**Triage Alert Suppression**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Level** | **Date** | **Author** | **Changes Made**  **(Include Change Request # if applicable)** |
| 1 | 21-10-2021 | Vinoth Ravi |  |
|  |  |  |  |

**Table of Contents**

[1. Problem Statement 4](#_Toc36832465)

[2. Assumptions 4](#_Toc36832466)

[3. Existing System and Design 4](#_Toc36832467)

[4. Solution Design](#_Toc36832468) 5

[4.1. Solution Approach](#_Toc36832468) 5

[4.2. High level architecture diagram](#_Toc36832468) 6

[4.3. Technology stack](#_Toc36832468) 7

[4.4. Use Case diagram](#_Toc36832468) 8

[5. Proposed Design](#_Toc36832468) 9

[6. Integrations](#_Toc36832473) 13

[7. WebServices](#_Toc36832473) 14

[8. Component List](#_Toc36832472) 14

[9. Messages](#_Toc36832473) 14

[10. Changes](#_Toc36832474) 14

[11. External Interfaces](#_Toc36832475) 14

[12. Internal Interfaces](#_Toc36832476) 14

13. Technology Stack…….……………………………………………………………………………………………………………………………………………. 15

# Problem Statement

Currently, alert suppression in Triage during OS patching is a manual task and time consuming. During OS Patching, to suppress the alert,we need to provide Node, Ticket, Schedule type, Schedule start and End date time and Reason manually.

# Assumptions/Constraints

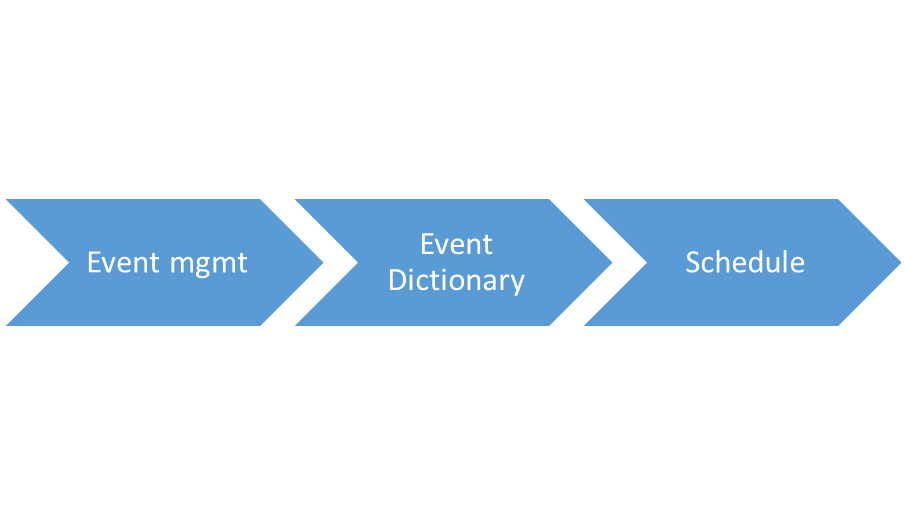
# Basic authentication

# Existing System and Design

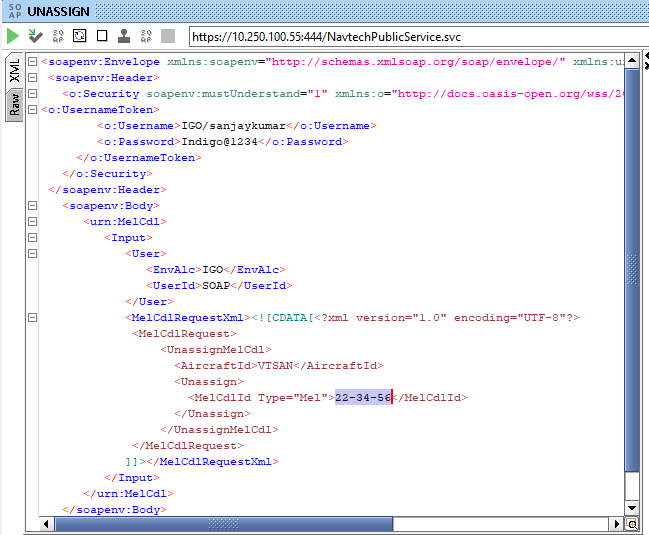
* **Current Alert Supprestion.**

In Triage toolbox, Navigate to Event Mgmt, then Event dictionary, then Schedule screen.

Provide required info like provide Node, Ticket, Schedule type, Schedule start and End date time and Reason manually.



Triage Toolbox

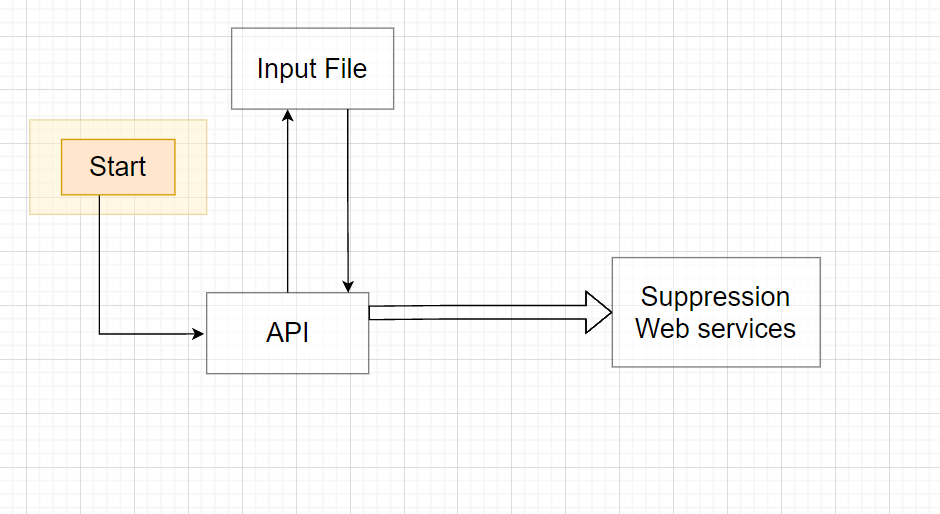
****

# 4.Solution Design

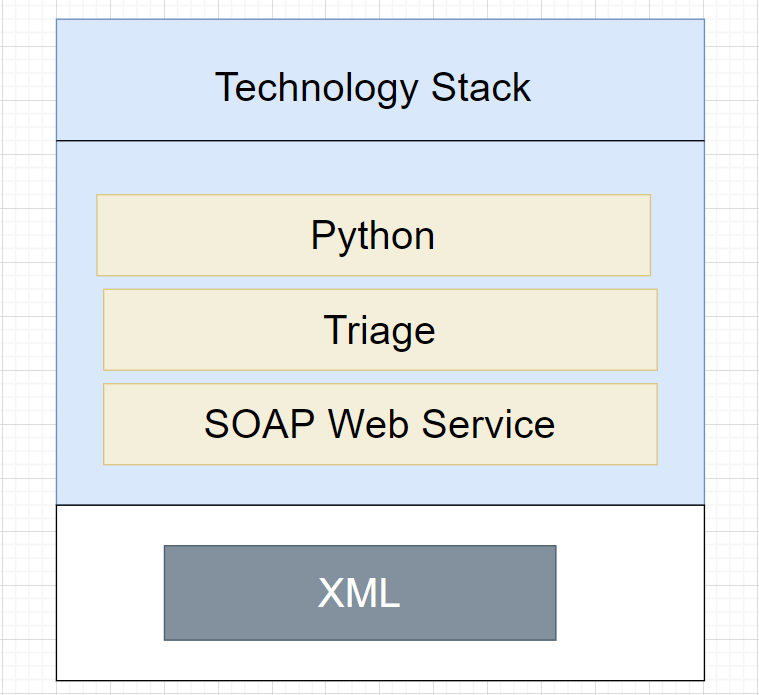
## 4.1 Solution Approach

* + 1. Create Python API script to call the suppression webservice.
    2. Create the input file with required parameters.
    3. Call suppression webservice from Automation API to suppress the alerts.

## 4.2 High Level Architecture Diagram:



## 4.3 Technology Stack

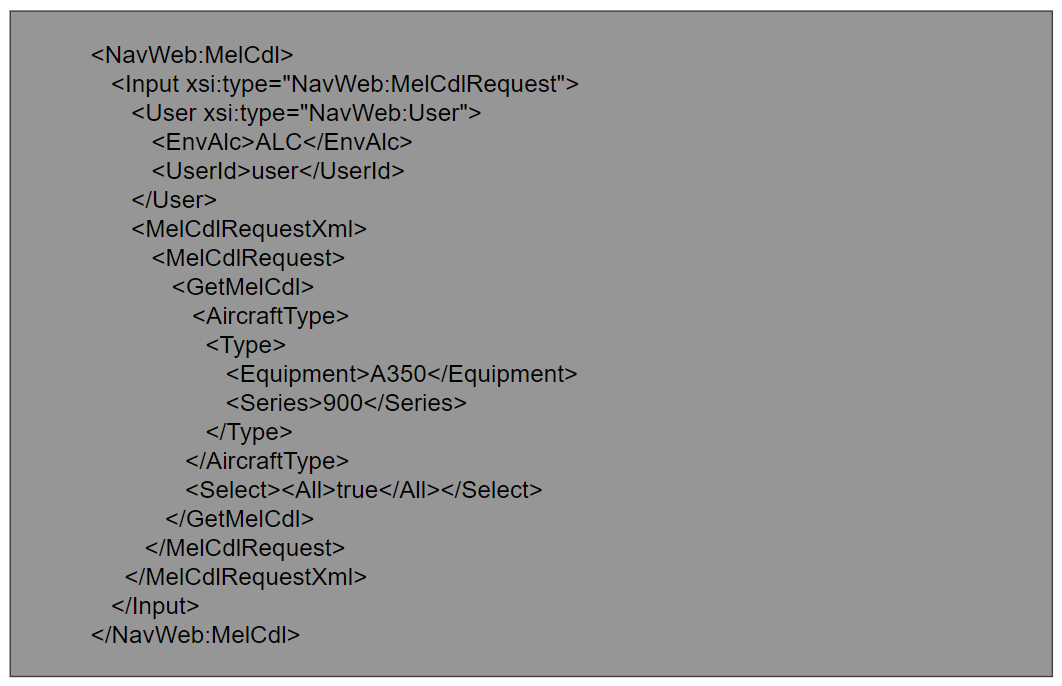


# 5.Proposed Design

1. On real time basis webservice will run which would capture the latest MEL/CDL information coming on aircrafts and push the data into the NavBlue system
2. Navblue will consume the webservice to reflect the latest MEL/CDL in their FRC page whenever the flight plan is computed and released.
3. The interface to send latest, MEL/CDL data from MCC application to NAVBLUE on real time basis as & when MEL/CDL are invoked/revoked. This activity will happen as and when a new data is released by MCC team, the same is updated in NavBlue.
4. Navblue to be updated with latest MEL/CDL data at least 2 hours 30 mins prior till D-45
5. Revoked MEL/CDL to be removed from FRC
6. The most recent MEL / CDL to be taken and pushed to Navblue FRC.
7. The Latest MEL/CDL data would be captured by interface and pushed to NavBlue.
8. MEL/CDL data can be retrived by using XML entity encoding or CDATA enclosure.

## XML Format:

## XML entity encoding



## CDATA enclosure:



## 6. Integrations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Application Name** | **Type of Data** | **Frequency** | **One/Two way Transfer** | **Additional Notes** |
| Triage Tollbox | MEL/CDL reference numbers, tail no., | Near real time | One way |  |

## 7.WebServices

N-Flight planning Web services is used to get latest MEL/CDL information.

# 8.Component List

|  |  |  |
| --- | --- | --- |
| Component Name | Location | New / Existing |
|  |  |  |

# 9. Messages

NA

# 10. Changes

NA

# External Interfaces

Navblue

# Internal Interfaces

NA

# Technology Stack

|  |  |
| --- | --- |
| **Technology Stack** | **Details** |
| Languages | Python |
| Technologies |  |
| Development Tool / IDE |  |
| Framework | Webservice |
| Databases |  |
| Persistence |  |
| Source Control | Git |
| Web Servers / App. Servers(Installation and configuration) |  |
| Development Tools |  |
| Monitoring tools / methodology |  |
| Review tools |  |
| Built Tools | NA |
| Source Code Control Tools | Git |
| Testing tools / methodology |  |
| ITSM Tool | Manual Testing |
| Methodologies /Frameworks | Web services |
| Change / Process  Management tools |  |
| Operating Systems |  |