earkweb – Repository für die digitale Bewahrung

earkweb ist ein Open-Source-Archivierungs- und Digital Preservation-System, das auf dem Referenzmodell für ein Open Archival Information System (OAIS)[1] basiert und Funktionen für Ingest, Archivierung, Zugriff und Verwaltung von Informationspaketen bietet. Die Informationspakete für Ingest, Archival Storage und Access entsprechen den Spezifikationen für eArchiving (E-ARK) Informationspakete, die von der eArchiving-Initiative der Europäischen Kommission definiert wurden.[2]

Der Lebenszyklus eines Informationspakets beginnt mit der Bereitstellung eines E-ARK-Submission-Informationspakets (E-ARK SIP)[3] für den Ingest. Dieses kann entweder mit einem externen Tool erstellt werden, das in der Lage ist, Informationspakete gemäß der E-ARK-SIP-Spezifikation zu erzeugen, oder mit dem integrierten SIP-Creator von earkweb. Während des Ingests führt das System eine Reihe von Workflow-Schritten aus - einschließlich der Validierung des Submission Information Package gegenüber den Anforderungen der Spezifikation -, die im Erfolgsfall mit der Erstellung des Archival Information Package (AIP) endet. Es unterstützt auch die Erstellung von Dissemination Information Packages (DIPs) und deren Indizierung, um den Zugriff auf und die Volltextsuche in Informationspaketen zu ermöglichen.

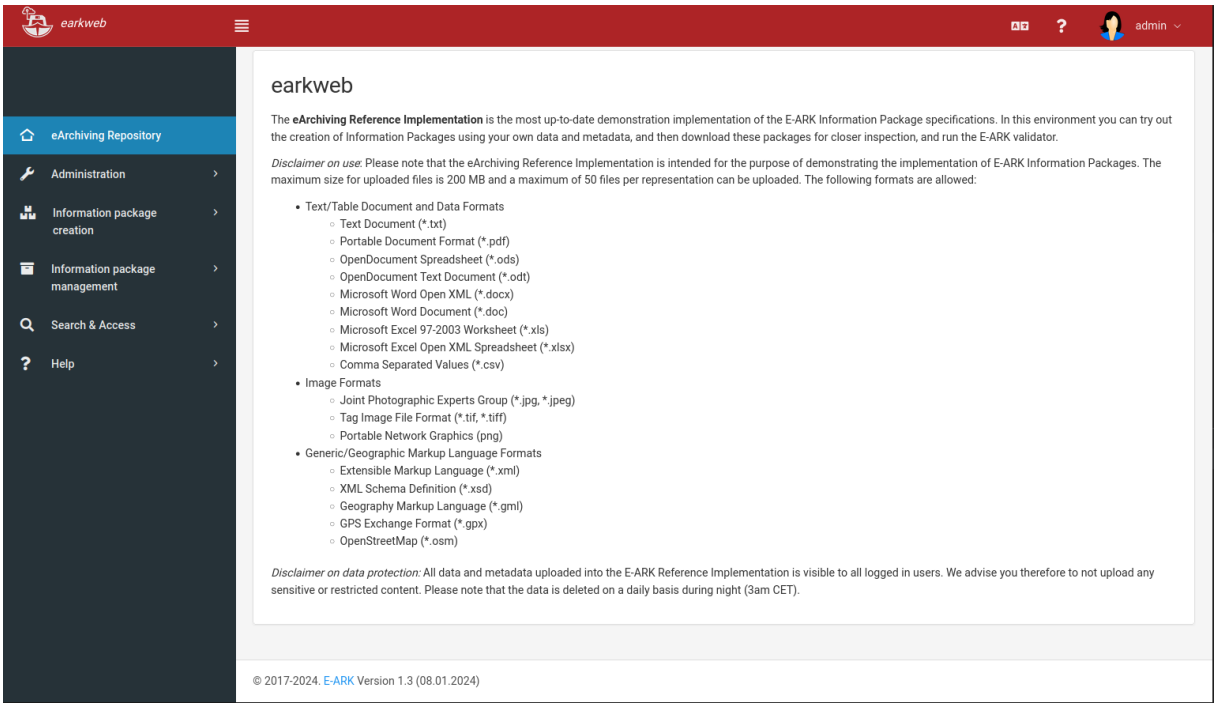

Abbildung 1 Startseite der Web-Anwendung earkweb

Abbildung 1 zeigt die Startseite nach dem Einloggen in das Repository-System mit der Navigationsleiste auf der linken Seite, die die Hauptfunktionen für die SIP-Erstellung

[1] Consultative Committee for Space Data Systems (CCSDS), ISO standard 14721:2012, https://public.ccsds.org/pubs/650x0m2.pdf

[2] https://digital-strategy.ec.europa.eu/en/activities/earchiving

[3] https://dilcis.eu/specifications/sip

(Erstellung von Informationspaketen), den Zugriff auf den Archivspeicher (Verwaltung von Informationspaketen) und den Zugriff (Suche und Zugriff) anzeigt.

Technische Merkmale

earkweb bietet ein Web-Frontend zusammen mit einem skalierbaren Aufgabenausführungssystem auf Basis von Celery[4] mit den folgenden vorteilhaften Eigenschaften:

- Das Backend für die Aufgabenausführung ermöglicht die Verteilung von Aufgaben auf mehrere Worker Nodes, was eine parallele Verarbeitung und eine effiziente Ressourcennutzung ermöglicht.
- Bei zunehmender Arbeitslast können zusätzliche Worker hinzugefügt werden, um ein größeres Volumen an Aufgaben zu bewältigen.
- Es bietet integrierte Mechanismen für die Behandlung von Aufgabenfehlern und Wiederholungsversuchen, die sicherstellen, dass Aufgaben auch bei Fehlern oder Ausfällen erfolgreich abgeschlossen werden.
- Die asynchrone Ausführung von Aufgaben ermöglicht die Freigabe von Anwendungsressourcen für die Bearbeitung anderer Aufgaben, während langlaufende oder ressourcenintensive Aufgaben im Hintergrund verarbeitet werden.
- Aufgaben können nach ihrer Wichtigkeit oder Dringlichkeit priorisiert werden, um sicherzustellen, dass kritische Aufgaben umgehend bearbeitet werden, während weniger kritische Aufgaben für eine spätere Ausführung in eine Warteschlange gestellt werden können.
- Das System integriert Tools zur Überwachung der Aufgabenausführung, zur Verfolgung des Aufgabenfortschritts und zur Verwaltung von Arbeitsknoten, die eine effektive Überwachung und Optimierung der Aufgabenverarbeitungs-leistung ermöglichen.

Die Aufgabenausführung kann über eine REST-API gesteuert und überwacht werden, ohne dass das Web-Frontend verwendet werden muss.

Der Ingest-Prozess ist als eine Reihe modularer und erweiterbarer Backend-Tasks implementiert. *earkweb* bietet auch einen vordefinierten Workflow für die Batch-Verarbeitung, der die gesamte Kette von Tasks für den vollautomatischen Ingest von großen Datenmengen ausführt.

Die *earkweb*-Anwendung ist eine Python/Django-basierte Software, die eine MySQL-Datenbank zur Speicherung von Informationen über Datensätze und ein Celery/RabbitMQ/Redis-Backend für die asynchrone Aufgabenverarbeitung verwendet. Eine docker-compose-Konfigurationsdatei[5] ermöglicht das einfache Einrichten einer

[4] http://www.celeryproject.org/

[5] https://gitlab.com/datamarket/conduit/blob/master/docker-compose.yml

lokalen Instanz der Anwendung, um die Funktionen zur Erstellung, Paketierung und Speicherung von Datensätzen zu testen.

Die *earkweb*-Anwendung ist für die Container-basierte Bereitstellung auf Basis von Docker[6] vorbereitet, um eine einfache und modulare Installation der Software in Cloud-Umgebungen zu unterstützen.

Docker ist eine Open-Source-Engine, die die Bereitstellung jeder Anwendung als leichtgewichtigen und portablen Container automatisiert, der auf jeder Plattform läuft, auf der die Docker-Engine unterstützt wird.[7] Um das Deployment von Diensten auf einer Docker-Plattform zu ermöglichen, wurden Docker-Container für die einzelnen Dienste des Frontends und Backends von earkweb erstellt.

Abbildung 2 gibt einen Überblick über die für das Deployment verwendeten Container. Jede Komponente mit einem "Blue Whale"-Symbol steht für eine Komponente, die als Docker-Komponente verfügbar ist.

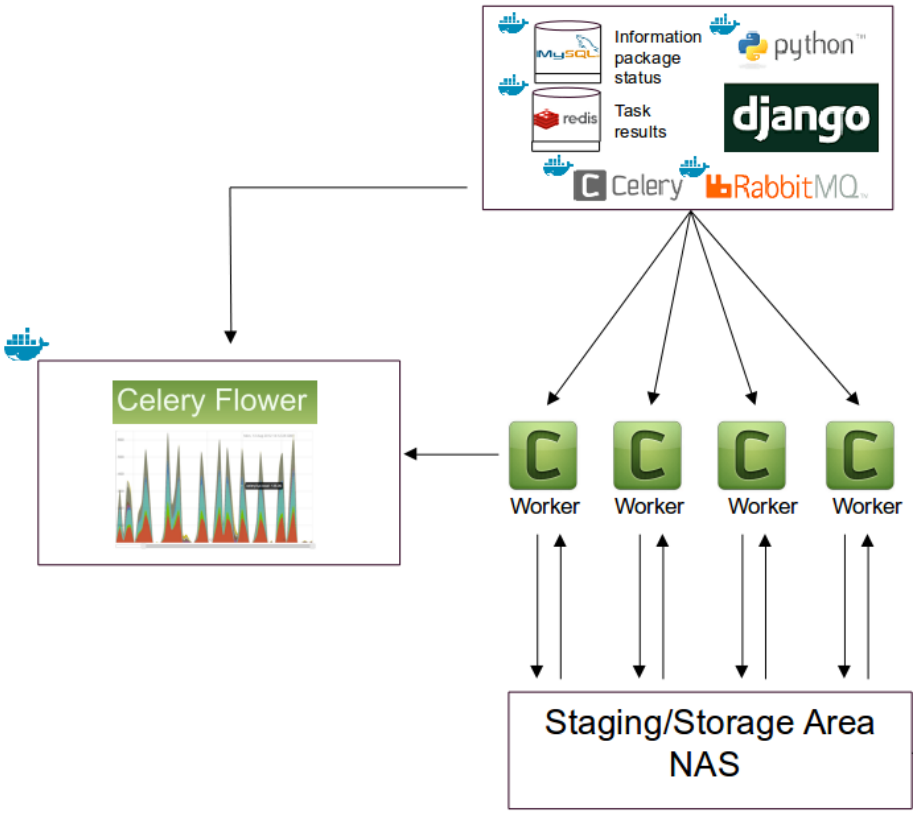

Abbildung 2 Übersicht der earkweb-Komponenten

[6] https://www.docker.com

[7] https://docs.docker.com/engine/installation

earkweb basiert auf den folgenden Container-Komponenten:

- MySQL[8]

- SolR[9]

- RabbitMQ[10]

- Redis[11]

- earkweb[12]

- Sellerie[13]

- Celery Flower[14]

Der Ingest-Prozess besteht aus einer Reihe von Einzelaufgaben, die in einer bestimmten Reihenfolge ausgeführt werden, um E-ARK-Submission-Information-Packages (SIPs) in E-ARK-Archival-Information-Packages (AIPs) umzuwandeln. Es handelt sich um einen erweiterbaren Workflow, der durch das Einfügen neuer Aufgaben an jedem beliebigen Punkt des Workflows an spezifische Bedürfnisse angepasst werden kann. earkweb verwendet einen modularen Ansatz für die Definition von atomaren Aufgaben, die einen bestimmten Transformationsschritt des Ingest durchführen, wie z.B. die Extraktion eines SIP oder die Validierung der darin enthaltenen beschreibenden Metadaten. Eine spezifische Aufgabe führt jedoch nicht notwendigerweise eine einzelne Aktion aus, sondern kann auch eine Reihe von Aufgaben oder einen kompletten Workflow initiieren.

[8] http://www.mysql.com

[9] https://lucene.apache.org/solr

[10] http://www.rabbitmq.com

[11] http://redis.io

[12] http://github.com/eark-project/earkweb

[13] http://www.celeryproject.org

[14] https://github.com/mher/flower