

# Blockchain And Content DAM: Myth, Reality And Practical Applications

Author: Ralph Windsor

This paper was first published on DAM News in December 2017

## Introduction

Recently, I have read a few articles which make some critical remarks about blockchain for DAM. There are some reasonable points advanced, however, there are also misunderstandings and myths about what blockchains are and how they might be applied to Content DAM. In fairness, there is an even greater amount of hyperbole, overblown expectations and misunderstandings from the supporters of this technology also.

In this piece, I am going to try to present a balanced case as to why blockchains will be relevant for Content DAM, but from a more practical, implementation-oriented perspective as I have yet to read much that explains that side in any depth. Some of this article is technical in nature, but I would expect most readers to be able to grasp the basic points, irrespective of their background.

## The Context Of DAM And Blockchains: Why Should We Care About This?

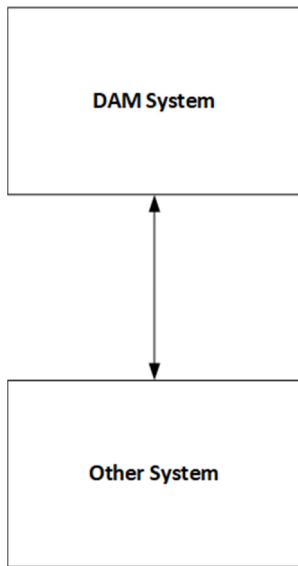
As we have covered at some great length this year (and before). The action in Digital Asset Management is [now moving towards Digital Asset supply chains at quite a rapid pace](#). All the clients I have spoken to about significant DAM initiatives in the last couple of months have a digital supply chain dimension to them where either multiple solutions are required or they are composed of lots of moving parts where the DAM is just one piece. Even among the solutions I have seen where there is just one primary vendor, there are nearly always multiple integrations with an array of different ancillary components.

As a result, interoperability is now simultaneously both a hot topic and a bottleneck; i.e. a source of considerable value and an untapped opportunity to improve efficiency and bottom-line ROI. The sharp end of DAM implementation is also currently proving to be a painful place for many, as I will illustrate. I believe, however, that blockchains might offer some solutions, but only for those willing to look at the facts of what they are and take a more pragmatic approach. Lastly, there are still impediments to using them which will need to be addressed (and they should be acknowledged also).

## DAM Interoperability: Past And Present

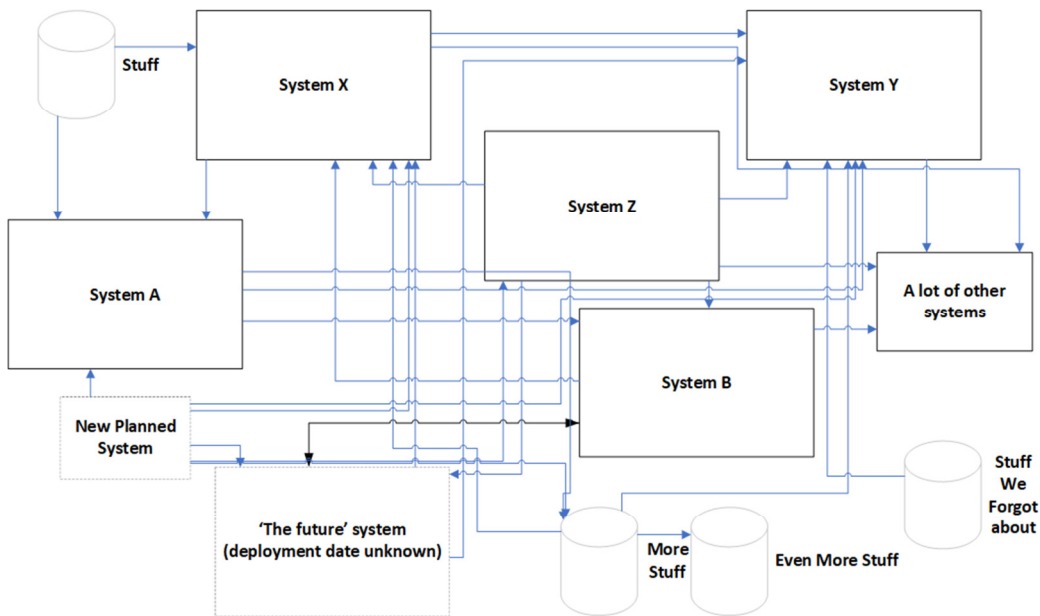
When the word 'interoperability' comes up, a familiar refrain from either vendors (or the end users who are their internal sponsors) goes something like 'that won't be a problem, XYZ application has an API' (Applications Programming Interface for the uninitiated). Nearly all DAM applications now possess an API and so also do the majority of other modern solutions that DAMs have to integrate with. This is analogous to assuming that just because you have a cable for an appliance, there will be somewhere compatible to plug it in. It isn't necessarily that straightforward (and this is particularly true in the fragmented world of Content Digital Asset Management).

In the past, the extent of most DAM user's interoperability issues could be visualised like this:



Interoperability requirements were occasional, ad-hoc and could be dealt with via the aforementioned APIs, or sometimes using 'belt and braces' methods like pushing data files to a counterparty system every 24 hours.

In the brave new digital asset supply chain world, however, many DAM users are finding they have interoperability requirements that look a bit like this:



This might look like a visual piece of comedy you would see in a Dilbert cartoon etc, but compared with some digital asset supply chains overviews I have seen recently, it is relatively simple.

As well as schematics that keep business analysts and solution architects hard at work late into the night producing Visio diagrams, there is also a not inconsiderable amount of IT politics and negotiation between enterprises and the vendors they have chosen to work with. Some parties may get on with each other, while open warfare has broken out between others. Here are a few examples of what I mean:

- Vendor X refuses to open their API to vendor Y because of a 'commercial dispute' (which might have been due to X's former account manager running off to Y and taking their biggest clients with them).

- Vendor X cannot be replaced because their system was deployed years ago and is embedded into the client organisation (and has many loyal users who are highly resistant to change).
- Vendor Z uses a completely different API protocol incompatible with X or Y.
- In-house system A cannot be accessed externally and CSV data dumps have to be fed to X on a daily basis. The reliability and quality of these is sometimes a bit hit-and-miss.
- Vendor B has developed an API which is required by X and Z, but it is flaky and unreliable.
- Vendor B's development team claim there is nothing wrong with their API and blame X and Z. Z further asserts that the problem is between X and B, only, is nothing to do with them and are demanding professional services fees before they will investigate the issue.
- Vendor B claims 'there is no need' for X, Y and Z since their system has 'everything they need' and the client organisation should migrate everything over to B and simply dump X, Y and Z to solve their integration problems, asserting that they are all 'silos'. Some of Vendor B's users inside department 1 agree, the other half (in department 2) do not and want them replaced.
- Hybrid, customised (and now, in-house) system C was never included in the original business analysis and has its own API which is incompatible with everyone else but is 'strategically significant' (according to the CIO).
- System N was missed from the business analysis due to some miscommunication early into the project plan, isn't finished yet and no API documentation exists for it. It will be 'mission critical', however, so everything else will need to integrate with it (once it is eventually completed).

These are hypothetical examples, but they are all based on real scenarios I have encountered in the course of managing DAM implementations (or being asked to provide advice to clients). I expect a number of readers will have similar experiences. From personal friends I know who work on IT projects in other sectors like financial services or logistics, these are equally common and can get far more complex.

The key issue here is fragmented centralisation. That might sound like an oxymoron, but the problem is actually quite simple. Many DAM solutions were commissioned to centralise digital assets into a single location that would be accessible by all stakeholders, the 'single source of truth' business case. While that can work adequately for a while, in larger enterprises different groups of users go off and buy their own independent solutions, either because they don't realise anything else exists, their needs are not being addressed or because of some other internal political issue where 'owning' a dedicated DAM just for their use seems like the best option on offer.

Talking one group of employees out of continuing to use a system that has been configured and adapted for them is quite difficult (for many reasons) so out of necessity, the focus moves to interoperability and integration instead. While this is reasonable, it depends on everyone cooperating, as well as having some agreed standards, conventions and ground rules. As discussed, this is especially lacking in DAM currently and something like a level playing field is required to enable progress to be made, even if imperfect.

An increasing number of vendor personnel (and in-house IT staff where there are internal counterparts) are spending their time feeding and watering these ever-expanding integration behemoths, not to mention dealing with the political complications. While connecting DAM systems and associated technologies together should theoretically enhance the opportunities for DAM innovation, if there are no standards, they will ultimately constrain it by using up all the available time and resources just to maintain operational stability.

In the past, there has been talk of DAM standards and various initiatives to try and kick start protocols, but the flaw in all of these endeavours is that there is little to force anyone to use them (even if they did exist). No one gets paid for coming up with DAM standards and everyone has a day job which has to take priority. Getting them off the ground is, therefore, a thankless task with little in the way of recognition (much less remuneration) offered for your trouble. The only way I can conceive to get something started is to engineer a situation whereby [the sell-side does not get paid if they don't play by the rules](#).

This is where blockchain concepts could have a role and I will describe how that might work later in this piece. There is, however, currently an excessive amount of both speculative froth and negative fallacies surrounding blockchains, so it is necessary to address some of these first.

## **The Ten Second Guide To Blockchains (And The Myths That Surround Them)**

There are [longer and better descriptions of blockchains](#), but the simple explanation is that they are transaction ledgers which record digital activity. The ‘blocks’ contain arbitrary data and act like containers. Each transaction (block) is appended with a pointer to the previous one to validate it (‘the chain’). The ledger is ‘immutable’ which means existing records cannot be altered. This feature adds an element of trust to the data contained (although only that the data was committed at a given point in time and has not been tampered with). Copies of the ledger are copied to all the ‘nodes’ so it is very difficult to completely destroy data (depending on how many nodes exist). Because of this characteristic, they are unsuitable for large volumes of data, but are ideal for key elements that might need to be referenced by multiple parties (e.g. a metadata for digital assets on a digital asset supply chain).

There are many myths about blockchains and I will deal with each of those below.

### *Blockchains = Bitcoin*

This is false, bitcoin is an implementation of blockchain or distributed transaction ledger concepts. Bitcoin is one of many, blockchain is a generic description of distributed ledger protocols, it isn’t ‘a thing’.

### *Blockchains are a Ponzi scheme*

Blockchains are simply the method by which transactions on a distributed ledger get stored. You could use a blockchain to run a Ponzi scheme, but the same is true of an Excel spreadsheet (or even a DAM system). As per the previous section, blockchain is an idea for a protocol (of which there are several) and what it gets applied to could be positive, negative or neither.

### *The immutability of the ledger makes blockchains unsuitable for DAM*

As with most transaction-based databases, the ledger describes everything up to a given point in time. As such, to denote a change, a new block is created with the modified data in the same way that version control systems exist in many DAM systems. You just add a new block, the immutability characteristic has no impact on the ability to denote that one record has been superseded by another.

### *It’s impossible to track different asset identifiers in blockchains*

Using a variation on my [DAM Interoperability 1.1 protocol](#) (which I note still remains criminally ignored by the DAM industry) this problem is quite easy to resolve, like so:

```
<asset>
  <extension name = "XYZ">
    <myID>123</myID>
  </extensions>
  <extension name = "ABC">
    <yourID>321</yourID>
  </extensions>
</asset>
```

Everyone puts their metadata into the block in their own designated extensions. Each block has a pointer to the last one, so if one party has not kept to the rules, everyone in the supply chain can see who was responsible. The identifiers are arbitrary and the blockchain identifier can be used as a ‘master’ to locate each version. If anyone adds data which deletes or modifies an existing node which

they are not responsible for, it is possible to see precisely who did it and when (and then [roll back to a prior version](#)). The blockchains gives complete visibility of who did what and when to digital asset metadata as it traverses the supply chain.

#### *Blockchains consume vast amounts of electricity*

The bitcoin blockchain depends on highly complex cryptography because the use-case was to provide a type of digital substitute for scarce assets like gold (although whether or not they really achieved that is debatable). This is why huge amounts of electricity are required for the bitcoin blockchain, but they are not for others (unless they have similar design objectives). For those blockchain use-cases where something far less intensive is required (like the one under discussion in this article) the power consumption is not an issue because there is a far lower burden of proof to validate supply chain transactions and the cryptography can be consequently less demanding. Examples of more business application-oriented blockchains are [Hyperledger](#), [MetroGnomo](#), [R3 Corda](#) and there are a lot of others that are now emerging.

#### *Blockchain is a massive bubble and when it pops, interest in blockchain will fade away*

As discussed, blockchain is an idea for a protocol, it's not even a single technology. The core principle behind it is not new, there have been distributed ledgers for quite some time now. What is at the heart of this is digital transactions between multiple parties who potentially do not trust each other. This is quite a simple idea and not particularly exciting nor 'revolutionary'. For this reason, while the current bubble in tradable digital assets that use blockchains might burst, because the problem is so universal and quite basic in-nature, it is unlikely to ever go away. To make a comparison, many dot com businesses went bankrupt, but the internet technology market has grown relentlessly for multiple decades because it is generically applicable for lots of different purposes and will remain so for an indefinite period.

#### *GDPR means no one can use blockchains*

General Data Protection Regulation (GDPR) is an [EU initiative](#) relating to data privacy and protection. One of the aspects of the legislation is that personal data cannot be stored in a form that is immutable. To make use of blockchains for DAM, it is not necessary to store everything on the blockchain, only the data that needs to be shared between different systems (and this is a key principle that underpins most well-designed conventional APIs already). It is true that a great deal of care will be required when committing data to an immutable data source to verify GDPR compliance. The legislation itself does not preclude the use of blockchains, only that you can't keep everything on them, which isn't advisable anyway in the same way APIs don't expose all types of data to third party systems which need to consume their data.

#### *AI & blockchain will change the world – they are 'the future'*

AI and blockchain have no direct relationship with each other. An AI system might utilise the data stored on a blockchain to analyse activity and draw inferences from it, but other than the fact the up-tick in interest in them has occurred at the same time, there is nothing tangible that connects them together.

#### *AI is a better technology than blockchain*

See previous point, a blockchain is a transactional ledger of digital activity. AI tools make decisions based on incomplete information and use statistical mathematics to assess their probability (and propose actions accordingly). Blockchain is about definitive and provable facts recorded in an immutable form. This is like saying 'chairs are better than tables', i.e. it's a flawed comparison because the purpose of the two and the intent behind them is not the same.

#### *Blockchain will change everything, they are the future of money*

I have covered this [in greater detail on Digital Asset News](#). The brief summary is this: blockchains are just logs of digital transactions of anything you like, money if you wish and something else, if you don't. They are not money, they are not the future, they are simply a log of activity in digital form. As described earlier, distributed ledgers have existed for some time now. The recent popularity, however, has made it possible for a far cheaper and open source form of this technology to be accessed more widely.

## Practically Applying Blockchains To Digital Asset Supply Chains

The earlier diagram (and list of example political issues with integrating DAM systems) should offer a clue about why a blockchain solution could help with DAM interoperability, however, for those who are still not entirely sure, below I present a rough idea of how this might work.

Instead of using ad-hoc arbitrary connections between systems, the blocks on the blockchain become a shared repository to keep metadata that might be required by different products. Using DAM Interoperability 1.1 protocol referred to above (which given the glacial progress towards other DAM interoperability standards, maybe only partly humorous) the rules can be made quite straightforward initially, so each participating supplier has a pre-designated node in the data which they can write to.

For example, let's say we start with this:

```
<asset>
  <extension name = "X">
    <myID>123</myID>
    <metadataFromX>A value</metadataFromX>
  </extensions>
</asset>
```

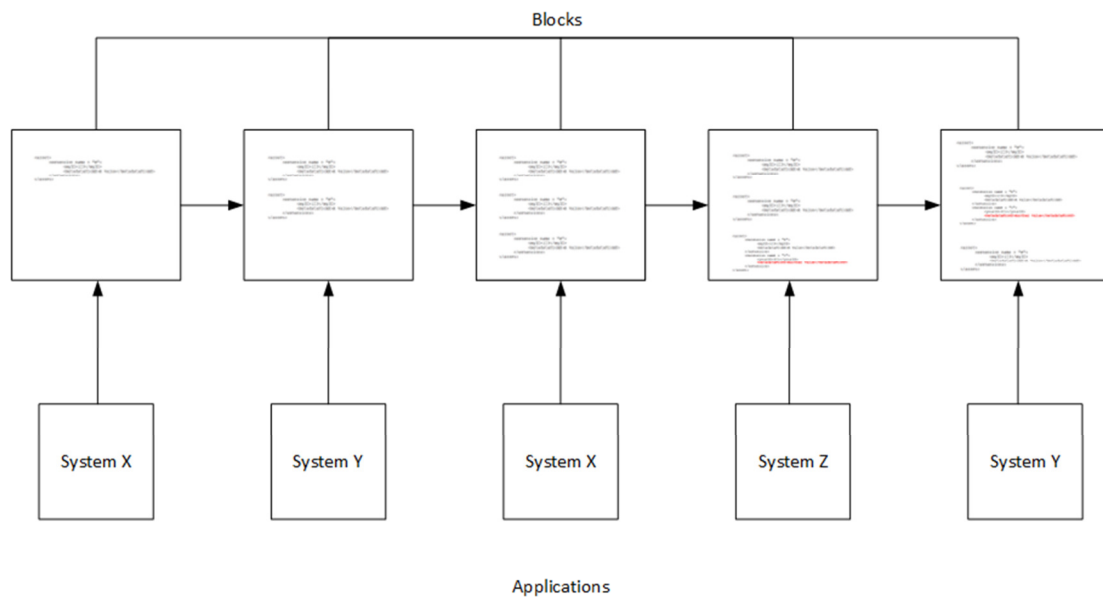
A counterparty can read the data from the block, identify the data it requires (if any) carry out the required operations and then write back to another block using a modified copy:

```
<asset>
  <extension name = "X">
    <myID>123</myID>
    <metadataFromX>A value</metadataFromX>
  </extension>
  <extension name = "Y">
    <yourID>321</yourID>
    <metadataFromY>Another value</metadataFromY>
  </extension>
</asset>
```

As can be seen above, System Y adds its data to what was supplied by System X. Y can read X, but not overwrite it (without breaking the rules, at least).

The transaction ledger provides ready-made versioning and the ability to see who made a change and when. Innovations like [smart contracts](#) enable validation to be confirmed directly by the blockchain itself, however, this isn't absolutely essential providing that participants assume responsibility for this themselves. Under normal circumstances, many managers might have doubts about whether each service provider could be trusted to do that, however, the fact that there is complete transparency over who did what (and when) makes it very difficult for the guilty party to blame someone else for their transgressions. This increases the likelihood of compliance before anything more technically sophisticated has to be introduced.

Here is an overview of the architecture:



Used in this fashion, blockchains act like a kind of shared whiteboard where each participant can read what everyone else has written and then append data to it, with a snapshot being taken after each contribution.

Theoretically, this could be replicated by some kind of neutral third party database that is solely responsible for storing packets of API data. In other enterprise software markets, EDI (Electronic Data Interchange) systems have been developed for larger trading partners to achieve this goal. The cost and effort involved, however, makes it far easier these days to use something like a blockchain where many of the problems have already been solved (and a number of other innovations are being worked on independently which can be incorporated more cost-effectively). In the same way that no one would now develop their own alternative to http, web servers and browsers, so it makes little sense to do the same with distributed ledger technology in 2017.

### **Drawbacks, Risks & Complicating Factors**

There are a number of risks, drawbacks and a complications that need to be resolved. Below I have listed a few.

#### *Lack of a Content DAM blockchain*

At this time, I do not know of any dedicated public blockchain which is used by DAM service providers (whether vendors or others). Although a private one could be implemented by an enterprise, even with the current uptick in interest in Content DAM, it is unlikely that the cost and complexity could be justified by most, unless it was shared with some other wider initiative. Ideally this needs to be a cross-industry undertaking, although this is unlikely to happen if left to vendors alone, for the same reason that they have failed to agree interoperability standards (i.e. lack of commercial motivation). I am currently discussing this with some people in DAM, but there is no tangible product at the stage.

#### *Limitations Of The DAM Interoperability 1.1 Standard*

This is a semi-humorous idea I devised for an article about [how little progress with DAM interoperability there has been](#), it doesn't exist, no one uses it (as far as I know) and the only two key benefits are:

- All the data is stored together in a single packet.
- Each service provider's data is clearly identified and distinct from everyone else's.



The only issue this issue solves is access to API data. Apart from a standard method to read it, interfaces still need to be written by every single system which must consume that data. This method helps alleviate some of the politics involved where one party has access to data and cannot or will not provide it to others, but it is not the interoperability 'silver bullet'.

One of the biggest issues with interoperability standards is the circular nature of these debates and a lack of acknowledgement that just getting a very basic protocol is more of an achievement than it often first appears. Software developers (whether DAM ones or others) tend to flip-flop between advocating complexity or simplicity, depending on whether they are responsible for a given state of affairs or someone else is. The nature of these arguments goes a bit like this:

“Here is a very simple framework for a protocol, what do you think?”

“I’m sure we can come up with something far more useful than just that. How about we add x, y or z?”

“I don’t know about z, let’s split it into z1 and z2”

“But say x and y need to access a, b, c and d, won’t we need to handle z1 and z2 with z3 and z2.2?”

“Sometimes x,y,a,b and c, but not d will not have the required data for z.2.2, let’s add z.2.2.1, unless x==y, then we’ll allow d to access z.2.2.”

“This protocol is insane! It’s unnecessarily complex. Simplicity is genius, didn’t you know? Here is a very simple framework for a protocol, what do you think?”

I have been in meetings and read email threads etc. with debates along these lines. I note also the similarity of these discussions with the points made earlier about politics between different vendors in digital asset supply chains. Having a protocol that everyone can stick to, even if it isn’t especially sophisticated is a significant step forward, but a lot more work and negotiation will still be necessary.

#### *Accidental commit of sensitive data*

GDPR was mentioned earlier, but there is more than just personal data which is potentially risky to commit to a blockchain. Once data is written to blocks, not only can it never be erased, but a copy of it is made to every node on the blockchain. If a public blockchain is used, this means thousands of instances and absolutely no possibility of ever erasing it.

A private blockchain might be easier to allow a limited form of deletion (or at least marking blocks as ‘withdrawn’) but this does not stop one party making independent copies of the data (or just reading the withdrawn block). If destructive methods are used, there is a security risk of data getting destroyed (either intentionally or by accident). There are various methods to reduce the impact of these problems, but they will add to the complexity.

#### **Conclusion**

In this article, I have sought to debunk a few of the myths about blockchains (especially that they are more of a concept than a product, per sé).

I do not envisage blockchains solving every interoperability or digital asset supply chain problem (and as described in the preceding section, there are risks). I believe, however, that they do offer a potentially useful basis for a solution to many of the problems that more DAM end-users are now encountering as their needs diversify and become more sophisticated than has been the case until now.

[As with AI, DAM developers \(and end-users\) need to look beyond the basic use-cases](#) and apply a lot more lateral thinking to how to apply these concepts to the Content DAM problem. Digital Asset Management continues to be largely an innovation-free zone, however, that is more a choice made by those involved with it rather than because there is no opportunity available to do so.