

# Das Projekt: SNET-NG

## Safety Net Next Generation (SNET-NG)

Das SNET-NG erhält die Positionsdaten von Flugzeugen (Tracks) und soll potenziell gefährliche Situationen erkennen und eine Warnung an die Konsole des Lotsen senden. Es ersetzt die bereits bestehenden Systeme STCA und CoSNET. Die Features von beiden Programmen sollen in SNET-NG in einem Programm vereint werden (STCA, APW etc. siehe Grafik). SNET-NG besteht aus Microservices und läuft in einem Kubernetes Kluster. Gemeinsame Dienste wie die Verarbeitung der Tracks, das bereit stellen von Geo-Informationen und das abschließende Senden der Meldungen werden für alle einzelnen Features bereit gestellt. Jedes Feature ist sein eigener Dienst. SNET-NG nutzt C++ für das Programm selbst und Python und Bash-Skripte für unterstützende Tools.



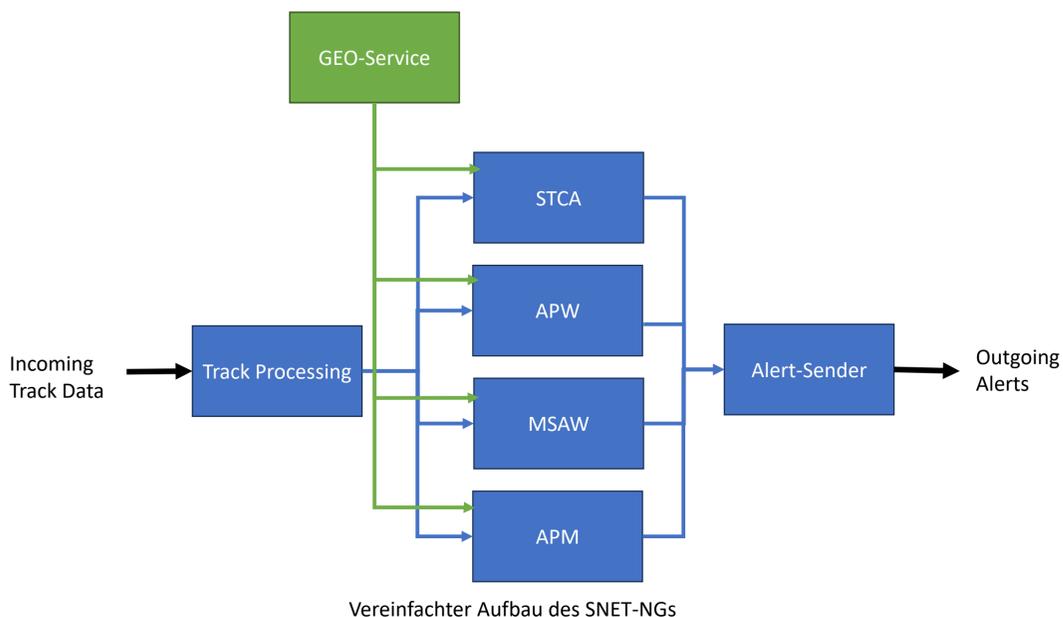
### Die einzelnen Features von SNET-NG

**STCA**  
Short Term Conflict Alert (STCA) assists the controller in preventing collision between aircraft by generating, in a timely manner, an alert of a potential or actual infringement of separation minima.

**APW**  
Area Proximity Warning (APW) warns the controller about unauthorised penetration of an airspace volume by generating, in a timely manner, an alert of a potential or actual infringement of the required spacing to that airspace volume.

**MSAW**  
Minimum Safe Altitude Warning (MSAW) warns the controller about increased risk of controlled flight into terrain accidents by generating, in a timely manner, an alert of aircraft proximity to terrain or obstacles.

**APM**  
Approach Path Monitor (APM) warns the controller about increased risk of controlled flight into terrain accidents by generating, in a timely manner, an alert of an unsafe aircraft flight path during final approach.



## Meine Aufgaben

### 8192 Functional Volumes

SNET-NG kann für verschiedene Sektoren (Functional Volumes) unterschiedlich konfiguriert werden, es können z.B. für manche Sektoren bestimmte Features deaktiviert werden.

#### Problem:

- Anforderung: SNET-NG soll 8192 Functional Volumes funktional verwenden können
- Functional Volumes werden in einer GML-Datei (Geographic Markup Language) konfiguriert
- Ein GML mit 8192 Functional Volumes wird zum Testen der Anforderung benötigt.

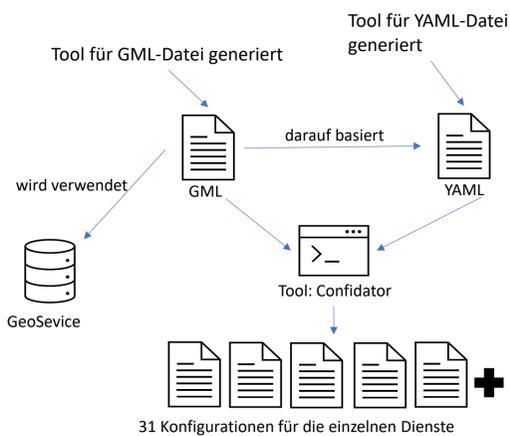
-> Ein Python-Tool soll diese GML-Datei automatisch generieren.

#### Verlauf des Projektes:

- Passende YAML-Datei für die Zuweisung der Features auf Volumes wird benötigt.
- Zweites Tool dafür wird erstellt
- Mit dem vorhandenen Tool Confidator lässt sich daraus eine Konfiguration, die aus 31 einzelnen Konfigurationen besteht, für den ganzen Dienst erstellen



4096 Sektoren über einer Deutschland Karte



### Availability Measurements

Laut Anforderung soll SNET-NG eine Verfügbarkeit von 99,85 % haben. Prometheus wird verwendet, um im Kubernetes Kluster Metriken zu sammeln. Aus diesen Metriken kann die Readiness-prob genutzt werden, um zu prüfen, zu welcher Zeit ein Dienst verfügbar war. Gesamtzeit durch verfügbare Zeit ist die Verfügbarkeit. Zuerst wird die Verfügbarkeit der einzelnen Dienste bestimmt, dann wird die Verfügbarkeit eines ganzen Features errechnet, dazu werden der Dienst und alle Dienste, die genutzt werden, miteinander multipliziert. Die Redundanz der Dienste wird dabei berücksichtigt.

Die Formel für das STCA-Feature:

$$Verfügbarkeit_{Track Processing} \times Verfügbarkeit_{STCA} \times Verfügbarkeit_{Alert Sender}$$

Dafür werden Prometheus Abfragen ausgearbeitet. Diese Abfragen werden in Grafana dargestellt.

#### Beispiel des Grafana Dashboards:

