Entwurfsbeschreibung

Web Annotation mit Fragment Ids

Gruppe: swp12-9

Inhaltsverzeichnis

1. Allgemeines

Im Rahmen dieses Projekts wird im Wesentlichen eine Möglichkeit zur Annotation und Markierung spezieller Textinhalte von HTML-Seiten geschaffen. Die Software soll einen Endbenutzer damit in die Lage versetzen, konkrete Inhalte und Informationen aus häufig unstrukturierten und überfüllten Websites zu filtern und performant mit anderen Webnutzern zu teilen. Daraus ergeben sich keine Einschränkungen für potenzielle Anwender der Software, da das Teilen und Kommentieren verschiedenster Inhalte im gesamten World Wide Web für eine Vielzahl von Anwendern von Interesse sein kann.
Die geplante und weitere mögliche technische Umsetzungsmöglichkeiten dieser Anforderungen werden im Folgenden zusammenfassend dargestellt.
Anschließend wird die Interaktion der einzelnen Pakete und Module, sowie deren Funktionsweise konkretisiert.

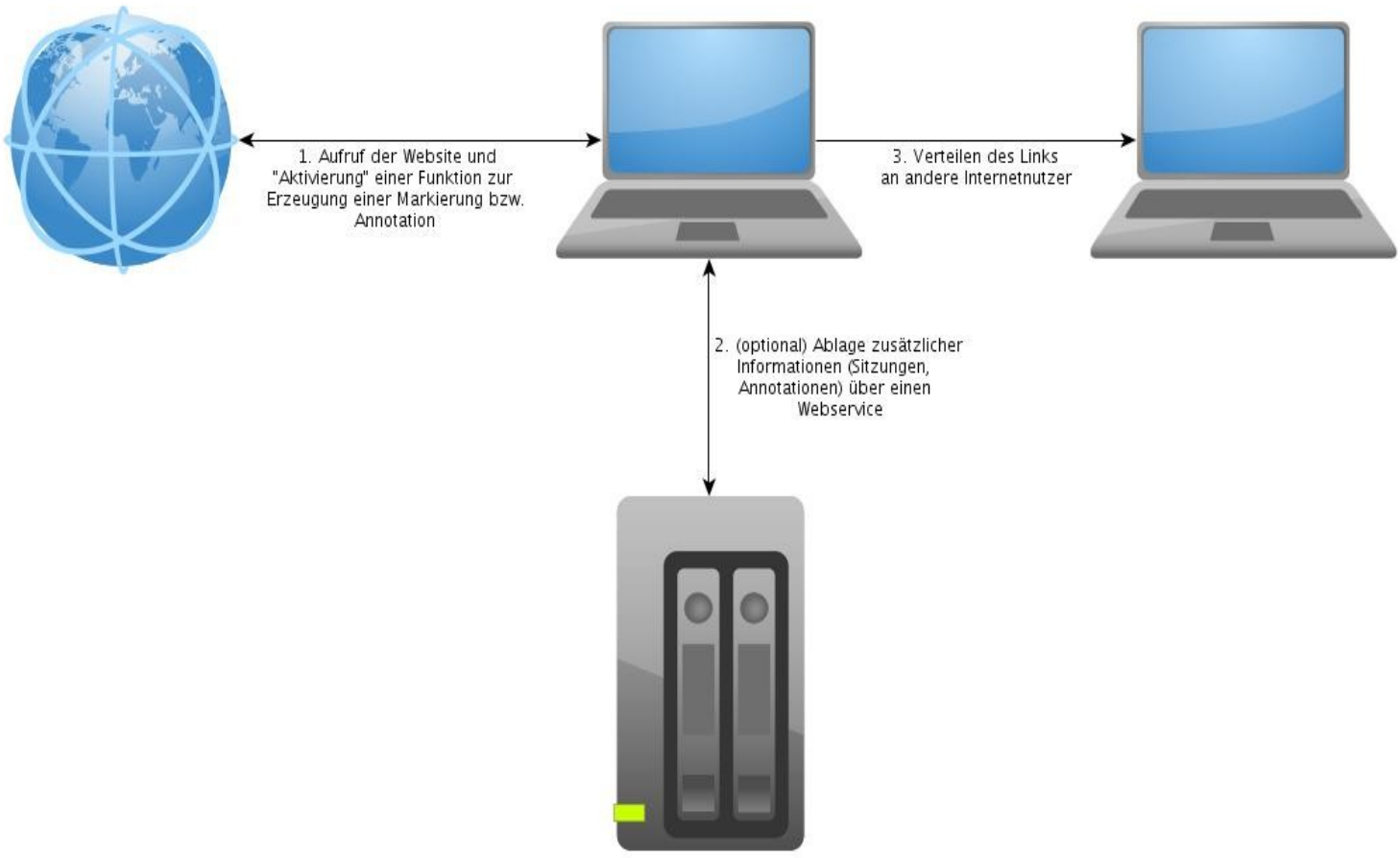


Abbildung 1: Funktionsweise

Grundlegend können bei der Implementierung eines Annotationssystem (mit einem Annotationsserver) 2 verschiedene Paradigmen unterschieden werden, entweder

- die **Manipulation und Auslieferung** einer vom Nutzer per Fragment Link abgefragten Websource geschieht **durch einen Webservice** (*PHP*)
  - Vorteile: Anzeigen von Annotationen und Markierungen ohne zusätzliche Software möglich (nach Eingabe eines Fragment Links),
  - Nachteile: parallele Verarbeitung von vielen Clientanfragen setzt genügend Serverressourcen vorraus
- oder die **Manipulation** geschieht, **nachdem** die **benötigten Informationen** nach Eingabe eines Fragment Links vom Server **übermittelt wurden**, **direkt im Browser des Clienten** (*JavaScript*)
  - Vorteile: serverseitige Verarbeitung von großen Webressourcen entfällt, geringerer Traffic
  - Nachteile: Software notwendig zur Anzeige von Annotationen/Markierungen

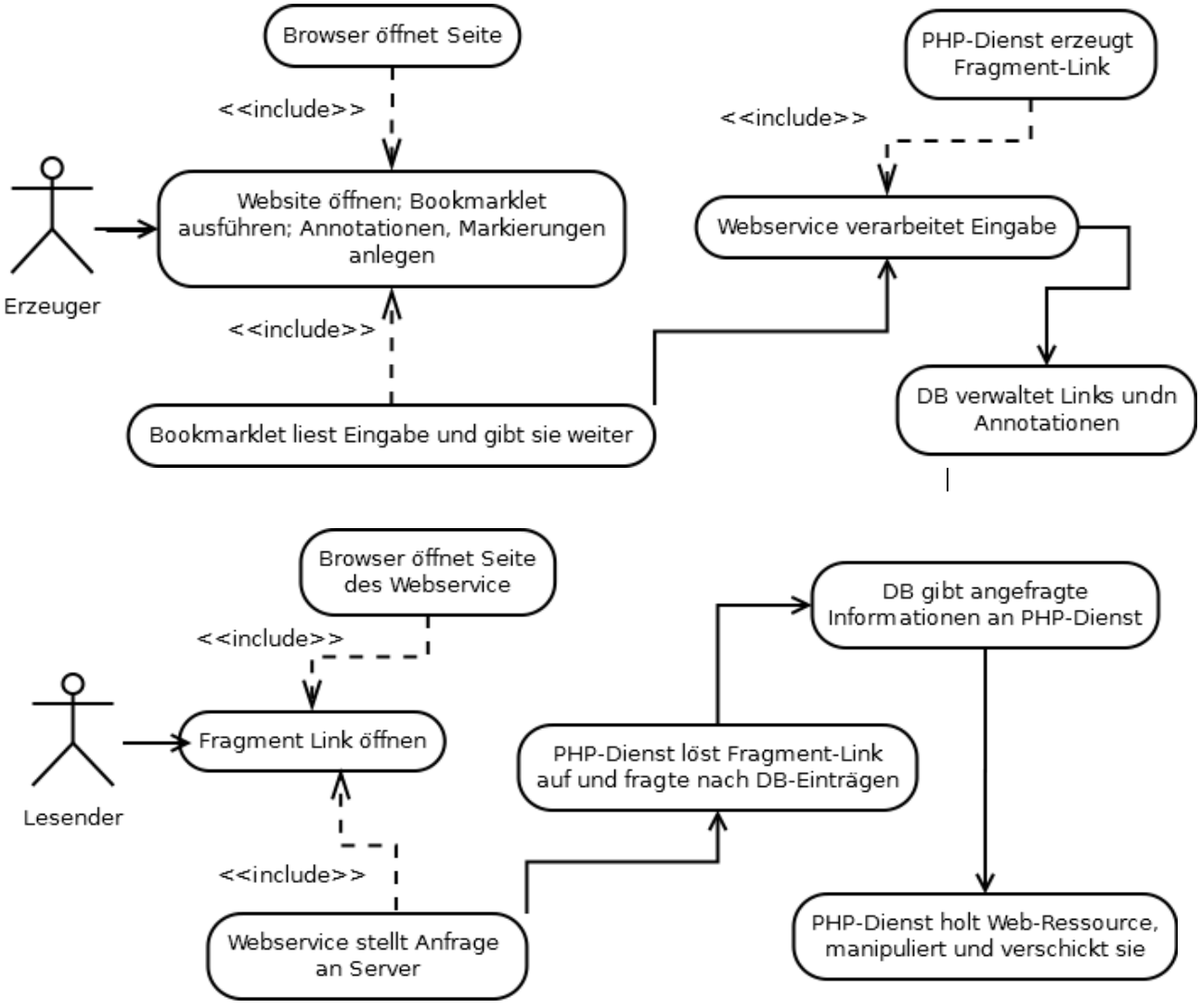
Desweiteren kann man die Art der Benutzer-Schnittstelle zum Server differenzieren:

- **Browser-Plugin**
  - Vorteile: einfache Installation, Funktionsgarantie für bestimmten Browser, komfortable Bedienung und Einstellung
  - Nachteile: Version- und Browserabhängigkeit, Wartung und Pflege notwendig
- **Bookmarklet**
  - Vorteile: funktioniert mit allen gängigen Browsern, einfache Installation
  - Nachteile: JavaScript muss aktiviert sein
- **WebVPN (Einbettung von Webressourcen in eigene Webpräsenz)**
  - Vorteile: keine Installtion von Clientsoftware notwendig
  - Nachteile: beschränkter Zugriff auf Netzlaufwerke, Flash-Inhalte u.a. fehlen oder funktionieren u.U. nicht

3. Produktübersicht

Die Umsetzung erfolgt in unserem Fall über einen konkreten Webservice, der Webseiten manipulieren und verschicken kann, sowie durch ein Bookmarklet, was notwendigen Programmcode laden kann, um Webressourcen zu markieren/annotieren und die Kommunikation mit dem Webserver zur Fragment Link Erzeugung sicher zu stellen. Das Produkt lässt sich demnach in zwei Bestandteile zerlegen. Zum einen die Erzeugung des Hash Links, der auf die zu markierende/ annotierende Stelle verweist. Darauf aufbauend wird im Anschluss der Fragment-Link erzeugt, der dann auch auf mehrere Markierungen oder Annotationen verweisen kann. Zum anderen kommt dem Webservice eine wesentliche Rolle zu, da dieser dafür verantwortlich ist, die annotierte und/ oder markierte Seite zu manipulieren, damit die Zusatzinfos sichtbar werden.

Usecasediagramm:



Beschreibung des Usecasediagramms:

Im oberen Abschnitt ist zu sehen wie der User eine gewünschte Markierung / Annotation zu einer Webseite erstellt um einen Fragment-Link zu erhalten. Im unteren Abschnitt wird gezeigt, wie der Fragment-Link im Browser angeschaut werden kann.

4. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

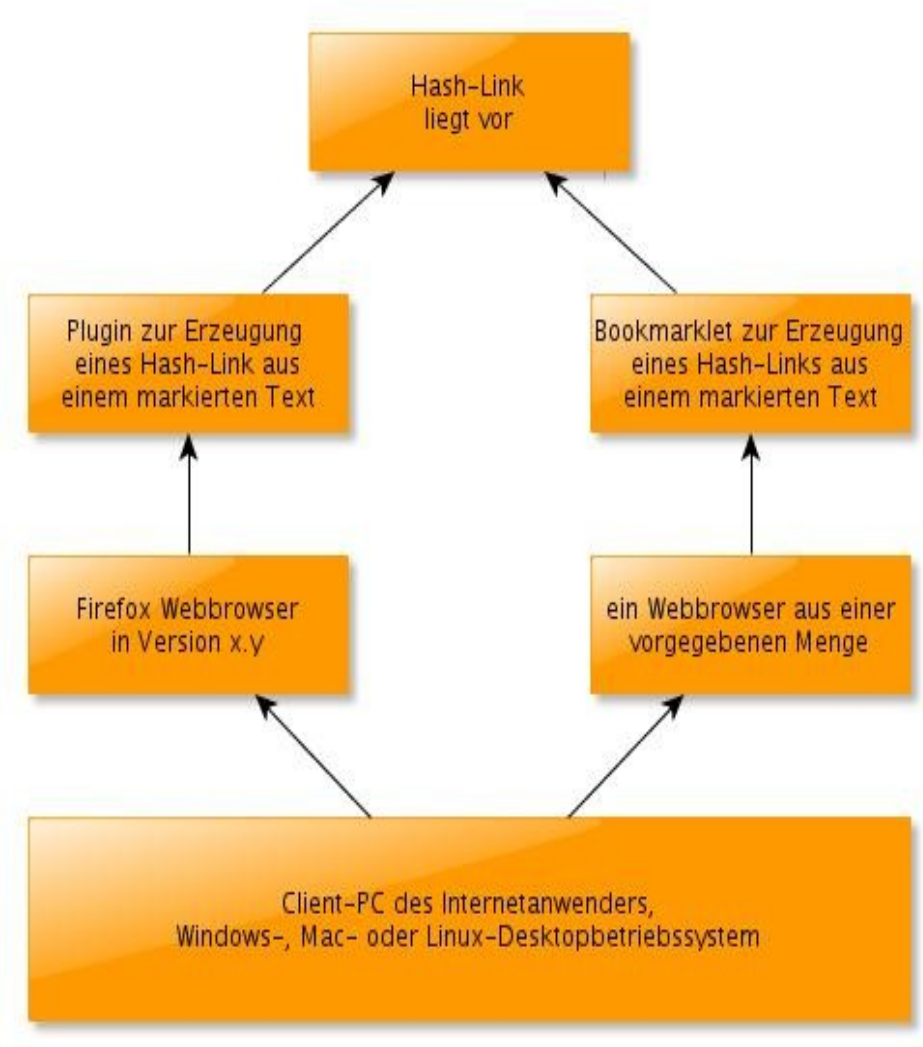


Abbildung 2: Erzeugung eines Hash Links – Gegenüberstellung Plugin/ Bookmarklet

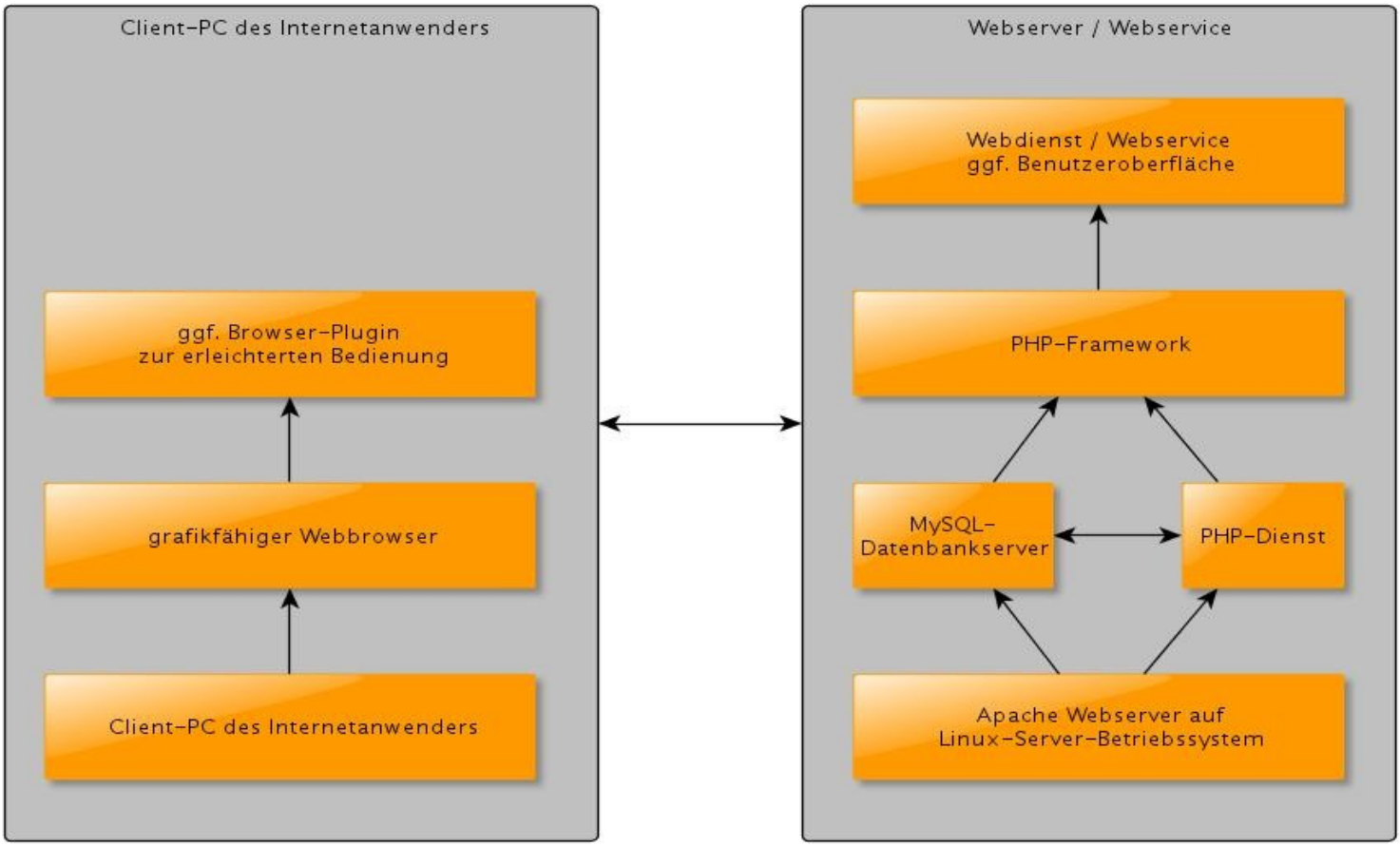


Abbildung 3: Webservice

5. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

Es wird ein Apache-Webserver mit Linux-Betriebssystem zur Verfügung stehen, welcher für das permanente Aufbewahren der Annotationen und Markierungen zuständig ist und sich um die Übertragung der annotierten/ markierten Webseiten kümmert. Für diese Zwecke wird eine Datenbank auf diesem installiert und weitere Funktionen, wie das Auslesen eines Fragment-Links durch entsprechende PHP-Dienste zur Verfügung gestellt. Die genutzte MySQL-Datenbank ist zudem für die Organisation und Verwaltung der gespeicherten Daten, wie Annotationen, Markierungen oder Sitzungen zuständig. Das Speicherungsformat wird so gewählt, dass die Daten sich einfach als RDF-Tripel (z.B. Hashlink, „Kommentar“, „Das ist ein Kommentar“) abrufen lassen.

5.1 Bookmarklet

Einem Internetnutzer (unter Abbildung 1 zentral dargestellt), der einen Webinhalt markieren oder annotieren und teilen möchte, wird eine Funktionalität zur Verfügung gestellt, die ihm ermöglicht aus der Browseransicht eines HTML-Dokuments einen gewünschten Inhalt auf Quelltextebene zu extrahieren. Als Resultat entsteht ein Hash-Link, welcher aus der Adresse der HTML-Seite und dem extrahierten String erzeugt und berechnet wird. Des Weiteren erhält der Nutzer die Möglichkeit, Kommentare anzufügen. Diese Funktionalitäten werden von einem Bookmarklet zur Verfügung gestellt.

Das Starten des Bookmarklets führt durch nachladen von Javascriptcode von unserem Webserver zur Erweiterung der gegebenen Browserfunktionen. Dabei wird eine GUI geladen, die dem Nutzer ermöglicht, zu markierten Elementen weitere Funktionen auszuführen (beispielsweise die Fragment-Link-Erzeugung oder Kommentierung). Außerdem sorgt diese geladene GUI für die Ausgabe des entstandenen Fragment-Links oder Hash-Links.

5.2 PHP-Dienst

Der PHP-Dienst wird die serverseitigen Grundfunktionalitäten bieten.

Die erste Funktionalität wird das Empfangen der Eingangsdaten (Hashlinks und ggf. Kommentare/ Fragment Links).

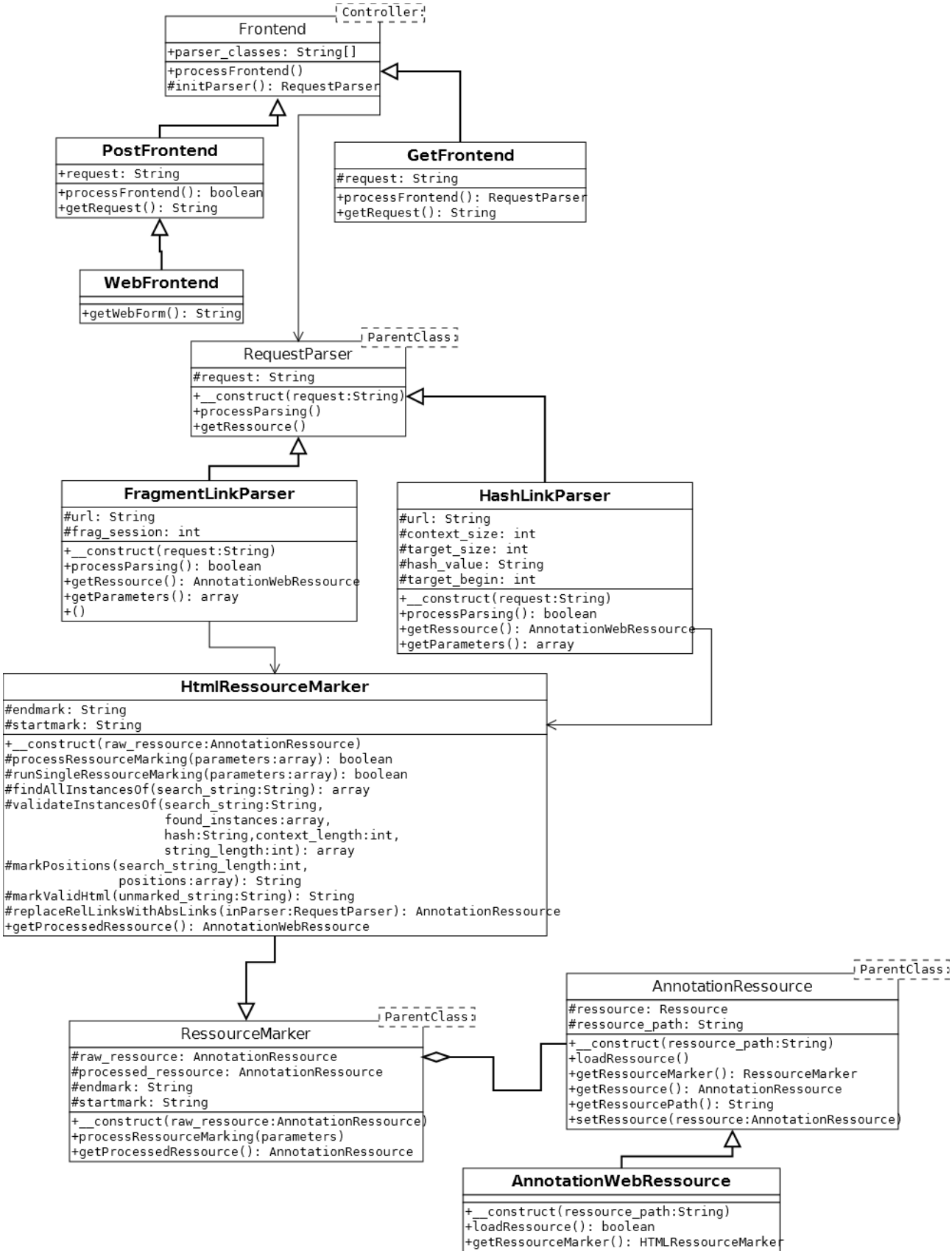
Der Hashlink oder Fragment-Link muss nachfolgend verstanden werden. Dass heißt, die Informationen, wie z.B. welche Ressource markiert und annotiert werden soll, müssen separiert werden.

Anschließend muss die Ressource (z.B. HTML) geladen werden und der im betrachteten Hashlink beschriebene Inhalt (Länge des Strings, die ersten 20 Zeichen und der MD5-Hash aus dem Kontext) wird in der Ressource gesucht.

Eine wesentliche Funktionalität stellt das Bearbeiten der Ressource dar. Hier soll der gefundene Inhalt markiert werden und die Seite mit einem möglichen Kommentar zur markierten Stelle erweitert werden. Außerdem müssen alle relativen Pfade aus dem Original in absolute Pfade umgeschrieben werden, damit z.B. Bilder im Anschluss weiterhin sichtbar sind.

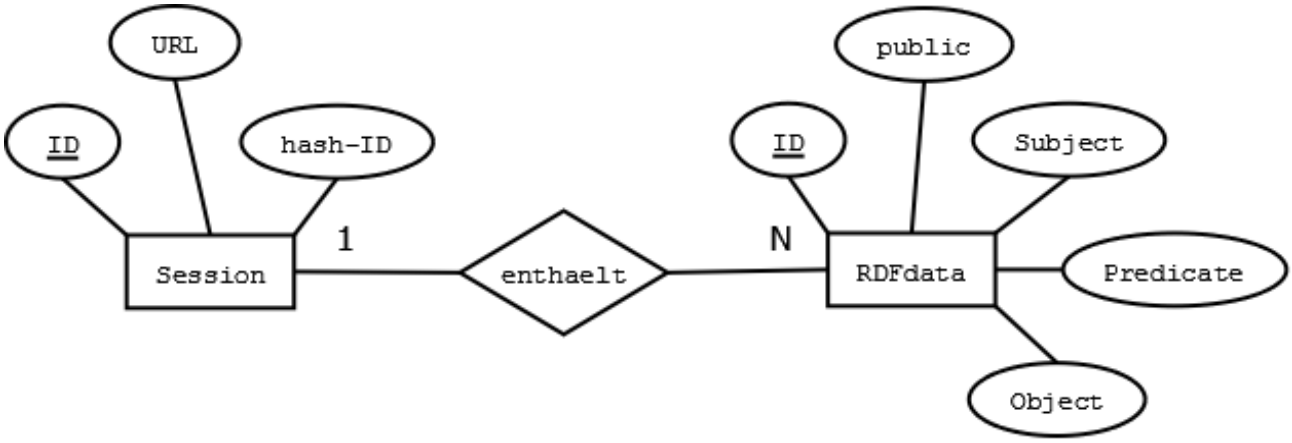
Das Speichern der Informationen in der Datenbank und das Ausliefern der bearbeiteten HTML-Seite bzw. des Weblinks, der sie ansteuert, sind die finalen Funktionalitäten des PHP-Dienstes.

UML-Diagramm:



5.3 Datenbank

ER-Modell:



5.4 OpenLetters Framework

Das OpenLetters Framework stellt php-Klassen zur Verfügung, die den Zugriff auf andere Quellen wie Datenbanken oder Dateien innerhalb des Programms vereinheitlichen sollen. Vor allem werden hiermit Logging, Datenbankzugriff und Datei- bzw. Datenbankzugriff realisiert.

6. Kommunikation zwischen Bookmarklet und Webservice

Die Kommunikation zwischen Bookmarklet und Webserivce wird mittels Ajax erfolgen. Grundsätzlich ist der einfachste und am besten Brower-übergreifend funktionierende Weg die Nutzung der Funktion jQuery.post http://docs.jquery.com/Post

Beispiel:

```
$.post(
"test.php", // aufzurufende URL
 { func: "getNameAndTime" }, // liste an Daten für den Webserver
  function(data){ // JavaScript-Funktion die nach Erfolg zu rufen ist, "data" ist das vom
                  Server zurückgegebene Array
  alert(data.name);
  console.log(data.time);
 },
  "json" // man kann auch mit anderen Formaten als Json kommunizieren, wollen wir
           aber nicht
 );
```

Erklärung zum Beispiel:

- test.php ist die zu rufende Internetadresse. Diese muss in unserem Projekt absolut sein (http://www.my-annotation.org/index.php?parameter1=wert1&paramater2=wert2)
- Der 2te Parameter ist ein Array von Daten (hier wird dem Array-Schlüssel "func" der Wert "getNameAndTime" zugewiesen und dem Server übermittelt), in unserem Fall sind darin die Json-encodierten RDF-Daten
- Die im Beispiel definierte "function" (enthält hier ein "alert" und ein "console.log") wird im Browser ausgeführt, sobald die Anfrage erfolgreich vom Server beantwortet wurde. In unserem Fall könnte dort das Anzeigen des FragmentLinks an den User geschehen. Die darin verwendete Variable "data" enthält die Antwort des Servers.

Unsere Internetadresse (erster Parameter URL) könnte so aussehen: http://www.my-annotation.org/index.php?parameter1=wert1&paramater2=wert2 Statt "parameter1" und "parameter2" gibt es bei uns diese Werte:

- application=webservice sagt dem Webserver, dass er mit dem Bookmarklet redet und nicht (wie bisher) eine Website ausliefern soll. Diesen Parameter muss das Bookmarklet immer mitliefern.
- annotation request=url führt dazu, dass der Nutzer zwar die von ihm gewünschte Seite sieht, sich aber physisch auf der Internetadresse von www.myannotation.org befindet. Die ist nötig, da die meisten Browser Cross-Site-Scripting blockieren
- return=rdf wird in Zukunft möglich sein, aktuell geben wir immer JSON zurück. Daher muss dieser Parameter nicht aufgelistet sein.
- action=add_many (für viele Datensätze) oder add (für einen Datensatz) soll mir sagen, dass im übermittelten JSON neue Hashlinks drin sind, die zu speichern sind.

Das JSON im 2ten Parameter der Post-Funktion sieht so aus:

```
{
   {  subject: http://www.meine-
      seite.de/Impressum#hash_4_26_wzucwecguqecuoqc_Such_Wort,
      predicate: hasMarkup oder hasComment,
      object: leer (bei hasMarkup) oder der Kommentar des Users,
      public: true oder false,
      fraglink: der Fragmentlink dem der Hashlink hinzugefügt werden soll (falls vorhanden,
sonst Parameter weglassen)
   },
   {
      ... ggf. ein weiterer Datensatz
   }
}
```

Der Server antwortet dann mit diesem JSON ((Miss-)Erfolgsmeldung und FragmentLink):

```
{
   success: true oder false,
   fraglink: www.meine-seite.de/Impressum#frag_(0-9a-f){32} (Dieser Parameter wird nur
zurückgegeben, wenn "success" den Wert "true" hat.)
}
```

Weitere Antworten sind bisher nicht vorgesehen, sind durch das Array aber sehr gut möglich.