2. Data Security and Access Control

PECE was designed to support and promote collaborative ethnographic projects which have particular needs when it comes to data archival, security, and sharing: our data is produced through interactions with human subjects, therefore, they carry potential privacy issues that cannot be solved with automated protocols for evaluating risks of publication. It is the responsibility of researchers of a particular project hosted on PECE to discuss with their research co-participants (called “subjects” in the language of ethics committees) and make informed decisions regarding what can be shared and what should not be uploaded to the Internet. All the data we produce as ethnographers, in sum, must be carefully evaluated before it can be shared. In our legal documents – terms of service and privacy statement – we discuss in detail the responsibility PECE users and administrators have when dealing with ethnographic data and setting permissions.

Give the special needs of ethnographic data management, we designed four levels of access based on four basic user roles: ***administrator, researcher, contributor, and anonymous***.

> **- “Administrators”** are data managers with preferably with Unix system administration. Although not strictly required, it is important for administrators to read our documentation and other relevant documents for managing and securing PECE and its back-end technologies. Administrators have unrestricted access to content, users' accounts, systems configuration and permissions, and backup files. Preferably, we recommend for PECE researchers to share administrative tasks between more than one user.
>
> **- “Researchers”** are IRB (Institutional Review Board)-certified and approved individuals of a particular research PECE-hosted project.
>
> **- “Contributors”** are research co-participants, that is, users of the platform that are interested in contributing content and helping in the analytic process without having the same time commitment and responsibility for managing content researchers and administrators have.
>
> **- “Anonymous”** users do not have accounts on the system, they represent any Internet user who can access content that is made open through the public interfaces of platform.

In addition to these four basic user roles, we also have three basic permission settings for pieces of content: ***open, restricted, and private***.

> **- “Open”** content is any content distributed under a flexible copyright license – we will cover the specifics on the section of this document on “Disposition” – or accessible in the public domain.
>
> **- “Restricted”** is content that is only available to “researchers” given its potential privacy issues and anonymity requirements a co-participant might have requested when a particular piece of ethnographic data was generated. Restricted content is shared among researchers and never exposed to “contributors” or anonymous visitors of PECE.

- "Private" content is content generated by a researchers or contributors to which access is denied to users other than the creator. This permission is useful for field notes and other types of ethnographic inscription that are not yet ready to be shared with a research group or a piece of content that may contain personal information.

These three permission types can be applied to any piece of content (artifact) and every PECE interface, including the public HTML views and the API. The table below provides a schematic representation of what we just described:

Permissions	Roles	Description
Open	All	Read (Write for researchers and contributors)
Restricted	Researcher	Read and Write for researchers
Private	Researcher, Contributor	Read and Write for individual creators
All	Administrator	Read and Write (*unrestricted*)

In regard to information security, PECE relies on standard "password strength" evaluation for Drupal, which uses a simple algorithm to measure user input as ***weak, moderate, or strong*** based on three basic variables: length, usage of numbers and letters, and usage of other non-alphanumeric characters, such as the following symbols: [!@#$%*()_+]. There are more powerful ways of providing better password suggestions to the users and, therefore, increase the security of the users' accounts. This provision will be included in the next version of PECE.

For security risk mitigation, PECE comes pre-configured with a "login security" extension which blocks and notifies the administrator of potential attempts at brute-force password guessing or cracking. After 5 (five) failed log-in attempts, the user's account is blocked and the administrator is notified. The tracking time between log-in attempts is of five hours, that is, the time that is used to track between failed log-in attempts. After 20 failed attempts, the administrator is informed of a potential break-in attempt. Another feature of this extension module is the information about last time the account was used, which allows for regular users to keep track of the usage of their account and notify the admin in case of unauthorized use. Extra features include blocking a particular IP from accessing any type of content on the platform, including the user-login form. This is a convenient feature for administrators because it is easy to use and dispense with system administration skills that would be required to configure them using PECE's Unix back-end.

For system administrators running the PECE VM distribution, `drush` is the best tool for managing blocked users and hosts in the back-end:

```
# Unblocking users:
$ drush user-unblock $USERNAME

# Setting new passwords for users:
$ drush upwd $USERNAME --password="NEW_PASSWD"
```

```
# Obtain one-time-login URL for a specific user:
$ drush uli $USERNAME
```

Alternatively for log-in security, two-factor authentication can be used. It comes pre-installed with PECE, but it is not activated by default. For ethnographic projects with sensitive data, such as oral history or medical anthropology projects, it is recommended to activate two-factor authentication on the system for all users with “administrator” and “researcher” roles.

In addition to this simple permission system based on user roles and content permissions, we are planning to implement public-key encryption for our data store in the next version of the platform. For PECE 2.0 (described in the Appendix), we will improve “password strength checking” by verifying randomness of the user's input in the password text-box. We will also implement RSA 4096-bit public key encryption, extend users' profiles for the storage of public keys, and offer PECE administrators a virtual server image of PECE/Drupal with support for an encrypted filesystem with additional encryption of the “`files/restricted`” system directory. For PECE 1.0, data encryption is only supported for backups (more information on the section on “Data Backups” of this document).

Administrators installing the platform for the first time are required to configure HTTP Secure (with SSL/TLS, Secure Socks Layer/Transport Layer Security). It is important to use HTTPS to mitigate security risks given the importance of protecting the communication between users and PECE web services, primarily when posting passwords and posting/retrieving sensitive information but also to ensure that all content is posted and loaded over HTTPS. We recommend using the software and the general guidelines of the project Let's Encrypt at `https://letsencrypt.org` in order to configure HTTPS for your PECE instance.

Instructions for Developers: Creation of 2 basic roles on the system, “researcher” and “contributor”. Creation of three types of content permission: open (for all – all the files are saved on `files` public directory), restricted (only open for researchers, all the files are saved on the `files/private` directory), and private (only accessible by the person who created the content, files are stored on the private directory with `umask 077`). Installation of **“login_security”** v. 7.x-1.9 and **“tfa”** v. 7.x-2.0beta2 module is required. The configuration for login_security is as follows: [tracking time: 5 hours; time before blocking user: 5 hours; time before soft blocking a host; failed attempts before blocking a host: 10 hours; failed attempts before detecting an attack: 20; notifications: display last login timestamp; display last access timestamp; user who should be notified when an account is blocked or ongoing attack is detected: admin].