Create Complex Token function, creates a Publicity Token with 11 state vars set on the !!!! ATENTION !!! token: - **state 0**: VAULT address - **state 1**: Publickey 1 vault script The TOKEN is created with a embedded script that controls all life cicle of the TOKEN based on - **state 2**: Publickey 2 vault script - **state 3**: Publickey 3 vault script his state and the state vars values, and protects itself as the transaction will must satisfy the - **state 4**: DAO address script conditions. - **state 5**: DAO publickey Though so far is created without this script, but - **state 6**: % Third party comission of the rewards to be received by DAO - **state 7**: % Final user comission of the rewards to be received by DAO all Tx and operations are done like the script - **state 8**: % of the value of rewards if are not minimas to be received by DAO were present, so adding the script it should work - **state 9**: Price of the token as well the right positions of inputs and outputs - **state 99** : A specific name for the token are done. - **state 100**; [MDAE] It means is a publicity token «statemachine» **Token States** When the token is bought it adds three more state Create Adverting Token vars before sending the token to the buyer: state: empty (0) - **state 10** : Buyer address - **state 11**: Amount of tokens to buy - **state 12** : Buyer publickey Buy Tokens from Advertiser Not Configured (0) ConfigureCampaign function creates an script based on the TOKENID & Camaign name, configures all state vars with the user input values, and send the choosen amount of configured tokens building a TX SPLITTING in units of one the first 150 units, to the Configure Campaign Configured (1) SCRIPT address, . - **state 13** : CAMPAIGN name - **state 14**: State of the token (0,1,2,3) non configured(empty), - **state 15**: Amount of rewards to be received by the Sent (2) Send Campaign dapp(developer) - **state 16**: Type of rewards dapp (tokenid) (usualy 0x00) - **state 17**: Amount of rewards to be received by the final user - **state 18**: Type of rewards final user (tokenid) (usually 0x00) - **state 19**: Type of advertising (0-6) text(23-31), web, image, survey, actions, contest, others Token watched & Pay Out Used (3) - **state 20** : Content according to type of publicity Web Page / - **state 21**: Actions (Interaction on twitter, discord, social networks) - **state 22**: Conditions on what needs to be done to considered a publicity watched (click) - **state 23** : Text 0 publicity - **state 24** : Text 1 publicity - **state 25** : Text 2 publicity - **state 26** : Text 3 publicity - **state 27** : Text 4 publicity SendCampaign sends one publicity token from the choosed - **state 28** : Text 5 publicity campaing to a final user adrdress script (generated from the - **state 29** : Text 6 publicity combination of user address and a developer address) as well - **state 30** : Text 7 publicity as wiht the rewards the token state vars indicate, as is the final - **state 31** : Text 8 publicity user'script that would hold the publicity tokens and the - **state 99** : A specific name for the token (not used so far) rewards to be payed when the publisity is watched. - **state 100**; [MDAE] It means is a publicity token

When the publicity is watched, the function ** get_publiciy_token_rewards ** pays all parties his rewards and send back token to the advertiser building a TX that satisfys what token state vars says, as the rewards and the token are into an script.