



Architektur

Studienarbeit FS-2020

24. März 2020

Autoren:

Mike SCHMID
mike.schmid@hsr.ch

Janik SCHLATTER
janik.schlatter@hsr.ch

Supervisors:

Prof. Stettler BEAT
beat.stettler@hsr.ch

Baumann URS
urs.baumann@hsr.ch

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Zweck

Dieses Dokument beschreibt die Architektur und liefert eine Übersicht über die Entscheidungen zum Design und der Architektur des Projektes.

Änderungsgeschichte

Datum	Version	Änderung	Autor
24.03.2018	1.0	Initial Setup	Janik Schlatter

Inhaltsverzeichnis

1	Designentscheidungen	1
1.1	Allgemein	1
1.2	Abstriche	1
1.2.1	GUI	1
1.2.2	Tests	1
1.2.3	Devices	1
1.2.4	Connection	1
1.3	Eingesetzte Technologien	1
2	Systemübersicht	2
2.1	NUTS	2
2.2	Netzwerk	2
2.3	Datenablage	2
3	Deployment	3
3.1	Deploymentdiagramm	3
3.2	Client	3
4	Datenspeicherung	4
5	Ausbauszenarien	5

1 Designentscheidungen

1.1 Allgemein

Bei der Architektur wurde vor allem Wert darauf gelegt, dass das Projekt in der Zukunft einfach erweiterbar ist und dass der Code möglichst abstrahiert von allen Technologien ist.

1.2 Abstriche

1.2.1 GUI

Auf das GUI wird vorerst komplett verzichtet, da dies vom Aufwand her zu gros ist, da zu viel Zeit für das Grundkonstrukt benötigt wird.

1.2.2 Tests

Grundsätzlich wird zum Beginn nur ein Test implementiert, sodass der Fokus ganz auf der Erweiterbarkeit steht. Falls es noch Zeit gibt werden noch mehr Tests implementiert.

1.2.3 Devices

Das automatische Erkennen von Devices wird im Rahmen von dieser SA weggelassen, da dies zeitlich nicht im Rahmen liegt. Das Projekt wird jedoch so angelegt, dass dies mit wenig Code-Umstellung noch einfach anzufügen ist.

1.2.4 Connection

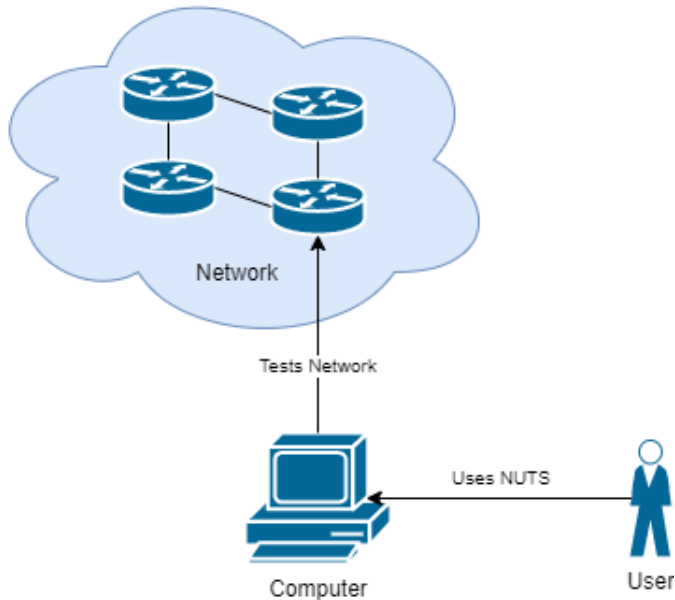
Grundsätzlich wird zum Beginn nur eine Connection implementiert, sodass der Fokus ganz auf der Erweiterbarkeit steht. Falls es noch Zeit gibt werden noch mehr Connections implementiert.

1.3 Eingesetzte Technologien

Das Projekt wird in Python umgesetzt. Zusätzlich wird Nornir verwendet um die Tests und die Verbindung zum Netzwerk zu vereinfachen. nornir hat mehrere Möglichkeiten auf ein Netzwerk zu verbinden (Napalm, Netmiko, Paramiko, Netconf).

2 Systemübersicht

Die Systemübersicht gibt einen Überblick über die verschiedenen Komponenten des Systems. Nachfolgend sind die einzelnen Komponenten detaillierter beschrieben.



2.1 NUTS

Dies entspricht unserem Client. Er besteht aus einer Python- App die mit Hilfe von Nornir und Netconf mit dem zu testenden Netzwerk kommuniziert.

2.2 Netzwerk

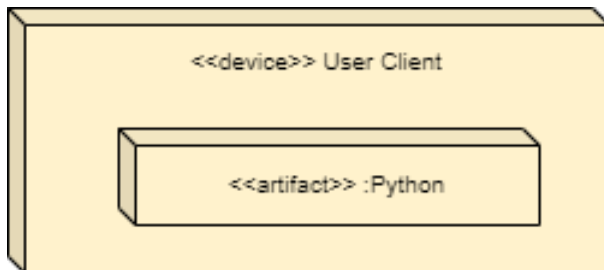
Dies ist das Netzwerk des Benutzers, welches dieser gerne automatisch testen möchte.

2.3 Datenablage

Alle unsere benötigten Daten werden mit Hilfe von YAML Files und Key-Value Stores abgespeichert.

3 Deployment

3.1 Deploymentdiagramm



3.2 Client

Der Client wird via PyCharm auf das Gerät des Benutzers verteilt. Später soll dies durch eine automatische Exe Datei passieren.

4 Datenspeicherung

Die Datenspeicherung wird mit Hilfe von YAML Dateien und key-value Stores realisiert. Deshalb ist es nicht nötig eine Datenbank anzulegen. Die hierbei entstandenen Files werden mit Hilfe eines FileHandlers eingelesen.

5 Ausbauszenarien