Hello, my name is Ernesto Lee from BlockIQ and today I want you to get really clear on the Token Taxonomy Framework and more importantly, how it can benefit **your** **business**, let’s start with **where** the TTF comes from and **why** it was created

---

I want to say that it started with Satoshi Nakamoto but that is not really where the need was clear for TTF. This story really all started with the ICO craze.

Since around 2016, startups and others really, decided to raise funds for their business using the new Initial Coin Offering method which consists of giving tokens instead of shares to their investors using blockchain technology.

That way the startups could raise capital \*quickly\* with lower costs and without as many legal requirements. Remember, this was a new concept and government agencies needed time to regulate the space. Plus, with ICOs you don't have to sell a portion of your company since you dictate the rules of your token. It was thought of as an easier way of bootstrapping your business.

---

The result of this was that Massive amounts of tokens were created!

You could choose to give tokens to your investors with voting rights. You could choose to give tokens whose only purpose is to give you money … kind of like a fundraiser. We could choose to create tokens that could be used inside your application as a payment method… Or tokens that operated like Disney Dollars... The uses of tokens are endless.

With So many people coming up with new and unique applications of blockchain tokens, it became obvious that this was starting to get a little crazy. Blockchain in general, and blockchain tokens in particular were clearly succeeding. It was obvious that blockchain tokens were here to stay. That’s why...

---

New token standards surfaced.

The incentive is what created many new token standards, particularly on the Ethereum blockchain called ERC20, ERC721 and others since we needed a way to make sense of all these new tokens. The Enterprise Ethereum Alliance played a crucial role in this epoch of token evolution.

Without token standards, companies were forced to create their own interpretation of these tokens which meant tokens were incompatible with each other. You couldn't have a single wallet with all of your tokens.

You had to have a software wallet specifically created for each token. That wasn't ideal since most tokens required a similar set of features to be sold on an ICO or to interoperate. That's why standards were created. To unify all the tokens and make them compatible with each other.

---

Tokens Provide many benefits to businesses and investors.

Standard, in particular the ERC20, allowed the creation of wallets such as Metamask that can hold any token that was offered and it also gave birth to token exchanges such as EtherDelta where you can sell your tokens in exchange for others.

Businesses didn't have to invest thousands of dollars in creating their own wallets and tokens while others could trade their tokens and keep them in general wallets such as metamask or myetherwallet instead of having to setup new software for each individual token.

---

Now you may be asking yourself "Why does this matter?" and that's a good question because so far, we haven't talked about the token taxonomy framework...

---

The answer is that right now, in 2020, we have many standards on many blockchains. The ecosystem has expanded over the years and because of that we are facing new problems...

---

The first is that there are Too many standards: There’s a new standard created almost everyday for very specific business cases. People don't have time to understand what's the purpose of an ERC20, an ERC721, a Security Token or a Smart Asset. We need a simpler way to understand them.

---

Also, there are several blockchains: Each blockchain has its own standards which makes it even more difficult to understand these tokens because of each blockchain’s limitations, we can’t know for sure if a specific protocol will work. The underlying structure is different making standards unique on each blockchain.

---

There are many new participants in the blockchain landscape: New people are getting into the space to utilize the benefits of blockchain or to invest in crypto by buying tokens. Often times, people or confused and tune out as soon as they hear "ERC20" or some other equally intimidating sounding acronym. We must simplify the ideas behind the blockchain functions. Business users, Investors just as developers need a way to learn what a token is doing without having to get into the technicalities.

---

Too many options: It's a difficult industry to get into. What if one day, you as a business owner, want to create your own token for selling smart lottery tickets to your customers that can be quickly verified in the blockchain? You have too many options and it would take you forever to identify the right platform for your needs.

---

Too much technical information: Because of the nature of blockchain and because these are new concepts, the only way to truly know what a token standard is doing is by diving into pages and pages of technical whitepapers that are understandable only by developers familiar with the platform making tokens seem obscure to the average consumer.

---

We need a way to **simplify** and **abstract** the main ideas behind the tokens to bring order to this chaos...

---

Presenting the Token Taxonomy Framework.

---

Presenting the token taxonomy framework. A solution to the token problems we are facing to help business create their own standards for their specific needs with a simple plug-and-play interface by taking just the features they need in a simple to understand framework.

---

Taxonomy is the science of naming, defining and classifying groups of biological organisms on the basis of shared characteristics.

It’s called the token taxonomy framework because we are classifying token functionalities into groups of shared characteristics meaning we take a token, we break it down into functions and we generate new tokens by connecting these pieces together like legos.

---

The first and most important question we need to answer is “How does it **benefit** meas a **business owner**?” Because the framework will be as good as the benefits it provides to its users.

---

Here are some benefits that you will get by using the token taxonomy framework.

Create your own **token standards** for your industry. You can define your own specification and use it to create tokens with the taxonomy framework.

Understand how tokens work **intuitively** by breaking them down into smaller components while using simple formulas to quickly understand what functions are implemented in the token smart contract and how it works without requiring a technical background.

Enhance your business with **smart tokens** and blockchain. Tokens can be used to improve existing industries such as lotteries, tickets, currencies, loyalty points, unique identificators and tracking numbers. Any type of data that can be represented digitally can be a token.

---

Here’s a high-level overview of the token taxonomy framework to see the main elements that define the framework.

Essentially the goal is to create a formula with the key components, in the next slides we’ll see an example.

The entire token functionality is represented in that formula which allows us to quickly identify how a token will behave and how it will be used since all the modules are described independently.

You generate those formulas by combining token artifacts and behaviours among other things that we’ll see in a moment.

We generate artifacts and behaviours from existing tokens by simply breaking down their functions into comprehensible parts that can be reused in other situations. You can create your own artifacts. That’s why it’s called the token \*taxonomy\* framework, because you’re extracting the pieces that make a token and you combine them to create new tokens.

---

Now that we have a better understanding of what is the ttf, let’s see some definitions to begin comprehending how it works.

First, we have artifacts which are each independent piece of the token definition. Every single element is an artifact.

---

Then we have templates which are the token definition formulas from which you generate new tokens. Templates are document describing the token and they are composed of 2 elements:

**The template formula**: which is the token definition **formula** from which you generate new tokens.

**And the template definition**: which a written document explaining what each artifact of the token means so that it’s perfectly understood. We need this to translate what each term used in the formula is supposed to mean to facilitate the complete understanding of a particular token.

Now templates can be used to generate...

---

Token classes which are the deployed token contracts on a specific blockchain. For example, a token with the symbol TRK could be deployed on the Ethereum blockchain.

Don’t confuse token classes with token instances. A token instance is 1 single token generated by a token class.

---

Next we have the base token artifacts. These are the base structures from which you build tokens. When you want to create a new token template, you start by taking a base token artifact that can either be fungible, non-fungible or hybrid.

Fungible tokens are those that don’t differentiate from each other. For instance, the USD dollars are fungible because each dollar is identical to the next. In the Ethereum blockchain they are defined as the ERC20 standard. You define fungible tokens with the symbol **tF**.

Non-fungible tokens on the other hand, are those that are unique from each other. For instance, art pieces. Each painting is generated from a blank canvas to generate a unique artwork that can be sold. Those are non-fungible. They have been used in a popular project called cryptokitties where users could purchase unique virtual collectible cats to be re-sold for a profit. You define non-fungible with the symbol **tN**

Finally we have the hybrid tokens. These tokens share properties from the previous ones. For instance, a movie ticket. Each movie will have a unique, non-fungible token and you’ll use that token to generate fungible copies that are identical from each other. You define them by typing **tN(tF)** or **tF(tN)**

---

Here’s a visual respresentation of how tokens are made to understand the deployment structure. First you create a token template which is made of a template definition and a formula. The formula specifies all your token configuration while the definition is used to describe each artifact with the specific parameters. Then you have classes which are deployed tokens on a specific blockchain. Because the template is a general specification, you can deploy token classes on any blockchain of your choice as long as it uses Smart contracts. Instances are each independent token that users get.

---

Behaviours are functionalities that your contract can execute. Thing like the Pausable behaviour meaning you can pause your token contract stopping all transfers, the Mintable behaviour where you can create new tokens to increase the total supply similar to what the USD dollar does and the Burnable behaviour where you can remove tokens from circulation, reducing the total supply.

Those are some behaviours but there are more and you can create your own. Behaviours are represented in lowercase letters such as m, p and b for those previous cases.

---

Behaviour groups then are a combination of those behaviours. They are represented with uppercase letters for instance SC which means Supply Control and includes the Mintable (m), Burnable (b) and Roles (r) behaviours. You can create your own groups to simplify common functionality since it happens that you repeat things across many different types of tokens.

---

Symbols are unique representations for your tokens. Think of them as short names to identify your unique token. For instance the USD or EUR symbols can be used as tokens.

Finally the property sets are special properties that you add to your token definition to store special data fields. These are necessary elements for your token to function.

Remember that you don’t have to remember all this. The token definition formula always has a descriptive document explaining what each term means and how it is used.

---

Finally the property sets are special properties that you add to your token definition to store special data fields. These are necessary elements for your token to function. For instance: the SKU unique identifier in ecommerce products. Represented with a phi greek letter (Φ) and the name: **ΦSKU.**

Remember that you don’t have to remember all this. The token definition formula always has a descriptive document explaining what each term means and how it is used.

---

Let’s see now how to build token formulas by seeing a quick example with each component explained. Don’t be alarmed by this because it may seem confusing at first, but it will make sense in a moment. At the top you have a currency token that emulates general currencies like Dollars or Bitcoins meaning you can create unlimited copies and burn them.

---

This is the syntax. You start by writing the token base type. This is a fungible type of token meaning you can create endless identical copies, all the tokens are the same just like each dollar is identical to the next.

---

Then inside curly brackets you write the behaviours. In this case we have the subdividable behaviour which is represented with a tilde and a d letter. Subdividable means that you can break the token down into smaller parts. For instance, if your token has 2 decimals, you’ll be able to send 0.01 worth of tokens which gives you flexibility to deal with smaller units that can be transfered easily. As a fun fact, Bitcoin has 8 decimals and Ethereum 18.

Then we have the transferable and burnable behaviour represented by a lowercase letter g. This behaviour allows you to transfer tokens to other users and to burn some from the total supply.

---

Next, we have the Supply Control behaviour group presented by an uppercase SC. The Supply Control behaviour group is used to manage the number of tokens in existance with the mintable, burnable and roles behaviours. That way you can create tokens, burn them and setup special user roles to manage those functions so that only trustworthy individuals can modify your token supply. Remember that behaviours are always in italics in lowercase and behaviour groups in uppercase italics.

---

Finally we add the property sets. You can add as many as you want with additional plus signs. If you remember, property sets are special data variables that we use to keep track of interesting elements. In this case we are adding a Stock Keeping Unit property set and we’re defining it as a 15-character word starting with a capital T. Each token will define the specific values that can be used in each property set using the template definition document, similarly to how it is described here.

Note that all the elements described here are artifacts.

---

Let’s take a look at a real example. Emoney is a project that has been created by Julio Faura and Daniel Lehrner whose purpose is to “enable the issuance of regulated electronic money on blockchain networks and its practical usage in real financial applications”. Basically a general currency that imitates fiat money.

This is extracted from an oficial document in the tokentaxonomy.org website where they publish all the different projects. Every token is described in this 66 pages paper containing extensive information about the token so that you can really understand with complete clarity how it is used, what are the uses of the token and how it works. The first thing we see is the token classification. In this case this is a Fungible token that can be fractioned into smaller units, meaning it can use decimal places, the value comes from a reference and the representation type is common.

The value type reference means that the token is valuable because it can be used for purchasing items such as a house, a ticket or food. The token on itself doesn’t have any value, it just comes from the reference which in this case is it’s utility. The more people buy the token, the more valuable it will become and the more purchasing power it will have to obtain goods in exchange for it.

Then the representation type common means that the token is represented as a number and it is assigned to a token owner using blockchain addresses which are unique to each wallet. You can create as many as many accounts as you want as long as you own the private keys of that particular account. In blockchain the private key is like your passwork but stored in an encrypted text. So what the common representation classification means is that all the transactions you execute with your token, get recorded on the blockchain as the ledger and all the tokens are identical.