# PySyft documentation audit, update & expansion into a standardized framework - OpenMined

## About your organization

OpenMined began in 2017 and is an open-source and not-for-profit community of now over 15,000 academics, engineers, mentors, educators, ethicists and privacy enthusiasts committed to making a fairer and more prosperous world by building and teaching privacy enhancing technologies. PySyft (current version 0.6.0, first release 2020) is an Apache 2.0 licensed library for secure and private Deep Learning in Python.

OpenMined continues to build and curate an ecosystem of open-source privacy enhancing tools that span techniques including homomorphic encryption, secure enclaves, federated learning, differential privacy, zero knowledge proofs and much more. We work to solve real-world privacy problems for a variety of important use cases.

## Problem Statement

OpenMined is dedicated to solving various problems in the field of online privacy. In particular, we have made great strides in the areas of remote execution, distributed computing, cryptography, and differential privacy - but **our documentation has been a source of confusion and discouragement for our community members**. The basic examples and tutorials illustrate the use of PySyft, but the examples are quickly broken by the fast pace of development. This ends up burying our development teams in support requests that are hard to keep up with.  The **lack of PySyft documentation is making it very difficult to onboard new developers** onto our teams, who are currently forced to read the codebase manually without proper direction or guidance.

We requested additional support in an effort to provide new guides covering functionalities of PySyft, as well as producing public API references, and general readme documentation.

## Proposal Abstract

PySyft makes it easy for machine learning engineers and researchers to train deep learning models using remote data and thus preserve privacy while using well-known libraries such as PyTorch and Tensorflow.

As we onboard more institutions and deploy larger-scale systems using federated learning, multi-party computation, and other privacy-preserving techniques, the ability to easily navigate the internal codebase is crucial to understanding and using the library properly. Moreover, many people worldwide join our channel to start contributing to PySyft, but overcoming the complexity and becoming familiar has a steep learning curve.

The goal of the project is to audit the existing files within the codebase, identify the gaps in the documentation and specifically improve inline documentation, the readme files and general API references. We will also identify core use-cases of PySyft and show how to use it to preserve privacy in different scenarios. We will build a Jupyter notebook containing tutorials and examples that encompass data analysis, model training, different branches of machine learning (e.g computer vision, NLP) focusing on how PySyft can be used in conjunction with other libraries in the OpenMined ecosystem.

OpenMined’s GSoD Project Ideas page

# Project Description

## Creating the proposal

OpenMined’s proposal developed out of an immediate need for dedicated technical writing resources to devote time and energy to building documentation of the PySyft library as it matures over time. As a force multiplier, documentation provides canonical reference materials that can be reviewed, actioned and tested in an independent manner. Gaps in documentation are quickly identified and remedied in a short period of time for the benefit of all.

## Budget

We utilized the full amount of granted funds for our project. An open collective host fee of 4% was unfortunately overlooked. This fee was deducted equally from funds budgeted for administration and team building. We spent the remaining amount of funds as planned with the remaining team building funds going directly to technical writers in equal measure. No other funds were allocated beyond what was granted through GSoD.

## Participants

Mrinal Walia - PySyft Tutorials
Shubham Palriwala - PySyft Internal Docs
Irina Bejan - Documentation Team Lead
Laura Ayre - Organizational Administrator

Once it was announced that OpenMined was successful in our application to GSoD, we announced this exciting development in our slack channels and gathered proposals from interested applicants throughout our community. We received over 25 applications, out of which we filtered the ones with the necessary experience, conducted interviews with the qualifying applicants and selected 2 best-fit technical writers to focus upon our GSoD project. @abhiwalia15 has extensive contributions to technical blogs, while @ShubhamPalriwala has great programming and DevOps skills which are very useful in tackling the more technical tasks.  Moreover, we would like to highlight the importance of choosing people who are genuinely interested in the mission and can connect well to it, as well as people whose values stay close to Open Source community core values. As a result, both technical writers were highly engaged and blended really well into our distributed team. Additionally, @IrinaMBejan had previously led documentation efforts within OpenMined and has been a long-term member of the community. She is knowledgeable of documentation needs, our product roadmap and is a good connector between technical writers and stakeholders.

## Timeline

May 2022 - Dec 2022

| Milestones | Key results | PRs | Duration (part-time work |
|---|---|---|---|

| | | | hours) |
|---|---|---|---|
| PySyft ramp up & audit | ● Recruit & hire 2 technical writers.<br>● Getting technical writers familiar with PySyft's vision, existing functionalities and target personas for documentation<br>● Collaboration kickstart with our lead engineer and product to connect the identified documentation gaps with organizations's immediate goals | - | 4 weeks |
| PySyft's README | ● Make the PySyft repository's README concise, comprehensive and engaging | #6633 | 2 weeks |
| Source code documentation | ● Add functionality to generate the source code documentation into Sphinx docs | #6598<br>#6815 | 2 weeks |
| Installation guides | ● Uniformize existing installation guides across macOS, Linux and Windows and create a generic installation guide | #6626<br>#6629 | 1 week |
| Contributor guidelines webpage | ● Update contributor guidelines to be more welcoming with new open-source contributors<br>● Include all critical information to start contributing | #6600 | 1 week |
| Resources page | ● Create a one-stop page to include all learning materials related to PySyft, given users reported a difficult learning curve | #6617 | 1 week |
| Quickstart installer | ● Make PySyft installation and deployment process easier, as it is currently bumpy and weak documented<br>● Create an onboarding experience for new users that checks the health of current install (libraries version, Docker), installs missing dependencies and helps launch first series of tutorials interactively | #6639<br>#6531<br>#6654<br>#6686 | 3 week |
| Data Owner How-to-guides | ● Create a series of tutorials for the Data Owner persona to showcase the functionalities developed for our latest release (0,7), including:<br>○ Part 1: Deploying your own Domain Server<br>○ Part 2: Uploading Private Data to a Domain Server<br>○ Part 3: Creating User Accounts on your Domain Server<br>○ Part 4: Joining a Network<br>○ Part 5: Creating a Network<br>○ Part 6: Configuring Privacy Budget on your Domain Server | #6693<br>#6800<br>#6992<br>#6663<br>#6669<br>#6789<br>#6759<br>#6837<br>#6855<br>#6860<br>#6878<br>#7099 | 9 weeks |

|  |  |  |  |
|---|---|---|---|
|  | ● Create a uniform styling for PySyft Docs<br>● Tutorials have a clear learning objective<br>● Tutorials help users with no prior experience with PySyft get across most important user journeys for Data Owner (tested within tinker session with new users)<br>● Tutorials come with a comprehensive guide and an attached notebook, which can also be followed independently<br>● Tutorials series are easy to discover (included in the README, website page)<br>● Tutorials include illustrations where needed to better explain the concept of PySyft |  |  |
| Data Scientist How-to-guides | ● Create a series of tutorials for the Data Scientist persona to showcase the functionalities developed for our latest release (0,7), including:<br>○ Part 7: Connect to a Domain<br>○ Part 8: Searching for Datasets on the Domain<br>○ Part 9: Exploring a Dataset in the Domain<br>● Tutorials have a clear learning objective<br>● Tutorials help users with no prior experience with PySyft get across most important user journeys for a prospective data scientist<br>● Tutorials come with a comprehensive guide and an attached notebook, which can also be followed independently | #6871<br>#7045<br>#7120 | 5 weeks |

When choosing what to focus on within our broad GSoD project, we prioritized technical writers' interests alongside PySyft's roadmap. The technical writers were contributing 20hrs/week each and duration is estimated on part-time weeks of work. However, we did not practice hard-deadlines given that documentation artifacts required cross-collaboration between engineering and product teams with longer feedback loops. We kept a rolling window of work while PRs were pending review.

## Results

We successfully hit the milestones set for 6 months, whilst having a few last PRs pending final review and those will be merged in the coming weeks. We would love to highlight the achievements as such:

- New easy-to-follow, visually-appealing README page to guide new users: link
- Functionality to generate source code documentation: link, example
- Guide on the installation of PySyft for different OS systems: main, OSX, Linux, Windows
- Contributor guideline updated: link
- New page with resources for users: link
- Quickstart CLI tool that hides complexity of deployment and launches a ready to use tutorial series to start experimenting with PySyft: usage example
- How-to-guide series for a Data Owner: main
  - Deploying your own Domain Server: full, notebook
  - Uploading Private Data to a Domain Server: full, notebook
  - Creating User Accounts on your Domain Server: full, notebook

    - Joining a Network: full, notebook
    - Creating a Network: full
    - Configuring Privacy Budget on your Domain Server: full, notebook (*miss*)
- How-to-guides series for a Data Scientist:
    - Connect to a Domain: full, notebook
    - Searching for Datasets on the Domain: full, notebook
    - Exploring a Dataset in the Domain: full (*miss*), notebook
- User studies: In addition to all the documentation artifacts shipped, we conducted our first user study targeted to the ease of getting started using PySyft by users with no prior knowledge of our library. The subjects were asked to complete a set of key actions of the data owner persona using PySyft and by exclusively using the documentation created as part of Google Season of Docs 2022. This highlighted various points of improvement such as difficult concepts, confusing library interface, unclear explanations and underlying bugs, which we managed to fix and increase the success rate of our first how-to-guides series.

We successfully covered, after merging the existing pending PRs, most of the key results we expected during Google Summer of Docs. The remaining items, marked with *miss* above, are easy to ship given their complementary documentation has already been done.

In addition to existing work, during the internal audit and various team conversations facilitated by Google Summer of Docs, we created a long backlog of high-impact documentation issues to be solved which is an important result, including:

- How-to-guides for the data scientist persona covering:
    - model training in a distributed setting and secure predictions
    - better understanding of secure multiparty computation and differential privacy
- How-to-guides for data engineers, including various deployments (Azure, GCP, Kubernetes) and custom networking
- Follow-up user testing sessions for the rest of the documentation artifacts
- Increasing discoverability of our docs (porting the website to our domain), better linking and visibility within Github and Slack
- Easier workflow for open source contributors to documentation issues
- Implement collecting of metrics within the documentation website

## Metrics

This year was our first attempt at collecting metrics for GSoD. Metrics are collected at month end. Updates for November were unavailable at the time of this case study submission deadline. These measures were taken for observation purposes and to establish benchmarks:

- Actionable issue pull requests = 70.1% actioned which is within 15% of our assumed target of 80%.
- New contributor pull requests averaged 5.6%
    - New Contributor PRs = 64 (10.9% were Padawan mentees)
    - New PRs attributed to GSoD 2022 = 30
    - New Documentation Issues opened = 14, out of which 9 are closed & merged
- Repeat contributor pull requests form the bulk of our commits averaging 94.4%
- Internal response time from PR to Merge for small, medium PRs was not observed directly. We focused on product testing measures instead - see Tinker Time metrics below
- We averaged 19 active PRs over 1 month old. This is reflective of the scope of PR and team and resource allocation challenges.

- Count visits to error code documentation vs. error code issues opened. This measurement is less valuable to us than our new Tinker Time metrics below.
- Increased number of forks and stars of the PySyft repository →target 10%
  - Forks increase of ~3% (actual number available after end of Nov)
  - Stars increase of ~3% (actual number available after end of Nov)
- Community growth by ~3000 members representing 4% growth (actual number available after end of Nov)

Visits to Error code documentation was a suggested measure that has been more effectively captured by phased activities undertaken by Tinker Time testers (user study participants). While PySyft has received more user testing over the past few months, our documentation has evolved to a point where it significantly and positively impacts the ability of a new user to successfully complete various phases of activity.

Tinker Time metrics:

- Cumulative % Passing Installation = 77.8%
- Cumulative % Passing Domain Node launch = 85.7%
- Cumulative % Passing Data Preparation = 85.7%
- Cumulative % Passing Add DP Metadata = 46.4%
- Cumulative % Passing Data Upload = 89.2%

## Analysis

OpenMined's focus during the past few months was to develop the functionalities for our incoming release, enabling a more robust and comprehensive user journey for both data owners and data scientists. The main goal was to create user-facing docs that cover these new changes. Unfortunately the full outcome of having a great suite of docs will not be visible until our new 0.7 release. However, during our **user studies** we saw that **the artifacts done during GSoD were critical for users to accomplish key actions using PySyft**, and we believe this is the best indicator for the projects' success and a first attempt at quantifying the usability of PySyft.

Google Season of Docs was a first time opportunity for us to collaborate with a highly-committed team for documentation, which required us to quickly adapt and create processes to enable better collaboration. We will list below what went well and our challenges:

- Engagement of technical writers: Both writers were highly engaged during the whole duration of the program and incentivized by getting impactful issues to work on, aligned with the org's mission. In addition, we offered flexibility to the writers to work on items they liked and were closest to their skillset.
- Communication: while particularly challenging as the docs team spanned across Canada, Europe and India, we managed to collaborate by setting ahead a weekly 1:1 at a time which worked for each writer and a team meeting in one of the few convenient times. Regarding collaboration with the wider engineering team, we held two weekly meetings, one convenient for US/Canada folks and second one for EMEA/APAC.
- Content: to scope our documentation and understand how, what and where to invest our resources, we used the Divio framework to decide our goals and communicate it with the technical writers. When developing each guide, we framed from the beginning the target persona and the learning objectives. Additionally, we decided ahead that guides should be engaging and reduce the cognitive load through good structure and simplification of unrelated items to our learning objectives.
- Feedback loops: A challenging process in the first milestones was dealing with the slow time to gather feedback from all stakeholders (engineering, UX, product) and building consensus on a proposed design, which added frustration when the changes required after review were significant and also a deadlock for our writers.  Our solution was developing all our artifacts in three reviewed steps (detailed proposal, first iteration, second iteration) and keeping a rolling window of work to focus on while the stakeholders would provide feedback.

- User studies: Before running the user studies, we were much more confident on our documentation, but a few guides had initially a small success rate (e.g. 10%) due to small issues and bugs within the underlying framework. Being able to test the ability of users to follow independently our documentation drastically improved the quality of it.
- Changes in library interface/bugs: A challenging issue which we have not tackled yet was the frequent change of the library interface (such as method namings), which leave tutorials and guides unfunctional. As solutions exist, such as versioning for docs and QA during CI, we did not prioritize it and will address it going further.

## Summary

GSoD was exceptionally resourceful for OpenMined, as it laid down the foundation for PySyft documentation and enabled us to gain momentum, build a comprehensive roadmap for documentation going forward and a stack of processes which enables us to spin-off new documentation with little friction during design/consensus/approval. Additionally, it helped us recruit two talented technical writers which is highly difficult in a regular setup, who continue to be active OS contributors to OpenMined, and with whom we have had a great work experience.

A few key lessons we captured were:

- Documentation prioritization and structuring (Divio framework)
- Learning how to draft a proposal for an incoming documentation artifact considering its audience, main and secondary learning objectives
- A process to deliver documentation in three reviewed iterations.

We would have definitely achieved more by using these principles from the start with increased emphasis during onboarding - not only on the technical aspects of the library and mission, but also product roadmap and UX, as documentation tends to lay at the intersection of all of these.

## Appendix

Thank you to Mrinal Walia (@abhiwalia15) and Shubham Palriwala (@ShubhamPalriwala) for their amazing commitment and work during Google Season of Docs 2022. Thank you to Irina Bejan (@Irina Bejan) for her leadership of the Documentation team. We would also like to acknowledge with thanks the help of Kyoko Eng, Shubham Gupta, Madhava Jay and Andrew Trask in crafting an effective and engaging user experience through documentation via their invaluable feedback.