



UNIVERSITÄT OSNABRÜCK

INSTITUT FÜR INFORMATIK
AG SOFTWARE ENGINEERING

Masterarbeit

Anbindung von Messaging-Systemen an Lernmanagementsysteme (am Beispiel von Stud.IP und Matrix)

Manuel Schwarz

Juli 2021

Erstgutachter: Dr. Tobias Thelen
Zweitgutachterin: Prof. Dr. Elke Pulvermüller

Zusammenfassung

Die vorliegende Arbeit ...

Inhaltsverzeichnis

1	Motivation	1
2	Hintergrund	3
2.1	Lernmanagementsysteme	3
2.1.1	Stud.IP	4
2.1.2	Stud.IP Schnittstellen	4
2.2	Anbindung von Fremdsystemen an Stud.IP	5
2.2.1	Meetings Plugin	5
2.2.2	Etherpad Plugin	7
2.2.3	Moodle-Connect Plugin	7
2.2.4	LTI-Tools / Alija Plugin	7
2.3	Messenger	8
2.3.1	Matrix	8
2.3.2	Matrix Schnittstellen	8
3	Anforderungsanalyse	9
3.1	Umfrage zur Anforderungsanalyse	9
3.2	Grundsatzentscheidung	10
3.3	Anforderungen	10
3.3.1	Übernahme von Veranstaltungsinformationen Veranstaltungen auf Räume abbilden + Übernahme der Teilnehmendenlisten	10
3.3.2	kein zusätzliches Login	10
3.3.3	Benachrichtigungsfunktion	10
3.3.4	Direktnachrichten	11
3.3.5	Transparenz	11
3.3.6	Übernahme von Rollen, Rechten und Status	11
3.3.7	mehrere Räume pro Kurs	12
3.3.8	Kompatibilität von Nachrichten in Blubber und Matrix	12
3.3.9	Übernahme des Stud.IP Profilbildes	12
3.3.10	Opensource	13
3.3.11	Stud.IP Plugin	13
3.3.12	keine zusätzliche API	13
3.3.13	Sicherheit	13

3.3.14	Robustheit, Ausfallsicherheit	13
4	Implementation	15
4.0.1	Übernahme von Veranstaltungsinformationen Veranstaltungen auf Räume abbilden + Übernahme der Teilnehmendenlisten	15
4.0.2	kein zusätzliches Login	15
4.0.3	Benachrichtigungsfunktion	15
4.0.4	Direktnachrichten	16
4.0.5	Transparenz	16
4.0.6	Übernahme von Rollen, Rechten und Status	16
4.0.7	mehrere Räume pro Kurs	17
4.0.8	Kompatibilität von Nachrichten in Blubber und Matrix	17
4.0.9	Übernahme des Stud.IP Profilbildes	17
4.0.10	Opensource	18
4.0.11	Stud.IP Plugin	18
4.0.12	keine zusätzliche API	18
4.0.13	Sicherheit	18
4.0.14	Robustheit, Ausfallsicherheit	18
5	Deployment	19
6	Evaluation	21
6.1	Technische Tests	21
6.1.1	Unit-Tests	21
6.1.2	Code Coverage	21
6.2	Chancen	21
6.3	Probleme	21
7	Ausblick	23

Kapitel 1

Motivation

Praesens, Passivkonstruktionen, KEIN Erlebnisbericht Ich habe für X/Y entschieden, eigene Schlussfolgerungen kenntlich machen Im Hintergrund keine wertenden Einschätzungen!

Lehren und Lernen im digitalen Zeitalter.

Moderne Kommunikationswege und - mittel studiumsunterstützend einsetzen.

Aktuelle Werkzeuge nutzen, um Studierenden eine möglichst niedrige Einstiegsschwelle bei Fragen oder Unklarheiten zu bieten.

Fortwährender Prozess der Weiterentwicklung und Anpassung der Vermittlung von Informationen.

Mailinglisten, Foren, Instant-Messenger . . .

Wie studiert man heute? Wie ist die durchschnittliche Nutzung der Studierenden von WhatsApp und Co.? (Bezug auf die Studien)

Digitalisierung generell hervorheben, speziell in den vergangenen Corona- bzw. Digitalsemestern noch einmal zusätzlich an Relevanz gewonnen.

Kapitel 2

Hintergrund

2.1 Lernmanagementsysteme

Lernmanagementsysteme, im Folgenden mit LMS abgekürzt, dienen der Unterstützung von Lehr- und Lernprozessen im Kontext von E-Learning sowie der Verwaltung von Lehrmaterialien und Nutzerdaten. Diese Online-Lernplattformen bilden an vielen Hochschulen aber auch anderen Institutionen die Basis einer E-Learning-Infrastruktur.

Die Hauptfunktion, die ein LMS in der Regel erfüllt, ist die Organisation des Lehrbetriebs, insbesondere das Anlegen von Veranstaltungen, die Bereitstellung von Lerninhalten und die Kommunikation zwischen Lehrenden und Lernenden.

Definition nach Schulmeister [3, S. 10]: Als Lernplattformen oder Learning Management Systeme werden im Unterschied zu bloßen Kollektionen von Lehrskripten oder Hypertext -Sammlungen auf Webservern solche Software -Systeme bezeichnet, die über folgende Funktionen verfügen:

- Eine Benutzerverwaltung (Anmeldung mit Verschlüsselung)
- Eine Kursverwaltung (Kurse, Verwaltung der Inhalte, Dateiverwaltung)
- Eine Rollen- und Rechtevergabe mit differenzierten Rechten
- Kommunikationsmethoden (Chat, Foren) und Werkzeuge für das Lernen (Whiteboard, Notizbuch, Annotationen, Kalender etc.)
- Die Darstellung der Kursinhalte, Lernobjekte und Medien in einem netzwerkfähigen Browser.

Die Abbildung 2.1 zeigt die Architektur eines idealtypischen LMS. Auf der Ebene der Datenbasis

- Content-Management; Archivierung, Distribution und Wiederverwendung von Lerninhalten
- Administrative Ebene: Nutzer- + Kursverwaltung, Institutionen, Evaluation
- Lernumgebung: Kommunikation, Werkzeuge

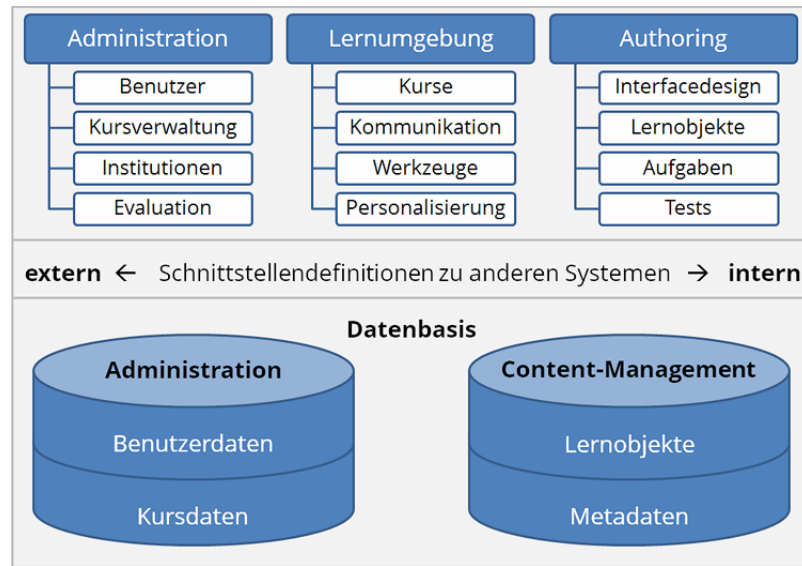


Abbildung 2.1: LMS Architektur [3, S. 11]

2.1.1 Stud.IP

Stud.IP ist eine Open Source Implementation eines LMS, dass Anfang der 2000er Jahre entwickelt wurde. Mit ihren offenen Schnittstellen erlaubt die digitale Lernplattform die Integration von externen Systemen und Anwendungen zum Einsatz in der digitalen Lehre. Neben Hochschulen gehören ebenfalls Schulen, Unternehmen, Verbände und Behörden zu den Einsatzgebieten von Stud.IP.

Insgesamt gibt es derzeit ca. 70 Installationen mit ungefähr 600.000 Nutzerinnen und Nutzern (Quelle: studip.de). Stud.IP versucht durch stetige Verbesserungen und Anpassungen den Anforderungen der sich ändernden Gegebenheiten der digitalen Lehre gerecht zu werden. Dabei ist ein kontinuierlicher Dialog zwischen Lehrenden, Lernenden und Entwicklern essentiell.

Stud.IP erfüllt alle im vorherigen Abschnitt 2.1 aufgeführten Anforderungen an ein LMS.

Neben einer ausgereiften Benutzer- und Kursverwaltung inklusive eines Rollen- und Rechtensystems, gibt es ebenfalls ein Forum und einen Chat zur Kommunikation sowie diverse Lernwerkzeuge (Etherpad, Wiki, Courseware, Vips)

2.1.2 Stud.IP Schnittstellen

Stud.IP verfügt über eine JSON-API, mit deren Hilfe man diverse Daten abrufen sowie dem System hinzufügen kann, vorausgesetzt man verfügt über die entsprechenden Rechte. Besonderes Augenmerk auf Kommunikation zwischen Lehrenden und Lernenden (natürlich auch untereinander). JSON-API (Blubber) <https://hilfe.studip.de/develop/Entwickler/JSONAPI>

2.2 Anbindung von Fremdsystemen an Stud.IP

Im Kontext des Lernmanagementsystems Stud.IP existieren mehrere Beispiele für die Anbindung oder Einbettung von anderen Systemen an/in die LMS-Software. Im Folgenden werden einige Beispiele dafür aufgeführt.

2.2.1 Meetings Plugin

Meetings ist ein Stud.IP Videokonferenzplugin (Quelle github), das das LMS mit verschiedenen/unterschiedlichen Videokonferenzsoftwarelösungen, wie z.B. BigBlueButton (Quelle bigbluebutton.org), verbindet. Das Plugin ermöglicht es direkt aus einer Stud.IP-Veranstaltung heraus Räume für z.B. Vorlesungen, Webinare, Vorträge oder Gruppenarbeiten anzulegen, denen die Teilnehmer*innen der Veranstaltung beitreten können (dank der Rechteverwaltung in Stud.IP). Weiterhin können auch externe Personen (ohne Stud.IP-Nutzerkonto) per Einladungslink an Videokonferenzen teilnehmen.

- Kommunikation über Mikrofon und Kamera + Screen Sharing
- ideal für dezentrales Lernen und Arbeiten
- Möglichkeit von Aufzeichnungen
- Screenshot des "Raum erstellen" Dialogs mit all seinen Optionen, kurz darauf eingehen
 - Funktionsweise kurz erläutern -> Nutzen der BBB API
- BigBlueButton API zum erstellen eines Meetings (create):

[http://yourserver.com/bigbluebutton/api/create?\[parameters\]&checksum=\[checksum\]](http://yourserver.com/bigbluebutton/api/create?[parameters]&checksum=[checksum])

In Stud.IP direkt

```

1  <? php
2  private function performRequest($endpoint, array $params = array(), array $options = [])
3  {
4      $uri = 'api/' . $endpoint . '?' . $this->buildQueryString($params);
5      .
6      ...
7      .
8      $method = (is_array($options) && count($options)) ? 'POST' : 'GET';
9      $request = $this->client->request($method, $this->url . '/' . $uri, $options);
10     return $request->getBody(true);
11 }

1  <?php
2  function getMeetingInfo(MeetingParameters $parameters)
3  {
4      $params = array(
5          'meetingID' => $parameters->getRemoteId() ?: $parameters->getMeetingId()
6      );
7
8      $response = $this->performRequest('getMeetingInfo', $params);

```

Raumkonfiguration ✕

Raumname

Konferenz Systemeinstellung

Konferenzsystem *

BigBlueButton ▾

Verfügbare Server *

Bitte wählen Sie einen Server aus ▾

Zusätzliche Funktionen

- ☐ Alle Teilnehmenden haben Moderationsrechte
- ☒ Alle Teilnehmenden initial stumm schalten
- ☒ Jeder Teilnehmer kann die Konferenz starten
- ☐ Nur Moderatoren können Webcams sehen ⓘ
- ☐ Nur Moderatoren können Webcams teilen
- ☐ Nur Moderatoren können Audio teilen
- ☐ Gemeinsame Notizen deaktivieren
- ☐ Private Chats deaktivieren
- ☐ Moderatoren vor Teilnehmendenzutritt fragen ⓘ
- ☐ Zugang via Link ⓘ

Maximale Teilnehmerzahl

Minuten Konferenzdauer (Max. Limit: 1440 Minuten) ⓘ

Willkommensnachricht ⓘ

Welcome to %%CONFNAME%%!

For help on using B

Aufzeichnung

- ☐ Sitzungen können aufgezeichnet werden. **beta**
- ☒ Aufzeichnungen für Teilnehmende sichtbar schalten ⓘ

Automatisches hochladen von Materialien ⓘ

Aktuell ausgewählter Ordner: Kein Ordner

Name
Allgemeiner Dateiordner
Aufgaben-Plugin
Martins
nils
Ordner tests

Abbildung 2.2: Meetings-Plugin: Dialog bei Raumerstellung

```

9      $xml = new \SimpleXMLElement($response);
10
11      if (!$xml instanceof \SimpleXMLElement) {
12          return false;
13      }
14
15      return $xml;
16  }

```

2.2.2 Etherpad Plugin

Etherpad ist ein kollaborativer echtzeit Editor, der das gleichzeitige Editieren eines Dokuments durch viele Nutzer*innen erlaubt (Quelle github Repository).

- (Uni-) eigene Installation
- Integration/Einbettung vom Etherpad per iframe
- Optik anhand eines Screenshots zeigen
- Versionshistorie
- Veröffentlichungsfunktion, sodass das Pad auch außerhalb von Stud.IP per generiertem Link zu erreichen ist
- beliebig viele Pads je Stud.IP Veranstaltung
- grob auf die Funktionsweise eingehen

2.2.3 Moodle-Connect Plugin

Mit Hilfe des Moodle-Connect Plugins lassen sich Veranstaltungen aus Stud.IP mit Veranstaltungen des Lernmanagementsystems Moodle verknüpfen (Quelle github).

Genauer gesagt lassen sich Stud.IP Veranstaltungen auf Moodle Veranstaltungen abbilden.

- Veranstaltung aus Stud.IP wird in Moodle übernommen
- Übernahme der Teilnehmer
- Screenshot(s)
- Button in Stud.IP
- Funktionsweise (Moodle + Stud.IP konfigurieren, API Nutzung, Tokengenerierung)
- mit Till sprechen!

2.2.4 LTI-Tools / Alija Plugin

Dieses Plugin bietet einen einfachen Mechanismus für eine Weiterleitung in eine externe Anwendung mit automatischer Anmeldung unter dem gleichen Account (Single-Sign-On) (Quelle svn Repo/Stud.IP Hilfe). Unterstützt werden dabei LTI-Tools (Version 1.x der Schnittstelle) sowie Stud.IP-Installationen (entweder über LTI oder das Alija-Protokoll). Das Plugin kann auf Veranstaltungs- oder Einrichtungsebene aktiviert werden. Ähnlich wie bei der "Freien Informationsseite" kann der Titel des Reiters sowie ein freier Text angegeben werden, der den Nutzern zur Erläuterung angezeigt wird.

Als Gegenstück zu diesem Plugin benötigt man auf der anderen Seite entweder ein mit dem LTI-Standard 1.x kompatibles LTI-Tool oder ein Stud.IP mit dem StudipAuthLTI oder StudipAuthAlija Auth-Plugin.

- Screenshot
- Einbettung per iframe oder Verlinkung zur jeweiligen externen Anwendung.
- evtl. mit Elmar sprechen.

2.3 Messenger

Was ist das?

2.3.1 Matrix

Konkretes Beispiel eines Messenger-Backends.

2.3.2 Matrix Schnittstellen

<https://matrix.org/docs/guides/client-server-api>

Kapitel 3

Anforderungsanalyse

Kernpunkt der Arbeit, auf dem der Hintergrund und die Implementation aufbauen. Empirischer Teil: Anforderungsanalyse mit Hilfe eines (Mini-) Fragebogens.

3.1 Umfrage zur Anforderungsanalyse

Was trifft am ehesten auf Sie zu?

20 responses

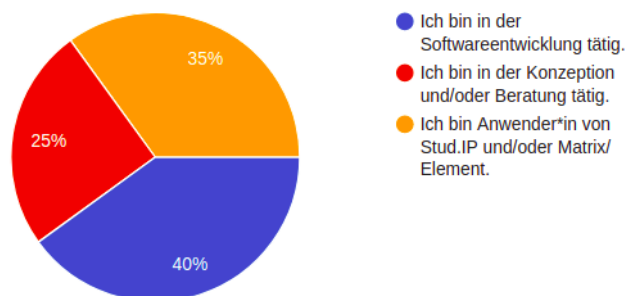


Abbildung 3.1: Umfrage: Teilnehmendenverteilung

Unterkapitel für jede herausgearbeitete Anforderung (funktional + nicht funktional). **Support-Hotline-Raum** Eine Support-Hotline-Funktion bei der man eine Gruppe von Personen erreichen kann, um ein Anliegen zu besprechen und der Chat beendet wird, nach dem das Anliegen erledigt ist bzw. nur der Kunde aus dem Chat entfernt wird.

3.2 Grundsatzentscheidung

Möglichkeit der beiden entwickelten Konzepte darstellen und die Entscheidung begründen. Konzept 1: vollständige Bridge (möglichst voller Funktionsumfang) / eigener Client - Aufwandsabschätzung, hohe Komplexität (element-web github Statistiken heranziehen) - hoher Pflegeaufwand - mit Element ist ein guter Client vorhanden, wieso alles nachbauen? Konzept 2: Lose Kopplung, Anbindung statt Integration - sinnvolle Diskussion, wie so eine Anbindung aussehen könnte - zwei klar getrennte Systeme (Transparenz) es wurden einige Anforderungen genannt, die nicht behandelt werden müssen, da diese bereits im Element-Client vorhanden sind Nachrichten senden und empfangen (Basisfunktionen) Übersichtliches UI Klare Übersicht (Nachrichten und Bedienung) undabhängig von OS/Browser/Screen size (Portabilität). Gute Usability, wenig Klicks zum Bilden von Arbeitsgruppen, Direktchats, usw. und ein modernes Erscheinungsbild. Matrix/Element in Blubber verlinken: Chat/Matrix/Element-Icon in Blubber anzeigen, mit dem man zu dem Uni-Matrix/Element-Raum gelangt. **Matrix als Hauptmessagingdienst** Weg von Blubber. Entscheidung der Dozenten, was genutzt wird Blubber abschalten, müssen die Betreiber entscheiden **Stummschalten von (Veranstaltungs-) Räumen** In Stud.IP ist es bisher nicht vorgesehen einzelne Blubber-Nachrichten als gelesen zu markieren.

3.3 Anforderungen

3.3.1 Übernahme von Veranstaltungsinformationen Veranstaltungen auf Räume abbilden + Übernahme der Teilnehmendenlisten

Kurse, Studengruppen, ... sollten (automatisch) auf Räume in Matrix abgebildet werden.

3.3.2 kein zusätzliches Login

Nur ein Login für beide Systeme/Dienste. Single Sign-On Stud.IP Nutzer sollten ohne zusätzliches Login in einen Matrix Raum gelangen können.

3.3.3 Benachrichtigungsfunktion

Synchronizität Zum einen sollten Matrix und Blubber zeitlich synchron sein. Zum anderen wäre es wünschenswert, wenn die in einem System als gelesen markierte Nachrichten auch im anderen System als gelesen markiert würden.

Man möchte in Matrix über relevante Ereignisse informiert werden. (-> später, was ist eigentlich relevant, Tabelle: Relevanzabschätzung für ereignisse, was ist relevant für alle? persönliche Präferenzen kann man mit dem hier entschiedenen Modell nicht berücksichtigen, deshalb sind folgende Konfiguration pro Veranstaltung sinnvoll? Welche Ereignisse sollen Erscheinen, oder

nicht? Diskussion, was spricht dafür, was spricht dagegen? Aus studentensicht evtl. verwirrend -> Argument gegen Konfig. Elmar oder Marcus -> Notifications, wenn z.B. eine Datei hochgeladen wird. Richtung Stud.IP -> Matrix, oder Matrix -> Stud.IP (nicht genannt, aber der Vollständigkeit halber dennoch genannt) braucht man eine Konfigurationsmöglichkeit dafür? wichtige Infos über die Kopplung hinweg. Informationsveranstaltung mit uniübergreifenden Infos, denkbar. (Eher ungeeignet, nach unten): unibezogene/veranstaltungsübergreifende Informationen, wie z.B. Rückmeldefristen, Unischließung, ... (dies wäre ein reiner Infochannel, in dem alle Studierenden sein müssten, keine Antwortmöglichkeit -> one-way Kommunikation -> eMail/Stud.IP Benachrichtigung wie bisher ist da vielleicht besser) Besser: veranstaltungsbezogene Informationen, z.B. eine Datei wurde hochgeladen", "Termin fällt aus"

3.3.4 Direktnachrichten

1-zu-1 Chats

Sprechstundenfunktion bei der man für eine kurze Zeit einen 1-zu-1 Chat mit einer Person führt und den Chat danach auch automatisch verlässt. Andere Personen sind in der Zeit in einer Warteschlange.

3.3.5 Transparenz

Verknüpfung zwischen Stud.IP und Matrix/Element deutlich machen. Welche Konsequenzen hat das eigene Handeln in Matrix, wenn ich in Stud.IP etwas mache? "Diese Aktion wird in Matrix erscheinen. Wie bekommen die Studierenden mit, dass es den Kanal gibt? Studierende sollten irgendwie informiert werden. Reiter: Messenger -> Infos + Kurzanleitung + evtl. Konfiguration Kann man deutlich machen, wie wäre es denn, wenn ich in Stud.IP sehen kann, wenn in Element etwas passiert. was heißt es eigentlich, wenn man zwei verschiedene Dienste hat? Was soll wo passieren? Hochladen einer Datei -> Dialog: erscheint auch im Matrix-Kanal, Schalter (Analyse Idee, schreiben, nicht umsetzen)? (Implementation: gibt Stud.IP)

3.3.6 Übernahme von Rollen, Rechten und Status

manuelle Nutzerverwaltung in Matrix sollte nicht genutzt werden, zwischen Doz und Tutoren nicht unterscheiden. Userstatus spielt in der Praxis kaum eine Rolle. Synchen der User. User = autoren Anbindung von Fremdsystemen mit einem anderen Rollen-Rechte-Konzept. Feingranulare Berechtigungen für Studierende ist nur eingeschränkt umsetzbar, da es in Matrix nur 3 Rechtstufen gibt. Denkbar wäre grob: Dozenten -> Moderator/Admin und Studierende -> Default. In Matrix lassen sich zusätzlich benutzerdefinierte Rechtstufen erstellen, eventuell lassen sich damit alle in Stud.IP verfügbaren Rechtstufen abbilden.

Der eigene Status sollte leicht einzustellen sein oder aus Stud.IP übernommen werden (sichtbar/nicht sichtbar) Man könnte in Matrix auch manuelle Rollen anlegen Datenschutzaspekt: Ich

möchte unsichtbar sein, in Matrix schwierig. Datenschutzrechtliche Begründung, warum steht der Name der Studenten in der Raumliste? beibehalten der Kommunikationskultur, keine anonymen Beiträge möglich in Matrix raumbezogene Einstellung: Teilnehmerliste ist sichtbar oder nicht in synchronisierten Räumen wir die Teilnehmerliste nicht angezeigt, kann Matrix das? Diskussion, was mache ich jetzt? 1. ich halte fest, es ist ein offenes Problem, Umsetzung trotz nicht datenschutzkonform 2. Leute, die sich unsichtbar geschaltet haben, werden nicht mitsynchronisiert, die bekommen evtl nichts mehr mit 3. Leute, die in min 1 VA unsichtbar sind -> mit Synonym sichtbar machen. - jeder Student muss von Stud.IP aus seine Zustimmung geben, das weitergeben des Nutzers an Matrix aktivieren (ist das sinnvoll?) gibt es Alternativen? ich sehe keinen guten Grund, das eine oder andere zu machen. -> Workshop mit Entwicklern und Didaktikern -> Ergebnisse -> Aus technischer Sicht gibt es keinen Grund, das eine oder das andere zu nehmen, Workshop hat ergeben: folgendes Ergebnis Beim Austragen oder Ändern der Rechtestufe in Matrix synchron zu halten. -> Cronjob einmal stündlich Lose Kopplung, nur einmal anlegen, was danach mit dem Raum passiert ist offen

3.3.7 mehrere Räume pro Kurs

Es sollten aus einer Stud.IP Veranstaltung mehrere Matrix-Räume erstellt werden können, die dann als Community zusammengefasst werden könnten. Mögliche Erweiterung (sinnvoll oder nicht?)

3.3.8 Kompatibilität von Nachrichten in Blubber und Matrix

Stud.IP (Blubber) sollte als regulärer Client agieren.

Dateien versenden Verlinken von Inhalten in Stud.IP Eventuell zu aufwändig.

Emoticons korrekt anzeigen

Nachrichten sollten editier- und löschar sein

3.3.9 Übernahme des Stud.IP Profilbildes

Personenbezogene Daten. Datenschutz. Sobald Matrixkopplung, dann auch Bild übertragen (im Reiter, s.o.), aktive Zustimmung verweist auf Profil->Einstellungen->Privatsphäre->Eigenes Bild Stud.IP: Wie darf mein Bild verwendet werden? "Darf auch extern angezeigt werden" gibt die API das überhaupt her, kann man auch weglassen, wenn es aus Zeitgründen nicht mehr klappt

3.3.10 Opensource**3.3.11 Stud.IP Plugin****3.3.12 keine zusätzliche API****3.3.13 Sicherheit****3.3.14 Robustheit, Ausfallsicherheit**

was passiert, wenn der Matrix Server für 3 Std. ausfaellt? Werden Messages gepuffert? Implementation direkt über eine Que laufen lassen, DB mit Ergebnissen, die noch so Matrix geschickt werden müssen + Cronjob 1x Std., gibt es noch Ereignisse, die nicht abgearbeitet wurden?

Kapitel 4

Implementation

Hier die Anforderungskapitelüberschriften kopieren und Stück für Stück erörtern. Sinnhaftigkeit, Umsetzung, Mögliche Umsetzung oder keine Umsetzung diskutieren.

4.0.1 Übernahme von Veranstaltungsinformationen Veranstaltungen auf Räume abbilden + Übernahme der Teilnehmendenlisten

Kurse, Studengruppen, ... sollten (automatisch) auf Räume in Matrix abgebildet werden.

4.0.2 kein zusätzliches Login

Nur ein Login für beide Systeme/Dienste. Single Sign-On Stud.IP Nutzer sollten ohne zusätzliches Login in einen Matrix Raum gelangen können.

4.0.3 Benachrichtigungsfunktion

Synchronizität Zum einen sollten Matrix und Blubber zeitlich synchron sein. Zum anderen wäre es wünschenswert, wenn die in einem System als gelesen markierte Nachrichten auch im anderen System als gelesen markiert würden.

Man möchte in Matrix über relevante Ereignisse informiert werden. (-> später, was ist eigentlich relevant, Tabelle: Relevanzabschätzung für ereignisse, was ist relevant für alle? persönliche Präferenzen kann man mit dem hier entschiedenen Modell nicht berücksichtigen, deshalb sind folgende Konfiguration pro Veranstaltung sinnvoll? Welche Ereignisse sollen Erscheinen, oder nicht? Diskussion, was spricht dafür, was spricht dagegegen? Aus studentensicht evtl. verwirrend -> Argument gegen Konfig. Elmar oder Marcus -> Notifications, wenn z.B. eine Datei hochgeladen wird. Richtung Stud.IP -> Matrix, oder Matrix -> Stud.IP (nicht genannt, aber der Vollständigkeit halber dennoch genannt) braucht man eine Konfigurationsmöglichkeit dafür? wichtige Infos über die Kopplung hinweg. Informationsveranstaltung mit uniübergreifenden

Infos, denkbar. (Eher ungeeignet, nach unten): unibezogene/veranstaltungsübergreifende Informationen, wie z.B. Rückmeldefristen, Unischließung, ... (dies wäre ein reiner Infochannel, in dem alle Studierenden sein müssten, keine Antwortmöglichkeit -> one-way Kommunikation -> eMail/Stud.IP Benachrichtigung wie bisher ist da vielleicht besser) Besser: veranstaltungsbezogene Informationen, z.B. "eine Datei wurde hochgeladen", "Termin fällt aus"

4.0.4 Direktnachrichten

1-zu-1 Chats

Sprechstundenfunktion bei der man für eine kurze Zeit einen 1-zu-1 Chat mit einer Person führt und den Chat danach auch automatisch verlässt. Andere Personen sind in der Zeit in einer Warteschlange.

4.0.5 Transparenz

Verknüpfung zwischen Stud.IP und Matrix/Element deutlich machen. Welche Konsequenzen hat das eigene Handeln in Matrix, wenn ich in Stud.IP etwas mache? "Diese Aktion wird in Matrix erscheinen. Wie bekommen die Studierenden mit, dass es den Kanal gibt? Studierende sollten irgendwie informiert werden. Reiter: Messenger -> Infos + Kurzanleitung + evtl. Konfiguration Kann man deutlich machen, wie wäre es denn, wenn ich in Stud.IP sehen kann, wenn in Element etwas passiert. was heißt es eigentlich, wenn man zwei verschiedene Dienste hat? Was soll wo passieren? Hochladen einer Datei -> Dialog: erscheint auch im Matrix-Kanal, Schalter (Analyse Idee, schreiben, nicht umsetzen)? (Implementation: gibt Stud.IP)

4.0.6 Übernahme von Rollen, Rechten und Status

manuelle Nutzerverwaltung in Matrix sollte nicht genutzt werden, zwischen Doz und Tutoren nicht unterscheiden. Userstatus spielt in der Praxis kaum eine Rolle. Synchen der User. User = autoren Anbindung von Fremdsystemen mit einem anderen Rollen-Rechte-Konzept. Feingranulare Berechtigungen für Studierende ist nur eingeschränkt umsetzbar, da es in Matrix nur 3 Rechtstufen gibt. Denkbar wäre grob: Dozenten -> Moderator/Admin und Studierende -> Default. In Matrix lassen sich zusätzlich benutzerdefinierte Rechtstufen erstellen, eventuell lassen sich damit alle in Stud.IP verfügbaren Rechtstufen abbilden.

Der eigene Status sollte leicht einzustellen sein oder aus Stud.IP übernommen werden (sichtbar/nicht sichtbar) Man könnte in Matrix auch manuelle Rollen anlegen Datenschutzaspekt: Ich möchte unsichtbar sein, in Matrix schwierig. Datenschutzrechtliche Begründung, warum steht der Name der Studenten in der Raumliste? beibehalten der Kommunikationskultur, keine anonymen Beiträge möglich in Matrix raumbezogene Einstellung: Teilnehmerliste ist sichtbar oder nicht in synchronisierten Räumen wir die Teilnehmerliste nicht angezeigt, kann Matrix das? Diskussion, was mache ich jetzt? 1. ich halte fest, es ist ein offenes Problem, Umsetzung trotz

nicht datenschutzkonform 2. leute, die sich unsichtbar geschaltet haben, werden nicht mitsynchronisiert, die bekommen evtl nichts mehr mit 3. leute, die in min 1 VA unsichtbar sind -> mit Synonym sichtbar machen. - jeder Student muss von Stud.IP aus seine Zustimmung geben, das weitergeben des Nutzers an Matrix aktivieren (ist das sinnvoll?) gibt es Alternativen? ich sehe keinen guten Grund, das eine oder andere zu machen. -> Workshop mit Entwicklern und Didaktikern -> Ergebnisse -> Aus technischer Sicht gibt es keinen Grund, das eine oder das andere zu nehmen, Workshop hat ergeben: folgendes Ergebnis Beim Austragen oder aendern der Rechtestufe in Matrix synchron zu halten. -> Cronjob einmal stündlich Lose Kopplung, nur einmal anlegen, was danach mit dem Raum passiert ist offen

4.0.7 mehrere Räume pro Kurs

Es sollten aus einer Stud.IP Veranstaltung mehrere Matrix-Räume erstellt werden können, die dann als Community zusammengefasst werden könnten. Mögliche Erweiterung (sinnvoll oder nicht?)

4.0.8 Kompatibilität von Nachrichten in Blubber und Matrix

Stud.IP (Blubber) sollte als regulärer Client agieren.

Dateien versenden Verlinken von Inhalten in Stud.IP Eventuell zu aufwändig.

Emoticons korrekt anzeigen

Nachrichten sollten editier- und löschar sein

4.0.9 Übernahme des Stud.IP Profilbildes

Personenbezogene Daten. Datenschutz. Sobald Matrixkopplung, dann auch Bild übertragen (im Reiter, s.o.), aktive Zustimmung verweist auf Profil->Einstellungen->Privatsphäre->Eigenes Bild Stud.IP: Wie darf mein Bild verwendet werden? "Darf auch extern angezeigt werden" gibt die API das überhaupt her, kann man auch weglassen, wenn es aus Zeitgründen nicht mehr klappt

4.0.10 Opensource**4.0.11 Stud.IP Plugin****4.0.12 keine zusätzliche API****4.0.13 Sicherheit****4.0.14 Robustheit, Ausfallsicherheit**

was passiert, wenn der Matrix Server für 3 Std. ausfaellt? Werden Messages gepuffert? Implementation direkt über eine Que laufen lassen, DB mit Ergebnissen, die noch so Matrix geschickt werden müssen + Cronjob 1x Std., gibt es noch Ereignisse, die nicht abgearbeitet wurden?

Kapitel 5

Deployment

Unter Umständen beschreiben, wie Matrix mit Hilfe von Ansible im Kontext der Praxisrelevanz aufgesetzt werden kann. (Wobei man auch einfach einen bereits bestehenden Matrix-Server voraussetzen könnte). Eventuell die Installation des Stud.IP Plugins und die nötige Konfiguration beschreiben.

Kapitel 6

Evaluation

Da vermutlich keine Zeit für eine empirische Evaluation mit einem IRL Test + Umfrage bleibt, werden hier technische Tests hinreichen müssen.

6.1 Technische Tests

6.1.1 Unit-Tests

Testbeschreibung

6.1.2 Code Coverage

6.2 Chancen

6.3 Probleme

Kapitel 7

Ausblick

Potentielle Nutzung in der Zukunft und Weiterentwicklung?

Beispieltext. Siehe auch [1, 3, 4]. URLs gehen auch: [2].

Literaturverzeichnis

- [1] BESL, P.; MCKAY, N.: A Method for Registration of 3-D Shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Februar, Nr. 2, S. 239–256
- [2] S. ASSMANN: *You(r) Study - Eigensinnig Studieren im 'digitalen Zeitalter'*. – <https://your-study.info/>
- [3] SCHULMEISTER, R.: *Lernplattformen für das virtuelle Lernen: Evaluation und Didaktik*. München : Oldenbourg Wissenschaftsverlag, 2005
- [4] THRUN, S.; FOX, D.; BURGARD, W.: A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Osnabrück, Juli 2021