

Tezos and SmartPy: Accessible Smart Contracts on an Upgradeable Platform

by Ronald Chichester¹

1. Introduction

The definitions of “smart contracts” differ depending upon the orientation of the definer. For example, the Smart Contract Alliance,² defines a smart contract as “computer code that, upon the occurrence of a specified condition or conditions, is capable of running automatically according to pre-specified functions. Alternatively, the developer community defines “[a] smart contract, like any contract, establishes the terms of an agreement. But unlike a traditional contract, a smart contract’s terms are executed as code running on a blockchain like Ethereum³.”⁴ Similar definitions, but not quite the same.

As with definitions, there are multiple “distributed application” (dApp) blockchains. Besides Ethereum, there is Neo,⁵ Tezos,⁶ Waves,⁷ and several others. “Smart contracts allow developers to build apps that take advantage of blockchain security, reliability, and accessibility while offering sophisticated peer-to-peer functionality — everything from loans and insurance to logistics and gaming.”⁸ However, because the design of the blockchain itself differs, so too does the behavior of the smart contracts written for those disparate platforms.

With that advent of Vermont’s blockchain-based limited liability corporations⁹, and Wyoming’s new corporate form for decentralized autonomous organizations¹⁰, attorneys are encountering clients who want an attorney to opine about the legal effect of the source code for dApps. For those attorneys who have some programming experience, the advent of dApps can be a lucrative addition to a standard business practice. In the recent past, however, programming dApps involved some arcane technologies. However, newer dApp platforms make the development of dApps easier.

1 Ronald Chichester is a solo attorney in Texas whose practice focuses on technology-related legal matters, particularly those involving blockchains, smart contracts, distributed autonomous organizations, and artificial intelligence. Ron is also an accomplished software developer who creates tools for automating the legal practice, performing digital forensics, and enabling robots to classify legal opinions for machine learning.

2 <https://digitalchamber.org/initiatives/smart-contracts-alliance/> (last visited on October 29, 2021). The Smart Contract Alliance is an initiative of the Digital Chamber of Commerce, <https://digitalchamber.org>.

3 Ethereum is the community-run technology powering the cryptocurrency ether (ETH) and thousands of decentralized applications. <https://ethereum.org/en/> (last visited on October 28, 2021).

4 *What is a smart contract?*, Coinbase, <https://www.coinbase.com/learn/crypto-basics/what-is-a-smart-contract> (last visited on October 28, 2021).

5 Neo was founded 2014 and has grown into a first-class smart contract platform. <https://neo.org/> (last visited on October 28, 2021).

6 Tezos is another platform for implementing smart contracts and other dApps. However, Tezos was designed to be upgradeable (without forking) and is distributed under an open source license, both of which distinguish it from other dApp platforms.

7 “Waves is a community-based stack of decentralized open-source technologies to build scalable, user-friendly apps.” <https://waves.tech/> (last visited on October 28, 2021).

8 *Supra*, note 1.

9 *See*, 11 V.S.A. § 4173 et. seq., <https://legislature.vermont.gov/statutes/section/11/025/04173> (last visited on October 28, 2021).

10 *See*, Wyoming Senate Bill SF0038 (2021), <https://www.wyoleg.gov/Legislation/2021/SF0038> (last visited on October 28, 2021).

This paper will take one blockchain and associated development environment as a vehicle to discuss (briefly) some of the mechanisms to remedy disputes involving smart contracts. This problem is even more acute because whole corporations are becoming dependent upon blockchains and smart contracts. This paper will take one blockchain and associated development environment as a vehicle to discuss (briefly) some of the mechanisms to remedy disputes involving smart contracts.

2. An Example

Tezos is an open-source and decentralized blockchain network that can perform peer-to-peer transactions and deploy smart contracts. It has a modular architecture and formal upgrade mechanism that allows its network to facilitate formal verification. For those reasons, Tezos has garnered a considerable amount of interest in the dApp developer space.

An organization called Smart Chain Arena has created a Python¹¹ library called SmartPy¹² that is tailored specifically for developing smart contracts. For anyone even moderately versed in Python, SmartPy is immediately accessible, particularly as compared to the code needed to actually execute on the Tezos platform (see Figure 1).



Figure 1: A comparison of a simple smart contract in Python (left) and Michelson (right)¹³

Development is even easier, because the SmartPy folks have developed an online integrated development environment that you can find at <https://smartpy.io/ide>.

2. The Dilemma

11 Python is one of the worlds most popular programming languages, and is particularly popular with scientists and engineers. <https://www.python.org/> (last visited on October 28, 2021).

12 SmartPy is available at <https://smartpy.io/> (last visited on October 28, 2021).

13 This image is from the SmartPy.io website.

The Tezos blockchain requires Michelson Contracts (low level) language in order to operate. Consequently, contracts written in SmartPy need to go through a compilation (of sorts) to turn the Python code into Michelson code. Michelson code is arcane and cryptic, which would make it difficult for a jury to follow. Python's syntax, however, is much easier for potential jurors (and everyone else) to grasp. To get to Michelson code, the Python code is first interpreted by a virtual machine called SmartML that is written in OCaml.¹⁴ Then the SmartML code is then compiled to Michelson code that can be executed on the Tezos blockchain. Consequently, the Michelson code is a translation of a translation as shown in Figure 2.

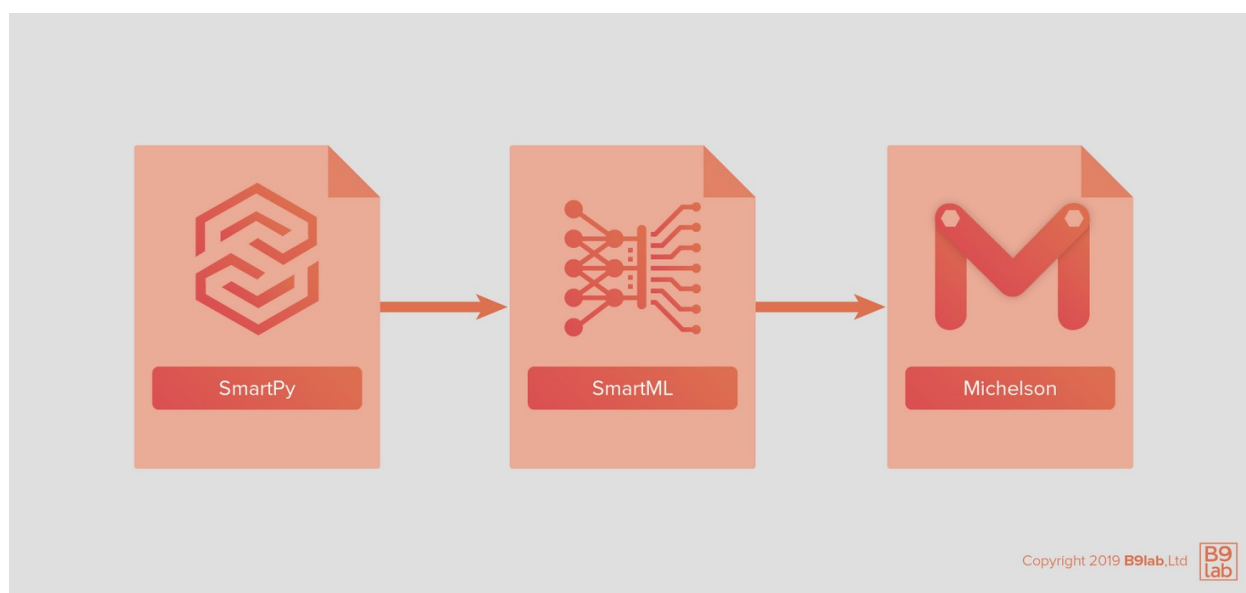


Figure 2. The transition from SmartPy to Michelson code

Whether that double-translation leaves litigators some fodder remains to be seen. Since the Michelson code is what is actually implemented on the blockchain may force the litigator to focus on the Michelson code and ignore the original Python code, which might be a mistake. When considering the Parol Evidence Rule, to which set of code does one turn to discern the mets and bounds of the contract? What the parties intended, or what happened because the parties were ignorant of the (mis-)translation that would ensue? Such questions, however, bely the whole point of a smart contract, namely that the *code* provides the mechanism for arbitrating disputes among the parties.

“A Smart Contract contains no independent means of enforcement. It is simply executed when a predefined condition, determined by a sensor or a so-called "oracle",¹⁵ either occurs or, within a specified period of time or under some other constraint, does not occur. Many aspects of legal contracts, such as those which rely on the exercise of human judgment and insight, are presently incapable, and

14 OCaml is a distinct programming language. <https://ocaml.org/> (last visited on October 28, 2021).

15 Oracles are external servers or processes that retrieve and/or verify external data for blockchains and smart contracts. Since every transaction on the blockchain involves some expense, oracles provide a mechanism for offloading multi-step functionality from the blockchain in order to reduce costs.

may never be capable, of being represented by condition-based functions used in Smart Contracts.”¹⁶

Okay, so the code is the arbiter of any issues between the contracting parties? Is that it? Michaelson would say that it is.¹⁷ However, opinions differ. Andrew Hinkes suggests that there should be limits to deference to code.¹⁸ Hinkes points out that “... neither legal contracts nor code can prevent a party from filing a lawsuit.¹⁹ Does this leave attorneys having to go through the code at one point or another, and also how to explain this code to a jury? Perhaps not. Amy Schmitz and Colin Rule propose an online dispute resolution mechanism for smart contracts.²⁰ Amy also suggests that the *blockchain* itself should include a mechanism for online dispute resolution.²¹ Tezos doesn’t do that, but it is the only blockchain App platform that could be modified to do so. Which begs the question, should some external force (such as a government) mandate that Tezos be modified to include an arbitration mechanism?

The very nature of a public blockchain is that the nodes making up the blockchain do not need (and often do not) lie within a single jurisdiction. Consequently, blockchains such as Tezos tend to be community oriented, and governed in a public manner, which tends to preclude interference by any one jurisdiction. For example, for power consumption reasons (among others), China banned nodes (and thus mining) of Bitcoin within China itself.²² At that time, China had the *majority* of Bitcoin nodes. After China enacted its ban, however, Bitcoin remained in service – without the need for any Chinese nodes.²³ In other words, the distributed nature of the blockchain ensured that problems with any one jurisdiction are obviated simply by locating its nodes outside of that particular jurisdiction.

3. Conclusion

Smart contracts are here to stay. There is simply too many aspects about smart contracts that reduce transaction costs for companies.²⁴ Since the decentralized nature of the blockchain

16 Peter L. Michaelson, Esq. and Sandra A. Jeskie, Esq., *Arbitrating Disputes Involving Blockchains, Smart Contracts and Smart Legal Contracts*, SSRN-id3720876 (2020), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3720876 (last visited on October 29, 2021).

17 *See, ib.*

18 Andrew Hinkes, *The Limits of Code Deference*, *Journal of Corporation Law* Vol. 46, Issue 4 (2021) at 869, https://jcl.law.uiowa.edu/sites/jcl.law.uiowa.edu/files/2021-08/Hinkes_Final_Web_0.pdf (last visited on October 29, 2021).

19 *Id.* at 896.

20 Amy J. Schmitz and Colin Rule, *Online Dispute Resolution for Smart Contracts*, 2019 *Journal of Dispute Resolution* 103 (2019). <https://scholarship.law.missouri.edu/facpubs/726> (last visited on October 29, 2021).

21 Amy J. Schmitz, *Making Smart Contracts “Smarter” with Arbitration*, Alternate Dispute Resolution website, <https://go.adr.org/rs/294-SFS-516/images/Making%20Smart%20Contracts%20Smarter%20with%20Arbitration%20by%20Amy%20Schmitz.pdf> (last visited on October 29, 2021).

22 Alun John and Samuel Shen, Tom Wilson, China’s top regulators ban crypto trading and mining, sending bitcoin tumbling, *Reuters* (September 24, 2021), <https://www.reuters.com/world/china/china-central-bank-vows-crackdown-cryptocurrency-trading-2021-09-24/> (last visited on October 29, 2021).

23 *See, e.g.*, Will Feuer, *US passes China as biggest bitcoin mining hub after Beijing ban*, *New York Post* (October 13, 2021), <https://nypost.com/2021/10/13/us-passes-china-as-biggest-bitcoin-mining-hub-after-beijing-ban/> (last visited on October 29, 2021).

24 *See, e.g.*, Mikko Ketokivi and Joseph T. Mahoney, *Transaction Cost Economics as a Theory of the Firm, Management, and Governance*, <https://doi.org/10.1093/acrefore/9780190224851.013.6> (Published online: 26

platforms means that they are inherently resistant to pressure imposed by traditional authorities, any kind of arbitration or other resolution mechanisms within a blockchain must come from the developer community.