GreenBlocks: A New Green Building Rating System Created on the Blockchain

GreenBlocks: A New Green Building Rating System Created on the Blockchain

Kristen Bacorn

A Thesis in the Field of Environmental Management

for the Degree of Master of Liberal Arts

Harvard University

May, 2020

Frontispiece, Subject Building Site in Las Vegas

Abstract

This thesis explored the possibility of developing a new alternative to conventional, centralized green building rating systems like LEED, Energy Star and Green Globes. It further explored the possibility of originating and maintaining the rating system's data on the blockchain via Ethereum.

Although real estate has been sold or rented with cryptocurrency, the underlying technology has not yet been applied to evaluating and recording green building construction attributes. This thesis tested the extent to which a new rating system can offer advantages over conventional green building ratings systems. Assessments also gauged the extent to which the Ethereum blockchain can successfully provide operational advantages such as peer verification, decentralization, transparency, and decreased costs.

The hypothesis was: ***It is feasible to create an effective green building rating system with a new and different approach, and that system can be successfully processed, maintained and delivered on the Ethereum platform.***

The research method used was to design a pilot green building rating system (GreenBlocks), and then beta-test and adapt it for implementation on the Ethereum platform, using a large commercial project as a test model. The site used for this live research is 38.5-acres in Las Vegas prepared for a mixed-use, new green development to be constructed. Criteria for the rating system, along with documented facts about the project, were submitted to Ethereum experts to test the technological feasibility of

loading and maintaining a large amount of specialized detail using the blockchain and smart contracts.

The goal of this thesis project was twofold: to develop an improved and highly granular method for evaluating green buildings, and to maintain the application system on the Ethereum platform, using blockchain technology. Success in loading and maintaining data for the test building on the Ethereum platform may permit exploration of the comparative merits of the blockchain approach, and the extent to which this approach may apply to other projects in future.

Results of the investigation were that the GreenBlocks system was technologically feasible. However, loading and maintaining the data on a peer network was very labor-intensive. Unlike traditional systems like LEED, where building owners assume the labor costs of uploading data to the central online application, in GreenBlocks, data is submitted by building owners (and their team) to blockchain experts. This raised practical, non-technological questions about how labor costs are to be handled in a business sense.

This thesis tested whether it was possible to create a completely new green building rating system that does not borrow from existing systems like BREEAM, LEED, Well, Green Globes, et al. Further, it tested whether the system could be created as a blockchain app and if live data from an actual building project could be successfully stored on the blockchain peer to peer network.

With a positive outcome to the hypothesis, the next step is to determine what next steps might be taken in future research. The logical next step, in future projects, is to determine if the program can be made financially feasible and if the system can find any significant market acceptance. Needless to say, the system also requires refinement both in design and technology, to such an extent that if such a project were to be created for release to the public, it might bear little or no resemblance to the GreenBlocks pilot. In addition, the intensive skilled labor necessary to create a product beyond the GreenBlocks pilot may prove to be financially unsustainable.

It will be interesting to see if GreenBlocks has any kind of viable future as a construction industry product.

Author's Biographical Sketch

This student researcher holds degrees in Literature and Architecture as well as a New York State Real Estate Broker license and a Series 7, General Securities Representative Qualification. She established her own business, Green Estate Consulting, which provides expert environmental services to clients such as developers, property managers, businesses and local government. She has worked on environmental building projects at the Solaire, Empire State Building, Sony Tower, Sikorsky Aircraft, the Plaza Hotel, JFK Airport and other notable structures. This thesis student also teaches environmental science, LEED accreditation and real estate licensing at State University of New York and private institutes. As a public speaker, she has addressed the United Nations Commission on Sustainable Development, American Institute of Architects, New York State Society of Certified Public Accountants, American Council of Engineering Companies, National Public Radio (NPR) and National Talk Radio. Her work has appeared in *Newsweek*, *The Wall Street Journal*, *The New York Times*, and other major publications. She also is author of a top selling book on real estate and a textbook on LEED. This student is originally from Syracuse, New York, known for its progressive environmentalism,[1] and now resides in New York City.

Dedication

This work is dedicated to a wonderful man - my father, Dr. Robert Bacorn. As New York State Regional Health Director, he was a national environmental leader, in charge of air and water quality for millions of families and businesses. Throughout his life, Dr. Bacorn embraced free thought, innovation and technology; he would undoubtedly advocate a system which democratized distribution of environmental data.

Figure 1 Photo of Dr. Robert W. Bacorn

Acknowledgements

Ethereum and the blockchain are extremely complex in terms of cyber engineering and economics. They also bear implications in law, securities, real estate and the built environment. This thesis was made possible by the visionaries who conceived of this new technology during the last decade. Only with their collective effort of refining new cyber-technologies could the new decentralized, transparent green building rating system be created. This thesis depends on the accomplishments of a new generation of brilliant, dedicated theorists.

Table of Contents

Table of Figures

Definitions

Bitcoin	The first and most popular cryptocurrency
Bitproperty	A digital REIT
Blockchain	An open computer ledger recording currency transactions; data is stored in blocks connected to each other in a chain
BREEAM	(Building Research Establishment Environmental Assessment Method), established in 1990 by British Building Research Establishment (BRE); world's longest established method of assessing, rating, and certifying the sustainability of buildings
Cryptocurrency	Digital money instead of paper money. Defined in coins rather than dollars, but can be purchased for dollars. Instead of being managed by a central bank, it is managed by a network of peer computers who record an agreed transaction ledger (blockchain).
Distributed application	Software that runs on multiple computers within a network at the same time and can be stored on servers or with cloud computing.
Energy Star	A numeric system for rating energy and water efficiency of buildings. Developed jointly by Environmental Protection Agency and Dept. of Energy
Enterprise Green	A popular green building rating system similar to LEED
Ethereum	The second most popular cryptocurrency which offers data processing opportunities beyond simple currency exchanges
Green Globes	A popular green building rating system recognized by the U.S. General Services Administration (world's largest real estate manager)
ICO	Initial Coin Offering: a way for a new cryptocurrency venture to raise funds. A percentage of the cryptocurrency is sold to early backers of the project in exchange for traditional money or other cryptocurrencies.
LEED	Leadership in Energy & Environmental Design. A green building rating system developed by U.S. Green Building Council, a private non-profit group. LEED differs from Energy Star as it goes beyond energy conservation to address issues like biodiversity, human health, air quality, etc.
Living Building Challenge (LBC)	A popular green building rating system based on LEED
Miners	Computer specialists who maintain the blockchain in return for compensation in the form of crypto coins.
Pangea	A blockchain platform for shared property ownership, by ConsenSys Real Estate; a particular private company
REIDAO	A public trust company for buying and selling real estate shares. It is the name of a particular company, even though it sounds like a type of business formation.

REIT	Real Estate Investment Trust. A type of business formation permitted by law; a publicly traded company that invests in real estate. People can buy shares in the REIT and thus become an indirect owner of real estate and receive dividends on the property's income.
REOC	Real Estate Operating Company. A publicly traded real estate investment company in which shareholders profit from property's appreciation rather than short term income.
REX	A crypto real estate listing system
Small cap stocks	Stocks in companies with share value of $300 million to $2 billion. Value is calculated by market value of shares times number of shares.
Smart contract	A computer routine that automatically initiates performance of certain standards, protocols or contracts
Well	A popular green rating system based on LEED

Chapter I. Need for Research on New Green Rating System

This thesis involves the creation of a new green building rating system. Before creating anything, the necessity for that thing should be established. Accordingly, this Chapter explains the reasoning behind the research goals, questions and hypothesis established in this thesis. Chapter I also includes background on green building rating systems and blockchain technology – the two subjects being united in this thesis.

Introduction

A new technology – the blockchain – is revolutionizing the way currency is defined and used worldwide. In this thesis project, blockchain technology was applied in a different sector: green building rating.

Although green building rating/certification systems have been popular and successful for decades, improvements can be considered, especially with the advent of new technologies like the blockchain. Inevitably, limitations and drawbacks typically appear in systems over time, and new technologies develop which may allow even the best systems to be enhanced or replaced.

Blockchain technology and the proof of work concept that validate smart contracts in a leading cryptocurrency, Ethereum, could conceivably be used for green building rating. This thesis examined: a) the extent to which a new green building rating system can be a step forward from traditional, centralized systems like LEED (USGBC, 2013), Energy Star, Green Globes (GBI, 2016), et al., and b) the extent to which this system can be successfully supported by decentralized blockchain technology. This examination was accomplished by designing a pilot green rating system on the Ethereum platform, using a large commercial building as a test subject.

Objectivity and transparency in data handling are probably the most critical differences between traditional building rating systems and the pilot rating system in this thesis, GreenBlocks. In traditional systems, buildings are awarded points and a certification, such as Silver or Gold, with the reason points are awarded remaining confidential. In GreenBlocks, buildings are not judged or awarded points. Instead, large amounts of a building's factual data related to environment are stored publicly on the blockchain, allowing consumers and expert reviewers to draw their own conclusions about the building's green characteristics.

This new concept in green building rating was developed by the student author of this thesis, in cooperation with the owners/developers of a live commercial real estate project and an expert Ethereum consultant. The new rating system, GreenBlocks, has been developed and tested using live subject data. The objective was to test the hypothesis: ***It is feasible to create an effective green building rating system with a new and different approach, and that system can be successfully processed, maintained and delivered on the Ethereum platform.***

Research Significance and Goals

During the last decade, a new system of managing currency transactions has emerged. Cryptocurrencies Bitcoin and Ethereum have become household names, and have been applied to real estate sales and rentals.[2] Using blockchain technology for real estate points to a next step: applying blockchain technology to environmental buildings – the topic of this thesis.

Ethereum was selected as platform for this thesis on green buildings because Ethereum is not limited to currency, but can be applied to a wide range of sectors, from insurance, to car sales, to online dating (Decentralizing Everything with Ethereum's Vitalik Buterin, 2017). In fact, the original developers of Ethereum are leading the industry in applying the blockchain to tokenizing real estate assets, not just using an alternative currency to buy or rent (ConsenSys Media, 2018). Other advantages of Ethereum over other blockchain systems are that Ethereum incorporates Smart Contracts – computer routines that automatically initiate performance of certain standards, protocols or contracts. Accordingly, Ethereum uniquely lends itself to creating a class of rated green real estate assets based on quantitative records. Details on both the blockchain and Ethereum are offered in Appendices I and II, respectively.

When green building rating systems were introduced 30 years ago, they presented an unprecedented way of looking at structures. For centuries, builders did not really consider issues like water waste, daylighting or materials' provenance. Undoubtedly, green rating systems like BREEAM, LEED, Green Globes, Living Building Challenge and many others raised public consciousness about building in harmony with the environment. Most of these systems are structured as award incentive programs based on points for certain features determined by a certification body. The advent of blockchain technology offers the opportunity to completely restructure building ratings, making them decentralized, non-award based, and objective. The ideal result of such restructuring would be a building rating system with improved transparency and verifiability of data.

Would developers be incentivized to build green without a reward system, such as giving Silver or Gold plaques to compliant buildings? Strong evidence suggests yes. In 2016, respected research firm Dodge Data & Analytics did a survey of World Green Building Trends (Dodge, 2016). They asked 1000 building owners in 69 countries what were the biggest benefits of green construction. By far, the clear majority worldwide – 69% - agreed one thing was the most important benefit of a green building: lower costs. The percent of respondents citing lower costs (for utilities, energy, maintenance, etc.) as the biggest benefit was even higher in America, at 81%. The benefit of getting green certification was not on the list. Here are the other benefits, with the percent of green building owners who named each benefit.

Important Benefits of Green Building	World Average
Lower Operating Costs	69%
Quality Assurance	31%
Sustainability Education for Occupants	31%
Higher Value at Point of Sale	30%
Future Proofing Assets	26%
Design Flexibility	25%
Increased Productivity for Tenants	22%
Higher Rental Rates	17%
Higher Occupancy	16%

The method of exploration in this thesis project was to design a pilot green building rating system, verifiable on the blockchain, for recording numerically the environmental attributes of a specific large commercial building and its site. The objective was to develop a green rating system that would present an alternative to traditional building rating systems. The pilot rating system was created and tested on the Ethereum platform.

The pilot rating system created for this thesis is named GreenBlocks. The name is intended to evoke imagery of building blocks while referencing the blockchain, which is the system's distinguishing technological feature. (A name search was performed to avoid accidental copyright or trademark infringement.[3])

Before creating anything, the necessity for that thing should be established. A reasonable question would certainly be, "With multiple popular green rating systems in use for decades, why is a new green rating system necessary?" The answer, which is an underlying premise for this thesis, is that existing green rating systems have undeniably fine attributes which have provided a service to the construction industry and the public, but limitations and drawbacks typically appear in systems over time, and new technologies develop which may allow systems to be enhanced or replaced. The most popular green rating systems, such as LEED, Green Globes, Energy Star, et al., were created decades ago, and it may be time for more modern solutions that build on the successes of these systems. The advent of blockchain technology may offer an opportunity for growth and enhancement of green building ratings.

Many environmental experts have noted limitations and drawbacks of existing rating systems, some of which are listed in the following subsections. Also listed are possible solutions which GreenBlocks – the pilot system in this thesis – is intended to provide. In the following subsections, the contrast between existing systems and GreenBlocks is intended to show the significance of this thesis research.

Decentralization

The essential attribute of the blockchain is decentralization, with transactions verified by a peer network instead of a central authority.[4] Traditional rating systems are centralized. In LEED, Green Globes, or Energy Star, for example, data is stored privately by a central authority, and points are awarded by that central authority. Decentralized green rating ideally would allow the rating process to be more objective (based on computerized metrics) rather than subjective (determined by interpretations of a central review team). With decentralization, data is processed, maintained and distributed by a large, democratic group, rather than a tiny, elite, anonymous group.

Objectivity

Traditional ratings are often demonstrably subjective, with points awarded through decisions of a small, unknown rating team. For example, one company was awarded LEED points for having educational signs and a dedicated bicycle room, even though the room was never used (nor could be used) for bicycles. In GreenBlocks, features like signs or employee amenities are considered too subjective to have any value in quantitative measurement.[5]

During renovation and certification of Empire State Building (ESB), owner Anthony Malkin was an outspoken critic of the arbitrariness of rating systems such as LEED. Mr. Malkin was quoted in a panel discussion as calling it "outrageous" and "a crime" (along with some vulgar language) that sustainability programs don’t do enough to push landlords to improve energy efficiency (Troianovski, 2010). In a speech at Urban Land Institute, he joked that even though his 2.7 million square foot building had cut energy use by some 40%, he would have had more LEED points if he put flower pots on the ESB roof (Troianovski, 2010).[6] Mr. Malkin contended that building systems should be more like cars, where landlords are required to disclose numeric energy efficiency, the same way car manufacturers disclose gas efficiency. In other words, they would disclose actual efficiency numbers instead of simply showing a plaque from an arbitrary authority.

In GreenBlocks, data such as energy use intensity and air particulate matter concentrations are recorded, as opposed to subjective info like views, education, and bicycle showers. During GreenBlocks' development, every attempt was made to minimize subjectivity in the type of data reported, and to use measurement systems standardized by local laws and best engineering practices. Quantitative data are not interpreted, judged or rewarded, but merely recorded on the blockchain. Numeric data in the blockchain are managed by a large peer group, based on consensus only about verification of the numbers. Building consumers may make their own determinations about the relative importance of things like plants, educational signs or energy use because the information is decentralized, freely available and not weighted per "green building element category".

Objectivity and transparency in data handling are probably the most critical differences between traditional building rating systems and GreenBlocks. In traditional systems, buildings are awarded points and a certification, such as Silver or Gold, with the reason points are awarded remaining confidential. The only information consumers have available to decide whether to buy or rent a given property is the single fact of its certification level. In GreenBlocks, buildings are not judged or awarded points. Instead, large amounts of a building's factual data related to environment are stored on the blockchain and made public, allowing consumers to draw their own conclusions about the building's green characteristics. If consumers, defined and the building industry or the public, lack expertise to draw such conclusions, they can refer to expert reviews. This is how most products in America, from movies to cars to restaurants, are rated. That is, by providing factual specifications open to reviews by experts or the public.

Adaptability

Most rating systems group all office buildings in the same category, all hotels in the same category, and so forth. One drawback is that the way points are structured, a large, energy-efficient urban office would receive more LEED points than a small suburban office that saves less energy but protects wildlife and trees (which provide carbon sequestration). Traditional rating systems have been criticized for being too rigid, and for arbitrarily favoring certain environmental benefits over others. GreenBlocks is designed to offer consumers open information about the exact attributes of each particular building, which are obscured by ratings like "Silver" and "Gold." A rating of Silver or Gold tells the consumer only one fact: that an authority conferred upon the building a certain number of points, for reasons kept confidential.

Clarity

The objective of green rating systems is to provide a framework for buildings to exceed performance required by construction code. But the way performance is measured is seldom simple or transparent. Many traditional rating systems are so complicated that a specialized, accredited professional must be hired to explain compliance to project owners. Qualifying to become such a professional is a difficult challenge met by few in the construction industry. The pass rate on the LEED Accredited Professional exam is reported to be only 30% (Ward, 2014) and the *LEED Reference*

Guide is 814 pages. Green Globes, Well, National Green Building Standard, et al. are similar.

Here is an example of the esoteric and complex rules used in many traditional rating systems. One LEED system requires owners to give their materials suppliers a "supply chain survey" with questions about conservation, social equity, health, etc. When this thesis author contacted U.S. Green Building Council for survey details, they said, repeatedly, that they did not know their own requirements, and users should "find someone" outside USGBC who might know. It took five months to finally get a USGBC rep to describe the survey requirement on the phone, (as it is not in any written record).[7] If USGBC has no standard for such a survey, how can anyone know when a project has met the requirement? In addition, conservation, social equity and health cannot be objectively measured.

GreenBlocks was designed to minimize such ambiguity by recording specific factual data. Instead of conservation, social equity and health, concrete info could be recorded, such as recycled content and miles shipped, along with published Environmental Product Declarations (EPDs).

Data such as energy use or air quality can be presented relative to local code. For example. "this building uses 20% less energy than required by local building code." This is how buildings were characterized before systems like BREEAM or LEED were developed, and is an accurate and descriptive way of conveying full information to the public. Comparing a building's performance to code is a time-honored technique which was used long before rating systems were invented. GreenBlocks adheres to local codes and standards whenever possible.

The intended outcome of using GreenBlocks is that instead of concentrating on understanding LEED compliance requirements to get points, the project team (architects, engineers, etc.) could concentrate on exceeding building code with measurable metrics (see Appendix I for list of measured green criteria.) This could simplify matters by requiring project teams to understand only one system – the local laws and how to surpass them – rather than a second system, such as Green Globes or Well.

Continuum

Most traditional green rating systems (like LEED, for example) let buildings earn/not earn 100 credits in a binary system which does not allow for degrees of achievement along a continuum. For example, if 50% of a building's power is green, the building gets a point, but if 49% is green, the building gets no points. This could actually act as a disincentive to saving resources, because designers might seek points rather than resource conservation. If a building can't achieve 50% green power and get a point, then designers may choose to have no green power at all, since they are not getting a point anyway.

GreenBlocks is designed to allow real estate consumers (builders and the public) to consider nuances of a property's actual environmental characteristics, such as 49% versus 50% green power supply.

Wide Range of Data

In GreenBlocks, green attributes are not filtered according to the concentration of any particular organization, such as human health for Well (IWBI, 2017), energy use for Energy Star or climate change for LEED. The blockchain records a wide range of characteristics, like a product specifications sheet. Hundreds - or thousands - of a building's attributes are considered, unlike traditional third-party ratings which consider very few attributes of a building (only one or two in Energy Star and only a few dozen in Green Globes, Well, LEED, etc.). This is due to the structural fabric and design of these rating systems.

An analogy might be a car. If a consumer is trying to decide which car to buy, is it helpful to know if the car gets an overall score of "Gold" or "75" from some organization? Or is better to know the car's miles per gallon, time 0-60, weight, cabin volume, tire radius, etc.? Another example is MSDS (materials safety data sheets) for building products, which inform the construction team about all the salient, quantified attributes of a product, rather than giving the products "gold stars," as many building rating systems literally do.

In fact, this thesis author is unable to think of any other product that consumers buy based only on one single piece of information – a Gold or Silver plaque – with zero details as to why the product received that rating. Foods list ingredients and nutrition factors (called the "cereal box"), movies are subject to reviews and audience votes, and appliances are tested by many third party organizations who disclose their rating criteria (which LEED does not). Even hotels, which may use the Michelin star system, are supplemented by explicit consumer ratings on TripAdvisor, Expedia, Yelp and other public websites. Organizations that rate products are household names, such as UL (Underwriters' Labs), Better Homes & Gardens, Consumer Reports, Environmental Working Group, et al.

Instead of an outside authority choosing 100 arbitrary features and judging in a binary manner whether a building meets/does not meet an arbitrary performance threshold, GreenBlocks allows reporting of virtually unlimited characteristics. The building consumer can decide which features are important to them, just as a car shopper does. Real estate buyers who are not building science experts can read reviews, just as car buyers and other consumers do.

Therefore, building characteristics were not ranked, weighted or calibrated, but merely listed in a string. The objective was to create an environmental profile for a building – or DNA strand, so to speak – for each building, to be recorded with consensus on the validity of numbers maintained by the blockchain computerized peer network.

Building science is complicated. Architects, engineers and other construction team professionals probably understand a concept like PM 2.5 measurement (for air quality), whereas a potential tenant does not. So the question arises: without a gold plaque, how can laymen interpret the many specs to know if a building is green? The first answer is that consumers use experts in every realm, to analyze data and help them choose everything from cars to clothes to movies. LEED data is kept confidential, so the only thing consumers – and outside experts – can know is what color plaque the building has. A second answer is that a building's data can be categorized, so consumers concerned with air quality, or those concerned with biodiversity, can find specific data quickly. Finally, an executive summary of a building's characteristics can be provided so consumers do not have to sift through voluminous technical data.

Although building characteristics were not ranked, weighted, measured or calibrated in GreenBlocks, it is likely that certain consumers (builders and the public) might prefer weightings or actual measurements. For example, some consumers care about daylighting while others care about clean water – just as some car consumers want high performance and others want style. Executive summaries could provide detail on the areas of environmentalism of particular concern.

Numeric System

GreenBlocks is an almost completely numeric system. It was designed to report numeric performance data such as watts per square foot, or gallons per flush. Unlike most rating systems, it was not a scoring system with points awarded for non-numeric achievements like coming up with an innovation or having a sky view.[8]

Skeptics might argue that the environmental integrity of a building cannot be quantified in simple numbers, a position which may have some merit. However, existing rating systems are numeric to a significant degree. In LEED, for example, project teams can only earn credit points in categories like air quality and habitat by meeting certain metrics in air component testing and square footage of undisturbed open space. Energy Star is even more straightforward, evaluating buildings on Energy Use Intensity (EUI) reflected in energy bills. LEED is perceived by some as less accurate than Energy Star because it is largely based on *predicted* performance numbers (usually from computer modeling) rather than *actual* performance numbers (from utility bills or meters).

Large amounts of building data, such as water quality, type of refrigerants, and maximum occupants per square foot – to use diverse examples – are determined by local government authorities, and can simply be uploaded to the blockchain in GreenBlocks.

This data would apply to all buildings covered by that building code. For example, New York City has banned the use of number 6 heating oil. Therefore, it would be unnecessary for a project team to show evidence that number 6 oil is not used. Building inspectors in Manhattan usually do not test for radon, because radon rarely occurs geologically there, based on NY State Health Dept. data (Mulder). But the Well system still requires this unnecessary radon testing. (At this writing, LEED and Well accept only building-specific data, not city-wide data.[9])

Building-specific energy and water consumption or air quality can be measured by meters and testing devices, or derived from utility bills, as in Energy Star. In LEED, project teams must hire licensed, professional testers, and the Well program sends their own verifiers to take measurements.

This leads to the question of how data can be verified for accuracy in GreenBlocks. In other words, what would prevent a building owner from uploading false data to the blockchain? One possible solution is to allow certain critical data to be submitted only by licensed professionals such as the project architect or engineer, who would be risking their license if false data were submitted. However, the objective of this thesis is to test the feasibility of a blockchain rating system. Administrative details on who would submit data or how data could be verified are, for purposes of this thesis, considered to be part of a future implementation phase once feasibility of GreenBlocks is established.

DNA Profile

Buildings are as complicated and unique as humans, and GreenBlocks aims to emulate in buildings the concept of a unique DNA identity, so to speak, that identifies its environmental characteristics. GreenBlocks' software generates a unique hash code that is placed on the blockchain via a smart contract to permanently identify the property address, the building metrics, zoning, and quantify the salient environmental characteristics of a building. Because of this, and the large amount of data accepted by the blockchain, each building in GreenBlocks has a unique profile.

This uniqueness is another departure from traditional rating systems. For example, Energy Star and HERS (Home Every Rating System) assign properties a numeric score representing energy performance (EPA, 2019). Conceivably, thousands, if not millions, of buildings could have the same score.[10]

Dynamic vs. Static Data

Information in GreenBlocks can be updated periodically to make it dynamic instead of static, with traditional rating systems, by contrast, being mostly static.

In this context, static means that in traditional systems like LEED, projected data on a criterion such as energy consumption is estimated by a computer model. Once the LEED plaque is awarded, the LEED certification body never reviews the project for any changes in energy consumption, or to see if actual energy consumption meets projections (although they reserve the right for such a review). GreenBlocks is designed to be updated periodically with actual consumption data based on onsite meters and utility records.

Transparency

Most traditional certification systems come up with a final numeric rating for a building and sometimes a "Silver/Gold/Platinum" rating. However, the public cannot find out details of energy use, ventilation or other stats unless the building owner wishes to release them.[11] By contrast, auto specs are made public. Consumers who want a green car can find excellent reviews by *Car & Driver Magazine* (2018) or American Automobile Association (AAA, 2018). Although the blockchain is an encrypted system, annual specs about participating buildings could be released, and why a building was "Gold" would no longer be a secret.

Accuracy

Green rating systems that use a ranking system with an award (as most do) have been found in many cases to reward buildings that are not green at all. In a famous 2010 case, a tenant sued developers of a LEED Gold building because their apartment was freezing, and investigation found that the wall assembly did not even meet local building code (Del Percio, 2010). In 2011, a person named Henry Gifford filed a $100 million suit against U.S. Green Building Council for fraudulent misrepresentation, alleging that LEED buildings in fact perform worse than conventional buildings in energy consumption (Alter, 2011). In a third example, a recent study found that houses following traditional design managed to outperform 6- and 8-star buildings under the Green Star rating system (Daley, 2018).

Experts usually account for these discrepancies between rating and performance by the fact that "most ratings tools are predictive," rather than based on the building's actual performance (Daley, 2018). GreenBlocks does not allow predicted performance unless it is replaced by actual performance as quickly as it becomes available.

False Data

Green ratings systems have, at times, been the targets of misrepresentations, exaggerations and even hoaxes.[12] LEED is described by USGBC as “self-certifying,” meaning that building owners and project teams upload their own data without verification by the certifying body. This can open up certification systems to unintentional acceptance of false data.

Some rating systems award points to projects that use carbon offsets. Unfortunately, fraud and corruption in carbon offset programs have occurred, and Natural Resources Defense Council has written a consumer protection guide on the subject (NRDC, 2016).

Although all data can be subject to human manipulation, GreenBlocks relies on local water codes, utility meter readings, surveys filed with the city, building plans filed with the department of buildings, municipal inspections, and so forth; not just representations by owners of what they plan to do. A requirement could be placed that only licensed professionals are allowed to submit certain types of sensitive data. And because GreenBlocks is an alpha-numeric data system, not a reward system, the incentive for deception to get the reward can be removed.[13] USGBC, developers of LEED, say on their website, “The more points, the higher the reward. With LEED, there are many rewards” (Samarasekera, 2017). GreenBlocks offers no rewards, only facts, which could be construed as yielding their own rewards, more subtle than points.

Quality controls on the type of data GreenBlocks will accept from which sources is an administrative detail to be resolved in the future implementation phase.

Cost of Green Certification

Costs of certification can be considerable. For example, this thesis author's client, a 4000-sf urban store, decided in April 2018 not to certify with Green Globes because the fee alone was $12,000. Unfortunately, this is not an uncommon decision at such a price point. In this author's personal experience, some of the greenest buildings are never certified in any rating system, because the owners would rather spend money on green features than on paying the certifiers. A prime example is headquarters of The New York Times, created as a model of green design without certification by any green rating system.

In a blockchain green rating system, certification fees would not be paid to a central authority. Certification fees in Green Globes, for example, are paid to finance an on-site inspection by a Green Globes rep, and also to pay for personnel time to review the data submitted and determine if it meets Green Globe's criteria.

Another cost of certification is environmental consultants to advise and verify compliance with the rating system. In GreenBlocks, a construction team probably still would need to hire their own green consultants on the job site - the same way they hire all kinds of experts, like lighting specialists and interior designers. However, the consultants' role would not be to determine compliance with a particular rating system, but to advise on ECMs (energy conservation measures) and other green features of the project.

Ethereum charges small fees for currency transactions, which at current exchange rates are less than one dollar per transaction. (BitInfo, 2018). Fee structure will be investigated later, as GreenBlocks is implemented after completion of this thesis.

Goal

GreenBlocks is conceived as a next generation green building assessment including granularity and kinesis (in lieu of stasis), peer review, etc. The goal of this research is to determine if an alternative green rating system can be effective, using a specific building as an experimental subject to test the hypothesis. A second goal is to determine if the blockchain is the ideal place to collect, store and distribute such data. Deliverables are a study narrative with a pilot system, along with an analysis of whether Ethereum can host a new, decentralized green rating system.

Background

This thesis brings together two technologies, green rating systems and the Ethereum platform with data stored on blockchain. Both of these technologies have a rich background before being developed in the same application. This section of Chapter I described the respective backgrounds.

Green Property Rating

"Buildings account for 39% of CO_2 emissions in the U.S." (USGBC) - even more than transport or industry. Considering that CO_2 emissions are the leading indicator of global climate change, this is an alarming statistic. In recent decades, environmental consciousness has been aroused, and in 1990, the building rating system BREEAM (Building Research Establishment Environmental Assessment Method), was established in England. BREEAM, which scores points for a building's environmental achievements, has been followed by thousands of point systems worldwide.

Consumers of environmental rating systems are construction industry professionals (developers, architects, contractors, etc.) and businesses, governments or families who buy, rent and inhabit structures.

Administrators of rating systems include government entities (like EPA, with Energy Star), trade associations (like National Association of Homebuilders, with National Green Building Standard), or private companies, sometimes with 501c3 status (like U.S. Green Building Council, with LEED). These are certainly centralized groups, with USGBC reporting $65 million income on their most recent annual report (USGBC).

These rating systems are said to have contributed to raising public consciousness about the environment. In recent years, a majority of consumers say they have resolved to live a greener lifestyle (Sustainable Brands, 2014), and 72% of firms say they participate in sustainability (Bureau of Labor Statistics, 2018).

Given the positive impact of existing rating systems, it is never the intention of this thesis author to denigrate traditional systems or their developers. The purpose of this thesis is to explore enhancements, refinements and alternatives to those systems, while pursuing the same environmental ideals.

Systems Are Inter-Related

Nearly all major green building rating systems in America, such as LEED, Green Globes, Living Building Challenge, Well, Enterprise Green Communities and National Green Building Standard use the same premise. The premise is that an authority assigns points if a building meets certain criteria chosen by that authority. Building performance is evaluated in categories like energy use, water efficiency, air quality and so forth. (Energy Star is limited to energy and water consumption.) Because each group has certain mission priorities, performance criteria are weighted by relative importance. For example, Well is explicitly oriented to human health, LEED prioritizes climate change, and Energy Star, as the name implies, emphasizes energy.

Since the introduction of the BREEAM system in England in 1990, few have questioned the fundamental premise of a centralized third-party authority imposing a green rating system. In fact, rating systems actually keep recycling each other's ideas. A green educational website explains that Green Globes "was based off of BREEAM, which was developed in the UK in the 1980's. Many other Green Building rating systems, including LEED, were based off of BREEAM" (Fuller, 2016). Living Building Challenge (LBC) and Well could both be called "spinoffs" of LEED and BREEAM, because LBC was created by the Canada chapter of USGBC/LEED, and Well's CEO is the founder of LEED. A glance at other, unrelated systems like Enterprise Green Communities reveals many similarities to LEED, including most of the same point categories. So close is the resemblance, that one of this author's clients, a multi-family architect, referred to Enterprise Green Communities as "LEED Lite."

Industry Leadership

With hundreds of competing rating systems worldwide, no single system has emerged as the definitive arbiter. Even General Services Administration, the world's largest property manager (GSA, 2018),[14] does not rely on only one rating system. Although GSA recognizes LEED and Green Globes, their official position is that these are just two of many tools for evaluating building performance, and government agencies are welcome to use the green rating system of their choice (Wilcox, 2013).

No green building rating system is perfect, and LEED and Energy Star certainly have their detractors. But an imperfect system is better than no system at all; without any criteria for determining if a property is environmentally friendly, all assets would be treated equally. One objective of GreenBlocks is to provide more information, transparently, to the consuming public.

History of Blockchain

"An explosion in the cryptocurrency market" has occurred in recent years according to *Fortune Magazine* (Ingram, 2017). Senator Thomas Carper, Chairman of Homeland Security Committee, told CNBC News, "Virtual currencies have captured the imagination of some, struck fear among others, and confused the heck out of the rest of us" (Ranasinghe, 2013).

The first recorded cryptocurrency purchase was May 22, 2010 when a man in London bought two Papa John's pizzas in Florida for 10,000 units[15] of Bitcoin (Evans, 2016). Today, Microsoft, Expedia, Whole Foods and thousands of other retailers accept Bitcoin. As of 2019, some 2000 different cryptocurrency systems developed, with Bitcoin and Ethereum by far the industry leaders (CoinLore, 2019).

If an individual wishes to make a purchase with cryptocurrency, they simply establish an online "wallet" (a file), buy electronic virtual coins using a credit/debit card, and check out using the "pay with Bitcoin" option.[16] A coin purchase is as simple as any ordinary online purchase. The process is scalable for large transactions, in millions or billions, such as a real estate acquisition.

Fiat and Cryptocurrency

Conventional currency, like the dollar, is backed by the full faith and credit of the U.S. Government and regulated by a central authority, the Federal Reserve. These currencies are called "fiat," from Latin for "let it be done." Consumer electronic transactions in fiat currency are administered by central groups such as PayPal or Visa.

By contrast, cryptocurrency is not regulated by a central authority, but rather democratized. Instead of a few people – notably the Fed Chairman – making policy decisions, cryptocurrency is verified by a large peer group network. Currency values are entirely market dependent. Transactions are recorded in a public ledger (blockchain) maintained by a consensus of independent peer computers (miners) using a common algorithm.

Real Estate

Cryptocurrency is used to purchase everything from pizza to airline tickets - including real estate. Several companies have begun to experiment with connecting real estate to digital currency. A notable one, involving Ethereum co-creator Joe Lubin, is called Meridio, a project creating a blockchain platform for shared property ownership (ConsenSys, 2018). Other real estate companies capitalizing on Ethereum's versatility are structured as listing services, REITs, brokerages or home sharing sites.

Blockchain for Property Rating

A logical next step is using the blockchain to certify green real estate – the purpose of this thesis.

The versatility and complexity of the Ethereum platform potentially may be revolutionary for certifying green performance. In a system based on stored mathematical data, environmental standards also can be quantified, recorded and verified mathematically. Just as the blockchain eliminates central authorities and excludes middlemen, Ethereum can be used to rate green buildings without central authorities or middlemen, and with minimal subjective judgments by influential people.

The online resource *Investopedia* explains Ethereum's unique capability well.

> "Ethereum can be used to codify, decentralize, secure and trade just about anything. The primary purpose of Ether is not to establish itself as a payment alternative (unlike Bitcoin) but…to enable developers to build and run distributed applications (Bajpai, 2018)."

This raises the question of why the blockchain is being tested in this thesis, when large amounts of data can be stored using conventional computer systems. The reasons are:

- Blockchain databases are maintained by a peer group, not a central authority like PayPal or Amazon.
- All transactions must be verified by the peer group, reducing the possibility for an individual tampering with the database.
- Verification of transactions by the peer group allows the data to be updated more easily, making the system potentially dynamic instead of static (as traditional rating systems are).

- Smart contracts can be used, which are a technology unique to Ethereum.

Other Initiatives for Blockchain

Because the blockchain is a ledger that can record any type of numeric data, no matter how complex, it would seem that every industry would gravitate to this new opportunity. Indeed, many articles have been written about use of the blockchain for documenting environment-related statistics (O'Connell, 2017). The following graphic figure, titled Sectors Where Blockchain May Be Used, organizes in a circular scheme the myriad application of the blockchain, many of which could be linked to the environment (Walker L. , 2017).

Types of environmental performance which could be tracked include:

- Land titles – the first such suggestion (Buterin, 2013) [17]
- Supply chain & provenance of sustainable products (Vorabutra, 2016)
- Green facilities management (Aamidor, 2017)
- Renewable energy use - wind or solar (Henderson, Knoll, & Rogers, 2018)
- Building automation - to save energy (Saleem, 2018)
- Demand response - for cities to use energy at certain times (NIH, 2018)
- Corporate social responsibility reporting (Stoddard, 2018).

- RECs, renewable energy credits (Henderson, Knoll, & Rogers, 2018)
- Carbon tax (Ivanova & Gilliland, 2017)

This thesis author has tried mightily to find, among the creative ideas floated for applying the blockchain to the environment, a single suggestion for using the blockchain to quantify green building attributes leading to rating. None were found as of the submission time of this thesis. A recent online article suggested using the blockchain for reporting a "company's environmental, social and governance (ESG) policies," holding up LEED as an ideal without ever mentioning building rating on the blockchain (Stoddard, 2018). This is the closest, apparently, that either environmental or blockchain experts seem to have come, at this writing, to suggesting the use of the blockchain for building rating.

Naturally, this raises the question of the meaning of, "tried mightily to find" such suggestions. It is incumbent upon the thesis author to list the range and type of materials reviewed.

Because development of the blockchain is so recent, and the field so dynamic, monographs are probably not a good reference source for research. Instead, the following data sources were consulted.

- Journals (with emphasis on blockchain and green building)
  - *The New York Times, NY Real Estate Journal, Building Green*, *High Performance Buildings*
- Conference proceedings
  - Greenbuild, Green Building Expo

- Environmental news feeds (RSS)
    - *EcoWatch, Mother Nature Network*, *Global Zero-Energy Buildings Market 2017-2021, Treehugger, Greenbiz, Inhabitat, Grist, Green Architecture & Building Report*
- Films and Videos
    - Netflix, YouTube, Hulu, Amazon Prime, Daily Motion, Ted Talks
- Courses
    - Stanford, *Coursera on Cryptocurrency*
    - Dapp University, *Intro To Blockchain Programming (Ethereum, Web3.js & Solidity Smart Contracts)*
- Interviews with USGBC officials (off the record)
- Ethereum Applications lists
    - State of the Dapps, Kintu.co, Consensys.net, Defiprime.com, Block123.com, Coinsutra.com, Coindesk.com, Medium.com, Blokt.com, Hackernoon.com

Of the resources in the preceding list, the category with best information on current Ethereum projects is Ethereum Applications lists. Several websites are devoted to tracking new applications being developed on the Ethereum platform. Although these sites do list some applications connected to real estate, most such apps involve listing or payment systems. At time of writing this thesis, no applications could be found under development for rating green buildings.

Many explanations for the absence of linking building rating and blockchain are possible. It may be no one has thought of the idea, no one finds it beneficial, no one finds it feasible, or perhaps powerful centralized organizations like Energy Star or U.S. Green Building Council oppose such an initiative. The true reason is not known to this thesis author, nor does it seem to be readily discoverable.

To compensate for lack of exploration of using the blockchain for green building rating, the objective of this thesis was to do original research to explore the feasibility of creating an alternative to traditional building systems, and to determine if Ethereum is the ideal platform for such a system.

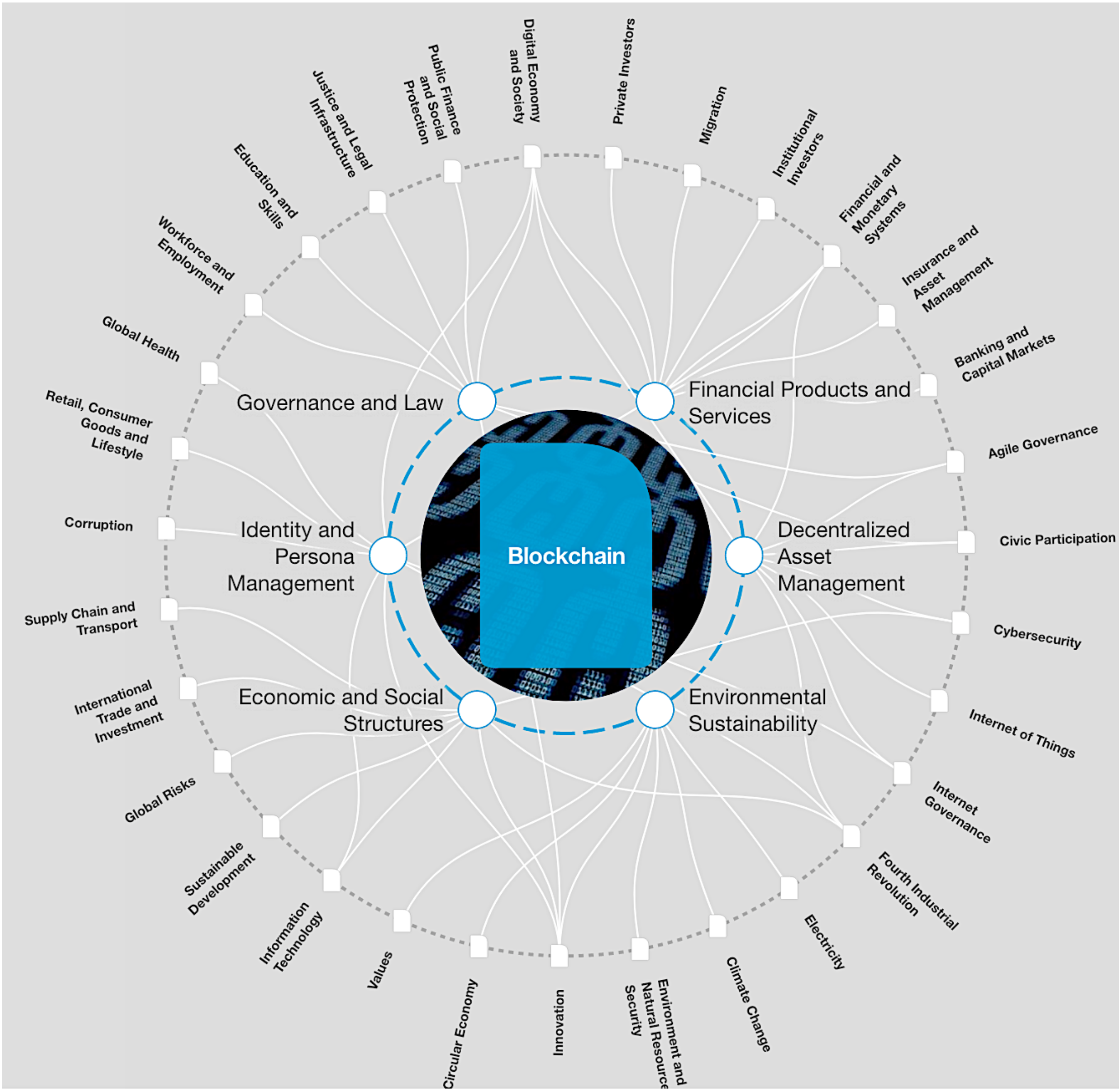

Figure 2 *Sectors Where Blockchain May Be Used*

Research Questions and Hypotheses

"All good research starts with a question…[which] involve[s] being skeptical or suspending belief about something, not automatically believing something just because it has been stated" (Harvard, 2012).

This thesis author has often questioned why so many building rating systems worldwide are all based on the same premise: an invisible central authority awards points for certain arbitrary environmental achievements.

Accordingly, the following questions arise with regard to traditional green building rating systems:

- To what extent is rating by a central authority the appropriate way to evaluate a property's environmental integrity?
- How are the evaluators evaluated, if at all?
- What are the intrinsic biases of the central authorities, and how can these be removed or minimized?
- Why is the number of criteria in the rating systems so limited?
- Why is performance binary rather than on a continuum? In other words, why do rating systems ask; "did the building meet/not meet an arbitrary performance threshold" rather than asking what was the building's exact performance metric?
- Why do traditional rating systems provide the public with a single piece of data, such as a score of "Gold," as opposed to releasing

complete building specifications, the way a manufacturer releases complete car specs?

- If the public seeks complete details on any product they buy, from toasters to bicycles, by reading *Consumer Reports* or *Good Housekeeping*, why should something as important as building science be represented by only a metal plaque, with the details kept private?
- To what extent can these concerns be resolved without use of the blockchain? What makes the blockchain useful in the context of this study and to rating of building performance in general?

In the past two or three decades, few if any critics have asked these questions. The last question, of course, could not be raised until recently, because blockchain technology as well as the process of composite, quantitative indexing of building environmental performance is still quite young. Only in the last year or two have blockchain and cryptocurrency activities matured to the point at which these technologies may possibly be applied to decentralized green building rating.

Therefore, the primary research questions in this thesis were: *To what extent* can *a viable new green building rating system be created which represents an improvement over traditional systems? And, in this context, to what extent can the blockchain and smart contracts offer an improvement over traditional computer information processing?*

The *ALM Thesis Guide* (p. 15) says to review: “Has someone already answered the question satisfactorily?” The technology here is so new as to be developing at an hourly rate. A review of internet literature reveals no records of anyone reporting or blogging about changing the basic premise of existing green rating systems by introducing blockchain technology.

Also critical is whether any teams have the expertise, motivation and wherewithal for such a project. Creating a green building rating system on the blockchain would require a team with expertise in architecture/engineering, real estate, building rating, computer science, the Ethereum (or equally powerful) platform and data mining. It would also require the financial resources for such an effort. It is quite likely that, at this writing, few such competing teams exist.

This thesis tested two outcomes: a) the feasibility of creating an effective new rating systems without using the traditional, existing rating system model, and b) that this system can be effectively processed and maintained on the Ethereum platform.

The hypothesis: ***It is feasible to create an effective green building rating system with a new and different approach, and that system can be successfully processed, maintained and delivered on the Ethereum platform.***

By creating a pilot rating system on the blockchain, the thesis project examined how a decentralized, peer-approved green building rating system can be an alternative to traditional, centralized systems.

Chapter II Research Methods and Characteristics of GreenBlocks

The main research method in this thesis was to create a pilot green building rating system, called GreenBlocks, on the Ethereum platform and test it using live data from a large new green construction project in Las Vegas. It was not necessary to upload complete construction data to discover if the pilot system was feasible. Therefore, to avoid unnecessary labor, approximately 20% of a complete project's data, mostly related to the site, was uploaded. The specific data uploaded is listed in Appendix I.

This chapter offers details on the research method and describes the characteristics of GreenBlocks.

Methods

Objective

The objective of this thesis is to create a pilot green rating system using Ethereum to test the hypothesis. The hypothesis is: *It is feasible to create an effective green building rating system with a new and different approach, and that system can be successfully processed, maintained and delivered on the Ethereum platform.*

The objective of this thesis ends with creation and testing of the program. In future, it may be possible to refine, scale, test and implement such a system in the construction industry. This would be a large project which may take years to complete. LEED, for example, took over two years for a large team to develop, even though it was based on an already existing system, BREEAM. Therefore, the objective is limited to testing the hypothesis.

Deliverables

Developing an original building rating system is an ambitious undertaking. The development of LEED took two years of work by a committee, even though LEED was based on an already existing system, BREEAM. Therefore, to keep the scope of this thesis within manageable size and time parameters, the deliverable is a pilot rating system, GreenBlocks, which has been tested on one subject project (mixed-use new construction in Las Vegas).

Deliverables are included in the appendices:

- GreenBlocks proposed criteria
- Details on How to Build Blockchain App and Ethereum To Do List
- Technical description of how GreenBlocks was developed.

Subject Property

The research method involved a live project, not a theoretical study. It also involved a large commercial property valued in the hundreds of millions of dollars. The building will be a new construction, with readily available data. Data specific to the subject property came from due diligence documents, civil engineering/site plans and arch/eng plans. Generalized data came from local and state codes and standards. The thesis author was responsible for all data collection and processing.

It is likely the building will be LEED registered; whether it will qualify for LEED certification is unknown. (LEED registration means the building has applied, or signed up for the LEED rating system. Certification means the building has earned enough points to be considered green under the LEED system.)

Project Consultancy

The project involves technical material. Perhaps a possible reason no one can be found who is attempting a similar project is the project's size and intricacy.

This thesis author was involved as both real estate broker and environmental expert on a large new mixed-use construction project in Las Vegas, which served as the subject building for developing the pilot green rating system, GreenBlocks. This student author conceived of the thesis because of her extensive experience as an architectural consultant using most of the major green rating systems in existence today.

For the thesis, each step of the process of creating and testing a green building rating system was documented. The ultimate objective, however, is not simply to complete a university thesis, but to test the feasibility of creating a scalable rating system that can serve the international real estate and construction communities, as well as consumers.

Because this thesis author has studied Ethereum extensively but is not an expert, an outside consultant was required to perform certain technical services. As documented in Appendix IV, the Ethereum experts provided instruction to the thesis author, who built and tested GreenBlocks herself. The thesis author also worked closely with the Ethereum consultant to evaluate the success of the feasibility study and hypothesis test.

Data Sources

Data sources for the building and site used in testing GreenBlocks were:

- Evaluations and inspections by local authorities (or accepted by local authorities)[18]
- Field measurements by licensed experts
- Architectural schematics and renderings
- Maps, photos, surveys, etc.
- Title searches, Dept. of Buildings records, deeds, surveys, appraisals
- Federal Aviation Administration correspondence
- Local building codes, zoning, environmental and water standards
- Local climate, topographic, hydrologic and geologic data

Characteristics of Pilot Rating System, GreenBlocks

GreenBlocks is intended to:

- List key environmental specs of a site and/or a building (listed in the Appendix)
- List non-environmental specs which can affect a building's green potential (e.g. a building that is landmarked or has buildable air rights may have hidden opportunities not recognized in traditional rating systems)
- Remove, to the extent possible, subjectivity, making rating a mathematic/scientific exercise

- Remove rewards or incentives in favor of objective, unfiltered facts
- Provide, to the extent possible, objective, verifiable metrics that any tester could replicate (For example, anyone could calculate the same energy use intensity based on the same square footage and same consumption data.)
- Provide, if possible, for dynamics as a building matures, which other systems do not[19]
- Account for differences in testing methods and differences in devices and calibrations hopefully by introducing a means for standardization
- Reduce or eliminate certification fees
- Decentralize by storing data on the blockchain instead of submitting it to a third party for approval

Criteria

A preliminary list of the type of project criteria recorded in the blockchain appears in Appendix I: Examples of Specifications.

Chapter III Hypothesis Testing and Results

The purpose of this thesis is hypothesis testing, which is reported, along with results of the test, in this chapter.

Hypothesis Testing

The thesis tested two outcomes: a) the extent to which it is feasible to create an effective new rating systems without using the traditional, existing rating system model, and b) the extent to which this system can be effectively processed and maintained on the Ethereum platform.

The hypothesis: ***It is feasible to create an effective green building rating system with a new and different approach, and that system can be successfully processed, maintained and delivered on the Ethereum platform.***

Of course, every hypothesis has a null, listed in the table below.

Hypothesis	*It is feasible to create an effective green building rating system with a new and different approach, and that system can be successfully processed, maintained and delivered on the Ethereum platform.*
Null hypothesis	A feasible system cannot be created, and/or it fails to meet the intended goals which distinguish it from existing rating systems.
Alternative hypothesis	A feasible system can be created with a new, different approach, but it cannot be supported on the Ethereum platform.

Feasible, of course, means possible to do easily or conveniently. A feasible system means that the system can work within reasonable constraints of technology, labor, security, complexity, and other parameters. The purpose of this thesis was to determine if GreenBlocks is feasible in the technological sense, not a business sense. No attempt was made to analyze financial costs or to predict market acceptance of GreenBlocks as a product.

The thesis has three elements:

1. Comparison of relative merits of a new means for rating building sustainability as opposed to traditional rating systems (e.g. LEED, BREEAM, Green Globes, National Association of Homebuilders, et al.)
2. Comparison of the relative merits of Ethereum over traditional computer data systems
3. Balancing the foregoing objectives in a synthesis (i.e. how the specific advantages of the new rating system emerge, and how the blockchain (specifically Ethereum) is a useful platform to store, verify and convey large amount of technical data.

The hypothesis is considered supported if a basic pilot system can be constructed and applied to a subject building, and, further, environmental building data can be maintained in a blockchain on the Ethereum platform. If the hypothesis is supported, a new tool may become available in the construction industry.

The null hypothesis is supported if it is found that Ethereum cannot process, maintain or convey the data properly and efficiently. Perhaps Smart Contracts may prove not to be an efficient way of processing the data. Another reason for a null outcome could be a breakdown of the pilot program itself; if it is found that the sheer volume of data, or data format or weighting of data, for example, make the pilot system unfeasible for describing and rating green buildings.

The purpose of this thesis is to examine technological feasibility, not social or financial feasibility. It is not within the scope of this thesis to assess potential public acceptance of the new system, or to estimate costs of setting up or delivering the system.

The alternative hypothesis is that a new system can be created, but it cannot be supported on Ethereum. However, even if the decentralized blockchain is inappropriate for data storage, the concept of an objective, numeric, decentralized non-reward system (similar to car specs) possibly could be used with conventional computer systems.

It is believed that any of the three outcomes have academic value. With a null outcome, the possibility for an unbiased system – such as the owner of Empire State Building called for – has been, at least, explored. Under the hypothesis and alternative hypothesis, development of a viable new green rating system could ideally be a useful result.

Results

To execute this thesis project, the author created a green rating system called GreenBlocks, which is basically a list of value characteristics of the site and the proposed building. The author, with instructions from technical experts, created the smart contracts and blockchain apps that would be used for GreenBlocks data. Consistent with the commitment to data transparency which underlies this thesis, the complete, verbatim steps for this action appear in Appendix IV.

The exercise of creating GreenBlocks proved that this thesis author, a relative newcomer to Ethereum, could develop from scratch a feasible green rating system using blockchain technology and smart contracts. The next step was to extract factual data meeting GreenBlocks criteria for the subject building, and upload it successfully to the blockchain. This data came from due diligence documents, proposed arch/eng plans and local codes.

The conclusion from these actions was that the hypothesis of this thesis was supported. GreenBlocks showed that: *It is feasible to create an effective green building rating system with a new and different approach, and that system can be successfully processed, maintained and delivered on the Ethereum platform.*

The purpose of this thesis is hypothesis testing, which is reported, along with results of the test, in this chapter.

Chapter IV Conclusions

With the hypothesis supported, this thesis is complete. The procedure of creating the pilot system GreenBlocks as well as blockchain apps and smart contracts, and loading data sustained the hypothesis that: *It is feasible to create an effective green building rating system with a new and different approach, and that system can be successfully processed, maintained and delivered on the Ethereum platform.*

GreenBlocks, as created, stands in dramatic contrast to traditional rating systems in the following respects:

- Decentralization: The blockchain is, by its nature and structure, decentralized. GreenBlocks does not use a central authority to process or judge data submissions, nor to give merit awards.
- Objectivity: The vast majority of the data used in GreenBlocks is quantitative and objective, and contains little to no relativity or value judgments.
- Adaptability: GreenBlocks could apply to any type of building.
- Clarity: Most data used in GreenBlocks requires no specialized professional knowledge. Anyone with the expertise to work on a certain aspect of construction, like an electrician, can submit the electrical data required by GreenBlocks. This is in contrast to traditional systems which usually require an expert to explain the rating system itself, such as a LEED Accredited Professional (AP), Well AP, Green Globes Professional (GGP) or Living Future AP.

- Continuum: GreenBlocks data is based on definite, quantifiable numbers, not minimum quotas or thresholds for "success."
- Wide Range of Data: GreenBlocks allows for submission of thousands of pieces of information, not the relative few facts selected by traditional systems.
- Numeric: The rating system did work out to be primarily based on numeric data.
- "DNA Profile": The technology does create a unique profile for a property. The extent of data uploaded to GreenBlocks is, however, up to the building owner. This is because GreenBlocks is decentralized and has no central authority to require owners to contribute, or not contribute, certain types or amounts of data.
- Dynamic Data: The data in this system definitely can be updated as often as desired. However, this raises the financial question of how updates are paid for.
- Transparency: GreenBlocks data is totally transparent, unless users request certain very limited information to be kept confidential for security reasons.
- Accuracy: GreenBlocks relies on actual, rather than predictive data. If predictive data were to be used, it would be clearly identified as such.
- False Data: Data is only as good as its source. If a person on the building owner's team were determined to commit fraudulent misrepresentation, they could potentially submit inaccurate information to GreenBlocks. However, this would be more difficult to do with GreenBlocks than with traditional systems, because so much data about the building is included, that it could call attention to flagrant misrepresentations. For example, traditional systems might ask for a room's noise levels in decibels. GreenBlocks might ask for the exact same information. But GreenBlocks asks for so much supplementary factual info, such as wall

construction details, type, quantity and R-value of insulation and windows, etc., which would make it possible for a building science expert to know if decibel readings were reasonable, given the room's construction. As an analogy, if a dishonest person reported that a car gets 50 miles per gallon highway gas mileage, but reported elsewhere in GreenBlocks that the car has heavy weight, high air displacement, V-8 engine, poor aerodynamics, and high mechanical resistance, an astute reader would immediately see that something must be wrong with the 50 mpg figure. In a traditional system that awards points for gas mileage without asking about the car's other characteristics, a lie would be much easier to submit.

- Cost: The most complicated part of this project was to determining how to allocate costs. Traditional systems not only have high certification fees but also externalize many large costs to the consumer (building owner). Although GreenBlocks is not conceived with certification fees (because there is no "certification"), the question of how labor costs are calculated and paid remains to be resolved in future studies.

Although testing found the hypothesis to be supported – that GreenBlocks is technologically feasible on the blockchain – the testing raised interesting questions outside the scope of this thesis. It was discovered that this system is much more labor-intensive and therefore costly for the administrators than traditional systems. In traditional systems, the bulk of labor, labor costs and other costs are borne by the building owner. So, division of responsibilities and costs must be distributed fairly for the system to be feasible in a business sense.

In traditional systems like LEED, Well, Green Globes, et al. customers (building owners) bear the costs of uploading data to the online centralized authority. Customers also pay to hire consultants, such as Accredited Professionals to interpret and coordinate the system's requirements, Engineering/Computer experts for energy modeling, and Commissioning Agents (CxAs) to verify systems. Using LEED as an example, Accredited Professionals typically charge between $.50 - $3.00 per square foot (depending on project size), and CxAs are in a similar range. Commissioning usually starts at $15,000 for a small project under 10,000 square feet (Nicolow, 2008).[20] Energy modeling costs $0.05 to $0.45 per square foot, depending on project size, or in dollar amounts, $15,000 to $30,000 per project (American Chemistry Council, 2003).

The costs of collecting and uploading data are more difficult to quantify, as they are typically performed by project workers as an addition to their regular work. However, USGBC estimates the cost of documentation at $30,000 - $60,000, regardless of project size (USGBC, 2002).[21]

On top of this, customers must pay for manuals, reports and most importantly, certification fees. Certification means that a central authority reviews the projects documents, comments on them, and awards green certification.

In LEED, for example, a building the size of the Las Vegas project would cost $1500 to register (sign up). On the LEED Certification Fees table reproduced below (USGBC, 2020), USGBC does not release certification cost of projects over 700,000 sf, saying customers must "request a quote." However, according to the LEED Certification Fees table, certification for buildings 500,000 - 700,000 sf is $.60 per square foot, a sizeable expense by any measure. The table does not state a maximum fee or a cap. It is not known what certification fees would be for the subject project in Las Vegas, which is over 9 million square feet, although LEED does have a "volume" program for registering more than one building at a time. For a project of this size, it appears the certification fee would be negotiated.

Regardless of project size or the fees for any specific project, one client of this thesis author described LEED registration and certification fees as "astronomical," considering that the fees include no site visits, technical or design assistance, or verification – expenses which are entirely assumed by the consumer. Despite LEED's worldwide recognition as a green building standard, certification fees have been a deterrent. As of 2019, only 100,000 buildings were either LEED registered (signed up) or certified (Stanley, 2019).[22] To put this in perspective, New York City alone has 1 million buildings (NYC Dept. of Buildings, 2020). If all LEED buildings were in New York City, it would only be 10% of just one city's inventory.[23]

Building Design and Construction Fees

Building Design and Construction Fees per Building	Silver, Gold and Platinum Level Members		Organizational or Non-members	
Registration	$1,200		$1,500	
Precertification				
Flat fee (per building)	$4,000		$5,000	
Expedited review (reduce from 20-25 business days to 10-12, available based on GBCI review capacity)	$5,000			
Combined Certification Review: Design and Construction	**Rate**	**Minimum**	**Rate**	**Minimum**
Project gross floor area (excluding parking): less than 250,000 sq ft	$0.057 /sf	$2,850	$0.068 /sf	$3,420
Project gross floor area (excluding parking): 250,000 - 499,999 sq ft	$0.055 /sf	$14,250	$0.066 /sf	$17,100
Project gross floor area excluding parking): 500,000 - 749,999 sq ft	$0.050 /sf	$27,500	$0.060 /sf	$33,000
Project gross floor area (excluding parking): 750,000 sq ft or greater	Request a quote		Request a quote	
Expedited review (reduce from 20-25 business days to 10-12, available based on GBCI review capacity)	$10,000			

Figure 3 - LEED Certification Fees 2020

An additional – and even greater – cost incurred in compliance with traditional rating systems is the technologies, products and design/construction techniques needed to meet the rating system's requirements. GreenBlocks has no additional design/construction costs for compliance whatsoever, because the system requires no compliance to any standard; it merely reports factual data on the attributes of the project.

The additional costs of green design and green construction are estimated at 5% extra for design, and 3-8% extra for construction, compared to conventional, non-LEED buildings (American Chemistry Council, 2003). These estimates are highly subjective, however. USGBC, who administer LEED, have consistently insisted that LEED costs “not one penny more,” than conventional construction, except for a platinum project, which is 6% more (Fedrizzi). On the other hand, some other experts say construction costs for LEED-compliant buildings are 30% more than conventional buildings.

To be sure, LEED certification is not free, and rating systems like Well, Green Globes, Living Building Challenge, National Association of Homebuilders and Enterprise Green have relatively similar costs. Nearly all of the costs described here do not exist in GreenBlocks. GreenBlocks does not require energy modeling, commissioning, expert AP assistance, registration/certification fees, or any special design/construction compliance. GreenBlocks does require users to spend some time on collecting documentation, but not as much time as traditional systems, which are said to be ‘self-certifying,’ so to speak.

The major expense in GreenBlocks is labor. The blockchain is complex, and is maintained by a small number of experts with very specialized knowledge and equipment. Self-certification, meaning that building owners upload their own data to a central website, is not possible in GreenBlocks. Instead, owners must submit data to blockchain experts, who enter data into the chain for them. Developers estimated the time investment of uploading building data to be 200-800 hours for a project the size of the subject in Las Vegas. These skilled workers must be compensated for their labor.

Because this thesis was limited to determining technological, not economic feasibility of GreenBlocks, no analysis was done of labor costs, nor how to structure costs or compensate workers. Having determined that the hypothesis was supported, the question of next steps arises. Hypothesis testing revealed that GreenBlocks is fairly labor intensive, and a next step, following completion of this thesis, must be to determine how to pay for the labor.

A stated goal of GreenBlocks development is to keep costs way below traditional, centralized rating systems. With the rough estimate of the many direct and collateral costs of LEED given in this conclusion, it seems readily apparent that a charge levied to GreenBlocks users to pay for labor could be comparable, or possibly less than the cost of traditional rating systems.

Even if costs are made comparable to other systems, there is no guarantee that the public will see the value in using the blockchain for rating green buildings. Consumers will only pay for the service if it is capable of achieving strong market acceptance. Whether or not the public will understand and accept a program like GreenBlocks is an important variable that requires further study.

Now that testing on a live project is complete, and GreenBlocks has been found to be technologically feasible, it is possible that it might be developed to the point of becoming feasible financially. The GreenBlocks criteria in the pilot program are quite basic and would require expansion and refinement, preferably by an expert interdisciplinary committee, like the developers of traditional rating systems.

It may merit more study to learn if GreenBlocks has a future as more than a pilot project for academic exploration.

.

Appendix I GreenBlocks Criteria

<table>
<tr><th colspan="3">Project Title Information</th></tr>
<tr><th>Datum</th><th>Criteria</th><th>Value</th></tr>
<tr><td>Tentative project name</td><td></td><td>LVB Hotel, Casino & Residences Project</td></tr>
<tr><td>Current owner</td><td>Clark County Clerk APN records</td><td>Desert Land LLC</td></tr>
<tr><td>Project Type</td><td></td><td>Mixed-use, Multi-building development</td></tr>
<tr><td>Site identificatio n</td><td>Las Vegas Assessor Parcel Number (APN)
Title report, June 11, 2018 by 1st American Title Insurance Co.</td><td>(APNs) 162-28-202-013, 162-28-301-001, 002, 010, 029, 032, 033, 034, 036, 037, 162-28-302-001, and 162-28-310-001
<table>
<tr><th>APN</th><th>ADDRESS</th></tr>
<tr><td>162-28-202-013</td><td>95 East Ali Baba Lane</td></tr>
<tr><td>162-28-301-001</td><td>47 East Mandalay Bay Road</td></tr>
<tr><td>162-28-301-002</td><td>47 East Mandalay Bay Road</td></tr>
<tr><td>162-28-301-010</td><td>3953 Las Vegas Boulevard</td></tr>
<tr><td>162-28-301-029</td><td>N/A</td></tr>
<tr><td>162-28-301-032</td><td>3951 Las Vegas Boulevard</td></tr>
<tr><td>162-28-301-033</td><td>N/A</td></tr>
<tr><td>162-28-301-034</td><td>3965 Las Vegas Boulevard</td></tr>
<tr><td>162-28-301-036</td><td>3941 Las Vegas Boulevard</td></tr>
<tr><td>162-28-301-037</td><td>N/A</td></tr>
<tr><td>162-28-302-001</td><td>N/A</td></tr>
<tr><td>162-28-310-001</td><td>N/A</td></tr>
</table></td></tr>
<tr><td>Lot size in acres</td><td>Clark County Clerk APN records
Title report, June 11, 2018 by 1st American Title Insurance Co.</td><td>The site consists of 12 tax parcels totaling approximately 38.475 acres.</td></tr>
<tr><td>Real estate tax district of county</td><td>Clark County Treasurer</td><td>Paradise CC Redevelopment</td></tr>
</table>

Zoning	Clark County Land Use Planning Office	All parcels are H-1, AE-60&65 and MUD-1. AE is Airport Environs Overlay District. MUD is Mixed Use Overlay District 1. Unlike other major urban centers, Clark County does not have floor area restrictions that establish a maximum number of buildable square feet. This site lends itself to a very dense mixed-use project capitalizing on the many uses permitted under the existing zoning.
Percent previously developed	Defined as "Land which is or was occupied by a permanent structure"	70% of Property is either currently developed, under construction, or historically developed. Masonry block walls and/or fencing surround developed portions. A chain link fence is located along most of western portion of property south of Mandalay Bay Road.
County		Clark
City		Las Vegas
Neighborhood		Paradise
Zip code	USPS	89119
Professional Appraisal	Atelier Consultants, Inc. March 15, 2019	Total of all lots combined with approx. size of 38+ acres vacant land is valued at $19,737,917,314.
Professional Appraisal	Atelier Consultants, Inc. March 15, 2019	Land only $460,000,000
Deed limitations & restrictions	Deed	Drainage easement for McCarran Airport
Transportation access, road	Driving and parking data supplied by owner. Source of parking calculations: https://sciencing.com/facts-7576253-parking-square-footage-per-car.html	Street access through 750 feet of frontage on Las Vegas Blvd. (The Strip). Project will have 1,225,000 sf total of parking spaces (approx. 3500 spaces) in on-site parking garage.
Transportation access, public transit	Las Vegas Monorail Company, owner/operators	An extension to the Las Vegas Monorail is approved, which will stop at the subject site (retail sector) and proceed to McCarren Airport (2 stations).

Site Climate Information		
Datum	**Criteria**	**Value**
Latitude	The North American Datum of 1983 (NAD 83) is the horizontal and geometric control datum for US, Canada, Mexico & Central America. It was released in 1986.	36-05-43.26N
Longitude		115-10-13.20W
Altitude		2126-40 feet site elevation. Terrain slopes down to northeast
Nearest airport (for meteorological data)	McCarran International Airport, 5757 Wayne Newton Boulevard, Las Vegas, NV	
Climate	(Köppen climate classification BWh)	
Temperatures, average monthly high/low in degrees F	National Weather Service National Oceanic & Atmospheric Administration	Jan 58.0/39.4 F Feb 62.5/43.4 Mar 70.3/49.4 Apr 78.3/56.1 May 88.9/65.8 Jun 98.7/74.6 Jul 104.2/80.9 Aug 102/79.3 Sep 94/71.1 Oct 80.6/58.5 Nov 66.3/46.5 Dec 56.5/38.7
Humidity average by month in %	National Weather Service National Oceanic & Atmospheric Administration	Jan 45.1 Feb 39.6 Mar 33.1 Apr 25 May 21.3 Jun 16.5 Jul 21.1 Aug 25.6 Sep 25 Oct 28.8 Nov 37.2 Dec 45

Rainfall average per month in inches	National Weather Service National Oceanic & Atmospheric Administration	4.2 inches per year; 26 rainy days per year https://www.weather-us.com/en/nevada-usa/las-vegas-climate#climate_text_1 Jan 0.5 Feb 0.8 Mar 0.4 Apr 0.2 May 0.1 Jun 0.1 Jul 0.4 Aug 0.3 Sep 0.3 Oct 0.3 Nov 0.4 Dec 0.5
Rainfall average number of days per month	National Weather Service National Oceanic & Atmospheric Administration	Jan 3.1 Feb 4 Mar 2.9 Apr 1.6 May 1.2 Jun 0.6 Jul 2.5 Aug 2.6 Sep 1.6 Oct 1.7 Nov 1.7 Dec 3
Sun/Clouds		Average 310 sunny days per year https://www.weather-us.com/en/nevada-usa/las-vegas-climate#climate_text_1
Daylight/Sunshine hours average monthly	National Weather Service National Oceanic & Atmospheric Administration	Jan 10/7.9 Feb 10.9/8.8 Mar 12/10.1 Apr 13.2/11.5 May 14.1/12.5 Jun 14.6/13.4 Jul 14.3/12.6 Aug 13.5/11.9 Sep 12.4/11.2 Oct 11.2/9.8 Nov 10.3/8.2 Dec 9.7/7.6
UV index average by month	National Weather Service	Jan 3 Feb 4 Mar 6 Apr 8

	National Oceanic & Atmospheric Administration	May 9 Jun 10 Jul 11 Aug 10 Sep 8 Oct 5 Nov 3 Dec 2
Windspeed average by month in mph	McCarran Airport https://weather-and-climate.com/average-monthly-Wind-speed,las-vegas,United-States-of-America	Jan 3 Feb 4 Mar 5 Apr 5 May 5 Jun 5 Jul 5 Aug 4 Sep 4 Oct 4 Nov 4 Dec 3
Planting zones	USDA Plant Hardiness Zone and Sunset Climate Zone	USDA 9 and Sunset Climate Zone 11

Site Geologic Information		
Datum	**Criteria**	**Value**
Topography	Onsite notation	Site is flat with gentle slope of only about 10-20 feet.
Topography	US Geological Survey	Located in Las Vegas Valley, which is in Basin and Range Geomorphic Province of North America. Basin and Range topography is the result of Pliocene and Miocene east-west extensional tectonic movement (spreading) creating parallel north-south oriented block fault (normal faults) mountain ranges (horsts) with intervening north-south oriented desert valleys (grabens).
Vegetation	Onsite survey	Vegetation is severely limited because vacant parcels have been graded and cleared.
Vegetation, native to area	Hart Merriam Life Zones	https://www.birdandhike.com/Veg/HabType/Shad/_Shad.htm Upper Sonoran Life Zone in Mojave Desert has

		shadescale bushy scrub in higher elevation desert flats with poor drainage.
Soil type	Onsite samples Unified Soil Classification System	Soil type C, dense to very dense (defined as "soil that will allow a steel foundation probe to penetrate no more than 2 inches when manually pushed.") Clay, sand and gravel, with some gypsum
Soil characteristics	Lab tests for: • Moisture content • Specific gravity • Dry density • Atterberg limits (water content) • Compaction	Native soil, not compacted, moderate corrosivity
Presence/Absence of contaminants in soil	Onsite sampling and lab testing	None found as of August 2004
Water table level	Onsite borings	17-19.5 feet below grade
Groundwater conditions	Onsite sampling	Las Vegas Valley has aquifers recharged by surface water from surrounding Spring Mountains
Geology	General local conditions	In south central part of Las Vegas Valley, underlain by thick alluvial deposits, generally consisting of interstratified mixture of fine & coarse grains with irregular zones of cemented & granular soils.
Geological site class	Table 1615.1.1 of International Building Code	Site Class D. (It is possible to be Class C or B if seismic testing is performed at the site.)
Seismic conditions	General local conditions	No recent faulting; no recent earthquakes.
Drainage	Onsite notation	Natural drainage is surface sheet flow trending toward existing roadways
Drainage	US Geological Survey	Las Vegas Valley is drained northwest- to southeast-oriented Las Vegas Wash [known as an oasis, wetland or 'urban river'] which discharges into Lake Mead. Surface water flow is easterly from Spring Mountains toward Las Vegas Wash.
FINDS listing Yes/No	FINDS is EPA Facilities InDex System	Yes, but no violations or hazardous conditions found
RECs Recognized	ASTM defines REC in E1527-13	None

Environmental Conditions Yes/No and if Yes what they are	standard as "the presence or likely presence of any hazardous substances or petroleum products in, on, or at a property: (1) due to release to the environment; (2) under conditions indicative of a release to the environment; or (3) under conditions that pose a material threat of a future release to the environment. de minimis conditions are not recognized environmental conditions."	

Financial Information (supplied voluntarily at owner's discretion)		
Datum	**Criteria**	**Value**
Project name	Supplied by owner	LVB Hotel, Casino & Residences Project
Owner name	Deed	Desert Land, LLC
Hotel operator	Supplied by owner	Mandarin Oriental Hotel Group, International 888 Seventh Avenue, Suite 510, New York, NY 10106 1 (212) 207 8880 https://www.mandarinoriental.com
Nevada gaming operator	Supplied by owner	Magic City, 450 NW 37th Avenue Miami, FL 33125 https://www.magiccitycasino.com/
State tax concessions	Nevada Governor's Office of Economic Development	Property is located in Qualified Opportunity Zone. "The Opportunity Zone program, established in The Tax Cuts and Jobs Act of 2017, is a tax incentive, designed to encourage long-term private investment."

		https://www.diversifynevada.com/programs/opportunity-zones/
Real estate tax on parcels	Clark County Treasurer	Property tax rate in 2020 for subject tax district is $2.9328 per hundred dollars of assessed value.
Real estate tax district of county	Clark County Treasurer	Paradise CC Redevelopment
Assessed value	Clark County Treasurer, Assessment Roll	http://gisgate.co.clark.nv.us/secroll/secroll.asp
Real estate tax abatements	Nevada Revised Statute (NRS) 361.4723 provides partial abatement of taxes.	Property is not eligible, as it is a new construction.
Land acquisition purchase price	Clark County Recorder's Office 500 S. Grand Central Pkwy, 2nd Floor Box 551510 Las Vegas, NV 89155-1510	$812,000,000 (figure supplied by owner)
Transfer tax	Clark County Recorder's Office	*Transfer* tax rate is $2.55 per $500 of value
Estimated development cost	Supplied by owner	*Phase 1 development is estimated at this time. No estimates have yet been drawn up for future development.* *Phase 1 consists of:* 9,130,400 sf total (4 Hotel Towers with 5,948 keys, 1,380 condos) 750,000 sf Retail Podium Base 400,000 sf Food & Beverage outlets (21 restaurants) 500,000 sf Club/Lounges, Main Club with Pool, and 1 Casino *105,400 sf other* Complete development cost of the first phase is $6,307,000,000, and net profit after the 10 year hold period, and after the sell down of the retail component and sellable condominiums of the residential tower and the hotel keys, is estimated at $11,711,658,926 leaving the venture a profit of $6,203,758,126.

Estimated annual Net Operating Income (NOI)		Stabilized NOI from the owned assets is estimated at $1.2 billion annually.

Basic Building Information		
Datum	**Criteria**	**Value**
Project name	Supplied by owner	LVB Hotel, Casino & Residences Project
Number of buildings	Supplied by owner; confirmed by Dept. of Buildings & Safety construction permits https://www.lasvegasnevada.gov/Government/Departments/Building-Safety	*4 Towers total* Studies have been performed by architects to determine this property could accommodate 10,000 hotel rooms.
Tower height	Supplied by owner; confirmed by Federal Aviation Administration clearance letter	Federal Aviation Administration has provided determination letters for the site approving height allowance of 505 feet on west side fronting Las Vegas Blvd (The Strip) and decreasing to 354 feet on eastern portion of site. Rough calculations according to: https://www.convertunits.com/from/feet/to/story 505 feet = 46 stories 354 feet = 32 stories Tallest building on the site is planned to be 45 stories; smallest will be 32
Typical units	Supplied by owner	2 types of units in Phase 1: Penthouse and Luxury Penthouse Deluxe Plan A - 2 Bedroom + Den Phase 1 starting $4,200,000 Luxury Collection- 1 Deluxe Bedroom + Den Phase 1 starting $1,275.00 Luxury Collection- 1 Bedroom (Plan A) 1,530 SF Phase 1 starting

		$1,190.00 Penthouse Collection 1750 SF Phase 1 starting $3,325,000 The above are standard condo units. However, condos may be purchased as "grey shell," allowing customization by owners.

Building Systems		
Civil Engineering		
Datum	**Criteria**	**Value**
Storm water piping		
Water retention		
Sanitary systems (connections to sewer		
Natural gas		
Landscaping		
Erosion, sedimentation, dust, control		
Electrical		
Estimated Kwh use per year		
Service in volts and amps		
Power distribution system		
Generators, type, size and number		
Lighting		
Plug loads		

Elevators & escalators		Elevators consume about 5% of a building's overall electricity; Escalators generally account for anywhere from 2-10% of a building's energy output;
Fire alarm systems		
Energy		
Alternative power sources, type and number		
Kwh generated by alternate power sources		
Solar PV		
RECs		
Mechanical		
HVAC		HVAC is a big part of mechanical engineering for building systems and affects energy usage the most.
Central boiler plant		
Rooftop air handling units		
Exhaust fans		
Fresh air intake		
Absorption chillers		
Cooling towers		
Air quality testing		

Plumbing		
Rooftop water towers		
Water sampling		
		sinks and toilets. plumbing engineers design domestic hot water heating systems, water softening systems,
Water circulating pumps		
Domestic hot water		
Storm piping for roof drainage		
Sprinkler systems		
Water softening systems		
Sinks and toilets		
Alternative water supply		
Drinking water quality		
Structural		
Steel or concrete frame		Roofing, ceiling height, elevators & escalators acoustics
Foundation type		
Footing depth		
Roof support (truss)		
Roof type		
Roof depth		

Roof insulation type and r-factor		
Walls exterior cladding		
Vapor barrier type and characteristics		
Wall assembly type		
Wall insulation r-factor & u-factors		
Fenestration % of surface		
Window solar heat gain coefficient, e-coatings, u- and r-factors		
Window number, type and models		
Insulation		
Acoustics		
Technology		
		This particular system could be considered a subset of electrical, but it technology engineers have specialized skills. Technology covers a broad range of products such as security and card access, cameras, IT infrastructure such as wireless internet access, telephone systems, audio visual (AV) equipment, and centralized clock systems. They typically collaborate with interior designers and electrical engineers during their design. BIS systems. Elevator control system

Appendix II Beginner's Guide to Blockchain

The following article published in Forbes Magazine explains the basics of blockchain technology and applications (Marr, 2017).

"You may have heard the term 'blockchain' and dismissed it as a fad, a buzzword, or even technical jargon. But I believe blockchain is a technological advance that will have wide-reaching implications that will not just transform the financial services but many other businesses and industries.

A blockchain is a distributed database, meaning that the storage devices for the database are not all connected to a common processor. It maintains a growing list of ordered records, called blocks. Each block has a timestamp and a link to a previous block.

Users can only edit the parts of the blockchain that they "own" by possessing the private keys necessary to write to the file. Cryptography ensures that everyone's copy of the distributed blockchain is kept in synch.

Blockchains are secure databases by design. The concept was introduced in 2008 by Satoshi Nakamoto, and then implemented for the first time in 2009 as part of the digital bitcoin currency; the blockchain serves as the public ledger for all bitcoin transactions. By using a blockchain system, bitcoin was the first digital currency to solve the double spending problem (unlike physical coins or tokens, electronic files can be duplicated and spent twice) without the use of an authoritative body or central server.

The security is built into a blockchain system through the distributed timestamping server and peer-to-peer network, and the result is a database that is managed autonomously in a decentralized way. This makes blockchains excellent for recording events — like medical records — transactions, identity management, and proving provenance. It is, essentially, offering the potential of mass disintermediation of trade and transaction processing.

How does blockchain really work?

Some people have called blockchain the “internet of value” which I think is a good metaphor.

On the internet, anyone can publish information and then others can access it anywhere in the world. A blockchain allows anyone to send value anywhere in the world where the blockchain file can be accessed. But you must have a private, cryptographically created key to edit only the blocks you “own.”

Using your private key and someone else's public key you can transfer the value of whatever is stored in that section of the blockchain.

So, to use the bitcoin example, keys are used to transfer blocks, which contain units of currency that have financial value. This fills the role of recording the transfer, which is traditionally carried out by banks.

It also fills a second role, establishing trust and identity, because no one can edit a blockchain without having the corresponding keys. Edits not verified by those keys are rejected by the network. Of course, the keys — like a physical currency — could theoretically be stolen, but a few lines of computer code can generally be kept secure at very little expense. (Unlike, say, the expense of storing a cache of gold in a proverbial Fort Knox.)

This means that the major functions carried out by banks — verifying identities to prevent fraud and then recording legitimate transactions — can be carried out by a blockchain more quickly and accurately.

Why is blockchain important?

We are all now used to sharing information through a decentralized online platform: the internet. But when it comes to transferring value – e.g. money, ownership rights, intellectual property, etc. – we are usually forced to fall back on old-fashioned, centralized institutions or establishments like banks or government agencies. Even online payment methods which have sprung into existence since the birth of the internet – PayPal being the most obvious example – generally require integration with a bank account or credit card to be useful.

Blockchain technology offers the intriguing possibility of eliminating this “middleman”. It does this by filling three important roles – recording transactions, establishing identity and establishing contracts – traditionally carried out by the financial services sector.

This has huge implications because, worldwide, the financial services market is the largest sector of industry by market capitalization. Replacing even a fraction of this with a blockchain system would result in a huge disruption of the financial services industry, but also a massive increase in efficiencies.

The third role, establishing contracts, opens up a treasure trove of opportunities. Apart from a unit of value (like a bitcoin), blockchain can be used to store any kind of digital information, including computer code.

That snippet of code could be programmed to execute whenever certain parties enter their keys, thereby agreeing to a contract. The same code could read from external data feeds — stock prices, weather reports, news headlines, or anything that can be parsed by a computer, really — to create contracts that are *automatically* filed when certain conditions are met.

These are known as "smart contracts," and the possibilities for their use are practically endless.

For example, your smart thermostat might communicate energy usage to a smart grid; when a certain number of wattage hours has been reached, another blockchain automatically transfers value from your account to the electric company, effectively automating the meter reader and the billing process.

Or, smart contracts might be put to use in the regulation of intellectual property, controlling how many times a user can access, share, or copy something. It could be used to create fraud-proof voting systems, censorship-resistant information distribution, and much more.

The point is that the potential uses for this technology are vast, and I predict that more and more industries will find ways to put it to good use in the very near future."

Appendix III What is Ethereum?

Following is an article from the official website by the original developers of Ethereum (Ethereum.org, 2019). The article is intended to provide beginners with a basic explanation of the technology.

“Ethereum is the foundation for a new era of the internet:

- An internet where money and payments are built in.
- An internet where users can own their data, and your apps don’t spy and steal from you.
- An internet where everyone has access to an open financial system.
- An internet built on neutral, open-access infrastructure, controlled by no company or person.

Launched in 2015, Ethereum is the world’s leading programmable blockchain.

Like other blockchains, Ethereum has a native cryptocurrency called Ether (ETH). ETH is digital money. If you’ve heard of Bitcoin, ETH has many of the same features. It is purely digital, and can be sent to anyone anywhere in the world instantly. The supply of ETH isn’t controlled by any government or company - it is decentralized, and it is scarce. People all over the world use ETH to make payments, as a store of value, or as collateral.

But unlike other blockchains, Ethereum can do much more. Ethereum is programmable, which means that developers can use it to build new kinds of applications.

These decentralized applications (or “dapps”) gain the benefits of cryptocurrency and blockchain technology. They can be trustworthy, meaning that once they are “uploaded” to Ethereum, they will always run as programmed. They can control digital assets in order to create new kinds of financial applications. They can be decentralized, meaning that no single entity or person controls them.

Right now, thousands of developers all over the world are building applications on Ethereum, and inventing new kinds of applications, many of which you can use today:

- Cryptocurrency wallets that let you make cheap, instant payments with ETH or other assets
- Financial applications that let you borrow, lend, or invest your digital assets
- Decentralized markets, that let you trade digital assets, or even trade “predictions” about events in the real world
- Games where you own in-game assets, and can even make real money
- And much, much more.

The Ethereum community is the largest and most active blockchain community in the world. It includes core protocol developers, crypto-economic researchers, cypherpunks, mining organizations, ETH holders, app developers, ordinary users, anarchists, fortune 500 companies, and, as of now, you.

There is no company or centralized organization that controls Ethereum. Ethereum is maintained and improved over time by a diverse global community of contributors who work on everything from the core protocol to consumer applications. This website, just like the rest of Ethereum, was built - and continues to be built - by a collection of people working together. Welcome to Ethereum.”

Appendix IV How to Build Blockchain App and Ethereum To Do List

The student author of this thesis has expertise in environmental architecture and real estate, but only a working knowledge of Ethereum. Therefore, it was necessary to retain Ethereum experts for the limited purpose of creating the necessary smart contracts, deploy them to the blockchain and load data for the subject building and site.

The expert contractors' assignment was not simply to create a green building rating system on the blockchain, but to give explicit, transparent instructions to the thesis author so that she could execute the steps herself if necessary.

The following text is an unedited version of the experts' detailed report on how the pilot project was designed and executed. It includes casual language and the use of friendly nicknames. Because the building rating system in this thesis pilot project was not named at the outset, expert blockchain developers used a working title of "Kris Green Rating System." The text, reprinted verbatim, follows. Please note that this text deviates from Harvard's double-space standard due to the spacing requirements of computer code.

Verbatim Programming Instructions

"I'm going to show you how to build your blockchain application. Let's create a todo list powered by Ethereum smart contracts. First, we'll create a smart contract with the Solidity programming language. Then, we'll write tests against the smart contract, and deploy it to a blockchain. Finally, we'll create a client side application to for the todo list.

I've chosen a todo list application for this project because it is one of the most common ways to learn any new programming language. It will teach us how to read and write data from the blockchain, as well as execute business logic that will govern the behavior of our todo list application. It shows the fundamentals about how a blockchain works and how to write Ethereum smart contracts.

In order to understand how a blockchain application works, let's first look at how a todo list might work as a web application. To access the todo list, you would use a web browser that would communicate with a web server over the Internet. The server contains all of the code and data for the todo list.

Web Application Diagram

Here is a list of what you would find on the server:

Client side files in HTML, CSS, and JavaScript

Back end code responsible for the application's business logic for the Kris Rating System

Database that stores the tasks in the todo list

This server is a centralized entity that has full control over every aspect of the application. Anyone with full access to the server can change any part of the code or the data at any time. A blockchain application works quite differently. All of the code and the data to the todo list does not lie on a centralized server. Instead, it is distributed across the blockchain. All of the code and the data is shared and unchangeable on the blockchain. This can be used to store rating information on a particular building.

To illustrate this, let's examine how our blockchain-based todo list will work.

Blockchain Application Diagram

To access the blockchain todo list, we'll use a web browser to talk to the client side application, which will be written in HTML, CSS, and JavaScript. Instead of talking to a back end web server, the client side application will talk directly to the blockchain.

What is a Blockchain?

A blockchain is a peer-to-peer network of computers, or nodes, that talk to one another. It's a distributed network where all of the participants share the responsibility of running the network. Each network participant maintains a copy of the code and the data on the blockchain. All of this data is contained in bundles of records called "blocks" which are "chained together" to make up the blockchain. All of the nodes on the network ensure that this data is secure and unchangeable, unlike a centralized application where the code and data can be changed at any time. That's what makes the blockchain so powerful! Because the blockchain is responsible for storing data, it fundamentally is a database. And because it's a network of computers that talk to one another, it's a network. You can think of it as a network and a database all in one.

I should also highlight another fundamental distinction between traditional web applications and blockchain applications: instead of being a user of the application itself, you are a user of the blockchain network. The application does not manage any user data. That is the responsibility of the blockchain!

What is a Smart Contract?

All of the code on the blockchain is contained in smart contracts, which are programs that run on the blockchain. They are the building blocks of blockchain applications. We'll write a smart contract in this tutorial to power our todo list. It will be responsible for fetching all of the tasks in our todo list from the blockchain, adding new tasks, and completing tasks.

Smart contracts are written in a programming language called Solidity, which looks a lot like JavaScript. All of the code in the smart contract is immutable, or unchangeable. Once we deploy the smart contract to the blockchain, we won't be able to change or update any of the code. This is a design feature that ensures that the code is trustless and secure. I often compare smart contracts to microservices on the web. They act as an interface for reading and writing data from the blockchain, as well as executing business logic. They're publicly accessible, meaning anyone with access to the blockchain can access their interface.

How Blockchain Todo List Works

Let's recap to understand how the application will work that we'll build in this tutorial. We'll create a client side application for the todo list that will talk directly to the blockchain. We'll use the Ethereum blockchain in this tutorial, which we can access by connecting our client side application to a single Ethereum node. We'll write a smart contract in Solidity that powers the todo list, and we'll deploy it to the Ethereum blockchain. We'll also connect to the blockchain network with our personal account using an Ethereum wallet in order to interact with the todo list application.

Application Preview

Here is a preview of the todo list application that we'll build in this tutorial. We'll be able to list out all of the tasks in the todo list, create new ones, and complete them.

Ethereum Todo List Application

Installing Dependencies

Now let's install all of the dependencies we need to build our project. First, we'll set up a personal blockchain to develop the application locally.

Ganache Personal Blockchain

The dependency is a personal blockchain, which is a local development blockchain that can be used to mimic the behavior of a public blockchain. I recommend using Ganache as your personal blockchain for Ethereum development. It will allow you to deploy smart contracts, develop applications, and run tests. It is available on Windows, Mac, and Linux as a desktop application and a command line tool!

Ganache Personal Blockchain Application

I'll walk you through setting up the desktop application in this tutorial. You can find the latest release for your operating system here. Once you've downloaded the archived package, extract the installer and run through the setup steps.

Now you have a personal blockchain network running! You can see some details about the server Ganache is running on, along with a list of accounts connected to the network. Each account has been credited with 100 ether. This is a huge time saver! If you were to you create your own personal blockchain network from scratch, or develop your application on a test network, you would have to create all 10 accounts manually and credit each account with ether. Thankfully Ganache has already done this for us so that we don't have to worry about it.

Node.JS

Now that you have a private blockchain running, you need to configure your environment for developing smart contracts. The first dependency you'll need is Node Package Manager, or NPM, which comes with Node.js. You can see if you have node already installed by going to your terminal and typing:

```
$ node -v
```

If you don't have node already installed you can visit the Node.js website to download it.

Truffle Framework

Now let's install the Truffle Framework, which provides a suite of tools for developing Ethereum smart contacts with the Solidity programming language.

Truffle Blockchain Smart Contract Development Framework

Here is an overview of all the functionality we'll get with the Truffle Framework:

Smart Contract Management - write smart contracts with the Solidity programming language and compile them down to bytecode that will be run on the Ethereum Virtual Machine (EVM).

Automated Testing - write tests against your smart contracts to ensure that they behave the way you want them to. These tests can be written in JavaScript or Solidity, and can be run against any network configured by Truffle, including public blockchain networks.

Deployment & Migrations - write scripts to migrate and deploy smart contracts to any public Ethereum blockchain network.

Network Management - connect to any public Ethereum blockchain network, as well as any personal blockchain network you might use for development purposes.

Development Console - interact with smart contracts inside a JavaScript runtime environment with the Truffle Console. You can connect to any blockchain network that you've specified within your network configuration to do this.

Script Runner - write custom scripts that can run against a public blockchain network with JavaScript. You can write any arbitrary code inside this file and run it within your project.

Client Side Development - configure your truffle project to host client side applications that talk to your smart contracts deployed to the blockchain.

You can install Truffle with NPM in your command line like this. NOTE: It's important to use this exact version of truffle specified below in order to follow along with this tutorial.

```
$ npm install -g truffle@5.0.2
```

Metamask Ethereum Wallet

Now it's time to turn your web browser into a blockchain browser. Most major web browsers do not currently connect to blockchain networks, so we'll have to install a browser extension that allows them to do this.

Metamask Ethereum Wallet Browser Extension

I'll use the Metamask extension for Google Chrome. To install Metamask, visit this link or search for the Metamask Chrome plugin in the Google Chrome web store. Once you've installed it, be sure that it is checked in your list of extensions. You'll see the fox icon in the top right hand side of your Chrome browser when it's installed. Reference the video walk through if you get stuck!

Metamask will also allow us to manage our personal account when we connect to the blockchain, as well as manage our Ether funds that we'll need to pay for transactions.

Project Setup

Now let's create the project! I'll first create a project directory, and enter into it like this:

```
$ mkdir eth-todo-list
```

```
$ cd eth-todo-list
```

Now we initialize a new truffle project to develop our project like this:

```
$ truffle init
```

Great! Your terminal output should show that the project was created successfully. You can open your text editor and see that some new files and directories were created once you ran that command. Now let's create a package.json file to install some development dependencies that will need for the project. You can do that from the command line like this:

```
$ touch package.json
```

You can bootstrap all of the dependencies for your project by simply copy-and-pasting the code below into your package.json file:

```
{

 "name": "eth-todo-list",

 "version": "1.0.0",

 "description": "Blockchain Todo List Powered By Ethereum",

 "main": "truffle-config.js",

 "directories": {

  "test": "test"

 },

 "scripts": {

  "dev": "lite-server",

  "test": "echo \"Error: no test specified\" && sexit 1"

 },

 "author": "kris@kato.com",

 "license": "ISC",

 "devDependencies": {

  "bootstrap": "4.1.3",

  "chai": "^4.1.2",

  "chai-as-promised": "^7.1.1",
```

```
  "chai-bignumber": "^2.0.2",

  "lite-server": "^2.3.0",

  "nodemon": "^1.17.3",

  "truffle": "5.0.2",

  "truffle-contract": "3.0.6"

 }

}
```

Now you can install the dependencies from the command line like this:

$ npm install

Now that the dependencies are installed, let's examine the project directory structure that we just created:

Truffle Project Directory Structure

contracts directory: this is where all smart contacts live. We already have a Migration contract that handles our migrations to the blockchain.

migrations directory: this is where all of the migration files live. These migrations are similar to other web development frameworks that require migrations to change the state of a database. Whenever we deploy smart contracts to the blockchain, we are updating the blockchain's state, and therefore need a migration.

node_modules directory: this is the home of all of our Node dependencies we just installed.

test directory: this is where we'll write our tests for our smart contract.

truffle-config.js file: this is the main configuration file for our Truffle project, where we'll handle things like network configuration.

Now let's start developing the smart contract that will manage our todo list. We can do this by creating a new file in the contracts directory like this:

$ touch ./contracts/TodoList.sol

Inside here, let's develop our todo list smart contract. First, we'll start by specifying the version like this:

```
pragma solidity ^0.5.0;
```

Now we can declare the smart contract like this:

```
pragma solidity ^0.5.0;

contract TodoList {

  // Code goes here...

}
```

We create a smart contract called TodoList followed by curly braces. We'll add all of the code for the smart contract inside of them. The thing we'll do is just keep track of the number of tasks inside the todo list. This will allow us to write some simple code that will help us ensure that the project is set up properly, and that our code is working on the blockchain. We'll simply create a state variable called taskCount to track the number of tasks like this:

```
pragma solidity ^0.5.0;

contract TodoList {

 uint taskCount;

}
```

Here taskCount is a special kind of variable called a "state variable". Any data that we store inside this state variable is written to storage on the blockchain. It changes the smart contract's state, and has scope within the entire smart contract, as opposed to local variables which only have scope inside of functions. We can set a default value of 0 for this state variable like this:

```
pragma solidity ^0.5.0;

contract TodoList {

 uint taskCount = 0;

}
```

Now, we can create a way to access the value of this state variable outside of the contract. We can do this with a special modifier keyword called public in Solidity. When we do this, Solidity will magically create a taskCount() function so that we can access this variable's value outside of the smart contract. This will be useful when we are interacting with the smart contract in the console, from the client side application, and inside the test files.

Now let's compile the smart contract and ensure that there are no errors:

$ truffle compile

Yay! 🎉 You've just written your first Ethereum smart contract. You should notice that a new file was generated whenever you compiled the smart contract at the following path: `./build/contracts/TodoList.json`. This file is the smart contract ABI file, which stands for "Abstract Binary Interface". This file has many responsibilities, but two that I will highlight here:

It contains the compiled bytecode version of the Solidity smart contract code that can be run on a the Ethereum Virtual Machine (EVM), i.e., an Ethereum Node.

It contains a JSON representation of the smart contract functions that can be exposed to external clients, like client-side JavaScript applications.

Our next goal is to access the smart contract inside the Truffle console. However, we cannot run the Truffle console because our application is not yet connected to the Ganache personal blockchain network we set up in the dependencies section. To talk to the smart contract on the personal blockchain network inside the Truffle console, we must do a few things:

Update our project's configuration file to specify the personal blockchain network we want to connect to (Ganache).

Create a migration script that tells Truffle how to deploy the smart contract to the personal blockchain network.

Run the newly created migration script, deploying the smart contract to the personal blockchain network.

First, we'll update the project configuration file to specify the personal blockchain network we want set up in the first section. Find the file truffle-config.js and paste the following code:

```
module.exports = {

 networks: {

   development: {

     host: "127.0.0.1",

     port: 7545,
```

```
    network_id: "*" // Match any network id
  }
 },
 solc: {
  optimizer: {
    enabled: true,
    runs: 200
  }
 }
}
```

Note: these should match the default settings provided by the Ganache personal blockchain network. If you changed any settings inside the Ganache settings page, like the port, those should be reflected here.

Next, we'll create a migration script inside the migrations directory to deploy the smart contract to the personal blockchain network. From your project root, create a new file from the command line like this:

```
$ touch migrations/2_deploy_contracts.js
```

Let me explain what this file does. Any time we create a new smart contract, we are updating the state of the blockchain. Remember, I said that a blockchain fundamentally is a database. Hence, whenever we permanently change it, we must migrate it from one state to another. This is very similar to a database migration that you might have performed in other web application development frameworks.

Notice that we number all of our files inside the migrations directory with numbers so that Truffle knows which order to execute them in. Inside this newly created migration file, you can use this code to deploy the smart contract:

```
var TodoList = artifacts.require("./TodoList.sol");

module.exports = function(deployer) {

 deployer.deploy(TodoList);
```

};

First, we require the contract we've created, and assign it to a variable called "TodoList". Next, we add it to the manifest of deployed contracts to ensure that it gets deployed when we run the migrations. Now let's run this migration script from the command line like this:

$ truffle migrate

Now that we have successfully migrated the smart contract to the personal blockchain network, let's open the console to interact with the smart contract. You can open the truffle console from the command line like this:

$ truffle console

Now that we're inside the console, let's get an instance of our deployed smart contract and see if we can read the taskCount from the contract. From the console, run this code:

todoList = await TodoList.deployed()

Here TodoList is the name of the variable that we created in the migration file. We retrieved a deployed instance of the contract with the deployed() function, and assigned it to an todoList. Also, note the use of the await keyword. We must interact with the blockchain in an asynchronous fashion. Thus, JavaScript is an excellent choice for client-side interactions with blockchain smart contracts. There are several strategies for handling asynchronous actions in JavaScript. One of the most popular ways is the async/await pattern which I have chosen here. Truffle has recently released support for this inside the Truffle console. You can read more about the async/await pattern here.

First, we can get the address of the smart contract that was deployed to the blockchain like this:

todoList.address

// => '0xABC123...'

Now we can read the value of taskCount from the storage like this:

taskCount = await app.taskCount()

// => 0

You have successfully completed the first section. You have done all of the following:

Set up your machine for blockchain development

Created a new truffle project

Created your first smart contract

Interacted with your newly created smart contract on a live blockchain

If you got stuck on any of the steps, feel free to clone the project code for this section from github.

List Tasks

Now let's start listing out the tasks in the todo list. Here are all of the steps that we'll complete in this section:

Write code to list tasks in the smart contract

List tasks from the smart contract inside the Truffle console

List tasks in the client side application

Write a test for listing tasks

In order to list the tasks inside the smart contract, we'll need a way to model a task in solidity. Solidity allows you to define your own data types with structs. We can model any arbitrary data with this powerful feature. We'll use a struct to model the task for our todo list like this:

```
pragma solidity ^0.5.0;

contract TodoList {

 uint public taskCount = 0;

 struct Task {

   uint id;

   string content;

   bool completed;

 }

}
```

First we model the task with the struct keyword followed by the name of the new struct

Task. Note, that this does not represent an instance of a Task, but simply the definition of a Task struct. The lines contained in the curly braces define the attributes of the Task struct:

uint id - this is the unique identifier for the struct. It will have an id, just like a traditional database record. Note, we declare the data type for this identifiers as a uint, which stands for "unsigned integer". This simply means that it is a non-negative integer. It has no "sign", i.e. a - or + sign, in front of it, implying that it is always positive.

string content - this is the text of the task in the todo list contained in a string.

bool completed - this is the checkbox status of the todo list, which is true/false. If it is true, the task will be "completed" or checked off from the todo list.

Now that we've modeled a task, we need a place to put all of the tasks in the todo list! We want to put them in storage on the blockchain so that the state of the smart contract will be persistent. We can access the blockchain's storage with a state variable, just like we did with taskCount. We'll create a tasks state variable. It will use a special kind of Solidity data structure called a mapping like this:

```
pragma solidity ^0.5.0;

contract TodoList {

 uint public taskCount = 0;

 struct Task {

   uint id;

   string content;

   bool completed;

 }

 mapping(uint => Task) public tasks;

}
```

A mapping in Solidity is a lot like an associative array or a hash in other programming languages. It creates key-value pairs that get stored on the blockchain. We'll use a unique id as the key. The value will be the task itself. This will allow us to look up any task by id!

Now let's create a function for creating tasks. This will allow us to add new tasks to the todo list by default so that we can list them out in the console.

```
pragma solidity ^0.5.0;

contract TodoList {

 uint public taskCount = 0;

 struct Task {

   uint id;

   string content;

   bool completed;

 }

 mapping(uint => Task) public tasks;

 function createTask(string memory _content) public {

   taskCount ++;

   tasks[taskCount] = Task(taskCount, _content, false);

 }

}
```

I'll explain this function. First, we create the function with the function keyword, and give it a name createTask()

We allow the function to accept one argument called _content, which will be the text for the task. We specify that this argument will be of string data type, and that it will persist in memory

We set the function visibility to public so that it can be called outside of the smart contract, like in the console, or from the client side for example

Inside the function, we create an id for the new task. We simply take the existing taskCount and increment it by 1.

Now we create a new task struct by calling Task(taskCount, _content, false); and passing in the values for the new task.

Next, we store the new task on the blockchain by adding it to the tasks mapping like this: task[taskCount] = ...

Now we want to add one task to the todo list whenever the smart contract is deployed to the blockchain so that it will have a default task that we can inspect in the console. We can do this by calling the createTask() function inside of the smart contract's constructor function like this:

```
contract TodoList {

  // ....

  constructor() public {

    createTask("Check out GreenBlocksRating.com");

  }

  // ....

}
```

We create the constructor function with the constructor keyword as you can see above. This function will get run only once, whenever the contract is initialized, i.e., deployed to the blockchain. Inside of this function, we have created one new default task with the string content "Check out GreenBlocksRating.com".

Now let's deploy this smart contract to the blockchain. In order to do this, we must deploy a new copy of our code. Remember, smart contract code is immutable! It cannot change. Therefore, we must create a new smart contract any time we make code changes. Luckily Truffle provides a shortcut to assist with this. We can re-run the migrations like this:

```
$ truffle migrate --reset
```

Viola! Now we have a new copy of the smart contract on the blockchain. Now let's list out the tasks in the console.

```
$ truffle console
```

Inside the console, let's get a deployed copy of the new smart contract.

```
todoList = await TodoList.deployed()
```

Now we can get the task from the todo list by calling the tasks() function. This will allow us to access values from the tasks mapping by id. We will simply pass in the id of the first task in the list when we call this function:

task = await todoList.tasks(1)

Yay! 🎉 How you can inspect the values of this task in the console. 😃

Now that we've migrated this smart contract to the blockchain, let's create the client side code to interact with the todo list smart contract. You'll need to create the following files for your project:

bs-config.json

src/index.html

src/app.js

We'll fill the code for all of these files one-by-one. We are using lite-server to serve all of the project files for the client side. We'll need to tell lite-server where all these files are located. We can do this by updating the browsersync configuration for lite-server inside the bs-config.json file. Paste this configuration into your project file:

```
{

 "server": {

  "baseDir": [

   "./src",

   "./build/contracts"

  ],

  "routes": {

   "/vendor": "./node_modules"

  }

 }

}
```

This configuration tells lite-server to expose all the files in the src and build/contracts directories to the root of our web server. It also adds an alias for any files in the node_modules directory to appear in the vendor route. This will allow us to pull in any project dependencies like bootstrap into the client side with the vendor route, which we'll see momentarily.

Now let's fill in HTML code for our todo list. This tutorial focuses primarily on blockchain technology, so I don't want to spend too much time on the HTML & CSS portion. I'll simply paste in the HTML code here:

```
<!DOCTYPE html>

<html lang="en">

 <head>

  <meta charset="utf-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags -->

  <title>Kris Green Rating System for Ethereum | Todo List</title>

  <!-- Bootstrap -->

  <link href="vendor/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">

  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->

  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

  <!--[if lt IE 9]>

   <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

   <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

  <![endif]-->

  <style>

   main {

    margin-top: 60px;

   }
```

```
    #content {
      display: none;
    }
    form {
      width: 350px;
      margin-bottom: 10px;
    }
    ul {
      margin-bottom: 0px;
    }
    #completedTaskList .content {
      color: grey;
      text-decoration: line-through;
    }
  </style>
</head>
<body>
  <nav class="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0 shadow">
    <a class="navbar-brand col-sm-3 col-md-2 mr-0" href="http://www.kato.com/free-
download123" target="_blank">Kris | Todo List</a>
    <ul class="navbar-nav px-3">
      <li class="nav-item text-nowrap d-none d-sm-none d-sm-block">
        <small><a class="nav-link" href="#"><span id="account"></span></a></small>
```

```
      </li>
    </ul>
  </nav>
  <div class="container-fluid">
    <div class="row">
      <main role="main" class="col-lg-12 d-flex justify-content-center">
        <div id="loader" class="text-center">
          <p class="text-center">Loading...</p>
        </div>
        <div id="content">
      <!-- <form onSubmit="App.createTask(); return false;">
            <input id="newTask" type="text" class="form-control" placeholder="Add
task..." required>
            <input type="submit" hidden="">
          </form> -->
          <ul id="taskList" class="list-unstyled">
            <div class="taskTemplate" class="checkbox" style="display: none">
              <label>
                <input type="checkbox" />
                <span class="content">Task content goes here...</span>
              </label>
            </div>
          </ul>
          <ul id="completedTaskList" class="list-unstyled">
```

```
          </ul>
        </div>
      </main>
    </div>
  </div>
  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
  <!-- Include all compiled plugins (below), or include individual files as needed -->
  <script src="vendor/bootstrap/dist/js/bootstrap.min.js"></script>
  <script src="vendor/truffle-contract/dist/truffle-contract.js"></script>
  <script src="app.js"></script>
 </body>
</html>
```

This file scaffolds all of the HTML we need for the project. I have commented-out the form code which we'll enable in a later section. The file pulls in all of the dependencies for the project like the bootstrap templating framework that will allow us to create nice-looking UI elements without having to write too much CSS. It also uses the Truffle Contract library that will allow us to interact with the todo list smart contract with JavaScript.

Now let's fill in the JavaScript code for this section. We'll add code to the newly created app.js file like this:

```
App = {
 loading: false,
 contracts: {},
 load: async () => {
  await App.loadWeb3()
```

```
  await App.loadAccount()

  await App.loadContract()

  await App.render()

 },

 // https://medium.com/metamask/https-medium-com-metamask-breaking-change-injecting-web3-7722797916a8

 loadWeb3: async () => {

  if (typeof web3 !== 'undefined') {

   App.web3Provider = web3.currentProvider

   web3 = new Web3(web3.currentProvider)

  } else {

   window.alert("Please connect to Metamask.")

  }

  // Modern dapp browsers...

  if (window.ethereum) {

   window.web3 = new Web3(ethereum)

   try {

    // Request account access if needed

    await ethereum.enable()

    // Acccounts now exposed

    web3.eth.sendTransaction({/* ... */})

   } catch (error) {

    // User denied account access...
```

```
    }
  }
  // Legacy dapp browsers...
  else if (window.web3) {
    App.web3Provider = web3.currentProvider
    window.web3 = new Web3(web3.currentProvider)
    // Acccounts always exposed
    web3.eth.sendTransaction({/* ... */})
  }
  // Non-dapp browsers...
  else {
    console.log('Non-Ethereum browser detected. You should consider trying MetaMask!')
  }
},
loadAccount: async () => {
  // Set the current blockchain account
  App.account = web3.eth.accounts[0]
},
loadContract: async () => {
  // Create a JavaScript version of the smart contract
  const todoList = await $.getJSON('TodoList.json')
  App.contracts.TodoList = TruffleContract(todoList)
  App.contracts.TodoList.setProvider(App.web3Provider)
```

```
  // Hydrate the smart contract with values from the blockchain
  App.todoList = await App.contracts.TodoList.deployed()
},
render: async () => {
  // Prevent double render
  if (App.loading) {
    return
  }
  // Update app loading state
  App.setLoading(true)
  // Render Account
  $('#account').html(App.account)
  // Render Tasks
  await App.renderTasks()
  // Update loading state
  App.setLoading(false)
},
renderTasks: async () => {
  // Load the total task count from the blockchain
  const taskCount = await App.todoList.taskCount()
  const $taskTemplate = $('.taskTemplate')
  // Render out each task with a new task template
  for (var i = 1; i <= taskCount; i++) {
```

```
    // Fetch the task data from the blockchain
    const task = await App.todoList.tasks(i)
    const taskId = task[0].toNumber()
    const taskContent = task[1]
    const taskCompleted = task[2]
    // Create the html for the task
    const $newTaskTemplate = $taskTemplate.clone()
    $newTaskTemplate.find('.content').html(taskContent)
    $newTaskTemplate.find('input')
                    .prop('name', taskId)
                    .prop('checked', taskCompleted)
                    // .on('click', App.toggleCompleted)
    // Put the task in the correct list
    if (taskCompleted) {
      $('#completedTaskList').append($newTaskTemplate)
    } else {
      $('#taskList').append($newTaskTemplate)
    }
    // Show the task
    $newTaskTemplate.show()
  }
},
setLoading: (boolean) => {
```

```
  App.loading = boolean
  const loader = $('#loader')
  const content = $('#content')
  if (boolean) {
    loader.show()
    content.hide()
  } else {
    loader.hide()
    content.show()
  }
 }
}
$(() => {
 $(window).load(() => {
  App.load()
 })
})
```

Let me explain this code. We have created a new App object that contains all the functions we need to run the JavaScript app. I will explain the important functions here.

loadWeb3() web3.js is a JavaScript library that allows our client-side application to talk to the blockchain. We configure web3 here. This is default web3 configuration specified by Metamask. Do not worry if you don't completely understand what is happening here. This is a copy-and-paste implementation that Metamask suggests.

loadContract() This is where we load the smart contract data from the blockchain. We create a JavaScript representation of the smart contract with the Truffle Contract library. Then we load the smart contract data with web3. This will allow us to list the tasks in the todo list.

renderTasks() This is where we actually list the tasks in the todo list. Notice that we create a for loop to access each task individually. That is because we cannot fetch the entire tasks mapping from the smart contract. We must first determine the taskCount and fetch each task one-by-one.

Now let's start the web server and ensure that the project will load in the browser.

$ npm run dev

You have successfully loaded the client side application. Notice that your application says "Loading...". That's because we're not logged in to the blockchain yet! In order to connect to the blockchain, we need to import one of the accounts from Ganache into Metamask. You can watch me set up Metamask in the video at 43:55.

Ethereum Todo List Loading

Once you're connected with Metamask, you should see all of the contract and account data loaded.

Ethereum Todo List Loaded

Boom! There is your todo list!

Testing

Now let's write a basic test to ensure that the todo list smart contract works properly. First, let me explain why testing is so important when you're developing smart contracts. We want to ensure that the contracts are bug-free for a few reasons:

All of the code on the Ethereum blockchain is immutable; it cannot change. If the contract contains any bugs, we must disable it and deploy a new copy. This new copy will not have the same state as the old contract, and it will have a different address.

Deploying contracts costs gas because it creates a transaction and writes data to the blockchain. This costs Ether, and we want to minimize the amount of Ether we ever have to pay.

If any of our contract functions that write to the blockchain contain bugs, the account who is calling this function could potentially waste Ether, and it might not behave the way they expect.

Let's create a test file like this:

$ test/TodoList.test.js

We'll write all our tests in JavaScript inside this file with the Mocha testing framework and the Chai assertion library. These come bundled with the Truffle framework. We'll write all these tests in JavaScript to simulate client-side interaction with our smart contract, much like we did in the console. Here is all the code for the tests:

```
const TodoList = artifacts.require('./TodoList.sol')

contract('TodoList', (accounts) => {

 before(async () => {

  this.todoList = await TodoList.deployed()

 })

 it('deploys successfully', async () => {

  const address = await this.todoList.address

  assert.notEqual(address, 0x0)

  assert.notEqual(address, '')

  assert.notEqual(address, null)

  assert.notEqual(address, undefined)

 })

 it('lists tasks', async () => {

  const taskCount = await this.todoList.taskCount()

  const task = await this.todoList.tasks(taskCount)

  assert.equal(task.id.toNumber(), taskCount.toNumber())

  assert.equal(task.content, 'Check out GreenBlocksRating.com')

  assert.equal(task.completed, false)

  assert.equal(taskCount.toNumber(), 1)

 })

})
```

Let me explain this code. First, we require the contract and assign it to a variable, like we did in the migration file. Next, we call the "contract" function, and write all our tests within the callback function. This callback function provides an "accounts" variable that represents all the accounts on our blockchain, provided by Ganache.

The first test checks that the contract was deployed to the blockchain properly by inspecting its address.

The next test checks that the smart contract lists task properly by checking the default task that we created in the initializer function.

Now let's run the tests from the command line like this:

```
$ truffle test
```

Create Tasks

We've already created a function for creating tasks, but it is not complete just yet. That's because I want to trigger an event any time that new task is created. Solidity allows us to trigger arbitrary events which external consumers can subscribe to. It will allow us to listen for these events inside client side applications, etc... Let's create a TaskCreated() event and trigger it anytime a new task is created in the createTask() function like this:

```
pragma solidity ^0.5.0;

contract TodoList {

 // ...

 event TaskCreated(

   uint id,

   string content,

   bool completed

 );

 // ...

 function createTask(string memory _content) public {

   taskCount ++;

   tasks[taskCount] = Task(taskCount, _content, false);
```

```
  emit TaskCreated(taskCount, _content, false);

 }

}
```

Now let's create a test to ensure that this event is triggered any time a new task is created. We will inspect the transaction receipt when the new task is created. This will contain all of the log information that will contain the event data. We can inspect this data inside our test like this to ensure that the event was triggered properly:

```
it('creates tasks', async () => {

 const result = await this.todoList.createTask('A new task')

 const taskCount = await this.todoList.taskCount()

 assert.equal(taskCount, 2)

 const event = result.logs[0].args

 assert.equal(event.id.toNumber(), 2)

 assert.equal(event.content, 'A new task')

 assert.equal(event.completed, false)

})
```

Now let's run the test:

```
$ truffle test
```

Yes, they pass! Now let's deploy a new copy of the smart contract to the blockchain since the code has changed:

```
$ truffle migrate --reset
```

Now let's update the client side code. We'll un-comment the form code in the index.html file first:

```
<form onSubmit="App.createTask(); return false;">

 <input id="newTask" type="text" class="form-control" placeholder="Add task..." required>
```

```
 <input type="submit" hidden="">

</form>
```

Now we'll add a createTask() function in the app.js file like this:

```
createTask: async () => {

 App.setLoading(true)

 const content = $('#newTask').val()

 await App.todoList.createTask(content)

 window.location.reload()

},
```

Now you should be able to add new tasks from the client side application! Notice, there is no "submit" button on the form. I left it off to simplify the user interface. You must hit the "enter" key on your keyboard to submit a task. Once you do, you'll see a Metamask confirmation pop up. You must sign this transaction in order to create the task.

Complete Tasks

The accompanying video footage for this portion of the tutorial begins at 1:16:40. Now the last thing we'll do in this tutorial is "check off" the tasks in the todo list. Once we do, they will appear in the "completed" list, striked though. First, we'll update the smart contract. We'll add a TaskCompleted() event, and trigger it inside a new toggleCompleted() function like this:

```
pragma solidity ^0.5.0;

contract TodoList {

 // ...

 event TaskCompleted(

   uint id,

   bool completed

 );

 // ...
```

```
 function toggleCompleted(uint _id) public {
   Task memory _task = tasks[_id];
   _task.completed = !_task.completed;
   tasks[_id] = _task;
   emit TaskCompleted(_id, _task.completed);
 }
}
```

Now we'll write a test like this:

```
it('toggles task completion', async () => {
 const result = await this.todoList.toggleCompleted(1)
 const task = await this.todoList.tasks(1)
 assert.equal(task.completed, true)
 const event = result.logs[0].args
 assert.equal(event.id.toNumber(), 1)
 assert.equal(event.completed, true)
})
```

Now let's run the test:

```
$ truffle test
```

Yes, it passes! 🎉 Now let's deploy a new copy of the smart contract to the blockchain since the code has changed:

```
$ truffle migrate --reset
```

Now let's update the client side code. First we'll un-comment the event listener inside the renderTasks() function:

```
$newTaskTemplate.find('input')
```

```
 .prop('name', taskId)

 .prop('checked', taskCompleted)

 .on('click', App.toggleCompleted)
```

Now we'll add a toggleCompleted() function in the app.js file like this:

```
toggleCompleted: async (e) => {

 App.setLoading(true)

 const taskId = e.target.name

 await App.todoList.toggleCompleted(taskId)

 window.location.reload()

},
```

Now, find a task in the client side application and click the checkbox. Once you sign this transaction, it will check off the task from the todo list!

Congratulations! You have successfully built a full stack blockchain application powered by Ethereum smart contracts! Now we can start developing the green building blockchain rating system [GreenBlocks].”

End.

Appendix V Description of GreenBlocks

This appendix describes the technical approach to setting up the GreenBlocks green building rating system.

GreenBlocks uses Ethereum, and Ether and the “Gas” that fuels it is based on Solidity. Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behavior of accounts within the Ethereum state. Solidity was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, multi-signature wallets, and in our case rating a building. Some of the technology we had to use and work through are shown below.

Blocks and Validity

Anything done on the blockchain is called a “transaction” and GreenBlocks uses transactions to add information of any type on the global chain. One major obstacle to overcome is what is called a “double-spend attack”: What happens if two transactions exist in the network that both want to modify the same “account”? Only one of the transactions can be valid, typically the one that is accepted first. The problem is that “first” is not an objective term in a peer-to-peer network.

The abstract answer to this is that you do not have to care. A globally accepted order of the transactions will be selected for you, solving the conflict. The transactions will be bundled into what is called a "block" and then they will be executed and distributed among all participating nodes. If two transactions contradict each other, the one that ends up being second will be rejected and not become part of the block.

These blocks form a linear sequence in time and that is where the word "blockchain" derives from. Blocks are added to the chain in rather regular intervals - for Ethereum this is roughly every 17 seconds. As part of the "order selection mechanism" (which is called "mining") it may happen that blocks are reverted from time to time, but only at the "tip" of the chain. The more blocks are added on top of a particular block, the less likely this block will be reverted.

GreenBlocks adds a Validity Stamp to its User Interface to make the point reference to the first accepted "transaction" and certify on our private node. If a transaction needs to be completed on a timescale, the smart contract can have an appropriate alarm clock set at: https://www.ethereum-alarm-clock.com with a TimeNode. A TimeNode is an off-chain execution agent that acts as the counterparty to transactions that are scheduled on the Ethereum Alarm Clock. When someone schedules a transaction, it will later be executed by a TimeNode for profit.

The Ethereum Virtual Machine

The Ethereum Virtual Machine or EVM is the runtime environment for smart contracts in Ethereum. It is not only sandboxed but actually completely isolated, which means that code running inside the EVM has no access to network, filesystem or other processes. Smart contracts even have limited access to other smart contracts.

Most systems use either a public OR a private blockchain. GreenBlocks uses two kinds of accounts in Ethereum which share the same address space: External accounts that are controlled by public-private key pairs (i.e. humans) and contract accounts which are controlled by the code stored together with the account.

The address of an external account is determined from the public key while the address of a contract is determined at the time the contract is created (it is derived from the creator address and the number of transactions sent from that address, the so-called "nonce"). This was done to solve the internal record keeping issues of GreenBlocks and the certification hash that will take the data from rating a building (private node), and inextricably linking that information to the blockchain hash (public node) for confirmation.

Regardless of whether or not the account stores code, the two types are treated equally by the EVM. Every account has a persistent key-value store mapping 256-bit words to 256-bit words called storage. Furthermore, every account has a balance in Ether (in "Wei" to be exact, 1 ether is 10**18 wei) which can be modified by sending transactions that include Ether.

Transactions

A transaction is a message that is sent from one account to another account (which might be the same or empty, see below). It can include binary data (which is called "payload") and Ether. If the target account contains code, that code is executed and the payload is provided as input data. If the target account is not set (the transaction does not have a recipient or the recipient is set to null), the transaction creates a new contract. As already mentioned, the address of that contract is not the zero address but an address derived from the sender and its number of transactions sent (the "nonce"). The payload of such a contract creation transaction is taken to be EVM bytecode and executed. The output data of this execution is permanently stored as the code of the contract, in the case of GreenBlocks – our green building rating. This means that in order to create a contract, you do not send the actual code of the contract, but in fact code that returns that code when executed.

Upon creation, each transaction is charged with a certain amount of gas, whose purpose is to limit the amount of work that is needed to execute the transaction and to pay for this execution at the same time. While the EVM executes the transaction, the gas is gradually depleted according to specific rules.

Expressions and Control Structures

Most of the control structures known from curly-braces languages are available in Solidity:

There is: `if`, `else`, `while`, `do`, `for`, `break`, `continue`, `return`, with the usual semantics known from C or JavaScript. Solidity also supports exception handling in the form of `try`/`catch`-statements, but only for external function calls and contract creation calls.

Parentheses can *not* be omitted for conditionals, but curly braces can be omitted around single-statement bodies. Note that there is no type conversion from non-Boolean to Boolean types as there is in C and JavaScript, so `if (1) { ... }` is *not* valid Solidity.

Function Calls-Internal Function Calls

Functions of the current contract can be called directly ("internally"), also recursively, as seen in this nonsensical example:

```
pragma solidity >=0.4.16 <0.7.0;  contract C {     function g(uint a) public pure returns (uint ret) { return a + f(); }     function f() internal pure returns (uint ret) { return g(7) + f(); } }
```

These function calls are translated into simple jumps inside the EVM. This has the effect that the current memory is not cleared, i.e. passing memory references to internally-called functions is very efficient. Only functions of the same contract instance can be called internally.

You should still avoid excessive recursion, as every internal function call uses up at least one stack slot and there are only 1024 slots available.

External Function Calls

The expressions `this.g(8);` and `c.g(2);` (where `c` is a contract instance) are also valid function calls, but this time, the function will be called "externally", via a message call and not directly via jumps. Please note that function calls on `this` cannot be used in the constructor, as the actual contract has not been created yet. Functions of other contracts have to be called externally. For an external call, all function arguments have to be copied to memory.

```
pragma solidity >=0.4.0 <0.7.0;  contract InfoFeed {     function info() public payable returns (uint ret) { return 42; } }  contract Consumer {     InfoFeed feed;     function setFeed(InfoFeed addr) public { feed = addr; }     function callFeed() public { feed.info{value: 10, gas: 800}(); } }
```

You need to use the modifier `payable` with the `info` function because otherwise, the `value` option would not be available.

Named Calls and Anonymous Function Parameters

Function call arguments can be given by name, in any order, if they are enclosed in `{ }` as can be seen in the following example. The argument list has to coincide by name with the list of parameters from the function declaration, but can be in arbitrary order.

```
pragma solidity >=0.4.0 <0.7.0;  contract C {     mapping(uint => uint) data;
function f() public {        set({value: 2, key: 3});     }      function set(uint key, uint
value) public {        data[key] = value;     }  }
```

Omitted Function Parameter Names

The names of unused parameters (especially return parameters) can be omitted. Those parameters will still be present on the stack, but they are inaccessible.

```
pragma solidity >=0.4.16 <0.7.0;  contract C {     // omitted name for parameter
function func(uint k, uint) public pure returns(uint) {        return k;     } }
```

Creating Contracts via `new`

A contract can create other contracts using the `new` keyword. The full code of the contract being created has to be known when the creating contract is compiled so recursive creation-dependencies are not possible.

```
pragma solidity >=0.5.0 <0.7.0;  contract D {     uint public x;     constructor(uint a) public payable {         x = a;     } }  contract C {     D d = new D(4); // will be executed as part of C's constructor      function createD(uint arg) public {         D newD = new D(arg);         newD.x();     }      function createAndEndowD(uint arg, uint amount) public payable {         // Send ether along with the creation         D newD = new D{value: amount}(arg);         newD.x();     } }
```

As seen in the example, it is possible to send Ether while creating an instance of `D` using the `value` option, but it is not possible to limit the amount of gas. If the creation fails (due to out-of-stack, not enough balance or other problems), an exception is thrown.

Salted contract creations / create2

When creating a contract, the address of the contract is computed from the address of the creating contract and a counter that is increased with each contract creation.

If you specify the option `salt` (a bytes32 value), then contract creation will use a different mechanism to come up with the address of the new contract. It will compute the address from the address of the creating contract, the given salt value, the (creation) bytecode of the created contract and the constructor arguments.

In particular, the counter (“nonce”) is not used. This allows for more flexibility in creating contracts: You are able to derive the address of the new contract before it is created. Furthermore, you can rely on this address also in case the creating contracts creates other contracts in the meantime.

The main use-case here is contracts that act as judges for off-chain interactions, which only need to be created if there is a dispute.

```
pragma solidity >0.6.1 <0.7.0;  contract D {     uint public x;     constructor(uint a) public {         x = a;     } }  contract C {     function createDSalted(bytes32 salt, uint arg) public {         /// This complicated expression just tells you how the address         /// can be pre-computed. It is just there for illustration.         /// You actually only need ``new D{salt: salt}(arg)``.         address predictedAddress = address(bytes20(keccak256(abi.encodePacked(             byte(0xff),             address(this),             salt,             keccak256(abi.encodePacked(                 type(D).creationCode,                 arg             ))         ))));          D d = new D{salt: salt}(arg);         require(address(d) == predictedAddress);     } }
```

Order of Evaluation of Expressions

The evaluation order of expressions is not specified (more formally, the order in which the children of one node in the expression tree are evaluated is not specified, but they are of course evaluated before the node itself). It is only guaranteed that statements are executed in order and short-circuiting for Boolean expressions is done.

Assignment-Destructuring Assignments

Solidity internally allows tuple types, i.e. a list of objects of potentially different types whose number is a constant at compile-time. Those tuples can be used to return multiple values at the same time. These can then either be assigned to newly declared variables or to pre-existing variables (or LValues in general). Tuples are not proper types in Solidity, they can only be used to form syntactic groupings of expressions.

```
pragma solidity >0.4.23 <0.7.0;  contract C {     uint index;      function f() public pure returns (uint, bool, uint) {         return (7, true, 2);     }      function g() public {         // Variables declared with type and assigned from the returned tuple,         // not all elements have to be specified (but the number must match).         (uint x, , uint y) = f();         // Common trick to swap values -- does not work for non-value storage types.         (x, y) = (y, x);         // Components can be left out (also for variable declarations).         (index, , ) = f(); // Sets the index to 7     } }
```

It is not possible to mix variable declarations and non-declaration assignments, i.e. the following is not valid: `(x, uint y) = (1, 2);`

Complications for Arrays and Structs

The semantics of assignments are a bit more complicated for non-value types like arrays and structs. Assigning *to* a state variable always creates an independent copy. On the other hand, assigning to a local variable creates an independent copy only for elementary types, i.e. static types that fit into 32 bytes. If structs or arrays (including `bytes` and `string`) are assigned from a state variable to a local variable, the local variable holds a reference to the original state variable. A second assignment to the local variable does not modify the state but only changes the reference. Assignments to members (or elements) of the local variable *do* change the state.

In the example below the call to `g(x)` has no effect on `x` because it creates an independent copy of the storage value in memory. However, `h(x)` successfully modifies `x` because only a reference and not a copy is passed.

```
pragma solidity >=0.4.16 <0.7.0;  contract C {     uint[20] x;      function f() public {
g(x);        h(x);     }      function g(uint[20] memory y) internal pure {        y[2] = 3;
}      function h(uint[20] storage y) internal {        y[3] = 4;     } }
```

Scoping and Declarations

A variable which is declared will have an initial default value whose byte-representation is all zeros. The “default values” of variables are the typical “zero-state” of whatever the type is. For example, the default value for a `bool` is `false`. The default value for the `uint` or `int` types is `0`. For statically-sized arrays and `bytes1` to `bytes32`, each individual element will be initialized to the default value corresponding to its type. For dynamically-sized arrays, `bytes` and `string`, the default value is an empty array or string. For the `enum` type, the default value is its first member.

Scoping in Solidity follows the widespread scoping rules of C99 (and many other languages): Variables are visible from the point right after their declaration until the end of the smallest `{ }`-block that contains the declaration. As an exception to this rule, variables declared in the initialization part of a for-loop are only visible until the end of the for-loop.

Variables that are parameter-like (function parameters, modifier parameters, catch parameters, …) are visible inside the code block that follows - the body of the function/modifier for a function and modifier parameter and the catch block for a catch parameter.

Variables and other items declared outside of a code block, for example functions, contracts, user-defined types, etc., are visible even before they were declared. This means you can use state variables before they are declared and call functions recursively.

As a consequence, the following examples will compile without warnings, since the two variables have the same name but disjoint scopes.

```
pragma solidity >=0.5.0 <0.7.0; contract C {     function minimalScoping() pure
public {        {           uint same;           same = 1;        }        {           uint same;
same = 3;        }     } }
```

As a special example of the C99 scoping rules, note that in the following, the first assignment to `x` will actually assign the outer and not the inner variable. In any case, you will get a warning about the outer variable being shadowed.

```
pragma solidity >=0.5.0 <0.7.0; // This will report a warning contract C {     function f() pure public returns (uint) {         uint x = 1;         {             x = 2; // this will assign to the outer variable             uint x;         }         return x; // x has value 2     } }
```

Error handling: Assert, Require, Revert and Exceptions

Solidity uses state-reverting exceptions to handle errors. Such an exception undoes all changes made to the state in the current call (and all its sub-calls) and flags an error to the caller.

When exceptions happen in a sub-call, they “bubble up” (i.e., exceptions are rethrown) automatically. Exceptions to this rule are `send` and the low-level functions `call`, `delegatecall` and `staticcall`: they return `false` as their first return value in case of an exception instead of “bubbling up”.

Exceptions can be caught with the `try`/`catch` statement.

`assert` and `require`

The convenience functions `assert` and `require` can be used to check for conditions and throw an exception if the condition is not met.

The `assert` function should only be used to test for internal errors, and to check invariants. Properly functioning code should never reach a failing `assert` statement; if this happens there is a bug in your contract which you should fix. Language analysis tools can evaluate your contract to identify the conditions and function calls which will reach a failing `assert`.

An `assert`-style exception is generated in the following situations:

1. If you access an array or an array slice at a too large or negative index (i.e. `x[i]` where `i >= x.length` or `i < 0`).
2. If you access a fixed-length `bytesN` at a too large or negative index.
3. If you divide or modulo by zero (e.g. `5 / 0` or `23 % 0`).
4. If you shift by a negative amount.
5. If you convert a value too big or negative into an enum type.
6. If you call a zero-initialized variable of internal function type.
7. If you call `assert` with an argument that evaluates to false.

The `require` function should be used to ensure valid conditions that cannot be detected until execution time. This includes conditions on inputs or return values from calls to external contracts.

A `require`-style exception is generated in the following situations:

1. Calling `require` with an argument that evaluates to `false`.
2. If you call a function via a message call but it does not finish properly (i.e., it runs out of gas, has no matching function, or throws an exception itself), except when a low level operation `call`, `send`, `delegatecall`, `callcode` or `staticcall` is used.

The low level operations never throw exceptions but indicate failures by returning `false`.

3. If you create a contract using the `new` keyword but the contract creation does not finish properly.
4. If you perform an external function call targeting a contract that contains no code.
5. If your contract receives Ether via a public function without `payable` modifier (including the constructor and the fallback function).
6. If your contract receives Ether via a public getter function.
7. If a `.transfer()` fails.

You can optionally provide a message string for `require`, but not for `assert`.

The following example shows how you can use `require` to check conditions on inputs and `assert` for internal error checking.

```
pragma solidity >=0.5.0 <0.7.0;  contract Sharer {     function sendHalf(address
payable addr) public payable returns (uint balance) {        require(msg.value % 2 == 0,
"Even value required.");        uint balanceBeforeTransfer = address(this).balance;
addr.transfer(msg.value / 2);         // Since transfer throws an exception on failure and
// cannot call back here, there should be no way for us to         // still have half of the
money.         assert(address(this).balance == balanceBeforeTransfer - msg.value / 2);
return address(this).balance;     } }
```

Internally, Solidity performs a revert operation (instruction `0xfd`) for a `require`-style exception and executes an invalid operation (instruction `0xfe`) to throw an `assert`-style exception. In both cases, this causes the EVM to revert all changes made to the state. The reason for reverting is that there is no safe way to continue execution, because an expected effect did not occur. Because we want to keep the atomicity of transactions, the safest action is to revert all changes and make the whole transaction (or at least call) without effect.

In both cases, the caller can react on such failures using `try`/`catch` (in the failing `assert`-style exception only if enough gas is left), but the changes in the caller will always be reverted.

The foregoing description is the developer's account of how the GreenBlocks system was developed. A printout of data uploaded to the blockchain would look simply like a data printout, and would not be proof of work. Therefore the developers wrote this technical explanation of the procedures they used to successfully develop the app and upload live building data to it.

References

(2018). (ConsenSys Real Estate) Retrieved March 21, 2018, from Pangea: http://www.pangea.io/home

AAA. (2018, May 8). *Green Car Guide.* Retrieved May 25, 2018, from AAA Newsroom: https://newsroom.aaa.com/tag/green-car-guide/

Aamidor, J. (2017, September 19). *Blockchain: Coming to a Smart Building Near You?* Retrieved October 2, 2018, from GreenTech Media: Industry Perspective: https://www.greentechmedia.com/articles/read/blockchain-coming-to-a-smart-building-near-you#gs.C21aa0I

Alter, L. (2011, August 17). *Henry Gifford's $100M LEED Lawsuit.* Retrieved May 11, 2019, from Treehugger: https://www.treehugger.com/sustainable-product-design/henry-giffords-100m-leed-lawsuit-dismissed-with-prejudice.html

Alter, L. (2011, August 17). *Henry Gifford's $100M LEED Lawsuit Dismissed "With Prejudice".* Retrieved May 9, 2018, from Treehugger: https://www.treehugger.com/sustainable-product-design/henry-giffords-100m-leed-lawsuit-dismissed-with-prejudice.html

American Chemistry Council. (2003, April 16). *Analyzing the Cost of Obtaining LEED Certification.* Retrieved February 25, 2020, from Green Building Solutions: https://www.greenbuildingsolutions.org/wp-content/uploads/2016/05/LEED-Cost-Analysis-Report.pdf

American Chemistry Council. (2003, April 16). *Analyzing the Cost of Obtaining LEED Certification.* Retrieved February 25, 2020, from https://www.greenbuildingsolutions.org/wp-content/uploads/2016/05/LEED-Cost-Analysis-Report.pdf

ASHRAE. (2016). *Energy Standard for Buildings Except Low-Rise Residential Buildings.* Atlanta: American Society of Heating Refrigerating and Air-Conditioning Engineers.

Bacorn, K. (2018). *LEED Simplified - Finally!* (1st ed.). Charleston, SC: Amazon CreateSpace.

Bajpai, P. (2018, February 6). *Bitcoin Vs Ethereum: Driven by Different Purposes.* Retrieved March 21, 2018, from Investopedia: https://www.investopedia.com/articles/investing/031416/bitcoin-vs-ethereum-driven-different-purposes.asp

Bitcoin ATMs in New York. (2018, March 21). Retrieved March 21, 2018, from Coin ATM Radar: https://www.google.com/search?q=photo+bitcoin+atm+nyc&oq=photo+bitcoin+atm+nyc&aqs=chrome..69i57j69i59l2j69i60l2j69i61.5226j0j4&sourceid=chrome&ie=UTF-8#

BitInfo. (2018, October 19). *Ethereum Average Fee Chart.* Retrieved October 19, 2018, from Bit Info Charts: https://bitinfocharts.com/comparison/ethereum-transactionfees.html

Blockchain & Ethereum Security (2016). [Motion Picture].
Bureau of Labor Statistics. (2018). *Careers in Green Construction.* Retrieved October 22, 2018, from Green Jobs: https://www.bls.gov/green/construction/
Buterin, V. (2013, November). *Ethereum White Paper: A Next Generation Smart Contract and Decentralized Application Platform.* Retrieved October 22, 2018, from Ethereum Project: https://whitepaperdatabase.com/wp-content/uploads/2017/09/Ethereum-ETH-whitepaper.pdf
Buterin, V. (2017, September 18). Decentralizing Everything. *Disrupt SF*, https://techcrunch.com/video/decentralizing-everything-with-ethereums-vitalik-buterin/. (N. Ravikant, Interviewer) Disrupt SF.
Car & Driver. (2018). *Hybrid and Electric Cars 2018-2019: The Best and the Rest.* Retrieved May 25, 2018, from Car & Driver: https://www.caranddriver.com/best-hybrid-electric-cars
Carter, S. M. (2017, December 8). *What a 20-something bitcoin millionaire learned from going to prison and starting over*. Retrieved March 21, 2018, from CNBC Money: https://www.cnbc.com/2017/12/08/what-bitcoin-millionaire-charlie-shrem-learned-from-going-to-prison.html
CNN. (2015, October 20). *The world's 10 most legendary luxury hotels.* Retrieved March 21, 2018, from CNN Travel: https://www.cnn.com/travel/article/world-most-iconic-hotels/index.html
Coindesk. (2019, May 19). *Bitcoin Price (BTC).* Retrieved May 19, 2019, from Coindesk Data: https://www.coindesk.com/price/bitcoin
CoinLore. (2019, May 10). *Cryptocurrencies 2101.* Retrieved May 10, 2019, from CoinLore.com: https://www.coinlore.com/all_coins
ConsenSys. (2018). *Meridio*. Retrieved October 12, 2018, from https://www.meridio.co/?ref=pangea
ConsenSys Media. (2018, January 11). *Real Estate Needs a Revolution.* Retrieved May 10, 2019, from Meridio: https://media.consensys.net/using-blockchain-to-expand-access-to-real-estate-4a2e3fb15f90
Cornel, J. (2017, October). *How many Americans have heard of Bitcoin?* Retrieved from Bitcoin: https://steemit.com/bitcoin/@jrcornel/how-many-americans-have-heard-of-bitcoin-check-out-this-survey-to-find-out
Daley, B. (2018, February 21). *Greenwashing the property market.* Retrieved May 11, 2019, from The Conversation: http://theconversation.com/greenwashing-the-property-market-why-green-star-ratings-dont-guarantee-more-sustainable-buildings-91655
Davis, K. (2018, March 19). *Bitcoin prosecutions may break new legal ground.* Retrieved March 21, 2018, from San Diego Union Tribune: http://www.sandiegouniontribune.com/news/courts/sd-me-bitcoin-prosecutions-20180319-story.html
Del Percio, S. (2010, May 26). *Unit Owners File Suit Against LEED Gold-Hopeful Riverhouse.* Retrieved May 11, 2019, from Green Real Estate Law Journal: http://www.greenrealestatelaw.com/2010/05/unit-owners-file-suit-against-leed-gold-hopeful-riverhouse-in-battery-park-city/
Dodge. (2016). *World Green Building Trends.* Retrieved March 2018, from Smart Market Report:

http://fidic.org/sites/default/files/World%20Green%20Building%20Trends%202016%20SmartMarket%20Report%20FINAL.pdf
Ely, B. (2017, December 11). *Bitcoin is a Ponzi scheme, and it will collapse like one.* Retrieved April 19, 2018, from The Hill: http://thehill.com/opinion/finance/364306-bitcoin-is-a-ponzi-scheme-and-it-will-collapse-like-one
EPA. (2019). *What your 1 – 100 ENERGY STAR score means.* Retrieved October 28, 2019, from energystar.gov: https://www.energystar.gov/buildings/facility-owners-and-managers/existing-buildings/use-portfolio-manager/interpret-your-results/what
Ethereum.org. (2019). *What is Ethereum?* Retrieved October 26, 2019, from Ethereum.org: https://www.ethereum.org/beginners/
Evans, J. (2016, January 2). *Why Bitcoin Matters.* Retrieved March 21, 2018, from Tech Crunch: https://techcrunch.com/2016/01/02/why-bitcoin-matters/
Fedrizzi, S. R. (n.d.). *Voice of LEED Certification.* Retrieved February 25, 2020, from American Business Journal: http://www.abjusa.com/associations/u_s_green_building_council_the_voice_of_leed_certification.html?print
Fuller, F. (2016, January 28). *A Comparison of LEED and Green Globes.* Retrieved May 3, 2018, from GBRI Online: https://www.gbrionline.org/a-comparison-of-leed-and-green-globes/
GBI. (2016). *Green Globes Technical Reference Manual.* Portland, OR: Green Building Initiative.
Geyl, A. (2016, November 6). *Bitcoin Network Statistics.* Retrieved May 19, 2019, from Neighbourhood Pool Watch: http://organofcorti.blogspot.com/
GSA. (2018, February 16). *GSA Properties.* Retrieved March 21, 2018, from https://www.gsa.gov/real-estate/gsa-properties
Harvard. (2012). *A Guide to the ALM Thesis* (7th ed.). Cambridge, MA: Harvard University Extension School Master of Liberal Arts Program. Retrieved May 3, 2018, from Harvard University Extension School Master of Liberal Arts Program: https://www.extension.harvard.edu/sites/default/files/forms/ext_almg1.pdf
Henderson, K., Knoll, E., & Rogers, M. (2018, March). *What every utility CEO should know about blockchain.* Retrieved October 12, 2018, from McKinsey.com: https://www.mckinsey.com/industries/electric-power-and-natural-gas/our-insights/what-every-utility-ceo-should-know-about-blockchain
Hobey, E. (2018, February 1). *IdentityMind Global Introduces KYC and Anti-Money Laundering Plug-in for ICOs.* Retrieved April 6, 2018, from CrowdFund Insider: https://www.crowdfundinsider.com/2018/02/127840-identitymind-global-introduces-kyc-anti-money-laundering-plug-icos/
Ingram, M. (2017, May 29). *Bitcoin Fans Say Cryptocurrency Tokens Are the Future of Tech Funding.* Retrieved March 21, 2018, from Fortune Magazine: http://fortune.com/2017/05/29/bitcoin-crypto-tokens/
Investing. (2018, March 21). *All Cryptocurrencies.* Retrieved March 21, 2018, from Investing.com: https://www.investing.com/crypto/currencies

Ivanova, E., & Gilliland, M. (2017, September 11). *7 Ways The Blockchain Can Save The Environment and Stop Climate Change.* Retrieved October 2, 2018, from Future Thinkers: https://futurethinkers.org/blockchain-environment-climate-change/

IWBI. (2017). *The Well Building Standard* (Vol. 1). International Well Building Institute.

Marr, B. (2017, January 24). *A Complete Beginner's Guide To Blockchain.* Retrieved October 27, 2019, from Forbes Magazine: https://www.forbes.com/sites/bernardmarr/2017/01/24/a-complete-beginners-guide-to-blockchain/#6b83dcf56e60

Mulder, J. (n.d.). *NY counties with highest levels of cancer-causing radon.* Retrieved October 27, 2019, from newyorkupstate.com: https://www.newyorkupstate.com/news/2017/03/see_ny_counties_with_highest_levels_of_cancer-causing_radon_gas.html

Nicolow, J. (2008, November 1). *Measuring The Cost To Become LEED Certified.* Retrieved from Building Operations Management: https://www.facilitiesnet.com/green/article/Measuring-The-Cost-To-Become-LEED-Certified--10057

NIH. (2018, January 9). *Blockchain Based Decentralized Management of Demand Response Programs in Smart Energy Grids.* Retrieved October 12, 2018, from National Center for Biotechnology Information, U.S. National Library of Medicine, National Institutes of Health: https://www.ncbi.nlm.nih.gov/pubmed/29315250

NRDC. (2016, April 28). *Should You Buy Carbon Offsets? A practical and philosophical guide to neutralizing your carbon footprint.* Retrieved May 9, 2018, from Natural Resources Defense Council: https://www.nrdc.org/stories/should-you-buy-carbon-offsets

NYC Dept. of Buildings. (2020). *Buildings.* Retrieved February 25, 2020, from NYC.gov: https://www1.nyc.gov/site/buildings/about/about.page

O'Connell, J. (2017, November 18). *Blockchain is Making Inroads into the Environment Sustainability Industry.* Retrieved May 25, 2018, from CCN: https://www.ccn.com/blockchain-making-inroads-sustainability-industry/

O'Leary, K. (2018, March 19). NYC hotel wants to use a $400 million cryptocoin offering to sell ownership like a stock. *CNBC Squawk Box.* (R. Quick, Interviewer) CNBC. New York.

Orcutt, M. (2018, April 25). *How secure is blockchain really?* Retrieved May 25, 2018, from MIT Technology Review: https://www.technologyreview.com/s/610836/how-secure-is-blockchain-really/

Overdeck, L. (2018, July 24). *How Many Houses?* Retrieved February 25, 2020, from Bedtime Math: http://bedtimemath.org/fun-math-buildings-in-world/

Ranasinghe, D. (2013, December 24). *What you need to know about digital currencies.* Retrieved March 21, 2018, from CNBC: https://www.cnbc.com/2013/12/24/what-you-need-to-know-about-digital-currencies.html

Saleem, O. (2018, April 17). *Uses of Blockchain in Facility Management.* Retrieved October 12, 2018, from A New Deal for Buildings: https://newdeal.blog/uses-of-blockchain-in-facility-management-53e3f76d62ca

Samarasekera, R. (2017, March 6). *LEED credits, prerequisites and points.* Retrieved May 11, 2019, from USGBC.org: https://www.usgbc.org/articles/whats-difference-between-leed-credit-leed-prerequisite-and-leed-point

Smith, A. (2018, March 7). *Bitcoin plunges after SEC seeks to rein in cryptocurrencies*. Retrieved March 21, 2018, from CNN Tech: http://money.cnn.com/2018/03/07/technology/bitcoin-sec-cryptocurrency/index.html

Stanley, S. (2019, November 20). *USGBC Announces Vision for LEED Positive.* Retrieved February 25, 2020, from Green Business Certification, Inc.: https://www.gbci.org/usgbc-announces-vision-leed-positive

Stoddard, R. (2018, April 24). *How the blockchain could transform sustainability reporting.* Retrieved May 2, 2018, from Greenbiz: https://www.greenbiz.com/article/how-blockchain-could-transform-sustainability-reporting

Sustainable Brands. (2014, December 31). *60% of Americans Resolving More Environmentally Responsible Lifestyles in 2015.* Retrieved October 22, 2018, from Stakeholder Trends & Insights: https://www.sustainablebrands.com/news_and_views/stakeholder_trends_insights/sustainable_brands/survey_60_americans_resolving_more_env

Troianovski, A. (2010, August 16). *Empire State Manager Fuels Energy Debate.* Retrieved October 22, 2018, from The Wall Street Journal: https://www.wsj.com/articles/SB1000142405274870429670457543164083355223 2

USGBC. (2002, July 12). *Environmental Design & Construction.* Retrieved February 25, 2020, from https://www.greenbuildingsolutions.org/wp-content/uploads/2016/05/LEED-Cost-Analysis-Report.pdf

USGBC. (2013). *LEED Reference Guide.* Washington, DC: U.S. Green Building Council.

USGBC. (2015). *2014 Annual Report.* Retrieved October 12, 2018, from usgbc.org: https://readymag.com/usgbc/2014AnnualReport/13/

USGBC. (2020). *LEED Certification fees.* Retrieved January 13, 2020, from U.S. Green Building Council: https://new.usgbc.org/cert-guide/fees

USGBC. (n.d.). *Buildings and Climate Change.* Retrieved October 12, 2018, from Environmental and Energy Study Institute: https://www.eesi.org/files/climate.pdf

Vitalik Buterin already works on countering quantum computer hacks threat. (2017, November 15). Retrieved April 18, 2018, from BitNews Today: https://bitnewstoday.com/news/ethereum/vitalik-buterin-already-works-on-countering-quantum-computer-hacks-threat/

Vorabutra, J.-A. (2016, October 3). *Why Blockchain is a Game Changer for Supply Chain Management Transparency.* Retrieved October 2, 2018, from Supply Chain 247.com: https://www.supplychain247.com/article/why_blockchain_is_a_game_changer_fo r_the_supply_chain

Walker, L. (2017, September 19). *This new carbon currency could make us more climate friendly.* Retrieved May 29, 2018, from Sustainable Developmenmt Impact Summit:

https://intelligence.weforum.org/topics/a1Gb00000038qmPEAQ?tab=publications
Walker, L. (2017, September 19). *This New Carbon Currency Could Make Us More Climate Friendly.* Retrieved May 29, 2018, from Sustainable Development Impact Summit: https://intelligence.weforum.org/topics/a1Gb00000038qmPEAQ?tab=publications
Ward, S. (2014, March 14). *Harder than Passing the Bar?* Retrieved October 22, 2018, from Poplar: https://www.poplarnetwork.com/news/how-hard-leed-exam-harder-passing-bar
Wilcox, K. (2013, November 5). *GSA Issues Recommendations on Green Rating Systems.* Retrieved March 21, 2018, from Civil Engineering Magazine: https://www.asce.org/magazine/20131105-gsa-issues-recommendations-on-green-ratings-systems/

Films and Videos Consulted

Glickman, A. (2018, May 21). Green Building Expo 2018 - Introduction To Blockchain & Its Applications to Renewable Energy. [Video File]. https://www.youtube.com/watch?v=Cm5iRXujT8s

Cannucciari, C. (2017, January 6). *Banking on Bitcoin.* [Full-length Documentary]. https://www.netflix.com/watch/80154500?trackId=13752289&tctx=0%2C0%2Caaae0dcc3a3a0ca56a57c17756fae491dde31d6f%3A5d821567edf657f56aadd0c51935275c2ed3f41b%2C%2c

[Yale School of Forestry & Environmental Studies]. (2018, March 2). *Blockchain for Sustainable Solutions*. [Full-Length Documentary]. https://www.youtube.com/watch?v=uC_IvtXKcHQ

[Future Thinkers.org]. (2017, June 15). *19 Industries The Blockchain Will Disrupt*.

Article is at: https://futurethinkers.org/industries-blockchain-disrupt/ Video is at: https://www.youtube.com/watch?v=G3psxs3gyf8

[IBM Taiwan]. (2017, February 28). *Blockchain Technology in Sustainability.* [Video File]. https://www.youtube.com/watch?v=Jxnk5CkH4vM

Ivanova, E. & Gilliland, M. (2017, September 11). *7 Ways The Blockchain Can Save The Environment and Stop Climate Change.* [Video File]. https://www.youtube.com/watch?v=58xtN6Dw8kw

Complexity Labs. (2018, May 30). Blockchain & Environment. [Video File]. https://www.youtube.com/watch?v=wXcRTt4dB9Q&t=671s

TriUnity Partners, PTE, Ltd. (2018, November 8). Blockchain for Green Building. https://www.youtube.com/watch?v=bMTZzHBm1G4

Endnotes

[1] U.S. Green Building Council was formed in Syracuse, and Syracuse University's College of Environmental Science and Forestry, where this student author has taught, is ranked by U.S. News & World Report (2016) in the top 30 public colleges in the country.
[2] Cryptocurrencies such as Bitcoin, have not been widely used for real estate sales due to volatility of value. In spring 2019, the value of Bitcoin in dollars doubled in less than one month (Coindesk, 2019). Only a few dozen sales transacted entirely in Bitcoin are on record nationwide.
[3] Name searches were performed with U.S. Patent & Trademark Office, National Business Register, National Corporation Directory, Small Business Administration and other sources. These searches are not cited in References, because name of the rating system is considered tangential to the hypothesis of this thesis.
[4] Peer verification means that a large group of data miners operating computer nodes agree on veracity of data, and no one miner, or group of miners, can change the data without knowledge and agreement of the others.
[5] Skanska, the Swedish construction company, was awarded LEED points at Empire State Building for "education" because of signs saying how faucets save water, windows save electricity, and stairs save energy, etc. The same office, Skanska, got LEED points for offering a shower to bicycle commuters, but the shower was filled with office supplies, and could not be used for the stated purpose of showering. This information is based on this thesis author's personal tour of Skanska headquarters in which Skanska told her they received LEED points for "innovation" in education for the signs, as well as points for bike showers used as storage. As a very experienced LEED AP, this author has seen similar phenomena on many projects; education is in fact the most common way projects get "bonus points" in LEED.
[6] Mr. Malkin was not being entirely facetious. LEED, at the time, limited points for energy conservation, but Mr. Malkin was told the skyscraper would qualify for a higher Platinum grade (the building is Gold) if he spent another $4.5 million for plants on its set-backs, thus qualifying for credits such as green roof, open space, water management, etc.
[7] The requirement was a mandatory prerequisite, meaning the property could not be certified without it. During the five months of desperately searching for someone in the organization who understood the requirement, the client (a food retailer) decided not to certify for LEED on the basis that such an intrusive survey would alienate their suppliers and impede their ability to do business.
[8] USGBC. (2013). *LEED Reference Guide.* Washington, DC: U.S. Green Building Council. Page 775 for innovation and page 737 for sky.
[9] For example, this thesis author worked on a LEED project that required complicated energy calculations. Although the calculations were completely unnecessary - as any building that meets NYC building code exceeds LEED requirements - USGBC required the team to perform, and pay for, the costly, inapplicable calculations.
[10] The official website, Energystar.gov, explains the numeric rating. "The 1 – 100 ENERGY STAR score is a screening tool that helps you assess how your building is performing. It'll help you identify which buildings in your portfolio to target for improvement or recognition. A score of 50 is the median. So if your building scores below 50, it means it's performing worse than 50 percent of similar buildings nationwide, while a score above 50 means it's performing better than 50 percent of its peers. And a score of 75 or higher means it's a top performer and may be eligible for ENERGY STAR certification.
[11] On October 30, 2018, a rep from USGBC specifically confirmed this in an email. He offered, as a rare example of transparency, a developer who does make their LEED Scorecard public. https://www.usgbc.org/projects/wells-fargo-tower?view=scorecard
[12] The author of this thesis has personal knowledge of a group who once certified a building that did not exist. They were able to accomplish this because LEED's self-certification system requires an agent, hired by the building owner, to verify the building's energy performance. LEED did not, at the time, require

submission of deeds, building permits, certificates of occupancy or even photos to prove that the building existed.

[13] In 2011, New York building super Henry Gifford sued U.S. Green Building Council for $100 million for fraudulent misrepresentation (Alter, Henry Gifford's $100M LEED Lawsuit Dismissed "With Prejudice", 2011) over their claim that LEED buildings save energy; his contention was that they actually use more energy, based on USGBC's own published data. The court dismissed Gifford's case without benefit of judicial review, based on Gifford's lack of standing as a plaintiff.

[14] GSA's portfolio includes nearly 400 million square feet.

[15] As of October 28, 2019, the dollar value of that 10,000 Bitcoin pizza would be $94,157,000 according to Coindesk.com Bitcoin Calculator.

[16] At this writing, New York City has 131 Bitcoin ATMs. https://coinatmradar.com/state/33/bitcoin-atm-new-york/

[17] "In 2005, Nick Szabo came out with the concept of "secure property titles with owner authority", a document describing how "new advances in replicated database technology" will allow for a blockchain-based system for storing a registry of who owns what land, creating an elaborate framework including concepts such as homesteading, adverse possession and Georgian land tax. However, there was unfortunately no effective replicated database system available at the time, and so the protocol was never implemented in practice."

[18] Local governments collect large amounts of environmental data. For example, New York City requires all buildings above 50,000 square feet to have an ASHRAE Level II energy audit. This involves having a professional firm analyze 2-3 years of utility bills to evaluate the facility's energy demand and use profiles to identify appropriate energy conservation measures.

[19] Once a building is certified in LEED, it keeps the status forever, though a building is allowed to certify more than once. Although USGBC reserves the right to de-certify a building, to this date, it has never happened. USGBC does not update a project's rating, although owners are allowed to buy a "dynamic plaque" which shows fluctuating energy use.

[20] These ballpark estimates come from the thesis author's extensive experience working as a LEED AP and Green Globes Professional as well as interviewing and hiring numerous engineers, APs and commissioning agents on many projects.

[21] Documentation costs do not vary much by project size because LEED requires all projects to document approx. 100 criteria. Small, medium and large buildings all must submit the same 100 criteria, so time demands are relatively stable.

[22] The registration fee to apply for LEED certification is $1500. If a property never submits any documentation for review, the owner will never pay certification fees, nor will the building be certified; it will remain registered for a few years. As an analogy, signing up and paying the entry fee to compete in a marathon does not make someone a marathon runner. Completing 26 miles makes them a marathon runner. A person who buys a ticket to compete in a marathon is "registered." If they actually run the race, they are "certified." USGBC's 100,000 building number includes both registered and certified projects. Therefore, the 100,000 number includes many buildings that are not green at all; the owner has simply filled out a form and paid $1500 to "register." According to a statistical analysis of the number of registered and certified buildings performed personally by this author in 2012-2015, as many as half of all registered buildings never become certified. Therefore, the 100,000 number implies there are that many green buildings, when in fact a large percentage of those buildings have never and will never "run the green race," but their owners merely filled out a qualifying form.

[23] Although the number of buildings in the world is impossible to determine, estimates range from 20 – 35 million (Overdeck, 2018). Assuming the low figure of 20 million, LEED buildings would be half of one percent of all buildings, including houses.