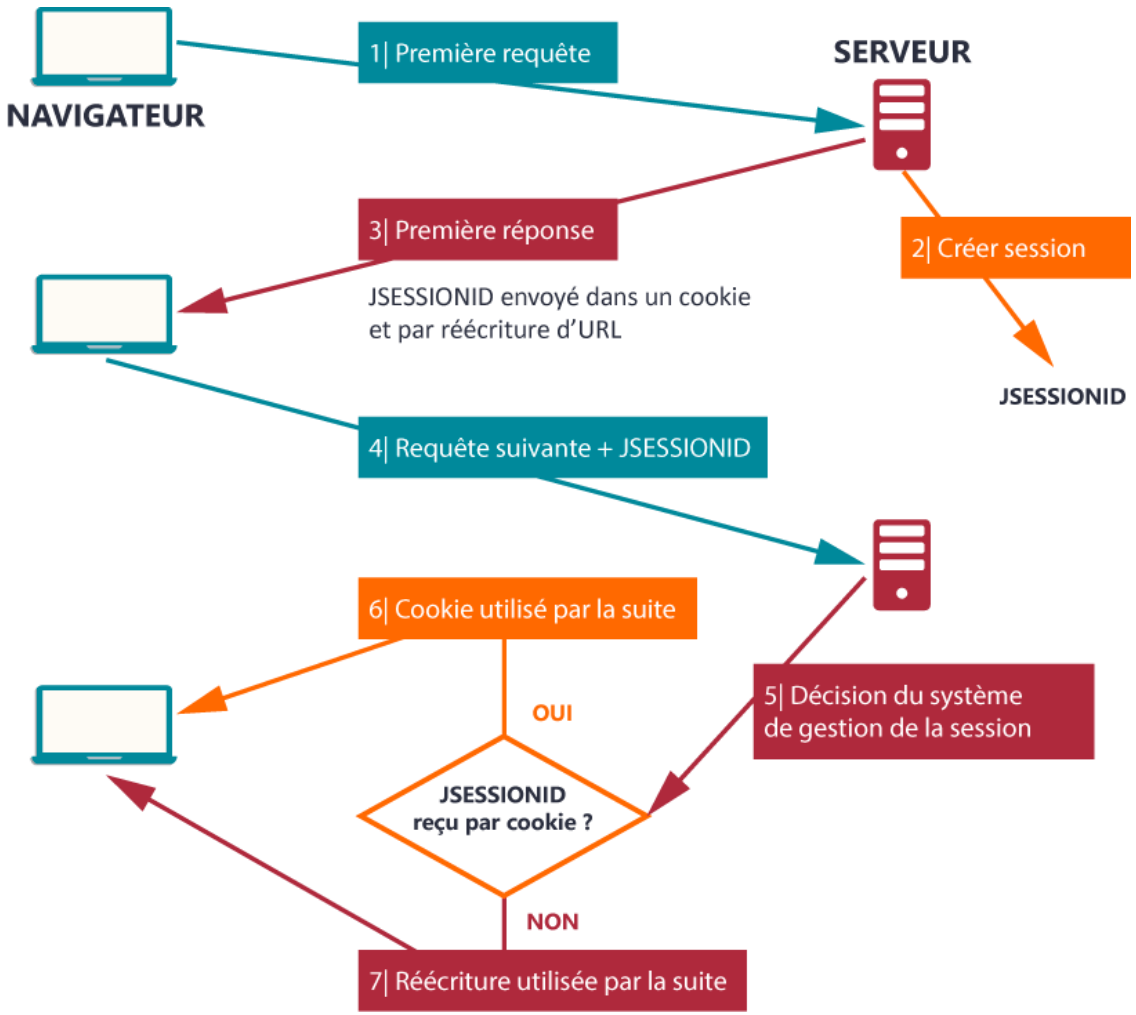
```
protected void doXxx(
        HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
{
        HttpSession session = request.getSession();
        ...
}
```

Le suivi de session

JSESSIONID

COOKIE

REECRITURE D'URL



Demo
La session

une session permet de conserver des infos
dans toutes les pages de notre application

-Exécution d'un programme pour un utilisateur donné

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();
```

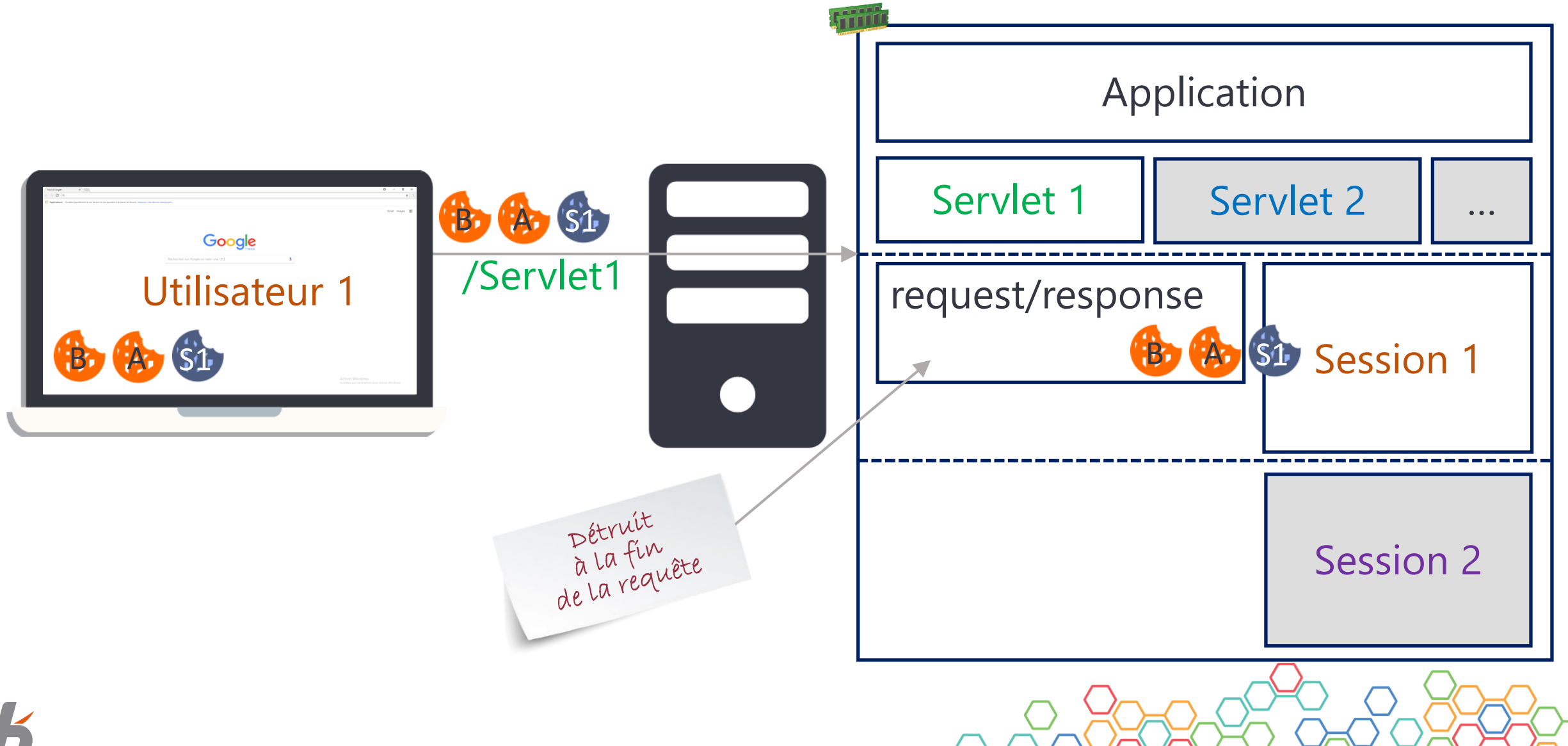
- Accessible au travers de la variable `application` depuis une JSP

La classe ServletContext

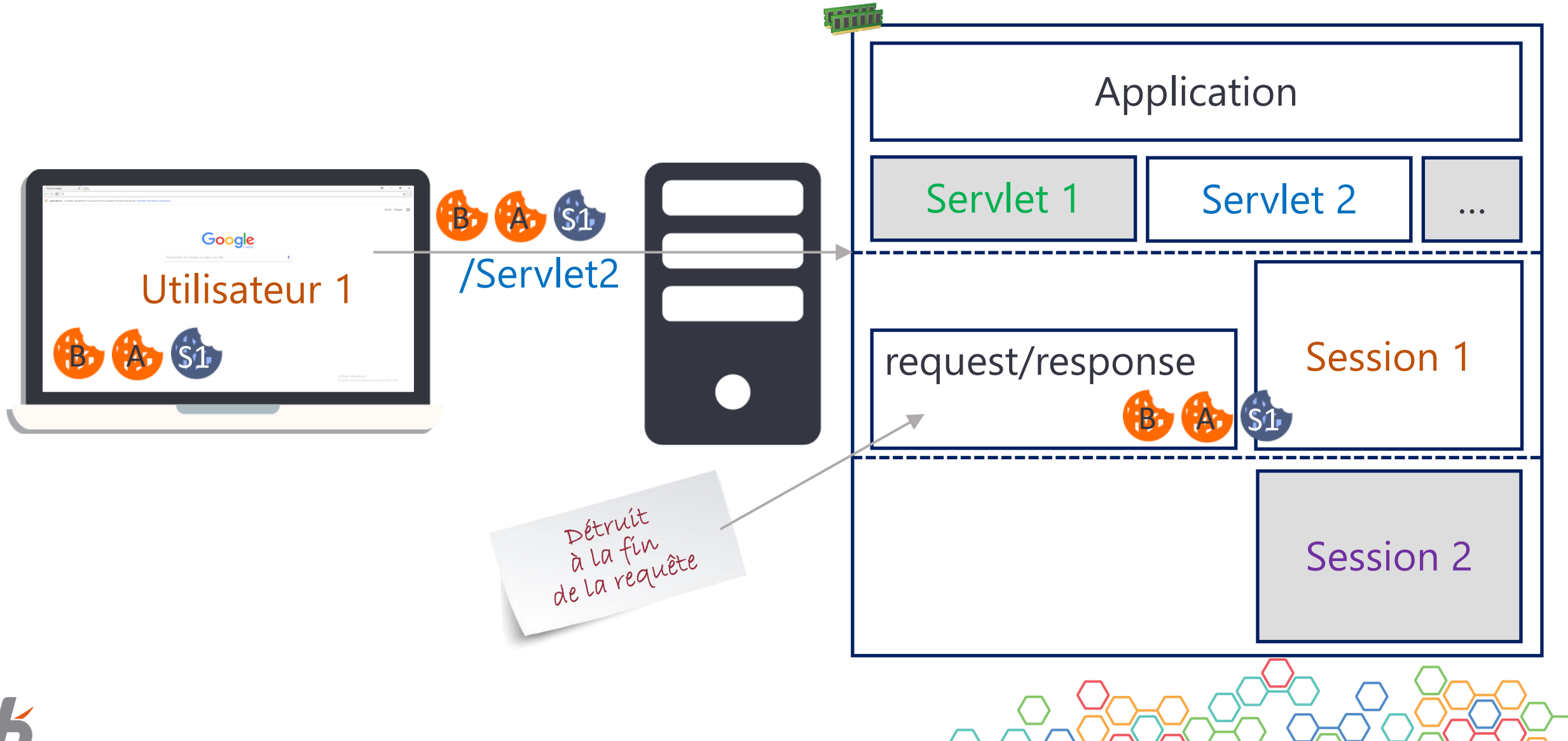
ServletContext
setAttribute(String name, Object value) getAttribute():Object removeAttribute(String name) setInitParameter(String name, String value); getInitParameter():String getNamedDispatcher(String name):RequestDispatcher getRequestDispatcher(String path):RequestDispatcher

Demo
Le contexte d’application

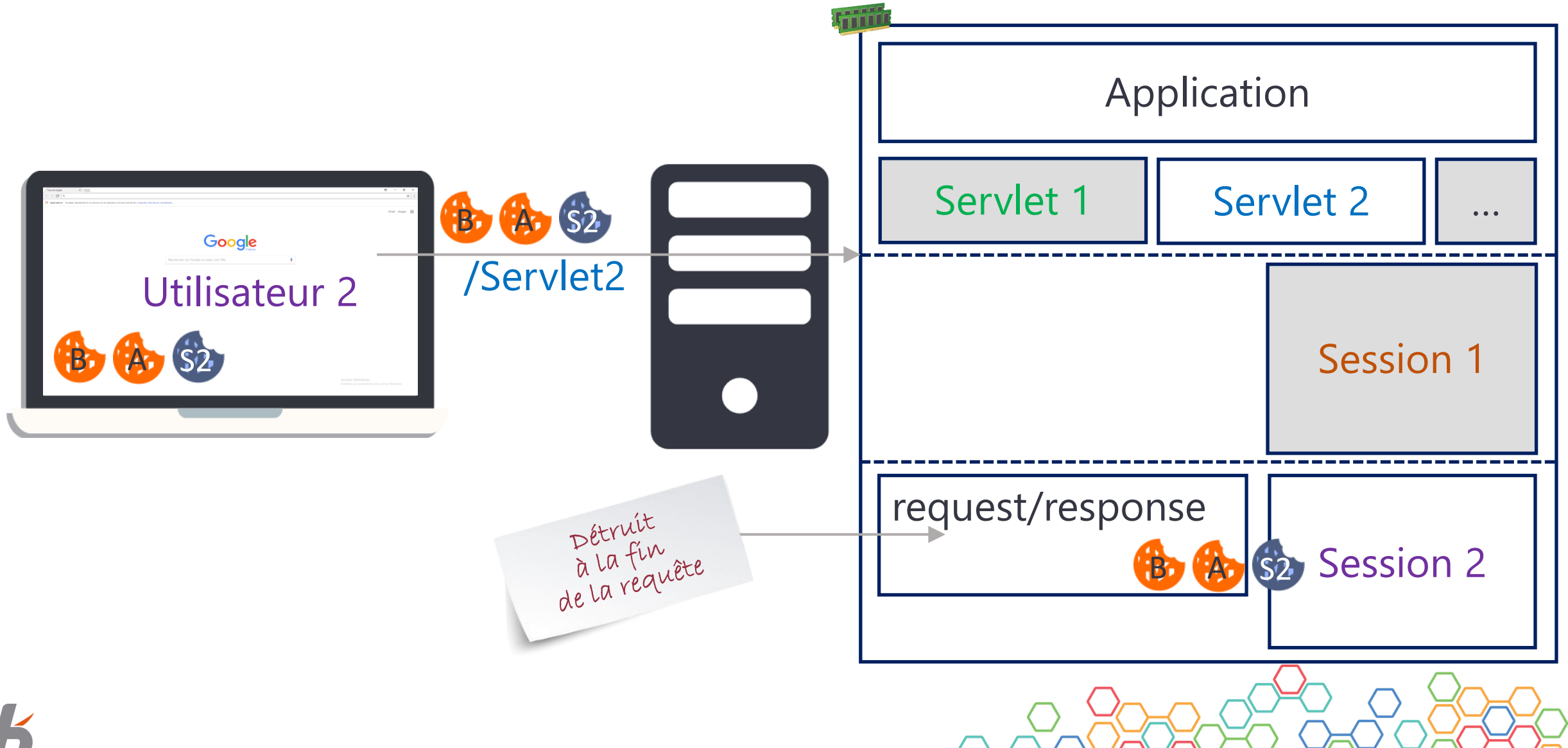
L’utilisation des contextes



L’utilisation des contextes



L’utilisation des contextes



Atelier
Préférences d’usage

Java avancé
Les contextes d'exécution