ABSTRACT
Against the background that all future air traffic participants are requested to act as System Wide Information Management (SWIM) communicating sub-systems by the future Single European Sky Air Traffic Management Research (SESAR) SWIM 'Intranet for ATM' concept, SWIM compliance of future ATM information sharing software modules is of utmost importance. The paper describes Tcl/Tk usage in this context, focusing on a prototype referring future space / air traffic integration needs.

1. INTRODUCTION
The SWIM concept can be summarized to its role of providing the right data to the right target(s) at the right time, associated by the connectivity between the right sources and sinks. Strictly obeying to this elemental rule, generates the need that future commercial spacecraft will have fulfill SWIM compliance, especially against the background of the seamless integration into normal air traffic and the generation and distribution of emergency information in case of a major hazard event [1].

2. PROTOTYPE
2.1 General Architecture
The prototype’s general architecture is shown in Figure 1. It is realized as a Hypertext Transfer Protocol Secure (HTTPS) web server, containing a provisional hazard area model which assumes a Space Shuttle like trajectory of the hypothetical spacecraft. The model performs inter-/extrapolation based on the United States National Aeronautics and Space Administration’s (NASA) Columbia space shuttle accident debris data [3].

2.2 SESAR SWIM / FAA SWIM
The Europe / U.S. SWIM harmonization architecture is presented in Figure 2. It demonstrated the connection between SESAR SWIM and the United States Federal Aviation Administration’s (FAA) SWIM concept of Global Enterprise Messaging Service (GEMS), where Service by NEAR (SBN) was one representational implementation of a GEMS provided by the Next Generation ERAU Applied Research (NEAR) lab. The demonstration use case included Flight Object data exchange using SWIM’s data interchange standard Flight Information Exchange Model (FIXM) version 3.0.1 between an FAA research and development Flight Object Exchange Service, an airline Electronic Flight Bag (EFB), and the SpacecraftReentryHazardAreaServer, consuming the aircraft state data within the Flight Object FIXM message and published its output using SWIM’s data interchange standard Aeronautical Information Exchange Model (AIXM) version 5.1 for consumption and display by other interested parties.

3. REALIZATION
The SpacecraftReentryHazardAreaServer prototype is developed in 100 % Tcl, benefiting from the following packages:
- TclHttpd as the web server,
- Web Services for Tcl for the server side web service creation on top of TclHttpd,
- TclTLS for using HTTPS,
- Rpcvvar for complex data type definitions,
- CriTcl for improved performance by the usage of C code runtime embedding.

Figure 1. General architecture of the SpacecraftReentryHazardAreaServer

Figure 2. SESAR SWIM / FAA SWIM demonstration architecture
- BaseXClient-Tcl for using the BaseX server protocol to communicate with the hazard area model database server.

Five Tcl modules build the prototype’s structure (Figure 3). The SpacecraftReentryHazardAreaServer.tcl main module represents the generic start component, setting up the general webserver frame and loading the hazard area model database. Definewebservices.tcl is just a mediator for sourcing services’ code files, only one in the prototype’s case, the SpacecraftReentryHazardAreaService.tcl module. Its core functionality refers to the web service request / response handling in the service procedure, mainly governed by XML in parsing and XML out generation of the FIXM in and AIXM out payloads. The ZoneCalc.tcl module is responsible for calculating the hazard zone’s coordinate vertices, derived from the spacecraft’s 3d position and orientation, taking into account one-dimensional table interpolation in the hazard area model’s database representation, prepared for future vehicle specific tabulated values containing aerodynamic coefficients data. C_coding.tcl contains alternative calculation procedures in embedded C code for the CriTcl C Runtime in Tcl package [4][5], resulting in a more than 30% performance improvement by replacing the CalculateHeading and CalculateHazardZone Tcl procedures by C code equivalents [6].

![Figure 3. Sourcing structure of the prototype](image)

The principle is shown in Figure 4 to Figure 6, taking the CalculateHeading procedure as an example. Its equivalent in C only needs integration in the Tcl source code and can be easily switched to this alternative by commenting out the invocation of the associated Tcl procedure in ZoneCalc.tcl.

```tcl
# Figure 4. CalculateHeading procedure in Tcl
```

---

**4. OUTLOOK**

Although SESAR SWIM will implement functions that secure information exchanges by infrastructure security, the development of a SecSWIM security application is planned. It will act as a SWIM participating data analysis sub-system, able to perform attack identification by SOAP message classification taking into account SOAP characteristics like service identifier, message length, header length, message transaction duration, etc. The solution is anticipated to benefit from a module based serial-in-the-large-iterative-in-the-small machine learning big data river approach (Figure 7), incorporating a data river ➔ data lake module sequencing (Figure 8).

![Figure 7. SeeSWIM module](image)
SecSWIM is foreseen to be developed in Tcl, too. Benefit potential referring the usage of the kafkatcl package [7] and the TensorFlow C++ API (being used inside CriTcl) [8] is under examination.

5. REFERENCES


