XML, blockchain and regulatory reporting in the world of finance

Combining the strengths and weaknesses of mature and modern technologies to drive innovation and regulatory compliance

Lech Rzedzicki

Abstract

The regulatory burden for financial institutions makes it a hard environment to innovate, yet the unfavourable market conditions mean that banks must adopt latest technologies to cut costs and keep up with the market demands.

This paper aims to show a technical proof of concept of how to combine the seemingly opposite goals using a combination of tried and tested XML technologies such as XSLT and Schematron in conjunction with experimental distributed ledged technology to drive both regulatory compliance and implement innovative features such as inter-bank trade settlemnt using blockchain technology.

Table of Contents

Background

XML in regulatory reporting

XML 1.0 is by now a very mature standard. As most of the audience will know, the standard hasn't changed in 20 years. XML 1.1 hasn't been widely adopted and no one is contemplating XML 2.0 to replace XML 1.0 anytime soon.

The strenght of that is stability, great tool support, human readability and the reach.

This means XML still is, probably more than before, a good archival and reporting markup. Worst case scenario is someone will open it in 20 years time in a text editor and it will still be readable.

Enter the magical world of finance where common sense quickly disappears and is replaced by exotic, derivative products such as index variance swaps, whose main goal is to depart the client from their money.

It is hard to prove whether the complex financial product were engineered as a way to improve the bottom line of financial institutions or are they truly filling a market need for the clients such as shielding them from volatility of the equities markets or commodity prices.

What is universally true however is that the products are complex and it is often hard to figure out who is exposed to risks and to what extent.

As a result various regulatory bodies, such as the FCA have started requiring more and more reporting on the trades, positions and risks.

As it happens XML is a great fit for this purpose and XML (and more specifically XML standard called FpML) has been a de facto standard for regulatory reporting to the extent that the regulatory bodies require that the reporting be done in XML

Regulatory reporting is one of the major considerations and driving forces for any established or upcoming financial organisations in 2016.

Fintech innovation and blockchain

Another trend in 2016 is the need to radically disrupt or reinvent and optimise financial services institutions.

Even armed with a full suite of exotic financial products and masses of individual and institutional clients, in an era of negative interest rates and fierce competition from non-traditional players such as supermarkets and tech companies, financial institutions find that a combination of low yields and high regulatory costs make it impossible to retain status quo and still provide sufficient returns back to clients and shareholders.

Executives in financial institutions are desperately looking for solutions and for many they have found the holy grail by the name of blockchain.

Originally devised by the mysterious Satoshi Nakamoto for the purposes of using with bitcoin, the concept of a distributed ledger is a powerful one.

Blockchain or distributed ledger uses solid and well known cryptography around factoring large prime numbers to make it hard to calculate and verify transactions on the ledger and mathematically nearly impossible to alter them by any single contributor.

By now the technology has enjoyed a massive wave of early adopter hype and financial institution have been heavenly looking at using it for trade settlement and smart contracts.

A single project called r3cev alone has gathered over 40 international financial institutions and has recently completed the early tests with 11 of them.

it is therefore an excellent example of an innovative technology that many financial institutions would like to try and we will be using it in the paper to showcase how innovation, XML and regulatory reporting can play together nicely.

Distributed ledgers history

Ledgers have been used throughout human history, from Sumerian records of harvests and crop usage 4500 years ago through to the tracking of bad debts by the Medici bank in the 14th and 15th Century. Today’s modern double-entry ledgers can trace their roots back to early Islamic records in 7th Century and were first publicly codified in Venice in 1494 by Friar Luca Pacioli based on the practice of Venetian merchants of the Renaissance. Distributed ledgers represent the next evolution of these devices.

What is a distributed ledger?

A distributed ledger is essentially an asset database that can be shared across a network of multiple sites, geographies or institutions. All participants within a network can have their own identical copy of the ledger and any changes to the ledger are reflected in all copies in minutes, or in some cases, seconds. The security and accuracy of the assets stored in the ledger are maintained cryptographically through the use of ‘keys’ and signatures to control who can do what within the shared ledger. Entries can also be updated by one, some or all of the participants, according to rules agreed by the network.

It is important to understand that typically the design is such that you can only add new entries to the ledger and can not remove previous entries. If you wish to amend entries, you add a new, cerrected entry and a reference the old one.

This also means that the ledger, whether it is stored as a file or in a database, will grow over time which is both a strength and a weakness. t makes it almost impossible to falsify previously added information, but it also makes it hard when you truly need to change the information – a person changes name, or there was any other factual error when entering the information on the blockchain.

What can you use distributed ledgers for?

There are three core use cases where distributed ledgers can be used:

- Distributed ledgers can be used to secure an enormous range and type of value transactions – from simple money transfers, of which bitcoin provides the best exemplar, to complex registration of asset ownership and transfer such as in the financial services industry.

- Distributed ledgers can be used to provably convey trust between parties and verify provenance, reputation or competence – certification of university qualifications, the status of components within a complex supply chain such as an aircraft engine manufacture.

- Distributed ledger can serve as the backend execution environment, a distributed "computer in the cloud", where algorithms are executed on the blockchain rather than on any particular machine.

There is also a number of areas where people see blockchain as the holy grail and mistakenly think it can be applied to anything from preventing diamond forgeries to solving blood bank reserves inefficiencies.

As a general rule, whenever there is a part of the system that can not be put “on the blockchain”, for example physical assets, or 3rd party IT systems, a blockchain solution is unlikely to work as information can be changed easily outside the blockchain.

Blockchain challenges

There are currently a number of fundamental challenges with blockchain based distributed ledgers, particularly around scalability and privacy of information on a public blockchain (many of these are resolved in a private blockchain systems).

The *scalability challenges* are numerous - the transaction limits for blockchain based systems are low (20 per second for Ethereum, a common next generation blockchain system), the energy requirement for proof of work based consensus systems is huge (the Bitcoin network is estimated to use as much energy as the republic of Ireland), although distributed amongst many parties, the requirement for each node to hold a copy of the full ledger means the cost of storing data on the blockchain is computationally high (although this is mitigated by other distributed storage systems such as IPFS).

Privacy challenges exist as a result of each node holding a copy of the ledger – all transactions are visible to all parties currently unless encryption is used on the data. The immutability of the blockchain also raises a potential legislative challenge around the European “right to be forgotten”. How do you erase records on the blockchain?

Due to the above concerns, usually the blockchain stores just a hash – a signature of the transaction or information and the private information is stored separately in a wallet and protected by a private key. This is turn means that if a private key is lost, it makes it impossible to verify any previous or future information for that key.

Distributed ledger can only be used to verify the information on the blockchain. If the data is falsified or tampered with before or outside of blockchain, a distributed ledger can only verify that the data is not on the ledger. An example here is diamonds and Everledger. Everledger takes several measurements of the diamonds (the way it is cut, the colour, size etc) and puts on the distributed ledger. If the diamond is lost or stolen and surfaces ever again, it can be easily identified by taking the measurements again and comparing the database stored on the distributed ledger. This parallel verifiable digital certification on the blockchain can assure buyers and sellers of the provenance of the item.

Unfortunately, all it takes for a savvy thief to avoid this is to change the qualities of the diamond slightly – in the process the diamond might lose some value, but it is very likely that a thief will prefer that to getting caught. To sum up and generalise this issue, any inputs or outputs that are not on the blockchain are vulnerable hence the movement to smart contracts and trying to do as much as possible inside the blockchain.

More research and good education for decision makers is key here – this will prevent using blockchain where it is not appropriate – for example where data changes often, or high performance or efficiency is required. Many of these difficulties are the subject of active research projects, and in the UK EPSRC are launching a call this summer to support the research further with up to 10 small scale project grants with a total of £3M.

The opportunity for distributed ledgers

Despite all the identified challenges, the opportunities for distributed ledgers are potentially huge. Systems such as Ethereum, Eris, NXT, R3CEV, with integrated smart contracts offer the ability to place complex business logic on a public or private system where they can be triggered by transactions – contract signing, approval of work, external events etc.

The most direct beneficiary of distributed ledgers technologies are the platform developers (initially developing the private blockchain solutions) and the application developers. In common with many open source companies, the platform business model is to sell proprietary value around an open source core. One analogous example is Red Hat – a linux distribution provider that offers their core product free of charge and then offers consultancy, integration and development services to enterprises – recently they were valued at $2B.

Assuming there is large scale adoption, application developers stand to create value in the same way that the app ecosystem has developed on the Apple and Android mobile platforms.

Distributed ledgers allow for complex business processes between parties who do not implicitly trust each other to be automated and hence significant cost savings to be achieved in many sectors. The classic example is financial instrument trade reconciliation but supply chain is another commonly talked about use case. In many of the use cases, the core focus is about cost reduction through the removal of unwanted middlemen or through the reduction of duplicated effort across untrusted parties. Distributed ledgers may also improve transparency and efficiency, by ensuring that regulators and other third parties have full, real time views of transactions.

Smart contracts and Distributed Autonomous Organisations

Smart contracts are beyond the scope of this short presentation, but many in the world of blockchain speak about distributed autonomous organisations (DAO), written in code and deployed on the blockchain these lend themselves strongly to new business structures or digitisation of e.g. cooperatives. How these will develop is unknown at this point, but initial DAO's have raised millions of pounds in blockchain crowdsales.

An example of that is Ethereum, where any computation cost can be covered by spending a virtual currency – Ether. Software developers wishing to execute their programs on Ethereum network can choose to outright buy the Ether computing units or provide the computational resources themselves and even sell the excess power in exchange for Ether/money. Such an infrastructure is somewhat similar to cloud services provided by Amazon or Google, where the price is set by the factors such as electricity costs and demand, but with distributed ledger, it is much more fair to smaller players and there is no single entity that can control the network and switch off an application.

The existence of such networks enables the execution of "Smart Contracts" - autonomous code running on the blockchain (as opposed to a single machine). Given certain conditions, the code can execute automatically.

A simple example here is bond coupon payment – in a typical bond issue, the buyers buy the bond for 100% of the price and the issuer repay the bond in instalments.

Traditionally this involves issuing paper certificates of bond ownership and manually sending money via cheque or bank transfer every month and keeping track of what's been repaid etc. Assuming that all the participants – bond issuer and the buyers are also participating in the same distributed ledger network that is capable of executing smart contracts and sending virtual currency, the process can be simplified vastly, possibly even removing the need for intermediaries such as banks (for sending money) and law firms (for writing up the contracts). In such a scenario a bond issuer, would issue a smart contract to be viewed, audited, verified and accepted by the buyers and upon accepting, the funds would be automatically transferred to the issuer. Likewise every instalment the issuer would automatically pay back the coupon payments.

Smart contracts can enable a whole range of scenarios, from distributing aid money, through voting, secure communications and probably a number of areas that have not been discovered yet. Still the challenges described above remain.

The successful exploitation of smart contracts requires solving the technical challenges, but also changing laws and regulation and ensuring that the disruption caused by the automation has a net positive effect on societies and the economy.

Escape velocity for financial institutions

To many inside and outside the finance industry, the suffocating combination of a low yielding market and the burden of regulatory reporting may feel like a fatality combo from Mortal Combat.

It isn't and like many combos in fighting games it can be blocked or better yet countered.

What this paper describes is a proof of concept technical solution to solve two seemingly opposite problems - use tried and tested XML technologies to solve compliance issues for financial institutions while at the same time allow for innovative technologies such as distributed ledger to be used alongside.

Thesis

XML is a very mature and stable standard. It has built a great ecosystem of technologies that enable financial institutions to reliably solve their regulatory reporting requirements.

XML does allow mixing in with modern ideas such as distributed ledger or modern DevOps stacks such as Docker and modern noSQL solutions such as Marklogic and in fact makes it easy to do so, thanks to a few surrounding standards such as XPath, XSLT, Xquery or Schematron.

The aim of this presentation is to show that XML is a great fit for financial institutions to deliver both the business as usual activities such as regulatory reporting and to explore new areas such as distributed ledger at the same time.

Technical Description

This section describes a sample journey of a financial transaction - a client requests FX swap between a client and a bank, the bank executes the transaction, which is then published to the distributed ledger and the authenticity is jointly verified on the distributed ledger by the client, bank and the regulatory body.

To best illustrate this, there are three separate instances running blockchain - one for the client, one for the bank and one for an imaginary regulatory body called the Fictional Compliance Authority.

We aim to show how to add a transaction, how to sign it, add it to the blockchain, how to prove the authenticity of a given transaction, how to run a few basic tests for the correctness of the message (using Schematron), what happens when a bank or a rogue party tries to falsify data on a blockchain and finally showcase a few good use case where XML technologies show their strengths - transforming from raw CSV input to FpML, transforming from FpML to a hash and plain text ledger, generating reports and graphs.

Infrastructure

Financial institutions have stringent requirements about robustness of the infrastructure. As result they often go for the tried and tested technology as opposed to the cutting edge. Our early proof of concept was a setup of Docker containers running Python code + open source database. None of that proved to be a good fit and the following setup more accurately depicts the needs and wants of the IT managers at the financial institutions that we talked to:

- Operating System: Digital Ocean Virtual Machines running CentOS 7 Linux. No special requirements here, as long as the Operating System is capable or running Java.

  Possible environments where this proof of concept can run (with slight modifications) are: raw Linux Debian or Ubuntu, Linux on top of Docker, Amazon AWS, Azure etc.

- Blockchain and "glue" code - Java on Java Runtime Environment 8. We opted for Oracle as this caused least amount of problems and mimics the bank environments that we know of. We are not using any specific version 8 features so this could be downgraded to version 6. As discussed earlier implementation for other platforms and languages are possible - for example, we have started our proof of concept using Python and most of Bitcoin code is written in C.

- Marklogic 8. It is being described as the enterprise noSQL database, has excellent XML technologies support, has replication features that we needed to implement anyway. Having said that, is it possible to use another data persistence mechanism, with additional work.

We chose to run three nodes: chain1.kode1100.com, chain2.kode1100.com and chain3.kode1100.com to represent three types of institutions - someone initiating a transaction (the client), someone facilitating, executing, and reporting the transaction (the bank), and an independent regulatory body which we called FCA (Fictional Compliance Authority).

To simplify these will be referred to as node 1-3 or client, bank and FCA respectively.

The minimum setup is a single node and there is no theoretical limit to the maximum amount of nodes, although some steps, such as reaching consensus within the network will take longer as the network size increases.

To accurately mimic a realistic setup that could actually reliably work in production for a major financial institution, we have configured a high availability Marklogic cluster, where both the configuration, transactional data and the blockchain itself are protected from failures using Marklogic enterprise features such as clustering and automatic failovers.

In addition to that, it possible and very easy to configure additional, automatic cluster replication, to ensure additional availability in different data centres or different time zones.

Sample operations

Adding a new transaction

Verifying the business rules of a transaction

Summary and the future

We have showed that XML is still a good fit for business as usual activities such as trade onboarding and trade reporting. We also showed how strengths of the XML ecosystem can work together with new and emerging technologies, using distributed ledger as an example.

The future steps in this area will largely be market driven, possible next steps include open sourcing some or all of the blockchain code for the community to use and evolving the blockchain code to enable distributed,verifiable code execution. Such a development would then allow applications such as smart contracts and autonomous distributed organisations, which are a bit beyond the scope of this presentation, but we are more than happy to discuss them.