CONTRACTED SERV ICES AGREEMENT
RFC Errata Merge Tool

THIS CONTRACTED SERVICES AGREEMENT (this “**Agreement**”) is made effective as of the 06th day of May 2019, (the “**Effective Date**”) by and between **IETF Administration LLC** (“**IETF**”) and **James Schaad DBA Soaring Hawk Consulting** (the “**Developer**“).

1. Services. Developer shall provide to the Company the software development and other services described in any Statement of Work(s) (“**SOW(s)**”) executed by the parties or as otherwise mutually agreed by the parties (collectively, the “**Services**”). Developer will perform the Services in a professional and workmanlike manner and in accordance with the prevailing industry standard for the performance of comparable work, will use best current practice for software development and Developer will adhere to the instructions and practices described in “Instructions for IETF Software Development Contractors” located at https://trac.tools.ietf.org/tools/ietfdb/wiki/ContractorInstructions?version=26

2. Independent Developer. Developer’s relationship with IETF will be that of an independent contractor and nothing in this Agreement is intended to, or should be construed to, create a partnership, agency, joint venture, or employment relationship. Consistent with broad direction set by IETF, Developer will determine what actions are required to perform the Services and to achieve the relevant objectives. Developer will provide its own equipment (e.g., laptop and phone, and related services) and set its own hours. Developer may engage on other projects during the term of this Agreement, provided such work does not present a conflict of interest, result in disclosure of Confidential Information or otherwise interfere with Developer’s ability to complete the Services under this Agreement in a satisfactory manner. Developer shall not be provided any training by IETF and is expected to have all the expertise necessary to carry out the Services. Developer shall not engage the services of third party contractors, subcontractors or consultants (“Subcontractors”) in the performance of its obligations under this Agreement without the prior written consent of the IETF, specifying both the specific Subcontractor and the scope of work which it is permitted to undertake. To the extent IETF consents, Developer shall be fully responsible for each such Subcontractor’s compliance with the terms of this Agreement, and Developer shall be liable, without limitation, for all actions and omissions of such Subcontractors in their performance or failure to perform as required hereunder. Prices or fees as specified in SOWs are exclusive of taxes. Each party assumes responsibility for all taxes, levies, or charges by any governmental entity that are applicable to it under local laws or binding international agreements associated with its status under this Agreement.

3. Term & Termination. The initial term of this Agreement will begin on 13 May 2019 and end on 24 June 2019, provided that any SOWs in effect at termination will remain in full force and effect until the SOW term expires, unless expressly terminated pursuant to this Agreement. The parties may mutually agree in writing to renew the engagement for successive renewal periods after expiration of the term of this Agreement, and the terms of this Agreement will continue to apply. Either party may terminate this Agreement and SOW at any time and for any reason with 30 days’ prior written notice to the other party. Either party may terminate this Agreement immediately upon written notice if the other party materially breaches an obligation of this Agreement and does not cure such breach within 30 days of receiving notice of such breach. Upon termination for any reason, Developer agrees to cooperate in good faith with IETF to wind down and transition any work in progress.

4. Compensation and expenses.

IETF will compensate Developer for the Services at the rates set forth in the applicable SOW and in accordance with the schedule set forth therein. If the fees based on an hourly rate, Developer will invoice IETF on a monthly basis by sending a timelog and invoice to exec-director@ietf.org. IETF will pay undisputed amounts of any invoice within 30 days of receipt via wire transfer.

IETF will reimburse Developer for direct expenses described in the SOW or incurred specifically in connection with the Services, subject to IETF's prior written approval for any such expenses. Other than pre-approved direct expenses, Developer will be solely responsible for all other costs and expenses incurred in performance of this Agreement. IETF will reimburse Developer for appropriate approved expenses within 30 days after receiving an invoice from Developer, together with written documentation and receipts itemizing all such expenses. Developer must invoice IETF for expenses no more than one month from when they are incurred.

5. Confidentiality. In the course of Developer's engagement with IETF, parties may be exposed or have access to information, materials or documents that are confidential to the other party, or its member or community ("**CI**"). Parties agree to maintain CI as confidential, and to not disclose or cause to be disclosed any CI, nor use CI for any purpose except as necessary to perform the Services under this Agreement. Parties will return or destroy any CI of the other party in its possession upon termination of this Agreement.

6. Intellectual Property.

A. **Assignment.** The IETF Trust is a Virginia common law non-profit Trust whose beneficiary is the IETF. The IETF Trust shall own all right, title and interest in and to all information, materials and other proceeds that Developer creates in the course of, or that otherwise result from, the Services or Developer's engagement with IETF ("**Work Product**"). All Work Product shall be deemed "works made for hire" to the extent permissible under the copyright law, and to the extent any Work Product may not be so deemed, Developer hereby assigns all right, title and interest in and to all intellectual property and other proprietary rights in such Work Product to the IETF Trust. Developer retains ownership in all other works Developer created prior to this Agreement or creates in the future outside of the scope of the Services and Developer's engagement with IETF. Upon termination of this Agreement, Developer will provide to IETF any working drafts or other interim phases of deliverables Work Product as they exist upon termination.

B. **Open Source Software**. The IETF Trust intends to release some or all of the Work Product to the public under the Simplified BSD Software License or another open source software license, and Developer hereby represents and warrants that Developer will not use, integrate, or develop software as part of the Work Product performed by it hereunder that is incompatible with the Simplified BSD Software License or another open source software license identified to it by IETF (via electronic mail or in writing).

C. **Required Rights**. Prior to commencing any work, or as promptly as possible once identified if so identified after work has commenced, Developer shall describe in writing:

- Any intellectual property rights owned or licensed by Developer which may cover all or part of the work, including a list and description of all U.S. and foreign patents and patent applications;
- To the extent known by Developer, any intellectual property owned or licensed by third parties which is required to utilize all or part of the work in the manner contemplated by the Agreement; and
- To the extent known by Developer, any claims or disputes relating to the intellectual property embodied, or claimed to be embodied, in all or part of the work.

Intellectual property and claims described in the bullets above are termed "**Required Rights**". In addition to the descriptions required above, Developer shall provide to IETF a description of the cost and other terms of any license required to use and operate under any Required Rights in the manner contemplated by this Agreement.

Developer shall not be authorized to commence any work as to which any Required Rights exist unless and until IETF has affirmed in writing that it understands the nature of such Required Rights and the Parties have mutually agreed upon a license arrangement (including allocation of its costs) that will enable the full use of any Required Rights in the manner contemplated hereby.

If Developer fails to notify IETF of any Required Rights owned or licensed by Developer in the manner required by this section, then Developer shall be deemed to have granted the IETF Trust a perpetual, irrevocable, royalty-free, paid-up, worldwide, non-exclusive, freely sub-licensable right and license to exploit such owned Required Rights (and to the extent permitted under any such license of any such Required Rights, such licensed Required Rights) in any manner in connection with the Work and any modifications or derivatives thereof.

D. **Data**. All data used or collected by the Work Product shall be the property of the IETF Trust.

7. Warranties; indemnification. Developer warrants that (i) it will perform all Services in a professional and workmanlike manner, in accordance with highest ethical standards, (ii) it has the right and authority to enter into this Agreement, (iii) it has the knowledge and skills to provide the Services, (iv) it will comply with all applicable laws, statutes, or regulations in the performance of the Services, and (v) to the best of Developer's knowledge, the Services and Work Product will not violate or in any way infringe upon the rights of third parties. Developer agrees to indemnify and hold IETF and its member, directors and officers harmless from any claim, loss or expense arising in connection with Developer's (or any Subcontractor's) breach of this Agreement, or in connection with its negligence or willful misconduct. Developer warrants that the work will perform in accordance with any applicable documentation, specifications and written descriptions provided by IETF to Developer, and in a reliable and secure manner.

Developer agrees promptly to fix any errors, bugs or deficiencies in the work that are identified within thirty (30) days from the date of delivery of the work to IETF for no additional charge and to deliver to IETF, install in the work and demonstrate to IETF's reasonable satisfaction such fixes. The development and delivery of such fixes shall be performed by Developer in accordance with the provisions of this Agreement and all software developed and/or delivered shall conform to the requirements hereof.

IETF shall test each deliverable within thirty (30) days of its receipt from Developer and shall report to Developer, in writing, whether or not the deliverable complies with agreed specification. If IETF fails to report to Developer, whether or not deliverable complies with agreed specification, within thirty (30) days of its receipt from Developer, such deliverable shall be considered finally accepted by IETF. If any deliverable does not comply with agreed specification and IETF has reported it to Developer within thirty (30) working days period, IETF shall provide Developer with details of the failure and Developer shall use commercially reasonable efforts to modify the deliverable so that it complies with agreed specification. The parties shall repeat the above process until the deliverable complies with agreed specification.

8. Insurance. Developer shall maintain any required and customary insurance in amounts and types reasonable for the type of business conducted by Developer and at IETF's request, shall make IETF an additional insured under any such applicable policy. Developer shall provide evidence of insurance and endorsements at IETF's request, and shall cooperate with IETF in good faith to adjust such insurance coverage as appropriate for the Services.

9. Miscellaneous. This Agreement will be governed by Delaware law. Developer may not assign or delegate any of its rights or obligations set forth in this Agreement without IETF's prior written consent. This Agreement is binding and shall inure solely to the benefit of the parties hereto (and to the benefit of the IETF Trust with respect to intellectual property), and their respective successors and permitted assigns. The IETF Trust is the only intended third party beneficiary to this Agreement. Other than the IETF Trust, nothing in this Agreement shall be enforceable by a third party (being any person or entity other than the Parties hereto). This Agreement, together with any SOWs, constitutes the sole and entire agreement of the parties with respect to the subject matter contained herein, and supersedes all prior and contemporaneous understandings, agreements, representations and warranties, both written and oral, with respect to such subject matter. This Agreement may be executed in counterparts, each of which shall be deemed an original, but all of which together shall be deemed to be one and the same agreement. All notices, requests, directions, approvals or other communications to be provided hereunder will be in writing and will be deemed to have been sufficiently given (a) upon receipt if delivered in person; or (b) on the date transmitted if by email or facsimile. All notices will be sent to the applicable party at the address(es) set forth below (or as otherwise instructed in writing by such party):

If to IETF:

IETF Administration LLC
Attn: Executive Director
exec-director@ietf.org
1000 N. West St, Ste. 1200
Wilmington, DE 19801

If to Developer

James Schaad
Soaring Hawk Consulting
17050 SW Rocky Ramble Ln
Sherwood, OR 97140-9215

10. Publicity. Developer must seek IETF's written preapproval prior to any use of the name and logo of IETF as well as short project description on all its marketing collaterals (online, offline, website, papers, brochures, press releases) and any disclosure of the existence of relationship with IETF for reference and marketing purposes.

11. Non-Solicitation of Employees. Both parties agree that during the term of this Agreement, any extension thereof and one year after its termination, neither party will offer employment or hire any employee, agent or subcontractor of the other party performing work under this Agreement.

Read and agreed to by:

IETF ADMINISTRATION LLC

[redacted]

By: Portia Wenze-Danley

Title: Executive Director (Interim)

Email: exec-director@ietf.org

James Schaad DBA Soaring Hawk Consulting

[redacted]

By: Jim Schaad

Title:

Email: ietf@augustcellars.com

Statement of Work

This Statement of Work (“SOW”) is entered into effective as of its signature by both parties (the “Effective Date”) by and between Soaring Hawk Consulting (“**Developer**”) and IETF Administration LLC (“**IETF**”). This SOW is incorporated into, and forms a part of, the Contracted Services Agreement, dated 06 May, 2019 by and between the parties (the “**Agreement**”). Any term not defined herein shall have the meaning ascribed to it in the Agreement. This SOW must be signed by both parties to be effective.

All of the terms, covenants and conditions set forth in the Agreement are incorporated herein by reference as if the same had been set forth herein in full.

Developer agrees to provide the following Services to IETF pursuant to this SOW:

Project Approach & Plan

General

Command lines will be provided with all options detailed for review prior to implementing any specific option. Where possible all options will be provided as early as possible for review.

The tool is written in Python version 3 and will include a PIP installer object.

There will be a single deliverable executable file. The difference between the requested support tool and the merge tool is identified by using different command line arguments. The tool will keep some state information in a file in the local directory to remember some command line options between invocations. This allows for information such as the source and destination of information and which errata are to be applied to be remembered between invocations.

ERRATA MERGE SUPPORT PHASE

This phase will download the errata json file from the server if both needed and requested. The download will be suppressed both if there is no need or if a command line argument exists not to check. It will be an error if the local file does not exist and the download is suppressed. The time of last modification on the server is compared to the data of last modification on the local machine of the file to determine if a download is needed.

After the file has been downloaded it will be parsed up and some normalization is performed to deal with either errors or additional fields that make later processing easier. The set of all errata will then be grouped by the RFC number and filtered for which groups of errata are to be applied for later application.

The appropriate set of errata will then be applied to either all of the RFCs or to the requested RFCs.

ERRATA MERGE TOOL

The merge tool will start by looking for the most recent update/submit time in the set of errata to be applied to the text file. That will be compared against an extant html file if one exists to determine if the errata need to be applied. A command line option will allow this test to be skipped. If required the RFC text file is then downloaded. The merge tool assumes an optimization that the text versions of the RFCs can be kept around and therefore does not clean them up after modifications are applied.

All errata will be applied to a text file at the same time, that the program will not just apply new errata but will apply all errata from the original file. This is required as the current rule of applying errata by number means that a later approval may need to be applied before an existing one. This is needed so that the same result will be created in all cases.

The document is sectioned using one of two algorithms, this is currently the only place were specific information dealing with specific RFCs is encoded. Attempting to combine the current and historical matching algorithm leads to failures as the heuristics needed for old RFCs causes false positive matches to occur in modern RFCs.

Code for removing line breaks was lifted from the id2xml utility and does a good job in general, some work might be put into improving some cases, but removal of page breaks can never be completely correct due to the existence of widows and orphans which mean that one cannot just remove based on the number of lines in a document.

Errata for which the section does not match to a section identifier based on the sectioning of the document are will be matched from the top to the bottom of the document assigning them to the first section that a match occurs. This means that the case called out in the RFP will successfully place Errata 571, if an incorrect section number is given and can be found in the document the errata would be placed as a section erratum rather than an inline erratum.

For simplicity, errata are formatted by the use of a number of templates targeted each of the different locations where an erratum is placed in the document. This allows for simple modifications of how an erratum is presented in the final document. For example, an inline erratum could be either to include or not include the new text as that text would now be part of the document. This also means the fields such as the verifier name can be added or removed by simply updating the templates.

ERRATA FILE CHANGES REQUESTED

There are a number of changes in the errata file that are quested:

- Sections are represented both as numbers and strings, making them all strings is requested
- RFC numbers are setup to have leading zeros in the numbers. For example, RFC 822 is represented as RFC0822 in the file. Removing these leading zeros makes things simpler as one would not need to find and remove this information. An assumption is made that the output files do not have leading zeros. One of the worries is what happens when the 10K number is reached.
- End of line normalization in fields should occur. Doing a pass of normalization for end of line characters in the database may affect the display of errata on the RFC Editor website, currently it is needed to do this as the manipulation code assumes that only the newline character is present. The errata file is inconsonant about the presence or absence of carriage returns.

- Change the format of the submit date field so that it is the same format as the update date field. Not having the time field means that a large gap exists where a race condition exists.

DELIVERABLES

- The python program,
- Four CSS files,
- One JavaScript file.
- Four template files

Schedule

- Key project dates are outlined below. Dates are best-guess estimates and are subject to change until a contract is executed.
- The schedule presented below assumes that the different tools will be developed in sequence within the phases. The expectation is that they will be developed in parallel and only the phase dates are firm.
- The initial version setup on a server will be under on the winery website.

Description	**Start Date**	**End Date**	**Duration**
<Project Start>	13 May, 2019		
MILESTONE: Initial version up on a server		20 May, 2019	
Errata Merge Tool Delivery	13 May, 2019	24 June, 2019	6 weeks

Test Plan

General

The primary source control repository that Developer will be using for development will be GitHub, as required by the contract when specific features are finished then a check-in to the IETF subversion repository will be done, this allows for the use of Circle-CI (Ubuntu and OSX) and AppVeyor (Windows x64) to perform testing on each modification to the project as it occurs. All of the scripts to run the tests as well as the test cases themselves will be part of the repository that is provided to the IETF.

As part of the test scripts, checks of PEP8 and PYFLAKES will be run on all source code to ensure that it meets these standards.

ERRATA MERGE SUPPORT TOOL

Testing for this tool will consist of two parts:

* A live run against the RFC Editor website to download the JSON file and any required RFCs. In order to reduce the load, some of the RFC text files may be checked into the test system. This will check that no new issues arise from new errata as they appear.

* A set of unit tests will be run against the different components to check that time checks produce the minimal needed updates mostly using prebuilt fixed information.

The test harness is expected to be a python script for maximum cross platform compatibility.

ERRATA MERGE TOOL

Testing for this tool will consist of a number of unit tests that can be run against the tool for the purpose of validating that the output conforms to what is expected. Tests will be developed based on identified corner cases and will include parsing of documents as well as applying the errata to the documents.

Cost & Payment Schedule

[redacted]

[redacted]

[redacted]

Warranty & Late Delivery Consequence

The software will be delivered as-is. No warranty is included.

Technical Support & Maintenance

Developer will provide support for bug fixes at no cost as long as Developer is participating with the IETF. If new features beyond those described in this SOW are desired by IETF, additional payment may be required but would be negotiated and mutually agreed upon at that time.

Documentation

The PIP installation on PYPI will contain the documentation. Documentation is only expected to consist of command line options.

Subcontractor Information

Developer currently does not envision to enter into any formal subcontractor relationships, and shall comply with the terms of the Agreement prior to subcontracting any of its responsibilities under this SOW.

Assumptions

None.

Intellectual Property

The Agreement shall govern the intellectual property rights associated with the deliverables.

Miscellaneous

N/A

[redacted]

By: Portia Wenze-Danley
Title: Executive Director (Interim)
Email: exec-director@ietf.org

[redacted]

By: Jim Schaad
Title:
Email: ietf@augustcellars.com