Auctions and Mechanism Design for Decentralized Marketplaces

Christopher Maree

MRXCHR013

christopher.maree@gmail.com

February 2020

Submitted to the University of Cape Town. In partial fulfilment of the requirements for the degree Master of Philosophy in Financial Technology.

African Institute of Financial Markets and Risk Management, Faculty of Commerce

Supervisor: Assoc. Prof. Co-Pierre Georg

Declaration

I, Christopher Maree, hereby declare that the work on which this dissertation is based is my original work (except where acknowledgments indicate otherwise) and that:

- I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
- I have used the required Harvard convention for citation and referencing. Each contribution to and quotation in this assignment from the work(s) of other people has been attributed, and has been cited and referenced.
- I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.
- I acknowledge that copying someone else's assignment or essay, or part of it, is wrong, and declare that this is my own work.
- Neither the whole work nor any part of this dissertation has been, is being or is to be submitted for another degree in this or any other University. I empower the University of Cape Town to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever.

Christopher Maree

December 2019

Abstract

Those that come up with commercially viable ideas are often not the best suited to implement them. This can lead to allocation inefficacy in the deployment of good ideas. The transfer or licensing of patents is a means of commercializing ideas. However, in the current patent market, the idea seller and the idea buyer often don't match which results in the proliferation of adverse selection.

This thesis examines the existing patent market and finds many examples of opacity. Pitfalls abound for both sellers and buyers which result in inefficiencies when attempting to find the best fit for seller and buyer. Improvements in allocating ideas to the best implementers would help inventors and companies alike.

Brilliant ideas are frequently generated from universities. This thesis presents a means to commercialize these ideas by issuing licenses on the blockchain in an innovative marketplace for ideas. This commercialization of ideas generates funds that support the institution that originally conceived the ideas and indirectly supports foundational research.

The marketplace for ideas is based on sealed bid auctions which ensure that the company that values the idea the most is allocated the license. An optional Harberger Tax system is included to generate constant revenue for the universities from the licensed ideas. This mechanism decreases information asymmetries, increases market liquidity and provides representative license pricing.

Smart contracts deployed on the Ethereum blockchain are used to eliminate auction corruption through trustless sealed bid auctions. Smart contracts also automate license issuance, payments and act as a public ledger of license ownership and provenance. A full front-end web application is presented to interface with the marketplace for all users.

Acknowledgements

I would first like to acknowledge my supervisor, Co-Pierre Georg, for his constant support, availability and insight throughout writing this thesis. I have learnt much from you in the past year and your input has been invaluable in helping me form the ideas in this thesis and my views on the world at large.

I could not have reached this stage in my studies without my beloved parents Gill and Mike. Thank you for your unbridled support throughout the years from preschool all the up to a post-graduate level. Thank you for dedicating your time and input in helping to edit and shape this thesis and for all the support along the way.

To Luke Meiklejohn for the dedicated work and brainstorming sessions on the original UniCoin idea. Your critical sharp mind and way of looking at problems was invaluable in shaping the ideas created in that project, which heavily influenced this thesis.

Thank you to Allan Davids for adding focus and direction to my writing. I appreciate all the time taken to guide me in refining my writing.

Lastly, I want to say thank you to Veronica Coutts for always being a sounding board for ideas, providing valuable input and for your support throughout this year.

Contents

Chapter 1

Introduction

New ideas, and the implementation thereof, lay the foundation for economic growth. Ideas are however not always born to the best users. Commercialization of ideas requires tacit industry knowledge, capital and labour investments (Love et al., 2019). Furthermore, ideas and innovations are sometimes best utilized outside the contexts of their creation (McClure, 2015). This creates an allocative efficiency problem: the ideas of one person are in some cases best implemented by someone else.

This thesis presents the concept of a *marketplace for ideas* as a mechanism to improve allocation efficiency between ideas and implementation. This marketplace for ideas leverages sub-licensing of patents to create a legally permissible way to implement and protect an idea. This facilitates the monetization and trade of ideas to third parties who intend to implement them. A primary use case for this marketplace is commercially viable research published from universities. In this context, the marketplace for ideas helps academics to monetize their work while incentivizing open science and foundational research. Content within the platform remains open and free for any non-commercial usage.

Patents were once seen as arcane legal instruments with little relevance to boardroom strategies but have now emerged as a key asset that defines enterprise value (McClure, 2015). 2018 saw the highest number of patent applications of all time with 3.3 million filed worldwide.[1] Patents form an important part of a company's intangible assets with 84% of the SP 500 market value in 2015 consisting of intellectual capital, represented by patents, trademarks and copyrights (Ocean Tomo, 2010). Within this intellectual capital, the vast majority of value comes from the patents themselves.[2] These intangible assets are growing at an ever increasing rate and so effective mechanisms to trade them is becoming increasingly important.

The monetization of patents is, however, difficult due to their inherent complexity and the specialized nature of each patent. This makes the process of due diligence expensive and complicated for potential buyers (Kelleyf, 2019). The modern day patent market operates without reliable quality and valuation benchmarks which makes it challenging to quantify the usefulness of a patent McClure (2015). Added to this, patents are a probabilistic right in that owning a patent does not necessarily guarantee absolute enforceability as once a patent is granted the right might be subsequently eliminated without recourse or

[1] WIPO (2019) who found that 46.4% of applications came from China.

[2] Love et al. (2017) shows that patents are above 95% of this intangible asset value.

compensation.[3] In the sale of a patent, it may not be apparent who the potential buyers and competing sellers are, especially in situations where companies want to keep their strategies secret. This is because the vast majority of the market for licensing and buying patent rights operates in near total darkness.[4] Parties to a patent transaction mostly opt for confidentiality due to uncertainty about how the market and their competitors will interpret both the sale and purchase. Resultantly, the majority of the market comprises bilateral negotiations wherein the price and terms of the transaction are kept confidential.[5] These difficulties make patent trade challenging, which results in a thin market. This thesis aims to address some of these problems through a more transparent and efficient market mechanism which emphasises allocation efficiency of ideas to companies that value them the most.

A leading use case of this marketplace is the 2019 exclusive licence agreement between Tesla and the University of Dalhousie, who created a new battery design that improved upon the previous market best by more than two fold.[6] The university published the paper openly enabling anyone to build on it non-commercially but issued an exclusive licence agreement with Tesla, protected by a patent, enabling Tesla to use the ideas commercially in their vehicles (Oberhaus, 2019). This open publication goes against the standard practice in the electric vehicle industry where battery designs are normally kept as a closely guarded secret. This licence issuance and paper publication could have been run on the proposed marketplace for ideas which would have enabled the researchers to reach a much wider pool of potential buyers and therefore would have enabled them to generate more revenue from their intellectual property.

This thesis first examines the status quo of patent markets and analyzes the complexity in selling ideas. A full implementation of the proposed marketplace is then conceptualized, implemented and discussed. The marketplace presented aims to disrupt the current opaque and inefficient market. The proposed market acts like an *intellectual property broker* and achieves increased idea allocation efficiency through sealed bid auctions when granting licenses to ideas. This mechanism preserves bidders valuation and ensures confidentiality while enabling the bidder with the highest valuation to win the auction. Note the thesis assumes the existence of a robust global legal framework that can fully defend a patents exclusivity (i.e preventing non-patent holders from copying the ideas without owning the patent).

Additional mechanisms are discussed and implemented including a Harberger Tax model.[7] This provides a way for continuous re-sale and representative pricing of licences after an auction is complete, thus ensuring that the owner of an idea is always the company that values it the most, thereby improving allocation efficiency. The taxation imposed from the

[3] Enforceability depends the patent's owner, the validity, and claim coverage. This can be changed by subjective interpretation in a court of law and so a patent can be found to be invalid (McClure, 2015).

[4] Love et al. (2017) estimated that only 20% to 25% of patent trades happen on brokerage markets with the rest happening privately with no public information available.

[5] Love et al. (2014) showed that brokered patent sales, which have publicly available information, contribute less than one percent of all revenue earned annually worldwide from patent and licence sales.

[6] Oberhaus (2019) reports that the battery drives vehicles over 1 million miles before needing replacement. This is revolutionary as electric cars now have a longer life before replacement than petrol.

[7] Harberger Tax aims to find a balance between pure private ownership and total commons ownership to increase general society welfare and productivity through economic policy (Posner and Weyl, 2017).

Harberger Tax model is paid directly to the idea creator, creating a stream of recurring income for valuable ideas. Additional restrictions to protect idea owners from exploitation and system gaming are considered.

The implementation presented leverages the Ethereum blockchain and a series of custom designed smart contracts to build a trustless, corruption free, peer to peer marketplace. This removes any opportunity for the market operator to influence auctions, extract unfair rent or manipulate licence issuance. Each of these components are discussed in detail along with a design decisions rational.

The rest of this thesis is structured as follows: Chapter 2 examines related literature and the background of patents and patent marketplaces. This lays the foundation for the discussion and provides insight into where the market should be positioned. Chapter 3 examines the economics of pricing and selling patents and the complexities thereof. This chapter builds the argument for how an improved marketplace for ideas should be constructed. Chapter 4 introduces the proposed marketplace and mechanism design from a conceptual level and examines the different use cases and considerations in building a idea marketplace. Chapter 5 presents a blockchain solution for the market proposed in the previous chapter. This starts with an introduction into blockchain and Ethereum and is followed by technical implementation details on how the market proposed can be built using a blockchain solution. Chapter 6 discusses the implementation details and examines the market at all levels of the technology stack. This includes examining the smart contracts, network design and front end. Chapter 7 critically analyzes the proposed marketplace from a legal and regulatory perspective, market mechanism perspective and implementation viability perspective. Chapter 8 concludes.

There is also a detailed Appendix attached at the end of of this thesis. It is referenced when appropriate throughout. This includes detailed contract sequence diagrams in Appendix F, application screenshots in Appendix I and future research in Appendix J.

Chapter 2

Background and related literature

The available literature on patents, patent markets and the history of trading ideas is extremely broad. This chapter examines the mechanisms for transferring patents and the actors within exchanges. Importantly, university involvement within this market is also investigated to better understand how the proposed market will operate within the existing environment.

2.1 The history of patents monetization

Exclusive patent rights have a long history. As early as 500 BC the Greek colony of Sybaris granted exclusive rights for one year to creators of "unique culinary dishes", an early example of an exclusive right (Goldman, 2000). During the Ming Dynasty in China, potters were required to mark their products to ensure quality so that people could properly direct complaints (Bloch, 1960). In 1331 Edward III granted a letter patent to John Kempe, a Flemish weaver, to entice skilled foreign labor to settle in England (Tim, 2012).

The modern notion of a patent is a temporary government granted monopoly right on an idea. This gives the owner the legal right to exclude others from using the intellectual property for a period thus enabling the owner to freely capitalize on their idea (Conway, 1956). The ideas within patents are public as an encouragement for further innovation. Modern patents grant exclusive rights for 20 years after patent issuance (Harhoff et al., 2007). Once the patent expires it becomes open and anyone can freely use the ideas presented within.

The first example of a "patent" in the modern definition was granted to a 1421 Florentine architect, Filippo Brunelleschi, who received a three year patent granting him exclusivity on his invention, a barge with a hoisting device, to transport slabs of marble (Harhoff et al., 2007).

Trading patents is not a new concept either with early sales dating back to the 18th century. Walter Hunt invented and patented the safety pin in 1849 and later sold it to W. R. Grace and Company for $10,000. W. R. Grace and Company went on to mass produce the pin and make millions (Harhoff et al., 2007).

Intermediaries have also been around for a long time, in fact the market for patents has been aided by trade intermediaries for almost as long as patents themselves (Abrams et al., 2019). Due to the complexity of patents and information asymmetry the market for patents suffers from internal and financial frictions that make it hard to match buyers and sellers. Intermediaries can help decrease this friction by helping match buyers and sellers(Lemley, 2011). Inventors with valuable ideas look for channels to sell their idea. According to Lamoreaux et al. (2013) there were 550 intermediaries matching buyers and sellers of patents in the 1880's across the US.

2.2 Importance of patents in modern business

Over the past forty years company value has shifted from tangible to intangible assets (Akcigit et al., 2016). In 1975 17% of the SP 500 market value consisted of intangible assets like patents, trademarks and copyrights. In 2015 this had grown to 84% (Ocean Tomo, 2010). Haigh (2018) reports that global intangible value had reached 57.3 trillion USD at the beginning of 2018's financial year, constituting 52% of all enterprise value of all publicly traded companies world wide. Of these intangible assets, patents constitute a key component.

2.2.1 Impact of patents on research and development

Innovation is expensive and risky. In order to generate new ideas companies must invest in Research and Development (R&D) which is not guaranteed to generate a valuable idea (Hansen and Birkinshaw, 2007). Companies often have to start dozens or even hundreds of research projects in order to achieve one commercial success (for Europe, 2011). For example, pharmaceutical companies will research hundreds of molecules to create one marketable drug (Mestre-Ferrandiz et al., 2012). Added to this, once a new product or service is on the market competitors will invariably try to mimic it. A patent gives temporary exclusive rights to inventors, thereby increasing their chances to recover the upfront investments made to generate innovations and to bring them to market (McClure, 2015).

The protection of an idea and the creation of a monopoly provides incentives for economically efficient R&D (Olson, 2009). The 2018 EU Industrial R&D Investment Scoreboard showed that worldwide, the total investment into R&D of the top 2500 companies accounted for 90% of the world's business-funded R&D, amounting to €736.4 billion (Hernández et al., 2018). Without proper patent laws it is difficult to justify financial expenses into R&D. Legal protection is therefore necessary to give companies a chance to fully exploit the idea after its creation to recoup expenses (Akcigit et al., 2016).

The environment surrounding R&D activities is also changing due to increasing technology complexity (Love et al., 2017). It is increasingly difficult and inefficient for companies to research and develop all the technology components of their products on their own (Yanagisawa and Guellec, 2009). For example, modern cell phones can have thousands of patented components and software (Akcigit et al., 2016). The ability to buy and sell intellectual property in the form of patents has influenced how companies approach their R&D and innovation in general (Akcigit et al., 2016). Company R&D sometimes generates useful ideas and other times it does not. A useful idea increases a company's productivity if the idea is close to its technology class.[1] A company may choose to sell the idea if it is not close to its class. Conversely, a company may choose to buy an idea if it has failed to innovate (Akcigit et al., 2016). On the one hand, the ability for a company to sell ideas provides an incentive to engage in R&D and on the other hand, the ability to buy an idea reduces the reward from doing R&D.

2.2.2 Value of a patent to a company

A high-tech startup's value is often tied to the idea that gave birth to the company (Raffoul and Brion, 2011). The more important and impactful the idea, the higher the potential valuation of the company. Resultantly, better protection around that idea may also increase the viability of the company which adds measurable value (Raffoul and Brion, 2011; Akcigit et al., 2016).

Patents provide companies with the freedom to operate as they know that they are protected from litigation when working on the specific product or idea (Yanagisawa and Guellec, 2009). For example when a company brings an external idea or technology into their business they must carefully examine if the idea can be used without infringing on the legal rights of the idea creator's patent (Yanagisawa and Guellec, 2009). If they already have the patents to an idea they are free to operate in that domain with decreased litigation concerns.

2.3 The market for patents as an asset

Markets enable the exchange of goods and services, including intangible assets like patents. The objective of a market is to establish a price, quantity and a set of rights that results in the transfer of ownership from one party to another (Jarosz et al., 2010). Patents are a complicated, nuanced and specialized asset which makes the exchange and price discovery thereof complex (Akcigit et al., 2016). These complexities will be examined once an overview of the market as a whole has been presented.

2.3.1 Actors in the patent marketplace

Patent sellers might sell their ideas if they are not aligned with their core business. Akcigit et al. (2016) concludes that a patent contributes more to a company's value if it is closer in terms of "technological distance" and as such patents that are far from a companies primary offering are more likely to be sold.[2] Alternatively, inventors, universities or companies who do not have the means to commercialize an idea might want to sell it (Akcigit et al., 2016). This is where the marketplace for ideas presented in this thesis operates: ideas that are better suited to be implemented by other companies.

Patent brokers are the intellectual property equivalent of real estate agents and operate on both the buy and sell side of a patent transaction (Love et al., 2017; Yanagisawa and

[1]Each company operates within a particular technology class, which is fixed over time. This defines the domain within which they operate and will dictate if an idea is useful or not (Lemley, 2011).

[2]Technological distance is the length of technological space between two things. In this case a company and an idea. If the idea is aligned with the companies core offering then the distance is lower.

Guellec, 2009). They help to market patents on the sellers behalf and attempt to find and negotiate deals with buyers in exchange for a fee.[3]

The mechanism presented in this thesis operates within this brokered patent market, aiming to decrease the friction between buyers and sellers and increase allocation efficiency while minimizing the exploitation from intermediaries.

Buyers of patents, like sellers, either interact directly in the market or via an intermediary third party who represents them. McClure (2015) and Lamoreaux et al. (2013) find that companies buy patents for three primary reasons:

1. to gain access to the underlying intellectual property (IP transfer),
2. as a defensive tactic to fend off litigation (defensive acquisition) or
3. with the intention of pursuing legal action against companies that are infringing on the patent (offensive acquisition).

Patent aggregators indirectly represent buyers and facilitate coordination between multiple companies with aligned interests (Lemley and Myhrvold, 2009). For example the Allied Security Trust acts as a defensive patent aggregator to represent its member companies. It charges a membership fee which is used to buy patents of interest to its members (Love et al., 2017). Love et al. (2017) summarized patents purchased by an aggregator tend to be in one of three categories:

1. patents that are being enforced against members,
2. patents that members suspect could be used against them in the future or
3. patents that members want to hold as defence to guard against competitors.

Non-practicing entities (NPEs) are companies whose primary source of revenue is from patent licensing fees or patent litigation awards (Lemley and Myhrvold, 2009). The complex nature of patents, lack of benchmarks, exchange and search friction and the opaque nature of the market creates a perfect environment for the proliferation of uncertainty (McClure, 2015). In this environment adverse selection thrives which has enabled NPEs to emerge. NPEs encompases individual inventors who do not implement their inventions, university IP licensing offices and shell companies that file hundreds of lawsuits to patent aggregators whose revenue comes from licensing fees (Yanagisawa and Guellec, 2009). While some NPEs are legitimate licencing entities, such as universities, many are exploitative and their sole aim is to extract value from infringing companies.

Patent assertion entity (PAEs) are a subset of NPEs and aggregators that hold a portfolio of patents and monetize them through litigation and licencing fees rather than supporting development or transfer of technology (Love, 2013). They can be seen as speculators in the patent market (McClure, 2015). PAEs are known to execute a form of "holdup", by suing infringing companies on the patents they hold (Abrams et al., 2019). PAEs do not compete with the companies that they sue and so they are able to take advantage of several legal opportunities that are generally not available to operating

[3] Love et al. (2019) reports that on average these fees are 20% of the deal closed.

companies. For example, because PAEs do not sell products that compete with those produced by alleged infringers, they are able to avoid countersuits (Love, 2013).

Added to this PAEs sue to recover monetary damages rather than injunctions to protect market share (Love, 2013). Resultingly, PAEs can strategically delay suits until their alleged infringers are locked in to using the allegedly infringing technology through manufacturing a product and so can not easily switch to an alternative that does not infringe on the patent (Abrams et al., 2019). PAEs are commonly referred to as "patent trolls" and are seen as a burden on innovation in most settings (Abrams et al., 2019).

Love et al. (2017) estimates that 50% of patents sold wind up in the hands of PAEs and aggregators, neither of which commercialize the technology. The activities of PAEs has grown dramatically in the last 20 years. McClure (2015) found that in 2012 PAEs brought over 2544 lawsuits in the US, which accounts for 61% of all patent suits. Bessen and Meurer (2014) found that this litigation activity has a direct cost of $29 billion per year on the world economy.

Patent services include self-operated software to help companies identify problematic patents and patent management consultants (McClure, 2015). Consultants offer patent and prior art search, infringement analysis, product teardowns, including reverse-engineering and patent validation (McClure, 2015).[4] Ultimately these companies help to identify a companies IP risk and better understand the uncertainty associated with outstanding patents against their products and services (McClure, 2015). An extensive list of self-operated software and consulting services can be seen in Appendix A.

2.3.2 Mechanisms for transferring patent rights

Today the market for patents is extremely fragmented and opaque. Most dealings happen in total secrecy due to uncertainty about how the market and their competitors will interpret the sale (Love et al., 2017). Patent sales happen via a number of different mechanisms that can be split between the quasi-public brokered market and private bilateral negotiations (Love et al., 2017).

The quasi-public brokered market encompasses all market mechanisms where a third party matches patent buyers and sellers (Abrams et al., 2019). Some of this market is public and open, enabling anyone to browse catalogs of patents, others require sign up and registration fees and some are more exclusive and can be accessed via invitation only (Yanagisawa and Guellec, 2009). This role was originally filled by companies who would seek out buyers on sellers behalf, and is now filled by third party platforms who facilitate matching (Love et al., 2017).

Patent clearing houses fall under the definition of patent brokers (Van Overwalle et al., 2007). These act to bring buyers and sellers together with varying levels of involvement in the exchange. There are three main types of clearing houses:

1. *Information clearing houses* which simply provide a mechanism to exchange technical and relevant patent information about a patent sale to potential buyers. This

[4] Prior art is a previously publicized version of a very similar technology which can impact a claim of originality. Consulting services aid in finding similar patents that could have an overlap with an idea.

is a public, "for sale", billboard for patents (Van Overwalle et al., 2007).

2. *Technology exchange clearing house*, which provides private information exchange to potential buyers that the seller has deemed acceptable. This can involve a due diligence process from the clearing house to vet the bidders (Love et al., 2017).

3. *Royalty-collection clearing house* who provide that defined in (1) and (2) but also act to facilitate the actual patent exchange between buyers and sellers (Yanagisawa and Guellec, 2009).

The market presented in this thesis encompasses much of the functionality described in (3) while expanding the scope to include the issuance of blockchain based licences.

An example of a technology exchange clearing house is the Intellectual Asset Management (IAM) market, which was started by the IAM magazine and now accounts for 25% of all patents sold in the brokerage market (Yanagisawa and Guellec, 2009). Importantly, no pricing information is listed in the IAM market; rather sellers list patents or patent portfolios and buyers reach out to sellers through the platform and then negotiate listing prices offline (Love et al., 2017). This mechanism therefore acts to bring buyers and sellers together in a public form and enables private bilateral negotiations to occur once the matching friction has been overcome.

The private patent market consists of secret bilateral negotiations directly between buyers and sellers without the assistance of a broker (Love et al., 2017). These trades tend to be of high value between large companies and consist of back and forth negotiations between buyers and sellers until terms are reached (Jarosz et al., 2010). Love et al. (2019) estimates that the private patent market is upwards of 10 times larger than the brokered market. These private sale's terms tend to be bespoke and are motivated by a wider set of factors that include more than just the transfer of patents, such as strategic and business relations (Abrams et al., 2019). Actual patent ownership transfer tends to occur only after extensive due diligence is conducted on both sides (Jarosz et al., 2010). The details of these dealings are mostly unknown unless they are brought to public light through a court case or acquisition (McClure, 2015).

Private bilateral negotiations are often used by large companies due to uncertainty about how the market and their competitors will interpret the sale (Love et al., 2017). Strategically choosing a limited number of negotiating partners, bound by non disclosure agreements, enables patent sellers to reveal far less proprietary information regarding potential patent value. Private negotiations also enable more bespoke agreements to be reached that can accommodate buyers and sellers specific requirements and needs (Jarosz et al., 2010).

The lack of information in the private patent market makes it difficult to quantify the price of patents traded, the size of the market and the number of market participants (McClure, 2015). Lemley and Myhrvold (2009) compares this private market to a stock market in which buyers and sellers can't find the price at which anyone else exchanged a stock. If one wanted to buy or sell a stock they would have to guess its price. The result of this is massive inefficiency. Willing buyers and sellers would miss each other, the sale price would vary widely between sales and those with private or insider information would be easily able to exploit others. Overall this market would be nowhere close to what today's stock market is in terms of volume and efficiency.

Due to the opacity in the private patent market this same inefficiency and ensuing exploitation flourishes. Willing buyers and sellers can't find each other and prices are all over the map. This results in companies often ignoring patent rights altogether because they cannot make reasonable forecasts for the cost of obtaining a licence. This lack of pricing information permits patent owners to exploit companies that make products by demanding high royalties from a jury that has no way of quantifying the actual worth of a patent. Lemley and Myhrvold (2009) suggests that the solution to this lack of information is to require the publication of patent assignment and licencing terms.

Other than the lack of efficiency in the private market, bilateral negotiations have several other drawbacks. Firstly, it is difficult for a seller to find an appropriate buyer without incurring the cost of widely advertising the availability of their patents (Jarosz et al., 2010). At the same time buyers don't want to publicize the desire to buy patents as this could reveal competitive or legal weaknesses in their patent portfolios to competitors (McClure, 2015). Negotiations are time consuming and due diligence is expensive. Additionally, there is an opportunity cost associated with devoting a substantial portion of a patent's limited life to negotiation rather than patent exploitation (Jarosz et al., 2010). If negotiations break down, these costs become particularly acute because negotiations must start over again (Kelleyf, 2019).

These reasons ultimately mean that the private market for patents is thin, with little opportunity for robust bidding and counter bidding resulting in poor pricing discovery.

2.4 The current status and size of patent trades

WIPO (2019) recorded 3.3 million patents registered worldwide in 2018. The patent market as a whole (including private transactions and patent licencing) is estimated to be between \$35 and \$100 billion a year (Love et al., 2017). This estimate range is so large due to the opaque nature of the market and, as such, different estimates will look at varying data sets and metrics. Akcigit et al. (2016) shows that on average 16% of all patents registered in the United States Patent and Trademark Office(USPTO) are traded and that new patents are sold within 5.48 years.

As of 2016 the largest buyers of patents are product producing technology companies (Love et al., 2017). These companies also receive the highest number of litigations with the top 5 being Google, Apple, Microsoft, Samsung and Cisco (Love et al., 2017). To understand the scale of this market and the volume of trade between these larger tech players, the Nortel and subsequent Motorola acquisitions are offered as primary examples (Owen, 2012). In 2011 a consortium called Rockstar Bidco formed by Apple, Microsoft, RIM, Sony, Ericsson and EMC bought 4000 patents from the recently bankrupt Nortel Networks (Owen, 2012). This acquisition ran on an auction where Google and Rockstar engaged in a sequential bidding war until Rockstar won out with a \$4.5 billion bid (Akcigit et al., 2016). After Nortel's bankruptcy the companies assets were sold for approximately \$3 billion, yet the auction finished at \$4.5 billion demonstrating how a company's patents can be worth more than all of its other assets (Love et al., 2017).

Soon after the acquisition Rockstar Bidco engaged in multiple litigations against Google for a number of patents that they were actively infringing (Mullin, 2013). To build a defensive portfolio to protect against further litigation, Google proceeded to buy Motorola to acquire 17,000 patents at $12.5 billion in cash (Johnston, 2011). They held the company for only three years and sold them in 2014 for $2.91 billion to Lenovo (Johnston, 2011).

These two acquisitions show the scale of the patent market in single, private, bilateral trades.

2.5 Monetization of IP from universities

As the primary use case for the market presented in this thesis is for commercially viable research published from universities, it is apt to look at how universities view IP monetization. Transferring research created within universities to companies is important in promoting innovation (Yanagisawa and Guellec, 2009). This is a mutually beneficial situation wherein universities can raise funding from licencing and patent sales and companies can access better ideas directly from academic institutions.

However market opaqueness makes it difficult to quantify university participation. On the one hand (Love et al., 2019) found that between 1990 and 2017 more than 20,000 US patents were transferred by universities. On the other hand, using raw transfer numbers to quantify involvement is difficult as patent transfers by universities are often as a result of diverse circumstances. For example, universities often transfer IP assets to spin out companies started by universities who attempt to monetize the IP (Love et al., 2019). Indeed Autm (2017) finds that over 1000 spin outs came from universities in 2017 alone with over 10,000 startups being formed from 1995.

Universities also regularly collaborate with industry partners who fund and sometimes actively participate in research. This is usually with the understanding that the industry partner will receive rights to use the IP after the fact (Love et al., 2019). This kind of collaboration was seen in the leading example of the Tesla and the University of Dalhousie wherein Tesla funded part of the research in exchange for access to the produced IP (Oberhaus, 2019).

One common theme amongst the literature examining universities in the IP market is that it is comparatively rare for universities to transfer assets following negotiations with third parties that are unrelated to the institution (Lemley and Myhrvold, 2009). For example, universities seldom participate in the brokerage market. Between 2012 and 2016 only fourteen patent packages were sold by universities or other research institutions, which amounts to 3% of the total brokerage market in that period (Love et al., 2017; Levy, 2015).

Love et al. (2019) finds that of patents sold by universities, about two-thirds end up in the hands of NPEs. They also found that 29% of patents transferred are used for defensive or offensive acquisitions by operating companies hoping to avoid or impose litigation risk. All in all Love et al. (2019) finds that roughly 90% of the patent market from universities revolves around mitigating company litigation and risk. Resultantly, patent acquisition from universities closely mirrors that of the brokerage market with the intention of mitigating litigation risk, not the transfer of technology.

Interestingly, Zuniga (2011) finds that the majority of patent transfers occur from public universities. Similarly Love et al. (2019) concluded from their study that elite US research universities almost entirely refrain from selling patents to third parties. This is partly a result of US regulation around government funded institutions monetizing IP, in accordance with the Bayh-Dole Act COGR (1999). Additionally, elite US universities are often better positioned to litigate patent violations than NPEs with Love et al. (2019) finding that US jurors view US universities more favourably than NPEs in court.

In terms of patent sales and litigations in the public domain, a number of high profile cases have occurred in recent years. In 2010, Cornell University won a $184 million litigation against Hewlett-Packard in a case that later settled on confidential terms. According to Love et al. (2014) Carnegie Mellon University received a $1.5 billion judgment suit against Marvell Semiconductors. Despite these large litigations it is estimated by Ryan and Frye (2018) that on average university patent programs collectively earn a negative rate of return with an overall loss of more than 3%.

2.5.1 University patent incentives

Ryan and Frye (2018) defines universities as charitable organizations which aim to promote innovation for public good. Many academic institutions rely on significant government investment to support research to promote the dissemination and growth of knowledge which stimulates innovation. According to Ryan and Frye (2018) patents could, in theory, do even more to encourage universities to invest additional resources into innovation.

However Ryan and Frye (2018) acknowledge that the monetization of patents is not necessarily aligned with the goals of academic institutions. For one, the patent system uses private economic incentives to promote innovation. This creates incentives for universities to invest in patentable innovation to maximize private economic value. Ultimately this means that universities will utilize publicly funded research to generate patentable innovations for private gain.

Love et al. (2014) conducted a series of interviews with academics at a number of US universities and found that 85% of researchers did not think that patent rights are among the top four factors motivating their research and that patents do not motivate their research objectives. In fact Love et al. (2014) claims that university patent programs may reduce the quality and quantity of university research by harming professors ability to obtain funding, collaborate with others and disseminate their work as a result of legal restrictions introduced by the patents. Additionally the authors finds that fewer than 50% of spin-off companies report that having a patent affirmatively helped their idea commercialization.

Alumni (2006) argues that "universities would better serve the public interest by ensuring appropriate use of their technology by requiring their licensees to operate under a business model that encourages commercialization and does not rely primarily on threats of infringement litigation to generate revenue". This sentiment is echoed Love et al. (2019) in that the primary goal of university technology transfer is to facilitate university research commercialization.

These insights are invaluable in defining the goals of the proposed marketplace for ideas: Enable academic institutions to licence and monetize ideas while not interfering with the normal academic processes. Monetization should support the foundational research and institutions that made the research possible while not corrupting the incentives of research. Additionally, in accordance with university sentiment and the Bayh-Dole Act, universities should remain the owners of the patents and monetization should occur through a licencing scheme. This will facilitate technology transfer while minimizing opportunity for NPEs to exploit university patents through litigation.

2.6 Ocean Tomo patent auction marketplace

A core component of the mechanism presented in this thesis is a sealed bid auction used in the initial allocation of licences after the publication of an idea. It is therefore appropriate to examine the existing patent auction market more closely.

There are a number of patent marketplaces that use auctions as their main pricing mechanism. One of the most well known is Ocean Tomo who offers intellectual property related services and runs one of the worlds largest and most successful patent auction marketplaces (Love et al., 2017). In 2006 Ocean Tomo held the world's first "live" patent auction employing an open-outcry format to generate maximum audience participation and buzz (Jarosz et al., 2010).

Auctions are not new to the patent world with the first recorded auction dating back to 1993 where the IRS auctioned off patents to recover taxes (Jarosz et al., 2010). What Ocean Tomo did differently was to build hype and excitement around the auction they were holding through a live auction process (Jarosz et al., 2010). Ocean Tomo held a total of 10 open-outcry English auctions covering patent rights before their acquisition by ICAP Patent Brokerage in 2009 (Jarosz et al., 2010).

Ocean Tomo's auctions facilitated the sale of both licences and outright patent transfer (Jarosz et al., 2010). Auctions usually facilitate the sale of "lots" of patents; a bundle of one or more patents sold at the same time. Bidders preregistered for auctions at which point they were provided with a selection of curated information which aimed to decrease the due diligence required (Stathis, 2009). Ocean Tomo employed their own proprietary patent rating system which aimed to quantify an "objective" measure of patents value by looking at a number of key metrics they defined. As part of this they tried to identify patents that are suitable for a live auction (Ramer, 2006). Ocean Tomo takes a listing fee of between \$1,000 and \$6,000 depending on the lot, sellers pay an additional 15% upon conversion and buyers pay a 10% premium (Jarosz et al., 2010).

Ocean Tomo generated a total of \$114.6 million in revenue selling a total of 282 lots of patents between 2006 and 2009 (Jarosz et al., 2010). Ocean Tomo succeed in some ways and failed in others.

They succeeded in reducing the seller's search cost of marketing IP to buyers (Benassi and Di Minin, 2009). They also reduced negotiation times as an auction facilitates faster price discovery than bilateral negotiations (Jarosz et al., 2010). Additionally, they offered anonymity to their bidders through remote bidding, proxy bidding and even shell corporations set up specifically for the auction (Jarosz et al., 2010).

Some of the shortcomings from OceanTomo's auctions include the high listing and auction fees which are significant enough to decrease transaction volume (Jarosz et al., 2010). A number of lots received no bids at all, which makes listing in an auction risky. Had the auction failed, it may have been better to not enter the auction at all, as potential purchases or licensees will have observed that the perceived value of the technology is low and will subsequently offer much less than their original valuation (Yanagisawa and Guellec, 2009). Several auctions also experienced bids which did not meet the reserved price which resulted in post-auction negotiation between buyers and sellers. In fact Ocean Tomo encourages post-auction sales (Jarosz et al., 2010). The problem with this is that it undermines the point of the auction as a mechanism to elicit bidders true valuation: rather the auction acts as a means of limited price discovery which is then used in the post-auction negotiations (Wang et al., 2016).

2.7 Chapter conclusion

This chapter outlined the history of patents and their importance in modern day business as intangible assets. The market for patents was examined, along with its actors and intermediaries. Importantly, the large size and scope of patent aggregates and third parties was presented. These patent intermediaries often engage in defensive patent acquisition with the primary goal of trading patents being risk transfer. This is relevant as the contributions of this thesis aims to disrupt the patent intermediary acting as an open, free and non-rent seeking substitute for the existing brokerage market.

The implications of idea monetization for universities was also discussed. The core takeaway from this is that the proposed market should aim to provide academics with a means to monetize their research. The market should try to align with an academic institution's core goals: meaningful knowledge contribution to the commons while providing an environment for foundational research. University IP monetization should leverage a licensing scheme wherein the university remains the owner of the underlying idea through a patent. Monetization should encourage technology transfer from universities to companies to stimulate innovation and growth. Universities should in turn benefit financially for their idea contribution to said company which will ultimately support the core goals of the institution.

The next chapter examines the economics of pricing patents and the difficulties thereof. This is built on the foundation laid within this chapter and later leads to the proposed marketplace for ideas in the chapters that follow.

Chapter 3

The Economics of selling patents

Quantifying and discovering the price of an asset is a key component in any exchange. This chapter examines patents and the marketplace for ideas from an economics perspective. The complexities in pricing ideas are explored, as well as the different mechanisms that are used in today's patent marketplace. Lastly, ideas as a subset of data is discussed as well as an initial introduction into why blockchain is relevant in this context.

Economic theory justifies patents as a means to solve market failures that arise from the free rider problem[1] (Ryan and Frye, 2018). Without patents the creation of new ideas becomes a pure public good which is completely non-rivalrous[2] and non-excludable[3] (Bator, 2012). Free riding prevents inventors from recovering the fixed and opportunity cost of invention which results in market failure.[4] Through this lens, patents solve market failures in innovation by creating a temporary exclusive right for inventors which enables them to reclaim their fixed costs thus incentivising innovation (Ryan and Frye, 2018).

Patents can, however, cause market failures through inefficient rights and high transaction costs. Once a patent is granted the inventor's costs are sunk meaning that any future research and development will result in further costs which stifles marginal improvement over what was patented (Chien, 2010). There are a number of other market and search frictions which make it difficult to exchange patents which further cause market failures. These are now explored.

3.1 Challenges in pricing patents

Finding a market clearing price to exchange a patent at is difficult due to a number of reasons which make patents a unique, nuanced asset. Some of these challenges are now discussed.

[1] Free riding is the burden on a shared resource from overuse by people who don't pay their fair share.

[2] Non-rivalrous means that one's consumption does not diminish the consumption of others. This differs from a rival good that can only be consumed by one person at a time. Non-rivalrous goods tend to have large fixed costs and low marginal costs (Hryshko, 2001).

[3] Non-executable means that non-paying consumers cannot be prevented from accessing the good.

[4] Market failure is a net loss in economic value when allocation by a free market is not Pareto efficient.

3.1.1 Information asymmetry in the marketplace for patents

One of the core challenges in quantifying a patent price is information asymmetry between buyers and sellers which results in market inefficiency (McClure, 2015). This inefficiency results from uncertainty regarding patent scope, validity and value. Another element of information asymmetry is patent complexity. By definition, patents are used to protect new or novel ideas. This means that they have complex, cutting edge and highly technical domain knowledge within them. This limits the individuals that can value them which makes due diligence expensive and time consuming (Kelleyf, 2019). This exposes buyers to adverse selection.

An additional complexity is that important information in quantifying the value of a patent is often not disclosed by the seller due to legal restrictions, further compounding the information asymmetry (Jarosz et al., 2010). This makes it difficult for buyers to know how much to value a patent. For example, a patent might have an existing confidential licencing agreement associated with it that is not publicly disclosed. This information would alter a company's valuation, if they knew about it.

3.1.2 The probabilistic nature of patents

A patent is an exclusive monopoly right, but not an absolute right (Akerlof, 1970). Therefore, owning a patent does not necessarily guarantee absolute enforceability as a violation claim has to be finalized and settled in court. The enforceability depends on the patent's owner, the validity, claim coverage and how the court interprets all these factors. Added to this, litigation itself is very unpredictable due to different jurisdictions, length of court cases and how the court interprets the law itself (Odasso et al., 2015).

3.1.3 Lack of reliable quality benchmarks and market opacity

Quantifying the usefulness of a patent and coming up with a probability of patent validity (and subsequent success in possible litigation) is challenging due to the lack of discoverable and useful information (Akerlof, 1970). There is no reliable way to quantify the value of a patent or standard measure against which a patent can be compared (Lemley, 2011). These reasons further add to the complexity in pricing patents.

The goal of a market is the "preservation and use of unique information contained in the price" (Hayek, 2012). This is to say that the price of an asset in a market should fully reflect all available information in the market, thereby making the market efficient. The patent marketplace however lacks any form of transparency or benchmarks which makes quantifying fair price challenging (Akerlof, 1970). Added to this, most companies have secretive patent portfolios and most transaction information remains private. Without knowing who owns what, how much they paid for it and how it is transferred, it is hard to quantify price (Akerlof, 1970).

Bone (2010) draws a parallel to the real estate market, posing the question "if realtors can value houses why can't we value patents". In the case of real estate, however, you can compare a property to any number of comparable houses in the same area to perform relative benchmarking. You also have access to historic pricing information for similar houses in similar areas. Neither of these metrics are available for the sale of patents which makes quantifying their relative value a real challenge.

3.1.4 The patent market is a "market for lemons"

A useful parallel in the complexity of quantifying an idea's price can be drawn from a "Market for Lemons", applying the ideas of George Akerlof (winner of the Nobel Prize in Economic Sciences). Akerlof (1970)'s classic article examined the car market and outlines the reasons why only slightly used cars sell for measurably less than new cars. The model introduced in the article assumes that some cars are high quality and others are "lemons". Two separate markets would exist if buyers could distinguish between the two classes. However, buyers often can't distinguish and so the car market experiences high levels of information asymmetry with sellers knowing if their car is a lemon (or not) while the buyer has no such information. As a result, the buyer does not know if she is buying a lemon or a high quality car and ends up paying less (Akerlof, 1970). Additionally the cost of discovering if the vehicle is a lemon or not is high because it involves buying the car and driving it until it fails (Akerlof, 1970).

This ultimately means that an exchange that would benefit both the buyer and seller fails to occur which reduces efficiency (Akerlof, 1970). Akerlof commented 30 years later that "asymmetric information was potentially an issue in any market where the quality of goods would be difficult to see by anything other than casual inspection" (Akerlof, 2001).

Indeed, the patent market involves the exchange of complex assets and as there is no standard way of valuing them; there is certainly no way to quantify their value through "casual inspection". Additionally, as a result of the deterministic nature of a patent it's true value can only be discovered in court which makes the cost of finding true value extremely high, like that of the used cars. Therefore the patent marketplace is a market for lemons where some lemons are attributed value and some true high-quality patents will be undervalued (McClure, 2015).

The main insight from these challenges in pricing is that the market mechanism needs to focus on decreasing information asymmetries as far as possible. A buyer should know if they are likely to buying a "lemon patent", and if they do, the mechanism should provide a means of recourse that decreases the total incurred cost of this mistake. This publication of patent ownership information can't directly address the probabilistic nature of patents, nor can it help with the lack of reliable quality benchmarks, but it can help provide relative pricing to compare ideas as well as provide increased market information.

3.2 Different patent sale configurations

An additional consideration in pricing an idea is how the actual transfer of the patent occurs. The market for patent rights can be split into two main configurations:

1. sales that licence rights to use patents and
2. sales that lead to outright transfer of patents (Love et al., 2019).

Each of these settings have different incentives and advantages which need to be considered in how a market is designed and where the market operates (i.e. within the university context).

Bessy et al. (2011) finds that a licencing and royalty model is preferred in many contexts as it reduces the risk of buying a bad patent. If the licence holder no longer requires the licence or deems it's value lower than they had originally anticipated, it is easier to walk away from the exchange than if you've purchased the patent outright. Therefore lump sum payments resulting in the transfer of a patent require much higher presale measurement of the exact value of the technology through an extensive due diligence process (Abrams et al., 2019).

An additional consideration in licencing vs selling of patents is the rights granted to the buyer. A licencing arrangement can stipulate specific terms around how the licence can be used whereas a patent sale forgoes all control from the seller. This has an impact on the litigation opportunities for the buyer. A licence holder cannot use their licence to seek litigation against infringers as the ownership of the patent remains with the original idea creator, not the licensee.

Licencing versus patent transfer is particularly relevant in the context of university IP monetization. Ayres and Ouellette (2017) argue that the transfer of exclusive patent rights from a university to a commercialization third party makes little sense where numerous firms are already eager to use the invention without exclusivity. The goal of a university patent monetization system should be to generate revenue for the university while facilitating technology transfer, not to enable one company to gain exclusivity and opportunity for exploitation by PAEs (Fusco et al., 2019). Hence a patent licencing structure is more relevant in the case of university IP monetization.

It must be noted however that this is not strictly always the case and that a university publisher should be given the opportunity to select how they want their publications to be licenced. This design is discussed in 4.2.

3.3 Quantifying the value of a patent

Now that the complexity around patent pricing and rights is better understood, existing frameworks that quantify the value of ideas can be explored. Due to the unique and highly contextual nature of patents there are a wide array of methods of pricing. A patents value is a function of a number of factors including: the breadth of claims, how widely it is being used (or will be used in the future) and how easily it can be enforced. Ultimately how much economic value can be generated from the rights granted by the patent will define its value. Some additional considerations are now explored.

Kelleyf (2019) identifies 5 numerical indications of a patents value being: 1) the number of third party citations 2) number of citations to prior art 3) the remaining duration of the patent 4) the number of times the patent was traded and 5) the number of claims contained within the patent. Other factors that influence value are the context of the patent portfolio as many patents are complementary when used in conjunction with other similar patents (Sadao and Yoichiro, 2014). Odasso et al. (2015) further identifies methods ranging from patent renewal metrics, market based estimations, surveys of addressable markets and biographical indicators as indicators for patent quality.

Added to this, different buyers have different valuations for the same patent based on its usefulness, legal implications and future applicability within their business domain. This means that patents can have a high range of private valuations based on company and context. Akcigit et al. (2016) concludes that the private valuation of a patent is inversely proportional to the technological distance from the patent to the company. Technological distance is how relevant and applicable technology is within a company's business offering. As a result, the more aligned an idea is to a company's core offering the more the company values the patent.

3.4 Brief introduction to blockchain

The details of how a blockchain works and its implementation for the marketplace for ideas is outlined in later chapters. For the context of this chapter, a blockchain can be thought of as an immutable ledger with universally agreed upon public state. Smart contracts run within the blockchain substrate and enable the permissioned modification of the ledger through the incorruptible enforcement of preprogrammed logic.

3.5 Ideas as a subset of data

Ideas within patents are non-rivalrous as they can be copied at zero marginal cost (Ryan and Frye, 2018). At the same time, one company having access to a patent acts as a negative externality to a competitor as there is a risk of potential litigation (Lemley, 2011). Patents are also inherently combinatorial as they often make up part of a larger portfolio and their value is influenced by the other patents in the portfolio (McClure, 2015).

These three properties (non-rivalry, competitor ownership as a negative externality and combinatorial value) makes ideas and patents fundamentally similar to another emergent non-rivalrous asset class: data. Fundamentally a patent is a blueprint for how to implement a widget. Data is the physical embodiment of information which makes patents (and therefore ideas) a subset of data. Any formal process or set of instructions like an algorithm, patent or even code are simply stored information, making them all subsets of data. In fact patents are nothing more than text and images, which is data itself.

Similarly, both data and ideas are time consuming to produce while being free to replicate, creating infinite supply. Data and ideas can be copyrighted or licenced but it is difficult to track their destinations and uses once publicly released. This makes the notion of ownership fundamentally similar between between data and ideas.

As will be discussed in detail in later chapters, blockchain makes data unique as once data is registered and recorded it can be traced to its origins. This occurs through sequential hashed time stamping.

A similar principle can be applied to ideas through a blockchain based licensing scheme that helps academics monetize their research, while maintaining ownership of it. In this scheme, the university remains the owner of their IP by keeping ownership of the underlying patent. The blockchain is then used as a means to sell licences to implement these ideas. This is analogous to selling permissioned access to data registered on a blockchain. The blockchain provides a public and verifiable history of licence ownership as all transactions involving the licence can be easily located, tracked and verified.

This public record of licence ownership increases market transparency and efficiencies while decreasing adverse selection. The design details of this blockchain system are explored in the chapters that follow.

3.6 Chapter conclusion

This chapter examined the complexity in selling patents. Specifically, patents are a complex nuanced asset which makes applying any kind of evaluation framework to them challenging and expensive. A number of different sale mechanisms were also explored with the key take away being that a sub-licensing system wherein universities remain the owner of the idea in the form of a patent will be the most appropriate. How ideas, as a non-rivalrous good, relates to data was also introduced. This is relevant when framing the market and trying to find comparable models.

The next chapter takes these conclusions as an input and defines the desired properties of an ideal idea market. From this, a proposed marketplace design is presented. After this, the blockchain implementation is presented.

Chapter 4

Proposed marketplace for ideas

There are a number of market mechanisms that aim to find an optimal market clearing price. This is the price at which the quantity supplied is equal to the quantity demanded. As has been discussed, the vast majority of patent exchanges occur using private bilateral negotiations which suffer from a number of core problems resulting in poor efficiency. The quasi-public brokerage market offers an alternative but still suffers from inefficiencies such as high fees, poor allocation efficiency and adverse selection. This chapter discusses a number of desired properties in a marketplace for ideas and then proposes a market mechanism which aims to achieve them. This proposal starts at a mechanism and conceptual level and becomes more technically specific in chapters that follow.

4.1 Desired marketplace properties

The desired market mechanism should reduce search frictions, break down information asymmetries and increase allocation efficiencies. This mechanism should allocate licences to buyers with the highest private valuation. The market should be inclusive thereby providing sellers with the largest possible potential pool of buyers. Elements from the current brokered market can be improved upon, while eliminating the inefficiencies and exploitation that is sometimes present.

4.2 Proposed mechanism high level overview

To achieve these desired properties, this thesis proposes *a peer to peer intellectual property broker* which acts as a marketplace for ideas, enabling the efficient trade and monetization of patents. To reduce the scope of this problem, a subset of all patentable ideas is explored: monetizable research published from universities. This marketplace provides a brokerage service where researchers publish works onto the platform and then companies and individuals buy licences to use the ideas in a derivative or commercial work.

These licences enable one of two main use cases:

1. A company wants to commercialize the ideas presented in a paper and buys a licence to use the ideas. This licence can either be exclusive to one specific company or co-exclusive, enabling multiple companies to simultaneously commercialize the idea.

2. A non-commercial context in which an academic citing a paper acts to "buy" an attribution of social capital. Commercialization of derivative works will share profits generated with contributing academics.

Academic works within the platform remain open and free for anyone who wants to read or build on top of them assuming they don't commercialize their work. This commercialization of ideas enables academics to monetize their work while incentivizing open science.

The sections that follow focus specifically on the commercialization use case. The non-commercial applications are explored later in this chapter.

4.3 Auctions as a price discovery mechanism

To price the initial sale of a licence, a mechanism is required to elicit the true private valuations of prospective buyers and then allocate the licences to those that value them most. To achieve this, a first-price sealed bid auction (FPSBA) is used to price the initial sale of licences. Auctions are a useful mechanism for price discovery when the value of a good being sold is not explicitly known beforehand by the seller (Roberts and Sweeting, 2013). An auction is then used to find the highest possible market clearing price, thereby allocating the good to the buyer with the highest private valuation.

A first-price, sealed bid auction works by bidders submitting sealed bids, such that no bidder knows anyone else's bid. After a submission period, the auctioneer reveals all bids publically and the highest bidder pays the price they submitted and wins the auction. This is a strategy-proof mechanism which has been shown to effectively elicit the buyers' true valuation which results in efficient allocation (Mendes, 2018).

A sealed bid auction is used because it prevents a number of possible auction manipulation and collusion opportunities by bidders that are prevelent with auction mechanisms with open information (eg English outcry auction). For example a FPSBA prevents bid shilling, a practice where the seller in the auction (or people associated with the seller) artificially inflate the auction price by deceptively bidding on an item (Mamun and Sadaoui, 2013). Because bids are sealed a shill bidder can't artificially inflate the price. A sealed bid auction also preserves bidders private valuation as this information is valuable to competitors (Ocean Tomo, 2010). A sealed bid auction prevents collusion between bidders as whenever the identity of other bidders are known there is a risk that a ring of bidders can be formed to manipulate the auction results (Roberts and Sweeting, 2013). A bidding ring works by colluders agreeing only to bid against outsiders, and never against a ring member. As a result, competition becomes measurably weaker which has been shown to drastically affect the final closing price. If, however, all bids are sealed then this form of collusion is not possible (Marshall and Meurer, 2005). Lastly, a first price sealed bid auction also prevents bid sniping which is a prevalent problem in many online auction mechanisms wherein an actor will add a winning bid right above the second highest bid right before the close of the auction timer. For a detailed comparison into other bidding mechanisms and further justification into why the FPSBA was chosen see Appendix C.

Ultimately an auction acts to overcome information barriers separating buyers while reducing transaction and search costs (Jarosz et al., 2010). This mechanism drastically speeds up price discovery while eliciting true valuations from bidders.

4.3.1 Auctions with adversarial auctioneers

Sealed bid auctions are, however, susceptible to adversarial and corrupt auctioneers who collude with bidders and manipulate auction outcomes. Corruption in auctions is a major problem with the World Bank estimating a volume of US$200 billion per year spent on bribes for public sector procurement auctions (Knack et al., 2019).

Auction corruption in sealed bid auctions takes the form of the auctioneer taking advantage of their position of trust through leaking sellers preferences to other bidders for bribes (Lengwiler and Wolfstetter, 2010). This information leakage occurs during the bid submission phase wherein the auctioneer opens the sealed bids and reveals the bids of others to a favoured bidder. This means that the favoured bidder effectively plays a second-price auction while the other bidders play their bids as if they win. This collusion enables the favoured bidder to win auctions at the best possible price which breaks the mechanism and destroys efficiency as the favored bidder might not have the highest valuation and yet wins the auction.

The inefficiency cost of collusion is of the same order as the auctioneers manipulation capacity (Burguet and Perry, 2007). The manipulation capacity is the degree to which the auctioneer can manipulate the auction through exploiting their position of trust by leaking seller preferences to other bidders. The cost of a corruptible sealed bid auction mechanism is now explored.

Note that this description, and the analysis that follows, only refers to auctioneers that collude with privileged bidders by leaking information about other bidders preference before the auction has concluded, enabling the privileged bidders to modify their winning bid downwards, thereby introducing an inefficiency. This framing assumes that the non-colluding bidders are rational economic actors who bid up to their true evaluation.

4.3.2 Auctioneer manipulation inefficiency

The efficiency cost of corruption in a sealed bid auction is now explored by looking at an example of a sealed bid auction with no reserve price for a single item. In this auction there are N bidders whose valuations are $v_i (i = 1, \dots N)$ which are independent, private and distributed on the interval $[\underline{v}, \bar{v}]$. The valuations are distributed identically and independently according to a continuous distribution function F. Assume that before the auction takes place the auctioneer has reached an agreement with one of the bidders about leaking the bids of her rivals private valuation, and then allowing her to modify her original bid either upwards or downwards if she wishes to do so. For the bidder who bribes the auctioneer they effectively take part in a sequential auction where all her rivals bid in the first stage and she can then bid in the section stage after seeing information about her rivals.

For this setup, assume that only one dishonest bidder is colluding with the auctioneer at a time. If a bidder is honest then she believes that one of her rivals is colluding with probability p. Therefore an honest bidder is facing one dishonest bidder and $N-2$ honest bidders in her same situation with probability p. As a result an honest bidder, with probability $1-p$, believes they are facing $N-1$ honest rivals in her exact same situation.

The possibility of corruption has a significant effect on the auction efficiency (Arozamena and Weinschelbaum, 2009). Let d be the dishonest bidder with revised bid b_d, private valuation v_d. Let b_h be the highest honest bid amongst the honest bidders defined by $b_h = max_{i \neq d} b_i$. At the conclusion of the auction there are a number of possible outcomes:

1. If d wins the auction with her original bid then she will revise her bid downwards and set $b_d = b_h + \epsilon$ where epsilon is a small number ensuring that she wins the auction but minimizes any additional expenses that would be allocated to the seller.
2. If d loses the auction with the original bid then she has two options:
   (a) If $v_d > b_h$ then her private valuation is higher than the current highest honest bid and she will set $b_d = b_h + \epsilon$ and win the auction.
   (b) If, however, $v_d < b_h$ then her private valuation is lower than the highest honest bid and she will submit a revised bid $b_d < b_h$, or she leaves the original bid, and will lose. This means that from an honest bidders perspective they will be bidding against the dishonest rival's private valuation instead of competing against their bid.

Using this framework, one can quantify the inefficiency cost of a corrupt auction. Every honest bidder will bid lower than her actual valuation because she knows she is bidding with probability p against a dishonest rival with an unfair advantage. This is known as shade bidding (Athey et al., 2011).

Next, the magnitude of this decrease in bids is quantified. Let $v_h = max_{i \neq d} v_i$ be the highest honest valuation. Honest bidders have an asymmetric increasing equilibrium bidding strategy $b_h(v)$ which defines the highest bid as $b_h = b_h(v_h)$. In this case the dishonest bidder is allowed to examine bids and resubmit her own. If the dishonest bidder's valuation is higher than the highest bid as $v_d > b_h$ but lower than the highest true valuation as $v_d < v_h$ then she wins the auction but introduces inefficiency into the outcome. This inefficiency is $\zeta = v_h - b_h$ which can be seen in the Figure 4.1 below.

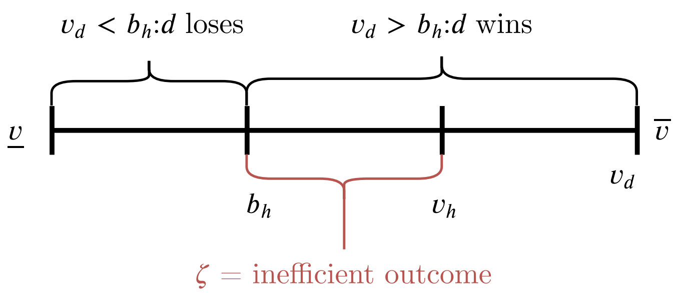


Figure 4.1: Inefficiency cost due to decreased true bidder bid amount

The inefficiency introduced by this information leakage represents the auctioneers manipulation capacity and is directly proportional to the probability of a dishonest bidder p (Arozamena and Weinschelbaum, 2009). Under a truly trusted auctioneer with no manipulation capacity $p = 0$ and the auction is efficient. This results in $\zeta = 0$ and the auction efficiently allocates the good to the bidder with the highest true valuation with no deviation between bidder's bids and their valuation.

An auction system should therefore aim to reduce the probability p of a corrupt auctioneer to zero thereby making the auction efficient through the absolute elimination of corruption. This makes the auction fair for all individuals and removes the ability for leaked bids to extract value from the auction.

A secondary inefficiency that arises from the auctioneer revealing information to a favoured bidder occurs in the case that the dishonest bidder wins the auction and then decreases their bid to just above the highest true bid b_h. In this case all the surplus that would have been allocated to the seller is lost due to the dishonest bidders ability to retroactively reduce their bid. This inefficiency is $\zeta = v_d - (b_h + \epsilon)$ where ϵ can be a small amount to ensure the dishonest bidder wins the auction over the highest honest bidder b_h. This inefficiency can be seen in Figure 4.2.

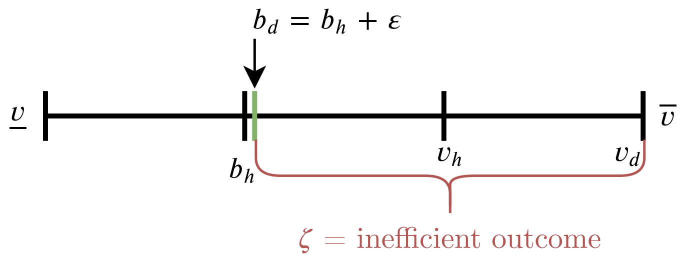


Figure 4.2: Inefficiency cost due to corrupt bidder decreasing submitted bid

If the ability for corruption or bid collusion is prevented this inefficiency cost can be removed. This will ensure that the dishonest bidder has to pay their true valuation when bidding. The goal of an auction mechanism needs to be to reduce the auctioneers manipulation capacity to zero, which in turn removes any ability for auction corruption.

4.3.3 Preventing auctioneer corruption with smart contracts

Auctioneer manipulation capacity can be set to zero by running the auction using a smart contract executed on a public blockchain. This enables the creation of *trustless, sealed bid auctions* which removes any manipulation capacity by the auctioneer, thereby ensuring optimal allocation efficiency (Xiong and Wang, 2019). How a blockchain operates and the proposed implementation that removes auction manipulation capacity are outlined in Section 5.2.1 which presents a case for blockchain within a marketplace for ideas.

4.3.4 Auction in University of Dalhousie and Tesla example

The Dalhaousie and Tesla example shows why an auction is an appropriate pricing mechanism: the owner of the exclusive battery licence should be the company that has the highest valuation of it. This results in increased allocation efficiency and maximum funding raised for the idea creator.

This example also highlights the need for a trustless auction mechanism. Say, for the sake of argument, that the University of Dalhousie listed their revolutionary battery technology for sale in a traditional sealed bid auction marketplace run by a normal auctioneer and Tesla placed a bid for this licence. A corrupt auctioneer can look at Tesla's bid and then solicit Tesla's competitor to outbid Tesla for a bribe. While the auction will then close at a higher valuation, the surplus that would have been allocated to the university is extracted by the corrupt auctioneer.

If, however, the mechanism prevented the auctioneer from knowing any private information about bidders preference through a trustless sealed bid auction, then the auctioneer would not have any ability to solicit bribes. As a result both Tesla and their competitor would have to compete in the auction knowing that neither could have corrupted the mechanism and both would therefore have to bid their true valuation. This trustless mechanism would be efficient, allocating the licence to the bidder with the highest valuation. This also ensures that the University of Dalhousie would generate the maximum revenue possible.

4.3.5 A note on the revenue Equivalence theorem

The revenue equivalence theorem states that for certain economic environments, the expected revenue and bidder preferences for a number of difference auction mechanisms will all produce the same outcome, for bidders that use equilibrium strategies(Jonathan Levin, 2000). From this theorem, it could be argued that a simpler auction mechanism (such as an English out-cry) would be economically equivalent to the first price sealed bit auction proposed, while avoiding the problem of a corrupt auctioneer as all information is public so no privileged information can be leaked. This would effectively side step all inefficiencies introduced by a corrupt auctioneer.

In many regards, this is true. However, the problems with open auctions, as discussed in 4.3 (and further in appendix C) such as bid shilling, bidder collusion, bidder rings and bid sniping are all only exist within the context of *open* auction structures, such as English out-cry. If bidders don't have any information around other bidders bids then these kinds of manipulations become impossible. For this reason, a mechanism is required that can not be manipulated by bidders (sealed bid auction) and is corrupt auctioneer proof (smart contract based auctioneer).

4.4 Fixed rate payment structure

Auctions are just one way to price the allocation of licences. What is important is that the idea seller should have optionality in picking the mechanism that they deem appropriate for their specific sales. To this end, a fixed rate licencing scheme can also be chosen by the seller wherein they define a fixed "take it or leave it" price for their patent licences. This is appropriate when the seller wants to create an unlimited number of licences.

4.5 Harberger Tax for patent licences

As an extension to the marketplace for ideas the idea seller can choose to impose a variant of Harberger Tax on the issued licence. This provides a continuous stream of revenue from ideas for idea sellers, while allocating the idea to the buyer with the highest valuation. Harberger Tax is an economic policy that aims to find a balance between pure private ownership and total commons ownership to increase general society welfare and productivity. At its core Harberger Tax balances market power while reducing the inefficiency in allocation (Posner and Weyl, 2017).

The mechanism works by introducing two principles:

1. The owner values the licence and pays tax on that value. This is a self-assessed tax.
2. At any point in time anyone can buy the licence from the owner at that price, forcing a sale. The owner is given the right of first offer to increase their licence's valuation to continue owning it.

4.5.1 Harberger Tax allocation efficiency

Allocative efficiency in this context is defined as distributing licences to the most productive owners. In general, resources can be misallocated through prices being set too high or too low which results in deadweight loss, which is a net economic inefficiency in society (Posner and Weyl, 2017). In the case of property, for example, an owner can "hold out" for a price that the buyer may not be able to pay, leading to delay or a failed transaction even when the buyer can use the property more productively than the current owner can (Posner et al., 2019). Added to this, it's always in the interest for a seller to quote a higher selling price than their lowest sell price while it's always in the in the interest of the buyer to quote a buy price less than the highest price they are willing to pay. These two forces reduce allocation efficiency in normal bilateral exchanges (De la Rouviere, 2018).

Harberger Tax increases the allocation efficiency of exchange and removes the need for a secondary market. Harberger Tax disincentivizes the seller from setting high, monopolistic hold-out prices because they will have to pay an increased tax (Posner and Weyl, 2017). At the same time it removes the additional costs due to bargaining as the seller always sets the price. This means that if someone else values the asset being sold more than the current owner they can buy it, thereby always allocating the asset to the owner that values it the most. Therefore Harberger Tax reduces investment efficiency (hold-out pricing) for a few, while increasing allocation efficiency of ownership which increases social welfare for many (Posner and Weyl, 2017). Harberger Tax aims to achieve partial common ownership. Posner et al. (2019) showed that this model achieves 70% to 90% of the maximum possible allocation of welfare efficiency while eroding only 10% to 20% of investment efficiency.

4.5.2 Harberger Tax advantages

Harberger Tax offers a number of advantages over the increased allocation efficiency in the context of patent licences. For one, it forces licence holders to self-assess the licence's worth, thereby providing the market with the most recent price information. The ability for anyone to submit a buyout for the licence introduces liquidity to the marketplace for ideas. This mechanism provides funding for the original scientific efforts that created the idea through the continuous collection of tax. Additionally, patent trolls and PAEs are financially disincentivized by being forced to pay tax on holding the patents.

Added to this, Fama (1970) introduces the notion of an "ideal market" as one that at any point in time "fully reflects all available market information". This enables companies to make better product investment decisions as the price acts as a useful signal for resource allocation. Harberger Tax aims to find an ideal price for the item being taxed, which in turn should reflect as much market information as possible. This reduces information asymmetry as the owner is setting the price.

4.5.3 Ideal Harberger Tax rate

The tax rate in Harberger Tax systems vary. Posner and Weyl (2017) proposed that the optimal rate for Harberger Tax should be about half of the observed rate of asset trade in an existing market. This insight can be used as a basis for an ideal rate for a Harberger Tax model for ideas. Akcigit et al. (2016) analyzed a marketplace for ideas wherein patents were openly traded. They concluded that across the available data about 16% of patents are sold and it takes about 5.5 years on average to sell them. Using this as an input the observed rate of asset trade is ≈3% per patent per year. Applying the half asset trade rule from Posner and Weyl (2017) yields a yearly tax of 1.5% as a starting basis for all ideas taxed with this model.

Due to the low asset trade rate of ideas, and because some ideas are traded far more frequently than others, the tax rate should also be a function of how frequently that specific idea has traded in the past. This results in frequently traded ideas paying more tax which aligns the payments that idea sellers get, with the demand for those ideas. The simple linear Equation 4.1 is used to define the tax charged per annum. n is the number of times that the licence has traded in its lifetime.

$$r = 0.015(n + 1) \tag{4.1}$$

This licence valuation starts at the sell price of the sealed bid auction. At any point in time the licence can be bought from the current licence owner, at its current valuation, assuming they raise the valuation by a minimum of 5%. The current licence owner has a right of first offer to raise the current valuation of the licence as a response to hold onto the licence. They will however have to pay more tax on the new valuation of the licence if they choose to beat the new offer. Importantly Equation 4.1 can be modified by the seller if they choose. It is set to 1.5% per year as a suggestion.

4.5.4 Buyout implications for seller

The risk of losing a licence is potentially devastating for a company who has built their whole business around the idea. This potential loss can be mitigated by two factors:

1. The licence holder can always increase their private valuation in the event someone tries to perform a buyout through the right of first offer.
2. After licence revocation, the previous holder is permitted to keep their licence for a period of time. This serves as a grace period which enables the company to find another way to licence their product, create a new product or buy back the licence.

This grace period should be long enough that companies that have their licence revoked are not immediately put out of business. It should be representative of how much time the company had held the patent for. To keep the implementation simple the period of validity after licence revocation is set to $t_g = \frac{t_h}{2}$ where t_g is the grace after revocation and t_h is how long the ex-licensee held the licence for. Alternative constructions for this equation are also possible.

4.5.5 Tesla example with licence buyout

Say for example Tesla thinks that the battery idea from the University of Dalhousie is worth $1 Billion. If they buy access to the licence from the proposed market there is an operational risk to them of losing the licence equal to how much they think the licence is worth. Say Tesla sets their valuation on the market to their private value and Toyota successfully buys it out for $1.1 Billion, resulting in Tesla losing the licence. Tesla will in fact generate more revenue from the buyout than the cost of losing the licence because they were paid more than their private valuation of the licence. The buyout period comes into play here in that Tesla can re-evaluate if they want to buy the licence back.

4.5.6 Harberger Tax on ideas and the commons

A key element of Harberger Tax is that it attempts to balance pure private ownership and total commons ownership to increase the welfare of society at large (Fama, 1970). Patents are seen as a temporary monopoly given to a company enabling them to recoup costs of R&D, thus incentivising and protecting innovation. However, patents expire after 20 years and then their ideas enter the public domain.[1] At this point the intellectual property can be seen to be owned by the commons.

Harberger Tax therefore provides a taxation mechanism on this monopoly right until the idea enters the commons. Additionally, because universities are seen as a common good, sending tax to the institution that funded the idea directly benefits the commons until the idea fully enters the public domain at patent expiration.

4.5.7 Disabled transferability

Items that have an applied Harberger Tax should not be able to be traded in secondary markets. This is to ensure that all licences are traded through the Harberger buyout process. Through the submission of buyout requests by bidders and potential subsequent valuation raises by the current owner, the market for patent exchange is always maximally efficient.

4.5.8 Unit licence exclusivity

Each licence can only be implemented by one company at a time (excluding during a buyout). This is to prevent holding companies from sublicensing and charging fees. This rent seeking behaviour would extract value from the marketplace and decrease the overall welfare.

[1] This is the case in South Africa (CIPC, 2019), US (Joe Runge, 2016) and worldwide (WIPO, 2019).

4.5.9 Seller influence on market dynamics

The seller who publishes the idea can choose how many licences they want to issue into the marketplace. This optionality coupled with the choice between an auction or fixed rate sale and the ability to enable Harberger Tax provides the academic with an influence over how the market will operate. As a result the seller can be strategic based on how they perceive the market interest in their specific patent.

An example configuration is to set the max concurrent license to one and enable Harberger Tax. This would create an artificial monopoly on the idea for whoever holds the licence. In the case of the Tesla example this might be the situation that the University of Dalhousie would want given their prior arrangement with Tesla. Alternatively, maximum concurrent licences could be set to two and enable Harberger Tax to create a duopoly on the idea.

Multiple licences coupled with Harberger Tax results in all licence valuation tending to the same average for that specific patent. This is a result of the lowest valued licence always receiving the most bids. This will drive up this licence value as licence holders try to prevent buyouts. At the same time the highest valued licence has an unnecessarily high valuation as it won't get attempted buyouts as these are all taken by the licence with a lower valuation. This will drive down the price of the highest valuation.

4.5.10 Harberger Tax in the Tesla example

In the case of the Tesla example, at the conclusion of the auction Tesla would be issued a licence which would have a Harberger Tax associated with it. This results in continuous taxation of the licence which can be claimed by the University of Dalhousie. At any point any other user can submit a buyout proposal. For this example say Toyota is the competitor. After the buyout submission Tesla has two weeks to submit a counter offer that exceeds Toyota's buyout proposal.

Tesla will continue to increase their licence valuation until it exceeds their future private valuation of the licence considering all costs and profits. If Toyota continue to bid past Tesla's private valuation then Tesla does not lose out from a monetary perspective as a result of licence revocation because they make their future value at the conclusion of the buyout.

One could argue that this mechanism enables Tesla to "lock in" the future valuation of the licence while removing any uncertainty. If Tesla holds onto the licence there is a non-zero chance that they can not ever achieve their self defined future valuation. However, if they are bought out by Toyota they are guaranteed to receive that money in cash.

4.5.11 Implementing Harberger Tax with smart contracts

A complexity in dealing with Harberger Tax is enforcement as it considers both payments for taxation and licence reallocation if the licence holder is insolvent. A centralized solution results in high administrative overhead for paying taxes, and relocation after an auction (Posner et al., 2019).

One solution to this is to automate the whole process using blockchain based smart contracts. A smart contract system enables idea creators to claim outstanding tax at any point in time without requiring any centralized administration. In the event that the licence holder is not solvent, a smart contract system can automatically list the licence for reauction. A smart contract system will also prevent any kind of manipulation of the taxation system by the centralized provider. Lastly, smart contracts can be used to prevent licence transfer thereby preventing secondary markets and enforcing that all exchange occurs through the Harberger Tax mechanism. The full details for this design and rational are discussed in Section 5.2.2 with implementation in Chapter 6.

4.6 Public record of patent licences

An additional component of the proposed design is a public record of all patent licences. This is achieved using a blockchain as a publicly displayed history of all licences issued from the market with full transparency and auditability on who owns them, how much they paid for them and the ownership trail. Additionally all terms and conditions for the licences are publicly shown. The full design and implementation details of this public registry for licences issued from the market can be found in Section 5.2.3.

4.6.1 Increased market efficiency through market transparency

The primary reason for making the market as transparent as possible through publishing patent ownership history is to decrease information asymmetries and enable relative pricing. If this information is publicly accessible buyers will be more informed around the true valuation of an idea. This transparency will aid in effective Harberger Tax price allocation as licence owners will have additional information to inform their private valuations.

4.6.2 Licensee strategic implications from public information

One downside to publicising all licence holder information is the loss of strategic advantage by licence holders. As soon as your competitors know about the licences you hold you lose a strategic advantage. This can result in a reduction of the private valuation of the licence. Additionally, this might make the proposed system a non-starter for some industries. In others this marketplace transparency should not be a problem. Whether or not this is an acceptable public display of information will depend on the industry and business involved.

For one, the current quasi-public brokerage market is already mostly open and so there is a large opportunity to disrupt this sector. Secondly, the advantages of reflective pricing, market efficiency and liquidity brought about through public information outweigh the disadvantages for loss of strategic advantage in most cases.

4.7 Automatic programmable payments

Payments in the proposed marketplace should automatically execute after auction conclusion and after Harberger Tax is claimed. Ideally this payment system should require as little administration as possible while providing strong security guarantees to market participants. It should also be easy to transact in different currencies on the platform enabling a wide range of market participants.

Blockchain enables low cost, trustless, cross border payments. Its applicability in the context of payments is examined in section 5.2.4.

4.8 Platform fees and monetization

A core goal of the proposed marketplace is to provide value to the commons while an idea is protected by a patent until the idea returns to the commons at expiry. The proposed market aims to remove rent seeking behaviour from middlemen who provide limited value to the inventor and other market participants. Rather, value generated from the market is directed back to the creators thereby stimulating innovation and supporting the commons. To this end, all transactions are free in the platform. This includes auction payment settlement as well as Harberger Tax enforcement. This is vastly different to the majority of existing brokerage services that offer a comparable scope of features who tend to charge exorbitant feeds.

Platform monetization is not a direct consideration for the context of this thesis. Future implementations of the marketplace could offer additional services on top of the provided free exchange, such as professional due diligence and collation services which could be charged for.

4.9 Attribution of social credit

Upon listing an academic work the author can choose to share a fraction of the proceeds with their citation's authors or co-authors. This enables value to be attributed to an influential author or fellow contributors. This can be extended to sharing revenue with the publisher's institution.

As an extension, citation networks can be built by analyzing the papers referenced within a publication and then a fraction of generated revenue can be shared with the most influential contributors. By identifying papers that have the highest number of references as well as considering cross references, a papers "influence" score can define how much a citation would get paid relative to other citations. This mechanism aims to indirectly support foundational science and research by giving back to the most influential contributors to a specific idea. This thesis was born from an academic project called UniCoin which implemented an initial version of the marketplace for ideas proposed. One of the original goals of UniCoin was this notion of a citation network to distribute proportional revenue among contributors. This concept of citation networks is being actively researched in parallel with Meiklejohn (2020).

4.10 Licence for issued publications

The model presented grants patent rights through licences issued by the patent holder (the university). After the sale, the university remains the patent holder which is congruent with the requirements of Bayh-Dole thereby making the marketplace legally permissible in the US (COGR, 1999). The Bayh-Dole act defines the process that US universities must follow if a federally funded publication is patented. This is relevant as the US has the highest number of litigation cases out of any major patent issuing country and therefore it is important that the market can operate in the US. This design decision also makes the licencing process compliant with South African Copyright Guidelines and rules for works published from universities (Wits, 2019).

Licences issued need to fit within the regulatory framework within which the patent was filed. These non-exclusive licence agreements are bespoke documents signed between the university and the licensee. WIPO (2019) has an example non-exclusive licence agreement template generated from Harvard. The translation from this legally binding agreement to a blockchain token is not unique to this thesis and is an area of active research for a number of blockchain companies. Further discussion on the complexities of this process are delegated to Section 7.1.3.

As has been mentioned, all papers within the market remain open and free for anyone to download and read. Other researchers can read them and build further research around them, assuming they do not commercialize the ideas presented within. This aims to break down the current exploitative monetization framework that surrounds a lot of cutting edge research, such as Elsevier's paywall (Else, 2019). In the proposed model value flows from the licence holder to the original researcher. this supports the institution and ultimately supports open science.

4.11 Chapter conclusion

This chapter introduced the conceptual elements of the proposed marketplace for ideas. This includes the use of a sealed bid auction for the initial allocation of licences to companies that value them the most. This is appropriate as a real world sale of ideas will most likely have a number of competing companies who all have their own private valuation of the idea which will inform their bidding price. An auction therefore ensures that the maximum revenue can be generated for the idea seller. An auction decreases the matching costs between a seller and buyer. This is important in the university context as the process of selling an idea needs to be as simple and approachable as possible for an academic. However, auctions are susceptible to a corrupt auctioneer who can extract value from a sale. As a result, this inefficiency was considered to find the impact of auctioneer manipulation on auction efficiency. A Harberger Tax mechanism is also presented which requires licence holders to define their own valuation of the licences they hold. This provides representative pricing and creates market liquidity for licences. Considerations and examples associated with the Harberger Tax are also discussed, including the strategic implications of buyouts. How payments are processed, the creation of a public licence registry and platform minimization are also discussed.

The implementation details of each component at a theoretical and technical level are explored in the next two chapters. This starts with a case for blockchain within the context of the marketplace for ideas and is followed by a technical implementation discussion.

Chapter 5

A case for blockchain

This chapter takes the conceptual elements presented in the previous chapter and discusses how these can be built using a blockchain system. Importantly the rational into why blockchain is an appropriate solution for the proposed market is identified.

5.1 Blockchain technology contextualization

A brief contextual understanding of blockchain is required to understand how it can be used to create the proposed marketplace. This begins with an introduction into blockchain and is followed by an overview of Ethereum and smart contracts.

5.1.1 Blockchain

A blockchain is a peer-to-peer, digital ledger, used to record transactions chronologically. It represents an immutable, persistent record that can never be altered or changed by anyone due to the distributed nature of the consensus algorithm. Miners, all around the world, offer computational power to verify transactions (in blocks) to secure the network in exchange for rewards. This is known as proof-of-work consensus. The distribution of this consensus algorithm results in the inherent security of the network: if one wants to alter the ledger, one would require more than half of the total computational power, commonly known as a 51% attack. Note that there are other consensus algorithms, such as proof-of-stake which are far more energy efficient, but the underlying immutability concepts remain the same.

Blockchain was first proposed in 2008 by Satoshi Nakamoto. In his seminal white paper he introduced a blockchain version of a "Peer-to-Peer Electronic Cash System" known as Bitcoin (Nakamoto, 2008). Its invention was "hailed as a radical development in money and currency, being the first example of a digital asset which simultaneously has no backing or intrinsic value and no centralized issuer or controller" (Vitalik Buterin, 2018). Digital cash is, however, only one of hundreds of potential use cases of blockchain technology.

Other uses of this technology include bespoke application specific blockchains to represent digital assets to create currencies[1], proof of ownership[2], and tokenization of non-fungible assets like domain names [3].

5.1.2 Smart contracts

Other blockchain platforms use smart contracts which execute code in a Turing-complete virtual environment[4]. Smart contract platforms act as a generic execution substrate, enabling the implementation of arbitrary logic. Smart contracts can be used to create all the above example implementations in a few lines of code. Smart contracts can make many other complex systems like fungible[5] and non-fungible tokens[6] (including stablecoins[7]), complex financial products (DeFi)[8], governance mechanisms (DAOs)[9] and self sovereign identity systems[10].

The immutable nature of the blockchain ensures that the logic defined within the smart contracts is executed correctly while preventing all third parties from manipulating it. In order to change the execution logic or outcome one would need to attack the Ethereum blockchain itself which is prohibitively expensive as this would involve require buying and maintaining 51% of the network hash rate to modify the execution or state of a smart contract.

5.1.3 Ethereum

For this thesis, the Ethereum blockchain is used. For a full rational see appendix B.

Ethereum aims to create the ultimate protocol for building decentralised applications. Ethereum implements a Turing-complete virtual machine called the Ethereum Virtual Machine (EVM) which is built into its blockchain. This enables anyone to create and deploy smart contracts to the public network (Vitalik Buterin, 2018). Ethereum can be thought of as a "world computer" that enables arbitrary code and logic to be executed in

[1]Bitcoin is only one of many cryptocurrencies who aim to act as a "digital cash". Others include Zcash for private transactions, Dash for scalable commercial payments.

[2]Proof of ownership, such as "smart property" where ownership of an asset is defined by a blockchain contract. Examples include wiki (2012) where a house is owned by Bitcoin script.

[3]Namecoin (2010) for example creates a blockchain representation of the ownership of a domain name.

[4]Turing-completeness defines the ability for a computational system to execute any arbitrary logic.

[5]A fungible token can be used to represent anything that is divisible.

[6]A non-fungible token(NFT) is used to represent something cryptographically unique and is non-divisible. They introduce digital scarcity into applications. For example Cryptokitties lets you trade digital cats. They reached impressive size and volume during their peak trading periods with 3.2 million transactions and the most valuable cat being traded for 110 000 USD.

[7]Stablecoins have a fixed value against a reference currency. For example Dai is pegged 1 to 1 to the USD. This is discussed in detail in section 5.2.4.1.

[8]DeFi is decentralized finance. DeFi implements complex financial products like bonds, swaps and futures built on the blockchain without an intermediary. This decreases trust requirements and costs while increasing market access to a world wide audience. For example UMA who are building a generic substrate for running financial contracts on the blockchain.

[9]DAO is a Decentralized Autonomous Organization which represents an organization who's decisions are made through votes of its members. This system has hard coded rules programmed into the blockchain. Example: Aragon who build a generic frameworks for deploying Daos in real world settings.

[10]Self Sovereign Identity describes systems wherein individuals are given ownership and control over their digital identities without intervening administrative authorities. For example Uport who have created a platform for issuing blockchain based identities and certificates which are owned by individuals.

a trustless, immutable way. The network is public, enabling anyone to contribute to the consensus, transact or interact with smart contracts.

5.1.4 Solidity

Smart contracts for Ethereum can be written in a multitude of languages that are all compiled down to Ethereum bytecode, the stack-based language, executed by the EVM (Solidity, 2018). The most popular language for writing smart contracts is Solidity, a "high-level imperative statically typed language compiled to EVM" (Vitalik Buterin, 2018). Solidity is considered "contract-oriented" with semantics similar to C++ and Javascript (Solidity, 2018).

For the purposes of this thesis, in an attempt to keep the conceptual elements of blockchain simple, smart contracts can be thought of as automatically executing software that run when specific predefined conditions are met (Vitalik Buterin, 2018).

5.2 Use of blockchain in the marketplace for ideas

Ethereum smart contracts enable the creation of tamper-proof, corruption free and verifiable mechanisms. Smart contracts are censorship resistant in that no centralized entity can control or modify their execution. Smart contracts are used to remove any trust assumptions that would usually need to be placed into third party custodial services.

The four main components of the marketplace for ideas where blockchain is used are:

1. Running trustless, corruption proof sealed bid auctions,
2. Automatic and trustless enforcement of Harberger Tax,
3. Store of public record of tokenized patent licences and
4. Instantaneous, cheap and non-custodial payments.

The rest of this chapter discusses how blockchain is used to achieve these four components.

5.2.1 Building trustless auctions using smart contracts

There are a number of constructions available to create trustless sealed bid auctions using smart contracts (Galal and Youssef, 2019). The simplest is a commit reveal scheme wherein bidders submit a cryptographic hash of their bid to a smart contract during a commit phase and then reveal them during the reveal phase.[11] At the end the bidder with the highest submitted value wins the auction. This mechanism guarantees that no private information is leaked during the bidding phase, thereby preventing collusion. This mechanism also enables any other bidder to validate the correctness of auction execution by just observing the public blockchain.

[11] A cryptographic hash is a one way deterministic function which maps an arbitrary length input to a fixed length output. It effectively generates a unique digital fingerprint for the input. Because it is one way, there is no way to infer the input data string (in this case the bid) from the hash.

In relation to Section 4.3.2 which discussed the impact of a corrupt auctioneer on auction efficiency, this commit reveal mechanism prevents any possibility of auctioneer manipulation by building the auction protocol on trusted primitives (i.e. proven secure one way hash functions and the blockchain itself). As a result, there is no need to place any form of trust within the auctioneer or third party to run the auction. This means that the probability p that the auctioneer is corrupt is set to zero which results removing all inefficiency from corruption as $\zeta = 0$.

5.2.1.1 Trustless sealed bid auction mechanism

The process of running a trustless auction using smart contracts is now explored. Say a single item is being sold at auction, with a reserve price R. In this auction there are N bidders whose bids are $b_i(i = 1, \dots N)$ in the interval $[\underline{b}, \bar{b}]$. The auction is split into four phases: pending, commit, reveal and finalized.

1. **Pending phase:** before the auction starts, potential bidders will conduct due diligence and the seller can advertise the auction. This has a recommended duration of upwards of 6 months but can be changed by the seller depending on their preference.

2. **Commit phase:** bidders submit sealed bids to the blockchain. No one can see the valuations of other bidders at any point. This has a recommended duration of 2 weeks but can be changed by the seller depending on their preference.

   This phase involves the bidder generating a random number, called a `salt`. This is used to add entropy to their `bidHash` commitment. When generating the `bidHash` commitment, the `salt` is concatenated with the bidder's `bidAmount` before applying the hash function. The bidder keeps their `bidAmount` and `salt` private until the reveal phase.

   The `salt` is used to prevent an adversary from guessing the `bidAmount` using the `bidHash` through an iterative brute force attack. Without the added entropy an attacker could generate thousands of hashes until they find the bidder `bidAmount`. If the `salt` has sufficient entropy, say 256 bits, then this attack becomes infeasible.

   Let bidder i's random salt be s_i which is generated using a random number generator with a large range running within the bidders browser.[12] The `SHA3` hash function is used to generate the bid hash, defined by $H(x)$.[13] The bid hash commitment h_i is therefore defined by Equation 5.1.

   $$h_i = H(b_i \frown s_i) \tag{5.1}$$

   $b_i \frown s_i$ represents the concatenation of the values b_i and s_i. For example if the bid was $b_i = 1234$ and the salt was $s_i = 5678$ then $b_i \frown s_i = 12345678$.

3. **Reveal phase**: after the auction has finished, bidders return to the platform to reveal their `bidAmount` and `salt` to the blockchain smart contract for verification. If

[12] This random number generation process must occur in the bidders web browser such that it is shielded from the outside world. If someone knows this salt they can brute force to find the bidder's bid amount.

[13] SHA3 is implemented in the bidders browser using Javascript.

a bidder does not reveal their bid it is forfeited. This has a one week recommended duration.

This phase involves submitting the bidAmount and salt in a transaction to the smart contract. The smart contract then checks if these match the bidHash h_i commitment from the commit phase. If they match, the bid is marked as valid and is eligible to win the auction. If they don't match, the bid is marked as invalid and is removed from the auction.

Verification occurs within the smart contract through the implementation of the predicate Function 5.2. Let the submitted bidAmount and salt presented to the smart contract by the bidder be $\hat{b_i}$ and $\hat{s_i}$. These are the values the bidder is claiming to be the values they committed to during the commit phase. The predicate function is defined as follows for bidder b_i's sealed bid and known captured bidHash h_i.

$$\phi_i = \begin{cases} 1 & if \quad H(\hat{b_i} \frown \hat{s_i}) = h_i \\ 0 & if \quad H(\hat{b_i} \frown \hat{s_i}) \neq h_i \end{cases} \tag{5.2}$$

If the hash of the claimed bidAmount and salt $\hat{b_i}$ and $\hat{s_i}$ generate the hash originally submitted by the bidder h_i then the bid is valid. Otherwise it is not valid and is rejected. After a bid is deemed valid the verified bidAmount and salt are stored within the smart contract. This enables on-chain selection of the highest bid and external validation of the correctness of the bidding process by any third party.[14]

4. **Finalize phase:** the auction is finalized by finding the winning bid and performing payment from the winner to the seller. The auction remains in this phase henceforth (it is forever finalized, assuming there is a winner).

   The smart contract implementation of this function involves looping through all the bids submitted to identify the highest bidAmount. The highest bid b_h is defined by Equation 5.3.

$$b_h = max\,\{b_i : i = 1 \ldots N\} \tag{5.3}$$

   After identification of the highest bidder, this is checked against the reserve R. If $b_h >= R$ then payment is made from the winner to seller.

This mechanism is used when initially auctioning off a licence on the platform as well as for the resale of licences after the licence holder fails to pay their Harberger Taxes.

5.2.1.2 Commit reveal sealed bid auction liveness considerations

The mechanism presented guarantees that no private information is leaked from bidders by the auctioneer and that no bids can be modified after they are committed. This in turn, eliminates any opportunity for corruption making the auction maximally efficient.

This mechanism however introduces a liveness trade off due to the asynchronous nature

[14] An observer can check that the submitted bid matches that of the committed bid. As a result the bidder and seller know that the auctioneer could not cheat because they trust and can observe the mechanism. This enables them to verify the legitimacy of the auction without needing to trust anyone.

of the commit reveal process. Specifically, a bidder can commit a bid and then not return online during the reveal phase to reveal their bid. In this case the bid is lost, even if they would have won the auction. This also gives the bidder the option to withhold revealing their bid if they can see that other revealed bids would result in them losing the bid, thereby enabling them to preserve their private valuation.

To deal with this liveness consideration, the auction phases are set to run over a long period of time. The reveal phase, for example, is a full week. This gives the bidder ample opportunity to come back to the platform during the reveal phase.

Additionally, an email service can be set up to direct the bidder back to the platform so they can reveal their bid. This system would email the bidder periodically with a direct link to reveal their bid. If the bidder does not reveal their bid promptly additional emails can be sent until such time that the bidder reveals or the auction is over.

It is important to note that a centralized solution to the liveness problem is to get a third party to store the `bidAmount` and `salt` for the bidder. This third party can then perform the reveal on the bidders behalf and as such the bidder does not have to return online. While this solution does solve the liveness consideration, it opens up the auction to corruption wherein this third party leaks information to other bidders about the bids they are responsible for revealing.

5.2.1.3 Alternative sealed bid auction mechanisms

There are a number of other mechanisms that have been presented in the literature to run sealed bid auctions. Some of these use zero knowledge proofs or other complex cryptography like time-locked encryption schemes (Galal and Youssef, 2019; Xiong and Wang, 2019; Sonnino et al., 2019). These methods are, however, far more complex and require much more sophisticated trusted step ups, execution and implementation. Additionally, some of these methods introduce other trust assumptions that degrade the overall system security. For example many of the zero knowledge implementations require a trusted set up and time-locked encryption requires a trusted time server.

A commit reveal mechanism, while having a liveness consideration, provides strong guarantees for correctness and security. With that said, the mechanism used in this thesis is modular and therefore could be swapped out to a more sophisticated mechanism in the future if wanted. For further details into alternative implementations see Appendix C.

5.2.2 Automatic and trustless enforcement of Harberger Tax

Harberger Tax collection can be built using smart contracts thereby enabling the licence issuer to claim outstanding payments at any time. If a licence holder is insolvent, the smart contracts automatically place the licence up for auction. This whole process can operate without any administrative oversight in a completely decentralized and autonomous way. This decreases cost and chance for human error. Any opportunity for a corrupt enforcer is also removed.

5.2.2.1 Outstanding tax calculation

The amount of tax owed is a function of the self assets price p, the tax rate r and the duration from when the last payment was made. This duration is represented using unix timestamps and is expressed as running from t_1 to t_2 where t_1 is the last payment date and t_2 is current time. The duration, as a fraction of a year, is therefore expressed as $\tau = \frac{t_2 - t_1}{60*60*24*365}$. The Function 5.4 below shows outstanding tax O to be paid from the last payment until the current timestamp.

$$O = p(e^{r\tau} - 1) \tag{5.4}$$

This expression enables the current outstanding tax to be calculated at any point in time. This means the licence owner can claim their tax when they want as it does not have to be restricted by a tax payment schedule.

5.2.2.2 Licence holder solvency

When the licence issuer calls to claim tax on an outstanding licence, the licence holder must be solvent in order to make the payment. If they are not solvent then the licence is revoked and placed for auction. Licence holder solvency is found by evaluating the inequality shown in the predicate Function 5.5 where s_i is either zero or one depending on if bidder i's balance b_i is more than their outstanding debt O_i.

$$s_i = \begin{cases} 1 & if \quad b_i > O_i \\ 0 & if \quad b_i < O_i \end{cases} \tag{5.5}$$

At any point in time the licence holder can claim this tax and if Function 5.5 evaluates to 0 then the licence is revoked and placed on auction. The auction process follows that of initial licence allocation outlined in section 5.2.1.

5.2.3 Non-fungible tokenized patent licences

The blockchain acts as an excellent mechanism to store a public record of *who owns what*. Ultimately this is the point of a ledger: a chronological list of asset ownership. In the context of the marketplace for ideas, a non-fungible token(NFT) is created to represent the ownership of a licence. A NFT is a special kind of token used to represent something digitally scarce and unique. The prototypical example of NFTs is CryptoKitties. This is an Ethereum application wherein NFTs are used to represent in-game digital cats. As of late 2019 over 1 million Kitties were created and traded with totalling \$80 million in trades (Cryptostats, 2019).[15] NFTs are implemented in Ethereum using the ERC721 token standard. This standard defines a number of common interfaces and type definitions to create a compliant token.

NFTs are, by definition, not interchangeable. This is quite different from a traditional cryptocurrency token such as Bitcoin or Ether which are fungible and therefore interchangeable: one Bitcoin is always equal to another Bitcoin. However, patent licences for

[15] The total traded volume was 57,668 Ether, with an all time high value of \$1,400 per Ether equates to \$80 million in total volume traded for crypto kitties.

one company are not interchangeable for another as they have different terms, conditions and owners. These terms, details of trade and other information are embedded into the NFT to give a unique, tokenized version of a licence.

5.2.3.1 Storing licence information within an NFT

A key part of an NFT is attaching metadata to a token. Each licence created from the marketplace stores the purchase price, the licence owner, the unique licence number and the licence status within the token. The status indicates if it is currently active or if it has been revoked. The structure can be dynamic to enable future fields to be added.

5.2.4 Programmable payments using smart contracts

Smart contracts enable programmable money. This is powerful when complex payments need to be performed between a number of recipients. In the marketplace for ideas, there are two main situations in which payments are made:

1. At auction finalization, the winning bidder pays the issuer and all contributors.
2. When claiming Harberger Tax, the licencee pays the issuer and all contributors.

Both situations involve receiving one payment from the payer and splitting this into multiple components for the respective recipients. This can involve quite a number of payments in the case where a patent has many contributors to be paid. A traditional centralized way to implement this would be to bundle payments for contributors over a period of time and then perform one bulk payment for each contributor. For example, payments could be processed once a month to all contributors. Using smart contracts these payments can be processed automatically immediately following an auction.

5.2.4.1 Stablecoin payments

Ethereum has a number of stablecoins that are actively traded. A stablecoin is a blockchain token that is pegged 1:1 with a fiat currency, such as the Dollar. They therefore have all the benefits of a cryptocurrency (programmable, cheap & fast transactions, trustless ownership etc.) while not suffering from the volatility of the rest of the crypto market.[16][17] In this way a stablecoin brings reliability to crypto which makes them far more appropriate for real world applications.

There are a number of stablecoin constructions on Ethereum. They differ primarily in how token collateralization works. This thesis utilizes the Dai stablecoin to handle all payments on the platform. For a full rational into this decision as well as a comparison to other stablecoins available see appendix D.

Dai is an ERC20 token with all standard token functionality enabled.[18] Dai is one of the few truly decentralized and trustless stablecoins that is not controlled by any centralized

[16] A fiat currency is any currency that is backed by a sovereign government (Rands, Dollars ect.)

[17] Live Coin Watch (2019) shows that between 2016 and 2019 the price of Ether went from a low of $1 to over $1400 back down to less than $100.

[18] ERC is the Ethereum Request for Comment which enables community members to propose standard interfaces for common implementations within the Ethereum ecosystem. The ERC20 is the standard interface for making fungible tokens on Ethereum. All major tokens issued today conform to this standard.

issuer. It uses overcollateralized Ether deposits and incentive alignment to help bring stability while maintaining decentralization (MakerDao, 2017). Importantly however, the design of the system is not locked into exclusively using Dai and future iterations could easily use another payment token if desired.

Running payments on the platform with Dai enables very low cost transactions. In late 2019 the cost of performing a Dai transaction was of the order of a fraction of a US cent per transaction.[19]

5.2.4.2 Payment splitting

Splitting the payment is done by multiplying a fraction to be paid to each contributor by the total payment amount. Say a publication has N contributors. Let the total amount to be paid for the interaction be T. Interactions are the sale of a licence or claiming Harberber Tax. Let the fraction that each contributor receives be defined by $\alpha_i(i = 1, ...N)$ where $\sum_{i=1}^{N} \alpha_i = 1$ such that the total amount being paid is divided amongst the contributors. Each contributor payment p_i is therefore defined by Function 5.6 below. Each contributor is transferred this amount of tokens p_i at the conclusion of the payment splitting.

$$p_i = \alpha_i T \tag{5.6}$$

5.2.4.3 Permissioned payments

The ERC20 token standard enables a user to grant permission to another wallet to move tokens on their behalf. This is useful as it enables the smart contract system to fulfil token payments on the users behalf. For example when creating a bid the bidder will grant the smart contract system permission to move tokens out of her wallet. At the conclusion of the auction the smart contract can then automatically extract the required payment and send it to the recipients without requiring any additional input from the bidder.

5.3 Chapter conclusion

This chapter contained a high level overview of what blockchain is, how it works and why it is useful in building a marketplace for ideas. The four key mechanisms used (sealed bid auctions, automatic Harberger Tax, non-fungible token licences and programmable payments) were outlined from a theoretical perspective. This included a generic functional definition of how to build these mechanisms.

One key takeaway is that blockchain enables the implementation of immutable, highly tamper resistant mechanisms that can be used to construct marketplaces. Smart contracts remove middleman from the construction and operation which reduces costs of operation. This decreases the cost to the end user. Blockchains deterministic nature coupled with an immutable history result in strong security guarantees to the robustness of mechanisms deployed on them.

The next chapter examines the implementation details of the marketplace from a technical and implementation perspective. This implements the ideas presented in this chapter.

[19] $TransactionCost = gasUsed * gasPrice$. For a standard ERC20 token transfer is $\approx 0.001 USD$.

Chapter 6

Marketplace implementation details

The implementation of the marketplace proposed in the previous chapters is now presented. This discussion starts with a high level overview of the different system components and then each component is discussed in detail. Additionally, a number of design considerations, risks and challenges and what was done to overcome them is discussed.

All code for this thesis' implementation is open source, and is published on Github here. This repo contains all code for all components discussed in this chapter. This Github repository is a fork of the original Unicoin project, which can be found on Github here. The final web dapp (decentralized app) can be found online at unicoin.win. Alternatively, there are screenshots of key components presented later in this chapter.

The code contribution of this thesis is a full redesign of the smart contracts from the original project wherein all implementation logic was rebuilt from the ground up to accommodate the sealed bid auctions, Harberger Tax and a number of further improvements are discussed later in this chapter. Additionally, the front end implementation was radically overhauled for the thesis implementation. All smart contracts have been deployed on the Ethereum Kovan testnet. Deployment details can be found on Github here. Details on how to run the code presented in this chapter can be found within the Github repository here. Application screenshots for the implementation can be found in Appendix I.

6.1 High level system overview

There are three main layers to the technology stack:

1. Web3 layer consisting of a series of Solidity smart contracts which handle all core platform functionality including registration of publications, operation of auctions, settlement of payments, Harberger Tax enforcement and allocation of NFT licences.

2. Web2 layer front end application which runs in the users browser and acts as the primary interface for all user interactions with web3. This includes connections to academic profile validation services like ORCID, submitting transactions and browsing ideas for sale.

3. Decentralized file hosting layer for publication and profile information stored on the InterPlanetary File Storage (IPFS) network. This includes storage of PDF files for publications.

Each of these layers is examined in the sections that follow, along with how they interconnect. Supplementary system components, like user private key management are also discussed when appropriate. Figure 6.1 shows a high level interconnection of these three layers as well as all key technology components.

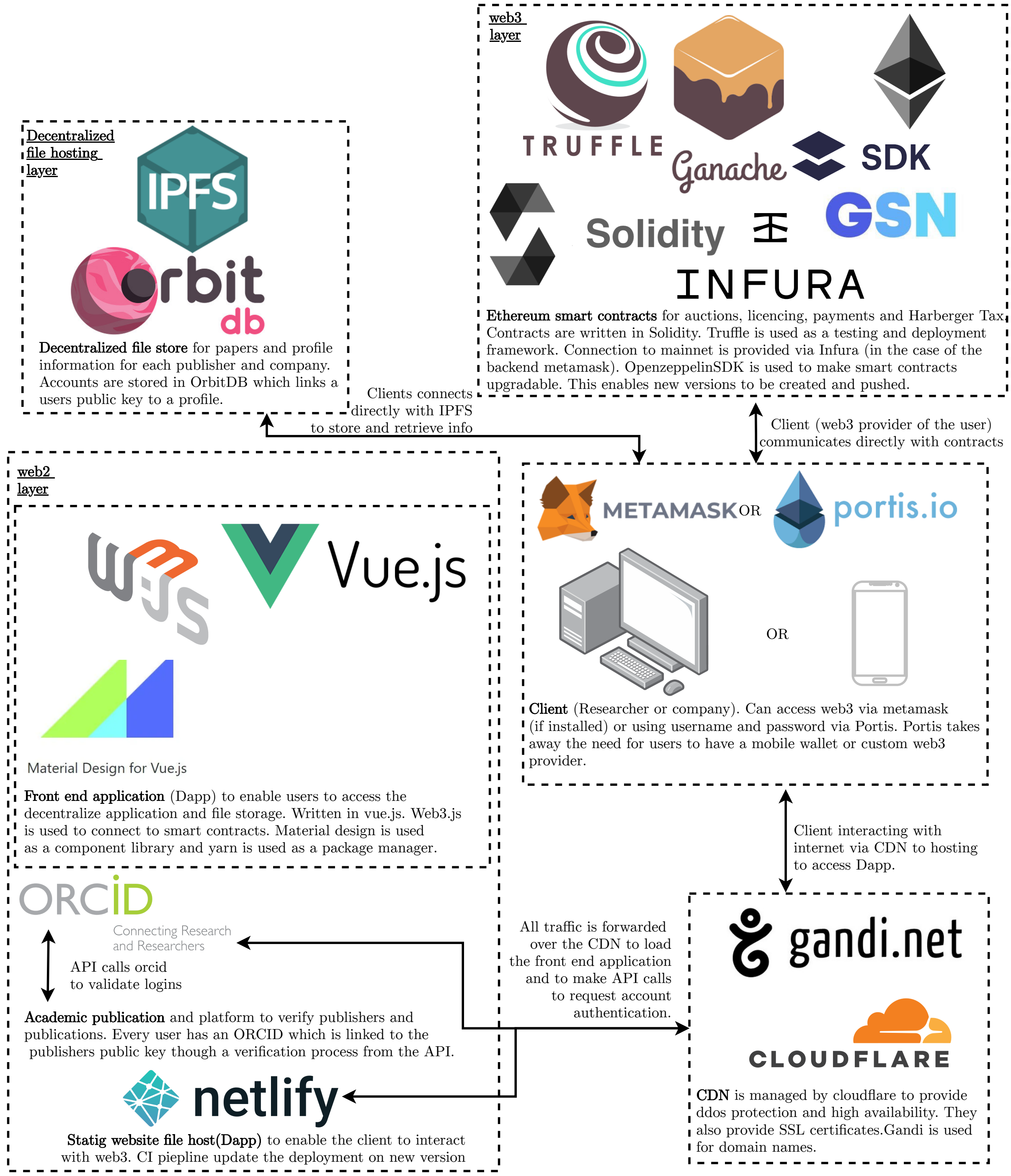


Figure 6.1: Technology stack overview showing all the interconnected components used to create the marketplace. Each dashed rectangle represents a component within the application.

This Figure contains two other blocks which are not "layers" of the application: the client and the content delivery network (CDN). The client is the users browser on either a computer or mobile device and the CDN is a networking stack to provide access to the application over the internet. Both are discussed in the sections that follow.

6.2 Smart contracts

The smart contracts were designed to be simple, extensible and modular. To achieve these properties a *contract oriented design* approach was followed which considers a strong focus on separation of concerns. This is similar in principle to *object oriented programming* from other domains. In this paradigm a large complex problem is broken down into multiple smaller objects (or smart contracts) which have scoped domains of responsibility. These contracts are interconnected to create a larger, more complex system.

A *hub and spoke pattern* was used to define the interconnection of contracts wherein a central controller (hub) connects to external logic and data units (spokes). This pattern enables all external connections to be directed to one central hub which then delegates execution flow to a specific spoke based on the context. This makes the overall system easier to understand as one can break the conceptual functions of the system into smaller, isolated chunks. Integration from external systems also becomes easier as there is only one integration point: the hub. This pattern enables extensibility as new functionality can be added by adding new spokes.

The implementation consists of seven separate smart contracts. The hub and spoke design along with all contract names and core functionality can be seen in Figure 6.2 below. The relationship between the spokes and the base hub is a 1..1 relationship on both sides because there is only ever one spoke of each kind for a hub.

This Figure uses the class Figure syntax. A filled diamond indicates composition ("has a" relationship). Eg the UnicoinRegistry has a Vault. The 1..1 syntax indicates the composition relationship between two contracts. For example the 1..1 between the UnicoinRegistry and the Vault means that the UnicoinRegistry has exactly one Vault. An open arrowhead with a solid line indicates inheritance (an "is a" relationship). For example the LicenceManger is a ERC721. External libraries and their connections are also shown, such as the connection to the ERC20 Dai stablecoin from the vault. This is a <<use>> relationship, indicating that the vault uses the ERC20 token.

The caller of the UnicoinRegistry can be any externally owned account such as a user, trader or publisher.[1] It can also be a smart contract if someone has created an integration into the marketplace for ideas. Each contract is now briefly discussed:

1. UnicoinRegistry is the central hub of the hub and spoke. All contract calls that happen to the marketplace for ideas are sent to this contract and then forwarded to the spokes.

2. UserManager controls all user account related information and logic. This spoke stores account details and authenticates user calls to check if a user is registered and the account type of callers.

3. Vault is responsible for all token payments in the system. It is the only interface that

[1] Externally owned account (EOA) is an Ethereum account that has a private key managed by a user. This includes all accounts that are not smart contracts. For example Metamask is an EOA.

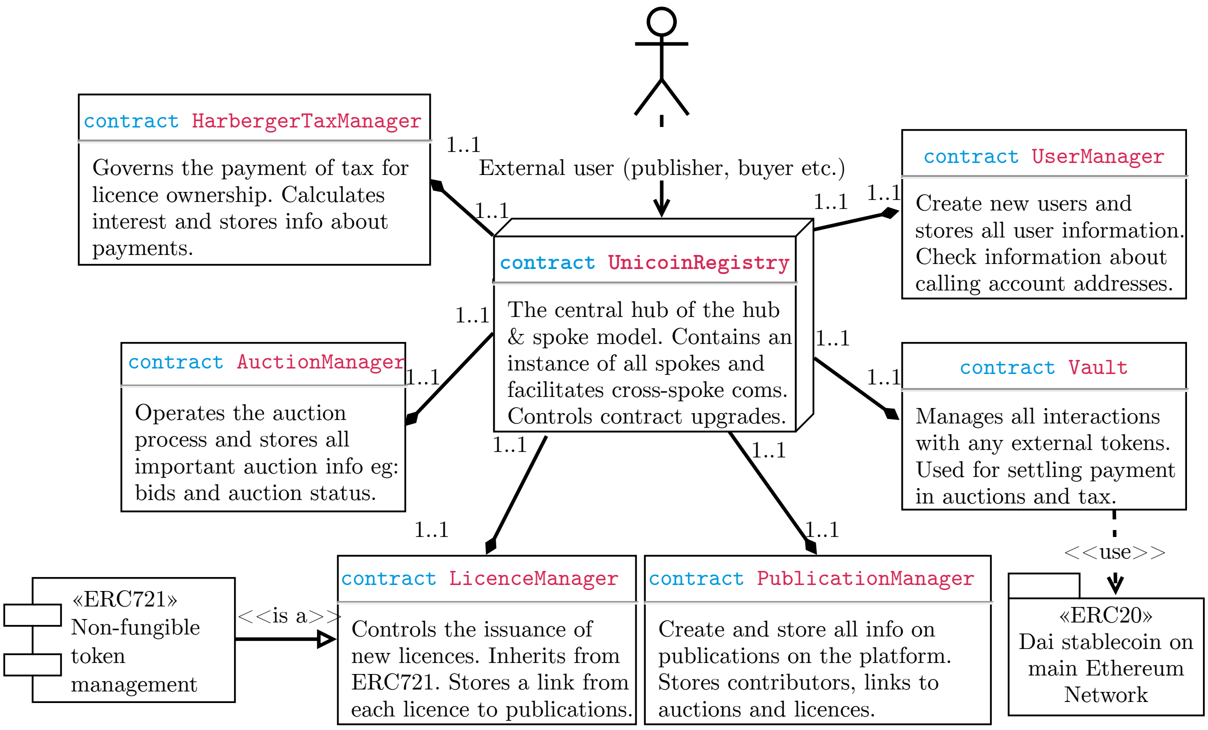


Figure 6.2: Hub and spoke design pattern. All user calls are made to the UnicoinRegistry(hub) which then delegates to different domains of responsibility. When multiple domains are required in a function call the registry takes care of cross-spoke communication. External libraries are indicated as well as external contracts such as the ERC721 and ERC20 contracts.

communicates with the Dai ERC20 token. It handles payments to an individual or bulk payments to a collection of users. In order for the vault to transfer tokens on the users behalf the user must first approve the vault. This is in accordance with section 5.2.4. This is done by the user calling the `approve` function on the Dai ERC20 contract and specifies the `vault` contract address. Once this is done the vault can send user tokens with the `transferFrom` function and automatically perform payments.

4. `PublicationManager` stores all information about publications, including pricing configurations and contributors.

5. `LicenceManager` inherits from ERC721 and provides all logic and storage to facilitate the creation of new licences. This inheritance means that the licence manager gains the interfaces of a ERC721 token. To implement this the Openzeppelin Solidity library was used. This provides a collection of audited contract definitions making building on top of current standards more streamlined and minimizes risk. The standard ERC721 interface enables the transfer of tokens. This has been disabled in this implementation by overriding the `transfer` and `transferFrom` functions in accordance with section 5.2.3.

6. `AuctionManager` operates the sealed bid auction system. This involves enabling users to 1) submit hashed bid commitments 2) reveal and validate bids 3) finalize auctions. As discussed in 5.2.1, revealing bids involves hashing the provided `bidAmount` with a `salt`. This is done using the `keccak256` function which is Solidity's implementation of `SHA-3`.

7. `Harberger TaxManager` implements and enforces Harberger Tax on licences, if the publisher chose to enable this for their listing. This includes the logic for calculating outstanding tax on an issued licence. This is calculated as discussed in section 5.2.2. Solidity is quite restricted in the numerical operations that can be performed. To calculate the outstanding tax an exponent is taken as $O = p(e^{r\tau} - 1)$. However solidity does not have any notion of the natural number e and can't take exponents with fractions. To overcome this fundamental limitation in Solidity the `optimalExp` function was taken from the Bancor bonding curve implementation.[2] This function enables arbitrary natural exponentiation in Solidity using a set of precomputed constants.

6.2.1 Smart contract data structure

Each spoke contract stores data about its respective domain of responsibility. For example the `UserManager` stores information about all users within the system. All spoke data follows the same pattern: an array of struct where the struct stores structured data for the spoke. An example of this structure can be seen in Figure 6.3. The internals of the struct are defined to encompass all information needed for that specific entity. For example Figure 6.3 shows the data required to create a `user`. Similar structs are defined for other entities.

```
//Struct to define the user object
struct User {
  address owned_address;
  string profile_uri;
}
//Array of registered users
User[] public users;
```

Figure 6.3: Data structure used to store a user. Each data structure is an array of structs which impose strict structure on the data stored. Array index represents each entity's ID.

Each instance of the struct within the array has a unique array index. This index acts as an entity identifier (ID) and can be thought of as a unique "database" identifer for that specific entity. In Figure 6.3 for example the array of `users` defines each `User` a unique `ID` as the index of the `User` within the array `users`. This use of an index `ID` as an entity ID enables interconnected data structures to be formed between different spoke contract data structures. For example an issued licence could have an entity ID of 4. A publication can "hold" this licence by storing this entity ID within an array.

Figure 6.4 shows all data objects and structures within the smart contracts. Each object represents an array of struct of that specific type. The relationships between different entities are also shown. Here an open diamond indicates an optional association. For example a `User` *can have* a `Publication`. A closed diamond shows a required relationship. For example a `Bid` *must have* an `Auction`. This Figure also shows the numerical relationships between different entities. For example an `Auction` has `0..* Bids`, meaning

[2] Bancor implemented bonding curves in Solidity. This requires integral calculus to be defined within a smart contract. Solidity does not contain enough mathematical operations to perform this on chain so Bancor implemented a number of useful functions to overcome this which they published online.

that an auction can have at least zero bids and at most an undefined upper bound of bids. However, a `Bid` has `1..1 Auctions` meaning that for every `Bid` there must be exactly one `Auction`; a bid can't exist without an auction and can't be part of multiple auctions.

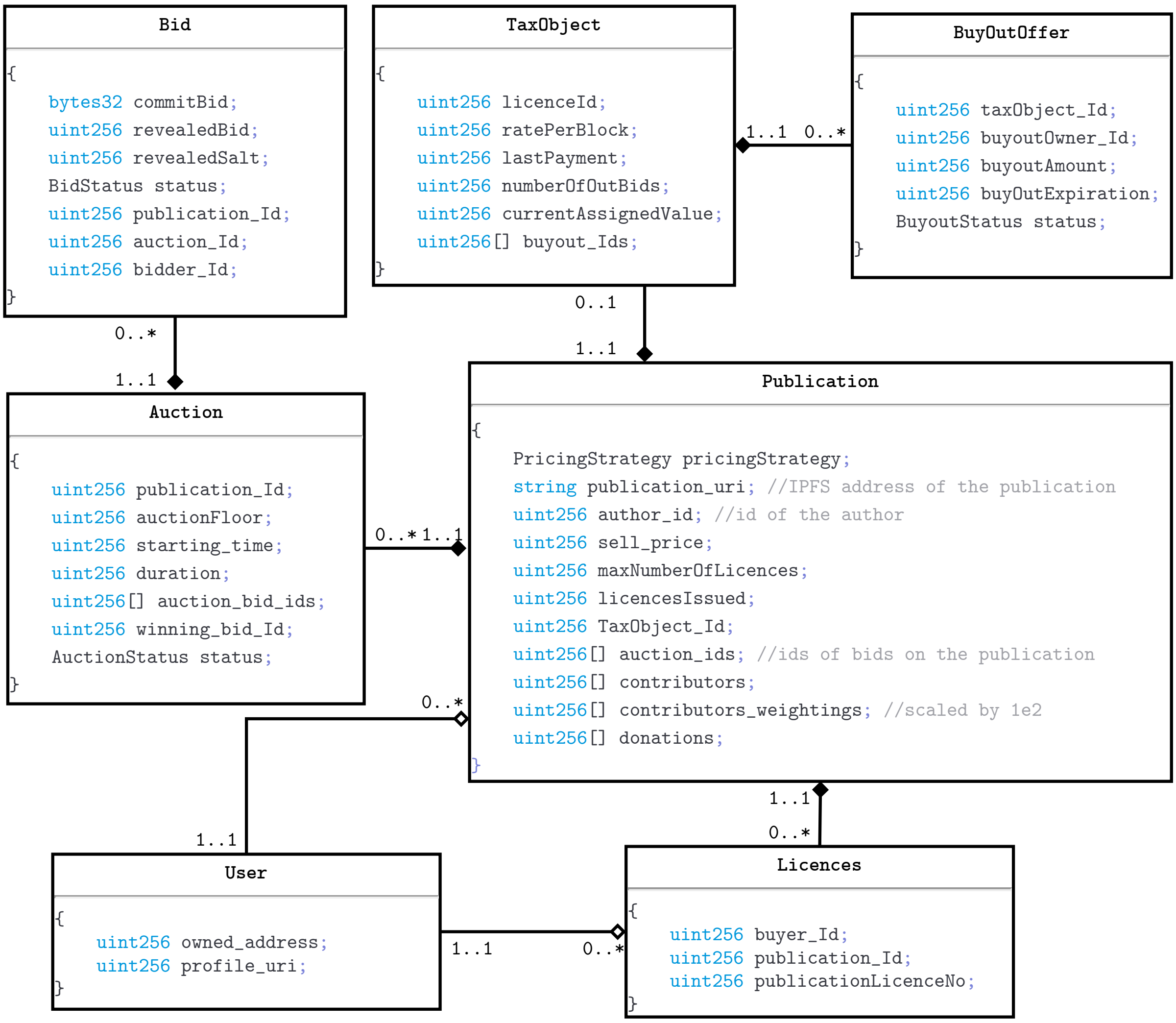


Figure 6.4: Contract data structure. Data is stored in arrays of each type. This is linked between different objects via IDs. For example, there is an array of Bid called bids which assigns an array ID to each bid. These IDs are stored within an array for each auction.

6.2.2 Smart contract functionality

As has been discussed, all function calls originate at the hub of the system, the `UnicoinRegistry`. From here functions within the spokes are called. The hub has no state or logic other than storing the addresses of the spokes and logic on how to connect the spokes together. As a result all implementation state and logic is defined within the spokes.

The core functionality of the marketplace can be broken down into nine main functions. Each of these functions are briefly discussed below. Each function has a full sequence Figure as well as additional discussion in Appendix F for brevity.

1. `registerUser` creates a new user account. The `_profile_uri` is specified which is a unique fingerprint created for each user when they register. Registration involves storing profile information on IPFS. Upon adding the profile to IPFS a URI is automatically created. See section 6.4 for IFPS details. Appendix Figure F.1.

2. `createPublication` creates a new publication object in the marketplace. All key information about the publication is specified such as the pricing strategy chosen by the seller (auction or fixed rate), auction details like duration and start time and publication contributor's divisions. Appendix Figure F.2.

3. `commitSealedBid` adds a `bidHash` commitment to a specific publication for a specific auction. This is the hashed concatenation of the `bidAmount` and the `salt`. The bidder will store their bid and hash until the reveal phase. This `bidHash` is generated by the user in their web browser before calling this function within the smart contract. Appendix Figure F.3.

4. `revealSealedBid` is used by a bidder to reveal their `bidAmount` and `salt`. The values specified are checked against the `bidHash` that was committed in (3). If the bid is valid it is marked as such and is thus eligible to win the auction. The bid validation process is in accordance with section 5.2.1. Appendix Figure F.4.

5. `finalizeAuction` loops through all valid bids and identifies the winner of the auction. At the end, bulk payment is settled between the winner of the auction and all contributors. This involves the transfer of Dai to a number of wallets, in accordance with section 5.2.4. Appendix Figure F.5.

6. `claimHarberger Tax` enables the licence issuer to claim outstanding tax on a specific publication. All tax from the previous payment accrued to the time of the call is paid out, in accordance with section 5.2.2. Appendix Figure F.7.

7. `updateLicenceHarbergerValuation` enables the current owner to update their valuation of the licence. The value that they define here will be the amount that they are taxed. One would update their value if they believe the value has dropped and thus they want to pay less tax or if someone has created a buyout request and they want to raise their valuation above the buyout request. Appendix Figure F.6.

8. `createHarbergerBuyOut` creates a bid to buyout a licence based on the current Harberger Tax allocated price. The buyout offer must exceed the current valuation by at least 5% or is rejected, as outlined in section 5.2.2. Appendix Figure F.8.

9. `finalizeBuyOutOffer` closes out a buyout request. This either results in the licence being revoked from the previous owner and transferred to the user that created the buyout or is rejected if the offer is below the threshold. Appendix Figure F.9.

6.2.3 Smart contract life cycle

Now that the core functions of the smart contract system have been introduced, a number of usage life cycles can be discussed. This showcases the main functionality of the different user groups interacting with the smart contract system. These figures show execution steps which effectively models the contracts of the system as a state machine. The key for the Figures are self explanatory and a detailed version can be found in appendix E.

6.2.3.1 Publication listing and auction life cycle

Figure 6.5 shows the execution flow for creating a publication and subsequent bidding. This life cycle flow mimics the principle Tesla and Dalhaousie example introduced earlier. Here a competitor Toyota is also bidding in the auction for a licence. Tesla wins the auction, pays the licence fee in Dai which is distributed amongst the contributors, and the non-fungible licence is created.

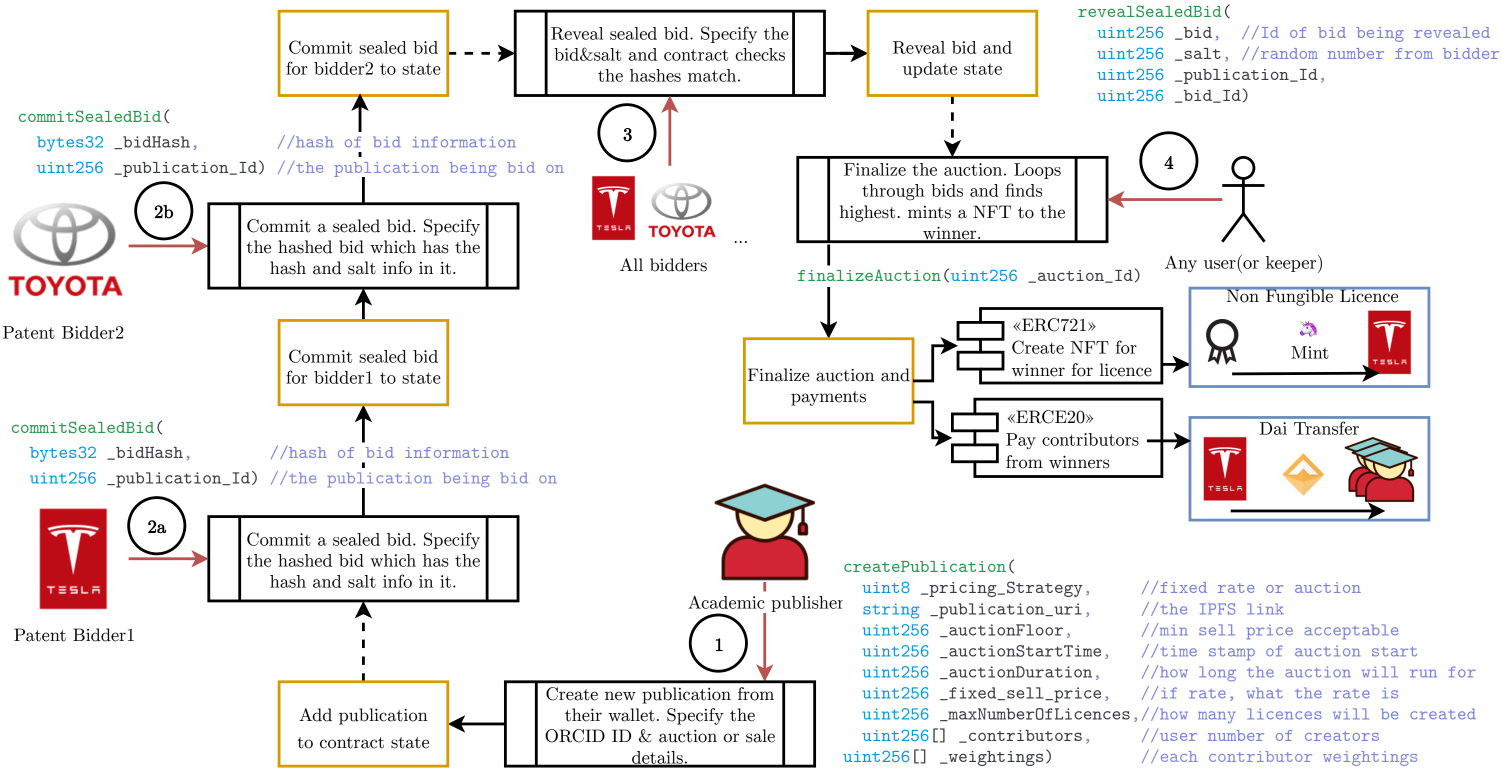


Figure 6.5: Publication and bidding life cycle which involves four main steps: 1) An academic lists a publication, 2) two companies bid in the auction, 3) the companies reveal their bids, 4) auction finalization, licence allocation and the associated token transfers.

This Figure introduces a new user that has not been discussed up until now: a *keeper*. This user is calling the `finalizeAuction` function in step 4. This function loops over all valid bids and identifies the winner. This functionality does not require any special permissions; in fact it can be called by a user that is not even registered on the platform. This is different to other functions that require registration and user accounts. For example `createPublication` requires that the caller is a registered academic (has created an account with the `createAccount` function). In contrast, the finalization of an auction does not require any special checks. As a result, a third party provider can call this function such that no bidder or publisher needs to return online to finalize the auction. This improves the user experience and flow for bidders and publishers by abstracting away this step for them. Thus this should be called by the operator of the marketplace.

At the conclusion of step 4 there is a Dai token transfer from Tesla to the contributors. This is facilitated by the permissioned payments introduced in section 5.2.4. This transfer occurs by calling the `transferFrom` function within the Dai ERC20 token which transfers tokens from the auction winner to the respective contributors. This is done iteratively for each contributor as defined in the previous chapter.

6.2.3.2 Claiming Harberger Tax life cycle

Figure 6.6 shows the claiming Harberger Tax life cycle. This has two possible outcomes after Harberger Tax is claimed:

1. The current holder of the licence is solvent which results in a payment to the contributors. This payment is similar process to that executed after an auction.
2. The licence holder is insolvent which results in the revocation of the licence and the relisting of the licence in a new auction within the marketplace. After relisting another bidder (Toyota), submits bids and wins the auction for the licence.

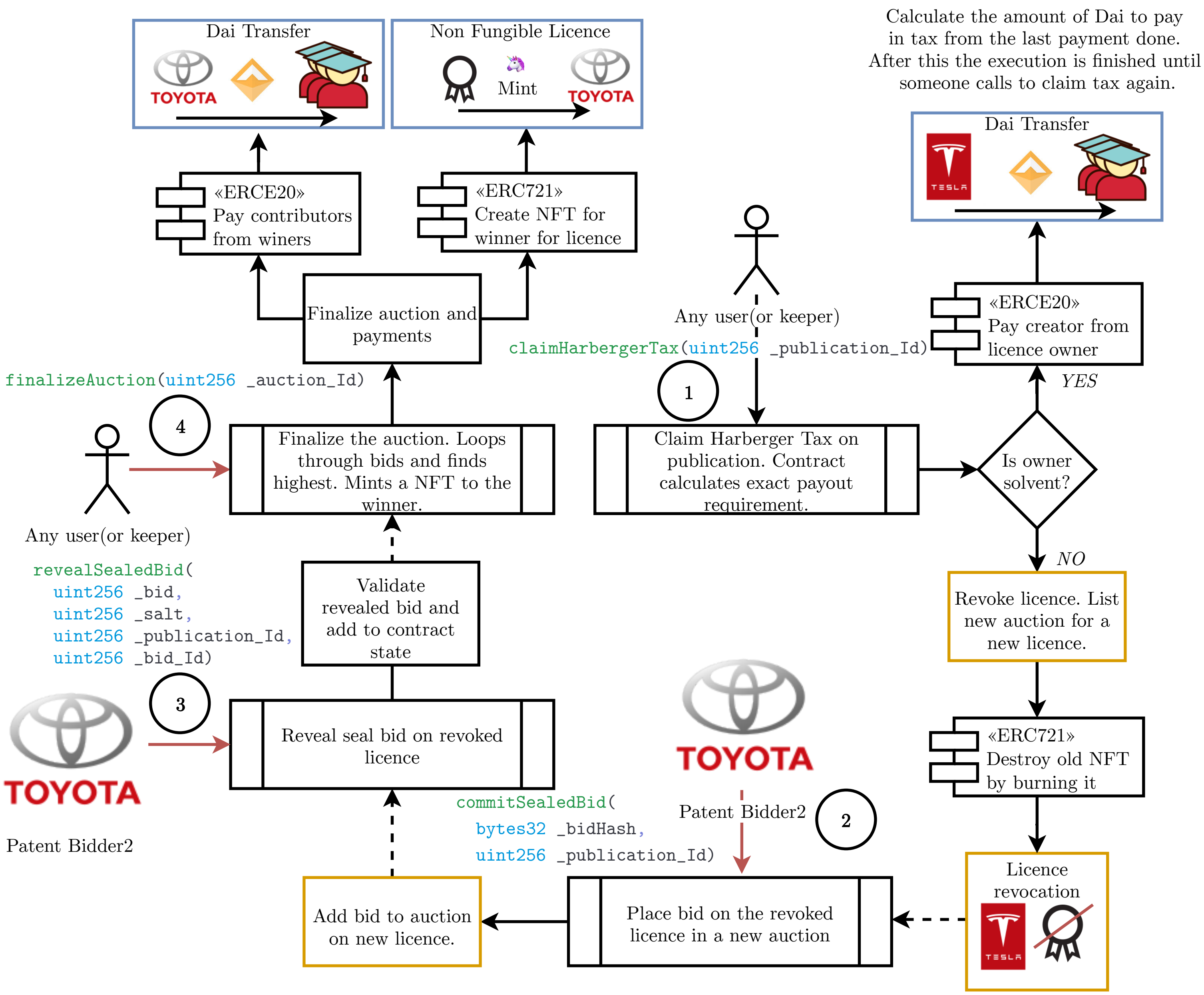


Figure 6.6: Process of claiming Harberger Tax. This results in the licence owner paying the content creator in the case of solvency or licence revocation if the holder is insolvent. In the case of the latter a new patent auction is created. Steps 2, 3 and 4 show a new patent buyer bidding and winning in the resultant auction that follows the revocation.

6.2.3.3 Harberger Tax buyout life cycle

Figure 6.7 shows the Harberger Tax buyout life cycle. This flow occurs when a bidder tries to buy a licence from the current licence holder. This consists of three main steps:

1. The bidder creates a buyout request specifying the amount they are willing to pay.
2. The current licence holder can choose to update their licence valuation (or leave it).
3. If they increase it above the buyout request then the buyout will fail. Else if they do not update it or update it to less than the threshold then the licence is transferred away from them and to the bidder. In this Figure this is shown as the NFT transferring from Tesla to Toyota. To pay for the buyout, a token transfer then occurs from the user that submitted the buyout to the previous owner and contributors.

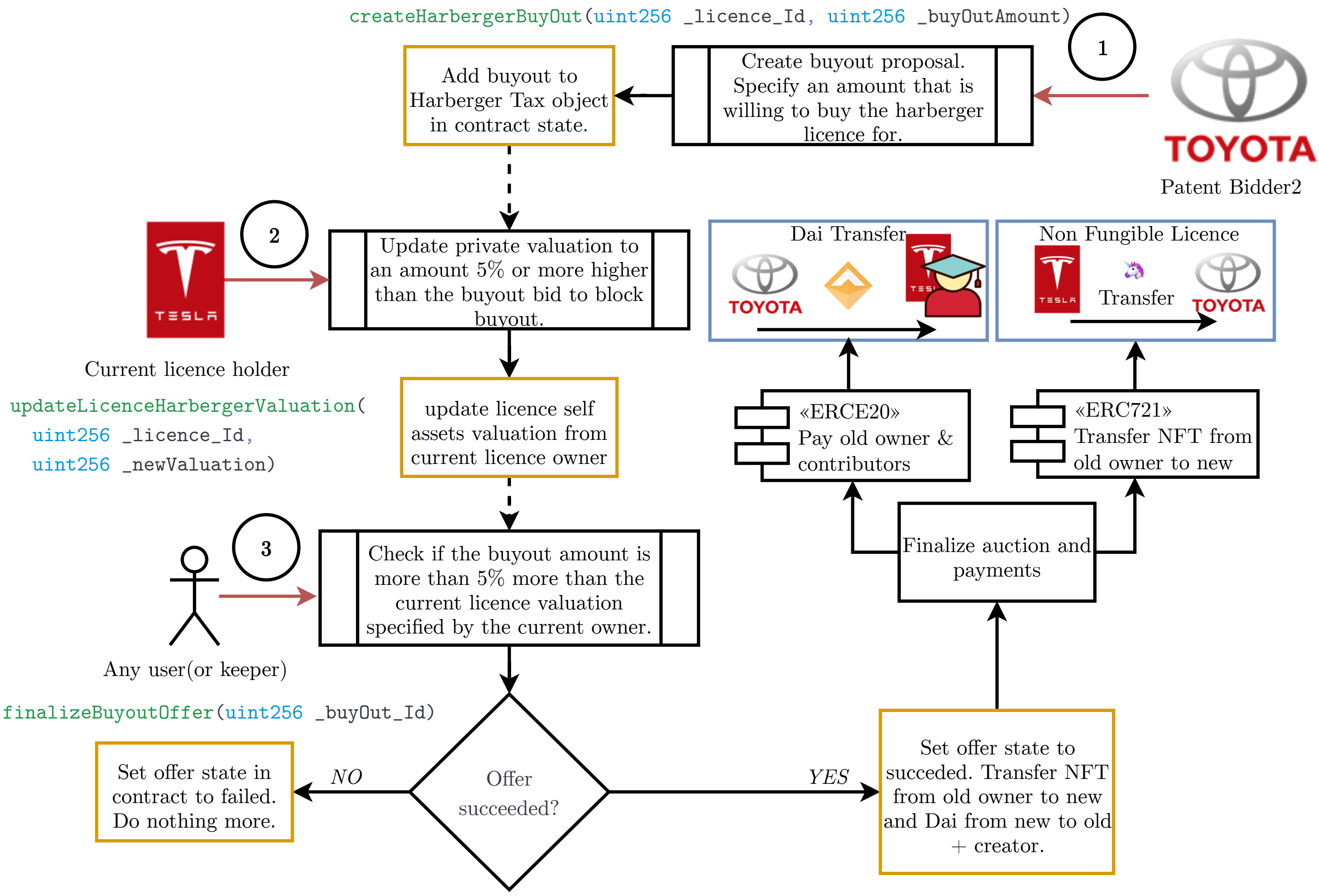


Figure 6.7: Buyout process where Bidder2 creates a buyout offer. Bidder1 can choose to outbid this offer by raising it a minimum of 5% in the time frame allocated. If Bidder1 does, then the offer is not successful and the execution ends. If Bidder1 does not raise the bid, then the licence is revoked and transferred to Bidder2. As a result of the purchase Bidder2 pays Dai to Bidder1 in exchange for the licence.

These life cycle shows a key advantage of smart contracts: the automatic *"if this then that"* of complex financial systems. This is in the form of a NFT licence coupled with automatic payments. Upon the revocation of the licence the NFT is revoked which removes it from circulation. Following this a new auction is created for the re-sale of this licence. This automation decreases the administrative overhead by the platform operator. At the same time the deterministic nature of the code execution and the public verification thereof removes any opportunity for mistakes on the part of the platform operator.

Next, additional system design considerations are discussed.

6.2.4 Smart contract upgradability pattern

Smart contracts are by design immutable. However, modern software development relies on the ability to upgrade and patch source code to produce iterative releases. Blockchain systems rely on their immutabile design as a core property but some degree of mutability is needed for bug fixing and product improvement.

The smart contracts created in the market are designed using a *proxy pattern*, utilizing an unstructured storage mechanism to provide smart contract upgradability. This is achieved in part by utilizing the OpenZeppelin SDK which provides wrappers for this functionality.

Upgradability using a proxy pattern works by directing all traffic to a proxy contract which wraps the logic and functionality of the implementation contract. The proxy directs all calls to the implementation contract which contains all contract logic. State, on the other hand, is stored within the proxy. The key component is that the implementation contract logic can be updated but the proxy access point is never changed. This ensures that the state is kept during upgrades wherein the system logic can be swapped out without loosing any system data. The system remains immutabile in the sense that the code or state can't be changed but the implementation logic can be swapped to another contract, thereby updating it. The proxy access point will direct transactions to a different (newer) implementation and in doing so, the software will have been upgraded. This is seen in the Figure 6.8.

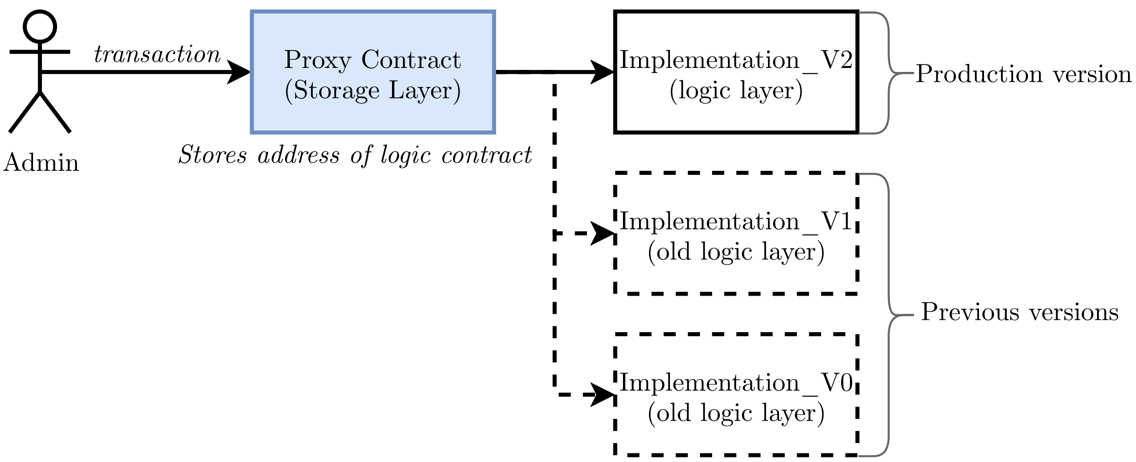


Figure 6.8: Upgradable smart contract pattern. Each smart contract deployed (including all hubs and spokes) use this pattern. All traffic is forwarded from a proxy to an implementation contract. The implementation contract can be updated between different versions which keeps state such as token balances, NFTs issued or running auctions as these are stored in the proxy contract.

OpenZeppelin SDK is used to automatically create proxy forwarding contracts which act to expose the full interface of the child contracts which are thus upgradable. This is done using a fallback function, written in Solidity assembly. This mechanism acts to forward all calls and state from the proxy to the implementation contracts such that from the user's perspective the proxy appears to be the implementation contract.

Ultimately this mechanism enables modular upgradability of each component of the hub and spoke. This is useful because upgrades can be performed on one specific smart spoke contract without having to effect the rest of the smart contracts. For example, say the way that auctions are run is updated. The new logic is deployed as a new implementation of `AuctionManager`, the auction state is inherited from the previous version due to unstructured storage and the rest of the system remains functional without need for update due to the proxy contract. After the upgrade the whole system continues to operate but with the new section of logic for the `AuctionManager`.

Upgradability is performed by deploying a new implementation contract version then updating the proxy to forward traffic to the new version. Because the proxy keeps its address eternally the collection of smart contracts appear to not change from a callers perspective, despite the internal logic upgrade. This can be seen in Figure 6.9 below.

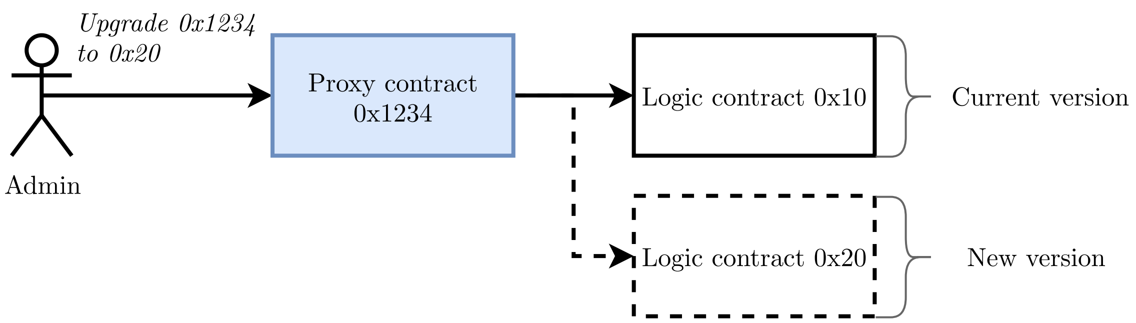


Figure 6.9: Upgrade process. A new implementation version is first deployed and then the proxy is instructed to forward all contract calls to the newly deployed contract. The proxy then stores this new address and uses it until the admin updates the proxy again.

6.2.5 Meta transactions

Smart contracts that modify contract state cost gas. Gas is a payment in Ether made by the caller of the function to pay for the state update. For example performing a transaction that transfers ERC20 tokens costs gas. The cost is a function of the gas used and the gas price. The gas used is dependent on the underlying op codes executed within the EVM during the function call. The gas price is dependent on current demand on the Ethereum network.

The need to pay gas for transactions significantly degrades the usability of blockchain applications. A user needs to have set up a wallet and deposited Ether before they can engage in a meaningful way with the application. In the case of an academic, for example, they would need to have bought Ether from a crypto exchange and deposited this into their wallet before creating a publication. This restriction drastically reduces the number of users that could actually use the platform.

A solution to this poor user experience (UX) is to abstract the wallet that pays for gas away from the wallet that submits the transaction using *meta transactions*. Simply put, get someone else to pay the gas for a transaction. In the case of an academic publishing a listing in the market, the market operator could pay for their transactions on their behalf. Meta transactions have far reaching consequences in the UX of blockchain applications: a new user no longer needs to acquire cryptocurrency to use a blockchain application.

An important detail here is that the user who submits the transaction should still remain the owner of their private keys. Ergo they will submit a signed transaction to someone else (relayer) who will then execute it on their behalf. Because the user signed the transaction the relayer paying for the gas cannot manipulate the contents of the transaction. This makes meta transactions trustless. Figure 6.10 shows how meta transactions work.

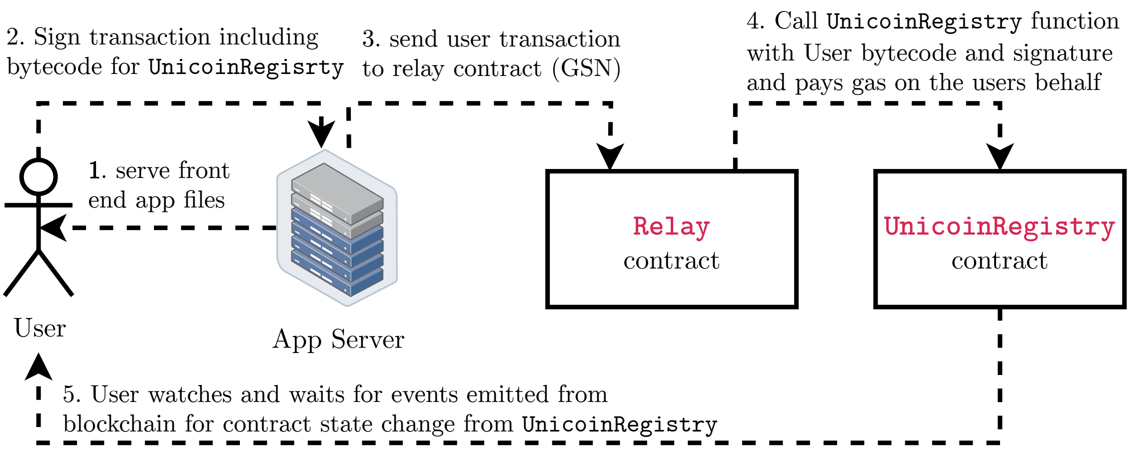


Figure 6.10: Meta transaction flow. Submitted transaction gas paid for by a relayer on the users behalf thereby abstracting away the need for the user to have Ether.

The worst "attack" a malicious relayer can do is withhold a transaction from the blockchain. This can be solved by creating a permissionless network of relayers who can all transmit transactions and compete for submission fees. The relayer network chosen for this thesis is the Gas Station Network (GSN). GSN provides a peer to peer network of relayers who push transactions to the Ethereum blockchain in exchange for a fee.

Integration with the GSN involves inheriting from a library that has all the logic defined to connect with the network. This library can be integrated directly into the Open-Zeppelin upgradability SDK through the GSN provider. Secondly, define the function `acceptRelayedCall` to indicate whose fees should be paid for. For this thesis' implementation of this function will pay the gas for all wallets that call functions within the `UnicoinRegistry`. More sophisticated curation of callers is possible in future versions and would ensure that only registered users will have their gas sponsored.

6.2.6 Non-custodial wallets

A key element of building decentralized, trust minimized applications is private key management. Who owns and controls user private keys is important when considering how centralized a solution is. This is ultimately affects the points of failure within a system. For example, a traditional crypto currency exchange stores its users private keys on their behalf. In this setup users don't "own" their crypto; rather they have an account with a company that promises to grant access to their deposits. This creates a central point of failure for attackers to target.

An alternative design principle is to build systems wherein the user is always in control of their private keys; a decentralized, trustless system. This drastically reduces the risk of a hack as a compromised key will only affect one user, rather than the whole application. It is also more aligned with the ideology of decentralized trust minimized applications which aim to put the user in control of their wallet, identity and data.

However there is a usability trade off in wanting users to manage their own private keys. In order to have a wallet to interact with an Ethereum application a user normally needs to install an extension into their browser, such as Metamask. This is intimidating and foreign for new users.

Portis offers a non-custodial solution to this UX challenge. They create a wallet which is stored in the local storage of the users browser which is encrypted with an email and password. This encrypted wallet is then uploaded to IPFS. When a user signs in to a new application or on a new device they download this encrypted wallet from IPFS and decrypt it in their new browser. They then have access to their wallet from the browser to sign transactions and access their funds. This whole process is automated for the user and injected into the browser from the application. They simply need to type in an email address and password and their wallet will be downloaded and unlocked on their behalf.

This drastically improves the UX as users no longer need to install a custom extension and are presented with a login flow that they are already familiar with. At the same time this mechanism keeps the user in total control of her account thereby making onboarding easier.

6.2.7 Dai to fiat on and off ramping

As has been discussed, all payments done on the platform for auctions and Harberger Tax are processed using the Dai stablecoin. One consideration is how users within the system will convert from their native fiat currency into Dai. For example, how does Tesla convert from USD into Dai and then how does the University of Dalhousie spend this Dai either in USD or another currency (they are a Canadian university so they might like to spend it in Canadian Dollars).

Conversion from fiat currencies to cryptocurrencies and back is a common problem for many blockchain applications. Making this process as easy and approachable for users as possible is important in achieving any kind of adoption. There are a number of solutions that have entered the blockchain ecosystem in the last few years including Wyre, Limepay and Moonpay who all provide approachable APIs enabling user application developers to integrate fiat to crypto onramps directly into their applications.

Wyre was selected as the payment onramp for the marketplace as they provide low fees, simple API and excellent documentation. It also offers the lowest fees out of the three which makes it more economical. As with any external integration, the provided code base is modular enabling easy exchange of this component to another API if desired.

6.2.8 Upper bound on gas usage considerations

The Ethereum Virtual Machine (EVM) is Turing complete, assuming execution remains within the block gas limit. The block gas limit is the maximum computation that can be performed in one single block of the EVM. Any computations that exceed this are reverted. This means that certain computations are not possible to be performed within the EVM as they will exceed the gas limit.

This is relevant in the marketplace when specific high gas operations are performed, such as transferring tokens. As a result, there is an upper bound on how many token transfers can be safely performed within one block. For example at the conclusion of an auction (or when paying Harberger Tax) each contributor receives a token transfer from the licence holder. This requires gas to be spent for each transaction which in turn limits the number of transactions that can be placed within each block. If there are too many transactions attempted within a block then the function execution will exceed the gas limit and the whole transaction will revert. This is commonly referred to as a gas limit denial of service and is susceptible to contracts with unbounded operations. As a result, the function will never be able to be executed which means that no contributor will get paid and funds will be locked within the contract. This will put the system in a broken, irrecoverable state requiring an upgrade to fix. There are two possible solutions to this problem:

1. Limit the number of contributors that can be added to a publication such that the number added is always less than the number of transactions that can be performed in one block.
2. Split token payments over multiple blocks so no single block exceeds the gas limit.

Option 1 is easier to implement and does not impose overly onerous restrictions on the system as one block can still facilitate upwards of 65 transactions. The upper bound for the number of contributors is therefore set to 50 to ensure consistent operation. Future implementations can expand on this through splitting transactions over multiple blocks.

6.2.9 Security and testing

Testing is an important part of building any kind of software system. In the case of blockchain applications, testing is vital to ensure that the smart contract behaves as expected. These systems move around value and if programmable money has a bug, money can be lost or stolen. Blockchain's immutability makes the cost of failure very high. If a contract is compromised there is no easy way to "roll back" the state to undo the hack.

The smart contracts were thoroughly tested using the Javascript framework Mocha and Chai. Truffle was used to deploy and interact with the smart contracts within the testing environment. A total of 62 tests were written with a code coverage of 93%. A full list of all the tests created and a coverage report can be seen in Appendix G.

6.2.10 Smart contract deployment

Before smart contracts can be used on the blockchain they need to be deployed. A migration script was created that deploys the smart contracts and then captures the deployment addresses so the front end application knows where the contracts are located on the blockchain (i.e the deployment addresses). The deployment process creates new instances of the contracts on either a local test blockchain or a public test network. A local test blockchain can be run with Ganache and acts as a high speed testing environment for rapid development iteration. A test network, like Kovan acts as a public, free test environment that mimics the main Ethereum blockchain for testing applications. As soon as multiple users will test an application it is appropriate to put it on a public test network. Additionally having the app on the test network means that one does not need to start up a local test network every time to interact with the application. Additional details on the the Kovan deployment addresses and how to recreate the deployment can be found on Github here.

This migration script is first deployed on the hub and then all spoke contracts. Each spoke contract needs to know the hub address before it can be deployed. This is because some functions within spokes require to only be called by the hub. For example, permissioned calls that modify system state should only originate from the hub, as this acts as the central point of the system. This hub address is specified in the `initialize` function for each spoke. This `initialize` function is similar to a constructor except it can be called separately from deployment. Once all spokes are deployed the hub is `initialize`d with all spoke addresses. More details and flow Figures on the deployment process can be seen in Appendix F.9.

6.2.11 Interface contracts

The last relevant design consideration was the use of interface contracts to define the interconnection between the hub and spoke contracts. In order for the hub to communicate with each spoke contract it requires to have an instance of said spoke contract instantiated within it. The normal process for doing this is to import all spoke code into the hub and then create an instance of it within the `initalize` function (or within the constructor, if you are using constructors). The hub can then call functions within the instance of each spoke.

This pattern however introduces bloat in the hub as importing all code of each spoke into the hub results in the hub containing unnecessary logic which is independent of its domain of responsibility (the hub should not need to know how the spokes work, only how to connect to them). A secondary issue can arise wherein the byte code of the hub can become so large that it can no longer be deployed.

A solution to these issues is to define an additional interface contract for each spoke contract which has all function definitions associated with the spoke, but no implementation logic. The hub contract then imports this interface contract and instantiates an instance of it at the address of each respective spoke implementation. Because this interface has the same function definitions as the implementation contract, it has the same Application Binary Interface (ABI) and so the hub can communicate with the spoke in the exact same way as it would if the hub had imported all implementation details of each spoke. Because the hub has only imported an interface and not the full contract definition the hub bloat can be avoided which drastically reduces the hub contract size as no unnecessary logic is included within the import.

6.3 Front end design and implementation

The next section examines the front end created for the marketplace. The front end is the portal that all users use to access the marketplace. It performs two primary functions:

1. Enable the user to interact with smart contracts. This is for all core platform functionality including issuing new publications, bidding on licences, creating buyout requests ect.
2. View blockchain information for the marketplace such as listings and bids.

The user interacts directly with smart contracts from their browser in the front end. The front end supports any injected web3 provider like Metamask or Portis. Portis is the default as this is what most non-crypto users will feel the most comfortable with.

6.3.1 Javascript framework

Vuejs was selected as the javascript framework of choice to provide a mechanism to interconnect the application front end components to the smart contracts and decentralized storage. For a full rational into why Vue was selected over other frameworks like Angular or React, see Appendix H.

While a javascript framework is not strictly required to build a reactive, modern website, it makes the underlying javascript complexities a lot simpler, such as state management, routing and separation of concerns through component based design. Additionally, the integration with third party libraries becomes easier. Moreover, the use of a framework makes synchronization between the state and the user interface possible. This synchronization is hard to achieve without the use of a framework. Lastly, the framework allows for the creation of complex, yet efficient and easy to maintain user interface that would not be realisable with vanilla javascript.

6.3.2 Javascript connection to Ethereum

Certain client interactions result in the submission of a transaction to the blockchain. For example, creating a user account, creating a publication or submitting a bid all require the user to submit a transaction. This is handled with the web3.js library. This library submits transactions from the user's wallet via their browser and enables direct interaction with smart contracts. Figure 6.11 show this life cycle from calling data to it being returned and rendered on the front end.

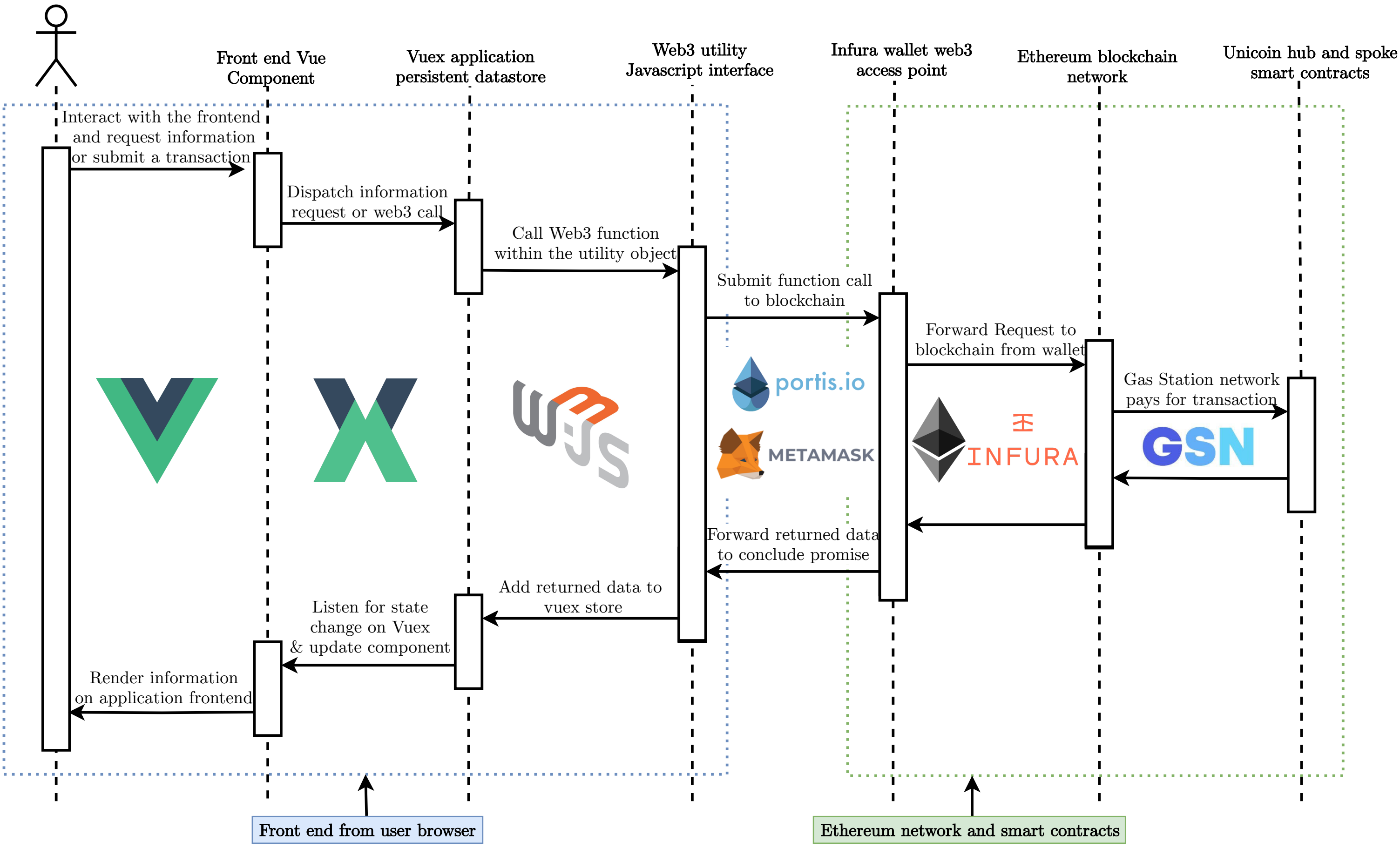


Figure 6.11: Data flow from the user front end to smart contracts and back. Call originates from user, is then processed by the store and submitted to the blockchain. From here the transaction is processed by the network before being returned to the user.

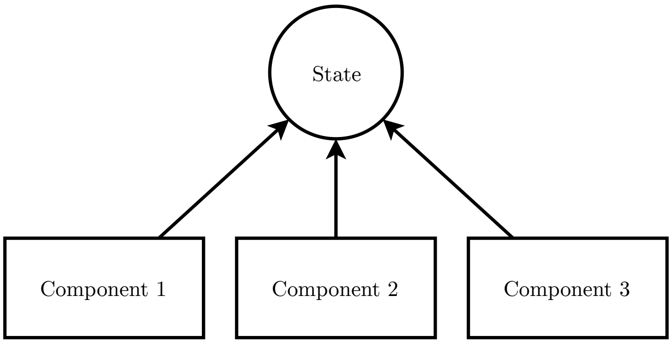
6.3.3 Component based design

In order to make the front end as modular as possible to facilitate upgradability, easy addition and modification, every element within the front end is a series of nested components. This means that the front end can programmatically render selective elements based on the users interaction. For example, if the user selects to create a new publication this specific information is rendered within the front end Document Object Model (DOM). Each component directly accesses the state. This state effectively acts as common storage for all front end component's rendered data. This can be seen in Figure 6.12a.

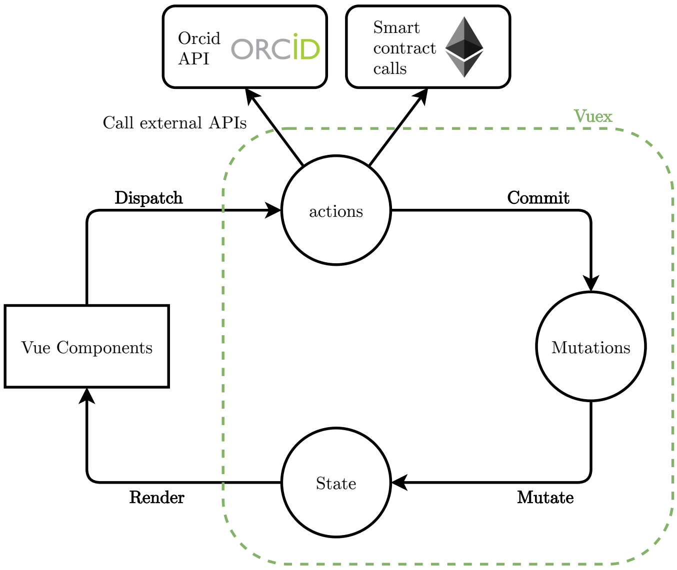
6.3.4 Javascript State Management

In order for the front end to display on-chain information it needs to read data from the blockchain, process it and then store it in the client's browser. To do this while ensuring modularity and upgradability, a Vuex state management pattern was embodied to act as a centralized store for all on-chain information in the client's web browser. This means that there is effectively a "database" that runs in the client's web browser's local storage which stores the data retrieved on-chain. State access and mutation rules are used to ensure that the state can only be mutated in a particular fashion. Resultingly, the mechanism used to retrieve information is decoupled from how information is displayed on the front end. Figure 6.12a shows how the store is reused between different components, returning specific information when required.

Figure 6.12b shows how Vuex state mutation operates within the front end application. A front end component will `Dispatch` an `Action` to request some data. For example, the user might want to view all publications listed within the marketplace. This action will in turn execute a call to the blockchain or external API based on the context. The `Commit` is then used to add data through a `Mutate` function to the `State`. The returned data is `Render`ed in the front end component and shown to the user.



(a) Components based design



(b) Front end state management

Figure 6.12: Front end component and state management. All components share the same global state. This can be seen as a local database that runs within the users web client. This state is updated in a predictable, reproducible and deterministic way. Any Vue component can dispatch an action to update the state. For example clicking ”create publication” will result in an action being dispatched from the Vuex store.

6.3.5 Login with ORCID

All academics who publish on the platform need to verify their identity to enable buyers to validate their claims of authenticity. This is achieved through integration with ORCID. ORCID is a non-profit organization that provides persistent digital alphanumeric codes to uniquely identify scientific and other academic authors and contributors (Orcid, 2019). ORCID provides an OAuth.[3] delegated login API which enables academics to login to the platform using their existing academic institution credentials. This is used specifically during the account creation process wherein an academic will login to their ORCID account which will return their unique identifier which is then stored in their account profile on IPFS. This IPFS account creation flow is shown in section 6.4. Screenshots for this login flow can be found in Appendix I.

When a potential bidder is performing due diligence on a publication they can view

[3]OAuth is a mechanism of delegated authentication. For example, login with Google will use Google’s authentication system to generate a login token for the application.

the publishers ORCID associated with their account. They can then view their ORCID profile and verify that they are indeed the publisher of the paper they claim to be. This verification works by ORCID linking to other publication services, like Arxiv.

6.3.6 Pending transaction steps

Blockchain based applications require consideration of additional user interface interactions for blockchain specific process, like pending transactions. For example when an academic creates a publication there is a multiple step process their web client will go through including: 1) uploading file to IPFS 2) submitting transaction to blockchain 3) listening for transaction mining confirmation 4) informing the user once the transaction has been mined. Each of these steps can (and should) be abstracted away from the user interacting with the application. However, some of these processes should provide feedback to the user as her client steps through these processes as they can take a while to complete. For example, the submitting of a transaction to Ethereum will take a minimum of 15 seconds and could be longer depending on the network congestion and gas price used by the user. To this end, each step of the interaction should inform the user with modals that update as the process progresses. Screenshots of this can be seen in Appendix I.

6.3.7 Front end devops, netops and application deployment

The VueJs front end application can be built to generate a set of static HTML, CSS and Javascript files. These files can be served from a web server to put the website online. One of the easiest and cheapest ways to achieve this is to use Netlify, a free static content hosting website. Netlify also provides built in continuous integration pipelines which will automatically deploy new content when developers push to Github. Once deployed, Netlify provides a semi-usable domain name for a project as `project-name.nelifty.com`.

To get a custom domain name, Gandi was used to register unicoin.win. The Domain Name Service (DNS) for Unicoin.win was configured to run using Cloudflare as well as a Content Delivery Network (CDN). This CDN increases system responsiveness and decreases page load times by hosting a copy of the website on a number of servers located around the world.

Figure 6.13 shows the networking interconnection to get the web application online. When a client loads the website, their traffic is routed via the cloudflare CDN to the Netlify server. Cloudflare caches a copy of the website to achieve high speed loading times. Importantly the users client interacts directly with the Ethereum blockchain via their web3 provider (Metamask or Portis). This is vital as this ensures that the application remains trustless in that the platform operator cannot interfere with the blockchain element if the user connects directly to the Ethereum network.

All traffic between the client, CDN and ultimately hosting on Netlify is encrypted using SSL. Certificates are issued by Cloudflare and installed within Netlify who serves them on page load. This enables the users web browser to validate the content loaded and prevents anyone online from sniffing the content sent to and from the server.

This design pattern is simple, highly performant and scalable. The CDN enables low

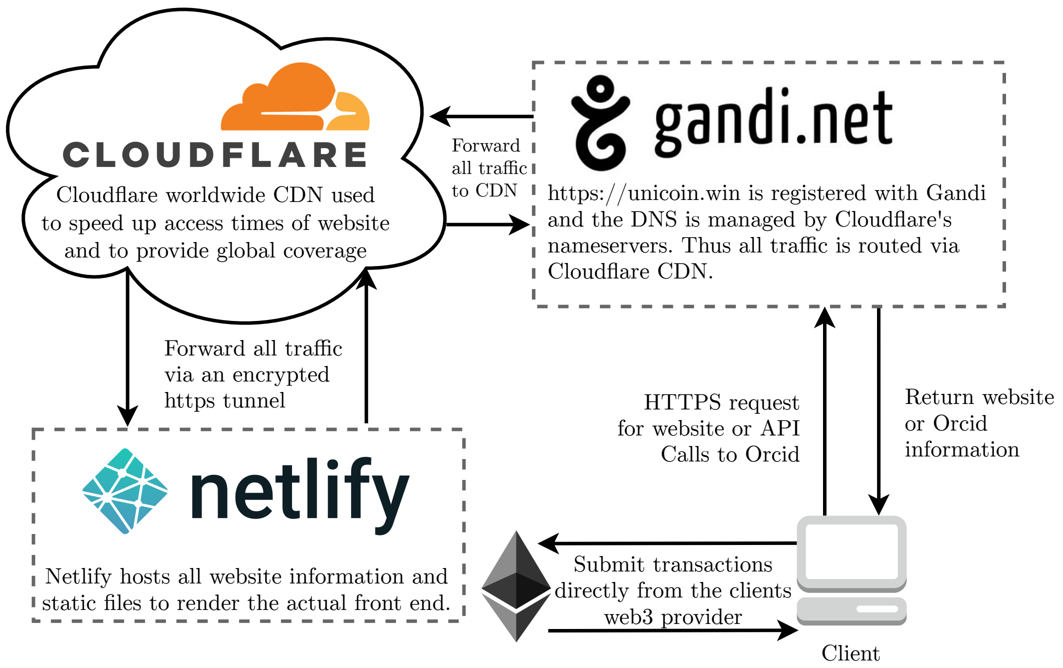


Figure 6.13: Network flow of application from web hosting via the CDN to the web client.

latency all around the world and Netlify's CI pipeline integration with Github makes rapid development possible with minimal setup overhead. However, this solution is inherently centralized. If an attacker was able to compromise any of the three services here (Netlify, Cloudflare or Gandi) they could inject malicious code which would then get served to the user when they load the website. This could be used to get the user to sign a malicious transaction in an attempt to steal their funds. Future improvements would involve the use of IPFS as a file hosting mechanism. While this is ideologically aligned, it suffers from poor developer experience and adds technical overhead to deployment and continuous integration operations.

6.4 Decentralized file storage

The last layer in the technology stack is the decentralized file storage. Specifically, the marketplace requires storage for:

1. Storing account profile information such as publisher name, institution and ORCID.
2. Storing papers published on the platform in the form of a PDF.

Neither of these should be stored directly on the blockchain as storing data on the blockchain is prohibitively expensive.[4] The easiest solution to this storage requirement would be to use a centralized file storage solution like Amazon Web (AWS) services or Google Cloud Compute (GCC). However, centralized solutions have a fundamentally different model which needs to be trusted when using the platform. All user and account data on a centralized system like AWS or GCC is under the jurisdiction of the company

[4]The blockchain is not a database and should not be treated as such. Storing 1GB of data on the Ethereum blockchain will cost $\approx$ \$ 500,000 based on the current gas cost. This is based off the yellow paper which reports a fee of 20k gas to store a 256 bit word. A kilobyte is thus 640k gas. Based off current gas costs, this is around 32,000 Ether for 1GB of storage.

that runs the service. This means that users can be censored, data leaks occur and privacy concerns are rampant.

An alternative approach, which is aligned with the ethos and ideology of a decentralized application, is decentralized file storage. The Interplanetary File Storage System (IPFS) is the most common way to attach decentralized file storage to a decentralized application. IPFS works by chunking up files and then storing them on a decentralized network. All files added to the network are uniquely identified by a hash of the file contents which enables validation of the file integrity when retrieving it later. Whenever someone downloads a file from IPFS they act to support the network by hosting it for others for a fixed period of time. Permanent storage on IPFS can be achieved by "pinning" the file, indicating to your IPFS client that you want to continue to host this file indefinitely.

The hash of a file added to IPFS acts as a Unique Resource Identifier (URI). This URI is stored within a smart contract to provide a reference to the file hosted on IPFS so that the information can be accessed later on. For example, consider the process of creating a user account within the marketplace for ideas. The process entails 1) creating the object 2) adding it to IPFS and 3) adding it to the Ethereum blockchain. This flow is shown in Figure 6.14.

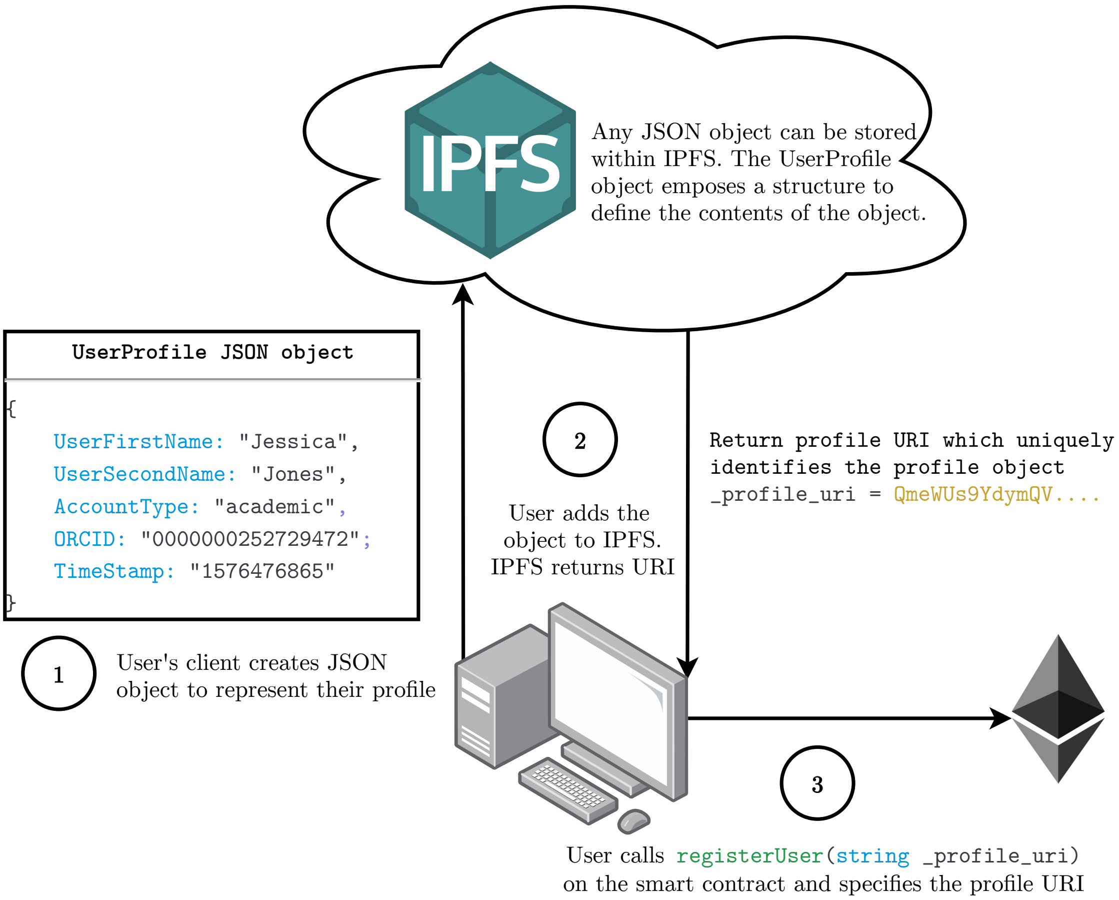


Figure 6.14: Decentralized file storage. The user first creates an account object which is then added to IPFS. The returned URI is then added to the smart contract when adding the user.

Later when retrieving profile information the URI is retrieved from the blockchain and then information is requested from IPFS.

PDFs of publications and patents are also stored on IPFS. The process to achieve this is very similar to that laid out above for creating a user profile. The PDF is read in by the users browser and converted into a data object. The data object is then parsed into JavaScript Object Notation (JSON) which is then stored on IPFS as a storage blob. Each file has a URI created for it which is stored within the smart contracts in the same manner as the user profiles.

Upon retrieving information about a publication or a user, the client will retrieve information from the blockchain, such as the object URI, and then get data from IPFS to display the profile.

To connect to IPFS an IPFS client and IPFS gateway are required. This was achieved through the use of ipfs.js as the client to run within the users browser and the Infura IPFS gateway which removes the need for the market to host it's own IPFS access point.

6.5 Chapter conclusion

This chapter outlined the technical implementation details of the marketplace for ideas. This discussion included all core components developed as well as the interconnection thereof. Ultimately this consists of three main application layers: 1) Smart contract layer to operate the core marketplace logic. These smart contracts are written in Solidity, use an upgradability pattern and leverage the gas station network to make transactions free for users. 2) A front end application to enable users to interact with the platform. This is written in Vue.js and leverages separation of concerns and Vuex state management. 3) Decentralized file storage to store user profiles and publications. This also incorporated an integration with Orcid for academic profile verification. Additional resources, screenshots and deployment information can be found in the Appendix and on Github.

The next chapter critically analyses the model and implementation presented. This includes a number of recommendations and future improvements.

Chapter 7

Market & implementation analysis

The marketplace for ideas has now been discussed at varying levels of detail. Next, the market and its implementation are critically analyzed. This analysis is broken into three main sections: 1) regulatory and legal, 2) market mechanisms and incentive design and 3) blockchain technical and implementation viability.

7.1 Regulatory and legal analysis

There are a number of regulatory obstacles that need to be overcome to make the proposed market a reality. Some of these are now explored.

7.1.1 Validity of IP claim through non-fungible licences

One of the core premises of the marketplace for ideas is that the non-fungible token represents permission from the content creator to monetize their work. This NFT acts as a "non-fungible licence", granting permission from the creator to the licence holder. For this to be a valid legal construction in the real world, it would need to be defensible in a court of law. This non-fungible licence, while cryptographically unique and guaranteed to originate from the content creator, is not presently a legal construct within *any* jurisdiction. It is therefore not trivial to use this as a valid claim of explicit consent granted from the licence issuer to the licence holder. As a result, bespoke legal contracts would need to be drafted and codified to make this construction legally permissible.

This problem is not unique to this thesis' implementation. Indeed, there are many projects that aim to link blockchain assets to real world defensible contracts. Projects like OpenLaw are leading the way in making legally defensible real world contractual agreements, backed by smart contracts. A system similar to this would be applicable in future versions of the marketplace for ideas.

7.1.2 Verification of publication authorship

A second premise of the marketplace is that only the author of an idea should be able to publish it within the market. This means that a random user should not be able to get someone else's paper, claim it as their own, publish it, and ultimately profit from it. The design presented enables a publisher to "prove" their identity through ORCID login which associates a user with their academic profile. Additionally, patent documentation which contains details around whom the patent was issued to can be uploaded . This enables potential bidders and other users to validate that a user is indeed who they say they are by corroborating their claim with their ORCID and submitted patent. However this mechanism is not bullet proof.

While this solution does help the problem of authorship validation, it is not full proof. For one, this mechanism assumes that the bidder will conduct sufficient due diligence when bidding based on the ORCID and the submitted patent. This leaves the onus on the bidder, rather than the market, to protect the user. This is not ideal as the bidder already has a high overhead of due diligence that they need to conduct on the value of the patent they are bidding for. Secondly, this mechanism requires a sufficiently strong legal framework that can legally prosecute individuals that break this assumption and steal others IP on the platform. This assumption is almost impossible to make in practice due to the exorbitant cost of international litigation. Future versions of the marketplace for ideas presented in this thesis will require a more robust mechanism to prevent this kind of behaviour.

7.1.3 Legal framework to pursue licence agreement violators

The marketplace has no meaningful way to legally pursue people that do not follow the stipulated licence agreement. This means that someone can download the paper for open-science usage and then commercialize the idea with limited repercussions. Ideas on the platform are protected by patents and so it is the idea creators responsibility to pursue the infringer of the patent in a court of law. This is not strictly a fault with the marketplace but rather patents at large.

In the Tesla example, Toyota could take the idea presented in the Dalhousie paper and implement it in their own battery design, never disclosing having done this. There is no way for the marketplace to prevent this kind of behaviour as the information is in the public domain already. It is then the university of Dalhousie's responsibility to litigate against Toyota for this infringement.

Ultimately with or without the marketplace for ideas plagiarism and idea theft will be present whenever there is public information. This is not a fault of the marketplace or the mechanism but rather IP protection at large.

As a result, the this thesis assumes the existence of a robust global legal framework that can fully defend a patents exclusivity to enforce the correct ”ownership” of patents. This assumption is not tractable in real life and could arguably be the largest downfall of the overall system due to the impracticality of this assumption.

7.2 Market mechanisms and incentive design analysis

Next, the market mechanism is critically analyzed. This analysis includes the proposed sealed bid auction and Harberger Tax systems and their impact on user's behaviour and incentives.

7.2.1 Reduction in information asymmetry

The marketplace for ideas aims reduce information asymmetry between buyers and sellers to decrease adverse selection. This is achieved through publicly displaying patent and research information associated with the idea to give the buyer maximum information about the patent they are bidding for. However, a paper and a patent alone in no way quantifies sufficient due diligence for a potential buyer. A potential buyer would need to conduct extensive research to value the idea. Additional third party services would still be required to quantify any meaningful patent valuation. From a due diligence perspective, the market for ideas provides little advantage over existing brokerage services.

With that said, the proposed marketplace for ideas does provide a meaningful decrease in past pricing information asymmetry by storing a public record of historic ownership and pricing information. Each idea in the marketplace stores information on who owned it, when it was owned and how much was paid for it. This is in stark contrast to the current patent market where it is extremely difficult to know the provenance of any licence or patent. By publicizing this information potential buyers have considerably more information at their disposal when trying to quantify a patent's value as they can compare to past trades for similar patents.

7.2.2 Market coordination inefficiencies

A problem with selling anything on auction is coordinating potential bidders to be present during the auction. The market presented in this thesis, tries to decrease this inefficiency by running long duration auctions, upwards of two weeks, thereby giving potential bidders more time to participate. This also gives seller's opportunity to advertise the auction. While a long duration does give more time to find bidders it does not ensure that the optimal bidder knows about the auction. This leaves the onus on the seller to sufficiently advertise their auction to attract buyers.

As a counter measure, Harberger Tax effectively acts as a continuous auction mechanism enabling bidding on licenses after the auction has closed. This helps to decrease coordination inefficiency as new bidders that missed the auction can still participate through Harberger Tax buyout submission.

7.2.3 Failed auctions and post auction negotiations

If valid bids fail to reach the reserve price then the auction will not clear. In this case no exchange occurs and the auction fails. This is not an ideal outcome for any participant.

As a point of comparison, Ocean Tommo's auctions experience post-auction negotiation when bids fail to meet the reserve price (Jarosz et al., 2010). These post auction negotiations change the incentives of sellers and bidders. Sellers are incentived to set reserve prices too high. Additionally bidders may use the auction as a means of limited price discovery and then use the information in post-auction negotiations.

In the proposed marketplace post auction negotiation is difficult to perform because post auction direct sales require coordination. A direct exchange from a seller to a predefined buyer is possible through a cumbersome interaction between the buyer and seller. For example a post auction sale would require the seller to make a single licence fixed price sale at the privately negotiated price. Then, the buyer buys the licence as soon as it is listed on the market. If orchestrated correctly, this technique enables private bilateral negotiations to occur offline and then the trade is settled within the market.

Whether this is a positive or negative result remains an open question. On the one hand, there are situations in which private bilateral negotiations are required to protect a sellers strategy. Say for example a patent holder is selling patent licences as a last ditch effort to save their company. In this case they might not want to publicise that they are selling and so a public auction is not appropriate. Rather, they want to negotiate with buyers while protected by a non-disclosure agreement and behind closed doors. The proposed marketplace still provides benefits like payment settlement and recording licence information, even in this situation.

On the other hand, private bilateral negotiations result in a continuation of the current market opacity and shallowness. One could argue, therefore, that ideally all sales should be public for the betterment of the market. This defeats the purpose of having an auction in the first place. To achieve true price discovery and improved allocation for ideas within the market the seller should not set unrealistically high reserve prices and enable Harberger Tax to ensure that the licence is always owned by the person that values it the most.

7.2.4 Improvements in allocation efficiency

There are two stages of a patent sale process that have improved allocation efficiency as a result of the proposed design. Firstly, the initial allocation of licences to bidders through public sealed bid auctions ensures that the bidder who values the licence the most will win the auction. Secondly, Harberger Tax acts as a *continuous auction mechanism*. If after an auction concludes and a future company values the idea more they will be able to purchase it.

These two points mean that ideas are always allocated to companies that value them the most, even after a sale has concluded through Harberger Tax buyouts. Further research would involve simulation and financial modeling to numerically validate these claims.

7.2.5 Public auctions and licence information disadvantages

Companies keep their patent portfolios and transactions private to protect strategic information from competitors (McClure, 2015). However the model proposed in this thesis takes a strong stance that patent information, trades and ownership should be public. Openness in this regard improves allocative efficiency, creates a more effective pricing scheme and increases market liquidity overall. This is beneficial for patent buyers and sellers but it might not be for existing patent holders who would want to keep their portfolios private.

As a result, the marketplace proposed in this thesis will not work for all kinds of patents and intellectual property. In fact, many patent holders will be actively against this design because they would sacrifice the strategic advantage of obscurity in portfolio holdings.

For this reason the design focuses primarily on works that should be kept in the public domain; works that are published from universities. Licences issued within this platform can form part of a larger private patent portfolio if need be, but the individual licences and the owners thereof remain public if the ideas came from universities.

7.2.6 Implications of licence revocation through Harberger Tax

If a licence held by a company is revoked as a result of a Harberger Tax buyout then the company that loses access to the licence could be significantly disadvantaged. This is compounded if they have invested into infrastructure to capitalize on the licence. Say, for example, Tesla invests billions into building a factory to build the battery patented by the University of Dalhousie. Toyota submits a buyout request, which is successful. This results in the transfer of the licence from Tesla to Toyota. This could be very detrimental to Tesla who has invested vast sums of money into the manufacturing facilities. There are two main considerations here in dealing with this situation which are now explored.

Firstly, as identified outlined in section 4.5.4, after a licence is revoked from Harberger Tax the previous licence owner can continue using the licence for a calculated period of time before it becomes invalid. This grace period is a function of how long the licence holder held the licence for before revocation. This grace period of time will enable Tesla to either:

1. try and buy the licence back from Toyota,
2. build batteries for the remaining duration of their licence then stop or
3. find another licence/patent that they can capitalize on in their production facility or change their design.

None of these options are necessarily ideal. It is important to remember with Harberger Tax that if Tesla had set their valuation of the licence equal to their true valuation then they will not lose any future profit from having the licence revoked. Say Tesla calculates their profit and costs of the factory and determines the value of the licence to be $1 billion. If Toyota values the licence, at $2 billion and buys it off Tesla for $1.1 billion then Tesla has not lost anything because they will have been paid the amount that they had valued the licence at in the first place.

This ability to revoke a licence issued to a company who has invested in it could expose companies to unpredictable litigation risk. For example a PAE could buy Tesla's licence from them with the intention of suing them for infringing on the patent. This risk would make owning licences that are issued in this way risky for companies unless they knew that they could protect against this risk. However, the licence validity period after revocation mitigates this risk. Now, the PAE would need to hold the licence until their targets licence expires. This makes this form of attack very expensive as they will need to pay tax on it for the duration. Further research into the implications of Harberger Tax with intellectual property and the implications for PAEs is required to better understand this particular scenario.

7.2.7 Harberger Tax and representative pricing

If an idea is listed in the marketplace with Harberger Tax then at all points a market representative price can be attained. This is a significant improvement over the current patent marketplace as each and every idea listed with Harberger Tax will have a current asking price assessed by the current owner. This means that the price of the idea at any point in time should fully reflect as much available market information as possible. Because the current owner must set this price, the pricing allocation becomes more reflective of the true valuation. This results in an increased market efficiency.

7.2.8 Strategic implications of Harberger Tax right of first offer

The Harberger Tax model presented has a notion of right of first offer. This enables the current licence owner to match a buyout offer submitted for a licence they own. For a buyout offer to be successful it needs to submit a valuation of at least 5% more than the current listed price. Once a buyout has been submitted the current licence holder has two weeks to either increase their private valuation to match the buyout offer and keep the licence or do nothing and choose to lose the licence.

This mechanism introduces an opportunity for licence holders to get away with unduly low tax rates if the licence in question is not highly demanded. This works by the licence holder setting an arbitrary low valuation for the licence. This can be a small fraction of their true valuation. If the licence is not in high demand then the holder does not pay a representative tax because no one will attempt a buyout. If at any point someone tries to buyout the licence the licence holder increases their valuation until 1) the buyout offers stop if the licence holders valuation is larger than the buyout bidders valuation or 2) the buyout succeeds if the buyout bidders valuation is larger than the current licence holder. In either case there is an inefficiency introduced due to the period of time wherein the value was under reported by the licence holder and so the licence issuer misses out on some potential revenue.

This means that the Herberger Tax model might be only appropriate in markets where there is active demand for licences and where licence holders are incentivized to set true valuations. An alternative solution is to remove the right of first offer for licence holders. If this is removed then current holders are incentivized to more accurately report their licence valuations as at any point the licence can be revoked through a buyout. As a result, the current holder will set the value to their true valuation to ensure that even if there is a buyout they will extract their valuation from the licence rather than getting undercompensated. However, the removal of the right of first offer makes the Harberger Tax system extremely risky for licence holders. It is for this reason that it was introduced in the first place: to protect licence holders against opportunistic buyouts.

7.3 Technical implementation viability analysis

Next, the presented marketplace's implementation is analyzed in terms of usability, viability and sustainability. A number of real world security considerations are also discussed.

7.3.1 Application usability

Blockchain application user experience (UX) is notoriously difficult to get right. This is primarily due to lack of tooling within the ecosystem which is constantly evolving and changing. As a result, a lot of work is left up to the developer to make the UX approachable. This is in contrast to the current web2.0 world of today where there is a wealth of packages, libraries and tools at a developers disposal to make easy to use applications.

An additional problem is that many applications make assumptions around their users technical competency by targeting a specific user group of people already within the crypto ecosystem. For example an application might assume that its users have a wallet and know how to use it. However, this makes the application unapproachable by a "normal" user who does not have this spesific skill set. While this is fine for crypto specific applications it makes gaining any kind of real world traction difficult in broader contexts.

In an effort to make the application as usable as possible the marketplace presented in this thesis aims to abstract as much complexity away from the user as possible. This is done primarily by removing gas costs through meta transactions, offering non-custodial web based wallet integration with Portis and ease Fiat onboarding with Wyre. While these considerations help to simplify the application, they do not necessarily mitigate all of the cognitive overhead associated with a blockchain application. Further development and tooling is required to make the market as approachable as toady's web2.0 application.

Ultimately any blockchain application should be simple enough that the user does not even know they are using blockchain or how it works. You should not need to know how TCP/IP works to use Facebook. The same mantra should apply to blockchain based applications.

7.3.2 Need for a wallet and foreign nomenclature

Portis successfully abstracts away much of the complexity with having a wallet as the user no longer needs to manage their private keys in a complex way. However it does not completely remove the cognitive overhead for new users as you still need to interact with a foreign interface, understand what a wallet is and how to interact with it. There is a learning curve associated with this. In time, this will become more common place in other applications but for now it is quite a foreign concept.

Many blockchain applications introduce a set of jargon which is difficult for new users to pick up and understand. To try and mitigate this the proposed marketplace used as simple terminology and wording as possible when describing the user interface and process to the user. Future implementations should include a set of tutorial videos explaining every part of the user interaction flow.

7.3.3 On and off boarding

Fiat on and off boarding complexity has been minimized through Wyre. However it is still challenging for new users and this approach makes assumptions around the kinds of payment options companies have. For example Wyre requires the buyer to have access to a credit card to buy Dai. This restriction might limit who can interact with the platform.

Ultimately the proposed UX improvements will help in bringing new users into the application but real world testing is required to further validate these claims. Moreover, the blockchain ecosystem as a whole needs time to mature further before wide scale adoption is truly possible. This is not unique to the marketplace presented but any blockchain based application in general.

7.3.4 Company registration verification

The registration process proposed does not restrict any user from signing up as a company to participate in auctions. This openness is part of the design to try and increase the number of bidders, thereby maximizing the sellers potential revenue. Additionally, open registration does not discriminate against one user group, size or location.

Licence buyers are incentivised to correctly report their registration information as if this is done incorrectly the licence may be invalid in a court of law. Say, for example, that Tesla incorrectly registers on the platform before an auction as a different company. Even if they win the auction and have a licence issued to them they will not be legally protected by the licence as it was issued to a different company. Clearly holding and subsidiary company configurations are posible involving sub licensing but even in these settings the core premise remains: incorrectly reporting your company information runs a risk of not being able to legally execute on the licence issued.

Future versions of the platform should incorporate a more robust mechanism of validating company registration.

7.3.5 Auction gaming through multiple bid submissions

An adversarial bidder can game the proposed auction mechanism by submitting multiple sealed bids and only choosing to reveal the bid that maximize their utility. During the commit phase of the auction, all bids are kept secret and not shared between bidders. However, during the reveal phase all bid information becomes public to all auction bidders. As a result, a determined bidder could choose to submit many bids over a range of values in the commit phase and then only reveal the bid that will guarantee the win. This means that the bidder will underpay on what their true valuation of the licence is by winning the auction with a bid that is lower than their maximum willingness to pay.

There are a number of ways to minimize this form of mechanism exploitation. Firstly, all bidders on the platform must register before they can participate. Coupled with this, all company accounts are limited to submit exactly one bid in each auction. Any subsequent bid past their initial bid is rejected. Therefore a bidder must cancel and re-submit to update their bid amount. This, however, does not prevent the auction mechanism from sybil attack wherein an adversarial bidder creates multiple accounts and chooses to only revel the bids on one of their accounts thereby achieving the same mechanism exploitation.[1] This kind of attack is difficult to prevent against given the

[1]A sybil attack is an attack on an identity system where an attacker creates many pseudonymous

company verification mechanism presented is not extremely robust, as discussed in point 7.3.4 above.

One proposed way to defend against this attack is to require all bids to be accompanied by a refundable stake. Ideally, this stake should be proportional to the bidders bid but this fundamentally defeats the point of a sealed bid auction as this will leak information. Alternatively, the stake could be proportional to the auction price floor, thereby setting the same monetary requirement for all bidders. This, however, does not prevent this attack as a malicious bidder with a high budget can still create multiple accounts and fund them accordingly to conduct this attack.

Further research into more robust company registration mechanisms and additional design considerations are required to fully mitigate this attack vector.

7.3.6 Smart contract front running

Smart contract systems introduce a number of potential attack vectors and vulnerabilities. For one, block mining and propagation time opens systems up to front running of transactions. This occurs when someone monitors the blockchain for transactions they can take advantage of and submits a transaction with a higher gas price which results in the transaction being mined before the exploited transaction.[2] This kind of attack is common in decentralized exchanges (Dexes) wherein front runners look for trades that they can exploit through submitting an order before the legitimate order.

Auction front running is not possible due to the commit reveal bid process used. If the bids were open then front running becomes a major problem: a bidder can simply wait until the end of the auction and bid just above the highest bid, winning the auction. This is commonly referred to as bid sniping in traditional auctions (Trevathan and Read, 2006). In this case the blockchain makes this behaviour even easier to do due to the slow block processing times. If multiple bidders are both attempting to snipe a bid then they will engage in a gas bidding war wherein each progressively raises the gas associated with their bid, not the bid amount, to ensure that their transaction is included in the block and that their competitors is not. Sealed bids prevent this behaviour as bidders have no knowledge of their competitors bids.

The Harberger Tax mechanism however is vulnerable to a kind of front running attack. Take the case of a licence buyout. Say Tesla owns a licence and Toyota has submitted a buyout request with a sufficiently high proposed amount. Tesla can choose to increase their private valuation, thereby making the Toyota buyout invalid. The front running attack can come in where Toyota waits until just before the end of the buyout period (defined as two weeks in section 5.2.1.1) and submits a transaction that will *just* beat Tesla's valuation. If this attack is done correctly this transaction should be included in the last block before the end of the buyout period. Toyota can monitor the blockchain to see if Tesla submits an increase and if they do they can immediately increase their buyout

identities to manipulate the system.

[2] Gas price acts as a mechanism to compensate miners and to prioritize transactions. If a transaction has higher gas price than another then it will get mined first. This is effectively equivalent to a first-price auction.

request. This behaviour will prevent Tesla from responding in a meaningful way as they do not have time to respond to the buyout request.

To mitigate this problem a number of solutions could be introduced in further versions. The simplest is to enforce that the buyout requests and price updates have a safe number of blocks between them and the end of the buyout period. This ensures that both parties have sufficiently long to respond in a buyout attempt. Another option is to place gas limits on transactions submitted to the contract. This limits the ability for users to front run to some extent as gas bidding wars become impossible by construction.

7.3.7 Front end security vulnerabilities

Due to the hosting setup used in the market the front end is susceptible to a number of attacks. If an attacker was able to access the account details for any of the centralized hosting services used (Gandi, Cloudflare or Netlify) they could upload a malicious front end application in place of the existing application with the intention stealing users funds. For example, the malicious website could replace all smart contract interactions with a transaction that siphons users wallets to the attacker.

The provided defence against this is Secure Socket Layer (SSL) through Hypertext Transfer Protocol Secure(HTTPS). This ensures that all web pages loaded by a user have a signature and certificate and if either of these change the user is informed. This defence works against some attack vectors but would not prevent others, such as a DNS takeover wherein an attacker points the legitimate domain name to a malicious web server.

Furure implementations, as mentioned in Section 6.3.7, should host all website files directly on IPFS. When a user loads the page their browser will be able to validate that the content loaded matches that claimed by the developer on IPFS. This prevents any malicious code from being injected as the hashes will no longer match and the user will be informed.

7.4 Chapter conclusion

This chapter critically analyzed the proposed design from three main points of view including regulatory and legal, mechanism and incentives and technical design and implementation. The design presented improves in some ways the current brokerage market and provides valuable visibility to the opaque market. Increased allocation efficiency is achieved through the market making mechanisms and Harberger Tax system. However, there are still many legal considerations that need to be made as well as regulatory clearance before the proposed marketplace for ideas could become a real world product in today's ecosystem. The technical implementation is robust and maintainable, but requires additional development and tooling until the application is truly production ready. Ultimately the proposed system requires extensive testing before it is appropriate to launch in a real world setting.

Additionally, the mechanisms presented in this thesis are only one of many possible constructions. Appendix J presents a number of additional areas of research that were considered in the writing of this thesis.

Chapter 8

Conclusion

Commercially viable ideas are not always born to their best users. This results in an allocation efficiency problem between idea creators and implementors. As a subset of all idea creators, academics often publish works that would be valuable to industry, but they lack a simple monatization mechanism to allocate them to their best implementors. More broadly, there are complexities in trading ideas such as information asymmetry, patent complexity and regulation. This complexity is present in the current patent market which results in the proliferation of adverse selection and poor market efficiency.

This thesis presented posible solutions to some of these problems by proposing a marketplace for ideas which enables academics to monetize valuable ideas through a blockchain based patent licensing scheme. The main technical contribution of this paper is the automatic market mechanism including sealed bid auctions used in the initial allocation of licenses. This auction leverages a privacy preserving, publicly verifiable commit reveal scheme which runs on the Ethereum blockchain. Additionally, smart contract enforced Harberger Tax is proposed to fairly tax ideas that generate commercial value for license owners. These smart contracts also create cryptographically unique tokenized licenses. This enables academics to retain ownership of their intellectual property, while issuing licenses. Lastly, atomic, trustless and secure payments are performed without any administrative intervention using smart contracts. Thanks to the blockchain, all operations are publicly verifiable and trustless thereby preventing any possible manipulation from the market operator or market participants.

The market mechanism is successful in publicising licensing and ownership information, thereby decreasing information asymmetries. Additionally, the implemented auction and Harberber Tax systems ensure that those that value the licences the most own them. This improves allocation efficiency while increasing social welfare by funding the institutions that supported the original research.

Bibliography

David S. Abrams, Ufuk Akcigit, Gokhan Oz, and Jeremy Pearce. The Patent Troll: Benign Middleman or Stick-Up Artist? *SSRN Electronic Journal*, pages 1–55, 2019. doi: 10.2139/ssrn.3361215.

Ufuk Akcigit, Murat Alp Celik, and Jeremy Greenwood. Buy, Keep, or Sell: Economic Growth and the Market for Ideas. *Econometrica*, 84(3):943–984, 2016. ISSN 0012-9682. doi: 10.3982/ecta12144.

George A. Akerlof. Perspective des parcours de vie et sociologie de l'individuation. *Sociologie et sociétés*, 45(1):279–300, 1970. ISSN 0038-030X. doi: 10.7202/1016404ar. URL http://dx.doi.org/10.1016/j.refiri.2017.07.010{%}0Ahttp://coop-ist.cirad.fr{%}0Ahttp://www.theses.fr/2014AIXM5048{%}0Ahttp://www.cairn.info/revue-management-et-avenir-2010-6-page-84.htm{%}0Ahttp://www.cairn.info/bifurcations--9782707156006-page-349.htm{%}0Ahttp://w.

George A. Akerlof. Writing the "The Market for 'Lemons'": A Personal Interpretive Essay - NobelPrize.org, 2001. URL https://www.nobelprize.org/prizes/economic-sciences/2001/akerlof/article/.

Wisconsin Alumni. Nine to Points to Consider in Licensing University Technology. *Time*, pages 1–17, 2006.

Leandro Arozamena and Federico Weinschelbaum. The effect of corruption on bidding behavior in first-price auctions. *European Economic Review*, 53(6):645–657, 2009. ISSN 00142921. doi: 10.1016/j.euroecorev.2008.11.001.

Susan Athey, Jonathan Levin, and Enrique Seira. Comparing open and sealed bid auctions: Evidence from timber auctions. *Quarterly Journal of Economics*, 126(1):207–257, 2011. ISSN 00335533. doi: 10.1093/qje/qjq001.

Autm. Driving the Innovaon Economy, academic technology transfer in numbers From. (June): 1–2, 2017.

Ian Ayres and Lisa Larrimore Ouellette. A market test for bayh-dole patents. *Cornell Law Review*, 102(2):271–331, 2017. ISSN 00108847. doi: 10.31228/osf.io/kwvzg.

By Francis M Bator. The Anatomy of Market Failure Author (s): Francis M . Bator Reviewed work (s): Source : The Quarterly Journal of Economics , Vol . 72 , No . 3 (Aug ., 1958), pp . 351-379 Published by : Oxford University Press THE ANATOMY OF MARKET FAILURE. 72(3):351–379, 2012.

Mario Benassi and Alberto Di Minin. Playing in between: Patent brokers in markets for technology. *R and D Management*, 39(1):68–86, 2009. ISSN 00336807. doi: 10.1111/j.1467-9310.2008.00537.x.

James Bessen and Michael J. Meurer. The direct costs from NPE disputes. *Cornell Law Review*, 99(2):387–424, 2014. ISSN 00108847. doi: 10.2139/ssrn.2091210.

Christian Bessy, Eric Brousseau, and Stéphane Saussier. Payment Schemes in Technology Licensing Agreements: A Transaction Cost Approach. *SSRN Electronic Journal*, 0033(1):1–28, 2011. doi: 10.2139/ssrn.1259394.

David S Bloch. Monetization of Patent and Other IP Rights An Introduction. Technical report, 1960. URL http://www.infoplease.com/spot/trademarks1.html.

John Bone. View from the Federal Circuit: An Interview with Chief Judge Randall R. Rader | Stout, 2010. URL https://www.stout.com/en/insights/article/view-federal-circuit-interview-chief-judge-randall-r-rader/.

Roberto Burguet and Martin K. Perry. Bribery and favoritism by auctioneers in sealed-bid auctions. *B.E. Journal of Theoretical Economics*, 7(1), 2007. ISSN 19351704. doi: 10.2202/1935-1704.1219.

Colleen V. Chien. From arms race to marketplace: The complex patent ecosystem and its implications for the patent system. *Hastings Law Journal*, 62(2):297–356, 2010. ISSN 00178322.

CIPC. Companies and Intellectual Property Commission, 2019.

COGR. The Bayh-Dole Act. A Guide to the Law and Implementing Regulations. Technical Report October 1999, 1999.

Alan Conway. A History of the United States, 1956. ISSN 0020-5850. URL http://www.ladas.com/Patents/USPatentHistory.html.

Cryptostats. RDC:Kitty stats, 2019. URL http://www.rolldice.club/cryptokitties/index.php.

Simon De la Rouviere. What is Harberger Tax & Where Does The Blockchain Fit In?, 2018. URL https://medium.com/@simondlr/what-is-harberger-tax-where-does-the-blockchain-fit-in-1329046922c6.

Holly Else. Thousands of scientists run up against Elsevier's paywall. *Nature*, feb 2019. ISSN 0028-0836. doi: 10.1038/d41586-019-00492-4.

Eugene F. Fama. Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2):383, may 1970. ISSN 00221082. doi: 10.2307/2325486.

United Nations Economic Commission for Europe. Intellectual Property Intellectual Property Commercialization Commercialization POLICY OPTIONS AND PRACTICAL INSTRUMENTS. Technical report, 2011.

Stefania Fusco, Francesco Lissoni, Catalina Martinez, and Valerio Sterzi. Monetization Strategies of University Patents Through Paes: An Analysis of US Patent Transfers. *SSRN Electronic Journal*, (June), 2019. doi: 10.2139/ssrn.3410086.

Hisham S. Galal and Amr M. Youssef. Verifiable sealed-bid auction on the ethereum blockchain, 2019. ISSN 16113349.

G.F. Why bitcoin uses so much energy - The Economist explains, 2018. URL https://www.economist.com/the-economist-explains/2018/07/09/why-bitcoin-uses-so-much-energy.

Robert Goldman. *Schwartzs Patent Law and Practice*. 2000. ISBN 978-1617467554.

David Haigh. Brand Finance - Global Intangible Value Exceeds US Trillion for the First Time. pages 1–3, 2018.

Morten T. Hansen and Julian Birkinshaw. The innovation value chain, 2007. ISSN 00178012. URL https://hbr.org/2007/06/the-innovation-value-chain.

Dietmar Harhoff, Bronwyn H Hall, Georg Graevenitz von, Karin Hoisl, Stefan Wagner, Alfonso Gambardella, and Paola Giuri. The strategic use of patents and its implications for enterprise and competition policies. *European Commission*, pages 1–308, 2007.

F. A. Hayek. The Pure Theory of capital. *The Pure Theory of Capital*, 12:1–439, 2012. doi: 10.4324/9780203820247.

H. Hernández, N. Grassano, A. Tübke, L. Potters, P. Gkotsis, and A. Vezzani. The 2018 EU Industrial R&D Investment Scoreboard; EUR 29450 EN. Technical report, Publications Office of the European Union, 2018. URL http://iri.jrc.ec.europa.eu/scoreboard18.html.

Instructor Dmytro Hryshko. What is Technology? *Computer Fraud and Security*, 2001(7): 17–19, 2001. ISSN 13613723. doi: 10.1016/S1361-3723(01)00720-5.

John Jarosz, Robin Heider, Coleman Bazelon, Christine Bieri, and Peter Hess. Patent Auctions Patent Auctions: How Far Have We Come? les Nouvel(March):11–30, 2010.

Joe Runge. How Long Does a Patent Last? | LegalZoom, 2016. URL https://www.legalzoom.com/articles/how-long-does-a-patent-last.

Casey Johnston. Google, needing patents, buys Motorola wireless for $12.5 billion, 2011. URL http://arstechnica.com/gadgets/news/2011/08/google-to-buy-motorola-in-effort-to-defend-itself-from-patent-bullies.ars.

Jonathan Levin. Notes on the Revenue Equivalence Theorem. 2000.

Anne Kelleyf. Practicing in the Patent Marketplace Author (s): Anne Kelley Source : The University of Chicago Law Review , Vol . 78 , No . 1 (Winter 2011), pp . 115-137 Published by : The University of Chicago Law Review Stable URL : https://www.jstor.org/stable/41. 78(1):115–137, 2019.

Stephen Knack, Nataliya Biletska, and Kanishka Kacker. Deterring Kickbacks and Encouraging Entry in Public Procurement Markets: Evidence from Firm Surveys in 90 Developing Countries. Technical Report 2, 2019. URL http://econ.worldbank.org.

Naomi R. Lamoreaux, Kenneth L. Sokoloff, and Dhanoos Sutthiphisal. Patent alchemy: The market for technology in US history. *Business History Review*, 87(1):3–38, 2013. ISSN 00076805. doi: 10.1017/S0007680513000123.

Mark A. Lemley. Are universities patent trolls? *Perspectives on Commercializing Innovation*, (April 2007):531–546, 2011. doi: 10.1017/CBO9781139051262.021.

Mark A Lemley and Nathan Myhrvold. How to make a Patent Market. *HOFSTRA LAW REVIEW (Rk. 68 Law Journals)*, 36(2):101–103, 2009. URL http://ssrn.com/abstract=1012726.

Yvan Lengwiler and Elmar Wolfstetter. Auctions and corruption: An analysis of bid rigging by a corrupt auctioneer. *Journal of Economic Dynamics and Control*, 34(10):1872–1892,

2010. ISSN 01651889. doi: 10.1016/j.jedc.2010.03.002. URL http://dx.doi.org/10.1016/j.jedc.2010.03.002.

Matt Levy. Why Universities Oppose Real Patent Reform: Money - Patent Progress, 2015. URL https://www.patentprogress.org/2015/05/07/why-universities-oppose-real-patent-reform-money/.

Live Coin Watch. Ethereum (ETH) live coin price, charts, markets & liquidity | ETH to USD, 2019. URL https://www.livecoinwatch.com/price/Ethereum-ETH.

Brian J. Love. Testimony of Brian J. Love, California Assembly Select Committee on High Technology, Informational Hearing on Patent Assertion Entities. *SSRN Electronic Journal*, 369, 2013. doi: 10.2139/ssrn.2347138.

Brian J Love, Y J L Ale, and T Ech. Brian J. Love * 16 Y. 285:285–343, 2014.

Brian J. Love, Kent Richardson, Erik Oliver, and Michael Costa. *An Empirical Look at the 'Brokered' Market for Patents*, volume 83. 2017. ISBN 2303812104576. doi: 10.2139/ssrn.3056444.

Brian J Love, Erik Oliver, and Michael Costa. U . S . PATENT SALES BY UNIVERSITIES. pages 1–30, 2019.

MakerDao. The Dai Stablecoin System Whitepaper. Technical report, 2017. URL https://makerdao.com/.

James E Malackowski, Keith Cardoza, Cameron Gray, Rick Conroy, and James E Malackowski. The Intellectual Property Marketplace: Emerging Transaction and Investment Vehicles. 27 (2):1–11, 2007.

Kazi Mamun and Samira Sadaoui. Combating shill bidding in online auctions. *International Conference on Information Society, i-Society 2013*, (June 2013):170–176, 2013.

Robert C. Marshall and Michael J. Meurer. The Economics of Auctions and Bidder Collusion, 2005.

Francis T Mcandrew and Knox College. Encyclopedia of Evolutionary Psychological Science. *Encyclopedia of Evolutionary Psychological Science*, (January), 2019. doi: 10.1007/978-3-319-16999-6.

Ian David McClure. From a patent market for lemons to a marketplace for patents: benchmarking IP in its evolution to asset class status.(Trolls or Troll-Takers: Do Intellectual Property Non-Practicing Entities Add Value to Society?). *Chapman Law Review*, 18(3):759–797, 2015.

Luke Meiklejohn. *How to attribute credit if you must*. PhD thesis, 2020. URL https://www.dropbox.com/s/qtqckmqaitch6nu/Thesis27-Dec.pdf?dl=0.

Philip Mendes. To License a Patent – or , to Assign it : Factors Influencing the Choice. (July), 2018.

J. Mestre-Ferrandiz, Jon Sussex, and Adrian Towse. *The R & D Cost of a New*. 2012. ISBN 9781899040193.

Joe Mullin. Patent war goes nuclear: Microsoft, Apple-owned "Rockstar" sues Google | Ars Technica, 2013. URL https://arstechnica.com/tech-policy/2013/10/patent-war-goes-nuclear-microsoft-apple-owned-rockstar-sues-google/.

Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report, 2008. URL https://bitcoin.org/bitcoin.pdf.

Namecoin. Namecoin, 2010. URL https://www.namecoin.org/.

Daniel Oberhaus. Tesla May Soon Have a Battery That Can Last a Million Miles. *Wired*, 2019. ISSN 1059-1028. URL https://www.wired.com/story/tesla-may-soon-have-a-battery-that-can-last-a-million-miles/.

Ocean Tomo. Intangible Asset Market Value | Ocean Tomo. 2010. URL http://www.oceantomo.com/productsandservices/investments/intangible-market-value.

Cristina Odasso, Giuseppe Scellato, and Elisa Ughetto. Selling patents at auction: An empirical analysis of patent value. *Industrial and Corporate Change*, 24(2):417–438, 2015. ISSN 14643650. doi: 10.1093/icc/dtu015.

David S. Olson. Taking the utilitarian basis for patent law seriously: The case for restricting patentable subject matter. *Temple Law Review*, 82(1):181–240, 2009. ISSN 08998086. doi: 10.2139/ssrn.933167.

Orcid. ORCID, 2019. URL https://orcid.org/.

Thomas Owen. Apple Takes Over Former Nortel Patents - Business Insider, 2012. URL https://www.businessinsider.com/apple-rockstar-bidco-nortel-patents-2012-11?IR=T.

David C. Parkes, Michael O. Rabin, Stuart M. Shieber, and Christopher Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. *Electronic Commerce Research and Applications*, 7(3):294–312, 2008. ISSN 15674223. doi: 10.1016/j.elerap.2008.04.006.

Eric Posner, Bill Vickrey, and Henry George. Reimagining Property A Philosophical Look at Harberger Taxation 1 . A Quick and Dirty Introduction to Harberger Taxation. (April), 2019.

Eric A. Posner and E. Glen Weyl. Property is only another name for monopoly. *Journal of Legal Analysis*, 9(1):51–123, 2017. ISSN 19465319. doi: 10.1093/jla/lax001.

Natalie Raffoul and Art Brion. Reasons for Patent Protection and Cost-effective Patent Filing Options for SMEs. *Technology Innovation Management Review*, 1(3):29–33, 2011. ISSN 1927-0321. doi: 10.22215/timreview505.

Andrew Ramer. Success Story: Ocean Tomo Revolutionizing IP Auctions. Technical report, 2006. URL www.oceantomo.com.

James W. Roberts and Andrew Sweeting. When should sellers use auctions? *American Economic Review*, 103(5):1830–1861, 2013. ISSN 00028282. doi: 10.1257/aer.103.5.1830.

Rsk. RSK - Home, 2019. URL https://www.rsk.co/.

CJ Ryan and Brian L. Frye. An Empirical Study of University Patent Activity, 2018. URL https://jipel.law.nyu.edu/vol-7-no-1-2-ryan/.

N Sadao and N Yoichiro. Complementarity, Fragmentation, and the Effects of Patent Thickets. 2014. URL http://ideas.repec.org/p/eti/dpaper/14001.html.

Solidity. Solidity — Solidity 0.4.23 documentation, 2018. URL https://solidity.readthedocs.io/en/v0.4.23/.

Alberto Sonnino, Michal Król, Argyrios G. Tasiopoulos, and Ioannis Psaras. AStERISK: Auction-based Shared Economy ResolutIon Markets for Blockchain Platforms. 2019. doi: 10.14722/diss.2019.23001.

Kristi Stathis. Ocean Tomo, LLC Announces 10-Year Relationship with ICAP | Business Wire, 2009. URL https://www.businesswire.com/news/home/20090616006261/en/Ocean-Tomo-LLC-Announces-10-Year-Relationship-ICAP.

Shaw Tim. Daily Medieval: Patent Law, 2012. URL http://dailymedieval.blogspot.com/2012/08/patent-law.html.

Jarrod Trevathan and Wayne Read. Undesirable and fraudulent behaviour in online auctions. *SECRYPT 2006 - International Conference on Security and Cryptography, Proceedings*, pages 450–458, 2006. doi: 10.5220/0002100704500458.

Geertrui Van Overwalle, Esther Van Zimmeren, Birgit Verbeure, and Gert Matthijs. Dealing with patent fragmentation in ICT and genetics: Patent pools and clearing houses, 2007. ISSN 13960466. URL https://firstmonday.org/ojs/index.php/fm/article/view/1912/1794{#}v6.

Vitalik Buterin. Ethereum White Paper, 2018. URL https://github.com/ethereum/wiki/wiki/White-Paper.

Xiang Wang, Zhenzhe Zheng, Fan Wu, Xiaoju Dong, Shaojie Tang, and Guihai Chen. Strategy-proof data auctions with negative externalities. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 1269–1270, 2016. ISSN 15582914.

Bitcoin wiki. Smart Property - Bitcoin Wiki, 2012. URL https://en.bitcoin.it/wiki/Smart{_}Property.

Bitcoin wiki. Script - Bitcoin Wiki, 2019. URL https://en.bitcoin.it/wiki/Script.

WIPO. Intellectual property policies for universities, 2019. URL https://www.wipo.int/wipo{_}magazine/en/2019/si/article{_}0010.html.

Wits. SA Copyright Law - Copyright Guidelines for Staff & Students of the University of the Witwatersrand, Johannesburg (WITS) - LibGuides at University of the Witwatersrand, 2019. URL https://libguides.wits.ac.za/Wits{_}Copyright{_}Guidelines/SA{_}Copyright{_}Law.

Jie Xiong and Qi Wang. Anonymous Auction Protocol Based on Time-Released Encryption Atop Consortium BlockChain. *International Journal of Advanced Information Technology*, 09(01):01–16, 2019. ISSN 22311920. doi: 10.5121/ijait.2019.9101.

Tomoya Yanagisawa and Dominique Guellec. the Emerging Patent Marketplace. *OECD Science, Technology and Industry Working Papers*, pages 1–52, 2009.

Pluvia Zuniga. The State of Patenting at Research Institutions in developing countries: Policy approaches and practices. *... Report to the publication: Intellectual property and the ...*, (4):96, 2011. URL http://www.wipo.int/econ{_}stat/en/economics/pdf/WP4{_}Zuniga{_}to{_}be{_}published.pdf.

Appendix A

Marketplace service providers

The tables A.1 and A.2 below show a number of popular software and consulting services to aid companies in evaluating patents.

Self-use software	Website
Innography	innography.com
Pantros IP	ip.com
PatentRatings	patentratings.com
Thomson Innovation	info.thomsoninnovation.com
Ambercite	ambercite.com
Google Patents	patents.google.com
MaxVal-IP	maxval.com
Astamuse	astamuse.co.jp/en
Sumobrain	sumobrain.com
PatentBuddy	patentbuddy.com
PetaPator	petapator.com

Table A.1: Self-use software. They provide due diligence services that quantify patent value. Primarily these self-use software products aggregate publicly available patent data and run algorithms to produce useful patent search and evaluation outputs.

Consulting services	Website
IPVision	ipvis.com
IP Capital Group	ipcg.com
Evalueserve	evalueserve.com
Red Oak IP, LLC	redoakip.com
Ocean Tomo, LLC	oceantomo.com
Red Chalk Group, LLC	redchalk.com
TAEUS	taeus.com
IPGenix	ipgenix.com
Black Stone IP	blackstoneip.com
Chipworks	chipworks.com
3LP Advisors	3lp.com
284 Partners	284partners.com
Questel	questel.com

Table A.2: Consulting companies. Services include due diligence, trade negotiations, patent legal consulting, reverse-engineering and teardown services.

Appendix B

Ethereum selection rational

There are a number of blockchains that could potentially be used in the marketplace for ideas. Ultimately the primary consideration is a flexible smart contract platform that can facilitate trustless execution of arbitrary logic. This is required to implement the trustless sealed bid auctions, Harberger Tax, payment processing and licence registration. The primary considerations when picking a blockchain for this application are:

- developer tooling to make the development approachable, maintainable and high quality,
- ecosystem of libraries and standards such as standard interfaces and stablecoins,
- strong developer community to ensure the code base is maintained into the future and
- high quality documentation and developer experience.

A number of popular blockchain platforms are considered below and compared to provide a full rational into why Ethereum was chosen.

Bitcoin is the original and most well known use of blockchain. Bitcoin does enable a scripting language, called Bitcoin script, that can be used to write basic logic into the blockchain (wiki, 2019). It is, however, very limited and does not offer a Turing complete programming environment capable of providing dynamic, autonomous execution of contract code. It's functionality is limited to the transfer of value from one party to another. As a result, if the Bitcoin blockchain was used, heavy modification to the underlying code base would be required. A relatively recent development is RSK for Bitcoin (Rsk, 2019). This new technology enables the deployment of smart contracts on a side chain that is tied to the Bitcoin network. However RKS is still in its very early days, does not have a strong developer community and does not have extensive tooling built around it.

Added to this, Bitcoin can only facilitate three transactions per second which results in exorbitant transaction fees when network demand is high**?**. As a result, transactions can be slow to process, sometimes taking up to twelve hours or more when the network is under high load. This would make the user experience of the platform unacceptably slow. A solution could be to build a payment channel system or to leverage the Lightning Network. However this would add a lot of complexity to the development process which ultimately makes it not worth it.

Lastly, Bitcoin uses proof-of-work as its consensus algorithm making it inherently inefficient. To secure the Bitcoin network 22 terawatt-hours of energy per year are used (G.F, 2018). This is almost the same as Ireland's yearly energy usage. This is unsustainable and so a more energy efficient solution is required.

IOTA is designed to be more scalable than traditional blockchains, with no theoretical maximum throughput. Additionally, it has no miner fees as there are no miners within the IOTA network. Rather, each node within the network performs the action that a traditional miner would do when processing a transaction. IOTA does not have blocks and as a result, no chain. It is a stream of interlinked transactions that are distributed and stored across the network through a data structure called a Tangle, a form of a directed Directed Acyclic Graph (DAG).

Despite the apparent advantages of IOTA, it does not offer any platform to create smart contracts and as such would require the use of extensive modification to accommodate any form of complex licencing and token exchange settlements. As the marketplace aims to be a proof of principal, IOTA is not considered appropriate due to the implementation complexities associated with using it as the underlying blockchain.

NEO was formally known as Antshares and is seen as the Chinese version of Ethereum by many. Fundamentally, it offers much of the same functionality as Ethereum with full smart contract support. Moreover, NEO's smart contract platform is language agnostic meaning that any Turing complete programming language can compile down to NEO's smart contract bytecode. For example, one can write C++, C# or Javascript smart contracts that are then deployed to the NEO blockchain. NEO uses delegated byzantine fault tolerance (dBFT) as its consensus algorithm. This makes it efficient and fast but this results in the consensus becoming more centralised than other platforms and as such makes it more susceptible to censorship. NEO has a limited English speaking developer base making the interaction with the documentation cumbersome at times. NEO's platform is still under active development and is not considered production ready. The lack of tooling, documentation and community size make it unsuitable for the proposed marketplace.

Next generation blockchains are constructions that are set to launch in 2020 that could in theory provide the required functionality needed for the proposed marketplace. These include Algorand, Aergo, Insolar, Cardano and Nano. While these projects all show promise at providing powerful scalable blockchain systems, none of them are in a state of their development that is sufficiently mature to use for the proposed project. All of them still need to prove themselves in the real world before being deemed appropriate.

Ethereum is a blockchain platform offering Turing complete smart contract execution. It has the largest developer mindshare of all the blockchain platforms in the ecosystem by an order of magnitude. Ethereum is currently used in more smart contract related projects than any other blockchain, with 96 out of the top 100 token based applications running on the Ethereum blockchain (Live Coin Watch, 2019).

Ethereum is currently using proof-of-work, but will soon move to proof-of-stake under the Casper update with the launch of Eth 2.0 in the next 2 years (Vitalik Buterin, 2018). This will make Ethereum far more scalable and efficient. Ethereum has extensive documentation and a wide range of community built and maintained tools.

Due to the simplicity of implementation, the ability to write Turing complete smart contracts, the future sustainability due to developer mind share, true decentralization and current implemented standards, Ethereum is chosen as the blockchain of choice for this thesis.

It is important to note that ideas discussed in this paper can be applied to using any Distributed Ledger Technology(DLT) or blockchain; this does not have to be in the form of Ethereum. However, many components would need to be rebuilt within another blockchain framework. For example, having a native stablecoin within Ethereum (Dai) makes building applications that use payments far easier. If built on another blockchain this becomes considerably more difficult as every element of the application needs to be built in-house.

Appendix C

Alternative sealed bid constructions

The proposed marketplaces uses sealed bid auctions over a public English outcry auction to:

- prevent shill bidding,
- prevent bid sniping,
- prevent bidder collusion in the formation of bidding rings and
- preserve bidders private valuation and strategy.

The mechanism proposed utilizes a commit reveal mechanism which has strong privacy guarantees. However, this mechanism introduces a liveness trade off due to the asynchronous nature of the reveal process which requires bidder interactivity. This means that bidders could loose their bids if they do not return to the platform during the reveal phase. Additionally, a user could loose their bid if they clear their web browser cache and forget their bid amount and salt.

While these disadvantages are quite severe, the mechanism is simple to understand and implement. There are a number of other mechanisms proposed in the literature that aim to provide sealed bid auction mechanisms without the same liveness trade offs.

Galal and Youssef (2019) propose a mechanism that is private, verifiable, fair and non-interactive based on zero knowledge cryptography and the Ethereum blockchain. All of these properties are present in a commit reveal scheme, except for the interactive element which is required in the reveal phase. Galal and Youssef (2019)'s mechanism involves bidders submitting homomorphic commitments of their sealed bids to a smart contract. Bidders then reveal their bids to an auctioneer via a public key encryption scheme. The auctioneer then determines the winner of the auction. An interactive zero knowledge proof protocol is then used between the smart contract and auctioneer to verify the correctness of the auctioneer's claim. This mechanism is partially privacy preserving in that no bidder information leaked by the bidders but the auctioneer can still leak information to other bidders during the bid phase. Additionally, this technique requires a trusted setup to be performed to orchestrate the zero knowledge proof system.

Parkes et al. (2008) proposes a system that leverages Time-Lapse Cryptography (TLC) which prevents ciphertext decryption until a set point in the future. This is used to keep the bids *sealed* until the end of the auction, preventing any information leakage. In theory, this is an excellent way to achieve the goals of preventing any manipulation or corruption. However, the TLC mechanism presented in their paper requires the use of a trusted time server in the operation of the TLC. This introduces a trust assumption which could be acceptable in some cases but, by its definition, is not trustless and outcome verification becomes impossible; there is no way for an observer to know that the trusted time server did not leak information to a competing bidder during the bidder phase. This trust assumption therefore shifts the opportunity of manipulation away from the auctioneer and to a third party.

Ultimately, the mechanism used in this thesis is simply a modular component to facilitate trustless sealed bid auctions. Future implementations could exchange this mechanism for a more robust, non-interactive process. However, for the case of this proof of concept, the commit reveal mechanism is simple and can be easily implemented while having strong security and privacy guarantees.

Sealed bid auctions with commit reveal encryption

Another option to replacing the commit reveal mechanism is to improve its downsides. The primary downside is that the user looses their bid and salt amount if they clear their browser which will make it impossible to reveal their bid. One solution to this is to encrypt the users bid amount and salt and store the encrypted data on the blockchain. When revealing their vote they first download the encrypted bid and salt, decrypt it and then use it to vote. In this construction the bidder encrypts their bid and salt using a Pretty Good privacy(PGP) encryption key paring. First, the bidder generates a private encryption key using a signature generated from their Ethereum wallet. This uses their wallet private key entropy to create the encryption key, without exposing the wallet private key to the browser. This makes it far more secure and less invasive. Additionally, most wallets don't let the user expose their private key in any case as this is bad practice.

When the user comes to revealing they can simply rederive their encryption private key from a fresh wallet signature. This keeps the private key safe at all times and enables the user to recover their salt and bid amount.

Appendix D

Dai stablecoin selection rational

Stablecoins are used as the defacto payment mechanism for all settlement within the proposed marketplace. There are a number of different stablecoins that can be chosen from and the design proposed by no means dictates which stablecoin *must* be used. In fact any ERC20 token could be used as a payment token within the platform. However, the use of a stablecoin is strongly recommended as other crypto tokens have been known to be extremely volatile.

There are a number of different stablecoins within the Ethereum ecosystem. Their primary differences are in how collateralization works to maintain token stability. Most stablecoins can ultimately be broken up into two main kinds: fiat-collateralized and crypto collateralized.

Fiat-collateralized stablecoins encompass most of the stablecoins on the market and work by a company issuing a token which is backed 1:1 with an equal amount of fiat currency in a bank account. The issuing company is responsible for maintaining appropriate reserves against their issued token. This means that users need to trust that the issuer has not misreported their collateralization by claiming to have more currency in the bank than what they actually have. In theory the stable crypto currency is redeemable for underlying fiat currency from the issuer which helps maintain the peg.

Two of the most common fiat-collateralized stablecoins on Ethereum are:

- Tether ($USDT) is issued by a Hong Kong based company, Tether Limited. Tether has been under a lot of scrutiny due to the lack of transparency of their fiat holdings which has lead investors to believe that it has been undercollateralized. The lack of clarity on Tether's solvency removes it from the options for the payment token for the marketplace.

- UDC Coin ($USDC) is issued by CENTRE, a joint venture between Circle and Coinbase. It is fully collateralized and open source and operates within US money transmission laws. It works exclusively with established banks and auditors. Because USDC is backed by large banks, open source and transparent and is commonly viewed as safe within the crypto community it is a contender for the stablecoin offered within the market.

Crypto-collateralized stablecoins work by locking cryptocurrency in a smart contract as collateral when issuing the stablecoin. In the case of MakerDao's Dai token, Ether and

BAT tokens can be used as collateral to generate Dai by network participants. These deposits are required to be over collatatralized by a minimum threshold of 150%, or a position is liquidated.

For example if you want to generate 100 Dai you would need to collateralize the position with a minimum of 150 USD worth of Ether, based on the current Ether price. For example say that Ether is worth 100 USD. This means you would need to lock away a miniumim of 1.5 Ether to generate the 100 Dai. If you lock away 2 Ether then your position is 200% over collateralized and safe from liquidation. If at any point the price of Ether drops such that the value of the Ether in the vault is less than 150% of the outstanding debt of 100 Dai then the position is liquidated with a 13% penalty. i.e if the 2 Ether is worth *less* than $150 the position will be liquidated. This mechanism incentives users that issue Dai to always keep their reserves sufficiently backed.

The main attraction to Dai is that there is no centralized point of failure that can take the system offline. Other stablecoin systems, even those backed by credible institutions like USDC, can be stopped, corrupted or shut down by government regulation or company decisions. However, Dai is unstoppable in the sense that there is no regulatory oversight that can impede its growth or usage and thus no one can stop it.

One Major criticism of crypto collateralized stablecoins is stabilitity. Dai was first launched when the price of Ether was over 1400 USD and during its life it has withstood the price of Ether falling all the way to 70 USD. During this time the Dai to USD peg has remained fixed with deviations not exceeding more than 5 cents in either direction. This is a remarkable achievement and is a testament to the Dai's stable mechanism.

Due to its trustless nature and historic price stability Dai is the stablecoin of choice for the marketplace for ideas. However, it is important to note that future implementations can very easily incorporate multiple payment tokens. If there is a demand for USDC or any other ERC20 token the market could offer users this option.

Lastly, front end integration with exchanges like Uniswap.io could enable direct atomic swaps from one payment token to another such that users could use multiple tokens which are then always converted to the same base currency in the backend.

Appendix E

Sequence figure key

The Figure below is the key for the life cycles presented in Figure 6.5, 6.6 and 6.7.

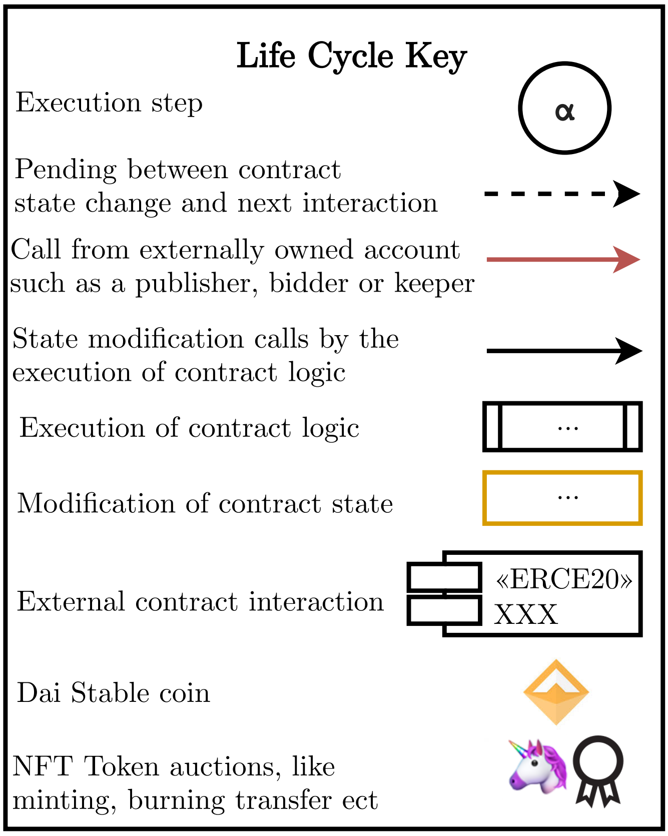


Figure E.1: Sequence Figure key

Appendix F

Smart contract sequence diagrams

Each of the primary functions discussed in Section 6.2.2 are now broken down with a sequence Figure. These Figures showcase the interconnection of the 7 smart contracts within the marketplace. At each point in the Figure a rectangle represents where the contract logic is executing. For example in F.10 the `NewUser` calls the `registerUser` function within the `UnicoinRegistry`. This function then calls another internal function from the `userManager`, `_registerUser`. Along this execution a rectangle represents where the logic is being called from.

At appropriate points additional comment is shown next to rectangles notating the associated state change or input validation. When needed, If statements are defined using an ”alt” block. These blocks encompass a block of code execution based on the if statement.

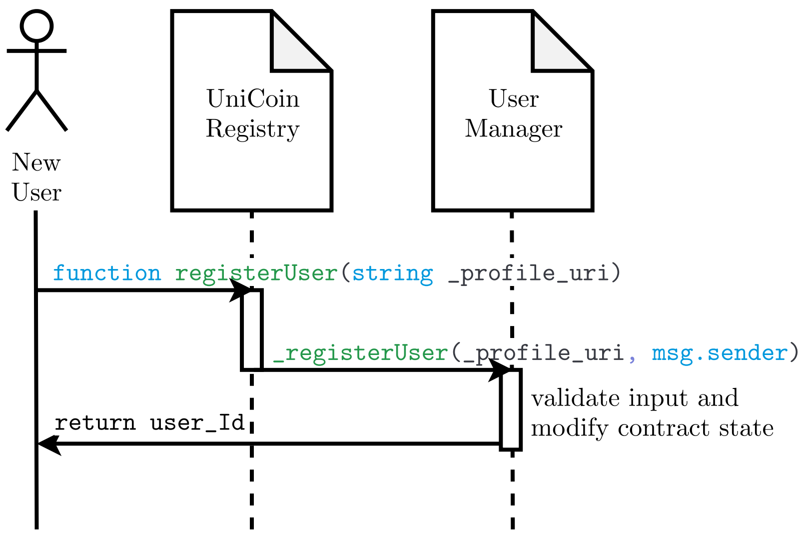


Figure F.1: User registration adds user account details to the user manager.

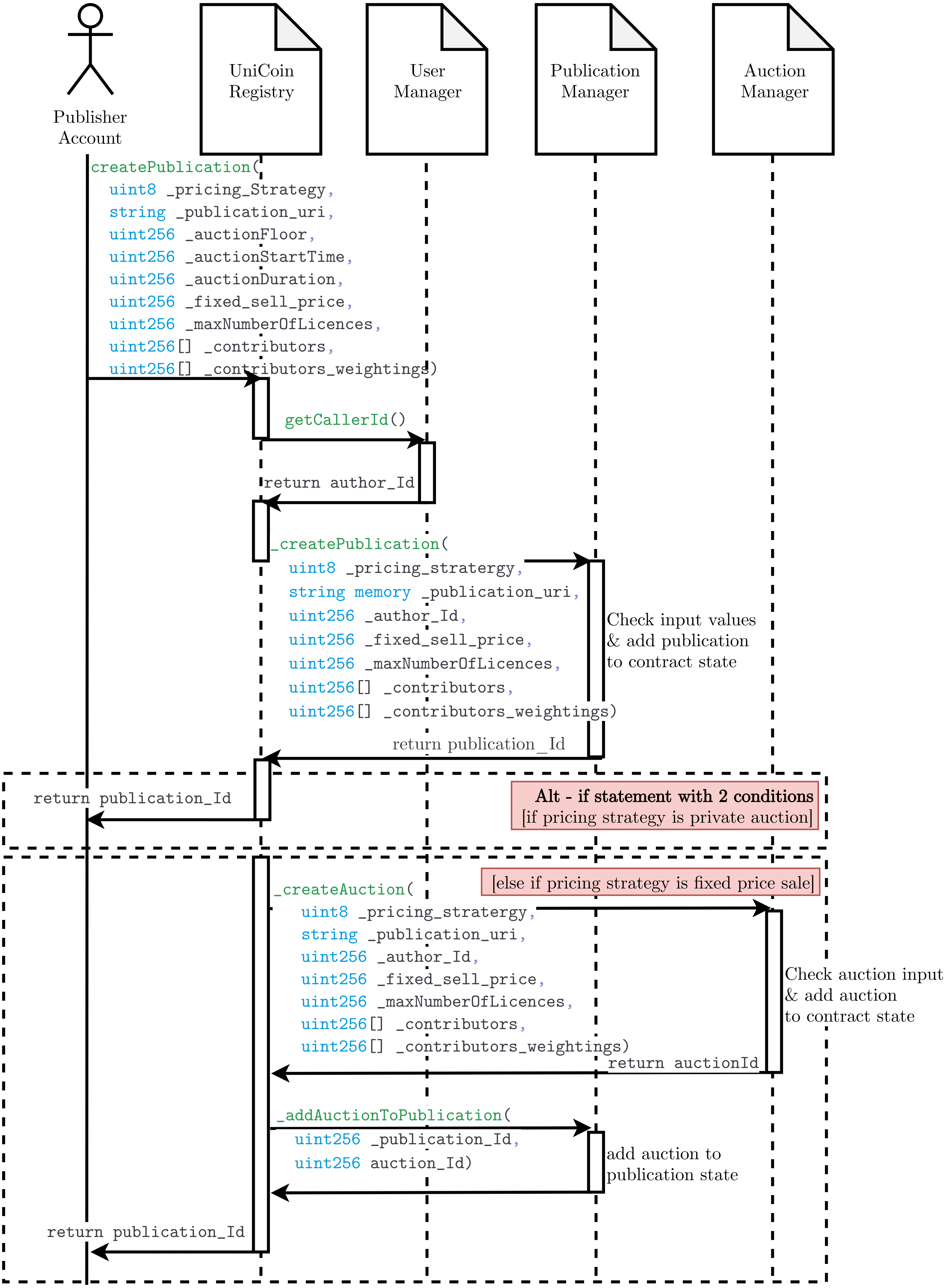


Figure F.2: Publication listing including a logical statement that conditionally creates an auction in the case of a private auction pricing strategy.

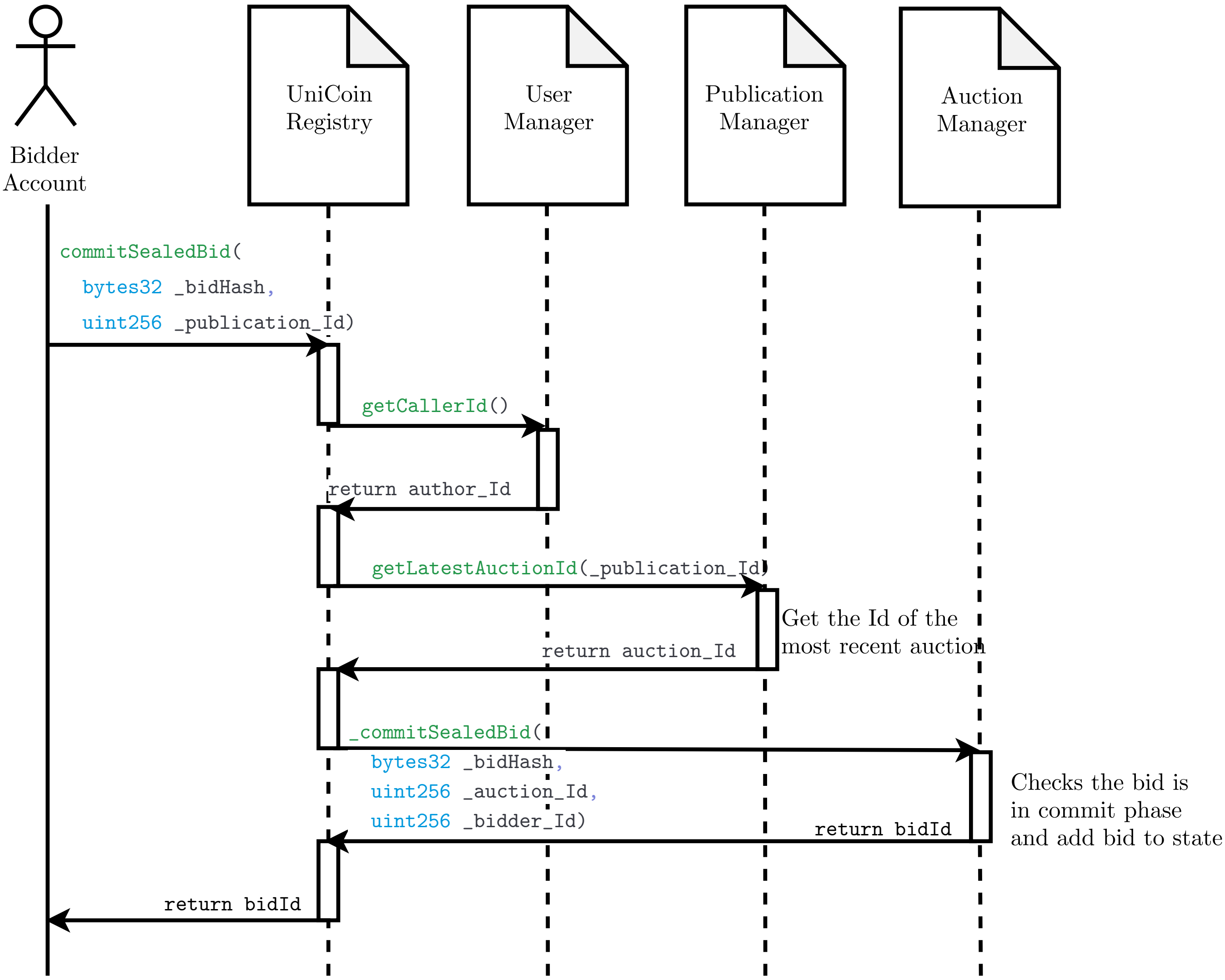


Figure F.3: Committing sealed bid involves submitting the bid hash and publication Id. The bid is added to the bid auction for the specific publication.

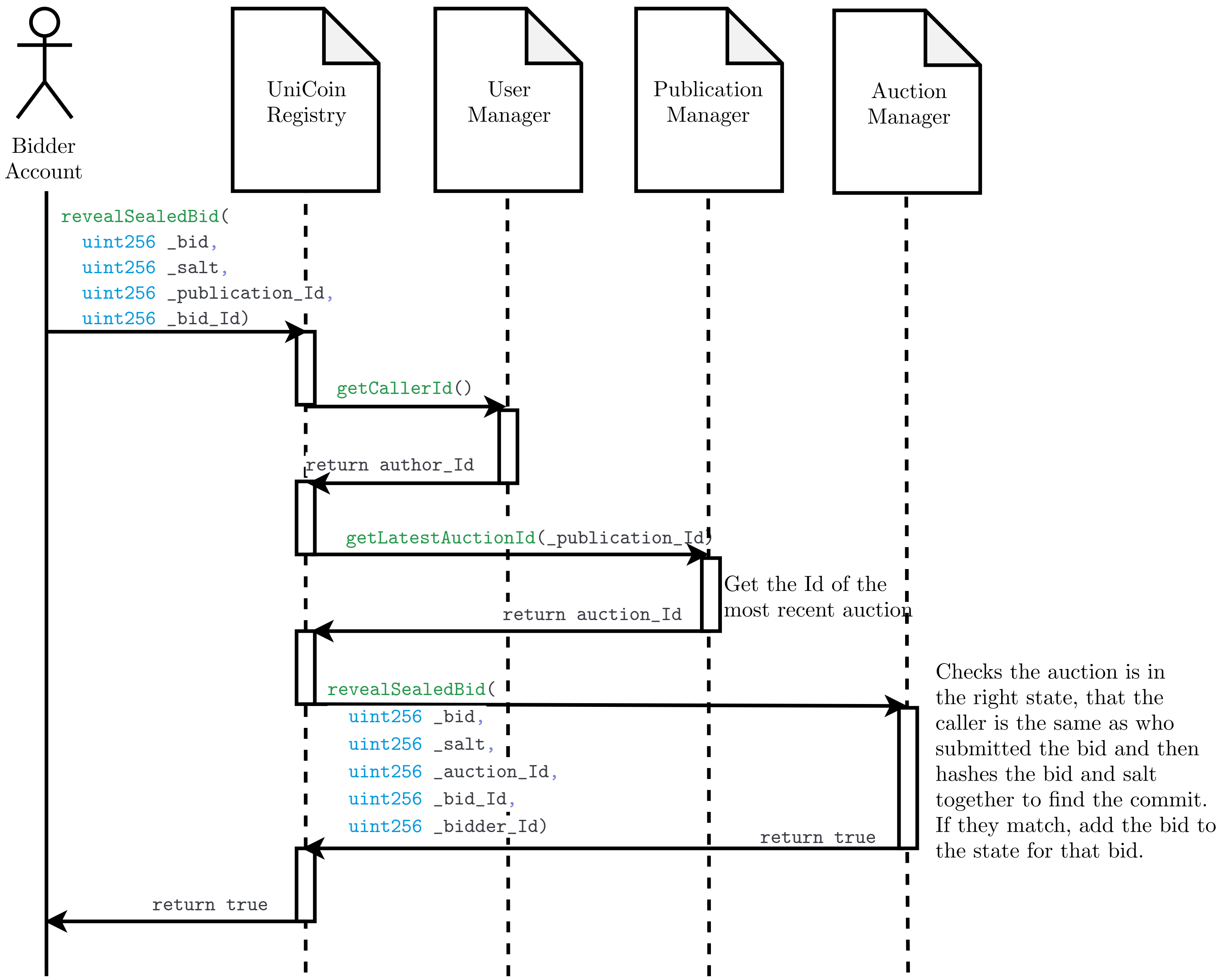


Figure F.4: Bid reveal involves the contract checking that the hash of the bid amount and the salt equal the committed value specified during the commit phase. If they match the bid value is added to the contract state.

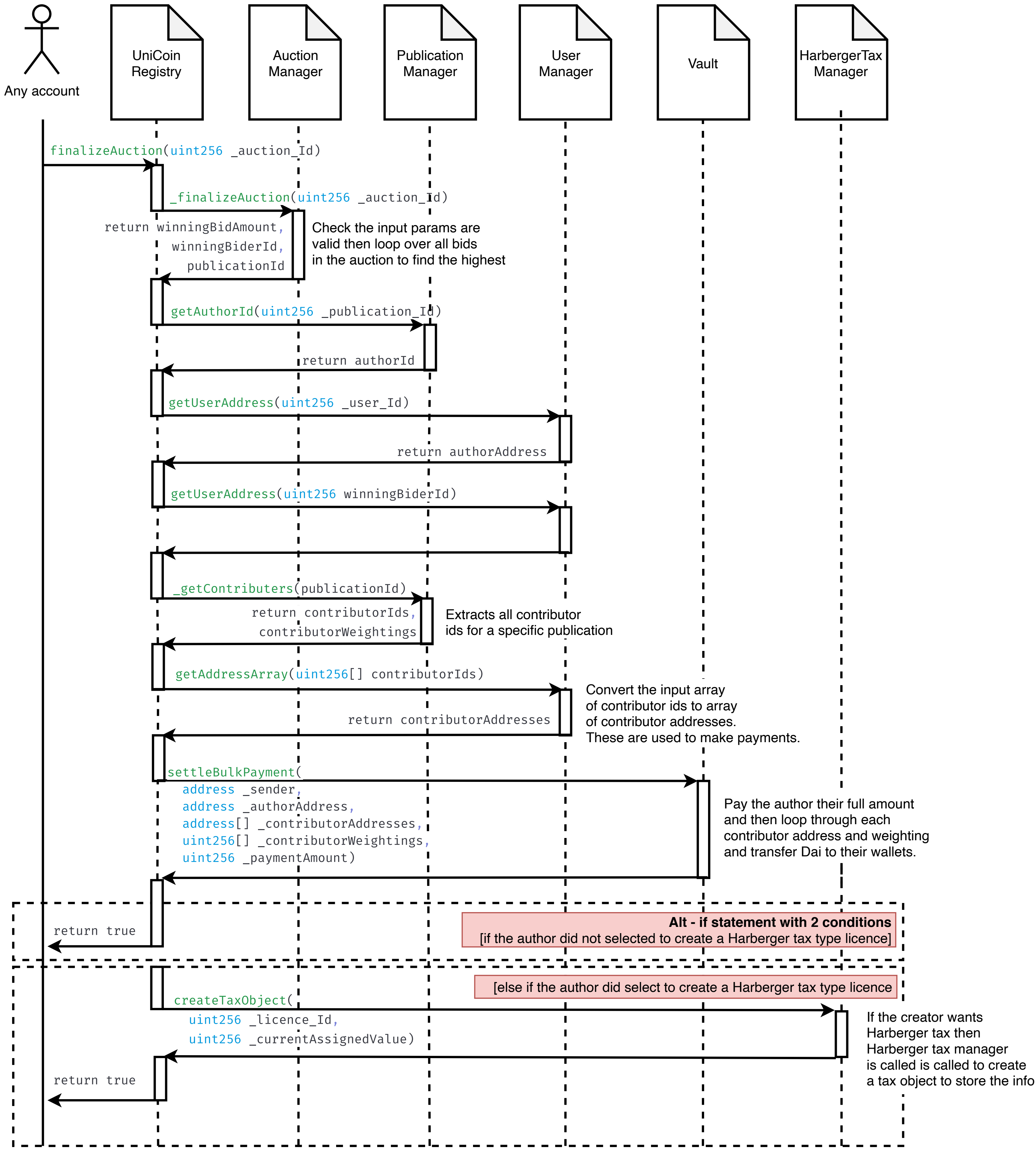


Figure F.5: Auction finalization involves looping through all revealed bids for the auction and identifying the winner. The winner payment is then split amounts all contributors. If the publisher chose to use a Harberger Tax model, this is added to the licence.

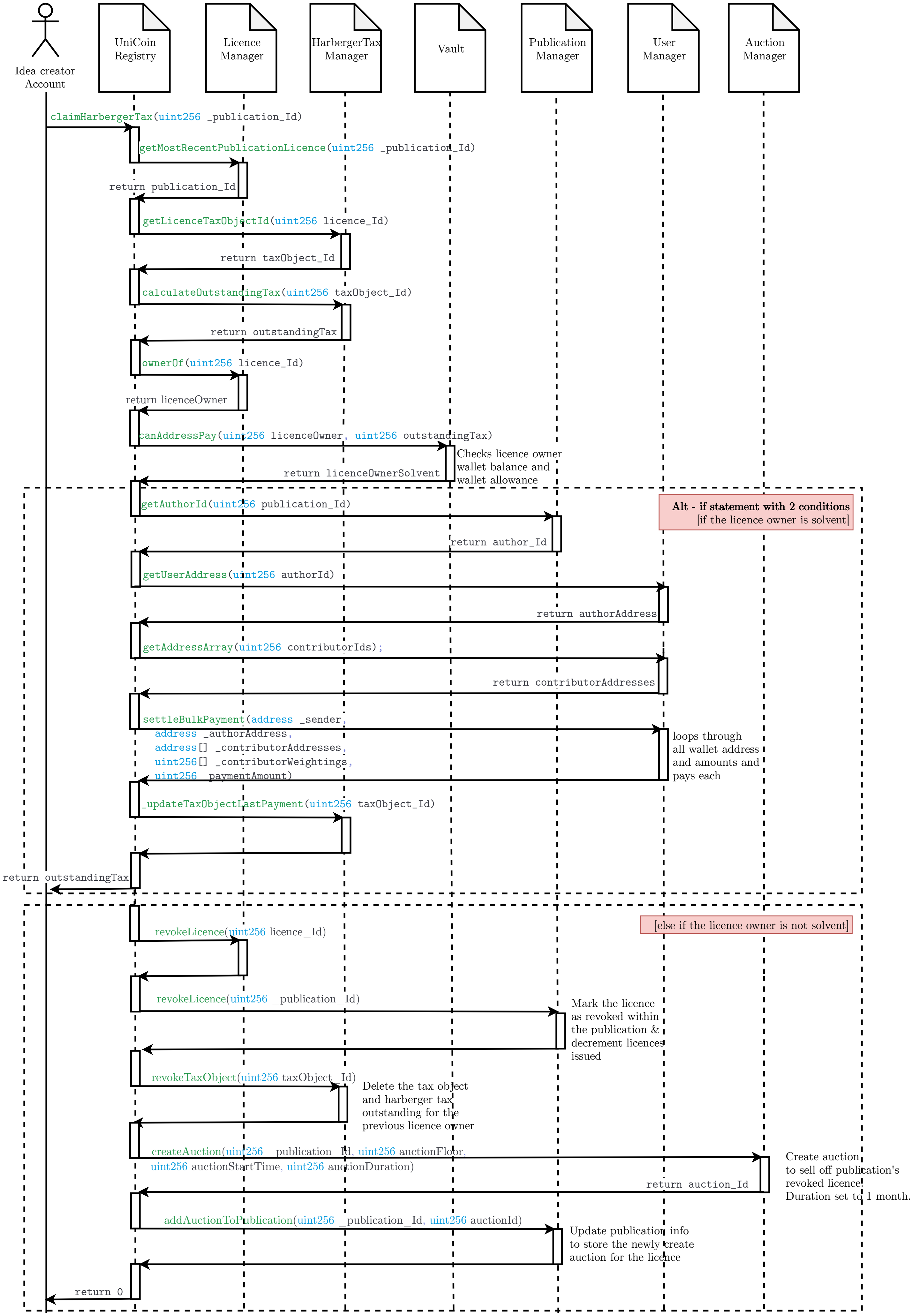


Figure F.7: Claiming Harberger Tax either involves: 1) the licence owner has enough Dai and can pay (solvent) then all contributors are paid and Harberger object is updated or 2) the licence owner does not have enough Dai (not solvent) and the licence is taken away from them and listed for public auction. The auction duration starts at 30 days.

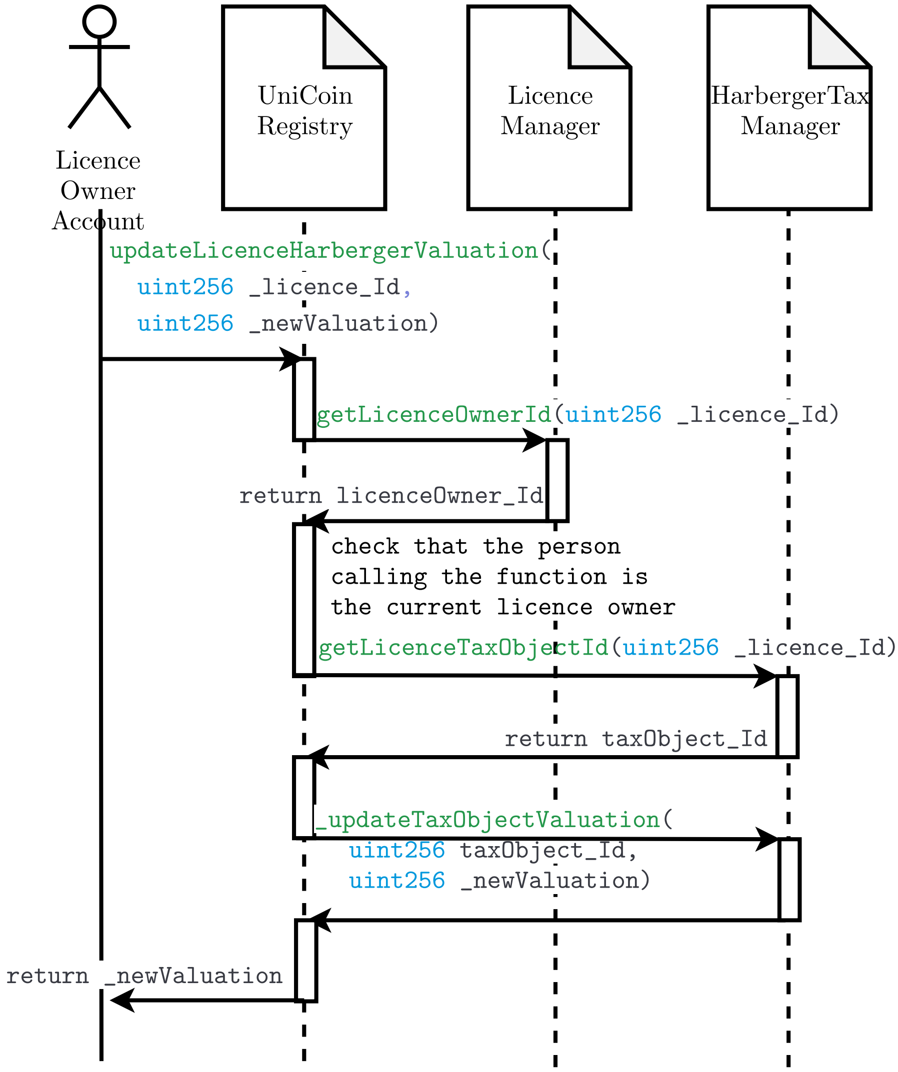


Figure F.6: Updating licensee private valuation involves specifying the licence ID and the new valuation. The registry calls the licence manager and Harberger Tax manager accordingly to update the values. A licence holder would do this if they were updating in the case of a buy out or if they deem the licence to be worth more or less.

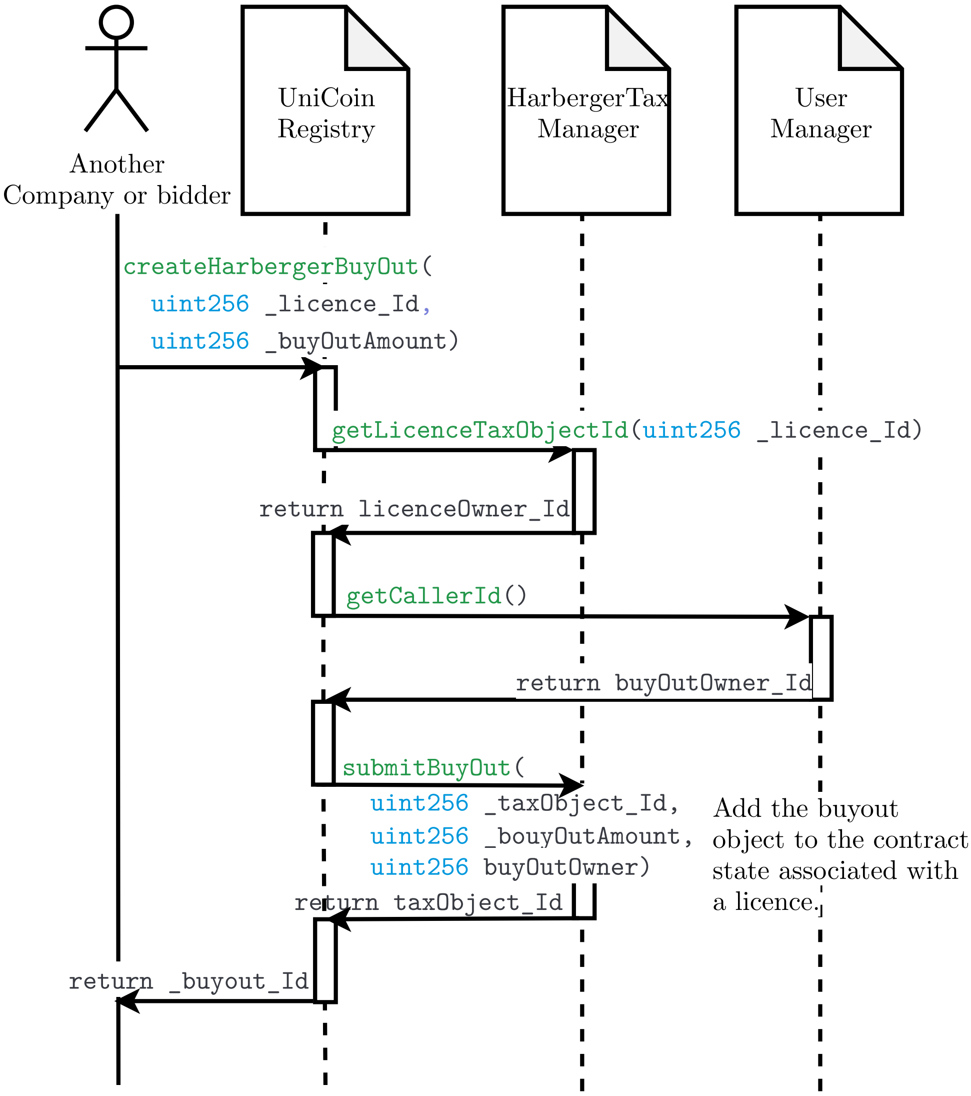


Figure F.8: Creating Harberger Tax buying can be done by anyone at any point in time. To create this the user specifies the licence they want to buyout along with their buy out amount. This amount is how much the buyer is willing to pay to take the licence.

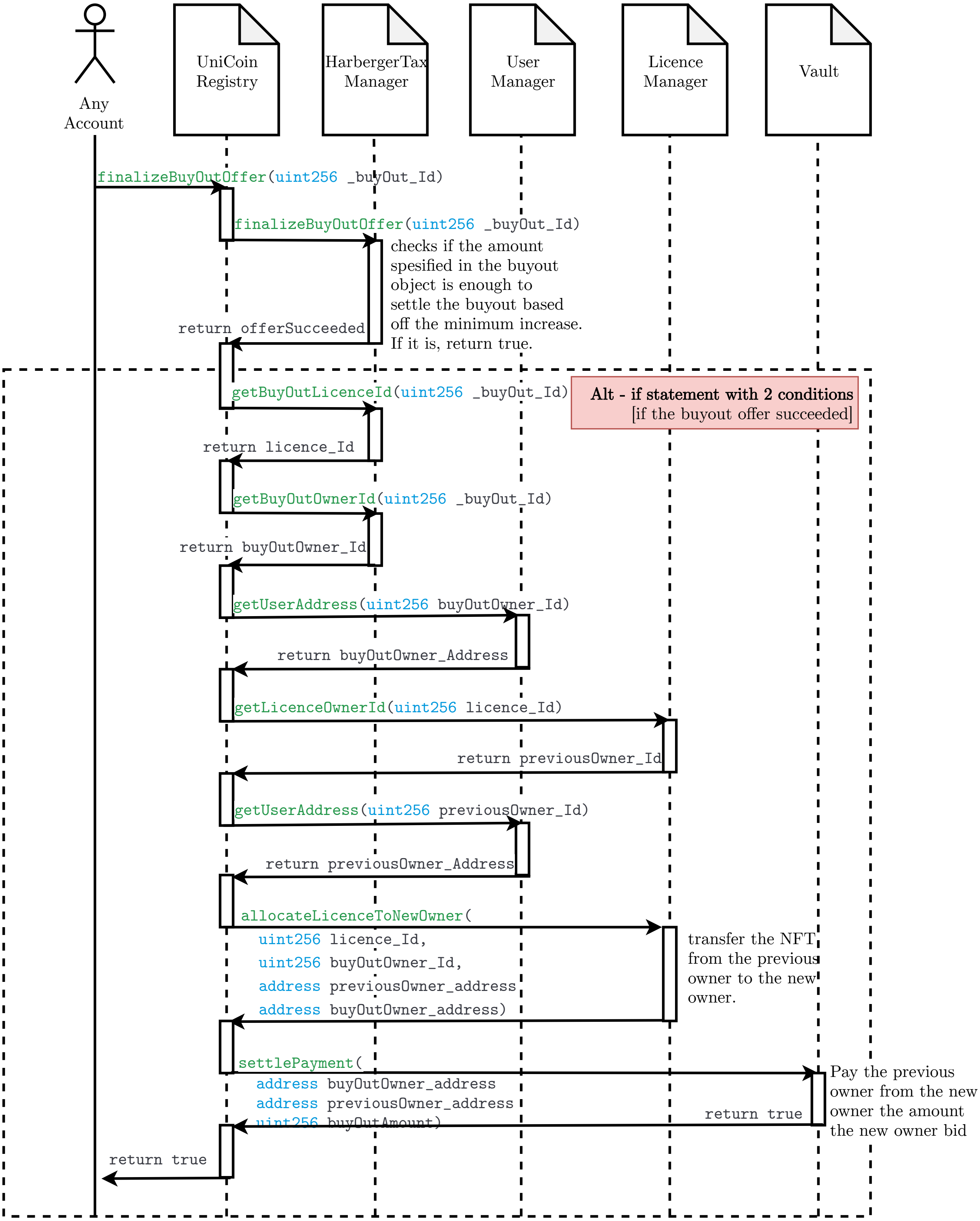


Figure F.9: Buyout finalization involves the validity of the buyout claim being checked to ensure that the buyout is more than the minimum threshold for that specific licence. If it is, then the licence is transferred away from the old owner and is allocated to the new owner.

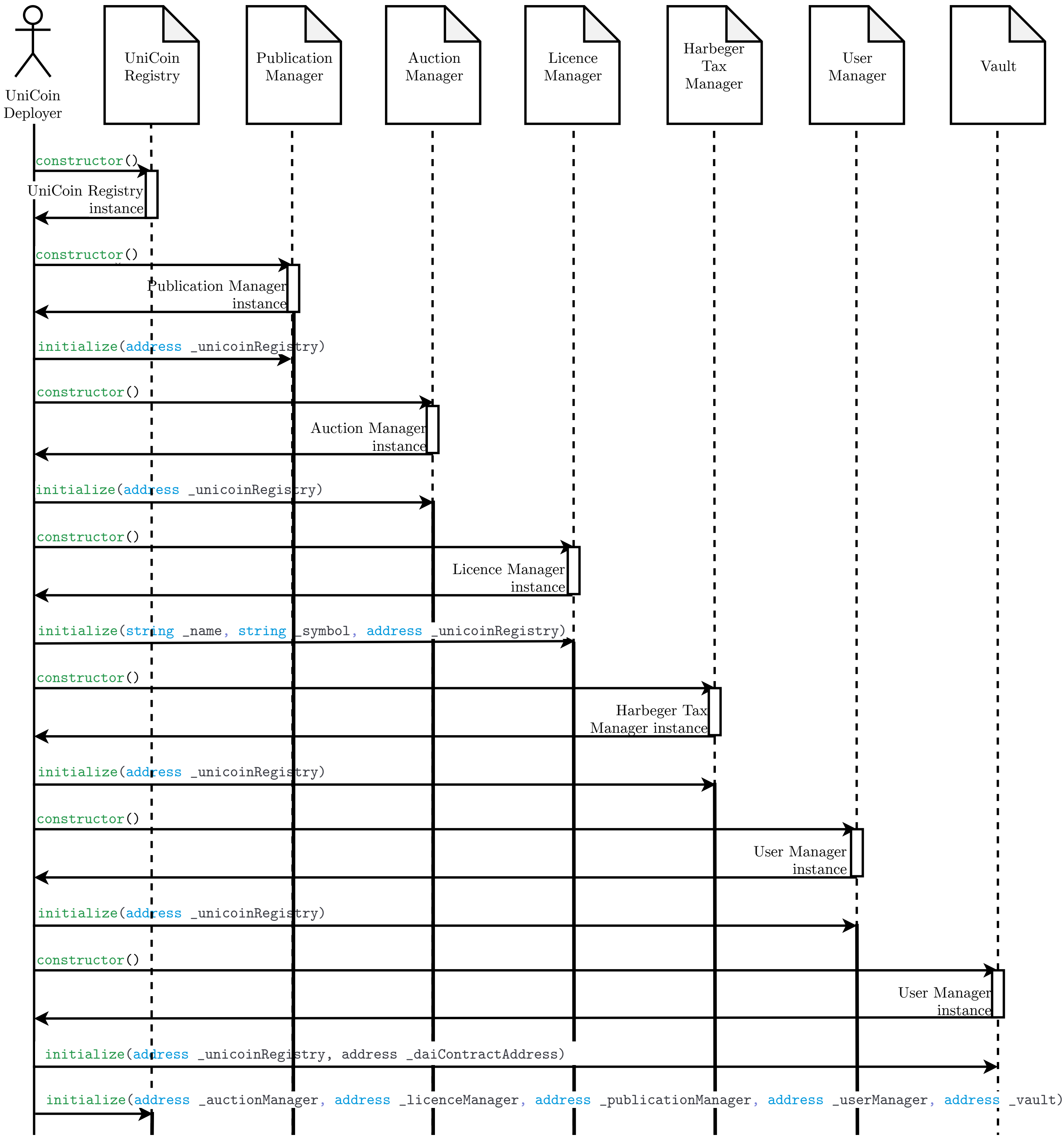


Figure F.10: Deployment flow of the smart contracts involves deploying each spoke and then initializing them with the UniCoin Registry address (hub). After all spokes are deployed the Registry itself is initialized with all spoke addresses. Some spokes require special parameters like the Dai contract address or additional information like the NFT token name and ticker for the Licence Manager.

Appendix G

Smart contract tests

The smart contracts were thoroughly tested in javascript using the Truffle framework. Each core function has both a positive affirmation test to ensure that it does the correct execution as well as a negative test to ensure that it rejects invalid inputs.

See the Github repo here to run the tests locally on your computer. The output of the tests is shown below. Each section encompasses a set of functionality within the tests.

```
Contract: HarbergerTaxManager: Math 🔬
    Helper Maths 💪
      ✓ Exponentual calculation loop (494ms)
      ✓ Can correctly find capatalization function (92ms)
      ✓ Can correctly calculate future value of NACC position (72ms)

  Contract: Unicoin Registry Full system test 🧪🔬
    User Management 👷‍♂️
      ✓ Reverts if invalid user input (88ms)
      ✓ Can add new user (480ms)
      ✓ Revert if user already added (76ms)
      ✓ Can retrieve user profile information (116ms)
    Publication Management: Creation 📚
      ✓ Can create a valid publication: FixedRate (115ms)
      ✓ Can create a valid publication: Private Auction (142ms)
    Publication Management: Reject on negative input 👷‍♂️
      ✓ Reverts if invalid publication: input non-registered user (112ms)
      ✓ Reverts if invalid publication: input URI too short (169ms)
      ✓ Reverts if invalid publication: invalid Fixed price call (163ms)
      ✓ Reverts if invalid publication: invalid auction call (auction should not have fixed price) (161ms)
      ✓ Reverts if invalid publication(auction): invalid start time (216ms)
      ✓ Reverts if invalid publication(auction): invalid duration time (246ms)
    Publication Management: retrieve publication information 🔎
      ✓ Correctly get publication length (59ms)
      ✓ Correctly getPublication auctions (122ms)
      ✓ Correctly get publication information (191ms)
    Auction Bidding: Commit stage 🤝
      ✓ Reverts if invalid bid time (86ms)
      ✓ Auction Status set correctly (236ms)
      ✓ Place bid in auction (381ms)
    Auction Bidding: retrieve bid information 🔎
      ✓ Can retrieve bids for publication (73ms)
      ✓ Can retrieve bid Id information (173ms)
      ✓ Can retrieve bider information (78ms)
      ✓ Can retrieve the number of publication auction bids (62ms)
      ✓ Can retrieve bid information (108ms)
    Auction Bidding: Reveal stage 💴
      ✓ Can correctly reveal bid (211ms)
      ✓ Revert if wrong user tries to reveal bid (186ms)
      ✓ Revert if wrong bid committed (both bid and salt) (186ms)
      ✓ Can correctly reveal 2nd bid (241ms)
      ✓ Revert if wrong time to reveal bid (238ms)
    Auction Bidding: finalize auction 🏁
      ✓ Can finalize auction correctly (637ms)
      ✓ Reverts if invalid reveal time: before auction (103ms)
      ✓ Reverts if invalid reveal time: during auction (149ms)
    Auction Bidding: retrieve finalized bid information 🔎
      ✓ Can get information about loosing bid (66ms)
      ✓ Can get information about winning bid (76ms)
    Licence Management: retrieve issued licence information 📝
      ✓ Can get information about issued licence (69ms)
      ✓ Can get licenses associated with a publication (59ms)
      ✓ Should be no licences associated with other publications (62ms)
    Licence Management: Interacting with NFT 💼
      ✓ Can get information about issued NFT (63ms)
      ✓ Can get info about a publication's licence (91ms)
    Harberger Tax: publication management 📕
      ✓ Can correctly create a licence with harberger tax enabled (259ms)
      ✓ Can bid (and win) auction on harberger tax (765ms)
      ✓ Can get taxObjects length (55ms)
      ✓ Can get info about tax object (70ms)
      ✓ Can get licenses associated with harberger publication (95ms)
    Harberger Tax: taxation calculations and payments 🤑
      ✓ Can correctly calculate outstanding tax (85ms)
      ✓ Can correctly calculate min buy out price (65ms)
    Harberger Tax: claiming claimHarbergerTax 👮‍♂️
      ✓ correctly distributes tax to recipients (488ms)
      ✓ correctly updates tax object after paying tax (95ms)
    Harberger Tax: buyout creation 💳
      ✓ Can correctly check if a licence has harberger tax (88ms)
      ✓ Can correctly create a valid buyout request (267ms)
      ✓ Reverts if not a harberger tax licence (92ms)
      ✓ Reverts if value sent less than min buyout price (117ms)
    Harberger Tax: buyout finilization 🏎
      ✓ reverts if invalid finilization time (119ms)
      ✓ can finalize buyout (291ms)
      ✓ reverts if buyout already finalized (108ms)
      ✓ correctly updates buyout information (62ms)
      ✓ correctly updates tax object information (78ms)
      ✓ correctly updates licence information (57ms)
    Harberger Tax: solvent licence owner 🖋
      ✓ places licence for auction if owner is underfunded (641ms)
      ✓ can bid on auction and win new licence (926ms)


  62 passing (20s)

✨  Done in 39.77s.
```

Appendix H

Vue selection rational

There are numerous Javascript frameworks available for building modern, responsive applications. Three of the most popular framworks (Angular, React and Vue) are compared in Table H.1 below. Ultimately any Javascript framework would be able to build the proposed marketplace. What ultimately matters is picking a framework that is:

1. *Future proof*: the platform chosen should continue to be supported into the future to ensure the application built continues to be supported.
2. *Well documented, easy to use and has strong tooling*: developer experience matters. Picking a framework that is easy to understand and that has good developer tooling will ensure that others can continue to build on the platform.
3. *Open source and free*: the ideology of the Javascript framework needs to align with the overall application. Using a closed source framework is therefore inappropriate in this context. This will also help to ensure that the framework is maintained into the future as open source software attracts developers from all around the world.

	Angular	**React**	**Vue**
Type	A Framework	Library to build UI	A Framework
Why Choose	If you want to use TypeScript	If you want to go for "everything-is-JavaScript" approach	Easy JavaScript and HTML
Founders	Powered by Google	Maintained by Facebook	Created by Former Google Employee
Initial Release	September 2016	March 2013	February 2014
Application Types	If you want to develop Native apps, hybrid apps, and web apps	If you want to develop SPA and mobile apps	Advanced SPA and started supporting Native apps
Ideal for	If you want to focus on large-scale, feature-rich applications	Suitable for modern web development and native-rendered apps for iOS and Android	Ideal for web development and single-page applications
Learning Curve	A steep learning curve	A little bit easier than Angular	A small learning curve
Developer-friendly	If you want to use the structure-based framework	If you want to have flexibility in the development environment	If you want to have separation of concerns
Model	Based on Model-View-Controller	Based on Virtual Document Object Model	Based on Virtual Document Object Model
Written in	TypeScript	JavaScript	JavaScript
Community Support	A large community of developers and supporters	Facebook developers community	open-source project sponsored through crowd-sourcing
Language Preference	Recommends the use of TypeScript	Recommends the use of JSX – JavaScript XML	HTML templates and JavaScript
Popularity	Widely popular among developers	More than 27,000 stars added over the year	More than 40,000 stars added on GitHub during the year
Companies Using	Used by Google, Forbes, Wix, and weather.com	Used by Facebook, Uber, Netflix, Twitter, Reddit, Paypal, Walmart, and others	Used by Alibaba, Baidu, GitLab, and other

Table H.1: Table comparing diffrent properties of Angular, React and Vue

Appendix I

Marketplace screenshots

This appendix contains a number of screenshots that showcase the core functionality of the Marketplace. The Marketplace can also be found online at unicoin.win. Each screenshot has an accompanying description to outline what is being shown along a user story. This user story encompases the full flow and interaction with the market from loading the website, creating an account, publishing a paper to the market and finally with a company bidding and winning in an auction.

Dapp home page. Provides high level information into operation of the marketplace and briefly explains how the system works through a timeline.

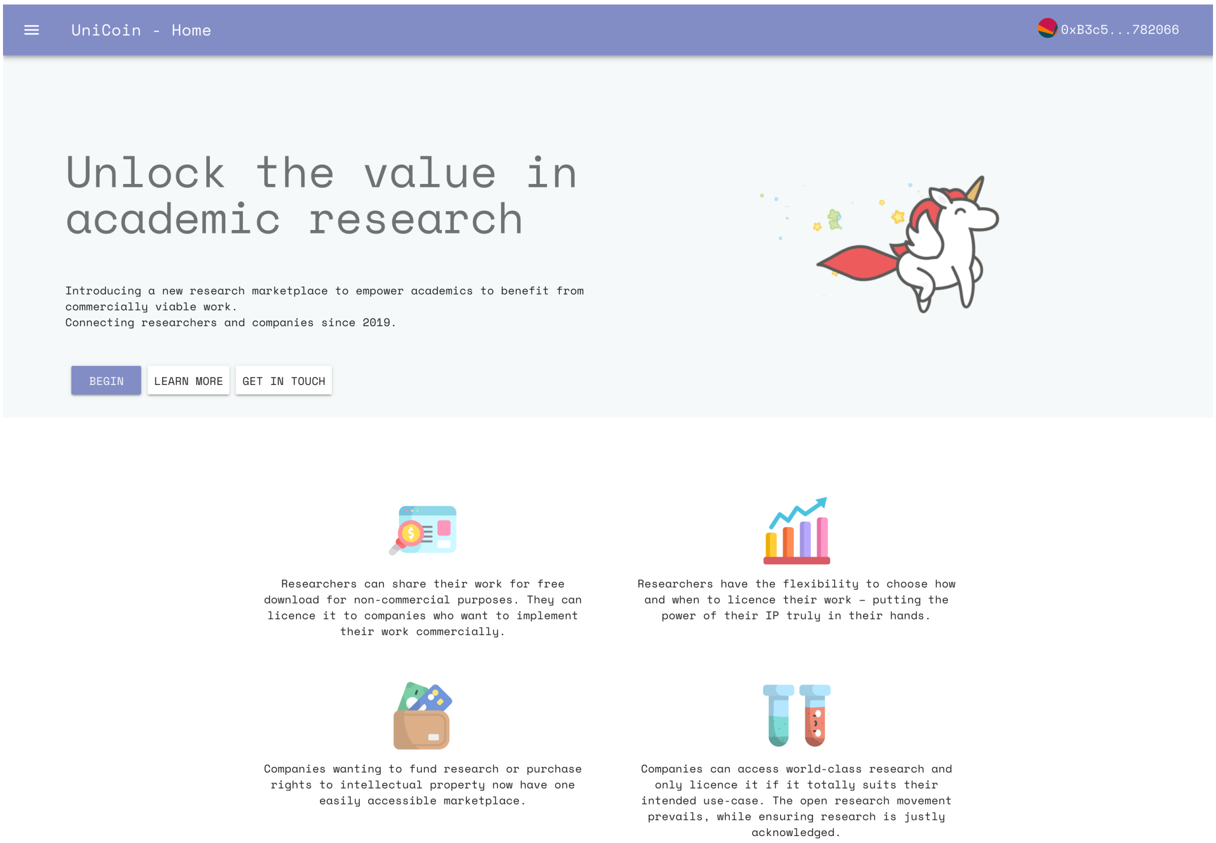


Figure I.1: Home page. Includes a short description of functionality. Notice that top right has the users wallet address. This is automatically loaded from their web3 wallet.

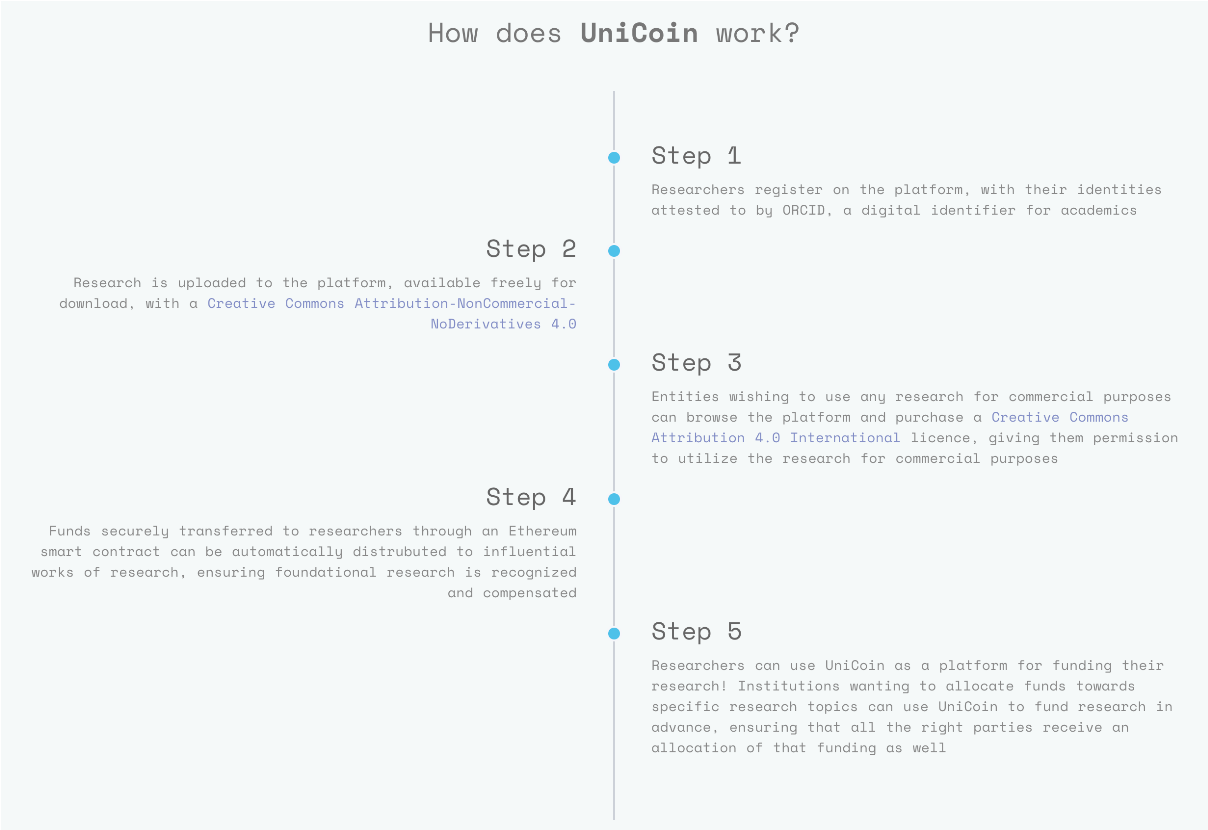


Figure I.2: Home page. Timeline of the life cycle of a publication on the platform.

Figure I.3: Home page. Expandable description of the diffrent technologies used in the market. When expanded, shows additional details about the technology stack.

Create an account. All users are required to register. This involves making a json object in their web browser which is then uploaded to IPFS and registered in the smart contract. For an academic, this registertration process includes fetching their ORCID ID via OAuth. This is shown in the screenshots that follow. For a company a similar process would be followed except they would specify company related information such as company name, location, contact details and website (screenshot not shown for brevity).

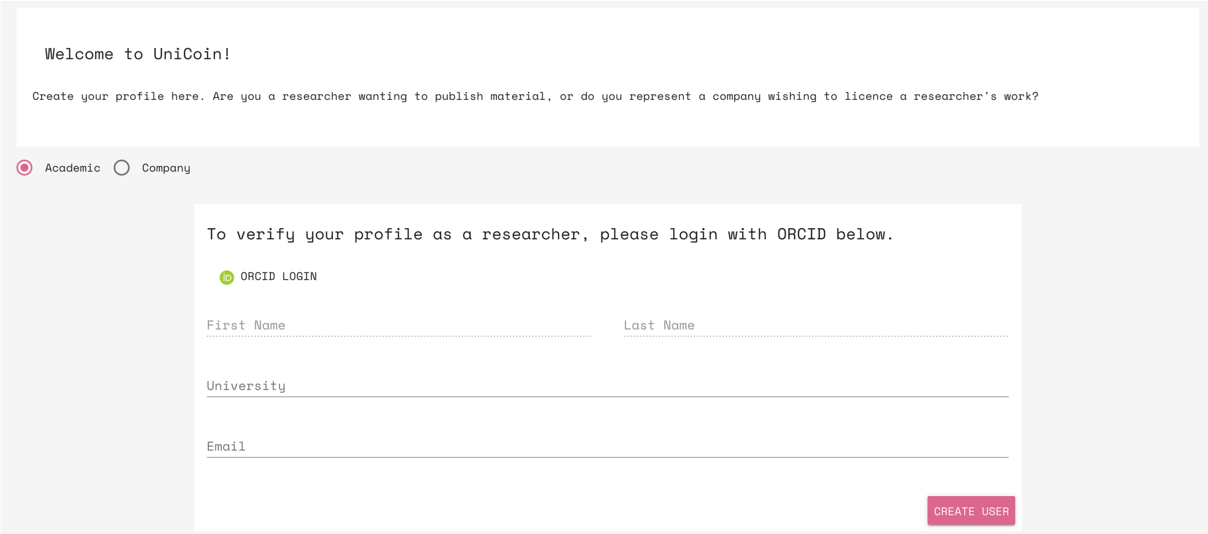


Figure I.4: Create an account. Two main options are either for an academic who intends to publish within the market or a company who wants to buy licences from the market. The first name and last name fields are disabled as these must be populated from an academics ORCID account.

Figure I.5: Once the user clicks on the "ORCID LOGIN" button they are re-routed to the ORCID website where they are able to login using their institution username and password. In this screenshot the University of Cape Town is selected. ORCID enables any academic from any institution to create a profile. There is no additional on boarding required from institutions to enable their academics to make an account as all major universities are already set up with this system.

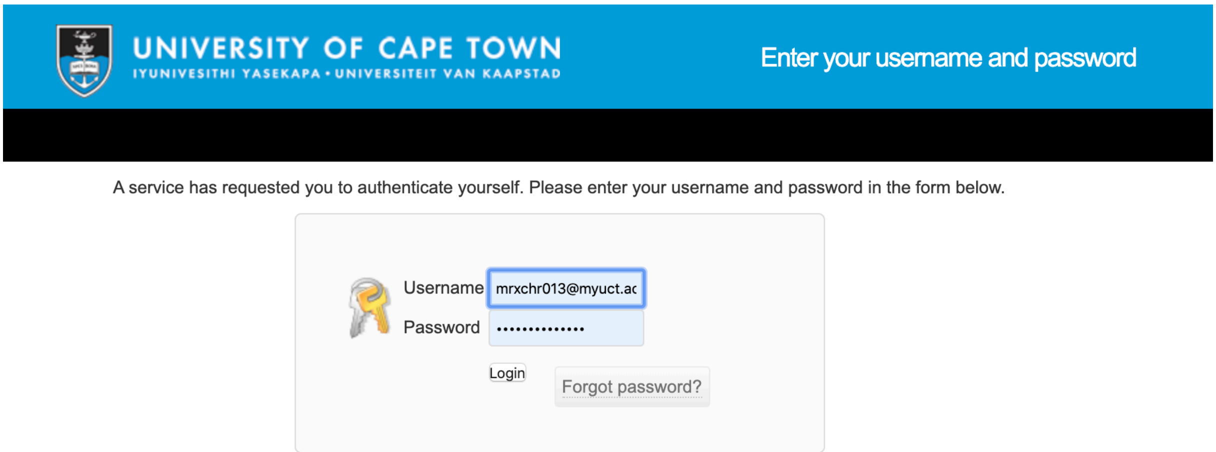


Figure I.6: After selecting the university of Cape Town the user is re-routed to a UCT login page where they can enter their normal UCT Username and password.

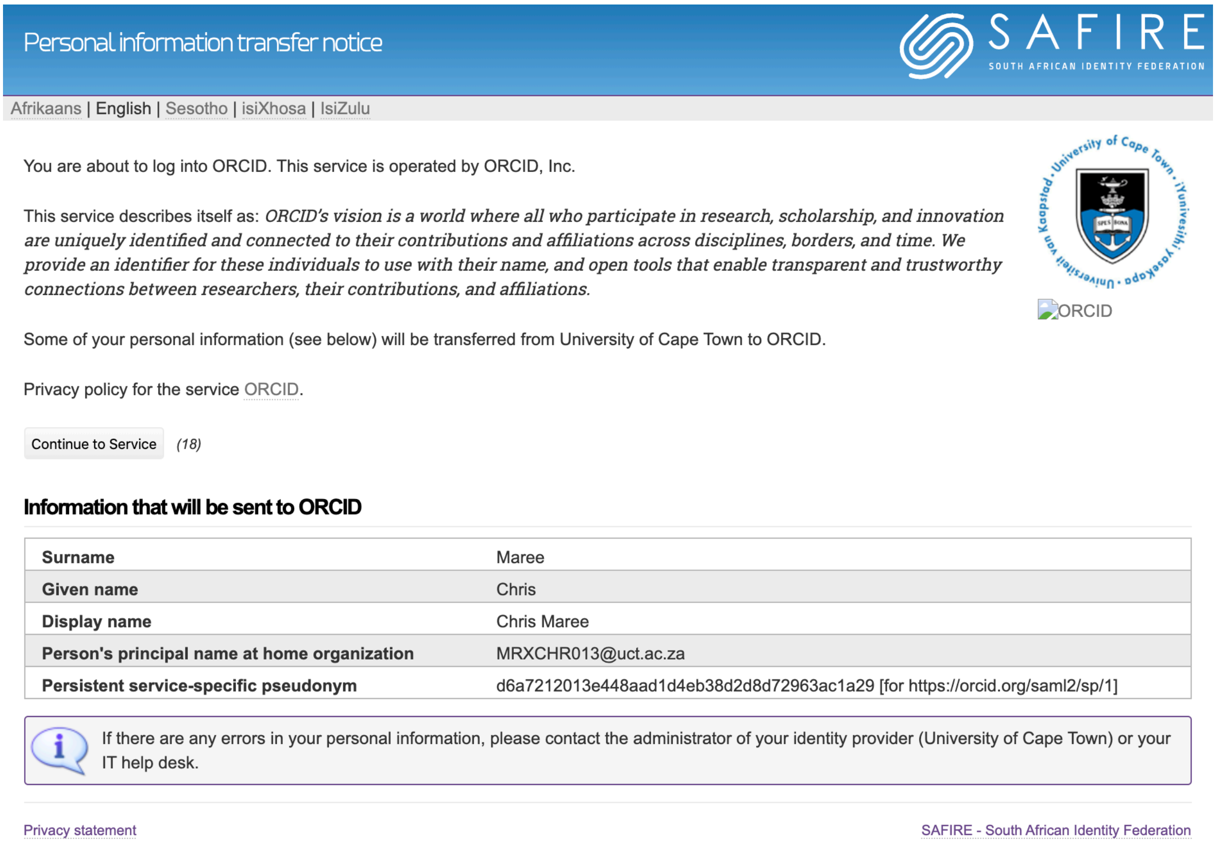


Figure I.7: As part of the UCT login the academic must grant ORCID permission to access some credentials. This service is offered by an authentication as a service company called SAFIRE who integrate directly with UCT.

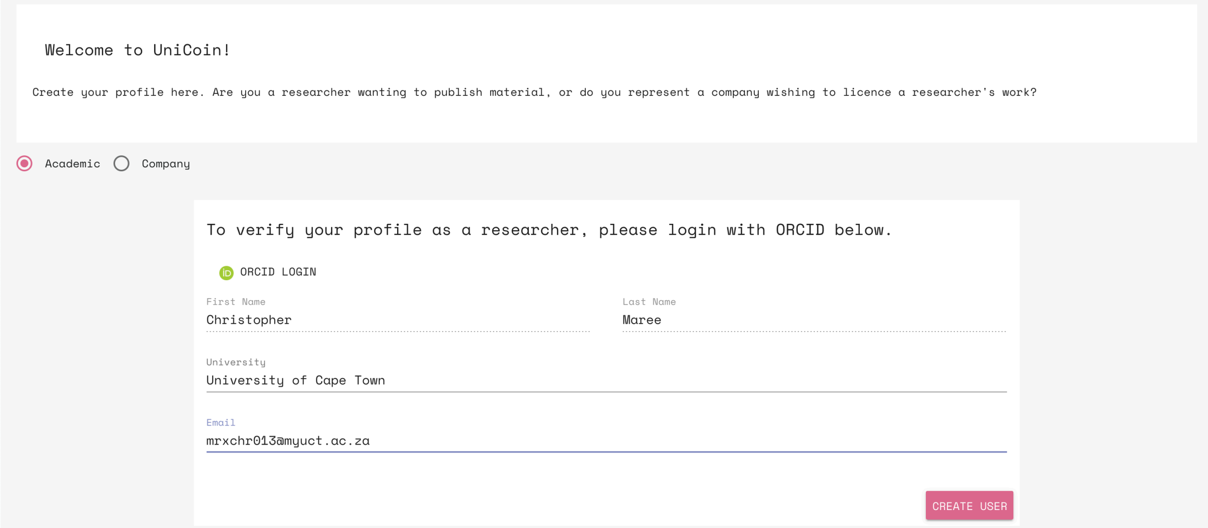


Figure I.8: After granting permission to access account details the user is re-routed back to the market. Their details are now automatically populated into the create account page.

At this point the user has created an account. This has involved publishing a JSON object onto IPFS which represents the users account profile and adding the URL for this profile to the Ethereum smart contract. An example profile URI can be found here and the Ethereum transaction used to register the user can be found here.

Publication listing involves the academic uploading their paper and additional information such as title, abstract and key words. The academic can also specify contributors to split the received revenue with. For this thesis this process is manual where the academic selects how much to go to each person. An additional function is available to upload a CSV which contains recipiants names, addresses and percentages attributed. It is at this point where the work done by Meiklejohn (2020) would fit in to automatically attribute social credit based off influence and impact of a paper.

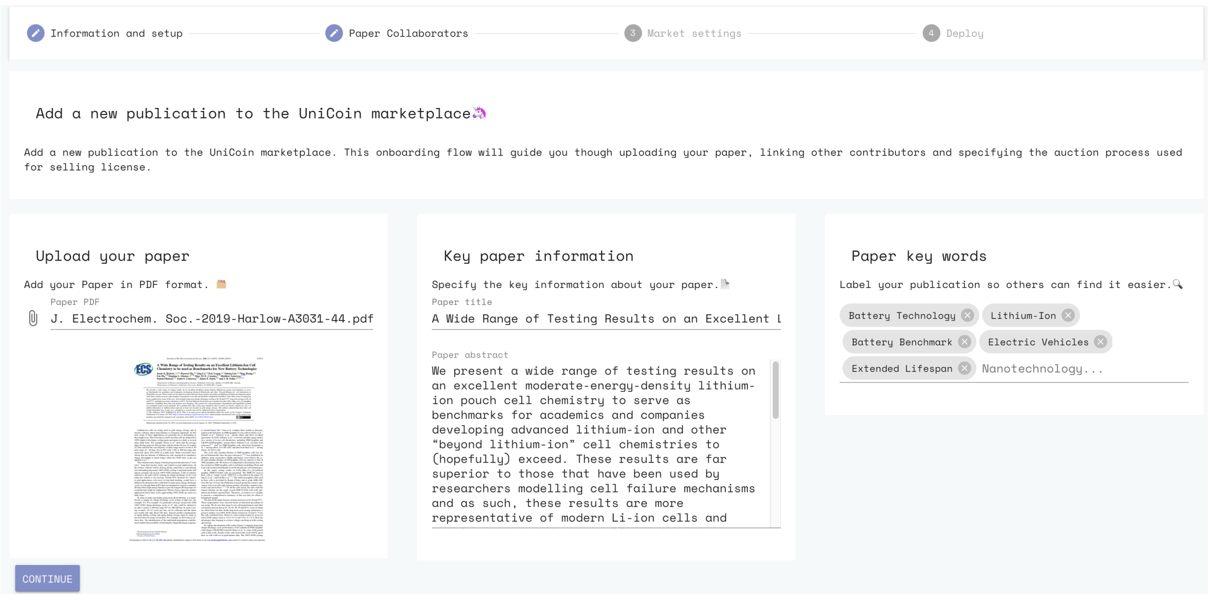


Figure I.9: Create publication step 1. The academic uploads the PDF, which is previewed, specifies the publication name, abstract and keywords.

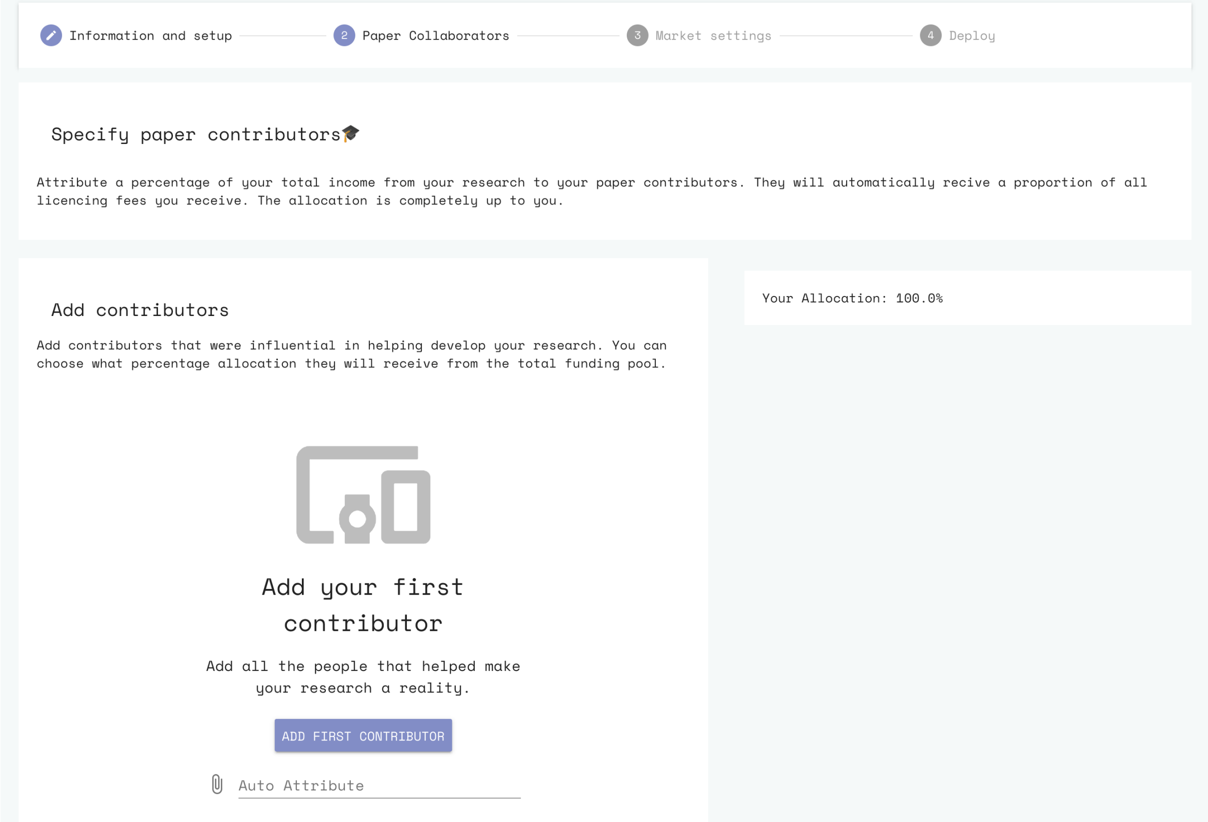


Figure I.10: Create publication step 2.a. The academic Then specifies who they want to split the revenue with. This screenshot shows the empty state if the academic has not selected anyone to share with.

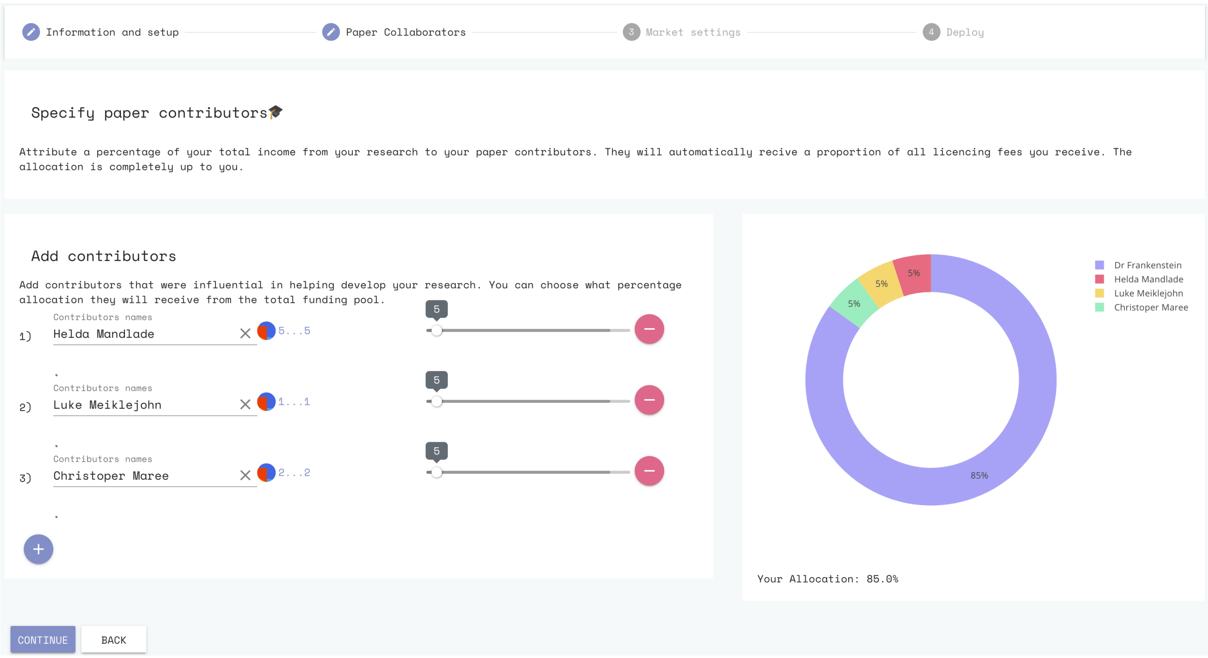


Figure I.11: Create publication step 2.b. Ideally, the academic will want to share their revenue with contributes. This figure shows 3 co-authors added to the publication. The pie chart shows who the contribution will be split with and their respective amounts.

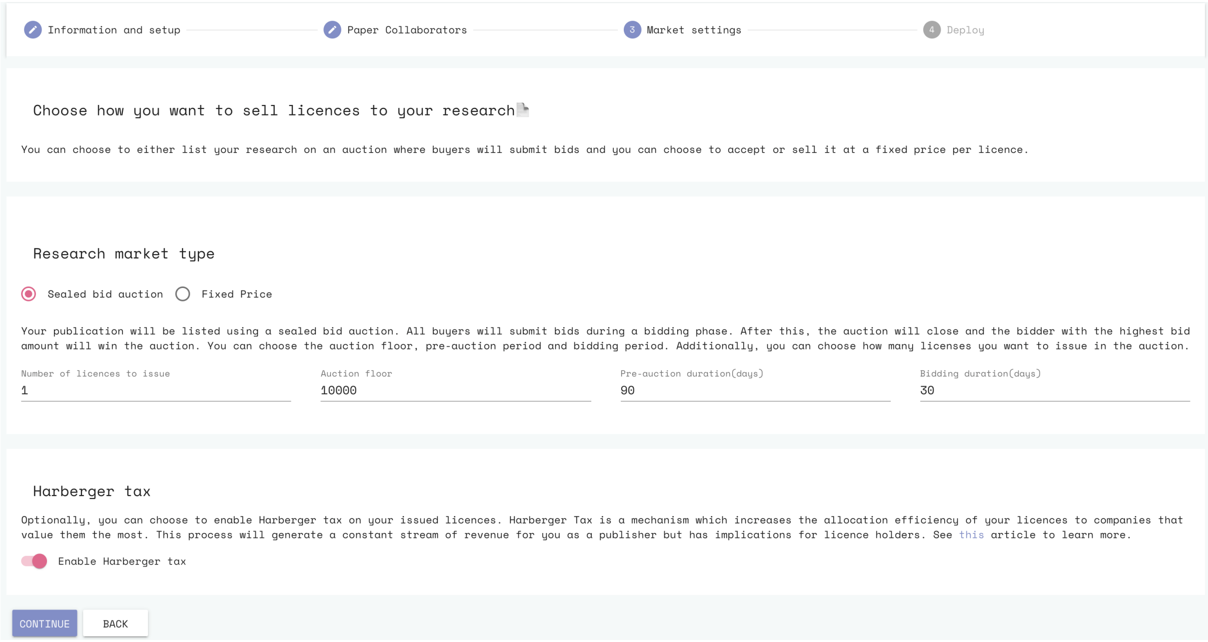


Figure I.12: Create publication step 3. The academic can choose how they want to sell their licences. This is a choice between a sealed bid auction or a fixed price. If a sealed bid auction is chosen, then they can also pick the number of licences to issue, the auction floor, pre-auction duration and bidding duration. Optional Harberger Tax is also shown.

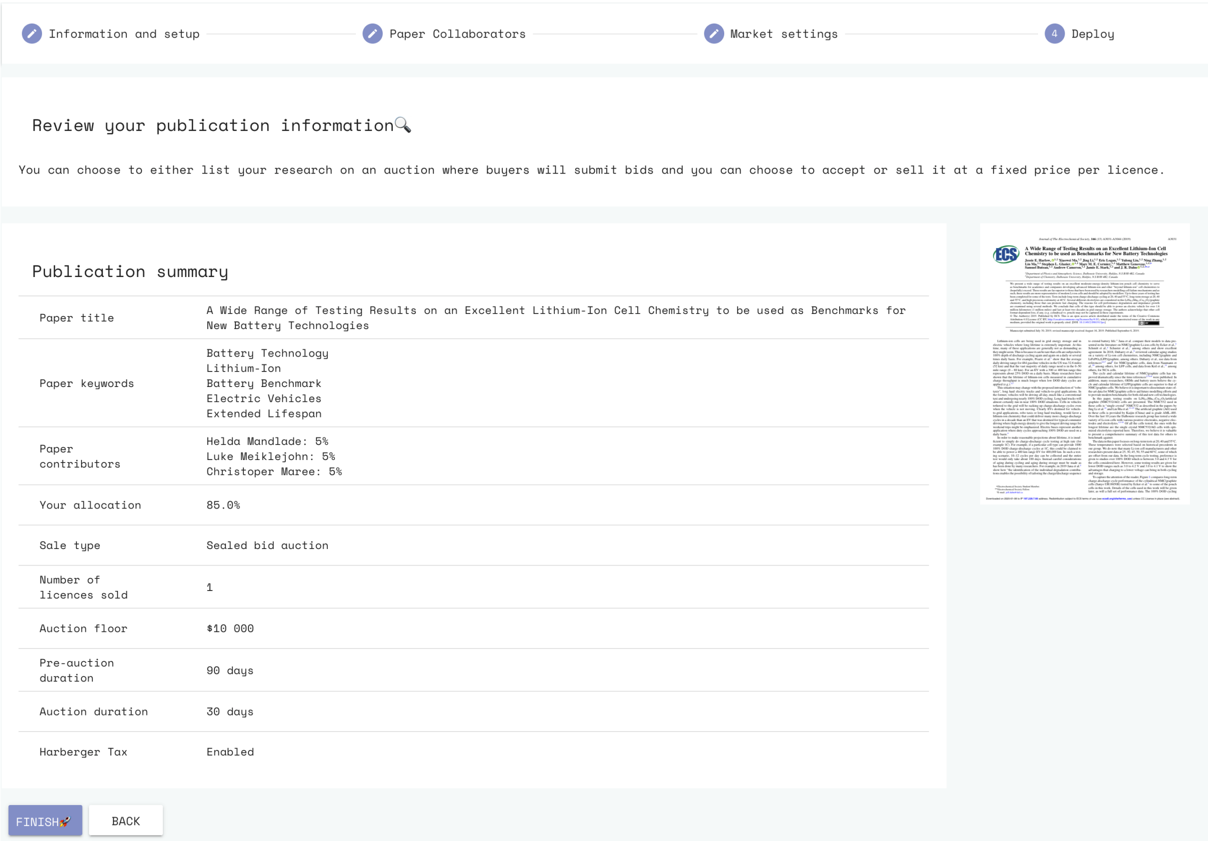


Figure I.13: Create publication step 4. Publication listing summary enables the academic to double check all provided information. Upon clicking "FINISH" their publication will be uploaded to IPFS and the sale information will be added to the blockchain.

Transaction mining process is now shown. This steps the user through the 3 diffrent stages while the transaction mines from 1) uploading files to IPFS including profile and

PDF, 2) interacting with smart contract and submitting a transaction to the blockchain and 3) waiting until the transaction mines and reporting back to the user when it has.

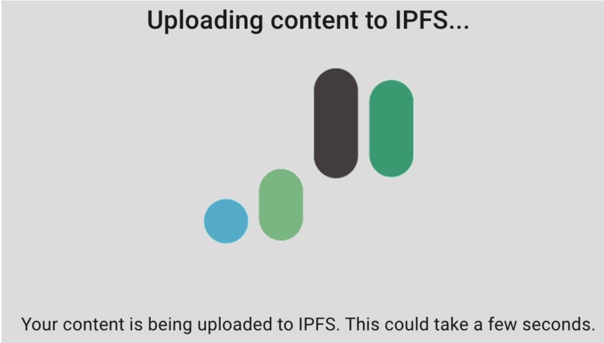


Figure I.14: Pending state while PDF and publication information upload to IPFS. This can take up to 15 seconds depending on the size of the PDF.

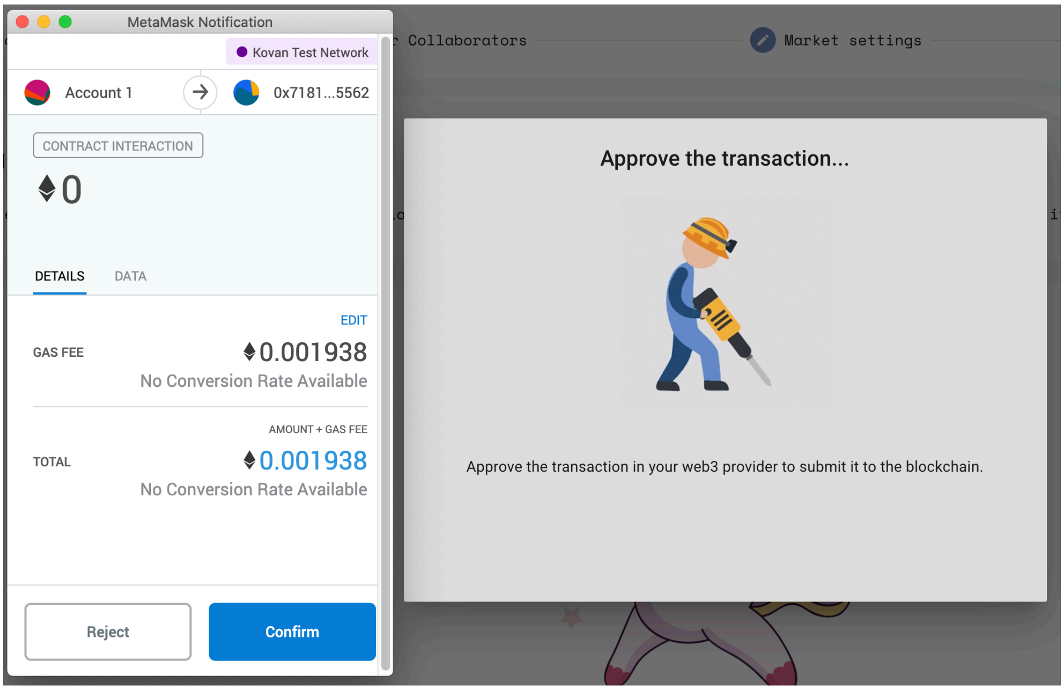


Figure I.15: Sign transaction modal with Metamask to submit a transaction to the blockchain to list in the marketplace and store provided information.

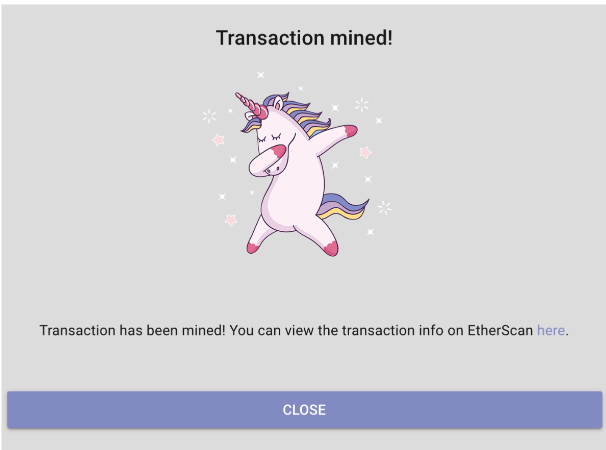
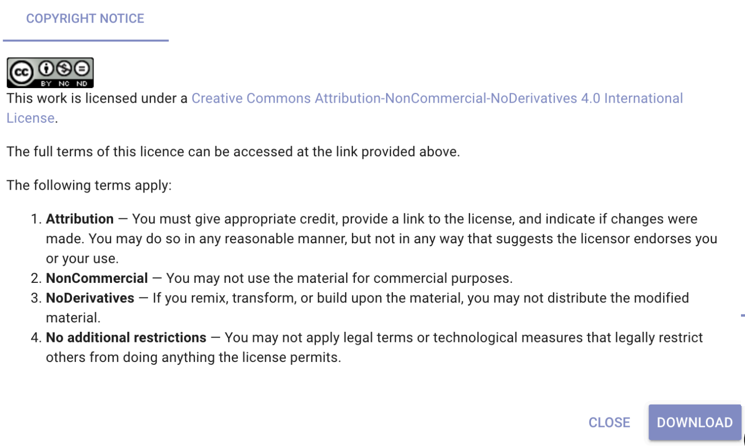


Figure I.16: Once a transaction has been processed and everything has complected the user is informed. A hyperlink to the transaction hash is included.

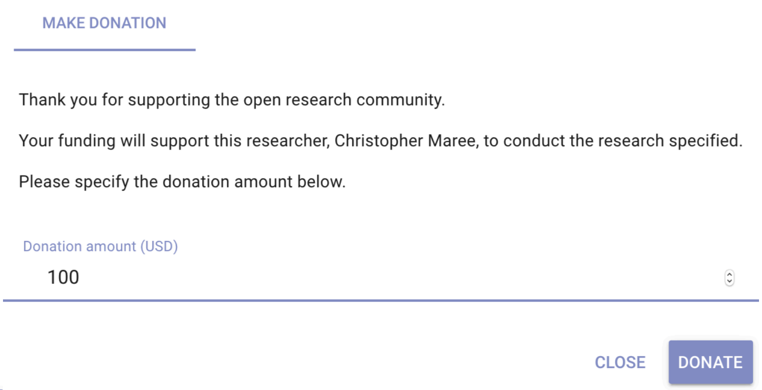
Publication marketplace lists all the publications that have been listed within the marketplace. Each publication has it's author, abstract and keywords listed along with some details around it's auction status. Based on the state of the publication auction diffrent auctions can be performed.

Figure I.17: All publications within the market are shown in order of publication. Here, a user can see the current status of the auction (currently "Pre-Auction") and the licensing terms.

A user can also download the paper for non-commercial usage, as shown in Figure I.18a or donate directly to the author of the paper, as shown in Figure I.18b.



(a) To download a paper with non-commercial intentions you must agree to the licensing terms before the PDF will be downloaded.



(b) Alternatively, you can choose to donate directly to the authors. This donation will be split based on the authors division defined when creating the listing.

Auction bidding starts once enough time has passed from the initial publication listing. During the bidding phase any user can submit a sealed bid to the auction. This process is outlined in section 5.2.1 and involves creating a bid `salt` which is submitted along with the `bidAmount`. Figure I.20 shows a publication that is ready for bidding, notated by "Auction running" and the presence of a "BID IN AUCTION" button.

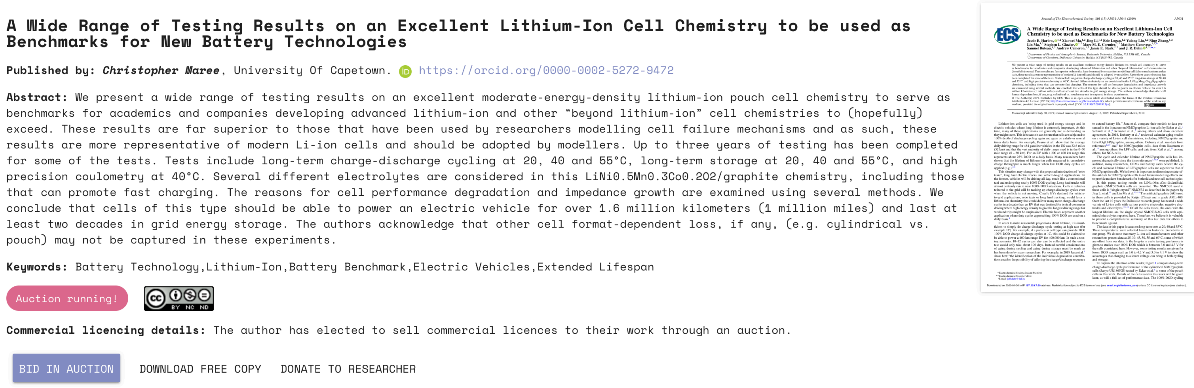


Figure I.19: Once a transaction has been processed and everything has compleated the user is informed. A hyperlink is included that will direct them to the blockchain if they wish to see the processed transaction information.

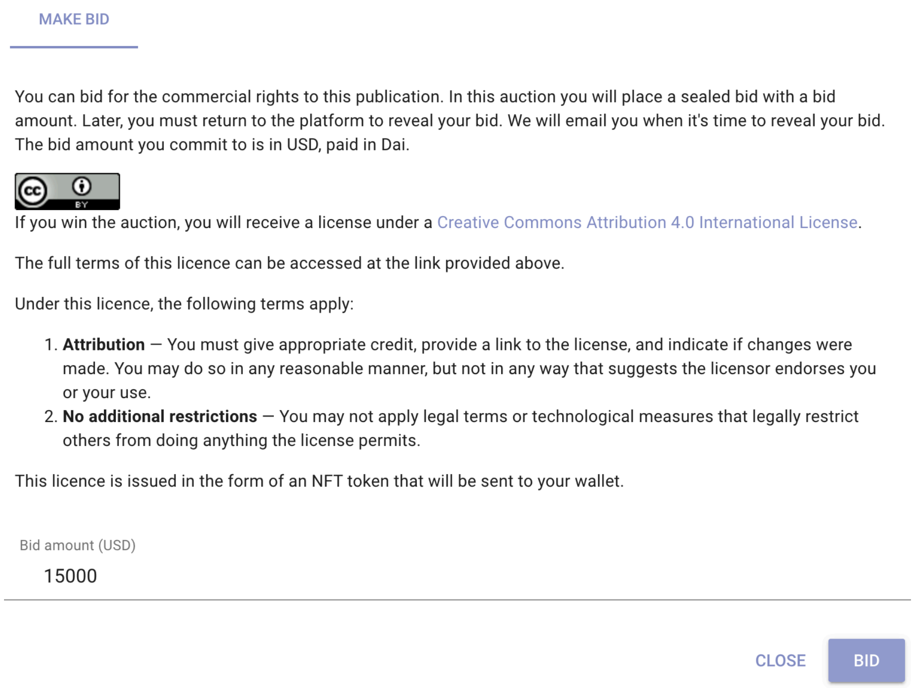


Figure I.20: A user can bid on a licence by specifying the bid amount. After clicking BID they will sign a transaction which will submit their bid commitment to the blockchain.

Auction finalization can occur by any user finding that an auction has entered the finalization phase and then calling the `finalizeAuction` function. Users are incentivized to call this function as it pays twice their spent gas back. For the initial implementation presented in this thesis this function is called manually. This is not shown here for brevity but was done using a web3 call directly from web3.js run within a shell.

The finalization process involves looping through all bids and identifying the winner. Then, payment is automatically made from the winner to all contributors and a NFT is automatically issued. An example issued NFT can be found on Kovan Etherscan here. This is in accordance with section 5.2.1.1.

Figure I.21 show the licence issued to auction winners profile. Figure I.22 shows information shown to the academic who created the publication. Here they can claim the outstanding Harberger Tax associated with their publication at any point in time. The tax paid is the amount accrued from the most recent payment to present block time.

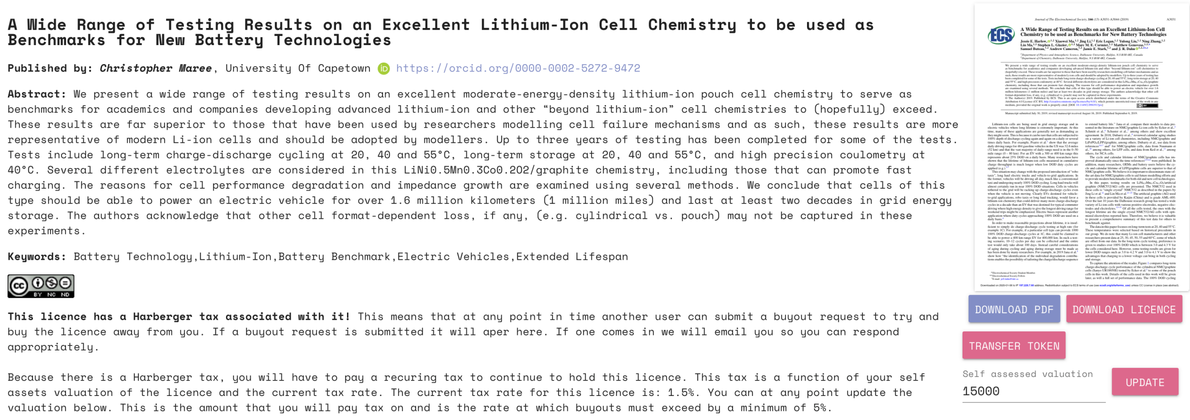


Figure I.21: Auction winners licence information associated with an issued licence. The licence owner can transfer the NFT, download the paper's PDF or their licence. Additionally, they can also update their Harberger Tax valuation.

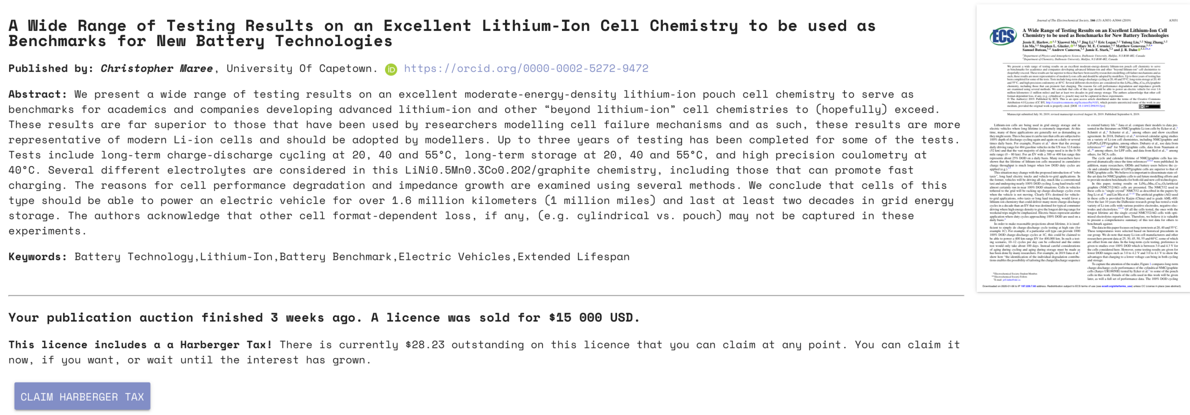


Figure I.22: Academic publishers information associated with the licence. They can view when the auction completed, how much it settled for and what the current outstanding Harberger Tax is. At any point they can choose to claim the tax. If they do, instant settlement occurs via the smart contract.

Appendix J

Areas of further research

There are a number of other interesting mechanisms that can be coupled with the sale of ideas. These mechanisms could be offered as additional pricing options available to sellers when listing their ideas. Some of these additional mechanisms are now discussed.

Bonding curve with unit licencing

Unit licensing rights (ULR) is a mechanism to sell licences to be used in exactly one produced product (Jarosz et al., 2010). These are normally sold in bundles enabling the buyer to produce a fixed number of items. For example a bundle of 1 million telephony licences could be bought by a smartphone manufacturer, granting rights to make 1 million smart phone units containing said licensed technology. This constriction enables secondary markets by design.

This licencing scheme can be combined with a bonding curve to create an automatic market maker for unit licences. Bonding curves are a kind of smart contract marketmaker that enables the continuous sale of algorithmically priced tokens. In this implementation these tokens are unit licences granting permission to use an idea in a commercial context.

A bonding curve works by issuing (minting) tokens in exchange for users bonding another token. The exchange rate is defined by a mathematical function which relates the current token price to the current circulating token supply. For example $price = supply^2$. This function is arbitrary and can be any function to relate the price of the issued token to the supply of issued token. The bonded token in this case is Dai. Upon depositing Dai a token is minted to the buyer.

Within this construction a company will deposit Dai into the smart contract which will mint a licence for the exact number of items they would want to make. The process of bonding Dai mints a publication specific token called the Uni token. Uni tokens are non-fungible between different ideas; each idea therefore has its own unique Uni token and as such its own Uni token market.

In order to "use" the licence, when constructing a product for example, they must "burn" the token into the bonding curve. This acts to destroy the token and removes it from supply. When a token is burnt the underling bonded Dai remains in the pool. This acts to increase the price of all tokens because the total supply of tokens has decreased for the same amount of bonded collateral. The burning process works by decreasing the supply by half of the tokens sent and sending the other half to the idea creator and all contributors. This acts as a constant stream of revenue for the idea creator as every time someone uses the idea, the idea creator can unbond their tokens for the Dai in the pool.

This mechanism also enables people to "stake" on a specific idea by buying tokens from the idea's curve but not burning them in manufacturing an item. As others buy tokens and burn them while making products the investors token will appreciate in value enabling them to sell at a profit later on. The investor being involved in the market is beneficial to the idea creator as they push up the price along the bonding curve which will yield higher return for every unit licence the idea creator gets from burnt tokens.

Note that an investor does not burn their tokens at any point. Rather, their act of "selling" is done by unbonding their UNI tokens to get back Dai. The unbonding process applies a sell tax which is sent directly to the idea creators. This is done to tax the investors for profiting from someone else's idea thus generating additional revenue for the idea creator. This also prevents pump and dump schemes from affecting the bonding curve. This effectively means that an investor has separate buy (bond) and sell (unbond) curves. This is due to the fee between the two. An arbitrary example of this kind of bonding curve is shown in Figure J.1.

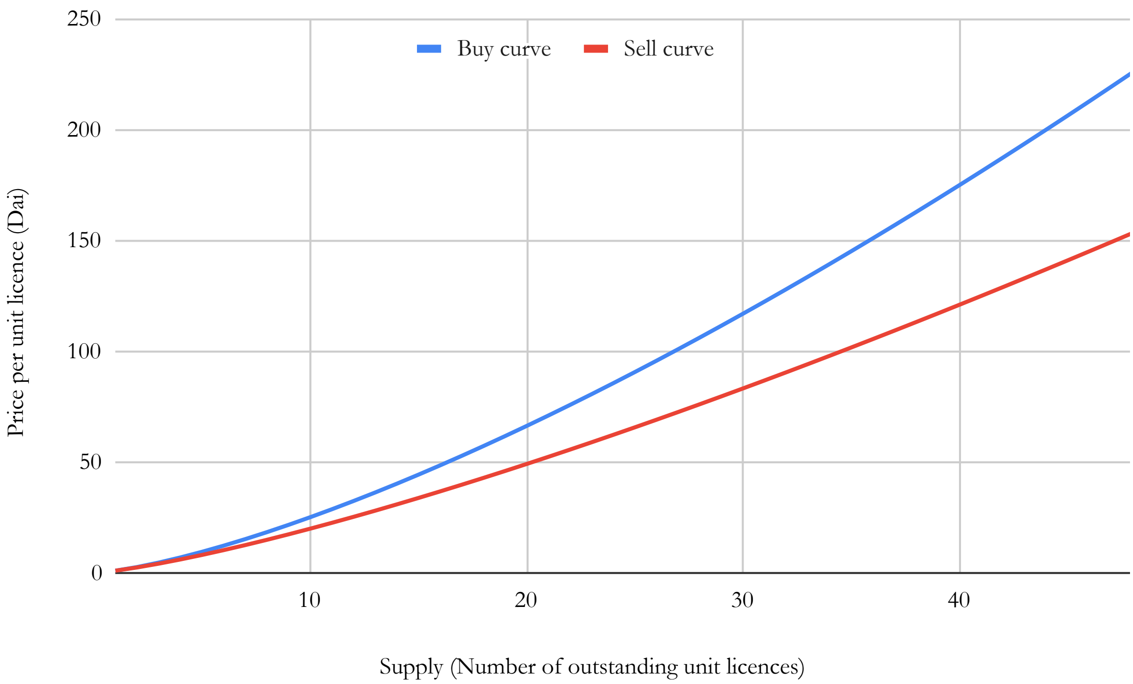


Figure J.1: Bonding curve with separate buy and sell functions

Mechanism advantages

This mechanism is useful because it creates a continuous, instantaneous market. This is advantageous over the auction and Harberger Tax model which takes longer from a buyer seeing an idea of interest, bidding, revealing and then finalizing an auction. The bonding curve mechanism on the other hand enables instant purchases for implementers.

Additionally this mechanism does not exclude any particular individual or group. This is not an artificial monopoly in the same way that an exclusive licence would be. This can be said to democratize access to commercially viable ideas. Additionally, often a company will not require a monopoly licence but rather needs to build a fixed number of products with a licence. Using a unit licence means that these companies only need to buy the exact number of licenses required for their product. This decreases licensing uncertainty and overhead cost versus acquiring a generic unlimited licence. In this way the unit licencing mechanism can be seen as a more precise licencing technique.

The ability to "stake" on an idea by buying tokens with the intention of unbonding them later at a higher price changes how investors interact with intellectual property and ideas. This new mechanism enables an investor to directly benefit from the success of an idea which effectively creates a new accessible asset class. Therefore you can "invest" into a patented idea such that you get a share of future revenue generated from the idea.

This process of staking additionally acts as a signaling mechanism to identify the most important ideas. This is a costly signaling mechanism due to the opportunity cost and market risk of staking which makes it an effective way to curate content (Mcandrew and College, 2019). Ideas that have a lot of value staked in them show that the community of investors believe that it will be profitable into the future. This can be used to build a token curated registry. At the same time, investors benefit inventors as their holding tokens drives up the price of new minted ULRs which generates more revenue.

Mechanism disadvantages

There could be difficulty in corroborating the number of products produced with the number of licenses purchased. For example, if the smartphone manufacturer buys 1 million licences but in fact creates 1.5 million devices it is hard to validate this without trusting the phone manufacturer. If there is uncertainty here this will enable licence buyers to under pay for their licences. Ultimately additional modeling and idea validation is required before adding this mechanism to any real world system.

IP index funds through the tokenization of ideas

IP index funds have been around for over a decade with early implementations being launched by Ocean Tomo in 2006 (Malackowski et al., 2007). The fund consisted of a portfolio of 300 companies that own the most valuable patents as assessed by Ocean Tomo, relative to the companies' book value. Companies included in the fund were chosen based off key metrics Ocean Tomo defined.

The tokenization of ideas presented in this thesis lends itself to a potential next evolution of IP based index funds. A fund could simply own a portfolio of licences generated from this platform and then sell a derivative token which represents fractional ownership in the portfolio. The kind of licenses that the fund owns can be governed by a number of different rules, including but not limited to a Decentralized Autonomous Organization (DAO) which would choose what kind of tokens to include.

Alternatively, an index fund of different idea domains could be created. For example patents relating to telecommunication could be owned by one specific fund and patents relating to healthcare to another. This model interacts well with the bonding curve as the fund owns bonding curve tokens to represent fractional investment into the ULR market.