

# dmBridge: A public access framework and alternative templating system for CONTENTdm

Alex.Dolski@unlv.edu Web & Digitization Application Developer  
Brian.Egan@unlv.edu Web/Multimedia Designer  
University of Nevada, Las Vegas Libraries

# Topics covered

- Introduction
- Feature-by-feature walkthrough
- Template crash course
- Planning & deployment logistics

# INTRODUCTION

# What is dmBridge?

- An alternative templating system for CONTENTdm with additional features
- Enables enhancement and modernization of public digital collection web interfaces
- Provides a lot of extra features without any custom programming

# What is dmBridge not?

- A content management system
- A general web application framework
- A CONTENTdm replacement
- A CONTENTdm customization
- A set of drop-in replacement templates
  - You have to write your own HTML/CSS; dmBridge only streamlines the process

# Who can use dmBridge?

- CONTENTdm users with web design experience
- Web designers with zero programming experience
- Programmers from novice to expert
- Both of the above, collaboratively!
- Beginners can get started faster and more easily
- Advanced users can accomplish more in less time
- At UNLV, we have been using it since last fall
  - All of our work with the default templates has been suspended

# Why develop dmBridge?

- The default templates were no longer acceptable to us
  1. Anti-modular; not conducive to customizations
  2. Code base beyond repair
  3. Force the programmer to work within an ad hoc “code soup” paradigm
  4. Heavily dependent on unnecessary JavaScript and invalid, non-semantic HTML
  5. Ignores recent trends and best practices in web application architecture
- Bottom line: pervasive systemic defects

# Our dmBridge-enhanced collections

- *“Welcome Home, Howard!”*
  - dmBridge pilot project, launched November 2008
  - <http://digital.library.unlv.edu/hughes/>
- *Southern Nevada: The Boomtown Years*
  - Launched May 2009
  - <http://digital.library.unlv.edu/boomtown/>
- More to come...

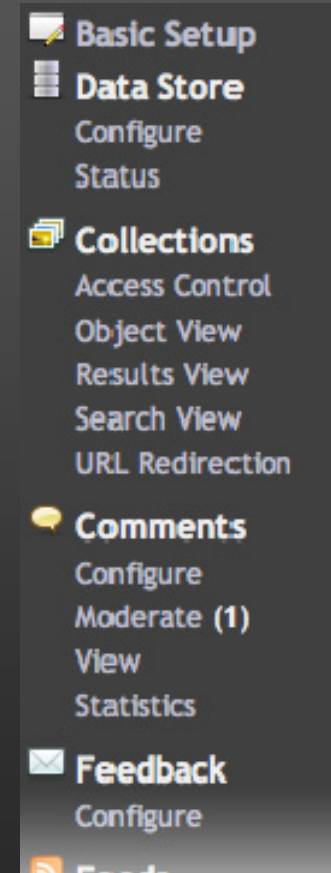
# FEATURE TOUR

# Some features unique to dmBridge

- Scheduled “object of the day/week/month”
- “Random object” from one or all collections
- Output format independence (HTML, XML/XSLT, JSON...)
- RSS/Atom search subscriptions
- Commenting with moderation and notification
- Extensions support
- Relatively pretty URLs
- Object-oriented PHP API
- XML/JSON REST API
- Search suggestions based on CONTENTdm field indices
- Server-side page caching
- Object ratings
- Metadata output in multiple formats
- Integrated object email feedback
- Configurable tag clouds based on DC metadata
- Easy web-based control panel for configuring most of this stuff
- More to come...

# The Control Panel

- Graphical interface for configuration and maintenance tasks
- Uses the same authentication system as the CONTENTdm Administration Module
- Functionality is mutually exclusive with the admin module (no overlapping features)



# Object commenting

- Optional comment moderation
- Optional e-mail notification of new comments
- For compound object pages, comments are associated with the parent object and appear on every page
- Can be enabled or disabled independently on a per-template-set basis

# Object feedback

- Like a comment, but gets sent as an email instead of being posted to the page
- Email contains a reference to the object
- Cleaner & more professional than a mailto: link
- Can be enabled on a per-template-set basis

# Object ratings

Are you **HOT** or **NOT**?  
[Submit Your Photo](#)

Select a rating to see the next picture.

**NOT**  1  2  3  4  5  6  7  8  9  10 **HOT**

- View highest-rated objects in the Control Panel
- Can appear on any scale
- Can be enabled on a per-template-set basis

# The highlighted & random object

- Set up in the Control Panel
- Highlighted object can be returned as an XML feed:

<http://digital.library.unlv.edu/hughes/dm.php/object/highlighted>

- Both can be returned in code:

```
$obj = dmObject::getHighlighted();
```

```
$obj = dmObject::getRandom();
```

# Tag clouds

- Based on collection DC metadata fields
- Can define tag cutoff, randomization, and tag CSS classes based on term frequency
- Code to draw this cloud:

```
$c = new dmCollection('/snv');  
echo Draw::tagCloud(  
    $c, array('subjec')  
);
```



Round Mountain **Mine buildings**  
Horse teams **Automobiles** Business &  
finance Railroad strikes Mills  
**Business districts Headframes**  
(Mining) **Railroads Men**  
**Houses Tents Women**  
Columbia Mountain Hills Crowds  
Interiors Portraits Miners Carts &  
wagons Architecture Dogs

# Search views

- Advanced search
  - You choose which search forms to include:
    - Field, date, and/or proximity search
  - Collection checkboxes are generated dynamically based on template & CONTENTdm access privileges
- Compound object search
  - Shows the full text of all pages for which there are matches, with matching terms highlighted

# Search queries





- Simple
  - `http://myserver/mytemplates/dm.php/?q=cheese`
- Advanced
  - Supports the full capabilities of the CONTENTdm search engine
  - `/?modes[]=all&fields[]=title&strings[]=cheese&aliases[]=/uw`

# Search suggestions

- Use indexed field terms from CONTENTdm
- Can be hooked up to text inputs via JavaScript/AJAX
- See our dmBridge collections for examples

# RSS, Atom, & other output formats

*Example:* <http://digital.library.unlv.edu/hughes/dm.php/?format=xml>

|            | Object View   | Results View  | Obj.-Results View | Favorites View | Search View |
|------------|---|---|-------------------|----------------|-------------|
| HTML       | ✓   | ✓   | ✓                 | ✓              | ✓           |
| Atom       |  |  |                   |                |             |
| RSS        |  |  |                   |                |             |
| RDF/XML    | ✓   |   |                   |                |             |
| JSON       | ✓   | ✓   | ✓                 |                |             |
| XML        | ✓   | ✓   | ✓                 |                |             |
| XML + XSLT | ✓   | ✓   | ✓                 | ✓              | ✓           |

# The data store

- Used to store dmBridge-specific data
- Modular and abstracted
- Currently, modules exist for:
  - On-disk XML files
  - PDO (MySQL, PostgreSQL, etc.)
- All CONTENTdm data continues to live in CONTENTdm, of course

# Extensions

- “Extensions” let you:
  - Override any of the “Draw” (helper) methods
  - Write new custom methods that do whatever
- Extensions:
  - Isolate & modularize custom code
  - Should be easy for any programmer to grasp
  - Can be shared with other dmBridge users
  - Can be made available on a global or per-template-set basis

# Image viewer challenges

- “As long as we’re rewriting the entire templating system...”
- All alternative viewers have at least one of the following problems:
  - Require manual pyramidal tile pregeneration
  - Google logo in the corner
  - Spotty support for JPEG2000
  - Server requirements (e.g. Djatoka)

# Image viewer

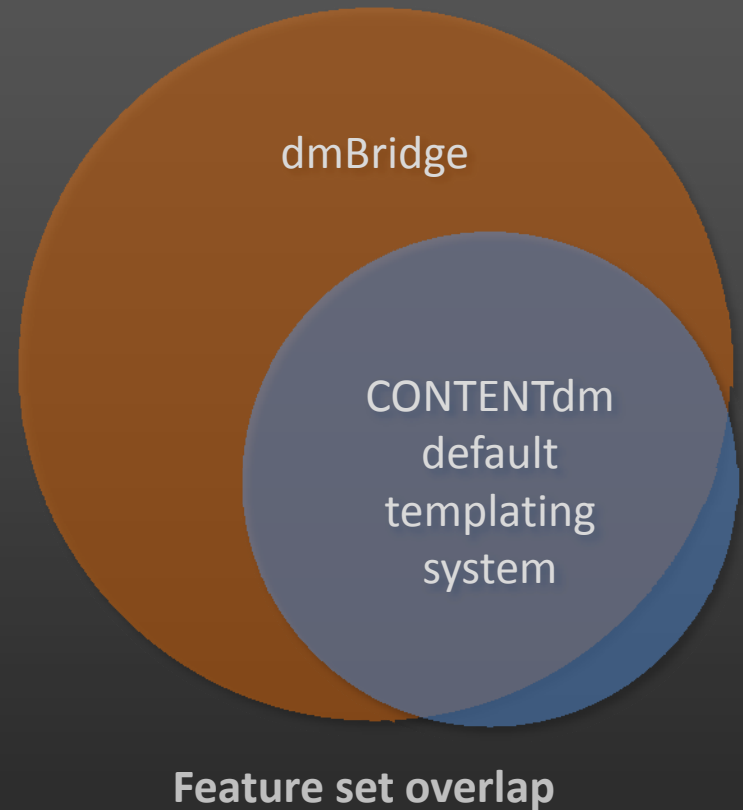
- Written by Brian in JavaScript/jQuery
- Separate from dmBridge; works with or without it
- Image tiles generated on the fly by getimage.exe
  - No pregeneration or extra software needed
  - Supports JPEG2000
- Easy dragging and zooming
- Customizable control appearance
- Demo:
  - <http://digital.library.unlv.edu/hughes/dm.php/hughes/9>

# Other media viewers

- CONTENTdm supports any file type; dmBridge can as well\*
  - (\*showfile.exe sends everything as application/octet-stream; nothing we can do about that)
- Viewers are modular; can plug in your own
- Custom viewers (like all other extensions) are not clobbered by dmBridge upgrades
- Assign any viewer for a particular file type to any collection

# What is dmBridge missing?

- dmBridge offers most the functionality of the default templates
- Notable absences:
  - Comparative object viewer
  - Faceted browsing
  - Slideshow viewer
  - Some other minor stuff
  - These are possible; we just haven't got around to them yet



# THE PAGE TEMPLATES

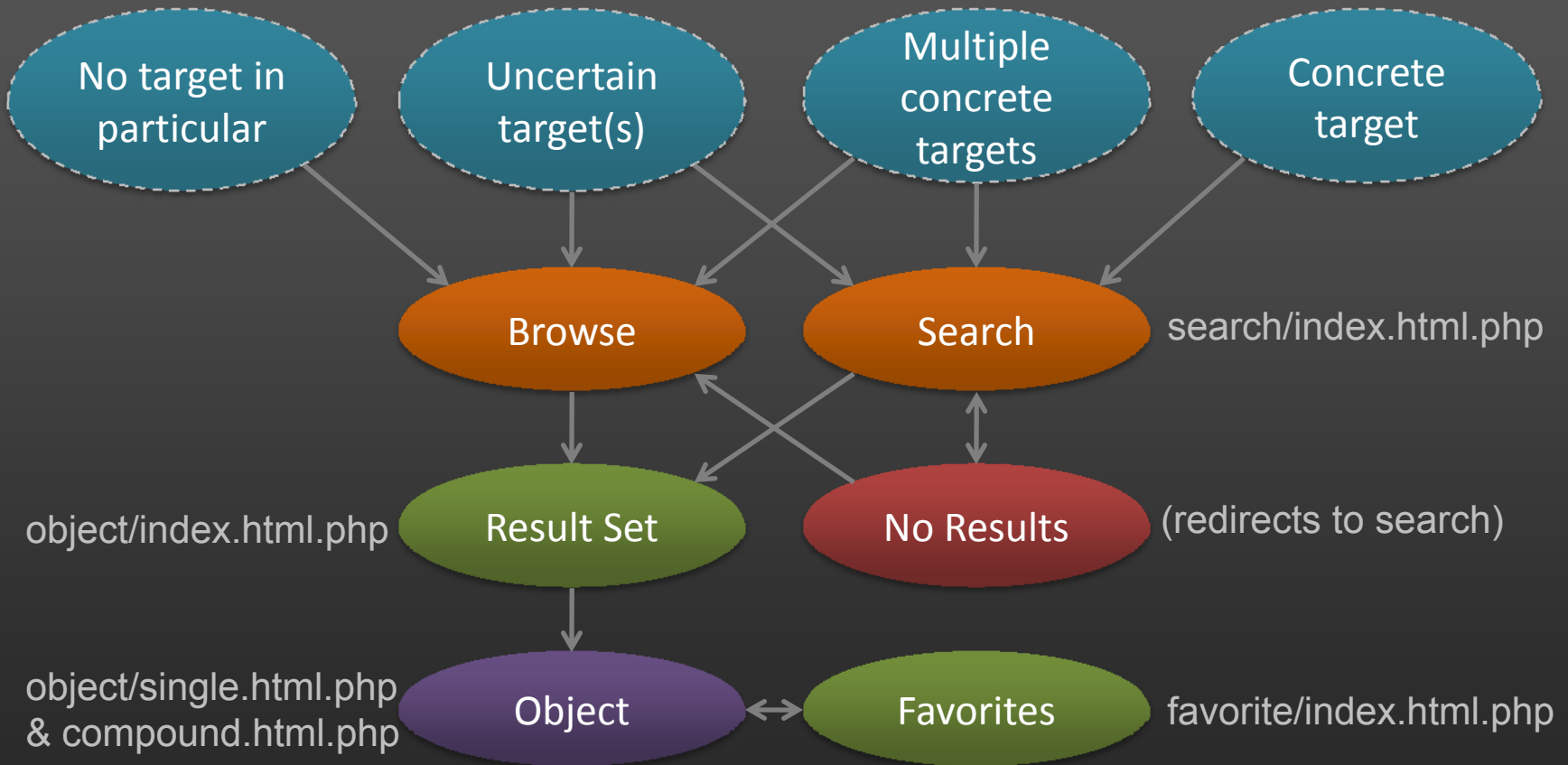
# Working with the templates

- The template creation process is simpler, but different
- dmBridge leaves most aspects of template design up to you
- You decide what elements to include on the page and how to style them
- Fully-functional sample templates are provided

# Working with the templates (cont'd)

- In dmBridge, you have six different views
  - Single object, compound object, results, object-results, favorites, search (see next slide)
- Mark up and style each one using standard HTML/CSS
- Call up dynamic functionality with one-line calls to dmBridge helper (“Draw”) methods
- No knowledge of dmBridge inner workings is necessary

# Template-view correlation



# Using the helper methods

This code in the object view  
template...

```
<div id="metadata">  
  <?= ObjectDraw::metadata() ?>  
</div>
```

ObjectDraw::metadata() is  
just one of over 50 built-in  
helpers available for doing all  
kinds of stuff

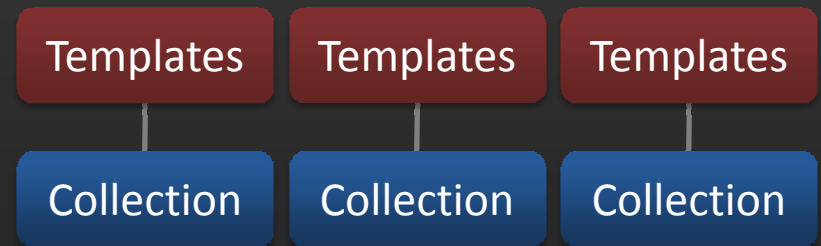
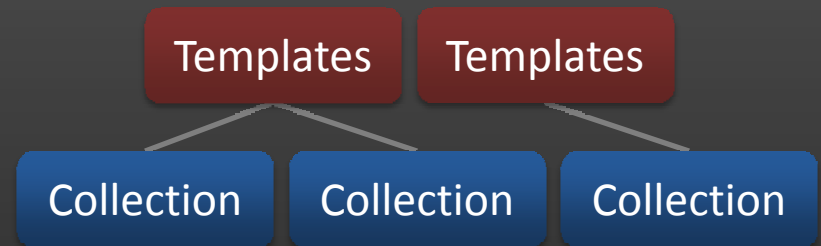
Becomes...

```
<div id="metadata">  
  <dl>  
    <dt>Title</dt>  
    <dd>Some great object</dd>  
    <dt>Description</dt>  
    <dd>A great object from one of our  
    great collections.</dd>  
    <!-- etc. -->  
  </dl>  
</div>
```

# PLANNING & DEPLOYMENT

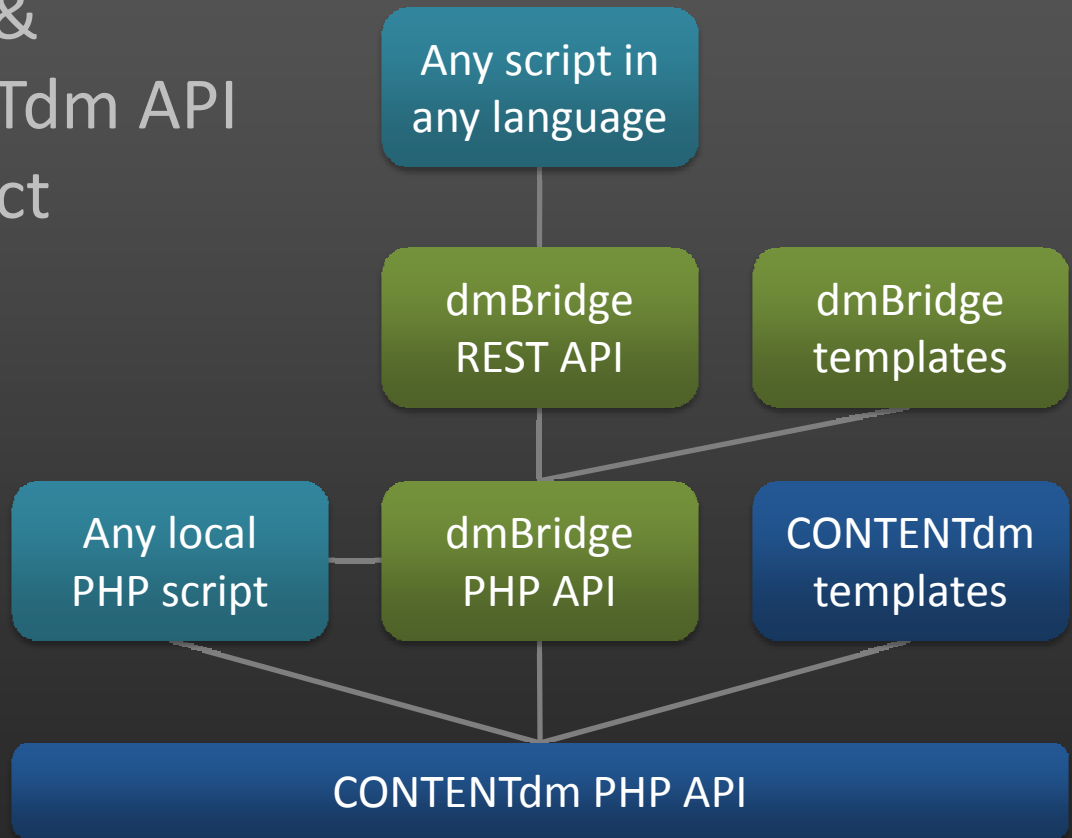
# How can dmBridge be used?

1. As a standalone templating system
  - One or more template sets connected to a single instance of the dmBridge core
  - Each set can have its own “look,” collection access privileges, etc.



# How else can dmBridge be used?

2. As a more useful & intuitive CONTENTdm API for any PHP project
3. As a web service



# Migrating to dmBridge

- dmBridge templates can coexist with the default templates
- The default templates are never touched; nothing happens to them. Just change your links
- Reference URLs can be rerouted on a per-collection basis (use them!)
- For most, an iterative migration (like ours) will be more comfortable than all-at-once

# Is dmBridge a long-term commitment?

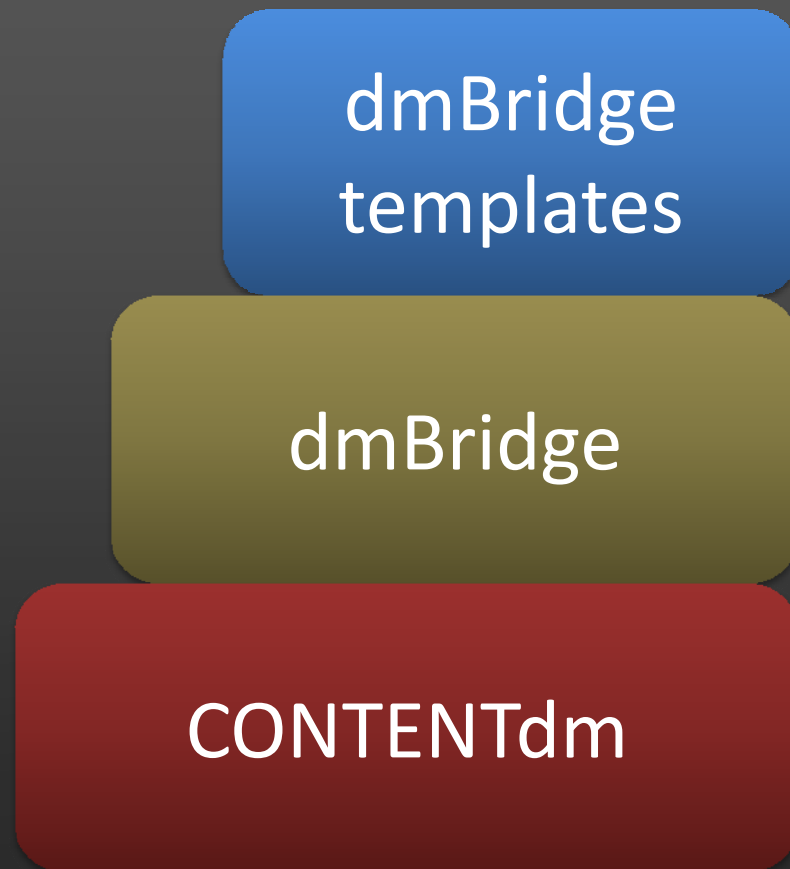
- It's already working and proven at UNLV, and is a core part of our technical strategy
  - We plan on converting all of our collections to it eventually
- Its data schemas are simple and open
- Code base is clean and understandable by anyone with MVC & classical OO experience
- You can always return to the default templates

# How hard is it to install?

1. Copy the dmBridge core folder into your web server's document root
  2. Fill in some info in the Control Panel
  3. Set some config file paths
  4. Create as many custom template sets as you like
- A few more steps, but nothing complicated

# What about upgrades?

- Components (→) can be swapped out more-or-less independently
- Minimal (if any) template changes should be necessary when upgrading either CONTENTdm or dmBridge
- Older dmBridge templates can be easily upgraded to use new features



# Requirements

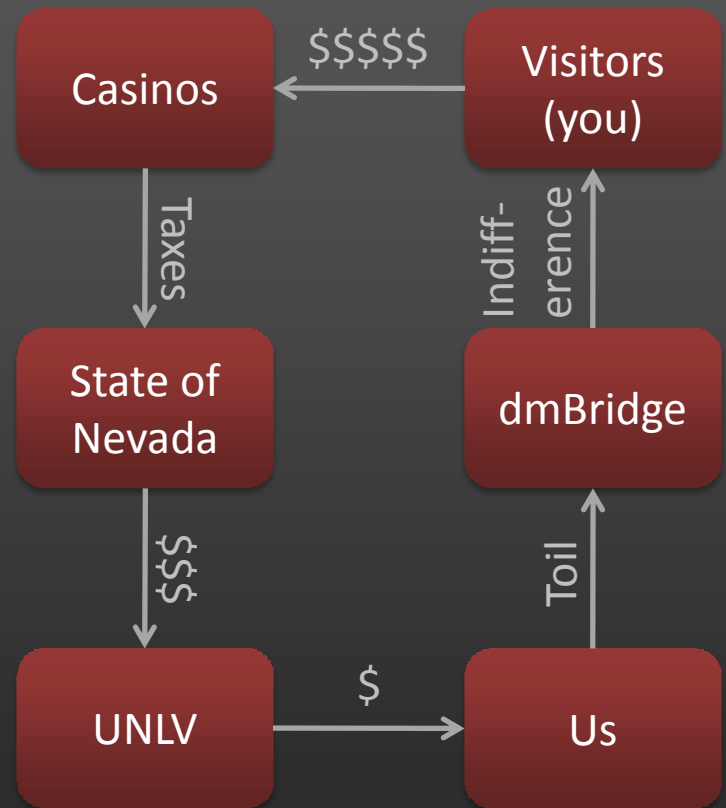
- Personnel:
  - HTML/CSS aptitude
  - Sense of adventure!
- Software:
  - If you are running CONTENTdm, you probably already meet the requirements
    - The PHP cURL extension is required
    - PHP 5.3 is needed for custom object viewers
    - The PDO & XSL extensions are optional

# Availability

- We're still exploring our options

# Thank you!

- Thanks UNR!
- Enjoy Reno!
- Bike safe!
- alex.dolski@unlv.edu
- brian.egan@unlv.edu
- <http://digital.library.unlv.edu/dmbridge/>



The circle of dmBridge life