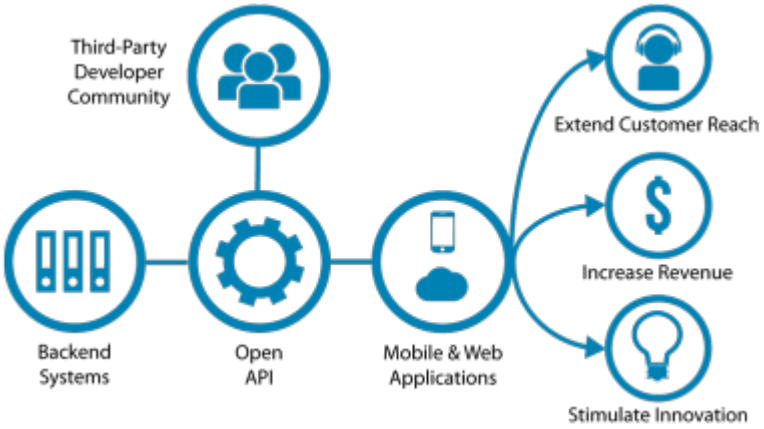
Gatekeepers in Digital Archaeology

written by Jeremy Huggett | 11/11/2016



We're becoming increasingly accustomed to our digital technologies acting as gatekeepers – perhaps most obviously in the way that the smartphone acts as gatekeeper to our calendar and/or email. In fact, this technological gatekeeping functionality appears everywhere you look, whether it's in the form of physical devices providing access to information, software interfaces providing access to tools, or web interfaces providing access to data, for example. A while ago, I mused about the way that archaeological data are increasingly made available via key gatekeepers, and that consequently "negotiating access is often not as straightforward or clear-cut as it might be – both in terms of the shades of 'openness' on offer and the restrictions imposed by the interfaces to those data." Since writing that, I've essentially left that statement hanging. What was I thinking of?

Looking back, I had archaeological data archives in mind – the cyberinfrastructures we're in the process of building and using, and the availability (or otherwise) of programmable interfaces to these. However, there are in reality numerous providers of archaeological or archaeologically-related data who make their information available via such interfaces. Here in the UK, for example, government agencies such as Historic Environment Scotland (HES) make their data available as Web Feature Services (WFS) or Web Mapping Services (WMS). These enable us to connect remote data sources to our local GIS tools, although as the data are frequently provided only in bitmap form, subsequent analysis and use can be rather limited. HES and other organisations like them, including data archives such as tDAR and Open Context, provide invaluable access to their resources that either sidestep their public interfaces, or provide access to data that might otherwise be inaccessible.

These programmable interfaces, or Application Program Interfaces (APIs), are increasingly widespread beyond archaeology: the ProgrammableWeb lists over 16000 APIs at the last count, and most people are familiar with (in the sense of encountering, at least!) the use of APIs associated with Google Maps, Twitter, Facebook, etc., in the form of embedded maps through to social media buttons on web pages. In fact, APIs extend far beyond these – for instance, programmers will be familiar with the use of APIs which provide them with functions, classes, methods etc. for addressing the user interfaces of operating systems. Indeed, there has been an explosion in the use of APIs, not least because they avoid reinventing wheels, provide greater flexibility and functionality without the associated overheads, and facilitate innovation (for example, Murphy and Sloane 2016). Murphy and Sloane even write of APIs leading to a rethinking of the value of data:

> "In the past, the biggest companies were those closest to the data … In the API economy, the biggest companies may be the ones that aggregate the most data smartly and open it up to others."

Aggregation and opening up data is of course precisely what national archaeological agencies and archaeological data archives are doing at the moment through their use of APIs. But without denying their value, what are the implications of this kind of access? I've previously argued that open data can paradoxically increase the distance between the data and the data user (Huggett 2015a, 89-90; Huggett 2015b, 13); in much the same way, APIs surreptitiously increase the distance between developer and source code and disguise this to some degree by exposing a programmable interface which is both controllable and restrictive.

Yuanbo Qiu has recently observed that there are different degrees of openness in relation to APIs – some (many) Open APIs are actually Closed Source APIs, and even 'true' Open APIs can be implemented in ways that ensure or reinforce various forms of control (Qui 2016, 8-9). Openness is redefined to mean the availability of a public interface to an internally managed system that enables external users to create tools, apps and their like on top of that system without access to the underlying code. This means that when the API is changed or withdrawn, the external user is cut adrift until either they have modified their tools in the light of the API changes, or they have found an alternative provider (Qiu recalls the effects of Google's closure of its Reader API on those who use and follow RSS feeds extensively, for example). Of course, this shouldn't happen – the very idea of an API is that significant changes can be made to the underlying code while the API remains unchanged so that the developer or end-user does not have to start over (for example, consider the way that much software written for early versions of the Windows operating system still operates on modern versions as long as the API calls were used correctly in the first place, even to the extent of the window chrome updating without any further intervention or recompiling of the software). Benefits aside, what this also underlines is that using an API places the developer and/or end-user in a dependent relationship with the provider – or, as Qiu puts it, the conveniences of the API are attractive to users while simultaneously intensifying the power asymmetries in the user-supplier relationship (2016, 10). This asymmetry is part of what defines a gatekeeper.

In addition, the lack of exposure of the underlying code – seemingly made more palatable through the general availability of the API and the attractiveness of its relative simplicity and greater flexibility – means that the drawbacks of non-Open Source software cited in an archaeological context (for example, Marwick 2016; Ducke 2015) apply here too. The person accessing the API calls is constrained by the functionality provided through the interface, and regulated and restricted in largely unrealised ways by the underlying, invisible code. The API gives the illusion of transparency, even neutrality, but it does so by hiding its fundamental functionality. Or, to put it another way, an API can provide the end user with a cushion against changes in the format and behaviour of the underlying system, while at the same time acting as a shield for the provider against the user, controlling and restricting their access and use of the system. Here, the gatekeeper allows access but strictly on their terms, and those terms – in the sense of the software code and its manipulation of the associated data – are not available for examination.

Furthermore, even fully Open Source APIs can change with the passing of time. As Qiu points out, some Open APIs start open but introduce restrictions later because of changed circumstances – he

cites the example of Twitter (2016, 13) which now requires every request to its API to be authenticated via a key, with special keys and additional conditions applied to 'heavy' users. Keys are one common way in which Open APIs can introduce monitoring of access and use, and, if need be, close down access. Needless to say, keys and gatekeepers are frequently closely linked!

Of course, we could argue that none of this applies to archaeology and our use of APIs. After all, as archaeologists we believe in making our data freely available, and we're in a situation where, to a large extent, archaeologists are doing this for ourselves. Nor is this particularly linked to creeping commercialisation, despite concerns about the sustainability of the various archaeological digital archives. All this is beside the point. The technological systems we build and use tend to increasingly hold us at arms-length, whether from the data or the tools that are used to manipulate those data, while at the same time wrapping us in the warm reassurance of our ability to access data in greater quantities and with greater ease than ever before. Consequently, whether we realise it or not, the tools and facilities which make our work possible and provide opportunities for innovation and knowledge creation, also act as constraints, limiting what we can do, and doing so covertly on the whole. Far from Lamar Tyler's bold proclamation that the gatekeepers are gone (2014), they are very much present, and are increasingly running the show (although not in the way described by Llewellyn King (2016!)).

References

Ducke, B. 2015. 'Free and Open Source Software in Commercial and Academic Archaeology', in A. Wilson and B. Edwards (eds.) *Open Source Archaeology. Ethics and Practice*. Berlin: De Gruyter Open. pp. 92-110. http://dx.doi.org/10.1515/9783110440171-008

Huggett, J. 2015a 'A Manifesto for an Introspective Digital Archaeology', *Open Archaeology* 1, 86-95. http://eprints.gla.ac.uk/104047/

Huggett, J. 2015b 'Digital Haystacks: Open Data and the Transformation of Archaeological Knowledge', in A. Wilson and B. Edwards (eds.) *Open Source Archaeology. Ethics and Practice*. Berlin: De Gruyter Open, pp. 6-29. http://eprints.gla.ac.uk/114652/

King, L. 2016 'The Gatekeepers are Running the Government', *Huffington Post*, August 15, 2016. http://www.huffingtonpost.com/llewellyn-king/the-gatekeepers-are-runni_b_11486830.html

Marwick, B. 2016. 'Computational Reproducibility in Archaeological Research: Basic Principles and a Case Study of Their Implementation', *Journal of Archaeological Method and Theory* (in press). http://dx.doi.org/10.1007/s10816-015-9272-9

Murphy, M. and Sloane, S. 2016 'The rise of APIs', *Techcrunch* May 21, 2016. https://techcrunch.com/2016/05/21/the-rise-of-apis/

Qiu, Y. 2016 'The openness of Open Application Programming Interfaces', *Information, Communication & Society*, http://dx.doi.org/10.1080/1369118X.2016.1254268

Tyler, L. 2014 *The Gatekeepers are Gone: Hustle + Technology = Success*. Atlanta GA: Tyler New Media.