



Darc: Empowering LEP/MOD Test Data with Efficient Archive and Search

Yash Raj Singh, Jeffrey Yum, Robin Lee Simpson
WCI/ASQ



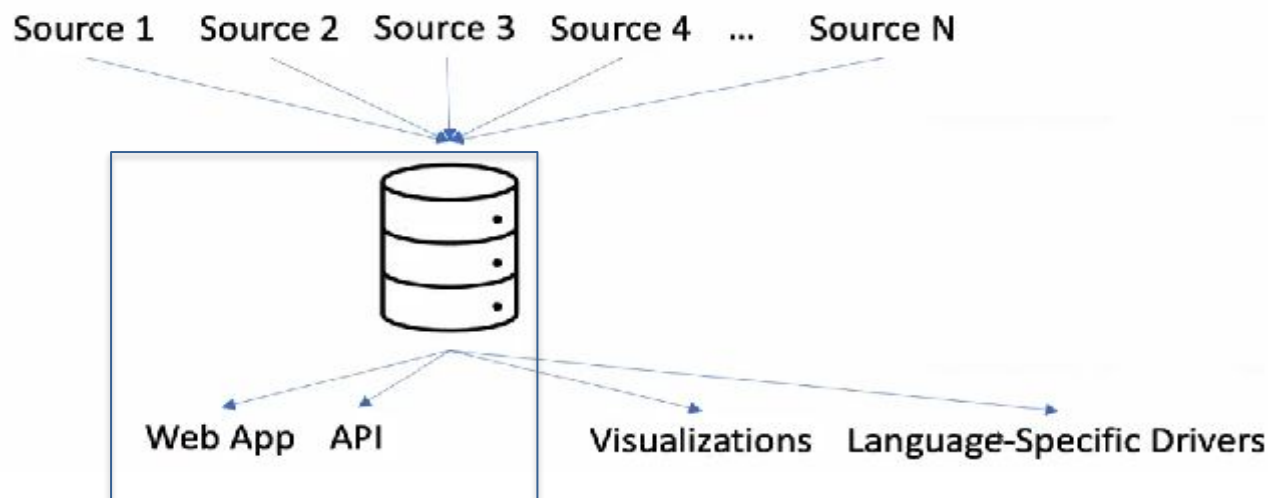
LLNL-PRES-851997

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC



Darc = Data Archive of test data

- Greater objective: Support the Life Extension Program, Modification Program
- Need assurance of healthy stockpile □ motivates heavy testing of individual components, generating large volumes of data (petabytes)
- Different customers, varying structures of data
- Goals: Discoverability, many interfaces for access





How this Affects the other Lab Divisions

- Darc replaces existing solutions like shared drives with a more robust data storage system accessible through an API.
- Part of the DLM (Data Lifecycle Management Team) and a sub-group of WCI-Apps focused on developing and deploying applications for test and material data management.
- At its most basic – just a better file browser
 - Windows Explorer-like file navigator
 - Advanced search
 - Batch import/export
 - Metadata tagging
 - Access control and user management





Internship Work Overview

- Testing
 - Validate actual API usage from a client's point-of-view
 - Validate that code functions as expected
- Feature additions
 - New endpoints, improved search, added relationships
- Bug fixes

~1000 intern contributions in the form of commits, comments, reviews, and merges.

Project Highlights – Darc / Darc Migrator



■ Darc

- Frontend: Angular
 - Popular front-end web application framework
 - Used for building our user-facing Darc application
- Backend: MarkLogic □ MongoDB + Elasticsearch
 - Why? 🤔 Due to scalability, developer-friendliness, & cost-effectiveness



■ Darc Migrator

- Project to transfer over documents from MarkLogic to MongoDB



mongoDB



elasticsearch



Project Developments

- Darc Migrator
 - Added features to improve data migrating capabilities
 - Supported transforming files into binary in smaller chunks
 - Reinforced consistency of transferred data
- Darc REST API Endpoints
 - Added endpoint for bulk file metadata retrieval
 - Reduced errors by automating record creation from CSVs



Searching / Testing Reliability

- Searching for data:
 - Should be fast and tolerant of user inputs ☐ Better User Experience
 - Can allow the user to be expressive ☐ Fewer, more accurate results
- Testing: Need to make sure everything works properly
 - Unit testing: Ensure that each component of codebase works exactly as intended
 - Integration testing: Ensure that components working together produce the expected results and side effects

