Research Data Alliance

PRACTICAL POLICIES RECOMMENDATIONS FOR DATA MANAGEMENT

September 22, 2015
(draft v0.9.1)

PECE

Platform for Experimental Collaborative Ethnography

Table of Contents

INTRODUCTION

What is the Platform for Experimental, Collaborative Ethnography (PECE)?

The Platform for Experimental, Collaborative Ethnography (PECE) is a Free and Open Source (Drupal-based) digital platform designed to support distributed, collaborative hermenuetics research with diverse types of research data. While especially designed to support experimental ethnography projects, it provides a general model for the digital humanities, and particularly for what we have termed the empirical digital humanities (including work in history, anthropology, folklore, and other fields that collect and analyze primary data, using hermeneutic or interpretive techniques). PECE provides a place to archive and share primary data generated by empirical humanities scholars, facilitates analytic collaboration (within the humanities and with researchers in other fields), and encourages experimentation with diverse modes of publication. The platform allows humanities scholars to experiment with digitally-mediated, interdisciplinary collaboration, provides opportunities to involve students in humanities research as it progresses, and quickens the public availability of humanities research in an open access form. PECE also enables experimentation with new forms of peer review for humanities research, and functions as a portal to a suite of web tools useful for humanities research, including tools developed in data science for other scientific communities. PECE is at the center of an on-going research project that explores how digital infrastructure can be designed to support collaborative hermeneutics.

PECE has been built and is governed by an interdisciplinary design group centered at Rensselaer Polytechnic Institute (Troy, New York, USA), including Michael Fortun, Kim Fortun, Lindsay Poirier, Dominic DiFranzo, Luis Felipe Murillo, Brandon Costelloe-Kuehn, Alli Morgan, Brian Callahan, and Rodolfo Hernandez. On-going software development is being conducted on Github under the account "PECE-project.” The release of the version 1.0 is planned for December 2015.

What is the Research Data Alliance (RDA)?

The Research Data Alliance is an international initiative to facilitate the development of effective data practices, standards and infrastructure in particular research areas, and across research areas – aiming to enhance capacity to archive, preserve, analyze and share data, and for collaboration both within and across research communities. Within the RDA, Working Groups (WGs) are formed to advance recommendations for best (or at least shared) practice relevant to the building of robust, sustainable research infrastructure. In 2014, RDA’s Practical Policies WG released recommendations for data processing, stewardship, and overall data management for repositories. The focus is on computer actionable, automated policies to be built into repository platforms. The Practical Policies WG was motivated by concern to increase trust among data producers, users and funders (thus increasing investment in data sharing, preservation and re-use), and capacity for reproducibility in scientific research. Data federation initiatives such as EUDAT and the DATANET Foundation

Consortium (USA) helped define the WG's policy recommendations, and are expected adopters. The recommend policies are in 11 policy areas: 1) contextual metadata extraction; 2) data access control; 3) data backup; 4) data format control; 5) data retention; 6) data disposition; 7) integrity (including replication); 8) notification; 9) restricted searching; 10) storage cost reports; and 11) use agreements.

PECE is a much smaller enterprise than initiatives like DATANET and EUDAT, and functions not only as a repository but also as a collaborative work space and publisher of diverse research products (digital exhibits, for example). PECE is also customized to support humanities research. PECE's adoption of the WG on Practical Policies recommendations thus assessing and working on their implementation to the humanities. The adoption process has thus been iterative, tracking between effort to implement the WG's policies, and effort to appropriately characterize data management needs and expectations special to the humanities.

Humanities researchers work with an array of data types – many unstructured – that are less common or foundational in other research fields, such as in-depth ethnographic interviews. Researchers in the empirical humanities also have special access control requirements given the sensitivity and legal regulation of the "human subjects" data they collect and work with; the context-specific judgments often required to determine appropriate access often can't be automated. Further, reproducibility of research results is not always a goal or means to validation in humanities research; collaboration is thus motivated by (and undercut) by different concerns.

Practical Policies for Data Management

This document describes the data management practical policies that were designed for PECE. Since February 2015, the PECE design group has been working to implement the recommendations of the Research Data Alliance's (RDA) "Practical Policies" Working Group (WG-PP). Our goal with this document is to describe how PECE implements best-practices in data management to meet the special needs of empirical humanities researchers. The work described here has been developed with the support from RDA.

PECE Data Management Definition

We designed and implemented a set of practical policies for data management per recommendation of the RDA's WG-PP and the National Science Foundation's Data Management Plan (DMP) of January, 18th, 2011. Qualitative research data management in PECE encompasses four inter-related domains for human and computer-actionable policies: *preservation*, *disposition*, *privacy*, and *collaboration*. During the design phase, we aimed for a balance between the necessity of preserving privacy and anonymity and the need for creating conditions for data sharing and collaborative analysis among ethnographers, historians, and co-participants of our projects. In practical terms, we implemented and extended features of the Drupal framework to ensure proper data management. This a preliminary version of our

report for circulation among RDA members to gather feedback before releasing version 1.0.

In sum, there are four constitutive dimensions for data management in PECE:

> **Archival:** Defining a data model is the first step for the organization of digital ethnographic data and metadata for future storage in data repositories. We do not aim to substitute robust, big data repository solutions with PECE but first and foremost to help in organizing ethnographic collections for long-term archival. In order to contribute to the task of ethnographic data preservation, we describe data types with rich meta-data and linked data mappings (which allows for better discoverability, replication, and, what is key for ethnographers and historians, the capacity to ask relevant research questions across several ethnographic collections). In practical terms, PECE offers a way to organize, analyze, and replicate digital ethnographic collections that is useful not only for ethnographers but for other professionals working on issues of digital preservation of scholarly archives. We included in the PECE metadata description important fields to account for provenance (which, in the context of ethnographic projects, has to do with information about the project, field sites, researchers, and methodological and theoretical orientations) in addition to fields for contributors, licensing, shared tags, and permissions.
>
> **Openness:** For the purpose of data preservation, we took one step further in organizing our data for archival, collaborative analysis, and sharing by creating open (as in public) interfaces for data science experts to harvest PECE Creative Commons-licensed and public domain data. We also have specified methods for manipulating data and triggering actions on the platform according to their status (such as the capability of deleting files when they reach their expiration dates).
>
> **Privacy:** Ethnographic projects are based fundamentally on engagement between researchers and research participants for the interpretation of sociocultural processes. Out of the experience of engagement, a myriad of privacy and ethical issues are potentially involved. Various types of content (from participant observation or interviews, for instance) cannot be shared publicly due its sensitive, and potentially privacy-infringing, information. Having the commitment to preserve our research co-participants privacy, the PECE design team designed the platform around the need to flag certain types of content as private or restricted from public view. We have also specified a workflow for helping researchers define what are the permissions that are necessary for certain types of content, including types of data that contributors should refrain from uploading to the platform. In addition to the PECE permission system, we are planning to implement public-key encryption for our data store in the next version of the platform.
>
> **Collaboration:** By leveraging Free and Open Source-based web technologies, open standards, and open data with semantic extensions to support ethnographic projects, PECE aims to help advance modes of collaborative

inquiry. For this purpose, data management involves supporting and enforcing the usage of open formats, flexible copyright licenses, and web standards to facilitate present and future collaborative endeavors. The PECE Design Team made the choice of running the platform on an established web framework, Drupal, in order to foster collaboration on many levels: Drupal's community size and geographic distribution which spans across East Asia, Western and Eastern Europe and the Americas; its development community constituted of companies (big and small), local community chapters and conferences, large international conferences, as well as numerous book publications and web resources with rich documentation for all levels of technical skill. Several companies, community projects, and news outlets (with big and small datasets) run on Drupal with millions of articles, creating a vibrant community around Drupal which cooperates to develop public, common resources by sharing code and documentation. For PECE's sustainability in particular, this collaborative dimension is vital. We decided to rely and contribute to upstream Drupal development and help with contributed modules by testing, reporting, and fixing bugs. We are contributing, in specific, a set of tools for multimedia annotation that will be of great value for the academic community and the public.

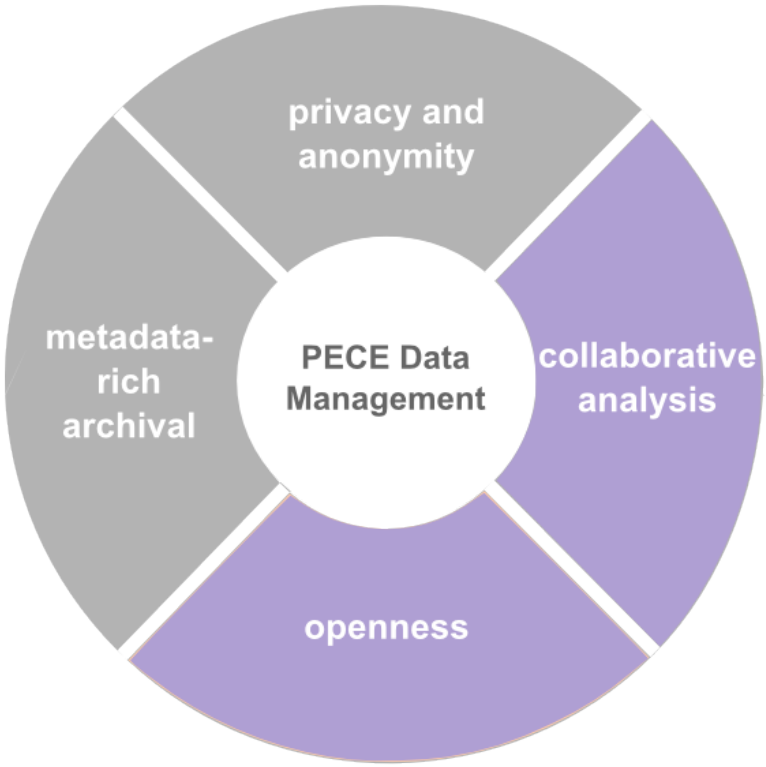


Figure 1. PECE data management basic fields

Next, we will discuss the design and implementation of practical policies for data management on the following topics: contextual meta-data extraction, data control, backup,

format control, retention, disposition, integrity and replication, notification, restricted searching, instance cost reports, user agreements (including privacy statement and users' code of conduct). In appendix, we attached a copy of the current PECE data model and the configuration files that were described for the practical policies.

DATA MANAGEMENT PRACTICAL POLICIES

1. Contextual Metadata Extraction

According to the RDA's definition, the process of “contextual metadata extraction” consists in “creating metadata associated with files and collections[1].” In order to offer this functionality to our users, PECE installations come by default with a pre-configured Open API which allows for indexing, creating, updating, and deleting data, metadata, and provenance information of digital objects.

Our Open API is composed of two servers: 1) a REST server with JSON and XML endpoints (URLs from which to interface with PECE) and 2) a SPARQL endpoint which can be used to query RDF datasets, leveraging semantic web extensions in PECE.

Our REST service is divided into two basic endpoints: one for anonymous, non-authenticated users access to public content and one for authenticated users with access to restricted content. The public endpoint has three resources: *files, nodes, and artifacts.* The restricted endpoint includes the three resources plus another one for handling user information: *users*. In the restricted endpoint, three resources (*files, nodes, users*) support the following functions: Retrieve (GET), Create (POST), Update (PUT) and Delete (DELETE).

The resource *artifacts* has a filtered output for consumption by web systems other than Drupal, therefore it only accepts (GET) requests. It contains all the data and metadata fields for each artifact archived in your PECE instance and can be accessed by displaying the name of the resource the user wants to consume: *image_artifacts, video_artifacts, audio_artifacts, text_artifacts, webpage_artifacts, bundle_artifacts*.

Here is the table with the relation of supported functions for each endpoint and resource:

RESOURCE	ENDPOINTS							
	api/public				api/auth			
	GET	POST	PUT	DELETE	GET	POST	PUT	DELETE
Files	X				X	X	X	X
Nodes	X				X	X	X	X
Users					X	X	X	X
Artifacts	X				X			

Public and CC-licensed content can be accessed through the “public” and “auth” endpoints, whereas restricted content can only be accessed through “auth” (private) endpoint. Authorized users and administrators have much more flexibility to create, modify, and delete content for which they have permission. This flexibility includes the ability to manipulate user accounts and content in batches.

1 Source: http://smw-rda.esc.rzg.mpg.de/index.php/Contextual_metadata_extraction

We will describe below how to obtain structured and serialized data from the public interface, then we will describe how to use the Open API to modify digital objects, which is extremely useful for the purposes of data migration (and syncing across data repositories and web applications).

PECE Open API can be accessed through the following URLs (changing the portion with your respective domain name):

```
// For public, anonymous users:
https://your-domain.org/api/public/files
https://your-domain.org/api/public/nodes
https://your-domain.org/api/public/image_artifact, video_artifact, and so on.

//For authenticated users:
https://your-domain.org/api/auth/files
https://your-domain.org/api/auth/nodes
https://your-domain.org/api/auth/nodes
https://your-domain.org/api/auth/image_artifact, video_artifact, and so on.
```

Responses can be formatted either in XML (Extensible Markup Language) and JSON (JavaScript Object Notation), “application/xml” (default) and “application/json” respectively.

Suppose a user wants to request machine-readable data and metadata from your PECE instance. The following command would return a JSON document with all the data and metadata fields for a particular node whereas “nid” is the “Node Identifier Number”:

```
$ curl -X GET https://your-domain.org/api/public/nodes/nid.json
```

The following output would be the result, exposing data and metadata for the requested node:

```
{
    "changed": "1439121431",
    "comment": "1",
    "comment_count": "0",
    "created": "1439121000",
    "field_collaborators": [],
    "field_critical_commentary":,
    "field_group_audience":,
    "field_format":
    "field_image_annotation": [],
    "field_licensecc":,
    "field_location":,
    [...]
}
```

To render the previous output in XML, the syntax would be the similar, except that the termination (.json) would have to be modified (or omitted) as in the example below:

```
$ curl -X GET https://your-domain.org/api/public/nodes/nid.xml
```

If the user wants to retrieve index lists of nodes or files, you just have to omit the last portion of the URL with “Node ID”. Please note that the GET function only lists 20 items by default. If you need to retrieve more (or less) items, it necessary to pass a parameter in the URL.

For the purposes of interoperability with other web frameworks and data repositories, we created filtered XML and JSON outputs for each PECE content type (with permissions fields to render data publicly accessible or not). Filtered outputs were specified to be both machine-readable and comprehensible by humans. In order to obtain, for instance, a listing of “image artifacts,” the following commands could be executed:

```
// For the complete listing in XML:
$ curl -X GET https://your-domain.org/api/public/image_artifacts

// For the complete listing in JSON:
$ curl -X GET -H "Accept: application/json" \
       https://your-domain.org/api/public/image_artifacts
```

The filtered output follows the convention of the PECE Data Model (in appendix). Consult this document to understand the data types and the relationships between fields:


```
{
    "URI": "F3EA8139A6B43ECBC56BB7CF51E51",
    "Title": "Orion Nebula",
    "Date of Creation": "1439121000",
    "Revision Number": "23",
    "Author": "John Public",
    "Collaborators": "Alice S.",
    "Format": "JPEG",
    "Project": {
        "Name": "Minority Astronomers Multi-Disciplinary Collaborations",
        "Description": "This project investigates how women scientists engaged in
        collaborative, multidisciplinary research build relationships and the
        effects of these relationships on their careers [...]",
        "Members": "Bob M., Alice S., John Public, Mary B.",
        "Funding Agency": "NSF EAGER"
    },
    "Fieldsites": "Astroinformatics",
    "Annotations": [],
    "Commentary": "Image captured by the \"ACS\". According to the Hubblesite,
                  more than 3,000 stars of various sizes appear in this image.",
    "License": "//creativecommons.org/licenses/by/3.0/",
    "Tags": "NASA, Hubble, astroinformatics, Creative Commons",
    "Image URL": "https://astroanthro.net/public/nebula.jpg",
    "Location": {
        "lat": "20",
        "lat_cos": "0.93969262078591",
        "lat_sin": "0.34202014332567",
        "lng": "-20",
        "lng_rad": "-0.34906585039887"
    },
    "Group audience": "NSF/EAGER Astroinformatics research group"
    [...]
```

In the example above, we have information on a particular artifact with provenance fields such as “project” and “fieldsite” as relational information about the field in which the data was produced by a team of ethnographers – plus other fields, such as “group audience,” “collaborators,” (which lists ethnographers who contributed content, but are not the “authors” of a particular piece of data) and “annotation” (which lists all the annotations that were generated by one or multiple users).

For complete data manipulation capabilities through the “auth” endpoint, it is necessary to have an account in the platform (as well as permission to manipulate the content you are requesting). If you are a registered PECE user identified with a “researcher” role, you are granted control over the content you generated, including the possibility to create, modify, retrieve, and delete content or specific fields of particular types of content.

Administrators are the recommended users to perform most tasks through the “auth” endpoint. For security purposes, we can restrict access to the “auth” endpoint only to users or disable it entirely (or grant access to it only to certain machines, see the section on PECE Security for further information on access control).

Let's suppose that, at some point, the necessity to update a particular field has appeared in a hypothetical project. It became necessary for a member of the research team to change the “critical commentary” to include further critical evaluation of a particular artifact. This command would accomplish this task by changing content of the field “critical commentary” with the text “New Kritik”:

```
$ curl -X PUT -H "Content-Type: application/json" \
       -H "Cookie: EXAMPLE_SESS02caabc123=ShBy6ue5TTabcdefg" \
       -H "X-CSRF-Token: EXAMPLE_w98sdb9udjiskdjs" \
       -H "Accept: application/json" \
       -d '{"nid":"18", "field_critical_commentary":"New Kritik"}' \
       https://your-domain.org/api/auth/nodes/18
```

As in the example above, there many parameters to pass to curl when creating, deleting, or modifying a node, file, or user on the plaftform. First, it is necessary to log-in through the “users” resource:

```
$ curl -X POST -H "Content-Type: application/json" \
       https://astroanthro.net/api/auth/users/login.json \
       -d '{"username":"user","password":"password"}' \
       -c session.txt
```

Since we are using the restricted “auth” endpoint, please observe that it fundamental to collect and then pass the information about your X-CSRF (cross-site request forgery) token and session information (“cookie”) as header parameters in every subsequent request. This can be accomplished in many ways. For instance, the user can save it to a text file with the -c parameter with curl then execute every POST or PUT request passing the -b parameter plus the name of the file you created:

```
$ curl -X GET -H "Content-Type: application/json" \
       https://your-domain.org/api/auth/users/nid.json \
       -b session.txt
```

The command above would provide the information on a particular user. A similar syntax applies for requesting other types of data. Please observe that it is necessary to pass the parameter of Node ID (“nid”) or User ID (“uid”) if you are accessing, modifying, or deleting a resource. The request must also include the body data (which is identified by the machine name of the field you want to modify – please consult the document “PECE Data Model” for the description of mappings from “field_machine_name” to “field name”).

There are many benefits in using the Open API for administrative tasks. It is possible to perform tasks in bulk, modifying large swaths of data in batches. It is also useful to modify punctually and quickly any type of data, including artifacts, files, and users. For the purposes of promoting Open Data exchange and Open Access among ethnographers and historians more generally, our API allows for automated tasks of contextual metadata extraction as well as harvesting. The technical details regarding Open Data exchange are further discussed under the section “Disposition” of this document.

2. Data Security and Access Control

Data access control policies specify who has access and what type of access is granted for any data objects of a digital collection. In this respect, PECE was designed to support and promote collaborative ethnographic projects which have particular needs when it comes to data archiving, security, and sharing: our data is produced through interactions with human subjects, and therefore, carry potential privacy issues that cannot be solved with automated protocols for assessing risks of publication. It is the responsibility of PECE researchers of a particular project to discuss with their research co-participants (called “subjects” in the language of ethics committees) and make informed decisions regarding what can be shared publicly, what can be shared privately with other PECE users, and what should not be uploaded to the Internet at all. Broadly speaking, all the data we produce as ethnographers must be carefully evaluated before it can be shared in the context of a research collaboration or the Internet. 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.

Given 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 preferably with Unix system administration skills. 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 often 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 authorization to access restricted content. They do not have 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. Content that is released in public domain is also categorized as open.

- **Restricted** is content that is only accessible 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.

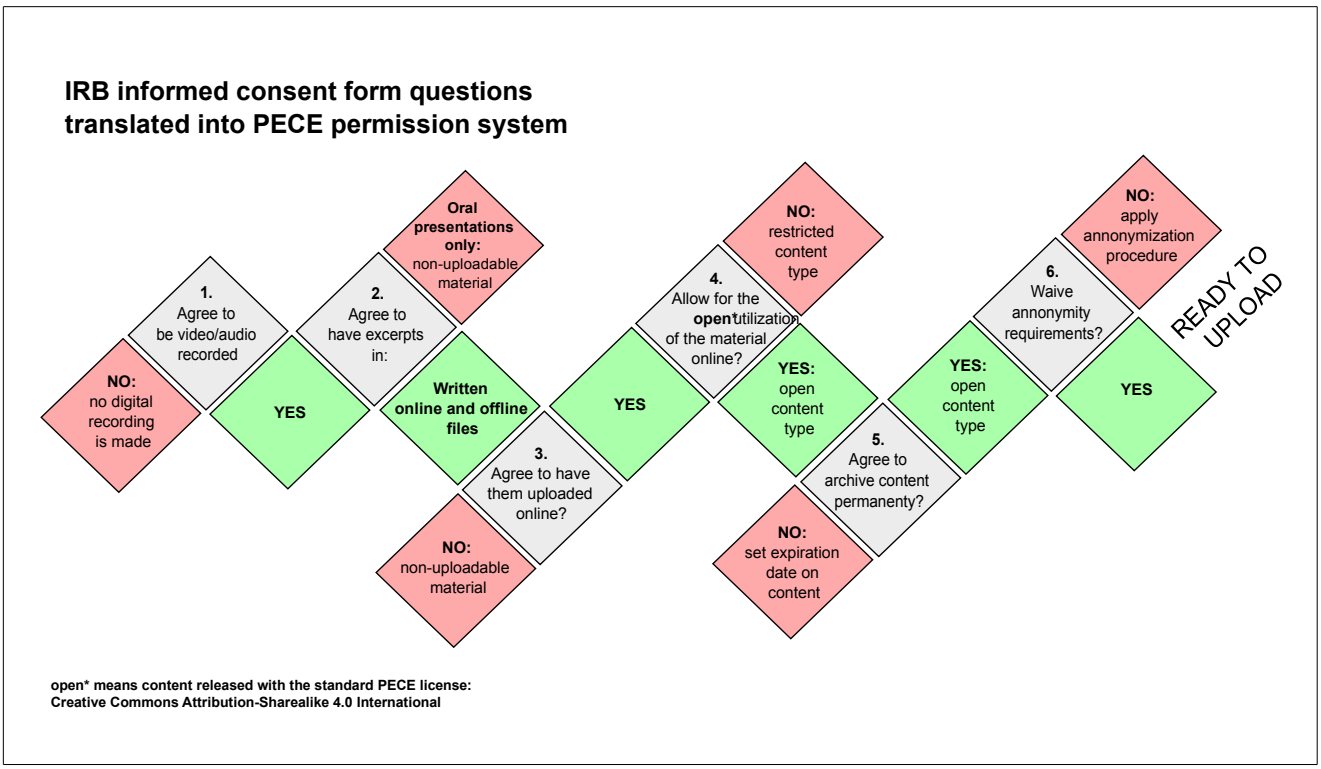
- **Private** content is content generated by researchers or contributors. Only the content creator can access "private" content. This permission is useful for managing access to field notes and other types of ethnographic inscription that are not ready to be shared publicly or with the research group.

These three types of permission can be applied to any piece of content (*artifact*). 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)

The PECE permission system was designed to translate specific access restrictions and expectations (often encoded in IRB-approved consent forms) that are required of ethnographic projects. Translation is performed by identifying the type of permission that is necessary based on a set of questions that are presented to the subject in the consent form. The fluxogram below demonstrates how to identify and translate from specific answers to privacy and anonymity questions into the permissions system.

For cases of extreme sensitivity involving potential damage to research subjects, we advise PECE users to refrain from uploading content to the Internet. PECE cannot secure data beyond normal security expectations of state-of-the-art web technologies. That is, PECE cannot be guarantee nor protect privacy when secure storage and data encryption are not used, despite the effort our design team has made to enforce the usage of strong passwords and data encryption of our backups that are stored in different machines for redundancy. Issues of data security will be further discussed under the section on "Data Control". For now, it is important to observe the need for using the category of "non-uploadable material" when applicable to sensitive data as described in the graph below:



For information security in Drupal, PECE relies on standard “password strength” evaluation which uses a simple algorithm to evaluate user's 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 strength assessment to the users and, therefore, increase the security of their accounts. This improvement 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 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. Another feature of this extension module is the information about the 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 security 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. PECE will also support RSA 4096-bit public key encryption, extend users' profiles for the storage of public keys, and offer 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 vital importance of protecting the communication between users and web services, primarily when posting passwords and posting/retrieving sensitive information as well as to ensure that all content is transported 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 a PECE instance.

3. Data Backup

Regular and redundant data backup is a vital necessity of every digital information system. When defining a backup solution for PECE, we followed the general guidelines of the Drupal community and the RDA practical policies for data management. In a nutshell, the overall goal of our backup policy is to ensure PECE instances have, at all times, three backup copies in distinct machines.

The first backup level is the PECE backup, which is performed automatically on a regular basis by the Drupal framework. **The second level** is, generally, performed by the hosting company or data repository which must provide regular, automated backups on the system level, that is, generating regular snapshots of a virtual machine where PECE is running. This is beyond the reach of automation of our platform and has to be set-up with the hosting company directly. We describe the technical requirements of PECE backups for hosting companies in the section on “Instance Hosting Costs” of this document. **The third and last but not least important form of redundant backup** is to generate an offline copy of PECE in a safe environment[2] (in addition to the other two forms of automated online backup).

The third form of regular backups is generated through the extension “Backup and Migrate” which performs full backup of the database and the PECE directory structure on the file system. The generated tarball file is useful for quickly restoring the system in case of data or system failure. The backup functionally provides full Integration with drush (Drupal Shell) for facilitating the administrative tasks of more experienced system admins as well as a GUI for new PECE administrators who are not used to command-line interfaces. For users of the PECE VM distribution, we provide both options out-of-the-box.

Given the key importance and sensitivity of this data management task, only administrators (users with the “administrator” role on the system) are allowed by default to generate and access backup files and system configurations. Administrative backup functions include:

- Database backup;
- File system backup;
- AES 256 encryption of backup files;
- Export and import previously generated backup files;
- Setup backup schedules (to run on cron jobs);
- Setup $PATH for backup files;
- Usage sftp to send backup files to other machines.

Backups are generated with timestamp, AES encryption (given the sensitivity of the data they include in .tar.gz files) and then replicated to a different machine. Two options, thus, are offered to PECE administrators: to either use the GUI or the command-line interface (both offering automated backup solutions). Command-line tools facilitate the process of

2 "Safe environment" in this context means the usage of an encrypted disk or partition to store sensitive research data, including documents with sensitive information (such as co-participants personal information, signatures, etc.). In GNU/Linux systems, we recommend the usage of dm-crypt which is the default solution provided by the kernel.

automation.

```
# Perform a new backup using PECE's backup profile
$ drush bam-backup pece_bkp

# Lists all the backups already generated (outputs backup ID numbers)
$ drush bam-backups

# Restore a particular backup, using its ID number
$ drush bam-restore $BACKUP_ID
```

These commands are based on drush to generate, list, and restore backups. Shell scripts can additionally be used, added as cron job, to 1) put the server in maintenance mode for backup purposes; 2) dump the contents of the database to a file; 3) generate a tarball of the Drupal directory structure; 4) assemble the DB dump and the tarball into another .tar.gz file; 5) use AES 256 to encrypt the package file; and 6) finally, upload the encrypted file to a different server via sftp (or, alternatively, synced with rsync). The option of scheduling and running a shell script for automated backup will be shipped with the PECE distribution, thus offering an alternative for experienced system administrators running off of the PECE VM distribution or their own server infrastructure.

In order to respect the state of each and every artifact with respect to their permissions, automated backups are generated as a snapshot, that is, older versions are not maintained so as to avoid keeping old copies of content that has already expired or had its permissions changed.

4. Data Format Control

Data format control is a data management policy which describes what tasks must be performed with ingested files in order to enforce file format restrictions. System-level control over data formats is crucial for PECE's Open Knowledge mission which comprises clear guidelines for generating, archiving, analyzing, and distributing Free and Open Source Software, Open Data, and Open Access publications. Data format control, for this reason, was considered on PECE's design for increased data accessibility, usability, and interoperability among heterogeneous information systems.

In respect to its general guidelines for data format control and improved accessibility, PECE follows the Open Knowledge Foundation's Open Data definition observing three general principles for design and implementation of PECE's data management policies: 1) data must be discoverable and indexable through the web; 2) if the data is not machine-readable and distributed in an open format, it is not reusable; 3) open data must not not have legal restrictions for its usage, repurposing, and redistribution. For the purposes of data management, the PECE design team has adopted the OKF definition of "Open Knowledge" in working with the ethnographic data produced: "Open knowledge is what open data becomes when it's useful, usable and used" in the context of ethnographic projects.

In terms of technical specification, we described and implemented restrictions for content types and file formats that can be uploaded to the platform. The following table describes all the content types and the formats we use:

Content Type	Format	Extension	Commentary
Text	Hypertext Markup Language, Open Document Format, JavaScript Object Notation, Extensible Markup Language, JavaScript Object Notation for Linked Data, Resource Description Framework (UTF-8 encoded)	HTML, XML, JSON, JSON-LD, RDF, ODT, ODF	Serialized exchange file formats are delivered through the PECE Open API
Audio	OGG Vorbis, Opus, Advanced Audio Coding (Low Complexity), MPEG-1 Part 3, Microsoft WAVE Format 1	OGG, MP4, M4A, MP3, AAC, WAV (containers)	MPEG1 Part 3 (MP3), AAC, and WAV are proprietary technologies
Video	Theora, VP8, VP9, MPEG-4 Part 10 AVC (H.264)	OGG, OGV, WEBM, MPEG4, MP4 (containers)	MPEG4 Part 1 AVC and its MP4 container are proprietary technologies
Image	Joint Photographic Experts Group, Graphics Interchange Format, Portable Network Graphics, Scalable Vector Graphics	JPG, JPEG, GIF, SVG, PNG	
PDF document	Portable Document Format	PDF	

As the table demonstrates, we made an effort to adopt only “Web safe" and Open Document formats and standards. In doing so, we followed the guidelines of the W3C HTML5 standardization committee. There are, however, a few important exceptions to our Open format policy given the adoption of proprietary technologies (for containers and codecs of media files) as part of the W3C HTML5 specification. This is rather unfortunate given the state of dependency on proprietary video and audio technologies for the web. These exceptions include the adoption by the HTML5 video and audio tags with MPEG-4 part 10 AVC, as noted on the table above.

In terms of the actual implementation, data format control is executed at the interface level on PECE; that is, it is executed for data upload, presentation, and download. Through the web interface only permitted formats are allowed to be uploaded. The user is presented with an error message when trying to upload a file that is not compliant with our Open format policy. After uploading a permitted file, we will use native support from web browsers that respect Open standards and formats (such as Mozilla Firefox, Chrome, Chromium, and Opera) to decode and render files on the browser (for all the supported formats: audio, video, texts, PDF documents, and images). For data harvesting purposes or for bulk operations, our Open API (as specified in the first section of this document on “Contextual Metadata Extraction”) operates with web standards for communication, authentication, and data manipulation and exchange (with JSON and XML formats).

In the roadmap for PECE 2.0 is the automatic, back-end transcoding of file formats: from proprietary and closed to open formats. We are testing and planning to implement audio and video transcoding capabilities on the platform as well as to offer automatic conversion of proprietary formats such as Microsoft Office Open XML to Open Document Formats, given their wider compatibility and sustained efforts to create interoperable, open, and community-driven formats.

5. Data Retention

Data retention policies for data management specify the operations the system must execute for the purposes of evaluating data objects in respect to their expiration dates and embargo periods. Ethnographic projects, however, tend not to have “embargo periods” and ethnographic data tends not to have “expiration dates” whereas both are common for digital data management in science and engineering disciplines. There are particular reasons that account for this difference. First, ethnographers tend not to share “raw data” but drafts of partial and preliminary analyses with other ethnographers and other research groups. The very concept of “raw data” is quite foreign to most contemporary ethnographic projects since data only acquires meaning in the context of a particular ethnographic project. To put in different terms, data must refer to what we call “conditions of production” to acquire particular meaning and become useful for ethnographic or historiographic purposes. Ethnographic data is data generated in the context of human relationships in general and forms of human and non-human interaction in particular. Without information on these basic foundations of data production, ethnographic research data is not useful and not usable by other researchers. Lastly, the reason why expiration dates are not common for ethnographic data is because ethnographic data represent documents of, not only anthropological and sociological interest, but of historical importance in many cases. They can be used for building archives and for comparative efforts at any point in the future as long as they are properly stored, extensively described, and made available through flexible licensing schema and interoperable data management systems with open, public interfaces.

In the course of specifying and implementing PECE 1.0, we made design decisions with the goal of questioning and changing the current understanding and usage of data retention policies. The aim was to pose the trade-off between data protection and openness under a different framework with a focus on Open Source technologies, Open standards, and Open Data. Instead of focusing on data protection against competition in the sciences for priority of publication, which tends to be the current norm and practice in the sciences, we channeled our efforts onto the task of creating infrastructures to foster collaborative ties in which data are contributed to a common pool – from which many researchers and related disciplines can draw. PECE, in this sense, aims first and foremost to be a contribution to a digital commons for the humanities and social sciences. Therefore, the current notion of “data retention” is not particularly useful nor central to our mission. There are, however, very important exceptions in which “data retention” should be used in observance of ethical guidelines and privacy issues.

Ethical guidelines and privacy issues (such as the ones we described in the sections on “Disposition” and “User Agreements” of this document) are key topics of debate and concern in respect to retention periods as ethnographic data is meant to be kept secure and private given potential privacy concerns or expressed intent of research subjects. “Retention periods” for ethnographic projects, therefore, are usually established around the sensibilities of our co-participants, observance of their rights to privacy and anonymity and, ultimately, the needs of a particular project to protect, analyze, and then delete a particular piece of data under the request of a research co-participant.

In respect to its technical affordances, PECE provides its users with the ability to identify sensitive pieces of datum and change its status after a certain period of time (from published

to unpublished, for instance) and for certain functions to be performed (such as deleting a file or artifact after a certain period). This is important for the ethical and privacy concerns we mentioned above, but, particularly to remind our users that certain pieces of data must be deleted after the project is over. Compliance with requests for deletion of data can be accomplished on PECE by setting up a “timer” on PECE artifacts. Under “Publishing Options” for every artifact, the user has the option of setting up an expiration date at the time of submission in the following format: YEAR-MM-DD (year-month-day).

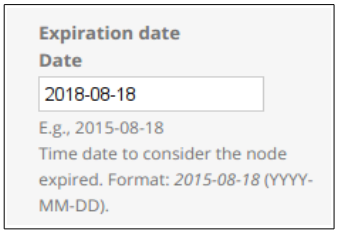


Figure 3. Setting up the expiration date for an artifact

Alternatively, deleting artifacts per requirement of research co-participants can be performed in batches. It is necessary, first, to collect the “Node ID#” of every exception and save it into an unordered list, such as [1. 3. 10. 49. 321. 5423. 43, etc.]. Then, a simple shell script can be used to remove ethnographic data that was requested to be deleted:

```
#!/bin/sh
# Declare the array with the nodes that were requested to be deleted
array = (Node IDs # i.e. 1 2 3 4)

# Iterate over the array items and delete one-by-one from PECE
for i in "${array[@]}"
do
    drush node_delete $i
done
```

There are ways to collect Node IDs with specific expiration dates by executing a query on the PECE database. This can be done using drush and Drupal “Entity API” with the following command:

```
# Query for nodes with expiration dates, saving the output to a file:
$ drush php-script expired_nodes.php > expired_node_ids.txt

# 'expired_nodes.php'
<?php
$now = new DateTime(); // time when the query was executed
$query = new EntityFieldQuery(); // make usage of Entity API
$query
   ->entityCondition('entity_type', 'node')
   ->fieldCondition('field_expirationdate', 'value',
       $now->format('Y-m-d'), '<')
   ->addMetaData('account', user_load(1));

$result = $query->execute();
drush_print_r($result); // terminal output as an example
```

?>

It is part of our roadmap to create an automated way of marking and deleting “private” content with “expiration dates” for PECE 2.0. The improvement of this data management policy will include the identification of sensitive data through tagging, regular, scheduled scanning across the dataset for sensitive, private content, and systematic deletion of data upon completion of a research project as specified on the “Project” section on the platform.

6. Disposition

According to the Research Data Alliance's workgroup on “practical policies” for data management (RDA WG-PP) “disposition” policies are triggered at every event in which a retention period has been reached to delete or archive a digital object. For the needs of the PECE project in particular, “disposition” represents the need for organizing information in a way that allows for ethnographic data to be readily available for sharing across platforms and research groups in the humanities and social sciences.

There are two specific approaches to disposition which encompass both the general orientation of the RDA WG-PP and the specific needs of the PECE project: 1) make it simple and straightforward for users to use flexible copyright content in their pieces of data; and 2) to trigger a disposition policy when an expiration period has been reached (as described in the section on “Data Retention” of this document).

The first approach consists in attributing by default a Creative Commons (CC) license with injunctions for authorship attribution and redistribution under the same license as well as provisions for portability of the license in its version 4.0 (that is, the usage of the International version of the license that is useful for data that travels across national jurisdictions). The information on the CC license is included as metadata for every digital object of the platform by default and displayed as a small logo on the platform, so users can have convenient access to the text of the license:

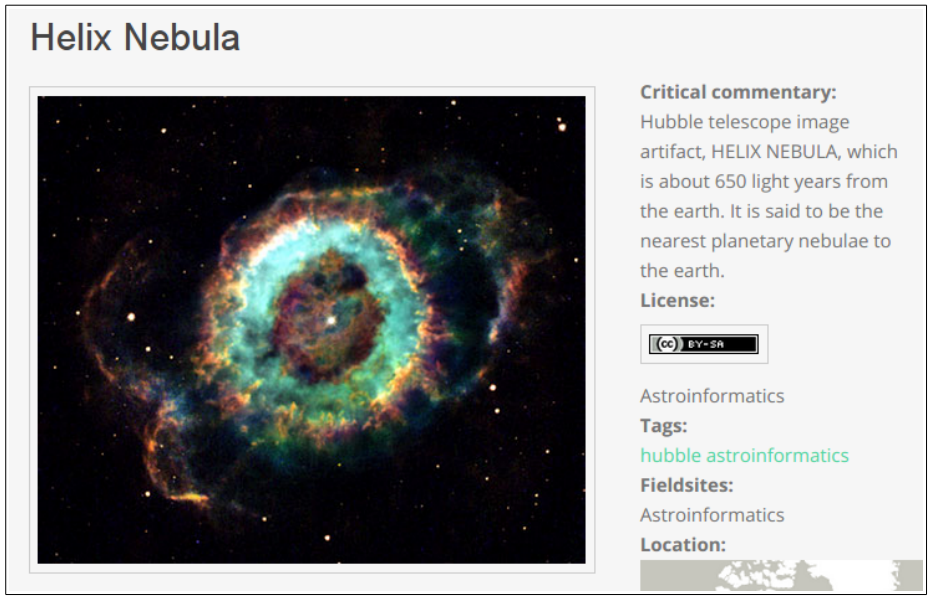


Figure 4. Display of licensing information for an artifact

The metadata for the artifact, which be can obtained via PECE Open API, also describes its “disposition” from a legal standpoint with the specification of the license:

```
{
        "Commentary": "Hubble telescope image artifact, HELIX
                  NEBULA, which is about 650 light years from the earth.
                  It is said to be the nearest planetary nebulae to the earth.",
        "Fieldsites": "Astroinformatics",
        "Format": "JPEG",
        "Group audience": "Astroinformatics",
        "Image Annotation": [],
        "Image URL": [],
        "License": "//creativecommons.org/licenses/by-sa/4.0/",
        "Location": {
           "lat": "20",
           "lat_cos": "0.93969262078591",
           "lat_sin": "0.34202014332567",
           "lng": "-10",
           "lng_rad": "-0.17453292519943"
        },
        "Tags": "hubble astroinformatics",
        "URI": [],
        "nid": "18",
        "node_created": "1434859251",
        "node_revision_vid": "18",
        "node_title": "Helix Nebula",
        "users_node_name": "sharon"
}
```


The second approach to disposition as per orientation of the RDA WG-PP is the rule for automatic deletion of artifacts that are marked as “expired.” This rule is turned-off by default for the PECE distribution, but it can be activated at any point by the administrator if needed. As discussed on the section on “Data Retention,” PECE is, differently from other projects for data management, specifically targeted for data sharing among ethnographers, so retention and expiration periods are not the rule but the exception in the context of our data practices. Other disposition policies can be configured to be triggered automatically in the system as well.

It is important to observe that “retention” and “expiration” are not common practice in the context of ethnographic projects, except for cases in which interviewees explicitly request that an interview or other any type of data with sensitive information must be destroyed after the research project is over. This can be done on PECE by setting up “expiration” dates as explained in the previous section on “Data Retention”. In the example below, whenever an artifact reaches the expiration date as defined by a user (if expiration date is needed since this is a *non-mandatory* artifact field), the disposition rule to remove the artifact is automatically executed. This is achieved with the following Drupal rule:

x


```
{ "rules_pece_disposition_rule": {
    "LABEL": "PECE Disposition rule",
    "PLUGIN": "reaction rule",
    "OWNER": "rules",
```

```
    "REQUIRES": ["rules", "node_expire"],
    "ON": {"node_expired": [] },
    "DO": [{ "entity_delete": {"data": ["node"]}}]
  }
}
```

7. Integrity and Replication

According to the RDA Practical Policies report, integrity policies consist in conducting a series of steps to guarantee file integrity in a collection. These steps of evaluation include regular checking of files checksums and data replication so as to ensure easy replication of corrupted files. In PECE, data integrity checking is performed primarily by the Drupal framework (through its Schema API) in conjunction with its database back-end, MariaDB: CRUD operations are handled by the Schema API, offering an abstraction layer for database operations on PECE/Drupal data structures, and the database server guarantees integrity through ACID (atomicity, consistency, isolation and durability) conditions for all data transactions. For automatic checking the integrity of database tables, we use the extension module “dba” which allows for checking, reporting, and repairing data corruption on a regular basis.

Data replication can be handled in many ways on PECE: 1) automated replication between production, testing, and backup instances for redundancy and/or performance (for advanced PECE administrators using our VM distribution: we discuss this configuration in the “PECE Technical Specification” document); 2) scheduled, automated server “snapshot” generation performed by the hosting service company to save the state of a particular instance; and last but not least 3) using PECE Open API to replicate the data of a particular instance. This last option allows for easy integration with large-scale data repositories as described in the section on “Metadata extraction” of this document. For administrators with *nix expertise, replication is also conveniently done with Drush (and batch operations using shell scripting).

```
# Replicating all the artifacts of a particular type, i.e. “images”
$ drush ne-export -t images -f images_output.txt

# Replicating all the artifacts of all types
$ for i in {images, text, audio, video, etc.}; \
     do drush ne-export --type $i >> “$i”.output.txt; \
     done

# Importing all the artifacts of a particular type
$ drush node-export-import --file=$filename.output.txt

# Export and import the entire instance for replication/redundancy
$ drush archive-dump default --destination=PECE.tar.gz
$ drush archive-restore PECE.tar.gz

# Export and import the database only
$ drush sql-dump > PECE_db.sql
$ drush sql-cli < PECE_db.sql
```

This command returns all the images with their respective metadata for replication purposes. In order to replicate binary files, it is necessary to also execute wget if replication of the PECE Image artifacts is successful. Please observe that checksum verification for binary files is currently not supported, it is a planned feature for PECE version 2.0:

```
# Replicating all the artifacts of a particular type, i.e. “images”
```

```
# Copying all the respective public binary image files as well
$ drush ne-export --type image >> images_output.txt && \
     wget --no-certificate -r -l1 -A “gif, jpg, png, svg” \
     https://your-domain.org/sites/default/files/
```

8. Notification

Drupal core provides logging capabilities through its `watchdog()` function which basically operates by registering system events, such as available updates, security issues, and user account events which can be, then, notified to administrators, researchers, and collaborators. Severity of events on Drupal is determined after the RFC3164 (which specifies the BSD syslog protocol). PECE has specific needs, however, that require extending the standard email notification system of Drupal.

Automated notification capabilities are handled on PECE by security modules (as explained in the “Data Access and Security” section) and messaging modules. These capabilities include the ability to report all sorts of events to the user on various levels: **system level** (related to the platform itself), **account level** (related to specific users), and **content level** (related to additions, modifications, and deletion of artifacts). PECE's notification system follows “user roles” when addressing specific users with respect to the nature of the event. It also supports notifications that are addressed to research groups via PECE's group functionality: OG member subscribe and OG new content creation, change, or deletion.

There are two types of notification: **email** and **in-system,** respectively, notifying users and administrators based on their email contact or upon log-in (as shown below as an example, the information about last successful log-in):

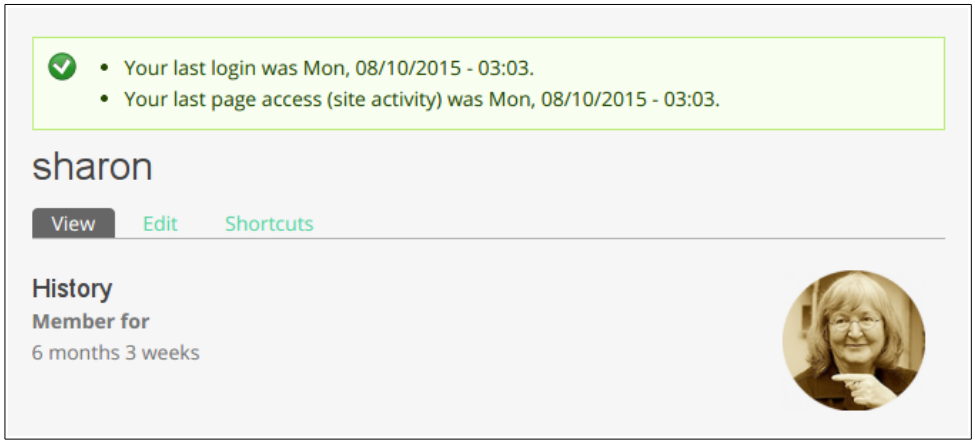


Figure 5. Successful authentication in-system notification, source: astroanthro.net

This type of notification is not only useful for security reasons (as explained in the section on “Data Security and Access Control”) but also for keeping users and administrators informed about the overall activity on the website with relation to different types of content.

Email notifications are by default configured to display: subject string, site name, addressee name, notification body text, and link (if relevant to a piece of content that was created, modified, deleted or expired). They can address individual user accounts or groups.

The table below describes the configuration of PECE's notification system in regards to scope, notification message, type, and addressee:

Scope	**Notification** Message	Type	Addressee
System	update	*email*	admin
	successful or failed backup	*email*	admin
	disk almost full (90%)	e*mail*	admin
User Accounts	creation (by admin)	*email*	researcher, collaborator
	awaiting approval	*email*	researcher, collaborator
	blocking	*email*	researcher, collaborator
	activating	*email*	researcher, collaborator
	cancelling	*email*	researcher, collaborator
	deletion	*email*	researcher, collaborator
	break-in attempt	*email*	admin
	password recovery	*email*	researcher, collaborator
	last login date/time	*in-system*	admin, researcher, collaborator
	last site activity date/time	*In-system*	admin, researcher, collaborator
Artifacts	creation	*in-system*	group
	change	*in-system*	content creator, group
	deletion	*in-system*	content creator, group
	expiration	*email*	content creator

Notifications are sent automatically depending on the configuration described above. They are configured and triggered by the “rules” module which monitors the system log and executes an action. Here is an example of an exported machine-actionable rule for notifying a particular user that his or her artifact has expired:

```
"rules_pece_artifact_expired" : {
   "LABEL" : "PECE Artifact Expired",
   "PLUGIN" : "reaction rule",
   "OWNER" : "rules",
   "REQUIRES" : [ "rules", "node_expire"],
   "ON" : { "node_expired" : [] },
   "DO" : [
     { "mail" : {
     "to" : [ "node:author:mail" ],"subject" :
     "[[site:name]]: \u0022[node:title]\u0022 has expired",
     "message" : "Dear [node:author],\r\n\r\nThe content for the
                      artifact [node:title] has expired on
                     [node:field-expirationdate].\r\nYou can access
                     the artifact at thisURL:\r\n[node:url]\r\n\r\n
                     This is an automatic notification from PECE\u0027s
                     [site:name].\r\n\t"}
```

This rule, for instance, is executed every time an artifact is modified in the system. It collects the title of the node that was modified and reports to the author of the node. Another example is the notification of a modification in an artifact if the modification was not performed by the author him or herself:

```
 "rules_pece_artfact_change" : {
   "LABEL" : "PECE Artfact Change",
   "PLUGIN" : "reaction rule",
   "OWNER" : "rules",
   "REQUIRES" : [ "rules" ],
   "ON" : { "node_update" : [] },
   "IF" : [
     { "NOT data_is" : { "data" : [ "node:author" ],
        "value" : [ "site:current-user" ] } }
     ],
   "DO" : [
     { "drupal_message" : { "message" : "Artifact
        \u0022[node:title]\u0022 has been updated." } }
 [...]
```

For these two examples, notifications are generated as an email and as a system status message.

One important observation regarding the notification system is that the logging function is pre-configured differently for the two main types of PECE distribution: **distro package format** (tar.gz) and **distro virtual machine (VM) image**. For the former, the core module “dblog” is used for logging events on the database whereas for the former, syslog at the OS level is used instead for better performance. This technical difference does not impact the management or usage of the system (except for the a small decrease in performance when using “dblog”).

9. Restricted Searching

According to the RDA Practical Policies report, “restricted searching can be viewed as a form of restricted access control” which can be implemented, for instance, using user roles and access control lists. Restricted searching is controlled on PECE through user roles – as explained in the section of this document on “Data Access” – and on an individual artifact-basis. Administrator, Researcher, and Collaborator roles, having different levels of access to content, also have different levels of access to the search functionalities of the system, being only allowed to search and find content that is available to them through the permission system. Administrators and Researchers have the ability to exclude particular nodes from search results, for instance. This option is available when creating or editing a particular type of content as shown below:

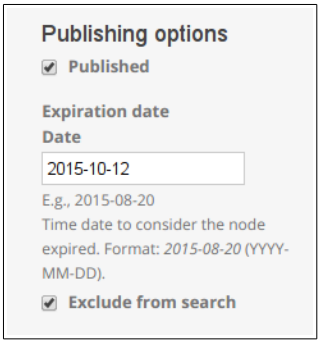


Figure 6. Excluding an artifact from the search index

Following the permission settings on the platform, content is only visible through the built-in search function to the authenticated users with specific roles:

Artifact Permissions	Authorized Roles	Access through Search API
Open	All	Non-restricted to all
Restricted	Researchers, Administrators	Restricted to other roles
Private	Individual user and artifact-based	Restricted to all

Administrators and users have the option of using **three search back-ends:** one is **Drupal's native search mechanism**; another is a connector from our platform to an **ElasticSearch back-end** (which can be used with in the future with our ElasticSearch cluster or your own infrastructure); and, finally, we will provide a **SPARQL endpoint** to communicate with a dedicated Semantic Web search server infrastructure. We will use the ElasticSearch and the SPARQL back-ends for searching content in the platform following the RDA policy for restricted content as well, but mostly for content that is open for non-restricted distribution under flexible copyright licenses. Both, the ElasticSearch and the SPARQL back-end will be used to index and query across several PECE instances in the near future.

In order to achieve our mission of promoting data exchange and enhance collaboration among ethnographers, we encourage users to release their data as often and open as possible (while being quite observant of the privacy and ethical issues when doing so). For this purpose, all the artifacts with “open” permissions are available to indexing and searching through our Open API and the pluggable extensions for ElasticSearch server back-end via Drupal Search API.

In terms of technical capability, PECE is shipped with scalable search server extensions in addition to the built-in restricted searching functionality of Drupal. Specifically, the platform distribution comes with an extension for the ElasticSearch search server back-end. Search servers are key for our web framework because they allow for powerful discovery capabilities in big corpus of texts and across different corpora of texts. It is a known limitation of the native search capability of Drupal to underperform with a SQL database with more than 50k documents/nodes. Another important benefit of having a search database back-end is the ability to perform searching across different PECE instances for identifying ethnographic content as well as for asking research questions across several ethnographic collections.

We have tested alternatives such as ApacheSolr and ElasticSearch and planned but not yet configured our scalable searching back-end. However, PECE comes ready to interface with ElasticSearch (if you are planning to use your own back-end instead of ours).

10. Instance Cost Reports

PECE depends on a set of Free and Open Source technologies that constitute the Drupal framework: *nix system tools (such as cron, drush, df, awk, bash, and other multimedia manipulation tools, such as FFmpeg), database server (such as MariaDB), scripting languages (such as PHP and Javascript), and a set of contributed libraries that are used for data manipulation, management, and security purposes.

Given the level of complexity of system administration in general, we recommend PECE users rely on managed hosting services offered by their universities or commercial web hosting companies. These options are recommended to PECE administrators who are not experienced in *nix system administration. For experienced administrators, we suggest contracting a virtual private server (VPS) that matches the size and the needs of your project as described below.

In order to provide PECE administrators with data on monthly usage for calculating costs, PECE relies on basic descriptive statistics that are generated by the Drupal core module “statistics” as well as information about disk usage that is gathered in the back-end at every cron run. This information is very useful when estimating data transfers and calculating incurring hosting costs. Fully automated gathering and reporting of the usage of computational resources (such as CPU time, IO, individual artifact sizes) is a functionality that is being planned for the version 2.0 of the platform. It is not currently supported on PECE itself, since this information can be easily obtained on a monthly basis when contracting a Drupal managed hosting company. Please observe that this is one of the benefits of having a managed *versus* an unmanaged host: the ability to obtain fine-grained information on usage of the platform and not having to dedicate considerable time administering it.

Hosting Recommendations

For calculating the cost of running and maintaining a PECE instance, we collected estimates from more than twelve web companies that specialize in Drupal and described their services along three tiers (small, medium, large) and four of the most important variables for assessing hosting costs: number of authenticated users, disk consumption for both the file system and the database (in GB or TB), system memory (in GB), data transfer allowance (in GB or TB) and available bandwidth (in Gbps), and vCPU (per number of allocated virtual CPU core units) as demonstrated in the table below:

Instance	Users	vCPU	Disk	RAM	Data Allowance
Small	10	2	10 GB	2 GB	100 GB
Medium	50	4	100 GB	4 GB	1 TB
Large	100	8	1 TB	8 GB	10 TB

These numbers represent a rough estimate of the recommended specs for the virtual private host (VPS) in cloud services of hosting companies.

Additional Considerations and Estimation Tools

It is important to bear in mind that these numbers can be quite different depending on the nature of the data that is hosted on PECE: audio and video files, for example, would create a different need in respect to the usage of disk, disk I/O, and RAM with substantial increase in the data transfer, therefore creating the need for bigger transfer allowances, if not for dedicated hosts and content delivery networks (according to the geographical distribution of users in a particular research collaboration).

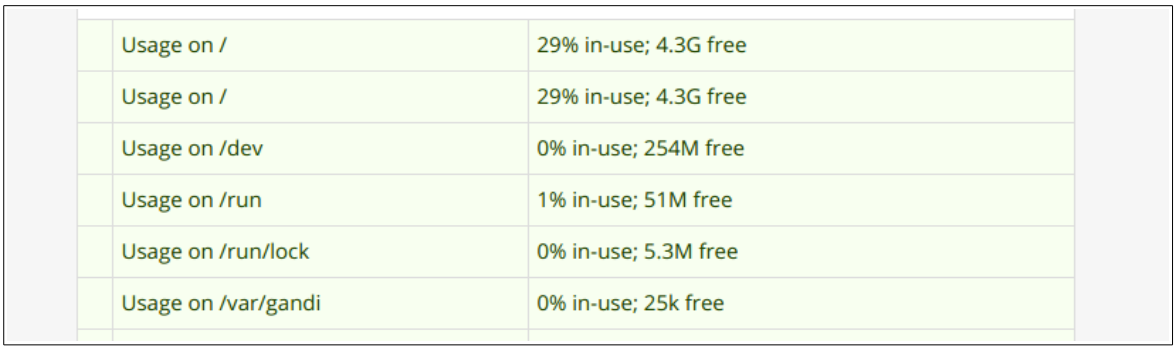
Another important factor to take into consideration is the number of published artifacts on the platform, which impacts overall performance and determines the need for more or less computational resources, making it difficult to estimate with precision. This estimation of basic hosting requirements was informed by the market research conducted by the PECE team throughout the summer of 2015. For estimating with more precision, PECE automatically generates access reports for individual artifact pages on a monthly basis (and comes with built-in modules to assess database and storage usage). The total bandwidth usage can be monitored and generated monthly by a hosting service provider, represented in the example image below, and notified to the administrator by email for VPS instances:

Top pages in the past *1 month 3 weeks*

Hits	Page	Average page generation time	Total page generation time
101	Data as Discursive Objects node/6	*306* ms	31 sec
37	Hubble's Diverse Universe content/hubbles-diverse-universe	*423* ms	16 sec
9	Statistics admin/config/system/statistics	*388* ms	3 sec
8	node	*2278* ms	18 sec
6	annotation/api/search	*329* ms	2 sec
6	Recent hits admin/reports/hits	*677* ms	4 sec
5	Status report admin/reports/status	*1664* ms	8 sec
5	Diskfree admin/config/system/diskfree	*529* ms	3 sec
4	Top pages in the past 3 months 3 weeks admin/reports/pages	*513* ms	2 sec

Figure 6. System resources utilization report (on PECE and on the PECE hosting service)

Coupled with basic statistics, PECE comes pre-configured with the “diskfree” module to run the df command and obtain information on disk usage for a particular instance. Whenever the disk usage reaches 90%, the administrator is informed by email that the disk is almost full:



Usage on /	29% in-use; 4.3G free
Usage on /	29% in-use; 4.3G free
Usage on /dev	0% in-use; 254M free
Usage on /run	1% in-use; 51M free
Usage on /run/lock	0% in-use; 5.3M free
Usage on /var/gandi	0% in-use; 25k free

Figure 7. Disk utilization report

The general orientation for administrators obtaining the PECE distribution via release package file, public repository, or one of our pre-configured virtual machine images is to dedicate one or more instances per project, that is, if a new project is created out of an ongoing project, it is recommended for one or more PECE instances to be created in addition. Using the PECE Open API, it is possible for users and administrators to share and harvest data from different PECE instances. Another important suggestion is for PECE administrators to rely on Drupal managed hosting companies in order to use their backup and system usage reporting capabilities. These services are important for redundant backup purposes as described in this document on the “backup” section.

11. User Agreements

According to the RDA Practical Policy report not all data management policies can be automated, which include user agreements, privacy policies, and other legal documents.

Upon registering an account on PECE, user agreement documents are displayed to the user – who has to read and agree with them, marking a check-box before being allowed to continue registering for an account on the system. The following documents are under revision at the Cyberlaw Clinic at the Berkman Center for Internet and Society at Harvard: *user agreement, privacy policy, and users' conduct*. After legal revision, we will include their final version in the PECE distribution, version 1.0.

User Agreement

The Platform for Collaborative and Experimental Ethnography (PECE), hereby represented by the PECE Team (see “Team-members-list”), is a web platform for collaborative work around the tasks of archival, analysis, sharing, and publication of ethnographic data. By using PECE, you accept the following terms and conditions, including those in our code of conduct and privacy policy documents. If you do not accept these terms, please refrain from using our platform. PECE is licensed under the GNU General Public License (GPL) version 3. The platform is based on the Drupal framework which is licensed under the General Public License (GPL) version 2 or later, including its contributed modules. Other third-party software included in Drupal and PECE is licensed under compatible Free Software licenses, which are included in our source code repository for public access.

PECE was created to promote Open Access, Open Data, and Open Standards in the humanities and social sciences. In order to achieve this goal, all of our generated content is licensed under the “Creative Commons Attribution-ShareAlike 4.0 International” by default unless otherwise noted for specific pieces of content. Users are responsible for describing the license they want for their own content (or the license chosen by the copyright owner for collected materials, if the content is being uploaded by a contributor not the original author).

All uploaded content is the sole responsibility of the person who published it. The PECE team is not responsible for the content posted by users of the platform and cannot monitor all the published content. Authenticated researchers and contributors are responsible for the content they upload to the platform, as well as for its usage. They are also responsible for anonymizing the ethnographic data they upload to the platform if the data carries any potential privacy issue. PECE comes with no warranty or guarantee of fitness for any particular use as described by its software license, GPL v.3. There are no restrictions for its use, copy, study, modification, or redistribution as described in the GPL v.3 license. We make no warranty as to the reliability, accessibility, or quality of our web services. When using the platform you agree that the usage of our services is at your sole and exclusive risk. The PECE team is not liable for any direct, indirect, incidental, consequential, or exemplary

damages, including but not limited to harm and damage to research participants of any kind and in the context of any research project making usage of the platform. We worked to minimize the security risks of the platform, but we cannot guarantee the complete security, anonymity, and confidentiality of data posted on the platform.

When in doubt regarding the privacy and ethical implications of a particular piece of data, please refrain from uploading it to the web. Always contact your research co-participants and your IRB committee.

Privacy Policy

PECE does not collect nor store any data on its users, except for the personal data that is given by the users themselves when creating profiles (“registration information”).

In order to achieve its mission of promoting collaborative work among researchers in humanities and social sciences, PECE privileges open data and open standards. It follows closely the best practices of Free and Open Source communities when dealing with privacy and security concerns. Open and full disclosure of any security problem is our responsibility. Following the Free and Open Source community practice, we do not hide technical problems from our users.

When designing and implementing software for our platform, we prioritized our research participants' right to anonymity, confidentiality, and privacy. PECE users (in the major roles of collaborators and researchers) are responsible for specifying the permission settings for every piece of content they upload. That is, if a piece of data will be public (accessible to anyone) or private (only accessible to the registered researchers, PECE researchers (with IRB approval) or collaborators and Internet anonymous users).

PECE was designed in accordance to the ethical guidelines of professional anthropological associations, such as the American Anthropological Association (AAA), Associação Brasileira de Antropologia (ABA) and the World Council of Anthropological Associations (WCAA). We follow the core ethical principles of protecting our research participants' rights to privacy, anonymity, and confidentiality. We aim to cause no harm to research participants or to any social group directly or indirectly as a consequence of research work in our platform. We subordinate our research goals to the ethical concerns and privacy needs of our research participants. Our goal is to encourage wider data sharing while protecting privacy and sharing of research data. PECE will be used for academic research and dissemination of data and research results in collaboration with other researchers and academic collaborators: data contributed by researchers and contributors will be controlled by themselves with permission settings they must specify during data entry. We do not collect nor share users' data, browser fingerprints, nor do we read, collect, or analyze communication between users.

We reserve the right to change this policy at any time. If we make changes, we will notify our users in a clear and prominent manner.

APPENDIX

PECE Data Model

0. User Roles

- Researcher
    - Access to restricted data
- Contributor
    - Access to group contributed data
- Anonymous
    - Public access data

1. Data Types

List of data types, their fields, and their relations.
References to other data types within a particular type are marked with *italics*.

Projects

- URI
- Title (required)
- Description
- Institution (Field link com title, multiple, and required)
- *Researchers*
- *Contributors*
- *Fieldsites*
- *Design and Substantive Logics*
- Funding Agency (Field link w/ title, multiple, and required)
- Interview Request Form
    - File Attachment (public)
- Consent Form
    - File Attachment (public)
- Start and End Date field (Only the start date is required)

Groups

Contributors and researchers can create groups

- URI
- Title

- Description
- Banner Image
    - File Attachment
- Email
- *Members*
    - *Researchers*
    - *Contributors*
- *Citations*
- *Memos*
- *Field Diaries*
- *Artifacts*
    - *Fieldnote*
    - *Text*
    - *PDF document*
    - *Image*
    - *Audio*
    - *Video*
    - *Website*
- Forum
    - Title
    - Topic
        - Title
        - *Author*
        - *Comments*
- *Permissions*

Fieldsites

Contributor and researcher can create fieldsites

Open to all users

- URI
- Title
- Description
- Location

Design and Substantive Logics

Admin and researchers can create design items

- URI
- Title
- *Researchers*
- Description
- *Image*

- *Authors*
    - *Researchers*
    - *Contributors*
- Type (Design/Substantive)
- *Citation*
- *Tags*

Artifacts

- **Fieldnote**
    - URI
    - Title
    - Date of creation
    - Date of publication
    - Date(s) of modification
        - Revision number
        - *Author*
    - *Author*
    - *Contributor (if different from authors)*
    - Text
    - *Fieldsite*
    - *Annotations*
    - *License*
    - *Permissions*
    - *Tags*

- **Text**
    - URI
    - Title
    - Date of creation
    - Date of publication
    - Date(s) of modification
        - Revision number
        - *Author*
    - *Authors*
    - *Contributors (if different from authors)*
    - *Fieldsite*
    - *Annotations*
    - Critical Commentary
    - *License*
    - *Permissions*
    - *Tags*

    - *Citation*
    - *Group Audience*
        - *Groups*
        - *Personal workspace*

- **PDF Document**
    - URI
    - Title
    - Date of creation
    - Date of publication
    - *Authors*
    - *Collaborators (if different from authors)*
    - *Fieldsite*
    - *Annotations*
    - Critical Commentary
    - *License*
    - *Permissions*
    - *Tags*
    - *Citation*
    - File attachment
    - *Group Audience*
        - *Groups*
        - *Personal workspace*

- **Image**
    - URI
    - Title
    - Date of creation
    - Date of publication
    - Date(s) of modification
        - Revision number
        - *Author*
    - *Author*
    - *Collaborator (if different from author)*
    - Format
    - *Fieldsite*
    - *Annotations*
    - Critical Commentary
    - *License*
    - *Permissions*
    - *Tags*

    - File attachment
    - *Location (if different from fieldsite)*
    - *Group Audience*
        - *Groups*
        - *Personal workspace*

- **Audio**
    - URI
    - Title
    - Date of creation
    - Date of publication
    - *Author*
    - *Collaborator (if different from author)*
    - Format
    - Duration
    - Transcript
    - *Fieldsite*
    - *Annotations*
    - Critical Commentary
    - *License*
    - *Permissions*
    - *Tags*
    - *Citation*
    - File attachment
    - *Location (if different from fieldsite)*
    - *Group Audience*
        - *Groups*
        - *Personal workspace*

- **Video**
    - URI
    - Title
    - Date of creation
    - Date of publication
    - *Author*
    - *Collaborator (if different from author)*
    - Format
    - Duration
    - Transcript
    - *Fieldsite*
    - *Annotations*

    - Critical Commentary
    - *License*
    - *Permissions*
    - *Tags*
    - *Citation*
    - File attachment
    - *Location (if different from fieldsite)*
    - *Group Audience*
        - *Groups*
        - *Personal workspace*

- **Website**
    - URI
    - Title
    - Date of creation
    - Date of publication
    - *Author*
    - *Collaborator (if different from author)*
    - *Fieldsite*
    - *Annotations*
    - Critical Commentary
    - *License*
    - *Permissions*
    - *Tags*
    - *Citation*
    - File attachment
    - *Group Audience*
        - *Groups*
        - *Personal workspace*

- **Bundle**
    - URI
    - Title
    - Date of creation
    - Date of publication
    - *Author*
    - *Collaborator (if different from author)*
    - *Fieldsite*
    - *Annotations*
    - *License*
    - *Permissions*

        - *Tags*
        - *Citation*
        - *Reference to other artifacts (unlimited)*
        - *Group Audience*
            - Groups
            - Personal workspace

License

- URI
- Name
- Type
- Description
- Logo
    - File attachment

Memo

- Title
- Text
- Author
- *Tags*
- *Comments*
- *Group Audience*
    - *Groups*
    - *Personal workspace*

Bibliography

(pulls all the biblio metadata from Zotero API)

- *Biblio entry*
    - **All the biblio fields,** including keywords
- *Tags*
- *Group Audience*
    - *Groups*
    - *Personal workspace*

PECE Annotation Scheme

Structured Analytics (Question set)

- Description: Collection of Questions/Annotations
- Type: Entity
- Title
- *Reference to Questions*

Analytics (questions)

- Type: entity
- URI
- Title (Question)
- *Author*
- Date
- *Tags*
- *Reference to Question Set*

Annotation (“Response to a question”)

- Type: entity
- URI
- Text body (long text)
- *Author*
- Date
- *Reference to annotation question*
- *Reference to content where was created.*
- Permissions
- License
- *Tags*

PECE Planned Features for Version 2.0

- Internationalization with translation to other languages
- Multi-author, collaborative real-time writing
- OpenGPG data encryption support
- Improved "password strength" estimator
- Support for collection-based persistent identifiers
- Support for perma.cc
- Support for back-end video/audio transcoding
- Update to Drupal 8
- Improved mobile support, ability to use PECE + smartphone as an ethnographic field tool
- Support for WebRTC-based audio/video recording of interviews and group meetings

PECE Open Data Management Issues

- Compare and align PECE Data management policies with other Data Management solutions (such as DataONE's Data Management tool)
- Revise PECE data model based on community feedback
- Improved metadata description based on community feedback
- Integration with large-scale data repository and management solutions, such as iRODS
- Support for embedded metadata for binary, multimedia files (ID3, EXIM, etc.)
- Ability to store encrypted documents, such as consent forms and other research organization documents with sensitive personal data

PECE Terminology

Artifact: any digital object produced in the context of an ethnographic project. For PECE, artifacts are of different types: text, images, websites, video, audio, PDF documents.

Artifact bundle: a combination of artifacts in one bundled artifact.

Ethnographic file: curated artifact that contains many content types to convey ethnographic analyzes.

Design logics: theoretical assumptions that undergird and orient design of digital infrastructures. PECE design logics are drawn from the work of humanities, animated by post-structuralist theories of language.

Substantive logics: logics of cultural and historical practices under study of a particular research project.

Structured Analytics: an open question set produced by a researcher to analyze particular artifacts or draw out analyzes from multiple artifacts; an entity that references several "analytics".

Analytic: an entity containing one question.

Annotation: an entity containing a response to one question (which includes tags, author name, date of creation, permissions, licensing information and other relevant metadata. On PECE, annotations are considered "nano publications".

Entity: instance of a digital object on the Drupal system (which is allowed to have any relationship to any type of content as well as to contain any field and metadata field).

Research Co-Participant: human subjects who collaborate with ethnographic projects. On PECE, they have the "collaborator" user role assigned to them.