

Data Management Plan

SPHERE project will produce data in terms of design documents, backend and frontend code to manage SPHERE hardware resources, user documentation, promotional documentation, representative experiment environments contributed by us and by CS&P research community, research artifacts stored in SPHERE libraries, educational modules contributed to SPHERE by us and by other instructors and success metrics about the SPHERE's implementation progress and later, about SPHERE's operational use. For manageability reasons these products may be stored at different GitLab or GitHub repositories, or internally in SPHERE storage system. Most products will be accessible to general public, but some may be shared with a limited community, e.g., an artifact author may share their artifact only with the members of their own project. We thus discuss separately *public data* and *restricted data*. *To facilitate findability of all SPHERE products, we will create a parent GitLab repository for SPHERE and will include in it each product either as data or as a link to another location where the product can be accessed by either the general public or by authorized users.* Below we discuss how we will manage data of each type of the product.

We discuss separately user-produced experimental data, including data from human user studies.

Design Documents: We will produce design documents including detailed information about components comprising each device in SPHERE, the device mounting approach, power and network connectivity details, and any relationships between dependent devices (e.g., IoT nodes and their sensors and actuators). These design documents will be made publicly available on our project Web page.

Research Infrastructure Code: SPHERE project will start with the existing open-source Merge software (<https://gitlab.com/mergetb/>) and extend it to control more resource types and to support multi-facility experiments. All our software will be released as open source via our current Gitlab repository (<https://gitlab.com/mergetb/>). All Merge software components are hosted on GitLab through an Ultimate license, which GitLab provides free of charge to open source projects. This provides many useful services such as version control, issue tracking, automated build systems, continuous integration and deployment (CI/CD) routines, container repositories, and automatically generated API documentation.

Developers outside of our team (open-source contributors) will be able to contribute new features and bug fixes by creating development branches. Changes can be submitted to the main branch through “merge requests”, which are reviewed by the Merge DevOps team. Once approved, changes are merged into the main branch and undergo CI processes for building and testing.

Additionally, Merge software treats research infrastructure as code. It consumes descriptions of device's connectivity and capabilities, and deploys specific scripts to manage each device, and create network connections in experiments. These description documents will become part of our public SPHERE GitLab repository, and they will be updated each time new devices are added to SPHERE or old devices are retired.

Merge is free and open-source software, and will be released under the GNU GPLv2 license.

User and Promotional Documentation: We will develop a number of tutorials, videos and documentation showing how to use SPHERE and all its nodes and services, and advertising SPHERE to different audiences. These will be publicly accessible via our SPHERE user portals and our project page. To facilitate findability of user documentation, we will also create a parent, public GitLab repository and include all documentation there. User portals and our project page will populate their content by pulling items relevant for their functionalities from this single repository, to remain consistent. All the documentation will be released under Creative Commons license.

Developers outside of our team (open-source contributors) will be able to suggest modifications to our documentation by filing feature requests, bug reports, and by committing their contributions to a new branch and issuing a merge request, which will be reviewed by the project team.

Representative Experimentation Environments: SPHERE will host a set of representative experimentation environments (REEs), which will be seeded by us and later contributed by a broad research community. These REEs will be stored on SPHERE system, in a format that SPHERE can interpret to create new experiments or to equip existing experiments with contributed tools and datasets. Thus all REEs will be available to SPHERE users, but they will require user login. To make REEs more findable and accessible, we will support limited-time guest accounts, which members of general public will be able to use to try REEs on SPHERE. In addition to this, we will publish all REE-related documentation (e.g., description of each REE, instructions for use, use statistics, etc.) in a public GitLab repository for SPHERE, and display it on our project Web page. All REEs will be released under GNU GPLv3 license, which allows for free, non-commercial use.

Research Artifacts: SPHERE will host research artifacts that are positively evaluated in artifact evaluation efforts in per-venue research artifact libraries. These artifacts will be available to general public. SPHERE will also host user-contributed research artifacts, where the user can decide the users authorized to access it. Restricted artifacts will only be available to selected users, while public artifacts will be available to all users and to general public. Just like for REEs, to make public research artifacts more findable and accessible, we will support limited-time guest accounts, and will publish all public-artifact-related documentation in a public GitLab repository for SPHERE, and display it on our project Web page. Artifact authors will be able to choose the license under which they publish their artifact.

In addition to the artifacts intentionally created by users, SPHERE will produce logs of user actions, and it will log all IoT outputs (network traffic will be provided as files in PCAP format, video will be provided in MP4 format using H265+ codec, and audio in AAC format; both video and audio can be offered as streams or as files) at NEU IoT Lab. These logs will be stored at local datacenter storage, due to their high volume, and they will be made available to the users running the experiments for a limited time (e.g., one month) and then removed to preserve space. A user may choose to export part or all of the logged information to our persistent global storage during that time period.

Education Materials: SPHERE will host a set of education materials to be used for teaching CS&P in schools and universities. These materials will be seeded by us, and also contributed by other researchers and course instructors. They will be stored in a public GitLab repository, with teacher manuals only accessible on SPHERE to verified teachers. Materials contributed by us will be released under Creative Commons license.

Success Metrics: Our success metrics will not be made public, but will be reported to the NSF as part of our quarterly reports. We will also preserve all raw data we collect to calculate these metrics, using long-term data storage. This will also include off-site backup solutions and disaster recovery plans, which are already in place for the data stored on our testbeds.

User-produced Experimental Data: User-produced experimental data includes data generated by a user's experiment on SPHERE, including actions by human-study participants. SPHERE will not automatically save this data, but it will offer controls to users to record and store any data produced by their experiments, with optional encryption. Such data will be protected by our access control policies, and available only to the user that recorded it. It is the user's responsibility to manage this data, including enforcement of privacy protections for human-study participants.

Long-Term Data Storage: All data and software produced by this project will be preserved for at least 10 years, and all public data will remain accessible to public during this period. After 10 years we will evaluate if the data/code are still popular and if not, we will archive them to long-term storage and remove them from the server. The long term storage of relevant data (e.g., DVDs) will remain available upon request for the foreseeable future.