Exercise Sheet 8

Exercise 1 (Web Services)

1. What is a web service?
   (*Explain in just a few sentences.*)
2. What is XML-RPC?
3. Which three roles contains the theoretical implementation of SOAP web services?
4. Which markup language is used by SOAP web services for interaction?
5. What is WSDL and for what purpose is it used?
6. What is UDDI and for what purpose is it used?
7. Explain the difference between UDDI and WS-Inspection.
8. Describe the difference between the theoretical implementation of SOAP web services and the way, SOAP web services operate in practice.
9. Which protocol is used by RESTful web services for interaction?
10. Which four methods are sufficient to initiate all necessary functions on objects inside RESTful web services?
    (*Name the methods and explain the purpose of each method.*)
11. Which two methods are often used addition to the methods of subtask 10?
    (*Name the methods and explain the purpose of each method.*)

Exercise 2 (RESTful Web Services)

1. Do some basic interaction with the Python library boto [1] and an cloud infrastructure service of your choice.
2. Check your requests and the replies of the infrastructure service via Wireshark [2] to understand the way of interaction.

[1] `https://github.com/boto/boto`
[2] `http://www.wireshark.org/`

This simple python script helps you with your first steps. It fetches the list of buckets of a user from Amazon S3.

```
#!/usr/bin/env python
from boto.ec2.connection import *
from boto.s3.connection import *

try:
  calling_format=boto.s3.connection.OrdinaryCallingFormat()
  connection = boto.s3.connection.S3Connection(
                           aws_access_key_id="<ACCESS_KEY>",
                           aws_secret_access_key="<SECRET_ACCESS_KEY>",
                           is_secure=False,
                           validate_certs=False,
                           host="s3.amazonaws.com",
                           calling_format=calling_format,
                           path="/")
except S3ResponseError:
  print "Error!"
else:
  print "Connection established."

requestbuckets = connection.get_all_buckets()
print (requestbuckets)

for entry in requestbuckets:
  print entry.name
  print entry.creation_date
  print entry.get_acl
  # If the bucket is located anywhere other than inside region us-east-1,
  # the location is printed next to it's name. Otherwise, it will be blank
      .
  print entry.get_location()
```

Exercise 3 (Private Cloud Storage Service)

Install a Private Cloud storage service, which implements the S3 API. You can use e.g. one of these solutions:

- OpenStack Swift [3]
- Eucalyptus Walrus [4]
- Nimbus Cumulus [5]
- S3 ninja [6]

You can also use any other solution, which is free software. You can install the storage service natively or inside a virtual machine.

[3] http://www.openstack.org
[4] http://www.eucalyptus.com
[5] http://www.nimbusproject.org
[6] http://www.s3ninja.net

Exercise 4 (Web Service Interaction)

Develop an application with the boto library, which interacts with the Private Cloud storage service, you have already installed. The application should...

- fetch and print out the list of your buckets
- create a bucket
- upload a local file into the new bucket
- fetch and print out the list of objects in the new bucket
- download the previously uploaded object from the bucket
- erase the previously uploaded object in the new bucket
- erase the new bucket

To simplify the task, you can hard code the file name and bucket name.