



Case Study

OverStory joins the Cambridge University Press Academic group to design and build their bespoke academic publishing infrastructure

Executive Summary:


Cambridge University Press' innovative new academic platform was moving forward, however the Content Engine was struggling to make the same progress. Whilst the new platform's user journeys, branding and interface work was progressing well, the Content Engine's architecture, data models and implementation of was not shaping up as had been planned. Consultants had been brought in but little progress had been made. This case study explores how OverStory came to be involved and helped CUP achieve their goals.

Content Engine challenges:

- 1) The Cambridge Core platform was a major initiative by CUP that depended heavily on a robust underlying Content Engine.
- 2) CUP wanted to base the system on a MarkLogic backend, however there was no in-house expertise.
- 3) The project had been struggling for some time before OverStory's engagement, and milestones were being missed.
- 4) Prior technical Consultancy was not providing the solutions that CUP needed to meet their programme launch commitments.
- 5) There was no Content Engine architecture or Content Model in place, these needed to be designed from scratch.

About:

Cambridge University Press is the world's oldest and largest university press (founded 1534). When OverStory met them in 2015, CUP already knew its digital publishing infrastructure needed a complete overhaul to support their goals for the future. They had already begun the process of imagining their own modern in-house platform.



Cambridge Core replaced CUP's 2 legacy platforms – Cambridge Journals Online and Cambridge Books Online, including over 360 journals and 30,000 ebooks. All content from Cambridge Histories Online, Cambridge Companions Online, Shakespeare Survey Online, and partner publisher content from University Publishing Online is also available on the new platform.

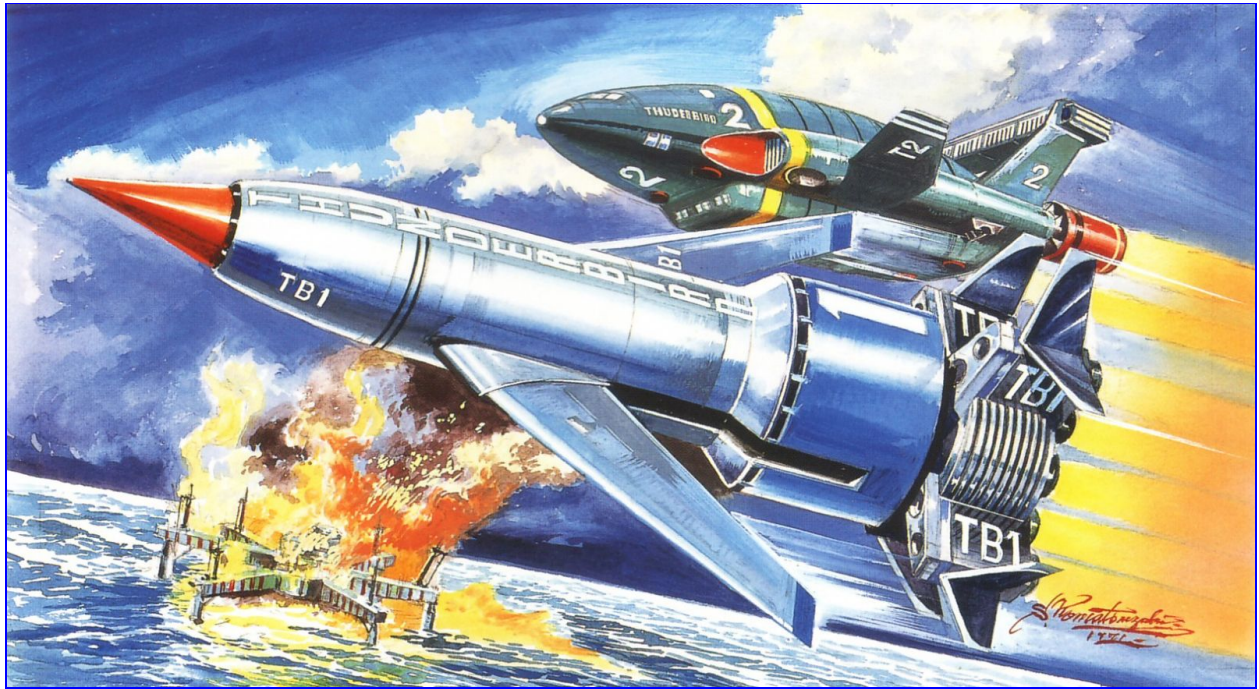
Cambridge Core promotional video: <https://vimeo.com/180916259>

This case study details how OverStory was brought into the project to assist with the large scale transition and creation of the new Cambridge CORE platform.

Fact check:

- CUP was founded in 1534, and is the largest University Press
- OverStory joined forces with CUP in April 2015
- The switch to Cambridge Core as a live platform was on September 4, 2016
- Cambridge Core hosts 30,000 eBooks and 545 journals, containing 1,496,470 million scholarly articles
- Cambridge Core can be found at: <https://www.cambridge.org/core>

Thunderbird 2 To Thunderbird 1, Do You Read Me?



The Situation as of April 2015

CUP knew the platform they wanted. Their existing platforms had become unwieldy, siloed, and inefficient for an organisation with so much publishing muscle. CUP wanted a ground-breaking bespoke platform to host their journals and books, that could serve as a springboard for new growth and functionality. The new platform also aimed to rejuvenate the way users interacted with, and consumed the vast amount of books, journals, articles and other academic material.

Problems Below the Water Line

The Personas had been designed around the typical user types. Scenarios and user flows had been developed for each. The branding was coming together and the User Interface was shaping up. Technologies had been selected for entitlements, and for the Content Engine's architecture, MarkLogic had been purchased. This was the sensible choice for a large corpus of academic content. All good.



There was a problem though. After some time-consuming false starts, the system architecture for storing, managing and delivering content still wasn't clear. Time was ticking on as deadline commitments were approaching. In April 2015, when OverStory were introduced, there was still no real consensus on the essential questions surrounding the design of the architecture and data model, and no working Content Engine implementation.

In early exchanges with CUP, it became clear that there was no Content Engine architectural plan in place, and the need for a foundational system was becoming ever more urgent, as the intended launch date was coming up fast (originally planned for December 2015).

Personalities Involved

Whilst it's common to create a Case study in the third person. Names make it more tangible, and so this is where we introduce Ron Hitchens, OverStory's Principal Architect and Nik Louch, Platform Technologies Director (Academic and Global) at CUP.

What Brought OverStory and CUP together?

Ron and his OverStory consultancy were known as the go-to people for MarkLogic solutions in the real world. The Consultancy had been set up originally for that purpose. Ron had made things work for Springer, Wiley, CABI, IOP Publishing and others. Before moving to the UK and forming OverStory, Ron spent several years on the Core Engineering Team at MarkLogic in California. In those early days (employee #20), Ron created several key features of MarkLogic.

MarkLogic is a powerful and complex data management platform. When used well it yields tremendous business value. If misunderstood or used inappropriately however, customers can struggle - never realizing its full potential. OverStory was founded as a specialist consultancy to assist clients in making the most of their significant investment, and to help with the design and implementation of new undertakings such as this.

When CUP went looking for MarkLogic expertise they sought recommendations from respected fellow publishers like Wiley and Elsevier, the unanimous consensus was that OverStory could help. Thus the relationship began.



He's Got To Have Made It. He's GOT To



Following an initial requirements discussion where CUP laid out the plan and the objectives of the new Cambridge Core system in general, and the Content Engine in particular, OverStory and CUP developed a two-pronged approach. The first and most pressing need was to get the Content Engine's QA and ingestion process started, as the sheer volume of content that had to be checked and migrated. This was expected to take many months, and Core couldn't launch until it was completed.

The second prong would consist of content search, entitlement policies and content delivery to support the web front end. These would all depend on the Content Engine's API being available. Search and delivery did not concern OverStory, CUP were developing those components themselves. There were of course overlaps with the capabilities with MarkLogic, but CUP wanted to explore other options.

Ron and the team set about designing a flexible deployment architecture, a robust and extensible content model (including semantic/RDF features) and a comprehensive Content Engine API. Integral to this content system were features that support a sophisticated workflow embodied in a Content Quality Assurance application developed by CUP technical staff, using the OverStory-designed API.

Putting this in place gave CUP a solid foundation to build upon, consisting of MarkLogic (where the content was stored and indexed), a robust content model (which enabled efficient indexing, fast retrieval and semantic enrichment), and a well thought-out microservice REST API (to make integration easy and minimize dependencies).

Approach Vector Set

This separation of concerns allowed Nik to make a strategic decision, which was to commence work on the second prong in parallel, using development resources inside CUP. This team went with a standalone search engine (ElasticSearch) and other technologies (Node.js, Neo4j, etc), choosing not to extend the MarkLogic footprint in Cambridge Core beyond the Content ingestion, QA and enrichment capabilities.



This decision initially accelerated development. OverStory was freed from concerns about the very complex entitlements model that Cambridge Core required, and the CUP team did not need to climb the rather steep learning curve of MarkLogic and XQuery. Things moved along quickly.

There is however, no such thing as a free lunch. When it came time to integrate these two halves of the system, there was some duplication of efforts. The content flowed into MarkLogic, being validated and approved by the QA team along the way, and was then available via REST endpoints. But as MarkLogic's native search was not being used, the data had to be fed out again to be imported into ElasticSearch. The Content Engine's APIs were all XML based but the other system expected JSON. Although resolved in the end, this illustrates there can sometimes be future integration costs beyond the initial separation benefits.

An important aspect of splitting development like this is the inevitable trade off consideration between efficiency and risk. If everything had been built on MarkLogic, its native search engine would be far faster and accurate and the need for replication of data and conversion to JSON would have been completely eliminated. But on the other hand, entitlements would've had to be implemented on MarkLogic (in XQuery) and CUP would have been putting all their eggs in one basket, which carried significant risk. OverStory would have preferred the all-MarkLogic route, but we're admittedly biased, as we know how well this type of system can work. We won't second guess CUP's decision though, as given the circumstances, it made good sense. The system that ultimately emerged works, it's robust and delivers great value for Cambridge University Press.

How Did the Platform Design Emerge?

Ron presented a very high level architecture concept to Nik's team, who readily agreed.



Time was short, and although it would have been good to have more upfront design time before implementation started, Nik put his trust in OverStory, and our prior experience, that they could deliver. So the nod was given to begin and the agile phase got underway.

Overstory's experience in this area meant that there were some definite knowns around the technology stack required. This eliminated the feasibility analysis time, and minimised discovery phases (where technology is trialled to identify risks and weaknesses). The Agile delivery phase had begun, leaving details of design and implementations to emerge along the way.

Commence Main Engine Sequence

Microservices were at the heart of the architecture, with each service being responsible for a specific business capability. In addition to the primary Content Engine, there was an XML transformation service and a PDF stamping service, which applied branding and watermarking to PDFs delivered on demand.

The microservices all run on Java Virtual Machines (JVMs). Rather than using traditional servlet containers, OverStory prefers the new generation of lightweight micro containers that create high performance, highly parallel asynchronous platforms. We strive to adhere as much as practical to the the [Twelve Factor Apps](#) philosophy. The use of these technologies allow quick and easy spin up and shutdown of cloud-based compute units as needed, without the complexity and bloat of heavy-weight servlet containers. For CUP we went with [Ratpack](#), which is the least restrictive and most flexible. Ratpack is also friendly to the Groovy language, which we like very much as a superset of Java with many productivity enhancements.

Contact Magnets: Activated

We ploughed forward and followed our guiding architectural plan. We also designed a data model (an internal XML format for structuring documents) that would accommodate existing journals and books, but also be flexible enough for future needs. This was essentially greenfield design, luckily there was very little existing legacy that we had to deal with.

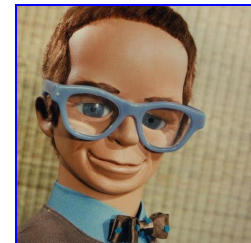
There was however, as with any project of this size and complexity, emerging requirements that needed to be understood and factored into the design. Notably, the Citation service, the ability to extract text from PDFs, and JSON data feeds, were among the additional needs that had to be accommodated.

Adapting to Emergent Processes

During this project, CUP were also transforming how they used Agile. The tools and processes improved from early 2016, with the disparate project teams converging onto common project management tools for sprint planning, reporting and communications. The OverStory team was then able integrate with with these tools as well. OverStory continually adapted to best fit for CUP's needs, to deliver value increments as planned for CUP project milestones.

Brains, What's the Situation?

The project is now complete. Cambridge Core was launched successfully in September 2016. Nik and CUP team are very happy with the system we have produced for them. They have no concerns about its capability, performance or scalability. This is really important to us at OverStory, as we strive to make our client the heros, with systems that meet their needs, and exceed expectations wherever possible.



Cambridge Core is now functioning smoothly, and as originally envisaged: i.e., as a springboard platform for new capabilities and business opportunity. Well done CUP, the effort has paid off. Long may it continue.

Epilogue, Back on Tracey Island:

Training

During the Core project, we also helped CUP by providing training to their staff. We often find training fits well with Consultancy. We can apply our skills and expertise to address client needs, while helping to improve their in-house skills before, during, or after our engagement.



At CUP they needed to upskill their production team members on XML, XSLT, and related topics. We organised a two day course led by one of the most respected XML experts in the UK. Feedback survey data was uniformly very positive.

Awards

Cambridge Core was also shortlisted for 3 awards in 2016:

- [UXUK Design Award](#)
- [Futurebook's Platform of the Year](#)
- [Quantum Publishing Innovation Award](#)

References:

Nik Louch is happy to be contacted to be provide references for OverStory.

Can We Help You Achieve Your Goals?

OverStory are unique in the world of MarkLogic consultancies, our founder is a veteran of MarkLogic having designed and implemented key features of the product. But we're technology agnostic and if MarkLogic isn't right for you, we'll say so. We often integrate MarkLogic with open source technology and other commercial products.

[Talk to us today.](#)

Ron Hitchens (ron@overstory.co.uk)

Chris Hayes (chris@overstory.co.uk)

overstory.co.uk

[@overstory](#)