

What is Blockchain?

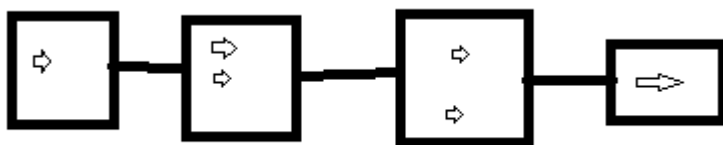
A simple and easy way to understand Blockchain is to think of a group of computers using a Data base. These computers share bits of Data mB, over a specific program with specific rules. The computers store and access these bits of Data in blocks that lock permanently. The locking up of these blocks of information by Miners is performed over a set amount of time that allows a number transactions to enter the Block of Data that is to be added on the Chain. Each block opens and closes over an amount of time, and the closing of the block is then confirmed by all computers on the Network and a new block starts to be produced. After several blocks are produced the Network agrees on which blocks are valid, and thus adding new blocks to the chain every so often. Each block only has a specific amount of “write” time, reading of the block information is always available (in some models not) but write to the Chain, ie change information or input Data is open for specific sets of time. We can call it Block production Time and the “Miners” do the Production. “Nodes” verify this production. All the computers on the Data base are on the same “time” on the network writing to the same “Block” that is currently being Produced onto the Chain. The Database (Blocks) of Data are compiled into a Chain of Blocks that stack one after the other. As each block is confirmed to the Chain its Data becomes Locked to any other Changes or Inputs of new Data. No new information is then allowed to be added or changed on the previous Blocks of Data.

That’s how to make just an ordinary Data base with several users become an Immutable storage of transaction data between the users. A big difference to a normal data base is that in the data shared between the users is Locked or Secured in the building blocks of the Data chain. The security of the Chains ability to resist changes to previous blocks of information is dependent on the “Hash Function” used by that chain.

The Data we want to store and Share in Bitcoin Blockchain is the UTXO data, the (unspent Transactions) is actually the information that makes up Blocks of Data, plus Proof of Work (or Stake) “DASH”. This information block is coded into a special Hash Function (Bitcoin is SHA 256). Each block will contain Hashes that code and can be Decoded for verification. The building and Verification of the SHA 256 hash functions in each block is dependent upon the

previous blocks. If any amount of Data is changed to a Block then the entire Hash Function will change and therefore Nodes will flag the Block as Invalid. Due to the misaligned Hash Function Encoding.... If someone or anyone wishes to change or rearrange the Data to their benefit it will be Very Very difficult due to the way that Blockchain uses Each piece of Data to Build Blocks of information and stack these blocks on top of eachother (encoding all into Hashes). Hash functions are the encoded bits of Data that ensure the original Data will match to the original Input, any changes to the Input change the Hash and thus Every Block thereafter will reject a Bad Hash.

This is the job of the Miners, to assemble the Blocks of Transaction Data using the Proof of Work algorithm. Changing previously produced blocks means that any user must also change the subsequent Blocks of Data on all the Computers at the Same time. This is how the Blockchain is able to Secure Data in a Way that makes Very Very difficult if not impossible to change what was happening in the History of this Data Base.



Each new block of data is dependent on the Previous Block information. Each arrow would represent a Transaction that the miners add to the Blocks of Data. These groups of transactions once assembled and Mined become the Blocks that make up Blockchain.

For Users of Blockchain we must understand why it is such a Secure means to maintain Immutable Data, and What benefits we can achieve from keeping an Accurate measure of History. Prevent double spending of the UTXO is the primary Benefit from Bitcoin Blockchain.

An entity would not be able to change any previous blocks of Data without affecting the entire Chain.

Ok so the Basic way that a Blockchain works is kinda described, for anyone wishing to understand more and Learn more a Deep Dive is really required. Blockchain Technology is a revolutionary way of making Decentralized Systems function with eachother and provide a means for all the

participants to transact in a Trustless way. Meaning I do not need to Trust anyone but the Hash Function and the program Code used to create Blocks of Information in our Chain Database.

Ok, so Blockchain is better than Sliced Bread! How will it help me? What advantages will Blockchain give our businesses and personal lives? Which Blockchain is the Best?

A really deep dive into this new technology will lead anyone who is curious down a path of constant learning and Growing. The intention of this Article is to arrive at a point of Awareness that should enlighten the reader to new possibilities and potential unforeseen use from this amazing Technology. Basically learning about Blockchain can be a computer Science course of study as you may have started to figure out. We will just cover Basics, very Basic, less than that even, ,, hahahahah lol. At some point though it can be imagined that generally a majority of People will understand and use Blockchain the same as they use Facebook or Twitter. Incorporating this technology into their lives, maybe without even realizing it, possibly understanding as well, even I who write this have many things to learn and struggle with.

What is [Ethereum](#)?

Ethereum Virtual machine is a specific Blockchain that was developed as a result of the Bitcoin Blockchain; lack of features, lack of adaptation ability, as a way to increase transactions and speed of Block production. Some of these features are also why Bitcoin is still remaining as the top UTXO Blockchain in Global Markets. Resistance to change has some advantage, we can just say it like that.

We will utilize [Etherscan](#) as the Primary Block explorer for examples and ease of use. That ease of integration with and use of our Contracts can be accomplished using One additional tool [MetaMask](#) webBrowser extension. To make user information a bit more friendly to Find and Look At inside [Ethereum](#) Virtual Machine. The UTXO model is not used in Ethereum and a different way of accounting the Balances of the Wallets is used called the Account Model.

Ethereum has grown into the second largest by market capital Blockchain currently operating in the World. The features and functions that can be

performed in and recorded on the Ethereum Blockchain have surpassed Bitcoin in Abilities and in ease of Development.

Ethereum like Bitcoin uses a hashing function to secure the bits of Data, and like Bitcoin users will interface with the Blockchain using Public and Private Addresses Keys, Wallets, and nodes. However the UTXO mode of transactions are specific to Bitcoin, as Ethereum uses an Account Model of transactions. The Account Model of transactions is easier to Track and Find the sources of all transactions related, and a UTXO method removes the ease of finding the sender in some cases, depending on how the transaction is assembled for broadcast.

Ethereum blockchain data is as well immutable (meaning the past blocks can't be changed) and public use of the Network is not restricted. So a large public access and publicly available networked Blockchain is the Basic View of Ethereum. The advantages that Ethereum has brought into Blockchain Technology are the very different reasons and compelling use cases that cause it to be the selected Blockchain of Choice for Our Scenario and Our Productions, for us to Use. Other types and variations of Blockchains exist, however that will not be discussed here as Ethereum is currently the most popular and therefore used World Wide and is still a developing Blockchain in its core features. Bitcoin may well be developing in its core, however it retains several reasons for remaining in the undeveloped stage for indefinite amount of time. Additionally Ethereum provides the ability to have an "Operational Code" inside of a created Public Address this will be our "Smart Contract"

The most Specific advantage of Ethereum use and the Ethereum Blockchain is the ability to Operate fully Automatic Smart Contracts within the Network. Other possible Smart Contract operating blockchains do exist however, Ethereum has the most development and Most ability for interaction of Smart Contracts with eachother on scale.

So our main point of discussion with Ethereum is the "Smart Contract". Ethereum as a "coin" or "transaction enabler" is part of how the Network funds itself, the miners, and end users. Ethereum transactions are paid with bits of Gas, to the miners that compile the various transactions into the Blocks during mining. Smart Contracts are "Lines of Code" sitting in the Ethereum Blockchain. These "Lines of Code" are given a public Address, the same as any

user Address, however they do not have in themselves a Private Key to make the Transactions. The transactions that a Smart Contract can perform, receive, and react with must be written into the “Lines of Code” prior to Deployment onto the Blockchain. The immutability feature of Blockchain keeps the code safe, and all its functions, in some cases the code has bugs or failures and this results in bad things happening on the Blockchain. The famous “DAO” hack is something everyone should become aware of. This resulted in Ethereum actually hard forking as many Users disagreed on the Future of Ethereum, and Ethereum Classic ETC was born from that “hardfork”.

ETC has same features and functions of ETH however it has taken many turns that disable it from our USE so we will not discuss or use Ethereum Classic or ETC network at the moment. As well ETC suffered a 51% attack which resulted in some double spends. So network Security is something we must consider when choosing a Blockchain, or our Data could be compromised. And I don't want my Wallets to loose the receiving data for example.

Ethereum ETH *has remained the largest, most used, most interactive* chain and will continue this way for the foreseeable future. The difference of hard fork to soft fork is that a Network of computers actually splitting from each other and run separate versions of the initial base code program, “hard fork”. “Soft fork” is just an adoption of a number of users in the current network to a version change to the base program code. Segwit is an example of this on the Bitcoin BTC network, BCH was a hard fork BSV is a Retard Fork, lol,lol, lol. And the various “soft forks” of Ethereum; Casper, Constantinople, from inception Ethereum has been ever evolving.

Ok so you Read all that, and still wondering What is Ethereum?

<https://academy.ivanontech.com/a/17936/fLfayyHx>

Here is a Link to The Best Blockchain School on the Planet, In my opinion. Learn everything about Blockchain and get some interaction with new Students as well.



IVAN ON TECH
— A C A D E M Y —

**SecurityFundToken, SecurityToken,
AviationSecurityToken, what are they? Why would I care
about some silly Tokens? Is this just a Rip off ponzi scheme,
“like central banking”? lol hahahah not so funny**

The Security Series of Tokens are ERC-721 contracts that live on the Ethereum Virtual Machine. These contracts are Deployed to the Ethereum Blockchain and provide us with some specific features unlike some Random ERC-20 fungible token contracts. So we should discuss the types of “Smart Contracts” and the format which smart contracts are written or rules that they follow in order to have a Standardization across the Blockchain.

Standardization makes everything easier when all the users can easily interact and use the Smart contracts. Like standardized hardware (nuts and bolts), or electrical cord, makes life better and easier when everyone uses the same Standardizations. Ethereum currently has multiple standardizations for the Smart Contracts, however ERC-20 fungible and ERC-721 Nonfungible token standards are the most popular and is what we will discuss here.

ERC-20 Smart contracts that are deployed on Ethereum basically all have fungible tokens associated with them. This was the ICO craze of 2017 when everyone made their own coins and there was an explosion of Ponzi schemes and bad actors entering this space to take advantage of the growth of uneducated participants. Unfortunately a few fungible tokens ended up in my wallets as well [TRM2 \(Terraminer\)](#) for an example, an ICO for a mining company that went Bust. The tokens exist still, but the company does not and the Ethereum traded for the tokens is Gone. Such is the case for Thousands of projects. What can I do with the [TRM2](#) tokens? Well nothing really, just send and receive if I want to waste Gas money to interact with that contract. The

company is gone but the crap token is still in my wallet. Well what is the fungible token anyway? It is a part of the Smart Contract that was deployed onto the Blockchain, this contract will keep track of and handle all the transactions regarding those [TRM2](#) tokens, forever! Until Ethereum changes the basic functions this “Smart Contract” that was a ponzi will live and survive on the Blockchain FOREVER, Even after the bad actors ran away with the start up funds. So in end effect ERC-20 is a token standard Contract that has ability to issue fungible tokens, all tokens are the same, and most likely they don’t actually Do anything, other than posses the ability to be Transacted (some ERC-20 contracts have more functions). Utility tokens as some might want to advertise, meaning you would need some of these tokens to Utilize specific websites or businesses.

[ARN Aeron Token](#) is an example of ERC-20 standard fungible token “Smart Contract” the tokens in this case are more than likely intended to be issued as utility or as a vehicle to gather investment funds for this company. The actual use of these types of tokens is very limited though, send and receive the tokens and that is all. *“The improvement in aviation safety today can be achieved by using blockchain technology. It will exclude the possibility of loss, distortion or forgery of vital log data, therefore significantly reducing the risk of fatal accidents. Information should be transparent and available for the authorities to improve safety.”* [Aeron.aero](#)

[CBM Crypto Bonus Miles Token](#) another ERC-20 standard token. *“Best Airline Rewards Programs Airline rewards programs offer excellent perks like free checked bags, free hotel stays or airport lounge access while also making it easier to rack up points or miles and save money on flights.”* [Cryptobonusmiles.com](#)

I am not affiliated nor do I have permission from these Businesses to discuss them, however they are good examples of ERC-20 tokens and some of the Utility functions that they can offer will be provided by the issuer of these tokens.

[HEX](#) on one hand has an actual use case built inside of the Contract of the fungible tokens. Not affiliated as well, but I have dabbled with the token and enjoy the game theory and mechanics of the actual “Code”. This token is as well not having any central business, however we will see how the start funds will be moved, in any case the starting funding is for sure gone forever. And only the code will remain, in my case I hope the code value goes up hahahahah.

<https://etherscan.io/token/0x2b591e99afe9f32eaa6214f7b7629768c40eeb39>

So what are these “Security” tokens? Actually there are different and new Token standards for “Security” tokens ERC-1400 as an example. The name for [AviationSecurityToken](#) was only a name and a name that stuck, its based on the “Security” of the Blockchain it operates on and the Fact that it will “Secure” whatever you want to add to it. These tokens [AviationSecurityToken](#), [SecurityToken](#), and [SecurityFundToken](#) are just the names of the ERC-721 Smart Contracts deployed on [Ethereum](#).

The choice to use ERC-721 as the standard for these Contracts was a simple one, due to the Use case intended for each Token. So the Contracts are not a Ponzi, and are not Reliant on anyone except the user of the Tokens. In essence the design idea and concept was to produce a fully Decentral way for Users to “addTokenData” ie a Certificate, License, Title Document as they would need or desire. However the current Stage of Standardized, document issuance is still in Development. That’s where We as Users of the Technology come in, we must Develop, maintain, and Teach others about the Capability that this Tech brings us all.

Each ERC-721 standard token, has the Benefits of being a single serialized unit. Each is Non-Fungible and each is Unique, as something we have already in our current document issuance. My [AML](#) is different than yours, and that Pilot has a different license all together. So for the Aviation Industry we can see how an individualized and customizable Token would be a huge Benefit. If we can add the immutability of Blockchain it further secures the Authenticity of such Documents. A few other Document type Tokens and Services exist however the use of these other tokens is Limited to your interaction with such Companies. The intent for this Development is a Decentral interaction and Decentral way of issuance and maintenance of such Tokens (Document references).

The Decentralization of the Project will allow anyone to use it, anyone to have specific rights within the System, and will not rely on any Single entity to ensure its success or be a cause of failure. The system should be useable by World Wide authorities, Individuals, and Businesses without any Single point of responsibility other than the Smart Contract.

So what is the Difference Between the tokens [AVNS](#), [SET](#), and [SFT](#)? We start with the differences in Development. AVNS was the start of Development

with only Aviation in Mind. [AVNS](#) is restricted by its own code and by the functionality of Ethereum Blockchain. Basically it is the first and prototype Token, and the other tokens [SET](#) and [SFT](#) are all based on the first Developments achieved with AVNS.

The intent of [AVNS](#) is to provide a way Aviators can add Documents to Ethereum Blockchain, secure the Authenticity of the Documents, and possess a Blockchain enabled system of governance in Aviation. So the Basic old Smart Contract of AVNS is more than 600 days now in Operation and still going. The lessons learned from the Deployment and Use of this Smart Contract have enabled further development in the Blockchain Space.

So what can I do with the Token [AVNS](#)? As the user owner of a Token I have the ability to “[addTokenData](#)” which is the end effect I was hoping to achieve. I can add my License information to the Ethereum Blockchain, store it there for anyone to see, or Verify the Authenticity.

“[getTokenData](#)” function is used to find the data that was previously added to the Blockchain. If I enter the #3 into that data field the result will be the information that is Securely Stored Immutably on Ethereum. Or enter the #8 to find the information about the Data on Token #8. The Certificate data is available for use to see as well a location for the Image of the Document. We are limited by block and transaction size, that’s why the original document image is not actually on chain, however we can still use the Blockchain to Verify and Authenticate the Original Documents. Which is still a leap forward in Decentral Development, for this activity of Documents Authenticity, and Immutability.

The next and Upcoming Smart Contract will be a Replacement for the old [AviationSecurityToken](#), and enable the ability for more participants to assist in the Design and Development of the Contract, so we all, will Build something we all can Use. Initially I propose the new name to be BlockchainAviatorToken. The details regarding the functionalities and Who will have What authorizations in the Contract are also still in Development. Addition of Identity Block of information is planned for Each Token!!

At the moment the differences from SET, SFT, and AVNS are only the developmental processes and other functions that I was personally seeking to

achieve. A specifically decentral system Fully Automated, on Ethereum, provided Eth network is still operational. So after Deployment and Use of AVNS some lessons learned were then applied to the further tokens, and each time new capabilities were added and tested, on Live working Ethereum Main Chain.

As the roll out of Ethereum 2.0 is being released and tested also there will be more lessons learned and potential failures of any piece of the Code or chain functionality. So we, eagerly are awaiting the transition into a Proof of Stake concept and operation on Ethereum. Currently Bitcoin and Ethereum are similar in that they use Proof of Work to secure the Blocks on the Chain, with the new version, soft fork, of Ethereum into ETH 2.0 we will see differences in Contract Operation across the network. Gas pricing is one function that will be affected, and Gas prices are an extreme important Design consideration when writing the Contract Code. The Gas costs to operate each contract and the details of such can affect the Contract if you will have Successful additions to the Blockchain or endless failed transactions and not adding the information as desired to the Blockchain.

Some Use Cases of the three tokens can be found here, SecurityFundToken.com. The pictures are not exactly the greatest, however the concept and functionality of the tokens can be found.

Development of Direct interacting website is in process, Currently only the Smart Contracts exist, and all interactions with them must be accomplished through currently available interfaces; ie, [Etherscan.io](https://etherscan.io) using ([MetaMask.io](https://metamask.io) ext), or [MyEtherWallet.com](https://myetherwallet.com)

At some point a Directly interacting Wallet or Website user Friendly , Token Specific, User interface will be up and Running. The Backend work to connect such designs and keep everything still Decentral is a bit daunting to say the absolute least.

Alright So If you read this far Maybe you are still interested. A Question you might have is “Yes I am interested in

Blockchain, but How can [AviationSecurityToken](#) help me now, today?”

We know that Blockchain keeps my Data (transactions, and such) safe, by having a way for networked computers to communicate the Data and without fear of some entity tampering with or Changing the original Data set. And we know this is somehow accomplished with hash function and Block creation.

We discussed slightly what a smart contract is and why we use Ethereum due to the ability of our “Smart Contract Code” to operate automatically in this network. We must also understand that the Economic way for all this to actually occur Automatic and Decentral, is by use of Gas which is spending small amounts of Ethereum Token ETH. But don’t worry ETH has 18 decimal places so if we buy just 0.053078 ETH we should have enough Gas to perform several Transactions (Data input) on chain. Depends on the Contract code and network costs derived from the Miners. So again, we are operating on a Pay as you Go system, that is Immutable, and Decentral. We pay for the transactions we wish to make, then our transactions have a record on the Blockchain. This record is what I or We really desire. This is our end station point, a record of my license or your license or perhaps something else.

Ok enough Blah Blah Blah, How to do it? “What to do to get a token in my hands with my license” asks Grease Monkey Mechanic or Joe Schmoe Pilot. Actually your Token “will live on chain” in your Ethereum Wallet, so you don’t actually have a copy in your hands to spill coffee on or get lost. But you do need an Ethereum Wallet, my own recommendation ([MetaMask.io](#)). But without getting into a Masters Degree course of Study on Wallets and Security of Wallets we must stop the discussion. Please research on your own Wallets, and private Key Security! This is actually the **Most important** thing anyone can learn entering the Blockchain space, “**How to store and keep safe your Keys?**”

But for our case of Grease Monkey Mechanic and Joe Schmoe Pilot, we will need a Wallet that interfaces with ERC-721 tokens, or will at least interface with the Smart Contracts directly. That is why I specifically recommend ([MetaMask.io](#)) or ([MyEtherWallet.com](#)), however MetaMask is much easier for a beginning User, in my opinion. So once we have some Wallet on our Device, lets say a Laptop, we need to have a small amount of Ethereum in the Address to

pay for our Gas, the Miners need Gas money too. That's how this whole economy of the Blockchain works. Any way, we take the Ethereum in the Contract and we can initialize a Transaction with it. So we broadcast to the miners who are building the Blocks out of the Transactions and request that our Transaction of "[addTokenData](#)" will be added to the next mined Block and become permanent in the Chain.

Ah! But I can't just [addTokenData](#) if I don't have a Token, so I need a Token and then I can put some information on it. So in the Case of [AVNS](#) or [AviationSecurityToken](#), I need to ask Isaac, because the Token Mint or creation of each Token is Manager Controlled. Or if I interact with [SET](#) or [SFT](#), I can pay some Ethereum to the Contract and it will automatically mint a token for me and send it to my Address. No manager required for that function of new token minting on those Smart Contracts. So now we start to find some real differences between [AVNS](#) and [SET](#) or [SFT](#). Basically the idea and Concept for the New BlockchainAviatorToken will most likely have a similar way to manage Token minting in a Decentralized way.

Not only the mint is Decentral managed but addition of the information on each individual Token that is minted is as well Performed in this case by the User / Owner, (an Issuing Authority verification can also be an option and for sure is the intent, once enough Authorities can agree to follow suit.)

So once the Token is in your possession, the key holder is authorized to "[addTokenData](#)" as Desired and add this to the Blockchain with the confirmed Transaction. Retrieval of this transacted information is done with the "[getTokenData](#)" case of SET or "[getTokenData](#)" for [AVNS](#) you will need to input the Token number you want to get the information about. tokenId (uint256) Like id #8 for [AVNS](#).

So as for the "[addTokenData](#)" to get my [License](#) actually referenced on Blockchain, I will need to Type or Copy the License information to the line "liscence (string)" wrong spelling check Code before Deployment hahahahahaaha.

Then in my case I added and image of the Document to [imgur](#) a Decentral location for the image to be stored, unfortunately the Ethereum chain will not support putting the entire Image information onChain directly and if you could do it, it would be extreme high Gas cost.

As a future Design consideration for the new BlockchainAviatorToken, it would be beneficial to have an interface with some Web 3, storage space for the Images that we need to have Referenced. The interface should be included as part of the front end of this Dapp.

So "[addTokenData](#)" and of course, connect the MetaMask wallet and approve the transaction request once you click Write.

In my opinion, yes we Certainly should replace traditional Finance. Immediately!!

FIAT should go the way of the "Stuck in the Mud" concept, and people who dont find the exit strategy of Digital, Immutable, Public Blockchain, financial technology.

Of course thats why I am here BuidL Yeahh yeah



[Ethereum \(ETH\) Blockchain Explorer](#)

[Contract Address 0x73cc407fbae89d69f20cf15d51aa98171dc5703c | Etherscan](#)

The Contract Address 0x73cc407fbae89d69f20cf15d51aa98171dc5703c page allows users to view the source code, transactions, balances, and analytics for the contract address. Users can also interact and make transactions to the contract directly on...

I can easily see a future where the stores, landlords, even banks will take and use a Handfull of Digital Tokens, like Bitcoin, Ethereum, Litecoin, and Dash

some of the DeFi Contracts have yet to really go through the tests of Time so, its a waiting Game

Its a fight to the Top at the moment