STEPTOE
BLOCKCHAIN BLOG

Steptoe
STEPTOE & JOHNSON LLP

Best Practices for Limiting Liability Arising from Smart Contract Vulnerabilities

By Jared Butcher on April 10, 2017
Posted in Smart Contracts

It is no secret that smart contracts have vulnerabilities. Today's post suggests a mix of best practices to limit potential liabilities that may arise when vulnerabilities interfere with smart contract performance.

But first, some background: One recent survey of 19,366 Ethereum-based contracts found vulnerabilities in 45% of them. Perhaps the most publicized example of a vulnerability was the DAO hack in June of last year, but hacking is certainly not the only way that smart contracts may be compromised. There is potential for manipulation by insiders, which is of particular concern for smart contracts that operate based on "proof of stake" protocols, given the ongoing concerns that those protocols will not be effective in ensuring that the parties play by the rules. Even without intentional interference by hackers or insiders, smart contracts may have software bugs that disrupt performance, and there is the possibility of unintended outcomes if the smart contract's code fails to anticipate an unusual situation. (Consider, for example, a complicated contractual pricing formula that depends on several variables and may cause the price to drop or skyrocket simply because the variables align in unanticipated ways.)

The real-world impact of these vulnerabilities is likely to increase now that more companies are investing in commercial applications of smart contracts. Some are calling 2017 the year of the smart contract. And one recent estimate by industry members predicts a 52% growth in spending through 2019 on applications of blockchain technology in the capital markets alone. This sort of rapid growth is likely to lead to innovation that, among other things, reduces vulnerabilities, but it also is likely to lead to some unfortunate manifestations of vulnerabilities—of the sort that may prompt claims for damages.

The time to prepare for potential claims is before they arise, not after. To that end, here are six best practices to consider when implementing a smart contract:

1. **Prepare a Vulnerabilities Memorandum**: Due diligence should be performed before launching the smart contract, with the goal of identifying potential vulnerabilities. This process should involve legal, compliance, and business personnel working with the smart contract developers to understand exactly what the "smart" part of the contract does (and doesn't). On some level, it may seem counter-intuitive to document pre-existing knowledge of potential issues, but on balance, it may be better to have documentation confirming to an aggrieved party (or a court if necessary) that reasonable steps were taken to prevent foreseeable risks.

2. **Develop a Backup Plan:** Smart contracts are often touted for their ability to overcome certain vulnerabilities in traditional systems, but the reverse may be true when a smart contract succumbs to one of its own vulnerabilities. The plan may be as simple as having a team prepared to fix bugs that manifest in the software, or it may be much more complicated—for example, implementing protocols for responding to cybersecurity threats. The important thing is to think through the threats and prepare to deal with them.

3. **Communicate with Other Parties:** Consider sharing two basic pieces of information with the other participants in the smart contract system—the list of vulnerabilities and the options for dealing with them. The amount of information will vary with the nature of the contract, the

underlying industry dynamics, and the level of sophistication of the other participants. But in many cases, the other participants will be the best allies in the task of identifying and neutralizing problems.

4. **Contractually Provide for Contingencies**: There may be parts of the backup plan that can (and should) be incorporated into the contractual terms (meaning the terms written in lawyer-speak, not in code). Are there alternative procedures to follow if the smart contract malfunctions? Is there a way to correct a transaction that is improperly recorded? What is the process for rectifying unintended outcomes? All of these questions are capable of resolution by well drafted contractual terms.

5. **Contractually Limit Remedies**: Contractual limitations of remedies are common in traditional contracts, and they should work just as effectively in smart contracts. Simply put, contractual remedies should be tailored to encourage the parties to resolve problems caused by vulnerabilities.

6. **Contractually Prohibit Consequential Damages**: Courts typically will enforce contractual terms that prohibit liability for lost profits, lost time, loss of use, and other consequential damages. Be sure that the language of the prohibition clearly encompasses all aspects of the performance of the smart contract.

These six best practices are not exhaustive, but they do provide a framework to consider when launching any smart contract application. The proper planning and attention to contractual terms in advance can go a long way towards mitigating the liability risks that may accompany the launch of new technology.