HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

dxb
gmbh

RFID Webauthentifizierung

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Frühjahrssemester 2017

Autoren: Andreas Eder, Pascal Kistler
Betreuer: Ivan Bütler
Projektpartner: dxb GmbH

Abstract

Ausgangslage
Die Firma dxb GmbH entwickelt und vertreibt eine webbasierte Managementlösung (Gordo), mit welcher die Arbeiten bei Events oder die Abläufe in einer Reparaturwerkstatt koordiniert und optimiert werden können. Die Kunden melden sich mit Benutzername und Passwort mehrmals täglich bei Gordo an. Es ist der Wunsch entstanden, den Anmeldeprozess zu vereinfachen und dem Benutzer einen Login mittels NFC Karte zu ermöglichen. Diese Arbeit hat zum Ziel, eine RFID Webauthentifizierung für Gordo zu implementieren.

Vorgehen / Technologie
Der Prototyp besteht aus einem Internet- und NFC fähigen Lesegerät (Raspberry Pi), NFC Karten des Typ NTAG 213/216 und einem neuen Modul zur NFC Authentifizierung an Gordo. Darüber hinaus verwaltet Gordo die NFC Reader mit einem neu entwickelten Provisionierungsservice.
Die neuen Komponenten im Überblick:

- Provisionierungsservice
- NFC Karten
- NFC Reader
- Authentifizierungsmodul für Gordo

Der Provisionierungsservice basiert auf einer komplett neu entwickelten PHP und MySQL Anwendung. Diese verwaltet die Zuordnung der Reader zu einem Kundenprojekt.
Im Rahmen des Projektes wurden zudem geeignete NFC Karten evaluiert, welche mit einem Access Passwort geschützt werden können. Es wurde darauf geachtet, dass der Verlust oder Diebstahl einer NFC Karte das Gesamtsystem nicht kompromittiert.
Die Software des NFC Readers wurde auf der Basis eines Raspberry Pi und NFC Hardwaremodul von NXP entwickelt. Die Reader verwenden individuelle RSA Schlüssel für die abgesicherte Kommunikation mit Gordo. Der Prozess der Erstinstallation der Reader, deren Nutzung, Ersatz bei Verlust oder Diebstahl wird durch den Prototypen abgedeckt. Das neue NFC Authentifizierungsmodul wurde ins Gordo integriert.

Ergebnis
Im Rahmen der SA wurde ein funktionierender Prototyp entwickelt, welcher die meisten Ziele der Aufgabenstellung erfüllt. Im Testaufbau kann sich der Benutzer mit seiner persönlichen NFC Karte und einem vorkonfigurierten NFC Reader an Gordo anmelden. Hierbei wird kein Passwort oder Benutzername mehr benötigt. Die Management Prozesse für das Provisionieren der NFC Reader und Karten wird durch den Prototypen abgedeckt. Bei Verlust einer NFC Karte oder eines Readers können diese einfach ersetzt werden. Der Login mittels NFC dauert durchschnittlich 4-5 Sekunden. Ein visuelles oder akustisches Feedback des Readers für den Benutzer über den aktuellen Zustand des Anmeldeprozesses konnte nicht mehr realisiert werden.

Eigenständigkeitserklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- das ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum:
Raperswil-Jona, 2. Juni 2017

Name, Unterschrift

Andreas Eder

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- das ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum:
Raperswil-Jona, 2. Juni 2017

Name, Unterschrift

Pascal Kistler

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FS2017

SA

März 20., 2017

Document Name:	Aufgabenstellung_SA_RFID_Web_Auth_Korrigenda.docx
Version:	v1.0
Author	Ivan Buetler
Date of Delivery:	März 20., 2017
Classification:	SA

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

Table of Content

HSR Hochschule für Technik
Postfach 1475
CH-8640 Rapperswil
www.hsr.ch

1 Ausgangslage

1.1 Einleitung

Die Firma dxb aus Weinfelden möchte im Rahmen der HSR Studienarbeiten FS2017 mit Pascal Kistler und Andreas Eder einen Prototyp für die Umsetzung der RFID basierten Web Authentisierung entwickeln.

Die grundsätzlichen Gedanken und Lösungsvarianten wurden zu Beginn der SA in gemeinsamen Workshop entwickelt. Dieses Dokument beschreibt das gemeinsam vereinbarte Zielsystem.

1.2 Business Case

Grundsätzlich soll es möglich sein, dass bei einem Event wie OpenAir St. Gallen, die Mitglieder der Organisation sich mittels RFID Karte an der Open Air Staff & Event-Webapplikation authentisieren, ohne dabei das Passwort eingeben zu müssen. Zu diesem Zweck erhalten alle Staff Mitglieder eine RFID Karte. An verschiedenen Stellen eines Events stehen netzwerkfähige RFID Reader mit dazugehörigen Web Terminal bereit.

1.3 Auftrag

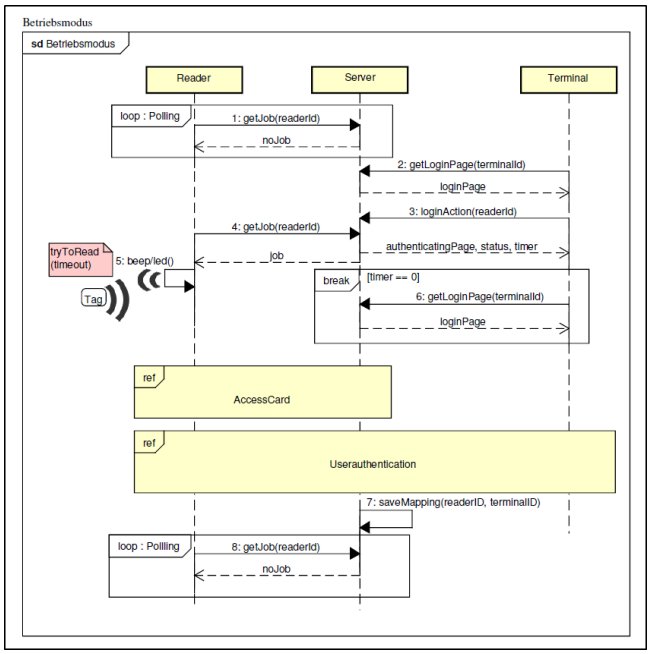
Die existierende Business Applikation der dxb soll um eine RFID Authentisierung erweitert werden. Das Projekt umfasst folgende Komponenten

- SW Erweiterung in Business Applikation
  - Verwaltung RFID Karten
  - Zuordnung RFID Karten zu Benutzer und Projekten
  - Verwaltung RFID Reader
- Rasperry basierter RFID Reader
  - Internet (Wifi) fähiges Gerät
  - Verbindet sich mit der Business Applikation
  - Keine physische (USB, Serial) Verbindung zu einem PC
- Web Terminal
  - PC mit Webbrowser
  - Smartphone
- RFID Karte
  - n-tag213 respektive n-tag216 Karten (Vorgabe dxb)

1.4 Grundsätzlicher Use-Case

Das untenstehende Diagramm beschreibt den Ablauf der Web Authentifizierung. Das Enrollment des Readers, Karte und Web-Terminal (PC mit Browser) sind im Ablauf nicht dargestellt. Deshalb gelten folgende Rahmenbedingungen

- Das Web Terminal ist der Business Applikation bekannt und dem Projekt zugewiesen
- Der RFID Reader ist der Business Applikation bekannt und dem Projekt zugewiesen
- Die RFID Karte ist einem Benutzer und dem Projekt zugewiesen



Alle Reader pollen die Business Anwendung und fragen dort nach einem Job.

Wenn der Benutzer an einem Web Terminal einloggen will, klickt er in der Web Anwendung auf "LOGIN". Dies löst einen Job in der Business Anwendung aus.

Beim nächsten Polling holt der dem Web Terminal zugeordnete RFID Reader den Job und signalisiert dem User über ein akkustisches oder visuelles Ereignis, dass nun die Karte an den Reader gehalten werden soll.

Durch den Kontakt der Karte mit dem Reader wird der User identifiziert (AccessCard) und der Business Anwendung mitgeteilt. Der User wird dadurch in der Business Anwendung angemeldet (Userauthentication). Nach Abschluss des Login signalisiert der Reader über ein akkustisches oder visuelles Signal den Erfolg/Misserfolg der Anmeldung. Nach Abschluss der Authentisierung geht der Reader wieder in die Polling Phase zurück.

HSR Hochschule für Technik
Postfach 1475
CH-8640 Rapperswil
www.hsr.ch

1.5 Vereinbarte Rahmenbedingungen

Die folgenden Rahmenbedingungen wurden in den ersten Workshops mit dem Industriepartner und den HSR Studenten vereinbart.

Was	Anforderungen	Details
RFID Karte	Die Identität (respektive eine eindeutige Zahl die auf den User referenziert) eines Users ist im geheimen Teil der n-tag Karte gespeichert. Der Zugriff auf die Identität erfordert entsprechend ein Passwort (Access Passwort). Jede Karte hat ein anderes AccessPasswort. Das Passwort zur Karte wird beim Server hinterlegt. Der Server kennt die Zuordnung zwischen der Zahl und dem User. Es sollen RFID Karten vom Typ n-tag213 respektive n-tag216 eingesetzt werden.	Die Personalisierung der RFID Karte und Zuordnung zu einem Projekt in der Business Applikation erfolgt manuell über eine neue Funktionalität in der Business Anwendung.
RFID Reader	Der RFID Reader soll auf einem Rasperry Pi 3 betrieben werden. Die Kommunikation zwischen RFID Reader und Web Anwendung erfolgt über SSL/TLS. Jeder RFID Reader ist eindeutig identifizierbar und einem Projekt in der Business Anwendung zugeordnet. Jeder Reader hat seine eigene Konfiguration und dazugehörige Keys (Verschlüsselung). Der Provisionierungsserver der Lösung verwaltet die Projektzuordnung. Die Konfiguration wird vom neuen Auth Modul der Business Applikation verwaltet.	Die RFID Reader werden manuell zu einem Projekt zugeordnet. Auf ein auto-learning bei der Projektzuordnung wird verzichtet.

Was	Anforderungen	Details
Web Terminal	Ein PC mit Browser und Verbindung zur Business Applikation wird als Web Terminal bezeichnet. Jedes Web Terminal wird nach dem ersten Besuch der Business Anwendung mittels einem Browser Fingerprint identifiziert. Dieser Fingerprint dient jedoch nur zu Statistik Zwecken und spielt keine Rolle beim Enrollment (Anlegung, Sperrung) von Web Terminals.	Es können beliebige Web Terminals eingesetzt werden. Damit sind auch Mobile User Logins möglich.
Business Applikation	Ein zusätzliches Auth Modul in der dxb Business Applikation soll um die Bereitstellung des RFID Login erweitert werden.	Der Industriepartner stellt hierfür eine Testumgebung mit Source Code bereit.
Provisionierungs Server	Für die Verwaltung der Reader ist der Einsatz eines Provisionierungsserver vorgesehen. Dieser verwaltet die Projektzuordnung.	Der Industriepartner stellt hierfür eine Testumgebung (VirtualMachine mit Public IP) bereit.
Verlust RFID Reader	Der Verlust eines RFID Reader ist als Prozess in der Lösung abgedeckt.	Ein gestohlener RFID Reader darf das Gesamtsystem Security mässig nicht gefährden
Verlust RFID Karte	Der Verlust einer RFID Karte durch ein Staff Member ist im Prozess der Lösung abgedeckt	Der User kann eine neue Karte bekommen und die alte wird ungültig

HSR Hochschule für Technik
Postfach 1475
CH-8640 Rapperswil
www.hsr.ch

1.6 Erwartetes Ergebnis

Es wird ein funktionierender Prototyp erwartet, welcher die Kernfunktionen umfasst. Der Prototyp wird auf der vom Industriepartner bereitgestellten Infrastruktur entwickelt (Business Applikation, Provisionierungsserver).

Inhaltsverzeichnis

1 Management Summary

1.1 Ausgangslage

Der Industriepartner dxb gmbh entwickelt und vertreibt eine webbasierte Managementlösung zur Verwaltung diverser Geschäftsprozesse. Diese Software wird im Weiteren als Gordo bezeichnet. Gordo wird von diversen Organisationen aus unterschiedlichen Branchen verwendet. Dazu gehören beispielsweise Event-Organisationen, welche damit die Auf- und Abbauarbeiten koordinieren oder auch Reparaturwerkstätten, die mit Hilfe von Gordo die Abläufe und Aufträge verwalten können. Gerade bei solchen Kunden wird Gordo auf unterschiedlichen, zentral platzierten Geräten verwendet. Diese Geräte besitzen eine hohe Frequentierung an verschiedenen Benutzern, welche sich mehrmals täglich an- und abmelden müssen, was den Arbeitsablauf in Effizienz, Geschwindigkeit und Zeit negativ beeinflusst. Diese Ineffizienz brachte den Wunsch hervor, den Anmeldeprozess zu optimieren und zu vereinfachen ohne dabei die Sicherheit des Gesamtsystems zu reduzieren.
Das Ziel des RFID Webauthentifizierungsmoduls ist, dass sich ein Mitarbeiter an einem Notebook, Computer oder Smartphone via einem netzwerkfähigen NFC Lesegerät, auch Reader genannt, mit einer personalisierten Karte kontaktlos anmelden kann.

1.2 Vorgehen/Technologien

In mehreren Workshops mit der Firma dxb gmbh wurden die Anforderungen an das RFID Webauthentifizierungssystem erarbeitet. Dabei wurde der Anmeldeprozess als der zentrale Business Prozesse für das neue System identifiziert, dabei soll sich ein Benutzer, ohne Benutzernamen und Passwort eingeben zu müssen, nur mit seiner NFC Karte am System authentifizieren können.
Die Hardware für den Reader wurde von der Firma dxb gmbh vorgegeben, aber die genaue NFC Techonolgie für die Karten wurde erst während dem Projekt evaluiert. Das neue Authentifizierungssystem wurde auf einer Testumgebung des Auftraggebers entwickelt und getestet.

Das neue Authentifizierungssystem benötigt die folgenden Komponenten:

- NFC Karte (NTAG 213 / NTAG 216)
- internetfähiger Reader (Raspberry Pi 3 mit NFC Hardwaremodul von NXP)
- Authentifizierungsmodul (Erweiterung des Gordo-Systems)

- Provisioning Service (Service für zentrale Verwaltung der Reader)

Alle Komponenten wurden mit Technologien der Netzwerksicherheit und Verschlüsselungstechnik gesichert. Weitere Sicherheitsmechanismen decken auch die Fälle von entwendeten Readern oder Karten ab.

1.3 Ergebnisse

Das Ziel der Ausgangslage ein Authenifizierungssystem zu entwickeln, dass auf der NFC Technologie basiert, wurde erreicht.
Zur Evaluierung des neuen RFID Webauthentifizierungssystem können Endkunden Testgeräte, wie Karten und Reader sowie ein Testsystem bei der Firma dxb gmbh anfordern. Für das Authentifizierungssystem können verschiedenen Accessoires, wie Karten, Armbänder oder Schlüsselanhänger mit der entsprechenden NFC Technologie verwendet werden.
Durch die Optimierung wird der zeitintensive Anmeldeprozesses mit Benutzername und Passwort vereinfacht und sicherer, da keine potentiell unsicheren Passwörter eingesetzt werden.

Abbildung 1.1: Anmeldebildschirm des Gordos mit Readerauswahl für NFC Anmeldung

2 Technischer Bericht

2.1 Einleitung

Der Technische Bericht ist für Ingenieure, welche die Hintergründe und Abläufe der Prozesse genauer verstehen wollen.

Für das bessere Verständnis der nachfolgenden Kapitel wird hier nochmals eine kurze Übersicht (Abbildung 2.1) über die Systemkomponenten gegeben.
Der Benutzer will sich über ein Terminal (Kapitel 2.7.4) mit einem Webbrowser am Gordo anmelden. Anstatt seinen Benutzernamen und Passwort einzugeben, wählt er ein NFC Reader (Kapitel 2.7.1) aus und legt seine persönliche NFC Karte auf den Reader. Im Hintergrund liest dieser die Karte und sendet die Daten an das Authentifizierungsmodul (Kapitel 2.7.2), welches die Daten überprüft und den Benutzer am Gordo einloggt.
Die Reader sind nicht statisch einem Projekt oder Kunden zugeordnet und können so je nach Bedarf ausgetauscht und einem andern Projekt oder Kunden zugewiesen werden. Diese Flexibilität wird durch einen Provisioning Service (Kapitel 2.7.3) ermöglicht. Bei jedem Starten prüft der Reader beim Provisioning Service, ob seine Projektzuordnung noch aktuell ist und aktualisiert diese gegebenenfalls.

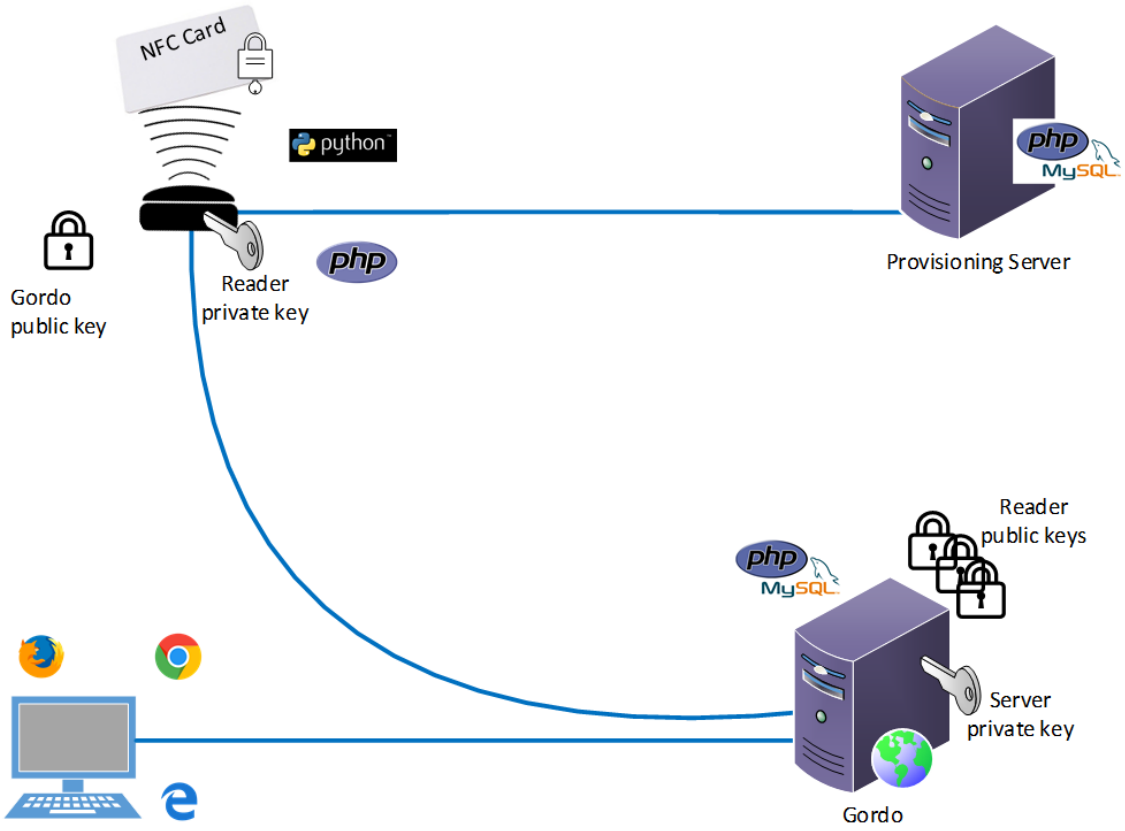


Abbildung 2.1: Übersicht vom RFID-Webauthentifizierungssystem [11] [2] [13] [10] [5] [4] [6]

2.2 Anforderungen

2.2.1 Use Case Brief

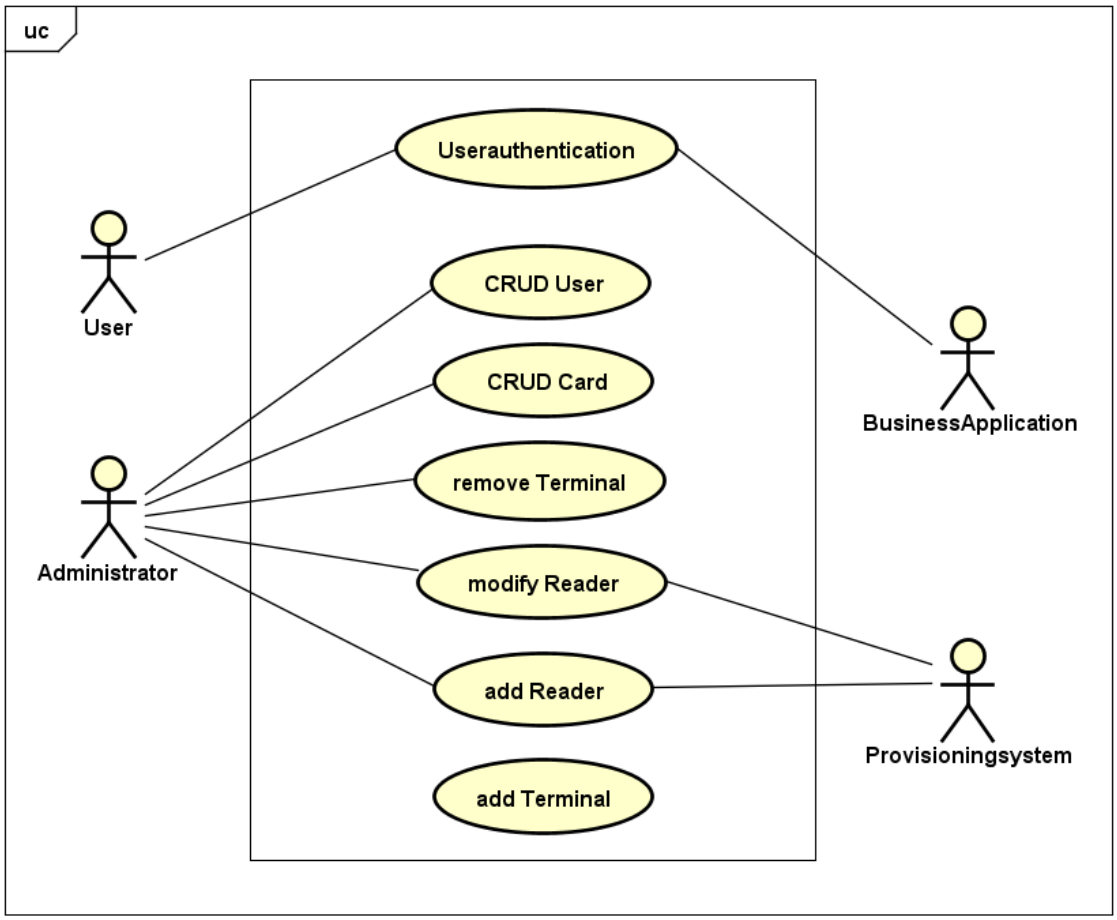


Abbildung 2.2: Use Cases des RFID-Webauthentifizierungssystems

2.2.1.1 UC1: Userauthentication

Ein User kommt an ein Terminal, legt seine Karte auf den Reader und wird automatisch am Gordo angemeldet.

2.2.1.2 UC2: CRUD User

Der Administrator muss den User im System erfassen, bearbeiten und löschen können.
Es existiert bereits eine Schnittstelle im Gordo für diesen Prozess.

2.2.1.3 UC3: CRUD Card

Der Administrator muss eine Karte im System erfassen, bearbeiten und wieder entfernen können. Dabei ist die Zuordnung einer Karte an einen neuen oder bereits existierenden Benutzer die Kernaufgabe.

2.2.1.4 UC4: remove Terminal

Beim Ersatz oder dem einfachen Ausscheiden eines Terminals aus dem Unternehmen, soll dieses auch vom System ausgeschlossen werden können.

2.2.1.5 UC5: modify Reader

Beim Ersatz oder Verlust eines Readers soll dieser vom Administrator auf dem System gesperrt, aber nicht gelöscht werden können, damit so eine Historie existiert.

2.2.1.6 UC6: add Reader

Der Administrator muss den Reader für das System vorbereiten, damit dieser anschliessend automatisiert erfasst und einem Kundenprojekt zugeordnet werden kann.

2.2.1.7 UC7: add Terminal

Ein beliebiges Terminal im Netzwerk wird beim Aufrufen der Loginpage des Gordo automatisch vom System erfasst.

2.2.1.8 Änderungen

Der Use Case **CRUD User** (Abschnitt 2.2.1.2) konnte komplett vom bestehenden System übernommen werden. Die existierende Schnittstelle wurde für die Zuweisung eines Benutzers zu einer Karte gemäss **UC3: CRUD Card** (Abschnitt 2.2.1.3) benötigt.

Beim Use Case **remove Terminal** (Abschnitt 2.2.1.4) hat sich gezeigt, dass dieser Prozess keine Relevanz für das System darstellt, da die Erfassung eines beliebigen Terminals jederzeit automatisch stattfinden kann. Das Bedürfnis zum Entfernen eines Terminals kann jedoch via der direkten Bearbeitung in der Datenbank trotzdem wahrgenommen werden. Zu beachten ist dabei jedoch, dass das Terminal jederzeit wieder erfasst wird, wenn es sich erneut am System anmeldet.

2.2.1.9 Genauere Spezifizierung

Beim Use Case **add Reader** (Abschnitt 2.2.1.6) muss der Administrator auf dem Reader ein Image und die Reader-Software installieren. Die restliche Konfiguration wird im Gordo erfasst und beim Aufstarten des Readers automatisch verteilt.

2.2.2 Nicht Funktionale Anforderungen

2.2.2.1 Funktionalität

Richtigkeit
Die Authentisierung der Benutzer soll in mindestens 99% der Fälle beim ersten Versuch gelingen.

Security
Die Reader, Karten und das neue Authentifizierungsmodul dürfen die Sicherheit des Gordo nicht kompromittieren.
Die Reader erhalten eine Vorkonfiguration, wodurch sie sich am System authentifizieren können und anschliessend eine eindeutige Identifikation erhalten. Zudem müssen die Karten kopiergeschützt sein.

Interoperabilität
Das neue Authentifizierungsverfahren muss problemlos mit dem bestehenden Gordo zusammenarbeiten können.

2.2.2.2 Zuverlässigkeit

Fehlertoleranz
Die Verfügbarkeit von Gordo ist beim Ausfall des Provisioning Service nicht betroffen. Beim nichtfunktionieren eines Readers besitzt der Benutzer eine Auswahl an Alternativen.

Konformität
Die Zuverlässigkeit des Systems wird durch eine stetige Kommunikation zwischen Reader und Server sichergestellt und entsprechend auf dem Reader sichtbar sein.

2.2.2.3 Benutzbarkeit

Verständlichkeit und Erlernbarkeit
Die Erweiterung des bestehenden Systems bietet, durch die schnellere Benutzeranmeldung via Karte und weg von einer manuellen Anmeldung via Tastatur, eine Vereinfachung. Dieser vereinfachte Prozessablauf ist schnell erlernbar und verständlich.

Bedienbarkeit
Der Aufwand zur Verwendung des Systems mit der RFID-gestützten Authentifizierung wird deutlich verringert, da bis zur Anmeldung nur eine geringe Interaktion mit dem System notwendig ist. Die geschätzte Zeiteinsparung liegt bei ungefähr 5-10 Sekunden pro Anmeldung, was sich bei einem Terminal mit vielen täglich wechselnden Benutzern mehrere Minuten pro Tag ausmacht.

2.2.2.4 Effizienz

Zeitverhalten/Performance
Der Benutzer sollte innerhalb von weniger als 4-5 Sekunden am System authentifiziert und angemeldet sein.

2.2.2.5 Wartbarkeit

Modifizierbarkeit
Das entstehende System beinhaltet einen unabhängigen Service (Provisioning) und ein Modul, welches an das bereits existierende Gordo angebunden wird. Es wird darauf geachtet, dass möglichst wenige Schnittstellen entstehen. Dadurch wird vermieden, dass Änderungen an einem System, die jeweils andere Umgebung möglichst nicht beeinflussen.

Stabilität
Das Authentifizierungsmodul soll bei Änderungen im Gordo nicht betroffen sein.

2.2.2.6 Portabilität

Installierbarkeit
Das neue Authentifizierungsmodul soll mit einem einfachen Update des Gordo und dem Einführen von Karten und Readern einsatzbereit sein.

Koexistenz
Die neue Art der Authentifizierung kann zusätzlich zur herkömmlichen Benutzeranmeldung im System existieren und verwendet werden.

2.3 Vorgaben und Voraussetzungen

2.3.1 Architektonische Ziele

- Das neue Authentifizierungsmodul soll eine einfache Erweiterung der bestehenden Business Applikation darstellen, wodurch das Modul nach Wunsch einfach aktiviert bzw. eingeführt werden kann, wenn Gordo bereits im Einsatz ist.
- Der Reader speichert und verwaltet seine Konfiguration lokal nach deren Erhalt vom Gordo.
- Der Reader ist durch eine eindeutige vom Provisioning Service generierte ID und selber generierten Private/Public Key eindeutig identifizierbar.
- Der Provisioning Service ist für alle Kundensysteme derselbe, da dieser nur für die erstmalige Zuweisung des Readers zu einem Kundenprojekt benötigt wird.
- Das gesamte Authentifizierungssystem wird nach der Zuordnung des Readers zum Kundenprojekt komplett unabhängig vom Provisioning Service agieren können.
- Die gesamte Kommunikation läuft über HTTPS. Zusätzlich wird bei der Übertragung sensibler Daten mit asymmetrischer Verschlüsselung und digitalen Signaturen gearbeitet.
- Es sollen alle möglichen Terminals (browserfähige Geräte) wie Notebook, PC oder Tablet vom neuen Authentifizierungsmodul profitieren können.
- Die Verwaltung der Benutzer und Karten ist pro Kunde im Gordo zentral organisiert.
- Die Erstellung und Zuweisung einer neuen Karte soll mit jedem Reader und von überall im System möglich sein, solange der Reader im selben Gordo ist.
- Das System muss auch zusammen mit Firewalls und NAT-Systemen funktionieren.

2.3.2 Einschränkungen

- Der Auftraggeber möchte kein Softwaremodul in der Programmiersprache Java und auch keine Plugin-Lösung.
- Die Kommunikation soll nicht über Sockets gelöst werden.
- Das neue Modul und alle dazugehörenden Komponenten sollen, wo sinnvoll mit den Programmiersprachen PHP und JQuery implementiertwerden. Dies vereinfacht die Wartung oder Erweiterung, da Gordo auch mit diesen Sprachen implementiert ist.
- Die Speicherung von Daten soll wie Gordo in der MySQL-Datenbank realisiert werden.
- Der Reader ist während einer Authentifizierung für andere Terminals gesperrt.
- Beim Autolearning Prozess sind alle Reader für eine kurze Zeitdauer durch die Anfrage eines Terminal besetzt.
- Der Prototyp des RFID-Webauthentifizierungssystems wird für einen bestimmten Reader von NXP für Raspberry Pi in Verwendung mit den Karten vom Typ NTAG213 und NTAG216 entwickelt.
- Das System wird nur für Karten mit einem passwortgeschützten Bereich ausgelegt sein, damit der geforderte physikalische Schutz der Karten garantieren werden kann.
- Die Karten respektive die sensitiven Daten auf der Karte soll kopiergeschützt sein.

2.3.3 Hardware

2.3.3.1 Raspberry Pi

Der Auftraggeber definierte den Raspberry Pi 3 Model B als Hardwaregrundlage für den NFC Reader bereits vor Beginn der Studienarbeit und stellte diese auch zur Verfügung.

2.3.3.2 NFC Hardwaremodul

Das NFC Hardwaremodul EXPLORE-NFC-WW von elements14 mit einem Chip von NXP wurde ebenfalls durch den Auftraggeber vor Beginn der Arbeit definiert. Die Dokumentationen zu diesem Modul sind eher spärlich vorhanden. Die Unterlagen zum Chip und den Unterstützten Technologien existieren meist nur auf internen Webseiten des Herstellers NXP.

2.3.3.3 NFC Karte

Die Vorgaben des Auftraggebers in Bezug auf die einzusetzenden NFC Karten war zu Beginn des Projektes sehr gering. Der wichtigste Punkt war, dass die Karten oder zumindest der Bereich mit den heiklen Informationen für die Authentifizierung kopiergeschützt sein muss.
Bei der Einarbeitung in die Technologie und der Evaluierung der Karten, konnte das Projektteam einen USB-NFC-Reader vom Institut ICOM ausleihen.
Nach einer ersten Einschränkung auf drei unterschiedliche Karten-Technologien, wurden die grundsätzlichen Funktionalität und deren Nutzung dem Auftraggeber präsentiert. Auf Grund der zur Verfügung stehenden Projektzeit und im Hinblick auf eine einfache und dennoch sichere Variante, sprach das Projektteam eine Technologie-Empfehlung für die NTAGs mit einem passwortgeschützten Bereich aus. Diese umfasste die NTAG 213 und NTAG 216. Die NTAG 216 wurden bereits mit dem NFC Hardwaremodul mitgeliefert, weshalb diese von Beginn an zur Verfügung standen. Der Auftraggeber traf die Entscheidung, dass für seine Kunden der kleinere Tag NTAG 213 reichen wird, aber das System beide NTAG-Typen unterstützen soll, damit eine Ausweichmöglichkeit existiert.
Diese NTAGs sind in diversen Accessoires, wie Karten, Armbänder, Sticker, Schlüsselanhänger erhältlich.

2.4 Konzeptionierung

In den ersten Wochen des Projektes führten das Projektteam, Andreas Eder und Pascal Kistler, einige Workshops mit dem Betreuer Ivan Bütler und dem Auftraggeber Daniel Bohl von der Firma dxb gmbh durch. In diesen Workshops wurde der Ablauf sowie diverse Wünsche des zu entwickelnden Systems gemeinsam erarbeitet. Das Team stellte die Ergebnisse dieser Meetings jeweils in Sequenzdiagrammen zusammen, entwickelte sie weiter und präsentierte diese wiederum an den folgenden Meetings für Rückfragen oder Feedbacks.
Diese Workshops fanden in der Elaborationsphase statt. Anhand der Diagramme konnte für die Kommunikation aller beteiligten Parteien und für die Definition des Scopes eine gemeinsame Grundlage geschaffen werden.
Das Team benutzte die Diagramme auch später im Projekt bei der Entwicklung, also in der Constructionphase, wodurch alle Diagramme stetig auf dem aktuellen Stand gehalten wurden.
Die gesamte Kommunikation zwischen Reader und Service, Reader und Server und Terminal und Server laufen via HTTPS ab, was in den Diagrammen nicht noch zusätzlich aufgezeigt wurde.
Im folgend Kapitel wird für den Server der webbasierten Managementlösung, auch Gordo genannt, der konzeptionelle Begriff **Business Applikation Server** eingeführt.

2.4.1 Sequenzdiagramme

Die folgenden Sequenzdiagramme zeigen die Abläufe des aktuelle Systems im Detail.

2.4.1.1 Übersicht

Zuerst betrachten wir die Übersicht (Abbildung 2.3) des gesamten RFID Webauthentifizierungssystems, bei welchem drei relevante Business Prozesse erkennbar sind.

Business Prozesse:

- Provisioning (Kapitel 2.4.1.2)
- Autolearning (Kapitel 2.4.1.5)
- Betriebsmodus (Kapitel 2.4.1.8)

Der Provisioning Prozess stellt den Einstiegspunkt für alle Reader zum System bereit und ist zuständig für die Zuteilung der Reader zum jeweiligen Kundenprojekt. Diese Zuteilung wird durch die Auslieferung von Kundenspezifischen Angaben über deren Business Applikation gemacht.

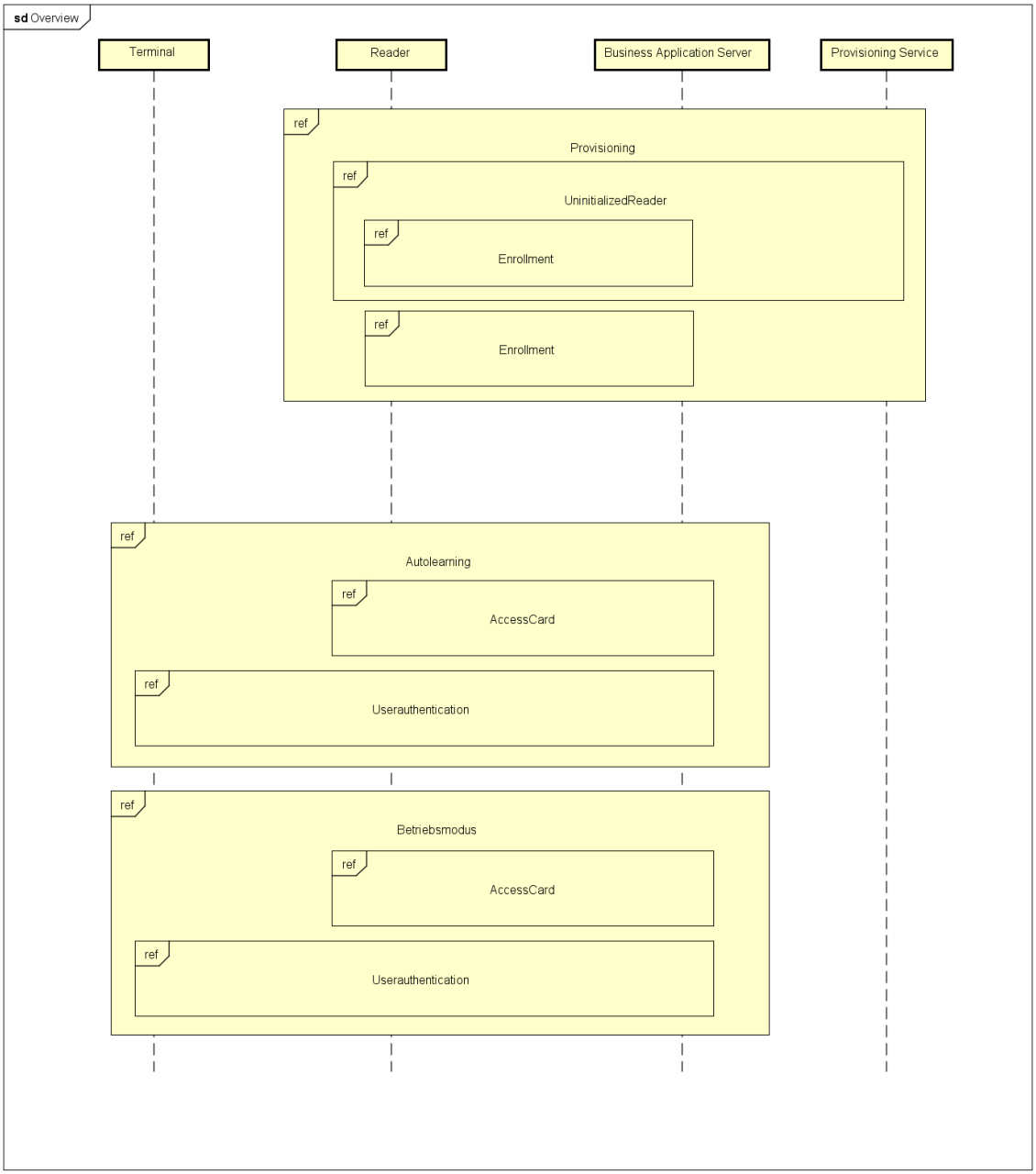


Abbildung 2.3: Darstellung des Sequenzdiagramm Overview

Die Prozesse Autolearning und Betriebsmodus sind grösstenteils gleich, da sie beide den Authentifizierungsvorgang bearbeiten. Der Unterschied besteht darin, dass der Prozess Autolearning eine Art Initialisierungsprozess widerspiegelt, bei welchem eine automatisierte Zuordnung zwischen Reader und Terminal vorgenommen wird.

2.4.1.2 Provisioning

Der Provisioning Prozess wird jedes Mal durchlaufen, wenn ein Reader gestartet wird, unabhängig davon ob dieser bereits einem Kundensystem (Business Applikation) zugeordnet ist oder nicht. Für diesen Prozess wird ein Provisioning Service vorausgesetzt, welcher zentral für alle Kunden geführt wird. Über diesen wird die Zuteilung eines Readers zu einem Kundenprojekt verwaltet. Er fungiert als Einstiegspunkt für alle Reader.

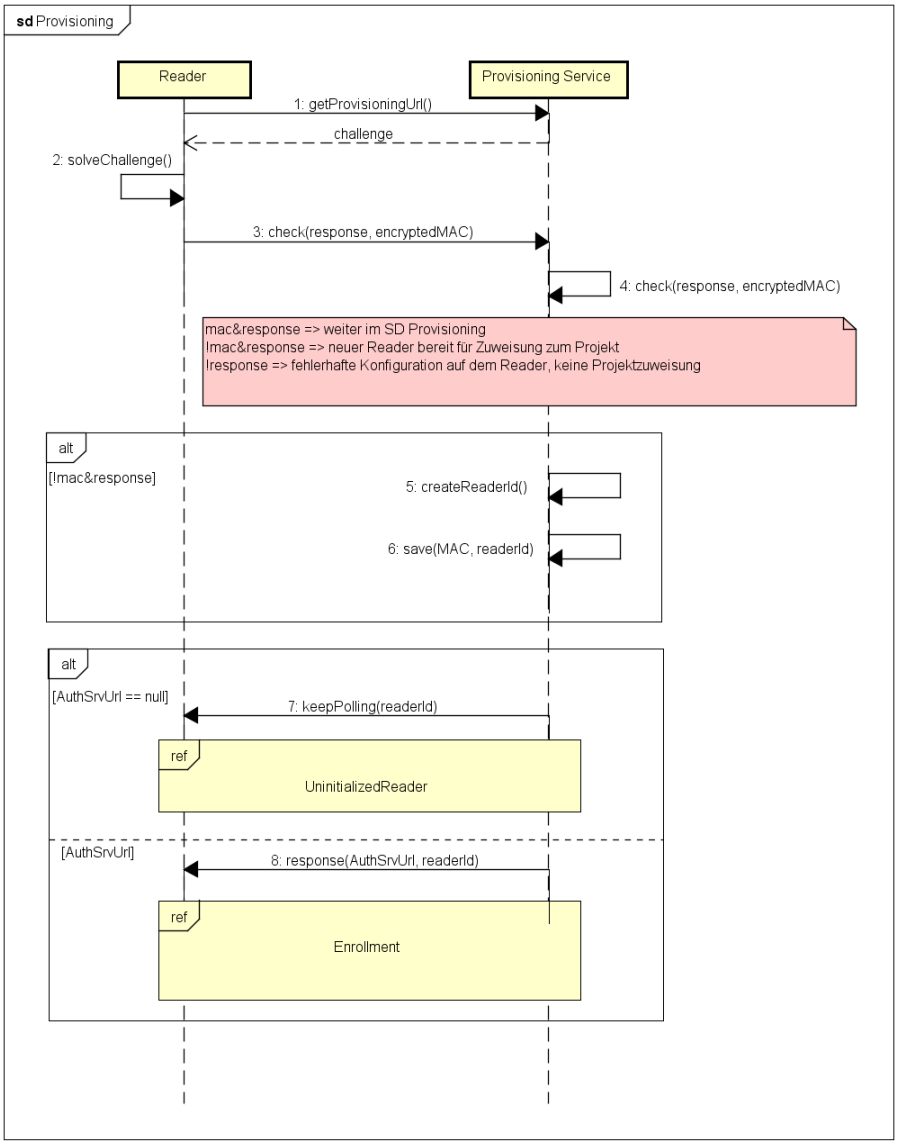


Abbildung 2.4: Darstellung des Sequenzdiagramms Provisioning

Ein Reader meldet sich beim Aufstarten beim Provisioning Service, welcher dem Reader eine Challenge ausliefert. Der Reader löst die Challenge und sendet die Response zusammen mit seiner verschlüsselten MAC-Adresse zurück an den Provisioning Service.

Besitzen beide das selbe SharedSecret, also einen gemeinsamen AES Schlüssel, dann wird der Reader vom Provisioning Service akzeptiert. Dieser merkt sich die MAC-Adresse des Readers und stellt ihm eine eindeutige ReaderId zu.
Solange der Reader keinem Kundenprojekt zugeordnet ist, fragt er beim Provisioning Service nach einer Konfiguration. Dieses Polling ist im Unterprozess Uninitialized Reader (Kapitel 2.4.1.3) dargestellt. Nun kann ein Administrator dem Reader manuell über die Benutzeroberfläche ein Projekt zuteilen.
Ist der Reader bereits im System bekannt, durch Vorerfassung oder weil er früher einem Kunden zugeordnet wurde, dann durchläuft der Reader den Unterprozess Enrollment (Kapitel 2.4.1.4).

2.4.1.3 Uninitialized Reader

Dieser Unterprozess wird durchlaufen, wenn sich ein Reader erfolgreich beim Provisioning Service registrieren konnte, dort jedoch noch keinem Kundensystem zugeordnet ist. Daher fragt der Reader in regelmässigen Abständen beim Provisioning Service an, ob er mittlerweile einem Kundenprojekt zugeordnet ist und bezieht so die Projekt-URL.
Ist die Projekt-URL leer wird der aktuelle Prozess so lange wiederholt bis eine gültige Projekt-URL zurückgegeben wird.
Beim Erhalt eines gültigen Wertes, der einer URL eines Authentifizierungsservers entspricht, startet der Reader mit dem Prozess Enrollment (Kapitel 2.4.1.4).

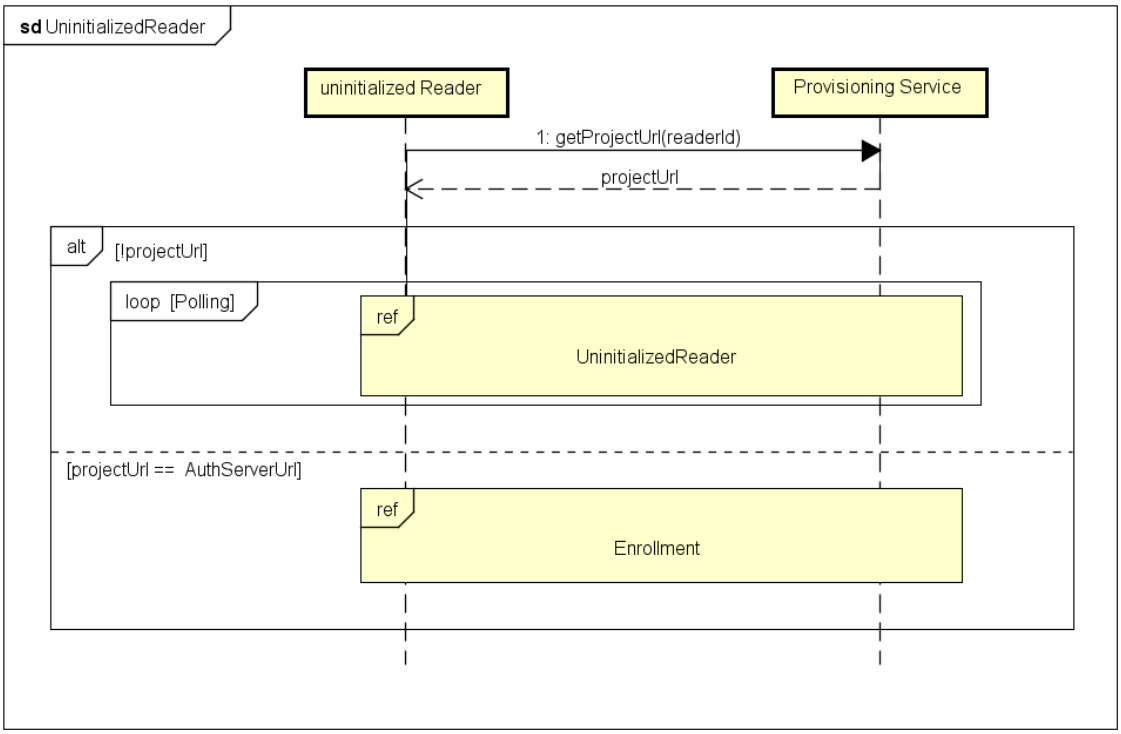


Abbildung 2.5: Darstellung des Sequenzdiagramms UninitializedReader

2.4.1.4 Enrollment

Besitzt der Reader eine URL eines Kundenprojektes, auch als Business Applikation Server bezeichnet, startet er nach einem Neustart automatisch mit dem Prozess Enrollment. Dabei erstellt er zuerst sich selbst einen privaten und öffentlichen Schlüssel für die asymmetrische Verschlüsselung. Anschliessend sendet er seinen öffentlichen Schlüssel (readerPubKey) und seine ReaderId dem Authentifizierungsserver (Auth Server), welcher das Authentifizierungsmodul des Business Applikation Server verkörpert. Der Authentifizierungsserver speichert sich den Reader mit dessen Id und dessen öffentlichen Schlüssel für die zukünftige Kommunikation. Der Prozess wird vom Server abgeschlossen, indem er dem Reader wiederum seinen öffentlichen Schlüssel (serverPubKey) mitteilt.

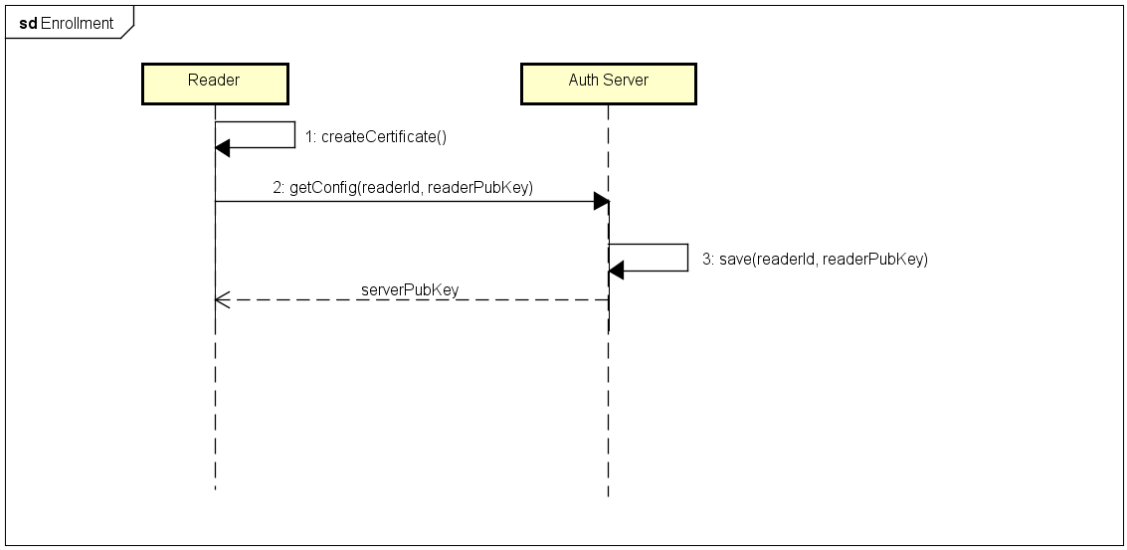


Abbildung 2.6: Darstellung des Sequenzdiagramms Enrollment

2.4.1.5 Autolearning

Der Autolearning Prozess kommt dann zum Einsatz, wenn sich ein Benutzer mittels seiner NFC Karte über ein Terminal am System authentifizieren will und den zu verwendenden Reader nicht kennt. Zum Einen tritt dieser Fall ein, wenn das benutzte Terminal noch keinen priorisierten Reader besitzt. Andererseits kann es auch sein, dass der Reader neben dem Terminal nicht demjenigen entspricht, welcher am Terminal vorausgewählt ist und somit nach dem nächstgelegenen Reader suchen will. In dieser Situation wählt der Benutzer die Variante den Reader automatisch zu suchen aus. Dies löst beim Business Applikation Server eine Anfrage für alle Reader aus. Der Benutzer kann sich dann mit seiner NFC Karte beim nächstgelegenen Reader authentifizieren. Der Business Applikation Server speichert sich zum gemeldeten Terminal den gewählten Reader.

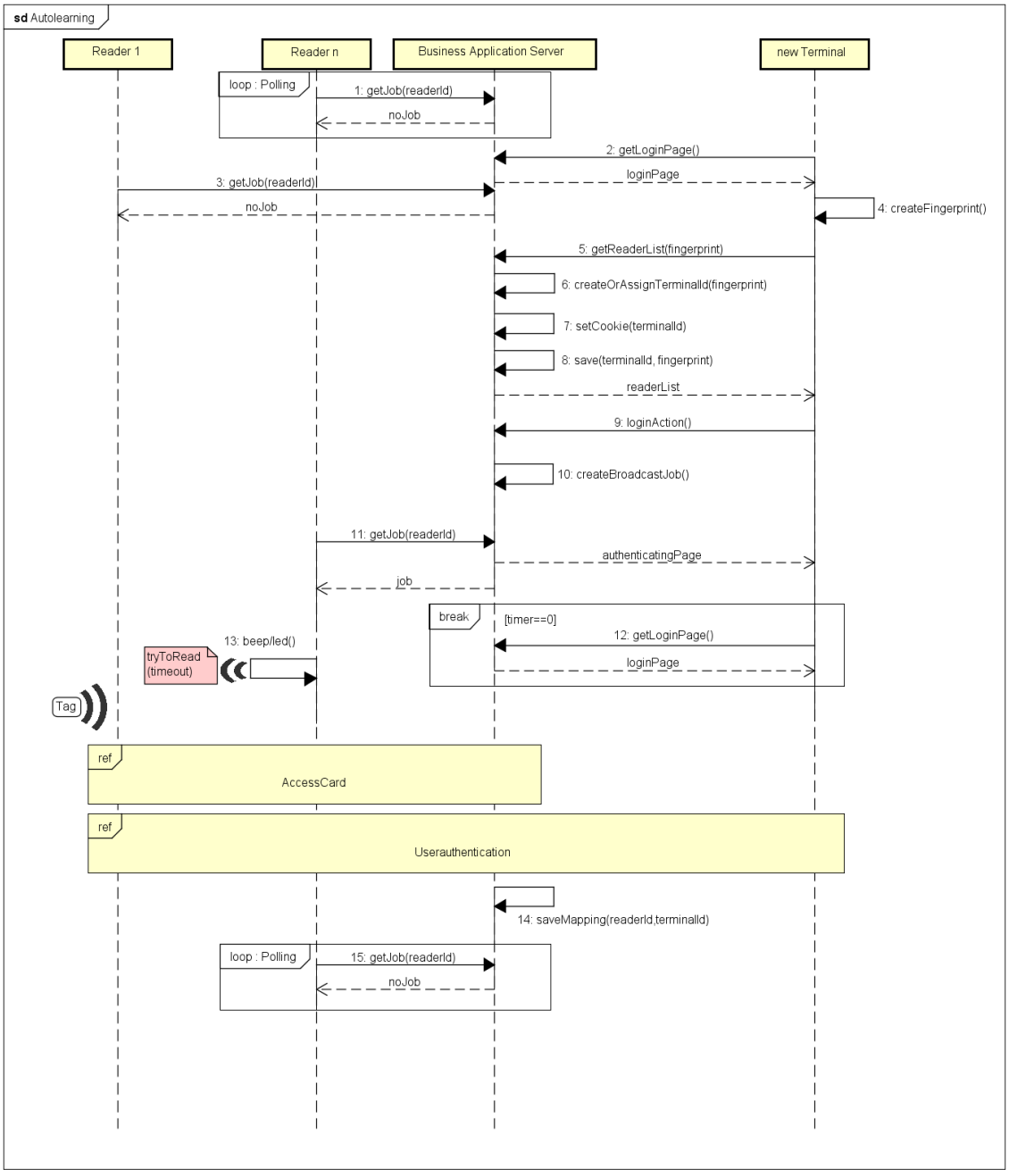


Abbildung 2.7: Darstellung des Sequenzdiagramms Autolearning

Alle Reader fragen in regelmässigen Abständen beim Business Applikation Server an, ob er einen Job für sie hat. In den häufigsten Fällen werden diese Anfragen mit Nein (noJob) beantwortet. Nun will sich ein Benutzer am System anmelden, wofür er die Anmeldeseite des Business Applikation Server aufruft, welche einen Fingerprint des verwendeten Terminals generiert. Dieser wird beim Abrufen der Readerliste dem Server zurückgesendet. Der Server ermittelt, ob der Terminal bereits im System eingetragen ist und erstellt gegebenenfalls

eine TerminalId, die er sich mit dem Fingerprint des Terminals speichert. Die TerminalId und eine vom System vorkonfigurierte Liste der Reader wird dem Terminal mitgeteilt. Der Benutzer wählt die automatische Suche nach Readern aus und klick Anmelden an. Daraufhin wird beim Server ein Broadcast-Job ausgelöst. Dieser Job wird von allen Readern abgeholt, wodurch diese zu blinken oder zu piepen beginnen. Dies ist das Signal für den Benutzer, dass er seine NFC-Karte auf den Reader legen kann.
Legt der Benutzer sein Karte auf den Reader, werden die Prozesse Access Card (Kapitel 2.4.1.6) und Userauthentication (Kapitel 2.4.1.7) abgearbeitet, bis sich der Server am Ende den ausgewählten Reader zum Terminal merkt, der Broadcast-Job beendet wird und alle Reader wieder gewohnt nach einem Job fragen.
Während der Reader für das Lesen der Karte bereit ist, läuft beim Terminal, solange der Broadcast-Job existiert, ein Timer ab. Wenn diese Zeit vorbei ist bevor die Authentifizierung des Benutzers stattgefunden hat, dann wird das Terminal automatisch wieder die Anmeldeseite des Business Applikation Servers anfordern.

2.4.1.6 Access Card

Der Prozess im folgenden Diagramm AccessCard (Abbildung 2.8) gehört zu einem der heikelsten Abläufe, da hierbei der Reader beim Server das Passwort für den Kartenzugriff erfragen muss. Bei diesem Ablauf ist der Reader am lesen und wartet bis eine NFC Karte oder ein anderer Gegenstand mit der verwendeten NFC-Technologie in seinen Lesebereich gehalten wird. So liest der Reader zuerst die CardId aus, verschlüsselt diese mit dem öffentlichen Schlüssel des Servers und signiert dies mit seinem privaten Schlüssel. Anschliessend sendet der Reader diesen String inklusive seiner eigenen Id (readerId) und derjenigen des zuvor erhaltenen Jobs (jobId) an den Business Applikation Server. Dieser verifiziert die Response mit Hilfe des ihm bekannten öffentlichen Schlüssels vom Reader und entschlüsselt die CardId mit seinem privaten Schlüssel. Kann die cardId erfolgreich entschlüsselt werden und ist die im System erfasst, erstellt der Server eine Challenge. Diese Challenge und das Passwort, um auf die Karte zuzugreifen, werden mit dem öffentlichen Schlüssel des Readers verschlüsselt (encryptedReaderPubKey()) und übertragen. Der Reader entschlüsselt diese Nachricht und kann mit dem erhaltenen Passwort auf den passwortgeschützten Bereich der Karte zugreifen, wo das gespeicherte Geheimnis (Secret) abgespeichert ist.

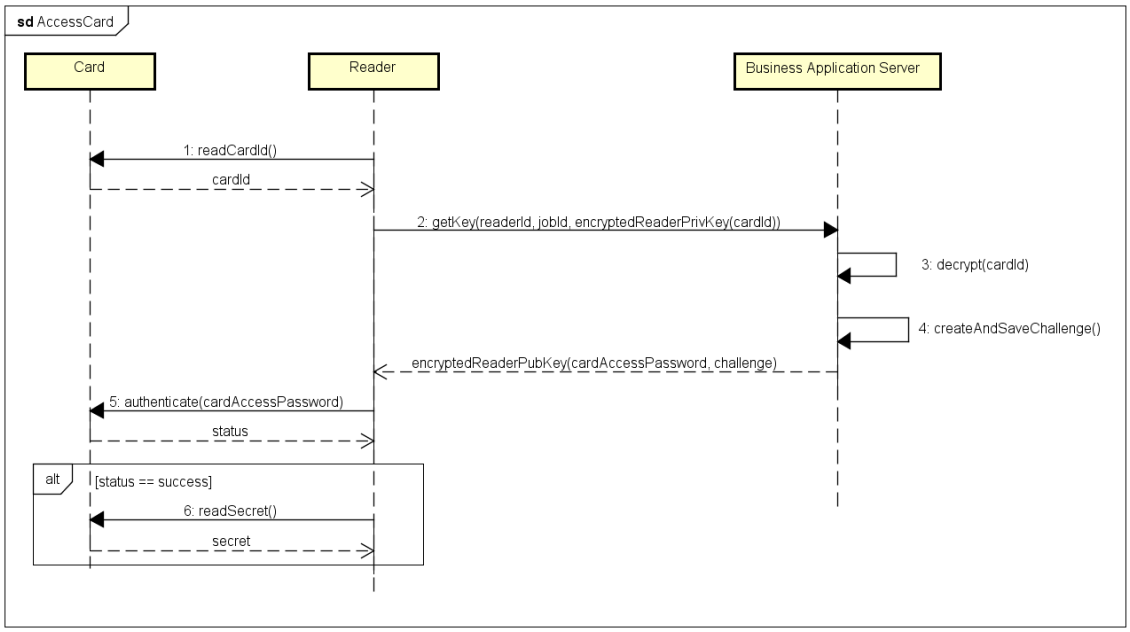


Abbildung 2.8: Darstellung des Sequenzdiagramms AccessCard

Nach dem Access Card Ablauf folgt immer der Prozess Userauthentication (Kapitel 2.4.1.7).

2.4.1.7 Userauthentication

Nachdem der Reader erfolgreich das Geheimnis (Secret) auf der Karte gemäss dem Ablauf Access Card (Kapitel 2.4.1.6) auslesen konnte, wird automatisch mit dem Prozess Userauthentication (Abbildung 2.9) fortgefahren.
Der Reader löst die erhaltene Challenge. Das Resultat (challengeResult) und die Id der Karte werden jeweils mit dem öffentlichen Schlüssel des Servers verschlüsselt und mit dem privaten Schlüssel des Readers signiert und zusammen mit der Id des Readers zurück an den Authentifizierungsserver gesendet. Dieser überprüft die Antwort des Readers, holt die Informationen des Kartenbesitzers aus der Datenbank und erstellt eine neue Session mit der Id des Benutzers.
Der Reader erhält eine Rückmeldung über den Status der Authentifizierung, welche auch für den Benutzer ersichtlich sein sollte. Daraufhin startet auch dieser Reader wieder mit den regelmässigen Anfragen an den Server. Zur gleichen Zeit erhält das Terminal einen Redirect zur Business Applikation, wie zum Beispiel das Gordo, welches in dieser Arbeit als Testsystem verwendet wurde.

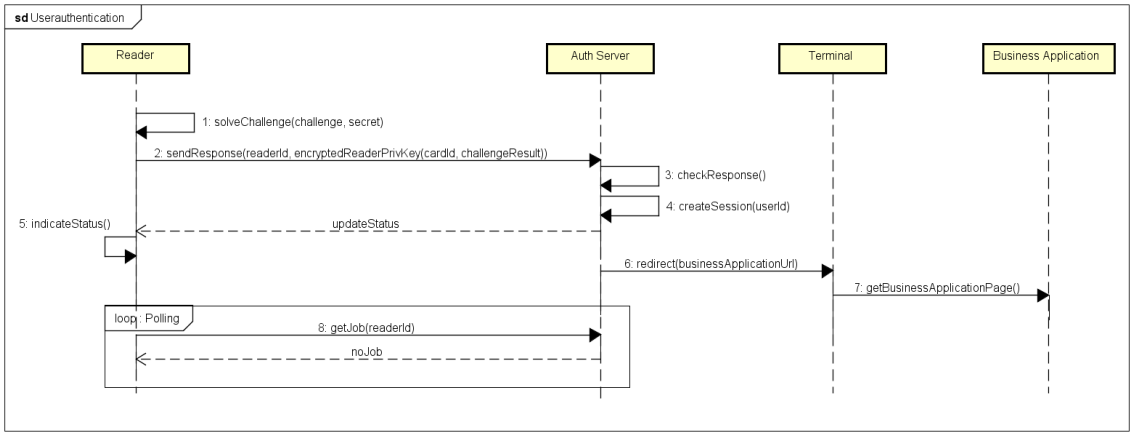


Abbildung 2.9: Darstellung des Sequenzdiagramms Userauthentication

Nur wenn alle Teilschritte von den Prozessen AccessCard (Kapitel 2.4.1.6) und Userauthentication (Kapitel 2.4.1.7) erfolgreich waren und somit von Anfang bis Ende die selben zwei Kommunikationspartner also selber Reader und Business Applikation Server sich unterhalten haben, kann die Karte vollständig identifiziert und der Benutzer erfolgreich authentifiziert werden.

2.4.1.8 Betriebsmodus

Der Betriebsmodus ist dem Prozess des Autolearning sehr ähnlich. Der Betriebsmodus ist derjenige Ablauf, der am Ende mehrheitlich im Einsatz sein wird und behandelt ebenfalls den Prozess, wenn sich ein Benutzer mittels seiner NFC Karte an einem Terminal authentifizieren will. Allerdings ist in diesem Falle ein konkreter Reader auf den Anmeldeseite ausgewählt worden. Diese Situation löst beim Business Applikation Server nur eine Anfrage für

einen spezifischen Reader aus. Der Benutzer kann sich dann mit seiner NFC Karte nur über diesen Reader authentifizieren.
Natürlich kann der Benutzer auch hier die Variante des automatischen Suchens auswählen, jedoch folgt der restliche Prozess dann wieder dem Autolearning (Kapitel 2.4.1.5).

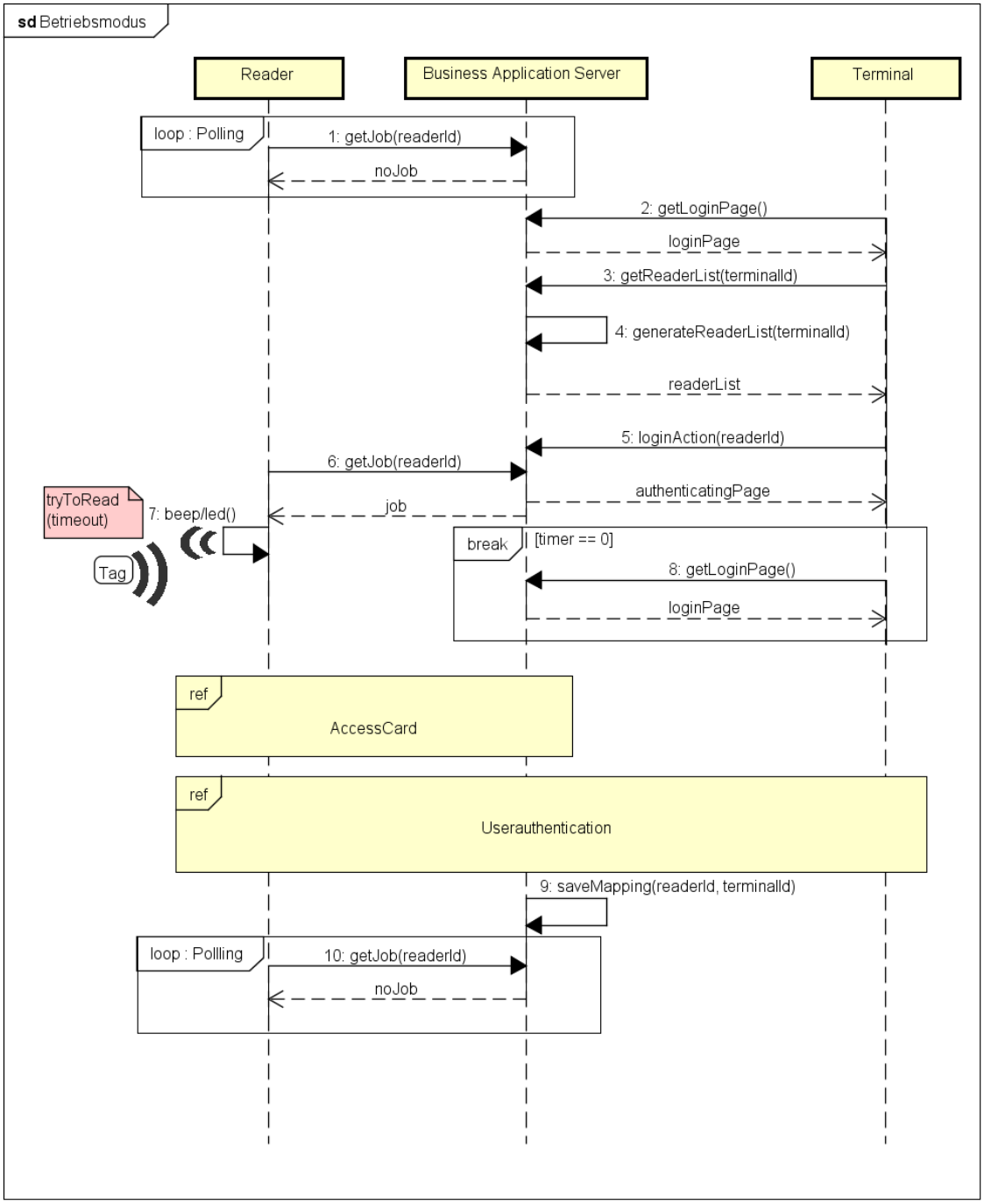


Abbildung 2.10: Darstellung des Sequenzdiagramms Betriebsmodus

Grundsätzlich fragen stetig alle Reader in regelmässigen Abständen beim Business Applikation Server an, ob er einen Job für sie hat. In diesem Diagramm ist dies vereinfacht mit einem Reader dargestellt.
Ein Benutzer will sich am System anmelden, wofür er die Anmeldeseite des Business Applikation Servers aufruft. Da der Terminal bereits eine TerminalId besitzt, berechnet er keinen Fingerprint, wie dies beim Autolearning der Fall ist. Stattdessen kann er seine TerminalId direkt mit der Anfrage nach der Liste mit den Readern an den Server senden. Anhand der TerminalId, generiert der Server eine terminalspezifische Readerliste, welche er ausliefert. Der Benutzer wählt einen Reader aus und klickt auf der Anmeldeseite anmelden, wodurch das Terminal die Statusseite der Authentifizierung angezeigt bekommt.
Der Reader holt sich in dieser Zeit seinen Job beim Server ab und signalisiert via einer LED oder einem Piepser dem Benutzer, dass dieser für seine Authentifizierung am System die Karte auf den Reader legen kann.
Nutzt der Benutzer diese Möglichkeit, dann werden die Prozesse Access Card (Kapitel 2.4.1.6) und Userauthentication (Kapitel 2.4.1.7) abgearbeitet, bis sich der Server am Ende den ausgewählten Reader zum Terminal merkt und der Reader wieder wie gewohnt nach einem neuen Job fragt.
Während der Reader bereit für das Lesen der Karte ist, läuft beim Terminal ein Timer ab, solange der Job existiert. Wenn diese Zeit vorbei ist bevor die Authentifizierung des Benutzers stattgefunden hat, dann wird das Terminal automatisch wieder die Anmeldeseite des Business Applikation Servers anfordern.

2.4.1.9 Initialize Card

Dieser Prozess ist auf der Übersicht des Gesamtsystems nicht ersichtlich, da er nicht zum normalen Ablauf gehört. Stattdessen ist dies ein spezieller Prozess nur für den Administrator. Ausserdem ist an diesen Ablauf die Vorbedingung geknüpft, dass mindestens ein Reader bereits dem System zugeordnet sein muss.

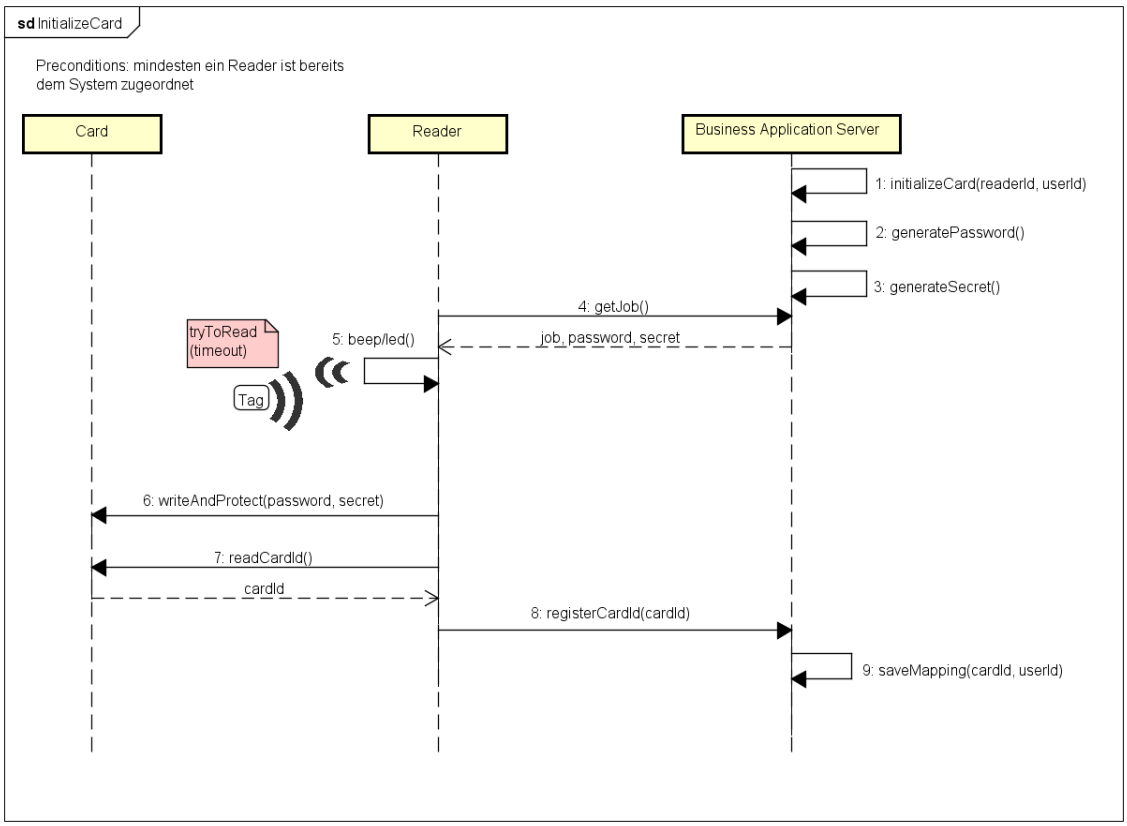


Abbildung 2.11: Darstellung des Sequenzdiagramms InitializeCard

Hierbei startet ein Administrator des Systems den Initialisierungsprozess über das Userinterface innerhalb der Business Applikation. Dabei wählt er den Reader, mit welchem er die Karte beschreiben will, den Benutzer für wenn die Karte ist und startet den Prozess. Danach wird automatisch ein Passwort und ein Geheimnis (Secret) generiert, sowie ein Job für den Reader erstellt. Der Reader holt sich den neuen Job inklusive Passwort und Geheimnis ab und signalisiert dem Administrator, dass er bereit für das Beschreiben einer Karte ist. Der Reader schreibt das Geheimnis auf die Karte, schützt diese mit dem Passwort und liest die Id der Karte aus. Die CardId sendet der Reader zurück an den Server. Dieser speichert die Karte mit allen notwendigen Informationen in der Datenbank und weist den Karteneintrag dem zuvor gewählten Benutzer zu.

2.4.2 Systemsequenzdiagramme

Diese Diagramme zeigen die direkte Interaktion zwischen dem Administrator und dem Endsystem auf. Bei den Prozessen, welche die Karte betreffen, wird das Endsystem als Business Applikation Server dargestellt.
Bei den Prozessen mit dem Reader werden noch weitere Systeme miteinbezogen, da der Administrator entweder direkt oder indirekt mit mehreren Endsystemen wie dem Provisioning Service, Business Applikation Server und dem Reader kommuniziert.

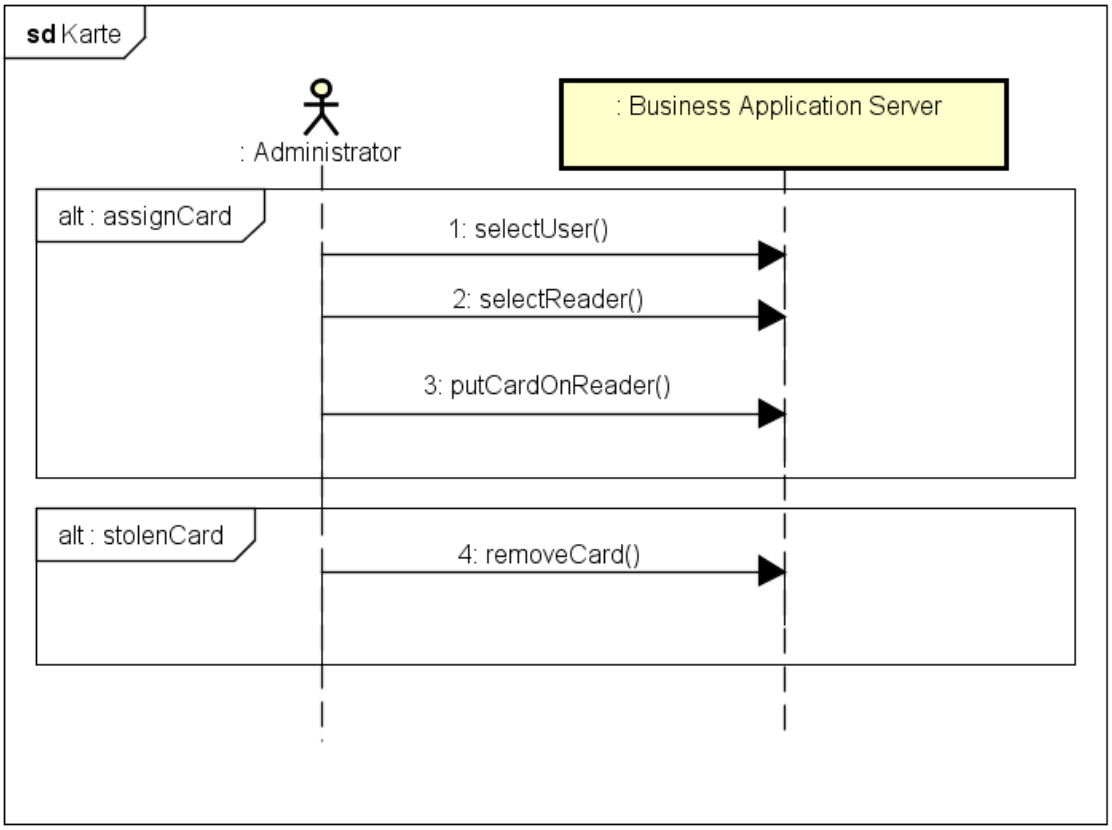


Abbildung 2.12: Darstellung des Systemsequenzdiagramms zu der Komponente Karte

2.4.2.1 Zuweisen einer Karte

Dieser Prozess wird im Systemsequenzdiagramm Karte (Abbildung 2.12) als **assignCard** bezeichnet. Hierbei will der Administrator eine Karte, egal ob neu oder recycelt, einem Benutzer zuweisen. Um dies zu erreichen, wählt er den neuen Besitzer der Karte und den Reader, mit welchem er die Karte im System registrieren will. Anschliessend legt er die Karte auf den Reader und löst somit die Zuteilung aus.

2.4.2.2 Gestohlene Karte

Dieser Prozess ist im Systemsequenzdiagramm Karte (Abbildung 2.12) als **stolenCard** aufgeführt. Dabei kann der Administrator die Karte ganz einfach auf dem Business Applikation Server deaktivieren.

2.4.2.3 Ende des Reader-Lebenszyklus

Es wurde ermittelt, dass ein Prozess benötigt wird, der das Lebensende eines Readers (Abbildung 2.13) behandelt. Dieser Prozess lautet **EndOfLifeReader** und kommt dann zum Einsatz, wenn ein Reader nicht mehr länger zu einem Kundensystem gehören soll. Dies kann unterschiedliche Gründe haben, wie beispielsweise, dass ein Reader nur temporär bei einem Kunden im Einsatz war oder der Reader defekt ist.
Für diesen Ablauf muss der Administrator zuerst auf dem Business Applikation Server den Reader deaktivieren, jedoch nicht löschen, damit noch die Historie existiert. Dieser Verlauf wurde vom Auftraggeber explizit gewünscht. Anschliessend entfernt der Administrator noch die Projektzuweisung des Readers, wobei der Reader auch hier noch im System erhalten bleibt, damit er gegebenenfalls später einem neuen Projekt zugeordnet werden kann. Dies wurde ebenfalls vom Auftraggeber explizit gefordert. Beim nächsten Neustart des Readers meldet sich dieser wie gewohnt als Erstes beim Provisioning Service, welcher ihm die aktuell zugewiesene Projekt-URL übermittelt. Da diese URL der Zuweisung zu einem Business Applikation Server entspricht, überschreibt der Reader seine Kundenprojektzuordnung hiermit und wird sich nur noch mit dem Provisioning Service verbinden können, bis er von diesem eine neue Projektzuweisung erhält.

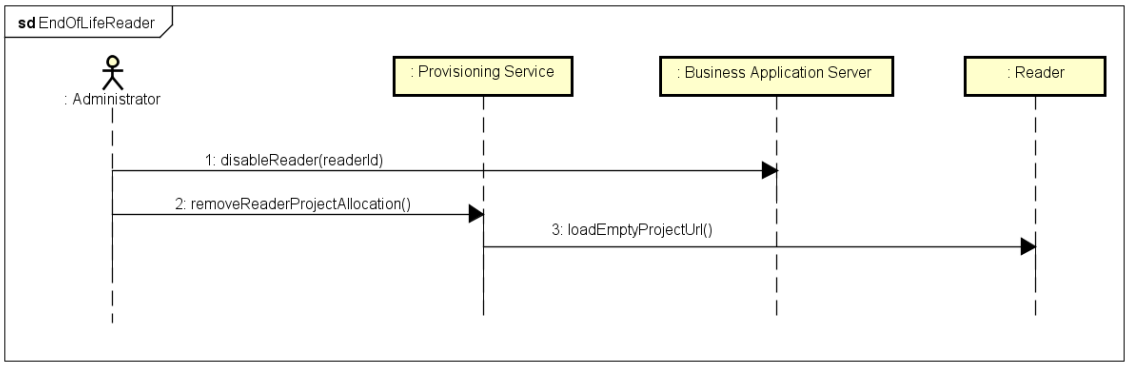


Abbildung 2.13: Darstellung des Systemsequenzdiagramms zum Prozess **End of Life** des Readers

2.4.2.4 Erfassen eines neuen Readers

Die Erfassung eines neuen Readers (Abbildung 2.14) wird als Prozess **newReader** aufgelistet. Bei dieser Systeminteraktion hat der Administrator den Reader zuerst physisch bei sich auf dem Tisch, weil er auf diesen gemäss der Anleitung **Inbetriebnahme Reader** als erstes ein Image und die Reader-Software installieren muss. Anschliessend erfasst der Administrator manuell die MAC-Adresse des Readers sowie dessen Projektzuteilung. Startet der Reader, dann bezieht er die URL des Kundenprojektes, also die Business Applikation Server URL und registriert sich somit bei diesem.

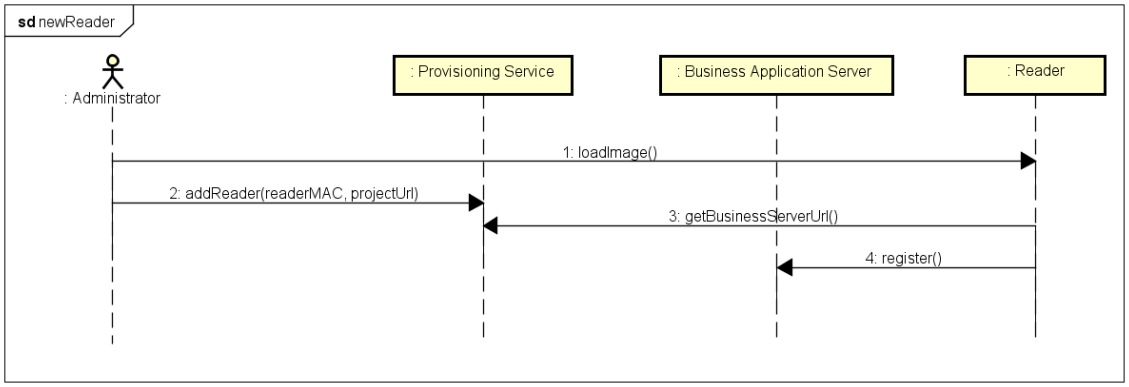


Abbildung 2.14: Darstellung des Systemsequenzdiagramms zum Prozess der Einführung eines neuen Readers

2.4.2.5 Gestohlener Reader

Mit dem folgenden Prozess wird der Diebstahl eines Readers (Abbildung 2.15) unter dem Begriff **stolenReader** erläutert. Dieser Ablauf wird ausgelöst, wenn ein Reader gestohlen wird, da in diesem Fall von einem geplanten Angriff ausgegangen wird, weil dadurch die Sicherheit des Systems in Gefahr ist. Alternativ wird der Prozess auch verwendet, wenn ein Reader physisch nicht mehr gefunden werden kann, weil dieser Verlust ebenfalls ein Sicherheitsrisiko bedeutet, da der unbekannte Aufenthaltsort überall und somit auch bei einem Angreifer sein kann.
Hierbei deaktiviert der Administrator zuerst den verschwundenen Reader auf dem Business Applikation Server und erstellt als Nächstes ein neues gemeinsames Geheimnis (SharedSecret) auf dem Provisioning Service. Das Geheimnis wird dort gespeichert und dem Administrator angezeigt, welcher es manuell per Mail an die Administratoren aller Business Applikation Server sendet. Die Administratoren tragen das neue Geheimnis wiederum im System der Business Applikation ein. Dort wird jeweils ein Update-Job für jeden Reader erstellt, welcher somit das neuste Geheimnis des Provisioning Service erhält, welches er bei einem Neustart dringend für eine erfolgreiche Kommunikation mit dem Provisioning Service benötigt. Sollte ein Reader in diesem Zeitraum neu starten bevor er das Update erhalten hat, dauert es einfach länger bis er sich mit seiner aktuellen Konfiguration wieder beim Business Applikation Server meldet, wo er dann sein Update verspätet erhält, da er auf dem Business Applikation Server noch aktiviert ist.
Der Administrator muss bei diesem Ablauf ebenfalls darauf achten, dass er das gemeinsame Geheimnis (SharedSecret) des Provisioning Service auch für das Reader-Images aktualisiert.

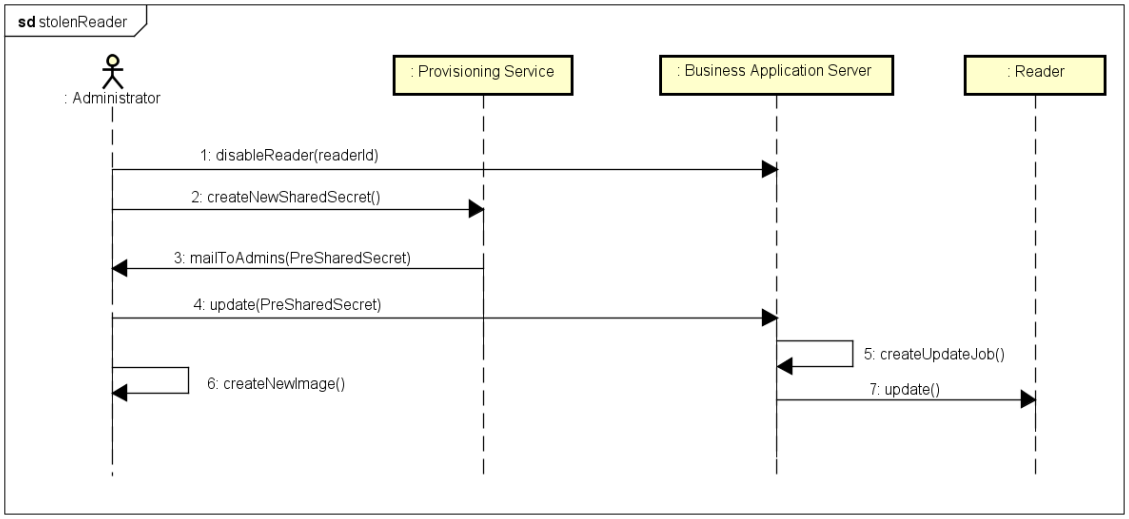


Abbildung 2.15: Darstellung des Systemsequenzdiagramms zum Prozess, wenn ein Reader gestohlen wird

2.4.2.6 Reader updaten

Der Prozess **UpdateReader** wird benutzt um einen Reader mit einer neuen Konfiguration zu aktualisieren (Abbildung 2.16).
Hierbei macht der Administrator auf dem Business Applikation Server eine Änderung an der kundenspezifischen Konfiguration, welche für alle Reader eines Kunden dieselbe ist. Dies löst automatisch ein Update für alle Reader im Kundensystem aus.

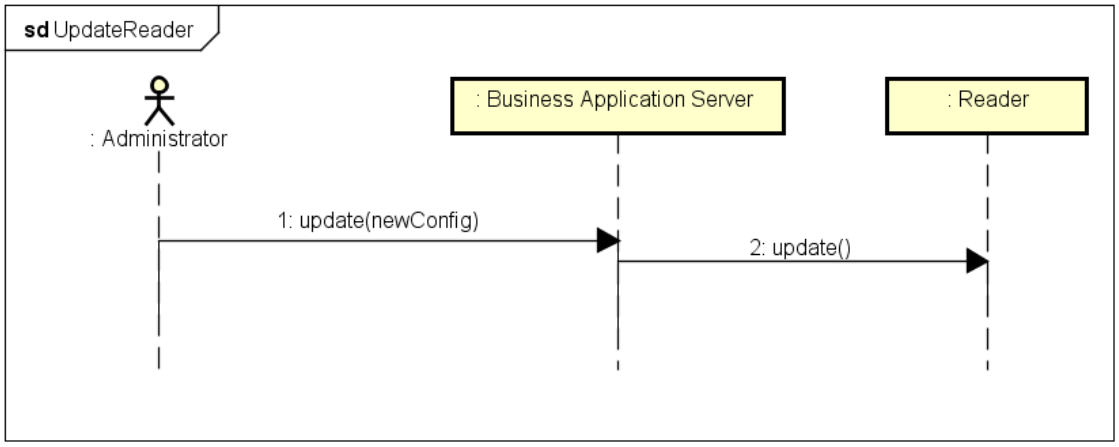


Abbildung 2.16: Darstellung des Systemsequenzdiagramms zum Prozess, wenn ein Reader upgedatet wird

Dieses Systemsequenzdiagramm stellt auch einen Teilprozess des Ablaufes des gestohlenen Readers (Kapitel 2.4.2.5) dar.

2.4.3 Deploymentdiagramm

In den gemeinsamen Workshops mit dem Auftraggeber entstand das detaillierte Verständnis für den grossen Zusammenhang zwischen allen beteiligten Systemen. Dieses Zusammenspiel des Authentifizierungsmoduls, des Provisioning Service, des Readers und des Terminals ist im folgenden Deploymentdiagramm (Abbildung 2.17) dargestellt. Die Server und der Reader laufen mit einem Linux-Betriebssystem.
Die gesamte Kommunikation zwischen den einzelnen Systemen läuft über HTTPS. Die Verbindungen zu den Datenbanken wird nicht zusätzlich abgesichert, da nur innerhalb des abgeschlossenen Systems darauf zugegriffen wird.

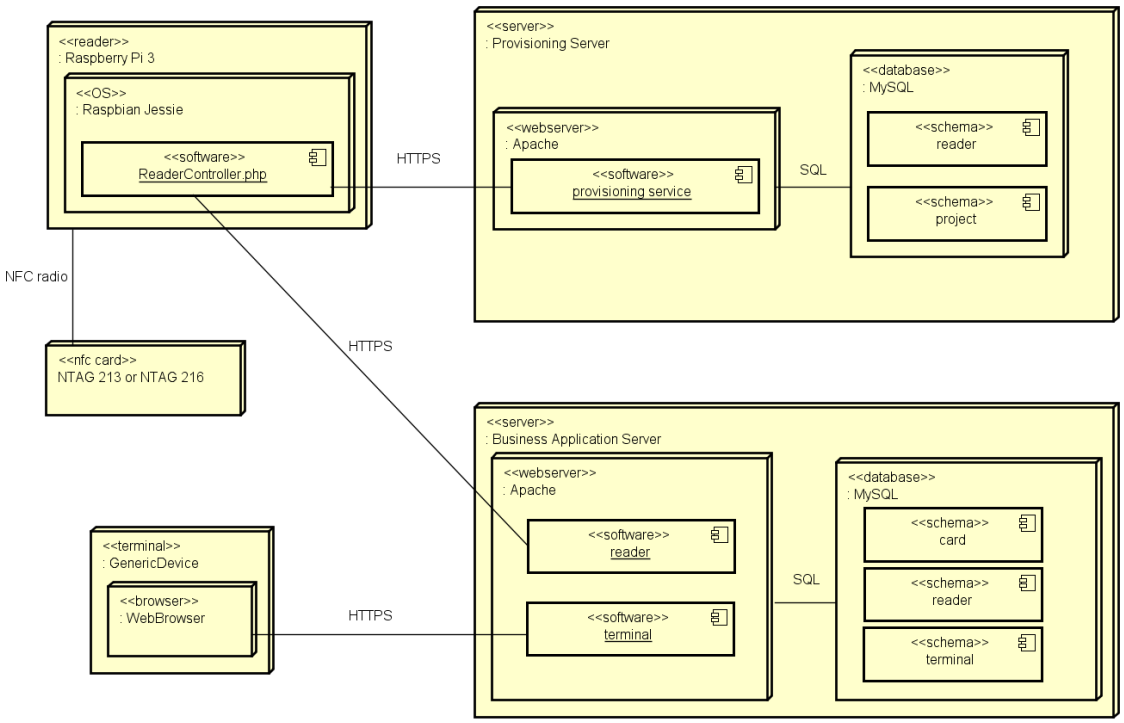


Abbildung 2.17: Deploymentdiagramm des RFID Webauthentifizierungssystems

2.4.3.1 Reader

Der Reader besteht hardwareseitig aus einem Raspberry Pi 3 mit einem NFC Hardwaremodul. Das NFC Modul wird softwaremässig angesteuert und stellt die Kommunikation mit der Karte sicher. Die Software des Readers wird beim Einschalten automatisch gestartet und ist für die gesamte Kommunikation zu den anderen Systemkomponenten verantwortlich.

2.4.3.2 Provisioning Service

Der Provisioning Service ist ein eigenständiger Service, der prinzipiell nur einmal im gesamten Verwaltungssystem existiert, daher ist er im Deplyomentdiagramm (Abbildung 2.17)

auch auf einem dedizierten Server eingetragen. Allerdings ist er so aufgebaut, dass er problemlos auch parallel zu der Business Applikation auf dem selben Server betrieben werden kann.
Der Service benötigt für seine Aufgaben Zuordnungsinformationen zwischen Readern und Projekten, welche er in seiner Datenbank abspeichert.

2.4.3.3 Authentifizierungsmodul

Das Authentifizierungsmodul des RFID Webauthentifizierungssystems ist bei dessen Verwendung immer fixer Bestandteil der Business Applikation, weshalb dies auch auf demselben Server läuft. Mit dem Authentifizierungsmodul wird die Zuordnung zwischen Reader und Terminal sowie diejenigen zwischen Karten und Benutzer bewerkstelligt. Des Weiteren werden Reader, Terminal, Karte und Benutzer mit diesem Modul eindeutig identifiziert, wodurch sich der Anwender mit seiner personalisierten NFC Karte am System authentifizieren kann. Für seine Aufgaben speichert sich das Authentifizierungsmodul sämtliche Informationen zu den Karten, Readern und Terminals in der Datenbank. Die benötigten Informationen der Benutzer bezieht das Authentifizierungsmodul aus der Datenbank der Business Applikation.

2.4.3.4 Terminal

Das Terminal symbolisiert ein beliebiges Gerät mit einem Webbrowser, wie beispielsweise ein Computer, Laptop, Smartphone oder Tablet. Die Terminal-Applikation ist betriebssystemunabhängig. Auf dem Terminal wird vom Benutzer die Webseite der Business Applikation geöffnet und der Authentifizierungsprozess mittels der NFC Anmeldung auf der Anmeldeseite gestartet.

2.4.3.5 Datenbanksysteme

Die Daten werden in einem MySQL-Datenbanksystem auf dem jeweiligen Server gespeichert. Der Provisioning Service benötigt nur eine kleine Datenbank. Für das Authentifizierungsmodul muss die bestehende Datenbank des Business Applikation Servers um ein paar Tabellen erweitert werden.

2.4.3.6 NFC Karte

Das System ist für die NFC Karten vom Typ NTAG 213 und NTAG 216 implementiert. Diese beiden NTAG-Typen besitzen einen passwortgeschützten Speicherbereich, in welchem die geheimen Anmeldeinformationen gespeichert werden. Das individuelle Passwort ist beim Authentifizierungsmodul hinterlegt und wird vom Reader angefordert, um für die Benutzerauthentifizierung die geheimen Daten von der Karte zu lesen.

2.4.4 Schichtenmodelle

Schichtenmodelle der neu zu entwerfenden Systeme erlaubten dem Projektteam einen strukturierten Aufbau der neuen Teile der Systemumgebung.

2.4.4.1 Reader

Der Reader besteht aus drei Schichten (Abbildung 2.18), welche alle unterschiedliche Funktionalitäten aufweisen. Dabei wird die Access-Schicht für den Zugriff auf das NFC-Hardwaremodul und die benötigten Dateien des Readers sowie für die Kommunikation über das Netzwerk benutzt. Die Service-Schicht ist verantwortlich für alles was mit der Kryptographie zu tun hat, wie der Ver- und Entschlüsselung. In der Application-Schicht befindet sich das Hauptprogramm für den Reader, welches beim Einschalten des Readers automatisch gestartet wird.

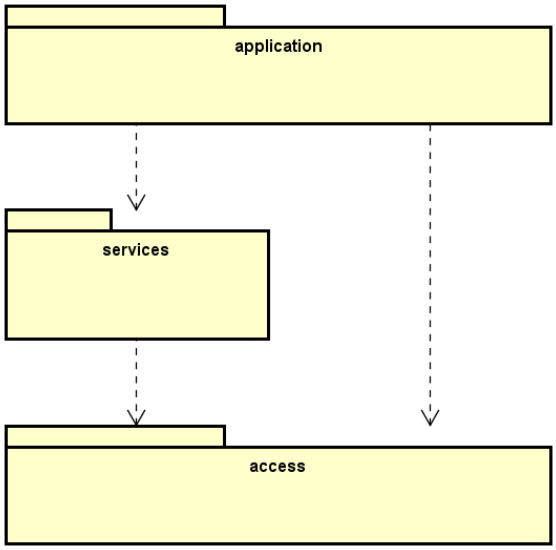


Abbildung 2.18: Schichtenmodell des Readers

2.4.4.2 Provisioning Service

Der Provisioning Service besitzt zwei eigene Schichten und eine fremde (Abbildung 2.19). Die Persistence-Schicht realisiert den Zugriff auf die Datenbank. Die mittlere Schicht besteht aus zwei Paketen, welche unterschiedliche Aufgaben ausüben. Das Business-Paket enthält die Logik des Provisioning Service, was unter anderem die Behandlung von Datenbankobjekten oder auch die kryptografischen Aufgaben umfasst. Das Service-Paket stellt die ganze Kommunikation zwischen dem Reader und dem Provisioning Service sicher, in dem es Anfragen empfängt und bearbeitet in Zusammenarbeit mit den Klassen aus dem Business-Paket.
Da der Provisioning Service als eigenständiger Dienst betrachtet wird, besitzt er keine eigene Presentation-Schicht, stattdessen bietet er alles notwendige an um eine beliebige Benutzeroberfläche anzubinden. Diese Umsetzung wurde gewählt, da der Auftraggeber seine momentan aktuelle Benutzeroberfläche der Business Applikation auch für die manuelle Verwaltung des Provisioning Service verwenden will. Daher wurden für den Provisioning Service Anpassungen in der bereits existierenden Benutzeroberfläche gemacht, damit die Verwaltung des Dienstes möglich ist, aber diese angepassten Dateien werden als externer Bestandteil des Provisioning Service betrachtet.

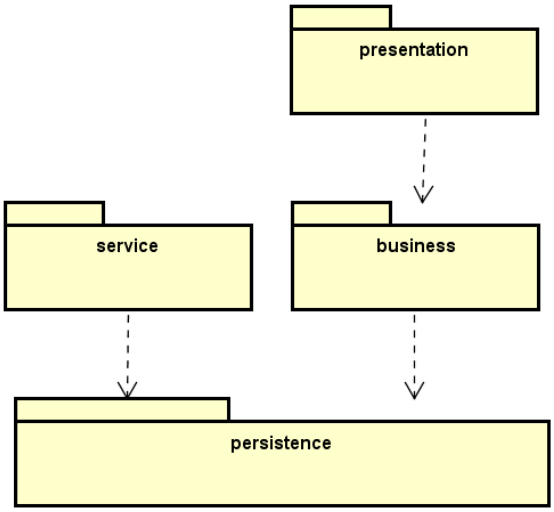


Abbildung 2.19: Schichtenmodell des Provisioning Service

2.4.4.3 Authentifizierungsmodul

Zum Authentifizierungsmodul gibt es kein Schichtenmodell, da dieses nicht in Schichten unterteilt werden konnte. Die Begründung liegt in der Struktur der bereits existierenden Business Applikation. Da das Authentifizierungsmodul als eine Erweiterung von dieser Software fungiert, wurde entschieden sich auch an dessen Struktur anzulehnen.

2.4.5 Domainmodelle

Das Projektteam orientiert sich beim Softwareaufbau am Grobkonzept der bestehenden Business Applikation und speichert somit ebenfalls alle Objekte der neuen serverseitigen Softwarekomponenten in einer Datenbank ab.
Daher widerspiegeln die folgenden Domainmodelle zu den neuen Komponenten nicht nur das Verhalten innerhalb der Software sondern auch den Aufbau und die Beziehungen der Objekte in der Datenbank.

2.4.5.1 Provisioning Service

Das Domainmodell des Provisioning Service (Abbildung 2.20) zeigt die Objekte Challenge, Reader und Projekt, welche untereinander in Beziehung stehen.
Startet ein Reader, dann löst er als erstes eine Challenge beim Provisioning Service, die er erfolgreich lösen muss bevor er sich mit einer eindeutigen Identifikationsnummer und mit seiner MAC-Adresse registriert wird. Solange der Reader noch keinem Projekt zugewiesen ist, wird er sich nicht mit einem Business Applikation Server eines Kunden verbinden. Die URL eines solchen Projektservers wird zusammen mit den projektspezifischen Daten wie Kunden- und Projektname in der Datenbank erfasst.

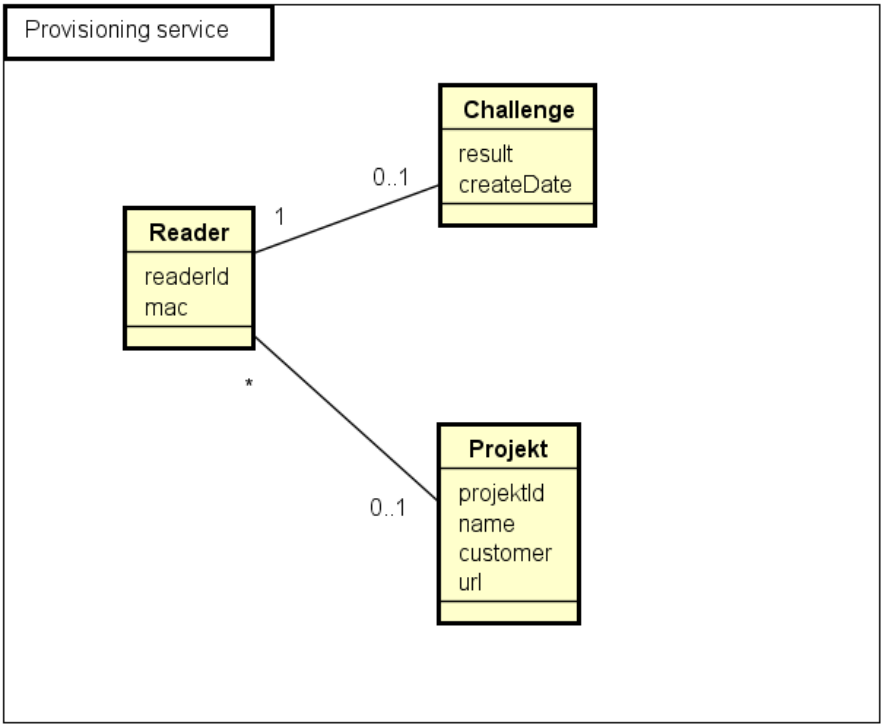


Abbildung 2.20: Domainmodell des Provisioning Service

2.4.5.2 Authentifizierungsmodul

Wie das Domainmodell des Authentifizierungsmoduls (Abbildung 2.21) zeigt muss die Zuweisung zwischen Karte (Card) und Benutzer (User) mit dem Beziehungsverhältnis zwischen Terminal und Reader über den Job miteinander verknüpft sein, damit eine erfolgreiche Authentifizierung statt finden kann.

Eine im System registrierte Karte wird einem Benutzer zugewiesen, welcher über die bestehende Benutzerverwaltung der Business Applikation erstellt wurde. Der Benutzer wählt über einen Terminal einen Reader aus, was als Beziehung geseichert wird. Die Kommunikation zwischen Terminal und Reader erfolgt über Jobs, wodurch die Beziehung zwischen Benutzer und Karte wieder ins Spiel kommt, damit die Authentifizierung oder die Initialisierung der Karte funktioniert.

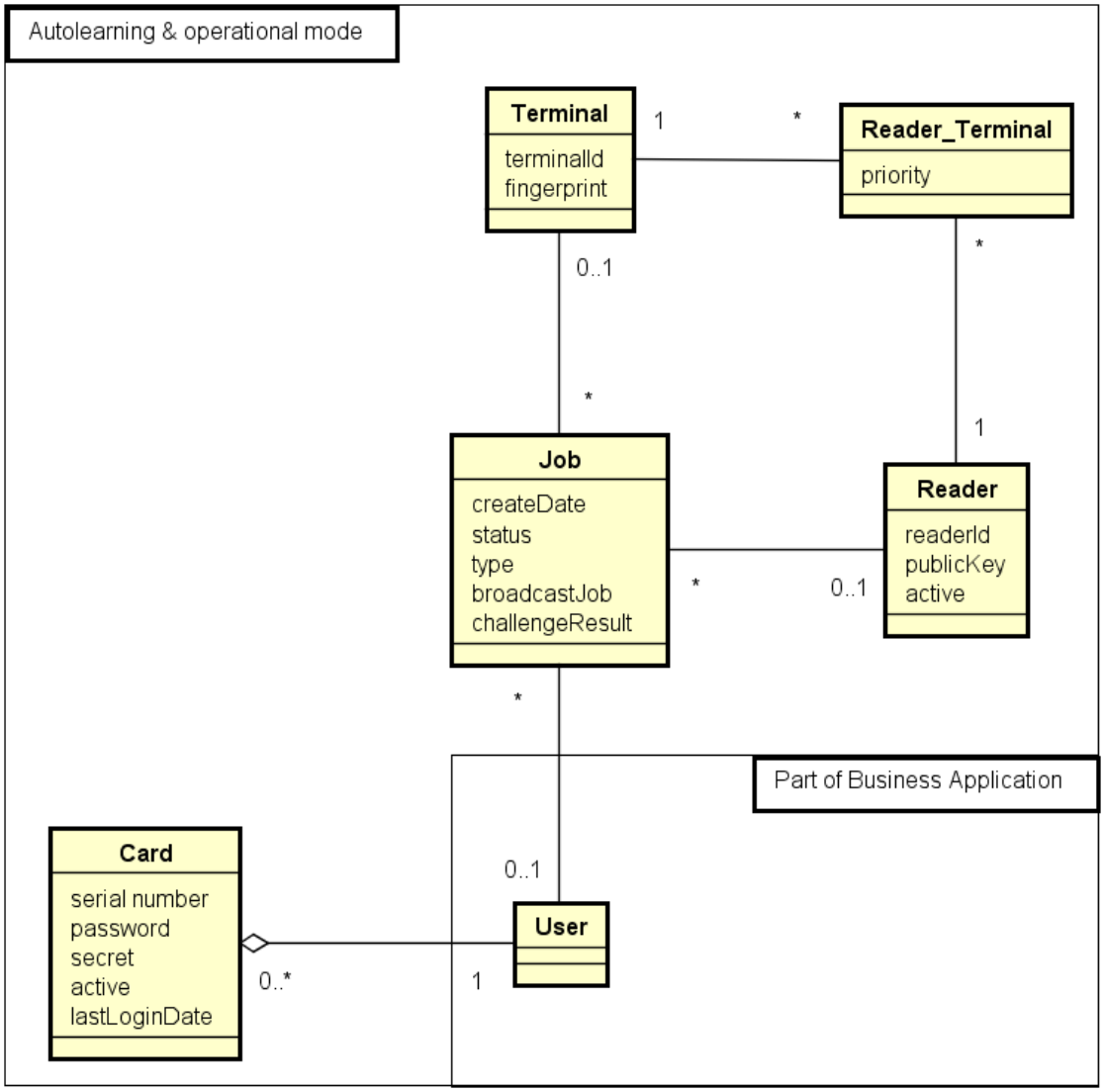


Abbildung 2.21: Domainmodell des Authentifizierungsmoduls

Der Job ist das zentrale Objekt dieses Domainmodells, weshalb er folgend noch etwas genauer beschrieben wird.
Jobs werden indirekt immer durch einen Benutzer oder Administrator ausgelöst und sind nur für eine gewisse Zeitdauer gültig. Alle Jobs werden nach wenigen Sekunden automatisch wieder gelöscht. Ein Update-Job wird speziell behandelt und erst gelöscht, wenn der Reader ihm abgeholt hat. Die Jobs werden nacheinander in die Datenbank-Tabelle eingetragen und nach Eingangsdatum abgearbeitet.

2.4.6 Klassendiagramme

2.4.6.1 Reader

Das Klassendiagramm des Readers (Abbildung 2.22) zeigt klar die Verantwortlichkeiten der einzelnen Komponenten auf.
In der Access Schicht existieren die Klassen Config für den Zugriff auf lokale Konfigurationsdateien, Ntag für den Zugriff auf NFC Karten mit der NTAG-Technologie und Network für die Netzwerkkommunikation. Die Service Schicht beherbergt die Klassen SymmetricEncryptionHandler und AsymmetricEncryptionHandler für die Sicherstellung und Handhabung der jeweiligen Verschlüsselungstechniken. In der Schicht Application befindet sich die Klasse ReaderController über welche das ganze Verhalten des Readers kontrolliert wird.

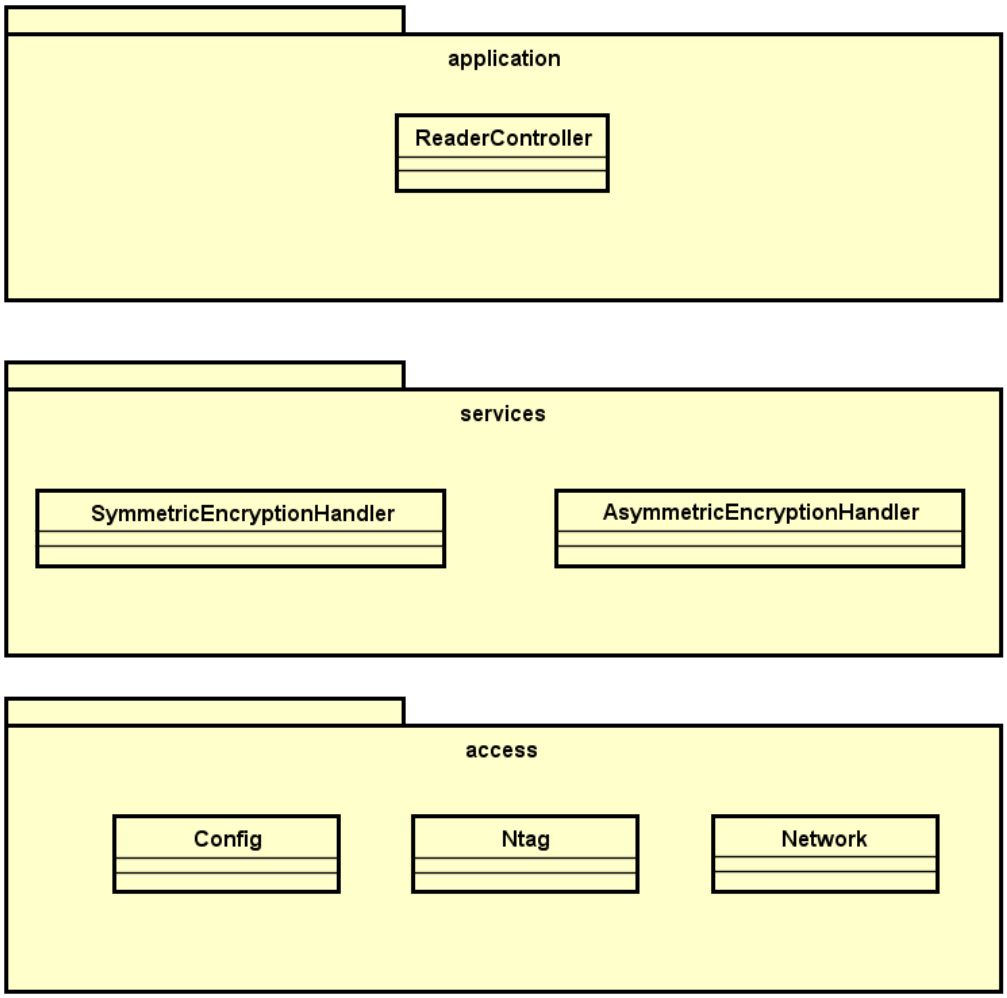


Abbildung 2.22: Klassendiagramm des Readers

2.4.6.2 Provisioning Service

Im Klassendiagramm des Provisioning Service (Abbildung 2.23) sind wiederum die selben Pakete ersichtlich wie bereits im Schichtenmodell.
In der Persistence Schicht greift nur die Klasse DbAccess auf die MySQL-Datenbank zu. Die Klassen ChallengeAccess, ReaderAccess und ProjectAccess erben alle von der Klasse DbAccess und bieten Objekt spezifisch vorbereitete Funktionen für sicherheitstechnisch gekapselte Datenbankabfragen an. Im Paket Business sind alle diese Objektklassen abgebildet und stellen wiederum objektbezogene Funktionen zur Verfügung. Ausserdem liegt hier die Klasse SymmetricEncryptionHandler, welche alle Abläufe zur symmetrischen Verschlüsselung behandelt. Das Paket Service beinhaltet keine Klassen sondern stellt Endpunkte für die Kommunikation vom Reader bereit. Jede Endpunkt-Datei steht für einen anderen Schritt im Kommunikationsablauf zwischen Provisioning Service und Reader.

Zuordnung des Endpunktes zu einem Kommunikationsschritt:

- **provisioning.php**
  Start ein Readers meldet er sich an diesem Endpunkt und erhält eine Challenge vom Provisioning Service.

- **register.php**
  Über diesen Endpunkt erhält der Provisioning Service die Antwort der gelösten Challenge des Readers und liefert dem Reader bei erfolgreicher Prüfung die entsprechende Konfiguration aus.

- **instruction.php**
  Besitzt ein Reader keine gültige Konfiguration für einen Business Applikation Server, dann fragt er regelmässig über diesen Endpunkt den Provisioning Service nach einer gültigen Konfiguration.

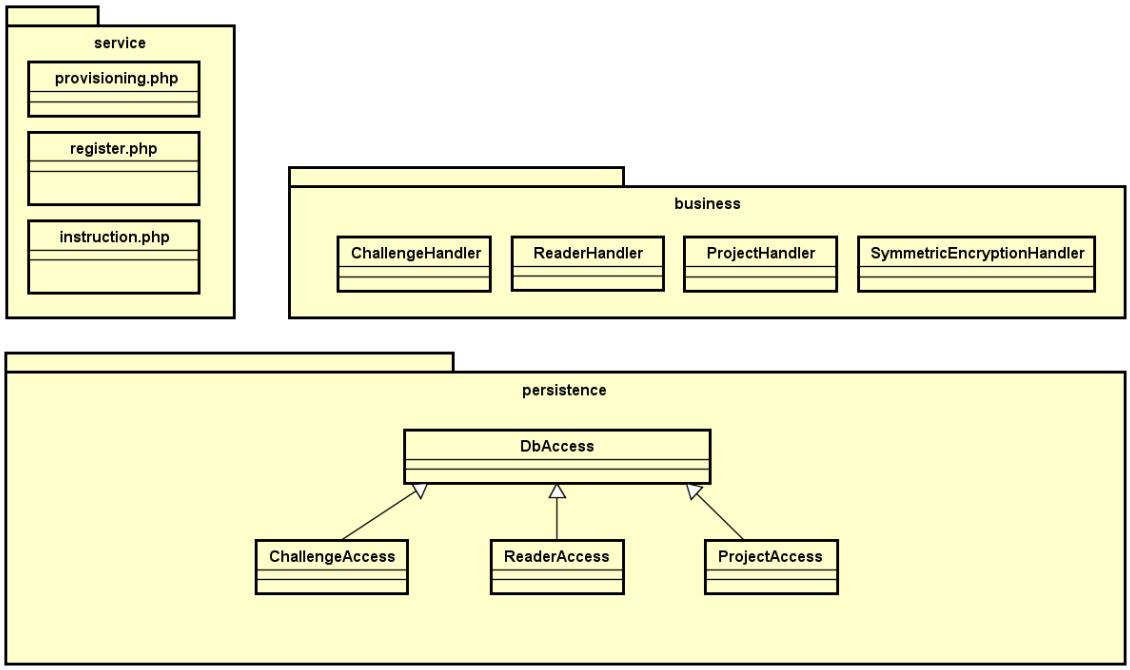


Abbildung 2.23: Klassendiagramm des Provisioning Service

2.4.6.3 Authentifizierungsmodul

Das Authentifizierungsmodul besitzt kein Schichtenmodell (Kapitel 2.4.4.3), aber enthält trotzdem die Klassen Reader und AsymmetricEncryptionHandler (Abbildung 2.24). Dabei wird die Klasse Reader verwendet um ein Softwareobjekt anzulegen mit welchem dann die Funktionen der asymmetrischen Verschlüsselung verwendet werden können.

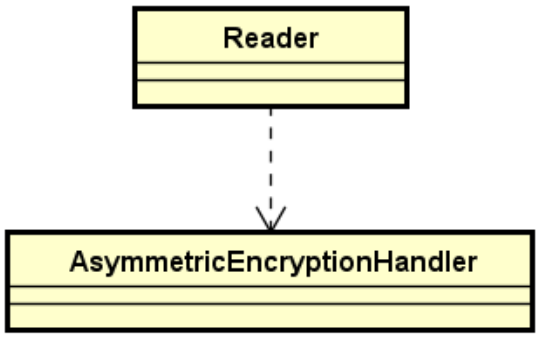


Abbildung 2.24: Klassendiagramm des Authentifizierungsmoduls

2.5 Technologie

Die zur Authentifizierung verwendeten Karten kommunizieren über NFC. Diese Funktechnologie hat den Vorteil, dass die Karten nicht in ein Lesegerät eingeschoben werden müssen, sondern nur in die Nähe gehalten werden können, um die Daten auszulesen. Diese Technologie wird in den folgenden Kapiteln im Detail erläutert.

2.5.1 RFID

RFID steht für Radio-frequency identification und bezeichnet die Identifizierung von Objekten über Radiowellen. Ein RFID System besteht aus einem Lesegerät und einem Tag. Das Lesegerät ist eine aktive Komponente mit einer Antenne, über welche die elektromagnetischen Signale gesendet und empfangen werden können. Ein Tag ist ein kleiner Mikrocontroller, ebenfalls mit einer Antenne. Bei den Tags unterscheidet man zwischen aktiven und passiven Tags. Aktive Tags besitzen eine eigene Stromversorgung und können dadurch eine sehr viel höhere Reichweite aufweisen als passive Tags, welche die benötigte Energie aus dem vom Lesegerät gesendeten Signal beziehen. Sowohl aktive als auch passive Tags können als kleine Mikrocontroller mit Speicher betrachtet werden. Dadurch ist es je nach Modell möglich, nur auf den Speicherbereich zuzugreifen oder ganze kryptographische Operationen auf den Kontrollern auszuführen.

RFID arbeitet auf drei unterschiedlichen Frequenzbändern, die sich auch in der Reichweite und Datenübertragungsraten unterscheiden. Während kleinere Frequenzen (120–150 kHz Low-Frequency, 13.56 MHz High-Frequency) eine Reichweite von bis zu einem Meter haben und die Komponenten relativ günstig erworben werden können, haben höhere Frequenzen (865-928 MHz Ultra-High-Frequency) Reichweiten von bis zu 12m und eine höhere Datenübertragungsrate. Vor allem bei passiven Tags ist die Reichweite auch abhängig von der durch das Lesegerät generierten Feldstärke.

2.5.2 NFC

Eine auf RFID aufbauende Technologie ist NFC (Near-Field-Communication). NFC nutzt das durch RFID spezifizierte High-Frequency Band (13.56 MHz) und hat eine Reichweite von bis zu 10 cm mit passiven Tags. Ein grosser Vorteil von NFC ist, dass es inzwischen sehr verbreitet ist und von fast allen modernen Smartphones unterstützt wird. Durch diese Verbreitung sind auch die Kosten für die Komponenten relativ niedrig.

Da diese Technologie für eine Authentifizierungslösung verwendet werden soll, war ein wichtiger Punkt die Vervielfältigung zu Unterbinden. Im NFC Standard ist keine Art von Schutz für die auf der Karte gespeicherten Daten vorgesehen. Daher wurden Karten mit erweiterten Funktionalitäten gesucht und wurden schiesslich beim Hersteller NXP gefunden.

2.5.3 NXP

Als der grösste Halbleiter Hersteller in Europa hat NXP auch eigene Tags, die den NFC Standard um Funktionalitäten wie einen Passwortschutz erweitern. Da auch das verwendete NFC-Hardware Module einen Chip von NXP besitzt, ist sichergestellt, dass auch die zusätzlichen nicht im NFC Standard enthaltenen Funktionalitäten wie der Passwortschutz verwendet werden können. Da das vom Auftraggeber vorgegebene NFC Hardwaremodul auch von NXP ist, wurde relativ früh bei NXP nach kompatiblen Karten gesucht, welche die Voraussetzungern erfüllen.

2.5.4 NTAG

NTAG ist eine Produktfamilie von NXP, welche neben dem NFC Standard auch einen Passwortschutz in die Tags eingebaut hat. Diese NTAGs sind in diversen Speichergrössen erhältlich.

2.5.4.1 Funktionsweise

NTAGs besitzen einen Mikrocontroller, der die Signale des Lesegerätes über eine Antenne empfängt und interpretiert. Der Mikrocontroller regelt auch den Speicherzugriff, das heisst es wird nie direkt auf den Speicher zugegriffen. So kann auch sichergestellt werden, dass auf geschützte Speicherbereiche nur mit dem richtigen Passwort zugegriffen werden kann. Der Ablauf eines Zugriffs wird anhand des folgenden Zustandsdiagramms erklärt:

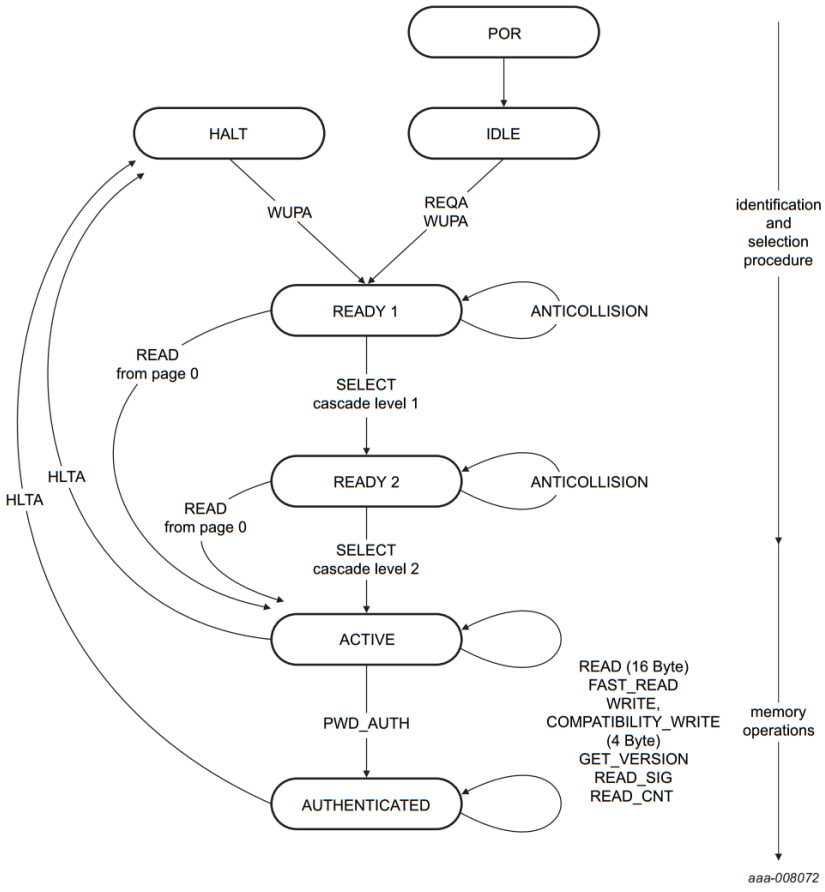


Abbildung 2.25: NTAG Zustandsdiagramm

Nachdem der Tag in Reichweite des Lesegerätes ist und dadurch mit Energie versorgt wird, befindet er sich im Zustand **IDLE**. Um von diesem Zustand nach **ACTIVE** zu kommen, muss in zwei Schritten die Unique ID (UID) der Karte ausgelesen werden. Durch diesen Prozess wird auch sichergestellt, das diese nun identifizierte Karte eindeutig angesprochen werden kann, da es auch möglich ist, dass sich mehrere Karten gleichzeitig in Reichweite des Lesergerätes befinden. Nach dem die beiden **READY** Zustände durchlaufen wurden, befindet sich die Karte im Zustand **ACTIVE**. In diesem Zustand ist es möglich, auf alle ungeschützten Speicherbereiche der Karte zuzugreifen und zu schreiben. Auf die geschützten Bereiche kann nur im **AUTHENTICATED** Zustand zugegriffen werden. Durch eine Authentifizierung mit dem PWD_AUTH Befehl und dem Passwort kann in diesen Zustand gewechselt werden. In den Zuständen **ACTIVE** oder **AUTHENTICATED** kann beliebig oft auf den Speicher zugegriffen und geschrieben werden, solange kein Fehler auftritt (z.B. unautorisierter Zugriff auf einen geschützten Bereich). Tritt in einem der beiden Zustände ein Fehler auf, wird in den **HALT** Zustand gewechselt. Dieser Zustand ist sehr ähnlich wie der **IDLE** und kann auch mit dem Auslesen der UID verlassen werden.

2.5.4.2 Passwortschutz

Bereiche der Karte können durch ein 32-Bit Passwort geschützt werden. Dazu wird in den Konfigurationspages (CFG0, CFG1, siehe Abbildung 2.28) die Startadresse angegeben, ab welcher der Speicher geschützt werden soll. Standardmässig wird der definierte Bereich schreib-, aber nicht lesegeschützt. Die Art des Schutzes kann ebenfalls in den Konfigurationspages angegeben werden.
Zum Schutz gegen Brute-Force Attacken kann optional ein Passwortlimit gesetzt werden. Nachdem dieses Limit an falschen Passworteingaben erreicht wurde, wird der geschützte Bereich gesperrt und es kann auch mit dem korrekten Passwort nicht mehr darauf zugegriffen werden. Der Zähler für falsche Passworteingaben wird nach einer erfolgreichen Authentifizierung wieder zurückgesetzt. Der Passwortschutz ist erst nachdem der ACTIVE Zustand verlassen wurde wirksam.

2.5.4.3 Speicherlayout

Die gesamte Karte wird durch das Schreiben bestimmter Speicherbereiche konfiguriert. Die Abbildung2.26 zeigt das Speicherlayout eines NTAG 216 auf. Andere NTAGs der selben Familie (NTAG 210 bis NTAG 216) unterscheiden sich im Speicherlayout nur dadurch, dass das User memory kleiner ist und sich deshalb die Speicheradressen der Konfigurationsseiten etwas nach unten verschieben. Die Speicher der NTAGs sind in Pages von jeweils 4 Bytes unterteilt.

Page Adr		Byte number within a page				
Dec	Hex	0	1	2	3	Description
0	0h	serial number				Manufacturer data and static lock bytes
1	1h	serial number				
2	2h	serial number	internal	lock bytes	lock bytes	
3	3h	Capability Container (CC)				Capability Container
4	4h	user memory				User memory pages
5	5h					
...	...					
224	E0h					
225	E1h					
226	E2h	dynamic lock bytes			RFUI	Dynamic lock bytes
227	E3h	CFG 0				Configuration pages
228	E4h	CFG 1				
229	E5h	PWD				
230	E6h	PACK		RFUI		

Abbildung 2.26: NTAG Speicherlayout

Serial number
Die ersten zwei Pages enthalten die Seriennummer des Tags plus Prüfbyte.

Capability Container
Im Capability Container kann unter anderem ausgelesen werden, um welchen Tagtypen es sich handelt. Die genauen Bezeichnungen der Typen sind im Kapitel 2.5.4.4 aufgelistet.

User memory
Das User memory ist grundsätzlich frei les- und beschreibbar.

CFG
Die zwei CFG Pages enthalten diverse Konfigurationsparameter (Abbildung 2.28). Hier kann unter anderem definiert werden, ab welcher Adresse der Speicherbereich geschützt werden soll. Zusätzlich zur Art des Schutzes (nur schreibgeschützt oder schreib- und lesegeschützt) kann auch die maximale Anzahl Passwortversuche definiert werden.

PWD
In der PWD Page kann das 32-Bit Passwort gesetzt werden. Dieser Bereich kann nur geschrieben und nicht gelesen werden.

2.5.4.4 Capability Container

Damit ein Lesegerät erkennen kann, was für einen Typ von Tag sich in Reichweite befindet, ist an einer bestimmten Adresse im Tag die unterstützte NFC Spezifikation vermerkt. Dieser Bereich wird Capability Container genannt und ist in der dritten Page auf dem Tag. Für die Unterscheidung der Tags wird das dritte Byte verwendet. Der Inhalt des Bytes ist in der Abbildung 2.27 dargestellt. Zusätzliche Informationen sind in der NTAG Dokumentation vom NXP [Ref. 9, Seite 16] oder direkt in der NFC Spezifikation [Ref. 3, Seite 20] verfügbar.

Table 4. NDEF memory size

IC	Value in byte 2	NDEF memory size
NTAG213	12h	144 byte
NTAG215	3Eh	496 byte
NTAG216	6Dh	872 byte

Abbildung 2.27: NTAG Typen

2.5.4.5 Konfigurationspages

In diesem Kapitel werden die für diese Arbeit relevanten Konfigurationsparameter näher erläutert.

Table 8. Configuration Pages

Page Address[1]		Byte number			
Dec	Hex	0	1	2	3
41/131/227	29h/83h/E3h	MIRROR	RFUI	MIRROR_PAGE	AUTH0
42/132/228	2Ah/84h/E4h	ACCESS	RFUI	RFUI	RFUI
43/133/229	2Bh/85h/E5h	PWD			
44/134/230	2Ch/86h/E6h	PACK		RFUI	RFUI

[1] Page address for resp. NTAG213/NTAG215/NTAG216

Table 9. MIRROR configuration byte

Bit number							
7	6	5	4	3	2	1	0
MIRROR_CONF		MIRROR_BYTE		RFUI	STRG_MOD_EN	RFUI	

Table 10. ACCESS configuration byte

Bit number							
7	6	5	4	3	2	1	0
PROT	CFGLCK	RFUI	NFC_CNT_EN	NFC_CNT_PWD_PROT	AUTHLIM		

Abbildung 2.28: NTAG Konfigurationsbereich

AUTH0
Das Byte AUTH0 definiert die Startadresse des geschützten Speicherbereiches. Um den Zugriffsschutz zu deaktivieren, wird dieses Byte auf eine Adresse, die grösser als der Speicherbereich des Tags ist, gesetzt.

ACCESS
Im ACCESS Byte werden mehrere Parameter konfiguriert. Für diese Arbeit sind nur die zwei Parameter PROT und AUTHLIM relevant. Die Positionen der einzelnen Parameter sind in der Tabelle 10 in Abbildung 2.28 dargestellt.

- **PROT** Durch dieses Bit wird festgelegt, ob der definierte Bereich nur schreibgeschützt oder schreib- und lesegeschützt werden soll (0 : schreibgeschützt, 1 : schreib- und lesegeschützt).
- **AUTHLIM** In diesen drei Bits kann die Anzahl erfolgloser Passwortversuche definiert werden (1-7 Versuche oder 0 für keine Einschränkung).

2.6 Prototyp

Dieses Kapitel befasst sich mit dem Prototypen und den Überlegungen und Erkenntnissen, die durch diesen gewonnen wurden.
Mit dem Prototypen wurden verschiedene Ziele verfolgt. Einerseits sollten wichtige Erfahrungen auf einer zuvor relativ unbekannten Technologie gesammelt werden, andererseits sollte dieser auch ein Proof of Concept der gewählten Technologie sein.

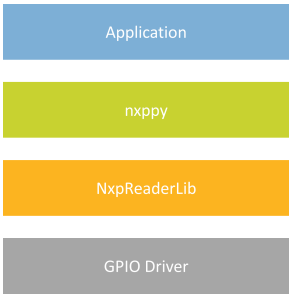


Abbildung 2.29: Applikationsschichten

Für die Kommunikation mit dem NFC-Modul, welches über die GPIO Pins verbunden ist, wird eine von NXP angebotene Bibliothek verwendet (NxpReaderLib). Diese bietet diverse Funktionen für die Interaktion mit verschiedenen Tag Typen an. Um die Verwendung der Bibliothek zu vereinfachen wird Nxppy verwendet.

2.6.1 Nxppy

Nxppy[14] ist ein in Python geschriebener Wrapper, der für Funktionen der NXP Bibliothek jeweils Python Schnittstellen zur Verfügung stellt. Nxppy ist kein Produkt einer Firma, sondern wird durch Entwickler, welche Nxppy nutzen und eine neue Funktionalität benötigen weiterentwickelt. Da die Authentifizierung auf dem Tag in Nxppy noch nicht implementiert war, wurde diese im Rahmen des Prototyps entwickelt und dem offiziellen Nxppy Repository hinzugefügt. Mit Hilfe dieses Wrappers konnten nun mit wenig Aufwand Pythonskripte für die diversen Operationen erstellt werden. Diese Skripte werden dann von der Readersoftware verwendet, um mit der Karte zu interagieren.

2.6.2 Beispielskript

Diesers Kapitel beschreibt die einzelnen Python Skripts, welche im Rahmen des ersten Prototypen entstanden sind und später auch für die Readerimplementation verwendet wurden.

UID lesen

Dieses Skript gibt nur die UID der Karte zurück. Es wird im Reader verwendet um zu überprüfen ob ein Tag in der Nähe ist.

```
pi@raspberrypi:~/Desktop/sa/src/access/nxppy $ python readUid.py
045DF7D2245580
```

Abbildung 2.30: UID Lesen

Daten schreiben

Mit den Parametern wird definiert ab welcher Adresse was für Daten auf den Tag geschrieben werden.

```
pi@raspberrypi:~/Desktop/sa/src/access/nxppy $ python write.py 08 12345678
UID: 0450F8D2245580
```

Abbildung 2.31: Daten schreiben

Passwortschutz aktivieren

Im Hintergrund werden in mehreren Schreiboperationen das Passwort und die entsprechenden Konfigurationsbits gesetzt. Als Parameter wird der Start des Konfigurationsbereiches der Karte, das Passwort und die Adresse ab welcher der Speicherbereich geschptzt werden soll, angegeben.

```
pi@raspberrypi:~/Desktop/sa/src/access/nxppy $ python enableProtection.py 29 12345678 08
```

Abbildung 2.32: Passwortschutz aktivieren

Daten lesen

Beim Lesen von Daten muss mindestens die Adresse, ab welcher gelesen werden soll, angegeben werden. Zusätzlich kann noch die Länge der zu lesenden Daten und ein Passwort angegeben werden. Ohne passwort kann in dieser Konfiguration nicht von einem geschützten Speicherbereich gelesen werden.

```
pi@raspberrypi:~/Desktop/sa/src/access/nxppy $ python read.py 08 4 12345678
UID: 0450F8D2245580
12 34 56 78
```

Abbildung 2.33: Daten lesen

Passwortschutz deaktivieren

Um den Passwortschutz zu deaktivieren wird wiederum der Start des Konfigurationsbereiches und das Passwort benötigt.

```
pi@raspberrypi:~/Desktop/sa/src/access/nxppy $ python disableProtection.py 29 12345678
```

Abbildung 2.34: Passwortschutz deaktivieren

2.7 Implementierte Software

2.7.1 Reader

Der Reader bildet zusammen mit dem Authentifizierungsmodul die zwei zentralen Komponenten. Seine Hauptaufgabe ist, die Karten der Benutzer zu verifizieren. Nach dem Aufstarten prüft der Reader beim Provisioningserver, ob seine Konfiguration noch aktuell ist oder holt sich allenfalls eine Neue. Danach meldet er sich beim Authentifizierungsmodul und hinterlegt seinen öffentlichen Schlüssel. Seinen öffentlichen und privaten Schlüssel generiert der Reader, falls noch keiner in seiner Konfiguration gespeichert ist. Durch diesen öffentlichen Schlüssel kann der Server später Anfragen des Readers verifizieren. Alle aktiven Reader im System pollen den Server nach Jobs, das heisst sie senden in Intervallen Anfragen an den Server und erhalten als Antwort einen Job oder die Aufforderung, später nochmals nachzufragen. Die Reader pollen solange, bis sie einen Job haben. Nach Beendigung des Jobs beginnen sie erneut zu pollen.

2.7.1.1 Reader Jobs

Ein Job wird durch eine Loginanfrage oder eine Administratoraktion serverseitig erstellt und dem zugewiesenen Reader bei der nächsten Nachfrage mitgeteilt. Es gibt zwei unterschiedliche Arten wie Jobs verteilt werden. In den meisten Fällen ist ein Job direkt einem Reader zugeordnet und wird nur diesem Reader gesendet. Im Gegensatz dazu wird für das Autolearning ein spezieller Job erstellt. Dieser Broadcast-Job ist keinem bestimmten Reader zugeordnet und wird von allen Readern abgeholt und gleichzeitig bearbeitet. Damit verschiedene Anweisungen an die Reader gegeben werden können, existieren drei verschiedene Jobtypen. Diese werden im Folgenden erläutert:

Authentifizieren
Bei einer Authentifizierung wird zunächst versucht, mit einer Karte eine Verbindung aufzubauen. Die dabei ausgelesene Uid wird verwendet, um beim Authentifizierungsmodul das zur Karte passende Passwort zu holen. Dabei verschlüsselt und signiert der Reader die Uid. Damit kann serverseitig festgestellt werden, ob die Anfrage von einem legitimen Reader kommt. Als Antwort bekommt der Reader das Passwort und eine Challenge. Mit dem Passwort kann das Secret von der Karte ausgelesen werden, welches für das Lösen der Challenge verwendet wird. Das Resultat wird an den Server zurückgesendet und dort überprüft. Bei einem Broadcastjob werden die meisten Reader keine Karte lesen können. Deshalb gibt es einen Timeout von 7 Sekunden. Nach Ablauf dieses Timers beginnt der Reader wieder mit dem Polling.

Konfiguration aktualisieren
Mit diesem Job werden zugleich auch die zu aktualisierenden Parameter der Konfiguration übertragen. Die im Job definierten Felder werden in der lokalen Konfiguration überschrieben und das Ablaufdatum der Konfiguration wird aktualisiert.

Karte initialisieren

Der Job enthält die zwei auf dem Server generierten Werte Passwort und Secret. Zuerst wird das Secret auf die Karte geschrieben und anschliessend wird dieser Bereich mit dem Passwort geschützt. Da die zu initialisierende Karte zuvor nicht erfasst werden muss, wird nun noch die Uid ausgelesen und an den Server übermittelt.

2.7.1.2 Detailierter Anmeldeprozess

Bei der Authentifizierung eines Benutzers laufen verschiedene Prozesse ab, welche in diesem Abschnitt erklärt werden. Auf den Karten ist durch ein Passwort geschützt ein Secret gespeichert. Das Secret ist ein zufälliger 32-Bit Wert, welcher bei der Initialisierung der Karte generiert wird. Dieses Secret wird verwendet, um zu überprüfen, ob es sich um eine vom System ausgestellte Karte handelt. Da das Passwort, welches verwendet wird, um das Secret auszulesen, auf jeder Karte unterschiedlich ist, liest der Reader zuerst nur die UID der Karte aus und verlangt dann vom Authentifizierungsmodul das dazugehörige Passwort. Damit nicht jeder so das Passwort der Karte anfordern kann, signiert der Reader seine Anfrage mit seinem privaten Schlüssel. Auf dem Server wird die Signatur verifiziert und überprüft, ob dem Reader ein Auftrag erteilt worden ist eine Karte zu lesen. Sind alle Überprüfungen erfolgreich, wird dem Reader verschlüsselt das Passwort der Karte sowie eine Challange gesendet. Durch diese Challange werden Replay-Attacken vorgebeugt. Mit dem Passwort kann der Reader nun auf die Karte zugreifen und mit dem Secret die Challenge lösen. Das Resultat wird wiederum signiert und verschlüsselt an den Server gesendet, welcher nach einer erfolgreichen Überprüfung den Benutzer am System anmeldet und dem Reader eine Statusmeldung zurückgibt. Dadurch ist für den Reader der Vorgang abgeschlossen und er beginnt wieder mit dem Polling.

2.7.1.3 Konfiguration

Die Konfiguration des Readers ist in zwei PHP-Dateien aufgeteilt, welche beim Start des Readers geladen werden. Die eine Datei enthält alle Parameter für den Provisionierungsprozess, während die andere Datei alle Parameter enthält, welche im Betriebsmodus benötigt werden. Die Konfiguration enthält ein Ablaufdatum, das immer wenn der Reader eine Konfiguration vom Gordo erhält, sei dies beim Aufstarten oder durch ein Update, aktualisiert wird. Falls der Reader keine Updates mehr bekommt, zum Beispiel weil er im Gordo deaktiviert wurde, und deshalb das Ablaufdatum überschritten wird, wird die Projektspezifische Konfiguration gelöscht. Im folgenden sind die Konfigurationsparameter des Readers im einzelnen beschrieben:

```
<?php return array (
  'RSA_CONFIG' => array (
    'digest_alg' => 'sha512',
    'private_key_bits' => 4096,
    'private_key_type' => 0,
  ),
```

Hier wird der verwendete Hashalgorythmus sowie die Schlüssellänge definiert.

```
  'ENCRYPT_BLOCK_SIZE' => 400,
  'DECRYPT_BLOCK_SIZE' => 512,
  'HASH_ALGORITHM' => 'sha256',
```

Diese Blocklängen werden benötigt, um grössere Daten korrekt in einzelne Blöcke aufzuteilen, bevor sie ent- oder verschlüsselt werden.

```
  '12'=>'NTAG213',
  '6d'=>'NTAG216',
  '6f'=>'NTAG216',

  'NTAG213' => array (
    'addr_secret' => '10',
    'addr_protection_config' => '29',
    'length_secret' => 32,
  ),

  'NTAG216' => array (
    'addr_secret' => '10',
    'addr_protection_config' => 'E3',
    'length_secret' => 32,
  ),
```

Da der Konfigurationsbereich auf dem NTAG hinter dem User memory liegt und das User memory je nach NTAG Typ unterschiedlich gross ist, kann hier für jeden unterstützten Typen die Adresse der Konfiguration definiert werden. Dieser Bereich ist so gegliedert, dass im ersten Abschnitt der Wert, welcher aus dem Capability Container (Byte 2) der Karte ausgelesen wird, einem NTAG Typen zugeordnet wird. So ist es möglich, dass mehrere Typen, die sich nur durch ihre Funktionalität, nicht aber durch ihre Speichergrösse unterscheiden, der gleichen NTAG Konfiguration zugeordnet werden können.
In der jeweiligen NTAG Konfiguration wird definiert ab welcher Adresse das Secret geschrieben wird (dies ist gleichzeitig auch die Adresse ab welcher der Speicher passwortgeschützt wird), wo der Konfigurationsbereich der Karte beginnt und wie lange das Secret ist.
Werden NTAGs mit einem kleineren User memory als 80 Bytes eingesetzt, muss die Startadresse des Secrets in der Konfiguration für diesen Typ angepasst werden. Da das Secret 32 Bytes lang ist, muss sichergestellt werden, dass ab der Startadresse des Secrets ('addr_secret') mindestens Platz ist für diese 32 Bytes. Ansonsten kann die Karte nicht für die Authentifizierung verwendet werden.

```
  'NTAG_READ_TIMEOUT' => 7,
  'NTAG_READ_INTERVAL_MS' => 100,
);
```

Hier wird angegeben, wie lange und in welchem Intervall versucht werden sollte, einen Tag in der Nähe des Readers zu detektieren.

```
<?php return array (
  'METHOD' => 'AES-256-CBC',
  'AES_KEY' => 'E8rGB/bJxRt9p+XaGggwatkEzqMoUHrtORUuwXpW0Gc=',
  'PROVISIONING_SERVER' => 'https://nfc01.dxb.ch/ProvisioningService
   ',
);
```

Dies ist die Konfiguration des Provisioning Prozesses. In dieser wird der AES Algorithmus und Schlüssel gespeichert, sowie die Adresse des Provisioning Servers.

2.7.1.4 Unterstützte Kartentypen

Der Reader wurde mit den Kartentypen NTAG 213, NTAG 216 und NTAG 216F getestet. Es können auch andere Karten dieser Familie (NTAG 210 - NTAG 216) verwendet werden, da alle bis auf den NTAG 210 Micro die benötigte Funktionalität unterstützen. Eine Übersicht dieser Familie ist auf der NXP Website[8] ersichtlich.

2.7.1.5 Verlust eines Readers

Beim Verlust eines Readers kann dieser im Gordo und auf dem Provisioning Service deaktiviert werden. Da der Server nur Anfragen von aktiven Readern beantwortet, werden deaktivierte Reader nicht mehr bedient. Somit kann ein deaktivierter Reader auch über eine korrekte Anfrage kein Kartenpasswort vom Gordo erhalten.

2.7.2 Authentifizierungsmodul

Das neue Authentifizierungsmodul kann sehr einfach in die bestehende Applikation integriert werden. Um die Integration möglichst einfach zu halten, wurden alle nicht ins GUI der bestehenden Applikation integrierten Dateien in separaten Ordnern angelegt.
Dieses Modul antwortet im Hintergrund auf die Anfragen der Reader und Terminals. Im Gordo stellt es eine Oberfläche für die Verwaltung der Reader und Karten zur Verfügung. Diese beiden Teile greifen auf die selbe Datenbank zu, sind aber ansonsten voneinander unabhängig und könnten auch auf verschiedenen Servern laufen.

2.7.2.1 Backend

Das Backend registriert neue Reader und verwaltet die Jobs im System. So wird bei einer Anmeldung an einem Terminal diese Anfrage entgegengenommen und ein Job für den ausgewählten Reader erstellt. Ist kein Reader ausgewählt, wird ein Broadcastjob erstellt und an alle Reader verteilt. Andere Aufgaben betreffen das Ausliefern von Kartenpasswörtern auf korrekte Anfragen und schlussendlich das Authentifizieren des Benutzers, falls das Secret der Karte erfolgreich überprüft werden konnte.
Von den Terminals werden für Loginanfragen Jobs erstellt und die Anfragen nach dem Loginstatus beantwortet. Für jedes Terminal wird die Liste der zuletzt verwendeten Reader verwaltet und an das Terminal gesendet.

2.7.2.2 GUI

Für den Administrator wird im Gordo eine Übersicht über alle registrierten Reader und Karten dargestellt. Zwei weitere Interfaces stellen die Initialisierung von Karte sowie das Aktualisieren der Konfiguration zur Verfügung.

Karte (re-)initialisieren

Karten können unter dem Menupunkt «Karte initialisieren» einem Benutzer zugeordnet werden. Dazu werden der Benutzer und der Reader, über welchen die Karte initialisiert werden soll, ausgewählt. Nun kann die Karte auf den Reader gelegt werden und die Initialisierung gestartet werden. Über diesen Prozess können auch bereits im System erfasste Karten einem neuen Benutzer zugeordnet werden. Im Hintergrund wird dabei ein Initialisierungsjob erstellt und dem Reader das zu verwendende, neu generierte Passwort und Secret mitgeteilt.

Abbildung 2.35: Karte initialisieren

Konfiguration aktualisieren

Ein Update der URL des Provisioning Servers oder des AES Keys kann unter «Update Konfiguration» vorgenommen werden. Dies löst einen Updatevorgang für alle im System registrierten Reader aus. Damit auch Reader das Update erhalten, die zur Zeit ausgeschaltet sind, wird für jeden aktivierten Reader ein eigener Job erstellt. Diese Jobs werden entfernt, sobald der Reader diesen abgeholt hat. Diese Jobs werden im Unterschied zu den Initialisierungs- und Authentifizierungsjobs nicht automatisch nach einer fixen Ablaufzeit gelöscht, sodass sichergestellt ist, das alle Reader das Update erhalten.

Abbildung 2.36: Konfiguration der Reader aktualisieren

Reader Übersicht

Reader können unter «Reader» aktiviert beziehungsweise deaktiviert werden. Deaktivierte Reader können nicht mehr für die Authentifizierung und Initialisierung von Karten verwendet werden und werden auch nicht mehr mit Updates versorgt.

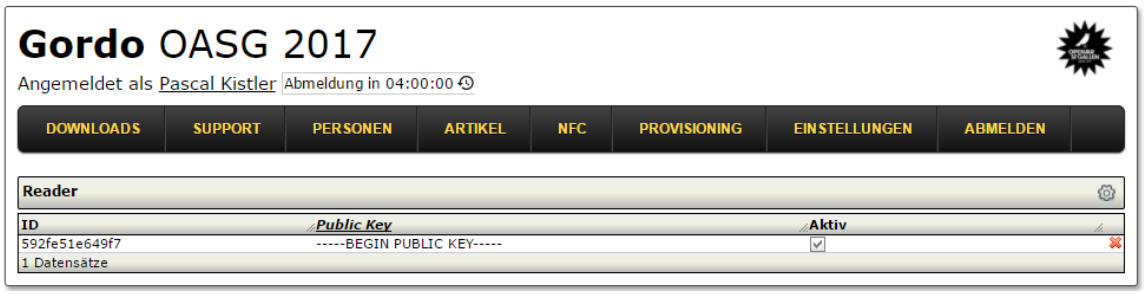


Abbildung 2.37: Readerübersicht

Kartenübersicht

In der Kartenübersicht ist ersichtlich, welchem Benutzer diese aktuell zugeordnet sind und wann sich dieser zuletzt mit der Karte angemeldet hat. Hier können Karten deaktiviert werden, um sie vom System auszuschliessen. Ist eine Karte deaktiviert, kann mit ihr keine erfolgreiche Authentifizierung mehr vorgenommen werden. Deaktivierte Karten können wiederverwendet werden, da sie über das System wieder aktiviert oder einem neuen Benutzer zugeordnet werden können.

Abbildung 2.38: Kartenübersicht im Gordo

2.7.3 Provisioning Service

Der Provisioning Service verwaltet alle Reader und Projekte. Hier sind alle Projekte und Reader an einer zentralen Stelle erfasst.
In der Projektverwaltung können Kunden erfasst werden, welche neu in ihren Projekten mit dem NFC-Authentifizierungsmodul arbeiten wollen. Pro Kunde kann es unterschiedliche voneinander getrennte Systeme geben, die hier verwaltet werden. So kann ein Kunde zum Beispiel ein Testsystem und ein Produktivsystem parallel betreiben.
Reader können in der Readerliste manuell mit ihrer Mac Adresse vorerfasst und bereits einem Projekt zugeordnet werden. Dadurch übernehmen diese beim ersten Start direkt die Zuordnung und melden sich beim entsprechenden Gordo. Nicht vorerfasste Reader werden automatisch erfasst und in der Liste ohne Projektzuordnung angezeigt. Nicht zugeordnete Reader warten und holen sich automatisch die Konfiguration, sobald sie einem Projekt zugeordnet wurden. Der gesamte Provisioning Prozess setzt voraus, dass die Reader das korrekte SharedSecret kennen. Ohne dieses erhalten die Reader keine Konfiguration und werden auch nicht automatisch im Provisioning Service erfasst.
Auf der Konfigurationsseite des Provisioning Service kann ein neues SharedSecret(AES Schlüssel) generiert werden. Dies wird empfohlen, falls die Möglichkeit besteht, dass der AES Key ausserhalb des Systems bekannt ist. So kann sichergestellt werden, dass nur rechtmässige Reader ins System kommen.
Wurde eine Konfiguration angepasst, so muss diese gespeichert und in allen jedem Gordo aktualisiert werden. Dies geschieht über die im Kapitel 2.7.2 beschriebene Aktualisation der Konfiguration. Über diesen Prozess erhalten alle Reader im System das neue SharedSecret.

2.7.4 Terminal

Als Terminal wird ein beliebiges Gerät mit einem Webbrowser bezeichnet. Die Terminals werden von den Kunden für den Login benutzt. Dabei spielt es keine Rolle, ob das Terminal ein normaler Computer, ein Smartphone oder ein Tablet ist.
Im Gordo wird von jedem Terminal ein Fingerabdruck gespeichert, damit ein Terminal auch nachdem die Cookies gelöscht wurden wiedererkannt werden kann. Mit Hilfe dieses Fingerabdruckes kann dann auch ein solches Terminal wiedererkannt werden und erhält die zugeordnete Readerhistory.
Der Fingerprint wird wird durch die Javascript Library Fingerprint2JS[12] generiert. Eine Liste der verwendeten Parameter ist im zitierten GitHub Repository verfügbar.

Loginablauf

Nachdem der Benutzer ein Terminal ausgewählt hat, wird er auf eine Statusseite weitergeleitet, welche ihn auffordert seine Karte auf den Reader zu legen und den aktuellen Status der Operation zeigt. Nach einer erfolgreichen Authentifizierung wird er anschliessend ins Gordo weitergeleitet. Falls die Karte des Benutzers nicht authentifiziert werden konnte oder keine Karte auf den Reader gelegt wurde, wird er nach einem Timeout automatisch wieder auf die Anmeldeseite umgeleitet.

2.8 Probleme

2.8.1 Lieferverzug der NTAG Karten

Nach der technischen Evaluierung der zu verwendenden NTAG-Typen, wurde mit dem Auftraggeber entschieden, dass das RFID Webauthentifizierungssystem für NTAG-Typen NTAG 213 und NTAG 216 ausgelegt werden soll. Da mit jedem NXP NFC Hardwaremodul EXPLORE-NFC-WW nur eine Karte mit der NTAG216F-Technologie mitgeliefert wird, wurde entschieden, dass noch weitere NTAG-Chips notwendig sind, damit alle Tests und Versuchsläufe durchgespielt werden können. In welcher Form, ob als Karte, Schlüsselanhänger oder sonst wie, die NTAGs vorliegen, ist dem Projektteam egal. Bei der Evaluierung der Technologie hat das Projektteam auch die Empfehlung für die zwei Lieferanten NFC21 GmbH (https://www.nfc-tag-shop.de/) und GoToTags (https://www.gototags.com/) ausgesprochen. Diese Empfehlungen basierten auf den angegebenen Lieferzeiten und Preisen der Webshops. Der Auftraggeber dxb gmbh bestellte Ende März die Karten beim einem anderen Lieferanten, aber hatte diese bis zur letzten Woche der Studienarbeit nicht erhalten.
Anfangs Mai wurde das Projektteam immer abhängiger davon, dass zusätzliche NTAGs zur Verfügung stehen, für die weitere Entwicklung und Tests der Software.

2.8.1.1 Lösungsvorgehen

Da die Zeit knapp wurde, um die Lieferung noch länger abzuwarten, hat sich das Projektteam am 16. Mai dazu entschlossen, die Bestellung selbst in die Hand zunehmen. Nach einer kurzen Budgetabsprache mit dem Auftraggeber bestellte das Projektteam beim Lieferanten Shop NFC (www.shopnfc.it) Karten und Armbänder mit der NFC Techonologie NTAG 213 und NTAG 216, welche 2 Tage später geliefert wurden.

2.8.2 Nicht Erreichbarkeit des Servers

Ab der 2. Constructionphase musste das Projektteam immer wieder mehrere Ausfälle des Testservers vom Auftraggeber in Kauf nehmen. Die Ursachen der Ausfälle waren zu Beginn eine kurzzeitige Serverüberlastung, darauf folgte der Wechsel der öffentlichen IP-Adresse des Servers, der Ausfall der vorgeschalteten physischen Firewall bis am Ende noch ein wiederkehrendes Problem mit dem Glasfaseranschluss dazu kam.
Durch diese Ausfälle während der Entwicklungs- und Testphase wurde das Projektteam mehrmals in ihrer geplanten Arbeit unterbrochen oder behindert.

2.8.2.1 Lösungsvorgehen

Das Projektteam hat sich immer zeitnah beim Erkennen eines Problems beim Auftraggeber gemeldet, damit dieser das Problem bei seinem Server beheben konnte. Diese Einschränkung verursachte, dass das Projektteam vor jeder Code-Änderung, zuerst überprüft hat, ob die Testumgebung überhaupt zur Verfügung steht, was einen zusätzlichen Zeitaufwand bedeutete.
Als die Probleme mit dem Glasfaseranschluss nicht mehr unter Kontrolle waren, stellte die Firma dxb gmbh noch zwei weitere virtuelle Server von anderen Standorten bereit.

2.8.2.2 Lösungsvariante

Beim nächsten Projekt, bei dem mindestens ein Teil der Testumgebung extern ist, ist ein Backup-System zu empfehlen. Dieses Backup kann entweder auf einem Server bei der HSR intern aufgebaut werden, auf bei weiteren Server des Auftraggebers zur Verfügung stehen oder die Testumgebung bei einem Cloud Provider, wie Amazon oder Azure, in Betrieb zu nehmen.

2.8.3 RGB LED

Der Ablauf der Authentifizierung sieht auch eine visuelle Statusanzeige auf dem Reader für den Endbenutzer vor. Das Projektteam erhielt am 16. Mai die vom Auftraggeber gewünschte RGB LED BlinkM (Abbildung 2.39) geliefert.

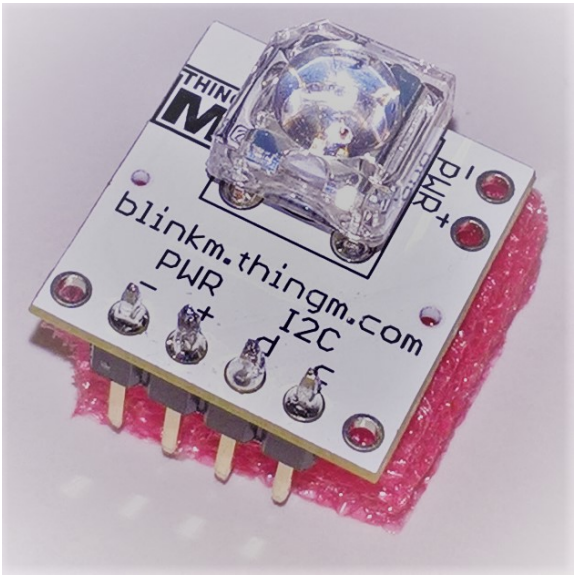


Abbildung 2.39: Selbst aufgenommenes Bild der RGB LED BlinkM von ThingM

Die RGB LED BlinkM ist vollständig kompatibel mit dem Raspberry Pi und wurde durch das Projektteam auch getestet. Das Anschliessen und Ansteuerung der LED ist relativ simpel und funktioniert mit einem zusätzlichen Softwarepaket auf Anhieb.
Die LED wird über den I2C-Bus angesteuert, jedoch sind diese Pins auf dem Raspberry Pi bereits physisch durch das NFC Hardwaremodul belegt. Das NFC Hardwaremodul wäre technisch in der Lage auch über den I2C-Bus verwendet zu werden, aber die verwendete Kombination in diesem Projekt läuft nur über den SPI-Bus. Diese doppelte physikalische Belegung der Pins auf dem Raspberry Pi wäre grundsätzlich kein Problem, da über das Bussystem mehrere Komponenten angesteuert werden können. Im Schaltplan des NFC Hardwaremodul (Abbildung 2.41) ist jedoch ersichtlich, dass die Anschlüsse für die I2C-Bus-Pins und die SPI-Pins gekoppelt sind, wodurch der I2C-Bus immer implizit belegt ist. Dadurch verursacht das gleichzeitige Anschliessen von der LED und dem NFC Hardwaremodul eine Fehlermeldung beim Ansteuern der RGB LED. Auf dem Pin-Belegungsplan (Abbildung 2.40) des Raspberry Pi ist zu sehen, dass keine anderen I2C-Bus-Pins existieren.
Es gibt unterschiedliche Lösungsvarianten(Kapitel 2.8.3.1), wie die LED mit dem NFC Hardwaremodul auf dem Raspberry Pi trotz der Hardwareinkompatibilität eingesetzt werden kann, jedoch sind diese alle mit einem grösseren Aufwand verbunden, entweder in Bezug auf die Hardware oder Software.
Auf Grund von inkompatibler Hardware und unbestimmten Lieferzeiten von Ersatzhardware wurde in Absprache mit dem Auftraggeber entschieden, dass die visuelle Statusanzeige auf dem Reader mittels einer LED nicht mehr zum Projektlieferumfang der Studienarbeit zählt. Ausserdem ist es nicht im Interesse des Auftraggebers, dass bei der Hardware für den Prototyp aufwändige Änderungen von Hand gemacht werden müssen.

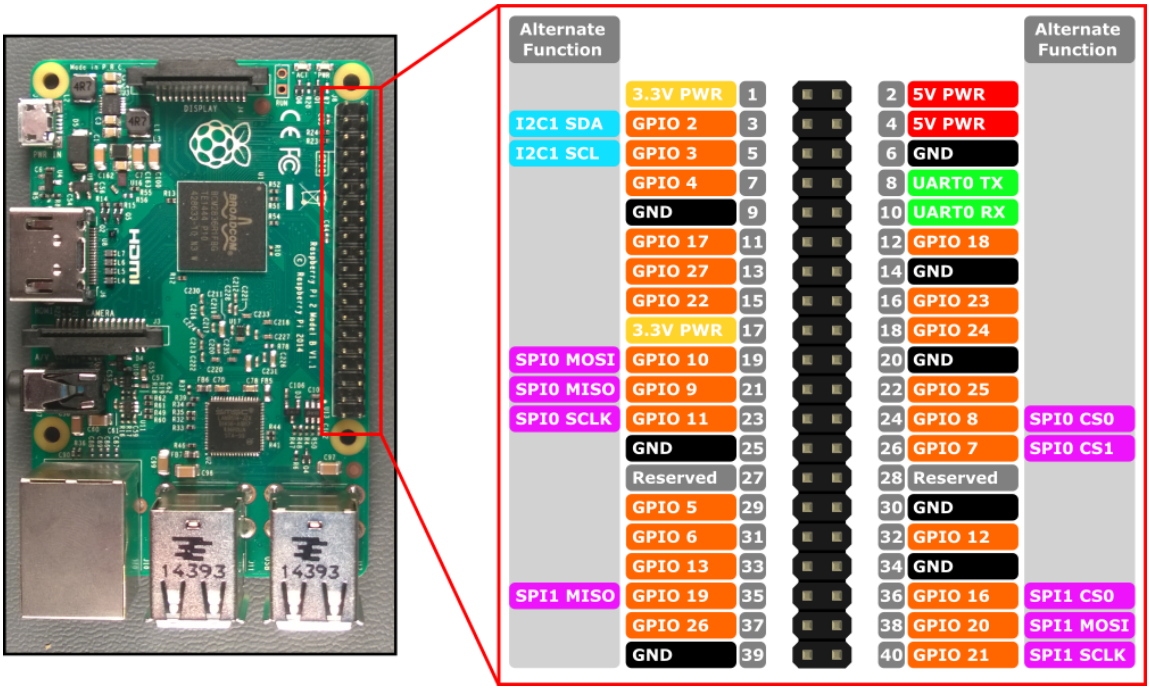


Abbildung 2.40: Pinout des Raspberry Pi 3 Model B[7]

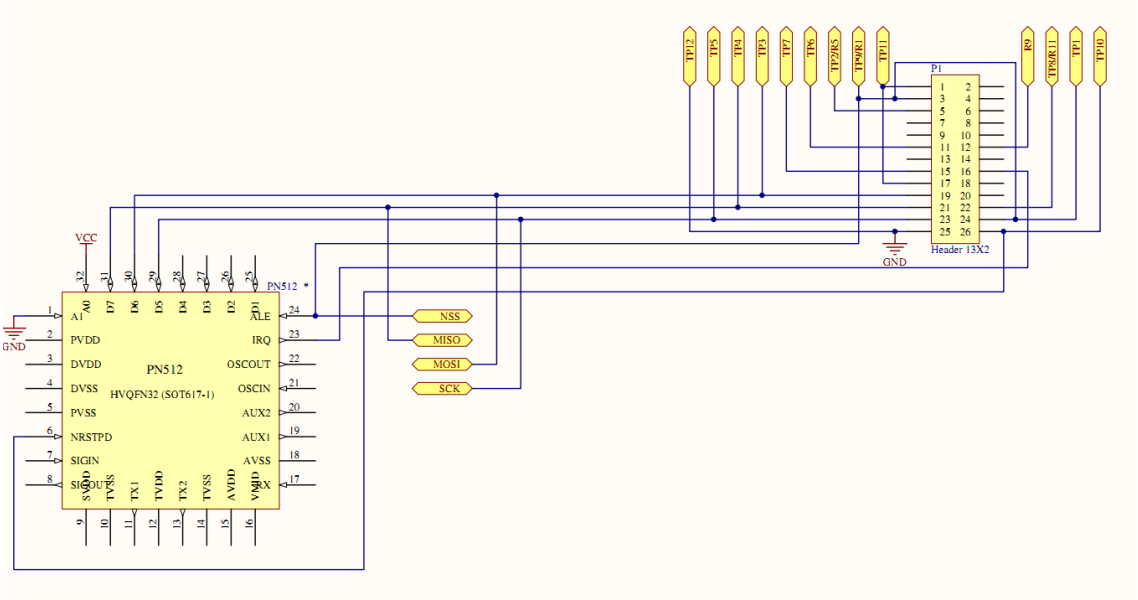


Abbildung 2.41: Schaltplan des NFC Hardwaremodul EXPLORE-NFC-WW[1]

2.8.3.1 Lösungsvarianten

Folgend sind ein paar verschiedene Lösungsvorschläge aufgelistet, welche teilweise auch nachverfolgt wurden.

1. Lösungsvariante:
Die RGB LED könnte vom existierenden Bauteil ausgelötet und mit Vorwiderständen wieder auf freie GPIO-Pins gelötet werden. So kann die jeder Eingang der LED separat via einen freien GPIO-Pin angesteuet werden.

2. Lösungsvariante:
Einen eigenen I2C-Bus implementieren und mit diesem freie GPIO-Pins ansteuern, an welchen die RGB LED angeschlossen wird. In dieser Alternative muss jedoch noch darauf geachtet werden, dass zwei I2C-Anschlüsse am RGB-LED-Bauteil jeweils mindestens einen 10-kOhm-Widerstand erhalten müssen, welche mit Power verbunden sind. Diese Modifikation ist notwendig für den erfolgreichen Nachbau eines Hardwareseitigen I2C-Busses. Dieser Lösungsansatz wurde kurzzeitig verfolgt. Nach den Vorbereitungen des Bauteiles und einigen Nachforschungen wurde der Ansatz jedoch fallen gelassen, da sich die Implementierung eines solchen Busses doch schwerer gestaltete als vermutet.

3. Lösungsvariante:
Es gibt einen I2C-Multiplexer mit dem eine Splittung des I2C-Busses möglich ist. Jedoch ob das oben beschriebene Problem damit gelöst werden kann müsste ausprobiert werden. Die Vermutung liegt nahe, dass diese Variante nicht erfolgversprechend ist. Dieses Bauteil besitzt wiederum unbestimmte Lieferfristen, weshalb bei den HSR Instituten INS (Informatik-Institut) und ICOM (Elektrotechnik-Institut) abgeklärt, ob sie ein solches Modul besitzen, jedoch war die Antwort negativ.

4. Lösungsvariante:
Eine weitere Möglichkeit wäre die Verwendung von anderen separaten LEDs, welche dann auch wieder einen Vorwiderstand erhalten müssten, aber relativ primitiv angesteuert werden könnten. Diese Lösung würde bestimmt funktionieren, jedoch verursacht sie noch höhere Material und Aufwandskosten für jeden Reader, was nicht im Interesse des Auftraggebers ist.

5. Lösungsvariante:
Eine weitere Alternative ist die Verwendung einer USB-LED (https://blink1.thingm.com/) ebenfalls vom Hersteller ThingM, welche über einen freien USB-Port angeschlossen und angesteuert werden kann. Mit dieser Variante würde sich auch das Problem der Anzeige relativ einfach lösen lassen, da die Statusanzeige durch die externe LED trotz komplett geschlossenem Raspberry Pi für den Benutzer gut sichtbar ist.

6. Lösungsvariante:
Es gibt noch eine heiklere Lösung, bei welcher die Leiterbahnen des NFC Hardwaremodul bearbeitet werden müssten[1]. Da hierbei jedoch der Schaltplan und der physische Aufbau des Modules genau studiert werden müssen und dieser Vorgang nur schwer wieder Rückgängig gemacht werden kann, wird auf diesen Versuch verzichtet.

2.8.3.2 Lösungsempfehlung

Das Projektteam empfiehlt im Rahmen der Abklärungen die Lösungsvariante 5 weiterzuverfolgen, da bei dieser Variante der manuelle Aufwand zum Zusammenbauen eines Readers am kleinsten ist und die Statusanzeige für den Benutzer am besten sichtbar wird.

2.9 Ergebnisse

Im Rahmen der Arbeit wurde ein funktionierender Prototyp entwickelt, der die meisten zu Beginn mit dem Kunden spezifizierten Anfroderungen erfüllt. Zu den implementierten Komponenten gehören der Provisioning Service, das Authentifizierungsmodul im Gordo und der Reader. Es ist nun möglich sich am Gordo mit einer NFC Karte anzumelden. Die Verwaltung der Karten und Reader ist im Authentifizierungsmodul des Gordo implementiert. Die Verwaltung der Reader und Kundenprojekte stehen in der Benutzeroberfläche des Provisioning Service zur Verfügung und erleichtern die Projektzuordnung.
Ein offener Punkt ist noch ein visueller Statusindikator am Reader für den Benutzer. Dieser konnte auf Grund von Hardwareinkompatibilität und unbestimmten Lieferzeiten nicht mehr rechtzeitig implementiert werden.

2.10 Erweiterungen

2.10.1 Authentifizierungsmodul

Das Authentifizierungsmodul stellt an sich bereits eine Erweiterung des Gordo dar, welches einfach und reibungslos für eine Umgebung aktiviert werden kann.

2.10.1.1 Broadcast-Job

Eine technische Erweiterung beziehungsweise Optimierung des Systems betrifft die Handhabung der Jobs. Dabei wird aktuell bei einem Broadcast-Job für jeden im System existierenden Reader einen eigenen Job erstellt und zugewiesen. Dies verursacht temporär die Erstellung einer grösseren Anzahl Jobs. Diese kurzzeitige hohe Auslastung könnte durch die Verwendung eines gemeinsamen Broadcast-Jobs reduziert werden.

2.10.1.2 Bezeichnung für Reader

Die Tabelle der Reader im Authentifizierungsmodul soll noch eine Spalte für die Bezeichnung des Readers erhalten. Dabei wird dann der Name oder der Standort des Readers erfasst, damit die Benutzer bei der Readerauswahl nicht mit generierten ReaderIds hantieren müssen.

2.10.2 Provisioning Service

2.10.2.1 Autolearning

Für den Provisioning Prozess ist bereits eine Erweiterung angedacht, welche dessen Ablauf mit Hilfe eines Autolearnings vollautomatisiert. Dadurch würden die Reader anhand der erstmals aufgelegten Karte direkt dem entsprechenden Kundenprojekt zugeordnet, wodurch dieser Prozess weitere Komponenten umfassen wird. Somit würde die manuelle Vorerfassung der Reader wegfallen. Diese erweiterte Version wurde während der Workshops bereits diskutiert und daher in einem Sequenzdiagramm (Abbildung 2.42) festgehalten. Daher konnte dieser Ablauf gezielt beurteilt werden, woraus sich ergab, dass dieser erweiterte Prozess für den Auftraggeber nicht im Fokus des Prototypen liegt.

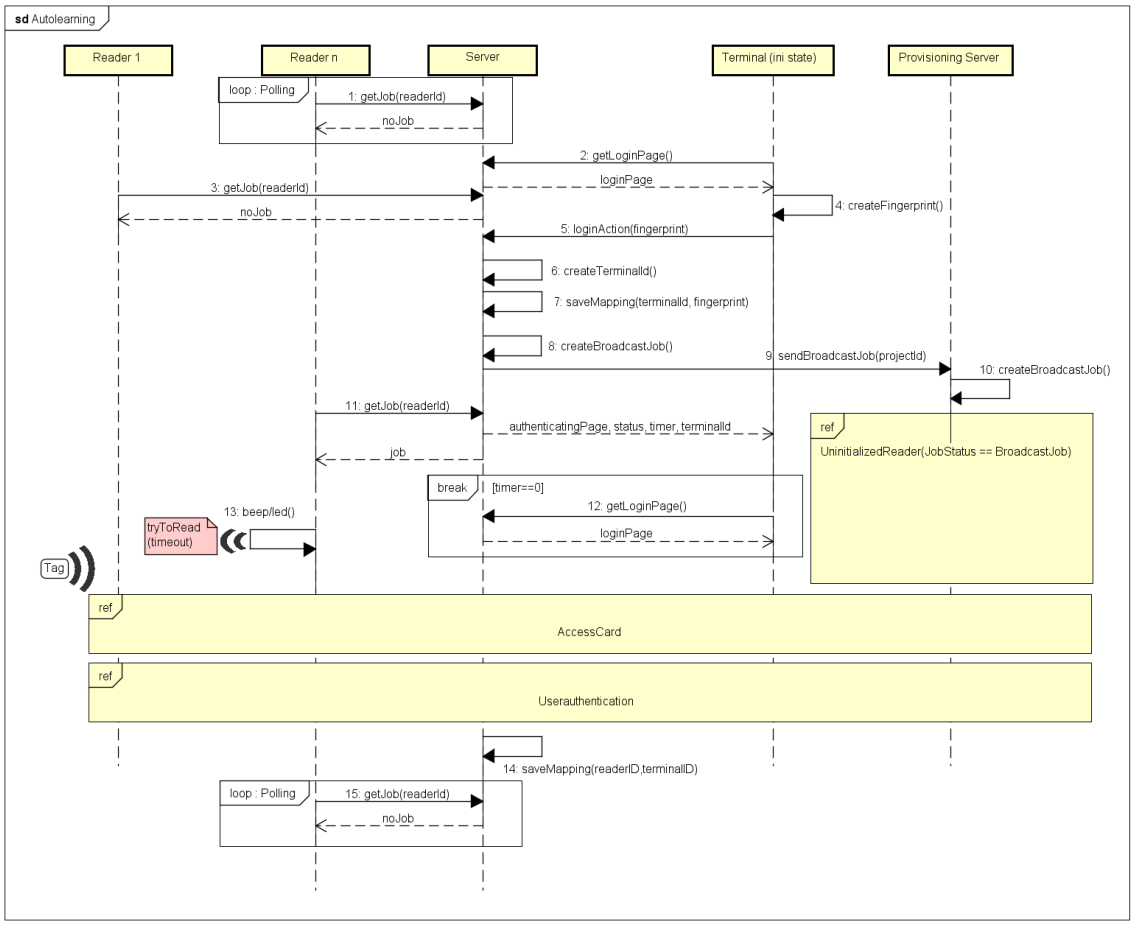


Abbildung 2.42: Sequenzdiagramm mit der Idee des Autolearning Prozess für den Provisioning Service aus einem frühen Stadium des Projektes

2.10.2.2 Verteilung eines neuen symmetrischen Keys

Eine weitere Ergänzung betrifft die Generierung eines neuen AES Key für die symmetrische Verschlüsselung. Nach der Erstellung eines neuen AES Key auf dem Provisioning Service, könnte dieser automatisch den Kundensystemen übertragen werden, von wo aus deren Gordo den neuen AES Key wiederum automatisch den Readern verteilt. Für eine sichere Übertragung könnte der neu generierten AES Key vom Provisioning Service mit dem alten AES Key verschlüsselt werden.

2.10.2.3 Letzte Anmeldung

Jede Reader-Anmeldung am Provisioning Service soll in Zukunft zeitlich erfasst und ausgewiesen werden. Dies soll die Fehlersuche für den Administrator deutlich vereinfachen.

2.10.3 Reader

2.10.3.1 Visuelles Benutzerfeedback

Die Verwendung eines visuellen oder akustischen Feedbacks an den Endbenutzer konnte auf Grund von Hardwareinkompatibilität nicht umgesetzt werden. Daher wird die Umsetzung dieser Kommunikation ebenfalls als Erweiterung aufgeführt (Details siehe im Kapitel 2.8.3).

2.10.3.2 Unterstützung von anderen NFC-Modulen

Eine möglicherweise gewünschte Erweiterung des Systems betrifft die künftige Unterstützung von RFID-Readern mit neueren Chips und von anderen Herstellern. Andere Hersteller, welche NCF-Module für den Raspberry Pi verkaufen, könnten künftig auch interessant werden, sei es wegen des Preises oder weil ähnlichen Systeme bereits beim Endkunden vorhanden sind. Die Unterstützung von neueren Chips wird vermutlich weniger Änderungen benötigen als der Umstieg auf einen anderen Hersteller.

2.10.3.3 App für die Unterstützung von Mobilgeräten als NFC-Reader

Die nächste Erweiterung ist ein Lösungsansatz, um die Abhängigkeit von NXP abzuschwächen, indem die NFC Funktion der Smartphones und Tablets verwendet werden kann.

2.10.3.4 Mehrere Projekte pro Reader

Eine weitere bereits gewünschte Erweiterung seitens des Auftraggebers, die während dem Projekt erwähnt wurde, ist eine die Zuordnung eines Reader zu mehreren Projekten gleichzeitig. So kann ein Administrator von mehreren Projekten mit einem einzigen Reader Karten für alle betreuten Projekte ausstellen.

2.10.3.5 Vereinfache Readerinstallation

Für die Installation eines Reader existiert momentan ein Installationsdokument, dieses sollte für standardmässige Installationen nicht mehr notwendig sein. Zukünftig soll nach dem installieren des Raspberry Pi Image nur noch eine Befehlszeile für die Konfiguration und Installation der Readersoftware benötigt werden.

2.10.3.6 Kein Desktop

Der normale Betriebsmodus des Reader nach der Installation benötigt im produktiven Einsatz kein Benutzeroberfläche mehr. Daher soll der Raspberry Pi nach der Installation künftig nur im Shell-Modus laufen und dafür ein Root-Passwort generieren, dass Reader-spezifisch ist.

2.10.3.7 Online Shell

Für die Administration der Reader ist eine Online Shell angedacht, welche in der Readerverwaltung des Provisioning Service oder auch des Authentifizierungsmoduls aufgerufen werden kann.

2.10.3.8 Neues Schlüsselpaar generieren

Aktuell generiert der Reader beim ersten Zuweisen zu einem Projekt sein Schlüsselpaar von öffentlichem und privatem Schlüssel. Damit die Sicherheit erhöht wird entstand noch der Wunsch, dass der Reader bei jedem Aufstarten sich ein neues Schlüsselpaar generiert, welches er dann bis zum nächsten Neustart verwendet.

2.10.4 Karten

2.10.4.1 Unterstützung von weiteren NTAG

Die Reader unterstützen aktuell die NTAG-Versionen NTAG 213 und NTAG 216. Sollte die Unterstützung weiterer NTAG Karten gewünscht sein, können die Konfigurationsparameter gemäss der Erklärung im Kapitel 2.7.1.2 ergänzt werden.

2.10.4.2 Unterstützung von Karten mit integrierten Rechenmodulen

Soll das System irgendwann Karten mit integrierten Rechenmodulen verwenden können, dann müssen die Rollen aller Beteiligten Komponenten in der Karten-Reader-Server-Kommunikation nochmals genauer betrachtet werden. Dies ist notwendig, weil somit nur durch die Verschiebung von Sicherheitsaspekten und Anpassung gewisser Prozesse auf den einzelnen Komponenten eine optimale Lösung garantiert werden kann.

2.10.5 Gesamtsystem

2.10.5.1 Kompatibilität zu anderen Webanwendungen

Eine deutlich grössere Erweiterung stellt die folgende Idee dar. Das gesamte System könnte so generell weiterentwickelt werden, dass es sich an jede beliebige Webapplikation einbinden lässt. Allerdings müssten selbst dann noch einige Anforderungen an das bestehende System existieren.
Da Gordo sehr flexibel in seinem Einsatzgebiet ist, kann das neue NFC-basierte Authentifizierungssystem bereits für diverse Anwendungsbereiche in unterschiedlichen Branchen verwendet werden.

2.11 Schlussfolgerung

Bei der Entwicklung eines Softwaresystems ist der Einsatz beziehungsweise die Verwendung von Hardware grundsätzlich sehr interessant und lässt einen ganz anderen Spiel- und Entwicklungsraum offen. Jedoch entstehen dadurch auch zusätzliche Risiken, welche einkalkuliert werden müssen. Dazu gehören Voraussetzungen des Auftraggebers in Bezug auf die Hardware, herstellerspezifische Unterschiede, Lieferverzögerungen oder auch die Inkompatibilität von Hardware.

Die entwickelte Software des Prototyp-Systems funktioniert einwandfrei und ist einsatzbereit für die ersten Testläufe bei Endkunden. Die einzig fehlende Komponente ist die visuelle Anzeige auf dem Reader für den Benutzer, um zu erkennen in welchem Status sich der Reader befindet. Diese Erweiterung sollte wenn möglich noch vor dem ersten Kundentest umgesetzt werden. Dafür muss sich der Auftraggeber allerdings zuerst noch für eine kompatible Hardware entscheiden. (Empfehlungsdetails sind unter dem Kapitel 2.8.3.2 zu finden)

Das erarbeitete System ist momentan sehr spezifisch auf die vom Auftraggeber existierende Business Applikation Gordo zugeschnitten und kann daher nicht für beliebige Webanwendungen eingesetzt werden. Da Gordo jedoch sehr flexibel in seinem Einsatzgebiet ist, kann das neue NFC-basierte Authentifizierungssystem bereits für diverse Anwendungsbereiche in unterschiedlichen Branchen verwendet werden. Dies erlaubt der Firma dxb gmbh zu ermitteln in welchen Bereichen das neue Authentifizierungssystem auf grosse Akzeptanz stösst.

2.12 Anhang

2.12.1 Literaturverzeichnis

[1] Blecky. Explore nfc layout. URL https://hackaday.io/project/1812-ourticket/log/5159-explore-nfc-layout. [Zugegriffen 27. Mai 2017].

[2] Free Clipart clker.com. Symbol eines readers. URL http://www.clker.com/cliparts/Q/Z/O/P/n/Z/rfid-reader-hi.png. [Zugegriffen 1. Juni 2017].

[3] NFC Forum. Type 2 tag operation specification, 2011. URL http://apps4android.org/nfc-specifications/NFCForum-TS-Type-2-Tag_1.1.pdf. [Zugegriffen 27. Mai 2017].

[4] icons8. Chorme logo, . URL https://maxcdn.icons8.com/Share/icon/Logos//chrome1600.png. [Zugegriffen 1. Juni 2017].

[5] icons8. Firefox logo, . URL https://maxcdn.icons8.com/Share/icon/Logos//firefox1600.png. [Zugegriffen 1. Juni 2017].

[6] icons8 for Microsoft. Edge logo. URL https://maxcdn.icons8.com/Share/icon/Logos//ms_edge1600.png. [Zugegriffen 1. Juni 2017].

[7] Microsoft. Raspberry pi 2 & 3 pin mappings. URL https://developer.microsoft.com/en-us/windows/iot/docs/pinmappingsrpi. [Zugegriffen 27. Mai 2017].

[8] NXP Semiconductors N.V. Ntag. URL http://www.nxp.com/products/identification-and-security/smart-label-and-tag-ics/ntag:MC_71717. [Zugegriffen 27. Mai 2017].

[9] NXP Semiconductors N.V. Ntag213/215/216, 2015. URL https://www.nxp.com/documents/data_sheet/NTAG213_215_216.pdf. [Zugegriffen 27. Mai 2017].

[10] Soarogo. Php+mysql-logo. URL https://www.soarogo.com/images/2017/04/logos-php-mysql_HyPQCO2Re.png. [Zugegriffen 1. Juni 2017].

[11] Unbekannt. Symbol einer karte. URL https://4.imimg.com/data4/JL/IE/MY-3093254/rfid-card-250x250.jpg. [Zugegriffen 1. Juni 2017].

[12] Valentin Vasilyev. Fingerprint2js. URL https://github.com/Valve/fingerprintjs2. [Zugegriffen 27. Mai 2017].

[13] Colin Viebrock. Php logo. URL https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/PHP-logo.svg/500px-PHP-logo.svg.png. [Zugegriffen 1. Juni 2017].

[14] Scott Vitale. nxppy. URL https://github.com/svvitale/nxppy. [Zugegriffen 27. Mai 2017].

2.12.2 Testprotokoll

2.12.2.1 Allgemeine Voraussetzungen

Die folgenden Tests setzen voraus, dass ein Provisioning Service und ein Gordo in Betrieb sind und verwendet werden können. Bei jedem Test wird ein frischer Provisioning Service und Gordo verwendet.

2.12.2.2 Testübersicht

Testfall 1	Automatische Konfiguration mit Provisioning Service	**Erfüllt**
Testfall 2	Projektzuordnung	**Erfüllt**
Testfall 3	Neue Projektzuordnung	**Erfüllt**
Testfall 4	Falsches SharedSecret	**Erfüllt**
Testfall 5	Neues SharedSecret	**Erfüllt**
Testfall 6	Ungültiges Zertifikat	**Erfüllt**
Testfall 7	Keine Projektzuordnung	**Erfüllt**
Testfall 8	Benutzeranmeldung	**Erfüllt**
Testfall 9	Autolearning	**Erfüllt**
Testfall 10	Autolearning 2	**Erfüllt**
Testfall 11	Anmeldung mit ungültiger ReaderId	**Erfüllt**
Testfall 12	Reader deaktivieren	**Erfüllt**
Testfall 13	Reader deaktivieren 2	**Erfüllt**
Testfall 14	Smartphone Login	**Erfüllt**
Testfall 15	Cookies gelöscht	**Erfüllt**
Testfall 16	Karte(NTAG213) initialisieren	**Erfüllt**
Testfall 17	Karte(NTAG216) initialisieren	**Erfüllt**

Testfall 18	Mehrere Karten pro Benutzer	**Erfüllt**
Testfall 19	Karte deaktivieren	**Erfüllt**
Testfall 20	Karte reinitialisieren	**Erfüllt**
Testfall 21	Maximale Anzahl Passwortversuche	**Erfüllt**
Testfall 22	Ablaufdatum der Konfiguration abgelaufen	**Erfüllt**
Testfall 23	Ablaufdatum aktualisieren	**Erfüllt**
Testfall 24	Verschlüsselter Netzwerkverkehr	**Erfüllt**
Testfall 25	Provisioning Service nicht erreichbar	**Erfüllt**
Testfall 26	Reader Statusanzeige	**Fehlgeschlagen**

2.12.2.3 Testfälle

Testfall 1	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Die Readerliste im Provisionierungsservice ist leer.
Ablauf	• Der Reader wird ans Stromnetz angeschlossen
Erwartetes Ergebnis	Der Reader ist in der Readerliste des Provisioning Servers mit seiner ReaderId, seiner MAC-Adresse und ohne Projektzuordnung ersichtlich.

Abbildung 2.43: Readerliste auf dem Provisioning Service

Testfall 2	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader wurde bereits einmal gestartet und wird im Provisioning Service aufgelistet. Im Provisioning Service ist ein Projekt erfasst.

Ablauf	• Der Reader wird im Provisioning Service ausgewählt • Im Popup wird ein Projekt für diesen Reader ausgewält • Die Änderung wird gespeichert • Es wird 40 Sekunden gewartet (Der Reader generiert seine Schlüssel)
Erwartetes Ergebnis	Der Reader ist mit seinem öffentlichen Schlüssel in der Readerliste des Gordo Servers ersichtlich.

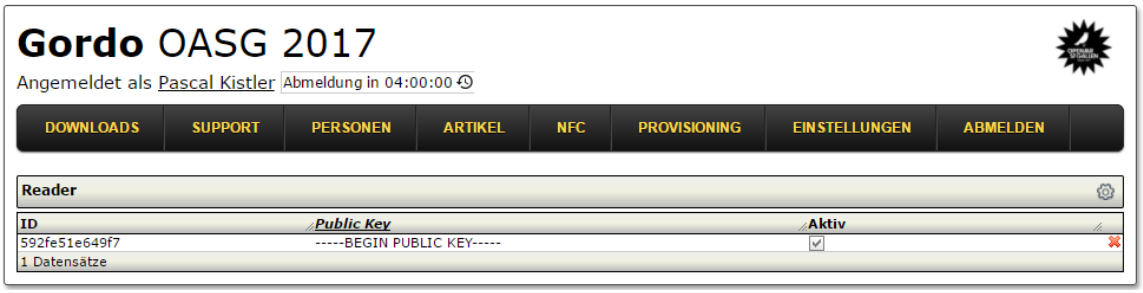


Abbildung 2.44: Readerliste im Gordo

Testfall 3	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist im Provisioning Service einem Projekt zugeordnet.

Ablauf	• Der Reader wird im Provisioning Service ausgewählt • Im Popup wird ein anderes Projekt für diesen Reader ausgewält • Die Änderung wird gespeichert • Der Reader wird neu gestartet
Erwartetes Ergebnis	Nach dem Neustart ist der Reader in der Readerliste des neu zugewiesenen Gordo Servers ersichtlich.

Abbildung 2.45: Readerliste im Gordo

Testfall 4	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden.
Ablauf	• Auf dem Reader wird der AES Key im Installationsordner in der Datei configAES.php im Ordner config abgeändert. • Der Reader wird gestartet
Erwartetes Ergebnis	Nach dem Start ist der Reader aufgrund des falschen AES Keys nicht in der Readerliste des Provisioning Service ersichtlich.

Abbildung 2.46: Readerliste im Gordo

Testfall 5	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist im Provisioning Service einem Projekt zugeordnet.
Ablauf	• Im Provisionierungsservice wird unter Provisioning -> Konfigurationsverwaltung ein neuer AES Key generiert • Der neu generierte AES Key wird kopiert • Auf dem Projektserver, welchem der Reader zugeordnet ist, wird der AES Key unter NFC -> Update Konfiguration eingefügt und gespeichert. • Der Readereintrag auf dem Provisioning Service wird gelöscht • Der Reader wird neugestartet
Erwartetes Ergebnis	Der Reader kann sich mit dem neuen AES Key beim Provisioning Service anmelden und erscheint nach dem Aufstarten in dessen Readerliste.

Testfall 6	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden.
Ablauf	• Auf dem Provisionierungsserver wird ein ungültiges Zertifikat installiert • Der Reader wird gestartet
Erwartetes Ergebnis	Der Reader verbindet sich nicht mit dem Server und erscheint nicht in der Readerliste.

Testfall 7	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist im Provisioning Service einem Projekt zugeordnet.
Ablauf	• Auf dem Provisioningserver wird in der Readerliste der Reader ausgewählt • Im DropDown der Projektauswahl wird kein Projekt ausgewählt • Die Änderung wird übernommen • Der Reader wird neugestartet
Erwartetes Ergebnis	Der Reader hat die URL des zuletzt zugeordneten Gordos gelöscht.

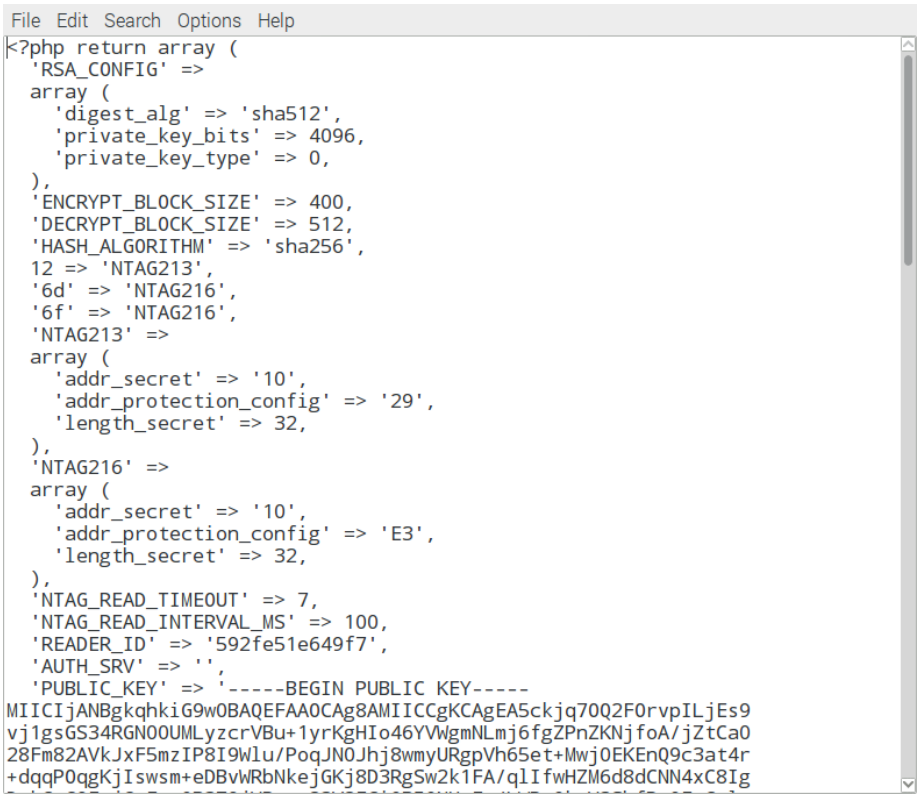


Abbildung 2.47: Reader Konfiguration mit leerer Gordo URL (AUTH_SRV)

Testfall 8	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist in einem Projekt aktiv. Der Benutzer befindet sich auf der Loginseite. Eine Karte ist für den Benutzer ausgestellt worden.

Ablauf	• Der Benutzer wählt den Reader aus und wählt anmelden aus • Der Benutzer legt seine Karte auf den Reader
Erwartetes Ergebnis	Der Benutzer wird im Gordo authentifiziert.

Testfall 9	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist in einem Projekt aktiv. Der Benutzer befindet sich auf der Loginseite. Eine Karte ist für den Benutzer ausgestellt worden.
Ablauf	• Der Benutzer wählt keinen Reader aus und wählt anmelden aus • Der Benutzer legt seine Karte auf den Reader
Erwartetes Ergebnis	Der Benutzer wird im Gordo angemeldet.

Testfall 10	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist in einem Projekt aktiv. Der Benutzer befindet sich auf der Loginseite. Eine Karte ist für den Benutzer ausgestellt worden.
Ablauf	• Der Benutzer wählt keinen Reader aus und meldet sich an • Der Benutzer legt seine Karte auf den Reader • Nach der Anmeldung meldet sich der Benutzer wieder ab
Erwartetes Ergebnis	Auf der Anmeldeseite ist der verwendete Reader vorausgewählt.

Testfall 11	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist in einem Projekt aktiv. Der Benutzer befindet sich auf der Loginseite.

Ablauf	• Der Benutzer ändert mit den Development Tools die ReaderId in der DropDown Liste • Der Benutzer versucht sich mit dem geänderten Readereintrag anzumelden
Erwartetes Ergebnis	Der Anmeldeversuch wird zurückgewiesen und der Benutzer gelangt auf die Anmeldeseite zurück.

Testfall 12	**Erfüllt**
Voraussetzung	Zwei neue Reader wurden gemäss der Anleitung **Inbetriebnahme Reader** installiert und sind mit dem Internet verbunden. Die Reader laufen und sind dem selben Projekt zugeordnet.
Ablauf	• Im Gordo wird der eine Reader deaktiviert
Erwartetes Ergebnis	Der deaktivierte Reader wird auf der Seite zur Karteninitialisierung nicht in der Readerauswahl angezeigt.

Testfall 13	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet.
Ablauf	• In der Readerliste des Gordo wird der Reader deaktiviert • Der Benutzer meldet sich von Gordo ab und kehrt zur Anmeldeseite zurück
Erwartetes Ergebnis	Auf der Anmeldeseite wird der deaktivierte Reader nicht mehr angezeigt.

Testfall 14	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet. Der Benutzer befindet sich mit dem Smartphone auf der Loginseite

Ablauf	• Der Benutzer wählt einen Reader zur Anmeldung aus • Der Benutzer legt seine Karte auf den Reader • Der Benutzer meldet sich an
Erwartetes Ergebnis	Der Benutzer wird am Gordo angemeldet.

Testfall 15	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet. Der Benutzer hat sich bereits einmal mit dem Reader am System angemeldet.
Ablauf	• In den Einstellungen des Browsers werden die Cookies dieser Seite gelöscht • Die Loginseite wird aktualisiert
Erwartetes Ergebnis	Der zuletzt verwendete Reader ist vorausgewählt

Testfall 16	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet.
Ablauf	• Im Gordo auf der Seite NFC -> Karte initialisieren wählt der Administrator einen Benutzer und Reader aus • Die zu initialisierende Karte (NTAG213) wird auf den Reader gelegt • Die Initialisierung wird gestartet
Erwartetes Ergebnis	Der Benutzer kann sich mit seiner neu erstallten Karte im Gordo anmelden.

Testfall 17	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet.

Ablauf	• Im Gordo auf der Seite NFC -> Karte initialisieren wählt der Administrator einen Benutzer und Reader aus • Die zu initialisierende Karte (NTAG216) wird auf den Reader gelegt • Die Initialisierung wird gestartet
Erwartetes Ergebnis	Der Benutzer kann sich mit seiner neu erstellten Karte im Gordo anmelden.

Testfall 18	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet. Der Benutzer besitzt bereits eine ihm zugeordnete Karte.

Ablauf	• Im Gordo auf der Seite NFC -> Karte initialisieren wählt der Administrator den Benutzer und Reader aus • Die zu initialisierende Karte wird auf den Reader gelegt • Die Initialisierung wird gestartet
Erwartetes Ergebnis	Der Benutzer kann sich mit beiden auf ihn ausgestellten Karten im Gordo anmelden.

Testfall 19	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet. Der Benutzer besitzt bereits eine ihm zugeordnete Karte.
Ablauf	• Im Gordo wird die Karte des Benutzers deaktiviert • Der Benutzer versucht sich mit der deaktivierten Karte im Gordo anzumelden
Erwartetes Ergebnis	Der Benutzer kann sich nicht im Gordo anmelden

Testfall 20	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet. Eine Karte ist bereits einem Benutzer zugeordnet.
Ablauf	• Im Gordo wird die Karte mit einem anderen Benutzer initialisiert • Der neue Benutzer versucht sich im Gordo anzumelden
Erwartetes Ergebnis	Der neue Benutzer kann sich erfolgreich im Gordo anmelden.

Testfall 21	**Erfüllt**
Voraussetzung	Eine neue Karte wurde mit dem Python Script enableProtection.py (unter <Installationsverzeichnis>/src/access/nxppy/enableProtection.py) mit einem Passwort versehen.

Ablauf

- Mit dem Script read.py im selben Verzeichnis wird 8 Mal versucht mit einem falschen Passwort auf die Karte zuzugreifen (Beispiel in Abbildung 2.48)
- Nun wird versucht mit dem korrekten Passwort auf die Karte zuzugreifen

---

Erwartetes Ergebnis Der Leseversuch scheitert.

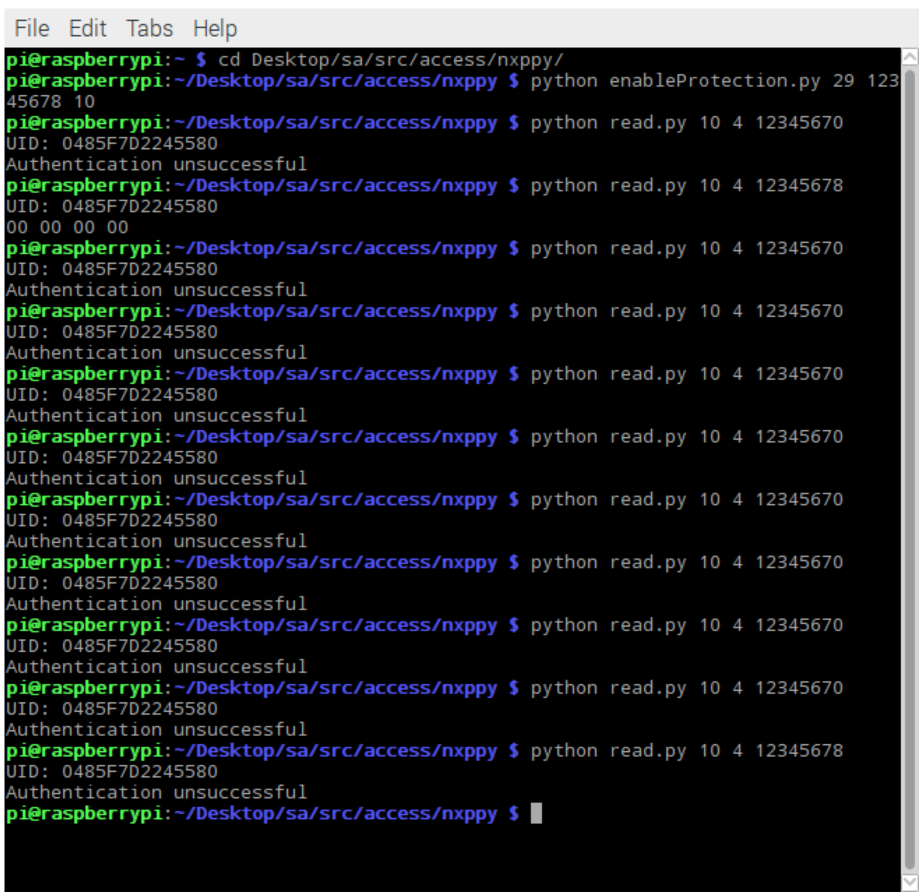


Abbildung 2.48: Maximale NTAG Passwortversuche erreicht

Testfall 22	Erfüllt
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet.
Ablauf	• Auf dem Reader wird in der Datei <Installationsverzeichnis>/config/config.php das Ablaufdatum (EXPIRATION_DATE) zwei Monate in die Vergangenheit gesetzt • Der Reader wird neu gestartet
Erwartetes Ergebnis	In der Konfigurationsdatei des Readers ist die readerspezifische Konfiguration gelöscht worden

File Edit Search Options Help

```
<?php return array (
  'RSA_CONFIG' =>
  array (
    'digest_alg' => 'sha512',
    'private_key_bits' => 4096,
    'private_key_type' => 0,
  ),
  'ENCRYPT_BLOCK_SIZE' => 400,
  'DECRYPT_BLOCK_SIZE' => 512,
  'HASH_ALGORITHM' => 'sha256',
  12 => 'NTAG213',
  '6d' => 'NTAG216',
  '6f' => 'NTAG216',
  'NTAG213' =>
  array (
    'addr_secret' => '10',
    'addr_protection_config' => '29',
    'length_secret' => 32,
  ),
  'NTAG216' =>
  array (
    'addr_secret' => '10',
    'addr_protection_config' => 'E3',
    'length_secret' => 32,
  ),
  'NTAG_READ_TIMEOUT' => 7,
  'NTAG_READ_INTERVAL_MS' => 100,
  'READER_ID' => '',
  'AUTH_SRV' => '',
  'PUBLIC_KEY' => '',
  'PRIVATE_KEY' => '',
  'AUTH_PUBLIC_KEY' => '',
  'EXPIRATION_DATE' => '',
);
```

Abbildung 2.49: Gelöschte Readerkonfiguration

Testfall 23	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet.

Ablauf	• Auf dem Reader wird in der Datei <Installationsverzeichnis>/config/config.php das Ablaufdatum (EXPIRATION_DATE) ein paar Tage in die Vergangenheit gesetzt • Auf dem Reader wird in der Datei <Installationsverzeichnis>/config/configAES.php ein ungültiger Provisioning Service angegeben • Der Reader wird neu gestartet
Erwartetes Ergebnis	In der Konfigurationsdatei des Readers ist das Ablaufdatum aktualisiert worden

Testfall 24	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet.

Ablauf	• Der Netzwerkverkehr zwischen Reader und Gordo wird aufgezeichnet • Der Benutzer meldet sich am Gordo an
Erwartetes Ergebnis	Im aufgezeichneten Netzwerkverker kann kein Kartenpasswort oder Secret ausgelesen werden.

Testfall 25	**Erfüllt**
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet.
Ablauf	• Auf dem Reader wird die URL des Provisioning Services im Installationsordner in der Datei configAES.php im Ordner config abgeändert. • Der Reader wird neugestartet
Erwartetes Ergebnis	Der Reader kann nach dem Neustart wieder für die Authentifizierung verwendet werden.

Testfall 26	Fehlgeschlagen
Voraussetzung	Ein neuer Reader wurde gemäss der Anleitung **Inbetriebnahme Reader** installiert und ist mit dem Internet verbunden. Der Reader läuft und ist einem Projekt zugeordnet. Eine Karte ist bereits einem Benutzer zugeordnet.
Ablauf	• Der Benutzer wählt auf der Loginseite den Reader aus • Der Benutzer klickt auf Anmelden
Erwartetes Ergebnis	Der Reader zeigt mit einer grünen LED, dass der Benutzer nun seine Karte in die Nähe halten soll.