

Recognising Achievement with Badges and Blockchain in a Connectivist MOOC

Stephen Downes

Digital Technologies Research Centre, National Research Council, Canada

Abstract: While previous work has recognised the potential for open badges and blockchain to play a role in online courses, this potential has yet to be realised in a fully decentralised cMOOC. This paper describes the design objectives of an application that integrates open badges and blockchain with a cMOOC. The work described was undertaken during the offering of an online course, and, thus, development took place in an actual course context with interaction with course participants. The full workflow from course content to storage on the blockchain is described, and some concluding comments are offered on the results of this course, and the potential for future applications.

Keywords: MOOC, cMOOC, badge, blockchain, Webmention, RSS, decentralised.

Introduction

The ‘open badge’ is the name for a technical infrastructure and set of specifications for recognising learning achievements with a digital icon, or ‘badge’. Open Badges were introduced by the Mozilla Foundation with a working document in 2012. (Mozilla, 2012) Open Badges became an IMS technical specification with version 2.0 being released in 2018. (IMS Global, 2018) The specification “describes a method for packaging information about accomplishments, embedding it into portable image files as digital badges, and establishing resources for its validation.”

Open badges constitute a part of a wider set of practices called ‘open pedagogy’ (Conole, 2013, p. 210). They offer support for reflective practice and peer review. Along with other aspects of open pedagogy, and in particular the massive open online course (MOOC), open badges help address the issues of demand, affordability and accessible to technology in learning in the developing world. This is seen in such work as IIT Kanpur’s mooKIT, launched in 2012, and in the Commonwealth of Learning’s MOOC4D project, launched in 2015. Open badges play a role in promoting lifelong learning, credible credentials, and, for their recipients, employment opportunities.

Created around the same time, ‘blockchain’ is the name for a set of technologies commonly associated with creating and managing a decentralised digital currency ledger such as Bitcoin. Blockchain was first proposed by a person or entity named Satoshi Nakamoto in a white paper published in 2009. (Nakamoto, 2009) This paper contained elements common to most subsequent work in the field: a mechanism of linking or ‘chaining’ sets of data entries (called ‘blocks’) using cryptographic ‘hashing’ functions; and a mechanism for distributing the creation and management of blockchain data in a decentralised network using a ‘proof of work’ mechanism.

The potential for the use of blockchain to encode educational records was quickly realised. MOOCs and similar non-traditional models face the widely reported problem of academic fraud and



corruption. Efforts have been made to combat the problem with legal mechanisms but, ultimately, agencies and institutions are demanding digital solutions (Børresen & Skjerven, 2018). Hence, for example, Sharples and Domingue discuss using blockchain to provide proof of intellectual work or as a form of intellectual reputation management (Sharples & Domingue, 2016, p. 492). MIT researchers proposed a 'Blockcerts' infrastructure in 2016 (Lab, 2016). Grigore Albeanu discusses the application of blockchain to various educational applications ranging from authentication to certification. (Albeanu, 2017, p. 271). Gräther et al. (2018) propose blockchain "as a practical solution for issuing, validating and sharing of certificates."

The Massive Open Online Course (MOOC) was created in 2008 as a means of creating a distributed open access online course (McAuley, Stewart, Siemens, & Cormier, 2010). "MOOC integrates the connectivity of social networking, the facilitation of an acknowledged expert in a field of study, and a collection of freely accessible online resources. Perhaps most importantly, however, a MOOC builds on the active engagement of several hundred to several thousand "students" who self-organize their participation according to learning goals, prior knowledge and skills, and common interests (p. 4)."

With the launch of the Artificial Intelligence MOOC by Norvig and Thrun in 2011, MOOCs diverged into two paths: on the one hand, the 'eXtended' MOOC, or xMOOC, characterised by the online course offerings from major educational institutions in a traditional course-package format as offered by agencies such as Coursera, Udacity or EdX, and on the other hand the 'Connectivist' MOOC, or cMOOC, based on a network of associated websites and resources, as described by McAuley, Stewart, Siemens, & Cormier (2010) (Bates, 2014).

The use of blockchain to encode open badges for Coursera xMOOCs was proposed by Melanie Swan (Swan, 2015; Albeanu, 2017) and the application of blockchain for certification in MOOCs is suggested by Hood and Littlejohn: "a blockchain system records each transaction so that the student has a verified set of qualifications associated with him or her" (Hood & Littlejohn, 2016, p. 13).

Design and Methodology

The work described in this paper has the objective of developing an open credentialing system for a cMOOC based on badges and blockchain. This work was undertaken in the context of the design and delivery of E-Learning 3.0 (eI30), a connectivist MOOC offered by the National Research Council in October-December, 2019. (<https://eI30.mooc.ca>) The purpose of this work, and hence the research design methodology employed, is to establish a *proof of concept* for the employment of badges and blockchain in a connectivist MOOC, that is, to show whether it is feasible at all, and how it might be done. This is distinct from what would be a separate exercise of advocating this solution by, say, showing the benefits, or of an exercise whereby a methodology of badges and blockchain are embedded in a theoretical context beyond that assumed in the employment of a connectivist MOOC.

This course was advertised (via Twitter and in email newsletters) as a design experiment both developing the concepts of current and emerging distributed learning technology, and implementing those concepts in a course management system called gRSShopper. The implementation was described in a series of blog posts made available to course participants, and gRSShopper source code was made available on GitHub (<https://github.com/Downes/gRSShopper>).

As a cMOOC the course did not require formal registration. However, participation statistics were collected through the gRSShopper aggregator as follows: newsletter subscriptions (177), harvested

feeds (15), with the number of unique visitors to the E-Learning 3.0 course website reaching 3000. A total of 39 participants completed an associated research survey (Fournier, Molyneaux & Kop, 2019, p. 232). As has been the case with other MOOCs, participants were well-educated with previous experience in online learning.

The course consisted of two weeks of preparation, and nine weeks of instruction. Course modules were as follows:

- Data – developing the use of dynamic and decentralised data instead of documents
- Cloud – employing ubiquitous internet data storage and services
- Graph – identifying connections between data elements to create new knowledge
- Identity – distributed and graph-based models of authentication and individuation
- Recognition – distributed recognition networks, open badges, and blockchain records
- Community – exploring decision-making and the concept of community as consensus
- Experience – integration of content and creativity in cooperative networks
- Agency – defined as security, identity, opportunity and voice.

As a cMOOC, the course was designed in the form of a network, with participants expected to use their own websites and blogs (some participants also used their own instances of gRSShopper as personal learning environments (PLEs). Each week was structured around a video introduction, a live webinar with a guest, a daily email newsletter, contributions (in the form of posts or videos) from course participants, and a summary article authored by the course instructor.

This paper describes one aspect of the course, the design of a distributed data-based system for recognition of completion of tasks with open badges, with the recognition encoded in a blockchain. Tasks were disseminated by means of the course newsletter, were completed by course participants in their own personal learning environment, and aggregated by gRSShopper to begin the process of review and recognition. Much of the development of the software enabling this process was developed during the course in the context of course delivery and interaction with course participants.

Data on course participation was collected in accordance with the National Research Council Research Ethics Board (REB) regulations and under REB review. Participants were informed of all aspects of data collection and gave their consent prior to taking the course. Participation in data collection was not a requirement for participation in the course. All course content, activities and results were freely accessible without charge.

In the next few sections of this paper we will describe the achievement recognition system developed during the course. This technology was actually deployed during the course (or in the base of blockchain, shortly after the course). It should be understood that the software developed consists of prototypes only, and was not designed for a production environment, or indeed any environment outside the context of this particular investigation. The result of this work should be understood as a proof of concept demonstrating the feasibility of a distributed blockchain-based badge network.

A note on typology: in order to distinguish elements that are defined as objects, and therefore have a distinct record in the gRSShopper MOOC management system, we capitalise the name of the element

when we refer to it. Examples of objects identified in this way include Course, Module, Task, Link and Badge.

Modules and Tasks

A course module is a section of a course associated with a specific set of skills or competencies. The definition of skills or competencies is out of scope for this paper, however, this definition will typically include at a minimum a required mechanism whereby course participants can demonstrate their achievement of the skill or competency. In the cMOOC this requirement is defined as a Task.

Modules are associated with courses (University of Sussex, 2012) (free-standing modules may also be defined as learning objects, but in the context of a cMOOC modules are not free-standing). Courses can be extensively defined but for the purposes of a cMOOC the course requires only limited metadata: course title, and course description. Other metadata about the course is defined by associating or linking the course object with other objects, and in particular, objects such as the course Author, the offering Institution, a course Section (referring to a specific offering of the course), and relevant to the current context, course Modules.

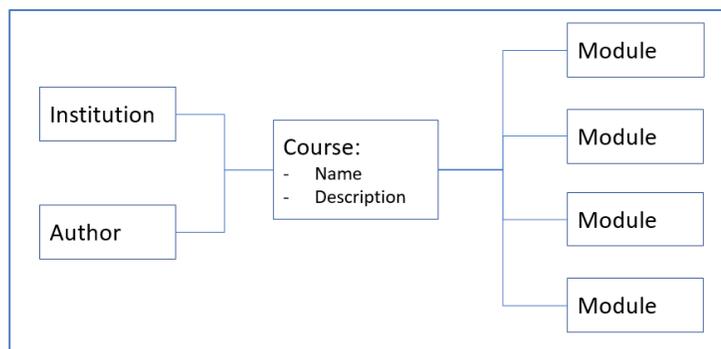


Figure 1: Course definition. Lines depict associations between entities as defined in the course graph. Note that all associations are many-to-many unless otherwise indicated.

Similarly, a module can be defined with minimal metadata consisting of a title and description, with the remainder of the module definition created by means of association with additional objects. These objects include the course (as described above), skills and competencies, links and resources, activities, and, as relevant to the current discussion, tasks.

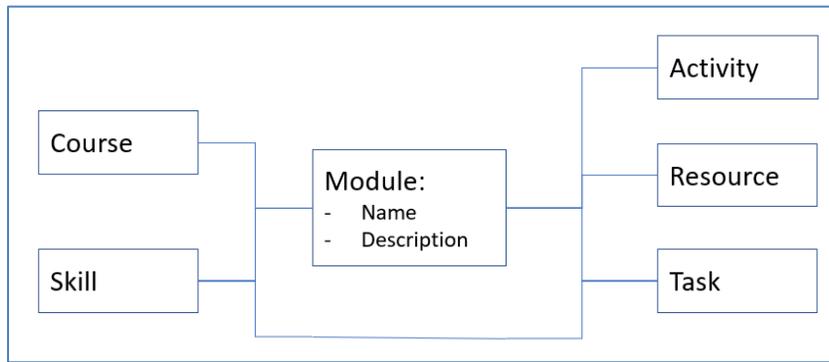


Figure 2: Module definition. Lines depict associations between entities as defined in the course graph.

In a cMOOC management system, course designers are presented with an interface to a database allowing them to create the objects in question by entering text in the form, upload files that may be associated with the object, and to associate the current object with any other object. Figure 3 shows a sample course with all modules defined and associated.

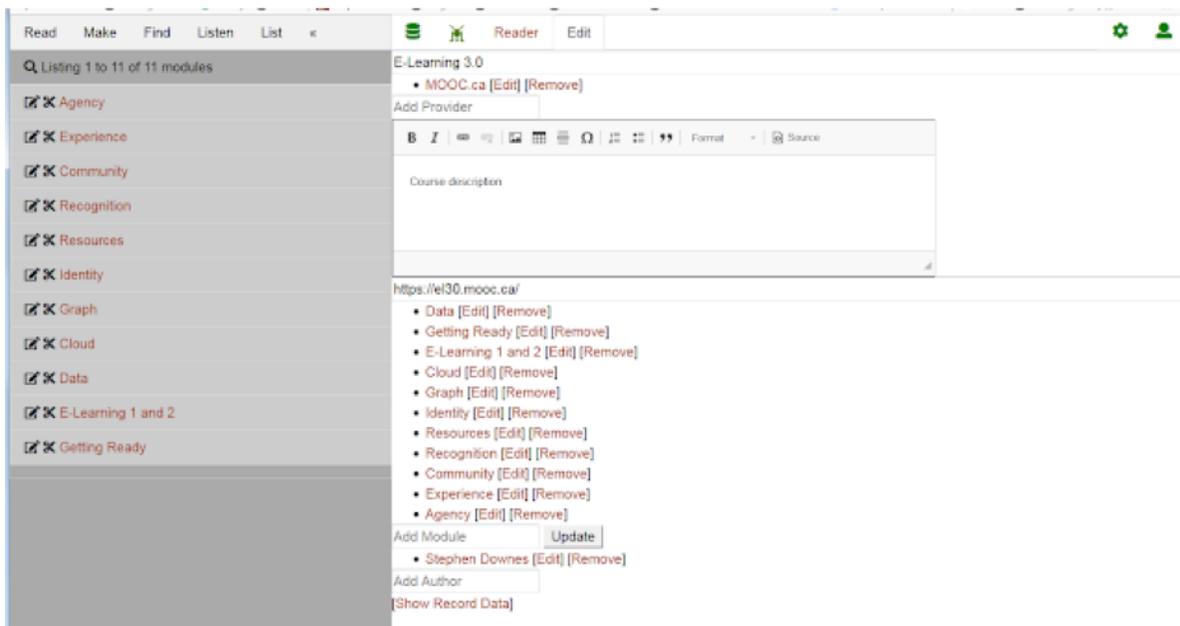


Figure 3: Course defined and associated with Modules. Screenshot from gRSShopper.

A task may be defined however desired by the course author. Students completing the task use their own online environment to record the outcome – they may, for example, create a blog post, or upload a presentation to SlideShare, or create a photo montage in Flickr, amongst numerous other options.

Completing and Submitting Tasks

In order to make the outcome of the task available to the instructor, the student could send the location of the web artifact (that is, the URL) directly to the instructor, either by means of a direct email or via an online form. This is less than ideal, however, because it is a labor-intensive task.

Most online services such as Blogger, Flickr or Slideshare allow users to share an RSS feed for that service. This publishes a machine-readable summary of recent website contents authored by the user. (RSS stands for either 'Rich Site Summary' or 'Really Simple Syndication'). These summaries are available to content-consuming applications, known as 'readers', 'aggregators' or 'harvesters', in a variety of formats, including RSS, Atom, or Javascript Object Notation (JSON).

```

<feed>
  <author>
    <name>Joe Author</name>
  </author>
  <item>
    <title>item 1</title>
    <url>URL 1</url>
  </item>
  <item>
    <title>item 2</title>
    <url>URL 2</url>
  </item>
  ....
</feed>

```

Figure 4: Sample simplified RSS feed. Elements of this electronic document are read by a harvester and saved as Links and associated with a Feed and Author.

gRSShopper includes an RSS reader (hence the 'RSS' in its name) which can aggregate several versions of RSS, Atom, and JSON. These files, once harvested, are analysed and an entry created for each post or item listed in the feed. Internally, these are depicted by gRSShopper as 'Link' entities. A user makes their entries available to gRSShopper by submitting the URL of their RSS feed, which is depicted internally by gRSShopper as a 'Feed' entity, and associated with the contributing 'Author' entity, where the author is the student submitting the contents.

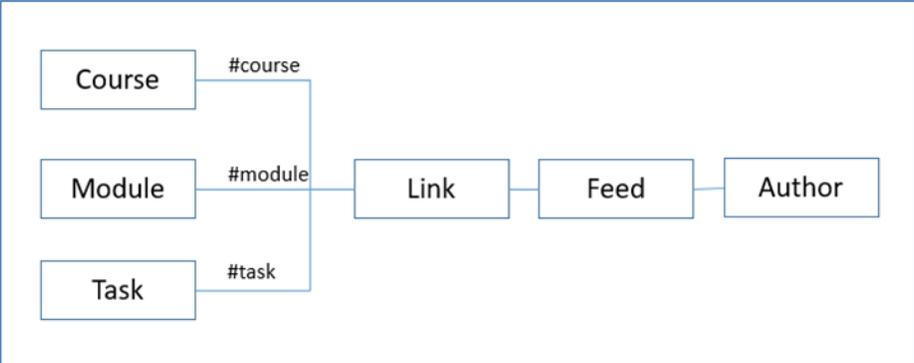


Figure 5: Link, Feed and Author, associated with Course, Module or Task by means of a hashtag. Lines depict associations between entities as defined in the course graph.

When the harvester reads the Feed, it extracts one or more Link. It analyses the Link and associates it with a task, and optionally, with a module and a feed.

There are numerous ways this association may be made. In the current case, the association was completed manually. In previous MOOCs, the mechanism of a 'hashtag' was used, whereby a distinct string, preceded by a '#', was inserted into the content or title of the blog post; this allowed gRSShopper to automatically create the association. In the future, a more intelligent analytics system may recognise the topic or skill demonstrated in the Link, as associate the Link with the most relevant Task.

Reviewing Submissions and Awarding Badges

In the current course, for any task, an associated badge is created (future courses could associate badges with groups of tasks, or modules, or entire courses).

In the current course, we worked with a third-party badge issuer, Badgr, though the standard open web badges protocol and specification was used. (<https://www.badgr.com>) This is a multipart protocol, and we used entities previously defined to satisfy the protocol. The workflow was as follows:

1. Create a Badgr account. You need to go to the Badgr website and register with an account (that is, create a UserID and a password). This only needs to be done once.
2. Obtain an Access Token. To make a request using an API you need to create an access code and put it into the header of your request. So, your first API request will be to create an access token. This also only needs to be done once.
3. Create an Issuer. An 'issuer' is the person who actually creates and awards badges. You would think it's the same person who just created the access token, but no. This is done once per issuer.
4. Create a Badge Class – this is the badge itself, and is defined (in our case) as being associated with the completion of a task. The badge name, description and criteria are defined by the associated Task entity, and the badge URL is the Task URL. This is done once for each badge.
5. Award a Badge. Information about the person being awarded the badge (being in our case the Author) is submitted to Badgr, which returns an instance of the badge class, and specifically: the 'evidence' URL (i.e., the URL of the Author's Link associated with the Task), an image of the badge with badge information embedded in the image file (i.e., 'baked'), JSON data associated with the badge, and a verification process.

To be clear, although each badge class is created only once, a new copy of a badge is created for each person who receives the badge. That is because the information about the recipient is 'baked' into the badge. This can be seen in the IMS Global Open Badges Baking Specification example of a "well baked SVG with a hosted assertion" which includes the recipient as identified by email address (IMS, 2018a). By analogy, we might think of the 'badge class' as the master copy of a printed certificate, with the 'badge' being the printed certificate with the recipient's name written on it.

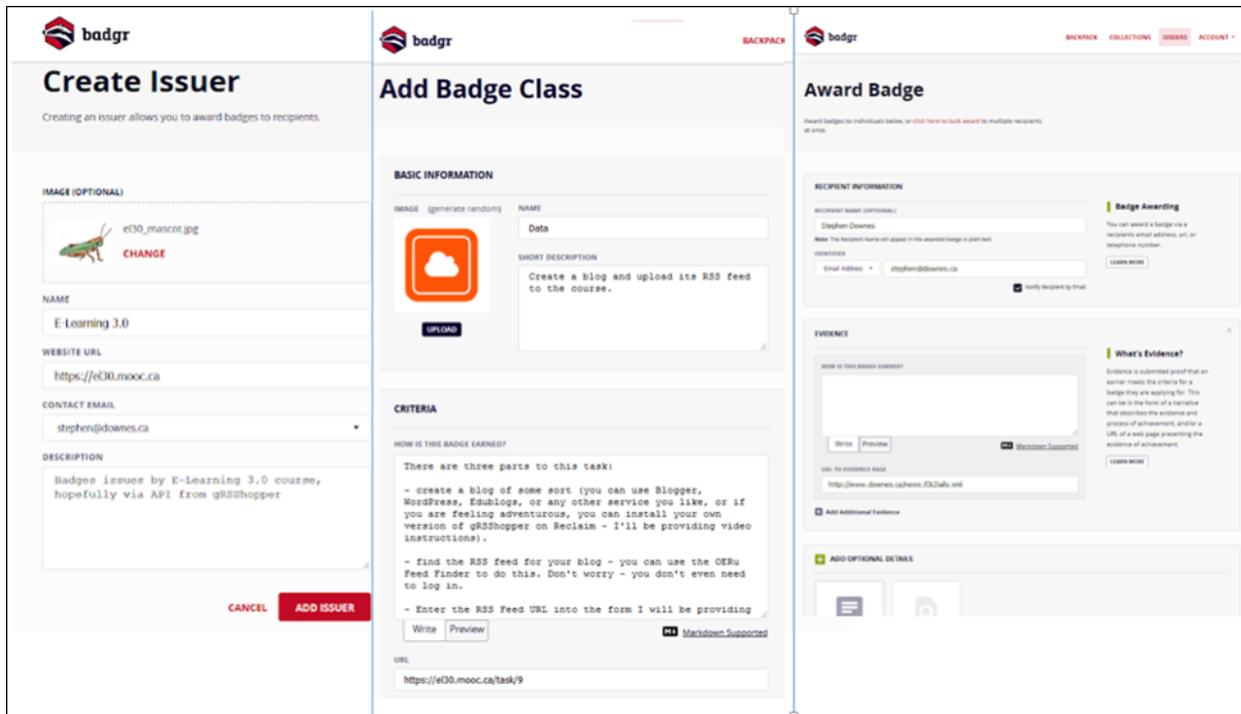


Figure 6: Three stages of badge creation in Badgr: create an issuer, create a badge class, award a badge. Three screen shots from Badgr.

In gRSShopper, each badge is depicted internally with a 'Badge' entity. The Badge entity contains data created by Badgr about the badge, including the Badge identifier on Badgr, and the URL on Badgr where the badge information is located.

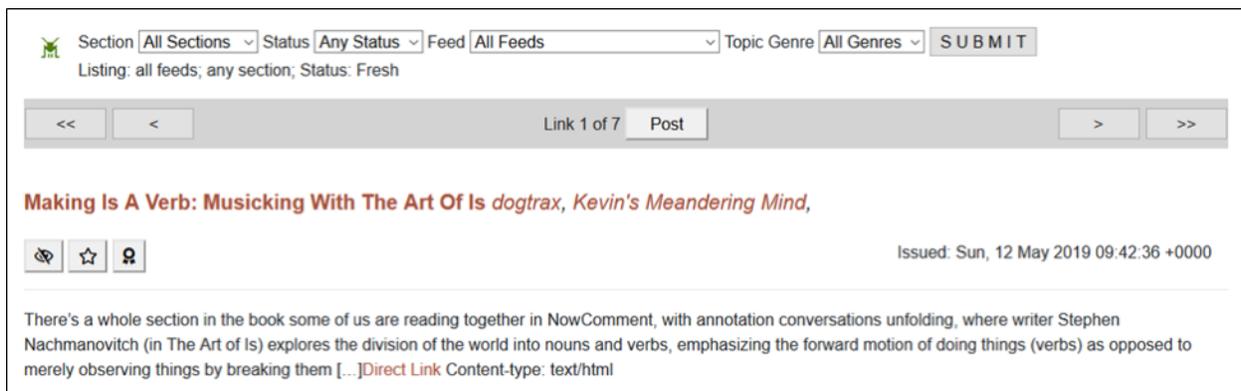


Figure 7: gRSShopper RSS reader with option to award a badge. Screen shot from gRSShopper.

To award the badge, the student's submission is reviewed, and if appropriate, the badge is awarded. A course instructor does this by means of a gRSShopper tool called the 'Viewer' or 'Reader', which displays the contents of recently harvested RSS feeds. Links, recall, are associated with Feeds and Authors, as well as with the Task associated with a Badge. By clicking the 'badge' icon (right next to the star in the image) the instructor awards the badge to the Author. When the badge is awarded,

gRSShopper employs the Badgr API to issue the badge to the Author. The Author is identified by email, the badge by the previously defined Badge identifier, and the evidence in the form or the URL in the Link submitted by the author.

Notification and Recording Awarded Badges to a Blockchain

Finally, the badge recipient (designated here as the Author) is notified that the badge has been awarded. Badgr automatically sends an email when the badge is awarded. Additionally, gRSShopper may also send an email.

A more direct mechanism also employed by gRSShopper is to use a specification called WebMention. WebMention is a W3C specification use to allow individual websites to notify each other automatically when they cite each other (World Wide Web Consortium, 2017). WebMention works completely behind the scenes in gRSShopper and doesn't require an additional human step to operate.

Web pages that support WebMention specify a Webmention link in the HTTP header or document head. A publishing service such as gRSShopper can send a notification by sending a 'ping' to that URL in the form of a simple HTTP request with the URL of the published document. The recipient of the ping may request the page; when it does so it looks for metadata in the page text describing the reference; in the 'Indieweb' response post that is marked up with microformats that describe the response and what it is in response to.

gRSShopper, when it publishes a Badge, sends a ping to the URL specified in the Link submitted for that badge. The badge recipient may respond to the notification however they wish; a typical use case would be to post the badge image to the page being awarded the badge.

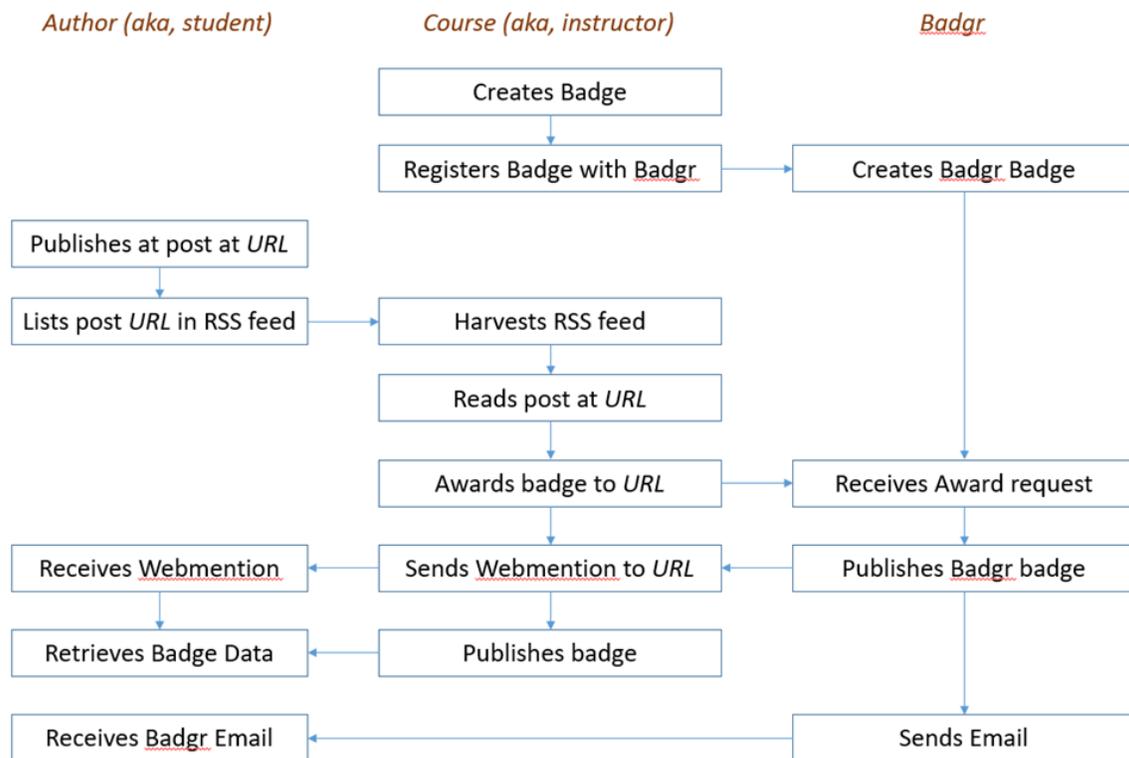


Figure 8: Badge award and notification workflow.

After notification, badge awards are recorded to a data store. One mechanism employed elsewhere is to record such activity to a learning record store (LRS) using the Experience API (xAPI). In the current case, however, we constructed a blockchain, and recorded it there (note that most online course services would deploy a third-party blockchain service; we created our own for testing purposes only).

A blockchain is essentially a series of records; in our case, the records consist of information about awarded badges. The data may be digitally signed, though for the present purpose we skipped this step. A collection of records may then be grouped together to form a 'block'. This block is then given a unique identifying key by means of an algorithm that encrypts the block and generates a fixed-length 'hash' of the block.

Blocks are chained together by means of this hash. When one block is created, the hash from the previous block is included with the record data, along with a 'nonce', which is a one-time value created specifically for the purposes of creating a new hash. All of these – the previous hash, the data and the nonce – are used to create the hash for the current block, which will then be used to create the next block. In this way, the blocks are chained together.

The purpose of a blockchain is to enable multiple institutions to share and add to the same data record. Each of them has a copy of the full blockchain, and may add data to it. When new data is created (for example, a new badge is awarded), all institutions are notified, and each of them adds the information to the newest block they are creating. When the block reaches a certain size (in the case of gRSShopper, it is five records, though this would obviously be much larger in production) then the institutions race to create a hash using a nonce to satisfy a certain condition; the resulting nonce is called the 'proof of work' required to create the hash.

The idea is that the nonce is difficult to produce but easy to verify. In the base of gRSShopper, it was necessary to produce a nonce such that the resulting hash would begin with '0000'. The first institution to produce such a hash distributes the newly formed block to other institutions, who check the block, verify that it is acceptable, and they all begin to produce the next block.

```

Save Copy Collapse All Expand All
length: 1
chain:
  0:
    index: 0
    timestamp: 1545417481
    transactions:
      0: {}
      1:
        recipient:
          email: "stephen@downes.ca"
          name: "Stephen Downes"
        sender:
          email: null
          url: "https://eL30.mooc.ca/"
          name: "E-Learning 3.0"
        badge:
          badge_entityid: "m1-bQL4ISHeL_oPGLKgDYw"
          badge_url: "https://eL30.mooc.ca/badge/1/21812"
        evidence:
          url: "http://www.downes.ca/post/68924/rd"
      2: {}
      3: {}
      4: {}
    proof: 761
    previous_hash: "BNuBNG87FYj8zpoRHPm/e60uGNTK1kckooXE+tRPPTs="

```

Figure 9: Sample blockchain output from gRSShopper badge blockchain recording.

It should be noted that the information stored in the blockchain, as noted in Figure 9, is the same as the information stored in the badge itself, and as such, records the name and email of the badge recipient. It is arguable that this could violate the ‘right to be forgotten’ provision of the General Data Protection Regulation (GDPR). In application, it is likely the blockchain record would contain an encrypted hash of the data, the key to which could be forgotten, and the content of which would be unknowable without already knowing the content of the data in the first place.

It may reasonably be asked why a blockchain was constructed for this trial, since many public blockchains already exist. Part of the reason was complexity. Libraries for accessing and using these public blockchains need to be installed and configured. This introduces a range of issues peripheral to the main line of enquiry. By creating a minimum viable blockchain that could be accessed from the MOOC application these issues were elided, and did not distract from the proof of concept.

Results and Discussion

Course evaluators reported that “general statistics were also collected through the gRSShopper aggregator as follows: newsletter subscriptions (177), harvested feeds (15), with the number of unique visitors to the E-Learning 3.0 course website reaching 3000” (Fournier, Molyneaux, & Kop, 2019). The evaluators found that all modalities of the distributed architecture were used by course participants.

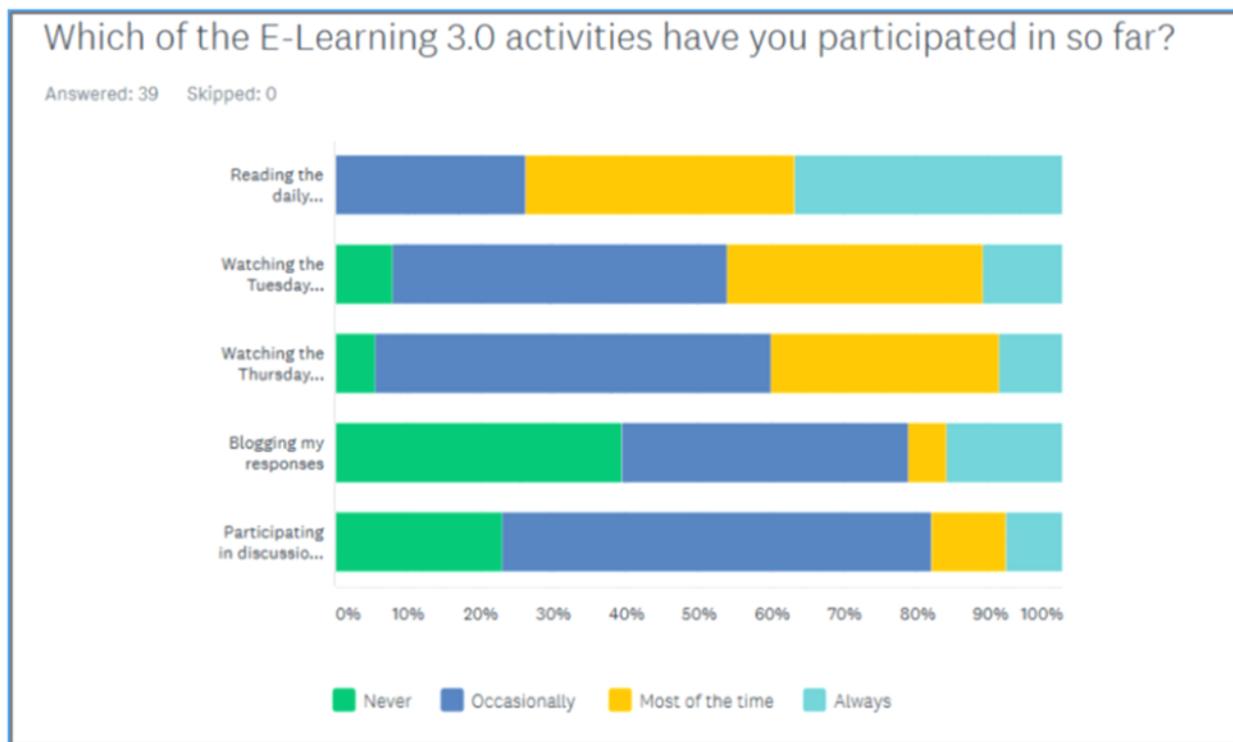


Figure 10: Preferred mode of interaction in the E-Learning 3.0 MOOC
(Fournier, Kop and Molyneaux, 2019).

Course data showed that 290 posts were created as part of the course. More than half of these were either external resources added to the course by the instructor or articles authored by the instructor. Twenty-eight active student feeds were harvested. Sixty-three posts were authored on participants’ personal learning environments and harvested by the course aggregator using the #el30 tags. Another 30 posts (approximately) were authored on participants’ personal learning environments and added manually by the instructor.

Obviously, a larger participation rate would have provided for more robust results and a better test of the software. That said, since the software was being developed during the course, a larger participation rate might actually have slowed down development, since some elements still needed manual input before the software was complete.

In the original design of the course, the intention was that students would submit their own proposals for tasks, emulating the model of the ds106 assignment bank (Levine, 2014). Unfortunately, because of the low number of participants, tasks were not submitted, and, hence, all tasks were created by the instructor. Nonetheless, the tasks were undertaken with enthusiasm by the participants.

Similarly, the low number of students (and the newness of the technology) made testing of the Webmention impractical. Though pings were sent to badge recipients, none of the participants appeared to be using Webmentions, and so the link back to the badge was not enabled (the Webmention functionality was also turned on for the inclusion of any post in the newsletter, and not only for badges; later testing of the same software on another site confirmed its functionality).

It was not expected that all aspects of the course design could be implemented during the actual offering of the course, and this turned out to be the case. Although the blockchain software had been authored in the sprint, integration took longer than expected, and so it was not functioning until after the course. As a result, course participants did not have the opportunity to experience being awarded badges. Similarly, the publishing of course resources, such as posts, to a distributed content network was accomplished only in test cases, and not in the course software itself.

Nonetheless, the successful development of the distributed badging infrastructure in a cMOOC offers the possibility of even more experiments in the future. For example, since the contents created by course participants are openly accessible, and since the badge infrastructure is decentralised, anyone with the appropriate technology (currently, gRSShopper only) could award badges for course contents, and save them to the same blockchain as the 'official' badge, thus enabling decentralised recognition for course achievements with no central criteria.

Additionally, again because course, content and assessment are distributed, parts of one course could be merged with parts of another course to create decentralised hybrid courses, which could in turn support their own assessment system, again recording the results to the same blockchain.

The distributed badging infrastructure described in this paper offers a range of potential applications for learning and development. The employment of blockchain-based certificates provides the measure of security and verifiability needed worldwide. Combining this approach with a model of open online education makes such certification more widely accessible. The digital content and badge-based recognition create portable credentials easily shared with potential employers. And the decentralised design of the badge infrastructure enables institutions in all parts of the world to participate as equal partners in the education infrastructure.

Future work will investigate some of these possibilities, as well as with integration of this model with (for example) IMS Learning Tool Interoperability and the Experience API (xAPI), as well as with distributed content networks such as IPFS or GitHub. A fully-functioning trial will be conducted in a future version of the MOOC. With a functioning model in place, so we know what a solution would look like from beginning to end, we can now ask in an informed way whether the combination of badges and blockchain are the best, or even an appropriate, way to recognise achievement in MOOCs.

References

- Albeanu, G. (2017). Blockchain technology and education. In M. Vlada (Ed.), *The 12th International Conference on Virtual Learning ICVL 2017* (pp. 271-275). Bucharest University Press. Retrieved from https://www.researchgate.net/profile/Snejana_Dineva/publication/327557908_The_Benefits_of_Combining_Social_Media_and_e-learning_for_Training_Improving_in_FTT_Yambol/links/5b966f34a6fdccfd543a4e6d/The-Benefits-of-Combining-Social-Media-and-e-learning-for-Training-Improving-in-FTT-Yambol.pdf#page=271
- Bates, T. (2014, 10 13). *Comparing xMOOCs and cMOOCs: Philosophy and practice*. Retrieved from <https://www.tonybates.ca/2014/10/13/comparing-xmoocs-and-cmoocs-philosophy-and-practice/>

- Børresen, L.J. & Skjerven, S.A. (2018). Detecting fake university degrees in a digital world. *University World News*, 14 September 2018. Retrieved from <https://www.universityworldnews.com/post.php?story=20180911120249317>
- Conole, G. (2013). *Designing for learning in an open world*. Springer: New York.
- World Wide Web Consortium (2017, 1 12). *Webmention*. Retrieved from <https://www.w3.org/TR/2017/REC-webmention-20170112/#respecHeader>
- Fournier H., Molyneaux H., Kop R. (2019) Human factors in new personal learning ecosystems: Challenges, ethical issues, and opportunities. In C. Stephanidis (Eds.), *HCI International 2019 - Posters. HCII 2019. Communications in Computer and Information Science, 1034*, (pp. 230-238). Springer, Cham.
- Gräther, W., et al. (2018). Blockchain for education: Lifelong learning passport. *Proceedings of 1st ERCIM Blockchain Workshop 2018*. Retrieved from https://dspace.wineme.fb5.uni-siegen.de/bitstream/20.500.12015/3163/1/blockchain2018_07.pdf
- Hood, N., & Littlejohn, A. (2016). *Quality in MOOCs: Surveying the terrain*. Commonwealth of Learning. Retrieved from http://oro.open.ac.uk/50931/1/2015_QualityinMOOCs-Surveying-the-Terrain.pdf
- IMS Global. (2018). *Open Badges v2.0 IMS Final Release*. Retrieved from <https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/index.html>
- IMS Global. (2018a). *Open Badges Baking Specification IMS Final Release*. Retrieved from <https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/baking/index.html>
- Lab, M. M. (2016, 10 24). *Blockcerts — An open infrastructure for academic credentials on the blockchain*. Medium. Retrieved from: <https://medium.com/mit-media-lab/blockcerts-an-open-infrastructure-for-academic-credentials-on-the-blockchain-899a6b880b2f>
- Levine, A. (2014). A MOOC or not a MOOC: ds106 Questions the Form. In S. D. Krause, & C. Lowe, *Invasion of the MOOCs: The promises and perils of Massive Open Online Courses* (pp. 29-38). Anderson, South Carolina: Parlor Press. Retrieved from https://s3.amazonaws.com/academia.edu.documents/33176016/invasion_of_the_moocs.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1558035008&Signature=7AuLgZdYikFcPuCclRBj3fOyZUY%3D&response-content-disposition=inline%3B%20filename%3DAnother_Colonialist_Tool
- McAuley, A., Stewart, B., Siemens, G., & Cormier, D. (2010). The MOOC model for digital practice. University of Prince Edward Island. Retrieved from https://oerknowledgecloud.org/sites/oerknowledgecloud.org/files/MOOC_Final.pdf
- Mozilla. (2012, August 27). *Open badges for lifelong learning*. The Mozilla Foundation and Peer 2 Peer University, in collaboration with The MacArthur Foundation. Mozilla Foundation. Retrieved from https://wiki.mozilla.org/images/5/59/OpenBadges-Working-Paper_012312.pdf
- Nakamoto, S. (2009). *Bitcoin: A peer-to-peer electronic cash system*. Retrieved from <https://bitcoin.org/bitcoin.pdf>
- Sharples, M., & Domingue, J. (2016). The blockchain and kudos: A distributed system for educational record, reputation and reward. In K. Verbert, M. Sharples, & T. Klobučar (Eds.), *Adaptive and Adaptable Learning. EC-TÉL 2016. Lecture Notes in Computer Science, 9891*, (pp. 490-496). Springer, Cham. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-45153-4_48
- Swan, M. (2015). *Blockchain: Blueprint for a new economy*. O'Reilly.
- University of Sussex. (2012, 8 29). *Improving Moodle import. Part 1: The database schema*. Retrieved from <http://blogs.sussex.ac.uk/elearningteam/2012/08/29/improving-moodle-import-part-1-the-database-schema/>.

Author:

Stephen Downes is a Researcher, Digital Technologies Research Centre, National Research Council of Canada. Email: stephen@downes.ca

Cite this paper as: Downes, S. (2019). Recognising Achievement with Badges and Blockchain in a Connectivist MOOC. *Journal of Learning for Development* 6(3), 273-286.