**Pre-meeting info about the API and system**

The swagger API definition files in the attached tarball are intended to provide insight regarding our system's API. The API is still being developed, so it will probably be a bit different when we release the system.

The system my project is developing (tentatively named "SCAIFE") is intended to provide an architecture with APIs and an open-source prototype system to enable users to:

- Quickly start to use automated classifiers for their static analysis alerts. It does NOT require:
  - a large labeled audit archive ahead of time (or even *any* labeled audit archive)
  - a statistics expert (a stats expert could tweak and improve the system for a particular org)
  - the user to create their own framework for using classifiers
- Quickly start to use powerful prioritization formulas that allow users to prioritize static analysis alerts using factors they care about. These prioritization formulas can be quite sophisticated, combining classifier-derived confidence with mathematical symbols. The formulas can combine classifier confidence and other values (e.g., risk, cost, etc.) used by the system, using math symbols including: '*', '/', '+', '-', '(', and ')'
- Use the API definition to build upon the original prototype system, to enable use of additional flaw-finding static analysis tools, code metrics tools, adaptive heuristics, classification techniques, etc.

Our architecture involves 4 servers, with API calls for all communications between the servers. The "UI Module" server is where SCALe, CERDEC SwAT, and DHS SWAMP will go. Integration challenges include working with fields in some but not others, different meanings of concepts (e.g., project and package), and DHS SwAMP third-party viewing systems. The other 3 servers are:

- Classification Module (handles classification, adaptive heuristics, reclassification, etc.)
- Prioritization Module (stores, retrieves, and modifies prioritization formulas, enabling sophisticated formulas that can combine classifier confidence and other values (see details at top of page)
- DataHub Module (stores, retrieves, and forwards data for other servers)

To extract the files, in a bash shell terminal, you can use this command:

```
tar -xvf shareAPI_20180806.tar.gz
```

We have developed a RESTful API with the widely-used swagger tool. (We used open-source `swagger-editor` v2, testing with `swagger-ui`, and automated code generation with `swagger-codegen`). More info on REpresentational State Transfer here.

For recipients who are not already using the `swagger-editor` tool, there is no need to install it. We recommend you view the API using the `rapid_models_API.html` file. The file `rapid_models_API.html` can be viewed as a file in your browser. E.g., in your browser you can enter the following URL (substitute your own filepath for the blue-font text):

```
file:///<YOUR_PATH_TO_DIRECTORY_FILE_IS_LOCATED>/rapid_models_API.html
```

For recipients who are already using the `swagger-editor` tool, the .yaml files are included for you. They are best viewed in the swagger-editor. We've included 4 separate .yaml files, one per server. Also, for ease of viewing, two files include API calls between all the servers (one with some example data).

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.


DM18-0927

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.