CAA2016
Oslo 30 march-3 april

Oceans of Data

Proceedings of the 44th Conference on Computer Applications and Quantitative Methods in Archaeology

Edited by

Mieko Matsumoto and Espen Uleberg

CAA
Computer Applications & Quantitative Methods in Archaeology

Archaeopress Publishing Ltd
Summertown Pavilion
18-24 Middle Way
Summertown
Oxford OX2 7LG

www.archaeopress.com

ISBN 978 1 78491 730 2
ISBN 978 1 78491 731 9 (e-Pdf)

Language Editing: Rebecca Cannell
Cover Design: Bjarte Einar Aarseth

Printed in England by Oxuniprint, Oxford

This book is available direct from Archaeopress or from our website www.archaeopress.com

Contents

Mieko Matsumoto and Espen Uleberg (eds) 2018. *CAA2016: Oceans of Data*
Proceedings of the 44th Conference on Computer Applications and Quantitative Methods in Archaeology.
Oxford: Archaeopress

When Data Meets the Enterprise: How Flanders Heritage Agency Turned a Merger of Organisations into a Confluence of Information

Koen VAN DAELE, Maarten VERMEYEN, Sophie MORTIER, and Leen MEGANCK

Flanders Heritage Agency

Corresponding author: koen.vandaele@vlaanderen.be

Abstract
In 2012 Flanders Heritage Agency was created as a central agency dealing with immovable cultural heritage — broadly defined as archaeology, architectural heritage and cultural landscapes — in Flanders. Prior to this, tasks of this agency were carried out by several independent agencies. The merger created a very heterogeneous set of business processes, IT-components and systems. Together with new heritage legislation, this prompted a re-evaluation of these systems.
This paper will delve into our system architecture, built on a core separation of concerns between data-driven and process-driven applications. The resource oriented focus of REST services has served us well in creating datasets interlinked through URIs. From there it's only a small step to publishing the data through semantic technologies such as RDF and linking it with other, external, data sources.

Keywords: enterprise architecture, web services, linked data, open data, URI

The winds of change

The Flanders Heritage Agency (FHA) was created in 2012 as an agency of the Flemish Government that deals with immovable cultural heritage: broadly defined as archaeological heritage, architectural heritage and cultural landscapes. The main responsibilities of this agency are to maintain inventories of and protect heritage, to support the management and conservation of this heritage, to help define the policies surrounding this heritage, to conduct research about the effect of these policies, and to disseminate information about this heritage and the relevant policies. To this end it interacts with several different stakeholders: heritage owners, the general public, other Flemish government agencies, local governments, spatial planners, heritage professionals, etc.

Before 2012 these responsibilities were already a task of the Flemish Government, but they were carried out by different agencies (Figure 1). The main participants were called the Flemish Heritage Institute and Ruimte en Erfgoed. While both dealt with all types of immovable cultural heritage, the first mainly focused on heritage research (inventories, excavations, etc.) while the second handled the management, and conservation aspects and tasks such as excavation permits. Further back in history, 2004, two different organisations existed. At that time, the division was mainly about the type of cultural heritage they dealt with. The Monuments and Landscapes Agency handled all matters concerning architectural heritage and cultural landscapes; the Institute for Archaeological Heritage, handled archaeological matters.

These organisational changes have had an important impact upon the kinds of data that need to be recorded and the information systems used to record them. Every change has been a disruptive event that challenges existing opinions and offers opportunities for growth and innovation. We have previously detailed (Van Daele *et al.*, 2016) how this has affected the users of a single system (the heritage inventory). This paper is more concerned with the resulting overarching systems architecture that allows heterogeneous processes to interact in a sustainable way.

Since the creation of the Flanders Heritage Agency, other events have transpired that have been instrumental in defining the current state of information systems within the agency. First, and foremost, in 2013 new heritage legislation was voted in. This legislation is the first ever unified legislation for archaeology, architectural heritage and cultural landscapes. In 2015 it came into full effect, although most of the legislation concerning archaeology took until 2016 to be fully implemented. While the new heritage legislation built upon the preceding instruments, it also created new responsibilities and tasks for the agency, especially where archaeological heritage was concerned. De Roo, De Maeyer and Bourgeois (2015) offers a more in-depth analysis of the new legislation, although certain aspects have changed somewhat since then.

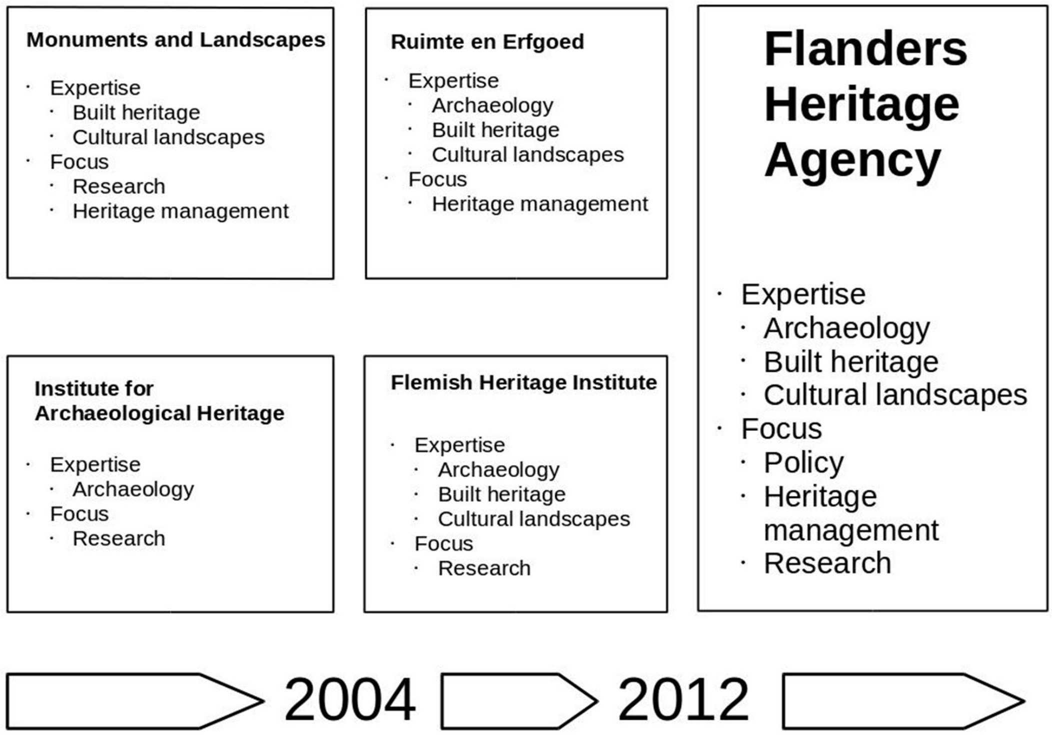

Figure 1. Evolution of heritage organisation in Flanders in the 21st century.

Other policy decisions of the Flemish Government have also had an impact. In 2015 a policy concerning open data was adopted. This stipulated that most government data should be available as open data under one of several possible licenses (Vlaamse Overheid, 2016). In the same year a special program of the Flemish Government called 'Radicaal Digitaal' was launched (Vlaamse Overheid, 2015). This project aims to digitise as many interactions as possible between citizens and their government by 2020.

A new beginning

When the Flanders Heritage Agency was created in 2012, it found itself the inheritor of two very different IT-environments. The differences in focus of the two previous agencies were very obvious both in the kind of data that was generally collected, curated and published and in the technologies used in building the systems to hold and process this data.

One agency, Ruimte en Erfgoed, dealt with both spatial planning and management of heritage. Its focus lay more with the heritage as a part of the spatial planning process, than with the heritage itself. Most of its work was process and workflow driven. There were applications for case-management such as requesting an excavation permit, a restoration grant or permission to modify a listed building. Most of the systems created by this organisation were inward facing. They were built by the agency for use by its own staff to help them carry out their work as efficiently as possible. Almost everything was done in one giant, rather old, case-management system. This system had originally been built for spatial planning, but afterwards heritage workflows had been integrated in the system. Since the system was not built with specific heritage purposes in mind, this was never an easy fit. Technologically speaking the organisation was mostly organised around proprietary software. Software was written in .Net or Java, generally with Oracle or SQL Server databases and Windows servers.

The second agency, the Flemish Heritage Institute, was an agency that mostly dealt with research on cultural heritage. Its work was much more data-driven. It carried out excavations, maintained a library, a depot and an archive and created heritage inventories. Since the latter were the focus of a lot of other processes, they received a good deal of attention. Although the inventories were created by the institute they were readily shared with the world through websites and databases. Most of them were located in a single inventory management system (Van Daele *et al*., 2016) that, while not as old as the case management system, by 2012 had received several major updates expanding the capabilities and complexity of the system. It was felt that it did too many things at once since the system also contained actors, events, images and thesauri. Technologically speaking the Flanders Heritage

Institute was favouring FOSS (Free and Open Source Software) and *nix (Unix or Linux) systems, using PHP as the main programming language and PostgreSQL and MySQL as the main database servers.

Fairly quickly after the merger, a decision was made regarding the technological stacks. The new agency adopted the preference for FOSS of the Flanders Heritage Institute. As can be evidenced from a study about the adoption of FOSS by the Flemish Government (Ven and De Bruyn, 2011), several factors can contribute to such a decision. For the FHA the reduced license costs were important, but not the only factor. Prior internal knowledge, the avoidance of vendor lock-in, the presence of boundary spanners and general ideology all played a part as well. For the servers of the new agency, the *nix and PostgreSQL technologies of the Flanders Heritage Institute were retained. The programming language of choice became Python. It's a good general purpose language, well-suited to the web and with a healthy support for GISwork. Contacts at the time between the Flanders Heritage Agency and the Getty Conservation Institute concerning the Arches project (Myers *et al.*, 2016) certainly played a part as well.

A new enterprise architecture design emerged as well. The focus was shifted from building a few big, monolithic systems toward building many smaller systems and integrating them. Such a Service Oriented Architecture (SOA) is very much focussed upon the interactions between the different systems. It's essential to define the service contracts i.e. the questions a service should be able to answer, rather than the service implementations i.e. where the service should get the answers from.

Types of applications

Within the new enterprise application architecture there is a distinction between the two types of applications (Flanders Heritage Agency, 2016a). This corresponds to the dichotomy between data-driven and process-driven. We refer to the first type as *Authoritative Sources*. An authoritative source is the one and only source for a certain piece of information. Its data should always be referenced, never copied. The data in these systems is very long-lived or even permanent. Its relevance does not greatly diminish by age. Queries done in these kinds of systems are very much about the data itself, e.g. 'List all sites that have a Gothic church'. Within the authoritative sources a further distinction can be made between primary and secondary sources. The primary sources are our core-business, such as heritage inventories or the register of accidental archaeological finds. The secondary sources support the primary ones, but are not our core-business. Some of them are not even maintained by Flanders Heritage. A prime example would be thesauri and controlled vocabularies. They are essential for querying the primary systems, but they are a means, not an end.

A typical authoritative source would be the decrees authoritative source.[1] This source contains legal documents that designate or alter the designation of heritage objects. In essence the source captures what type of decision was made, by what responsible entity, at what point in time and the documents that make that decision binding. Whenever an application needs to reference a decree, it refers to this authoritative source. The authoritative source itself refers to other authoritative sources where necessary, such as thesauri or an authoritative source of actors and agents involved in our information systems.

The second type of applications we refer to as *Process applications*. These applications guide one or more (internal or external) users through a business process. The applications and the queries executed are mostly concerned with the workflow, e.g. 'List all requests for an excavation permit that have to be handled within the next 5 business days'. Quite a lot of the data in these systems is of a more temporary nature than with the authoritative sources. While we distinguish these types of applications from the authoritative sources, they are actually closely integrated with them since the process application reads from and writes to several different authoritative sources.

A typical process application would be our Accidental Finds Process Application (Figure 2). All accidental archaeological finds by the general public need to be reported to FHA through an online form. Whenever such a report is filed, a workflow is started. This process-application keeps track of scheduling (every report needs to be handled in a certain amount of time) and parties involved (using the actors authoritative source). It tracks communications between FHA and the reporter (stored in the Mail authoritative source) and interacts with several GIS services and an external authoritative source for address data (CRAB) to provide location information. Sometimes the archaeologist handling the report decides a small excavation is necessary. In such a case, an excavation permit is automatically generated in the authoritative source for excavation permits. The process application offers our archaeologists, responsible for tracking this business process, an easy and intuitive interface by interacting with a lot of different authoritative sources.

Apart from the two types of applications we also maintain several different *components*. These are pieces of software that are not intended to be used as stand-alone applications, but as modular building blocks

[1] https://besluiten.onroerenderfgoed.be

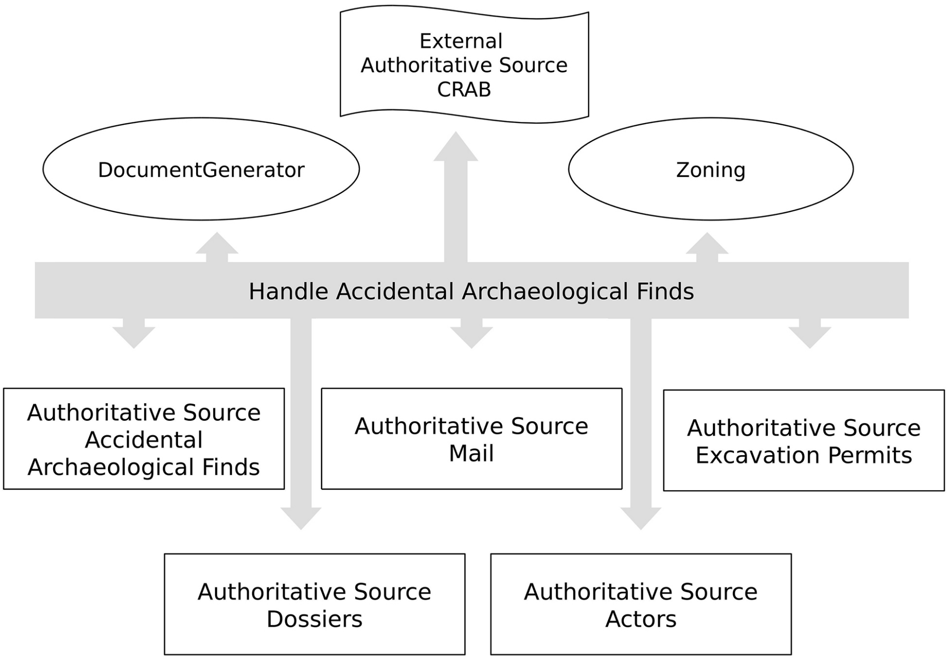

Figure 2. Handling reported accidental archaeological finds through the process application.

that help create authoritative sources or process applications. Not all of them have a user interface. One example is the DocumentGenerator. This service generates pdf documents based on a certain template and data sent to it. Our process-applications rely heavily on this service for generating the communications with other parties. This component has no user interface of its own. In contrast, a component that is all about the user interface would be our Zoning component. This is a Javascript module that allows a user to create a geometry by interacting with various services and components. In one application this is used to record the boundaries of an excavation report. In another, it is used to record the boundaries of a spatial planning application. The component itself does nothing more than interact with lots of different services and present results to the user. But it relies upon the application in which it is embedded to actually store the geometry it captures for the user.

Resource Oriented Architecture

When building a SOA, a key decision has to be made regarding the type of services to implement. Two main types of services exist. The first, Remote Procedure Call (RPC), consists of calling remote services that act similar to a function in a programming language. This style is exemplified by technologies such as SOAP and XML-RPC. A well know example would be the WMS or WFS GIS services. In recent years this type of service has become less popular and is often being replaced by Resource Oriented Architectures, exemplified by the REST paradigm (Fielding, 2000; Webber *et al*., 2010). In this paradigm, *everything* is a resource. Where RPC services focus on the actions to be undertaken, REST has a clear focus on the resource upon which the action is undertaken. The Flanders Heritage enterprise architecture is based heavily upon the REST paradigm. But we keep employing RPC services where deemed necessary and more suitable, such as the aforementioned WMS and WFS services. Within both RPC and REST services, the exchange format can be either XML or JSON. Within Flanders Heritage, JSON is the preferred format, mainly because it is very lightweight, taking up less bandwidth than a similar XML document and very easy to interact with for humans. Our services have become cornerstones of the new systems and the principal means for our own User Interfaces (mostly so-called Single Page Javascript applications) to interact with our data. This ensures that all services are thoroughly tested and used on a regular basis.

We define a resource as any information object we wish to describe. This can be tangible such as an object in a heritage inventory, a report or a person, or something less tangible such as an event or a decision. Every resource is uniquely identifiable by a Uniform Resource Identifier and is addressable on the World Wide Web through the HTTP protocol. A set of rules and guidelines (Flanders Heritage Agency, 2016b) was created for what is and what is not a good URI within our architecture. The first rule is that a URI should always identify a resource, never an action. The action to be undertaken should be coded by using the typical HTTP methods (GET, POST, PUT, DELETE). A URI such as https://

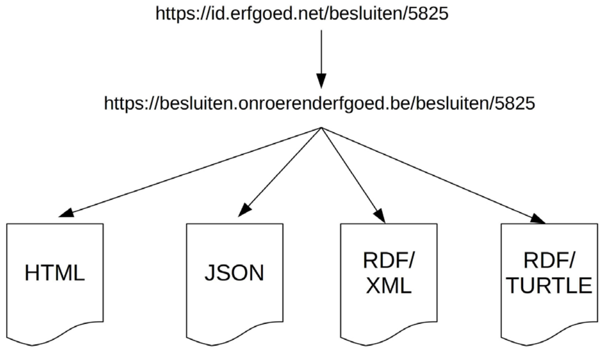

Figure 3. Redirecting from a resource URI to a document URI and serving different representation of this document through content-negotiation.

besluiten.onroerenderfgoed.be/besluiten/addNew.py is not allowed and should be POST https://besluiten.onroerenderfgoed.be/besluiten. A second very important rule is that a URI should be as meaningless as possible. Anything more than a very general resource type and some kind of identifier within that type is generally frowned upon. In the past we have observed a tendency to insert meaning into URIs, such as including a community name in the URI for a heritage object. This becomes problematic after a while when the community's name changes or when it turns out it was spelt incorrectly to begin with. This then leads to confusion because the heritage object is in a different community than that which its URI announces. Or the URI is changed to reflect the new situation, leading to broken links unless a careful redirect scheme is in place. Therefore, our URIs generally look like *https://id.erfgoed.net/<resourcetype>/<id>*. Other, more cosmetic, guidelines were put in place to remove anything technology specific from a URI (such as the .py suffix) and to always use plural style resource names.

Following the Cool URIs specification (Ayers and Völkel, 2008), we make a distinction between a URI for a resource or information object (e.g. an archaeological site) and a URI for the document about that object (e.g. a webpage about that site). The resource URI identifies a more abstract concept, the document URI a description of that concept. While we find this distinction to be relevant, it also allows us to keep our resource URIs very stable. We wanted to ensure that our resource identifiers were as permanent as possible. Our organisation name, and the corresponding domain name, has changed a few times in the past, and we foresee this might happen again. While there are other ways of handling the same need (May *et al.*, 2015), we have chosen to host our resource URIs on a separate domain that we control[2] but which is not tied to our organisation name. From such a URI we use a HTTP 303 response to redirect to the relevant document on a domain that does carry our organisation name.[3] This document could be a typical HTML webpage, a JSON object or a RDF representation (Figure 3). All these different documents offer us a similar, but not necessarily identical, view on an information object. An HTML webpage might show a map that can be navigated by a user. While the JSON representation might provide a GEOJSON representation of a geometry that needs to be visualised in some other way before it becomes useful to a human. Since all the document representations share the same URI, a mechanism is needed to select which representation is requested. This is done through a core HTTP mechanism called content-negotiation. A client requests a URI and includes an Accept header indicating what kind of representation it's looking for. The server analyses this request and sends back the representation that best matches the client's request.

Our URIs are being used everywhere, even on paper. Several of our applications run a process wherein our agency has to communicate with other parties such as archaeologists who have submitted a report or people who have submitted an accidental archaeological find. This could be a letter of acceptance or a letter stating that a certain process was started with some further information of what is expected of the other party. For legal reasons, most of these letters are still mailed out as paper letters. With these types of communication it's common to include an identifier to facilitate further communications. While previously this would have been a random piece of text, we are now using URIs directly in these letters. This makes it much easier to go from an analogue document (the letter) to the corresponding dossier. Previously, one would have had to go to a website or database and enter the random piece of text in a search form. Now it's a simple matter of taking the URI and entering it in a web browser. As long as the user has the necessary security credentials they will receive further information on the dossier.

Our new resource oriented architecture leads to a giant web of resources and looks very similar to the World Wide Web as experienced by humans (Figure 4) or the subject-predicate-object paradigm used in RDF. Every resource is identified by a URI. As noted, all URIs for the information objects are hosted on a different domain than the URIs for the documents about those objects. When viewing the web of resources, it is no longer clear which application is managing a certain piece of information. The client does not need to know where it needs to search for the document. It simply follows the URI and the URI server redirects the client to the correct document. When we decide to move a certain

[2] https://id.erfgoed.net

[3] https://something.onroerenderfgoed.be

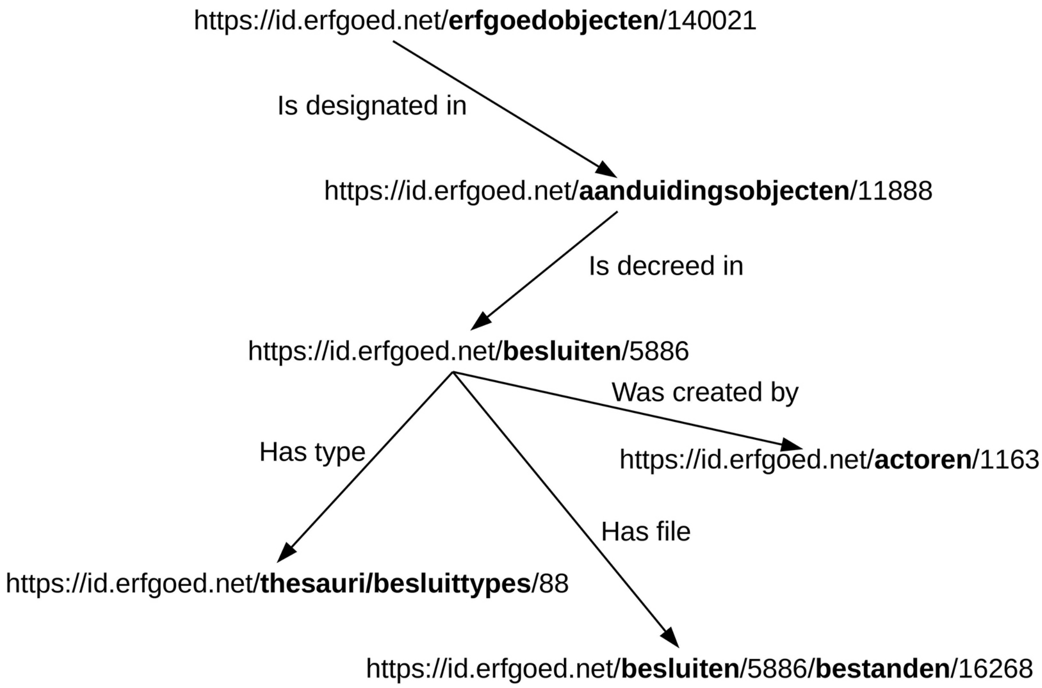

Figure 4. A small part of the web of resources detailing the links between a heritage object, a designation object and a decree.

document to a new application, we also update the mapping from the resource to the document. The client remains blissfully unaware. It just keeps on requesting the resource URI and gets a redirect to a new document URI. From a technical point of view this provides the application developers with a tremendous amount of flexibility. As long as we honour the agreed upon contracts, our architecture keeps on functioning.

Maintaining referential integrity

Every change offers new opportunities, but also creates new issues to be solved. While we have gained a lot of flexibility, we have also lost some convenient solutions to common problems. When all data is hosted in a single database, it's fairly simple to maintain referential integrity. By this we mean ensuring that all links between objects remain valid. If we assign 'person X' as the author of 'document Y' and a user tries to delete 'person X', we want to stop them (or at least warn them). A typical RDBMS like PostgreSQL is well equipped to handle this. But, when you spread out documents over several different systems, things become more complicated. Not every system uses the same database, some systems might use a different database technology (e.g. a NoSQL database) and some might not use a database at all. To alleviate this problem, we have moved the responsibility of handling referential integrity from the database layer to the application layer. Whenever a client asks a server to delete a certain resource, the server contacts a central registry. This registry receives a few parameters from the client and consults a set of servers to determine if a particular URI is in use somewhere. Every server responds with a clear yes or no answer and some additional information, such as how often the URI is being referenced, and a small list of its own resources that reference this particular URI. The central registry aggregates all responses of the different servers and communicates these back to the client who can then determine if the original request to delete a certain resource should be honoured or not. To help our editors, the same service can also be queried proactively. This allows them to check if a certain resource is still in use before actually trying to delete it. To implement this registry, a common JSON exchange format was created, as well as an open source implementation of the registry in Python (Flanders Heritage Agency, 2015). We'd like to emphasise that the exchange format is open to be used from several different platforms. It is currently being used to communicate between our newer Python applications and our older inventory management system written in PHP. Adding newer platforms is a simple matter of implementing a few services.

Publishing Linked Data

As mentioned before, government agencies are encouraged and even required to publish as much of their data as possible under an open data license. We follow the Flemish Open Data guidelines by publishing under the Flemish Free Open Data License. To make it

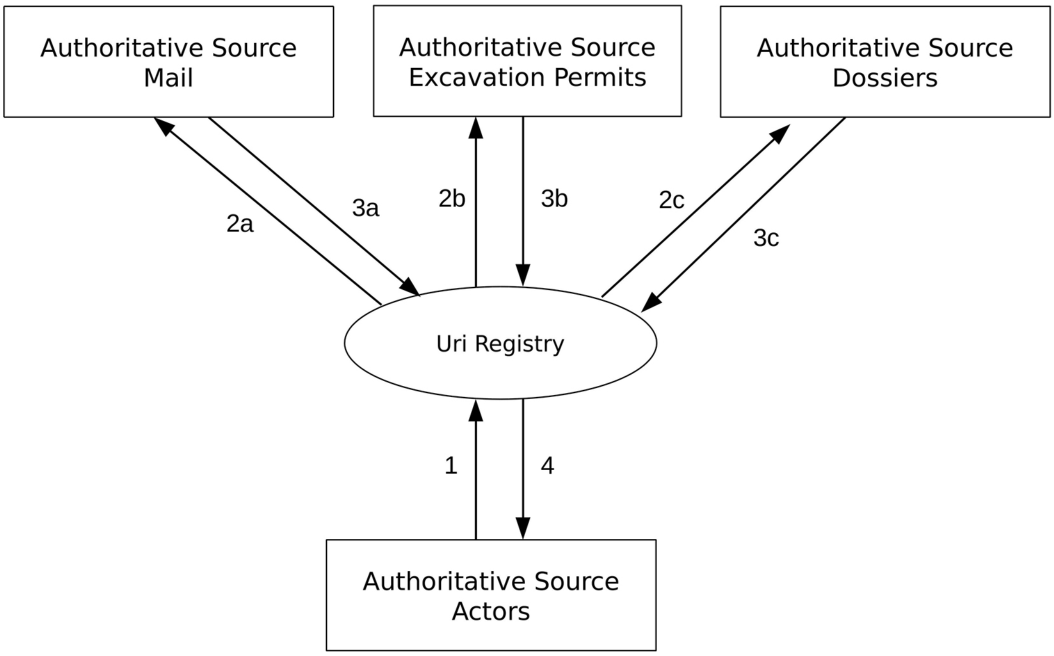

Figure 5. Evaluating referential integrity through the URI Registry. 1. Client asks the registry where a certain URI is being used. 2. Registry consults authoritative sources. 3. Authoritative sources reply. 4. Registry collects responses and replies to the client.

easier on external providers who do not operate within Flanders, we also publish as CC-BY 4.0.

From a resource oriented architecture it is a surprisingly small step towards publishing linked data (Berners-Lee, 2009) and RDF. In effect our architecture is built around the concepts of linked data in the broadest sense, by linking resources that can be dereferenced through URIs. Our standard HTML webpages and JSON services already adhere to this. If we look at linked data in a somewhat stricter sense, there is one key aspect missing. A JSON document for a heritage object might refer to the URI for a church in our thesaurus in its type field, but the type field itself has no URI that can be dereferenced.

All that remains to be done to publish our data as RDF data, is therefore to map fields in our service to predicates. We have started doing this for newer systems. Where possible we have reused existing vocabularies such as Dublin Core, FOAF, SKOS, etc. Most of our predicates come from these vocabularies, but where necessary we have created new ones (Van Daele, 2016). We have also created new classes for each of our resource types, mainly so we can use them in rdfs:domain and rdfs:range statements. In later iterations we plan to map our classes to existing classes in other ontologies. When publishing a certain resource as RDF, it is now very simple to publish this resource as a graph of triples. The resource becomes the subject of the triples. The attributes of the resource become predicates. And the values of those attributes either become literals when dealing with intra-resource information, or they become URIs to other resources when dealing with inter-resource information. These transformations are done within the application itself making them easy to maintain and very lightweight to publish. The RDF data does not live in a separate system requiring a separate Extract-Transform-Load (ETL) step. Because of this, our data is always guaranteed to be up to date.

Our services typically consist of two different types of documents. The most prevalent group contains documents about individual resources. Each of these documents represents a single resource with its attributes and relations to other resources. The document itself lives at a document URI,[4] that is the document about a certain resource, identified by a resource URI.[5] Our document can be retrieved in different serialisations through content-negotiation (Figure 6). The first serialisation, HTML, presents a web page that is meant to be read by the people interacting with a system. It tries to be as attractive as possible with little technical jargon. The other serialisations, JSON, RDF/XML and RDF/Turtle, are meant for machines.

Apart from these documents about a single resource, we also offer documents about a collection of resources.

[4] e.g. https://besluiten.onroerenderfgoed.be/besluiten/5825
[5] e.g. https://besluiten.erfgoed.net/besluiten/5825

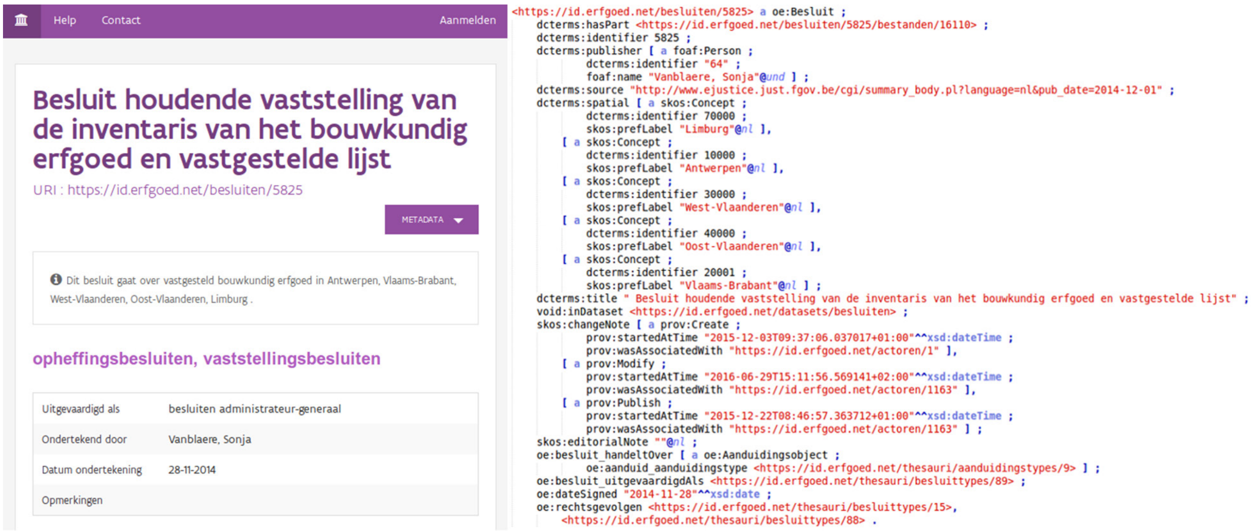

Figure 6. Different serialisations of one resource. On the left an HTML web page meant to be seen by a human user. On the right a Turtle document meant to be read by a machine.

These have a document URI, but not a resource URI. They are typically used in querying and searching. A URI such as https://besluiten.onroerenderfgoed.be/besluiten?beschermingstypes=https://id.erfgoed.net/thesauri/aanduidingstypes/1&sort=id produces the collection of resources that are decrees that have created listings of monuments, sorted by id. Again this document URI supports content-negotiation. In general, paged serialisations as HTML and JSON will be available. In certain cases, we also offer a serialisation as csv. The HTML version contains a typical search interface for human end users. The csv interface is meant to be downloaded by machines or humans for further processing. The JSON version is solely aimed at machines. At the moment, a collection like this cannot be serialised in an RDF format. We are considering implementing this by adopting the Hydra Core Vocabulary (Lanthaler and Gütl, 2013) when it reaches a stable enough state.

In a later stage we might add a triplestore and a SPARQL service to our architecture. At this moment we are still analysing how to implement this and whether this is actually a valuable addition to our setup. While we recognise that this provides some powerful querying features, this would also require setting up a separate infrastructure and creating a new ETL procedure that might become a burden to maintain. A fully functional SPARQL endpoint offers a lot of flexibility, but can also put a serious strain on resources. See Verborgh *et al.* (2016) for a full discussion on this topic. As an alternative we are following the work done on Linked Data Fragments and the Triple Pattern Fragments interface (Verborgh *et al.*, 2016) that offers an intermediary approach.

Spatial Data Infrastructure

Within Flanders Heritage we generally make very little distinction between spatial and non-spatial data. Whenever we build JSON serialisations of our data, we include our spatial data as GeoJSON. However, we have always realised the potential in GIS system for integrating very heterogeneous data sets. As indicated by the term immovable cultural heritage, our data sets have a very strong spatial component. Thus, a well-equipped Spatial Data Infrastructure (SDI) is needed. Just as in other countries (McKeague *et al.*, 2012), most of our data sets have been published under the Annex I Protected Sites theme according to the INSPIRE directive. We publish Open Geospatial Consortium (OGC) services on our own SDI node that gets harvested by INSPIRE through the intermediary Mercator and Geopunt SDI nodes.

It has been noted before that there is no good application schema for cultural heritage within INSPIRE (Uriarte González *et al.*, 2013). We have worked around this limitation by using our network of linked data, adding the URI attribute of the corresponding resource to the features returned by our geographic services. This enables a client to easily request further data on a certain object. While we still publish other attribute data through our geographic services, we see these as a mere convenience. Within our geoportal,[6] we integrate our own WMS layers with other INSPIRE services (Figure 7). This geoportal is custom built to provide a tight focus on cultural heritage and uses our URIs where possible. Whenever a user clicks on the map they get a report of all features at this location. By

[6] https://geo.onroerenderfgoed.be

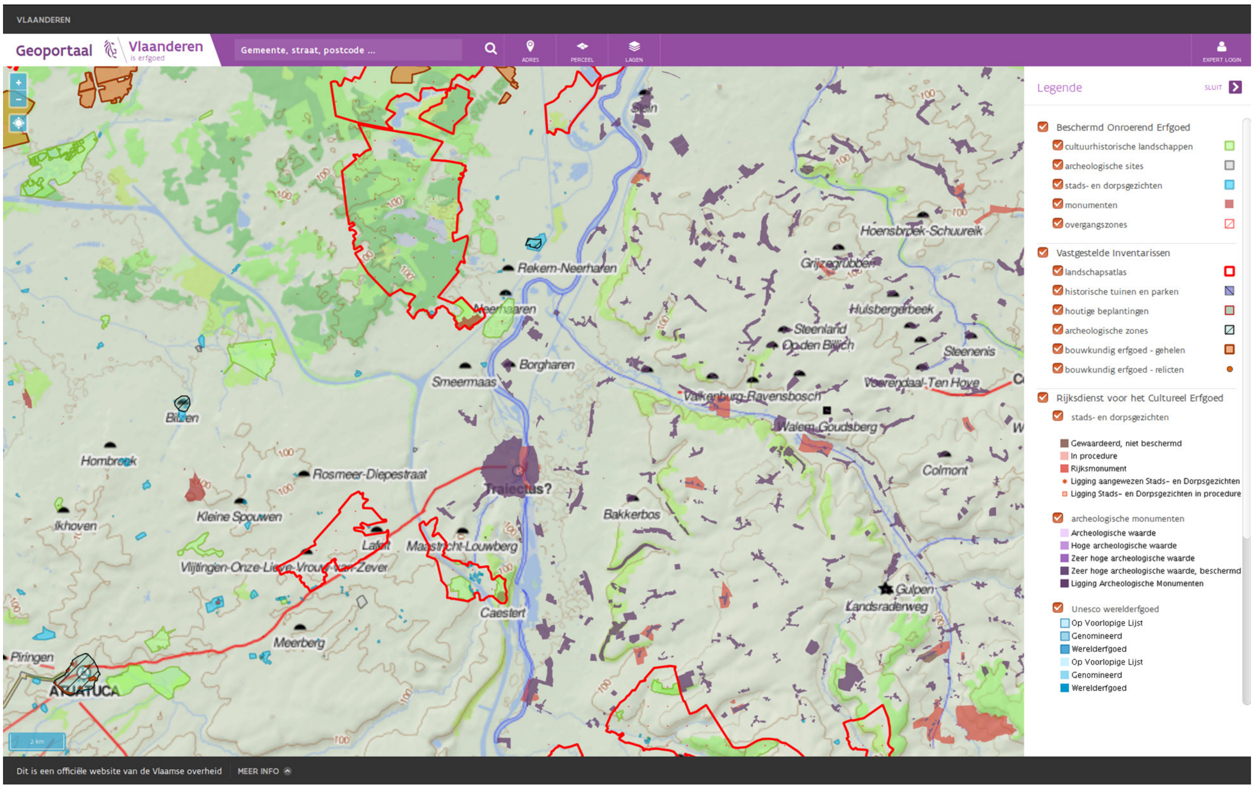

Figure 7. The geoportal combines internal and external layers. This example shows a map of the Roman Empire provided by Pelagios, combined with FHA layers (on the left) and RCE layers (on the right).

following the URI attributes of these features, links to web pages containing more information are presented to the user.

Outside of the standard suite of OGC services, we have also created a custom geosearch service. This is a very simple service that allows a client to query all heritage within a certain perimeter (a geometry), possibly enhanced with a buffer and filter by a limited list of parameters. This custom service was built because it allowed taking control of the generated output, tailoring this to our needs. It also made it possible to make some performance optimisations. Currently, only JSON output is supported. As always, an integral part of the output is URIs that point toward the full information objects that were found. While the service was created to support our Zoning component, it is also being made available to the wider public and can be used by external parties to query our data.

Joining the ocean

While we have created our own archipelago, we are still somewhat removed from the huge ocean of data out there. So far we have found surprisingly few linked data sets we could link to. Our explorations of the Flemish[7] and Belgian[8] open data portals have turned up lots of aggregated data sets (e.g. The population divided by sex and age at a certain point in time) that quite often have little to do with cultural heritage and are unsuitable for linking because of their aggregated nature. The most promising data sets to link to are geodata-sets such as CRAB (address data). But while these do contain stable identifiers for geographic objects, these have not been turned into URIs at this point of writing. Similarly, there exists a data set of administrative subdivisions of Belgium (regions, provinces, communities) that contains identifiers but not URIs. We do sometimes link to other external webpages from within our heritage inventories, but these links generally only point to HTML webpages that do not necessarily have a very stable URI and there is no structured data available at this URI.

On an international level we have found a few more prospects with the cultural heritage community. So far we have been creating links between our own controlled vocabularies and thesauri and international thesauri. This is fairly easy to do on both a technical and a conceptual level. We currently link to the Art and Architecture Thesaurus provided by the Getty Research Institute and the heritagedata.org thesauri (May *et al.*,

[7] https://opendata.vlaanderen.be

[8] http://data.gov.be

2015). A further step might be to connect our thesaurus of periods to the PeriodO gazetteer (Golden and Shaw, 2016).

On a purely cartographic level there does exist a wealth of information within Belgium. As noted before, we do make regular use of GIS open data sets of other government agencies in Flanders that are available. While these allow us to display our data on different basemaps in our geoportal, it does not provide for a truly integrated and linked data experience. On an international level we have incorporated both the Pelagios Imperium Romanum basemap (Åhlfeldt, 2012) and some cultural heritage map layers by the Cultural Heritage Agency of the Netherlands in our geoportal (Figure 6). While this does visualize data across borders it provides for little actual integration. It has, however, been an interesting first step that required little effort. The fairly simple integration of WMS and TMS services seems to be much easier to achieve than the more powerful but much more complicated integration of linked datasets.

Conclusion

Often linked data is seen as a good way to publish data and link it to other data. While this is certainly the case, we have found that it can be much more. By adopting the basic principle of linked data, naming things with HTTP URIs and linking them with other HTTP URIs, we have made linked data as the representation of our information resources the cornerstone of our enterprise architecture. It has proven to be as efficient for creating and maintaining data as for publishing it. Flanders Heritage will keep on adding islands to its little archipelago. Hopefully we'll also be able to find some new shipping lanes to other seas.

Acknowledgements

We wish to express our gratitude towards Philip Carlisle (Historic England) for proofreading this work and helping us find matching labels for often similar but not exactly the same concepts. We would also like to thank all colleagues at the Flanders Heritage Agency, both the IT colleagues who help shape and expand the system architecture and those heritage professionals who interact with it on a daily basis.

References

Ayers, D. and Völkel, M. (2008) *Cool URIs for the Semantic Web. W3C Interest Group Note 03 December 2008*. Edited by Leo Sauermann and Richard Cyganiak. Available at: https://www.w3.org/TR/cooluris/ (Accessed: 21 July 2016).

Åhlfeldt, J. (2012) 'A digital map of the Roman Empire', *Pelagios Commons*. Available at: http://commons.pelagios.org/2012/09/a-digital-map-of-the-roman-empire/ (Accessed: 26 July 2016).

Berners-Lee, T. (2009) *Linked Data*. Available at: https://www.w3.org/DesignIssues/LinkedData.html (Accessed: 25 July 2016).

De Roo, B., De Maeyer, P., and Bourgeois, J. (2015) 'Information flows as bases for archaeology-specific geodata infrastructures: an exploratory study in flanders', *Journal of the Association for Information Science and Technology,* 67 (8), pp. 1928–1942. doi: 10.1002/asi.23511.

Fielding, R.T. (2000) *Architectural styles and the design of network-based software architectures*. PhD Thesis. University of California, Irvine. Available at: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf (Accessed: 7 April 2017).

Flanders Heritage Agency (2015) *UriRegistry* [v: 0.1.1]. Available at: http://uriregistry.readthedocs.io/en/0.1.1/ (Accessed: 22 July 2016).

Flanders Heritage Agency (2016a) *Architectuur*. Available at: https://doc.onroerenderfgoed.be/docs/documentatie/architectuur/index.html (Accessed: 26 July 2016).

Flanders Heritage Agency (2016b) *Ontwerpen van REST services*. Available at: https://doc.onroerenderfgoed.be/docs/documentatie/richtlijnen/rest.html (Accessed: 22 July 2016).

Golden, P. and Shaw, R. (2016) 'Nanopublication beyond the sciences: the PeriodO period gazetteer', *PeerJ Computer Science*, 2. doi: 10.7717/peerj-cs.44.

Lanthaler, M. and Gütl, C. (2013) 'Hydra: a vocabulary for hypermedia-driven Web APIs', in Bizer, C., Heath, T., Berners-Lee, T., Hausenblas, M., and Auer, S. (eds) *Proceedings of the WWW2013 Workshop on Linked Data on the Web (LDOW2013).* Series: CEUR-WS.org Vol-996. Available at: http://ceur-ws.org/Vol-996/ (Accessed: 7 April 2017).

May, K., Binding, C., and Tudhope, D. (2015) 'Barriers and opportunities for Linked Open Data use in archaeology and cultural heritage', *Archäologische Informationen*, 38, pp. 173–184. doi: 10.11588/ai.2015.1.26162.

McKeague, P., Corns, A., and Shaw, R. (2012) 'Developing a spatial data infrastructure for archaeological and built heritage', *International Journal of Spatial Data Infrastructures Research*, 7, pp. 38–65. doi: 10.2902/1725-0463.2012.07.art3.

Myers, D., Dalgity, A., and Avramides, I. (2016) 'The Arches heritage inventory and management system: a platform for the heritage field', *Journal of Cultural Heritage Management and Sustainable Development,* 6 (2), pp. 213–224. doi: 10.1108/JCHMSD-02-2016-0010.

Uriarte González, A., Parcero Oubiña, C., Fraguas Bravo, A., Fábrega Álvarez, P., Vicent García, J.M., Pérez Asensio, E., Fernández Freire, C., and del Bosque

González, I. (2013) 'Cultural Heritage Application Schema: a SDI framework within the Protected Sites INSPIRE Spatial Data Theme', in Earl, G., Sly, T., Chrysanthi, A., Murrieta-Flores, P., Papadopoulos, C., Romanowska, I., and Wheatley, D. (eds) *Archaeology in the Digital Era. Papers from the 40th Annual Conference of Computer Applications and Quantitative Methods in Archaeology (CAA), Southampton, 26–29 March 2012*. Amsterdam: Amsterdam University Press, pp. 279–290.

Van Daele, K. (2016) *Onroerend Erfgoed ontology*. Available at: https://id.erfgoed.net/vocab/ontology (Accessed: 26 July 2016).

Van Daele, K., Meganck, L., and Mortier, S. (2016) 'On data-driven systems and system-driven data: twenty years of the Flanders heritage inventory', *Journal of Cultural Heritage Management and Sustainable Development*, 6 (2), pp. 153–165. doi: 10.1108/JCHMSD-01-2016-0004.

Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., and Colpaert, P. (2016) 'Triple Pattern Fragments: a low-cost knowledge graph interface for the Web', *Web Semantics: Science, Services and Agents on the World Wide Web*, 37–38, pp. 184–206. doi: 10.1016/j.websem.2016.03.003.

Ven, K. and De Bruyn P. (2011) 'Factors affecting the development of absorptive capacity in the adoption of open source software', *International Journal of Open Source Software and Processes*, 3 (1), pp. 17–38. doi: 10.4018/jossp.2011010102.

Vlaamse Overheid (2015) *Vlaanderen Radicaal Digitaal 2015–2019*. Available at: https://overheid.vlaanderen.be/sites/default/files/Conceptnota%20Vlaanderen%20Radicaal%20digitaal.pdf (Accessed: 25 July 2016).

Vlaamse Overheid (2016) *Open Data handleiding*. Informatie Vlaanderen. Available at: https://overheid.vlaanderen.be/sites/default/files/Open_data_handboek_2016.pdf (Accessed: 25 July 2016).

Webber, J., Parastatidis, S., and Robinson, I. (2010) *REST in practice. Hypermedia and systems architecture*. Sebastopol, CL: O'Reilly Media, Inc.