

Kraken

Ein Mediator für Datenkonflikte in der Netzwerkverwaltung

Einleitung: Netzwerke zu verwalten war und ist oft noch mühsame Handarbeit. Doch auch die Netzwerkwelt ist der Automatisierung unterworfen, mit dem Ziel, dass sich Netzwerke ähnlich wie Software programmieren lassen. Zu diesem Wandel gehört auch die Netzwerkverwaltung. Heute sind es häufig händisch gepflegte Textdateien, die den Zustand des Netzwerkes abbilden - eine Quelle für Fehler, Inkonsistenzen und Unübersichtlichkeit.

Eine sogenannte "Single Source of Truth" (SSoT) soll im Automatisierungszeitalter Abhilfe schaffen: Ein einziges Inventar, das alle Netzwerkinformationen als strukturierte Daten zentral an einer Stelle verwaltet. Existierende Lösungen versprechen bereits eine solche SSoT. Doch im Arbeitsalltag gibt es selten ein Inventar mit alleinigem Anspruch an die Informationshoheit. Gerade in grossen Netzwerken existieren parallel solche Inventare, die für sich verbindliche Datenquellen sind, und überlappend diese "Single Source of Truth" darstellen.

Ziel der Arbeit: An diesem Punkt setzt die vorliegende Arbeit an: Gibt es mehrere, voneinander unabhängige SSoT-Inventare, ist das Problem der inkonsistenten Daten wieder da. Eine Applikation soll Daten von diesen verschiedenen Inventaren sammeln, auf ein gemeinsames Format bringen, sie vergleichen und Inkonsistenzen beheben. Als Resultat liefert die Software den Zustand des kompletten Netzwerkes, wie es die Inventare abbilden. Ziel ist, bestehende SSoT-Inventare nicht abzulösen, sondern vielmehr das fehlende Gesamtbild zu liefern - eine vergleichbare Lösung auf dem Markt gibt es noch nicht.

Ergebnis: Kraken ist im Kern eine Python-Applikation, die auf Basis des Data-Engineering im Netzwerkbereich Daten zusammenführt und aufbereitet. Sie lässt sich über die Kommandozeile oder als Web-Applikation bedienen. Kraken kann sich über APIs auf zwei SSoT-Inventare verbinden, um deren Daten abzufragen. Von einem Inventar ist es auch möglich, die Daten offline einzulesen. Neben allen Daten kann auch ein Subset bezogen werden (Filterfunktion), beispielsweise nur Geräteinformationen, die zu einem gewissen Standort gehören. Kraken normalisiert die gehaltenen Daten auf eine eigens entwickelte Datenstruktur, erkennt Differenzen und löst diese auf. Dies geschieht auf zwei Varianten: Entweder automatisch über eine interne Logik von Kraken, oder manuell gesteuert von der Nutzerin über die Web-Applikation. Für den zweiten Fall generiert Kraken zuerst einen Konfliktreport, der alle gefundenen Differenzen speichert. Er kann jederzeit abgerufen werden, um daran weiter zu arbeiten. Sind alle Differenzen geklärt, erzeugt Kraken aus all den Daten eine End-Konfiguration - das Gesamtbild des Netzwerkes.

Für eine künftige Weiterentwicklung muss vor allem die Datenstruktur überarbeitet werden, auf die Kraken die Daten normalisiert: Sie erschwert, neue Systeme an Kraken anzubinden oder die Struktur selbst zu erweitern. Ausserdem holt Kraken zwar Daten von den SSoT-Inventaren, synchronisiert die Daten jedoch nicht zurück. Eine weitere Ergänzung wäre, dass Kraken die End-Konfiguration auch auf die Inventare zurückschreiben könnte, um dort Fehler zu korrigieren.

Diplomanden



Julia Fritsche



Méline Sieber

Examinator

Prof. Beat Stettler

Experte
Basile Bluntschli,
WLAN-Partner.com AG,
Zürich, ZH

Themengebiet
Networks, Security &
Cloud Infrastructure,
Software

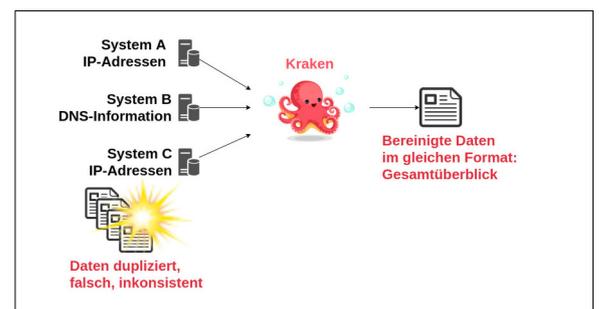
Kraken-Logo

user2104819 von Freepik.com



Konzept, wie Kraken funktioniert

Eigene Darstellung



Verwendete Technologien in Kraken

Eigene Darstellung

