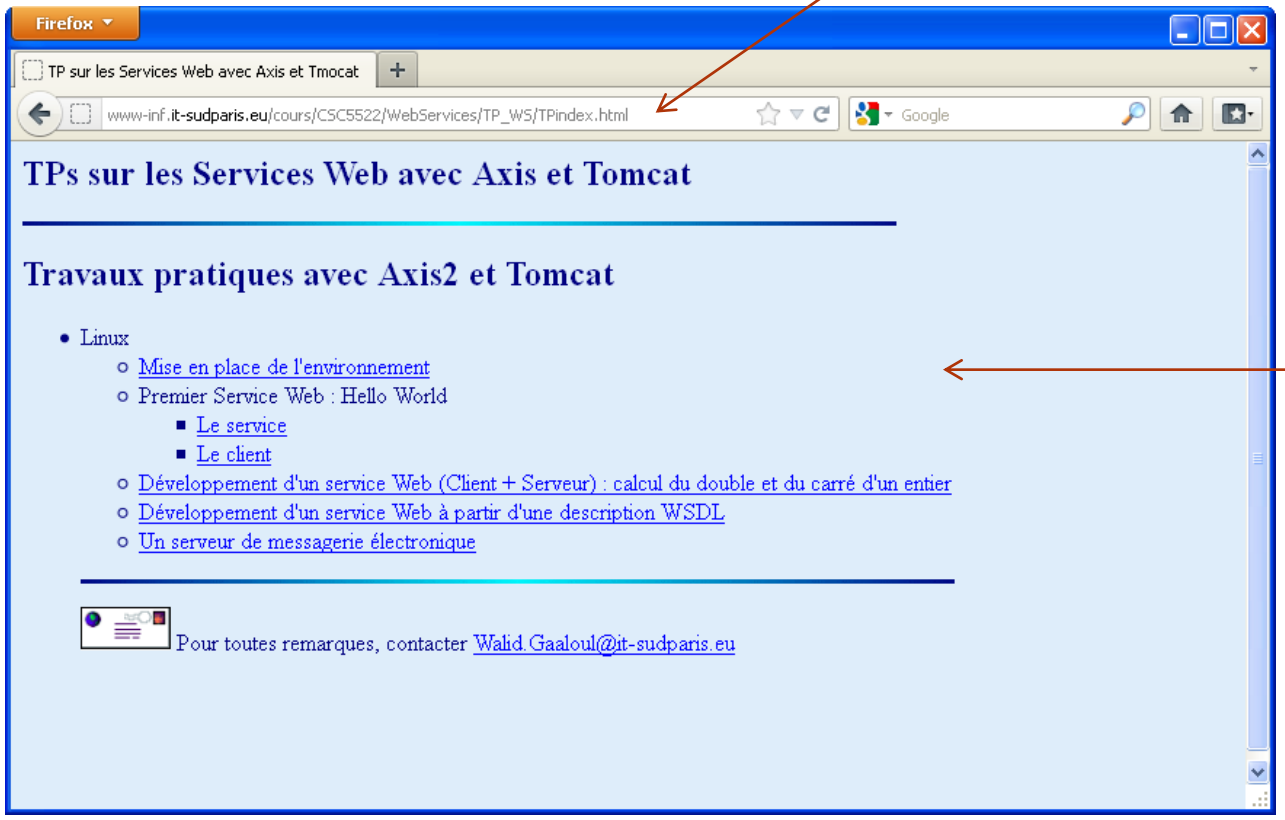
REpresentational State Transfer

Nguyen Ngoc Chan

Principles

Resource Identifier



Resource Representation

URI: http://www-inf.it-sudparis.eu/cours/CSC5522/WebServices/TP_WS/TPindex.html

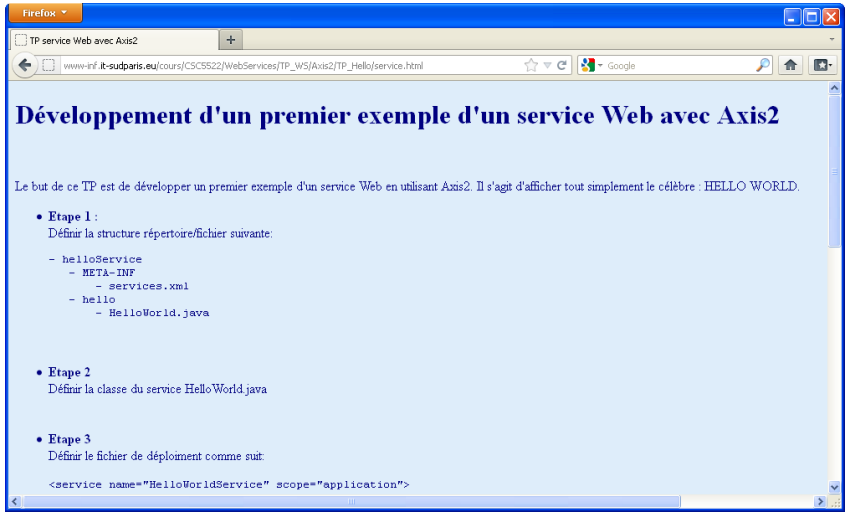
Resource representation: can be in HTML, XML, Figure, Text, English, French, etc.

Principles

http://www-inf.it-sudparis.eu/cours/CSC5522/WebServices/TP_WS/TPindex.html

Link between resources

Click on a hyperlink



http://www-inf.it-sudparis.eu/cours/CSC5522/WebServices/TP_WS/Axis2/TP_Hello/service.html

Principles

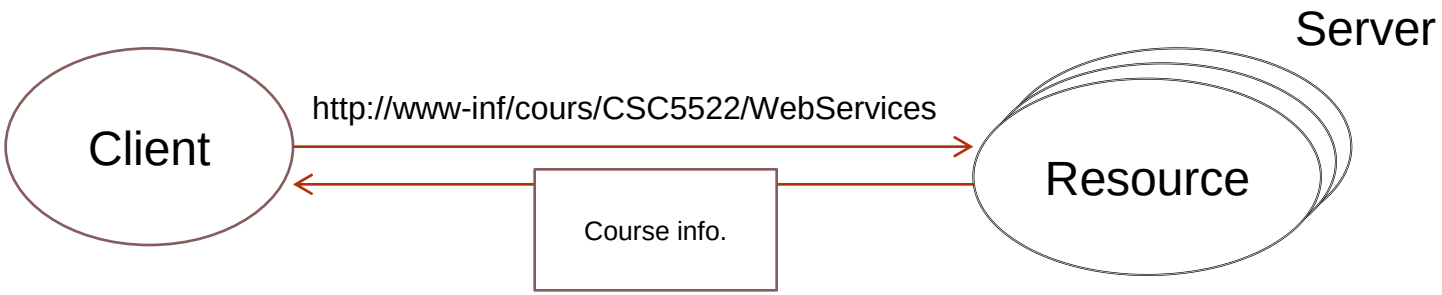
- Static web:
  - Q1: list all the labs of the course WS?
  - A1: create a page: http://www-inf/cours/CSC5522/WebServices/TP_WS/liste_tp.html
  - Q2: list all the courses?
  - A2: create a page: http://www-inf /cours/all.html
  - Q3: Add a new courses?
  - A3: create a new page.

- Disadvantages:
  - Whenever we create/modify/delete a resource, we create/modify/delete an URI and links between resources.
  - Not flexible to represent resources in different document types: HTML, XML, JSON, …

- Solution:
  - Identify resources by logical URLs.
  - Define the resource representations.
  - Design the links between resources.

REST

- Application state is divided into resources.
- Resources are addressable using a universal syntax.
- Resources are linked.
- Standard HTTP methods is used for data transfer.
- Client server architecture and stateless communication between.



- A representation of the resource (with links to other resources) is returned to the client.
- Client is in a new state.
- The client click on the link to another resource, it get another resource representation and move to a new state.
- On the client, the states are transferred.

REST

- Standard HTTP methods:
  - GET: retrieve a representation of a resource
  - POST: create a new resource
  - PUT: create or modify an existing resource
  - DELETE: delete an existing resource

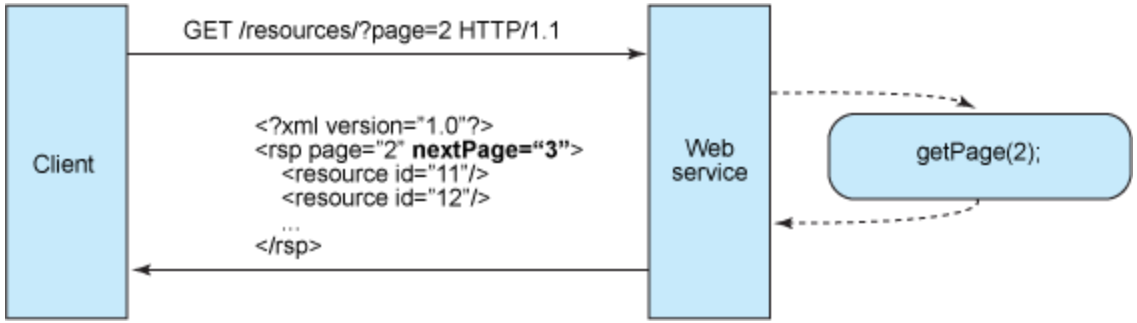
REST

- Universal syntax:
  - http://www-inf/cours/CSC5522/modules/WebServices
  - Syntax: http://www-inf/cours/{course_id}/modules/{name}
  - http://www-inf/cours/CSC5522/lecturers
  - Syntax: http://www-inf/cours/{course_id}/lecturers
  - http://www-inf/cours/inf/20102
  - Syntax: http://www-inf/cours/{departement_id}/{years}
  - http://www-inf/cours/lecturers/Walid
  - Syntax: http://www-inf/cours/lecturers/{name}

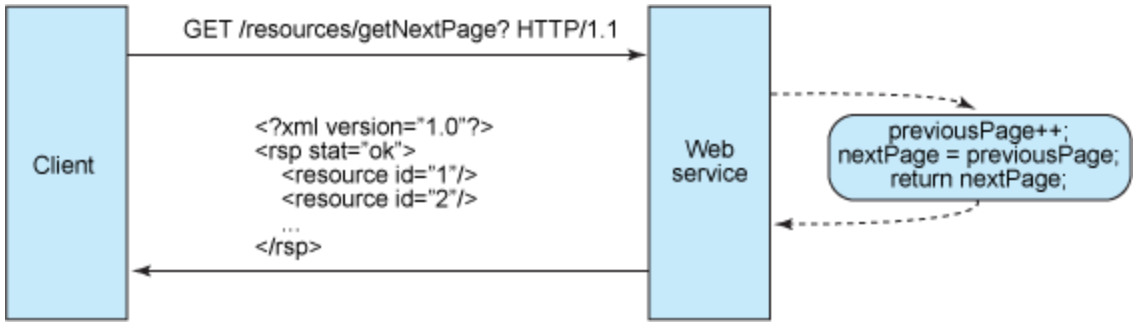
- Other examples:
  - http://theme.wordpress.com/themes/rusty-grunge/
  - http://developers.facebook.com/blog/post/494

REST

- Stateless communication:



stateless

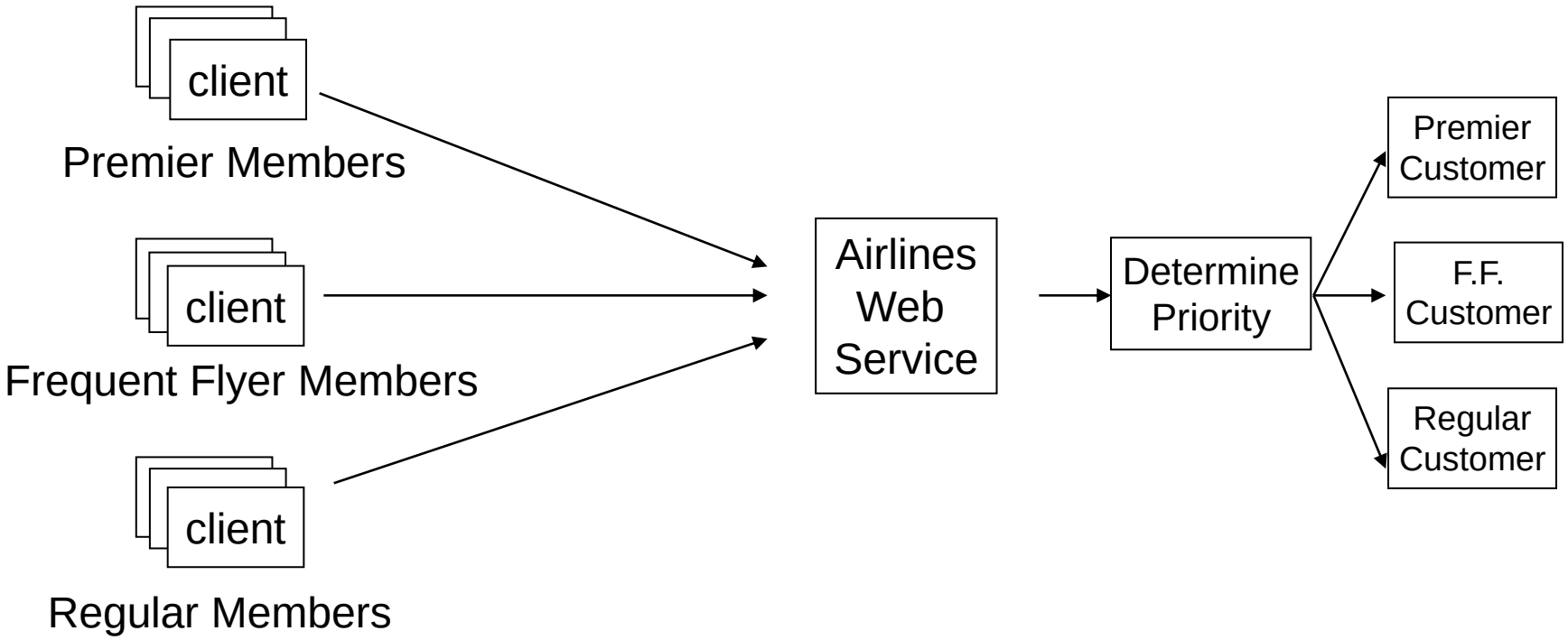


stateful

REST vs SOAP

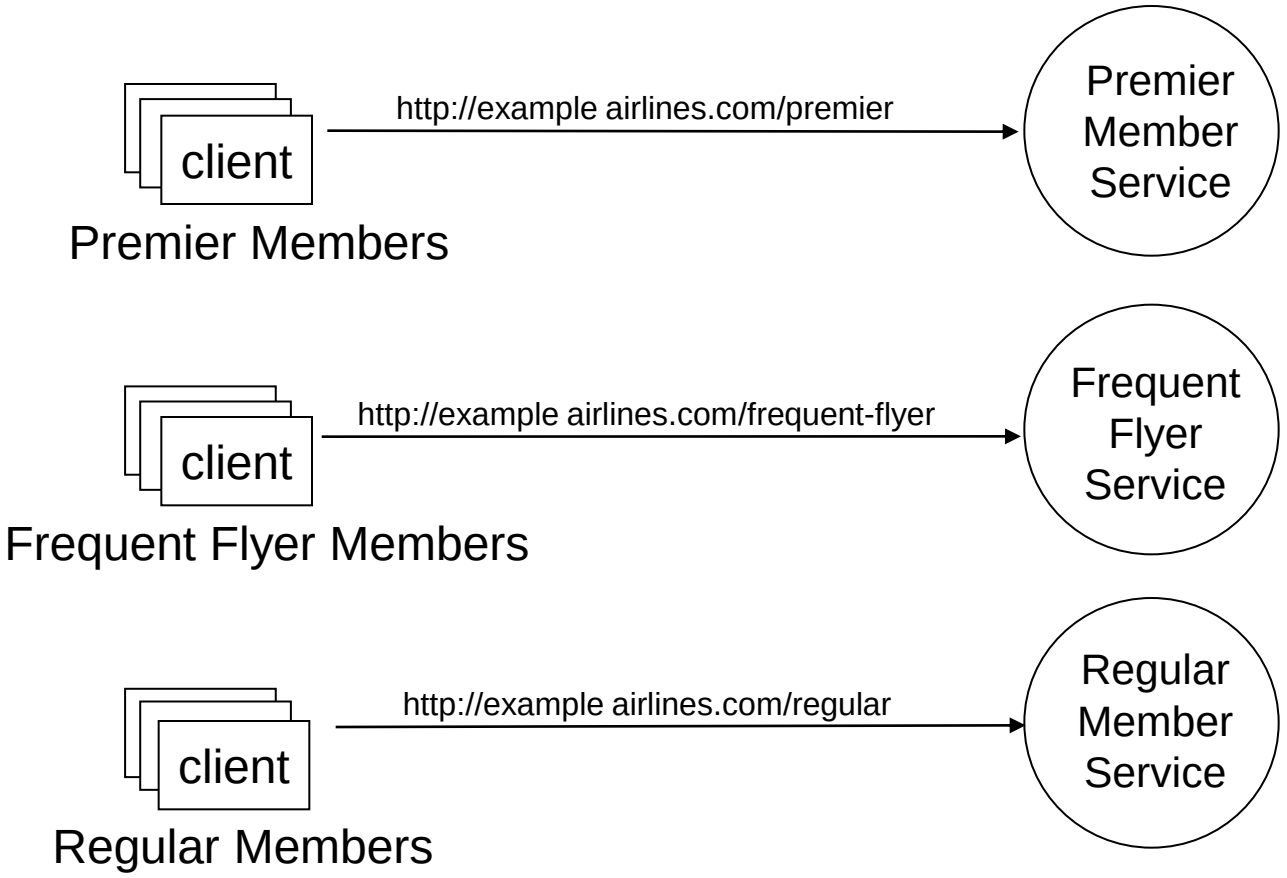
- REST
  - is just a design pattern
  - is not a standard
  - does prescribe the use of standards
  - represents (states) in HTML, XML, Text, JSON, …
- SOAP
  - is a protocol for exchanging structured information
  - is a standard
  - represents (contents) in XML

REST vs WS-*


client
Premier Members
client
Frequent Flyer Members
client
Regular Members
Airlines Web Service
Determine Priority
Premier Customer
F.F. Customer
Regular Customer

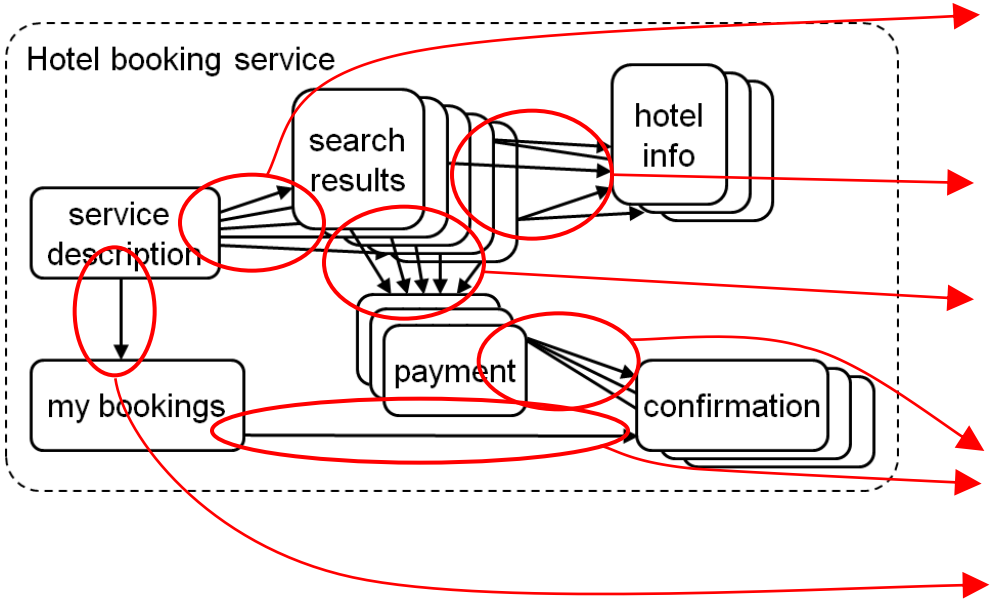
WS-*

REST vs WS-*



REST

REST vs WS-*



search(date, city)

→ list of hotels & rates

getHotelDetails(hotel)

→ hotel details

reserve(rate, creditCard)

→ confirmationID

getConfirmationDetails(confID)

→ confirmation details

listMyBookings()

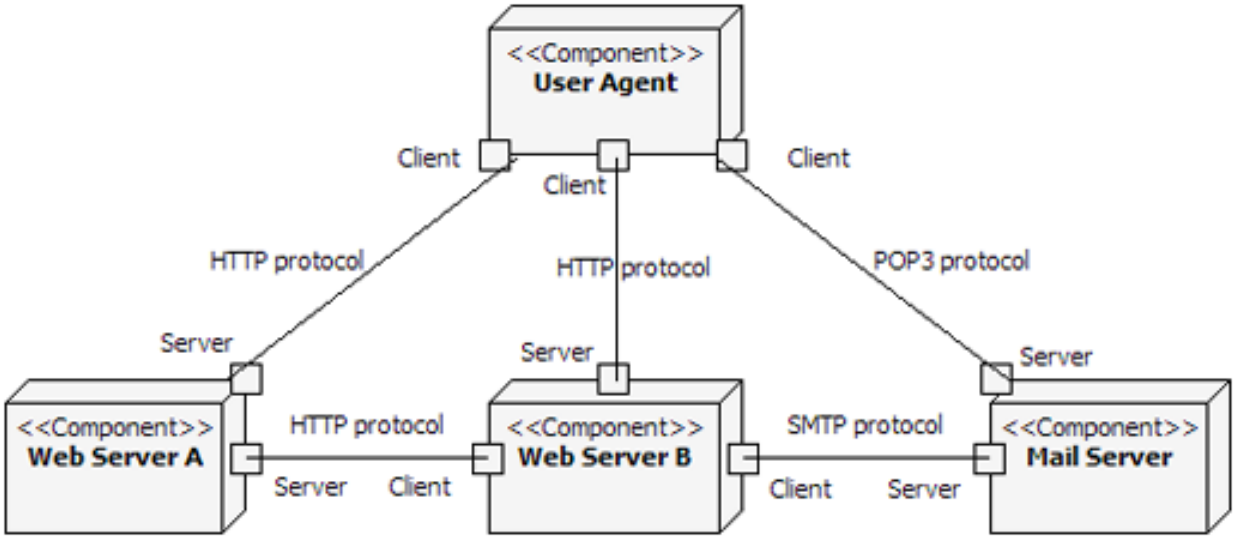
→ list of confirmationIDs

nouns vs. verbs

(REST) (WS-*)

Restlet framework

- http://www.restlet.org/



References

- http://en.wikipedia.org/wiki/Representational_state_transfer
- http://www.ibm.com/developerworks/webservices/library/ws-restful
- http://www.ibm.com/developerworks/library/x-restfulsoa/
- http://www.infoq.com/articles/rest-introduction
- http://www.restlet.org/