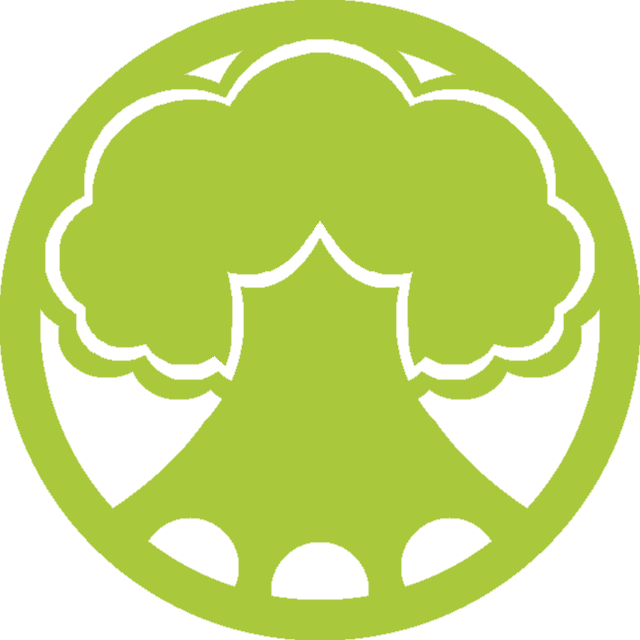
**Projekthandbuch**

**Backup Magican**

**(Cloudbasiertes Backupmanagement System)**

**Von den Schülern der 6BBKIF**

**Morhammer Roland, Rus Alex & Polak Rene**



Backup Magician

Inhalt

[Einführung 2](#_Toc513563773)

[Projektdefinition 2](#_Toc513563774)

[Ziele, Nicht-Ziele und Milestones 2](#_Toc513563775)

[System Overview 2](#_Toc513563776)

[System Komponenten Beschreibung 2](#_Toc513563777)

[Hardware Komponenten 2](#_Toc513563778)

[Software Komponenten 2](#_Toc513563779)

[System Design 2](#_Toc513563780)

[Technologie Auswahl 2](#_Toc513563781)

[Systemarchitektur 2](#_Toc513563782)

[System Implementierung 2](#_Toc513563783)

[Client Software 2](#_Toc513563784)

[Server Software 2](#_Toc513563785)

[Web Monitoring Software 2](#_Toc513563786)

Einführung

Projektdefinition

Unser Projekt, der BackupMagican, ist eine Backupmanagmentsystem welche aus 2 großen Komponenten besteht:

Dem Server und Clients, dieses wird in einer Server-Client Architektur umgesetzt.

Ziel ist es Backups von einer Zentral, in diesem Fall der Server, zu steuern; Der Client ist der Zuhörer & Arbeiter

Um sich in das Netzwerk „einzubinden“ muss man sich entweder über einen Client oder über das Webinterface einen Account anlegen. Hat man die Clientsoftware noch nicht am Client installiert muss man dies nachholen.

Jeder Client der auf den angelegten User angemeldet ist und den Client mit dem User verknüpft hat, steht dem Server zu Verfügung, egal ob online oder offline. Der Server speichert sich sogenannte Aufträge und schickt sie wenn möglich an den Client.

Damit der Server immer weiß welche Client aktiv ist oder nicht authentifizieren sich die Clients beim Server, sie sagen einfach Hallo.

Sobald die Authentifikation abgeschlossen ist, der User eingeloggt ist, muss der User Clients zu seinem Account hinzufügen. Das passiert ganz einfach; Der Server merkt sich einfach die IP-Adresse des Clients, sobald dieser Vorgang abgeschlossen ist, kann der User gleich anfangen Aufträge zu schreiben. Man kann das entweder über das Webinterface machen oder lokal über den Client (greift auf das selbe Interface zu).

Aufträge sind kompakte JSON Files, zusätzlich gibt es auch normale Messages; Diese sind auch JSON Files. Diese Files werden über den body eines HTTP Packets zwischen Server und Host versendet.

Sobald der User seine Aufträge erstellt hat fängt der Server an mit den aktiven Clients zu kommunizieren und schickt diesen die Aufträge; Diese beinhalten Daten fürs Backup und Zugangsdaten zu Ziel-Host.

Der Client empfängt dieses und teilt das auch dem Server mit. Daraufhin verarbeitet der Client die Aufträge und teilt dies dem Server mit.

Der Server überprüft nun ob das Backup möglich ist (Ev. Cloud Speicher down, nicht genug Speicher etc), falls ja kriegt der Client das OK um die Daten an den Cloud-Service zu schicken.

Ist das Task fertig bearbeitet schickt der Client dem Server eine Nachricht das diese Task abgeschlossen ist. Falls nicht kriegt der Server einen LOG des Fehlers.

Das Webinterface selbst kann man auch über die Website ansprechen. Hierzu muss man sich nicht beim Client einloggen, wichtig ist nur das der Client online ist. Sobald dieser Umstand gegeben ist kann man Aufträge verteilen ohne Probleme.

Ziele, Nicht-Ziele und Zeitplan

**Projektziele**

1. Erstellung eines funktionalen Prototypen mit allen wichtigen Funktionen

2. Erstellung von Client-Backupsoftware

3. Erstellung von Webinterface

4. Erstellung von Serververwaltungssoftware

5. Kommunikations zwischen Server und Client

6. Senden von Files an Cloud Services

**Nicht Ziele (Professionelle Ansätze):**

1. Usermanagmentsystem für Authentifizierung (oder z.b OAuth)

2. Super ausgereifte Networksecurity

3. Extrem belastbare Server-software welche Millionen Nutzer aushält.

**Projekt-Zeitplan**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sep. | Okt. | Nov. | Dez. | Jan. | Feb. | März | April | Mai | Juni |
| Planung des Projektes |  |  |  |  |  |  |  |  |  |  |
| Software |  |  |  |  |  |  |  |  |  |  |
| Testen der Software |  |  |  |  |  |  |  |  |  |  |
| Dokumentation |  |  |  |  |  |  |  |  |  |  |

System Overview

System Komponenten Beschreibung

Hardware Komponenten

Software Komponenten

System Design

Technologie Auswahl

Systemarchitektur

System Implementierung

Client Software

Server Software

Web Monitoring Software