

Telecommunication Services Engineering (TSE) Lab



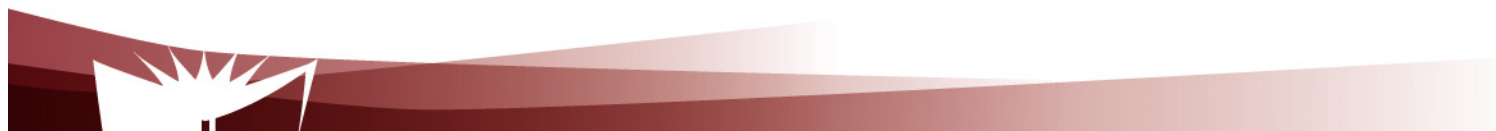
Chapter II – Addendum (Stepwise procedure for REST modelling)

Case Study – REST for Conferencing

<http://users.encs.concordia.ca/~glitho/>

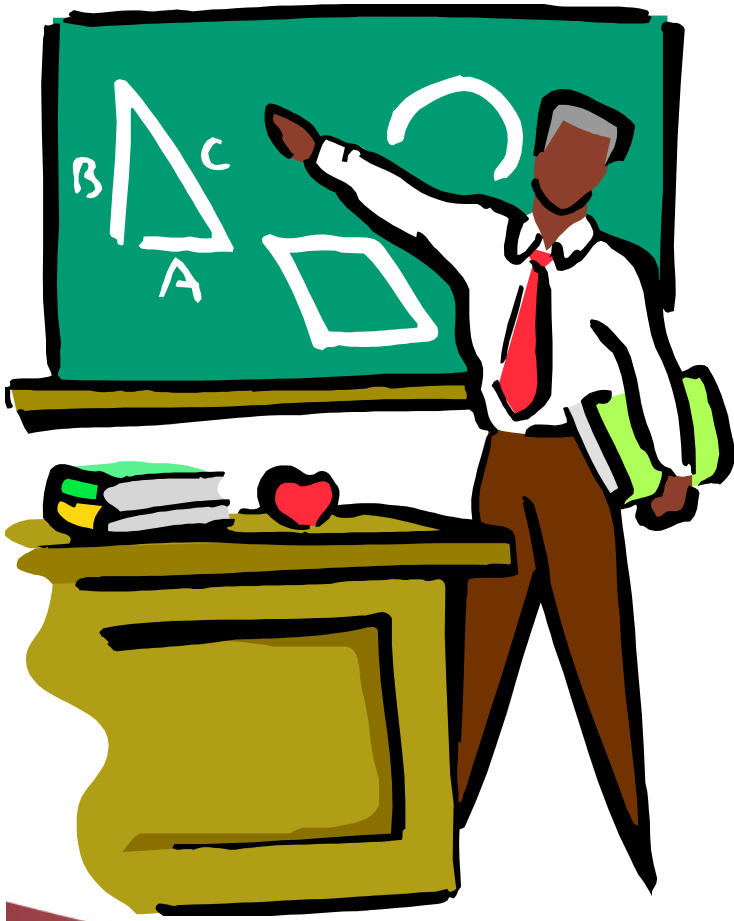
References

- F. Belqasmi, C. Fu, R. Glitho, Services Provisioning in Next Generation Networks: A Survey, *IEEE Communications Magazine*, December 2011
- F. Belqasmi, J. Singh, S. Bani Melhem, and R. Glitho, SOAP Based Web Services vs. RESTful Web Services: A Case Study for Multimedia Conferencing Applications, *IEEE Internet Computing*, July/August 2012



Case Study On Conferencing

1. A stepwise procedure
2. On conferencing semantics
3. Applying the procedure to conferencing



The procedure – First Part

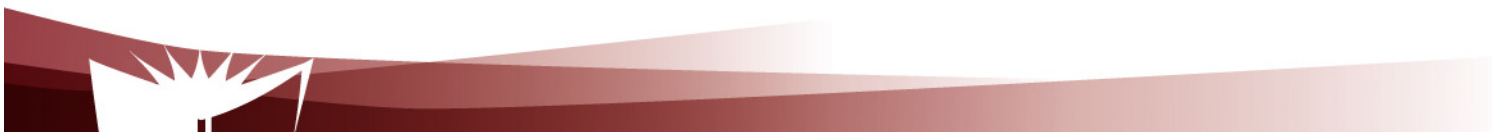
- Figure out the data set
- Split the data set into resources



The procedure – Second Part

For each resource:

- Name the resources with URIs
- Identify the subset of the uniform interface that is exposed by the resource
- Design the representation(s) as received (in a request) from and sent (in a reply) to the client
- Consider the typical course of events by exploring and defining how the new service behaves and what happens during a successful execution



On Conferencing semantics

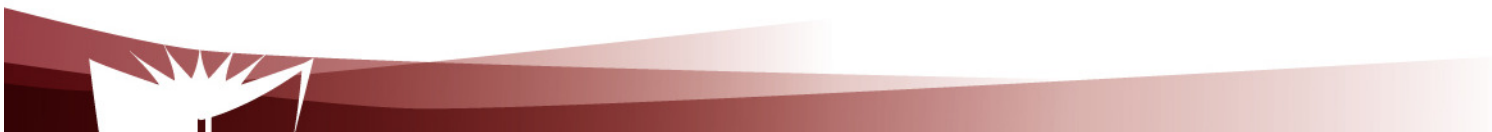
- The conversational exchange of multimedia content between several parties
 - About multimedia
 - Audio, video, data, messaging
 - About participants
 - Any one who wants to participate the conference



On Conferencing semantics

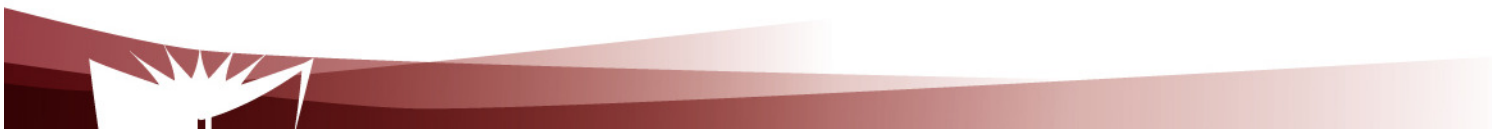
Classification:

- Dial-in / dial-out
- Open/close
- Pre-arranged/ad hoc
- With/without sub-conferencing (i.e. sidebar)
- With/without floor control



On conferencing semantics

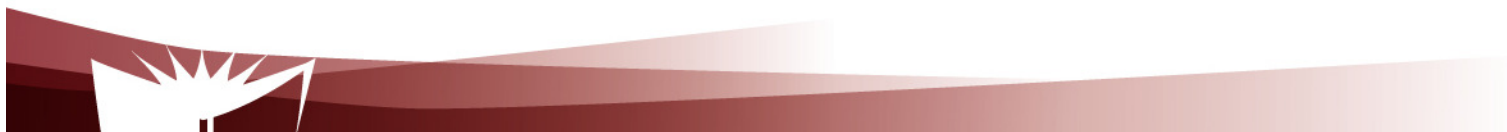
- Case considered in the use case
 - Create a service that allows a conference manager to :
 - Create a conference
 - Terminate a conference
 - Get a conference status
 - Add users to a conference
 - Remove users from a conference
 - Change media for a participant
 - Get a participant media



Applying the procedure – First part

1. Data set

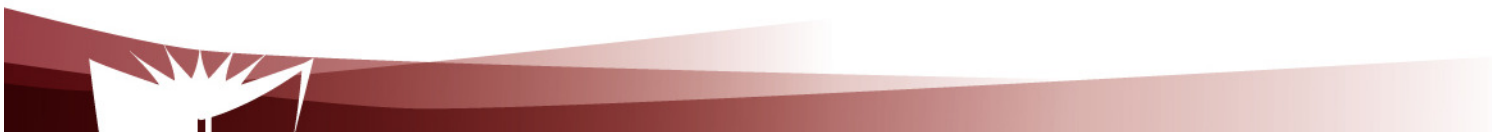
- Conferences
- Participants
- Media



Applying the procedure – First part

2. Split the data set into resources

- Each conference is a resource
- Each participant is a resource
- One special resource that lists the participants
- One special resource that lists the conferences (if we consider simultaneous conferences)



Applying the procedure – Second part

3. Name the resources with URIs

- I'll root the web service at
<http://www.confexample.com/>
- I will put the list of conferences at the root URI
- Each conference is defined by its ID:
<http://www.confexample.com/{confId}/>
- A conference participants' resources are subordinates of the conference resource:
 - The lists of participants:
<http://www.confexample.com/{confId}/participants/>
 - Each participant is identified by his/her URI:
<http://www.confexample.com/{confId}/participants/{participantURI}/>



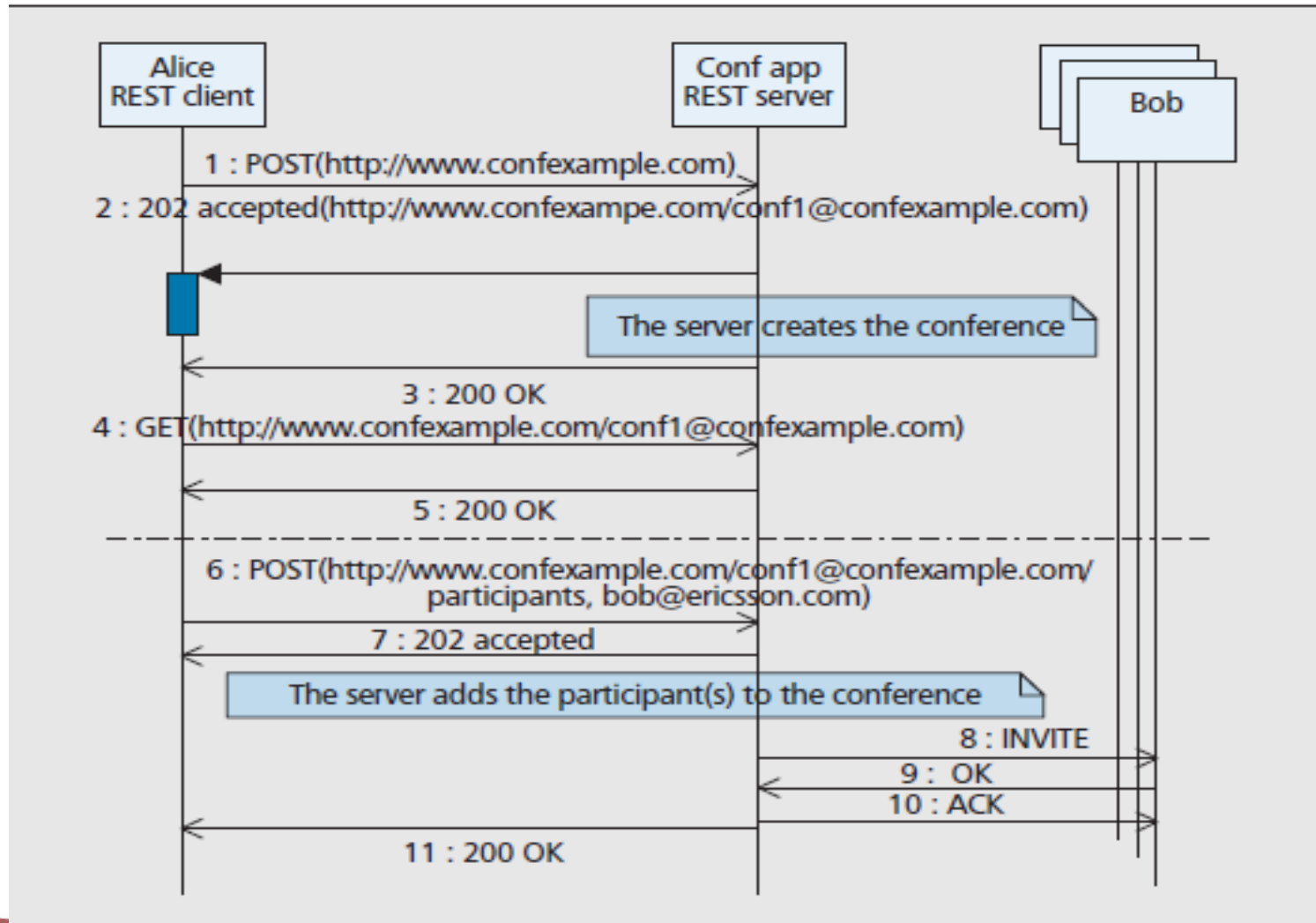
Telecommunication Services Engineering (TSE) Lab

Applying the procedure – Second part

Resource	Exposed subset of the uniform interface		Data representation operation	
	Operation	HTTP action	Client->server	Server->client
Conference	Create: establish a conference	POST: http://confexample.com/	<conference> <description> discuss project </description> <maxParticipants>10</maxParticipants> </conference>	http://www.confexample.com/conf23@exam ple.com
	Read: Get conference status	GET: http://confexample.com/{confid}	None	<status>Active</status>
	Delete: end a conference	DELETE: http://confexample.com/{confid}	None	None
List of participant(s)	Read: Get list of participants	GET: http://confexample.com/{confid}/ participants	None	<participants> <participant> <uri>alice@ericsson.com</uri> <status>Connected</status> </participant> </participants>
	Create: Add a participant	POST: http://confexample.com/{confid}/ participants	<participant> alice@ericsson.com </participant>	<participant> <uri>alice@ericsson.com</uri> <link>http://confexample.com/{confid}/ participants/alice@ericsson.com</link> </participant>
	Read: Get a participant status	GET: http://confexample.com/{confid}/ participants/{participantURI}	None	<status>Invited</status>
	Delete: remove a participant	DELETE: http://confexample.com/{confid}/ participants/{participantURI}	None	None

conference?

Applying the procedure – Second part



Applying the procedure – Second part

9. What might go wrong?

- Conference

Operation	Server->Client	Way it may go wrong
Create (POST)	Success: 200 OK Failure: 400 Bad Request	The received request is not correct (e.g. has a wrong body)
Read (GET)	Success: 200 OK Failure: 404 Not Found	The targeted conference does not exist
Delete (DELETE)	Success: 200 OK Failure: 404 Not Found	The targeted conference does not exist



Applying the procedure – Second part

9. What might go wrong?

- Participant(s)

Operation	Server->Client	Way it may go wrong
Create (POST)	Success: 200 OK Failure: 400 Bad Request Failure: 404 Not Found	<ul style="list-style-type: none">• The received request is not correct (e.g. has a wrong body)• The target conference does not exist
Read (GET)	Success: 200 OK Failure: 404 Not Found	<ul style="list-style-type: none">• The target conference does not exist• The target participant does not exist
Update (PUT)	Success: 200 OK Failure: 400 Bad Request Failure: 404 Not Found	<ul style="list-style-type: none">• The received request is not correct• The target conference does not exist• The target participant does not exist
Delete (DELETE)	Success: 200 OK Failure: 404 Not Found	<ul style="list-style-type: none">• The target conference does not exist• The target participant does not exist

Telecommunication Services Engineering (TSE) Lab

