EMFI-Lastenheft

FabInfra-Team

March 2020

Inhaltsverzeichnis

1 Anforderungsliste

Id	Stichwort	Anforderung	Kategorie	Quelle / Ansprechpartner
1	RFID-Karten	Muss MiFare DESFire Karten lesen können.	Sensor	EMFI-Chat
2	Maschinen sicher schalten	Für jede Maschine muss ein Schaltkonzept hinterlegt werden können, welche auf die Eingabe von Sensoren reagiert, diese überprüft und mit den jeweiligen Aktoren agiert.	Core / GUI	EMFI-Chat
3	Einfache "Maschinen" abbilden	Es soll z.B. möglich sein, auch Tische / Schraubenzieher / ... abzubilden, die keine Einweisung und / oder kein Berechtigungssystem benötigen.	Core / GUI	Metapad (TMu)
4	Override	Es soll z.B. möglich sein, mit verschiedenen "Master-Karten" Maschinen direkt am Node ein- und auszuschalten. Auch wenn keine Netzwerkverbindung besteht.	Node	Erfahrung Card2Go (TMu)
5	Übergabe	Es soll möglich sein, innerhalb einer *kurzen* Karenzzeit (z.B. 15 s) eine Maschine von einem Nutzer an einen neuen und dazu berechtigten Nutzer zu übergeben. Z.B. für Einweisungen ($\geq 5 \rightarrow 1$) oder unkomplizierte Übergabe mit Übertragung der Verantwortung für Putzen und weiteres ($\geq 3 \rightarrow \geq 3$)	Core / Node	Erfahrung Card2Go (TMu)
6	Abnahme	Eine abnahmepflichtige Maschine die genutzt wurde, muss erst von einem zur Abnahme Berechtigten (z.B. Stufe ≥ 4) abgenommen werden, bevor sie wieder von einem Nutzer der nicht zur Abnahme berechtigt ist (z.B. Stufe ≤ 4) in Betrieb gesetzt werden kann.	Core / Node	Erfahrung Card2Go (TMu)

7	Qualifikationsmatrix	Qualifikation der Kombination Nutzer-Maschine muss in z.B. 5[1] Stufen abgebildet werden können und im Schaltkonzept berücksichtigt werden.	Core / GUI	MetaPad (TMu)
8	Nutzerhinweise ausgeben	Nutzer sollen z.B. auf abgelaufene Einweisungen hingewiesen werden und verständliche Fehlermeldungen erhalten, falls eine Aktion abgelehnt wird.	Core / GUI / Node	Metapad (TMu)
9	Module	Zur Laufzeit ladbare Module mit denen zumindest neue Sensoren & Aktoren hinzugefügt werden können	Core / Module	EMFI-Chat
10	Maschinen Monitoring	Überwachung von Aktivität und Eigenschaften(Stromverbrauch) von Maschinen	GUI	EMFI-Chat
11	Reservierung von Maschinen	Maschinen sollen *kurzfristig*[2] von Usern reserviert werden können und sind dann entsprechend für andere User gesperrt	Core	EMFI-Chat (TMu)
12	Blockieren von Maschinen	Maschinen können z.B. für Wartung oder geplante Veranstaltungen von Administratoren blockiert werden und sind dann für alle oder mit Ausnahme bestimmter User bis zur Aufhebung der Blockade gesperrt	Core / Node	EMFI-Chat
13	Hinweise zur Relation Zeit - Maschine abbilden	Es soll z.B. möglich sein, für eine Maschine einen Hinweis zu hinterlegen, der angezeigt wird wenn absehbar ist dass eine geplante Nutzung (3D-Druckauftrag) länger als bis ${Zeit} dauern wird.	Core / GUI / Node	Metapad (TMu)
14	Nutzungsende ankündigen	Vor Ende einer geplanten Nutzungszeit soll in einem bestimmbaren Zeitraum das Ende der Nutzungszeit angekündigt werden (rückwärtscounter).	Core / GUI / Node	EMFI-Chat (TMu)

[1]Das Stufenmodell soll mit sensible defaults voreingestellt, jedoch für die individuelle Berechtigungssteuerung auch durch die Administratoren in den Nutzergruppen auf die jeweiligen Bedürfnisse anpassbar sein.

[2]Die kurzfristige Reservierung soll Nutzern ermöglichen, eine freie Maschine z.B. während der Fahrt zum Labor nicht zu verlieren. Es soll NICHT möglich sein, "ein Handtuch auf die Maschine zu legen" und potentiell nutzbare Maschinen aus der freien Nutzung zu nehmen und (aus welchen ehrbaren Gründen auch immer) für andere zu blockieren.

15	Öffnungszeiten festlegen	Das Schließsystem und die Maschinen sollen Nutzungszeiten haben, für einzelne Maschinen (z.B. 3D-Drucker) sollen Ausnahmen gesetzt werden können	Core / GUI	EMFI-Chat
16	Gruppieren von Maschinen	Maschinen sollen Gruppen hinzugefügt werden können, dabei kann eine Maschine in mehreren Gruppen sein	Core/GUI	TheJoKlLa
17	App orientierter Client	Es soll unabhängig alle von uns entwickelten Anwendungen in den Client eingebunden werden können	GUI	TheJoKlLa
18	Plattform unabhängiger Client	Der Client soll Plattform unabhängig sein. Primäre Windows, Android, iOS	GUI	EMFI-Chat
19	Statusanzeige	Statusanzeige soll sowohl spezifisch für eine einzelne Maschine als auch für eine Gruppe von Maschinen möglich sein. In der Statusanzeige der Maschine soll schnell erkennbar sein, wer für die Maschine verantwortlich ist - bis die Maschine durch einen Benutzer mit der Berechtigung zur Abnahme der Maschine wieder freigegeben ist.	Core / GUI / Node	EMFI-Chat (TMu)
20	Anzeige von Verantwortlichen	In der Statusanzeige von Maschinen soll schnell erkennbar sein, wer aktuell für eine Maschine verantwortlich ist. Bis eine reinigungspflichtige Maschine durch einen Benutzer mit der Berechtigung zur Abnahme der Maschine wieder freigegeben ist soll sie gesperrt bleiben.	Core / Node	Erfahrungen Card2Go (TMu)
21	Nutzermanagement	Nutzer intern & extern möglich	Core	EMFI-Chat
22	Authentifizierung	Authentifizerung von Benutzer passiert über SASL um erweiterbar zu bleiben	Core	Erfahrung mit Backends.

23	Schnittstelle zu externer Nutzerdatenbank	Wenn eine externe Nutzerdatenbank zur Verfügung (AD / LDAP / ...) steht, soll diese genutzt werden. Im Core werden dann nur zusätzlich notwendige Daten (Einweisungen / Berechtigungen / ...) über eine NutzerId verknüpft verwaltet. Diese Daten sollen nach einem DGSVO-konformen Löschkonzept gelöscht werden, wenn sie nicht mehr benötigt werden.	Core / GUI	Erfahrung mit Backends. (TMu)
24	Interne Nutzerdatenbank	Wenn keine externe Nutzerdatenbank zur Verfügung steht, soll eine interne Verwaltung der Nutzer über eine einfache Datenbank möglich sein.	Core / GUI	Erfahrung mit Backends. (TMu)
25	Audit	Audit-Eventstream als strukturiertes Log über gut dokumentierte Schnittstelle an externe Programme	Core / Audit	EMFI-Chat (deq)
26	Abrechnung	Abrechnung passiert in externer Software (s. Non-Features)	Abrechnung	EMFI-Chat
27	Prepaid	Es soll möglich sein, auf geeignete Chip-Karten ein Guthaben aufzubuchen, das bei Nutzung von Geräten aufgebraucht wird. Am Ende der gebuchten Nutzung soll die Maschine sinnvoll ausgeschaltet werden.	Core / GUI	Kundenwunsch (tmu)

1.1 Non-Features

Bestimmte Sachen sind im Zusammenhang mit dem Gesamtsystem sinnvoll, sind aber unserer Meinung nach im Kern nicht richtig aufgehoben. Diese Dinge sollten besser extern[3] gelöst werden.

Stichwort	Anforderung	Alternative
User Monitoring	Nachvollziehen wann welcher Nutzer welche Maschine genutzt hat.	Wird über externe Loganalyse-Werkzeuge gelöst.
Abrechnung	Abrechnung von benutzen Maschinen wie z.b. Laser-Cutter.	Software wie Odoo ist für sowas wesentlich besser geeignet.

[3] Non-Features können innerhalb von FabAccess, jedoch nicht im Kern sondern in externen Referenz-Modulen - oder auch in vollständig getrennten Projekten implementiert werden.

2 Ausgangssituation

2.1 Wie kam es zur Projektidee?

Sowohl im FabUniverse-Netzwerk[4] als auch im Verbund Offener Werkstätten wurde Ende 2019 deutlich erkennbar, dass viele Akteure an ähnlichen Zugangssystemen arbeiten. Viele sehr spezifische Lösungen wurden sichtbar, aber keine, die Open Source sowie ausreichend generisch gebaut ist, um den unterschiedlichen Bedürfnissen vieler Werkstätten zu entsprechen. Um die Aktivitäten an der Stelle zu bündeln, wurde gemeinsam durch die beiden Netzwerke das Projekt FabAccess bzw. die Gruppierung FabInfra ins Leben gerufen.

2.2 Welches Problem ist aufgetreten?

Zugangskontrolle und Management in Fablabs ist unzureichend automatisiert. Dadurch entsteht zum einen viel (unnötiger) Aufwand auf Seite der Betreiber von offenen Werkstätten, zum anderen könnten durch ein zuverlässiges System an der Stelle Gefahrensituationen verhindert und minimiert werden.

2.3 Wie wurde damit in der Vergangenheit umgegangen?

Es wurde eine Reihe von Fablab-/Makerspace-spezifischen Programmen entwickelt, die alle für das Fablab speziell ausreichen aber nicht für alle Labs ausreichen. Teilweise wurden auch für spezifische Probleme geeignete Insellösungen eingesetzt, die jedoch auch nur einzelne Probleme lösten (z.B. Nuki als Tür-Öffner, der aber keine Maschinen absichern kann).

2.4 Wieso besteht Handlungsbedarf?

Das Finanzierungsmodell von kommerziellen Alternativen wie Fabman.io ist für viele Makerspaces nicht erschwinglich. Es soll in den offenen Werkstätten den Betreibern Arbeit abgenommen werden, damit sie ihre Ressourcen auf wichtigere Dinge konzentrieren können.

2.5 In welche längerfristige Strategie soll das Projekt eingebunden werden?

Im Verbund offener Werkstätten organisierte Werkstätten; Netzwerk FabUniverse und damit deutschlandweit in Hochschul-FabLabs/-Laboren; Spezifisch zunächst im FVM-Labor der Beuth Hochschule für Technik Berlin, ...

[4] Im FabUniverse-Netzwerk sind Akteure der FabLabs und MakerSpaces innerhalb von Hochschulen aus dem deutschsprachigen Raum vertreten.

3 Warum überhaupt FabAccess?

4 Zielsetzung gemäß der SMART-Kriterien

Specific – target a specific area for improvement. Measurable – quantify or at least suggest an indicator of progress. Assignable – specify who will do it. Realistic – state what results can realistically be achieved, given available resources. Time-related – specify when the result(s) can be achieved.

5 Produkteinsatz

Unter welchen Rahmenbedingungen soll das Produkt zum Einsatz kommen? (z. B. Temperatur, klimatische Bedingungen, Druck, Umfeld, etc.) Von wem soll das Produkt bedient werden? Was soll das Produkt unter welchen Rahmenbedingungen leisten?

6 Funktionale Anforderungen

Welche Funktionen sollen vorhanden sein? Was soll das Produkt können oder leisten?

Nutzung von Maschinen soll nur denjenigen möglich sein die eine Unterweisung oder vergleichbare Praxiserfahrung mit Maschinen dieser Art besitzen. Maschinen können ohne Interaktion durch die Werkstatbetreibereingeschaltet werden wenn eine zugriffsberechtigte (befähigte und eingewiesene) Person das wünscht.

Welche konkreten Funktionen muss das Produkt bieten?

6.1 Verbindung von Sensoren und Aktoren

Die Verbindung dieser passiert rein im Kern und ist N:M, Sensoren können mehrere Aktoren beschalten, ein Aktor kann von mehreren Sensoren beschaltet werden.

6.2 Authentifizierung von Benutzer

FabAccess unterstützt SASL.

6.3 Rollen-basiertes Berechtigungssystem (RBAC)

Zur Zugriffskontrolle wird RBAC genutzt. Es gelten die üblichen Regeln:

- Ein Mensch ist Mitglied von N Rollen ($N \geq 0$).
- Eine Rolle stellt eine bestimme Tätigkeit oder Ausbildung dar.
- Eine Berechtigung erlaubt Zugriff auf eine Ressource oder Ausführung einer Aktion.
- Eine Rolle gibt eine oder mehr Berechtigungen.

- Rollen sind Teil einer zyklenlosen Hierarchie, $x \geq y$ bedeutet die Rolle x enthält alle Berechtigungen der Rolle y.

Ein Gruppenidentifier hat sowohl einen lokalen als auch einen remote Teil die spezifieren wie die Gruppe heisst und auf welcher Instanz sie definiert wurde

Ein Berechtigungsstufensystem wie:

-1 explizit gesperrt

0 darf bei der Arbeit zusehen und lernen (nicht einschalten / aktivieren)

1 darf unter Aufsicht und Anleitung arbeiten (Maschine muss von $\geq$4/5 aktiviert und übergeben worden sein)

2 darf selbständig arbeiten (aber z.B. Maschinen nicht übergeben)

3 kann einrichten und einstellen (ist an Feierabend gebunden, kann Maschinen übergeben)

4 kann warten / instandsetzen (z.B. auch nach Feierabend einschalten) und reinigungspflichtige Maschinen abnehmen.

5 kann schulen & einweisen (ist /root, kann alles.)

Kann dann dargestellt werden als: Pro Maschine sind die Berechtigungslevel 0-5 jeweils eine Rolle, die aus der unterliegenden erbt. Jede Rolle definiert die für das Level spezifisch notwendigen Berechtigungen. Ein Mitglied mit Ausweis ist an den meisten Maschinen automatisch Teil aller Rollen-0[5]. Berechtigung -1 enspricht einem Nutzer der in keiner der 6 Rollen eingetragen ist.

7 Nichtfunktionale Anforderungen

7.1 Erweiterbarkeit

Nach Anforderung 1 kann Kommunikationscode um mit spezifischen Sensoren und Aktoren zu interagieren in Form von zur Laufzeit ladbaren Modulen hinzugefügt werden. Diese Module sollten zumindest in den Programmiersprachen Python und Lua schreibbar sein und an gut dokumentierte Schnittstellen andocken können.

8 Lieferumfang

Nach Anforderung 1 und 1 bietet FabAccess keine eingebaute Möglichkeit zur Abrechnung von Maschinenstunden oder zur minutösen Nachverfolgung der Belegung von Maschinen im Nachhinein.

[5] Ausnahme gefährliche Maschinen wo auch beim zusehen Gefahr besteht (Lasercutter?)

9 Phasenplanung und Meilensteine des Projektes

10 Offene Punkte, die noch zu klären sind

Was konnte noch nicht geklärt werden? Wer kümmert sich bis wann um die Klärung? Wie werden Entscheidungen im Projekt herbei geführt? Wer darf sie einbringen? Wer muss zustimmen? Wer hat ggf. ein Vetorecht?

11 Abnahmekriterien und Qualitätsanforderungen

Anhand welcher Kriterien wird die Projektleitung zum Projektende und an den Meilensteinen entlastet? Welchen Gremien gegenüber soll die Projektleitung berichten? Welche Qualitätsanforderungen werden an das Projekt gestellt? Welches QM-System und welche Unterlegen daraus gelten für das Projekt?

12 Threat Model

Angreifer sind meistens gelangweilte Studierende und der Schaden vergleichsweise klein.

13 Zukünftige Erweiterungen

13.0.1 Version 2.0

- Federation
- Reservierung anstatt nur in situ Nutzung, z.b. für Konferenzräume ganz sinnvoll.
-

Id	Stichwort	Anforderung	Kategorie	Quelle
28	SSO / Weitergabe Möglichkeit von Usern	Zu FabRega und FabLock Systemen. Vielleicht direkt Keycloak nutzten	Core	TheJoKlLa
29	Reservierung von Maschinen	Maschinen sollen von Usern reserviert werden können und sind dann entsprechen für andere User gesperrt	Core	EMFI-Chat / Beuth