

Le développement Web côté serveur avec Java EE

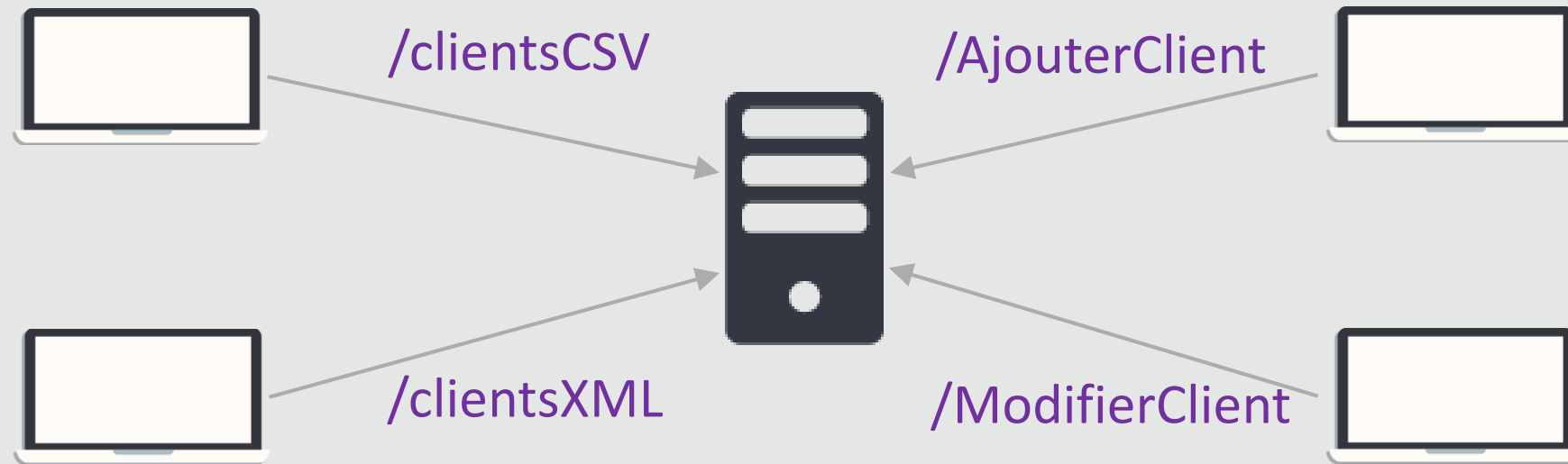
Module 9 – Les services web REST



Objectifs

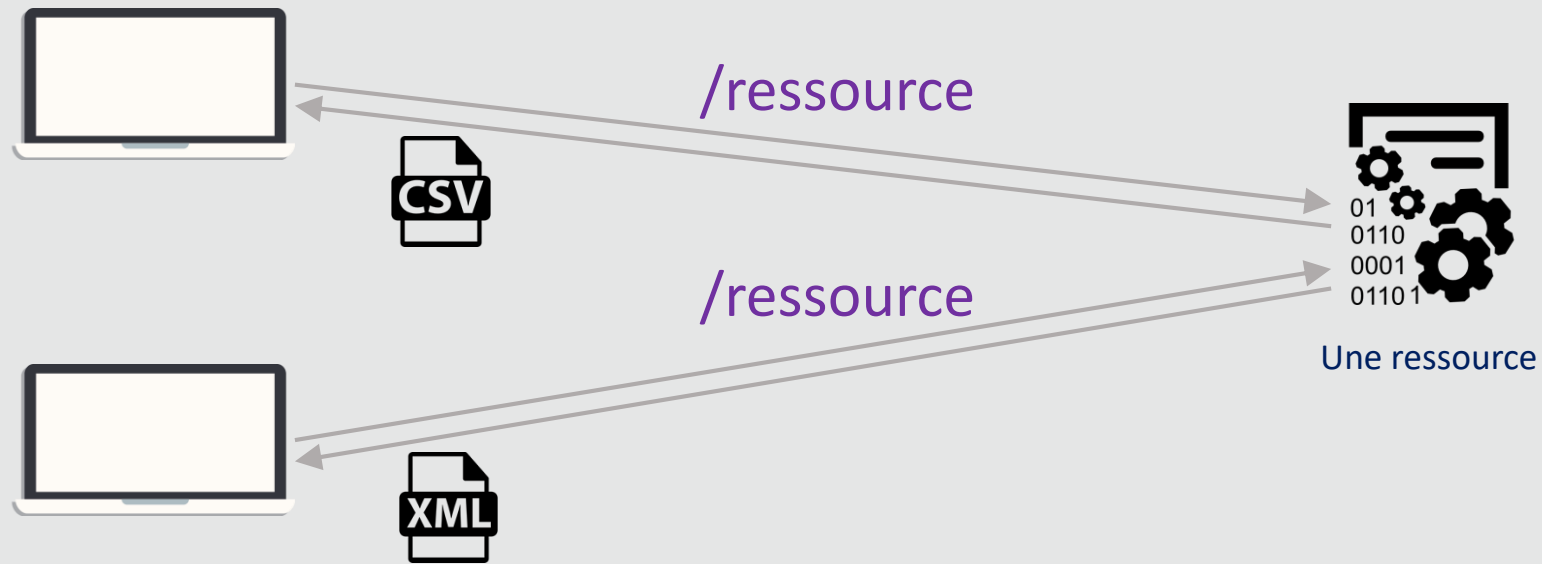
- Comprendre le style d'architecture REST
- Connaître l'essentiel de la spécification JAX-RS
- Savoir mettre en œuvre un service web REST
- Savoir mettre en œuvre un client AJAX/JSON

La problématique



Les services web REST

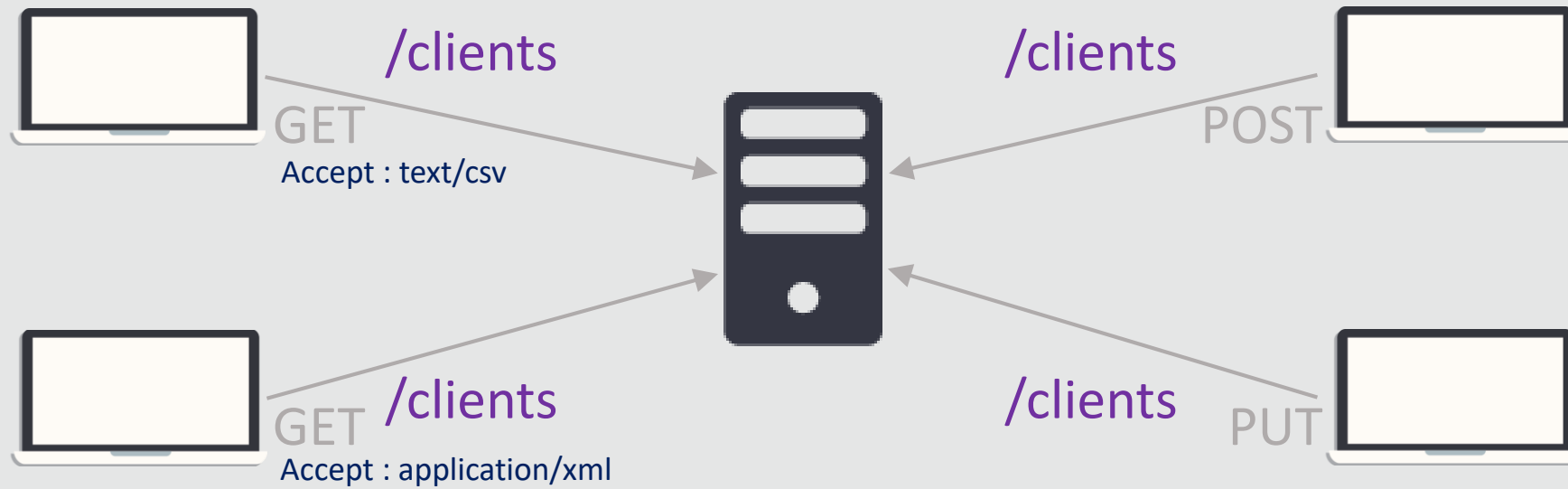
Le concept



REpresentational State Transfer

Les services web REST

La solution



GET=Lecture POST=Ajout PUT=Modification DELETE=Suppression

Les services web REST

L'URL

- Lire tous les clients :
`http://www.exemple.com/clients`
- Lire un client :
`http://www.exemple.com/clients/268`
- Ajouter un client :
`http://www.exemple.com/clients`
- Modifier un client :
`http://www.exemple.com/clients/268`
- Supprimer un client :
`http://www.exemple.com/clients/268`

Les services web REST

La spécification


JAX-RS 2.0

The Java API for RESTful Web Services



JSR 339

Le mécanisme

- Du code Java et des  @

```
@ApplicationPath("/url/racine/des/services/rest")
```

```
@Path("/url/complémentaire/vers/un/service/rest")
```

```
@GET, @POST, @PUT, @DELETE
```


La configuration de l'URL racine

```
import javax.ws.rs.ApplicationPath;  
import javax.ws.rs.core.Application;  
  
@ApplicationPath("/url/racine")  
public class ConfigurationREST extends Application  
{  
  
}
```

Les services web REST

La configuration

Démonstration



La lecture

```
import javax.ws.rs.GET;  
import javax.ws.rs.Path;  
import javax.ws.rs.PathParam;
```

```
@Path("/datas")  
public class GestionDatas  
{
```



/url/racine/datas

```
@GET  
public String getDatas() {...}
```



/url/racine/datas/info

```
@GET @Path("/info")  
public String getInfo() {...}
```



/url/racine/datas/5

```
@GET @Path("/{cle : \\d+}")  
public String getDatasCiblees(@PathParam("cle") int id) {...}
```

```
}
```

Les services web REST

La lecture

Démonstration



Le format de la réponse

- Principaux formats supportés nativement :
 - Les chaînes de caractères
 - Les types primitifs
 - Le XML
- Pour les autres formats :
 - Créer ou déployer un `MessageBodyWriter`

Les services web REST

Une réponse XML

Démonstration



Les services web REST

Un constat

*Inadapté à l'affichage direct
sur un navigateur*

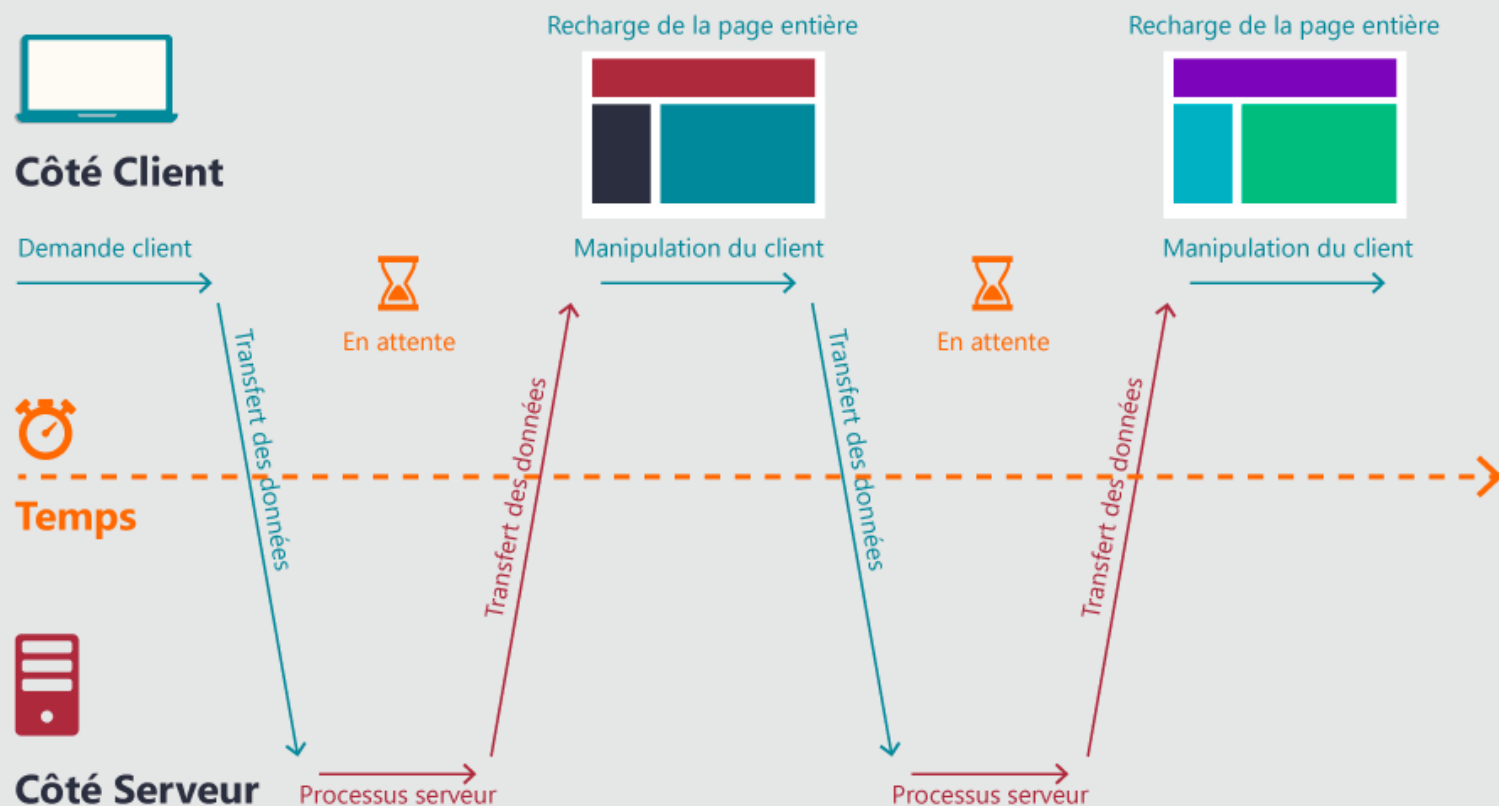
AJAX + JSON

Asynchronous JavaScript Xml

JavaScript

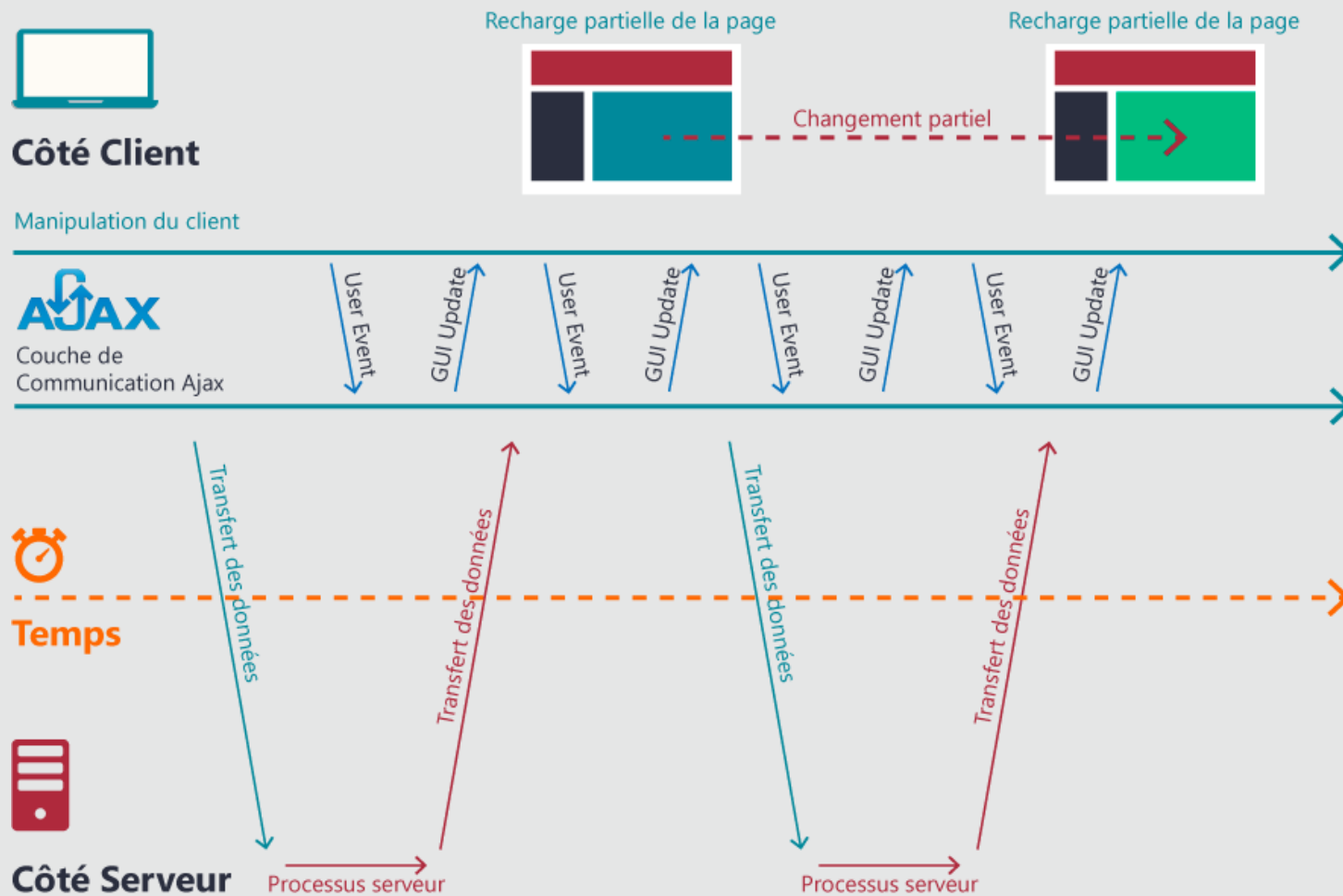


Le concept AJAX



Les services web REST

Le concept AJAX



XMLHttpRequest

- L'objet XMLHttpRequest est un rêve de développeur parce que l'on peut :
 - Mettre à jour une page sans la recharger
 - Requêter le serveur après le chargement de la page
 - Recevoir des données du serveur après le chargement de la page
 - Envoyer des données au serveur en tâche de fond



L'envoi d'une requête

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        // Action typique à réaliser lorsque la réponse est prête :
        document.getElementById("demo").innerHTML = xhttp.responseText;
    }
};
xhttp.open("GET", "url resource", true);
xhttp.send();
```

L'exploitation JSON côté client



- Un contenu JSON

```
[  
  {"id":1,"prop1":"valeurA","prop2":"valeur1"},  
  {"id":40,"prop1":"valeurB","prop2":"valeur2"},  
  {"id":82,"prop1":"valeurC","prop2":"valeur3"}  
]
```

- Un traitement JavaScript

```
var contenuJSON = JSON.parse(datasRecues);  
for(i=0; i< contenuJSON.length;i++)  
{  
    unDiv.innerHTML += contenuJSON[i].prop1 + " - ";  
}
```

Les services web REST

Une réponse JSON

Démonstration



Les services web REST

L'écriture

```
import javax.ws.rs.GET;  
import javax.ws.rs.Path;  
import javax.ws.rs.PathParam;
```

```
@Path("/datas")  
public class GestionDatas  
{
```

```
    @POST
```

```
    /url/racine/datas public String ajouterDatas(@FormParam("champ") String param){...}
```

```
    @PUT @Path("/{cle : \\d+}")
```

```
    /url/racine/datas/5 public String modifierDatas( @PathParam("cle") int id,  
                                                       @FormParam("champ") String param){...}
```

```
    @DELETE @Path("/{cle : \\d+}")
```

```
    /url/racine/datas/5 public String supprimerDatas(@PathParam("cle") int id){...}
```

```
}
```

Les services web REST
L'écriture

Démonstration



Les services web REST

Prise de notes

TP



Conclusion

- Vous maîtrisez les basiques de la spécification JAX-RS pour mettre en œuvre des services web REST