



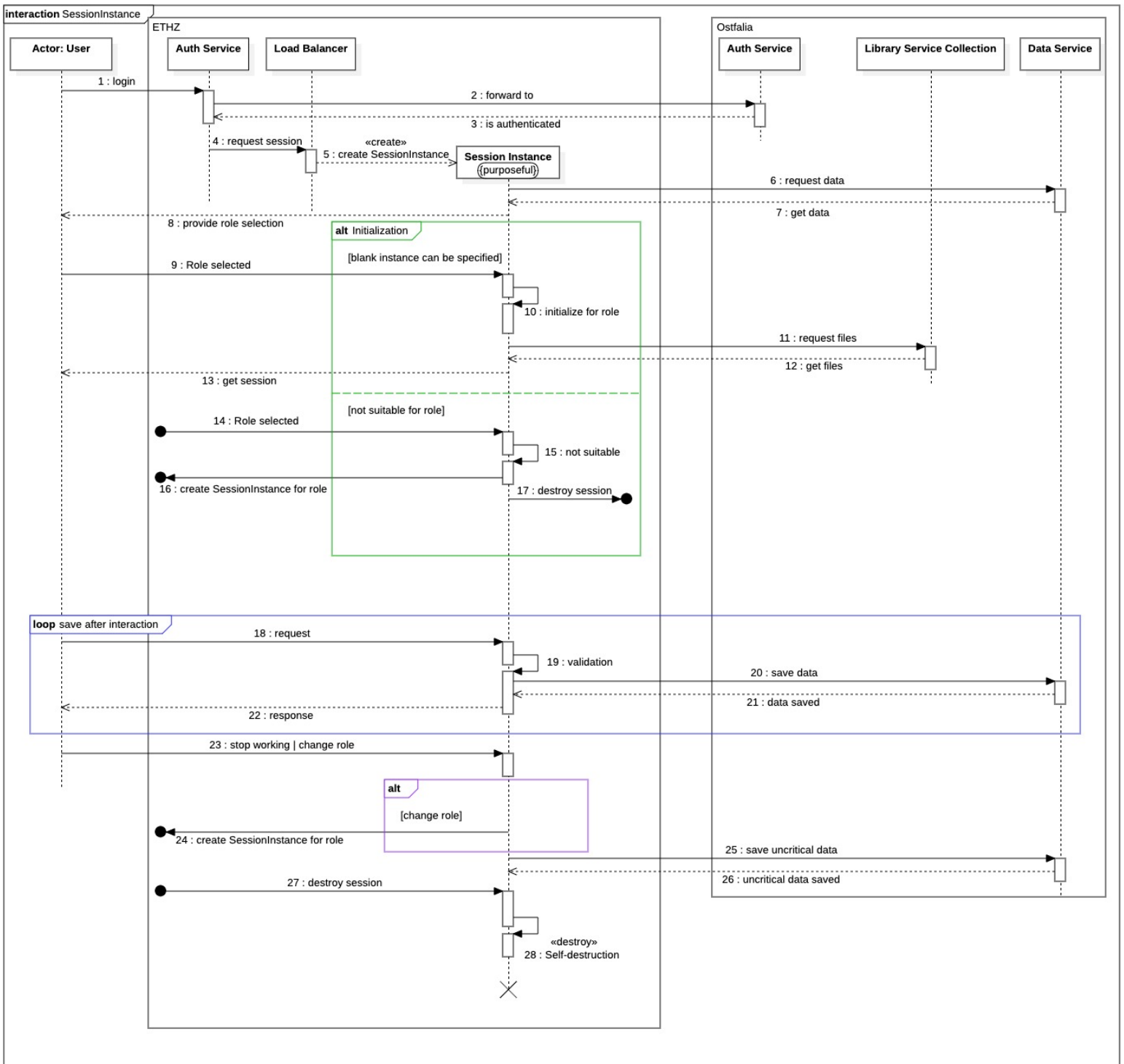
APPENDIX

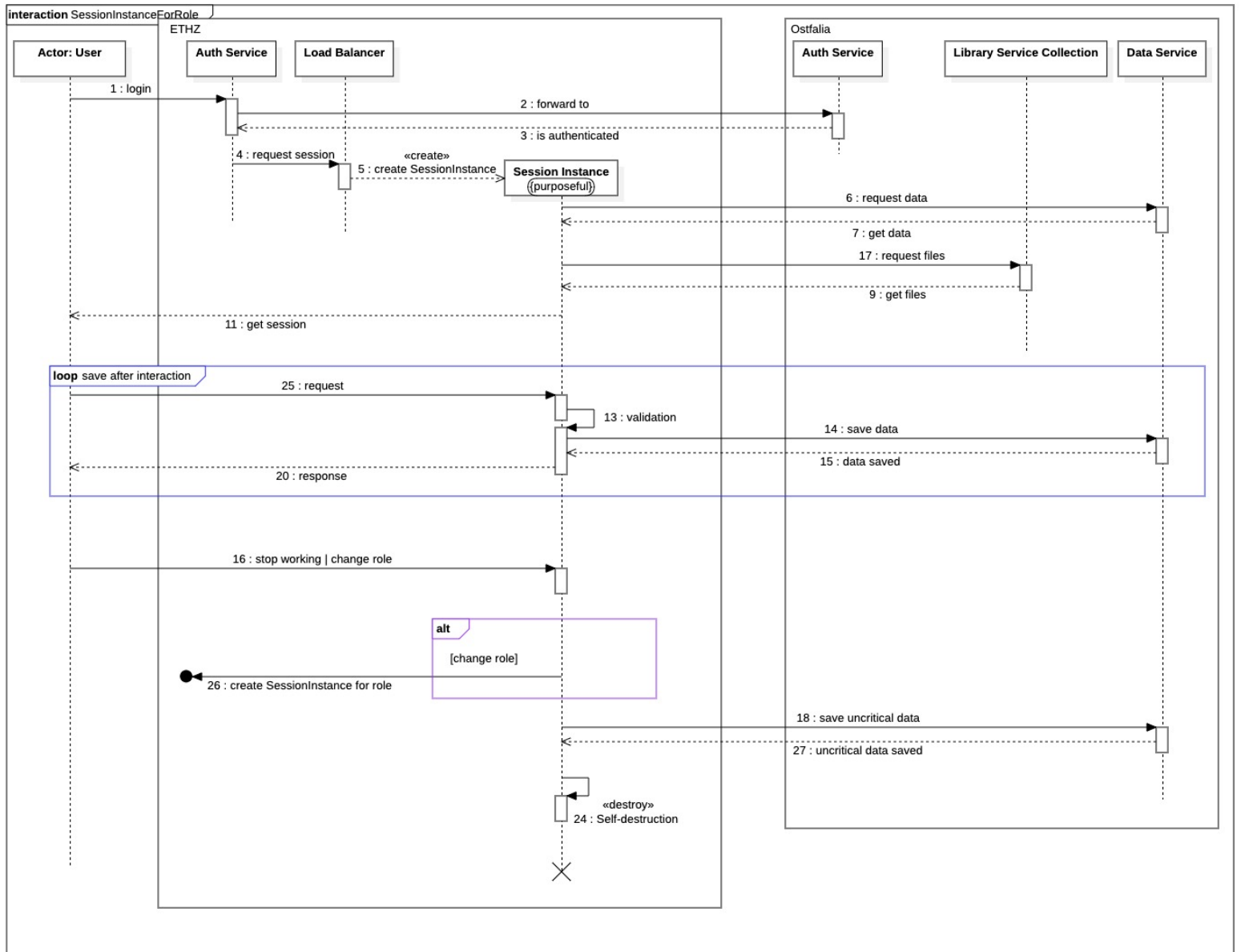
---

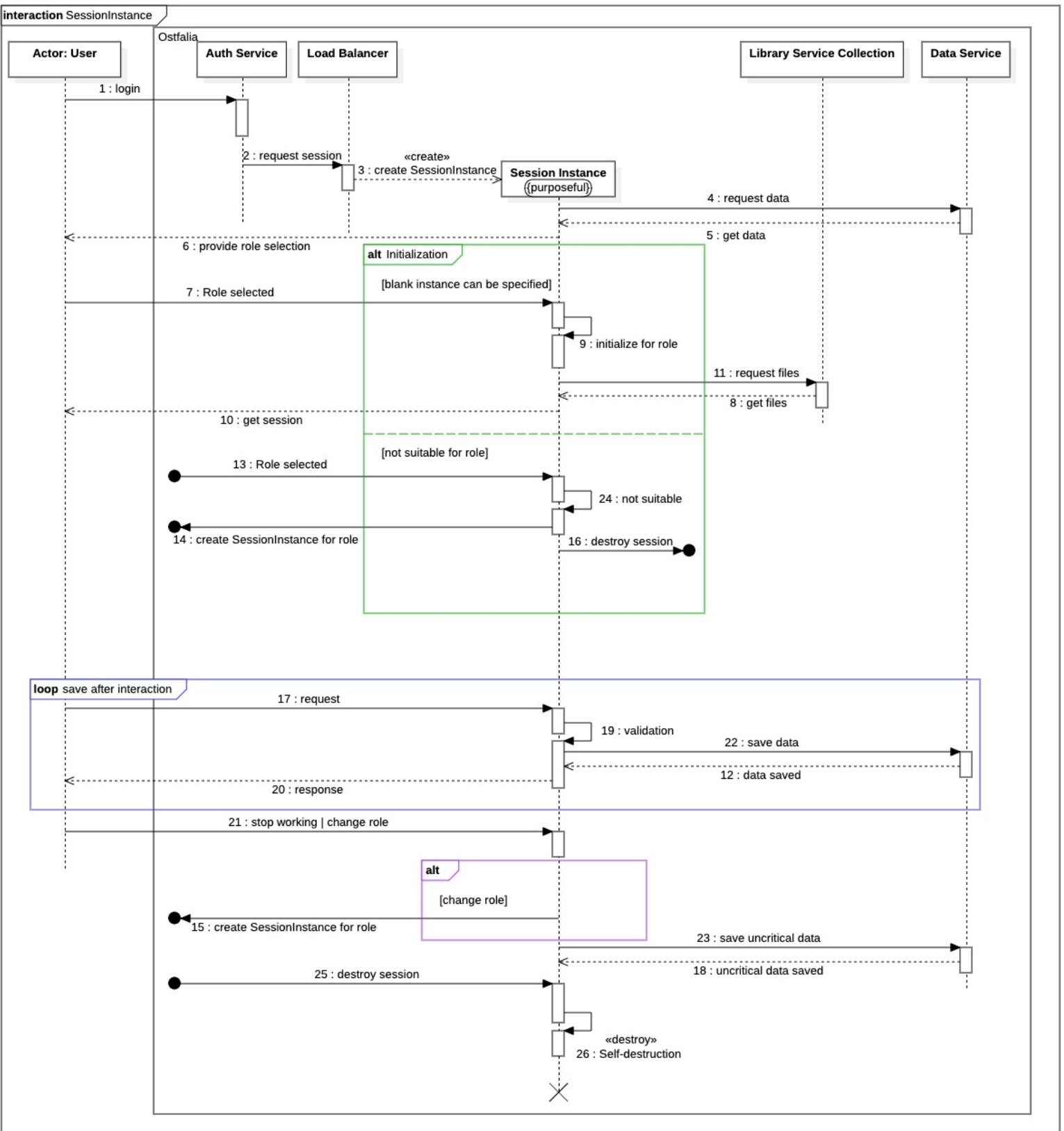
An education record blockchain to  
provide a lifelong verifiable  
educational history for students

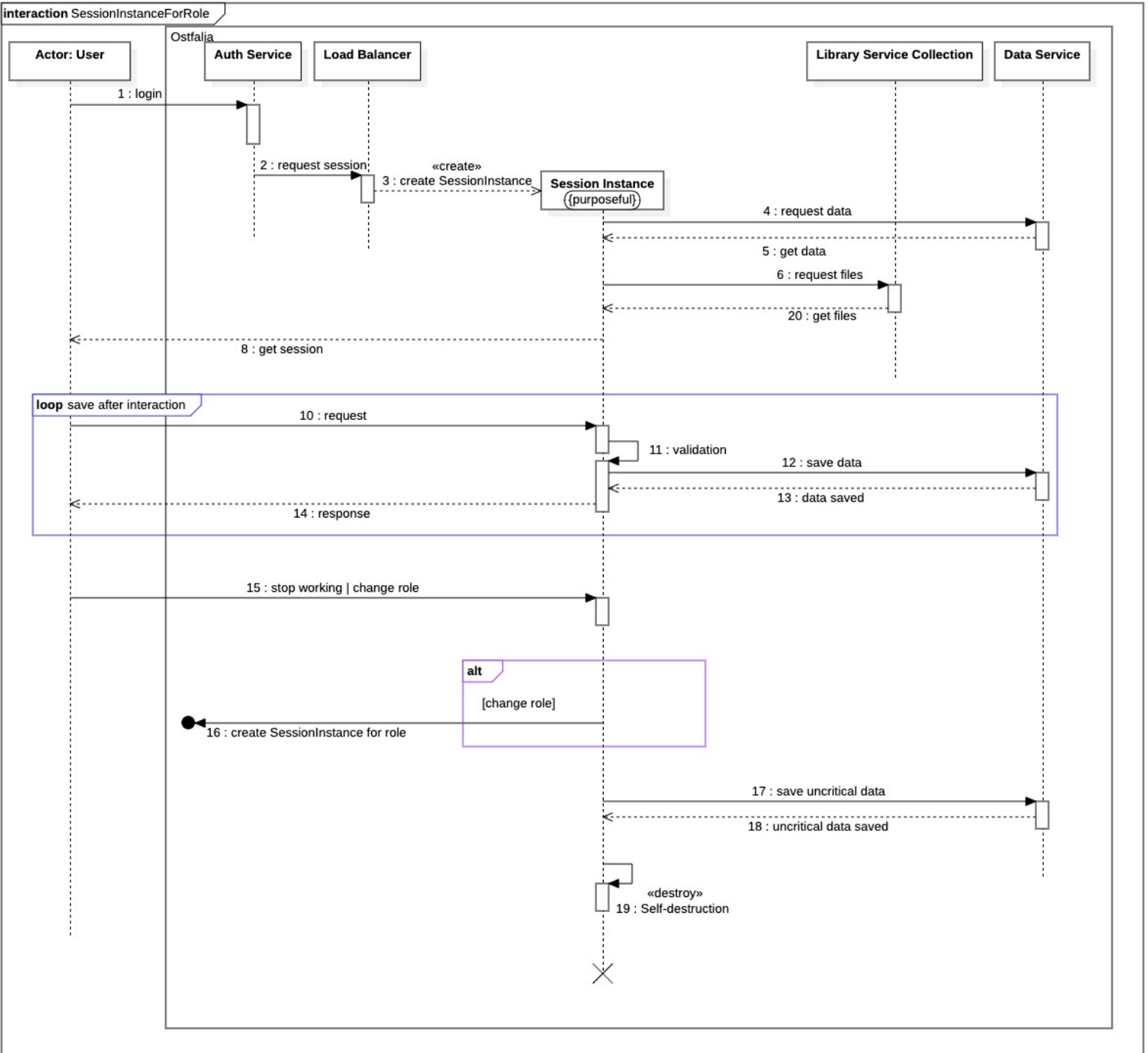
---











## B. Analysis

### B.1. What if no resources are available for an author?

getrennter Pool für Autor und Studenten? Was meinte ich mit diesem Abschnitt?

### B.2. Authentication

The concept must work for the most varied of systems with the most varied of login procedures. Apart from the fact that the concept must provide an abstraction that is independent of the authentication protocol, in addition to the login data and two-factor authentication, various processes must also be taken into account. On the one hand, to be future-proof, on the other hand, because different processes already exist today or will exist soon.

example Schweiz lebenslange ID, Ostfalia?

- Es muss für unterschiedliche Systeme und Login verfahren funktionieren, siehe Schweiz und ostfalia als Beispiel - login.msu.edu

- Ostfalia funktioniert vielleicht x@uni.com

2do:

- bei der Schweiz logt man sich vorher mit dem Lebenslagen Swiss edu ein
- beispielsweise kann man sich auch mit id oder mail einloggen, würde man dann es automatisch rausbekommen können, oder müsste man einfach erst ab der Authentifizierung starten?
- <https://www.switch.ch/edu-id/>

## B.3. REST-API & Mobile Applications

The provision of a REST-API is to be viewed critically, as we mostly work with learning-relevant data, which must be persisted at all times. This means that further work can only be carried out after successful storage. As a result, synchronous communication is required, or the system has to block until it receives a response. However, since the developer determines whether the REST request they send is synchronous or asynchronous, we cannot influence this. Also, on the one hand, we do not know how the developer handles the received data, and on the mobile side, we do not know when, whether or what kind of response we will get. Another option would be only to allow certified apps whose workflows have been approved. However, this would, in turn, entail human resources and the implementation of appropriate test procedures. Furthermore, it must be taken into account that the offline capability of mobile applications will not be possible at any time.

Synchronous communication creates unwanted monolith

## B.4. Need push service?

- Während der Klausur ändern sich Daten für den Studenten
  - benötigt etwas wie push events, wenn try, duration oder die Aufgabe sich zum Beispiel ändern.
  - früher gab es 10 Minuten Intervalle, dies ist aber schlecht und Studenten werden unruhig, wenn dies beispielsweise in einer Prüfungssituation ist und diese 10 Minuten warten müssen bis sie die Aktualisierung erhalten
  - es gibt zwei Szenarien Klausuren/Normale aufgaben und Übungsaufgaben
- 2do:
- in der normalen Aufgaben und Klausuren muss ein synchroner Prozess bestehen, wenn nicht gespeichert werden kann, dann darf es auch nicht weitergehen
  - in der normalen Aufgaben und Klausuren muss ein synchroner Prozess bestehen, wenn nicht gespeichert werden kann, dann darf es auch nicht weitergehen
  - Asynchrones Speichern ist nur bei der Übungsaufgaben zulässig, da es hier nicht schlimm wäre, wenn die Daten verloren gehen, hier existiert keine critical data safty

Definieren eines Push-Services



## B.5. LTI

möglich? Verschieden Versionen, sonst in iFrame?

## B.6. Migration Autoren-bereich

Hätte den Vorteil das eine Autoren-Rolle eine RealePerson abbilden würde  
\* LTI hat Architektur Auswirkungen \* welche Daten bei uns? \* welche über Leitung? \* LTI ins openLCMS aufnehmen



## C. Concept

In this chapter, neither the terms server, service and service collection are used as follows. A server is a physical machine or part of a physical machine which serves as a provider for a service or a service collection. A service collection is an amalgamation of two or more services. A service is a self-contained module that fulfils exactly one task.

### C.1. Concept Components

#### C.1.1. Authentication Service

The Authentication Service grants the user access to the system by validating the authentication and providing a response for successful or unsuccessful authentication. If the authentication is successful, a token is generated, and a session is requested from the load balancer. Otherwise, the user receives an error message, and a new authentication has to be requested.

In the current system, on which the user logs in, not his own home domain, the Authentication Service of the home domain of the user is required. In order to be able to request this directly, in addition to the information for authentication, the home domain must also be submitted with his credentials. Since different variants of the authentication must be provided (see [B.2](#)), the way this information is transmitted to the system also differs. Therefore there must be an interpreter for each of the different types of authentication in the Authentication Service. However, since the systems only exchange tokens with one another, this is not a problem elsewhere in the concept.

## C.1.2. Load Balancer

Vor dem Load Balancer ist eine Firewall  
z.B. TRAFIK? ist angeblich ein dynamischer Load Balancer (Erlaubt dynamisch Regeln festzulegen. Z.b. diese IP greift jetzt auf die folgende Session Instanz zu) & Kubernetes

The load balancer decides, based on the current load of the provided session server, which server will host the new session instance. Also, it can start up new session servers whenever this is necessary. If the new instance cannot be assigned to a session server and there are no other session servers available to start up, the load balancer forwards the request to host a session instance to the next university.

## C.1.3. Session Server

The session server, previously the access server, holds multiple session instances. It primarily represents the physical layer for hosting the session instances. Besides, it sends feedback to the load balancer about the current workload every time an instance is started or shut down. A resource protocol is to be introduced for this purpose. In this way, the load balancer knows the system load at all times and can initiate forwarding to another university if no more resources are available.

## C.1.4. Session Instance

A session instance is an independent per-user container hosted by a session server. An instance can have three different states. The purposeful, the blank or the unsuitable. Most of the time, a session instance is in a purposeful state. The purposeful state means that the session instance has been initialised for a specific role and can only be used to process tasks for the selected role. For example, an instance can be initialised in the purposeful state for the roles of learner, author, instructor or course coordinator. If a user has only one role, the session instance can be initialised directly for this role without having to make a previous selection. Otherwise, the role has to be chosen by the user. Until the role is selected, the instance remains in the blank state.

After selecting a role, the session instance is entirely initialised and changes to the purposeful state. However, if the user has multiple roles and chooses one that is not suitable for the physical location of the current session instance, the instance changes to the unsuitable state.

For example, the unsuitable state can occur in the following scenario. The user is assigned to an instance on a server at University A. However, the user selects the author role at University B. Since this role requires a physical session instance at University B, as this is the only place where access to the data not yet published can be granted, it means that the current instance cannot be used. This leads to the creation of a new session instance at University B and the destruction of the current session instance at University A.

pro User ein Container -> führt zu recht vielen Container, wie sollen diese heruntergefahren werden? Die meisten User werden sich nicht abmelden, somit sollte eine automatische Abmeldung bei Inaktivität durchgeführt werden. (Ähnlich wie im Online-Banking)  
Welche Zeitspanne ist sinnvoll? Welcher Teil des System ist dadurch inaktiv?

Ressourcen des Containers können basierend auf der Rolle unterschiedlich sein.

### C.1.5. Library Service Collection

The Library Service Collection consists of the Course Service, which provides the composition of all courses created by the university, and the Educational Resource Service. The latter provides all the resources that have been created and published by members of this domain as well as copies of the resources of authors from other domains that have been added to a course. This decision was made because these two services work closely together. Among other things, a request to one service often results in delegating tasks to the other service. For example, when creating a Session Instance after the course was requested from the Course Service, a copy of the course files is transferred from the Educational Resource Service to the Session Instance.

integrations of the knowledge of all domains

Each domain must know the others, otherwise a single point of failure or if one cannot be reached so that the other roles can be used

## C.1.6. Author Service

Loop ist als Student Synchron als Autoren eventuell asynchron?

## C.1.7. Data Service

The file server holds all of the user's data for its own domain as well as all references to the other domains. The user also has a home domain. The home domain

describe  
hält die Daten bis  
sie publiziert wer-  
den

Der Datenserver hält alle Daten des Users für die eigene Domäne sowie alle verweise auf die anderen Domänen. Der User verfügt darüber hinaus über eine Heimatdomäne. Oder nur verweis auf Hauptdomäne? Und nur die Hauptdomäne kennt alle andere Domänen?

integrations of the  
knowledge of all do-  
mains in which the  
user is represented

## C.2. Process Description

diagrams: add do-  
main management,  
getting data from  
domains

Beschreibung der[? ] Diagramme [? , ... SequenceDiagram]

add a reference to  
the diagrams

First, every user has to log in. It does not matter which physical location on the system the user logs on to. If he does not log into his own home domain, his authentication will be forwarded to it. After successful authentication, the local Load Balancer requests a session. If the user has several roles, all relevant data and the other domains in which he is represented are requested from his home domain first. Afterwards, all relevant data of the other domains are requested. After receiving the data, the role selection is generated from the received data and presented to the user for selecting a role. If only one role exists, it will be selected automatically. From this point on, there are two possible scenarios.

Es kann sein, dass  
der erste Satz nicht  
korrekt ist (First,  
every user has to  
log in). In loncapa  
ist das momentan  
der Fall. Autoren  
können Aufgaben  
so veröffentlichen,  
dass sie ohne Log-  
in verfügbar sind.  
Die Aufgaben sind  
dann direkt über  
die URL verfüg-  
bar und im practise  
mode. Siehe z.B.  
<https://vita.ostfalia.de/res/fhwf/bisitz/Demo/Moeglichkeiten/Auswahl/MultipleChoice.problem>

First, the Session Instance fits the selected role. Then the Session Instance for this role is initialised, and the files are requested from the corresponding domain. After successful initialisation and receipt of all required files, the session is made available to the user.

Second, the Session Instance is unsuitable for the selected role. When this occurs, a new Session Instance for the role is requested at a suitable point in the system with the selected role directly initialised and transferred to the user. The current Session Instance will be destroyed (see C.1.4).

After the user has received the session, he always communicates with it directly. The user will send requests which are validated by the Session Instance and saved directly on the associated Data Service. After validation and a successful or unsuccessful save, the user receives his response. This process needs to be synchronous, as we are mostly dealing with critical data safety<sup>1</sup>.

describe critical data safety

If the user decides to end his work, to change his role or if the user is inactive for too long, the uncritical data<sup>2</sup> is saved, and the Session Instance will be destroyed. When changing the role, a new Session Instance is initialised for the selected role and made available to the user.

describe uncritical data

Neue Instanz braucht mehr Ressourcen als verfügbar?

Domain kennt immer HomeServer des Users

Aus URL die Heimatdomäne lesen?

Nach login müssen Daten von Fremddomänen geholt werden, alle Domänen abfragen

Ressourcen lokal nicht vorhanden, dann home Server kontaktieren (alle LibServer der Heimatdomäne des authors) hier wird noch nach und nach jeder server abgefragt in zukunft sollte nur der "LibServer(Software)" in der Heimatdomäne angefragt werden ob etwas da ist oder nicht

statt Heimatdomäne überall diese Informationen halten an welchen Domänen der User noch existiert? Und beim Login falls für die Domäne an der er sich physisch mit seiner Session Instanz befindet nur eine Rolle vorhanden ist für diese direkt initialisieren? Die Heimatdomäne des Users hält darüber hinaus auch noch die Fremddomänen in denen der User existiert und somit Daten von ihm. Die Heimatdomäne ist per default

Fremddomänen integrieren

URL: Server/domaene/author Aufbau Datei URL

lon-capa hat auch ein URL Hiding feature

<sup>1</sup>(You can read here what is meant by this.)

<sup>2</sup>(here it is said what that is)