Digital Repository 2.0
Lessons Learned and Applied

Kurt Nordstrom
Brandon Fredericks

University of North Texas Libraries

Where we've been

The Beginnings

- Grant awarded in 2002 to create Portal to Texas History
- Newly formed Digital Projects group in Library
- Challenges of Specifying a System to manage nonexistent objects
- Index Data chosen to build initial system

Toolkit Lite (TKL)/Keystone

- Digital Library Web Framework created by Index Data
- Open Source
- Standards compliant (including Z39.50)
- Linux/Apache/PHP based platform
- Filesystem-based XML Metadata storage and retrieval
- XML/XSLT configuration and templating

TKL/Keystone (continued)

- Several facets of key functionality dependent upon software tools with small user base
- Limited support community
- Limited scalability options
- URLs tightly coupled with filesystem on disk
- Particular implementation and leverage of XSLT not well-suited to our workflow
  - Inextricably linked with a particular implementation of XSLT (Sablotron)

Data Model

- “Homegrown” XML Schema for storing Descriptive Metadata, Preservation Metadata and Digital Objects all in one file
- Expanded from base TKL model
- Several revisions
- Single, linear Digital Object representation

System Architecture

- “Shared Everything” - All services for the Portal existing on a single, “high-powered” server.
- Second “clone” server for redundancy.
- Test server for development.
- Scaling issues.
  - CPU Constraints.
  - Disk Space Constraints.

Solr and Beyond

- Powerful, self-contained indexer as a web service
- First realizations of a distributed system
- Active user and development support community
- Seeds planted for new system design

Where we're going

Technologies

- Python
- Django
- Subversion w/Trac

Python

- Easy to understand and use
- Easy to deploy
- Has a pre-built library for most use cases
- Works well in every facet of our development

Web Frameworks

- Rapid development
- Active Community
- Collective Solutions
- VERY Deployable (with help from code repositories)
- MVC – loosely coupled

Revision Control and Tracking System

- All the basic version control features
- Web visualization
- Built in workflow tools
- Simple Deployment

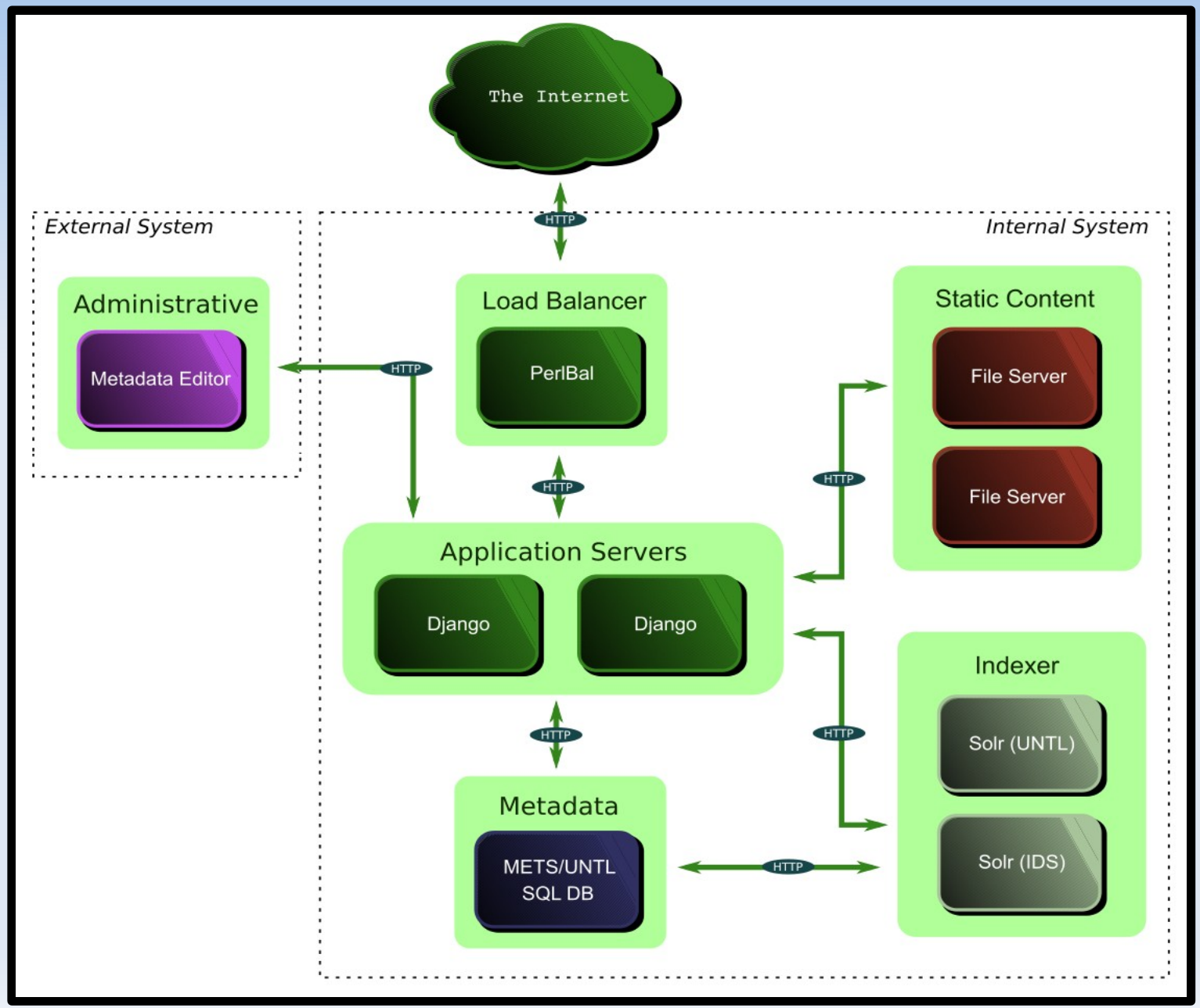
Metadata Models

- UNTL-BS
  - Fits our descriptive needs
  - Easily converted to Dublin Core, MODS, etc.
- METS
  - Well supported, well documented, often used
  - Manifestation support
  - Embedded or referenced metadata
  - File descriptions and structures

"Shared-Nothing"

- Multiple static file nodes (w/ redundancy)
- No concept of "THE server for a task"
- Still served under the same URLs

Architecture and Scaling



Conclusion

- Smaller, well supported/developed, interchangeable parts
- Experiment with technologies and find the best option
- Require an active community
- Your vision is bigger than the technologies used to create it