ADA — Phases of Development:
Proof of Concept to Community Handover

– an interoperable framework for microservices in scholarly publishing

Advanced Document Architecture (ADA)

A Software Paper

Software repository: https://github.com/consortium/ADA_infra

Authors: Johannes Amorosa (Developer, Endocode); Lisa Nöth (CEO, Endocode); Simon Worthington – correspondence author simon.worthington@tib.eu (Open Science Lab, TIB)

Open Science Lab, TIB – German National Library for Science and Technology: https://tib.eu/osl

Endocode AG: https://endocode.com/

Publication information

Type: software paper

Title: ADA — Phases of Development

Subtitle: Proof of Concept to Community Handover

Authors: Simon Worthington, Johannes Amorosa, & Lisa Nöth

Description: Advanced Document Architecture (ADA), a software paper. an interoperable framework for microservices in scholarly publishing to introduce a P2P economy in research publishing for institutional takeback of infrastructures.

DOI: 10.20389/q6t3-4h29

Published: Self published. v01. Hannover, Germany. On GitHub, hash #

Cite as: ADA — Phases of Development. Simon Worthington, Johannes Amorosa, & Lisa Nöth. Published 2019 via Publishing Consortium. DOI: 10.20389/q6t3-4h29

Keywords: microservices, scholarly publishing, research, infrastructure as code, Apache Kafka, P2P, open science, open access, infrastructure, future publishing, Kubenetes, DevOps, deployment, economy

Abstract

The motivation behind the ADA software is to enable technological approaches to support research publications and related artifacts in the research cycle to become interoperable and free from restrictions placed on ‘use’ by systems provided by large scholarly publishers.

ADA is a publishing microservices framework intended to lower the costs of publishing by automating publishing processes and drastically speed up software development.

The technology employed for the ADA microservices are automatic deployment using Kubernetes clusters and content distribution using Apache Kafka data streaming. Microservices are a cornerstone of DevOps in commercial computing, where the FOSS suppliers have come to dominate business IT infrastructure provision, driving out closed source providers on price and innovation. ADA has a mission to support a similar change over in academe, with a vision that institutions use such cost savings to take back service provision and infrastructure in-house.

ADA also aims to accelerate P2P networking of the wide array of existing FOSS publishing platforms to disintermediate the publishing market away from existing scholarly publishers who are not in the business of innovation. To support this P2P approach ADA proposes a metering service to allow for the payment for service provision by platforms.

The ADA partners are building a working example of the microservice system as a proof-of-concept for community handover as a Kubenetes cluster and cloud deployment on Google Cloud Product (GCP). Example core components would be: services connector; multi-format document transformer; document validator; pipelines; metering and billing of services; and cryptographic IDs.

ADA is takes inspiration from and sits alongside other ‘liberation technologies’ for research which are working on future publishing systems: Open Research Knowledge Graph; Solid and Dokieli; Project Jupyter; Open Knowledge Maps; or as in Herbert van de Sompel’s presentation ‘Scholarly Communication: Deconstruct & Decentralize?’; and as well as from support initiatives, such as: Invest in Open Infrastructure, or Research Software Engineering organisations such as De-RSE.

About ADA

The technology problem space of ‘scholarly publishing’ that the ADA project is approaching is the much needed technical improvement in the creation and use of research content which is being held back by current technology suppliers and stakeholders. Much of the problem is that content tends to *not* be interoperable or machine interpretable. Many problems are at play in ‘scholarly publishing’ but one important strand where ADA is designed to play a role is in system development processes where a **root and branch overhaul** is needed.

$2.2 trillion is spent on global research & development each year (Tennant 2019) the majority of the research publishing produced is not available to the public and stakeholders that could make use of it.

Mission – to contribute to changing scholarly publishing by enabling efficient, accelerated, and open exchange of research.

We do this in two ways:

- firstly, create a **scholarly publishing services connector** — the ADA Module — with metering to support a variety of economic models — free, subsidised, or by direct payment, and;
- secondly, by the use of **automatic deployment and Infrastructure as Code (IasC)** to increase the easy adoption of new scholarly software and services, and speed up system development innovation.

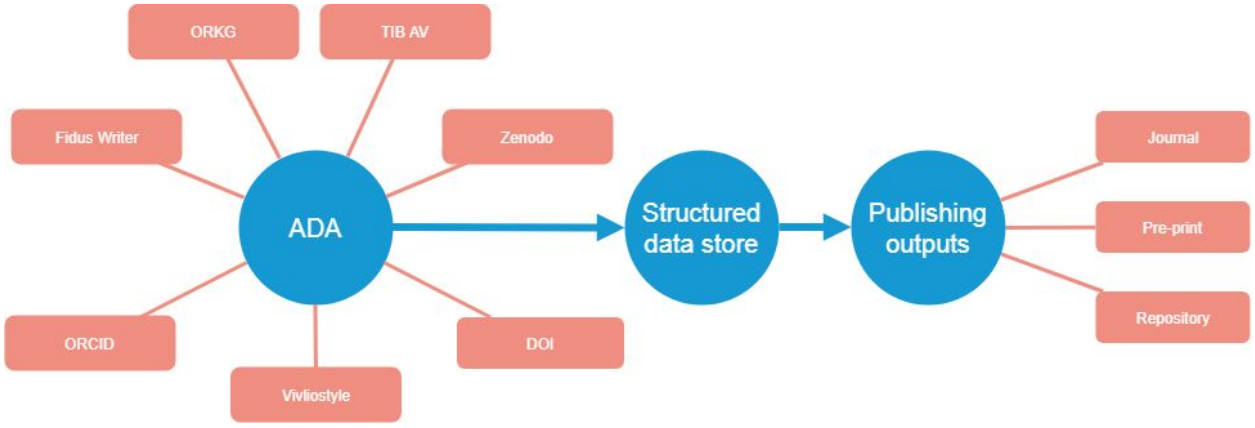
Objective: Create a demand for 'processing' and 'information' services by making 'foldable infrastructures' and 'development methodology' that institutions can adopt and integrate, as well as being able to add their own components to the infrastructure.

ADA is an industry and academic R&D consortium led by TIB – German National Library for Science and Technology. Members include: Endocode AG, Le-tex GmbH, Fidus Writer, and the Vivliostyle organisation.

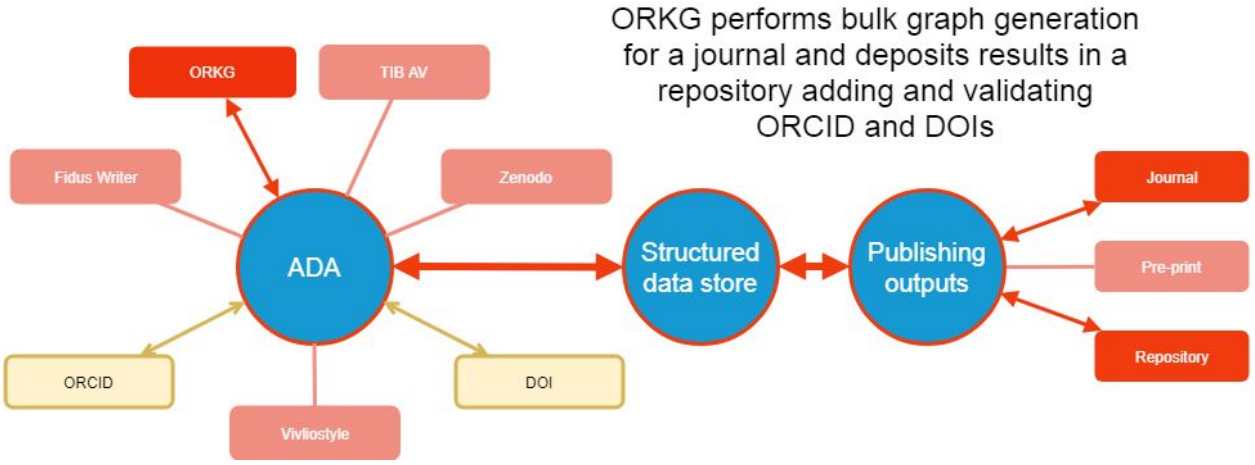
Example ADA use cases

Researcher publishing workflow

https://genr.eu/wp/making-research-workflow-open-source/



An example service: Open Research Knowledge Graph (ORKG)



Note: ORKG performs bulk graph generation for a journal and deposits results in a repository adding and validating ORCID and DOIs.

ADA features

ADA is for connecting multiple services in academic publishing workflows to process document parts, for example format validation or document delivery, or for adding information, such as semantically enriched videos or author information from ORCID IDs.

Modules

Admin Module

- **Services connector** for document and publication production and management, via API, e.g., semantic graph enrichment (ORKG), DOIs, semantic video media, semantic document structuring, and validation against standards, etc.
- **Customisable multi-format document transformer** for publication outputs and document types.
- **Customisable document validator** with feedback and interactive feedback GUI
- **Customisable workflows and pipelines** using Transpect
- **Bootstrap *like* multi-format design** facility for publications typesetting designs
- **Metering and billing of services**
- **Automatic deployment** – Infrastructure as Code: CI, CD, cloud and Kubernetes Cluster

User Module

- **User ownership, storage, and tracking of content** with cryptographic IDs

ADA Document configuration

- **Develop a master document storage configuration** based on Le-tex HUB
- **Support the JATS Dar Document sharing format** – Dar stands for (Reproducible) Document Archive (*Dar: Reproducible Document Archive* [2017] 2018)
- Be document format agnostic
- To create interchangeable document configurations

- Adhere to division of concerns in document standardization
- Support encoding of advanced document elements: semantic graphs, Jupyter elements, semantically enriched video media
- Be **Publication Ready Outputs** (PROs) aware and support validation. PROs is a Consortium term for validating against all aspects of a chosen set of publication output targets, e.g., typesetting, metadata, to accompanying data deposits.

Functionality

- Real time validation and processing
- Automatic infrastructure deployment
- Infrastructure as Code methods applied to Consortium members and scholarly tools. This means moving infrastructural functions from software to infrastructure: user ID, authentication, ACLs, storage, etc.
- Encouraging machine readable content practices
- Enable Consortium member products to be interoperable and IasC compatible

Systems and services ADA could connect with

Different web software packages that could be installed with ADA and be used with ADA, e.g.,

- OJS/OMS
- Pads e.g., CrypPad
- Static website generators e.g., HUGO
- Mattermost
- Mastodon
- Jupyter Notebooks
- MS Office Word plugin
- LibreOffice Writer plugin
- Atom

TIB related services to connect to:

- ORKG
- DataCite
- TIB AV video
- ORCID
- NOA
- VIVO
- OJS/OMS

Phases of development

The goal is to have an executable playbook to fold up an entire infrastructure deployment consists of frontend, backend with all required business logic. There will be a well documented Jupyter Notebooks for easy installation, and to demo TIB service APIs also running in the Jupyter Notebook, with precise operational steps to recreate ADA. For the initial version of ADA we will use as much GCP managed services as possible to speed up the development. In later phases a platform agnostic installation targets based on Kubernetes deployments will be implemented.

Phase A: Core

In this phase we will focus on getting a software deployment framework based on best practices inside the GCP platform. This consists of setting up a basic infrastructure partly with mockup

services and basic modules. This is the fundament of a continuous testing and deployment pipeline which speeds up development.

Fundamentals

- Setting up GCP organisation
- Git based hosting of code with Google Container builder infrastructure
- Hashicorp's Terraform or similar infrastructure deployment in GCP
- Programmatically configuration with Terraform of project, IAM, certificates, authentication, build environment, networks and DNS

Initial services for PoC

- Setting up Kubernetes Container Runtime (GKE) with 3-5 nodes as staging environment
- Database backend for Fidus Writer in CloudSQL
- Kubernetes deployment of Fidus Writer with ingress rules
- Setting up a mockup serverless service in GCP App Engine or cloud functions with complete lifecycle for developing event based Transpect document transformation
- Setting up a mockup metadata enrichment API inside Kubernetes with complete lifecycle
- Testing different persistence layer for document and metadata storage

Phase B: Persistence and services

Additionally for refining the infrastructure we will implement the data persistence layer and implement alpha versions of the Transpect document transformation pipeline and the metadata enrichment API.

Data persistence layer

- Implementing the data infrastructure based on PoC phase. Candidate is Apache Kafka. (Svingen 2017)
- Based on PoC phase and the requirements a Storage API is needed. This API could be implemented with GCP managed services (Apigee)

Software development

- Implementation the Transpect document transformation queue and a reference client
- Create metadata enrichment api endpoint and pipeline. DOIs/PIDs, subject semantic graph data. E.g., TIB AV Portal, ORKG for textual research paper documents, NOA for images, Leibniz Data Store, Datacite DOIs, and ORCID IDs, etc.

Phase C: Jupyter Notebooks

Jupyter Notebooks components: 1. To run ADA, 2. To demo TIB APIs. TIB APIs can be run as a demo and trigger APi actions from Jupyter Notebooks in the browser, like: graph my document and draw diagram of its RDF structure with Open Research Knowledge Graph; issue my doc DOIs, connect DOIs to author ORCIDs, deposit my associated data in Leibniz Research repository, Zenodo Sandbox, etc.

Phase D: Rinse and repeat

In this phase we will focus on refining the infrastructure with an additional testing and production environment. We will re-evaluate deployment, monitoring and metrics options to fulfill a complete lifecycle for production usage and administrative views for account management.

Refactoring infrastructure

- Implementation of test, staging, production environment
- Improve monitoring, implement disaster recovery procedures, improve availability

Integrate Administrative Views

- Development and Integration for administrational views for billing, Account management, multi-tenancy

Phase E: Modularize

In this phase we will consolidate the deployment to decrease operation costs and improve modularization and platform agnostic operation. This is useful for hybrid cloud installations or on a on-premise cluster.

Platform agnostic conversion

- Implement Kubernetes hosted serverless functions with Knative
- Convert GCP managed services like GKE to platform agnostic counterparts

Phase F: Reference Frontend implementation

Additionally to the reference commandline implementation of an ADA client we will concentrate to merge the web based reference frontend implementation as part of the deployment.

Integrate reference frontend

- Interaction between front and backend
- UI/UX design and implementation

Phase G: External identity services

The main focus in this phase is to further integrate external identity services into the ADA ecosystem. There is a broad ecosystem which is worth implementing. We will focus on the most viable service which has the most impact on the framework.

Integrate external services

- Use external identity management services like ORCID, Eduroam, WebID or DID utilizing Sovrin.

Phase H: External data sources

For special purpose workflows for example: interacting with other data sources or data sinks additional microservices needs to be implemented. There is a wide variety of potential data

sources and sinks which are used in a scholarly or academic context. We will evaluate the most viable contender and integrate these into the ADA framework

Phase X: Hand over to community

ADA is open source and a community should take over leadership. ADA is an important infrastructure to enable the scholarly community to make use of the Open Science and Open Access dividend coming down the line and urgently needs to be created.

References

Auer, Sören. “Towards an Open Research Knowledge Graph.” *Zenodo*, January 22, 2018. https://doi.org/10/gfv3rv.

“Solid: Social Linked Data | MIT CSAIL.” Accessed October 21, 2018. https://www.csail.mit.edu/research/solid-social-linked-data.

“Dokieli.” Accessed March 27, 2018. https://dokie.li/.

“Project Jupyter.” Accessed December 6, 2017. http://www.jupyter.org.

Maps, Open Knowledge. “Open Knowledge Maps - A Visual Interface to the World’s Scientific Knowledge.” Open Knowledge Maps. Accessed November 27, 2018. https://openknowledgemaps.org/.

Van de Sompel, Herbert. *Scholarly Communication: Deconstruct & Decentralize?* Washington, DC: CNI: Coalition for Networked Information, 2017. https://www.youtube.com/watch?v=o4nUe-6Ln-8.

“Home.” Invest in Open Infrastructure. Accessed August 8, 2019. https://investinopen.org/.

“De-RSE.Org - Research Software Engineers (RSEs) - Responsible for Scientific Software.” Accessed August 8, 2019. https://www.de-rse.org/de/.

Svingen, Boerge. 2017. ‘Publishing with Apache Kafka at The New York Times’. Confluent. 6 September 2017. https://www.confluent.io/blog/publishing-apache-kafka-new-york-times/.

Tennant, Jon. 2019. ‘These Figures Always Blow My Mind. We Spend $2.2 Trillion on Global Research & Development Each Year. Yet, We Can’t Find a Better Way to Communicate the Outputs than by Outsourcing the Sharing of Our Knowledge to Commercial Third Parties. We Can, and Must, Do Better. #seed2019pic.Twitter.Com/5Kwl5tYDdT’. Tweet. *@Protohedgehog* (blog). 25 February 2019. https://twitter.com/Protohedgehog/status/1099973052710965248.

Dar: Reproducible Document Archive. 2017. Reprint, Substance, 2018. https://github.com/substance/dar.