



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

<b>1</b>	<b>Motivation</b>	<b>1</b>
<b>2</b>	<b>Hintergrund</b>	<b>3</b>
2.1	Lernmanagementsysteme . . . . .	3
2.1.1	Stud.IP . . . . .	3
2.1.2	Stud.IP Schnittstellen . . . . .	3
2.2	Anbindung von Fremdsystemen an Stud.IP . . . . .	3
2.2.1	Meetings Plugin . . . . .	3
2.2.2	Etherpad Plugin . . . . .	4
2.2.3	Moodle-Connect Plugin . . . . .	4
2.2.4	LTI-Tools / Alija Plugin . . . . .	4
2.3	Messenger . . . . .	5
2.3.1	Matrix . . . . .	5
2.3.2	Matrix Schnittstellen . . . . .	5
<b>3</b>	<b>Anforderungsanalyse</b>	<b>7</b>
3.1	Umfrage zur Anforderungsanalyse . . . . .	7
3.2	Grundsatzentscheidung . . . . .	7
3.3	Anforderungen . . . . .	7
3.3.1	Veranstaltungen auf Räume abbilden . . . . .	7
3.3.2	kein zusätzliches Login . . . . .	8
3.3.3	Übernahme der Teilnehmendenlisten . . . . .	8
3.3.4	Übersichtliches UI . . . . .	8
3.3.5	Benachrichtigungsfunktion . . . . .	8
3.3.6	Nachrichten senden und empfangen (Basisfunktionen) . . . . .	8
3.3.7	Direktnachrichten . . . . .	8
3.3.8	Transparenz . . . . .	8
3.3.9	Übernahme von Rollen, Rechten und Status . . . . .	8
3.3.10	mehrere Räume pro Kurs . . . . .	9
3.3.11	Synchronizität . . . . .	9
3.3.12	Matrix als Hauptmessagingdienst . . . . .	9
3.3.13	Kompatibilität von Nachrichten in Blubber und Matrix . . . . .	9
3.3.14	Stummschalten von (Veranstaltungs-) Räumen . . . . .	9

3.3.15	Übernahme des Stud.IP Profilbildes . . . . .	9
3.3.16	Support-Hotline-Raum . . . . .	9
3.3.17	Opensource . . . . .	10
3.3.18	Stud.IP Plugin . . . . .	10
3.3.19	keine zusätzliche API . . . . .	10
3.3.20	Sicherheit . . . . .	10
<b>4</b>	<b>Implementation</b>	<b>11</b>
<b>5</b>	<b>Deployment</b>	<b>13</b>
<b>6</b>	<b>Evaluation</b>	<b>15</b>
6.1	Technische Tests . . . . .	15
6.1.1	Unit-Tests . . . . .	15
6.1.2	Code Coverage . . . . .	15
6.2	Chancen . . . . .	15
6.3	Probleme . . . . .	15
<b>7</b>	<b>Ausblick</b>	<b>17</b>

# Kapitel 1

## Motivation

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

Was ist das?

#### 2.1.1 Stud.IP

Konkretes Beispiel eines LMS

#### 2.1.2 Stud.IP Schnittstellen

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

### 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

Unterkapitel für jede herausgearbeitete Anforderung (funktional + nicht funktional).

### 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)

### 3.3 Anforderungen

#### 3.3.1 Veranstaltungen auf Räume abbilden

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

### 3.3.2 kein zusätzliches Login

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

### 3.3.3 Übernahme der Teilnehmendenlisten

### 3.3.4 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.

### 3.3.5 Benachrichtigungsfunktion

Eher ungeeignet: 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"

### 3.3.6 Nachrichten senden und empfangen (Basisfunktionen)

### 3.3.7 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.

### 3.3.8 Transparenz

Verknüpfung zwischen Stud.IP und Matrix/Element deutlich machen.

### 3.3.9 Übernahme von Rollen, Rechten und Status

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 Rechtestufen abbilden.

Der eigene Status sollte leicht einzustellen sein oder aus Stud.IP übernommen werden (sichtbar/nicht sichtbar)

### 3.3.10 mehrere Räume pro Kurs

Es sollten aus einer Stud.IP Veranstaltung mehrere Matrix-Räume erstellt werden können, die dann als Community zusammengefasst werden könnten.

### 3.3.11 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.

### 3.3.12 Matrix als Hauptmessagingdienst

Weg von Blubber.

### 3.3.13 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**

### 3.3.14 Stummschalten von (Veranstaltungs-) Räumen

In Stud.IP ist es bisher nicht vorgesehen einzelne Blubber-Nachrichten als gelesen zu markieren.

### 3.3.15 Übernahme des Stud.IP Profilbildes

### 3.3.16 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.

**3.3.17**    **Opensource**

**3.3.18**    **Stud.IP Plugin**

**3.3.19**    **keine zusätzliche API**

**3.3.20**    **Sicherheit**



## Kapitel 4

# Implementation

Hier die Anforderungskapitelüberschriften kopieren und Stück für Stück erörtern.



## 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