Contractor Performance Work Statement – Task Order #1
Earthquake Science Center ShakeAlert® Scientific Software Developer
Performance Work Statement

1.0 GENERAL INFORMATION/SCOPE

The USGS, in partnership with universities like the California Institute of Technology (Caltech), UC Berkeley, the University of Washington, and other partners, has developed the ShakeAlert® earthquake early warning system. Alerts are currently being sent to the West Coast via different mechanisms. The intent of this project is to send both accurate and timely alerts to give as many people as possible a warning before heavy shaking is experienced. The system uses seismic and geodetic data from stations in the field. The field data are relayed to data centers for processing and analysis, to determine in real-time if an earthquake has occurred. If an earthquake is detected and the magnitude of the event and anticipated ground motion of the shaking are above specified thresholds, then alerts are sent through a variety of mechanisms to warn the public and inform technical users. Because ShakeAlert® is a cooperative project between USGS and university partners the contractor will be expected to collaborate with USGS and university scientists, software developers, and IT professionals.

This work can be accomplished remotely, with an expectation of attending ShakeAlert® tele-meetings and video conferences, engaging in frequent communication with the Earthquake Science Center (ESC) Pasadena office, and visiting occasionally.

2.0 SCOPE OF WORK:

Contractor Requirements to Support ShakeAlert®:

2.1 Task I. Geophysical Software Development, Test, and Documentation (60%)

(A) Become an expert in an existing Python code base for testing and ShakeAlert® as a whole, revise this code base, and write new Python code for geophysical applications. The goal is to write high quality scientific code that will help test various ShakeAlert® applications in a messaging environment. Multiple messaging protocols are used, including ActiveMQ, NATS, and earthworm. In addition, ShakeAlert® applications are written in several languages including C++, Java, Python, and Golang.

(B) Become an expert in existing BASH scripts, revise them, and write new BASH scripts as appropriate to accomplish scientific and programming objectives.

(C) Work with existing Golang, MATLAB/Octave/ShakeMap and GMT code and associated configurations for the testing environment. Verify code results for scientific soundness.

(D) Generate documentation for the code base on the GitLab wiki.

(E) Participate in planning meetings for mapping out the future development of the testing platform code for new ShakeAlert® scientific algorithms, applications, and supporting code.

(F) Develop new Python code to deposit and extract test platform results into a database using PostgreSQL, MongoDB, or other appropriate database languages.

(G) Develop and maintain tools to appropriately run the ShakeAlert® testing platform in a cloud computing environment, be it in the Amazon cloud or another cloud provider. This includes using Infrastructure as Code (IaC), tools such as Terraform or the AWS Cloud Development Kit (CDK), automation software for configuration management and deployment (e.g. basic scripts, Ansible, etc.), as well as analyzing results for scientific soundness.
(H) Help redesign the ShakeAlert® testing platform for cloud computing architecture.
(I) Develop build pipelines from Gitlab through automated deployments and testing in AWS or other cloud infrastructure.

2.2 Task II. Websites, Web Applications and Database Maintenance and Development (20%)

(A) Learn the code bases for various ShakeAlert® websites, web applications, and for calls to associated databases, specifically PostgreSQL and MongoDB. This will involve back-end programming (Django, C, etc.) and front-end programming (JavaScript, etc.).
(B) Develop new code for updating and maintaining the database calls.
(C) Develop new code for updating and maintaining the websites and web applications. This includes ensuring the web content is scientifically accurate.
(D) Ensure all website access uses strong authentication and properly encrypted connections to protect user credentials and data, and adhere to NIST Special Publication 800-53 Revision 5 – Security and Privacy Controls for Information Systems and Organizations.

2.3 Task III. Running Tests (20%)

(A) Learn how to run routine tests of ShakeAlert® algorithms using the code base for the testing platform.
(B) When needed, run routine tests of ShakeAlert® algorithms, analyze results for scientific soundness, and document results.
(C) Help generate and automate the creation of new scientific data sets for testing using appropriate scripts.

2.4 The Contractor Will:

- Be required to attend (in person or by participating in a teleconference or videoconference) ShakeAlert® meetings as scheduled and respond to communications (via phone, email etc.) within one business day if possible.
- Become an expert in the existing ShakeAlert® applications, supporting code, and testing platform code. This includes developing a thorough understanding of the geophysics underpinning the code.
- Develop new code in Python, C, BASH scripting, or other languages as needed for further development of ShakeAlert® applications, supporting code, and testing platform.
- Develop cloud computing tools and help redesign the testing platform for cloud computing architecture.

- Maintain and develop further ShakeAlert® websites, web applications, and associated databases.
- Learn how to run tests and perform them as needed. Determine if test results are physically reasonable and scientifically sound.
- Work with PostgreSQL and Mongo DB databases
- Check code into the ShakeAlert® GitLab code repository and use version control.
- Document code on the ShakeAlert® GitLab wiki.
- Provide weekly reports on progress made.

3.0 TECHNICAL/PERFORMANCE REQUIREMENTS

ShakeAlert®'s operations depend on:

- Approximately 1000+ seismic stations and potentially hundreds of geodetic stations distributed along the West Coast, with data telemetry to distributed data centers.
- Networking and system administration of computers running CentOS and Red Hat Linux operating systems, with Virtual Machines (VMs).
- ActiveMQ and NATS messaging systems to pass waveform-derived data between different West Coast data centers.
- C++ ShakeAlert® algorithms that process waveform data and associate information to generate earthquake alerts and downstream ground motion products.
- XML alert messages that are then sent to Technical users and to a program that generates WEA messages for public dissemination.
- Replay servers and test machines for evaluating new candidate algorithm performance before deploying to the production environment.
- Testing scripts and code in Python and other languages, to replay data sets through candidate algorithms, analyze performance, calculate warning times, and generate estimates of ground motions.
- PostgreSQL database for scientists to query test results, MongoDB/website for visualization of test results, web application and database for analyzing ShakeAlert® performance, and various other ancillary applications.

3.1 TECHNICAL REQUIREMENTS

In the performance of these tasks, the contractor shall:

i) Review and learn existing code bases to modify, extend, and write new code for the set of tools related to testing of ShakeAlert® algorithms and ancillary applications. This will include interfacing with a PostgreSQL database and a MongoDB, as well as thoroughly understanding the geophysics underlying ShakeAlert®.

ii) Review and learn existing BASH and Python scripts to modify, extend, and write new scripts to establish and streamline ShakeAlert® testing tasks.

iii) Review and learn existing Python/GMT/MATLAB/Golang/Octave/ShakeMap code/config for the testing environment to modify, extend, and write new code for enhanced functionality.

iv) Review and learn existing code for ancillary ShakeAlert® applications, websites, and web applications to modify, extend, test, and write new code.

v) Work on remote servers over SSH, use command-line shell utilities for development, system administration and troubleshooting.
vi) Become sufficiently knowledgeable of the specified cloud environment to create tools for the cloud.
vii) Learn code base for ShakeAlert® websites, web applications, and database calls which will involve knowledge of front-end languages like JavaScript and back-end languages like Django and C. Maintain these code bases and add features as requested. If the manager agrees, the contractor can rewrite code using different front-end and back-end languages.
viii) Use and maintain the Gitlab repository including checking in new code, merging new code into the relevant master branch and testing viability of the new code with unit/system tests.
ix) Generate documentation for the Gitlab wiki on processes and code.
x) As needed learn and run routine tests of candidate ShakeAlert® algorithms. Analyze the results for physical reasonableness and scientific soundness, then document results.
xi) Generate new test event data sets as needed from existing geophysical data archives.
xii) Participate in ShakeAlert® meetings which will involve different working groups and planning out code development in collaboration with others.

3.2 KNOWLEDGE AND QUALIFICATIONS

To perform these tasks, it is expected that the contractor has the following knowledge and experience:

- Thorough knowledge of geophysics, seismology, physics, and / or mathematics to support developing and enhancing software for geophysical applications.
- Excellent programming language experience.
- Excellent proficiency working on remote servers over SSH and demonstrated strong fluency with command-line shell utilities for development, system administration, and troubleshooting.
- Experience with the following Python packages would be helpful:
    - ObsPy, NumPy, and Matplotlib.
- C/C++ experience sufficient to read, understand a pre-existing code base, and develop new code.
- Experience with cloud computing.
- BASH scripting skills.
- Experience with CentOS and / or Linux operating systems.
- Experience with source code management.
- Experience with databases.
- Experience with front-end and back-end web design including JavaScript and Django.
- Ability to learn new programming languages as needed.

Given the highly collaborative nature of the job, the contractor is expected to have the following soft skills:

- Good communication skills.
- Ability to work well on a team with a positive attitude.

- Comfortable with an agile development environment, where programming goals are refined and iterated upon as initial results are analyzed.

Other programming languages/technical skills used by the testing platform the contractor is expected to become familiar with after starting the job:

- C/C++
- Golang programming language
- MATLAB or Octave programming languages
- GMT programming language
- ShakeMap configurations and running the modules
- ActiveMQ messaging
- NATS messaging
- Earthworm package for earth science data processing
- Gitlab source code management
- MongoDB
- PostgreSQL

4.0 ACCEPTANCE/MEASUREMENT

The Task Owner will set priorities and goals, deadlines for any product deliveries, and evaluate the contractor's performance on all contract requirements during the period of performance of this contract. The contractor's performance will be measured against the following relevant criteria:

i) Geophysics, Seismology, Physics, and / or Mathematics Performance: Acceptable Performance Standards: Provide analysis of code results and accurately assess their physical reasonableness and scientific soundness. Meet target deliverables on time 80% of the time. Assessments must be legible to a knowledgeable technical audience.

ii) Python Coding Performance: Acceptable Performance Standards: Meet target deliverables on time 80% of the time. This includes generating unit tests and system tests for the deliverables. The code delivered will pass unit/system tests, and work without crashing. Code will follow standard coding practices and be checked into the Gitlab code repository.

iii) GMT/Golang/MATLAB/Octave Coding Performance: Acceptable Performance Standards: 1-2 weeks to generate rudimentary familiarity with a programming language if it is brand new to the contractor. Meet target deliverables on time 80% of the time. The code must work without crashing, follow standard coding practices, and be checked into the Gitlab repository.

iv) Cloud Computing Performance: Acceptable Performance Standards: 1 month to generate familiarity with the cloud environment if it is brand new to the contractor. Meet target deliverables on time 80% of the time. The code must work without crashing, follow standard coding practices, and be checked into the Gitlab repository.

v) Website Programming Performance: Acceptable Performance Standards: Meet target deliverables on time 80% of the time. Use a standard front end like JavaScript and a back-end language like C or Django. Front-end and/or back-end can be

rewritten in different languages if the rewrite is pre-approved by the USGS and accomplished in 1 month. Code must work without crashing, follow standard coding practices, and be checked into the Gitlab repository.

vi) BASH Scripting Performance: Acceptable Performance Standards: Meet target deliverables on time 80% of the time. Scripts must work without crashing, follow standard coding practices, and be checked into the Gitlab repository. Scripts will be both effective and efficient.

vii) Documentation Performance: Acceptable Performance Standards: When a coding deliverable is completed (a modification, extension, or addition to the existing code base), the contractor is expected to generate documentation on the Gitlab repository wiki. When the run scripts for relevant packages are changed, the contractor will document these changes in an email. Last, the contractor will generate weekly reports of tasks accomplished.

viii) Gitlab Performance: Acceptable Performance Standards: It is expected that the contractor will check in all new code to the repository. When a development module is completed, if the module passes unit tests and system tests, the contractor will merge the finished module into the master branch of the GitLab repository. It is expected that the contractor will be able to generate new branches, merge branches, and help maintain the test suite GitLab repository.

ix) Testing Performance: Acceptable Performance Standards: When requested, the contractor will complete routine testing of algorithms, analysis of results, and development of documentation with a three-week turnaround.

x) Communication Performance: Acceptable Performance Standards: Contractor should attend (in-person or remotely) the ShakeAlert® meetings as scheduled and stay in communication with the technical liaison and others throughout the week as necessary. Contractor will be expected to respond to communications (i.e., phone calls, emails, etc.) within one business day of receipt. Contractor attendance at weekly meetings is required and should exceed 90% attendance.