Software-Development at the MDZ

04.04.2019, Marcus Bitzl and Ralf Eichinger, DB/MDZ/IWA
Software developed at the MDZ

- Webapps / Viewer
- Data-Processing / Workflows
- APIs / REST-Servers
- Reusable components / libraries
Webapps / Viewer

- Technologies: Java (Spring Boot), HTML/CSS/Bootstrap 4, JavaScript, React
- Nginx-LoadBalancer/-Caching,
- Debian-Linux-Server

- Viewer: Mirador ([http://projectmirador.org/](http://projectmirador.org/))

- cudami Object Management and CMS
  (OpenSource at [https://github.com/dbmdz/cudami](https://github.com/dbmdz/cudami))
Data-Processing / Workflows

- Technologies:
  Java, Python
  Rabbit MQ (Message Queue)
  Debian-Linux-Server

- Flusswerk-Workflow-Engine (OpenSource at GitHub: https://github.com/dbmdz/flusswerk)
APIs / REST-Servers

- Technologies:
  - Java (Spring Boot)
  - neo4j (Graph database), Solr (Search Server), PostgreSQL (Relational database)
  - Debian-Linux-Server

- Hymir IIIF-Server (OpenSource at GitHub: https://github.com/dbmdz/iiif-server-hymir)
- Euphoria Streaming Server (OpenSource at https://github.com/dbmdz/streaming-server-euphoria)
- DZP (Data Access Platform) (Inhouse)

You can access the IIIF API endpoints of the Bavarian State Library at the following endpoints:

- Presentation API: https://api.digitale-sammlungen.de/iiif/presentation/v2/{object_id}/manifest
- Image API: https://api.digitale-sammlungen.de/iiif/image/v2/{image_id}
Reusable components / libraries

• Technologies: Java, C, Python

• OpenSource at GitHub: https://github.com/dbmdz

• Examples
  • efficient OCR-Storage Solr-Plugin
  • JPEG2000, TurboJPEG image processing
  • File access libraries
  • IIIF-presentation and -image libraries
Software Development Process

- Continuous Integration:
  - Local Java development with OpenJDK 11, Apache Maven Buildtool and IntelliJ/Netbeans IDEs
  - Code in Git (GitLab/GitHub)
  - CI-Builds: GitLab CI
  - Quality-checks with Maven plugins (e.g. checkstyle, spotbugs) and Quality tools: dependabot, SonarQube
  - Built software-artifacts in Nexus Repository
IIIF Production Process

- Images from scanning to Web
Lessons Learned

- **Legacy** architecture dictates a lot of design decisions: Store everything on the NAS, JPEG as the source format

- Performance bottleneck is often not the image transformations, but the transfer from network storage → **Caching** is important at every layer, not just the frontend

- Generating new formats for hundreds of thousands of objects can lead to some nasty surprises: **Corrupted/Invalid/Duplicate legacy data**…

- A common, **standardized API for serving digitized objects** is a game changer → “IIIFify all the things”, reduced maintenance costs, better monitoring, easier sharing, collaboration with the community
IIIF Roadmap

- Provide IIIF access to all copyright-free objects (around 2 Mio.)
- Define and add machine-readable IIIF core metadata to make filtering (e.g. in Bookshelf webapp) and discovery easier
- Expose IIIF Content Search API endpoints for all OCRed objects
- Introduce collections to group all manifests semantically
- Offer support for annotating content, both privately and publicly
- Improve and maintain OpenSource products: Hymir IIIF Server, Java IIIF Image and Presentation APIs, IIIF Bookshelf
- Continue with participation in development of Mirador
Links

Web Services
• BSB IIIF Portal: https://iiif.digitale-sammlungen.de
• IIIF Presentation API endpoint:
  https://api.digitale-sammlungen.de/iiif/presentation/v2/{object_id}/manifest
• IIIF Image API endpoint:
  https://api.digitale-sammlungen.de/iiif/image/v2/{image_id}

BSB Open Source Products
• Java-Library IIIF Image API: https://github.com/dbmdz/iiif-image-api
• Java-Library IIIF Presentation API: https://github.com/dbmdz/iiif-presentation-api
• IIIF Server Demo: https://github.com/dbmdz/iiif-server-demo
• IIIF Server “Hymir”: https://github.com/dbmdz/iiif-server-hymir
• IIIF Bookshelf Webapp: https://github.com/dbmdz/iiif-bookshelf-webapp
• BSB Mirador-Plugins: https://github.com/dbmdz/mirador-plugins