



REQUEST FOR PROPOSAL (RFP):

Allotrope Data Format (ADF) Python Library – PyADF 2020

Allotrope Foundation
1500 K Street, NW
Washington, DC 20005

July, 2020

Contents

1.0	Summary and Background	3
2.0	Proposal Guidelines	3
2.1	Additional Conditions.....	3
3.0	Purpose and Description.....	4
3.1	Background	4
3.2	Purpose	4
3.3	Project Description Requirements.....	5
3.4	Infrastructure Requirements	5
3.5	Detailed Requirements	5
3.6	ADF Python API Wrappers Development Phases.....	8
4.0	Project Scope	11
4.1	Development Phases and Cycles.....	11
4.2	Design, Test, Document and Release Specifications.....	11
4.3	Timeline.....	12
5.0	Budget	13
5.1	Additional Budget Considerations	13
6.0	Bidder Qualifications.....	13
6.1	Specific Technical Questionnaire to the Bidder	14
7.0	Access to Applicable Material for Response.....	14
8.0	Proposal Evaluation Criteria.....	14
8.1	Peer Questions and Answers	15
8.2	Submission	15
9.0	References	15

1.0 Summary and Background

Allotrope Foundation is currently accepting proposals for the Development of a Pythonic interface as well as the related Python packaging, the supported documentation and artifacts to extend the existing APIs of the Allotrope Data Format (ADF) software library originally written in Java. Named as ADF Python Library - PyADF, the package is developed within the [Allotrope Foundation's framework](#) of products.

The purpose of this Request for Proposal (RFP) is to solicit proposals from various candidate organizations, conduct a fair and extensive evaluation based on criteria listed herein, and select the candidate who best presents the capability to meet Allotrope Foundation's strategic development and support needs.

Allotrope Foundation is a not-for-profit organization founded in 2013 and focused on developing new technological solutions to revolutionize the way insights are acquired and shared by standardizing and linking scientific data. Our international community consists of scientific instrument and solution vendors as well as biopharmaceutical and technology companies of various sizes who are seeking quicker and deeper insight into their data.

The Allotrope Foundation headquarters is in Washington, D.C. with a remote Product Team of full-time employees and Allotrope Foundation Members working from their respective company sites.

2.0 Proposal Guidelines

This Request for Proposal represents the requirements for an open and competitive process. Please submit your response electronically. Submission details can be found in the Section 4 of this document. Responses received after the requested due date may not benefit from full consideration and may be excluded from the selection process. All proposals must be signed by either an official agent or representative of the company submitting the proposal.

If the organization submitting a proposal must outsource or contract any work to meet the requirements contained herein, this must be clearly stated in the proposal. Additionally, all costs included in proposals must be all-inclusive to include any outsourced or contracted work. Any proposals which call for outsourcing or contracting work must include a name and description of the organizations being contracted as well as any limitations to the use of third-party or other licensing within the software, including cost or duration of use of such licenses.

All costs must be itemized to include an explanation of all fees and costs.

Contract terms and conditions will be negotiated upon selection of the winning bidder for this RFP. All contractual terms and conditions will be subject to review by Allotrope Foundation Secretariat and will include scope, budget, schedule where applicable, and other necessary items pertaining to this RFP.

2.1 Additional Conditions

1. This RFP is not an offer or a contract
2. Proposals submitted in response to this RFP become property of Allotrope Foundation (AF)
3. Respondents will not be compensated or reimbursed for any costs incurred as part of the RFP process
4. Responses to RFPs should contain only high-level discussions of product development efforts and should not contain trade secrets or confidential information. AF does not make any confidentiality

commitments with respect to RFP submissions, but agrees not to publicly distribute the RFP responses outside the consortium or share RFP responses with other respondents or members of the APN.

5. AF is not obligated to contract for any of the products and services described in this RFP
6. AF reserves the right to:
 - a. Accept or reject any or all proposals
 - b. Waive any anomalies in proposals
 - c. Negotiate with any or all respondents
 - d. Modify or cancel this RFP at any time

3.0 Purpose and Description

3.1 Background

The Allotrope Data Format (ADF) is a vendor, platform, and method-agnostic format adaptable to store scientific data, contextual metadata, and any ancillary files. Advanced semantic technology is used as the foundation for the contextual metadata layer.

The current ADF software library is written in Java which is the language of choice for many enterprise applications. An additional transpiled version is written in C#. In recent years Java and C# are being overshadowed within the data science community by other languages with a growing momentum such as Python.

Python stands out as a concise, easy to use and to read language with an extensive set of tools for scientific computing. Allotrope Foundation is looking to implement a Pythonic interface to the ADF library originally written in Java that provides not only technical functionality but provides the “look and feel” expected of a Python software library.

3.2 Purpose

A Pythonic interface to the ADF Java library software stack has the following goals and objectives:

1. **Adoption:** Addressing different user groups and applications:
 - a. Scientific computing community (Python)
 - b. Enterprise application and instrumentation integration (Java, C#)
2. **Domain Access:** Access to the ADF files for early experimentation and assessment by analytics and data science applications
3. **Positioning & Growth:** Better positioned to integrate into existing and future data environments
4. **Enablement:** Simplify the path for Python, language of choice of the data science community, to align with their existing development tools and applications
5. **Compatibility:** API compatibility with the existing ADF Java library implementation

Both Java and Python are widely used programming languages and have been deployed in a variety of applications and systems. Developers have wanted to get the two languages to integrate cleanly and developed open source tools to enable it. A “Bridging” solution that utilizes those tools takes into consideration the following objectives in mind:

1. Quick proof of concept

2. Shorter time for project completion
3. Enabling the AF community to start with early experimentation and assessment of an ADF Python library and file format for analytics and data science applications.
4. Evaluation of Python APIs to address the need for "Ease of Use" of the ADF file
5. Possible style alignment with the HDF5 Python library – h5py where applicable and to elevate "Ease of Use" of the ADF file
6. Maintaining the Java code as the baseline.
7. Use existing off-the-shelf open source tools
8. Profiled as a software integration project and less as a software development project
9. Minimizing the initial project resources, cost, and risk
10. Evaluation and development of requirements for the ADF library APIs.

3.3 Project Description Requirements

The goal of this project is to author a set of Python APIs which can enable a Python software developer to create, read, write, update and delete ADF files content based on the ADF Library written in Java. The Python APIs are written in a Pythonic style and the usage requires no prior Java APIs skills, expertise, nor visibility to them. These APIs should provide access using native Python language classes and objects for a developer to interact with, and mechanisms for writing these objects to ADF files as well as populating the objects from ADF files. The Python APIs are expected to leverage the existing Allotrope ADF APIs to perform ADF-related operations in order to ease uptake by Python developers with a focus on scientific computing and the related Python libraries.

3.4 Infrastructure Requirements

#	Requirement	Details
3.4.1	JPyype	JPyype https://jpyype.readthedocs.io/en/latest/ is the selected Python module for this project to provide full access to the Java code within Python. It allows Python to make use of Java specific libraries, explore and visualize Java structures, develop, and test Java libraries, make use of scientific computing, and much more. It is expected from the selected bidder to get familiarized with the JPyype open source package to the level of bug fixing. The JPyype package is written in Python with a backend written in C++. Currently the JPyype community is active and open for accepting new features if required.
3.4.2	h5py	The h5py package http://api.h5py.org/ is a Pythonic interface to the HDF5 binary data format. h5py uses straightforward NumPy and Python metaphors, like dictionary and NumPy array syntax. It is expected from the selected bidder to get familiarized with the h5py open source package and to check for style alignments between h5py and the PyADF APIs

3.5 Detailed Requirements

#	Requirement	Details
3.5.1	Language	The API will be compatible with the most recent releases of Python versions 3.7 and 3.8
3.5.2	Bridging solution	
3.5.2.1		JPyPe will be used as a Python-Java “bridging” solution https://jpype.readthedocs.io/en/latest/
3.5.2.2		Python Wrapper using JPyPe tooling
3.5.2.3	Pythonic style and interface	
3.5.2.4		Customization and extension of Java classes to provide Python native syntax (e.g. property access instead of getters and setters, utilizing Python’s object composition and inheritance paradigms, etc.)
3.5.2.5		Casting of scalar and complex entities from Java types into native Python classes
3.5.2.6		Proxy classes in Python to wrap Java objects
3.5.2.7		Handling Java buffers as native Python IOBuffers
3.5.2.8		Use of Numpy arrays and other primitives to represent data
3.5.2.9		Support for generators, iterators, list comprehensions instead of for /while loops
3.5.2.10		Autocompletion in IDEs
3.5.2.11		Use of Type Hinting as specified by PEP484
3.5.2.12		Full documentation of Python API using reStructuredText and autogeneration of API documentation using Sphinx
3.5.3	Compatibility	
3.5.3.1		ADF Java library. Latest version 1.5.1 Newer minor version (with no Java API breaking change) shall be supported. Client Connect link: https://highq.in/1vjlaqsng
3.5.3.2		The Java API must function under the minimum OpenJDK versions: <ul style="list-style-type: none"> - OpenJDK 8 - OpenJDK 11
3.5.3.3		No modifications may be required of the Java source code unless approved in writing by AF
3.5.4	Packaging	
3.5.4.1		Standard Python packaging according to PyPa Python Packaging Guide https://www.pypa.io/en/latest/ <ul style="list-style-type: none"> - Installation: PIP
3.5.4.2		Distribution: Private and secured <ul style="list-style-type: none"> - Preferred option: Jfrog secure, private PyPI repository https://jfrog.com/integration/python-repository/ - AWS option: Setup and install a private and secured package distribution server on AWS
3.5.4.3		Conda Package, dependency, environment management and install https://docs.conda.io/en/latest/ in addition to a pip-installable tarball

3.5.4.4		Minimize use of any dependencies
3.5.5	SDLC and Tools	
3.5.5.1		Coding conventions and best practices: PEP 8 https://www.python.org/dev/peps/pep-0008/ using the Black formatter (https://github.com/psf/black)
3.5.5.2		docstring conventions: Google Python style guide: https://github.com/google/styleguide/blob/gh-pages/pyguide.md#38-comments-and-docstrings
3.5.5.3		Auto-generate API documentation tool: Sphinx https://www.sphinx-doc.org/en/master/
3.5.5.4		Source Code Management: Gitlab https://gitlab.com/allotrope-internal/adf-library
3.5.5.5		Continuous Integration: Gitlab https://gitlab.com/allotrope-internal/adf-library
3.5.5.6		Artifacts repository: jFrog https://allotrope.jfrog.io/
3.5.5.7		Release copy on Allotrope's Client Connect
3.5.6	Testing	
3.5.6.1		Static code analysis: Pylint https://pypi.org/project/pylint/
3.5.6.2		Unit tests code for API testing using unittest and pytest (https://docs.pytest.org/)
3.5.6.3		Automated functional tests
3.5.6.4		Performance characteristics tests comparing read/write operation scenarios between: <ul style="list-style-type: none"> - ADF Java library - PyADF
3.5.6.5		Test automation scripts
3.5.6.6		IDE test - PyADF functionality in: <ul style="list-style-type: none"> - Jupyter - Google Colab
3.5.7	Documentation Generation	
3.5.8		API autogenerated documentation and traditional documentation using Sphinx: https://www.sphinx-doc.org/en/master/index.html rendered as a Read-the-docs style static HTML site
3.5.9	Content Documentation	
3.5.9.1		Userguide <ul style="list-style-type: none"> - Including a tutorial under Jupyter notebook - Including ADF Primer for Python similar to the ADF Java Primer for Java https://gitlab.com/allotrope-open-source/adf-primer-java
3.5.9.2		Quickstart guide
3.5.9.3		Installation guide

3.5.9.4		Developer guide
3.5.9.5		Product release: <ul style="list-style-type: none"> - Release Note - Required information (including licensing) - Release package guide - Release build protocol - Release shipping document
3.5.10	Licensing	Dependencies that require a copyleft style license must not be used. Dependencies that require any licensing fees or royalties must not be used.

3.6 ADF Python API Wrappers Development Phases

The ADF Pythonic API wrappers development will be developed and implemented in three phases. The table below groups the APIs by categories and the functionality that is required in each phase. The related ADF Java library packages to the APIs category is detailed in a separate table below. As stated in the table, the completion of Phase 3 includes **“full”** implementation of the related APIs categories as defined in the section “Project Scope” under the section “Design, Test, Document and Release Specifications”

#	APIs category	Phase 1	Phase 2	Phase 3
3.6.1	ADF Data Cube	read (including the phase related implementation as defined in the section "Project Scope" under the section "Design, Test, Document and Release Specifications")	create, write, update, delete	full
3.6.2	ADF Data Description	read (including the phase related implementation as defined in the section "Project Scope" under the section "Design, Test, Document and Release Specifications")	create, write, update, delete	full
3.6.3	ADF Data Package	read (including the phase related implementation as defined in the section "Project Scope" under the section "Design, Test, Document and Release Specifications")	create, write, update, delete	full
3.6.4	ADF Quadratic Store	as required to support the above	as required to support the above	full
3.6.5	ADF Audit Trail	as required to support the above	as required to support the above	full
3.6.6	ADF Check Sums	as required to support the above	as required to support the above	full
3.6.7	HDF5	as required to support the above	as required to support the above	full
3.6.8	Other	as required to support the above	as required to support the above	full

ADF Java library packages and their relation to the APIs category:

ADF Java library package	APIs category
<i>org.allotrope.adf</i>	Other
<i>org.allotrope.adf.audit.events</i>	ADF Audit Trail
<i>org.allotrope.adf.audit.service</i>	ADF Audit Trail
<i>org.allotrope.adf.audit.service.impl</i>	ADF Audit Trail
<i>org.allotrope.adf.config</i>	Other
<i>org.allotrope.adf.converter</i>	Other
<i>org.allotrope.adf.audit.service.impl</i>	ADF Audit Trail
<i>org.allotrope.adf.config</i>	Other
<i>org.allotrope.adf.converter</i>	Other
<i>org.allotrope.adf.dc.config</i>	ADF Data Cube
<i>org.allotrope.adf.dc.config.cube</i>	ADF Data Cube
<i>org.allotrope.adf.dc.config.datatype</i>	ADF Data Cube

<i>org.allotrope.adf.dc.config.orderfunction</i>	ADF Data Cube
<i>org.allotrope.adf.dc.config.scale</i>	ADF Data Cube
<i>org.allotrope.adf.dc.config.selection</i>	ADF Data Cube
<i>org.allotrope.adf.dc.config.structure</i>	ADF Data Cube
<i>org.allotrope.adf.dc.event</i>	ADF Data Cube
<i>org.allotrope.adf.dc.model</i>	ADF Data Cube
<i>org.allotrope.adf.dc.model.scale</i>	ADF Data Cube
<i>org.allotrope.adf.dc.service</i>	ADF Data Cube
<i>org.allotrope.adf.dc.service.converter</i>	ADF Data Cube
<i>org.allotrope.adf.dc.service.internal</i>	ADF Data Cube
<i>org.allotrope.adf.dc.type</i>	ADF Data Cube
<i>org.allotrope.adf.dd.batch</i>	ADF Data Description
<i>org.allotrope.adf.dd.bimap</i>	ADF Data Description
<i>org.allotrope.adf.dd.bptree.generic</i>	ADF Data Description
<i>org.allotrope.adf.dd.bptree.string</i>	ADF Data Description
<i>org.allotrope.adf.dd.event</i>	ADF Data Description
<i>org.allotrope.adf.dd.jena</i>	ADF Data Description ADF Quadratic Store
<i>org.allotrope.adf.dd.jena.bptree</i>	ADF Data Description ADF Quadratic Store
<i>org.allotrope.adf.dd.jena.impl</i>	ADF Data Description ADF Quadratic Store
<i>org.allotrope.adf.dd.model</i>	ADF Data Description
<i>org.allotrope.adf.dd.service</i>	ADF Data Description
<i>org.allotrope.adf.dd.utils</i>	ADF Data Description
<i>org.allotrope.adf.dp</i>	ADF Data Package
<i>org.allotrope.adf.dp.config</i>	ADF Data Package
<i>org.allotrope.adf.dp.event</i>	ADF Data Package
<i>org.allotrope.adf.dp.model</i>	ADF Data Package
<i>org.allotrope.adf.dp.service</i>	ADF Data Package
<i>org.allotrope.adf.dp.util</i>	ADF Data Package
<i>org.allotrope.adf.enums</i>	Other
<i>org.allotrope.adf.event</i>	Other
<i>org.allotrope.adf.exception</i>	Other
<i>org.allotrope.adf.hashing</i>	ADF Check Sums
<i>org.allotrope.adf.hashing.hdf10</i>	ADF Check Sums
<i>org.allotrope.adf.hashing.hdf2</i>	Other
<i>org.allotrope.adf.hashing.util</i>	Other
<i>org.allotrope.adf.hdf5</i>	HDF5
<i>org.allotrope.adf.hdf5.event</i>	HDF5
<i>org.allotrope.adf.hdf5.exception</i>	HDF5
<i>org.allotrope.adf.hdf5.hashing</i>	HDF5
<i>org.allotrope.adf.hdf5.model</i>	HDF5
<i>org.allotrope.adf.hdf5.service</i>	HDF5
<i>org.allotrope.adf.hdf5.type</i>	HDF5
<i>org.allotrope.adf.hdf5.type.annotation</i>	HDF5
<i>org.allotrope.adf.hdf5.util</i>	HDF5
<i>org.allotrope.adf.ioc</i>	Other
<i>org.allotrope.adf.locking</i>	Other
<i>org.allotrope.adf.model</i>	Other
<i>org.allotrope.adf.service</i>	Other

<i>org.allotrope.adf.upgrade</i>	Other
<i>org.allotrope.adf.util</i>	Other
<i>org.allotrope.vocabulary</i>	Other
<i>org.hdfgroup.hdf5.*</i>	HDF5

4.0 Project Scope

The scope of this project includes the full lifecycle: requirements gathering, design, development, coding, testing and continuous integration of Allotrope Foundation’s new or enhanced product, and production release. Documents for all lifecycle phases shall be generated by the selected bidder: analysis, design, build, testing, and release to be shared with the Allotrope Foundation Product Team.

The selected bidder will be responsible for planning and aligning on specifications with assistance from Allotrope Foundation designated representatives.

4.1 Development Phases and Cycles

The following criteria are in scope and must be met to achieve a successful project:

#	Cycle	Details
4.1.1	Project Phases	The project will be implemented in 3 phases as detailed in the section “ <i>ADF Python API Wrappers Development Phases</i> ”. At the end of each phase a product demonstration will be conducted
4.1.2	Short weekly project meetings	A weekly meeting with the Allotrope Foundation product team will be conducted to discuss product design considerations, project execution and demonstrations. This cycle may be extended to Bi-weekly pending schedule, progress and/or group alignment.
4.1.3	Community participation	Throughout the development phases the selected bidder will get involved with the Allotrope community to seek for early testers, Python APIs evaluation and feedbacks with the goal to develop a Pythonic “looks & feel” and to elevate the “ease of use” of the ADF library. Any related communications, activities (such as α/β testing) and schedules with the Allotrope Foundation community is subject to coordination and approval by the Allotrope Foundation product team.

4.2 Design, Test, Document and Release Specifications

The following criteria are in scope and must be met to achieve a successful project:

#	Requirement	Details
4.2.1	Pythonic API	
4.2.1.1		Build and package as PyADF a Pythonic API library which wraps the ADF Java library as specified in the section <i>“ADF Python API Wrappers Development Phases”</i>
4.2.1.2		Users of the Pythonic APIs are not required to have any background nor skills in Java
4.2.1.3		Detailed requirements for the Pythonic wrappers and the Java/Python bridging solution is specified in the section <i>“Detailed Requirements”</i>
4.2.2	Testing	
4.2.2.1		Test the Pythonic APIs for completeness and correctness of output, and usability of the APIs
4.2.2.2		Testing requirements are specified in the section <i>“Detailed Requirements”</i> under the sub section <i>“Testing”</i>
4.2.2.3		Allotrope shall test the Pythonic API for conformance and usability, through a test suite provided by the vendor as well as internal expertise
4.2.3	Packaging	Detailed requirements for packaging are specified in the section <i>“Detailed Requirements”</i> under the sub section <i>“Packaging”</i>
4.2.4	SDLC and Tools	Detailed requirements for SDLC and Tools are specified in the section <i>“Detailed Requirements”</i> under the sub section <i>“SDLC and Tools”</i>
4.2.5	Documentation	Detailed requirements for Documentation are specified in the section <i>“Detailed Requirements”</i> under the sub section <i>“Documentation Generation”</i> and <i>“Content Documentation”</i>
4.2.6	Source code ownership and licensing	All content produced by this project, including but not limited to source code, testing materials, and documentation will be owned by Allotrope Foundation and may be licensed by AF in any manner it so chooses, including the release of the package using an open-source or similar permissive license. As such, it is important that the respondent’s strategy to deliver the product takes into account any licensing issues for an open source project including not only avoiding reliance on any software tools or libraries where use in software released under an open source license is not legally permissible; similarly, any copyleft or “viral” licenses should be similarly avoided. Please include in your proposal any information on how your organization can facilitate or support this as an open source project.

4.3 Timeline

All timelines are anticipated dates and are subject to change. All proposals in response to this RFP are due via email no later than 5pm EST Aug 21st, 2020. Submissions received after this date are not guaranteed to have full consideration.

Evaluation of proposals will be conducted starting Aug 21st, 2020 until Sep 4th, 2020. If additional information or discussions are needed with any bidders during this time window, the bidder(s) will be notified.

The selection decision for the winning bidder will be made no later than Sep 8th, 2020, and contract negotiations with the winning bidder will commence immediately.

Notifications to bidders who were not selected will be completed by Sept 30th, 2020 or as appropriate.

The specific dates and timeline given in this section are subject to change without notice.

5.0 Budget

All proposals must include proposed costs to complete the tasks described in the scope for each of the 3 phases as detailed in the section *“ADF Python API Wrappers Development Phases”*. Costs should be stated separately for each of the 3 phases as one-time, non-recurring costs (NRC) or monthly recurring costs (MRC) if applicable.

5.1 Additional Budget Considerations

- Travel Time will not be reimbursed
- Pre-approved Travel Expenses will be subject to the Allotrope travel guidelines which will be provided to the selected vendor

6.0 Bidder Qualifications

The organization selected and contracted to conduct the work effort must be a member in good standing of the Allotrope Partner Network (APN) on or before the date the contract is executed and throughout the duration of the contract.

Bidders should provide the following items as part of their proposal for consideration:

- Description of experience in designing, building, and deploying an integrated software product solution related to or within semantics or similar technical taxonomy or ontology-based product
- List of how many full time, part time, and contractor staff in your organization, as well as which staff member(s) would be supporting Allotrope Foundation’s Support & Maintenance contract if known
 - Anticipated resources you will assign to this service (total number, role, title, experience)
- Examples of 3 or more products designed and implemented or supported by your organization
- Testimonials from past clients on similar or related projects
- Details of applicable best practices with regards to software development processes which are currently used within your company as they relate to this RFP
- Project management methodology

6.1 Specific Technical Questionnaire to the Bidder

As it relates to your team's experience and expertise, please provide high level examples or references (if any exist) to projects or integrations of the following:

#	Domain	Please provide some details or references
6.1.1	General Pythonic API library or SDK development	
6.1.2	General Java API library or SDK development	
6.1.3	ADF library API	
6.1.4	HDF5 and h5Py library	
6.1.5	JPytype, wrappers or any other Java-Python "bridging" code	
6.1.6	C++ library software development (may be required for modifications or extensions within the JPytype backend)	

7.0 Access to Applicable Material for Response

For non-Allotrope APN members to gain access to relevant API information, please inquire to the below email for the necessary paperwork.

8.0 Proposal Evaluation Criteria

Allotrope Foundation will evaluate all proposals based on the following criteria. *Please note that as a policy Allotrope Foundation does not share or discuss the results of the proposal evaluation process nor provide any feedback why a bidder was not selected.*

To ensure consideration for this Request for Proposal, your proposal should be complete and include all the following criteria:

- Overall proposal suitability: proposed solution(s) must meet the scope and needs included herein and be presented in a clear and organized manner
- Organizational Experience: Bidders will be evaluated on their experience as it pertains to the scope of this RFP
- Previous work: Bidders will be evaluated on examples of their work pertaining to the technical design and or support as applicable, as well as client testimonials and references if included
- Value and cost: Bidders will be evaluated on the cost of their solution(s) based on the work to be performed in accordance with the scope of this RFP
- Technical expertise and experience: Bidders must provide descriptions and documentation of staff technical expertise and experience

8.1 Peer Questions and Answers

In the event that clarifying questions are asked to the Allotrope team, the prudent questions will be deidentified as necessary to shield the requester and then the questions and answers will be posted to the Allotrope website where the RFP document is posted. This is to keep all interested parties informed of insights or oversights by the Allotrope team and to ensure an equal opportunity to all potential bidders.

8.2 Submission

Each bidder must submit their proposal to the email address below by the day and time noted above in Section 4:

Allotrope Foundation Product Team Email Contact for RFP Submission: matthew.fox@allotrope.org

9.0 References

1. Description of Allotrope Framework: <https://www.allotrope.org/allotrope-framework>
2. ADF Java library APIs: <https://community.allotrope.org/files/apidoc/index.html>
 - Note: This is only available to Allotrope Foundation Members. If you are interested in attaining a trial membership, please contact Allotrope at the email listed above and/or secretariat@allotrope.org
3. ADF Java library version 1.5.x Client Connect link: <https://highq.in/1vj1ajsqng>
 - Note: This is only available to Allotrope Foundation Members. If you are interested in attaining a trial membership, please contact Allotrope at the email listed above and/or secretariat@allotrope.org
4. HDF5: <https://www.hdfgroup.org>
5. h5py: <http://api.h5py.org/>
6. JPytype: <https://jpytype.readthedocs.io/en/latest/>
7. RDF Data Model: <https://www.w3.org/RDF>
8. W3C: <https://www.w3.org>