



Architektur

Studienarbeit FS-2020

26. 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
26.03.2020	1.0	Designentscheidungen	Mike Schmid

Inhaltsverzeichnis

1	Designentscheidungen	1
1.1	Architekturentscheidungen (AE's)	1
1.1.1	AE 1: Anwendung von Patterns für die Testerstellung	1
1.1.2	AE 2: Anwendung von Patterns für die Kommunikationsschnittstellen	1
1.1.3	AE 3: Auswahl der Testdefinitionssprache	1
1.1.4	AE 4: Weglassen einer Grafischen Benutzeroberfläche (GUI)	2
1.1.5	AE 5: Verwendung von Nornir	2
1.1.6	AE 6: Verzicht auf eine Datenbank	2
1.1.7	AE 7: Deployment als Python-Skript	3
1.2	Umsetzungsentscheidungen	3
2	Systemübersicht	4
2.1	NUTS	4
2.2	Netzwerk	4
2.3	Datenablage	4
3	Deployment	5
3.1	Deploymentdiagramm	5
3.2	Client	5
4	Datenspeicherung	6
5	Ausbauszenarien	7

1 Designentscheidungen

1.1 Architekturentscheidungen (AE's)

1.1.1 AE 1: Anwendung von Patterns für die Testerstellung

- Im Kontext der Erweiterbarkeit des zu entwickelnden Systems,
- damit weitere Netzwerktest einfach hinzugefügt werden können,
- wurde entschieden, dass für die Testerstellung das Strategy und Factory-Method Pattern angewandt wird,
- um die Instanziierungslogik von den einzelnen Testklassen zu entkoppeln.
- Dabei wird akzeptiert, dass die Implementation des Systems alles in allem komplexer, und die Erstellung umfangreicher wird.

1.1.2 AE 2: Anwendung von Patterns für die Kommunikationsschnittstellen

- Im Kontext der Erweiterbarkeit um weitere Netzwerkschnittstellen,
- damit das Hinzufügen von weiteren Schnittstellen vereinfacht wird,
- wurde entschieden, dass die Auswahl und Instanzierung von Verbindungen über ein Factory Method- und Strategy Pattern realisiert wird.
- um die Verbindungsauswahl vom System zu entkoppeln.
- Es wird dabei akzeptiert, dass das System komplexer wird und die Erstellung mehr Aufwand benötigt.

1.1.3 AE 3: Auswahl der Testdefinitionssprache

- Um die Testdefinitionen in einer möglichst menschenlesbaren Form zu halten,
- und die verwendbarkeit im zu entwickelnden System zu gewährleisten,
- haben wir entschieden, die Testdefinitionen in YAML zu verfassen,
- um ein Format zu verwenden, dass auch von Netzwerkleuten verwendet wird.
- Andere Technologien, wie JSON, XML usw. werden dabei voraussichtlich weggelassen,
- auch wenn diese in spezifischen Anwendungen wie die Speicherung von Daten geeignet wären.

1.1.4 AE 4: Weglassen einer Grafischen Benutzeroberfläche (GUI)

- Im Rahmen der zeitlichen Beschränkung der Arbeit,
- um uns auf die Implementation der Logik zu konzentrieren,
- werden wir kein GUI entwickeln.
- Stattdessen wird, wo nötig, eine Kommandozeileninteraktion implementiert.
- Es wird akzeptiert, dass die Benutzerfreundlichkeit des zu entwickelnden Systems dadurch eingeschränkt wird.

1.1.5 AE 5: Verwendung von Nornir

- Für die Kommunikation mit verschiedenen Netzwerken,
- um die Implementation möglichst zu vereinfachen,
- wurde entschieden, dass das System das Nornir-Automations-Framework verwendet wird,
- weil Nornir komplett in Python geschrieben ist und sehr gut mit Python-Code interagiert.

1.1.6 AE 6: Verzicht auf eine Datenbank

- Um Daten persistent zu speichern,
- damit sie vom System wieder abgerufen werden können,
- werden die Daten in form von Key-Value Paaren im YAML-Format abgelegt,
- und nicht in einer Datenbank,
- wobei ein Datenbanksystem für die zentralisierte Speicherung möglicherweise besser geeignet wäre.

1.1.7 AE 7: Deployment als Python-Skript

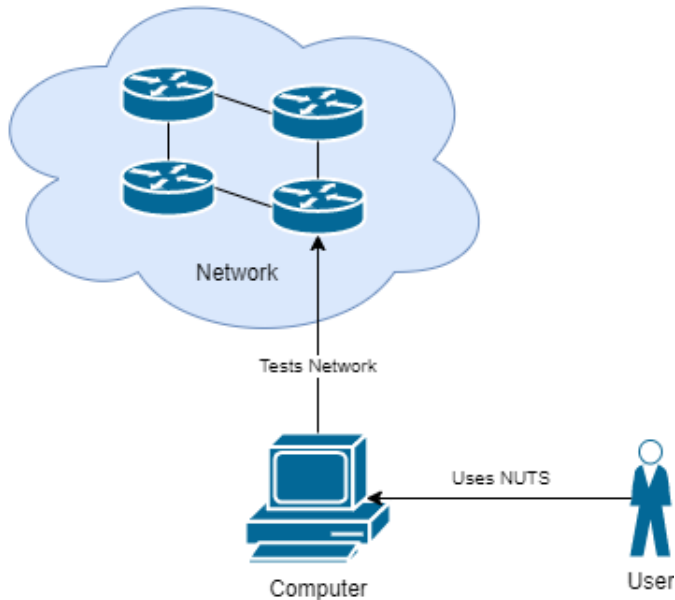
- Im Kontext des Deployments, der Auslieferung der Software,
- um das Programm auf möglichst vielen Plattformen einfach einsetzen zu können,
- wird das System in form eines Python-Scripts ausgeliefert,
- und nicht als ausführbare Datei im .exe Format,
- da Python auf den meisten Linux-Basierten Systemen und einigen modernen Windows-Systemen bereits vorinstalliert ist,
- womit sich die Installation/Ausführung des Programms stark vereinfacht.
- Die Folge, dass auf einigen Geräten für die Programmausführung zuerst Python installiert werden muss, wird akzeptiert.

1.2 Umsetzungsentscheidungen

Ein Test nach dem anderen Vorerst nur eine Verbingung Auslagerung der automatischen Deviceerfassung Unittesting

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

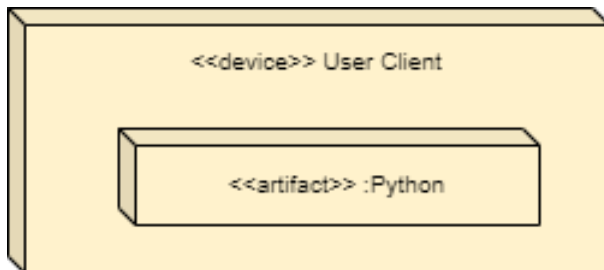
Dies ist das Netzwerk des Benutzers, welches dieser gerne automatisch testen möchte.

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