Romina Zorzini
Legal Architect at Trakti

Luigi Telesca
Trakti co-funder and CEO

Dennis Buonfiglio
Marketing

Giulia Bertotti
Graphics

Trakti

How to create flexible smart contracts

INTRO

What we will be discussing

Smart contracts are a great new instrument that enables an increase in profit and in trust between parties thanks to automation. As with everything smart contracts do come with downfalls, like immutability and flexibility.
This whitepaper aims at addressing the problems that smart contracts are posing in order to spread the adoption of revolutionary technology. In particular, we will be discussing the use of tokens in order to introduce flexibility and suggesting strategies to tackle smart contract's immutability. Plus other cool things, check out the index.

Index

WHAT ARE SMART CONTRACTS?

Definition and characteristics

A clear definition of **smart contract** comes from a study conducted by Barclays and UCL (Smart contract template: foundation, design landscape and research directions) which describe the smart contract as follows:

"A smart contract is an automatable and enforceable agreement. Automatable by computer, although some parts may require human input and control. Enforceable either by legal enforcement of rights and obligations or via tamper-proof execution of computer code."

-UCL

Therefore, a smart contract is an agreement characterized by being automatable at least in part.

In this regard we must distinguish between operational and not operational clauses because only the former is implemented by the code: single or recurring payments, delivery digital or digitized goods, for example.

Automation has the undoubted advantage of making the fulfilment of the contract safe, improving the chances of success in achieving the objectives set by the parties; this leads to an increase in mutual trust and mutual commitment.

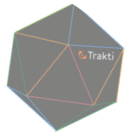
The advantages are huge in terms of efficiency, transparency, cost reduction, speed of reaction, and reduction of contract risks.
Furthermore, always in the execution phase, the smart contract can speed up the transfers of wealth: think of the repayments in insurance contracts, the distribution of dividends, the payment of commissions to agents.
Lastly, the risks of errors are eliminated, not only in calculations but in identifying the creditor and transferring the asset.

As regards the management of contractual contingencies, smart contracts make it possible to ascertain in real-time the events that cause the financial imbalance between the performances or the supervening impossibility.
Thanks to oracles, the occurrence of an unpredictable event is ascertained; consequently, the program measures the economic imbalance between the performances, with the possibility of terminating the contract or bringing it back to fair conditions, on the basis of what was established ex-ante by the parties during the negotiation.

It is possible to modulate the contract in a dynamic way, flexible to objective changes and not merely arbitration by the parties.

Smart contracts based on these features are rapidly developing. In the latest years, they have received relevant attention from lawyers, developers, industries and financial experts which are looking for new and more efficient business models.
The Legal Statement gives the green light to the use of this technology in the UK, confirming the application of established principles of contract law to smart contracts and thus the ability of a "smart contract" to be a valid and enforceable contract under English law.

More recently the Law Commission has launched a call for evidence to help ensure that the technology of smart contracts can thrive in England and Wales.
According to the Law Commission "the growing use of smart contract technology is expected to increase efficiency and certainty in business, and reduce the need for contracting parties to have to trust each other; instead, the trust resides in the code."



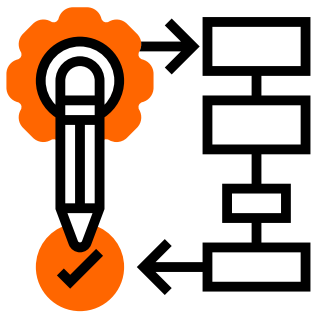
Indeed, there are still some legal and technical issues to be solved before they can be adopted in complex commercial transactions: lack of flexibility and immutability are the major downsides of this technology point out by some works.

These are the problems we will confront by suggesting possible solutions.

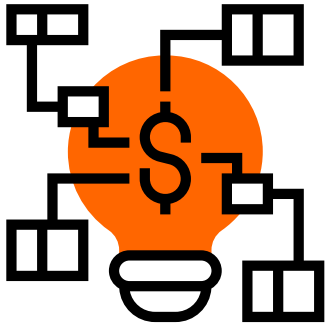
The aim of this work is to delve into these aspects, starting from the problem of inflexibility, which can be addressed for instance by the use of tokens, tolerance and the like, passing through the problem of the immutability of the smart contract where we will see how we can implement some strategies to deal with this issue, ending with the attempt to measure legal concepts to encode abstract concept and offering a solution to the oracle problem.

Have a look at [what Trakti can do for you](#).

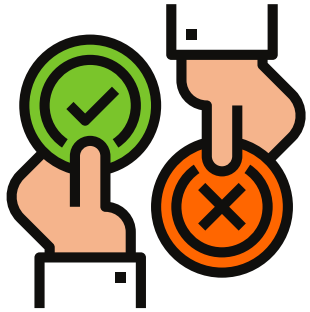
SMART CONTRACTS IN 3 STEPS



A Smart contract is created and the terms are inserted in the code. Now the clauses can self-execute

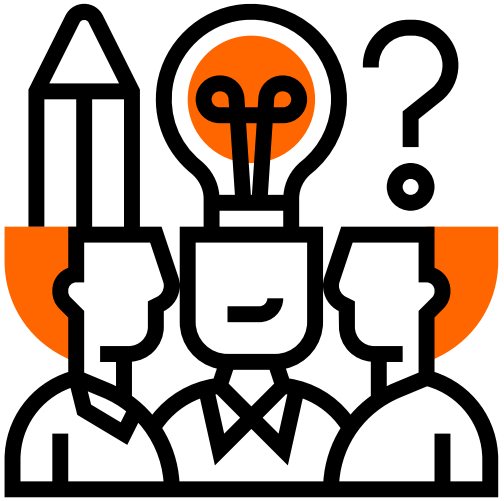


Oracles comunicate with the Smart Contract feeding it information from the world and informing it about the actual performance of the parties



Smart Contracts can then automatically execute the agreed terms with no human interaction, for e.g. compensating the counterparty based on performance

CRITICALITY OF SMART CONTRACTS AND ATTEMPTS AT RESOLUTION



CRITICALITY OF SMART CONTRACTS AND ATTEMPTS AT RESOLUTION

- **Inflexibility**
- **Amendments/Modification**
- **Encoding obligations**
- **The oracle problems**
- **Prospects and best practices**

INFLEXIBILITY

Semantic contracts are flexible. They present two forms of flexibility: linguistic ambiguity and enforcement discretion. Parties are free to shape and adapt their contractual relationship through modifications and re-negotiations.
Smart contracts are changing how legal matters are drafted, moving to an outcome-based approach incentivizes logistics to perform better.

To do so, a comprehensive and clear picture of the business and operational practices for involved parties is necessary when defining and agreeing on terms in order to automate contracts.
In other words, contracts will no longer be able to accommodate vagaries. Operational clauses should be simple, logical, executional and automated. There is no room for ambiguous and vague terms. Redundancies and imprecise terms are not allowed.

In the foreseeable future, if there is a need to use ambiguous clauses in a smart contract, it is likely that interpretational difficulties will be resolved by assigning them to a human-based oracle (**Cardozo Blockchain Project**).
This way of re-planning the contract improves the chances of success in achieving the objectives set by the parties, it becomes functional to the purposes; increases mutual trust and mutual commitment.
The advantages are efficiency, transparency, cost reduction, speed of reaction, and reduction of contract risks.

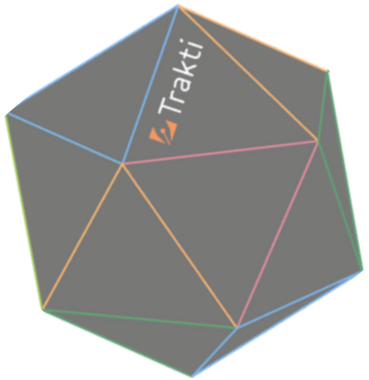
On one hand, parties do not need to monitor each other; on the other hand, flexibility is reduced.

How can we cope with the lack of flexibility?

Here some inputs.
If a valued client is late with its payment the vendor can decide that preserving a long-term relationship is more important than terminating it.

In this case, in order to create a certain level of flexibility, we can insert the provision that in the event of non-payment of an instalment/rate, the contract, instead of ending immediately, provides for the possibility of asking for the amount that has expired and not paid a specified number of times. If despite the payment requests, the debtor does not pay, then the contract ends.
Similarly, in a traditional contractual relationship, a party may be willing to accept a partial performance, because of an interest in maintaining the relationship. Here again, we can implement a system of compensation based on tokens.

A system of Tokens can be used as specific rewards and/or penalties for achieving, or failing to achieve, performance levels that affect the bottom line.



Tokens and penalty clause: most of the time compliant parties do not intend, in case of breach of the contract, to request immediately the penalty to the defaulting party. Parties could provide together with the penalty, several tokens that act as a **warning** and **incentive** to remedy the violation of the clause. Therefore, in case of violation, the party will gradually remove a number of tokens from the defaulting party and will continue to remove them until the violation is remedied. In case of non-compliance and persistent non-fulfilment, the smart contract will be activated with the request for a penalty.

Flexibility can be reached also by setting **tolerance** or different fees/price according to the quality of the services received by the customers.
In one of our smart legal contracts, we set up the following smart clause:

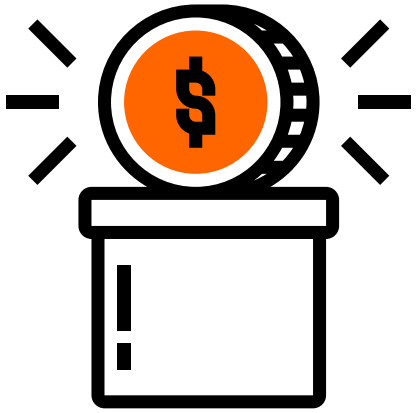
"In case of expected service level, Customer shall pay, as indicated above, to the Company, a subscription fee of (subscription fee amount for expected level) Ether for the Service, which corresponds to Wei. The performance level of the service is measured by the interface address API json call to verify the performance. The expected level is set up on the number/percentage of the expected level value.

If instead, the quality of the service exceeds goals, as above defined, then the Customer shall pay to the Company a subscription fee of (subscription fee for greater service) Ether, which corresponds to Wei; if the quality of service fails to meet the expected service level described in 8.2, then the Customer shall pay to the Company a subscription fee of (subscription fee for under expected level) Ether, which corresponds to Wei".

Do you want to see smart contracts in action?
Set up a free product demo with us

INFLEXIBILITY

How to address it



- **Tokens used as reward or penalties based on performance indicators**
- **Compensation assigned based on objective performance indicators**
- **Tokens used as warnings in case of contract breach or partial performance**

AMENDMENTS/ MODIFICATION

At the present, there is no simple path to amend a smart contract. Given that blockchain is immutable, modifying a smart contract is far more complicated than modifying standard software code.

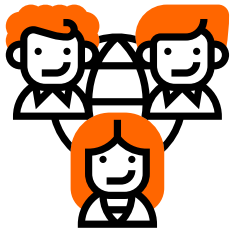
However, **there are a couple of strategies** to tackle this issue. All of them have their benefits and drawbacks. Below is a short description of each strategy

AMENDMENTS/MODIFICATION

How to modify a smart contract

"Kill" the old contract with a new one

or



Modify it through a multiple signature/consensus mechanism

If a smart contract has been fully performed by code, rectification is itself futile. In this case, the claimant can ask for deploying a new smart contract (novation) with the necessary alterations at a new address, **“killing”** or **“self-destructing”** the original one.

In a different scenario, where the code is still running (e.g., recurring payment) the claimant can modify/amend the smart contract using a multi-sig wallet where a list of parties must agree and sign to update it.

However, this type of remedy works only with an upgradeable smart contract. Furthermore, a smart contract can be programmed to “call” and run the code of another smart contract to implement a smart contract structure at the outset that enables the redirection of a master smart contract’s call to an erroneous smart contract to an alternate smart contract.

ENCODING OBLIGATIONS

We must distinguish between operational and not operational clauses.
Operational clauses define actions: parties are required to perform certain actions; could be single or recurring payments, or deliver digital or digitized goods (token, rent a car).
These clauses present the following logic: "**if-this-then-that**"; parties are therefore called upon to define conditions, actions, instruction, consequences within the agreement.

Some types of contractual performance, instead, rely on abstract concepts, i.e., "good faith" or "reasonableness" "the parties will work together on continuous improvement efforts" and seem to be impossible to represent as a closed catalogue of actions or in the form of an objectively measurable result. This language does not specify responsibility for achieving goals nor any details on how any of the tasks would be taken, tracked and reviewed.

Can we measure legal concepts?

“Measure what is measurable and make measurable what is not.”
-Galileo Galilei

In the discipline of the contract, good faith must be understood as mutual loyalty of conduct and a fundamental canon of correctness to which all the parties in a contractual relationship should be inspired.
The obligations of information, cooperation and communication fall within the concept of good faith.

We can elaborate **good faith KPIs** to track predefined organizational behaviour: average time for communication, numbers of follow-ups/meetings, numbers of issues/problems, numbers of problems solved, response time, compliance with the terms of notice, compliance with the obligation to deliver relevant documents, reliable, accurate and up to date data, collaboration rate and the like.

Moral damage has been defined as the **transitory disturbance of the state of mind resulting from an illegal act, or as the inner suffering** (pretium doloris) **suffered by the injured party.**

How can moral damage be compensated?
How is suffering quantified?

In this regard, the Italian jurisprudence (precisely the Milan judges) has tried to give an answer, elaborating parameters to quantify the non-pecuniary damage. The merit of these parameters is that of having established the maximum percentages of personalization of the compensation based on the degree of permanent disability of the subject. The application of these parameters prevents similar cases from being liquidated in different ways.

To measure legal concepts we can proceed as follows:

Break down the legal institution

identify the key elements that characterize it

Measure through graduation scales (i.e., from 0 to 10)*

*Measure them through graduation scales (where each number has a specific motivation, and the algorithm represents a reasoned choice)

We can also analyse how courts describe a certain abstract concept, analyse cases and identify the fundamental and recurring elements that define the concept. These elements will then be measured by translating them into indices, percentages, objective evaluations.

A great example of the latter method is represented by **ReMida** software realized by **G. D`Aietti** (former President of the Court of Sondrio, lecturer in legal informatics at Bocconi-Pavia University).

The software calculates the child and spousal maintenance; it was implemented by analysing over 2500 real cases and is based on predictive and objective mathematical models.

"Reasonable" is a vague term often used in commercial agreements. **J. Cummins** and **C. D. Clack** in their work on "Transforming Commercial Contracts through Computable Contracting", propose due approaches to manage vagueness: the first approach is to define what is meant by 'reasonable' within the contract - internal management.

This would require a set of clear statements on what the parties would consider ‘reasonable’ to mean in the context that it is used and may involve the definition of multiple scenarios.

A second approach is to define how disagreements that may arise will be managed, perhaps through both parties agreeing on some form of dispute resolution process or third-party arbitration.

This approach enables an objective view to be secured on whether something is reasonable, if and when such a view is required.

This approach may also be fully or partially automated.

“Best effort” is another vague term used to indicate all reasonable steps to be taken to achieve an objective.

Maybe we can measure this abstract concept by introducing parameters like defect rate, on-time delivery, the accuracy of the company`s delivery forecast, employee skills, experience, technology capability, product recall rate, number of missed contract milestones and so on.

Parties should include in the contract a benchmark against which a party's performance should be measured: even if individual activities cannot be specified, it is possible to define the term "best efforts" in a way that is flexible, yet quantifiable.

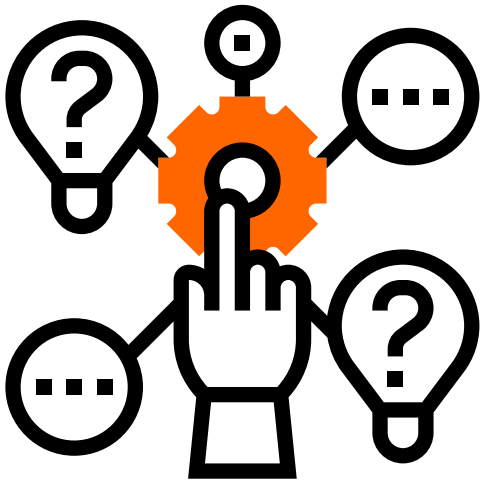
A method to measure and monitor parties' performance (two-way) and automate their reporting is using a balanced scorecard, linking contracts to contract balance scorecard.
A **balanced scorecard** involves both qualitative, as well as quantitative measures; it monitors the adherence to the KPIs agreed, tracks, visualizes, and rates the contract performance.

If you want to know more about balanced scorecards and how to implement them you can **read this article.**

THE ORACLE PROBLEMS

A reliable mechanism that facilitates communication between smart contracts and the outside world is vital for the global adoption of blockchains. Without **oracles**, smart contracts could only rely on information already within their networks, which would severely limit their capabilities.

The crucial challenge in oracle design is the fact that if an oracle is compromised, the smart contract that is based on it is also compromised.



TWO DIFFERENT APPROACHES

We have identified **two different approaches** to tackle the issue:

1) Third-party involved: an authenticated data feed system called **Town Crier (TC).** TC acts as a bridge between smart contracts and existing websites, which are already commonly trusted for non-blockchain applications. It combines a blockchain front end with a trusted hardware back end to scrape HTTPS-enabled websites and serve source-authenticated data to relying on smart contracts. (Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. Town crier: An authenticated data feed for smart contracts.)

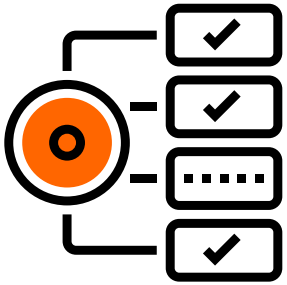
2) Decentralized oracles: a group of independent blockchain oracles that provide data to a blockchain. Every independent node or oracle in the decentralized oracle network independently retrieves data from an off-chain source and brings it on-chain. The data is then aggregated so the system can come to a deterministic value of truth for that data point.

Chainlink and **Astraea** are two great examples of decentralized blockchain oracles.

Smart contracts are giving shape to contracts of the future but still need to evolve before they could be widely adopted in complex business relationships; certainly, as more assets are tokenized or go “on-chain”, smart contracts will become more sophisticated.

THE ORACLE PROBLEMS

Solutions



Involve a trusted third party

Entrust decentralized oracles

PROSPECTS AND BEST PRACTICES

In the meantime, we can deal with the above issues by adopting a few precautions:

- **Parties should be focused on the negotiation stage** more than the execution stage: a comprehensive and clear picture of the business/economic operation is necessary;

- **Parties must move from a risk-based to outcome-based thinking:** incentive logic to perform better;

- **The language used should be plain,** ambiguity and vagueness are not allowed;

- When relying on oracles, **establish what to do in case of erroneous/outdate data;**

- **Testing and simulating in a test environment** the behaviour of the smart contract code to ensure it has no errors and it performs as intended;

- Parties should insert in the agreement a statement that, **if the smart contract results are inconsistent** or diverge from the purpose expressed in the agreement, **then the parties will cooperate** in good faith to carry out the intent embodied in their agreement;

- **Risk allocation** in case of a bug or erroneous code **must be clearly defined** by the parties.

Trakti

Activate a free trial today

CONTACTS

Email
info@trakti.com

LinkedIn

Twitter

Facebook