

Iterating from Prototype to MVP: A Blue Core Project Report

Kalliopi M. Mathios¹  

¹ Stanford University

Abstract

The Blue Core project completed development on a prototype in 2025, realizing the vision for a community-operated and owned BIBFRAME data store and providing a valuable environment for testing integrations with two open-source linked data editors: Sinopia and Marva. Thoroughly tested for scalability, metadata operations, and integration with local Library Services Platforms, the prototype informs plans for a Blue Core MVP that leverages agentic AI to enhance cataloger workflows.

1. Introduction

After two years of careful planning and software development, the Blue Core prototype provides catalogers the opportunity to create and edit BIBFRAME metadata in a shared data store with multiple open-source linked data editors: a first in the field. To realize our goals to bring production BIBFRAME cataloging to our institutions efficiently and effectively, thorough testing of the prototype took place in late 2025 and early 2026. In addition to testing prototype functionality, we also assessed our organizational processes, team best practices, and analyzed workflows to define requirements for an MVP system that supports production cataloging.

2. A Blue Core Prototype

Building on the successful design of the Sinopia cataloging environment, Blue Core serves as a central hub for authoritative metadata that transcends institutional silos and offers catalogers more choice in tooling for BIBFRAME cataloging. At the time of writing, the Blue Core prototype holds over one million BIBFRAME Work and Instance entities provided by the Library of Congress.

The Blue Core Prototype is deployed on a large EC2 instance on AWS and includes the following services:

- Blue Core API - A FastAPI-based REST API and MCP endpoint for Blue Core. Linked data editors CRUD endpoints and allows LLMs and AI Agents to use Blue Core through the MCP endpoints of the API.
- Blue Core Datastore - A Postgres database stores BIBFRAME Works and Instances as normalized JSONB with full-text search. Stubs for other resources, including RDF authorities, are stored usually with a RDF type and label triples as JSONB. Full Sinopia RDF templates are also stored as JSONB.
- Blue Core Workflows - An Apache Airflow 3.x instance for batch processing and integrations with external Library Services Platforms like FOLIO and Alma. Workflows are DAGs that can be triggered by the API or directly by users.

Project Report

Available Aug 01, 2026

Correspondence to
Kalliopi M. Mathios
kmathios@stanford.edu



Copyright © 2026 Mathios. This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license, which enables reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator.

- Sinopia - A forked version of the Sinopia linked data editor.
- Marva - A forked version of the Marva BIBFRAME editor.
- Graph Toolbox - A pyscript application used for experimenting with different client-side services include search, SPARQL query and search, marc2bibframe conversion, bibframe2marc conversions, and batch editing of a locally loaded graph of Blue Core Works and Instances.
- Keycloak - provides authorization and authentication middleware for the other services.
- Milvus - vector database supporting a Triples RAG for BIBFRAME Works and Instances in Blue Core. Underutilized because of the unavailability of GPU resources at this time.

3. Blue Core Prototype Assessment

The Metadata and Cataloging team developed a process for assessing the Blue Core prototype, with user testing design and coordination led by John Mark Ockerbloom at the University of Pennsylvania. This assessment included over 120 user tasks, organized by type. User tasks included adding a statement as a literal and/or lookup to an external data source, deleting a URI and label from a description set, and editing both literal and lookup values. Search tasks focused on finding bibliographic entities using various methods including exact titles, keywords, creator names, standard identifiers (ISBN/ISSN), subjects, and different script types (Chinese, Japanese, Arabic, Persian). The tasks also included filtering and limiting search results, searching for related works, and handling variant titles and cataloger information.

4. A Blue Core MVP

Blue Core prototype assessment results informed the planning and requirements for a Blue Core MVP. The cataloging of a single description and the bulk ingest of description set requirements will enhance the existing Blue Core prototype functionality. We plan to make use of the full Library of Congress BIBFRAME database for Hubs, Works, and Instances, greatly expanding the number of available entities. Plans for deduplication and clustering are underway, and cross-functional analysis of data provenance and version history, in collaboration with the BIBFRAME Interoperability Group, will suggest standards for responsible BIBFRAME data stewardship.

For the MVP, we will host three on-premises environments at Stanford University. Development, stage, and production environments will have a large virtual machine hosting the Blue Core stack along with a dedicated Postgres database server. The team is exploring the use of AI tooling and agents for deduplication and clustering, application profile editing, and metadata mapping workflows in Blue Core using the existing relational and vector databases for semantic search.