Task 1.1 (Restful Web Services) (9 Points)

You want to start an online shop for hipster clothing. Since the lecture covered RESTful Web services, you want to create one for your application. Your shopping service has the resources *products* and *clients*. A *product* has a *name*, a *description* and a *brand*. A *client* has a *first name*, a *last name*, a *street*, a *city*, a *post code* and a *country*.

a) Please describe the *REST* API specification of the RESTful Web service providing the two resources, using appropriate HTTP methods. You want to retrieve products, and create, update and remove users.

b) For each method list the necessary parameters.

c) Additionally, specify in total at least four different meaningful HTTP error- and success codes and returned objects where applicable.

a)	b)	c)	
Ressource Products	**Parameter**	**HTTP Status Codes**	**Rückgabe**
http://example.store.com/v1/products GET: List the URIs and names of all products		200	

http://example.store.com/v1/products/p
GET: List the name description and brand of p
		200	{
			"p":{
			"name": "Bartöl",
			"description": "Macht Bärte samtig und sinnlich…",
			"brand": "HipYaLive"
			}
			}

Ressource Client			
http://example.store.com/v1/clients POST: Create new Client	String firstname String lastname String street int number String city int post code String country	201	
http://example.store.com/v1/clients/c DELETE: Remove Client c from clients		401	
PUT: Update Client c with a new client	String firstname String lastname String street int number String city int post code String country		

Task 1.2 (Microservices) (9 Points)

Your online shop has become a tremendous, world-wide success and your customer database is growing beyond expectations. To cope with this high amount of users, you need to reorganize your service by splitting your customer base into four different bases: Africa, the Americas, Asia-Pacific and Europe.

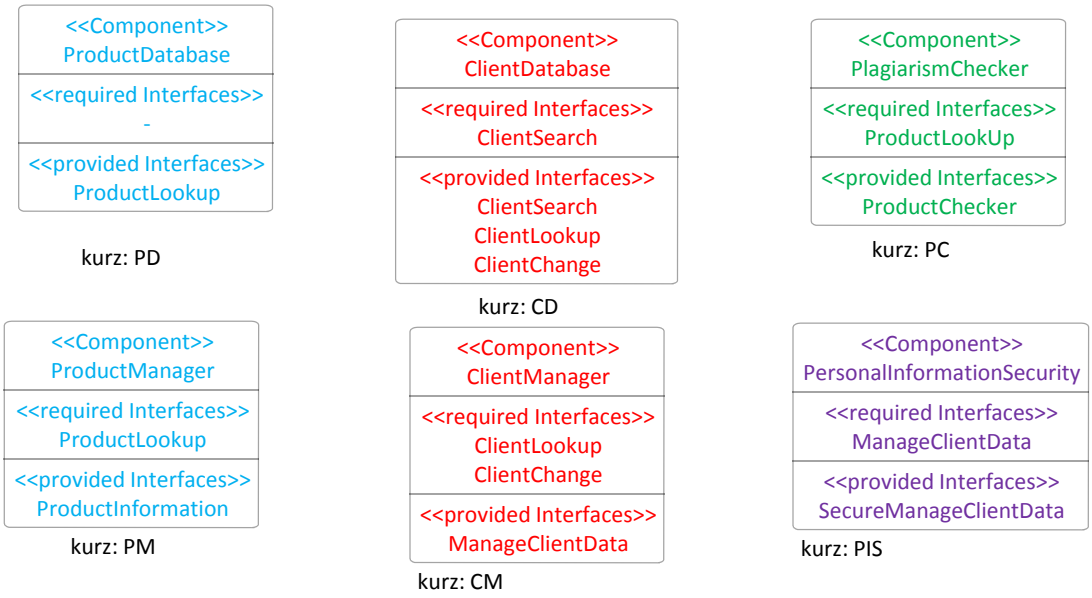
a) Specify, what kind of load balancer you would use to deal with this new setting.

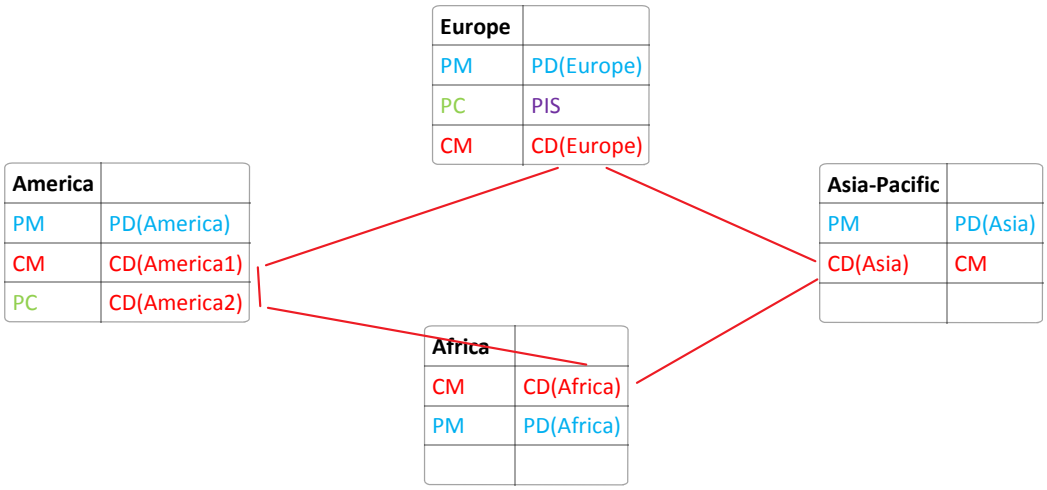
b) Please redesign your RESTful Web service from the first task as a microservice architecture, taking into account the additional regional information. Use a UML component diagram. What are the challenges of such a distributed architecture?

a) We would use a Source IP Hash load balancer to ensure clients are connected to the same server for example when the connection broke retaining their session (for example cart).

b) UML component diagramm:
The challange is to keep all distributed parts of the architecture consistent and complete. For example if a new product is add to the supply it has to be add to the databases of all areas. Moreover Clients will not allways access the services of the area they originally registered for (therefore in our design the distributed ClientDatabases may communicate with each other

through the ClientSearch Interface).





Task 1.3 (Single Sign-On) (7 Points)

To stay up-to-date with recent authorization and authentication technology, your online shop service should now support a single sign-on solution.

a) Please explain the difference between authorization and authentication. Give examples from your hipster clothing service for both concepts.

b) What happens on the HTTP level when using OAuth 2.0?

c) Please specify possible attack scenarios on single sign-on solutions and describe how two-factor authentication can help preventing them.

a) By authenticating a person verifies who he is. Needs a personal secret (for example password, ID-Card,...).
By authorization someone verifies that he is allowed to gain access to a ressource. Requires permission management by the one who offers the ressource.

For example if a Client logs in with a username and the corresponding password he authenticated himself regarding the respective profile.
If there is an adminpassword to delete profiles it is only authorization, because the system doesn't know who you are, but that you are allowed to delete profiles.
Of course the authorization can be done by authenticate yourselfe, for example if certain methods are available to all members of a group, you can authorizate your call of the method by authenticate as a member of the group.

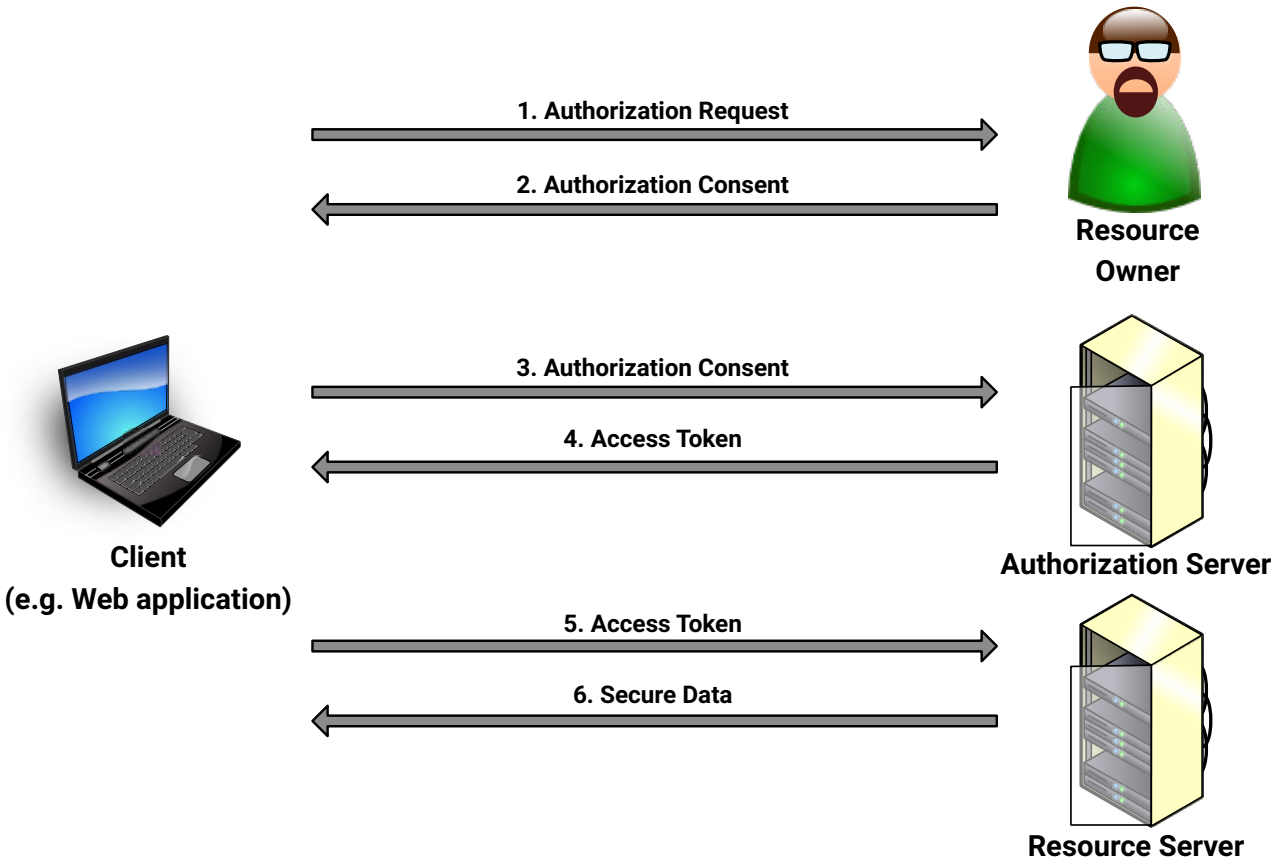
Task 1.3 b)

Like in the lecture visualized on the http level happen the following steps:

1. The application requests authorization to access service resources from the user
2. If the user authorized the request, the application receives an authorization grant
3. The application requests an access token from the authorization server (API) by presenting authentication of its own identity, and the authorization grant
4. If the application identity is authenticated and the authorization grant is valid, the authorization server (API) issues an access token to the application. Authorization is complete.
5. The application requests the resource from the resource server (API) and presents the access token for authentication
6. If the access token is valid, the resource server (API) serves the resource to the application

(https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2)

OAuth 2.0 Authorization Flow



Lehrstuhl Informatik 5 (Information Systems)
Prof. Dr. M. Jarke

Task 1.3 c)

If an attacker has acces to the network he can eavesdrop on the traffic and gain acces. Creating tokens and shared-secrets that are long, random and resistant to eavesdrop can help to avoid this.
Also it is possible, that the secrets aren't stored safety. Solution: The secrets can be isolated and stored in a database or file-system with proper access control, file permission, physical security, and even database or disk encryption.