Hochschule Worms
Fachbereich Informatik
Studiengang Angewandte Informatik - dual (M.Sc)

Projektbericht
Deep Dive

Entwicklung einer Lösung zur Berechtigungsverwaltung von Secrets zwischen Entwicklerinnen und Entwicklern

In der Arbeitsumgebung des Partnerunternehmens Medienagenten oHG

Version 1.0

Vorgelegt von

Leon Etienne, 676838
inf4437@hs-worms.de
Im Wintersemester 2024/25

bei Professor Dr. Heinemann
heinemann@hs-worms.de

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist noch nicht in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung vorgelegt worden.

Ludwigshafen am Rhein, 21. Februar 2025

Leon Etienne _____

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

1P 1Password

MASA Medienagenten Secret Authority

API Application Programmer Interface

Glossar

Docker

Eine arrivierte Container-Engine für Anwendungsentwicklung.

Ansible-Playbook/s

Ansible-Playbooks sind Skripte, mit dem Ziel einen deklarierten Zustand herzustellen. [Red Hat, Inc., 2025]

1. Einleitung

1.1. Problemstellung

In der Arbeitsumgebung des Partnerunternehmens besteht zum Zeitpunkt der Themenfindung der hier beleuchteten Arbeit kein Management für Secrets und Logindaten zwischen Entwicklern. Logindaten zu den Projekten des Unternehmens liegen schlicht in einem 1Password (1P)-Vault. 1P ist der vom Unternehmen verwendete Passwortmanager. Auf diesen Vault haben sämtliche interne Entwickler Zugriff, jedoch keine externen Entwickler. Das ist so, weil anderenfalls dem externen Entwickler Lesezugriff auf sämtliche Einträge dieses Vaults gegeben werden müssten. 1P unterstützt keine Freigaben einzelner Einträge an andere Nutzer, ohne diese Einträge in einen eigenen Vault zu kopieren. Würden diese manuell in einen eigenen Vault kopiert werden, müssten diese Einträge fortan redundant gepflegt werden. Das ist eine Fehlerquelle, die zu asynchronen Einträgen führt. Außerdem ist das ein großer Arbeitsaufwand. All das gestaltet das Einbinden von externen Entwicklern, wie z.B. Freelancern, schwer.

Ein weiteres Problem ist, dass Secrets in Konfigurationsdateien, die firmeninternen Ansible-Scripten beilegen, unverschlüsselt einsichtig sind. Das macht es zu einem großen Sicherheitsrisiko und somit impraktikabel externen Entwicklern Zugriff auf dieses Ansible-Repository zu gewähren. Dieses Ansible-Repository ist jedoch zwingend erforderlich, um eine Entwicklungsungebung für Firmenprojekte auf dem lokalen Rechner zu schaffen. Auch hier sind Lösungen für externe Entwickler zumeist unschöne Workarounds.

1.2. Zielsetzung

Ziel ist es, eine Umgebung zu schaffen, in der beliebigen Entwicklern bestimmte 1P-Einträge zugewiesen werden können. Der Pflegeaufwand sollte hierbei überschaubar bleiben. Das heisst, dass z.B. ganze Gruppen von Einträgen Entwicklern zugewiesen werden können. Wenn z.B. einem Projekt viele Einträge zugeordnet sind, sollten diese idealerweise mit einer einzigen Configzeile einem Entwickler zugeordnet werden können. Außerdem sollte eine Möglichkeit ausgearbeitet werden, um 1P-Einträge in Ansible auszulesen, damit keine Secrets mehr in den beiliegenden Konfigurationsdateien stehen, die das Freigeben dieser zu einem Sicherheitsproblem machen.

1.3. Methodische Vorgehensweise

Einige Anforderungen sind bereits im Voraus definiert. Weiterführende Anforderungen werden im Rahmen einer Anforderungserfassung ermittelt. Anschließend werden verschiedene Lösungsansätze betrachtet und auf Tauglichkeit geprüft. Nachdem ein akzeptabler Lösungsweg gefunden ist, wird dieser umgesetzt. Abschließend wird der Erfolg des Unterfanges evaluiert und mögliche, auf dieses Projekt aufbauende Arbeiten in Ausblick gestellt.

2. Grundlagen

2.1. Die Arbeitsumgebung

Die Arbeitsumgebung des Partnerunternehmens besteht für diese Themenstellug nennenswert aus:

- Cloudbasierten Web- und Datenbankservern
- Git-Repositories bei Bitbucket
- Der lokalen, Docker-basierten Arbeitsumgebung
- Ein Ansible-Playbook, das ein Projekt mit Daten aus der Cloudumgebung und Code aus Bitbucket in der lokalen Entwicklungsumgebung bereitstellt.

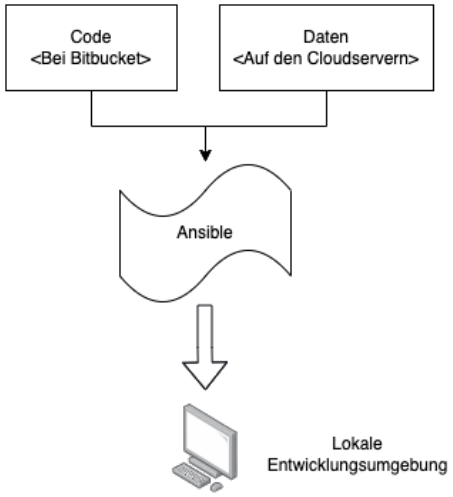

Abbildung 1.: Relationsdiagramm: Bereitstellen von Projekten des Partnerunternehmens in einer Entwicklungsumgebung

Quelle: Eigene Darstellung

Die lokalen Arbeitsumgebungen der Entwickler liegen großteils außerhalb des Firmennetzwerkes, da diese Entwickler oft oder ausschließlich

im mobilen- bzw, Homeoffice arbeiten. Ein Firmen-VPN-Netz existiert nicht und ist auch nicht erwünscht.

2.2. 1Password

1P ist der vom Partnerunternehmen verwendete Passwort-Manager. Bereits vor Beginn der Bearbeitung dieser Themenstellung wurde deutlich gemacht, dass es Ziel ist, 1P auch für das Verwalten von Secrets in Ansible zu verwenden.

2.3. Ansible

Ansible ist ein Automatisierungswerkzeug von Red Hat, Inc. und hat das Ziel, einen definierten Zustand im behandelten System herzustellen. [Red Hat, Inc., 2025] Ein Administrator definiert also nicht die erforderlichen Schritte, um einen Zustand z zu erreichen, sondern lediglich z selbst. Ansible kann über speziell gefertigte Python-Module um Schnittstellen erweitert werden.

3. Anforderungen

3.1. Anforderungserfassung

Obwohl bereits vor Beginn des Projektes einige Anforderungen bekannt sind, müssen manche Details nachträglich in Erfahrung gebracht werden. Hierfür wurde ein semistrukturiertes Interview mit dem Stakeholder durchgeführt. Im Rahmen dieses Interviews wurden vorbereitete Fragen gestellt, dem Stakeholder aber auch die Möglichkeit gegeben frei heraus zu sprechen und Wünsche zu äußern. Notizen zu diesem Interview befinden sich im Anhang unter *⟨⟨A Stakeholder-Interview⟩⟩*.

3.2. Ergebnisse

Das Ergenis der Anforderungserfassung ist ein Lastenheft, das in constraints, funktionale und nicht-funktioniale Anforderungen zu unterteilen ist.

Funktionale Anforderungen
Entwickler erhalten verschiedene Zugänge, definiert in einer YAML-Datei.
Wildcard-Matching auf den 1P-Eintragstitel für zusammenhängende Einträge.
1P-Einträge sollen einzeln zuweisbar sein.
Nicht im YAML gelistete Zugänge sollen bei Anwendung entfernt werden.
Ansible Secrets müssen aus 1Password dereferenziert werden können.
Nicht-funktionale Anforderungen
Das System muss Berechtigungen von Entwicklern verwalten.
Das System muss benutzerfreundlich sein.
Das System darf nicht aufwändig zu pflegen sein.
Die benötigte Zeit zur Ausführung der Anwendung soll nicht sehr lange sein.
Das System muss robust gegenüber Misskonfigurationen sein, die zur Löschung der zugrunde liegenden 1P-Einträgen führen könnten.
Constraints
Nutzung von 1Password ist zwingend erforderlich.
Die Übermittlung der Secrets muss über ds Internet erfolgen.

Tabelle 1.: Anforderungen

4. Technische Umsetzung

4.1. Berechtigungsverwaltung

Ausarbeitung der Herangehensweise

Ansatz 1

Zunächst wurde gebrainstormed, welche Herangehensweisen hier möglich sind. Ein Artefakt des Brainstormings ist eine Mind-Map, die unter *⟨⟨B Ideensammlung⟩⟩* zu finden ist. Der aus dieser Mindmap, nach individueller Meinung des Autors, vielversprechenste Ansatz ist es, die 1P-Restful-API zu verwenden. Bei diesem Ansatz würden Administratoren und Entwickler API-Keys für 1P erhalten. Entwickler haben mit ihren Keys bestimmte Leseberechtigungen r und Administratoren die Berechtigung r zu verändern.

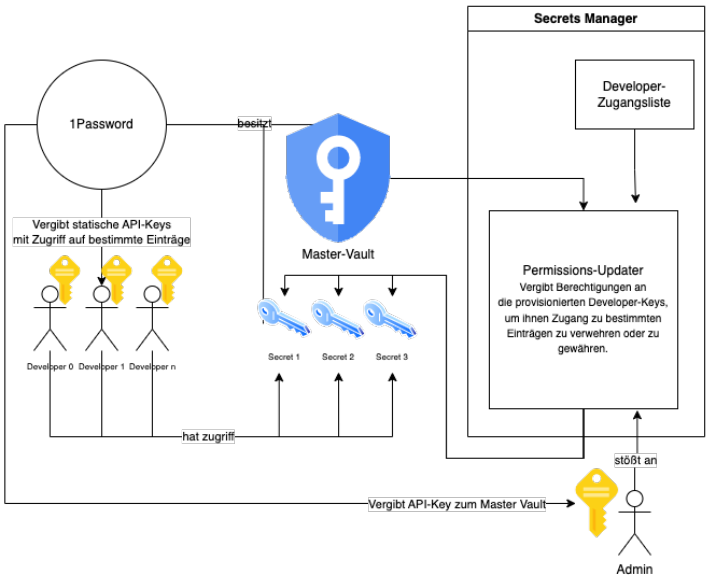

Abbildung 2.: Relationsdiagramm: Ansatz 1 | 1Password-API

Quelle: Eigene Darstellung

Dieser Ansatz wurde zeitnah als unumsetzbar erkannt und verworfen, da 1P das nachträgliche Verändern von API-Key-Berechtigungen nicht erlaubt.

Ansatz 2

Der nächste Lösungsansatz befasst sich mit einer Abstraktionsebene: Der Medienagenten Secret Authority (MASA). Hier ist die grundlegende Idee, dass es eine serverseitige Anwendung gibt, die sich MASA nennt. Diese Anwendung übernimmt die Aufgabe anhand eines hinterlegten 1P-API-Keys Secrets aus dem 1P-Vault des Partnerunternehmens abzufragen und an Entwickler weiterzureichen. Die MASA provisioniert eigene API-Keys an Entwickler und vermermerkt serverseitig, welcher API-Key berechtigt ist, welche 1P-Einträge abzufragen. Der API-Key könnte grundlegende Informationen wie zum Beispiel Entwicklernamen und Ablaufzeitpunkte des Keys einbetten. Dieser Ansatz trägt viel Sicherheitsverantwortung, da eine mögliche Ausnutzung einer Sicherheitslücke der MASA direkt in den Firmen-Passwortmanager führen würden. Um diesem Risikofaktor entgegenzuwirken würde der 1P-Key der MASA verschlüsselt werden und die MASA würde nur einen Teil des Entschlüsselungs-Keys vorrätig halten. Der andere Teil wäre in jedem Entwickler-Key eingebettet. Dadurch wäre gewährleistet, dass ein Angreifer, selbst bei sehr weitreichendem Zugriff in die MASA, nicht auf das Innere des Passwort-Managers zugreifen könnte, da die MASA dazu selbstständig gar nicht im Stande wäre. Da Entwickler lediglich ein Schlüsselfragment des Verschlüsselungs-Schlüssels in ihrem Key eingebettet hätten, der ohne einen serverseitigen Schlüssel der MASA nicht auslesen werden kann, bestünde auch keine Gefahr, dass ein Entwickler anhand seines bzw. ihres Keys ungeschützten Zugang zum Passwort-Manager erhaltne würde. Dieser Ansatz erlaubt für weitreichende Flexibilität, da sämtliche Logik, die sich mit Berechtigungen beschäftigt, selbst geplant und umgesetzt wäre.
Letztendlich entschied sich der Stakeholder gegen die Umsetzung der MASA, da dieser Ansatz für zu Aufwändig betrachtet wird und für den

durch sie erbrachten Vorteil zu viel Aufwand und Angriffsfläche schaffen würde.

Ansatz 3

Kodierung

4.2. Integration in Ansible

5. Evaluation

6. Fazit

6.1. Ausblick

6.2. Offene Problemstellungen

Literaturverzeichnis

[Maral et al., 1991] Maral, G., de Ridder, J.-J., Evans, B. G., und Richharia, M. (1991). Low earth orbit satellite systems for communications. *International Journal of Satellite Communications*, 9(4):209–225.

[Red Hat, Inc., 2025] Red Hat, Inc. (2025). Ansible Collaborative - What is Ansible? . `https://www.redhat.com/en/ansible-collaborative`. Zugriff: Januar 2025.

[TYPO3 Association, 2024] TYPO3 Association (2024). TYPO3 — the Professional, Flexible Content Management System . `https://typo3.org/`. Zugriff: Mai 2024.

Anhang

A. Stakeholder-Interview

!!!TODO TODO TODO ADD APPENDIX INTERVIEW!!!

B. Ideensammlung

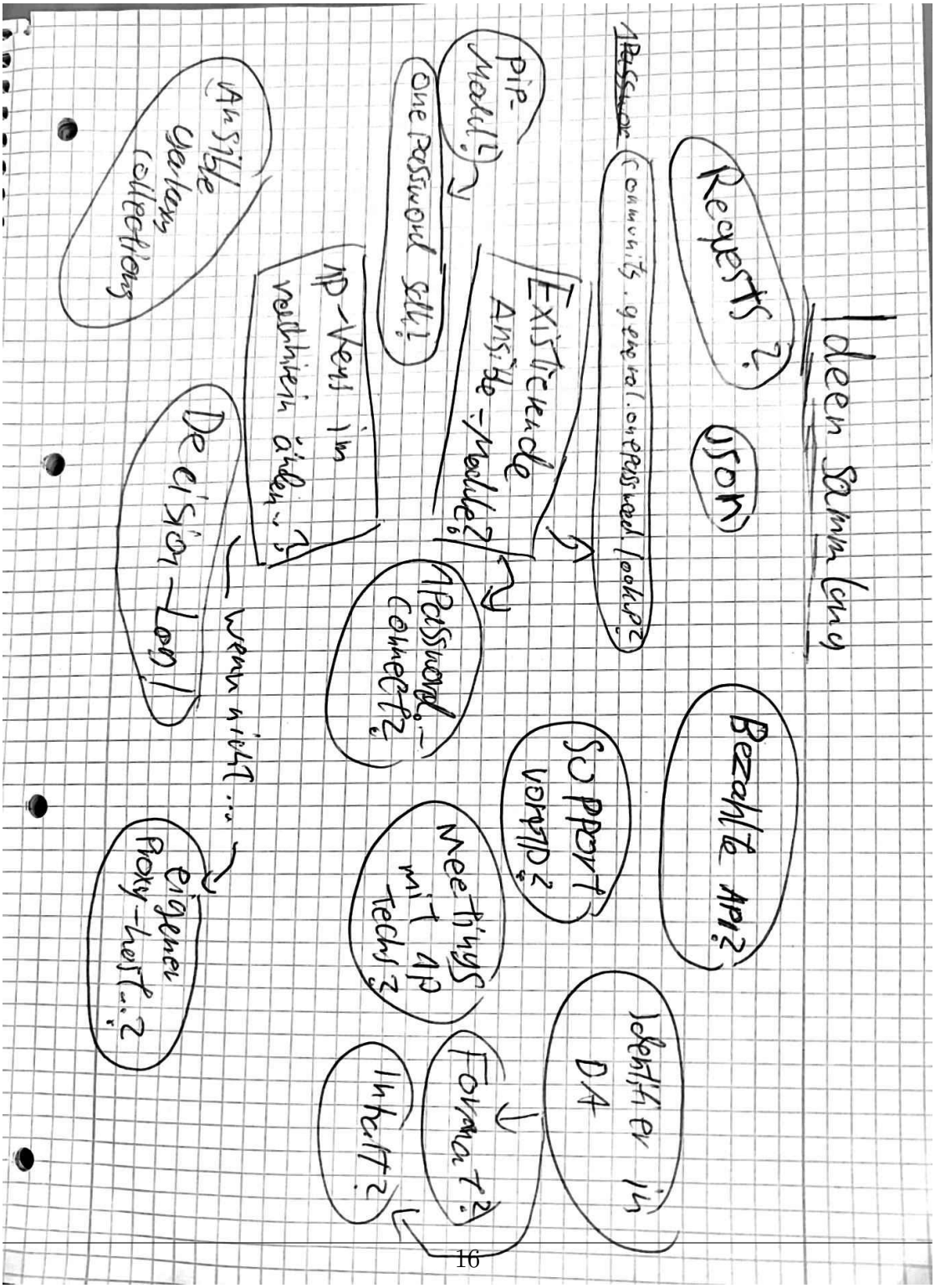

Abbildung 3.: Ideensammlung

Quelle: Eigene Darstellung

C. Relationsdiagramm (Überholt)

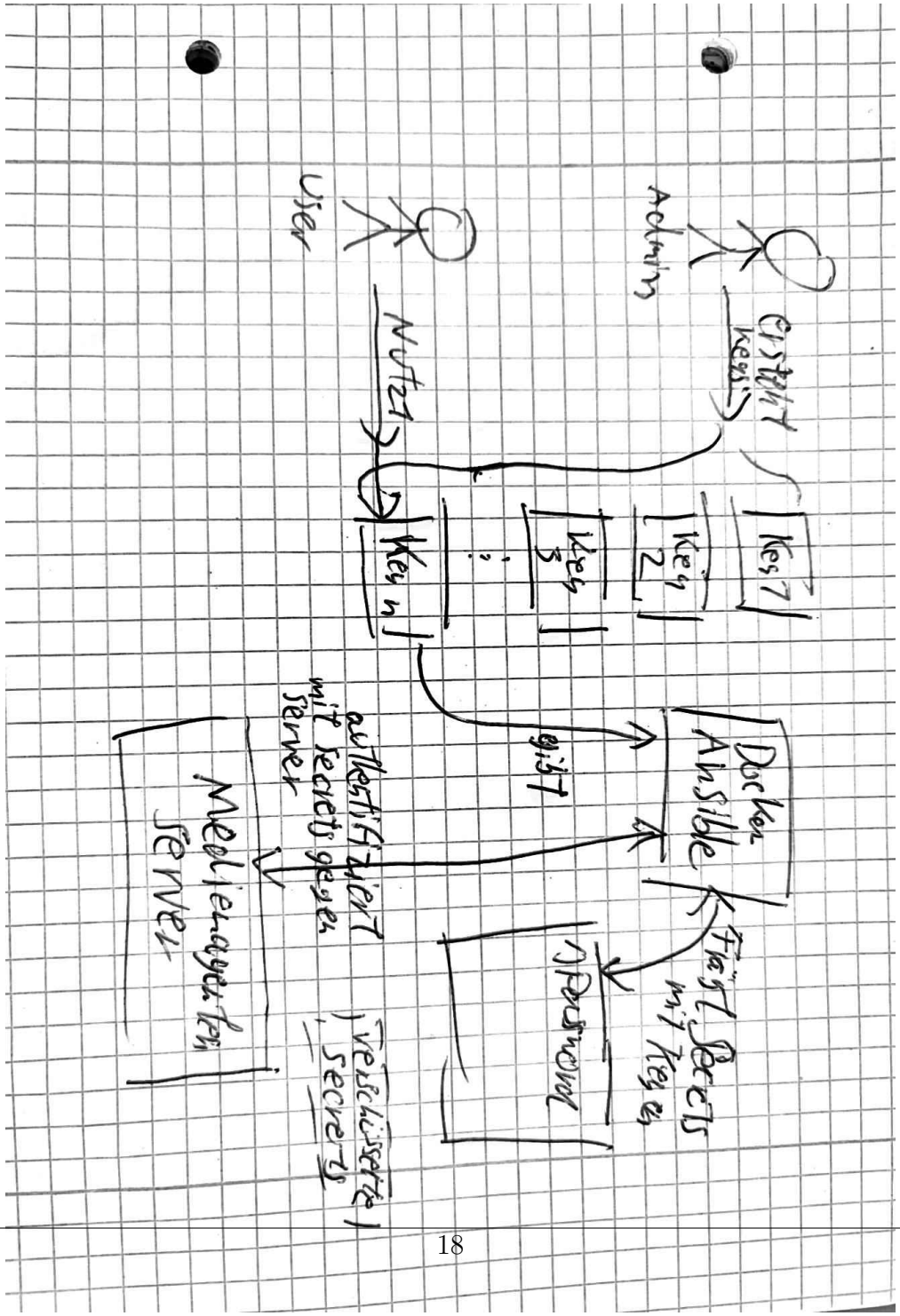

Abbildung 4.: Relationsdiagramm: (Überholt) Relationsdiagramm

Quelle: Eigene Darstellung