Digital Humanities Application Development in the Cloud

John Mulligan, Ph.D.
Rice University (Houston)
Center for Research Computing (CRC)
jcm10@rice.edu
Working in the Cloud

The virtualization of IT labor
The availability of scalable resources promises:

• Scalability through networking
  • Not requests for but self-provisioning of resources

• Fungibility through virtualization
  • Raw resources can be repurposed to any specific end
An ideology critique of the cloud

• The acceleration of research
• Resources on demand
• The fetishization of computing:
  • Not automation
  • But the invisibilization of labor
Automation *and* fetishization (a double-edged sword)

[T]he technologies and techniques of IT are not necessarily fused to the doxa of postindustrialism (restructuring, lean production, and all the rest of the new management dicta). As a mode of development, IT also generates what amounts to a semi-autonomous doxa (a *belief* in information or in technology), akin to the faith that a rural laborer in revolutionary France, about to burn the house of the nearest aristocrat, might have expressed by picking up a pitchfork and saying, “*This* is what I believe.”

Use-Case

Developing Custom Humanities Software as a Service in the Cloud
Zotero lock-in
Tropy lock-in

https://docs.tropy.org/using-tropy/add_files

https://forums.tropy.org/t/exporting-as-csv-file/365/3
Manage Drive metadata (beta)

This feature is available with G Suite Enterprise, Enterprise for Education, Drive Enterprise, Business, Education, and Nonprofits edition. Compare editions

With Drive metadata (now in beta), you can add custom categories and properties to files and folders in Drive, in a consistent way across your business or school. Users can then search for content using specific terms that are meaningful to your organization.

What is metadata?

Metadata is simply information about your files. In Google Drive, metadata is organized into categories that map to common types of files in your organization. (A category is equivalent to "schema" in common metadata terminology.)

Each category is a set of structured properties that are relevant for a particular category of files.

For example:

• You might have a category called Contract.
• The Contract could have properties called Type, Company, Status, and Due Date.
• You assign each property a data type (number, date, person, text, or selection). If the type is selection, you define the possible values. For example, Contract's Status property might be a selection of Draft, Awaiting Signature, Rejected, Signed, or Expired.

Use metadata categories to...

• Categorize common file types for everyone at your organization
  Administrators can create categories that all your users can apply to their files.

• Define and scope relevant categories to team content

https://support.google.com/a/answer/9292382?hl=en
Local vs. Networked Image Archiving Apps

Scalable CentOS VM(s):
- Custom database
- IIIF Loris image server
- Python Flask server
- Background scripts

Web interface

Locally-hosted image interface & background scripts

Plugins, e.g.,
- OMEKA export
- HTC for OCR
- Google Drive Sync

Custom database

Image store

Scalable networked image store

“Scalable” plugins, e.g.,
- HTC for OCR
- Google Drive Sync
Prototype interface screenshots
Advantages of assuming scalable networked storage:

- No hard limit on archive size
- Backups/migration easier to automate and scale (multiple locations/platforms)
- Batch operations scalable (affordances for supercomputing)
Reflection on Methods

From jump-starting to incubating
Jump-starting vs. incubating

- **Jump-start**
  - Upsides:
    - Knowledge-sharing
    - PI experience
    - Seed funding
  - Downsides
    - Redundancy in labor (too many cooks)
    - Learning curve

- **Incubation**
  - Upsides:
    - Division of labor & domain expertise
    - Rapid prototyping
    - Institutional embedding
  - Downsides
    - Redundancy in software (reinventing the wheel)
    - Rapid obsolescence
Digital Humanities Application Development in the Cloud

John Mulligan, Ph.D.
Rice University (Houston)
Center for Research Computing (CRC)
jc10@rice.edu