JOURNAL

THE CAPCO INSTITUTE JOURNAL OF FINANCIAL TRANSFORMATION 04.2017 N°45

FinTech/RegTech

Algorithmic Regulation:
Automating Financial Compliance
Monitoring and Regulation Using
AI and Blockchain

Philip Treleaven, Bogdan Batrinca

RECIPIENT OF THE APEX AWARD FOR PUBLICATION EXCELLENCE

TRANSFORMATION

CAPCO

EMPOWERING THE FINANCIAL WORLD

Pushing the pace of financial technology, we help our clients solve technology challenges for their business – whether it's capital markets in Mumbai or community banking in Macon.

We leverage knowledge and insights from our clients around the world:

20,000 clients in towns everywhere are becoming more efficient, modern and scalable.

27 billion transactions processed help solve clients' challenges – big and small.

$9 trillion moved across the globe in a single year empowers our clients' communities to build storefronts, homes and careers.

55,000 hearts and minds have joined forces to bring you greater capabilities in even the smallest places.

Empowering the Financial World

FISGLOBAL.COM

FIS

JOURNAL

THE CAPCO INSTITUTE JOURNAL OF FINANCIAL TRANSFORMATION

Recipient of the Apex Award for Publication Excellence

Transformation

FinTech/RegTech

Investments

Banking

Algorithmic Regulation: Automating Financial Compliance Monitoring and Regulation Using AI and Blockchain

Philip Treleaven – Professor of Computer Science and Head of Financial Computing Centre, University College London
Bogdan Batrinca – Honorary Research Associate in Financial Computing, University College London

Abstract
Efficient financial regulation is crucial to the future success of the financial services industry and especially the rapidly evolving new financial technology (FinTech) area. The concept of "algorithmic regulation," modelled on "algorithmic trading systems" [Treleaven et al. (2013)], is to stream compliance, social networks data, and other kinds of information from different sources to a platform where compliance reports are encoded using distributed ledger technology and regulations are "codifiable" and "executable" as computer programs, using the same technology being developed for blockchain smart contracts. In this paper, five areas are discussed: a) an "intelligent regulatory advisor" as a front-end to the regulatory handbook; b) "automated monitoring" of online and social media to detect consumer and market abuse; c) "automated reporting" using online compliance communication and big data analytics; d) "regulatory policy modeling" using smart contract technology to codify regulations and assess impact before deployment; and e) "automated regulation" employing blockchain technology to automate monitoring and compliance. We refer to algorithmic regulation for systems that facilitate compliance and regulation decision-making in financial services using advanced mathematical tools and blockchain technology.[1]

1 Tim O'Reilly originated the term more generally for "government by computer algorithms." Algorithmic Regulation, Wikipedia, http://bit.ly/2i1pT2O

INTRODUCTION

There is a growing concern about regulation and compliance, which is increasingly perceived to have negative effects on the development of financial services, discouraging innovation by requiring an ever-growing amount of data reporting. Overcoming this impasse requires radical automation, especially for regulation of new FinTech entrants [Brummer and Gorfine (2014), PayPal (2013)].

This paper explores five regulatory technology (RegTech) areas ripe for automation in regulation using blockchain technology (see Figure 1):

- **Intelligent regulatory advisor:** an artificial intelligent front-end to the regulatory handbook to simplify registration.
- **Automated monitoring:** monitoring of online and social media, and using natural language processing and sentiment analysis to monitor consumer opinions, concerns, and level of trust and identify market abuses.
- **Automated reporting:** using the FinTech paradigms of online communication, big data analytics, and distributed ledger technology to automate compliance and regulation reporting [known as RegTech in the U.K.: U.K. Government Office for Science (2015)].
- **Regulatory policy:** using smart contract technology to codify regulations; and using computational modeling, such as agent-based systems, for assessing regulatory proposals' potential market impact before deployment (e.g., Basel IV, MiFID II, Solvency III).
- **Automated regulation:** the most interesting, using blockchain distributed ledger technology to record compliance reports and use smart contract technology [U.K. Government Office for Science (2016), Norton Rose Fulbright (2016)] to codify, computerize, and automate financial regulation and compliance (cf. algorithmic trading).

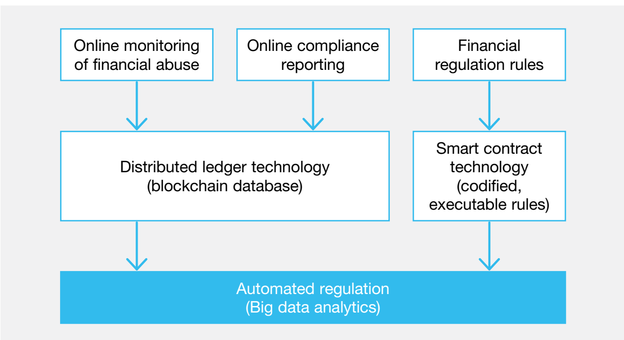

Figure 1 – Algorithmic regulation using blockchain technology

AUTOMATING REGULATION AND COMPLIANCE

Financial regulation is becoming increasingly burdensome. Research from the American Action Forum has suggested that as of July 2016 U.S. banks had paid U.S.$24 bln and allocated 61 million employee hours to comply with Dodd-Frank Wall Street Reform and Consumer Protection Act, passed in the U.S. amid outcry over the financial crisis [Batkins and Goldbeck (2016)].

That said, financial regulation faces a myriad of pressures: political pressure to curb excesses (e.g., Libor); escalating international and European Union regulations (e.g., MiFID II); individual firms simultaneously regulated in multiple jurisdictions and with frameworks; and institutions asked to produce increasing amounts of financial, risk, and compliance data. All this pressure has generated the negative perception that data is being requested "speculatively" and not being used by the regulators. The challenge is to simplify and balance regulation while encouraging innovation for new FinTech alternative finance entrants, in rapidly changing environments [U.K. Government Office for Science (2015)].

In recent years, a number of technologies that can help handle this increased demand for detailed reporting have been developed and have reached commercial maturity:

- **Data scraping:** the technique in which a computer program extracts data from human-readable output coming from the Internet or another program. This involves scraping social networking sites such as Twitter, Facebook, etc., but also web pages, forums, blogs, RSS feeds, online newspapers, and product/service reviews or feedback.
- **Natural language processing:** content interpretation of natural language by means of algorithms mainly based on machine learning.
- **Sentiment analysis (or opinion mining)[2]:** the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral [Medhat et al. (2014)].
- **Automated fraud detection:** identifying suspicious patterns in credit card transactions, identity theft, insurance claims, money laundering, insider dealing, etc.[3]

2 Sentiment analysis techniques, Wikipedia, http://bit.ly/1e2zqkS
3 Data analysis techniques for fraud detection, Wikipedia, http://bit.ly/1OQnzX1

- **Big data analytics:** the process of examining large data sets containing a variety of data types to uncover hidden patterns, unknown correlations, market trends, customer preferences, and other useful business information.[4]

Potential solutions for automating regulation and compliance include an intelligent regulatory advisor, automated systems for monitoring and reporting, regulatory policy modeling, and ultimately an automated regulation system.

Intelligent regulatory advisor

A major challenge for new financial companies is navigating a regulator's handbook and completing the registration process. A solution is to provide an artificial intelligence front-end that supports the location of relevant information and guides that user through registration.

Automated monitoring

The monitoring challenges faced by regulators are illustrated by the U.K. Financial Conduct Authority (FCA). Previously, the FCA monitored 25,000 large and medium size firms. With essentially the same resources, the FCA now has to supervise an additional 21,000 small firms. The obvious solution is to monitor social media for financial abuses.

Developed for brand management and customer profiling, there are a number of sophisticated data scraping and sentimental analysis tools that can equally be deployed by regulators for automated monitoring. Examples include Adobe Social, Brandwatch, Google Alerts, and Mention [Batrinca and Treleaven (2015)].

Automated reporting

One of the recommendations of the U.K. Chief Scientist's review of the emerging new financial technology (FinTech) sector was the so-called RegTech [U.K. Government Office for Science (2015)], in order to use FinTech-style online analytics software techniques (cf. peer-to-peer) to improve compliance and regulation of FinTech companies. Regulation of major financial institutions is largely immutable, set by international, U.S., and E.U. authorities. In contrast, regulation of rapidly evolving FinTech companies arguably provides an opportunity to pioneer lightweight automated reporting.

The three key requirements for automating compliance are [Brummer and Gorfine (2014), PayPal (2013)]: a) **Reporting language** – employing a standard (XML) compliance reporting language, the emerging standard is ISO 20022[5]; b) **Reporting platform** – employing a standard, lightweight, client-side reporting platform that interfaces to industry standard accounting systems, especially for small firms[6]; and c) **Regulatory analytics** – for transparency, employing standard compliance software applications, such as anti-money laundering (AML) or know your customer (KYC), used by both the reporting firms and the regulators.

Regulatory policy modeling

Another emerging area is the use of (agent-based) computational models to evaluate laws and regulations prior to deployment. For example, a number of the regulatory proposals considered after the 2010 Flash Crash (e.g., lodging algorithms with regulators, best price quotes, trading pauses, tick sizes, etc.), if implemented, may have actually increased systemic risk [U.K. Government Office for Science (2012)].

Automated regulation

Here, the concept – inspired by algorithmic trading systems – is a comprehensive automated system for compliance and regulation, where analytics is driven by regulations encoded as computer programs, leveraging blockchain smart contract technology. Below, as background, we explain in simple terms blockchain and distributed ledger technology in the context of cryptocurrencies, and then how the technology is being developed for smart contracts. Having laid this groundwork, we then discuss the possible design of an algorithmic regulation platform.

BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGY

Blockchain[7] [Lewis (2015)], originally conceived for Bitcoin and other digital currencies (or cryptocurrencies), is now recognized to have far-reaching potential in other areas, such as computer-executable contracts. People use the term "blockchain technology" to mean different things, and it can be confusing. Sometimes they are talking about the bitcoin blockchain, sometimes it is other virtual currencies or digital tokens, sometimes it is smart contracts, but mainly it is about distributed ledgers.

A distributed ledger is where all transactions are kept in a shared, replicated, synchronized, distributed bookkeeping record, which is secured by cryptographic sealing and made

4 Big Data Analysis, Wikipedia, http://bit.ly/1FSZacQ
5 ISO20022 Regulatory Reporting XML, https://www.iso20022.org/
6 OpenMRS and other medical records systems, http://bit.ly/1p9iQWN
7 Smart contracts, Wikipedia, http://bit.ly/2im8wXZ

hard to alter by a computationally costly proof-of-work. Every participant (node) has a ledger replica. Nodes synchronize the ledger periodically by approving blocks of transactions. The validity of a block is established by the next block attaching to it, forming, therefore, a chain. The blockchain is the chronological list of all blocks of transactions from the genesis block.

Bitcoin blockchain is a public ledger of all "coin" transactions. The proof-of-work is a crucial element. It is constantly growing as "completed" blocks are added. The blocks are added to the blockchain in a linear, chronological order (every 10 minutes). Each node (i.e., computer connected to the digital currency network) uses a client that performs the task of validating and relaying transactions. The blockchain has complete information about the addresses and their balances right from the genesis block to the most recently completed block. Consequently, in simple terms, a "block" is an encrypted, linked record, and a "blockchain" is a continuously growing list of data records held in a replicated, distributed database or ledger.

For regulatory reporting, the blockchain would be fundamentally a record of the transaction history, delivering a fully transparent, accessible transactional database for regulatory bodies.

Smart contracts

As discussed, a smart contract is a codified legal contract, executable as a computer program, which can initiate actions (e.g., payments). Smart contracts can interact with any software system including other contracts, and potentially highlight when they are no longer valid (e.g., due to changes in the law)[8] [Oasis (2007)].

The potential benefits of smart contracts codified and executable by a computer include formal verification [Walker (1990)], lowering the cost of contracting for low-value transactions, automation, enforcement, and compliance.

Hence, smart contracts can define strict rules and consequences in the same way that a traditional legal document would, stating the obligations, benefits, and penalties that may be due to either party in various different circumstances. But, unlike a traditional contract it can also take information as an input, process that information through the rules set out in the contract, and take any actions required of it as a result. A contract could potentially recognize when it is no longer valid or legal.

Automated trading contract example

As a further illustration of smart contracts' potential applications, consider a manufacturer in China shipping a product to a retailer in Europe. The manufacturer has a contract with the shipping agent, the agent with a shipper, the shipping agent with the receiving agent in Europe, in turn a contract with a haulier, a contract with the distributor, and lastly the distributor with a retailer. At each stage in the supply chain, the appropriate contract executes, the next stage is informed, responsibility is transferred, and the previous stage is paid.

Perhaps more interestingly, as currencies fluctuate, trade tariffs are applied, and laws change, the various contracts could automatically apply the new rates, alert their owners, or potentially reconfigure.

In preparing smart contracts for the above example, it is important to understand that supply chains are complex by their nature, with various parties involved, from manufacturers, shippers, distributors, and retailers, all the way to the consumer. This is especially true when the supply chain partners are in different countries and each partner is responsible for their own working capital and inventory. Trade is typically financed via a Letter of Credit (LC), which, although guarantees payment, is acknowledged to be costly (2%-4% on an annual basis), error prone, and necessitates intermediaries.

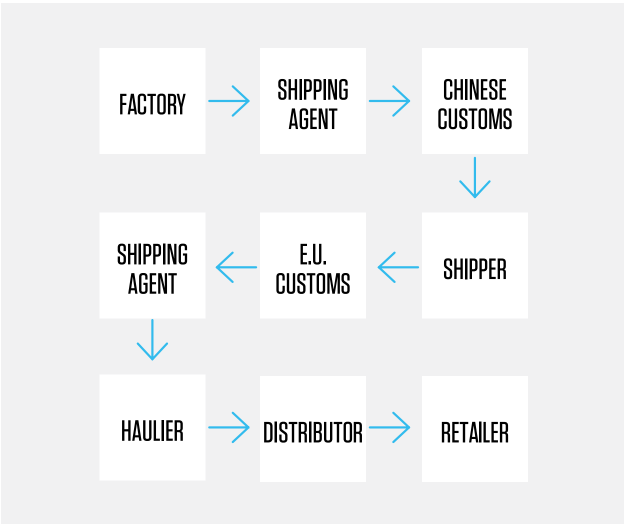

Figure 2 – Smart contracts in a supply chain

8 Computational law, Wikipedia, http://bit.ly/2iSiHHv

In summary, blockchain smart contract technology offers major potential for business efficiency: automation of back-office functions, increased control, reduction in errors, and a major reduction in cost. Currently, most smart contract developments focus on smart "financial" contracts between major financial institutions, notably the R3 Consortium,[9] rather than general legal contracts or statutes.

SMART CONTRACT PROGRAM NOTATION

Smart contract platforms, such as Ethereum,[10] have developed their own proprietary contract notation. The obvious question to ask is whether we can use traditional programming languages such as Haskell, Python, or Java, given their wealth of associated content, in order to code contracts. Notations broadly cover: a) declarative, functional, and logic languages that are mathematically concise (e.g. Haskell, F#, Prolog); and b) imperative languages (e.g. Python, Java):

- **Declarative languages –** a programming paradigm that expresses how to accomplish the problem or the logic of a computation without describing its explicit steps.
  - **Special-purpose languages –** a specification that describes the problem to be solved for a specific domain, such as database programming (e.g., SQL) or smart contracts (e.g., Ethereum).
  - **Functional languages –** a style of programming that models computations as the evaluation of expressions (e.g., Haskell, F#).
  - **Logic languages –** a programming paradigm based on formal logic, where a program is a set of sentences expressing facts and rules about some problem domain (e.g., Prolog).
- **Imperative languages –** a programming paradigm that uses statements that tell the computer what to do and that change a program's state.
  - **Procedural languages –** a programming paradigm that specifies a series of well-structured steps and procedures to complete a computational task or program (e.g., C).
  - **Object-oriented languages –** a programming paradigm that defines not only the type of a data structure, but also the types of operations (functions) that can be applied to the data structure (e.g., Java, C++, Python).

The benefits of declarative languages are that they are more concise, and amenable to mathematical analysis and verification [Walker (1990)], but these languages are less popular for general programming. In contrast, imperative languages are computationally powerful, efficient, and popular, but the semantics of a program can be more complex and difficult to prove due to so-called side-effects.

As an illustration of the possible use of traditional languages for programming smart contracts, we show some pseudo-code in a declarative subset of Python; a "multi-paradigm" language. Although programming in a declarative style of Python[11] may seem an odd constraint to work under, it brings a number of benefits. From a mathematical viewpoint, the benefits include formal provability, modularity, composability, and ease of debugging and testing, whereas pragmatically, the benefits consists of the wealth of associated code, and seamless analytics. Figure 3a illustrates Python pseudo-code for a simple smart contract.

Returning to markup languages, a significant feature is that they are translatable into a human-readable format (cf. HTML to web page), which could be a major smart contract benefit when collaborating with lawyers. Arguably, in addition to choosing a declarative programming notation for smart contracts, we should also ensure it is renderable into plain text. This is illustrated by Figure 3b.

In Figure 3a, code is in blue and black, and values in red.

As discussed, we believe algorithmic trading is an interesting model for the proposed fully automated algorithmic regulation systems.

9 R3 Distributed Ledger Consortium, http://bit.ly/2jdOTBL
10 Ethereum blockchain platform, http://bit.ly/29sDD4H
11 Kuchling, A., "Functional programming (in Python) HOWTO," http://bit.ly/2j0UBaE

a. Simple smart contract code

```
from generateContract import generateContract

contractData = dict()

contractData['firstParty'] = {'name': 'First Company',
'additionalAgreement': 'additional provision text',
'signaturePrivateKey': '12gdf953&sd!815_7vx9bfgn4ngh874ng3$4'}

contractData['secondParty'] = {'name': 'Second Company',
'additionalAgreement': 'additional provision text for second company',
'signaturePrivateKey': '9bd$vs7&5309vdms0)fsd_kdv8vd'}

contractData['date'] = '08/08/2016'

contractData['state'] = 'UK'

contractData['provisions'] = ['First provision text', 'Second provision
text', 'Third provision text']

contract = generateContract(contractData)
```

b. Corresponding Plain Text

First company, known as "First Party," agrees to enter into this contract with second company, known as "Second Party", on 08/08/2016.

This agreement is based on the following provisions:

1. First provision text
2. Second provision text
3. Third provision text

Furthermore, First Party agrees:
additional provision text

and Second Party agrees:
additional provision text for second company

Invalidity or unenforceability of one or more provisions of this agreement shall not affect any other provision of this agreement. This agreement is subject to the laws and regulations of the state of U.K.

Signed:

First Company	Valid signature
Second Company	Valid signature

Figure 3 – Simple smart contract – declarative (Python) pseudo-code and corresponding Plain Text

```
# Example for checking if designated country is on US Treasury OFAC List:
firstParty = {'Country': 'UK', 'Credit': 100000}
secondParty = {'Country': 'North Korea', 'Credit': 250000}
def checkSanctionCountry(countryParty_1, countryParty_2):
    contractState = True
    contractTerminationReason = "Valid countries for a legal money transfer."
    sourceURL = "https://www.treasury.gov/resource-center/sanctions/Programs/Pages/Programs.aspx"
    # Get an up-to-date list of US Treasury sanctions countries from the supplied URL, e.g. currentSanctionList = getUpdatedSan-
ctionList(sourceURL, "US")
    # This returns a list similar to the following line:
    currentSanctionList = ["Iran", "North Korea", "Sudan"]
    if countryParty_1 in currentSanctionList:
        contractState = False
        contractTerminationReason = "The first party's country is part of the US Treasury's list of sanctions."
    elif countryParty_2 in currentSanctionList:
        contractState = False
        contractTerminationReason = "The second party's country is part of the US Treasury's list of sanctions."
    return (contractState, contractTerminationReason)
def sendMoney(firstParty, secondParty, transferValue):
    # Check the countries are not on the Sanction list
    contractState, contractTerminationReason = checkSanctionCountry(firstParty['Country'], secondParty['Country'])
    if contractState:
        if firstParty['Credit'] >= transferValue:
            firstParty['Credit'] -= transferValue;
            secondParty['Credit'] += transferValue;
            print "The transfer was successful."
        else:
            print "The transfer failed because of the following reason: " + "The first party has insufficient funds"
        return (firstParty['Credit'], secondParty['Credit'])
    else:
        print "The transfer failed because of the following reason: " + contractTerminationReason
        return (firstParty['Credit'], secondParty['Credit'])
transferValue = 50000 # USD
firstParty['Credit'], secondParty['Credit'] = sendMoney(firstParty, secondParty, transferValue)
print "The final credit for the first party is: " + `firstParty['Credit']`
print "The final credit for the second party is: " + `secondParty['Credit']`
```

Figure 4 – Smart regulation notation for U.S. Treasury sanctioned countries

ALGORITHMIC REGULATION

We now explore the potential structure of algorithmic regulation systems built upon blockchain smart contract technology. We start by looking at smart "regulation" contracts notation, and then discuss the components of an algorithmic regulation system.

Smart regulation contract

As an example, Figure 4 shows the Python code to check a cross-border payment against an abbreviated list of U.S. Government sanctioned/embargoed countries (e.g., North Korea). The complete list of countries is on the U.S. Department of the Treasury Office of Foreign Assets Control (OFAC) website,[12] with levels of sanctions varying by country. This example is purely illustrative.

In Figure 4, code is in blue and black, comments in green, and values in red. The two principal routines are checkEmbargoCountry and sendMoney.

Algorithmic regulation system

The proposed system (see Figure 5) comprises five main components for: a) "intelligent regulatory advisor" front-end to the regulatory handbook; b) "automated monitoring" of online and social media data to identify individuals', firms' and sector-wide potential abuse; c) "automated reporting" by regulated firms, notably FinTech companies; d) "regulatory policy" specified by international, government, and regulatory bodies; and e) "automated regulation" where regulations are codified, compliance reports are stored in a blockchain, and regulatory analytics is applied to identify abuse, regulatory breaches, and potential risks.

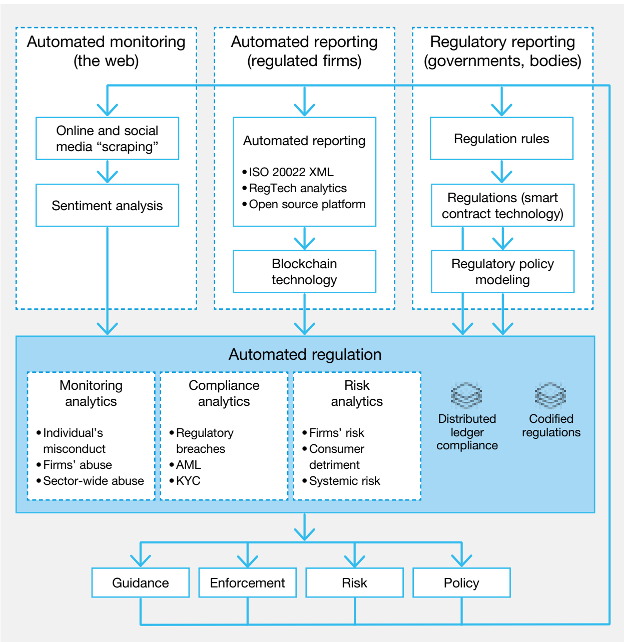

Figure 5 – Algorithmic regulation system

Automated monitoring: this covers scraping the web, social media sites, newspapers, blogs, and chat rooms, seeking to identify complaints about individuals and firms, and sector-wide abuse, such as the incorrect selling of Payment Protection Insurance (PPI) in the U.K. Although there is a number of commercial tools for harvesting web data, such as Adobe Social, Brandwatch, and Synthesio, identification of potential sources of online information remains a big challenge, since disadvantaged victims of small financial firms are unlikely to use Twitter or Facebook to air their grievances.

Automated reporting: as discussed, multiple E.U. and U.S. regulatory bodies are already adopting the ISO 20022 XML standard for reporting. The additional requirement is the need for a "light-weight" (open-source) platform using ISO 20022 XML for compliance reports for small financial companies.

Regulatory policy: for regulatory policy, firstly a declarative smart contract notation is required to encode regulations, and secondly the requirement to use agent-based modeling of proposed regulations for assessing the impact of proposals before deployment.

Automated Regulation: lastly, automation comprises five components: 1) the monitoring analytics component that uses sentiment analysis to identify individuals, firms, and sector-wide problems that may cause concern; 2) the compliance reports encoded using blockchain distributed ledger technology; 3) the compliance analytics component that seeks to identify regulatory breaches, AML, KYC, etc.; 4) the systemic risk component that seeks to identify major firms at risk (e.g., Solvency II); and 5) the regulatory rules component that contains codified regulations using Smart Contract technology.

12 U.S. Department of the Treasury: Office of Foreign Assets Control, http://bit.ly/23za3iL

CONCLUSION

This paper presents the concept of algorithmic regulation modeled on the algorithmic trading paradigm, and employing technology under development for blockchain distributed ledgers and smart contracts. The five major components are: "intelligent regulatory advisor," "automated monitoring" of abuse, light-weight "automated reporting" principally for FinTech companies, "regulatory policy modeling," and "automated regulation." As discussed, algorithmic regulation applied to finance builds on the pioneering work of the R3 consortium of banks in the area of smart "financial" contracts, and any results will be applicable to smart "legal" contracts in general, and the "algorithmic regulation" paradigm applied to government, as proposed by Tim O'Reilly, the founder and CEO of O'Reilly Media Inc. What is clear is that blockchain smart contract technology will have a more major "disruptive" effect on legal services than FinTech is having on financial services.

REFERENCES

- Batkins, S., and D. Goldbeck, 2016, "Six years after Dodd-Frank: higher costs, uncertain benefits," The American Action Forum, July 20
- Batrinca, B., and P. Treleaven, 2015, "Social media analytics: a survey of techniques, tools and platforms," AI & Society 30:1, 89-116
- Brummer, C., and D. Gorfine, 2014, "FinTech: building a 21st century regulator's toolkit," Center for Financial Markets, Milken Institute, http://bit.ly/1CRgQBP
- Lewis, A., 2015, "A gentle introduction to blockchain technology," BraveNewCoin, http://bit.ly/2jdE8iz
- Medhat, W., A. Hassan, and H. Korashy, 2014, "Sentiment analysis algorithms and applications: a survey," Ain Shams Engineering Journal 5:4, 1093–1113
- Norton Rose Fulbright, 2016, "Smart contracts: coding the fine print," http://bit.ly/2iZCdz8
- Oasis, 2007, "eContracts version 1.0: committee specification," http://bit.ly/2iM1SNZ
- PayPal, 2013, "21st century regulation: putting innovation at the heart of payment regulation," http://bit.ly/2hZzNid
- Szabo, N., 2002, "A formal language for analyzing contracts," http://bit.ly/2i2IRpB
- Treleaven, P., M. Galas, V. Lalchand, 2013, "Algorithmic trading review," Communications of the ACM 56:11, 76-85
- U.K. Government Office for Science, 2012, "The future of computer trading in the financial markets," a report of the U.K. Government Chief Scientific Advisor, http://bit.ly/2j0LXZH
- U.K. Government Office for Science, 2015, "FinTech futures: the UK as a world leader in financial technologies," a report of the U.K. Government Chief Scientific Advisor, March 18, http://bit.ly/1FCBDgS
- U.K. Government Office for Science, 2016, "Distributed ledger technology: beyond blockchain," a report of the U.K. Government Chief Scientific Advisor, http://bit.ly/1WreVPL
- Walker, H. M., 1990, "Program verification (tutorial session): techniques and uses," SIGCSE '90 Proceedings symposium on Computer science education

Layout, production and coordination:
Cypres – Daniel Brandt, Kris Van de Vijver and Pieter Vereertbrugghen

BANGALORE
BRATISLAVA
BRUSSELS
CHICAGO
DALLAS
DÜSSELDORF
EDINBURGH
FRANKFURT
GENEVA
HONG KONG
HOUSTON
KUALA LUMPUR
LONDON
NEW YORK
ORLANDO
PARIS
SINGAPORE
TORONTO
VIENNA
WASHINGTON D.C.
ZURICH

CAPCO.COM