TLM - PRELIMINARY COMENTS

All of the findings were addressed in the contracts. On the next page of this text, we updated the project’s documentation in order to clarify it.

**GLOBAL-01:** perhaps we are misunderstanding this issue, but every file has the compiler version at the first line.

**GLOBAL-02**: a system that recreates a multisignature wallet added to the outOfSale() function. Two out of three signatures will be needed to execute this function. If the admin wallet is compromised, the hacker won’t be able to withdraw the TLM at the vault and convert them to BTCb.

**TLM-01:** fixed

A Boolean variable was added. When the function mint() is executed, the variable changes to false, preventing minting again.

**TVT-01:** fixed

**TVT-02:** fixed

**TVT-03:** fixed

**TVT-04:** fixed

**TVT-05:** fixed

The tax was missing when redefining tlmWithOutBacking and tlmWithBacking

**TVT-06:** The procedure is the one assumed in the description. The 0.5% is a transaction commission. It was done this way because the project administrator thought it’s a simple way of doing it.

Another way if putting it, people buys xx amount of BTCb in the form of TLM, but receive a little bit less because of the commission.

**TVT-07:** fixed

Same explanation asTLM-01

**TVT-08**: fixed

**TVT-09:** yes. tlmWithBacking + tlmWithOutBacking = TLM.totalsupply

**TVT-10**: fixed

TLM

The idea of the project is the creation of an ERC-20 token (TLM) with a fixed supply.

The TLM will be backed with BTCb.

There will be a Vault in which the admin will make the first deposit of bBTC and this will mint the TLM at an initial first rate.

Along with the Vault, there will be a store (TIENDA), in which people will buy TLM with bBTC. This bBTC are transferred to the admin address.

The company will mine BTC, and will distribute this benefit along TLM holders. This is going to be done this way: once the BTC are mined, they’ll be bridged to the Binance Smart Chain, generating bBTC and transferring them to the Vault. This process will decrease the rate (TLM / bBTC) making possible for the holders to swap its TLM for a bigger ammount of bBTC.

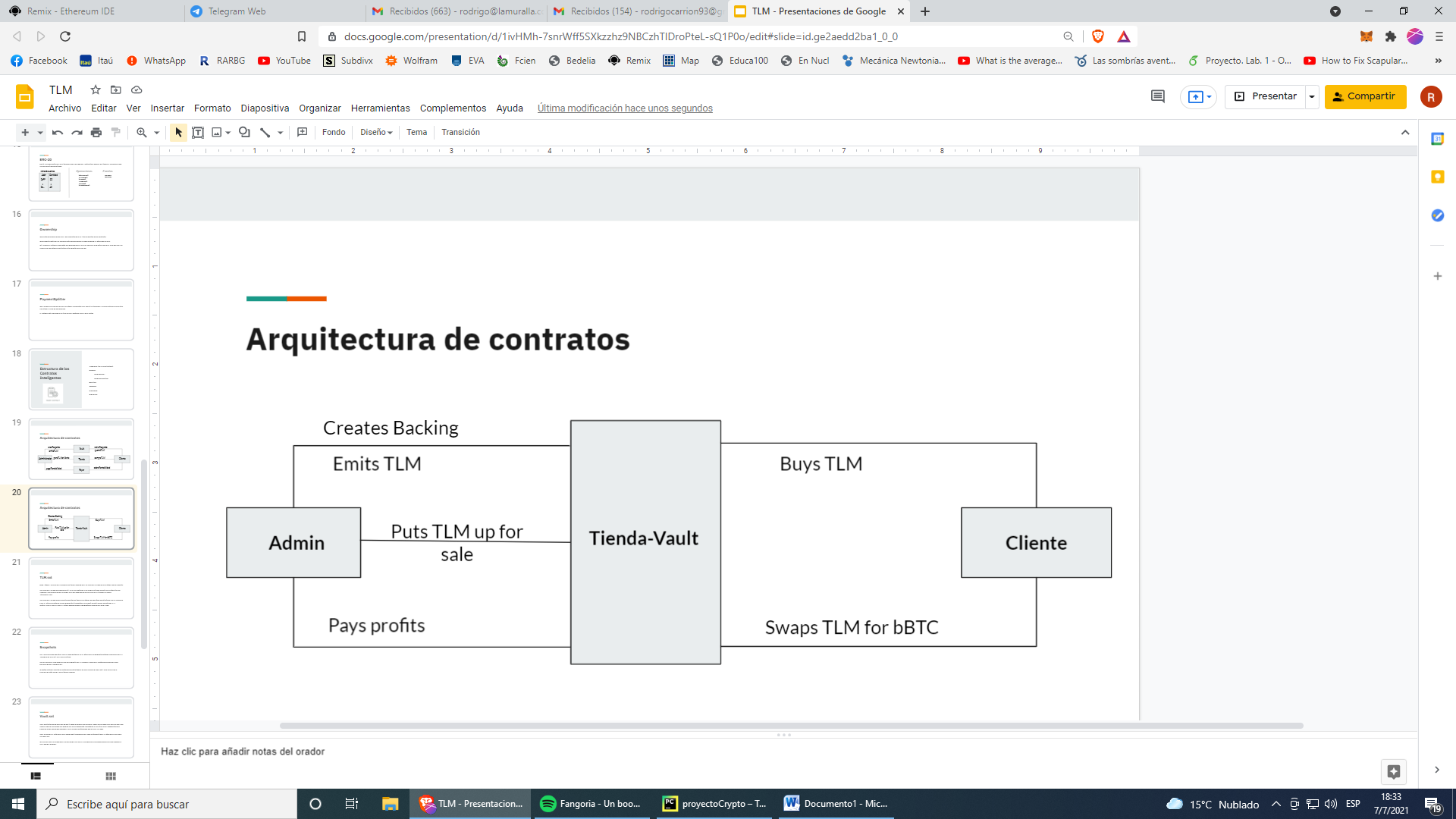
At every moment people must be enabled to swap the TLM at the vault, at the exchange rate. Transactions have a 0.5% of commission, which is also transferred to the admin.

TLM is going to be listed in Pancakes as soon it’s deployed. Eventually, it will be listed in many other exchanges.

Overtime, TLM holders must be consulted with many decisions involving the future of the coin, so it will be linked with the snapshot platform. <https://snapshot.org/>

Inside the .zip file there are 5 files:

* **ERC20** (used to simulate bBTC)
* **TiendaVault** (The store and the vault are in the same smart contract)
* **TLM** (inheritance of the functions of TLMbase)
* **TLMbase** (ERC20 contract, only its title is modified)
* **Vaulteable** (The Ownable contract with its title modified, in order to have a second one, because some functions must be only executed by the Vault)



To clarify some design concepts:

1. TLM will be only minted once. Its supply is fixed, (not minted and not burned) in the whole process.
2. The centralization problem is inherent to the project because of how it was created. Anyway, is important to us to make clear that if someone buys TLM there is no way for them to lose their BTCb.
3. Being clear that investors can’t lose their funds, the multisignature process was added to protect the project admin’s initial funds. In the case the main wallet is hacked, and TLM are being held in (TIENDA-VAULT), this mechanisms prevents the hacker to execute the function outOfSale(), and then converts them.
4. Because of (1), when people convert their TLM to BTCb, the TLM go directly to the vault, but the exchange rate must not be affected by this.   
   That’s why tlmWithBacking() and tlmWithoutBacking() were created, this way a registry can be maintained.  
   tlmWithBacking() + tlmWithoutBacking() = totalSupply