## software IP Submission FORM

INFORMATION FOR COMPLETING THE SOFTWARE IP SUBMISSION FORM

1. Software IP (SWIP) creators or Software Developers summarize the key information/concept of the software generated from the work done at I2R. This information will be treated as confidential and will be used to justify for subsequent exploitation of the software.
2. Subject matter in each software should be described clearly outlining **the** technical merits, usefulness, and applications of the software.
3. Software IP (SWIP) bank is repository of software codes developed during a course of work, that could be reused as fit for use software. SWIP may be derived from associated TDs (Technology Disclosure) / Patents or from project(s). Engineering effort is essential to create reusable software. It should be made sure the the SWIP, from a project, has not violated any legal terms (example: if a project has contractual obligation that the FIP (Foreground IP) belongs to the collaborator and can not be used in other projects, then that SWIP can not be submitted). However if a software IP is exclusively licensed and there is restriction of use of software developed under the contract, then the SWIP cannot or might not be reused within the stipulated duration set forth in the contract.
4. Please DO NOT submit the duplicate copy of project archival or Prototype archived which was submitted to PMO. SWIP is reusable software NOT the project archival and should not have project specific customer data / confidential information.
5. SWIP source code/ documents/library/Executable should be ready to upload in gitlab within 14 working days upon submitting the form. This is to plan accordingly the review meeting of the SWIP submitted. SWIP creator(s) must make sure uploading in gitlab of the complete code, libraries, documents which can be built, installed and run as working software.
6. SWIP is accounted on FY basis for KPI counting. However internally I2R keeps track of SWIPs as CY (Calendar Year) basis for Depatment/Unit achievement records. **The last date for submission of SWIP, in a calender year, is by the 10th of December**. Submitter shall plan accordingly to submit the SWIP to be counted for the said CY. SWIP is counted in the CY upon review and approval by the review panel.
7. Upon approval of the SWIP by reviewers, this submit form is then emailed to A\*ccelerate OIC to record as the approved SWIP. Evaluation records, approved by reviewers of the SWIP, are maintained internally by SWIP coordinator.

Please submit the original, completed Software IP submission form (hardcopy), **duly signed,** and in pdf form, to:

D Banerjee, SWIP Coordinator - Special Projects

[banerjee@i2r.a-star.edu.sg](mailto:banerjee@i2r.a-star.edu.sg), DID: 64082817

#### Software IP Submission Form

|  |
| --- |
| 1. **TITLE OF SOFTWARE IP** *(a short but sufficiently descriptive title to identify the general nature of the software.)* |
| **Ethereum-IPFS database authentication system** |

|  |  |
| --- | --- |
| 1. **DESCRIPTION OF THE SOFTWARE** *Please provide as complete a description as possible. This is essential for: A. The purpose of your description is to enable a person with similar skills in your field to be able to make and use the software you submitted. B. Please do not withhold any key elements of the software submitted* | |
| 2.1 **Field of The Software:**  *A sentence or paragraph identifying the general* ***field of technology*** *to which the software relates, which* ***Department/unit/clusters*** *could potentially reuse this software? How the problem was previously solved?*  *Authentication and automatic logging on database that could be potentially used for Cybersecurity as well as any other department with needs of auditing on database access. Previously, passwords are used to protect database, and a separate log server is required if auditing/logging is required.* | |
| 2.2 **Brief summary of the software:** *A brief paragraph (similar to the abstract of a TD) describing the key feature(s) of the software with some background context.* *What is its core?* A solution that utilizes Ethereum blockchain to authenticate the user before accessing database, and naturally log the authentication information into the blockchain through smart contract. Meanwhile, a distributed database solution is used to store the DB output in coordinate with the Ethereum system. | |
| 2.3 **Detailed Description of the Software:**  *This section should be detailed enough for a person having ordinary skill in your technical field to construct and use the software you describe.*  (i) A detailed description of the software with key technical features, how it works and to give example of usage scenarios, where possible. The software is built based on the project requirements to achieve database authentication in modern factory, where users need to access the database and store the output into another distributed data storage system. The whole process should be logged into a system that is unchangeable.   (ii) Does your software possess any of its disadvantages or limitations? Can they be overcome?  Due to the blockchain algorithm we select(Ethereum), the transaction speed is limited, which means that it takes time for the user to retrieve the data from DB. Unfortunately this cannot be resolved if we stick to the current design structure.  The solution is currently only working on Linux system (and only tested under ubuntu 16.04 and 18.04, yet should be working on all linux based system) due to some libraries only provide support to Linux system. This can be overcome rewriting the libraries for Windows system in the future.  iii) What are the programming language(s) for the software implementation and what platform(s) is it designed for delivery?  Python3  iv) Is this software?SDK Library UI based Application Command line Application Mobile App Web App Others *(You can select multiple if applicable)*  If ‘Others’, please state: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| 1. **COMMERCIALISATION** | |
|  | |
| i) List the following details of associated TD/Patent/Project(s) from which the software IP is originated   |  |  |  |  | | --- | --- | --- | --- | | TD/Patent #  (if any associated TD/Patent) | TD/Patent Title | PSN (e.g. EC-20XX-YY)  associated project from which the SWIP is originated | Remarks | | Nil | Nil | Nil | N.A. | |  |  |  |  | |  |  |  |  | | |
| ii) Is any part of the source code obtained under an **Open Source** license (e.g. BSD, GPL, MIT, Apache, etc.)?  Yes No  If yes, please provide a list of the sources:  *(Use of GPL is not recommended for its stringent terms and conditions towards distribution and commercialization)*   |  |  |  |  | | --- | --- | --- | --- | | Name/Title of the open source software used, source url and version # | License type (e.g., BSD, MIT, Apache, GPL etc.) | Has the open source used as a library (Y/N)? | Has the source code of the open source modified (Y/N)? | | Pyethereum, <https://github.com/ethereum/pyethereum/tree/snapshot> | MIT | Y | N | | Py-ipfs, <https://github.com/ipfs/py-ipfs> | MIT | Y | N | | Flask, <https://github.com/pallets/flask> | BSD 3 | Y | N | | Py-solc,  <https://github.com/ethereum/py-solc> | MIT | Y | N | | |
| iii) Does the software require any **3rd party / commercial software** (e.g. a 3rd party library?) Yes No  (3rd party software is NOT an open source. Example: a shareware/freeware)  If yes, please provide a list of the sources:   |  |  |  |  | | --- | --- | --- | --- | | Name/Title of the 3rd party software used, source and version # | Type of the 3rd party software (Freeware/ Shareware /commercial) | Is it Free of Cost (Yes/No)? | Remarks | |  |  |  |  | |  |  |  |  | | |
| iv) Are there any third-party rights associated with the creation of the software? List grants or contracts if any, with third parties. | |
| v) What is the TRL (Technology Readiness Level) of the software (**TRL: 1-9**)? \_\_\_\_\_\_6\_\_\_\_\_\_\_\_\_\_\_    [ Please refer to this link for more information on TRL: <https://intranet.i2r.a-star.edu.sg/IPO/Shared%20Documents/Software%20QC%20contents/TRL-info.pdf>]    Is the software?  A working prototype Tested in fully functional end user version?  (if used/tested in user environment, please give details) | |

|  |  |
| --- | --- |
| 1. **SUBMITTER/S’ PARTICULARS & DECLARATIONS** *(Signatures are required.)* | |
| I / We\* hereby declare to the best of my / our\* knowledge the information provided in this software submission form are true and correct. The uploaded software in gitlab is the complete code, libraries and team has installed the software successfully to run and test as working software. | |
| **Software IP creators/developer(s) Names**  *(****Please note*** *that SWIP is the engineering effort to make a reusable software with quality. Thus the creator(s) should be those involved in realization of the software. The list of the SWIP creators or developers may not be the same as inventors in the associated TD/Patent. Include only those creators for the Software IP.)* | |
| **Principal** SWIP Creator’s **Name, Signature & Date**  **Deng Gelei 15/11/2019** | Department/Unit/Programme  Cybersecurity |
| Percentage and Aspects of Contribution (***must enter both % and aspects****. Example of aspects: design, architecting, coding, integrating, validating etc*. *Non-engineering* *aspects such as involved in coordination, procurement, meetings etc. will not qualify for creator’s contribution)* | |
| **Other creators’ names and contributions** (add more creators if applicable) | |
| ii) SWIP Creator’s Name, Signature & Date | Department/Unit/Programme |
| Percentage and Aspects of Contribution (***must enter both % and aspects****. Example of aspects: design, architecting, coding, integrating, validating etc*. *Non-engineering* *aspects such as involved in coordination, procurement, meetings etc. will not qualify for creator’s contribution* | |
| iii) SWIP Creator’s Name, Signature & Date | Department/Unit/Programme |
| Percentage and Aspects of Contribution (***must enter both % and aspects****. Example of aspects: design, architecting, coding, integrating, validating etc*. *Non-engineering* *aspects such as involved in coordination, procurement, meetings etc. will not qualify for creator’s contribution* | |
| iv) SWIP Creator’s Name, Signature & Date | Department/Unit/Programme |
| Percentage and Aspects of Contribution (***must enter both % and aspects****. Example of aspects: design, architecting, coding, integrating, validating etc*. *Non-engineering* *aspects such as coordination, procurement, meetings etc. will not qualify for creator’s contribution* | |
| **Supported by: -** | |
| Name, Signature of Head of Department/Unit/Programme and Date  Guo Huaqun 15/11/2019 | |

**Evaluation**

[This table to be filled by reviewer ONLY]

|  |  |
| --- | --- |
| **SWIP/Title** | **Reviewed by** |
| <SWIP ID> /  <Title> | \*Reviewer1, Reviewer 2, Reviewer 3 |

\*Consolidate ratings, assessment and approval

**Software Acceptance Criteria**

[**Submitter** shall fill the relevant column as shown]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Topics** | **Criteria** | **Please tick √**  [to be filled up by **submitte**r] | | | [To be filled up by review committee] | **Remarks** |
| **Yes** | **No** | **NA** | **Rating Scale (1-5)**  **5** = Outstanding  **4** = Good  **3** = Satisfactory  **2** = Poor  **1** = Unsatisfactory |
| Software Usability and Maintainability | Documentation | †**Technical guide document** (to prepare using the given template and to be submitted later prior to review) contains release note, installation steps, configuration and how to re-compile & re-build the software. It should spell out the operating computing requirement and environment to run the software, step by step user guide and technical information. |  |  |  |  |  |
| Documentation | SDK Manual / API documentation (if applicable) is mandatory for SDK / libraries software release. (source code documentation be generated using tools like Doxygen) |  |  |  |  |  |
| Documentation | All functions/procedures in source code must have a clear and concise comment blocks with input, output and processing description for easier understanding and maintenance.  *Note: Comments with the source codes should focus on function interaction, design approach rather than literally explaining what the code does.* |  |  |  |  |  |
| Software Release Readiness | Coding Standard | Comments / code ratio is recommended to be about 40%. |  |  |  |  |  |
| Coding Standard | Software Version is displayed on application GUI (if applicable) correctly during application execution |  |  |  |  |  |
| System Testing | System Test Cases covering 100% of the features has been created and test results are documented. |  |  |  |  |  |
| Memory Leak Test for C/C++ | Memory Leak Test has been conducted and passed. All defects found are recorded in defect tracking tool. |  |  |  |  |  |
| †Unit Testing | Unit Testing has been conducted and unit test results are documented. |  |  |  |  |  |
| †Code Review (Static Code Analysis) | Static Code Analysis has been performed. Its results have been reviewed. |  |  |  |  |  |
| †Defect Tracking | Defect tracking (gitlab issue tracker tool) is used to track defects found during software development |  |  |  |  |  |
| Software Quality | Defect Tracking | There is no known open critical and major defects for a release software |  |  |  |  |  |
| Modular Design | Application are partitioned into different standalone libraries with no dependencies on each other |  |  |  |  |  |
| Modular Design | Each module has well defined APIs |  |  |  |  |  |
| †Coding Standard | Cross Platform Programming Languages should be used.  Preferred Language: C/C++, Java, Python, PHP, JavaScript, R, Objective C. If other language is used, a justification is needed |  |  |  |  |  |

† **Technical guide template,** User guides on Unit testing, code review (aka static code analysis), defect tracking and Coding standard are available in intranet in SWIP site .The artefacts need to be uploaded in gitlab at least 3 working days prior to review.

**[The following section is ONLY for reviewers to fill for post review evaluation. The section will be filed as separate evaluation records and not filled with the submit form]**

Date of Review: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Overall Assessment**:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Approved to be deposited in software IP bank?**  Yes/No

Reasons:-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name, Signature of Reviewer and Date