The Bank of the Future

By

Willi Brammertz, Allan I. Mendelowitz, and Luka Müller-Studer*

July 17, 2017

*The authors are respectively:

President, ARIADNE Business Analytics,

President, ACTUS Financial Research Foundation, and

- Co-founder and Partner, MME;

Former Lecturer, Institute of Financial Services Zug (IFZ)

The Bank of the Future

Contents

Bitcoin, Block-Chain and Smart Contracts

Interest in fintech has focused on three evolving areas: digital currencies (Bitcoin, Ether, etc.), distributed ledgers (Block-chain, Ethereum, etc.), and Smart Contracts.

Bitcoin attracted early attention because it succeeded where the alchemists of old failed: Making money (gold) out of ingredients with little intrinsic value. While the appeal of Bitcoin has diminished somewhat, broader interest in distributed ledgers has continued to grow. The initial appeal of Block-chain (Bitcoin's distributed ledger) was its ability to offer both privacy and security, which are valuable attributes. In addition, from a user perspective, Block-chain creates a capability and right of direct user access to values, whereas in the present banking system that access is limited to intermediaries. However, like many useful innovations, these attributes also supported law-evading transactions such as human trafficking, money laundering, terrorist financing, drug trade and more recently collecting ransom payments.

Despite this sordid history, the security aspects of distributed ledgers have continued to hold the attention of financial firms. While it is an important issue, it is not clear how much added value a distributed ledger contributes in this regard. Banks may be accused of any number of failings; however, they have kept their ledgers with great diligence since the Middle Ages. Cases of deliberate fraud have occurred, but fraud has not been rampant. In fact, the individual cases have been sufficiently few that they have achieved a level of notoriety. Creating banks with distributed ledgers will substitute one safe system with – hopefully – another safe system[i].

An area where distributed ledgers should make a contribution is a reduction in the costs of operating a payments system. This could have a significant impact in the developing countries where inefficiencies in the financial sector are still quite large. The impact in the developed world will likely be less, but could still be significant.

The most promising innovation in fintech is the introduction of the third element: The smart contract.[ii] If done correctly this innovation holds out the promise of significant reductions in operating expenses. There will be some reductions in transaction processing costs. However, more important savings will come from reductions in analytic costs. By analytic costs we mean much more than just risk management. Managing a bank comes down to essentially managing numbers. The analysis of those numbers is central to the operation of the entire bank: treasury functions, business planning, risk management, financial statements, regulatory reporting, etc.

The appeal of fintech should be the recognition that it will enable increased operating efficiency and significant reductions in operating costs. We make the case in this article that the most important gains from fintech will come from the use of smart contracts, especially the smart financial contract. Their contribution is three-fold: 1) they are unchangeable and hence create, combined with the accounting function of the block-chain, a right which has all the functionalities of a property right; 2) they are self-executing when it comes to generating payment obligations and; 3) the same mechanisms of the smart contract that compute a contract's payment obligations for transaction processing can also reduce analytic costs. Those mechanisms can be used to compute a contract's state-contingent cash flow, which is the necessary input for all financial analysis relating to value, liquidity, and income. Value, liquidity and income are the basis of any analysis whether it is in the risk management, financial statements or regulatory reporting. This consistent base will overcome the current reconciliation problems between the different areas of analysis that are the main cost drivers of financial institutions.

The adoption of smart financial contracts will not only improve the efficiency of the operations of financial firms, it will also increase transparency and security for investors and consumers, and last but not least, for regulators.

The Current Situation in Banks

Smart contracts, as part of distributed ledger systems, securely register rights and execute payments. In this respect, they can replace current payment and messaging systems. As already indicated, these systems can be improved, especially in the banking sector.

Current technology deployed by banks operates very differently. For example, let us assume that Bank ABC and Bank DEF agree on a certain financial contract, say an interest rate swap. The two banks sign the contract and each counterparty enters the contract's obligations into its own transaction processing system. Bank ABC enters the contract into its system "abc" and Bank DEF enters the contract into its system "def". Each system calculates the expected payments. The numbers computed by each of the two systems should match perfectly, save for the sign. What bank ABC must pay be will be received by Bank DEF and vice versa.

In reality, numbers that match perfectly save for the sign are a rare event. The computed numbers typically differ because the algorithms and the software used by Bank ABC and Bank DEF to represent the transaction in their transaction processing systems differ due interpolation methods, day count methods etc. However, the computed numbers generally do not differ by much. Since it is often impossible to say who is right and who is wrong, differences are typically resolved by splitting the difference. For example, banks have rules for resolving such differences: If the difference is not bigger than $50 (or $25) USD, the counterparties agree to split the difference. Most differences are settled in this way without resort to the courts.

Fintech, with the deployment of correctly implemented smart contracts, will improve the situation. However, the benefits of such an improvement are not as significant as the potential for improvements on the analytic level.[iii] The analytic level provides the basis for understanding the financial position of the bank and the consequences of actions under the purview of almost all management functions. This understanding includes the different bookkeeping views, control of risk, forward business planning, and regulatory reporting. The cost of such activities may aggregate to as much as 80% of operating costs. One way to understand its importance is that it largely coincides with the management level of a bank.

These expenses have less to do with the high pay of executive level employees and more to do with the organizational structure of banks and the chaotic way in which data is represented, stored and used. The root of the problem is the lack of a financial contract standard, which we will explain next.

Absence of a Smart Contract Standard is the Source of the Problem

Banks tend to have myriad transaction processing systems handling the different financial contracts that make up their assets and liabilities. There might be different systems for savings accounts, demand deposits, mortgages, commercial loans, swaps, futures etc. There might even be multiple systems for mortgages, swaps and other instruments. Two financial contracts that should have the exact same cash-flow obligations, but are categorized differently from a product perspective, may be processed in different systems. Consequently, when the specific payment obligations are crunched through the different transaction processing systems they do not turn out to be exactly the same.

Each of the multiple transaction processing systems – both within a bank and between different banks - works independent of the others and is viewed as working satisfactorily. In most cases, after signing and entering a contract into the transaction processing system, the processing system takes over. Each payment is calculated and executed until maturity of the contract. The minor friction between banks caused by the small differences in individual payments are a nuisance, however, they do not put the institution itself at risk.

The more serious problems arise at the analytic level[iv]. Financial analytics is analysis of expected cash-flows under different states of the world, such as interest rates, foreign exchange rates, credit quality, et cetera. Such analysis is typically performed on the enterprise level using data that starts out in all of the various transaction processing system. However, as we have discussed, the same contracts can look very different depending on which transaction processing system they reside in. The current solution to this problem is to move the necessary data to intermediate data pools called data warehouses, data marts, data lakes, etc. The data moved into these intermediary databases are the terms or parameters of the contracts (such as nominal value, interest rate, maturity data, etc.). The process of moving these data to a central store also involves trying to standardize the meaning of the contract terms. However, what is not moved to these intermediate data pools are the cash-flow generating algorithms used by the transaction processing systems that are used to turn contract terms into actual cash flow obligations. These algorithms are left behind and lost to the analytic level's work. Consequently, all of the algorithms created for transaction processing have to be recreated from scratch for analytical purposes and incorporated into the analytical tools; examples include valuation tools, Value at Risk calculators, amortized cost calculators, etc.

Algorithms are not moved for a number of reasons; not the least of which is that the banks never gave much thought using the algorithms in the transaction processing systems for analytical purposes. They took the view that all they needed to do in order to understand what was on the balance sheets was to standardize the contract terms. A second reason is that there is no standard representation of contract algorithms with respect to logic and programming language. Data appears easier to centralize and standardize. Practitioners believe they know how to normalize and move a nominal value, an interest rate, or a maturity date. However, an algorithm cannot be efficiently centralized for analytics without a previously established standard.

However, even the ability to standardize the terminology for a contract term in a central data store is in many cases more illusion that reality. At first blush we think we understand what is meant by a term such as "MaturityDate." This conclusion is not necessary warranted. A closer examination reveals that the meaning of "MaturityDate" depends on the type of financial instrument and its contractually agreed upon cash-flow pattern. If a contract is for a bullet bond, then "MaturityDate" is the data at which the bond's entire principal is due and payable. In the case of a self-amortizing mortgage it has a very different meaning. For such a contract the initial loan amount is paid back in continuous incremental installments over the life of the mortgager and "MaturityDate" is a parameter in the amortization formula specifying the data of the last payment of principal and interest. That payment includes only a tiny fraction of the original principal amount of the loan. In the case of a contract with a requirement for a balloon payment before the "MaturityDate", the contractual cash flow ends before the "MaturityDate." The "MaturityDate" is a hypothetical end point for computing the amortization schedule, not the point in time at which the obligation is extinguished.

These examples lead to an important conclusion: in many cases financial contract data that represents the terms or parameters of a financial contract has no precise meaning without a link to the cash-flow generating function of the contract. In other words, the meaning of the attributes is unambiguously defined only in relation to a specific algorithm.

The complex starting point for all analytical tools is recreating the cash-flow generation of the financial instruments being analyzed. Since each analytic tool contains its own cash-flow generators the system is highly error prone. Results coming from different systems can even contradict each other. The cost of reconciling differing results is exorbitantly high. A disproportionally large share of the cost of operation on the analytic level falls on data cleansing and reconciling inconsistent analytical results.

The problem can only be overcome with a new type of standard for financial contracts. An ontological standard based on the exact meaning of attributes is not sufficient, since the cash-flow generating algorithms are inexorably linked to determining the meaning of the attributes. What is needed in addition are standardized cash-flow generating algorithms or in other words, a standard for smart financial contracts.

Fintech, in order to achieve its goal of cost reduction, must solve this problem. If solved, fintech has the potential to reduce the cost of analysis, as well as improve its quality. Without an algorithmic cash-flow generating standard even fintech, with all of its promise, may even exacerbate the current chaos in financial data and analytics.

Core Features of such a Standard

To understand the core features of this standard, we start with the concept of a financial contract. While most economic contracts are agreements to exchange goods or services for money, financial contracts are different. They are agreements to exchange money today for money in the future. Money as units of a currency are a pure unambiguous – single dimensional – unit of account, similar to a meter used to measure distance or a second used to measure time. A cash obligation is represented by a single number. A cash flow obligation is represented by a series of numbers along a time dimension. Because the obligations on both sides of a financial contract are expressed in money (or cash), all of the contractual obligations for all counterparties can be expressed with great precision as simple numbers[v].

Financial contracts are therefore a sequence of payments following patterns that can be fully described by computer writeable algorithms. Except for a small number of hyper exotic instruments, all existing financial contracts can be represented by less than three-dozen cash flow patterns. Furthermore, given the volumes of the different types of financial instruments that dominate financial markets, as few as a half dozen algorithms may be sufficient to represent as much as 80 percent of all existing financial contracts.

On a more abstract level, a financial contract's obligations can be described as a series of numbers generated by an algorithm. This precision means that financial contracts are better suited to be represented by smart contracts than any other type of contract. There is no ambiguity as to what the exact obligations are. There is no wiggle room for one party to refuse to pay a second party on the grounds of non-performance based on the assertion that the required quality of the good or service provided was not met. The financial contract is probably the only contract where the expression "code is law" can be applied quite generally.

This discussion might appear to the reader as only a grand idea to be developed and realized in the future. In reality, such a standard exists, not only conceptually but also in code[vi].

Smart Contracts and Finance

A smart contract should self-execute (that is, generate the contracted payments) and, to the extent possible, also enforce the terms of a contract. There are a few interesting examples available of enforcement mechanisms. For example, if a contract is for the exchange of a good in return for a payment, the enforcement mechanism can be viewed as a type of escrow mechanism. The good in question is held until the payment has been received and, only then, released by the smart contract. There are even some examples of enforcement mechanism that might work for a limited number of smart contracts. For example, a "smart contract" car loan could have the capability to disable the car purchased with the loan via a wireless signal if a payment were missed. However, it is unlikely that many such cases can be implemented fully for smart contracts in a real-world setting. Financial contracts are a cash payment today against cash payments in the future. Fully enforcing today the future cash payment obligations of a contract would defeat the very meaning of a financial contract.[vii] Making judgments about the probability of loan repayments is a core function and competence of lenders. To the extend that credit risk is mitigated with guarantees and collateral, the smart contract algorithms can faithfully represent such contract provisions and self-enforce. However, at the limit, human judgment, courts and legal proceedings will continue to have the final word in the event of a default.

In the following we will focus on the promised exchange of cash-flows without default, which can be fully represented by algorithms. For discussion purposes we will use as the example the financial contract Annuity. In shorthand, we refer to it as ANN. This type of contract amortizes principal over the life of the contract. Consequently, each payment includes more principal than the previous payment. Most U.S. home mortgages are self-amortizing and follow the payment pattern of an ANN.

A typical fixed rate mortgage works as follows: After agreeing on the terms, a bank loans to the borrower a certain amount to buy (or build) a house. The borrower is obliged to pay a certain amount on regular schedule, say monthly. Part of payment is interest; the remaining amount is repayment of principal. Initially the first payments are primarily interest, with very little principle repaid. However, due to the continuous repayment of principle, the interest part of the payment is continuously reduced and principal is repaid at an ever-increasing rate.

This simple example can be extended to variable rate mortgages, as well. For such mortgages interest rates are reset at specified periods and the payment amounts are recalculated at each reset. Other standard smart financial contracts can incorporate more complex rules, such as options or futures. However, all these examples have the same common attribute: They can be represented by computer readable algorithms that calculate the amount, time, meaning and currency of the expected cash flows.

In order to fix the current chaotic situation in financial data (and not to enhance it), fintech must be capable of dealing with such series of payments in a standardized way.

An important benefit of the approach discussed above is that the same algorithms can be used on the transaction processing level and on the analytic level. The current disconnect seen between these two levels in all banks and financial institution would disappear and with it the huge reconciliation costs. De novo fintech banks could easily implement this superior level of operations and realize significantly lower operating costs as compared to legacy financial institutions and at the same time much higher analytic transparency.

The difference between the smart contract presented here and the smart contracts that have been presented in connection with block-chains is notable. The examples of smart contracts that we have seen to date are for the exchanges of goods or services for cash, be it fiat or crypto currencies. Such contracts are not financial contracts as discussed above. With only such contracts fintech and the use of a distributed ledger are merely substitutes for legacy payment systems. The potential benefits of fintech in improving the overall efficiency of financial firms cannot be realized. Furthermore, non-public distributed ledger initiatives do not even propose to use smart contracts. For the most part they employ what is merely a proprietary messaging standard that is not that dissimilar in principle from existing financial messaging standards like FpML and FIX. These messaging templates are used to transmit only the contract terms or parameters without a standardized algorithmic representation of the cash flow generating provisions of the contracts.

Turing Complete Language vs. well Defined API´s

At the theoretical level there are two different approaches to representing smart financial contracts. The first proposal is to offer a Turing complete language. Turing complete languages enable doing anything a computer can do, which at one level is a great advantage. However, this advantage creates significant risks. This extent of this risk became apparent with the events following the introduction of the DAO on Ethereum.[viii] Within the DAO code it was possible to use instructions in a way that was not anticipated. This capability – or vulnerability - was exploited to withdraw about $50 million (USD) in Ether from the DAO's pool of funds. Because of the anonymity made possible by transacting on the distributed ledger it was not possible to uncover the "thief". However, the community decided to branch the code and isolate the "stolen" money.

We set "thief" and "stolen" in quotation marks because it is not entirely clear whether it was a theft or not. The "Ether Classic" community argues, that the move was legal, since the existing possibilities within Ethereum and the DAO were used. Code is law and anything made possible by the code is legal. The subsequent branching is considered illegal. The main community argues that it was an abuse and considers it a theft and an illegal move.

We do not need to resolve this matter here. However, the experience with the DAO made clear the risks that come with a Turing complete language. Anything is possible. Even in the absence of indisputable fraud, it is a problematic approach. As we have shown above, the problem of finance is that the same contract can be represented in endlessly different ways, which seriously complicates the ability to undertake efficient analysis, the most critical function in a financial firm. Fintech based on everyone programming his or her own contracts will not be able to realize the efficiencies that Fintech proponents claim can be realized.

The alternative to the use of a Turing complete language is the use of a well defined, fully tested, and secure standard. As mentioned above, there are less than three dozen of patterns of cash-flow exchanges in the world, and a half a dozen of those patterns can represent as much as 80% of existing financial contracts. It is possible to program these patterns, publish them publicly, and make them available under an Open Source license. As such they can be used by anyone through API´s.

It would work in practice in the following way: Any two (or more) participants in the financial market can refer to the same financial contract through such API´s, enter the contract terms (or parameters) such as notional value, interest rate, tenor etc. and run the expected events. The API would return the same exact list of future payments to all parties to the agreement.

This exercise can also be performed by all parties pre-deal as a simulation pre-deal to gain a better understanding of the transaction. The two counterparties would run the transaction under different parameters and assumptions. If the contract contains conditional clauses, for example a variable rate or some optionality, then it would be possible to run the algorithms under different scenarios (forecasts). In this case, the state contingent cash-flows would be visible and both parties could see the expected consequences of the deal under different states of the world.

After the simulations and final agreement between the counterparties the deal would be declared signed and the payment stream would be executed. The exchange of cash-flows can happen via traditional channels or a distributed ledger. The smart contract would support analysis on a continual base. This is a level of analysis and insight not imaginable today.

Important Conditions for a Standard

The following conditions are important for the success of a smart contract standard:

1. The program library and the underlying logic must be open source: The code must be useable by any participant in the financial system without any party being able to profit from rationing its use.
2. The code must be well controlled because in a world that uses distributed ledgers "Code is law". The counterparties must be absolutely certain that the code does exactly what is supposed to do in the transaction. Protecting the integrity of the code is analogous to the important work that lawyers do in reviewing natural language contracts. Hash codes can be used protect the integrity of the code.
3. The standard must be broad: The standard must cover almost all available financial contracts extant in financial markets (say more than 99%). It must be extendable as the market evolves
4. There must be a single standard: Theoretically, any standard fulfilling the three previous conditions can do the job. However, having multiple standards will defeat the purpose.

ACTUS is currently the only standard covering requirements one to three. It is the only standard that represents financial contracts algorithmically and includes a fully developed data dictionary that includes all of the contract terms or parameters that are needed to run the algorithms. For this reason, it is the prime candidate for point four.

Fintech and Banking

Fintech promises to transform the current inefficient banking landscape and to improve its efficiency by an order of magnitude. This would happen via a significant reduction in operating expenses. As we have argued above, there are two main areas of inefficiency. The first (although less important) area is transaction processing. Transaction processing includes the payment function, which can be executed via either a distributed ledger or traditional channels. Using the same code to generate the cash flow commitments will solve the reconciliation problem between counterparties that is currently caused by the use of different transaction processing systems. Using the same code would significantly reduce trade breaks and the associated need for manual reconciliation with counterparties.

However, the main cost driver in the financial sector is on the analytic level where the major part of the cost saving will take place. This opportunity for efficiency improvements is not yet on the general fintech radar.

Adopting a smart financial contract standard that can be used for both transaction processing and analytics will be required to realize these efficiency gains. It will also end the disconnect between transaction processing and the analytic level that was described above. Adoption and deployment of such a standard will enable straight through processing (STP) in its full sense. The contract we transact is the contract we analyze. Such a smart financial contract would not only be self-executing but also support any desired analysis. It would be a stretch to call it self-executing and self-analyzing. However, it does provide the algorithms necessary to undertake all the important analyses of a transaction.

The following sketch demonstrates how this would work in practice:

- Banker and client discuss a deal
- In order to better understand the potential transaction, they access the appropriate contract type from the open source program library using an API

- They enter the terms or parameters of the deal (notional amount, interest rates, maturity date, etc.) into the API
- The API returns the expected cash-flows
- If conditional clauses apply, this would be run under different scenarios and the expected cash-flows under those scenarios would be produced
- They simulate the transaction using different parameters and scenarios until an agreement is reached
- If both parties agree, they enter the deal in a safe environment, such as a distributed ledger

The same software algorithm used to analyze the transaction "pre-trade" would also handle the transaction processing through the life of the contract. The same code is deployed to produce the contract events and the actual cash flow obligations/payments. This stage of transaction processing is similar to what is done today, except for the fact that all counterparties would be using the same smart contract standard, i.e. the same software algorithm, to compute the exact same cash flow obligations. Actual payments could be recorded on legacy ledgers or on a distributed ledger.

In addition to the contractual obligations being produced automatically, real-time analysis would be available. The following numbers would be continuously available:

- Balance sheets according to multiple principles
  - Nominal value, fair value, IFRS, local GAP
  - Values derived under multiple valuation engines (especially for options)
- P&L statements for any interval, consistent with the balance sheet statements
- Expected cash flow (liquidity)
- Any derivation and/or combination of these reports

For more complex analytics, such as stress tests, a risk manager would have to define the stress scenarios, which could then be immediately applied to the same transactions. The same data can be used in planning and budgeting, in the ALM function, and preparing regulatory reports, such as capital adequacy stress tests.

It should be expected that a plethora of analytic tools would appear on the market. Since the underlying cash-flow generating engine is known, such tools will be easy to supply. The focus of risk management (and other analytic functions) would shift from reconciliation and data cleaning/correction to planning, forecasting and making risk judgments.

Beyond this, the system must be able to handle irregularities like early payments, delayed payment and default. Any change will be reflected in real-time in the (multiple) balance sheets, P&L Statements, and expected cash-flows.

Fintech and Regulation

A standard for cash-flow generation applied by the financial community will have dramatic, positive effects on regulation. The regulatory burden will be reduced while its quality will increase.[ix]

Much of today's regulation is accounting based and balance sheet driven which, as some have put it, is like driving a car by looking into the rear window mirror. This type of analysis will still be possible.

However, the adoption of a cash-flow generating algorithmic contract standard will enable efficient forward-looking analysis. Regulators will become risk managers on a national or systemic scale.

This can be demonstrated by the way the value of stress tests can be significantly enhanced

- A stress test applied of a portfolio of financial contracts following a well-defined and known standard would change it from a black-box exercise to a well understood and reliable exercise
- It would even be possible for regulators to collect only the granular data in the smart contact standard and perform their own stress tests. This would work for all "static"[x] tests like market shocks, credit downgrades, etc.
- Where future business plans enter the picture (dynamic analysis like in CCAR), the analysis would have to be done by the banks, but the planning basis would be known and transparent to the regulator
- The cost of regulation would drop dramatically while its quality would increase
- While it currently takes months to produce stress test results, the time required for analysis could shrink to days if not hours
- Currently, because each bank conducts its stress test under different models and assumptions, no inter-bank comparisons of results can be made. This situation would change.
- The smart contract standard would also make it possible to directly measure the liquidity flows between major players in the financial markets under different stress scenarios. With this new capability regulators could directly see the extent of the interdependence in the market and directly assess the risk of cascading failures.

The greatest benefit for regulation would be an improved understanding of systemic risk. Due to the disparity of the underlying data, such type of analysis is not possible today except by approximation. Current analysis is invariantly based on nominal value, the only available value metric. Nominal value is backward looking and has in most cases – especially in risk management –little or no relevance. Take for example the case of swaps for which nominal values can be in the hundreds of millions while the market value can be zero, close to zero, or even negative.

Having a forward-looking system that is consistent from the single contract to the systemic level is the real revolution.

---

[i] The experience of Ethereum with the DAO should be a cautionary tale. This course of events proved that if enough people agree it is possible to change the system even when using an "immutable" distributed ledger.

[ii] The origin of the term "smart contract" goes back to the mid-1990's and is credited by multiple sources to Nick Szabo, a computer scientist and cryptographer. He defines a smart contract as "A computerized transaction protocol that executes the terms of a contract."

[iii] A more complete discussion of this argument can be found in Brammertz, W. and Mendelowitz, A "From Digital Currencies to Digital Finance: The Case for a Smart Financial Contract Standard", forthcoming in The Journal of Risk Finance.

[iv] ibid

[v] Cash as a pure unit of account or value is the only economic good describable by a single number. Goods are multi-dimensional and cannot be fully described by a single number.

[vi] The standard is called ACTUS (Algorithmic Contract Types Unified Standard). Please visit: www.actusfrf.org/
[vii] True enforcement of a loan, for example, would demand a cash deposit from the borrower to the lender. However, if the borrower has to have the cash he or she is borrowing, there is no need of the loan.
[viii] Frances Coppola, "A Painful Lesson for the Ethereum Community", Forbes.com, July 21, 2016.
[ix] See Brammertz, W. and Mendelowitz, A "The Great Data Challenge", Risk Professional, August 2010, pp. 52-56.
[x] We use the term "static test" to refer to the impact of a change in the state of the world on the balance sheet of a firm absent any action to change the composition of that balance sheet.