

## Smart Contracts Were Around Long Before Cryptocurrency

By Allan I. Mendelowitz and Willi Brammertz

November 17, 2016

The recent emergence of smart contracts is assumed by most to have corresponded with the emergence of cryptocurrencies — the value exchanged in many self-executing deals. But in reality, smart contracts predate and can be independent of virtual currency. This fact has implications for the future of both.

Let's start with the basics. All businesses rely on contracts to conduct their business. A typical contract is written in a natural language and is an agreement between counterparties to provide goods and/or services in exchange for a payment. Like any other industry, finance relies on contracts to conduct business. However, financial contracts are unique. They are agreements between counterparties to exchange cash flows without the exchange of any other goods or services.

The obligations on both sides of a financial contract can be precisely represented mathematically by smart contracts. This unique aspect of financial contracts presents significant, but as yet unrealized, opportunities for more efficient bank operations, better risk management, and less burdensome and more effective regulatory oversight.

A smart contract is self-executing software that is able to autonomously and precisely determine each payment required by the contract. In other words, a smart financial contract represents the black-letter legal obligations contained in a natural language contract. For example, a smart financial contract implementation "knows" when to reset the rate of interest for a variable-rate self-amortizing mortgage. The smart financial contract also knows where to find the reference rate for the reset, how much to add to the reference rate, how to compute the correct principal payment according to the amortization obligation, etc.

The financial world has tens of thousands of unique financial products. At first blush the task of turning such a large universe of financial products into smart contracts seems like a daunting task. However, on closer examination, it turns out that there are only a limited number of cash flow exchange patterns generated by financial obligations. In fact, almost all financial obligations can be represented by less than three dozen cash flow patterns and, by extension, less than three dozen smart contracts.

For example, annuities and self-amortizing mortgages are perceived as very different products and they are sold by different parts of a bank. However, the cash flow computations of these two products can be represented by the same smart contract. Each new payment of either type of financial contract consists of a bit more principal and a bit less interest than the previous payment. And, both have a maturity risk that needs to be modeled: prepayment risk in the case of a mortgage and mortality risk in the case of an annuity.

Web Seminar

**OUTSMARTING FRAUDSTERS**  
with cognitive fraud detection

Sponsor Content From **IBM** Hosted By **AMERICAN BANKER.**

**DATE: 12/20/16**  
**TIME: 12 PM ET / 9 AM PT**  
**DURATION: 1HR**

**REGISTER NOW**

Many assume that a "smart contract" is something totally new. However, banking has used imperfect implementations of smart contracts for at least three decades. When the volume of business became so large that it overwhelmed the capabilities of manual processes, banks turned to smart contracts. The transaction processing systems (TPS) of banks use automated systems based on software algorithms, which resemble smart contracts, to compute the bank's daily payment obligations and expected receipts. These systems grew up incrementally and without any standardization. A bank may have multiple TPS that are programmed in different languages and use different data dictionaries to handle different products.

TPS do not conform to a standard; however, they generate payment obligations, which while not perfect, are deemed to be good enough. The TPS of different counterparties generate numbers that are close, but usually not identical. The differences are not large enough to go to court to resolve, but they do impose costs required for reconciliation.

In the 1990s, modern financial analysis grew out of the response to the financial crisis of the 1980s. All such analyses start with the same type of cash flow computations generated by the TPS, albeit with one key difference. A TPS computes cash flow obligations one day at a time using a known state of the world (i.e. interest rates, foreign exchange rates, etc.). Out of necessity, forward-looking analysis must model the unknown future states of the world in order to generate projected future cash flows and perform forward-looking analysis.

When trying to build analysis systems banks faced a serious problem in dealing with the chaos created by the lack of standardization in their data. In an attempt to resolve this problem banks created central data stores or data warehouses (DW). Significant resources were devoted to these initiatives; however, they did not solve the problem. The only data that was moved to the DW were the terms of financial contracts: principal, interest rate, maturity date, etc. The algorithms used by the TPS to generate the cash flow obligations were left behind and were never standardized. Consequently, while the analysis systems were able to retrieve the terms of individual contracts from the DW, they had to recreate the wheel by building their own algorithmic representations of the contractual obligations. The result has been significant recurring reconciliation costs and less than stellar quality analysis.

The recent interest in smart contracts provides a way forward if smart contracts meet the following conditions. Smart contracts need to be able to represent virtually all on- and off-balance-sheet obligations throughout a bank's operations for both transaction processing and analysis in order to make a material contribution to better-managed and lower-cost bank operations. Furthermore, smart contracts must be standardized, fully tested and validated. In a world of smart contracts, "code is law" and errors or unexpected outcomes are unacceptable. The recent problems experienced by Ethereum with its DAO contract demonstrate the necessity of having complete assurance that the code implements clearly understood contractual obligations with precision. Smart contracts also need to be open source and freely available if they are to become a widely used standard.

A smart contract standard can be used: 1) For analysis in support of risk management, finance, accounting and asset/liability management; 2) transaction processing; and 3) as the computational infrastructure for lower-cost and more insightful regulatory oversight. Furthermore, if blockchain is to play a role in finance beyond simply supporting cryptocurrency transactions, then smart contracts are essential in order to calculate the multiple payments of a typical long-lived financial obligation.

*Allan I. Mendelowitz, a former chairman of the Federal Housing Finance Board, is president of the ACTUS Financial Research Foundation, which is leading an initiative to build open-source smart contracts for finance. Willi Brammertz is the chairman of the board for the ACTUS Users Association.*