

# UTM UAS Service Supplier Development

## Sprint 1 Toward Technical Capability Level 4

*Joseph L. Rios  
Ames Research Center, Moffett Field, California*

*Irene S. Smith  
Ames Research Center, Moffett Field, California*

*Priya Venkatesen  
SGT Inc., Moffett Field, California*

*David R. Smith  
SGT Inc., Moffett Field, California*

*Vijayakumar Baskaran  
SGT Inc., Moffett Field, California*

*Sheryl Jurcak  
SGT Inc., Moffett Field, California*

*Randy Strauss  
SGT Inc., Moffett Field, California*

*Shankar Iyer  
SGT Inc., Moffett Field, California*

*Punam Verma  
Universities Space Research Association, Moffett Field, California*

## NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

**TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

**TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

**CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

**CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

**SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

**TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

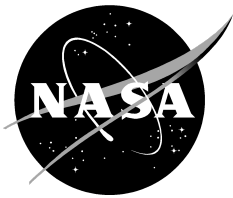
For more information about the NASA STI program, see the following:

Access the NASA STI program home page at <http://www.sti.nasa.gov>

E-mail your question to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)

Phone the NASA STI Information Desk at 757-864-9658

Write to:  
NASA STI Information Desk  
Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199



# UTM UAS Service Supplier Development

## Sprint 1 Toward Technical Capability Level 4

*Joseph L. Rios  
Ames Research Center, Moffett Field, California*

*Irene S. Smith  
Ames Research Center, Moffett Field, California*

*Priya Venkatesen  
SGT Inc., Moffett Field, California*

*David R. Smith  
SGT Inc., Moffett Field, California*

*Vijayakumar Baskaran  
SGT Inc., Moffett Field, California*

*Sheryl Jurcak  
SGT Inc., Moffett Field, California*

*Randy Strauss  
SGT Inc., Moffett Field, California*

*Shankar Iyer  
SGT Inc., Moffett Field, California*

*Punam Verma  
Universities Space Research Association, Moffett Field, California*

This report is available in electronic form at  
<https://www.sti.nasa.gov/>

## 1. Executive Summary

NASA's UAS Traffic Management (UTM) Project has been tasked with developing concepts and initial implementations for integrating and managing small unmanned aircraft systems (UAS) into the low altitude airspace. To accomplish this task, the Project planned a phased approach based on four Technical Capability Levels (TCLs). As of this writing, TCL4 is currently in development for a late Spring 2019 flight demonstration. This TCL is focused on operations in an urban environment and includes the handling of high density environments, large-scale off-nominal conditions, vehicle-to-vehicle communications, detect-and-avoid technologies, communication requirements, public safety operations, airspace restrictions, and other related goals.

Through research and testing to date, NASA has developed an architecture for UTM that depends on commercial entities collaboratively providing services that are traditionally provided by the Air Navigation Service Provider (ANSP) in manned aviation. A key component of this architecture is the UAS Service Supplier (USS), which acts as a communications bridge between UAS operators and the ANSP when necessary. In addition, the collection of USSs form a USS Network to collaboratively manage the airspace through the sharing of data and the adherence to a standard or set of standards required to participate in this USS Network. This document provides a record of the first step in the development of interoperable USSs that will ultimately support TCL4 flight testing and formalization of the overall UTM concept. To develop these USSs and the underlying specifications for them, NASA has planned a series of "Sprints" to work with industry partners in implementing the features and proposed specifications for USSs to participate in TCL4. This report describes Sprint One.

In this Sprint, the focus was on establishing a baseline for the Application Programming Interfaces (APIs) and their associated data models. In addition, the concept of UAS Volume Reservations (UVR) (areas that impose restrictions on sUAS that are allowed to operate) was tested. NASA provided the specifications and iterated on them with partners while implementers developed to those specifications. NASA then tested each partner's implementation to ensure compatibility with all other implementers. This process helped all stakeholders gain confidence that the foundation for future Sprints was solid.

## 2. Background

For a comprehensive overview on the UTM concept and NASA's efforts to develop it in partnership with the FAA, the UTM Project currently has a document repository [NASA 2018b] that contains documentation of many UTM R&D capabilities. In terms of singular references, the two Concept of Operations papers, one from NASA in 2016 [Kopardekar 2016] and one from the FAA in 2018 [FAA 2018], are the best current options for gaining an initial understanding of UTM.

At a high level, UTM is focused on access to the low-altitude airspace for beyond visual line-of-sight (BVLOS), commercial, sUAS operations. To date, such access has been governed by lengthy, one-off applications for waivers or are limited to restricted airspace. To make widespread sUAS operations safe, fair, efficient, and routine, a set of services to manage the airspace will be required. The functional decomposition of these services in terms of which stakeholders provide which services is an open research question. The major drivers of these questions come from the scale of operations (potentially orders of magnitude more than current manned operations[FAA 2018b]) and the automation envisioned to support that scale. As an illustrative example, flight planning services will likely be an operator function while authorization

to access the airspace will likely be an ANSP function, while some conflict management services may reside with the vehicle itself. The NASA UTM Project has provided initial insights into how such an architecture should look in the future. [Kopardekar 2016]. The Project has also provided inspiration and guidance for systems already providing services to sUAS that did not exist prior to NASA's efforts. The primary example is the FAA's Low Altitude Authorization and Notification Capability (LAANC), which "provides access to controlled airspace near airports through near real-time processing of airspace authorizations below approved altitudes in controlled airspace" [LAANC 2018].

A key component of the UTM architecture are USSs. A USS is a state-appointed or third party operated system that will provide service options that are similar to what is traditionally offered to manned operations solely by a state-appointed ANSP. These USSs are components that do not have an exact analogy in manned aviation, but are inspired by elements such as Airline Operations Centers and Flight Service Stations. One of the key differences between those examples and a USS is the set of responsibilities and capabilities envisioned for a USS. A USS will likely provide services that are typically only provided by state-appointed ANSPs in manned aviation. These might include strategic deconfliction of operations, conformance monitoring of live operations, and other services. Many of these services are to be provided in a collaborative and mostly automated manner via communications between USSs. In UTM, there will not be a central authority providing guidance in every traffic management decision, as there typically is in manned aviation. The overall architecture (evolved from [Kopardekar 2016]) is shown in Figure 1 below.

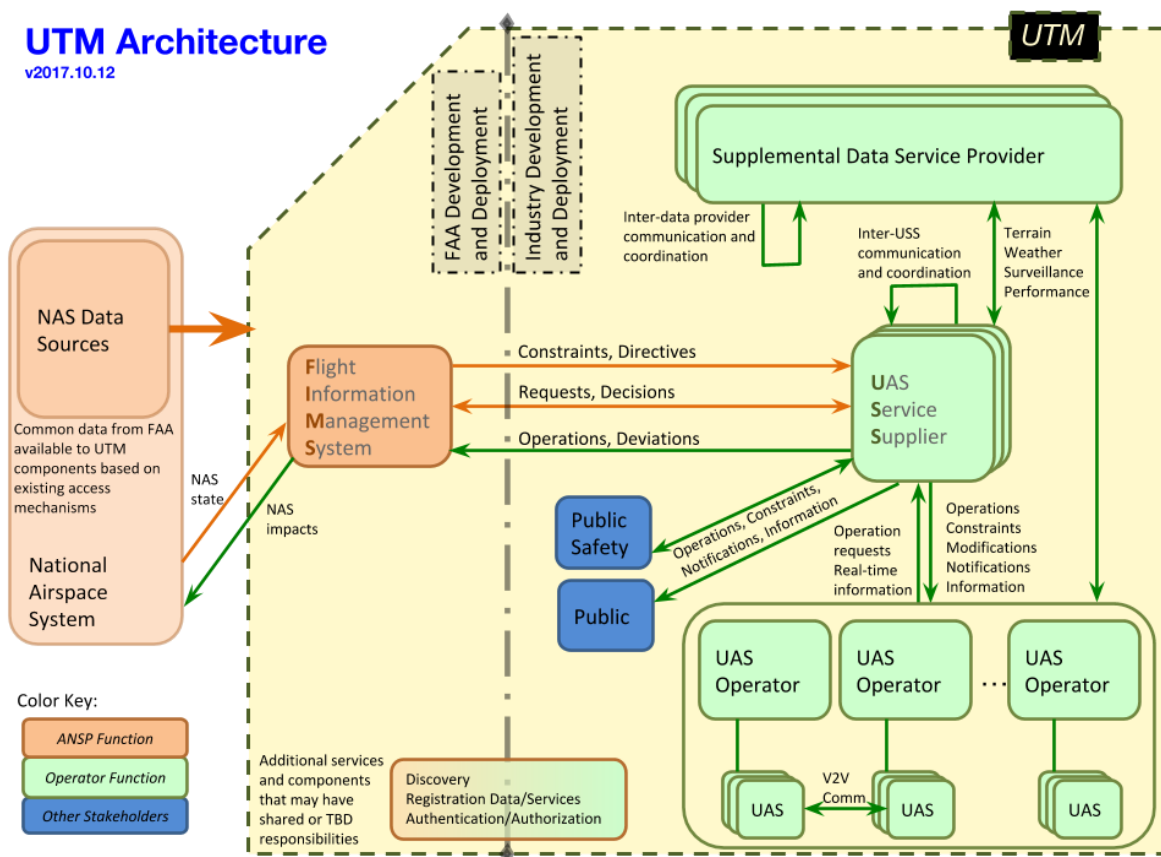


Figure 1. UTM architecture.

To ensure compatibility and interoperability for USSs, NASA has developed a set of API and protocols for exchanging data [NASA 2018]. These APIs describe many of the connections in Figure 1. The subsequent sections of this document are a description of how NASA and its partners are preparing for flight activities with enhancements to these APIs and protocols.

### **3. Progression of USS Development**

The check out process for components of UTM is important for several reasons. Primarily it ensures that the various components are built to the specification and will be appropriately interoperable. Secondly, the process allows for conceptual shakeout by performing data exchanges between components, researchers can check whether the defined use cases are being met. The key components in the initial phase of UTM development are the UAS Operator Client, the USSs, and FIMS. These three components are central to all of the key use cases in UTM and their appropriate interconnection must be verified to ensure proper execution of simulations and flight tests.

In this section, a brief summary of how USSs were developed and “checked out” in previous UTM TCL flight activities is provided. The evolution of this process provides will ultimately provide insight into how an operational version of a USS should be evaluated in the future.

#### **3.1. TCL1: August 2015 and Spring 2016**

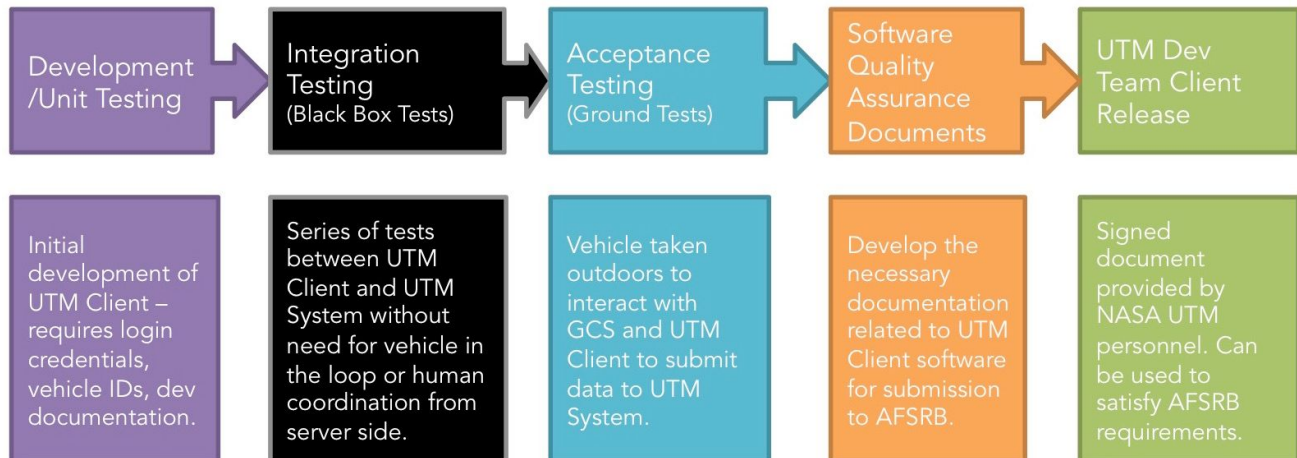
The TCL1 flight demonstration took place in August 2015 [Johnson 2017b] and were followed up with a distributed demonstration executed by FAA designated UAS test sites [Rios 2016]. At that time, there was no concept of a USS. UTM operations were envisioned, at that time, to be managed through a centralized service guaranteeing strategic deconfliction of operations and alerts to operators about conditions in the airspace. As such, there was no need for developing a USS-USS communication scheme and no process for partners to develop USSs. The closest analogy during TCL1 was the process of partners to develop a client to UTM which acted as a communication bridge directly between the ground control station and UTM. The checkout process for those clients was not formalized. This checkout process for those clients was not formalized nor uniform; NASA test directors confirmed each individual client could reliably participate in the TCL1 flight demonstration. Individual partners collaboratively tested their clients with the UTM server until the NASA side felt comfortable the client was doing the right thing.

Overall the checkout and testing process demonstrated the viability of a collaborative approach to partner development. Lessons learned included clarifying requirements for a checkout process for all interacting systems. In addition, the NASA team began to learn about which architectural decisions were more immutable and long standing than others, which led to better decision making for the future of UTM development. For example, the move from eXtensible Markup Language (XML) data exchanges to Javascript Object Notation (JSON) was an important move determined through integration testing of various UTM components since JSON is less verbose, more flexible, and generally has better tool support in many programming languages. Documenting these exchanges as a formal API was solidified at this early stage as well. Finally, this effort also established an initial reference implementation of a UAS Operator to UTM interface, which proved an important template for future research and development efforts.

### 3.2. TCL2: October 2016 and Spring 2017

TCL2 was performed in two key steps. First, NASA flew its own demonstration in Oct 2016 [Johnson 2017][Homola 2017]. The architecture was moving toward a distributed and federated network of USSs, but in this demo there was only a single USS provided by NASA. So the checkout process looked much like TCL1 wherein partners were checked as UAS Operator Clients against NASA's USS. There were no USS-USS communications tested.

Based on experience in TCL1, the following flow was used and described in depth in an internal testing document shared with client implementers. It involved performing black box tests against the UTM system and recording artifacts of that exchange. Those artifacts were submitted to the UTM team for analysis to ensure proper formatting and processing of data exchanges. Upon successful completion of that step, more detailed, interactive tests including some with vehicle hardware in the loop were completed with the NASA team and the client implementers. The testing flow is described in Figure 2 below extracted from the internal testing document. Note that some of the requirements for testing were driven by NASA's Airworthiness Flight Safety Review Board (AFSRB) as referenced in Figure 2. These reviews encompassed all components of the flight test including the aircraft itself, the ground control station (GCS), and related components, hence their reference in Figure 2.



**Figure 2. TCL2 UAS Operator Client checkout flow.**

The completion of this flow set the stage for further testing in the Spring of 2017 performed by six FAA-designated UAS Test Sites under contract with NASA. In this activity, each of the sites independently developed partnerships with USS implementers. A collaborative demonstration hosted by NASA acted as a check that USS implementations were sufficient [Rios 2017]. This demonstration was the first of its kind in exercising a distributed and federated USS Network.

### 3.3. TCL3: Spring 2018

Flight tests for TCL3 were performed between March and May of 2018. A different approach for USS checkout was attempted in TCL3 due to the increased requirements related to USS-USS communications in the management of the airspace.

To aid in this process, NASA established a sandbox wherein operations were submitted and executed on a regular, automated schedule by the NASA USS. This allowed other USS



implementers to interact with this automated USS to begin testing their systems. The automated USS would emit operation plans, position reports, and various messages to other USSs that registered themselves per UTM protocols. Upon reaching a self-evaluated “sane” state with their implementations, USSs could then move on to the next phase of checkout.

USSs were then asked to collaboratively checkout each others’ systems over a series of specific tests. This was accomplished by having at least two other USSs attest that another USS performed a given exchange appropriately. So, for example, if the test involved the exchange of position information, a USS would perform position data exchanges with two other USSs. The USSs tracked their checkout process on a shared spreadsheet by having the “verifying” USS fill in cells to indicate the USS under test performed to expectation.

Finally, there was a human checkout process wherein the USS under test would interact via phone with a NASA engineer. That engineer would manage a NASA USS to interact with the USS under test to check out certain features to further ensure compatibility. These tests were not extensive, but added additional confidence that the implementations were reasonable.

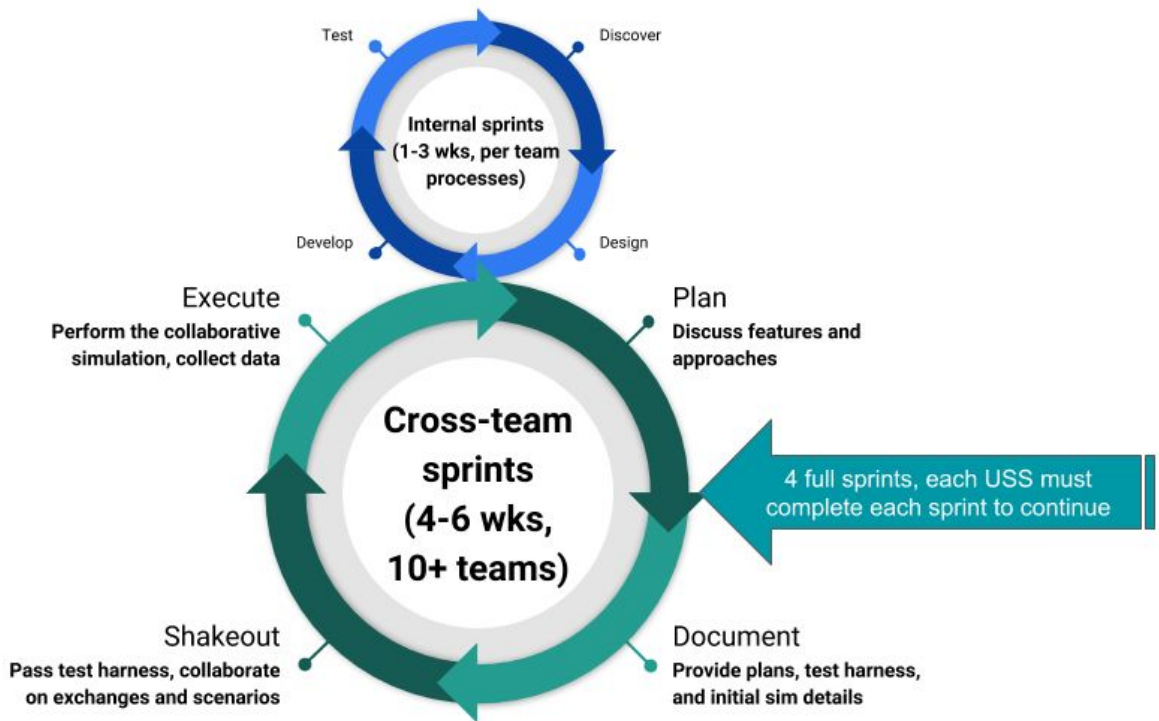
### **3.4. Progression Summary**

The approaches for TCLs 1-3 described in the section above left several gaps in the checkout process. In the TCL2 flow, NASA needed to be more formal with individual USSs regarding verification at the cost of interoperability with other USSs. In the TCL3 flow, NASA provided a great deal of freedom to the USS implementers leading to incomplete requirement satisfaction and poor interoperability due to “cliques” of USSs checking each other out, but not interacting with more USSs. This also impacted the quality of data received from testing due to differences in implementations and approaches.

## **4. TCL4 USS Checkout Processes**

Internalizing the lessons learned from TCLs 1-3, and planning for TCL4, NASA decided to take a more proactive and prescriptive approach to development. A plan was developed to implement features as a cohort of USSs over the course of many months. The features were broken into 4 major “Sprints” with a capstone activity in each sprint being a collaborative simulation exercising the features.

This activity is collaborative in that plans are discussed with all USS implementers, feedback is provided by those implementers, plans are documented, and a schedule is agreed upon for testing and execution within each sprint cycle. Each team may have its own internal sprint cycles to achieve the goals of the cross-team sprints. For example, the NASA USS development team uses 2 week sprints to implement and test features to prepare for the collaborative simulation. The process is depicted in Figure 3.



**Figure 3. Development process for TCL4 USSs..**

Since feature implementation may change as lessons are learned and goals for TCL4 are formalized, we'll only focus on the Sprint 1 features and process within this Technical Memo.

## **5. Sprint 1 Overview**

In this section we provide an overview of the initial sprint, including the participants and overall goals.

### **5.1. Participants**

Nine industry USS implementers participated in Sprint 1. They are listed in alphabetical order in Table 1 along with their respective call sign used during the activity:

**Table 1. Sprint 1 industry participants**

|      |                      |
|------|----------------------|
| ARMP | Airmap               |
| AROS | AiRXOS, A GE Venture |
| ONES | OneSky               |
| AMZN | Amazon Prime Air     |
| ANRA | ANRA Technologies    |
| AVIS | Avision Robotics     |
| ROCK | Rockwell Collins     |
| SIMU | Simulyze             |
| UBER | Uber                 |

There were also two NASA implementations (NUSS and AOLN) of a USS that participated, giving a total of 11 USS implementations available to evaluate requirements, performance, and overall interoperability. This report will not provide details on the performance of any given USS in an identifiable manner.

## **5.2. Goals**

The following list of goals/features were agreed to by NASA and industry USS implementers and targeted during Sprint 1. Further context for these items is provided in the subsequent subsections.

1. Achieve baseline of the core USS APIs and models
2. Enforce protection of endpoints per API docs
3. Encourage use of single-scoped access\_tokens
4. Exercise new credential naming based on DNS names
5. Test initial concept for message signing for integrity and authentication
6. Prove concept of USS-managed constraints

### **5.2.1. Achieve baseline of the core USS APIs and models**

Some US implementers were just beginning to partner with NASA UTM and needed to get an initial implementation ready. Partners from TCL3 needed to incorporate API and model changes from that event to the new baseline. Models and APIs are shared online and version controlled [NASA 2018].

### **5.2.2. Enforce protection of endpoints per API docs**

The USS API published by NASA has a specification for the scopes of access to each endpoint. It was a priority to begin implementing that access control early and to specification for the USS implementers.

### **5.2.3. Encourage use of single-scoped access\_tokens**

The UTM research platform leverages OAuth 2.0 for access control (authorization). This is achieved by USSs requesting access\_tokens from a centralized token server and then using

those to access endpoints on other USSs. It is possible to request several scopes of access in a single token. Through a threat modeling activity internal to NASA, it was determined that single-scoped tokens were appropriate for UTM and multi-scoped tokens should be disallowed. As such, socializing this requirement during this sprint with an eye toward more difficult implementations in future sprints was begun.

**5.2.4. Exercise new credential naming based on DNS names**

In previous testing, NASA allowed USSs to be identified through a single identifier of that USSs choice using any “reasonable” format. There are benefits to formalizing and limiting the type of identifier a USS may use. This led NASA to defining a USS name as being a DNS name. This has some interesting security features related to the sharing of certificates and their mapping to USS identities within UTM as well as aiding in the prevention of token reuse by another party. In this sprint, the naming scheme was used, but names were not explicitly checked for adherence to the specification.

**5.2.5. Test initial concept for message signing for integrity and authentication**

For the messages exchanged between USSs within UTM, it is important that those messages be authenticated, non-repudiable, and have integrity guarantees. Through discussions within NASA, with partners, and with the FAA, it was determined that the most appropriate approach to this need within UTM was the signing of messages via JSON Web Signatures (JWS). To test this approach, partners were required to create JWS of their data payloads and exchange them with others within UTM.

**5.2.6. Prove concept of USS managed constraints**

In NASA’s future, operational concept for UTM, it will be important for certain qualified USSs to be able to manage constraints in the airspace on behalf of authorized entities (i.e. not on its own behalf). Even if USSs are not “managing” the constraint, it is likely that USSs will need to exchange data related to such constraints. This goal was designed to show that such data could be exchanged and acted upon by USSs.

**6. Test Organization**

To execute the collaborative simulation with USS implementers, the following definitions were developed:

**Table 2. Definitions used for simulation planning.**

|               |   |
|---------------|---|
| Definitions:  |   |
| Actor         | An entity participating in this simulation activity. In this case, it is primarily USS implementers.  |
| Role          | A set of steps for an actor to enact.   |
| Scene         | A defined collection of Roles along with notes for how those Roles are expected to interact.  |
| Configuration | Each Scene may be run several times. Each run will have different Roles assigned to different Actors. A Configuration is defined as a unique assignment of Roles to Actors. Each role-test pairing will specify the relevant geography/geographies to be used by the Actor. |

The “Actors” were the USSs defined in the table above in addition to various NASA actors: two for the NASA USSs, one for test coordination, and one for data collection.

The “Roles” were defined as follows:

**Table 3. Roles defined for use within simulation.**

| Roles:            |   |
|-------------------|---|
| RegularRay        | A nominal plan.   |
| DisruptedDory     | A nominal operation affected by a UAS Volume Reservation (UVR).   |
| LeadTimeLoach     | A nominal operation where USS/operator receives a UVR notice, but has enough time to complete its planned operation.  |
| ModifiedMullet    | An operation where USS/operator receives a UVR notice, decides to modify plan to continue mission, but adhering to the UVR.                                       |
| ExemptedEel       | An operation where USS/operator receives a UVR notice, USS recognizes that the operation is in the "permitted" list(s) so operation continues.                    |
| WildcardWahoo     | An operation where USS/operator receives a UVR notice. USS helps decide if operation should complete or return to base (RTB) based on the UVR and operation data. |
| RestrictionRemora | A USS that generates a UVR for other USSs.  |

The “Scenes” were defined as follows:

**Table 4. Scenes defined for use within simulation.**

| Scenes:                |   |
|------------------------|---|
| A Simple Plan          | Nominal operations for all Roles, no conflicts or dynamic information added during the Scene.                               |
| Army of Darkness       | Multiple operations interrupted by a UVR.   |
| The Gift               | Multiple operations interrupted by a UVR, some may be permitted to continue flying.   |
| The Quick and the Dead | Multiple operations interrupted by a UVR, some may be able to complete mission.   |
| Spider-Man             | Multiple operations affected by a UVR, USS must react according to UVR data. UVR details are not completely known a priori. |

Each Configuration is provided a geographical region within which the role will be executed. Each configuration for Sprint 1 will not be published here, but below is a sample from “The Quick and the Dead” scene:

**Table 5. Configuration table for various runs of “The Quick and the Dead” scene.**

| Role Label            |             | <i>Alpha</i>       | <i>Beta</i>    | <i>Gamma</i>   | <i>Delta</i>    | <i>Epsilon</i> | <i>Zeta</i>    |
|-----------------------|-------------|--------------------|----------------|----------------|-----------------|----------------|----------------|
|                       | <b>Role</b> | Restriction Remora | Disrupted Dory | LeadTime Loach | Modified Mullet | LeadTime Loach | Disrupted Dory |
|                       | <b>Geog</b> | r1_yellow          | d2_geo_brown   | d2_geo_green   | d2_geo_red      | d2_geo_silver  | d2_geo_blue    |
| <b>Configurations</b> | <b>A</b>    | NUSS               | ARMP           | UBER           | ANRA            | ROCK           | AOLN           |
|                       | <b>B</b>    | NUSS               | AROS           | AVIS           | SIMU            | ONES           | UBER           |
|                       | <b>C</b>    | NUSS               | AVIS           | SIMU           | AMZN            | ARMP           | ROCK           |
|                       | <b>D</b>    | NUSS               | ROCK           | AMZN           | AROS            | ANRA           | AOLN           |

The geographical information was shared with partners in the form of a GeoJSON file. The various geographies had labels for common referencing. A lightly populated area of North Dakota, one of the FAA test sites, away from major airports and national parks, was chosen for this activity. Outlines of the various geographies are visible in Figure 4.

The large, unfilled polygons represent potential UVR that may impact operations. Operations are planned to occur within the smaller, filled polygons. So, for example in Configuration A in the table above, NUSS will enact a UVR using the large, filled, yellow polygon from the geographies represented in the image, while ARMP would plan a mission to occur within the bounds of small, filled, brown geography. Forcing the mapping of roles to geographies provides some determinism in terms of the expected interactions and responses.

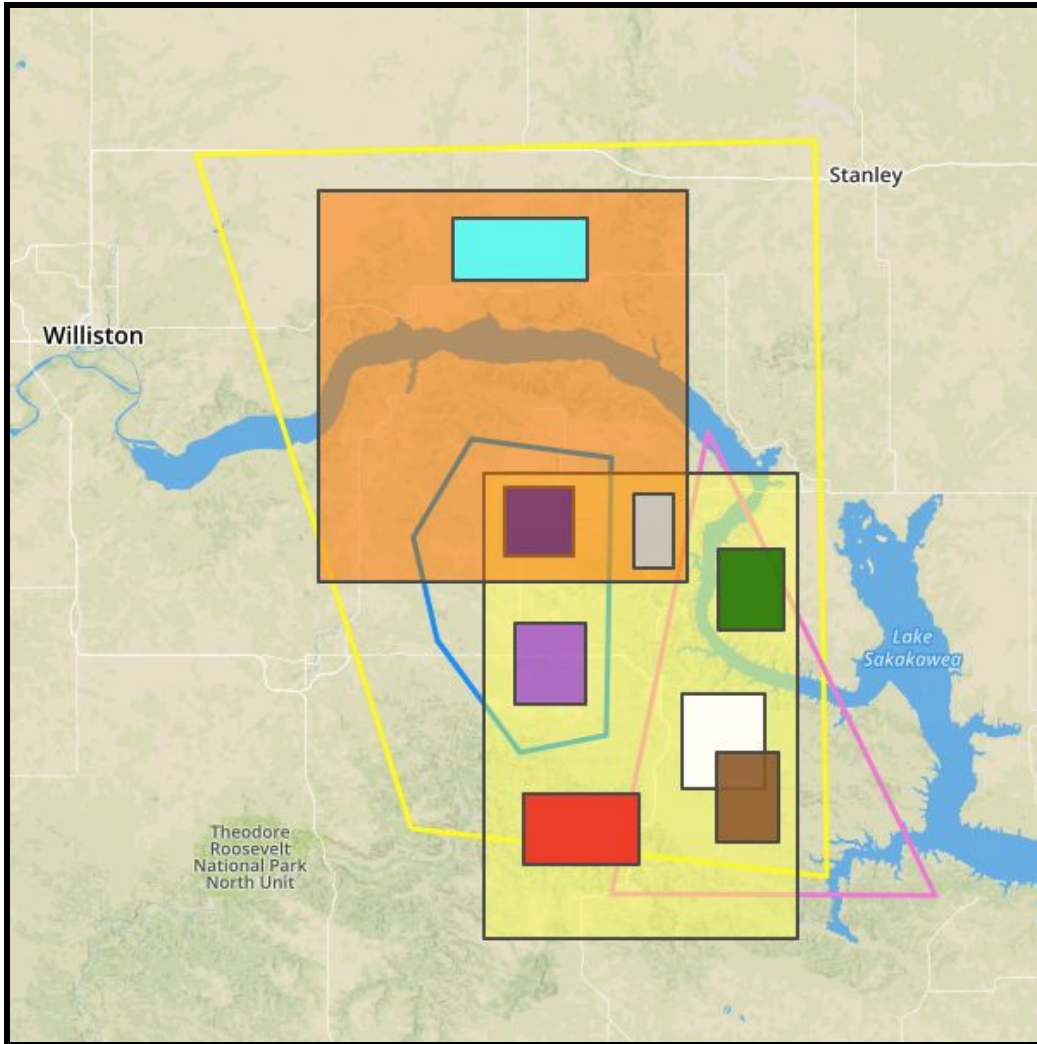


Figure 4. Operational geographies used in testing, each has a referenceable label (not shown).

## 7. Data Collection

During the event, NASA had a system named the “USS Data Collector” (UDC) which serves the same API as the USSs, but is not an actual USS. It is active solely to receive data from other USSs. All operations, position reports, and UTM messages are sent to UDC whenever they might be sent to any other system. UDC allows for real-time collection and viewing of operational data.

After the event, the USS implementers were asked to provide data related to the data exchanges. These data included expected and actual responses, latency, endpoint names, etc. From this self-reported data, NASA can begin to measure overall system latency and interoperability amongst USSs. A sample of the USSEExchange data model provided by one of the partners after the simulation is shown below with certain fields redacted. A partner may develop thousands of such USSEExchange instances depending on the volume of data exchanged during the simulation.

```

{
  "measurement_id": "6235b2b4-f58a-4ff0-ab43-f1cd4ba26c69",
  "event_id": "TCL4_USS_Sprint_1",
  "exchanged_data_pk": "c734ef47-6c3e-4e4d-9c82-faa2976bec9d",
  "exchanged_data_type": "UTM_MESSAGE",
  "source_uss": "<redacted>",
  "target_uss": "<redacted>",
  "reporting_uss_role": "TARGET_USS",
  "time_request_initiation": "2018-08-21T16:03:17.998Z",
  "time_request_completed": "2018-08-21T16:03:18.108Z",
  "endpoint":
  "https://<base_url_redacted>/utm_messages/c734ef47-6c3e-4e4d-9c82-faa2976be
  c9d",
  "expected_http_response": 204,
  "actual_http_response": 204,
  "jws": "<redacted for brevity>",
  "jws_public_key": "<redacted>",
  "comments": ""
}

```

Figure 4. Sample of data collected after event in JSON format..

## 8. Initial Results and Discussion

The USS implementers submitted files containing arrays of instances of the USSExchange model. Some files were malformed and could not be used for analysis. Most of the usable files needed some data re-formatting. Some of these fixes included fixing spelling of enumeration values, standardizing the naming scheme for USSs (needed the same name for each USS across submissions of data), and relaxing of the “required” fields for implementers that did not submit particular data elements. These fixes did not affect the results provided below, but may have precluded further analysis.

In future simulations, the partners will be submitting data to an automated system that will perform data validation which will provide immediate feedback on most of the common data submission errors. Thus, in future analysis, it is hoped more analysis and insight will be gleaned.

### 8.1. General Data

The following table summarizes some of the general information regarding the data collected.

Table 6. Summary of data collected after simulation from USSs.

|   |        |
|---|--------|
| USSs submitting valid USSExchange data                | 7      |
| Total USSExchange instances included in analysis      | 40,851 |
| Number of unique servers targeted with data exchanges | 14     |



## 8.2. Message Types

USS-USS communication involves the exchange of many types of data. Using the USSExchange model, the following message types and their frequency are described in Figure 5.

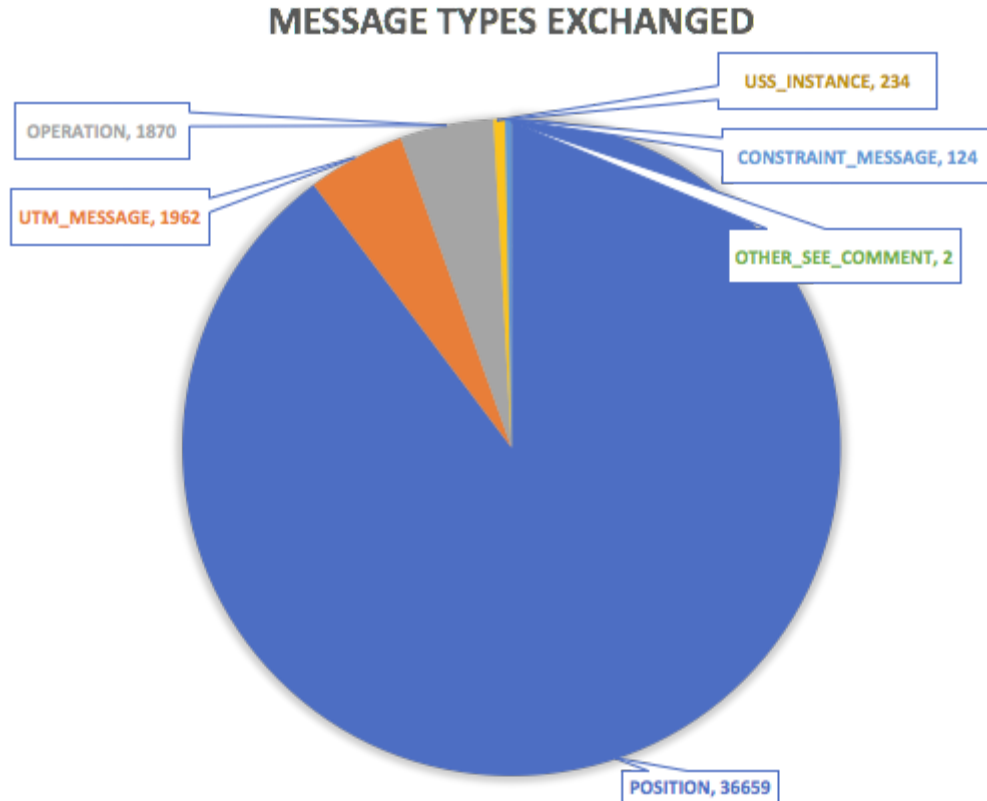


Figure 5. Breakdown of various message types exchanged between USSs during the simulation.

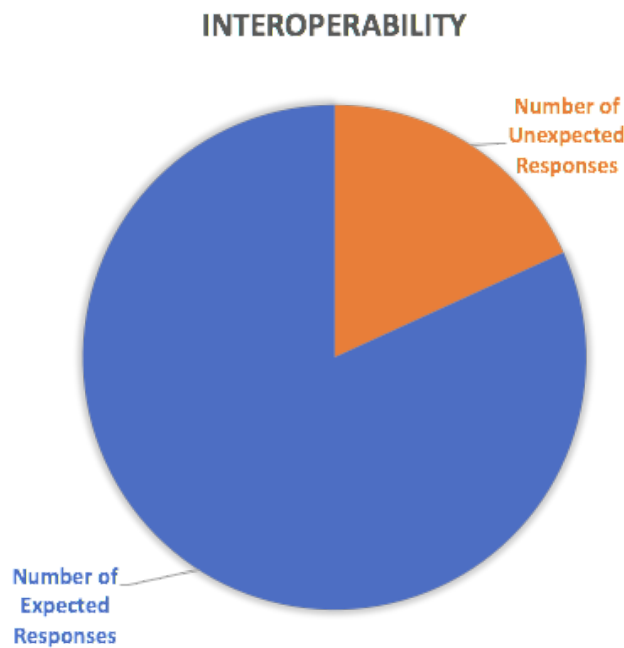
As expected due to the USS data exchange protocols requiring 1Hz updates of positions provided to other stakeholders upon request, the most frequent data exchanged are position messages. This points to a major opportunity for future optimization of message exchanges. An improvement in the approach to sharing position data would result in a major improvement in overall data exchanges.

## 8.3. Interoperability

There is not a readily evident, or accepted, means of measuring “interoperability.” As an initial approach, we count the number of exchanges that ended with the requesting server receiving the response that it was expecting. If every system with UTM receives what it was expecting from every data exchange, then we may conclude that the system has perfect interoperability. Since this is a RESTful architecture, wherein remote Hypertext Transfer Protocol (HTTP) calls are made to well-defined uniform resource locators (URLs) using specific data models, the USSs can log whether they receive the expected HTTP response or not. For example, when performing an HTTP GET request for data from another USS, it might be expected that the response returns an “HTTP 200” as opposed to an “HTTP 404.” This approach may not be

sufficiently informative in that when an unexpected response is received from another server, it does not mean that the systems are not interoperating appropriately. It might be better to classify the “expected\_http\_response” field within the USSEExchange model as “desired\_http\_response” since this is likely how the data element is interpreted by the USS implementers.

For true interoperability in terms of responses to requests, we may need to measure “non-error” responses or something similar in future data collections. For now, however, the metric of comparing the expected HTTP response code with the actual HTTP response code provides reasonable initial insight. The results are summarized in Figure 6.



**Figure 6. Breakdown of expected vs. unexpected responses during the simulation.**

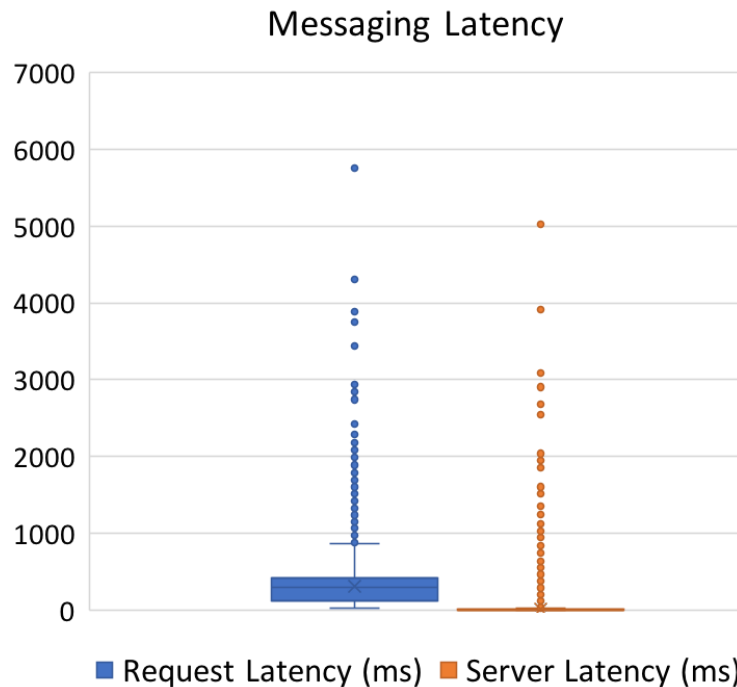
With this approach, over 80% interoperability was achieved. The vast majority (> 97%) of the unexpected status codes were HTTP 400. This typically implies that the receiving server detected problems with the data that were sent from the source server. There are examples from deeper inspections of these exchanges which show the true problem could be on either side of the exchange. This points to the need for more stringent checkout processes for USSs moving forward.

The other examples of unexpected HTTP status codes included HTTP 403 (a problem with authentication/authorization), HTTP 405 (payload too large), HTTP 413 (trying to access an endpoint with the wrong HTTP method), and various 50X responses indicating a problem with the target server.

#### **8.4. Latency**

The TCL4 Sprint 1 simulation was the first time that latency information between major components in UTM were collected. Using the USSEExchange model, latency from two perspectives were calculated and collected: first, the time it takes to complete a request from

the perspective of the requestor, and second, the time it takes for a target server to complete a request. These are indicated as “Request Latency” and “Server Latency,” respectively, in the box and whisker chart below.



**Figure 7. Latency measures for message exchanges between distributed USSs.**

Note that the Server Latency values are embedded in the Request Latency values since the Server Latency includes the time it takes for the remote server to process the request together with the network latency. Individual requests are not correlated in this study, so it is not possible to determine the network latency for any individual request. However, results provide insight into the fact that most requests are completed (with server and network latency) in under half of a second.

Four outlier values in the 10’s of seconds were removed from the Request Latency data set. Those requests would need further analysis, likely in the form of debriefs with the USS implementers, to determine the cause of their values. Those four requests were between the same source and target server pair, so the issue is likely unique to the connections between those servers.

## 9. Summary

In preparation for NASA’s TCL4 flight demonstrations, development is underway on USS implementations. These implementations are being tested in a collaborative manner to ensure the goals of a future flight test will be met while calculating interoperability and performance metrics. This initial collaborative simulation provides insight into the UTM system and design.

In this sprint, a baseline for the Application Programming Interfaces (APIs) and their associated data models was established for TCL4 testing. The concept of UAS Volume Reservations was tested, which provides confidence in the ability of the implementation to support this feature in

TCL4 flight tests. NASA gained confidence that the foundation for future sprints and TCL4 was solid.

## References

[Aweiss 2018] Aweiss, Arwa S., Brandon D. Owens, Joseph L. Rios, Jeffrey Homola, Christoph P. Mohlenbrink, "Unmanned Aircraft Systems (UAS) Traffic Management (UTM) National Campaign II," AIAA SciTech Forum, Kissimmee, FL, Jan 2018.

[FAA 2018] Federal Aviation Administration, "Unmanned Aircraft System (UAS) Traffic Management (UTM), Concept of Operations v1.0," Washington, DC, May 2018.

[FAA 2018b] Federal Aviation Administration, "FAA Aerospace Forecasts: Fiscal Years 2018-2038," [https://www.faa.gov/data\\_research/aviation/aerospace\\_forecasts/media/FAA\\_Aerospace\\_Forecasts\\_FY\\_2018-2038.pdf](https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/FAA_Aerospace_Forecasts_FY_2018-2038.pdf). Washington, DC, retrieved Oct 2018.

[Homola 2017] Homola, Jeffrey, Christoph Mohlenbrink, Quang Dao, Lauren Claudatos, Lynn Martin, Joey Mercer, "UAS Technical Capability Level 2 Unmanned Aircraft System Traffic Management (UTM) Flight Demonstration: Description and Analysis," IEEE-DASC, St. Petersburg, FL, Sep 2017.

[Johnson 2017] Johnson, Marcus, Jaewoo Jung, Joseph Rios, Joey Mercer, Jeffrey Homola, Thomas Prevot, Daniel Mulfinger, Parimal Kopardekar, "Flight Test Evaluation of an Unmanned Aircraft System Traffic Management (UTM) Concept for Multiple Beyond-Visual-Line-of-Sight Operations," 12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, Jun 2017.

[Johnson 2017b] Johnson, M., et al., "Flight Test Evaluation of a Traffic Management Concept for Unmanned Aircraft Systems in a Rural Environment," Moffett Field, CA, to be published.

[Kopardekar 2016] Kopardekar, Parimal, Joseph Rios, Thomas Prevot, Marcus Johnson, Jaewoo Jung, John E. Robinson III, "Unmanned Aircraft System Traffic Management (UTM) Concept of Operations," NASA Ames Research Center. 16th AIAA Aviation Technology, Integration, and Operations Conference, Washington, DC, Jun 2016.

[LAANC 2018] Federal Aviation Administration, "UAS Data Exchange," [https://www.faa.gov/uas/programs\\_partnerships/uas\\_data\\_exchange/](https://www.faa.gov/uas/programs_partnerships/uas_data_exchange/), Washington, DC, retrieved Oct 2018.

[NASA 2018] National Aeronautics and Space Administration, "NASA UTM APIs," <https://github.com/nasa/utm-apis>, retrieved Oct 2018.

[NASA 2018b] National Aeronautics and Space Administration, "NASA UTM Documents," <https://utm.arc.nasa.gov/documents.shtml>, retrieved Oct 2018.

[Rios 2016] Rios, Joseph, Daniel Mulfinger, Jeff Homola, Priya Venkatesan, "NASA UAS Traffic Management National Campaign - Operations Across Six UAS Test Sites," IEEE-DASC 2016, Sacramento, CA, Sep 2016.

[Rios 2017] Rios, Joseph, Daniel G. Mulfinger, Irene S. Smith, Priya Venkatesan, David R. Smith, Vijayakumar Baskaran, Leo Wang, "NASA Technical Memorandum-UTM Data Working Group Demonstration 1 Final Report," Moffett Field, CA, Apr 2017.